

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*), 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), STM32L0 (UM1749), STM32L1 (UM1816), STM32L4/L4+ (UM1884), STM32L5 (UM2659), STM32MP1 (https://wiki.st.com/stm32mpu), STM32U5 (UM2883), STM32WL (UM2642), STM32WB (UM2442), and STM32WBA (UM3131).





November 2024

UM1718 Rev 46

1/557

Contents

1	STM3	2Cube	overview	25
2	Gettir	ng starte	ed with STM32CubeMX	26
	2.1	Principle	es	26
	2.2	Key feat	tures	28
	2.3	Rules a	nd limitations	30
3	Instal	ling and	d running STM32CubeMX	31
	3.1	System	requirements	31
		3.1.1	Supported operating systems and architectures	. 31
		3.1.2	Memory prerequisites	. 32
		3.1.3	Software requirements	. 32
	3.2	Installing	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	ng 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	. 53
		3.4.4	Installing STM32 MCU package patches	. 54
		3.4.5	Installing embedded software packs	. 54
		3.4.6	Removing already installed embedded software packages	. 59
		3.4.7	Checking for updates	. 61
4	STM3	2Cubel	/IX user interface	63
	4.1	Home p	age	64
		4.1.1	File menu	. 65
		4.1.2	Window menu and Outputs tabs	. 66
		4.1.3	Help menu	. 68
		4.1.4	Social links	. 68



4.2	New P	roject window	9
	4.2.1	MCU selector	0
	4.2.2	Board selector	2
	4.2.3	Example selector	3
	4.2.4	Cross selector	4
4.3	Project	t page	7
4.4	Boot ch	hain (STM32MPUs)	0
	4.4.1	Boot mode configuration	0
	4.4.2	Coprocessor initializers (STM32MP2x)8	3
	4.4.3	Boot device selection (STM32MP25)8	4
4.5	Pinout	& Configuration view	5
	4.5.1	Component list	6
	4.5.2	Component Mode panel	8
	4.5.3	Pinout view	9
	4.5.4	Pinout menu and shortcuts9	0
	4.5.5	Pinout view advanced actions9	2
	4.5.6	Keep Current Signals Placement	3
	4.5.7	Pinning and labeling signals on pins9	4
	4.5.8	Pinout for multi-bonding packages9	5
	4.5.9	System view	17
	4.5.10	Component configuration panel9	8
	4.5.11	User Constants configuration window10	2
	4.5.12	GPIO configuration window10	6
	4.5.13	DMA configuration window10	8
	4.5.14	NVIC configuration window 11	0
	4.5.15	FreeRTOS configuration panel 11	6
	4.5.16	Setting HAL timebase source12	2
4.6	Pinout	& Configuration view for STM32MPUs 12	6
	4.6.1	Run time configuration	27
	4.6.2	Boot stages configuration12	27
4.7	RIF co	nfiguration	8
	4.7.1	Configuration approach12	8
	4.7.2	RIF global configurations12	9
	4.7.3	Peripherals protection13	0
	4.7.4	Peripheral instance protection13	0
	4.7.5	IP feature protection	5



	4.7.6	Software constraints validation	
	4.7.7	Masters configuration	
	4.7.8	Service peripherals protection	
	4.7.9	System peripherals (STM32MP2 and STM32N6 series)	
	4.7.10	Memories protection for STM32MP2 series	154
	4.7.11	Memories protection for STM32N6 series	
	4.7.12	RIF code generation	165
	4.7.13	Implementation of illegal access controller (IAC) feature on STM32N6 series	167
4.8	Pinout &	& Configuration view for STM32H7 dual-core products	167
4.9		g security in Pinout & Configuration view 2L5 and STM32U5 series only)	169
	4.9.1	Privilege access for peripherals, GPIO EXTIs and DMA requests	169
	4.9.2	Secure/nonsecure context assignment for GPIO/peripherals/middleware	. 173
	4.9.3	NVIC and context assignment for peripherals interrupts	173
	4.9.4	DMA (context assignment and privilege access settings)	173
	4.9.5	GTZC	. 175
	4.9.6	OTFDEC	. 176
4.10	Clock C	Configuration view	177
	4.10.1	Clock tree configuration functions	178
	4.10.2	Securing clock resources (STM32L5 series only)	181
	4.10.3	Recommendations	184
	4.10.4	STM32F43x/42x power overdrive feature	185
	4.10.5	Clock tree glossary	186
4.11	Project	Manager view	187
	4.11.1	Project tab	188
	4.11.2	Code Generator tab	193
	4.11.3	Advanced Settings tab	196
4.12	Import F	Project window	198
4.13	Set unu	sed/reset used GPIOs windows	203
4.14	Update	Manager windows	205
4.15	-	e Packs component selection window	
	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	211
	4.15.6	Updating the tree view for additional software components	212
4.16	LPBAN	/ Scenario & Configuration view	213
4.17	CAD R	Resources view section	214
4.18	Boot pa	ath	218
	4.18.1	Available boot paths	218
	4.18.2	Creating a boot path project: an example	223
	4.18.3	How to configure an OEM-iRoT boot path	223
	4.18.4	How to configure an ST-iRoT boot path	237
	4.18.5	How to configure an ST-iRoT with a secure manager NS application boot path	243
	4.18.6	How to configure an assembled boot path	250
	4.18.7	How to configure OEM-uRoT (STiRot uROT) boot path	255
	4.18.8	How to configure ST-iRoT boot path with STM32H7RS devices 2	260
4.19	User a	uthentication	264
	4.19.1	Login with an existing my.st.com account	264
	4.19.2	Create a my.st.com account	269
	4.19.3	Password restoration	270
	4.19.4	Authentication through command line interface	272
4.20	About	window	273
STM	32Cube	MX tools	274
5.1	Externa	al Tools	274
5.2	Compa	are Projects	275
	5.2.1	User interface of the Compare Projects tool	275
	5.2.2	Steps of comparing two projects	276
	5.2.3	The output of the comparison of 2 projects	278
	5.2.4	Saving the comparison result of the two projects	281
5.3	Power	Consumption Calculator view	283
	5.3.1	Building a power consumption sequence	284
	5.3.2	Configuring a step in the power sequence	288
	5.3.3	Managing user-defined power sequence and reviewing results	291
	5.3.4	Power sequence step parameters glossary	294
	5.3.5	Battery glossary	296
	5.3.6	SMPS feature	296
	5.3.7	Bluetooth Low-Energy [®] /ZigBee [®] support (STM32WB series only)	302



5

		5.3.8	Sub-GHz support (STM32WL series only)			
		5.3.9	Example feature (STM32MPUs and STM32H7 dual-core only)	304		
	5.4	DDR S	Suite (STM32MPUs only)	306		
		5.4.1	DDR configuration	307		
		5.4.2	Connection to the target and DDR register loading	309		
		5.4.3	DDR testing	311		
	5.5	STM3	2CubeMX Memory Management Tool	313		
		5.5.1	STM32U5, STM32H5, STM32WBA, and STM32WBAM with TrustZone activated	314		
		5.5.2	An end-to-end usage example	316		
		5.5.3	STM32H7 single core and STM32U5 without TrustZone activated	328		
		5.5.4	STM32WBxx	334		
		5.5.5	STM32H7 Dual-core without Trust Zone activated	336		
		5.5.6	STM32H7RS	349		
		5.5.7	STM32WB0	361		
		5.5.8	Notification MMT/boot path (STM32H7RS and STM32H5)	363		
6	STM	STM32CubeMX C Code generation overview				
	6.1		2Cube code generation using only HAL drivers It mode)	366		
	6.2	•	2Cube code generation using Low Layer drivers			
	6.3		n 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		onal settings for C project generation			
-	Code		ation for dual care MOUs			
7			ation for dual-core MCUs lual-core product lines only)	381		
8	Code	e gener	ation with TrustZone [®] enabled (STM32L5 series only)	. 383		
9	Devi	ce tree	generation (STM32MPUs only)	387		
	9.1	Device	e tree overview	387		
	9.2	STM3	2CubeMX Device tree generation	389		
10			additional software components using	391		
6/557			UM1718 Rev 46			
0/00/						

11		Tutorial 1: From pinout to project C code generationusing an MCU of the STM32F4 series				
	11.1	Creating a new STM32CubeMX Project				
	11.2	Configuring the MCU pinout 396				
	11.3	Saving the project				
	11.4	Generating the report 400				
	11.5	Configuring the MCU clock tree 400				
	11.6	Configuring the MCU initialization parameters				
		11.6.1 Initial conditions				
		11.6.2 Configuring the peripherals				
		11.6.3 Configuring the GPIOs 406				
		11.6.4 Configuring the DMAs				
		11.6.5 Configuring the middleware				
	11.7	Generating a complete C project 412				
		11.7.1 Setting project options				
		11.7.2 Downloading firmware package and generating the C code				
	11.8	Building and updating the C code project 418				
	11.9	Switching to another MCU 423				
12		rial 2 - Example of FatFs on an SD card using 32429I-EVAL evaluation board				
13		Tutorial 3 - Using the Power Consumption Calculator to optimize the embedded application consumption and more 432				
	13.1	Tutorial overview				
	13.2	Application example description				
	13.3	Using the Power Consumption Calculator				
		13.3.1 Creating a power sequence				
		13.3.2 Optimizing application power consumption				
14		rial 4 - Example of UART communications with FM32L053xx 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				
	17.0					



	14.4	Configu	ring the MCU clock tree from the Clock Configuration view 446
	14.5	Configu	ring the peripheral parameters from the Configuration view 447
	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	•	n 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 458
17	Tutor	rial 7 – L	Ising the X-Cube-BLE1 software pack
18	Creat	ting LPE	3AM projects 470
	18.1	LPBAM	overview
		18.1.1	LPBAM operating mode
		18.1.2	LPBAM firmware
		18.1.3	Supported series
		18.1.4	LPBAM design
		18.1.5	LPBAM project support in STM32CubeMX471
	18.2	Creating	g an LPBAM project
		18.2.1	LPBAM feature availability
		18.2.2	Describing an LPBAM project
		18.2.3	Managing LPBAM applications in a project
	18.3	Describ	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 480
		18.3.8	Queue description: configuring the DMA channel hosting the queue . 481
		18.3.9	Node description: accessing contextual help and documentation 482
		18.3.10	Node description: configuring node parameters



		18.3.11	Node description: configuring a trigger	484
		18.3.12	Node description: reconfiguring a DMA for Data transfer	485
1	18.4	Checkir	ng the LPBAM design	487
1	18.5	Genera	ting a project with LPBAM applications	488
1	18.6	LPBAM	application for TrustZone [®] activated projects \ldots	489
Appendix	A S	TM32Cı	ıbeMX pin assignment rules	493
ŀ	۹.1	Block co	onsistency	493
ŀ	۹.2	Block in	ter-dependency	497
A	۹.3	One blo	ck = one peripheral mode	499
ŀ	۹.4	Block re	emapping (STM32F10x only)	499
ŀ	۹.5		n remapping.	
ŀ	۹.6		hifting (only for STM32F10x and when Current Signals placement" is unchecked)	501
ŀ	۹.7	Setting	and clearing a peripheral mode	502
ŀ	۹.8	Mapping	g a function individually	502
ŀ	۹.9	GPIO si	ignals mapping	502
Appendix	в с [.]	TM22CI	ubeMX C code generation design	
Appendix			and limitations	503
E	3.1	STM32	CubeMX generated C code and user sections	503
E	3.2	STM32	CubeMX design choices for peripheral initialization	503
E	3.3		CubeMX design choices and limitations for	
			vare initialization	504
		B.3.1	Overview	504
		B.3.2	USB host	505
		B.3.3	USB device	505
		B.3.4	FatFs	505
		B.3.5	FreeRTOS.	506
		B.3.6	LwIP	507
		B.3.7	Libjpeg	509
		B.3.8	Mbed TLS	510
		B.3.9	TouchSensing	513
		B.3.10	PDM2PCM	516
		B.3.11	STM32WPAN BLE/Thread (STM32WB series only)	517
		B.3.12	CMSIS packs selection limitation	521



		B.3.13	OpenAmp and RESMGR_UTILITY (STM32MPUs and STM32H7 dual-core products)
Appendix	C S	Г M 32 m	icrocontrollers naming conventions
Appendix	D S	Г М 32 m	icrocontrollers power consumption parameters 527
	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	65
Table 3.	Window menu	66
Table 4.	Help menu shortcuts	68
Table 5.	Component list, mode icons and color schemes	87
Table 6.	Pinout menu and shortcuts	90
Table 7.	Configuration states	
Table 8.	Peripheral and Middleware configuration window buttons and tooltips	
Table 9.	Clock configuration view widgets	
Table 10.	Clock Configuration security settings	
Table 11.	Voltage scaling versus power overdrive and HCLK frequency	
Table 12.	Relations between power over-drive and HCLK frequency	. 186
Table 13.	Glossary	
Table 14.	Additional software window - Filter icons	
Table 15.	Additional Software window – Packs panel columns	
Table 16.	Additional Software window – Packs panel icons	
Table 17.	Component dependencies panel contextual help	
Table 18.	Boot paths without TrustZone [®] (TZEN = 0)	
Table 19.	Boot paths with TrustZone [®] (TZEN = 1)	
Table 20.	Boot paths for STM32H7RS devices	
Table 21.	LL versus HAL code generation: drivers included in STM32CubeMX projects	
Table 22.	LL versus HAL code generation: STM32CubeMX generated header files	
Table 23.	LL versus HAL: STM32CubeMX generated source files	
Table 24.	LL versus HAL: STM32CubeMX generated functions and function calls	
Table 25.	Files generated when TrustZone [®] is enabled.	
Table 26.	Connection with hardware resources	
Table 27.	Document revision history	. 533



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	
Figure 13.	Package installation	
Figure 14.	Installation completed	
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.		
Figure 28.	Embedded software pack release - Successful installation	
Figure 29.	Removing libraries	
Figure 30.	Removing library confirmation message.	
Figure 31.	Library deletion progress window	
Figure 32.	Updates are available	
Figure 33.	Help menu: checking for updates	
Figure 34.	STM32CubeMX Home page.	
Figure 35.	Window menu	
Figure 36.	Output view	
Figure 37.	Link to social platforms	
Figure 38.	New Project window shortcuts	
Figure 39.	Enabling TrustZone	
Figure 40.	Adjusting selector results	
•	New Project window - MCU selector.	
Figure 41.	•	
Figure 42.	Marking a favorite	. / 1
Figure 43.		
Figure 44.	New project window - Example selector	
Figure 45.	Popup window - Starting a project from an example	
Figure 46.	Cross selector - Data refresh prerequisite	
Figure 47.	Cross selector - Part number selection per vendor.	
Figure 48.	Cross selector - Partial part number selection completion	. 75



Figure 49.	Cross selector - Compare cart	6
Figure 50.	Cross selector - Part number selection for a new project	
Figure 51.	STM32CubeMX Main window upon MCU selection	
Figure 52.	STM32CubeMX Main window upon board selection (peripherals not initialized)79	
Figure 53.	STM32CubeMX Main window upon board selection	-
i igui e ee.	(peripherals initialized with default configuration)	0
Figure 54.	Project choice interface	
Figure 55.	Contexts	1
Figure 56.	IPs interface assignment	2
Figure 57.	TrustZone option	2
Figure 58.	Selected context	3
Figure 59.	Assign IP context	3
Figure 60.	OP-TEE selected	4
Figure 61.	U-Boot selection	4
Figure 62.	FSBL synchronization output	5
Figure 63.	Contextual Help window (default)	6
Figure 64.	Contextual Help detailed information	7
Figure 65.	Pinout view	
Figure 66.	Modifying pin assignments from the Pinout view	2
Figure 67.	Example of remapping in case of block of pins consistency	3
Figure 68.	Pins/Signals Options window	
Figure 69.	Pinout view: MCUs with multi-bonding	6
Figure 70.	Pinout view: multi-bonding with extended mode	6
Figure 71.	System view	
Figure 72.	Configuration window tabs (GPIO, DMA, and NVIC settings for STM32F4 series)98	8
Figure 73.	Peripheral mode and Configuration view	9
Figure 74.	Formula when input parameter is set in No Check mode	1
Figure 75.	User Constants tab	2
Figure 76.	Extract of the generated main.h file 102	2
Figure 77.	Using constants for peripheral parameter settings 10	3
Figure 78.	Specifying user constant value and name 104	4
Figure 79.	Deleting an user constant is not allowed when it	
	is already used for another constant definition 104	4
Figure 80.	Confirmation request to delete a constant for parameter configuration	4
Figure 81.	Consequence when deleting a user constant for peripheral configuration 10	5
Figure 82.	Searching for a name in a user constant list 10	5
Figure 83.	Searching for a value in a user constant list	5
Figure 84.	GPIO configuration window - GPIO selection	6
Figure 85.	GPIO configuration grouped by peripheral 10	7
Figure 86.	Multiple pins configuration	7
Figure 87.	Adding a new DMA request	
Figure 88.	DMA configuration	
Figure 89.	DMA MemToMem configuration	0
Figure 90.	NVIC configuration tab - FreeRTOS disabled 11	1
Figure 91.	NVIC configuration tab - FreeRTOS enabled 112	2
Figure 92.	I2C NVIC configuration window	2
Figure 93.	NVIC Code generation – All interrupts enabled	
Figure 94.	NVIC Code generation – IRQ Handler generation	6
Figure 95.	FreeRTOS configuration view	
Figure 96.	FreeRTOS: configuring tasks and queues 118	
Figure 97.	FreeRTOS: creating a new task	
Figure 98.	FreeRTOS - Configuring timers, mutexes and semaphores	0



		404
	FreeRTOS heap usage	
•	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	
Figure 105.	STM32MPUs boot devices and runtime contexts	126
Figure 106.	STM32MPUs: assignment options for GPIOs	126
Figure 107.	Select peripherals as boot devices	127
Figure 108.	Default configuration.	129
Figure 109.	Default configuration for the STM32MP2 series	129
•	RISUP configuration panel	
	Software context configuration vs. RISUP configuration	
	Example of IP assignment to one context and result in RISUP	
	Example of IP assignment to two contexts and result in RISUP.	
	Lock and privilege in RISUP table	
	Pseudo RIF-aware IP assignment	
	Peripherals (RISUP) panel for the STM32N6 series.	
	Creation of a new project for the STM32N6 series - Secure projects.	
	Peripherals (RISUP) panel for the STM32N6 series - Secure projects	
	FMC configuration	
0	RIF FMC panel	
	RTC features	
0	RTC mode	
	RTC parameters setting	
	Color coding system and instructions	
Figure 125.	RIMU user interface	140
	Assigning a CID to an IP in RIMU.	
Figure 127.	Modification of the security and privilege values	141
	IP assignment to a context	
Figure 129.	Result in RISUP of an IP assignment to a context	142
	Inheritance of CID, state of security, and privilege from RISUP	
	Default values for IPs and user modification restrictions	
	Domains (RIMU) panel for STM32N6 series	
	RIF HSEM panel	
	RIF TAMP panel (STM32MP2 devices)	
	RIF TAMP panel (STM32N6 devices)	
	RIF-aware peripherals for STM32N6 MCUs.	
	IO protection inheritance for a non-RIF-aware IP (I2C)	
	GPIO IP panel.	
	Inheritance in RIF GPIO panel	
	PIN reservation	
	HPDMA1 features with RIF implementation (STM32N6 MCUs).	
0		
•	I2C mode panel	
•	I2C features panel	
0	DMA RIF-aware IP inheritance	
	RIF RCC panel (STM32MP2 MPUs)	
	RCC features with RIF implementation (STM32N6 MCUs)	
	RIF panel for EXTI1 (STM32N6 MCUs)	
	RISAF configuration	
Figure 150.	Configuration of a new subregion	156



Figure 151.	Non editable columns	156
	Warning	
Figure 153.	OCTOSPI1&2 configuration 1	157
Figure 154.	OCTOSPI1&2 memory mapping1	158
Figure 155.	OCTOSPI1&2 region size configuration 1	158
Figure 156.	OCTOSPI1&2 inheritances from RISUP1	158
Figure 157.	OCTOSPI1&2 Master CID activation example 1	159
Figure 158.	DDR memory configuration 1	159
Figure 159.	DDR_CTRL_PHY activation	160
Figure 160.	Configuration of RISAF4 (DDR) 1	160
	PCIE memory configuration 1	
Figure 162.	Global lock in RISAF panel for STM32N6 MCUs 1	162
Figure 163.	RISAF configuration for STM32N6 series 1	164
Figure 164.	Sub-regions activation in RISAF (showing activated subregions) 1	165
Figure 165.	Sub-regions activation in RISAF (check the filtering parameter)	165
Figure 166.	Example: RISUP configuration and generated code1	166
Figure 167.	Example: RISAF configuration and generated code1	166
	IAC feature	
Figure 169.	STM32H7 dual-core: peripheral and middleware context assignment 1	68
	STM32H7 dual-core: GPIOs context assignment	
	Pinout & Configuration view for TrustZone [®] -enabled projects	
	Setting privileges for peripherals	
	Setting privileges for GPIO EXTIS 1	
	Configuring security and privilege of DMA requests	
	RCC privilege mode	
	Configuring security and privilege of DMA requests	
	Securing peripherals from GTZC panel	
Figure 178.	OTFDEC secured when TrustZone [®] is active	176
	STM32F469NIHx clock tree configuration view 1	
	Clock tree configuration view with errors	
	Clock tree configuration: enabling RTC, RCC clock source	
- gane ren	and outputs from Pinout view	184
Figure 182.	Clock tree configuration: RCC peripheral advanced parameters	185
	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.	
	Generated init functions without C language "static" keyword	
	Automatic project import	
	Manual project import	
	Import Project menu - Try Import with errors	
	Import Project menu - Successful import after adjustments	
	Set unused pins window	
	Reset used pins window	
1 iyule 201.	Neset used pills withdow	-03



	Set unused GPIO pins with Keep Current Signals Placement checked	
-	Additional software window	
-	Component dependency resolution	
	Details and Warnings panel	
•	Selection of additional software components	
	Additional software components - Updated tree view	
	LPBAM window.	
	CAD Resources view	
0	CAD Resources not available.	
•	CAD Resources selection for download	
-	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 paths (OEM-iRoT and secure manager projects).	
	Application boot path (OEM-uRoT assembled)	
	Application boot path: ST-iRoT and uRoT secure/nonsecure project.	
	Application boot path:	. 22 1
riguie zzz.	ST-iRoT and secure/nonsecure user application assembled	222
Figure 223	Application boot path: ST-iRoT dual figure	
-	Application boot path:	
riguie 224.	(OEM-iRoT and secure/nonsecure user application assembled)	222
Figure 225	Select the device or board	
•	Select the STM32H5 device	
•	Peripheral initialization	
	Boot paths for STM32H56x devices	
	Activate TrustZone	
	Device and peripherals configuration	
	Configuring the project	
-	Saving the project	
	Boot path selection	
•	Select OEM-iRoT	
0	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	
-	Board provisioning	
-	On-screen instructions	
•	Error message	
90.0 201.		. 201



Figure 252.	Select ST-iRoT	238
	Final boot path stage	
Figure 254.	Boot path and Debug Authentication tab	239
	Select the project structure	
	Code is generated	
•	Secure project completed	
•	IDE post build commands	
	Board provisioning	
	On-screen instructions	
	Environment configuration file	
•	EWARM patch installation	
Figure 263.	First boot path stage	245
Figure 264.	Second boot path stage	245
Figure 265.	Final boot path stage	246
Figure 266.	Boot path and Debug Authentication tab	246
Figure 267.	Select the project structure	247
Figure 268.	Code is generated	247
Figure 269.	Nonsecure project completed	248
Figure 270.	IDE post build commands	249
Figure 271.	Secure manager API configuration	249
	Code generated with secure manager API	
	The flash layout.h file	
Figure 274.	The map.properties file	252
	Boot path project.	
	Secure generated project	
Figure 277.	Non Secure generated project	253
	Compilation project	
•	Project folder.	
•	Boot path project.	
-	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	
	Non secure 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	
		200



Figure 306 Felter the email address 271 Figure 307 Enter the email address 271 Figure 308 Password restoration 272 Figure 310 About window 273 Figure 311 ST Tools 274 Figure 312 User interface of the Compare Projects tool 275 Figure 313 Load the first ice file 276 Figure 314 Starting the comparison process 277 Figure 317 The result of comparing two projects having the same structure 278 Figure 317 The result of comparing two projects having the same structure 278 Figure 317 The result of comparing two projects having the same structure 278 Figure 320 The structure of the Paripet stable 280 Figure 321 The structure of the Paripet Settings table 281 Figure 322 Comparison result in Excel format 282 Figure 323 Power Consumption Calculator default view 282 Figure 326 Power Consumption Sequence: New Step default view 285 Figure 327 Frabiling the transition checker option on an already 287 Figure 331. ADC	Figure 305.	The myST display after a user logs in	269
Figure 308. Password restoration 271 Figure 309. Reset password form 272 Figure 311. ST Tools 273 Figure 313. Load the first loc file. 276 Figure 313. Load the first loc file. 276 Figure 314. Starting the comparing two projects 277 Figure 315. Loading the same project 278 Figure 316. Loading the same project saving the same structure 278 Figure 317. The estructure of the Peripherals & Middleware table 279 Figure 317. The structure of the Peripherals & Middleware table 280 Figure 321. Choosing the Excel format to save the comparison result 282 Figure 322. Comparison result in Excel format 282 Figure 323. Power Consumption Calculator default view 284 Figure 324. Battery selection 285 Figure 325. Step management functions 285 Figure 326. Power consumption sequence. New Step default view 286 Figure 327. Enabling the transition checker option on an already 287 configured sequence - Al least one transition invalid			
Figure 309. Reset password form 272 Figure 310. About window 273 Figure 311. ST Tools 274 Figure 312. User interface of the Compare Projects tool. 276 Figure 312. Load the first ioc file. 276 Figure 314. Starting the comparison process 277 Figure 315. The result of comparing two projects having the same structure 278 Figure 317. The result of comparing two projects having the same structure 278 Figure 317. The structure of the Project Settings table 280 Figure 320. The structure of the Project Settings table 281 Figure 321. Choosing the Excel format 282 Figure 323. Power Consumption Calculator default view 284 Figure 325. Step management functions. 285 Figure 326. Power consumption sequence: New Step default view 286 Figure 327. Fransition checker option on an already 287 configured sequence - All transition invalid 287 Figure 328. Fnabling the transition checker option on an already 289 Figure 329. Transition che	•		
Figure 311. ST Tools 274 Figure 312. User interface of the Compare Projects tool 275 Figure 313. Load the first ioc file 276 Figure 314. Starting the comparing two projects 277 Figure 315. The result of comparing two projects having the same structure 278 Figure 316. The structure of the Peripherals & Middleware table 279 Figure 317. The structure of the Project Settings table 280 Figure 320. The structure of the Project Settings table 281 Figure 322. Comparison result in Excel format 282 Figure 323. Power Consumption Calculator default view 284 Figure 327. Enabling the transition checker option on an already 286 Figure 327. Enabling the transition checker option on an already 287 Configured sequence - All transition invalid 287 Figure 331. ADC selected in Pinout view. 280 Figure 332. Power Consumption Calculator configuration window: 287 Figure 333. Power Consumption Calculator configuration window: 287 Figure 333. Power Consumption Calculator configuration window:	-		
Figure 312. User Interface of the Compare Projects tool. 275 Figure 313. Load the first ico file. 276 Figure 314. Starting the comparison process 277 Figure 315. The result of comparing two projects 277 Figure 317. The result of comparing two projects having the same structure. 278 Figure 317. The result of the Target table 279 Figure 320. The structure of the Project Settings table 280 Figure 321. Choosing the Excel format to save the comparison result 282 Figure 322. Comparison result in Excel format to save the comparison result 282 Figure 323. Battery selection 285 Figure 324. Battery selection 285 Figure 325. Step management functions. 285 Figure 326. Power consumption sequence: New Step default view 286 Figure 327. Enabling the transition checker option on an already configured sequence - Al teast one transition invalid 287 Figure 331. ADC selected in Pinout view. 290 Figure 333. ADC selected in Pinout view. 290 Figure 333. Power Consumption Calculator	Figure 310.	About window	. 273
Figure 313. Load the first icc file. 276 Figure 314. Starting the comparison process 277 Figure 315. The result of comparing two projects 277 Figure 316. Loading the same project. 278 Figure 317. The result of comparing two projects having the same structure. 278 Figure 319. The structure of the Peripherals & Middleware table 280 Figure 321. Choosing the Excel format to save the comparison result 282 Figure 322. Comparison result in Excel format. 282 Figure 323. Dewer Consumption Calculator default view 284 Figure 324. Battery selection 285 Figure 325. Step management functions. 285 Figure 326. Power consumption calculator default view 286 Figure 327. Enabling the transition checker option on an already 287 Figure 328. Enabling the transition checker option on an already 287 Figure 331. ADC selected in Pinout view. 280 Figure 332. Nuterpoint calculator view after sequence building 287 Figure 333. ADC selected in Pinout view. 280 Figure 333. Decker option - Show log 287 Figure 334. Sequence table management functions 290 Figure 333. Drespidatabase management functions<	Figure 311.	ST Tools	. 274
Figure 314. Starting the comparison process 277 Figure 315. The result of comparing two projects 277 Figure 316. Loading the same project. 278 Figure 317. The result of comparing two projects having the same structure 278 Figure 319. The structure of the Project Settings table 280 Figure 320. The structure of the Project Settings table 281 Figure 322. Comparison result in Excel format to save the comparison result 282 Figure 323. Power Consumption Calculator default view 284 Figure 324. Battery selection 285 Figure 325. Step management functions. 285 Figure 326. Power consumption sequence: New Step default view 286 Figure 327. Enabling the transition checker option on an already 287 configured sequence - All teast one transition invalid 287 Figure 330. Interpolated power consumption. 289 Figure 332. Power Consumption Calculator view after sequence building 292 Figure 333. Power Consumption Calculator view after sequence building 292 Figure 333. Power Consumption Calculator vie	Figure 312.	User interface of the Compare Projects tool	275
Figure 315. The result of comparing two projects 277 Figure 317. The result of comparing two projects having the same structure 278 Figure 317. The result of comparing two projects having the same structure 278 Figure 318. The structure of the Paripherals & Middleware table 280 Figure 321. Choosing the Excel format to save the comparison result 282 Figure 322. Comparison result in Excel format 282 Figure 323. Power Consumption Calculator default view 284 Figure 324. Battery selection 285 Figure 325. Step management functions 285 Figure 326. Power consumption checker option on an already configured sequence - All transitions valid 287 Figure 328. Enabling the transition checker option on an already configured sequence - Al least one transition invalid 287 Figure 331. ADC selected in Pinout view. 290 Figure 332. Power Consumption Calculator configuration window: ADC enabled using import pinout. 291 Figure 333. Power Consumption Calculator view after sequence building 292 Figure 333. Power Consumption Calculator view after sequence building 292 Figure 333. Power Consumption Calculator view after sequence building 293 Figure 333. Sequence table management functions 293 <t< td=""><td>•</td><td></td><td></td></t<>	•		
Figure 316. Loading the same project 278 Figure 317. The result of comparing two projects having the same structure 278 Figure 318. The structure of the Parighetable 279 Figure 319. The structure of the Peripherals & Middleware table 280 Figure 321. Choosing the Excel format to save the comparison result 282 Figure 322. Comparison result in Excel format to save the comparison result 282 Figure 323. Power Consumption Calculator default view 284 Figure 325. Step management functions. 285 Figure 326. Power consumption sequence: New Step default view 284 Figure 327. Enabling the transition checker option on an already 287 Figure 328. Enabling the transition checker option on an already 287 Figure 331. ADC selected in Pinout view. 290 Figure 332. Power Consumption Calculator configuration window: 290 Figure 333. Power Consumption Calculator view after sequence building 292 Figure 334. Sequence table management functions 292 Figure 335. Power Consumption: Peripherals consumption chart. 293	-		
Figure 317. The result of comparing two projects having the same structure 278 Figure 318. The structure of the Target table 279 Figure 319. The structure of the Peripherals & Middleware table 280 Figure 320. The structure of the Project Settings table 281 Figure 321. Choosing the Excel format to save the comparison result 282 Figure 322. Comparison result in Excel format 282 Figure 323. Power Consumption Calculator default view 284 Figure 324. Battery selection 285 Figure 327. Enabling the transition sequence: New Step default view 286 Figure 327. Enabling the transition checker option on an already 287 Figure 328. Enabling the transition checker option on an already 287 Figure 329. Transition checker option on an already 287 Figure 331. ADC selected in Pinout view. 280 Figure 332. Power Consumption Calculator configuration window: 290 Figure 333. Power Consumption Calculator on wafter sequence building 292 Figure 333. Power Consumption Calculator view after sequence building 292 Figure 333. Power Consumption Calculator view after sequence building 292 Figure 334. Sequence table management functions 293 Figure 335.	-		
Figure 318. The structure of the Parget table 279 Figure 319. The structure of the Project Settings table 280 Figure 320. The structure of the Project Settings table 281 Figure 321. Choosing the Excel format to save the comparison result 282 Figure 322. Comparison result in Excel format 282 Figure 323. Power Consumption Calculator default view 284 Figure 324. Battery selection 285 Figure 325. Step management functions 285 Figure 326. Power consumption sequence: New Step default view 286 Figure 327. Enabling the transition checker option on an already 287 configured sequence - All transitions valid 287 Figure 329. Transition checker option - Show log 287 Figure 320. Interpolated power consumption 280 Figure 331. ADC selected in Pinout view. 290 Figure 332. Power Consumption Calculator configuration window: 290 ADC enabled using import pinout. 291 Figure 333. Power Consumption Calculator view after sequence building 292 Figure 334. Securic table management functions 292 Figure 335. Power Consumption 291 Figure 336. Description of the Results area. 293 <td></td> <td></td> <td></td>			
Figure 319. The structure of the Peripherals & Middleware table 280 Figure 320. The structure of the Project Settings table 281 Figure 321. Choosing the Excel format to save the comparison result 282 Figure 322. Comparison result in Excel format 282 Figure 323. Power Consumption Calculator default view 284 Figure 325. Step management functions 285 Figure 326. Power consumption sequence: New Step default view 286 Figure 327. Enabling the transition checker option on an already 287 configured sequence - All transitions valid 287 Figure 328. Enabling the transition checker option on an already 287 Figure 329. Transition checker option - Show log 287 Figure 331. ADC selected in Pinout view 290 Figure 332. Power Consumption Calculator configuration window: 290 ADC enabled using import pinout. 291 Figure 333. Power Consumption Calculator view after sequence building 292 Figure 334. Sequence table management functions 292 Figure 335. Power Consumption Calculator wiew after sequence building 292 Figure 333. Power Consumption Calculator offiguration window: 292 Figure 334. Sequence table management functions 2			
Figure 320. The structure of the Project Settings table . 281 Figure 321. Choosing the Excel format to save the comparison result . 282 Figure 322. Comparison result in Excel format . 282 Figure 323. Power Consumption Calculator default view . 284 Figure 326. Step management functions . 285 Figure 327. Enabling the transition checker option on an already configured sequence - All transitions valid . 287 Figure 328. Enabling the transition checker option on an already configured sequence - At least one transition invalid . 287 Figure 329. Transition checker option - Show log . 287 Figure 320. Interpolated power consumption . 289 Figure 331. ADC selected in Pinout view . 290 Figure 332. Power Consumption Calculator configuration window: ADC enabled using import pinout . 291 Figure 333. Power Consumption . 293 Figure 334. Sequence table management functions . 293 Figure 335. Power Consumption . 293 Figure 335. Power Consumption . 293 Figure 335. Power Consumption . 293 Figure 336			
Figure 321. Choosing the Excel format to save the comparison result 282 Figure 322. Comparison result in Excel format 282 Figure 323. Power Consumption Calculator default view 284 Figure 325. Step management functions 285 Figure 326. Power consumption sequence: New Step default view 286 Figure 327. Enabling the transition checker option on an already configured sequence - All transitions valid 287 Figure 328. Enabling the transition checker option on an already configured sequence - At least one transition invalid 287 Figure 330. Interpolated power consumption. 289 Figure 331. ADC selected in Pinout view. 290 Figure 332. Power Consumption Calculator configuration window: ADC enabled using import pinout. 291 Figure 333. Power Consumption Calculator view after sequence building 292 Figure 334. Sequence table management functions 292 Figure 335. Power Consumption: Peripherals consumption chart. 293 Figure 336. Description of the Results area. 293 Figure 337. Overall peripheral consumption 295 Figure 338. Selecting SMPS for the			
Figure 322 Comparison result in Excel format 282 Figure 323 Power Consumption Calculator default view 284 Figure 325 Step management functions 285 Figure 326 Power consumption sequence: New Step default view 286 Figure 327 Enabling the transition checker option on an already configured sequence - All transitions valid 287 Figure 328 Enabling the transition checker option on an already configured sequence - All transition invalid 287 Figure 329 Transition checker option - Show log 287 Figure 321 ADC selected in Pinout view 290 Figure 332 Power Consumption Calculator configuration window: ADC enabled using import pinout 291 Figure 333 Power Consumption Calculator view after sequence building 292 Figure 334 Sequence table management functions 293 Figure 335 Power Consumption: Peripherals consumption chart 293 Figure 336 Description of the Results area. 293 Figure 337 Overall peripheral consumption 295 Figure 338 Selecting a different SMPS model 298 Figure 339 SMPS database - Adding new SMPS model 298 <td></td> <td></td> <td></td>			
Figure 323. Power Consumption Calculator default view 284 Figure 324. Battery selection 285 Figure 325. Step management functions 285 Figure 326. Power consumption sequence: New Step default view 286 Figure 327. Enabling the transition checker option on an already configured sequence - AI least one transition invalid 287 Figure 328. Enabling the transition checker option on an already configured sequence - AI least one transition invalid 287 Figure 329. Transition checker option - Show log 287 Figure 331. ADC selected in Pinout view 290 Figure 332. Power Consumption Calculator configuration window: ADC enabled using import pinout. 291 Figure 333. Power Consumption Calculator view after sequence building 292 Figure 334. Sequence table management functions 292 Figure 335. Power Consumption 293 Figure 336. Description of the Results area. 293 Figure 337. Overall peripheral consumption 295 Figure 338. Selecting SMPS for the current project. 297 Figure 339. SMPS database - Selecting a different SMPS model 298			
Figure 324. Battery selection 285 Figure 325. Step management functions 285 Figure 326. Power consumption sequence: New Step default view 286 Figure 327. Enabling the transition checker option on an already configured sequence - All transitions valid 287 Figure 328. Enabling the transition checker option on an already configured sequence - At least one transition invalid 287 Figure 329. Transition checker option - Show log 287 Figure 331. ADC selected in Pinout view. 290 Figure 332. Power Consumption Calculator configuration window: ADC enabled using import pinout. 291 Figure 333. Power Consumption Calculator view after sequence building 292 Figure 335. Power Consumption: Peripherals consumption chart. 293 Figure 336. Description of the Results area. 293 Figure 337. Overall peripheral consumption 295 Figure 340. SMPS database - Adding new SMPS models 298 Figure 341. Current project configuration updated with new SMPS model 299 Figure 342. SMPS database - Selecting a different SMPS model 299 Figure 343. SMPS transition checke			
Figure 325.Step management functions285Figure 326.Power consumption sequence: New Step default view286Figure 327.Enabling the transition checker option on an already configured sequence - All transitions valid287Figure 328.Enabling the transition checker option on an already configured sequence - All transitions valid287Figure 329.Transition checker option - Show log287Figure 320.Interpolated power consumption289Figure 331.ADC selected in Pinout view.290Figure 332.Power Consumption Calculator configuration window: ADC enabled using import pinout.291Figure 333.Power Consumption Calculator view after sequence building292Figure 335.Power Consumption: Peripherals consumption chart.293Figure 336.Description of the Results area.293Figure 337.Overall peripheral consumption295Figure 338.Selecting SMPS for the current project.297Figure 340.SMPS database - Adding new SMPS models298Figure 341.Current project configuration updated with new SMPS model299Figure 342.SMPS database - Selecting a different SMPS model299Figure 343.SMPS transition checker and state diagram helper window.300Figure 343.SMPS transition checker and state diagram helper window.300Figure 343.SMPS transition checker and state diagram helper window.300Figure 344.Configuring the SMPS mode for each step301Figure 345. <t< td=""><td>-</td><td>·</td><td></td></t<>	-	·	
Figure 326. Power consumption sequence: New Step default view 286 Figure 327. Enabling the transition checker option on an already configured sequence - All transitions valid 287 Figure 328. Enabling the transition checker option on an already configured sequence - At least one transition invalid 287 Figure 329. Transition checker option - Show log 287 Figure 320. Interpolated power consumption 289 Figure 331. ADC selected in Pinout view. 290 Figure 332. Power Consumption Calculator configuration window: ADC enabled using import pinout. 291 Figure 333. Power Consumption Calculator view after sequence building 292 Figure 335. Power Consumption: Peripherals consumption chart. 293 Figure 336. Description of the Results area. 293 Figure 337. Overall peripheral consumption 295 Figure 338. Selecting SMPS for the current project. 297 Figure 339. SMPS database - Adding new SMPS models 298 Figure 341. Current project configuration updated with new SMPS model 299 Figure 342. SMPS database management window with new model selected 299 Figure 343. <td></td> <td></td> <td></td>			
Figure 327. Enabling the transition checker option on an already configured sequence - All transitions valid 287 Figure 328. Enabling the transition checker option on an already configured sequence - At least one transition invalid 287 Figure 329. Transition checker option - Show log 287 Figure 330. Interpolated power consumption 289 Figure 331. ADC selected in Pinout view. 290 Figure 332. Power Consumption Calculator configuration window: ADC enabled using import pinout. 291 Figure 333. Power Consumption Calculator view after sequence building 292 Figure 335. Power Consumption: Peripherals consumption chart. 293 Figure 336. Description of the Results area. 293 Figure 337. Overall peripheral consumption 295 Figure 338. Selecting SMPS for the current project. 297 Figure 330. SMPS database - Adding new SMPS model 298 Figure 341. Current project configuration updated with new SMPS model 299 Figure 342. SMPS database management window with new model selected. 299 Figure 343. SMPS transition checker and state diagram helper window. 300 Figure 3			
configured sequence - All transitions valid287Figure 328.Enabling the transition checker option on an already configured sequence - At least one transition invalid287Figure 329.Transition checker option - Show log287Figure 330.Interpolated power consumption289Figure 331.ADC selected in Pinout view.290Figure 332.Power Consumption Calculator configuration window: ADC enabled using import pinout291Figure 333.Power Consumption Calculator view after sequence building292Figure 334.Sequence table management functions292Figure 335.Power Consumption: Peripherals consumption chart293Figure 336.Description of the Results area.293Figure 337.Overall peripheral consumption295Figure 338.Selecting SMPS for the current project.297Figure 340.SMPS database - Adding new SMPS models298Figure 341.Current project configuration updated with new SMPS model299Figure 343.SMPS transition checker and state diagram helper window.300Figure 344.Configuring the SMPS mode for each step301Figure 345.RF related consumption (STM32WB series only).302Figure 346.RF sub-GHz configuration304Figure 347.ZigBee configuration (STM32WB series only).303Figure 348.Re sub-GHz configuration304Figure 349.Power Consumption Calculator – Example sequence loading305Figure 349.Power Consumption Calcula			. 286
Figure 328. Enabling the transition checker option on an already configured sequence - At least one transition invalid 287 Figure 329. Transition checker option - Show log 287 Figure 330. Interpolated power consumption 289 Figure 331. ADC selected in Pinout view. 290 Figure 332. Power Consumption Calculator configuration window: ADC enabled using import pinout. 291 Figure 333. Power Consumption Calculator view after sequence building 292 Figure 335. Power Consumption: Peripherals consumption chart. 293 Figure 336. Description of the Results area. 293 Figure 337. Overall peripheral consumption 295 Figure 338. Selecting SMPS for the current project. 297 Figure 339. SMPS database - Adding new SMPS models 298 Figure 340. SMPS database - Selecting a different SMPS model 299 Figure 342. SMPS transition checker and state diagram helper window. 300 Figure 343. SMPS mode for each step 301 Figure 344. Configuring the SMPS mode for each step 301 Figure 345. RF related consumption (STM32WB series only). 302	Figure 327.		~~-
configured sequence - At least one transition invalid287Figure 329.Transition checker option - Show log287Figure 330.Interpolated power consumption289Figure 331.ADC selected in Pinout view290Figure 332.Power Consumption Calculator configuration window: ADC enabled using import pinout291Figure 333.Power Consumption Calculator view after sequence building292Figure 334.Sequence table management functions292Figure 335.Power Consumption: Peripherals consumption chart.293Figure 336.Description of the Results area.293Figure 337.Overall peripheral consumption295Figure 338.Selecting SMPS for the current project.297Figure 340.SMPS database - Adding new SMPS models298Figure 341.Current project configuration updated with new SMPS model299Figure 342.SMPS database management window with new model selected.299Figure 343.Selecting SMPS mode for each step300Figure 344.Configuring the SMPS mode for each step301Figure 345.RF related consumption (STM32WB series only)302Figure 346.RF Bluetooth Low-Energy mode configuration (STM32WB series only)303Figure 348.RF sub-GHz configuration304Figure 349.Power Consumption Calculator – Example sequence loading305Figure 349.Power Consumption Calculator – Example sequence loading305Figure 345.Power Consumption Calculator – Example se		- · ·	. 287
Figure 329.Transition checker option - Show log287Figure 330.Interpolated power consumption.289Figure 331.ADC selected in Pinout view.290Figure 332.Power Consumption Calculator configuration window: ADC enabled using import pinout.291Figure 333.Power Consumption Calculator view after sequence building292Figure 334.Sequence table management functions292Figure 335.Power Consumption: Peripherals consumption chart.293Figure 336.Description of the Results area.293Figure 337.Overall peripheral consumption295Figure 338.Selecting SMPS for the current project.297Figure 339.SMPS database - Adding new SMPS models298Figure 340.SMPS database - Selecting a different SMPS model299Figure 341.Current project configuration updated with new SMPS model299Figure 343.SMPS database management window with new model selected.299Figure 343.SMPS transition checker and state diagram helper window.300Figure 344.Configuring the SMPS mode for each step301Figure 345.RF related consumption (STM32WB series only)303Figure 347.ZigBee configuration (STM32WB series only)303Figure 348.RF sub-GHz configuration (STM32WB series only)303Figure 349.Power Consumption Calculator – Example sequence loading305Figure 349.Power Consumption Calculator – Example sequence loading305Figure 350.	Figure 328.		
Figure 330.Interpolated power consumption.289Figure 331.ADC selected in Pinout view.290Figure 332.Power Consumption Calculator configuration window: ADC enabled using import pinout.291Figure 333.Power Consumption Calculator view after sequence building292Figure 334.Sequence table management functions292Figure 335.Power Consumption: Peripherals consumption chart.293Figure 336.Description of the Results area.293Figure 337.Overall peripheral consumption295Figure 338.Selecting SMPS for the current project.297Figure 339.SMPS database - Adding new SMPS models298Figure 340.SMPS database - Selecting a different SMPS model299Figure 341.Current project configuration updated with new SMPS model299Figure 342.SMPS database management window with new model selected.299Figure 343.SMPS transition checker and state diagram helper window.300Figure 344.Configuring the SMPS mode for each step301Figure 345.RF related consumption (STM32WB series only).302Figure 347.ZigBee configuration (STM32WB series only).303Figure 348.RF sub-GHz configuration304Figure 349.Power Consumption Calculator – Example set305Figure 349.Power Consumption Calculator – Example sequence loading.305Figure 345.Power Consumption Calculator – Example sequence loading.305Figure 349.Power Consum		- · ·	
Figure 331.ADC selected in Pinout view.290Figure 332.Power Consumption Calculator configuration window: ADC enabled using import pinout.291Figure 333.Power Consumption Calculator view after sequence building292Figure 334.Sequence table management functions292Figure 335.Power Consumption: Peripherals consumption chart.293Figure 336.Description of the Results area.293Figure 337.Overall peripheral consumption295Figure 338.Selecting SMPS for the current project.297Figure 339.SMPS database - Adding new SMPS models298Figure 340.SMPS database - Selecting a different SMPS model299Figure 341.Current project configuration updated with new SMPS model299Figure 343.SMPS transition checker and state diagram helper window.300Figure 343.SMPS transition checker and state diagram helper window.300Figure 344.Configuring the SMPS mode for each step301Figure 345.RF related consumption (STM32WB series only).303Figure 346.RF Bluetooth Low-Energy mode configuration (STM32WB series only)303Figure 348.RF sub-GHz configuration304Figure 349.Power Consumption Calculator – Example sequence loading305Figure 345.Power Consumption Calculator – Example sequence loading305Figure 345.Power Consumption Calculator – Example sequence new selection305	-	· •	
Figure 332.Power Consumption Calculator configuration window: ADC enabled using import pinout.291Figure 333.Power Consumption Calculator view after sequence building292Figure 334.Sequence table management functions292Figure 335.Power Consumption: Peripherals consumption chart.293Figure 336.Description of the Results area.293Figure 337.Overall peripheral consumption295Figure 338.Selecting SMPS for the current project.297Figure 339.SMPS database - Adding new SMPS models298Figure 340.SMPS database - Selecting a different SMPS model298Figure 341.Current project configuration updated with new SMPS model299Figure 342.SMPS database management window with new model selected299Figure 343.SMPS transition checker and state diagram helper window.300Figure 344.Configuring the SMPS mode for each step301Figure 345.RF related consumption (STM32WB series only).303Figure 347.ZigBee configuration (STM32WB series only).303Figure 348.RF sub-GHz configuration304Figure 349.Power Consumption Calculator – Example set305Figure 341.Power Consumption Calculator – Example sequence loading305Figure 345.Power Consumption Calculator – Example sequence new selection306			
ADC enabled using import pinout.291Figure 333.Power Consumption Calculator view after sequence building292Figure 334.Sequence table management functions292Figure 335.Power Consumption: Peripherals consumption chart.293Figure 336.Description of the Results area.293Figure 337.Overall peripheral consumption295Figure 338.Selecting SMPS for the current project.297Figure 339.SMPS database - Adding new SMPS models298Figure 340.SMPS database - Selecting a different SMPS model298Figure 341.Current project configuration updated with new SMPS model299Figure 342.SMPS database management window with new model selected299Figure 343.SMPS transition checker and state diagram helper window.300Figure 344.Configuring the SMPS mode for each step301Figure 345.RF related consumption (STM32WB series only).303Figure 347.ZigBee configuration (STM32WB series only).303Figure 348.RF sub-GHz configuration			. 290
Figure 333.Power Consumption Calculator view after sequence building292Figure 334.Sequence table management functions292Figure 335.Power Consumption: Peripherals consumption chart293Figure 336.Description of the Results area.293Figure 337.Overall peripheral consumption295Figure 338.Selecting SMPS for the current project.297Figure 339.SMPS database - Adding new SMPS models298Figure 340.SMPS database - Selecting a different SMPS model.298Figure 341.Current project configuration updated with new SMPS model299Figure 342.SMPS database management window with new model selected.299Figure 343.SMPS transition checker and state diagram helper window.300Figure 344.Configuring the SMPS mode for each step301Figure 345.RF related consumption (STM32WB series only).303Figure 346.RF Bluetooth Low-Energy mode configuration (STM32WB series only)303Figure 348.RF sub-GHz configuration304Figure 349.Power Consumption Calculator – Example set305Figure 350.Power Consumption Calculator – Example sequence loading305Figure 351.Power Consumption Calculator – Example sequence new selection306	Figure 332.		004
Figure 334.Sequence table management functions292Figure 335.Power Consumption: Peripherals consumption chart293Figure 336.Description of the Results area293Figure 337.Overall peripheral consumption295Figure 338.Selecting SMPS for the current project297Figure 339.SMPS database - Adding new SMPS models298Figure 340.SMPS database - Selecting a different SMPS model298Figure 341.Current project configuration updated with new SMPS model299Figure 342.SMPS database management window with new model selected299Figure 343.SMPS transition checker and state diagram helper window300Figure 344.Configuring the SMPS mode for each step301Figure 345.RF related consumption (STM32WB series only)303Figure 347.ZigBee configuration (STM32WB series only)303Figure 348.RF sub-GHz configuration304Figure 349.Power Consumption Calculator – Example set305Figure 350.Power Consumption Calculator – Example sequence loading305Figure 351.Power Consumption Calculator – Example sequence new selection306	E :		
Figure 335.Power Consumption: Peripherals consumption chart.293Figure 336.Description of the Results area.293Figure 337.Overall peripheral consumption295Figure 338.Selecting SMPS for the current project.297Figure 339.SMPS database - Adding new SMPS models298Figure 340.SMPS database - Selecting a different SMPS model298Figure 341.Current project configuration updated with new SMPS model299Figure 342.SMPS database management window with new model selected299Figure 343.SMPS transition checker and state diagram helper window.300Figure 344.Configuring the SMPS mode for each step301Figure 345.RF related consumption (STM32WB series only).302Figure 347.ZigBee configuration (STM32WB series only)303Figure 348.RF sub-GHz configuration304Figure 349.Power Consumption Calculator – Example set305Figure 350.Power Consumption Calculator – Example sequence loading305Figure 351.Power Consumption Calculator – Example sequence new selection306			
Figure 336.Description of the Results area.293Figure 337.Overall peripheral consumption295Figure 338.Selecting SMPS for the current project.297Figure 339.SMPS database - Adding new SMPS models298Figure 340.SMPS database - Selecting a different SMPS model.298Figure 341.Current project configuration updated with new SMPS model299Figure 342.SMPS database management window with new model selected.299Figure 343.SMPS transition checker and state diagram helper window.300Figure 344.Configuring the SMPS mode for each step301Figure 345.RF related consumption (STM32WB series only).302Figure 347.ZigBee configuration (STM32WB series only)303Figure 348.RF sub-GHz configuration304Figure 349.Power Consumption Calculator – Example set305Figure 351.Power Consumption Calculator – Example sequence loading305Figure 351.Power Consumption Calculator – Example sequence new selection306			
Figure 337.Overall peripheral consumption295Figure 338.Selecting SMPS for the current project.297Figure 339.SMPS database - Adding new SMPS models298Figure 340.SMPS database - Selecting a different SMPS model.298Figure 341.Current project configuration updated with new SMPS model299Figure 342.SMPS database management window with new model selected.299Figure 343.SMPS transition checker and state diagram helper window.300Figure 344.Configuring the SMPS mode for each step301Figure 345.RF related consumption (STM32WB series only).302Figure 346.RF Bluetooth Low-Energy mode configuration (STM32WB series only)303Figure 348.RF sub-GHz configuration304Figure 349.Power Consumption Calculator – Example set305Figure 351.Power Consumption Calculator – Example sequence new selection306			
Figure 338.Selecting SMPS for the current project.297Figure 339.SMPS database - Adding new SMPS models298Figure 340.SMPS database - Selecting a different SMPS model298Figure 341.Current project configuration updated with new SMPS model299Figure 342.SMPS database management window with new model selected299Figure 343.SMPS transition checker and state diagram helper window.300Figure 344.Configuring the SMPS mode for each step301Figure 345.RF related consumption (STM32WB series only).302Figure 346.RF Bluetooth Low-Energy mode configuration (STM32WB series only)303Figure 347.ZigBee configuration (STM32WB series only)303Figure 348.RF sub-GHz configuration304Figure 349.Power Consumption Calculator – Example set305Figure 350.Power Consumption Calculator – Example sequence loading305Figure 351.Power Consumption Calculator – Example sequence new selection306	•	•	
Figure 339.SMPS database - Adding new SMPS models298Figure 340.SMPS database - Selecting a different SMPS model298Figure 341.Current project configuration updated with new SMPS model299Figure 342.SMPS database management window with new model selected299Figure 343.SMPS transition checker and state diagram helper window300Figure 344.Configuring the SMPS mode for each step301Figure 345.RF related consumption (STM32WB series only)302Figure 346.RF Bluetooth Low-Energy mode configuration (STM32WB series only)303Figure 347.ZigBee configuration (STM32WB series only)303Figure 348.RF sub-GHz configuration304Figure 349.Power Consumption Calculator – Example set305Figure 351.Power Consumption Calculator – Example sequence new selection306			
Figure 340.SMPS database - Selecting a different SMPS model298Figure 341.Current project configuration updated with new SMPS model299Figure 342.SMPS database management window with new model selected299Figure 343.SMPS transition checker and state diagram helper window300Figure 344.Configuring the SMPS mode for each step301Figure 345.RF related consumption (STM32WB series only)302Figure 346.RF Bluetooth Low-Energy mode configuration (STM32WB series only)303Figure 347.ZigBee configuration (STM32WB series only)303Figure 348.RF sub-GHz configuration304Figure 349.Power Consumption Calculator – Example set305Figure 351.Power Consumption Calculator – Example sequence new selection306			
Figure 341.Current project configuration updated with new SMPS model299Figure 342.SMPS database management window with new model selected299Figure 343.SMPS transition checker and state diagram helper window300Figure 344.Configuring the SMPS mode for each step301Figure 345.RF related consumption (STM32WB series only)302Figure 346.RF Bluetooth Low-Energy mode configuration (STM32WB series only)303Figure 347.ZigBee configuration (STM32WB series only)303Figure 348.RF sub-GHz configuration304Figure 349.Power Consumption Calculator – Example set305Figure 350.Power Consumption Calculator – Example sequence loading305Figure 351.Power Consumption Calculator – Example sequence new selection306	•	•	
Figure 342.SMPS database management window with new model selected.299Figure 343.SMPS transition checker and state diagram helper window.300Figure 344.Configuring the SMPS mode for each step301Figure 345.RF related consumption (STM32WB series only).302Figure 346.RF Bluetooth Low-Energy mode configuration (STM32WB series only)303Figure 347.ZigBee configuration (STM32WB series only)303Figure 348.RF sub-GHz configuration (STM32WB series only)304Figure 349.Power Consumption Calculator – Example set305Figure 350.Power Consumption Calculator – Example sequence loading305Figure 351.Power Consumption Calculator – Example sequence new selection306			
Figure 343.SMPS transition checker and state diagram helper window.300Figure 344.Configuring the SMPS mode for each step301Figure 345.RF related consumption (STM32WB series only).302Figure 346.RF Bluetooth Low-Energy mode configuration (STM32WB series only)303Figure 347.ZigBee configuration (STM32WB series only)303Figure 348.RF sub-GHz configuration304Figure 349.Power Consumption Calculator – Example set305Figure 350.Power Consumption Calculator – Example sequence loading305Figure 351.Power Consumption Calculator – Example sequence new selection306			
Figure 344.Configuring the SMPS mode for each step301Figure 345.RF related consumption (STM32WB series only)302Figure 346.RF Bluetooth Low-Energy mode configuration (STM32WB series only)303Figure 347.ZigBee configuration (STM32WB series only)303Figure 348.RF sub-GHz configuration304Figure 349.Power Consumption Calculator – Example set305Figure 350.Power Consumption Calculator – Example sequence loading305Figure 351.Power Consumption Calculator – Example sequence new selection306			
Figure 345.RF related consumption (STM32WB series only).302Figure 346.RF Bluetooth Low-Energy mode configuration (STM32WB series only)303Figure 347.ZigBee configuration (STM32WB series only)303Figure 348.RF sub-GHz configuration304Figure 349.Power Consumption Calculator – Example set305Figure 350.Power Consumption Calculator – Example sequence loading305Figure 351.Power Consumption Calculator – Example sequence new selection306			
Figure 346.RF Bluetooth Low-Energy mode configuration (STM32WB series only)	•		
Figure 347.ZigBee configuration (STM32WB series only)303Figure 348.RF sub-GHz configuration304Figure 349.Power Consumption Calculator – Example set305Figure 350.Power Consumption Calculator – Example sequence loading305Figure 351.Power Consumption Calculator – Example sequence new selection306			
Figure 348. RF sub-GHz configuration 304 Figure 349. Power Consumption Calculator – Example set 305 Figure 350. Power Consumption Calculator – Example sequence loading 305 Figure 351. Power Consumption Calculator – Example sequence new selection 306			
Figure 349.Power Consumption Calculator – Example set305Figure 350.Power Consumption Calculator – Example sequence loading305Figure 351.Power Consumption Calculator – Example sequence new selection306			
Figure 350.Power Consumption Calculator – Example sequence loading305Figure 351.Power Consumption Calculator – Example sequence new selection306			
Figure 351. Power Consumption Calculator – Example sequence new selection			



