

UM1718 User manual

STM32CubeMX for STM32 configuration and initialization C code generation

Introduction

STM32CubeMX is a graphical tool for STM32 products. It is part of the STM32Cube initiative (see *Section 1: STM32Cube overview*), and is available as a standalone application as well as in the STM32CubeIDE toolchain.

STM32CubeMX has the following key features:

- Easy microcontroller selection covering the whole STM32 portfolio
- **Board selection** from a list of STMicroelectronics boards
- Easy microcontroller configuration (pins, clock tree, peripherals, middleware) and generation of the corresponding initialization C code
- Easy switching to another microcontroller by importing a previously-saved configuration to a new MCU project
- Easy exporting of current configuration to a compatible MCU
- Generation of configuration reports
- Generation of embedded C projects for a selection of integrated development environment tool chains (STM32CubeMX projects include the generated initialization C code, MISRA 2004 compliant STM32 HAL drivers, the middleware stacks required for the user configuration, and all the relevant files for opening and building the project in the selected IDE)
- Power consumption calculation for a user-defined application sequence
- Self-updates allowing the user to keep STM32CubeMX up-to-date
- Download and update of STM32Cube embedded software required for user application development (see *Appendix E* for details on the STM32Cube embedded software offer)
- Download of CAD resources (schematic symbols, PCB footprints, and 3D models)

Although STM32CubeMX offers a user interface and generates C code compliant with STM32 MCU design and firmware solutions, users need to refer to the product technical documentation for details on actual implementation of peripherals and firmware. The following documents are available on *www.st.com*:

- STM32 microcontroller reference manuals and datasheets
- STM32Cube HAL/LL driver user manuals for STM32C0 (UM2985), STM32F0 (UM1785), STM32F1 (UM1850), STM32F2 (UM1940), STM32F3 (UM1786), STM32F4 (UM1725), STM32F7 (UM1905), STM32G0 (UM2303), STM32G4 (UM2570), STM32H5 (UM3132), STM32H7 (UM2217), STM32H7RS (UM3309), STM32L0 (UM1749), STM32L1 (UM1816), STM32L4/L4+ (UM1884), STM32L5 (UM2659), STM32MP1/MP2 (https://wiki.st.com/stm32mpu/wiki/STM32CubeMX_release_note), STM32N6 (UM3425), STM32U0 (UM3307), STM32U5 (UM2883), STM32WL (UM2642), STM32WB (UM2442), STM32WBA (UM3131), and STM32WB0 (UM3363).





February 2025

UM1718 Rev 47

Contents

1	STM3	2Cube	overview	25
2	Gettir	ng starte	ed with STM32CubeMX	26
	2.1	Principle	es	26
	2.2	Key fea	tures	28
	2.3	Rules a	nd limitations	30
3	Instal	ling and	d running STM32CubeMX	31
	3.1	System requirements		
		3.1.1	Supported operating systems and architectures	. 31
		3.1.2	Memory prerequisites	. 32
		3.1.3	Software requirements	. 32
	3.2	Installin	g/uninstalling STM32CubeMX standalone version	33
		3.2.1	Installing STM32CubeMX standalone version	. 33
		3.2.2	Installing STM32CubeMX from command line	. 41
		3.2.3	Uninstalling STM32CubeMX standalone version	. 42
	3.3	Launchi	ing STM32CubeMX	44
		3.3.1	Running STM32CubeMX as a standalone application	. 44
		3.3.2	Running STM32CubeMX in command-line mode	. 44
	3.4	Getting	updates using STM32CubeMX	48
		3.4.1	Running STM32CubeMX behind a proxy server	. 49
		3.4.2	Updater configuration	. 50
		3.4.3	Installing STM32 MCU packages	. 52
		3.4.4	Installing STM32 MCU package patches	. 53
		3.4.5	Installing embedded software packs	. 53
		3.4.6	Removing already installed embedded software packages	. 58
		3.4.7	Checking for updates	. 59
4	STM3	2Cube	MX user interface	62
	4.1	Home p	age	62
		4.1.1	File menu	. 63
		4.1.2	Window menu and Outputs tabs	. 64
		4.1.3	Help menu	. 65
		4.1.4	Social links	. 66



4.2	New Pro	oject window	66
	4.2.1	MCU selector	. 68
	4.2.2	Board selector	. 70
	4.2.3	Example selector	. 70
	4.2.4	Cross selector	. 72
4.3	Project	page	75
4.4	Boot cha	ain (STM32 MPUs)	78
	4.4.1	Boot mode configuration	. 78
	4.4.2	Coprocessor initializers (STM32MP2x)	. 81
	4.4.3	Boot device selection (STM32MP25)	. 82
4.5	Pinout 8	& Configuration view	83
	4.5.1	Component list	. 84
	4.5.2	Component Mode panel	. 86
	4.5.3	Pinout view	. 87
	4.5.4	Pinout menu and shortcuts	. 88
	4.5.5	Pinout view advanced actions	. 90
	4.5.6	Keep Current Signals Placement	. 91
	4.5.7	Pinning and labeling signals on pins	. 92
	4.5.8	Pinout for multi-bonding packages	. 93
	4.5.9	System view	. 95
	4.5.10	Component configuration panel	. 96
	4.5.11	User Constants configuration window	. 98
	4.5.12	GPIO configuration window	103
	4.5.13	DMA configuration window	105
	4.5.14	NVIC configuration window	107
	4.5.15	FreeRTOS configuration panel	113
	4.5.16	Setting HAL timebase source	119
4.6	Pinout &	& Configuration view for STM32 MPUs	123
	4.6.1	Run time configuration	124
	4.6.2	Boot stages configuration	124
4.7	RIF con	figuration	125
	4.7.1	Configuration approach	125
	4.7.2	RIF global configurations	125
	4.7.3	Peripherals protection	128
	4.7.4	Peripheral instance protection	128
	4.7.5	IP feature protection	133



	4.7.6	Software constraints validation	. 136
	4.7.7	Masters configuration	. 137
	4.7.8	Service peripherals protection	. 142
	4.7.9	System peripherals (STM32MP2 and STM32N6 series)	. 145
	4.7.10	Memory protection for STM32MP2 series	. 152
	4.7.11	Memory protection for STM32N6 series	. 159
	4.7.12	RIF code generation	. 163
	4.7.13	Implementation of illegal access controller (IAC) feature onSTM32N6 series	. 164
4.8	Pinout &	Configuration view for STM32H7 dual-core products	165
4.9		g security in Pinout & Configuration view L5 and STM32U5 series only)	166
	4.9.1	Privilege access for peripherals, GPIO EXTIs and DMA requests	. 167
	4.9.2	Secure/nonsecure context assignment for GPIO/peripherals/middleware	. 171
	4.9.3	NVIC and context assignment for peripherals interrupts	. 171
	4.9.4	DMA (context assignment and privilege access settings)	. 171
	4.9.5	GTZC	. 173
	4.9.6	OTFDEC	. 174
4.10	Clock C	onfiguration view	175
	4.10.1	Clock tree configuration functions	. 176
	4.10.2	Securing clock resources (STM32L5 series only)	. 179
	4.10.3	Recommendations	. 182
	4.10.4	STM32F43x/42x power overdrive feature	. 183
	4.10.5	Clock tree glossary	. 184
4.11	Project	Manager view	185
	4.11.1	Project tab	. 186
	4.11.2	Code Generator tab	. 191
	4.11.3	Advanced Settings tab	. 194
4.12	Import F	Project window	196
4.13	Set unu	sed/reset used GPIOs windows	201
4.14	Update	Manager windows	203
4.15	Softwar	e Packs component selection window	203
	4.15.1	Introduction on software components	
	4.15.2	Filter panel	
	4.15.3	Packs panel	
	4.15.4	Component dependencies panel	



	4.15.5	Details and Warnings panel 209
	4.15.6	Updating the tree view for additional software components
4.16	LPBAM	I Scenario & Configuration view
4.17	CAD R	esources view
4.18	Boot pa	ath
	4.18.1	Available boot paths
	4.18.2	Creating a boot path project: an example
	4.18.3	How to configure an OEM-iRoT boot path
	4.18.4	How to configure an ST-iRoT boot path
	4.18.5	How to configure an assembled boot path
	4.18.6	How to configure OEM-uRoT (STiRot uROT) boot path
	4.18.7	How to configure ST-iRoT boot path with STM32H7RS devices 251
4.19	User a	uthentication
	4.19.1	Login with an existing my.st.com account
	4.19.2	Create a my.st.com account
	4.19.3	Password restoration
	4.19.4	Authentication through command line interface
4.20	About v	window
STM	32Cube	MX tools
5.1	Externa	al Tools
5.2	Compa	are Projects
0.2	5.2.1	User interface of the Compare Projects tool
	5.2.2	Comparing two projects
	5.2.3	The output of the comparison
	5.2.4	Saving the comparison result of the two projects
5.3		Consumption Calculator view
0.0	5.3.1	Building a power consumption sequence
	5.3.2	Configuring a step in the power sequence
	5.3.3	Managing user-defined power sequence and reviewing results 284
	5.3.4	Power sequence step parameters glossary
	5.3.5	Battery glossary
	5.3.6	SMPS feature
	5.3.7	Bluetooth Low-Energy [®] /ZigBee [®] support (STM32WB series only) 295
	J.J.	
	5.3.8 5.3.9	Sub-GHz support (STM32WL series only)



5

	5.4	DDR Si	uite (STM32MPUs only)	9
		5.4.1	DDR configuration	0
		5.4.2	Connection to the target and DDR register loading	2
		5.4.3	DDR testing	4
	5.5	STM32	CubeMX Memory Management Tool	6
		5.5.1	STM32H5, STM32U5, STM32WBA, STM32WBAM, and STM32WBA6 with TrustZone activated	7
		5.5.2	An end-to-end usage example	9
		5.5.3	STM32H7 single core and STM32U5 without TrustZone activated 32	1
		5.5.4	STM32WBxx	0
		5.5.5	STM32H7 Dual-core without Trust Zone activated	1
		5.5.6	STM32H7RS	3
		5.5.7	STM32WB0	8
		5.5.8	Notification MMT/boot path (STM32H7RS and STM32H5)	0
6	STM3	2Cubel	MX C Code generation overview	3
	6.1		Cube code generation using only HAL drivers	3
	6.2	•	Cube code generation using Low Layer drivers	
	6.3		code generation	
	0.0	6.3.1	STM32CubeMX data model for FreeMarker user templates	
		6.3.2	Saving and selecting user templates	
		6.3.3	Custom code generation	
	6.4		nal settings for C project generation	
7	Code	genera	tion for dual-core MCUs	
		•	ual-core product lines only)	8
8	Code	genera	tion with TrustZone [®] enabled (STM32L5 series only) .380	0
9	Devic	e tree g	generation (STM32MPUs only)	4
	9.1	Device	tree overview	4
	9.2	STM32	CubeMX Device tree generation	6
10			dditional software components using standard	8



11		rial 1: From pinout to project C code generation g an MCU of the STM32F4 series
	11.1	Creating a new STM32CubeMX project
	11.2	Configuring the MCU pinout
	11.3	Saving the project
	11.4	Generating the report
	11.5	Configuring the MCU clock tree
	11.6	Configuring the MCU initialization parameters
		11.6.1 Initial conditions
		11.6.2 Configuring the peripherals
		11.6.3 Configuring the GPIOs
		11.6.4 Configuring the DMAs
		11.6.5 Configuring the middleware
	11.7	Generating a complete C project 409
		11.7.1 Setting project options 409
		11.7.2 Downloading firmware package and generating the C code
	11.8	Building and updating the C code project 415
	11.9	Switching to another MCU 420
12		rial 2 - Example of FatFs on an SD card using 32429I-EVAL evaluation board
13		rial 3 - Using the Power Consumption Calculator otimize the embedded application consumption and more 429
	• 13.1	Tutorial overview
	13.2	Application example description
	13.3	Using the Power Consumption Calculator
	10.0	13.3.1 Creating a power sequence
		13.3.2 Optimizing application power consumption
14		rial 4 - Example of UART communications with TM32L053xx Nucleo board
	14.1	Tutorial overview
	14.2	Creating a new STM32CubeMX project and selecting the Nucleo board
	14.3	Selecting the features from the Pinout view
	17.5	



	14.4	Configu	ring the MCU clock tree from the Clock Configuration view 443
	14.5	Configu	ring the peripheral parameters from the Configuration view 444
	14.6	Configu	ring the project settings and generating the project
	14.7	Updatin	g the project with the user application code
	14.8	•	ng and running the project
	14.9		ring Tera Term software as serial communication
	14.5	•	the PC
15	Tutor	rial 5: Ex	porting current project configuration to
			MCU
16	Tutor	rial 6 – A	Adding embedded software packs to user projects 455
17	Tutor	ial 7 – L	Ising the X-Cube-BLE1 software pack
40			
18		-	BAM projects
	18.1		overview
		18.1.1	LPBAM operating mode
		18.1.2	LPBAM firmware
		18.1.3	Supported series
		18.1.4 18.1.5	LPBAM design
	10.0		LPBAM project support in STM32CubeMX
	18.2		g an LPBAM project
		18.2.1	LPBAM feature availability
		18.2.2 18.2.3	Describing an LPBAM project
	40.0		Managing LPBAM applications in a project
	18.3		ing an LPBAM application
		18.3.1	Overview (SoC & IPs configuration, runtime scenario)
		18.3.2	SoC& IPs: configuring the clock
		18.3.3	SoC & IPs: configuring the IPs
		18.3.4	SoC & IPs: configuring low power settings
		18.3.5	LPBAM scenario: managing queues
		18.3.6	Queue description: managing nodes
		18.3.7	Queue description: configuring the queue in circular mode
		18.3.8	Queue description: configuring the DMA channel hosting the queue . 478
		18.3.9	Node description: accessing contextual help and documentation 479
		18.3.10	Node description: configuring node parameters



		18.3.11	Node description: configuring a trigger	481
		18.3.12	Node description: reconfiguring a DMA for data transfer	482
	18.4	Checkin	g the LPBAM design	484
	18.5	Generat	ing a project with LPBAM applications	485
	18.6	LPBAM	application for TrustZone [®] activated projects \ldots	486
Appendix	A ST	FM32Cu	beMX pin assignment rules	490
,	A.1	Block co	pnsistency	490
,	A.2	Block in	ter-dependency	494
	A.3	One blo	ck = one peripheral mode	496
	A.4	Block re	mapping (STM32F10x only)	496
	A.5	Functior	n remapping	497
,	A.6		hifting (only for STM32F10x and when current Signals placement" is unchecked)	498
	A.7	Setting a	and clearing a peripheral mode	499
	A.8	Mapping	g a function individually	499
	A.9	GPIO si	gnals mapping	499
Appendix	B ST	FM32C u	beMX C code generation design	
			ind limitations	500
I	B.1	STM320	CubeMX generated C code and user sections	500
I	B.2	STM320	CubeMX design choices for peripheral initialization	500
I	B.3	STM320	CubeMX design choices and limitations for	
		middlew	rare initialization	
		B.3.1	Overview	501
		B.3.2	USB host.	502
		B.3.3	USB device	
		B.3.4	FatFs	502
		B.3.5	FreeRTOS	
		B.3.6	LwIP	
		B.3.7	Libjpeg	
		B.3.8	Mbed TLS	
		B.3.9	TouchSensing	
		B.3.10	PDM2PCM	
		B.3.11	STM32WPAN BLE/Thread (STM32WB series only)	514
		B.3.12	CMSIS packs selection limitation	518



		B.3.13	OpenAmp and RESMGR_UTILITY (STM32MPUs and STM32H7 dual-core products)
Appendix	C S 1	Г М 32 m	icrocontrollers naming conventions
Appendix	D S	Г М 32 m	icrocontrollers power consumption parameters 524
	D.1	Power n	nodes
		D.1.1	STM32L1 series
		D.1.2	STM32F4 series
		D.1.3	STM32L0 series
	D.2	Power c	onsumption ranges
		D.2.1	STM32L1 series features three VCORE ranges
		D.2.2	STM32F4 series features several VCORE scales
		D.2.3	STM32L0 series features three VCORE ranges
Appendix	E S	TM32Cu	be embedded software packages
Revision	history	/	



List of tables

Table 1.	Command line summary	45
Table 2.	Home page shortcuts	63
Table 3.	Window menu	64
Table 4.	Help menu shortcuts	65
Table 5.	Component list, mode icons and color schemes	85
Table 6.	Pinout menu and shortcuts	88
Table 7.	Configuration states	
Table 8.	Peripheral and Middleware configuration window buttons and tooltips	97
Table 9.	Clock configuration view widgets	. 179
Table 10.	Clock Configuration security settings	. 180
Table 11.	Voltage scaling versus power overdrive and HCLK frequency	. 184
Table 12.	Relations between power over-drive and HCLK frequency	. 184
Table 13.	Glossary	
Table 14.	Additional software window - Filter icons	. 205
Table 15.	Additional Software window – Packs panel columns	. 206
Table 16.	Additional Software window – Packs panel icons	. 206
Table 17.	Component dependencies panel contextual help	
Table 18.	Boot paths without TrustZone [®] (TZEN = 0)	. 216
Table 19.	Boot paths with TrustZone [®] (TZEN = 1)	217
Table 20.	Boot paths for STM32H7RS devices	
Table 21.	LL versus HAL code generation: drivers included in STM32CubeMX projects	. 366
Table 22.	LL versus HAL code generation: STM32CubeMX generated header files	
Table 23.	LL versus HAL: STM32CubeMX generated source files	. 367
Table 24.	LL versus HAL: STM32CubeMX generated functions and function calls	
Table 25.	Files generated when TrustZone [®] is enabled	. 382
Table 26.	Connection with hardware resources	. 464
Table 27.	Document revision history	. 530



List of figures

Figure 1.	Overview of STM32CubeMX C code generation flow.	. 27
Figure 2.	Full disk access for macOS	
Figure 3.	Select install mode	. 34
Figure 4.	Welcome panel	. 35
Figure 5.	License agreement	. 35
Figure 6.	Terms of use	. 36
Figure 7.	Default installation path	. 36
Figure 8.	Setup of shortcuts	. 37
Figure 9.	Package installation	. 37
Figure 10.	Installation script	. 38
Figure 11.	Installation path	. 38
Figure 12.	Current user shortcut creation	. 39
Figure 13.	Package installation	. 39
Figure 14.	Installation completed	. 40
Figure 15.	Example of installation in interactive mode.	
Figure 16.	STM32Cube installation wizard	
Figure 17.	Displaying Windows default proxy settings	
Figure 18.	Updater Settings window	
Figure 19.	Connection Parameters tab - Manual Configuration of Proxy Server	
Figure 20.	Connection failure	
Figure 21.	Embedded Software Packages Manager window	
Figure 22.	Managing embedded software packages - Help menu	
Figure 23.	Managing embedded software packages - Adding a new url	
Figure 24.	Checking the validity of vendor pack.pdsc file url	
Figure 25.	User-defined list of software packs	
Figure 26.	Selecting an embedded software pack release	
Figure 27.	License agreement acceptance	
Figure 28.	Embedded software pack release - Successful installation	
Figure 29.	Removing a package	
Figure 30.	Confirmation message	
Figure 31.	Checking for available updates.	
Figure 32.	Help menu: checking for updates	
Figure 33.	STM32CubeMX home page	
Figure 34.	Window menu	
Figure 35.	Output view	
Figure 36.	Link to social platforms	
Figure 37.	New Project window shortcuts	
Figure 38.	Enabling TrustZone	
Figure 39.	Adjusting selector results	
Figure 40.	New Project window - MCU selector.	
Figure 41.	Marking an MCU as favorite	
Figure 42.	New Project window - Board selector	70
Figure 43.	New project window - Example selector	
Figure 44.	Popup window - Starting a project from an example	
Figure 45.	Cross selector - Data refresh prerequisite	
Figure 46.	Cross selector - Part number selection per vendor.	
Figure 47.	Cross selector - Partial part number selection completion	
Figure 48.	Cross selector - Compare cart	



Figure 49. Figure 50. Figure 51. Figure 52.	Cross selector - Part number selection for a new project	76 77
	(peripherals initialized with default configuration)	
Figure 53.	Project choice interface	79
Figure 54.	Contexts	79
Figure 55.	IPs interface assignment	80
Figure 56.	TrustZone option	
Figure 57.	Selected context	81
Figure 58.	Assign IP context	
Figure 59.	OP-TEE selected	82
Figure 60.	U-Boot selection	
Figure 61.	FSBL synchronization output	83
Figure 62.	Contextual Help window (default)	
Figure 63.	Contextual Help detailed information	85
Figure 64.	Pinout view	
Figure 65.	Modifying pin assignments from the Pinout view	
Figure 66.	Example of remapping in case of block of pins consistency	91
Figure 67.	Pins/Signals Options window	
Figure 68.	Pinout view: MCUs with multi-bonding	
Figure 69.	Pinout view: multi-bonding with extended mode	94
Figure 70.	System view	
Figure 71.	Configuration window tabs (GPIO, DMA, and NVIC settings for STM32F4 series)	
Figure 72.	Peripheral mode and Configuration view	
Figure 73.	Formula when input parameter is set in No Check mode	98
Figure 74.	User Constants tab	
Figure 75.	Extract of the generated main.h file	
Figure 76.	Using constants for peripheral parameter settings 1	00
Figure 77.	Specifying user constant value and name 1	01
Figure 78.	Deleting an user constant is not allowed when it	
	is already used for another constant definition	
Figure 79.	Confirmation request to delete a constant for parameter configuration	
Figure 80.	Consequence when deleting a user constant for peripheral configuration 1	
Figure 81.	Searching for a name in a user constant list	
Figure 82.	Searching for a value in a user constant list	
Figure 83.	GPIO configuration window - GPIO selection	
Figure 84.	GPIO configuration grouped by peripheral 1	
Figure 85.	Multiple pins configuration	
Figure 86.	Adding a new DMA request 1	
Figure 87.	DMA configuration	
Figure 88.	DMA MemToMem configuration	
Figure 89.	NVIC configuration tab - FreeRTOS disabled	
Figure 90.	NVIC configuration tab - FreeRTOS enabled 1	
Figure 91.	I2C NVIC configuration window1	
Figure 92.	NVIC Code generation – All interrupts enabled 1	
Figure 93.	NVIC Code generation - IRQ Handler generation 1	
Figure 94.	FreeRTOS configuration view1	
Figure 95.	FreeRTOS: configuring tasks and queues1	
Figure 96.	FreeRTOS: creating a new task	
Figure 97.	FreeRTOS - Configuring timers, mutexes and semaphores	
Figure 98.	FreeRTOS heap usage1	18



	Selecting a HAL timebase source (STM32F407 example)	
	TIM1 selected as HAL timebase source	
	NVIC settings when using SysTick as HAL timebase, no FreeRTOS	
	NVIC settings when using FreeRTOS and SysTick as HAL timebase	
•	NVIC settings when using FreeRTOS and TIM2 as HAL timebase	
•	STM32MPUs boot devices and runtime contexts	
	STM32MPUs: assignment options for GPIOs	
•	Select peripherals as boot devices	
	Default configuration	
	Default configuration for the STM32MP2 series	
	RIF configuration extension in IPs panel for the STM32MP2 series	
•	RIF configuration extension in IPs panel for the STM32N6 series	
	RISUP configuration panel	
	Software context configuration vs. RISUP configuration	
	Example of IP assignment to one context and result in RISUP	
Figure 114.	Example of IP assignment to two contexts and result in RISUP	130
	Lock and privilege in RISUP table	
	Pseudo RIF-aware IP assignment	
	Peripherals (RISUP) panel for the STM32N6 series.	
Figure 118.	Creation of a new project for the STM32N6 series - Secure projects	132
Figure 119.	Peripherals (RISUP) panel for the STM32N6 series - Secure projects	133
Figure 120.	FMC configuration	134
Figure 121.	RIF FMC panel	134
Figure 122.	RTC features	135
Figure 123.	RTC mode	135
Figure 124.	RTC parameters setting	136
Figure 125.	Color coding system and instructions	137
Figure 126.	RIMU user interface	138
Figure 127.	Assigning a CID to an IP in RIMU.	139
Figure 128.	Modification of the security and privilege values	139
	IP assignment to a context	
Figure 130.	Result in RISUP of an IP assignment to a context	140
Figure 131.	Inheritance of CID, state of security, and privilege from RISUP	141
Figure 132.	Default values for IPs and user modification restrictions	141
Figure 133.	Domains (RIMU) panel for STM32N6 series	142
Figure 134.	RIF HSEM panel	143
Figure 135.	RIF TAMP panel (STM32MP2 devices)	144
	RIF TAMP panel (STM32N6 devices)	
	RIF-aware peripherals for STM32N6 MCUs	
	IO protection inheritance for a non-RIF-aware IP (I2C)	
	GPIO IP panel	
Figure 140.	Inheritance in RIF GPIO panel	147
	PIN reservation	
Figure 142.	HPDMA1 features with RIF implementation (STM32N6 MCUs)	148
Figure 143.	I2C IP panel	148
Figure 144.	I2C mode panel	149
Figure 145.	I2C features panel	149
Figure 146.	DMA RIF-aware IP inheritance	150
	RIF RCC panel (STM32MP2 MPUs)	
	RCC features with RIF implementation (STM32N6 MCUs)	
Figure 149.	RIF panel for EXTI1 (STM32N6 MCUs)	152
Figure 150.	RISAF configuration	153



-	Configuration of a new subregion	
•	Warning	
•	OCTOSPI1&2 configuration	
	OCTOSPI1&2 memory mapping	
	OCTOSPI1&2 region size configuration	
	OCTOSPI1&2 inheritances from RISUP	
	OCTOSPI1&2 Master CID activation example	
	DDR memory configuration	
	DDR_CTRL_PHY activation	
	Configuration of RISAF4 (DDR)	
-	PCIE memory configuration	
	Global lock in RISAF panel for STM32N6 MCUs	
	RISAF configuration for STM32N6 series.	
	Sub-regions activation in RISAF (showing activated subregions)	
	Sub-regions activation in RISAF (check the filtering parameter)	
	Example: RISUP configuration and generated code	
	Example: RISAF configuration and generated code.	
•	IAC feature	
•	STM32H7 dual-core: peripheral and middleware context assignment	
-	STM32H7 dual-core: GPIOs context assignment.	
	Pinout & Configuration view for TrustZone [®] -enabled projects $\dots \dots \dots \dots \dots \dots$	
	Setting privileges for peripherals	
	Setting privileges for GPIO EXTIS	
	Configuring security and privilege of DMA requests.	
	RCC privilege mode	
	Configuring security and privilege of DMA requests.	
	Securing peripherals from GTZC panel	
	OTFDEC secured when TrustZone [®] is active \dots	
	STM32F469NIHx clock tree configuration view	
	Clock tree configuration view with errors	
	Clock tree configuration: enabling RTC, RCC clock source	170
riguie 102.	and outputs from Pinout view	182
Figure 183	Clock tree configuration: RCC peripheral advanced parameters	
	Project Settings window	
-	Project folder.	
	Selecting a basic application structure	
•	Selecting an advanced application structure	
-	OpenSTLinux settings (STM32MPUs only)	
	Selecting a different firmware location	
	Firmware location selection error message	
	Recommended new firmware repository structure	
	Project Settings code generator	
	Template Settings window	
•	Generated project template	
•	Advanced Settings window.	
Figure 195.	Generated init functions without C language "static" keyword	106
	Automatic project import	
-	Manual project import	
	Import Project menu - Try Import with errors	
	Import Project menu - Successful import after adjustments	
	Set unused pins window	
i iguie 201.		201



Eiguro 202	Reset used pins window	201
	Set unused GPIO pins with Keep Current Signals Placement checked	
	Set unused GPIO pins with Keep Current Signals Placement unchecked	
	Additional software window	
-	Component dependency resolution	
	Details and Warnings panel	
Figure 208.	Selection of additional software components	211
Figure 209.	Additional software components - Updated tree view	211
Figure 210.	LPBAM window	212
Figure 211.	CAD Resources view	213
Figure 212.	CAD Resources not available	214
Figure 213.	CAD Resources selection for download	214
-	CAD Resources in Tools panel.	
	CAD Resources for STM32CubeMX project.	
	Boot path configuration ecosystem.	
	Boot paths for STM32H57x devices	
	Boot paths for STM32H56x devices	
•	Application boot paths (legacy and ST-iRoT projects)	
	Application boot path (OEM-iRoT)	
	Application boot path (OEM-uRoT assembled)	
	Application boot path: ST-iRoT and uRoT secure/nonsecure project.	219
Figure 223.	Application boot path:	000
E : 004	ST-iRoT and secure/nonsecure user application assembled	
-	Application boot path: ST-iRoT dual figure	220
Figure 225.	Application boot path:	
	(OEM-iRoT and secure/nonsecure user application assembled)	
	Select the device or board	
•	Select the STM32H5 device	
	Peripheral initialization	
Figure 229.	Boot paths for STM32H56x devices	223
Figure 230.		
	Activate TrustZone	
Figure 231.	Activate TrustZone	223
	Activate TrustZone Device and peripherals configuration	223 224
Figure 232.	Activate TrustZone Device and peripherals configuration Configuring the project	223 224 225
Figure 232. Figure 233.	Activate TrustZone Device and peripherals configuration Configuring the project Saving the project	223 224 225 225
Figure 232. Figure 233. Figure 234.	Activate TrustZone Device and peripherals configuration Configuring the project Saving the project Boot path selection	223 224 225 225 226
Figure 232. Figure 233. Figure 234. Figure 235.	Activate TrustZone Device and peripherals configuration Configuring the project Saving the project Boot path selection Select OEM-iRoT	223 224 225 225 226 226
Figure 232. Figure 233. Figure 234. Figure 235. Figure 236.	Activate TrustZone Device and peripherals configuration Configuring the project Saving the project Boot path selection Select OEM-iRoT First boot path stage	223 224 225 225 226 226 227
Figure 232. Figure 233. Figure 234. Figure 235. Figure 236. Figure 237.	Activate TrustZone Device and peripherals configuration Configuring the project Saving the project Boot path selection Select OEM-iRoT First boot path stage Select Secure Application	
Figure 232. Figure 233. Figure 234. Figure 235. Figure 236. Figure 237. Figure 238.	Activate TrustZone Device and peripherals configuration Configuring the project Saving the project Boot path selection Select OEM-iRoT First boot path stage Select Secure Application Last boot path stage	
Figure 232. Figure 233. Figure 234. Figure 235. Figure 236. Figure 237. Figure 238. Figure 239.	Activate TrustZone Device and peripherals configuration Configuring the project Saving the project Boot path selection Select OEM-iRoT First boot path stage Select Secure Application Last boot path stage Project provisioning	
Figure 232. Figure 233. Figure 234. Figure 235. Figure 236. Figure 237. Figure 238. Figure 239. Figure 240.	Activate TrustZone Device and peripherals configuration Configuring the project Saving the project Boot path selection Select OEM-iRoT First boot path stage Select Secure Application Last boot path stage Project provisioning Flash size not aligned	
Figure 232. Figure 233. Figure 234. Figure 235. Figure 236. Figure 237. Figure 238. Figure 239. Figure 240. Figure 241.	Activate TrustZone Device and peripherals configuration Configuring the project Saving the project Saving the project Select OEM-iRoT First boot path stage Select Secure Application Last boot path stage Project provisioning Flash size not aligned Boot path and debug authentication panel	
Figure 232. Figure 233. Figure 234. Figure 235. Figure 236. Figure 237. Figure 238. Figure 239. Figure 240. Figure 241. Figure 242.	Activate TrustZone Device and peripherals configuration Configuring the project Saving the project Boot path selection Select OEM-iRoT First boot path stage Select Secure Application Last boot path stage Project provisioning Flash size not aligned Boot path and debug authentication panel Authentication and encryption keys regeneration	
Figure 232. Figure 233. Figure 234. Figure 235. Figure 236. Figure 237. Figure 238. Figure 239. Figure 240. Figure 241. Figure 242. Figure 243.	Activate TrustZone Device and peripherals configuration Configuring the project Saving the project Boot path selection Select OEM-iRoT First boot path stage Select Secure Application Last boot path stage Project provisioning Flash size not aligned Boot path and debug authentication panel Authentication and encryption keys regeneration Secure image configuration	
Figure 232. Figure 233. Figure 234. Figure 235. Figure 236. Figure 237. Figure 238. Figure 239. Figure 240. Figure 241. Figure 242. Figure 243. Figure 244.	Activate TrustZone Device and peripherals configuration Configuring the project Saving the project Boot path selection Select OEM-iRoT First boot path stage Select Secure Application Last boot path stage Project provisioning Flash size not aligned Boot path and debug authentication panel Authentication and encryption keys regeneration Secure image configuration	
Figure 232. Figure 233. Figure 234. Figure 235. Figure 236. Figure 237. Figure 238. Figure 239. Figure 240. Figure 241. Figure 242. Figure 243. Figure 244. Figure 245.	Activate TrustZone Device and peripherals configuration Configuring the project Saving the project Boot path selection Select OEM-iRoT First boot path stage Select Secure Application Last boot path stage Project provisioning Flash size not aligned Boot path and debug authentication panel Authentication and encryption keys regeneration Secure image configuration Nonsecure image configuration Generate the code	
Figure 232. Figure 233. Figure 234. Figure 235. Figure 236. Figure 237. Figure 238. Figure 239. Figure 240. Figure 241. Figure 242. Figure 243. Figure 244. Figure 245. Figure 246.	Activate TrustZone Device and peripherals configuration Configuring the project Saving the project Saving the project Solution Boot path selection Select OEM-iRoT First boot path stage Select Secure Application Last boot path stage Project provisioning Flash size not aligned Boot path and debug authentication panel Authentication and encryption keys regeneration Secure image configuration Nonsecure image configuration Generate the code Code is generated Code is generated	
Figure 232. Figure 233. Figure 234. Figure 235. Figure 236. Figure 237. Figure 238. Figure 240. Figure 240. Figure 241. Figure 242. Figure 243. Figure 244. Figure 245. Figure 246. Figure 247.	Activate TrustZone Device and peripherals configuration Configuring the project Saving the project Saving the project Boot path selection Boot path selection Select OEM-iRoT First boot path stage Select Secure Application Last boot path stage Project provisioning Flash size not aligned Secure inage configuration Authentication and encryption keys regeneration Secure image configuration Nonsecure image configuration Generate the code Code is generated Secure and nonsecure IDE directories	
Figure 232. Figure 233. Figure 234. Figure 235. Figure 236. Figure 237. Figure 239. Figure 240. Figure 240. Figure 241. Figure 242. Figure 243. Figure 244. Figure 245. Figure 245. Figure 247. Figure 248.	Activate TrustZone	
Figure 232. Figure 233. Figure 234. Figure 235. Figure 236. Figure 237. Figure 238. Figure 240. Figure 240. Figure 241. Figure 242. Figure 243. Figure 244. Figure 245. Figure 246. Figure 247. Figure 248. Figure 249.	Activate TrustZone Device and peripherals configuration Configuring the project Saving the project Boot path selection Select OEM-iRoT First boot path stage Select Secure Application Last boot path stage Project provisioning Flash size not aligned Boot path and debug authentication panel Authentication and encryption keys regeneration Secure image configuration Nonsecure image configuration Generate the code Code is generated Secure and nonsecure IDE directories IDE post build commands . Trusted Package Creator output directory	
Figure 232. Figure 233. Figure 234. Figure 235. Figure 236. Figure 237. Figure 238. Figure 239. Figure 240. Figure 241. Figure 242. Figure 243. Figure 244. Figure 245. Figure 245. Figure 247. Figure 248. Figure 249. Figure 250.	Activate TrustZone	



	Error message	
	Select ST-iRoT	
•	Final boot path stage	
-	Boot path and Debug Authentication tab	
	Select the project structure	
	Code is generated	
	Secure project completed	
	IDE post build commands.	
	Board provisioning	
	On-screen instructions	
-	The flash_layout.h file	
•	The map.properties file	
	Secure generated project	
	Nonsecure generated project	
	Compilation project	
-	Project folder.	
•	Project creation.	
	Save the project	
	Boot path and debug authentication panel	
	First (left) and second (right) boot path stage	
	Final boot path stage	
	Boot path and debug authentication tab	
	map.properties file	
-	Code generation with EWARM	
	Nonsecure generated project	
-	Secure generated project	
	Boot path project.	
	Use default configuration	
	Configure the project	
-	Select the project	
-	First boot path stage	
	Final boot path stage	
	Boot path and debug authentication panel	
•	Generate the code	
	Application IDE directories	
	Home page without the login form	
-	Install or remove a software package	
	Missing myST information	
	Authentication from myST tab	
	User Authentication Dialog	
-	The myST display after login	
	Blocked login cases	
	Account creation form	
0	Enter the email address	
•	Password restoration	
•	Reset password form	
0	About window	
•	ST Tools	
	Reaching Compare Project from the Tools panel	
	Reaching Compare Project from the home page	
Figure 303.	User interface of the Compare Projects tool.	266



Figure 305. Figure 306.	Load the first .ioc file Starting the comparison Result of the comparison	267 268
	Loading the same project	
	The result of comparing two projects having the same structure	
	Compare the current non saved project with another project	
-	Compare a currently open project with itself.	
	Target table	
-	Peripherals & Middleware table	
	Project Settings table	
	Choosing the Excel format to save the comparison result	
-	Comparison result in Excel format	
	Comparison result in Excel format - Peripherals and middleware	
	Comparison result in Excel format - Project settings	
	Power Consumption Calculator default view	
	Battery selection	
	Step management functions	
	Power consumption sequence: New Step default view	279
Figure 322.	Enabling the transition checker option on an already	
	configured sequence - All transitions valid	280
Figure 323.	Enabling the transition checker option on an already	
	configured sequence - At least one transition invalid	280
Figure 324.	Transition checker option - Show log	280
Figure 325.	Interpolated power consumption.	282
Figure 326.	ADC selected in Pinout view	283
Figure 327.	Power Consumption Calculator configuration window:	
	ADC enabled using import pinout	284
Figure 328.	Power Consumption Calculator view after sequence building	285
Figure 329.	Sequence table management functions	285
Figure 330.	Power Consumption: Peripherals consumption chart	286
Figure 331.	Description of the Results area.	286
Figure 332.	Overall peripheral consumption	288
Figure 333.	Selecting SMPS for the current project.	290
	SMPS database - Adding new SMPS models	
	SMPS database - Selecting a different SMPS model	
	Current project configuration updated with new SMPS model	
	SMPS database management window with new model selected	
	SMPS transition checker and state diagram helper window	
	Configuring the SMPS mode for each step	
	RF related consumption (STM32WB series only)	
	RF Bluetooth Low-Energy mode configuration (STM32WB series only)	
•	ZigBee configuration (STM32WB series only)	
	RF sub-GHz configuration	
	Power Consumption Calculator – Example set	
	Power Consumption Calculator – Example sequence loading	
	Power Consumption Calculator – Example sequence new selection	
	DDR pinout and configuration settings	
-	DDR3 configuration	
	DDR Suite - Connection to target	
-	DDR Suite - Target connected	
-	DDR activity logs	
	DDR interactive logs	
i iguie 002.		505



	DDR register loading	
•	DDR test list from U-Boot SPL	
•	DDR test suite results	
	DDR tests history	
•	Regions settings to peripherals ON	
	Regions settings to linker files ON	
-	Regions settings to peripherals OFF	
	MMT usage	
	MMT view	
•	Start a project	
0	Use TrustZone	
•	Default settings	
Figure 365.	Region information	311
Figure 366.	Tooltip	311
Figure 367.	IP configuration	312
Figure 368.	IP under control	312
Figure 369.	Linker files update	313
Figure 370.	Configure an external memory	314
Figure 371.	New region created.	314
Figure 372.	Adding a new region	315
Figure 373.	Adding a new memory	315
	Memory assignment	
	Left panel configuration	
	Allocating a region	
	Middleware memory allocation	
	Middleware heap configuration	
	Remapping the memory	
	Remapped region is renamed	
	Remapped start address	
	New region remapped	
•	Resizing default region	
	Region security change	
	Memory map in linker file	
•	MMT usage (STM32U5)	
	MMT usage (STM32H7 single core).	
Figure 388	MMT view for U5 without TrustZone	323
	MMT view for H7 single core	
	Default data region	
•	FMC activation	
•	Default mapping	
-	Before the swap	
•	After the swap	
0	Before remapping	
•	After remapping	
•	ETH MMT regions.	
•	ETH configuration for STM32H723VETx MCU.	
	Defined memory regions under the linker file	
-		
•		
•	Firmware version	
	MMT configuration for STM32WB5x.	
	Cortex_M7 mode and configuration	
rigure 404.	Cortex_M4 mode and configuration	332



	Default settings	
Figure 406.	Choose an STM32H7 dual-core product	334
	Region 0 added	
Figure 408.	Activate Memory Management support	335
Figure 409.	Default setting for new application region	336
	Adding a new region	
Figure 411.	Configure NVIC1 and NVIC2, and select their HSEM global interrupt	337
Figure 412.	OPENAMP_M7 parameters settings	337
Figure 413.	OPENAMP_M4 parameters settings	338
Figure 414.	Reserved memory regions for OPENAMP using MMT.	338
Figure 415.	Linker files update (stm32h755xxx_flash_cm4.icf)	339
Figure 416.	Linker files update(stm32h755xxx_flash_cm7.icf)	339
	Configuration of ETH IP	
	ETH MMT regions.	
-	IP configuration	
	Defined memories under the linker file (Cortex-M7)	
	Defined memories under the linker file (Cortex-M4)	
	MMT usage	
	Default settings	
	Choose an STM32H7R product	
	Initialization dialogue	
-	Region0 added	
	Activate Memory Management support	
-	Warning message.	
	Configure the XSPI	
	EXT_MEM_MANAGER	
-	Tooltip	
	Linker files update.	
	Memory assignment for context Boot H7RS.	
	EXTMEM_MANAGER "Select boot code generation" disabled	
•	MMT Execute In Place	
•	Load and Run	
•	MMT Load and Run	
•	Linker files.	
	Flash option bytes.	
	ECC_ON_SRAM enabled and DTCM_AXI_SHARED set to 2	
	ETH MMT regions for STM32H7R3A8Ix.	
	ETH configuration for STM32H7R3A8Ix	
•	Application of the MMT configuration to the linker file	
	Defined memory regions under the linker file of the application context.	
•	MMT usage	
Figure 448.	User interface	359
Figure 449.	Linker files update	360
Figure 450.	Impact on RADIO (STM32WB09)	360
Figure 451.	MMT/boot path (STM32H7RS)	361
Figure 452.	MMT/boot path (STM32H5)	361
Figure 453.	Linker files location (STM32H7RS on the left, STM32H5 on the right).	362
	App_User declaration (STM32H7RS).	
	App_User declaration (STM32H5)	
	Labels for pins generating define statements	



E. 453		~~ 4
	User constant generating define statements	
	Duplicate labels	
	HAL-based peripheral initialization: usart.c code snippet	
-	LL-based peripheral initialization: usart.c code snippet	
	HAL versus LL: main.c code snippet	
Figure 462.	Default content of the extra_templates folder	371
Figure 463.	extra_templates folder with user templates	372
Figure 464.	Project root folder with corresponding custom generated files	373
Figure 465.	User custom folder for templates	373
Figure 466.	Custom folder with corresponding custom generated files	374
-	Update of the project .ewp file (EWARM IDE)	
U U	for preprocessor define statements	376
Figure 468.	Update of stm32f4xx_hal_conf.h file to enable selected modules	
	New groups and new files added to groups in EWARM IDE	
	Preprocessor define statements in EWARM IDE	
	Code generation for STM32H7 dual-core devices	
	Startup and linker files for STM32H7 dual-core devices	
	Building secure and nonsecure images with $ARMv8-M$ TrustZone [®]	
	Project explorer view for STM32L5 TrustZone [®] enabled projects	
	Project settings for STM32CubeIDE toolchain	
	STM32CubeMX generated DTS – Extract 1	
	STM32CubeMX generated DTS – Extract 1	
•	STM32CubeMX generated DTS – Extract 3	
	Project settings to configure Device tree path	
•	Selecting a CMSIS-Pack software component.	
	Enabling and configuring a CMSIS-Pack software component	
•	MCU selection	
	Pinout view with MCUs selection	
	Pinout view without MCUs selection window	
•	GPIO pin configuration	
	Timer configuration	
•	Simple pinout configuration	
	Save Project As window	
	Generate Project Report - New project creation.	
	Generate Project Report - Project successfully created	
	Clock tree view	
•	HSI clock enabled	
•	HSE clock source disabled	
	HSE clock source enabled	
	External PLL clock source enabled	
	Pinout & Configuration view	
•	Case of Peripheral and Middleware without configuration parameters.	
•	Timer 3 configuration window	
	Timer 3 configuration	
	Enabling Timer 3 interrupt	
	GPIO configuration color scheme and tooltip	
Figure 503.	GPIO mode configuration	404
	DMA parameters configuration window	
•	Middleware tooltip	
	USB Host configuration	
Figure 507.	FatFs over USB mode enabled	406



	System view with FatFs and USB enabled	
	Project Settings and toolchain selection	
	Project Manager menu - Code Generator tab	
	Missing firmware package warning message	
	Error during download	
	Updater settings for download	
	Updater settings with connection	
	Downloading the firmware package	
	Unzipping the firmware package	
	C code generation completion message	
	C code generation output folder	
	C code generation output: Projects folder	
	C code generation for EWARM	
	STM32CubeMX generated project open in IAR™ IDE	
	IAR™ options	
•	SWD connection	
•	Project building log	
	User Section 2	
•	User Section 4	
0	Import Project menu	
	Board peripheral initialization dialog box	
	Board selection	
	SDIO peripheral configuration	
-	FatFs mode configuration	
	RCC peripheral configuration	
	Clock tree view	
•	FATFS tutorial - Project settings.	
	C code generation completion message	
	IDE workspace	
	Power Consumption Calculation example	
	VDD and battery selection menu	
	Sequence table.	
	sequence results before optimization	
	Step 1 optimization	
	Step 5 optimization	
•	Step 6 optimization	
•	Step 7 optimization	
-	Step 8 optimization	
•	Step 10 optimization	
	Power sequence results after optimizations	
	Selecting NUCLEO L053R8 board	
•	Selecting debug pins	
	Selecting TIM2 clock source.	
	Selecting asynchronous mode for USART2	
	Checking pin assignment	
	Configuring the MCU clock tree	
	Configuring USART2 parameters	
-	Configuring TIM2 parameters	
-	Enabling TIM2 interrupt	
	Project Settings menu.	
	Generating the code	



Figure 560.	Checking the communication port	449
	Setting Tera Term port parameters	
	Setting Tera Term port parameters	
	Existing or new project pinout.	
Figure 564.	List of pinout compatible MCUs - Partial match	
-	with hardware compatibility.	452
Figure 565.	List of pinout compatible MCUs - Exact and partial match	452
Figure 566.	Selecting a compatible MCU and importing the configuration	453
Figure 567.	Configuration imported to the selected compatible MCU	453
Figure 568.	Additional software components enabled for the current project	455
Figure 569.	Pack software components: no configurable parameters	456
Figure 570.	Pack tutorial: project settings	456
Figure 571.	Generated project with third party pack components	457
Figure 572.	Hardware prerequisites	458
Figure 573.	Embedded software packages	459
Figure 574.	Mobile application	459
Figure 575.	Installing Embedded software packages	460
Figure 576.	Starting a new project - selecting the NUCLEO-L053R8 board	461
Figure 577.	Starting a new project - initializing all peripherals	461
	Selecting X-Cube-BLE1 components	
Figure 579.	Configuring peripherals and GPIOs	463
Figure 580.	Configuring NVIC interrupts	464
Figure 581.	Enabling X-Cube-BLE1.	465
Figure 582.	Configuring the SensorDemo project	466
Figure 583.	Open SensorDemo project in the IDE toolchain	466
Figure 584.	LPBAM project	468
Figure 585.	Project timeline	469
Figure 586.	Project with LPBAM capability	469
Figure 587.	LPBAM Scenario & Configuration view	470
Figure 588.	Adding an application	471
	SoC and IPs configuration	
Figure 590.	LPBAM scenario: creation and configuration panels	472
Figure 591.	Clock tree configuration	473
	Available IPs	
Figure 593.	IP configuration: advanced settings	474
Figure 594.	LPBAM low power settings	475
	Adding nodes to a queue	
	Queue in circular mode	
	Queue looping back on IP data transfer	
	LPBAM queue: DMA configuration	
	LPBAM functions contextual help	
	LPBAM queue node configuration	
Figure 601.	LPBAM node: configuring hardware resources	481
	LPBAM node trigger configuration	
	LPBAM node triggered using timer channel	
	LPBAM node: reconfiguring a DMA	
	Reconfiguring DMA for data transfer when destination is memory	
	Design check	
-	STM32CubeMX project generated with LPBAM applications	
	STM32CubeMX project - Peripheral secure context assignment	
	STM32CubeMX project - Clock source secure context assignment	
Figure 610.	LPBAM project - Peripheral no context assignment	488



	LPBAM application - Clock source no context assignment
	LPBAM application - Secure context assignment
Figure 613.	LPBAM design security coherency check
	Block mapping
Figure 615.	Block remapping
Figure 616.	Block remapping - Example 1
Figure 617.	Block remapping - Example 2
Figure 618.	Block inter-dependency - SPI signals assigned to PB3/4/5
Figure 619.	Block inter-dependency - SPI1_MISO function assigned to PA6
Figure 620.	One block = one peripheral mode - I2C1_SMBA function assigned to PB5
Figure 621.	Block remapping - Example 2
Figure 622.	Function remapping example
Figure 623.	Block shifting not applied
Figure 624.	Block shifting applied
Figure 625.	FreeRTOS HOOK functions to be completed by user
Figure 626.	LwIP 1.4.1 configuration
Figure 627.	LwIP 1.5 configuration
Figure 628.	Libjpeg configuration window
	Mbed TLS without LwIP
Figure 630.	Mbed TLS with LwIP and FreeRTOS
	Mbed TLS configuration window
Figure 632.	Enabling the TouchSensing peripheral
	Touch-sensing sensor selection panel
Figure 634.	TouchSensing configuration panel
Figure 635.	BLE and Thread middleware support in STM32CubeMX514
Figure 636.	STM32CubeWB Package download
Figure 637.	STM32CubeWB BLE applications folder
Figure 638.	BLE Server profile selection
Figure 639.	BLE Client profile selection
Figure 640.	Thread application selection
Figure 641.	Enabling OpenAmp for STM32MPUs
Figure 642.	Enabling the Resource Manager for STM32MPUs
	Resource Manager: peripheral assignment view
	STM32 microcontroller part numbering scheme
Figure 645.	STM32Cube Embedded Software package



1 STM32Cube overview

STM32Cube is an STMicroelectronics original initiative to improve designer productivity significantly by reducing development effort, time, and cost. STM32Cube covers the whole portfolio of STM32 devices, based on 32-bit Arm^{®(a)} Cortex[®] cores.

STM32Cube includes:

- A set of user-friendly software development tools to cover project development from conception to realization, among which are:
 - STM32CubeMX, a graphical software configuration tool that allows the automatic generation of C initialization code using graphical wizards
 - STM32CubeIDE, an all-in-one development tool with peripheral configuration, code generation, code compilation, and debug features
 - STM32CubeCLT, an all-in-one command-line development toolset with code compilation, board programming, and debug features
 - STM32CubeProgrammer (STM32CubeProg), a programming tool available in graphical and command-line versions
 - STM32CubeMonitor (STM32CubeMonitor, STM32CubeMonPwr, STM32CubeMonRF, STM32CubeMonUCPD), powerful monitoring tools to fine-tune the behavior and performance of STM32 applications in real time
- STM32Cube MCU and MPU Packages, comprehensive embedded-software platforms specific to each microcontroller and microprocessor series (such as STM32CubeH5 for the STM32H5 series), which include:
 - STM32Cube hardware abstraction layer (HAL), ensuring maximized portability across the STM32 portfolio
 - STM32Cube low-layer APIs, ensuring the best performance and footprints with a high degree of user control over hardware
 - A consistent set of middleware components, such as ThreadX, FileX / LevelX, NetX Duo, USBX, USB-PD, mbed-crypto, secure manager API, MCUboot, and OpenBL
 - All embedded software utilities with full sets of peripheral and applicative examples
- STM32Cube Expansion Packages, which contain embedded software components that complement the functionalities of the STM32Cube MCU and MPU Packages with:
 - Middleware extensions and applicative layers
 - Examples running on some specific STMicroelectronics development boards



a. Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



2 Getting started with STM32CubeMX

2.1 Principles

Customers need to quickly identify the MCU that best meets their requirements (core architecture, features, memory size, performance...). While board designers main concerns are to optimize the microcontroller pin configuration for their board layout and to fulfill the application requirements (choice of peripherals operating modes), embedded system developers are more interested in developing new applications for a specific target device, and migrating existing designs to different microcontrollers.

The time taken to migrate to new platforms and update the C code to new firmware drivers adds unnecessary delays to the project. STM32CubeMX was developed within STM32Cube initiative which purpose is to meet customer key requirements to maximize software reuse and minimize the time to create the target system:

- Software reuse and application design portability are achieved through STM32Cube firmware solution proposing a common Hardware Abstraction Layer API across STM32 portfolio.
- Optimized migration time is achieved thanks to STM32CubeMX built-in knowledge of STM32 microcontrollers, peripherals and middleware (LwIP and USB communication protocol stacks, FatFs file system for small embedded systems, FreeRTOS).

STM32CubeMX graphical interface performs the following functions:

- Fast and easy configuration of the MCU pins, clock tree and operating modes for the selected peripherals and middleware
- Generation of pin configuration report for board designers
- Generation of a complete project with all the necessary libraries and initialization C code to set up the device in the user defined operating mode. The project can be directly open in the selected application development environment (for a selection of supported IDEs) to proceed with application development (see *Figure 1*).

During the configuration process, STM32CubeMX detects conflicts and invalid settings and highlights them through meaningful icons and useful tool tips.



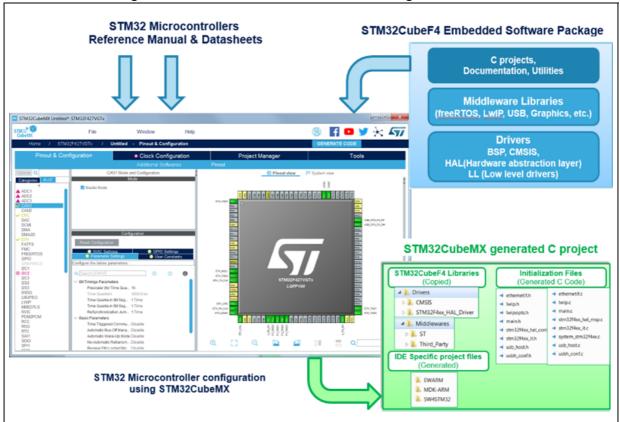


Figure 1. Overview of STM32CubeMX C code generation flow



2.2 Key features

STM32CubeMX comes with the following features:

Project management

- STM32CubeMX allows the user to create, save, and load previously saved projects:
- When STM32CubeMX is launched, the user can choose to create a new project or to load a previously saved project.
- Saving the project saves user settings and configuration performed within the project in an .ioc file to be used when the project will be loaded in STM32CubeMX again.

STM32CubeMX also allows the user to import previously saved projects in new ones. STM32CubeMX projects come in two flavors:

- MCU configuration only: .ioc file is saved in a dedicated project folder.
- MCU configuration with C code generation: in this case .ioc files are saved in a dedicated project folder along with the generated source C code. There can be only one .ioc file per project.
- Easy project creation starting from an MCU, a board, or an example

The new project window allows the user to create a project by selecting a microcontroller, a board, or an example project from STMicroelectronics STM32 portfolio. Different filtering options are available to ease the MCU and board selection. There is also the possibility to select an MCU through the Cross selector tab by comparing characteristics to those of competitors. Comparison criteria can be adjusted.

- Easy pinout configuration
 - From the **Pinout** view, the user can select the peripherals from a list and configure the peripheral modes required for the application. STM32CubeMX assigns and configures the pins accordingly.
 - For more advanced users, it is also possible to directly map a peripheral function to a physical pin using the **Pinout** view. The signals can be locked on pins to prevent STM32CubeMX conflict solver from moving the signal to another pin.
 - Pinout configuration can be exported as a .csv file.

• Complete project generation

The project generation includes pinout, firmware and middleware initialization C code for a set of IDEs. It is based on STM32Cube embedded software libraries. The following actions can be performed:

- Starting from the previously defined pinout, the user can proceed with the configuration of middleware, clock tree, services (such as RNG, CRC) and peripheral parameters. STM32CubeMX generates the corresponding initialization C code. The result is a project directory including generated main.c file and C header files for configuration and initialization, plus a copy of the necessary HAL and middleware libraries as well as specific files for the selected IDE.
- The user can modify the generated source files by adding user-defined C code in user dedicated sections. STM32CubeMX ensures that the user C code is preserved upon next C code generation (the user C code is commented if no longer relevant for the current configuration).
- STM32CubeMX can generate user files by using user-defined freemarker .ftl template files.



From the Project Settings menu, the user can select the development toolchain (IDE) for which the C code has to be generated. STM32CubeMX ensures that the IDE relevant project files are added to the project folder so that the project can be directly imported as a new project within STM32Cube or third party IDEs (IAR[™] EWARM, Keil[™] MDK-Arm, KITWARE[™] CMake, FSF[™] Makefile).

Power consumption calculation

Starting with the selection of a microcontroller part number and a battery type, the user can define a sequence of steps representing the application life cycle and parameters (choice of frequencies, enabled peripherals, step duration). STM32CubeMX Power Consumption Calculator returns the corresponding power consumption and battery life estimates.

Clock tree configuration

STM32CubeMX offers a graphic representation of the clock tree as it can be found in the device reference manual. The user can change the default settings (clock sources, prescaler and frequency values). The clock tree is then updated accordingly. Invalid settings and limitations are highlighted and documented with tool tips. Clock tree configuration conflicts can be solved by using the solver feature. When no exact match is found for a given user configuration, STM32CubeMX proposes the closest solution.

• Automatic updates of STM32CubeMX and STM32Cube MCU packages

STM32CubeMX comes with an updater mechanism that can be configured for automatic or on-demand check for updates. It supports self-updates as well as firmware library package updates. The updater mechanism also allows deleting previously installed packages.

Report generation

.pdf and .csv reports can be generated to document the user configuration work.

Support of embedded software packages in CMSIS-Pack format (Software Packs)

STM32CubeMX allows getting and downloading updates of embedded software packages delivered in CMSIS-Pack format. Selected software components belonging to these new releases can then be added to the current project.

• Generating Software Packs with STM32PackCreator

STM32PackCreator is a graphical tool installed with STM32CubeMX in the Utilities folder. It allows the user to create Software Packs and STM32Cube Expansion packages enhanced for STM32CubeMX. It can be launched from the ST Tools tab found in the Tools view.

Contextual help

Contextual help windows can be displayed by hovering the mouse over Cores, Series, Peripherals and Middleware. They provide a short description and links to the relevant documentation corresponding to the selected item.

• Access to ST tools

From STM32CubeMX project, the Tools tab allows the user to launch Tools directly or to access tools download pages on *www.st.com*.

• Video tutorials

STM32CubeMX allows the user to browse and play video tutorials. The video tutorial browser is accessible from the Help menu.



2.3 Rules and limitations

- C code generation covers only peripheral and middleware initialization. It is based on STM32Cube HAL firmware libraries.
- STM32CubeMX C code generation covers only initialization code for peripherals and middleware components that use the drivers included in STM32Cube embedded software packages. The code generation of some peripherals and middleware components is not yet supported.
- Refer to *Appendix A* for a description of pin assignment rules.
- Refer to *Appendix B* for a description of STM32CubeMX C code generation design choices and limitations.



3 Installing and running STM32CubeMX

3.1 System requirements

3.1.1 Supported operating systems and architectures

- Windows[®] 10 32 bits (x86) or 64 bits (x64), and Windows[®] 11 64 bits (x64)
- Linux[®]: Ubuntu[®] LTS 22.04, and LTS 24.04, and Fedora[®] 41
- macOS[®] 14 (Sonoma), macOS[®] 15 (Sequoia)

Note: Windows is a trademark of the Microsoft group of companies. Linux[®] is a registered trademark of Linus Torvalds. Ubuntu[®] is a registered trademark of Canonical Ltd. Fedora[®] is a trademark of Red Hat, Inc. macOS[®] is a trademark of Apple Inc., registered in the U.S. and other countries and regions.

For macOS the full disk access is required to load project files or install other packages from the file system. To enable full disk access for STM32CubeMX:

- 1. Go to "System preferences" and click to open "Security & Privacy" window (Figure 2)
- 2. Select "Privacy" tab
- 3. Select "Full Disk Access" from the left panel
- 4. Click the checkbox to enable full disk access to STM32CubeMX



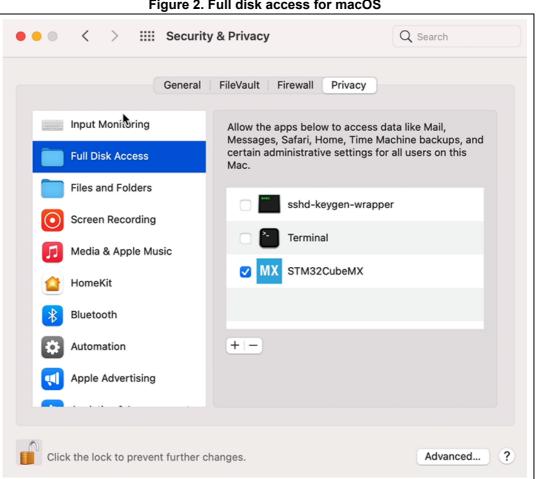


Figure 2. Full disk access for macOS

3.1.2 Memory prerequisites

Recommended minimum RAM: 2 Gbytes

3.1.3 Software requirements

If the initial installation was completed with administrator privileges, the user also needs these privileges to download and install the latest update package. Additionally, the user needs administrator rights to successfully apply the update at the next start of STM32CubeMX.

Java[™] Runtime Environment

For STM32CubeMX 6.13 the bundled JRE is openJDK Runtime Environment Temurin™ 21.0.3+9 (build 21.0.3+9-LTS) and JavaFX-21.0.3.

Starting with version V6.2.0, STM32CubeMX embeds the Java Runtime Environment (JRE^{TM(a)}) required for its execution and no longer uses the one installed on the user machine.

- For STM32CubeMX 6.3 the bundled JRE is AdoptOpenJDK-11.0.10+9 and JavaFX-11.0.2
- For STM32CubeMX 6.2 the bundled JRE is Liberica 1.8.0 265 of BellSoft



Versions earlier than STM32CubeMX V6.2.0 require to install a JRE, whose constraints are:

- 64-bit version mandatory, 32-bit version not supported
- the STM32PackCreator companion tool requires JRE supporting JavaFX
- minimum JRE version is 1.8_45 (known limitation with 1.8_251)
- version 11 is supported, versions 7, 9, 10, 12 and upper are not supported
- STMicroelectronics promotes the use of the following JREs:
- Oracle^(a), subject to license fee
- Amazon Corretto^{™(a)}, no-cost solution based on OpenJDK, JDK installer recommended.

STM32CubeMX operation is not guaranteed with other JREs.

macOS software requirements

- Xcode must be installed on macOS computers
- Both Xcode and Rosetta must be installed on macOS computers embedding Apple[®] M1 processor.

3.2 Installing/uninstalling STM32CubeMX standalone version

3.2.1 Installing STM32CubeMX standalone version

To install STM32CubeMX:

- 1. From an Internet browser, open the page www.st.com/stm32cubemx
- 2. Click "Get Software" to go to the software download section

On Windows

- a) On STM32CubeMX-Win line, click "Get software" to download the package
- b) Extract (unzip) the downloaded package
- c) Double-click on SetupSTM32CubeMX-VERSION-Win.exe to launch the installation wizard
- d) The installation wizard is displayed (see *Figure 3*), it gives the choice between two modes, namely "Install for all users", and "Install for me only (recommended)"

a. All other trademarks are the properties of their respective owners.



a. Oracle and Java are registered trademarks of Oracle and/or its affiliates.

Select Set	up Install Mode	>
MX	Select install mode STM32CubeMX can be installed for you only, or for all users (administration privileges required).	
	\rightarrow Install for me only (recommended)	
	Install for all users	
		Cancel

Figure 3. Select install mode

If you choose "Install for all users" mode:

- > Enter administrator credentials
- > Welcome panel (*Figure 4*)
- > License agreement (*Figure 5*)
- > Terms of use (*Figure 6*)
- The default installation path is set to C:\Program Files\STMicroelectronics\STM32Cube\STM32CubeMX (*Figure 7*)
- > The shortcuts for all users are created by default (*Figure 8*)
- > Package installation (*Figure 9*)
- > Installation script (*Figure 10*)

If you choose "Install for me only (recommended)" mode:

- > Welcome panel (*Figure 4*)
- > License agreement (*Figure 5*)
- > Terms of use (*Figure 6*)
- > The installation path is set on the home director by default (*Figure 11*): note that the default installation folder is, by default, a system hidden folder
- > The shortcut can be created only for the current user (*Figure 12*)
- > Package installation (*Figure 13*)
- > Installation script (*Figure 14*)

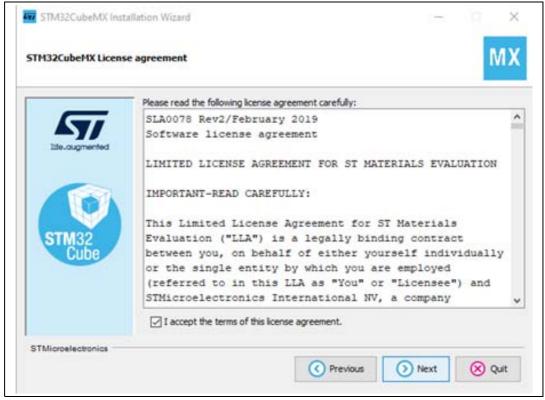




Figure 4. Welcom

STM32CubeMX Insta				
Welcome to the Instal	lation of STM32CubeMX 6.11.0			M)
Ē	Starting STM32CubeMX 6.11.0 installation			
life.augmented	The homepage is at: https://www.st.com/stm32cube			
STM32				
STM32 Cube				
STMicroelectronics				_
	0	Next	8	Quit

Figure 5. License agreement

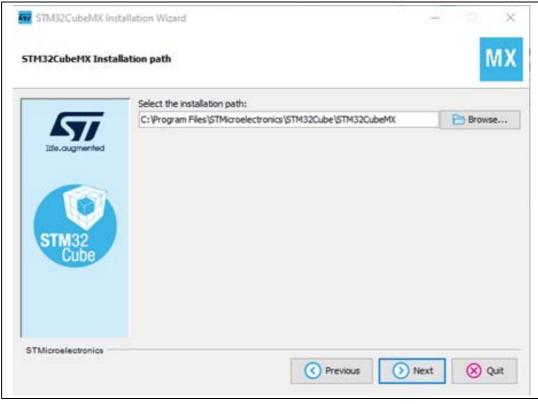




STM32CubeMX In:		
ST Privacy and Term	is of Use	MX
Life.ougmented	☑ I have read and understood the ST Terms of U	lse.
STM32 Cube	ST int N.V (as data controller) collects and uses statistics (directly or by ST affiliates) when you or purpose of continuously improving the application You can stop the collection of your features usa the application at any time with effect for the futu or update your preferences via the menu Help > User Preferences > General Settings	use the application for the n. ge statistics when you use
STMicroelectronics		
	Previous	Next 🚫 Quit

Figure 6. Terms of use

Figure 7. Default installation path





Figuro	o	Satur	of	chartcute
гідиге	о.	Secup	U	shortcuts

E	Create shortcuts in the Start-Menu	
57/	Create additional shortcuts on the desktop	
life.ougmented	Select a Program Group for the Shortcuts:	
	(Default)	create shortcut
1000	Accessibility	() current user
ABA	Accessories	(i) all users
	Administrative Tools CCleaner	101
STM32	Osco	
Cube	JetBrans	
COBC	Launch-4j	
	Microsoft Endpoint Manager v	
	STMicroelectronics/STM32Cube/STM32CubeMX	Default
	STMicroelectronics/STM32Cube/STM32CubeMX	Default

Figure 9. Package installation

🕶 STM32CubeMX Insta	llation Wizard — 🗆 🗙
STM32CubeMX Packag	e installation MX
	Pack installation progress: C:\Apps\STM32CubeMX6.11.0\db\mcu\config\lConfig\TIM-STM32MP1xx_DefMapping.xml
	Core
life.augmented	Overall installation progress: 1 / 14
STM32 Cube	
STMicroelectronics	
	Previous Next Quit



STM32CubeMX Insta	llation Wizard —			\times
STM32CubeMX Install	ation done		Ν	ЛХ
	Installation has completed successfully.			
	An uninstaller program has been created in:			
life.augmented	C:\Apps\STM32CubeMX6.11.0\Uninstaller			
STM32 Cube	Generate an automatic installation script			
STMicroelectronics		0) Do	ne

Figure 10. Installation script

Figure 11. Installation path

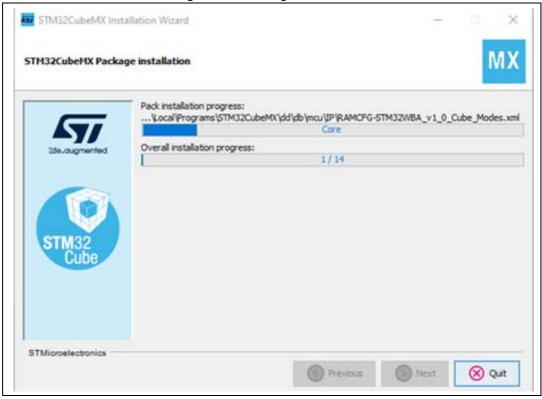
STM32CubeMX Instal	lation Wizard	- 🗆 X
STM32CubeMX Installa	tion path	MX
	Select the installation path:	
life.augmented	C:\Apps\STM32CubeMX6.11.0	Browse
STM32 Cube		
STMicroelectronics	Previous	Next Quit



M32CubeMX Shorte	outs setup	M.
Life-augmented	Create shortcuts in the Start-Menu Create additional shortcuts on the desktop Select a Program Group for the Shortcuts:	
STM32 Cube	(Default) Accessibility Accessories Administrative Tools Git Maintenance Startup System Tools Windows PowerShell	create shortout () current user) all users
	STMicroelectronics\STM32Cube\STM32CubeMX	Default

Figure 12. Current user shortcut creation

Figure 13. Package installation





TM32CubeMX Insta	llation done M
_	Installation has completed successfully.
	An uninstaller program has been created in:
life.ougmented	C: \Users\marroukh\AppData\Local\Programs\STM32CubeMX\dd\Uninstaller
STM32 Cube	
	Generate an automatic installation script

Figure 14. Installation completed

Note:

Upon successful installation, the STM32CubeMX icon is displayed on the desktop and the application is available from the Program menu. STM32CubeMX .ioc files are displayed with a cube icon, double-clicking on it opens the project in STM32CubeMX. Only the latest installation of STM32CubeMX is enabled in the Program menu. Previous versions can be kept on your PC (not recommended) when different installation folders have been specified. Otherwise, the new installation overwrites the previous one(s).

On Linux:

- a) On STM32CubeMX-Lin line, Click "Get software" to download the package
- b) Extract (unzip) the downloaded package
- c) Make sure you have administrator rights to access the target installation directory. You can run the installation as root (or sudo) to install STM32CubeMX in shared directories.
- d) Do **chmod 777 SetupSTM32CubeMX-VERSION** to change the properties, so that the file is executable
- e) Double-click on the **SetupSTM32CubeMX-VERSION** file, or launch it from the console window

On macOS:

- a) On **STM32CubeMX-Mac** line, Click "Get software" to download the package
- b) Extract (unzip) the downloaded package
- c) Make sure you have administrator rights
- d) Double-click **SetupSTM32CubeMX-VERSION.app** application file to launch the installation wizard



In case of error, try to fix it: - \$sudo xattr -cr <Folder where the zip was extracted>

3.2.2 Installing STM32CubeMX from command line

There are two ways to launch an installation from a console window: either in console interactive mode or via a script.

Interactive mode

To perform interactive installation, proceed as follows:

- 1. Extract (unzip) to folder the auto-extract installation file (SetupSTM32CubeMX-VERSION-Win.exe)
- 2. Open a standard console window to install for the current user, or the console window with administrator rights to install for all users
- 3. Go to the extracted folder (cd <folder path>)
- 4. Run the command jre\bin\java -jar SetupSTM32CubeMX-<VERSION>.exe console

At each installation step, an answer is requested (see Figure 15).

Figure 15. Example of installation in interactive mode

Administrator: C:\Windows\system32\cmd.exe
Press 1 to accept, 2 to reject, 3 to redisplay
Select target path [C:\Program Files\STMicroelectronics\STM32Cube\STM32CubeMX] C:\Program Files\MX set uninstallName=STM32CubeMX(3)
Press 1 to continue, 2 to quit, 3 to redisplay
Create shortcuts in the Start-Menu Enter Y for Yes, N for No:
Create additional shortcuts on the desktop Enter Y for Yes, N for No:
create shortcut for: all users Enter Y for Yes, N for No:
<pre>[Starting to unpack] [Processing package: Core (1/3)] [Processing package: Old DataBases (2/3)] [Processing package: Help (3/3)] [Unpacking finished]</pre>
Generate an automatic installation script Enter Y for Yes, N for No:
n Installation was successful application installed on C:\Program Files\MX [Writing the uninstaller data] [Console installation done]
C:\Users\>



During the installation, ignore the warnings.



Auto-install mode

At end of an installation, performed either using STM32CubeMX graphical wizard or console mode, it is possible to generate an auto-installation script containing user preferences (see *Figure 16*).

STM32CubeMX Ir	stallation Wizard	
STM32CubeMX Inst	Ilation done	MX
	Installation has completed successfully.	
life.augmented	An uninstaller program has been created in: C:\Program Files\STMicroelectronics\STM32Cube\STM32C	
STM32 Cube		
	Generate an automatic installation s	ript
STMicroelectronics		🔗 Done

Figure 16. STM32Cube installation wizard

You can then launch the installation by typing, from a console window (with or without administrator rights, according to your needs), the command:

SetupSTM32CubeMX-VERSION-Win.exe ABSOLUTE_PATH_TO_AUTO_INSTALL.xml

3.2.3 Uninstalling STM32CubeMX standalone version

Uninstalling STM32CubeMX on macOS[®]

- Move STM32CubeMX.VERSION.app to the trash
- Use the following command line:
 - For STM32CubeMX 6.2.x and later versions:
 cd SetupSTM32CubeMX-VERSION.app/Contents/Resources/Uninstaller
 ./uninstall.sh
 - For STM32CubeMX 6.1.x and older versions:
 java -jar SetupSTM32CubeMX VERSION.app/Contents/Resources/Uninstaller/uninstaller.jar.

Uninstalling STM32CubeMX on Linux

- From a shell prompt by launching the uninstall script
 - For STM32CubeMX 6.2.x and later versions:
 - cd <STM32CubeMX installation path>/Uninstaller uninstall.sh
 - For STM32CubeMX 6.1.x and older versions:
 java -jar <STM32CubeMX installation path>/Uninstaller/uninstaller.jar.
- From a file explorer
 - Go to <STM32CubeMX installation path>/Uninstaller
 - For STM32CubeMX 6.2.x and later versions: double-click the uninstall.sh script
 - For STM32CubeMX 6.1.x and older versions: double-click the start uninstall desktop shortcut

Uninstalling STM32CubeMX on Windows

- Through the Windows Control Panel:
 - a) Select **Programs and Features** from the **Windows Control** Panel to display the list of programs installed on your computer.
 - b) Right-click STM32CubeMX and select uninstall.
- From a shell prompt, by using the following commands:
 - For STM32CubeMX 6.10.x and later versions:
 with administrator rights:
 cd <STM32CubeMX installation path>/Uninstaller
 admin_uninstall.bat
 without administrator rights:
 cd <STM32CubeMX installation path>/Uninstaller
 uninstall.batcd <STM32CubeMX installation path>/Uninstaller
 uninstall.batcd <STM32CubeMX installation path>/Uninstaller
 From STM32CubeMX 6.2.x to STM32CubeMX 6.9.x versions:
 - cd <STM32CubeMX installation path>/Uninstaller admin_uninstall.bat
 - For STM32CubeMX 6.1.x and older versions:

java -jar <STM32CubeMX installation path>/Uninstaller/uninstaller.jar

- Through a Windows File Explorer window:
 - a) For STM32CubeMX 6.2.x and later versions:
 - Go to the Uninstaller folder in STM32CubeMX installation directory, then:
 - > with administrator rights, right-click on admin_uninstall.bat and "run as administrator"
 - > without administrator rights, click on uninstall.bat
 - b) For STM32CubeMX 6.1.x and older versions:

Go to the Uninstaller folder in STM32CubeMX installation directory Double-click startuninstall.exe, or double-click the uninstall shortcut on the desktop



3.3 Launching STM32CubeMX

When running STM32CubeMX behind a proxy, see Section 3.4.1.

3.3.1 Running STM32CubeMX as a standalone application

To run STM32CubeMX as a standalone application on Windows, select STM32CubeMX from Program Files > ST Microelectronics > STM32CubeMX,or double-click STM32CubeMX icon on your desktop.

To run STM32CubeMX as a standalone application on Linux, launch the STM32CubeMX executable from STM32CubeMX installation directory.

To run STM32CubeMX as a standalone application on macOS, launch the STM32CubeMX application from the launchpad.

Note: There is no STM32CubeMX desktop icon on macOS.

3.3.2 Running STM32CubeMX in command-line mode

To facilitate its integration with other tools, STM32CubeMX provides command-line modes. Thanks to the commands listed in *Table 1* it is possible to:

- load an MCU
- load an existing configuration
- save a current configuration
- set project parameters and generate corresponding code
- generate user code from templates
- load a board identified through its part number
- refresh the list of embedded software packages (packs and STM32Cube MCU packages) and install/remove a package
- select additional software (packs) components to add to the project.

Three command-line modes are available:

- To run STM32CubeMX in interactive command-line mode, use the following command lines:
 - On Windows:
 - cd <STM32CubeMX installation path>

```
jre\bin\java -jar STM32CubeMX.exe -i
```

– On Linux:

```
cd <STM32CubeMX installation path>
```

- ./STM32CubeMX -i
- On macOS:

cd <STM32CubeMX installation path> cd Contents/MacOs

```
./STM32CubeMX -i
```

The "MX>" prompt is displayed, to indicate that the application is ready to accept commands.

- To run STM32CubeMX in command-line mode, getting commands from a script, use the following command lines:
 - On Windows:

UM1718 Rev 47



UM1718

```
cd <STM32CubeMX installation path>
```

```
jre\bin\java -jar STM32CubeMX.exe -s <script filename>
```

```
- On Linux and macOS:
```

```
./STM32CubeMX -s <script filename>
```

All the commands to be executed must be listed in the script file. An example of script file content is shown below:

```
load STM32F417VETx
project name MyFirstMXGeneratedProject
project toolchain "MDK-ARM v4"
project path C:\STM32CubeProjects\STM32F417VETx
project generate
exit
```

- To run STM32CubeMX in command-line mode getting commands from a script and without UI, use the following command lines:
 - On Windows:

```
cd <STM32CubeMX installation path>
```

```
jre\bin\java -jar STM32CubeMX.exe -q <script filename>
```

- On Linux and macOS:
 - ./STM32CubeMX -q <script filename>

Here again, the user can enter commands when the MX prompt is displayed.

Command line	Purpose	Example
help	Displays the list of available commands.	help
swmgr refresh	Refreshes the list of embedded software package versions available for download.	swmgr refresh
swmgr install stm32cube_ <series> _<version> ask</version></series>	Installs the specified STM32Cube MCU package version. ⁽¹⁾	swmgr install stm32cube_f1_1.8.0 ask
swmgr remove stm32cube_ <series> _<version></version></series>	Removes the specified STM32Cube MCU package version.	swmgr remove stm32cube_f1_1.8.0
swmgr install <packvendor>.<packname>. <packversion> ask</packversion></packname></packvendor>	Installs the specified pack version.	swmgr install STMicroelectronics. X-CUBE-NFC4.1.4.1 ask
swmgr remove <packvendor>.<packname>. <packversion></packversion></packname></packvendor>	Removes the specified pack version.	swmgr remove STMicroelectronics. X-CUBE-BLE1.4.2.0

Table 1. Command line summary



Table 1. Command line summary (continued)			
Command line	Purpose	Example	
pack enable <vendor> <pack>[/bundle] <version> <class> <group>[/subgroup] [variant]</group></class></version></pack></vendor>	Selects a software pack component to add in the project. The presence of "/" in the second and/or the fifth parameter(s) indicates, respectively, the explicit mention of a bundle and/or a subgroup (reference: Arm CMSIS pack pdsc format). To find out the pack / bundle / class / group / subgroup names of the component to enable, select the component and click "Hide/Show details" from the Additional Software window.	pack enable STMicroelectronics "X-CUBE-BLE1/BlueNRG-MS" 1.0.0 "Wireless" "Controller"	
pack validate	Applies in the project all pack components enabled since the "pack validate" command was last called.	pack validate	
load <mcu></mcu>	Loads the selected MCU.	load STM32F101RCTx load STM32F101Z(F-G)Tx	
load <board number="" part=""> <allmodes nomode></allmodes nomode></board>	Loads the selected board with all peripherals configured in their default mode (allmodes) or without any configuration (nomode).	loadboard NUCLEO-F030R8 allmodes loadboard NUCLEO-F030R8 nomode	
config load <filename></filename>	Loads a previously saved configuration.	config load "C:\Cube\ccmram\ccmram.ioc"	
config save <filename></filename>	Saves the current configuration.	config save "C:\Cube\ccmram\ccmram.ioc"	
config saveext <filename></filename>	Saves the current configuration with all parameters, including those for which values have been kept to default.	config saveext "C:\Cube\ccmram\ccmram.ioc"	
config saveas <filename></filename>	Saves the current project under a new name.	config saveas "C:\Cube\ccmram2\ccmram2.ioc"	
csv pinout <filename></filename>	Exports the current pin configuration as a csv file. This file can be (later) imported into a board layout tool.	Csv pinout mypinout.csv	
script <filename></filename>	Runs all commands in the script file. There must be one command per line.	script myscript.txt	
project couplefilesbyip <0 1>	This option allows the user to choose between 0 (to generate the peripheral initializations in the main) and 1 (to generate each peripheral initialization in dedicated .c/.h files).	project couplefilesbyip 1	
setDriver <peripheral name=""> <hal ll="" =""></hal></peripheral>	For the supported series, STM32CubeMX can generate peripheral initialization code based on LL or on HAL drivers. This command line allows the user to choose, for each peripheral, between HAL- and LL-based code generation. By default code generation is based on HAL drivers.	setDriver ADC LL setDriver I2C HAL	

Table 1. Com	mand line sum	mary (continued)
--------------	---------------	------------------



Command line	Purpose	Example
generate code <path></path>	Generates only "STM32CubeMX generated" code and not a complete project (including STM32Cube firmware libraries and toolchains project files). To generate a project, use "project generate".	generate code C:\mypath
set tpl_path <path></path>	Sets the path to the source folder containing the .ftl user template files. All the template files stored in this folder are used for code generation.	set tpl_path C:\myTemplates\
set dest_path <path></path>	Sets the path to the destination folder that will hold the code generated according to user templates.	set dest_path C:\myMXProject\inc\
get tpl_path	Retrieves the path name of the user template source folder.	get tpl_path
get dest_path	Retrieves the path name of the user template destination folder.	get dest_path
SetStructure <advanced basic=""></advanced>	Selects the project structure to generate.	SetStructure Basic
SetCopyLibrary <copy <br="" all="" copy="" only="">copy as reference></copy>	Selects how the reference libraries are copied to the projects.	SetCopyLibrary "copy all"
project setCustomFWPath <customfwlocation> Specifies a path to STM32Cube MC software libraries different from STM32Cube repository path (specifi under Help > Updater settings).</customfwlocation>		project SetCustomFwPath "F:/SharedRepository/STM32Cube_F W_F0_V1.11.0"
project toolchain <toolchain></toolchain>	Specifies the toolchain to be used for the project. Use the "project generate" command to generate the project for that toolchain.	EWARM MDK-Arm STM32CubeIDE Makefile CMake
project name <name></name>	Specifies the project name.	project name ccmram
project path <path></path>	Specifies the path where to generate the project.	project path C:\Cube\ccmram
project generate	Generates the full project. ⁽¹⁾	project generate
login < email_adress> <password> <remember_me></remember_me></password>	Allows you to login to download software packages.	login john.smith@st.com mypassword y
exit	Ends STM32CubeMX process.	exit

Table 1	Command line	summarv	(continued)
		5 Summary	(continueu)

1. Use the login command before using this command.



3.4 Getting updates using STM32CubeMX

STM32CubeMX implements a mechanism to access the Internet and to:

- download embedded software packages: STM32Cube MCU packages (full releases and patches) and third-party packages (.pack) based on the Arm[®] CMIS pack format
- manage a user-defined list of third-party packs
- check for STM32CubeMX and embedded software packages updates
- perform self-updates of STM32CubeMX
- refresh STM32 MCUs descriptions and documentation offer.

Installation and update related submenus are available under the **Help** menu and from the home page as well.

Off-line updates can also be performed on computers without Internet access (see *Section 3.4.3*). This is done by browsing the filesystem and selecting available STM32Cube MCU packages.

If the PC on which STM32CubeMX runs is connected to a computer network using a proxy server, STM32CubeMX needs to connect to that server to access the Internet, get self-updates and download firmware packages. Refer to *Section 3.4.2* for a description of this connection configuration.

To view Windows default proxy settings, select Internet options from the Control panel and select LAN settings from the **Connections** tab (see *Figure 17*).

Internet Properties
General Security Privacy Content Connections Programs Advanced
To set up an Internet connection, click Setup Setup.
Dial-up and Virtual Private Network settings
Add
Add V <u>P</u> N
<u>R</u> emove
Choose Settings if you need to configure a proxy Settings
Never dial a connection
Dial whenever a network connection is not present
Always dial my default connection
Current None Set default
Local Area Network (LAN) settings
LAN Settings do not apply to dial-up connections. Choose Settings above for dial-up settings.
OK Cancel Apply

Figure 17. Displaying Windows default proxy settings



Several proxy types exist, and different network configurations are possible:

- Without proxy: the application directly accesses the web (Windows default configuration).
- Proxy without login/password
- Proxy with login/password: when using an Internet browser, a dialog box opens and prompts the user to enter its login/password.
- Web proxies with login/password: when using an Internet browser, a web page opens and prompts the user to enter its login/password.

If needed, contact your IT administrator for proxy information (proxy type, http address, port).

STM32CubeMX does not support web proxies. In this case, the user cannot benefit from the update mechanism and must manually copy the STM32Cube MCU packages from http://www.st.com/stm32cube to the repository. To do it, follow the sequence below:

- 1. Go to http://www.st.com/stm32cube and download the relevant STM32Cube MCU package from the *Associated Software* section.
- 2. Unzip the zip package to your STM32Cube repository. Find out the default repository folder location in the **Updater Settings** tab as shown in *Figure 18* (you might need to update it to use a different location or name).

3.4.1 Running STM32CubeMX behind a proxy server

When proxies are implementing full SSL inspection, STM32CubeMX must be configured to use the proxy certificate.

• On Windows:

Typically, it comes down to using Windows certificate list.

- a) there is no additional configuration necessary to run STM32CubeMX executable (it is already configured to use Windows certificate list)
- b) the command line must be adjusted to run STM32CubeMX from the command line:

```
cd <STM32CubeMX install path>
jre\bin\java -Djavax.net.ssl.trustStoreType=WINDOWS-ROOT -jar
STM32CubeMX.exe
```

- On Mac/Linux and on Windows systems when the proxy certificate is not in Windows certificate store, the certificate must be manually imported. This is done using keytool from a command prompt, as follows:
 - \$ cd <CUBEMX_INSTALL_DIR>/jre

```
$ bin/keytool -importcert -alias <your certificate alias name> -
keystore lib/security/cacerts -file <path to you proxy certificate
file>.crt
```

When prompted, enter the password: changeit

When prompted, accept to trust the certificate: yes

Then (Windows only) edit file <*CUBEMX_INSTALL_DIR*>/*STM32CubeMX.l4j.ini* and remove the line: -Djavax.net.ssl.trustStoreType=WINDOWS-ROOT



3.4.2 Updater configuration

To perform STM32Cube new library package installation or updates, the tool must be configured as follows:

- 1. Select Help > Updater Settings to open the Updater Settings window.
- 2. From the Updater Settings tab (see Figure 18)
 - a) Specify the repository destination folder where the downloaded packages will be stored.
 - b) Enable/Disable the automatic check for updates.

Figure 18. Updater Settings window

Updater Settings
Updater Settings Connection Parameters
Firmware Repository
Repository Folder
C:\Users\JohnDoe\STM32Cube\Repository Browse
Check and Update Settings
O Manual Check
Automatic Check Interval between two Checks (days) 5
Data Auto-Refresh
O No Auto-Refresh at Application start
 Auto-Refresh Data-only at Application start
O Auto-Refresh Data and Docs at Application start
Interval between two data-refreshs (days) 3
OK Cancel

- 3. In the **Connection Parameters** tab, specify the proxy server settings appropriate for your network configuration by selecting a proxy type among the following possibilities (see *Figure 19*):
 - No Proxy
 - Use System Proxy Parameters

On Windows, proxy parameters are retrieved from the PC system settings. Uncheck "Require Authentication" if a proxy server without login/password configuration is used.



- Manual Configuration of Proxy Server
 Enter the Proxy server http address and port number. Enter login/password information or uncheck "Require Authentication" if a proxy server without login/password configuration is used.
- 4. Optionally uncheck **Remember my credentials** to prevent STM32CubeMX to save encrypted login/password information in a file. This implies reentering login/password information each time STM32CubeMX is launched.
- 5. Click the **Check Connection** button to verify if the connection works. A green check mark appears to confirm that the connection operates correctly Check Connection

Figure 19. Connection Parameters tab - Manual Configuration of Proxy Serv	Figure 19	9. Connection	Parameters tab	- Manual Confi	ouration of Prop	xv Server
---------------------------------------------------------------------------	-----------	---------------	----------------	----------------	------------------	-----------

Updater Settings	X
Updater Settings Connection Parameters	
Proxy Server Type	
O No Proxy	
O Use System Proxy Parameters	
Manual Configuration of Proxy Server	
Manual Configuration of Proxy Server	
Proxy HTTP myproxy.mycompany.com	Port 8080
Authentication	
Require Authentication V Remember my Credentials	
User Login JohnDoe	
Password ••••••	
	Check Connection
	OK Cancel

- 6. Select **Help > Install New Libraries** submenu to select among a list of possible packages to install.
- 7. If the tool is configured for manual checks, select **Help > Check for Updates** to find out about new tool versions or firmware library patches available to install.
- Note: If STM32Cube MX is not connected to the network, or if it detects a connection failure, an icon is displayed close to the myST menu item showing that there is no network connection. When the user clicks on that icon, "Configure network" menu is displayed, and by clicking on it, the "Updater Settings/Connection parameters" dialog pops up. Once the STM32CubeMX is connected to the network, the network icon disappears.



Figure 20. Connection failure



3.4.3 Installing STM32 MCU packages

To download new STM32 MCU packages, follow the steps below:

 Select Help > Manage embedded software packages to open the Embedded Software Packages Manager (see Figure 21), or use Install/Remove button from the Home page.

Expand/collapse buttons + - expands/collapses the list of packages, respectively.

If the installation was performed using STM32CubeMX, all the packages available for download are displayed along with their version including the version currently installed on the user PC (if any), and the latest version available from <u>www.st.com</u>.

If no Internet access is available at that time, choose "From Local ...", then browse to select the zip file of the desired STM32Cube MCU package that has been previously downloaded. An integrity check is performed on the file to ensure that it is fully supported by STM32CubeMX.

The package is marked in green when the version installed matches the latest version available from *www.st.com*.

2. Click the checkbox to select a package then "Install Now" to start the download.

See *Figure 21* for an example.

Figure 21. Embedded Software Packages Manager window

🔤 Embedded Software P	Packages Manager								\times	
STM32Cube MCU Packages and embedded software packs releases										
Releases In	nformation was last refreshed 2	days ago.								
Infineon	RealThread 32Cube MCU Packages	SEGGER	WES STMicroelectroni	emotas	portGml Cesanta		quantropi ledOffice		olfSSL IA_DB	
Description			STWICTOElectroni	cs	Cesanta	Installed Versi		Available Ver	_	
STM32MP2									Sion	
STM32N6										
STM32Cube MC	U Package for STM32N6 Serie	s			1.0.0	1		1.0.0		
_									•	
► STM32U0										
Details										
STM32CubeN6 Firmw Main Changes	vare Package V1.0.0 / 30-Octo	ber-2024								
	of STM32CubeN6 firmware pa	ckage supporting S	TM32N6xx device	es						
Drivers										
	IS device drivers supporting STI									
	LL drivers available for all periph <u>Advivers supporting NUCLEO-N</u>		N6570-DK hoarde							
From Local	From Url					Refresh	Install	Remove	Close	



3.4.4 Installing STM32 MCU package patches

Use the procedure described in Section 3.4.3 to download STM32 MCU package patches.

A library patch, such as STM32Cube_FW_F7_1.4.1, can be easily identified by the version number, whose third digit is non-null (e.g. '1' for the 1.4.1 version).

The patch is not a complete library package but only the set of library files that need to be updated. The patched files go on top of the original package (e.g. STM32Cube_FW_F7_1.4.1 complements STM32Cube_FW_F7_1.4.0 package).

Prior to 4.17 version, STM32CubeMX copies the patches within the original baseline directory (e.g. STM32Cube_FW_F7_V1.4.1 patched files are copied within the directory called STM32Cube_FW_F7_V1.4.0).

Starting with STM32CubeMX 4.17, downloading a patch leads to the creation of a dedicated directory. As an example, downloading STM32Cube_FW_F7_V1.4.1 patch creates the STM32Cube_FW_F7_V1.4.1 directory that contains the original STM32Cube_FW_F7_V1.4.0 baseline plus the patched files contained in STM32Cube_FW_F7_V1.4.1 package.

Users can then choose to go on using the original package (without patches) for some projects and upgrade to a patched version for others projects.

3.4.5 Installing embedded software packs

Starting from the release 4.24, STM32CubeMX offers the possibility to select third-party embedded software packages coming in the Arm[®] Keil[™] CMSIS-Pack format (.pack), whose contents are described thanks to the pack description (.pdsc) file. Reference documentation is available from http://www.keil.com.

 Select Help > Manage embedded software packages to open the New Libraries Manager window (see *Figure 22*), or use Install/Remove button from the Home page, or from the project Pinout & Configuration view (select Software Packs > Manage Software Packs).

Use Expand/collapse buttons + - to expand/collapse the list of packages, respectively.



Embedded Software Pa	ackages Manager e MCU Packages and embe	edded software pad	cks releases STMicroelectronic	s packages		× + -
	formation was last refreshed					
Infineon	RealThread	SEGGER	WES Vemota	s portGm Cesanta	hbH quantropi EmbeddedOffice	wolfSSL ITTIA_DB
Status	Description	X-CUBE-AI packa	ge			Available Version
	Artificial Intelligence	(Size : 21.48 MB)				10.0.0
	Artificial Intelligence	(Size : 21.2 MB)				9.1.0
Details						
Artificial Intelligence				Click Refresh	to get the latest version	
From Local	From Url				Refresh Install	Remove Close

Figure 22. Managing embedded software packages - Help menu

2. Click **From Local** ... button to browse the computer filesystem and select an embedded software package. STM32Cube MCU packages come as .zip archives and embedded software packs come as .pack archives.

This action is required in the following cases:

- No Internet access is possible but the embedded software package is available locally on the computer.
- The embedded software package is not public and hence not available on Internet. For such packages, STM32CubeMX cannot detect and propose updates.
- Click From URL... button to specify the download location from Internet for one of the pack .pdsc file or from the vendor pack index (.pidx).
 Proceed as follow:
 - a) Choose From URL ... and click New (see Figure 23).
 - b) Specify the .pdsc file url. As an example, the url of Oryx-Embedded middleware pack is https://www.oryx-embedded.com/download/pack/Oryx-Embedded.Middleware.pdsc (see *Figure 24*).



. iguio	- Lo. managin	ig embedded	Solution pe	Ionagoe	, naanig		<u></u>
MX Embedde	er Defined Packs Mana	ger				×	×
	Manage Urls	for user defined em	bedded software	packs			+ -
Infineon	Vendor	Name		UR	L	1	wolfSSL TTIA_DB
.							Version
	Add new U	Jrl :ks, please enter url to	a aithar ana af tha	fallowing	×	-	0.0
	- A valid pd	lsc (Ex: http://www.ve ick index (Ex: http://w	endor.com/pack/V	endor.Pack			1.0
Details			Check	OK	Cancel	-	
Release ver			Check	UK	Cancer		
Release info							
Artificial Inte improvemen - stedgeai - bug fixing							
From Loci			New	lemove	ОК	Cancel	Close

Figure 23. Managing embedded software packages - Adding a new url

c) Click the **Check** button to verify that the provided url is valid (see *Figure 24*).

Figure 24. Checking the validity of vendor pack.pdsc file url



d) Click **OK**. The pack pdsc information is now available in the user defined pack list (see *Figure 25*).

To delete a url from the list, select the url checkbox and click **Remove**.



Vendor	Name	URL
Oryx-Embedded	Middleware	http://www.oryx-embedded.com/download/pack/Oryx-Embedded.Middleware.pds

Figure 25. User-defined list of software packs

e) Click **OK** to close the window and start retrieving psdc information. Upon successful completion, the available pack versions are shown in the list of libraries that can be installed. Use the corresponding checkbox to select a given release.

Image: Manager	×
STM32Cube MCU Packages and embedded software packs releases	+ -
Releases Information was last refreshed 1 days ago.	
Infineon RealThread SEGGER WES emotas portGmbH quantropi	wolfSSL
ন্দ STM32Cube MCU Packages ন্দ STMicroelectronics Cesanta EmbeddedOffice	ITTIA_DB
Status Description A	vailable Version
▼ a a a a I-CUBE-FS-RTOS	
EmbeddedOffice.I-CUBE-FS-RTOS (Size : 951.69 KB)	1.0.1
EmbeddedOffice.I-CUBE-FS-RTOS (Size : 950.49 KB)	1.0.0
Details	
Release version : 1.0.1	
Release date : 2024-06-19	
Release information :	
Added: - Small getting startet description	
Fixed:	
From Local From Url Refresh Install Remov	e Close

Figure 26. Selecting an embedded software pack release



f) Click Install Now to start downloading the software pack. A progress bar opens to indicate the installation progress. If the pack comes with a license agreement, a window pops up to ask for user's acceptance (see *Figure 27*). When the installation is successful, the check box turns green (see *Figure 28*).

The user can then add software components from this pack to its projects.

MX Embedded Software Packages Manager				\times
STM32Cube MCU Packages and embedded software particular	cks releases			+ -
Rel Licensing Agreement			×	wolfSSL
577 STM32 STMicroelectronics X-CUBE-AI 10.0.0 License Agree	ement			ITTIA_DB
Please read and accept the following agreement carefu	lly to finish the ins	stallation:		able Version
				10.0.0
Click here to open the licen	se agreement			9.1.0
Details				
Release version				
I have read, and I agree to the terms of this license	agreement			
Release inform O I do not accept the terms of this license agreement				
Artificial Intellig improvements/e - stedgeai command line based on ST Edge AI Core 1.0.0		Finish	Cancel	
- stedgear command the based of ST Edge AF Core 1.0.0				
From Local From Url	Refresh	Install	Remove	Close

Figure 27. License agreement acceptance



Embedded Software Packages M	lanager			×		
STM32Cube MCU F	STM32Cube MCU Packages and embedded software packs releases					
Releases Informatio	n was last refreshed less than one hour ag	0.				
Infineon RealThread	SEGGER WES emota	s portGi Cesanta	mbH quantrop EmbeddedOffice			
Status	Description			Available Version		
•	X-CUBE-AI			•		
•	Artificial Intelligence			10.0.0		
	Artificial Intelligence (Size : 21.2 MB)			9.1.0		
Details						
From Local From U	1	Refresh	Install Rem	ove Close		

Figure 28. Embedded software pack release - Successful installation

3.4.6 Removing already installed embedded software packages

To clean up the repository from old library versions, thus saving disk space, proceed as follows (see figures 29 and 30)

- 1. Select Help > Manage embedded software packages to open the Embedded Software Packages Manager, or use Install/Remove button from the Home page.
- 2. Click a green checkbox to select a package available in stm32cube repository.
- 3. Click the **Remove Now** button and confirm. A progress window opens to show the deletion status.



Figure 29. Removing a package

🔤 Embedded Software Packages Manager X						
STM32Cube MCU Packa	ges and embedded softwa	are packs releas	ses			
Releases Information was	last refreshed 1 hours ago.					
RealThread SEGGER	WES emotas				wolfSSL	
🖙 STM32Cube MCU Packages	κ STMicroelectronics	Cesanta E	mbeddedOffice	ITTIA_DB	Infineon	
Description			Installed Versi	on Available	Version	
STM32Cube MCU Package for	STM32N6 Series		1.0.0	1.(0.0	
▼ STM32U0						
Details						
STM32CubeN6 Firmware Package	V1.0.0 / 30-October-2024					
Main Changes						
First official release of STM32Cube	eN6 firmware package sup	porting STM32	N6xx devices			
Drivers						
- OMOLO devices drivers expensions OTM20NGrun devices						
From Local From Url		Refresh	Install	Remove	Close	

Figure 30. Confirmation message

Packag	ges Manager ×
?	You are about to remove the following package(s) : Please note: Once package will be removed, You will not anymore be able to generate projects that were based on this package.
	- FW.N6.1.0.0 (C:\Users\daymarwa\STM32Cube\Repository\STM32Cube_FW_N6_V1.0.0)
	Please confirm package(s) deletion
	Yes No

3.4.7 Checking for updates

Starting with version V6.12.0, if there is a new CubeFW, X-Cube, or I-Cube available for update, an icon is displayed close to the myST menu. The same dedicated icon is displayed left to the "CHECK FOR UPDATES" button. When the user clicks on that icon, the Update Manager window opens.



I32	File	Window	Help	🔹 🚺 💿 🚰 🔼 💭 🍭 🔆 🟹
ome STM32N6454	A0HxQ > Untitled - Pinout a	& Configuration		Check for update ALN RATE CODE
Existing Projects		New Project		Manage software installations
Recent Opened Pro	pjects	I need to :		Check for STM32CubeMX and embedded soft
NVIC_threadx_is Last modified date : 2		Start My projec	ct from MCU	CHECK FOR UPDATES
STM32H57x_boo Last modified date : 2			ct from ST Board	Install or remove embedded software packages
STM32H57x_boo Last modified date : 2			BOARD SELECTOR	
H562_bootpath.id Last modified date : 2			ct from Example	Construction Construction
H562_bootpath.ie Last modified date : 2				Δ
1 Data collection in	formation notice	Other Services		About STM32 🔗 External Tools

Figure 31. Checking for available updates

When the updater is configured for automatic checks, it regularly verifies if updates are available.

When automatic checks have been disabled in the updater settings window, the user can manually check if updates are available:

- Click the icon to open the Update Manager window or Select Help > Check for Updates. All the updates available for the user current installation are listed.
- 2. Click the check box to select a package, and then Install Now to download the update.

Warning:	 When performing STM32CubeMX self-updates. administrator rights are required when downloading the self-update package and during the STM32CubeMX launch that completes the update process: 1. Launch STM32CubeMX with administrator account 2. Go to Help > Check for updates menu, select MX update package and click "Install now" to start the download 3. Re-launch STM32CubeMX with the administrator account
	to finish the update process



Figure	32.	Help	menu:	checking	for	updates
i iguio	~	11010	monai	onooning		apaatoo

K Check	Update Manager	×
	Updates are available for STM32CubeMX, STM32Cube MCU Packages.	
	Update Information was last refreshed 14 days ago.	
	new STM32CubeMX Release Administrator rights are required to download the update package and at next launch to complete the update proces	s
	MX.6.1.0 New version of STM32CubeMX Software	
	new STM32Cube MCU Package patches	
	FW.F0.1.11.1 STM32Cube MCU Patch Package version 1.11.0 for STM32F0xx Series. (Size : 32.3 MB).	
Details		
	Refresh Install Now Close	



4 STM32CubeMX user interface

STM32CubeMX user interface comes with three main views the user can navigate through using convenient breadcrumbs, namely the **Home** page, the **New project** window, and the project page. They come with panels, buttons and menus allowing users to take actions and make configuration choices with a single click.

The user interface is detailed in the following sections.

For C code generation, although the user can switch back and forth between the different configuration views, it is recommended to follow the sequence below:

- 1. From the Project Manager view, configure the project settings.
- From the Mode panel in the Pinout & Configuration view, configure the RCC peripheral by enabling the external clocks, master output clocks, audio input clocks (when relevant for your application). This automatically displays more options on the Clock configuration view (see *Figure 180*). Then, select the features (peripherals, middlewares) and their operating modes relevant to the application.
- 3. If necessary, adjust the clock tree configuration from the clock configuration view.
- 4. From the Configuration panel in the **Pinout & Configuration** view configure the parameters required to initialize the peripherals and middleware operating modes.
- 5. Generate the initialization C code by clicking **GENERATE CODE**

4.1 Home page

This is the first window that opens up when launching STM32CubeMX (see *Figure 33*). Closing it closes down the application. It offers shortcuts for some top level menus, an image carousel displaying STM32 latest news, as well as links to social network sites and external tools. Top-level menus and social network links remain accessible from the subsequent project page and are detailed in the following sections.

STM32CubeMX Untitled: STM32N645A0HxQ a 🕲 🖪 🖸 X 🞧 🍭 🔆 🖅 File Window Help Home STM32N645A0HxQ Untitled - Pinout & Configuration Existing Projects New Project Manage software installations Recent Opened Projects Check for STM32CubeMX and embedded soft. I need to : CHECK FOR UPDATES NVIC_threadx_issue.ioc ΜХ Last modified date : 24/01/2025 8:25:13 Start My project from MCU Install or remove embedded software packages ACCESS TO MCU SELECTOR STM32H57x_bootpath.ioc МΧ Last modified date : 23/01/2025 15:50:56 Start My project from ST Board STM32H57x_bootpath.ioc MX ESS TO BOARD SELECTOR Last modified date : 23/01/2025 15:50:32 H562_bootpath.ioc MX Start My project from Example Last modified date : 23/01/2025 15:43:17 TO EXAMPLE SEI H562 bootpath.ioc ΜХ Last modified date : 23/01/2025 15:41:27 Other Projects Other Services Data co S TO COMPARE PRO 🏶 About STM32 👘 🌮 External Tools

Figure 33. STM32CubeMX home page



UM1718 Rev 47



4.1.1 File menu

Refer to Table 2 for a description of the File menu and shortcuts.

Table 2. Home page shortcuts		
Description	Home n	

Name Keyboard shortcut	Description	Home page shortcut
New Project Ctrl-N	Opens a new project window showing all supported MCUs and a set of STMicroelectronics boards to choose from ⁽¹⁾ .	To create a new project starting from a board click ACCESS TO BOARD SELECTOR To create a new project starting from an MCU click ACCESS TO MCU SELECTOR
Load Project Ctrl-L	Loads an existing STM32CubeMX project configuration by selecting an STM32CubeMX configuration .ioc file (see <i>Caution:</i>).	Under Other project, click browse icon
Import Project Ctrl-I	Opens a new window to select the configuration file to be imported as well as the import settings. The import is possible only if you start from an empty MCU configuration. Otherwise, the menu is disabled ⁽²⁾ .	None
Save Project Ctrl-S	Saves current project configuration (pinout, clock tree, peripherals, middlewares, Power Consumption Calculator) as a new project. This action creates a project folder including an .ioc file, according to user defined project settings.	None
Save Project as Ctrl-A	Saves the current project.	None
Close Project Ctrl-C	Closes the current project and switches back to the welcome page.	None
Recent Projects none	Displays the list of the five most recently saved projects.	Under Recent Project , click icon next to the project name.
Generate Report Ctrl-R	Saves the project current configuration as two documents (pdf and text formats).	None
Exit Ctrl-X	Proposes to save the project (if needed), then closes the application.	To close the window and the application click on .

On **New project**: to avoid any popup error messages at this stage, make sure an Internet connection is available (Connection Parameters tab under Help > Updater settings menu) or that Data Auto-refresh settings are set to No Auto-Refresh at application start (Updater Settings tab under Help > Updater Settings menu). 1.

2. On Import, a status window displays the warnings or errors detected when checking for import conflicts. The user can then decide to cancel the import.



Caution: On project load: STM32CubeMX detects if the project was created with an older version of the tool and if this is the case, it proposes the user to either migrate to use the latest STM32CubeMX database and STM32Cube firmware version, or to continue. Prior to STM32CubeMX 4.17, clicking Continue still upgrades to the latest database "compatible" with the STM32Cube firmware version used by the project. Starting from STM32CubeMX 4.17, clicking Continue keeps the database used to create the project untouched. If the required database version is not available on the computer, it is automatically downloaded. When upgrading to a new version of STM32CubeMX, make sure to always backup your projects before loading the new project (especially when the project includes user code).

4.1.2 Window menu and Outputs tabs

The **Window menu** allows the user to access the **Outputs** function.

Name	Description
	Selecting/deselecting Outputs from the Window menu hides/shows the following Outputs tabs at the bottom of STM32CubeMX project page (see <i>Figure 34</i>)
Outruite	 MCUs selection: lists the MCUs of a given family matching the user criteria (series, peripherals, package,) when an MCU was selected last⁽¹⁾.
Outputs	 Outputs: displays a non-exhaustive list of the actions performed, raised errors and warnings (see <i>Figure 35</i>) found upon user actions.
	 IP assignment rules
	– MMT Output Log
Font size	Makes possible to change font size settings. STM32CubeMX must be re-launched for changes to take effect.

Table 3. Window menu

1. Selecting a different MCU from the list resets the current project configuration and switches to the new MCU. The user is then prompted to confirm this action before proceeding.

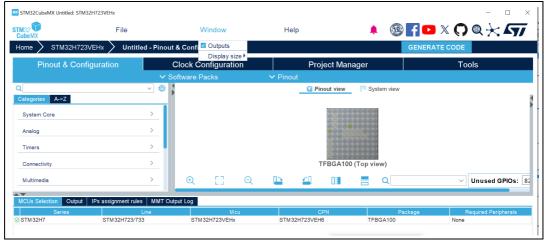


Figure 34. Window menu



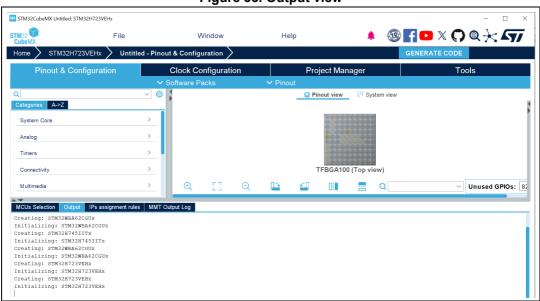


Figure 35. Output view

4.1.3 Help menu

Refer to *Table 4* for a description of the **Help** menu and shortcuts.

Name Keyboard shortcut	Description	Home page shortcut
Help F1	Opens the STM32CubeMX user manual.	None
About Alt-A	Shows version information.	None
Docs & Resources Alt-D	Displays the official documentation available for the MCU used in the current project.	None
Video Tutorials Alt-V	Opens the Video Tutorial browser that proposes a list of videos and allows the user to launch a video in one click.	None
Refresh Data Alt-R	Opens a dialog window that proposes to refresh STM32CubeMX database with STM32 MCU latest information (description and list of official documents), and allows the user to download of all official documentation in one shot.	None
Check for Updates Alt-C	Shows the software and firmware release updates available for download.	Click CHECK FOR UPDATES
Manage embedded software packages Alt-U	Shows all the embedded software packages available for installation. A green check box indicates that the package is already installed in the user repository folder (the repository folder location is specified under Help > Updater Settings menu).	Click INSTALL/REMOVE

Table 4. Help menu shortcuts



······································				
Name Keyboard shortcut	Description	Home page shortcut		
Updater Settings Alt-S	Opens the updater settings window to configure manual versus automatic updates, proxy settings for Internet connections, repository folder where the downloaded software and firmware releases will be stored.	None		
User Preferences	Opens the user preference window to enable or disable collect of features usage statistics.	None		

Table 4. Hel	p menu shortcuts	(continued)

4.1.4 Social links

Developer communities on popular social platforms such as FacebookTM, STM32 YouTubeTM channel, as well as ST Community can be accessed from the STM32CubeMX toolbar (see *Figure 36*).

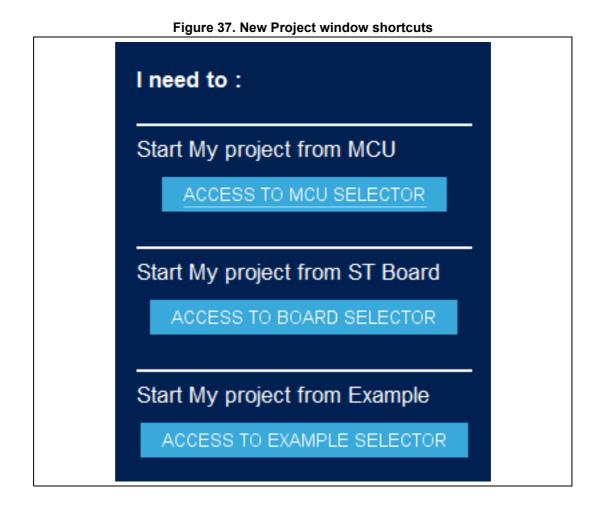
Figure 36. Link to social platforms



4.2 New Project window

The New Project window is accessible through the File Menu, or directly through shortcuts from the Home page (see *Figure 37*).





The main purpose is to select from the STM32 portfolio the microcontroller or board that best fits the user application needs, or simply to get started using an example project.

This window shows three tabs to choose from:

- an MCU selector tab (offering a list of target processors)
- a Board selector tab (showing a list of STMicroelectronics boards)
- an **Example selector** tab (allows the user to browse and open an example project)

The new project window also features a **Cross selector** tab (allows the user to find, for a given MCU/MPU part number and for a set of criteria, the best replacement within the STM32 portfolio)

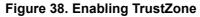
For the STM32L5 series the security features of the Arm Cortex-M33 processor and its Arm[®] TrustZone^{®(a)} for Armv8-M are combined with ST security implementation. Selecting an STM32L5 MCU or board requires to choose whether to activate Arm[®] TrustZone[®] (hardware security) or not (see *Figure 38*). The project is adjusted accordingly:

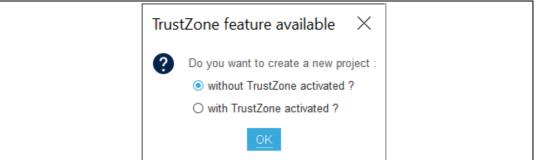
if Arm[®] TrustZone[®] is not activated, the solution is the same as for other STM32Lx series

a. TrustZone is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



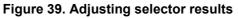
• if Arm[®] TrustZone[®] is activated, the project configuration and the generated project shows specificities related to the security features (refer to dedicated sections in this manual).





The selectors result view can be adjusted (see *Figure 39*):

- Left click the column to sort
- Right click to add/remove columns



UMPU Filters 📩 🔂 🔂 🔿		Features	Block Dia	igram Docs & Resources	CAD Resources	s 📑 Datas	heet	📑 Buy		Start Project
Commercial wba ~	•									
۹ 🔍 🚽 + -	-					(IM32			
RODUCT INFO	~			57						
Segment	>									
Series	>									
Line	>			22000	****					
Marketing Status	>									
Price	<u>`</u>									
	<u> </u>									
Package	>			The first high pe						
Package Core	>			The first high-pe			G	77		
-				The first high-pe STM32 MCU with			5	7/		
Core	>						5	7/		
Core Coprocessor	>			STM32 MCU with			Ş	77		
Core Coprocessor AEMORY	> > ~		all and an	STM32 MCU with			5	77		
Core Coprocessor KEMORY Fissh From 512 to 2048 (kBytes)	> > >	Js/MPUs List: 85 it	all and an	STM32 MCU with			5	77		🔥 Expo
Core Coprocessor HEMORY Flash From 512 to 2048 (kBytes) 512 20	> > >	•	ems display	STM32 MCU with	AI accel	leration	× RAM	× 10	× Flash	× Frequency
Core Coprocessor HEMORY Flash From 512 to 2048 (kBytes) 512 20	> > >	÷ ☆ STM	ems display Commercial Part No 32WBA50KGU6	STM32 MCU with	AI accel	eration ۲۰۰۲ (UFQFPN 32 5x5x0.55	× RAM . 64 kBytes	× 10 20	1024 kBytes	× Frequency 100 MHz
Core Coprocessor TEMORY Flash From 512 to 2048 (HBytes) 512 20 EPROM = 0 (Bytes) 0	> > >	☆ STM	Commercial Pert No 32WBA50KGU6 32WBA50KGU6TR	Ickon the column name to the list of the possible Processible Plant and the possible Plant	AI accel	Package UFQFPN 32 5x5x0.55 UFQFPN 32 5x5x0.55	< RAM 64 kBytes 64 kBytes	× ¥0 20 20	1024 kBytes 1024 kBytes	× Frequency 100 MHz 100 MHz
Core Coprocessor Flagh From 512 to 2048 (kBytes) 512 20 CEEPCOM = 0 (Bytes) 0 RAM Total From 54 to 512 (kBytes)	- MC	☆ STM ☆ STM ☆ STM	commercial Part No 32WBA50KGU6 32WBA50KGU6TR 32WBA52CEU6	STM32 MCU with ick on the column name to the list of the possible Perios Meterers Marketon Perios Marketon Perios Remove Current Column	AI accel	Packape UFGFPN 32 5x50.55 UFGFPN 42 5x50.55 UFGFPN 48 7x70.55	< RAM 64 kBytes 64 kBytes 86 kBytes	× ro 20 20 35	1024 kBytes 1024 kBytes 512 kBytes	× Frequency 100 MHz 100 MHz 100 MHz
Core Coprocessor TEMORY From 512 to 2048 (LBytes) 612 20 ERROM = 0 (Bytes) 0 RM Total From 54 to 512 (LBytes) 64	- MC	☆ STM ☆ STM ☆ STM ☆ STM ☆ STM	ems display Commercial Part No 32WBA50KGU6 32WBA50KGU6TR 32WBA50CEU6 32WBA52CEU6TR	STM32 MCU with lick on the column name to the lick of the possible Peter Revenue Menkelow Peter Re	Sistur X Board	Peskep UFGFPN 32 6x5x0.55 UFGFPN 32 6x5x0.55 UFGFPN 48 7x7x0.55 UFGFPN 48 7x7x0.55	< RAM 64 kBytes 64 kBytes 86 kBytes 96 kBytes	× vo 20 20 35 35	1024 kBytes 1024 kBytes 512 kBytes 512 kBytes	× Frequency 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz
Core Coprocessor TEMORY From 512 to 2048 (LBytes) 612 20 ERROM = 0 (Bytes) 0 RM Total From 54 to 512 (LBytes) 64	- MC	· ☆ STM ☆ STM ☆ STM ☆ STM ☆ STM	commercial Part No 322WBA50KGU6TR 322WBA50KGU6TR 322WBA52CEU6 32WBA52CEU6TR 32WBA52CEU7	STM32 MCU with:		Peckage UFQFPN 32 δx5x0.55 UFQFPN 32 δx5x0.55 UFQFPN 48 7x7x0.55 UFQFPN 48 7x7x0.55 UFQFPN 48 7x7x0.55	< RAM 64 kBytes 64 kBytes 86 kBytes 96 kBytes 96 kBytes	× vo 20 20 35 35 35 35	1024 kBytes 1024 kBytes 512 kBytes 512 kBytes 512 kBytes 512 kBytes	Frequency 100 M Hz
Core Coprocessor TEMORY From 512 to 2048 (LBytes) 612 20 ERROM = 0 (Bytes) 0 RM Total From 54 to 512 (LBytes) 64	48 MCI		commercial Part No 32WBA50KGU6 32WBA50KGU6TR 32WBA52CEU6 32WBA52CEU6TR 32WBA52CEU7 32WBA52CEU7TR	Itickon the column name to the list of the passible Peter Research and the source Peter Research	Status × Board	Package Processe UPGFPN 32 5x800 55 UPGFPN 48 7x700 55 UPGFPN 48 7x700 55 UPGFPN 48 7x700 55 UPGFPN 48 7x700 55	 RAM 64 kBytes 64 kBytes 66 kBytes 96 kBytes 96 kBytes 	× 10 20 20 35 35 35 35	1024 kBytes 1024 kBytes 512 kBytes 512 kBytes 512 kBytes 512 kBytes	Frequency 100 M Hz 100 M Hz
Core Coprocessor Flight, From 512 to 2048 (LB/tiss) 512 512 20 513 20 6 512 6 51 64 51	48 MCI	☆ STM	ems display 2009 Automatical Part No 32WBA50KGU6 32WBA50KGU6TR 32WBA52CEU6TR 32WBA52CEU7TR 32WBA52CEU7TR 32WBA52CEU7TR	STM32 MCU with: lick on the column name to the lick of the possible Vertice of the possible Patie Al Columns for Selected Criterias Reset to Defue Columns Reset to Defue Columns Vertice of Unselect al Columns Vertice of Unselect al Columns		Pressor UFGFPN 32 5580.56 UFGFPN 32 5580.56 UFGFPN 48 77/0.55 UFGFPN 48 77/0.55	 RAM 64 kBytes 64 kBytes 66 kBytes 96 kBytes 96 kBytes 96 kBytes 128 kBytes 	× 10 20 20 35 35 35 35 35 35 35	1024 kBytes 1024 kBytes 512 kBytes 512 kBytes 512 kBytes 512 kBytes 1024 kBytes	X Frequency 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz
Core Coprocessor EMORY Filesh Filesh From 512 to 2048 (kBytes) 512 20 ERROM = 0 (kBytes) 20 RM Total From 64 to 512 (kBytes) 64 51 67 67 68 5 64 5 65 5 66 5 67 5 68 5	48 MCI	☆ STM	display Commercial Part No 32WBA50KGU6TR 32WBA50KGU6TR 32WBA50CEU6TR 32WBA52CEU7TR 32WBA52CEU7TR 32WBA52CEU6TR 32WBA52CEU6TR 32WBA52CEU6TR 32WBA52CEU6TR 32WBA52CEU6TR 32WBA52CEU6TR	Iteron the column name to the iter of the passible Peter Research and the source Pack All Columns Horizontal Scrold Remove Current Columns Select of Unesket all Columns Select of Unesket all Columns © Commercial Part No © Part No	Status × Board	Peckage UFGTPN 32 5x5x0.55. UFGTPN 32 5x5x0.55. UFGTPN 32 5x5x0.65. UFGTPN 45 7x70.65. UFGTPN 46 7x70.55. UFGTPN 48 7x70.55. UFGTPN 48 7x70.55.	 RAM G4 kBytes G4 kBytes G6 kBytes 96 kBytes 96 kBytes 96 kBytes 128 kBytes 128 kBytes 	× 10 20 20 35 35 35 35 35 35 35 35	1024 kBytes 1024 kBytes 512 kBytes 512 kBytes 512 kBytes 512 kBytes 512 kBytes 1024 kBytes 1024 kBytes	X Frequency 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz
Core Coprocessor Filesh From 512 to 2048 (H8ytes) 512 512 20 6 Coprocessor 6 From 54 to 512 (H8ytes) 64 From 64 to 512 (H8ytes) 64 Com RAM 64 612 (H8ytes) 64 0 (H8ytes) 64 0 (H8ytes)	48 MCI		display Commercial Part No 32WBA50KGU6 32WBA50KGU6TR 32WBA52CEU6 32WBA52CEU7TR 32WBA52CEU7TR 32WBA52CGU6 32WBA52CGU6TR 32WBA52CGU7T	STM32 MCU with: tex on the column name to the list of the possible Columns for the possible Park All Columns Horizontal Seroid Reset to Defaul Columns Reset to Defaul Columns Commercial Columns Commercial Columns Commercial Reset to Columns Reset to Reset to Reset to Columns Reset to Reset to Reset to Columns Reset to Reset to	Status X Board	Pressee (ΓΟΓΡΓΝ 32 6500 55. UFGΓΡΝ 32 6500 55. UFGΓΡΝ 32 6500 55. UFGΓΡΝ 48 7700 55. UFG	RAM 64 kBytes 64 kBytes 96 kBytes 96 kBytes 96 kBytes 96 kBytes 128 kBytes 128 kBytes 128 kBytes	20 20 35 35 35 35 35 35 35 35 35	1024 kBytes 1024 kBytes 512 kBytes 512 kBytes 512 kBytes 512 kBytes 1024 kBytes 1024 kBytes 1024 kBytes	× Frequency 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz
Core Coprocessor EMORY Filesh Filesh From 512 to 2048 (kBytes) 512 20 ERROM = 0 (kBytes) 20 RM Total From 64 to 512 (kBytes) 64 51 67 67 68 5 64 5 65 5 66 5 67 5 68 5	48 MCI	Image: Constraint of the second se	Commercial Part No 32/WBA50/KGU6 32/WBA50/KGU6TR 32/WBA52CEU6 32/WBA52CEU6 32/WBA52CEU7 32/WBA52CGU7TR 32/WBA52CGU8TR 32/WBA52CGU7TR	Itickon the column name to the its of the ossible Peter Research and the source Pack All Columns Horizontal Scrold Remove Current Columns Select of Unesket all Columns Select of Unesket all Columns © Commercial Part No © Part No © Part No © Marketing Status	AI accel	Person υFGPPN 32 6x50.55. υFGPPN 32 6x50.55. υFGPPN 32 6x50.55. υFGPPN 477.05.	RAM 64 kBytes 64 kBytes 66 kBytes 96 kBytes 96 kBytes 128 kBytes 128 kBytes 128 kBytes 128 kBytes 128 kBytes	× 10 20 22 35 35 35 35 35 35 35 35 35 35 35	1024 kBytes 1024 kBytes 512 kBytes 512 kBytes 512 kBytes 1024 kBytes 1024 kBytes 1024 kBytes 1024 kBytes	X Frequency 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz
Core Coprocessor Filesh From 512 to 2048 (H8ytes) 512 512 20 6 Coprocessor 6 From 54 to 512 (H8ytes) 64 From 64 to 512 (H8ytes) 64 Com RAM 64 612 (H8ytes) 64 0 (H8ytes) 64 0 (H8ytes)	48 MCI	¹	display Commercial Part No 32WBA50KGU6 32WBA50KGU6TR 32WBA52CEU6 32WBA52CEU7TR 32WBA52CEU7TR 32WBA52CGU6 32WBA52CGU6TR 32WBA52CGU7T	STM32 MCU with: tex on the column name to the list of the possible Columns for the possible Park All Columns Horizontal Seroid Reset to Defaul Columns Reset to Defaul Columns Commercial Columns Commercial Columns Commercial Reset to Columns Reset to Reset to Reset to Columns Reset to Reset to Reset to Columns Reset to Reset to	Status X Board	Pressee (ΓΟΓΡΓΝ 32 6500 55. UFGΓΡΝ 32 6500 55. UFGΓΡΝ 32 6500 55. UFGΓΡΝ 48 7700 55. UFG	64 RBytes 64 kBytes 66 kBytes 96 kBytes 96 kBytes 96 kBytes 96 kBytes 128 kBytes 128 kBytes 128 kBytes 128 kBytes	20 20 35 35 35 35 35 35 35 35 35	1024 kBytes 1024 kBytes 512 kBytes 512 kBytes 512 kBytes 512 kBytes 1024 kBytes 1024 kBytes 1024 kBytes	× Frequency 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz 100 MHz

4.2.1 MCU selector

MCU selection

The MCU selector enables filtering on a combination of criteria: series, lines, packages, peripherals, or additional characteristics such as price, memory size, number of I/Os (see *Figure 40*), and on their graphics capabilities as well.



U	Μ	17	18

New Project MCU/MPU Selector Board Selector E	xample Selector	Cross Sele	stor				Starting the selecting an	project after MCUs
ICU/MPU Filters				, Quick Access to resources	;		4	
Siltering Darameters 📩 📩 🔂 🔂 🗘	-Load favorite MCUs -Save/Load search criteria -Reset search criteria	eria Fe	atu Block Diagr	Docs & Resour CAD Resour	r 📔 Data	sh 🖸	G→ St	art Proj
Commercial	~	ST	132C0 Series					
Q) +-	\$ S	TM32C011D6Y3TF	Mainstream Arm Cortex-M memory, 6 Kbytes RAM, 4 comm. I/F, 2-3.6V				c,
PRODUCT INFO	~			Unit Price for 10kU (US\$) : NA				
Segment	>		COMING SOON Stay tuned !		<u>.</u>	WLCSP 12 1.7x	x1.42x0.6 P 0.35	i mm
Series	>							
Line	>			instream microcontrollers are based on				
Marketing Status				MHz frequency. Offering a high level of industrial and appliance domains and re				of
				memory protection unit (MPU), high-sp				and up
			22 Khudan of Back stars	and management to the same and to the surface to	(aa) DMA aa ay	analise vanue of		
Price	>							
Price Package	> >							
		MCUs/MP	Us List: 4682 items 🗘	MCUs list	Export the I	VICUs list into an e	xcel file	📩 Export
Package Core	>	MCUs/MP	Commercial Pa	t No. Reference Marketin X Board >		MCUs list into an e		Trequency ×
Package		MCUs/MP	Commercial Pa	t No. Reference Marketin X Board >		RAM × 1/0		Frequency ×
Package Core	>			t No. Reference Marketin X Board >	< Package ×	RAM × ⊮O kBytes 18	× Flash ×	Frequency × 48 MHz
Package Core	>	- \$	Commercial Pa Part STM32C011D6Y	t No Reference Marketin × Board > 32 STM 32C0 Coming s	Package X WLCSP 12 1 6 WLCSP 12 1 6	RAM × ⊮O kBytes 18	× Flash × 32 kBytes	Frequency > 48 MHz 48 MHz
Package Core Coprocessor MEMORY	> > > >	• ☆ ☆	Commercial Pa Part STM32C011D6Y STM32C011D6	t No Reference Marketin × Board × 32 STM 32C0 Coming s STM 32C0 Active	Package × WLCSP 12 1 6 WLCSP 12 1 6 TSSOP-20 6	RAM × 1/0 kBytes 18 kBytes 18	× Flash × 32 kBytes 32 kBytes	Frequency> 48 MHz 48 MHz 48 MHz 48 MHz
Package Core Coprocessor	> > > >	• ☆ ☆ ☆ ☆ ☆ ☆ ☆	Commercial Pa Part STM32C011D6Y STM32C011D6 STM32C011F4P3 STM32C011F4P6 STM32C011F4P6 STM32C011F4P7	t No Reference Marketin X Board X 32 STM 32C0 Coming s STM 32C0 Active STM 32C0 Active STM 32C0Active STM 32C0 Active STM 32C0 Active	Package X WLCSP 12 1 6 WLCSP 12 1 6 WLCSP 12 1 6 TSSOP-20 TSSOP-20 6 TSSOP-20 6 TSSOP-20 6	RAMVOi kBytes18i kBytes18i kBytes18	× Flash × 32 kBytes 32 kBytes 16 kBytes 16 kBytes 16 kBytes	48 MHz 48 MHz 48 MHz 48 MHz 48 MHz 48 MHz 48 MHz
Package Core Coprocessor MEMORY	> > > >	• ☆ ☆ ☆ ☆ ☆ ☆ ☆ ☆ ☆	Commercial Pa., C Part STM32C011D6Y., STM32C011D6Y., STM32C011F4P3 STM32C011F4P6 STM32C011F4P7 STM32C011F4., STM3	No Reference MarketinX Board X STM32COComing s STM32COActive STM32COSCM3CO	Package X WLCSP 12 1 6 X WLCSP 12 1 6 X TSSOP-20 6 TSSOP-20 6 TSSOP-20 6 TSSOP-20 6 TSSOP-20 6 TSSOP-20 6	RAM I/O kBytes 18	 Flash 32 kBytes 32 kBytes 16 kBytes 16 kBytes 16 kBytes 16 kBytes 16 kBytes 	Frequency > 48 MHz
Package Core Coprocessor MEMORY Flash From 0 to 4096 (kBytes)	> > > ~	• ☆ ☆ ☆ ☆ ☆ ☆ ☆	Commercial Pa Part STM32C011D6Y STM32C011D6 STM32C011F4P3 STM32C011F4P6 STM32C011F4P6 STM32C011F4P7	t No Reference Marketin X Board X 32 STM 32C0 Coming s STM 32C0 Active STM 32C0 Active STM 32C0Active STM 32C0 Active STM 32C0 Active	Package X WLCSP 12 1 6 X WLCSP 12 1 6 X TSSOP-20 6 TSSOP-20 6 TSSOP-20 6 TSSOP-20 6	RAM VO k Bytes 18 k Bytes 18	× Flash × 32 kBytes 32 kBytes 16 kBytes 16 kBytes 16 kBytes	Frequency > 48 MHz 48 MHz

Figure 40. New Project window - MCU selector

Export to Excel

By clicking on the discrete Export icon, the user can save the MCU table information to an Excel file.

Show favorite MCUs

Clicking the 🕎 icon for an MCU from the list marks it as favorite, see *Figure 41*.

w Project CU/MPU Selector Board Selector Example Select	or Cross Selector	1							
U/MPU Filters									
ovw vorite 🗢 🛨 🔂 🔂 🔾 🕚	Fe	atures Block	Diagram Doc	s & Resources	CAD Resources	Datasheet	📑 Bư	y ⊡→ Sta	rt Project
Commercial WBA V Part Number	*					~			
۹ 🔍 🚽 🗕						STM32			
PRODUCT INFO			5						
Segment >			STM32	N6					
Series >									
Line >									
Marketing Status									
Price									
Package >									
Core >			The first	high-perfo	ormanc	e 🦷			
Coprocessor >			STM32 N	ICU with A	I accel	eration	7/		
MEMORY ~		•							
Flash From 512 to 2048 (kBytes)							-	ort to get an excel	
512 2048	MCUs/MPUs L	ist: 85 items to groupe favorite items at th	he top					aining the MCUs list	A Export
EEPROM = 0 (Bytes)	V	Commercial Part No		ference Marketing State	us X Board ⇒	X Package X RAM	× vo	X Flash X	Frequency
	save the	STM32WBA65RIV6	STM32WBA6 STM32		NUCLE_STM32	VFQFPN 68 8x8x1.0 mm 512 kBytes	46		100 MHz
Click the star to		STM32WBA65RIV7	STM32WBA6 STM32	WBA65R Active		VFQFPN 68 8x8x1.0 mm 512 kBytes	46		100 MHz
Click the star to MCU as favorite		STM32WBA65RGV6	STM32WBA6 STM32	WBA65R Active		VFQFPN 68 8x8x1.0 mm 256 kBytes	46	1024 kBytes 1	
	*	OT IN DE TRUNCOT OF TO							100 M Hz
MCII as favorite	*	STM32WBA65RGV7	STM32WBA6 STM32	WBA65R Active		VFQFPN 68 8x8x1.0 mm 256 kBytes	46	1024 kBytes 1	100 M Hz 100 M Hz
RAM Total From 64 to 512 (kBytes)	*		STM32WBA6 STM32 STM32WBA6 STM32			VFQFPN 68 8x8x1.0 mm 256 kBytes UFBGA 121 6x6x0.6 P 512 kBytes		,	

Figure 41. Marking an MCU as favorite



4.2.2 Board selector

The **Board selector** enables filtering on STM32 board types, series and peripherals (see *Figure 42*). Only the default board configuration is proposed. Alternative board configurations obtained by reconfiguring jumpers or by using solder bridges are not supported.

When a board is selected, the **Pinout** view is initialized with the relevant MCU part number along with the pin assignments for the LCD, buttons, communication interfaces, LEDs, and other functions. Optionally, the user can choose to initialize it with the default peripheral modes.

When a board configuration is selected, the signals change to "pinned", that is, they cannot be moved automatically by STM32CubeMX constraint solver (an user action on the peripheral tree, such as the selection of a peripheral mode, does not move the signals). This ensures that the user configuration remains compatible with the board.



Figure 42. New Project window - Board selector

4.2.3 Example selector

The Example selector allows the user to browse a large set of examples and to start a new project from a selected example.

Note: An example is always related to a specific board, and, consequently, for the MCU available with that board.

Thanks to the filter panel it is possible to filter down the example list for a specific board type, series, peripheral or middleware as well as other characteristics (see *Figure 43*).

	L _	Features				🕞 Start Proj	ect	
Name WD ~						_		
Keyword V	🗋 🕁 🕻	Projects/NUCLEO-G070RB/Examples/IWDG/IWDG_F	Reset/					
					IWDG	_Reset		
Vendor >		STM32G0			V1.3	0		
Board V			_		• 1.0	.0		
board		FW pack STM32Cube_FW_G0_V1.3.0	Board	NUCLEO-G070RB		Mounted device		
Name		CubeMX 5.2.0				STM32G070RBTx		
Turne		Toolchain/IDE		1220		LOEP64		
Type ~		EWARM, MDK-ARM, SW4STM32		and the second second				
Check/Uncheck All		Keywords						
Discovery Kit								
Evaluation Board								
Evaluation Board Nucleo-144	Examp	ples List: 7 items						📤 Exp
		Name 🗘	Board	× Board Type	× Series	× Project Type		× CubeMX ver
 ☐ Nucleo-144 ☐ Nucleo-32	÷	Name A IWDG_RefreshUntilUserEvent	NUCLEO-F411RE	Nucleo-64	STM32F4	Example	LL	X CubeMX ver
□ Nucleo-144	· ☆ ☆	Name * IWDG_RefreshUntilUserEvent IWDG_RefreshUntilUserEvent_Init	NUCLEO-F411RE NUCLEO-G070RB	Nucleo-64 Nucleo-64	STM32F4 STM32G0	Example Example	LL LL	X CubeMX ven NA 5.2.0
 □ Nucleo-144 □ Nucleo-32		Name Mume IWDG_RefreshUntilUserEvent IWDG_RefreshUntilUserEvent_Init IWDG_RefreshUntilUserEvent_Init	NUCLEO-F411RE NUCLEO-G070RB NUCLEO-G071RB	Nucleo-64 Nucleo-64 Nucleo-64	STM32F4 STM32G0 STM32G0	Example Example Example	LL LL	X CubeMX ver NA 5.2.0 5.2.0
 □ Nucleo-144 □ Nucleo-32	· ☆ ☆ ☆	Name ** IWDG_RefreshUntilUserEvent IWDG_RefreshUntilUserEvent_Init IWDG_RefreshUntilUserEvent_Init IWDG_Reset	NUCLEO-F411RE NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G070RB	Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64	STM32F4 STM32G0 STM32G0 STM32G0	Example Example Example Example	LL LL HAL	X CubeMX ver NA 5.2.0 5.2.0 5.2.0 5.2.0
☐ Nucleo-144 ☐ Nucleo-32 ✔ Nucleo-64		Name * WDG_RefreshUntilUserEvent_Init WDG_RefreshUntilUserEvent_Init WDG_Reset WDG_Reset WDG_Reset	NUCLEO-F411RE NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G070RB NUCLEO-G071RB	Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64	STM32F4 STM32G0 STM32G0 STM32G0 STM32G0	Example Example Example Example Example	LL LL HAL HAL	X CubeMX ver NA 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0
☐ Nucleo-144 ☐ Nucleo-32 ✔ Nucleo-64	☆ ☆ ☆ ☆ ☆ ☆	Name * WDG_RefreshUntilUserEvent_ WDG_RefreshUntilUserEvent_init WDG_Reset WDG_Reset WDG_WindowMode	NUCLEO-F411RE NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G071RB NUCLEO-G071RB NUCLEO-G070RB	Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64	STM32F4 STM32G0 STM32G0 STM32G0 STM32G0 STM32G0	Example Example Example Example Example Example	LL LL HAL	CubeMX ver NA 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0
□ Nucleo-144 □ Nucleo-32 ☑ Nucleo-64 MCU / MPU ~ Name ~		Name * WDG_RefreshUntilUserEvent_ WDG_RefreshUntilUserEvent_init WDG_Reset WDG_Reset WDG_WindowMode	NUCLEO-F411RE NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G070RB NUCLEO-G071RB	Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64	STM32F4 STM32G0 STM32G0 STM32G0 STM32G0	Example Example Example Example Example	LL LL HAL HAL HAL	X CubeMX ver NA 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0
□ Nucleo-144 □ Nucleo-32 ☑ Nucleo-64 MCU / MPU ~	☆ ☆ ☆ ☆ ☆ ☆	Name * WDG_RefreshUntilUserEvent_ WDG_RefreshUntilUserEvent_init WDG_Reset WDG_Reset WDG_WindowMode	NUCLEO-F411RE NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G071RB NUCLEO-G071RB NUCLEO-G070RB	Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64	STM32F4 STM32G0 STM32G0 STM32G0 STM32G0 STM32G0	Example Example Example Example Example Example Example	LL LL HAL HAL HAL HAL	CubeMX ver NA 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0
□ Nucleo-144 □ Nucleo-32 ☑ Nucleo-64 MCU / MPU ~ Name ~	☆ ☆ ☆ ☆ ☆ ☆	Name * WDG_RefreshUntilUserEvent_ WDG_RefreshUntilUserEvent_init WDG_Reset WDG_Reset WDG_WindowMode	NUCLEO-F411RE NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G071RB NUCLEO-G071RB NUCLEO-G070RB	Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64	STM32F4 STM32G0 STM32G0 STM32G0 STM32G0 STM32G0	Example Example Example Example Example Example Example	LL LL HAL HAL HAL HAL HAL	NA 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0
Nucleo-144 Nucleo-144 Nucleo-32 Nucleo-64 MCU / MPU	☆ ☆ ☆ ☆ ☆ ☆	Name * WDG_RefreshUntilUserEvent_ WDG_RefreshUntilUserEvent_init WDG_Reset WDG_Reset WDG_WindowMode	NUCLEO-F411RE NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G071RB NUCLEO-G071RB NUCLEO-G070RB	Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64	STM32F4 STM32G0 STM32G0 STM32G0 STM32G0 STM32G0	Example Example Example Example Example Example Example A version can be l	LL LL HAL HAL HAL HAL On is set wi	X CubeMX ver NA 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0
Nucleo-144 Nucleo-144 Nucleo-32 Nucleo-54 MCU / MPU V Name Check/Uncheck All Strid32F4	☆ ☆ ☆ ☆ ☆ ☆	Name * WDG_RefreshUntilUserEvent_ WDG_RefreshUntilUserEvent_init WDG_Reset WDG_Reset WDG_WindowMode	NUCLEO-F411RE NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G071RB NUCLEO-G071RB NUCLEO-G070RB	Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64	STM32F4 STM32G0 STM32G0 STM32G0 STM32G0 STM32G0	Example Example Example Example Example Example Example A version can be l If not ave	LL LL HAL HAL HAL HAL On is set willoaded in S vailable, th	X CubeMX ver NA 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5
Nucleo-144 Nucleo-32 Nucleo-54 MCU / MPU	☆ ☆ ☆ ☆ ☆ ☆	Name * WDG_RefreshUntilUserEvent_ WDG_RefreshUntilUserEvent_init WDG_Reset WDG_Reset WDG_WindowMode	NUCLEO-F411RE NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G071RB NUCLEO-G071RB NUCLEO-G070RB	Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64	STM32F4 STM32G0 STM32G0 STM32G0 STM32G0 STM32G0	Example Example Example Example Example Example Example A version can be 1 If not av solely b	LL LL HAL HAL HAL HAL On is set willoaded in S vailable, th	X CubeMX ver NA 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0 5.00 5.0

Figure 43. New project window - Example selector

Selecting an example and clicking "Start project" allows STM32CubeMX to copy the example as a new project (the user can change the default location at this stage).

Warning:	For some examples the "Start Project" button is shown with
	an "Under Development" warning icon. Projects created from
	these examples may be not functional (they do not compile).
	Fixes are in development.

Several options are available to open the newly created project (see *Figure 44*):

- with STM32CubeMX (available only for examples listed with an STM32CubeMX version set)
- with a File explorer
- with one of the supported toolchains (provided the toolchain is already installed on your computer)



MX Sta	rt Project from Example		×
0	Status:	Active	
	Name:	IWDG_Reset	
	Board:	NUCLEO-G070RB	
	Required Software Package:	STM32Cube_FW_G0_V1.6.2	
	Install Project Directory:	C:/Users/bekrisli/STM32Cube/Example	िर
	Open with:	STM32CubeMX ~ STM32CubeMX EWARM MDK-ARM STM32CubeIDE	
		Explorer	

Figure 44. Popup window - Starting a project from an example

Note:

If the STM32Cube MCU package needed for the example is missing from the repository, STM32CubeMX automatically starts the download process.

4.2.4 Cross selector

Part number selection

The Cross selector allows users to find the products that best replace the MCU or MPU they are currently using (from ST or other silicon vendors).

To access this functionality, STM32CubeMX data must be up to date. This is ensured using Refresh Data from the Help menu (see *Figure 45*).

STM32CubeM	X Untitled: STM32WBA65Plbx							- 0 ×
STM32	File	Window	Help				🐵 📑 🕒 X 🔿 C	XX 57
	STM32WBA65PIIx VIntitled - Pinout & C	onfiguration >	Help About	F1 At A			GENERATE CODE	
	Pinout & Configuration	Clock Cor	Docs & Resources	At-D At-V	Project Manag	jer 🛛	Tools	
Q		✓ Softw	Refresh Data	Al-R	Dinaut view	System view		
	A->Z	• •	User Pr Refresh data and docu Check for Updates	ments for MCUs, Box At-C	ards and Examples	System view		8
System Co	re)		Manage embedded software Connection & Updates	e packages At-U At-S			_	
Analog			(PD	36363	P02 (V0011) (V00)			
Timers			5000	SM. (PB13) (PB14)		DOHPA) (DEC.D.) (VOLANA) (VESHF) (M	RST	
Connectivit	y :		(hos					
Multimedia								
Security			PB	MI PAB VSS	PB12 PA7 PA10	ATAUT PBIE PATRIT PBZ F	105 J	
Computing			(ver	UT+ VSSA PAG	POD PAG PED (
Middleware	e and Software Packs		(voi		PDI PDI PED (PET PC12 PATT PEO P	91 D	
Trace and	Debug		(n)					
Power and	Thermal							
Utilities								
				89 (VD0) (VSS)		POD (PES) (PER(PER)	ss)	
			500	USB POS PBS		Post Post (00	
			PO			PGT PG4 PC12 PD5 F	195	
			<u></u>	at hand hand	Sand Sand Sand			
				U	FBGA121_SMPS_US	B (Top view)		
			<u>ର</u> 🖸 ବ	b 4		Q~	Unused GPIOs: 86/86	

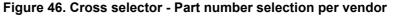
Figure 45. Cross selector - Data refresh prerequisite

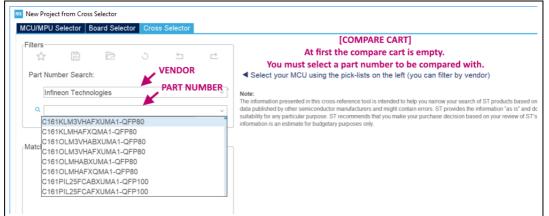
UM1718 Rev 47

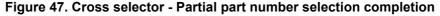


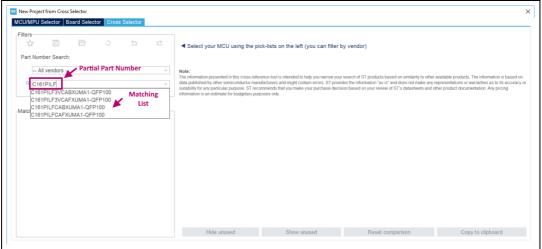
Clicking "ACCESS TO CROSS SELECTOR" under the "Start my project from Cross Selector" section of the main page opens the New Project window on the Cross selector tab.

Two drop downs menus allow the user to select the vendor and the part number of the product to be compared to (see *Figure 46*). A part number can also be entered partially: STM32CubeMX proposes a list of matching products (see *Figure 47*).









Compare cart

Once a part number is selected, a list of matching ST part number candidates is displayed along with their matching ratio in the Matching ST candidates panel.

By default, the three closest matches are selected and added to the compare cart along with the part number to be compared to (see *Figure 48*).



ICU/MPU Selector Board Selector Exa	mple Selector Cross	Selector	Re	isult of Mate	ning a part num	nber with other STMicroe	lectronics products			
Filters		Compa	ring STI	M32C01	1D6Yx with	other STMicroel	ectronics solutio	ons e products that are havin	g the higher percent of m	tching with the cur
☆ 🔂 🗟 こ	5 ⊂	Used 1	Importan	Category	Parametric	STM32C011D6Yx	STM32L4S9ZIYx C+	STM32L4S9ZITx C+	STM32L4S9ZIJx	
Part Number Search:				-	Û	Comparing creteria	-			1
STMicroelectronics	~	\bigcirc		Product	Public Price	no	7.083 USD (for 10K)	7.700 USD (for 10K)	8.318 USD (for 10K)	
Q STM32C011D6Yx-WLCSP 12 1.7x1.	42x0.6 P 0.35 mm ~		-=0	System Core	busArch	32 bit	32 bit	32 bit	32 bit	
			_00	System Core		ARM Cortex-M0+ at 48 MHz	ARM Cortex-M4 at 120 MHz	ARM Cortex-M4 at 120 MHz	ARM Cortex-M4 at 120 MHz	
Matching ST candidates (500)			he ratio of i	matching ch System Core	anges with impo package	rtance level_creteria WLCSP 12 1.7x1.42x0.6 P	CSP144	QFP144	BGA144	
Part number	Match	\sim				0.35 mm0				
STM32L4S9ZIYx	100 %			System Core	000	10 io	110 io	110 io	112 io	
STM 32L4S9ZITX	100 %		-00					11010	11210	
STM32L4S9ZIJx	100 %	-				⇒0:Nice to have/1:Sho				
STM32L4S9VITx	100 %			System Core	Temperature range	-40 °C to 85 °C	-40 °C to 85 °C	-40 °C to 85 °C	-40 °C to 85 °C	
STM32L4S9Allx	100 %									
STM32L4S7ZITx	100 %		_00	System Core	Voltage range	2.00 V to 3.60 V	1.71 V to 3.60 V	1.71 V to 3.60 V	1.71 V to 3.60 V	
STM32L4S7VITx	100 %		-00							
STM32L4S7Allx	100 %			Producer Co.	0.000	81/0	640 KB	540 V P		
STM32L4S5ZIYx	100 %		_=0	System Core	rowd	6 KB	640 KB	640 KB	640 KB	
STM32L4S5ZITx	100 %									
STM32L4S5VITx	100 %	\bigcirc		System Core	eeprom	no	no	no	no	
STM32L4S5QIIx	100 %	\bigcirc								
STM32L4S5Allx	100 %							2048 KB Matching		
STM32L4R9ZIYxP	100 %		_=0	System Core	tiash	32 KB	2048 KB		2048 KB	
STM32L4R9ZIYx	100 %							value for		
STM32L4R9ZITx	100 %	\frown		System Core	Touch Sensing	no	yes	yes each	yes	
STM32L4R9ZIJx	100 %	\bigcirc					· · · · · · · · · · · · · · · · · · ·	creteria		
STM32L4R9ZGYx	100 %	-								
STM32L4R9ZGTx	100 %			Analog	ADC	13xADC 12-bit	16xADC 12-bit	16xADC 12-bi	16xADC 12-bit	
STM32L4R9ZGJx	100 %						4xADC 24-bit	4xADC 24-bit	4xADC 24-bit	
STM32L4R9VITx	100 %	_								
STM32L4R9VGTx	100 %	\odot	- T	Analog	Comparator (COMP)		Z	Z Tarda ta tanza Itala da a 11	2	
STM32L4R9AIIx	100 %		⊫>⊵	comparing v	vnen the feature	is not present for the sel	ected MCU or when the	into is invalide for the m	cu to compare with .	
STM32L4R9AGIx	100 %	\bigcirc		Analog	OPAMP	no	2	2	2	
STM32L4R7ZITx	100 %									
STM32L4R7VITx	100 %			-	-					
STM32L4R7AIIx	100 %			Timers	Timer	5xTimer 16-bit	9xTimer 16-bit	9xTimer 16-bit	9xTimer 16-bit	
STM32L4R5ZIYx	100 %						2xTimer 32-bit	2xTimer 32-bit	2xTimer 32-bit	
STM32L4R5ZITxP	100 %									
STM32L4R5ZITx	100 %	\bigcirc		Timers	Timer (HRTIM)	no	no	no	no	
STM32L4R5ZGYx	100 %									
STM32L4R5ZGTx	100 %	\bigcirc		Timers	Timer (LPTIM)	no	yes	yes	yes	
STM32L4R5VITx	100 %	\cup								
STM32L4R5VGTx	100 %								Copy and past	the compare card
STM32L4R5QIIxP	100 %								caspy and past	L
STM32L4R5QIIX	100 %									/
RTM 32L4N JOUR	100 %		Hide u	nused		Show unused		Reset comparison		Copy to clipboard

Figure 48. Cross selector - Compare cart

This selection can be changed anytime in the Matching ST candidates panel.

The comparison can be customized: the features to be used for comparison can be unselected when considered as irrelevant and their level of importance can be adjusted. These choices affect the computed matching ratio.

The comparison is disabled for features that are not supported on the part number to be compared with, or when the feature information is unavailable.

Buttons are available to manipulate and save a copy of the compare cart view:

- to hide criteria not used for the comparison, or show all of them
- to come back to default STM32CubeMX comparison settings
- to copy and paste the current cart view in a document or email.

MCU/MPU selection for a new project

Clicking an STM32 part number from the compare cart selects it in the MCU/MPU Selector tab, and clicking on C+ Start Project creates a new project for that part number (see Figure 49).



New Project from Cross Selector ICU/MPU Selector Board Selector							Click to create a n with this part (
Filters		Comparing D	17618AB	GN100V by I	Renesas Electro	nics with STMicroele			
		Used 2 1	mportan	. Category	Parametric	D17618ABGN100	STM32H750IBKx [-+	STM32F745IEKx E+	STM32F745IGKx 🕞
Part Number Search:			· · · ·						
Renesas Electronics	~		-00	Product	Public Price	No info	4.651 USD (for 1DK)	6.991 USD (for 10K)	7.916 USD (for 10K)
Q D17618ABGN100V-BGA176	~		-=0	System Core	busArch	32 bit	32 bit	32 bit	32 bit
			-00	System Core	core	SH-2 at 100 MHz	ARM Cortex-M7 at 400 MHz	ARM Cortex-M7 at 216 MHz	ARM Cortex-M7 at 216 MHz
Matching ST candidates (500) Part number	Match		-=0	System Core	package	BGA176	BGA176	BGA176	BGA176
TM32H750IBKx TM32F207ICHx TM32F207IEHx	95 % 94 %		-00	System Core	GPIO	78 io	138 io	140 io	140 io
	864 400000000000000000000000000000000000			System Core	Temperature range	-20 °C to 75 °C	-40 °C to 105 °C	-40 °C to 105 °C	-40 °C to 85 °C
STM32E207/GHX STM32E745/EKX STM32E407/GHX	New Project from Cross Sele		C						
STM32F746IEKx SIM32E21ZIGH×			iloss Sele	2101					X
STM32E427/GHX STM32E417/GHX STM32E429/GHX	MCU/MPU Filters	Ea 3		Fe	atures Bl	ock Diagram D	ocs & Resources	atasheet 🛛 🖾 Buy	E+ Start Project
TM32F437IGHX TM32F745IGKx STM32F469IGHx	Part Number Search		~	_		-			
5TM32E439IGHx 5TM32E746IGKx 5TM32E765IGKx 5TM32E765IGKx	Q STM32H750IE	3 ~		☆ [STM3	2H750IB				
STM32F765)GKX STM32F4271Hx STM32F4279GHx STM32F756GKX STM32F7670GKX STM33F7670GKX	Core		~	MCUs/MP	'Us List: 1 item		+ Display similar ite	ms	đi
51M32F429IIHx 5TM32F437IIHx 5TM32F460IIHx	Check/Uncheck All				Part No * Refer		O GFX S CORDIC DDR DEB		A PWR RF SHA TAMF
STM32E43911Hx STM32E76511Kx STM32E76711Kx	ARM Cortex-M7			✿ ST	M32H7 STM32I	H750 12 10 138	3 1960 0 0 1	0 0 0 0	0 0 0 0
8+M35E57511DC	Series								

Figure 49. Cross selector - Part number selection for a new project

Clicking the Cross Selector Tab allows the user to go back to the cart and change the current selection for another part number.

4.3 **Project page**

Once an STM32 part number or a board has been selected or a previously saved project has been loaded, the project page opens, showing the following set of views (refer to dedicated sections for their detailed description):

- Pinout & Configuration
- Clock Configuration
- Project Manager
- Tools

Users can move across the different views without impacting their project configuration.

A **GENERATE CODE** button is always accessible for the user to click and allows to generate the code corresponding to the current project configuration. Moreover, thanks to convenient navigation breadcrumbs (see *Figure 50*), the user can detect what its current location is in STM32CubeMX user interface, and can move to other locations:

- to the home page by clicking the Home breadcrumb
- to the new project window by clicking the part number
- back to the project page by clicking the project name (or Untitled if the project does not have a name yet).



U	50. STM32CubeMX M			••
STM32CubeMX Untitled: STM32F439VIT	x			
STM32 Bread crumbs File	Window	Help	🥸 🕇	🖻 🎽 🔆 🌆
Home / STM32F439VITx	/ Untitled - Pinout & Configu	uration	GENERATE	CODE
Pinout & Configuration	Clock Configuration	Project Man	ager	Tools
		Pinout		
Cptions Q Stategories A->Z	-	Pinout view III System	riew	t i
System Core >		980 986 987 986 986 986 986 986 986 986 986 986 986	POO POOL	
Analog >	PE3 PE4 PE4		VSS VCA.	
Timers >	986 (98A		PA12 PA12 PA11	
Connectivity >	PCI. PCI.		PA0 PA0 PA8	
Multimedia >	100 100		PCA PC7 PC7	
Security >	222 233 245 245 245 245 245 245 245 245 245 245		203 203 203 203 203 203 203 203 203 203	
Computing >	PC2 PC3	STM32F439VITx LQFP100	PD12 PD11 PD11	
Middleware >	V55. V98.	Larriou	P00 P08 P815 P814	
Application >	240. 241		P814 P813 P812	
	AAA 2015	PAG PAF PAF PAF PCA PCA PCA PCA PCA PCA PCA PCA PCA PCA	PERS PERSON	
				~
MCUs Selection Output				
Series	Lines	Мси	Package	Required Peripherals
STM32F4 STM32F42 STM32F4 STM32F42				one 🗧
STM32F4 STM32F4 STM32F4 STM32F4				

Figure 50. STM32CubeMX Main window upon MCU selection



Selecting a board, then answering **No** in the dialog window requesting to initialize all peripherals to their default mode, automatically sets the pinout for this board. However, only the pins set as GPIOs are marked as configured, i.e. highlighted in green, while no peripheral mode is set. The user can then manually select from the peripheral tree the peripheral modes required for its application (see *Figure 51*).

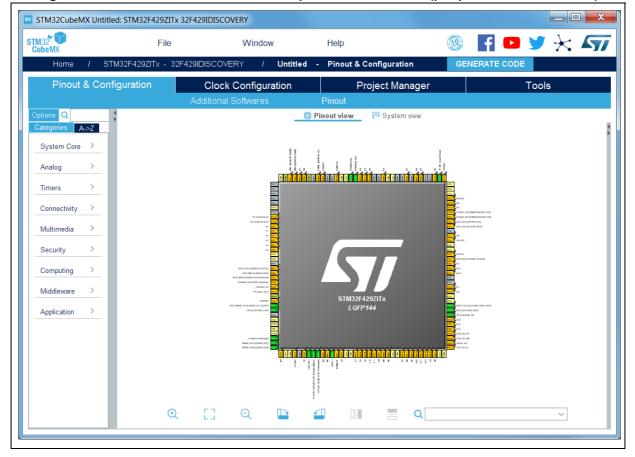


Figure 51. STM32CubeMX Main window upon board selection (peripherals not initialized)



Selecting a board and accepting to initialize all peripherals to their default mode automatically sets both the pinout and the default modes for the peripherals available on the board. This means that STM32CubeMX generates the C initialization code for all the peripherals available on the board and not only for those relevant to the user application (see *Figure 52*).

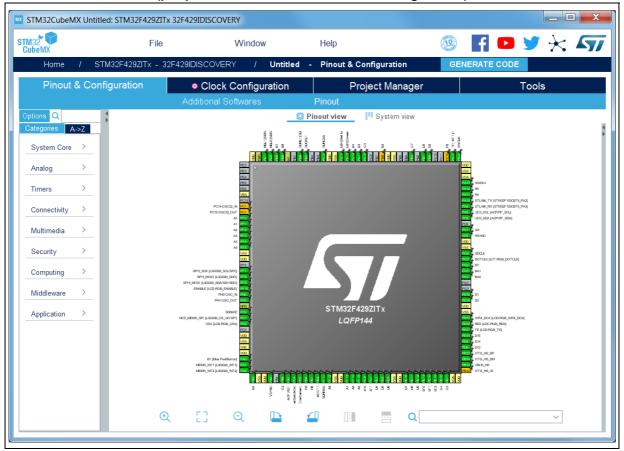


Figure 52. STM32CubeMX Main window upon board selection (peripherals initialized with default configuration)

4.4 Boot chain (STM32 MPUs)

4.4.1 Boot mode configuration

ST embedded software can support complex architectures (such as OpenSTLinux), which require a complex boot chain, involving several processors, firmware, and a complex boot sequence. An overview is given in the STM32MPU Wiki portal.

The boot mode defines the processor that starts the software, defines the boot sequence scheme, and which software services can be started (such as secure services, also known as TrustZone[®]).



Creating a project for a dual core (Cortex-A35 and Cortex-M33) MPU

The first example uses the following boot mode: Cortex-A35 is the master processor, Cortex-M33 is the secondary one, in non-secure mode.

The master always runs in a secure mode.

- Select an STM32MP257x MPU
- Select the option "with A35 Master without Cortex M33 TrustZone activated?" on the popup window (see *Figure 53*)

Recent Opened Projects Intect to : PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 20/05/2024 22:28:37 PL0_Nucleo_NUCLEO-WL5 PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 20/05/2024 22:28:37 Image: Check for STM32CubeMX and PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 20/05/2024 22:28:37 Image: Check for STM32CubeMX and PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 22/05/2024 18:22:12 Image: Check for STM32CubeMX and PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 22/05/2024 18:22:12 Access to BoARD SELECTOR PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 22/05/2024 18:22:147 Start My project from Example PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 22/05/2024 18:22:147 Access to Example selector PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 20/05/2024 22:27:53 Image: Check for Example	Existing Projects	New Project	Manage software installations
PLO_Nucleo_NUCLEO-WL5 MX Last modified date : 20/05/2024 22:28:37 PLO_Nucleo_NUCLEO-WL5 MX Last modified date : 20/05/2024 22:28:37 PLO_Nucleo_NUCLEO-WL5 MX Last modified date : 22/05/2024 18:22:12 PLO_Nucleo_NUCLEO-WL5 MX	Recent Opened Projects	I need to :	Check for STM32CubeMX and
PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 20/05/2024 22:28:37 O A35 Master with Cortex M33 TrustZone activated ? PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 22/05/2024 18:22:12 ACCESS TO BOARD SELECTOR PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 22/05/2024 18:21:47 Start My project from Example PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 22/05/2024 18:21:47 Start My project from Example PL0_Nucleo_NUCLEO-WL5 MX			CHECK FOR UPDATES
Last modified date : 22/05/2024 18:22:12 ACCESS TO BOARD SELECTOR PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 22/05/2024 18:21:47 Start My project from Example PL0_Nucleo_NUCLEO-WL5 MX			
Last modified date : 22/05/2024 18:21:47 PL0_Nucleo_NUCLEO-WL5 MX CESS TO EXAMPLE SELECTOR PL0_Nucleo_NUCLEO-WL5 MX CESS TO EXAMPLE SELECTOR		OK ACCESS TO BOARD SELECTOR	
PL0_Nucleo_NUCLEO-WL5 MX			
			New 402 MB: howflow: MCV with measurements and security

Figure 53. Project choice interface

• Six contexts are created in the configuration panel (see *Figure 54*)

Figure	54.	Contexts
--------	-----	----------

oot time:		Runtime context	ts:		
A35 ROM	A35S (TF-A_BL2)	A35S (OP-TEE)	A35NS (U-Boot)	A35NS (Linux)	M33NS (Cube)
			~		

- The Cortex-A35 runs under the OpenSTLinux operating system. It uses the following firmware:
 - TF-A BL2
 - OP-TEE
 - U-Boot
 - Linux
- The Cortex-M33 is configured using Cube firmware: M33NS Cube FW (HAL & LL)



				Г	iyun	- 55.	IPS Interta	100 455	iyiiiie			
Home > STI	M32MP2	51FAlx	> Un	titled - P	inout & (Configura	tion >					
P	inout 8	Config	juratior	ı			Clock Config	uration			RIF	
									 Software 	Packs		
Q				~		٢			ADC1 Mode	and Configuratio	n	
Categories A-	>Z									Mode		
		A35S	A35S	A35NS	A35NS	M33NS	Boot time:		Runtime cont	exts:		
÷	A35 R		(OP-TEE) (U-Boot)	(Linux)	(Cube)	A35 ROM	A35S	A35S	A35NS	A35NS	M33NS
ADC1				 ✓ 				(TF-A_BL2)	(OP-TEE)	(U-Boot)	(Linux)	(Cube)
ADC2				~	V					✓	✓	
ADC3							INO Single-e	nded				
ADF1							IN1 IN1 Different	ial				~
BOOT							IN2 Disable					~
* BSEC	~	\$	\checkmark									
CORTEX_M33			\checkmark		~		IN3 Disable					~
							IN4 Disable					\sim
CRYP1		~			~		IN5 Disable					~
CRYP2							IN6 Single-er	had				
CSI					~							
DCACHE						~	IN7 Single-er	nded				
DCMI					V				Cor	figuration		
DCMIPP					~		Reset Configura	tion				
DDR_CTRL		1	\checkmark				Reset Conligura	tion				
DEBUG							Parameter Sett	ings 🛛 📀 GIC	Settings 🛛 😔	DMA Settings	🥝 GPIO Setting	js -
DSIHOST				~	1							
DTS			~				Search Signals					
ETH1					×		Search (Ctrl+F)]				Show only M
FILEX									00 11 1			
							Pi Signal Pir ANA0 ADC1 A3	and the second se	PIO Maxi /a n/a	Retime Invert n/a n/a	Doubl Delay n/a n/a	. Delay Invert Us∢ n/a n/a
GIC			~		1			5N Analo n 5N Analo n		n/a n/a n/a n/a	n/a n/a n/a n/a	n/a n/a n/a n/a
GPIO							ANAT ADOT AS	ana Anaio n	a ///a	iva iva	iva iva	iva iva
HASH	~	~	~									
HDP					V							
HPDMA1		1	8	1	1	1						
HPDMA2		1	1	~	1	~						
HPDMA3		1	1	~	1	~						

Figure 55. IPs interface assignment

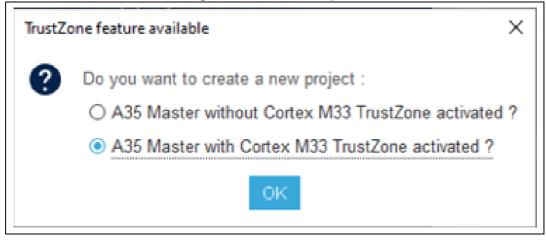
After assigning the IPs context go to "Project Manager" view, save the project, and generate the code.

The second example uses the following boot mode: Cortex-A35 is the master processor, Cortex-M33 core is the secondary one, in secure mode.

The master always runs in a secure mode.

- Select an STM32MP257x MPU
- Select the option "with A35 Master with Cortex M33 TrustZone activated?" on the popup window (see *Figure 56*)

Figure 56. TrustZone option



UM1718 Rev 47



• Six contexts created in the configuration panel (see *Figure 57*)

Boot time:		Runtime contexts:									
A35 ROM	A35S (TF-A_BL2)	A35S (OP-TEE)	A35NS (U-Boot)	A35NS (Linux)	M33S (TF-M)	M33NS (Cube)					
				V							

Figure 57. Selected context

Cortex-A35 runs under the OpenSTLinux operating system. It uses the following firmware:

- TF-A BL2
- OP-TEE
- U-Boot
- Linux

Cortex-M33 secure is configured using Cube firmware: TF-M

To assign IPs context go to "Pinout & Configuration" and configure IPs.

Figure 58. Assign IP context

ome 🗡	STM32MP251	FALx 🔰 Untitled - Pinou	t & Configu	ration >														GE	NERATE CO	DE
	Pinout & C	Configuration		Cloc	ck Confi	guration	1			RIF			1	Projec	t Manager				Тос	ols
							✓ So	ftware Packs		✓ Pinout										
	~						0			ADC1	Mode and Co	nfiguration				1	_	🖸 Pinout v	iew 💾 S	iystem view
legories	A->Z										Mode									
÷	A35 ROM	Cortex-A35 secure loader (TF-A. BL2)	A35S (OP-TEE	A35NS (U-Boot)	A35NS (Linux)	M33S (TF-M)	M33NS (Cube)	Boot time:	A35S	Runtime conter A35S	ds: A35NS	A35NS	M3	35	M33NS					
DC1								A35 ROM	(TF-A_BL2)	(OP-TEE)	(U-Boot)	(Linux)			(Cube)					
DC2													0]						
								INO Single-	haded											
								IN1 IN1 Differen	cai						~					
SEC	1	2	1			12		IN2 Disable							~					
ORTEX					1			IN3 Disable							~		ăăăă			
								IN4 Disable							~		êêee			
RYP1		2						IN5 Disable							~		0000			ÖÖÖÖ
								IN6 Single-e							÷		0000			
SI					V															
CACHE								IN7 Single-e	nded								(D) (D) (D) (D)			
CMI					V			IN8 Single-e	nded								0000			
CMIPP											Configuratio	n								
DR_CTR.		2	1																	
								Reset Configura									ěeee			0000
				1	1			Parameter Set	tings 🛛 😔 GII	C Settings 🛛 🥥	DMA Settings	GPI0	Settings							0000
TS			.																	
TH1					2			Search Signals									8000			
								Search (Ctrl+F)						Show	w only Modified Pi	ins	terr R. R.	VERG	A361 (Top v	(wai)
								D. A Sout D	- 0 0000		Outine Inco	a Deviat	Datas IData		. Here Mort			VFBG	noon (10p v	iem)
GIC			1		1			ANA0 ADC1 A	ISNS Analo	n/a n/a	n/a n/a	n/a r	Va n/a	y. n/a	t User Mod					
PIO								ANA1 ADC1 A			n/a n/a		va n/a	n/a						
ASH	1	2	S								10.0									
DP					2															
PDMA1		5	1	1																
PDMA2		2	2		2	2														
PDMA3		2	1	1	1	1											Q []	. Θ	12	<u><1</u>
																			_	_

After assigning the IPs context go to "Project Manager" view, save the project, and then generate code.

4.4.2 Coprocessor initializers (STM32MP2x)

The STM32MP2xx comes with two possible coprocessors (Cortex-M33 or Cortex-M0+). STM32CubeMX manages only Cortex-M33.

The STM32CubeMX tool indicates which programs running on the main processor can be started, or if to use the secondary processor.

When the system source code is generated, the settings that determine how the main processor can use the coprocessor are included in the device tree. These settings are found in the "rproc" sections (nodes) for each software component that can interact with the coprocessor. This ensures that, when the system is running, it knows how to handle the coprocessor according to the predefined configuration.



As an example:

• OP-TEE is eligible to load the main processor.

Boot time:		Runtime conte	exts:			
A35 ROM	A35S	A35S	A35NS	A35NS	M33S	M33NS
A35 KOW	(TF-A BL2)	(OP-TEE)	(U-Boot)	(Linux)	(TF-M)	(Cube)
		~		~		

- Linux Kernel is eligible to load for the main processor.
- U-Boot will be available when Linux is selected.

Boot time:		Runtime conte	exts:			
A35 ROM	A35S (TF-A BL2)	A35S (OP-TEE)	A35NS (U-Boot)	A35NS (Linux)	M33S (TF-M)	M33NS (Cube)
		~	V	\checkmark		

Figure 60. U-Boot selection

4.4.3 Boot device selection (STM32MP25)

The term boot device refers to any storage device from which a microcontroller can load the initial software used to boot up the system. This initial software is part of the boot process that starts the computer and loads the operating system.

STM32CubeMX does not handle the configuration of the pins used by STM32 devices to select the boot source. To configure a correct boot, ensure that the boot device settings align with the boot pins configuration, programmed in the MCU hardware. This requires checking the datasheet or reference manual, to understand the boot pin settings, and then manually configuring the system to match those settings.

A boot device must be assigned to the ROM firmware and the early-stage Boot Loader (such as TF-A BL2 for OpenSTLinux).

When configuring a microcontroller, consider the constraints that affect the choice of boot devices, and their dependency upon the selected boot mode. STM32CubeMX checks the boot configuration of against a set of constraints to ensure that the system boots properly. This service is called Flexible Software Loader synchronization verification. The results of this verification are displayed in a dedicated output window (FSBL synchro output), providing developers with important diagnostic information.

The "FSBL synchro output" panel is displayed with the rule "Faulty state detected for SDMMC1: FSBL-A assignments possible only if assigned in BootRom". Users can refer to this panel to align any misconfigurations.



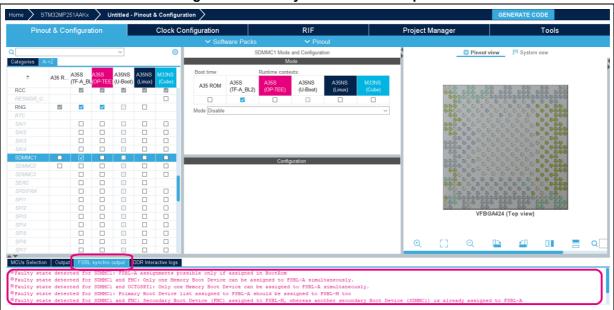


Figure 61. FSBL synchronization output

4.5 Pinout & Configuration view

The **Pinout & Configuration** view comes with the following main panels, function and menu:

- A **Component list** that can be visualized in alphabetical order and per categories. By default, it consists of the list of peripheral and middleware that the selected MCU supports. Selecting a component from that list opens two additional panels (**Mode** and **Configuration**) that allow the user to set its functional mode and configure the initialization parameters that will be included in the generated code.
- A **Pinout view** that shows a graphic representation of the pinout for the selected package (e.g. BGA, QFP) where each pin is represented with its name (e.g. PC4) and its current alternate function assignment, if any.
- A **System view** that gives an overview of all the software configurable components: GPIOs, peripherals, middleware and additional software components. Clickable buttons allow opening the configuration options for the given component (Mode and Configuration panels). The button icon color reflects the status of the configuration status.
- A Software Packs menu with two sub-menus:
 - Select Components to select, for the current project, software components not available by default. This selection updates the Pinout & Configuration view accordingly
 - **Manage Software Packs** to install/uninstall software packs.
- An Additional Software function that allows to select, for the current project, software components that are not available by default. Selecting an additional software component updates the **Pinout & Configuration** view accordingly.
- A **Pinout** menu that allows the user to perform pinout related actions such as clear pinout configuration or export pinout configuration as csv file.



Tips

- You can resize the panels: hovering the mouse over a panel border displays a two-ended arrow: right-click and pull in a direction to extend or reduce the panel.
- You can show/hide the Configuration, Mode, Pinout and System views using the open and close arrows.

4.5.1 Component list

The component list shows all the components available for the project. Selecting a component from the component list, opens the Mode and Configuration panels.

Contextual help

The **Contextual Help** window is displayed when hovering the mouse over a peripheral or a middleware short name.

By default, the window displays the extended name and source of configuration conflicts if any (see *Figure 62*).



Figure 62. Contextual Help window (default)

Clicking the *details and documentation* link (or CTRL+d) provides additional information such as summary and reference documentation links (see *Figure 63*). For a given peripheral, clicking *Datasheet* or *Reference manual* opens the corresponding document, stored in STM32CubeMX repository folder, at the relevant chapter. Since microcontrollers datasheets and reference manuals are downloaded to STM32CubeMX repository only upon user request, a functional Internet connection is required:

- To check your Internet connection, open the **Connection** tab from the **Help > Updater Settings** menu.
- To request the download of reference documentation for the currently selected microcontroller, click **Refresh** from the **Help > Refresh Data** menu window.

Pinout & Configuration	Clock Configuration
	Additional Softwares
Options Q 🗸	IRTIM Mode and Configuration
Categories A->Z	Mode
IRTIM:	
M InfraRed Interface	
N Status:	
OF Not available:	
Channel 1 of TIM16 and TIM17 must be configured i	n one of the availables output modes
Of Summary:	
IRTIM (InfraRed Interface) offers important advantage	es as a form of wireless communication.
RI Nowadays, almost all audio and video equipment ca	
RI At the receiving end, a receiver detects the light puls	ses, which are processed to retrieve/decode
SF the information they contain.	
SF Related documentation:	
SF - <u>Datasheet</u>	
✓ SY - <u>Reference manual</u>	
TINT	
TIM2	

Figure 63. Contextual Help detailed information

Icons and color schemes

Table 5 shows the icons and color scheme used in the component list view and the corresponding color scheme in the Mode panel.

Display	Component status	Corresponding Mode view / Tooltips
Plain black text Example: UART5	The peripheral is not configured (no mode is set) and all modes are available.	Mode Mode Disable Asynchronous Single Wire (Hall-Duplex) Multiprocessor Communication I/DA LIN
Gray italic text Example: <i>LWIP</i>	Peripheral is not available because some constraints are not solved. See tooltip.	LWIP MBEDT Lightheight TCP/IP stack POLICIC Status QUADST Status QUADST Active only if: ETH IP configured / FREERTOS is enabled when MBEDTLS is enabled. RNC RNC
✓ ⊗ Example:: ✓ ETH	The peripheral is configured (at least one mode is set) and all other modes are available. The green check mark indicates that all parameters are properly configured, a cross indicates they are not.	Mode Mode Mill V Activate Rx Err signal

Table 5. Component list, mode icons and color schemes



Display	Component status	Corresponding Mode view / Tooltips
Example:	The peripheral is not configured (no mode is set) and at least one of its modes is unavailable.	Mode Edemail Phil Disable Internal PS Phil Disable Activate_S Disable Internal PS Phil Disable Activate_Vi-Hots Only Device_Only
Example:	The peripheral is configured (one mode is set) and at least one of its other modes is unavailable.	Mode Enternal Phy Datable Stable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable Disable
Example:I2C2	The peripheral is not configured (no mode is set) and no mode is available. Move the mouse over the peripheral name to display the tooltip describing the conflict.	Mode 20 Disable 20 SMBus-Alert-mode SMBus-Alert-mode Conflict with ETH : Mode Mil
Example: IRTIM	Peripheral is not available because of constraints.	IRTIM InfraRed Interface Status: Not available: Channel 1 of TIM16 and TIM17 must be configured details and documentation (Ctrl+d)

Table 5. Comp	onent list, m	node icons and	color schemes	(continued)

4.5.2 Component Mode panel

Select a component from the component list on the left panel to open the **Mode** panel.

The **Mode** panel helps the user configuring the MCU pins based on a selection of peripherals and of their operating modes. Since STM32 MCUs allow a same pin to be used by different peripherals and for several functions (alternate functions), the tool searches for the pinout configuration that best fits the set of peripherals selected by the user. STM32CubeMX highlights the conflicts that cannot be solved automatically (see *Table 5*).

The **Mode** panel also allows to enable middleware and other software components for the project.

Note: For some middleware (USB, FATS, LwIP), a peripheral mode must be enabled before activating the middleware mode. Tooltips guide the user through the configuration. For FatFs, a user-defined mode has been introduced. This allows STM32CubeMX to generate FatFs code without a predefined peripheral mode. Then, it is up to the user to connect the middleware with a user-defined peripheral by updating the generated user_diskio.c/.h driver files with the necessary code.



4.5.3 Pinout view

Select **Pinout view** to show for the selected part number, a graphic representation of the pinout for the selected package (such as. BGA, QFP), where each pin is represented with its name (such as PC4), its configuration state and its current alternate function assignment, if any (such as ETH_MII_RXD0). See *Figure 64* for an example.

TM32 TO CubeMX	File	Window	Help	🎯 🖪 🖻 🄰 🔆 🏹
	STM32F429VITx /	Untitled - Pinout & Configuratio	n	GENERATE CODE
Pinout & C	Configuration	Clock Configuration	Project Manager	Tools
		Additional Softwares	✓ Pinout	
Options Q Categories A->Z	~ >		Pinout view System view	
System Core	>			
Analog			PB7 PB6 PB4 PD7 PD7 PD7 PD7 PD7 PD7 PD7 PD7 PD7 PD7	BA14 BA14 BA14
Timers	<u> </u>	PE3 PE4		VSS VCA
		PE5		PA13
Connectivity		PE6 VBAT		PA12 PA11
CAN1 CAN2		PC13 PC1.		PA10 PA9
V ETH FMC		PC1		PA8 PC9
12C1 12C2		VDD		PCB
SDIO SPI1		PH0/ PH1/		PC7 PC8
SPI2 SPI3		NRST PCO		PD15 PD14
SPI4 UART4		ETH_MDC PC1 ETH_TXD2 PC2		PD13 PD12
UART5 UART7		ETH_TX_CLK PC3	STM32F429VITx	PD11 PD10
UART8 USART1		VSSA	LQFP100	PD9
USART2 USART3 USART6		VRE VDDA		PD8 PB15
USART6 USB_OTG_FS A USB OTG HS		ETH_CRS PAO/ ETH_RX_CLK PA1		PB14 PB13 ETH_TXD1
005_010_10				
Multimedia	<u>></u>	 PA3 PA4 PA6 PA6 PA7 PA7 	A 20 00 10 10 10 10 10 10 10 10 10 10 10 10	
Security	<u>></u>	ETH_COL	8 5 8 (36)-PB1: 2 4 4 4 H H H H H H H H H H H H H H H H H H H	Tooltip
Computing	<u>></u>	Ш		Ė
Middleware	>	€ [] €	R 🕒 🖆 💷	Q ×

Figure 64. Pinout view

The **Pinout** view is automatically refreshed to match the user's component configuration performed in the **Mode** panel.

Assigning pins directly through the **Pinout** view instead of the **Mode** panel requires a good knowledge of the MCU since each individual pin can be assigned to a specific function.



Tips and tricks

See Table 2 for list of menus and shortcuts.

- Use the mouse wheel to zoom in and out.
- Click and drag the chip diagram to move it.
- Click best fit to reset it to best suited position and size.
- Use Pinout > Export pinout menus to export the pinout configuration as .csv text format.
- Some basic controls, such as insuring consistency for blocks of pins, are built-in. See *Appendix A* for details.

4.5.4 Pinout menu and shortcuts

Name or Icon	Shortcut	Description
Keep Current Signals Placement	Ctrl-K	Prevents moving pin assignments to match a new peripheral operating mode. It is recommended to use the new pinning feature that can block each pin assignment individually and leave this checkbox unchecked.
Show User Label	None	Displays user defined labels in the Pinout view.
Undo Mode and pinout	Ctrl-Z	Undoes last configuration steps (one by one).
Redo Mode and pinout	Ctrl-Y	Redoes steps that have been undone (one by one). Warning (limitation): configurations in the platform settings tabs are not restored.
Disable All Modes	Ctrl-D	Resets to "Disabled" all peripherals and middleware modes that have been enabled. The pins configured in these modes (green color) are consequently reset to "Unused" (gray color). Peripheral and middleware labels change from green to black (when unused) or gray (when not available).
Clear Pinouts	Ctrl-P	Clears user pinout configuration in the Pinout view. Note that this action puts all configured pins back to their reset state and disables all the peripheral and middleware modes previously enabled (whether they were using signals on pins or not).
Pins/Signals Option	Ctrl-O	Opens a window showing the list of all the configured pins together with the name of the signal on the pin and a Label field allowing the user to specify a label name for each pin of the list. For this menu to be active, at least one pin must have been configured. Click the pin icon to pin/unpin signals individually. Select multiple rows then right click to open contextual menu and select action to pin or unpin all selected signals at once. Click column header names to sort alphabetically by name or according to placement on MCU.
Clear Single Mapped Signals	Ctrl-M	Clears signal assignments to pins for signals that have no associated mode (highlighted in orange and not pinned).

Table 6. Pinout menu and shortcuts



Name or Icon	Shortcut	Description
List Pinout Compatible MCUs	Alt-L	 Provides a list of MCUs that best match the pin configuration of the current project. The matching can be: An exact match A partial match with hardware compatibility: pin locations are the same, pin names may have been changed A partial match without hardware compatibility: all signals can be mapped but not all at the same pin location Refer to Section 15.
Export pinout with Alternate functions	-	Generates pin configuration as a .csv text file including alternate functions information.
Export pinout without Alternate functions	Ctrl-U	Generates pin configuration as a .csv text file excluding alternate functions information.
Reset used GPIOs	Alt-G	Opens a window to specify the number of GPIOs to be freed among the total number of GPIO pins that are configured.
Set unused GPIOs	Ctrl-G	Opens a window to specify the number of GPIOs to be configured among the total number of GPIO pins that are not used yet. Specify their mode: Input, Output or Analog (recommended configuration to optimize power consumption). Caution: Before using this menu, make sure that debug pins (available under SYS peripheral) are set to access microcontroller debug facilities.
Layout reset	-	-
Q	-	Zooms-in the pinout view.
53	-	Adjusts the chip pinout diagram to the best fit size.
Q	-	Zooms-out the pinout view.
•	-	Rotates 90 degrees clock wise.
4	-	Rotate 90 degrees counter-clock wise.
	-	Flips horizontally between bottom view and top view.
	-	Flips vertically between bottom view and top view.
Q 20 ~ 12C1_SCL 12C1_SDA	-	This Search field allows the user to search the Pinout view for a pin name, a signal name, a signal label or an alternate pin name When it is found, the pin or set of pins matching the search criteria blinks on the Pinout view. Click the Pinout view to stop blinking.

Table 6. Pinout menu and shortcuts (continued)



4.5.5 Pinout view advanced actions

Manually modifying pin assignments

To manually modify a pin assignment, follow the sequence below:

- 1. Click the pin in the **Pinout** view to display the list of all other possible alternate functions together with the current assignment highlighted in blue (see *Figure 65*).
- 2. Click to select the new function to assign to the pin.

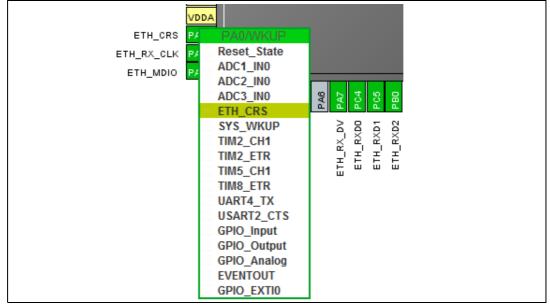


Figure 65. Modifying pin assignments from the Pinout view

Manually remapping a function to another pin

To manually remap a function to another pin, follow the sequence below:

- 1. From the **Pinout** view, hold down the CTRL key then left-click on the pin and hold: if any pins are possible for relocation, they are highlighted in blue and blinking.
- 2. Drag the function to the target pin.
- Caution: A pin assignment performed from the Pinout view overwrites any previous assignment.

Manual remapping with destination pin ambiguity

For MCUs with block of pins consistency (STM32F100x / F101x / F102x / F103x and STM32F105x / F107x), the destination pin can be ambiguous, e.g. there can be more than one destination block including the destination pin. To display all the possible alternative remapping blocks, move the mouse over the target pin.

Note: A "block of pins" is a group of pins that must be assigned together to achieve a given peripheral mode. As shown in Figure 66, two blocks of pins are available on STM32F107xx MCUs to configure the Ethernet peripheral in RMII synchronous mode: {PC1, PA1, PA2, PA7, PC4, PC5, PB11, PB12, PB13, PB5} and {PC1, PA1, PA2, PD10, PD9, PD8, PB11, PB12, PB13, PB5}.



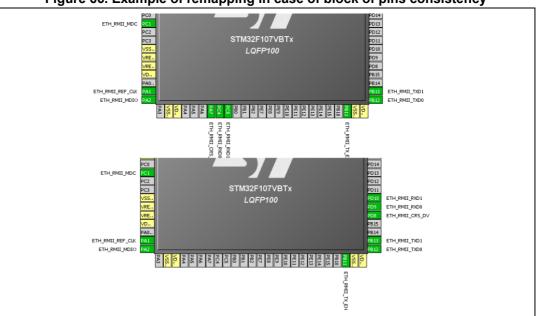


Figure 66. Example of remapping in case of block of pins consistency

Resolving pin conflicts

To resolve the pin conflicts that may occur when some peripheral modes use the same pins, STM32CubeMX attempts to reassign the peripheral mode functions to other pins. The peripherals for which pin conflicts cannot be solved are highlighted in fuchsia with a tooltip describing the conflict.

If the conflict cannot be solved by remapping the modes, the user can try the following:

- If the Keep Current Signals Placement box is checked, try to select the peripherals in a different sequence.
- Uncheck the Keep Current Signals Placement box and let STM32CubeMX try all the remap combinations to find a solution.
- **Manually remap** a mode of a peripheral when you cannot use it because there is no pin available for one of the signals of that mode.

4.5.6 Keep Current Signals Placement

This checkbox is available from the **Pinout** menu. It can be selected or deselected at any time during the configuration. It is unselected by default.

It is recommended to keep the checkbox unchecked for an optimized placement of the peripherals (maximum number of peripherals concurrently used).

The **Keep Current Signals Placement** checkbox should be selected when the objective is to match a board design.

Keep Current Signals Placement is unchecked

This allows STM32CubeMX to remap previously mapped blocks to other pins in order to serve a new request (selection of a new peripheral mode or a new peripheral mode function) which conflicts with the current pinout configuration.



Keep Current Signals Placement is checked

This ensures that all the functions corresponding to a given peripheral mode remain allocated (mapped) to a given pin. Once the allocation is done, STM32CubeMX cannot move a peripheral mode function from one pin to another. New configuration requests are served if feasible within current pin configuration.

This functionality is useful to:

- lock all the pins corresponding to peripherals that have been configured using the Peripherals panel
- maintain a function mapped to a pin while doing manual remapping from the Pinout view.

Тір

If a mode becomes unavailable (highlighted in fuchsia), try to find another pin remapping configuration for this mode by following the steps below:

- 1. From the Pinout view, deselect the assigned functions one by one until the mode becomes available again.
- 2. Then, select the mode again and continue the pinout configuration with the new sequence (see *Appendix A: STM32CubeMX pin assignment rules* for a remapping example). This operation being time consuming, it is recommended to deselect the **Keep Current Signals Placement** checkbox.

Note: Even if Keep Current Signals Placement is unchecked, GPIO_functions (excepted GPIO_EXTI functions) are not moved by STM32CubeMX.

4.5.7 Pinning and labeling signals on pins

STM32CubeMX comes with a feature allowing the user to selectively lock (or pin) signals to pins. This prevents STM32CubeMX from automatically moving pinned signals to other pins when resolving conflicts. Labels, that are used for code generation, can also be assigned to the signals (see Section 6.1 for details).

There are several ways to pin, unpin and label the signals:

- 1. From the **Pinout** view, right-click a pin with a signal assignment. This opens a contextual menu:
 - a) For unpinned signals, select **Signal Pinning** to pin the signal. A pin icon is then displayed on the relevant pin. The signal can no longer be moved automatically (for example when resolving pin assignment conflicts).
 - b) For pinned signals, select **Signal Unpinning** to unpin the signal. The pin icon is removed. From now on, to resolve a conflict (such as peripheral mode conflict), this signal can be moved to another pin, provided the Keep user placement option is unchecked.
 - c) Select **Enter User Label** to specify a user defined label for this signal. The new label replaces the default signal name in the **Pinout** view.



2. From the Pinout menu, select Pins/Signals Options

The Pins/Signals Options window (see Figure 67) lists all configured pins.

	Pin Name	Signal Name	User Label
-		ETH_MDC	
	C2	ETH_TXD2	
Ρ	C3	ETH_TX_CLK	
P	A0/WKUP	ETH_CRS	
P	A1	ETH_RX_CLK	
P	A2	ETH_MDIO	
P	A3	ETH_COL	
P	A4	DAC_OUT1	DAC1
P	A5	DAC_OUT2	DAC2
P	A7	ETH_RX_DV	
P	C4	FTH RXD0	
		Apply	OK Cancel

Figure	67.	Pins/Sic	inals (Options	window
	••••	1 1110/ 01g	,	opnono	

- a) Click the first column to individually pin/unpin signals.
- b) Select multiple rows and right-click to open the contextual menu and select **Signal(s) Pinning** or **Unpinning**.
- c) Select the User Label field to edit the field and enter a user-defined label.
- d) Order list alphabetically by Pin or Signal name by clicking the column header. Click once more to go back to default i.e. to list ordered according to pin placement on MCU.
- Note: Even if a signal is pinned, it is still possible however to manually change the pin signal assignment from the **Pinout** view: click the pin to display other possible signals for this pin and select the relevant one.

4.5.8 Pinout for multi-bonding packages

Multi-bonding has been introduced for packages with low pin counts (less than 20 pins) such as SO8N, TSSOP20 and WLCSP18 packages. it consists of having several MCU pads share a same pin on the package.

Multi-bonding has been introduced on the STM32G0 series for the STM32G031/G041 MCUs.

STM32CubeMX pinout view allows to displays all signals arriving on the pin and allows to select only one per pin, except for analog signals that can be combined with other analog GPIOs.



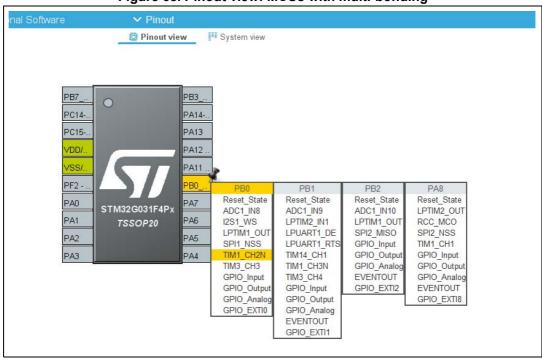


Figure 68. Pinout view: MCUs with multi-bonding

STM32CUbeMX offers also an extended mode selected by right-clicking the pin: it allows to select more than one signal per pin. This mode is meant for test purposes such as loopback tests. It is to be used with caution as it can lead to electrical conflicts or increased power consumption that can damage the device.

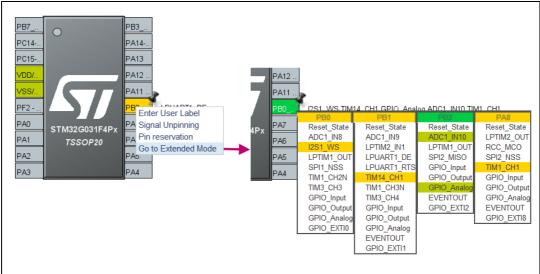


Figure 69. Pinout view: multi-bonding with extended mode



4.5.9 System view

Select **System view** to show all the software configurable components: GPIOs, peripherals and middleware. Clickable buttons allow the user to open the mode and configuration options of the component. The button icon reflects the component configuration status (see *Table 7* for configuration states and Figure System view).

When the user changes the component configuration from the Configuration panel, the system view is automatically refreshed with the new configuration state.

If the user disables the component from the Mode panel, the system view is automatically refreshed and there is no longer a button showing for that component.

STM32CubeMX	Untitled*: S	TM32F423VHHx						
		File	Window	r Hel	р	(19)	🗗 🕒	y 🗙 🖅
Home /	STM32F	423VHHx /	Untitled - Pinout	& Configuration		G	ENERATE CODE	
Pinout &	Configur	ration	Clock Configu		Project Ma	anager		Tools
			Additional Software		 Pinout 			
Options Q Categories A->Z				💭 Pin	out view Syste	m view		
System Core	>				Middlewares			
Analog	>				FREERTOS 😔			
Timers	>							
Connectivity	>	System Cor	e Analog	Timers	Connectivity	Multimedia	Security	Computing
Multimedia	>	DMA 🚣	DAC 😔	TIM1 😔	CAN2 🥝		AES 🥹	
Security	>	GPIO 😔			1201 😒			
Computing	>							
Middleware	>	RCC 🥹						

Table 7. Configuration states

lcon	Description
\odot	Configuration is complete and correct.
	Configuration is correct but some parts remain to be configured (optional).
8	Configuration is invalid and must be fixed for the generated C project to be functional.



Figure 70. System view

GPIO, DMA and NVIC settings can be accessed either via a dedicated button (like other peripherals, or via a tab in the Configuration panel (see *Figure 71*).

C :	74 0-		window tabs				LOW CTNIOOL	4
FIGURE	/1 LC	ontiduration	window tans	$((\mathbf{AP}))$		settings	TOP STIVISZE	4 series)
Igaio		, in garation		10110		000000		

I2C1 Mode and Configuration							
			Mo	ode			
12C 2C							\sim
			Config	uration			
Reset C	onfiguration						
O Parame	eter Settings	😔 User Consta	nts 📀 NVIC	Settings			ottings
	eter Settings	User Consta		Settings			ettings
Search Sig	inals						
Search (Ci	rtI+F)					Show only I	Nodified Pins
Pin Name 🕇	Signal on Pi	n GPIO Pin State	e GPIO mode	GPIO Pull-u	p/ Maximum out	User Label	Modified
°B8	I2C1_SCL	n/a	Alternate Fun		High	000120001	
°B9	I2C1_SDA	n/a	Alternate Fun	Pull-up	High		

4.5.10 Component configuration panel

This panel appears when clicking on a component name in the left panel. It allows the user to configure the functional parameters required to initialize the peripheral or the middleware in the selected operating mode (see *Figure 72*). STM32CubeMX uses these settings to generate the corresponding initialization C code.

The configuration window includes several tabs:

- **Parameter settings** to configure library dedicated parameters for the selected peripheral or middleware,
- NVIC, GPIO and DMA settings to set the parameters for the selected peripheral (see *Section 4.5.14*, *Section 4.5.12* and *Section 4.5.13*).
- User constants to create one or several user defined constants, common to the whole project (see Section 4.5.11).

Invalid settings are detected and are:

- reset to minimum / maximum valid value if user choice is, respectively, smaller / larger than minimum / maximum threshold
- reset to the previous valid value if the previous one is neither a maximum nor a minimum threshold value
- highlighted in fuchsia.



STM32CubeMX Untitl	led*: STM32F423VH	Hx					
STM32 CubeMX	File	Window	Help	(19)	F 🖸	У 🗙	57
Home / S ⁻	TM32F423VHHx	/ Untitled - Pinout & Conf	iguration	GEN	NERATE CODE		
Pinout & Conf	iguration	Clock Configuration	Project Manager			Tools	
			✓ Pinout				
Options Q			I2C1 Mode and Configuration				4
Categories A->Z			Mode				
FSIVIL [€]	12C 12C					\sim	
GPIO							_
12C2			Configuration				
12C3 12S1	Reset Configura	tion					
12S2	🔗 Parameter Sett		Settings 🛛 🥝 GPIO Settings 👘 😔 D	MA Setting	gs		
12S3 12S4	Configure the below	parameters :					
I2S5	Q Search (CrtI+F)	© 0					0
IWDG	✓ Master Features						
LPTIM1 MBEDTLS	I2C Speed N	lode	Standard Mode				
NVIC	I2C Clock Sp	beed (Hz)	100000				
	 Slave Features Clock No Str 	atab Marda	Disabled				
RCC		ress Length selection	7-bit				
RNG RTC	-	s Acknowledged	Disabled				
SAI1	Primary slav		0				
A SDIO SPI1	General Cal	address detection	Disabled				
SPI2							
SPI3 SPI4							
SPI5	Dual Address Ackno	owledged					
✓ STS ✓ TIM1	DualAddressMode Diagnostic:						
TIM2 TIM3		ss is 7 bits length we can have a dual add	ress				
TIM4							
TIM5							

Figure 72. Peripheral mode and Configuration view

Table 8 describes peripheral and middleware configuration buttons and messages.

Buttons and messages	Action
0	Shows / hides the description panel.
Tooltip Enabled Disabled Enabled Disabled I2C_DUALADDRESS_ENABLE	Guides the user through the settings of parameters with valid min-max range. To display it, move the mouse over a parameter value from a list of possible values.
I2C Clock Speed (Hz) 100000 ✓ Decimal Hexadecimal No check	Clicking on the gear icon allows to select whether to display hexadecimal or decimal values, or any value unchecked (No check option).
Search (CrtI+F)	Search
Reset Configuration	Resets the component back to its default configuration (initial settings from STM32CubeMX).

Table 8. Peripheral and Middleware configuration window buttons and tooltips



No check option

By default, STM32CubeMX checks that the parameter values entered by the user are valid. This check can be bypassed by selecting the option No Check for a given parameter. This allows entering you any value (such as a constant) that might not be known by STM32CubeMX configuration.

The validity check can be bypassed only on parameters whose values are of integer type (either hexadecimal or decimal). It cannot be bypassed on parameters coming from a predefined list of possible values or on those which are of non-integer or text type.

To go back to the default mode (decimal or hexadecimal values with validity check enabled), enter a decimal or hexadecimal value and check the relevant option (hexadecimal or decimal check).

Caution: When a parameter depends upon another parameter that is set to No Check:

- Case of a parameter depending on another parameter for the evaluation of its minimum or maximum possible value: If the other parameter is set to No Check, the minimum or maximum value is no longer evaluated and checked.
- Case of a parameter depending on another parameter for the evaluation of its current value: If the other parameter is set to No Check, the value is no longer automatically derived. Instead, it is replaced with the formula text showing as variable the string of the parameter set to No check (see *Figure 73*).

Ľ	TDC Mode and Configuration	
	Mode	
Display Type RGB888 (24 bits)		\sim
	Configuration	
Reset Configuration		
Parameter Settings Settings Settings	constants 📀 NVIC Settings 📀 GPIO Settings	
onfigure the below parameters :		
unigure the below parameters .		
Search (CrtI+F)		0
Synchronization for Width		
Horizontal Synchronization Width	MY_HSYNC_VALUE pixels	
Horizontal Back Porch	7 pixels	
Active Width	640 pixels	
Horizontal Front Porch	6 pixels	
HSync Width	MY_HSYNC_VALUE-1	
Accumulated Horizontal Back Porch Width	MY_HSYNC_VALUE-1+7	
Accumulated Active Width	MY_HSYNC_VALUE-1+7+640	
Total Width	MY_HSYNC_VALUE-1+7+640+6	
 Synchronization for Height 		
Vertical Synchronization Height	4 lines	
Vertical Back Porch	2 lines	
Active Height	480 lines	

Figure 73. Formula when input parameter is set in No Check mode

4.5.11 User Constants configuration window

The **User Constants** tab is available to define user constants (see *Figure 74*). Constants are automatically generated in the STM32CubeMX user project within the main.h file (see *Figure 75*). Once defined, they can be used to configure peripheral and middleware parameters (see *Figure 76*).



Figure 74. User Constants tab

	SWPMI1 Mode and Configuration							
Mode Full-Duplex (nor	mal mode)		vioue			~		
		Con	figuration		_			
Reset Configuration								
Parameter Settings	🛛 User Constants	NVIC Settings	Settings 😔	OMA Settings				
Search Constants								
Search (CrtI+F)					add	remove		
	Constant Name			Constant Value	9			
CONSTANT_1			10					
CONSTANT_2 CONSTANT_3			0xff CONSTANT 1					
CONSTANT_4				ONSTANT_1)*100/CONST	FANT 1			
CONSTANT_5			(CONSTANT_2 - CO					

Figure 75. Extract of the generated main.h file

/* Includes*/
/* USER CODE BEGIN Includes */
/* USER CODE END Includes */
/* Private define*/
#define CONSTANT_1 10
#define CONSTANT_2 0xff
#define CONSTANT_3 CONSTANT_1
<pre>#define CONSTANT_4 (CONSTANT_3+CONSTANT_1)*100/CONSTANT_1</pre>
<pre>#define CONSTANT_5 (CONSTANT_2 - CONSTANT_1)</pre>
/* USER CODE BEGIN Private defines */
/* USER CODE END Private defines */



	Configuration								
Reset Configuration									
⊘ Parameter Settings	🥝 User Constants	NVIC Settings	📀 GPIO Settings	OMA Settings					
Configure the below parame	ters :								
Q Search (CrtI+F)		\odot							
✓ Basic Parameters									
 Basic Parameters Voltage Class 			Class B						
			Class B CONSTANT_1						
Voltage Class	ncy								
Voltage Class Bit Rate Prescaler	ncy		CONSTANT_1						
Voltage Class Bit Rate Prescaler SWPMI Clock freque	-		CONSTANT_1 64000 kHz						

Figure 76. Using constants for peripheral parameter settings

Creating/editing user constants

Click the **Add** button to open the **User Constants** tab and create a new user-defined constant (see *Figure* 77).

A constant consists of:

- A name that must comply with the following rules:
 - It must be unique.
 - It must not be a C/C++ keyword.
 - It must not contain a space.
 - It must not start with digits.
- A value, which can be (see *Figure 74* for examples):
 - a simple decimal or hexadecimal value
 - a previously defined constant
 - a formula using arithmetic operators (subtraction, addition, division, multiplication, and remainder) and numeric value or user-defined numeric constants as operands
 - a character string: the string value must be between double quotes (example: "constant_for_usart").

Once a constant is defined, its name and/or value can be changed: double-click the row that specifies the user constant to modify. This opens the **User Constants** tab for edition. The change of constant name is applied wherever the constant is used. This does not affect the peripheral or middleware configuration state. Changing the constant value impacts the parameters that use it and might result in invalid settings (such as exceeding a maximum threshold). Invalid parameter settings are highlighted in fuchsia.



FIG	are 77. Specifying user constar	it value and name
	User Constants	X
	constant Name CONSTA	ANT_1
	constant Value 10	
	OK Cancel	

Deleting user constants

Click the **Remove** button to delete an existing user-defined constant.

The user constant is then automatically removed except in the following cases:

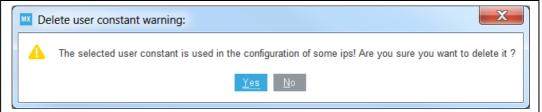
When the constant is used for the definition of another constant. In this case, a popup window displays an explanatory message (see Figure 78).

Figure 78. Deleting an user constant is not allowed when it is already used for another constant definition

De	elete user constant warning:
	Cannot delete, the selected user constant is used in the definition of another constant !
	OK Cancel

When the constant is used for the configuration of a peripheral or middleware library parameter. In this case, the user is requested to confirm the deletion since the constant removal results in a invalid peripheral or middleware configuration (see Figure 79).

Figure 79. Confirmation request to delete a constant for parameter configuration



Clicking Yes leads to an invalid peripheral configuration (see Figure 80).

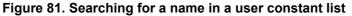


😵 Parameter Settings 🛛 📀 User Constants 🛛 📀 NVIC Settings 📄 🤡 GPIO Settings 📄 📀 DMA Settings Configure the below parameters : Q Search (CrtI+F) \odot ⊗ Basic Parameters Voltage Class Class B Bit Rate Prescaler ⊗ CONSTANT_1 SWPMI Clock frequency 64000 kHz Bit Rate 16000 kBits/s Transmission Buffering Mode No Software buffer Reception Buffering Mode No Software buffer

Figure 80. Consequence when deleting a user constant for peripheral configuration

Searching for user constants

The **Search Constants** field makes it possible the search of a constant name or value in the complete list of user constants (see *Figure 81* and *Figure 82*).



NVIC Settings Q GPIO Settings	OMA Settings
Parameter Settings	Ser Constants
Search Constants CONSTANT_1	add remove
Constant Name	Constant Value
CONSTANT_1	10
CONSTANT_3	CONSTANT_1 + CONSTANT_2

Figure 82. Searching for a value in a user constant list

NVIC Settings OPIO Settings	OMA Settings	
Parameter Settings		User Constants
Search Constants		add remove
Constant Name		Constant Value
CONSTANT_1	10	



UM1718

4.5.12 GPIO configuration window

Click **GPIO** in the **System view** panel to open the **GPIO configuration** window to configure the GPIO pin settings (see *Figure 83*). The configuration is populated with default values that might not be adequate for some peripheral configurations. In particular, check if the GPIO speed is sufficient for the peripheral communication speed, and select the internal pull-up whenever needed.

Note: GPIO settings can be accessed for a specific peripheral instance via the dedicated window in the peripheral instance configuration window. In addition, GPIOs can be configured in output mode (default output level). The generated code is updated accordingly.

	GPIO Mode and Configuration								
	Mode								
	ivide								
				Configuration			_		
				Configuration					
🗖 Group	By Periphera	ls							
	0.0004				A NR 40				
😔 GPIO	🕑 ADCT		🥝 I2C1		🕑 NVIC				
Search Sig	gnals								
Search (C	rtI+F)					🗖 S	how only Mod	lified Pins	
Pin Na 🗢	Signal on P	. GPIO outp	GPIO mode	GPIO Pull	Maximum o	Fast Mode	User Label	Modified	
PA9	n/a	n/a		No pull up		n/a			
PB15	n/a	Low		No pull up		n/a			
PC8	n/a	n/a	Input mode	No pull up	n/a	n/a			

Figure 83	. GPIO configuratio	n window -	GPIO	selection
-----------	---------------------	------------	------	-----------

Click on a row or select a set of rows to display the corresponding GPIO parameters:

GPIO PIN state

Changes the default value of the GPIO output level. It is set to low by default and can be changed to high.

- GPIO mode (analog, input, output, alternate function)
 Selecting a peripheral mode in the Pinout view automatically configures the pins with the relevant alternate function and GPIO mode.
- GPIO pull-up/pull-down

Set to a default value, can be configured when other choices are possible.

• GPIO maximum output speed (for communication peripherals only)

Set to Low by default for power consumption optimization, can be changed to a higher frequency to fit application requirements.

User Label

Changes the default name (such as GPIO_input) into a user defined name. The **Pinout** view is updated accordingly. The GPIO can be found under this new name via the **Find** menu.



The Group by Peripherals checkbox allows the user to group all instances of a peripheral under the same window (see Figure 84).

			(Configuratio	n			
🔽 Gro	up By Perip	herals						
😔 GPI	0 📀 AD)C 🛛 🥝 I2C						
	Signals							
	Signals (CrtI+F)					🗖 Sho	w only Modi	ified Pins
Search		GPIO outp	<mark>GPIO mode</mark>	GPIO Pull	Maximum		w only Modi User Label	
Search Pin Nam	(CrtI+F)	GPIO outp		GPIO Pull No pull-u				
Search	(<i>Crtl+F</i>) le Signal		Output Pu		Low	. Fast Mode		

As shown in Figure 85, row multi-selection can be performed to change a set of pins to a given configuration at the same time.

			(Configuratio	n			
Grou	p By Periphe	erals						
) 😔 ADC							
		0 120						
Search S	Signals							
Search	-					Sho	w only Mod	ified Pins
Search	Onirij							
Pin Name) Signal 🗢	GPIO outp	GPIO mode	GPIO Pull	Maximum	. Fast Mode	User Label	Modified
PB6	I2C1_SCL	n/a	Alternate	No pull-u	Low	Disable		
PB9	I2C1_SDA		Alternate	Pull-up	Low	Disable		
PF1		n/a	Alternate	Pull-up	Low	n/a		
PF0	I2C2_SDA	n/a	Alternate	Pull-up	Low	n/a		
PA8	I2C3_SCL	n/a	Alternate	Pull-up	Low	n/a		
PC9	I2C3_SDA	n/a	Alternate	No pull-u	Low	n/a		
PB9#PF	1#PF0#PA8	Configuratio	n :					
GPIO m	ode							~
GPIO Pu	ill-up/Pull-do	wn		Pull-up				\sim
Maximur	m output spe	ed						\sim
User Lab	hol							
User La								

Figure 85. Multiple pins configuration



4.5.13 DMA configuration window

Click DMA in the System view to open the DMA configuration window.

This window is used to configure the generic DMA controllers available on the MCU. The DMA interfaces allow to perform data transfers between memories and peripherals while the CPU is running, and memory to memory transfers (if supported).

Note: Some peripherals (such as **USB** or **Ethernet**) have their own DMA controller, which is enabled by default or via the Peripheral Configuration window.

Clicking **Add** in the **DMA configuration** window adds a new line at the end of the DMA configuration window with a combo box proposing to choose between possible **DMA requests** to be mapped to peripherals signals (see *Figure 86*).

	Conf	iguration		
📀 DMA1, DMA2 🛛 😒 MemToMem				
DMA Request	Stream	Direction		Priority
Select ~				
DMA_GENERATOR7 I2C1_RX I2C1_TX			, A	Add Delete
12C2_RX 12C2_TX			Peripheral	Memory
I2C3_RX I2C3_TX MEMTOMEM		Increment Address		
Use Fifo Threshold	~	Data Width	\sim	~
		Burst Size	\sim	
DMA Request Generator Settings —				
Request generation Signal				~
Signal polarity				~
Request number				

Selecting a DMA request automatically assigns a stream among all the streams available, a direction and a priority. When the DMA channel is configured, it is up to the application code to fully describe the DMA transfer run-time parameters such as the start address.

The DMA request (called channel for STM32F4 MCUs) is used to reserve a stream to transfer data between peripherals and memories (see *Figure 87*). The stream priority is used to decide which stream to select for the next DMA transfer.

DMA controllers support a dual priority system using the software priority first, and in case of equal software priorities, a hardware priority that is given by the stream number.



Figure 86. Adding a new DMA request

📀 DMA1, DMA2 🛛 📀 M		Configuration		
DMA Request	Stream	Direction		Priority
I2C1_TX I2C1_RX	DMA1 Stream 0 DMA1 Stream 1	Memory To Peripheral Peripheral To Memory	Low Low	
				Add Delete
DMA Request Settings			Peripheral	Memory
Mode Normal	~	Increment Address		
Use Fifo 🔲 Thres	hold 🗸 🗸	Data Width B	yte 🗸	Byte ~
		Burst Size	~	
DMA Request Generator	Settings			
Request generation Sign	al			~
Signal polarity				\sim

Figure 87. DMA configuration

Additional DMA configuration settings can be done through the **DMA configuration** window:

- **Mode:** regular mode, circular mode, or peripheral flow controller mode (only available for the SDIO peripheral).
- Increment Add: the type of peripheral address and memory address increment (fixed or postincremented, in which case the address is incremented after each transfer). Click the checkbox to enable the post-incremented mode.
- Peripheral data width: 8, 16, or 32 bits
- Switching from the default direct mode to the FIFO mode with programmable threshold:
 - a) Click the Use FIFO checkbox.
 - b) Configure the *peripheral and memory data width* (8, 16, or 32 bits).
 - c) Select between *single transfer* and *burst transfer*. If you select burst transfer, choose a burst size (1, 4, 8, or 16).

In case of memory-to-memory transfer (MemToMem), the DMA configuration applies to a source memory and to a destination memory.



Configuration						
🥝 DMA1, DMA2 🛛 📀	MemToMem	Coning				
DMA Request	Stream		Direction		Priority	
МЕМТОМЕМ	DMA1 Stream 2	1	Memory To Memory	Low		
					Add Delete	
DMA Request Settings						
DMA Request Octings				Src Memory	Dst Memory	
Mode Normal	\sim	I	Increment Address			
Use Fifo 🔽 Thre	eshold Full	~	Data Width	Byte 🗸	Byte ~	
		I	Burst Size	Single \checkmark	Single ~	
DMA Request Generat	or Settings					
Request generation Sig	gnal				\sim	
Signal polarity					~	
Request number						

Figure 88. DMA MemToMem configuration

4.5.14 NVIC configuration window

Click **NVIC** in the **System** view to open the Nested Vector interrupt controller configuration window (see *Figure 89*).

Interrupt unmasking and interrupt handlers are managed within two tabs:

- **NVIC**, to enable peripheral interrupts in the NVIC controller and to set their priorities
- **Code generation**, to select options for interrupt related code generation

Enabling interruptions using the NVIC tab view

The **NVIC** view (see *Figure 89*) does not show all possible interrupts, but only the ones available for the peripherals selected in the **Pinout & Configuration** panels. System interrupts are displayed but can never be disabled.

Check/uncheck the Show only enabled interrupts box to filter or not enabled interrupts.

When DMA channels are configured in the project, check/uncheck "Force DMA channels interrupts" to automatically enable/disable DMA channels interrupts in the generated code.

Use the **search field** to filter out the interrupt vector table according to a string value. As an example, after enabling UART peripherals from the **Pinout** panel, type UART in the NVIC search field and click the green arrow close to it: all UART interrupts are displayed.

Enabling a **peripheral interrupt** generates NVIC function calls **HAL_NVIC_SetPriority** and **HAL_NVIC_EnableIRQ** for this peripheral.



STM32CubeMX Unt	titled*: STM3	2F401CBUx				-	- 🗆 :
rm32 CubeMX		File	Window	Help	19	F 🖸 🎽 :	* 5
Home > STM	32F401CE	BUx 🔰 Untitled	Pinout & Configuration	\rangle	GENE	RATE CODE	
Pinout & (Configu	ration	Clock Configuration	Project Ma	nager	Tools	
		✓ Sot	tware Packs	✓ Pinout			
۹	~ @			NVIC Mode and Configurat	tion		
Categories A->Z	z	_		Configuration			
System Core	~	S NVIC S C₀	de generation	gui en en			
System Core		Priority Group 4 bi	ts for pre-emption priority 0 bits	for su	Priority and Sub Prior	rity 🗌 Sort by interrupts	s names
DMA [⊕]							
GPIO		Search Sea	rch (CrtI+F)	Show available interr	rupts 🗸	Force DMA chan	nels Interrupts
IWDG NVIC	_		NVIC Interru	pt Table	Enabled	Preemption Priority	Sub Priority
RCC		Non maskable interr	upt		×	0	0
 SYS 		Hard fault interrupt					0
WWDG		Memory manageme	nt fault		~	0	0
		Pre-fetch fault, mem	ory access fault		×	0	0
	>	Undefined instruction	or illegal state		✓	0	0
Analog		System service call	via SWI instruction		~	0	0
Timers	>	Debug monitor			\checkmark	-	0
Timers		Pendable request fo			\checkmark	-	0
Connectivity	>	Time base: System			\checkmark		0
,		PVD interrupt throug				-	0
Multimedia	>	Flash global interrup	1			-	0
		RCC global interrupt	- Catalana A				0
Computing	>	DMA1 stream0 glob DMA1 stream6 glob					0
		I2C1 event interrupt	arinterrupt				0
Middleware	\sim	I2C1 error interrupt				•	0
÷		SPI1 global interrupt					0
FATES		DMA2 stream0 glob	al interrupt				0
FREERTOS		DMA2 stream3 glob				-	0
LIBJPEG		USB On The Go FS				-	0
MBEDTLS		FPU global interrupt	9			-	0
PDM2PCM USB_DEVICE USB_HOST	E						

_. -----. . .

When FreeRTOS is enabled, an additional column is shown (see Figure 90).

In this case, all the interrupt service routines (ISRs) that are calling the interrupt safe FreeRTOS APIs must have a priority lower than the priority defined in the LIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY parameter (the highest the value, the lowest the priority). The check in the corresponding checkbox guarantees that the restriction is applied.



47/

If an ISR does not use such functions, the checkbox can be unchecked and any priority level can be set. It is possible to check/uncheck multiple rows (see rows highlighted in blue in *Figure 90*).

Configuration						
⊘ NVIC ⊘ Code generation						
Priority Group 4 bits for pre-emption	n priori	✓ Sort by	Premption	Priority and Sub Priority		
Search (Crt1+F)	0	Show of Sho	nly enabled	interrupts		
NVIC Interrupt Table	Enabl	. Preemption Pri	. Sub Prior	. Uses FreeRTOS fun		
Non maskable interrupt	\sim	0	0			
Hard fault interrupt	\sim	0	0			
Memory management fault	\sim	0	0			
Pre-fetch fault, memory access fault	\checkmark	0	0			
Undefined instruction or illegal state	\checkmark	0	0			
System service call via SWI instruction	\checkmark	0	0			
Debug monitor	\checkmark	0	0			
Pendable request for system service	\sim	15	0	\checkmark		
Time base: System tick timer	\sim	15	0	\checkmark		
PVD interrupt through EXTI line 16		5	0	✓		
Flash global interrupt		5	0	✓		
RCC global interrupt		5	0	✓		
I2C1 event interrupt		5	0	\checkmark		
I2C1 error interrupt		5	0	\checkmark		
SPI1 global interrupt		5	0	✓		
USB On The Go FS global interrupt		5	0	✓		
FPU global interrupt		5	0	✓		
Enabled Preemption Priority	~	Sub Priority	- 🔲 Use:	s FreeRTOS functions		

Figure 90. NVIC configuration tab - FreeRTOS enabled

Peripheral dedicated interrupts can also be accessed through the NVIC window in the configuration window (see *Figure 91*).

Figure 91. I2C NVIC configuration window

Configuration Reset Configuration							
NVIC Settings							
😔 Para	meter Settings			😔 User Const	ants		
NVIC Interrupt	t Table	Enabled	Preem	ption Priority	Sub Priority		
2C1 event interrupt			5		0		
2C1 error interrupt			5		0		



STM32CubeMX NVIC configuration consists in selecting a priority group, enabling/disabling interrupts and configuring interrupts priority levels (preemption and sub-priority levels):

1. Select a **priority group**

Several bits allow to define NVIC priority levels, they are divided in two groups, preemption priority and sub-priority. For example, in the case of STM32F4 MCUs, the NVIC priority group 0 corresponds to 0-bit preemption and 4-bit sub-priority.

- 2. In the interrupt table, click one or more rows to select one or more interrupt vectors. Use the widgets below the interrupt table to configure the vectors one by one or several at a time:
 - Enable checkbox: check/uncheck to enable/disable the interrupt.
 - Preemption priority: select a priority level. The preemption priority defines the ability of one interrupt to interrupt another.
 - Sub-priority: select a priority level. Defines the interrupt priority level.

Code generation options for interrupt handling

The **Code Generation** view allows customizing the code generated for interrupt initialization and interrupt handlers:

• Selection/Deselection of all interrupts for sequence ordering and IRQ handler code generation

Use the checkboxes in front of the column names to configure all interrupts at a time (see *Figure 92*). Note that system interrupts are not eligible for init sequence reordering as the software solution does not control it.

	Configuration	
📀 NVIC 🛛 📀 Code generation		
Enabled interrupt table	Select for init sequence ordering	Generate IRQ handler
Memory management fault		✓
Pre-fetch fault, memory access fault		
Undefined instruction or illegal state		✓
System service call via SWI instructi.		✓
Debug monitor		✓
Pendable request for system service	•	✓
Time base: System tick timer		✓
RCC global interrupt		
ADC1, ADC2 and ADC3 global inter.	. 🔽	✓
CAN1 TX interrupts		✓
I2C1 event interrupt		
Interrupt unmasking ordering table	(interrupt init code is moved after all	the peripheral init code)
Rank	Interrupt name	
	and ADC3 global interrupts	
2 CAN1 TX inte		
3 I2C1 event in	•	
4 RCC global i	nterrupt	
	≣l [≣]
	Move up	

Figure 92. NVIC Code generation – All interrupts enabled

UM1718 Rev 47



```
Default initialization sequence of interrupts
   By default, the interrupts are enabled as part of the peripheral MSP initialization
   function, after the configuration of the GPIOs and the enabling of the peripheral clock.
   This is shown in the CAN example below, where HAL NVIC SetPriority and
   HAL NVIC EnableIRQ functions are called within stm32xxx hal msp.c file inside the
   peripheral msp_init function.
   Interrupt enabling code is shown in bold:
  void HAL_CAN_MspInit(CAN_HandleTypeDef* hcan)
  £
  GPIO_InitTypeDef GPIO_InitStruct;
  if(hcan->Instance==CAN1)
  ł
   /* Peripheral clock enable */
    __CAN1_CLK_ENABLE();
   /**CAN1 GPIO Configuration
   PD0
           ----> CAN1 RX
   PD1
           ----> CAN1 TX
   */
   GPIO_InitStruct.Pin = GPIO_PIN_0 | GPIO_PIN_1;
   GPIO InitStruct.Mode = GPIO MODE AF PP;
   GPIO_InitStruct.Pull = GPIO_NOPULL;
   GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
   GPIO InitStruct.Alternate = GPIO AF9 CAN1;
   HAL_GPIO_Init(GPIOD, &GPIO_InitStruct);
/* Peripheral interrupt init */
   HAL_NVIC_SetPriority(CAN1_TX_IRQn, 2, 2);
  HAL NVIC EnableIRQ(CAN1 TX IRQn);
  }
 }
   For EXTI GPIOs only, interrupts are enabled within the MX GPIO Init function:
 /*Configure GPIO pin : MEMS_INT2_Pin */
  GPIO_InitStruct.Pin = MEMS_INT2_Pin;
  GPIO_InitStruct.Mode = GPIO_MODE_EVT_RISING;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  HAL_GPIO_Init(MEMS_INT2_GPIO_Port, &GPIO_InitStruct);
  /* EXTI interrupt init*/
  HAL_NVIC_SetPriority(EXTI15_10_IRQn, 0, 0);
  HAL NVIC EnableIRQ(EXTI15 10 IRQn);
```

For some peripherals, the application still needs to call another function to actually activate the interruptions. Taking the timer peripheral as an example, the *HAL_TIM_IC_Start_IT* function needs to be called to start the Timer input capture (IC) measurement in interrupt mode.



Configuration of interrupts initialization sequence

Checking **Select for Init sequence ordering** for a set of peripherals moves the HAL_NVIC function calls for each peripheral to a same dedicated function, named **MX_NVIC_Init**, defined in the main.c. Moreover, the HAL_NVIC functions for each peripheral are called in the order specified in the **Code generation** view bottom part (see *Figure 93*).

As an example, the configuration shown in *Figure 93* generates the following code:

```
/** NVIC Configuration
*/
void MX NVIC Init(void)
ſ
 /* CAN1_TX_IRQn interrupt configuration */
HAL_NVIC_SetPriority(CAN1_TX_IRQn, 2, 2);
HAL NVIC EnableIRQ(CAN1 TX IRQn);
 /* PVD_IRQn interrupt configuration */
HAL_NVIC_SetPriority(PVD_IRQn, 0, 0);
HAL NVIC EnableIRQ(PVD IRQn);
 /* FLASH_IRQn interrupt configuration */
HAL_NVIC_SetPriority(FLASH_IRQn, 0, 0);
HAL_NVIC_EnableIRQ(CAN1_IRQn);
 /* RCC_IRQn interrupt configuration */
HAL_NVIC_SetPriority(RCC_IRQn, 0, 0);
HAL_NVIC_EnableIRQ(CAN1_IRQn);
 /* ADC_IRQn interrupt configuration */
HAL_NVIC_SetPriority(ADC_IRQn, 0, 0);
HAL_NVIC_EnableIRQ(ADC_IRQn);
```

}

Interrupts handler code generation

By default, STM32CubeMX generates interrupt handlers within the stm32xxx_it.c file. As an example:

```
void NMI_Handler(void)
{
    HAL_RCC_NMI_IRQHandler();
}
void CAN1_TX_IRQHandler(void)
{
    HAL_CAN_IRQHandler(&hcan1);
}
```

The column **Generate IRQ Handler** allows the user to control whether the interrupt handler function call can be generated or not. Deselecting CAN1_TX and NMI interrupts from the **Generate IRQ Handler** column as shown in *Figure 93* removes the code mentioned earlier from the stm32xxx_it.c file.



🛛 🥝 NVIC 🛛 📀 Code g	eneration		
Enabled interrupt t	able 🗸 Sele	ect for init sequence ordering	Generate IRQ handler
Non maskable interrupt			
Hard fault interrupt			\checkmark
Memory management fau	lt		✓
Pre-fetch fault, memory ac	cess fault		\checkmark
Undefined instruction or il	legal state		\checkmark
System service call via SV	VI instructi		✓
Debug monitor			\checkmark
Pendable request for syst	em service		\checkmark
Time base: System tick tir	ner		\checkmark
RCC global interrupt		\checkmark	✓
CAN1 TX interrupts		\checkmark	
Rank		t init code is moved after all Interrupt name	the peripheral init code)
	C1, ADC2 and ADC	3 global interrupts	
	N1 TX interrupts		
3 120	C1 event interrupt		
	C global interrupt		

Figure 93. NVIC Code generation - IRQ Handler generation

4.5.15 FreeRTOS configuration panel

Through STM32CubeMX FreeRTOS configuration window, the user can configure all the resources required for a real-time OS application, and reserve the corresponding heap. FreeRTOS elements are def/ined and created in the generated code using CMSIS-RTOS API functions. Follow the sequence below:

- 1. In the **Pinout & Configuration** tab, click FreeRTOS to reveal the Mode and Configuration panels (see *Figure 94*).
- 2. Enable freeRTOS in the Mode panel.
- 3. Go to the configuration panel to proceed with configuring FreeRTOS native parameters and objects, such as tasks, timers, queues, and semaphores. In the Config tab, configure Kernel and Software settings. In the Include parameters tab, select the API functions required by the application and this way, optimize the code size. Both Config and Include parameters are part of the FreeRTOSConfig.h file.



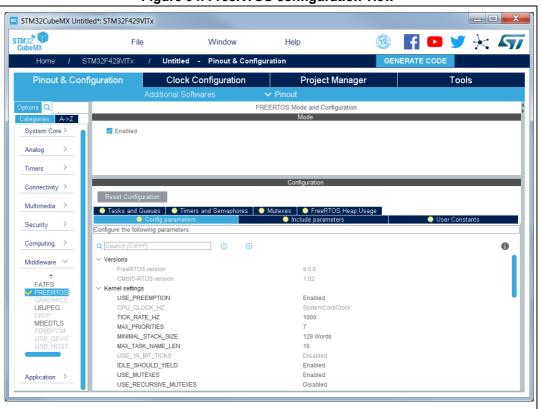


Figure 94. FreeRTOS configuration view

Tasks and Queues tab

As any RTOS, FreeRTOS allows structuring a real-time application into a set of independent tasks, with only one task being executed at a given time. Queues are meant for inter-task communications: they allow to exchange messages between tasks or between interrupts and tasks.

The **FreeRTOS Tasks and Queues** tab enables the creation and configuration of such tasks and queues (see *Figure 95*).

The corresponding initialization code is generated within main.c or freeRTOS.c if the option "generate code as pair of .c/.h files per peripherals and middleware" is set in the **Project Settings** menu, or within main.c by default, or within freeRTOS.c if the option "generate code as pair of .c/.h files per peripherals and middleware" is set in the **Project Manager** menu.



	Onfig paramet		hores 🥺 Mute	Include par	eRTOS Heap Usa; rameters		🕑 User Co	nstants
Tasks Task Name	Priority	Stack Size (Wor		Code Generat		Allocation	Buffer Nam	
defaultTask	osPriorityNormal		StartDefaultTask		NULL	Dynamic	NULL	NULL
Task_A myTask B	osPriorityHigh osPriorityLow	128 256	StartTask_A StartTask B	Default Default	NULL	Dynamic Dynamic	NULL	NULL
							, A	Add Delete
						D		
-			Harry Ola					
Queue Na		Queue Size	Item Siz		Allocation mic	Buffer N NULL		Control Block Name
Queues Queue Na myQueue_1 myQueue_2	ame (16 32	Queue Size	Item Size uint16_t uint16_t	e Dynai Dynai Dynai	mic	NULL	ame C NUL NUL	L

Figure 95. FreeRTOS: configuring tasks and queues

Tasks

Under the **Tasks** section, click the **Add** button to open the **New Task** window where task **name**, **priority**, **stack size** and **entry function** can be configured (see *Figure 96*). These settings can be updated at any time: double-clicking a task row opens again the new task window for editing.

The entry function can be generated as weak or external:

- When the task is generated as weak, the user can propose a definition different from the one generated by default.
- When the task is **extern**, it is up to the user to provide its function definition.

By default, the function definition is generated including user sections to allow customization.

Queues

Under the **Queues** section, click the **Add** button to open the **New Queue** window where the queue **name**, **size** and **item size** can be configured (see *Figure 96*). The queue size corresponds to the maximum number of items that the queue can hold at a time, while the item size is the size of each data item stored in the queue. The item size can be expressed either in number of bytes or as a data type:

- 1 byte for uint8_t, int8_t, char and portCHAR types
- 2 bytes for uint16_t, int16_t, short and portSHORT types
- 4 bytes for uint32_t, int32_t, int, long and float
- 8 bytes for uint64 t, int64 t and double

By default, the FreeRTOS heap usage calculator uses four bytes when the item size cannot be automatically derived from user input.

These settings can be updated at any time: double-clicking a queue row opens again the new queue window for editing.



		nores 🕑 Mute		`	je	😔 User Co	onstants
Priority	Stack Size (Wor	Entry Function			Allocation	Buffer Nar	
				NULL	Dynamic	NULL	NULL
							NULL
osPriorityLow	256	StartTask_B	Default	NULL	Dynamic	NULL	NULL
ame (Queue Size	Item Size	e	Allocation	Buffer N	ame (Control Block Name
16		uint16_t	Dynam	ic	NULL	NUI	
32		uint16_t	Dynam	ic	NULL	NU	LL
	Config paramet Priority osPriorityNormal osPriorityHigh osPriorityLow	Config parameters Priority Stack Size (Wor osPriorityNormal 128 osPriorityLigh 128 osPriorityLow 256 ame Queue Size 16	Config parameters Priority Stack Size (Wor Entry Function osPriorityNormal 128 StartDefaultTask osPriorityHigh 128 StartTask_A osPriorityLow 256 StartTask_B ame Queue Size Item Siz 16 uint16_t	Config parameters Include para Priority Stack Size (Wor. Entry Function Code Generati. osPriorityNormal 128 StartDefaultTask Default osPriorityLow 256 StartTask_A Default osPriorityLow 256 Include para Queue Size Item Size 16 Unt16_t Dynamic	Config parameters Config parameters Priority Stack Size (Wor Entry Function Code Generati Parameter osPriorityNormal 128 StartDefaultTask Default NULL osPriorityLow 256 StartTask_B Default NULL metric StartTask_B Default NULL ame Queue Size Item Size Allocation 16 uint16_t Dynamic	Config parameters Config parameters Priority Stack Size (Wor Entry Function Code Generati Parameter Allocation osPriorityNormal 128 StartDefaultTask Default NULL Dynamic osPriorityLow 256 StartTask_B Default NULL Dynamic osPriorityLow 256 StartTask_B Default NULL Dynamic ame Queue Size Item Size Allocation Buffer N 16 uint16_t Dynamic NULL	Config parameters Vuser C V Vuser C Vuser C Vuser

Figure 96. FreeRTOS: creating a new task

The following code snippet shows the generated code corresponding to *Figure 95*.

```
/* Create the thread(s) */
 /* definition and creation of defaultTask */
 osThreadDef(defaultTask, StartDefaultTask, osPriorityNormal, 0, 128);
 defaultTaskHandle = osThreadCreate(osThread(defaultTask), NULL);
 /* definition and creation of Task_A */
 osThreadDef(Task_A, StartTask_A, osPriorityHigh, 0, 128);
 Task_AHandle = osThreadCreate(osThread(Task_A), NULL);
 /* definition and creation of Task_B */
 osThreadDef(Task_B, StartTask_B, osPriorityLow, 0, 256);
 Task_BHandle = osThreadCreate(osThread(Task_B), NULL);
 /* Create the queue(s) */
 /* definition and creation of myQueue_1 */
 osMessageQDef(myQueue_1, 16, 4);
 myQueue_1Handle = osMessageCreate(osMessageQ(myQueue_1), NULL);
 /* definition and creation of myQueue_2 */
 osMessageQDef(myQueue_2, 32, 2);
 myQueue_2Handle = osMessageCreate(osMessageQ(myQueue_2), NULL);
```

Timers, Mutexes and Semaphores

FreeRTOS timers, mutexes and semaphores can be created via the FreeRTOS **Timers and Semaphores** tab. They first need to be enabled from the Config tab (see *Figure 97*).



			Configuration			
Reset Configu	ration					
🕑 Tasks and Qı ୍ର	ueues 🛛 🕑 Timers a Config parameters	and Semaphores 🛛 😔	Mutexes 🛛 😔 FreeRTO 😔 Include parame	DS Heap Usage ters	📀 Use	er Constants
Timers —						
Timer Name	Callback	Туре	Code Generation Op.	. Parameter	Allocation	Control Block Name
myTimer01	Callback01	osTimerPeriodic	Default	NULL	Dynamic	NULL
myTimer02	Callback02	osTimerOnce	Default	NULL	Dynamic	NULL
Binary Semaphore			A11 P		0.1.10	
	emaphore Name	Dynamic	Allocation	NU		lock Name
S		Dynamic	Allocation	NU		
S nyBinarySem01	emaphore Name	Dynamic	Allocation	NU		lock Name Add Delete
S myBinarySem01 Counting Semaphe	emaphore Name ores		Allocation			Add Delete
S myBinarySem01 Counting Semaph Semaph	iemaphore Name ores ————————————————————————————————————	Count		Allocation	LL Co	
S myBinarySem01 Counting Semaphe	iemaphore Name ores ————————————————————————————————————		Allocation			Add Delete

Figure 97. FreeRTOS - Configuring timers, mutexes and semaphores

Under each object dedicated section, clicking the **Add** button to open the corresponding **New <object>** window, where the object specific parameters can be specified. Object settings can be modified at any time: double- clicking the relevant row opens again the **New <object>** window for edition.

Note: Expand the window if the newly created objects are not visible.

Timers

Prior to creating timers, their usage (USE_TIMERS definition) must be enabled in the **software timer definitions section** of the **Configuration parameters** tab. In the same section, timer task priority, queue length and stack depth can be also configured.

The timer can be created to be one-shot (run once) or auto-reload (periodic). The timer name and the corresponding callback function name must be specified. It is up to the user to fill the callback function code and to specify the timer period (time between the timer being started and its callback function being executed) when calling the CMSIS-RTOS osTimerStart function.

Mutexes / Semaphores

Prior to creating mutexes, recursive mutexes and counting semaphores, their usage (USE_MUTEXES, USE_RECURSIVE_MUTEXES,

USE_COUNTING_SEMAPHORES definitions) must be enabled within the **Kernel** settings section of the **Configuration parameters** tab.

The following code snippet shows the generated code corresponding to Figure 97.

```
/* Create the semaphores(s) */
/* definition and creation of myBinarySem01 */
osSemaphoreDef(myBinarySem01);
myBinarySem01Handle = osSemaphoreCreate(osSemaphore(myBinarySem01), 1);
/* definition and creation of myCountingSem01 */
osSemaphoreDef(myCountingSem01);
```

```
myCountingSem01Handle = osSemaphoreCreate(osSemaphore(myCountingSem01),
7);
```



```
/* Create the timer(s) */
  /* definition and creation of myTimer01 */
 osTimerDef(myTimer01, Callback01);
 myTimer01Handle = osTimerCreate(osTimer(myTimer01), osTimerPeriodic,
NULL);
  /* definition and creation of myTimer02 */
 osTimerDef(myTimer02, Callback02);
 myTimer02Handle = osTimerCreate(osTimer(myTimer02), osTimerOnce, NULL);
  /* Create the mutex(es) */
  /* definition and creation of myMutex01 */
 osMutexDef(myMutex01);
 myMutex01Handle = osMutexCreate(osMutex(myMutex01));
  /* Create the recursive mutex(es) */
  /* definition and creation of myRecursiveMutex01 */
 osMutexDef(myRecursiveMutex01);
 myRecursiveMutex01Handle =
osRecursiveMutexCreate(osMutex(myRecursiveMutex01));
```

FreeRTOS heap usage

The **FreeRTOS Heap usage** tab displays the heap currently used and compares it to the TOTAL_HEAP_SIZE parameter set in the **Config Parameters** tab. When the total heap used crosses the TOTAL_HEAP_SIZE maximum threshold, it is shown in fuchsia and a cross of the same color appears on the tab (see *Figure 98*).

Figure 98. FreeRTOS heap usage

	Configuration
Reset Configuration	
🥝 User Constants 🛛 📀 Tasks and Queues 🛛 📀 T	
Config parameters	😔 Include parameters
	21 🛅 🚺
 Summary 	
IEAP STILL AVAILABLE	0 Bytes
TOTAL HEAP USED	36996 Bytes
Total amount for tasks	33328 Bytes
Total amount for queues	3396 Bytes
Total amount for timers	96 Bytes
Total amount for mutexes and semaphores	176 Bytes
 FreeRTOS tasks 	
Idle task (FreeRTOS internal)	624 Bytes
Timer service task (FreeRTOS internal) defaultTask	1136 Bytes 624 Bytes
Task A	15472 Bytes
mvTask B	15472 Bytes
IIIyrask_D	13472 Dytes
TOTAL HEAP USED	
fotal amount of the heap used by known objects (user ob	, internal freertos objects)
VARNING	
Current computed value is greater than the configTOTAL_	
	HEAP_SIZE, remove/adjust some defined objects (tasks, queues, timers, mutexes, semaphores) or change
he Memory Management scheme to heap_3	
More about FreeRTOS Heap:	
	I) to allocate memory for tasks, queues, timers, semaphores, mutexes and when dynamically creating
rearches a region or memory called freap (into the	to anotate memory for tasks, queues, timers, semaphores, mutexes and when dynamically creating



4.5.16 Setting HAL timebase source

By default, the STM32Cube HAL is built around a unique timebase source, the Arm[®] Cortex[®] system timer (SysTick).

However, HAL-timebase related functions are defined as weak, so that they can be overloaded to use another hardware timebase source. This is strongly recommended when the application uses an RTOS, since this middleware has full control on the SysTick configuration (tick and priority) and most RTOSs force the SysTick priority to be the lowest.

Using the SysTick remains acceptable if the application respects the HAL programming model, that is, does not perform any call to HAL timebase services within an Interrupt Service Request context (no dead lock issue).

To change the HAL timebase source, go to the SYS peripheral in the **Component list** panel and select a clock among the available sources, such as SysTick, TIM1, TIM2 (see *Figure 99*).

Pinout & Configuration	Clock Configuration	
	Additional Softwares	\sim
Options Q ~	SYS Mode and Configuration	
Categories A->Z	Mode	
÷	Debug Disable	\sim
RTC SAI1	System Wake-Up	
SDIO	Timebase Source SysTick	\sim
SPI1	SysTick	
SPI2 SPI3	TIM1	
SPI4	TIM2	
✓ SYS	TIM3	
TIM1	TIM4	_
TIM2	TIM5	
TIM3	TIM6	
TIM4	TIM7	

Figure 99. Selecting a HAL timebase source (STM32F407 example)

When used as timebase source, a given peripheral is grayed and can no longer be selected (see *Figure 100*).



Figure 100. TIM1 selected as HAL timebase source	
SYS Mode and Configuration	
Mode	
Debug Disable	\sim
System Wake-Up	
Timebase Source TIM1	\sim

As illustrated in the following examples, the selection of the HAL timebase source and the use of FreeRTOS influence the generated code.

Example of configuration using SysTick without FreeRTOS

As illustrated in *Figure 101*, the SysTick priority is set to 0 (High) when using the SysTick without FreeRTOS.

🛛 NVIC 🛛 🤇	Code generation			
Priority Group	4 bits for pre-emption priority 0 bits for $ \sim $	🗌 🗖 Sort	by Premption Priority	and Sub Priority
Search	Search (CrtI+F) ③	C Sho	w only enabled interru	pts
	NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
Non maskable i	nterrupt	\checkmark	0	0
Hard fault interru	upt	\checkmark	0	0
Memory manage	ement fault	\checkmark	0	0
Pre-fetch fault, n	nemory access fault	\checkmark	0	0
Undefined instru	uction or illegal state	\checkmark	0	0
System service	call via SWI instruction	\checkmark	0	0
Debug monitor		\checkmark	0	0
Pendable reque	st for system service	\checkmark	0	0
Time base: Syst	em tick timer	\checkmark	0	0
PVD interrupt thr	ough EXTI line 16		0	0
Flash global inte	errupt		0	0
RCC global inte	rrupt		0	0
SPI2 global inter	rrupt		0	0
TIM6 global inter	rrupt, DAC1 and DAC2 underrun error interrupts		0	0
FPU global inter	rupt		0	0

Figure 101. NVIC settings when using SysTick as HAL timebase, no FreeRTOS

Interrupt priorities (in main.c) and handler code (in stm32f4xx_it.c) are generated accordingly:

- main.c file
- /* SysTick_IRQn interrupt configuration */
 HAL_NVIC_SetPriority(SysTick_IRQn, 0, 0);



```
stm32f4xx_it.c file
/**
* @brief This function handles System tick timer.
*/
void SysTick_Handler(void)
{
    /* USER CODE BEGIN SysTick_IRQn 0 */
    /* USER CODE END SysTick_IRQn 0 */
    HAL_IncTick();
    HAL_SYSTICK_IRQHandler();
    /* USER CODE BEGIN SysTick_IRQn 1 */
    /* USER CODE END SysTick_IRQn 1 */
}
```

Example of configuration using SysTick and FreeRTOS

As illustrated in *Figure 102*, the SysTick priority is set to 15 (Low) when using the SysTick with FreeRTOS.

Figure 102	. NVIC settings	when using	FreeRTOS and	SysTick as	HAL timebase
------------	-----------------	------------	--------------	------------	--------------

🛛 🛇 NVIC	Code generation			
Priority Group	4 bits for pre-emption priority 0 bits for ~	Sort	by Premption Priority	and Sub Priority
Search	Search (CrtI+F) ③) 🗖 Sho	w only enabled interrup	ots
	NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
Non maskable i	nterrupt	\checkmark	0	0
Hard fault interru	ipt	\checkmark	0	0
Memory manage	ement fault	\checkmark	0	0
Pre-fetch fault, m	nemory access fault	\checkmark	0	0
Undefined instru	iction or illegal state	\checkmark	0	0
System service (call via SWI instruction	\checkmark	0	0
Debug monitor		\checkmark	0	0
Pendable reque	st for system service	\checkmark	0	0
Time base: Syst	em tick timer	\checkmark	15 ~	0
PVD interrupt thr	ough EXTI line 16		0	0
Flash global inte	errupt		0	0
RCC global inte	rrupt		0	0
SPI2 global inter	rupt		0	0
TIM6 global inter	rupt, DAC1 and DAC2 underrun error interrupts		0	0
FPU global inter	rupt		0	0

As shown in the following code snippets, the SysTick interrupt handler is updated to use CMSIS-os osSystickHandler function.

```
main.c file
/* SysTick_IRQn interrupt configuration */
HAL_NVIC_SetPriority(SysTick_IRQn, 15, 0);
```



```
stm32f4xx_it.c file
/**
* @brief This function handles System tick timer.
*/
void SysTick_Handler(void)
{
    /* USER CODE BEGIN SysTick_IRQn 0 */
    /* USER CODE END SysTick_IRQn 0 */
    HAL_IncTick();
    osSystickHandler();
    /* USER CODE BEGIN SysTick_IRQn 1 */
    /* USER CODE END SysTick_IRQn 1 */
}
```

Example of configuration using TIM2 as HAL timebase source

When TIM2 is used as HAL timebase source, a new stm32f4xx_hal_timebase_TIM.c file is generated to overload the HAL timebase related functions, including the HAL_InitTick function that configures the TIM2 as the HAL time-base source.

The priority of TIM2 timebase interrupts is set to 0 (High). The SysTick priority is set to 15 (Low) if FreeRTOS is used, otherwise is set to 0 (High).

📀 NVIC 🛛 📀 Code generation			
Priority Group $\[4\]$ bits for pre-emption priority 0 bits for $\[4\]$	Sort	by Premption Priority	and Sub Priority
Search (CrtI+F) (3 (2)	C Sho	w only enabled interru	ots
NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
Non maskable interrupt	\checkmark	0	0
Hard fault interrupt	\checkmark	0	0
Memory management fault	\checkmark	0	0
Pre-fetch fault, memory access fault	\checkmark	0	0
Undefined instruction or illegal state	\checkmark	0	0
System service call via SWI instruction	\checkmark	0	0
Debug monitor	\checkmark	0	0
Pendable request for system service	\checkmark	0	0
System tick timer	\checkmark	15	0
PVD interrupt through EXTI line 16		0	0
Flash global interrupt		0	0
RCC global interrupt		0	0
Time base: TIM2 global interrupt	\checkmark	0	0
SPI2 global interrupt		0	0
TIM6 global interrupt, DAC1 and DAC2 underrun error interrupts		0	0
FPU global interrupt		0	0

Figure 103. NVIC settings when using FreeRTOS and TIM2 as HAL timebase

The stm32f4xx_it.c file is generated accordingly:

- SysTick_Handler calls osSystickHandler when FreeRTOS is used, otherwise it calls HAL_SYSTICK_IRQHandler.
- TIM2_IRQHandler is generated to handle TIM2 global interrupt.



4.6 Pinout & Configuration view for STM32 MPUs

For STM32MPUs the **Pinout & Configuration** view allows the user to:

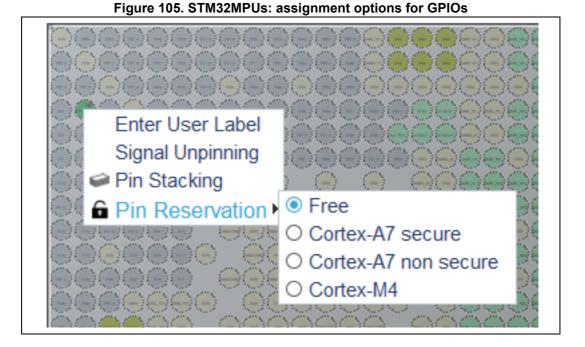
- assign components to one or several run time contexts
- configure peripherals as boot devices
- select the peripherals to be managed by boot loaders
- assign GPIOs to one runtime (see *Figure 105*).

These possibilities are offered in two different panels (see Figure 104):

- from the component tree panel, listing all supported peripherals and middleware (the "Show contexts" option must be enabled)
- from each component mode panel, opened by clicking the component name.

STM32CubeMX Untitled*: STM	M32MP151CAL	Эх				
STM32	File		Window	Help	🚳 🚺	• 🤟 🔀 🎝
Home > STM32MP151CAE	🛛 🔪 Untit	led - Pinou	t & Configui	ration >	GENERATE	CODE
Pinout & Configuratio	n C	lock Con	figuration	Project N	lanager	Tools
		l Softwares		✓ Pinout		
Options Q			\sim		TIM12 Mode and Configur	ation
Show contexts			- E		Mode	
Boot ROM A	7BL A7S	A7NS	M4		me contexts:	
TIM4				Boot ROM Boot load	er A7S A71	IS Cortex-M4
TIM5				Slave Mode Disable		
TIM6				Trigger Source Disable		
TIM7						Ŭ
TIM8				Internal Clock		
n TIM12				Channel1 Disable		\sim
TIM13				Channel2 Disable		\sim
TIM14				Combined Channels Disa	ble	\sim
TIM15						
TIM16					Configuration	

Figure 104. STM32MPUs boot devices and runtime contexts





4.6.1 Run time configuration

On these multi-core (Arm[®] Cortex[®]-A7 dual-core and Cortex-[®]M4) and multi-firmware devices, each firmware is executing on one of the cores. The association between firmware and core defines a runtime context. Three runtime contexts are available:

- 1. Cortex-A7 Non Secure running the Linux kernel
- 2. Cortex-A7 Secure running the SP_min
- 3. Cortex-M4 running the STM32Cube firmware.

Assigning a component to a runtime context means specifying which context(s) will control the component at runtime. Assignments to a Cortex-A7 context are reflected in the device tree code generation, while assignments to the Cortex-M4 context are reflected in STM32Cube based C code generation (refer to code generation sections for more details).

The component assignment to a context is done in the context dedicated column.

4.6.2 Boot stages configuration

Boot ROM peripherals selection

Several execution stages are needed by the microprocessor to be up and running.

The binary code embedded in the ROM is the first to be executed. It uses a default configuration to initialize the clock tree and all peripherals involved in the boot detection.

The peripherals managed by the boot ROM program can be selected as boot devices. This choice is done in the Boot ROM column (see *Figure 106*).

STM32CubeMX Untitled*: STM32N	1P151CADx						
STM32 CubeMX	File	Window	Help	(19)	F 🖸	♥★	57
Home > STM32MP151CADx	→ Untitled -	Pinout & Config	uration >	GEI	NERATE CODE		
Pinout & Configuration	Cloc	Configuration	n Pro	ject Manager		Tools	
ļ A			✓ Pinout				
Options Q		~		FMC Mode and	Configuration		1
Categories A->Z				Mod	e		
Boot ROM A7BL DTS ETH1	A7S A	7NS M4	Boot time: Boot ROM	Runtime contexts: loot loader A7S	A7NS	Cortex-M4	
ETZPC			V A NAND Flash 1				- H
A FMC	_		Chip Select N	CE2		\sim	- H
FREERTOS			Data/Address	8 bits		\sim	- H
GIC	\checkmark	✓	VWAIT Re	ady/Busy			- H
GPIO 🔽 🗌							- U
E HASH1 HASH2							

Figure 106. Select peripherals as boot devices

When a peripheral is set as boot device, it imposes a specific pinout: some signals have to be mapped exclusively on pins visible by the boot ROM and only these signals/pins are taken into account by the boot ROM program.

When a functional mode of a ROM-bootable peripheral is set, the pinout linked to this mode is the same of that for a runtime context except for the signals imposed on specific pins by the boot ROM code.

UM1718 Rev 47



During the boot step (boot ROM code execution), the peripheral is running only with the sub-set of bootable signals and pins. After boot, during runtime, the peripheral runs with all signals necessary to the selected functional mode.

Boot loader (A7 FSBL) peripherals selection

When the board starts, the launching of each of the Cortex-A7 runtime contexts (Secure and Non Secure) on which a firmware executes (for example Linux kernel for Cortex-A7 Non Secure) preceded by an early boot execution stage, that is before U-Boot relocation in DDR.

The Boot loader (A7 FSBL) column is used to define which devices can be managed during this Boot loader stage.

This assignment are reflected in the different device trees generated (refer to code generation sections for more details).

4.7 **RIF configuration**

Some STM32 products, like the STM32MP25x, have a special feature called RIF (resource isolation framework), used as a security guard for the their peripherals and memory. RIF decides which blocks the CPU can use, and manages the support systems for them. For details on how RIF operates, visit the STM32MPU Wiki website.

When the user sets up RIF in the STM32CubeMX program, the basic steps are the same, independently from the used device, even if there are several available options.

4.7.1 Configuration approach

In STM32CubeMX, the way the RIF keeps blocks safe is controlled by how user sets up them by software. When the settings change, STM32CubeMX checks them, translates what the user has done, and shows the updates in a special section called RIF panel.

User cannot set the access level or their special functions only by using software settings. This is managed by the main, trusted part of the software, with special access (Privileged mode). If there is need to use a setup where some blocks are used by less trusted software without special access (non-Privileged mode), user can make the changes in the RIF panel.

Blocks that user cannot set up with a software tool (like some memory areas in the STM32MP25), can still be protected by using the RIF panel.

The RIF panel is designed to display the security settings for the whole microcontroller (SOC level) in a way is similar to what detailed in the reference manual.

In the final steps:

- The system creates a set of rules (RIF configuration) that determine who is allowed to use different parts of the microcontroller. These rules are written out as source code.
- The code that sets up the microcontroller hardware blocks (like memory and peripherals) is made to match the software settings and the access rules user has set. This ensures that everything works together, without conflicts.

4.7.2 RIF global configurations

The RIF configuration panel can be conveniently accessed through the IP panel itself. This is because the RIF is integrated as a regular security IP within the STM32CubeMX system.



RIF global configurations for STM32MP2

The RIF configuration panel contains only one configuration, named Default configuration. The user can either lock down unused resources to prevent access, or leave them open for unrestricted use.

Two choices are proposed:

- No access: blocks the use of the resource. No one can read from it, write to it, or use it in any way.
- Full access: the resource can be used, it can be read and written without any restriction.

Pinout & Configu	ration	Clock Cor	Pinout & Configu	ration	Cloc
RIF Configuration	Default Configuration	n	RIF Configuration	Default Configuration	

Figure 107. Default configuration

For the STM32MP2 series, there is a unique RIF configuration provided to the user. The radio button is disabled and indicates "No access" (see *Figure 108*): the user cannot read, write, or use it in any form.



TM32 CubeMX	File	Window	Help	🐥 🚳 📑 🕒) X () Q 🔆 ភ
Home 🔰 STM32MP251AAKx	➢ Untitled - RIF	>		GENER	ATE CODE
Pinout & Configuration	Clock Config	guration	RIF	Project Manager	Tools
RIF Configuration	Default Configuration				
Peripherals (RISUP)					
Domains (RIMU)					
External memories (RISAF)					

RIF global configurations for STM32N6

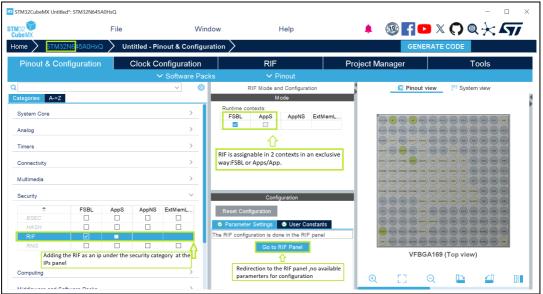
For the STM32N6 series, the RIF default configuration is not supported.



ligure 105.	Kii coning		tension				- 361163
STM32CubeMX Untitled: STM32MP25	i1AAlx						- 🗆 ×
STM32	File	Window		Help	🌲 🐵	f 🕒 X 🔿 🤇	Q 🔆 ភ
Home > STM32MP251AA	x 🔪 Untitled - Pinc	out & Configuration	\rangle			GENERATE CODE	
Pinout & Configuration	n Clock Co	nfiguration	RIF	2	Project Manager		Tools
		Software Packs		Pinout			
Q		~	٢		RIF Mode and Configuration	1	🚺 💭 📁 Syster
Categories A->Z					Mode		
System Core			>	Boot time: A35 ROM A3		435NS M33NS	
Analog			>	(TF	-A BL2 (OP-TEE) (U-Boot)	(Linux) (Cube)	
Timers			>		企		
Connectivity			>	following contex	IP2 series the RIF is assigned only tts A35S(TF-A BL2)and A35S(OP-T	EE) by default(the	
Multimedia			×	assignment is fo	rced so that the user can't modif	y it)	
Security			~		Configuration		
\$ A35 ROM	A35S A35S (TF-A_BL2) (OP-TEE)	A35NS A35NS (U-Boot) (Linux)	M33NS (Cube)	Reset Configu	ration		FBGA436 (Top viev
V BSEC				Parameter S	ettings 🔮 User Constants		
HASH				The RIF configura	tion is done in the RIF panel		
RIF RNG					Go to RIF Panel		
RNG				_			
Computing	is added as a security IP i	n the IPs panel	>		on button to the the RIF Configura arameters to set here	ation	Q []

Figure 109. RIF configuration extension in IPs panel for the STM32MP2 series







4.7.3 **Peripherals protection**

Microcontroller peripherals can be classified by their function or by how they are protected:

- Sorted by function:
 - Standard peripherals: do processing and can interact with other devices (such as I2C and UART).
 - Service peripherals: do processing but do not interact with other devices (such as CRYP and HASH).
 - System peripherals: provide services to other peripherals (such as RCC, GPIO, DMA).
- Sorted by protection scheme:
 - The whole peripheral is protected (non-RIF-aware IP). Access rules are set for the whole peripheral. The RISUP subsystem manages the protection.
 - Protection by specific function (RIF-aware and pseudo-RIF-aware IP). Access rules are based on specific functions/features. The peripheral itself controls the protection. For pseudo-RIF-aware IPs, although they are RIF-aware, their feature protection is managed by the RISUP.

In STM32CubeMX, the security for the microcontroller peripherals is set through the RIF, based on the software settings. The program figures out the security rules automatically, based on which parts (IP or IP features) are assigned to various parts of the software. When a part is assigned to a software area, it must be decided who can use, who can set it up, and what is allowed to do with it.

The configuration of access rights are available within the RIF panel:

- Non RIF-aware and pseudo RIF-aware IPs: access rights are managed through the RISUP panel.
- RIF-aware IPs: access rights for these IPs are configured in the RIF-aware IPs panel.

4.7.4 **Peripheral instance protection**

Peripheral instance protection for STM32MP2

The assignment of IPs (or IP features in the case of pseudo-RIF-aware IPs) to software contexts directly determines access rights. These rights are then displayed in the RIF RISUP configuration panel, which outlines the level of protection provided by the RIF, and where advanced configurations can be specified for each peripheral instance.

The RISUP configuration panel for STM32MP2 series is composed of:

- The list of IPs and features of pseudo-RIF-aware IPs
- IP identifiers (ID), as defined in the reference manual
- IP master owners compartment Identifiers (CID) and security states
- The RIF privilege level for each IP
- The lock state for each IP



Pinout & Configu	ıration	Clock Configuration		RIF		Project Manage
	Global lock : OFF					
RIF Configuration	Peripherals	ID	CID	Secure	Privilege	Lock
	ADC12	58	-			
	ADC3	59	-			
	ADF1	55	-			
	COMBOPHY	67	-			
	CRC	109	-			
	CRYP1	96	1			
	CRYP2	97	-			
	CSI	86	1			
	DCMI_PSSI	88	-			
	DCMIPP	87	1			
	DSI CMN	81	1			
	DSI_RDFIFO	123	1			
	DSI TRIG	122	1			
	DTS	107	1	✓		
	ETH1	60	1			
	ETH2	61	-			
	ETHSW ACM CFG	71	-			
ternal memories (RISAF)	ETHSW ACM MSGBUF	72	-			
	ETHSW_DEIP	70	-			
	FDCAN	56	-			
	GICV2M	112	1	✓		
	GPU	79				
	HASH	95	1	✓		
ernal memories (RISAB)	HDP	57	1			
	12C1	41				
	12C2	42				
	12C3	43	-			
	12C3	45				
	12C4	44				
RIF-Aware IPs	12C5	45	-			
	1206	46	-			
	12C8	47	-			
	12C0 13C1	114	-			
	1301	114	-		<u> </u>	

Figure 111. RISUP configuration panel

The Lock blocks any change after boot (that is, after configuration in STM32CubeMX), to prevent software from subsequently making changes to the RIF elements.

The Local Lock defines a Lock on independent elements.

Global Lock defines a Lock on a set of elements. By default, it is OFF. O Global lock : OFF

Configuration example

Figure 112 shows on left hand side the IP allocation per software context, and, on the right-hand side the equivalent in the RISUP configuration panel.

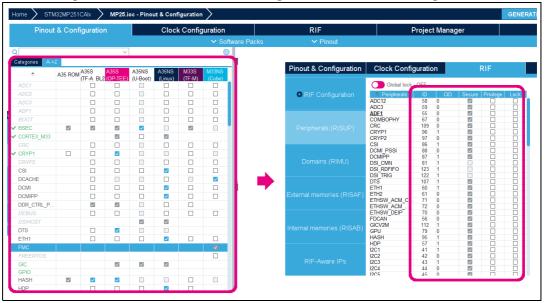


Figure 112. Software context configuration vs. RISUP configuration



For example, if the user sets ADC3 to Cortex-A35 secure context, on the RIF panel ADC3 is allocated to CID 1, and set secure. The user can then configure the privilege and the lock. If a peripheral is set in two contexts (Cortex-A35 and Cortex-M33), the allocated CID is 1&2.

Pinout & Co	nfigura	tion	Clock (Configu	iration		Pinout & Configuration	Clock Config	uratio	n		RIF		Project Mana
														,
Q	~					0		Global lock	: OFF					
Categories A-	>7						Peripherals (RISUP)	Peripherals	ID	CID	Secure	Privilege	Lock	
Gategories			_			_		ADC12	58	-				
÷	A35 R	A35S	A35S	A35NS	A35NS			ADC3	59	1	×			
		(TF-A_B	OP-TEE	(U-Boot)		(Cube)		ADF1	55	-				
ADC1								COMBOPHY	67	-				
ADC2						8	Domains (RIMU)	CRC	109	-				
ADC3		\checkmark						CRYP1	96			님		
ADF1								CRYP2	97	-		님	H	
BOOT								CSI	86 88			H	H	
BSEC	V							DCMI_PSSI DCMIPP	87	-		H	H	
		E.B					External memories (RISAF)	DSI_CMN	81	1		H	H	
✓ CORTEX_M33		_			1	_		DSI_RDFIFO	123	1		П	Н	
								DSI_TRIG	122			- D	ň	
CRYP1		~			~			DTS	107	1	~			
								ETH1	60	1				
CSI					~		Internal memories (RISAB)	ETH2	61					
DCACHE								ETHSW_ACM	71	1				
								ETHSW_ACM	72	1				
DCMI					V			ETHSW_DEIP	70	1				
DCMIPP					~			FDCAN	56					
DDR_CTRL		\checkmark	\checkmark				RIF-Aware IPs	GICV2M	112	1	\checkmark			
DEBUG								GPU	79			님	님	
DSIHOST				V	V			HASH HDP	95 57	1				
DTS			V					HUP	57	1		1		

Figure 113. Example of IP assignment to one context and result in RISUP

If the user selects an IP in a Cortex-A35 Non Secure context and a Cortex-M33 Non Secure context, the CID is set to 1&2 and the Secure column is unticked, as shown in *Figure 114*.

If the IP is not assigned to any software context, the CID column contain a –, and the Secure column is unticked (in the case of Full Access).

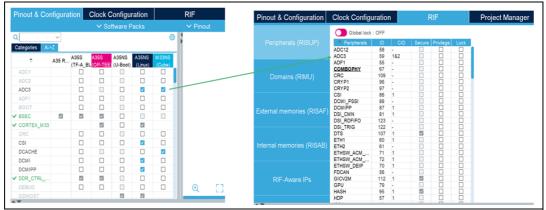


Figure 114. Example of IP assignment to two contexts and result in RISUP

The RISUP table contains entries for peripherals not supported by the MPU, such as CRYP1, CRYP2 on STM32MP251DAIx. These entries have the CID and lock cells permanently set and cannot be modified. The RISUP table is primarily read-only, with modifications limited to the MPU configuration.



STM32CubeMX Untitled*: STM32MP2	51AAIx									
Cube MX	File	Wind	low	н	elp	🔺 💿 🗗 🗵 🌾 🖓 🔍 🛧				
Home > STM32MP251AAIx	Untitled - RIF						GENI	ERATE CODE		
Pinout & Configuration	Clock Config	uration		RIF		Pro	ject Manager	Tools		
	Global lock : OFF									
RIF Configuration	Peripherals	ID	CID	Secure	Privilege	Lock				
	ADC12	58	-		 ✓ 					
	ADC3	59								
	ADF1	55								
	COMBOPHY	67			 ✓ 					
	CRC	109	-							
	CRYP1	96				\checkmark				
	CRYP2	97	-			\checkmark				
Domains (RIMU)	CSI	86								
	DCMI_PSSI	88								
	DCMIPP	87	-							
	DSI_CMN	81	1			\checkmark				
External memories (RISAF)	DSI_RDFIFO	123	1			\checkmark				
	DSI_TRIG	122	1			\checkmark				
	DTS	107			V					
	ETH1	60	-							
Internal memories (RISAB)	ETH2	61				\checkmark				
	ETHSW_ACM_CFG	71	-			\checkmark				
	ETHSW_ACM_MSGBUF	72				\checkmark				
	ETHSW_DEIP	70				\checkmark				
RIF-Aware IPs	FDCAN	56				\checkmark				
	GICV2M	112	1							
	GPU	79	-			\checkmark				

Figure 115. Lock and privilege in RISUP table

Note: Some IPs in RISUP do not exist in peripheral list, and some IPs are coupled. They show-up in the Peripheral column as one. As an example, ADC1 and ADC2 are shown as ADC12, ICACHE and DCACHE are shown as ICACHE_DCACHE.

The features of the pseudo RIF-aware IPs are also visible in the RISUP table, as shown in *Figure 116*.

		vinout &	Config	guration					Clock Conf	guration				RIF	
												✓ Software	Packs	✓ Pine	but
۲.				~			٢				LTDC Mode	and Configuration			
Categories 🗛	->Z											Mode			
	A35 ROM	A35S		A35NS	A35NS	M33S	M33NS	Boot time:		Runtime c					
		(TF-A BL	OP-TEE	(U-Boot)	(Linux)	(TF-M)	(Cube)	A35 ROM	A35S (TF-A BL	A35S (OP-TE		35NS J-Boot)	A35NS (Linux)	M33S (TF-M)	M33NS (Cube)
									(IF-A BL	2) (OP-11	-==) ((J-Boot)	(Linux)	(TF-W)	(Cube)
								Diada Tan							
								Display Type Di	sable						
LTDC		1									Cor	figuration			
MDF1								170.0 ()							
NVIC_NS								LTDC features				_			
NVIC_S						v		Features	A35 ROM	A35S	A35S	A35NS	A35NS	M33S	M33NS
		~		~	~					(TF-A_BL2)	(OP-TEE)	(U-Boot)	(Linux)	(TF-M)	(Cube)
							V	LTDC_CMN LTDC_L0L1				✓	✓	8	
OCTOSPIM			~					LTDC_L0L1			~		<u>~</u>		
OPENAMP								LTDC_ROT				 Image: A set of the set of the	<u></u>		
OTFDEC1			~												
		_	_	_	_	_									



Peripheral instance protection for STM32N6

The RISUP panel for STM32N6 series (*Figure 117*) does not have the CID column because the STM32N6 contains a single M55 core. Consequently, the CID column is unnecessary as all peripherals in the STM32N6 are allocated by default to CID1, and the indication of RIF unused with a – is no longer available.



The panel is composed of five columns:

- Peripherals column: list of pseudo-RIF-aware IPs and features
- ID column: IP identifiers (as defined in the reference manual)
- Secure column: security state for each IP
- Privilege column: privilege level for each IP
- Lock column: lock state for each IP

Figure 117. Peripherals (RISUP) panel for the STM32N6 series

Pinout & Configuration	Clock Config	R	IF	Proj	ect Manager	Tools	
	Global lock : OFF						
Peripherals (RISUP)	Peripherals	ID	Secure	Privilege	Lock		
	NPU	106					
	OTG1_HS	56	1	v			
	OTG2_HS	57	1	~			
	PKA	77					
Domains (RIMU)	RNG	76					
	SAES	78					
	SAI1	4					
Interrupts (IAC)	SAI2	8					
	SDMMC1	53					
	SDMMC2	54					
	SPDIFRX	61					
	SPI1/I2S1	0					
	SPI2/I2S2	1					
xternal memories (RISAF)	SP13/12S3	2					
	SPI4	3					
	SPI5	4					
	SP16/12S6	5					
	SYSCFG	62					
RIF-Aware IPs	TIM 1	27					
	TIM 10	36					
	TIM11	37					

If the user chooses to create a new project as full secure (*Figure 118*), the column "Secure" is hidden, as all peripherals are working in secure mode (*Figure 119*).

Figure 118. Creation of a new project for the STM32N6 series - Secure projects





	Global lock : OFF				
	Peripherals	ID	Privilege	Lock	
	ADC12	64			
	ADF1	51			
	CRC	67			
	CRYP1	80			
	CSI2HOST	92			
	DCMI	94			
	DCMIPP	93			
	DMA2D	101			
	ETH1	60			
	FDCAN1	26			
	FMC	90			
	GFXMMU	100			
	GFXTIM	45			
	GPU	99			
	HASH	79			
	1201	9			
	12C2	10			
	12C3	11			
	12C4	12			
	13C1	13			
xternal memories (RISAF)	13C2	14			
sternarmemones (KIOAL)	ICACHE	98			
	IWDG	68			
	JPEG	96			
	LPTIM 1	46			
	LPTIM2	47			
	LPTIM3	48			
	LPTIM4	49			
	LPTIM5	50			
	LPUART1	25			
	LTDC_CMN	102			

Figure 119. Peripherals (RISUP) panel for the STM32N6 series - Secure projects

4.7.5 IP feature protection

In certain scenarios, feature assignment can depend upon the feature assignment of another IP within the system.

Feature assignments are managed through the Features Configuration panel associated with each RIF-aware IP. For non-RIF-aware IPs, although access rights are inferred from the feature-to-software context assignments, they are documented in the IP sub-panel found within the RIF-aware IPs configuration panel.

The features assignment is combined with the IP modes:

- The features define which functionalities can be accessed, by which firmware
- The modes define which features are effectively used and initialized and open access to initialization parameters

The initialization parameters set depend on the corresponding feature assignment:

- HAL parameters when feature is assigned to a Cube firmware
- No parameters for firmware initialized via a device tree system (such as an OpenSTLinux firmware)

Configuration example

In the following example, the FMC IP is configured to work as a RIF-aware IP:

- Click on FMC IP in Pinout & Configuration panel
- The FMC related features is displayed on the configuration panel on the right-hand side
- Select A35S (OP-TEE) for the features FMC_CFGR
- In the FMC Mode and Configuration panel, pick "NE1" in the "Chip select" drop down
- In the Configuration panel, three tabs are displayed (Parameter Settings, Features, GPIO Settings)



Pinout & (1.	Clock	Configura	tion		RIF			Proj	ect Manag	jer		Т	ools	
					✓ Software	Packs	🗸 Pino	ut								
		. ×.				0			FMC M	ode and Config	puration			Pinout	view [**	System vie
ategories A->Z							2			Mode						
•	A35 ROM	A35S (TF-A_BL2)	A35S (OP-TEE)	A35NS (U-Boot)	A35NS (Linux)	M33NS (Cube)	Boot time: A35 ROM	A355	Runtime A35S	A35			M 33NS			
				21			,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	(TF-A_E		EE) (U-B	oot) (too	Linux)	(Cube)			
		0	0						22							
			0			0	Y NOR Flash		ROM/LCD 1							
						0	Chip Sel	ect NE1					~			
MC			8			1	Memory	type Disable					~			
HC .			55		22	1	Address	Disable								
PIO							100000000000000000000000000000000000000	pister Select								
HASH	22								150.019					Lot up		and all
IDP							Data Dis						194	ALC: NO		2 2 2 2 C
HPDMA1		121	51	23	2	5	Data/Adi	dress Disable				□ M	lax: Disable			
HPDMA2		53	22	23	23	23	Clock D	sable						133		10.02
HPDMA3		22	23	22	22	22										
										Configuration						1,1222
							Reset Confi	ouration						1853		A 100
							International Statements	Contraction of the second			-10			100.00	GA424 (Tor	100 1000
204							Parameter	Settings	Features C	GPIO Settin	25			VFB	GA424 (Top	o view)
200							FMC features	Ú.								
				EJ					A355	A35S	A35NS	A35NS	MISSING			
208							Features	A35 ROM	(TF-A_BL2)		(U-Boot)	(Linux)	(Cube)			
							FMC_CFGR									
							FMC_NOR/									
							FMC_NOR/ FMC_NOR/		8	8		8	H			
							FMC_NOR/		ö			ö				
							FMC_NAND		Ó							
					0											
						0								Q	53	Q
		0	0		0	0								A		~

Figure 120. FMC configuration

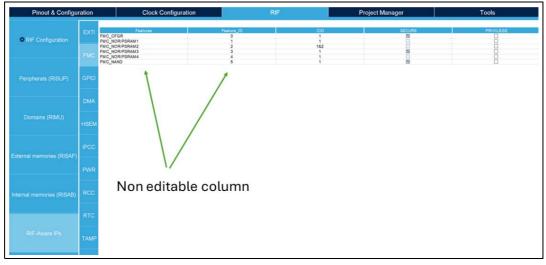
Configuring FMC in a RIF panel:

- Click on the RIF tab
- Select the RIF aware IP tab on the left-hand side
- Choose FMC

Each feature can be configured as secure or privileged.

- The CID column represents the hardware context
- The security column comes from the security of software context
- The privilege column is set to false by default

Figure 121. RIF FMC panel





ſ

Home	STI	M32MP2	251CAlx	>_м	P25.ioc	- Pinout	& Confi	guration	\rangle							
	Pinou	t & Co	onfigura	ation			Clock	Configu	ration			RIF			ĺ	Project Man
									✓ Software	Packs		✓ Pir				
Q					~			۲				RTC Mode an	nd Configurati	on		
Categor	ies 🗛	≻Z										M	ode			
-	,	A35 R	A35S	A35S (OP-TEE	A35NS (U-Boot)	A35NS (Lipux)		M33NS (Cube)	Boot time	0.350	Runtir A355	ne contexts:	5NS A	35NS	M330	1122110
PCIE			10				(11 -141)	(6006)	A35 ROI	M (TF-A I				J5NS Jinux)	M33S (TF-M)	M33NS (Cube)
PKA		\checkmark	~	~						(()	(0000)
PSSI									Activat	e Clock Sou	ITCR					
PWR		\checkmark						1	_	e Calendar						
RCC			\checkmark	\checkmark	5	\mathbf{V}	\$	1								
									Alarm A D							~
RNG		~	~	~				_	Alarm B D	isable						\sim
RTC									🗌 Timest	lamp						
SAES SAI1		\checkmark							WakeUp [Disable						\sim
									Calibration	Disable						~
										nce clock de	tection					
SAI4										nce clock de	rection					
SDM	/C1								_			Confi	guration			
												Com	gulation			
									RTC features	5						
											A35S	A35S	A35NS	A35NS	M33S	M33NS
	FRX								Features	A35 ROM	(TF-A_BL2		(U-Boot)	(Linux)		(Cube)
SPI1									Alarm A							
SPI2									Alarm B Wakeup ti							
									Timestamp							
SPI4									Calibration							
SPI5									Initialization							

Figure 122. RTC features

Figure 123. RTC mode

Pinou	ut & Co	nfigura	tion			Clock	Configu	uration			RIF			F	Project Ma
								✓ Software	Packs		✓ Pinc				
Q				\sim			٥			RT	C Mode and	d Configuratio	n		
Categories A	>Z -										Mo	de			
	A35 R	A35S	A35S	A35NS	A35NS	M33S	M33NS	Boot time:		Runtime	contexts:				
PCIE		(TF-A BL	(OP-TEE	(U-Boot)	(Linux)	(TF-M)	(Cube)	A35 ROM	A35S	A35S	A35			M33S	M33NS
PKA	~		~						(TF-A BI	.2) (OP-TEE) (U-B	oot) (Li	nux)	(TF-M)	(Cube)
PSSI								- • • • •	01 1 0						
PWR	1						~		Clock Source	e					
RCC		~	~	~	~	~	V	Activate	Calendar						
RESMGR_U								Alarm A Int	ernal Alarm A	4					~
RNG	1	~	~					Alarm B Di	sable						~
🖉 RTC								Timesta	mp						
	1	~	~					WakeUp D							~
SAI1															
SAI2								Calibration							~
SAI3								Referen	ce clock det	ection					
SAI4	_													_	
SDMMC1											Config	uration			
								Reset Conf	iguration						
								Parameter	Sattinge	User Const	ante 🔗	Features	NVIC Set	ttinge	
SPDIFRX								- anamotor	Genniga	Casi Cona		i catalea		unga	
SPUIFRX SPI1								RTC features							
								1110 10010100							
								Features			A35S (OP-TEE)	A35NS (U-Boot)	A35NS (Linux)	M33S (TF-M)	M33NS (Cube)
SPI4								Alarm A			(OF-TEE)	(0-8000)		(TP=W)	(Cube)
								Alarm B							
SPI6								Wakeup ti							
								Timestamp Calibration							
								Initialization							



Home > ST	M32MP	251CAlx	> •	1P25.ioc	- Pinout	& Confi	guration	\rangle						
Pinou	ut & Co	onfigur	ation			Clock	Configu	ıration		F	RIF			Project Man
								✓ Software P			 Pinout 			
Q				\sim			0			RTC N	lode and Cor	figuration		
Categories A	>Z -										Mode			
	A35 R	A35S	A35S	A35NS	A35NS	M33S	M33NS	Boot time:		Runtime cont	exts:			
PCIE	A33 R	(TF-A E	BL <mark>(OP-TE</mark>	(U-Boot)	(Linux)	(TF-M)	(Cube)	A35 ROM	A35S	A35S	A35NS	A35NS	M33S	M33NS
PKA	~								(TF-A BL2)	(OP-TEE)	(U-Boot)	(Linux)	(TF-M)	(Cube)
														1
PWR	~		-		-	-		Activate C	lock Source					
RCC		~	~	V	~	~		Activate C	alendar					
RESMGR U								Alarm A Inter	al Alarm A					~
RNG	~	~	~								Configuratio			
RTC							V				Conliguratio	л		
SAES	~	~	~					Reset Configu	ration					
SAI1								Parameter S	ettings 🛛 🥥	User Constant	s 🛛 📀 Feat	ures 😔 NV	1C Settings	
SAI2								Configure the belo					g.	
SAI3								Q Search (Ctrl+I		0				0
SAI4								✓ Alarm A		0				v
SDMMC1								V Alarm A Hours			0			
SDMMC2								Minute	s		0			
SDMMC3								Secon	ls		0			
								Sub S	conds		0			
SPDIFRX									Mask Date W	/eek day	Disal			
SPI1									Mask Hours		Disal			
									Mask Minutes Mask Second		Disal			
SPI3									Mask Second Sub Second I			arm SS fields	ara maskad	
SPI4									Date Week D		Date	ann 00 lielus	are masked.	
SPI5										.,	5010			
SPI6 SPI7								Hours Hours must be b		22				
								Diagnostic:	erween o and	23.				
SPI8							L	in a second						

Figure 124. RTC parameters setting

4.7.6 Software constraints validation

When integrating software, there are specific guidelines for setting up access to integrated peripheral features, and to ensure that the software works correctly with the intended STMicroelectronics software architecture. These guidelines are recommendations, not mandatory requirements. STM32CubeMX provides a helpful feature in the Feature Assignment panel to follow these guidelines. It uses a color coding system and instructions to make the process easier.



	Pinout	& Configu	ration				Cloc	Configuration RIF	Project Manager
								✓ Software Packs	
		~	4				٢	FMC Mode and Configuration	
legories A->Z								Mode	
0	A35 ROM	A35S (TF-A BL2)	A35S (OP-TEE)	A35NS (U-Boot)	A35NS (Linux)	M33S (TF-M)	M33NS (Cube)	Boot time: Runtime contexts:	
		(IF-A BL2)	(OP-TEE)	(U-Boot)	(Linux)	(IE-M)	(Cube)	A35 ROM A35S A35S A35NS A35NS M33S M33NS (TF-A BL2) (OP-TEE) (U-Boot) (Linux) (TF-M) (Cube)	(ves) (ves) (ves,0) (ves,0) (ves,0) (ves,0)
								(IF-A BL2) (OP-TEE) (O-Boot) (LINDX) (IF-M) (CUDB)	
								> NOR Flash/PSRAM/SRAM/ROM/LCD 1	THOR D SALE D SALE D THE D COLONY DIRACE DIRACE
								> NOR Flash/PSRAM/SRAM/ROMLCD 2	
								> NOR Flash/PSRAM/SRAM/ROMLCD 3	and a more a more a more a more and more a more and
BSEC	~							> NOR Flash/PSRAM/SRAM/ROMLCD 4	
ORTEX M33	1.0	-				1.0		> NAND Flash 1	and a stor of the of th
CRC								 Nend Hash 1 	
RYP1									
RTP1									
CRYP2 CSI									Experies, Experies, Experies, Experies, Incount,
DCACHE									(HII) (HII) (HII) (HII) (HII) (HIII) (HIIII) (HIII) (HIIII) (HIIII) (HIII) (HIII) (HIIII) (HIII) (HIII) (HIII) (HIIII) (HIIII) (HIIII) (HIIII) (HIIII) (HI
CMI					V				
CMIPP								Configuration	
DR_CTRL_PHY		2							600000
								FMC features	(PG12) (PG1) (PG) (PG) (PG) (SOCO
		_	<u> </u>		V			Features A35 ROM A35S A35S A35NS A35NS M33S M33	
DTS								Peatures A35 KOM (TF-A_BL2) (OP-TEE) (U-Boot) (Linux) (TF-M) (Cut	
ETH1					V				
FMC								EMC NOR/PS	to one context, then FMC CFGR should be assigned to this context.
FREERTOS			(· • • • • • • • • • • • • • • • • • • •		FMC CIORPE C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C<	
GIC			2	1	1			FMC_NORPS	
GPIO									
HASH	~	2	2						
HDP					V				
HPDMA1		1	2	1	1	V			
HPDMA2		1	2	1	1	V			2002 2002 10000 10000 (F2) (F2) (F2) (F2)
HPDMA3		1	2	2	1	1	2		
201					V			Color-coding system and instructions	
								~ ,	
13C4									
CACHE							2		
IPCC1			V						Q [] Q
WDG1	~	~	~						~ LJ ~

Figure 125. Color coding system and instructions

4.7.7 Masters configuration

Masters configurations for STM32MP2

The RIMU is one of the core components of RIF. It allows some IPs (with data transfer capabilities) to be configured as a master. It can be used to assign an IP to a security domain by defining the secure, privilege, and compartment ID (CID).

RIMU allows the user to:

- See the list of master IPs (in their default configuration) for new domain creation
- Create new domain from each masters
- Configure the RIF security level of each master
- Configure the RIF privilege level of each master
- Configure global lock

Between RIMU and RISUP there is an inheritance relationship for common IPs. This relationship allows the IP to inherit the CID, security state, and privilege state from RISUP when the user does not define its own values.



The user interface for STM32MP2 is composed of a table containing six columns:

- 1. IP name
- 2. IP id, which are unique
- 3. CID SELECTION, to select CID
- 4. Master CID, to change the CID value
- 5. Secure state, inherited from RISUP
- 6. PRIVILEGE state, when enabled

A global lock button on the top of RIMU table can be used to lock the RIMU.

Note:

The RIMU table may contain entries for peripherals not supported by the MPU. The configuration of these unsupported peripherals is possible, but the code generation process excludes any code related to them. This restriction is applicable only to MPU configurations within the RIMU table.

Demperais (RISUP)	Global lock : OFF U IP * RIMU ID > 10 6 7 0		MASTER CID	PRIVILEGE
RMM DCMIPP ETH1 ETH2 ETR Domains (RIMU) GPU	2 10 6 7		-	PRIVILEGE
ETH1 ETH2 ETR Domains (RIMU) GPU	6 7			
ETH2 ETR Domains (RIMU) GPU	7			
ETR Domains (RIMU) GPU				
Domains (RIMU) GPU	0			
Domains (RIMU) GPU		✓	1	
	9		-	
LTDC_L		✓	4	
LTDC_L		\checkmark	4	
LTDC_F		\checkmark	4	
PCIE	8		-	
SDMMC				
xternal memories (RISAF) SDMMC			-	
SDMMC			-	
USB3DF			-	
USBH VDEC	5		-	
VDEC	14		-	
nternal memories (RISAB)	15		-	

Figure 126. RIMU user interface

To define a CID for an IP, activate CID SELECTION, and then choose a value from 0 to 6, as shown in *Figure 127*.



Home 🔰 STM32MP251DAIx	A35S.ioc -				
Pinout & Configuration	Clock	Configuration	RIF		Project Mana
Peripherals (RISUP)	Global lock	RIMU ID 10	MASTER CID	SECURE	PRIVILEGE
Domains (RIMU)	ETH1 ETH2 ETR GPU LTDC_L1L2 LTDC_L3 LTDC_ROT	6 7 0 9 11 12	- - - 4 4 4		
External memories (RISAF)	PCIE SDMMC1 SDMMC2 SDMMC3 USB3DR	13 8 1 2 3 4 5	• • •		
Internal memories (RISAB)	USBH VDEC VENC	5 14 15	0 1 2 3 4		
RIF-Aware IPs			5		

Figure 127. Assigning a CID to an IP in RIMU

To change the security or privilege value for an IP, activate the appropriate CID SELECTION checkbox, as shown in *Figure 128*.

Pinout & Configuration	l Clock	Configuration	n	RIF		Project Manag
	Global lock	: OFF				
Peripherals (RISUP)	RIMU IP 单	RIMU ID	CID SELECTION	MASTER CID	SECURE	PRIVILEGE
	DCMIPP	10		1	~	
	ETH1	6				
	ETH2	7				
	ETR	0	\checkmark	1		
	GPU	9		-		
	LTDC_L1L2	11	\checkmark	4		
	LTDC_L3	12	\checkmark	4		
	LTDC ROT	13	\checkmark	4		
	PCIE	8				
	SDMMC1	1		5		
External memories (RISAF)	SDMMC2	2				
	SDMMC3	3				
	USB3DR	4				
	USBH	5		-		
	VDEC	14				
	VENC	15				
Internal memories (RISAB)						
RIF-Aware IPs						

Figure 128. Modification of the security and privilege values

The inheritance relationship between RISUP and RIMU is established and valid only if the IP is assigned to a context in the Pinout & Configuration panel.



In the context of inheritance relationships, the user cannot change the value of security and privilege if they are false in RISUP, it can only change them from true to false if they are true.