Figure 354.	DDR3 configuration	309
•	DDR Suite - Target connected	
	DDR activity logs	
	DDR interactive logs	
	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	
0	MMT view	
-	Start a project	
•	Use TrustZone	
•	Default settings	
Figure 370.	Region information	318
Figure 371.	Tooltip	318
Figure 372.	IP configuration	319
Figure 373.	IP under control	319
Figure 374.	Linker files update.	320
Figure 375.	Configure an external memory	321
Figure 376.	New region created.	321
	Adding a new region	
	Adding a new memory	
	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).	
	MMT view for U5 without TrustZone	
0	MMT view for H7 single core	
-	Default data region	
	FMC activation	
•	Default mapping	
•	Before the swap	
•	After the swap	
•	Before remapping	
0	After remapping	
•	MMT usage	
•	Firmware version	
	MMT configuration for STM32WB5x	
Figure 404.		335



	CORTEX_M7 Mode and Configuration	
	CORTEX_M4 Mode and Configuration	
•	Default settings	
-	Choose an STM32H7 dual-core product	
-	Choose an STM32H7 dual-core product	
	Region 0 added	
-	Activate Memory Management support	
	Default setting for new application region	
	Adding a new region	
	Configure the NVIC1, NVIC2 and select their HSEM global interrupt.	
	OPENAMP_M7 parameters settings	
	OPENAMP_M4 parameters settings	
	The reserved memory regions for OPENAMP using MMT	
	Linker files update(stm32h755xxx_flash_cm4.icf)	
	Linker files update(stm32h755xxx_flash_cm7.icf)	
Figure 420.	Configuration of ETH IP	346
Figure 421.	ETH MMT region	346
Figure 422.	IP configuration.	347
Figure 423.	The defined memories under the linker file (Cortex-M7).	348
Figure 424.	The defined memories under the linker file (Cortex-M4)	349
-	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.	
	IP configuration.	
-	Linker files update.	
	Memory assignment for context Boot H7RS.	
	EXTMEM_MANAGER "Select boot code generation" disabled	
	Execute In Place	
	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	
	MMT usage	
	User interface	
•		
	Linker files update.	
	Impact on RADIO (STM32WB09).	
	MMT/boot path (STM32H7RS).	
•	MMT/boot path (STM32H5)	
•	Linker files location (STM32H7RS on the left, STM32H5 on the right).	
•	App_User declaration (STM32H7RS).	
	App_User declaration (STM32H5)	
	Labels for pins generating define statements	
Figure 456.	User constant generating define statements	367



Eiguro 457	Duplicate labels	260
	HAL-based peripheral initialization: usart.c code snippet	
-	LL-based peripheral initialization: usart.c code snippet	
	HAL versus LL: main.c code snippet	
-	Default content of the extra_templates folder	
	extra_templates folder with user templates	
	Project root folder with corresponding custom generated files	
	User custom folder for templates	
-	Custom folder with corresponding custom generated files	377
Figure 466.	Update of the project .ewp file (EWARM IDE)	
	for preprocessor define statements	
Figure 467.	Update of stm32f4xx_hal_conf.h file to enable selected modules	379
Figure 468.	New groups and new files added to groups in EWARM IDE	379
	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	
-	Project generated with CMSIS-Pack software component	
•	MCU selection	
	Pinout view with MCUs selection	
	Pinout view without MCUs selection window	
	GPIO pin configuration	
	Timer configuration	
Figure 487.	Simple pinout configuration	398
Figure 488.	Save Project As window	399
Figure 489.	Generate Project Report - New project creation	400
Figure 490.	Generate Project Report - Project successfully created	400
Figure 491.	Clock tree view	401
Figure 492.	HSI clock enabled	402
Figure 493.	HSE clock source disabled	402
Figure 494.	HSE clock source enabled	402
	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	
	GPIO configuration color scheme and toolip	
Figure 502.	DMA peremeters configuration window	407
	DMA parameters configuration window	
-	Middleware tooltip	
	USB Host configuration	
	FatFs over USB mode enabled	
⊢igure 507.	System view with FatFs and USB enabled	410



•	FatFs define statements 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	
Figure 521.	STM32CubeMX generated project open in IAR™ IDE	420
	IAR™ options	
•	SWD connection	
•	Project building log	
	User Section 2	
•	User Section 4	
Figure 527.	Import Project menu	423
	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	
Figure 537.	Power Consumption Calculation example	434
Figure 538.	VDD and battery selection menu	434
Figure 539.	Sequence table	435
Figure 540.	sequence results before optimization	435
Figure 541.	Step 1 optimization	436
Figure 542.	Step 5 optimization	437
Figure 543.	Step 6 optimization	438
	Step 7 optimization	
Figure 545.	Step 8 optimization	440
	Step 10 optimization	
Figure 547.	Power sequence results after optimizations	441
Figure 548.	Selecting NUCLEO_L053R8 board	443
	Selecting debug pins	
	Selecting TIM2 clock source	
Figure 551.	Selecting asynchronous mode for USART2	445
-	Checking pin assignment	
-	Configuring the MCU clock tree	
-	Configuring USART2 parameters	
-	Configuring TIM2 parameters	
-	Enabling TIM2 interrupt	
•	Project Settings menu.	
	Generating the code	
Figure 559.	Checking the communication port	452



Figure 560.	Setting Tera Term port parameters
	Setting Tera Term port parameters
0	Existing or new project pinout
Figure 563.	List of pinout compatible MCUs - Partial match
	with hardware compatibility
•	List of Pinout compatible MCUs - Exact and partial match
	Selecting a compatible MCU and importing the configuration
	Configuration imported to the selected compatible MCU
	Additional software components enabled for the current project
•	Pack software components: no configurable parameters
	Pack tutorial: project settings
	Generated project with third party pack components
•	Hardware prerequisites
-	Embedded software packages
	Mobile application
	Installing Embedded software packages
	Starting a new project - selecting the NUCLEO-L053R8 board
	Starting a new project - initializing all peripherals
	Selecting X-Cube-BLE1 components
	Configuring peripherals and GPIOs
	Configuring NVIC interrupts
	Enabling X-Cube-BLE1
	Configuring the SensorDemo project
	Open SensorDemo project in the IDE toolchain
•	LPBAM project
•	Project timeline
	Project with LPBAM capability
	LPBAM scenario & configuration view
	Adding an application
	SoC and IPs configuration
	LPBAM scenario: creation & configuration panels
•	Clock tree configuration
	Available IPs
	IP configuration: advanced settings
	LPBAM low power settings
	Adding nodes to a queue
	Queue in circular mode
	Queue looping back on IP data transfer
•	LPBAM queue: DMA configuration
0	LPBAM functions contextual help
	LPBAM queue node configuration
	LPBAM node: configuring hardware resources
	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
	LPBAM project - Peripheral no context assignment
Figure 610.	LPBAM application - Clock source no context assignment



Figure 611	LPBAM application - Secure context assignment
	LPBAM design security coherency check
•	Block mapping
•	Block remapping
•	Block remapping - Example 1
	Block remapping - Example 2
Figure 617.	Block inter-dependency - SPI signals assigned to PB3/4/5
	Block inter-dependency - SPI1_MISO function assigned to PA6
	One block = one peripheral mode - I2C1_SMBA function assigned to PB5
-	Block remapping - Example 2
	Function remapping example
	Block shifting not applied
Figure 623.	Block shifting applied
Figure 624.	FreeRTOS HOOK functions to be completed by user
Figure 625.	LwIP 1.4.1 configuration
Figure 626.	LwIP 1.5 configuration
Figure 627.	Libjpeg configuration window
	Mbed TLS without LwIP
	Mbed TLS with LwIP and FreeRTOS
Figure 630.	Mbed TLS configuration window
Figure 631.	Enabling the TouchSensing peripheral
Figure 632.	Touch-sensing sensor selection panel
	TouchSensing configuration panel
	BLE and Thread middleware support in STM32CubeMX517
	STM32CubeWB Package download
•	STM32CubeWB BLE applications folder
•	BLE Server profile selection
	BLE Client profile selection
	Thread application selection
Figure 640.	Enabling OpenAmp for STM32MPUs
Figure 641.	Enabling the Resource Manager for STM32MPUs
	Resource Manager: peripheral assignment view
	STM32 microcontroller part numbering scheme
Figure 644.	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.



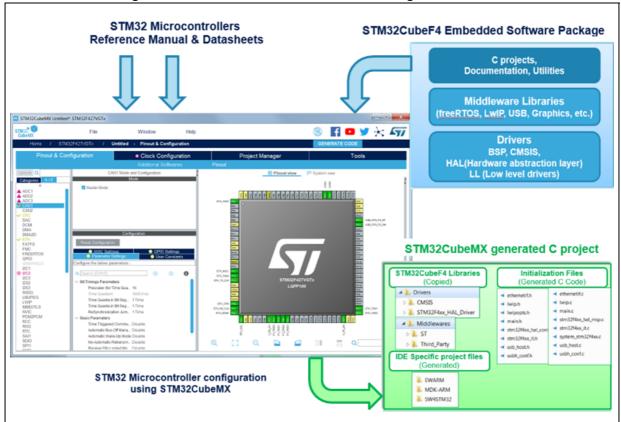


Figure 1. Overview of STM32CubeMX C code generation flow



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[®] 40
- 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



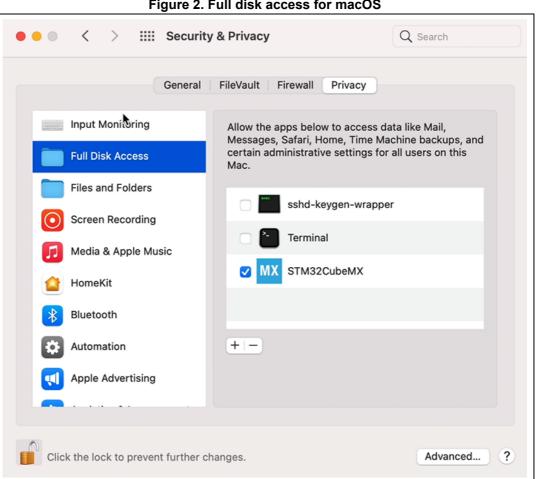


Figure 2. Full disk access for macOS

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*)

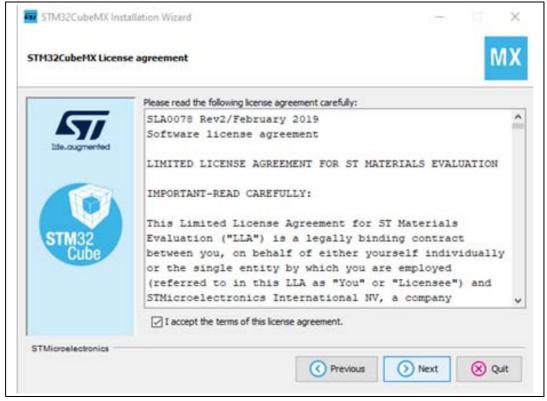




Figure 4. Welcom

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





od the ST Terms of Use.
od the ST Terms of Use.
ler) collects and uses anonymous features usage affiliates) when you use the application for the proving the application. If of your features usage statistics when you use with effect for the future via the menu General Settings

Figure 6. Terms of use

Figure 7. Default installation path

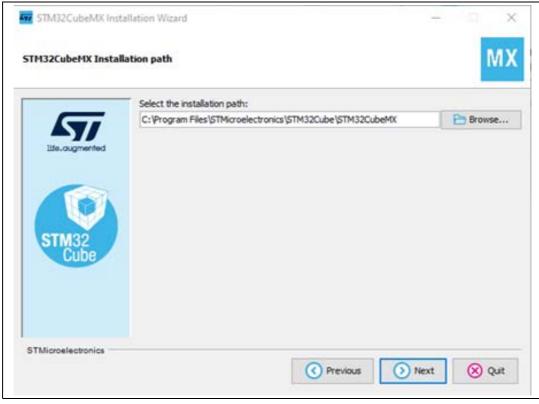




Figure	8.	Setup	of	shortcuts
	•••	00000	•••	0

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	107.0000
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	allation Wizard – 🗆 🗙
STM32CubeMX Packag	ge installation MX
	Pack installation progress: C:\Apps\STM32CubeMX6.11.0\db\mcu\config\lConfig\TIM-STM32MP1xx_DefMapping.xml
	Core
life.augmented	Overall installation progress:
STM32 Cube	



STM32CubeMX Insta	llation Wizard —	. [×
STM32CubeMX Install	ation done		ľ	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		6	🕗 Do	one

Figure 10. Installation script

Figure 11. Installation path

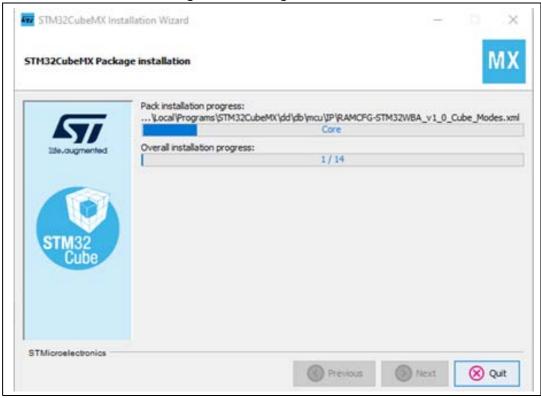
STM32CubeMX Instal	lation Wizard	– 🗆 X
STM32CubeMX Installa	tion path	MX
	Select the installation path:	
life.augmented	C:\Apps\STM32CubeMX6.11.0	Browse
STM32 Cube		
STMicroelectronics	O Previous (Next 🚫 Quit



M32CubeMX Shorte	uts setup	M.
Life-ougmented	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:
Greate additional shortcuts on the desktop Enter Y for Yes, N for No:
create shortcut for: all users Enter Y for Yes, N for No:
E Starting to unpack] E Processing package: Core (1/3)] E Processing package: Old DataBases (2/3)] E Processing package: Help (3/3)] E Unpacking finished]
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. An uninstaller program has been created in:	
life.augmented	C:\Program Files\STMicroelectronics\STM32Cube\STM32Ci	ubeMX_5_0_0\Uninsta
STM32 Cube		
	Generate an automatic installation so	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 46



```
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



Command line	Table 1. Command line summary (co 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.	Command	line summary	(continued)
	•••••		(001101000)



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></customfwlocation>	Specifies a path to STM32Cube MCU software libraries different from STM32Cube repository path (specified under Help > Updater settings).	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*).

Thernet 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
Remove
Choose Settings if you need to configure a proxy Settings
Never dial a connection
\bigcirc Dial <u>w</u> henever a network connection is not present
Always dial my default connection
Current None S <u>et default</u>
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
 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

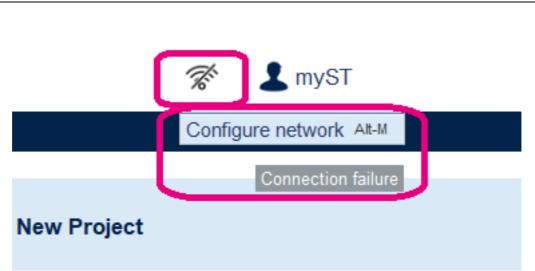
Figure 19	. Connection	Parameters tab	- Manual Confi	guration of Prox	v Server

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 STM32CubeMX 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.









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 *www.st.com*.

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 Packages Manager		X					
STM32Cube MCU Packages and embedded software packs releases							
Releases Information was last refreshed 2 hours ago.							
STM32Cube MCU Packages STMicroelectronics							
Description	Installed Version	Available Version					
► STM32F3							
▼ STM32F4							
STM32Cube MCU Package for STM32F4 Series (Size : 648 MB)		1.22.0RC1					
STM32Cube MCU Package for STM32F4 Series	1.21.0	1.21.0					
STM32Cube MCU Package for STM32F4 Series	1.19.0	1.19.0					
Details <u>STM32CubeF4 Firmware Package V1.21.0 / 23-February-2018</u> <u>Main Changes</u> • General updates to fix known defects and enhancements implementation. • Add new STemWin applications for STM32F4x9I_EVAL, STM32F429I-Discovery, STM32F469_EVAL and STM32F469-Discovery platforms.							
From Local From Url Refresh Install N	Now Remove N	low 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 its version number which 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.



rigure 22. Managing embedded software packages - help menu	
Embedded Software Packages Manager	
STM32Cube MCU Packages and embedded software packs releases	
Releases Information was last refreshed 2 hours ago.	-
STM32Cube MCU Packages STMicroelectronics ST Microelectronics Packs	
Description Available Version	
▼ X-CUBE-BLE1 ↓ X-Cube-BLE1 pack	
BLE stack and sample applications for BlueNRG-MS module 1.1.0	
Details	
Click Refresh to retrieve the latest versions	
From Local From Url Refresh Install Now Remove Now 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.
- 3. 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*).



	Figure 25. Managing embedded software packages - Adding a new un								
[🚾 User	Defined Pa	cks Manager						X
	Manage Urls for user defined embedded software packs								
		Vendor	Name			URL			
		_							
		MX	Add new Url						
F			To add packs, please en - A valid pdsc (Ex: http:/ - A valid pack index (Ex	//www.vendor.co	m/pack/Ven	dor.Pack			
				С	heck	OK	C	ancel	
				New	Remove	(ЭK	Cance	el

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**.

Figure 25. User-defined list of software packs

ſ	User Defined Packs Manager					
	Manage Urls for user defined embedded software packs					
		Vendor	Name	URL		
	Oryx-E	mbedded	Middleware	http://www.oryx-embedded.com/download/pack/Oryx-Embedded.Middleware.pdsc		
				New Remove OK Cancel		

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.



🔤 Ember	dded Software Packages Manager	×					
STM32Cube MCU Packages and embedded software packs releases							
- 🕀	Releases Information was last refreshed less than one hour ago.						
STM32C	ube MCU Packages Oryx-Embedded STMicroelectronics						
	Description	Available Version					
▼ Mie	ddleware						
	Middleware Package (CycloneTCP, CycloneSSL and CycloneCrypto) (Size : 10.28 MB)	1.8.2					
	Middleware Package (CycloneTCP, CycloneSSL and CycloneCrypto) (Size : 10.9 MB)	1.8.0					
	Middleware Package (CycloneTCP, CycloneSSL and CycloneCrypto)	1.7.8					
Middleware Package (CycloneTCP, CycloneSSL and CycloneCrypto) (Size : 6.6 MB) 1.7.6							
Details							
1.8.2 : - CoAP client implementation (including support for DTLS-secured CoAP, Observe and Block-Wise Transfers) - Support for TLS/DTLS Raw Public Keys (RPK) - Support for ECDH key exchange based on X25519 and X448 - Support for CUrve25519 and Curve448 elliptic curves (constant time implementation) - Support for SHA3 hash algorithm for digital signatures (RSA, DSA and ECDSA) - Added RC2 block cipher - Added Support for SNMP-MPD-MIB database (RFC 3412)							
From L	From Local From Url Refresh Install Now Remove Now Close						



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.

Figure 27.	License	agreement	acceptance
------------	---------	-----------	------------

ſ	Licensing Agreement
	Oryx-Embedded Middleware 1.8.2 License Agreement
	Please read and accept the following agreement carefully to finish the installation:
	GNU GENERAL PUBLIC LICENSE
	Version 2, June 1991
	Copyright (C) 1989, 1991 Free Software Foundation, Inc. 675 Mass Ave, Cambridge, MA 02139, USA
	Everyone is permitted to copy and distribute verbatim copies
	of this license document, but changing it is not allowed.
	Preamble
	The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public
	 I accept the terms of this license agreement
	O I do not accept the terms of this license agreement
	Finish Cancel

	rigare zei zinbedaed centrare paok reioace - eucococia		
🔤 Embe	dded Software Packages Manager		X
	STM32Cube MCU Packages and embedded software packs releases		
	Releases Information was last refreshed less than one hour ago.		
STM32C	ube MCU Packages Oryx-Embedded STMicroelectronics		
	Description	Available Versio	
▼ Mi	ddleware		
	Middleware Package (CycloneTCP, CycloneSSL and CycloneCrypto)	1.8.2	
	Middleware Package (CycloneTCP, CycloneSSL and CycloneCrypto) (Size : 10.9 MB)	1.8.0	
	Middleware Package (CycloneTCP, CycloneSSL and CycloneCrypto)	1.7.8	
	Middleware Package (CycloneTCP, CycloneSSL and CycloneCrypto) (Size : 6.6 MB)	1.7.6	
Details			
From L	ocal From Url Refresh Install Now Remo	ve Now Clo	ose

Figure 28. Embedded software pack release - Successful installation

3.4.6 Removing already installed embedded software packages

Proceed as follows (see figures 29 to 31) to clean up the repository from old library versions, thus saving disk space:

- 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 then opens to show the deletion status.



Embedded Software Packages Manager		X	
STM32Cube MCU Packages and embedded software packs releases			
Releases Information was last refreshed less than one hour ago.		+-	
STM32Cube MCU Packages Oryx-Embedded STMicroelectronics			
Description	Installed Version	Available Version	
► STM32F3			
▼ STM32F4			
STM32Cube MCU Package for STM32F4 Series	1.21.0	1.21.0	
STM32Cube MCU Package for STM32F4 Series	1.19.0	1.19.0	
Details			
STM32CubeF4 Firmware Package V1.19.0 / 29-december-2017 Main Changes • HAL CAN driver has been redesigned with new API's. • Support latest mbedTLS, LwIP and FatFs stacks.			
From Local From Url Refresh Install N	ow Remove N	low Close	

Figure 29. Removing libraries

Figure 30. Removing library confirmation message

Packages Manager	x
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 pa	ickage.
- FW.F4.1.19.0 (C:\Users\frq09031\STM32Cube\Repository\STM32Cube_FW_F4_V1.19.0)	
Please confirm package(s) deletion	
Yes No	

Figure 31. Library deletion progress window

Remove selected Fin	rmware	X
Preparing to delete selec	ted Files	
Preparing to delete selec	ted Firmware Packs	
	OK Car	icel

UM1718 Rev 46



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.

Check for update AIL-N New packages are available	Manage software installations
act from MCU	Check for STM32CubeMX and embedded software packages updates
SS TO MCU SELECTOR	Install or remove embedded software packages INSTALL / REMOVE

Figure 32. Updates are available

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:

- 1. 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



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 process		
	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		

Figure 33. Help menu: checking for updates



4 STM32CubeMX user interface

STM32CubeMX user interface comes with three main views the user can navigate through using convenient breadcrumbs:

- 1. the **Home** page
- 2. the **New project** window
- 3. 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 179*). 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

The Home page is the first window that opens up when launching STM32CubeMX (see *Figure 34*). 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.



MX STM32CubeMX Untitled		- 🗆 X
STM32 File	Window Help	🐵 🖪 🖻 🏏 🔆 🏹
Home >		
Existing Projects	New Project	Manage software installations
Recent Opened Projects	I need to :	Check for STM32CubeMX and embedded
CEC_DataExchange_Device_1.ioc Last modified date : 03/02/2021 10:44:13		CHECK FOR UPDATES
COMP_CompareGpioVsVrefInt_IT. Last modified date : 03/02/2021 10:35:42	- I	Install or remove embedded software pack
Other Projects	Start My project from ST Board ACCESS TO BOARD SELECTOR Start My project from Example ACCESS TO EXAMPLE SELECTOR	Explore STM32CubeNX capabilities and utorial videos utorial videos reserved reserved



4.1.1 File menu

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

Table 2. Home page shortcuts		
Home pag		

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 35</i>)	
Outputs	 MCUs selection tab that lists the MCUs of a given family matching the user criteria (series, peripherals, package,) when an MCU was selected last⁽¹⁾. 	
	 Outputs tab that displays a non-exhaustive list of the actions performed, raised errors and warnings (see <i>Figure 36</i>) found upon user actions. 	
Font size	Makes possible to change STM32CubeMX 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.



STM32CubeMX	Untitled: STM32F030K6	Tx										
M32 T	File		W	indow	יו	Help		(III		3 🖸	>*	: 5
Home /	STM32F030K6Tx	/ Untitle	d - 🗹	Outputs	figuration	i I			GENERA	TE CODE		
Pinout & C	Configuration	Clock	Config	uration		Proje	ect Manag	ger			Tools	
			Software			Pinout						
ptions Q	1		/		🖸 Pinout	view	System vie	9W				
Categories A->Z	Z	/		-								
•		/										
ADC		/										
CRC		/										
DMA							1					
FATES FREERTOS		/			-		223					
GPIO						57	222					
12C1						57	222					
IRTIM							-					
IWDG						BBBBBBB						
NVIC												
RCC RTC												
RCC												
RCC RTC		0	53	0	05	CI	718	_	0			
RCC RTC SPI1 SYS TIM1		Q	53	Q		4			Q		Ŷ	
RCC RTC SPI1 SYS		Q	[]	Q		4			٩		~	
RCC RTC SPI1 SYS TIM1 TIM3		Q	[]	Q		4			Q		~	
RCC RTC SPI1 SYS TIM1 TIM3 MCUs Selection	Output		[]	Q	_	4			۹		~	role.
RCC RTC SPI1 SYS TIM1 TIM3 ICUs Selection Serie	ies	Lines	53		Мси	4		Package	۹		equired Periphe	rals
RCC RTC SPI1 SYS TIM1 TIM1 TIM3 ICUs Selection Seri STM32F0	ies STM32F0x	Lines 0 Value Line	53	STM32F03	Mcu 0C6Tx	4	LQFP48		۹	None	equired Periphe	rais
RCC RTC SPI1 SYS TIM1 TIM3 ICUs Selection STM32F0 STM32F0	Ies STM32F0x STM32F0x	Lines 0 Value Line 0 Value Line	[]	STM32F030 STM32F030	Mcu 0C6Tx 0C8Tx	4	LQFP48 LQFP48		۹	None None	equired Periphe	rais
RCC RTC SPI1 SYS TIM1 TIM3 ICUs Selection STM32F0 STM32F0 STM32F0	ies STM32F0x STM32F0x STM32F0x STM32F0x	Lines 0 Value Line 0 Value Line 0 Value Line	[]	STM32F030 STM32F030 STM32F030	Mcu 0C6Tx 0C8Tx 0CCTx	4	LQFP48 LQFP48 LQFP48		۹	None None None	equired Periphe	rals
RCC RTC SPI1 SYS TIM1 ACUs Selection STM32F0 STM32F0 STM32F0 STM32F0	iés STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x	Lines 0 Value Line 0 Value Line 0 Value Line 0 Value Line	[]	STM32F030 STM32F030 STM32F030 STM32F030	Mcu 0C6Tx 0C8Tx 0CCTx 0F4Px	4	LQFP48 LQFP48 LQFP48 TSSOP20		۹	None None None None	equired Periphe	rais
RCC RTC SP11 SYS TIM1 MCUs Selection STM32F0 STM32F0 STM32F0 STM32F0 STM32F0 STM32F0	es STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x	Lines O Value Line O Value Line O Value Line O Value Line O Value Line	53	STM32F03 STM32F03 STM32F03 STM32F03 STM32F03	Mcu 0C6Tx 0C8Tx 0CCTx 0F4Px 0K6Tx	4	LQFP48 LQFP48 LQFP48 TSSOP20 LQFP32		Q	None None None None	equired Periphe	rais
RCC RTC RTC SPI1 SYS TIM1 CUS Selection Stm32F0 STM32F0 STM32F0 STM32F0 STM32F0 STM32F0 STM32F0	les STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x	Lines 0 Value Line 0 Value Line 0 Value Line 0 Value Line 0 Value Line 0 Value Line	[]	STM32F03 STM32F03 STM32F03 STM32F03 STM32F03 STM32F03 STM32F03	Mcu OC6Tx OC8Tx OCCTx OF4Px OK6Tx OK6Tx OR8Tx	4	LQFP48 LQFP48 LQFP48 TSSOP20		۵	None None None None	equired Periphe	rals
RCC RTC SPI1 SYS TIM1 TIM3 ACUs Selection	les STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x	Lines 0 Value Line 0 Value Line 0 Value Line 0 Value Line 0 Value Line 0 Value Line 0 Value Line	[]	STM32F03 STM32F03 STM32F03 STM32F03 STM32F03	Mcu OC6Tx OC8Tx OCCTx OF4Px OF4Px OR6Tx OR8Tx OR8Tx ORCTx	4	LQFP48 LQFP48 LQFP48 TSSOP20 LQFP32 LQFP64		٩	None None None None None None	equired Periphe	rais
RCC RTC SPI1 SPI1 SYS TIM1 TIM1 Series STM32F0 STM32F0	es STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x	Lines 0 Value Line 0 Value Line 0 Value Line 0 Value Line 0 Value Line 0 Value Line 0 Value Line 1	[]	STM32F030 STM32F030 STM32F030 STM32F030 STM32F030 STM32F030 STM32F030 STM32F030	Mcu OC6Tx OC8Tx OCCTx OF4Px DK6Tx OR8Tx OR8Tx ORCTx 1C4Tx	4	LQFP48 LQFP48 LQFP48 TSSOP20 LQFP32 LQFP64 LQFP64 LQFP64		٩	None None None None None None None None	equired Periphe	rais
RCC RTC RTC SPI1 SYS TIM1 TIM1 Serie STM32F0 STM32F0 STM32F0 STM32F0	ies STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x	Lines 0 Value Line 0 Value Line 0 Value Line 0 Value Line 0 Value Line 0 Value Line 1	[]	STM32F03/ STM32F03/ STM32F03/ STM32F03/ STM32F03/ STM32F03/ STM32F03/	Mcu OC6Tx OC8Tx OCCTx OCCTx OCCTx OF4Px OF4Px OR6Tx OR6Tx ORCTx 1C4Tx 1C6Tx	4	LQFP48 LQFP48 LQFP48 TSSOP20 LQFP32 LQFP64 LQFP64		٩	None None None None None None None	equired Periphe	rals
RCC RTC RTC SPI1 SYS TIM1 TIM3 Serie ACUs Selection Serie STM32F0 STM32F0	es STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x STM32F0x	Lines 0 Value Line 0 Value Line 0 Value Line 0 Value Line 0 Value Line 0 Value Line 0 Value Line 1 Value Line 1 1	[]	STM32F030 STM32F030 STM32F030 STM32F030 STM32F030 STM32F030 STM32F030 STM32F030 STM32F030	Mcu OC6Tx OC8Tx OCCTx OF4Px OR6Tx OR8Tx OR8Tx OR8Tx IC4Tx IC6Tx IC6Tx IC6Tx	4	LQFP48 LQFP48 LQFP48 TSSOP20 LQFP32 LQFP64 LQFP64 LQFP48 LQFP48		Q	None None None None None None None None	equired Periphe	rais

Figure 35. Window menu

Figure 36. Output view

MCUs Selection Output
Import Analysis: C:\Git MicroXplorer 4 P\MicroXplorer\microxplorer\src\main\resources\db\plugins\boardmanager\boards\A72 Nu
The Mcu (STM32F7222ETx) found in the Project being imported is not the same as the Mcu (STM32F030K6Tx) currently edited
Ø Import error: CORTEX M7 peripheral doesn't exist in STM32F030K6Tx
Import error: USB OTG FS peripheral doesn't exist in STM32F030K6Tx
Import :
Import RCC partly failed
⊗ error: Low Speed Clock (LSE) :Crystal/Ceramic Resonator mode doesn't exist in STM32F030K6Tx , it could not be importe
A GPIO_EXTI13 has been removed; it was locked on PC13 which is no longer existing
A GPIO_Output has been removed; it was locked on PG6 which is no longer existing
▲ GPI0_Input has been removed: it was locked on PG7 which is no longer existing
A GPIO_Output has been removed: it was locked on PB14 which is no longer existing
A Signal : SYS_JTDO-SWO cannot be imported.
A Signal : USB_OTG_FS_ID cannot be imported.
ASome parameters can't be imported for NVIC
Can't find in STM32F030K6Tx an interrupt equivalent to MemoryManagement_IRQn
Can't find in STM32F030K6Tx an interrupt equivalent to BusFault_IRQn
Can't find in STM32F030K6Tx an interrupt equivalent to UsageFault_IRQn
▲Interrupt SVCall_IRQn is replaced by interrupt SVC_IRQn
Can't find in STM32F030K6Tx an interrupt equivalent to DebugMonitor_IRQn
A Some parameters can't be imported for RCC
Can't import parameter:TIM Prescaler Selection, it doesn't exist in STM32F030K6Tx
Can't import parameter:Power Over Drive, it doesn't exist in STM32F030K6Tx
▲Can't import parameter:Power Regulatror Voltage Scale, it doesn't exist in STM32F030K6Tx
Importing project completed
$^{\otimes}$ Only compatible part of project is imported. Import RCC : FAILED; Set STM32F030K6Tx clock tree as 'DEFAULT'.



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 <i>Alt-V</i>	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
Updater Settings <i>Alt-S</i>	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. Help menu shortcuts

4.1.4 Social links

Developer communities on popular social platforms such as Facebook[™], Twitter[™], STM32 YouTube[™] channel, as well as ST Community can be accessed from the STM32CubeMX toolbar (see *Figure 37*).

Figure 37. Link to social platforms



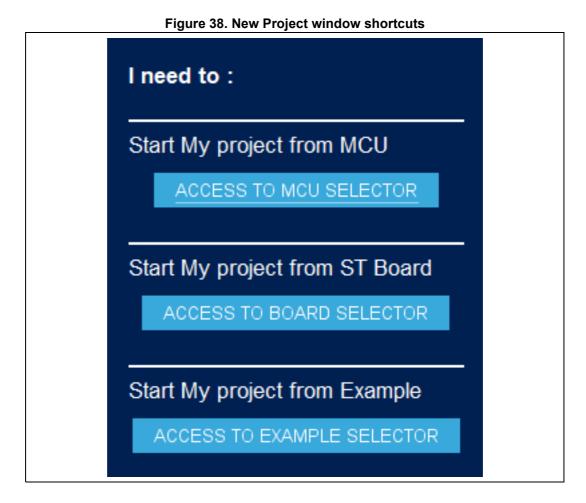
UM1718 Rev 46



UM1718

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 38*).



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 39*). The project is adjusted accordingly:

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

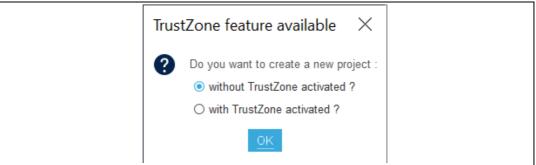


Figure 39. Enabling TrustZone

The selectors result view can be adjusted (see *Figure 40*):

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

Figure 40. Adjusting selector results

CUs/MF	PUs List: 1903 items	Display similar ite	ms				📤 Expo
*	Part No 🚔 Reference Marketin 🗡 Unit Price	e× Board >	Package	× Flash ≻	RAM	× io	× Freq.
☆	Pack All Columns		LQFP48	32 kBytes	4 kBytes	39	48 MHz
☆	Horizontal Scroll		LQFP48	64 kBytes	8 kBytes	39	48 MHz
☆	Remove Current Column		LQFP48	256 kBytes	32 kBytes	37	48 MHz
	Add Columns for Selected Criterias		TSSOP20	16 kBytes	4 kBytes	15	48 MHz
	Reset to Default Columns		LQFP32	32 kBytes	4 kBytes	25	48 MHz
☆	Part No	NUCL STM3	LQFP64	64 kBytes	8 kBytes	55	48 MHz
☆	✓ Reference		LQFP64	256 kBytes	32 kBytes	51	48 MHz
	Marketing Status		LQFP48	16 kBytes	4 kBytes	39	48 MHz
	✓ Unit Price for 10kU (US\$)		LQFP48	32 kBytes	4 kBytes	39	48 MHz
☆	Board		WLCSP25	32 kBytes	4 kBytes	20	48 MHz
☆	✓ Package		TSSOP20	16 kBytes	4 kBytes	15	48 MHz
	✓ Flash		TSSOP20	32 kBytes	4 kBytes	15	48 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 or number of I/Os (see *Figure 41*), and on their graphics capabilities as well.

UM1718 Rev 46



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

ew Pr. Reset search criteria	ria, ample Selector Cross	Selector		Quick access to	o resources						ting an MCU, project
ن 🖈 🖻 🖻 ک	\supset (Features	Block Diagram	Docs 8	Resources	CAD Resources	📔 Datasheel		📑 Buy) 🕞 🛛	tart Project
Commercial		STM32F0 Series									
Part Number	\$	STM32F030R8	BT6TR	Mainstream Arm C	ortex-M0 Value line	MCU with 64 Kbytes of Flash r	nemory, 48 M	MHz CPU			
Q ~	+-			Unit Price for 10kU (US\$):	4.0540						
PRODUCT INFO	~	ACTIVE Product is in mass production		one Price for Toko (0.34).	10042		LOFP 64	10x10x1.4 mm			
Marketing Status Price	<u> </u>	provides an overview of the cor	mplete range of STMS	32F030x4/x6/x8/xC periph	ierals proposed.	20 pins to 64 pins. Depending on the o					
Price Package Core Coprocessor		peripherals, gaming and GPS p	olatforms, industrial ap	pplications, PLCs, inverte		tions such as application control and t larm systems, video intercoms, and HV		nanonero equipri			
Package Core Coprocessor	> >	peripherals, gaming and GPS p	olatforms, industrial ap	pplications, PLCs, inverte				nanonero equipri		Export MCUs list to Excel	Export
Package Core Coprocessor	> >	peripherals, gaming and GPS p	olatforms, industrial ap	pplications, PLCs, inverte		larm systems, video intercoms, and Hv				Export MCUs list	
Package Core Coprocessor MEMORY	> >	peripherals, gaming and GPS p Features	olatforms, industrial ap Clo	pplications, PLCs, inverte	rs, printers, scanners, al natus X 1000 Proce for 10k 1.0542	Iarm systems, video intercoms, and Hv	ACs.		× u	Export MCUs list to Excel 8 Frequence 48 MHz	Export
Package Core Coprocessor MEMORY TIMER	> >	peripherals, gaming and GPS p Features s/MPUs List: 3542 items i Unit of the second Rest Mo ☆ STMACE/030R8151R ☆ STMACE/030R8151R	olatforms, industrial ap	pplications, PLCs, inverte se selector lay similar items Concord Amreshop st STM3 Active STM3 Active	rs, printers, scanners, al tatus X Unit Price for 10k 1.0542 1.691	arm systems, video intercoms, and HV <u>Board</u> <u>Peckage</u> LGPF 64 10x10x1 LGPF 64 10x10x1	ACs.	× RAM 8 kBytes 32 kBytes	× uc	Export MCUs list to Excel X Frequence 48 MHz 48 MHz	TTM32F0
Package Core Coprocessor MEMORY TIMER	> >	perpherals, gaming and GPS p Features stMPUs List: 3542 items Ω Δ	Cio	pplications, PLCs, inverte se selector lay similar items Beters Markeling st STM3 Active STM3 Active	rs, printers, scanners, al atus × <u>Veit Price for 10k</u> 1.0542 1.691	Arm systems, video intercoms, and HV Stand Stand Package LOFP 64 10x10x1 LOFP	ACs. 64 kBytes 256 kBytes 256 kBytes	8 kBytes 32 kBytes 32 kBytes	× uc 55 51 51	Export MCUs list to Excel 3 X Frequence 48 MHz 48 MHz 48 MHz	STM32F0 STM32F0
Package Core Coprocessor MEMORY TIMER AVALOG	> >	Perpherals, gaming and GPS p Features SMPUs List: 3542 items Composition of the second second second STM32F030RCT61R Composition of the second s	Cio	pplications, PLCs, inverte se selector lay similar items Beterce: Merceland State STM3. Active STM3. Active STM3. Active	rs, printers, scanners, al atus × Uut Price for 19k. 1.0542 1.691 1.3566	Beard Setting All Contents and HV Beard Content Conte	ACs.	× RAM 8 kBytes 32 kBytes 32 kBytes 4 kBytes	× 00 55 51 51 39	Export MCUs list to Excel 3 X Frequence 48 MHz 48 MHz 48 MHz 48 MHz	C Export STM32F0 STM32F0 STM32F0 STM32F0 STM32F0
Package Core Coprocessor MEMORY TIMER AUCLOG ADD Total Convertee: From 8 to 25		perpherais, gaming and GPS p Features MPUs List: 3542 Items Compared State State State Compared State State State Compared State State STM22F03RCT6TR Compared StateState STM22F03RCT6TR Compared StateStateState STM22F03RCT6TR	Cio Cio Partiko STM32F030R8 STM32F030RC	In the second se	rs, printers, scanners, al tatus X Unit Proce for 101 1.0542 1.691 1.691 1.3566 1.3566	Iam systems, video intercoms, and HV N Beard N Beard N LOPP 64 10x10x1 LOPP 64 10x10x1 LOPP 64 10x10x1 LOPP 64 10x10x1 LOPP 64 10x10x1 LOPP 64 10x10x1 LOPP 64 10x10x1 LOPP 64 10x10x1	ACs. 64 kBytes 256 kBytes 16 kBytes 16 kBytes	× RAM 8 kBytes 32 kBytes 32 kBytes 4 kBytes	× 000	Export MCUs list to Excel 3 Frequence 48 MHz 48 MHz 48 MHz 48 MHz 48 MHz	Export STM32F0 STM32F0 STM32F0 STM32F0 STM32F0 STM32F0
Package Core Coprocessor MEMORY MEMORY ADC Total Convertence From 0 to 25	> >	perpherals, gaming and GPS p Features stMPUs List: 3542 items stMA2r030R0T6TF stM32r031C4T6 stM32r031C4T6 stM32r031C4T6	Clo Clo STM32F030R8 STM32F031C4	In the second se	rs, printers, scanners, al atus X Unit Price for 10k. 1 0542 1 691 1 3566 1 3566 1 4169	arm systems, video intercoms, and HV Composition of the systems LOPP 54 toolsot. LOPP 54 toolsot. LOPP 54 toolsot. LOPP 54 Toolsot. LOPP 54 Toolsot. LOPP 54 Toolsot.	ACs. 64 kBytes 256 kBytes 256 kBytes 16 kBytes 32 kBytes	8 kBytes 32 kBytes 32 kBytes 4 kBytes 4 kBytes	× 000000000000000000000000000000000000	Export MCUs list to Excel 48 MHz 48 MHz 48 MHz 48 MHz 48 MHz 48 MHz 48 MHz 48 MHz 48 MHz	
Package Core Coprocessor MEMORY TIMER AUCLOG ADD Total Convertee: From 8 to 25		perperate, gaming and GPS p Features suPPUs List: 3642 energy distributions and personal person distributions and personal person distributions and personal personal distributions and personal distributions	Cio Cio Partiko STM32F030R8 STM32F030RC	bpplications, PLCs, inverte see selector lay similar items Borecal, Mitteding St STM3 Active STM3 Active STM3 Active STM3 Active STM3 Active STM3 Active STM3 Active STM3 Active	rs, printers, scanners, al atus 2 <u>ber Free tor 19k.</u> 1 0542 1 691 1 3566 1 3566 1 4169 1 4169	Iam systems, video intercoms, and HV Stands, video intercoms, and HV LOPP 64 (roote) LOPF 64 (roote) LOPF 64 (roote) LOPF 64 (roote) LOPF 64 (roote)	ACs. 54 KBytes 256 KBytes 16 kBytes 16 kBytes 16 kBytes 28 KBytes 28 kBytes	8 kBytes 32 kBytes 32 kBytes 4 kBytes 4 kBytes 4 kBytes	× uc 555 51 51 39 39 39 39	Export MCUs list to Excel 3 Mile 48 Mile 48 Mile 48 Mile 48 Mile 48 Mile 48 Mile 48 Mile	Control Contro Control Control Control Control Control Co
Package Coprocessor KEMORY MEMORY NULLOG ACC fotal (converting) From 0 to 25		Perpendieus gamma and GPS p Features Features Statutes S	Clo Clo STM32F030R8 STM32F031C4	se selector se selector by similar liens STM3. Active STM3. Active	rs, printers, scanners, al atus 5 (201 processor 10). 10542 10542 1591 13566 13566 14169 14169 15161	arm systems, video intercoms, and HV Iteration IOPP 54 10x06x1. IOPP 54 70x7x1 4.	ACs. 54 kBytes 256 kBytes 256 kBytes 16 kBytes 16 kBytes 32 kBytes 32 kBytes 32 kBytes	RAM B kBytes 32 kBytes 32 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes	× uc 55 51 39 39 39 39 39 39	Export MCUs list to Excel 48 MHz 48 MHz 48 MHz 48 MHz 48 MHz 48 MHz 48 MHz 48 MHz 48 MHz	Comparison C
Package Coprocessor KEMORY MEMORY NULLOG ACC fotal (converting) From 0 to 25		perpension Statures SumPlacities Statures SumPlacities Statures SumPlacities Statures SumPlacities Statures Statures Statures	Cio Cio Co Cio STM22F030RC STM32F030RC STM32F030RC STM32F031C6	ppikations, PLCs, inverte se elector International States STM3. Active STM3. Activ	rs, printers, scanners, at anto 31 outprice for 10x. 1.0542 1.551 1.3566 1.3566 1.4169 1.4169 1.5161	Lam systems, video intercore, and HV 1 Board 0 Core Advance 1 Core Advance Core Advance	ACs. 64 kBytes 256 kBytes 256 kBytes 16 kBytes 16 kBytes 32 kBytes 32 kBytes 32 kBytes	8 kBytes 32 kBytes 32 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes	× uc 55 51 39 39 39 39 39 39 39	Export MCUs list to Excel 3 NHz 48 MHz 48 MHz 48 MHz 48 MHz 48 MHz 48 MHz 48 MHz 48 MHz 40 MHz	
Poclage Cree Crprocessor MEMORY NAME ADD Table Conventional ADD Table Conventional ADD Table Conventional ADD Table Conventional From 0 to 25 Search criteria [5]		Perpendieus gamma and GPS p Features Features Statutes S	Clo Clo STM32F030R8 STM32F031C4	se selector se selector by similar liens STM3. Active STM3. Active	rs, printers, scanners, al atus 5 (201 processor 10). 10542 10542 1591 13566 13566 14169 14169 15161	arm systems, video intercoms, and HV Iteration IOPP 54 10x06x1. IOPP 54 70x7x1 4. IOPP 54 70x7x1 4. IOPP 54 70x7x1 4. IOPP 54 70x7x1 4. IOPP 44 70x7x1 4. IOPP 44 70x7x1 4.	ACs. 64 kBytes 256 kBytes 256 kBytes 16 kBytes 16 kBytes 32 kBytes 32 kBytes 32 kBytes	RAM B kBytes 32 kBytes 32 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes	× uc 55 51 39 39 39 39 39 39	Export MCUs list to Excel 48 MHz 48 MHz 48 MHz 48 MHz 48 MHz 48 MHz 48 MHz 48 MHz 48 MHz	Comparison C
Package Core Coprocessor MEMORY TIMER ANALOG AGC Total (convention From 8 to 25 Convention From 9 to 25 Convention From 9 to 25 Convention From 9 to 24 Convention C	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	perpherate.gamma and GPS p Features Features BMPUs Lat. 35/2 from C STMAP power to the second seco	Cio Cio Co Cio STM22F030RC STM32F030RC STM32F030RC STM32F031C6	by smlar files STM3. Active STM3. Active	rs, printers, scanners, al ana Y out the for the 1.0542 1.651 1.556 1.459 1.459 1.459 1.5567 1.556 1.556	3 Inter or office intercome, and HV 3 Control of the intercome, and HV LOPP 44 Intercome, and HV	ACs. 64 kBytes 256 kBytes 16 kBytes 16 kBytes 16 kBytes 32 kBytes 32 kBytes 32 kBytes 32 kBytes 32 kBytes 32 kBytes	8 kBytes 32 kBytes 32 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes	× rec 55 51 51 39 39 39 39 39 39 39 39 39 20	Export MCUs list to Excel 3 Mitz 48 Mitz	Comparing the second seco
Package Cre Coprocessor MEMOORY MEMOORY AND Collocation ACC Total Conventing From 8 to 25 ACC Total Conventing 10 ACC Total (Conventing 10 C) C 24 Add (Assessible From 1 to 42)	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	perpension gamma GPS y Features	Cio Cio Co STM32F03R8 STM32F03R8 STM32F03R6 STM32F03R6 STM32F031C6	se selector see selector in similar items into similar items into similar items into a Active STIM3. Active	rs, printers, scanners, al antia X org. Proc. for 108-c 1.6542 1.654 1.556 1.459 1.4169 1.5161 1.5161 1.5161 1.5161 1.5161	Immediate Annotation Immediate Annotation Immediate Immediate Immediate Immediate <td>ACS.</td> <td>8 kBytes 32 kBytes 32 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes</td> <td>× uc 55 51 39 39 39 39 39 39 39 39 39 39 39 39 39</td> <td>Export MCUs list to Excel 3 3 Area 48 MHz 48 MHz</td> <td>Comparison Comparison C</td>	ACS.	8 kBytes 32 kBytes 32 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes	× uc 55 51 39 39 39 39 39 39 39 39 39 39 39 39 39	Export MCUs list to Excel 3 3 Area 48 MHz 48 MHz	Comparison C
Package Core Coprocessor MEMORY MEMORY ADD Color	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	Perphenais, gamma and GPS p Features SNPUs Lat: 542 terms SNPUs Lat: 542 terms SND2POSECTER SND2POSECT	Cio Cio Co STM32F03R8 STM32F03R8 STM32F03R6 STM32F03R6 STM32F031C6	se selector se selector se selector set selector statutor strike selector strike selector stri	rs, printers, scanners, at 10542 10542 10542 10541 10561 11566 11459 11459 11561 11561 11561 11561 11561 11561 11565 03949 03949	Immediate	ACS. 64 kBytes 55 kBytes 255 kBytes 255 kBytes 16 kBytes 32 kBytes 32 kBytes 32 kBytes 32 kBytes 32 kBytes 32 kBytes 32 kBytes 32 kBytes	8 kBytes 32 kBytes 32 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes 4 kBytes	≥ 55 55 51 39 39 39 39 39 39 20 15 15	Export MCUs Bit to Excel 40 MHz 40 MHz	Comparing Strategy Strate
Package Coprocessor Coprocessor MEMORY TIMER ARACO Construction Constr	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	Perpherais, gamma and GPS p Fatures MAPUs Lat: 35/2 arms SINDAPOSICIE SINDAPOSIE SIND	Cio Cio Co STM32F03R8 STM32F03R8 STM32F03R6 STM32F03R6 STM32F031C6	se selector se selector by seniar tens by seniar te	Image: scanners, all ante 3 ⁻¹ out true for 10x- 1.0542 1.654 1.654 1.556 1.1566 1.4169 1.561 1.561 1.561 1.561 1.561 1.561 1.561 1.561 1.561	Immediate Control Control	ACs.	8 Bbjets 3 kBytes 32 kBytes 32 kBytes 4 Bytes 4 Byte	× 0 000 551 51 39 39 39 39 39 39 20 15 15 15	Export MCUs list to Excel 3 Arragement 48 MHz 48 MHz	Comparison of the second

Figure 41. New Project window - MCU selector

Export to Excel feature

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

Show favorite MCUs feature

Clicking the 🕎 icon for an MCU from the list marks it as favorite, see *Figure 42*.



MCU/MPU Filters												
	vorite item	ns	Features	Block Dia	ıgram Doc	s & Resource	es CAD Resou	urces	📑 Datasheet	📑 Buy	G→ St	art Project
Commercial	~		STM32U5 Se	ries								
Q	+ -	*	STM32	U585AII			er with FPU Ar ytes of Flash n		tex-M33 with Tru /	ust Zone,	MCU 16	0
PROPUST NEO	_ 		ACTIVE		Unit Pr	ice for 10kU (l	US\$):7.7937					
PRODUCT INFO			Product is production		Boards	: <u>B-U585I-IOT</u>	02A - STEVAL-STWI	NBX1	UFBGA	169 7x7x0.6 F	° 0.5 mm	
Segment	>											
Series	Clic	k colu	mn						family (STM32U5 🕐			
		k colu	ımn	Arm [®] Co	ortex [®] -M33 32-	bit RISC cor	re. They operate a	at a freq	uency of up to 160		ick to ex	
Series Line Marketing Status	to gro	ck colu oup fav s at the	ımn vorite	Arm [®] Co -M33 core	ortex [®] -M33 32-	-bit RISC cor ngle-precisio	re. They operate a on FPU (floating-p	at a freq				kport to Excel
Line	to gro	oup fav	ımn vorite	Arm [®] Co -M33 core	ortex [®] -M33 32- e features a si	-bit RISC cor ngle-precisio	re. They operate a on FPU (floating-p	at a freq	uency of up to 160			
Line Marketing Status	to gro	oup fav s at the	ımn vorite	Arm [®] Co -M33 core sing instru	ortex [®] -M33 32- e features a si	-bit RISC cor ngle-precisio the data typ	re. They operate a on FPU (floating-p	at a freq point unit	uency of up to 160			
Line Marketing Status Price	to gro	s at the	imn vorite e top	Arm [®] Co -M33 core sing instru	ortex [®] -M33 32- e features a sin actions and all	bit RISC cor ngle-precisio the data typ	re. They operate a on FPU (floating-p les. Display similar iten	at a freq point unit	uency of up to 160	the M	CU list	Export
Line Marketing Status Price Package Core	to gro	s at the	Imn vorite e top /MPUs List: 3	Arm [®] Co -M33 core ;ing instru 3542 items Part No	ortex [®] -M33 32- e features a sin actions and all Reference	bit RISC corngle-precision the data typ	re. They operate a on FPU (floating-p es. Display similar iten	ns	uency of up to 16(i), that supports al	the M × Flash ×		Export
Line Marketing Status Price Package Core Click star icon	to gro MCU:	s at the	Imn vorite e top /MPUs List: 3	Arm [®] Co -M33 core sing instru 3542 items Part No	e features a sin ictions and all Reference	bit RISC cor ngle-precisio the data typ Marketing Active . Active	re. They operate a on FPU (floating-p les. Display similar iten <u>Unit Price for ×</u>	ns Bo×L U	Package	the M	CU list	Export Export 100 MIPZ 136 160 MHz
Line Marketing Status Price Package Core	to gro MCU:	s at the CUs STI STI	IMN vorite e top /MPUs List: 3 mercial M32U585AII6	Arm [®] Co -M33 core ;ing instru 3542 items Part No	Reference STM32U585	bit RISC cor ngle-precisic the data typ Active Active Active Active	re. They operate a on FPU (floating-p les. Display similar iten <u>Vinit Price for ×</u> 7.7937	at a freq point unit ns Bo× U B S U	Package Pac	Ke M X Flash IIII 2040 K 2048 k	CU list	Export Export 100 MIPZ 136 160 MHz
Line Marketing Status Price Package Core Click star icon	to gro MCU:	CUs CUs CUs STI STI STI	Imn vorite e top /MPUs List: 3 mscusius M32U585All6 M32U585Al	Arm [®] Co -M33 core ;ing instru 3542 items Part No	rtex [®] -M33 32- e features a si ictions and all STM32U575 STM32U585 STM32U585	bit RISC cor ngle-precisic the data typ Active Active Active Active Active	re. They operate a on FPU (floating-p les. Display similar iten <u>Unit Price for X</u> 7.7937 7.7937	ns Bo× U B.S.U	uency of up to 16(i), that supports at <u>Processor</u> are read 20220X 1.4 II FBGA 169 7X7X0.6 P FBGA 169 7X7X0.6 P	Flash × 2048 k 2048 k 2048 k	CU list	Export Export 136 160 MHz 133 160 MHz 137 160 MHz
Line Marketing Status Price Package Core Click star icon to save MCU as favorite MEMORY		CUs CUs CUs CUs STI STI STI STI STI	Imn vorite e top //MPUs List: 3 mmercial M32U585All6 M32U585All6 M32U585Al M32U585Cl	Arm [®] Co -M33 core ing instru 3542 items Part No STM32U5	Reference STM32U585 STM32U585 STM32U585 STM32U585 STM32U585	bit RISC con ngle-precisio the data typ Active Active Active Active Active	re. They operate a on FPU (floating-p les. Display similar iten <u>v Unit Price for ×</u> 7.7937 7.7937 6.574	at a freq point unit	Package arP Package arP Package arP Package FBGA 169 7X7X0.6 P FBGA 169 7X7X0.6 P PFBGA 169 7X7X0.6 P	the M	CU list	Export Export C Export C Export Export C Export Expor
Line Marketing Status Price Package Core Click star icon to save MCU as favorite		CUs CUs CUs CUs STI ★ STI ★ STI ★ STI ★ STI ★ STI	Imn vorite e top //MPUs List: 3 mmercial M32U585All6 M32U585All6 M32U585Al M32U585Cl	Arm [®] Co -M33 core ing instru 3542 items Part No STM32U5	Reference STM32U855. STM32U855. STM32U585. STM32U585.	bit RISC con ngle-precisio the data typ Active Active Active Active Active Active Active	re. They operate a on FPU (floating-p less. Display similar iten <u>Vunit Price for X</u> 7.7937 7.7937 6.574 6.574	at a frequenciation of the second sec	uency of up to 16(), that supports al and 144 20X0X 1.4 m FBGA 169 7x7x0 6 P 2FP 48 7x7x1.4 mm 2FP 48 7x7x1.4 mm	the M Flash × 2048 k 2048 k	CU list	Export Ex
Line Marketing Status Price Package Core Click star icon to save MCU as favorite MEMORY		S at the CUs: CUs: ★ STI	Imn vorite e top //MPUs List: 3 mmercial M32U585All6 M32U585All6 M32U585Cl M32U585Cl M32U585Cl	Arm [®] Co -M33 core sing instru 3542 items Part No STM32U5	Reference STM32U585. STM32U585. STM32U585. STM32U585. STM32U585.	bit RISC corngle-precisic the data typ	re. They operate a on FPU (floating-p ees. Display similar iten (1.133) 7.7937 7.7937 6.574 6.574 7.0342	at a freq point unit ns Bo× U B.S.U L(U U U U U U U U U	uency of up to 16(), that supports al are <u>Package</u> FBGA 169 7x7x0.6 P FBGA 169 7x7x0.6 P 2FP 48 7x7x1.4 mm FQFPN 48 7x7x0.55	the M	CU list	Export Export 136 160 MHz 133 160 MHz 133 160 MHz 133 160 MHz 133 160 MHz 134 160 MHz 135 160 MHz
Line Marketing Status Price Package Core Click star icon to save MCU as favorite MEMORY Flash From 0 to 4096 (kBytes)		>up fav s at the >cus. Cus. ★ STI ★ STI	Imn vorite e top //MPUs List: : //MPUs List: : //M2U585AII6 //M32U585AII //M32U585CI //M32U585CI //M32U585CI //M32U585CI //M32U585CI///MAUSI	Arm® Ca -M33 core ing instru 3542 items Part No STM32U5 STM32U5	Reference STM32U585. STM32U585. STM32U585. STM32U585. STM32U585. STM32U585. STM32U585. STM32U585.	bit RISC cor ngle-precision the data typ Merketing- Active Active Active Active Active Active Active	re. They operate a on FPU (floating-p ees. Display similar item × <u>Unit Price for×</u> 7.7937 7.7937 6.574 6.574 6.574 7.0342 6.574	at a freq point unit ms Bo× U B.S.U L(U U U U U U U U U U U U	uency of up to 16f), that supports al urp 144 20X20X.1.4 If PEGA 169 7X70.6 P PEGA 169 7X70.6 P DFP 48 7X7X1 4 mm 2FP 48 7X7X1 4 mm 2FP 48 7X7X1 4 mm	the M Flash × Lash × 2048 k 2048 k	CU list	Export Ex



4.2.2 Board selector

The **Board selector** enables filtering on STM32 board types, series and peripherals (see *Figure 43*). 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', i.e. they cannot be moved automatically by STM32CubeMX constraint solver (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.

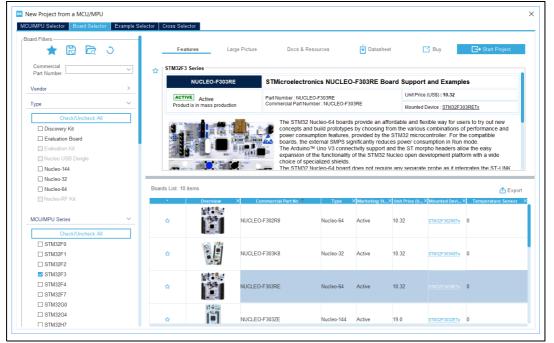


Figure 43. New Project window - Board selector

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 for 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 44*).

🌟 🔂 🖾 O															
Name IWD ~		Features			Start Project										
Keyword	→ ☆[rojects/NUCLEO-G070RB/Examples/IWDG/IWDG_I	Reset/												
				IWDG_Reset											
Vendor	>	STM32G0			V1.	3.0									
Board	~	FW pack			•••										
	_	STM32Cube_FW_G0_V1.3.0	Board	NUCLEO-G070RB		Mounted device									
Name	-	CubeMX 5.2.0		1111年二十		STM32G070RBT	<u>(</u>								
Туре	~	Toolchain/IDE				LQFP64									
	_	EWARM, MDK-ARM, SW4STM32		and the Ass											
Check/Uncheck All		Keywords													
Discovery Kit															
Evaluation Board															
□ Nucleo-144	Examp	les List: 7 items						🖒 Expo							
		Name 🗢	Board	× Board Type	× Series	× Project Type		× CubeMX vers							
□ Nucleo-144	2 2	Name A IWDG_RefreshUntilUserEvent	NUCLEO-F411RE	Nucleo-64	STM32F4	Example	LL	× CubeMX vers							
□ Nucleo-144 □ Nucleo-32		Name * WDG_RefreshUntilUserEvent WDG_RefreshUntilUserEvent_Init	NUCLEO-F411RE NUCLEO-G070RB	Nucleo-64 Nucleo-64	STM32F4 STM32G0	Example Example	LL LL	X CubeMX vers							
□ Nucleo-144 □ Nucleo-32	· ☆ ☆ ☆	Name C 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		CubeMX vers							
□ Nucleo-144 □ Nucleo-32 ☑ Nucleo-64	2 2 2 2 2 2 2 2 2	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	× CubeMX vers NA 5.2.0 5.2.0 5.2.0							
□ Nucleo-144 □ Nucleo-32 ☑ Nucleo-64	* * * *	None Number Numb	NUCLEO-F411RE NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G070RS 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	CubeMX vers NA 5.2.0 5.2.0 5.2.0 5.2.0 5.2.0							
Nucleo-144 Nucleo-32 ☑ Nucleo-64 MCU / MPU	2 2 2 2 2 2 2 2 2	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	× CubeMX vers NA 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_RefreshUntilUserEvent_Init WDG_Reset WDG_Reset WDG_Reset	NUCLEO-F411RE NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G070RB	Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64	STM32F4 STM32G0 STM32G0 STM32G0 STM32G0 STM32G0 STM32G0	Example Example Example Example Example Example	LL LL LL HAL HAL HAL	CubeMX vers 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 Wucleo-64 MCU / MPU Name	· · · · · · · · · · · · · ·	Name * WDG_RefreshUntilUserEvent WDG_RefreshUntilUserEvent_Init WDG_RefreshUntilUserEvent_Init WDG_Reset WDG_Reset WDG_Reset	NUCLEO-F411RE NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G070RB	Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64	STM32F4 STM32G0 STM32G0 STM32G0 STM32G0 STM32G0 STM32G0	Example Example Example Example Example Example Example	LL LL HAL HAL HAL HAL	X CubeMX vers 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							
Nucleo-144 Nucleo-32 Wucleo-64 MCU / MPU Name	· · · · · · · · · · · · · ·	Name * WDG_RefreshUntilUserEvent WDG_RefreshUntilUserEvent_Init WDG_RefreshUntilUserEvent_Init WDG_Reset WDG_Reset WDG_Reset	NUCLEO-F411RE NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G070RB	Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64	STM32F4 STM32G0 STM32G0 STM32G0 STM32G0 STM32G0 STM32G0	Example Example Example Example Example Example Example	LL LL HAL HAL HAL HAL ON is set w	X CubeMX vers 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							
Nucleo-144 Nucleo-32 Nucleo-64 MCU / MPU Name Series Check/Uncheck All	· · · · · · · · · · · · · ·	Name * WDG_RefreshUntilUserEvent WDG_RefreshUntilUserEvent_Init WDG_RefreshUntilUserEvent_Init WDG_Reset WDG_Reset WDG_Reset	NUCLEO-F411RE NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G070RB	Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64	STM32F4 STM32G0 STM32G0 STM32G0 STM32G0 STM32G0 STM32G0	Example Example Example Example Example Example Example A versi can be	LL LL HAL HAL HAL HAL ON is set w loaded in 1	X CubeMX vers 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 ben the exam STM32CubeM							
Nucleo-144 Nucleo-32 Nucleo-64 MCU / MPU Name Series Check/Uncheck All STM32F4	· · · · · · · · · · · · · ·	Name * WDG_RefreshUntilUserEvent WDG_RefreshUntilUserEvent_Init WDG_RefreshUntilUserEvent_Init WDG_Reset WDG_Reset WDG_Reset	NUCLEO-F411RE NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G070RB	Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64	STM32F4 STM32G0 STM32G0 STM32G0 STM32G0 STM32G0 STM32G0	Example Example Example Example Example Example Example A versi can be If not a	LL LL HAL HAL HAL HAL On is set w loaded in vailable, th	X CubekX vers NA 52.0 52.0 52.0 52.0 52.0 52.0 52.0 52.0							
Nucleo-144 Nucleo-32 Nucleo-64 MCU / MPU Name Series Check/Uncheck All Check/Uncheck All STM32F4 STM32F7	· · · · · · · · · · · · · ·	Name * WDG_RefreshUntilUserEvent WDG_RefreshUntilUserEvent_Init WDG_RefreshUntilUserEvent_Init WDG_Reset WDG_Reset WDG_Reset	NUCLEO-F411RE NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G070RB	Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64	STM32F4 STM32G0 STM32G0 STM32G0 STM32G0 STM32G0 STM32G0	Example Example Example Example Example Example Example A versi can be If not a solely I	LL LL HAL HAL HAL HAL On is set w loaded in vailable, th	CubeXX vers 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 c.explore 6.2.0 5.2.0 5.2.0 5.2.0 5.2.0 e.explore e.example cc with one of th							
Nucleo-144 Nucleo-32 Nucleo-64 MCU / MPU Name Series Check/Uncheck All STM32F4	· · · · · · · · · · · · · ·	Name * WDG_RefreshUntilUserEvent WDG_RefreshUntilUserEvent_Init WDG_RefreshUntilUserEvent_Init WDG_Reset WDG_Reset WDG_Reset	NUCLEO-F411RE NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G070RB NUCLEO-G071RB NUCLEO-G070RB	Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64 Nucleo-64	STM32F4 STM32G0 STM32G0 STM32G0 STM32G0 STM32G0 STM32G0	Example Example Example Example Example Example Example A versi can be If not a solely I	LL LL HAL HAL HAL HAL On is set w loaded in vailable, th	CubeXX vers 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 c.explore 6.2.0 5.2.0 5.2.0 5.2.0 5.2.0 e.explore e.example cc with one of th							

Figure 44. 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 45*):

- 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)



UM1718 Rev 46

K Start Project from Example		×
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	Ē
Open with:	STM32CubeMX ~ STM32CubeMX EWARM MDK-ARM STM32CubeIDE Explorer	

Figure 45. 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 46*).

	Help		🐵 F 🖻 🎽 🗘 🔆 🏹
	Help	F1	
	About	Alt-A	
	Docs & Resources	Alt-D	
our Droiset	Tutorial Videos	Alt-V	
ew Project	Refresh Data	Alt-R	ge software installations
	User Preferences		
	Check for Updates	its for N	MCUs, Boards and Examples
	Manage embedded software packages		
I need to :	Updater Settings	Alt-S	
			CHECK FOR UPDATES
Start My pr	oject from MCU		
	S TO MCU SELECTOR	Ins	nstall or remove embedded software packages
Start My pr	oject from ST Board		

Figure 46. Cross selector - Data refresh prerequisite

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 47*). A part number can also be entered partially: STM32CubeMX proposes a list of matching products (see *Figure 48*).

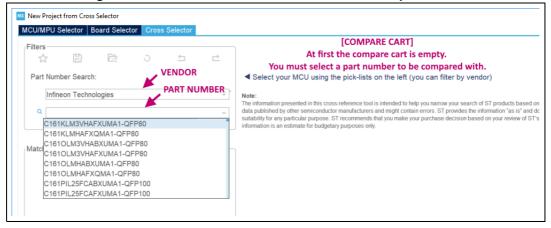
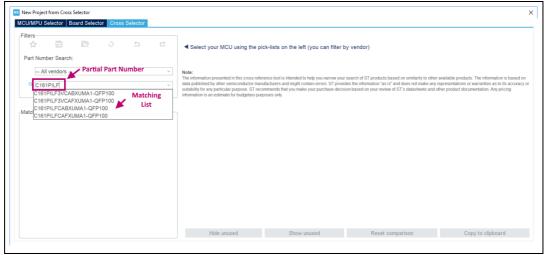


Figure 47. Cross selector - Part number selection per vendor





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 49*).



New Project from Cross Selector		[COMPARE CA	RT]		
Filters	omparing D17618ABG	Criteria MCU to compare w	utions	ST three closest	matches
☆ 🖻 🗟 ୬ 🛥 ല Part Number Search:	Used ? Importan	Category Parametric D17618ABGN100V	STM32H750IBKx G+	STM32F207ICHx 🕞	STM32F207IEHx C+
Renesas Electronics V	Co.	Product Change the criteria importan	COLONIAL (INC. 10K)	5.508 USD (for 10K)	6.063 USD (for 10K)
D176 Select one or more part numbers	•••	this will affect the matching		32 bit	32 bit
to add to compare cart	 0	System C Level 0 : Nice to have	7 at 400 MHz	ARM Cortex-M3 at 120 MHz	ARM Cortex-M3 at 120 MHz
Matching ST candidates (500) Part number Match	0	System C Level 1 : Should have Level 2 : Must have		BGA176	BGA176
STM322H750/BKX 95 %	· ·· ·	System Care and the second		140 io	140 io
STM332F217/EHX 94 % STM332F217/EHX 94 % STM332F217/EHX 94 %	•	Click to set to unused criteria that are not relevant	-40 °C to 105 °C	-40 °C to 85 °C	-40 °C to 85 °C
STN 35E 4339 EYX STN 35E 4739 EYX STN 35E 4745 EXX STN 35E 4745 EXX	0		1.62 V to 3.60 V	1.80 V to 3.60 Results	1.80 V to 3.60 V
81 N32E 2461 ETX ST M32E 2471 ETX ST M32E 2471 ETX 84 %	• ••	System Core RAM 4 KB	1024 KB	128 KB	128 KB
51M32E427(GHx 94 % 51M32E427(GHx 94 % 51M32E429(GHx 94 %	\bigcirc	No comparison	5	no	no
51M32E745IGKx 94 % 51M32E490IGHx 93 % 51M32E439IGHx 93 %		when feature is not present or when information is unavailable	:8 KB	256 KB	512 KB
ST 11/372 + 783 CK2 ST 11/372 + 783 CK2 ST 11/372 + 273 CH2 ST 11/372 + 273 CH2 ST 11/372 + 273 CH2 ST 11/372 + 273 CH2 ST 11/372 + 783 CH2	0	for the MCU to compare with	3	no	[©] Copy then past
	0	·	ADC 24-bit 32xADC 16-bit	24xADC 12-bit	24xJ Cart view
	Hide un	used Show unused		Reset comparison	Copy to clipboard

Figure 49. 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 Figure 50).



76/557



New Project from Cross Selector ICU/MPU Selector Board Selector			1704048				Click to create a n with this part i		
Filters		Comparing L	17618AB	GN100V by I	Kenesas Electro	nics with STMicroele			
☆ 🔂 🗟 🕻		Used ?	mportan	. Category	Parametric	D17618ABGN100	V STM32H750IBKx C+	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
STM32H750IBKx STM32E207ICHx STM32E207IEHx	95 % 94 %		-00	System Core	GPIO	78 io	138 io	140 io	140 io
Part number	864400000000000000000000000000000000000			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			_					
STM32F469IEHx STM32F746IEKx STM32F217IGHy	MCU/MPU Selector Board	I Selector 0	Cross Sele	tor					
51W32E427IGHx 51W32E417IGHx 51W32E429IGHx 51W32E429IGHx	MCU/MPU Filters	<u>ක</u> ා		Fe	atures Bl	ock Diagram De	ocs & Resources 📑 D	atasheet 📑 Buy	E+ Start Project
STM32F745IGKx STM32F469IGHx	Part Number Search		~	_					
311/03/27/29/06/X 11/03/27/26/66/X 11/03/27/20/06/X 11/03/27/11/X 11/03/27/11/X 11/03/27/20/06/X 11/03/20/06/X 11/03/20/00	Q STM32H750IE	v v		☆ [STM3	2H750IB				
21032E7570CKx 51M32E7671GKx 51M32E7671GKx 51M32E42911Hx	Core		~	MCUs/MP	'Us List: 1 item		+ Display similar ite	ms	
STM32E43711Hx STM32E4691Hx STM32E4391Hx STM32E4391Hx	Check/Uncheck All				Part No Refer M32H7 STM32		O GFX S. CORDIC DDR DEB B 1960 0 1	II FMAC HDP HMAC PK	A PWR RF SHA TAMP 0 0 0 0
STM32F765 Kx STM32F767 Kx									

Figure 50. 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 51*), 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).



Figure 5	1. STM32CubeMX Mair	i window upon MC	U selection	
STM32CubeMX Untitled: STM32F439VITx				
STM32 File	Window	Help	🐵 F 🕨) 🎽 🔆 🏹
Home / STM32F439VITx /	Untitled - Pinout & Configuration	on l	GENERATE COL	DE
Pinout & Configuration	Clock Configuration	Project Manager		Tools
_		Pinout		
Options Q Categories A->Z	🗿 Pi	nout view III System view		Į.
System Core >		985 985 985 985 905 905 905 906 906 906 906 906 906 906 906 906	24 KK	
Analog >			VSS	
Timers >	255 255 288		PA13 PA12 PA11	
Connectivity >	201	_	PA0 PA0 PA8	
Multimedia >	222 223 244 245 245 245 245 245 245 245 245 245		PC6 PC7	
Security >	941. 965. 2001 2011 2011		P06 P015 P014	
Computing >	PC2 PC2 PC3	STM32F439VITx	243 242 243 243 243 243 243 243 243 243 243 243 243 243 243 243 243 243 243 243 244 245	
Middleware >	255 255 272	LQFP100	P09 P08	
Application >	2001		P814 P813	
	A State of the sta	2010 2010 2010 2010 2010 2010 2010 2010	200	
	Q [] Q [i 🕘 💷 📑	Q	$\overline{}$
Molla Calcular			- • L	
MCUs Selection Output Series	Lines M	lcu Pack:	200	Required Peripherals
STM32F4 STM32F429	/439 STM32F429AGHx	UFBGA169	None	
STM32F4 STM32F429/ STM32F4 STM32F429/		UFBGA169 LQFP208	None	
0111021428	STINDE 420DETA	I SAME AND		

Figure 51. 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 52*).

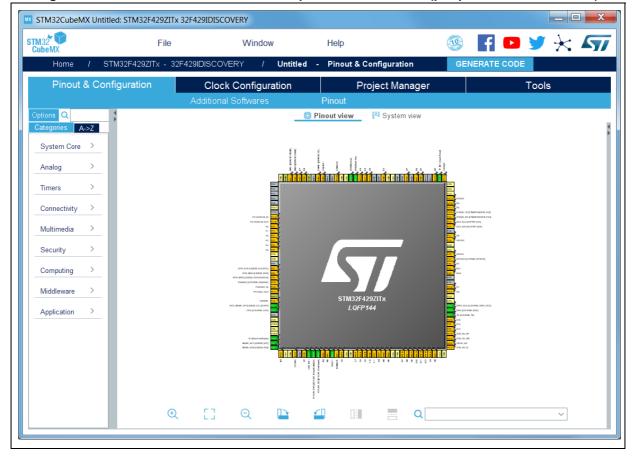


Figure 52. STM32CubeMX Main window upon board selection (peripherals not initialized)



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 53*).



Figure 53. STM32CubeMX Main window upon board selection (peripherals initialized with default configuration)

4.4 Boot chain (STM32MPUs)

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 54*)

Install or 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 PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 20/05/2024 22:28:37 PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 22/05/2024 18:22:12 PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 22/05/2024 18:22:12 PL0_Nucleo_NUCLEO-WL5 PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 22/05/2024 18:21:47 PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 22/05/2024 18:21:47 PL0_Nucleo_NUCLEO-WL5 MX	Existing Projects	New Project	Manage software installations
PLO_NUCLEO_NUCLEO-WL5 MX Last modified date : 20/05/2024 22:28:57 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 18:22:42 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 PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 22/05/2024 18:22:12 PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 22/05/2024 18:21:47 PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 22/05/2024 18:21:47 PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 22/05/2024 18:21:47 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			Install or remove embedded so INSTALL / REMOVE
Last modified date : 22/05/2024 18:21:47 PLO_NUCLEO_WLCS MX Start My project from Example ACCESS TO EXAMPLE SELECTOR		OK ACCESS TO BOARD SELECTOR	
PL0_Nucleo_NUCLEO-WL5 MX			
	PL0_Nucleo_NUCLEO-WL5 MX Last modified date : 20/05/2024 22:27:53		Here 402 MD: hostfant: MCT with mean performance, inclusivity and security

Figure 54. Project choice interface

• Six contexts are created in the configuration panel (see *Figure 55*)

Figure	55.	Contexts
--------	-----	----------

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

- 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)



Home > STI	M32MP2	51FAlx) Uni	titled - P	inout & (Configura	ation >									
P	inout 8	k Config	uration	h			Clock Configuration RIF									
							✓ Software Packs ✓ Pinout									
Q				~		٢	ADC1 Mode and Configuration									
Categories A->	>Z						Mode									
		A35S	A35S	A35NS	A35NS	M33NS	Boot time: Runtime contexts:									
÷	A35 R			(U-Boot)		(Cube)	A35S A35S A35NS A35NS M33NS									
ADC1				· · ·			A35 ROM (TF-A_BL2) (OP-TEE) (U-Boot) (Linux) (Cube)									
ADC2	-			~												
ADC3							□ IN0 Single-ended									
ADF1							IN1 IN1 Differential									
BOOT																
 BSEC 	~	~	~				IN2 Disable									
 CORTEX_M33 			1		1		IN3 Disable V									
							IN4 Disable V									
CRYP1		~			V		IN5 Disable									
							□ IN6 Single-ended									
CSI					V		- •									
DCACHE						V	IN7 Single-ended									
DCMI					V		Configuration									
DCMIPP					~		Reset Configuration									
DDR_CTRL		~	\checkmark													
DEBUG							Parameter Settings GIC Settings Ø DMA Settings Ø GPIO Settings									
DSIHOST		_		~	~											
DTS							Search Signals									
ETH1							Search (Ctrl+F)									
FILEX							Pi 🍨 Signal Pin C GPIO GPIO Maxi Retime Invert Doubl Delay Delay Invert U									
FMC		_					ANA0 ADC1 A35N Analo n/a n/a n/a n/a n/a n/a n/a									
GIC GPIO			1		~		ANA1 ADC1 A35N Analo n/a n/a n/a n/a n/a n/a n/a									
HASH	~	~	~													
HDP																
HPDMA1				2		2										
HPDMA2		~	~	~												
HPDMA3			~	~												

Figure 56. IPs interface assignment

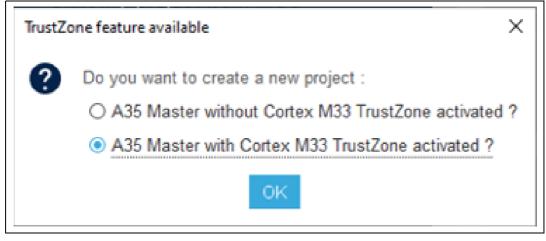
After assigning the IPs context go to "Project Manager" view, save the project, and generate the code.

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 57*)





UM1718 Rev 46



• Six contexts created in the configuration panel (see *Figure 58*)

Boot time:		Runtime contex	ds:				
A35 ROM	A35S (TF-A_BL2)	A35S (OP-TEE)	A35NS (U-Boot)	A35NS (Linux)			
				V			

Figure 58. 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 59. Assign IP context

ome 🗡	STM32MP251	FALx Vintitled - Pinou	it & Configu	ration /															GENER	RATE COD	DE
	Pinout & C	Configuration		Clo	k Confi	guratio	ı		RIF Project Manager							r	Tools				
	~						0			ADC1	Mode and Cr	onfiguration				1		🖸 F	Pinout view	U Sy	stem view
itegories	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:		Runtime conte	A35NS	A35N		1335	M33NS	. 1					
ADC1		(11-14_662)		(0-8001)	(cinox)	((1*****)	(Code)	A35 ROM	A35S (TF-A_BL2)	(OP-TEE)	(U-Boot)	A35N (Linu		133S (F-M)	(Cube)						
DC2																					
								INO Single-		0		_		<u> </u>							
								IN1 IN1 Differen	tial						· · · · ·		E				9000
BSEC		2				2		IN2 Disable							· · · · ·						
CORTEX							-	IN3 Disable								7 🚺	20				
CRC								IN4 Disable								ă 📗	C				
CRYP1								IN5 Disable								-					
																<u> </u>	9				
CSI								IN6 Single-e	nded								00				
DCACHE								IN7 Single-e	nded								0				
DCMI					~			IN8 Single-e	nded								(iii)				
DCMIPP											Configurati	on					9				
DDR CTR.		2							_								2				
DEBUG								Reset Configura	tion												
					12			Parameter Set	tings 🛛 😔 GI(Settings 🛛 🥥	DMA Setting	s 😔 GPN	O Settings			- 1					
DTS																	G				
ETH1								Search Signals									5				
								Search (Ctrl+F)						□ Sho	w only Modifie	d Pins	1.00	Rest Rest	NED 0 400	A CT	
FMC									-										VF BGA36	51 (Top vie	ew)
GIC					2			Pi* Signal P ANA0 ADC1A	n C GPIO ISNS Analo	GPIO Maxi	Retirne Inve	rt Doubl	DelayDe	layInve	t User N	lodifie					
GPIO								ANA0 ADC1 A		n/a n/a n/a n/a	n/a n/a	n/a n/a	n/a n/a n/a n/a								
HASH	2	1	.					ADUT ADUT A	ono Anaio	nva n/a	nva nva	n/a	iva iva	n/a							
HDP					V																
HPDMA1		5		1	1	12	12														
HPDMA2		5	1		2	1															
HPDMA3		2	1	1	12	1	1										Q	53	Θ	0.5	<0
																	\sim		\sim		

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 ROM	(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.

Figure 61. U-Boot selection						
Boot time:		Runtime conte	exts:			
A35 ROM	A35S (TF-A BL2)	A35S (OP-TEE)	A35NS (U-Boot)	A35NS (Linux)	M33S (TF-M)	M33NS (Cube)
		1		V		

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.



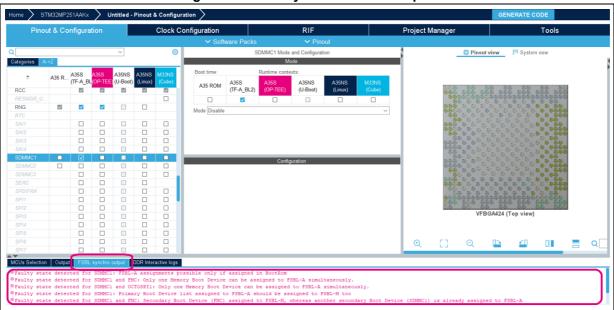


Figure 62. FSBL synchronization output

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 different panels at will: hovering the mouse over a panel border displays a two-ended arrow: right-click and pull in a direction to either 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 63*).



Figure 63. 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 64*). 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.



Figure 64. Contextual help detailed information							
Pinout & Conf	iguration	Clock Configuration					
		Additional Softwares					
Options Q 🗸 🗸 🗸		IRTIM Mode and Configuration					
Categories A->Z		Mode					
\$							
IRTIM:							
InfraRed Interface							
Status: OF Not available:							
	117 must be configured in (one of the availables output modes					
OF Summary:							
OF IRTIM (InfraRed Interface) o	OF IRTIM (InfraRed Interface) offers important advantages as a form of wireless communication. RC Nowadays, almost all audio and video equipment can be controlled using an infrared remote control.						
		s, which are processed to retrieve/decode					
	SF the information they contain.						
SF Related documentation:							
SF - <u>Datasheet</u>	SF - <u>Datasheet</u>						
TIM1							
TIM2							
-							

Figure 64. 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 (Half-Duplex) Multiprocessor Communication IDA LIN
Gray italic text Example: <i>LWIP</i>	Peripheral is not available because some constraints are not solved. See tooltip.	LWIP MBEDTL Lightheight TCP/IP stack IVIC Status GUADS Active only if: ETH IP configured / FREERTOS is enabled when MBEDTLS is enabled RCC RNG RTC
✓ ⊗ 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 ✓ 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 FS Phil Disable Activate_S Disable Oisable Oisable Oisable Oisable Oisable Disable Oisable Disable Oisable Disable Oisable Disable Oisable Disable Disable Disable Disable Disable Disable Disable Disable Disable
Example:	The peripheral is configured (one mode is set) and at least one of its other modes is unavailable.	Mode Enternal Phy Datable
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. Component list,	mode 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 <u>Pinout view</u> 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 65* for an example.

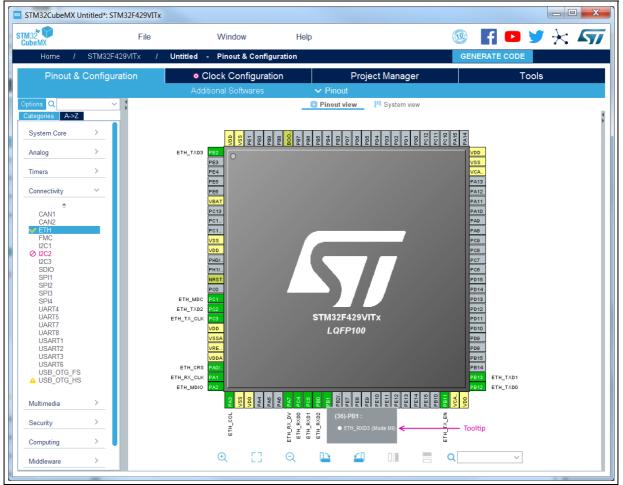


Figure 65. Pinout view

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 66*).
- 2. Click to select the new function to assign to the pin.

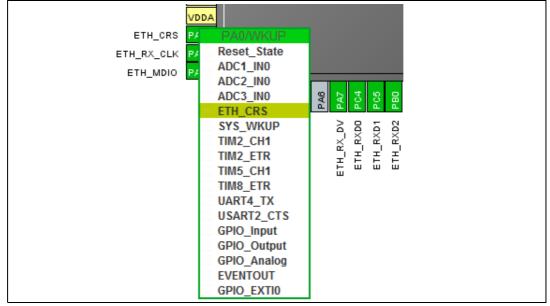


Figure 66. Modifying pin assignments from the Pinout view

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 67, two blocks of pins are available on an STM32F107xx MCU 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}.



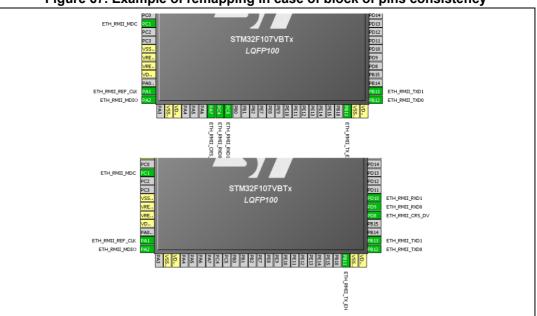


Figure 67. Example of remapping in case of block of pins consistency

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 68) lists all configured pins.

	Pin Name	Signal Name	User Label
	PUT	ETH_MDC	
	PC2	ETH_TXD2	
	PC3	ETH_TX_CLK	
	PA0/WKUP	ETH_CRS	
	PA1	ETH_RX_CLK	
	PA2	ETH_MDIO	
	PA3	ETH_COL	
ţ.	PA4	DAC_OUT1	DAC1
ţ.	PA5	DAC_OUT2	DAC2
	PA7	ETH_RX_DV	
	PC4	FTH RXD0	
		Apply	OK Cancel

Figure	68	Pins/Signals	Ontions	window
Iguie	00.	r ilis/Sigilais	options	WIIIGOW

- 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.



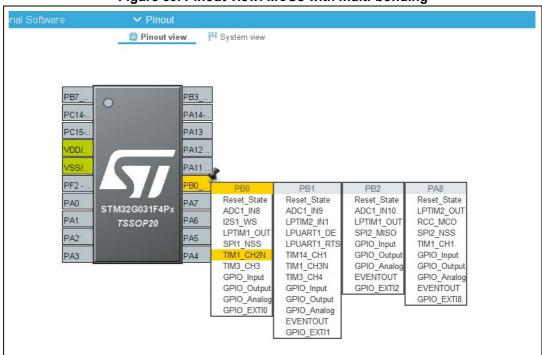
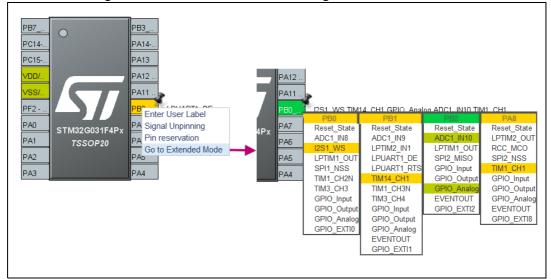
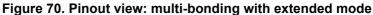


Figure 69. Pinout view: MCUs with multi-bonding

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.







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 71. 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 72*).

Figure 72. Configuration window tabs (GPIO, DMA, and NVIC settings for STM32F4 series)										
FIGURE 77 CONDOURATION WINDOW TADS (GPTO, DIMA, AND NVIG SETTINGS FOR STIM57E4 SERIES)	Ciaura	70 04	soficiuration	window tobo			and NIVIC	actions	for CTM22E	A a a mia a \
	Flaure	12. 60	onnouration	window labs	(GPIU.	DIVIA.		sellings		4 series)

			I2C1 Mode an	d Configuratio	on		
			Mo	ode			
12C 2C							\sim
			Config	uration			
Reset Co	onfiguration						
				o:			
🥑 Paramet	ter Settings	User Constant	nts 🥑 NVIC	Settings	GPIO Settings	🕑 DMA S	ettings
Search Sigr	nals						
Search (Crt		7				Show only N	Aodified Pins
Pin Name 🌻 988	Signal on Pir I2C1_SCL	n GPIO Pin State n/a	GPIO mode Alternate Fun		/ Maximum out	User Label	Modified
,B0	I2C1_SCL	n/a	Alternate Fun		High 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 73*). 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.





Home / STM32F423VHHx / Untitled Pinout & Configuration GENERATE CODE Pinout & Configuration Clock Configuration Project Manager Tools Additional Softwares Pinout Options Q I2C1 Mode and Configuration Categories A->Z Mode PNU Configuration Configuration I2C1 Configuration Configuration I2C2 I2C1 Configuration I2C2 I2C1 Configuration I2C2 I2C1 Configuration I2S1 I2S2 Isseet Configuration I2S3 ISS Isseet Configuration I2S4 Search (CrlHF) Isseet Configure the below parameters : I2S4 Search (CrlHF) Isseet Configure Searces UBJ/PEG Master Features Isseet Searces	
Pinout & Configuration Clock Configuration Project Manager Tools Additional Softwares	57
Additional Softwares Pinout Options I2C1 Mode and Configuration Categories A>2 FSWL Mode GPIO I2C2 I2C3 Configuration I2S1 Parameter Settings I2S4 Configure the below parameters : I2S4 Q Search (Crift+F) IVDG Master Features	
Options Q I2C1 Mode and Configuration Categories A->Z r Switc GPIO GPIO Configuration I2C2 Configuration I2C3 Reset Configuration I2S1 Parameter Settings I2S3 Configuration I2S4 Configure the below parameters : I2S4 Q Search (CrtH+F) IVDG V Master Features	
Categories A>Z Mode Configuration IZC IZC IZC1 Configuration IZC2 IZC3 IZC3 Reset Configuration IZC3 IZC3 IZC3 IZC3 IZC4 Configuration IZC3 IZC4 Configuration IZC3 IZC5 IZC4 IZC3 IZC5 IZC5 IZC4 IZC5 IZC6 IZC5 IZC6 IZC6 IZS4 IZS5 IZC6 IVDG ILIEUPEG V Master Features	
Consigning V/2 r Swit GPIO GPIO Configuration I2C3 Reset Configuration I2S3 Parameter Settings I2S3 Configuration I2S4 Configuration I2S5 ISS IWDG IMDG IUBJPEG Vaster Features	1
Point Configuration C2C1 Configuration I2C2 Reset Configuration I2S1 ● Parameter Settings I2S2 Configuration I2S3 Configuration I2S4 Configuration I2S5 Q Search (CrtH+F) IUBUPEG ✓ Master Features	
GPIO I2C1 Configuration I2C2 I2C3 Reset Configuration I2S1 Image: Configuration Image: Configuration I2S2 Image: Configuration Image: Configuration I2S2 Image: Configuration Image: Configuration I2S2 Image: Configuration Image: Configuration I2S3 Image: Configuration Image: Configuration I2S4 Image: Configuration Image: Configuration I2S5 Image: Configuration Image: Configuration I2S4 Image: Configuration Image: Configuration I2S5 Image: Configuration Image: Configuration I2S4 Image: Configuration Image: Configuration I2S4 Image: Configuration Image: Configuration I2S5 Image: Configuration Image: Configuration I2BUPEG Image: Master Features Image: Configuration	
I2C2 I2C3 I2C3 Reset Configuration I2S1 Parameter/Settings User Constants NVIC Settings GPIO Settings DMA Settings I2S1 Parameter/Settings User Constants NVIC Settings GPIO Settings DMA Settings I2S1 Parameter/Settings User Constants NVIC Settings GPIO Settings DMA Settings I2S4 I2S5 General-(crit+F) General-(crit+F) General-(crit+F) Master Features 	
I2C3 Reset Configuration I2S1 Image: Configuration I2S2 Image: Configuration I2S2 Image: Configuration I2S3 Image: Configuration I2S4 Image: Configuration I2S4 Image: Configuration I2S5 Image: Configuration IVDG Image: Configuration IUBJPEG Image: Master Features	
I2S2 ● Parameter Settings ● User Constants ● NVIC Settings ● GPIO Settings ● DMA Settings I2S3 Configure the below parameters: I2S4 I2S5 IVDG Search (CrtI+F) ILIBUPEG ✓ Master Features	
I2S4 I2S4 I2S5 Q Search (Crt(+F)) WDG VMDG UBJPEG ✓ Master Features	
I2S5 Q Search (CrtH+F) ③ IWDG UBJPEG ✓ Master Features	
INDG Value V	0
LPTIM1 I2C Speed Mode Standard Mode	
MBEDTLS NVIC I2C Clock Speed (Hz) 100000	
PDM2PCM Slave Features	
QUADSPI Clock No Stretch Mode Disabled RCC Primary Address Length caloring Z bit	
RVCC Primary Address Length selection 7-bit RNG Dual Address Acknowledged Disabled	
RTC Dual Address Acknowledged Disacted SAI1 Primary slave address 0	
▲ SDIO General Call address detection Disabled	
SPI1 SPI2	
SPI3	
SPI4 SPI5 Dual Address Acknowledged	
✓ SYS DualAddressNode	
TIM1 Diagnostic: TIM2 When nymany address is 7 hits length we can have a dual address	
TIM4 TIM5	
ims L	

Figure 73. 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 V 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 (Citl+F)	Search
Reset Configuration	Resets the component back to its default configuration (initial settings from STM32CubeMX).

Tabla O	Darinharal	and Middlowar	e configuration	window	suffana an	d toolting
I able o.	renunerar	anu muulewar	e comuuration	willdow L	Julions ar	ια ισοπισε

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 74*).

Figure 74. Formula when input parameter is set in No Check mode

Ľ	TDC Mode and Configuration Mode	
Display Type RGB888 (24 bits)	WOUE	\sim
	Configuration	
Reset Configuration		
✓ Parameter Settings	Constants 📀 NVIC Settings 📀 GPIO Settings	
Configure the below parameters :		
Q 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	



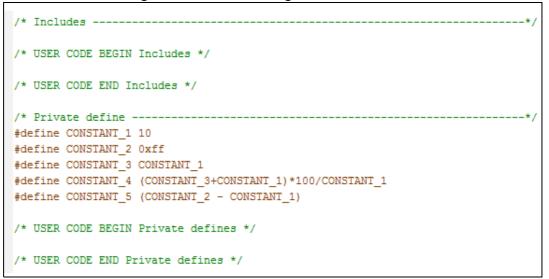
4.5.11 User Constants configuration window

An **User Constants** tab is available to define user constants (see *Figure 75*). Constants are automatically generated in the STM32CubeMX user project within the main.h file (see *Figure 76*). Once defined, they can be used to configure peripheral and middleware parameters (see *Figure 77*).

	i igule 75. 036		
	SWPMI1 Mode	and Configuration	4
	N	lode	
Mode Full-Duplex (normal mode)			\sim
	Carl	guration	
	Conii	guration	
Reset Configuration			
😔 Parameter Settings 🛛 😔 User Cons	tants 🛛 🥺 NVIC Settings	🥝 GPIO Settings 🛛 📀 DMA Settings	
Search Constants			
Search (CrtI+F)		add	remove
Constant Nar	ie	Constant Value	
CONSTANT_1		10	
CONSTANT_2		0xff	
CONSTANT_3		CONSTANT_1	
CONSTANT_4 CONSTANT_5		(CONSTANT_3 + CONSTANT_1)*100/CONSTANT_1 (CONSTANT 2 - CONSTANT 1)	
CONSTRAT_5		(CONSTAINT_2-CONSTAINT_T)	

Figure	75.	User	Cons	tants	tab

Figure 76. Extract of the generated main.h file



	Con	figuration	
Reset Configuration			
⊘ Parameter Settings	NVIC Settings	OPIO Settings	OMA 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		1455 kBits/s	
Transmission Buffering Mode		No Software buffer	
Reception Buffering Mode		No Software buffer	

Figure 77. 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 78*).

A constant consists of:

- A name that must comply with the following rules:
 - It must be unique.
 - It shall not be a C/C++ keyword.
 - It shall not contain a space.
 - It shall not start with digits.
- A value

The constant value can be (see Figure 75 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 its value can still be changed: double- click the row that specifies the user constant to be modified. 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. However 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.

User Constants	X
constant Name constant Value	
OK	
	Cancel

Figure 78. Specifying user constant value and name

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 79*).

Figure 79. Deleting an user constant is not allowed when it is already used for another constant definition



• 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 80*).

Figure 80. Confirmation request to delete a constant for parameter configuration

🔤 De	elete user constant warning:
	The selected user constant is used in the configuration of some ips! Are you sure you want to delete it ?
	<u>Yes</u> <u>N</u> o

UM1718 Rev 46



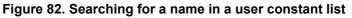
Clicking Yes leads to an invalid peripheral configuration (see Figure 81).

Figure 81. Consequence when deleting a user constant for peripheral configuration

😣 Parameter Settings 🛛 🥝 User Consta	ints 🛛 🥺 NVIC Settings	🥺 GPIO Settings	🥝 DMA Settings
Configure the below parameters :			
Q Search (CrtI+F)	© ©		
✓ ⊗ 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	

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 82* and *Figure 83*).



NVIC Settings OPIO Settings	OMA Settings
Parameter Settings	Subser Constants
Search Constants CONSTANT_1	add remove
Constant Name	Constant Value
CONSTANT_1	10
CONSTANT_3	CONSTANT_1 + CONSTANT_2

Figure 83. Searching for a value in a user constant list

OPIC Settings	OMA Settings	User Constants
Search Constants		add remove
Constant Name CONSTANT_1	10	Constant Value



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 84*). 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								
				Configuration	1			
Group	By Periphera	als						
	· ·							
🕑 GPIO	🕗 ADC1	O ADC2	📀 I2C1	2 12C2	🥺 NVIC			
Search S	lignals							
Search (CrtI+F)					🗖 S	how only Mod	dified Pins
Pin Na 🕈		GPIO outp		GPIO Pull		. Fast Mode	User Label	Modified
A9	n/a	n/a		No pull up		n/a		<u> </u>
B15	n/a	Low		No pull up		n/a		
C8	n/a	n/a	Input mode	No pull up	n/a	n/a		

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 85*).

By Periphe	rals						
ADC	⊘ I2C						
Search Signals Search (CrtI+F) Show only Modified Pins							
Signal 🔶	GPIO outp	GPIO mode	GPIO Pull	Maximum	Fast Mode	User Label	Modified
n/a	Low	Output Pu	No pull-u	Low	n/a		
n/a	n/a	Analog m	No pull-u	n/a	n/a		
n/a	n/a	Input mode	No pull-u	n/a	n/a		
	gnals rt/+F) Signal 🔷 n/a n/a	gnals rt/+ <i>F)</i> Signal GPIO outp n/a Low n/a n/a	gnals rt/+F) Signal GPIO outpGPIO mode n/a Low Output Pu n/a n/a Analog m	gnals rt/+F) Signal CPIO outpGPIO mode GPIO Pull n/a Low Output Pu No pull-u n/a Analog m No pull-u	gnals rt/+F) Signal CPIO outpGPIO mode GPIO Pull Maximum n/a Low Output Pu No pull-u Low n/a n/a Analog m No pull-u n/a	gnals gnals Image: Sho Signal GPIO outp GPIO outp GPIO mode GPIO Pull Maximum Fast Mode n/a Low Output Pu No pull-u Low Output Pu No pull-u In/a	gnals gnals Show only Modi Signal * GPIO outp GPIO mode GPIO Pull Maximum Fast Mode User Label n/a Low Output Pu No pull-u Low n/a n/a n/a Analog m No pull-u n/a n/a

As shown in *Figure 86*, **r**ow multi-selection can be performed to change a set of pins to a given configuration at the same time.

Configuration									
Group By Peripherals									
😔 GPIO	😔 ADC	⊘ I2C							
		_							
Search S	Search Signals								
Search (Crtl+F)					🗖 Sho	w only Mod	ified Pins	
Pin Name	Signal 🗘	GPIO outp	GPIO mode	GPIO Pull	Maximum	. Fast Mode	User Label	Modified	
PB6	I2C1 SCL		Alternate			Disable			
PB9	I2C1_SDA	n/a	Alternate		Low	Disable			
PF1	I2C2_SCL	n/a	Alternate	Pull-up	Low	n/a		Image: A start and a start	
PF0	I2C2_SDA	n/a	Alternate	Pull-up	Low	n/a		Image: A start and a start	
PA8	I2C3_SCL	n/a	Alternate	Pull-up	Low	n/a		Image: A start and a start	
PC9	I2C3_SDA	n/a	Alternate	No pull-u	Low	n/a			
PB9#PF1	#PF0#PA8	Configuratio	n :						
0.0010									
GPIO mo	GPIO mode V						~		
GPIO Pull-up/Pull-down Pull-up ~						\sim			
Maximum output speed								\sim	
User Lab	User Label								

Figure 86. 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 87*).

Configuration							
📀 DMA1, DMA2 🛛 📀 MemToMe	m						
DMA Request	Stream	Direction		Priority			
Select DMA_GENERATOR7 I2C1_RX I2C1_TX I2C2_RX I2C2_TX I2C3_RX I2C3_TX MEMTOMEM Use Fifo Threshold	~	Increment Address Data Width [Burst Size [Peripheral	Add Delete Memory			
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 88*). 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 87. Adding a new DMA request

오 DMA1, DMA2 🛛 📀 Me		Configuration		
DMA Request	Stream	Direction		Priority
2C1_TX 2C1_RX	DMA1 Stream 0 DMA1 Stream 1	Memory To Peripheral Peripheral To Memory	Low Low	
				Add Delete
DMA Request Settings			Peripheral	Memory
Mode Normal	\sim	Increment Address		V
Use Fifo 🔲 Threst	nold 🗸 🗸	Data Width E	Byte 🗸	Byte 🗸
		Burst Size	~	~
DMA Request Generator	Settings			
Request generation Signa	al			\sim
Signal polarity				\sim

Figure 88. 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.



⊘ DMA1, DMA2	MemToMem	Configuration		
DMA Request	Stream	Direction		Priority
IEMTOMEM	DMA1 Stream 2	Memory To Memory	Low	
				Add Delete
DMA Request Settings			Src Memory	Dst Memory
Mode Normal	~	Increment Address	√	
Use Fifo 🔽 Thre	shold Full	 Data Width 	Byte \vee	Byte ~
		Burst Size	Single ~	Single ~
DMA Request Generato	or Settings			
Request generation Sig	Inal			\sim
Signal polarity				\sim
Request number				

Figure 89. 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 90*).

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 90*) 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.



		File	Window	Н	elp	19	f	D y	$\times 47$
	F401CBUx	➢ Untitled - Pi	nout & Configuration	\rangle		GEN	IERATE	CODE	
Pinout & Co	onfiguration	• (Clock Configuration		Project Manager			Tool	s
		✓ Softwa	re Packs	✓ Pin					
Q ~	0			NVIC	Mode and Configuration				
Categories A->Z									
		NVIC _ ⊘ Code g	eneration		Configuration				
System Core	~				_		_		
÷	Prio	rity Group 4 bits fo	or pre-emption priority 0 bits	for su… ∨	Sort by Premption Priority	and Sub Pr	iority 🗌	Sort by interru	ipts names
DMA GPIO	Sea	rch Search	(CrtI+F)	0 0	Show available interrupts ~		~	Force DMA cl	nannels Interrupts
IWDG			. /						
NVIC			NVIC Interru	pt Table		Enable	ed Pre	emption Priority	 Sub Priority
RCC	Non	maskable interrupt				\checkmark	0		0
✓ SYS	Hard	fault interrupt				\checkmark	0		0
WWDG	Mem	ory management fa	ult			\checkmark	0		0
	Pre-f	etch fault, memory	access fault			\checkmark	0		0
Annin	> Unde	fined instruction or	illegal state			\checkmark	0		0
Analog	Syst	em service call via S	SWI instruction			\checkmark	0		0
Timers		ig monitor				\checkmark	0		0
Timers	Penc	lable request for sys				\checkmark	0		0
Connectivity		base: System tick				\checkmark	15		0
Connectivity	PVD	interrupt through E	XTI line 16				0		0
Multimedia	2	n global interrupt					0		0
		global interrupt					0		0
Computing		1 stream0 global in				\checkmark	0		0
1.0		1 stream6 global in	terrupt			\checkmark	0		0
Middleware	· ·	event interrupt					0		0
		error interrupt					0		0
¢ FATES		global interrupt					0		0
FREERTOS		2 stream0 global in				\checkmark	0		0
LIBJPEG		2 stream3 global in				\checkmark	0		0
MBEDTLS		On The Go FS glob	pal interrupt				0		0
PDM2PCM	FPU	global interrupt					0		0
USB_DEVICE USB_HOST					led Preemption Priority	✓ Sub Pi		~	



When FreeRTOS is enabled, an additional column is shown (see Figure 91).

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.



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 91*).

	Co	nfiguration		
⊘ NVIC ⊘ Code generation				
Priority Group 4 bits for pre-emption	n priori	✓ Sort by	Premption	Priority and Sub Priority
Search (Crtl+F)	0	Show of the second s	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	\sim	0	0	
Undefined instruction or illegal state	\sim	0	0	
System service call via SWI instruction	\sim	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	\checkmark
Flash global interrupt		5	0	✓
RCC global interrupt		5	0	\checkmark
I2C1 event interrupt		5	0	\checkmark
I2C1 error interrupt		5	0	\checkmark
SPI1 global interrupt		5	0	\checkmark
USB On The Go FS global interrupt		5	0	\checkmark
FPU global interrupt		5	0	✓
Enabled Preemption Priority	~	Sub Priority	🗸 🔲 Use	s FreeRTOS functions

Figure 91. NVIC configuration tab - FreeRTOS enabled

Peripheral dedicated interrupts can also be accessed through the NVIC window in the configuration window (see *Figure 92*).

Figure 92. I2C NVIC configuration window

Configuration Reset Configuration									
⊘ NVIC Settings	¥	🕑 DMA	Settings						
😔 Paran	neter Settings			😔 User Const	ants				
NVIC Interrupt	Table i	Enabled	Preem	ption Priority	Sub Priority				
I2C1 event interrupt			5		0				
I2C1 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 93*). Note that system interrupts are not eligible for init sequence reordering as the software solution does not control it.

	Configuration	n	
🛛 📀 NVIC 🛛 😔 Code generation			
Enabled interrupt table	🗸 Select for init sec	uence 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			\checkmark
Pendable request for system service			✓
Time base: System tick timer		-	✓
RCC global interrupt			✓
ADC1, ADC2 and ADC3 global inter	\checkmark		<u>✓</u>
CAN1 TX interrupts	✓		✓
I2C1 event interrupt	✓		✓
Interrupt unmasking ordering table (interrupt init code is	moved after all	the peripheral init code)
interrupt contracting classic (interrupt interested in	mored alter al	
Rank	Inte	errupt name	
	ind ADC3 global inte	rrupts	
2 CAN1 TX inter			
3 I2C1 event inte			
4 RCC global in	errupt		
	≣l I	L)]
		Move up	

Figure 93. NVIC Code generation – All interrupts enabled