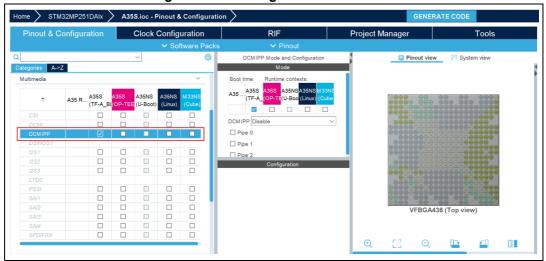


Figure 129. IP assignment to a context

Figure 130. Result in RISUP of an IP assignment to a context

Dinaut & Canfiguration	Cleak Comf			DIE		Project Manager		
Pinout & Configuration	Clock Config	guration		RIF		Proje	ct Manager	
	Global lock : OFF							
Peripherals (RISUP)	Peripherals	ID	CID	Secure	Privilege	Lock		
	ADC12	58	-					
	ADC3	59	2					
	ADF1	55	-					
	COMBOPHY	67	-					
	CRC	109	-					
Domoino (DIMLI)	CRYP1	96	-					
Domains (RIMU)	CRYP2	97	-					
	CSI	86	-					
	DCMI PSSI	88	-					
	DCMIPP	87	1	\checkmark				
	DSI_CMN	81	-					
	DSI_RDFIFO	123	-					
External memories (RISAF)	DSI_TRIG	122	-					
	DTS	107	1	\checkmark				
	ETH1	60	-					
	ETH2	61	-					
	ETHSW_ACM_CFG	71	-					
	ETHSW_ACM_MSGBUF	72	-					
Internal memories (RISAB)	ETHSW_DEIP	70	-					
	FDCAN	56	-					
	GICV2M	112	1	\checkmark				
	GPU	79	-					
	HASH	95	1	\checkmark				
	HDP	57	-					
RIF-Aware IPs	I2C1	41	-					
	12C2	42	-					
	12C3	43	-					
	12C4	44	-					
	1905	45						



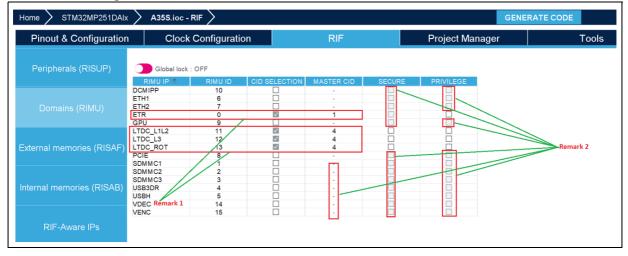
Home 🔰 STM32MP251DAI	x 🔪 A35S.ioc -						GENERATE CODE
Pinout & Configuratio	n Clock	Configurati	on	RIF		Project Manage	er T
Peripherals (RISUP)	Global lock	: OFF					
	RIMU IP 🗢 DCMIPP	RIMU ID	CID SELECTION	MASTER CID	SECURE	PRIVILEGE	
	ETHI	10		1	~		
	ETH2	7		-			
	ETR	0		- 1			
	GPU	9		-			
	LTDC_L1L2	11	✓	4			
	LTDC_L3	12	~	4			
	LTDC_ROT	13	~	4			
	PCIE	8		-			
	SDMMC1	1		-			
External memories (RISAF)	SDMMC2	2					
	SDMMC3	3					
	USB3DR	4		-			
	USBH	5		-			
	VDEC	14		-			
	VENC	15		-			

Figure 131. Inheritance of CID, state of security, and privilege from RISUP

Note that:

- Some IPs have default values
- If the user does not have the right to change values in the RIMU, these cells are greyed out
- If the user creates an STM32MP25xx project and selects the Cortex-M33 as the master, M33S (secured) the global lock button is activated by default
- When an IP is not used (CID = -) and the user checks the related CID SELECTION, a default value is assigned to the CID of the IP equal to the value of the TD CID

Figure 132. Default values for IPs and user modification restrictions





Masters configurations for STM32N6 series

The Domains (RIMU) panel for these series is composed of a table with five columns:

- RIMU IP: the name for each IP
- RIMU ID: unique for each IP
- MASTER CID: to set CID value for each IP
- SECURE: inherited from RISUP
- PRIVILEGE: privilege state if the IP is activated

Figure 133. Domains (RIMU) panel for STM32N6 series

	Global lock : OFF				
	RIMU IP 🗢	RIMU ID	MASTER CID	SECURE	PRIVILEGE
Peripherals (RISUP)	DCMIPP	9	0		
	DMA2D	8	0		
	ETH1	6	0		
	ETR	0	2		
	GPU	7	0		
	LTDC_L1	10	0		
	LTDC_L2	11	0		
	NPU	1	0		

4.7.8 Service peripherals protection

Service peripherals are special components that perform tasks or provide data for other parts of the system, and they do not have input/output ports (IOs). These peripherals are known as RIF-aware IPs, which means they are aware of the security and access framework RIF.

The user can set up these peripherals by configuring their features and modes to ensure that they are secure (protected) and ready to be used (enabled). The security settings are based on the assigned features, these settings are shown in a special area of the software called the RIF-aware IPs panel. Each type of RIF-aware IP has its own unique panel, different from the standard setup mentioned earlier. The peripherals available on STM32MP25 devices are detailed below.

HSEM

HSEM is not configurable in STM32CubeMX, but it is visible in RIF Panel to show and generate a default protection. It defines the filter access (secure and privilege) to HSEM features (16 HSEM semaphores).

As the IP is not used in the system, the protection is not configurable and forced to the "Default configuration".

HSEM contains two tables, the first represents the CPU allocation per context, the second contains the features, their "CPU Whitelist" (CPU_WL), the security states and privileges.

All the features (semaphores in case of HSEM IP) are secured and privileged, as shown in *Figure 134*.



Pinout & Configuration		Clock Configuration	on	RIF	Project Manager	Tools	
	EXTI	Resource CID Semaphores					
Configuration	FMC	Features Group 0 Semaphore 0	Feature_ID	CPU_WL	SECURE	PRIVILEGE	
		Group 0 Semaphore 0 Group 0 Semaphore 1 Group 0 Semaphore 2	1	· · ·	 	V	
		Group 0 Semaphore 3 Group 1 Semaphore 4	3 4	:			
	GPIO	Group 1 Semaphore 5 Group 1 Semaphore 6	6	:			
		Group 1 Semaphore 7 Group 2 Semaphore 8 Group 2 Semaphore 9	7 8 9	· · ·		2 2	
	DMA	Group 2 Semaphore 9 Group 2 Semaphore 10 Group 2 Semaphore 11	10 11	· · ·			
		Group 3 Semaphore 12 Group 3 Semaphore 13	12 13				
		Group 3 Semaphore 14 Group 3 Semaphore 15	14 15	:	2 2	v V	
	IPCC PWR						
	RCC						
	RTC						
	TAMP						

Figure 134. RIF HSEM panel

TAMP protection

TAMP for the STM32MP2 devices (*Figure 135*) contains two tabs:

- In the first, the user can configure the available resources, making them secure or privileged.
- In the second, the user can configure the memory zone area storing critical applications data.
 - Each zone can be resized using a dedicated panel available in the RIF configuration panel
 - Each zone is associated to a resource: the resource assignment defines the firmwares that can access a zone, and the access rights

For STM32N6 devices, the TAMP UI includes only a table that lists features along with their security and privilege statuses (see *Figure 136*). Users can select a value between 0 and 32 to define the security level (in SECURE column) for the two features backup registers protection offset (write, read/write) of the TAMP.



Pinout & Configuration		Clock	Clock Configuration		RIF		Project Manager			Tools		
RIF Configuration	EXTI	TAMP_BKP_REG										
		TAMP BKP_REG Zones	Sub-Zone Name	Start Address	Sub-Zone Size	Nb Backup Registers	CID of Resource 0		CID of Resource 1		CID of Resource 2	
	FMC						NS	S	NS	S	NS	S
		Zone1 ReadS WriteS	Zone1-RIF1	0x46010100	0x200	128				RW		
			Zone1-RIF2	0x46010300	0x0	0						RW
Peripherals (RISUP)	GPIO	Zone2 ReadNS WriteS	Zone2-RIF1	0x46010300	0x0	0			RO	RO	RO	RO
	0.10		Zone2-RIF2	0x46010300	0x0	0			RO	RO	RO	RW
			Zone3-RIF1	0x46010300	0x0	0	RO	RO	RW	RW	RO	RO
		Zone3 ReadNS WriteNS	Zone3-RIF0	0x46010300	0x0	0	RW	RW	RO	RO	RO	RO
	DMA		70 0150	0-40040000								014/
	DMA HSEM	Automaticall change	Zone3-RIF2	0x46010300 sizes	0x0	egisters = HextoDe	RO	RO	RO	RO	RW	RW
	HSEM			L	0x0	registers = HextoD The Re Sub-Zi	RO ec(Region S ead/Write a	ro ize)/4 access right e users (co	RO Ss map show	RO WS the acco	RW	o the TA
	HSEM IPCC			L	0x0	registers = HextoD The Re Sub-Zi	RO ec(Region S ead/Write a ones for th	ro ize)/4 access right e users (co	RO Ss map show	RO WS the acco	RW	o the TA
	HSEM IPCC PWR			L	0x0	registers = HextoD The Re Sub-Zi	RO ec(Region S ead/Write a ones for th	ro ize)/4 access right e users (co	RO Ss map show	RO WS the acco	RW	o the TA

Figure 135. RIF TAMP panel (STM32MP2 devices)

Figure 136. RIF TAMP panel (STM32N6 devices)

STM32CubeMX Untitled: STM32N645AI M32 ubeMX	он _х о File	Window	Help		🕸 🕇 🗖	 × () @ 🔆 /5	//
Home STM32N645A0HxC) Vntit	led - RIF >				RATE CODE	
Pinout & Configuration	CI	ock Configuration	RIF	Project	Manager	Tools	
Peripherals (RISUP)		Features Tamper protection (excluding monotor		SECURE			
		Monotonic counter 1 secure protection Backup registers write protection offs Backup registers read/write protection	et		32 0		
Domains (RIMU)	GPIO						
	HPDMA1						
Interrupts (IAC)	PWR						
External memories (RISAF)	RCC						
	RTC						
	TAMP						

IPCC configuration

In the IPCC tab, the user can configure available resources, such as Resource features 0, 1 and 2, by setting their security levels or assigning privileged status.

PWR configuration

The PWR tab allows the user to manage settings for Resource 0, Resource 1, and Resource 2, providing options to secure these resources, or grant them special privileges.



4.7.9 System peripherals (STM32MP2 and STM32N6 series)

System peripherals are components that share their functions and resources with other integrated peripherals (IPs). These system peripherals are designed to be RIF-aware, which means they are compatible with a certain security and access control system.

While these system peripherals generally use the same security setup as other RIF-aware IPs, they also have some unique features. The specific RIF configurations and what makes them different are described in the following subsections.

The RIF-aware IPs for STM32N6 are fewer than for STM32MP2, namely: EXTI1, GPDMA1, GPIO, HPDMA1, PWR, RCC, RTC, and TAMP. There is no need to display CID, as these MCUs are based on a single core.

	Features GPDMA1 channel 0	SECURE	PRIVILEGE
	GPDMA1 channel 1 GPDMA1 channel 2 GPDMA1 channel 3		
GPDMA1	GPDMA1 channel 4 GPDMA1 channel 5 GPDMA1 channel 6 GPDMA1 channel 7		
GPIO	GPDMA1 channel 8 GPDMA1 channel 9 GPDMA1 channel 10 GPDMA1 channel 11 GPDMA1 channel 12 GPDMA1 channel 13 GPDMA1 channel 13 GPDMA1 channel 14		
HPDMA1	GPDMA1 channel 15		
RTC			
TAMP			

Figure 137. RIF-aware peripherals for STM32N6 MCUs

IO configuration

There are two main types of IO (input/output) configurations:

- Alternate function IO (AF IO): used to transmit signals that the peripherals process.
- General purpose IO (GPIO) and external Interrupt IO (EXTI IO): serve general input/output functions and manage external interrupts.

For both types, security settings are automatically determined, based on their connections:

- For non-RIF-aware IPs, the security comes from the IP they are connected to
- For RIF-aware IPs, it is based on the specific features of the IP they are linked with

For GPIO and EXTI, the IO sets the security.

The assignments of IO to software contexts are displayed in the features panel specific to the GPIO IP. Additionally, the security settings (RIF protection) for these IO configurations can be found in the RIF-aware IP panel, under the GPIO sub-section



Home > S	TM32MP2	251CAlx) м	P25.ioc	- Pinout	& Confi	iguration	
		Pinout	& Con	figurati	on			Clock Configuration
								✓ Software
Q				~			٢	I2C1 Mode and Configuration
Categories 🖌	->Z							Mode
÷	A35 R	A35S	A35S	A35NS	A35NS	M33S	M33NS	Boot time: Runtime contexts:
TREEKTOS	A33 K	(TF-A BL	(OP-TEE	(U-Boot)	(Linux)	(TF-M)	(Cube)	A35 ROM A35S A35S A35NS A35NS M33NS M33NS
GIC			~	~	\checkmark			(TF-A BL2) (OP-TEE) (U-Boot) (Linux) (TF-M) (Cube)
GPIO								
HASH	\checkmark	~	~					12C 12C ~
HDP					~			Disable
HPDMA1		\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	12C
HPDMA2		\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	SMBus-Alert-mode
HPDMA3		\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	Res <mark>SMBus-two-wire-Interface</mark>
✓ I2C1								Parameter Settings GIC Settings DMA Settings GPIO Settings
12C2								
12C3								Search Signals
12C4								Search (Ctrl+F)
12C5								
12C6								Pi * Signal Pin C GPIO GPIO Maxi Retime Invert Doubl Delay Delay I PG13 I2C1 A35NS Altern No pu Low n/a n/a n/a n/a n/a n/a n/a
12C7								PI1 I2C1 A35NS Altern No pu Low n/a n/a n/a n/a n/a n/a r/a
12C8								

Figure 138. IO protection inheritance for a non-RIF-aware IP (I2C)

Figure 139. GPIO IP panel

		Pinout	& Conf	igurati	on					Clock	Configu	ration			
														`	 Software
Q				\sim			٢			G	PIO Mode a	nd Configura	ation		
Categories A	->Z										M	ode			
	A35 R	A35S (TF-A BL		A35NS (U-Boot)	A35NS (Linux)		M33NS (Cube)								
GIC			~	\checkmark	~										
GPIO											Config	guration			
HASH	~	~	~					Group By P	eripherals						~
HDP					~			⊘ GPI0	⊘ I2C	⊘ RTC (≥ NVIC				
HPDMA1		\checkmark	~	\checkmark	~	\checkmark	\checkmark	U GFIO	0120	V RIC V		• reatures			
HPDMA2		\checkmark	\checkmark	\checkmark	~	\checkmark	\checkmark	GPIO featu	ITOC						
HPDMA3		\checkmark	\checkmark	\checkmark	~	\checkmark	\checkmark	Of IO leal	lies						
I2C1					~			Features	A35 ROM	A35S	A35S	A35NS	A35NS	M33S	M33NS
12C2										· - ·		(U-Boot)	(Linux)	(TF-M)	(Cube)
/2C3								PG8 PG9							
12C4								PG10							
12C5								PG11							
12C6								PG12 PG13					✓		
I2C7								PG14							
12C8								PG15 PH2							
l2S1								PH2 PH3							
1282								PH4							
1283								PH5 PH6							
I3C1								PH7							
13C2								PH8							
								PH9 PH10							
I3C4								PH11							
ICACHE							V	PH12 PH13							
IPCC1			~					PI0							
IWDG1	~	~	~					PI1					~		
IWDG2	~			~	 Image: A second s			PI2 PI3							



UM1718

lome 🔰 STM32MP251CAlx	<u> </u>	/IP25.ioc - RIF	\geq		GENERATE	CODE
Pinout & Configuration	С	lock Config	uration	RIF	Project Manager	Tools
	EXTI	Features	Feature ID	CID	SECURE	PRIVILEGE
RIF Configuration	FMC	PG11 PG12	11 12	0	✓ ✓	
		PG13 PG14	13 14	1		
Peripherals (RISUP)		PG15 PH2 PH3	15 2 3	0	Y Y Y	
	DMA	PH3 PH4 PH5	4	0		
Domains (RIMU)	HSEM	PH6 PH7	6 7	0		
		PH8 PH9	8	0	✓ ✓	
External memories (RISAF)		PH10 PH11 PH12	10 11 12	0		
		PH12 PH13 PI0	13	0		
nternal memories (RISAB)	RCC	PI1 PI2 PI3	1 2 3	1		
		PI4 PI5	4	0	✓ ✓	
		PI6 PI7 PI8	6 7	0	Y Y Y Y	
	TAMP	P18 P19 P110	8 9 10	0		
-		DI11	11	0		

Figure 140. Inheritance in RIF GPIO panel

Figure 141. PIN reservation

	Pin	nout & C	Config	uratior	ı		Clock Config	uration			Project Ma	anager	Tools
							✓ Softw	are Packs	✓ F	Pinout			
۹					٥		GPI0 I	Mode and Configur	ation			🙁 Pinout view 🔛 System	n view
Categories 📝	A->Z ∣							Configuration					
•	Boot R			U-Boot	Linux	Group By Peripherals					~		
						GPI0 Single Mapped S	linnale 🔿 IZC						
						Configuration of Configuration	ignais • ico						
DCACHE						Search Signals							(mm) (mm) (mm) (mm) (mm) (mm)
DCMI						Search (Ctrl+F)				Show onl	/ Modified Pins	000000000000000000000000000000000000000	
												Enter User Label	
DDR_CTRL_		2	1			Pin Name * Signal on Pi PB10 n/a		PIO Pulkup/Pul lo pull-up and n	Maximum output	User Label	Modified	Signal Unpinning	
						PD0 n/a		lo pull-up and n					Free
DSIHOST													CortexA35S Secure OS
ETHI													CortexA35NS SSBL
													CortexA35NS OS
													Cortex//33S Secure OS
													Cortex///33NS
FMC			1	1	2								
GIC			~										
GPI0													
						PB10 Configuration :							◙◙⊖ ₽₽₽₽
					B								
						GPIO mode	External Inte	errupt Mode with R	ising edge trigger det	tection	~		
✓ I2C3			~			GPIO Pull-up/Pull-down	No pull-up a	nd no pull-down			~		
												TFBGA361 (Top view)	
						User Label							
/2C8													-

DMA configuration

For STM32MP2 MPUs, DMA channels can be secured to prevent unauthorized access. Each channel is treated as a security feature within the DMA IP (integrated peripheral).

The approach to protecting DMA channels is similar to how IO protection is handled:

- The settings for which software contexts can use a DMA channel are determined by the peripheral that needs the DMA service. These settings are then shown in the DMA feature panel.
- The specific protection for each DMA channel is based on these settings and is displayed in the RIF-aware IP panel, under the DMA section.

For STM32N6 MCUs, the security of DMA channels is user-defined and not automatically inherited from the IPs, see *Figure 142*.



	File		Win	fow	Help				3	∏⊡ ×∩ Q ≻4	7/
ome 🔪 STN32N545AD	no > Un	titled - Pincu	t & Configur	ution >						GENERATE CODE	
Pinout & Config	guration		Clock	Configuration		RIF		Project Manage	я	Tools	
				∀ 50	tware Packs	Y Phot					
			÷	0)#0	NA1 Mode and Config	wration.		O Phone view P System	rulew:
elegaries A2							Mada				
System Core				~		rits Intervel P.FD / 20 addres			~		
						rds Internal RIFO / 20 addres	-		~		
CONTEX_WIS_FISEL	F58.	Appli	AppMG	Editericeder	Chennel 12 - 16 We	nds Manal PIPO / 20 addres	sing Dechie		~		
CORTEX_MSE_NS					Chemiel 11 - 4 We	ds Internel FIPO	Deeble		~		
CORTEX_W66_S			-		Channel 10 - 4 Wo	da Internal FIPO	Oweble		~		
OPDMA1	8	8	8		Channel 9 - A Work	ta Imamail PIPO	Deathle		~		
GPID					Charriel 9 - 6 Word		Deable		~		
HECHAN			8	•	Channel 7 - 4 Work		Deathe				-
				0						22222222222	22
AND G	0	0	0	0	Chemiel 8 - 4 Work		Deable		Ŷ		
NVC1_3_Application		8	-		Parasi di Altar	ta Manaci Fill A	Configuration				
NVNC2_N5_Application	8		8				Configuration				22
NVNC_F58L AMICEO	0	0	0	0	Reset Configuration						22
ROC		0	0		Testans					00000000000	
¥ 010_N5			5	<u> </u>							22
					HPDIA1 Returns					0000 00000	
	0	0	0	0						0000 000000	
					Features	First Steps Boot Loader	Secure Application	Non Secure Application	ErMenLader	200000000000	100
					HPOMA1 channel 0	0	0	0		VFBGA163 (Top view)	
Analog				>	HEOMA1 channel 1 HEOMA1 channel 2	8	8	8			
Timera					HEOMA1 channel 3	ŏ	6	ŏ		11	
TITNE'S				· · ·	HFOM A1 channel 4 HFOM A1 channel 5	8	8	8			
Connectivity				,	HEOMA1 channel 6	8	8	8		11	
					HPOMA1 channel 7 HPOMA1 channel 8	8	8	8			
Nutineda				>	HEOMA1 charvel 2	8	8	8			
					HPOMA1 channel 10	0	8	8			
Security				· · ·	HPOMA1 channel 11 HPOMA1 channel 12	8	- 8	8		11	
Computing				>	HFOMA1 channel 13	Ū.	<u>p</u>	D D		a () a	10
					HEOMA1 channel 14 USPALL Channel 15			8		<u> </u>	-

Figure 142. HPDMA1 features with RIF implementation (STM32N6 MCUs)

An example (based on STM32MP2 MPUs) is given for the I2C peripheral.

Pino	ut & Co	nfigur	ation		(Clock (Configura	ation	RIF Project Man	ag
								Software Pac	ks 🗸 Pinout	
				\sim			٢		I2C1 Mode and Configuration	
ategories 🛛 A	λ->Ζ								Mode	
÷	A35 R	A35S (TF-A E	A35S BL(OP-TEE	A35NS (U- <u>Bo</u> ot)	A35NS (Linux)		M33NS (Cube)	Boot time: A35 ROM	Runtime contexts: A35S A35NS A35NS M33NS	
HPDMA1		\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	Assironi	(TF-A BL2) (OP-TEE) (U-Boot) (Linux) (TF-M) (Cube)	
HPDMA2		\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark			
HPDMA3		\checkmark	\checkmark	\checkmark	\checkmark	\checkmark		12C 12C	~	
2C1										
2C2									Configuration	
I2C3					~					
I2C4								Reset Config	uration	
12C5								📀 Parameter S	Settings 🛛 🛇 GIC Settings 🔄 OMA Settings 🔄 🛇 GPIO Settings	
12C6										
12C7										
12C8										
I2S1								12C1	requests should be configured in HPDMA1 or HPDMA2 or HPDMA3	
12S2									Go to HPDMA1 Go to HPDMA2 Go to HPDMA3	
									GO TO HPDMAT GO TO HPDMA2 GO TO HPDMAS	
I3C1										
13C2										
1303				-						_

Figure 143. I2C IP panel



Home 🔪 S	TM32MP2510	CAIx) м	P25.ioc	- Pinout	& Confi	guration	\rangle						GENERA
Pino	ut & Config	gura	tion		(Clock (Configur	ation		RIF		Project N	lanager	
								Software Pac						
Q				~			Ó			HPDM/	1 Mode and Co	Infiguration		
Categories A	->Z										Mode	5		
¢	A35 R A35 (TF		A35S (OP-TEE			M33S (TF-M)	M33NS (Cube)	Boot time:	. A35S	Runtime contexts: A35S	A35NS	A35NS	M33S	M33NS
HPDMA1		v				V		A35 ROI	(TF-A BL2)	(OP-TEE)	(U-Boot)	(Linux)	(TF-M)	(Cube)
HPDMA2		~	V	V	V	~			\checkmark	2	\checkmark		1 1 1	2
HPDMA3		~	~	\checkmark	\checkmark	~	V	Channel 15 -	32 Words Internal FIFC) / 2D addressing St	andard Request	Mode		~
I2C1					V			Channel 14 -	32 Words Internal FIFC) / 2D addressing Di	able			~
								Channel 13 -	32 Words Internal FIFC) / 2D addressing Di	able			~
I2C3					~					, to according by	Configuration			
I2C4											Configuration			
12C5								Reset Config	uration					
12C6								All Channel		CH15 OUser	Constants	Easturas N	1C Settings	
								-		Citta Coser	Constants		io Settings	
12C8								Configure the bel						
								Search (Ctrl+	,					•
								 Circular confi 						
									ar Mode		Disable	9		
								✓ Request Con			0.000			
								Reque	st Handle in IP Structure		SoftWa NONE	are		
									HW request protocol			Burst Level		
/3C4								Channel conf			Sirigie	Darat Certil		
/ ICACHE							V	Priorit			Low			
IPCC1			~					Trans	, action Mode		Norma			
IWDG1	×	~	~					Direct	on		Memor	y To Memory		
IWDG2	~			~	V			✓ Source Data						
IWDG3						~			e Address Increment A	fter Transfer	Disable	ed		
IWDG4								Data			Byte			
LPDMA1	V	\checkmark	~	~	1	~	1		Length		1			
									ted Port for Transfer		Port 0			
								✓ Destination □	ata Setting		Disable			

Figure 144. I2C mode panel

Figure 145. I2C features panel

Pino	ut & Co	onfigura	tion		(Clock C	Configura	ation	ŀ	RIF		Proj	ect Manage	ər	
							~	Software Packs		✓ Pinout					
Q				\sim			٢			HPDN	IA1 Mode and C	Configuration			
Categories 🖌	>Z										Mode				
÷	A35 R	A35S (TF- <u>A</u> B		A35NS (U-Boot)	A35NS (Linux)			Boot time: A35 ROM	A35S	Runtime contexts	A35NS	A35N	S M3	3S	M33NS
HPDMA1				V			V	ASSICOW	(TF-A BL2)	(OP-TEE)	(U-Boot)	(Linux			(Cube)
HPDMA2		~	~	~	~	~	V		✓	<	~	1		2	\checkmark
HPDMA3		~	1	1	1	~	\checkmark	Channel 15 - 32 \	Vords Internal FIFO	/ 2D addressing S	Standard Reques	st Mode			~
✓ I2C1					~			Channel 14 - 32 \	Vords Internal FIFO	/ 2D addressing)isable				~
								Channel 13 - 32 \	Vords Internal FIFO	/ 2D addressing)isable				~
I2C3					~					· · · · · · · · · · · · · · · · · · ·	Configuratio			_	
12C4											Configuratio	m			
12C5								Reset Configurat	ion						
12C6								All Channels		🔉 CH15 📔 📀 Use	er Constants		NVIC Settin		
								• All Channels	SECORITI C		Constants	• realures	VIVIC Settin	ys	
12C8								HPDMA1 features							
								The Divert reatures							
								Features	A35 ROM	A35S	A35S	A35NS	A35NS	M33S	M33NS
										(TF-A_BL2)	(OP-TEE)	(U-Boot)	(Linux)	(TF-M)	(Cube)
13C1								HPDMA1 channel (HPDMA1 channel 1							
								HPDMA1 channel 2							
								HPDMA1 channel 3 HPDMA1 channel 4							
13C4								HPDMA1 channel 4							
ICACHE							~	HPDMA1 channel 6							
IPCC1			~					HPDMA1 channel a HPDMA1 channel 8							
IWDG1	\checkmark	~	~					HPDMA1 channel 9							
IWDG2	\checkmark			×	~			HPDMA1 channel 1							
IWDG3						~		HPDMA1 channel 1 HPDMA1 channel 1							
IWDG4								HPDMA1 channel 1	3						
LPDMA1	~	~	~	~	\checkmark	\checkmark	\checkmark	HPDMA1 channel 1							
								HPDMA1 channel 1							



me 🔰 STM32MP251CAb	× > M	1P25.ioc - RIF >			GENERATE CODE	
inout & Configuration	Cloc	k Configuration	RIF	Project	Manager	Tools
	EXTI	HPDMA1 HPDMA	2 HPDMA3			
RIF Configuration	LXII	Features	Feature_ID	CID	SECURE	PRIVILEGE
		HPDMA1 channel 0	0	0	\checkmark	
		HPDMA1 channel 1	1	0	\checkmark	
	FMC	HPDMA1 channel 2	2	0	\checkmark	
		HPDMA1 channel 3	3	0	\checkmark	
		HPDMA1 channel 4	4	0	\checkmark	
		HPDMA1 channel 5	5	0	\checkmark	
		HPDMA1 channel 6	6	0	\checkmark	
Peripherals (RISUP)	GPIO	HPDMA1 channel 7	7	0	\checkmark	
		HPDMA1 channel 8	8	0	\checkmark	
		HPDMA1 channel 9	9	0	\checkmark	
		HPDMA1 channel 10	10	0	✓	
	DMA	HPDMA1 channel 11	11	0	\checkmark	
	DIVIA	HPDMA1 channel 12	12	0	\checkmark	
		HPDMA1 channel 13	13	0	\checkmark	
		HPDMA1 channel 14	14	0	\checkmark	
Domains (RIMU)		HPDMA1 channel 15	15	1		

Figure 146. DMA RIF-aware IP inheritance

Clock configuration

Clock is a RIF-aware IP. Each clock is a RIF feature that can be protected thanks to the software context assignments of the feature.

The feature protection is then reported in the RIF-aware IP RCC panel.

The RCC feature assignment follows a different scheme, dependent on its type.

Three clocks feature types exist:

- The root clocks:
 - Their assignment is SOC family dependent.
 - On STM32MP25 devices, these features are fixedly assigned to a unique context.
 - The HW resource clocks (RAM or peripherals clocks)
 - Their assignments are inherited from the HW resource assignments it clocks.
 - These clocks may be associated to an additional configuration (the System Mode) allowing to correctly protect the clock when it is shared between several CPU. This is the case for STM32MP25 devices.
- The system resource clocks:
 - These are the remaining clocks.
 - Their configuration should be done manually from the feature panel of RCC IP.

Example of the clock protection of the HW resource BKPSRAM:

- For RCC the user can lock the features.
- Some features have a system mode, it is enabled if the feature has a CID equal to "1&2" as we can see in case of BKPSRAM_CFGR feature above (*Figure 147*).

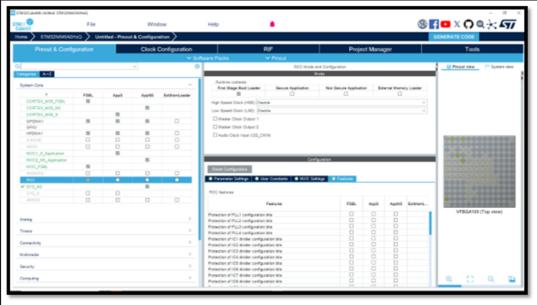


Pinout & Configu	ration	Clock	Configuration			Project Manager		Tools
	EXTI	Features CK_KER_USB2PHY1	Feature_ID 57	CID	SECURE	PRIVILEGE	LOCK	System Mode
RIF Configuration		CK_KER_USB2PHY2	58					
ertir coninguration		CK_ICN_M_GPU	59	1	2	8		
		CK_KER_ETHSWREF	60					
	FMC	CK_MCO1	61					
		CK_MCO2	62					
		CK_CPU1_EXT2F	63	. 1	2			
		CK_SYS_PLL4,5,6,7,8	64	1				
	GPIO	FCALC	65			H		
	GPIO	SYSRST	66	1	2	8	H	
		BOOT_STDB	67	1				
		RDCR	68	1				
		SYSCLK	69	1				
	DMA	CPU1_RES	70	1		H		
		CPU2_RES	71			H	H	
		CPU3_RES	72					
		DEBUG_CFGR	73					
	LICEN	SYSRAM_CFGR	74	-				
	HOEM	VDERAM_CFGR	75		<u> </u>			
		RETRAM_CFGR	76					
		BKPSRAM CFGR	77	182			H	
		SRAM1_CFGR	78	182				
	IPCC	SRAM2_CFGR	79					
		LPSRAM1_CFGR	80					
		LPSRAM1_CFGR	81					
		LPSRAM2_CFGR LPSRAM3_CFGR	82	:		H		
		HPDMA1_CFGR	83					
	FYYR	HPDMA1_CFGR HPDMA2_CFGR	84					
		HPDMA2_CFGR HPDMA3_CFGR	85					
		LPDMA_CFGR	85					
		IPCC1_CFGR	87					
ternal memories (RISAB)	RCC	IPCC2_CFGR	88	-				
		HSEM CFGR	89					
		GPIOA_CFGR	90	1	2	i i i i i i i i i i i i i i i i i i i		
		GPIOB_CFGR	91					
	RTC	GPIOC_CFGR	92					
	1110	GPIOD_CFGR	92					
		GPIOE_CFGR GPIOE_CFGR	93			8		
		GPIOE_CFGR GPIOF_CFGR	94 95					
		GPIOF_CFGR GPIOG_CFGR	95					
	TAMP	GPIOG_CFGR GPIOH_CFGR	96	-				
		GPIOH_CFGR GPIOI_CFGR	97					
		GPIOL_CFGR	98			L .		
		GPIOJ_CFGR	99	:		8		

Figure 147. RIF RCC panel (STM32MP2 MPUs)

For STM32N6 devices, features security are not automatically configured, but are defined by the user.







External interrupts protection

STM32CubeMX does not display a dedicated EXTI IP in the Pinout & Configuration section. However, EXTI can be secured in two ways:

- 1. From peripherals: the security for the interrupts is automatically taken from the peripheral that creates them. For example:
 - EXTI to wake up from peripheral: when assigning an IP, STM32CubeMX identifies and assigns the same security level and context ID to the pins connected to this IP. The security level and context ID are determined by the software context chosen, without any need to adjust settings in the Pinout View or GPIO configuration. The RIF configuration panel uses the IP software context to set the security level and context ID.
 - EXTI for PWR_WKUP: for power wake-up lines associated with a peripheral, the software context assignment is managed through the PWR configuration panel.
- 2. From system resources: these must be set up manually. To do this, adjust the settings in the EXTI sub-panel found within the RIF-aware IPs panel.

Any other EXTI not mentioned has a preset security configuration, which can viewed in the EXTI sub-panel of the RIF-aware IPs panel.

For STM32N6 MCUs, the security settings for EXTI are not automatically assigned but are instead defined by the user, who can set the privilege level.

TM32	File	Window	Help	🔺 🚳 📑 🕒	×Ω◎≿∽
Home > STM32N645A0Hx0	ב 🔪 Unti	itled - RIF >		GENERA	TECODE
Pinout & Configuration	с	Clock Configuration	RIF	Project Manager	Tools
Peripherals (RISUP)	EXTI1	Features EXTI0 EXTI1	EXTI index	SECURE	
	GPDMA1	EXTI2 EXTI3 EXTI4	2 3 4		
Domains (RIMU)	GPIO	EXT15 EXT16 EXT17 EXT18	5 6 7 8		
	HPDMA1	EXTI9 EXTI10 EXTI11 EXTI12	9 10 11 12		
Interrupts (IAC)	PWR	EXTI13 EXTI14 EXTI15 Reserved	13 14 15 16		
External memories (RISAF)	RCC	RTC secure wake-up RTC non-secure wake-up TAMP wake-up OTG1 VBUS plug/unplug	17 18 19 20		
	RTC	OTG2 VBUS plug /unplug I2C1 wake-up I2C2 wake-up I2C3 wake-up	21 22 23 24		
	TAMP	I2C4 wake-up I3C1 wake-up I3C2 wake-up USART1 wake-up	25 26 27 28 29		
		USART2 wake-up USART3 wake-up	30		

Figure 149. RIF panel for EXTI1 (STM32N6 MCUs)

4.7.10 Memory protection for STM32MP2 series

The memory protection is configured through two RIF controllers:

- RISAF (resource isolation slave unit for address space protection full) acts as a firewall, allowing to define access rights for memory regions of DDR and external mapped flash memories
- RISAB (resource isolation slave unit for address space protection block-based) acts as a firewall, allowing to define access rights for memory regions of the internal SRAM.



In the next we will cover only the RISAF, but the process is the same for RISAB. The first is for managing internal memory and the second is for external memory.

RISAF configuration

RISAF is a mechanism allowing the user to configure memory access. Each memory is divided into zones. Each zone can be configured to be read-only or read/write.

The user can also specify if privileges are required, if the memory zone should be secured or encrypted.

The configuration happens at a compartment level.

Through RISAF registers, a trusted application (or the application to which the configuration has been delegated) assigns memory regions and subregions to one or more security domains (secure, privilege, compartment). RISAF includes the DDR memory.

Through RISAF the user can:

- See the list of the different memories
- Access the memory configuration
- Configure the parameters of the memory regions (Start address, region size, Master CID, Read-Write-Privilege)
- Protect memory regions of DDR and external memories by clicking on the dedicated memory.

RISAF includes four memories, namely RISAF1 (BKPSRAM), RISAF2 (OCTOSPI 1&2), RISAF4 (DDR), and RISAF5 (PCIE).

	RISAF1 (E	SKPSRAM)	RISAF2 (OC	TOSPI1&2)	RISA	\F4 (C	DR)	RIS	AF5 (PCIE;																
Peripherals (RISUP)	RISAF	Region	Start	Region	M	aster Cl	D0	M	aster C	ID1	Ma	ister C	D2	М	aster C	ID3	M	aster Cl	D4	Ma	aster Cl	D5	М	laster C	1D6	
	region ID						Р	R			R		P	R						R	w			w	P	Encrypt
	1	bkpspram1	0x42000000	0x1000																						
	2	TFM-ITS	0x42001000	0x1000																						
Domains (RIMU)	3		0x42002000	0x0																						
	4		0x42002000	0x0																						
Internal memories (RISAB)																										



Each memory table contains several columns, such as region ID, region name, start address, region size in hexadecimal, seven groups for Master CID 0 to 6, secure and encrypt.

For each subregion, the user can change the region name and the region size. Each memory has its default configuration.



Pinout & Configu	ration		Clock Conf	iguration						RIF						Pr	oject	Mana	ger						Тос	ols	
																									Los	d OSTL Co	onfiguration
	RISAF1 (B	KPSRAM)	RISAF2 (OC	TOSPI1&2)	RISA	AF4 (0	DDR)	RIS	AF5 (PCIE)																	
	RISAF	Region	Start	Region	Ma	aster C	DO	Ma	ster C	ID1	Ма	ster Cl	D2	M	aster Cl	D3	Ma	ister CII	14	Ма	ster Cl	D5	Ma	ster C	D6		
	region ID	name	address	size	R	w	P	R	w	P	R	w	P	R	w	P	R	w	P	R	w	P	R	w	P	Secure	Encrypt
	1	bkpspram1	0x42000000																								
	2	TFM-ITS	0x42000100																								
	3	region 3	0x42000200																								
	4	region 4	0x420003e0	0x1c20																							

Figure 151. Configuration of a new subregion

RISAF1: backup static random access memory (BKPSRAM)

BKPSRAM is divided into four regions with id 1 to 4 by default. The memory is divided into two equal subregions. The user cannot add or remove regions.

To remove a region, the user must increase the size of another. To add a region, the user must decrease the size of another region.

Two columns are not editable: RISAF region id and start address. The user can change the name of subregion. If the name is empty or the region size is equal to 0, this subregion is not generated.

The start address and the ID column are not editable.

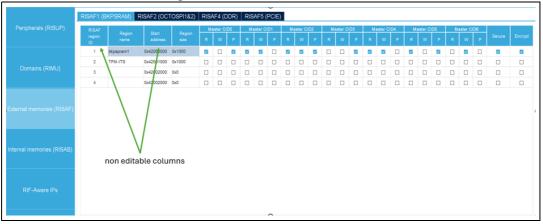
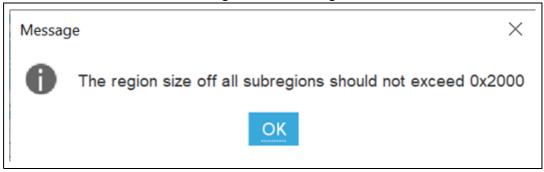


Figure 152. Non editable columns

The region size should not exceed the total region memory, or a warning is displayed.



Figure 153. Warning



The user can assign a subregion to a master CID 0 to 6. CID 7 (not configurable in the UI) is reserved for debugging.

RISAF2: OCTOSPI1&2 memory configuration

The OCTOSPI1&2 Master CID group column is inherited from RISUP peripherals OCTOSPI1 and OCTOSPI2. By default, in OCTOSPI1&2 memory there are two subregions, mm_ospi1 and mm_ospi2.

RISAF				Master CIDO)	Master CID1	l.	Master CID2	2	
region ID	name	address	size		P				P	
1	mm_ospi1	0x60000000	0x0000							
2	mm_ospi2	0x60000000	0x10000000							

Figure 154. OCTOSPI1&2 configuration

OCTOSPI1&2 use the Memory mapped mode: the two controllers are sharing the same 256 MB memory region.

By default, OCTOSPI2 takes the whole region. By clicking on the region size cell of mm_ospi1, a list appears, allowing the user to select the region size. Possible configurations are 0/256, 64/192, 128/128, 192/64, and 256/0 MB (see *Figure 155*). The start address changes automatically.



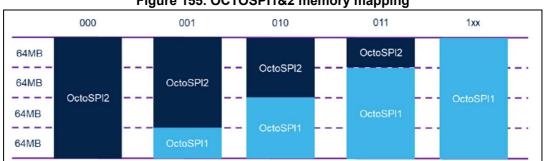




Figure 156. OCTOSPI1&2 region size configuration

me 🔰 STM32MP251CA	× > MP25.ioc											GENERAT	ECODE	
Pinout & Configu	ration	Clock Co	nfiguration		í	RIF		Proj	ect Mana	ager			Tools	
RIF Configuration	RISAF1 (BKP	SRAM) RISAF2 (OCTOSPI1&2)	RISAF4 (DDR)	RISAF	5 (PCIE)								
WRIP Conliguration	RISAF	Region	Start	Region		Master CIDO		Master CID1			Master CID2	2		
		name	address	size	R					R			Secure	Encrypt
	1	mm_ospi1	0x6000000	0x8000000 ~									3	
	2	mm_ospi2	0x68000000	0x0 0x4000000									3	
				0x8000000 0xC000000 0x1000000										
ernal memories (RISAF)														

The master CID column group read/write/privilege are inherited from the RISUP table.

If the OCTOSPI1 peripheral in RISUP is assigned to CID1, Master CID group column 1 is accessible, and the other CIDs are grayed out. If it is privileged in RISUP, it is privileged in Master CID1 privilege column, as shown in Figure 157.

Pinout & Configuration	Clock Confi	guration		F	RIF		P	Pinout & Configuration	Clock C	onfiguration		RIF		F	Proje	ct Ma	nage	r		Tool	s
	Global lock	: OFF							RISAF1 (I	BKPSRAM)	RISAF2 (OC	TOSPI1&2	R	5.AF4 (DDR) RI	5AF5	(PCIE)		
RIF Configuration	Peripherals LTDC_L2	1D 120 1	CID	Secure V	Privilege			RIF Configuration	RISAF		1			er CIDO		laster C					1
	LTDC_ROT	121 1							region			Region			_	_	-	_	V P	Secure	Encry
	LVDS MDF1	84 0 54 0		2					D	name	address	suze	к	W P	R	w	P	R	V P		
	OCTOSPI1	74 1		V		<u> </u>		Peripherals (RISUP)	1	mm_ospi1	0x60000000						1			1	
	OCTOSPI2 OCTOSPIM	75 2		2					2	mm ospi2	0x60000dd8	0xfff228									
	OTFDEC1	125 1		4									-								
	OTFDEC2 PCIE	126 1		2																	
Domains (RIMU)	PKA	93 1		2		ä		Domains (RIMU)													
	RAMCEG	108 1 92 1		 ✓ ✓ 	1																
	RNG	92 1		2	H	H															
	SAI1	49 0		2																	
External memories (RISAF)	SAI2	50 0		1				External memories (RISAF)													
	SAI3 SAI4	51 0 52 0		र र र																	
	SAM SDMMC1	76 0		20	H	님															

Figure 157. OCTOSPI1&2 inheritances from RISUP

If OCTOSPI1 is secure in RISUP, it is secure and graved out in RISAF2. The checkboxes inherit their values from RISUP. Changes to the secure or to the privilege state must be performed in the RISUP table.

If OCTOSPI1 is CID1&2 in RISUP, the two Master CID 1 and CID 2 are activated in RISAF2.



region ID 1 2 alins (RIMU)	P R	
2		
ains (RIMU)		

Figure 158. OCTOSPI1&2 Master CID activation example

OCTOSPI2 is not assigned to any CID in RISUP, so the Master CID 0 is activated by default.

RISAF4: DDR memory configuration

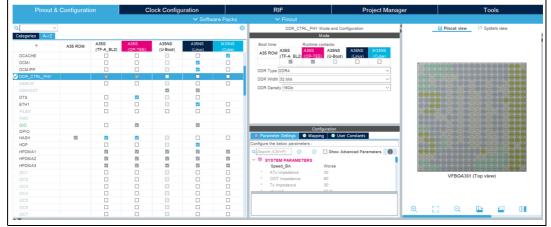
By default, the DDR is configured to handle 4 Gbytes of RAM divided into 15 subregions.

Peripherals (RISUP)	RISAF	-	(anal)	-	M	sster C	DØ	M	aster CI	D1	M	sster C	ID2	M	aster C	D3	M	ster CI	D4	M	aster C	D6	M	aster C	D6	
	region ID	Region name	Start address	Region size	R	w	P	R	w	P	R	w	P	R	w	Р	R	w	Р	R	w	P	R	w	P	Encryp
	1	TFM-code	0x80000000	0x100000																						
- And - Contractory of the Markowski Markowski	2	CM33-Cube-fw	0x80100000	0x800000																						
Domains (RIMU)	3	TFM-data	0x80900000	0x100000																						
	4	CM33-Cube-data	0x80a00000	0x800000																						
	Б	ipc-shmem	0x81200000	0x100000																						
	6	spare1	0x81300000	0xCC0000																						
External memories (RISAF)	7	BL31-context	0x81fc0000	0x40000																						
	В	OP-TEE	0x82000000	0x2000000																						
	9	linuxkernel1	0x84000000	0x76000000																						
nternal memories (RISAB)	10	gpu-reserved	0xfa000000	0x4000000																						
	11	vdec-reserved	0xfe000000	0x800000																						
	12	venc-reserved	0xfe800000	0x800000																						
	13	LTDC-sec-layer	0xff000000	0x800000																						
RIF-Aware IPs	14	LTDC-sec-rotation	0xff800000	0x800000																						
An Andre II a	15	linuxkernel2	0x100000000	0x80000000																						

Figure 159. DDR memory configuration

To change the memory size, go to the Pinout & Configuration tab, select the DDR_CTRL_PHY, and choose the desired memory size.







When returning to RISAF4 (DDR) panel, a new configuration of the DDR memory appears in table:

- The region sizes can be one of the following values: 521 Mbytes, 1 Gbyte, or 2 Gbytes, depending on the user's choices.
- The number of regions decreases from 15 to 14.
- If the user decreases the size of a region, the decreased value is added to the size of the linuxkernel1.
- There is no empty region in the new implementation.
- For the DDR 4 GBytes, if the user decreases the size of a region, the decreased value is added to the size of the linuxkernel2.

RIF Configuration	RISAF	Region		Region	0.000	M	aster O	D0	M	aster C	D1	Ma	aster C	D2	M	aster Cl	D3	М	aster C	D4	M	sster C	D5	M	aster C	ID6
	region ID	name	address	size			w	P				R		P		w		R	W					R		
	1	TFM-code	0x80000000	0x100000																						
	2	CM33-Cube-fw	0x80100000	0x800000																						
Peripherals (RISUP)	3	TFM-data	0x80900000	0x100000																						
	4	CM33-Cube-data	0x80a00000	0x800000																						
	5	ipc-shmem	0x81200000	0x100000																						
	6	spare1	0x81300000	0xCC0000																						
Domains (RIMU)	7	BL31-context	0x81fc0000	0x400000																						
	8	OP-TEE	0x823c0000	0x2000000																						
	9	linuxkernel1	0x843c0000	0x7a400000																						
	10	gpu-reserved	0xfe7c0000	0x40000																						
xternal memories (RISAF)		vdec-reserved	Oxfe800000	0x0000																						
	12	venc-reserved	0xfe800000	0x0000																						
	13	LTDC-sec-layer	0xfe800000	0x800000																						
	14	LTDC-sec-rotation	0xff000000	0x1000000																						

As BKPSRAM memory, the user can assign a subregion to a master CID 0 to 6, set the region as read / write, privilege, secure and encrypt.

RISAF 5: PCIE memory configuration

The PCIE memory is similar to BKPSRAM, except by default it has one subregion that takes the whole memory region size, and the user can add maximum three other regions, set the name, the read/write access rights, the privilege, secure and encrypt.

RISAF				M	aster Cl	ID0	M	ster CI	D1	Ma	ster CI	D2	Ma	aster Cl	D3	Ma	ister Cl	D4	Ma	aster C	ID5	M	aster C	ID6		
region ID	name	address	Region size	R	w	P	R	w	P	R	w	P	R	w	Р	R	w	P		w	Р	R	w	P	Secure	Enc
1	PCIE-device	0x10000000	0x10000000																							0
2		0x20000000	0x0																							0
3		0x20000000	0x0																							0
4		0x20000000	0x0																							[

UM1718 Rev 47



Default memory protection

When starting a new project using the RISAB / RISAF configuration panels, there is a default memory protection scheme already in place. This default setup includes predefined region names, start addresses, and sizes that correspond to a memory map designed for the ST software architecture user is targeting.

The configuration can be modified, even if some areas are reserved. The user can modify this default memory map to suit the needs of a new application. For instance, when working with the STM32MP25 hardware, the OpenSTLinux software configuration for the 4-GByte DDR memory is always pre-loaded as the default setting. However, there are certain regions, specifically those related to the Cortex-M33NS, where the names of the memory regions are fixed and cannot be altered.

Memory mapping generation (MPUs only)

This section discusses the generation of memory mappings for an MPU designed for use with the OpenSTLinux architecture, and supporting the RIF. Note that this MPU does not support the Memory Management Tool.

STM32CubeMX utilizes the RISAF/RISAB configuration as a basis for generating the memory mapping. This mapping is specifically for the master secured firmware associated with the MPU.

The memory mapping created by STM32CubeMX includes only the base-memory regions. These are the regions predefined in the RISAF/RISAB configuration panels.

If an application requires additional memory sub-regions beyond the base-memory regions, these must be defined manually. The definitions go into the User Sections within the initialization code of the firmware that necessitates these extra sub-regions.

For the STM32MP25 hardware, when it is set to A35TD boot mode, a specific memory mapping is produced for the OP-TEE firmware.

This mapping is saved in a file named <project name>-mx-resmem.dtsi, where <project name> is a placeholder for the actual name of the project.

4.7.11 Memory protection for STM32N6 series

The memory protection is configured through one RIF controller:

 RISAF (resource isolation slave unit for address space protection full) acts as a firewall, allowing to define access rights for memory regions of DDR and external mapped flash memories

RISAF configuration

RISAF is a mechanism allowing the user to configure memory access. Each memory is divided into zones. Each zone can be configured to be read-only or read/write.

The user can also specify if privileges are required, if the memory zone should be secured or encrypted.

The configuration happens at a compartment level.

Through RISAF registers, a trusted application (or the application to which the configuration has been delegated) assigns memory regions and subregions to one or more security domains (secure, privilege, compartment). RISAF includes the DDR memory.



Through RISAF the user can:

- See the list of the different memories
- Access the memory configuration
- Configure the parameters of the memory regions (Start address, region size, Master CID, Read-Write-Privilege)
- Protect memory regions of DDR and external memories by clicking on the dedicated memory.

Configure memory access with RISAF for STM32N6 MCUs

The STM32N6 RISAF panel has a global lock, unlike the STM32MP2.

Pinout & Configuration	Clock	Configuration	on	RIF		Project Ma	nager	Т	ools
	Global lock	: OFF							
Peripherals (RISUP)	RISAF 12(XS	PI2) RISAF	13(XSPI3)	RISAF 14(FMC)	RISAF 15(C	CACHEAXI cont	iguration port)	RISAF 21(AHB RAM1)
	Memory	egions conf	iguration	Memory sub-re	gions config	uration			
	Region ID	Region name	Start Address O.	Region size	Filtering	Secure	Read	Write	Privilege
Domains (RIMU)	1	region 1	0x0000	0x0000			0	0	0
	2	region 2	0x0000	0x0000			0	0	0
	3	region 3	0x0000	0x0000			0	0	0
Interrupts (IAC)	4	region 4	0x0000	0x0000			0	0	0
	5	region 5	0x0000	0x0000			0	0	0
	6	region 6	0x0000	0x0000			0	0	0
	7	region 7	0x0000	0x0000			0	0	0
dernal memories (RISAF) RIF-Aware IPs	non edit	able column							

Figure 163. Global lock in RISAF panel for STM32N6 MCUs

The RISAF is divided into subcontrollers for 17 memory zones, which are assigned to security domains through the RISAF subcontrollers listed below:

- RISAF1 (TCM)
- RISAF2 (CPU AXI RAM0)
- RISAF3 (CPU AXI RAM1)
- RISAF4 (NPU master 0)
- RISAF5 (NPU master 1)
- RISAF6 (CPU master)
- RISAF7 (FLEXRAM)
- RISAF8 (CACHE AXI RAM)
- RISAF9 (VENCRAM)
- RISAF11 (XSPI1)
- RISAF12 (XSPI2)
- RISAF13 (XSPI3)
- RISAF14 (FMC)
- RISAF15 (CACHEAXI configuration port)



- RISAF21 (AHB RAM1)
- RISAF22 (AHB RAM2)
- RISAF23 (Backup RAM)

Each RISAF subcontroller manages a specific number of memory regions. By default, it controls seven regions, but some subcontrollers manage 11 regions, while others handle only two regions.

In the user interface Each RISAF sub controller is represented by 2 tables: Memory regions configuration and Memory sub-regions configuration.

Memory regions configuration table contains the following columns:

- Region ID: this column is non editable.
- Region name
- Start Address Offset and Region size: This value must be within a specific range for each region. If the user sets an incorrect value, a popup appears indicating that the value is out of range.
- Filtering
- Secure
- Read, Write, and Privilege: Values range from 0 to 255.

STM32CubeMX Untitled: STM32N645	80HxQ			U					- 0
TM32 TM Cube MX	File		Window	Hel	р		🕸 🕇 🕒) 🗶 🎧 🕲	*
Home 🔪 STM32N645B0Hx	Q Vntitled	- RIF 🔪					GENE	RATE CODE	
Pinout & Configuration	n Cloci	k Configurat	ion	RIF		Project Ma	anager	T	ools
	Global lock	: OFF							
Peripherals (RISUP)	RISAF 1(TCI	M) RISAF 2	(CPU AXI RAM0)	RISAF 3(C	PU AXI RAM1)	RISAF 4(NF	PU master 0)	RISAF 5(NPU	master 1) 🔽
	Memory	regions conf	iguration	mory sub-reg	gions configu	ration			
	Region ID	Region name	Start Address O	Region size	Filtering	Secure	Read	Write	Privilege
Domains (RIMU)	1	region 1	0x0000	0x0000			0	0	0
	2	region 2	0x0000	0x0000			0	0	0
	3	region 3	0x0000	0x0000			0	0	0
	4	region 4	0x0000	0x0000			0	0	0
Interrupts (IAC)	5	region 5	0x0000	0x0000			0	0	0
	6	region 6	0x0000	0x0000			0	0	0
	7	region 7	0x0000	0x0000			0	0	0
	1 Non editable	7							
RIF-Aware IPs	parameter								

Figure 164. RISAF configuration for STM32N6 series

RISAF also covers the configuration of the memory sub-regions in the Memory sub-regions configuration table. Each regions have two dedicated sub-regions (see *Figure 165*).

Subregions are not accessible by default (grayed). To activate a given subregion, activate the related filtering parameter in the Memory regions configuration table (see *Figure 166*).

Memory sub-regions configuration table contains the following columns:

- RISAF region ID
- SubRegion name



- Start Address Offset and Region size: this value must be within a specific range for each region. If the user sets an incorrect value, a popup appears indicating that the value is out of range.
- SubRegion CID: values range from 1 to 7.
- Filtering
- Delegated CID
- Delegation enabled
- Read
- Write
- Secure
- Privilege
- Lock

	RISAF region	SubRegion	Start Address Offeet	Region	SubRegion CID	Delegated CID	Delegation enabled			
		A	0x0000	0x0000	0	0				
Domains (RIMU)	1	в	0x0000	0x0000	0	0				
	2	A	0x0000	0x0000	۰	0				
	2	в	0x0000	0x0000	0	0				
	3	A	0x0000	0x0000	0	0				
5/2 01 AAAA	3	В	0x0000	0x0000	0	0				
Interrupts (IAC)		A	0x0000	0x0000	0	0				
		в	0x0000	0x0000	0	0				
	5	A	0x0000	0x0000	0	0				
	ω.	в	0x0000	0x0000	0	0				
ernal memories (RISAF)	6	A	0x0000	0x0000	0	0	8			
annar mannara (reaso y		B	0x0000	0x0000	0	0				13
	7	A	0x0000	0x0000	0	0				
		B	0x0000	0x0000	0	0				E3

Figure 165. Sub-regions activation in RISAF (showing activated subregions)



ng 🐨 Set MX	File		Window	Help			🕸 🖬 🗖	XOQ	XG
me > STM32N64580HxQ	Untitled	I - RIF					GENER	ATE CODE	-
Pinout & Configuration	Cloc	k Configurati	on see	RIF		Project Mar	nager	Тс	ools
(Global loc	x : 0##							
Peripherals (RISUP)	RISAF 1(TC	M) RISAF 2(CPU AXI RAMO	RISAF 3(CP	U AXI RAMI) RISAF 4(NP	U master 0)	RISAF 5(NPU	master 1)
	Memory	regions confi	guration Me	mory sub-reg	ions config	uration			
Constant and the second second	Report O	Region name	Start Address O	Repon son	Fillering	Becare	Read	Write	Pristage
Domains (RIMU)	1	region 1	0x0000	0x0000	5	0	0	0	0
	2	region 2	0x0000	0x0000			0	0	0
	3	region 3	0x0000	0x0000	0		0	0	0
Interrupts (IAC)		region 4 region 5	0x0000	0x0000	0	0	0	0	0
		region 6	0x0000	0x0000	0	0	0	0	0
	7	region 7	0x0000	0x0000		0	0	0	0
ernal memories (RISAF)									
RIF-Aware IPs									

Figure 166. Sub-regions activation in RISAF (check the filtering parameter)

4.7.12 RIF code generation

The RIF configuration code generation is handled by the STM32CubeMX, which incorporates it into the initialization code of the project. The format of the generated code depends upon the type of driver used to manage the RIF. The options include HAL (Hardware Abstraction Layer) code for the Cube driver and dts-v1 (Device Tree Source version 1) code for the OpenSTLinux driver.

Note: Only dts-v1 code generation is supported.

In the context of the STM32MPU OpenSTLinux (OSTL), RIF configuration code is generated using the dts-v1 format.

The code generation adheres to the generic principles are outlined in Section 9.

The generated code is placed in a file named <project name>-mx-rif.dtsi, which is part of the master Secured firmware. Additionally, code relevant to the First Stage Bootloader (FSBL) firmware is generated in a file named <project name>-mx-fw-config.dts.

The specific syntax and semantic rules for the generated code are detailed in the RIF binding file. For more information, refer to the STM32MPU Wiki portal.

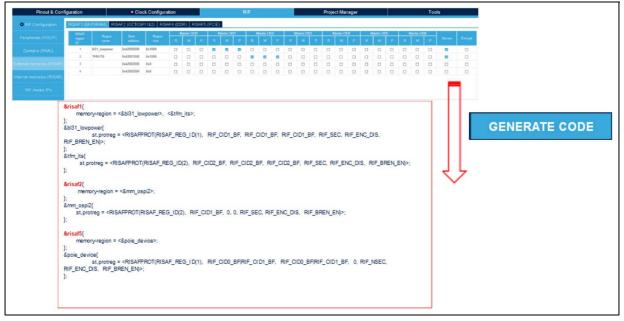
The next section details examples, including user interface screenshots and the corresponding generated code snippets. The procedure is straightforward: configure the RIF panels, click the GENERATE CODE button, and the code is produced in two files, with the RIF configuration landing in a file named project name>-mx-rif.dtsi.



	figuration	Clock Configuration			Project	Manager		
ORF Configuration	Global lock : OFF				220			
	Peripherals	0	co		Pinlege	Lock		
	ADC3 ADF1	69 66	2	21 21	8	8		
	COMBOPHY	67	1	0		ğ		
	CRYP1 CRYP2	96	1	12 12				
	CSI	97 86	2	0 0		H		
	DCMI PSSI DCMIPP	88 87	1	5		8		
	DSI_CMN DSI_R0FIFO	81 123	1	8	8	8		
	DSI_TRIG DTS	122	ò		B	8		
	ETH1 ETH2	60 61	1	10	8	Ē		
	ETHSW_ACM_CFG ETHSW_ACM_MSGBUE	71		0	i i	ġ		GENERATE CODE
RIFPROT(STM32MP25_RIFSC_AL STM32MP25_RIFSC_AL	C3_ID, RIF_UNUSED, RIF_UNUSED, RIF1_ID, EMPTY_SEMWL,	FUNLOCK, RIF_SEC	IF_PRIV. RIF_CID1. RIF_SE , RIF_NPRIV. RIF_CID2. RIF EC, RIF_NPRIV. RIF_CID0. I	SEM_DIS, RIF_CFEN) RIF_SEM_EN, RIF_CFEN)	N.	IJ.	
st.protreg = 2 RIPPROT(RIPPROT(RIPPROT(RIPPROT(RIPPROT(RIPPROT(RIPPROT(RIPPROT(RIPPROT(RIPPROT(RIPPROT(RIPPROT(RIPPROT(RIPPROT(RIPPROT(RIPPROT(RIPPROT(RIPPROT(TIM 321M P25, RIFSC, AL STIM 321M P25, RIFSC, AL STIM 321M P25, RIFSC, CR STM 321M P25, RIFSC, DS STM 321M P25, RIFSC	103 JD. RF_UNASED. RF FFID. EMFTY_SENWL. NMBOPHY JD. RF_UNU. D. EMFTY_SENWL. RF_ ID. RF_UNUSED. RF_ ID. RF_UNUSED. RF_ P.ID. RF_UNUSED. RF P.ID. RF_UNUSED. RF NG_ID. RF_UNUSED. NG_ID. RF_UNUSED. NG_ID. RF_UNUSED. NG_ID. RF_UNUSED. D. RF_UNUSED. RF_UNUSED. ID. RF_UNUSED. RF_UNUSED. ID. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSED. RF_UNUSE	FUNLOCK, RIF, SEC RIF, UNLOCK, RIF, SEC, RIF, UNLOCK, RIF, SEC, RIF, UNLOCK, RIF, SEC, RI UNLOCK, RIF, SEC, RIF, UNLOCK, RIF, SEC, RIF, UNLOCK, RIF, SEC, RIF, UNLOCK, RIF, RIF, UNLOCK, RIF, SEC, RIF, UNLOCK, RIF, SEC, RIF, UNLOCK, RIF, SEC, RIF, UNLOCK, RIF, SEC, RIF, UNCOK, RIF, SEC, RIF, RIF, RIF, RIF,	RIF_NPRIV, RIF_CID2, RIF	SEM_DIS_PRECEEN RF_SEM_EN_RF_CFEN) RF_SEM_EN_RF_CFEN EN_EN_RF_CFEN M_DIS_RF_CFEN M_DIS_RF_CFEN M_DIS_RF_CFEN EN_SEM_EN_RF_CFEN SED_RF_SEM_DIS_RF_CFU N_RF_SEM_DIS_RF_CFEN N_RF_SEM_DIS_RF_CFEN M_DIS_RF_CFEN M_DIS_RF_CFEN M_DIS_RF_CFEN	FD(S) (S) (S)	Ŷ	



Figure 168. Example: RISAF configuration and generated code



Additionally, as described in *Memory mapping generation (MPUs only)*, a partial memory mapping is generated.

4.7.13 Implementation of illegal access controller (IAC) feature on STM32N6 series

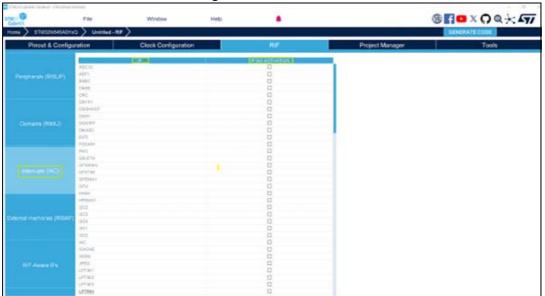
The STM32N6 MCUs support a new feature, the illegal access controller (IAC).

The IAC manages all the interrupts, and the RIF utilizes the IAC feature to centralize the detection of any illegal access related to the RIF, which is managed by a secure application.



The Interrupts (IAC) panel is composed of two columns:

- IP: The name of the IP.
- IP IAC Activation: Enables the user to activate or deactivate the interrupt related to a given IP.





4.8 Pinout & Configuration view for STM32H7 dual-core products

Some STM32H7 products come with an Arm Cortex-M7 core, an Arm Cortex-M4 core, and three power domains.

For such products, the **Pinout & Configuration** view allows the user to:

- For each peripheral and middleware: assign it to one core context or both, whenever possible. in case both contexts are selected, assign an "initializer" core to indicate on which core the peripheral or middleware initialization function shall be called.
- For each peripheral: view the power domain it belongs to.
- For GPIOs: assign it to a core or leave it free for other components that may require it. In this last case the GPIO initialization are performed on the same core as the component reserving it (code is generated accordingly).

For peripherals and middleware, these possibilities are offered in two different panels:

- 1. From the component tree panel, which lists all supported peripherals and middleware (clicking the gear icon enables the "Show contexts" option), see *Figure 170*
- 2. From each component mode panel, opened by clicking the component name.



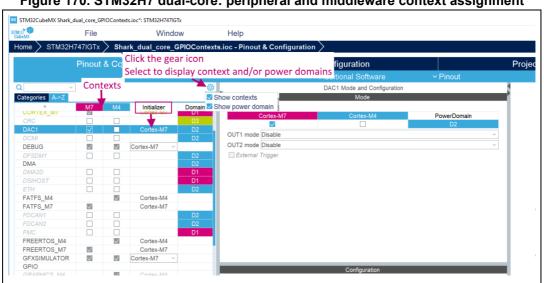


Figure 170. STM32H7 dual-core: peripheral and middleware context assignment

For GPIOs (see *Figure 171*), assignment is done through the **Pinout** view directly or later and automatically through its selection in the platform settings panel of a middleware.





4.9 Enabling security in Pinout & Configuration view (STM32L5 and STM32U5 series only)

The STM32L5 MCU series harnesses the security features of the Arm Cortex-M33 processor and its TrustZone[®] for Armv8-M combined with ST security implementation.



STM32L5 MCUs support

- two levels of privilege
 - unprivileged: software has limited access to system resources
 - privileged: software has full access to system resources, subject to security restrictions
- two security states, Secure and Nonsecure: TrustZone[®] security is activated when the TZEN option bit is set in the FLASH_OPTR register. Security states are orthogonal to mode and privilege, therefore, each security state supports execution in both modes and both levels of privilege.

In STM32CubeMX the choice to activate TrustZone[®] is made at project creation (see *Section 4.2*). When TrustZone[®] is enabled, STM32CubeMX Pinout & Configuration view is adjusted accordingly, with a split between secure (M33S) and nonsecure context (M33NS), and more security-related configuration options (see *Figure 172*).

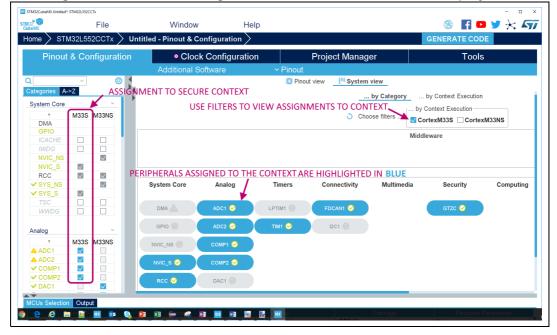


Figure 172. Pinout & Configuration view for TrustZone[®]-enabled projects

4.9.1 Privilege access for peripherals, GPIO EXTIs and DMA requests

Independently of TrustZone $^{\ensuremath{\mathbb{R}}}$, STM32CubeMX enables privilege access:

- for each peripheral: in the GTZC configuration panel (see *Section 4.9.5*), as shown in *Figure 173*
- for each GPIO EXTI: in the GPIO configuration panel, as shown in Figure 174
- for each DMA channel: in the DMA configuration panel (see Section 4.9.4), as shown in *Figure 175*.

Note: When TrustZone[®] is active, either all or none of the RCC registers can be put in privilege mode. In STM32CubeMX, this is done by selecting "Privileged-only attribute" check box from RCC mode panel (see Figure 176). In privilege mode, all RCC registers configuration are reserved for the privilege application through the PWR_CR_PRIVEN bit, which is secured when Trustzone[®] is activated.



UM1718 Rev 47

	privileges for peripherals
GTZC Mod	de and Configuration
	Mode
Runtime contexts:	
Cortex-M33 secure	Cortex-M33 non secure
√	
> GTZC	
Ca	onfiguration
Reset Configuration	
🥺 User Constants	⊘ NVIC Settings
Block-Based Memory Protection Controller	TrustZone Illegal access Controller
	- Memory Protection Controller WaterMark
	ontroller - Privilegeable Peripherals
-	rustZone Security Controller - Securable Peripherals
Configure the below parameters :	
Q Search (CrtI+F) ③ ③	0
~	
Configure Privilege IP	by Individual Privileging from full Not Privileged
Privilegeable Peripheral	Privilege Attribute
ADC1 2	not privileged
COMP1 2	not privileged
CRC	not privileged
CRS	privileged
DAC1	not privileged
DFSDM1	not privileged
FDCAN1	not privileged
FMC	not privileged
HASH	not privileged
I2C1	privileged
I2C2	privileged
I2C3	not privileged
I2C4	not privileged
ICACHE DEC	not privilaged

Figure 173. Setting privileges for peripherals



			C	onfiguratio	n			
Group E	By Periphera	als						~
😔 GPI	0 📀 UA	RT 🛛 📀 NV	1C					
	Signals	1						
Search	(Crtl+F)			`		Show of	only woo	med Pin
	Signal o		Pin Privilege access	GPIO o		GP Ma.	Fa L	Js Mo
PA5	n/a	Free	n/a	n/a	Analog mode	No n/a	n/a	~
PC13	n/a	Free	Privileged-only access	n/a	External Interrupt Mode .		n/a	 Image: A set of the set of the
PC15-0		Free	n/a	n/a	Input mode	No n/a	n/a	✓
PH1-OS.	n/a	Cortex	n/a	n/a	Input mode	No n/a	n/a	✓
PC13 C	onfiguration	:						
	onfiguration ntext Assigr		Free					~
Pin Cor	0	iement	Free Privileged-only	access				~
Pin Cor	ntext Assign vilege acces	iement	Privileged-only		vith Rising edge trigger def	tection		~
Pin Cor Pin Priv GPIO n	ntext Assign vilege acces	nement s	Privileged-only	upt Mode v	0 0 00	tection		~

Figure 174. Setting privileges for GPIO EXTIs



<u> </u>		nty and privilege of Bill		
	DMA Mode	e and Configuration		
		nfiguration		
🛛 DMA1, DMA2 🔤 😔 I	MemToMem			
DMA Request	Channel	Direction		Priority
MEMTOMEM	DMA1 Channel 1	Memory To Memory	Low	
UART4_RX	DMA1 Channel 2	Peripheral To Memory	Low	
UART4_TX	DMA1 Channel 3	Memory To Peripheral	Low	
SPI3_RX	DMA1 Channel 4	Peripheral To Memory	Low	
Add Delete	• 	Pe	ripheral	Memory
Mode Normal	~	Increment Address		✓
		Data Width Byte	\sim	Byte
DMA Request Security	/Privilege			
Enable Channel as Sec	cured 🔽	Enable Channel as Privil	eged 🔽	
Enable Source as Sect	ured 🔽	Enable Destination as S	ecured 🗸	

Figure 175. Configuring security and privilege of DMA requests

Figure 176. RCC privilege mode

	nd Configuration
Mo	ode
Runtime contexts:	
Cortex-M33 secure	Cortex-M33 non secure
\checkmark	\checkmark
Privileged-only attribute	
High Speed Clock (HSE) Disable	~
Low Speed Clock (LSE) Disable	~
Master Clock Output	
LSCO Clock Output	
SAI1 Extern CLock	
SAI2 Extern CLock	
CRS SYNC Disable	~



4.9.2 Secure/nonsecure context assignment for GPIO/peripherals/middleware

STM32CubeMX allows the user

- to assign each peripheral and middleware to one of the contexts
- to assign a GPIO input or output to one of the context or to leave it free for other components that may require it. In this last case the GPIO assignment is in the same context as the component reserving it. By default all IOs are secured.

The assignment is done in different panels:

- For peripherals and middleware only: from the component tree panel when "Show contexts" option is enabled (clicking the gear icon) or from the mode panel.
- For peripherals only: from the GTZC configuration panel (peripherals only).
- For GPIOs only: from the configuration panel or from the Pinout view, through a right-click on the GPIO pin and by selecting "Pin Reservation".
- For DMA requests: from the DMA configuration panel.

Note: RCC resources can be secured through the Clock configuration view (see Section 4.10.2).

Note: For middleware requiring a peripheral the middleware can only be assigned to the context the peripheral is already assigned to.

4.9.3 NVIC and context assignment for peripherals interrupts

When TrustZone[®] is enabled, the interrupt controller is split into NVIC_NS for the nonsecure context and NVIC_S for the secure context. Two SysTick instances are available as well, one for each context: they are visible, respectively, under SYS_NS and SYS_S.

By default, all interrupts are secured.

Peripherals interrupts are automatically assigned to the interrupt controller relevant to the context:

- For peripherals assigned to the nonsecure context, interrupts are enabled on NVIC_NS.
- For peripherals assigned to the secure context, interrupts are enabled on NVIC_S.

4.9.4 DMA (context assignment and privilege access settings)

STM32CubeMX allows the user to set as privileged the DMA channel and in some cases, to secure the DMA channel, source and destination see *Figure 177*.



	DMA Mode	e and Configuration		
	Co	nfiguration		
📀 DMA1, DMA2 🛛 📀	MemToMem			
DMA Request	Channel	Direction		Priority
MEMTOMEM	DMA1 Channel 1	Memory To Memory	Low	
UART4_RX	DMA1 Channel 2	Peripheral To Memory	Low	
UART4_TX	DMA1 Channel 3	Memory To Peripheral	Low	
SPI3_RX	DMA1 Channel 4	Peripheral To Memory	Low	
DMA Request Setting	s			
DMA Request Setting		Per	ripheral	Memory
DMA Request Settings		Per Increment Address	ipheral	Memory
	s		ripheral	Memory Byte
	s	Increment Address		
Mode Normal	sy/Privilege	Increment Address	· ·	Byte

Figure 177. Configuring security and privilege of DMA requests

The DMA channel is set to non-privileged by default. The choice to set it as privileged is always available.

The choice to secure the DMA channel, source, and destination depends on the request characteristics.

There are four cases:

- The request is either a memory to memory transfer request or a DMA generator request: the channel is not secure by default but can be secured. The source and destination can be secured only when the channel is secure.
- The request is for a peripheral assigned to the nonsecure context: channel, source and destination cannot be secured (checkboxes are disabled) and so they are forced to the nonsecure context.
- The request is a peripheral to memory request for a peripheral assigned to the secure context: channel and source are automatically secured (checkboxes enabled, cannot be disabled), while there is a choice to secure or not the destination.
- The request is a memory to peripheral request for a peripheral assigned to the secure context: channel and destination are automatically secured (checkboxes enabled, cannot be disabled), while there is a choice to secure or not the source.



4.9.5 GTZC

To configure TrustZone[®] system security, STM32L5 series come with a Global TrustZone[®] security controller (GTZC). Refer to RM0438 "*STM32L552xx and STM32L562xx advanced Arm*[®]-based 32-bit MCUs" for more details.

In STM32CubeMX, for projects with TrustZone[®] activated, GTZC is enabled by default and cannot be disabled. For projects without Trustzone[®] active, GTZC can be enabled and gives only the possibility to set privileges.

GTZC is made up of three blocks that can be configured through STM32CubeMX using dedicated tabs in GTZC configuration panel:

- TZSC (TrustZone[®] security controller)
 - Defines which peripherals are secured and/or privileged, and controls the nonsecure area size for the watermark memory peripheral controller (MPCWM). The TZSC block informs some peripherals (such as RCC or GPIOs) about the secure status of each securable peripheral, by sharing with RCC and I/O logic.
 - The privileges are set in the TrustZone[®] Security Controller Privilegeable Peripherals tab.
 - The secure states are set in TrustZone[®] Security Controller Securable Peripherals tab (they match the assignment to context (M33S or M33NS) done on the Tree view or in the Mode panel).
 - The MPCWM configuration is done through the TrustZone[®] Security Controller Memory Protection Controller Watermark tab.
- MPCBB (block-based memory protection controller)
 - Controls secure states of all blocks (256-byte pages) of the associated SRAM. It is configured through the Block-based Memory Protection Controller tab.
- TZIC (TrustZone[®] illegal access controller)
 - Gathers all illegal access events in the system and generates a secure interrupt towards NVIC. It is configured through the TrustZone[®] Illegal Access Controller tab.



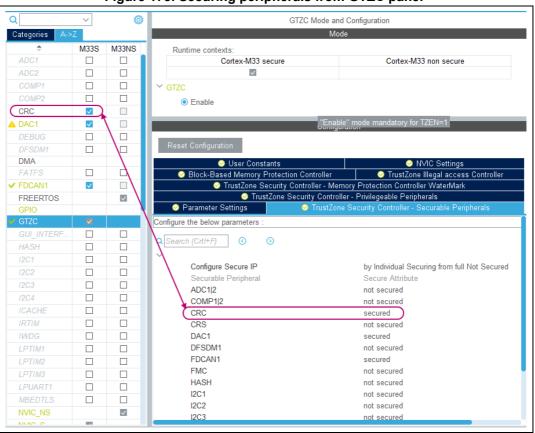


Figure 178. Securing peripherals from GTZC panel

4.9.6 OTFDEC

On-the-fly decryption engine (OTFDEC) allows the user to decrypt on-the-fly AHB traffic based on the read request address information. When security is enabled in the product OTFDEC can be programmed only by a secure host.

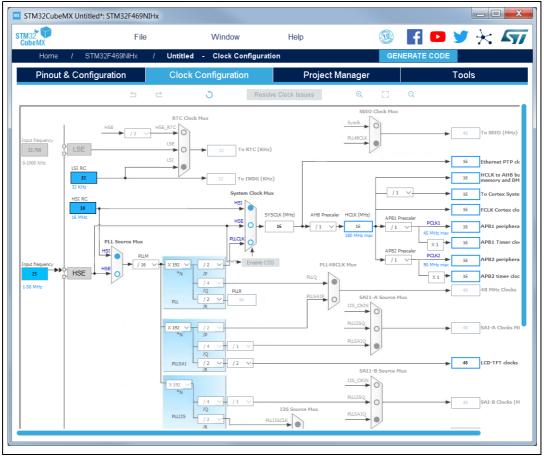
M	lode
Runtime contexts:	
Cortex-M33 secure	Cortex-M33 non secure
\checkmark	

Figure 179. OTFDEC secured when TrustZone [®] is active



4.10 Clock Configuration view

STM32CubeMX **Clock Configuration** window (see *Figure 180*) provides a schematic overview of the clock paths, clock sources, dividers, and multipliers. Drop-down menus and buttons can be used to modify the actual clock tree configuration, to meet the application requirements.





Actual clock speeds are displayed and active. The used clock signals are highlighted in blue.



Out-of-range configured values are highlighted (as shown in *Figure 181*) to flag potential issues. A solver feature is proposed to automatically resolve such configuration issues.

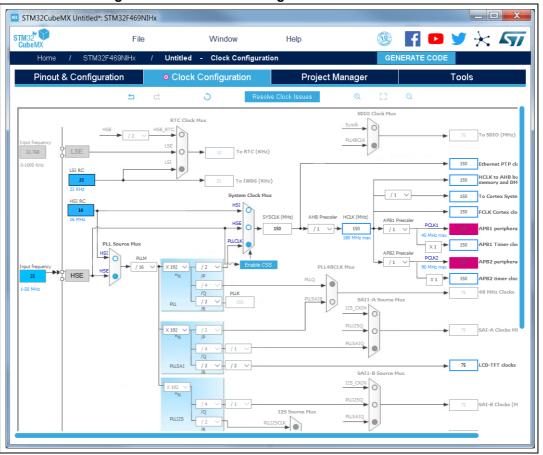


Figure 181. Clock tree configuration view with errors

Reverse path is supported: just enter the required clock speed in the blue filed and STM32CubeMX attempts to reconfigure multipliers and dividers to provide the requested value. The resulting clock value can then be locked by right clicking the field to prevent modifications.

STM32CubeMX generates the corresponding initialization code:

- main.c with relevant HAL_RCC structure initializations and function calls
- stm32xxxx_hal_conf.h for oscillator frequencies and V_{DD} values.

4.10.1 Clock tree configuration functions

External clock sources

When external clock sources are used, the user must previously enable them from the **Pinout** view available under the RCC peripheral.

Peripheral clock configuration options

Other paths, corresponding to clock peripherals, are grayed out. To become active, the peripheral must be properly configured in the **Pinout** view. This view allows the user to:

- Enter a frequency value for the CPU clock (HCLK), buses or peripheral clocks STM32CubeMX tries to propose a clock tree configuration that reaches the desired frequency while adjusting prescalers and dividers and taking into account other peripheral constraints (such as USB clock minimum value). If no solution can be found, STM32CubeMX proposes to switch to a different clock source or can even conclude that no solution matches the desired frequency.
- Lock the frequency fields for which the current value should be preserved Right click a frequency field and select Lock to preserve the value currently assigned when STM32CubeMX searches for a new clock configuration solution. The user can unlock the locked frequency fields when the preservation is no longer necessary.
- Select the clock source that will drive the system clock (SYSCLK)
 - External oscillator clock (HSE) for a user defined frequency.
 - Internal oscillator clock (HSI) for the defined fixed frequency.
 - Main PLL clock
- Select secondary sources (as available for the product)
 - Low-speed internal (LSI) or external (LSE) clock
 - I2S input clock
 - Other sources
- Select prescalers, dividers and multipliers values
- Enable the Clock Security system (CSS) on HSE when it is supported by the MCU This feature is available only when the HSE clock is used as the system clock source directly or indirectly through the PLL. It allows detecting HSE failure and inform the software about it, thus allowing the MCU to perform rescue operations.
- Enable the CSS on LSE when it is supported by the MCU
 This feature is available only when the LSE and LSI are enabled and after the RTC or LCD clock sources have been selected to be either LSE or LSI.
- Reset the Clock tree default settings by using the toolbar Reset button This feature reloads STM32CubeMX default clock tree configuration.
- Undo/Redo user configuration steps by using the toolbar Undo/Redo buttons
- Detect and resolve configuration issues

Erroneous clock tree configurations are detected prior to code generation. Errors are highlighted in fuchsia and the **Clock Configuration** view is marked with a fuchsia cross (see *Figure 181*).

Issues can be resolved manually or automatically by clicking the **Resolve Clock Issue** button that is enabled only if issues have been detected.

The underlying resolution process follows a specific sequence:

- a) Setting HSE frequency to its maximum value (optional).
- b) Setting HCLK frequency then peripheral frequencies to a maximum or minimum value (optional).
- c) Changing multiplexers inputs (optional).



- d) Finally, iterating through multiplier/dividers values to fix the issue. The clock tree is cleared from fuchsia highlights if a solution is found, otherwise an error message is displayed.
- Note: To be available from the clock tree, external clocks, I2S input clock, and master clocks must be enabled in RCC configuration in the **Pinout** view. This information is also available as tooltips.

The tool automatically performs the following operations:

- Adjust bus frequencies, timers, peripherals and master output clocks according to user selection of clock sources, clock frequencies and prescalers/multipliers/dividers values.
- Check the validity of user settings.
- Highlight invalid settings in fuchsia and provide tooltips to guide the user to achieve a valid configuration.

The **Clock Configuration** view is adjusted according to the RCC settings (configured in RCC **Pinout & Configuration** views) and vice versa:

- If in RCC **Pinout** view, the external and output clocks are enabled, they become configurable in the **Clock Configuration** view.
- If in RCC Configuration view, the Timer prescaler is enabled, the choice of Timer clocks multipliers is adjusted.

Conversely, the clock tree configuration may affect some RCC parameters in the configuration view:

- Flash latency: number of wait states automatically derived from V_{DD} voltage, HCLK frequency, and power over-drive state.
- Power regulator voltage scale: automatically derived from HCLK frequency.
- Power over-drive is enabled automatically according to HCLK frequency. When the power drive is enabled, the maximum possible frequency values for AHB and APB domains are increased. They are displayed in the **Clock Configuration** view.

The default optimal system settings that is used at startup are defined in the *system_stm32f4xx.c* file. This file is copied by STM32CubeMX from the STM32CubeF4 MCU package. The switch to user defined clock settings is done afterwards in the main function.



Figure 180 gives an example of Clock tree configuration for an STM32F429x MCU, and *Table 9* describes the widgets that can be used to configure each clock.

Format	Configuration status of the Peripheral Instance
HSI RC	Active clock sources
Audio Clock Input	Unavailable settings are blurred or grayed out (clock sources, dividers,)
AHB Prescaler	Gray drop down lists for prescalers, dividers, multipliers selection.
×1->[Multiplier selection
HSE OSC	User defined frequency values
HCLK (MHz) 	Automatically derived frequency values
16	User-modifiable frequency field
► 10 To KOCK (MHz) lock Vnlock	Right click blue border rectangles to lock/unlock a frequency field. Lock to preserve the frequency value during clock tree configuration updates.

Table 9. Clock configuration view widgets

4.10.2 Securing clock resources (STM32L5 series only)

When the TrustZone[®] security is activated, the RCC is able, through the security configuration register, to prevent nonsecure access to system clock resources.

Accordingly, STM32CubeMX allows the user to configure as secure:

- system clock sources with a fixed frequency: HSI, LSI, and RC48
- system clock sources with a configurable frequency: HSE (+CSS), MSI and LSE (+CSS)
- two multiplexers: CLK48 clock multiplexer, System Clock (+MCO source) multiplexer
- other system configurations: PLLSYS, PLLSAI1, PLLSAI2 phase-locked loops and AHB/APB1/APB2 bus pre-scalers



In the Clock Configuration view, these securable resources are highlighted with a key icon. Security is enabled using the Secure checkbox accessed through a right-click on the resource. Once the resource is secure, it is highlighted with a green square.

Configurable resources can be locked to prevent further configuration changes: this is done by selecting the Lock checkbox accessed through a right-click on the resource.

There is also a shortcut button to lock/unlock in one click all resources that are both securable and configurable.

When a peripheral is configured as secure, its related clock, reset, clock source and clock enable are also secure. In STM32CubeMX the peripheral is configured as secure in the Pinout & Configuration view and its clock source is automatically highlighted as secure using a green square in the Clock configuration view.

View	Description
y	Example of non-configurable system clock resource that is secured.
Input frequency 16 16 HHz 16 HHZ 18 HSE 0.48 MH ✓ Secure Lock	Example of the system clock HSE clock source that is secured and remains open for editing: the frequency value can be changed.
Input frequency 16 16 16 16 16 16 HSE 0-48 MH ✓ Secure ↓ Lock	Example of the system clock HSE clock source that is secured and has been locked for editing: the frequency value cannot be modified.
System Clock Mux MSI HSI HSI SYSCLK (MHz) 16 PLLCLK Enable CSS	Example of the system clock multiplexer that is secured and unlocked: the clock source can be changed.
PLL Source Mux MS HS Secure PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLLM PLM P	Example of the main PLL multiplexer that is secured and locked. The clock source is HSE and cannot be changed. PLLxxM, PLLxxN, PLLxxP, PLLxxQ and PLLxxR are secured and locked for editing as well.

Table 10. Clock Configuration security settings



View	Description
UART4 Clock Mux PCLK1 SYSCLK HSI LSE LOCK	Example of the UART4 clock source multiplexer: the clock source is secured because the UART4 peripheral is configured as secure in the Pinout & Configuration view. It is set to PCLK1 and can be changed as the Lock checkbox is unchecked.
UART4 Clock Mux PCLK1 SYSCLK HSI LSE LSE Lock	Example of the UART4 clock source multiplexer: the clock source is secured because the UART4 peripheral is configured as secure in the Pinout & Configuration view. It is set to PCLK1 and can no longer be changed as Lock is on.
AHB Prescaler HCLK (MHz) AHB Prescaler HCLK (MHz) APB2 Prescaler APB2 Prescaler APB2 Prescaler APB2 Prescaler	Example of securing and locking the access to AHB prescaler. APB1 and APB2 prescalers are locked as well.
LSI RC 32 32 KHz	Example of LSI highlighted as a securable resource using the key icon.
Clock Configuration	Lock/Unlock All button (active only for secure and configurable resources).

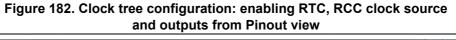
		• • • • •	
Table 10.	. Clock Configuration	n security settings	(continued)

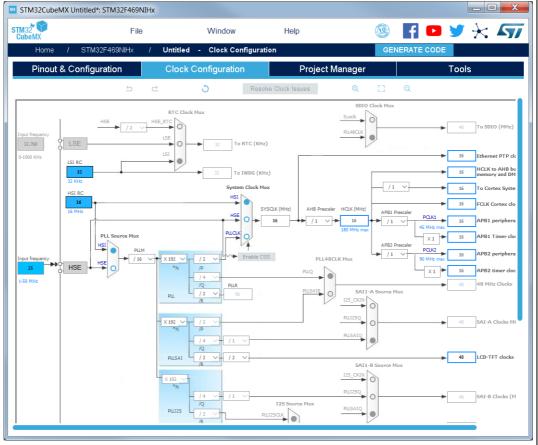


4.10.3 Recommendations

The **Clock Configuration** view is not the only entry for clock configuration, RCC and RTC peripherals can also be configured.

1. From the **Pinout & Configuration** view, go to the RCC mode panel to enable the clocks as needed: external clocks, master output clocks and Audio I2S input clock when available. Then go to the RCC configuration panel, and adjust the default settings if needed. Changes are reflected in the **Clock Configuration** view. The defined settings may change the settings in the RCC configuration as well (see *Figure 182*).







2. Go to the **RCC configuration** in the **Pinout & Configuration** view. The settings defined there for advanced configurations are reflected in the **Clock configuration** view. The defined settings may change the settings in the RCC configuration.

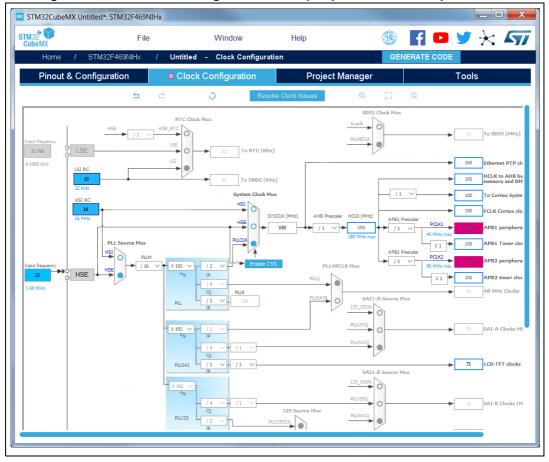


Figure 183. Clock tree configuration: RCC peripheral advanced parameters

4.10.4 STM32F43x/42x power overdrive feature

STM32F42x/43x MCUs implement a power overdrive feature that allows them to work at the maximum AHB/APB bus frequencies (for example, 180 MHz for HCLK) when a sufficient V_{DD} supply voltage is applied (for example, $V_{DD} > 2.1$ V).

Table 11 lists the different parameters linked to the power overdrive feature and their availability in STM32CubeMX user interface.



Parameter	STM32CubeMX panel	Value	
V _{DD} voltage		User-defined within a predefined range. Impacts power over-drive.	
Power regulator voltage scaling		Automatically derived from HCLK frequency and power over-drive (see <i>Table 12</i>).	
Power over-drive	Configuration (RCC)	This value is conditioned by HCLK and V _{DD} values (see <i>Table 12</i>). It can be enabled only if V _{DD} \geq 2.2 V. When V _{DD} \geq 2.2 V it is automatically derived from HCLK, or can be configured by the user if multiple choices are possible (as an example, HCLK = 130 MHz)	
HCLK/AHB clock maximum frequency value	Clock Configuration	Displayed in blue to indicate the maximum possible value. For example: maximum value is 168 MHz for HCLK when power overdrive cannot be activated (when $V_{DD} \le 2.1 \text{ V}$), otherwise it is 180 MHz.	
APB1/APB2 clock maximum frequency value		Displayed in blue to indicate the maximum possible value.	

 Table 11. Voltage scaling versus power overdrive and HCLK frequency

Table 12 gives the relations between power-over drive mode and HCLK frequency.

HCLK frequency range: V _{DD} > 2.1 V required to enable power over-drive (POD)	Corresponding voltage scaling and power over-drive (POD)
≤120 MHz	– Scale 3 – POD is disabled
120 to 144 MHz	 Scale 2 POD can be enabled or disabled
144 to 168 MHz	 Scale 1 when POD is disabled Scale 2 when POD is enabled
168 to 180 MHz	 POD must be enabled Scale 1 (otherwise frequency range not supported)

4.10.5 Clock tree glossary

Acronym	Definition
HSI	High speed Internal oscillator: enabled after reset, lower accuracy than HSE
HSE	High speed external oscillator: requires an external clock circuit
PLL	Phase locked loop: used to multiply above clock sources
LSI	Low speed Internal clock: low power clocks usually used for watchdog timers

Table 13. Glossary

UM1718 Rev 47



Table 15. Olossaly (continueu)	Table	3. Glossary	(continued)
--------------------------------	-------	-------------	-------------

Acronym	Definition
LSE	Low speed external clock: powered by an external clock
SYSCLK	System clock
HCLK	Internal AHB clock frequency
FCLK	Cortex free running clock
AHB	Advanced high performance bus
APB1	Low speed advanced peripheral bus
APB2	High speed advanced peripheral bus

4.11 **Project Manager view**

This view (see *Figure 184*) comes with three tabs:

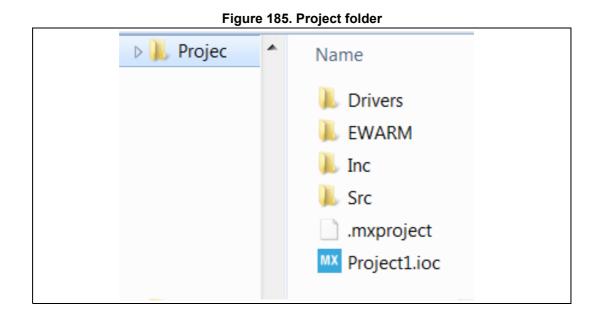
- General project setting: to specify the project name, location, toolchain, and firmware version.
- Code generation: to set code generation options such as the location of peripheral initialization code, library copy/link options, and to select templates for customized code.
- Advanced settings: dedicated to ordering STM32CubeMX initialization function calls.

Pinout & Configuration	Clock Configu	ration	Project Manager	Το	ols
Project	Project Settings Project Name Project Location	C:\STM32CubeMX Proj	erts		Browse
	Application Structure	Basic		∽ □ Do not	generate the main()
Code Generator	Toolchain Folder Location Toolchain / IDE	C:\STM32CubeMX_Proj		V8.32 → Genera	te Under Root
	/Linker Settings	EWARM MDK-ARM STM32CubeIDE Makefile 0x200			
Advanced Settings	Minimum Stack Size	0x400			
	/Thread-safe Settings Cortex-M4NS				
	Enable multi-threaded support				
	Thread-safe Locking Strategy (Mcu and Firmware Package	Default – Mapping suita	ble strategy depending on RTOS se	ection.	~
	Mcu Reference	STM32G431K6Tx STM32Cube FW G4 V	4 F 4		est available version
	Firmware Package Name and Version	STM52Gube FW_G4 V	1.5.1	Use lat	est available VEISION
	Firmware Relative Path	C:/Users/bekrisli/STM3/	2Cube/Repository/STM32Cube_FW_	G4_V1.5.1	Browse

Figure 184. Project Settings window

The code is generated in the project folder tree shown in *Figure 185*.





- *Note:* Some project setting options become read-only once the project is saved. To modify these options, the project must be saved as new, using the **File > Save Project as** menu.
- **Caution:** STM32CubeMX uses reserved folder names. User cannot create new folder named *Middlewares* or *Utilities* inside project folder generated by STM32CubeMX, because, after code regeneration, those folders are deleted or modified.

4.11.1 Project tab

The **Project** tab of the **Project Settings** window allows configuring the following options (see *Figure 184*):

- Project settings:
 - Project name: name used to create the project folder and the .ioc file name at a given project location
 - Project location: directory where the project folder is stored.
 - Application structure: select between Basic and Advanced options.

Basic structure: recommended for projects using one or no middleware. This structure consists in placing the IDE configuration folder at the same level as the sources, organized in sources and includes subfolders (see *Figure 186*)

Advanced structure: recommended when several middleware components are used in the project, makes the integration of middleware applications easier (see *Figure 187*)

- Toolchain folder location: by default, it is located in the project folder at the same level as the .ioc file.
- Toolchain/IDE: selected toolchain
- For the STM32MPUs, OpenSTLinux settings: location of generated device tree and manifest version and contents for current project (see *Figure 188*). These information enable the synchronization of the right SW components versions with STM32CubeMP1 for Cortex[®] M and Linux, tf-a, u-boot for Cortex[®] A. It is important to take them into account especially to ensure one Cube firmware



version is aligned with SW components for Cortex[®] A around OpenAMP / RPM link and resource management API.

Selecting *Makefile* under Toolchain/IDE leads to the generation of a generic gcc-based makefile.

Additional project settings for STM32CubeIDE toolchain:

Select the optional **Generate under root** checkbox to generate the toolchain project files in STM32CubeMX user project root folder or deselect it to generate them under a dedicated toolchain folder.

STM32CubeMX project generation under the root folder allows the user to benefit from the following Eclipse features:

- Optional copy of the project into the Eclipse workspace when importing a project.
- Use of source control systems such as GIT or SVN from the Eclipse workspace.

Choosing to copy the project into workspace prevents any further synchronization between changes done in Eclipse and changes done in STM32CubeMX, as there will be two different copies of the project.

- Linker settings: value of minimum heap and stack sizes to allocate for the application. The default values are 0x200 and 0x400 for heap and stack sizes, respectively. These values may need to be increased when the application uses middleware stacks.
- Firmware package selection when more than one version is available (this is the case when successive versions implement the same API and support the same MCUs). By default, the latest available version is used.
- Firmware location selection option

The default location is the location specified under the **Help > Updater Settings** menu. Deselecting the **Use Default Firmware Location** checkbox allows the user to specify a different path for the firmware that will be used for the project (see *Figure 189*).



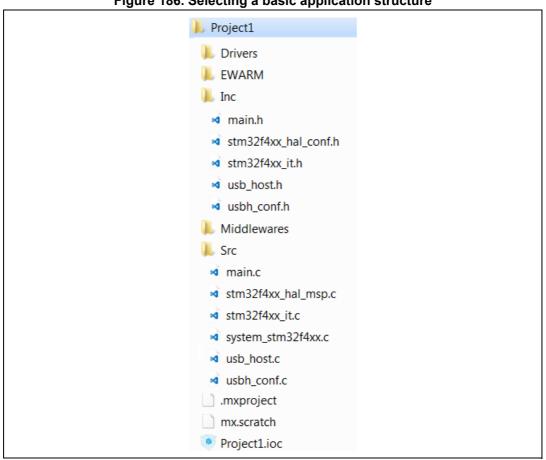


Figure 186. Selecting a basic application structure



📜 Project3
👢 Core
📜 Inc
🗃 main.h
stm32f4xx_hal_conf.h
stm32f4xx_it.h
👢 Src
🐋 main.c
stm32f4xx_hal_msp.c
it.c
system_stm32f4xx.c
📜 Drivers
📜 EWARM
k Middlewares
USB_HOST
👢 Арр
vi usb_host.c
🔹 usb_host.h
👢 Target
v usbh_conf.c
vsbh_conf.h
.mxproject
mx.scratch
🤍 Project3.ioc

Figure 187. Selecting an advanced application structure

Figure 188. OpenSTLinux settings (STM32MPUs only)

DeviceTree Root Local	on	
C:\STM32CubeMX_Pr	jects\DiscoMP1_project\DeviceTree\	
Manifest Version		
openstlinux-4.19	-thud-mp1-19-01-11	
Manifest Content:		
Firmware Name	Community Version	
TF-A	2.0	
Linux	4.19	
	STM32Cube FW_MP1 V1.0.0	
Cube		



Figure 189.	Selecting a	different	firmware	location
-------------	-------------	-----------	----------	----------

Mcu Reference		
STM32F401CDUx		
Firmware Package Name and Version STM322ube FW_F4 V1.24.0	✓ Use latest available version	
Use Default Firmware Location		
	2Cube FW F4 V1.24.0	Browse

The new location must contain at least a *Drivers* directory containing the HAL and CMSIS drivers from the relevant STM32Cube MCU package. An error message pops up if the folders cannot be found (see *Figure 190*).



Choice Firmware Library Directory	X
The selected Firmware should contain at least 'Drivers\STM	32F4xx_HAL_Driver" and 'Drivers\CMSIS' folders.
OK	

To reuse the same *Drivers* folder across all projects that use the same firmware location, select the **Add the library files as reference** from the **Code generator** tab allows (see *Figure 191*).

<u>File Edit View Tools H</u> elp			
Organize Include in library	Share with 🔻	New folder	ii • 🗍 🔞
CUBEMX MyProjectsRepository	Name Drivers Project1 Project2		Date modified 11/17/2016 12:16 11/17/2016 12:19 11/17/2016 2:14 PM
1.	•	III	

Figure 191. Recommended new firmware repository structure





Caution: STM32CubeMX manages firmware updates only for this default location. Choosing another location prevents the user from benefiting from automatic updates. The user must manually copy new driver versions to its project folder.

4.11.2 Code Generator tab

The **Code Generator** tab allows specifying the following code generation options (see *Figure 192*):

- STM32Cube Firmware Library Package option
- Generated files options
- HAL settings options
- Custom code template options

STM32Cube Firmware Library Package option

The following actions are possible:

- Copy all used libraries into the project folder
 STM32CubeMX copies to the user project folder the drivers libraries (HAL, CMSIS) and the middleware libraries relevant to the user configuration (e.g. FatFs, USB).
- Copy only the necessary library files: STM32CubeMX copies to the user project folder only the library files relevant to the user configuration (e.g., SDIO HAL driver from the HAL library).
- Add the required library as referenced in the toolchain project configuration file By default, the required library files are copied to the user project. Select this option for the configuration file to point to files in STM32CubeMX repository instead: the user project folder will not hold a copy of the library files but only a reference to the files in STM32CubeMX repository.

Generated files options

This area allows the user to define the following options:

- Generate peripheral initialization as a pair of .c/.h files or keep all peripheral initializations in the main.c file.
- Backup previously generated files in a backup directory

The .bak extension is added to previously generated .c/.h files.

Keep user code when regenerating the C code.

This option applies only to user sections within STM32CubeMX generated files. It does not apply to the user files that might have been added manually or generated via ftl templates.

• Delete previously generated files when these files are no longer needed by the current configuration. For example, uart.c/.h file are deleted if the UART peripheral, that was enabled in previous code generation, is now disabled in current configuration.

HAL settings options

This area allows selection one HAL settings options among the following:

- Set all free pins as analog to optimize power consumption
- Enable/disable Use the *Full Assert* function: the Define statement in the stm32xx_hal_conf.h configuration file is commented or uncommented, respectively.



Custom code template options

To generate custom code, click the **Settings** button under **Template Settings**, to open the Template Settings window (see *Figure 193*).

The user is then prompted to choose a source directory to select the code templates from, and a destination directory where the corresponding code will be generated.

The default source directory points to the extra_template directory, within the installation folder, to use for storing all user defined templates. The default destination folder is located in the user project folder. STM32CubeMX then uses the selected templates to generate user custom code (see Section 6.3).

Figure 194 shows the result of the template configuration shown on *Figure 193*: a sample.h file is generated according to sample_h.ftl template definition.

	oject Settings oject Code Generator Advanced Settings	
	STM32Cube Firmware Library Package	
	Copy all used libraries into the project folder	
	Copy only the necessary library files	
	Add necessary library files as reference in the toolchain project configuration file	
-0	Generated files	
	Generate peripheral initialization as a pair of '.c/.h' files per IP	
	Backup previously generated files when re-generating	
	Keep User Code when re-generating	
	Delete previously generated files when not re-generated	
rt.	HAL Settings	
	Set all free pins as analog (to optimize the power consumption)	
	Enable Full Assert	
ſ	Template Settings	
1	Select a template to generate customized code Settings	
	Ok Cance	

Figure 192. Project Settings code generator



Template Settings Source Folder Use default locat Location: Select your templates		electronics\STM32Cube\STM	I32CubeMX\db\extra_tem;	Browse
Available Templates			Selected Templates	
\common_h.ftl \default_c.ftl		Add >> > <td>\sample_h.ft</td> <td></td>	\sample_h.ft	
Destination Folder				
📝 Use default locat	ion			
Location:	C: \Users\JohnDoe\STM3	2Cube projects\Project1\Pro	vject1	Browse
			ОК	Cancel

Figure 193. Template Settings window



	arated project template
O → W « Project1 → Proje	ect1 • • • • Search Proj •
<u>File Edit View Tools H</u> elp	
Organize 🔻 Include in library	• » 🗄 • 🔟 🔞
 JohnDoe STM32Cube projects Project1 Project1 	 Name Drivers EWARM Inc Src .mxproject Project1.ioc sample.h
7 items	

Figure 194. Generated project template

4.11.3 Advanced Settings tab

This tab comes with three panels (see *Figure 195*):

- The **Driver selector** panel, to select the driver (HAL or LL) to be used when generating the initialization code of a peripheral instance.
- The **Generated Function Calls** panel, to choose whether the function calls must be generated or not, generated as static or not and in which order.
- The **Register callback** panel, to select the peripherals for which the register callback define must be generated as part of the stm32xxxx_hal_conf.h file.

As an example, when ADC is enabled in the register callback panel, STM32CubeMX generates

#define USE_HAL_ADC_REGISTER_CALLBACKS 1U

Choosing not to generate code for some peripherals or middlewares

By default, STM32CubeMX generates initialization code. This automatic generation can be disabled per peripheral or middleware in the Generate code column.



Ordering initialization function calls

By default, the generated code calls the peripheral/middleware initialization functions in the order in which peripherals and middleware have been enabled in STM32CubeMX. The user can then choose to re-order them by modifying the Rank number, using the up and down arrow buttons.

The reset button allows the user to switch back to alphabetical order.

Disabling calls to initialization functions

If the "**Not to be generated**" checkbox is checked, STM32CubeMX does not generate the call to the corresponding peripheral initialization function. It is up to the user code to do it.

Choosing between HAL and LL based code generation for a given peripheral instance

Starting from STM32CubeMX 4.17 and STM32L4 series, STM32CubeMX offers the possibility for some peripherals to generate initialization code based on Low Layer (LL) drivers instead of HAL drivers: the user can choose between LL and HAL driver in the **Driver Selector** section. The code is generated accordingly (see Section 6.2).

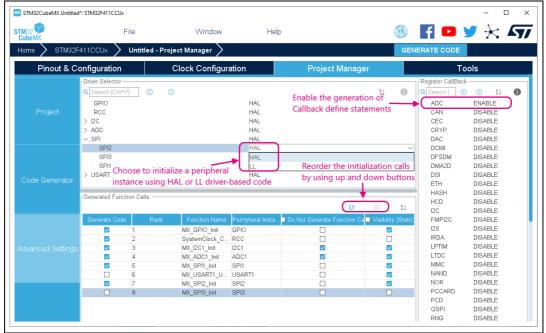


Figure 195. Advanced Settings window



Unselecting the **Visibility (Static)** option, as shown for MX_I2C1_init function in *Figure 195*, allows the generation of the function definition without the static keyword, and hence extends its visibility outside the current file (see *Figure 196*).

Figure 196. Generated init functions without C language "static" keyword

/* Pri	vate function prototypes
void S	<pre>ystemClock Config(void);</pre>
static	<pre>void MX GPIO Init(void);</pre>
static	<pre>void MX LPTIM1 Init(void);</pre>
static	<pre>void MX_LPTIM2_Init(void);</pre>
void M	X I2C1 Init (void);
static	void MX I2C2 Init (void);
static	void MX_SPI1_Init(void);
static	<pre>void MX_SPI2_Init(void);</pre>
static	void MX USART1 UART Init (void)
static	<pre>void MX_USART2_Init(void);</pre>

Caution: For the STM32MPUs

By default the SystemClock_Config function is called in STM32Cube Cube firmware *main()* function, as the 'Not generate Function call' box in Project Manager/Advanced Settings panel is not activated by default (see *Figure 195*).

This configuration is valid for running STM32Cube firmware in engineering (Cortex-M4 stand-alone) mode. and is not valid for running STM32Cube firmware in production mode: the 'Not generate Function call' box must be checked under Project Manager/Advanced Settings panel, so that there is no call to *SystemClock_Config()* in the *main()* function.

4.12 Import Project window

The **Import Project** menu eases the porting of a previously-saved configuration to another MCU. By default the following settings are imported:

- **Pinout** tab: MCU pins and corresponding peripheral modes. The import fails if the same peripheral instances are not available in the target MCU.
- Clock configuration tab: clock tree parameters.
- Configuration tab: peripherals and middleware libraries initialization parameters.
- **Project settings**: choice of toolchain and code generation options.

To import a project, proceed as follows:

1. Select the **Import project** icon **I** that appears under the **File** menu after starting a New Project and once an MCU has been selected.

The menu remains active as long as no user configuration settings are defined for the new project, that is just after the MCU selection. It is disabled as soon as a user action is performed on the project configuration.

- 2. Select **File > Import Project** for the dedicated Import project window to open. This window allows to specify the following options:
 - The STM32CubeMX configuration file (.ioc) pathname of the project to import on top of current empty project.
 - Whether to import the configuration defined in the Power Consumption Calculator tab or not.



- Whether to import the project settings defined through the Project > Settings menu: IDE selection, code generation options and advanced settings.
- Whether to import the project settings defined through the Project > Settings menu: IDE selection and code generation options.
- Whether to attempt to import the whole configuration (automatic import) or only a subset (manual import).
- a) Automatic project import (see *Figure 197*)

Figure 197. Automatic project import

Import Project
Imported Project
C:\STM32CubeMX_UM\Import IOC\IOC to import\f4_demo.ioc
/ Import MX Settings-
Import Power Consumption Calculator Settings
Import Project Settings
/Import Pinout/Clock Configuration/Configuration Settings
Automatic Import
O Manual Import
🗷 Import Pinning Status
🗷 Import Peripherals Configuration
∠Peripheral List
From STM To STM32F722ICKx
ETH None ADC1 ☑ import to ADC1 ✓
ADC1 Bit import to ADC2 V
Try Import Show View Pinout ~
/ Import Status
Initializing: STM32F427I(G-I)Hx Import Analysis: C:\STM32CubeMX_UM\Import IOC\IOC to import\f4_demo.ioc project The Mou (STM32F427IGHx) found in the Project being imported is not the same as the Mcu (STM32F722ICKx) currently edited Import error: ETH peripheral doesn't exist in STM32F722ICKx
OK Cancel
L



b) Manual project import

In this case, checkboxes allow the user to select manually the set of peripherals (see *Figure 198*). Select the **Try Import** option to attempt importing.

mportri	oject 📃 🗾
mported Proje	ect
:\STM32Cub	eMX_UM\Import IOC\IOC to import\f4_demo.ioc
mport MX Set	ttings
Import Pov	ver Consumption Calculator Settings
	ject Settings
	Ject Settings
nport Pinout/	/Clock Configuration/Configuration Settings
Automatic	Import
) Manual Im	port
	Pinning Status
Import	Peripherals Configuration
Periphera	I List
From STN	To STM32F722ICKx
ETH	None
ADC1	✓ import to ADC1
ADC2	✓ import to ADC2 ✓
ADC3	✓ import to ADC3 ✓
CAN1	CAN1
NVIC	VVIC
RCC	✓ RCC
SPI1	✓ import to SPI1 ✓
SPI5	✓ import to SPI5
SPI6	✓ import to SPI2 ✓
SYS	✓ SYS
	Try Import Show View Pinout ~
mport Status	å
Initializ	ring: STM32F427I(G-I)Hx
	alysis: C:\STM32CubeMX_UM\Import IOC\IOC to import\f4_demo.ioc project
	(STM32F427IGHx) found in the Project being imported is not the same as the Mcu (S
⊗ Import	error: ETH peripheral doesn't exist in STM32F722ICKx
	OK Cancel

Figure 198. Manual project import

The Peripheral List indicates:

- The peripheral instances configured in the project to be imported
- The peripheral instances, if any exists for the MCU currently selected, to which the configuration has to be imported. If several peripheral instances are candidate for the import, the user needs to choose one.



Conflicts can occur when importing a smaller package with less pins or a lower-end MCU with less peripheral options.

Click the **Try Import** button to check for such conflicts: the Import Status window and the Peripheral list get refreshed to indicate errors (see *Figure 199*), warnings and whether the import has been successful or not:

- Warning icons indicate that the user has selected a peripheral instance more than once, and that one of the import requests will not be performed.
- A cross sign indicates that there is a pinout conflict, and that the configuration cannot be imported as such.

The manual import can be used to refine import choices and resolve the issues raised by the import trial. *Figure 200* gives an example of successful import trial, obtained by deselecting the import request for some peripherals.

The **Show View** function allows switching between the different configuration tabs (pinout, clock tree, peripheral configuration) for checking influence of the "Try Import" action before actual deployment on current project (see *Figure 200*).

	Manual Im	port			
	Import	Pinning Status			
	✓ Import Peripherals Configuration				
	Peripheral List				
	From STN				
	ETH	None			
	ADC1	import to ADC1			
	ADC2 ADC3	✓ Ø import to ADC2 ✓			
	CAN1	V @ import to ADC3 V			
	NVIC	CAN1			
	RCC	NVIC			
	SPI1				
	SPI5	✓ import to SPI1 ✓ ✓ import to SPI5 ✓			
	SPI5	✓ import to SPI2 ✓			
	SYS	SYS			
8	The Mcu (Try Import Show View Pinout alysis: C:\STM32CubeMX_UM\Import IOC\IOC to import\f4_demo.ioc project (STM32F427IGHx) found in the Project being imported is not the same as the Mcu (Serror: ETH peripheral doesn't exist in STM32F722ICKx			
	Import Tr	-			
		ADC2 partly failed			
	-	or: External-Trigger-for-Injected-conversion:Set mode doesn't exist in STM32F722 ADC3 partly failed			
	-	ADC3 partly failed or: External-Trigger-for-Injected-conversion:Set mode doesn't exist in STM32F722			
		rameters can't be imported for RCC			
		t import parameter:Instruction Cache, it doesn't exist in STM32F722ICKx			
		t import parameter:Prefetch Buffer, it doesn't exist in STM32F722ICKx			
		t import parameter:Data Cache, it doesn't exist in STM32F722ICKx			
		project completed			
J					

Figure 199. Import Project menu - Try Import with errors



	e 200. Import Project menu - Successful import after adjustments
1	lock Configuration/Configuration Settings-
) Automatic Ir	nport
Manual Impo	nt
Import Di	nning Status
	5
Import P	eripherals Configuration
Peripheral I	ist
From STM.	. To STM32F722ICKx
ETH	None
ADC1 🔪	✓ import to ADC1 ∨
ADC2 <	□ Ø import to ADC2 V
ADC3	Comport to ADC3
CAN1	CAN1
NVIC 🔪	VIC NVIC
RCC	
SPI1	✓ import to SPI1
SPI5	✓ import to SPI5 ∨
SPI6	✓ import to SPI2 ∨
SYS	✓ SYS
	Try Import Show View Pinout ~
The Mcu (S	lysis: C:\STM32CubeMX_UM\Import IOC\IOC to import\f4_demo.ioc project IM32F427IGF.n found in the Project being imported is not the same as the Mcu (sproject completed
	OK Cancel

. . . ~ £...... _.. _ ...

3. Choose **OK** to import with the current status or **Cancel** to go back to the empty project without importing.

Upon import, the Import icon gets grayed since the MCU is now configured and it is no more possible to import a non-empty configuration.



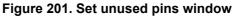
4.13 Set unused/reset used GPIOs windows

These windows are used to configure in the same GPIO mode several pins at the same time.

To open them:

Select **Pinout > Set unused GPIOs** from the STM32CubeMX menu bar.

Note: The user selects the number of GPIOs and lets STM32CubeMX choose the actual pins to be configured or reset, among the available ones.



Set unused GP	IOs	X	
Number of GPIOs	0 22	51	
GPIO Type	Input v		
		Ok Cancel	

• Select **Pinout > Reset used GPIOs** from the STM32CubeMX menu bar.

Depending whether the Keep Current Signals Placement option is checked or not on the toolbar, STM32CubeMX conflict solver is able to move or not the GPIO signals to other unused GPIOs:

- When Keep Current Signals Placement is off (unchecked), STM32CubeMX conflict solver can move the GPIO signals to unused pins in order to fit in another peripheral mode.
- When Keep Current Signals Placement is on (checked), GPIO signals is not moved and the number of possible peripheral modes is limited.

Refer to *Figure 203* and *Figure 204* and check the limitation(s) in available peripheral modes.



Figure 202. Reset used pins window



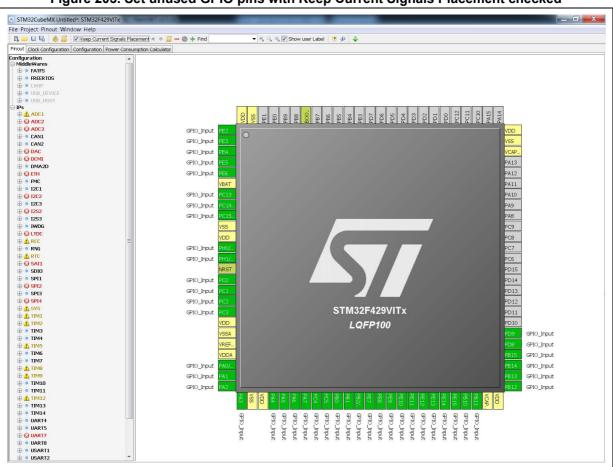


Figure 203. Set unused GPIO pins with Keep Current Signals Placement checked



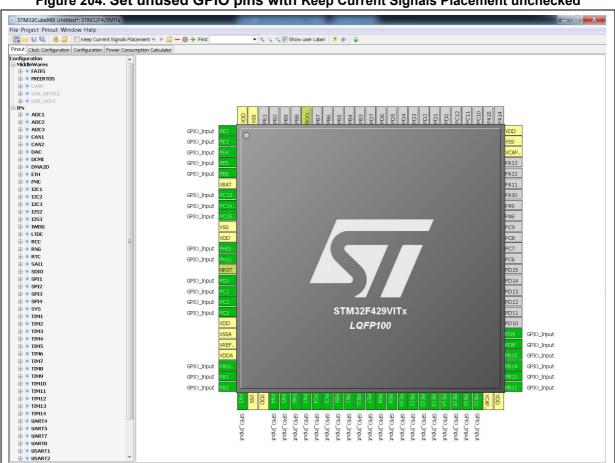


Figure 204. Set unused GPIO pins with Keep Current Signals Placement unchecked

4.14 Update Manager windows

Three windows can be accessed through the **Help** menu available from STM32CubeMX menu bar:

- 1. Select **Help > Check for updates** to open the **Check Update Manager** window and find out about the latest software versions available for download.
- Select Help > Manage embedded software packages to open the Embedded Software Package Manager window and find out about the embedded software packages available for download. It also allows checking for package updates and removing previously installed software packages.
- 3. Select **Help > Updater settings** to open the **Updater settings** window and configure update mechanism settings (proxy settings, manual versus automatic updates, repository folder where embedded software packages are stored).

Refer to Section 3.4 for a detailed description of these windows.

4.15 Software Packs component selection window

This window can be opened by clicking **Middleware and Software Packs** from the **Pinout & Configuration** tab, at any time when working on the project. It allows the user to



select Software Packs components for the current project. It features four panels, as shown in *Figure 205*:

• Filters panel

Can be hidden using the "Show/hide filters" button. It is located on the left side of the window and provides a set of criteria to filter the pack component list.

Packs panel

Main panel, displays the list of software components per pack that can be selected.

• Component dependencies panel

Can be hidden using the "Show/hide dependencies" button. It displays dependencies, if any, for the component selected in the packs panel. It proposes solutions when any is found.

Dependencies that are not solved are highlighted with fuchsia icons.

Once the dependency is solved (by selecting a component among the solution candidates) it is highlighted with green icons.

• Details and warnings panel

Can be hidden using the "Show/hide details" button. It is located on the right hand side. It provide informations for the element selected in the Pack panel.

This element can be a pack, a bundle or a component. It offers the possibility to install a version of the pack available but not yet installed, and allows the user to migrate the current project to a newer version of the pack, raising incompatibilities that cannot be automatically resolved.

ම 🛃 🔤 ව	🗮 💊 🕕 > Show compo	nents for co	ntext: Cortex-M			and warnings	
arch 🗸 🗸	Pack / Bundle / Component	Status	s Version	Selection -	5 Pa	ck STMicroelectronics.X-CUBE-BLE2.3.1.0	
	STMicroelectronics.X-CUBE-BLE2	A	3.1.0 ~	Select the pack ver	sion Bu	ndle BlueNRG-2	
~	✓ Wireless BlueNRG-2	A	3.1.0		Cla		
ck Vendor 🗸	BlueNRG-2 / Controller	A				oup BlueNRG-2	
ck vendor	BlueNRG-2 / HCI_TL	\odot		Basic Y Select co	mnonents	b-group Controller rsion 3.1.0	
ARM	BlueNRG-2 / HCI_TL_INTERFACE	\odot		UserBoard V		Add to favorites	
STMicroelectronics	BlueNRG-2 / Utils				¥	Add to lavorites	
	> Device BLE2_Applications		3.1.0				
ware Component Class	> STMicroelectronics.X-CUBE-DISPLAY	0	1.0.0		vva	rnings (1)	
	> STMicroelectronics.X-CUBE-EEPRMA1		3.0.0		A 1	A This component has unresolved dependencies.	
Artificial Intelligence	> STMicroelectronics.X-CUBE-GNSS1		5.1.0 ~			There are solutions within this pack.	
Audio	> STMicroelectronics.X-CUBE-MEMS1		8.2.0 ~		Des	scription	
Board Extension	> STMicroelectronics.X-CUBE-NFC4		2.0.1 ~				
Board Part	> STMicroelectronics.X-CUBE-SFXS2LP1		2.0.0		BLE	E stack and sample applications for BlueNRG-2 module	
Board Support	> STMicroelectronics.X-CUBE-SUBG2		3.0.1 😐 🗸	Install			
CMSIS	> STMicroelectronics.X-CUBE-TOUCHGFX	0	4.16.0 😐 🗸	Install	Doc	cuments	
DSP Library	> FreeRTOS	G				ense	
,	> HAL Drivers	G⊗			Do	cumentation: STMicroelectronics.X-CUBE-BLE2_GettingS	tan
Data Exchange	> PDM2PCM	\Box					
Device	Component dependencies						
Extension Board	Component BlueNRG-2 / Controller (from bun	dle Wireles	s BlueNRG-2)	Sho	w Resolve		
Graphics	Requires: component bundle BlueNRG-2	class Wire	eless, group Blu	eNRG-2, sub Utils, version 3.1.0	🔺 Missing		
Memory	Solutions in STMicroelectronics.X-CU	BE-BLE2.3.	1.0:				
Motion Libraries	Component BlueNRG-2/Utils			Sho	w Select		
Network							
	Re	solve r	omnoner	t dependencies			
Peripheral			e (Resolv				
RF Library				·			
RTOS	or	he by o	ne (Select)			
Sensors							

Figure 205. Additional software window

See <u>Section 10</u> for more details on how to handle additional software components through STM32CubeMX CMSIS-Pack integration.



4.15.1 Introduction on software components

Arm[®] Keil[™] CMSIS-Pack standard defines the pack (*.pdsc) format for software components to be distributed as Software Packs. A Software pack is a zip file containing a *.pdsc description file.

STM32CubeMX parses the pack .pdsc file to extract the list of software components. This list is presented in the Packs panel.

Arm[®] Keil[™] CMSIS-Pack standard defines a software component as a list of files. The component or each of the corresponding individual files can optionally refer to a condition that must resolve to true, otherwise the component or file is not applicable in the given context. These conditions are listed in the **Component dependencies** panel.

There are no component names. Instead, each component is uniquely identified for a given vendor pack by the combination of class name, group name and a version. Additional categories, such as sub-group and variant can be assigned. These details are listed in the **Details & Warnings** panel.

4.15.2 Filter panel

Click on to open the Filter panel

To filter the software component list, choose pack vendor names and software component classes or enter a text string in the search field.

The resulting software component table is collapsed. Click the left arrow to expand it and display all the components that match the filtering criteria.

lcon	Description
*	Show only favorite packs. A pack is set as favorite in the Details and Warnings panel by clicking 🖈 Add to favorites
\bigcirc	Show only selected components. Components are selected in the Packs panel through checkboxes or variant selection when several implementation choices are available for the same component.
B	Show only installed packs. Enables to show or hide not yet installed packs. Not yet installed packs are distinguished with the icon
МХ	Show only packs compatible with this version of STM32CubeMX. Packs not compatible with this version are distinguished with the icon 😵
	Show only packs compatible with the MCU used for the current project.
3	Reset all filters

Table 14. Additional software window - Filter icons

4.15.3 Packs panel

By default, the Packs panel shows a collapsed view: all known packs are displayed with their name and for one given version (latest version is the default). Icons are used only to highlight the status of a pack version or of a component (see Table Packs panel icons).



UM1718 Rev 47

Details and warnings and Component dependencies panels are used to provide detailed information.

The default view can be expanded by clicking the left arrows, revealing the next level, which can be a Bundle or a top component. The lowest level is the component level.

From this panel, clicking an icon highlighting a limitation or an action opens the relevant secondary panel (Details & Warnings or Component Dependency resolution).

- Some packs can have conditions on Arm[®] cores or STM32 series/MCUs, visible only when Note: the selected MCU meets the criteria. For example, a pack stating the "<accept Dcore="Cortex-M4"/>" condition shows up, but is grayed for MCUs without Cortex[®]-M4 core.
- Note: A pack may promote an API and be shown under the "exposed APIs" entry. Clicking the API name allows to display additional information in the Details & warnings panel. Selecting the component implementing the API selects the API itself. STM32CubeMX generates the project with both the API .h definition file and the API implementation .c file.
- Some components, highlighted in gray in the component panel, are shown as read-only. Note: They are software components (HAL peripheral drivers or middleware offers) coming with STM32Cube MCU embedded software package and are natively available in STM32CubeMX.

Column name	Description				
Pack/Bundle/Component	At pack level, shows the <name of="" pack="" software="" the=""> At bundle level, shows the <name class="" of="" the="">_<bundle any="" if="" name,=""> At component level, shows the <group name="">/<subgroup any="" if="" name,="">. Class names are standardized by the Arm CMSIS standard⁽¹⁾</subgroup></group></bundle></name></name>				
Version	Shows the version that has been selected from a list of one or more available versions of a pack. Bundle and components can either inherit the version of the pack or have their own specific version. The version is shown in the Details and Warning panel.				
Selection	Selects a component through a checkbox when only one implementation is available, or from a list if variants exist.				

Table 15. Additional Software window – Packs panel columns

The Arm[®] Keil[™] CMIS-Pack website, http://www.keil.com, lists the following classes: - Data Exchange: Software components for data exchange - File System: File drive support and file system - Graphics: Graphic libraries for user interfaces 1

- Network: Network stack using Internet protocols

- RTOS: Real-time operating systems

- Safety: Components for testing application software against safety standards

- Security: Encryption for secure communication or storage USB: Universal serial bus stack

- Wireless: Communication stacks such as Bluetooth[®]. WiFi[®], and ZigBee[®].

Table 16. Additional Software window – Packs panel icons

lcon	Description
★	The pack has been added to the user favorite list of packs. Use the Details and Warnings panel to add/remove packs from list of favorites.



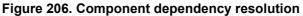
	Table 10. Additional Software window – Lacks panel icons (continued)						
Icon	Description						
8	The pack version is not compatible with this STM32CubeMX version. Solution: select a compatible version.						
↓	The pack version is not yet installed. Solution: go to the Details and Warnings panel to download the pack version to use it for a project.						
0	The component is not available for selection. Solution: download the pack this component belongs to.						
	A component is selected and at least one condition remains to be solved. Select the line of the component with such icon to refresh the Component dependencies panel with the list of dependencies, status and solutions if any found.						
\odot	At least one component is selected and all conditions, if any, are met.						
ſIJ	Other pack versions are available to switch to. Solution: use the Details and Warnings panel to proceed with a change.						
Ð	Highlights the components natively available in STM32CubeMX for the currently selected MCU. They correspond to peripheral drivers and middleware stacks. For such components, the dependencies cannot be automatically resolved: go to the STM32CubeMX pinout view and enable the relevant peripheral instance or middleware in the mode panel. They will appear as selected (green checkbox) in the Component Selector.						

Table 16. Additional Software window – Packs panel icon	s (continued)



The conditions are dependency rules applying to a given software component. When a component is selected, it shows with a green icon if there is no dependency to resolve, with a warning icon otherwise. Click **S** to open the dependency panel (see *Figure 206*).

📅 💊 👔 > Show compor	ients for cont	ext: Cortex-M7	\sim			
Pack / Bundle / Component	Status	Version	Selection			
> ARM.CMSIS-FreeRTOS	Ö	10.3.1	Colocion			
> STMicroelectronics.X-CUBE-AI		5.2.0				
> STMicroelectronics.X-CUBE-ALGOBUILD		1.1.0				
> STMicroelectronics.X-CUBE-BLE1		6.1.0 ~				
STMicroelectronics.X-CUBE-BLE2	4	3.0.0 ~				
✓ Wireless BlueNRG-2	4	3.0.0				
BlueNRG-2 / Controller	A		V			
BlueNRG-2 / HCI_TL	\odot		Basic 🗸			
BlueNRG-2 / HCI_TL_INTERFACE	\odot		UserBoard 🗸			
BlueNRG-2 / Utils						
✓ Device BLE2_Applications		3.0.0				
Application			Not selected $ \smallsetminus $			
> STMicroelectronics.X-CUBE-DISPLAY		1.0.0				
Component dependencies						
Component BlueNRG-2 / Controller (from bun	dle Wireless	BlueNRG-2)			Show	Resolve
✓ Requires: component bundle BlueNRG-2,	class Wirel	ess, group Blue	NRG-2, sub Utils,	version 3.0.0		🔺 Missing
✓ Solutions in STMicroelectronics.X-CUL	BE-BLE2.3.0	.0:				
Component BlueNRG-2/Utils					Show	Select



The panel is refreshed when selecting a component, providing details on the dependencies to solve and the available solutions, if found (see *Table 17*):

- click the Show button to show the component solving the dependency
- click the Select button to select the component solving the dependency
- when available, click Resolve button to automatically resolve the dependencies.



Contextual help	Description		
Component dependencies <i>Board Part</i> EEPROM in pack STMicroelectronics X-CUBE-EEPRMA1.3.0.0) All conditions are solved.	No dependency to solve.		
Component dependencies Construction Show Resolve ✓ Board Extension EEPROM in pack STMicroelectronics X-CUBE-EEPRMA1.3.0.0) Show Resolve ✓ Denies: component bundle EEPROM, class Board Part, group M24 Ø Issue Issue ✓ Conflicts in STMicroelectronics X-CUBE-EEPRMA1.3.0.0; Ø Issue	Dependency to solve but issue encountered (no solution found or conflict).		
Component dependencies Show Recore V Minetex BlavNRG MS in pack STMicroelectronics X-CUBE-BLE1.6.2.0) Show Recore ~ Requires: component bundle BlaveNRG MS, class Wireless, group BlaveNRG MS, sub Utils, version 5.1.0 Missing Missing ~ Solutions in STMicroelectronics X-CUBE-BLE1.6.2.0 Component BlaveNRG MS Utils Show Select	Dependency to solve and at least one solution found.		

Table 17. Component dependencies panel contextual help

4.15.5 Details and Warnings panel

Click on (i) to show the panel (see *Figure 207*).

This panel is refreshed upon selecting a line from the Packs panel.

The following actions are possible from this panel:

- Add/remove the pack from the list of favorite packs
- Install the pack
- Access the pack documentation through links
- Migrate the project to a new pack version

To migrate a project to a new software pack version:

- 1. Open the project
- 2. Migrate to the new pack version
- 3. Generate the code

Known issue: performing step 2 after step 3 (migrating after code generation) leads to errors (wrong file path generation and project compilation failure). To fix such issue, the project must be saved as new, and the code must be generated again. Actions are possible in this panel, namely adding/removing the pack to/from the list of favorite packs, installing a pack, accessing pack documentation through links.



Details and v - Pack detail	-			
Name	X-CUBE-MEMS1			
Vendor	STMicroelectronics			
Version 6.2.0				
🕁 Add to f	avorites			
	npatible with STM32CubeMX from 5.2.0 up to 5.2.1.			
-Description				
Drivore and	sample applications for MEMS components			
	sample applications for MEMS components			
Drivers and Documents				

Figure 207. Details and Warnings panel

4.15.6 Updating the tree view for additional software components

Once the selection of the software components required for the application is complete (see *Figure 208*), click **OK** to refresh STM32CubeMX window: the selected component appears in the tree view under Additional Software (*Figure 209*).

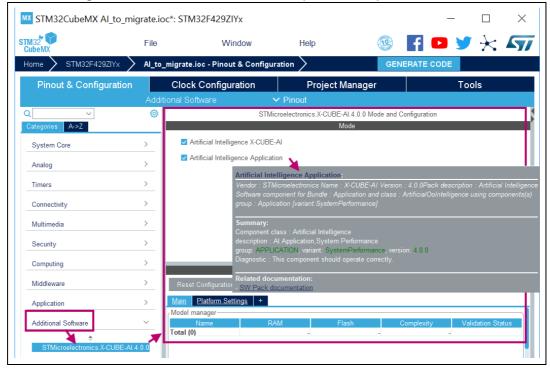
The current selection of additional software components appears in the tree view (see *Figure 209*). The software components must be enabled in the Mode panel and may be configured further if any parameter is proposed in the configuration panel. Hovering the mouse over the component name reveals contextual help with links to documentation.



1		Collapse all
Pack / Bundle / Component	Version	Selection
 STMicroelectronics.X-CUBE-AI 	4.0.0 ~	1
✓ ⊘ Artificial_Intelligence_Application		×
⊘ Application		SystemPerformance ~
✓ ⊘ Artificial_Intelligence_X-CUBE-AI		
⊘ Core		
> STMicroelectronics.X-CUBE-BLE1	4.4.0 ~	
> STMicroelectronics.X-CUBE-GNSS1	3.0.0	
> STMicroelectronics.X-CUBE-MEMS1	6.2.0 🙁 🗸 🗸	
> STMicroelectronics.X-CUBE-NFC4	1.4.0 😆 🕒	

Figure 208. Selection of additional software components

Figure 209. Additional software components - Updated tree view



4.16 LPBAM Scenario & Configuration view

Starting with STM32CubeMX 6.5.0, for projects without TrustZone[®] activated and on the STM32U575/585 product line, users can optionally create LPBAM applications using the LPBAM Scenario & Configuration view (see *Figure 210*).

Starting with STM32CubeMX 6.6.0, users can create LPBAM applications for projects with TrustZone[®] activated on the STM32U575/585 product lines.



Thanks to this view it is possible to:

- add/remove LPBAM applications
- for each LPBAM application, create queues
- for each queue, create functional nodes using the LPBAM firmware API available for peripherals on the Smart Run Domain
- for each LPBAM application, configure the pinout, the clock tree, and HAL-related configurations for the peripherals on the Smart Run Domain.

For details on how to work with this view, refer to Section 18: Creating LPBAM projects.

STM32CubeMX Untitled: STM32U575AG	File Vindo	v Help		×
Home STM32U575AGIX	Untitled - Pinout & Configuration		Configuration GENER	ATE CODE CHECK LPI
	LPBAMS	cenario & Configuration		
LPBAM Management				
✓ 🚾 LPBAM Manager				
	Please	create or activate a LPBAM appl	ication first	

Figure 210. LPBAM window

4.17 CAD Resources view

STM32CubeMX CAD Resources view allows the user to quickly access and download schematic symbols, PCB footprints and 3D CAD models for one or more design toolchains. It requires STM32CubeMX to be connected to the Internet.

To configure and check the Internet connection select **Help > Updater settings** to open STM32CubeMX updater settings window.

CAD Resources can be accessed from the MCU Selector window and from STM32CubeMX project view.



Access from MCU selector

- Open the MCU selector from STM32CubeMX homepage
- Select an MCU commercial part number (Marketing status must not be "Coming soon")
- Select the CAD Resources tab to see the CAD resources (see Figure 211).
- Use the slider to go down the panel and access the different resource views (Symbols, Footprint, and 3D models).

Note: For MCU commercial part numbers in "Coming Soon" Marketing status, there are no CAD resources available (see Figure 212).

To select the resources for download (see *Figure 213*)

- Select the design toolchain
- Select the CAD formats
- Accept terms and conditions
- Click to download
- Specify the download location

	Figure 211. CAD Resources view									
New Project from a MCU/MPU										
MCU/MPU Sel	lector Board Selector Example Selector Cross Selector									
MCU/MPU F	Features Block Diagram Docs & Resources CAD Resources 📑 Datasheet 🗹 Buy 🕞 Start Project									
Comme Part Nu	★ STM32G030F6P6									
Q										
PRODUC	Please choose CAD formats Remove CAD Product CAD Family									
Segmer	Lipidian									
Series										
Line										
Marketi										
Price	CAD LUT									
Packag	Models									
Core	I have read and agree to the Ultra Librarian Terms and Conditions									
Coproce	li Symbol									
MEMORY	•									
Flash =										
32	MCUs/MPUs List: 2 items 📥 Display similar items									
EEPRON	Commercial C Part No Reference Marketing Status X Unit PricX Board X Package X Flash X RAM X									
0	☆ STM32C030F6 STM32C030 Active 0.593 TSSOP-20 32 kBytes 8 kBytes ☆ STM32C030F6 STM32C030 Active 0.593 TSSOP-20 32 kBytes 8 kBytes ★ STM32C030F6 STM32C030 Active 0.593 TSSOP-20 32 kBytes 8 kBytes									
RAM To										

Figure 211. CAD Resources view



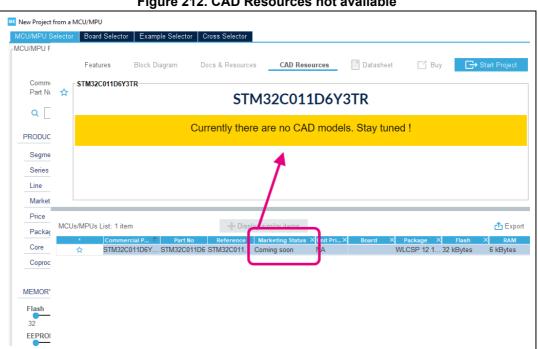
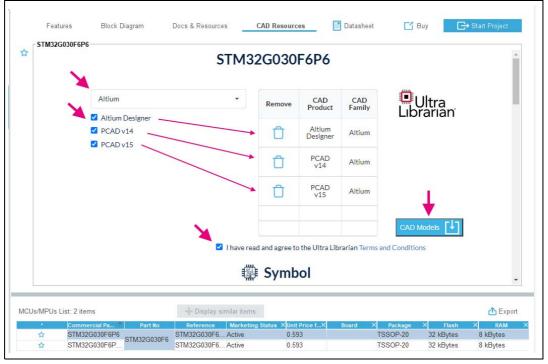


Figure 212. CAD Resources not available





Access from STM32CubeMX project view

- Open an STM32CubeMX project (the MCU must not be in "Coming Soon" Marketing status)
- Select the CAD tab from the Tools panel to access CAD Resources (see Figure 214).



Pinout & C	Configuration	Clock Configuration		Project N	/anager	Tools
		STM32G030F	6P6 _{0 or}	Pins config	Including lab	
PCC	Altium	•	Remove	CAD Product	CAD Family	Ultra Librarian
	 Altium Designer PCAD v14 PCAD v15 		Û	Altium Designer	Altium	Lidrarian
CAD						CAD Models
		I have re	ead and agree	to the Ultra Li	brarian Terms a	and Conditions
		ŧ	🖟 Sym	bol		
				1	1291_CK 20 PB7/PB8 1.0 DK1_BK2 2.0	
		24 927 27 07 071 28 0710 007 28 0700 007 28 0700 007 29 0700 007 29 0700 007 29 0700 007 29 0700 007 20 0000 007 20 0000 0000 0000 20 0000 00000000000			33.57	

Figure 214. CAD Resources in Tools panel

The Symbol view reflects the STM32CubeMX project pinout configuration and, optionally, the labeling (see *Figure 215*). The downloaded CAD files are aligned with the pinout configuration and optionally, with the labels as well.

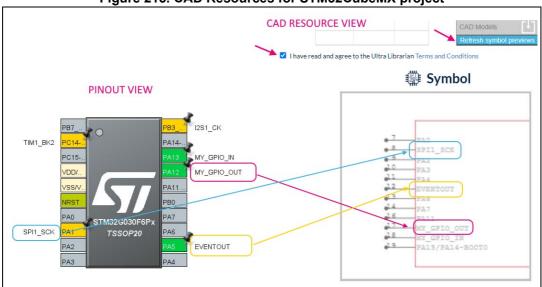
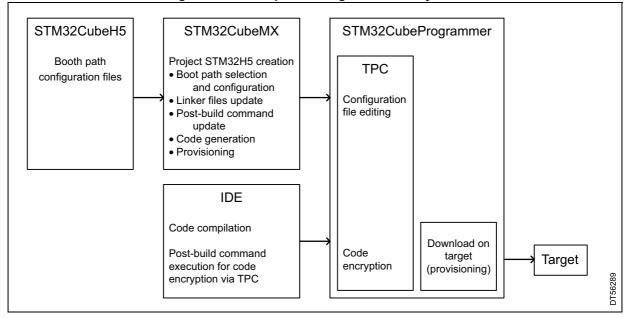


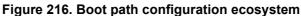
Figure 215. CAD Resources for STM32CubeMX project



4.18 Boot path

STM32CubeMX introduces the possibility to configure the boot path for the STM32H5 series.





Note: STM32H56x and STM32H503 do not support cryptographic hardware accelerator (a feature needed for the ST-iROT and ST-uROT), therefore the full spectrum of boot paths is not available for these MCUs.

For details about boot path and its usage, read the wiki page available on *www.st.com*, and the guide located under the Utilities folder of the STM32Cube firmware package.

This section details, through examples, how to configure a boot path and generate the associated code. It includes compilation, encryption, and provisioning.

4.18.1 Available boot paths

The following tables give an overview of the different boot paths supported by STM32CubeMX, depending upon the device.

MCU	Application	$\begin{array}{l} \textbf{OEM-iRoT} \\ \rightarrow \textbf{Application} \end{array}$	$\begin{array}{l} \textbf{OEM-iRoT} \rightarrow \textbf{uRoT} \\ \rightarrow \textbf{Application} \end{array}$	$\begin{array}{l} \text{ST-iRoT} \\ \rightarrow \text{Application} \end{array}$	$\begin{array}{l} \text{ST-iRoT} \rightarrow \text{uRoT} \\ \rightarrow \text{Application} \end{array}$	
STM32H503x	\checkmark	\checkmark	-	-	-	

Table 18. Boot paths without $TrustZone^{(R)}$ (TZEN = 0)



МСО	S/NS application	OEM-iRoT \rightarrow S/NS application, and OEM-iRoT \rightarrow S/NS application (assembled)	$\begin{array}{l} \text{ST-iRoT} \rightarrow \\ \text{S application} \end{array}$	ST-iRoT \rightarrow uRoT S/NS application, and ST-iRoT \rightarrow S/NS application					
STM32H56x	\checkmark	\checkmark	-	-					
STM32H57x	\checkmark	\checkmark	\checkmark	\checkmark					
STN32H523	\checkmark	ν	-	-					
STM32H533	\checkmark	\checkmark	\checkmark	-					

Table 19. Boot paths with TrustZone[®] (TZEN = 1)⁽¹⁾

1. S: secure, NS: nonsecure.

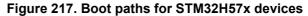
Table 20. Boot paths for STM32H7RS devices ⁽¹⁾

MCU	Application	$\textbf{OEM-iRoT} \rightarrow \textbf{application}$	$\textbf{ST-iRoT} \rightarrow \textbf{application}$	$\textbf{ST-iRoT} \rightarrow \textbf{OEM-uRoT} \rightarrow \textbf{application}$
STM32H7RSx	\checkmark	\checkmark	\checkmark	\checkmark

1. S: secure, NS: nonsecure.

The following figures indicate the boot paths that STM32CubeMX can configure, and the entry points after reset.