```
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 94*).

As an example, the configuration shown in *Figure 94* 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 94* removes the code mentioned earlier from the stm32xxx_it.c file.



Figu	ie 34. NVIC C	ode generation – IRQ Handle	er generation
		Configuration	
📀 NVIC 🛛 😔 C	ode generation	Coniguration	
Enabled inter		Select for init sequence ordering	Generate IRQ handler
Non maskable interr	rupt		
Hard fault interrupt			<u> </u>
Memory manageme			<u> </u>
Pre-fetch fault, mem	*		✓
Undefined instructio	-		<u>✓</u>
System service call \	/ia SWI instructi		✓
Debug monitor			✓
Pendable request fo			✓
Time base: System f	tick timer		✓
RCC global interrupt	t		
CAN1 TX interrupts			
A		-	
Interrupt unmasking	ordering table (interrupt init code is moved after all	the peripheral init code)
Rank		Interrupt pages	
rtdllk 4	ADC1 ADC2 -	Interrupt name	
0		nd ADC3 global interrupts	
2	CAN1 TX inter		
3	I2C1 event inte		
4	RCC global in	terrupt	

Figure 94. 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 95*).
- 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.



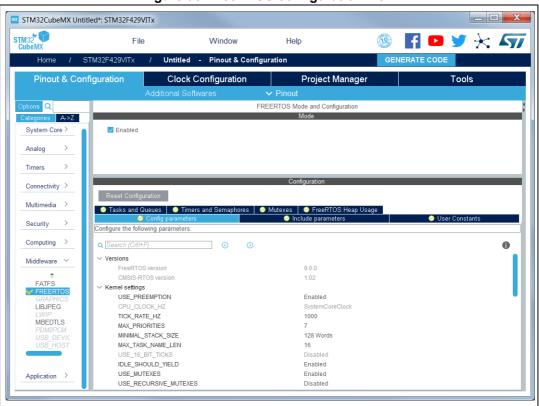


Figure 95. FreeRTOS configuration view

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 96*).

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.



	 ✓ Tasks and Queues ✓ Timers and Semaphores ✓ Mutexes ✓ FreeRTOS Heap Usage ✓ Config parameters ✓ User Constants ✓ Tasks 						nstants	
Task Name	Priority	Stack Size (Wor	Entry Function	Code Gener	rati Parameter	Allocation	Buffer Nam	e Control Block N
defaultTask	osPriorityNormal	128	StartDefaultTask	Default	NULL	Dynamic	NULL	NULL
	osPriorityHigh	128	StartTask_A	Default	NULL	Dynamic	NULL	NULL
myTask_B	osPriorityLow	256	StartTask_B	Default	NULL	Dynamic	NULL	NULL
Queues								
Queue Nan		Queue Size	Item Size		Allocation	Buffer N		ontrol Block Name
nyQueue_1	16		uint16_t		amic	NULL	NUL	-
nyQueue_2	32		uint16_t	Dyn	amic	NULL	NUL	L

Figure 96. 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 97*). 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* 97). 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.



Tasks and Queues O Timers and Semaphores O Mutexes O FreeRTOS Heap Usage O Config parameters O User Con- Casks					stants			
Task Name	Priority	Stack Size (Wor		Code Generati		Allocation	Buffer Name	
defaultTask Fask A	osPriorityNormal osPriorityHigh	128	StartDefaultTask StartTask A	Default	NULL	Dynamic Dynamic	NULL	NULL
myTask B	osPriorityLow	256	StartTask_B	Default	NULL	Dynamic	NULL	NULL
Queues								
Queue Na		Queue Size	Item Size	-	Allocation	Buffer Na		ontrol Block Name
myQueue_1	16		uint16_t	Dynam		NULL	NULL	
nyQueue_2	32		uint16_t	Dynam	IC	NULL	NULL	

Figure 97. FreeRTOS: creating a new task

The following code snippet shows the generated code corresponding to *Figure 96*.

```
/* 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 98*).



			Configuration			
Reset Config	uration					
🕑 Tasks and G		and Semaphores 🛛 🤡		RTOS Heap Usage		<u> </u>
	Config parameters		😔 Include para	meters	🕑 Use	er Constants
Timers —		_				
Timer Name		Туре	Code Generation (Allocation	Control Block Name
myTimer01	Callback01	osTimerPeriodic	Default	NULL	Dynamic	NULL
nyTimer02	Callback02	osTimerOnce	Default	NULL	Dynamic	NULL
						Add Delete
	res Semaphore Name	Dynamic	Allocation	NUL	Control Bl	
		Dynamic	Allocation	NUL		
		Dynamic	Allocation	NUL		
nyBinarySem01	Semaphore Name	Dynamic	Allocation	NUL		ock Name
nyBinarySem01 Counting Semapl Semap	Semaphore Name hores	Count	Allocation	Allocation	L	ock Name
nyBinarySem01	Semaphore Name hores		Allocation	Allocation	L	ock Name Add Delete

Figure 98. 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 98.

```
/* 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 99*).

Figure 99. FreeRTOS heap usage

	Con	figuration			
Reset Configuration					
	Timers and Semaphores	🥝 Mutexes 🛛 🔇	FreeRTOS Heap Usage		
😔 Config parameti	ers		😔 Include pa	arameters	
				A Z	1 🗎 🚺
Summary					
IEAP STILL AVAILABLE		0 Bytes			
OTAL 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)		624 Bytes 1136 Bytes			
defaultTask		624 Bytes			
Task A		15472 Bytes			
myTask_B		15472 Bytes			
otal amount of the heap used by known objects (u	ser objects, internal freertos objec	ts)			
VARNING					
urrent computed value is greater than the configT					
o avoid runtime issues, you should increase confi	g_TOTAL_HEAP_SIZE, remove/ac	ljust some defined	objects (tasks, queues, timer	s, mutexes, semap	hores) or change
ne Memory Management scheme to heap_3					
Nore about FreeRTOS Heap:					
reeRTOS uses a region of memory called Heap (i	ato the RAM) to allocate memory fo	artacke guouoe ti	merc comphores mutaves	and when dynamic	ally creating
reek roo uses a region of memory called Heap (i	to the reaction to anocate memory to	or tasks, queues, th	mers, semapriores, mutexes	anu when dynamic	any 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: SysTick, TIM1, TIM2,... (see *Figure 100*).

Pinout & Configuration	Clock Configuration	
	Additional Softwares	~
Options Q ~	SYS Mode and Configuration	
Categories A->Z	Mode	
÷	Debug Disable ~	1
RTC SAI1	System Wake-Up	
SDIO	Timebase Source SysTick 🗸 🗸	7
SPI1	SysTick	
SPI3	TIM1 TIM2	
SPI4	TIM3	
✓ SYS TIM1	TIM4	1
TIM2	TIM5	
TIM3	TIM6 TIM7	
TIM4	111117	

Figure 100. 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 101*).



Figure 101. 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 102*, the SysTick priority is set to 0 (High) when using the SysTick without FreeRTOS.

ODE Code generation ODE Code generation			
Priority Group $\begin{tabular}{cccc} 4 & \end{tabular}$ 4 bits for pre-emption priority 0 bits for $\label{eq:constraint}$	Sort	by Premption Priority	and Sub Priority
Search (CrtI+F) ③	Shov	w only enabled interrup	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
Time base: System tick timer	\checkmark	0	0
PVD interrupt through EXTI line 16		0	0
Flash global interrupt		0	0
RCC global interrupt		0	0
SPI2 global interrupt		0	0
TIM6 global interrupt, DAC1 and DAC2 underrun error interrupts		0	0
FPU global interrupt		0	0

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 103*, the SysTick priority is set to 15 (Low) when using the SysTick with FreeRTOS.

Figure 103. N	IVIC settings wi	en using FreeRTOS	and SysTick as	HAL timebase
---------------	------------------	-------------------	----------------	--------------

🛛 🛛 NVIC 🔤	Code generation			
Priority Group	4 bits for pre-emption priority 0 bits for $ \smallsetminus $	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 in	nterrupt	\checkmark	0	0
Hard fault interru	pt	\checkmark	0	0
Memory manage	ement fault	\checkmark	0	0
Pre-fetch fault, m	emory access fault	\checkmark	0	0
Undefined instru	ction or illegal state	\checkmark	0	0
System service of	all via SWI instruction	\checkmark	0	0
Debug monitor		\checkmark	0	0
Pendable reque	st for system service	\checkmark	0	0
Time base: Systematics	em tick timer	\checkmark	15 🗸	0
PVD interrupt thr	ough EXTI line 16		0	0
Flash global inte	rrupt		0	0
RCC global inter	rupt		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 $\ \sim$	Sort by Premption Priority and Sub Priority							
Search (CrtI+F) ③	🗖 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 104. 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 STM32MPUs

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

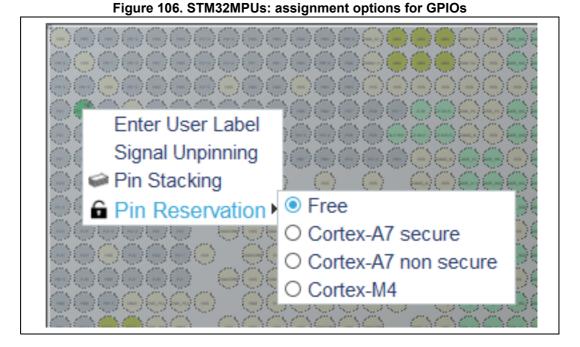
- assign components to one or several run time contexts
- configure peripherals as boot devices
- select the peripherals to be managed by boot loaders
- assign GPIOs to one runtime (see *Figure 106*).

These possibilities are offered in two different panels (see *Figure 105*):

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

STM32CubeMX Untitled*: STM3	2MP151CADx				
	File	Winde	ow Help	🛞 f	🖸 🄰 🔆 🏹
Home > STM32MP151CADx	> Untitled ·	Pinout & Con	figuration >	GENERAT	ECODE
Pinout & Configuration	Cloc	k Configurat	ion Projec	ct Manager	Tools
			✓ Pinout		
Options Q		~]	TIM12 Mode and Configu	uration 🛔
Show contexts				Mode	
Boot ROM A7BL		7 <u>NS M4</u>		untime contexts: loader A7S A7	7NS Cortex-M4
TIM5 TIM6			Slave Mode Disable		× · · · · · · · · · · · · · · · · · · ·
TIM7			Trigger Source Disabl	le	\sim
TIM8			🗌 Internal Clock		
= TIM12			Channel1 Disable		~
TIM13			Channel2 Disable		~
TIM14	I		Combined Channels	Disable	~
TIM15				Configuration	

Figure 105. STM32MPUs boot devices and runtime contexts



UM1718 Rev 46



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 107*).

STM32CubeMX Untitled*: STM32	MP151CADx						x
STM32 CubeMX	File	Window	Help	(19)	F 🕒 🔰	• 🗙 🗸	7
Home > STM32MP151CADx	Untitled	- Pinout & Config	uration >	GEN	ERATE CODE		
Pinout & Configuration	Cloc	ck Configuratio	n Proje	ect Manager		Tools	
			✓ Pinout				
Options Q		~		FMC Mode and C	Configuration		1
Categories A->Z				Mode			
DTS Boot ROM A7BL	A7S A	A7NS M4		Runtime contexts: ot loader A7S	A7NS C	ortex-M4	
ETH1	\checkmark		V A NAND Flash 1				L
🔺 FMC 🛛 🗹			Chip Select NCE	2		\sim	11
FREERTOS			Data/Address 8	bits		\sim	
GIC	\checkmark		VWAIT Read	ly/Busy			
GPIO 🔽 🗌							
HASH1							

Figure 107. 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.



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**

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 & Configur	ation	Clock Cor	Pinout & Configu	ration	Cloc
RIF Configuration	Default Configurat		RIF Configuration	Default Configurat	

Figure 108. 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 109*): the user cannot read, write, or use it in any form.



M32 CubeMX	File	Window	Help		隊ᠿᅊᅷᡘᢋ
Home STM32MP251/ Pinout & Configurat		nfiguration	RIF	Project Manager	Tools
RIF Configuration	Default Configuration				
Peripherals (RISUP)					
Domains (RIMU)					
	AF)				

RIF global configurations for STM32N6

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



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 & Configuration		Clock Configuration		RIF			
	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	-	✓			
ernal memories (RISAB)	HDP	57	1				
	12C1	41	1				
	12C2	41	-				
	12C2	42	-				
			-				
	12C4	44	-				
RIF-Aware IPs	12C5	45	-				
	12C6	46	-				
	12C7	47	-				
	12C8	48	-				
	I3C1	114	-				

Figure 110. 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 111 shows on left hand side the IP allocation per software context, and, on the right-hand side the equivalent in the RISUP configuration panel.

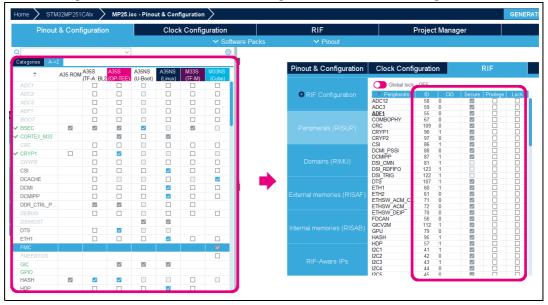


Figure 111. Software context configuration vs. RISUP configuration



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.

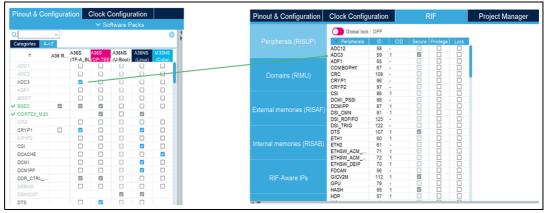


Figure 112. Example of IP assignment to one context and result in RISUP

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 113*.

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).

Pinout & Cor	Pinout & Configuration Clock Configuration			RIF Pinout & Configuration C				uratio	n	RIF			Project Manager			
			✓ So	ftware F	acks		🗸 Pinou	ut		Global lock	OFF					
۹	\sim					٢	1									
Categories A-	>Z						1		Peripherals (RISUP)	Peripherals ADC12	ID 58	CID	Secure	Privilege	Lock	
outegones	-	1050	A35S	4.05110	A35NS	1100110				ADC12 ADC3	58	- 182				
÷	A35 R	A35S	A355 (OP-TEE	A35NS	A35NS (Linux)	M33NS (Cube)				ADE1	55	-		П	п	
ADC1		(IF-A_BL		(U-BOOI)						COMBOPHY	67					
									Domains (RIMU)	CRC	109	-				
ADC2										CRYP1	96	-				
ADC3					~					CRYP2	97	•		Ц	<u> </u>	
ADF1										CSI DCMI PSSI	86 88	1		H	H	
BOOT										DCMI_PSSI	87	1		H	H	
V BSEC	V								External memories (RISAF)	DSI_CMN	81	1			Ö	
										DSI_RDFIFO	123					
✓ CORTEX_M33			1		1					DSI_TRIG	122	-				
CRC										DTS	107	1	~			
CSI					V				Internal memories (RISAB)	ETH1	60	1				
DCACHE									Internal memories (RISAB)	ETH2 ETHSW_ACM	61 71	-		H	H	
DCMI					Image: Contract of the second seco					ETHSW_ACM	72	1		H	H	
					_					ETHSW_DEIP	70	1				
DCMIPP					~					FDCAN	56					
DDR_CTRL		1	1						RIF-Aware IPs	GICV2M	112	1	1			
							Q	53		GPU	79	-				
DSIHOST		_	_		2		<u> </u>			HASH	95	1				
	_		_		1.0	_				HDP	57	1		<u> </u>	<u> </u>	

Figure 113. Example of IP assignment to two contexts and result in RISUP



Pinout & Configuration	Clock Configu	iration		F	RIF	
	Global lock	: OFF				
Parinharala (DISUD)	Peripherals	ID	CID	Secure	Privilege	Lock
Peripherals (RISUP)	ADC12	58	-			~
	ADC3	59	-		Image: A start and a start	
	ADF1	55	-		~	~
	COMBOPHY	67	-			~
	CRC	109	-		~	
	CRYP1	96	1			
Domains (RIMU)	CRYP2	97	-		~	~
	CSI	86	1			
	DCMI_PSSI	88				
	DCMIPP	87	1		~	
	DSI CMN	81	1			
	DSI RDFIFO	123				
External memories (RISAF)	DSI_TRIG	122				
	DTS	107	1	~		
	ETH1	60	1			
	ETH2	61	-			
	ETHSW_ACM	71	1			
	ETHSW_ACM	72	1			
Internal momeries (DISAR)	ETHSW DEIP	70	1			
Internal memories (RISAB)	FDCAN	56	-			
	GICV2M	112	1	~		
	GPU	79	-			
	HASH	95	1	~		
	HDP	57	1			
	12C1	41				
RIF-Aware IPs	12C2	42	-			
	12C3	43				
	12C4	44	-			
	1205	45				

Figure 114. 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 115*.

		inout ٤	k Config	juration					Clock Confi	guration				RIF	
2				\sim			٢				LTDC Mode	and Configuration			
Categories 🗛	->Z											Mode			
¢ LPTIM1	A35 ROM	A35S (TF-A B	A35S (OP-TEE)	A35NS (U-Boot)	A35NS (Linux)	M33S (TF-M)	M33NS (Cube)	Boot time: A35 ROM	A35S	Runtime co	A		A35NS	M33S	M33NS
									(TF-A BL	2) (OP-TE	E) (l	J-Boot)	(Linux)	(TF-M)	(Cube)
								Display Type D				LI .			
LPTIM4								Display Type D	ISable						
LPUART1															
											Cor	nfiguration			
								LTDC features							
NVIC_NS							V	LIDC features							
NVIC_S						1		Features	A35 ROM	A35S	A35S	A35NS	A35NS	M33S	M33NS
		~		~	~					(TF-A_BL2)	(OP-TEE)	(U-Boot)	(Linux)	(TF-M)	(Cube)
								LTDC_CMN LTDC_L0L1				✓ ✓	✓		
OCTOSPIM			V					LTDC_L0L1			 ✓ 				
OPENAMP								LTDC_ROT					V		
OTFDEC1			~												
		_	_	_	-	_									

Figure 115. Pseudo RIF-aware IP assignment

Peripheral instance protection for STM32N6

The RISUP panel for STM32N6 series (*Figure 116*) 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 116. Peripherals (RISUP) panel for the STM32N6 series

Pinout & Configuration	Clock Config	uration		IF	Proj	ect Manager	Tools
	Global lock : OFF						
Peripherals (RISUP)	Peripherals	ID	Secure	Privilege	Lock		
(NPU	106					
	OTG1_HS	56	1	~			
	OTG2_HS	57	1	~			
	PKA	77					
Domains (RIMU)	RNG	76					
	SAES	78					
	SAI1	4					
	SAI2	8					
Interrupts (IAC)	SDMMC1	53					
interrupts (IAO)	SDMMC2	54					
	SPDIFRX	61					
	SPI1/I2S1	0					
	SP12/12S2	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 117*), the column "Secure" is hidden, as all peripherals are working in secure mode (*Figure 118*).

Figure 117. 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			
xternarmemones (RISAF)	ICACHE	98			
	IWDG	68			
	JPEG	96			
	LPTIM 1	46			
	LPTIM2	47			
	LPTIM3	48			
RIF-Aware IPs	LPTIM4	49			
	LPTIM6	50			
	LPUART1	25			
	LTDC_CMN	102			

Figure 118. 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 & 0		1	Clock	Configura	lion		RIF			Proj	ect Manag	jer		To	ols	
					✓ Software	Packs	🗸 Pino	ut								
		~				0			FMC M	ode and Config	puration			Pinout v	iew III S	ystem vi
tegories A->Z							2			Mode				1		
•	A35 ROM	A35S (TF-A_BL2)	A35S (OP-TEE)	A35NS (U-Boot)	A35NS (Linux)	M33NS (Cube)	Boot time: A35 ROM	, A35S	Runtime A35S	contexts: A35	IS A	35NS	M33NS			
							A35 RUM	(TF-A_E	L2) (OP-TI	EE) (U-B	oot) (Linux)	(Cube)			
			0						22							
			0		0	-	Y NOR Flash	PSRAM/SRAM	ROM/LCD 1							
							Chip Sel	ect NE1					~			
LEX MC		- Ll	9				Memory	type Disable					~			
ic					2		Address						V			
PIO		had .					100000000000000000000000000000000000000									
IASH	23							gister Select	1580(8					0001000	.02 .02 .02 .0	00100
0P							Data Dis	able					196	1000		A Real
IPDMA1		2	53	53	52	5	Data/Adi	dress Disable				Q 14	lax: Disable	240.04	1 2 8 8 8 8 6 8 8 8 8 8 8 8 8 8 8 8 8 8 8	8.
IPDMA2		53	23	23	2	23	Clock Di	sable						1923		
IPDMA3		52	23	22	2	22										
										Configuration				1000	a a h a h a	1111
							Reset Confi	-						125.00		1000
							International Statements	Contraction of the second			-12				A424 (Top w	100
204							Parameter	Settings	Features C	GPIO Setting	25			VFBG	A424 (Top v	riew)
100							FMC features									
				EJ					A355	A35S	A35NS	A35NS	M33NS			
108							Features	A35 ROM	(TF-A_BL2)	(OP-TEE)	(U-Boot)	(Linux)	(Cube)			
							FMC_CFGR		8							
							FMC_NOR/		8	8						
							FMC_NOR/ FMC_NOR/		H			8	H			
							FMC_NOR/		ö							
							FMC_NAND									
														•	[] G	Ð
					0	D										2

Figure 119. FMC configuration

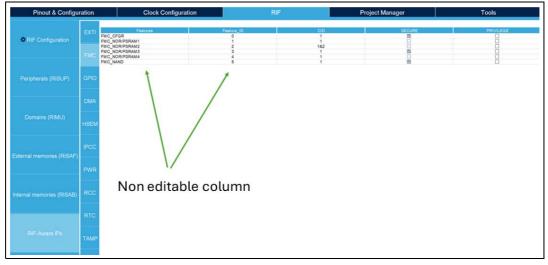
Configuring FMC in a RIF panel:

- 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 120. RIF FMC panel





Home 🔪	STM32MP	251CAlx) м	IP25.ioc	- Pinout	& Confi	guration	\rangle							
Pir	nout & Co	onfigura	ation			Clock	Configu	ration			RIF			F	Project Man
								✓ Software	Packs		Ƴ Pin	out			
Q				~			٢				RTC Mode an	d Configurati	on		
Categories	A->Z										М	ode			
÷	A35 R	A35S	A35S		A35NS		M33NS	Boot time		Runtim	e contexts:				
PCIE		(TF-A B	(OP-TEE	(U-Boot)	(Linux)	(TF-M)	(Cube)	A35 ROI	A35S	A35S			35NS	M33S	M33NS
PKA	~	Image: Control of the second secon	Image: Control of the second secon						" (TF-A E	IL2) (OP-T	EE) (U-I	Boot) (L	inux)	(TF-M)	(Cube)
PSSI									e Clock Sou						
✓ PWR	~						V			rce					
RCC		\$	1	\$	1	\$	1		le Calendar						
RESMGR_								Alarm A D	isable						\sim
RNG	\checkmark	~	~				_	Alarm B D	isable						\sim
RTC								Timest	tamp						
	1	~					_	WakeUp [Disable						\sim
SAI1 SAI2								Calibration	Disable						~
									nce clock de	te ell'e e					
SAI3								C Refere	псе сюск ае	tection					
SDMMC1								_			Confi	uration			
											Conni	Juración			
								RTC features	5						
										A35S	A35S	A35NS	A35NS	M33S	M33NS
SPDIFRX								Features	A35 ROM	(TF-A_BL2)	(OP-TEE)	(U-Boot)	(Linux)	(TF-M)	(Cube)
SPI1								Alarm A							
SPI2								Alarm B Wakeup ti							
								Timestamp							
SPI4								Calibration							
SPI5								initialization							

Figure 121. RTC features

Figure 122. RTC mode

		nfigura	ation			Clock	Configu	Iration			RIF			P	roject Ma
								✓ Software	Packs		🗸 Pin	out			
۹.				\sim			٢			R	TC Mode an	d Configuratio	n		
Categories A-	>Z										Mo	ode			
÷	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 BL	.2) (OP-TE	E) (U-E	loot) (L	inux)	(TF-M)	(Cube)
PSSI								a destinate	Clock Sourc						
PWR	~						1			e					
RCC		1	1	\$	1	1	1	Activate							
RESMGR_U								Alarm A Int	ernal Alarm A	4					\sim
RNG	~		~					Alarm B Di	sable						~
RTC								Timesta	Imp						
	~	✓	~					WakeUp D	isable						~
SAI1								Calibration							~
SAI2															÷
SAI3 SAI4								L Reterer	ce clock dete	ection					
SAI4 SDMMC1											Config	uration			
									_		Coning	uration			
SDMMC3								Reset Cont	iguration						
								Parameter	Settings	User Cons	stants 🛛 📀	Features	NVIC Set	ttings	
SPDIFRX															
SPI1								RTC features							
										A35S	A35S	A35NS	A35NS	M33S	M33NS
								Features		(TF-A_BL2)	(OP-TEE)	(U-Boot)	(Linux)	(TF-M)	(Cube)
SPI4								Alarm A							V
SPI5								Alarm B Wakeup ti							
SPI6								Timestamp							
								Calibration							



Home > ST	M32MP2	251CAlx	> •	1P25.ioc	- Pinout	& Confi	guration	\rangle						
Pinou	it & Co	onfigur	ation			Clock	Configu	iration		F	lF			Project Mar
								✓ Software P	acks		 Pinout 			
2				\sim			٢			RTC N	lode and Cor	nfiguration		
Categories A-	>Z -										Mode			
	A35 R	A35S	A35S	A35NS	A35NS		M33NS	Boot time:		Runtime cont	exts:			
PCIE	A33 N	(TF-A E	BL <mark>(OP-TE</mark>	(U-Boot)	(Linux)	(TF-M)	(Cube)	A35 ROM	A35S	A35S	A35NS	A35NS	M33S	M33NS
PKA	V	Image: A state of the state							(TF-A BL2)	(OP-TEE)	(U-Boot)	(Linux)	(TF-M)	(Cube)
PSSI														
PWR	~						~	Activate C						
RCC		1	1	V	~	1	~	Activate C	alendar					
RESMGR_U								Alarm A Inter	nal Alarm A					\sim
RNG	~	~	~								Configuratio	n		
RTC									_		Comgulatio			
SAES	~	~	~					Reset Config	iration					
SAI1								Parameter S	ettings 🛛 🥹	User Constant	s 🛛 📀 Feat	tures 🛛 📀 NV	/IC Settings	
SAI2								Configure the belo	w parameters	3 :				
SAI3								Q Search (Ctrl+I		0				0
SAI4	_							✓ Alarm A		Ŭ				v
SDMMC1								Hours			0			
SDMMC2								Minute	s		0			
SDMMC3								Secon	ds		0			
									econds		0			
SPDIFRX									Mask Date W	/eek day	Disat			
SPI1 SPI2									Mask Hours Mask Minute		Disat			
									Mask Minute Mask Second		Disat			
SPI3 SPI4									Sub Second I			larm SS fields	are masked.	
SPI5									Date Week D		Date			
SPI6								Hours						
SPI7								Hours must be b	etween 0 and	23.				
SPI8								Diagnostic:						
	-	_	-	-				10 Tel 10 4	011					

Figure 123. 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 for the ST software architecture, but are 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.