The related user option bytes are configured automatically (through Trusted Package Creator installed with STM32CubeMX), and programmed during the provisioning stage.



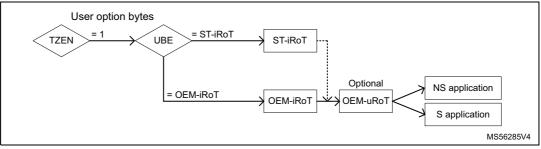
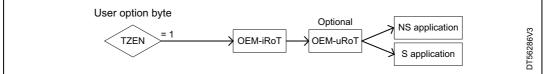


Figure 218. Boot paths for STM32H56x devices





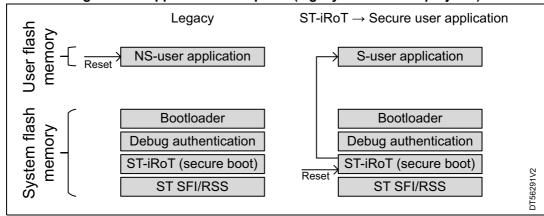
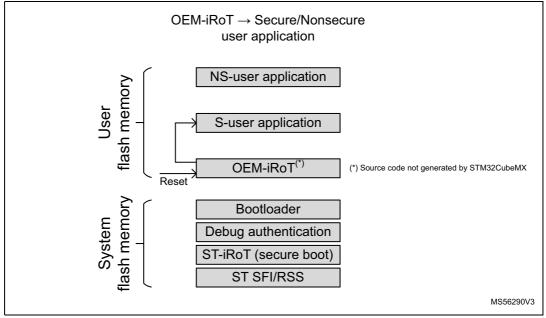


Figure 219. Application boot paths (legacy and ST-iRoT projects)







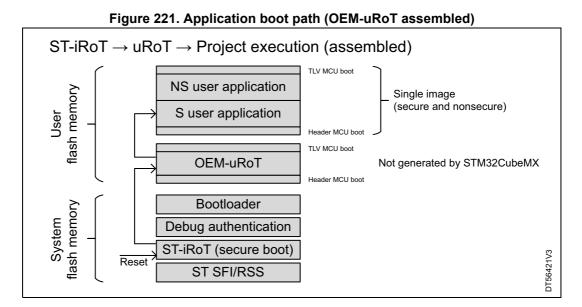
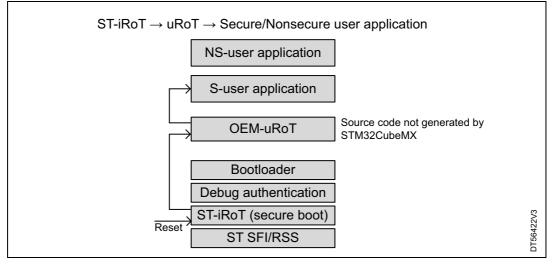


Figure 222. Application boot path: ST-iRoT and uRoT secure/nonsecure project





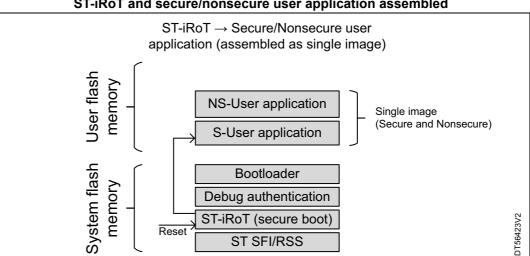


Figure 223. Application boot path: ST-iRoT and secure/nonsecure user application assembled



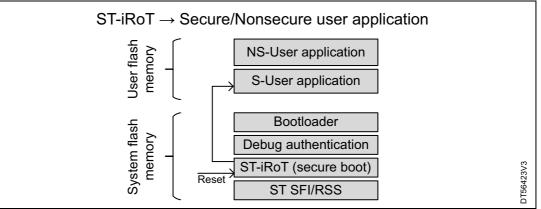
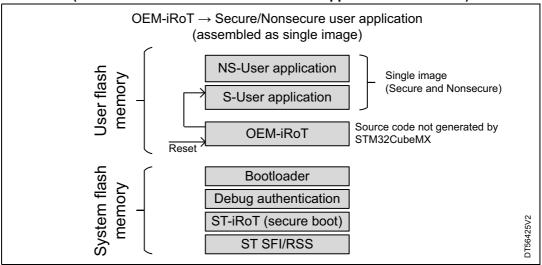


Figure 225. Application boot path: (OEM-iRoT and secure/nonsecure user application assembled)



UM1718 Rev 47



4.18.2 Creating a boot path project: an example

Prerequisites

- Hardware: Discovery board STM32H573I-DK-REVC
- Tools
 - STM32CubeMX-6.8.0 or later
 - Trusted Package Creator (embedded in STM32CubeMX installation folder)
 - CubeFW must be installed through STM32CubeMX
 - IAR Embedded Workbench[®] rev 9.20.4 or later

4.18.3 How to configure an OEM-iRoT boot path

The following instructions describe how to generate an OEM immutable Root of Trust (OEM-iRoT) boot path. The procedure to generate other boot paths is similar, but the data required for the configuration can be different.

Step 1: Selecting the MCU

82CubeMX Untitled	dow Help		
MX File Win	nop		S II - 7 A - 11
xisting Projects		New Project	Manage software installations
Recent Opened Projects		I need to :	Check for STM32CubeMX and embedded software packages updates
test_errorbootpath.ioc	MX	Slart My project from MCU	CHECK FOR UPDATES
Last modified date : 04/10/2022 11:41:15 test_6.8.0-H6.ioc	MX	ACCESS TO MCU SELECTOR	Install or remove embedded software packages
Last modified date : 13/09/2022 10:13:27		Start My project from ST Board	INSTALL / REMOVE
H563_test1.ioc Last modified date : 16/09/2022 16:58:56	MX	ACCESS TO BOARD SELECTOR	
Test2_Mustang_Main.ioc	MX	Star. My project from Example	
Last modified date : 09/09/2022 14:46:19 Test. MustangMainBranch.ioc	MX	ACCESS TO EXAMPLE SELECTOR	
Last modified date : 09/09/2022 10:34:03		/	
Other Projects			ST MCU Finder All STM32 & STM8
			MCUs in one place
Click here to access	the list of s	innorted boards	
or use the MCU se	elector for a	custom product	577
			About STM32 🐓 External Tools

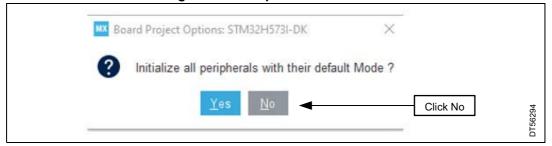
Figure 226. Select the device or board



	ple Selector Cri	oss Selector						>
ard Filters 📩 🔂 🔂 🔿		Features	Large Picture	Docs & Resources	📔 Datasheet	e	Buy	🕞 Start Project
Commercial		STM32H5 Series						
Part Number	<u>∼</u> ☆							
		STM32H573I-D	K Discovery kit with	STM32H573IIK3Q MCU				
۹ 🔜 🔸	-	PREVIEW	Part Number : STM32H57	N DV		Unit Price (US\$): 98.75		
PRODUCT INFO	~	Product is in design stage	Commercial Part Number	STM32H573HDK		Mounted Device : STM32H573I	K30	
Туре	>			The STM32H573I-DK Discovery I	kit is a complete demon	stration and development pla	atform for the STM32H573II	3Q microcontroller.
Supplier	>		· · · · · · · · · · · · · · · · · · ·	featuring an Arm® Cortex®-M33	core with Arm [®] TrustZo	ne [®] .		
MCU / MPU Series	~		1000000000	The full range of hardware featur (source/sink), Ethernet, microSD	res available on the boa	rd (such as an RGB interfac	e LCD with touch panel, US	ART, USB Type-C [®] FS
				and others) help users to interfac	ce with the STM32H5 M	CU peripherals and develop	their applications. Several c	onnectors such as
Check/Uncheck All (1)				ARDUINO [®] Uno V3, Pmod™, and	d STMod+ are also avai	lable on the board to provide	e an easy way to connect ex	tension shields or
Aa fa	61			daughterboards for specific appli The STM32H573I-DK Discovery	ications. kit integrates an STUNE	MRC embedded in circuit	debugger and programmer!	or the STM22
	51			microcontroller with a USB Virtual	I COM port bridge and a	omes with the STM32Cubel	15 Expansion Package, whic	h gathers in one single
STM32F0		ATT COM	the second	package all the generic embedde several examples and application	ed software components	s required to develop an app	lication on STM32H5 microc	ontrollers and provides
STM32F1		MB1677C		several examples and application	is for easy understand	ng.		
STM32F2		and sectors in the sector is a sector of the		J				
STM32F3			00 1	a .				
STM32F4		and the second second	avin 100100	·				
STM32F7			1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1					
STM32G0								
STM32G4								
STM32H5				·				
STM32H7		804 804						
STM32L0								
STM32L1	Boar	ds List: 3 items						📤 Export
STM32L4		•	Verview X Commercial F	wrt No 🗘 Type	×	Marketing Status X	Unit Price (US\$)	X Mounted Device X
STM32L4+								
STM32L5		*	NUCLEO-H503RB	Nucleo-64	Active	15.	0	STM32H503RBT6
STM32MP1								
STM32U5			cost-line					
STM32WB								
STM32WBA		\$	NUCLEO-H563ZI	Nucleo-144	Active	28.	75	5783285632776
STM32WL			_					
		*	STM32H573I-DK	Discovery Kit	Preview	96.	75	
STM32WL		^w	51M32H573HDK	Discovery Kit	b.téném	30.	15	
STAG2H5 ×	_							
STM02H5 × Marketing Status	\sim							
STMB2H5 ×	<u>`</u>					\		
STM02H5 × Marketing Status	>			_				
STM32HE X Marketing Status Price	> > >			7		L .		
STM02H5 X Marketing Status Price MEMORY	>	Sele	ect STM32H5	1		L .		
Marketing Status Price MEMORY FEATURES	> ~	Sele	ect STM32H5] _			-	
Arkating Status Price MEMORY FEATURES Embedded Sensor	> ~ >	Sele	ect STM32H5]	Click to a	open the	٦	
Marketing Status Price MEMORY FEATURES	> ~ > >	Sele	ect STM32H5]	Click to d	open the	7	
Arkating Status Price MEMORY FEATURES Embedded Sensor	> ~ >	Sele	ect STM32H5]	
Marketing Status Price MeMORY FEATURES Embedded Sensor User Button	> ~ > >	Sele	ect STM32H5			open the J selector]	
Marketing Status Price MEMORY FEATURES Embedded Sensor User Button Cannas	> > > > >	Sele	ect STM32H5]	
Entrator & Markeling Status Price MENORY FEATUREES Embedded Sensor User Defator Camera CAN	> > > > >	Sele	ect STM32H5]	

Figure 227. Select the STM32H5 device

Figure 228. Peripheral initialization



If you click yes, there will be an error during the secure code compilation. By default, all peripherals are set as secure, and the memory allocation for the secure code (defined through the OEM-iRoT_boot application) is too small.

Step 2: Project creation with OEM-iRoT boot path

For this example, enable $TrustZone^{(R)}$ (TZEN = 1).



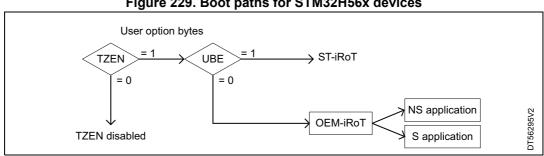


Figure 229. Boot paths for STM32H56x devices

Select the option "with TrustZone activated ?" on the popup window, as shown below.

TrustZo	ne feature available	\times
0	Do you want to create a new proj	ect :
	○ without TrustZone activated ?	
	with TrustZone activated ?	
	OK	

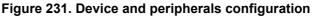
Figure 230. Activate TrustZone



Step 3: Device and peripherals configuration

The device and its peripherals can be configured. In this example, the default configuration is kept.







Step 4: Overall configuration

Configure the application (Figure 232), then save the project (Figure 233).

STM32CubeMX h573_boot_path_tz1.ioc*:	STM32H573AIIxQ			- 0
STM32	File Window	Help	.	🗗 🗖 🗴 🎧 🍭 🛧 🏹
Home 🔪 STM32H573AllxQ 🔪	h573_boot_path_tz1.ioc - Project I	Manager 🔪		GENERATE CODE
Pinout & Configuration	on Clock Configura	ation Pr	oject Manager	Tools
Project	Project Structure S- and NS-application code selected by default	h573_boot_path_tz1 >C:Users\daymarwa\OneDrive - S ✓ Secure Project ☑ Non Secu		
Code Generator	Application Structure Toolchain Folder Location (Toolchain / IDE)	Advanced C:\Users\daymarwa\OneDrive - S EWARM	TMicroelectronics\Desktop\cubem	O not generate the main() projects\h573_boot_path_tz1\
Advanced Settings	Minimum Heap Size Minimum Stack Size /Thread-safe Settings / ConteeM33S	M33S 0x200 0x400	M33NS 0x200 0x400]
Boot Path	Enable multi-threaded support Thread-safe Locking Strategy CortextM33NS-	Default – Mapping suitable strate	gy depending on RTOS selection.	
and Debug Authentication	Enable multi-threaded support Thread-safe Locking Strategy	Default – Mapping suitable strate	gy depending on RTOS selection.	
	Mcu and Firmware Packace Mcu Reference The record files are cogied to a second file and the second files are cogied to a second file and the second files are cogied files and the second files are cogied for a second file files and the second files are cogied for a second file and the second files are cogied for a second file and the second	STM 32H573AIIxQ STM 32Cube FW_H5 V1.5.0RC1	Current frimware package	
	(Firmware Relative Path) => repository	C:/Users/daymarwa/STM32Cube/	Repository/STM32Cube_FW_H5_	V1.5.0RC1 Browse

Figure 232. Configuring the project

Figure 233. Saving the project

32 🐨 beMX	File W	indow H	elp	🕨 💽 🕨	× () @ 🔆 ភ
me 🔪 STM32H573AllxQ 🤇	New Project Ctrl-N Load Project Ctrl-L	roject Manager >		GENERAT	
Pinout & Configurat	io Import Project Ctril ck Col	nfiguration	Project Manager		Tools
	Save Project CtrLS Save Save the current Project Close Project Generate Report CtrLR	EWARM	✓ Min Version	V9.20	Generate Under Root
	Recent Projects	Name	Status	Date modified	Туре
Code Generator	Minimum Stack Size Saved project for Thread-safe Settings		0 0 0	2/6/2025 6:05 PM 2/6/2025 6:05 PM 2/6/2025 9:04 PM	File folder File folder STM32CubeMX
	CortexM33S		itable strategy depending on RTO	S selection.	
Advanced Settings	CortexM33NS	ort	itable strategy depending on RTO		
Boot Path	Mcu and Firmware Package Mcu Reference Firmware Package Name and Ve	STM32H573AIIxQ	/1 5 0RC1		
and Debug Authentication					



Step 5: Boot path selection

The possible first stages are proposed according to selected device and project structure.

STM32CubeM	/X STM32H5_OEMIROT_Demo	.ioc: STM32H573IIKxQ STM	И32H573I-DK				- 🗆 X		
	132 File		Window	Help	L myST	🕸 🚹 🖻 🗙 🗘 🍯	× 57		
Home >	STM32H573IIKxQ - ST	мзан573нок 🔪	STM32H5_OEMIROT_Dem	o.ioc - Project Manager >		GENERATE CODE			
F	Pinout & Configurat		Clock Configurat	tion	Project Manager	Tools			
		STM32CubeProgra	th C:\Program Fil	es\STMicroelectronics\STM32Cu	be\STM32CubeProgrammer\bin\STM32	Browse			
Co		Boot Path Selection	Generate DA Folder						
Adv		Select	No boot path selected (Nothing to	- Wiki	ion (Default Boot Path)				
			\						
			Select boot path						

Figure 234. Boot path selection

Select OEM-iRoT for this example .

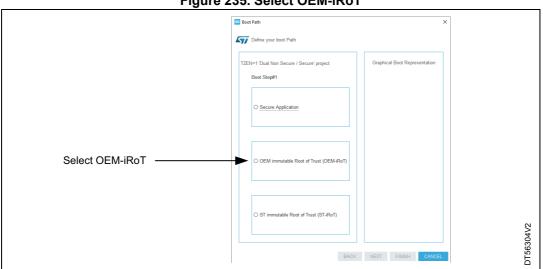


Figure 235. Select OEM-iRoT

	94.0 _0	0. T II ST DOOL P		
Boot Path		×		
Define you	r boot Path			
TZEN=1 'Dual N	Ion Secure / Secure' project	Graphical Boot Representation		
Boot Step	¥1			
⊖ Secure	• Application	Non Secure Secure		
• <u>OEM</u>	mmutable Root of Trust (OEM-iRoT)			
O ST imr	mutable Root of Trust (ST-iRoT)	DEM-IPOT STM32	———— The first stage is shown	
	BACK	NEXT FINISH CANCEL		DT56305V2

Figure 236. First boot path stage

- All possible boot paths for the second stage are proposed according to the selected device and project structure.
- Select "Secure Application", it generates secure and nonsecure codes.

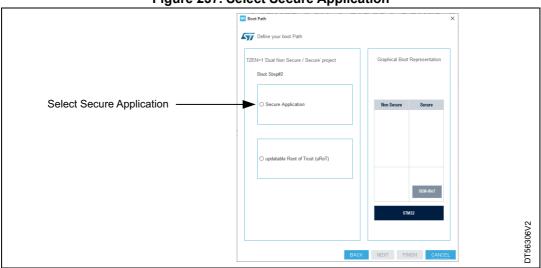


Figure 237. Select Secure Application



• Click on FINISH to generate the binary, RoT_Provisioning folder, and sub-folders.

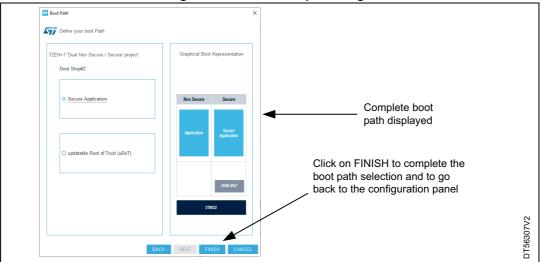
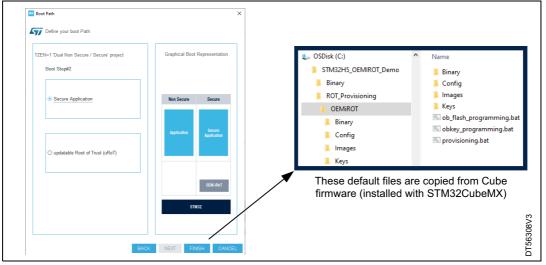


Figure 238. Last boot path stage





Note: If a selected boot path is not supported, a warning message is displayed, and the "FINISH" button is grayed out.

Note: For STM32H56x and STM32H523x devices, it is not possible to configure the OEM-iRoT boot path if the flash size of the current MCU is not aligned with the FLASH_SIZE entry in the map.properties file. A pop-up window (see Figure 240) is displayed.



Figure 240. Flash size not aligned



Figure 241. Boot path and debug authentication panel

2 MX	File	Window	Help	🐥 🗟 🥸	🛾 🔼 X 🖓 🍳 🔆 /
ne 🔪 STM32H573AII	хQ 🔪 Н56ТZ1_Ь	ootpath.ioc - Project Manager	\rangle	GENE	ERATE CODE
Pinout & Config	uration	Clock Configuration	Project M	anager	Tools
	STM 32Cub	eProgrammer Path fault Path C:\Program	n Files\STMicroelectronics\STM32Cube\S	STM32CubeProgrammer\bin\STM	Browse
		nfigure Generate DA Folder	STM32 Trusted Package Creator	nfigure button is clickable if the	Generate DA Folder is checked
Code Generator		fault Reference Location	Boot Path window to select the boo aymarwa\STM32Cube\Repository\STM3		Fill all the fields, ther click on Edit Config Files button to open Trusted Package Creator
Advanced Setting	Config File		aymarwa\OneDrive - STMicroelectronics		Edit Config Files
Boot Path	Reinitia	lize Linker File(s)	ker file content if checked		
		M33S 0xC018400	M 33NS 0 0x801E400	Memory location from which the initial program is loaded for a selected boot mode.	Application

Step 6: Authentication and encryption keys regeneration, option byte file generation

Customization of OEM-iROT configuration file (OEMiROT_Config.obk):

- The default configuration file of CubeFW can be used, but the default keys must be regenerated or replaced
- To customize the configuration file, proceed as follow:
 - a) Launch Trusted Package Creator and select STM32H5 (click edit in Project Manager as indicated in *Figure 239*)
 - b) Open OBkey tab
 - c) The default keys can be regenerated
 - d) The OEMiROT_Config.obk file is generated. The modified parameters are saved in OEMiROT_Config



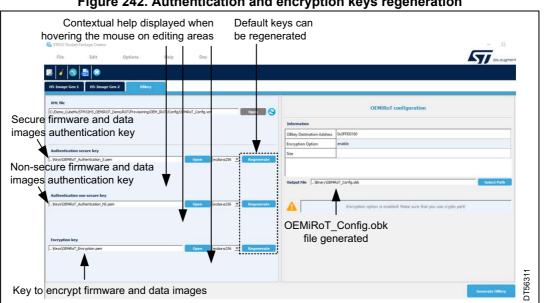


Figure 242. Authentication and encryption keys regeneration

The H5-Image Gen1 and Gen2 tabs indicate the location of the image configuration files and the path of the binary input and output files. Keep the default settings.

😘 STM32 Trusted Pao	kage Creator				– 🗆 🗙	
File	Edit	Options	Help	Doc		
3	a					
H5-Image Gen 1	H5-Image Gen 2	OBkey				
XML file						
C:/STM32CubeMX_Pro	jects/Demo_CubeMx/S	TM32H5_OEMIROT_Demo/	STM32H5_OEMIROT	Demo/ROT_Provisioning/OEMROT/Images/DEMROT_S_Code_Image.xml	Open 🔁	
				Firmware secure image generation		
Dependency with o	ther image					
1.0.0						
Version						
1.0.0						
Firmware binary in	-					
//Secure_nsdib/s	TM32H5_OEMIROT_D	emo_S.hex			Open	
Image output file						
//Binary/STM32H	5_OEMIROT_Demo_S	enc_sign.hex			Select Path	
The purpose of this tool is to generate signed and encrypted images using the imgtool. The tool input is an XML file describing the commands and parameters to be passed to the imgtool. The imgtool is embedded in the STM32CubeProgrammer package.						
					Save Configuration	

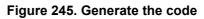
Figure 243. Secure image configuration



🚱 STM32 Trusted F	Package Creator				- 0
File	Edit	Options	Help	Doc	
3 6 8	a				
H5-Image Gen 1	H5-Image Ger	12 OBkey			
no-image Gen I	Ins-image der	Обкеу			
XML file					
C:/STM32CubeMX_F	Projects/Demo_CubeN	4x/STM32H5_OEMIROT_	Demo/STM32H5_OEMIR0	T_Demo/ROT_Provis	isioning/OEMROT/Images/OEMROT_INS_Code_Image.xml Open
					Firmware non secure image generation
Dependency wit	h other image				
Version					
1.0.0					
Firmware binary	input file				
//EWARM/Nor	Secure/STM32H5_OE	MIROT_Demo_NS/Exe/S	TM32H5_OEMIROT_Dem	o_NS.hex	Open
Image output fil					
//Binary/STM3	2H5_OEMIROT_Demo	o_NS_enc_sign.hex			Select Path
Tool Description	•				
The tool input is	an XML file descri	rate signed and encr ibing the commands TM32CubeProgramm	and parameters to b	he imgtool. e passed to the im	ngtool.
					Save Configuration

Figure 244. Nonsecure image configuration

Step 7: Code generation



32 T	File	Window	Help	L myST	🚳 🚹 🔼 X 🗘 🥹 🔀	57
me 🔪 STM32H573IIKxQ - S	sтмз2н573нок > sтмз	2H5_OEMIROT_D	emo.ioc - Project Manager >		GENERATE CODE	
Pinout & Cont	-		lock Configuration	Project Mana	ger Tools	
	STM32CubeProgrammer Pa				T	
	Use Default Path	C:\Program		be\STM32CubeProgrammer\bin\STM32 Browse		
	Debug Authentication				• 1	
		nerate DA Folder			Click here to generate the code	e
	Boot Path Selection	ROT 🔍			and the IDE environmen	
						it.
	Use Default Reference I					
	Reference Location			2Cube_FW_H5_V1.2.0\Projects\STM3; Browse		
	Config Files Location	C:\Users\k	routb\BootpathProjects\STM32H5_C	EMIROT_Demo\ROT_Provisioning\OEI Edit Conig F	iles	
	Linker				Boot Path Image	
	OEM-iRoT Location	C:\Users\k	routb\STM32Cube\Repository\STM3	2Cube_FW_H5_V1.2.0\Projects\STM32 Browse		
	Enable Linker File(s) Up	odate			Non Secure Secure	
	Reinitialize Linker File(s	;)				
	Signature				Application Secure Application	
	Sign Binary(ies)					
	Mapping	M33S	M33NS		0EM-HoT	
	Start address					
	End address				STM12	



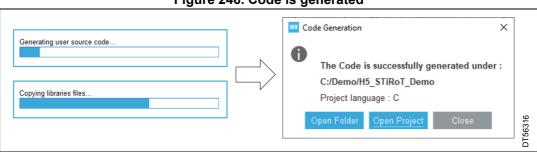


Figure 246. Code is generated

Additional directories, including the IDE environment, are created.

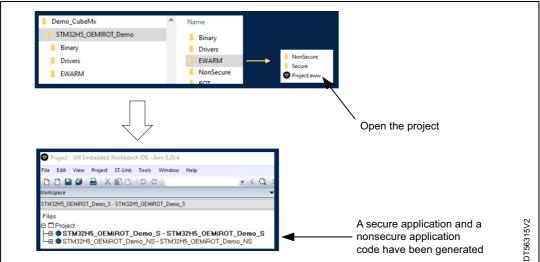


Figure 247. Secure and nonsecure IDE directories

The S and NS applications can be developed using the generated code skeletons.

Step 8: Code compilation

Select Project \rightarrow Option \rightarrow Build Actions. The links to the Trusted Package executable, and to the secure and nonsecure application xml files are filled automatically.



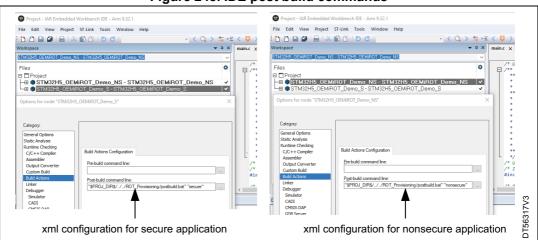
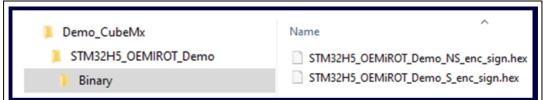


Figure 248. IDE post build commands

The secure code must be generated before the nonsecure one. Compile each code separately (right click on Project \rightarrow Rebuild all). The secure and nonsecure signed and encrypted binaries are generated during the post build phase.

Figure 249. Trusted Package Creator output directory



Step 9: Provisioning of the board

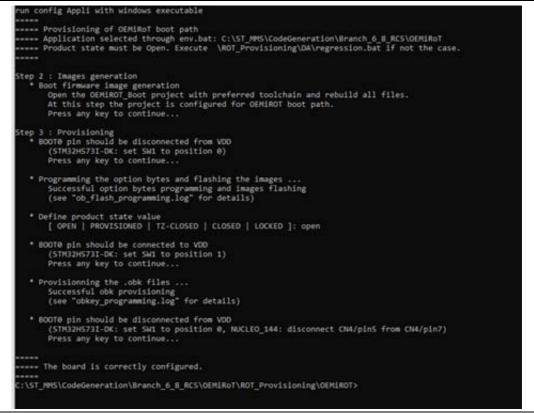
The program cannot be flashed using an IDE. Use provisioning scripts found in the user environment, and double click on the provisioning.bat file (*Figure 250*). During provisioning, log files are generated to inform the user about the activity. Follow the on-screen instructions (*Figure 251*).



File Home Share View	\Branch_6_8_RC4\OEMiRoT_Dual_				
Pin to Quick Copy Paste Copy Copy Copy Copy Copy Copy Copy Copy	Move Copy to to Organize	New item *	Properties History Open Select all	ion	
← → × ↑ 🖡 - Branch_6_8_RC4 > OI	EMiRoT_Dual_LED > ROT_Provisio	oning > OEMiROT	・ じ 、 Search OEM	Mirot	
CEMiRoT		^	Name	Date modified	Туре
OEMiRoT_Dual_LED			Binary	1/3/2023 4:42 PM	File folder
📒 Binary			Config	1/3/2023 4:42 PM	File folder
Drivers			Images	1/16/2023 4:35 PM	File folder
EWARM			📜 Keys	1/3/2023 4:42 PM	File folder
Middlewares			ob_flash_programming.bat	1/13/2023 3:08 PM	Windows Batch F
NonSecure			obkey_programming.bat	1/3/2023 4:42 PM	Windows Batch F
ROT_Provisioning			provisioning.bat	1/13/2023 3:08 PM	Windows Batch F
DA					
Binary					
Certificates					
Config					
Keys					
OEMIROT					
Secure					
Secure_nsclib					

Figure 250. Board provisioning



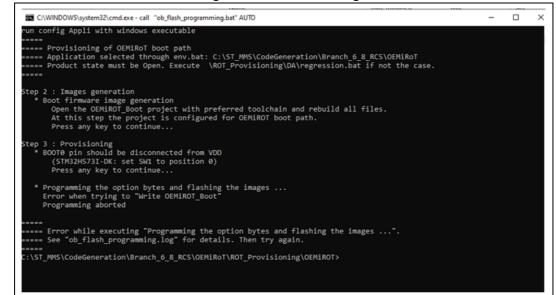


In the user environment, STM32CubeMX has generated an env.bat file, containing the information required for provisioning. Do not change this file.



A pop-up (see *Figure 252*) appears if you forget to compile the project OEMiRoT_Boot in the CubeFW.

Figure	252.	Error	message



4.18.4 How to configure an ST-iRoT boot path

The configuration for an ST immutable Root of Trust (ST-iRoT) boot path. The requirements are the same of the previous example.

Step 1: Generating the code

- Select an STM32H57x MCU
- Create a project with TrustZone[®] activated (TZEN = 1)
- In Project Manager, choose "Secure Project"
- Save the project
- Go to "Boot Path and Debug Authentication" tab, and press the Select button
- Choose ST immutable Root of Trust (ST-iRoT)



i igule 200. Select	,
🗱 Boot Path	×
Define your boot Path	
TZEN=1 'Only Secure' project Boot Step#1	Graphical Boot Representation
O Secure Application	Non Secure Secure
O OEM immutable Root of Trust (OEM-IRoT)	
	ST-IROT
ST immutable Root of Trust (ST-iRoT)	STM32
BACK	NEXT FINISH CANCEL

Figure 253. Select ST-iRoT

Select Secure Application

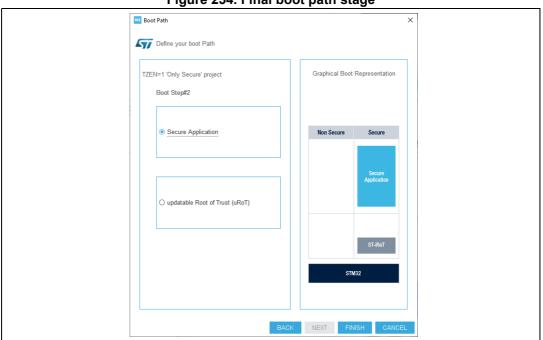


Figure 254. Final boot path stage

• Click "FINISH", the boot path configuration panel is displayed (see *Figure 255*), use it to configure the application, then press the GENERATE CODE button to generate the code for the selected toolchain

UM1718 Rev 47



132CubeMX ST-iROT.ioc*: STM32H5	73IIKxQ								- 0
MX .	File	Window	Help	💄 myST		(🖻 🗗 🖻	X 🔿 🎱 🖯	< 57
e 🔪 STM32H573IKxQ	ST-IROT.ioc - Project Ma	anager					GENER	ATE CODE	
Pinout & Con			lock Configuration		Project Manager			Tools	
	STM32CubeProgrammer Pa								
	Use Default Path	C:\Program			STM32 Browse				
	Debug Authentication								
		nerate DA Folder							
	Boot Path Selection	_							
	Select ST_IR	от 🔍							
	Use Default Reference	Location							
	Reference Location	C:\Users\kn		2Cube_FW_H5_V1.2.0\Projects	STM32 Browse				
	Config Files Location	C:\Users\kn	outb\BootpathProjects\ST-iROT\RO	T_Provisioning\STiROT	Edit Config Files				
	Linker					Boot Path Image			
		odate				boot Patri mage			
	Reinitialize Linker File(5)					Non Secure	Secure	
							Non Secure	Secure	
	Signature							Secure	
	Sign Binary(ies)							Application	
	Mapping								
		M33S						ST-IRoT	
	Start address								
	End address						ST	M32	

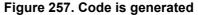
Figure 255. Boot path and Debug Authentication tab

Figure 256. Select the project structure

STM32CubeMX h573_boot_path_tz1.	ioc: STM32H573AIIxQ				-	
STM32	File	Window	Help	🌲 🚳 📑] 🕒 X 🔿 🍳 🔆	57
Home > STM32H573AllxC	a 🔪 h573_boot_path_tz	:1.ioc - Project Manager >		G	ENERATE CODE	
Pinout & Configur	ation Cl	ock Configuration	Project N	lanager	Tools	
Project	Use Default Refer	C:\Users\daymai	wa\STM32Cube\Repository\ST			
Code Generator	Config Files Location		wa\OneDrive - STMicroelectron	ics\Desktop\cubemx_projects	Boot Path Image	cure
Advanced Settings	(Mapping	M33S	M33NS			cure ication
Boot Path and Debug Authenticat	Start address End address	0xC000400 0xC01FFFF			STM32	IRoT

For this boot path, only the secure project is generated.





Generating user source code		Copying libraries files	
	ode Generation The Code is successfully gener C:/ST_MMS/CodeGeneration/B Project language : C Open Folder Open Project	ranch_6_8_Final_RC/STiRoT	

Additional directories, including the IDE environment are created.

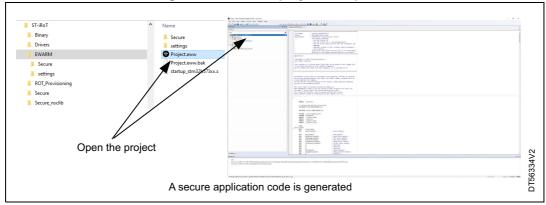


Figure 258. Secure project completed

Secure applications can be developed using the generated code skeletons.

The Post build command creates a secure compiled encrypted code for the provisioning.

Step 2: Code compilation

The generated binaries are automatically encrypted

- Open the project in the selected toolchain, for example IAR
 - Select: Project \rightarrow Option \rightarrow Build Actions
 - The links to the Trusted Package executable and to the secure application xml are filled automatically
 - Compile secure (right click on Project \rightarrow Rebuild all)



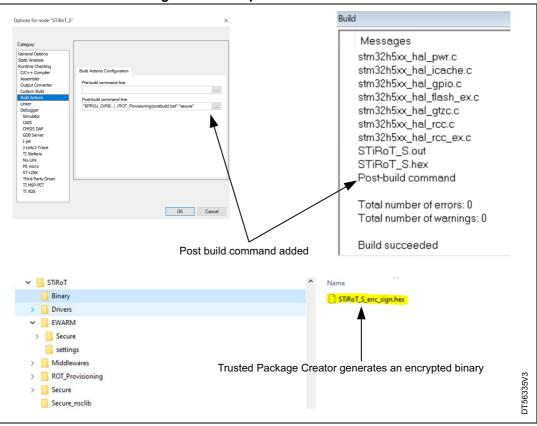


Figure 259. IDE post build commands

ST-iRoT board provisioning

The program cannot be flashed using an IDE, use the provisioning scripts found in the user environment.

• Double click on the provisioning.bat file (*Figure 260*)

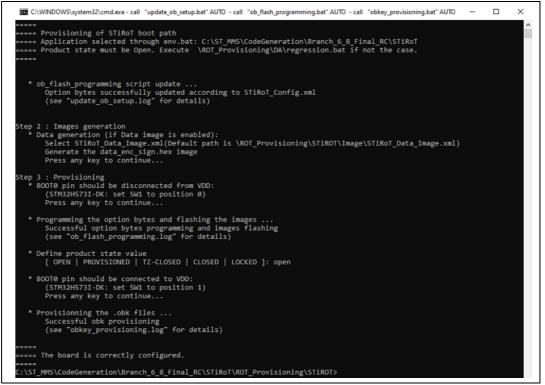


📜 ST-iRoT	^	Name
📜 Binary		Binary
📜 Drivers		Config
EWARM		📜 Image
📜 Secure		📕 Keys
📜 settings		ob_flash_programming.bat
ROT_Provisioning		obkey_provisioning.bat
DA		provisioning.bat
STIROT		update_appli_setup.bat
Binary		💁 update_ob_setup.bat
Config		
📕 Image		
📜 Keys		
Secure		
Secure_nsclib		
	 Binary Drivers EWARM Secure settings ROT_Provisioning DA STiROT Binary Config Image Keys Secure 	 Binary Drivers EWARM Secure settings ROT_Provisioning DA STIROT Binary Config Image Keys Secure

Figure 260. Board provisioning

- During provisioning, log files are generated to inform the user about the activity
- Follow the on-screen instructions (Figure 261)





In the user environment STM32CubeMX has generated an env.bat file containing the required data for provisioning, do not change it.



	Figure 262. Environment configuration	me
✓ 🔤 STiRoT	^ N	ame
Binary		DA
> Drivers		STIROT
V 📴 EWARM		env.bat
> Secure		
settings		
> 🔥 Middlewares		
✓ ROT_Provisioning		
> 🚺 DA		
STIROT		
Binary		
Config		
Image		
Keys		
> Secure		
Secure_nsclib		

Figure 262. Environment configuration file

4.18.5 How to configure an assembled boot path

The configuration described below is an example of an assembled boot path.

Prerequisites:

- Hardware: Discovery board STM32H573I-DK-REVC or later
- Required tools
 - Secure manager package, to be downloaded and installed from www.st.com
 - STM32CubeMX-6.9.0 or later
 - STM32 Trusted Package Creator (embedded in STM32CubeMX installation folder)
 - IAR Embedded Workbench rev 9.20.4 or later, and the patch in the STM32CubeH5 firmware (Version 1.1.0 or later), named EWARM/EWARMv8_STM32H5xx_Vx.x.zip.

Step 1: Configure flash_layout.h file

- Go to STM32Cube\Repository\STM32Cube_FW_H5_VX.X.X\Projects\ STM32H573I-DK\Applications\ROT\OEMiROT_Boot\Inc
- Open flash_layout.h
- Set the value of this define to 1 to assemble the secure and nonsecure binaries into one: #define MCUBOOT_APP_IMAGE_NUMBER 1.



132H5/3-UK_UEMIHUT_BOOK		
	Copyright (c) 2018 Arm Limited. All rights reserved.	
Project - STM32H573I-DK_OEMiROT_Boot	· · · · · · · · · · · · · · · · · · ·	
Application	* Licensed under the Apache License, Version 2.0 (the "License");	
	* you may not use this file except in compliance with the license.	
	You may obtain a copy of the License at	
He aes_altc	Tou may obtain a copy of the Etcense at	
Here and the services.c		
B boot hallc	http://www.apache.org/Licenses/LICENSE-2.0	
Here a booghand		
	* Unless required by applicable law or agreed to in writing, software	
He ecp_altc	* distributed under the License is distributed on an "AS IS" BASIS,	
HB C ecp_curves_alt.c	* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.	
He kevs_map.c	* See the License for the specific Language governing permissions and	
B low level com.c	* Limitations under the License.	
	L */	
- Iow_level_device.c		
🕀 🗈 low_level_flash.c	#ifndef _FLASH_LAYOUT_H_	
He low_level_obkevs.c	#define FLASH LAYOUT H	
B low level obkevs device.c	#detineFLASH_LAYOUT_H	
🕀 🗈 low_level_mg.c	I /* This header file is included from linker scatter file as well, where only a	
low_level_security.c	* limited C constructs are allowed. Therefore it is not possible to include	
- mpu_armv8m_drv.c	* here the platform retarget.h to access flash related defines. To resolve this	
-B C rsa_altc	* some of the values are redefined here with different names, these are marked	
	* with comment.	
He sha256_alt.c	- */	
Here is startup_stm32h5xx.c		
B stm32h5xx hal msp.c	/* Flash layout configuration : begin ************************************	
	/* Flash Layout configuration : pegin ****	
He lick.c	/* OEMiROT/STIROT_OEMuROT configuration */	
I 🖬 Doc	/*#define OEMUROT_ENABLE*/ /* Defined: the project is used for OEMiROT boot path */	
Drivers	/* Undefined: the project is used for STIROT_DEMuROT boot	path */
Middlewares	#define MCUBOOT OVERWRITE ONLY /* Defined: the FW installation uses overwrite method.	
	- UnDefined: The FW installation uses swap mode. */	
E 📫 Output		
	#define MCUBOOT_EXT_LOADER /* Defined: Use system bootloader (in system flash).	
	To enter it, press user button at reset.	
	To enter it, press user button at reset.	
	Undefined: Do not use system bootLoader. */	
	1	
	#define MCUBOOT_APP_IMAGE_NUMBER 1 /* 1: S and NS application binaries are assembled in one s	ingle image.
	2: Two separated images for S and NS application binari	rs. */
	#define MCUBOOT_S_DATA_IMAGE_NUMBER 0 /* 1: S data image for S application.	
	0: No S data image. */	
	or no 5 data diager y	
	#define MCUBOOT_NS_DATA_IMAGE_NUMBER 0 /* 1: NS data image for NS application.	
	#define MCUBODI_NS_DATA_IMAGE_NUMBER 0 /* 1: NS data image for NS application. 0: No NS data image. */	
	0: No NS data image. */	
	/* Flash layout configuration : end ***********************************	
	/* Total number of images */	
	- 1 Cl. Delanas sures luman (Delanas in Sures Dunans , Delanas e aist sures Dunans , Delanas	
ect	<	
Messages	I	File
tiv.c		
Project.out		
	I I	
OEMiROT_Boot.bin		
Post-build command		
run config Appli with windows executable		
2 · · ·		
T		
Total number of errors: 0		
Total number of warnings: 0		
-		
Build succeeded		
	¥	
a provide the second second second second		
	GE NUMBER 1 /* 1: S and NS application binaries are assembled :	n one single image.
#deline MCOBOOT APP IMA		
#deline MC0BOOT_APP_IMA	2. The computed images for C and WC	
#deline MC0BOOT_APP_IMA	2: Two separated images for S and NS application	binaries. */
#define MC0BOOT_APP_IMA	2: Two separated images for S and NS application	binaries. */

Figure 263. The flash_layout.h file

Step 2: Compile OEMiROT_Boot project

- Open OEMiROT_Boot with your preferred tool chain, and recompile the project.
 - The map.properties file is automatically updated (CODE_IMAGE_ASSEMBLY=0x01)
 - The image file (OEMiRoT_NS_Code_Image.xml) is automatically updated (firmware area size)

Step 3: OEMiROT (assembled) code generation

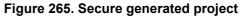
- Open STM32CubeMX application and create a new project with the H5 series (example: choose "STM32H573ZITxQ")
- Go to Project Manager window, and select secure and nonsecure application
- Add a name for the project and save it
- Go to Boot Path and Debug Authentication Panel: in Boot path selection, click on Select button
- Select OEM-iRoT in the boot path wizard window, and click Next
- Select Secure application, and click Finish



OEMiROT_Appli_TrustZone		^ Name	Date modified	Туре	Size
OEMiROT_Boot		Binary	6/14/2023 11:16 AM	File folder	
Binary		EWARM	6/14/2023 11:16 AM	File folder	
EWARM		Inc	6/14/2023 11:16 AM	File folder	
Inc		MDK-ARM	6/13/2023 5:20 PM	File folder	
MDK-ARM		Src	6/13/2023 5:20 PM	File folder	
Src		STM32CubeIDE	6/13/2023 5:20 PM	File folder	
STM32CubeIDE		APACHE-2.0.txt	6/9/2023 11:55 AM	Text Document	12 K
		auto_rot_update.bat	6/14/2023 11:12 AM	Windows Batch File	1 K
SMAK_Appli		auto_rot_update.sh	6/9/2023 11:55 AM	Shell Script	1 K
STiROT_Appli		map.properties	6/14/2023 11:17 AM	Properties Source	1 K
STiROT_Appli_TrustZone		📀 readme.html	6/9/2023 11:55 AM	Chrome HTML Do	61 K
STIROT_OEMuROT_Appli		README.md	6/9/2023 11:55 AM	Markdown file	6 K
	map.properties	- V			
	1 [BootPathType]		-		
	2 bootPath=OEMiRC	T			
	3 MCUBOOT OVERWRI	TTE ONLY=0x1			
	4 TRAILER SIZE=0x	40			
	5 CODE IMAGE ASSE	CMBLY=0x1			
	6 [Secure]				
	7 S CODE REGION S	TART=0xC018000			
	8 S CODE REGION S	SIZE=0x6000			
	9 <i>S</i> DATA=0x0				
	10 S DATA REGION S	START=0x0			
	11 S DATA REGION S	SIZE=0x0			
	12 HEADER SIZE=0x4	00			
	13 [NonSecure]				
	14 NS CODE REGION	START=0x801E000			
	15 NS CODE REGION	SIZE=0xA0000			
	16 NS DATA=0x0				
	17 NS DATA REGION	START=0x0			
	18 NS DATA REGION				
	19 HEADER SIZE=0x4	00			
	20 [Memory Region]				
	21 ROT REGION STAF				
	22 ROT REGION SIZE				
	23 SCRATCH REGION				
	24 SCRATCH REGION				
		E REGION START=0xBE000			
		DE REGION START=0x0			
		A REGION START=0x0			
		TA REGION START=0x0			
I	20 DOWINDOAD INS DAT	A REGION START-0X0			

Figure 264. The map.properties file

Generate and build the project



Project - IAR Embedded Workbench IDE - Arm 9.20.1	
ile Edit View Project ST-Link Tools Window Help	
) D C 🖸 🛍 🗶 🗎 🖬 🗳 D C	< Q > \$ HE < Q > R]
orkspace	Ф Ф Х
EMiRoT_Assembled_S	Options for node "OEMiRoT_Assembled_S"
Files	•
OEMiRoT_Assembled_S - OEMiRoT_Assembled_S	Category:
-⊞ ■ Application -⊞ ■ Doc	General Options
- Divers	Static Analysis
🕀 📠 Middlewares	Runtime Checking C/C++ Compiler Build Actions Configuration
	Assembler
	Output Converter Custom Build
	Build Actions Post-build command line:
	Linker Debugger
	Simulator
	CADI CMSIS DAP
	GDB Server
	I-jet J-Link/J-Trace
	TI Stellaris
	Nu-Link
	PE micro ST-LINK
	Third-Party Driver
	TI MSP-FET TI XDS
	OK Cancel
	OK Cancel



Vorkspace	★ ‡ X	
EMiRoT_Assembled_NS	✓	
Files OEMiRoT_Assembled_NS - OEMiRoT_ =9 = Application =9 = Doc =9 = Orivers =9 = Middlewares		
	Calegojy: General Options Static Analysis Runtime Checking C/C++ Compler Assembler Output Converter Output Converter Output Converter Custom Build Build Actions Configuration Pre-build command line: Post-build command line: Post-build command line: Post-build command line: "SPROJ_DIRS/././ROT_Provisioning/postbuild bat" "nonsecure" Post-build command line: "SPROJ_DIRS/././ROT_Provisioning/postbuild bat" "nonsecure" Simulator CADI CMSIS DAP GDB Server I yiet J citrk/J-Trace T Stellaris Nu-Link PE mircio ST-LINK Third-Party Driver TI MSP-FET TI XOS	•

Figure 266. Nonsecure generated project



OEMiRoT_Assembled_NS.out OEMiRoT_Assembled_NS.bin Postbuild command Wed Jun 111:3628 2023: [INF] ####################################
iRoT_Assembled_NS.bin'
Wed Jun 14 11:36:28 2023 : [INF] Extra padding will be added at the end of the first binary Wed Jun 14 11:36:28 2023 : [INF] Image assembly success
Wed Jun 14 11:36/28 2023. [INF] Final image size '29966'
STM32 Trusted Package Creator v2.14.0-B03
-pb C\ST_MMS\CodeGeneration\Branch_6_9_Beta_7\0EMiRoT_Assembled\ROT_Provisioning\0EMiROT\Images\0EMiROT_NS_Code_Image xml
Image generated successfully
Total number of errors: 0
Total number of warnings: 0
Build succeeded
Open the project folder. A Python script assembles both binaries (Secure, Non

- Open the project folder. A Python script assembles both binaries (Secure, Non Secure), then the TPC signs them:
 - Assembled_OEMiRot_Boot_Path_Example_assembled.bin \rightarrow File assembled by the Python script
 - Assembled_OEMiRot_Boot_Path_Example_enc_sign.hex \rightarrow File signed by the TPC





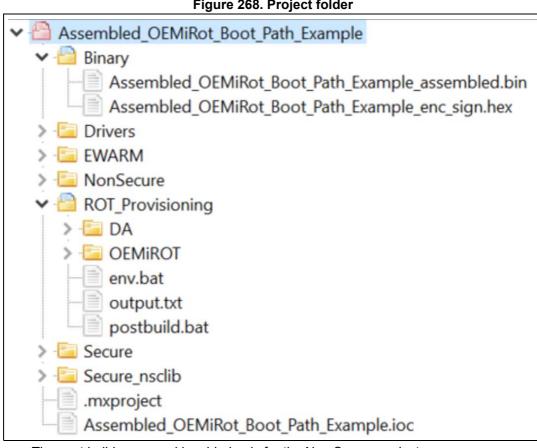


Figure 268. Project folder

The post build command is added only for the Non Secure project.

4.18.6 How to configure OEM-uRoT (STiRot uROT) boot path

- Select an STM32H57x MCU
- Create a project with TrustZone activated (TZEN = 1), see Figure 269
- In Project Manager, save the project, see Figure 270
- Go to "Boot Path and Debug Authentication" tab, and press the Select button, see Figure 271
- Select "ST immutable Root of Trust (ST-iRot)", then click "NEXT", see Figure 272
- Select "OEM updatable Root of Trust (OEM-uRoT)", then click "NEXT", see Figure 272
- Select "Secure Application", then click "FINISH", see Figure 273
- The panel of boot path configuration is displayed, use it to configure the boot path in the "Boot Path and Debug Authentication" tab, see Figure 274
- Generate and build the project, see Figure 277 and Figure 278



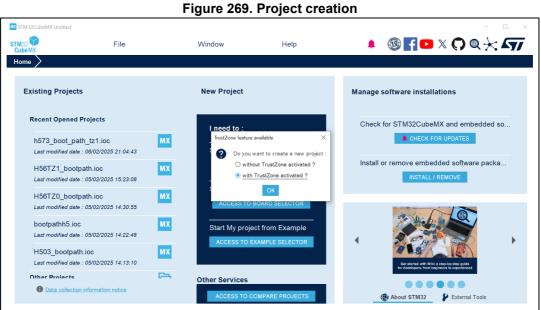


		Figure 270). Save the pro	ojeci	
STM32CubeMX Untitled: STM32H573Allx	Q File	Window	Help		- ° × • X () @ 🔆 / //
CubeMX Home STM32H573AllxQ			нер		
		inout & Configuration	Destad		
Pinout & Configuration	on	Clock Configuration	Project	t Manager	Tools
۹ ~	0	Save Project As		× view	
Categories A->Z		Save In: Lubemx_projects	· 🗸 🖓	6 🛱 🛗 🖿	
System Core	>	2m_II_generation 161G4FW test	cmake_security_flag		
Analog	>	bootpathh5	COMP_1M_code-ge	n	
Timers	>	build_error_C0 C011_code_gen_test	Comp_1m_gen COMP_EAMPLE_2N		
Connectivity	>	CM4	comp_example comp_example_m1		
Multimedia	>				
Security	>	Folder name: H57x_bootpat Files of Types STM32CubeN	h_project IX project Files		
Computing		Files of Types STM S2Cuber			
			Save	Cancel	
Middleware and Software Packs					
Trace and Debug				169 (Top view)	
Power and Thermal	>				
Utilities	>	● [] ④	L 4 II	= Q	✓ Unused GPIOs:



	ı ıy	jure 271. Bool pat	n anu uebuy	aumentica	lion panel	
MX STM32	CubeMX cubemx_projects.ioc: STM32H	I573AIIxQ				- 🗆 ×
STM32 CubeM	Fi	ile Window	Help		🕸 🗗 🗅 X 🗘	Q 🔆 🖅
Home	STM32H573AIIxQ	cubemx_projects.ioc - Project Mana	ger 🔪		GENERATE CODE	
	Pinout & Configuration	Clock Configurat	ion	Project Manager	Тс	ools
		STM32CubeProgrammer Path Use Default Path Debug Authentication Configure Generate DA Fol	rogram Files\STMicroelectronic	:s\STM32Cube\STM32CubePr	ogrammer\bin\STM Browse	
	Code Generator	/Boot Path Selection No boot path selecte	d (Nothing to configure) 🧟	(Default Boot Path)		
	Advanced Settings					
an	Boot Path d Debug Authentication					

Figure 271. Boot path and debug authentication panel



🚥 Boot Path		X Boot Path		×
Define your boot Path		Define your boot Path		
TZEN=1 'Dual Non Secure / Secure' project	Graphical Boot Representat	. TZEN=1 'Dual Non Secure / Secure' project	Graphical Boot Representat.	
Boot Step#1		- Boot Step#2		
	Non Secure Secur		Non Secure Secure	
O Secure Application		O Secure Application		
O OEM immutable Root of Trust (OEM-RoT)		OEM updatable Root of Trust (OEM-uRoT)		
			0EM-uRo	т
ST immutable Root of Trust (ST-iRoT)	ST-iRo		ST-iRoT	
	STM32		STM32	
		ICEL		
BACK	NEXT FINISH CA	BACK	NEXT FINISH CAN	ICEL



Boot Path		×
Define your boot Path		
TZEN=1 'Dual Non Secure / Secure' project	Graphical Boot	Representat
Boot Step#3	Non Secure	Secure
Secure Application		
	Application	Secure Application
O Secure Manager Non Secure Application		
		OEM-uRoT
		ST-IRoT
	STM	32
BACK	NEXT FINIS	H CANCEL

Figure 273. Final boot path stage

Figure 274. Boot path and debug authentication tab

) @ 🛧 ភ
	GENERATE CODE	
Project Manag	ger T	ools
ymarwa\STM32Cube\Repository\STM32Cu ymarwa\OneDrive - STMicroelectronics\Def		fig Files
ymarwa\STM32Cube\Repository\STM32Cu	ube_FW_H5_V1.5.0RC1\Proje Browse	Boot Path Image
M33NS		Application
0x801E400 0x80BDFFF		





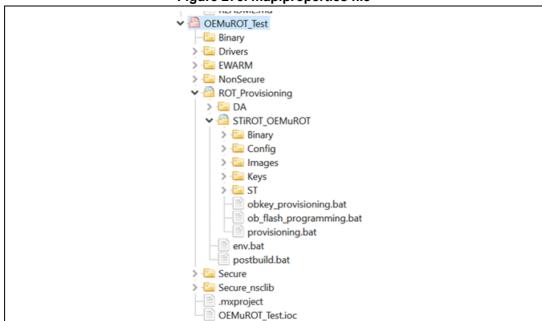
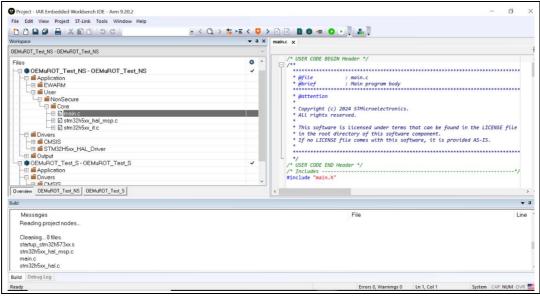


Figure 275. map.properties file

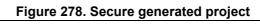






Project - IAR Embedded Workbench IDE - Arm 9.20.2		
File Edit View Project ST-Link Tools Window Help		
) 5 C 🖸 🛍 🗶 🔚 🖬 🖬 🗄	- < Q > \$ H < Q > @ 1 0 = 0 - 1 1 = 0 - 1 1 = 0 - 1 1 = 0 - 1 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0 1 = 0	
Workspace	Options for node "OEMuROT Test NS" ×	Т
OEMuROT_Test_NS - OEMuROT_Test_NS		
Files		
OEMuROT_Test_NS - OEMuROT_Test_NS Application Stm32h5xx_hal_msp.c Stm32h5xx_hal_msp.c Orivers Output OEMuROT_Test_S - OEMuROT_Test_S Drivers Output OEMuROT_Test_NS OEMuROT_Test_S Build	Category: General Options Static Analysis Runtime Checking (/C++ Compiler Assembler Output Converter Custom Build Build Actions Unker Debugger Simulator CADI CMSIS DAP GOB Server I-jet J-Link/ PE micro ST-LINK ST-LINK PE micro ST-LINK ST-LINK PE micro ST-LINK ST-LI	*** DG *** Dr. EE Ft is ***
Messages Reading project nodes	TI MSP-FET	
	TI XDS	
Cleaning 0 files.		
startup_stm32h573xx.s	OK Cancel	
stm32h5xx_hal_msp.c		
main.c		_
stm32h5xx hallo		





Vorkspace		> \$ HE < \$ > { } \$ @ # \$ @ # \$.	
DEMuROT_Test_NS - OEMuROT_Test_NS Files	Options for node "OEML	ROT_Test_S*	×
Messages Reading project nodes Cleaning0 files. startup_stm32h5xc_hal_msp.c minc	Category: General Options Static Analysis Runtime Checking (Cr++ Compiler Assembler Output Converter Custom Build Build Accom Debugger School OMSIS Dup OMSIS Dup OMSIS Dup OMSIS Dup OMSIS Dup OMSIS Dup Interfer Ti Stellaris Ni-Link PErmicro Ti Stellaris Ti Stellaris Ti Stellaris Ti Stellaris Ti Stellaris Ti Stellaris Ti Stellaris Ti Stellaris Ti Stellaris	Build Actions Configuration Pre-build command line: Post-build command line: SPROJ_DIRS/.J_ROT_Provisioning/postbuild baff "see	bdy onics. terms that can be found in the LICENSE file ftuare component. is software, it is provided AS-IS. Lin Lin



4.18.7 How to configure ST-iRoT boot path with STM32H7RS devices

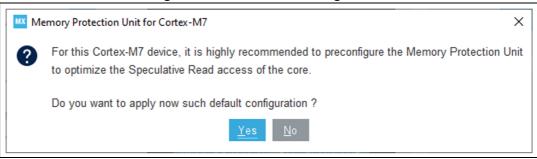
Go through the following steps:

- 1. Select an STM32H7S3Vx MCU (Figure 279)
- 2. A popup (see *Figure 280*) asks to preconfigure the Memory Protection Unit. It is recommended to optimize the speculative read access of the core. Select "Yes" to keep the default configuration.
- 3. In Project Manager Window, check only "Appli Project", name the project, and save it (*Figure 281*).
- 4. Go to "Boot Path and Debug Authentication" tab and press the Select button (*Figure 282*).
- 5. Select "ST immutable Root of Trust (ST-iRoT)", then click "NEXT" (*Figure 283*).
- 6. Select "Application", then click "FINISH" (Figure 284).
- 7. The panel of boot path configuration is displayed (see *Figure 285*), use it to configure the boot path in the "Boot Path and Debug Authentication" tab.
- 8. Generate and build the project (see *Figure 286*).

New Project from a MCU/MPU									
	ample Selector	Cross Sele	ctor						
cu/MPU Filters] _	Features	Block Diagram	Docs & Res	ources CAD Resources	📑 Datasheet	📑 Buy	Start Project
Commercial Part Number STM32H7S3V8H6			STM32H7 Series						
	+-		STM32H7	7S3V8H6		mance Arm Cortex-M7 DSP, cache, USB HS		, 64KB Bootflash, 6	20KB
PRODUCT INFO	~		ACTIVE		Unit Price for 10	(U (US\$):4.1131			
Segment	>			ass production			ÈT	FBGA 100 8x8x1.2 P 0.8 mm	
Series	>		OTN20U70w/	devices are be	and on the high	performance Arm [®] Cortex [®]	M7 22 bit DISC a	are energing at up to CC	O MUT The
Line	>		Cortex -M7 cor	re features a flo	ating point unit (FPU) which supports Arm d	ouble-precision (IE	EE 754 compliant) and	single-
Marketing Status	>					a types. The Cortex -M7 co support a full set of DSP ins			
Price	<u>></u>		enhance annli	ration security				,, ,	
Package	~~~	MCUs/N	IPUs List: 1 item						📩 Export
Core			• Comm	hercial Part No 🚊	Part No	Reference Marketl	tee for NR x Board	× Package × Planh	× RAM
Coprocessor	> >	1	STM32H7	S3V8H0	STM32H7S3V8	STM32H7S3V8Hx Active 4.113			
						of moen contractice 4.110	1	TFBGA 10 64 kBytes	620 kBytes
							1	TFBGA 10 04 kBytes	620 kBytes
MEMORY	~						1	TFBGA 10 04 kBytes	620 kBytes
MEMORY Flash = 64 (kBytes)							1	TFBGA 10 04 xBytes	620 kBytes
MEMORY Rash = 64 (kBytes) 64							1	THBOA 10 OH kBytes	620 kBytes
MEMORY Flash = 64 (B)r6s) 64 EPPROM = 0 (Byr6s) 0							1	Track Tu., 64 abytes	620 kBytes
MEMORY Rash = 64 (Bytes) 66 86 87 86 80 80 80 80 80 80 80 80 80 80							1	Traux 10 64 soyles	620 kBytes
MEMORY Rash = 64 (Bytes) 64 EPROM = 0 (Bytes) 0							1	Traux 10 64 soyles	620 kBytes

Figure 279. Boot path project







> stm32H7S3V8Hx >	H7S_StiRot_Demo.ioc - Project Ma	inager >	l .			GENERATE CODE
Pinout & Configu	ration	Clock Configuration Project M		roject Manager		Tools
	Project Settings					
	Project Name	H7S_StiRot_Demo				
	Project Location	C:\STM32CubeMX_Projects			Browse	
	Project Structure	🗌 Boot Project 🗹 Appli Project 🗌 E:	ttMemLoader Project			
	Application Structure	Advanced		✓ Do not generative	ate the main()	
	Toolchain Folder Location	C:\STM32CubeMX_Projects\H7S_StiRot_	Demo\			
	Toolchain / IDE	EWARM	Min Version V9.30	✓ 🔲 Generate Und	ler Root	
	Linker Settings					
		Boot App		ExtMemLoader		
	Minimum Heap Size	0×200 0×200		0×200]	
	Minimum Stack Size	0x400		0x400]	
	Thread-safe Settings					
	Enable multi-threaded support					
Advanced Settings		P. J. M. M. C. 1911	F		~	
	Thread-safe Locking Strategy	Default – Mapping suitable strategy depe	nding on RTOS selection.			
	Enable multi-threaded support					
	Thread-safe Locking Strategy	Default – Mapping suitable strategy depe	nding on RTOS selection.			
	Mcu and Firmware Package					
	Mcu Reference	STM32H7S3V8Hx				
	Firmware Package Name and Version	STM32Cube FW_H7RS V1.1.0RC1				
	Use Default Firmware Location					

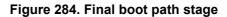
Figure 282. Select the project

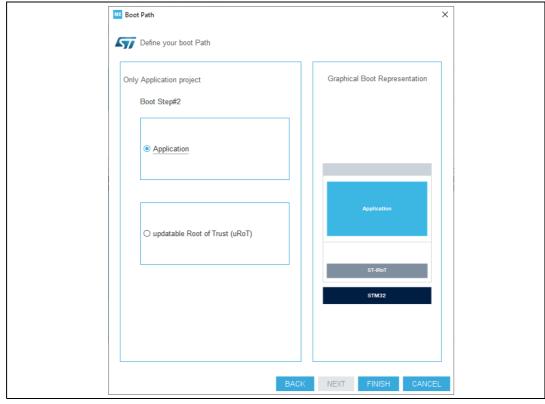
Home > STM32H7S3V8Hx > H	H7S_StiRot_Demo.ioc - Pr	oject Manager >		GENERATE CODE
Pinout & Configuration		Clock Configuration	Project Manager	Tools
Project	STM32CubeProgrammer Path Use Default Path Debug Authentication Configure Gen	C:Program Files\STMicroelectronics\STM32Cub	atSTM32CubeProgrammeribintSTM3] Browse	
	Boot Path Selection	path selected (Nothing to configure)		
Code Generator			Application (Default Boot Path)	
Advanced Settings				
Boot Path and Debug Authentication				



Boot Path
Only Application project Graphical Boot Representation Boot Step#1 Application O Application O OEM immutable Root of Trust (OEM-iRoT) ST-IROT ST immutable Root of Trust (ST-iRoT) STM32
Boot Step#1 Application O DEM immutable Root of Trust (OEM-iRoT) ST-IRoT STM32
Application OEM immutable Root of Trust (OEM-iRoT) ST-iRoT STM32
OEM immutable Root of Trust (OEM-iRoT) ST-IRoT STM32 ST immutable Root of Trust (ST-iRoT)
ST-IRoT ST immutable Root of Trust (ST-iRoT)
ST-IROT ST immutable Root of Trust (ST-iRoT)
ST immutable Root of Trust (ST-iRoT)
ST immutable Root of Trust (ST-iRoT)
BACK NEXT FINISH CANCEL
BACK NEXT FINISH CANCEL

Figure 283. First boot path stage







Home > STM32H7S3V8Hx >	GENERATE CODE			
Pinout & Configur	ation	Clock Configuration	Project Manager	Tools
Project	STM32CubeProgrammer Path Use Default Path Debug Authentication Configure Boot Path Selection Select ST_IROT @	E:Program Files/STMcroelectronce/STM32Cube	STA13CubeProgrammerbin/STA32 Browse	
Code Generator	Use Default Reference Location Reference Location Config Files Location	C. Wsen beknall STM32Cube Repository (STM32C C. NSTM32Cube MX_Projects W7S_Stiftot_Demo RV		
Advanced Settings	Reinitialize Linker File(s)		Loos L ans mage	Applaate
Boot Path and Debug Authentication	/Mapping Start address End address	Appi 0x2400400 0x2401FFFF		17.000T

Figure 285. Boot path and debug authentication panel

Figure 286. Generate the code

Home $ angle$ STM32H7S3V8Hx $ angle$ I	H7S_StiRot_Demo.ioc - Project M	Manager 🔪		GENERATE CODE
Pinout & Configura	ation	Clock Configuration	Project Manager	Tools
Project	STM32CubeProgrammer Path Use Default Path Debug Authentication Configure Generate DA /Boot Path Selection Select ST_IROT Q	E-Program Files/STMicroelectronics/STM32Cube/S Folder	TM32CubeProgrammer/bin/STM3 Browse	
Code Generator	Use Default Reference Location Reference Location Config Files Location Linker		tully generated under : Digloctulft75_StiRot_Demo	
Advanced Settings	Reinitialize Linker File(s)	Open Folder Open P	nged Close	Agatema
Boot Path and Debug Authentication	/Mapping Start address End address	Appli 0x24000400 0x2401FFFF		stati

Figure 287. Application IDE directories

✓ ☐ H7S_StiRot_Demo
> 🔚 Appli
— 🔚 Binary
> 🛅 Drivers
> 🛅 EWARM
ROT_Provisioning
🗸 😓 DA
> 🔚 Binary
🗸 🔤 STIROT
> 🔚 Binary
> 🔚 Config
- inv.bat
postbuild.bat
.mxproject

UM1718 Rev 47



4.19 User authentication

All downloads of ST packages (such as Cube firmware, X-Cube) through STM32CubeMX must be authenticated with a my.st.com account, which can be created on *www.st.com*, or directly from within the tool (see *Section 4.19.2*).

4.19.1 Login with an existing my.st.com account

The login form is accessible when the user performs any operation that requires or recommends package installation.

Removal of the connection status from the home page.

132CubeMX Untitled			- a ×
32 DeMX	File	Window Help	🔺 🔺 🐼 🚺 🕺 🕹 🖡 🕨 X 🕻
me >			
Existing Projects		New Project	Manage software installations
Recent Opened Projects		I need to :	Check for STM32CubeMX and embedded soft
COMP_EAMPLE_2M.ioc		Start My project from MCU	CHECK FOR UPDATES
COMP_EXP_1M.ioc	MX	ACCESS TO MCU SELECTOR	Install or remove embedded software packages
comp_example_m1.ioc	MX	Start My project from ST Board	
comp_example.ioc	MX	Start My project from Example	57
BLE_p2pServer.ioc Last modified date : 02/01/2025	MX 11:23:05	ACCESS TO EXAMPLE SELECTOR	The first high-performance STM32 MCU with At acceleration
Other Projects Data collection information in	otice		About STM32 External Tools
TM32CubeMtK Untitled*: STM32MP231AA A12 bbeMX orme STM32MP231AAJx	ix File	Window Help	- d × * 🌲 🎯 🗗 🗗 X 🔘 Q 🛧 🟹 GENERATE CODE
A32 ubeMX	File	Window Help New Project	
AB2 IDEMX orme STM32MP231AAJX	File		GENERATE CODE
Existing Projects	File	New Project	GENERATE CODE Manage software installations Check for STM32CubeMX and embedded soft
Existing Projects Recent Opened Projects COMP_EAMPLE_2M.ioc	File Untitled - RIF S 6.59:76 MX	New Project I need to : Start My project from MCU ACCESS TO MCU SELECTOR	GENERATE CODE Manage software installations Check for STM32CubeMX and embedded soft
Existing Projects Recent Opened Projects COMP_EAMPLE_2M.loc Lest modified date : 06/01/2022 COMP_EXP_1M.loc	File Untitled - RIF 5 8:59:76 5 8:43:54 MX	New Project I need to : Start My project from MCU	GENERATE CODE Manage software installations Check for STM32CubeMX and embedded soft CHECK FOR UPDATES Install or remove embedded software packages NSTALL / REMOVE
Existing Projects Recent Opened Projects COMP_EAMPLE_2M.ioc Lest modified date : 06/01/2022 COMP_EXP_1M.ioc Lest modified date : 06/01/2022 comp_example_m1.ioc	File Unblied - RIF 5 8.59:16 5 8.43:54 5 14:33:44 MX	New Project I need to : Start My project from MCU ACCESS TO MCU SELECTOR Start My project from ST Board	GENERATE CODE Manage software installations Check for STM32CubeMX and embedded soft CHECK FOR UPDATES Install or remove embedded software packages
Existing Projects Existing Projects COMP_EAMPLE_2M.ioc Lest modified date : 06/01/2022 COMP_EXP_1M.ioc Lest modified date : 06/01/2022 comp_example_m1.ioc Lest modified date : 03/01/2022 comp_example_ioc	File Unblied - RIF 5 8.59:16 5 8.43:54 5 14:33:44 MX	New Project I need to : Start My project from MCU ACCESS TO MCU SELECTOR Start My project from ST Board ACCESS TO BOARD SELECTOR Start My project from Example	GENERATE CODE Manage software installations Check for STM32CubeMX and embedded soft CHECK FOR UPDATES Install or remove embedded software packages NSTALL / REMOVE
Existing Projects Existing Projects COMP_EAMPLE_2M.loc Lest modified date : 06/01/2021 COMP_EXP_1M.loc Leat modified date : 06/01/2021 comp_example_m1.loc Leat modified date : 03/01/2021 comp_example_loc Lest modified date : 02/01/2021	File Untitled - RIF 5 8:59:16 5 8:43:54 5 14:33:44 5 14:33:44 5 22:58:19 E	New Project I need to : Start My project from MCU ACCESS TO MCU SELECTOR Start My project from ST Board ACCESS TO BOARD SELECTOR Start My project from Example	GENERATE CODE Manage software installations Check for STM32CubeMX and embedded soft CHECK FOR UPDATES Install or remove embedded software packages NSTALL / REMOVE

Figure 288. Home page without the login form



Authentication is required for operations involving ST packages, whether performed outside or within a project:

- External operations:
 - Installing software via Help and Shortcut menus.
 - Installing software using the Example Selector.
- Internal operations:
 - Installing software through the Embedded Software Manager panels.
 - Installing software via the Software Component Selector panel.
 - Installing software during code generation.
 - Installing recommended software when loading an .ioc file.

Examples of operations that need user authentication:

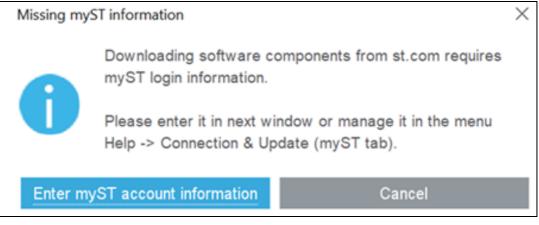
- User authentication to install or remove software packages (*Figure 289*).
- If not authenticated, clicking the install button under the Embedded Software packages Manager panel to get a new version of a given package. A window titled Missing myST information appears to indicate that the user needs to be authenticated (*Figure 290*).
- The user can provide myST login Information from two locations:
 - By clicking "Enter myST account information" button in the appeared window.
 - By clicking the same button under the help menu, more precisely in myST tab under "Connection & updates" (*Figure 291*).
- The field myST in STM32CubeMX display is a new UI element added to the "Connection & Updates" window (under the Help menu and previously named Updater Settings) to allow users to perform authentication when it is needed.
- After clicking on the "Enter myST account information" button, a user authentication dialog window appear (*Figure 292*).
- If the login action is performed successfully, myST display changes as illustrated in the *Figure 293*.
- If the user wants to save their credentials, they can check "Remember me on this computer" so that they do not need to authenticate again during the next sessions.
- If the user wants to sign out, they should click on "Clear myST account information for this session." If the user has clicked on "Remember me on this computer," the button "Clear myST account information for this session" is changed to "Clear myST account information for this computer." The login action can be blocked if the user provides wrong credentials or keeps the login fields empty (*Figure 294*).



TM32CubeMX Untitled: STM32MP233		- 0
M32	Embedded Software Packages Manager	🕸 🗗 🔼 X 🚺 Q 🔆 /
ome STM32MP233AAJ	STM32Cube MCU Packages and embedded software packs releases	GENERATE CODE
	Releases information was last refreshed 13 days ago.	
Pinout & Configuration	n portGmbH quantropi wotfSSL ITTIA_DB Infineon RealThread SEGGER WES emotas 477 STM32Cube MCU Packages 477 STMicroelectronics Cesanta EmbeddedOffice	r Tools
L	Description Installed Available	🍥 🚺 🖸 Pi 🕅 S
Categories A->Z	▼ STM32MP2	
System Core		>
Analog	STM32Cube MCU Package for STM32MP2 Series (Size : 1.2.0RC11	> .
Timers	Details	>
THINKS .	STM32CubeMP2 Firmware Package V1.2.0RC11 / 21-Nov-2024	
Connectivity	• v1.2.0RC11 release of STM32CubeMP2 Firmware Install action needs the user	<u> </u>
Multimedia	authentication	>
Security		>
	From Local From Url Refresh Install Remove Close	
Computing M32CubeMX Untitled: STM32MP233A		> TFBGA361 (Top v
M32CubeMX Untitled: STM32MP233A 132 13eMX	AAlx	© f □ X Q Q ★
M32CubeMX Untitled: STM32MP233A beMX me STM32MP233AA.Jx	STM32Cube MCU Packages and embedded software packs releases Releases Information was last refreshed 13 days ago.	
M32CubeMX Untitled: STM32MP233A 32	STM32Cube MCU Packages and embedded software packs releases STM32Cube MCU Packages and embedded software packs releases Releases Information was last refreshed 13 days ago. portGmbH portGmbH	
M32CubeMX Untitled: STM32MP233A beMX me STM32MP233AA.Jx	STM32Cube MCU Packages and embedded software packs releases STM32Cube MCU Packages and embedded software packs releases PortCombH quartorpi wolfSL TITLA Define RealTrace RealTrace For RealPackages 477 STM32Cube MCU Packages 477 STM32Cube MCU P	
M32CubeMX Unitiled: STM32MP233# beMX mme STM32MP233AAJx Pinout & Configuration	STM32Cube MCU Packages and embedded software packs releases STM32Cube MCU Packages and embedded software packs releases Releases Information was last refreshed 13 days ago. portGmbH quantropi wolfSSL portGmbH GasThread SEGGER WES emotas G7 STM32Cube MCU Packages A77 STMIcroelectronics Cesanta EmbeddedOffice Description Installed Available	
M32CubeMX Unitiled: STM32MP2334 beMX mre STM32MP233AAJx Pinout & Configuration alegories A->Z	STM32Cube MCU Packages and embedded software packs releases STM32Cube MCU Packages and embedded software packs releases PortCombH quartorpi wolfSL TITLA Define RealTrace RealTrace For RealPackages 477 STMIcroelectronics Cesanta EmbeddedOffice	
M32CubeMX Unitiled: STM32MP2334 BeMX me STM32MP233AAJx Pinout & Configuration ategories AZ System Core	STM32Cube MCU Packages and embedded software packs releases STM32Cube MCU Packages and embedded software packs releases Releases Information was last refreshed 13 days ago. portGmbH quantropi wolfSSL portGmbH GasThread SEGGER WES emotas G7 STM32Cube MCU Packages A77 STMIcroelectronics Cesanta EmbeddedOffice Description Installed Available	
M32CubeMX Undeket STM32MP2334 SEC me STM32MP233AAJx Pinout & Configuration ategories A-2 System Core Analog	STM32Cube MCU Packages and embedded software packs releases STM32Cube MCU Packages and embedded software packs releases STM32Cube MCU Packages and embedded software packs releases PortGmbH quartopi wolfSL TTIL_0B Infreeon RealTrees STM32Cube MCU Packages 477 STMIcroelectronics Cesanta EmbeddedOffice Description Installed Available STM32Cube MCU Package for STM32MP2 Series 1.0.0 1.0.0	GENERATE CODE
M32CubeMX Untilled: STM32MP2334 BeMX me STM32MP233AAJx Pinout & Configuration ategories A->Z System Core Analog		
M32CubeMX Undekd: STM32MP233A STM32MP233AAJx Pinout & Configuration Megones A->Z System Core Analog Timers		GENERATE CODE
M3ZCubeMX Unitled: STM32MP233# bbeMX mme STM32MP233AAJx Pinout & Configuration		GENERATE CODE
M32CubeMX Undek: STM32MP233A STM32MP233AAJx Pinout & Configuration Ategories A->Z System Core Analog Timers Connectivity		

Figure 289. Install or remove a software package

Figure 290. Missing myST information





M32 CubeMX Home STM32MP233AAJX	Simiscube mod Packages and embedded software packs releases	
	Releases information was last refreshed 13 days ago.	
Pinout & Configuration	n porGmoH quantropi voltSL ITTIA_DB Infineon RealThread SEGGER WES emotas 470 STM32Cube MCU Packages 477 STMicroelectronics Cesanta EmbeddedOffice	Tools
2	Description Installed Available	🎯 🚺 🖸 Pi 🏴 Syst
Categories A->Z	▼ STM32MP2	
System Core	- <u>-</u>	<u> </u>
Analog	STM32Cube MCU Package for STM32MP2 Series (Size : 1.2.0RC11	>
Timers	Details	
Connectivity	STM32CubeMP2 Firmware Package V1.2.0RC11 / 21-Nov-2024	
Multimedia	v1.2.0RC11 release of STM32CubeMP2 Firmware Install action needs the user authentication	
Wattheata		
Security	From Local From Url Refresh Install Remove Close	<u>></u>
TM32CubeMX Untitled: STM32MP233A	WE had a feature the second se	- 0
TM32CubeMX Untitled: STM32MP233A M32 bbeMX ome STM32MP233AAJx	STM32Cube MCU Packages and embedded software packs releases	
M32	Embedded Software Packages Manager STM32Cube MCU Packages and embedded software packs releases Releases Information was last refreshed 13 days ago.	
max ome STM32MP233AAJx Pinout & Configuration	STM32Cube MCU Packages and embedded software packs releases STM32Cube MCU Packages and embedded software packs releases Releases Information was last refreshed 13 days ago. PortGmbH quartropi TTTIA_DB Infineon RealThread SEGGER WES emotas ject Manager	Tools
max ome STM32MP233AAJx Pinout & Configuration	STM32Cube MCU Packages 477 STMicroelectronics Cesanta EmbeddedOffice STM32Cube MCU Packages 477 STMicroelectronics Cesanta EmbeddedOffice	Tools
me STM32MP233AAJx Pinout & Configuration	Embedded Software Packages Manager STM32Cube MCU Packages and embedded software packs releases Releases Information was last refreshed 13 days ago. PortGmbH quartropi wolfSSL TTTLA DB Infineon RealThread SEGGER WES emotas r#7 STM32Cube MCU Packages ## STMicroelectronics Cesanta EmbeddedOffice Description Installed Available	Tools
Mac Configuration	Embedded Software Packages Manager STM32Cube MCU Packages and embedded software packs releases Releases Information was last refreshed 13 days ago. PortGmbH quartropi wolfSSL TTTLA DB Infineon RealThread SEGGER WES emotas r#7 STM32Cube MCU Packages ## STMicroelectronics Cesanta EmbeddedOffice Description Installed Available	Tools
ome STM32MP233AAJx Pinout & Configuration Stegories A->Z System Core		Tools
STM32MP233AAJx Pinout & Configuration Categories A->Z System Core Analog	Impedded Software Packages Manager X STM32Cube MCU Packages and embedded software packs releases + - Releases Information was last refreshed 13 days ago. + - ImportGmbH Quantropi VITIA_DB Infineon Releases Information was last refreshed 13 days ago. + - ImportGmbH Quantropi ImportGmbH Releases for STM32MP2 Series 1.0.0 ImportGmbH ImportGmbH ImportGmbH ImportGmbH ImportGmbH ImportGmbH ImportGmbH ImportGmbH ImportGmbH ImportGmbH ImportGmbH ImportGmbH	Tools
STM32MP233AAJx Pinout & Configuration Categories AZ System Core Analog Timers		Tools Contraction Contractio
STM32MP233AAJx Pinout & Configuration tategories A->Z System Core Analog Timers Connectivity	Impedded Software Packages Manager X STM32Cube MCU Packages and embedded software packs releases + - Releases Information was last refreshed 13 days ago. + - ImportGmbH Quantropi VITIA_DB Infineon Releases Information was last refreshed 13 days ago. + - ImportGmbH Quantropi ImportGmbH Releases for STM32MP2 Series 1.0.0 ImportGmbH ImportGmbH ImportGmbH ImportGmbH ImportGmbH ImportGmbH ImportGmbH ImportGmbH ImportGmbH ImportGmbH ImportGmbH ImportGmbH	Tools

Figure 291. Authentication from myST tab

Figure 292. User Authentication Dialog

Pinout & Configura	tion	Clock Configuration	RIF	Project Manag
Analog Timers Connectivity Multimedia Security Computing		User Authentication Dialog Already registred users don't need to create account they have just by Already registered? Enter your e-mail address and password to login your myST user. E-mail address	New users need first to createan account the login with the created account New user? myST brings you a set of persona features: • Participate to ST Events • Stay informed with ST eNewsle	
Middleware and Software Packs Trace and Debug Power and Thermal Utilities		Password	Get help with ST Online Suppor Discuss on the ST Community Benefit from our Online Design Download Software Order free samples	rt
Other		CRemember me on this computer.	Manage your weekly product up Buy ST Products & Tools Create Account	odates

UM1718 Rev 47



Connection & Updates Updater Settings Connection Parameters myST	×
Downloading software components from st.com requires myST login information. This information can be stored for use in future STM32CubeMX sessions.	
Clear myST account information from this computer	
ОК	Cancel

Figure 293. The myST display after login

Figure 294. Blocked login cases

Enter your e-mail address and password to login your myST user.	myST brings you a set of personalized features:
The email address or password you provided does not match our	Participate to ST Events
records. You entered user-ext@st.com	 Stay informed with ST eNewsletters
E-mail address	 Get help with ST Online Support
	 Discuss on the ST Community
	Benefit from our Online Design Tools
	 Download Software
Password	• Order free samples
	 Manage your weekly product updates
	 Buy ST Products & Tools
Remember me on this computer.	
Login	Create Account
User Authentication Dialog	
User Authentication Dialog	New user?
Already registered?	New user?
Already registered?	myST brings you a set of personalized
Already registered?	
Already registered?	myST brings you a set of personalized
Already registered? Enter your e-mail address and password to login your myST user.	myST brings you a set of personalized features:
Already registered? Enter your e-mail address and password to login your myST user. Please enter User ID	myST brings you a set of personalized features: • Participate to ST Events
Already registered? Enter your e-mail address and password to login your myST user. Please enter User ID Please enter Password	myST brings you a set of personalized features: • Participate to ST Events • Stay informed with ST eNewsletters
Already registered? Enter your e-mail address and password to login your myST user. Please enter User ID Please enter Password	myST brings you a set of personalized features: Participate to ST Events Stay informed with ST eNewsletters Get help with ST Online Support
Already registered? Enter your e-mail address and password to login your myST user. Please enter User ID Please enter Password E-mail address	 myST brings you a set of personalized features: Participate to ST Events Stay informed with ST eNewsletters Get help with ST Online Support Discuss on the ST Community
Already registered? Enter your e-mail address and password to login your myST user. Please enter User ID Please enter Password	 myST brings you a set of personalized features: Participate to ST Events Stay informed with ST eNewsletters Get help with ST Online Support Discuss on the ST Community Benefit from our Online Design Tools
Already registered? Enter your e-mail address and password to login your myST user. Please enter User ID Please enter Password E-mail address	myST brings you a set of personalized features: • Participate to ST Events • Stay informed with ST eNewsletters • Get help with ST Online Support • Discuss on the ST Community • Benefit from our Online Design Tools • Download Software • Order free samples
Already registered? Enter your e-mail address and password to login your myST user. Please enter User ID Please enter Password E-mail address	myST brings you a set of personalized features: • Participate to ST Events • Stay informed with ST eNewsletters • Get help with ST Online Support • Discuss on the ST Community • Benefit from our Online Design Tools • Download Software • Order free samples • Manage your weekly product updates
Already registered? Enter your e-mail address and password to login your myST user. Please enter User ID Please enter Password E-mail address	myST brings you a set of personalized features: • Participate to ST Events • Stay informed with ST eNewsletters • Get help with ST Online Support • Discuss on the ST Community • Benefit from our Online Design Tools • Download Software • Order free samples
Already registered? Enter your e-mail address and password to login your myST user. Please enter User ID Please enter Password E-mail address Password	myST brings you a set of personalized features: • Participate to ST Events • Stay informed with ST eNewsletters • Get help with ST Online Support • Discuss on the ST Community • Benefit from our Online Design Tools • Download Software • Order free samples • Manage your weekly product updates • Buy ST Products & Tools
Already registered? Enter your e-mail address and password to login your myST user. Please enter User ID Please enter Password E-mail address Password Remember me on this computer.	myST brings you a set of personalized features: • Participate to ST Events • Stay informed with ST eNewsletters • Get help with ST Online Support • Discuss on the ST Community • Benefit from our Online Design Tools • Download Software • Order free samples • Manage your weekly product updates



4.19.2 Create a my.st.com account

The account can be created through STM32CubeMX:

- Click on "Create Account" button (*Figure 292*)
- Fill the account creation form (*Figure 295*)
- Click on "Register" button to create a new my.st.com account (*Figure 295*)

Figure 295. Account creation form

User Authentication Dialog
my.st.com account creation
Your Profile
Salutation:
Select one •
First name*:
Last name*:
Email*:
Email Confirmation*:
Function*:
Select one ·
Company/University*:
Industry*:
Select one
Country/Region*:
Select one ·
State/Province:
Select one
ZipiPostcode*:
Phone Number:
Please review our Privacy Statement that describes how we process your profile information and how to assert your personal data protection rights
Register
1 mg/seco

4.19.3 Password restoration

If you forget the password, it can be reset by following the steps below.

- 1. Go to the login page via myST tab.
- 2. Click on the "Forgot Password?" link located below the password field.



3. Enter the email address associated with your account in the dedicated field (*Figure 296*).

×	
New user?	
myST brings you a set of personalized features:	
Participate to ST Events Stav informed with ST eNewsletters	
Get help with ST Online Support	
Discuss on the ST Community	
Benefit from our Online Design Tools Download Software	
Order free samples	
Create Account	
	New user? myST brings you a set of personalized features: • Participate to ST Events • Stay informed with ST eNewsletters • Get help with ST Online Support • Discuss on the ST Community • Benefit from our Online Design Tools • Download Software • Order free samples • Manage your weekly product updates • Buy ST Products & Tools

Figure 296. Enter the email address

4. Click on the "Reset my password" button (*Figure 297*). You will receive an email containing a link to reset the password. If you do not receive the email within the next few minutes, check your spam folder or contact our support team.



User Authentication Dialog	<
Forgot your password?	
Enter your e-mail address and we will send you a message to reset your password.	
Email address*	
enter your email address	
Reset my password	
-	
-	



- 5. Click on the link in the email to access the password reset page.
- 6. Enter a new password in the dedicated field. Make sure that the new password is strong and secure.
- 7. Confirm the new password by entering it again in the confirmation field.
- 8. Click on the "Submit" button to save the new password.
- 9. Log in to the application, using the new password.

Figure 298. Reset password form

Reset password
Please order your email address and the new password. E-mail address* New Password* Confirm New Password* Confirm New Password* Confirm New Password* Contain characters in length O to contain spaces Contain characters from at least three out of the following four categories: E-mail address* Contain characters (A-Z) E-mglish upercase characters (A-Z) E-mglish towercase characters (a-Z) D d- gligtile Contain characters Contain characters Contain characters Contain character

If you suspect that your identity has been stolen, or that your account has been compromised, change the password immediately to protect your account. Follow the reset procedure described above to change it.

It is recommended to contact your ST referent to report any suspicious activity on your account, and take necessary measures to protect it.

If you experience difficulties resetting the password, contact your ST referent for assistance.

4.19.4 Authentication through command line interface

To facilitate the integration of authentication functionality with other tools, STM32CubeMX provides a command-line mode to login with an existing my.st.com account.

Use the following command lines:

On Windows:

```
cd <STM32CubeMX installation path>
jre\bin\java -jar STM32CubeMX.exe login <email_adress> <password>
<remember_me>
```

On Linux and macOS:

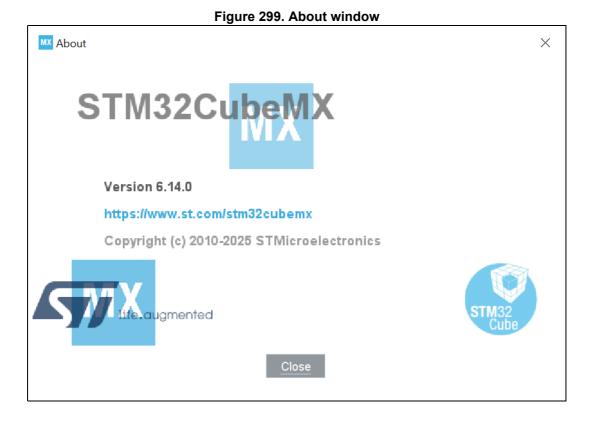
./STM32CubeMX login < email_adress> <password> <remember_me>

"remember me" parameter is either "Y" or "y". If not specified, this command must be run during the next sessions, to allow packages to be downloaded.the default value is no.



4.20 About window

This window displays STM32CubeMX version information. To open it, select **Help > About** from the STM32CubeMX menu bar.





5 STM32CubeMX tools

External Tools 5.1

This panel is accessible from the home page. It provides an overview of the tools relevant for the STM32 product portfolio (see Figure 300):

Figure 300. ST Tools

- to open the tool information note click
- to open the tool webpage on www.st.com click 🚺
- to launch the tool. click 🕥

STM32CubeMX Untitled				- 🗆 X
STM32	File	Window	Help	💿 🖪 🖻 🄰 🔆 ភ
Home 🔪				
Existing Projects		New Project		Manage software installations
Recent Opened Proje		I need to:		Check for STM32CubeMX a
test1.ioc Last modified date : 22/10	MX 0/2020 19:09:37	Start My proje	ect from MCU	
Other Projects	Ē		ect from ST B	INSTALL / REMOVE
		ACCESS TO B	OARD SELECTOR	STM32 Tools
		Start My proje	ect from Exam	STRC © Galarize a. d. d. d. d. d. d. d.
		ACCESS TO EX	AMPLE SELECTOR	
				STMicroelectronics Application Tools
				About STM32 Fexternal Tools

5.2 **Compare Projects**

This new feature is designed to enable the comparison of two projects, based on the same or on different microcontrollers. This tool allows users to efficiently analyze and correlate similarities and differences in IP configurations and project structure between two projects.

5.2.1 User interface of the Compare Projects tool

The user can activate this function from the Tools panel by clicking the **Compare Projects** field (see Figure 301), or from the home page (see Figure 302) by clicking ACCESS TO COMPARE PROJECT under Other services (even before creating any project).



STM32CubeMX Untitled: 5	TM32N645A0HxQ					– a ×
STM32 Cube MX	File	Window	Help			🎯 🖪 💶 X 🗘 🍳 🔆 🖅
	1645A0HxQ > Untitled - T	ools >				GENERATE CODE
Pinout & U Compare Projects	Configuration Project 1 C:\Users\daymarwa\OneDrive - S' Browse	Clock Configuration	s\NVIC_1 Browse	RIF	Project Manager	Tools
PCC	Use current project					ny
CAD						
				Refresh	Export	

Figure 301. Reaching Compare Project from the Tools panel



STM32CubeMX test_levels.in	cr: STM32N645A0HxQ				- σ	×
STM32	File	Window	Help		🎯 🚹 🔼 🔍 🖓 🍭 🛧 🕰	7/
Home STM32N6	645A0HxQ > test_levelx.ioc -	Pinout & Configuration	>		GENERATE CODE	
Existing Projec	ts		New Project		Manage software installations	
Recent Opened	Projects		I need to :		Check for STM32CubeMX and embedded software packages updates	
test_levelx.ioo Last modified dat	e : 17/02/2025 10:38:57	MX	Start My project from	m MCU	CHECK FOR UPDATES	
dma-interrupt Last modified dat	-issue.ioc e : 14/02/2025 12:43:35	MX	ACCESS TO MC		Install or remove embedded software packages NSTALL / REMOVE	
dma_issue.io Last modified dat	C e: 12/02/2025 13:34:52	MX	ACCESS TO BOA			
	no_code_issuee.ioc e : 12/02/2025 10:46:12	MX	Start My project from		and the second s	
H7S_btpath_ Last modified dat	conf.ioc e : 07/02/2025 15:14:05	MX				
Other Projects		Eà.	Other Services	PARE PROJECTS		•
					Get started with a series of step-by-step videos for developers, from beginners to experienced	
Data collection	on information notice				About STM32 External Tools	

The tool is composed of:

- Two main panels, named Project 1 and Project 2.
 - Each field contains a "Browse" button from where we can load the desired .ioc file.
 - The check box "Use Current project" in the first panel is to use the opened project in STM32CubeMX instead of loading a saved one in the local device.
- Three check boxes, named Project 1, Project 2, and "Show differences only":
 - Once the second project is loaded, the Project 1 and Project 2 check boxes are checked systematically, indicating that the output table contains all the parameters of Project 1 and Project 2 (whether they are different or not).



- If the user wants to see only the parameters that are different between the two projects, it can check "Show differences only".
- Refresh button: once clicked, it performs an instantiated comparison.
- Export button: allows users to save or transfer the result of the comparison to an external file in Excel format.

MX STM32CubeMX Untitled:	STM32C011D6Yx						- 0	×
STM32	File	Window		Help		🐵 f	🕒 X 🗘 Q 🔆 🕰	7
Home > STM32C	C011D6Yx 🔰 Untitled - Too	is >				GI	ENERATE CODE	
Pinout & C	Configuration	Clock Configurat		Р	roject Manager		Tools	
Compare Projects	Project 1 Browse Browse butto		Project 2 Browse	Browse butt	ton U	Project 1 Project 2		
PCC)(_				to sho	box checked only if the user want w only the differences between projects	
CAD								
			Ret	fresh button	Export button			
				Refresh	Export			

Figure 303. User interface of the Compare Projects tool

5.2.2 Comparing two projects

- 1. Load the first project from the local device (*Figure 304*).
 - Once the first .ioc is loaded, a popup appears to indicate the need to upload the second .ioc.
- 2. Load the second project (*Figure 305*).
 - After uploading the second .ioc file, the output is displayed in the UI (*Figure 306*).
 - If the user uploads the same .ioc file (having the same path) into the two fields, a popup appears to indicate that it is irrelevant to perform a comparison (*Figure 307*).
 - If the two projects have the same structure and the user checks "Show differences only", only the headers of the respective tables "Target", "Peripherals & Middleware", and "Project Settings" are shown, no data are displayed (*Figure 308*).





teres.							
STM32CubeMX Untitled:	STM32C011D6Yx						- 🗆 🗙
STM32	File	Wi	ndow	Help	Ŕ		🐵 🗗 🗖 X 🗘 🍳 🔆 🖌
Home > STM32C	C011D6Yx 🔰 Untitled - Too	ls 🔪					GENERATE CODE
Pinout & C	Configuration	Clock Con	figuration	Pro	oject Manager		Tools
Compare Projects PCC CAD	Project 1 Browse first step click on Browse button Use current project	Look In File Look In File Core File Name File Name	ASH_init_issue	lect the IOC file	Copen Cancel	Proje	et 1
				Refresh Ex	xport		

Figure 304. Load the first .ioc file

Figure 305. Starting the comparison

STM32CubeMX Untitled:							3
CubeMX	File	Window		Help	🌲 🚇	₩ • • • • • • • • • • • • • • • • • • •	5/
Home > STM320	C011D6Yx > Untitled -	Tools				GENERATE CODE	
Pinout & C	Configuration	Clock Configur	ation	Project Mar	nager	Tools	
	Project 1		Project 2		Pro	lest 1	
	op\cubemx_projects\FLASH_i	nit_issue\FLASH_init_issue.ioc	Desktop\cubemx_	projects\build_error_C0\build_er	rror C0.joc		
	Browse		Browse		Pro	ect 2	
	Use current project		None of Concession, Name		🔲 Sho	w differences only	
			0.11				
PCC	and the second se		Settings				
	CPN	icture		Proiect 1	STM32	Project 2 D092CCT6	
	Flash(KB)		Comparison in p	rogress	256		
CAD	RAM(KB)				30		
	lo Package		LQFP100		45 LQFP48		
	Package Core		ARM Cortex-M33			s ortex-MO+	
				L ₂ -			
				Refresh Export			



		i igure 300. Resul	t of the comparise	
K STM32CubeMX Untitled:	STM32C011D6Yx			- 🗆 X
STM32	File	Window	Help 🌲	🚳 🗗 🕨 X 🗘 🍳 🔆 🖅
Home 🔪 STM32C	C011D6Yx $>$ Untitled -	Tools >		GENERATE CODE
Pinout & C	Configuration	Clock Configuration	Project Manager	Tools
	Project 1 top\cubemx_projects\FLASH_ir Browse Use current project	Browse	nx_projects\build_error_C0\build_error_C0.loc	Project 1 Project 2 Show differences only
PCC	Target Peripherals &		Project 1	Project 2
CAD	CPN Flash(KB) RAM(KB) Io Package Core	STM 22045 2048 786 82 LOFP100 ARM Cortex-M		STM 32C092CCT6 256 30 45 LCPP48 ARM Cortex-M0+
			Refresh Export	

Figure 306. Result of the comparison

Figure 307. Loading the same project

MX STM32CubeMX Untitled	STM32C011D6Yx				×
STM32	File	Window	Help	🔺 🐵 🚰 🖻 X 🗘 🍳 🔆 ភ	ī
Home 🔪 STM320	C011D6Yx 🔪 Untitled -	Tools >		GENERATE CODE	
Pinout & 0	Configuration	Clock Configuration	Project Manage	r Tools	
	Project 1	Project 2		Project 1	
Compare Projects	cop\cubemx_projects\FLASH_ Browse	init_issue\FLASH_init_issue.ioc cop\cubem	nx_projects\FLASH_init_issue\FLASH_init_issu	e.ioc Project 2	
	Use current project			Show differences only	
PCC					
	MX Same IO	OC Loaded !		×	
CAD	🔔 it's	the same IOC as project 1. There's no poi	int in continuing the comparison.Choose anothe	er project and try again	
			OK		
			Refresh Export		
			Reliesh Export		



STM32CubeMX Untitled:	STM32C011D6Yx	•	<u> </u>		- 0	×
STM32	File	Window	Help	.) 🗗 🔼 🗶 🖓 🔍 🛧 🕰	//
	:011D6Yx > Untitled - 1	Tools >			GENERATE CODE	
Pinout & C	Configuration	Clock Configurati		ct Manager	Tools	
Compare Projects	Project 1 op\cubemx_projects\FLASH_in Browse Use current project		Project 2 electronics\Desktop\cubernx_projects\F Browse	LASH_init_issue.ioc	roject 1 roject 2 how differences only	
PCC	Target Peripherals & Struc		ettings Project 1		Project 2	
CAD						
			Refresh Export			

Figure 308. The result of comparing two projects having the same structure

The user has the following options for comparing a current project:

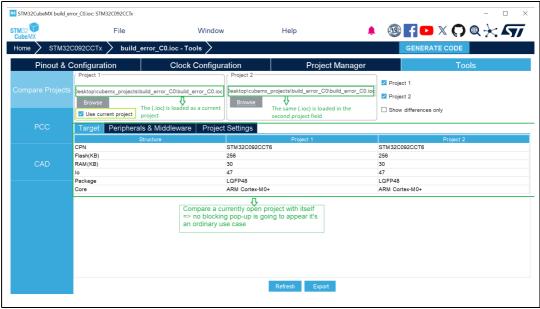
- 1. Load a project from local and start comparing it as the current project (Project 1).
- 2. Compare a newly created project (after configuration), even unsaved, with a saved project, see *Figure 309*.
- 3. Compare a currently open project with itself (the popup blocking the comparison of two similar projects does not appear), see *Figure 310*:
 - a) Load a the first project saved on the local as the current project (current project automatically ticked and the configuration can be modified after loading)
 - b) Load the same project file from the local in the Project 2 field.



гigu	re sua. com	pare the current r	ion saved p	project w	nth anothe	r projeci
STM32CubeMX Untitled: S	STM32C011D6Yx 🖒 An open p	project not saved				– a ×
STM32 CubeMX	File	Window	Help	٠	🕸 🕇 🖻 X	() @ 🔆 ភ
Home > STM32C	C011D6Yx 🔰 Untitled - 1	Tools >			GENERATE	CODE
Pinout & C	Configuration	Clock Configuration	Proje	ct Manager		Tools
Compare Projects PCC	Project 1 Browse V Use current project Target Peripherals &	Browse	bemx_projects\build_error_Cl	0\build_error_C0.ioc	 Project 1 Project 2 Show differences only 	
	Strue	cture	Project 1			roject 2
CAD	CPN Flash(KB) RAM(KB) Io Package	STM32C011 32 6 18 WLCSP12	D6Y3TR		STM32C092CCT6 256 30 47 LQFP48	
	Core	ARM Cortex-	M0+		ARM Cortex-M0+	
			Refresh Export	t		

Figure 309. Compare the current non saved project with another project

Figure 310. Compare a currently open project with itself





5.2.3 The output of the comparison

After starting the comparison, the following elements are added to the UI:

Target table (*Figure 311*): provides a comparison of the parameters used in each project. It clearly shows differences in:

- Part numbers
- Number of IOs
- Package types
- Core configurations
- Available flash memory sizes

The Target table is composed of 3 column:

- Structure: a listing of the MCU parameters.
- Project 1: the values corresponding to the MCU parameters of the first project.
- Project 2: the values corresponding to the MCU parameters of the second project.

The user has the option to:

- Side-by-side comparison: showing data from both projects simultaneously.
- Individual inspection: inspect each file separately by selecting the Project 1 or Project 2 checkbox.
- Focus on differences: exclusively view the differences between 2 projects by checking 'Show differences only'.

The Target table offers a quick and straightforward overview of key differences between the two projects. It represents an invaluable tool for project migrations and initial hardware evaluations.

Peripherals & Middleware table (*Figure 312*): displays the differences and similarities in the configuration of each peripheral or middleware used in the two projects (the sub-parameters and the corresponding values).

The data are presented by lines. It is composed of the following columns:

- Category name
- IP name
- Mode
- Parameters settings
- Project 1
- Project 2

The table includes a highlighting feature, which uses color coding to visually differentiate parameters:

- Specific parameters for project 1 are marked in blue color
- Specific parameters for project 2 are marked in pink color
- The common parameters for the two projects are marked in black color

All peripheral categories are displayed collectively, with the option of sorting in alphabetical order.

Project Settings table (*Figure 313*), helping the users to know the software environment required for each project, and to determine the necessary tools for ensuring project



compatibility and facilitating migration. The table contains information about the firmware package used for each MCU and about the toolchain used for building each project.

It is composed of three columns:

- Settings: •
 - CustomerFirmwarePackage _
 - FirmewarePackage _
 - ProjectStructure _
 - TargetToolchain _
- Project 1
- Project 2

			II Turgot tur		
KTM32CubeMX Untitled	STM32C011D6Yx				- 🗆 ×
STM32 CubeMX	File	Window	Help	🐥 🚳 📑 🕻	> X 🗘 Q 🔆 🖅
Home STM320	C011D6Yx > Untitled - T	ools >		GENE	ERATE CODE
Pinout & 0	Configuration	Clock Configuration	Projec	ct Manager	Tools
Compare Projects	Project 1 Desktop\cubemx_projects\build Browse Use current project	error_C0\build_error_C0.ioc	x_projects\FLASH_init_issue\Fl	✓ Project 1 ✓ LASH_init_issue.ioc ✓ Project 2 □ Show different	ces only
PCC	Target Peripherals &	ture	Project 1		Project 2
CAD	CPN Flash(KB) RAM(KB) Io Package	STM 32C05 256 30 45 LQFP48		STM32U575VIT6 2048 864 172 TFBGA240	
	Core	is no simil If a para	s taken by the parameters a arities between the 2 projec	ARM Cortes-M4 sppear in different color so that the cts regarding those parameters. In the 2 projects it's going to appear oject 2 columns	

Figure 311, Target table

Figure 312. Peripherals & Middleware table

132 V beMX	File	Window	Help	<i>*</i> 🔎		🚳 🚹 🔼 X 🗘 🍭	<u>×4</u>
ime 🔰 STM320	C011D6Yx 🔰 Untitled - Tools					GENERATE CODE	
Pin	out & Configuration	Clo	ck Configuration	Pro	ject Manager	Tools	
	Project 1		Project 2				
	STMicroelectronics\Desktop\cubemx		Cillegerideumanual	apDrive _ STMisseelestropics\Deskte	Project 1		
		_projectsir Exon_int_issuerrExon_			Project 2		
	Browse		Browse		□ Show diffe	renees only	
	Use current project				Show diffe	rences only	
	Target Peripherals & Mide	dleware Project Settings					
	Category Name *	Ip Name	Mode		meters Settings	Project 1 Project	2
	Power and Thermal	PWR	Privilege attributes	PWR Privilege	Disable	Disable	
	Power and Thermal	PWR	Privilege attributes	Voltage detection a	nd monitoring secure prote Disable	not set	
	Power and Thermal	PWR	Privilege attributes	Backup domain sec	ure protection Disable	not set	
	Power and Thermal	PWR	Privilege attributes	Privilege of PWR S	ecure Items Disable	not set	
	Power and Thermal	PWR	Privilege attributes	Pull-up/pull-down se	cure protection Disable	not set	
	Power and Thermal	PWR	Privilege attributes	Wake-Up 2 secure	protection Disable	not set	
	Power and Thermal	PWR	Privilege attributes	Wake-Up 1 secure	protection Disable	not set	
CAD	System Core	RCC			Not Used	Not Used	
	System Core	NVIC			Used	Used	
	System Core	NVIC		System tick timer	15	15	
	System Core	NVIC		Pre-fetch fault, mer	nory access fault 0	0	
	System Core	NVIC		System service cal		0	
	System Core	NVIC		Non maskable inter	rupt 0	0	
	System Core	NVIC		Memory manageme	ent fault 0	0	
	System Core	NVIC		Undefined instruction	n or illegal state 0	0	
	System Core	NVIC		Debug monitor	0	0	
	System Core	NVIC		Pendable request f		0	
	System Core	NVIC		Hard fault interrupt	0	0	
	System Core	CORTEX_M33_NS			Not Used	Not Used	
	System Core	SYS			Used	Used	
	System Core	SYS	Timebase Source		SysTick	SysTick	
	System Core	FLASH			Used	Not Used	
	System Core	FLASH	Enable		Enable	not set	
	System Core	FLASH	Enable	Activate	true	not set	
	System Core	FLASH	Enable	Activate	false	not set	
	System Core	FLASH	Enable	Activate NS BOOT	ADDRESS 0 false	not set	
	System Core	FLASH	Enable	Activate NS BOOT	ADDRESS 1 false	not set	
	System Core	FLASH	Enable	Activate AREA 1	false	not set	

UM1718 Rev 47



STM32CubeMX Untitled: IM32 Cube MX	STM32C011D69x File	Window	Help	<i>*</i>	® ₹ ◘ X () @ ★ 4 77
	C011D6Yx $ ightarrow$ Untitled - Tools $ ightarrow$				GENERATE CODE
Pin compare Projects	Dout & Configuration Project 1 STMicroelectronics\Desktopicubemx_proj Browse		ock Configuration Project 2 C:\Users\daymarwa\Or Browse	Project Manager	Tools
	Use current project Target Peripherals & Middlev	vare Project Setting:		Project 1	Show differences only Project 2
	CustomerFirmwarePackage FirmwarePackage ProjectStructure TargetToolchain		STM32Cube FW_U5 V		STM32Cube FW_H6 V1.3.0
				Refresh Export	

Figure 313. Project Settings table

5.2.4 Saving the comparison result of the two projects

In the user interface of the "Compare Projects" tool, there is an Export button that allows users to save the result of the comparison in an external Excel file.

By clicking the Export button, a window named Save appears to allow the user to choose a name for the resulting file and save it (*Figure 314*).

The available format:

- The result is exported into three sheets in an Excel format (Figure 315).
- Each sheet represents a table (Target, Peripherals & Middleware, Projects Settings).

If the user wants to get only the differences in the exported file, they should click on "Show differences only".



Fig	ure 314	. Choosin	g the Ex	cel format to s	save the c	omparison	result
STM32CubeMX Untitled:	STM32N645A0HxQ						- 🗆 ×
STM32 Cube MX	Fil	e	Window	Help		🐵 🗗 🖻 🕅	
Home > STM32N		Untitled - Tools	\rangle			GENERAT	CODE
Pinout & Conf	iguration	Clock Config		RIF	Project N	Vanager	Tools
	Project 1 a\OneDrive - STM Browse	icroelectronics\Docume		Project 2 p\cubemx_projects\FLASH_init_iss Browse	ue\FLASH_init_issue.ioc	Project 2	
PCC		oject pherals & Middlew Structure	Save In:		- ® @ !	K Show differences only X	Project 2
CAD	CPN Flash(KB) RAM(KB) Io Package		My Data Office So Sound re	cripts		SVIT6	
	Core		File <u>N</u> ame Files of <u>Typ</u> e	compare_result		c-M33	
			. 103 01 1 3 54		Save	Cancel	
				Refresh	xport		

Figure 315. Comparison result in Excel format

~	N		leview View Automate D										-	-		-		-	-	
ste	∑ Cut ☐ Copy → ≪ Format Painter	$ \begin{array}{c c} Aptos Narrow & & 11 & \bullet \\ \hline B & I & \sqcup & \boxplus & \bullet & \bullet \\ \end{array} $	A* ≖ = = ≫ - 8 - = = = = = = ■	Wrap Text	t Center ~	General \$ ~ 9	6 9 58		inditional f matting ~	Format as C Table * Sty	iell In les *	nsert Delete	Format	∑ AutoSur ↓ Fill ~ Clear ~	n Z Sort Filter	& Find & * Select *	Sensitivity	Add-ins	Analyze Data	
		Font										Cells						Add-ins		
10	v i x v	$f_X \sim$																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
		Project 1																		
	CPN	STM32C092CCT6	STM32U575VIT6																	
_	Flash(KB)	256	2048																	
	RAM(KB)	30	786																	
	lo	45	82																	
	Package	LQFP48	LQFP100																	
_	Core	ARM Cortex-M0+	ARM Cortex-M33																	

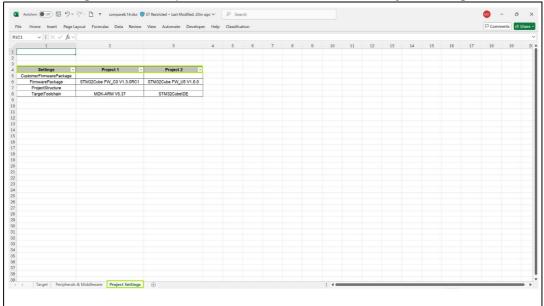


274/555

		ompare6.14.xlsx 💙 ST Restricted • Last Modified: 14m ago 🗸 🔎	Search			-		_
Home Insert	Page Layout Formulas	Data Review View Automate Developer Help Cla	assification			20	omments	ť2
\sim 1 \times	$/ f_x <$							
1	2	3	4	5	6 7	7 8	9	
ategory Name -	lp Name 👻	Mode	Parameters Settings		Project 2			
Middleware	FILEX			Used	Not Used			
Middleware	FILEX	FileX Core		FileX_Core	not set			
Middleware	FILEX	FileX Core	FX_EXFAT_MAX_CACHE_SIZE	512	not set			
Middleware	FILEX	FileX Core	FX_MAX_LAST_NAME_LEN	256	not set			
Middleware	FILEX	FileX Core	FX_DISABLE_ERROR_CHECKING	Disabled	not set			
Middleware	FILEX	FileX Core	FX_SINGLE_THREAD	Enabled	not set			
Middleware	FILEX	FileX Core	FX_NO_TIMER	Enabled	not set			
Middleware	FILEX	FileX Core	FX_DRIVER_USE_64BIT_LBA	Disabled	not set			
Middleware	FILEX	FileX Core	MAX_SECTOR_CACHE_NB_BIT	6	not set			
Middleware	FILEX	FileX Core	FX_DISABLE_FAT_ENTRY_REFRESH	Disabled	not set			
Middleware	FILEX	FileX Core	FX_FAULT_TOLERANT	Disabled	not set			
Middleware	FILEX	FileX Core	FX_FAULT_TOLERANT_DATA	Disabled	not set			
Middleware	FILEX	FileX Core	FX_UPDATE_RATE_IN_SECONDS	10	not set			
Middleware	FILEX	FileX Core	FX_DISABLE_ONE_LINE_FUNCTION	Disabled	not set			
Middleware	FILEX	FileX Core	FX_MEDIA_STATISTICS_DISABLE	Disabled	not set			
Middleware	FILEX	FileX Core	FileX memory pool size	1024	not set			
Middleware	FILEX	FileX Core	FX_MAX_LONG_NAME_LEN	256	not set			
Middleware	FILEX	FileX Core	FX_EXFAT_MAX_CACHE_SIZE_NB_BIT	9	not set			
Middleware	FILEX	FileX Core	FX_ENABLE_EXFAT	Disabled	not set			
Middleware	FILEX	FileX Core	FX_FAULT_TOLERANT_CACHE_SIZE	1024	not set			
Middleware	FILEX	FileX Core	FX_SINGLE_OPEN_LEGACY	Disabled	not set			
Middleware	FILEX	FileX Core	FX_DISABLE_FORCE_MEMORY_OPERATION	Disabled	not set			
Middleware	FILEX	FileX Core	FileX version	6.4.0	not set			
Middleware	FILEX	FileX Core	FX_ENABLE_FAULT_TOLERANT	Disabled	not set			
Middleware	FILEX	FileX Core	FX_DISABLE_FILE_CLOSE	Disabled	not set			
Middleware	FILEX	FileX Core	FX_DISABLE_FAST_OPEN	Disabled	not set			
Middleware	FILEX	FileX Core	FX_DISABLE_BUILD_OPTIONS	Disabled	not set			
Middleware	FILEX	FileX Core	FX_FAT_MAP_SIZE	128	not set			
Middleware	FILEX	FileX Core	FX_FAULT_TOLERANT_CACHE_SIZE_NB_SIZE	10	not set			
Middleware	FILEX	FileX Core	FX_RENAME_PATH_INHERIT	Disabled	not set			
Middleware	FILEX	FileX Core	FX_DISABLE_CACHE	Disabled	not set			
Middleware	FILEX	FileX Core	FX_MAX_FAT_CACHE	16	not set			
Middleware	FILEX	FileX Core	FX_UPDATE_RATE_IN_TICKS	1000	not set			
Middleware	FILEX	FileX Core	FX_DISABLE_CONSECUTIVE_DETECT	Disabled	not set			
Middleware	FILEX	FileX Core	MAX_FAT_CACHE_NB_BIT	4	not set			
Middleware	FILEX	FileX Core	FX_NO_LOCAL_PATH	Enabled	not set			
Middleware	FILEX	FileX Core	FX_FAULT_TOLERANT_BOOT_INDEX	118	not set			
Middleware Target Pe	FILEX ripherals & Middleware	FileX Core	FX MEDIA DISABLE SEARCH CACHE	Disabled	not set		_	-

Figure 316. Comparison result in Excel format - Peripherals and middleware

Figure 317. Comparison result in Excel format - Project settings





5.3 **Power Consumption Calculator view**

For an ever-growing number of embedded systems applications, power consumption is a major concern. To help minimizing it, STM32CubeMX offers the **Power Consumption Calculator** tab (see *Figure 318*), which, given a microcontroller, a battery model and a user-defined power sequence, provides the following results:

Average current consumption

Power consumption values can be taken from the datasheet or interpolated from a user specified bus or core frequency.

- Battery life
- Average DMIPs

DMIPs values are directly taken from the MCU datasheet and are neither interpolated nor extrapolated.

Maximum ambient temperature (T_{AMAX})

According to the chip internal power consumption, the package type, and a maximum junction temperature of 105 °C, the tool computes the maximum ambient temperature to ensure good operating conditions.

Current T_{AMAX} implementation does not account for I/O consumption. For an accurate estimate, I/O consumption must be specified using the Additional Consumption field. The formula for I/O dynamic current consumption is specified in the microcontroller datasheet.

The **Power Consumption Calculator** view allows developers to visualize an estimate of the embedded application consumption and lower it further at each power sequence step:

- make use of low power modes when available
- adjust clock sources and frequencies based on the step requirements
- enable only the peripherals necessary for each phase.

For each step the user can choose V_{BUS} as possible power source instead of the battery, impact battery life. If power consumption measurements are available at different voltage levels, STM32CubeMX also proposes a choice of voltage values (see *Figure 321*).

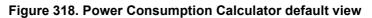
An additional option, the transition checker, is available for STM32L0, STM32L1, STM32L4, STM32L4+, STM32G0, STM32G4, STM32H7 and STM32WB series. When enabled, the transition checker detects invalid transitions within the currently configured sequence. It ensures that only possible transitions are proposed to the user when a new step is added.



5.3.1 Building a power consumption sequence

The default starting view is shown in *Figure 318*.

MX STM32Cub	eMX Untitled: STM32F469NIHx				- 0	×
STM32	File	Window	Help		💿 🖪 🖸 🎽 🛧 🖊	57
	STM32F469NIHx 🔪 Ur	ntitled - Tools 🔪			GENERATE CODE	
Pinc	out & Configuration	Clock Conf	-	Project Manager	Tools	
ST	M32F469NIHx	·····	✓ Power Step	Sequence		
TA	25°C / V _{DD} 3.3V	> New Step 🕆	± 41 ti 🖞	- C C 🛱 🛱 🛱 👫 🚺		
PCC Bat	ttery Selection Select	Step Mo	ode Vdd Rar		Config Peripherals Step Current Duration	on
Info	ormation Notes	>				
Hel	lp	>Display				
		Plot: All Steps	~ ሰ			
MCUs Sele						
STM32F4	Series ST	Lines TM32F469/479	Mc STM32F469NIHx	u Package TFBGA216	Required Peripherals None	



Selecting a V_{DD} value

From this view and when multiple choices are available, the user must select a V_{DD} value.

Selecting a battery model (optional)

Optionally, the user can select a battery model. This can also be done once the power consumption sequence is configured.

The user can select a predefined battery or choose to specify a new battery that best matches its application (see *Figure 319*).



2CubeMX PCC: Battery Database Man	agement				
User Battery					
d User Battery					
		Batteries Table			
	(mAh) Self Discharg			. Max Pulse Curr.	
Alkaline(AA LR6) 2850.0	0.3	1.5	1000.0	0.0	Default
Alkaline AAA L 1250.0	0.3	1.5	400.0	0.0	Default
Alkaline(CLR14) 8350.0	0.3	1.5	3000.0	0.0	Default
Alkaline(DLR20) 20500.0	0.3	1.5	7500.0	0.0	Default
Alkaline(9V 625.0	0.3	9.0	200.0	0.0	Default
Li-MnO2(CF 12 48.0	0.12	3.0	1.0	5.0	Default
Li-MnO2(CR 6 125.0	0.12	3.0	1.5	10.0	Default
Li-MnO2(CR20225.0	0.12	3.0	3.0	15.0	Default
Li-MnO2(CF Add Battery					
Li-MnO2(CF		User Ba	attery		
Li-SOCL2(A Name		E	Battery 29		
		-	0.0		
Li-SOCL2(A Capacity (mAh		(
Li-SOCL2(A Li-SOCL2(C) Capacity (mAh	,	-			
Li-SOCL2(A Li-SOCL2(C Li-SOCL2(D Self Discharge	(%/month)	C	0.0		
LI-SOCL2(A LI-SOCL2(C LI-SOCL2(D Self Discharge LI-SOCL2(D Nominal Voltag	(%/month) e (V)	C			
LI-SOCL2(A Li-SOCL2(C Li-SOCL2(D Li-SOCL2(D Nominal Voltag Ni-Cd(AA11 Max Cont Curre	(%/month) e (V) ent (mA)	c	0.0		
LI-SOCL2(A Li-SOCL2(C Self Discharge Li-SOCL2(D Noincl Voltag Ni-Cd(AA111 Ni-Cd(A1701 Max Pulse Curr	(%/month) e (V) ent (mA)		0.0 0.0		
LI-SOCL2(A LI-SOCL2(C LI-SOCL2(C Self Discharge LI-SOCL2(C) Nominal Voltag NI-Cd(AA11) Max Cont Curr NI-Cd(A170) Max Pulse Curr	(%/month) e (V) ent (mA) rent (mA)		0.0 0.0 0.0		
LI-SOCL2(A LI-SOCL2(C LI-SOCL2(C Self Discharge LI-SOCL2(C) Nominal Voltag NI-Cd(AA11) Max Cont Curr NI-Cd(A170) Max Pulse Curr	(%/month) e (V) ent (mA)		0.0 0.0 0.0		
LI-SOCL2(A Li-SOCL2(C Self Discharge Li-SOCL2(D Nominal Voltag Ni-Cd(A111 Max Cont Curr Ni-Cd(A170 Max Pulse Cur Ni-Cd(C300 * Max Pulse Cur	(%/month) e (V) ent (mA) rent (mA)		0.0 0.0 0.0		OK Canc
LI-SOCL2(A LI-SOCL2(C Self Discharge LI-SOCL2(D Ni-Cd(AA11 Ni-Cd(AA170 Ni-Cd(C300 Ni-Cd(D440) * Max Pulse Cur	(%/month) e (V) ent (mA) rent (mA)		0.0 0.0 0.0	0.0	OK Cano Denault

Figure 319. Battery selection

Power sequence default view

The user can now proceed and build a power sequence.

Managing sequence steps

Steps can be reorganized within a sequence (**Add** new, **Delete** a step, **Duplicate** a step, move **Up** or **Down** in the sequence) using the set of Step buttons (see *Figure 320*).

The user can undo or redo the last configuration actions by clicking the **Undo** button in the Power Consumption Calculator view or the Undo icon from the main toolbar

Figure 320. Step management functions

		-Step) —						eque	nce-		
New Step	Ø	D	Ē.Ĵ	31	5	¢	0	1	8	Ľ	1D	0

Adding a step

There are two ways to add a new step:

- Click Add in the Power Consumption panel. The New Step window opens with empty step settings.
- Or, select a step from the sequence table and click **Duplicate**. A **New Step** window opens duplicating the step settings (see *Figure 321*).



i igui	5 02	1.1 Ower co	nisun	iptio	in sequence. New Ste	
MX New Step						×
Reset Step Settings	Enab	le All IPs Disable	All IPs	Enable	e IPs from Pinout	
	-Pow	er/Memory			Peripherals Selection	Enabled Peripherals
Power Mode		RUN		~	Peripherals	ADC1 ADC2 ADC3 BKPSRAM
Power Scale		Scale1-High		~	- 🗸 ADC1	BusMatrix CAN1 CAN2 CRC
				~	- V ADC2	DAC DCMI DMA1 DMA2 DMA2D
Memory Fetch Type		FLASH/REGON		~	- V ADC3	DSI ETH FMC GPIOA GPIOB
V _{DD}		3.3		~	BusMatrix	GPIOC GPIOD GPIOE GPIOF
Voltage Source		Battery		~	- CAN1	GPIOG GPIOH GPIOI GPIOJ
	(Clocks			CAN2	GPIOK HASH I2C1 I2C2 I2C3
CPU Frequency	15	0 MHz		~		IWDG LTDC PVD/BOR PWR
Interpolation Ranges					∽ DAC	QUADSPI RNG RTC SAI1 SDIO
Interpolation Ranges					- 🗸 OUT1	SPI1 SPI2/I2S2 SPI3/I2S3 SPI4
User Choice (Hz)						SPI5 SPI6 SYS TIM1 TIM10
Clock Configuration	HS	E PLL		~		
Clock Source Frequency	41	MHz		~	- ✓ DCMI	
		nal Settings			- V 1_Stream	
	Орио	hai Settings			- 2_Streams	UART7 UART8 USART1 USART2
Step Duration	1		ms	~		USART3 USART6 USB_OTG_FS
Additional Consumption	0		mA	~	4_Streams	USB_OTG_HS[WWDG]
	—F	Results			5_Streams	
Step Consumption 103	.65 mA				6_Streams	
Without Peripherals 44 r	nA				7_Streams	
Peripherals Part 59.0	35 mA	(A: 5.6 mA - D: 54.0	5 mA)	_	✓ DMA2	
Ta Max (°C) 95.0	08				- V 1_Stream	
				v	Varnings	
						Add Cancel
						Add Calicer

Figure 321. Power consumption sequence: New Step default view

Once a step is configured, resulting current consumption and $\mathsf{T}_{\mathsf{AMAX}}$ values are provided in the window.

Editing a step

To edit a step, double-click it in the sequence table, this opens the Edit Step window.

Moving a step

By default, a new step is added at the end of a sequence. Click the step in the sequence table to select it and use the **Up** and **Down** buttons to move it elsewhere in the sequence.

Deleting a step

Select the step to be deleted and click the **Delete** button.



Using the transition checker

Not all transitions between power modes are possible. The Power Consumption Calculator power menu proposes a transition checker to detect invalid transitions or restrict the sequence configuration to only valid transitions.

Enabling the transition checker option prior to sequence configuration ensures that the user will be able to select only valid transition steps.

Enabling the transition checker option on an already configured sequence will highlight the sequence with a green frame if all transitions are valid (see *Figure 322*), or in fuchsia if at least one transition is invalid (fuchsia frame with description of invalid step highlighted in fuchsia, see *Figure 323*). In the latter case, the user can click the **Show log** button to find out how to solve the transition issue (see *Figure 324*).

Figure 322. Enabling the transition checker option on an already configured sequence - All transitions valid

Step	Mode	Vdd	Range/Scale	Memory	CPU/Bus Freq	Clock Config	Peripherals	Step Current	Duration
1	RUN	3.0	Range3-Low	FLASH	1000000 Hz	MSI		166.9 µA	1 ms
2	RUN	3.0	Range2-Medi	FLASH	8 MHz	HSEBYP		1.3 mA	1 ms
3	RUN	3.0	Range1-High	FLASH	8 MHz	HSEBYP	COMP1 COM	1.55 mA	1 ms
4	SLEEP	3.0	Range1-High	FLASH	8 MHz	HSEBYP		380 µA	1 ms
5	RUN	3.0	Range3-Low	FLASH	4.2 MHz	MSI	COMP1 COM	623.66 µA	1 ms
5	RUN	3.0	Range1-High	FLASH	8 MHz	HSEBYP		1.55 mA	1 ms
7	STOP	3.0	NoRange	n/a	0 Hz	ALL CLOCKS		410 nA	1 ms

Figure 323. Enabling the transition checker option on an already configured sequence - At least one transition invalid

Chan	Mode	Vdd	Range/Scale	Memory	CPU/Bus Frea	Clock Config	Peripherals	Step Current	Duration
Step	Iviode	vaa	Range/Scale	wemory	CP0/Bus Fleq	Clock Conlig	Periprierais	Step Current	Duration
1	RUN	3.0	Range3-Low	FLASH	1000000 Hz	MSI		166.9 µA	1 ms
2	RUN	3.0	Range2-Medi	FLASH	8 MHz	HSEBYP		1.3 mA	1 ms
3	RUN	3.0	Range1-High	FLASH	8 MHz	HSEBYP	COMP1 COM	1.55 mA	1 ms
			Range1-High		8 MHz				
5	RUN	3.0	Range3-Low	FLASH	4.2 MHz	MSI	COMP1 COM	623.66 µA	1 ms
6	RUN	3.0	Range1-High	FLASH	8 MHz	HSEBYP		1.55 mA	1 ms
7	STOP	3.0	NoRange	n/a	0 Hz	ALL CLOCKS		410 nA	1 ms
8	SLEEP	3.0	Range1-High	FLASH	8 MHz	HSEBYP		380 µA	1 ms

Figure 324. Transition checker option - Show log

Log for current sequence	_	\times
======================================		
Check transition between step 6 (RUN, Range1-High) and step 7 (STOP, NoRange) Possible next step(s): RUN [Range1-High, Range2-Medium, Range3-Low] Possible next step(s): LOWPOWER_RUN [Range3-Low] Possible next step(s): SLEEP [Range1-High, Range2-Medium, Range3-Low] Possible next step(s): LOWPOWER_SLEEP [Range3-Low] Possible next step(s): STOP [NoRange] ====================================		
Check transition between step 7 (STOP, NoRange) and step 8 (SLEEP, Range1-High) Possible next step(s): WU_FROM_STOP [NoRange] ====================================		
Close Save in a file		

UM1718 Rev 47



5.3.2 Configuring a step in the power sequence

The step configuration is performed from the **Edit Step** and **New Step** windows. The graphical interface guides the user by forcing a predefined order for setting parameters.

Their naming may differ according to the selected MCU series. For details on each parameter, refer to glossary in *Section 5.3.4* and to *Appendix D*, or to the electrical characteristics section of the datasheet.

The parameters are set automatically by the tool when there is only one possible value (in this case, the parameter cannot be modified and is grayed out). The tool proposes only the configuration choices relevant to the selected MCU.

To configure a new step:

- 1. Click **Add** or **Duplicate** to open the **New step** window or double-click a step from the sequence table to open the **Edit step** window.
- 2. Within the open step window, select in the following order:
 - The Power Mode

Changing the Power Mode resets the whole step configuration.

- The Peripherals

Peripherals can be selected/deselected at any time after the Power Mode is configured.

The Power scale

The power scale corresponds to the power consumption range (STM32L1) or the power scale (STM32F4).

Changing the Power Mode or the Power Consumption Range discards all subsequent configurations.

- The Memory Fetch Type
- The V_{DD} value if multiple choices available
- The voltage source (battery or VBUS)
- A Clock Configuration
 - Changing the Clock Configuration resets the frequency choices further down.
- When multiple choices are available, the CPU Frequency (STM32F4) and the AHB Bus Frequency/CPU Frequency(STM32L1) or, for active modes, a user specified frequency. In this case, the consumption value will be interpolated (see Using interpolation).
- 3. Optionally set
 - A **step duration** (1 ms is the default value)
 - An additional consumption value (expressed in mA) to reflect, for example, external components used by the application (external regulator, external pull-up, LEDs or other displays). This value added to the microcontroller power consumption will impact the step overall power consumption.
- 4. Once the configuration is complete, the **Add** button becomes active. Click it to create the step and add it to the sequence table.



Using interpolation

For steps configured for active modes (Run, Sleep), frequency interpolation is supported by selecting CPU frequency as User Defined and entering a frequency in Hz (see *Figure 325*).

New Step Reset Step Settings Enable All IPs Disable All IPs Enable IPs from Pinout Power/Memory Peripherals Selection Peripherals Selection Peripherals Selection Power Mode RUN Peripherals ADC1 ADC2 ADC Power Scale Scale1-High ✓ ADC1 BusMatrix CAN1 Memory Fetch Type FLASH/REGON ✓ ADC3 ØMA2D DIMA2D DIMA2D DISI ETH	
Power/Memory Peripherals Selection Enabled Peripherals Power Mode RUN Peripherals ADC1 ADC2 BusMatrix CAN1 Power Scale Scale1-High ✓ ADC2 BusMatrix CAN1 DAC DCM DMA1 Memory Fetch Type FLASH/REGON ✓ ADC3 DCM DMA1	C3 BKPSRAM
Power Mode RUN Peripherals Power Scale Scale1-High ADC1 Memory Fetch Type FLASH/REGON ADC3	C3 BKPSRAM
Power Scale Scale1-High ✓ ADC1 BusMatrix CAN1 Memory Fetch Type FLASH/REGON ✓ ✓ ADC2 ØAC DCM ØAC	\leq
Power Scale Scale1-High ✓ Memory Fetch Type FLASH/REGON ✓	CAN2 CRC
Memory Fetch Type FLASH/REGON - ADC3 DAC DCM DMA1	\
V _{DD} 3.3 V - KPSRAM DMA2D DSI ETH	DMA2
	FMC
Voltage Source Battery Voltage Source GPIOA GPIOB GF	
	IOG GPIOH
CPU Frequency User-defined - CRC GPIOI GPIOJ GPIC	OK) (HASH)
Interpolation Ranges 150 MHz - 168 M	INDG LTDC
	DUADSPI
	SDIO SPI1
Clock Source Frequency 4 MHz DCMI SPI2/2S2 SPI3/2S2	3 SPI4 SPI5
Optional Settings	TIM10 TIM11
Step Duration 1 ms V - Streams TIM12 TIM13 TIM	14) (TIM2)
Additional Consumption 0 mA V 3 Streams TIM3 TIM4 TIM5	тімб (тім7)
Results Image: Streams Image: Streams 0 5 5 5	4 UART5
	ISART1
Without Peripherals 44 mA USART2 USART3	USART6
	B_OTG_HS
Ta Max (°C) 95.08	
marininga	
Available use cases: 1 Max: 60 Add	Cancel

Figure 325. Interpolated power consumption



Importing pinout

Figure 326 illustrates the example of the ADC configuration in the **Pinout** view: clicking **Enable IPs from Pinout** in the Power Consumption Calculator view selects the ADC peripheral and GPIO A (*Figure 327*).

The **Enable IPs from Pinout** button allows the user to automatically select the peripherals that have been configured in the **Pinout** view.

Pinout & Configuration		Clock Configurat	ion
		Additional Softwares	
Options Q	ADC Mode and Configuration		
Categories A->Z	Mode		
¢ ✓ ADC	V INO		
O COMP1	V IN1		
COMP2 CRC	V IN2		
DAC	IN3		•
FATFS FREERTOS	IN4		
GPIO	IN5		
12C1 12C2	IN6		

Figure 326. ADC selected in Pinout view



Selecting/deselecting all peripherals

Clicking **Enable All IPs** allows the user to select all peripherals at once.

Clicking **Disable All IPs** removes them as contributors to the consumption.

Figure 327.	Power Consumption Calculator configuration window:
	ADC enabled using import pinout

	Power/Memory		Peripherals Selection	Enabled Peripherals
Power Mode	RUN	~	Peripherals	
ower Scale	Scale2-Medium	~		
Memory Fetch Type	FLASH/REGON	~		
0D	3.3	~		
/oltage Source	Battery		BusMatrix	×
	Clocks			^
PU Frequency	144 MHz	Enable IPs	from Pinout Peripherals Selection	Enabled Peripherals
Interpolation Ranges		Pe	ripherals	ADC1 GPIOA SYS
User Choice (Hz)		L	ADC1	
Clock Configuration	HSE PLL	~ -	ADC3	
Clock Source Frequency	4 MHz	~ -	BKPSRAM	
c	ptional Settings		BusMatrix	
Step Duration 1		ms ~	2_Streams	
dditional Consumption 0		mA ~	3_Streams	
	Results		5_Streams	
Step Consumption 40 m/	ι.		6_Streams	
Vithout Peripherals 40 m/	\		7_Streams	
Peripherals Part 0 nA	(A: 0 nA - D: 0 nA)		☐ 8_Streams	
a Max (°C) 101.1	7		- 1_Stream	
		W	/arnings	

5.3.3 Managing user-defined power sequence and reviewing results

The configuration of a power sequence leads to an update of the Power Consumption Calculator view (see *Figure 328*):

- The sequence table shows all steps and step parameters values. A category column indicates whether the consumption values are taken from the datasheet or are interpolated.
- The sequence chart area shows different views of the power sequence according to a display type (e.g. plot all steps, plot low power versus run modes)
- The results summary provides the total sequence time, the maximum ambient temperature (T_{AMAX}), plus an estimate of the average power consumption, DMIPS, and battery lifetime provided a valid battery configuration has been selected.



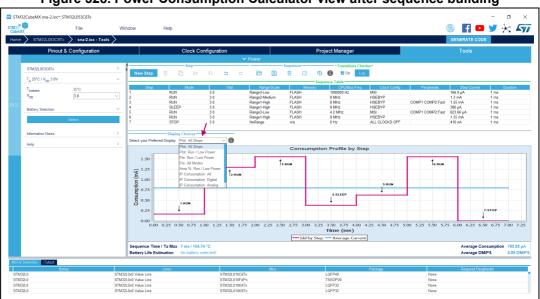
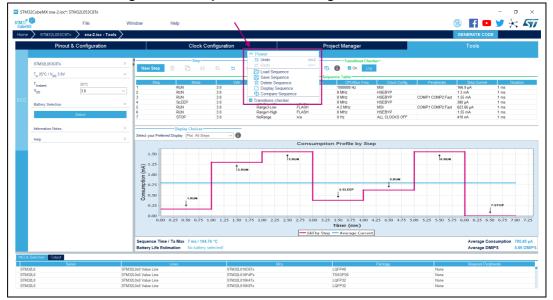


Figure 328. Power Consumption Calculator view after sequence building

Managing the whole sequence (load, save and compare)

From the power menu (see *Figure 329*), the current sequence can be saved, deleted or compared to a previously saved sequence that will be displayed in a dedicated popup window.



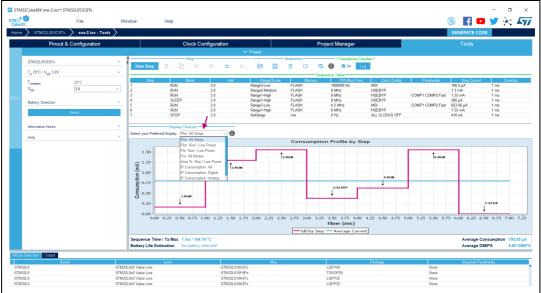




Managing the results charts and display options

In the Display area, select the type of chart to display (e.g. sequence steps, pie charts, consumption per peripherals). You can also click **External Display** to open the charts in dedicated windows (see *Figure 330*).

Right-click on the chart to access the contextual menus: **Properties**, **Copy**, **Save** as png picture file, **Print**, **Zoom** menus, and **Auto Range** to reset to the original view before zoom operations. **Zooming** can also be achieved by mouse selecting from left to right a zone in the chart and **Zoom reset** by clicking the chart and dragging the mouse to the left.





Overview of the Results summary area

This area provides the following information (see Figure 331):

- Total sequence time, as the sum of the sequence steps durations.
- Average consumption, as the sum of each step consumption weighed by the step duration.
- The average DMIPS (Dhrystone million instructions per second) based on Dhrystone benchmark, highlighting the CPU performance for the defined sequence.
- Battery life estimation for the selected battery model, based on the average power consumption and the battery self-discharge.
- T_{AMAX}: highest maximum ambient temperature value found during the sequence.

Figure 331. Description of the Results area

	results Summary					
	Sequence Time / Ta Max	7 ms / 104.42 °C	Average Consumption	1.33 mA		
l	Battery Life Estimation	8 months , 20 days & 9 hours	Average DMIPS	6.52 DMIPS		



5.3.4 Power sequence step parameters glossary

The parameters that characterize power sequence steps are the following (refer to *Appendix D: STM32 microcontrollers power consumption parameters* for more details):

Power modes

To save energy, it is recommended to switch the microcontroller operating mode from running mode, where a maximum power is required, to a low-power mode requiring limited resources.

• V_{CORE} range (STM32L1) or Power scale (STM32F4)

These parameters are set by software to control the power supply range for digital peripherals.

Memory Fetch Type

This field proposes the possible memory locations for application C code execution. It can be either RAM, FLASH or FLASH with ART ON or OFF (only for families that feature a proprietary Adaptive real-time (ART) memory accelerator which increases the program execution speed when executing from flash memory).

The performance achieved thanks to the ART accelerator is equivalent to 0 wait state program execution from flash memory. In terms of power consumption, it is equivalent to program execution from RAM. In addition, STM32CubeMX uses the same selection choice to cover both settings, RAM and flash memory with ART ON.

Clock Configuration

This operation sets the AHB bus frequency or the CPU frequency that will be used for computing the microcontroller power consumption. When there is only one possible choice, the frequencies are automatically configured.

The clock configuration drop-down list allows to configure the application clocks:

- the internal or external oscillator sources: MSI, HSI, LSI, HSE or LSE
- the oscillator frequency
- other determining parameters, among them PLL ON, LSE Bypass, AHB prescaler value, LCD with duty
- Peripherals

The peripheral list shows the peripherals available for the selected power mode. The power consumption is given assuming that peripherals are only clocked (e.g. not in use by a running program). Each peripheral can be enabled or disabled. Peripherals individual power consumptions are displayed in a tooltip. An overall consumption due to peripheral analog and digital parts is provided in the step Results area (see *Figure 332*).



Edit Step		•••••			ed in pinout view $ imes$
	Enable All IPs Disable		able IPs from Pinout	LIIADIG	
	Power/Memory	AITIPS	Peripherals Sel	ection	Enabled Peripherals
Power Mode	RUN	~	Peripherals		ADC1 GPIOA SYS ADC2
			- 🗹 ADC1		
Power Scale	Scale1-High	~	- 🗹 ADC2		
Memory Fetch Type	FLASH/REGON	~	- ADC3		CAN1 CAN2 CRC DAC DCMI
V _{DD}	3.3	\sim	─ ✓ BKPSRAM ─ ✓ BusMatrix		DMA1 DMA2 DMA2D DSI
Voltage Source	Battery	\sim	- CAN1		ETH FMC GPIOB GPIOC
	Clocks		- CAN2		GPIOD GPIOE GPIOF GPIOG
CPU Frequency	User-defined	\sim	- 🗹 CRC		GPIOH GPIOI GPIOJ GPIOK
Interpolation Ranges	150 MHz 168 MHz	~	>- DAC		(HASH) [2C1] [2C2] [2C3] [WDG]
User Choice (Hz)	16000000		– ✓ DCMI > DMA1		LTDC PVD/BOR PWR
					QUADSPI RNG RTC SAI1
Clock Configuration	HSE PLL	~	- V DMA2D		[SDIO] [SPI1] [SPI2/I2S2] [SPI3/I2S3
Clock Source Frequency	4 MHz	\sim	— 🗹 DSI		SPI4) (SPI5) (SPI6) (TIM1) (TIM10)
	Optional Settings		. — 🗹 ETH		
Step Duration	1	ms 🗸	- ✓ FMC		
Additional Consumption	D	mA 🗸			
	Results		- GPIOC		
Step Consumption 110.0	04 mA		— 🗹 GPIOD		UART5 UART7 UART8
Without Peripherals 46.7			- 🗹 GPIOE		USART1 USART2 USART3
Peripherals Part 63.26 mA (A: 5.6 mA - D: 57.66 mA)				USART6 USB_OTG_FS	
Ta Max (°C) 94.4	7		- GPIOH		USB_OTG_HS WWDG
		v	Varnings		
Available use cases: 0 Ma	ax: 60				OK Cancel

Figure 332. Overall peripheral consumption

The user can select the peripherals relevant for the application:

- none (Disable All)
- some (using peripheral dedicated checkbox)
- all (Activate All)
- or all from the previously defined pinout configuration (Import Pinout).

Only the selected and enabled peripherals are taken into account when computing the power consumption.

• Step duration

The user can change the default step duration value. When building a sequence, the user can either create steps according to the application actual power sequence or define them as a percentage spent in each mode. For example, if an application



spends 30% in Run mode, 20% in Sleep and 50% in Stop, the user must configure a 3-step sequence consisting in 30 ms in Run, 20 ms in Sleep and 50 ms in Stop.

Additional Consumption
 This field allows entering an additional consumption resulting from specific user configuration (e.g. MCU providing power supply to other connected devices).

5.3.5 Battery glossary

• Capacity (mAh)

Amount of energy that can be delivered in a single battery discharge.

• Self-discharge (% / month)

This percentage, over a specified period, represents the loss of battery capacity when the battery is not used (open-circuit conditions), as a result of internal leakage.

- Nominal voltage (V)
 Voltage supplied by a fully charged battery.
- Max. continuous current (mA)

This current corresponds to the maximum current that can be delivered during the battery lifetime period without damaging the battery.

• Max. pulse current (mA)

This is the maximum pulse current that can be delivered exceptionally, for instance when the application is switched on during the starting phase.

5.3.6 SMPS feature

Some microcontrollers (e.g. STM32L496xxxxP) allow the user to connect an external switched mode power supply (SMPS) to further reduce power consumption.

For such microcontrollers, the Power Consumption Calculator tool offers the following features:

- Selection of SMPS for the current project
 From the left panel, check the Use SMPS box to use SMPS (see *Figure 333*). By default, ST SMPS model is used.
- Selection of another SMPS model by clicking the Change button
- This opens the SMPS database management window in which the user can add a new SMPS model (see *Figure 334*). The user can then select a different SMPS model for the current sequence (see *Figure 335*, *Figure 336* and *Figure 337*)
- Check for invalid SMPS transitions in the current sequence by enabling the SMPS checker

To do this, select the checkbox to enable the checker and click the **Help** button to open the reference state diagram (see *Figure 338*).

Configuration of SMPS mode for each step (see *Figure 339*)
 If the SMPS checker is enabled, only the SMPS modes valid for the current step are proposed.



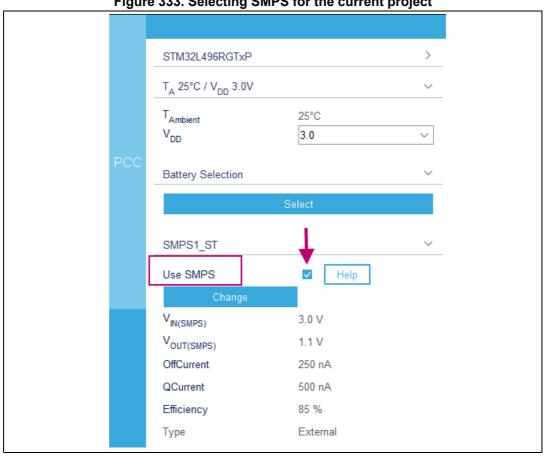


Figure 333. Selecting SMPS for the current project



STM32CubeMX PCC: SM	IPS Database Managem	ent				×
User SMP	s					
Add User SMPS	Edit					
		SMPS T				
Used Name ⊘ SMPS1_ST 3.0	VIN(SMPS) OffCurrent 250.0	QCurrent 500.0	V _{OUT(SMPS)}	Efficiency 85	Type External	Database ♀ Default
SIVIFS1_51 3.0	250.0	500.0	1.1	05	LXtemai	Delauit
	Edit SMPS		×			
	User	SMP S				
	Name	SMPS2_User				
	Vin (V)	2.5				
	OffCurrent (nA)	10				
	Quiescent Current (nA)	10				
	V _{OUT(SMPS)}	1.2				
	Efficiency (%)	85				
	Туре	External				
		OK (Cancel			
					OK	Cancel

Figure 334. SMPS database - Adding new SMPS models

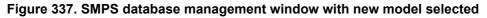
Figure 335. SMPS database - Selecting a different SMPS model

	User	SMP S						
	dd User SMPS	Ū B	dit					
				SMP S 1	able			
Jsed	Name	V _{IN(SMPS)}	OffCurrent	QCurrent	V _{OUT(SMPS)}	Efficiency	Туре	Database 🌲
	SMPS2_User	2.5	10.0	10.0	1.2	85	External	User
		3.0	250.0	500.0	1.1	85	External	Default



SMPS2_User	~
Use SMPS	Help
Change	
V _{IN(SMPS)}	2.5 V
V _{OUT(SMPS)}	1.2 V
OffCurrent	10 nA
QCurrent	10 nA
Efficiency	85 %
Туре	External

Figure 336. Current project configuration updated with new SMPS model



	User	SMPS						
A	dd User SMPS	Ū E	Edit					
				SMPS 1	[able			
sec	Name	V _{IN(SMPS)}	OffCurrent	QCurrent		Efficiency	Туре	Database 🌲
	CMDC0 Lists	2.5	10.0	10.0	1.2	85	External	User
9	SMPS2 User							



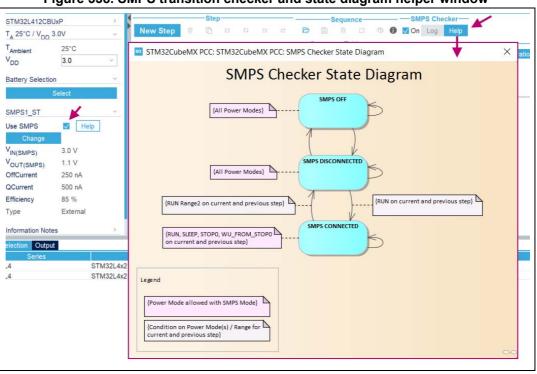


Figure 338. SMPS transition checker and state diagram helper window



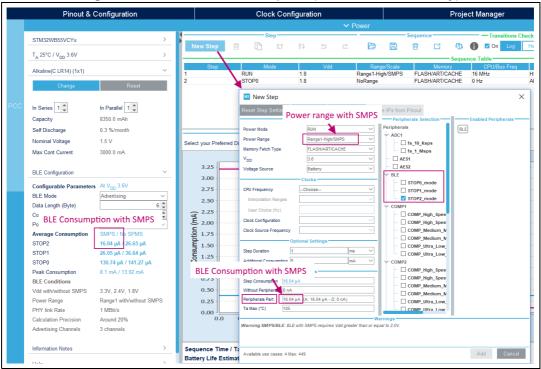
New Step			×
Reset Step Settings Er	nable All IPs Disable All IPs	Enable IPs from Pinout	
	ver/Memory	Peripherals Selection	Enabled Peripherals
Power Mode	RUN	✓ Peripherals	Endorar oriprorato
Power Range	Range1-High	✓ ↓ ADC1	
Memory Fetch Type	FLASH	└── □ fs_10_ksps	
V _{DD}	3.0	─ □ fs_1_Msps	
Voltage Source	Battery	└── □ fs_5_Msps	
_	SMPS	→ ADC2	
SMPS Mode	CONNECTED	fs_10_ksps	
QCurrent	10 nA	fs_1_Msps fs_5_Msps	
V _{OUT(SMPS)}	1.2 V	✓ ADC3	
Efficiency	85 %	_	
	Clocks	fs_1_Msps	
CPU Frequency	Choose	→ ☐ fs_5_Msps	
Interpolation Ranges		AHB_APB1_Bridge	
User Choice (Hz)		AHB_APB2_Bridge	
Clock Configuration			
Clock Source Frequency			
	onal Settings		
Step Duration 1	ms		
Additional Consumption 0	mA	UT1+OUT2-Buffei	
	Results	- OUT1+OUT2-Buffe	
Step Consumption 10 nA	noound	- OUT1-Buffer_OFF-	
Without Peripherals 10 nA		OUT1-Buffer_ON-N	
Peripherals Part 0 nA (A	A: 0 nA - D: 0 nA)	OUT1-Buffer ON-V	
		Warnings	
Available use cases: 18 Max	c: 856		Add Cancel

Figure 339. Configuring the SMPS mode for each step



5.3.7 Bluetooth Low-Energy[®]/ZigBee[®] support (STM32WB series only)

The Power Consumption tool allows the user to take into account the consumption related to the RF peripheral and corresponding Bluetooth Low-Energy functional mode, combined with the usage of the SMPS feature.





The Bluetooth Low-Energy mode can be selected from the left panel and configured to reflect the application relevant settings. For each new step enabling BLE, the peripheral consumption part is updated accordingly (see *Figure 341*). A similar approach is used for ZigBee (see *Figure 342*).



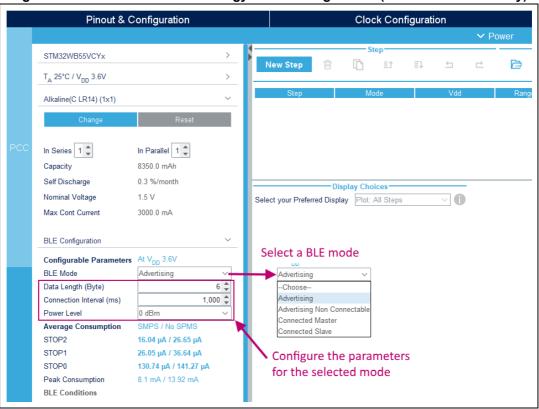
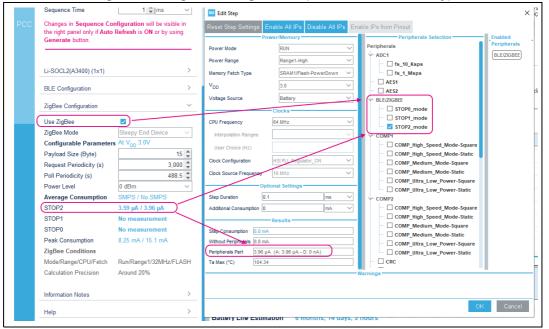


Figure 341. RF Bluetooth Low-Energy mode configuration (STM32WB series only)





UM1718 Rev 47

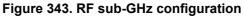


UM1718

5.3.8 Sub-GHz support (STM32WL series only)

Sub-GHz usage can be enabled from the left panel and configured to reflect the application relevant settings. For each new step enabling ZigBee, the peripheral consumption part is updated accordingly (see *Figure 343*).

			-	0.0.2	0.0				· · · ·	
	STOP2 Consumption 88	5 nA	Edit Step							×
PCC	Low Power Mode	STOP2 V	Reset Step Setti	ngs Enable All I	Ps Disable All IPs	Enabl	e IPs from Pir	out		
	RUN Step 0.1 ms / STO	P2 Step 0.9 ms		Power/Memo	ry			eripherals Sel	ection	Enabled
	0 20 40	60 80 100	Power Mode	RUN		\checkmark				Peripherals SUBGHZ
	Sequence Time	1 🗘 ms 🗸	Power Range		1-Medium SMPS-ON	\sim				(SUBGHZ)
		·	Memory Fetch Type	eChoo	Se_	~	— 🗌 РКА			
	Auto Refresh ON automatically update sequence table, display and results in the right panel.		V _{DD}			~		0		1
1			Voltage Source	Battery	r	<u> </u>		0		
				Clocks			- SPI2/12			
	Li-SOCL2(AAA700) (1x1) >	CPU Frequency			\leq	- SRAM			
	SUBGHZ Configuration	~	Interpolation Ran			<u> </u>	- SUBGHZ	, L		
	Configurable Parame	User Choice (Hz)		-	— 🗹 Rx				
	Frequency Band	High V	Clock Configuration							
	SMPS	On V	Clock Source Frequ	Jency		~	- C TIM1	2351		
	•Tx Parameters			Optional Settin	igs -	-	- 🗌 TIM2			
	Power Amplifier	Low Power V	Step Duration	0.1	ms	\sim	- TIM16			
	Output Power	10 dBm 🗸 🗸	Additional Consum	otion 0	mA	~	- 🗆 TS			
	Optimal Settings			Results						
	Tx Current	17.5 mA	Step Consumption Without Peripherals	5.1 mA		- 1	VREFBUF	2		
	Modulation	FSK	Peripherals Part	5.1 mA (A: 5.1 mA	- D: 0 nA)	- 1		ad_0_uA		
	Rx Boosted		Ta Max (°C)	125				ad_4_mA		
	Rx Current	5.1 mA	I			War	nings			
	Information Notes	>								
	Help	>							C	K Cancel
			ļ							



5.3.9 Example feature (STM32MPUs and STM32H7 dual-core only)

Under the section Sequence Examples, the PCC tool allows to access examples: each of them comes with an explanatory slide set and a ready-made sequence to load in PCC (see *Figure 344*).



STM32	CubeMX Untitled: STM32H745IGKx					- 🗆
2 ² T	File	Window	Help		(15) 🖪 🖸 🏏 🔀
ome	angle STM32H745IGKx $ angle$ Untitle	ed - Tools >			GENER	RATE CODE
Р	inout & Configuration	Clock Co	nfiguration	Project Manag	ger	Tools
			~ Power			
	6111152111461610	> Sequence H7 Dual Cor	ExamplesNew S	Step tep 🕆 🗋 🗊 🗈		juence 🗄 🖾 🕸 🔀
	T _{Junction} (°C) 25 V _{DD} 3.0	Step Mod	STM32CubeMX PCC: H7 Du	al Core Examples H7 Dual Core	Examples	×
	Battery Selection Select	× Lo	ead Example 1 ead Example 2 ead Example 3	Example 1 Presentat Example 2 Presentat Example 3 Presentat	ion I2C transmiss ion I2C transmiss ion I2C transmiss I2C transmiss	
	Information Notes	> Plot: All Steps		Benefits of H7 Dual C		ation
	Help	<u>></u>		Low-power app	lication example 3	-
				example completes the example y phase.	2 by adding a new data	
			STM3	urpose is to highlight the smart 2H7 Series (STM32H7x5 or ST domains.		
			L77	104 10	870	relients
				Slides S	how	
			< Previous	Slide 1	of 9	Next >

Clicking "Load Example N" loads the sequence corresponding to example N (see Figure 345).

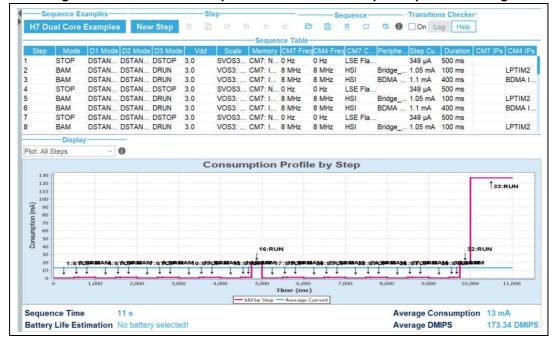


Figure 345. Power Consumption Calculator – Example sequence loading

Clicking "Example N Presentation" displays the explanations for that example.



The example can be changed anytime: the new sequence can be added to the current sequence, or replace it (see *Figure 346*).

Figure 346. Power Consumption Calculator – Example sequence new selection

Sec	quence Examples-				-Step-						Seque	ence-			-Transi	tions (Checker
H7 Du	ial Core Exampl	les Nev	w Step	8	D I	E1	5	₽	6		Ø	Ľ	213	0	On	Log	Help
Step	Mode D1	STM32Cube	MX PCC: H	17 Dual	Core Exam	ples	-									×	ration
Step	STOP DST					-H7 D	ual Co	re Exan	nnles								States in the second second second
1		ad Example	4		Eva	mple 1			inpiece		12C tr	ansmis	sion +	dat	a acquis	sition	ms
2		ad Example				mple 2									a acquis		ms
3	STOP DST LO	ad Example	3			mple 3						ansmis		GGG	a aoqui	in the second se	
4		do axompro	-			efits of						Preser					ms
Dial Doi								mole 2-									ms
Warning: load Example 2 with existing steps! X							EAdi	A LANGE									ms
Before loadin	g Example 2, keep	or remove	existing ste	eps.									-				ms
					Low-	DOME	er ap	oplica	itior	1 ex	am	ple 2	2 💻	1			ms
Keep Ste	ps Remove Step	ps Can	cel														
Plot: All S	Steps																
-			• 1	his exa	ample com	pletes th	e exam	ple 1 by a	adding	a new	data						
			a	cquisit	ion phase.												
130			. 7	henu	pose is to	inhlight	the em	art power	mono		t of						
110					H7 Series (ree					
100					fomains.						ang ar						
(PH 90																	
S 80																	
01 70 for 60																	
5 00			L7/														
S 50 -			-11				1044	1. C									3
Consumption (mA) 09 00 08 06							Clide										- 1
SLO 50																	
30 - 20 -	1:S12093-88		-					s Show-		1							90.94
30 - 20 - 10 -	1:SP20177500	<	Previous					1 of 9					Ne	xt >			9 0.94
30 - 20 - 10 - 0 -	1:S P.3934	2,000	Previous 3,0	00	4,000			1 of 9 6	,000		7,000		Ne 8,00		9,0	00	ļ
30 - 20 - 10 - 0 -			_	00	4,000		Slide	1 of 9	ms)		7,000		1011		9,0	100	3 0.3

Note: The examples are provided for a given part number and may require adjustments when used for a different part number. Also, after loading, it is recommended to edit each step and check settings.

5.4 DDR Suite (STM32MPUs only)

DDR SDRAMs are complex high speed devices that need careful PCB design.

The STM32MP15 devices support the following DDR types:

- LPDDR2
- LPDDR3
- DDR3 / DDR3L

They are specified by the JEDEC standard (standardization of interfaces, commands, timings, packages and ballout).

STM32CubeMX has been extended to provide an exhaustive tool suite for the DDR subsystem. It proposes the following key features.

- Configuration of DDR controller and PHY registers is managed automatically based on reduced set of editable parameters.
- DDR testing is offered based on a rich list. Tests go from basic to stress. User can also develop its own tests.

DDR configuration is accessible like the other peripherals in the **Pinout & Configuration** view: clicking the DDR from the component panel opens the mode and configuration panels.



DDR Test suite testing and tuning features are available from the Tools view.

The DDR suite relies on two important concepts:

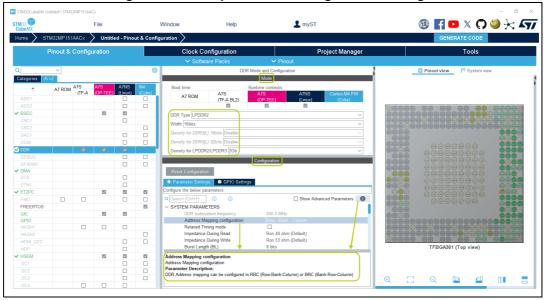
- the DDR timings as key inputs for the configuration of the DDR Controller and PHY
- the tuning of DDR signals to compensate board design imperfections.

5.4.1 DDR configuration

STM32CubeMX allows to set DDR system parameters and JEDEC core timings. The timing parameters are available in the DDR datasheet.

DDR type, width, and density

The DDR type, width, and density parameters must be set to proceed with the DDR configuration. This can be done in the Mode panel after selecting the DDR in the **Pinout & Configuration** view. See *Figure 347* for an example of LPDDR2 settings.





Another example: for a configuration with two "DDR3 16 bits 2 Gb" chips, settings are "DDR3/DDR3L", "32 bits" and 4 Gb".

Note: Contexts for DDR IP cannot be changed, DDR is tied to "Cortex-A7 nonsecure" identified as "Cortex-A7 NS" in the tool.

DDR configuration

Clicking on a parameter will show additional details in the DDR configuration footer.

- The DDR frequency is taken from the 'Clock configuration' tab, it cannot be changed in the DDR configuration.
- The 'Relaxed Timing' mode is used during bring-up phase for trying relaxed key DDR timings value (one t_{CK} added to t_{RC}, t_{RCD} and t_{RP} timings)
- Other parameters must be retrieved from the user DDR datasheet.
- Some parameters are read-only: they are for information only and depend on the DDR type.



Clicking "generate code" automatically computes the DDR node of the device tree (DDR Controller and DDR PHY registers values) based on these parameters.

DDR3 configuration

For DDR3, the configuration is made easier with the selection of a **Speed Bin Grade** combination, instead of manually editing timing parameters.

		540. DDR5 COM	.g	
	DDI	R Mode and Configu	iration	
		Mode		
Boot time:		Runtime contexts:		
	A7S	A7S	A7NS	Cortex-M4 FW
A7 ROM	(TF-A BL2)	(OP-TEE)	(Linux)	(Cube)
	~	\checkmark	\checkmark	
DDR Type DDR3 /	DDR3L			\sim
Width 16bits				~
Density for DDR3(I.) 16bits 1Gb			~
		Configuration		
	-	Conliguration		
Reset Configuration	1			
⊘ Parameter Settin	s 🛛 📀 Registers	🥝 GPIO Settings		
Configure the below pa	ameters :			
Q Search (Ctrl+F)	0 0		Show Adv	anced Parameters 🛛 🚯
✓ SYSTEM PARAM	TERS			
DDR subsy	tem frequency	400.0 MH	lz	
Speed Bin C	Grade	DDR3-10	66G / 8-8-8	
Impedance I	During Read	Ron 40 o	hm / ODT = 80 ohr	m (Default)
Impedance I	During Write	Ron 53 o	hm / ODT = 60 ohr	m (Default)
Address Ma	pping configuration	Row - Ba	nk - Column	
Relaxed Tim	ing mode			
Temperature	e case over 85°C sup	port 🗌		
Burst Lengt	n (BL)	8		
Speed Bin Grade Speed Bin Grade Parameter Descripti	on:			
JEDEC Standard for D				
Set value of each timir Tick 'Show Advanced	-		l Bin Grade' select	ion.

Figure 348. DDR3 configuration

The Speed Bin Grade combination must match the selected DDR. If the exact combination is not in the pick-list, select "1066E / 6-6-6" for faster DDR Speed Bin Grade, or "1066G / 8-8-8" for a relaxed configuration.

Timing edition is optional, and reserved for advanced users: select Show Advanced parameters to display the list.



5.4.2 Connection to the target and DDR register loading

To manage DDR tests and tuning, STM32CubeMX must establish a connection with the target and more specifically with **U-Boot SPL** using the **DDR interactive protocol**:

- the DDR interactive protocol is only available in the **Basic boot scheme U-Boot SPL** binary and supported over the UART4 peripheral instance
- when U-Boot SPL detects a connection to STM32CubeMX on UART4, it stops its initialization process and accepts commands from STM32CubeMX.

There are two connection options:

- 1. the U-Boot SPL binary is available in flash memory
- 2. the U-Boot SPL needs to be loaded in SYSRAM because the DDR has not yet been tested nor tuned (and, consequently, is not fully functional yet).

Prerequisites

- Installation of ST-Link USB driver to perform firmware upgrades: for Windows, latest version of STSW-LINK009, for Linux, use STSW-LINK007. Both can be downloaded from *www.st.com*.
- Installation of STM32CubeProgrammer (for SYSRAM loading only): installer can be downloaded from *www.st.com*.

Connection to the target

The COM port must be selected to connect to the target, as indicated in Figure 349.

32 V DeMX	File	Window	Help	L myST	3	F 🖻 🛛 🖓 🍎 🔆
	132MP157CACx 🔰 Untitled - Tools	>				GENERATE CODE
Pi	nout & Configuration	Clock Co	nfiguration	Project Manager		Tools
	DDR Interactive Connection Common Connect Status Connect Common C	USB UART		Targ	et Information 🜒	
	Not Connected Select	file stm32mp157c-ev1-basic	.stm32mp157c-ev1-basic.			
	Target DDR Tests	Connection Set	ings	- 6	I X	
		STM 32CubePro	grammer CubeProgrammer has been manua	lly selected		
		Executable File Browse		M32Cube\STM32CubeProgrammer/bin\STM32_Programmer_C	LI.exe	
		Version STM32CubePro	rammer version 2.0.0			
			0	k Cancel		

Figure 349. DDR Suite - Connection to target

If U-Boot SPL loading in SysRAM is required, it can be performed through UART or USB using the STM32CubeProgrammer tool. If not automatically detected by STM32CubeMX, the STM32CubeProgrammer tool location must be specified in the Connection settings window: click to open it. U-Boot SPL file must be manually selected in the build image folder.



Once up, the connection gives the various services and target information (see Figure 350).

🚥 STM32CubeMX Unti	tled*: STM32MP157CACx						– ø ×
STM32 CubeMX	File	Window	Help	L myST	<u></u>	📑 🕒 🗶 💭) 🍑 🔆 🖅
	132MP157CACx 🔰 Untitled - Too	ls >				GENERATE CODE	
Pi	inout & Configuration	Clock C	Configuration	Project Manager		Tools	
PCC	COM3 V Disconnect	USB UAR	Loading 🖲 😳	Config name: DDR3-1066/888 bin G 2x DDR Size: 8 GBits DDR Frequency: 533.0MHz	Target Information (1)		
CAD	⊘ Connected Se	elect file stm32mp157c-ev1-ba	asic.stm32mp157c-ev1-basic				
DDR Test Suite	Use the 'Load Registers' comm			I-Boot SPL will then be re-started and DI ectly on the configuration you have edite			
			Load Registers	I			

Output/Log messages

STM32CubeMX outputs DDR suite related activity logs (see *Figure 351*) and interactive protocol communication logs (see *Figure 352*). They are displayed by enabling outputs from the Window menu.

Figure 351. DDR activity logs



Figure 352. DDR interactive logs

Host	>	Target	info
Target	>	Host	step = 0 : DDR_RESET
Target	>	Host	name = DDR3-1066/888 bin G 2x4Gb 533MHz v1.45
Target	>	Host	size = 0x40000000
Target	>	Host	speed = 533000 kHz
Host	>	Target	step 3
Target	>	Host	step to 3:DDR_READY
Target	>	Host	1:DDR_CTRL_INIT_DONE
Target	>	Host	2:DDR PHY_INIT_DONE
Target	>	Host	3:DDR_READY
Host	>	Target	print mstr
Target	>	Host	mstr= 0x00040401
Host	>	Target	tuning help
Target	>	Host	tuning:5
Target	>	Host	0:Read DQS gating:software read DQS Gating:
Target	>	Host	l:Bit de-skew::
Target	>	Host	2:Eye Training:or DQS training:
Target	>	Host	3:Display registers::



DDR register loading (optional)

Once connected in DDR interactive mode, the current DDR configuration can be loaded in SYSRAM.

IIX STM32CubeM	X Untitled*: STM32MP157CACx						– 0 ×
STM32 CubeMX	File	Window	Help	L myST	(19)	f 🖻 🛛 🖓	🍏 🗙 🏹
Home >	STM32MP157CACx $ ightarrow$ Untitled - Tools	\rangle				GENERATE CODE	
	Pinout & Configuration	Clock	Configuration	Project Manager		Tools	
PCC CAD DDR Test S	DDR Interactive Connection Connection Connected DDR util Connected Connecte	USB UAF	M Loading	Config name: DDR3-1069/888 bin G 2x40 v1.45 DDR Size: 8 GBits DDR Frequency: 400.0MHz U-Boot SPL will then be re-started and DDR freetly on the configuration you have edited in	CTRL and PHY	Tools	
		•	Load Registers Progress Loading DDR Registers 27 out of 96 registers Cancel	×			

Figure 353. DDR register loading

This step is optional if the used U-Boot SPL already contains the required configuration. It triggers the DDR Controller and PHY initialization with those registers, and allows the user to quickly test a configuration without generating the device tree and dedicated U-Boot SPL binary file.

5.4.3 DDR testing

Prerequisites

To proceed with DDR testing:

- The DDR suite must be in connected state
- The DDR configuration must be available in memory, either with the U-Boot SPL (with DDR register file in Device Tree) or in the DDR registers (see *Section 5.4.2*).

DDR test list

DDR tests are part of the U-Boot SPL (see Figure 354).



I32	File	Window	Help	L myST	🚳 🚹 🗖 🗶 🖓 🤙		
ime 🔪 ST	M32MP157CACx 🔰 Untitle	ed - Tools 🔪			GENERATE CODE		
1	Pinout & Configuration	Clock Co	nfiguration	Project Manager	Tools		
	DDR Interactive	SYSRAM Lo	ading 🔀 🔕		Target Information 🚯		
	Target DDR Tests						
	ld(s) Test type(s)	Test name(s)	Execution				
	0 All	All		Run	test 🕜		
	1 Basic	Simple DataBus					
	2 Basic	DataBusWalking0		Parameter(s)	Value(s)		
	3 Basic	DataBusWalking1	Address	Parameter(s)	0x C0000000		
	4 Basic	AddressBus MemDevice	Loop		1		
	5 Intensive		Loop		1		
	6 Intensive	SimultaneousSwitchingOutput Noise	Verdict				
	7 Intensive 8 Intensive	Noise NoiseBurst	verdict				
		Random	_				
	10 Intensive with Stress	FrequencySelectivePattern		ltem(s)	Info		
	Conditions	r requercy oblectiver autom	Address		Run the Test		
	11 Intensive	BlockSequential	Loop(s)		Run the Test		
	12 Intensive	Checkerboard	Result		Run the test to have a verdict		
	13 Intensive	BitSpread	Result details		None		
	Details						
	ltem(s)			Info			
	Name	DataBusWalking0		mo			
	Purpose	Verifies each data bus signal can be drive	en low				
	Test Sequence	Writes patterns with 'moving' 0 on a 32 da	An use use supplication of univername. terms with "moving" 0 on a 32 data bus width. Example: write '11111111111111111111111111110' at given address, read back given address and check the pattern is OK. Write then 1111111111111111111111110' at given address, read back given address and check the pattern is OK, and so on.				
	Param1	[Address] The memory address where the test is executed. All writes and reads are performed on this address. The given address should be located in the DDR memory region [DDR base address DDR base address + DDR size].					
	Param2		dict. Same test is repeated [Loc	p] times. Verdict OK if all tests are OK, KO otherwi	Sē.		
	Restriction Limitations	If the data bus splits as it makes its way to more than one memory chip, you will need to perform the data bus test at multiple addresses, one within each chip.					
	Interest		Very basic best to be executed first to make sure the data bus is clean from connection issues.				
	Failure Type	Catastrophic failure					

Figure 354. DDR test list from U-Boot SPL

New tests can be added by modifying the U-boot SPL.

Most of the tests come with parameters to be set prior to execution, such as:

- Address: the memory address where the test is executed. All writes and reads are performed on this address. The given address has to be located in the DDR memory region [DDR base address, DDR base address + DDR size].
- On STM32MP15, DDR base address is 0xC0000000 (as an example, DDR size for 4 Gbits is 0x20000000).
- Loop: number of test iterations before verdict. Same test is repeated [Loop] times. Verdict OK if all tests are OK, KO otherwise.
- Size: the byte size of the region to test. It must be a multiple of 4 (read/writes are performed on 32-bit unsigned integers), with minimal value equal to 4, and up to DDR size.
- Pattern: the 32-bit pattern to be used for read / write operations.

The DDR Suite embeds an auto-correction feature preventing users to specify wrong values.

All tests are performed with Data cache disabled and Instruction cache enabled.

DDR test results

The test verdict is reported by the U-Boot SPL: the parameters used for the tests are recalled, along with Pass/Fail status and results details (see *Figure 355*). The test history is available in the output and Logs panels (see *Figure 356*).



```
Figure 355. DDR test suite results
```

Execution	
	Run test
Parameter(s)	Value(s)
Address	0xC000000
Loop	1
Verdict	
ltem(s)	Info
Address	0xC0000000
Loop(s)	1
Result	Pass
Result details	no error for 1 loops

Figure 356. DDR tests history

MCUs Se	lection	Output	DDR Interactive logs
Target	>	Host	step to 3:DDR_READY
Target	>	Host	1:DDR_CTRL_INIT_DONE
Target	>	Host	2:DDR PHY_INIT_DONE
Target	>	Host	3:DDR_READY
Host	>	Target	test 2 1 0xC0000000
Target	>	Host	execute 2:DataBusWalking0
Target	>	Host	running 1 loops at 0xc0000000
Target	>	Host	Result: Pass [no error for 1 loops]
Host	>	Target	test 3 1 0xC0000000
Target	>	Host	execute 3:DataBusWalkingl
Target	>	Host	running 1 loops at 0xc0000000
Target	>	Host	Result: Pass [no error for 1 loops]
Host	>	Target	test 4 4 0xC0000000
Target	>	Host	execute 4:AddressBus
Target	>	Host	Result: Pass [address 0xc0000000, size 0x4]
MCUs Se	lection	Output	DDR Interactive logs
Target 1	board (configura	tion name: DDR3-1066/888 bin G 2x4Gb 400.0.0.0.0MHz v1.45
Target 1	board I	DDR size:	8 GBits
Target 1	board l	DDR frequ	ency: 400.0MHz
Current	confi	guration	DDR registers loaded to the target board
DDR test	t #2 (1	DataBusWa	lking0) triggered with parameters: [loop] l [addr] 0xC0000000
DDR test	t #3 (1	DataBusWa	lkingl) triggered with parameters: [loop] l [addr] 0xC0000000
DDR test	t #4 ()	AddressBu	s) triggered with parameters: [size] 4 [addr] 0xC0000000

5.5 STM32CubeMX Memory Management Tool

The Memory Management Tool (MMT) displays the memory map and defines memory attributes applied in user projects opened/created in STM32CubeMX.

The tool is located in the "Tools" tab. It allows the user to declare memory regions (referred to as application regions or AppReg) at application level.

The HW constraints related to TrustZone, Memory Protection Unit, and the memory granularity are handled by MMT and made transparent to the user, so that the focus can be put on the memory regions. A linker file is generated according to the application regions declared and configured by the user.



The MMT key features are:

- Memory map display
- Application regions management
- Linker file generation

MMT interacts with peripherals starting from the moment the user enters its interface:

- Checks their settings
- Updates other peripherals involved in memory map configuration

The peripherals are updated only when the first toggle button is ON.

Figure 357. Regions settings to peripherals ON



MMT updates the linker scripts only when the second toggle button is ON.

Figure 358. Regions settings to linker files ON



The applicative regions are saved into the user project even if the first toggle button is OFF.

Figure 359. Regions settings to peripherals OFF



Apply Application Regions Settings to Linker Files: ON

0

5.5.1 STM32H5, STM32U5, STM32WBA, STM32WBAM, and STM32WBA6 with TrustZone activated

Feature: MMT usage, pinout, and configuration user interface

When the first toggle button is ON (see *Figure 357*), SAU, GTZC, Cortex-M33 (MPU), Cortex-M7_BOOT (MPU), Cortex-M7_APPLI (MPU), and FLASH configurations are under MMT control: their modes and parameters become read-only.



Figu	are 360. MMT usage	
	SAU Mode and Configuration	
	Mode	
Runtime contexts:		
Cortex-M33 secure	Cortex-M33 non secure	
۲		
✓ Enable SAU		
	Configuration	
	Comguration	
Reset Configuration		
Parameter Settings Subser Constants		
Configure the below parameters :		
Q Search (Ctrl+F) ③ ③		0
V Region 0		
Enable this region	Yes	
Start Address	0x0800000	
Block Size	0x04000000	
Secure Attribute	Non-Secure	
✓ Region 1		
Enable this region	Yes	
Start Address	0x0C1FE000	
Block Size	0x23E02000	
Secure Attribute	Non-Secure	
V Region 2		
Enable this region	Yes	
Start Address Block Size	0×9400000	
Secure Attribute	0x04000000 Non-Secure	
Secure Attribute	Non-Secure	
Enable this region	No	
Region 4	140	
Enable this region	No	
V Region 5		
Enable this region	No	
V Region 6		
Enable this region	No	
V Region 7		
Enable this region	No	

Feature: MMT usage and linker script



Apply Application Regions Settings to Peripherals: ON Apply Application Regions Settings to Linker Files: ON



Linker files content is generated according to the configuration of application regions.



Apply Application Regions Settings to Peripherals: ON 0 Apply Application Regions Settings to Linker Files: OFF 0 Apply Application Regions Settings to Peripherals: OFF A

Apply Application Regions Settings to Linker Files: ON

Linker files content is generated as if MMT is not used. SAU, GTZC, Cortex-M33 MPU, and FLASH are enabled, so that the user can modify the values supplied by MMT.

UM1718 Rev 47



a

	File Window	Help	L myST		🐵 🖪 💶 🄰 🖸 🔆
• > st	432U5A92JTxQ 🔰 mmt2.loc - Tools 🔪 LPBAM Scenar	io & Configura	ton >	GENERATE CODE	CHECK LPBAM DESIGN
	Pinout & Configuration		Clock Configuration	Project Manager	Tools
	Code Generation Configuration	~	Memory viewed by ARM Cottex-M33	Application Regions	+ 1
	Apply application Regions settings to peripherals: ON	0	Reserved		A 1
	Apply Application Regions settings to linker files: ON	0	OctoSP11 (128 MB)	NA	0
	Search an Application Region		Edenal RAM (OCTOSP11) (64 MB)	MyExternaRAM (RS)	
			External RAM (OCTOSP11) Remapped (64 MB)	MyRemappedRAM (3) Reserved Alias Region	
	Show selected Application Region	- I - I	740 Duild 201 MD	N/A	0
				NA	
		0	wissesson FMC Bankt (248 MD)	NA	0
			External Flash (FMC) (16 MB)	MyExternalFlash (S)	Cisplay Settings Reset View
			x50034000Reserved		Hide security beside regions names
			x51034400 Backup SRAM (S) (2 KB)		Hide Reserved Aliased regions names
			x40030000Reserved		8
			x40024400 Backup SRAM (NS) (2 KB)		
			x1800.cock		ê:
			SRAM4 Smart Run Domain (S) (16 KB)		
			Reserved		A
				MyThmadXOHeap (NS) Reserved Alias Region	A.
			x111A0000 SRAMS (3) (822 KB)		
		0	x30000000 (0) (0.1 (0.1 (0.1 (0.1 (0.1 (0.1 (0.1 (0.1	MBuffer INSI Reserved Alias Recion	
			SRAM2 (S) (64 KB)		
		0	x1000000	RAM (NS) Reserved Alias Region	
			x30000000 SRAM1 (5) (768 KB)	RAM (S)	
			*28004000 Reserved		ê :
			*21000000 SRAM4 Smart Run Domain (NS) (16 KB)		
			ATT TO THE RESERVED		8]
			SRAMS (NS) (832 KB)	MyThread/Disap (NS)	
			\$RAM3 (HS) (02 KB)		Logend
				MdBuffer (NS) Reserved Alag Region	Region allowing different types of security.
		- I.,	SRAM2 (NS) (64 KB)	RAM (NS)	
			x2000000 SRAM1 (NS) (768 KB)	RAM (SI Reserved Alast Region	Secure region (3).
					Non Secure region (NS).
			Nue conv	Harden and Annual Man Design	Non Secure Callable region (NSC).
			x10120000 SRAMS Code (S) (832 KB)	MyThreadOfHeap (NS) Reserved Alias Region	
			x16200000 SRAM3 Code (5) (832 KB)		Region accessible by Secure and Non-Secure.
			5RAM2 Code (5) (64 K8)	MyOuffer (NS) Reserved Alias Region	Reserved region.
			x1E00000	RAM (NS) Reserved Alias Region	
			xressource SRAM1 Code (S) (768 KB)	RAM (5) Reserved Alias Region	A new application region can be added here.
			x1C400000 Reserved		NA IP Configuration is expected. See tooltip into for further d
			sccassoco Flash Bank2 memory (8) (2 MB)	FLASH (NS) Reserved Alias Repon	e:
				FLASH (NSC (NSC)	
			Flash Bank1 memory (S) (2 MB)	FLASH (S)	
			-		A.

Figure 361. MMT view

5.5.2 An end-to-end usage example

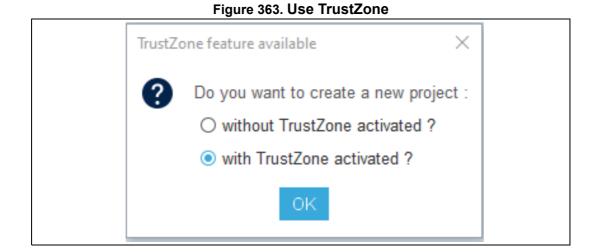
Choose a supported MCU (STM32U585x in this example).

Figure 362. Start a project





Press the "Start Project" button, and then choose the "with TrustZone activated ?" option.



Choose the "Tools" tab followed by the "Memory Management" option to display the Memory Management Tool (see *Figure 364*).

DOCUDENTA DES	ified*: STAB2US3SCEBx				- 0
2	File Window	Help 💄 myST			🚳 🖪 🗖 🎽 🕄 🖓
ne 🔪 sti	132U535CBTx 🔰 Untibled - Tools 🔪 LPBAW Scenario & Co	infiguration 🔪		GENERATE CODE	CHECK LPBAM DESIGN
	Pinout & Configuration		Clock Configuration	Project Manager	Tools
	Code Generation Configuration	1	Memory viewed by ARM Contex-M33	Application Re	aans +
	Apply application Regions settings to peripherals: OFF	(sakccoppo)			â
юry	Apply Application Regions settings to linker files: OFF	OctoSPI1 (256 MB)		NA	0
Inemeçe	Search an Application Region	outco accor Reserved			â
	Show selected Application Region	OxSO0364000 Backup SRAM (S) (2 KB)			
	Shew Selected Application Region	Ox40036000 Reserved			â
		0x40036400 Backup SRAM (NS) (2 KB)			
		0x35084000 Reserved			â
		Ox36000000 SRAM4 Smart Run Domain ((\$) (16 KB)		Display Settings Reset View
PCC		ox50040000Reserved			Hide security beside regions names
		0x30030000 SRAM2 (S) (64 KB)		RAM (NS) Reserved Alias Region	Hide Reserved Aliased regions names
		0x300000000 SRAM1 (S) (192 KB)		RAM (S)	Title Reserved Anases regults halves
		0x20004000 Reserved			â ;
		0x28000000 SRAM4 Smart Run Domain ((NS) (16 KB)		
		0x20040000 Reserved			â :
		(x20030000 SRAM2 (NS) (64 KB)		RAM (NS)	
CAD		0x200000000 SRAM1 (NS) (192 KB)		RAM (S) Reserved Alias Region	â ;
		Optitit 41000 Reserved			â
		OxCE030000 SRAM2 Code (S) (64 KB)		RAM (NS) Reserved Alias Region	â
		OxCE000000 SRAM1 Code (S) (192 KB)		RAM (S) Reserved Alias Region	ê
		0x0C620000 Reserved			û 🔭
				FLASH (NS) Reserved Alast Region	â
		Flash Bank1 memory (S) (12	18 KB)	FLASH_NSC (NSC)	
		0x0C000000		FLASH (S)	
		OxCAL42000 Reserved			Legend
		0x08/30000 SRAM2 Code (NS) (54 KB)		RAM (NS) Reserved Alias Region	
		OxC&/00000 SRAM1 Code (NS) (192 KB)		RAM (S) Reserved Alias Region	Region allowing different types of security.
		Ox06620000 Reserved			Secure region (S).
				FLASH (NS)	Non Secure region (NS).
		Flash Bank1 memory (NS) (1	(28 KB)	FLASH_NSC (S) Reserved Alias Region	
		0x0600000		FLASH (S) Reserved Alias Region	Non Secure Callable region (NSC).
		Om000000000 Reserved			Region accessible by Secure and Non-Secure.
					Reserved region.
					A new application region can be added here.
					NA IP Configuration is expected. See tooltp into far further
		1			

Figure 364. Default settings

The middle panel represents the memory, split into two columns: the left one is the memory seen by the core(s), the right one the memory set up for the application.



In this example there are two projects, a secure and a nonsecure one. The application region allocated to the secure project is green, the nonsecure application region is pink. The reserved memory regions are gray.

For the new project created under STM32CubeMX the tool creates the default application region to generate a valid project.

Region information

Clicking on a particular region in the Application Regions column shows the associated details on the left hand side.

You can choose to hide the name of the reserved region, or hide the Secure/Non Secure indication close to the region name (the secure/nonsecure indication is indicated by the color).



Figure 365. Region information

Code generation configuration

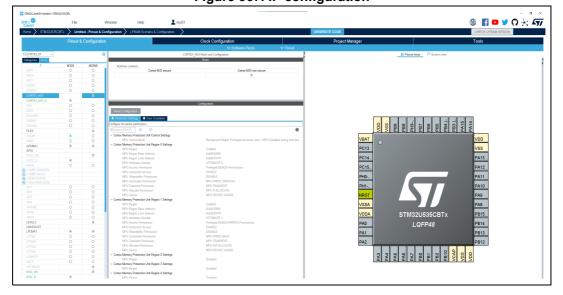
The application regions settings can be applied to peripherals on the left of the screen. The concerned peripherals are shown on the associated tooltip. This can impact their availability on the pinout screen configuration.

STM82Cube/MX Un	whiled": STM32U535CBTi							- 0 X
STM32	Fi	e Winds	ow H	lelp	💄 myST			🐵 🖪 🛛 🎽 🗘 🛧 🏹
Home > STI	W32U53SCBTx 🔪 L	Intitled - Tools $ angle$ LFBA	M Scenario & Config	juration 🔪		GENERATE CODE		CHECK LPBAM DESIGN
	Pino	ut & Configuration			Clock Configuration	Project Manager		Tools
	Code Generation Confi			*	Memory viewed by ARM Contex-N33	Application Regions		
	Apply applicati	on Regions settings to peripheral on Regions settings to linker file	s OFF	(walession)			â	
Memory Management	Search an Application	Region	to CO		s whether the Application Regions settings are applied LASH, GTZC (MPCBBs), SAU. OII, these peripherals become (partially) readonly	NA	0 â	
	Show selected Applica				sets Sets			
	RAM		Û	0x40036C00 ^{Res}	served SRAM (NS) (2 KB)		â	
	Name	RAM		0x40036400 0x38004000			£	
	Core Name	ARM Contex-M33	v		AM4 Smart Run Domain (S) (16 KB)		Display	Settings Reset View
PCC	Start address	0x3000000		0x30040000Res	served		🛍 💆 Hide	security beside regions names
	Size	192	KB v	0x30030000 SR/			â j	Reserved Aliased regions names
	Security	Secure	~	0x3000000 SR4	AM1 (S) (192 KB)	RAM		
	Default Data Region	2		(x21004000 Res	served		ŵ	
	Access Permission	RW by privileged code only	v	0x25000000 SR/	AM4 Smart Run Domain (NS) (16 KB)			
	Code execution	Permitted	v	0x20041000 Res	served		ŵ	
	Shareability	Non-Shareable	v	0x20030300 SR/	AM2 (NS) (64 KB)	RAM		
CAD	Cacheability	Write-Back Read Write Alloca	te for RA		AM1 (NS) (192 KB)		ê	

Figure 366. Tooltip



In this example, on the Pinout & Configuration panel, CORTEX_M33, FLASH, and GTZC are set, and correspond to the region configuration on the Memory Management Tool. They are grayed out, as they cannot be modified.





When an IP is under MMT control, a tooltip provides the info shown in *Figure 368*.

132 V JbeMX	File	W	/indow Help 💄 myST		
me 🔪 STM32U535	CBTx 🔪 Untit	led - Pinout & Con	nfiguration 🔪 LPBAM Scenario & Configuration 🛛 🔪		
	Pinout 8	Configuration		Clock Configuration	
				✓ Software Packs	
CORTEX_M ~		٢	GTZC_S M	Mode and Configuration	
ategories A->Z				Mode	
÷	M33S	M33NS	Runtime contexts:		
ADC1	0	0	Cortex-M33 secure	Cortex-M33 non secure	
ADC4	0	0	۲		
ADF1	0	0	Enable		
	0	0			
	0	0			
CORTEX_M33		۲			
CORTEX_M33_S	۲				
	0	0		Configuration	
	0	0	Reset Configuration		
DCACHE1	0	0			
	0	0	 TrustZone Illegal access Controller TrustZone Security Controller - Memory Protection Controlle 	er WaterMark Stock-Based Memory	NVIC Settings
	0	0	 Prost2one Security Controller - Memory Protection Controlle Parameter Settings TrustZone Security Controller - Security 		
FILEX		۲	Configure the below parameters :		T milegeable T enpirerais
FLASH	۲	0			
	0	0	Q Search (Ctrl+F) ③ ③		6
		0			
			V MPCWM1 (OCTOSPI1)		
GPDMA1			Configure Memory	from full Secured	
GPDMA1 GPIO			Configure Memory Area 1 Start Address	0×90000000	
GPDMA1 GPIO GTZC_NS		۲	Configure Memory Area 1 Start Address Area 1 Size	0×90000000 0×0	
GPDMA1 GPIO GTZC_NS GTZC_S: HASH Global TaistZor	۲	۲	Configure Mermory Area 1 Start Address Area 1 Sitze Area 1 Sicze	0x90000000 0x0 secured	
GPDMA1 GPIO GTZC_NS GTZC_S: HASH Global TaistZor	۲	0	Configure Memory Area 1 Start Address Area 1 Size Area 1 Secure Attribute Area 1 Privilege Attribute	0×90000000 0×0	
HASH I-CUBE- I-CUBE- Not available:	Controller	0	Configure Mermory Area 1 Start Address Area 1 Sitze Area 1 Secure Attribute	0x9000000 0x0 privileged	
GPDMA1 GPIO GTZC_NS GTZC_S: HASH I-CUBE I-CUBE I-CUBE I-CUBE I-DUBE IP under MMTO	e® Controller	0	Configure Memory Area 1 Start Address Area 1 Stare Area 1 Secure Atthobute Area 1 Phivliege Atthobute Lock the configuration of MPCWM1 Area 1 until next reset	0x90000000 0x0 secured privileged MPCWM1 Area 1 is not locked	
GPDMA1 GPIO GTZC_NS GTZC_S: HASH I-CUBE- I-CUBE- I-CUBE- I-CUBE- IP under MMT C I-Cube- details and docu	Controller	© 0	Configure Memory Area 1 Start Address Area 1 Stce Area 1 Secure Atthobute Area 1 Privilege Attribute Lock the configuration of MPCWM1 Area 1 until next reset Status of MPC/WM1 Area 1 Area 2 Start Address Area 2 Stce	0x9000000 0x0 secured privileged MPCWM1 Area 1 is not locked disabled	
GPDMA1 GPI0 GTZC_NS GTZC_NS GTZC_S: HASH Global TrustZor Status: ICUBE ICUBE IP under IMMT C ICUBE IP under MMT c details and doc	Controller	© 0 0	Configure Memory Area 1 Start Address Area 1 Stare Area 1 Secure Attribute Area 1 Phileige Attribute Lock the configuration of MPCVM1 Area 1 until next reset Status of MPCVM4 Area 1 Area 2 Start Address Area 2 Start Address Area 2 Stare	0x3000000 0x0 secured privileged MPCVMI1 Area 1 is not locked disabled 0x3000000 0x0 secured	
GPDMA1 GPIO GTZC_NS GTZC_S: HASH I-CUBE I-CUBE I-CUBE I-CUBE I-DUBE IP under MMTO	© te® Controller tontrol mentation (Ctrl+d)	© 0 0 0 0 0	Configure Memory Area 1 Start Address Area 1 Size Area 1 Size Area 1 Philege Attribute Lock the configuration of MPCWM1 Area 1 until next reset Status of MPCWM1 Area 1 Area 2 Start Address Area 2 Size Area 2 Size Area 2 Philege Attribute Area 2 Philege Attribute	0x3000000 0x0 secured MPCVMM1 Area 1 is not locked disabled 0x30000000 0x0 secured privileged	
GPDMA1 GPIO GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GT	Controller	 O O O O O O O O O 	Configure Memory Area 1 Start Address Area 1 Size Area 1 Secure Attribute Area 1 Secure Attribute Lock the configuration of MPCWM1 Area 1 until next reset Status of MPCWM1 Area 1 Area 2 Start Address Area 2 Size Area 2 Secure Attribute Area 2 Secure Attribute Area 2 Philiege Attribute Lock the configuration of MPCWM1 Area 2 until next reset	0x3000000 0x0 secured MPCVIMI Area 1 is not locked disabled 0x3000000 0x0 secured privileged MPCVMI Area 2 is not locked	
GPDMA1 GPI0 GFI2C_NS GTZC_NS GTZC_S: GCUBE GCUBE CCUBE CCUBE IP under MMT C IPCUBE IP under MMT C IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IPCUBE IP	Controller	 	Configure Memory Area 1 Start Address Area 1 Size Area 1 Size Area 1 Secure Attribute Lock the configuration of MPOWM1 Area 1 until next reset Status of MPCVM1 Area 1 Area 2 Size Area 2 Size Area 2 Size Area 2 Size Area 2 Privilege Attribute Lock the configuration of MPCWM1 Area 2 until next reset Status of MPCVM1 Area 2	0x3000000 0x0 secured MPCVMM1 Area 1 is not locked disabled 0x30000000 0x0 secured privileged	
GPDMA1 GPIO GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GTZC_1S GT	Controller	 O O O O O O O O O 	Configure Memory Area 1 Start Address Area 1 Size Area 1 Secure Attribute Area 1 Secure Attribute Lock the configuration of MPCWM1 Area 1 until next reset Status of MPCWM1 Area 1 Area 2 Start Address Area 2 Size Area 2 Secure Attribute Area 2 Secure Attribute Area 2 Philiege Attribute Lock the configuration of MPCWM1 Area 2 until next reset	0x3000000 0x0 secured MPCVIMI Area 1 is not locked disabled 0x3000000 0x0 secured privileged MPCVMI Area 2 is not locked	

Figure 368. IP under control

UM1718 Rev 47



Apply Application Regions settings to linker files

When this button is on, the linker scripts for the secure and non secure applications are generated, taking into account the configuration.

Figure	369.	Linker	files	update
--------	------	--------	-------	--------

ubeNX	Fi	e Window	Help 🔒	myST			🍥 🐵 🚮 🗖 🎽 🖓 🛧 🍕
ame 🔪 STM	ABQUEBECETX 🔪 I	Jintibled - Tools 🔪 LPBAM Scenario &	Contiguration 🔪		GENERATE CODE		CHECK LPBAM DESIGN
	Pinc	ut & Configuration		Clock Configuration	Project Manager		Tools
		n Regions settings to periphenals: ON	-	Memory viewed by ARM Contax M33		an Regions +	
emory anagement	Apply Applicat Search an Application	on Regions settings to linker files: ON	OutrossecooReserved	96 MD)		0	
	Show selected Applica	tion Region	Darkes SD.	AM (5) (2 KB)		â	
	RAM	D RAM	Ctex40036400 Backup SR	AM (NS) (2 KB)		â	
PCC	Core Name Start address	ARM Contex-M33 ~	Casteroccos SRAMA Sm Casteroccos Reserved	at Run Domain (S) (16 KB)		6	Display Settings Reset View
	Size Security	112 KB -				â	Hide Reserved Alased regions names
	Default Data Region	8	0x21004000Reserved	art Run Domain (MS) (16 KB)		Ĥ	
	Access Permission Code execution	RW by privileged code only	Cu20340000 Reserved			A	
64D	Shareability Cacheability	Non Shareable	0x210200000 SRAW2 (NS 0x210200000 SRAW1 (NS		RAM	8	

Configuring an external memory

This example uses the FMC. Go to the Pinout & Configuration window (see *Figure 370*) and enable the IP.



32 🗊 beMX		File	Window	Help	💄 myST			
	J5A9ZJTx0) Vntitl	ed - Pinout & Configuration	> LPBAM Scenario &	Configuration			
		Pinout &	Configuration		,	Clock C	onfiguration	
							✓ Software Packs	· · ·
	\sim	0			FMC Mode and	-		
egories A->Z					Moo	de		
ystem Core		>	Runtime contexts:	Cortex-M33 secure			Cortex-M33 non secure	
nalog		>		0			۲	
		>	VOR Flash/PSRAM/SRAM/R	OM/LCD 1				
mers			Chip Select NE1					~
onnectivity		~	Memory type NOR Flash					Max: 26 bits
\$	M33S	M33NS	Address 24 bits					▼ Nlax: 26 bits
FDCAN1	0	0	Data 8 bits	nia				~
FMC 12C1	0	•	Data 8 bits					Max: Disable
	0	0	Clock Disable					→ Max. Disable
	0	0						\vee
12C4	0	0	Address valid					
	0	0	Wait Disable					\sim
12C6	0	0	🔲 Byte enable					
IRTIM LPUART1	0	0	V NOR Flash/PSRAM/SRAM/R	OM/LCD 2				
	0	0	Chip Select Disable					~
	0	0	Memory type Disable					~
SDMMC1	0	0	Address Disable					🌲 Max: Disable
	0	0	LCD Register Select Disat	ble				\sim
SPI1	0	0	Data Disable					\sim
	0	0	Data/Address Disable					🌲 Max: Disable
UART4	0	0	Clock Disable					~
UART5	0	0			Configu	ration		
	0	0	Reset Configuration					
	0	0		Constants 🛛 📀 GPIO Sett	ings			
	0	0	Configure the below parameters :					
	0	0		0				
	0	0	V NOR/PSRAM control	-				
	-	-	Memory type			NOR Flash		
			Bank			Bank 1 NOR/PSRAM Disabled	1	
lultimedia		>	Write operation Write FIFO			Disabled Enabled		
ecurity		>	Extended mode			Disabled		
			V NOR/PSRAM timing			15		
omputing		>	Address setup time in Data setup time in HCL	K clock cycles	1	255		
Aiddleware and Soft	ware Packs	>	Data hold time in HCLP Bus turn around time in			D 15		
race and Debug		>						
Power and Thermal		>						

370 C nfic _ <u>____</u>

When going back to the MMT, a new region corresponding to the added FMC is created.

TMB2CubeMX Unt	aled": STMS2US492/Tr-Q							- 0 X
STM32	File	Window	Help	💄 myST			🚳 👖 🖸 🕽	। 🗘 🕆 🔊
Hame > STN	132U548ZJTxQ 🔰 Untitled - Tools	LPBAW Scen	rario & Configuration	\rangle	GENERATE CODE		CHECK LPBAM DES	IGN
	Pinout & Configur	ation		Clock Configuration	Project Manager		Tools	
	Code Generation Configuration		~	Memory viewed by ARM Contex-M33		Application Regions	+	1
Memory Management	Apply application Regions setting Apply Application Regions setting Search an Application Region Show selected Application Region		640 (1640 (1540 (1540 (1640 (Constraint	NA. NA. NA.		ů 0 0 0	Display Settings Reset Vis
PCC			014 014 013 013	Cotton Botton SPAM (5) (218) Cotton Botton SPAM (5) (218) Cotton Botton SPAM (5) (218) Cotton Botton Botton (5) (1518) Cotton Botton Domin (5) (1518)			â	Hide Security beside regic Hide Reserved Alased reg
CAD			163 163 163 163	Print Print Print <td></td> <td></td> <td>â</td> <td></td>			â	

Figure 371. New region created



Add a new region by pressing the plus button appearing in the white space when hovering with the mouse.

ACC D	File Window	He	elp 💄 myST		🕲 🖪 🖸 🄰 🗘 🛧 🔙
ame 🔪 STM	N32U549ZJTXQ $ angle$ Untitled - Tools $ angle$ LPBAM Scen	aria & Corfig	iguration 👌	GENERATE CODE	CHECK LPBAM DESIGN
	Pinout & Configuration		Clock Configuration	Project Manager	Tools
	Code Generation Configuration	_ ^ f	Memory viewed by ARM Contex-M33	Application F	Regions +
	Apply application Regions settings to peripherals: ON	0	takconcest Reserved		<u> </u>
етогу	Apply Application Regions settings to linker files: ON		ter9000000 OctoSPH (256 NB)	NA	0
anagement	Search an Application Region	\rightarrow	Instances FMC Bank3 (256 MB)	NA	0
	Show selected Application Region	>	1x7000000 OctuSPI2 (256 NB)	NA	0
			Infloccop FMC Bank1 (240 MB)	NA	0
			txecccccop External Flash (FMC) (16 MB)		
			ex5005ecto	Add an application region X	â
			tx50056400 Backup SRAM (S) (2 KB)	Name NyExternaFlash	Display Settings Rese
PCC			Ix40056C00 Reserved	Address [140000000	â 🛛 Hide security beside
			Info006400 Backup SRAM (NS) (2 KB)	Size 16 OB O KB ® MB	Hide Reserved Aliase
			1x30004100 Reserved		
			bx38000000 SRAM4 Smart Run Domain (S) (16 KB)	Cancel Add	
			executions Reserved		
			INCOLLAGON SRAMS (S) (S2 KB)		
		- I.	1x30000000 SRAM3 (S) (S2 KB) 		•



To add another external memory, go to the Pinout & Configuration view, and add OCTOSPI1 to Cortex-M33 Secure. Choose Single SPI, and specify Device Size and Device Type.

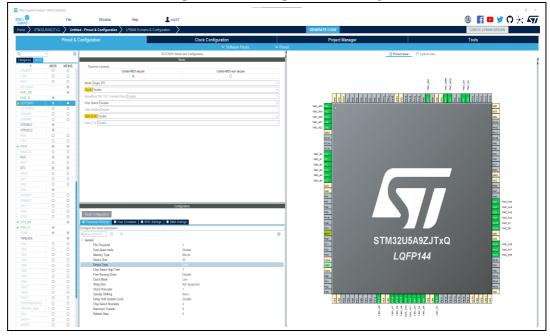


Figure 373. Adding a new memory

On the MMT there is now a new entry with OCTOSPI1.

- For our example, we need half of the available 128 Mbytes.
- Press the "+" button, set a name for the region (for instance: MyExternalRAM), and put 64 MB for its size.



ubeNX	F	ile Windo	w He	lp 💄 myST			🥸 🚹 🕨	🎽 🗘 🛧 🖣
ame 🔪 ST	MISQUISABIZITIKO 🔪	Untitled - Tools 🔪 LPBA	M Scenario & Corfig	praton >	GENERATE CODE		CHECK LPBAM DE	SIGN
	Pinc	out & Configuration		Clock Configuration	Project Manager		Tools	
	Code Generation Con		}	Memory viewed by ARM Contex-M33		Application Regions	+	
		tion Regions settings to peripherals tion Regions settings to linker files		n=200000000 Reserved			ê	
	 Other others 	our regula seurge to inverses	t on the	tx9000000 External RAM (OCTOSPI1) (256 MB)				
	Search an Application	n Region	\rightarrow	DECOCOCCUS FINC Bank3 (256 MB)	NA	Add an application region X	0	
	Show selected Applic	ation Region		1x7000000 OctoSPI2 (256 MB)	NA	Name MyExternalRAM	0	
				ExeCoccess FMC Bank1 (240 MB)	NA	Address 0x50000000	0	
	WyExternalFlash		Û	Execoccess External Flash (FMC) (16 MB)	NyExternalFlash	Size 64 O B O KB @ MB		
	Name	MyExternalFlash		Ex50034CE3 Reserved			â	
	Core Name	ARM Cates-M33	v	1x50036400 Backup SRAM (S) (2 KB)		Castel Add		Display Settings Re:
	Start address	0+5000000		Is40056000 Reserved			â	Hide security beside
	Size	16	MB v	Is40056400 Backup SRAM (NS) (2 KB)				Hide Reserved Alias
	Security	Secure	v	0x30004100 Reserved				
	Access Permission	RO by privileged code anly	×	88AM4 Smart Run Demain (S) (16 KB)				
	Code execution	Permitted	v	Reserved				
	Shareability	Nor-Shareable	×	(x20124(10) SPAM5 (5) (832 KB)				
	Cacheability	Write-Through Read Allocate fo	or ROM/FL Y	1x20000000 SPAMB (S) (S2 KB) 1x20000000 SPAMB (S) (S4 KB)				
				Indoocceep devalue (o) (or Ho) Indoocceep SRAM1 (S) (768 HB)	RAM			
				Inspected of the second of the				
				hz20004880 Westerner				
				hz20000000 Reserved			A	A.V.
				hz012/000 SRAM5 (NS) (882 KB)				
				1x2000000 SRAM3 (NS) (832 KB)	RAM			
				122000000 SRAM2 (NS) (64 KB)				1
				SRAM1 (NS) (768 KB)			â	

Figure 374. Memory assignment

Configuring a memory region using the left panel

With the left panel (see *Figure 375*) you can adjust items such as starting position and size. In this example, the added region must be adjusted: we want it to be allocated to the non secure project, and to start in the middle of the RAM. By adjusting those values, the expected results appear (see *Figure 376*). The color is now pink (nonsecure), and the region starts in the middle of the RAM (OctoSPI1).

		Untitled - Tools	LPBAM Scenario	1.0	\	GENERATE CODE		CHECK LPBAM D	🎽 🗘 🔆 石
me / SIN				s Compliation	/				IS/GN
	Pinc	out & Configuration	n		Clock Configuration	Project Manager		Tools	
	Code Generation Con	Iguration		~ }	Memory viewed by ARM Contex-M33		Application Regions	+	1
	Apply applicat	ion Regions settings to p	eripherals: ON	0	Reserved			â	
mory	Apply Application	ion Regions settings to lin	nkertles: ON	0 ^{1x3000}					
inagement	Search an Application	Region) (129000	External RAM (OCTOSPI1) (256 MB) C000	MyExternalRAM			
	Show selected Applic	ation Region		v 018000	FMC Bank3 (256 MB)	NA		0	
				0x7000	0110 OctoSPI2 (256 MB)	NA		0	
	MyExternalRAM		Û		(()) FMC Bank1 (240 MB)	NA		0	
	Name	MyExternalRAM			(()) External Flash (FMC) (16 MB)	MyExternalFlash			
	Core Name	ARM Cortex-M33			CT17 Reserved			â	Display Settings Rese
PCC	Start address	010000000			(41) Backup SRAM (S) (2 KB)				Hide security beside
	Size	64	WB	· 0x4003	ectop Reserved				Hide Reserved Aliase
	Security	Secure			Backup SRAM (NS) (2 KB)				
	Access Permission	RW by privileged code	cely	≤ 0x3800	(10) Reserved (10) SRAMA Smart Ran Domain (S) (16 KB)				
	Code execution	Pemited			Coop Reserved			A	
	Shareability	Non-Shareable Write-Back Read Write		0x3027	(*************************************				
CAD	Cacheability	VITRE-Dack Keap VITB	e Palocase for IoA		SRAM3 (S) (832 KB)			â	
					SRAM2 (S) (64 HB)				
					SRAM1 (S) (768 KB)	RAM			
					Rasserved			â	
				1+2000	SRAM4 Smart Run Domain (NS) (16 KB)				A.T.

Figure 375. Left panel configuration

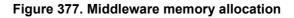


22 beWX	F	ile Window	Help	💄 myST			i 🛛 🚯 📑 🕒	0 🔆 🕻
me 🔪 STI	VISQUSA9ZJTKO 🔪	Untitled - Tools > LPBAM Sci	enario & Configuration	\rangle	GENERATE CODE		CHECK LPBAM DES	IGN
	Pin	out & Configuration		Clock Configuration	Project Manager		Tools	
	Code Generation Con	figuration	~	Memory viewed by ARM Contex-M33		Application Regions	+	1
		tion Regions settings to peripherals: ON dian Regions settings to linker files: ON		,,,,,,,Reserved			â	
mory nagement	Search an Application	1 Region	,	External RAM (OCTOSP11) (256 MB)	WexternalRAM			
			~ (at)	FMC Bank3 (256 MB)	NA		θ	
	NyExternalRAM			000000 Dote SP12 (256 MB) 	NA NA			
	Name	MiExternaRAM		External Flash (FWC) (16 MB)	WexternaFlash			
	Core Name	ARM Cotex-M33		13 ACCOR			â	Display Settings Re
PCC	Start address	0x50000000		135400 Backup SRAM (S) (2 KB)				Hide security besi
1.00	Size	64		1034CCc Reserved			â	Hide Reserved Alla
	Security	Non-Secure		036400 Backup SRAM (NS) (2 KB)				
	Secure Read/Write	Not Allowed		Notecon Reserved			â	
	Access Permission	RW by privileged code only		SRAM Smart Run Domain (S) (16 KB)				
	Code execution	Permitted	v (s3)	270000 Reserved			â	
	Shareability	Non-Shareable		140000 SRAWS (S) (832 KB)				
CAD	Cacheability	Write-Back Read Write Allocate for F		SRAM3 (S) (832 KB)			ê.	
				SRAN2 (S) (S4 KB)				
				SRAM1 (S) (768 KB)	RAM			
			(#28	1004000 Reserved			â	

Figure 376. Allocating a region

Setting up a middleware memory location

The application needs ThreadX. Go back to the "Pinout & Configuration" tab. Choose ThreadX, then use the Use Dynamic Allocation under Memory Configuration.



M32 W		File	Window Help	L myST			
ome 🗲 STM32	J6A9ZJT×C) 🔰 Untitle	ed - Pinout & Configuration > LPBAM Scena	rio & Configuration			GENERA
		Pinout & C	Configuration	Clo	ck Configuration		
		۵		THREADX Mode and Configuration	'n		
ategories A->Z				Mode			
	0	0	Runtime contexts:				
	õ	0	Cortex-M33 secure		Cortex-M33 non secure		
	0	0			۲		
	ō	0	Core				
	0	0	PerformanceInfo				
	0	0	TraceX Support				
	0	0	Low Power support				
	0	0					FMC_A23
	0	0		Configuration			FMC_A10
	0	0	Reset Configuration				FMG_A20
							PMC_A21 PMC_A22
		· · ·	ThreadX Sector Constants				PM0_A22
Multimedia			Configure the below parameters :				
Security		>	Q Search (Ctrl+F) ③ ④			0	
,			TX_TIMER_TICKS_PER_SECOND	3∠ 100			
Computing		>	ALIGN TYPE DEFINED	Disabled			FMC_AD
Middleware and Soft		~	TX MEMSET	memset			FMQ_A1
Middleware and Soft	ware Packs		TX_NO_FILEX_POINTER	Disabled			FMC_A2
•	M33S	M33NS	TX_THREAD_SECURE_STACK_MINIMUM	256			PMC_A3
FILEX		۲	TX_THREAD_SECURE_STACK_MAXIMUM				FMC_A4
FP-SNS-MOTE.			Enable BASEPRI support TX DISABLE ERROR CHECKING	Disabled			PMC_A8
I-CUBE-CANOP I-CUBE-Cesium			V Timer	Disabled			
I-CUBE-wolfSSL			TX_TIMER_PROCESS_IN_ISR	Disabled			
I-Cube-SoM-uG.			TX REACTIVATE INLINE	Disabled			
LEVELX		۲	TX_TIMER_THREAD_STACK_SIZE	1024 Bytes			
NETXDUO		۲	TX_TIMER_THREAD_PRIORITY	0			
THREADX		0	✓ Version				
		0	ThreadX version	6.2.0			
	0	0	 Memory Configuration Memory Pool Allocation 	Use Dynamic Al	landing		
USBX		۲		Ose Dynamic A	NC3000		
X-CUBE-AI			Memory Pool Allocation				
X-CUBE-ALS			Memory Pool Allocation				
★ X-CUBE-BLE1			Parameter Description: Use Static Allocation : use predefined threadX memo	onu poolo for allocation			
X-CUBE-BLE2			a static TX BYTE POOL is created using the tx byt		e with tx byte allocate()		
X-CUBE-BLEMC			Use Dynamic Allocation : use dynamic allocation sta	arting from the first free memory byte in the RA	M		
X-CUBE-DISPL			Application should used that first free memory location				
X-CUBE-FREE			know the first free memory region in the RAM to pass	it to the tx application define() in the "first			

To finish the configuration, go back to MMT. We want ThreadX to use a dedicated application region for its heap memory allocation. To do so, simply click the RAM region, and reduce its size to 17 Kbytes using the left panel. We then add a new region to the newly freed space, and call it MyThreadXHeap.

As ThreadX has been selected, on the Pinout & Configuration you can see a tick box called ThreadX Heap section. When this box is selected, the tool ensures that ThreadX memory allocation happens only in that particular region.



-	Jestied* STM32US46Z/B/Q			- a
32 teMX	File Window	Help L myST		🎯 🖬 🖸 🎽 🚳
me 🔪 ST	TM32U549ZJTxQ 🔰 Untitled - Tools 🔪 LPBAM Scena	is & Configuration	GENERATE CODE	CHECK LPBAM DESIGN
	Pinout & Configuration	Clock Configuration	Project Manager	Tools
	Code Generation Configuration	Wemory viewed by ARM Contex-1	M33 Application Regions	+ 1
	Apply application Regions settings to peripherals: ON Apply Application Regions settings to linker files: ON	e takonomi Reserved		â.
	Apply Application Regions settings to linker tiles: ON	•		
	Search an Application Region	External RAM (OCTOSPI1) (256 MB)	MyExternalRAM	
	Show selected Application Region	UNIXOUS PMC Bank3 (256 MB)	NA	0
		Darbourge OctoSPI2 (256 MB)	NA	0
		0x61000000 100 00000 1000	NA	0
	Name MyThread/Heap	Ins0000000 External Flash (FMC) (15 MB)	MyExternaFlash	Display Settings R
	Core Name ARM Contex-M33	v netocoscas Reserved		
	Start address (bx201A0000	astoo24400 Backup SRAM (S) (2 KB)		Hide security bes
		s v secondeces Reserved v secondeces Restruct SRAM (MS) (2 KB)		Hide Reserved Ali
	Security Inter-Secure Threadt Heap section	Executed and an and a second and a seco		
	Secure Read Write Not Allowed	bc35004689 SRAMA Smart Ran Demain (S) (15 KB)		
	Access Permission RW by privileged cade only	v natocross Reserved		â
	Code execution Permitted	REALING SPAMS (S) (622 KB)		6
	Sharabity No-Sharable	C 10000000 SRAM3 (5) (832 KB)		
	Cacheability Write-Back Read Write Allocate for RA.	SRAM2 (5) (64 KB)		
		0x500C0000		â
		0x30000000 SRAM1 (S) (766 KB)	RAM	
		1x22004100 Reserved		â
		5K220000000 SRAMA Smort Ran Domain (NS) (16 KB)		
		ex20270000 Reserved		â
		0x2012.000 SRAMS (NS) (822 KB)	MyThreadXHeap	
		0x20000000 SRAM3 (NS) (832 KB)		Legend
		SRAM2 (NS) (64 KB)	RAM	Region atoxi
		8x200001000 8x20000000 SIRAM1 (NS) (768 KB)	NAM .	A Secure reser
		0x20000000 text text (text)		
		Invitations SRAMS Code (S) (832 KB)		Non Secure I
		teriteccores SRAM3 Code (S) (832 KB)		Non Secure

Figure 378. Middleware heap configuration

Remap

For performance reasons, part of the application must run on the internal memory (much faster than the external memory). To do so, remap the added external RAM to an available internal memory region:

- Go to the Pinout & Configuration tab
- Enable ICACHE, select the Memory address remap tick box
- Select a region and set the memory size to 64 Mbytes
- Change the Remap address to 0x9000 0000

M32CubeMX Untitled*:	STM32U5A9	ZJTKQ			
32 🐨 beMX		File	Window Help	L myST	
	5A9ZJT×Q	> Untitl	led - Pinout & Configuration > LPBAM Scena	rio & Configuration 🔰	GENERATE CO
		Pinout &	Configuration	Clock Configuration	Pr
		0		ICACHE Mode and Configuration	1
itegories A->Z				Mode	
÷	M33S	M33NS	Runtime contexts:		
		۲	Cortex-M33 secure	Cortex-M33 non secure	
	۲		•	0	
	0	0	Memory address remap		
	0	0	Mode Disable		~
			Secure Attribute not secured		~
					FMC_A23 PE2
	0	0			
	0	0		Configuration	
	õ	0	Reset Configuration		FMC_A20 PE4
	0	õ			FMC_A21 PES
	0	0	Parameter Settings	MC Settings	PMC_A22 PE6
	0	0	Configure the below parameters :		VBAT
CACHE			Q Search (Ctrl+F) ③ ④		Pc13
	•	•	V Region 0		PC14.
	0	0	Region	Enable	PC15
	0	0	Size	2MB	FMC_AD PEO
EVELX		۲	Base Address Range	[0x0,0x07FFFFF]	FMC_A1 PE1
JNKEDLIST PDMA1	۲	۲	Base Address	0×0	FMC_A2 PF2
			Remap Address		FMC_A3 PF3
	0	0	Traffic Route	Master1 port	FMC_A4 PF4
	0	0	Output Burst Type	WRAP	FMC_A6 PF6
	0	0	✓ Region 1		VEE
	0	0	Region	Disable	VDD
	0	0	Region 2 Region 2		PFO
	0	0	Region	Disable	775 1977 1979 1979 1979 1970
UDF1	0	0	~ Region 3	Disable	PFB
IETXDUO		۲	Region	Unsable	PFB
IVIC_NS		۲			PF10
IVIC_S	۲				PHD
OCTOSPI1	۲	0			PH1
	0	0			NRST
OPAMP1	0	0			P00 P01 P02
	0	0			154

• Go back to the MMT tab. Region 0x9000 0000 is named with Remapped, with the amount of RAM previously selected.





.	Fil		Help	L myST		🧐 📑
ie 🗲 st			cenario & Configuration	>	GENERATE CODE	CHECK LPBA
		ut & Configuration		Clock Configuration	Project Manager	Tools
	Code Generation Config		~ F	Memory viewed by ARM Cortex-M	0	Application Regions
	Apply application	on Regions settings to peripherals: ON on Regions settings to linker files: ON		oosse Reserved		
	Appy Applicats	on Hegions settings to know lites: ON		00110 Octo8PI1 (128 MB)	NA	
	Search an Application I	Region	> 1894		MyExternalRAM	
	Show selected Applicat	ton Review		External RAM (OCTOSP11) Remapped (54 M5)		
		and ragon		PMC Bank3 (256 MB)	NA	
	MyThreadOUteap		0 0870	000110 OctuSP12 (256 MB)	NA	
	Name	MyThread@Heap	1161	ODDE PMC Bankt (240 MB)	NA	
	Core Name	ARM Cortex-M33	- 0x60	Course External Flash (FMC) (16 MB)	MyExternalFlash	
	Stat address	0x201A0000	1850	Office Reserved		
	Size	032	KB ~ (#50	State Backup SRAM (5) (2 KB)		
	Security	Nan-Secure	~ 0#40	acce Reserved		
	Thread/ Heap section	2	0840	24410 Backup SRAM (HS) (2 KB)		
	Secure Read/Write	Nat Allowed	- (atte	O+010 Reserved		
	Access Permission	RW by privileged code only	· 0+30	SRAM4 Smart Run Demain (5) (15 K8)		
	Code execution	Permitted	~ 0#300	TODIE Reserved		
	Shareability	Non-Shareable	V 0830	30310 SRAMS (S) (832 KB)		
	Cacheability	Write-Back Read Write Allocate for R	8A V 4#30	000110 SRAM3 (5) (832 KB)		
				SRAM2 (5) (64 K0)		
			0820	COMP		
			4820	000110 SRAM1 (S) (768 KB)	RAM	
			0+20	o asse Reserved		
				000010 SRAM4 Smart Run Demain (NS) (15 KB)		
			0x20	20016 Reserved		
			0.20	20010 SRAMS (NS) (832 KB)	MyThreadOleap	
			0820	DOTTE BRAM3 (NB) (832 KB)		
				SRAM2 (NS) (64 K9)		
			0.20	00000	RAM	
				00000 BRAM1 (NB) (768 KB)		
			0.00	Toose Reserved		
			1.00	ADDEC SRAMS Code (5) (832 KB)		
				00000 SRAM3 Code (8) (832 KB)		

Figure 380. Remapped region is renamed

• There is also a Remap – External RAM(OCTOSPI1) added at address 0x0000 0000.

Figure 381. Remapped start address

0x08400000	
0x08200000 Flash Bank2 memory (NS) (2 MB)	FLASH
Flash Bank1 memory (NS) (2 MB)	
0x08000000	
0x04000000Reserved	
0x0000000 Remap - External RAM (OCTOSPI1) (64 MB)	

• Add a new region named "MyRemappedRAM" at that address.

SRAM3 Code (NS) (832 KB)

SRAM2 Code (NS) (64 KB)

SRAM1 Code (NS) (768 KB)

Flash Bank1 memory (NS) (2 MB)

Reserved Flash Bank2 memory (NS) (2 MB)

Reserved Remap - External RAM (OCTOSPI1) (64 MB)

MB v

			· · · · · · · · · · · · · · · · · · ·	
onfiguration	~) }	Memory viewed by ARM Contex-M33	Application Regions
cation Regions settings to peripherals: ON cation Regions settings to linker files: ON	0	0x0E0C000	SRAM2 Code (S) (64 KB)	
		0x0E00000	SRAM1 Code (S) (768 KB)	
ion Region	>	0x0C40000		
lication Region	\vee		Flash Bank2 memory (S) (2 MB)	
	~		Flash Bank1 memory (S) (2 MB)	FLASH_NSC
м	U	0x0C00000		FLASH
MyRemappedRAM		0x0A27000	Reserved	
ARM Cortex-M33	~		SRAM5 Code (NS) (832 KB)	

MyBuffer

Figure 382. New region remapped

The default regions cannot be removed, but can be resized. As an example, the FLASH is where the application code is hosted. You cannot untick the Default Region.



8

Access P

Code ex

Secure RW by privileged code only

Permitted

Non-Shareable

Write-Back Read Write Allocate for RA... V

A.

<u>
</u>

â.

â

â

â

Г

ne 🔪 STRUZUEARZ/THO 🔪 mmt2.Jec - Tools 🔪 L/BEAM Scenario & Configuration 🔪 CENERATE CODE								CK LPBAM DESIGN		
	Pinout & Config	uration		Clock	Configuration	Project Mar	nager			Taals
	Code Generation Configur	ation	~	ř.	Memory viewed by ARM Cortex-1	//33	Application Regions	+ 1		
emory anagement		Regions settings to peripherals: ON Regions settings to linker files: ON	0	0x0E0C0001				â		
	Apply Application	regions settings to linker likes. On			SRAM1 Code (S) (768 KB)			â		
	Search an Application Re	gion	>	0x0C400000				â	Display Set	tings Reset View
PCC	Show selected Application	n Region	~		Flash Bank2 memory (S) (2 MB)			â	Hide set	curity beside regions names
			_			FLASH_NSC			Hide Re	eserved Aliased regions names
	FLASH		Ŭ	0x0C00000	Flash Bank1 memory (S) (2 MB)	FLASH				
CAD	Name	FLASH		0x0A270001				6		
CAD	Core Name	ARM Cortex-M33	~	0x03130000	SRAM5 Code (NS) (832 KB)			â		
	Start address	0x08200000	h an		SRAM3 Code (NS) (832 KB)		+		Legend	
	Size	2048 Non-Secure	KB V						<u> </u>	Region allowing different types of securi
	Default Code Region			0x0A0C0001	SRAM2 Code (NS) (64 KB)			â		region arrowing unream types of second
	ISR vector				SRAM1 Code (NS) (768 KB)			â	8	Secure region (S).
	Access Permission	RO by privileged code only	~	0x08400000				6	N	Non Secure region (NS).
	Code execution	Permitted	~		Flash Bank2 memory (NS) (2 MB)	FLASH			,	Von Secure Callable region (NSC).
	Shareability	Non-Shareable	~					â		ter erene ennen utfann (une).
	Cacheability	Write-Through Read Allocate for R	ow V	0x0800000	Flash Bank1 memory (NS) (2 MB)			â	P	Region accessible by Secure and Non-S
				0x04000000					Â	Reserved region.

Changing the security of an application region mapped on aliased RAM or FLASH moves it in an aliased RAM or FLASH corresponding to the new security setting. Graphically, the region moves up and down, depending on the area it will go, as the same physical memory is seen by the core at different locations.

	Code Generation Configur	ration V	2	Memory viewed by ARM Cortex-M33	Application Regions	+
		Regions settings to peripherals: ON Regions settings to linker files: ON	0x2800400			
Aemory Aanagement		Regions settings to linker lites. On	0x2800000	0 SRAM4 Smart Run Domain (NS) (16 KB)		
	Search an Application Re	gion >	0x2027000	0 ^{Reserved}		â
		- Desire	0x201A000	0 SRAM5 (NS) (832 KB)	MyThreadXHeap (NS)	
	Show selected Application	n Kegion Y		0 SRAM3 (NS) (832 KB)		
	MyBuffer	Û			MyBuffer (NS) Reserved Alias Region	6
	Name	MyBuffer	0x200C000	SRAM2 (NS) (64 KB)	RAM (NS)	
PCC	Core Name	ARM Cortex-M33		0 0 SRAM1 (NS) (765 KB)	RAM (S) Received Allas Region	â
	Start address	0x0A0C4400				 0
	Size	47 KB ~	0x0E27000			
	Security	Non-Secure V		0 0 8RAM5 Code (S) (832 KB)	MyThreadXHeap (NS) Reserved Alias Region	â
	ThreadX Heap section		0x0E0D000	0 SRAM3 Code (S) (832 KB)		
CAD	Secure Read/Write	Not Allowed		SRAM2 Code (S) (64 KB)	MyBuffer (NS) Reserved Alias Region	<u> </u>
0,10	Access Permission		0x0E0C000		RAM (NS) Reserved Alias Region	6
			0x0E00000	0 SRAM1 Code (S) (768 KB)	RAM (S) Reserved Alias Region	â
	Code execution	Permitted ~	0x0C40000			6
	Shareability	Non-Shareable V		0 Flash Bank2 memory (S) (2 MB)	FLASH (NS) Reserved Alias Region	6
	Cacheability	Write-Back Read Write Allocate for RA 🗸	0x0020000		FLASH_NSC (NSC)	
				Flash Bank1 memory (S) (2 MB)	FLASH (S)	
			0x0C00000			â
			0x0A27000	Oreserved		
				0 SRAM5 Code (NS) (832 KB)	MyThreadXHeap (NS) Reserved Alias Region	Â
			0x0A0D000	0 SRAM3 Code (NS) (832 KB)		
				SRAM2 Code (NS) (64 KB)	MyBuffer (NS)	
			0x0A0C000		RAM (NS) Reserved Alias Region	6
			0x020000	0 SRAM1 Code (NS) (768 KB)	RAM (S) Reserved Alias Region	â
			0x0840000			â
				0 Flash Bank2 memory (NS) (2 MB)	FLASH (NS)	
			0x0820000	0	FLASH_NSC (S) Reserved Alias Region	
				Flash Bank1 memory (NS) (2 MB)	FLASH (S) Reserved Alias Region	
			0x080000		runan (a) Reserved Allas Region	
			0x0400000	OReserved		â
			0x0000000	0Remap - External RAM (OCTOSPI1) (64 MB)	MyRemappedRAM (S)	

Figure 384. Region security change

Code generation

- Go to the project manager, set a name to your project, Choose CubeIDE as a toolchain and press GENERATE CODE
- Navigate to the generated Secure Project and open the linker definition file. Under the Memories definition you will see the defined memories with their start address and



length. This file shows only the secure regions in green. Open the nonsecure linker file and check the same location for the memory regions allocated to the nonsecure area.

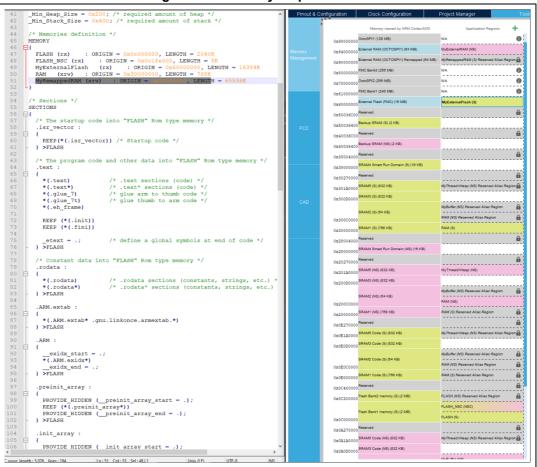


Figure 385. Memory map in linker file

5.5.3 STM32H7 single core and STM32U5 without TrustZone activated

Feature: MMT usage, pinout, and configuration user interface

When the first toggle button is ON, Cortex-M33 (MPU for STM32U5) and Cortex-M7 (MPU for STM32H7) are under MMT control (see, respectively, *Figure 386* and *Figure 387*): modes and parameters become read-only.

The middle panel (see, respectively, *Figure 388* and *Figure 389* for STM32U5 and STM32H7) represents the memory, split into two columns: the left one is the memory seen by the core(s), the right one the memory set-up for the application.

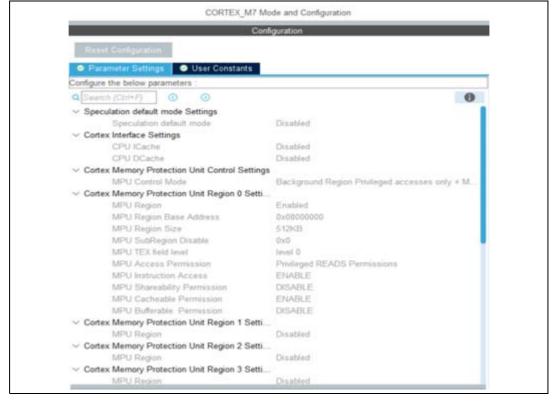
For the new project created under STM32CubeMX the tool creates the default application region to generate a valid project.



Figure 386	. MMT usage	(STM32U5)
------------	-------------	-----------

CORTEX_M	33 Mode and Configuration
	Configuration
Reset Configuration	
Parameter Settings Settings Settings Set User Constants	3
Configure the below parameters :	
Q Search (Ctrl+F) ③ ③	0
Cortex Memory Protection Unit Control Set	tings
MPU Control Mode	Background Region Privileged accesses only + M
Cortex Memory Protection Unit Region 0 S	etti
MPU Region	Enabled
MPU Region Base Address	0x08000000
MPU Region Limit Address	0x0803FFFF
MPU Attributes Number	ATTRIBUTE 0
MPU Access Permission	Privileged READS Permissions
MPU Instruction Access	ENABLE
MPU Shareability Permission	DISABLE
MPU Cacheable Permission	MPU WRITE THROUGH
MPU Transient Permission	MPU TRANSIENT
MPU Allocate Permission	MPU R ALLOCATE
MPU Device	MPU DEVICE nGnRnE
 Cortex Memory Protection Unit Region 1 S 	
MPU Region	Enabled
MPU Region Base Address	0×2000000
MPU Region Limit Address	0x2003FFFF
MPU Attributes Number	ATTRIBUTE 1
MPU Access Permission	Privileged READS/WRITES Permissions
MPU Instruction Access	ENABLE
MPU Shareability Permission	DISABLE
MPU Cacheable Permission	MPU WRITE BACK
MPU Transient Permission	MPU TRANSIENT





UM1718 Rev 47



Memory viewed by ARM Cortex-M33	Application Regions	+
xA0000000 Reserved		6
x90000000 OctoSPI1 (256 MB)	N/A	6
x40036C00 Reserved		6
x40036400 Backup SRAM (2 KB)		
x28004000 Reserved		â
x28000000 SRAM4 Smart Run Domain (16 KB)		
x20040000 Reserved		6
x20030000 SRAM2 (64 KB)	RAM	
x20000000 SRAM1 (192 KB)	RAW	
x0A040000 Reserved		6
x0A030000 SRAM2 Code (64 KB)	DAM Deserved Alian Design	
x0A000000 SRAM1 Code (192 KB)	RAM Reserved Alias Region	í de la companya de la compa
x08040000 Reserved		6
x08000000 Flash Bank1 memory (256 KB)	FLASH	
x00000000 Reserved		6

Figure 388. MMT view for U5 without TrustZone

Figure 389. MMT view for H7 single core

	32H723ZETx > Untitled - Tools >			GENERATE CODE
F	Pinout & Configuration	Clock Configuration	Project Manager	Tools
	Code Generation Configuration	Memory vie	wed by Arm Cortex-M7	Application Regions
	Apply Application Regions Settings to Peripherals: ON (0x90000000 OCUDPTI (200 MD)	IVA	, pproducer regions
emory	Apply Application Regions Settings to Linker Files: OFF	0x80000000	N/A	
anagement	Search an Application Region >	0x70000000 OctoSPI2 (256 MB)	N/A	
	Show selected Application Region >	0x60000000 FMC Bank1 (256 MB)	N/A	
	Show selected Application Region	0x38801000 Reserved		
		0x38800000 Backup SRAM (4 KB)		
		0x38004000 Reserved		
PCC		0x38000000 AHB SRAM Domain 3 (16 KB)	RAM_D3	
		0x30008000 Reserved		
		0x30004000 AHB SRAM2 (16 KB)	RAM_D2	
		0x30000000 AHB SRAM1 (16 KB)		
		0x24020000 Reserved	RAM	
		0x24000000 AXI SRAM (128 KB)	104M	
CAD		0x20020000 0x20000000 DTCM memory (128 KB)	DTCMRAM	
		0x1FF20000 0x1FF00000 System memory (128 KB)		
		0x10008000 Reserved		
		0x10008000 0x10004000 AHB SRAM2 Code (16 KB)		
		0x10004000 0x100000000 AHB SRAM1 Code (16 KB)		
R Test Suite		0x080800000 Reserved		
- rescould		0x08000000 Flash memory (512 KB)	FLASH	
		0x00010000 Reserved		
		0x000000000 [TCM memory (64 KB)	ITCMRAM	

The middle panel represents the memory, split into two columns: the left one is the memory seen by the core(s), the right one the memory set-up for the application.

For the new project created under STM32CubeMX the tool creates the default application region to generate a valid project. The default data region can be updated by the user to choose another region as RAM, but there must always be a default data region (*Figure 390*).



Pinout & Configuration CI		Clock Configuration	Project Manager		Tools		
	Code Generation C	onfiguration	~	Memory viewed	by Arm Cortex-M7	Application Regions	
		cation Regions Settings to Periphe cation Regions Settings to Linker f		0x50000000 CCUSPTT (200 MD) 0x80000000 FMC Bank3 (256 MB)	N/A		
	Search an Applicat	ion Region	>	0x70000000 0x60000000 FMC Bank1 (256 MB)	N/A N/A		
	Show selected App	lication Region	~	0x60000000 Reserved			
	RAM		Û	0x38800000 Backup SRAM (4 KB)			
	Name	RAM		0x38004000 0x38000000 AHB SRAM Domain 3 (16 KB)	RAM D3		
	Core Name	Arm Cortex-M7	~	0x380000000 Reserved			
	Start address Size	0x24000000 128	KB ~	0x30004000 AHB SRAM2 (16 KB)	RAM D2		
	Default Data Region		100	0x30000000 AHB SRAM1 (16 KB)			
	Access Permission	RW by any privilege level	~	0x24020000 Reserved 0x24000000 AXI SRAM (128 KB)	RAM		
	Code execution	Permitted	~	0x24000000 0x20020000 Reserved			
	Shareability Cacheability	Non-Shareable Write-Back Read Write Allocate	for RW Y	0x20000000 DTCM memory (128 KB)	DTCMRAM		
	Cacheability	White-Dack Read White Allocate		0x1FF20000 Reserved			
				0x1FF00000 System memory (128 KB) 0x10008000 Reserved			
				0x10004000 0x10004000 AHB SRAM2 Code (16 KB)			
				0x10000000 AHB SRAM1 Code (16 KB)			
				0x08100000 Reserved 0x08000000 Flash memory (1 MB)	FLASH		
				0x00000000 Reserved			

Figure 390. Default data region

FMC impact on MMT

When activating FMC and SDRAM Bank1, a tab mapping (see *Figure 391*) is displayed, with three options:

- 1. Default mapping (see *Figure 392*): MMT initializes as default position of SDRAM Bank1, SDRAM Bank2, and NOR PSRAM (default viewer of MMT)
- 2. NOR/PSRAM bank and SDRAM Bank1/2 are swapped: MMT swaps the position of SDRAM Bank1 and NOR PSRAM Bank1 (see *Figure 393* and *Figure 394*)
- 3. SDRAM Bank2 remapped on FMC Bank2 and still accessible at default mapping: MMT updates the position of SDRAM Bank1 to be remapped on position of FMC Bank2 (see *Figure 395* and *Figure 396*)



	FMC Mode and Configuration
	Mode
> NOR Flash/PSRAM/SRAM/ROM/LCD 1	
NOR Flash/PSRAM/SRAM/ROM/LCD 2	
NOR Flash/PSRAM/SRAM/ROM/LCD 3	
NOR Flash/PSRAM/SRAM/ROM/LCD 4	
> NAND Flash 1	
> SDRAM 1	
Y SDRAM 2	
Clock and chip enable SDCKE1+SDNE1	*
Internal bank number 2 banks	>
Address 13 bits	Max: 13 bits
Data 16 bits	×
16-bit byte enable	
	Configuration
Reset Configuration	
	User Constants Strings GPIO Settings
	User Constants NVIC Settings GPIC Settings
Considere the percent by	
Q Search (Ctrl+F) () ()	0
Mapping parameters	
FMC bank mapping	SDRAM Bank2 remapped on FMC bank2 and still accessible at default mapping
	Default mapping NOR/PSRAM bank and SDRAM bank 1/bank2 are swapped
	NORP-SKAW bank and SDRAW bank trbanc2 are swapped SDRAW Bank2 remapped on FMC bank2 and still accessible at default mapping



Figure 392. Default mapping

Memory viewed by ARM Cortex-M7		Application Regions	+
xE0000000 Reserved			6
xD0000000 FMC SDRAM Bank2 (256 MB)	N/A		
xC0000000 FMC SDRAM Bank1 (256 MB)	N/A		
xA0000000 Reserved			6
_{0x9000000} QuadSPI (256 MB)	N/A		
0x80000000 FMC Bank3 (256 MB)	N/A		
x70000000Reserved			6
0x64000000 FMC Bank1 (192 MB)	N/A		0
Dx60000000 External Flash (FMC) (64 MB)			
x38801000Reserved			6
0x38800000 Backup SRAM (4 KB)		+	
0x38010000 Reserved			6
x38000000 AHB SRAM Domain 4 (64 KB)	RAM_D3		
0x30048000Reserved			6

Figure 393. Before the swap

	Memory viewed by Arm Cortex-M7		Application Regions	+
0xE0000000	Reserved			â
0xD0000000	FMC SDRAM Bank2 (256 MB)	N/A		
	FMC SDRAM Bank1 (256 MB)	N/A		
0xA0000000	Reserved			â
0x90000000	OctoSPI1 (256 MB)	N/A		
	FMC Bank3 (256 MB)	N/A		
	OctoSPI2 (256 MB)	N/A		
0x64000000	FMC Bank1 (192 MB)	N/A		0
0x60000000	External Flash (FMC) (64 MB)			
0x38801000	Reserved			â (
0x38800000	Backup SRAM (4 KB)			
0x38004000				â
0x38000000	AHB SRAM Domain 3 (16 KB)	RAM_D3		
0x30008000				â



Figure	394.	After	the	swap
--------	------	-------	-----	------

Memory viewed by Arm Cortex-M7		Application Regions	+
xE0000000 Reserved			۵
xD0000000 FMC SDRAM Bank2 (256 MB)	N/A		
xC4000000 FMC Bank1 (192 MB)	N/A		0
xC0000000 External Flash (FMC) (64 MB)			
xA000000			6
0x90000000 OctoSPI1 (256 MB)	N/A		
x80000000 FMC Bank3 (256 MB)	N/A		
x70000000 OctoSPI2 (256 MB)	N/A		
x60000000 FMC SDRAM Bank1 (256 MB)	N/A		
x38801000 Reserved			a ;
x38800000 Backup SRAM (4 KB)			
1x38004000 Reserved			۵
x38000000 AHB SRAM Domain 3 (16 KB)	RAM_D3		1
x30008000 Reserved			â

Figure 395. Before remapping

	Memory viewed by ARM Cortex-M7		Application Regions	+
0xE0000000 Reserved				6
0xD0000000 FMC SDR4	M Bank2 (256 MB)	N/A		
0xC0000000 FMC SDRA	M Bank1 (256 MB)	N/A		
0xA0000000 Reserved				a
0x90000000QuadSPI (2	256 MB)	N/A		
0x80000000 FMC Bank	3 (256 MB)	N/A		
0x70000000 Reserved				۵
0x64000000 FMC Bank	1 (192 MB)	N/A		0
0x60000000 External FI				
0x38801000 Reserved				6
0x38800000 Backup SR	AM (4 KB)		+	100000
0x38010000 Reserved				â
0x38000000 AHB SRAM	1 Domain 4 (64 KB)	RAM_D3		
0x30048000 Reserved				6
0x30040000 AHB SRAM	13 (32 KB)			



Memory viewed by ARM Cortex-M7	Application Regions	+
0000000 Reserved		â
	N/A	6
FMC SDRAM Bank2 (256 MB) 0000000	AppReg2 Reserved Alias Region	6
0800000 FMC SDRAM Bank1 (248 MB)	N/A	0
0000000 External RAM (FMC) (8 MB)	AppReg1	
0000000 Reserved		6
0000000 QuadSPI (256 MB)	N/A	
0000000 FMC Bank3 (256 MB)	N/A	
0400000 FMC SDRAM Bank2 (252 MB)	N/A	0
0000000 External RAM (FMC) (4 MB)	AppReg2	
0000000 FMC Bank1 (256 MB)	N/A	
8801000 Reserved		6
8800000 Backup SRAM (4 KB)	AppReg0	
8010000 Reserved		â
8000000 AHB SRAM Domain 4 (64 KB)	RAM_D3	
0048000 Reserved		â
0040000 AHB SRAM3 (32 KB)		
0020000 AHB SRAM2 (128 KB)	RAM_D2	
0000000 AHB SRAM1 (128 KB)		
4080000 Reserved		6
4000000 AXI SRAM (512 KB)	RAM	
0020000Reserved		6
0000000 DTCM memory (128 KB)	DTCMRAM	

Figure 396. After remapping



ETH impact on MMT for STM32H7 single core

An example of MMT configuration of the ETH IP on the STM32H723VETx MCU

- 1. Activate the IP ETH:
 - MMT creates three application regions within the MMT view.
 - To change the start address and the size of each region, update the ETH parameters.
- 2. Press the radio button "Apply Application region Settings to Peripherals ON", ETH will be partially under MMT control.
- 3. Press the Generate Code button to generate code for both applications.
 - Apply Application Regions settings to linker files:
- 4. When this button is on, the linker scripts are generated, considering the configuration.
- 5. After the code generation, navigate to the generated folder:
 - Open the linker definition file.
 - Under the Memories definition you can see the memories with their start address and length, according to the configuration made in STM32CubeMX.

Home > STM32H	723VETx 🔪 Untitled - Tools 🔪					GENERATE CODE	
Pino	ut & Configuration	Clock C	Configuration	Project Manager		Tools	
Compare Projects	Code Generation Configuration Apply Application Regions Settings to Peripherals: OFF Apply Application Regions Settings to Linker Files: OFF	0	0x900000000 CotoSPI1 (256 MB) 0x800000000 FMC Bank3 (256 MB)	lemory viewed by Arm Cortex-M7	N/A	Application Regions	+
	Search an Application Region Show selected Application Region	>	0x70000000 0x600000000 FMC Bank1 (256 MB) 0x38801000 Reserved		N/A N/A		â
Memory Management			0x38000000 Backup SRAM (4 KB) 0x38004000 Reserved 0x38000000 AHB SRAM Domain 3 (14 0x30000000 Reserved 0x30004000 AHB SRAM2 (16 KB)	3 KB)	RAM_D3		8
PCC			AHB SRAM1 (16 KB) 0×30000000		Rx_PoolSection TxDescripSection RxDescripSection		
			0x24020000 Reserved 0x24000000 AXI SRAM (128 KB) 0x20020000 Reserved		RAM		ê
			0x20000000 0x1FF20000 0x1FF20000 System memory (128 KB		DTCMRAM		â
CAD			0x10008000 0x10004000 0x10004000 0x10000000 AHB SRAM1 Code (16 K				a
			0x08080000 Reserved				â

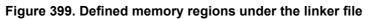
Figure 397. ETH MMT regions





STM32H723VETx Vintitled - I	Pinout & Configuration >			GENERATE CODE
Pinout & Configuration	Clock Configuration		Project Manager	Tools
	ETH Mode and Configuration		C Pinout view	System view
ries A+>Z	Mode			
activity ~	Mode MII ~	·		
÷	Activate Rx Err signal			
н	Activate Tx Err signal			
DCAN1 DCAN2	· · ·			
CAN3		_		2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
10	Configuration	_		
D1	Reset Configuration		PE3	
02		_		V(AP
03	NVIC Settings OPIO Settings		PC6	PA13.
C4 C5	Parameter Settings Ø User Constants		VEAT	
UART1	Configure the below parameters :		PC13	PAID
DIOS	Q Search (Ctrl+F) ()	0	100	7.40
CTOSPI1	General : Ethernet Configur	-	VII	PC0
CTOSPI2	Warning The ETH can work only when		710	20
DMMC1	Ethernet MAC Addr 00:80:E1:00:00:00		223	FC0
DMMC2	Tx Descriptor Length 4		TO A DECIMAL OF A	2211
911	First Tx Descriptor 0x30000080		ин ин ин колония ин ин ин ин ин ин ин ин ин ин ин ин ин	2246
912 913	Rx Descriptor Length 4		(TH_TAR) TO_	P942
P13	First Rx Descriptor 0x3000000		STM32H	1723VETx 🚟
26	Rx Buffers Address 0x30000100		<u></u>	-D100
WPM11	Rx Buffers Length 1536		ETH. CES 140	1723VETx 270
ART4			(18_82_0.K PAL	1914
IRT5			ETH_MBIO 742 ETH_COL PA1	7819 ETH, 7334 7917 ETH, 7334
RT7			ett Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Methodowe Me	
IRT8 IRT9				
ART9 SART1			AUCUL	2
SART2			Ecces	É
SARTS				
ART6				
SART10				
SB_OTG_HS			0 00 00	_
		Q []	Q 🕒 🖆 💷	Q Unused GPIOs:

Figure 398. ETH configuration for STM32H723VETx MCU



/	
8	/*-Sizes-*/ define symbolICFEDIT_intvec_start = 0x08000000;
10	define symboltrebit_intvet_start = 0x00000000,
11	define symbolICFEDIT_size_cstack_ = 0x400;
12 13	define symbolICFEDIT_size_heap = 0x200;
14	
15	/*-Start of symbols-Auto-generated By STM32CubeMX*/
16 17	define symbolICFEDIT_region_DTCMRAM_start = 0x20000000; define symbolICFEDIT_region_DTCMRAM_end = 0x2001FFFF;
18	define symbolICFEDIT_region_RAM_D3_start = 0x38000000;
19	define symbolICFEDIT_region_RAM_D3_end = 0x38003FFF;
20 21	define symbolICFEDIT_region_ITCMRAM_estart = 0x00000000; define symbolICFEDIT_region_ITCMRAM_end = 0xFFFF;
22	define symbolICFEDIT_region_RxDescripSection_start = 0x30000000;
23 24	define symbolICFEDIT_region_RxDescripSection_end0x3000007; define_symbolICFEDIT_region_RxDescripsection_etat0x3000007;
24 25	define symbolICFEDIT_region_TXDescripSection_start + 0X30000080; define symbolICFEDIT_region_TXDescripSection_end
26	define symbolICFEDIT_region_Rx_PoolSection_start = 0x30000100;
27 28	define symbolICFEDIT_region_Rx_PoolSection_end = 0x300006FF;
29	/*-End of MX Symbols*/
30	
31 32	/*-Symbols-*/ define symbolICFEDIT_region_RAM_start = 0x24000000;
33	define symbolICFEDIT_region_RAM_end = 0x24000000, define symbolICFEDIT_region_RAM_end = 0x2401FFFF;
34	define symbolICFEDIT_region_FLASH_start = 0x08000000;
35 36	define symbolICFEDIT_region_FLASH_end = 0x807FFFF; /**** End of ICF editor section_###ICF####/
37	/ End of let early sectors where the /
38	define memory mem with size = 4G;
39 40	/*-MEMORY Regions-*/
41	define region RAM_region = mem:[fromICFEDIT_region_RAM_start toICFEDIT_region_RAM_end];
42	define region FLASH_region = mem:[from _ICFEDIT_region_FLASH_starttoICFEDIT_region_FLASH_end];
45	/*-Start of Regions- Auto-generated By STM32CubeMX*/
45	define region DTCMRAM_region = mem:[fromICFEDIT_region_DTCMRAM_starttoICFEDIT_region_DTCMRAM_end];
46 47	define region RAM_D3_region = mem:[from _ICFEDII_region_RAM_D3_start_ to _ICFEDII_region_RAM_D3_end_]; define region_ITCMRAM_region = mem:[from ICFEDII region_ITCMRAM_start to ICFEDII region_ITCMRAM_end]; define region_ITCMRAM_region
48	
49	define region TxDescripSection_region = mem:[from _ICFEDIT_region_TxDescripSection_start_ to _ICFEDIT_region_TxDescripSection_end_]; define region Rx_PoolSection_region = mem:[from _ICFEDIT_region_Rx_PoolSection_start_ to _ICFEDIT_region_Rx_PoolSection_end_];
50 51	aetime Lediou xxThooirectiouTLediou = wew:[tLow TithinTuediouTxX_hooirectiouZtaut_ to TithinTuediouTxX_hooirectiouTeud_1;
52	/*-End of MX Regions-*/
53 54	/*-Blocks-*/
55	/~=010ckS-7/ define block CSTACK with alignment = 8, size =ICFEDIT_size_cstack {};
56	define block HEAPwith alignment = 8, size =ICFEDIT_size_heap{};
57 58	/*-Initialization strategies-*/
59	/ -antibilitation stategess / initialize by copy { readwrite };
60	do not initialize { section .noinit };
61 62	/*-Sections placements-*/
63	, securis procenters ;
64	/*-Start of _Sections- Auto-generated By STM32CUBEMX*/
65 66	place at address mem:_ICFEDIT_intvec_start { readonly section .intvec };
67	place in DTCMRAM_region { section DTCMRAM_section };
68	place in RAM_D3 region { section RAM_D3 section };
69 70	place in ITOWRAM region { section ITOWRAM section }; place in RVDescriptertion_region { section Rboscriptertion_section };
71	place in TxDescripSection_region { section TxDescripSection_section };
72	place in Rx_PoolSection_region { section Rx_PoolSection_section };
73	/*-end of MX Sections-*/
75	
76	/*-user Sections-*/
77	place in FLASM_region { readonly }; place in RAM_region { readonly };
79	place in FLASH_region { readwrite, block HEAP, block CSTACK };



5.5.4 STM32WBxx

Г

Feature: MMT usage, pinout, and configuration user interface

When the first toggle button is ON, Cortex-M33 is under MMT control: its modes and parameters become read-only (see *Figure 400*).



The user must select the core and the STM32Cube firmware from a list. It is possible to choose any STM32Cube firmware version (see *Figure 401*).

The list proposed to user contains only the firmwares found in STM32Cube_FW_WB_Vx/Projects / STM32_Copro_Wireless_Binaries/STM32WBxx (all .bin files). Firmware Update Service (FUS) and SafeBoot firmware are not proposed, so they are not in the MMT list.

This example is based on an STM32WB5x MCU, so the list must contain only stm32wb5x_x binaries. The button "Refresh" is used to refresh the binaries list version in the repository of STM32Cube firmware (see *Figure 402*).

Home > STM32WB55C0	CUx 🔪 Untitled - Tools 🔪			GENERATE CODE
Pinout &	Configuration	Clock Configuration	Project Manager	Tools
		Please Select a WB Cortex-M0+ firmware vers	sion from the list. Select Refresh	
Memory Management		stm32wb5x_BLE_HCILay	ver_fw.bin 1.19.0 V	
management		stm32wb5x_BLE_Stack_	fw.bin 1.4.0 ×	
		Stm32wb5x_BLE_Thread	_fw.bin 1.6.0 >	
		stm32wb5x_Mac_802_15	5_4_fw.bin 1.19.0 V	
PCC		stm32wb5x_rfmonitor_ph	y802_15_4_fw.bin 1.1.0 >	
		stm32wb5x_Thread_FTD	fw.bin 1.19.0 V	
		□ stm32wb5x_Thread_MTD	∫_fw.bin 1.19.0 ∨	
		stm32wb5x_BLE_HCI_Ac	dvScan_fw.bin 1.19.0 V	
CAD		□ stm32wb5x_BLE_LLD_fw	r.bin 1.18.0 V	
		stm32wb5x_BLE_Stack_	full_fw.bin 1.19.0 ~	
		stm32wb5x_BLE_Stack_	light_fw.bin 1.19.0 V	
		stm32wb5x_BLE_Thread	_dynamic_fw.bin 1.19.0 V	
DDR Test Suite		stm32wb5x_BLE_Thread	_static_fw.bin 1.19.0 V	
		stm32wb5x_BLE_Zigbee	_FFD_dynamic_fw.bin 1.19.0 V	
		stm32wb5x_BLE_Zigbee	_FFD_static_fw.bin 1.19.0 V	

Figure 400. MMT usage

Figure 401. Firmware version





Home > STN	132WB55CCUx 🔰 Untitled - Tools 🔪				GENERATE CODE
	Pinout & Configuration	Clock Config	uration	Project Manager	Tools
	Code Generation Configuration	~	Memory viewed by ARM Co	rtex-M4	Application Regions +
Memory Management	Apply Application Regions Settings to Peripherals: OFF Apply Application Regions Settings to Linker Files: OFF	0 0xA000000 0x9000000	Reserved QSPI (256 MB)	NA	<u>a</u> 0
	Search an Application Region Show selected Application Region	> 0x2004000	_o Reserved	CM0 Firmware Ram2b	≙ ≙
			SRAM2 (64 KB)	CM0 Firmware Ram2a	a
PCC		0x2003000 0x2001000		RAM_SHARED	â
		0x2000000		RAM	A
CAD		0x1001000	0 ⁻¹⁴²³⁸¹⁷⁸⁵⁴	CM0 Firmware Ram2b Rese	
			SRAM2 Code (64 KB)	CMD Firmware Ram2a Reser	
		0x1000000 0x0804000		RAM_SHARED Reserved Al	ias Region 🔒
DDR Test Suite		0x0800000	Flash memory (256 KB)	CM0 Firmware Flash FLASH	â
		0x0000000			a

Figure 402. MMT configuration for STM32WB5x

After selecting the binary firmware, the MMT view is displayed and the reserved regions of Cortex M0+ are created.

The middle panel represents the memory, split into two columns: the left one is the memory seen by the core(s) Cortex-M4, the right one the memory set-up for the application.

For the new project created under STM32CubeMX the tool creates the default application region to generate a valid project.

5.5.5 STM32H7 Dual-core without Trust Zone activated

Feature: MMT usage, pinout, and user interface configuration

When the first toggle button is ON, Cortex-M7_BOOT (MPU) and Cortex-M7_APPLI (MPU) are under MMT control: their modes and parameters become read-only.

8

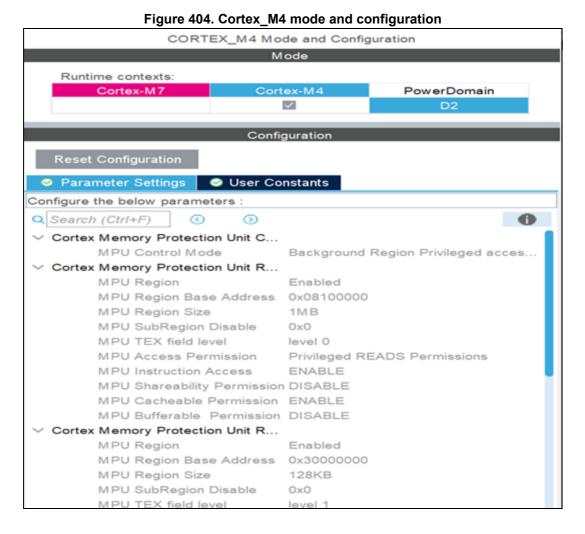
Apply Application Regions Settings to Peripherals: ON Apply Application Regions Settings to Linker Files: OFF





TM32	File	Window	Help		🕲 🚹 🗅 X 🗘 Q 🔆 🗸
Home STM32H75	i5XIHx > Untitled - Pinc	out & Configuration	\rangle		GENERATE CODE
Pinou	t & Configuration		Clock Configuration	Project Manager	Tools
<u>ر</u> ۲		٢		CORTEX_M7 Mode and Configuration	
ategories A->Z				Mode	
System Core		~ L	Runtime contexts:		
			Cortex-M7	Cortex-M4	PowerDomain
÷	Cortex-M7 Cort	tex-M4			D1
BDMA CORTEX_M4				Configuration	
CORTEX_M7	V				
DMA	_	Rea	et Configuration		
GPIO		🗢 Pa	rameter Settings 🛛 🤗 User Constants		
IWDG1	1	Configu	re the below parameters :		
IWDG2			rch (Ctrl+F) 0 0		
MDMA		Q 364			
NVIC1	V		CPU DCache tex Memory Protection Unit Control Settings	Disabled	
NVIC2		24	MPU Control Mode	Restaurad Reside Driviana	accesses only + MPU Disabled during hard fault, NMI and FAULTM
RAMECC			tex Memory Protection Unit Region 0 Settings	background Region Printeger	accesses only white orbitation during hard hade, hitr and though
RCC			MPU Region	Enabled	
✓ SYS_M4		2	MPU Region Base Address	0x0	
✓ SYS_M7	¥		MPU Region Size	4GB	
WWDG1			MPU SubRegion Disable	0x87	
WWDG2		12 C	MPU TEX field level	level 0	
			MPU Access Permission	ALL ACCESS NOT PERMIT	TED
			MPU Instruction Access	DISABLE	
Analog		>	MPU Shareability Permission	ENABLE	
			MPU Cacheable Permission	DISABLE	
Timers		>	MPU Bufferable Permission tex Memory Protection Unit Region 1 Settings	DISABLE	
0		> v cor	MPU Region	Enabled	
Connectivity			MPU Region MPU Region Base Address	0x0800000	
Multimedia		> •	MPU Region Size	1MB	
			MPU SubRegion Disable	0x0	
Security		>	MPU TEX field level	level 0	
			MPU Access Permission	Privileged READS Permission	18
Computing		>	MPU Instruction Access	ENABLE	
			MPU Shareability Permission	DISABLE	

Figure 403. Cortex_M7 mode and configuration



UM1718 Rev 47



0

A

Feature: MMT usage and linker script

When the two radio buttons are activated, the memory management parameters are available, and the linker file content is generated according to the configuration of application regions.

Apply Application Regions Settings to Peripherals: ON Apply Application Regions Settings to Linker Files: ON

There are two possible configurations of the application regions for the code generation:

Apply Application Regions Settings to Peripherals: ON Apply Application Regions Settings to Linker Files: OFF Second configuration: Apply Application Regions Settings to Peripherals: ON Apply Application Regions Settings to Linker Files: ON

The Cortex-M7 and Cortex-M4 contexts are managed by the MMT. Each context has its own application region (AppReg0 and AppReg1, respectively).

User interface

First configuration:

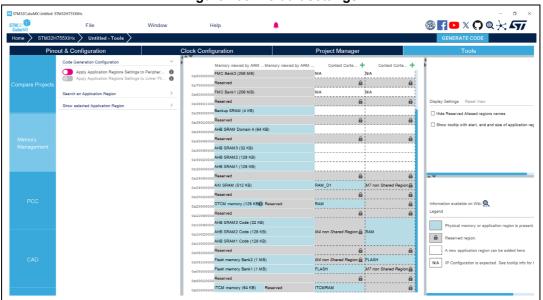


Figure 405. Default settings



The middle panel represents the memory, split into three columns: the left one is the memory seen by the cores (CM7 and CM4), the middle one the memory set-up for the application in Context Cortex-M7, the right one the memory set-up for the application in the Context Cortex-M4.

For the new project created under STM32CubeMX, the tool creates the default application region to generate a valid project.

Region information

Clicking on a particular region in the Application Regions column shows the associated details on the left hand side.

STM32CubeMX automatically adds a 4 Gbytes region for the system core, even if you are not planning to use the MMT.

An example of MMT configuration of the OPENAMP Middleware on the STM32H755XIH6TR MCU

Below are the steps for configuring the MMT with OPENAMP activated on the STM32H755XIH6TR MCU.

1. Choose a supported MCU.

A D C Commercial	_	Features	Block Diagram	Docs & Resources	CAD Resources	Datasheet	📑 Buy	. G•≋	Start Project
Part Number STM32H755XIH6TR Q + -	☆ ^S	TM32H7 Series	55XIH6TR	High-performance and D memory, 1MB RAM, 480 peripherals including a C	MHz CPU, Art Accelera	tor, L1 cache, externa			
RODUCT INFO V		ACTIVE Product is in mass prod	duction	Unit Price for 10kU (US\$) : 12.0232		TFBGA 2404	25 14x14x1.2 P 0.8	mm	
Line > Marketing Status > Price >		core at up to 240 MHz compliant), including a STM32H755xl device	z. Both cores feature a full set of DSP instru- is incorporate high-sp	igh-performance Arm [®] Cortex [®] -M7 au a floating point unit (FPU) which supp uctions and a memory protection unit seed embedded memories with a dual	ports Arm [®] single- and double-p (MPU) to enhance application	precision (Cortex [®] -M7 core) security. es, up to 1 Mbyte of RAM (in	operations and c	onversions (IEEE	754
Package >		matrix and a multi laye All the devices offer th control, five low-powe	er AXI interconnect si hree ADCs, two DACs er timers, a true rando	kup SRAM), as well as an extensive r upporting internal and external memor s, two ultra-low power comparators, a m number generator (RNG), and a cr advanced communication interfaces.	y access. a low-power RTC, a high-resolu yptographic acceleration cell. 1	tion timer, 12 general-purpo	e 16-bit timers, t	s, 2x32-bit multi-A wo PWM timers fo	HB bus
Core Check/lucheck All Check/lucheck All Arm Cortex-A35 Arm Cortex-A05 Arm Cortex-M0		matrix and a multi laye All the devices offer th control, five low-powe (DFSDM). They also f	er AXI interconnect si hree ADCs, two DACs er timers, a true rando	kup SRAM), as well as an extensive r upporting internal and external memor s, two ultra-low power comparators, a xm number generator (RNG), and a cr	y access. a low-power RTC, a high-resolu yptographic acceleration cell. 1	tion timer, 12 general-purpo	e 16-bit timers, t	s, 2x32-bit multi-A wo PWM timers fo	HB bus
Protage CheckUtcheck All Arm Contex-AS Arm Contex-AS Arm Contex-AS Arm Contex-AS Arm Contex-MO Arm Contex-MO Arm Contex-MO Arm Contex-MO	M CUs/N	mátrix and a multi lays All the devices offer th control, five low-powe (DFSDM). They also f Features IPUs List: 1 item	er AXI Interconnect s: hree ADCs, two DAC: r timers, a two rando feature standard and and the Pert No	Nup SRAMU, as well as an extensive rupporting internal and external memory, show turk-alwa power comparators, as munimer generators (RMS), and a extensive analyzanced communication interfaces. Pert No Reference Mathematication interfaces.	y access. I owe-power RTC, a high-resolu- yptographic acceleration cell. T	fion timer, 12 general-purpo The devices support four dig Package X RAM	ke 16-bit timers, t tal filters for exte	s, 2x32-bit multi-A wo PWM timers fo mal sigma-delta m	HB bus or motor iodulators
Prockage Pockage Core Check/lucheck Al Arm Cortex-A7 Arm Cortex-A35 Arm Cortex-M0 Arm Cortex-M0	M CUs/N	mátrix and a multi laye All the devices offer th control, five low-powe (DFSDM). They also f Features	er AXI Interconnect s: hree ADCs, two DAC: r timers, a two rando feature standard and and the Pert No	kup SRAM), as well as an extensive upporting internal memory typorting internal and external memory s, two ultra-low power comparators, and a c m number generator (RNO), autorator (RNO), autorator advanced communication interfaces.	y access. I owe-power RTC, a high-resolu- yptographic acceleration cell. T	tion timer, 12 general-purpo The devices support four dig	ke 16-bit timers, t tal filters for exte	s, 2x32-bit multi-A wo PWM timers fo	HB bus or motor kodulators
Protection Core Core Core Core Core Core CoreAlUncheck Al CoreAnAT Am Cortex-AT Am Cortex-AI	M CUs/N	mátrix and a multi lays All the devices offer th control, five low-powe (DFSDM). They also f Features IPUs List: 1 item	er AXI Interconnect s: hree ADCs, two DAC: r timers, a two rando feature standard and and the Pert No	Nup SRAMU, as well as an extensive rupporting internal and external memory, show turk-alwa power comparators, as munimer generators (RMS), and a extensive analyzanced communication interfaces. Pert No Reference Mathematication interfaces.	y access. I owe-power RTC, a high-resolu- yptographic acceleration cell. T	fion timer, 12 general-purpo The devices support four dig Package X RAM	ke 16-bit timers, t tal filters for exte	s, 2x32-bit multi-A wo PWM timers fo mal sigma-delta m	HB bus or motor iodulators
	M CUs/N	mátrix and a multi lays All the devices offer th control, five low-powe (DFSDM). They also f Features IPUs List: 1 item	er AXI Interconnect s: hree ADCs, two DAC: r timers, a two rando feature standard and and the Pert No	Nup SRAMU, as well as an extensive rupporting internal and external memory, show turk-alwa power comparators, as munimer generators (RMS), and a extensive analyzanced communication interfaces. Pert No Reference Mathematication interfaces.	y access. I owe-power RTC, a high-resolu- yptographic acceleration cell. T	fion timer, 12 general-purpo The devices support four dig Package X RAM	ke 16-bit timers, t tal filters for exte	s, 2x32-bit multi-A wo PWM timers fo mal sigma-delta m	HB bus or motor iodulators
	M CUs/N	mátrix and a multi lays All the devices offer th control, five low-powe (DFSDM). They also f Features IPUs List: 1 item	er AXI Interconnect s: hree ADCs, two DAC: r timers, a two rando feature standard and and the Pert No	Nup SRAMU, as well as an extensive rupporting internal and external memory, show turk-alwa power comparators, as munimer generators (RMS), and a extensive analyzanced communication interfaces. Pert No Reference Mathematication interfaces.	y access. I owe-power RTC, a high-resolu- yptographic acceleration cell. T	fion timer, 12 general-purpo The devices support four dig Package X RAM	ke 16-bit timers, t tal filters for exte	s, 2x32-bit multi-A wo PWM timers fo mal sigma-delta m	HB bus or motor iodulators
Protection Protection Core Core Core Core Core Core Core Core Corective Corect-A3 Corect-A3 Corective Am Cortex-A7 Corective Cortex-A4 Cortex-A5 Cortex-A5 Frequency - 400 (Mitz)	M CUs/N	mátrix and a multi lays All the devices offer th control, five low-powe (DFSDM). They also f Features IPUs List: 1 item	er AXI Interconnect s: hree ADCs, two DAC: r timers, a two rando feature standard and and the Pert No	Nup SRAMU, as well as an extensive rupporting internal and external memory, show turk-alwa power comparators, as munimer generators (RMS), and a extensive analyzanced communication interfaces. Pert No Reference Mathematication interfaces.	y access. I owe-power RTC, a high-resolu- yptographic acceleration cell. T	fion timer, 12 general-purpo The devices support four dig Package X RAM	ke 16-bit timers, t tal filters for exte	s, 2x32-bit multi-A wo PWM timers fo mal sigma-delta m	HB bus or motor iodulators

Figure 406. Choose an STM32H7 dual-core product

2. Click on the Start Project button, then choose Yes on the "Memory Protection Unit for Cortex-M7" dialog box.



STM324735XHx Untitled - Pinout & Configuration Clock Configuration Project Manager Tools Pinout & Configuration Clock Configuration Project Manager Tools Stm324735XHx Stm324735XHx Clock Configuration Configuration Stm324735XHx Clock Configuration Configuration Configuration Stm324735XHx Configuration Configuration Configuration Stm324735XHx Configuration Configuration Configuration Stm324735XHx Stm3247 Configuration Configuration Configuration Stm324735XHx Stm3247 Configuration Configuration Configuration Configuration Stm3247 Configuration Configuration Configuration Configuration Configuration MODA Configuration Configuration Configuration Configuration Configuration MOD2 Confi	12 ©MX	File	Window	Help			🍥 🔞 🚹 🕒 X 🗘 🍭 🔆 🕰
Contract Packs V Pinout Contract Mark Contract Mark Conter Mark Contrat		x 🔰 Untitled - Pinout	& Configuration >				
Integrine CORTEX_WT Mode and Configuration Integrine Mode Integrine Contraction Integrine Contraction </td <td>Pinout & C</td> <td>Configuration</td> <td></td> <td>Clock Configuration</td> <td></td> <td>Project Manager</td> <td>Tools</td>	Pinout & C	Configuration		Clock Configuration		Project Manager	Tools
Report Mode 1 Contact M // Contact M /				✓ Software Packs	✓ Pinout		
Provide AS2 Mode ystem Gore V 2 Contex M4 Contex M4 Perror Contain M4 BDMA Contex M4 Contex M4 Contex M4 Perror Contain M4 CONTEX_M4 Contex M4 Contex M4<		~	0			CORTEX M7 Mode and Configuration	
Participant Contract Mark Performa contracts • • Contract Mark	egories A->Z						
Ottow MT				Buntime contexts:			
BUM Configuration CORTEX_M Configuration CORTEX_M Configuration CORTEX_M Configuration CORTEX_M Configuration CONTEX_M Configuration GPI0 Configuration GPI0 Configuration MOD1 Configuration MOD2 Configuration MOD4 Configuration NNC1 Configuration Contiguration Contiguration NNC2 Configuration Contiguration Contiguration NNC2 Contiguration Contiguration Control failing Control failing Control failing Control Mode Bastronund Region Philoged accesses only + MPU Disabled during hard fault, VMO21 Control Mode Bastronund Region Philoged accesses only + MPU Disabled during hard fault, WV021 MPU Region Sea Address Oxid MVU Region Sea Address Oxid MVU Region Sea Address Oxid MVU Region Sea Address DisAllel MVU Region Sea Address Oxid MVU Region Sea Address	ystem Core		~			Cortex-M4	PowerDomain
CONTEX.M4 Image: Context MA Context MA DNA Image: Context MA Image: Context MA DNA Image: Context MA Image: Context MA MOD1 Image: Context MA Image: Context MA MOD2 Image: Context MA Image: Context MA MOD2 Image: Context MA Image: Context MA MOD3 Image: Context MA Image: Context MA MOD4 Image: Context MA Image: Context MA MOD2 Image: Context MA Image: Context MA Rec Image: Context MA Image: Context MA VerS_MA Image: Context MA Image: Context MA VWOD1 Image: Context MA Image: Context MA VWOD2 Image: Context MA Image: Context MA MVD Context Max Image: Context MA Image: Context MA MVD Context Max Image: Context MA Image: Context MA MVD Context Max Image: Context MA Image: Context MA MVD Context Max Image: Context MA Image: Context MA MVD Context Max Image: Context MA Image: Context MA MVD Context Max Image: Context MA Image: Context MA MVD Context Max Image: Context MA Image: Context MA MVD Context Max Image: Context MA	÷	Cortex-M7	Cortex-M4				D1
Contraction Met GPIQ						Configuration	
OAA Image: Certifying Image: Certif			21	Read Conferration			
GPI0 Import MOD1 Import MOD2 Import MOD2 Import MOD4 Import MOD4 Import MOD4 Import NNC1 Import AMMECO Import CC Import Securitation detail mode Enabled CPU Cache Dasabled CPU Cache Dasabled VMOD1 Import VMOD2 Import VMD Asplon Sas Address Ox07 MVD Asplon Sas Address Ox07 MVD Asplon Sas Address Ox07 MVD Asplon Sas Ad		V					
MODC1 Image: Configure the balow parameters : MODA2 Image: Configure the balow parameters : MODA Image: Configure the balow parameters : MOMA Image: Configure the balow parameters : Stription Configure the balow parameters : Image: Configure the balow parameters : Stription Configure the balow parameters : Image: Configure the balow parameters : Stription Configure the balow parameters : Image: Configure the balow parameters : Stription Configure the balow parameters : Image: Configure the balow parameters : Stription Configure the configure t				Parameter Settings Subser Co	nstants		
NNDC2 Image: Securitation default mode Settings Securitation default mode Settings NNDC1 Image: Securitation default mode Settings Disabiled AMMCO Image: Securitation default mode Settings Disabiled CC Image: Securitation default mode Settings Disabiled CPS_UAIA Image: Securitation default mode Settings Disabiled CPS_UAIA Image: Securitation default mode Settings Disabiled VMDO2 VMD default default mode Settings Disabiled VMDO2 VMD default default mode Settings Disabiled VMD default default mode Settings Disabiled VMD default default mode Settings VMD default default mode Settings Disabiled VMD default defa		2		Configure the below parameters :			
MDMA Speciation default mode Settings NINC1 Image: Speciation default mode Settings Enabled NINC2 Image: Speciation default mode Settings Cotte Interface Settings NINC2 Image: Speciation default mode Settings Cotte Interface Settings NINC3 Image: Speciation default mode Settings CPU Cottel Disabled NINC3 Image: Speciation default mode Settings CPU Cottel Disabled NINC3 Image: Speciation default mode Settings MDV Region Texterios Unit Footnot Settings MDV Region Set Address Cottel Memory Protection Unit Region Set Address NINC4 Image: Speciation default mode Settings MIV Region Set Address Cottel Memory Protection Unit Region Set Address Cottel Memory Protection Unit Region Set Address NINC4 Image: Speciation default mode Settings MIV Region Set Address Cottel Mit Protection Unit Region Set Address Cottel Mit Protection Unit Region Set Address NINT Region Set Address Image: Speciation Address Cottel Mit Protection Unit Region Set Address Cottel Mit Protection Address NINT Region Set Address Image: Speciation Address Permission Cottel Mit Protection Unit Region Settinge MIV Region Mit Region Settinge <td></td> <td>_</td> <td>2</td> <td>Q Search (Ctrl+F) (3) (5)</td> <td></td> <td></td> <td></td>		_	2	Q Search (Ctrl+F) (3) (5)			
NNCC Image Image Center starterse detings Disabled RAMCCO Image Disabled Disabled Disabled RCC Image Image Disabled Disabled RCC Image Image Disabled Disabled SYS_JAT Image Image Disabled Disabled WNDG2 Image Imagee Disabled Disabled WNDG2 Imagee Imagee Disabled Disabled WNDG2 Imagee Imagee Disabled Disabled WND Disabled Disabled			_	Speculation default mode Settings			
PAUMECO C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C </td <td>NVIC1</td> <td>52</td> <td></td> <td>Speculation default mode</td> <td></td> <td>Enabled</td> <td></td>	NVIC1	52		Speculation default mode		Enabled	
Image: Control in the image: Control Settings Control Settings MPU Databled Control Settings VMCD21 Image: Control Medical Settings Enabled Enabled VMCD21 Image: Control Medical Settings Enabled Image: Control Medical Settings VMCD2 Image: Control Medical Settings Enabled Image: Control Medical Settings VMCD2 Image: Control Medical Settings Image: Control Medical Settings Image: Control Medical Settings VMCD2 Image: Control Medical Settings Image: Control Medical Settings Image: Control Medical Settings VMCD2 Image: Control Medical Settings Image: Control Medical Settings Image: Control Medical Settings VMCD2 Image: Control Medical Settings Image: Control Medical Settings Image: Control Medical Settings VMCD2 Image: Control Medical Settings Image: Control Medical Settings Image: Control Medical Settings VMCD2 Image: Control Medical Settings Image: Control Medical Settings Image: Control Medical Settings VMTD Setting: Control Medical Settings Image: Control Medical Settings Image: Control Medical Settings Image: Control Medical Settings VMTD Region Tass Address	NVIC2		21				
NCC Mail Mail Cortes Memory Protocolo Unic Control Gettings 676_UA1 III MPU Control Memory Protocolo Unic Control Gettings MPU Control Memory Protocolo Unic Control Gettings 676_UA1 III MPU Control Memory Protocolo Unic Control Gettings Intel Memory Protocolo Unic Control Gettings WVD012 III MPU Control Memory Protocolo Unic Control Gettings Intel Memory Protocolo Unic Region Disclings MVD02 III MPU Region Sase Address 0x0 MPU Region Sase Address 0x0 MPU Region Sase Address 0x0 MPU Region Sase Address 0x0 MPU Region Sase Address 0x0 MPU Region Sase Address 0x0 MPU Region Sase Address 0x0 MPU Region Sase Address 0x0 MPU Region Sase Address 0x0 MPU Region Sase Address 0x0 MPU Region Sase Address 0x0 MPU Region Sase Address 0x0 MPU Region Sase Address 0x0 MPU Region Sase Address 0x0 MPU Region Sase Address 0x0 MPU Region Sase Address 0x0 MPU Region Sase Address 0x00	RAMECC						
MPU Control Mode Metry Control Mode Background Region Pholegod accesses only + MPU Disabled during hard fault. WWD01 Image: Second Region Pholegod accesses only + MPU Disabled during hard fault. Metry Region Second Region Pholegod accesses only + MPU Disabled during hard fault. WWD02 Image: Second Region Pholegod Accesses only + MPU Disabled during hard fault. Metry Region Second Region Pholegod accesses only + MPU Disabled during hard fault. wWD02 Image: Second Region Pholegod Accesses only + MPU Disabled during hard fault. Metry Region Second Region Pholegod Accesses only + MPU Disabled during hard fault. wWD02 Image: Second Region Pholegod Accesses only + MPU Disabled during hard fault. Metry Region Second Region Pholegod Accesses only + MPU Disabled during hard fault. wWD02 Image: Second Region Pholegod Accesses only + MPU Disabled during hard fault. Metry Region Second Region Pholegod Accesses only + MPU Disabled during hard fault. wWD02 Image: Second Region Pholegod Accesses only + MPU Disabled during hard fault. Metry Region Second Region Pholegod Accesses only + MPU Disabled during hard fault. wWD02 Image: Second Region Re	RCC	2	2			Disabled	
SYR_U/I B/d WVD01 B/d WVD02 B/d MVD Region Sear Address 0/d	SYS_M4		23		trol Settings	Destance of Desire Deside	and another MDU Disabled during hand facts MMI and
WMOD1 IM MMU Region Enabled WMO02 IM MMU Region Sale Address 0x37 MSD Region Sale Address 0x37 0x37 MSD Region Sale Address 0x38	SYS_M7				ion 0 Settings	Background Region Prime	ged accesses only + MPO Disabled during hard fault, NMT and
aligo Init Pro Region Sales Audres Cub aligo MPO Region Sales Cub aligo MPO Region Sales Cub mers MPO TEXT MERS DisA metsion DISALE Init O Construction metsion DISALE Enabled atimeda MPU Text Initiano DISALE cuty MPU Carbon Section Unit Region 1 Settings Unit Region 1 Settings cuty MPU Region Sea Address Cutodococc		2			ion o county	Enabled	
MPU QuBRegion Diacitie 0x87 Ango MPU TEX Refer level level 0 MPU TEX Refer level level 0 level 0 MPU TEX Refer level MPU TEX Refer level 1 level 0 MPU TEX Refer level MPU Textrictor Access DISALE MPU Usanteability Permission DISALE MPU Depression MPU Depression Permission DISALE MPU Depression MPU Repont Sea Address Ox800000 Enabled	WWDG2		21	MPU Region Base Address		0x0	
adog >> MUT TEX fed leval leval mers >> MUT Access Primision ALL ACCESS NOT PERMITTED mers >> MUT Instruction Access DISABLE inmedia >> MUT Access Primision DISABLE Atimedia >> MUT Access Primision DISABLE curly >> MUT Access Primision DISABLE curly >> MUT Access Primision DISABLE MUT Access Primision DISABLE DISABLE				MPU Region Size		4GB	
marka MM U Access Permission ALL ACCESS NOP PERMITTED marka MM U Instruction Access DISABLE montchild MM U Instruction Access DISABLE MM U Instruction Access DISABLE MM U Instruction Access MM U Instruction Access DISABLE MM U Instruction Access MM U Instruction Access DISABLE MM U Instruction Access Instruction Access DISABLE MM U Instruction Access Instruction Access DISABLE DISABLE Carter Among Protection Unit Region Statings Enabled MM U Region Task Address OnGODODO						0x87	
next% MOU Instruction Access DISABLE next/kig MPU Sharkability Permission ENABLE MPU Sharkability Permission DISABLE MPU Sharkability Permission DISABLE MPU Sharkability Permission DISABLE Control MPU Sharkability Permission DISABLE Vimedia MPU Sharkability Permission DISABLE Control Control DISABLE Vimedia MPU Region 1 Settings Enabled MPU Region Task Address OnS00000 DISABLE	alog		>				
wm w - MM view side Disklet mosthing - MM view side Disklet timeda - MM view side Disklet writy - - MM view side Disklet			× 1				IITTED
xmmetliky > MPU Cacheable Permission DISABLE xmmedia >> MPU darkeable Permission DISABLE xmmedia >> Codes Memory Protection Unit Region 1 Settings curlty > MPU Region Testings Enabled MPU Region Stark Address	ners						
Attractada >> M/U Bufferasias Permission DISABLE Control Munop Protocol Nut Region Settings MPU Region Stark Address Condectionau A Federal Settings MPU Region Stark Address Condectionau A Settings MPU Region Stark Address Condectionau A Settings MPU Region Stark Address Condectionau A Settings Condectionau A Settings Condectings Condectings Condectings Condectings Cond	ennectivity		>				
Attimuda Cortex Memory Protection Unit Region 1 Settings cwlty V Cortex Memory Protection Unit Region 1 Settings wMV Region State Address 0x6800000							
curity > MPU Region Base Address 0x08000000	ultimedia		>		ion 1 Settings		
MPU Region Base Address 0x08000000				MPU Region		Enabled	
	ecurity		>				
	mouting		、 、	MPU Region Size		1MB	
Imputing / MPU babRegion Diable 0x0 MPU DUTEX field Hewel Invest 1	amparing						

Figure 407. Region 0 added

Note: STM32CubeMX applies the default configuration, then adds a 4 Gbytes region called "Region 0" under the Cortex_M7 parameters. The new parameters can be checked using the Pinout and Configuration tab.

- 3. Select "Tools" in the toolbar
 - Choose Memory Management.
 - Activate the Memory Management Tool support by clicking the button "Apply Application Regions Settings to Peripherals".

	File	Window	Help			🧐 🕇 🕒	• X 🗘 @ 🔆 🖅
na nbeMX ome STM32H7	755XIHx 🔰 Untitled - Pinou	ut & Configuration >				GENE	RATE CODE
Pino	ut & Configuration		ock Configuration		Project Manager		Tools
		0	✓ Software Packs	✓ Pinout			
ategories A->Z	~				CORTEX_M7 Mode and Configurat Mode	ion	
M32CubeMX Untitled*: S	TM32H755XIHx						- 0
132 V JbeMX	File	Window	Help			🐵 🖬 🕒	• X 🗘 @ 🔆 🖅
ome > STM32H7	755XIHx 🔰 Untitled - Tools	\rightarrow				GENE	RATE CODE
Pino	ut & Configuration	Cl	ock Configuration		Project Manager		Tools
	Code Generation Configuration	~					
	Apply Application Regions	Settings to Peripherals: ON	Research	by ARM Cortex-M7 Mem	ory viewed by ARM Cortex-M4 Conte	ext CortexM7 + Context	CortexM4 +
mpare Projects	Apply Application Regions	Settings to Linker Files: OFF 🌘	0xD4000000	ank2 (64 MB)		N/A	
inpare i rojecta	Search an Application Region	>	0xD0000000 FMC SDRAM B		N/A	N/A	
	Show selected Application Region	. v	0xA0000000 Reserved			a	Display Se
			0x9000000 QuadSPI (256 N	AB)	N/A	N/A	Hide R
	RAM_D1	Ŭ	0x80000000 FMC Bank3 (25	6 MB)	N/A	N/A	□ Show
	Name RAM_D		0x7000000 Reserved			â	â
Memory Management		ortex-M7 ~	0x60000000 FMC Bank1 (25)	6 MB)	N/A	N/A	
munugement	Start address 0x2400		0x38801000 Reserved			â	â
	Size 512 Default Data Region	KB ~	0x38800000 Backup SRAM (4 KB)		A	A
		and additional features of the	0130010000	nain 4 (64 KR)		•	
	Code execution Permitte	any privilege level ~	0x38000000 Reserved			A	A
	Shareability Non-Sha		0x30048000 0x30040000 AHB SRAM3 (32	2 KB)			
PCC		ack Read Write Allocate for ~	0x30020000 AHB SRAM2 (12	28 KB)			Informatio
	Context sharing(M7,M4)		0x30000000 AHB SRAM1 (12	28 KB)			Legend
	3,,		0x24080000 Reserved			â	P P

Figure 408. Activate Memory Management support

The default application regions are in exclusive mode (context sharing is unselected). A reserved region in the other context is created and mentioned as "Mx non-shared region".