		& Configu	ration				Cloc	Configuration		RIF	Project Manager
								~	Software Packs	✓ Pinout	
		×					0		FMC Mode and Configuration		1
ategories A->Z									Mode		
0	A35 ROM	A35S	A35S (OP-TEE)	A35NS	A35NS	M33S (TE-M)	M33NS (Cube)		e contexts:		Parts and and and and
		(TF-A BL2)	(OP-TEE)	(U-Boot)	(Linux)	(IE-M)	(Cube)	A35 ROM A35S A (TF-A BL2) (0		35NS M33S M33NS Linux) (TF-M) (Cube)	(vos) (vos 0) (vos 0) (vos 0) (vos 0)
								(IF-A BL2)		Chox) (1P-W) (Coby)	
								> NOR Flash/PSRAM/SRAM/ROM/LCD 1			inara) inara) inara) inara) inara) inara) inana
								> NOR Flash/PSRAM/SRAM/ROM/LCD 2			INATATION
								> NOR Flash/PSRAM/SRAM/ROM/LCD 3			and a star a
BSEC	~	~	2			~		> NOR Flash/PSRAM/SRAM/ROM/LCD 4			
CORTEX M33	_	_	1		~	_		> NAND Flash 1			more more man man more care strend interest
CRC											AVER A LOS (VER) COLUMN AND A LOS (VER) COLUMN ADDRESS
CRYP1		~			V						
											AND D AVER D AVER D AVER D AVERAGE VERICE
CSI					V						
DCACHE											FILE FILE FOR FOR FOR VOR WEEK WEEK
DCMI					V						
DCMIPP					~				Configuration		- (PF14) (PG6) (PG10) (PG15) (PG) (PG15) (PG00) (VG6
DDR CTRL PHY		~	1						Coniguration		
DEBUG								FMC features			(PG) (PG) (PG) (PG) (PG) (PG)
				1	~						
DTS			V					Features A35 ROM A35S (TF-A_BL	A35S A35NS (OP-TEE) (U-Boot)	A35NS M33S M33NS (Linux) (TF-M) (Cube)	(NB) (PF12) (PG7) (PG7) (PH) (PB) (ND1
ETH1					~						
FMC		_	_	_	V	_	_	FMC NOR/PS			<pre>(ros) (NI2) (NI4) (rrIS) (ros) (rrI) (rrIS) \$0000</pre>
FREERTOS								FMC_CFGR		If all the features are assigned to one	context, then FMC_CFGR should be assigned to this context .
GIC			8	2	V			FMC_NOR/PS			(NE) (NE) (NE) (NEE) (NEE) (VEC)
GPIO								FMC_NAND			
HASH	~	×	~							<u> </u>	
HDP					~						
HPDMA1		1	1	2	~	~	1				
HPDMA2		1	2	1	1	1	1		V		
HPDMA3		1	1	2	~	~	1	0.1			
1201					~			Color-co	ding system and i	nstructions	BOOTS BOOTS MASTO VOS PUR ON PUR C. POR ON POR
											(OSC_O) (OSC_W) (PJS) (ANAG) (ANAG) (AGALE) (PCE
12C6											
											(RDD-T) (RJT) (RJT) (RDT) (
12C8											
											(NO12) (N
CACHE											
PCC1			V								Q [] Q
WDG1	1	~	~								~ ~

Figure 124. 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.

Figure 125. RIMU user interface

lome > STM32MP251DAlx	<u> </u>				_	
Pinout & Configuration	Clock	Configuration	า	RIF		Project Manage
Peripherals (RISUP)	Global lock	: OFF				
r enprieraie (raeer)	RIMU IP ≑	RIMU ID	CID SELECTION	MASTER CID	SECURE	PRIVILEGE
	DCMIPP	10		-		
	ETH1	6				
	ETH2	7				
	ETR	0	\checkmark	1		
Domains (RIMU)	GPU	9				
	LTDC_L1L2	11	\checkmark	4		
	LTDC_L3	12	V	4		
	LTDC_ROT	13	~	4		
	PCIE	8				
	SDMMC1	1				
xternal memories (RISAF)	SDMMC2	2		-		
	SDMMC3	3				
	USB3DR	4		-		
	USBH	5				
	VDEC	14 15				
	VENC	15		-		
nternal memories (RISAB) RIF-Aware IPs						

To define a CID for an IP, activate CID SELECTION, and then choose a value from 0 to 6, as shown in *Figure 126*.



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

Figure 126. 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 127*.

Pinout & Configuration	n Cloci	k Configuratior	1	RIF		Project Manag
Peripherals (RISUP)	Global lock					
	RIMU IP 🗘 DCMIPP	RIMU ID	CID SELECTION	MASTER CID	SECURE	PRIVILEGE
	ETH1	10 6		1	~	
	ETH2	7				
	ETR	0	✓	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 VDEC	5		-		
	VDEC	14 15		-		
	VENC	10				
Internal memories (RISAB)						

Figure 127. 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.

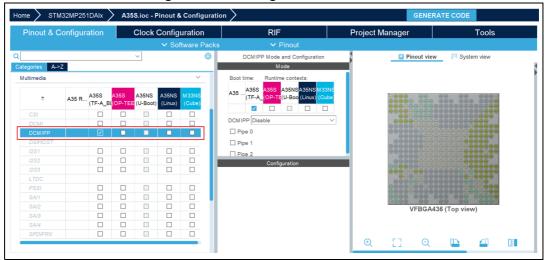


Figure 128. IP assignment to a context

Figure 129. Result in RISUP of an IP assignment to a context

Pinout & Configuration Clock		figuration		RIF	RIF		Project Manager	
	Global lock : OFF							
	Peripherals	ID	CID	Secure	Privilege	Lock		
	ADC12	58	-					
	ADC3	59	2					
	ADF1	55	-					
	COMBOPHY	67	-					
	CRC	109	-					
Domains (RIMU)	CRYP1	96	-					
Domains (Rivio)	CRYP2	97	-					
	CSI	86	-					
	DCMI PSSI	88	-					
	DCMIPP	87	1	\checkmark				
	DSI_CMN	81	-					
External mamories (DIRAE)	DSI_RDFIFO	123	-					
External memories (RISAF)	DSI_TRIG	122	-					
	DTS	107	1	\checkmark				
	ETH1	60	-					
Internal memories (RISAB)	ETH2	61	-					
	ETHSW_ACM_CFG	71	-					
	ETHSW_ACM_MSGBUF	72	-					
	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	-					
	1005	45						



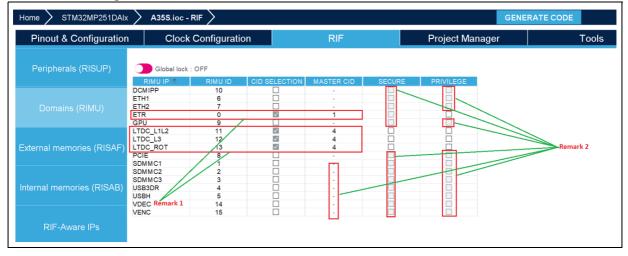
Pinout & Configuratio	n Cloci	k Configurati	on	RIF		Project Manage	er i i
Peripherals (RISUP)	Global lock : OFF						
	RIMU IP 💠	RIMU ID 10	CID SELECTION	MASTER CID	SECURE	PRIVILEGE	
	ETH1	6		-			
	ETH2	7					
	ETR	0	✓	1			
	GPU	9		-			
	LTDC_L1L2	11	~	4			
	LTDC_L3	12	✓ ✓	4			
	LTDC_ROT	13	\checkmark	4			
	PCIE	8		-			
	SDMMC1	1		-			
xternal memories (RISAF)	SDMMC2	2					
	SDMMC3	3					
	USB3DR	4		-			
	USBH	5		-			
	VDEC	14		-			
	VENC	15		-			

Figure 130. 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 131. 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 132. 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 133*.



Pinout & Configu	ration	Clock Configuration	n	RIF	Project Manager	Tools
Ť	EXTI	Resource CID Semaphores				
RIF Configuration		Features Group 0 Semaphore 0	Feature_ID	CPU_WL	SECURE	PRIVILEGE
	FMC	Group 0 Semaphore 0 Group 0 Semaphore 1 Group 0 Semaphore 2	1			7
		Group 0 Semaphore 3 Group 1 Semaphore 4	3			
Peripherals (RISUP)	GPIO	Group 1 Semaphore 5 Group 1 Semaphore 6	5		2	√
		Group 1 Semaphore 7 Group 2 Semaphore 8	7 8		2 2	
		Group 2 Semaphore 9 Group 2 Semaphore 10	9 10		 2	
	DMA	Group 2 Semaphore 10 Group 3 Semaphore 12	11 12			
		Group 3 Semaphore 13	13			51 52
		Group 3 Semaphore 14 Group 3 Semaphore 15	14 15			¥ ¥
	IPCC PWR					
	RCC					
	RTC					
	TAMP					

Figure 133. RIF HSEM panel

TAMP protection

TAMP for the STM32MP2 devices (*Figure 134*) 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 135*). 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 & Configu	ration	Clock	Configuration		RI	F	F	Project Manag	er		Tools	
	EXTI	TAMP TAMP_BKP_	REG									
RIF Configuration		TAMP BKP_REG Zones	Sub-Zone Name	Start Address	Sub-Zone Size	Nb Backup Registers		tesource 0	CID of Re			esource 2
	FMC						NS	S	NS	S	NS	S
		Zone1 ReadS WriteS	Zone1-RIF1	0x46010100	0x200	128				RW		
			Zone1-RIF2	0x46010300	0x0	0			RO	20	20	RW
Peripherals (RISUP)	GPIO	Zone2 ReadNS WriteS	Zone2-RIF1 Zone2-RIF2	0x46010300 0x46010300	0x0 0x0	0			RO	RO	RO	RO
			Zone3-RIF1	0x46010300	0x0	0	RO	RO	RW	RW	RO	RV
		Zone3 ReadNS WriteNS	Zone3-RIF0	0x46010300	0x0	0	RW	RU	RO	RO	RO	RO
	DMA	Zones Readins writeins	Zone3-RIF0 Zone3-RIF2	0x46010300 0x46010300	0x0	0	RO	RW	RO	RO	RW	RW
	HSEM IPCC	Automaticall change	y adapted to :	sizes	Nb Backup	registers = HextoDe						
	IPCC		y adapted to a	sizes	Nb Backup	The Re Sub-Zo	ad/Write	access right le users (co				
	IPCC		y adapted to a	sizes	Nb Backup	The Re Sub-Zo	ad/Write a	access right le users (co				
	IPCC PWR		y adapted to :	sizes	Nb Backup	The Re Sub-Zo	ad/Write a	access right le users (co				

Figure 134. RIF TAMP panel (STM32MP2 devices)

Figure 135. RIF TAMP panel (STM32N6 devices)

STM32CubeMX Untitled: STM32N645A M32 SubeMX	^{DHxQ} File	Window	Help		🕸 🖬 🗖	 X () @ \
Home > STM32N645A0HxC) Vntit	iled - RIF >				RATE CODE
Pinout & Configuration	С	lock Configuration	RIF	Projec	t Manager	Tools
Peripherals (RISUP)	EXTI1	Features Tamper protection (excluding monoto Monotonic counter 1 secure protectio		SECURE		
	GPDMA1	Backup registers write protection offs Backup registers read/write protection	et		3 <u>2</u> 0	
Domains (RIMU)	GPIO					
	HPDMA1					
Interrupts (IAC)	PWR					
ixternal 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 136. 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	TM32MP:	251CAlx) м	P25.ioc	- Pinout	& Confi	iguration	
		Pinout	& Con	figurati	on			Clock Configuration
								✓ Software
Q				\sim			Ô	I2C1 Mode and Configuration
Categories 🛛 A	->Z							Mode
÷	A35 R	A35S	A35S		A35NS		M33NS	Boot time: Runtime contexts:
TREEKTUS	A33 K	TF-A BL	(OP-TEE	(U-Boot)	(Linux)	(TF-M)	(Cube)	A35 ROM A35S A35S A35NS A35NS M33S M33NS
GIC			\checkmark	~	\checkmark			(TF-A BL2) (OP-TEE) (U-Boot) (Linux) (TF-M) (Cube)
GPIO								
HASH	\checkmark	~	\checkmark					12C 12C ~
HDP					~			Disable
HPDMA1		\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	I2C
HPDMA2		\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	SMBus-Alert-mode Res SMBus-two-wire-Interface
HPDMA3		\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	ResSMDus-two-wite-intenace
✓ I2C1								Parameter Settings
12C2								
12C3								Search Signals
12C4								Search (Ctrl+F)
12C5								Pi Signal Pin C GPIO GPIO Maxi Retime Invert Doubl Delay Delay I
12C6								Pi Signal Pin C GPIO GPIO Maxi Retime Invert Doubl Delay Delay D PG13 I2C1 A35NS Altern No pu Low n/a n/a n/a n/a n/a n/a r/a
12C7								PI1 I2C1 A35NS Altern No pu Low n/a n/a n/a n/a n/a n/a r
12C8								

Figure 137. IO protection inheritance for a non-RIF-aware IP (I2C)

Figure 138. GPIO IP panel

		Pinout	& Conf	figurati	on					Clock	Configu	ration			
															 Software
Q				\sim			0			GF	PIO Mode a	ind Configura	ation		
Categories A	->Z										M	lode			
÷	A35 R	A35S (TF-A BL			A35NS (Linux)	M33S (TF-M)	M33NS (Cube)								
GIC			~	~	~										
GPIO											Confi	guration			
HASH	~	~	~					Group By P	eripherals						~
HDP					~			⊘ GPI0	⊘ I2C	⊘ RTC (≥ NVIC				
HPDMA1		\checkmark	\checkmark	\checkmark	~	\checkmark	\checkmark	GPI0		VRIC		♥ Features			
HPDMA2		\checkmark	~	\checkmark	v	\checkmark	\checkmark	GPIO featu	100						
HPDMA3		\checkmark	~	\checkmark	~	\checkmark	\checkmark	GFIO lealt	lies						
I2C1					~			Features	A35 ROM	A35S	A35S	A35NS	A35NS	M33S	M33NS
12C2										(TF-A_BL2)		(U-Boot)	(Linux)	(TF-M)	(Cube)
12C3								PG8 PG9							
12C4								PG10							
12C5								PG11 PG12							
12C6								PG12 PG13					~		
12C7								PG14							
12C8								PG15 PH2							
12S1								PH3							
1282								PH4							
								PH5 PH6							
I3C1								PH7							
/3C2								PH8 PH9							
								PH9 PH10							
I3C4								PH11							
ICACHE							V	PH12 PH13							
IPCC1			V					PI0							
IWDG1	~	~	~					PI1					~		
IWDG2	~			~	~			PI2 PI3							



ome 🔰 STM32MP251CAlx	\rightarrow $^{\prime}$	MP25.ioc - RIF	\geq		GENERATE	CODE
Pinout & Configuration	С	lock Config	uration	RIF	Project Manager	Tools
	EVT					
RIF Configuration	EXTI	Features	Feature_ID		SECURE	PRIVILEGE
Configuration		PG11	11	0	\checkmark	
	FMC	PG12	12	0		
		PG13	13	1		
		PG14	14	0		
Peripherals (RISUP)	GPIO	PG15	15	0		<u>L</u>
		PH2	2	0		<u>L</u>
		PH3 PH4	3	0	✓ ✓	<u>_</u>
	DMA			0	✓ ✓	<u>_</u>
		PH5 PH6	5	0	✓ ✓	
Domains (RIMU)			6	U	× · · · · · · · · · · · · · · · · · · ·	<u>H</u>
	HSEM	PH/ PH8	8	0	×	
		PH8 PH9	9	0	× · · · · · · · · · · · · · · · · · · ·	
		PH9 PH10	10	0	✓	
	IPCC	PH10 PH11	11	0	✓	
External memories (RISAF)		PH12	12	0	✓	
		PH13	13	0	 ✓	
	PWR	PIO	0	0	 ✓	
		PI1	1	1		
	RCC	PI2	2	0		✓
nternal memories (RISAB)	Rec	PI3	3	ő	 ✓	
		PI4	4	0		
	RTC	PI5	5	0		
	- THE	PI6	6	0		
		PI7	7	0	V	
	TAMP	PI8	8	0	✓	
		PI9	9	0	\checkmark	
		PI10	10	0	✓	
		DI11	44	0		

Figure 139. Inheritance in RIF GPIO panel

Figure 140. PIN reservation

	Pi	nout &	Config	uration			Clock Configuration	Project	Manager	Tools
							✓ Software Packs	Pinout		
Q	\sim				٥		GPIO Mode and Configuration		1	Pinout view
Categories	A⊢>Z				_		Configuration			
•	Boot R	TF-A B	OP-TEE	U-Boot	Linux	Group By Peripherals		~		
						GPI0 Single Mapped Si	gnals 🔮 I2C			
									1	~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
						Search Signals				
						Search (Ctrl+F)		Show only Modified Pins		Enter User Label 100000000000
DOR CTRL						Pin Name 🍨 Signal on Pir	GPIO mode GPIO Pull-up/Pul Maximum output	User Label Modified		Signal Unpinning
DEBUG						PB10 n/a	External Interrupt No pull-up and n n/a			Pin Stacking
						PD0 n/a	Input mode No pull-up and n n/a			Pin Reserved O Free
DTS									666666	O CortexA35S Secure OS
									(max)	CortexA35NS SSBL
										CortexAsons OS
										Contex/ii333 Secure Cis
									00000	
FMC			1							
GPIO			2							
GPU									and the second the second the second the second the second the second terms of terms	
						/PB10 Configuration :				
HDP						and a stanger all off.				
HPDMA1										
HSEM										
						GPIO mode	External Interrupt Mode with Rising edge trigger	detection ~		
						GPIO Pull-up/Pull-down	No pull-up and no pull-down	~		
✓ I2C3						or to i an oper diridowit	the parried and the pull-down	÷		TFBGA361 (Top view)
						User Label				
					H					
			L.	-	-					

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 141*.



	File		Win	fow	Help				3	📲 🖸 X 🔿 @ -}	< A 77
me > 57X32N545A2	na > Un	titled - Pincu	t & Configur	ution >						GENERATE CODE	
Pinout & Config	guration		Clock	Configuration		RIF		Project Manage	ы	Tools	
			÷	0		HPQ	NA1 Mode and Config	wation		O Phone view	System view
riegories A-+2							Mada				
System Core				~		vite Internet R.FD / 2D addre			Ŷ		
	F10.					ords Internal RIFO / 20 addre	-		~		
CORTEX_MISS_FISEL	100	Appli	AppMG	Editoricador		ords Mernel PIPO / 20 eddre			¥		
CORTEX, WSS, NS					Chemiel 11 - 4 We	nds Internal FIPO	Deeble		Ψ.		
CORTEX_N65_S					Channel 10 - 4 Wo	rds Internal FIPO	Creatrie		~		
OPDMA1	8	8	8		Channel 9 - 4 World	ta imamai f1/F0	Dauble		~		
GPIO					Channel 8 - 6 Word	ta Internal PLFO	Deathle		~		
HOTOMAN					Channel 7 - 4 Work		Disable			THE OWNER WATER OF	and the second se
	0		0	0	Charriel 8 - 6 Third		Charles			and a state of the	2222
ANDG	0		0	0			Chemister		*	0.0000000	
NVNC1_8_Application			8		Annel Alter	An Informati FLATR	Configuration				
NVIC2_N5_Application NVIC_F58L	8					-				STATE & STATE OF	2222
AMICEO	0	0	0	0	Renet Configuration						6666
ROC		0	0	0	Testates					Contraction (Contraction)	
¥ 615,N5			5	-							1000
					HPORA1 Returns					0000-000	
WWD9	0	0	0	0	Ferrures	First Steps Boot Loader	Annual Annual Sector	No. Sec	ErMericader	0000-0000	0000
						First page boot cases.	Sector Approxim	Non-Secare Application	LowerLoader		
				,	HPOMA1 channel 0 HPOMA1 channel 1	8	8	8		VFBGA165 (Tep	viterec)
Analog				,	HPOMA1 channel 2	8	8	8			
Trees				>	HPCM A1 channel 3 HPCM A1 channel 4	8	8	8		1	
					HPOMA1 channel 5	8	8	8			
Connectivity				>	HEOMA1 channel 6 HEOMA1 channel 7	8	8	8		1	
Multimedia					HPOMA1 channel 2	8	8	8			
1211034				,	HEOMA1 channel 9	8	8	8		1	
Security				>	HFOMA1 channel 10 HFOMA1 channel 11	8	8	8		11	
					HEOMA1 channel 12	Ŭ.	D D	D D		1	
Computing				>	HFOM A1 channel 13 HFOM A1 channel 14		- 8	8		0 0 0	2 🕒

Figure 141. 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		R	IF			Project N	Manag
								Software Pac			Pinout				
				\sim			٥			12C1 M	ode and Co	nfiguration			
ategories 🛛 A	λ->Ζ										Mode				
÷	A35 R	A35S (TF-A E	A35S BL(OP-TEE	A35NS (U- <u>Bo</u> ot)	A35NS (Linux)	M33S (TF-M)	M33NS (Cube)	Boot time: A35 ROM	A35S	Runtime cor A35S	A35NS	A35NS	M33S	M33NS	
HPDMA1		\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	ASSINON	(TF-A BL2)		(U-Boot)	(Linux)	(TF-M)	(Cube)	
HPDMA2		\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark								
HPDMA3		\checkmark	\checkmark	\checkmark	\checkmark	\checkmark		12C 12C						\sim	
I2C1															_
12C2											Configuratio	ก			
I2C3					~			Denot Orac							
12C4								Reset Config	uration						
12C5								📀 Parameter S	iettings (🥝 GIC Setting	ys 🛛 🔗 Di	MA Settings	GPIO Setti	ngs	
12C6															- 17
12C7															
12C8															
I2S1								12C1	requests sl	nould be confi	gured in HP	DMA1 or HPDN	AA2 or HPDMA	43	
									Go to HPI	DMA1 0	o to HPDN	1A2 Co to	HPDMA3		
									00101111						
I3C1															
I3C2															
1202															

Figure 142. I2C IP panel



Home > S	TM32MP2	51CAlx) м	P25.ioc	- Pinout	& Confi	guration	\rangle						GENERA
Pino	ut & Cor	nfigura	ation			Clock (Configur	ation		RIF		Project N	lanager	
								 Software Pack 						
Q				\sim			0			HPDMA	1 Mode and Co	nfiguration		
Categories 🖊	k->Z										Mode			
¢		A35S (TF- <u>A</u> B	A35S L(OP-TEE			M33S (TF-M)		Boot time: A35 ROM	A35S	Runtime contexts: A35S	A35NS	A35NS	M33S	M33NS
HPDMA1								A33 ((0))	(TF-A BL2)	(OP-TEE)	(U-Boot)	(Linux)	(TF-M)	(Cube)
HPDMA2		\checkmark	1	\checkmark	\checkmark	1	1				\checkmark		¥	
HPDMA3		\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	~	Channel 15 - 3	2 Words Internal FIFC	0 / 2D addressing Sta	andard Request	Mode		~
I2C1					~			Channel 14 - 3	2 Words Internal FIFC) / 2D addressing Dis	able			~
								Channel 13 - 3	2 Words Internal FIFC) / 2D addressing Dis	able			~
I2C3					V						Configuration			
12C4									_		oomgananon			
								Reset Configu	ration					
12C6								All Channels	SECURITY	⊘ CH15 Subserver	Constants 6	🛛 Features 🛛 😒 NV	/IC Settings	
								Configure the belo						
12C8								Q Search (Ctrl+F						0
								 Circular config 						U
								Circular coning Circula			Disable			
								✓ Request Confi			Disable			
I3C1								Reques			SoftWa	are		
									andle in IP Structure		NONE			
								Block I	W request protocol		Single/	Burst Level		
13C4								 Channel config 	uration					
/ ICACHE								Priority			Low			
IPCC1	_		V						tion Mode		Normal			
IWDG1	~							Directio			Memor	y To Memory		
IWDG2	1			~	V	_	_	Source Data S	etting Address Increment A	Ass Transfer	Disable			
IWDG3								Data W		ater i fansier	Disable Byte	ru -		
IWDG4	_	_	_	_	-			Burst L			1			
LPDMA1	1								ed Port for Transfer		Port 0			
								✓ Destination Da	ta Setting					

Figure 143. I2C mode panel

Figure 144. I2C features panel

Pino	ut & Co	onfigura	tion		(Clock C	Configura	ation	I	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		~	~	~	~	~	~		✓	<	~	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	~			×	~			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 🔰 STM32MP251CAL	× > N	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	✓	
		HPDMA1 channel 1	1	0	✓	
	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 145. 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 146*).

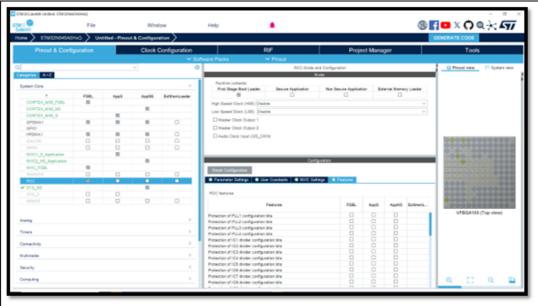


Pinout & Configu	ration	Clock	Configuration			Project Manager		Tools
	EXTI	Features CK_KER_USB2PHY1	Feature_ID 57	CID	SECURE	PRIVILEGE	LOCK	System Mode
RIF Configuration		CK_KER_USB2PHY2	58			H		
+ · · · · · · · · · · · · · · · · · · ·		CK_ICN_M_GPU	59	1	2	H		
		CK_KER_ETHSWREF	60			H		
	FMC	CK_MCO1	61			8		
		CK_MCO2	62			H		
		CK_CPU1_EXT2F	63		2	H		
		CK_SYS_PLL4,5,6,7,8	64			H		
	0010	FCALC	65			H		
	GPIO	SYSRST	66	- 1				
		BOOT_STDB	67		21 21	H		
		RDCR	68		21 21	H		
		SYSCLK	69	1	2	H		
	DMA	CPU1_RES	70		21 21			
		CPU1_RES CPU2_RES	70	1	24			
		CPU2_RES CPU3_RES	72			H		
		DEBUG_CFGR	73					
	LIGEN	SYSRAM_CFGR	73		V			
	HOEM	VDERAM_CFGR	75		<u>×</u>			
		RETRAM_CFGR	76			H		
		BKPSRAM CFGR	77	182		H	H	
		SRAM1_CFGR	78	182		H		
	IPCC	SRAM2_CFGR	79			H		
		LPSRAM1_CFGR	80			8		
		LPSRAM1_CFGR	81					
		LPSRAM2_CFGR LPSRAM3_CFGR	82	:				
		HPDMA1_CFGR	83					
	FVVR	HPDMA1_CFGR HPDMA2_CFGR	83					
		HPDMA2_CFGR HPDMA3_CFGR	85				8	
		LPDMA_CFGR	85					
		LPDMA_CFGR	87					
ternal memories (RISAB)	RCC	IPCC1_CFGR IPCC2_CFGR	88					
		HSEM_CFGR	89					
		HSEM_CFGR				H		
		GPIOA_CFGR	90		<u>1</u>			
	RTC	GPIOB_CFGR	91			8		
	KI0	GPIOC_CFGR	92					
		GPIOD_CFGR	93					
		GPIOE_CFGR	94			8		
		GPIOF_CFGR	95			8		
	TAMP	GPIOG_CFGR	96					
		GPIOH_CFGR	97			8		
		GPIOI_CFGR	98					
		GPIOJ_CFGR	99					

Figure 146. 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.

CubeMX	File	Window	Help	🐥 🛛 🕸 📑 🕒	× 🗘 🔍 🔆 🖅
Home > STM32N645A0HxC	2 🔪 Unti	itled - RIF >		GENERA	TECODE
Pinout & Configuration	c	lock Configuration	RIF	Project Manager	Tools
Peripherals (RISUP)	EXTI1	Features EXTI0 EXTI1 EXTI2	EXTI index 0 1 2	SECURE	
	GPDMA1	EXTI3 EXTI4 EXTI5	2 3 4 5		
Domains (RIMU)	GPIO	EXTI6 EXTI7 EXTI8 EXTI9	6 7 8 9		
	HPDMA1	EXTI10 EXTI11 EXTI12	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	I3C1 wake-up I3C2 wake-up USART1 wake-up	26 27 28		
RIF-Aware IPs	TAMP	12C4 wake-up 13C1 wake-up 13C2 wake-up	25 26 27		

Figure 148. RIF panel for EXTI1 (STM32N6 MCUs)

4.7.10 Memories 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	D0	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 150. 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.

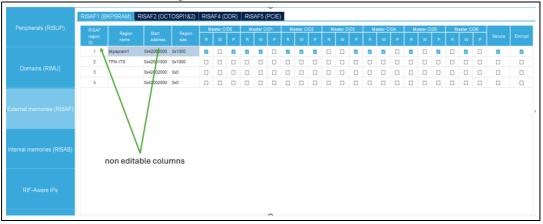
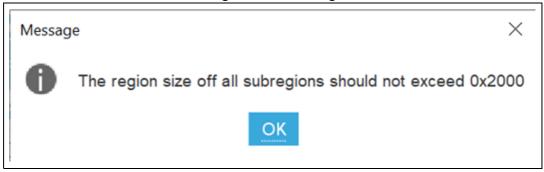


Figure 151. Non editable columns

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



Figure 152. 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		Master CID2	£		
region ID	name	address	size			w	R		P	Secure	Encrypt
.1	mm_ospi1	0x60000000	0x0000								
2	mm_ospi2	0x60000000	0x10000000								

Figure 153. 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 154*). The start address changes automatically.



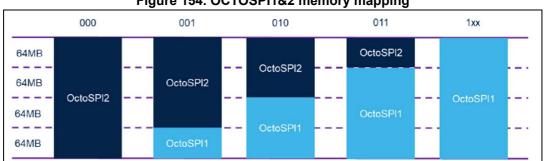




Figure 155. OCTOSPI1&2 region size configuration

me 🔪 STM32MP251CAI	MP25.ioc	- RIF 🔪										GENERAT	TE CODE	
Pinout & Configu	ration	Clock Co	onfiguration		í	RIF		Proj	ect Mana	ager			Tools	
RIF Configuration	RISAF1 (BKF	SRAM) RISAF2	(OCTOSPI1&2)	RISAF4 (DDR)	RISAF	5 (PCIE)								
WRIP Conliguration	RISAF	Region	Start	Region		Master CID	0	Master CID1			Master CID2	2		
	region ID	name	address	size						R			Secure	Encrypt
	1	mm_ospi1	0x60000000	0x8000000 ~									2	
	2	mm_ospi2	0x68000000	0×0 0×4000000									~	
				0x8000000 0xC000000 0x10000000										

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 156.

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	ID 120 1	CID	Secure V	Privilege			RIF Configuration	RISAF					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 156. 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.



Misson pp Region pp Start Section Region Pp No P Start Conc Pp Concrete Pp Pp Concrete Pp Pp Concrete Pp Concre Pp Concre Pp Concre<	
2 mm_ospi2 0x6400000 0xc000000	
Domains (RIMU)	

Figure 157. 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.

	RISAF	-	(and and	econo i	М	aster C	DØ	M	ster C	D1	M	sster C	ID2	Ma	ster C	D3	M	ster CI	D4	Ma	aster Cl	DS	M	ster C	D6	
	region ID	Region name	Start address	Region size	R	w	P	R	w	P		w	P	R	w	P		w	P		w	P	R	w	P	
	1	TFM-code	0x80000000	0x100000																						
	2	CM33-Cube-fw	0x80100000	0x800000																						
	3	TFM-date	0x80900000	0x100000																						
	4	CM33-Cube-data	0x80a00000	0x800000																						
External memories (RISAF)	Б	ipc-shmem	0x81200000	0x100000																						
	6	spare1	0x81300000	0xCC0000																						
	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																						
	15	linuxkernel2	0x100000000	0x80000000																						

Figure 158. 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.



		on	Clo	ock Config	uration		RIF	Project Manager	Tools
<u> </u>						٥	DDR_CTRL_PHY Mode	and Configuration	Pinout view III System view
Categories A->Z							Mode		
۰	A35 ROM	A35S (TF-A_BL2)	A35S (OP-TEE)	A35NS (U-Boot)	A35NS (Linux)	M33NS (Cube)	Boot time: Runtime context	ISINS A35NS M33NS	
DCACHE						V	A35 ROM (TF-A BL2) (OP-TEE) (U	-Boot) (Linux) (Cube)	
DCMI					V				
DCMIPP					V		DDR Type DDR4	~	
DDR_CTRL_PHY							DDR Width 32 bits		
DEBUG							DDR Density 16Gb		
				1	1		DDR Density 16Gb	¥	
DTS			V						
ETH1					V				
FILEX									
FMC									
GIC			1		1		Configurati		
GPIO								User Constants	
HASH	\checkmark	V	2					o oser constants	
HDP					V		Configure the below parameters :		
HPDMA1		1	1	1	1	1		how Advanced Parameters 👔	
HPDMA2		1	2	1	1	1	✓ ⊗ SYSTEM PARAMETERS		
HPDMA3		1	1	1	1	1	Speed_Bin Wor	se	
							* ATx Impedance 30		VFBGA361 (Top view)
							* ODT Impedance 80		
							* Tx Impedance 30		
12C4									
12C5									
12C6									ର୍ 📋 ବ୍ 🕒 🕮
									ex La ex la la

Figure 159. DDR_CTRL_PHY activation

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						Ma	aster Ci	D0	Ma	aster C	D1	Ma	ster C	D2	M	aster C	D3	M	aster C	D4	М	aster C	D5	M	aster Cl	D6
		Region name	address	size	Encrypt																					
	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																						
Domains /DIMU	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																						
ternal memories (RISAF)		vdec-reserved	0xfe800000	0x0000																						
	12	venc-reserved	0xfe800000	0x0000																						
	13	LTDC-sec-layer	0xfe800000	0x800000																						
	14	LTDC-sec-rotation	0xff000000	0x1000000																						
ernal memories (RISAB)																										

Figure 160	. Configuration	of RISAF4	(DDR)
------------	-----------------	-----------	-------

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	DO	Ма	ster CID1		laster C	1D2	Mas	er CID3		laster C	ID4	Ma	aster CI	D5	M	aster CI	D6	
region ID	Region name	Start address	Region size	R	w	Р	R	w	P R	w	P	R	W F	R	w	P	R	w	Р	R	w	P	
1	PCIE-device	0x10000000	0x10000000																				
2		0x20000000	0x0																				
3		0x20000000	0x0																				
4		0x20000000	0x0																				

Figure 161. PCIE memory configuration

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 Memories 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.

Figure 162. Global lock in RISAF panel for STM32N6 MCUs

Global lock				¢					
Pinout & Configuration	Cloc	k Configurati	on	RIF		Project Ma	nager	т	ools
Peripherals (RISUP)	Global loci RISAF 12(X		13(XSPI3)	RISAF 14(FMC)	RISAF 15(CACHEAXI cont	figuration port) RISAF 21(AHB RAM1) 🔽
	Memory	regions conf	iguration	Memory sub-re	gions config	uration			
	Region ID	Region name	Start Address C	D 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
xternal memories (RISAF) RIF-Aware IPs	non edi	table column							



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.



i32 🍞 be MX	File		Window	Help			🕸 🕇 🖻) 🗶 🎧 🔍	\times
me 🔪 STM32N645B0Hx0	D Untitled						GENER	ATE CODE	
Pinout & Configuration	Clock	k Configurati	on	RIF		Project Ma	nager	Т	ools
	Global lock	: OFF							
Peripherals (RISUP)	RISAF 1(TCI	() RISAF 2(CPU AXI RAM0)) RISAF 3(CP	U AXI RAM1)	RISAF 4(NP	U master 0)	RISAF 5(NPU	master 1)
	Memory	regions confi	iguration Me	mory sub-reg	ions 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
	Û								
	Non editable parameter	7							

Figure 163. 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 164*).

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 165*).

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





	egions configur	ation Memor	y sub-regions	configuration						
RISAF region	SubRegion name	Start Address Offeet	Region size	SubRegion CID	Delegated CID	Delegation enabled		Write	Privilege	
	A	0x0000	0x0000	0	0					
1	в	0x0000	0x0000	0	0					
2	A	0x0000	0x0000	۰ ۲	0					
£.	в	0x0000	0x0000	0	0					
3	A	0x0000	0x0000	0	0					
3	в	0x0000	0x0000	0	0		13			
4	A	0x0000	0x0000	0	0					
	В	0x0000	0x0000	0	0					
5	A	0x0000	0x0000	0	0					
<i>u</i> .	в	0x0000	0x0000	0	0					
6	A	0x0000	0x0000	0	0					
	В	0x000x0	0x0000	0	0					
7	A	0x0000	0x0000	0	0					
1 C	в	0x0000	0x0000	0	0					E3

Figure 164. Sub-regions activation in RISAF (showing activated subregions)

Figure 165. Sub-regions activation in RISAF (check the filtering parameter)

ome > STM32N54580Hx	D Vintitlec	1 - RIF >					GENE	RATE CODE	
Pinout & Configuration	Cloc	k Configurati	on	RIF		Project Ma	nager	Τc	ools
Peripherals (RISUP)	Global loc RISAF 1(TC	100,000,000	CPU AXI RAMO	RISAF 3(C	PU AXI RAM1)	RISAF 4(NP	U master 0)	RISAF 5(NPU	master 1)
	-	regions confi			gions configur	_			
Domains (RIMU)	Report D	Report Name	Ditert Address O Dx0000	0x0000	Fillering	Decure C	Heed 0	0	Prolege
	2	region 2	0x0000	0x0000	0		0	0	0
	3	region 3	0x0000	0x0000			٥	0	0
Interrupts (IAC)	4	region 4	0x0000	0x0000			0	0	0
monupes (we)	5	region 5	0x0000	0x0000			0	0	0
	6	region 6	0x0000	0x0000			0	0	٥
	7	region 7	0x0000	0x0000			٥	٥	0
ternal memorias (RISAF) RIF-Aware IPs									

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.



UM1718 Rev 46

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 cproject name>-mx-rif.dtsi.



Figure 166. Example: RISUP configuration and generated code

Figure 167. Example: RISAF configuration and generated code

Pinout & Co	nfiguration		• Clo	ick Configur	ation				R	IF					Proje	oct Man	ager					Т	ols						
O RIF Configuration	RISAF1 (B	HPSRAM RI	SAF2 (OCTOS	PI182) RIS	AF4 (DDR) RISA	AF5 (PCI	E)																					
	RESAF report	Region	Diat editess	Region	R	dae CECO	P R	Master CID	n P	R	W F	- 4	Master	CEU P	5	Aastes CED W	4 P	R	w CBD	6 E	Wash R	er C406 W P	Seco	e Except					
	3.	bi31_lexeposer	0x42000000	6x1000		0 1		8	23		0 0	1 0	D	1 🗆							3 1		5						
	2	TPM/TS	0x42001000	8x1000	-	0 1		0	0						0	0	-	0	0				8	0					
	4		9+42902000	Ex0	0	0	0 0	0	0				0	1 0					0			0 0		0					
	2																							_					
																							- Г						
	<pre>}: &bl31_low RIF_BREN }: &tfm_its{ st.fr }: &rrisaf2{ men }: &rrisaf5{ mem }: &pcie_den</pre>	st.protreg <u>EN</u> >; protreg = <r nory-region = i2{ otreg = <r!s ory-region =</r!s </r 	= <risaff ISAFPROT = <&mm_os SAFPROT(f <&poie_de = <risaff< th=""><th>PROT(RIS/ (RISAF_R spi2>; RISAF_RE evice>; PROT(RIS/</th><th>AF_REG EG_ID(2 G_ID(2).</th><th>3_1D(1) 2), RIF</th><th>F_CID2</th><th>_BF, R</th><th>IF_CI</th><th>D2_BF</th><th>RIF_</th><th>CID2_</th><th>BF,</th><th>RIF_SE</th><th>EC, R</th><th>IF_EN</th><th>C_DI</th><th>S, RI</th><th>F_BR</th><th></th><th>)>;</th><th></th><th>Į</th><th>ļ</th><th>GE</th><th>NEF</th><th>RATE</th><th>E CO</th><th>DE</th></risaff<></risaff 	PROT(RIS/ (RISAF_R spi2>; RISAF_RE evice>; PROT(RIS/	AF_REG EG_ID(2 G_ID(2).	3_1D(1) 2), RIF	F_CID2	_BF, R	IF_CI	D2_BF	RIF_	CID2_	BF,	RIF_SE	EC, R	IF_EN	C_DI	S, RI	F_BR)>;		Į	ļ	GE	NEF	RATE	E CO	DE

UM1718 Rev 46



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.

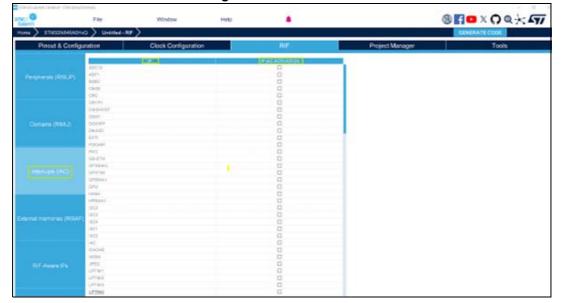


Figure 168. IAC feature

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).



UM1718 Rev 46

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 169*
- 2. From each component mode panel, opened by clicking the component name.

Figure 169. STM32H7 dual-core: peripheral and middleware context assignment

32 ² T	File		Window	v	Help			
ome 🔪 STM32H7	47IGTx	angle Sha	rk_dual_core_G	PIOContext	ts.ioc - Pinout & Configur	ation		
	Pinout	$\alpha \cup 0$	Click the gea Select to disp		text and/or power d	figuration omains	✓ Pinout	Proj
2 ~)	Cont	exts		- X _@		DAC1 Mode and Configuration	, mout	4
Categories A->Z					Show contexts	Mode		
	M7	M4	Initializer		Show power domain	mode		
CORTEX_M/	WI /	1914	Cortex M/	Domain	Cortex-M7	Cortex-M4	PowerDomain	
CRC			V	D3		501102-1114	D2	
DAC1	\checkmark		Cortex-M7	D2	OUT1 mode Disable		02	
DCMI				D2				
DEBUG	\checkmark	\checkmark	Cortex-M7 ~		OUT2 mode Disable			~
DFSDM1				D2	External Trigger			_
DMA				D2				_
DMA2D				D1				_
DSIHOST				D1				_
ETH				D2				_
FATFS_M4		\checkmark	Cortex-M4					
FATFS_M7	\checkmark		Cortex-M7					
FDCAN1				D2				
FDCAN2				D2				
FMC				D1				
FREERTOS_M4		\checkmark	Cortex-M4					
FREERTOS_M7	\checkmark		Cortex-M7					
GFXSIMULATOR	\checkmark	\checkmark	Cortex-M7 ~					
GPIO			Cortex-M4			Configuration		

For GPIOs (see *Figure 170*), assignment is done through the **Pinout** view directly or later and automatically through its selection in the platform settings panel of a middleware.

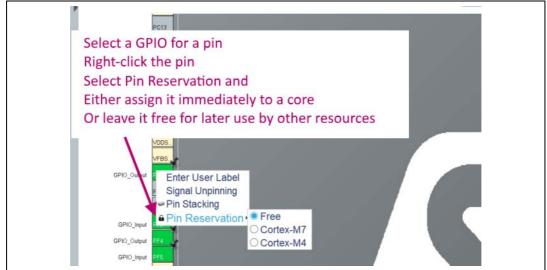


Figure 170. STM32H7 dual-core: GPIOs context assignment



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 171*).

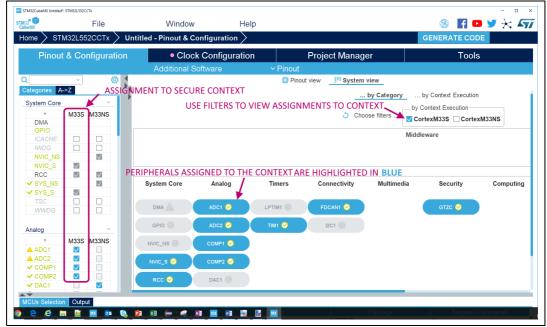


Figure 171. Pinout & Configuration view for TrustZone[®]-enabled projects

4.9.1 Privilege access for peripherals, GPIO EXTIs and DMA requests

Independently of TrustZone[®], STM32CubeMX enables privilege access:

- for each peripheral: in the GTZC configuration panel (see *Section 4.9.5*), as shown in *Figure 172*
- for each GPIO EXTI: in the GPIO configuration panel, as shown in *Figure* 173
- for each DMA channel: in the DMA configuration panel (see Section 4.9.4), as shown in *Figure 174*.



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 175). 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.

CT7C Med	e and Configuration
GIZC Midd	Mode
Runtime contexts:	Cortex-M33 non secure
> GTZC	
Co	nfiguration
Reset Configuration	
😔 User Constants	⊘ NVIC Settings
Block-Based Memory Protection Controller	TrustZone Illegal access Controller
	Memory Protection Controller WaterMark
	ntroller - Privilegeable Peripherals) ustZone Security Controller - Securable Peripherals
Configure the below parameters :	usizone decunty controller - decurable r enprierais
Conligure the below parameters .	
Q Search (CrtI+F) 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 privilogod

Figure 172. Setting privileges for peripherals



			Ci	onfiguratio	n			
Group E	By Periphera	als						~
😔 GPI	D 📀 UA	RT 🛛 📀 NV	1C					
Search								
Search	(CrtI+F)					Show o	only wo	dified Pin
	Signal o.		Pin Privilege access	GPIO o		GP Ma.	Fa	Us 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	~
PH1-0S	n/a	Cortex	n/a	n/a	Input mode	No n/a	n/a	 ✓
PC13 Co	onfiguration	:						
	onfiguration text Assign		Free					~
Pin Con		ement	Free Privileged-only	access				~
Pin Con	text Assign ilege acces	ement	Privileged-only		with Rising edge trigger def	tection		~ ~ ~
Pin Con Pin Privi GPIO m	text Assign ilege acces	ement s	Privileged-only	upt Mode v	0 0 00	tection		~ ~ ~

Figure 173. Setting privileges for GPIO EXTIs



<u> </u>	gg	ity and privilege of Bit		
	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	
Add Delete		Pe	eripheral	Метогу
Mode Normal	~	Increment Address		~
		Data Width Byte	~	Byte
(DMA Request Security	/Privilege			
Enable Channel as See	cured 🔽	Enable Channel as Privi	eged 🗸	
Enable Source as Sec	ured 🔽	Enable Destination as S	ecured 🗸	

Figure 174. Configuring security and privilege of DMA requests

Figure 175. 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 176*.



	DMA Mode	e and Configuration		
		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	
Add Delete				
DMA Request Settings		Pe	ripheral	Memory
		Pe Increment Address	ripheral	Memory
DMA Request Settings	s		ripheral	_
DMA Request Settings	s	Increment Address		
DMA Request Settings Mode Normal	sy/Privilege	Increment Address	· □ ·	Byte

Figure 176. 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.



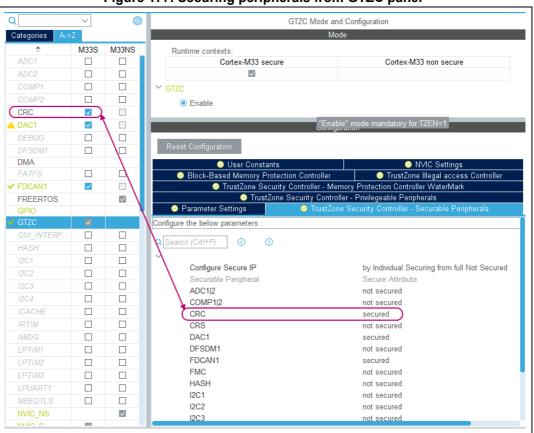


Figure 177. Securing peripherals from GTZC panel

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.

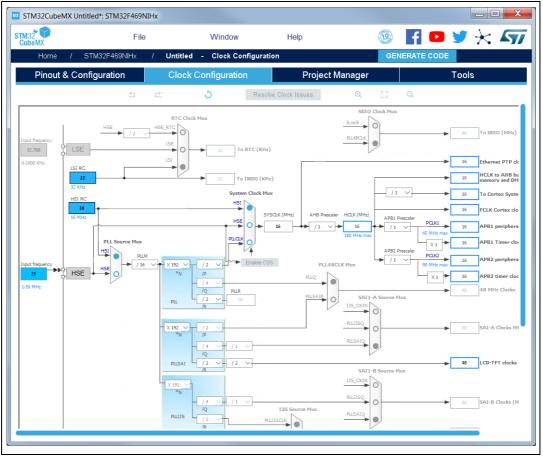
Ν	lode
Runtime contexts:	
Cortex-M33 secure	Cortex-M33 non secure
~	

Figure 178. OTFDEC secured when TrustZone[®] is active



4.10 Clock Configuration view

STM32CubeMX **Clock Configuration** window (see *Figure 179*) 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 180*) to flag potential issues. A solver feature is proposed to automatically resolve such configuration issues.

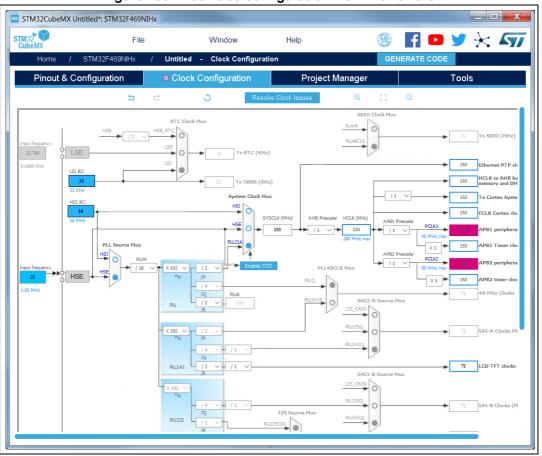


Figure 180. Clock tree configuration view with errors

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 180*).

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 179 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 16 16 MHz	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
► 40 T - XOCS (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 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 Max Mai Hei PLLM	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

UM1718 Rev 46



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).

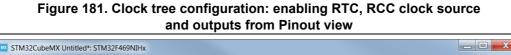
		• • • • •	
Table 10.	. Clock Configuration	n security settings	(continued)

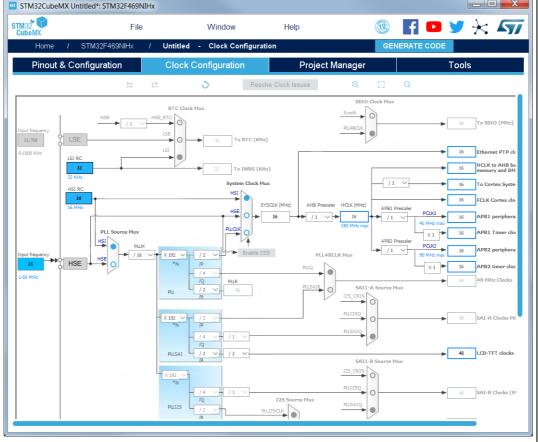


4.10.3 Recommendations

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 181*).







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.

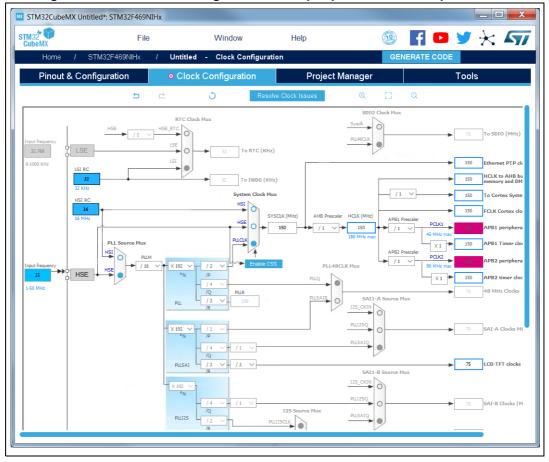


Figure 182. Clock tree configuration: RCC peripheral advanced parameters

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 46



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 183*) comes with three tabs:

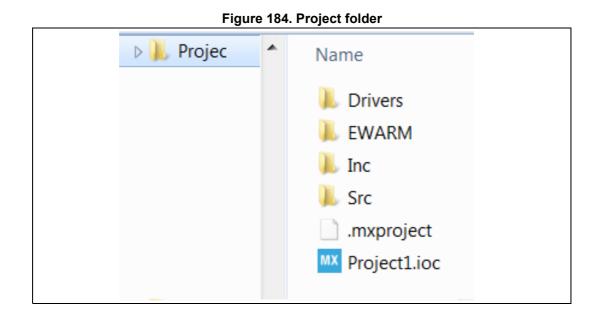
- General project setting: to specify the project name, location, toolchain, and firmware version.
- Code generation: to set code generation options such as the location of peripheral initialization code, library copy/link options, and to select templates for customized code.
- Advanced settings: dedicated to ordering STM32CubeMX initialization function calls.

Pinout & Configuration	Clock Configu	ration	Projec	t Manager		Tools	
	Project Settings Project Name						
	Project Location	C:\STM32CubeMX	Projects			В	rowse
	Application Structure	Basic				✓ □ Do not generate the	main()
	Toolchain Folder Location	C:\STM32CubeMX	Projects\				
	Toolchain / IDE	EWARM EWARM MDK-ARM	~	Min Version	/8.32	✓ □ Generate Under Roo	t
	Linker Settings	STM32CubeIDE Makefile					
	Minimum Heap Size	0x200					
	Minimum Stack Size	0x400					
	Thread-safe Settings						
	Cortex-M4NS						
	Enable multi-threaded support						
	Thread-safe Locking Strategy	Default – Mapping	suitable strategy deper	nding on RTOS sele	ction.		\sim
	Mcu and Firmware Package						
	Mcu Reference	STM32G431K6Tx					
	Firmware Package Name and Version	STM32Cube FW_G	64 V1.5.1			✓ ✓ Use latest available	version
	Use Default Firmware Location						
	Firmware Relative Path	C:/Users/bekrisli/S	TM32Cube/Repository/	STM32Cube_FW_G	4_V1.5.1	В	rowse
	L						

Figure 183. Project Settings window

The code is generated in the project folder tree shown in *Figure 184*.





- *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 183*):

- 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 185*)

Advanced structure: recommended when several middleware components are used in the project, makes the integration of middleware applications easier (see *Figure 186*)

- 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 187*). 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 188*).



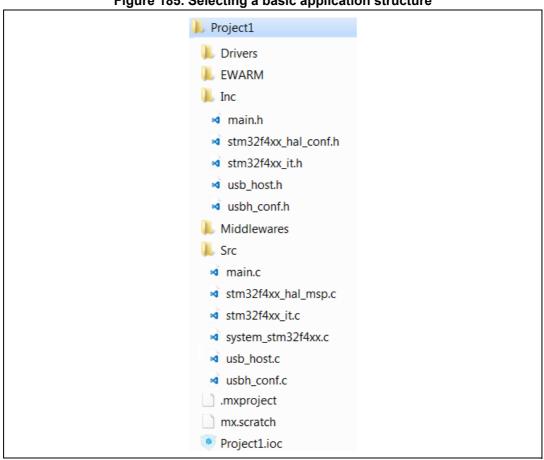


Figure 185. Selecting a basic application structure



rigate teel eeleeting an aavaneed appreaden edaet		
👢 Project3		
👢 Core		
👢 Inc		
🗃 main.h		
stm32f4xx_hal_conf.h		
🔹 stm32f4xx_it.h		
👢 Src		
🗃 main.c		
stm32f4xx_hal_msp.c		
✓ stm32f4xx_it.c		
system_stm32f4xx.c		
👢 Drivers		
👢 EWARM		
local Middlewares		
USB_HOST		
儿 Арр		
🖬 usb_host.c		
🛛 usb_host.h		
👢 Target		
🔹 usbh_conf.c		
🛛 usbh_conf.h		
.mxproject		
mx.scratch		
Project3.ioc		

Figure 186. Selecting an advanced application structure

Figure 187. OpenSTLinux settings (STM32MPUs only)

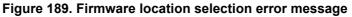
DeviceTree Root Local	on	
C:\STM32CubeMX_Pr	jects\DiscoMP1_project\DeviceTree\	
Manifest Version		
openstlinux-4.19	-thud-mp1-19-01-11	
Manifest Content:		
Firmware Name	Community Version	
TF-A	2.0	
Linux	4.19	
-	STM32Cube FW_MP1 V1.0.0	
Cube		



Figure 188. Se	electing a	different	firmware	location
----------------	------------	-----------	----------	----------

F401CDUx	
re Package Name and Version	
2 ube FW F4 V1.24.0	✓ Use latest available version
e Default Firmware Location	
Default Firmware Location	

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 189*).



Choice Firmware Library Directory	X
The selected Firmware should contain at least 'Drivers\STM32F4xx_HAL_Dri	ver" and 'Drivers\CMSIS' folders.
ОК	

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 190*).

Eile Edit View Tools Help		▼ 47 Sea	rch MyProjectsRepository 🖌
Organize - Include in library	Share with ▼	New folder	ii • 🗌 📀
CUBEMX MyProjectsRepository	Project2	*	Date modified 11/17/2016 12:16 11/17/2016 12:19 11/17/2016 2:14 PM
3 items	• •	III	

Figure 190. 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 191*):

- 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 192*).

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 193 shows the result of the template configuration shown on *Figure 192*: a sample.h file is generated according to sample_h.ftl template definition.

Projec	t Code Generator Advanced Settings	
ST	132Cube Firmware Library Package	
۲	Copy all used libraries into the project folder	
C	Copy only the necessary library files	
C	Add necessary library files as reference in the toolchain project configuration file	
Ger	nerated files	
	Generate peripheral initialization as a pair of '.c/.h' files per IP	
	Backup previously generated files when re-generating	
V	Keep User Code when re-generating	
V	Delete previously generated files when not re-generated	
	. Settings] Set all free pins as analog (to optimize the power consumption)] Enable Full Assert	
Ter	nplate Settings	
Sel	ect a template to generate customized code Settings]
	Ok Cance	

Figure 191. Project Settings code generator



Template Settings Source Folder Use default locati Location: Select your templates		pelectronics \STM32Cube \STM	32CubeMX\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	ion			
Location:	C: \Users\JohnDoe \STM:	32Cube projects\Project1\Pro	ject1	Browse
			ОК	Cancel

Figure 192. Template Settings window



i igaio ito. como	erated project template
OO- 🖟 « Project1 > Proje	ect1 >
<u>File Edit View T</u> ools <u>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 193. Generated project template

4.11.3 Advanced Settings tab

This tab comes with three panels (see *Figure 194*):

- 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).

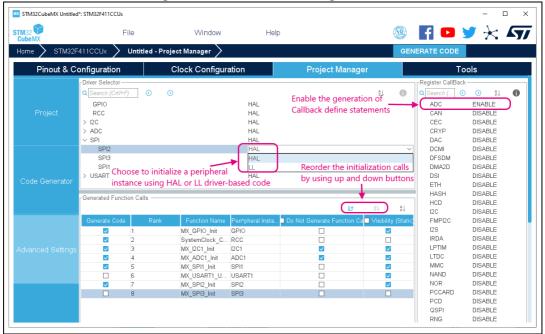


Figure 194. Advanced Settings window



Unselecting the **Visibility (Static)** option, as shown for MX_I2C1_init function in *Figure 194*, allows the generation of the function definition without the static keyword, and hence extends its visibility outside the current file (see *Figure 195*).

/* Pri	vate function prototypes
void S	<pre>ystemClock Config(void);</pre>
static	void MX GPIO Init(void);
static	void MX LPTIM1 Init (void);
static	void MX_LPTIM2_Init(void);
void M	X I2C1 Init (void);
static	void MX I2C2 Init (void);
static	void MX_SPI1_Init(void);
static	void MX_SPI2_Init(void);
static	void MX USART1 UART Init (void)
etatic	void MX_USART2_Init(void);

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 194*).

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 196*)

Figure 196. Automatic project import

Import Project
Imported Project
C:\STM32CubeMX_UM\Import IOC\OC to import\¥4_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
cPeripheral List-
From STM To STM32F722ICKx
ETH None ADC1
ADC2 I import to IADC2 V
Try Import Show View Pinout ~
Import Status
Initializing: STM32F427T(G-I)Hx
Import Analysis: C:\STM32CubeMX_UM\Import IOC\IOC to import\f4_demo.ioc project The Mcu (STM32F427IGHx) found in the Project being imported is not the same as the Mcu (STM32F722ICKx) currently edited
8 Import error: ETH peripheral doesn't exist in STM32F722ICKx
OK Cancel



b) Manual project import

In this case, checkboxes allow the user to select manually the set of peripherals (see *Figure 197*). Select the **Try Import** option to attempt importing.

Import Pr	
mported Proj	
:\STM32Cub	eMX_UM\Import IOC\IOC to import\f4_demo.ioc
mport MX Se	ttings
Import Pov	ver Consumption Calculator Settings
Import Pro	ject Settings
and the second second second	Clock Configuration/Configuration Settings-
Automatic	Import
🖲 Manual Im	port
Import	Pinning Status
0	Peripherals Configuration
Periphera	
From STN	
ETH	None
ADC1	✓ import to ADC1
ADC2	✓ import to ADC2 ✓
ADC3	✓ import to ADC3 ✓
CAN1	CAN1
NVIC	VIC NVIC
RCC	✓ RCC
SPI1	✓ import to SPI1 ✓
SPI5	✓ import to SPI5
SPI6	✓ import to SPI2
SYS	✓ SYS
	Try Import Show View Pinout ~
mport Status	3
allow or pass	ting: 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.
	error: ETH peripheral doesn't exist in STM32F722ICKx
< import	error: Ein peripheral doesn't exist in SiMS27/2210AX
	OK Cancel

Figure 197. 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 198*), 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 199* 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 199*).

	Manual Im	port						
	Import	Pinning Status						
	V 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 198. Import Project menu - Try Import with errors



	re 199. Import Project menu - Successful import after adjustments				
1	Clock Configuration/Configuration Settings				
) Automatic Import					
Manual Imp	ort				
V Import F	Pinning Status				
- ·	5				
M Import H	Peripherals Configuration				
Peripheral	List				
From STM	To STM32F722ICKx				
ETH	None				
ADC1	import to ADC1 V				
ADC2	D O import to ADC2				
ADC3	Import to ADC3				
CAN1	CAN1				
NVIC	VIC NVIC				
RCC					
SPI1	✓ import to SPI1 ✓				
SPI5	✓ import to SPI5 V				
SPI6	✓ import to SPI2 ✓				
SYS	✓ SYS				
	Try Import Show View Pinout ~				
mport Status					
	alysis: C:\SIM32CubeMX_UM\Import IOC\IOC to import\f4_demo.ioc project				
	STM32F427IGH. found in the Project being imported is not the same as the Mcu (
Importing	project 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.



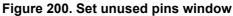
4.13 Set unused/reset used GPIOs windows

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 202* and *Figure 203* and check the limitation(s) in available peripheral modes.



Figure 201. Reset used pins window



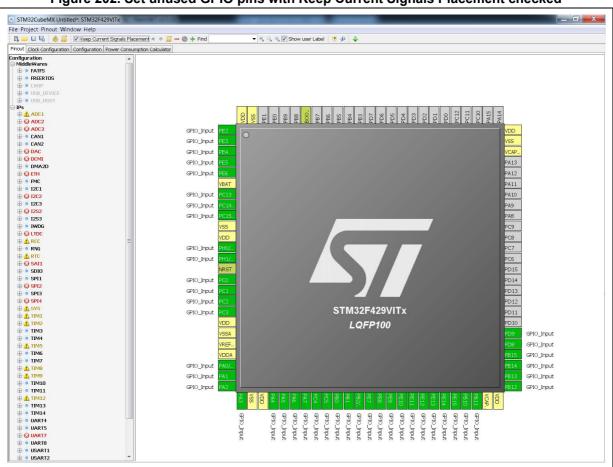


Figure 202. Set unused GPIO pins with Keep Current Signals Placement checked



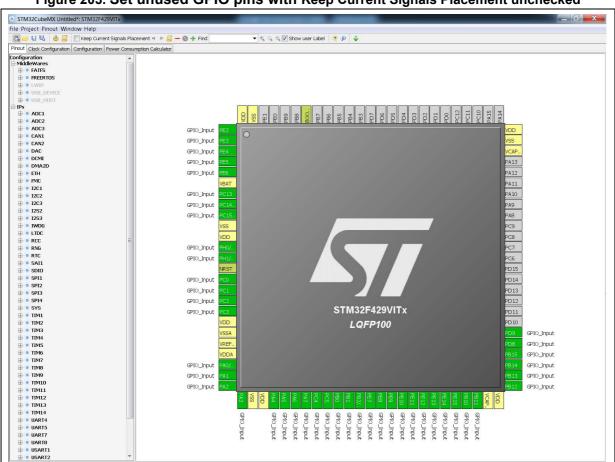


Figure 203. Set unused GPIO pins with Keep Current Signals Placement unchecked

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 204*:

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	Packs	nents for co	ntext: Cortex-M	7~	Details and warnings Component details	
arch V	Pack / Bundle / Component	Status	s Version	Selection -	Pack STMicroelectronics.X-CUBE-BLE2.3.1.0	
	STMicroelectronics.X-CUBE-BLE2	A	3.1.0 ~	Select the pack ver	rsion Bundle BlueNRG-2	
۲ <u>ــــــــــــــــــــــــــــــــــــ</u>	✓ Wireless BlueNRG-2	A	3.1.0		Class Wireless	
ik Vandar 🗸 🗸	BlueNRG-2 / Controller	A			Group BlueNRG-2	
k Vendor 🗸	BlueNRG-2 / HCI_TL	\odot		Basic V Select co	Sub-group Controller	
ARM	BlueNRG-2 / HCI_TL_INTERFACE	\odot		UserBoard 🗸	Version 3.1.0	
STMicroelectronics	BlueNRG-2 / Utils				☆ Add to favorites	
	> Device BLE2_Applications		3.1.0			
ware Component Class	> STMicroelectronics.X-CUBE-DISPLAY	0	1.0.0		Warnings (1)	
	> STMicroelectronics.X-CUBE-EEPRMA1		3.0.0		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 ~	·	Description	
Board Extension	> STMicroelectronics.X-CUBE-NFC4		2.0.1 ~	·	Description	
Board Part	> STMicroelectronics.X-CUBE-SFXS2LP1		2.0.0		BLE 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	Documents	
DSP Library	> FreeRTOS	Ð			License	
	> HAL Drivers	G⊗			Documentation: STMicroelectronics.X-CUBE-BLE2_GettingS	tar
Data Exchange	> PDM2PCM	\Box				
Device	Component dependencies					
Extension Board	Component BlueNRG-2 / Controller (from bur	dle Wireles	s BlueNRG-2)	Sho	ow 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	ow Select	
Network						
Peripheral	Re	esolve o	componer	nt dependencies		
			e (Resolv			
RF Library						
RTOS	or	ie by o	ne (Select)		
Sensors						

Figure 204. 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 46

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.



lcon	Description
⊗	The pack version is not compatible with this STM32CubeMX version. Solution: select a compatible version.
4	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.
Į)	Other pack versions are available to switch to. Solution: use the Details and Warnings panel to proceed with a change.
\Rightarrow	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 205*).

🛱 💊 🚺 > Show compor	ients for cont	ext: Cortex-M7	\sim			
Pack / Bundle / Component	Status	Version	Selection		-	
> ARM.CMSIS-FreeRTOS	٥	10.3.1				
> 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 	A	3.0.0 ~	1			
✓ Wireless BlueNRG-2	A	3.0.0				
BlueNRG-2 / Controller	4					
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	Ö	1.0.0				
Component dependencies						
 Component BlueNRG-2 / Controller (from bun 	dle Wireless	BlueNRG-2)			Show	Resolve
V Requires: component bundle BlueNRG-2,	class Wirel	ess, group Blue	NRG-2, sub Utils	, version 3.0.0		🔺 Missin
✓ Solutions in STMicroelectronics.X-CUL	BE-BLE2.3.0	. 0 :				
Component BlueNRG-2/Utils					Show	Select

Figure 205. Component dependency resolution

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 <u> Board Part EEPROM in pack STMicroelectronics.X-CUBE-EEPRMA1.3.0.0)</u> All conditions are solved.	No dependency to solve.		
Component dependencies 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 ⊘ Issue State ✓ Conflicts in STMicroelectronics X-CUBE-EEPRMA1.3.0.0;	Dependency to solve but issue encountered (no solution found or conflict).		
Component dependencies Show Reache V Winkscs BlueNRG-MS in pack STMicroelectronics X-CUBE-BLE1.6.2.0) Show Reache 	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 206*).

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.



Deteile and w	· · · · · · · · · · · · · · · · · · ·				
Details and w	0				
Pack details	S				
Name	X-CUBE-MEMS1				
Vendor	STMicroelectronics				
Version	6.2.0				
🕁 Add to f	avorites				
This version of the pack is not compatible with the current version of STM32CubeMX. It is compatible with STM32CubeMX from 5.2.0 up to 5.2.1.					
Description					
Drivers and	sample applications for MEMS components				
Documents					
License					
Documentat	ion				

Figure 206. 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 207*), click **OK** to refresh STM32CubeMX window: the selected component appears in the tree view under Additional Software (*Figure 208*).

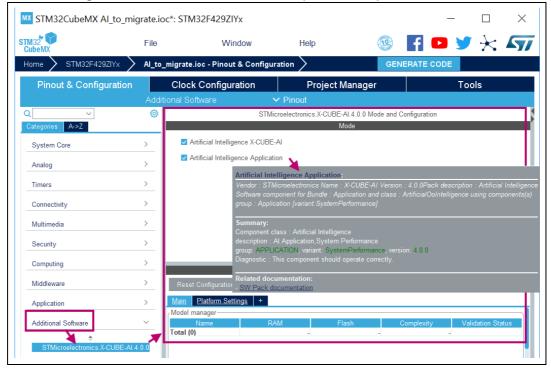
The current selection of additional software components appears in the tree view (see *Figure 208*). The software components must be enabled in the Mode panel and may be configured further if any parameter is proposed in the configuration panel. Hovering the mouse over the component name reveals contextual 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 V
✓		
⊘ Core		\checkmark
> 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 207. Selection of additional software components

Figure 208. 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 209*).

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: STM32U5754 STM32 CubeMX Home STM32U575AGIx	CLICK TO SWITCH VIEWS				
LPBAM Scenario & Configuration					
LPBAM Management ∨ ✓ IPBAM Manager → Add Application	Please create or activate a LPBAM application first				

Figure 209. LPBAM window

4.17 CAD Resources view section

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 210).
- 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 211).

To select the resources for download (see *Figure 212*)

- Select the design toolchain
- Select the CAD formats
- Accept terms and conditions
- Click to download
- Specify the download location

New Project from a MCU//MPU				
MCU/MPU F	Features Block Diagram Docs & Resources CAD Resources 📑 Datasheet 📑 Buy 🕞 Start Project			
Comme Part Nu	STM32G030F6P6 STM32G030F6P6			
۹ 🗌				
PRODUC	Please choose CAD formats Remove CAD Product CAD Family CAD Librarian			
Segmer				
Series				
Line				
Marketi				
Price	CAD L1			
Packag	Models			
Core	I have read and agree to the Ultra Librarian Terms and Conditions			
Coproce	🛱 Symbol			
MEMORY	· · · · · · · · · · · · · · · · · · ·			
Flash =				
32	MCUs/MPUs List: 2 items 🕂 Display similar items			
EEPRON	Commercial Part No Reference Marketing Status X Unit PricX Board X Package X Flash X RAM X STM32G030F6 STM32G030- STM32G030 Active 0.593 TSSOP-20 32 kBytes 8 kBytes			
0	☆ STM32G030F6 STM32G030F6			
RAM To				

Figure 210. CAD Resources view



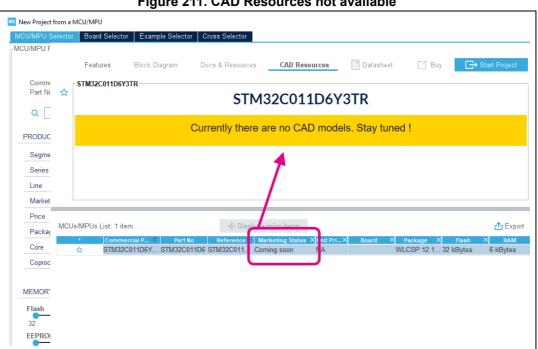
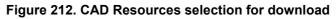
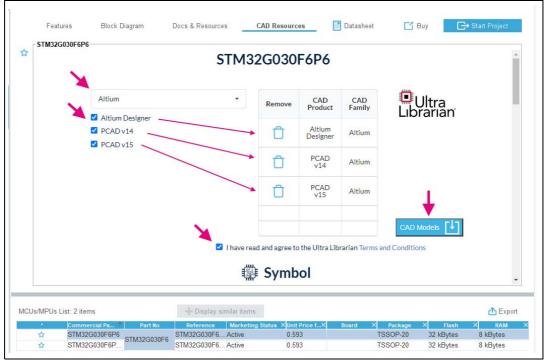


Figure 211. CAD Resources not available





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 213).



Pinout & C	Configuration	Clock Configuration		Project N	lanager	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	Librarian
CAD						CAD Models
		I have re	ead and agree	to the Ultra Li	brarian Terms a	and Conditions
			🖟 Sym	bol		
				1	1291_CK 25 987/988 1 101_8X2 2 132_00T 3	
		#4 PA1 #5 PA1 #7 Mr_GPI0_DUT #8 Mr_GPI0_TH #9 PA15/PA14-BOOTD			NRST 6 00/VDDA 55/VSSA 6	

Figure 213. CAD Resources in Tools panel

The Symbol view reflects the STM32CubeMX project pinout configuration and, optionally, the labeling (see *Figure 214*). The downloaded CAD files are aligned with the pinout configuration and optionally, with the labels as well.

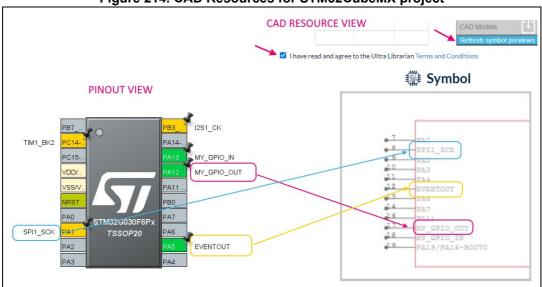
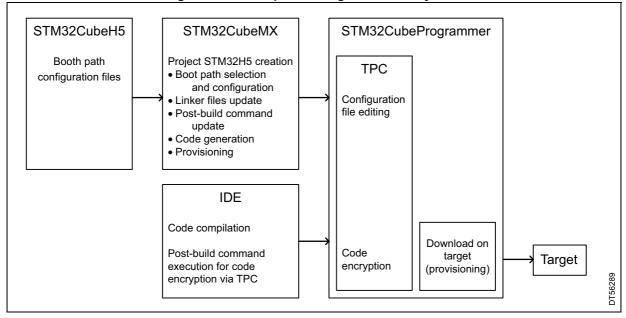


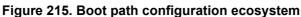
Figure 214. CAD Resources for STM32CubeMX project



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)



			bi patris with mus		,		
мси	S/NS application			ST-iRoT → S application	ST-iRoT → ST-uRoT → Secure manager → NS application	ST-iRoT → uRoT S/NS application, and ST-iRoT→ S/NS application	
STM32H56x	\checkmark	\checkmark	-	-	-	-	
STM32H57x	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	
STN32H523	\checkmark	\checkmark	-	-	-	-	
STM32H533	\checkmark	\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⁾

мси	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.

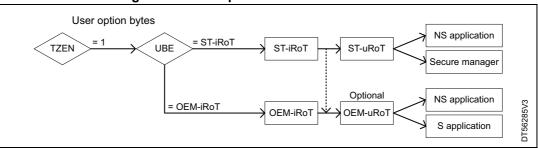
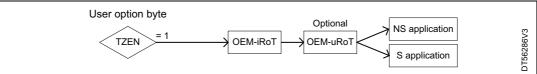


Figure 216. Boot paths for STM32H57x devices

Figure 217. Boot paths for STM32H56x devices





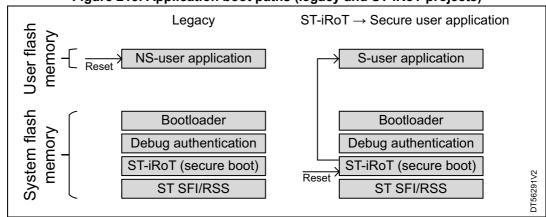
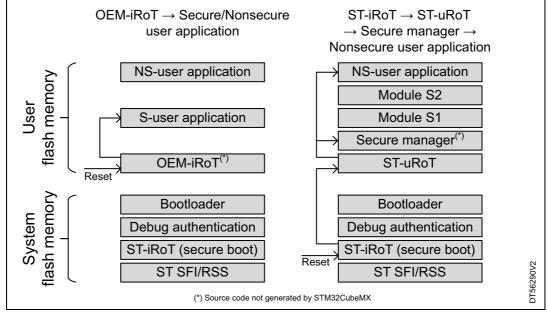


Figure 218. Application boot paths (legacy and ST-iRoT projects)







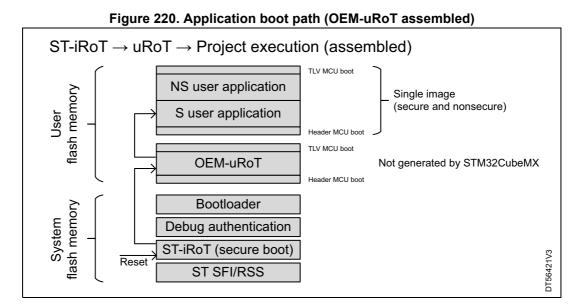
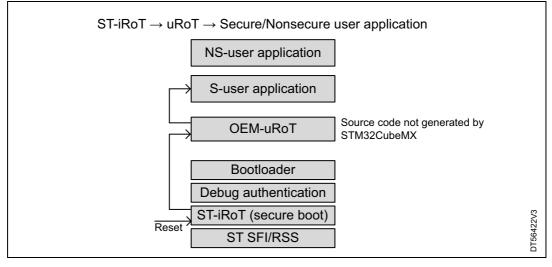


Figure 221. Application boot path: ST-iRoT and uRoT secure/nonsecure project





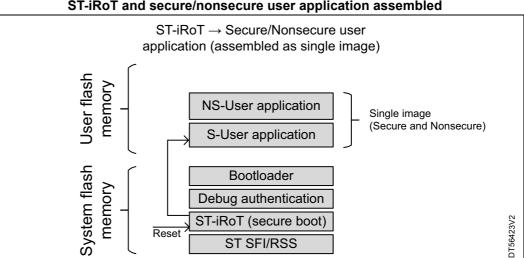


Figure 222. Application boot path: ST-iRoT and secure/nonsecure user application assembled



ST SFI/RSS

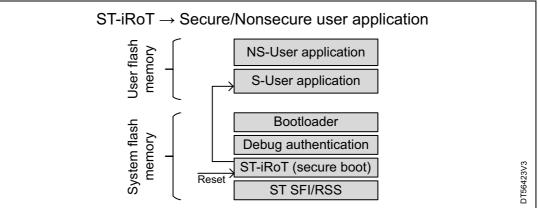
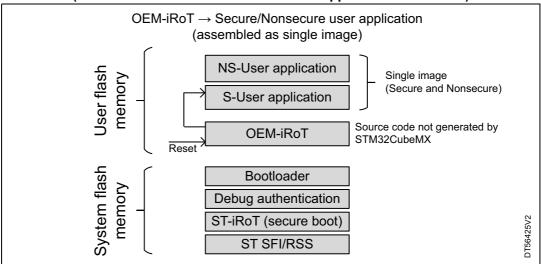


Figure 224. Application boot path: (OEM-iRoT and secure/nonsecure user application assembled)



UM1718 Rev 46



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

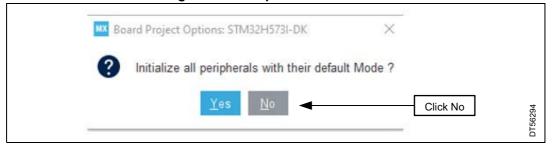
Figure 225. Select the device or board



ard Filters	pra denieror o	iross Selector						
★ 🗟 🛱 अ		Features	Large Picture	Docs & Resources	📔 Datasheet	1	lay	🕞 Start Project
Commercial Part Number	✓ 4	STM32H5 Series						
	1	STM32H573I-E	Discovery kit wi	th STM32H573IIK3Q MCU				
۹ 🔤 ۲	-	PREVIEW	Part Number : STM32H	573LDK		Unit Price (US\$): 98.75		
PRODUCT INFO	~	Product is in design stage	Commercial Part Num	ber: STM32H573HDK		Mounted Device : STM32H573IK3	2	
Type	>	140		The STM32H573I-DK Discover	y kit is a complete demon	stration and development platfo	rm for the STM32H573lik	(3Q microcontroller,
Supplier	>		light, follammin (featuring an Arm [®] Cortex [®] -M3 The full range of hardware feat				
MCU/ARVI Series Constitution AR () (match Article AR ()) (match Article AR ()) (match AR		He Lut 3 Pers		An example of the second secon	face with the STM32H5 M and STMod+ are also avail plications. y kit integrates an STLINH ual COM port bridge and a lided software components	CU peripherals and develop the lable on the board to provide ai C-V3EC embedded in-circuit det comes with the STM32CubeH5 Is required to develop an applic;	ir applications. Several con n easy way to connect ext bugger and programmer f Expansion Package, which	onnectors such as tension shields or for the STM32 h gathers in one single
STM32L5 STM32MP1 STM32U5		Ŷ	NUCLEO-H503RB	Nucleo-64	Active	15.0		5TM32H563R8T6
STM32WB STM32WBA STM32WL		Ŷ	NUCLEO-H563ZI	Nucleo-144	Active	28.75		5783285632776
	~	ŵ	STM32H573I-DK	Discovery Kit	Preview	98.75		
STM52H5 × Marketing Status	<u> </u>							

Figure 226. Select the STM32H5 device

Figure 227. 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).



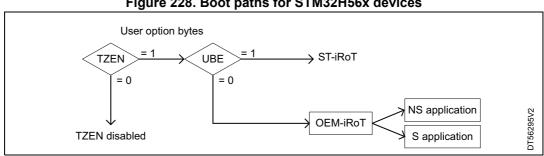


Figure 228. Boot paths for STM32H56x devices

Select the option "with TrustZone activated ?" on the popup window, as shown below.

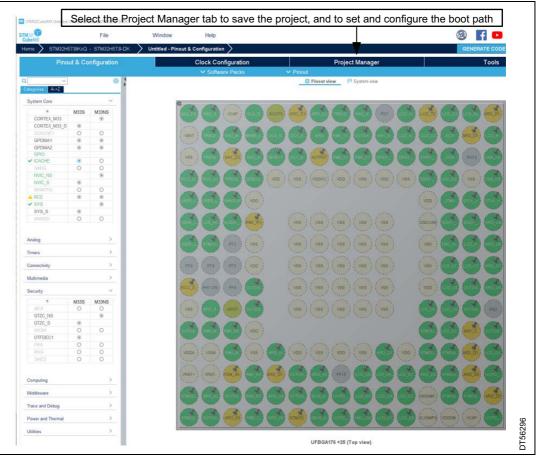
TrustZo	ne feature available	×
9	Do you want to create a new project	::
	○ without TrustZone activated ?	
	with TrustZone activated ?	
	ОК	

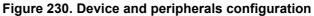
Figure 229. Activate TrustZone



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 231), then save the project (Figure 232).

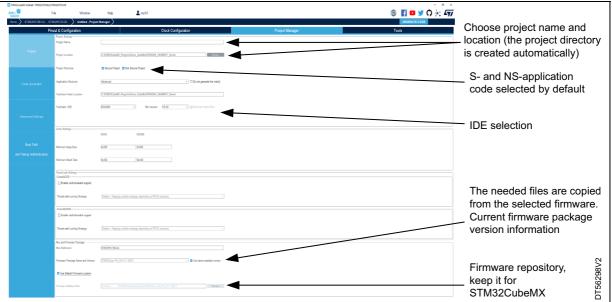
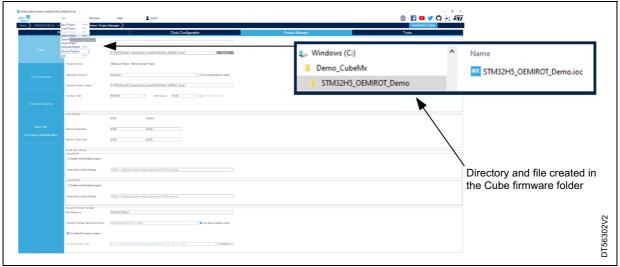


Figure 231. Configuring the project

Figure 232. Saving the project





Step 5: Boot path selection

The possible first stages are proposed according to selected device and project structure.

STM32CubeMX STM32H5_	OEMIROT_Demo.ioc: STM32H573IIKxC	2 STM32H573I-DK				- 🗆 ×
	File	Window	Help	L myST	🕸 🚹 🖻 🗶 💭 🇐	×57
Home > STM32H57	7311KxQ - STM32H573I-DK	STM32H5_OEMIROT_Dem	io.ioc - Project Manager 🔪	>	GENERATE CODE	
Pinout &	Configuration	Clock Configura	tion	Project Manager	Tools	
Project	STM32CubePro	t Path C:\Program Fi	les\STMicroelectronics\STM32C	ube\STM32CubeProgrammer\bin\STM32	Browse	
Code Gener	Configu	Generate DA Folder				
Advanced Se	Select	No boot path selected (Nothing to		ition (Default Boot Path)		
Boot Path and Debug Authe						
		Select b	poot path			

Figure 233. Boot path selection

Select OEM-iRoT for this example .

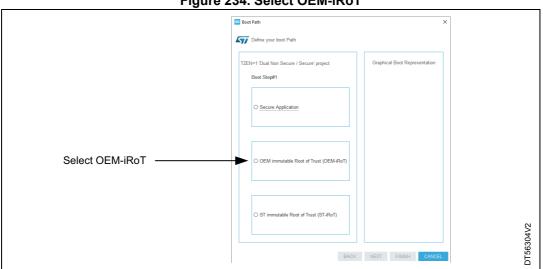


Figure 234. Select OEM-iRoT

i iguio	200. Thist boot pa		
📖 Boot Path	×		
Define your boot Path			
TZEN=1 'Dual Non Secure / Secure' project	Graphical Boot Representation		
Boot Step#1			
O Secure Application	Non Secure Secure		
OEM immutable Root of Trust (OEM-RoT)			
 ST immutable Root of Trust (ST-RoT) 	0EM-IRoT STM32	——— The first stage is shown	
BAG	CK NEXT FINISH CANCEL		D I 56305V2

Figure 235. 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.

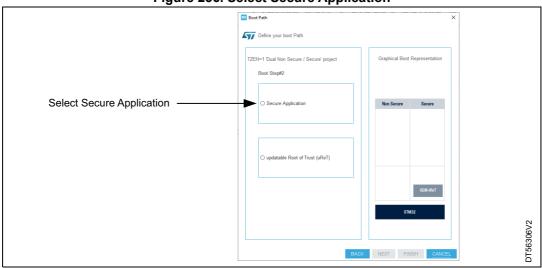


Figure 236. Select Secure Application



• Click on FINISH to generate the binary, RoT_Provisioning folder, and sub-folders.

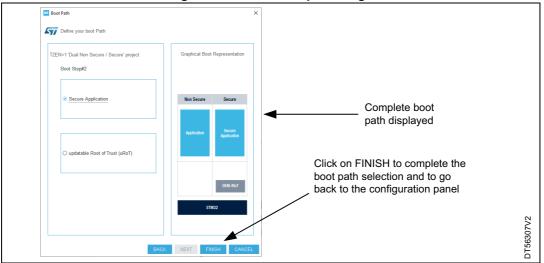
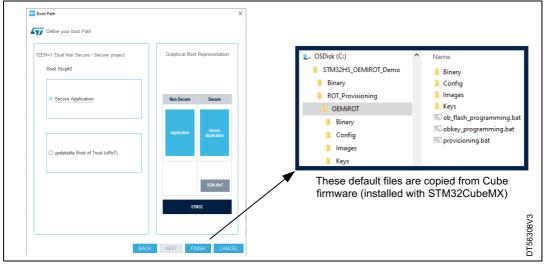


Figure 237. Last boot path stage

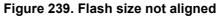




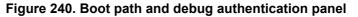
Note: If a selected boot path is not supported, a warning message is displayed, and the "FINISH" button is grayed out.

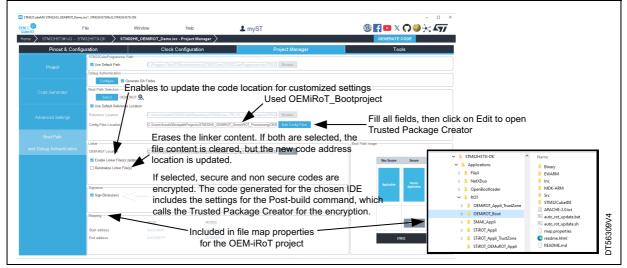
Note: For STM32H56x and STM32H523x devices, it is not possible to configure the OEM-iRoT boot path if the flash size of the current MCU is not aligned with the FLASH_SIZE entry in the map.properties file. A pop-up window (see Figure 239) is displayed.









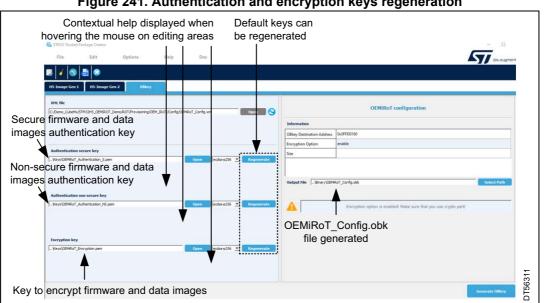


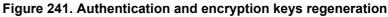
Step 6: Authentication and encryption keys regeneration, option byte file generation

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 238*)
 - 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







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 Package Creator – 🗆 🗙
File Edit Options Help Doc
HS-Image Gen 1 HS-Image Gen 2 OBkey
XHL file
[Ci;511432CabeH4, Jrsgicta]Demo, CabeH4;51143345, CB4IROT, Demo;R31143345, CB4IROT _ Demo;R31143345, CB4IROT _ Jeano;R3114345, CB4IROT _ Jeano;R3114545, CB4IROT _ Jeano;R3114555, CB4IROT _ Jeano;R3114555, CB4IROT _ Jeano;R3114555, CB4IROT _ Jeano;R31145555, CB4IROT _ Jeano;R31145555, CB4IROT _ Jeano;R31145555, CB4IROT _ Jeano;R31145555, CB4IROT _ Jeano;R311455555, CB4IROT _ Jeano;R311455555, CB4IROT _ Jeano;R311455555, CB4IROT _ Jeano;R311455555, CB4IROT _ Jeano;R311555555, CB4IROT _ Jeano;R311555555, CB4IROT _ Jeano;R311555555, CB4IROT _ Jeano;R311555555, CB4IROT _ Jeano;R3114555555555555, CB4IROT _ Jeano;R31155555555555555555555555555555555555
Firmware secure image generation
Dependency with other image
1.0.0
Version
1.0.0
Firmware binary input file
//Secure_nodbJSTM32H5_0EMIROT_Demo_S.hex
Image output file
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

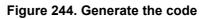
Figure 242. Secure image configuration



Sh32 hudd blacksge Creater File Exit File Exit File Exit File Exit File File Exit File File Exit File File Exit File File Exit File </th <th></th> <th></th> <th></th> <th></th> <th></th> <th></th>						
Image Construction It's image Construction Status Status Construction	🚱 STM32 Trusted	Package Creator				
XH file Cr_STM33CLdeRK_Projects/Demo_CLdeRKOT/Demo/STM32HS_CERIRGOT_Demo/ROT_Provisioning/DEMROT/Integer/DEMROT_INS_Code_Image.xml Open Firmware non secure image generation Provisioning DEMROT_Demo/STM32HS_CERIRGOT_Demo/ROT_Provisioning/DEMROT_INS_COde_Image.xml Image: Code Image State Loc Version Loc Version Image CodeROT_Demo JEG.Exe(STM32HS_CERIROT_Demo JEG.Exe) Junge CodeROT_Demo JEG.Exe(STM32HS_CERIROT_Demo JEG.Exe) Junge CodeROT_Demo JEG.Exe(STM32HS_CERIROT_Demo JEG.Exe) Version Loc Version Junge CodeROT_Demo JEG.Exe(STM32HS_CERIROT_Demo JEG.Exe) Junge CodeROT_Demo JEG.Exe(STM32HS_CERIROT_Demo JEG.Exe) Tool Description: Tool Description: Tool Description: Tool Description: The imption is on XMR, file description ig the commands and parameters to be passed to the imption. The imption is an XMR, file description ig the commands and parameters to be passed to the imption. The imption is an XMR file description ig the addite in the STM22CuberForgrammer package.	File	Edit	Options	Help	Doc	Life.ougment
XH file Cr_STM33CLdeRK_Projects/Demo_CLdeRKOT/Demo/STM32HS_CERIRGOT_Demo/ROT_Provisioning/DEMROT/Integer/DEMROT_INS_Code_Image.xml Open Firmware non secure image generation Provisioning DEMROT_Demo/STM32HS_CERIRGOT_Demo/ROT_Provisioning/DEMROT_INS_COde_Image.xml Image: Code Image State Loc Version Loc Version Image CodeROT_Demo JEG.Exe(STM32HS_CERIROT_Demo JEG.Exe) Junge CodeROT_Demo JEG.Exe(STM32HS_CERIROT_Demo JEG.Exe) Junge CodeROT_Demo JEG.Exe(STM32HS_CERIROT_Demo JEG.Exe) Version Loc Version Junge CodeROT_Demo JEG.Exe(STM32HS_CERIROT_Demo JEG.Exe) Junge CodeROT_Demo JEG.Exe(STM32HS_CERIROT_Demo JEG.Exe) Tool Description: Tool Description: Tool Description: Tool Description: The imption is on XMR, file description ig the commands and parameters to be passed to the imption. The imption is an XMR, file description ig the commands and parameters to be passed to the imption. The imption is an XMR file description ig the addite in the STM22CuberForgrammer package.	3 🤞 😵	5			l i	
CC3FN1322.64eM, Projects/Deno_GAdeMySTN13245_GBHROT_Deno/STN13245_GBHROT_Deno/RCT_Provisioning/GBHROT_ISS_Code_Image.and Open Firmware non secure image generation Dependency with other image 10.0 Firmware non secure image generation Teno Te	H5-Image Gen 1	H5-Image Ge	n 2 OBkey			
Firmware non secure image generation Dependency with other image Image colspan="2">Image colspan="2" Image colspan="2"	XML file	-				
Dependency with other image 1:0.0 Version 1:0.0 Firmware binary input file /FURMAM hindnessere/EDM2046_ODMRAT_Demo_JIS_INE_STM3345_OBMRAT_Demo_JIS_INE_STM3345_OBMRAT_Demo_JIS_ENE_STM3345_OBMRAT_DEmo_JIS_ENE_STM3345_OBMRAT_DEmo_JIS_ENE_STM3345_OBMRAT_DEmo_JIS_ENE_STM3345_OBMRAT_DEmo_JIS_ENE_STM3345_OBMRAT_DEmo_JIS_ENE_STM3345_OBMRAT_DEmo_JIS_ENE_STM3345_OBMRAT_DEmo_JIS_ENE_STM3345_OBMRAT_DEmo_JIS_ENE_STM3345_OBMRAT_DEmo_JIS_ENE_STM3345_OBMRAT_DEmo_JIS_ENE_STM3345_OBMRAT_DEMO_JIS_ENE_STM3345_OBMRAT_DEMO_JIS_ENE_STM3455_	C:/STM32CubeMX_	Projects/Demo_Cubel	Mx/STM32H5_OEMIROT_E	Demo/STM32H5_OEMIR	DT_Demo/ROT_Provisio	ioning/DEMROT/Images/DEMROT_NS_Code_Image.xml
Dependency with other image 1:0.0 Version 1:0.0 Firmware binary input file /FURMAM hindnessere/EDM2046_ODMRAT_Demo_JIS_INE_STM3345_OBMRAT_Demo_JIS_INE_STM3345_OBMRAT_Demo_JIS_ENE_STM3345_OBMRAT_DEmo_JIS_ENE_STM3345_OBMRAT_DEmo_JIS_ENE_STM3345_OBMRAT_DEmo_JIS_ENE_STM3345_OBMRAT_DEmo_JIS_ENE_STM3345_OBMRAT_DEmo_JIS_ENE_STM3345_OBMRAT_DEmo_JIS_ENE_STM3345_OBMRAT_DEmo_JIS_ENE_STM3345_OBMRAT_DEmo_JIS_ENE_STM3345_OBMRAT_DEmo_JIS_ENE_STM3345_OBMRAT_DEmo_JIS_ENE_STM3345_OBMRAT_DEMO_JIS_ENE_STM3345_OBMRAT_DEMO_JIS_ENE_STM3455_						
L0.0 Version L0.0 Finance binary input file //.EWARM/indicare_ESTM32Hs_CBMROT_Demo_HS_hex Open Image output file /						Firmware non secure image generation
Version 1.0.0 Firmware binary tiput file /RUMAN Montecere/ETM22H5_OBMROT_Demo_JIS_Res/ETM32H5_OBMROT_Demo_JIS_Res Image output file //		h other image				
10.0 Finance binary liput file	,					
Firmware binary input file Open ///RIVAYMonSecure/ETM3245_0EMBROT_Demo_JBS.tex Open Image output file Image output file ///RIVAYMS15_0EMBROT_Demo_JBS_enc_sign.hex Select Path Tool Description: The toin path is to a generate signed and encrypted images using the imgtool. The toin path is an XML file description generates to be passed to the imgtool. The imgtool is embedded in the STM32CubeProgrammer package.						
///.RWARMAndscore_RSTM33H5_054RK0T_Demo_16_ktws Open Image output file //.BineryISTM33H5_054RK0T_Demo_16_arc_sign.hex Select Path Tool Description: The purpose of this tool is to generate signed and encrypted images using the imgtool. The tool input is an XM4. file describing the commands and parameters to be passed to the imgtool. The tool is enbedded in the STM32CubeProgrammer pedage.	,	input file				
///BneryISTM32H5_0DeRROT_Demo_16_erc_sign.hex Select Path Tool Description: Image: Comparison 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.			EMIROT_Demo_NS/Exe/S	STM32H5_OEMIROT_Der	no_NS.hex	Open
Tool Description: The purpose of this tool is to generate signed and encrypted images using the imgtool. The tooi input is an XMA. file describing the commands and parameters to be passed to the imgtool. The imgtool is embedded in the STM32CubeProgrammer package.	Image output fi	le				
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 is embedded in the STM32CubeProgrammer package.	//Binary/STM:	2H5_OEMIROT_Dem	io_NS_enc_sign.hex			Select Path
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.	Tool Description					
Save Configuration	The tool input is	an XML file descr	ribing the commands a	and parameters to b	he imgtool. e passed to the img	igtool.
Save Configuration						
Save Configuration						
						Save Configuration

Figure 243. Nonsecure image configuration

Step 7: Code generation



32 Topological States and States	File	Window	Help	L myST	3	F 🗖 🗴 🗘 🍏 🔆 🖅
me 🔪 STM32H573IIKxQ - S	sтмз2н573нDк > sтм	132H5_OEMIROT_D	emo.ioc - Project Manager 🔪	`		GENERATE CODE
Pinout & Cont	<u> </u>		lock Configuration	Project Ma	anager	Tools
	STM32CubeProgrammer F					T
	Use Default Path	COProgram		ube\STM32CubeProgrammer\bin\STM32 Browse		
	Debug Authentication	enerate DA Folder			•	
	Boot Path Selection	enerate DA Folder			Click here to ge	nerate the code
		IROT 🔍			and the I	DE environment
	Use Default Reference	Location				
	Reference Location	C:\Users\k		2Cube_FW_H5_V1.2.0\Projects\STM32 Browse		
	Config Files Location	C:\Users\k	outb\BootpathProjects\STM32H5_0	DEMIROT_Demo\ROT_Provisioning\OEN Edit Con	nig Files	
	Linker				Boot Path Imag	9ê
	OEM-IRoT Location	C:\Users\k	outb\STM32Cube\Repository\STM	2Cube_FW_H5_V1.2.0\Projects\STM32 Browse		
	Z Enable Linker File(s) L	Jpdate				Non Secure Secure
	Reinitialize Linker File	(s)				
						6 mm
	Signature					Application Secure Application
	Sign Binary(ies)					
	Mapping					
		M33S	M33NS			0EM-IRoT
	Start address					
	End address					STM32



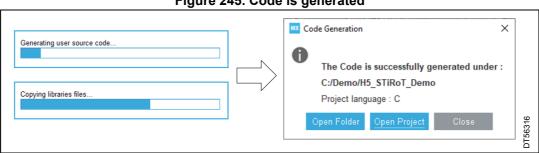


Figure 245. Code is generated

Additional directories, including the IDE environment, are created.