32 T	File	Window	Help				6	9 🚹 🕒	X () @ 🔆 🗲
me > STM32H755XI	Hx 🔰 Untitled - Pinou	t & Configuration >						GENERA	TE CODE
Pinout &	Configuration		Clock Configuration		Project Manage	er			Tools
			✓ Software Packs	✓ Pinout					
	~	0			NVIC1 Mode an	d Configuration			
tegories A->Z					Mo				
		~	Runtime contexts:						
lystem Core			Cortex-M7		Cortex	-M4			rDomain
÷	Cortex-M7	Cortex-M4							D1
BDMA		2			Configu	ration			
CORTEX_M4 CORTEX_M7	2	121	○ NVIC ② Code generation						
DMA	10		Priority Group 4 bits for pre-emption p	priority 0 bits for subpriority		Sort by I	Premption Priority	and Sub Priority	Sort by interrupts names
GPIO									
IWDG1	2		Search Search (Ctrl+F)			Show available	able interrupts \sim		Force DMA channels Interr
IWDG2		2	l						
MDMA				NVIC1 Interrupt Tab	ble		Enabled	Preempti	on Priority Sub Priorit
NVIC1	2		Non maskable interrupt					0	0
NVIC2			Hard fault interrupt					0	0
			Memory management fault					0	0
RCC	2		Pre-fetch fault, memory access fault					0	0
SYS M4		V	Undefined instruction or illegal state					0	0
SYS M7	2		System service call via SWI instruction					0	0
WWDG1	2		Debug monitor					0	0
WWDG2		2	Pendable request for system service Time base: System tick timer					0	0
111002			PVD and AVD interrupts through EXTI lin	10				0	0
			Flash global interrupt	10 10				0	0
nalog		· · · · · · · · · · · · · · · · · · ·	RCC global interrupt					0	0
ulaioy			CM4 send event interrupt for CM7					0	0
imers		>	FPU global interrupt					0	0
			HSEM1 global interrupt					6	0
Connectivity		>	Hold core interrupt					0	0
		>							
fultimedia									
tultimedia Necurity		>							
		>							

Figure 409. Default setting for new application region

4. Add a new region by pressing the "+" button that appears in the white space when hovering with the mouse.

Figure	410.	Adding	а	new	reaion
		/	~		

Pinout & C	Configuration	Clock Con	figuration	Project N	<i>l</i> lanager		Tools	
1	Memory view	ed by ARM Cortex-M7	Memory viewed by ARM Cortex-M4		Context CortexM7	+	Context CortexM4	+
ompare Projects	0xD4000000 Reserved			Add an applica	tion region	×		â
	0xD0000000 FMC SDRAM Ban	k2 (64 MB)		Name	new_app_reg		N/A	
	0xC0000000 FMC SDRAM Ban	k1 (256 MB)		Address	0x00000000		N/A	
	0xA0000000 Reserved			Size	Ов екв	OMB		â
	0x90000000 QuadSPI (256 MB)		Context			N/A	
	0x80000000 FMC Bank3 (256 I	//B)		Context sharin	g 🔲 (M7,M4)		N/A	
	0x70000000 Reserved							â
	0x60000000 FMC Bank1 (256 I	AB)			Cancel	Add	N/A	
	0x38801000 Reserved					â		â
PCC	0x38800000 Backup SRAM (4	<b)< td=""><td></td><td></td><td></td><td></td><td></td><td></td></b)<>						
	0x38010000 Reserved					â		â
	0x38000000 AHB SRAM Doma	A ICA KON					1	

5. Select "Context sharing (M7, M4)", automatically another region is created with the same name, start address, and size.

30048000Reserved		â	6
30040000 AHB SRAM3 (32 KB)			
30020000 AHB SRAM2 (128 KB)	AppReg0	AppReg0	

- 6. Select the Project Manager tab.
- 7. Give a name to the project and press the Generate Code button.



- 8. OPENAMP activation
 - Configure the NVIC1 and 2 and select their related HSEM global interrupts.
 - Activate the Middleware OPENAMP_M4.
 - MMT creates two application regions for each core. The Master regions are defined by attribute mode.

Figure 411. Configure NVIC1 and NVIC2, and select their HSEM global interrupt

STM32CubeMX Untitled*: STM32H7	55XIHx						- 6
M32	File	Window	Help			🕸 📑 🕒	X () @ 🔆 🖅
iome 🔪 STM32H755XIH	x 🔰 Untitled - Pinou	It & Configuration >				GENER	ATE CODE
Pinout & C	Configuration		Clock Configuration	on .	Project Manager		Tools
1	~	0	3		NVIC1 Mode and Configuration		
Categories A->Z					Mode		
System Core		~	Runtime contexts:	Cortex-M7	Cortex-M4	Pow	erDomain
÷	Cortex-M7	Cortex-M4		2			D1
BDMA					Configuration		
CORTEX_M4			NVIC Scode get	heration			
CORTEX_M7	×	/	Drinsity Comun. A late for	pre-emption priority 0 bits for subpriority	✓ Sert b:	v Premption Priority and Sub Priority	Sert by interrupts names
DMA GPIO		/	Priority Group 4 bits for	pre-emption priority o bits for subpriving		y Premption Priority and Gao Priority	Soft by interrupts names
IWDG1	2		Search Search (0	Strl+F)	Show average	ailable interrupts 🗸	Force DMA channels Interrupts
IWDG2	La	2					
MDMA				NVIC1 Interrupt	able	Enabled Preempt	ion Priority Sub Priority
NVIC1	×		Non maskable interrupt			0	0
NVIC2		V	Hard fault interrupt			0	0
		_	Memory management fault			0	0
RCC			Pre-fetch fault, memory ac			0	0
✓ SYS_M4		2	Undefined instruction or ille			0	0
✓ SYS_M7	×		System service call via SV Debug monitor	/l instruction		0	0
WWDG1			Pendable request for syste	am sanúsa			0
WWDG2		2	Time base: System tick tin			15	0
			PVD and AVD interrupts th			0	0
		1	Flash global interrupt			0	0
Analog		>	RCC global interrupt			0	0
			CM4 send event interrupt f	for CM7		•	0
Timers		>	FPU global interrupt			0	0
		>	HSEM1 global interrupt				0
Connectivity			Hold core interrupt			L V	0
Multimedia		>					
Security		>					
Computing		>					
					Enabled Preemption Priority	y V Sub Priority V	

Figure 412. OPENAMP_M7 parameters settings

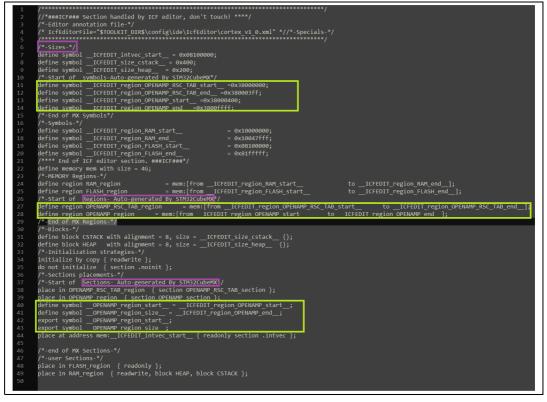
		ý (ý	OPEN	AMP_M7 Mode and Configuration		Pinout view System view
ategories A->Z				Mode	la de la dela del	
Middleware and Software Packs		~	Runtime contexts:			
			Cortex-M7	Cortex-M4	PowerDomain	
	Cortex-M7	Cortex-M4	5		D1	
AIROC-WI-FI-Bluetooth-STM32		2	Z Enabled			
FATES_M4		24				
FATFS_M7	5					
FP-SNS-MOTENVWB1						
FP-SNS-SMARTAG2						The second s
FREERTOS_M4		2				
FREERTOS_M7	5	M				
LCUBE-CANOPEN	58			Configuration		
I-CUBE-Cealum						
I-CUBE-FS-RTOS			Reset Configuration			termine (and (and (and (and)
I-CUBE-ITTIADB			Parameter Settings Subser Constants			මර්මමර් ලෙලෙල මම්මලිසි
I-CUBE-Mongoose			Configure the below parameters :			
+CUBE-embOS						
I-CUBE-wolfMQTT			Q Search (Ctrl+F) ③ ④		0	병원병원 보관했었는 전망산병원
I-CUBE-wolfSSH			~			
I-CUBE-wolfSSL			Instance	OPENAMP_M7		
I-CUBE-wolfTPM I-Cube-SoM-uGOAL			✓ Version			
LIBJPEG			OPENAMP version	v2018.10		
			Communication Mode			
LWIP			Mode	MASTER		
MBEDTLS			 Configuration 			TFBGA240 +25 (Top view)
OPENAMP_M4		2	METAL_MAX_DEVICE_REGIONS	2		1 S O V
OPENAMP_M7			RPMSG_BUFFER_SIZE	512		
PDM2PCM_M4		22	NUM_RESOURCE_ENTRIES	2		
	22		VRING_COUNT	2		
		5	VDEV_ID	0xFF		
	2		VRING0_ID	0		
		2				
	2					
X-CUBE-AI	-4					Q [] Q 🕒 🚄
A YOURS ALGORIND		-				





Figure 413. OPENAMP_M4 parameters settings

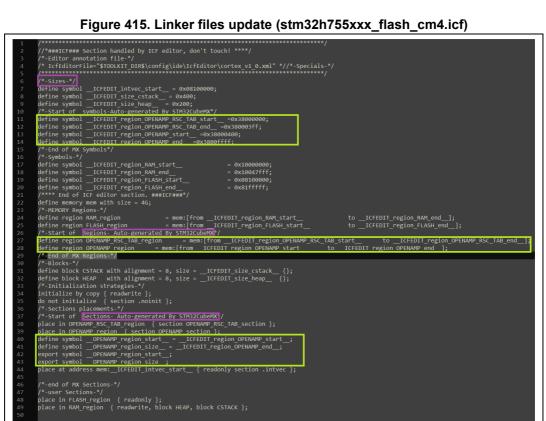


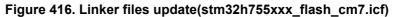


9. Press the Generate Code button to generate the code for both applications.

Apply Application Regions settings to linker files

Apply Application Regions Settings to Peripherals: ON (1) Apply Application Regions Settings to Linker Files: ON (1) When the second radio button is on, the linker scripts for the CM7 and CM4 projects are generated considering the configuration.









The middleware can be enabled or disabled:

- If disabled, it automatically chooses the configured memory along with the associated driver and sets the execution memory location in the linker file.
- If enabled, the two regions and corresponding 'export symbol' must be added in the generated linker file.

After the code generation, navigate to the generated folder to check the linker file updates.

Example of MMT configuration of the ETH on STM32H755XIH6TR MCU

1. Activate the IP ETH: MMT creates three application regions for each core. To change the start address and the size of each region, update the ETH parameters.

STM32CubeMX Untitled*:	STM32H755XIH	ĸ												- a x
STM32		File	Window	Help							19	- F -		Q ★ / 77
CubeMX											0	_		~~-//
Home > STM32H	4755XIHx 🗡	Untitled - P	nout & Configuration >									GENE	RATE CODE	
Pin	out & Conf	iguration	C	lock Configuration			Project	Manager					Tools	
				✓ Software Packs	✓ Pinout									
Q			ETH	HMode and Configuration	1				Q	Pinout view	💾 System	n view		
Categories A->Z				Mode	i i i i i i i i i i i i i i i i i i i									8
Connectivity		Ý	Runtime contexts:											1
^	Cortex-M7	Cortex-M4	Cortex-M7		verDomain		_							
S ETH					D2		0		(m) (m) (m	u) (m) (m) (u				
FDCAN1			Mode MII		~		2	e a a	m m (a land from the		m (m) (
FDCAN2			Activate Rx Err signal											
FMC			Activate Tx Err signal									e e		1 1
1201								5) (a) (a) (a)	(m) (m) (n			(m) (m) (ne) (ne) (ne) (ne)	
12C2							6		(m) (m) (w	n) (m,n) (mm) (*	01 (N.W. (VII)	m (m) (
/2C3				Configuration			2	ăăă.					čěčč	
12C4				Configuration										1 1
LPUART1			Reset Configuration				8			(m)(m)((***) (1 1
MDIOS			User Constants	NVIC Settings	GPIO Settings		6	na) (na) (na) ((m) (m)	(m)(m)((m) (m)	(10)		
QUADSPI				Parameter Settings			6			- AA	a a a	(m)		
SDMMC1 SDMMC2			Configure the below parameters :											
SDMMC2 SPI1			Q Search (Ctrl+F)		0				(m) (m)	(m)(m)(()		1 1
SPI2			General : Ethernet Configuration	n			6	ne) (ne) (ne) (m) (m) -	(m)(m)(a)(w)(w)	(10)		
SP/3			Warning	The ETH can work only	when RAM is poin		6					()		
SPI4			Ethernet MAC Address				2							
SPI5			Tx Descriptor Length	4			2							
SPI6			First Tx Descriptor Adde Rx Descriptor Length	ress 0x30000080 4			0							
SWPMI1			First Rx Descriptor Length				6		-				na) (m) (m) (nu)	
UART4			Rx Buffers Address	0x30000100			2		0.04					
UART5			Rx Buffers Length	1524										
UART7												(mm) (mm) (
UART8									1	FBGA240 +	25 (Top vie	w)		
USART1											(,		
USART2														
USART3											_			
USART6						Q	53	Q		4 0		Q		√ Uni
USB OTG FS													_	

Figure 417. Configuration of ETH IP

Figure 418. ETH MMT regions

x30020000 AHB SRAM2 (128 KB)		
	RxPool	RxPool
AHB SRAM1 (128 KB)	TxDescriptor	TxDescriptor
x30000000	RxDescriptor	RxDescriptor
A soppoor Reserved		a :

- 2. Press the radio button "Apply Application region Settings to Peripherals ON", ETH will be partially under MMT control.
- 3. Press the Generate Code button to generate code for both applications.



		Software Packs
ETH N	Mode and Configu	ration
	Mode	
Runtime contexts:		
Cortex-M7	Cortex-M4	PowerDomain
		D2
Mode MII		~
Activate Rx Err signal		
Activate Tx Err signal		
	Configuration	
	Configuration	
Reset Configuration	Configuration	
Reset Configuration	Configuration	GPIO Settings
		 GPIO Settings User Constants
 NVIC Settings Parameter Setting 	15	
 NVIC Settings Parameter Setting onfigure the below parameters 	15	
 NVIC Settings Parameter Setting onfigure the below parameters 	15	User Constants
 NVIC Settings Parameter Setting onfigure the below parameters Search (Ctrl+F) 	s: 3	 User Constants
 NVIC Settings Parameter Setting onfigure the below parameters Search (Ctrl+F) General : Ethernet Configure 	ration The ETH	 User Constants
 NVIC Settings Parameter Setting onfigure the below parameters Search (Ctrl+F) General : Ethernet Configur Warning 	ration The ETH ess 00:80:E1	 User Constants Can work only when RAM i
 NVIC Settings Parameter Setting onfigure the below parameters Search (Ctrl+F) General : Ethernet Configur Warning Ethernet MAC Addres 	ration The ETH ess 00:80:E1 h 4	Can work only when RAM i
 NVIC Settings Parameter Setting onfigure the below parameters Search (Ctrl+F) General : Ethernet Configur Warning Ethernet MAC Addred Tx Descriptor Lengt 	ration The ETH ess 00:80:E1 h 4 Address 0x30000	Can work only when RAM i
 NVIC Settings Parameter Setting onfigure the below parameters Search (Ctrl+F) General : Ethernet Configure Warning Ethernet MAC Addred Tx Descriptor Length First Tx Descriptor A 	s: Tation The ETH ess 00:80:E1 h 4 Address 0x30000 h 4	User Constants
 NVIC Settings Parameter Setting onfigure the below parameters Search (Ctrl+F) General : Ethernet Configure Warning Ethernet MAC Addred Tx Descriptor Length First Tx Descriptor Length 	s: The ETH ess 00:80:E1 h 4 Address 0x30000 ch 4 Address 0x30000	User Constants

Figure 419. IP configuration

Apply Application Regions settings to linker files:

- 4. When this button is on, the linker scripts for the CM7 project and CM4 project are generated, considering the configuration.
- 5. After the code generation, navigate to the generated folder:
 - Under the CM7 Project, open the linker definition file.
 - Under the Memories definition you can see the defined memories with their start address and length, according to the configuration made in STM32CubeMX.



<pre>/*Size-*/ drive yubd:CFEDIT_inte_text = 0.0000; drive yubd:LFEDIT_inte_text = 0.00000; drive yubd:LFEDIT_inte_text = 0.000000; drive yubd:LFEDIT_inte_text</pre>	
<pre>/**** / /*****************************</pre>	<pre>9 define symbolICFEDIT_intvec_start = 0x08000000; 10 11 define symbolICFEDIT_size_cstack = 0x400;</pre>
<pre>define symbolCFRDIT_region_Reduction_start = 0:2000000; define symbolCFRDIT_region_Reduction_end = 0:20000000; define region Index[region</pre>	<pre>13 14 14 14 14 15 /*-Start of symbols-Auto-generated By STM32CubeMX*/ 16 define symbolICFEDIT_region_ITCMRAM_start = 0x00000000; 17 define symbolICFEDIT_region_RAU_DI = 0xFFFF; 18 define symbolICFEDIT_region_RAU_DI = 0xF40000000; 17 18 19 10 10 10 10 10 10 10 10 10 10 10 10 10</pre>
<pre>/*-End of RX Symbols*/ /*-Symbols*/ /*-Symbols*/ /*-Symbols*/ define symbolCCFDIT_region_RAM_start = 0x20000000; define symbolCCFDIT_region_Adm_ICFBIT*/ define segion RAM_region = mem:[fromLCFEDIT_region_RAM_start toLCFEDIT_region_FLASH_termd]; define region RAM_region</pre>	20 define symbolICFEDIT_region_RxDescripSection_start = 0x30000007; 21 define symbolICFEDIT_region_RxDescripSection_end = 0x30000007; 22 define symbolICFEDIT_region_RxDescripSection_start 0x30000000; 23 define symbolICFEDIT_region_RxDescripSection_end = 0x30000000; 24 define symbolICFEDIT_region_RxDescripSection_start 0x300000100; 25 define symbolICFEDIT_region_Rx_PoolSection_start 0x300000100; 25 define symbolICFEDIT_region_Rx=PoolSection_end 0x300000000;
<pre>/*.Symbol:*/ define symbolCFEDIT_region_RAM_end</pre>	
<pre>define memory mem with size = 46; /*-MEMORY Regions-*/ define region RAM_region = mem:[from _ICFEDIT_region_RAM_start to _ICFEDIT_region_RAM_end_]; define region RLASH_region = mem:[from _ICFEDIT_region_ICMRAM_start to _ICFEDIT_region_RIASH_end_]; /*-Start of <u>Regions_Auto_semenated Bv_STM32CubeXX//</u> define region RAMD DI region = mem:[from _ICFEDIT_region_ITCMRAM_start to _ICFEDIT_region_RAM_end_]; define region RAMD DI region = mem:[from _ICFEDIT_region_RAMD start to _ICFEDIT_region_RAMD DI end_]; define region RAMD DI region = mem:[from _ICFEDIT_region_RAMD start to _ICFEDIT_region_RAMD DI end_]; define region RAMD DI region = mem:[from _ICFEDIT_region_RAMD start to _ICFEDIT_region_RAMD DI end_]; define region RAMD StartScience.to _ mem:[from _ICFEDIT_region_RAMD start to _ICFEDIT_region_RAMD ScripSection_end_]; define region RAMD StartScience.to _ mem:[from _ICFEDIT_region_RAMD StartScience.to _ ICFEDIT_region_RAMD ScripSection_end_]; define region RAMD startscience.to _ ICFEDIT_region_RAMD ScripSection_start to _ICFEDIT_region_RAMD ScripSection_end_]; define block HAMD with alignment = 8, size = _ICFEDIT_size_cstack(); define block HAMD with alignment = 8, size = _ICFEDIT_size_cstack(); define block HAMD with alignment = 8, size = _ICFEDIT_size_cstack(); define block HAMD with alignment = 8, size = _ICFEDIT_size_cstack(); define block HAMD with alignment = 8, size = _ICFEDIT_size_cstack(); define block HAMD with alignment = 8, size = _ICFEDIT_size_cstack(); define block HAMD with alignment = 8, size = _ICFEDIT_size_cstack(); define block HAMD with alignment = 8, size = _ICFEDIT_size_cstack(); define block HAMD with alignment = 8, size = _ICFEDIT_size_cstack(); define block HAMD with alignment = 8, size = _ICFEDIT_size_cstack(); define block HAMD with alignment = 8, size = _ICFEDIT_size_cstack(); define block HAMD with alignment = 8, size = _ICFEDIT_size_cstack(); define block HAMD with alignment = 8, size = _ICFEDIT_size_cstack(); define block HAMD with alignm</pre>	29 /*.Symbols-*/ 30 define symbolICFEDIT_region_RAM_start
<pre>/*-HENGKY Regions-*/ define region RAW_region = mem:[from _ICFEDIT_region_RAM_startto _ICFEDIT_region_RAM_end_]; define region FLASH_region = mem:[from _ICFEDIT_region_FLASH_startto _ICFEDIT_region_TLASH_end_]; /*-start of <u>Regions-Auto-generated By STM32CubeRXY/</u> define region RAM D1 region = mem:[from _ICFEDIT_region_RAM_startto _ICFEDIT_region_TLCMRAM_end_]; define region RAM D1 region = mem:[from _ICFEDIT_region_RAM_startto _ICFEDIT_region_RAM D1 end]; define region RAM D1 region = mem:[from _ICFEDIT_region_RADescripSection_startto _ICFEDIT_region_TXDescripSection_end_]; define region RAMD2scripSection_region = mem:[from _ICFEDIT_region_RADescripSection_startto _ICFEDIT_region_TXDescripSection_end_]; define region RADescripSection_region = mem:[from _ICFEDIT_region_RADescripSection_startto _ICFEDIT_region_RADescripSection_end_]; define holock CSTACK with alignment = 8, size = _ICFEDIT_size_cstack(); define holock CSTACK with alignment = 8, size = _ICFEDIT_size_heap{;;} /*-Initialization strategies-*/ initialize k cection .noinit }; /*-Sections placements-*/ //*-Start of <u>Sections Auto-generated By STM32CubeXX*/</u></pre>	36 define memory mem with size = 4G;
<pre>/*.start of <u>Regions: Auto-generated By STM32CubeRX*/</u> define region ITCHRMM_region = mem:[from ITCHDIT_region_ITCHRAM_starttoICFEDIT_region_BAM_D1 and _]; define region RAM_D1 region = mem:[from ITCHDIT_region_RAMescripSection_starttoICFEDIT_region_RAMescripSection_end_]; define region RAM_D2 region = mem:[fromICFEDIT_region_RAMescripSection_starttoICFEDIT_region_RAMescripSection_end_]; define region RAM_D2 region = mem:[fromICFEDIT_region_RAMescripSection_starttoICFEDIT_region_RAMescripSection_end_]; define region RAM_D2 region = mem:[fromICFEDIT_region_RAMescripSection_starttoICFEDIT_region_RAMescripSection_end_]; define region RAM_D2 region = mem:[fromICFEDIT_region_RAM_PoslSection_starttoICFEDIT_region_RAM_PoslSection_end_]; define region RAM_D2 region = mem:[fromICFEDIT_region_RAM_PoslSection_starttoICFEDIT_region_RAM_PoslSection_end_]; define hock KEAP with alignment = 8, size =ICFEDIT_size_cstack(); define block KEAP with alignment = 8, size =ICFEDIT_size_meap{;;} /*-Initialization strategies-*/ do not initialize (section .noinit); /*-Sections placements-*/ // *-Start of <u>Sections-Auto-generated By SIN32CubeRX*/</u></pre>	38 /*-MEMORY Regions-*/ 39 define region RAM_region = mem:[fromICFEDIT_region_RAM_start toICFEDIT_region_RAM_end_]; 40 define region FLASH_region = mem:[fromICFEDIT_region_FLASH_start toICFEDIT_region_FLASH_end_];
<pre>4 /*-End of MX Regions-*/ 5 5 /*-Blocks-*/ 5 define block HEAP with alignment = 8, size =ICFEDIT_size_cstack (); 5 define block HEAP with alignment = 8, size =ICFEDIT_size_heap (); 5 do not initialize by copy { readwrite }; 5 do not initialize { section .noinit }; 5 /*-Sections placements-*/ 6 /*-Start of Sections-Auto-generated By SIM32CubeMX*/ 6 place in ARMOregion { section ITCMRAM_section }; 6 place in ITCMRAM_region { section ITCMRAM_section }; 7 place in RxDescripSection_region { section SxDescripSection_section }; 7 place in RxDescripSection_region { section RxDescripSection_section }; 7 place in RxDescripSection_region { section RxDescripSection_region }; 7 place in RxDescripSection_region { section RxDescripSection_region }; 7 place in RxDescripSection_region { section R</pre>	42 /*.Start of Regions: Auto-generated BV_STM32CubeVx*/ 43 define region ITCMRAM_region = mem:[fromICFEDIT_region_ITCMRAM_starttoICFEDIT_region_ITCMRAM_starttoICFEDIT_region_ITCMRAM_end_]; 44 define region RXDescripSectionemen:[fromICFEDIT_region_RXDescripSection_starttoICFEDIT_region_RXDescripSection_end_]; 45 define region RXDescripSection region = mem:[fromICFEDIT_region_XDescripSection_starttoICFEDIT_region_RXDescripSection_end_]; 46 define region RXDescripSection region = mem:[fromICFEDIT_region TXDescripSection_starttoICFEDIT_region_RXDescripSection_end_];
<pre>51 /*-Blocks-*/ 52 define block CSTACK with alignment = 8, size = _ICFEDII_size_cstack (); 53 define block HEAP with alignment = 8, size = _ICFEDII_size_heap {}; 54 /*-Initialization strategies-*/ 55 /*-Initialize by copy { readwrite }; 56 do not initialize { section .nolnit }; 57 do not initialize { section .nolnit }; 58 /*-Sections placements-*/ 60 /*-Start of <u>Sections-Auto-generated By STM32CubeMX*/</u> 61 /*-Start of <u>Sections-Auto-generated By STM32CubeMX*/</u> 62 place in tAMP-region { section ITCMRAM_section }; 63 place in ITCMRAM_region { section ITCMRAM_section }; 64 place in RxDescripSection_region { section ITXMRAM_section }; 65 place in RxDescripSection_region { section RxDescripSection_section }; 66 place in RxDescripSection_region { section RxDescripSection_section }; 71 /*-end of MX Sections-*/ 72 /*-user Sections-*/ 73 place in fXR-region { readonly }; 74 place in fXR-region { readonly }; 75 place in fXR-region { readonly }; 74 place in fXR-region { readonly }; 75 place in fXR-region { readonly }; 74 place in fXR-region { readonly }; 75 place in fXR-region { readonly }; 74 place in fXR-region { readonly }; 75 place in fXR-region { readonly }; 75 place in fXR-region { readonly }; 76 place in fXR-region { readonly }; 76 place in fXR-region { readonly }; 76 place in fXR-region { readonly }; 77 place in fXR-region { readonly }; 78 place in fXR-region { readonly }; 79 place in fXR-region { readonly }; 70 place in fXR-region { readonly }; 71 place in fXR-region { readonly }; 72 place in fXR-region { readonly }; 73 place in fXR-region { readonly }; 74 place in fXR-region { readonly }; 74 place in fXR-region { readonly }; 75 place in fX</pre>	
<pre>55 /*.Initialize ycopy { readonite); 56 initialize y copy { readonite); 57 do not initialize { section .noinit }; 58 /*.Sections placements-*/ 60 /*.Start of <u>Sections Auto-generated By STM32CubeMX*/</u> 61 /*.Start of <u>Sections Auto-generated By STM32CubeMX*/</u> 62 place in Address memICFEDIT_intvec_start_ { readonly section .intvec }; 63 64 place in ITCMRAM_region { section ITCMRAM_section }; 65 place in RxDescripSection_region { section RxDescripSection_section }; 66 place in RxDescripSection_region { section RxDescripSection_section }; 67 place in RxDescripSection_region { section RxDescripSection_section }; 68 place in RxDescripSection_region { section RxDescripSection_section }; 69 /*-end of MX Sections-*/ 71 /*-user Sections-*/ 72 place in rKB-region { readonly }; 73 place in FXMS-region { readonly }; 74 place in FXMS-region { readonly }; 75 place in FXMS-region { readonly }; 76 place in FXMS-region { readonly }; 77 place in FXMS-region { readonly }; 78 place in FXMS-region { readonly }; 79 place in FXMS-region { readonly }; 70 place in FXMS-region { readonly }; 71 place in FXMS-region { readonly }; 72 place in FXMS-region { readonly }; 73 place in FXMS-region { readonly }; 74 place in FXMS-region { readonly }; 75 place in FXMS-region { readonly ; 75 place in</pre>	<pre>51 /*-Blocks.*/ 52 define block CSTACK with alignment = 8, size =ICFEDIT_size_cstack {}; 53 define block HEAP with alignment = 8, size =ICFEDIT_size_heap {};</pre>
<pre>/*-Sections placements-*/ /*-Start of Sections-Auto-generated By_STM32CubeMX*/ /*-Start of Sections-Auto-generated By_STM32CubeMX*/ /* place at address mem:_ICFEDIT_intvec_start_ (readonly section .intvec); /* place in nto-main faction TCMRAM_section ; place in nto-cripsection_region { section RAM D1 section }; place in nto-cripsection_region { section RAM-section ;; place in nto-cripsection_region { section RAM-section ;; place in nto-cripsection_region { section section section }; place in RX_PoolSection_region { section ntx_PoolSection_section }; /*-end of MX_Sections-*/ /*-user Sections-*/ /*-user Sections-*/</pre>	<pre>55 /*-Initialization strategies-*/ 56 initialize by copy { readwrite }; 57 do not initialize { section .noinit };</pre>
<pre>61 /*-start of Sections-Auto-generated BV_STM32CubeMXY/ 62 place at address mem:_ICFEDIT_intvec_start (readonly section .intvec); 63 place in nTCMRAM-region { section TCMRAM_section }; 64 place in RAM D1 region { section RAM D1 section }; 65 place in RAM D1 region { section RAM D1 section }; 66 place in RAMscripsection_region { section RAMescripsection_section }; 67 place in RAMscripsection_region { section RAMescripsection_section }; 78 place in RAM PoolSection_region { section RAMescripsection_section }; 79 place in RAM Sections-*/ 70 /*-end of MX Sections-*/ 71 /*-user Sections-*/ 72 /*-user Sections-*/ 73 place in FLASH_region { readonly }; 74 place in FLASH_region { readonly }; 75 place in FLASH_region { readonly }; 76 place in FLASH_region { readonly }; 77 place in FLASH_region { readonly }; 78 place in FLASH_region { readonly }; 79 place in FLASH_region { readonly }; 70 place in FLASH_region { readonly }; 71 place in FLASH_region { readonly }; 72 place in FLASH_region { readonly }; 73 place in FLASH_region { readonly }; 74 place in FLASH_region { readonly }; 75 place in FLASH_region { readonly }; 76 place in FLASH_region { readonly }; 77 place in FLASH_region { readonly }; 78 place in FLASH_region { readonly }; 79 place in FLASH_region { readonly }; 70 place in FLASH_region { readonly }; 70 place in FLASH_region { readonly }; 71 place in FLASH_region { readonly }; 72 place in FLASH_region { readonly }; 73 place in FLASH_region { readonly }; 74 place in FLASH_region { readonly }; 74 place in FLASH_region { readonly }; 75 place in FLASH_region { readonly ; 75 place in FLASH_region { read</pre>	59 /*-Sections placements-*/
<pre>64 place in ITCMRAM_region { section ITCMRAM_section }; 65 place in RAM D1 region { section RAM D1 section }; 66 place in RXDescripSection_region { section RXDescripSection_section }; 67 place in RXDescripSection_region { section TXDescripSection_section }; 68 place in RXPoolSection_region { section RX_PoolSection_section }; 69 /*-end of MX Sections-*/ 71 /*-user Sections-*/ 73 place in FLASH_region { readonly }; 74 place in FLASH_region { readonly };</pre>	<pre>61 /*-Start of Sections- Auto-generated By STM32CubeMX*/ 62 place at address mem:_ICFEDIT_intvec_start_ (readonly section .intvec);</pre>
<pre>70 /*-end of MX Sections-*/ 71 72 /*-user Sections-*/ 73 place in FLASM_region { readonly };</pre>	 place in ITCMRAM_region { section ITCMRAM_section }; place in RAM DI region { section RAM DI section }; place in RxDescripSection_region { section RxDescripSection_section }; place in RxDescripSection_region { section RxDescripSection }; place in RxDeolSection_region { section RxDescripSection }; place in RxDeolSection_region { section RxDescripSection };
<pre>72 /*-user Sections-*/ 73 place in FLASH_region { readonly };</pre>	70 /*-end of MX Sections-*/
	<pre>72 /*-user Sections-*/ 73 place in FLASH_region { readonly };</pre>

Figure 420. Defined memories under the linker file (Cortex-M7)



/*- <u>Sizes-</u> */ define symbolICFEDIT_intvec_start_ = 0x08100000;
define symbolICFEDIT_size_cstack = 0x400; define symbolICFEDIT_size_heap = 0x200;
/*-Start of symbols-Auto-generated By SIM32CubeMX*/ define symbolICFEDIT_region_RxDescripSection_start = 0x30000000; define symbolICFEDIT_region_TxDescripSection_end_ = 0x3000007F; define symbolICFEDIT_region_TxDescripSection_end_ = 0x300000FF; define symbolICFEDIT_region_Rx_PoolSection_end_ = 0x300000100; define symbolICFEDIT_region_Rx_PoolSection_end_ = 0x300000FF; define symbolICFEDIT_region_Rx_PoolSection_end_ = 0x300000FF;
/*-End of MX Symbols*/
/*-Symbols-*/ define symbolICFEDIT_region_RAM_start = 0x100000000; define symbolICFEDIT_region_RAM_end = 0x10047FFF; define symbolICFEDIT_region_FLASH_start = 0x08100000; define symbolICFEDIT_region_FLASH_end = 0x81FFFFF; /**** End of ICF editor section. ###ICF####/
define memory mem with size = 4G;
/*-MEMORY Regions-*/ define region RAM_region = mem:[from _ICFEDIT_region_RAM_startto _ICFEDIT_region_RAM_end]; define region FLASH_region = mem:[from _ICFEDIT_region_FLASH_startto _ICFEDIT_region_FLASH_end];
/*-Start of Regions- Auto-generated By STN32CubeMX*/ define region TxDescripSection_region = mem:[fromICFEDIT_region_RxDescripSection_start toICFEDIT_region_RxDescripSection_end define region TxDescripSection_region = mem:[fromICFEDIT_region_TxDescripSection_start toICFEDIT_region_TxDescripSection_end
/*-End of MX Regions-*/
/*-Blocks-*/ define block CSTACK with alignment = 8, size =ICFEDIT_size_cstack {}; define block HEAP with alignment = 8, size =ICFEDIT_size_heap {};
/*-Initialization strategies-*/ initialize by copy { readwrite }; do not initialize { section .noinit };
/*-Sections placements-*/
/*-Start of Sections- Auto-generated By STM32CubeMX*/ place at address mem:ICFEDI1_intvec_start { readonly section .intvec };
<pre>place in RxDescripSection_region { section RxDescripSection_section }; place in TxDescripSection_region { section TxDescripSection_section }; place in Rx_PoolSection_region { section Rx_PoolSection_section };</pre>
/*-end of MX Sections-*/
/*-user Sections-*/ place in FLASH_region { readonly }; place in RAM_region { readwrite, block HEAP, block CSTACK };

Figure 421. Defined memories under the linker file (Cortex-M4)

5.5.6 STM32H7RS

Feature: MMT usage, pinout, and configuration user interface

When the first toggle button is ON, Cortex-M7_BOOT (MPU) and Cortex-M7_APPLI (MPU) are under MMT control: their modes and parameters become read-only.

Apply Application Regions Settings to Peripherals: ON Apply Application Regions Settings to Linker Files: OFF





n

A

0

6

6

0

M32 ubeMX		File		Window	Help	A 🕸		× C	27	54/
ome 🔪 STM32H7F	RSA8Ix	> Untitle	d - Pinout &	Configuration >			GENER	ATE CODE		
Pinout & Co	nfigurat	tion	C	Clock Configuration	Project I	Manager		Т	ools	
			✓ Sc	ftware Packs	✓ Pinout					
()				CORTEX_M	7_APPLI Mode and Configu	ration	C F	inout view	Sy	stem view
ategories A->Z					Mode		ľ			
System Core			~	Runtime contexts:						
System Core					Configuration		1			
\$	Boot		ExtMemL	Reset Configuration						
CORTEX_M7_AP										
CORTEX_M7_B	\checkmark			Parameter Settings OU	ser Constants					
FLASH				Configure the below parameters			00	0000		
GPDMA1	\checkmark	\checkmark		Q Search (Ctrl+F)	0	0	00			
GPIO				✓ Speculation default mode Set	tings		22			
HPDMA1	\checkmark	\checkmark		Speculation default mo			and and	čěčě		
IWDG				✓ Cortex Interface Settings			00	õõõõ		Géé
NVIC_APPLI		\checkmark		CPU ICache	Disabled		00			
NVIC_BOOT	\checkmark			CPU DCache	Disabled					
RAMECC				Cortex Memory Protection Ur	nit Cont					
RCC	\checkmark	\checkmark		MPU Control Mode	Background Regio	n Privileged accesse	000			
SYS				Cortex Memory Protection Ur	nit Regi					
WWDG				MPU Region	Enabled		ěě	ěěěě		
				MPU Region Base Ad	dress 0x0			UFBGA16	69 (Top vie	w)
				MPU Region Size	4GB					,
Analog			>	MPU SubRegion Disal						
				MPU TEX field level	level 0					
Timers			>	MPU Access Permissi		T PERMITTED				
				MPU Instruction Acces						
Connectivity			>	MPU Shareability Perr						
				MPU Cacheable Perm MPU Bufferable Perm			•	53	e	12

Figure 422. MMT usage

Feature: MMT usage and linker script

Apply Application Regions Settings to Peripherals: ON Apply Application Regions Settings to Linker Files: ON

Linker files content is generated according to the configuration of application regions.



Apply Application Regions Settings to Peripherals: ON



Apply Application Regions Settings to Linker Files: OFF

Apply Application Regions Settings to Peripherals: ON Apply Application Regions Settings to Linker Files: ON

Only "Boot" and "Appli" contexts are managed by the MMT. Each context has its own application region (AppReg0 and AppReg1, respectively).



User interface

Home > ST	W2CU535CBTx 🔰 Untitled - Teols 🔪 LPBAM Scenario & Corf	uation >	GENERATE CODE	CHECK LPBAM DESIGN
	Pinout & Configuration	Clock Configuration	Project Manager	Tools
	Colo Generation Canlignation > One Apply registrices Registers stretting to benefamilia. CPF 0 Apply Application Registers and the first content fills. CPF 0 Statistical and Application Registers > Devention and Application Registers >	Nerroy Need () ADD Cafes ADD () ADD Cafes ADD () ADD Cafes ADD () ADD Cafes ADD () ADD	Agérdes Papes	+ •
PCC		Cartering Road (2004 (2015)) Cartering Road (2004 (2015)) Special Road (2014) Cartering Road (2014)	Avr / 10 Promot Aus Paper 400 (10 Promot Aus Paper 400 (1)	Diplay Setting - Four Vow
CAD			Sker (5) Sker (5) Skar (5) Roment Alsa Roper Sker (5) Roment Alsa Roper Skar (5) Roment Alsa Roper	
		exception Reserved Paulo Back memory (D) (CB KB) exception Reserved	AUSH (K) Annexe Alas Tago AUSH (K) Annexe Alas Tago AUSH (K) AUSH RASH (S)	A A A A A A A A A A A A A A A A A A A
		orilazosos SRAAZ Code (HS) (54 KB) orilazosos SRAAT Code (HS) (152 KB) orizazoso Resented	RAH (IC) Research Also Region RAH (IC) Research Also Region	Region allowing different types of security Secure region (5).
		Plank Bank1 memory (NS) (128 KB) bathstocoo	PLANH INS) PLANH INS: (S) Pleanner Alias Aligon PLASH (S) Pasemed Alias Region	Iton Secure region (NS) Iton Secure region (NS) Iton Secure Calable region (NSC).
		(entropy), Remod		Poplar accessibility because ad the Social Poplar accessibility because ad the Social Poplar addressibility of the Social

Figure	423.	Default	settings
--------	------	---------	----------

The middle panel represents the memory, split into three columns: the left one is the memory seen by the core(s), the middle one the memory set-up for the application in Context Boot, the right one the memory set-up for the application in the Context Appli.

For the new project created under STM32CubeMX, the tool creates the default application region to generate a valid project.

Region information

Clicking on a particular region in the Application Regions column shows the associated details on the left hand side.

STM32CubeMX automatically adds a 4-Gbyte region for the system core, even if you are not planning to use the MMT.



1. Choose a supported MCU (the following example is based on STM32H7R3A8I6).

lew Project from a MCU/MPU							
CU/MPU Selector Board Selector	Example Selector	Cross	Selector				
CU/MPU Filters							
★ 🔂 🛱 🗸	5		Features Block Diagram	Docs & Resources	CAD Resources	Datasheet [Buy 🕞 Start Project
Commercial Part Number	\sim	\$	STM32H7 Series	-			
۹ 📃 🗸	+-		STM32H7R3A8I6	High-performance and Flash memory, 620 KB external memory inter	SRAM, 550 MHz CPU	, L1 cache, grap	phic accelerations,
PRODUCT INFO	~			Unit Price for 10kU (US\$): NA		147	
Segment	>		COMING SOON Stay tuned !			UFBGA 169 7	x7x0.6 P 0.5 mm
Series	>]
Line	~		High-performance and DSP with				M, 550 MHz CPU, L1 cache,
Line	~		High-performance and DSP with graphic accelerations, external				M, 550 MHz CPU, L1 cache,
Line Check/Uncheck All (2)	~		graphic accelerations, external				M, 550 MHz CPU, L1 cache,
_							M, 550 MHz CPU, L1 cache,
	Aa (ab)		graphic accelerations, external				M, 550 MHz CPU, L1 cache,
_			graphic accelerations, external Features				M, 550 MHz CPU, L1 cache,
Check/Uncheck All (2)			graphic accelerations, external Features • N/A				
Check/Uncheck All (2)		мси	graphic accelerations, external Features				M, 550 MHz CPU, L1 cache,
Check/Uncheck All (2)		мси	graphic accelerations, external Features NVA Is/MPUs List: 34 items Commerce	memory interfaces, USB High S	Speed PHY and large set of	t peripherals	₫ Export bit (converters) 💥 Marketing Status
Chack/Uncheck All (2)		мси	graphic accelerations, external Features - NA Is/MPUs List: 34 items ☆ STM3247R33	memory interfaces, USB High S Interfaces Part No Ref STM32H7R3A8	Speed PHY and large set of Reference STM32H7R3A8ix	r peripherals	
Check/Uncheck All (2) CSTM32H7B0 Value line SSTM32H7B0 Value line SSTM32H7R3/7S3 SSTM32H7R3/7S7 SSTM32L0x0 Value Line		мси	graphic accelerations, external Features • N/A Is/MPUs List: 34 Rems	memory interfaces, USB High S Isl Part No. Part No. ABIG STM32H7R3A8 8K6 STM32H7R3A8	Reference STM32H7R3A8bx STM32H7R3A8bx STM32H7R3A8bx	r peripherals	Export Export Converters Coming soon Coming soon
Check/Uncheck All (2) Check/Uncheck All (2) STM32H7B0 Value line STM32H7R3/7S3 STM32H7R3/7S7 STM32L0x0 Value Line STM32L0x1 STM32L0x1		MCU	grāphic accelerations, external Features ■ NA Is/MPUs List: 34 items ☆ STM324/TR31 ☆ STM324/TR31 ☆ STM324/TR31	memory interfaces, USB High S Interfaces, USB High S Interfaces Interfaces, USB High S Interfaces,	Beterent STM32HTR3BKx STM32HTR3BKx	se ADC 12- 0 0 0	د Espot bit (convertors) کا Marketory Status Coming soon Coming soon Coming soon
Check/Uncheck All (2) Check/Uncheck All (2) STM32H780 Value line STM32H7R3/7S3 STM32L7/R3/7S7 STM32L0x0 Value Line STM32L0x1 STM32L0x2 STM32L0x3		MCU	graphic accelerations, external Features • NA Is/MPUs List: 34 items	Imperiation Part No No STM32H7R3A8 SY6 STM32H7R3B8 SY6 STM32H7R3B	Beteren STM32HTR38bx STM32HTR38bx STM32HTR38bx STM32HTR38bx	se ADC 12- 0 0 0 2	Export Converters) Marketing Status Coming soon Coming soon Coming soon Coming soon Coming soon
Check/Uncheck All (2) STM324/780 Value line STM324/780 Value line STM324/787/757 STM32L0x1 STM32L0x1 STM32L0x2 STM32L0x2 STM32L0x3 STM32L0x4 STM32L0x4 STM32L0x4 STM32L004 STM32L100 Value Line		MCU	graphic accelerations, external Features • N/A Is/MPUs List: 34 items ☆ STM32H7R3 ☆ STM32H7R3 ☆ STM32H7R3 ☆ STM32H7R3 ☆ STM32H7R3 ☆ STM32H7R3	memory interfaces, USB High S ABI6 STM32H7R3A8 BK6 STM32H7R3B BK6 STM32H7R3L8 BH6H	Beforen STM32HTR38bx STM32H	ce ADC 12 0 0 0 1 2 H 2	t converters) × Marketing Status Coming soon Coming soon Coming soon Coming soon Coming soon
Check/Uncheck All (2) Check/Uncheck All (2) STM32H780 Value line STM32H787/7S3 STM32L77S7 STM32L0x0 Value Line STM32L0x1 STM32L0x2 STM32L0x3 STM32L0x3		MCU	graphic accelerations, external Features • N/A Is/MPUs List: 34 items ☆ STM32/H783	Interfaces, USB High S Interfaces, USB H	PHY and large set of Referance STM32H7R38bx STM32H7R38bx	ce ADC 12- 0 0 0 1 2 H 2 c 0	La Export bit (converters) >>> Matrixeting Status Corning soon Corning soon Corning soon Corning soon Corning soon
Check/Uncheck All (2) STM324/780 Value line STM324/780 Value line STM324/787/757 STM32L0x1 STM32L0x1 STM32L0x2 STM32L0x2 STM32L0x3 STM32L0x4 STM32L0x4 STM32L0x4 STM32L004 STM32L100 Value Line		MCU	graphic accelerations, external Features • N/A Is/MPUs List: 34 items • STM32H7R3 ☆ STM32H7R3	Isal Part No Part No No Total State ABI6 STM32H7R3A8 8K6 STM32H7R3B 8H6 STM32H7R3B 8H6 STM32H7R3B 8H6 STM32H7R3B 8H66 STM32H7R3B	Breferen Tradition Breferen STM32H7R3Bkx STM32H7R3Bkx	ce ADC 12 0 0 1 1 2 1 1 2 1 1 2 1 2 1 1 2 1 2 1	Expon Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon
Check/Uncheck All (2) Check/Uncheck All (2) STM32H780 Value line STM32H787/7S3 STM32L77S7 STM32L0x0 Value Line STM32L0x1 STM32L0x2 STM32L0x3 STM32L0x3		MCU	graphic accelerations, external Features • N/A Is//MPUs List: 34 items ☆ STM32H7R3 ☆ STM32H7R3	Interfaces, USB High S Interfaces, USB High S Not No Not No STM32H7R3A8 R/6 STM32H7R3A8 R/6 STM32H7R3A8 R/6 SIM64 SIM64 SIM64 SIM64 SIM64 SIM64 SIM64 SIM32H7R3R8 V8/6 STM32H7R3R8 V8/6 STM32H7R3R8 V8/6 STM32H7R3R8 V8/6 STM32H7R3R8	2 Before of STM32H7R3ABk STM32H7R3ABk STM32H7R3B7k STM32H7R3B7k STM32H7R3B7k STM32H7R3B7k STM32H7R3B7k STM32H7R3VFK STM32H7R3VFK STM32H7R3VFK	ke ADC 12. 0 0 0 0 0 2 H 2 c 0 c 0 c 0 c 0 c 0 c 0	Convertors) × Marketing Status Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon
Check/Uncheck All (2) Check/Uncheck All (2) STM32H780 Value line STM32H783/7S3 STM32L0x0 Value Line STM32L0x1 STM32L0x2 STM32L0x2 STM32L0x3 STM32L100 Value Line STM32L151/152 STM32L162		MCU	graphic accelerations, external Features • N/A Is/MPUs List: 34 items • STM32H7R3 ☆ STM32H7R3	Interfaces, USB High S Interfaces, USB H	Breferen Tradition Breferen STM32H7R3Bkx STM32H7R3Bkx	ke ADC 12. 0 0 0 0 0 2 H 2 c 0 c 0 c 0 c 0 c 0 c 0	Expon Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon

Figure 424. Choose an STM32H7R product

2. Click on the Start Project button, then choose "Yes" on the "Memory Protection Unit for Cortex-M7" dialog box.

	Figure 425. Initialization dialogue						
MX Me	emory Protection Unit for Cortex-M7 ×]					
0	For this Cortex-M7 device, it is highly recommended to preconfigure the Memory Protection Unit to optimize the Speculative Read access of the core.						
	Do you want to apply now such default configuration ?						
	<u>Y</u> es <u>N</u> o						

STM32CubeMX applies the default configuration, then adds a 4-Gbyte region called "Region 0" under the CORTEX_M7_BOOT parameters, and a 4-Gbyte region called "Region 0" under the CORTEX_M7_APPLI parameters. The two regions start at the same address, adjust it to avoid overlap.

The new parameters can be checked using the Pinout and Configuration tab.



P	inout & Configuration			Clock Configuration	
	nout a configuration			✓ Software Pack	ks
		~	0		BOOT Mode and Configuration
tegories A->Z			<i>v</i>	CORTEX_III_I	Mode
			~	Runtime contexts:	
System Core				Boot	Application ExternalMemoryLoader
CORTEX M7 APPLI	Boot	Application	ExternalMemoryLoader		
CORTEX_M7_APPEI					
FLASH					
GPDMA1		2			
GPIO					
HPDMA1					
NVIC_APPLI		2			
NVIC_BOOT					
RCC					
ultimedia incurity imputing dideware and Software Packs ace and Debug wer and Thermal			> > > > >	Parent Configuration • Paranteer Settings • User Constants Configuration • O Configuration • O	Configuration
				 Speculation default mode Settings 	·
Itilities			>	Speculation default mode	Enabled
ther			>	 Cortex Interface Settings 	
uner				CPU ICache	Disabled
				CPU DCache Cortex Memory Protection Unit Control Settings MPU Control Mode Cortex Memory Protection Unit Region 0 Settings	Disabled Background Region Privileged accesses only + MPU Disab
				MPU Region	Enabled
				MPU Region Base Address	0x0
				MPU Region Size	4GB
				MPU SubRegion Disable	0x87
				MPU TEX field level	level 0
				MPU Access Permission	ALL ACCESS NOT PERMITTED
				MPU Access Permission MPU Instruction Access	ALL ACCESS NOT PERMITTED DISABLE
				MPU Instruction Access	ALL ACCESS NOT PERMITTED DISABLE ENABLE
					DISABLE
				MPU Instruction Access MPU Shareability Permission MPU Cacheable Permission	DISABLE ENABLE DISABLE
				MPU Instruction Access MPU Shareability Permission MPU Cacheable Permission MPU Bufferable Permission	DISABLE ENABLE
				MPU Instruction Access MPU Shareability Permission MPU Cacheable Permission	DISABLE ENABLE DISABLE
				MPU Instruction Access MPU Shareability Permission MPU Cacheable Permission MPU Bufferable Permission Cottex Memory Protection Unit Region 1 Settings	DISABLE EINABLE DISABLE DISABLE

Figure 426. Region0 added

- 3. Select the Tools tab:
 - a) Choose Memory Management
 - b) Activate the Memory Management Tool support by clicking on "Apply Application Regions Settings to Peripherals"

Pinout & Configuration	Clock Configuration		Project Manager			
Code Generation Configuration	Memory viewed by ARM Cottex-M7		Context Boot	+	Context Appli	
Apply Application Regions Settings to Peripherals: ON Apply Application Regions Settings to Linker Files: OFF	Reserved			<u>a</u> :		
Appy Appication regions bettings to Linker Files: OFF	FMC SDRAM Bank2 (256 MB)	NA		NA		
Search an Application Region	FMC SDRAM Bank1 (256 MB)	NA		2vA		
Show selected Application Region	cuarcosco Reserved			â :		
	(whereason KSPI1 (256 MB)	NA		N/A		
	Cw10000000 FMC Bank3 (256 MB)	NA		AVG		
	c#7000000 KSPI2 (256 MB)	NA		2VA		
	Cwtococco FMC Bank1 (256 MB)	NA		2VA		
	Cw38801000 Reserved			e :		
	Cx38800000 Backup SRAM (4 KB)					
	Calcootcoo Reserved			<u> </u>		
	Cis30004000 AHD SRAM2 (16 KB)					
	Ex30000000AHB SRAM1 (16 KB)					
	Cw24072000 Reserved			<u> </u>		
	AVI SRAM4 (72 KB)					
	6w24060000					
	0x24040000 AXI SRAM3 (128 KB)	RAM		RAM		
	0#24020000 AXI SRAM2 (128 KB)					
	C#24000000 AXI SRAM1 (128 KB)					
	c#20010000 Reserved					
	cx2000000DTCM memory (64 KB)	DTCM		DICM		
	C#1FF20000 Reserved					
	CH1FT00000 System memory (128 KB)					
	castolocoo Reserved	FLASH		: : HRAIK		
	Cast Council Flash memory (64 KB)					
	cxsoc1cccoReserved cxsocccccccoTCM memory (64 KB)	TCM		а : лсм		

Figure 427. Activate Memory Management support



- 4. Select the Project Manager tab
- 5. Give a name to the project and press the Generate Code button: a warning message is displayed.

nessage
×
sh of context Appli are overlapped.

The flash region overlap issue can be solved in different ways, the preferred one goes through the following steps:

- a) Select the Pinout and configuration tab
- b) Enable XSPI1 for the boot context and choose the 'Single SPI' mode

Figure 429. Configure the XSPI

lome	STM32H7R3A8Ix	/ Mer	norymapror	51W52H/R.10	c - Pinout & Configuration	"/		GENERATE CODE	
	Pinout & Configu	ration		Clock Co	onfiguration	Project Man	ager	Tools	
				 Software P 					
				~ 🔕		XSPI1 Mod	e and Configuration		
ategori	ies A->Z						Mode		
	÷	Boot	Application	ExtMemLo	Runtime contexts:				
					Boot	A	oplication	ExternalMemoryLoader	
	BE-TOUCHGFX								
	BE-TOF1				Mode Single SPI				~
	BE-SUBG2				Mode Single SPI				
					Port Port1 Single				\sim
	BE-SFXS2LP1				HyperBus(TM) 1.8V Invert	led Clock Disable			
	BE-NFC7								
	BE-NFC6				Chip Select Override Disa	ble			~
	BE-NFC4					<u> </u>	nfiguration		
	BE-MEMS1					00	Ingulation		
					Reset Configuration				
	BE-BLEMGR				Parameter Settings	MMT 🛛 🥺 User Constants	OMA Settings	GPIO Settings	
	BE-BLE2				Configure the below paramete	rs :			
					Q Search (Ctrl+F)	0			
					✓ Generic	0			
	BE-ALGOBUILD				Fifo Threshold		1		
X-CUE	BE-AI				Memory Mode		Disable		
WWD	G						Micron		
	BUE				Memory Type				
							to Bytes		
	DTG_HS				Memory Size Chip Select High Tir	me Cycle	16 Bytes 1		

- c) Activate the Middleware EXTMEM_MANAGER for the boot context:
 - > MMT solves the issue
 - Press the Generate Code button to generate code for both applications. The overlap message does not appear any longer.



me 🔪 STM32H7R3A8Ix 🕽	> Men	noryMapforSTM32H7F	ioc - Pinout & Configuration	\rangle		GENERATE CODE
Pinout & Configura	tion	Clock	Configuration	Project Manager		Tools
		✓ Softwa	e Packs 🗸 🗸 Pir	nout		
		~	0	EXTMEM_MANAGER M	ode and Configura	a
tegories A->Z				Mode	9	
omputing		2	Runtime contexts:			
		~	Boot	Applica	tion	ExternalMemoryLoader
iddleware and Software Packs		~				
÷	Boot	Application ExtMemL	Activate External Memory	orv Manager		
AIROC-Wi-Fi-Bluetooth-ST						
EXTMEM_LOADER		V				
EXTMEM_MANAGER	\checkmark					
FATFS						
FP-SNS-MOTENVWB1						
FP-SNS-SMARTAG2	_			Configura	ation	
FREERTOS			Reset Configurati			
I-CUBE-CANOPEN						
I-CUBE-ITTIADB			🗢 Boot usecase 🛛 📀 Mem	nory 1 🛛 🥺 Memory 2 🛛 🥥 User	Constants	
I-CUBE-embOS			Configure the below parameter	rs :		
I-CUBE-wolfSSL			Q Search (Ctrl+F)	0		0
I-Cube-SoM-uGOAL			✓ Boot			_
LWIP			select boot code ge	neration	/	
			Selection of the boo		xecute In Place	
USB_DEVICE			~ XIP	-		
USB_HOST			select the memory	м	lemory 1	
X-CUBE-AI						

Figure 430. EXT_MEM_MANAGER

Code generation configuration

The application regions settings can be applied to peripherals on the left-hand side of the screen. The concerned peripherals are shown on the associated tooltip. This can impact their availability on the pinout screen configuration.

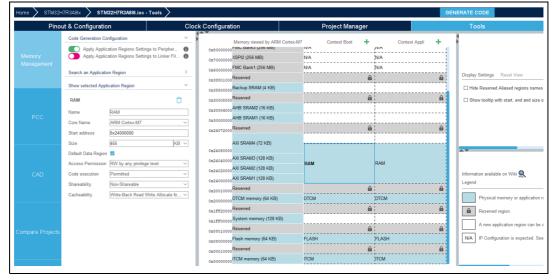


Figure 431. Tooltip

In this example, on the Pinout & Configuration panel, Cortex-M7_BOOT (MPU) and Cortex-M7_APPLI (MPU) are set and correspond to the region configuration on the Memory Management Tool. They are grayed out, as they cannot be modified.



Pinou	it & Co	nfiguration		Clock Configuration		Project Manager
				✓ Software Packs	✓ Pinout	
			~ 6	CORTEX	M7_APPLI Mode and Configura	tion
tegories A->Z					Mode	
System Core			~	Runtime contexts: Boot	Application	ExternalMemoryLoader
÷	Boot	Application	ExtMemL	Boot	Application	ExternalmemoryLoader
CORTEX M7 AP	0001		Lottering	1		
CORTEX M7 BO	2				Configuration	
FLASH				D 10 C 1		
GPDMA1	~			Reset Configuration		
GPIO		-		Parameter Settings Subser Constants		
HPDMA1	~			Configure the below parameters :		
				Q Search (Ctri+F) ③ ④		
NVIC APPLI				Speculation default mode Settings		
NVIC BOOT				Speculation default mode Settings	Disabled	
RCC	1			V Cortex Interface Settings	unsabled	
SYS				CPU ICache	Disabled	
				CPU DCache	Disabled	
				 Cortex Memory Protection Unit Control Settings 		
				MPU Control Mode	Background Region	Privileged accesses only + MPU Disable
Analog			>	Cortex Memory Protection Unit Region 0 Settings		
- and - grand				MPU Region	Enabled	
Timers			>	MPU Region Base Address	0x08000000	
				MPU Region Size	64KB	
Connectivity			>	MPU SubRegion Disable	0×0	
				MPU TEX field level	level 0	
Multimedia			>	MPU Access Permission	Privileged READS F	Permissions
				MPU Instruction Access	ENABLE	
Security			>	MPU Shareability Permission	DISABLE	
				MPU Cacheable Permission	ENABLE	
Computing			>	MPU Bufferable Permission	DISABLE	
				 Cortex Memory Protection Unit Region 1 Settings 		
Middleware and Software	e Packs		>	MPU Region	Disabled	
Trace and Debug			>	 Cortex Memory Protection Unit Region 2 Settings 		
				MPU Region	Disabled	

Figure 432. IP configuration

Apply Application Regions settings to linker files

When this button is on, the linker scripts for the Boot project and Appli project are generated, taking into account the configuration.

Home > STM3	2H7R3I8Kx 🔰 Untitled - Tools 🔪					GENERATE CODE
F	Pinout & Configuration	C	Clock Configuration	Project Manager		Tools
	Code Generation Configuration	~	Memory viewed by ARM Cortex-M7	Context Boot	Conte	ext Appli 🕂
	Apply Application Regions Settings to Peripherals: ON Apply Application Regions Settings to Linker Files: ON	0	0xE0000000 Reserved	â	1	â :
Memory Management	Apply Application Regions Settings to Linker Files. ON	•	OxD0000000 FMC SDRAM Bank2 (256 MB)	N/A	N/A	
management	Search an Application Region	>	0xC0000000 FMC SDRAM Bank1 (256 MB)	N/A	N/A	
	Show selected Application Region	>	0xA0000000 Reserved	a		a
			0x90000000XSPI1 (256 MB)	N/A	N/A	
			0x0000000 FMC Bank3 (256 MB)	N/A	N/A	
			0x70000000XSPI2 (256 MB)	N/A	N/A	
PCC			0x60000000 FMC Bank1 (256 MB)	N/A	N/A	
			0x38801000 Reserved	â		â
			0x38800000 Backup SRAM (4 KB)			
			0x30008000 Reserved	â		â
			0x30004000 AHB SRAM2 (16 KB)			
			0x30000000 AHB SRAM1 (16 KB)			
CAD			0x24072000			
			AXI SRAM4 (72 KB)			
			0x24060000 0x24040000 AXI SRAM3 (128 KB)			
			0x24040000 AXI SRAM2 (128 KB)	RAM	RAM	
			0x24020000 0x24000000 AXI SRAM1 (128 KB)			
			0x24000000 0x20010000 Reserved			â
DDR Test Suite			0x20010000 0x20000000 DTCM memory (64 KB)	DTCM	DTCM	
			0x1FF20000	â		â
			0x1FF00000 System memory (128 KB)			
			Received	A		A

Figure 433. Linker files update



ome > ST	M32H7S3IBKx 🔪 test.ioc - Tools 🔪			GENERATE CODE
	Pinout & Configuration	Clock Configuration	Project Manager	Tools
enory inegement	Code Cesentation Configuration Code Cesentation Configuration Apply Application Regions Settings to Enter Fil. Bearch an Application Region Show selected Application Region	Manody sineed by ARM Cotes MT Cald/070007 Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserv	Context Boot + NA 92A NA 92A NA 92A NA 92A NA 92A NA 92A NA 92A	Context Appi
		eadlocades AHD SRAAD (16 KD) Nam Gabloocod AHD SRAAH (16 KD) Add Gabloocod AHD SRAAH (16 KD) Add Gabloocod AN SRAAH (12 KD) Gabloocod AN SRAAH (120 KD) Gabloocod AN SRAAH (120 KD)	ess 0x38800000 4 ○ B ● KB ○ MB	e Promation publicito on Weis & Legard
		Ca200000 Reserved Ca200000 DTCM evenny (64 KB) Ca2770000 Reserved Ca2770000 Reserved Ca2000000 Reserved Ca2000000 Reserved Ca2000000 Reserved Ca2000000 Reserved	ртом ўтом ртом ўтом ртам рт.434 г.434 г.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.434 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт.4344 рт	Physical memory or application region Physical memory or application region A new application region can be ad NA P Configuration is expected. See to

Figure 434. Memory assignment for context Boot H7RS

EXTMEM_MANAGER when using H7Rx/H7Sx

The middleware can be used with the "Select boot code generation" disabled or enabled.

If disabled, MMT automatically chooses the configured memory along with the associated driver, and sets the execution memory location in the linker file. This is the most straightforward way of configuring an external memory.

If enabled, by activating the "Select boot code generation" you can choose "Execute in Place" or "Load and Run"

- Execute in Place chooses and configures the memory zones
- Load and Run lets the user choose source, destination memory, and addresses to jump to. The configuration is translated into the linker file. The user must provide the source and destination addresses.

Pinout & Confi	guration	_	Clo	ck Configuration		Project Manager	Tools
				✓ Software Packs	✓ Pinout		
		~ ·	0	EX	TMEM_MANAGER Mode and Cor	nfiguration	Pinout view III System view
egories A->Z					Mode		
iddleware and Software Packs			~	Runtime contexts:			
AIROC W Fi Burkoth STAD	Boot	Application	ExtMemLoader	Boot	Application	ExternalMemoryLoader	
EXTMEM LOADER			8	Z Activate External Memory Ma	mager		
EXTMEM MANAGER					1.5		
FAITES							
FF-SNS-MOTENVMB1							
FP-SNS-SMARTAG2							
		0					
ICUBE CANOPEN							
							0000000000000000
/-Cube-SmM+/GOAL							
							000000000000000000000000000000000000000
		0					
	-						
X-CUBE ALGOBUILD							
X-CURE-ALS							000000000000000000000000000000000000000
X-CUBE-BLE1					Configuration		
X-CUBE BLE2					Substitute Automation		
X-CUBE-BLEMGR				Reset Configuration			
X-CUBE-EEPRMA1				Bost usecase Memory 1	A Marrier 2	darde	
X-CUBE-GNSS1					• Warmery 2 • User Cens	14115	
X-CUBE-ISPU X-CUBE-4EMS1				Configure the below parameters :			000000000000000000000000000000000000000
X-CUBE-NFC4				Q Search (Chit+F) 0 0		0	
X-CUBE-NPC6				~ Bost			
X CUBE AFCT			-	select boot code generation	m		
X-CUBE-SFXSRPt				and the search of the search			UFBGA169 (Top view)
X-CUBE-SNBUS							
X-CUBE-SUBG2							
X-CUBE-TOF1							
X-CUBE-TOUCHGFX							

Figure 435. EXTMEM_MANAGER "Select boot code generation" disabled



Pinout & Configuration	Application E		K Configuration Software Packs ExtM Buttime contexts Boot C C C Activate External Memory Manag	Pinout MANAGER Mode and Confi Mode Application	project Manager guistion Externat/Kentory,Coader छ	Pinsut view ¹⁰ System size
defector and Software Packs Boot All 1 AIRED NAL DOORD Boot All EXTRUDAL LONGRA C All EXTRUDAL LONGRA C All PARIS MARCHAR C All AREDINAL CONSTRUMENT C All PARIS MARCHAR C All PARIS MARCHAR C C I CARE CONSTRUMENT C C PARIS MARCHAR C C I CARE CONSTRUMENT C C	Application E	ExtMemLoader	EXTM Runtime contexts: Boot	M_MANAGER Mode and Conf Mode Application	ExternalMemoryLoader	Phoet view 1 ⁰ System size
Addresser and Software Packs Boot All Add/Coll C Boot All ERMEMA (Software Packs) C C C Filesando Contraventaria C C C C Filesando Contraventaria C C C C Filesando Contraventaria C C C C C C C C C C C	Application E	ExtMemLoader	Runtime contexts: Boot	Mode Application	ExternalMemoryLoader	Pinaut view ¹⁰ System view
Idensity and Software Packs Boot All ALRCS INF ACCER C All ExtraCall (MACCER) C All ExtraCall (MACCER) C All ExtraCall (MACCER) C All FREENTOR C All Colled C-Asonger C All Colled C-Asonger C All Colled C-Asonger C C Colled C-Based C C Colled C		ExtMemLoader	Boot	Application		
P Boxt A EXTEM (JAOGER		ExtMemLoader	Boot			1 000000000000000000000000000000000000
URDC VM-26.Basebo 37.027 Image: Comparison of the comparison o		8	8			P
EXTINUEA (LAADER EXTINUEA (LAADER TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS TARIS T				e		
XMMEM_NAMAGER CI XMMEM_NAMAGER CI YANS_MATACA Image: City Composition of the			a nuovat Evenna vernury varua			
APP3 Image: Control of Con		~				
PLANS.MORTSVYNBY PLANS.MORTSVYNBY REXITOR						
PLANS.SMATAG2 PLANTAG2 PLANTAG						
Imperation Imperation CodeE CANOPER Imperation CodeE Comm Imperation CodeE Common Imperation CodeE Common Imperation CodeE Amage Imperation CodeE Amage Imperation State Imperation CodeE Amage Imperation </td <td></td> <td></td> <td></td> <td></td> <td></td> <td>000000000000000000000000000000000000000</td>						000000000000000000000000000000000000000
Calle C-Assing A Caller Control Contro						
Calle C. Andre N Calle C. Calles m Calles C. Market Calles C. Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Market Marke						
Cube Cosinin Cube Carlino Cube Analos Cube Analos Sinip D Sinip D Sinip D Sinip D Sinip D Sinip D Cube Analos Cube Analos Cu						
Cuele Annole Cuele Annole Cuele Annole Annol						
Cutler a-raids Cutler a-raids Cutler a-raids Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp Simp			1			Concy
Cutter Sum State Cutter Sum State State WP Cuter Sum State State State State State State State State State State Cuter State State State Cuter State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State						000000000000000000000000000000000000000
Code Selvidora, WWP Code Selvidora, Moderno Selvidora, Moderno Selvidora, Moderno Code Selvidora, Moderno Selvidor Moderno Selvidora, Moderno Selvido	0		4			
John Image: Control of Con			1			220000000000000
SINPO ID SINPOST ID SIN_POST ID CAREE AN CONTRACT ID CAREE AN CONTRACT ID CAREE AND CONTRACT <t< td=""><td></td><td></td><td></td><td></td><td></td><td></td></t<>						
GR_DOVICE						
SR /1037 Cutte Arabourg Cutte Arabourg Cutte Arabourg Cutte Arabourg Cutte Arabourg Cutte Refain Cutte Refai						
						000000000000000000000000000000000000000
				Configuration		
						000000000000000000000000000000000000000
			Reset Configuration			
			And a second sec			<u></u>
			Boot usecase O Memory 1	Memory 2 🛛 User Consta	ints	
			Configure the below parameters :			
			Q Search (Ctrl+F)		0	
			✓ Boot			000000000000000000000000000000000000000
			select boot code generation			
			Selection of the boot system	Execute In Pla	ce v	
			~ 10P	Execute In Pla		
			select the memory	Load and Run		UFBGA169 (Top view)
			and the memory	promo and Hun		
			1			
			1			
						1
e and Debug						
er and Thermal						
ies						
w						

Figure 436. Execute In Place

Figure 437. MMT Execute In Place

Pinout & Conf	figuration		Clock Configuration	Pi	roject Manager	Tools	
Code Generation C	*	~	Memory viewed by ARM	/ Cortex-M7	Context Boot +	Context Appli	4
	cation Regions Settings to Peri cation Regions Settings to Link		OxE0000000 Reserved		₽ ;	â	
Appry Appr	cation Regions Settings to Link	er Piles: ON D	PMC SDRAM Bank2 (256 MB)	N/A	'N/A		
Search an Applicat	ion Region	>	0xC0000000 FMC SDRAM Bank1 (256 MB)	N/A	N/A		
Show selected App	dication Dagion	~	OxA0000000 Reserved		a	a	
Gilow attected App	incariori regioni						
RAM		Û	External Flash (XSPI1) (256 MB) 0x90000000		FLASH		Display
Name	RAM		0x10000000 FMC Bank3 (256 MB)	N/A	N/A		Hide
Core Name	ARM Cortex-M7	~	0x70000000 XSPI2 (256 MB)	N/A	N/A		□ Sho
Start address	0×24000000		0x600000000 FMC Bank1 (256 MB)	N/A	N/A		
Size	120	KB V	0x38801000 Reserved		a :	â	
Default Data Region	n 🖬		0x388000000 Backup SRAM (4 KB)				
Access Permission	RW by any privilege level	~	0x30008000 Reserved		a ;	â	
Code execution	Permitted	~	0x30004000 AHB SRAM2 (16 KB)				
Shareability	Non-Shareable	~	0x30000000 AHB SRAM1 (16 KB)				
Cacheability	Write-Back Read Write Alloc	ate for RA 🗸	0x24072000 Reserved		≙ :	a	
			0x24060000 AXI SRAM4 (72 KB)				
			0x24040000 AXI SRAM3 (128 KB)				
			0x24020000 AXI SRAM2 (128 KB)				
			AXI SRAM1 (128 KB)				
			0x24000000	RAM	RAM		Informat
			0x20010000 Reserved		a :	â	Legend
			0x20000000 DTCM memory (64 KB)	DTCM	ртсм		
			0x1FF20000Reserved		<u> </u>	â	
			0x1FF00000 System memory (128 KB)				â
			0x08010000 Reserved		<u> </u>	<u> </u>	
			0x080000000 Flash memory (64 KB)	FLASH	Reserved	I FLASH(Boot)	
			0x00010000 Reserved		£	<u> </u>	N/A



e 🔪 STM32H7R3A8k 🔪 U	Untitled - Pino	ut & Configurat	tion >			GENERATE CODE
Pinout & Config	uration		C	lock Configuration	Project Manager	Tools
				✓ Software Packs	✓ Pinout	
		Ŷ	Ø	and the second	I_MANAGER Mode and Configuration	Pinout view I ¹¹ System view
gones A->Z				6	Mode	
dleware and Software Packs			· · ·	Runtime contexts:		
	-			Boot	Application ExternalMemoryLoader	
AIROC-Wi-Fi-Bluetooth-STM32	Boot	Application	ExtMemLoader	2	5	
EXTMEM LOADER			5	Activate External Memory Manager		
EXTMEM MANAGER		1	8			
EATES						
	U	L				
	0	D				
	0					
						000000000000000000000000000000000000000
	0					
	0					
		0				
	0					
	U	U.				
					Configuration	
					Computation	
				Reset Configuration		
				Boot usecase Memory 1	Memory 2 Oser Constants	000000000000000000000000000000000000000
X-CUBE-EEPRNIA1				Configure the below parameters :		
X-CUBE-GNSS1				Q Search (Chil+F) O O	(
			1	~ Boot		
				select boot code generation		
				Selection of the boot system	Load and Run	
				 LRUN source 	Costs and Home	
				select the source memory	Memory 1	UFBGA169 (Top view)
				source address offset	0x00000000 hex	a. control (rop tien)
				source address onset	0x00000000 hex	
				V LRUN destination	- Annonna ma	
				selection of the memory	Internal Memory	
				destination address	0x24050000 hex	
ce and Debug			>			
ver and Thermal			>			
ities						
her						Q [] Q [] 4 11

Figure 438. Load and Run

Figure 439. MMT Load and Run

	Pinout & Confi	guration		Clock Co	nfiguration	Project Ma	nager		Tools	
	Code Generation Co		~		Memory viewed by ARM Cortex-M7	Co	ntext Boot +	Context App	+	
		ation Regions Settings to P		0xE0000000	Reserved		8	8	a :	
ry gement	Apply Applic	ation Regions Settings to L	inker Files: ON 🌘		FMC SDRAM Bank2 (256 MB)	N/A		λ/A		
	Search an Application	on Region	>		FMC SDRAM Bank1 (256 MB)	N/A		N/A		
	Show selected Appl	ication Design	~		Reserved		â		â	
	Show selected App	Cation Region		0x90000000	External Flash (XSPI1) (256 MB)	Reserved LOAD_REGIO	W(Appli)	Reserved LOAD_REGION	a	
	RAM		Û	0x80000000	FMC Bank3 (256 MB)	N/A		A/V		Disp
	Name	RAM			XSPI2 (256 MB)	N/A		λ/A		
	Core Name	ARM Cortex-M7	~	0x60000000	FMC Bank1 (256 MB)	N/A		\$N/A		
	Start address	0×24000000		0x38801000	Reserved		8	l.	a :	
	Size	120	KB ~	0x38800000	Backup SRAM (4 KB)					
	Default Data Region				Reserved		â		a :	
	Access Permission	RW by any privilege level	~	0x30004000	AHB SRAM2 (16 KB)					
	Code execution	Permitted	~	0x30000000	AHB SRAM1 (16 KB)					
	Shareability	Non-Shareable	~	0x24072000	Reserved		8	l <u>.</u>	<u> </u>	
	Cacheability	Write-Back Read Write Al	locate for RA 🗸	0x24060000	AXI SRAM4 (72 KB)					
					AXI SRAM3 (128 KB)	Reserved FLASH(Appli)	<u></u>	FLASH		-
				0x24040000						
				0x24020000	AXI SRAM2 (128 KB)					
					AXI SRAM1 (128 KB)					
				0x24000000		RAM		RAM		Infor
				0820010000	Reserved		â		<u> </u>	Lege
				0440000000	DTCM memory (64 KB)	DTCM		DTCM		
				0819920000	Reserved		â		a	
				0812200000	System memory (128 KB)					6
				0x08010000						
				0x08000000	Flash memory (64 KB)	FLASH		Reserved FLASH(Boot)	<u> </u>	
				0x00010000	Reserved ITCM memory (64 KB)	ITCM	<u>é</u>	l)	Ê	1

After the code generation, navigate to the generated folder.

- Under the boot Project, open the linker definition file.
- Under the Memories definition you can see the defined memories with their start address and length, according to the configuration made in STM32CubeMX.





ENTRY (Reset_Handler)					
/* Highest address o	f the user mode stack */				
) + LENGTH (RAM) ; /* end of Ram type memory */				
	00; /* required amount of heap */				
Min Stack Size = 0x	400; /* required amount of stack */				
/* Memories definiti	on */				
MEMORY					
FLASE (IX) : OR					
	IGIN = 0x08000000, LENGTH = 64% IGIN = 0x00000000, LENGTH = 64%				
	IGIN = 0x200000000, LENGTH = 64R				
	IGIN = 0x240000000, LENGTH = 455E				
LI CALAN : ON	Iota = Oklayovovy, Elapin = apph				
/* Sections */					
SECTIONS					
/* The startup cod	e into "FLASH" Rom type memory */	Memory viewed by ARM Contex-M7	Context Boot	+ Context Apple	ί.
.isr vector :		Caroossee man as party many		(***	
Ð (FMC Bank1 (256 MB)	NA	104	******
. = ALIGN (4) ;		(1260000000) Here Dank ((200 Here)			
	or)) /* Startup code */	Gu311111000 Reserved		e):	
. = ALIGN(4);		Burghow (Marked in Later		1	
-) >FLASH		CERTIFICATION CONTRACTOR CONTRACTOR			
		(x20000000 Reserved		e):	
.text :	e and other data into "FLASH" Rom type memory */	CELSON AND SRAM2 (16 KB)		1	
- (
. = ALIGN(4);		(1x10000000 AHB SRAM1 (15 KB)			
(.text)	/ .text sections (code) */	Reserved		4.	
(.text)	/* .text* sections (code) */	0x24072000 Mesenved		·····	
(.glue 7)	/ glue arm to thumb code */	AU SRAMA (72 KEI)		1	
(.glue 7t)	/ glue thumb to arm code */	the second particular			
*(.eh frame)		0x24040000			
		(128 KB)	RAM	RAM	
KEEP (*(.init))		AVI SRAM2 (128 KB)	No.	i come	
KEEP (*(.fini))		CL2412000			
		0x2400000 ANI SRAM1 (128 KB)			
. = ALIGN(4);		Reserved		6	
_etext = .;	<pre>/* define a global symbols at end of code */</pre>		DTCM		
-) >FLASH		tradecounter (64 KS)	DICM	,DTCM	
		Reserved			
11 m	ato "FLASH" Rom type memory */	(x1770000 System memory (128 KB)			******
/* Constant data i		System memory (125 K25)		2	
.rodata :		0x1FF00000			
.rodata :		Deserved		6	
.rodata : (. = ALIGN(4);	/* rodata martices (constants, strings, ato 1 #/	Reserved			
.rodata : { (. = ALIGN(4); *(.rodata)	/* .rodata sections (constants, strings, etc.) */ /* .rodata* sections (constants, strings, etc.) *	Reserved	FLASH	FLASH	
.rodata : (. = ALIGN(4);	/* .rodata sections (constants, strings, etc.) */ /* .rodata* sections (constants, strings, etc.) *	Reserved	FLASH	6 FLASH	

Three option bytes can be used to configure the regions in the MMT. To see them, activate the IP FLASH on the Pinout and Configuration tab.

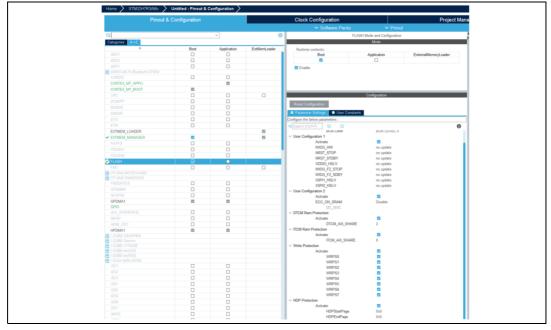


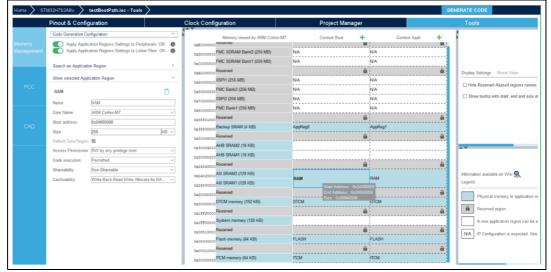
Figure 441. Flash option bytes



The option bytes interacting with the MMT are:

- ECC_ON_SRAM:
 - Linked to the AXI SRAM4 region on the MMT
 - When value is "disable" or "no update", the AXI SRAM4 region size is set to 72 KB
 - When value is set to "enable" the AXI SRAM4 region is removed
- DTCM_AXI_SHARED:
 - Linked to the AXI SRAM3 region on the MMT
 - When set to 0 or 3, the AXI SRAM3 region size is set to 128 KB, and the size of region named DTCM is set to 64 KB
 - When set to 1, the AXI SRAM3 region size is set to 64 KB, and the size of region named DTCM is set to 128 KB
 - When set to 2, the AXI SRAM3 region is removed, and the size of region named DTCM is set to 192 KB
- ITCM_AXI_SHARED:
 - Linked to the AXI SRAM1 region on the MMT
 - When set to 0 or 3, the AXI SRAM1 region size is set to 128 KB
 - When set to 1, the AXI SRAM1 region size is set to 64 KB
 - When set to 2, the AXI SRAM1 region size is removed

Figure 442. ECC_ON_SRAM enabled and DTCM_AXI_SHARED set to 2





ETH impact on MMT when using H7RS/H7SX

An example of MMT configuration of the ETH IP on the STM32H7R3A8Ix MCU

- 1. Activate the IP ETH:
 - MMT creates three application regions for each context.
 - To change the start address and the size of each region, update the ETH parameters.
- 2. Press the radio button "Apply Application region Settings to Peripherals ON", ETH will be partially under MMT control.
- 3. Press the Generate Code button to generate code for both applications.
 - Apply Application Regions settings to linker files:
- 4. When this button is on, the linker scripts are generated, considering the configuration.
- 5. After the code generation, navigate to the generated folder:
 - Open the linker definition file.
 - Under the Memories definition you can see the memories with their start address and length, according to the configuration made in STM32CubeMX.

STM32CubeMX Untit	led*: STM32H7R3A8Ix								- 5 X
STM32	File		Window		Help				🎯 🗗 🕨 X 🗘 🍳 🕂 🖅
Home 🔪 STM	132H7R3A8Ix $ angle$ Untit	tled - Pinout &	Configuration	\rangle					GENERATE CODE
F	Pinout & Configurati	on		• Cl	ock Configurat	ion		Project Manager	Tools
Q			v	٥	ET	H Mode and Config	uration	🖸 😳 Pir	out view III System view
Categories A->2	Z					Mode		·	
Connectivity	_		~		Runtime contexts:			1	1
			1		Boot	Application	ExtMemLoader	1	00000000
C ETH	Boot V	Application	ExtMemLoader		<				POT POS PHILI PHIL PHIL PHIL PHIL POT
FDCAN1					Mode MII		~	VEAT OF A PROVIDE	
					ETH PHY_INTN				
FMC					Activate Rx Err si	anal		PC15-0) PC14-0) (VDD) (VSS) (VCD	VSS PWJ V0033 VSS V00 PD3 PE12 PE11
						Configuration		i KXXXX	
				15		oomgaration		NDDSM. (VSSSM.) STH. MIL (PB9) (PE1	PF0 PW13 VSSUSE PW8 P50 VSS PD2 PD1
					Reset Configuration				A A A A A A A A A A A A A A A A A A A
13C1					NVIC Setting		GPIO Settings	(PESM) (VLSML) (PE4) (PC13) (VCAP4	P8474 80075 DVD0 P80 P00 V00 P011 P012
LPUART1					Parameter Se	ettings	User Constants		COLDO VSS SOLLOO VOARS PAIR VSS PAIRE PCID
MDIOS				Cont	figure the below para	meters :			
				٩	Search (Ctrl+F)	0 0	0	PF8 PHD-05 VD PF7 PF9	VSS VSS VSS PCS PCT VDD PAI2 PAISU
				\vee (General : Ethernet Co	onfigurati			
SPI1 SPI2					Warning		H can work only when RA	PHI-OS THEY PCO MART STHEY	COLDO VSS SODLDO (VCP2) PCS (VSS PAS) PAS
						Address 00:80:E	E1:00:00:00		
SPI4					Tx Descriptor	Length 4 iptor Ad 0x2402	0090	THE V VSS (VREFN) (VSSA) STHEM	VCAP1 PEI PEI2 PEIA PD15 PCB PA0 PA11
					Rx Descriptor		0000	l Acce	AAAAAA
SPI6						iptor Ad 0x2402	0000	PAO (VD) (VREFP) (VDA) STILLE	PET POT YOU YS PAIL POL
UART4					Rx Buffers Ad		0100		
UART5					Rx Buffers Ler	ngth 1504			00000000
UART7								PA2 PB1 PB1 PE11 PE12	PP2 P05 P26 P26 P27 P010 P013 P013

Figure 443. ETH MMT regions for STM32H7R3A8Ix



STM32CubeMX Untitled*: STM32Hi TM32	File	Window	Help				🐵 🚹 🔼 X 🗘 🤇	- ° R 🔆 丌
Home STM32H723VE	Tx 🔰 Untitled - Pinout	& Configuration >					GENERATE CODE	
Pinout & 0	Configuration	c	lock Configuration		Project Man	nager	Tools	
ol	V @ -		✓ Software Packs	✓ Pinout				
Categories A->Z	•	EIHN	Node and Configuration Mode			Pinout view	System view	
Connectivity		Mode MII Activate Rx Err signal		~				
FDCAN1 FDCAN2 FDCAN3 FMC 12C1 (2C2 12C3 12C4 12C5 LPUART1 MDIOS		Reset Configuration	Configuration GPIO Setting		194,108 193 193 193 193 193 193 193 193 193 193	•		
CCTOSPI1 OCTOSPI2 SDMMC1 ▲ SDMMC2 SPI1 SPI2 SPI3 SPI4 SPI6		Parameter Setting Parameter Setting Configure the below parameter Search (Ctrl+F) G General : Ethernet Configur Warning Ethernet MAC Addre Tx Descriptor Lengt	Image: signal	nts ①	673 866 866 867 867 878 878 878 878 878 878	STM32F	1723VETx #8	
SWPM11 UART5 UART5 UART7 UART8 UART9 USART1 USART2		First Tx Descriptor A Rx Descriptor Lengt First Rx Descriptor A Rx Buffers Address Rx Buffers Length	lddress 0x30000080 h 4		8004 1970,528 1970,528 1970,528 1970,528 1970,528 1970,528			
USART3 USART6 USART10 A USB_OTG_HS				Q	[] Q	🗅 🖆 💷		Unused GPI

Figure 444. ETH configuration for STM32H7R3A8Ix

Figure 445. Application of the MMT configuration to the linker file

M32 V abeMX	File		Window	н	elp					🕸 📑 🕒 🗙 🧲) @ 🗙 🗛
ome > STM32H	17R3A8lx 🔰 Untit	tled - Tools 🔪								GENERATE COL	E
Pin	out & Configurati	on		Clock Conf	iguration		Project Mar	nager		Tools	
	Code Generation Co	infiguration	×		Memory viewed	by ARM Cortex-M7		Context Boot	+	Context App	6 +
		ation Regions Settings t ation Regions Settings t			FMC Bank1 (256 MB)		N/A			N/A	
	Apply the MMT con		-	0x38801000	Reserved				6		â
	Search an Applicatio		>	0x38800000	Backup SRAM (4 KB)						
	Show selected Appli	cation Region	~						Ê		â
	RAM		Ĥ		AHB SRAM2 (16 KB)						
					AHB SRAM1 (16 KB)						
	Name	RAM		0x24072000					â		â
	Core Name	ARM Cortex-M7	~	0.24000000	AXI SRAM4 (72 KB)						
	Start address	0x24000000		0x24040000	AXI SRAM3 (128 KB)					l 	
	Size Default Data Region		KB \vee				Rx_PoolSection			Rx_PoolSection	
	Access Permission	RW by any privilege k	vel ~		AXI SRAM2 (128 KB)		TxDescripSection			TxDescripSection	
	Code execution	Permitted		0x24020000			RxDescripSectio			RxDescripSection	
	Shareability	Non-Shareable	~	0x24020000							
	Cacheability	Write-Back Read Wri	e Allocate f V	0x24000000	AXI SRAM1 (128 KB)		RAM			RAM	
				0x20010000					â		â
				0x20000000	DTCM memory (64 KB)		DTCM			DTCM	
				0x1FF20000	Reserved				۵		۵
				0x1FF00000	System memory (128 KB)						
				0x0900000					۵		â i
					Flash Read Only (2 KB)						
				0x08FFF400					â		â
					User OTP (61 KB)						
				0x08010000	Reserved				6		â



are 440. Defined memory regions under the linker me of the application contex
/*************************************
/*-Editor annotation file-*/ /* IcfEditorFile="\$TOOLKIT_DIR\$\config\ide\IcfEditor\cortex_v1_0.xml" *//*-Specials-*/
/**************************************
/*-Sizes-*/ define symbolICFEDIT_intvec_start = 0x08000000;
define symbolICFEDIT_size_cstack = 0x400;
define symbolICFEDIT_size_head = 0AAD0, define symbolICFEDIT_size_head = 0A2D0;
/*-Start of symbols-Auto-generated By STM32CubeMX*/ define symbolICFEDIT_region_ITCM_start = 0x000000000; define symbolICFEDIT_region_ITCM_end = 0xFFFF;
<pre>define symbolICFEDIT_region_ITCM_end = 0xFFFF; define symbolICFEDIT_region_RxDescripSection_start = 0x24020000;</pre>
define symbolICFEDIT_region_RxDescripSection_end = 0x2402007F;
define symbolICFEDIT_region_TxDescripSection_start = 0x24020080; define symbolICFEDIT_region_TxDescripSection_end = 0x240200FF;
define symbolICFEDII_region_Rx_PoolSection_start = 0x24020100; define symbolICFEDII_region_Rx_PoolSection_end = 0x240206DF;
/*-End of MX Symbols*/
/*-Symbol_s-*/ define symbolICFEDIT_region_DTCM_start = 0x20000000;
define symbolICFEDIT_region_DTCM_end = 0x2000FFFF;
define symbolICFEDIT_region_RAM_start = 0x24000000; define symbolICFEDIT_region_RAM_end = 0x240027FF;
define symbolICFEDIT_region_RAM_end = 0x240027FF; define symbolICFEDIT_region_FLASH_start = 0x0800000; define symbolCFEDIT_region_FLASH_end = 0x8000FFF;
/**** End of ICF editor section. ###ICF###*/
define memory mem with size = 4G;
/*-MEMORY Regions-*/
define region DTCM_region = mem:[from _LCFEDII_region_DTCM_starttoICFEDII_region_DTCM_end_]; define region RAM_region = mem:[from _LCFEDII_region_RAM_starttoICFEDII_region_RAM_end_]; define region FLASH_region = mem:[from _LCFEDII_region_PLASH_starttoICFEDII_region_FLASH_end_];
define region_FLASH_region_FLASH_end;
/*-Start of Regions- Auto-generated By STM32CubeMX*/ define region IICM region = mem: from ICFEDIT region ITCM start to ICFEDIT region ITCM end];
<pre>define region RxDescripSection_region = mem:[fromICFEDIT_region_RxDescripSection_start toICFEDIT_region_RxDescripSection_end];</pre>
<pre>define region TxDescripSection_region = mem:[fromICFEDIT_region_TxDescripSection_start toICFEDIT_region_TxDescripSection_end]; define region Rx_PoolSection_region = mem:[fromICFEDIT_region_Rx_PoolSection_start toICFEDIT_region_Rx_PoolSection_end];</pre>
/*-End of MX Regions-*/
/*-Blocks-*/
define block CSTACK with alignment = 8, size =ICFEDIT_size_cstack {}; define block HEAP with alignment = 8, size =ICFEDIT_size_heap {};
<pre>/*-Initialization strategies-*/ initialize by copy { readwrite }; do not initialize { section .noint };</pre>
do not initialize { section .noinit };
/*-Sections placements-*/
/* Start of Sections- Auto-generated By STM32CubeMX*/
place at address mem:irrburi_intvec_start { readonly section .intvec };
place in ITCM region { section ITCM section }: place in KNDescripSection_region { section NxDescripSection_section };
place in TxDescripSection_region { section TxDescripSection_section }; place in Rx_PoolSection_region { section Rx_PoolSection_section };
/*-end of WX Sections-*/
/*-user Sections-*/ place in FLASH_region { readonly };
place in RAM_region { readwrite, block HEAP, block CSTACK };

Figure 446. Defined memory regions under the linker file of the application context

5.5.7 STM32WB0

Feature: MMT usage, pinout, and configuration user interface

When the first toggle button is ON, Cortex-M0+ (MPU) is under MMT control: its modes and parameters become read-only (see *Figure 447*).





Figure 447. MMT us	sage
--------------------	------

Configuration								
Reset Configuration O Parameter Settings: O User Constants								
Q Search (Ctrl+F) ③ ③	0							
Cortex Memory Protection Unit Control S	ettings							
MPU Control Mode	Background Region Privileged accesses only + M							
Cortex Memory Protection Unit Region 0	Setti							
MPU Region	Enabled							
MPU Region Base Address	0×1000000							
MPU Region Size	1MB							
MPU SubRegion Disable	0x82							
MPU TEX field level	level 0							
MPU Access Permission	Privileged READS Permissions							
MPU Instruction Access	ENABLE							
MPU Shareability Permission	DISABLE							
MPU Cacheable Permission	ENABLE							
MPU Bufferable Permission	DISABLE							
Cortex Memory Protection Unit Region 1	Setti							
MPU Region	Disabled							
Cortex Memory Protection Unit Region 2	Setti							
MPU Region	Disabled							
Cortex Memory Protection Unit Region 3	Setti							
MPU Region	Disabled							
Cortex Memory Protection Unit Region 4	Setti							
MPU Region	Disabled							
Cortex Memory Protection Unit Region 5	Setti							
MPU Region	Disabled							
Cortex Memory Protection Unit Region 6	Setti							

User interface

The middle panel represents the memory, split into two columns: the left one is the memory seen by the core Cortex-M0+, the right one the memory set-up for the application.

Figure 448. User interface	
----------------------------	--

inout & Configuration	Clock Config	uration	Project Manager		Tools	
Code Generation Configuration	~ }	Memory vi	ewed by ARM Cortex-M0+		Application Regions	+
Apply Application Regions Settings to Peripherals: ON Apply Application Regions Settings to Linker Files: OFF		Reserved				
Apply Application Regions Settings to Linker Files. OFF	0x2000C000	SRAM3 (16 KB)				
Search an Application Region		SRAM2 (16 KB)		RAM		
Show selected Application Region		SRAM1 (16 KB)		10.00		
		SRAM0 (16 KB)				
	0x100C0000	Reserved				
		⁰ Flash memory (512 KB)		FLASH		
	0x1000180	Reserved				
		0ROM memory (6 KB)		REGION_ROM		
	0x0000000	Reserved				

For a new project created under STM32CubeMX, the MMT creates the default application region to generate a valid project.



Apply Application Regions settings to linker files

When this button is on, the linker scripts for the project are generated, considering the configuration.

- The REGION_ROM is a default code region used in linker.
- The linker file copies the STM32Cube firmware linkers files and only MMT region is updated or added.
- OTA tag is not managed by MMT and usually exists in the linker file.

Figure 449. Linker files update

Code Generation Configuration	~	Memory viewed by ARM Cortex-M0+	Application Regions
Apply Application Regions Settings to Peripherals: ON	0	0x20010000Reserved	
Apply Application Regions Settings to Linker Files: ON	•	0x2000C000 SRAM3 (16 KB)	
Search an Application Region	>	0x20008000 SRAM2 (16 KB)	RAM
Show selected Application Region		0x20004000 SRAM1 (16 KB)	100m
		0x20000000 SRAM0 (16 KB)	
		ox100c0000 Reserved	
		0x10040000 Flash memory (512 KB)	FLASH
		0x10001800 Reserved	
		0x10000000 ROM memory (6 KB)	REGION_ROM
		exceeceeee Reserved	

Impact on STM32WB09 RADIO

When this IP is activated, a reserved region "Blue Core Config" calculated by value of CFG_NUM_RADIO_TASKS, which varies from 1 to 128, is added.

Figure 450. Impact on RADIO (STM32WB09)

Pinout & Configuration		(Clock Configuration		Project Mana	Project Manager		Tools	
	Code Generation Co	infiguration	<u>~]</u>		Memory viewe	d by ARM Cortex-M0+		Application Regions	+
		ation Regions Settings to Periph ation Regions Settings to Linker		0x2001000	Reserved				â
	Search an Application Region		>	0x2000C00					
Sh	Show selected Appl	Show selected Application Region $\qquad \qquad \lor$		0x20000000 SRAM2 (16 KB) 0x20004000 SRAM1 (16 KB)		RAM			
	RAM		Û	0x2000400			-		
	Name	RAM			SRAM0 (16 KB)		Blue Core Config		ê
	Core Name	ARM Cortex-M0+	~	0x2000000	5				
	Start address	0x200001A0		0x100C000	Reserved				â
	Size	63	KB 🗸	0x1004000	Flash memory (512 KB)		FLASH		
	Default Data Region	2		0x1000180	Reserved				e
	Access Permission	RW by any privilege level	~	0x1000000	ROM memory (6 KB)		REGION_ROM		
	Code execution	Permitted	~	0x0000000					6
	Shareability	Non-Shareable	~						

5.5.8 Notification MMT/boot path (STM32H7RS and STM32H5)

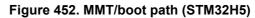
After the activation of boot path and MMT, all regions of MMT are deleted and replaced by the regions of Boot path in Appli context.

In this example, we use the boot path OEM-iRoT for STM32H7RS and for STM32H5.



Memory viewed by ARM Cortex-M7	Context Boot 🕂	Context Appli	+
External Flash (XSPI1) (256 MB)		CODE_DWL	â
0x9000000		CODE_IN_EXT_FLASH	6
0x80000000 FMC Bank3 (256 MB)	N/A	N/A	
0x70000000 XSPI2 (256 MB)	N/A	Ν/Α	
0x60000000 FMC Bank1 (256 MB)	N/A	N/A	
0x38801000Reserved	â	.	â
0x38800000 Backup SRAM (4 KB)			
0x30008000 Reserved	â		â
AHB SRAM2 (16 KB) 0x30004000		RAM	
0x30000000 AHB SRAM1 (16 KB)			
0x24072000 Reserved	â		6
0x24060000 AXI SRAM4 (72 KB)			
0x24040000 AXI SRAM3 (128 KB)			
0x24020000 AXI SRAM2 (128 KB)			
0x24000000 AXI SRAM1 (128 KB)			
0x20010000 Reserved	â		â
0x20000000 DTCM memory (64 KB)		DTCM	
0x1FF20000Reserved	â		â
0x1FF00000 System memory (128 KB)			
0x08010000 Reserved	ĥ	1	
0x08000000 Flash memory (64 KB)		BOOT_PATH_EXEC	â
0x00010000 Reserved			â

Figure 451. MMT/boot path (STM32H7RS)



Memory viewed by ARM Cortex-M33	Application Regions	+
0x0C080000Reserved		â
Flash Bank2 memory (S) (256 KB)		
0x0C040000	DOWNLOAD_NON_SECURE_CODE_REGION (NS) Reserve	d Alias R
	DOWNLOAD_SECURE_CODE_REGION	â
	FLASH (NS) Reserved Alias Region	A
Flash Bank1 memory (S) (256 KB)		
	FLASH_NSC (NSC)	
	FLASH (S)	
0x0C000000	ROT_REGION (NS) Reserved Alias Region	<u> </u>
0x0A044000 Reserved		â
0x0A034000 SRAM3 Code (NS) (64 KB)		
0x0A020000 SRAM2 Code (NS) (80 KB)		
SRAM1 Code (NS) (128 KB)	RAM (NS) Reserved Alias Region	â
0x0A00000	RAM (S) Reserved Alias Region	6
0x08080000 Reserved		â
Flash Bank2 memory (NS) (256 KB)		
0x08040000	DOWNLOAD_NON_SECURE_CODE_REGION	۵
	DOWNLOAD_SECURE_CODE_REGION (S) Reserved Alias	Region 🔒
	FLASH (NS)	

The linker files are copied from STM32Cube firmware of boot path, and MMT integrates all added application regions ("App_User").

- Open the linker files STM32H7S3I8KX_OEMiROT_Appli_app.ld or STM32H523CETX_FLASH.ld (respectively, left or right side of *Figure 453*)
- Look at the memory definition: check the "App_User" declaration in the Appli project in case of an OEM-iRoT boot path (see *Figure 454* and *Figure 455*).



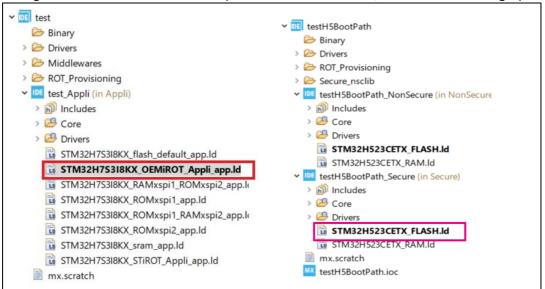
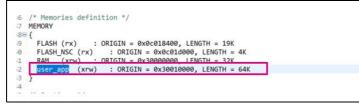


Figure 453. Linker files location (STM32H7RS on the left, STM32H5 on the right)



RAM RAM_NONCA CONF FLAC	ACHEABLEBUFFER (xrw) : ORIGIN =	RAM, LENGTH =RAM_SIZERAM_NONCACHEABLEBUFFER_SIZERAM_CONF_FLAG_SIZE RAM_NONCACHEABLEBUFFER, LENGTH =RAM_NONCACHEABLEBUFFER_SIZE RAM_CONF_FLAG, LENGTH = _RAM_CONF_FLAG_SIZE
RAMECC_HA		RAM_CONT_FLAG, LENGTH =RAM_CONT_FLAG_SIZE RAM_RAMECC_HANDLER, LENGTH =RAM_RAMECC_HANDLER_SIZE
ІТСМ	(xrw) : ORIGIN = 0x00000000,	LENGTH = 4K
DTCM	(rw) : ORIGIN = 0x20000000,	LENGTH = 4K
FLASH	(xrw) : ORIGIN =FLASH_BEGIN,	LENGTH =FLASH_SIZE
EXTRAM	(rw) : ORIGIN =EXTRAM_BEGIN,	LENGTH =EXTRAM_SIZE
	(xrw) : ORIGIN = 0x30000000, L	

Figure 455. App_User declaration (STM32H5)





6 STM32CubeMX C Code generation overview

6.1 STM32Cube code generation using only HAL drivers (default mode)

During the C code generation process, STM32CubeMX performs the following actions:

- If it is missing, it downloads the relevant STM32Cube MCU package from the user repository. STM32CubeMX repository folder is specified in the Help > Updater settings menu.
- 2. It copies from the firmware package, the relevant files in *Drivers/CMSIS* and *Drivers/STM32F4_HAL_Driver* folders and in the *Middleware* folder if a middleware was selected.
- 3. It generates the initialization C code (.c/.h files) corresponding to the user MCU configuration and stores it in the *Inc* and *Src* folders. By default, the following files are included:
 - stm32f4xx_hal_conf.h file: this file defines the enabled HAL modules and sets some parameters (e.g. External High Speed oscillator frequency) to predefined default values or according to user configuration (clock tree).
 - stm32f4xx_hal_msp.c (MSP = MCU Support package): this file defines all initialization functions to configure the peripheral instances according to the user configuration (pin allocation, enabling of clock, use of DMA and Interrupts).
 - main.c is in charge of:

Resetting the MCU to a known state by calling the *HAL_init()* function that resets all peripherals, initializes the flash memory interface and the SysTick.

Configuring and initializing the system clock.

Configuring and initializing the GPIOs that are not used by peripherals.

Defining and calling, for each configured peripheral, a peripheral initialization function that defines a handle structure that will be passed to the corresponding peripheral *HAL init* function which in turn will call the peripheral HAL MSP initialization function. Note that when LwIP (respectively USB) middleware is used, the initialization C code for the underlying Ethernet (respectively USB peripheral) is moved from main.c to LwIP (respectively USB) initialization C code itself.

- main.h file:

This file contains the define statements corresponding to the pin labels set from the **Pinout** tab, as well as the user project constants added from the **Configuration** tab (refer to *Figure 456* and *Figure 457* for examples):

#define	MyTimeOut	10
#define	LD4_Pin	GPIO_PIN_12
#define	LD4_GPI0_Port	GPIOD
#define	LD3_Pin	GPIO_PIN_13
#define	LD3_GPI0_Port	GPIOD
#define	LD5_Pin	GPIO_PIN_14
#define	LD5_GPI0_Port	GPIOD
#define	LD6_Pin	GPIO_PIN_15
#define	LD6_GPI0_Port	GPIOD



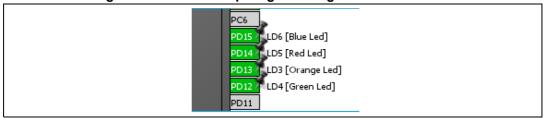


Figure 456. Labels for pins generating define statements



Co	onfiguration
Reset Configuration	
🥺 NVIC Settings 🛛 📀 DMA Settings	😔 GPIO Settings
🥺 Parameter Settings	🗢 User Constants
Search Constants Search (CrtI+F)	add remove
Constant Name	Constant Value
TimeOut	10

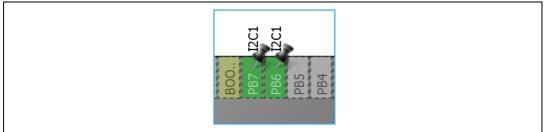
In case of duplicate labels, a unique suffix, consisting of the pin port letter and the pin index number, is added and used for the generation of the associated define statements.

In the example of a duplicate I2C1 labels shown in *Figure 458*, the code generation produces the following code, keeping the I2C1 label on the original port B pin 6 define statements and adding B7 suffix on pin 7 define statements:

#define	I2C1_Pin	GPIO_PIN_6
#define	I2C1_GPI0_Port	GPIOB
#define	I2C1B7_Pin	GPIO_PIN_7
#define	I2C1B7_GPI0_Port	GPIOB



Figure 458. Duplicate labels



In order for the generated project to compile, define statements shall follow strict naming conventions. They shall start with a letter or an underscore as well as the corresponding label. In addition, they shall not include any special character such as minus sign, parenthesis or brackets. Any special character within the label is replaced by an underscore in the define name.

If the label contains character strings between "[]" or "()", only the first string listed is used for the define name. As an example, the label "**LD6** [Blue Led]" corresponds the following define statements:

#define LD6_Pin GPI0_PIN_15

#define LD6_GPI0_Port GPIOD

The define statements are used to configure the GPIOs in the generated initialization code. In the following example, the initialization of the pins labeled *Audio_RST_Pin* and *LD4_Pin* is done using the corresponding define statements:

```
/*Configure GPIO pins : LD4_Pin Audio_RST_Pin */
GPIO_InitStruct.Pin = LD4_Pin | Audio_RST_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_LOW;
HAL_GPIO_Init(GPIOD, &GPIO_InitStruct);
```

4. Finally it generates a *Projects* folder that contains the toolchain specific files that match the user project settings. Double-clicking the IDE specific project file launches the IDE and loads the project ready to be edited, built and debugged.

6.2 STM32Cube code generation using Low Layer drivers

For all STM32 series except STM32H7 and STM32P1, STM32CubeMX allows the user to generate peripheral initialization code based either on the peripheral HAL driver or on the peripheral Low Layer (LL) driver.

The choice is made through the Project Manager view (see Section 4.11.3: Advanced Settings tab).

The LL drivers are available only for the peripherals which require an optimized access and do not have a complex software configuration. The LL services allow performing atomic operations by changing the relevant peripheral registers content:

- Examples of supported peripherals: RCC, ADC, GPIO, I2C, SPI, TIM, USART,...
- Examples of peripherals not supported by LL drivers: USB, SDMMC, FSMC.



The LL drivers are available within the STM32CubeL4 package:

- They are located next to the HAL drivers (stm32l4_hal_<peripheral_name>) within the *Inc* and *Src* directory of the STM32Cube_FW_L4_V1.6\Drivers\STM32L4xx_HAL_Driver folder.
- They can be easily recognizable by their naming convention: stm32l4_II_<peripheral_name>

For more details on HAL and LL drivers refer to the *STM32L4 HAL and Low-layer drivers* user manual (UM1884).

As the decision to use LL or HAL drivers is made on a peripheral basis, the user can mix both HAL and LL drivers within the same project.

The following tables shows the main differences between the three possible STM32CubeMX project generation options: HAL-only, LL-only, and mix of HAL and LL code.

Table 21. LL versus HAL code generation: drivers included in STM32CubeMX projects

Project configuration and drivers to be included	HAL only	LL only	Mix of HAL and LL	Comments
CMSIS	Yes	Yes	Yes	-
STM32xxx_HAL_Driver	Only HAL driver files	Only LL driver files	Mix of HAL and LL driver files	Only the driver files required for a given configuration (selection of peripherals) are copied when the project settings option is set to "Copy only the necessary files". Otherwise ("all used libraries" option) the complete set of driver files is copied.

Table 22. LL versus HAL code generation: STM32CubeMX generated header files

Generated header files	HAL only	LL only	Mix of HAL and LL	Comments
main.h	Yes	Yes	Yes	This file contains the include statements and the generated define statements for user constants (GPIO labels and user constants).
stm32xxx_hal_conf.h	Yes	No	Yes	This file enables the HAL modules necessary to the project.
stm32xxx_it.h	Yes	Yes	Yes	Header file for interrupt handlers
stm32xx_assert.h	No	Yes	Yes	This file contains the assert macros and the functions used for checking function parameters.



Table 23. LE VEISUS TIAL. 3 TM32CUDEMIX generated source mes							
Generated source files	HAL only	LL only	Mix of HAL and LL	Comments			
main.c	Yes	Yes	Yes	Contains the main functions and, optionally, STM32CubeMX generated functions.			
stm32xxx_hal_msp.c	Yes	No	Yes	Contains the following functions: – HAL_MspInit – for peripherals using HAL drivers: HAL_ <peripheral>_MspInit, HAL_<peripheral>_MspDeInit, These functions are available only for the peripherals that use HAL drivers.</peripheral></peripheral>			
stm32xxx_it.c	Yes	Yes	Yes	Source file for interrupt handlers			

Table 23. LL versus HAL: STM32CubeMX generated source files

Table 24. LL versus HAL: STM32CubeMX generated functions and function calls

Generated source files	HAL only	LL only	Mix of HAL and LL	Comments
Hal_init()	Called in main.c	Not used	Called in main.c	 This file performs the following functions: Configuration of flash memory prefetch and instruction and data caches Selection of the SysTick timer as timebase source Setting of NVIC group priority MCU low-level initialization.
Hal_msp_init()	Generated in stm32xxx_hal_msp.c and called by HAL_init()	Not used	Generated in stm32xxx_hal_msp.c And called by HAL_init()	This function performs the peripheral resources configuration ⁽¹⁾ .
MX_ <peripheral>_Init()</peripheral>	[1]: Peripheral configuration and call to HAL_ <peripheral>_Init()</peripheral>	[2]: Peripheral and peripheral resource configuration ⁽¹⁾ using LL functions Call to LL_Peripheral_Init()	 When HAL driver is selected for the <peripheral>, function generation and calls are done following [1]: Peripheral configuration and call to HAL_<peripheral>_In it()</peripheral></peripheral> When LL driver selected for the <peripheral>, function generation and calls are done following [2]: Peripheral and peripheral and peripheral resource configuration using LL functions</peripheral> 	This file takes care of the peripherals configuration. When the LL driver is selected for the <peripheral>, it also performs the peripheral resources configuration⁽¹⁾.</peripheral>



Generated source files	HAL only	LL only	Mix of HAL and LL	Comments
HAL_ <peripheral> _MspInit()</peripheral>	[3]: Generated in stm32xxx_hal_msp.c when HAL driver selected for the <peripheral></peripheral>	Not used	Only HAL driver can be selected for the <peripheral>: function generation and calls are done following [3]: Generated in stm32xxx_hal_msp.c when HAL driver selected for the <peripheral></peripheral></peripheral>	Peripheral resources configuration ⁽¹⁾
HAL_ <peripheral> _MspDeInit()</peripheral>	[4]: Generated in stm32xxx_hal_msp.c when HAL driver selected for the <peripheral></peripheral>	Not used	Only HAL driver can be selected for the <peripheral>: function generation and calls are done following [4]: <i>Generated in</i> <i>stm32xxx_hal_msp.c</i> <i>when HAL driver</i> <i>selected for the</i> <<i>Peripheral</i>></peripheral>	This function can be used to free peripheral resources.

Table 24. LL versus HAL: STM32CubeMX generated functions and function calls (continued)

Peripheral resources include:

 peripheral clock
 pinout configuration (GPIOs)
 peripheral DMA requests
 peripheral Interrupt requests and priorities.



```
USART Peripheral initialization - HAL-based
void MX_USART1_UART_Init(void)
{
                                             Peripheral Configuration
  huart1.Instance = USART1;
  huart1.Init.BaudRate = 115200;
  huart1.Init.WordLength = UART_WORDLENGTH_7B;
 huart1.Init.StopBits = UART STOPBITS 1;
  . . .
  if (HAL_UART_Init(shuart1) != HAL_OK)
  {
    Error_Handler();
  }
}
void HAL_UART_MspInit(UART_HandleTypeDef* uartHandle)
{
                                       Peripheral Resources Configuration
  GPI0_InitTypeDef GPI0_InitStruct;
  if (uartHandle->Instance==USART1)
  {
    /* Peripheral clock enable */
     _HAL_RCC_USART1_CLK_ENABLE();
    /* USART1 GPIO Configuration */
   GPIO_InitStruct.Pin = GPIO_PIN_10;
   GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
    GPIO_InitStruct.Pull = GPIO_PULLUP;
    . . .
    HAL GPIO Init(GPIOB, &GPIO InitStruct);
  }
3
void HAL_UART_MspDeInit(UART_HandleTypeDef* uartHandle)
{
                                        Peripheral Resources Release
  if (uartHandle->Instance==USART1)
  {
      /* Peripheral clock disable */
     _HAL_RCC_USART1_CLK_DISABLE();
    /* USART1 GPIO Configuration */
    HAL_GPIO_DeInit(GPIOA, GPIO_PIN_10);
    HAL_GPI0_DeInit(GPI0B, GPI0_PIN_6);
  }
```

Figure 459. HAL-based peripheral initialization: usart.c code snippet



```
Figure 460. LL-based peripheral initialization: usart.c code snippet
```

```
USART Peripheral Initialization using LL drivers
void MX_USART1_UART_Init(void)
 LL_USART_InitTypeDef USART_InitStruct;
 LL_GPIO_InitTypeDef GPIO_InitStruct;
  /* Peripheral clock enable */
 LL APB2 GRP1 EnableClock(LL APB2 GRP1 PERIPH USART1);
                                     Peripheral Resources Configuration
    /**USART1 GPIO Configuration
    PA10
          ----> USART1 RX
            ----> USART1_TX
   PB6
    */
 GPIO_InitStruct.Pin = LL_GPIO_PIN_10;
 GPIO_InitStruct.Mode = LL_GPIO_MODE_ALTERNATE;
 GPIO_InitStruct.Speed = LL_GPIO_SPEED_FREQ_VERY_HIGH;
 GPIO_InitStruct.Pull = LL_GPIO_PULL_UP;
 GPIO InitStruct.Alternate = LL GPIO AF 7;
 LL_GPIO_Init(GPIOA, &GPIO_InitStruct);
 GPIO_InitStruct.Pin = LL_GPIO_PIN_6;
 GPIO_InitStruct.Mode = LL_GPIO_MODE_ALTERNATE;
 GPIO_InitStruct.Speed = LL_GPIO_SPEED_FREQ_VERY_HIGH;
 GPIO_InitStruct.Pull = LL_GPIO_PULL_UP;
 GPIO_InitStruct.Alternate = LL_GPIO_AF_7;
 LL_GPIO_Init(GPIOB, &GPIO_InitStruct);
                                            Peripheral Configuration
 USART_InitStruct.BaudRate = 115200;
 USART_InitStruct.DataWidth = LL_USART_DATAWIDTH_7B;
 USART_InitStruct.StopBits = LL_USART_STOPBITS_1;
 USART_InitStruct.Parity = LL_USART_PARITY_NONE;
 USART InitStruct.TransferDirection = LL USART DIRECTION TX RX;
 USART_InitStruct.HardwareFlowControl = LL_USART_HWCONTROL_NONE;
 USART_InitStruct.OverSampling = LL_USART_OVERSAMPLING_16;
 LL_USART_Init(USART1, &USART_InitStruct);
 LL_USART_ConfigAsyncMode(USART1);
```

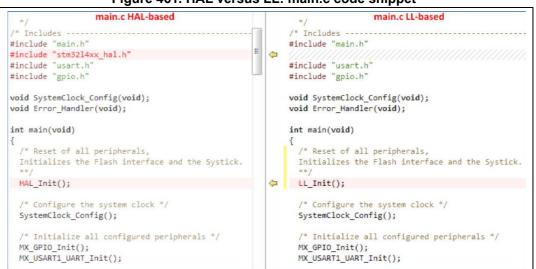


Figure 461. HAL versus LL: main.c code snippet



6.3 Custom code generation

STM32CubeMX supports custom code generation by means of a FreeMarker template engine (see http://www.freemarker.org).

6.3.1 STM32CubeMX data model for FreeMarker user templates

STM32CubeMX can generate a custom code based on a FreeMarker template file (.ftl extension) for any of the following MCU configuration information:

- List of MCU peripherals used by the user configuration
- List of parameters values for those peripherals
- List of resources used by these peripherals: GPIO, DMA requests and interrupts.

The user template file must be compatible with STM32CubeMX data model. This means that the template must start with the following lines:

```
[#ft1]
[#list configs as dt]
[#assign data = dt]
[#assign peripheralParams =dt.peripheralParams]
[#assign peripheralGPIOParams =dt.peripheralGPIOParams]
[#assign usedIPs =dt.usedIPs]
and end with
```

[/#list]

A sample template file is provided for guidance (see Figure 462).

STM32CubeMX will also generate user-specific code if any is available within the template.

As shown in the below example, when the sample template is used, the ftl commands are provided as comments next to the data they have generated:

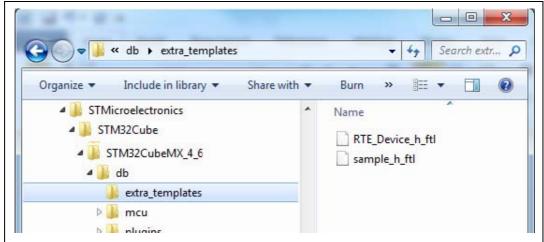
FreeMarker command in template:

```
${peripheralParams.get("RCC").get("LSI_VALUE")}
```

Resulting generated code:

```
LSI_VALUE : 32000 [peripheralParams.get("RCC").get("LSI_VALUE")]
```

Figure 462. Default content of the extra_templates folder





6.3.2 Saving and selecting user templates

The user can either place the FreeMarker template files under STM32CubeMX installation path within the db/extra_templates folder or in any other folder.

Then for a given project, the user will select the template files relevant for its project via the **Template Settings** window accessible from the Code Generator Tab in the **Project Manager** view menu (see Section 4.11)

6.3.3 Custom code generation

To generate custom code, the user must place the FreeMarker template file under STM32CubeMX installation path within the db/extra_templates folder (see *Figure 463*).

The template filename must follow the naming convention <user filename>_<file extension>.ftl in order to generate the corresponding custom file as <user filename>.<file extension>.

By default, the custom file is generated in the user project root folder, next to the .ioc file (see *Figure 464*).

To generate the custom code in a different folder, the user shall match the destination folder tree structure in the extra_template folder (see *Figure 465*).

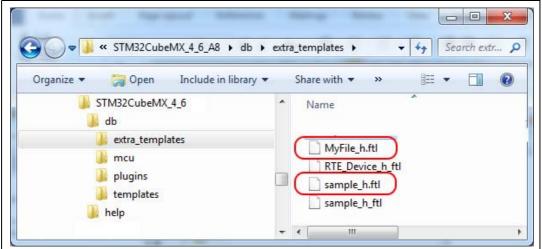


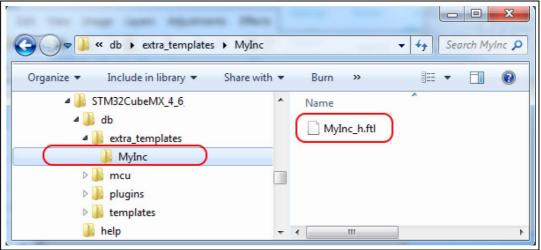
Figure 463. extra_templates folder with user templates



Organize Include in library Drivers Inc MyInc Projects Src Src MyFile.h MyFile.h Sample.h	🚰 💭 🗢 🚺 « Custom Code project 🕨 CustomC	od	eGen ►
Inc I	Organize 👻 Include in library 👻 Share with	•	
	 Drivers Inc MyInc Projects 		Drivers 1 Inc 1 MyInc 1 Projects 1 Src 1 .mxproject 1 CustomCodeGen.ioc 1 MyFile.h 1

Figure 464. Project root folder with corresponding custom generated files







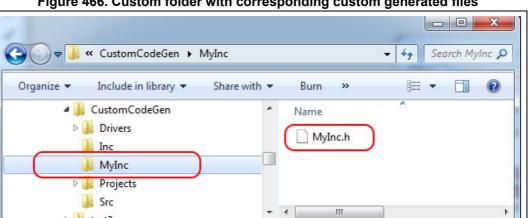


Figure 466. Custom folder with corresponding custom generated files

Additional settings for C project generation 6.4

STM32CubeMX allows specifying additional project settings through the .extSettings file. This file must be placed in the same project folder and at the same level as the .ioc file.

As an example, additional settings can be used when external tools call STM32CubeMX to generate the project and require specific project settings.

Possible entries and syntax

All entries are optional. They are organized under the followings three categories: ProjectFiles, Groups or Others.

[ProjectFiles]: section where to specify additional include directories

Syntax

HeaderPath = <include directory 1 path>;< include directory 2 path > Example

HeaderPath=../../IIR_Filter_int32/Inc ;

[Groups]: section where to create new groups of files and/or add files to a group Syntax

<Group name> = <file pathname1>;< file pathname2>

Example

Doc=\$ PROJ_DIR\$\..\readme.txt

```
Lib=C:\libraries\mylib1.lib; C:\libraries\mylib2.lib;
Drivers/BSP/MyRefBoard = C:\MyRefBoard\BSP\board_init.c;
C:\MyRefBoard\BSP\board_init.h;
```

- [Others]: section where to enable HAL modules and/or specify preprocessor define statements
 - Enabling preprocessor define statements
 - Preprocessor define statements can be specified using the following syntax after the [Others] line:

Svntax

Define = <define1_name>;<define2_name>



Example

Define= USE_STM32F429I_DISCO

Enabling HAL modules in generated stm32f4xx_hal_conf.h HAL modules can be enabled using the following syntax after the [Others] line: Syntax

HALModule = <ModuleName1>; <ModuleName1>;

Example

HALModule=12S;12C

.extSettings file example and generated outcomes

For the purpose of the example, a new project is created by selecting the STM32F429I-DISCO board from STM32CubeMX board selector. The EWARM toolchain is selected in the Project tab of the **Project Manager** view. The project is saved as *MyF429IDiscoProject*. In the project folder, next to the generated .ioc file, a .extSettings text file is placed with the following contents:

[Groups]

Drivers/BSP/STM32F429IDISCO=C:\Users\frq09031\STM32Cube\Repository\STM3 2Cube_FW_F4_V1.14.0\Drivers\BSP\STM32F429I-Discovery\stm32f429i_discovery.c; C:\Users\frq09031\STM32Cube\Repository\STM32Cube_FW_F4_V1.14.0\Drivers\ BSP\STM32F429I-Discovery\stm32f429i_discovery.h Lib=C:\Users\frq09031\STM32Cube\Repository\STM32Cube_FW_F4_V1.14.0\ Middlewares\Third_Party\FreeRTOS\Source\portable\IAR\ARM_CM4F\portasm.s Doc=\$PROJ_DIR\$\..\readme.txt

[Others]

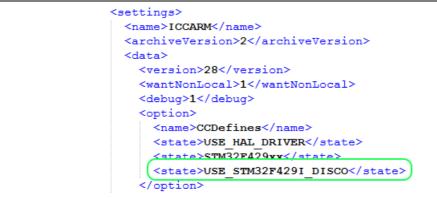
Define = USE_ STM32F429I_DISCO HALModule = UART;SPI

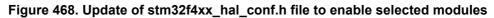
Upon project generation, the presence of this .extSettings file triggers the update of:

- the project MyF429IDiscoProject.ewp file in EWARM folder (see Figure 467)
- the stm32f4xx_hal_conf.h file in the project Inc folder (see *Figure 468*)
- the project view within EWARM user interface as shown in *Figure 469* and *Figure 470*.



Figure 467. Update of the project .ewp file (EWARM IDE) for preprocessor define statements





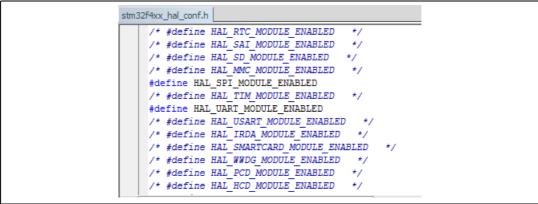


Figure 469. New groups and new files added to groups in EWARM IDE

Files	82
🗆 🗇 MyF429IDiscoProject - MyF429ID	×
- 🕀 🗀 Application	
📔 🖵 🖹 readme.txt	
📙 🔚 🖵 STM32F429IDISCO	
H 🕂 🕂 🔂 stm32f429i_discovery.c	
📋 📙 🦶 📓 stm32f429i_discovery.h 🚽	
📙 🛏 🗀 STM32F4xx_HAL_Driver	
📙 4🕀 歸 portasm.s 🌙	
📙 🗀 Output	

Category: General Options	•	Multi-file Compilation
Static Analysis Runtime Checking		Discard Unused Publics
C/C++ Compiler	_	Language 1 Language 2 Code Optimizations Output List Preproce
Output Converter Custom Build Build Actions Linker Debugger Simulator Angel CADI	m	Ignore standard include directories Additional include directories: (one per line) SPROJ_DIR\$//Inc SPROJ_DIR\$//Drivers/STM32F4xx_HAL_Driver/Inc SPROJ_DIR\$//Drivers/STM32F4xx_HAL_Driver/Inc/Legacy SPROJ_DIR\$//Drivers/CMSIS/Device/ST/STM32F4xx/Include Preinclude file:
CMSIS DAP GDB Server IAR ROM-monitor I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor PE micro RDI		Defined symbols: (one per line) USE_HAL_DRIVER STM32F429xx USE_STM32F429I_DISCO

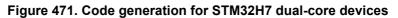
Figure 470. Preprocessor define statements in EWARM IDE



7 Code generation for dual-core MCUs (STM32H7 dual-core product lines only)

For working with Arm Cortex-M dual-core products, STM32CubeMX generates code for both cores automatically according to the context assignment and initializer choices made in the user interface (see *Section 4.8: Pinout & Configuration view for STM32H7 dual-core products* for details).

	Pinout & Co	onfigurat	ion			📜 CN	
				0		📜 CN	М7
Categories A->Z						🕨 📜 Co	ommon
÷ HRTIM	Cortex-M7	Cortex-M4	Initializer	r		📜 Dr	ivers
12C1	\checkmark		Cortex-M			📜 EV	VARM
12C2	\checkmark	\checkmark	Cortex-M Cortex-M7	4 ~		🗋 .m	xproject
12C4						MX ST	M32H747 dualcore project1.ioc
12S1 12S2							p.o,eeee
<u>File Edit Viev</u>	v <u>P</u> roject	<u>S</u> T-Link	<u>T</u> ools <u>W</u> i	indow <u>H</u> e	lp		
File Edit Viev				indow <u>H</u> e	lp	- < Q	
Eile Edit View				indow <u>H</u> e	lp — д :	- < Q	Select the context to
Vorkspace				indow <u>H</u> e		• < Q	Select the context to
Workspace STM32H747_dua STM32H747_dua	Icore_project	1_CM4		indow <u>H</u> e			Select the context to work with
Workspace STM32H747_dua STM32H747_dua STM32H747_dua	Icore_project1	1_CM4 1_CM4 1_CM4 1_CM7	5 C		▼ ‡ :	< Q	
Workspace STM32H747_dua STM32H747_dua STM32H747_dua C STM32H747_dua	Icore_project1 Icore_project1 Icore_project1 Icore_project1 I747_dualo	1_CM4 1_CM4 1_CM4 1_CM7	5 C		▼ ‡ :		
Workspace STM32H747_dua STM32H747_dua STM32H747_dua	licore_project1 licore_project1 licore_project1 1747_dualo ation	1_CM4 1_CM4 1_CM4 1_CM7	5 C		▼ ‡ :		
Workspace STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua	Icore_project Icore_project Icore_project Icore_project I747_duato ation ARM itartup_stm	1_CM4 1_CM4 1_CM7 core_pr 32h747x	oject1 - ST x_CM4.s		▼ ‡ :	 < Q × 	
Vorkspace STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm32H747_stm3	Icore_project Icore_project Icore_project Icore_project I747_duato ation ARM tartup_stm: tartup_stm:	1_CM4 1_CM4 1_CM7 core_pr 32h747x	oject1 - ST x_CM4.s		 		
Vorkspace STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H	Icore_project Icore_project Icore_project Itation ARM ARM tartup_stm atartup_stm	1_CM4 1_CM4 1_CM7 core_pr 32h747x	oject1 - ST x_CM4.s		. ✓		
Vorkspace STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H	Icore_project Icore_project Icore_project It747_duato ation ARM tartup_stm startup_stm s	1_CM4 1_CM4 1_CM7 core_pr 32h747x	oject1 - ST x_CM4.s		 		
Vorkspace STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H747_dua STM32H STM3H STM	Icore_project Icore_project Icore_project It747_duato ation ARM tartup_stm startup_stm s	1_CM4 1_CM4 1_CM7 Core_pr 32h747x 32h747x	oject1 - ST x_CM4.s x_CM7.s		 		



Generated initialization code

The code is generated in CM4, CM7 and Common folders. The Common folder holds the system_stm32h7xx.c, that contains the clock tree settings.

When a peripheral or middleware is assigned to both contexts, the function MX_<name>_init will be generated for both contexts but will be called only from the initializer side.



Generated startup and linker files

Each configuration (_M4 or _M7) of the project shall come with a startup file and a linker file, each suffixed with _M4 or _M7 respectively.

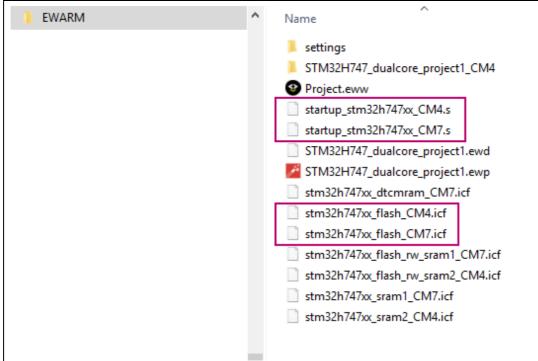


Figure 472. Startup and linker files for STM32H7 dual-core devices

Generated boot mode code

STM32CubeMX supports only one mode of boot for now, where both ARM Cortex-M cores boot at once.

The other boot modes will be introduced later as a project option in the project manager view:

- Arm Cortex-M7 core booting, Arm Cortex-M4 gated
- Arm Cortex-M4 core booting, Arm Cortex-M7 gated
- A first core booting executing from flash, loads the second core code to the SRAM then enables the second core to boot.

STM32CubeMX uses template files delivered with STM32CubeH7 MCU packages as reference.



8 Code generation with TrustZone[®] enabled (STM32L5 series only)

In STM32CubeMX project manager view, all project generation options remain available.

However, the choice of toolchains is limited to the IDEs/compilers supporting the $\mbox{Cortex}^{\mbox{$\mathbb R$}}\mbox{-}M33$ core:

- EWARM v8.32 or higher
- MDK-ARM v5.27 or higher (ARM compiler 6)
- STM32CubeIDE (GCC v4.2 or higher)
- Makefile (GCC v4.2 or higher)

Upon product selection, STM32CubeMX requires to choose between enabling $\mathsf{TrustZone}^{\texttt{®}}$ or not.

- When TrustZone[®] is enabled, STM32CubeMX generates two C projects: one secured and one non-secured. After compilation, two images are available for download, one for each context.
- When TrustZone[®] is disabled, STM32CubeMX generates a non-secured C project, as for other products not supporting it.

Specificities

When TrustZone[®] is enabled, the project generation must be adjusted to ensure that secure and nonsecure images can be built.

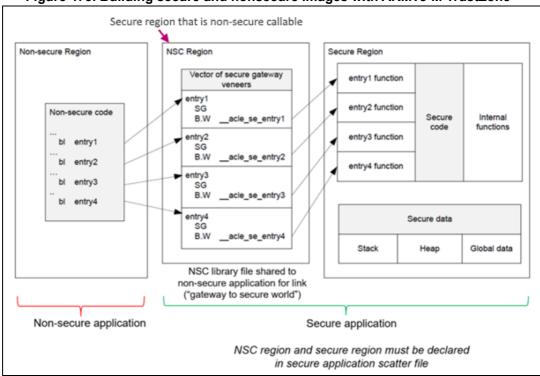


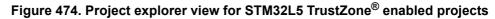
Figure 473. Building secure and nonsecure images with ARMv8-M TrustZone®



When TrustZone[®] is enabled for the project, STM32CubeMX generates three folders:

- NonSecure for nonsecure code
- Secure for secure code
- Secure_nsclib for nonsecure callable region

See *Figure* 474 (use TZ_BasicStructure_project_inCubeIDE.png) and *Figure* 475 (use STM32L5_STM32CubeMX_Project_settings_inCubeIDE.png).



Project Explorer 🛛 📄 🔄 🔽 🗖	j
✓	
✓ IDE stm32I5_TZ_BasicStructure project1 NonSecure (in NonSecure)	
> 🔊 Includes Secure_nsclib	
> C Drivers	
Src Secure_insc.in	
> 🖻 main.c	
stm32l5xx_hal_msp.c	
> ic stm32l5xx it.c	
> c syscalls.c	
> c sysmem.c	
> 💽 system_stm32l5∞_ns.c 🗡	
V 🤔 Startup	
S startup_stm32l562cetx.s	
V 🍃 Inc	
📓 main.h	
stm32I5xx_hal_conf.h	
b stm32l5∞_it.h	
🗟 STM32L562CETX_FLASH.Id 🧲	
🗟 STM32L562CETX_RAM.Id 🗡	
IDE stm32I5_TZ_BasicStructure_project1_Secure (in Secure)	
> 🔊 Includes	
> 🚑 Drivers	
V 😕 Src	
> 🖻 main.c	
> 🖻 secure_nsc.c 🚩	
> 🚺 stm3215xx_hal_msp.c	
> 🖻 stm3215xx_it.c	
> c syscalls.c	
> 🖻 sysmem.c	
> 🖻 system_stm32l5xx_s.c 🕊	
🗸 🔁 Startup	
> is startup_stm32l562cetx.s	
V 🗁 Inc	
📓 main.h	
📓 partition_stm32I562xx.h 🚩	
stm32l5xx_hal_conf.h	
li⊨ stm32l5∞_it.h	
STM32L562CETX_FLASH.Id	
🗟 STM32L562CETX_RAM.Id 🗡	



Figure 475. Project settings for STM32CubeIDE toolchain

Project Name	
stm32l5_TZ_BasicStructure_p	roject1
Project Location	
C:\STM32CubeMX_Projects	
Application Structure	
Basic	✓ □ Do not generate the main()
Toolchain Folder Location	
C:\STM32CubeMX_Projects\st	tm32I5_TZ_BasicStructure_project1\
Toolchain / IDE	×.
STM32CubeIDE	🛰 🔽 Generate Under Root

STM32CubeMX also generates specific files, detailed in Table 25.

File	Folder	Details
The product core secure/nonsecure partitioning .h "template" file Example: partition_stm32l552xx.h	Secure	Initial setup for secure/nonsecure zones for ARMCM33 based on CMSIS CORE V5.3.1 partition_ARMCM33.h Template. It initializes Security attribution unit (SAU) CTRL register, setup behavior of Sleep and Exception Handling, Floating Point Unit and Interrupt Target.
secure_nsc.h file	Secure_nsclib	Must be filled by the user with the list of nonsecure callable APIs. Templates are available as reference in STM32L5Cube embedded software package in Templates\TrustZone [®] \Secure_nsclib folders.
System_stm32l5xx_s.c	Secure	CMSIS Cortex-M33 device peripheral access layer system source file to be used in secure application when the system implements security.

Table 25. Files generated when TrustZone[®] is enabled



File	Folder	Details
System_stm32l5xx_ns.c	NonSecure	CMSIS Cortex-M33 device peripheral access layer system source file to be used in nonsecure application when the system implements security.
STM32L562CETX_FLASH STM32L562CETX_RAM or STM32L552CETX_FLASH STM32L552CETX_RAM	Secure, NonSecure	Linker files for the secure and nonsecure memory layouts. File extensions and naming conventions: – .icf (EWARM) – .sct (MDK-ARM), or – .ld (GCC compiler toolchains)

Table 25. Files generated when TrustZone[®] is enabled (continued)



9 Device tree generation (STM32MPUs only)

The Device tree in Linux is used to provide a way to describe non-discoverable hardware. STMicroelectronics is widely using the device tree for all the platform configuration data, including DDR configuration.

Linux developers can manually edit device tree source files (dts), but as an alternative STM32CubeMX offers a partial device-tree generation service to reduce effort and to ease new comers. STM32CubeMX intends to generate partially device trees corresponding to board level configuration. Partial means that the entire (board level) device-trees are not generated, but only main sections that usually imply huge efforts and can cause compilation errors and dysfunction:

- folders structure and files to folders distribution
- dtsi and headers inclusions
- pinCtrl and clocks generation
- System-On-Chip device nodes positioning
- multi-core related configurations (Etzpc binding, resources manager binding, peripherals assignment)

9.1 Device tree overview

To run properly, any piece of software needs to get the hardware description of the platform on which it is executed, including the kind of CPU, the memory size and the pin configuration. OpenSTLinux firmware has put such non-discoverable hardware description in a separate binary, the device tree blob (dtb). The device tree blob is compiled from the device tree source files (dts) using the dtc compiler provided with the OpenSTLinux distribution.

The device tree structure consist of a board level file (.dts) that includes two device tree source include files (.dtsi): a soc level file and a –pinctrl file, that lists the pin muxing configurations.

The device tree structure is very close to C language multiple level structures with the "root" (/) being the highest level then "peripherals" being sub-nodes described further in the hierarchy (see figures 476, 477 and 478).

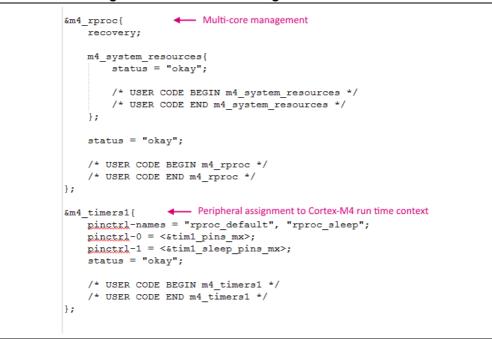
STM32CubeMX generation uses widely overloading mechanisms to complete or change some SOC devices definitions when user configurations require it.



```
System and Board information
 🔪 model = "STMicroelectronics custom STM32CubeMX board";
   compatible = "st,stm32mp157c-project2-mx", "st,stm32mp157";
   memory@c0000000 {
     ...
   };
  Full clock configuration
   clocks {
      clk_lsi: clk-lsi {
        #clock-cells = <0>;
        compatible = "fixed-clock";
        clock-frequency = <32000>;
        u-boot, dm-pre-reloc;
      };
      . . .
   };
}; /*root*/
u-boot, dm-pre-reloc;
   tim1_pins_mx: tim1_mx-0 {
     pins {
        bias-disable:
        drive-push-pull;
        slew-rate = <0>;
      };
   1:
};
```

Figure 476. STM32CubeMX generated DTS – Extract 1

Figure 477. STM32CubeMX generated DTS – Extract 2





```
&timers2{
                     Peripheral node structure with
                                                PinCtrl configuration
    status = "okay";
                                                 Status configuration
                                                User customization
    /* USER CODE BEGIN timers2 */
    /* USER CODE END timers2 */
    pwm {
        pinctrl-names = "default", "sleep";
        pinctrl-0 = <&tim2_pwm_pins_mx>;
        pinctrl-1 = <&tim2_pwm_sleep_pins_mx>;
        status = "okay";
        /* USER CODE BEGIN timers2_pwm */
        /* USER CODE END timers2_pwm */
    };
};
/* USER CODE BEGIN dts_addons */
/* USER CODE END dts_addons */
```

Figure 478. STM32CubeMX generated DTS – Extract 3

For more details refer to "Device Tree for Dummies" from Thomas Petazzoni, available on https://elinux.org.

For more information about STM32MPUs device tree specificities, refer to ST Wiki https://wiki.st.com/stm32mpu.

9.2 STM32CubeMX Device tree generation

For STM32MPUs, STM32CubeMX code generation feature has been extended to generate Device trees (DT) configuring the firmware.

DTS generation is accessible through the same

GENERATE CODE button.



The DT generation path can be configured from the Project Manager view, in the Advanced Settings tab, under OpenSTLinux Settings (see *Figure 479*). For each Device tree STM32CubeMX generates Device tree source (DTS) files.

	File	Window	Help	
Home > STM32MF	P151CAAx > STM32MP1_	Test1.ioc - Project Manager	\rangle	
Pinout &	Configuration	Clock Config	guration	Project Manager
Advanced Settings	Linux 4.19		STM32M Device Drivers Inc Src SW4ST	kerne

Figure 479. Project settings to configure Device tree path

The Device tree structure consists of:

- a complete clock-tree
- a complete pin control
- a complete multi-cores references definition
- a set of device nodes and sub-nodes
- user sections that can be filled to have complete and bootable Device trees (contents are not lost at next generation).

The generated DTS files reflect the user configuration, such as the assignment of peripherals to runtime contexts and boot loaders, or clock tree settings.

STM32CubeMX DT generation ensures the coherency between the different DTs. Additionally, it generates the DDR configuration file as part of the boot loader Device trees.

These files, along with the files they include, are compiled to create the device tree blob for the targeted firmware.

The STM32CubeMX Device tree structure depends upon the targeted firmware and, in a few cases, upon the OpenSTLinux manifest version and/or the MPU family. The structures are detailed in https://wiki.st.com/stm32mpu/wiki/Category:Platform_configuration.

The device tree nodes generated by STM32CubeMX can be completed by filling the user sections following the device tree bindings of the different firmware.

Note: To continue the process and learn how to use the generated files, see the dedicated Wiki pages for MPUs.



10 Support of additional software components using CMSIS-Pack standard

The CMSIS-Pack standard describes a delivery mechanism for software components, device parameters, and evaluation board support.

The XML-based package description (pdsc) file describes the content of a software pack (file collection). It includes source code, header files, software libraries, documentation and source code templates. A software pack consists of the complete file collection along with the pdsc file, shipped in ZIP-format. After installing a software pack, all the included software components are available to the development tools.

A software component is a collection of source modules, header and configuration files as well as libraries. Packs containing software components can also include example projects and user code templates.