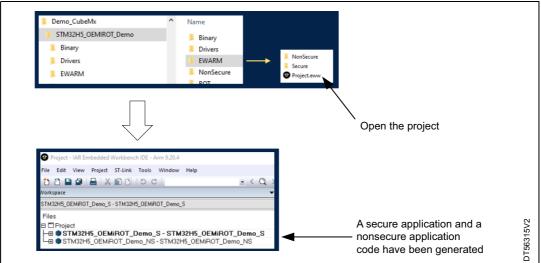


Figure 246. Secure and nonsecure IDE directories

The S and NS applications can be developed using the generated code skeletons.

Step 8: Code compilation and encrypted binaries generation

If the "Sign Binary(ies)" option has been ticked at Step 7: Code generation, the application binaries are encrypted. 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.



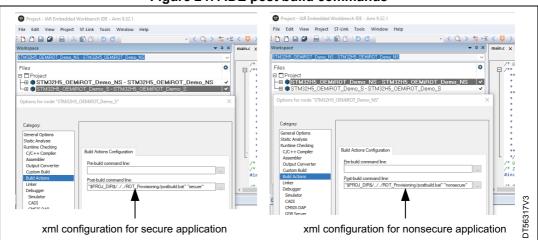
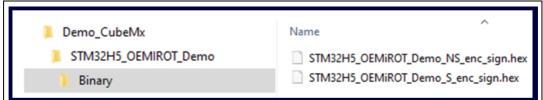


Figure 247. IDE post build commands

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 248. 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 249*). During provisioning, log files are generated to inform the user about the activity. Follow the on-screen instructions (*Figure 250*).



📙 🗹 📜 🗧 🛛 C:\ST_MMS\CodeGeneration	\Branch_6_8_RC4\OEMiRoT_Dual_	LED\ROT_Provisioning\OE	EMIROT		-
File Home Share View					
Pin to Quick Copy Paste access	Move Copy to* to*	New item •	Properties History	'n	
Clipboard	Organize	New	Open Select		
← → ✓ ↑ 🖡 - Branch_6_8_RC4 > O	EMiRoT_Dual_LED > ROT_Provisi	oning > OEMiROT	・ じ / Search OEM	IROT	
OEMiRoT		^	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			1		
📜 Keys					
OEMIROT					
Secure					
Secure_nsclib					

Figure 249. Board provisioning



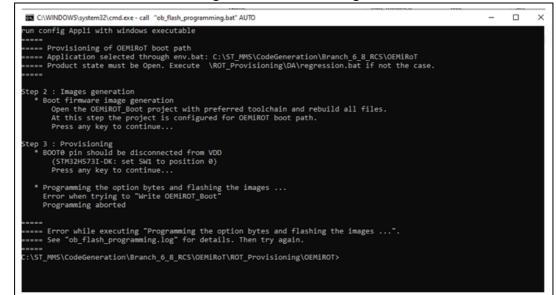


In the user environment, STM32CubeMX has generated an env.bat file, containing the information required for provisioning. Do not change this file.



A pop-up (see *Figure 251*) appears if you forget to compile the project OEMiRoT_Boot in the CubeFW.

Figur	e 251.	Error	messa	ge
-------	--------	-------	-------	----



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 igure 202. Gereet	
M3 Boot Path	x
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 immutable Root of Trust (ST-iRoT)	ST-IRoT STM32
BACK	NEXT FINISH CANCEL

Figure 252. Select ST-iRoT

Select Secure Application

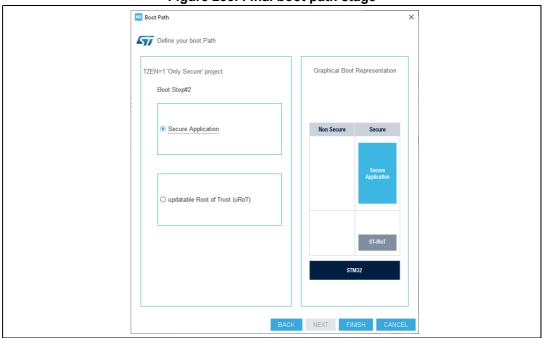


Figure 253. Final boot path stage

• Click "FINISH", the boot path configuration panel is displayed (see *Figure 254*), use it to configure the application, then press the GENERATE CODE button to generate the code for the selected toolchain



I2CubeMX ST-IROT.ioc*: STM32H57								
X	File	Window	Help	💄 myST		(🧐 f 🕒 🎗	< 🗘 🍏 🔆 🕰
STM32H573IKkQ	ST-IROT.ioc - Project Ma	inager 🔪					GENERAT	ECODE
Pinout & Con	figuration	Clo	ck Configuration		Project Manager		Т	ools
	STM32CubeProgrammer Pa	th-						
	Use Default Path	C:\Program Fil			TM32 Browse			
	Debug Authentication							
	Configure 🗹 Ger	nerate DA Folder						
Code Generator	Boot Path Selection							
	Select ST_IRC	от 🔍						
	Use Default Reference L	Location						
Advanced Settings	Reference Location	C:\Users\krout			TM32 Browse			
	Config Files Location	C:\Users\krout	tb\BootpathProjects\ST-iROT\RO	T Provisioning\STIROT	Edit Config Files			
	Linker					Boot Path Image		
	Reinitialize Linker File(s) Op							
	C Reinibalize Linker Pile(s	9					Non Secure	Secure
	Signature							
	Sign Binary(ies)							Secure Application
	Mapping							
		M33S					l l	ST-IRoT
	Start address							51-IN01
	End address						STM32	

Figure 254. Boot path and Debug Authentication tab

Figure 255. Select the project structure

	S2CubeMII ST-illo Tioc*: STMS2HS73H	ъq	Generate the	e code and the IDE environment 🧹 .
Provide Circle Configuration Project Manager Tools	2 ()	File Window	Help 💄 myST	📃 🖬 🗖 💆 🖓 🖓 🖓
Appendix maintaine	н 🔪 57М3245731Кою 🔪	ST-iRoT,isc - Project Manager 🔪		
Part Image: Image	Pi		Clock Configuration	Project Manager Tools
Perform Texture Perform <td></td> <td></td> <td>8T-ReT</td> <td></td>			8T-ReT	
<pre></pre>		Project Location	E (STR20CubeRC Projects/Dens CubeRASTR2015 OF/MROT Dans	
Add wordsow Add wordsow Add wordsow		- appendix and		
Current of the second of the seco		Project Structure	🖬 Secure Project 🗌 Non Secure Project	
Averal Market Normania and Averal Market Normani		Application Structure	Advanced 💛 🗆 Do not generate the main()	
Abarda data data data data data data data		Toolchain Folder Location	C ISTM32OubeMX_Projects/Demo_OubeMxISTM32H5_OEMIROT_DemoIST+RoTi	
No Full Not No Full Not No Full Not No Full Not Not Not Not Not Not Not Not Not Not Not Not Not Not Not Not Not Not Not Not Not Not Not Not Not Not Not Not Not		Tealchain / IDE	EWARM V Min Version V9.20 V 🖂 Generate Under Root	
No Pade Note and Note Name Del				
In Production IDM Normalized Production		Linker Battings		
None Poly None Poly		Canal Security	M338 M33N6	
Nome the basis Note:		Minimum Heap Size	0-200	
Nariak Inny- Collection Instruction from generation and the structure Instructure Collection Instructure Collection				
Control Desk refer konneger Desk ref konneger Desk ref konneger </td <td></td> <td>Minimum Stack Size</td> <td>0x400</td> <td></td>		Minimum Stack Size	0x400	
I Dada Huda Huandi Indian Jahanga Huandi Huangi Jahangi		Thread-safe Settings		
Chattrade				
Clockin McKnadi Laget Nazi da Laget partier Programment Size af Transport		Thread-safe Locking Strategy	Default – Mapping suitable strategy depending on RTOS selection.	
		ColorMWWR		
la sel franza holagi Ha defansa Taman Padaga Hana at Vana Franza Padaga Hana at Vana B Ua defat Franza Locate				
la col frances houses la col frances houses la col frances houses there and there frances houses there and there la col frances incoler				
So deferrer		Thread-safe Locking Disategy	Default – Mapping suitable stratingy depending on IRTUS selection.	
Freman Package Name and Yearson (2010) (2011				
2 Use Orbital Frement Location		MCU reterence	Dimpanoranon	
		Firmware Package Name and Version	STM32Cube FW_HS V1.1.0RC1 🗠 🗹 Use latest available version	
Person Rodon Pala Excess - 2010/2014 Personal VERSEAR Per of VERse		Use Default Firmware Location		
		Firmute Relative Path	E/Users/ /STM32Cube/Repentory/STM32Cube_PW_H6_VL1.0RC1 Brokse	

For this boot path, only the secure project is generated.





Generating user source code		Copying libraries files	
	de Generation The Code is successfully genera C:/ST_MMS/CodeGeneration/Bra Project language : C Open Folder Open Project	nch_6_8_Final_RC/STiRoT	

Additional directories, including the IDE environment are created.

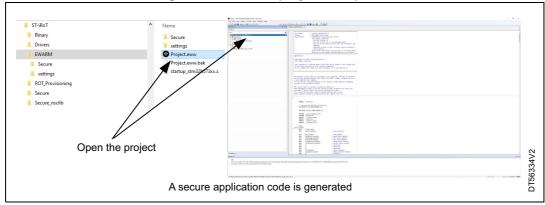


Figure 257. Secure project completed

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 and encrypted binaries generation

If the "Sign Binary(ies)" option is ticked during boot path and debug authentication configuration, the generated application binaries are encrypted.

- Open the project in the selected toolchain, then, for 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)
 - After the Post build command the secure signed and encrypted binaries are generated



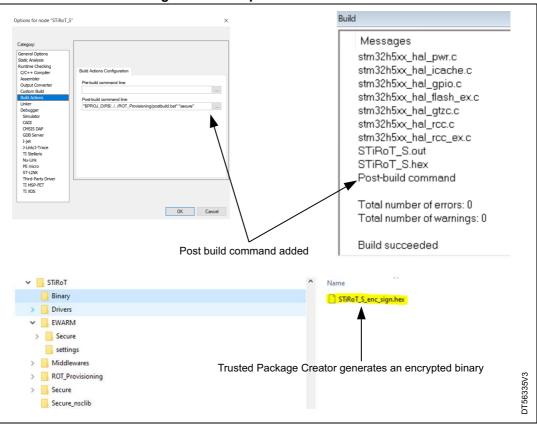


Figure 258. IDE post build commands

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 259*)

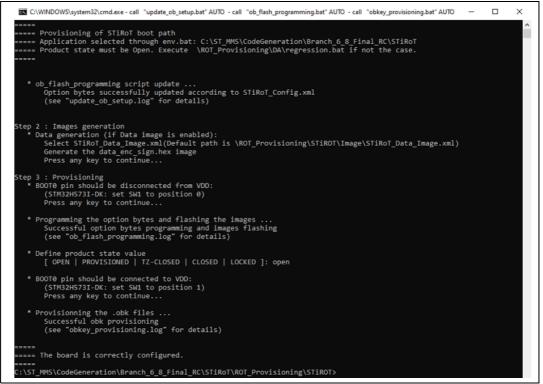


		5
📜 ST-iRoT	^	Name
📜 Binary		📕 Binary
📜 Drivers		📙 Config
EWARM		📕 Image
Secure		📜 Keys
settings		ob_flash_programming.bat
ROT_Provisioning		obkey_provisioning.bat
DA		provisioning.bat
STIROT		🕾 update_appli_setup.bat 🕾 update_ob_setup.bat
Binary		update_ob_setup.bat
📜 Config		
📜 Image		
📜 Keys		
📜 Secure	1.0	
📜 Secure_nsclib		

Figure 259. Board provisioning

- During provisioning, log files are generated to inform the user about the activity
- Follow the on-screen instructions (Figure 260)





In the user environment STM32CubeMX has generated an env.bat file containing the required data for provisioning, do not change it.



	Figure 201. Environment Co	onngurati	on me
🗸 📙 STiRoT		^	Name
- Binary			DA
> Drivers			STIROT
V 📙 EWARM			lenv.bat
> Secure			
settings			
> Middlewares			
✓ ROT_Provisioning			
> DA			
STIROT			
Binary			
Config			
Image			
Keys			
> Secure			
Secure_nsclib			

Figure 261. Environment configuration file

4.18.5 How to configure an ST-iRoT with a secure manager NS application boot path

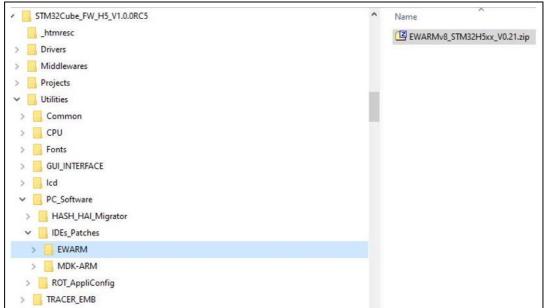
The boot path configuration described below is the ST-iRoT \rightarrow Secure manager, also known as SM (secure manager).

Prerequisites:

- Hardware: Discovery board STM32H573I-DK-REVC or later
- Required tools
 - Secure manager package, to be downloaded and installed from www.st.com
 - STM32CubeMX-6.8.0 or later
 - STM32 Trusted Package Creator (embedded in STM32CubeMX installation folder)



Figure 262. EWARM patch installation



Step 1: SM code generation

- Select an STM32H57x MCU
- Create a project with TrustZone[®] activated (TZEN = 1)
- In Project Manager, choose "Non-secure Project"
- Save the project
- Go to "Boot Path and Debug Authentication" tab and press the Select button
- Only ST immutable Root of Trust (ST-iRoT) is proposed



Boot Path	×
Define your boot Path	
TZEN=1 'Only Non Secure' project Boot Step#1	Graphical Boot Representation
Secure Application	Non Secure Secure
OEM immutable Root of Trust (OEM-IRoT)	
ST immutable Root of Trust (ST-iRoT)	ST-IROT STM32

Figure 263. First boot path stage

• Updatable Root of Trust (uRoT) option is set by default and cannot be modified

Figure 264. Second boot path stage

TZEN=1 'Only Non Secure' project Boot Step#2	Graphical Boot Representation
Secure Application	Non Secure Secure OEM-uRoT ST-IRoT STM32

• Secure manager nonsecure application button is checked by default and cannot be modified



Boot Path	×
Define your boot Path	
TZEN=1 'Only Non Secure' project Boot Step#3	Graphical Boot Representation
Secure Application	Non Secure Secure
 Secure Manager Non Secure Application 	Application Secure Manager
	ST-uRoT ST-iRoT
	STM32
BACK	K NEXT FINISH CANCEL

Figure 265. Final boot path stage

• Click "FINISH": the panel of boot path configuration is displayed (*Figure 266*), use it to configure the boot path in the "Boot Path and Debug Authentication" tab

Figure 266. Boot path and Debug Authentication tab

STM32CubeMX SMAK.ioc: STM32H573A	lbQ							- 0
132 T	File	Window	Help	L myST		(🕸 🗗 🕒	X 🗘 🍏 🔆 🕰
me 🔪 STM32H573AllxQ 🔪	SMAK.ioc - Project Manag	er 🔪					GENER	ATE CODE
Pinout & Cont	figuration	С	lock Configuration	Pr	oject Manager			Tools
	STM32CubeProgrammer Patl							
	Use Default Path	C:\Program			Browse			
	Debug Authentication							
		erate DA Folder						
	Boot Path Selection							
		LUROT_SECURE_M	ANAGER 👷					
	Use Default Reference Lo							
	Reference Location							
	Config Files Location	C:\Users\kr	outb\BootpathProjects\SMAK\ROT	_Provisioning\SM	Edit Config Files			
	Linker					Boot Path Image		
	Enable Linker File(s) Upd	late						
	Reinitialize Linker File(s)						Non Secure	Secure
	Signature							
	Sign Binary(ies)						Application	Secure Manager
	Mapping							ST-uRoT
	A		M33NS					ST-iRoT
	Start address End address							
	End address						STI	432

UM1718 Rev 46



• Press the "GENERATE CODE" button to generate the configuration code for the selected toolchain

			Generate the code	and the IDE envi	ronment
//32CubeMX SMAK.ioc*: STM32F	H573AllxQ				- 🗆 X
2 T	File Wind	tow Help	L myST	® F 🖻	y 🗘 🔀 🗖
ie 🔪 STM32H573AllxQ	SMAK.ioc - Project Manag	er >		GENERATE O	ODE
Pinout & Conf	iguration	Clock Configuration	Project Manager		Tools
	Project Settings Project Name Project Location	SMAK C.\STM32CubeMX_Pro	jects\Demo_CubeMx\STM32H5_OEMIROT_Demo	Browse]
	Project Structure Application Structure Toolchain Folder Location	Secure Project Advanced C:\STM32CubeMX_Pro	Non Secure Project	Do not generate the main()]
	Toolchain / IDE / Linker Settings	EWARM	V9.20	Generate Under Root	
	Minimum Heap Size	M33S 0×200	M33NS 0×200		
	ion Minimum Stack Size	0×400	0x400		
	CortexM33S		table strategy depending on RTOS selection.		
	Enable multi-threaded su Thread-safe Locking Strateg		table strategy depending on RTOS selection.		
	Mcu and Firmware Package- Mcu Reference	STM32H573AllxQ]
	Firmware Package Name and		1.1.0RC1	✓ Use latest available version	
	Firmware Relative Path	C:/Users/ /STM3	2Cube/Repository/STM32Cube FW H5 V1.1.0RC1	Browse	

Figure 267. Select the project structure

Figure 268. Code is generated

Generating user source code		Copying libraries files	
™ co 1	te Generation The Code is successfully gener C:/ST_MMS/CodeGeneration/B Project language : C Open Folder Open Project	Branch_6_8_Final_RC/SMAK	

Additional directories including the IDE environment are created.



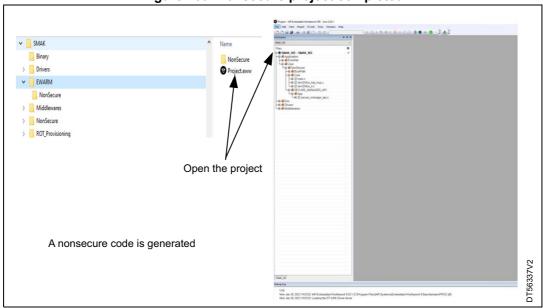


Figure 269. Nonsecure project completed

The nonsecure application can be developed using the generated code skeletons.

SM code compilation and encrypted binaries generation

If the "Sign Binary(ies)" option is ticked during boot path and debug authentication configuration, the generated application binaries are encrypted.

- Open the project in IAR
- Select: Project \rightarrow Option \rightarrow Build Actions
- The link to the STM32 Trusted Package executable and the link to the secure application xml are filled automatically
- Compile secure (right click on Project → Rebuild all)
- After the post build phase, the secure signed and encrypted binaries are generated



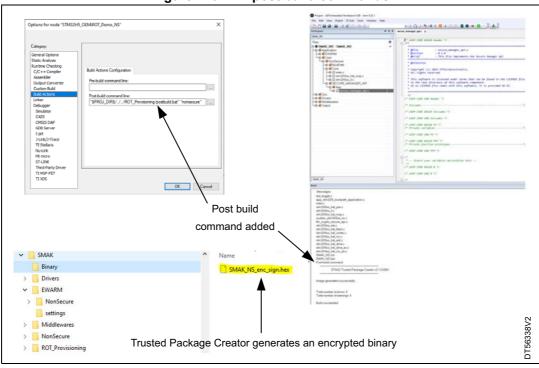


Figure 270. IDE post build commands

Secure manager API

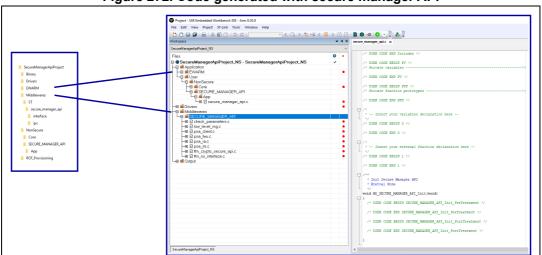
When SM boot path is set, the middleware "Secure Manager API" can be configured (see *Figure 270*).

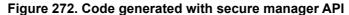
2 MX		File	Window	Help				6	👂 🖪 💶 🄰 🖓 🔆
е 🔪 STM32H573A	dx >	SecureMan	agerApiProject.ioc - Pinout & Conf	iguration >					GENERATE CODE
Pinc	out & C	onfiguration	n	Clock Co	nfiguration		Project Manager		Tools
				~ :	Software Packs	✓ Pinout			
			SEC	JRE_MANAGER_API M		A.		Pinout view	ian .
orias A->Z			320	Mode	oue and configuration		_	P mout view	15W
			Defense to the	111000					
			Runtime contexts: Cortex-M33 secu		Cortex-M33 non secure				
unty		,	Contraction and				Participante anticipation anticip	and a series and a series of	The second second second second
puting		· · ·	Y Secure Manager services					P85 (984943.) (VD0102) (P010) (P	00 (100) (POIT) (PAI40.) (PO
denced.									
deware and Software P	acks	~	Cryptography				PC140.) (PED) (VSS) (VCAP)	80070) (PGIS) (VSS) (PD7) (P	CT2 (VS3 (PAIS(L) (PTI) (PD)
			Initial Attestation						
CMSIS-FreeRTOS	M33S	M33NS	Internal trusted storage				POIS 0.3 (PES) (PIS) (PIA) (PED (PES) PD14) (PD12) P	D2 PCIO PIS VSS VCO
CMSIS-FreeRIOS FILEX		۲	Firmware update				0000		
I-CUBE-CANOPEN								PEL (PET) (PEL) (PES) (P	
			Secure Manager profile Large			~	Con Cas Cas Cas Cas C		no (men) (men) (men) Sonnak
-CUBE-wolfSSL							FFT VEAT (FIS) POTS)	P88 (F696/F.) (P011) (P08) (P	DI PANO FAO PAO PANSUL PANZ
							while while while while		the state state state
LEVELX		۲					(era) (era) (era) (era) (PT3 (PTS) (PD3) (PD4) (P	CO FOT POD PAG PAG
NETXDUO		۲							
SECURE_MANAGE		•					(VOD) (VSS) (PED) (PED) (953 (2513) PER (252) 2	ED POS PER POS POS
THREADX		۲							
	0	0		Configurati			PHO.O.) PHILO.) PPA		
		۲		Conigurati			Superior Superior Superior S		DIS (POIL) (POIA) (VSS) (VCO)
X-CUBE-AI X-CUBE-ALGOBUILD			🔥 Warning:	his peripheral has no	parameters to be configured.				
CUBE-ALS							(PCD) (PCI) (PCI) (PAD) (D9) (P315) (PD13) (PG2) (PG4)
X-CUBE-BLE1									
							(VREF-) VISA PH2 PA6	PA2 F81 F01 FE12 F	818 FH8 F812 F012 F013
K-CUBE-BLEMGR									the second second second second second
							(VODA) (VIDT+) (PAC) (PAC)	PED (PED) (PED) (P	E13 FHT TOT TOT VOD
X-CUBE-EEPRMA1 X-CUBE-FREERTOS									
X-CUBE-FREERTOS							(VOD) (VSD) (PHE) (VSD)	and form (see) form (see	811 PHR PHR VSS PRIM
X-CUBE-ISPU							Care Care Care V	Con Con Case Case of Ca	Dana and and
X-CUBE-MEMS1							$ \frown \frown \frown \frown $		
X-CUBE-NFC4							PH4 (PH3 (PA3) (VDD)	PC4 PF12 V00 PE7 P	EIS) (VCAP) (VDD) (PHIT) (PHIZ)
X-CUBE-NFC7							- and have been have	and the first that is	and some some some
X-CUBE-SFXS2LP1 X-CUBE-SMBUS								UFBGA169 (Top view)	
								or pownoa (Lob Alem)	
							- Q [] Q 🕒	4 0 5	Q ~

Figure 271. Secure manager API configuration

Depending upon the configuration, the code is generated, and the "Secure Manager API" is added. Additional services (such as cryptography or initial attestation) can be added with the middleware.







4.18.6 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 Non Secure binaries into one: #define MCUBOOT_APP_IMAGE_NUMBER 1.



M32H573FUK_UEMIRUT_BOOK	V
les	o P /2
Project - STM32H573I-DK_OEMiROT_Boot	Copyright (c) 2018 Arm Limited. All rights reserved.
Application	* Licensed under the Apache License, Version 2.0 (the "License");
	You may not use the file except in compliance with the License.
Lep 🛋 User	You may not use this file except in complainte with the Litense.
He laes_altc	· rou may obtain a copy of the Litense at
HB log_nv_services.c	http://www.apache.ora/licenses/LICENSE-2.0
He boot hallo	nttp://www.dpuche.org/ titenses/Litense
- B cdsa alt.c	• Unless required by applicable law or agreed to in writing, software
He le eco alto	 Inters required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
	 WITHOULEW UNKERNATE OR CONDITIONS OF ANY KIND, either express or implied.
He ecp_curves_alt.c	WITHOUT WARKAWITES OF CONDITIONS OF ANY KIND, either express or implied.
Heys_map.c	* See the License for the specific language governing permissions and
HE low_level_com.c	* limitations under the License.
E low level device.c	
to low_level_flash.c	
-B low_level_obkeys.c	= #ifndefFLASH_LAYOUT_H
	#defineFLASH_LAYOUT_H
Heref and the second se	
E low_level_mg.c	↓ /* This header file is included from linker scatter file as well, where only a
E low_level_security.c	Ilimited C constructs are allowed. Therefore it is not possible to include
- E mpu_armv8m_drv.c	* here the platform_retarget.h to access flash related defines. To resolve this
- B Irsa_altc	* some of the values are redefined here with different names, these are marked
-B sha256_alt.c	* with comment.
He istartup_stm32h5xx.c	
He istm32h5xx_hal_msp.c	/* Flash layout configuration : begin ************************************
L-⊞ 🖬 tick.c	/* OEMIROT/STIROT_OEMuROT_configuration */
Doc	/*#define OEMUROT_ENABLE*/ /* Defined: the project is used for OENiROT 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.
E Output	UnDefined The FW installation uses swap mode. */
	Undefine NCUBOOT_APP_THAGE_NAMEER 1 /* 1: S and KS application binaries are assembled in one single (mage. 2: ho separates images for S and KS application binaries. */ Bdefine MCUBOOT_S_DATA_THAGE_NAMEER 0 /* 1: S data image for KS application. 0: No S data image. */ 0: No S data image. */ madefine NCUBOOT_NS_DATA_THAGE_NAMEER 0 /* 1: NS data image. */ madefine NCUBOOT_NS_DATA_THAGE_NAMEER 0 /* 1: NS data image. */ madefine NCUBOOT_NS_DATA_THAGE_NAMEER 0 /* 1: NS data image. */ madefine NCUBOOT_NS_DATA_THAGE_NAMEER 0 /* 1: NS data image. */ /* Flash layout configuration : end ************************************
ject	<
1	
Messages	File
tiv.c	
Project.out	
OEMiROT_Boot.bin	
Post-build command	
run config Appli with windows executable	
Total number of errors: 0	
Total number of warnings: 0	
rolaritumber or wainings. o	
Build succeeded	\checkmark
(A A) (A) (A) (A)	
the property in the property in the property of the	an warmen 1 /* 1. C and WC amplication binaries and accepted in any simple income
#define MCUBOOT APP I	
#define MCUBOOT_APP_I	TAGE_NUMBER 1 /* 1: S and NS application binaries are assembled in one single image.
#define MCUBOOT_APP_I	AGE NOMBER 1 /* 1: S and NS application binaries are assembled in one single image. 2: Two separated images for S and NS application binaries. */
#define MCUBOOT_APP_I	

Figure 273. 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



Г

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		STM32CubelDE	6/13/2023 5:20 PM	File folder	
STM32CubeIDE		APACHE-2.0.txt	6/9/2023 11:55 AM	Text Document	12 KI
		auto_rot_update.bat	6/14/2023 11:12 AM	Windows Batch File	1 KI
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 KI
STIROT_Appli_TrustZone		o readme.html	6/9/2023 11:55 AM	Chrome HTML Do	61 KI
STIROT_OEMuROT_Appli		README.md	6/9/2023 11:55 AM	Markdown file	6 K
71 112	ap.properties	\checkmark			
	1 [BootPathType]				
	2 bootPath=OEMiROT				
	3 MCUBOOT OVERWRITH	C ONLY=0x1			
	4 TRAILER SIZE=0x40)			
	5 CODE IMAGE ASSEM	3LY=0x1			
	6 [Secure]				
	7 S CODE REGION ST	ART=0xC018000			
	8 S CODE REGION SI	ZE=0x6000			
	9 <i>S</i> DATA=0x0				
	10 S DATA REGION ST				
	11 S DATA REGION SI	<i>E</i> =0x0			
	12 HEADER SIZE=0x400)			
	13 [NonSecure]				
	14 NS CODE REGION S	TART=0x801E000			
	15 NS CODE REGION S.	<i>IZE</i> =0xA0000			
	16 NS DATA=0x0				
	17 NS DATA REGION S	TART=0x0			
	18 NS DATA REGION S.	<i>IZE</i> =0x0			
	19 HEADER SIZE=0x400)			
	20 [Memory Region]				
	21 ROT REGION START	0x0			
	22 ROT REGION SIZE=(
	23 SCRATCH REGION S				
	24 SCRATCH REGION S.				
		REGION START=0xBE000			
	26 DOWNLOAD NS CODE				
	27 DOWNLOAD S DATA				
	28 DOWNLOAD NS DATA				

Figure 274. The map.properties file

Figure 275. Boot path project

F	File	Window	Help	L myST	🕲 🖪 🖻 🛚 🏵 🏷 🕰
STM32H573ZITXQ	Assembled_OEMiROT_Bo	ot_Pathe_Example.ioc - Pi	roject Manager >		GENERATE CODE
Pinout & Config			nfiguration	Project Manager	Tools
	STM32CubeProgrammer Path	C:\Program Files\STM		ubeProgrammer/bin/STM32 Browse	
	Boot Path Selection Select OEM_IRC	wark			
	Use Default Reference Loc Reference Location Config Files Location	C:\Users\kroutb\STM3	2Cube/Repository/STM32Cube_FW_ athProjects/Assembled_OEMiROT_I	H5_V1.2.0/Projects/STM33 Browse Boot_Pathe_Example/ROT] Edit Config Files	
	OEM-iRoT Location	C:\Users\kroutb\STM3	2Cube\Repository\STM32Cube_FW_	H5_V1.2.0\Projects\STM32 Browse	Boot Path Image
	Enable Linker File(s) Upda Reinitialize Linker File(s)	ite			Non Secure Secure
	Signature				Application Application
	Mapping	M33S	M33NS		0EM-Port
	Start address End address				STM32

UM1718 Rev 46



• Generate and build the project

Project - IAR Embedded Workbench IDE - Arm 9.20.1	
File Edit View Project ST-Link Tools Window Help	
1 0 🖌 📲 📲 🕹 🛍 🗋 1 0 C 1 🗌 🖷 🗳 1 1	> \$ H < 🔍 > 🛛 🗈 🖿 🌒 🖷 🔘 > 📜 👪 📜
Workspace 🔻 🕈 🗙	Options for node "OEMiRoT Assembled S"
OEMiRoT_Assembled_S ~	Options for node "OEMiRoT_Assembled_S"
Files 🗘	
	Category: General Options Static Analysis Runtime Choding C/C++ Compiler Output Converter Custon Build Build Actions Uniter Debugger Smulator CAD1 CAD1 CAD1 CAD1 CAD2 CMSIS DAP CD8 Server 1 jet Junk/C-Trace TI Stellaris Nu-Link Périmo ST-LINK Tird Party Driver TI MSP-FET TI XDS DK Cancel

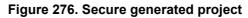


Figure 277. Non Secure generated project

Category: General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator CADI OMSIS DAP GDB Server I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Tind Party Driver TI MSP-FET	General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Cutotom Build Build Actions Configuration Pre-build command line: Pre-build command line: Pre-build command line: Post-build command line: Static Analysis Post-build command line: "SPROJ_DIRS/./././ROT_Prov Smulator CADI CMSIS DAP GDB Server I-jet J-thick/J-Trace TJ Stellaris Nu-Link PE micro ST-LINK Third Party Driver

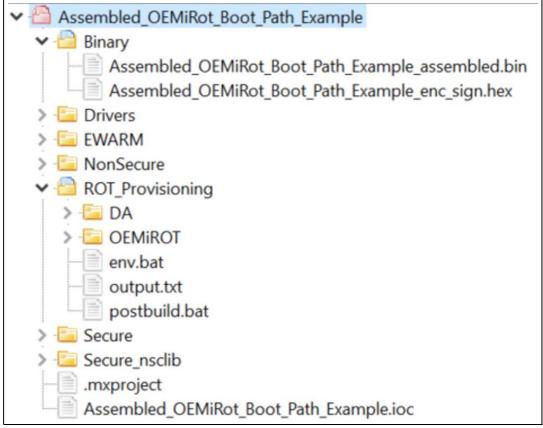


Figure 278. Compilation project

OEMiRoT_Assembled_NS.bin OeMiRoT_Assembled_NS.bin Post-built command Wed Jun 1411:3628 2023. [INF] ####################################
STM32 Trusted Package Creator v2.14.0-B03
-pb C\ST_MMS\CodeGeneration\Branch_6_9_Beta_7\OEMiRoT_Assembled\ROT_Provisioning\OEMiROT\Images\OEMiROT_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 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





• The post build command is added only for the Non Secure project.



4.18.7 How to configure OEM-uRoT (STiRot uROT) boot path

- Select an STM32H57x MCU
- Create a project with TrustZone activated (TZEN = 1), see Figure 280
- In Project Manager, save the project, see Figure 281
- Go to "Boot Path and Debug Authentication" tab, and press the Select button, see *Figure* 282
- Select "ST immutable Root of Trust (ST-iRot)", then click "NEXT", see Figure 283
- Select "OEM updatable Root of Trust (OEM-uRoT)", then click "NEXT", see Figure 283
- Select "Secure Application", then click "FINISH", see Figure 284
- 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 285*
- Generate and build the project, see Figure 288 and Figure 289

e > STM32MP257CAlx > Untitled - Tools >			GENERATE CODE
xisting Projects		New Project	Manage software installations
Recent Opened Projects		I need to :	Check for STM32CubeMX and embedded software packages updates
MP25xx.ioc Last modified date : 29/05/2024 16:32:01	MX	Start My project from MCU	CHECK FOR UPDATES
MP25.ioc Last modified date : 29/05/2024 15:12:03	MX	ACCESS TO MCU SELECTOR TrustZone feature available ×	Install or remove embedded software packages INSTALL / REMOVE
ADC_ContinuousConversion_TriggerSW_Init.ioc Last modified date : 29/05/2024 10:27:51	MX	Start My ? Do you want to create a new project : acc without TrustZone activated ? O with TrustZone activated ?	
CubeMXH56.ioc Last modified date : 24/05/2024 12:10:59	MX	Start My OK ACCESS TO EXAMPLE SELECTOR	Bootflash MCUs:
I5project.ioc Last modified date : 17/05/2024 14:15:10	MX	NOULD TO LOWINGLE SELECTOR	Configure simultaneously boot and application projects
Other Projects	ित्		

Figure 280. Boot path project

Figure 281. Save the project

TM32 CubeMX		File		Window	Help	L myST	🎯 🖪 🕒 🛛 🖓 🍏 🔆 🖅
Home		New Project Load Project	CIrl-N Cirl-L	Manager			GENERATE CODE
	Pinout & Configura	Import Project Save Project	CH-I CH-S	Clock Configuration		Project Manager	Tools
		Save Project As Close Project Generate Report		lettings Vame		OemuRot_Bp	
		Recent Projects	•	ocation		C:\Users\arbisour\Desktop\tmp\ex\	
		Exit	Cel-X Project	Structure		Secure Project Non Secure Project	
			Applica	tion Structure		Advanced	✓ □ Do
		s	Toolcha	ain Folder Location		C:\Users\arbisour\Desktop\tmp\ex\OemuRot_Bp\	
			Toolcha	ain / IDE		EWARM Version	V9.20 🗸 🖂 Ge
		ation	Linker S	ettings			
						M33S M33NS	
			Minimu	m Heap Size		0x200 0x200	
			Minimu	m Stack Size		0x400	
			Thread-i Cortex	afe Settings //33S			
			D En	able multi-threaded support			
			Thread	I-safe Locking Strategy		Default - Mapping suitable strategy depending on RTOS select	tion.
			Cortex	12210			



File	Window	Help	L myST	🕲 🖪 💶 X 🖓 🍑 🔆 ភ
· · · · · · · · · · · · · · · · · · ·	OemuRot_Bp.ioc - Project Manager	\rangle		GENERATE CODE
Pinout & Configuration	Clock Con	figuration	Project Manager	Tools
	STM32CubeProgrammer Path		iles\STMicroelectronics\STM32Cube\STM	132CubeProgrammer/bin\STM32_Programmer_CL Browse
	Debug Authentication	nerate DA Folder		
	Boot Path Selection Select No boo	ot path selected (Nothing to cor	figure)	
			Application (Default Boot Path)	
and Debug Authentication				

Figure 282. Boot path and debug authentication panel



K 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	
O Secure Application	Non Secure Secure	O Secure Application	
O OEM immutable Root of Trust (OEM-IRoT)		OEM updatable Root of Trust (OEM-uRoT) OEM-uBoT	
ST immutable Root of Trust (ST-IRoT)	ST-IRoT STM32	ST-IROT STM32	
BACK	EXT FINISH CANCEL	BACK NEXT FINISH CANC	EL

UM1718

MX Boot Path				×
Define	your boot Path			
TZEN=1 'Dua	Non Secure / Secure' project		Graphical Boot	t Representat
Boot St	ep#3		Non Secure	Secure
Set	oure Application			
			Application	Secure Application
O Sec	ure Manager Non Secure Appli	cation		0EM-uRoT
				ST-IRoT
			STN	132
	TZEN=1 'Dual Boot St	Define your boot Path TZEN=1 'Dual Non Secure / Secure' project Boot Step#3	Define your boot Path TZEN=1 'Dual Non Secure / Secure' project Boot Step#3	TZEN=1 'Dual Non Secure / Secure' project Boot Step#3 • Secure Application • Secure Manager Non Secure Application

Figure 284. Final boot path stage

Figure 285. Boot path and debug authentication tab

File	Window	Help	L myST		🧐 두 🕨 🗴	K 🖸 🏷
> STM32H573VITxQ > 0e	muRot_Bp.ioc - Project Manager >				GENERATE	E CODE
Pinout & Configur		Clock Configuration	n	Project Manager	Tools	
	STM32CubeProgrammer Path			ner/bin/STM32_Programmer_CLBrewse		
		C. Program Priesto i Microen		herbinds1M32_Programmer_ct_		
	Configure Generat	te DA Folder				
	Boot Path Selection					
	Select ST_IROT_U	IROT 🔍				
	Use Default Reference Location	n				
	Reference Location	C:\Users\arbisour\STM32Cub		ojects/STM32H673I-DK/ROT_Pr Browse		
	Config Files Location	C:\Users\arbisour\Desktop\tmp	plex\OemuRot_Bp\ROT_Provisioning\STiRO	Edit Config Files		
	Linker				Boot Path Image	
	OEM-uRoT Location	C:\Users\arbisour\STM32Cub	e\Repository\STM32Cube_FW_H5_V1.2.0\Pr	ojects/STM32H573I-DK/Applica Browse	Non Secure	Secure
	Enable Linker File(s) Update					
	Reinitialize Linker File(s)					
						Secure
	Signature				Application	Application
	Sign Binary(ies)					
						0EM-uRoT
	Mapping	M335	M33NS			UEM-UHOI
						ST-IRoT
	Start address					
	End address				STM	132



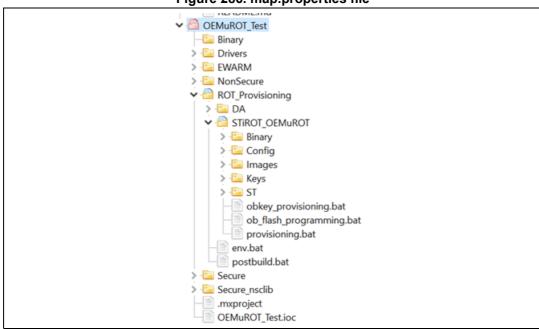


Figure 207 Code concretion with FMADM

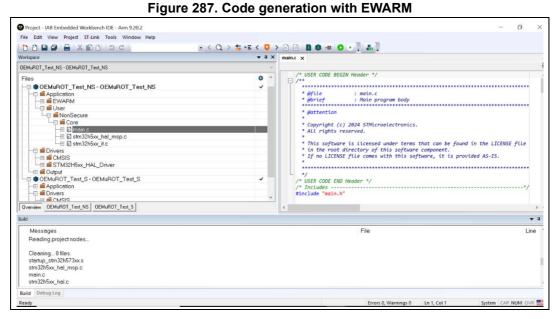
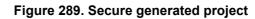


Figure 286. map.properties file



gaio 200		generatea project
Project - IAR Embedded Workbench IDE - Arm 9.20.2		
File Edit View Project ST-Link Tools Window Help		
1 1 🗈 🖻 🔮 🚔 🙏 🗈 🖬 1 5 C 🗌	- < Q	> \$ == < 🔍 > 🕢 🖻 🗈 🖷 🔍 🕨 📜 🏭
Workspace	Options for node "OEMul	
OEMuROT_Test_NS - OEMuROT_Test_NS	Options for hode OEMu	KOT_lest_NS ^
	-	
	Category:	
	General Options Static Analysis	
	Runtime Checking	
	C/C++ Compiler	Build Actions Configuration
	Assembler	
🕀 🗈 main.c	Output Converter	Pre-build command line: 2/
— ⊕ stm32h5xx_hal_msp.c	Custom Build	
– ⊞ 🗟 stm32h5xx_it.c	Build Actions	Post-build command line:
🛛 🖂 🗖 Drivers	Linker	Post-build command line: "\$PROJ_DIR\$/.//ROT_Provisioning/postbuild.bat" "nonsecure"
	Debugger Simulator	
IIII	CADI	
Let Output	CMSIS DAP	
CEMUROT_Test_S - OEMuROT_Test_S	GDB Server	
→ → → → → → → → → → → → → → → → → → →	I-jet	
	J-Link/J-Trace	
Overview OEMuROT Test NS OEMuROT Test S	TI Stellaris	
	Nu-Link PE micro	
Build	ST-LINK	
Messages	Third-Party Driver	
	TI MSP-FET	
Reading project nodes	TI XDS	
Cleaning 0 files.		
startup_stm32h573xx.s		
stm32h5xx hal_msp.c		OK Cancel
main.c	L	
stm32h5xx hal.c		

Figure 288. Non secure generated project



/orkspace	-	\$\$ HE < \$\$ > \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$	
EMuROT_Test_NS - OEMuROT_Test_NS	Options for node "OEMuROT	_Test_S*	×
Fues User Core Stra255xx,hal_msp.c Stra255xx,hal_msp.c Stra255x,hal_msp.c Stra2555x,hal_msp.c Stra255x,hal_msp.c Stra555x,hal_msp.c Stra555x,hal_msp.c	Calegoy: General Options Static Analysis Runtime Clocking C(C++ Compler Assembler Output Converter Caston Build Build Actions Build Actions Build Actions Build Actions Debugger Simulator OADI COB Server 13et Ti Selaris Nu-Link Patiens Ti Selaris Nu-Link Third-Party Diver Ti NDS	Post-build command line:	ed



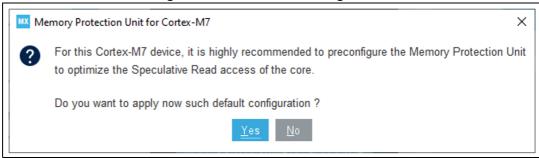
Go through the following steps:

- 1. Select an STM32H7S3Vx MCU (*Figure 290*)
- 2. A popup (see *Figure 291*) 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 292*).
- 4. Go to "Boot Path and Debug Authentication" tab and press the Select button (*Figure 293*).
- 5. Select "ST immutable Root of Trust (ST-iRoT)", then click "NEXT" (Figure 294).
- 6. Select "Application", then click "FINISH" (*Figure 295*).
- 7. The panel of boot path configuration is displayed (see *Figure 296*), use it to configure the boot path in the "Boot Path and Debug Authentication" tab.
- 8. Generate and build the project (see *Figure 297*).