Refer to http://www.keil.com website for more details.

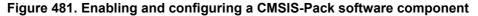
STM32CubeMX supports third-party and other STMicroelectronics embedded software solutions, delivered as software packs. STM32CubeMX enables to:

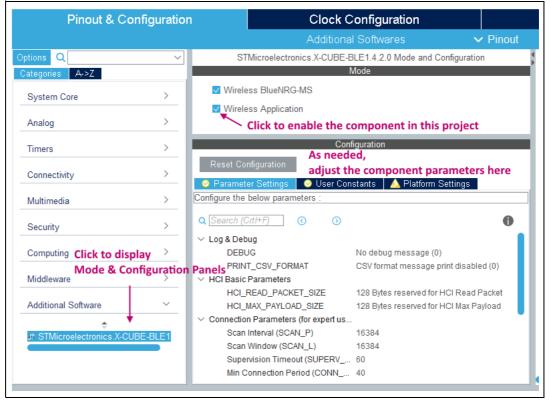
- 1. Install software packs and check for updates (see Section 3.4.5).
- 2. Select software components for the current project (see *Section 4.15*). Once this is done, the selected components appear in the tree view (see *Figure 480*).
- 3. Enable the software component from the tree view (see *Figure 481*). Use contextual help to get more details on the selection.
- 4. Configure software components (see *Figure 481*). This function is possible only for components coming with files in STM32CubeMX proprietary format.
- 5. Generate the C project for selected toolchains (see Figure 482).
 - a) Software components files are automatically copied to the project.
 - b) Software component configuration and initialization code are automatically generated. This function is possible only for components coming with files in STM32CubeMX proprietary format.



🛱 💊 🎁 > Show compone	nts for cont	ext: Cortex-M7	\sim		
Pack / Bundle / Component	Status	Version	Selection		
> RoweBots.I-CUBE-UNISONRTOS		5.5.0-4 ڬ 😂	Install		
> SEGGER.I-CUBE-embOS		1.2.0 😂			
> STMicroelectronics.X-CUBE-AI	1	6.0.0 ~			
> STMicroelectronics.X-CUBE-ALGOBUILD		1.2.0 ~			
> STMicroelectronics.X-CUBE-AZRTOS-H7	+	1.0.0			
 STMicroelectronics.X-CUBE-BLE1 	\odot	6.2.0 ~			
✓ Wireless BlueNRG-MS	\odot	5.1.0			
BlueNRG-MS / Controller	\odot				
BlueNRG-MS / HCI_TL	\odot		Basic	~	
BlueNRG-MS / HCI_TL_INTERFACE	\odot		UserBoard	~	
BlueNRG-MS / Utils	\odot				
Device BLE1_Applications	\odot	6.1.0			
Application	\odot		SensorDemoBLESensor	\sim	
> STMicroelectronics.X-CUBE-BLE2		3.2.0 ~			









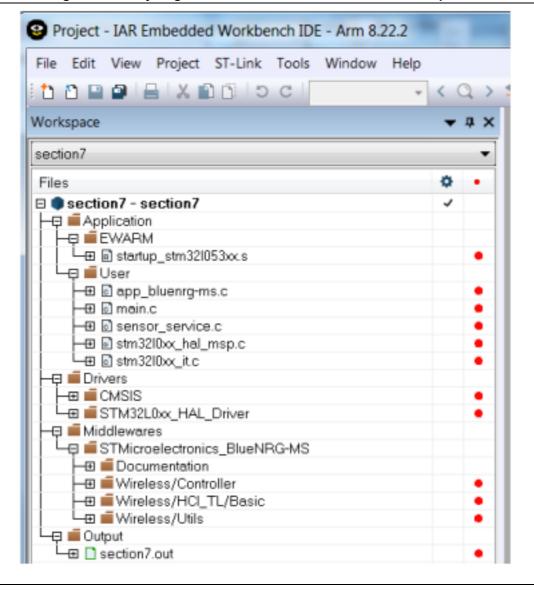


Figure 482. Project generated with CMSIS-Pack software component



11 Tutorial 1: From pinout to project C code generation using an MCU of the STM32F4 series

This section describes the configuration and C code generation process. It takes as an example a simple LED toggling application running on the STM32F4DISCOVERY board.

11.1 Creating a new STM32CubeMX project

- 1. Select **File > New project** from the main menu bar or **New project** from the Home page.
- Select the MCU/MPU Selector tab and filter down the STM32 portfolio by selecting STM32F4 as 'Series', STM32F407 as 'Lines', and LQFP100 as 'Package' (see *Figure 483*).
- 3. Select the STM32F407VGTx from the MCU list and click **OK**.

MPU Selector Board Selecto	or Example Selector	Cross Selector			
MPU Filters					
★ 🔂 🖻	3	Featu Block Diagr	Docs & Resour CAD I	Resour 📑 Datash	🖬 🕞 Start Proj
Commercial Part Number	~	STM32F4 Series			
۹	→ + -	STM32F407VGT6	High-performance foun and FPU, 1 Mbyte of Fla Accelerator, Ethernet, F	ash memory, 168 MHz	
STM32MP1		ACTIVE	Unit Price for 10kU (US\$) : 5.8519	-	
STM 32N6		Product is in mass production	Board: STM32F407G-DISC1	LOFP	100 14x14x1.4 mm
STM32U0		production	Board: STM32F407G-DISCT		
STM32U0				e high-performance Arm [®] Cor	ex [®] -M4 32-bit RISC core
		The STM32F405xx and	STM32F407xx family is based on th		
		The STM32F405xx and operating at a frequency	STM32F407xx family is based on th	core features a floating-point	unit (FPU) single precision
☐ STM 32U3 ☐ STM 32U5		The STM32F405xx and operating at a frequency	STM32F407xx family is based on th v of up to 168 MHz. The Cortex [®] -M4	core features a floating-point	unit (FPU) single precision
 □ STM 32U3 □ STM 32U5 □ STM 32WB 	v	The STM32F405xx and operating at a frequency which supports all Arm s	STM32F407xx family is based on th of up to 188 MHz. The Cortex [®] -M4 ingle-precision data-processing inst	core features a floating-point ructions and data types. It also	unit (FPU) single precision implements a full set of
STM 32U3 STM 32U5 STM 32WB STM 32F4 ×		The STM32F405xx and operating at a frequency which supports all Arm s	STM32F407xx family is based on th of up to 168 MHz. The Cortex [®] -N4 ingle-precision data-processing inst 	core features a floating-point ructions and data types. It also	unit (FPU) single precision pimplements a full set of the Expo 10 × Flash × Frequen
☐ STM32U3 ☐ STM32U5 ☐ STM32WB STM32F4 ×		The STM32F405xx and operating at a frequency which supports all Arm s MCUs/MPUs List: 12 items Commercial Part No \$2 STM32F407VET6TR STM32F407VET6TR	STM32F407xx family is based on th or up to 166 MHz. The Cortex [®] MM ingle-precision data-processing inst 	core features a floating-point ructions and data types. It also rd X Package X RAM X LOFP 100 192 kBytes LOFP 100 192 kBytes	unit (FPU) single precision implements a full set of the Expo 82 512 kBytes 168 MH 82 512 kBytes 168 MH
STM 32U3 STM 32U5 STM 32WB STM 32F4 ×	∥(1)	The STM32F405xx and operating at a frequency which supports all Arm s MCUs/MPUs List: 12 items Commercial Part No ☆ STM32F407VET6 ☆ STM32F407VET6TR ☆ STM32F407VET6TR	STM32F407xx family is based on th r of up to 168 MHz. The Cortex [®] -M4 ingle-precision data-processing inst 	core features a floating-point ructions and data types. It also rd X Package X RAM X LQFP 100 192 kBytes LQFP 100 192 kBytes Fdy7 LQFP 100 192 kBytes	unit (FPU) single precision implements a full set of the Expo to the
STM 32U3 STM 32U5 STM 32WB STM 32F4 × Jne Check/Uncheck A		The STM32F405xx and operating at a frequency which supports all Arms MCUs/MPUs List: 12 items Commercial Part No \$\$TM32F407VET6T \$\$TM32F407VET6T \$\$TM32F407VET6T6 \$\$TM32F407VGT6J\$	STM32F407xx family is based on th r of up to 168 MHz. The Cortex [®] -M4 ingle-precision data-processing inst STM32F4Active STM32F4Active STM32F4Active STM32F4Active STM32F4Active STM32F4Active	core features a floating-point ructions and data types. It also d X Package X RAM X LQFP 100 192 kBytes LQFP 100 192 kBytes LQFP 100 192 kBytes LQFP 100 192 kBytes	unit (FPU) single precision implements a full set of Expo 82 512 kBytes 168 MH 82 512 kBytes 168 MH 82 1024 kByt168 MH 82 1024 kByt168 MH
STM 32U3 STM 32U5 STM 32WB STM 32F4 ★ .ine Check/Uncheck AI STM 32F407/417	∥(1)	The STM32F405xx and operating at a frequency which supports all Arm s MCUs/MPUs List: 12 items Commercial Part No \$\$TM32F407VET6 \$\$TM32F407VET6 \$\$TM32F407VET6 \$\$TM32F407VGT6 \$\$TM32F407VGT6 \$\$TM32F407VGT6TR	STM32F407xx family is based on th or up to 166 MHz. The Cortex [®] M4 ingle-precision data-processing inst STM32F4Active STM32F4Active STM32F4Active STM32F4Active STM32F4Obsolete STM32F4Obsolete STM32F4Active	core features a floating-point ructions and data types. It also LOFP 100 192 kBytes LOFP 100 192 kBytes	unit (FPU) single precision implements a full set of 100 × Fast > Frequen 82 512 KBytes 168 MH 82 1024 kByt 168 MH 82 1024 kByt168 MH 82 1024 kByt168 MH
STM 32U3 STM 32U5 STM 32WB STM 32F4 × Jne Check/Uncheck A	∥(1)	The STM32F405xx and operating at a frequency which supports all Arm s MCUs/MPUs List: 12 items Commercial Part No \$TM32F407VET6 \$TM32F407VET6 \$TM32F407VET6 \$TM32F407VET6 \$TM32F407VGT6 \$TM32F407VGT6	STM32F407xx family is based on th rof up to 168 MHz. The Cortex [®] -M4 ingle-precision data-processing inst STM 32F4Active STM 32F4Active STM 32F4Active STM 32F4Active STM 32F4Active STM 32F4Active STM 32F4Active STM 32F4Active STM 32F4Active	core features a floating-point ructions and data types. It also devices and data types. It also LGPP 100 192 kBytes LGPP 100 192 kBytes LGPP 100 192 kBytes LGPP 100 192 kBytes LGPP 100 192 kBytes	unit (FPU) single precision implements a full set of 2 Expo 82 512 kBytes 168 MH 82 1024 kBytes 168 MH
STM 32U3 STM 32U5 STM 32WB STM 32F4 ★ .ine Check/Uncheck AI STM 32F407/417	∥(1)	The STM32F405xx and operating at a frequency which supports all Arm s MCUs/MPUs List: 12 items Commercial Part No \$\$TM32F407VET6 \$\$TM32F407VET6 \$\$TM32F407VET6 \$\$TM32F407VGT6 \$\$TM32F407VGT6 \$\$TM32F407VGT6TR	STM32F407xx family is based on th rof up to 168 MHz. The Cortex [®] -M4 ingle-precision data-processing inst STM 32F4Active STM 32F4Active STM 32F4Active STM 32F4Active STM 32F4Active STM 32F4Active STM 32F4Active STM 32F4Active STM 32F4Active	core features a floating-point ructions and data types. It also LOFP 100 192 kBytes LOFP 100 192 kBytes	unit (FPU) single precision implements a full set of Expo 82 512 kBytes 168 MH 82 512 kBytes 168 MH 82 1024 kByt 168 MH

Figure 483. MCU selection

STM32CubeMX views are then populated with the selected MCU database (*Figure 484*). Optionally, remove the MCUs Selection bottom window by deselecting **Window > Outputs** submenu (see *Figure 485*).



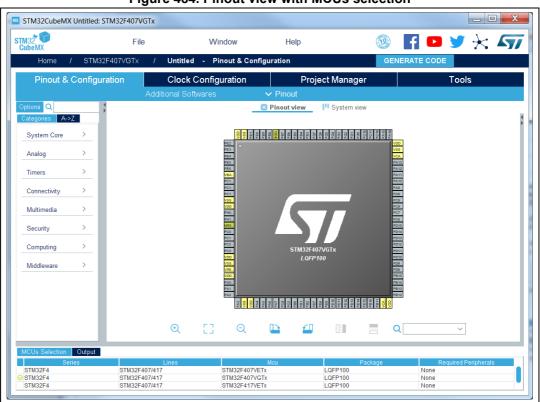
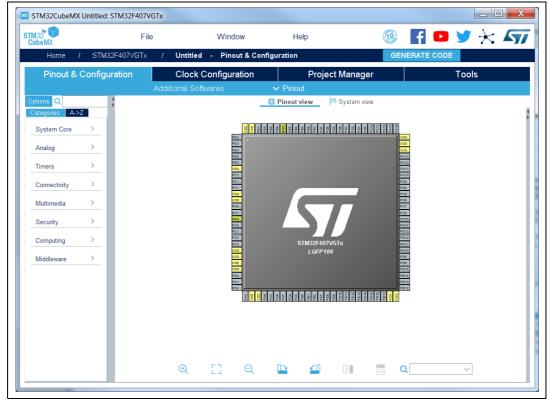


Figure 484. Pinout view with MCUs selection

Figure 485. Pinout view without MCUs selection window



UM1718 Rev 47



11.2 Configuring the MCU pinout

For a detailed description of menus, advanced actions and conflict resolutions, refer to *Section 4* and *Appendix A*.

- 1. By default, STM32CubeMX shows the **Pinout** view.
- 2. By default, Keep Current Signals Placement is unchecked allowing STM32CubeMX to move the peripheral functions around and to find the optimal pin allocation, that is the one that accommodates the maximum number of peripheral modes.

Since the MCU pin configurations must match the STM32F4DISCOVERY board, enable Keep Current Signals Placement for STM32CubeMX to maintain the peripheral function allocation (mapping) to a given pin.

This setting is saved as a user preference in order to be restored when reopening the tool or when loading another project.

- 3. Select the required peripherals and peripheral modes:
 - a) Configure the GPIO to output the signal on the STM32F4DISCOVERY green LED by right-clicking PD12 from the **Pinout** view, then select GPIO_output:

PD11 PD10

Figure 486. GPIO pin configuration



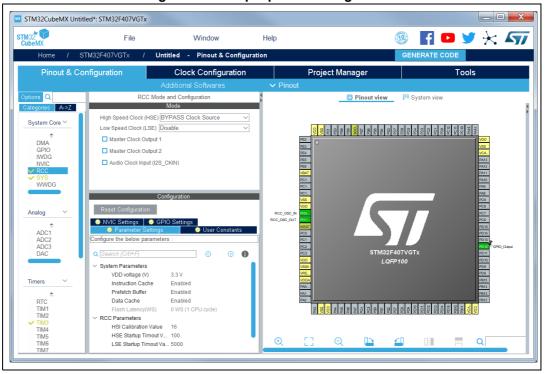
b) Enable a timer to be used as timebase for toggling the LED. This is done by selecting Internal Clock as TIM3 clock source from the peripheral tree (see *Figure 487*).

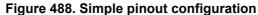
Pinout & Con	figuration	Clock Configuration		Proje
		Additional Softwares	🗸 Pir	nout
Options Q		TIM3 Mode and Configuration		
Categories A->Z		Mode		
Analog >	Slave Mode Dis	sable	\sim	
	Trigger Source	Disable	\sim	
Timers ~	Clock Source	Internal Clock	\sim	
÷	Channel1 Disa	ible	\sim	
RTC TIM1	Channel2 Disa	ible	\sim	
TIM2	Channel3 Disa	ble	\sim	
V TIM3 TIM4	Channel4 Disa	ble	\sim	
TIM5	Combined Cha	nnels Disable	\sim	
TIM6 TIM7	Use ETR as	s Clearing Source		
TIM8	XOR activa	tion		
TIM9 TIM10	One Pulse	Mode		
TIM11				
TIM12 TIM13		Configuration		
_	Reset Configu	ration		
Connectivity >	🛛 😔 NVIC Setting:			
			ser Constant	is
Multimedia >	Configure the below	w parameters :		
	Q Search (Crtl+F	۵ ۵		0
Security >	✓ Counter Setting	IS		-
Computing >				
Middleware >				





c) You can also configure the RCC to use an external oscillator as potential clock source (see *Figure 488*).





This completes the pinout configuration for this example.

Note: Starting with STM32CubeMX 4.2, the user can skip the pinout configuration by directly loading ST Discovery board configuration from the **Board selector** tab.



11.3 Saving the project

1. Click 🖶 to save the project.

When saving for the first time, select a destination folder and filename for the project. The .ioc extension is added automatically to indicate this is an STM32CubeMX configuration file.

Figure 489. Save Project As window	Figure	489.	Save	Project	As	window
------------------------------------	--------	------	------	---------	----	--------

Save In: STM32Cube_simpleLedToggle Folder name: STM32Cube_simpleLedToggle Files of Types STM32CubeMX project Files Cancel	S 🔤	ave Pr	rojec	t As		1							X
Files of Types STM32CubeMX project Files	Sav	e <u>I</u> n: [)) S	TM32Cub	e_simple	LedTog	gle	~	•	ඛ	C ∂	0.0. 0.0.	82
Files of Types STM32CubeMX project Files													
Files of Types STM32CubeMX project Files	Fol	der nar	ne:	STM32C	ube sim	pleLedTo	oggle						
		_							_				

2. Click \blacksquare to save the project under a different name or location.



11.4 Generating the report

Reports can be generated at any time during the configuration:

1. Click 🔄 to generate .pdf and .txt reports.

If a project file has not been created yet, a warning prompts the user to save the project first and requests a project name and a destination folder (see *Figure 490*). An .ioc file is then generated for the project along with a .pdf and .txt reports with the same name.



STM32 CubeMX Home / STM	File		Window	Holp	
Home / STN				Help	(19)
	M32F031E6Yx /	Untitled -	Project Manager		GENERA
Pinout & Config	guration	Clock Co	nfiguration	Project Manager Generate Report	
Project		e is generally use ot created now, y	rou will be asked for a ro	o project is currently saved.	Browse

Answering **No** will require to provide a name and location for the report only. As shown in *Figure 491*, a confirmation message is displayed when the operation is successful.

Figure 491. Generate Project Report - Project successfully created

6	Reports (Pdf and Text) are successfully generated under C:/STM32CubeMX_Projects/5_0_UM_Tuto1
	Open Folder Close

2. Open the .pdf report using Adobe Reader or the .txt report using your favorite text editor. The reports summarize all the settings and MCU configuration performed for the project.

11.5 Configuring the MCU clock tree

The following sequence describes how to configure the clocks required by the application based on an STM32F4 MCU.

STM32CubeMX automatically generates the system, CPU and AHB/APB bus frequencies from the clock sources and prescalers selected by the user. Wrong settings are detected



UM1718 Rev 47

and highlighted in fuchsia through a dynamic validation of minimum and maximum conditions. Useful tooltips provide a detailed description of the actions to undertake when the settings are unavailable or wrong. User frequency selection can influence some peripheral parameters (e.g. UART baud rate limitation).

STM32CubeMX uses the clock settings defined in the Clock tree view to generate the initialization C code for each peripheral clock. Clock settings are performed in the generated C code as part of RCC initialization within the project main.c and in stm32f4xx_hal_conf.h (HSE, HSI and external clock values expressed in Hertz).

Follow the sequence below to configure the MCU clock tree:

1. Click the **Clock Configuration** tab to display the clock tree (see *Figure 492*).

The internal (HSI, LSI), system (SYSCLK) and peripheral clock frequency fields cannot be edited. The system and peripheral clocks can be adjusted by selecting a clock source, and optionally by using the PLL, prescalers and multipliers.

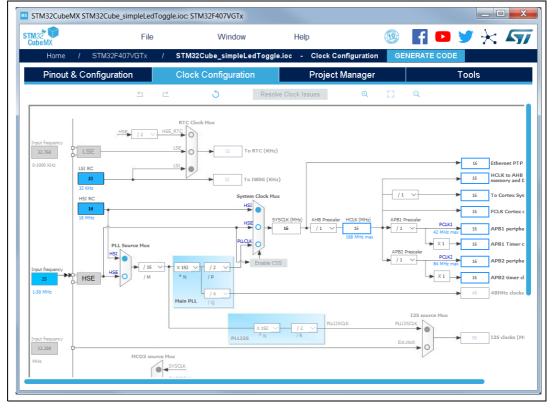


Figure 492. Clock tree view



 Select the clock source (HSE, HSI or PLLCLK) that will drive the system clock. In the example taken for the tutorial, select HSI to use the internal 16 MHz clock (see *Figure 493*).

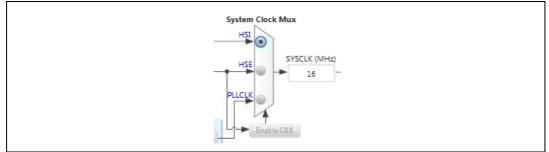
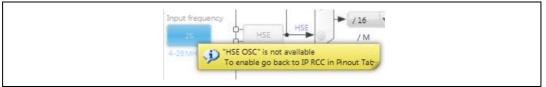


Figure 493. HSI clock enabled

To use an external clock source (HSE or LSE), the RCC peripheral must be configured in the **Pinout** view, as pins will be used to connect the external clock crystals (see *Figure 494*).

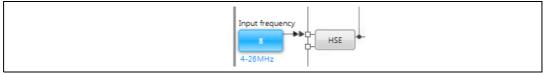
Figure 494. HSE clock source disabled



Other clock configuration options for the STM32F4DISCOVERY board:

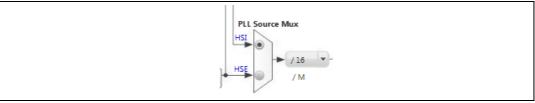
 Select the external HSE source and enter 8 in the HSE input frequency box since an 8 MHz crystal is connected on the discovery board:

Figure 495. HSE clock source enabled



 Select the external PLL clock source and the HSI or HSE as the PLL input clock source.

Figure 496. External PLL clock source enabled





- 3. Keep the core and peripheral clocks to 16 MHz using HSI, no PLL and no prescaling.
- *Note:* Optionally, further adjust the system and peripheral clocks using PLL, prescalers and multipliers:

Other clock sources independent from the system clock can be configured as follows:

- USB OTG FS, RNG and SDIO clocks are driven by an independent PLL output.
- I2S peripherals come with their own internal clock (PLLI2S), alternatively derived by an independent external clock source.
- USB OTG HS and Ethernet clocks are derived from an external source.
- 4. Optionally, configure the prescaler for the Microcontroller Clock Output (MCO) pins that allow to output two clocks to the external circuit.
- 5. Click \blacksquare to save the project.
- 6. Go to the **Configuration** tab to proceed with the project configuration.

11.6 Configuring the MCU initialization parameters

Caution: The C code generated by STM32CubeMX covers the initialization of the MCU peripherals and middlewares using the STM32Cube firmware libraries.

11.6.1 Initial conditions

From the **Pinout & Configuration** tab, select and configure (one by one) every component (peripheral, middleware, additional software) required by the application using the **Mode** and **Configuration** panels (see *Figure 497*).

Tooltips and warning messages are displayed when peripherals are not properly configured (see *Section 4* for details).

Note: The **RCC** peripheral initialization uses the parameter configuration done in this view as well as the configuration done in the **Clock tree** view (clock source, frequencies, prescaler values).



	2Cube_simpleLedToggle.io	C*: 5110132F407VG1X					
132 Tologian States State	File	Window	Help		<u>(19</u>)	🗗 🖸 🎽	\star 5
Home / ST	M32F407VGTx / S1	M32Cube_simpleLedToggle.io	oc - Pinout&Com	ïguration	GI	ENERATE CODE	
Pinout & Co	onfiguration	Clock Configuration	n	Project Manager		Tools	
tions Q		TIM3 Mode and Configuration		8	Pinout view	System view	
ategories A->Z		Mode					
	Slave Mode Disable		\sim				Middlewares
System Core >	Trigger Source Disable		~				
Analog >	Clock Source Internal		~				
	Channel1 Disable		~				
Timers 🗸	Channel2 Disable		~				
÷	Channel3 Disable		~	System Core	Analog	Timers	Connectivity
RTC	Channel4 Disable		~				
TIM1	Combined Channels D		~	DMA 🔬		тімз 🥹	
TIM2			~				
TIM4	Use ETR as Clearin	g Source		GPIO 🥹			
TIM5 TIM6	XOR activation						
TIM7	One Pulse Mode			NVIC 🥹			
TIM8							
TIM9 TIM10		Configuration					
TIM11	Reset Configuration						
TIM12 TIM13		📀 User Constants 🛛 📀 NVIC Set					
TIM15	Configure the below param		ttings 🥑 DMA Setting				
	Conligure the below parall						
	Q Search (CrtI+F)	· · · · · · · · · · · · · · · · · · ·	0				
Connectivity >	> Counter Settings						
	> Trigger Output (TRGO) I	Parameters					
Multimedia >							
Security >							
Security /							

Figure 497. Pinout & Configuration view

11.6.2 Configuring the peripherals

Each peripheral instance corresponds to a dedicated button in the main panel. Some peripheral modes have no configurable parameters, as illustrated below.

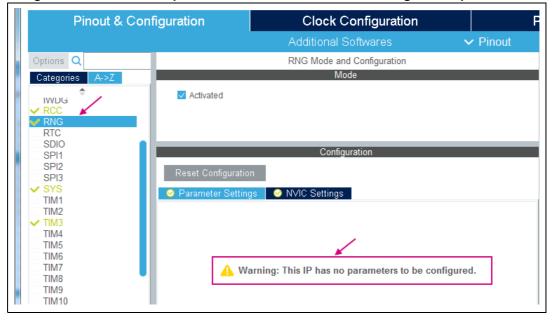


Figure 498. Case of Peripheral and Middleware without configuration parameters



Follow the steps below to proceed with peripheral configuration:

- 1. Click the peripheral button to open the corresponding configuration window. In our example
 - a) click **TIM3** to open the timer configuration window.

Figure 499. Timer 3 configuration window

C	onfiguration	
Reset Configuration		
🦻 Parameter Settings 📔 🥝 User Constants 📔 📀 NVIC Set	tings 🛛 📀 DMA Settings 📄	
nfigure the below parameters :		
Search (CrtI+F) ()		0
Counter Settings		
Prescaler (PSC - 16 bits value)		
Counter Mode	Up	
Counter Period (AutoReload Register - 16 bits value)	0	
Internal Clock Division (CKD)	No Division	
Trigger Output (TRGO) Parameters		
Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)	
Trigger Event Selection	Reset (UG bit from TIMx_EGR)	
energian (DSC _ 4C bits unlus)		
rescaler (PSC - 16 bits value) rescaler must be between 0 and 65 535.		

b) with a 16 MHz APB clock (Clock tree view), set the prescaler to 16000 and the counter period to 1000 to make the LED blink every millisecond.

Figure 500. Timer 3 configuration

Ci	onfiguration	
Reset Configuration		
📀 Parameter Settings 🛛 📀 User Constants 🛛 📀 NVIC Set	tings 🛛 🥝 DMA Settings 📄	
Configure the below parameters :		
Q Search (CrtI+F) O O		0
✓ Counter Settings		
Prescaler (PSC - 16 bits value)	16000	
Counter Mode	Up	
Counter Period (AutoReload Register - 16 bits value)	0	
Internal Clock Division (CKD)	No Division	
 Trigger Output (TRGO) Parameters 		
Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)	
Trigger Event Selection		
Trigger Event Selection TIM_MasterOutputTrigger		
TIM_WasterOutputTigger		



- 2. Optionally, and when available, select:
 - The NVIC Settings tab to display the NVIC configuration and enable interruptions for this peripheral.
 - The DMA Settings tab to display the DMA configuration and to configure DMA transfers for this peripheral.

In the tutorial example, the DMA is not used and the GPIO settings remain unchanged. The interrupt is enabled, as shown in *Figure 501*.

- The GPIO Settings tab to display the GPIO configuration and to configure the GPIOs for this peripheral.
- Insert an item:
- The User Constants tab to specify constants to be used in the project.

Figure 501. Enabling Timer 3 interrupt

Configuration					
Reset Configuration					
Parameter Settings	🥺 User Constants	🛛 🥹 NVIC Se	ttings	🥺 DMA Settings	
NVIC Interrup	Enabled	Pr	eemption Priority	Sub Priority	
TIM3 global interrupt		0		0	

11.6.3 Configuring the GPIOs

The user can adjust all pin configurations from this window. A small icon along with a tooltip indicates the configuration status.

			Middlewares			
System Core	Analog	Timers	Connectivity	Multimedia	Security	Computing
DMA 🛕		тімз 😔)		RNG 😔)
NVIC 🧭 GPIO: This IP	General Purpose Inp is correctly configure	ut Output ed. You can generate	code using current value	es.		

Figure 502. GPIO configuration color scheme and tooltip



Follow the sequence below to configure the GPIOs:

- 1. Click the **GPIO button** in the Configuration view to open the **Pin Configuration** window.
- The first tab shows pins that have been assigned a GPIO mode, but not for a dedicated peripheral and middleware. Select Pin Name to open the configuration for that pin.
 In the tutorial example, select PD12 and configure it in output push-pull mode to drive the STM32F4DISCOVERY LED (see *Figure 503*).

Figure 503. GPIO mode configuration	Figure 503.	GPIO mode	configuration
-------------------------------------	-------------	-----------	---------------

			GPIO Mode a	and Configuration			
			Confi	guration			
Group B	y Peripherals						
😔 GPIO	🥝 RCC 🛛						
Search Sign	vale						
Search (Crt.						Show of the second s	only Modified Pins
Pin Nam		GPIO output level	GPIO mode	CPIO Pull-un/Pull-	Maximum output s	User Label	Modified
D12	n/a	Low	Output Push Pull	No pull-up and no		User Laber	
-							
-			[Low			~
PD12 Config GPIO output GPIO mode	t level			Low Output Push Pull			~
GPIO output	t level		[l-down		
GPIO output GPIO mode	t level p/Pull-down		[Output Push Pull	l-down		~

11.6.4 Configuring the DMAs

This is not required for this example. It is recommended to use DMA transfers to offload the CPU. The DMA Configuration window provides a fast and easy way to configure the DMAs (see *Figure 504*):

- 1. add a new DMA request and select among a list of possible configurations.
- 2. select among the available streams.
- 3. select the Direction: Memory to Peripheral or Peripheral to Memory.
- 4. select a Priority.
- 5. enable the FIFO.

Note: Configuring the DMA for a given peripheral and middleware can also be performed using the Peripheral and Middleware configuration window.



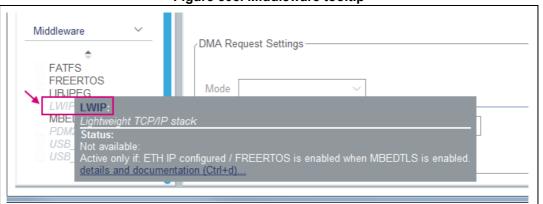
		DMA Mode :	and Configuration		
	Ann Tablan	Conf	iguration		
OMA1 OMA2 DMA Reques OMA Reques Select TIM3_CH4/UP	MemToMem	Stream	Direction		Priority
DMA Request Settings				Peripheral	Add Delete Memory
Mode	\sim		Increment Address		
Use Fifo 🔲	Threshold	~	Data Width	~	

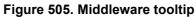
Figure 504. DMA parameters configuration window

11.6.5 Configuring the middleware

This is not required for the example taken for the tutorial.

If a peripheral is required for a middleware mode, the peripheral must be configured in the **Pinout** view for the middleware mode to become available. A tooltip can guide the user as shown below.







1. Configure the USB peripheral from the **Pinout** view.

Figure	506.	USB	Host	configuration
--------	------	-----	------	---------------

Options Q		USB_OTG_FS Mode and Configuration
Categories A->Z		Mode
USARIS USART6	Mode Host_Only	
USB_OTG	Activate_SOF	
	Activate_VBUS	

- 2. Select MSC_FS class from USB Host middleware.
- 3. Select the checkbox to enable FatFs USB mode in the tree panel.

Figure 507. FatFs over USB mode enabled
FATES Mode and Configuration
Mode
External SRAM
SD Card
USB Disk
User-defined
Configuration
Reset Configuration
🛛 🥝 User Constants 🛛 📀 Platform Settings
Set Defines Set Vanced Settings
Configure the below parameters :
Q Search (CrtI+F)
✓ Version
FATFS version R0.12c



4. Select the **Configuration** view. FatFs and USB buttons are then displayed.

STM32CubeMX S	TM32Cube_si	mpleLedToggle.ioo	*: STM32F407VGTx						. D X
STM32 CubeMX		File	Window	He	lp	12	f 🖸	ע י	< 57
Home /	STM32F407	7VGTx / ST	M32Cube_simpleLed	Toggle.ioc ·	Pinout & Configura	ation GE	NERATE C	ODE	
Pinout & C	onfiguratio	n 🛛 🛛 O	lock Configuration	ı	Project Manage	r		Tools	
				🗸 Pinot	ıt				
Options Q	\$			🔯 Pinout vi	w System vie	w			
Categories A->Z					Middlewares				
Timers >	_								
Connectivity >				FAIF	is 🤡 USB_H	osi 🖌			
Multimedia >		System Core	Analog	Timers	Connectivity	Multimedia	a s	Security	Comput
Security >									
Computing >		DMA 🚣		тімз 😔	USB_FS 😔			RNG 😔	
Middleware ~		gpio 🥝							
÷									
FATES FREERTOS		RCC 🥑							
LIBJPEG									
MBEDTLS PDM2PGM									
USB_DEVICE									

Figure 508. System view with FatFs and USB enabled



5. FatFs and USB using default settings are already marked as configured . Click **FatFs** and **USB** buttons to display default configuration settings. You can also change them by following the guidelines provided at the bottom of the window.

	Configuratio	n	
Reset Configuration			
🥝 Advanced Settings	🥝 User Constants	😔 Platform Settings	
	🥝 Set Defin	es	
Configure the below param	ieters :		
Q Search (CrtI+F)	0	\odot	0
✓ Version			
FATFS version	R0.12	с	
✓ Function Parameters			
FS_READONLY (R	ead-only mode) Disab	led	
FS_MINIMIZE (Mini	mization level) Disab		
USE_STRFUNC (S	String functions) Enable	ed with LF -> CRLF conversion	n
USE_FIND (Find fu	inctions) Disabl	led	
USE_MKFS (Make	filesystem fun Enable	ed	
0: All basic functions are er	0.12 efines minimization leve nabled.	el to remove some functions. te() and f_rename() functions	are

Figure 509. FatFs define statements



11.7 Generating a complete C project

11.7.1 Setting project options

Default project settings can be adjusted prior to C code generation as shown in *Figure 510*.

- 1. Select the **Project Manager** view to update project settings and generation options.
- 2. Select the **Project Tab** and choose a Project **name**, **location**, a **toolchain** and a **toolchain version** to generate the project (see *Figure 510*).

Pinout & Configuration	Clock Configuration	Project Manager	Tools
	Project Settings		
	Project Name	l5project	
Project	Project Location	C:\STM32CubeMX_Projects	
	Application Structure	Advanced	\vee
	Toolchain Folder Location	C:\STM32CubeMX_Projects\\5project\	
Code Generator	Toolchain / IDE	EWARM ✓ Min Version	V9.30 ×
	Linker Settings		
		M33S M33NS	
	Minimum Heap Size	0x200 0x200	
Advanced Settings	Minimum Stack Size	0x400 0x400	
	/Thread-safe Settings		
	Enable multi-threaded support		
	Thread-safe Locking Strategy	Default – Mapping suitable strategy depending on RTOS	selection.
	CortexM33NS		
	Enable multi-threaded support		
	Thread-safe Locking Strategy	Default – Mapping suitable strategy depending on RTOS :	selection.
	Mcu and Firmware Package		
	Mcu Reference	STM32L552MEYxP	
	Firmware Package Name and Version	STM32Cube FW_L5 V1.5.1	\vee
	✓ Use Default Firmware Location		
	Firmware Relative Path	C:/Users/bekrisli/STM32Cube/Repository/STM32Cube_FV	V_L5_V1.5.1

Figure 510. Project Settings and toolchain selection



- 3. Select the **Code Generator** tab to choose various C code generation options:
 - The library files copied to *Projects* folder.
 - C code regeneration (e.g. what is kept or backed up during C code regeneration).
 - HAL specific action (for example, set all free pins as analog I/Os to reduce power consumption).

In the tutorial example, select the settings as displayed in *Figure 511*, and click **OK**.

Note: A dialog window appears when the firmware package is missing. Go to next section for explanation on how to download the firmware package.

Pinout & Cor	nfiguration	Clock Configuration	Project Manager
			Generate Report
Project	 Copy all used Copy only the Add necessar 	vare Library Package libraries into the project folder necessary library files y library files as reference in the toolchain projec	t configuration file
Code Generator	Backup previo Keep User Co	oheral initialization as a pair of '.c/.h' files per per usly generated files when re-generating de when re-generating sly generated files when not re-generated	ipheral
Advanced Settings	Template Settings) Settings
		-	

Figure 511. Project Manager menu - Code Generator tab

11.7.2 Downloading firmware package and generating the C code

1. Click **GENERATE CODE** to generate the C code.

During C code generation, STM32CubeMX copies files from the relevant STM32Cube MCU package into the project folder so that the project can be compiled. When generating a project for the first time, the firmware package is not available on the user PC and a warning message is displayed:

Figure 512. Missing firmware package warning message

Project	Manager Settings
▲	The Firmware Package (STM32Cube FW_F4 V1.22.0RC1) or one of its dependencies required by the Project is not available in your STM32CubeMX Repository. Do you want to download this now ?
	<u>Yes</u> No



 STM32CubeMX offers to download the relevant firmware package or to go on. Click Download to obtain a complete project, that is a project ready to be used in the selected IDE.

By clicking **Continue**, only *Inc* and *Src* folders will be created, holding STM32CubeMX generated initialization files. The necessary firmware and middleware libraries will have to be copied manually to obtain a complete project.

If the download fails, an error message is displayed.

Figure 513. Error during download

Probler	n during Download and/or Unzip
8	Error during Access to HTTP Server. Please check Proxy settings under 'Help > Updater Settings > Connection Parameters'.
	ОК

To solve this issue, execute the next two steps. Skip them otherwise.

3. Select **Help > Updater Settings** menu and adjust the connection parameters to match your network configuration.

Vpdater Settings
Updater Settings Connection Parameters
Proxy Server Type
O No Proxy
O Use System Proxy Parameters
Manual Configuration of Proxy Server
Manual Configuration of Proxy Server
Proxy HTTP Port 8080
Authentication
Require Authentication Remember my Credentials
User Login JohnDoe
Password ••••••
Check Connection
OK Cancel

Figure 514. Updater settings for download

4. Click **Check connection.** The check mark turns green once the connection is established.

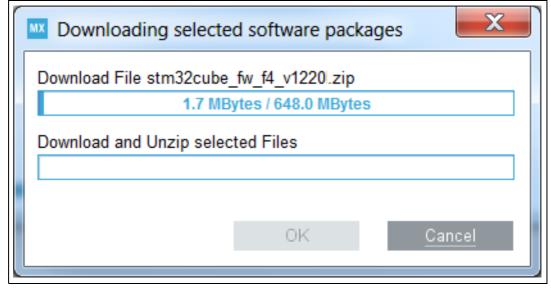


Updater Settings	X
Updater Settings Connection Parameters	
Proxy Server Type	
O No Proxy	
O Use System Proxy Parameters	
 Manual Configuration of Proxy Server 	
Manual Configuration of Proxy Server	
Proxy HTTP do.it.mycompany.com	Port 8080
Authentication	
Require Authentication Remember my Credentials	
User Login JohnDoe	
Password ••••••	
	OK Cancel

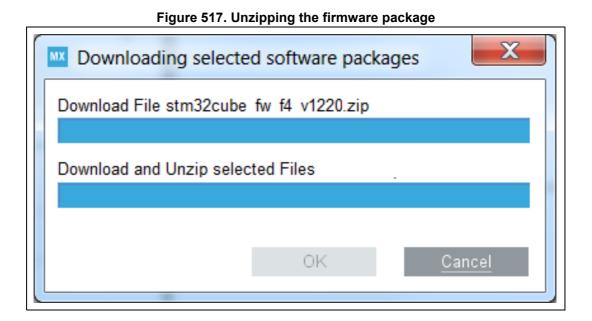
Figure 515. Updater settings with connection

5. Once the connection is functional, click **GENERATE CODE** to generate the C code. The C code generation process starts and progress is displayed (see next figures).

Figure 516. Downloading the firmware package







6. Finally, a confirmation message is displayed to indicate that the C code generation has been successful.



ſ	Co	de Generation
	0	The Code is successfully generated under C:/STM32CubeMX_Projects/stm32f429_fatfs_sd_test
		Open Folder Open Project Close



7. Click **Open Folder** to display the generated project contents or click **Open Project** to open the project directly in your IDE. Then proceed with Section 11.8.

<u>File Edit View Tools H</u> elp		
Organize 🔹 🥞 Open 🛛 Include in libra	ary 🕶	• » 📰 • 🔟 📀
 STM32Cube_SimpleLedToggle Drivers Inc Middlewares Projects Src 	•	Drivers Inc Middlewares Projects Src

Figure 519. C code generation output folder

The generated project contains:

- The STM32CubeMX .ioc project file located in the root folder. It contains the project user configuration and settings generated through STM32CubeMX user interface.
- The *Drivers* and *Middlewares* folders hold copies of the firmware package files relevant for the user configuration.
- The *Projects* folder contains IDE specific folders with all the files required for the project development and debug within the IDE.
- The *Inc* and *Src* folders contain STM32CubeMX generated files for middleware, peripheral and GPIO initialization, including the main.c file. The STM32CubeMX generated files contain user-dedicated sections allowing to insert user-defined C code.
- **Caution:** C code written within the user sections is preserved at next C code generation, while C code written outside these sections is overwritten.

User C code will be lost if user sections are moved or if user sections delimiters are renamed.



11.8 Building and updating the C code project

This example explains how to use the generated initialization C code and complete the project, within IAR[™] EWARM toolchain, to have the LED blink according to the TIM3 frequency.

A folder is available for the toolchains selected for C code generation: the project can be generated for more than one toolchain by choosing a different toolchain from the **Project Manager** menu and clicking Generate code once again.

 Open the project directly in the IDE toolchain by clicking **Open Project** from the dialog window or by double-clicking the relevant IDE file available in the toolchain folder under STM32CubeMX generated project directory (see *Figure 518*).

STM32Cube_simpleLedTog	gle → ← 4y Search STM P
Eile Edit View Tools Help Organize Include in library Share with ▼ ▲ STM32Cube_simpleLedToggle ▲	Burn New folder III ♥ 🗍 🕢
	 Drivers EWARM Inc MDK-ARM Src SW4STM32 TrueSTUDIO .mxproject STM32Cube_simpleLedToggle.ioc STM32Cube_simpleLedToggle.btt STM32Cube_simpleLedToggle.configuration.pdf
11 items	

Figure 520. C code generation output: Projects folder



2. As an example, select .eww file to load the project in the IAR™ EWARM IDE.

<u>File Edit View Tools H</u> elp		
Organize 🔹 🎉 Open 🔹 Burn 🛛 New fold		III • 🔟 🔞
 STM32Cube_simpleLedToggle Drivers EWARM Inc MDK-ARM Src SW4STM32 TrueSTUDIO 	Name settings STM32Cube_simpleLedToggle Co Project.eww STM32Cube_simpleLedToggle.ew STM32Cube_simpleLedToggle.ew stm32f407xx_flash.icf stm32f407xx_sram.icf	7/28/2015 2:3 vd 7/28/2015 2:3
	<	

Figure 521. C code generation for EWARM



3. Select the main.c file to open in editor.

le <u>E</u> dit <u>V</u> iew <u>P</u> roject <u>T</u> ools <u>W</u> indow <u>H</u> elp	
) 🖆 🖬 🎒 🎒 🐰 🖻 🛍 🗠 🖂	- 🗸 🍾 🎠 📜 🗈 🐢 📣 🎒 🔤 👯 🤌 🕭
	× main.c*
TM32Cube_simpleLedToggle Configuration	
	includes later 2054 m bal bu
Files 🐔 🛤	<pre>#include "stm32f4xx_hal.h" /* USER CODE BEGIN Includes */</pre>
TSTM32Cube_simpleLedTogqle 🗸	/* USER CODE END Includes */
	/* OBER CODE END INCIDEES */
	/* Private variables
	TIM HandleTypeDef htim3;
– ⊕ © main.c	/* USER CODE BEGIN PV */
—⊞ [c] stm32f4∞_hal_msp.c	/* Private variables
L	/* USER CODE END PV */
	/* Private function prototypes
	void SystemClock_Config(void);
	<pre>static void MX_GPIO_Init(void);</pre>
	<pre>static void MX_TIM3_Init(void);</pre>
	/* USER CODE BEGIN PFP */
	/* USER CODE END PFP */
	/* USER CODE BEGIN 0 */
	/* USER CODE END 0 */
	int main (void)
	/* USER CODE BEGIN 1 */
	/* USER CODE END 1 */ /* MCU Configuration
	/* Reset of all peripherals, Initializes the Flash interface and the Sy.
	HAL Init();
	/* Configure the system clock */
	SystemClock Config();
	/* Initialize all configured peripherals */
	MX GPIO Init();
	MX_TIM3_Init();
	/* USER CODE BEGIN 2 */
	/* USER CODE END 2 */
	/* Infinite loop */
	/* USER CODE BEGIN WHILE */
	while (1)
	/* USER CODE END WHILE */
	/* USER CODE BEGIN 3 */
	/* USER CODE END 3 */
TM32Cube_simpleLedToggle	
	۰ ۲ (III) کې د د د د د د د د د د د د د د د د د د

Figure 522. STM32CubeMX generated project open in IAR™ IDE

The htim3 structure handler, system clock, GPIO and TIM3 initialization functions are defined. The initialization functions are called in the main.c. For now the user C code sections are empty.



4. In the IAR[™] IDE, right-click the project name and select **Options**.

STM32Cube_Simpl	eLedToggle Configurati	on	•
Files	4		ð;
🗆 🗇 STM32Cube	_SimpleLedToggle	× .	
🛛 🛏 🗀 Application			. * .
📙 🕀 🗀 Drivers			. * .
🕂 🕀 🗀 Example			. *
🖵 🕀 🗀 Output			

Figure 523. IAR™ options

5. Click the ST-LINK category and make sure SWD is selected to communicate with the STM32F4DISCOVERY board. Click **OK**.

Linker Debugger Simulator Angel CMSIS DAP GDB Server IAR ROM-monitor I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor PE micro RDI	Interface ITAG @	Clock setup CPU clock: SWO clock:	72.0 Auto 2000	MHz kHz		
ST-LINK						

Figure 524. SWD connection

6. Select **Project > Rebuild all**. Check if the project building has succeeded.

Figure 525. Project building log

Messages
stm32f4xx_hal_tim.c stm32f4xx_hal_tim_ex.c stm32f4xx_it.c stm32f4xx_ll_sdmmc.c system_stm32f4xx.c Linking
Total number of errors: 0 Total number of warnings: 0

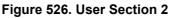


7. Add user C code in the dedicated user sections **only**.

Note: The main while(1) loop is placed in a user section.

For example:

- a) Edit the main.c file.
- b) To start timer 3, update User Section 2 with the following C code:



```
HAL_Init();
/* Configure the system clock */
SystemClock_Config();
/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_TIM3_Init();
/* USER CODE BEGIN 2 */
HAL_TIM_Base_Start_IT(shtim3);
/* USER CODE END 2 */
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
```

c) Then, add the following C code in User Section 4:

```
Figure 527. User Section 4
```

```
/* USER CODE BEGIN 4 */
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    if ( htim->Instance == htim3.Instance )
      {
         HAL_GPI0_TogglePin(GPI0D, GPI0_PIN_12);
         }
    }
    /* USER CODE END 4 */
```

This C code implements the weak callback function defined in the HAL timer driver (stm32f4xx_hal_tim.h) to toggle the GPIO pin driving the green LED when the timer counter period has elapsed.

- 8. Rebuild and program your board using . Make sure the SWD ST-LINK option is checked as a Project options otherwise board programming will fail.
- 9. Launch the program using 2. The green LED on the STM32F4DISCOVERY board will blink every second.
- 10. To change the MCU configuration, go back to STM32CubeMX user interface, implement the changes and regenerate the C code. The project will be updated, preserving the C code in the user sections if ✓ Keep User Code when re-generating option in Project Manager's Code Generator tab is enabled.



11.9 Switching to another MCU

STM32CubeMX allows loading a project configuration on an MCU of the same series.

Proceed as follows:

- 1. Select File > New Project.
- 2. Select an MCU belonging to the same series. As an example, you can select the STM32F429ZITx that is the core MCU of the 32F429IDISCOVERY board.
- Select File > Import project. In the Import project window, browse to the .ioc file to load. A message warns you that the currently selected MCU (STM32F429ZITx) differs from the one specified in the .ioc file (STM32F407VGTx). Several import options are proposed (see *Figure 528*).
- 4. Click the **Try Impor**t button and check the import status to verify if the import has been successful.
- 5. Click **OK** to really import the project. An output tab is then displayed to report the import results.
- 6. The green LED on 32F429IDISCOVERY board is connected to PG13: CTRL+ right click **PD12** and drag and drop it on PG13.
- 7. From **Project Manager** project tab configure the new project name and folder location. Click **Generate icon** to save the project and generate the code.
- 8. Select **Open the project** from the dialog window, update the user sections with the user code, making sure to update the GPIO settings for PG13. Build the project and flash the board. Launch the program and check that LED blinks once per second.

Imported Project CLSTM32CubeMX_Projects/5_0_UM_Tuto1/STM32Cube_simpleLedToggle/STM32Cube_simpleLedToggle.ioc Import Nower Consumption Calculator Settings Import Project Settings Import Prinot/Clock Configuration Settings Automatic Import Automatic Import Automatic Sconfiguration Automatic Sconfiguration Automatic Import Import Prinoty Status Import Prinoty Status Import Prinoty Status Import Prinoty Status Import Prinoty Status Import Sconfiguration Try Import Show View Pinout ~ Import Status Intializing: STM32F407V(E=0)Tx Import Analysis: C:\STM32CubeMX_Projects\5_0_UM_Tuto1\STM32Cube_simpleLedToggle\STM32Cube_simpleLedToggle.ioc project The Kou (STM32F407VGrx) found in the Project being Imported is not the same as the Kou (STM32F4292Tx) currently edited Import Try : Import Try : Importing project completed	Import Pr	-
Import N2 Status Import Project Settings Import Project Settings Import Project Settings Automatic Import Automatic Import Manual Import Import Peripherals Configuration Peripheral List To STM32F4292ITX PATES NVC RCC RCC RCC RCC RCC RCC RCC R	1 1	
Import Power Consumption Calculator Settings Import Project Settings Import Prinout/Clock Configuration/Configuration Settings Automatic Import Manual Import Import Peripherals Configuration Peripheral List Form STMAL To STM32F4292ITX FATES NVIC Import Status Intralizing: STM32F407V(E-6)Tx Import Analysis: Import Try :	C:\STM32Cub	eMX_Projects\5_0_UM_Tuto1\STM32Cube_simpleLedToggle\STM32Cube_simpleLedToggle.ioc
Import Project Settings Import Prinout/Clock Configuration/Configuration Settings ● Automatic Import ● Manual Import ● Import Prining Status ● Import Status ● Import Status ■ Intializing: STM32F407V(E-G)Tx Import Analysis: C:\STM32CubeMX_Projects\5_0_UM_Tutol\STM32Cube_simpleLedToggle\STM32Cube_simpleLedToggle.icc project The Mou (STM32F407VGTx) found in the Project being imported is not the same as the Mou (STM32F4292ITx) currently edited Import Try : Importing project completed	mport MX Se	tings
Automatic Import Manual Infort	Import Pov	ver Consumption Calculator Settings
Automatic Import Manual Import Import Prinning Status Import Peripherals Configuration Peripheral List To STM32F4292TX FATFS FATFS FATFS FATFS FATFS To STM32F4292TX FATFS FAT	Import Pro	ject Settings
Automatic Import Manual Import Import Prinning Status Import Peripherals Configuration Peripheral List To STM32F4292ITx FATFS FATFS FATFS FATFS To STM32F4292ITx FATFS FATFS To STM32F4292ITx FATFS To STM32F4292ITx FATFS To STM32F4292ITx FATFS To STM32F4292ITx FATFS Try Import Show View Pinout Import Status Initializing: STM32F407V(E-G)Tx Import Analysis: C:\STM32CubeMX_Projects\5_0_UM_Tutol\STM32Cube_simpleLedToggle.ioc project The Mou (STM32F407VGTx) found in the Project being imported is not the same as the Mou (STM32F4292ITx) currently edited Import Try : Import Try : Import Try : Import completed	mport Pinout	Clock Configuration/Configuration Settings
> Manual Import ■ Import Prinning Status ■ Import Peripherals Configuration > Peripheral List From STM3. ■ FATFS NVIC ■ NVIC RCC RNG eve ■ Import Status Initializing: STM32F407V (E-G) Tx Import Analysis: C: Status Import Xnalysis: C: STM32F407V (E-G) Tx Import Analysis: C: Status Import Xnalysis: C: STM32F407V (E-G) Tx Import Xnalysis: C: Status		
Import Prining Status Peripheral List To STM32F4292ITx FATFS NVIC RCC RCC RCC RCC RCC Try Import Show View Pinout Import Status Initializing: STM32F407V(E-G)TX Import Analysis: C:\STM32CubeMX_Projects\5_0_UM_Tuto1\STM32Cube_simpleLedToggle\STM32Cube_simpleLedToggle.ioc project The Mou (STM32F407V(E-G)Tx Import Try : Import Try : Import Try : Import completed		
Import Peripherals Configuration Peripheral List To STM32F4292ITx FATFS FATFS NVIC RCC RCC RCC RCC RG STM32F407V(E-G) Tx Import Analysis: C:\STM32F407V(E-G) Tx Import Analysis: C:\STM32CubeKX_Projects\5_0_UM_Tutol\STM32Cube_simpleLedToggle.ioc project The Mcu (STM32F407VGTx) found in the Project being imported is not the same as the Mcu (STM32F4292ITx) currently edited Import Try : Import Try : Import Import completed		
Peripheral List Farrs To STM32F429ZITX FATFS FATFS NVIC NVIC RCC RCC RNG NNG eve Try Import Show View Pinout Import Status Initializing: STM32F407V(E-G)Tx Import Analysis: C:\STM32CubeKX_Projects\5_0_UM_Tutol\STM32Cube_simpleLedToggle\STM32Cube_simpleLedToggle.icc project The Mcu (STM32F407VGTx) found in the Project being imported is not the same as the Mcu (STM32F429ZITx) currently edited Import Try : Import Try : Import ing project completed		
From STM3 To STM32F4292ITx FATFS FATFS NVIC NVIC RCC RCC RNG NNG Eve Try Import Show View Pinout Import Status Initializing: STM32F407V(E-G) Tx Import Analysis: C:\STM32CubeXx_Projects\5_0_UM_Tutol\STM32Cube_simpleLedToggle\STM32Cube_simpleLedToggle.icc project The Mcu (STM32F407VGTx) found in the Project being imported is not the same as the Mcu (STM32F429ZITx) currently edited Import Try : Import Try : Import ing project completed		
FATES FATES NVIC NVIC RCC RCC RNG RNG eve Try Import Show View Pinout Import Status		
INVIC NMIC RCC RCC RNG RNG Eve Try Import Show View Pinout Import Status		
RCC RCC RNG RNG Import Status		
Try Import Try Import Show View Pinout Import Status Initializing: STM32F407V(E-G)Tx Import Analysis: C:\STM32CubeMX_Projects\5_0_UM_Tutol\STM32Cube_simpleLedToggle\STM32Cube_simpleLedToggle.icc project The Mcu (STM32F407VGTx) found in the Project being imported is not the same as the Mcu (STM32F4292ITx) currently edited Import Try : Importing project completed		
Try Import Show View Pinout Import Status		
Import Status Initializing: STM32F407V(E-G)Tx Import Analysis: C:\STM32CubeHX_Projects\5_0_UM_Tuto1\STM32Cube_simpleLedToggle\STM32Cube_simpleLedToggle.ioc project The Mcu (STM32F407VGTx) found in the Project being imported is not the same as the Mcu (STM32F429ZITx) currently edited Import Try : Importing project completed	eve	
Initializing: STM32F407V(E-G)Tx Import Analysis: C:\STM32CubeMX_Projects\5_0_UM_Tuto1\STM32Cube_simpleLedToggle\STM32Cube_simpleLedToggle.ioc project The Mcu (STM32F407VGTx) found in the Project being imported is not the same as the Mcu (STM32F429ZITx) currently edited Import Try : Importing project completed	Import Statu	
<pre>Import Analysis: C:\STM32CubeMX_Projects\5_0_UM_Tutol\STM32Cube_simpleLedToggle\STM32Cube_simpleLedToggle.icc project The Mcu (STM32F407VGTx) found in the Project being imported is not the same as the Mcu (STM32F429ZITx) currently edited Import Try : Importing project completed</pre>		
The Mcu (STM32F407VGTx) found in the Project being imported is not the same as the Mcu (STM32F429ZITx) currently edited Import Try : Importing project completed		
Import Try : Importing project completed		
Importing project completed		
	Import Tr	Y :
OK Cancel	Importing	project completed
Old Cancel		
		OK Cancel

Figure 528. Import Project menu



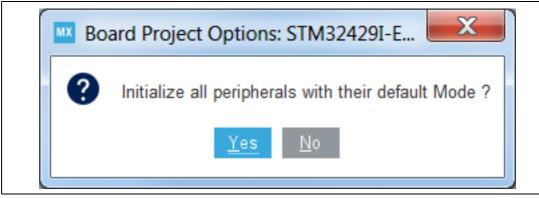
12 Tutorial 2 - Example of FatFs on an SD card using STM32429I-EVAL evaluation board

The tutorial consists in creating and writing to a file on the STM32429I-EVAL1 SD card using the FatFs file system middleware.

To generate a project and run tutorial 2, follow the sequence below:

- 1. Launch STM32CubeMX.
- 2. Select File > New Project. The Project window opens.
- 3. Click the **Board Selector** Tab to display the list of ST boards.
- 4. Select EvalBoard as type of Board and STM32F4 as Series to filter down the list.
- 5. Answer Yes to Initialize all peripherals with their default mode so that the code is generated only for the peripherals used by the application.
- 6. Select the STM32429I-EVAL board and click **OK**. Answer No in the dialog box asking to initialize all peripherals to their default modes (see *Figure 529*). The **Pinout** view is loaded, matching the MCU pinout configuration on the evaluation board (see *Figure 530*).

Figure 529. Board peripheral initialization dialog box



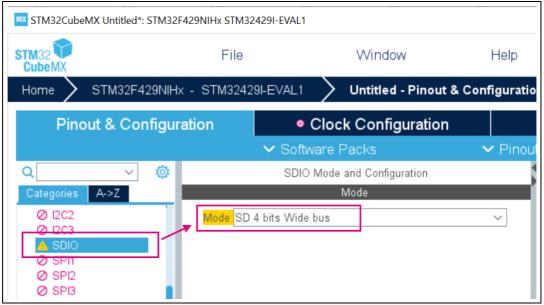


lew Project from a Board											
ICU Selector Board Selector											
Board Filters											
★ 🔂 🛱	J	_	Feat Larg	e Pi D	ocs & Resou	[🕌 Datas		1	G→ St	art Pr
	~		STM32429I-EVAL								
Part Number Search											-
۹ 🗌	~		C 1		electronic and Exa		324291	Evalua	ation Bo	ard	
Vendor	>		STM32 F4	ACTIVE	Active		Unit F	Price (US	\$):389.0		
Туре	~				in mass pro	duction	Moun	ted devic	e: <u>STM32F</u>	429NIHx	
Check/Uncheck All											
Discovery											
Evaluation Board		Boards	List: 7 items								x∎
Nucleo144		*	Overview	Part No 🌻	Type Market.	Unit Pric	. Mounte	MCU Se	.Custom Fo.	Memory	ROM
Nucleo64											
		☆		STM3241G	Ev Active	349.0	STM32F	STM32	0	1	0
MCU Series	\sim										
Check/Uncheck All					· · · ·				·		
STM32F0		☆		STM32429I	Ev Active	389.0		STM32	. 0	0	1
STM32F1											

Figure 530. Board selection

- 7. From the Peripheral tree on the left, expand the SDIO peripheral and select "SD 4 bits wide bus" (see *Figure 531*). In the configuration panel, from the DMA settings tab, add SDIO_RX and SDIO_TX DMA requests.
- 8. Finally, go pack to the peripheral tree panel, select NVIC and enable the SDIO global interrupt from the configuration panel.







9. Under the Middlewares category, check SD card as FatFs mode (see *Figure 532*).

FATFS Mode and Configuration
Mode
External SDRAM
External SRAM
✓ SD Card
USB Disk
User-defined

Figure 532. FatFs mode configuration

From the Pinout view on the right, enable, as GPIO input, a pin to be used for the SDIO detection.

In the configuration panel below the mode panel, go to the platform settings tab and configure the SD_detection using the pin previously enabled.

Finally, go to FatFs "Advanced settings tab" and enable "Use DMA template".

- 10. Configure the clocks as follows:
 - a) Select the RCC peripheral from the **Pinout** view (see *Figure 533*).

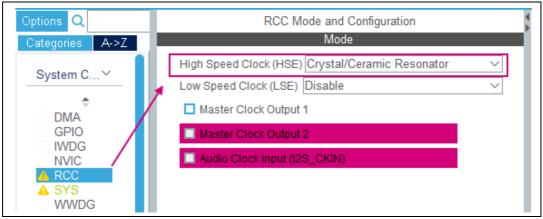


Figure 533. RCC peripheral configuration



b) Configure the clock tree from the clock tab (see *Figure 534*).

HSIRC	System Clock Mux
16	HSI
16 MHz	HSE SYSCLK (MHz) AHB Prescaler HCLK (MHz
	168 /1 168 180 MHz n
	PLL Source Mux
	HSI (25 + X336 + (2 + Enable C85
Input frequency	HSE
	• /M *N /P
4-26MHz	17 -
	Main PLL / Q

Figure 534. Clock tree view

11. In the **Project** tab, specify the project name and destination folder. Then, select the EWARM IDE toolchain.

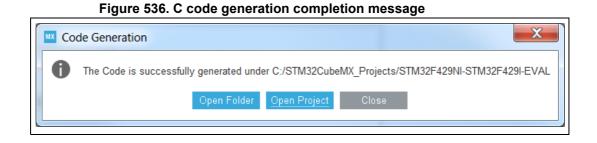
Note that project heap and stack size can be adjusted to the minimum required for the FATFS application.



Pinout & Conf	iguration	Clock C	onfiguration	Project Manager	То
				Generate Repo	rt
Project	Project Setting Project Name STM32F429NI Project Locatio C:\STM32Cube Application Str	STM32F429I-EVA n MX_Projects	L		
Code Generator	Basic Toolchain Fold	er Location MX_Projects\STM	> 32F429NI-STM32F429	Do not generate the main() I-EVAL Generate Under Root	
Advanced Settings	Linker Settings Minimum Heap Minimum Stac	Size 0>	200		
	Mcu and Firmw Mcu Reference STM32L053C8				

- 12. Click **Ok**. Then, on the toolbar menu, click **GENERATE CODE** to generate the project.
- 13. Upon code generation completion, click **Open Project** in the **Code Generation** dialog window (see *Figure 536*). This opens the project directly in the IDE.





14. In the IDE, check that heap and stack sizes are sufficient: right click the project name and select **Options**, then select **Linker**. Check **Override default** to use the icf file from STM32CubeMX generated project folder. if not already done through STM32CubeMX User interface (under Linker Settings from Project Manager's project tab), adjust the heap and stack sizes (see *Figure 537*).

Options for node "fatfs_s Category: General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator Angel CMSIS DAP GDB Server IAR ROM-monitor I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor PE micro RDI	d_test" X Factory Settings Corfig Library Input Optimizations Advanced Output List 4 4 Linker configuration file Image: Configuration file symbol definitions: (one per line) Image: Configuration file symbol definitions: (one per line) Image: Configuration file editor Image: Configuration file editor
ST-LINK Third-Party Driver TI XDS	Vector Table Memory Regions Stack/Heap Sizes CSTACK 0x800 HEAP 0x400 Save Cancel

Figure 537. IDE workspace

Note:

When using the MDK-Arm toolchain, go to the Application/MDK-ARM folder and double-click the startup_xx.s file to edit and adjust the heap and stack sizes there.



- 15. Go to the Application/User folder. Double-click the main.c file and edit it.
- 16. The tutorial consists in creating and writing to a file on the evaluation board SD card using the FatFs file system middleware:
 - a) At startup all LEDs are OFF.
 - b) The red LED is turned ON to indicate that an error occurred (e.g. FatFs initialization, file read/write access errors).
 - c) The orange LED is turned ON to indicate that the FatFs link has been successfully mounted on the SD driver.
 - d) The blue LED is turned ON to indicate that the file has been successfully written to the SD card.
 - e) The green LED is turned ON to indicate that the file has been successfully read from file the SD card.
- 17. For use case implementation, update main.c with the following code:
 - a) Insert main.c private variables in a dedicated user code section:

```
/* USER CODE BEGIN PV */
/* Private variables -----*/
FATFS SDFatFs; /* File system object for SD card logical drive */
FIL MyFile; /* File object */
const char wtext[] = "Hello World!";
static uint8 t buffer[ MAX SS]; /* a work buffer for the f mkfs() */
/* USER CODE END PV */
   b) Insert main functional local variables:
int main (void)
{
  /* USER CODE BEGIN 1 */
 FRESULT res;
                          /* FatFs function common result code */
 uint32_t byteswritten, bytesread; /* File write/read counts */
                               /* File read buffer */
 char rtext[256];
  /* USER CODE END 1 */
 /* MCU Configuration-----*/
 /* Reset of all peripherals, Initializes the Flash interface and the
Systick. */
HAL Init();
   c) Insert user code in the main function, after initialization calls and before the while
       loop, to perform actual read/write from/to the SD card:
int main(void)
{
      MX_FATFS_Init();
        /* USER CODE BEGIN 2 */
      /*##-0- Turn all LEDs off(red, green, orange and blue) */
         HAL_GPIO_WritePin(GPIOG, (GPIO_PIN_10 | GPIO_PIN_6 | GPIO_PIN_7 |
      GPIO PIN 12), GPIO PIN SET);
```



```
if(retSD == 0) {
      /* success: set the orange LED on */
     HAL GPIO WritePin(GPIOG, GPIO PIN 7, GPIO PIN RESET);
/*##-2- Register the file system object to the FatFs module ###*/
    if(f mount(&SDFatFs, (TCHAR const*)SDPath, 0) != FR OK) {
     /* FatFs Initialization Error : set the red LED on ^{\star/}
       HAL GPIO WritePin(GPIOG, GPIO PIN 10, GPIO PIN RESET);
       while(1);
      } else
                  {
/*##-3- Create a FAT file system (format) on the logical drive#*/
 /* WARNING: Formatting the uSD card will delete all content on the
device */
if(f mkfs((TCHAR const*)SDPath, FM ANY, 0, buffer, sizeof(buffer))
!= FR OK) {
   /* FatFs Format Error : set the red LED on */
   HAL GPIO WritePin(GPIOG, GPIO PIN 10, GPIO PIN RESET);
   while(1);
  } else {
/*##-4- Create & Open a new text file object with write access#*/
 if (f open(&MyFile, "Hello.txt", FA CREATE ALWAYS | FA WRITE) !=
FR OK) {
 /* 'Hello.txt' file Open for write Error : set the red LED on */
 HAL GPIO WritePin(GPIOG, GPIO PIN 10, GPIO PIN RESET);
 while(1);
    } else {
 res = f write(&MyFile, wtext, sizeof(wtext), (void
*)&byteswritten);
 if((byteswritten == 0) || (res != FR OK)){
   /* 'Hello.txt' file Write or EOF Error : set the red LED on */
   HAL GPIO WritePin(GPIOG, GPIO PIN 10, GPIO PIN RESET);
   while(1);
   } else {
 /*##-6- Successful open/write : set the blue LED on */
   HAL GPIO WritePin (GPIOG, GPIO PIN 12, GPIO PIN RESET);
   f close(&MyFile);
 /*##-7- Open the text file object with read access #*/
   if(f open(&MyFile, "Hello.txt", FA READ) != FR OK) {
   /* 'Hello.txt' file Open for read Error : set the red LED on */
   HAL GPIO WritePin(GPIOG, GPIO PIN 10, GPIO PIN RESET);
   while(1);
   } else {
 /*##-8- Read data from the text file #########/
   res = f read(&MyFile, rtext, sizeof(wtext), &bytesread);
   if((byteswritten == 0) || (res != FR OK)) {
  /* 'Hello.txt' file Read or EOF Error : set the red LED on */
   HAL GPIO WritePin(GPIOG, GPIO PIN 10, GPIO PIN RESET);
   while(1);
   } else {
  /* Successful read : set the green LED On */
   HAL GPIO WritePin(GPIOG, GPIO PIN 6, GPIO PIN RESET);
```





13 Tutorial 3 - Using the Power Consumption Calculator to optimize the embedded application consumption and more

13.1 Tutorial overview

This tutorial focuses on STM32CubeMX Power Consumption Calculator (Power Consumption Calculator) feature and its benefits to evaluate the impacts of power-saving techniques on a given application sequence.

The key considerations to reduce a given application power consumption are:

- Reducing the operating voltage
- Reducing the time spent in energy consuming modes

It is up to the developer to select a configuration that gives the best compromise between low-power consumption and performance.

- Maximizing the time spent in non-active and low-power modes
- Using the optimal clock configuration

The core should always operate at relatively good speed, since reducing the operating frequency can increase energy consumption if the microcontroller has to remain for a long time in an active operating mode to perform a given operation.

- Enabling only the peripherals relevant for the current application state and clock-gating the others
- When relevant, using the peripherals with low-power features (e.g. waking up the microcontroller with the I2C)
- Minimizing the number of state transitions
- Optimizing memory accesses during code execution
 - Prefer code execution from RAM to flash memory
 - When relevant, consider aligning CPU frequency with flash memory operating frequency for zero wait states.

The following tutorial shows how the STM32CubeMX Power Consumption Calculator feature can help to tune an application to minimize its power consumption and extend the battery life.

Note: The Power Consumption Calculator does not account for I/O dynamic current consumption and external board components that can also affect current consumption. For this purpose, an "additional consumption" field is provided for the user to specify such consumption value.



13.2 Application example description

The application is designed using the NUCLEO-L476RG board, based on an STM32L476RGTx device, and supplied by a 2.4 V battery.

The main purpose of this application is to perform ADC measurements and transfer the conversion results over UART. It uses:

- Multiple low-power modes: Low-power run, Low-power sleep, Sleep, Stop and Standby
- Multiple peripherals: USART, DMA, Timer, COMP, DAC and RTC
 - The RTC is used to run a calendar and to wake up the CPU from Standby when a specified time has elapsed.
 - The DMA transfers ADC measurements from ADC to memory
 - The USART is used in conjunction with the DMA to send/receive data via the virtual COM port and to wake up the CPU from Stop mode.

The process to optimize such complex application is to start describing first a functional only sequence then to introduce, on a step by step basis, the low-power features provided by the STM32L476RG microcontroller.

13.3 Using the Power Consumption Calculator

13.3.1 Creating a power sequence

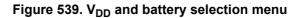
Follow the steps below to create the sequence (see Figure 538):

- 1. Launch STM32CubeMX.
- 2. Click new project and select the Nucleo-L476RG board from the Board tab.
- 3. Click the **Power Consumption Calculator** tab to select the Power Consumption Calculator view. A first sequence is then created as a reference.
- 4. Adapt it to minimize the overall current consumption. To do this:
 - a) Select 2.4 V V_{DD} power supply. This value can be adjusted on a step by step basis (see *Figure 539*).
 - b) Select the Li-MnO2 (CR2032) battery. This step is optional. The battery type can be changed later on (see *Figure 539*).



	32CubeMX Untitled	d*: STM32L476RG	Γx											_	
N32 ubeM)		File		V	/indow		Help					(19)	f		\star S
ome	STM32L476R	GTx 🔪 Untitl	ed - Tool	\$ >								GENE	RATE CO	DDE	
	Pinout & Cont	figuration		Clo	ock Configu	uration			Project	Manage	ər			Tools	
							ower								
	STM32L476RGTx		~ 1		_	St	ep —				Seque	nce		— — Tran	sitions Checke
		07140014	- 1	New St	ep 👘	ß	三介	El 🛨					5 <u>7</u> 3 (🚺 🗹 On	Log Hel
	Series Line	STM32L4 STM32L4x6	- 14							ence Table	e				
	Datasheet	025976_Rev4	- I.	Step	Mode	1	Vdd	Range/Scale		PU/Bus Freq		onfig P	eripherals	Step Curre	ent Duration
		_	1		RUN	2.4		Range1-High	24 MHz		HSE	ADO	01:fs_5	5.9 mA	1 ms
	T ₄ 25°C / V _{DD} 2.4V	,	2		STANDBY	2.4		NoRange	0 Hz		LSI RTC	RTC	2 *	464 nA	1 ms
be l	A 25 C / VDD 2.4V		<u> </u>		WU_FROM_			NoRange	4 MHz		MSI FAS	Т		1.7 mA	20.1 µs
	TAmbient	25°C	4		RUN	2.4		Range1-High	16 MHz		HSE	RTC		2.16 mA	1 ms
			5		RUN	2.4		Range2-Med			HSE	ADO	C1:fs_5	4.47 mA	1 ms
	V _{DD}	2.4	~ 6		SLEEP	2.4		Range2-Med			HSE			589 µA	1 ms
			7		RUN	2.4		Range2-Med			HSE		C1:fs_5	4.47 mA	1 ms
	Li-MnO2(CR2032) ((1x1)	~ 8		STOP1	2.4		NoRange	0 Hz		ALL CLO	CK USA	ART1*	6.65 µA	1 ms
	EI-WIIO2(01(2032) (9		WU_FROM_			NoRange	16 MHz		HSI16			1.62 mA	6.3 µs
	Change	Reset	10		RUN	2.4		Range2-Med			HSE LSI RTC	RIC	USART1		1 ms
	enenge		1		STANDBY	2.4		NoRange	0 Hz		LSIRIC			464 nA	1 ms
			- E		Dis	play Cho	ices —								
	In Series 1 🌲	In Parallel 1 🌲	S	elect vour l	Preferred Displa	v Plot:	All Step	s v	0						
	Capacity	225.0 mAh		· ·		·			-						
									Consumptio	on Profile	by Step				
	Self Discharge	0.12 %/month		6					5:RUN		7:RUN				
	Nominal Voltage	3.0 V		25	1-RUN										
	Ŭ.			é °	11:KUN										
	Max Cont Current	3.0 mA		(Full 2											
				£ 3				4:RUN						10:RUN	
	Information Notes		>	Ê			3.0	VS DBY					9:WS	OP1	
	Information Notes		<u></u>	ns 2			1			6:SLEEP			1		
	Help		>	5.1		2:STAN	DBY			1		8:ST0	P1		11:STANDBY
	rieip		<u>^</u>			Ļ			L			+			Ļ
				ا ں			-						_		
				0.0	0.5 1.0	1.5	2.0	2.5 3.0	3.5 4.0	4.5 5.0		6.5	7.0 7.	5 8.0	8.5 9.0
										Time (ms	()				
									-Idd by Step	- Averad	e Current				
			S	equence	Time / Ta Ma	ax 9 ms	s / 104.	36 °C					Average	Consumpti	ion 2.16 mA





STM32L476RGTx		\sim
Series Line <u>Datasheet</u>	STM32L4 STM32L4x6 025976_Rev4	
T _A 25°C / V _{DD} 2.4V		~
T _{Ambient} V _{DD}	25°C 2.4	\sim
Li-MnO2(CR2032) (1)	(1)	\sim
Change	Reset	
In Series 1 🜲	In Parallel 1 韋	
Capacity	225.0 mAh	
Self Discharge	0.12 %/month	
Nominal Voltage	3.0 V	
Norminal Voltage	3.0 V	
Max Cont Current	3.0 v 3.0 mA	



- 5. Enable the **Transition checker** to ensure the sequence is valid (see *Figure 539*). This option allows verifying that the sequence respects the allowed transitions implemented within the STM32L476RG.
- 6. Click the Add button to add steps that match the sequence described in *Figure 539*.
 - By default the steps last 1 ms each, except for the wake-up transitions preset using the transition times specified in the product datasheet (see *Figure 540*).
 - Some peripherals for which consumption is unavailable or negligible are highlighted with '*' (see *Figure 540*).

Step	Mode	Vdd	Range/Scale	CPU/Bus Freq	Clock Config	Peripherals	Step Current	Duration
1	RUN	2.4	Range1-High	24 MHz	HSE	ADC1:fs_5	5.9 mA	1 ms
2	STANDBY	2.4	NoRange	0 Hz	LSI RTC	RTC*	464 nA	1 ms
3	WU_FROM	2.4	NoRange	4 MHz	MSI FAST		1.7 mA	20.1 µs
4	RUN	2.4	Range1-High	16 MHz	HSE	RTC	2.16 mA	1 ms
5	RUN	2.4	Range2-Med	16 MHz	HSE	ADC1:fs_5	4.47 mA	1 ms
6	SLEEP	2.4	Range2-Med	16 MHz	HSE		589 µA	1 ms
7	RUN	2.4	Range2-Med	16 MHz	HSE	ADC1:fs_5	4.47 mA	1 ms
8	STOP1	2.4	NoRange	0 Hz	ALL CLOCK	USART1*	6.65 µA	1 ms
9	WU_FROM	2.4	NoRange	16 MHz	HSI16		1.62 mA	6.3 µs
10	RUN	2.4	Range2-Med	16 MHz	HSE	RTC USART1	1.89 mA	1 ms
11	STANDBY	2.4	NoRange	0 Hz	LSI RTC		464 nA	1 ms

Figure 540. Sequence table

7. Click the **Save** button to save the sequence as SequenceOne.

The application consumption profile is generated. It shows that the overall sequence consumes an average of 2.01 mA for 9 ms, and that the battery lifetime is only four days (see *Figure 541*).

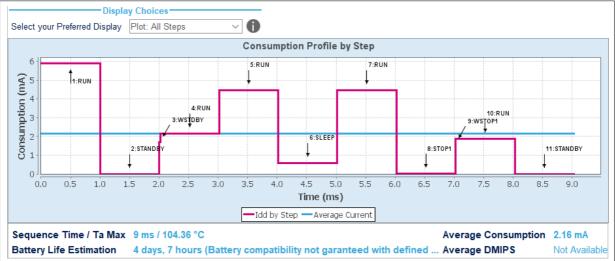


Figure 541. sequence results before optimization

13.3.2 Optimizing application power consumption

Let us now take actions to optimize the overall consumption and the battery lifetime. These actions are performed on steps 1, 4, 5, 6, 7, 8 and 10.

The next figures show on the left the original step, and on the right the step updated with optimization actions.



Step 1 (Run)

Findings

All peripherals are enabled although the application requires only the RTC.

- Actions
 - Lower the operating frequency
 - Enable only the RTC peripheral
 - To reduce the average current consumption, reduce the time spent in this mode
- Results

The current is reduced from 9.05 to 2.16 mA (see Figure 542).

Edit Step	w Edit Step
Reset Step Settings Enable All IPs Disable All IPs Enable IPs from Pinout	Reset Step Settings Enable All IPs Disable All IPs Enable IPs from Pinout
Power Mode RUN Peripherals Power Mode RUN Peripherals Power Mode RUN Peripherals Nemory Fetch Type FLASHIARTICache Peripherals Volage Source Battery If s. 10, ksps Clocks If s. 10, ksps If s. 10, ksps CPU Frequency 24 MHz Pass J. Msps User Choice (Hz) If s. 10, ksps If s. 10, ksps Clock Source Frequency 24 MHz If s. 10, ksps Clock Source Frequency If S. 5, Msps If s. 5, Msps Clock Source Frequency If S. 5, Msps If s. 5, Msps Clock Source Frequency If S. 5, Msps If s. 5, Msps Clock Source Frequency If Miz If s. 1, Msps Clock Source Frequency If Miz If s. 10, ksps Step Duration If max If out If 40112 Clock Source Frequency If Max If out If 40114 Without Peripherals 318 mA If out If 40114 If out If 40114 Peripherals Part Zizz mA (A: 258 mA - D: 141.6 µA) If out If 40114 The step consumption is higher than the max continuous current (3 mA) of the selected	Att Power Mode RUN - IZC3 RTC Power Range Range1-High - IZC3 RTC Power Range Range1-High - ICO Buffer_OFF Voo 2.4 - ICO ILO ILO Votage Source Battery - ILPTIMA - ILPTIMA Votage Source Battery - ILPTIMA - ILPTIMA User Choice (Hz) - ILPUBOR - ILPUBOR User Choice (Hz) - - Normal - PWer Globe Clock Configuration HSE - Normal - PWBOR User Choice (Hz) - - PUBOR - PWR - Quadost Step Duration 0.1 ms - RTC - SAH - SAH Pauffer Step Consumption E16 mA - SOMMC1 - SPH Pauffer Step Consumption E16 mA - SPH - SNM Pauffer
Available use cases: 1 Max: 520	Available use cases: 1 Max: 520

Figure 542. Step 1 optimization

Step 4 (Run, RTC)

Action

Reduce the time spent in this mode to 0.1 ms



Step 5 (Run, ADC, DMA, RTC)

- Actions
 - Change to Low-power run mode
 - Lower the operating frequency
- Results

The current consumption is reduced from 6.17 mA to 271 µA (see Figure 543).

		1 igure 040. Ote	p 5 optimization
🚾 Edit Step			Optimized settings
Reset Step Settings	Enable All IPs Disable All IPs E	nable IPs from Pinout	Edit Step
	Power/Memory	Peripherals Selection	
Power Mode	RUN ~	Peripherals	Reset Step Settings Enable All IPs Disable All IPs Enable IPs from Pinout
Power Range	Range2-Medium V	Y-ADC1	Power/Memory Peripherals Selection
		fs_10_ksps	Power Mode LOWPOWER_RUN V Peripherals
Memory Fetch Type	FLASH/ART/Cache V	- s_1_Msps	Power Range
V _{DD}	2.4 ~	Is_5_Msps	fs_10_ksps
Voltage Source	Battery ~	Y-ADC2	Memory Fetch Type FLASH/ART/Cache fs_1_Msps
vollage Source	Dattery	fs_10_ksps	V _{DD} 2.4
	Clocks	- s_1_Msps	Voltage Source Battery V ADC2
CPU Frequency	16 MHz 🗸	✓ fs_5_Msps	fs_10_ksps
Interpolation Ranges		Y-ADC3	Clocks fs_1_Msps
Interpolation Ranges	Ŭ	fs_10_ksps	CPU Frequency 2 MHz fs_6_Msps
User Choice (Hz)		- C fs_1_Msps	Interpolation Ranges V ADC3
Clock Configuration	HSE V	_ ✓ fs_5_Msps	User Choice (Hz)
Clock Configuration		AHB_APB1_Bridge	fs_1_Msps
Clock Source Frequency	y 16 MHz 🗸	AHB_APB2_Bridge	Clock Configuration MSI fs_5_Msps
	Optional Settings		Clock Source Frequency 2 MHz
Step Duration			Optional Settings — AHB_APB2_Bridge
		∽ DAC1	
Additional Consumption	0 mA ~	OUT1+OUT2-Buffer	Step Duration 1 ms ~ CRC
	Results	OUT1+OUT2-Buffer	Additional Consumption 0 mA V DAC1
Step Consumption 4.4	7 mA	OUT1+OUT2-Buffer	Results OUT1+OUT2-Buffer
	1 mA	OUT1-Buffer_OFF-	Step Consumption 271 µA
		OUT1-Buffer_ON-N	OUT1+OUT2-Buffer
	6 mA (A: 2.58 mA - D: 76.8 μA)	OUT1-Buffer_ON-V	Without Peripherals 271 µA
Ta Max (°C) 104	4.52		Peripherals Part 0 nA (A: 0 nA - D: 0 nA) - OUT1-Buffer_ON-N
The star encounting is t	higher than the max continuous current (3 m	Warnings	Ta Max (°C) 104.97 OUT1-Buffer ON-V
The step consumption is r	nigher man me max continuous current (3 m	A) of the selected battery.	Warnings

Figure 543. Step 5 optimization



Step 6 (Sleep, DMA, ADC, RTC)

- Actions
 - Switch to Lower-power sleep mode (BAM mode)
 - Reduce the operating frequency to 2 MHz
- Results

The current consumption is reduced from 703 µA to 93 µA (see Figure 544).

Figure 544. Step 6 optimization	Figure	544.	Step	6	optimization
---------------------------------	--------	------	------	---	--------------

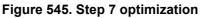
Edit Step	Optimized settings	
Reset Step Settings Enable All IPs Disable All IPs Enable	IPs from Pinout 🗰 Edit Step	
Power/Memory Power/Memory Power Mode SLEEP Power Range Range2-Medium Flash Status Vo Vo Vo Clocks Bus Frequency Tis Mitz	Image Peripherals Selection Peripherals Selection Peripherals Selection Peripherals Power/Memory Power/Memory Peripherals Power/Memory Power/Memory Power/Memory Peripherals Power/Memory Power/Memory Power/Memory Peripherals Power/Memory Peripherals	
Optional Settings Step Duration 1 ms ✓ Additional Consumption 0 mA ✓ Results Step Consumption 509 µA ✓ Without Peripherals 589 µA ✓ ✓ Peripherals Part 0 nA. (A: 0 nA - D: 0 nA) ✓	□ Bus-Matrix □ CAN1 □ CAN1 □ Optional Settings □ CRC Step Duration ↓ DAC1 Additional Consumption ↓ OUT1+OUT2-Buffer Additional Consumption ↓ OUT1+OUT2-Buffer Bus-Matrix ↓ OUT1+Buffer OFF- Without Peripherals S3.4 µA ↓ DitT1-Buffer ON-M Peripherals Part	



Step 7 (Run, DMA, RTC, USART)

- Actions
 - Switch to Low-power run mode
 - Use the power efficient LPUART peripheral
 - Reduce the operating frequency to 1 MHz using the interpolation feature
- Results

The current consumption is reduced from 1.92 mA to 42 μ A (see *Figure 545*).



MX New Step			Optimized Settings
Reset Step Settings Enable All IPs Disable All IPs			🚾 Edit Step
Power/Memory	Peripherals Selection	Enal	
Power Mode RUN	COMP_Ultra_Low_Pov		
Power Range Range2-Medium	COMP_Ultra_Low_Pov		Power/Memory Peripherals Selection Finabled Peripherals Selection
	COMP_Ultra_Low_Pov		Power Mode LOWPOWER_RUN - DFSDM1
Memory Fetch Type FLASH/ART/Cache	COMP_Ultra_Low_Pov		Power Range V - DMA1
V _{DD} 2.4	COMP_Ultra_Low_Pov	•	Memory Fetch Type FLASH/ART/Cache V DMA2
Voltage Source Battery	- 🗆 TIM1		V _{DD} FLASH
	TIM2		
Clocks	- 🗌 TIM3		
CPU Frequency 16 MHz	~ — — TIM4		Clocks Clocks
Interpolation Ranges	TIM6		CPU Frequency User-defined V GPIOC
	П ТІМ6		Interpolation Ranges 100 kHz - 2 MHz
User Choice (Hz)	- 🗌 TIM7		User Choice (Hz) 100000 GPC - GPCH
Clock Configuration HSE	- I TIM8		
Clock Source Frequency 16 MHz	- C TIM15		Clock Configuration
	TIM16		Clock Source Frequency 100 kHz
Optional Settings	- 🗌 TIM17		Optional Settings
Step Duration 1 ms	- TS		Step Duration
Additional Consumption 0 mA	TSC		
	UART4		Additional Consumption 0 mA ~ LPTIM1
Results	UART5		Results VI LPUART1
Step Consumption 1.89 mA	- 🔽 USART1		Step Consumption 42.06 µA Y OPAMP1
Without Peripherals 1.81 mA	USART2		Without Peripherals 41.8 µA
Peripherals Part 84.8 μA (A: 0 nA - D: 84.8 μA)	USART3		Peripherals Part 260 nA (A: 0 nA - D: 260 nA)
	USB_OTG_FS		Ta Max (*C) 105 \/ OPAMP2
Ta Max (°C) 104.8	- WWDG		



Step 8 (Stop 0, USART)

- Actions
 - Switch to Stop1 low-power mode
 - Use the power-efficient LPUART peripheral
- Results

The current consumption is reduced (see Figure 546).

Figure	546.	Step 8	optimization

MX New Step		_	🔤 Edit Step	Optimized Settings
	Enable All IPs Disable All IPs Enabl	and the second se	Reset Step Settings E	Enable All IPs Disable All IPs Enable IPs from Pinout
	Power/Memory	Peripherals Selection En		
Power Mode	STOP0 ~	COMP_Medium_Power USART1*	P0	
Power Range	NoRange ~	COMP_Medium_Power	Power Mode	STOP1 V - GPIOB*
		COMP_Medium_Power	Power Range	NoRange
Memory Fetch Type	n/a 🗸	COMP_Medium_Power		
V _{DD}	3.0 ~	COMP_Medium_Power	Memory Fetch Type	
Voltage Source	Battery	COMP_Medium_Power	V _{DD}	2.4 12C1*
votage source	Battery	COMP_Medium_Power	Voltage Source	Battery
	Clocks	COMP_Medium_Power	Vollage Source	
CPU Frequency	0 Hz ~	COMP_OFF_VREFBUF		- Clocks I I IWDG
Interpolation Ranges		COMP_OFF_VREFBUF	CPU Frequency	0 Hz V LCD*
Interpolation ranges		COMP_OFF_VREFBUF		LPTIM1*
User Choice (Hz)		COMP_Ultra_Low_Pow	Interpolation Ranges	— □ LPTIM2*
Clock Configuration	ALL CLOCKS OFF	COMP_Ultra_Low_Pow	User Choice (Hz)	- V LPUART1*
-		COMP_Ultra_Low_Pow	a a. c	ALL CLOCKS OFF
Clock Source Frequency	0 Hz 🗸	COMP_Ultra_Low_Pow	Clock Configuration	Low_Power
	Optional Settings	COMP_Ultra_Low_Pow	Clock Source Frequency	0 Hz V Normal
Step Duration 1	ms ~	COMP_Ultra_Low_Pow		tional Settings
Additional Consumption 0		COMP_Ultra_Low_Pow	-	Low_Power
Additional Consumption	mA ~	COMP_Ultra_Low_Pow	Step Duration 1	ms V Normal
	Results	- 🗌 UART4*	Additional Consumption 0	mA V PVD/BOR
Step Consumption 111	AL	- UART5*		Results RTC*
Without Peripherals 111	A	– 🗹 USART1*		- SWPMI1*
		- USART2*	Step Consumption 6.65 µ	✓ SYS-VREFBUF/COMP1
	(A: 0 nA - D: 0 nA)	- USART3*	Without Peripherals 6.65 µ/	A COMP_High_Spee
Ta Max (°C) 104.9	99	USB_OTG_FS*	Peripherals Part 0 nA ((A: 0 nA - D: 0 nA)



Step 10 (RTC, USART)

- Actions
 - Use the power-efficient LPUART peripheral
 - Reduce the operating frequency to 1 MHz
- Results

The current consumption is reduced from 1.89 mA to 234 μ A (see *Figure 547*).

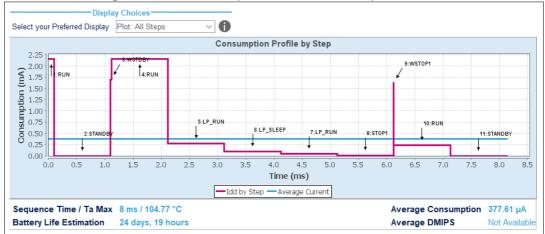
The example given in *Figure 548* shows an average current consumption reduction of 155 μ A.

Edit Step	_	Edit Step Optimized S	ettings
		Coptimized 5	ettings
Reset Step Settings Enable All IPs Disable All IPs Enable	able IPs from Pinout	Reset Step Settings Enable All IPs Disable All IPs En	ble IPs from Pinout
Power/Memory	Peripherals Selection — Enabled P	Power/Memory	Peripherals Selection — Enabled Periphe
Power Mode RUN V	Peripherals RTC USART1	Power Mode	OUT2-Buffer_ON-V RTC LPUART1
Power Range Range2-Medium V	fs 10 ksps	Power Range Range2-Medium V	- DFSDM1
Memory Fetch Type FLASH/ART/Cache V	- fs_1_Msps	Memory Fetch Type FLASH/ART/Cache ~	- DMA1
V _{DD} 2.4 ~	fs_5_Msps	V _{DD} 2.4 ~	- FW
Voltage Source Battery ~	✓ ADC2 ☐ fs_10_ksps	Voltage Source Battery V	
Clocks	fs_1_Msps	Clocks	
CPU Frequency 16 MHz V	└── ☐ fs_5_Msps └── ADC3	CPU Frequency	
Interpolation Ranges	fs_10_ksps	Interpolation Ranges	- GPIOH - I2C1
User Choice (Hz)	fs_1_Msps	User Choice (Hz)	- 12C2
Clock Configuration HSE V	fs_5_Msps	Clock Configuration HSE ~	- 🗆 12C3
Clock Source Frequency 16 MHz	AHB_APB1_Bridge AHB_APB2_Bridge	Clock Source Frequency 1 MHz ~	- 🗆 IWDG
	- CAN1	Optional Settings	Y-LCD
Optional Settings	- CRC	Step Duration 1 ms ~	Buffer_OFF ILPTIM1
Step Duration 1 ms ~	Y-DAC1		
Additional Consumption 0 mA ~	- OUT1+OUT2-Buffer	Additional Consumption 0 mA ~	
Results	- OUT1+OUT2-Buffer	Results	V OPAMP1
	- OUT1+OUT2-Buffer	Step Consumption 234.2 µA	
Step Consumption 1.89 mA	OUT1-Buffer_OFF-	Without Peripherals 232 µA	- Normal
Without Peripherals 1.81 mA	- OUT1-Buffer_ON-N	Peripherals Part 2.2 µA (A: 0 nA - D: 2.2 µA)	✓ OPAMP2
Peripherals Part 84.8 µA (A: 0 nA - D: 84.8 µA)	OUT1-Buffer ON-V	Ta Max (°C) 104.97	
	Varnings	l	Narnings

Figure 547. Step 10 optimization

See *Figure 548* for the overall results: 7 ms duration, about two months battery life, and an average current consumption of 165.25 μ A.

Use the **compare** button to compare the current results to the original ones saved as SequenceOne.pcs.





UM1718 Rev 47



14 Tutorial 4 - Example of UART communications with an STM32L053xx Nucleo board

This tutorial aims at demonstrating how to use STM32CubeMX to create a UART serial communication application for a NUCLEO-L053R8 board.

A Windows PC is required for the example. The ST-Link USB connector is used both for serial data communications, and firmware downloading and debugging on the MCU. A Type-A to mini-B USB cable must be connected between the board and the computer. The USART2 peripheral uses PA2 and PA3 pins, which are wired to the ST-Link connector. In addition, USART2 is selected to communicate with the PC via the ST-Link Virtual COM Port. A serial communication client, such as Tera Term, needs to be installed on the PC to display the messages received from the board over the virtual communication Port.

14.1 Tutorial overview

Tutorial 4 will take you through the following steps:

- 1. Selection of the NUCLEO-L053R8 board from the **New Project** menu.
- 2. Selection of the required features (debug, USART, timer) from the **Pinout** view: peripheral operating modes as well as assignment of relevant signals on pins.
- 3. Configuration of the MCU clock tree from the Clock Configuration view.
- 4. Configuration of the peripheral parameters from the **Configuration** view
- 5. Configuration of the project settings in the **Project Manager** menu and generation of the project (initialization code only).
- 6. Project update with the user application code corresponding to the UART communication example.
- 7. Compilation, and execution of the project on the board.
- 8. Configuration of Tera Term software as serial communication client on the PC.
- 9. The results are displayed on the PC.

14.2 Creating a new STM32CubeMX project and selecting the Nucleo board

To do this, follow the sequence below:

- 1. Select **File > New project** from the main menu bar. This opens the **New Project** window.
- 2. Go to the **Board selector** tab and filter on STM32L0 series.
- 3. Select NUCLEO-L053R8 and click **OK** to load the board within the STM32CubeMX user interface (see *Figure 549*).





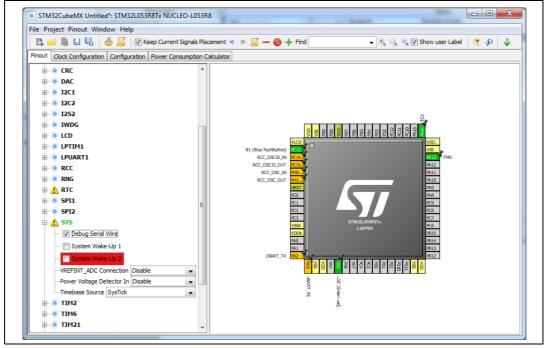
Figure 549. Selecting NUCLEO_L053R8 board



14.3 Selecting the features from the Pinout view

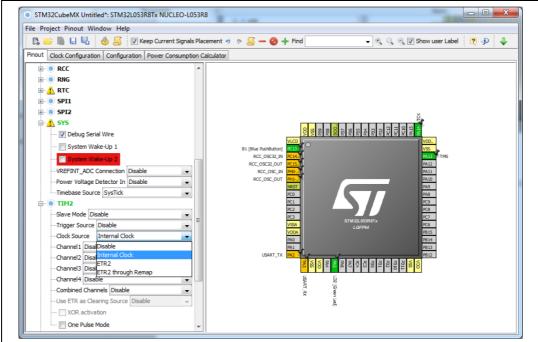
1. Select Debug Serial Wire under SYS (see *Figure 550*).





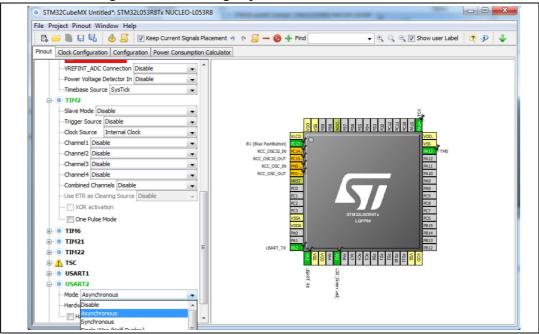
2. Select Internal Clock as clock source under TIM2 peripheral (see *Figure 551*).

Figure 551. Selecting TIM2 clock source





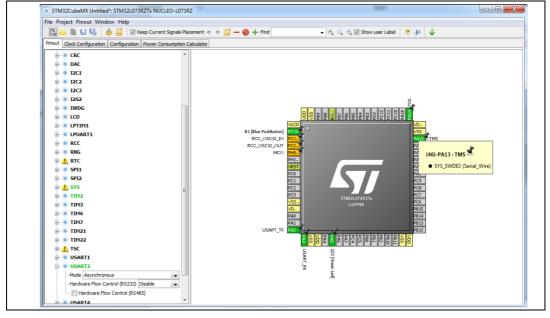
3. Select the Asynchronous mode for the USART2 peripheral (see *Figure 552*).





- 4. Check that the signals are properly assigned on pins (see *Figure 553*):
 - SYS_SWDIO on PA13
 - TCK on PA14
 - USART_TX on PA2
 - USART_RX on PA3







14.4 Configuring the MCU clock tree from the Clock Configuration view

1. Go to the **Clock Configuration** tab and leave the configuration untouched, in order to use the MSI as input clock and an HCLK of 2.097 MHz (see *Figure 554*).

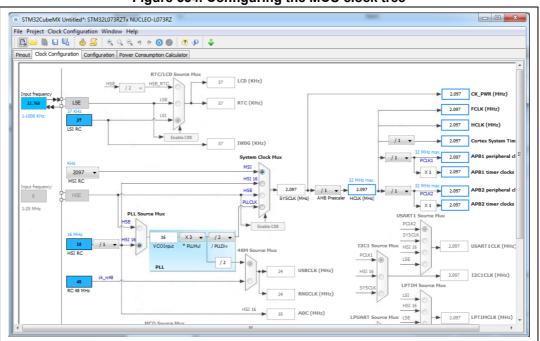


Figure 554. Configuring the MCU clock tree



14.5 Configuring the peripheral parameters from the Configuration view

- 1. From the **Configuration** tab, click **USART2** to open the peripheral **Parameter Settings** window and set the baud rate to 9600. Make sure the Data direction is set to "Receive and Transmit" (see *Figure 555*).
- 2. Click **OK** to apply the changes and close the window.

USART2 Configuration	
Parameter Settings 🔣 User Constants 🛛	🖋 NVIC Settings 🗹 GPIO Settings 🗹 DMA Settings
nfigure the below parameters :	
arch : Search (Crtl+F)	•
1 P	•
Basic Parameters	
Baud Rate	9600 Bits/s
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1
Advanced Parameters	
Data Direction	Receive and Transmit
Over Sampling	16 Samples
Single Sample	Disable
Advanced Features	
Auto Baudrate	Disable
TX Pin Active Level Inversion	Disable
RX Pin Active Level Inversion	Disable
Data Inversion	Disable
TX and RX Pins Swapping	Disable
Overrun	Enable
DMA on RX Error	Disable
MSB First	Disable



3. Click **TIM2** and change the prescaler to 16000, the Word Length to 8 bits and the Counter Period to 1000 (see *Figure 556*).

ionfigure the below parameters : Search : Search (Crtl+F)	
Counter Settings	
Prescaler (PSC - 16 bits value)	16000
Counter Mode	Up
Counter Period (AutoReload Register - 16 bits valu.	. 1000
Internal Clock Division (CKD)	No Division
Trigger Output (TRGO) Parameters	
Master/Slave Mode	Disable (no sync between this TIM (Master) and its Slaves
Trigger Event Selection	Reset (UG bit from TIMx_EGR)
)

Figure 556. Configuring TIM2 parameters



4. Enable TIM2 global interrupt from the **NVIC Settings** tab (see *Figure 557*).

TIM2 Configuration		X
🖋 Parameter Settings 🛛 🎻 User Constants 🛛 🖋 NVIC Settings	🗹 DMA Settings	
Interrupt Table	Enabled	Preemption Priority
TIM2 global interrupt		0
	A	Apply Ok Cancel

Figure 557. Enabling TIM2 interrupt



14.6 Configuring the project settings and generating the project

1. In the **Project Settings** menu, specify the project name, destination folder, and select the EWARM IDE toolchain (see *Figure 558*).

	dvanced Settings
Project Settings	
Project Name	
Nucleo_L073_UART_Comm	n
Project Location	
C:\STM32CubeMX_Project	ts\Tutoriel
Toolchain Folder Location	ts\Tutoriel\Nucleo_L073_UART_Comm
Toolchain / IDE	
EWARM	✓ Generate Under Root
Minimum Heap Size Minimum Stack Size	0x400
	2
Mcu Reference STM32L073RZTx	nd Version
Mcu Reference STM32L073RZTx Firmware Package Name ar	
Mcu and Firmware Package Mcu Reference STM32L073RZTx Firmware Package Name ar STM32Cube FW_L0 V1.7.0	
Mcu Reference STM32L073RZTx Firmware Package Name ar	

Figure 558. Project Settings menu

If the firmware package version is not already available on the user PC, a progress window opens to show the firmware package download progress.



2. In the **Code Generator** tab, configure the code to be generated as shown in *Figure 559*, and click **OK** to generate the code.

Project Settings	
roject Code Generator Advanced Settings	
STM32Cube Firmware Library Package	
Copy all used libraries into the project folder	
Opy only the necessary library files	
\bigcirc Add necessary library files as reference in the toolchain project configuration file	
Generated files	
Generate peripheral initialization as a pair of '.c/.h' files per peripherals	
Backup previously generated files when re-generating	
Keep User Code when re-generating	
Delete previously generated files when not re-generated	
HAL Settings	
Set all free pins as analog (to optimize the power consumption)	
Enable Full Assert	
Template Settings	
Select a template to generate customized code	Settings
Ok	Cancel

Figure	559	Generating	the	code
Iguie	JJJJ.	Ocherating	UIE	COUE

14.7 Updating the project with the user application code

Add the user code as follows:

```
/* USER CODE BEGIN 0 */
#include "stdio.h"
#include "string.h"
/* Buffer used for transmission and number of transmissions */
char aTxBuffer[1024];
int nbtime=1;
/* USER CODE END 0 */
```

Within the main function, start the timer event generation function as follows:

```
/* USER CODE BEGIN 2 */
```



```
/* Start Timer event generation */
    HAL_TIM_Base_Start_IT(&htim2);
    /* USER CODE END 2 */
/* USER CODE BEGIN 4 */
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim){
    sprintf(aTxBuffer,"STM32CubeMX rocks %d times \t", ++nbtime);
HAL_UART_Transmit(&huart2,(uint8_t *) aTxBuffer, strlen(aTxBuffer), 5000);
}
/* USER CODE END 4 */
```

14.8 Compiling and running the project

- 1. Compile the project within your favorite IDE.
- 2. Download it to the board.
- 3. Run the program.

Г

14.9 Configuring Tera Term software as serial communication client on the PC

1. On the computer, check the virtual communication port used by ST Microelectronics from the Device Manager window (see *Figure 560*).

Figure 560. Checking the communication port

A Device Manager	
<u>File Action View Help</u>	
Microsoft Virtual WiFi Miniport Adapter #2	A
D E Portable Devices	
a 🖤 Ports (COM & LPT)	
ECP Printer Port (LPT1)	
STMicroelectronics STLink Virtual COM Port (COM17)	
Processors	=
Security Devices	
Sensors	*



2. To configure Tera Term to listen to the relevant virtual communication port, adjust the parameters to match the USART2 parameter configuration on the MCU (see *Figure 561*).

Port:	COM17 -	ок
Baud rate:	9600 -	
Data:	8 bit 🔹	Cancel
P <u>a</u> rity:	none 🔻	
<u>S</u> top:	1 bit 🔹	<u>H</u> elp
Elow control:	none 🔹	
Transmit dela 0 mse	vy c <u>/c</u> har 0 msec/ <u>l</u> i	ne

Figure 561. Setting Tera Term port parameters

3. The Tera Term window displays a message coming from the board at a period of a few seconds (see *Figure 562*).



File Edit Setup Control Window Help STM32CubeMX rocks 6 times STM32CubeMX rocks 7 times s 8 times STM32CubeMX rocks 9 times STM32CubeMX STM32CubeMX rocks 1 times STM32CubeMX rocks 12 times	📒 COM17 - Tera Term VT		
s 8 times 🛛 STM32CubeMX rocks 9 times 🔹 STM32CubeMX 🚽	<u>File Edit Setup Control</u>	<u>W</u> indow <u>H</u> elp	
	s 8 times STM32	CubeMX rocks 9 times	STM32CubeMX



15 Tutorial 5: Exporting current project configuration to a compatible MCU

When **List pinout compatible MCUs** is selected from the **Pinout** menu, STM32CubeMX retrieves the list of the MCUs which are compatible with the current project configuration, and offers to export the current configuration to the newly selected compatible MCU.

This tutorial shows how to display the list of compatible MCUs and export your current project configuration to a compatible MCU:

1. Load an existing project, or create and save a new project:

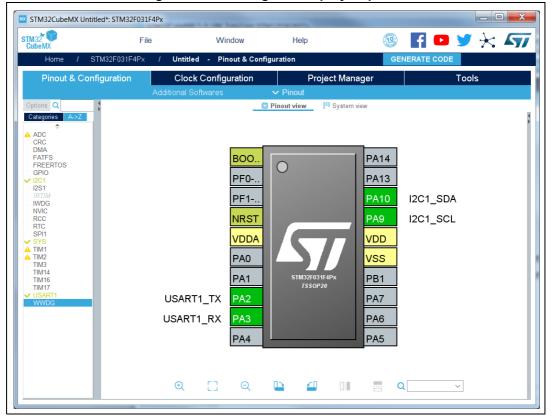


Figure 563. Existing or new project pinout

2. Go to the **Pinout** menu and select **List Pinout Compatible MCUs**. The **Pinout compatible** window pops up (see *Figure 564* and *Figure 565*).

If needed, modify the search criteria and the filter options and restart the search process by clicking the **Search** button.

The color shading and the Comments column indicate the level of matching:

- Exact match: the MCU is fully compatible with the current project (see *Figure 565* for an example).
- Partial match with hardware compatibility: the hardware compatibility can be ensured but some pin names could not be preserved. Hover the mouse over the desired MCU to display an explanatory tooltip (see *Figure 564* for an example).



 Partial match without hardware compatibility: not all signals can be assigned to the exact same pin location and a remapping will be required. Hover the mouse over the desired MCU to display an explanatory tooltip (see *Figure 565* for an example).

Figure 564. List of pinout compatible MCUs - Partial match
with hardware compatibility

Series :	MCUs List: 104 Item:						
	MCU	Package	Flash	Ram	Signals to remap	Comments	Ę
STM32F0 V	STM32F031K6Tx	LQFP32	32	4	2	Need HW change	
Packages :	STM32F030K6Tx	LQFP32	32	4	2	Need HW change	
All 🗸	STM32F031K4Ux	UFQFPN32	16	4	2	Need HW change	
	STM32F031K6Ux	UFQFPN32	32	4	2	Need HW change	
Search Options	STM32F098VCTx	LQFP100	256	32	4	Need HW change	
Ignore Pinning Status	STM32F070C6Tx 🔪	LQFP48	32	6	4	Need HW change	
Ignore Power Pins	STM32F USART1 TX I	remaps from Pii	n(8)-PA2 to	Pin(30)-	PA9	Need HW change	
✓ Ignore System Pins	STM32F USART1_RX	remaps from Pi	n(9)-PA3 t	o Pin(31)-	PA10	Need HW change	
	STM32F I2C1_SCL rer	naps from Pin(1	17)-PA9 to	Pin(6)-PF	1-OSC_OUT	Need HW change	
Onersh	STM32F US 1551A rei	maps from Pin(18)-PA10 t	0 Pin(5)-F	FU-OSC_IN	Need HW change	
Search	STM32F031E6Yx	WLCSP25	32	4	4	Need HW change	
	STM32F030RCTx	LQFP64	256	32	4	Need HW change	
	STM32F030R8Tx	LQFP64	64	8	4	Need HW change	
	STM32F030CCTx	LQFP48	256	32	4	Need HW change	



MCUs Filters	MCUs List: 3 Items	·					
Series :	MCU	Package	Flash	Ram	Signals to remap	Comments	G
All	STM32F030F4Px	TSSOP20	16	4	0	Full Compatible	
Packages :	STM32F031F6Px	TSSOP20	32	4	0	Full Compatible	
TSSOP20	STM32F038F6Px	TSSOP20	32	4	0	Full Compatible	
✓ Ignore System Pins Search							
						K, Import Clo)se

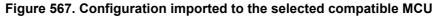


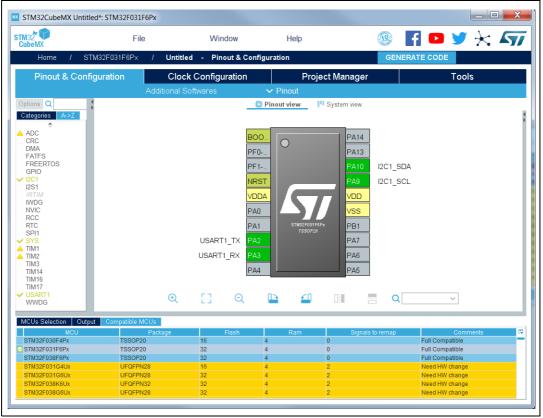
3. Select an MCU to import the current configuration to, and click **OK**, **Import**:

MCUs Filters	MCUs List: 104 Ite	ms				
Series :	MCU	Package	Flash	Ram	Signals to remap	Comments
STM32F0 V	STM32F030F4Px	TSSOP20	16	4	0	Full Compatible
Packages :	STM32F031F6Px	TSSOP20	32	4	0	Full Compatible
All 🗸 🗸	STM32F038F6PX	TSSOP20	32	4	0	Full Compatible
	STM32F031G4Ux	UFQFPN28	16	4	2	Need HW change
Search Options	STM32F031G6Ux	UFQFPN28	32	4	2	Need HW change
Ignore Pinning Status	STM32F038K6Ux	UFQFPN32	32	4	2	Need HW change
Ignore Power Pins	STM32F038G6Ux	UFQFPN28	32	4	2	Need HW change
0	STM32F031K6Tx	LQFP32	32	4	2	Need HW change
Ignore System Pins	STM32F030K6Tx	LQFP32	32	4	2	Need HW change
	STM32F031K4Ux	UFQFPN32	16	4	2	Need HW change
Search	STM32F031K6Ux	UFQFPN32	32	4	2	Need HW change
	STM32F098VCTx	LQFP100	256	32	4	Need HW change
	STM32F070C6Tx	LQFP48	32	6	4	Need HW change
	STM32F070CBTx	LQFP48	128	16	4	Need HW change
	OTMOOFOZODDTy	LOEDRA	100	16	4	Nood LW/ obongo

Figure 566. Selecting a compatible MCU and importing the configuration

The configuration is now available for the selected MCU:







4. To see the list of compatible MCUs at any time, select **Outputs** under the **Window** menu.

To load the current configuration to another compatible MCU, double-click the list of compatible MCUs.

- 5. To remove some constraints on the search criteria, several solutions are possible:
 - Select the **Ignore Pinning Status** checkbox to ignore pin status (locked pins).
 - Select the **Ignore Power Pins** checkbox not to take into account the power pins.
 - Select the **Ignore System Pins** not take into account the system pins. Hover the mouse over the checkbox to display a tooltip that lists the system pins available on the current MCU.



16 Tutorial 6 – Adding embedded software packs to user projects

In this tutorial, the Oryx-Embedded.Middleware.1.7.8. pack is taken as an example to demonstrate how to a to add pack software components to STM32CubeMX projects. The use of this package shall not be understood as an STMicroelectronics recommendation.

To add embedded software packs to your project, proceed as follows:

- Install Oryx-Embedded.Middleware.1.7.8.pack using the .pdsc file available from http://www.oryx-embedded.com (see Section 3.4.5: Installing embedded software packs).
- 2. Select New project.
- 3. Select STM32F01CCFx from the MCU selector.
- 4. Select Additional Software from the Pinout & Configuration view to open the additional software component window and choose the following software components: Compiler Support, RTOS Port/None and Date Time Helper Routines from the CycloneCommon bundle (see Section 4.15: Software Packs component selection window).
- 5. Click **OK** to display the selected components on the tree view and click the checkbox to enable the software components for the current project (see *Figure 568*).

STM32CubeMX Untitl	ed*: STM32F401	CCFx								
STM32 CubeMX	File	e	Window	Help)	19	f 🗖	9	\star	57
Home / ST	FM32F401CCF×	/ Untitled -	Pinout & Conf	iguration		GEN	IERATE COL	DE		
Pinout & Conf	iguration	Clock Cor	nfiguration		Project Manage	r		Tool	s	
				✓ Pinout						
Options Q Categories A->Z	~ /×	Embedded.Middleware.	.1.7.8 Mode and Co ode	onfigurat	Ø F	'inout view	Syster	n view		
System Core	>	CycloneCommon	CycloneCommon	•				Mie	ddlewar	es
Analog	>									
Timers	>							Additio	onal Sof	tware
Connectivity	>								dleware (
Multimedia	>							MIG	dieware	`
Computing	>	Config	guration	- I I	System Core	Analog		Timers	(Connectivi
Middleware	>				DMA 🙏					
Additional Software	~									
Oryx-Embedded.Mic	ddleware									

Figure 568. Additional software components enabled for the current project

The pack name highlighted in green indicates that all conditions for the selected software components resolve to true. If at least one condition is not resolved, the pack name is highlighted in orange.



6. Check that no parameters can be configured in the **Configuration** tab (see *Figure 569*).

Additional Softwares	✓ Pinout
Oryx-Embedded.Middleware.1.7.8 Mode and Configuration Mode	Pinout view System view
CycloneCommon CycloneCommon	Middlewares
Configuration	
No configuration available	Additional Software

Figure 569. Pack software components: no configurable parameters

7. Select the **Project manager** project tab to specify project parameters (see *Figure 570*), and choose IAR[™] EWARM as IDE.

Pinout & Cor	nfiguration	Clock Configuration	Project Manager
			Generate Report
Project	Project Settings Project Name Oryx_project1 Project Location C:\STM32CubeM2 Application Struct		Browse
Code Generator	Basic Toolchain Folder L	✓ □ Do n ocation (_Projects\Oryx_project1\	ot generate the main()
Advanced Settings	Linker Settings Minimum Heap Si Minimum Stack S		
	STM32Cube FW	Name and Version F4 V1.22.0	V1.22.0 Browse

Figure 570. Pack tutorial: project settings



- 8. Generate your project by clicking GENERATE CODE . Accept to download the STM32CubeF4 MCU package if it is not present in STM32Cube repository.
- 9. Click **Open project**. The Oryx software components are displayed in the generated project (see *Figure 571*).

Project - IAR Embedded Workbench IDE - ARM 8.20.1 File Edit View Project ST-Link Tools Window Help		
	(···	
	→ *=	
Norkspace	-	φ ×
project1		•
Files	\$	•
🗆 🛡 project1 - project1 *	~	
		-
		•
		•
H⊞ 🖬 Drivers H⊟ 📹 Middlewares		
□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □		
		•
L → ⊞ 🖸 date_time.c		•
└─── 🖬 CycloneCommon/RTOSOoPort/None		
└─⊞ io os_port_none.c		٠
Le Cutput		

Figure 571. Generated project with third party pack components



17 Tutorial 7 – Using the X-Cube-BLE1 software pack

This tutorial demonstrates how to achieve a functional project using the X-Cube-BLE1 software pack.

Below the prerequisites to run this tutorial:

- Hardware: NUCLEO-L053R8, X-NUCLEO-IDB05A1 and mini-USB cable (see Figure 572)
- Tools: STM32CubeMX, IDE (Atollic[®] or any other toolchain supported by STM32CubeMX)
- Embedded software package: STM32CubeL0 (version 1.10.0 or higher), X-Cube-BLE1 1.1.0 (see *Figure 573*).
- Mobile application (see *Figure 574*): STMicroelectronics BlueNRG application for iOS[®] or Android[™]

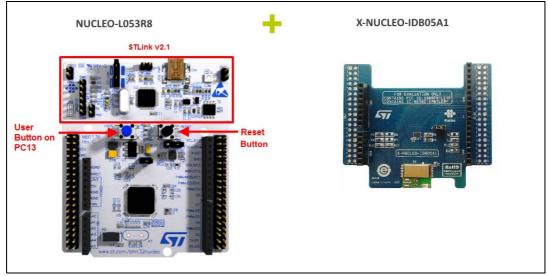
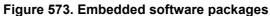


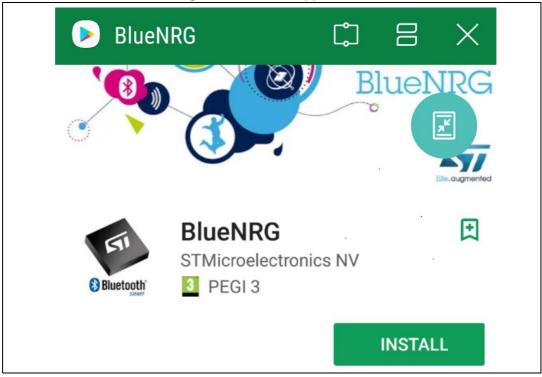
Figure 572. Hardware prerequisites





Embedded Software Pac	kages Manager		X	1
STM32Cube MCU	Packages and embedded software packs r	releases		
Releases Info rmati	on was last refreshed 3 hours ago.		+ -	
STM32Cube MCU Packages	Alibaba Oryx-Embedded STMicroelect	ronics IwIP		
Description		Installed Version	Available Version	
▼ STM32L0				
STM32Cube MCU F	Package for STM32L0 Series	1.10.0	1.10.0	
STM32Cube MCU F	Package for STM32L0 Series	1.9.0	1.9.0	
STM32 Embe	dded Software Packages Manager			X
	STM32Cube MCU Packages and embedd	led software packs releases		
Details	Releases Information was last refreshed 3 ho	ours ago. 🖌		
STM32C	ube MCU Packages Alibaba Oryx-Embed	Ided STMicroelectronics Iw		
	Description		Availa	ble Version
▼ X.(CUBE-BLE1			
From Local	BLE stack and sample applications for BlueN	RG-MS module		1.1.0
► X.0	CUBE-MEMS1			
Details				
From L	.ocal From Url	Refresh Install Now	v Remove Now	Close

Figure 574. Mobile application





Proceed as follows to install and run the tutorial:

- 1. Check STM32CubeMX Internet connection:
 - a) Select the **Help > Updater Settings** menu to open the updater window.
 - b) Verify in the **Connection** tab that the Internet connection is configured and up.
- 2. Install the required embedded software packages (see Figure 575):
 - a) Select the Help > Manage Embedded software packages menu to open the embedded software package manager window.
 - b) Click the **Refresh** button to refresh the list with the latest available package versions.
 - c) Select the **STM32Cube MCU Package** tab and check that the STM32CubeL0 firmware package version 1.10.0 or higher is installed (the checkbox must be green). Otherwise select the checkbox and click **Install now**.
 - d) Select the **STMicrolectronics** tab and check that the X-Cube-BLE1 software pack version 1.0.0 is installed (checkbox must be green). Otherwise, select the checkbox and click **Install now**.

Embedded Software	Packages Manager		X	
STM32Cube I	MCU Packages and embedded software packs relea	ases		
Releases for	rmation was last refreshed 3 hours ago.		+ -	
3TM32Cube MCU Packag	iges Alibaba Oryx-Embedded STMicroelectronic	s IwiP		
Description		Installed Version	Available Version	
STM32L0				
STM32Cube M	ICU Package for STM32L0 Series	1.10.0	1.10.0	
STM32Cube M	ICU Package for STM32L0 Series	1.9.0	1.9.0	
] STM32 🔤 Er	mbedded Software Packages Manager			
Details	STM32Cube MCU Packages and embedded	-		+
	Releases Information was last refreshed 3 hours M32Cube MCU Packages Alibaba Oryx-Embedded	· ·	IwiP	
310	Description	Stivictoelectronics		ilable Version
•	X-CUBE-BLE1			
From Local	BLE stack and sample applications for BlueNRG-	MS module		1.1.0
•	X-CUBE-MEMS1			
Deta	ails			
			/	
	rom Local From Url	Refresh Install N	ow Remove Now	Close

Figure 575. Installing Embedded software packages

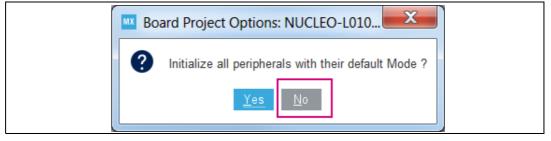
- 3. Start a new project:
 - a) Select New Project to open the new project window.
 - b) Select the **Board selector** tab.
 - c) Select Nucleo64 as board type and STM32L0 as MCU Series.
 - d) Select the NUCLEO-L053R8 from the resulting board list (see *Figure 576*).
 - e) Answer **No** when prompted to initialize all peripherals in their default mode (see *Figure* 577).



New Project from a Board		X
MCU Selector Board Selector		
٩	Large Pic Docs & Resour 🕒 Datas 📑 🕞 Star	t Pro
Vendor >	☆ 「NUCLEO-L010RB	
Type ~	STMicroelectronics NUCLEO-L010RB Board Support and Examples	t
Discovery	STM32 L0	
Evaluation Board	Mounted device: STM32L010RBTx	
Nucleo32		
MCU Series ~ Check/Uncheck All	Boards List: 3 items Overview Part No Type Markett. Unit Price Mounted MCU SerCustom For Memory	ROM
STM32F1		0
STM32F4	☆ NUCLEO-L0 NucActive 13.0 STM32L. STM32L0.0 0	0
▼ STM32L0 □ STM32L1		

Figure 576. Starting a new project - selecting the NUCLEO-L053R8 board

Figure 577. Starting a new project - initializing all peripherals



- 4. Add X-Cube-BLE1 components to the project:
 - a) Click Additional Software from Pinout & Configuration view to open the Additional Software component Selection window.
 - b) Select the relevant components (see *Figure 578*)

The Application group comes with a list of applications: the C files implement the application loop, that is the *Process()* function. From the Application group, select the **SensorDemo** application.

Select the Controller and Utils components

Select the **Basic** variant for the **HCI_TL** component. The Basic variant provides the STMicroelectronics implementation of the HCI_TL API while the template option requires users to implement their own code.

Select the **UserBoard** variant as **HCI_TL_INTERFACE** component. Using the UserBoard option generates the <boardname>_bus.c file, that is nucleo_I053r8_bus.c for this tutorial, while the template option generates the custom bus.c file and requires users to provide their own implementation.

Refer to the X-Cube-BLE1 pack documentation for more details on software components.



c) Click **OK** to apply the selection to the project and close the window. The left panel **Additional Software** section is updated accordingly.

🛱 🗞 🏮 > Show compone	nts for cont	ext: Cortex-M7	~		
Pack / Bundle / Component	Status	Version	Selection		
> RoweBots.I-CUBE-UNISONRTOS		5.5.0-4 ڬ 🚱	Install		
> SEGGER.I-CUBE-embOS		1.2.0 😉			
> STMicroelectronics.X-CUBE-AI		6.0.0 ~			
> STMicroelectronics.X-CUBE-ALGOBUILD		1.2.0 ~			
> STMicroelectronics.X-CUBE-AZRTOS-H7		1.0.0			
 STMicroelectronics.X-CUBE-BLE1 	\odot	6.2.0 ~			
✓ Wireless BlueNRG-MS	\odot	5.1.0			
BlueNRG-MS / Controller	\odot				
BlueNRG-MS / HCI_TL	\odot		Basic	~	
BlueNRG-MS / HCI_TL_INTERFACE	\odot		UserBoard	~	
BlueNRG-MS / Utils	\odot		✓		
✓ Device BLE1_Applications	\odot	6.1.0			
Application	\odot		SensorDemoBLESensor	~	
> STMicroelectronics.X-CUBE-BLE2		3.2.0 ~			

Figure 578. Selecting X-Cube-BLE1 components

- 5. Enable peripherals and GPIOs from the **Pinout** tab (see *Figure 579*):
 - a) Configure **USART2** in **Asynchronous** mode.
 - b) Configure **SPI1** in **Full-duplex master** mode.
 - c) Left-click the following pins and configure them for the required GPIO settings:
 PA0: GPIO_EXTI0
 - PA1: GPIO_Output

PA8: GPIO_Output

d) Enable **Debug Serial Wire** under **SYS** peripheral.







Figure 579. Configuring peripherals and GPIOs

- 6. Configure the peripherals from the **Configuration** tab:
 - a) Click the **NVIC** button under the **System** section to open the **NVIC configuration** window. Enable EXTI line 0 and line 1 interrupts and click **OK** (see *Figure 580*).
 - b) Click the SPI button under the Connectivity section to open the SPI configuration window. Check that the data size is set to 8 bits and the prescaler value to 16 so that HCLK divided by the prescaler value is less or equal to 8 MHz.
 - c) Click **USART2** under the **Connectivity** section to open the **Configuration** window and check the following parameter settings:

Under Parameter Settings:

Baud rate: 115200 bits/s

Word length: 8 bits (including parity)

Parity: none

Stop bits: 1

Under GPIO Settings:

User labels: USART_TX and USART_RX



STM32CubeMX Ur	ntitled*: STM32L010RBTx NUCLE	O-L010RB					X
TM32 TM CubeMX	File	Window	Help		f 🖻		57
Home /	STM32L010RBTx - NUCLEO	L010RB / Untitled	- Pinout & Configuration	n GENE	RATE CODE		
Pinout & Co	onfiguration Clo	ock Configuration	Project Manag	jer	Т	ools	
	Additiona	Softwares	✓ Pinout				
Options Q			NVIC Mode and Configuration				
Categories A->Z			Configuration				
System C ~	📀 NVIC 🛛 📀 Code generati	on					
				Sor	t by Premption P	riority and Sub	Priority
DMA GPIO	Search Search (CrtI+F)			💿 💿 🗖 Sho	w only enabled in	nterrupts	
		NVIC Interrupt Ta	ble		Enabled Pr	eemption Priorit	ty
V RCC	Non maskable Interrupt Hard fault interrupt				✓ 0 ✓ 0		
A SYS WWDG	System service call via SWI instru	ction					
- WWWDG	Pendable request for system ser	ice			V 0		
	Time base: System tick timer Flash and EEPROM global interru				✓ 0		
	RCC global interrupt	pi					
Analog >	EXTI line 0 and line 1 interrupts				V 0		
	EXTI line 4 to 15 interrupts				0		
Timers >	SPI1 global interrupt				0		
	USART2 global interrupt / USART	2 wake-up interrupt through EXTI	line 26		0		
Connectivity~							
÷							
I2C1							
LPUART1							
V SPI1							
A USART2							
			Enabled Preem	ption Priority 0	-		

Figure 580. Configuring NVIC interrupts

- 7. Enable and configure X-Cube-BLE1 pack components from the **Pinout & Configuration** view:
 - a) Click the pack items from the left panel to show the mode and configuration tabs.
 - b) Click the check boxes from the Mode panel to enable X-Cube-BLE1, the configuration panel appears showing the parameters to configure. An orange triangle indicates that some parameters are not configured. It turns into a green check mark once all parameters are correctly configured (see *Figure 581*).
 - c) Leave the Parameter Settings Tab unchanged.
 - d) Go the Platform settings tab, configure the connection with the hardware resources as indicated in *Figure 581* and *Table 26*.

Name	IPs or components	Found solutions
BUS IO driver	SPI in Full-duplex master mode	SPI1
EXTI Line	EXTI Line GPIO:EXTI	
CS Line	GPIO:output	PA1
Reset Line	GPIO:output	PA8
BSP LED	GPIO:output	PA5
BSP Button	GPIO:EXTI	PC13
BSP USART	USART in Asynchronous mode	USART2

Table 26. Connection with hardware resources

Check that the icon turns to
Click OK to close the Configuration window.



M32 Total State	File	Window	Help		🐵 🗗 🕒) 🄰 🔆 🏹
	BTx - NUCLEC	D-L010RB / Untitled	- Pinout & Configuration	n	GENERATE CO	DE
Pinout & Configuratio	n	Clock Configuratio		roject Manager		Tools
ptions Q	~	Additional Softwares STMicroelectr	✓ Pinout ronics.X-CUBE-BLE1.1.1.0 Mod	e and Configuration	💭 Pinout vie	ew System view
Categories A->Z System Core	>	Vireless BlueNRG-MS	Mode		\$Middl	lewares
Analog Timers	> >		Configuration		-	
Connectivity	>	Reset Configuration Parameter Settings Image: Configuration Platform proposal Image: Configuration	User Constants 💿 Platform	Settings		al Software -BLE1 ⊘
Middleware	>	HCI_TL_INTERFACE Name IPs or Compor Exti Line GPIO:EXTI	nents Found Solutions	BSP API	ER Tir	mers Connectiv
Additional Software	.1.1.0	BUS IO driver SPI:Full-Duplex CS Line GPIO:Output Reset Line GPIO:Output	Master V SPI1 V PA1 V PA8	BSP_BUS_DRIVE	ER	SPI1 🕑 USART2 (
		BSP Name IPs or Compo BSP BUTTON GPIO-EXTI BSP USART USART-Asynch BSP LED GPIO-Output	V PC13 [B1 [Blue Push	BSP API But V BSP_COMMON_DRIVI	ER	

Figure 581. Enabling X-Cube-BLE1

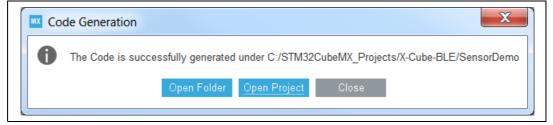
- 8. Generate the SensorDemo project:
 - a) Click GENERATE CODE to generate the code. The **Project Settings** window opens if the project has not yet been saved.
 - b) Click **GENERATE CODE** to generate the code once the project settings have been properly configured (see *Figure 582*). When the generation is complete, a dialog window requests to open the project folder (Open Folder) or to open the project in IDE toolchain (Open Project). Select **Open Project** (see *Figure 583*).



Pinout & Con	figuration	Clock Configura	ation	Project Ma	nager	Tools
				Generate F		
	Project Settings-					
	Project Name					
Project	SensorDemo					
	Project Location					
	C:\STM32CubeM	IX_Projects\X-Cube-BLE			Browse	
	Application Struct	ture				
	Basic		✓ □ Do not ge	enerate the main()		
	Toolchain Folder	Location				
Code Generator		IX_Projects\X-Cube-BLE\Senso	rDemo\			
	Toolchain / IDE					
	TrueSTUDIO		✓ Generat	e Under Root		
		ware Library Package				
		l libraries into the project folder				
Code Generator		e necessary library files				
	O Add necessar	ry library files as reference in th	e toolchain project c	onfiguration file		
	Generated files —					
		ipheral initialization as a pair of		eral		
		ously generated files when re-ge	enerating			
	-	ode when re-generating				
	🔽 Delete previou	usly generated files when not re	-generated			
	Driver Selector					
	Q Search (CrtI+F)	\odot	\odot			
dvanced Settings	RCC				HAL	
uvanceu Seuriys		cs.X-CUBE-BLE1.1.1.0			HAL	
	GPIO				HAL	
	Generated Function	Calls				
	Rank	Function Name	IP Insta	nce Name	Not Generate Function (Call 🔲 Visibility (Sta
	1	MX GPIO Init	GPIO			
	2 3	SystemClock_Config MX_X_CUBE_BLE1_Init	RCC	X-CUBE-BLE1.1.1.0	8	

Figure 582. Configuring the SensorDemo project

Figure 583. Open SensorDemo project in the IDE toolchain





18 Creating LPBAM projects

18.1 LPBAM overview

Disclaimer: to learn about the LPBAM mode and its usage, it is recommended to read the LPBAM application note available on *www.st.com*, and the LPBAM utility getting started guide located under the Utilities folder of the STM32Cube firmware package.

18.1.1 LPBAM operating mode

LPBAM stands for low power background autonomous mode. It is an operating mode that allows peripherals to be functional and autonomous independently from power modes and without any software running. It is performed thanks to a hardware subsystem embedded in STM32 products. Thanks to DMA transfers in Linked-list mode, the LPBAM subsystem can chain different actions to build a useful functionality (peripheral configurations and transfers). Optionally, it can generate asynchronous events and interrupts. It operates without any CPU intervention. Consequently, the two major benefits from using the LPBAM subsystem mechanisms are an optimized power consumption, and an offloaded CPU.

18.1.2 LPBAM firmware

The LPBAM firmware has been designed to help users create LPBAM applications: the LPBAM utility is a set of modular drivers located under the Utilities folder of the STM32Cube firmware package. Each module comes as a pair of C file that provides the APIs needed to build an application scenario. Each module manages the configurability and the data transfers for a given peripheral. The LPBAM utility is designed to be compatible with any STM32 devices supporting LPBAM subsystem mechanisms through a configuration module: it requires a configuration file stm32_lpbam_conf.h aligned with the application needs. The LPBAM utility has a single application entry point, the stm32_lpbam.h, that must be included in the project.

18.1.3 Supported series

The LPBAM firmware supports STM32U575/585, STM32U595/5A5 and STM32U599/5A9 products, for projects with or without TrustZone[®] activated.

STM32CubeMX 6.5.0 introduces LPBAM for projects without TrustZone[®] activated on the STM32U575/585 product line: users can create LPBAM applications for their project using STM32CubeMX LPBAM Scenario & Configuration view and generate the corresponding code. The generated C project embeds the LPBAM firmware.

STM32CubeMX 6.6.0 adds LPBAM support for projects with TrustZone[®] activated.



18.1.4 LPBAM design

It is recommended to use LPBAM to save power and offload the CPU.

- The LPBAM mechanism supports the following set of peripherals on the Smart Run Domain: ADC4, COMP1/2, DAC1, I2C3, LPDMA1, LPGPIO, LPTIM1/2/3, LPUART1, OPAMP1/2, SPI3, VREFBUF.
- According to the LPDMA implementation in the Smart run domain, the LPBAM has access only to SRAM4.
- The LPBAM mechanism implementation can run autonomously until Stop2 mode.
- To reach the lowest power consumption, the system power usage, the system clock and the autonomous peripheral kernel clock can be configured:

18.1.5 LPBAM project support in STM32CubeMX

An LPBAM project is composed of a main project, and of one or more LPBAM applications.

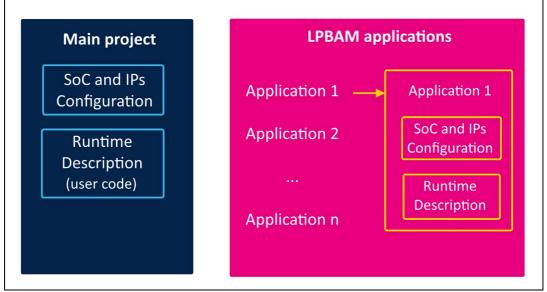


Figure 584. LPBAM project

The "Main project" contains the "SoC and IPs configuration" at initialization time and a runtime description of the main application. STM32CubeMX allows to describe the "SOC and IPs Configuration" part.

Each LPBAM application contains a "SoC and IPs configuration" and a runtime description. STM32CubeMX allows to describe both.

STM32CubeMX generated code for "SoC and IPs configurations" uses the STM32Cube HAL and/or LL APIs, for both the main project and the LPBAM application. The code generated for the LPBAM application runtime uses the LPBAM firmware API.

Figure 585 is an example of what can be executed at runtime for a simple LPBAM project composed of the main application and of one LPBAM application.



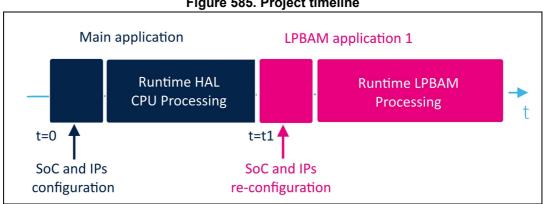
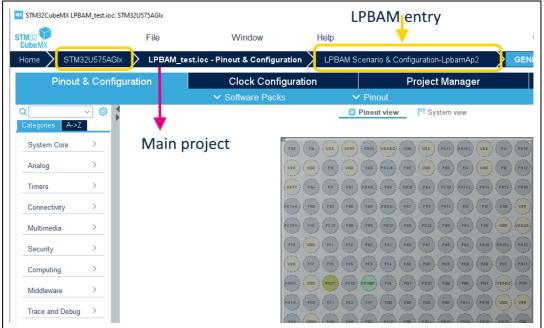


Figure 585. Project timeline

18.2 Creating an LPBAM project

18.2.1 LPBAM feature availability

When a project with LPBAM feature capability is opened, a dedicated entry is shown in the user interface (see *Figure 586*). The feature is optional and when it is not used, it has no impact on the generated project.





18.2.2 Describing an LPBAM project

Describing an LPBAM project in STM32CubeMX consists in describing the main project using STM32CubeMX main project page, and one or more LPBAM applications using the dedicated LPBAM Scenario & Configuration page.



UM1718 Rev 47

Starting with STM32CubeMX 6.5:

- Create a project by selecting an MCU or board part number from the STM32U575/585 product line.
- Do not activate TrustZone[®] for the project.
- Click "LPBAM Scenario & Configuration" ribbon to view LPBAM dedicated page.

The LPBAM context is highlighted with a pink border. You can switch back and forth between the main project configuration and the LPBAM Scenario & Configuration by clicking the corresponding ribbon.

TM32CubeMX Untitled: STM32U575A STM32 CubeMX Home STM32U575AGk	Gix Main project File CLICK TO S File Window Untitled - Pinout & Configuration	SWITCH VIEWS Help LPBAM Scenario & Configuration	- D X
	LPBAM Sc	enario & Configuration	
LPBAM Management ✓ ✓ IPBAM Manager └ ♣ Add Application		reate or activate a LPBAM application first	

Figure 587. LPBAM Scenario & Configuration view

18.2.3 Managing LPBAM applications in a project

When entering the LPBAM Scenario & Configuration view, you must first add an LPBAM application.

Adding, removing, renaming, and switching between LPBAM applications is done from the left panel under the LPBAM manager section.

To add the first LPBAM application, click "Add Application":

- If the default name is kept, the application "LpbamApp1" is created.
- The first Queue "Queue1" of LpbamApp1 is created.
- The configuration views (LPBAM scenario, pinout & ip, clock) necessary to describe Lpbam App1 are available.

To add more queues, click "Add Queue"

To delete an application (or a queue), right-click the application (or the queue) name and select "Delete".



To rename an application (or a queue), right-click the application (or the queue) name and select "Rename". Note that the application name is used in the generated project.

To switch between LPBAM applications, click the application name, this loads the LPBAM panel for the selected application.

To switch between queues in an LPBAM application, click the queue name: the middle and right panels are refreshed to display the selected queue and its configuration.

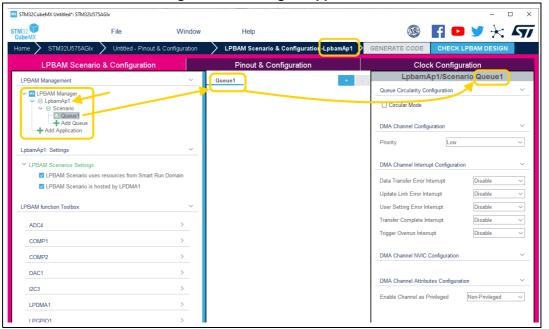


Figure 588. Adding an application

18.3 Describing an LPBAM application

18.3.1 Overview (SoC & IPs configuration, runtime scenario)

Describing an LPBAM application consists in configuring the SoC and IPs, as it is done for a standard STM32CubeMX project, as well as describing the runtime part of the application.

SoC and IPs configuration

To configure IP and SOC in the context of an LPBAM application, use the Pinout & Configuration and Clock configuration provided with the LPBAM application.



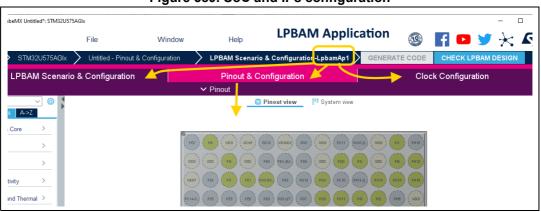


Figure 589. SoC and IPs configuration

Runtime description (scenario)

With standard STM32CubeMX projects, the user must add the code to manage the runtime behavior of the main application based on STM32Cube HAL or LL driver APIs, such as HAL_COMP_Start, HAL_TIM_Start, HAL_TIM_Stop.

For LPBAM applications, STM32CubeMX provides the LPBAM Scenario & Configuration panel to create the runtime description (scenario). As shown in *Figure 590*, this panel is divided in three parts.

STM32CubeMX Untitled*: STM32U575AGIx		/	- D X
STM32 File	Window	Help	🐵 🖪 🕨 🏏 😽 🜆
Home > STM32U575AGIx > Voltitled - Pinout & C	onfiguration	LPBAM Scenario & Configuration-LpbamAp1	GENERATE CODE CHECK LPBAM DESIGN
LPBAM Scenario & Configuration		Pinout & Configuration	Clock Configuration
LPBAM Management	~ [Queue1 -	LpbamAp1/Scenario/Queue1
V 🗰 LPBAM Manager			Queue Circularity Configuration
✓ O LpbamAp1 ✓ O Scenario O Queue1		COMP1:Start_1	Circular Mode
Add Queue		COMP1:OutputLevel 2	DMA Channel Configuration
LpbamAp1: Settings	~		Priority Low ~
✓ LPBAM Scenarios Settings			DMA Channel Interrupt Configuration
LPBAM Scenario uses resources from Smart Run I LPBAM Scenario is hosted by LPDMA1	Domain		Data Transfer Error Interrupt Disable ~
CEPDAM Scenario is nosted by LPDMAT			Update Link Error Interret Disable V
LPBAM function Toolbox	~		User Setting Error Inter upt 3
ADC4	>		Transfer Complete Inter upt
COMP1	~		Trigger Overrun Interrupt Disable ~
Start	+		DMA Channel NVIC Configuration
OutputLevel	+		
			DMA Channel Attributes Configuration $\qquad \qquad \qquad$
COMP2			Enable Channel as Privileged Non-Privileged ~



Note: LPBAM applications use the LPBAM firmware APIs and consist of chained DMA transfers.



In the context of an LPBAM application, the first panel is used for:

- Managing queues for the application.
- Browsing and adding nodes to the queue currently selected in STM32CubeMX user interface.
- Application specific settings. These settings cannot be changed nor disabled when using LPBAM on STM32U5 series.

The second panel displays the diagram of the queue currently selected for one selected queue of the LPBAM application.

The third panel lets the user to configure either the queue (if the queue name is clicked), or a node (if the node is selected on the diagram).

18.3.2 SoC& IPs: configuring the clock

The LPBAM subsystem is functional down to STOP2 mode and supports only IPs on the Smart run domain. Consequently, in the LPBAM context, only a subset of the clock tree can be configured. Refer to *Section 4.10* for details on how to configure a clock tree in STM32CubeMX.

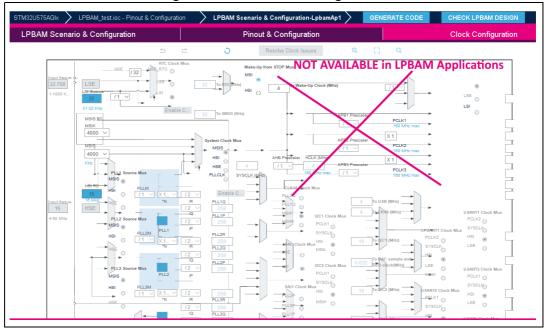


Figure 591. Clock tree configuration

18.3.3 SoC & IPs: configuring the IPs

Only IPs of the Smart run domain are available in the LPBAM context.

In the LPBAM context, most IPs show the same configuration possibilities as the main project. However, for some IPs, some additional configuration is needed. For example, when an IP internal interrupt can be used in the LPBAM context, a dedicated configuration Tab is shown.



Pinout & Configuration	LPBAM Scenario & Configuration
Connecting Ac2 System Core ✓ CORTEX_M0+ ✓ DMA GRO NOO NVC NOO NVC NOO NVC NOO NVC NOO NVC RCC ✓ YS ✓ Main project context Connecting ✓ IC1 * IC1 * IC21 * IC31 * Connecting × IC31 * Camputing * CRC *	Q ✓ ● System Core ✓ O ● G ● G ● G ● G ● G ● G ● G ● G ● G ● G ● Commetive ✓ Connective ✓ Connective ✓ Connective ✓ Connective ✓ ADF1 ● ADF1 ● Power and Thermal × ● PWR ●

Figure 592. Available IPs

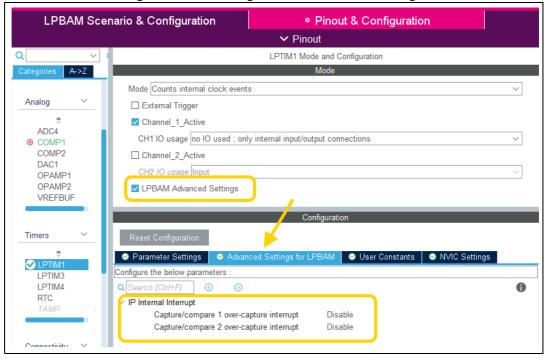


Figure 593. IP configuration: advanced settings

All IPs used at runtime by the LPBAM must be configured in the Pinout & Configuration view. Their configuration must be coherent with the LPBAM scenario.



Clicking "Check LPBAM Design" on the upper right corner of the user interface returns, for each IP used but not configured in an LPBAM application, a warning in the output window.

Warning: "Check LPBAM Design" checks only that the IPs are configured in the "Pinout & Configuration", it does not check whether the HAL configuration is coherent with the LPBAM APIs used in the scenario.

18.3.4 SoC & IPs: configuring low power settings

Starting with STM32CubeMX6.5, users can configure low power settings for their project. These settings (to be found under the PWR IP) are very important to minimize the power consumption of an LPBAM application.

LPBA	M Scei		Pinout & Configuration
		✓ Pin	out
Q ~	0	PWR Mode and Configuration	
Categories A->Z		Mode	
System Core	>	 WakeUp from Standby configuration Debug Pins 	
Analog	>	> Monitoring	
Timers	>	✓ Low Power ☑ Dead Battery Signals disabled	
Connectivity	>	Power saving mode	
Power and Thermal	~	Security/Privilege attributes	

Figure 594. LPBAM low power settings

18.3.5 LPBAM scenario: managing queues

An LPBAM scenario consists of one or more queues, each with one or more nodes. The center panel describes the scenario of the LPBAM application: click the queue name to display its diagram in the center panel and its configuration in the right panel. The name of the selected queue is underlined in blue.

To add more queues, click the "+" button in that panel, or click "Add queues" from the LPBAM management section in the left panel:

- The maximum number of queues is four on STM32U5 series, limited by the number of LPDMA1 channels.
- Adding an LPBAM application to the project automatically creates one empty queue for that application.

Warning: For LPBAM applications with multiple queues, STM32CubeMX does not manage the runtime synchronization between queues. It is the user's



responsibility when assembling its final application to "start" the different queues at runtime.

The "LPBAM Management" section allows to remove and rename queues:

- To delete a queue, right-click the queue name and select "Delete".
- To rename a queue, right-click the queue name and select "Rename".
- To switch between queues in an LPBAM application, click the queue name: the middle and right panels are refreshed to display the selected queue and its configuration.

18.3.6 Queue description: managing nodes

A queue description consists of a sequence of functional nodes on a timeline: the sequence is displayed as a diagram in the central panel and the queue configuration in the right panel.

To add nodes to a queue:

- Click the name of the queue to be updated.
- Use the "LPBAM function Toolbox", in the left panel to browse the list of IPs and functions (LPBAM firmware APIs) that can be used to create nodes.
- Click the IP name to expand and see the list of available functions.
- Click the "+" sign next to the function name to add the function as a node in the queue: the queue diagram in the center panel is updated accordingly.
- Example: on Queue1 of LpbamAp1, COMP1 is started, then data transfer on COMP1 Output is performed (see *Figure 595*).

To remove nodes from the diagram, click the cross on the node right-end-upper corner.

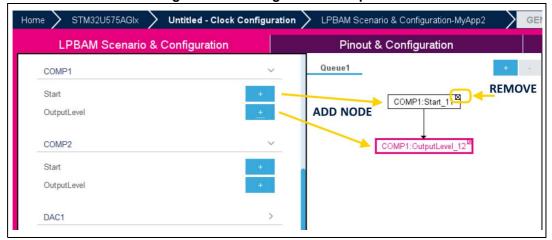


Figure 595. Adding nodes to a queue



18.3.7 Queue description: configuring the queue in circular mode

STM32CubeMX offers the possibility to design circular queues:

- Select the queue to be configured by clicking the queue name in the center panel: the queue configuration is displayed in the right panel.
- Click the Circular mode checkbox to configure the queue in circular mode: by default, the queue loops back to the first node (see *Figure 596*).
- To loop back to a different node, click the end of the arrow and drag it to the node of choice.
- To remove the loop, uncheck Circular mode.

· LPBAM Scenario & Configuration-MyApp2 🛛 🔪	GENERATE CODE	CHECK LF	BAM DESIGN	
Pinout & Configuration	Clo	ock Configu	ration	
Queue1 Queue2 +	MyA	pp2/Scenar	rio <mark>l</mark> Queue2	
	Queue Circularity	Configuration		~
Conf ADC4:Conversion Config_13 [™]	Circular Mode	9		
	DMA Channel Co	onfiguration		~
Data ADC4:Conversion_data_14	Priority	Low		~
Conf ADC4:Analog_W;tchdog_15 [™]	DMA Channel Int	errupt Configurat	lion	~
	Data Transfer Err	or Interrupt	Disable	~
	Update Link Erro	r Interrupt	Disable	\sim

Figure 596. Queue in circular mode

Some functions first configure the IP, then manage the data transfer. In case of circular mode, the loop can be plugged on the configuration ("Conf") or on the data part ("Data") of the function.

An example is provided in *Figure 597*: when the queue is executed, the two first nodes and the configuration of the third node are executed once. whereas the data transfer is repeated as part of the loop.



on 🔰 LPBAM Scenario & Configuration-MyAp	p2 >	GENERATE CODE	CHECK LPBAM					
Pinout & Configuration		Clock Cor	figuration					
Queue1 Queue2 Queue3 +	. [MyApp2/Sc	enario/Queue3					
		Queue Circularity Configura	ition					
Conf ADC4:Conversion_Config_16		Circular Mode						
Conf ADC4:Analog Watchdog 17		DMA Channel Configuration	1					
		Priority	_OW					
Conf ADC4:Conversion_data_18		DMA Channel Interrupt Con Data Transfer Error Interrup	t Disable					
ADC4:Conversion data 18								

Figure 597. Queue looping back on IP data transfer

18.3.8 Queue description: configuring the DMA channel hosting the queue

The execution of an LPBAM queue consists of LPDMA chained transfers. The DMA hosting the queue execution must be configured as needed by the application (see *Figure 598*).

MyApp2/Scenario	Queue3		LpbamAp	2/Scenario/Queue3
			Queue Circularity Configuration	~
DMA Channel Configuration		~	Circular Mode	
Priority		~	DMA Channel Configuration	~
DMA Channel Interrupt Configuration	I	~	Priority	Low
Data Transfer Error Interrupt	Disable	$\overline{}$	DMA Channel Interrupt Configurati	ion 🗸
Update Link Error Interrupt	Disable	~	Data Transfer Error Interrupt	Enable ~
User Setting Error Interrupt	Disable	$\overline{}$	Update Link Error Interrupt	Disable ~
Transfer Complete Interrupt	Disable	~	User Setting Error Interrupt	Disable ~
Trigger Overrun Interrupt	Disable	~	Transfer Complete Interrupt Trigger Overrun Interrupt	Disable ~
DMA Channel NVIC Configuration		~	DMA Channel NVIC Configuration	~
			Preemption Priority	0 ~
DMA Channel Attributes Configuration	on	~	Sub Priority	0 ~
Enable Channel as Privileged	Ion-Privileged	~	DMA Channel Attributes Configura	ation ~
			Enable Channel as Privileged	Non-Privileged ✓

Figure 598. LPBAM queue: DMA configuration

Basic configuration

Select the queue to be configured by clicking the queue name on the center panel, the configuration of the DMA channel hosting that queue is shown in the right panel.



Note that some settings usually available for configuring a DMA channel are not provided in the user interface, as they are directly managed either by STM32CubeMX or by the LPBAM driver.

DMA channel NVIC configuration

NVIC settings are available only if one DMA channel interrupt is enabled (see right panel in *Figure 598*). The preemption priority and sub priority ranges in the LPBAM context depend on the NVIC priority group set for the whole project (the main project with the LPBAM applications).

Warning: Always check preemption and sub-priorities in the LPBAM context after changing the NVIC priority group from the main project Pinout & Configuration view.

18.3.9 Node description: accessing contextual help and documentation

STM32CubeMX provides contextual help and link to reference documentation on LPBAM functions to guide the user during the function selection process:

- From the "LPBAM function Toolbox" in the left panel, hover the mouse on an IP name to show the contextual help with links to reference documentation (see *Figure 599*).
- It is recommended to read carefully the LPBAM global documentation and the IP "Description, Usage and Constraint" to learn how to assemble nodes in a queue, several queues, what can be done and what cannot be done. Some restrictions apply and are due to the LPBAM mechanism. They are not coming from the IP itself or from HAL constraints.

L	PBAM Scenario & Configuration	Pinout & Co						
		Queue1 Queue						
LPBAM f	unction Toolbox 🗸							
ADC4	ADC LPBAM Utility:							
	Analog to digital converter 16-bits							
Conv	Summary:	-						
Conv	This driver provides the following list of features :							
	(+) Configure the ADC peripheral for conversion.							
Conv	Conv (+) Starts the ADC conversion.							
Analo	(+) Configure and starts the ADC conversion.							
7 4164	Anale (+) Configure and starts only the watchdog input signal monitoring without transfer to SRAM.							
	Related documentation:							
COM	COM - <u>LPBAM Utility Getting Started</u> - <u>LPBAM Utility for ADC : Description, Usage and Constraints</u>							
	- Draw ounty for ADO : Description, osage and constraints							
COM	P2 >							

Figure 599. LPBAM functions contextual help



18.3.10 Node description: configuring node parameters

Once a function is chosen from the "LPBAM Function Toolbox" and added to a queue, it can be configured. In the center panel, click on a node to select it: the function is highlighted in pink, and its configuration is shown in the right panel (see *Figure 600*).

The example shows the "Start" parameters of the LPBAM COMP1_Start function. The HAL driver uses the same parameter names to configure a COMP IP. As mentioned before, the LPBAM firmware is not a HAL driver. However, the IP being unique, the LPBAM driver has been designed so that the IP parameters use, whenever possible, the same naming as found in the HAL driver.

Pinou	Pinout & Configuration			Clo	ock Configuration
Queue1	Queue2	Queue3	+ -	MyApp2/Scer	nario/Queue1/COMP1:Start_11
				Enter the Function	Name Start_11
	COMP	1:Start_11 [™]		Start	
				Input Plus	PC5
	COMP1:0	↓ utputLevel_12		Input Minus	1/4 Internal Vref
				Output Polarity	COMP Output on GPIO is not inverted
				Trigger Configura	tion
				The Function exe	ecution is not conditionned by a Trigger

Figure 600. LPBAM queue node configuration

Warning: LPBAM IP functions access IP hardware resources, to be properly configured in the "Pinout & Configuration" view.

When a parameter is set to a hardware resource such as a GPIO, the resource must be configured in the Pinout & Configuration view.

In the example shown in *Figure 600*, the COMP "Input Plus" is set to PC5. If PC5 is not configured in the "Pinout & configuration" view, the generated LPBAM application can gets a "null signal" on Input Plus, and will be not functional.

To fix this issue:

- Go to the Pinout&Configuration view
- Search PC5 using the search field
- Right-click the PC5 pin and select COMP_Inp (see *Figure 601*)



Pino	Pinout & Configuration			Clo	ock Configuration
Queue1	Queue2	Queue3	+ -	MyApp2/Scer	nario/Queue1/COMP1:Start_11
				Enter the Function	Name Start_11
	COMP	:Start_11 [™]		Start	
				Input Plus	PC5
	COMP1:0	↓ utputLevel_12 [™]		Input Minus	1/4 Internal Vref
				Output Polarity	COMP Output on GPIO is not inverted
				Trigger Configura	tion
				The Function exe	ecution is not conditionned by a Trigger

Figure 601. LPBAM node: configuring hardware resources

Another example can be made using a timer to generate a PWM signal. The HAL driver requires a timer channel to be configured as output. Same applies when using the LPBAM firmware.

Note: All constraints concerning the initial configuration of the IP are mentioned in the LPBAM firmware documentation. Use STM32CubeMX "LPBAM Design check" mechanism (see dedicated section) to detect missing configurations.

18.3.11 Node description: configuring a trigger

For all IPs and functions, with the LPBAM firmware it is possible to use a hardware signal to trigger a node. STM32CubeMX allows to configure such trigger from the node configuration panel. By default, the node execution is not triggered. When trigger is enabled, all possible trigger signals are listed.

Warning: It is the user responsibility to properly configure the triggers. STM32CubeMX does not check for configuration errors.

Taking the COMP function "Start" as an example (see *Figure 602*), choose the function execution to be triggered on the rising edge of hardware signal, for the example, then, select the hardware signal among the list of hardware signals proposed.



3		
Queue1 Queue2 + -	LpbamAp1/Scenario/Queue1/COMP1:Start_11	
	Enter the Function Name Start_11	
COMP1:Start_11 [™]	Start	~
	Input Plus PC5	~
	Input Minus 1/4 Internal Vref	~
COMP1:OutputLevel_12	input winds	
	Output Polarity COMP Output on GPIO is not inverted	\sim
	Trigger Configuration Trigged The Function execution is trigged on the Rising Edge of the Hardware Sig Trigger Hardware Signal is EXTL Line 0 EXTL Line 0 EXTL Line 1 EXTL Line 1 Select signal EXTL Line 2 EXTL Line 2 EXTL Line 3 EXTL Line 3 EXTL Line 4 TAMP TRG1 TAMP TRG3 TAMP TRG3	~

Figure 602. LPBAM node trigger configuration

If a node is a function managing LPTIM1_CH1, it is possible to select LPTIM1_CH1 as the trigger (see *Figure 603*).

Figure 603. LPBAM node triggered using timer channel

Pinout & Configuration	Clock Configuration
Queue1 + -	LpbamAp1/Scenario/Queue1/COMP1:Start_11
	Enter the Function Name Start_11
COMP1:Start_11	Start ~
	Input Plus PC5 ~
COMP1:OutputLevel_12 [™]	Input Minus 1/4 Internal Vref 🗸
	Output Polarity COMP Output on GPIO is not inverted ~
	Trigger Configuration V
	The Function execution is $$$$ trigged on the Rising Edge of the Hardware Signal \checkmark
	Trigger Hardware Signal is EXTI Line 0 🗸
	TAMP TRG2
	TAMP TRG3
	LPTIM1 CH1 LPTIM1 CH2
	LPTIM3 CH1
	LPTIM4 OUT
	COMP1 OUT
	COMP2 OUT

18.3.12 Node description: reconfiguring a DMA for data transfer

Nodes set to a function managing data transfers (all functions with associated data transfer and with a name not ending with _Config), come with a specific configuration section: "Reconfigure DMA for Data Transfer" (see *Figure 604*).

Each DMA data transfer is based on a specific configuration, including, among others, data size, buffer address, address increment. The DMA default settings are functional.



galo 00 li 21 27 ali 100	ier reconniganing a blint
Pinout & Configuration	Clock Configuration
Queue1 + -	LpbamAp1/Scenario/Queue1/COMP1:OutputLevel_13
	Enter the Function Name OutputLevel_13
COMP1:OutputLevel_13 ¹²	Data Buffer Name DataBufferName
	Data Buffer Offset DataBufferOffset
	Number of Data myDataSize
ADC4:Conversion_Config_15	
	Trigger Configuration
	The Function execution is not conditionned by a Trigger
×	
	Reconfigure DMA for Data Transfer / Default Setting is functional
	Source Address Increment After Transfer Disabled
	Destination Address Increment After Transfer Enabled
	Src Data Width
	Dest Data Width Word
	Transfer Event Generation TC and HT generated on the last lin
I	

Figure 604. LPBAM node: reconfiguring a DMA

DMA settings can be changed, but they depend upon the IP and the function.

For example, for "COMP Output Level":

- Data transferred are output data and are transferred from the register IP to the memory. The "Source Address" referring to the IP data register is not incremented: STM32CubeMX user interface shows that the "Source address increment after transfer" parameter cannot be enabled.
- Data transferred to memory can be saved at the same memory address, or in a Table: in this case, the "Destination Address increment after transfer" can be disabled or left enabled (see *Figure 604*).

Figure 605.	Reconfiguring	DMA for dat	a transfer when	destination i	s memorv
i iguio oooi	roooninganing	Bill/ Clor dat		acountation	••

Source Address Increment After Transfer	Disabled	
Destination Address Increment After Transfer	Enabled	
Src Data Width	Disabled Enabled	
Dest Data Width	Word	
Transfer Event Generation	TC and HT generated on the las	t lir



18.4 Checking the LPBAM design

STM32CubeMX offers users with the possibility to check their LPBAM design for coherency and completeness, by detecting:

- Incoherences between the IP LPBAM function selected for a node and the corresponding IP configuration.
- Wrong queue designs (the sequence of nodes is invalid).

Click CHECK LPBAM DESIGN to check all LPBAM applications currently available in the project. Results appear in the LPBAM output log window (see *Figure 606*).

Note: Messages raised on the LPBAM design do not prevent users to generate the C code for their project. Supported type of messages are ERROR (in red), Warning (in orange), and Information (in blue).

KTM32CubeMX Untitled*: STM32U575AGixQ			×
STM32 File Window	Help Click to chec	° ^K 🍳 🗗 🗖 🎽 🔿 🔆 /	57
Home 🔪 STM32U575AGIxQ 🔪 Untitled - Project Manager 刘	LPBAM Scenario & Configuration-MyApp2	GENERATE COLLECK LPBAM DESIGN	
LPBAM Scenario & Configuration	Pinout & Configuration	Clock Configuration	
LPBAM Management ~	Queue1 < > +	MyApp2/Scenario/Queue1	
✓ III LPBAM Manager		Queue Circularity Configuration	~
≻ LpbamAp1 ∽ ⊙ MyApp2 <mark>-</mark> ⊙ Scenario	12C3:Master_Receive_Config_3	Circular Mode	
Oueue1 Add Queue	↓ I2C3:Master Receive 7 [™]	DMA Channel Configuration	<u>~</u>
Add Application	1200.Waater_Kecewe_r	Priority	~
MyAr 2: Settings Y		DMA Channel Interrupt Configuration	<u>~</u>
Cortex-M33 secure Cortex-M33 non secure		Data Transfer Error Interrupt Disable	~
		Update Link Error Interrupt Disable	v
✓ LPBA I Scenarios Settings ✓ LPBAM Scenario uses resources from Smart Run Domain		User Setting Error Interrupt Disable	$\overline{}$
LPBAM Scenario is hosted by LPDMA1	Results	Transfer Complete Interrupt Disable	$\overline{}$
MCUS Se ston Outpu LEBAM Cutput Log LEBAM Nesign Check log LEBAM Selection Scenario Scenario Publication Scenario Scenario Multiputation i Configuration of Queuel is the Astiguration when focusing on the Queue in the centra MyApp2 opplication Scenario Scenario Peripheral Configuration: 12C3 is used through Queuel Peripheral Configuration: 12C3 is used through Queuel Application Scenario Scenario Peripheral Configuration: 11 Trigger Hardware Signal panel © Scenario Design: function 12C3:HASTERRECEIVE_CONFIG is reconfiguration is necessary, it's recommended to use M © Queue Configuration: Configuration of Queuel is the Configuration" when focusing on the Queue in the centra	<pre>1 panel /'I2C3:Master_Receive_7' and is not configur /'I2C3:Master_Receive_Config_3' and is not co .s used in this scenario should be configure ASTERRECEIVE instead of MASTERRECEIVE_CONFIG default configuration. Queue can be configur</pre>	red. IP must be configured in this Application configured. IP must be configured in this 1 in this application 'Finout & Configuration' Config_3'. For memory optimization purpose, who and MASTERRECEIVE_DATA	en a

Figure 606. Design check



18.5 Generating a project with LPBAM applications

Click Generate Code from the main project view. As exemplified in *Figure 606*, the resulting project shows, in addition to the main project files and folders, the stm32_lpbam_conf.h file, a dedicated folder for the configuration code, and the utilities folder with the LPBAM utility firmware.

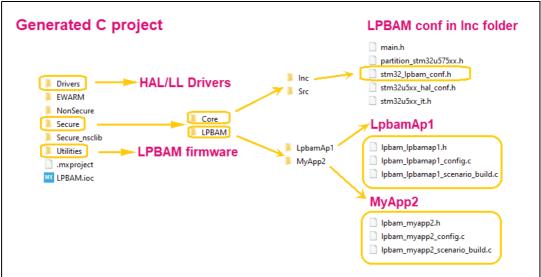


Figure 607. STM32CubeMX project generated with LPBAM applications

STM32CubeMX generates:

- In the Core/Inc folder, the stm32_lpbam_conf.h file that defines all the LPBAM modules enabled for the LPBAM applications, to be used by the LPBAM utility firmware.
- In the LPBAM folder, the code for the LPBAM applications and their scenarios. The lpbam_<application name>.h file provides the prototypes of the functions to call in the main project to initialize the application, build and initialize the scenario, link it with the DMA, start it, stop it, unlink it, and de-initialize it.

As an example, for the LpbamAp1 application, STM32CubeMX generates the following functions:

```
/* LpbamAp1 application initialization */
void MX_LpbamAp1_Init(void);
/* LpbamAp1 application - scenario initialization */
void MX_LpbamAp1_Scenario_Init(void);
/* LpbamAp1 application - scenario build */
void MX_LpbamAp1_Scenario_Build(void);
/* LpbamAp1 application - scenario link */
void MX_LpbamAp1_Scenario_Link(DMA_HandleTypeDef *hdma);
/* LpbamAp1 application - scenario start */
void MX_LpbamAp1_Scenario_Start(DMA_HandleTypeDef *hdma);
```



```
/* LpbamAp1 application - scenario stop */
void MX_LpbamAp1_Scenario_Stop(DMA_HandleTypeDef *hdma);
/* LpbamAp1 application - scenario unlink */
void MX_LpbamAp1_Scenario_UnLink(DMA_HandleTypeDef *hdma);
/* LpbamAp1 application - scenario de-initialization */
void MX_LpbamAp1_Scenario_DeInit(void);
```

18.6 LPBAM application for TrustZone[®] activated projects

Starting with STM32CubeMX 6.6.0, users can create LPBAM applications for projects with TrustZone $^{\mbox{\scriptsize I\!R}}$ activated.

- 1. Access to MCU selector and select an STM32U575/585 device
- 2. Click Create a new project
- 3. Choose the option "with TrustZone activated"

STM32CubeMX standard project view

STM32CubeMX standard project view proposes security settings for peripherals (*Figure 608*) and the clock tree (*Figure 609*).

STM32CubeMX LPBAM view

In STM32CubeMX LPBAM Application configuration context, the peripherals and the clock tree do not come with dedicated security settings (see *Figure 610* and *Figure 611*). The choice of context, secure or nonsecure, is done at LPBAM application level (*Figure 612*).

Security settings coherency check

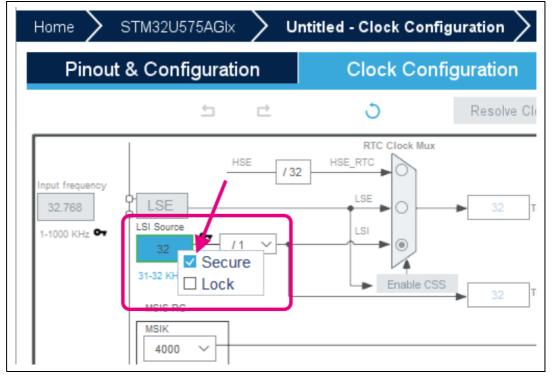
- 1. Click CHECK LPBAM DESIGN
- 2. Enable Show Attribute Warning Messages to see details about LPBAM security related configuration issues (see *Figure 613*)



g			et i empheral cooure context acorgi	
Home 🔰 STM	32U575AG	x > u	ntitled - Pinout & Configuration >	LPBAM Sc
Pinout & C	onfigura	ation	Clock Configuration	Р
		~	Software Packs 🗸 🗸	' Pinout
Q	\sim	0	COMP1 Mode and Configurat	tion
Categories A->Z			Mode	
\$	M33S	M33NS	Runtime contexts:	
ADC1	0	0	Cortex-M33 secure Cortex-M33	non secure
ADC4	0	0	 C)
ADF1	0	0	Input [+] Disable	\sim
COMP1	0	0	Input [-] Disable	~
COMP2	۲	0		
CORDIC	0	0	ExternalOutput	
CORTEX_M33		۲		
CORTEX_M33_S	۲			
CRC	\cap	Ο		

Figure 608. STM32CubeMX project - Peripheral secure context assignment







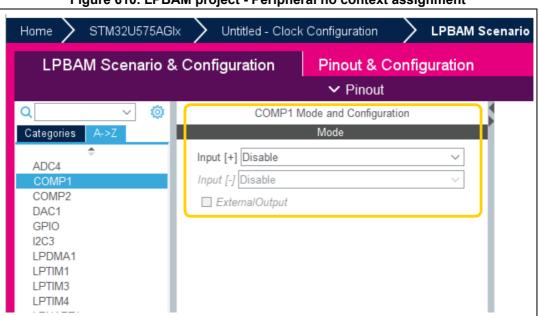
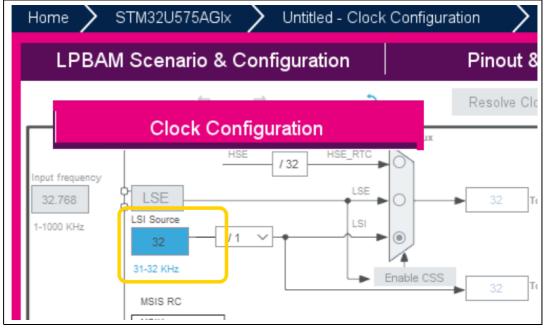


Figure 610. LPBAM project - Peripheral no context assignment

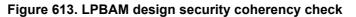


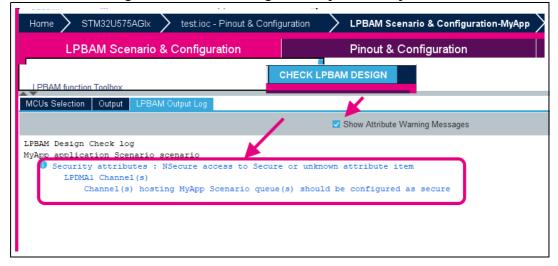




Home 🔪 ST	rm32U575AGIx	> Untit	ed - Clock C	onfigura	tion		Scenario &
LPBAM	Scenario & C	Configura	tion		Pinou	ut & Configu	ıration
LPBAM Manag	gement				~	Queue1	
	I Manager App Scenario ⊘ Queue1 ♣ Add Queue Application						
MyApp: Settin	igs				~		
Runtime o							
Cort	ex-M33 secure	Cort	ex-M33 non s	ecure	_		
	۲		0				
✓ LPBAM So	cenarios Settings						
🔽 LPE	BAM Scenario use	s resources	from Smart R	un Domai	n		
✓ LPE	BAM Scenario is h	osted by LP	DMA1				

Figure 612. LPBAM application - Secure context assignment







The following pin assignment rules are implemented in STM32CubeMX:

- Rule 1: Block consistency
- Rule 2: Block inter-dependency
- Rule 3: One block = one peripheral mode
- Rule 4: Block remapping (only for STM32F10x)
- Rule 5: Function remapping
- Rule 6: Block shifting (only for STM32F10x)
- Rule 7: Setting or clearing a peripheral mode
- Rule 8: Mapping a function individually (if Keep Current Placement is unchecked)
- Rule 9: GPIO signals mapping

A.1 Block consistency

When setting a pin signal (provided there is no ambiguity about the corresponding peripheral mode), all the pins/signals required for this mode are mapped and pins are shown in green (otherwise the configured pin is shown in orange).

When clearing a pin signal, all the pins/signals required for this mode are unmapped simultaneously and the pins turn back to gray.

Example of block mapping with an STM32F107x MCU

If the user assigns I2C1_SMBA function to PB5, then STM32CubeMX configures pins and modes as follows:

- I2C1_SCL and I2C1_SDA signals are mapped to the PB6 and PB7 pins, respectively (see *Figure 614*).
- I2C1 peripheral mode is set to SMBus-Alert mode.



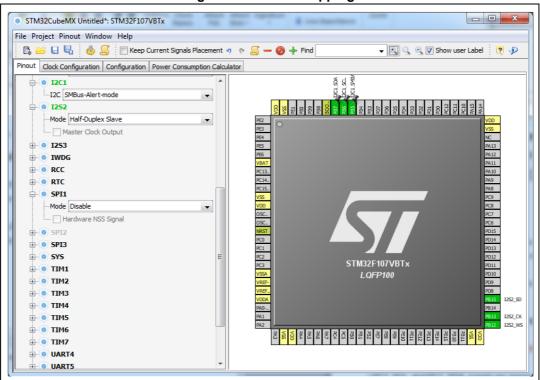


Figure 614. Block mapping

Example of block remapping with an STM32F107x MCU

If the user assigns GPIO_Output to PB6, STM32CubeMX automatically disables I2C1 SMBus-Alert peripheral mode from the peripheral tree view and updates the other I2C1 pins (PB5 and PB7) as follows:

- If they are unpinned, the pin configuration is reset (pin grayed out).
- If they are pinned, the peripheral signal assigned to the pins is kept and the pins are highlighted in orange since they no longer match a peripheral mode (see *Figure 615*).



FI	gure 615. Block remapping
STM32CubeMX Untitled*: STM32F107VBTx	
File Project Pinout Window Help	
🕞 📴 🛃 🍓 💁 📄 Keep Current Signals Placement 🤊 🕫	🗸 🗕 🖉 🔶 Find 🔍 🔍 🔍 🖓 Show user Label 🤅 🦻 🥠
Pinout Clock Configuration Configuration Power Consumption Calculator	
Clock Configuration Configuration Power Consumption Calculator	
• • • CAN1	got_SDA @pro_oubut
Master Mode	
e • • CAN2	
Slave Mode	VDD VDD VDD VDD VDD VDD VDD VDD VDD VDD
⊕ • CRC	000
⊕ • • ETH	PE2 O VSS
Mode Disable	PE4 NC
B € 12C1	PE5 PA13 PA12 PA12
Disable 🗸	PE6 PA12 PA11
1252	PC13
Mode Disable	PC14
Master Clock Output	PC15 PA8
⊕ • • 1253	VSS PC9 PC8
De Structure E	OSC
⊕ • © RCC	OSC
	NRST PD15
Mode Disable	PC1 PD14
Hardware NSS Signal	PC2 PD12
⊕- @ SPI2	PC3 STM32F107VBTx PD11
⊕-	VSSA LQFP100 PD10
⊕-	VRE. PD8
⊕- ● TIM1	VDDA PB15
• • TIM2	PA0
⊕- © TIM3 ⊕- © TIM4	PA1 PB13 PB13 PB12
TIM6	WDD VSS PB11 PB12 PB13 PB14 PB15 PB16 PE17 PE18 PE19 PE10 PE11 PE11 PE11 PE12 PE13 PE14 PE15 PE17 PE18 PE19 PE19 PE19 PE19 PE19 PE20 PE21 PE32 PE41 PE33
• • TIM7 •	

Figure 615. Block remapping

For STM32CubeMX to find an alternative solution for the I2C peripheral mode, the user will need to unpin I2C1 pins and select the I2C1 mode from the peripheral tree view (see *Figure 616* and *Figure 617*).



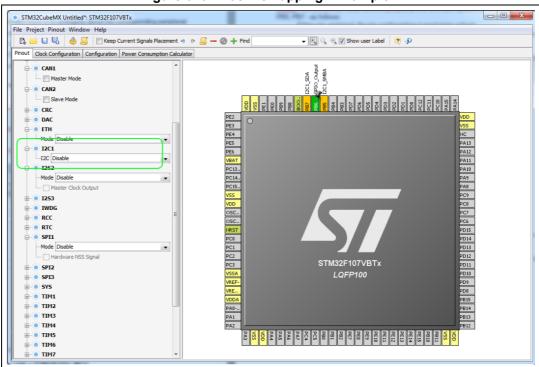
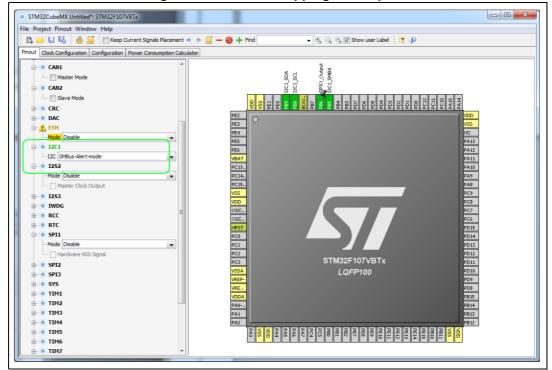


Figure 616. Block remapping - Example 1

Figure 617. Block remapping - Example 2





A.2 Block inter-dependency

On the **Pinout** view, the same signal can appear as an alternate function for multiple pins. However it can be mapped only once.

As a consequence, for STM32F1 MCUs, two blocks of pins cannot be selected simultaneously for the same peripheral mode: when a block/signal from a block is selected, the alternate blocks are cleared.

Example of block remapping of SPI in full-duplex master mode with an STM32F107x MCU

If SPI1 full-duplex master mode is selected from the tree view, by default the corresponding SPI signals are assigned to PB3, PB4 and PB5 pins (see *Figure 618*).

If the user assigns to PA6 the SPI1_MISO function currently assigned to PB4, STM32CubeMX clears the PB4 pin from the SPI1_MISO function, as well as all the other pins configured for this block, and moves the corresponding SPI1 functions to the relevant pins in the same block as the PB4 pin (see *Figure 619*).

(by pressing CTRL and clicking PB4 to show PA6 alternate function in blue, then drag and drop the signal to pin PA6)

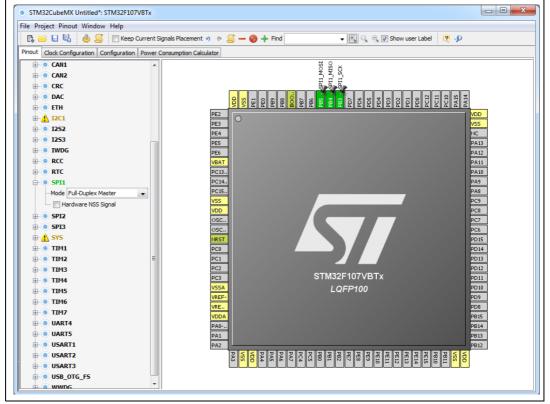


Figure 618. Block inter-dependency - SPI signals assigned to PB3/4/5

UM1718



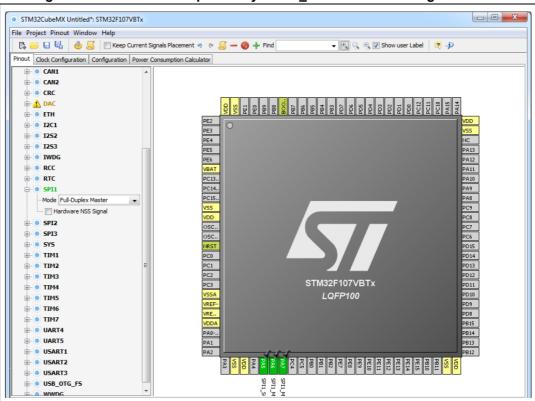


Figure 619. Block inter-dependency - SPI1_MISO function assigned to PA6



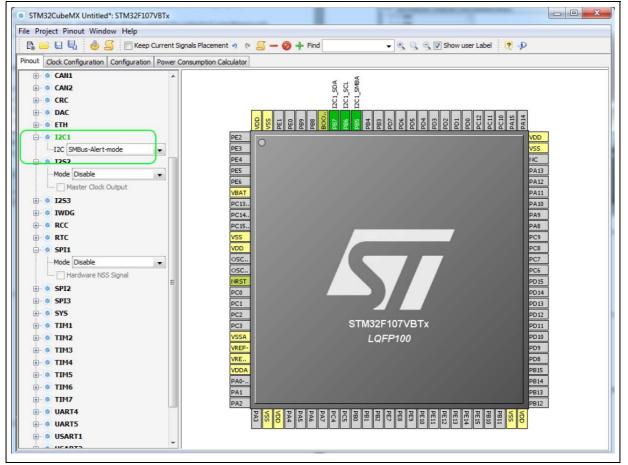
A.3 One block = one peripheral mode

When a block of pins is fully configured in the **Pinout** view (shown in green), the related peripheral mode is automatically set in the Peripherals tree.

Example of STM32F107x MCU

Assigning the I2C1_SMBA function to PB5 automatically configures I2C1 peripheral in SMBus-Alert mode (see Peripheral tree in *Figure 620*).

Figure 620. One block = one peripheral mode - I2C1_SMBA function assigned to PB5



A.4 Block remapping (STM32F10x only)

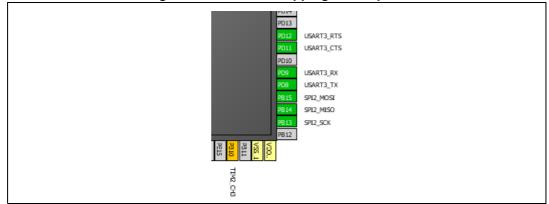
To configure a peripheral mode, STM32CubeMX selects a block of pins and assigns each mode signal to a pin in this block. In doing so, it looks for the first free block to which the mode can be mapped.

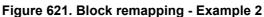
When setting a peripheral mode, if at least one pin in the default block is already used, STM32CubeMX tries to find an alternate block. If none can be found, it either selects the functions in a different sequence, or unchecks Keep Current Signals Placement, and remaps all the blocks to find a solution.



Example

STM32CubeMX remaps USART3 hardware-flow-control mode to the (PD8-PD9-PD11-PD12) block, because PB14 of USART3 default block is already allocated to the SPI2_MISO function (see *Figure 621*).





A.5 Function remapping

To configure a peripheral mode, STM32CubeMX assigns each signal of the mode to a pin. In doing so, it will look for the first free pin the signal can be mapped to.

Example using STM32F415x

When configuring USART3 for the Synchronous mode, STM32CubeMX discovered that the default PB10 pin for USART3_TX signal was already used by SPI. It thus remapped it to PD8 (see *Figure 622*).

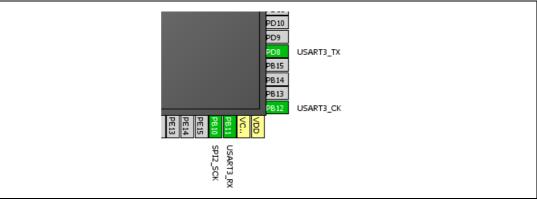


Figure 622. Function remapping example



A.6 Block shifting (only for STM32F10x and when "Keep Current Signals placement" is unchecked)

If a block cannot be mapped and there are no free alternate solutions, STM32CubeMX tries to free the pins by remapping all the peripheral modes impacted by the shared pin.

Example

With the Keep current signal placement enabled, if USART3 synchronous mode is set first, the Asynchronous default block (PB10-PB11) is mapped and Ethernet becomes unavailable (shown in red) (see *Figure 623*).

Unchecking Keep Current Signals Placement allows STM32CubeMX shifting blocks around and freeing a block for the Ethernet MII mode. (see *Figure 624*).

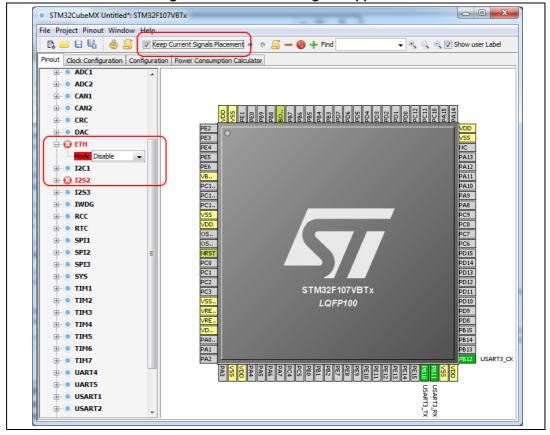


Figure 623. Block shifting not applied



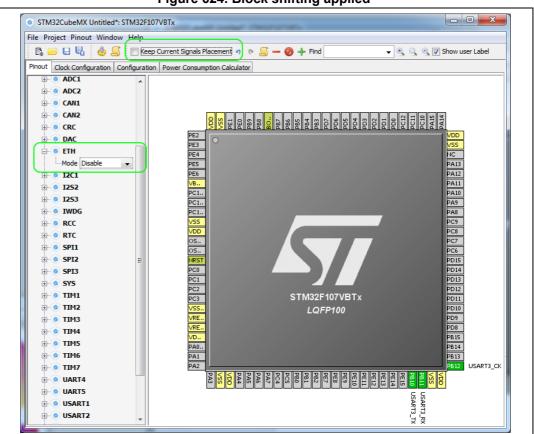


Figure 624. Block shifting applied

A.7 Setting and clearing a peripheral mode

The Peripherals panel and the **Pinout** view are linked: when a peripheral mode is set or cleared, the corresponding pin functions are set or cleared.

A.8 Mapping a function individually

When STM32CubeMX needs a pin that has already been assigned manually to a function (no peripheral mode set), it can move this function to another pin, only if Keep Current Signals Placement is unchecked and the function is not pinned (no pin icon).

A.9 GPIO signals mapping

I/O signals (GPIO_Input, GPIO_Output, GPIO_Analog) can be assigned to pins either manually through the **Pinout** view or automatically through the **Pinout** menu. Such pins can no longer be assigned automatically to another signal: STM32CubeMX signal automatic placement does not take into account this pin anymore since it does not shift I/O signals to other pins.

The pin can still be manually assigned to another signal or to a reset state.



UM1718 Rev 47

UM1718

STM32CubeMX C code generation design Appendix B choices and limitations

B.1 STM32CubeMX generated C code and user sections

The C code generated by STM32CubeMX provides user sections as illustrated below. They allow user C code to be inserted and preserved at next C code generation.

User sections shall neither be moved nor renamed. Only the user sections defined by STM32CubeMX are preserved. User created sections will be ignored and lost at next C code generation.

```
/* USER CODE BEGIN 0 */
(..)
/* USER CODE END 0 */
```

Note:

STM32CubeMX may generate C code in some user sections. It will be up to the user to clean the parts that may become obsolete in this section. For example, the while(1) loop in the main function is placed inside a user section as illustrated below:

```
/* Infinite loop */
  /* USER CODE BEGIN WHILE */
 while (1)
  {
  /* USER CODE END WHILE */
  /* USER CODE BEGIN 3 */
  3
/* USER CODE END 3 */
```

B.2 STM32CubeMX design choices for peripheral initialization

STM32CubeMX generates peripheral _Init functions that can be easily identified thanks to the MX prefix:

```
static void MX_GPIO_Init(void);
static void MX_<Peripheral Instance Name>_Init(void);
static void MX_I2S2_Init(void);
```

An MX <peripheral instance name> Init function exists for each peripheral instance selected by the user (e.g, MX_I2S2_Init). It performs the initialization of the relevant handle structure (e.g, &hi2s2 for I2S second instance) that is required for HAL driver initialization (e.g., HAL_I2S_Init) and the actual call to this function:

```
void MX I2S2 Init(void)
{
hi2s2.Instance = SPI2;
 hi2s2.Init.Mode = I2S_MODE_MASTER_TX;
 hi2s2.Init.Standard = I2S_STANDARD_PHILLIPS;
 hi2s2.Init.DataFormat = I2S_DATAFORMAT_16B;
 hi2s2.Init.MCLKOutput = I2S_MCLKOUTPUT_DISABLE;
```

```
hi2s2.Init.AudioFreq = I2S_AUDIOFREQ_192K;
hi2s2.Init.CPOL = I2S_CPOL_LOW;
hi2s2.Init.ClockSource = I2S_CLOCK_PLL;
hi2s2.Init.FullDuplexMode = I2S_FULLDUPLEXMODE_ENABLE;
HAL_I2S_Init(&hi2s2);
```

}

By default, the peripheral initialization is done in *main.c.* If the peripheral is used by a middleware mode, the peripheral initialization can be done in the middleware corresponding .c file.

Customized *HAL_<Peripheral Name>_MspInit()* functions are created in the stm32f4xx_hal_msp.c file to configure the low-level hardware (GPIO, CLOCK) for the selected peripherals.

B.3 STM32CubeMX design choices and limitations for middleware initialization

B.3.1 Overview

STM32CubeMX does not support C user code insertion in Middleware stack native files although stacks such as LwIP might require it in some use cases.

STM32CubeMX generates middleware *Init* functions that can be easily identified thanks to the MX_ prefix:

MX_LWIP_Init(); // defined in lwip.h file MX_USB_HOST_Init(); // defined in usb_host.h file MX_FATFS_Init(); // defined in fatfs.h file

Note however the following exceptions:

- No *Init* function is generated for FreeRTOS unless the user chooses, from the Project Settings window, to generate *Init* functions as pairs of .c/.h files. Instead, a *StartDefaultTask* function is defined in the *main.c* file and CMSIS-RTOS native function (*osKernelStart*) is called in the main function.
- If FreeRTOS is enabled, the *Init* functions for the other middlewares in use are called from the *StartDefaultTask* function in the main.c file.
 Example:

void StartDefaultTask(void const * argument)

```
{
   /* init code for FATFS */
   MX_FATFS_Init();
   /* init code for LWIP */
   MX_LWIP_Init();
   /* init code for USB_HOST */
   MX_USB_HOST_Init();
   /* USER CODE BEGIN 5 */
   /* Infinite loop */
   for(;;)
   {
```



```
osDelay(1);
}
/* USER CODE END 5 */
}
```

B.3.2 USB host

USB peripheral initialization is performed within the middleware initialization C code in the *usbh_conf.c* file, while USB stack initialization is done within the *usb_host.c* file.

When using the USB Host middleware, the user is responsible for implementing the USBH_UserProcess callback function in the generated usb_host.c file.

From STM32CubeMX user interface, the user can select to register one class or all classes if the application requires switching dynamically between classes.

B.3.3 USB device

USB peripheral initialization is performed within the middleware initialization C code in the *usbd_conf.c* file, while USB stack initialization is done within the *usb_device.c* file.

USB VID, PID and String standard descriptors are configured via STM32CubeMX user interface and available in the *usbd_desc.c* generated file. Other standard descriptors (configuration, interface) are hard-coded in the same file preventing support of USB composite devices.

When using the USB Device middleware, the user is responsible for implementing the functions in the *usbd_<classname>_if.c* class interface file for all device classes (such as usbd_storage_if.c).

USB MTP and CCID classes are not supported.

B.3.4 FatFs

FatFs is a generic FAT/exFAT file system solution well suited for small embedded systems.

FatFs configuration is available in *ffconf.h* generated file.

The initialization of the SDIO peripheral for the FatFs SD card mode and of the FMC peripheral for the FatFs External SDRAM and External SRAM modes are kept in the *main.c* file.

Some files need to be modified by the user to match user board specificities (BSP in STM32Cube embedded software package can be used as example):

- bsp_driver_sd.c/.h generated files when using FatFs SD card mode
- *bsp_driver_sram.c/.h* generated files when using FatFs External SRAM mode
- *bsp_driver_sdram.c/.h* generated files when using FatFs External SDRAM mode.

Multi-drive FatFs is supported, which means that multiple logical drives can be used by the application (External SDRAM, External SRAM, SD card, USB disk, User defined). However support of multiple instances of a given logical drive is not available (e.g. FatFs using two instances of USB hosts or several RAM disks).

NOR and NAND flash memory are not supported. In this case, the user shall select the FatFs user-defined mode and update the *user_diskio.c* driver file generated to implement the interface between the middleware and the selected peripheral.



B.3.5 FreeRTOS

FreeRTOS is a free real-time embedded operating system well suited for microcontrollers.

FreeRTOS configuration is available in *FreeRTOSConfig.h* generated file.

When FreeRTOS is enabled, all other selected middleware modes (e.g., LwIP, FatFs, USB) will be initialized within the same FreeRTOS thread in the main.c file.

When GENERATE_RUN_TIME_STATS, CHECK_FOR_STACK_OVERFLOW, USE_IDLE_HOOK, USE_TICK_HOOK and USE_MALLOC_FAILED_HOOK parameters are activated, STM32CubeMX generates *freertos.c* file with empty functions that the user shall implement. This is highlighted by the tooltip (see *Figure 625*).

Figure 625. FreeRTOS HOOK functions to be completed by user

Versions		-
CMSIS-RTOS version	1.02	
FreeRTOS version	8.2.3	
Kernel settings		
Hook function related definitions		
USE_IDLE_HOOK	Disabled	
USE_TICK_HOOK	Disabled	
USE_MALLOC_FAILED_HOOK	Disabled	
CHECK_FOR_STACK_OVERFLOW	Disabled	
Run time and task stats gathering relate	definitions	
USE_TRACE_FACILITY	Enabled	
GENERATE_RUN_TIME_STATS	Disabled	
Co-routine related definitions		
USE_CO_ROUTINES	Disabled	
MAX_CO_ROUTINE_PRIORITIES	2	
Software timer definitions		
USE_TIMERS	Enabled	-



B.3.6 LwIP

LwIP is a small independent implementation of the TCP/IP protocol suite: its reduced RAM usage makes it suitable for use in embedded systems with tens of Kbytes of free RAM.

LwIP initialization function is defined in *lwip.c*, while LwIP configuration is available in *lwipopts.h* generated file.

STM32CubeMX supports LwIP over Ethernet only. The Ethernet peripheral initialization is done within the middleware initialization C code.

STM32CubeMX does not support user C code insertion in stack native files. However, some LwIP use cases require modifying stack native files (e.g., *cc.h, mib2.c*): user modifications shall be backed up since they will be lost at next STM32CubeMX generation.

Starting with LwIP release 1.5, STM32CubeMX LwIP supports IPv6 (see Figure 627).

DHCP must be disabled, to configure a static IP address.

Configure the below parameters : Search : Search (CrtI+F)		Checksum	Debug	User Constants
Search : Search (CrtI+F) □ LwIP Version LwIP Version (Version of LwIP sup 1.4.1 □ DHCP Option LWIP_DHCP (DHCP Module) Enabled □ RTOS Settings WITH_RTOS (Use FREERTOS ** Disabled □ Protocols Options LWIP_ICMP (ICMP Module Activati Enabled LWIP_IGMP (IGMP Module) Disabled LWIP_IGMP (IGMP Module) Disabled LWIP_UDP (UDP Module) Enabled	√ General Settings		🖉 Key Options	Perf/Checks
 LwIP Version LwIP Version (Version of LwIP sup 1.4.1 DHCP Option LWIP_DHCP (DHCP Module) Enabled RTOS Settings WITH_RTOS (Use FREERTOS ** Disabled Protocols Options LWIP_ICMP (ICMP Module Activati Enabled LWIP_IGMP (IGMP Module) Disabled LWIP_DNS (DNS Module) Disabled LWIP_UDP (UDP Module) Enabled LWIP_UDP (UDP Module) Disabled LWIP_UDP (UDP Module) Enabled LWIP_UDP (UDP Module) Enabled LWIP_UDP (UDP Module) Enabled LWIP_UDP (UDP Module) Enabled LWIP_UDP (UDP Module) Enabled LWIP_UDP (UDP Module) Enabled LWIP_UDP (UDP Module) Enabled LWIP_UDP (UDP Module) Enabled LWIP_UDP (UDP Module) Enabled LWIP_UDP (UDP Module) Enabled LWIP_UDP (UDP Module) Enabled Enabled	Configure the below paramete	ers :		
 LwIP Version LwIP Version (Version of LwIP sup 1.4.1 DHCP Option LWIP_DHCP (DHCP Module) Enabled RTOS Settings WITH_RTOS (Use FREERTOS ** Disabled Protocols Options LWIP_ICMP (ICMP Module Activati Enabled LWIP_IGMP (IGMP Module) Disabled LWIP_DNS (DNS Module) Disabled LWIP_UDP (UDP Module) Enabled LWIP_UDP (UDP Module) Disabled LWIP_UDP (UDP Module) Enabled LWIP_UDP (UDP Module) Enabled LWIP_UDP (UDP Module) Enabled LWIP_UDP (UDP Module) Enabled LWIP_UDP (UDP Module) Enabled LWIP_UDP (UDP Module) Enabled LWIP_UDP (UDP Module) Enabled LWIP_UDP (UDP Module) Enabled LWIP_UDP (UDP Module) Enabled LWIP_UDP (UDP Module) Enabled LWIP_UDP (UDP Module) Enabled Enabled	Search (Crtl+E)			
LwIP Version (Version of LwIP sup 1.4.1 DHCP Option Enabled LWIP_DHCP (DHCP Module) Enabled RTOS Settings Disabled WITH_RTOS (Use FREERTOS ** Disabled Protocols Options LWIP_ICMP (ICMP Module Activati LWIP_IGMP (IGMP Module) Disabled LWIP_DNS (DNS Module) Disabled LWIP_UDP (UDP Module) Enabled			• •	E
 DHCP Option LWIP_DHCP (DHCP Module) Enabled RTOS Settings WITH_RTOS (Use FREERTOS ** Disabled Protocols Options LWIP_ICMP (ICMP Module Activati Enabled LWIP_IGMP (IGMP Module) Disabled LWIP_DNS (DNS Module) Disabled LWIP_UDP (UDP Module) Enabled 	LwIP Version			·
LWIP_DHCP (DHCP Module) Enabled RTOS Settings Disabled WITH_RTOS (Use FREERTOS ** Disabled Protocols Options LWIP_ICMP (ICMP Module Activati LWIP_IGMP (IGMP Module) Disabled LWIP_DNS (DNS Module) Disabled LWIP_UDP (UDP Module) Enabled	LwIP Version (Versio	on of LwIP sup	1.4.1	
 RTOS Settings WITH_RTOS (Use FREERTOS ** Disabled Protocols Options LWIP_ICMP (ICMP Module Activati Enabled LWIP_IGMP (IGMP Module) Disabled LWIP_DNS (DNS Module) Disabled LWIP_UDP (UDP Module) Enabled 	DHCP Option			
WITH_RTOS (Use FREERTOS ** Disabled Protocols Options LWIP_ICMP (ICMP Module Activati Enabled LWIP_IGMP (IGMP Module) Disabled LWIP_DNS (DNS Module) Disabled LWIP_UDP (UDP Module) Enabled	LWIP_DHCP (DHCP	Module)	Enabled	
WITH_RTOS (Use FREERTOS ** Disabled Protocols Options LWIP_ICMP (ICMP Module Activati Enabled LWIP_IGMP (IGMP Module) Disabled LWIP_DNS (DNS Module) Disabled LWIP_UDP (UDP Module) Enabled	RTOS Settings			
LWIP_ICMP (ICMP Module Activati Enabled LWIP_IGMP (IGMP Module) Disabled LWIP_DNS (DNS Module) Disabled LWIP_UDP (UDP Module) Enabled	WITH_RTOS (Use F	REERTOS **	Disabled	
LWIP_IGMP (IGMP Module) Disabled LWIP_DNS (DNS Module) Disabled LWIP_UDP (UDP Module) Enabled	Protocols Options			
LWIP_DNS (DNS Module) Disabled LWIP_UDP (UDP Module) Enabled	LWIP_ICMP (ICMP)	Module Activati.	Enabled	
LWIP_UDP (UDP Module) Enabled	LWIP_IGMP (IGMP)	Module)	Disabled	
	LWIP_DNS (DNS Mo	dule)	Disabled	
MEMP NUM LIDP PCB (Number of 4	LWIP_UDP (UDP Mo	dule)	Enabled	
	MEMP NUM LIDP P	CB (Number of	4	

Figure 626. LwIP 1.4.1 configuration



Figure 627. LwIP 1.5 configuration

Perf/Checks General Settings	Statistics	Checksum	Debug	SNMP	Constants
onfigure the below param	V		V	V 1	V 2
Search : Search (Crtl+F)		•			
LwIP Version					*
LwIP Version (Ve	rsion of LwIP support	ed b 1.5.0_RC0_	20160211		
DHCP Option					
LWIP_DHCP (DH	CP Module)	Enabled			
RTOS Settings					
WITH_RTOS (Us	e FREERTOS ** Cube	MX Disabled			
Protocols Options					E
LWIP_ICMP (ICM	IP Module Activation)				
LWIP_IGMP (IGN	-	Disabled			
LWIP_DNS (DNS		Disabled			
LWIP_UDP (UDP		Enabled			
	PCB (Number of UD				
LWIP_TCP (TCP	Module)	Enabled			-

STM32CubeMX generated C code reports compilation errors when specific parameters are enabled (disabled by default). The user must fix the issues with a stack patch (downloaded from Internet) or user C code. The following parameters generate an error:

- MEM_USE_POOLS: user C code to be added either in *lwipopts.h* or in *cc.h* (stack file).
- PPP_SUPPORT, PPPOE_SUPPORT: user C code required
- MEMP_SEPARATE_POOLS with MEMP_OVERFLOW_CHECK > 0: a stack patch required
- MEM_LIBC_MALLOC & RTOS enabled: stack patch required
- LWIP_EVENT_API: stack patch required

In STM32CubeMX, the user must enable FreeRTOS in order to use LwIP with the netconn and sockets APIs. These APIs require the use of threads and consequently of an operating system. Without FreeRTOS, only the LwIP event-driven raw API can be used.



B.3.7 Libjpeg

Libjpeg is a widely used C-library that allows reading and writing JPEG files. It is delivered within STM32CubeF7, STM32CubeH7, STM32CubeF2 and STM32CubeF4 embedded software packages.

STM32CubeMX generates the following files, whose content can be configured by the user through STM32CubeMX user interface:

• libjpeg.c/.h

The *MX_LIBJPEG_Init()* initialization function is generated within the libjpeg.c file. It is empty. It is up to the user to enter in the user sections the code and the calls to the libjpeg functions required for the application.

jdata_conf.c

This file is generated only when FatFs is selected as data stream management type.

• jdata_conf.h

The content of this file is adjusted according to the datastream management type selected.

jconfig.h

This file is generated by STM32CubeMX. but cannot be configured.

jmorecfg.h

Some but not all the define statements contained in this file can be modified through the STM32CubeMX libjpeg configuration menu.



Figure 628. Libjpeg configuration window

Configure the below parameters :		
Search : Search (Crtl+F)	🗢 🔶 📝 Show Advanced Parameters	
Version		
LIBJPEG version	8d	
MW configuration		
Data Stream management type	FatFS	
FREERTOS	Enabled	Ξ
General Settings		-
Use FREERTOS Memory Allocator	Disabled	
 JPEG basic settings 		
BITS_IN_JSAMPLE	8 bits	
MAX_COMPONENTS	12	
JCOEF	short	
JPEG encoder and decoder common ca		
DCT_ISLOW_SUPPORTED	Enabled	
DCT_IFAST_SUPPORTED	Enabled	
DCT_FLOAT_SUPPORTED	Enabled	
JPEG encoder options		
C_ARITH_CODING_SUPPORTED	Enabled	
C_MULTISCAN_FILES_SUPPORT.	Disabled	Ŧ
Data Stream management type		
LIBJPEG_FS_type		
Parameter Description: Default Value : FatFS		-

B.3.8 Mbed TLS

Mbed TLS is a C-library that allows including cryptographic capabilities to embedded products. It handles Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols, that are used for establishing a secure, encrypted and authenticated link between two parties over an insecure network. Mbed TLS comes with an intuitive API and minimal coding footprint. Visit https://tls.mbed.org/ for more details.

Mbed TLS is delivered within STM32CubeF2, STM32CubeF4, STM32CubeF7 and STM32CubeH7 embedded software packages.

Mbed TLS can work without LwIP stack (see Figure 629).

If LwIP stack is used, FreeRTOS must be enabled as well (see Figure 630).



STM32CubeMX generates the following files, whose contents can be modified by the user through STM32CubeMX user interface (see *Figure 631*) and/or using user sections in the code itself:

- mbedtls_config.h
- mbedtls.h
- *net_sockets.c* (generated only if LwIP is enabled)
- mbedtls.c

Figure	629.	Mbed	TLS	without	LwIP
--------	------	------	-----	---------	------

MBEDTLS Configuration	X
Modules Modules Co	
Version and modes	e support 🛛 🛷 Alternate implementation
Configure the below parameters :	
Search : Search (Crtl+F)	- ◆ ◆
Version	2.4.0
MBEDTLS version	2.4.0
	None
	None
RNG IP	SW RNG
Modes	500 1010
MBEDTLS_SSL_CLI_C	Not Defined
MBEDTLS_SSL_SRV_C	Not Defined
Restore Default	Apply Ok Cancel



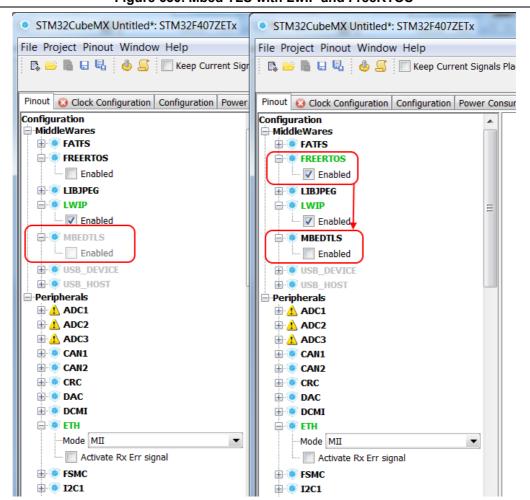


Figure 630. Mbed TLS with LwIP and FreeRTOS



in 🔠

Figure 631. Mbed TLS configuration window

B.3.9 TouchSensing

The STM32 TouchSensing library is a C-library that allows the creation of higher-end human interfaces by replacing conventional electromechanical switches by capacitive sensors with STM32 microcontrollers.

It requires the touch-sensing peripheral to be configured on the microcontroller.

STM32CubeMX generates the following files, whose contents can be modified by the user through STM32CubeMX user interface (see *Figure 632*, *Figure 633*, and *Figure 634*) and/or using user sections in the code itself:

- touchsensing.c/.h
- tsl_user.c/.h
- tsl_conf.h



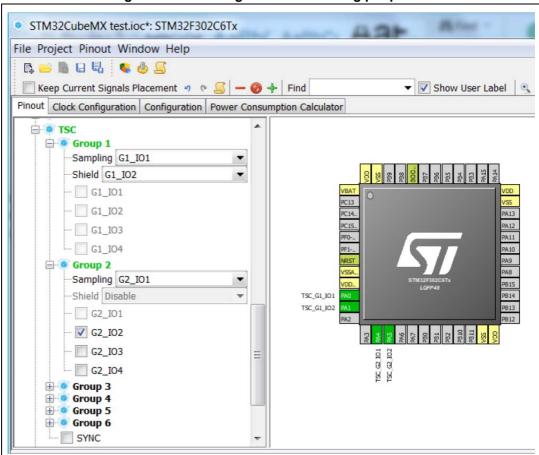


Figure 632. Enabling the TouchSensing peripheral



Configure the below parameters : Search : Search (CrtI+F) Summary	
Summary	
TSLPRM_TOTAL_OBJECTS 1	
TSC_ACTIVE_CHANNELS = 1	
TSLPRM_TOTAL_CHANNELS 1	
± Linear/Rotary sensors used	
TouchKey sensors	
TSLPRM_TOTAL_TOUCHKEYS 1	
IO_TOUCHKEY1 G2_IO2	
TSLPRM_TOTAL_TOUCHKE 0	
TSLPRM_TOTAL_OBJECTS TSLPRM_TOTAL_OBJECTS must be 1. Parameter Description: Total number of sensors/objects in application Count all TouchKeys, Linear and Rotary sensors	

Figure 633. Touch-sensing sensor selection panel



🖋 Sensors selection 🗹 Config param	eters 🎻 User Constants	
Configure the below parameters :		
Search : Search (Crtl+F)		
Version and modes		
TouchSensing version	2.2.0	
Optional features		
Acquisition limits		
Calibration		
Thresholds for TouchKey sensors		
Thresholds for Linear and Rotary s		
Linear/Rotary sensors position		
Debounce counters		
Environment Change System (ECS) Detection Time Out (DTO)		
Detection Time Out (DTO) Detection Exclusion System (DXS)		
Miscellaneous parameters		

Figure 634. TouchSensing configuration panel

B.3.10 PDM2PCM

The PDM2PCM library is a C-library that allows converting a pulse density modulated (PDM) data output into a 16-bit pulse-code modulation (PCM) format. It requires the CRC peripheral to be enabled.

STM32CubeMX generates the following files, whose content can be modified by the user through STM32CubeMX user interface and/or using user sections in the code itself:

• pdm2pcm.h/.c



B.3.11 STM32WPAN BLE/Thread (STM32WB series only)

STM32WPAN BLE and Thread middleware are now supported in STM32CubeMX.

Pinout & Configuration	Clock Configuration	Project Manager
	Additional Softwa	ares V Pinout
Options Q ~	STM32_WPAN Mode an	d Configuration
Categories A->Z	Mode	
System Core >	BLE Disabled: Active only if RF, RTC &	HSEM are enabled & FreeRTOS is disabled
Analog >	THREAD	
Timers >		
Connectivity >		
Multimedia		
Security >		
Computing >		
Middleware ~	Configuratio	n
FATFS FREERTOS STM32_WPAN TOUCHSENSING USB_DEVICE	Comgarato	

Figure 635. BLE and Thread middleware support in STM32CubeMX

They are exclusive in a given project and configuration with FreeRTOS is not yet supported.



Application projects generated with STM32CubeMX can be found in the project folder of the STM32CubeWB MCU package.

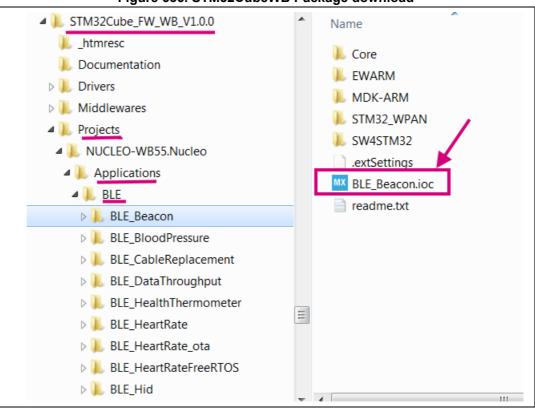


Figure 636. STM32CubeWB Package download



This package can be installed through STM32CubeMX following the standard procedure described in *Section 3.4.3: Installing STM32 MCU packages*.

_		1.94.000	••	
	MX Er	mbedded Software Packages M	1anager	X
ſ		STM32Cube MCU Packag	jes and embedded software packs releases	
		B Releases Information was la	ast refreshed less than one hour ado	
	STN	132Cube MCU Packages		
		Description	Installed Version Available V	ersion
	۲	STM32F7		
	•	STM32G0	Downloading selected software packages	
	•	STM32H7	Unzip File : stm32cube_fw_wb_v100.zip	
	•	STM32L0	Download and Unzip selected Files	
	•	STM32L1		
	•	STM32L4	OK Cancel	
	•	STM32MP1		
	•	STM32WB		
		STM32Cube MCU Package f	for STM32WB Series (Size : 70.8 MB) 1.0.0	
	Deta	ails		
		132CubeWB Firmware Package V	V1.0.0	
	Mar	n Changes	D (CT) (200-4- 8- CT) (200D C) CT) (200D C 4	
		First release of STM32CubeWI	B (STM32Cube for STM32WB Series) supporting STM32WB55xx devices.	
	Fr	om Local From Url	Refresh Install Now Remove Now C	lose

Figure 637. STM32CubeWB BLE applications folder	beWB BLE applications folder
-------------------------------------------------	------------------------------

BLE configuration

To enable BLE some peripherals (RTC, HSEM, RF) must be activated first.

Then, an application type must be selected, it can be one among Transparent mode, Server profile, Router profile or Client profile.

Finally, the mode and other parameters relevant to this application type must be configured.

Note: The BLE Transparent mode and all Thread applications require either the USART or the LPUART peripheral to be configured as well.



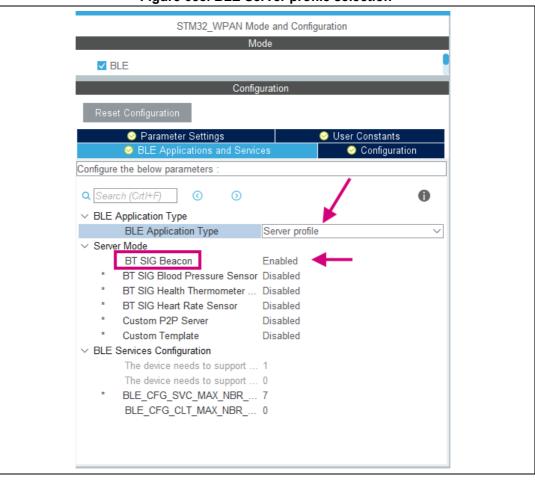
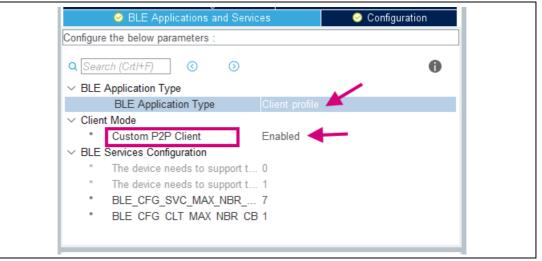


Figure 638. BLE Server profile selection







Thread configuration

To enable Thread some peripherals (RTC, HSEM, RF) must be activated first.

Then, an application type must be selected and the relevant parameters configured.

	Mode		
BLE			
THREAD			
	Configuration		
Reset Configuration			
_		<u> </u>	
🥝 Parameter Settings		😔 User Consta	nts
State of the second	and Services		nts guration
-	and Services		
THREAD Applications a Configure the below parameters :	and Services		
THREAD Applications a Configure the below parameters :			
 ✓ THREAD Applications a Configure the below parameters : Q Search (CrtI+F) Q (0) 	0		

Figure 640. Thread application selection

B.3.12 CMSIS packs selection limitation

The restriction about applications comes from a simple generated code consideration: an application is meant to be the root of the execution (excluding the main function).

This means that the generated function defines the execution of the selected application. In that sense, it is meant to be the last call of the main method, and must not give hand back to the main function. Two applications cannot be called, as this means generating calls in the main function, and then the second call is never reached.

If you need to call both applications:

- An RTOS must run them in threads, or
- You manually add the right code to execute them (in that context, they are not applications, as they are not at the root of the execution), or
- Change the meaning of the application components.





B.3.13 OpenAmp and RESMGR_UTILITY (STM32MPUs and STM32H7 dual-core products)

New software and hardware have been introduced on dual-core products to enable multi-core cooperation.

- For STM32MPUs only: the inter-processor communication controller (IPCC) used to exchange data between two processor instances relies on the fact that shared memory buffers are allocated in the MCU SRAM and that each processor owns specific register bank and interrupts.
- For STM32MPUs only: the OpenAMP middleware for intercommunication between Cortex-A and Cortex-M cores implements the RPMsg messaging protocol (see *Figure 641*).
- The resource manager library (RESMGR_UTILITY) for system resource management: multi-processor devices give the possibility to run independent firmware on several cores (see *Figure 642*). This implies a core could use some peripherals without knowledge of the usage of these same peripherals: the role of the resource management library is to control the assignment of a peripheral to a dedicated core and to provide a method to configure the system resources used to operate that peripheral (see *Figure 643*).

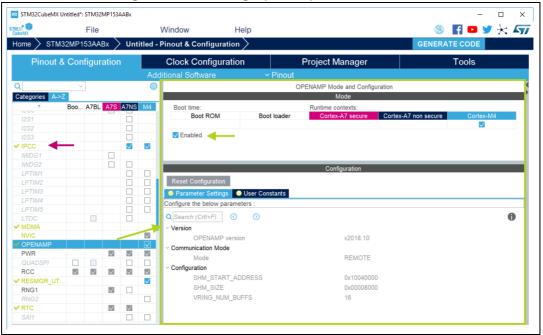


Figure 641. Enabling OpenAmp for STM32MPUs



Options Q ~	RESMGR_UTILITY Mode and Configuration
Categories A->Z	Mode Boot time: Runtime contexts: Boot ROM Boot loader A7S A7S A7NS Cortex-M4 ✓ ✓ ✓ ✓ Peripheral assignment request ✓ ✓ Dynamic system resources update ✓ Configuration Configuration ✓ Peripherals assignment ✓ User Constants
RNG2	Configure the below parameters :
SAI1 SAI2	Q Search (CrtI+F)
SAI3 SAI4 SDMMC1 SDMMC2	RESMGR_UTILITY version mp1/v1.4.0 Configuration use the RPMSG/OpenAMP-based extension true
SDMMC3 SPDIFRX SPI1 SPI2	
SPI3 SPI4 SPI5	

Figure 642. Enabling the Resource Manager for STM32MPUs



Pinout & Configuration					Clock Configu	ration	Project Mar	nager		
					Add	itional Software	~	Pinout		
Q	~				i		RESMG	R_UTILITY Mode and Config	juration	
Categories A->	Z							Mode		
•	Boo	A7BL	A7S	A7NS	M4	Boot time:		Runtime contexts:		
ADC1				\checkmark		Boot ROM	Boot loader	Cortex-A7 secure	A7NS	Cortex-M4
ADC2				\checkmark						\checkmark
BSEC			\checkmark	\checkmark		Peripheral assig	nment request			
CRC1				\checkmark		Dynamic system	resources update	9		
CRC2					\checkmark			-		
DAC1					\checkmark					
DCMI										
DDR				\checkmark						
DEBUG										
DFSDM1				\checkmark				Configuration		
✓ DMA						Reset Configuration	1			
DTS										
ETH1			-		_			s assignment 🛛 📀 User Con	stants	
ETZPC			\checkmark	\checkmark		Configure the below p	arameters :			
FDCAN1				\checkmark		Q Search (CrtI+F)	\odot \bigcirc			0
FDCAN2 FMC					\checkmark	~ RESMGR_UTILITY	Common			
FREERTOS							-	nt for ADC1 in resource table	is assid	ned to Cortex-A7NS
GIC			\checkmark	\checkmark	V			t for ADC2 in resource table		ned to Cortex-A7NS
GPIO			V.	$\overline{}$		1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -		it for CEC in resource table	0	ied to contex-M/NS
HASH1									no	
HASH2								t for CRC1 in resource table	0	ned to Cortex-A7NS
HDMI CEC								t for CRC2 in resource table	-	ned to Cortex-M4
HDP								nt for CRYP1 in resource tak		
HSEM						1		nt for CRYP2 in resource tab		
12C1						request per	ipheral assignmer	nt for DAC1 in resource table	e is assig	ned to Cortex-M4
12C2						request per	ripheral assignmer	nt for DBGMCU in resource	table no	
12C3						request per	ipheral assignmer	nt for DCMI in resource table	e no	
12C4						request per	ipheral assignmer	nt for DFSDM1 in resource t	able is assig	ned to Cortex-A7NS
12C5						request per	ipheral assignmer	nt for DLYB QUADSPI in re	sourc no	
12C6								t for DLYB SDMMC1 in res		
10.04						l loquot por				



For more details visit STM32MPUs dedicated wiki site at https://wiki.st.com/stm32mpu.



Appendix C STM32 microcontrollers naming conventions

STM32 microcontroller part numbers are codified following the below naming conventions:

Device subfamilies

The higher the number, the more features available.

For example STM32L0 line includes STM32L051, L052, L053, L061, L062, L063 subfamilies where STM32L06x part numbers come with AES while STM32L05x do not. The last digit indicates the level of features. In the above example:

- 1 = Access line
- 2 = with USB
- 3 = with USB and LCD.
- Pin counts
 - F = 20 pins
 - G = 28 pins
 - K = 32 pins
 - T = 36 pins
 - S = 44 pins
 - C = 48 pins
 - R = 64 (or 66) pins)
 - M = 80 pins
 - O = 90 pins
 - V = 100 pins
 - Q = 132 pins (e. g. STM32L162QDH6)
 - Z = 144 pins
 - I = 176 (+25) pins
 - B = 208 pins (e. g. STM32F429BIT6)
 - N = 216 pins
- Flash memory sizes
 - 4 = 16 Kbytes of flash memory
 - 6 = 32 Kbytes of flash memory
 - 8 = 64 Kbytes of flash memory
 - B = 128 Kbytes of flash memory
 - C = 256 Kbytes of flash memory
 - D = 384 Kbytes of flash memory
 - E = 512 Kbytes of flash memory
 - F = 768 Kbytes of flash memory
 - G = 1024 Kbytes of flash memory
 - I = 2048 Kbytes of flash memory
- Packages
 - B = SDIP
 - H = BGA



- M = SO
- P = TSSOP
- T = LQFP
- U = VFQFPN
- Y = WLCSP

Figure 644 shows an example of STM32 microcontroller part numbering scheme.

Example:	STM32	F	439 V	Т	6	xxx
Device family						
STM32 = ARM-based 32-bit microcontroller						
Product type						
F = general-purpose						
Device subfamily						
437= STM32F437xx, USB OTG FS/HS, camera interface, Ethernet, cryptographic acceleration						
439= STM32F439xx, USB OTG FS/HS, camera interface, Ethernet, LCD-TFT, cryptographic acceleration						
Pin count						
V = 100 pins						
Z = 144 pins						
A = 169 pins						
I = 176 pins						
B = 208 pins						
N = 216 pins						
Flash memory size						
G = 1024 Kbytes of Flash memory				-		
I = 2048 Kbytes of Flash memory						
Package						
T = LQFP						
H = BGA						
Y = WLCSP						
Temperature range						
6 = Industrial temperature range, -40 to 85 °C.						
7 = Industrial temperature range, -40 to 105 °C.						
Options						
xxx = programmed parts						
TR = tape and reel						



UM1718 Rev 47

Appendix D STM32 microcontrollers power consumption parameters

This section provides an overview on how to use STM32CubeMX Power Consumption Calculator.

Microcontroller power consumption depends on chip size, supply voltage, clock frequency and operating mode. Embedded applications can optimize STM32 MCU power consumption by reducing the clock frequency when fast processing is not required and choosing the optimal operating mode and voltage range to run from. A description of STM32 power modes and voltage range is provided below.

D.1 Power modes

STM32 MCUs support different power modes (refer to STM32 MCU datasheets for full details).

D.1.1 STM32L1 series

STM32L1 microcontrollers feature up to 6 power modes, including 5 low-power modes:

Run mode

This mode offers the highest performance using HSE/HSI clock sources. The CPU runs up to 32 MHz and the voltage regulator is enabled.

• Sleep mode

This mode uses HSE or HSI as system clock sources. The voltage regulator is enabled and the CPU is stopped. All peripherals continue to operate and can wake up the CPU when an interrupt/event occurs.

• Low- power run mode

This mode uses the multispeed internal (MSI) RC oscillator set to the minimum clock frequency (131 kHz) and the internal regulator in low-power mode. The clock frequency and the number of enabled peripherals are limited.

• Low-power sleep mode

This mode is achieved by entering Sleep mode. The internal voltage regulator is in lowpower mode. The clock frequency and the number of enabled peripherals are limited. A typical example would be a timer running at 32 kHz.

When the wake-up is triggered by an event or an interrupt, the system returns to the Run mode with the regulator ON.

Stop mode

This mode achieves the lowest power consumption while retaining RAM and register contents. Clocks are stopped. The real-time clock (RTC) an be backed up by using LSE/LSI at 32 kHz/37 kHz. The number of enabled peripherals is limited. The voltage regulator is in low-power mode.

The device can be woken up from Stop mode by any of the EXTI lines.

• Standby mode

This mode achieves the lowest power consumption. The internal voltage regulator is switched off so that the entire V_{CORE} domain is powered off. Clocks are stopped and the real-time clock (RTC) can be preserved up by using LSE/LSI at 32 kHz/37 kHz.



RAM and register contents are lost except for the registers in the Standby circuitry. The number of enabled peripherals is even more limited than in Stop mode.

The device exits Standby mode upon reset, rising edge on one of the three WKUP pins, or if an RTC event occurs (if the RTC is ON).

Note: When exiting Stop or Standby modes to enter the Run mode, STM32L1 MCUs go through a state where the MSI oscillator is used as clock source. This transition can have a significant impact on the global power consumption. For this reason, the Power Consumption Calculator introduces two transition steps: **WU_FROM_STOP** and **WU_FROM_STANDBY**. During these steps, the clock is automatically configured to MSI.

D.1.2 STM32F4 series

STM32F4 microcontrollers feature a total of 5 power modes, including 4 low-power modes:

• Run mode

This is the default mode at power-on or after a system reset. It offers the highest performance using HSE/HSI clock sources. The CPU can run at the maximum frequency depending on the selected power scale.

• Sleep mode

Only the CPU is stopped. All peripherals continue to operate and can wake up the CPU when an interrupt/even occurs. The clock source is the clock that was set before entering Sleep mode.

• Stop mode

This mode achieves a very low power consumption using the RC oscillator as clock source. All clocks in the 1.2 V domain are stopped as well as CPU and peripherals. PLL, HSI RC and HSE crystal oscillators are disabled. The content of registers and internal SRAM are kept.

The voltage regulator can be put either in normal Main regulator mode (MR) or in Lowpower regulator mode (LPR). Selecting the regulator in low-power regulator mode increases the wake-up time.

The flash memory can be put either in Stop mode to achieve a fast wake-up time. or in Deep power-down to obtain a lower consumption with a slow wake-up time.

The Stop mode features two sub-modes:

- Stop in Normal mode (default mode)

In this mode, the 1.2 V domain is preserved in nominal leakage mode and the minimum V12 voltage is 1.08 V.

Stop in Under-drive mode

In this mode, the 1.2 V domain is preserved in reduced leakage mode and V12 voltage is less than 1.08 V. The regulator (in Main or Low-power mode) is in under-drive or low-voltage mode. The flash memory must be in Deep-power-down mode. The wake-up time is about 100 μ s higher than in normal mode.

• Standby mode

This mode achieves very low power consumption with the RC oscillator as a clock source. The internal voltage regulator is switched off so that the entire 1.2 V domain is powered off: CPU and peripherals are stopped. The PLL, the HSI RC and the HSE crystal oscillators are disabled. SRAM and register contents are lost except for registers in the backup domain and the 4-byte backup SRAM when selected. Only RTC and LSE oscillator blocks are powered. The device exits Standby mode when an



external reset (NRST pin), an IWDG reset, a rising edge on the WKUP pin, or an RTC alarm/ wake-up/tamper/time stamp event occurs.

V_{BAT} operation

It allows to significantly reduced power consumption compared to the Standby mode. This mode is available when the V_{BAT} pin powering the Backup domain is connected to an optional standby voltage supplied by a battery or by another source. The V_{BAT} domain is preserved (RTC registers, RTC backup register and backup SRAM) and RTC and LSE oscillator blocks powered. The main difference compared to the Standby mode is external interrupts and RTC alarm/events do not exit the device from V_{BAT} operation. Increasing V_{DD} to reach the minimum threshold does.

D.1.3 STM32L0 series

STM32L0 microcontrollers feature up to 8 power modes, including 7 low-power modes to achieve the best compromise between low-power consumption, short startup time and available wake-up sources:

Run mode

This mode offers the highest performance using HSE/HSI clock sources. The CPU can run up to 32 MHz and the voltage regulator is enabled.

Sleep mode

This mode uses HSE or HSI as system clock sources. The voltage regulator is enabled and only the CPU is stopped. All peripherals continue to operate and can wake up the CPU when an interrupt/event occurs.

• Low-power run mode

This mode uses the internal regulator in low-power mode and the multispeed internal (MSI) RC oscillator set to the minimum clock frequency (131 kHz). In Low-power run mode, the clock frequency and the number of enabled peripherals are both limited.

• Low-power sleep mode

This mode is achieved by entering Sleep mode with the internal voltage regulator in low-power mode. Both the clock frequency and the number of enabled peripherals are limited. Event or interrupt can revert the system to Run mode with regulator on.

• Stop mode with RTC

The Stop mode achieves the lowest power consumption with, while retaining the RAM, register contents and real time clock. The voltage regulator is in low-power mode. LSE or LSI is still running. All clocks in the V_{CORE} domain are stopped, the PLL, MSI RC, HSE crystal and HSI RC oscillators are disabled.

Some peripherals featuring wake-up capability can enable the HSI RC during Stop mode to detect their wake-up condition. The device can be woken up from Stop mode by any of the EXTI line, in 3.5 μ s, and the processor can serve the interrupt or resume the code.

• Stop mode without RTC

This mode is identical to "Stop mode with RTC ", except for the RTC clock which is stopped here.

• Standby mode with RTC

The Standby mode achieves the lowest power consumption with the real time clock running. The internal voltage regulator is switched off so that the entire V_{CORE} domain



is powered off. The PLL, MSI RC, HSE crystal and HSI RC oscillators are also switched off. The LSE or LSI is still running.

After entering Standby mode, the RAM and register contents are lost except for registers in the Standby circuitry (wake-up logic, IWDG, RTC, LSI, LSE crystal 32 kHz oscillator, RCC_CSR register).

The device exits Standby mode in 60 μ s when an external reset (NRST pin), an IWDG reset, a rising edge on one of the three WKUP pins, RTC alarm (Alarm A or Alarm B),

RTC tamper event, RTC timestamp event or RTC wake-up event occurs.

• Standby mode without RTC

This mode is identical to Standby mode with RTC, except that the RTC, LSE and LSI clocks are stopped.

The device exits Standby mode in 60 μ s when an external reset (NRST pin) or a rising edge on one of the three WKUP pin occurs.

Note: The RTC, the IWDG, and the corresponding clock sources are not stopped automatically by entering Stop or Standby mode. The LCD is not stopped automatically by entering Stop mode.

D.2 Power consumption ranges

STM32 MCUs power consumption can be further optimized thanks to the dynamic voltage scaling feature: the main internal regulator output voltage V12 that supplies the logic (CPU, digital peripherals, SRAM and flash memory) can be adjusted by software by selecting a power range (STM32L1 and STM32L0) or power scale (STM32 F4).

Power consumption range definitions are provided below (refer to STM32 MCU datasheets for full details).

D.2.1 STM32L1 series features three V_{CORE} ranges

 High performance Range 1 (V_{DD} range limited to 2.0-3.6 V), with the CPU running at up to 32 MHz

The voltage regulator outputs a 1.8 V voltage (typical) as long as the V_{DD} input voltage is above 2.0 V. Flash program and erase operations can be performed.

 Medium performance Range 2 (full V_{DD} range), with a maximum CPU frequency of 16 MHz

At 1.5 V, the flash memory is still functional but with medium read access time. Program and erase operations are still possible.

 Low performance Range 3 (full V_{DD} range), with a maximum CPU frequency limited to 4 MHz (generated only with the multispeed internal RC oscillator clock source)

At 1.2 V, the flash memory is still functional but with slow read access time. Program and erase operations are no longer available.



D.2.2 STM32F4 series features several V_{CORE} scales

The scale can be modified only when the PLL is OFF and when HSI or HSE is selected as system clock source.

- Scale 1 (V12 voltage range limited to 1.26 1.40 V), default mode at reset. HCLK frequency range = 144 MHz to 168 MHz (180 MHz with over-drive). This is the default mode at reset.
- Scale 2 (V12 voltage range limited to 1.20 1.32 V).
 HCLK frequency range is up to 144 MHz (168 MHz with over-drive).
- Scale 3 (V12 voltage range limited to 1.08 1.20 V), default mode when exiting Stop mode.

HCLK frequency ≤120 MHz.

The voltage scaling is adjusted to $f_{\mbox{HCLK}}$ frequency as follows:

- STM32F429x/39x MCUs:
 - Scale 1: up to 168 MHz (up to 180 MHz with over-drive)
 - Scale 2: from 120 to 144 MHz (up to 168 MHz with over-drive)
 - Scale 3: up to 120 MHz.
- STM32F401x MCUs:

No Scale 1

- Scale 2: from 60 to 84 MHz
- Scale 3: up to 60 MHz.
- STM32F40x/41x MCUs:
 - Scale 1: up to 168 MHz
 - Scale 2: up to 144 MHz

D.2.3 STM32L0 series features three V_{CORE} ranges

- Range 1 (V_{DD} range limited to 1.71 to 3.6 V), with CPU running at a frequency up to 32 MHz
- Range 2 (full V_{DD} range), with a maximum CPU frequency of 16 MHz
- Range 3 (full V_{DD} range), with a maximum CPU frequency limited to 4.2 MHz.



Appendix E STM32Cube embedded software packages

Along with STM32CubeMX C code generator, embedded software packages are part of STM32Cube initiative (refer to *DB2164 databrief*): these packages include a low-level hardware abstraction layer (HAL) that covers the microcontroller hardware, together with an extensive set of examples running on STMicroelectronics boards (see *Figure 645*). This set of components is highly portable across the STM32 series. The packages are fully compatible with STM32CubeMX generated C code.

Evaluation board demonstration (Demo builder framework) Discovery board demonstration Dedicated board demonstration Application level demonstrations							
Middleware examples							
TCP/IP IwIP stack + Polar SSL Middleware level TCP/IP IwIP stack + Device library Middleware level TCP/IP USB Host & Device library STEmWin Image State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State State St							
HAL examples Hardware Abstraction Layer APIs (HAL) Board Support Package (BSP) HAL APIs Utilities Utilities							
MCU Series (STM32F4, F1, F2, F3) Hardware		MSv34720V2					

Figure 645. STM32Cube Embedded Software package

Note:

STM32CubeF0, STM32CubeF1, STM32CubeF2, STM32CubeF3, STM32CubeF4, STM32CubeL0 and STM32CubeL1 embedded software packages are available on st.com. They are based on STM32Cube release v1.1 (other series will be introduced progressively) and include the embedded software libraries used by STM32CubeMX for initialization C code generation.

The user should use STM32CubeMX to generate the initialization C code and the examples provided in the package to get started with STM32 application development.



Revision history

Date	Revision	STM32CubeMX release number	Changes
17-Feb-2014	1	4.1	Initial release.
		4.2	Added support of STM32CubeF2 and STM32F2 Series in cover page, Section 2.2: Key features, Section 5.14.1: Peripherals and Middleware Configuration window, and Appendix E: STM32Cube embedded software packages.
04-Apr-2014	2		Updated Section 11.1: Creating a new STM32CubeMX project, Section 11.2: Configuring the MCU pinout, Section 11.6: Configuring the MCU initialization parameters.
			Section "Generating GPIO initialization C code move to Section 8: Tutorial 3- Generating GPIO initialization C code (STM32F1 Series only) and content updated.
			Added Section 18.6: Why do I get the error "Java 8 update 45" when installing "Java 8 update 45" or a more recent version of the JRE?.
	3	3 4.3	Added support of STM32CubeL0 and STM32L0 Series in cover page, Section 2.2: Key features, Section 2.3: Rules and limitations and Section 5.14.1: Peripherals and Middleware Configuration window
			Added board selection in <i>Table 13: File menu functions</i> , Section 5.7.3: Pinout menu and Section 4.2: New Project window. Updated <i>Table 15: Pinout menu</i> .
24-Apr-2014			Updated <i>Figure 318: Power Consumption Calculator default view</i> and added battery selection in <i>Section 5.3.1: Building a power consumption sequence</i> .
247012014	0		Updated note in Section 5.3: Power Consumption Calculator view
			Updated Section 11.1: Creating a new STM32CubeMX project.
			Added Section 19.7: Why does the RTC multiplexer remain inactive on the Clock tree view?, Section 19.8: How can I select LSE and HSE as clock source and change the frequency?, and Section 19.9: Why STM32CubeMX does not allow me to configure PC13, PC14, PC15, and PI8 as outputs when one of them is already configured as an output?.



Date	Revision	STM32CubeMX release number	Changes
19-Jun-2014	4	4.4	Added support of STM32CubeF0, STM32CubeF3, STM32F0 and STM32F3 Series in cover page, Section 2.2: Key features, Section 2.3: Rules and limitations, Added board selection capability and pin locking capability in Section 2.2: Key features, Table 2: Home page shortcuts, Section 4.12: New Project window, Section 5.7: Toolbar and menus, Section 4.13: Set unused/reset used GPIOs windows, Section 4.11: Project Manager view, and Section 5.15: Pinout view. Added Section 5.15.1: Pinning and labeling signals on pins. Updated Section 5.16: Configuration view and Section 4.10: Clock Configuration view and Section 5.3: Power Consumption Calculator view. Updated Figure 50: STM32CubeMX Main window upon MCU selection, Figure 140: STM32CubeMX Pinout view, Figure 120: Chip view, Figure 318: Power Consumption Calculator default view, Figure 319: Battery selection, Figure 87: Building a power consumption sequence, Figure 321: Power consumption sequence: New Step default view, Figure 323: Power Consumption Calculator view after sequence building, Figure 83: Sequence table management functions, Figure 88: PCC Edit Step window, Figure 83: Power consumption sequence: new step configured (STM32F4 example), Figure 326: ADC selected in Pinout view, Figure 327: Power Consumption Calculator configuration window: ADC enabled using import pinout, Figure 331: Description of the Results area, Figure 100: Peripheral power consumption tooltip, Figure 538: Power Consumption Calculation example, Figure 155: Sequence table and Figure 156: Power Consumption Calculator view and Figure 39: STM32CubeMX Configuration view - STM32F1 Series titles. Added STM32L1 in Section 5.3: Power Consumption Calculator view. Removed Figure Add a new step using the PCC panel from Section 8.1.1: Adding a step. Removed Figure Add a new step to the sequence. Updated Section 8.2: Reviewing res

Table 27. Document revision history (continued)



Date	Revision	STM32CubeMX release number	Changes
Date	Revision		Added support of STM32CubeL1 Series in cover page, Section 2.2: Key features, Section 2.3: Rules and limitations, Updated Section 3.2.3: Uninstalling STM32CubeMX standalone version. Added off-line updates in Section 3.4: Getting updates using STM32CubeMX, modified Figure 21: Embedded Software Packages Manager window, and Section 3.4.3: Installing STM32 MCU packages. Updated Section 4: STM32CubeMX user interface introduction, Table 2: Home page shortcuts and Section 4.2: New Project window. Added Figure 42: New Project window - Board selector. Updated Figure 192: Project Settings code generator. Modified step 3 in Section 4.11: Project Manager view. Updated Figure 39: STM32CubeMX Configuration view - STM32F1 Series. Added STM32L1 in Section 5.14.1: Peripherals and Middleware Configuration window. Updated Figure 83: GPIO configuration window - GPIO selection; Section 4.5.12: GPIO configuration window and Figure 88: DMA Mem ToMem configuration. Updated introduction of Section 4.10: Clock Configuration view. Updated Section 4.10.1: Clock tree configuration view. Updated Section 4.10.1: Clock tree consumption sequence: New Step default view, Figure 321: Power consumption Sequence: New Step default view, Figure 328: Power Consumption sequence: New Step default view, Figure 328: Power Consumption sequence: new step configured (STM32F4 example), and Figure 327: Power Consumption claculator configuration window: ADC enabled using import pinout. Added Figure 330: Power Consumption: Peripherals consumption chart and updated Figure 100: Peripheral power consumption claculator configuration View Code generation overview. Updated Section 6: STM32CubeMX C Code generation overview.

Table 27. Document revision history (continued)



Complete project generation, power consumption clock tree configuration now available on all STM3 Updated Section 2.2: Key features and Section 2. limitations. Updated Eclipse IDEs in Section 3.1.3: Software of Updated Figure 18: Updater Settings window, Fig Software Packages Manager window and Figure window - Board selector, Updated Section 4.11: F	
 view and Section 4.14: Update Manager windows. Updated Figure 299: About window. Removed Figure STM32CubeMX Configuration v Series. Updated Table 17: STM32CubeMX Chip view - Ic scheme. Updated Section 5.14.1: Peripherals and Middlew window. Updated Figure 86: Adding a new DMA request a MemToMem configuration. Updated Figure 319: Battery selection, Figure 87: consumption sequence, Figure 88: PCC Edit Step Added Section 6.3: Custom code generation. Updated Figure 492: Clock tree view and Figure 4 Configuration view. Updated Pigure 503: GPIO mode configuration. Updated Figure 503: GPIO mode configuration. Updated Figure 503: GPIO mode configuration. Updated Figure 155: Sequence table. Updated Figure 155: Sequence table. Updated Appendix A.1: Block consistency, A.2: Bi dependency and A.3: One block = one peripheral Appendix A.4: Block remapping (STM32F10x only Example. Appendix A.6: Block shifting (only for STM32F10x Current Signals placement" is unchecked): updated Example. Updated Appendix A.8: Mapping a function individ Updated Appendix A.8: Mapping a function individ Updated Appendix A.8: Mapping a function individ Updated Appendix D.1.3: STM32L0 series. 	32 Series. .3: Rules and requirements. Jure 21: Embedded 42: New Project Project Manager riew - STM32F1 tons and color vare Configuration nd Figure 88: DMA functions. Building a power to window. 497: Pinout & Figure 499: Timer 3 ing the peripherals. in C code ation example and lock inter- mode. 1): updated Section : (Cand when "Keep ed Section :

Table 27. Document revision history (continued)



Date	Revision	STM32CubeMX release number	Changes
			Section 2.2: Key features: removed Pinout initialization C code generation for STM32F1 Series from; updated Complete project generation. Updated Figure 21: Embedded Software Packages Manager window, Figure 42: New Project window - Board selector.
			Updated IDE list in <i>Section 4.11: Project Manager view</i> and modified <i>Figure 184: Project Settings window</i> . Updated <i>Section 4.10.1: Clock tree configuration functions</i> . Updated
19-Mar-2015	7	4.7	 Figure 180: STM32F469NIHx clock tree configuration view. Section 5.3: Power Consumption Calculator view: added transition checker option. Updated Figure 318: Power Consumption Calculator default view, Figure 319: Battery selection and Figure 87: Building a power consumption sequence. Added Figure 322: Enabling the transition checker option on an already configured sequence - All transitions valid, Figure 323: Enabling the transition checker option on an already configured sequence - At least one transition invalid and Figure 324: Transition checker option - Show log. Updated Figure 328: Power Consumption Calculator view after sequence building. Updated Section : Managing sequence steps, Section : Managing the whole sequence (load, save and compare). Updated Figure 88: PCC Edit Step window and Figure 331: Description of the Results area. Updated Figure 538: Power Consumption Calculation example, Figure 155: Sequence table, Figure 156: Power Consumption Calculation results - IP consumption chart.
			Updated Appendix <i>B.3.1: Overview</i> and <i>B.3.5: FreeRTOS</i> .
28-May-2015	8	4.8	Added Section 3.2.2: Installing STM32CubeMX from command line and Section 3.3.2: Running STM32CubeMX in command-line mode.
09-Jul-2015	9	4.9	 Added STLM32F7 and STM32L4 microcontroller Series. Added Import project feature. Added Import function in Table 13: File menu functions. Added Section 4.12: Import Project window. Updated Figure 321: Power consumption sequence: New Step default view, Figure 88: PCC Edit Step window, Figure 83: Power consumption sequence: new step configured (STM32F4 example), Figure 327: Power Consumption Calculator configuration window: ADC enabled using import pinout and Figure 87: Peripheral power consumption tooltip. Updated command line to run STM32CubeMX in Section 3.3.2: Running STM32CubeMX in command-line mode. Updated note in Section 5.16: Configuration view. Added new clock tree configuration functions in Section 4.10.1. Updated Figure 505: Middleware tooltip. Modified code example in Appendix B.1: STM32CubeMX generated C code and user sections. Updated Appendix B.3.1: Overview. Updated generated .h files in Appendix B.3.4: FatFs.

Table 27. Document revision history (continued)



Date	Revision	STM32CubeMX release number	Changes
27-Aug-2015	10	4.10	Replace UM1742 by UM1940 in Section : Introduction. Updated command line to run STM32CubeMX in command-line mode in Section 3.3.2: Running STM32CubeMX in command-line mode. Modified Table 1: Command line summary. Updated board selection in Section 4.2: New Project window. Updated Section 5.16: Configuration view overview. Updated Section 5.14.1: Peripherals and Middleware Configuration window, Section 4.5.12: GPIO configuration window and Section 4.5.13: DMA configuration window. Added Section 4.5.11: User Constants configuration window. Updated Section 4.10: Clock Configuration view and added reserve path. Updated Section 11.1: Creating a new STM32CubeMX project, Section 11.5: Configuring the MCU clock tree, Section 11.6: Configuring the MCU initialization parameters, Section 11.7.2: Downloading firmware package and generating the C code, Section 11.8: Building and updating the C code project. Added Section 11.9: Switching to another MCU. Updated Section 12: Tutorial 2 - Example of FatFs on an SD card using STM32429I-EVAL evaluation board and replaced STM32F429I- EVAL by STM32420I EVAL
16-Oct-2015	11	4.11	 EVAL by STM32429I-EVAL. Updated Figure 21: Embedded Software Packages Manager window and Section 3.4.7: Checking for updates. Character string constant supported in Section 4.5.11: User Constants configuration window. Updated Section 4.10: Clock Configuration view. Updated Section 5.3: Power Consumption Calculator view. Modified Figure 538: Power Consumption Calculation example. Updated Section 13: Tutorial 3 - Using the Power Consumption Calculator to optimize the embedded application consumption and more. Added Eclipse Mars in Section 3.1.3: Software requirements
03-Dec-2015	12	4.12	Code generation options now supported by the Project Settings menu. Updated Section 3.1.3: Software requirements. Added Project Settings in Section 4.12: Import Project window. Updated Figure 197: Automatic project import; modified Manual project import step and updated Figure 198: Manual project import and Figure 199: Import Project menu - Try Import with errors; modified third step of the import sequence. Updated Figure 83: Clock Tree configuration view with errors. Added mxconstants.h in Section 6.1: STM32Cube code generation using only HAL drivers (default mode). Updated Figure 538: Power Consumption Calculation example to Figure 547: Step 10 optimization. Updated Figure 548: Power sequence results after optimizations.

Table 27. Document revision history (continued)



Updated Section 2.2: Key features: – Information related to .ioc files. – Clock tree configuration – Automatic updates of STM32CubeMX and STM32Cube. Updated limitation related to STM32CubeMX C code general Section 2.3: Rules and limitations.	
 Added Linux in Section 3.1.1: Supported operating systems architectures. Updated Java Run Time Environment release in Section 3.1.3: Software requirements. Updated Section 3.2.1: Installing STM32CubeMX standalone versis Section 3.3.1: Downloading STM32CubeMX standalone versis Section 3.3.1: Downloading STM32CubeMX as a stand application. Updated Section 4.11: Project Manager view and Section 4. Updated Section 5.15.1: Pinning and labeling signals on pin Added Section 4.5.16: Setting HAL timebase source Updated Section 4.5.16: Setting HAL timebase source Updated Figure 143: Configuration window tabs for GPIO, D NVIC settings (STM32F4 Series). Added note related to GPIO configuration window, updated Figure 4.5.12: GPIO selection. Modified Figure 318: Power Consumption Calculator default Figure 86: Building a power consumption sequence, Figure 3.2: Enabling the transition of option on an already configured sequence - All transitions vier Figure 322: Enabling the transition invalid. Added Section : Selecting/deselecting all peripherals. Modified Figure 323: Enabling the transition invalid. Added Section : Selecting: Analoging the whole sequence (a save and compare). Updated Figure 331: Description of the area and Figure 100: Peripheral power consumption tooltip. Updated Section : Selecting: Section : Importing pinou Added Section : Selection : Importing pinou Added Section : Section : Managing the whole sequence (a save and compare). Updated Figure 331: Description of the area and Figure 100: Peripheral power consumption tooltip. Updated Section : Section : Managing the whole section and Figure 540: Sequence table. Updated Section : Section and the section in the section and figure 485: Pinout view with MCUs selection and Figure 540: Sequence table. 	e generation in systems and release number andalone version, one version and installation a standalone ection 4.14: 's on pins. GPIO, DMA and mode in Figure 83: GPIO r default view, Figure 320: Step insition checker sitions valid, n an already og pinout. s. Modified er sequence guence (load, in of the Results o tooltip. on example and

536/555



Table 27. Document revision history (continued)			
Date	Revision	STM32CubeMX release number	Changes
15-Mar-2016	14	4.14	 Upgraded STM32CubeMX released number to 4.14.0. Added import of previously saved projects and generation of user files from templates in Section 2.2: Key features. Added MacOS in Section 3.1.1: Supported operating systems and architectures, Section 3.2.1: Installing STM32CubeMX standalone version, Section 3.2.3: Uninstalling STM32CubeMX plug-in from Eclipse IDE. Added command lines allowing the generation of user files from templates in Section 3.3.2: Running STM32CubeMX in command-line mode. Updated new library installation sequence in Section 3.4.2: Updater configuration. Updated Figure 107: Pinout menus (Pinout tab selected) and Figure 108: Pinout menus (Pinout tab not selected) in Section 5.7.3: Pinout menu. Modified Table 16: Window menu. Updated Figure 101: NVIC settings when using SysTick as HAL timebase, no FreeRTOS and Figure 102: NVIC settings when using FreeRTOS and SysTick as HAL timebase in Section 4.5.16: Setting HAL timebase source. Updated Figure 74: User Constants tab and Figure 75: Extract of the generated main.h file in Section 4.5.11: User Constants configuration window. Section 4.5.12: GPIO configuration window: updated Figure 83: GPIO configuration window - GPIO selection, Figure 84: GPIO configuration. Updated Section 4.5.14: NVIC configuration window.
18-May-2016	15	4.15	Import project function is no more limited to MCUs of the same Series (see Section 2.2: Key features, Section 5.7.1: File menu and Section 4.12: Import Project window). Updated command lines in Section 3.3.2: Running STM32CubeMX in command-line mode. Table 1: Command line summary: modified all examples related to config comands as well as set dest_path <path> example. Added caution note for Load Project menu in Table 13: File menu functions. Updated Generate Code menu description in Table 14: Project menu. Updated Set unused GPIOs menu in Table 15: Pinout menu. Added case where FreeRTOS in enabled in Section : Enabling interruptions using the NVIC tab view. Added Section 4.5.15: FreeRTOS configuration panel. Updated Appendix B.3.5: FreeRTOS and B.3.6: LwIP.</path>

Table 27. Document revision history (continued)



Date	Revision	STM32CubeMX release number	Changes
23-Sep-2016	16	4.17	 Replaced <i>mxconstants.h</i> by <i>main.h</i> in the whole document. Updated <i>Introduction, Section 3.1.1: Supported operating systems</i> <i>and architectures</i> and <i>Section 3.1.3: Software requirements.</i> Added <i>Section 3.4.4: Installing STM32 MCU package patches.</i> Updated Load project description in <i>Table 2: Home page shortcuts.</i> Updated Clear Pinouts function in <i>Table 15: Pinout menu.</i> Updated Section <i>4.11.3: Advanced Settings tab</i> to add Low Layer driver. Added <i>No check</i> and <i>Decimal and hexadecimal check</i> options in <i>Table 17: Peripheral and Middleware Configuration window buttons</i> <i>and tooltips.</i> Updated Section : <i>Tasks and Queues tab</i> and <i>Figure 98: FreeRTOS</i> <i>heap usage.</i> Updated <i>Figure 83: GPIO configuration window - GPIO selection.</i> Replaced PCC by Power Consumption Calculator in the whole document. Added <i>Section 6.2: STM32Cube code generation using Low Layer</i> <i>drivers</i>; updated <i>Table 23: LL versus HAL: STM32CubeMX generated</i> <i>source files</i> and <i>Table 24: LL versus HAL: STM32CubeMX generated</i> <i>functions and function calls.</i> Updated <i>Figure 561: Pinout view - Enabling the RTC.</i> Added Section 14: <i>Tutorial 4 - Example of UART communications</i> <i>with an STM32L053xx Nucleo board.</i> Added correspondence between STM32CubeMX release number and document revision.
21-Nov-2016	17	4.18	 Removed Windows XP and added Windows 10 in Section 3.1.3: Software requirements. Updated Section 3.2.3: Uninstalling STM32CubeMX standalone version. Added setDriver command line in Table 1: Command line summary. Added List pinout compatible MCUs feature: Updated Table 15: Pinout menu. Added Section 15: Tutorial 5: Exporting current project configuration to a compatible MCU Added Firmware location selection option in Section 4.11.1: Project tab and Figure 184: Project Settings window. Added Restore Default feature: Updated Table 8: Peripheral and Middleware configuration window buttons and tooltips Updated Figure 76: Using constants for peripheral parameter settings.



Date	Revision	STM32CubeMX release number	Changes
12-Jan-2017	18	4.19	Project import no more limited to microcontrollers belonging to the same Series: updated Introduction, Figure 197: Automatic project import, Figure 198: Manual project import, Figure 199: Import Project menu - Try Import with errors and Figure 200: Import Project menu - Successful import after adjustments. Modified Appendix B.3.4: FatFs, B.3.5: FreeRTOS and B.3.6: LwIP.
			Added Appendix <i>B.3.7: Libjpeg</i> .
02-Mar-2017	19	4.20	Table 17: STM32CubeMX Chip view - Icons and color scheme:- Updated list of alternate function example.
			 Updated example and description corresponding to function mapping on a pin.
			 Added example and description for analog signals sharing the same pin.
			Updated Figure 87: Peripheral Configuration window (STM32F4 Series), Figure 74: User Constants tab, Figure 80: Consequence when deleting a user constant for peripheral configuration, Figure 81: Searching for a name in a user constant list and Figure 82: Searching for a value in a user constant list.
			Added Section 5.3.6: SMPS feature.
			Added Section 6.4: Additional settings for C project generation. Added STM32CubeF4 to the list of packages that include Libjpeg in Appendix B.3.7: Libjpeg.
05-May-2017	20	20 4.21	Minor modifications in Section 1: STM32Cube overview. Updated Figure 40: New Project window - MCU selector and Figure 184: Project Settings window.
			Updated description of Project Settings in Section 4.11.1: Project tab. Updated Figure 195: Advanced Settings window.
			In Appendix <i>B.3.7: Libjpeg</i> , added STM32CubeF2 and STM32CubeH7 in the list of software packages in which Libjpeg is embedded.
			Modified <i>Figure 645: STM32Cube Embedded Software package</i> look- and-feel.

Table 27. Document revision history (continued)



Date	Revision	STM32CubeMX release number	Changes
06-Jul-2017	21	4.22	Added STM32H7 to the list of supported STM32 Series. Added MCU data and documentation refresh capability in Section 3.4: Getting updates using STM32CubeMX and updated Figure 18: Updater Settings window. Added capability to identify close MCUs in Section 4.2: New Project window, updated Figure 40: New Project window - MCU selector, added Figure 29: New Project window - MCU list with close function and Figure 30: New Project window - List showing close MCUs., updated Figure 483: MCU selection.
			Updated <i>Figure 50: STM32CubeMX Main window upon MCU</i> <i>selection.</i> Added Rotate clockwise/Counter clockwise and Top/Bottom view in
			Table 15: Pinout menu. Added Section 4.1.4: Social links.
			Updated Figure 339: Configuring the SMPS mode for each step. Updated Section 6.2: STM32Cube code generation using Low Layer drivers.
			Updated Figure 510: Project Settings and toolchain selection.
05-Sep-2017	22	4.22.1	Added STM32L4+ Series in Introduction, Section 5.3: Power Consumption Calculator view and Section 6.2: STM32Cube code generation using Low Layer drivers. Added guidelines to run STM32CubeMX on MacOS in Section 3.3.1: Running STM32CubeMX as a standalone application. Removed MacOS from Section 3.4.3: Running STM32CubeMX plug-in from Eclipse IDE. Added Section 19.10: Ethernet configuration: why cannot I specify DP83848 or LAN8742A in some cases?
18-Oct-2017	23	4.23	Added Section 1: General information. Renamed Display close button into Display similar items in Section 4.2: New Project window. Added Refresh Data and Docs & Resources menus in Section 5.7.5: Help menu. Added STM32F2, STM32F4 and STM32F7 Series in Section 6.2: STM32Cube code generation using Low Layer drivers. Added Appendix B.3.8: Mbed TLS. Updated STM32CubeMX release number corresponding to user manual revision 22.

Table 27. Document revision history (continued)



	Table 27. Document revision history (continued)		
Date	Revision	release number	Changes
16-Jan-2018	24	4.24	Replaced "STM32Cube firmware package" by "STM32Cube MCU package". Updated Section 1: STM32Cube overview. Updated MacOS in Section 3.1.1: Supported operating systems and architectures. Updated Eclipse requirements in Section 3.1.3: Software requirements. Section 3.4: Getting updates using STM32CubeMX: – updated section introduction – updated Figure 13: Connection Parameters tab - No proxy – Section 3.4.3 renamed into "Installing STM32 MCU packages" and updated. – renamed Section 3.4.4 into "Installing STM32 MCU packages" and updated. – renamed Section 3.4.4 into "Installing STM32 MCU package patches" – added Section 3.4.5: Installing embedded software packs – updated Section 3.4.7: Checking for updates Updated Figure 42: New Project window - Board selector. Updated Figure 51: STM32CubeMX Main window upon board selection (peripherals not initialized) and introductory sentence. Updated Figure 52: STM32CubeMX Main window upon board selection (peripherals initialized with default configuration) and introductory sentence. Added "Select additional software components" menu in Table 14: Project menu. "Install new libraries" menu renamed "Manage embedded software packages" and corresponding description updated in Table 17: Help menu. Updated Section 4.16: Removing already installed embedded software packages. Updated Section 4.16: Software Packs component selection window. Added Section 4.17: Software Packs component selection window. Added Section 4.16: Software Packs component selection window. Added Section 4.17: Software Packs component selection window. Added Section 4.16: Software Packs component selection window. Added Section 1.16: Software Packs component selection window. Added Section 1.2: STM32Cube code generation using Low Layer driv
16-Jan-2018	24 (conťd)	4.24	Added Appendix <i>B.3.9: TouchSensing</i> and <i>B.3.10: PDM2PCM.</i> Section 4.5.14: NVIC configuration window/Default initialization sequence of interrupts: changed color corresponding to interrupt enabling code from green to black bold.

Table 27. Document revision history (continued)



Date	Revision	STM32CubeMX release number	Changes	
07-Mar-2018	25	4.25	 Updated Introduction, Section 1: STM32Cube overview, Section 2.3: Rules and limitations, Section 3.2.1: Installing STM32CubeMX standalone version, Section 4: STM32CubeMX user interface, Section 4.11.1: Project tab and Section 5.13.1: Peripheral and Middleware tree panel. Minor text edits across the whole document. Updated Table 13: File menu functions and Table 12: Relations between power over-drive and HCLK frequency. Updated Figure 40: New Project window - MCU selector, Figure 27: Enabling graphics choice in MCU selector, Figure 184: Project Settings window, Figure 189: Selecting a different firmware location, Figure 77: Enabling STemWin framework, Figure 116: Configuration view for Graphics, Figure 562: Pinout view - Enabling LSE and HSE clocks and Figure 563: Pinout view - Setting LSE/HSE clock frequency. Added Export to Excel, Show favorite MCUs and Section 4.4.16: Graphics frameworks and simulator. Added Section 17: Tutorial 8 – Using STemWin Graphics framework, Section 18: Tutorial 9: Using STM32CubeMX Graphics simulator and their subsections. Added Section B.3.11: Graphics. 	

Table 27. Document revision history (continued)



Date	Revision	STM32CubeMX release number	Changes	
05-Sep-2018	26	4.27	Updated STM32Cube logo on cover page. Replaced STMCube [™] by STM32Cube [™] in the whole document. Updated <i>Section 1: STM32Cube overview</i> . Updated <i>Figure 1: Overview of STM32CubeMX C code generation</i> <i>flow</i> . Updated <i>Section 2.2: Key features</i> to add new features: graphic simulator feature, Support of embedded software packages in CMSIS-Pack format and Contextual Help. Changed <i>Section 3.4</i> title into "Getting updates using STM32CubeMX". Suppressed figures <i>Connection Parameters tab</i> - <i>No proxy</i> and <i>Connection Parameters tab</i> - <i>Use System proxy</i> <i>parameters</i> . Updated <i>Figure 22: Managing embedded software</i> <i>packages - Help menu</i> . In <i>Section 3.4.5: Installing embedded software packs</i> , updated step 3f of the embedded software pack installation sequence and added <i>Figure 27: License agreement acceptance</i> . <i>Section 4.2: New Project window:</i> updated <i>Figure 40: New Project</i> <i>window - MCU selector, Figure 41: Marking an MCU as favorite</i> and <i>Figure 42: New Project window - Board selector</i> . <i>Section 5.7.1: File menu:</i> added caution note for New Project in <i>Table 13: File menu functions.</i> Updated <i>Figure 107: Pinout menus</i> <i>(Pinout tab selected)</i> and <i>Figure 108: Pinout menus (Pinout tab not</i> <i>selected).</i> <i>Section 4.11: Project Manager view:</i> - Added note related to project saving (step 3). - Updated <i>Figure 184: Project Settings window</i> - Updated <i>Section 4.11: Project tab</i> and <i>Figure 189: Selecting a</i> <i>different firmware location.</i> Added <i>Section 1.15.4: Component dependencies panel, Contextual</i> <i>help, Section 10: Support of additional software components using</i> <i>CMSIS-Pack standard</i> and <i>Section 17: Tutorial 7 – Using the X-Cube-</i> <i>BLE1 software pack.</i>	
12-Nov-2018	27	4.28	Updated Section 3.4.3: Installing STM32 MCU packages, Section 3.4.5: Installing embedded software packs, Section 3.4.6: Removing already installed embedded software packages, Section 3.4.7: Checking for updates and the figures in it. Updated Section 4: STM32CubeMX user interface, its subsections and the figures and the tables in them. Updated Section 10: Support of additional software components using CMSIS-Pack standard, sections 11.6.1 to 11.6.5, Section 11.7.1: Setting project options, Section 11.7.2: Downloading firmware package and generating the C code, Section 11.8: Building and updating the C code project, Section 11.9: Switching to another MCU, Section 12: Tutorial 2 - Example of FatFs on an SD card using STM32429I-EVAL evaluation board and the figures in it, Section 15: Tutorial 5: Exporting current project configuration to a compatible MCU and the figures in it, Section 16: Tutorial 6 – Adding embedded software packs to user projects and Section 17: Tutorial 7 – Using the X-Cube-BLE1 software pack.	

Table 27. Document revision history (continued)



Date	Revision	STM32CubeMX release number	Changes		
12-Nov-2018	27 (conťď)	5.0	Added Section 19: Tutorial 10: Using ST-TouchGFX framework and its subsections. Updated Table 24: LL versus HAL: STM32CubeMX generated functions and function calls. Removed former Figure 164: Enabling and configuring a CMSIS- Pack software component, Figure 192: FatFs peripheral instances, Figure 213: Project Import status, Figure 254: Saving software component selection as user preferences and Figure 268: Configuring X-Cube-BLE1. Updated Figure 1: Overview of STM32CubeMX C code generation flow, Figure 16: STM32CubeMX perspective, Figure 32: Overall peripheral consumption, Figure 49: Opening Eclipse plug-in, Figure 10: STM32CubeMX perspective, Figure 32: Overall peripheral consumption, Figure 457: User constant generating define statements, Figure 480: Selecting a CMSIS-Pack software component, Figure 481: Enabling and configuring a CMSIS-Pack software component, Figure 482: Project generated with CMSIS- Pack software component, Figure 483: MCU selection, Figure 484: Pinout view with MCUs selection, Figure 485: Pinout view without MCUs selection window, Figure 487: Timer configuration, Figure 484: Simple pinout configuration, Figure 487: Pinout view without MCUs selection vindow, Figure 487: Pinout & Configuration view, Figure 491: Generate Project Report - New project creation, Figure 492: Clock tree view, Figure 497: Pinout & Configuration view, Figure 492: Clock tree view, Figure 501: Enabling Timer 3 interrupt, Figure 502: GPIO configuration color scheme and tooltip, Figure 503: GPIO mode configuration, Figure 504: DMA parameters configuration vindow, Figure 505: Middleware tooltip, Figure 506: USB Host configuration, Figure 504: DMA parameters, configuration window, Figure 505: Hist define statements, Figure 510: Project Settings and toolchain selection, Figure 511: Project Manager menu - Code Generator tab, Figure 512: Missing firmware package warming message, Figure 514: Updater settings for download, Figure 515: Updater settings with connection, Figure 516: Downloading the firmware packages, Figure 577:		

Table 27. Document revision history (continued)



Date	Revision	STM32CubeMX release number	Changes
19-Feb-2019	28	5.0	Updated Introduction, Section 1: STM32Cube overview, Section 2.2: Key features, Section 3.1.3: Software requirements, Section 3.4.3: Installing STM32 MCU packages, Section 4: STM32CubeMX user interface, Resolving pin conflicts, Section 4.5.10: Component configuration panel, Section 4.10: Clock Configuration view, Section 4.11: Project Manager view, Section 4.11.1: Project tab, Section 4.11.3: Advanced Settings tab, Using the transition checker, Section 9.2: STM32CubeMX Device tree generation, Section 6.3.2: Saving and selecting user templates, .extSettings file example and generated outcomes and Section 11.6.4: Configuring the DMAs. Added Section 4.6: Pinout & Configuration view for STM32 MPUs, Section 9: Device tree generation, Section 5: STM32CubeMX tools, Section 9: Device tree generation (STM32MPUs only), Section B.3.11: STM32WPAN BLE/Thread (STM32WB series only), Section B.3.13: OpenAmp and RESMGR_UTILITY (STM32MPUs and STM32H7 dual-core products) and their subsections. Removed former Section 1: General information. Updated Table 2: Home page shortcuts, Table 5: Component list, mode icons and color schemes, Table 6: Pinout menu and shortcuts and title of Table 9: Clock configuration view widgets. Updated Figure 184: Project Settings window, Figure 185: Project folder, Figure 189: Selecting a different firmware location, Figure 197: Automatic project import, Figure 198: Manual project import, Figure 201: Set unused pins window, Figure 478: STM32CubeMX generated DTS – Extract 3, Figure 480: Selecting a CMSIS-Pack software component, Figure 481: Enabling and configuring a CMSIS- Pack software component, Figure 481: Enabling and configuring a CMSIS- Pack software component, Figure 535: FATFS tutorial - Project settings and Figure 536: C code generation completion message.
16-Apr-2019	29	5.1	 Updated Introduction. Section 3.1.3: Software requirements, Section 4.2: New Project window, MCU close selector feature, External clock sources, Importing pinout, Selecting/deselecting all peripherals, Section 4.6: Pinout & Configuration view for STM32 MPUs, Section 4.15: Software Packs component selection window, Section 5.4.1: DDR configuration, Section 6.2: STM32Cube code generation using Low Layer drivers, BLE configuration and Section B.3.13: OpenAmp and RESMGR_UTILITY (STM32MPUs and STM32H7 dual-core products). Added Section 4.2.1: MCU selector, Section 4.2.2: Board selector, Section 4.2.4: Cross selector, Section 4.8: Pinout & Configuration view for STM32H7 dual-core products, Section 5.3.9: Example feature (STM32MPUs and STM32H7 dual-core only) and Section 7: Code generation for dual-core MCUs (STM32H7 dual-core product lines only). Removed former Section 3.3: Installing STM32CubeMX plug-in version and its subsections, and former Section 3.4.3: Running STM32CubeMX plug-in from Eclipse IDE.

Table 27. Document revision history (continued)



Date	Revision	STM32CubeMX release number	Changes
16-Apr-2019	29 (conťď)	5.1	Updated <i>Table 3: Window menu.</i> Updated figures 27 to 42, <i>Figure 195: Advanced Settings window</i> , figures 318 to 325, 327 to 330 and 332 to 341, <i>Figure 510: Project</i> <i>Settings and toolchain selection</i> and figures 538 to 548, Added <i>Figure 37: New Project window shortcuts, Figure 105:</i> <i>STM32MPUs: assignment options for GPIOs, Figure 643: Resource</i> <i>Manager: peripheral assignment view</i> and <i>Figure 645: STM32Cube</i> <i>Embedded Software package.</i>
01-Oct-2019	30	5.2	 Updated Introduction. Section 2.2: Key features, Section 3.3.2: Running STM32CubeMX in command-line mode, Part number selection, Section 4.15: Software Packs component selection window, Section 4.15.1: Introduction on software components, Section 4.15.2: Filter panel, Section 4.15.3: Packs panel, Section 4.15.4: Component dependencies panel, Section 4.15.6: Updating the tree view for additional software components, Section 5.3: Power Consumption Calculator view and Section 6.2: STM32Cube code generation using Low Layer drivers. Updated Table 1: Command line summary, Table 6: Pinout menu and shortcuts, Table 16: Additional Software window – Packs panel icons and Table 17: Component dependencies panel contextual help. Updated Figure 33: STM32CubeMX home page, Figure 208: Selection of additional software components, Figure 209: Additional software components - Updated tree view, Figure 480: Selecting a CMSIS-Pack software component and Figure 578: Selecting X-Cube- BLE1 components. Added Section 4.5.8: Pinout for multi-bonding packages and Section 4.15.5: Details and Warnings panel. Added Table 15: Additional Software window – Packs panel columns

Table 27. Document revision history (continued)



Date	Revision	STM32CubeMX release number	Changes
Date 13-Dec-2019	Revision		Changes Updated Introduction, Section 1: STM32Cube overview, Section 4.2: New Project window, MCU/MPU selection for a new project and Section 11.7.1: Setting project options. Added Section 4.9: Enabling security in Pinout & Configuration view (STM32L5 and STM32U5 series only) with its subsections, Section 4.10.2: Securing clock resources (STM32L5 series only) and Section 8: Code generation with TrustZone [®] enabled (STM32L5 series only). Removed former Section 4.4.16: Graphics frameworks and simulator, Section 17: Tutorial 8 – Using STemWin Graphics framework, Section 18: Tutorial 9: Using STM32CubeMX Graphics simulator, Section 19: Tutorial 10: Using ST-TouchGFX framework and Section B.3.11: Graphics. Minor text edits across the whole document. Updated Table 1: Command line summary. Updated Figure 68: Pinout view: MCUs with multi-bonding, Figure 69: Pinout view: multi-bonding with extended mode, Figure 105: STM32MPUs: assignment options for GPIOs, Figure 184: Project Settings window, Figure 349: DDR Suite - Connection to target, Figure 350: DDR Suite - Target connected, Figure 351: DDR activity logs, Figure 354: DDR interactive logs, Figure 353: DDR register loading, Figure 354: DDR test list from U-Boot SPL, Figure 355: DDR test suite results, Figure 356: DDR tests history, Figure 175: DDR tuning pre-requisites, Figure 176: DDR tuning process, Figure 177: Bit deskew, Figure 178: Eye training (centering) panel, Figure 179: DDR Tuning - saving to configuration, Figure 475: Project settings for STM32CubeIDE toolchain and Figure 510: Project Settings and
			toolchain selection. Added Figure 38: Enabling TrustZone.

Table 27. D	Document	revision	history	(continued)



Date	Revision	STM32CubeMX release number	Changes
10-Jul-2020	32	6.0	Updated Section 2.2: Key features, Section 3.1.1: Supported operating systems and architectures, Section 3.1.3: Software requirements, Section 3.2.1: Installing STM32CubeMX standalone version, Section 3.4: Getting updates using STM32CubeMX, Section 3.4.5: Installing embedded software packs, Section 4.2: New Project window, Export to Excel, Section 4.5: Pinout & Configuration view, Section 4.11.3: Advanced Settings tab and Section 18.6: Why do I get the error "Java 8 update 45" when installing "Java 8 update 45" or a more recent version of the JRE?. Added Section 4.2.3: Example selector, Section 5.1: External Tools, Section 19.2: Since I changed my login to access the Internet, some software packs appear not available. and Section 19.3: On dual- context products, why some peripherals or middleware are not available for a given context?. Removed former MCU selection based on graphics criteria. Updated Table 4: Help menu shortcuts and Table 14: Additional software window - Filter icons. Updated Figure 33: STM32CubeMX home page, Figure 37: New Project window shortcuts, Figure 42: New Project window - Board selector, Figure 45: Cross selector - Data refresh prerequisite, Figure 195: Advanced Settings window, Figure 205: Additional software window, Figure 200: Device tree generation for the Linux kernel, Figure 201: STM32CubeMX Device tree generation for TF-A, Figure 578: Selecting X-Cube-BLE1 components and Figure 306: Java Control Panel.
10-Nov-2020	33	6.1	Updated Introduction, Section 3.1.3: Software requirements, Section 3.4.7: Checking for updates, Section 4.15.3: Packs panel, Section 5.1: External Tools, Section 12: Tutorial 2 - Example of FatFs on an SD card using STM32429I-EVAL evaluation board and Section 18.6: Why do I get the error "Java 8 update 45" when installing "Java 8 update 45" or a more recent version of the JRE?. Added Choosing not to generate code for some peripherals or middlewares. Updated Table 1: Command line summary. Updated Figure 32: Help menu: checking for updates, Figure 33: STM32CubeMX home page, Figure 195: Advanced Settings window, Figure 205: Additional software window, Figure 300: ST Tools and Figure 531: SDIO peripheral configuration.

Table 27. Document revision history (continued)



Date	Revision	STM32CubeMX release number	Changes	
12-Feb-2021	34	6.2	Updated Section 3.1.1: Supported operating systems and architectures, Section 3.1.3: Software requirements, Section 3.2.1: Installing STM32CubeMX standalone version, Section 3.2.2: Installing STM32CubeMX from command line, Section 3.2.3: Uninstalling STM32CubeMX standalone version, Section 3.3.2: Running STM32CubeMX in command-line mode, Warning: in Section 3.4.7: Checking for updates, Section 4.1: Home page, Section 4.15: Software Packs component selection window, Section 4.15.2: Filter panel, Section 4.15.3: Packs panel, Section 4.15.4: Component dependencies panel, Section 4.15.5: Details and Warnings panel and Section 12: Tutorial 2 - Example of FatFs on an SD card using STM32429I-EVAL evaluation board.	
			Updated <i>Table 6: Pinout menu and shortcuts</i> . Added <i>Figure 2: Full disk access for macOS</i> and <i>Figure 206:</i>	
			Component dependency resolution. Updated Figure 33: STM32CubeMX home page, Figure 38: Enabling TrustZone, Figure 205: Additional software window.	
			Removed former <i>Figure 5: Auto-install command line</i> and former Section 18.6: Why do I get the error "Java 8 update 45" when installing "Java 8 update 45" or a more recent version of the JRE?.	
	35	5 6.3	Updated Section 3.1.1: Supported operating systems and architectures, Section 3.1.3: Software requirements, Section 4.2: New Project window, Section 4.3: Project page, Section 4.5.5: Pinout view advanced actions, Section 4.9: Enabling security in Pinout & Configuration view (STM32L5 and STM32U5 series only) and code in Section 12: Tutorial 2 - Example of FatFs on an SD card using STM32429I-EVAL evaluation board.	
22-Jun-2021			Added Figure 39: Adjusting selector results and Section 19.1: I encountered a network connection error during a download from STM32CubeMX.	
			Updated Table 1: Command line summary, Table 16: Additional Software window – Packs panel icons and Table 17: Component dependencies panel contextual help. Updated Figure 480: Selecting a CMSIS-Pack software component and Figure 578: Selecting X-Cube-BLE1 components.	
05-Nov-2021	36	6.4	Updated Section 2.2: Key features, Section 3.3.1: Running STM32CubeMX as a standalone application, Section 3.4: Getting updates using STM32CubeMX, Section 4.2: New Project window, Enabling interruptions using the NVIC tab view, Section 4.9: Enabling security in Pinout & Configuration view (STM32L5 and STM32U5 series only), Section 4.11.1: Project tab and Section 5.3.7: Bluetooth Low-Energy [®] /ZigBee [®] support (STM32WB series only). Added Section 3.4.1: Running STM32CubeMX behind a proxy server and Section 5.3.8: Sub-GHz support (STM32WL series only).	
			Updated Figure 89: NVIC configuration tab - FreeRTOS disabled.	

Table 27. Document revision history (continued)



Date	Revision	STM32CubeMX release number	Changes
			Updated Introduction and Section 3.1.1: Supported operating systems and architectures.
18-Feb-2022	37	6.5	Added Section 18: Creating LPBAM projects with its subsections, and Section 19.11: How to fix MX_DMA_Init call rank in STM32CubeMX generated projects?.
			Minor text edits across the whole document.
			Updated Introduction, Section 2.2: Key features, Section 3.3.2: Running STM32CubeMX in command-line mode, Boot loader (A7 FSBL) peripherals selection, Section 4.11.1: Project tab, Section 4.16: LPBAM Scenario & Configuration view, Section 9.1: Device tree overview, and Section 9.2: STM32CubeMX Device tree generation.
14-Jun-2022	38	6.6	Updated Table 1: Command line summary.
14-Juli-2022	50	0.0	Updated Figure 299: About window.
			Added Section 4.17: CAD Resources view and Section 18.6: LPBAM application for TrustZone [®] activated projects.
			Removed former Section 9.2.1: Device tree generation for Linux kernel, Section 9.2.2: Device tree generation for U-boot, and Section 9.2.3: Device tree generation for TF-A.
			Minor text edits across the whole document.
		39 6.7	Updated Section 2.2: Key features and Section 17: Tutorial 7 – Using the X-Cube-BLE1 software pack.
47.11. 0000			Added Section 19.12: When is the PeriphCommonClock_Config() function generated? and Section 19.13: How to handle thread-safe solution in STM32CubeMX and STM32CubeIDE?.
17-Nov-2022	39		Updated Figure 40: New Project window - MCU selector, Figure 41: Marking an MCU as favorite, Figure 29: New Project window - MCU list with close function, Figure 30: New Project window - List showing close MCUs, and Figure 299: About window.
			Minor text edits across the whole document.
		40 6.8	Updated Section 3.2.1: Installing STM32CubeMX standalone version, Section 3.3.2: Running STM32CubeMX in command-line mode, Section 3.4.1: Running STM32CubeMX behind a proxy server, and Section 4.11.1: Project tab.
04 5-6 0000	40		Added Section 4.18: Boot path and its subsections.
21-Feb-2023	40		Removed former Section 5.3.4: DDR tuning and DDR tuning tab (read-only).
			Updated Figure 40: New Project window - MCU selector, Figure 184: Project Settings window, and Figure 606: Design check.
			Minor text edits across the whole document.

Table 27. Document revision history (continued)



Date	Revision	STM32CubeMX release number	Changes	
03-Jul-2023	41	6.9	 Updated Introduction, Section 3.1.1: Supported operating systems and architectures, Java™ Runtime Environment, Section 4.15: Software Packs component selection window, Section 4.18: Boot path, Section 4.18.2: Creating a boot path project: an example, Section 4.18.5: How to configure an ST-iRoT with a secure manager NS application boot path, and note in Section 18.4: Checking the LPBAM design. Updated Table 1: Command line summary. Added note to Section 9.2: STM32CubeMX Device tree generation. Added figures 221 to 225 and Figure 272: Code generated with secure manager API. Added Section 4.18.5: How to configure an assembled boot path, Section 4.19: User authentication, Section 4.18: STM32CubeMX Memory Management Tool and their subsections, and Section B.3.12: CMSIS packs selection limitation. Updated Figure 45: Cross selector - Data refresh prerequisite, Figure 217: Boot paths for STM32H57x devices, Figure 227: Select the STM32H5 device, Figure 229: Boot paths for STM32H56x devices, figures 232 to 245, figures 167 to 256, figures 258 to 260, figures 263 to 267, Figure 271: Secure manager API configuration, and Figure 299: About window. Minor text edits across the whole document. 	
08-Sep-2023	42	6.9.2	 Updated for the replacement of "boot path settings" with "boot path and debug authentication" in Section 4.18.4: How to configure an ST-iRoT boot path Section 4.18.5: How to configure an ST-iRoT with a secure manager NS application boot path Figure 241, Figure 255, and Figure 266 titles Updated Figure 266: Boot path and Debug Authentication tab. Updated figures 217 to 225 in Section 4.18.1: Available boot paths. Updated Section 1: STM32Cube overview. Minor text edits across the whole document. 	

Table 27. Document revision history (continued)



Date	Revision	STM32CubeMX release number	Changes
20-Nov-2023	43	6.10.0	Updated Section 4.11: Project Manager view, Section 4.18.5: How to configure an ST-iRoT with a secure manager NS application boot path, Step 3: OEMiROT (assembled) code generation, Step 6: Authentication and encryption keys regeneration, option byte file generation, and Section 4.18: STM32CubeMX Memory Management Tool. Added Section 4.19.3: Password restoration. Removed former MCU close selector feature. Updated Table 18: Boot paths without TrustZone [®] (TZEN = 0) and Table 19: Boot paths with TrustZone [®] (TZEN = 1). Updated Figure 221: Application boot path (OEM-uRoT assembled), Figure 222: Application boot path: ST-iRoT and uRoT secure/nonsecure project, Figure 224: Application boot path: ST-iRoT dual figure, Figure 239: Project provisioning, Figure 241: Boot path and debug authentication panel, Figure 248: IDE post build commands, Figure 354: DDR test list from U-Boot SPL. Removed former Figure 167: Selection of the OEMiRoT_Boot project and Figure 195: Generated project. Minor text edits across the whole document.
13-Mar-2024	44	6.11.0	Updated Section 3.1.1: Supported operating systems and architectures, Section 3.2.1: Installing STM32CubeMX standalone version, Section 3.2.2: Installing STM32CubeMX from command line, Uninstalling STM32CubeMX on Windows, Feature: MMT usage, Pinout, and Configuration UI, and Section 4.18.5: How to configure an assembled boot path. Added footnote to Table 1: Command line summary. Updated Table 10: Clock Configuration security settings, Table 18: Boot paths without TrustZone [®] (TZEN = 0), and Table 19: Boot paths with TrustZone [®] (TZEN = 1). Added Section 4.18.6: How to configure OEM-uRoT (STiRot uROT) boot path, When using H7Rx/H7Sx with MMT, When using H7Rx/H7Sx, and their subsections. Added Figure 236: MMT view (H7Rx-H7Sx devices) and Figure 255: Memory assignment for context Boot H7RS. Updated Figure 9: Package installation, Figure 10: Installation script, Figure 11: Installation path, Figure 222: Application boot path: ST- iRoT and uRoT secure/nonsecure project, Figure 224: Application boot path: ST-iRoT dual figure, Figure 234: Boot path selection, Figure 241: Boot path and debug authentication panel, Figure 245: Generate the code, Figure 255: Boot path and Debug Authentication tab, Figure 275: Boot path and Debug Authentication tab, and Figure 275: Boot path project. Minor text edits across the whole document.



Date	Revision	STM32CubeMX release number	Changes
26-Jun-2024	45	6.12.0	 Updated Section 2.2: Key features, Java™ Runtime Environment, Section 3.4.7: Checking for updates, Step 5: Boot path selection, Section 4.6: Pinout & Configuration view for STM32 MPUs, Section 4.18.6: How to configure OEM-uRoT (STiRot uROT) boot path, Section 4.19: User authentication, Section 4.19.1: Login with an existing my.st.com account, and Section 8: Code generation with TrustZone[®] enabled (STM32L5 series only). Added note to Section 3.4.2: Updater configuration. Added Section 4.4: Boot chain (STM32 MPUs), Section 4.7: RIF configuration, Section 4.18.7: How to configure ST-iRoT boot path with STM32H7RS devices, Section 5.5: STM32CubeMX Memory Management Tool, and their subsections. Updated Table 1: Command line summary and Table 19: Boot paths with TrustZone[®] (TZEN = 1). Added Figure 20: Connection failure and Figure 31: Checking for available updates. Updated Figure 44: Popup window - Starting a project from an example, Figure 269: Project creation, Figure 510: Project Settings and toolchain selection, and Figure 592: Available IPs. Removed former Section 4.18: STM32CubeMX Memory Management Tool, Section 19: FAQ, and their subsections. Minor text edits across the whole document.
20-Nov-2024	46	6.13.0	Updated Section 3.1.1: Supported operating systems and architectures, Section 3.1.3: Software requirements, Section 4.7.2: <i>RIF global configurations, Section 4.7.4: Peripheral instance</i> protection, Section 4.7.7: Masters configuration, Section 4.7.9: System peripherals (STM32MP2 and STM32N6 series), Section 4.7.10: Memory protection for STM32MP2 series, Section 4.19: User authentication, and Section 5.5.1: STM32H5, STM32U5, STM32WBA, STM32WBAM, and STM32WBA6 with TrustZone activated Added Section 4.7.10: Memory protection for STM32NP2 series, Section 4.7.11: Memory protection for STM32NP2 series, Section 4.7.13: Implementation of illegal access controller (IAC) feature on STM32N6 series, Section 5.2: Compare Projects, and Section 5.5.5: STM32H7 Dual-core without Trust Zone activated

Table 27. Document revision history (continued)



Revision history

Date	Revision	STM32CubeMX release number	Changes
24-Feb-2025	47	6.14.0	Updated Introduction, Section 3.1.1: Supported operating systems and architectures, Section 4.7.2: RIF global configurations, Configuration example, Masters configurations for STM32MP2, Step 8: Code compilation, Step 2: Code compilation, Section 5.2.1: User interface of the Compare Projects tool, Section 5.2.2: Comparing two projects, and Section 5.5.1: STM32H5, STM32U5, STM32WBA, STM32WBAM, and STM32WBA6 with TrustZone activated. Updated figures 20 to 31, Figure 33: STM32CubeMX home page, figures 39 to 42, Figure 45: Cross selector - Data refresh prerequisite, Figure 115: Lock and privilege in RISUP table, Figure 217: Boot paths for STM32H57x devices, Figure 220: Application boot path (OEM- iRoT), Figure 232: Configuring the project, Figure 233: Saving the project, Figure 241: Boot path and debug authentication panel, Figure 245: Generate the code, Figure 256: Select the project structure, figures 269 to 271, Figure 274: Boot path and debug authentication tab, Figure 288: Home page without the login form, Figure 289: Install or remove a software package, Figure 291: About window, figures 302 to 303, 305 to 311, 397 to 398, 405 to 409, 414 to 416, 418 to 419, Figure 444: ETH configuration for STM32H7R3A8Ix, and Figure 483: MCU selection. Added Figure 109: RIF configuration extension in IPs panel for the STM32H57x devices, Figure 220: Application boot path (OEM-iRoT), Figure 316: Comparison result in Excel format - Peripherals and middleware, and Figure 317: Comparison result in Excel format - Project settings. Updated Table 3: Window menu and Table 19: Boot paths with TrustZone [®] (TZEN = 1). Removed former Figure 31: Library deletion progress window and Figure 275: Boot path project. Removed former Section 4.18.5: How to configure an ST-iRoT with a secure manager NS application boot path. Added ETH impact on MMT for STM32H7 single core and ETH impact on MMT when using H7RS/H7SX. Minor text edits across the whole document.

Table 27. Document revision history (continued)



IMPORTANT NOTICE - READ CAREFULLY

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgment.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2025 STMicroelectronics - All rights reserved



UM1718 Rev 47