U/MPU Selector Board Selector Exam	nple Selector	Cross Sel	ector						
🔺 🗋 🔂 🔾] .	Features	Block Diagram	Docs & Re	sources CAD Resources	📑 Datasheet	📑 Buy	🖃 Start Project
Commercial Part Number	\sim	4	STM32H7 Serie	es					
STM32H7S3V8H6	+-	ж	STM 3	32H7S3V8H6		rmance Arm Cortex-M7 I h DSP, cache, USB HS P		, 64KB Bootflash, 6	20KB
RODUCT INFO	~		ACTIVE		Unit Price for 10	0kU (US\$):4.1131			
Segment	>			s in mass production			- T	FBGA 100 8x8x1.2 P 0.8 mm	1
Series	>								
Line	>					n-performance Arm [®] Cortex [®] -N (EPU) which supports Arm dou			
Line Marketing Status	>		Cortex -M precision	17 core features a fli data-processing inst	oating point unit structions and da	(FPU) which supports Arm dou ata types. The Cortex -M7 core	ble-precision (II includes 32 Kb	EE 754 compliant) and s tes of instruction cache	single- and 32
Marketing Status			Cortex -M precision Kbytes of	17 core features a fli data-processing inst	oating point unit structions and da 2H7Sxx8 devices	(FPU) which supports Arm dou	ble-precision (II includes 32 Kb	EE 754 compliant) and s tes of instruction cache	single- and 32
	>		Cortex -M precision Kbytes of enhance	17 core features a fli data-processing ins data cache. STM32 application security	oating point unit structions and da 2H7Sxx8 devices	(FPU) which supports Arm dou ata types. The Cortex -M7 core	ble-precision (II includes 32 Kb	EE 754 compliant) and s tes of instruction cache	single- and 32 PU) to
Marketing Status Price	>	MCUs/	Cortex -M precision Kbytes of	I7 core features a fli data-processing ins data cache. STM32 application security	oating point unit structions and da 2H7Sxx8 devices	(FPU) which supports Arm dou ata types. The Cortex -M7 core support a full set of DSP instri	ble-precision (II includes 32 Kb uctions and a m	EE 754 compliant) and a des of instruction cache emory protection unit (MF	single- and 32 PU) to
Price Package	>		Cortex -M precision Kbytes of enhance	17 core features a fli data-processing ins data cache. STM32 application security	oating point unit structions and da 2H7Sxx8 devices	(FPU) which supports Arm dou ata types. The Cortex -M7 core	ble-precision (II includes 32 Kb uctions and a m	EE 754 compliant) and s tes of instruction cache	single- and 32 PU) to
Marketing Status Price Package Core	>		Cortex -M precision Kbytes of enhance	17 core features a fli data-processing ins data cache. STM32 application.security m	oating point unit structions and da 2H7Sxx8 devices	(FPU) which supports Arm dou ata types. The Cortex -M7 core support a full set of DSP instru- support a full set of DSP instru-	ble-precision (II includes 32 Kb uctions and a m	EE 754 compliant) and stress of instruction cache emory protection unit (MF	single- and 32 PU) to Export
Markeling Status Price Package Core Coprocessor	>		Cortex -M precision Kbytes of enhance	17 core features a fli data-processing ins data cache. STM32 application.security m	oating point unit structions and da 2H7Sxx8 devices	(FPU) which supports Arm dou ata types. The Cortex -M7 core support a full set of DSP instru- support a full set of DSP instru-	ble-precision (II includes 32 Kb uctions and a m	EE 754 compliant) and stress of instruction cache emory protection unit (MF	single- and 32 PU) to Export
Markeing Status Pice Package Core Coprocessor EMORY	> > > > >		Cortex -M precision Kbytes of enhance	17 core features a fli data-processing ins data cache. STM32 application.security m	oating point unit structions and da 2H7Sxx8 devices	(FPU) which supports Arm dou ata types. The Cortex -M7 core support a full set of DSP instru- support a full set of DSP instru-	ble-precision (II includes 32 Kb uctions and a m	EE 754 compliant) and stress of instruction cache emory protection unit (MF	single- and 32 PU) to Export
Marketing Status Price Proce Coprocessor EMORY Lab. = 64 (ABytes) 04	> > > > >		Cortex -M precision Kbytes of enhance	17 core features a fli data-processing ins data cache. STM32 application.security m	oating point unit structions and da 2H7Sxx8 devices	(FPU) which supports Arm dou ata types. The Cortex -M7 core support a full set of DSP instru- support a full set of DSP instru-	ble-precision (II includes 32 Kb uctions and a m	EE 754 compliant) and stress of instruction cache emory protection unit (MF	single- and 32 PU) to Export
Marksting Status Price Proce Core Coprocessor EMORY Isah = <6 (kBytes) of EEPROM = 0 (Bytes)	> > > > >		Cortex -M precision Kbytes of enhance	17 core features a fli data-processing ins data cache. STM32 application.security m	oating point unit structions and da 2H7Sxx8 devices	(FPU) which supports Arm dou ata types. The Cortex -M7 core support a full set of DSP instru- support a full set of DSP instru-	ble-precision (II includes 32 Kb uctions and a m	EE 754 compliant) and stress of instruction cache emory protection unit (MF	single- and 32 PU) to Export
Marketing Status Price Package Core Coprocessor EMORY Stath = 64 (05/sst) 64 55 56 57 50 50 50 50 50 50 50 50 50 50	> > > > >		Cortex -M precision Kbytes of enhance	17 core features a fli data-processing ins data cache. STM32 application.security m	oating point unit structions and da 2H7Sxx8 devices	(FPU) which supports Arm dou ata types. The Cortex -M7 core support a full set of DSP instru-	ble-precision (II includes 32 Kb uctions and a m	EE 754 compliant) and stress of instruction cache emory protection unit (MF	single- and 32 PU) to Export
Marketing Status Price Package Core	> > > > >		Cortex -M precision Kbytes of enhance	17 core features a fli data-processing ins data cache. STM32 application.security m	oating point unit structions and da 2H7Sxx8 devices	(FPU) which supports Arm dou ata types. The Cortex -M7 core support a full set of DSP instru-	ble-precision (II includes 32 Kb uctions and a m	EE 754 compliant) and stress of instruction cache emory protection unit (MF	single- and 32 PU) to Export

Figure 290. Boot path project





UM1718 Rev 46

UM1718



Figure 2	292.	Configure	the	project
----------	------	-----------	-----	---------

Home > STM32H7S3V8Hx >	H7S_StiRot_Demo.ioc - Project Mar	nager 🔪			GENERATE CODE
Pinout & Configura	ation	Clock Configuration	l l	Project Manager	Tools
Project	, Project Settings Project Name Project Location Project Structure Application Structure Toolchain Folder Location	H7S_StiRot_Demo		Brow	
	Toolchain / IDE - Linker Settings	EWARM ~	Min Version V9.30 App 0x200 0x400	Generate Under Root ExtMemLoader 0x200 0x400	
	Thread-safe Settings	Default – Mapping suitable strate	gy depending on RTOS selection.		
	Thread-safe Locking Strategy Mcu and Firmware Package Mcu Reference Firmware Package Name and Version Z Use Default Firmware Location	Default – Mapping suitable strate STM32H7S3V8Hx STM32Cube FW_H7RS V1.1.0RC	1		
	Firmware Relative Path	C:/Users/bekrisli/STM32Cube/Rep		1.1.0RC1 Brow	se

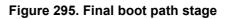
Figure 293. Select the project

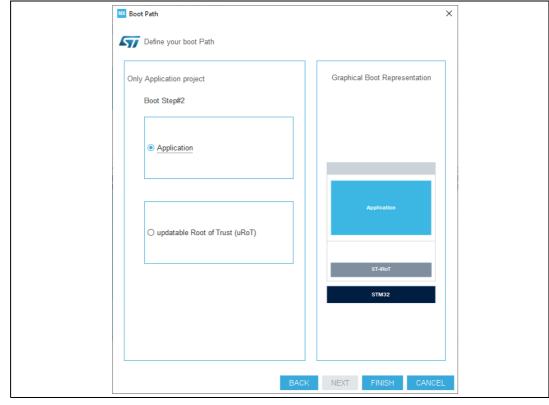
Home $ angle$ STM32H7S3V8Hx $ angle$ H	17S_StiRot_Demo.ioc - Project Manaş	ger 🔪		GENERATE CODE
Pinout & Configurat	tion C	Clock Configuration	Project Manager	Tools
Project	/Debug Authentication Configure Generate DA Folde /Boot Path Selection	Program Files/STMicroelectronics/STM32Cube	STM32CubeProgrammer/bin/STM33 Brewse	
Code Generator			Application (Default Boot Path)	
Advanced Settings				
Boot Path and Debug Authentication				



Only Application project Bot Step#1 Application OEM immutable Root of Trust (OEM-iRoT) STM32				1
Only Application project Boot Step#1 Application O DEM immutable Root of Trust (DEM-iRoT) ST-IROT STM32	MX Bo	ot Path	×	
Boot Step#1 Application OEM immutable Root of Trust (OEM-iRoT) ST-IROT STM32	L77	Define your boot Path		
O OEM immutable Root of Trust (OEM-iRoT)	Oni		Graphical Boot Representation	
ST-IRoT STM32		O Application		
STM32		O OEM immutable Root of Trust (OEM-iRoT)		
		ST immutable Root of Trust (ST-iRoT)		
		BACK	NEXT FINISH CANCEL	

Figure 294. First boot path stage





UM1718 Rev 46



Home > STM32H7S3V8Hx > I	H7S_StiRot_Demo.ioc - Project	· Manager >		GENERA	
Pinout & Configura	ation	Clock Configuration	Project Manager		Tools
Project	STM32CubeProgrammer Path Use Default Path /Debug Authentication Configure Generate [/Boot Path Selection Select ST_JROT Q	C Program Féles/STMicroelectronics/STM32Cube/S A Folder	TM32CubeProgrammer/bin/STM32 Browse		
Code Generator	Use Default Reference Location Reference Location Config Files Location	C Usero baknali STM32Cubel Repository (STM32Cu C \STM32CubeMY, Projects H78_StiRot, Demo/RO	[_Provisioning\STiROTEdit Con		
Advanced Settings	Reinitialize Linker File(s)			Aggenation	
Boot Path and Debug Authentication	Start address End address	Appli 0x24006400 0x2401FFFF		17.840T 611622	

Figure 296. Boot path and debug authentication panel

Figure 297. Generate the code

Home > STM32H7S3V8Hx >	H7S_StiRot_Demo.ioc - Proje	t Manager >		GENERATE CODE
Pinout & Configur	ation	Clock Configuration	Project Manager	Tools
	STM32CubeProgrammer Path Use Default Path Debug Authentication Configure Generate Boot Path Selection Select ST_IROT		STM32CubeProgrammer/bm/STM32 Browse	
	Use Default Reference Location Config Files Location Linker	n C:UsersUsekristist C:UsersUsekristist C:USTM32CubeMX Project language : C	fully generated under : rojects/H/S_StRot_Demo	
	Reinitialize Linker File(s)	Open Folder Open F	riyed Close	Application
	√Mapping Start address End address	Appli 0:224000.100 0:22401FFFF		STM22

Figure 298. Application IDE directories

✓ A H7S_StiRot_Demo	
> 🔚 Appli	
—🔄 Binary	
> 🔚 Drivers	
> 🔚 EWARM	
V 🔤 ROT_Provisioning	
🗸 🔤 DA	
> - 🔚 Binary	
🗸 🔤 STIROT	
> 🔚 Binary	
> 🔚 Config	
— iii env.bat	
postbuild.bat	
.mxproject	



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.

STM 12 U CubeMX	File	Window	Help		ঙা	_® ∏⊡ X ೧	57/
Home >			New Project	N	Aanage software i	nstallations	
Recent Opened Pr ADC_DMA_Trans Last modified date : 3 LPBAM_LPOPIO Last modified date : 3 bitlet.ioc Last modified date : 4 MP15-M4.ioc Last modified date : 4 TriabH5_MMT.io Last modified date : 3 Other Projects	isfer io: 2605204 16 47 01)_IOTogle ioc 2703204 16 49 12 1601204 18 53 29 0412203 18 47 18 06 2771 1203 22 08 40	₩X ₩X ₩X ₩X	I need to : Start My project from MOU Access To MOU SELECTO Start My project from ST Boo Access To BOARD SELECT Start My project from Europh Access To Example select	rd Der	Install or remove	Creck FCI UFCATES embedded software packages embedded software packages RSTALL/REMOVE emeredded software packages	•
STM:2 CubeMX Home STN32M	File	Pinout & Centigur		Help	٠		57
Existing Project	ts		New Project		Manage s	oftware installations	
OpenAMP_p Last modified da	te : 1710/2024 15:01:29 rojec_linker_files.ioc te : 0910/2024 14:31:57	MX MX	I need to : Start My project from M ACCESS TO MOUSEL Start My project from S	ECTOR		for STM32CubeMX and embedded a CHECK FOR UPDATES In remove embedded software packa INSTALL / RENOVE	
Last modified da cubernx_proj Last modified da STM32H5030	ne : 0810/2024 15:33:15 CBTx.ioc ne : 0509/2024 11:26:18	MX MX MX	ACCESS TO BOARD SE Start My project from E ACCESS TO EXAMPLE S	Example	•	STM2007 Image: Compared and the co	Þ
	ion information notice	Ľα				About STM32	

Figure 299. Home page without the login form

UM1718

UM1718 Rev 46



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 300*).
- 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 301*).
- 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 302*).
- 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 303*).
- If the login action is performed successfully, myST display changes as illustrated in the *Figure 304*.
- 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 305*).

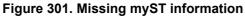


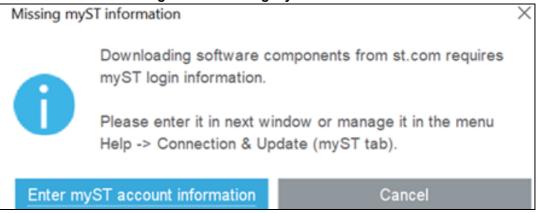
0	File		Window		Help			B f	DX	n a +	: 5
STM32MP233AAJX	- Unbedd	led Software Packages N	tanager.						X	_	
inout & Configuration	100	STM32Cube MCU F Releases Information	n was last refree	hed 4 days ago	e packs releas	es		+	-	Tools	
. स्रो स	Infrect AV 51	n RealThres M32Cube MCU Pack		GGER WI W STMicroelectron			ortGmbH reddedOffice	wolfsst.	6		
airies AirZ		Description				Installed	Version Av	stable Versio			
tem Core	• STN	432MP2							1		
ing	STN	132Cube MCU Pecka	oe for STM32MF	2 Seties (Size): 11	O MED			1.2.0RC7	10		
**											
mectivity	STN	132Cube MCU Packag	pe for GTM 32M7	12 Series		1.0.0		1.0.0			
tineda	► 575	43296							8		
wity.	Details	0000									
0.19	STM32Cu	beMP2 Firmware Pa	ckage V1.2.0R0	7 / 14-October-20	224						
nputing				Without a lat		-	-				
deware and Software Packs		1.2.0RC7 release of 5	110.320089192	11111111		Action reads the					
te and Debug						Authentication					
rer and Thermal	-					-					
ties	From Lo	cal			Refresh	a ostat	Remove	Citte			
er Dictatentite (tentimed S	File		Window	Q []	Q Help			= ® f		0@}	
Dilanda Manual		ed Software Packages M	Window		Help				a > X ×	೧ @}	-
		BTM32Cube MCU P Releases Information	Window Anaper Packages and e	mbedded software hed 4 days ago.	Help e packs releas	-	•	= • •	A X × ation		
kisting Projects	Imbeds	BYM32Cube MCU P Ralesses Information Resiltives M32Cube MCU Pace	Window tenaper h was last refres	mbedded software hed 4 days app.	Help e packs releas	otan p		+	× -		-
	Imbeds	STM32Cube MCU P Releases Information RealThread	Window tenaper h was last refres	mbedded softwar hed 4 deys ego GGER W	Help e packs releas	otan p	ortGmbH	+	× ation		< 45
kisting Projects		BYM32Cube MCU P Ralesses Information Resiltives M32Cube MCU Pace	Window Anager Packages and e n was last rafes ages	mbedded softwar hed 4 days epo GGER W W STMicroelector	Help e packs releas	otan p	ortGmbH	+	× ation	15	< 45
Manufix (Helling)	Lebeds	BTM32Cube MCU P Releases Information RealThree M32Cube MCU Pace Description	Window Anape Packages and e was last refres ages of the STM3200	mbedded software hed 4 days epo GOER W W STMcroelector Carlies	Help e packs releas ES and ES Ces	otas p anta Erri Installed	ortGmbH	+ wortsst	× ation	nd embedde I <mark>OATES</mark> software par	d sof
Kisting Projects Recent Opened Projects smak.ioc Let medified date : 11/10/20	E trabedd Million C N STU C STU	BTM32Cube MCU P Realises Information M32Cube WCU Pace Description 732Cube MCU Packag 732Cube MCU Packag	Window Anagen An	mbedded software hed 4 days apo. Gotin W W STMicroelector Garies Garies Garies	Help e pecks release ES em Stas Ces 5 M(t)	otas p anta Erri Installed	ortGmbH	+ worss	× ation	nd embedde POATES	d sof
Kisting Projects Recent Opened Projects smak.ioc Let medified date : 11/10/20 OpenAMP_project_ink Let medified date : 09/10/20 ETH_LINKER_FILE_et	Mineco Mi	STM32Cube MCU P Releases Information RealTypes M32Cube MCU Packer I32Cube MCU Packer	Window Anagen An	mbedded software hed 4 days apo. Gotin W W STMicroelector Garies Garies Garies	Help e pecks release ES em Stas Ces 5 M(t)	otas p anta Erri Installed	ortGmbH	+ worsse ITTA_ 1.1.0	× ation	nd embedde I <mark>OATES</mark> software par	d sof
Kisting Projects Kisting Projects Smak.loc Let modified date : 11/10/20 OpenAMP_project_lini Let modified date : 09/10/20	Mineco Mi	BTM32Cube MCU P Realises Information M32Cube WCU Pace Description 732Cube MCU Packag 732Cube MCU Packag	Window Anagen An	mbedded software hed 4 days apo. Gotin W W STMicroelector Garies Garies Garies	Help e pecks release ES em Stas Ces 5 M(t)	otas p anta Erri Installed	ortGmbH	+ worss	× ation	nd embedde ICATES Software par CVE	d sof
kisting Projects smak.ioc Let medified date : 11/10/20 OpenAMP_project_ink Let medified date : 09/10/20 ETH_LINKER_FILE_et	Unification Unifi	BTM32Cube MCU P Realises Information M32Cube WCU Pace Description 732Cube MCU Packag 732Cube MCU Packag	Window Anapa Packages and en n use leat refree at SEE ope for STM32CO pe for STM32CO pe for STM32CO	mbedded softwar hed 4 days ego GGER W w STMicroelection Geries Geries (Size : 70 5 Series (Size : 57 2	Help e pecks release ES em Stas Ces 5 M(t)	otas p anta Erri Installed	ortGmbH	+ worss	× ation	nd embedde I <mark>OATES</mark> software par	d sof
Kisting Projects Kisting Projects Recent Opened Projects Smak.ioc Lear modified date : 11/10/20 OpenAMP_project_link Lear modified date : 08/10/20 ETH_LINKER_FILE_et Lear modified date : 09/10/20	Letters	ETH32Cube MOU P Releases Information RealTives MI2Cube MOU Packag IS2Cube MOU Packag IS2Cube MOU Packag IS2Cube MOU Packag DeCD Fernware Pack	Window Anapa Packages and en n use leat refree at SEE ope for STM32CO pe for STM32CO pe for STM32CO	mbedded softwar hed 4 days ego GGER W w STMicroelection Geries Geries (Size : 70 5 Series (Size : 57 2	Help e pecks release ES em Stas Ces 5 M(t)	otas p anta Erri Installed	ortGmbH	+ worss	× stier	nd embedde ICATES Software par CVE	d sof
Kisting Projects Kisting Projects Kisting Projects Kisting Projects Kisting Projects Kisting Project Link Kisting Control Cont	Contractor Contra	ETH32Cube MOU P Releases Information RealTives MI2Cube MOU Packag IS2Cube MOU Packag IS2Cube MOU Packag IS2Cube MOU Packag DeCD Fernware Pack	Window Anapa Packages and e to use lief refee pert acceler pert STM 3200 perfor STM 3200 perfor STM 3200 perfor STM 3200	mbedded softwar hed 4 days ego GGER W w STMicroelection Geries Geries (Size : 70 5 Series (Size : 57 2	Help e pecks release ES em Stas Ces 5 M(t)	otas p anta Erri Installed	ortGmbH	+ worss	× stier	nd embedde IOATES Software par OVE	d sof
Kisting Projects Kisting Projects Recent Opened Projects Smak.loc Leat modified date : 11/10/20 OpenAMP_project_link Leat modified date : 08/10/20 ETH_LINKER_FILE_et Leat modified date : 08/10/20 cubems_projects.loc Leat modified date : 08/10/20 STIM32H503CBTx.loc	Contractor Contra	ETH32Cube MCU P Releases Information Reaffree M220de MCU Packey M220de MCU Packey M220de MCU Packey MCU Packey MCU Packey MCU Packey Dec0 Firmware Pack Dec0 Firmware Pack	Window Anapa Packages and e to use lief refee pert acceler pert STM 3200 perfor STM 3200 perfor STM 3200 perfor STM 3200	mbedded softwar hed 4 days ego GGER W w STMicroelection Geries Geries (Size : 70 5 Series (Size : 57 2	Help e pecks release ES em Stas Ces 5 M(t)	otas p anta Erri Installed	ortGmbH	+ worss	× stier	nd embedde IOATES Software par OVE	d sof

Figure 300. Install or remove a software package

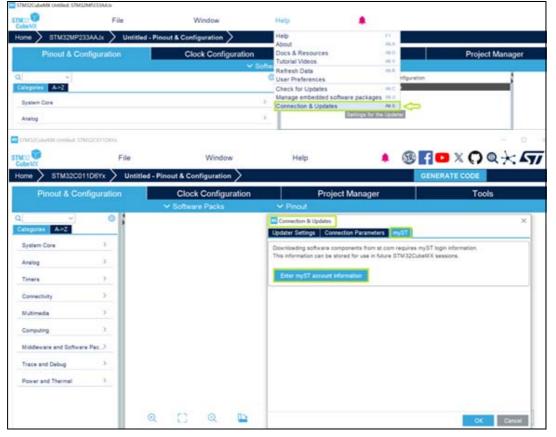














CubeMX File	Window Help		
Home > STM32MP233AAJx > Untitled	d - Pinout & Configuration >		
Pinout & Configuration	Clock Configuration	RIF	Project Manage
Pinout & Configuration	Clock Configuration	New users need first to createan account the login with the created account New user? Mew user? Mew user? New user? • Participate to ST Event • Stay informed with ST e • Get help with ST Online • Discuss on the ST Com	Personalized 5 aNewsletters 5 Support munity
Power and Thermal Utilities		Benefit from our Online Download Software Order free samples	
Other	Remember me on this computer.	Manage your weekly pr Buy ST Products & Too Create Account	

Figure 303. User Authentication Dialog

Figure 304. The myST display after a user logs in

Updater Settings Updater Settings Connection Parameters	×
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 for this session	
OK.	Cancel



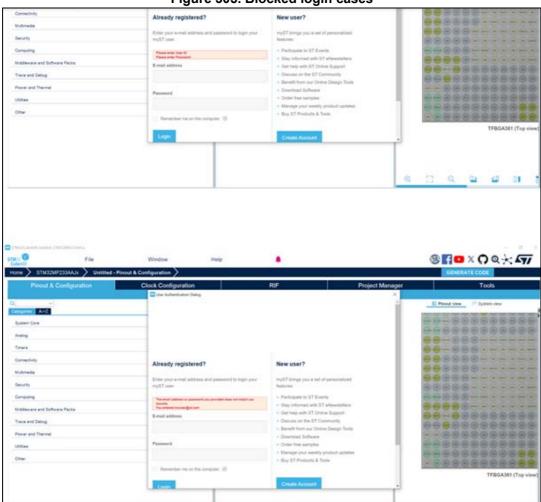


Figure 305. Blocked login cases

4.19.2 Create a my.st.com account

The account can be created through STM32CubeMX:

- Click on "Create Account" button (*Figure 303*)
- Fill the account creation form (*Figure 306*)
- Click on "Register" button to create a new my.st.com account (Figure 306)



nti suori adi suori adi adi suori adi suori ad	
aut autoritation backet bac	
and canadad" and an and an	
and canadad" and an and an	
And and a second s	
And and a second s	
waan' Marene Marene Marene Marene Marene Marene Marene Marene Marene Marene Marene	
hange hangelen hangel	
hange hangelen hangel	
hange hangelen hangel	
hange hangelen hangel	
aangi . Sano na Sano hayanga . Sano na . Sano na . Sano na .	
Alap : han m han m han han han han han han han han han han han	
many. Tao ing Configure Co	
haar ay haardaga haardag haaraag hahaaga hahaaga hahaaga hahaaga hahaaga hahaaga hahaaga hahaaga hahaaga hahaaga hahaaga hahaaga hahaaga hahaaga hahaaga hahaan hahaa ha	
haningkapan haninan haninanan galakanan hun kuala	
Sancion Marinan Tano ou Spinistan Nau Anato	
Sancion Marinan Tano ou Spinistan Nau Anato	
kadinasa Santon Iphanasa Nasikata	
land og gehalener	
lani og Iphalane Nar kalle	
And Andre	
Pears more cal Propp Harrison Pad modified for the engineering property of the statement of the of th	-
1 Jacoust the Berry of part"	

Figure 306. Account creation form

4.19.3 Password restoration

If the user forgets their password, they can reset it 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 307*).

r igaio con Enter	
User Authentication Dialog	×
Already registered? Enter your e-mail address and password to login your myST user.	New user? myST brings you a set of personalized features:
E-mail address	O Finite of the second se
Password Remember me on this computer.	Benefit from our Online Design Tools Download Software Order free samples Manage your weekly product updates Buy ST Products & Tools
Login Forgot password?	Create Account

Figure 307. Enter the email address

4. Click on the "Reset my password" button (*Figure 308*). 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.

Figure 308.	Password	restoration
-------------	----------	-------------

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 309. Reset password form

Reset password	
E-mail address*	se note that password cannot contain your for full name (even parts of it), over, it must: at least 12 characters in length d contain spaces ain characters from at least three out of allowing four categories: glish uppercase characters (A-Z) glish uppercase characters (a-Z) olight contrast of these allowed calc characters =-l@#5%-&+=T(0) ~_{co}, 70

If you suspect that your identity has been stolen, or that your account has been compromised, it is important to 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:

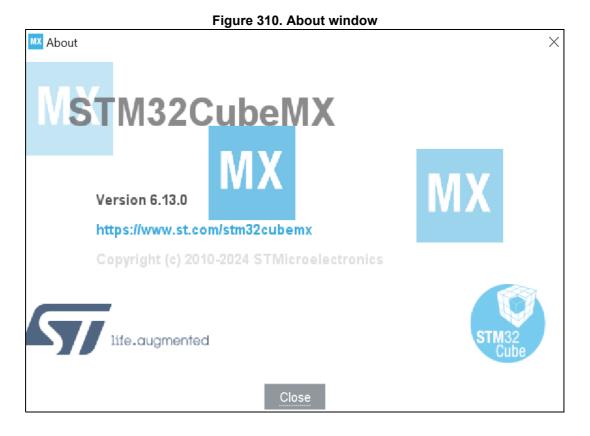
./STM32CubeMX login < email_adress> <password> <remember_me>

"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

5.1 External Tools

This panel is accessible from the home page. It provides an overview of the tools relevant for the STM32 product portfolio (see *Figure 311*):

- click 🚺 to open the tool webpage on *www.st.com*
- click () to launch the tool.

STM32CubeMX Untitled		J		- 0
	File	Window	Help	🎯 🖪 🖻 🎽 🔆 🗸
ome				
Existing Projects		New Project		Manage software installations
Recent Opened Proje	cts	I need to:		Check for STM32CubeMX a
test1.ioc Last modified date : 22/10	MX 0/2020 19:09:37		ect from MCU	CHECK FOR UPDATES
Other Projects	Ē		MCU SELECTOR	INSTALL / REMOVE
	LQ		OARD SELECTOR	STM32 Tools
			ect from Exam	
		ACCESS TO EX	AMPLE SELECTOR	STMicroelectronics Application Tools
				About STM32 Fxternal Tools

Figure 311. ST Tools



5.2 Compare Projects

The **Compare Projects** feature is a newly integrated tool in STM32CubeMX, designed to enable the comparison of two projects, whether they are based on the same or 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 tool is composed of:

- Two main panels named respectively 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 respectively 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 between the two projects).
 - If the user wants to see only the parameters that are different between the two projects, they can check "Show differences only".
- Refresh button: once it is 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.

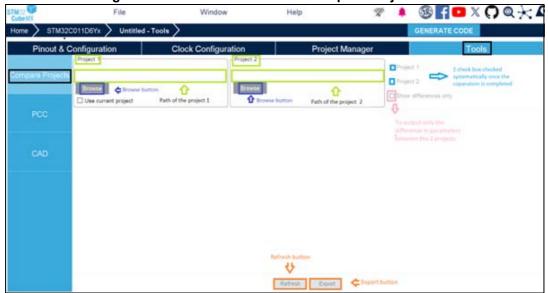


Figure 312. User interface of the Compare Projects tool



5.2.2 Steps of comparing two projects

- 1. Load the first project from the local device (*Figure 313*).
 - Once the first .ioc is loaded, a popup appears to indicate to the user that they need to upload the second .ioc.

Note: For the first project, the user can use the .ioc file of the opened project in STM32CubeMX instead of uploading one from the local device.

STM32CubeMX Untitled:	STM32C011D6Yx						- 🗆 ×
STM32	File	Window	Help	Ŕ	۰	🥸 📑 🕒 X	$0 < \times 4$
Home > STM320	C011D6Yx 🔰 Untitled - Too	ls >				GENERATE CODE	
Pinout & C	Configuration	Clock Configuration	Project Ma	anager		Tools	s
Compare Projects	Project 1 Browse Browse button Use current project	Project 2	» ق (م	× Ce 22 8:	Proje		
PCC		ELASH_init_issue.ioc Sele	ct the IOC file		4	Window appears after clicking 'Browse' button	
CAD		File Name FLASH_init_issue.ioc	• ##1507.00.00				
		Files of Iypes STM32CubeMX project		Cancel			
			Refresh Export				

Figure 313. Load the first .ioc file

- 2. Load the second project (*Figure 314*).
 - After uploading the second .ioc file, the output is displayed in the UI (*Figure 315*).
 - 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 316*).
 - 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 is displayed (*Figure 317*).



Ne > STM32	2C011D6Yx Vintitled	Tools /		GEN	ERATE CODE
Pinout &	Configuration	Clock Configuration	Project Ma	anager	т
	Browse	Int_Issue/FLASH_Int_Issue.loc Project 2 Desktop/cuberr Browse	x_projects/build_error_CO/build	_error_C0.ioc 2 Project 1	nces only
PCC		& Middleware Project Settings			
GAD	CPN Flash(KB) RAM(KB) Io	Comparison i	n progress	1092007	Project
	Package Core	LOFP100 ARM Contar-M3	19.2	LOFP48 ARM Cortex-MD-	2

Figure 314. Starting the comparison process

Figure 315. The result of comparing two projects

Pinout & Configuration Project 1	Ited - Tools Clock Configuration	Project Manager	GENERATE CODE Tools		
Project 1		Project Manager	Tools		
() M	Project 2				
mpare Projects Spicuberrs_projects/PL					
Errente	ASH_int_issuel/FLASH_int_issue.ioc Setitoprovbersy Srowse	_projects/build_error_C0/build_error_C0.loc	Project 1 Project 2 Show differences only		
and a second	rals & Middleware Project Settings				
There is a second secon	Bitucture	Project 1	Project 2		
CPN	5TM32U675V/T		STM 32C092CCTH		
Flash(KB)	2548		258		
CAD RAM(KB)	796		30		
lo	82		45		
Package	LG##100		LOFP48		
Core	ARM Contee-M3	a	ARM Cortax-M0+		



STM DCLEWRY LIVERS		ile	Window	Help		99 F D X	O Q X ATT
Home STM3:		Untitled - Tools				GENERATE	
Pinout & Cor	nfiguration	Clock Configur	ation / Proje	RIF	Project /		Tools
	Browse	ectalFLASH int lasue/FLAS	Manuffrances and Residences	ens proets/FLASH_int_le	isue/FLASH_int_issue.loc	C Project 1 C Project 2 Show differences only	
PCC						,	
CAD		Same IOC Loaded 1	es project 1. There's no	point in continuing the comp OK	arison Choose another pro	× ject and try again	
				and and a second second			

Figure 316. Loading the same project



Pinout & C	Configuration	Clock Configural		Project Manager		Tools
	Project 5 Substanta projecto/PLASH Economi Use convert project	NU BRANT ADAL MU BRAN INC.	Project 2 reDrive - STMonoelectronice/D Encode	eathopFLASH_int_insue inc	Project 1 Project 2 Show differences	ente
PCC		& Middleware Project S	Settings			Prost 2
CAD						
CAD						

5.2.3 The output of the comparison of 2 projects

After starting a comparison of 2 projects, the following elements are added to the UI:

Target table

The Target table (*Figure 318*) provides a comparison of the microcontroller's 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.

STM02CubeMX UnitFiel	STM32C011D6Yx					
STM:0	File	Window	н	lelp 🕱		Sfox C Q ★
Home STM32	CO11D6Yx 🔪 Untitled	-Tools				GENERATE CODE
Pinout & C	Configuration	Clock Configure	ation	Project Manager		Tools
		int_issue/FLASH_int_issue.loc		icraelectronics/Desktop/Desktop.loc	Project	
PCC		& Middleware Project	Settings	Project 111	Ð	appears in different takes
CAD	CPN Flesh(KB) RAM(KB) Ig Package Core		575/32057500176 2048 786 82 0777100 MRM Contex-M33		510 32460 258 272 36 LGFP48 ARM Corte	SSCC14
	Cher MCU passesses		Sanda	the bit the Core parameter		

Figure 318. The structure of the Target table

Peripherals & Middleware table

It is a table (*Figure 319*) that 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 is presented in this continuous table with hundreds of lines. It is composed of the following columns:

- Category name
- IP name
- Mode
- Parameters settings
- Project 1
- Project 2

The "Peripherals & Middleware table includes a highlighting feature. This feature 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 for sorting in alphabetical order

Figure 319. The structure of the Peripherals & Middleware table

32 V beMX	File	Window	Help	Ŕ 🔶	1	F 🗗 🛛 🎧 🖓 🛰 🖍
me 🔪 STM32	C011D6Yx 🔰 Untitled - Tools	\rightarrow				GENERATE CODE
Pin	out & Configuration	Clos	k Configuration	Project Manager		Tools
	Project 1 STMicroelectronics/Desktop/cubernx Browse Use current project Target Peripherals & Mid		ht_issue.ioc C:Usersidaymarwa\OneC Browse	Drive - STMicroelectronics\Desktop\Desktop.ioc	Project 1 Project 2 Show differences only	
	Category Name	Ip Name	Mode	Parameters Settings	Project 1	Project 2
	Power and Thermal	PWR	Privilege attributes	PWR Privilege	Disable	Disable
PCC	Power and Thermal	PWR	Privilege attributes	Voltage detection and monitoring secure prote	. Disable	not set
	Power and Thermal	PWR	Privilege attributes	Backup domain secure protection	Disable	not set
	Power and Thermal	PWR	Privilege attributes	Privilege of PWR Secure Items	Disable	not set
	Power and Thermal	PWR	Privilege attributes	Pull-up/pull-down secure 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
	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, memory access fault	0	0
	System Core	NVIC		System service call via SWI instruction	0	0
	System Core	NVIC		Non maskable interrupt	0	0
	System Core	NVIC		Memory management fault	0	0
	System Core	NVIC		Undefined instruction or illegal state	0	0
	System Core	NVIC		Debug monitor	0	0
	System Core	NVIC		Pendable request for system service	0	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

Project Settings table

The table (Figure 320) contains:

- Information about the firmware package used for each MCU.
- Information about the toolchain used for building each project.

The table helps the users to:

- Know the software environment required for each project.
- Determine the necessary tools for ensuring project compatibility and facilitating migration.



It is composed of three columns:

- Settings:
 - CustomerFirmwarePackage
 - FirmewarePackage
 - ProjectStructure
 - TargetToolchain
- Project 1
- Project 2

Figure 320. The structure of the Project Settings table

STM32CubeMX Untitled: 5	TM32C011D6Yx					X
STM32	File	Window	Help	R 🔺	3	F 🗖 🗴 🗘 🍳 🔆 🟹
Home STM320	CO11D6Yx > Untitled - Tools >					GENERATE CODE
Pin	out & Configuration	Clock Con	figuration	Project Manager		Tools
Compare Projects	STMicroelectronicsi/Desktopicuberrx_project Browse Use current project Target Peripherals & Middlewa			- STMicroelectronics/Desktop/Desktop.loc	Project 1 Project 2 Show differences only	
	Settir			Project 1		Project 2
PCC	CustomerFirmwarePackage FirmwarePackage ProjectStructure TargetToolchain		STM32Cube FW_U5 V1.6.0 STM32CubeIDE		STM32Cube FW_H5 V1.3.0 EWARM V9.20	
CAD						
				Refresh Export		

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 321*).

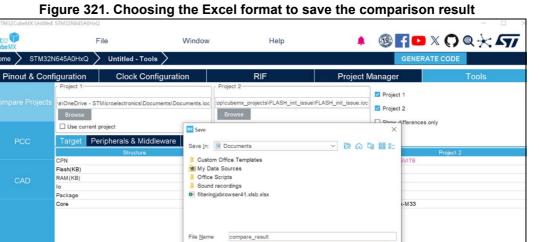
The available format:

- The result is exported into three sheets in an Excel format (Figure 322).
- Each sheet represents a table (Target, Peripherals & Middleware, Projects Settings).

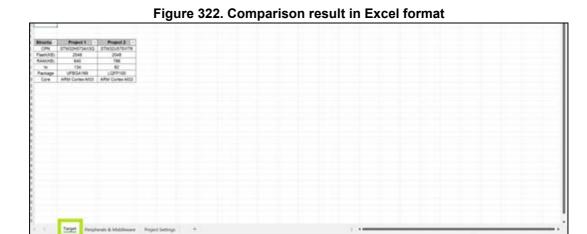
If the user wants to get only the differences in the exported file, they should click on "Show differences only".



.



Save Cancel



Files of Types Excel files



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 323*), 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 326*).

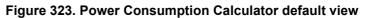
An additional option, the transition checker, is available for STM32L0, STM32L1, STM32L4, STM32L4+, STM32G0, STM32G4, STM32H7 and STM32WB series. When enabled, the transition checker detects invalid transitions within the currently configured sequence. It ensures that only possible transitions are proposed 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 323.

X STM32CubeMX	Untitled: STM32F469N	lHx							-		×
M32 TO ubeMX	File		Window	Help				<u>s</u>] 🖸 🄰	\star	5
	32F469NIHx >	Untitled - 1	Tools 🔪					GENERATE	CODE		
Pinout &	& Configuratio	on	Clock Cor	figuration	F	Project Manager		Т	ools		
				~ Power							
STM32F	469NIHx	<u> </u>		Step		Sequence					
Т _А 25°С	/ V _{DD} 3.3V	>	New Step	fi ti 🗍 🗑	5 ¢	C 🕯 🗐 🖆	1B 🚯				
Battery	Selection	~									
PCC	Select		Step M	/lode Vdd	Range/Scale	Memory CPU/Bus F.	Clock Cor	ntig Peripherals S	tep Current	Duratio	on
	ion Notes	· · · · ·									
Help		<u> </u>	Displa	av							
			Plot: All Steps	~ 0							
		_									
	Output										
STM32F4	Series	STM32F469	Lines /479	STM32F469NIF	Mcu Iv	Pa TFBGA216	ckage	Red None	quired Peripl	ierals	
01110214		0111021400	-10	0111021 400141	10	11 204210		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 324*).



CubeMX PCC: Battery Database	Management						
User Battery	-						
d User Battery 📋 Edit							
		Deffective Table					
		-Batteries Table			Databas		
	acity (mAh) Self Discha						
Alkaline (AA LR6) 2850.		1.5	1000.0	0.0	Default		
Alkaline AAA L 1250.		1.5	400.0	0.0	Default		
Alkaline(CLR14) 8350.		1.5	3000.0	0.0	Default		
Alkaline(DLR20) 20500		1.5	7500.0	0.0	Default		
Alkaline(9V 625.0		9.0	200.0	0.0	Default		
Li-MnO2(CF12 48.0	0.12	3.0	1.0	5.0	Default		
Li-MnO2(CR 6 125.0		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		F	Battery 29				
Li-SOCL2(A Capacity (ma A la \	-	0.0				
LI-SOCL2(U	,	-					
Li-SOCL2(D Self Discha	arge (%/month)	C	0.0				
Li-SOCL2(D Nominal Vo	oltage (V)	C	0.0				
Ni-Cd(AA11 Max Cont	Current (mA)	C	0.0				
	Current (mA)	c c	0.0				
Ni Cd/C200		12					
Ni-Cd(D440	e Current is not current	tly used in calculation					
Ni-Cd(F7000					OK Ca		
	30.0	1.4	100.0	0.0	Derault		
Ni-MH(AAA800) 000.0					Default		

Figure 324. 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 325*).

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 325. Step management functions

	Step							Sequence					
New Step	窗	D	ŝţ	31	5	2	2	1	Ø	В	₫D)	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 326*).



	ble All IPs Disable A wer/Memory RUN Scale1-High FLASH/REGON	All IPs	Enable	Peripherals Selection	Enabled Peripherals ADC1 ADC2 ADC3 BKPSRAM
Power Mode Power Scale Memory Fetch Type V _{DD}	RUN Scale1-High	All IPs	~	Peripherals Selection Peripherals	ADC1 ADC2 ADC3 BKPSRAM
Power Mode Power Scale Memory Fetch Type V _{DD}	RUN Scale1-High			Peripherals	ADC1 ADC2 ADC3 BKPSRAM
Power Scale Memory Fetch Type V _{DD}	Scale1-High				
Memory Fetch Type V _{DD}			~		
V _{DD}	FLASH/REGON				BusMatrix CAN1 CAN2 CRC
V _{DD}	Dibititeoon		~	- V ADC2	DAC DCMI DMA1 DMA2 DMA
	0.0		~	BKPSRAM	DSI ETH FMC GPIOA GPIOB
Voltage Course	3.3			- 🗸 BusMatrix	GPIOC GPIOD GPIOE GPIOF
vollage Source	Battery		~	- 🗸 CAN1	GPIOG GPIOH GPIOI GPIOJ
	Clocks			- 🗸 CAN2	GPIOK HASH 12C1 12C2 12C3
CPU Frequency 1	50 MHz		~		IWDG LTDC PVD/BOR PWR
Interpolation Ranges			~	✓ DAC	QUADSPI RNG RTC SAI1 SDI
User Choice (Uz)			-		SPI1 SPI2/I2S2 SPI3/I2S3 SPI4
User Choice (Hz)					SPI5 SPI6 SYS TIM1 TIM10
Clock Configuration	ISE PLL		~		TIM11 TIM12 TIM13 TIM14 TIM
Clock Source Frequency 4	MHz		~	Y DMA1	TIM3 TIM4 TIM5 TIM6 TIM7
Opti	onal Settings			- 🔽 1_Stream	TIM8 TIM9 UART4 UART5
Step Duration 1		ms	~	2_Streams	UART7 UART8 USART1 USART
Additional Consumption 0		mA	~	3_Streams	USART3 USART6 USB_OTG_FS
				4_Streams	USB_OTG_HS WWDG
	Results			5_Streams	
Step Consumption 103.65 m	A		_	- 7_Streams	
Without Peripherals 44 mA				- 8_Streams	
Peripherals Part 59.65 mA	eripherals Part 59.65 mA (A: 5.6 mA - D: 54.05 mA)				
Ta Max (°C) 95.08				- 🔽 1_Stream	
				/arnings	

Figure 326. 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 327*), or in fuchsia if at least one transition is invalid (fuchsia frame with description of invalid step highlighted in fuchsia, see *Figure 328*). In the latter case, the user can click the **Show log** button to find out how to solve the transition issue (see *Figure 329*).

Figure 327. Enabling the transition checker option on an already configured sequence - All transitions valid

				Seque					
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
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

Figure 328. Enabling the transition checker option on an already configured sequence - At least one transition invalid

01	Mode	Vdd	Dense /Osele	Memory	CPU/Bus Frea	Clock Config	Peripherals	Step Current	Duration
Step	Iviode	vaa	Range/Scale	iviemory	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
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 329. Transition checker option - Show log

Log for current sequence	_	×
======================================		
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		



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 330*).

	rigare oou. I		···· •	
🔤 New Step				×
Reset Step Settings	Enable All IPs Disable A	II IPs Enal	ble IPs from Pinout	
	Power/Memory		Peripherals Selection	Enabled Peripherals
Power Mode	RUN	\sim	Peripherals	ADC1 ADC2 ADC3 BKPSRAM
Power Scale	Scale1-High	\sim	— ✓ ADC1 — ✓ ADC2	BusMatrix CAN1 CAN2 CRC
Memory Fetch Type	FLASH/REGON	\sim	ADC2	DAC DCMI DMA1 DMA2
V _{DD}	3.3	\sim	- 🗹 BKPSRAM	DMA2D DSI ETH FMC
Voltage Source	Battery	\sim	─ ✓ BusMatrix ─ ✓ CAN1	GPIOA GPIOB GPIOC GPIOD
	Clocks			GPIOE GPIOF GPIOG GPIOH
CPU Frequency	User-defined	\sim	- CRC	GPIOI GPIOJ GPIOK HASH
Interpolation Ranges	150 MHz 168 MHz	\sim	✓ DAC	[2C1] [2C2] [2C3] [WDG] LTDC
User Choice (Hz)	16000000		— ✓ OUT1 — □ OUT1+OUT2	PVD/BOR PWR QUADSPI
Clock Configuration	HSE PLL	~		RNG RTC SAI1 SDIO SPI1
Clock Source Frequency	4 MHz	~	- V DCMI	SPI2/12S2 SPI3/12S3 SPI4 SPI5
0	ptional Settings		✓ DMA1	SPI6 SYS TIM1 TIM10 TIM11
Step Duration 1		ns 🗸	- 2_Streams	TIM12 (TIM13) (TIM14) (TIM2)
Additional Consumption)	nA 🗸	- 3_Streams	TIM3 TIM4 TIM5 TIM6 TIM7
			- 4_Streams	TIM8 (TIM9) (UART4) (UART5)
24 0	Results		— □ 5_Streams — □ 6_Streams	UART7 UART8 USART1
Step Consumption 103.6 Without Peripherals 44 m				USART2 USART3 USART6
	5 mA (A: 5.6 mA - D: 54.05 mA		8_Streams	USB OTG FS USB OTG HS
		<u></u>	V- DMA2	
Ta Max (°C) 95.08			- 🗹 1_Stream	
		W	arnings	
Available use cases: 1 Ma	ах: 60			Add Cancel

Figure 330. Interpolated power consumption



Importing pinout

Figure 331 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 332*).

The **Enable IPs from Pinout** button allows the user to automatically select the peripherals that have been configured in the **Pinout** view.

Pinout & Con	figuration	Clock Configuration
Options Q Categories A->Z	ADC	Mode and Configuration Mode
COMP2 CRC DAC DMA FATFS FREERTOS GPIO I2C1 I2C2	 IN2 IN3 IN4 IN5 IN6 	

Figure 331. 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 332. Power Consumption Calculator configuration window:
ADC enabled using import pinout

	Power/Memory		Peripherals Selection	Enabled Peripherals
Power Mode	RUN	~	Peripherals	
Power Scale	Scale2-Medium	~		
Memory Fetch Type	FLASH/REGON	~		
V _{DD}	3.3	~		
Voltage Source	Battery		BusMatrix	
Voltage Cource	Clocks			×
CPU Frequency	144 MHz	Enable IPs	from Pinout	
	144 MINZ		Peripherals Selection	Enabled Peripherals
Interpolation Ranges		Pe	ripherals	ADC1 GPIOA SYS
User Choice (Hz)		~	ADC2	
Clock Configuration	HSE PLL	~ -	ADC3	
Clock Source Frequency	4 MHz		BKPSRAM	
	Optional Settings		BusMatrix	<u> </u>
Step Duration		ms ~	2_Streams	
Additional Consumption		mA ~	3_Streams	
			4_Streams	
Step Consumption 40 m	Results		5_Streams	
			- 7_Streams	
Without Peripherals 40 m			8_Streams	
	(A: 0 nA - D: 0 nA)		V DMA2	
Ta 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 333*):

- 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.



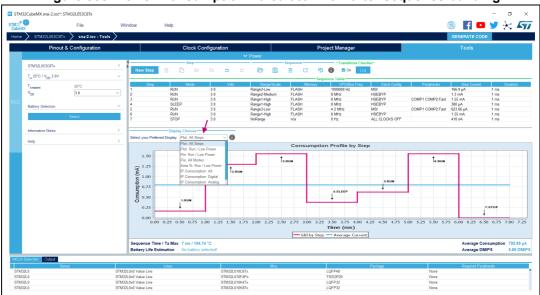
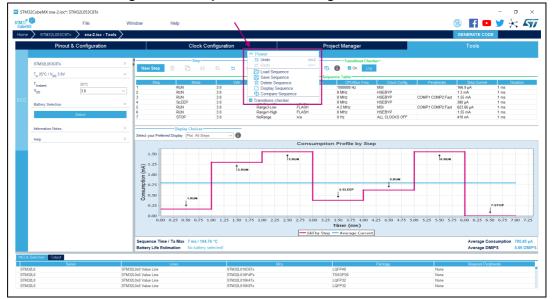


Figure 333. Power Consumption Calculator view after sequence building

Managing the whole sequence (load, save and compare)

From the power menu (see *Figure 334*), the current sequence can be saved, deleted or compared to a previously saved sequence that will be displayed in a dedicated popup window.



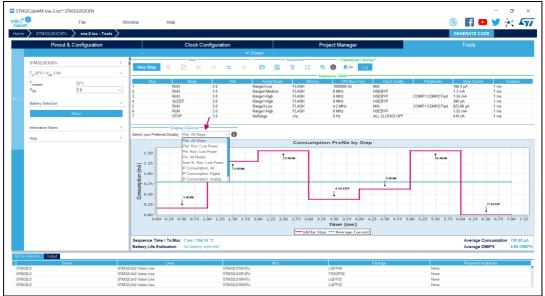


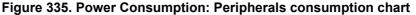


Managing the results charts and display options

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 335*).

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 336):

- 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 336. Description of the Results area

Results Summary			
Sequence Time / Ta Max	7 ms / 104.42 °C	Average Consumption	1.33 mA
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 337*).



						-
🔤 Edit Step					Enable	ed in pinout view $ imes$
Reset Step Settings	Enable All IPs Disab	le All IPs E	inak	ole IPs from Pinout		
	Power/Memory		-	Peripherals Sel	lection —	Enabled Peripherals
Power Mode	RUN	~	2	Peripherals		ADC1 GPIOA SYS ADC2
Power Scale	Scale1-High	~		- ADC1		ADC3 BKPSRAM BusMatrix
Memory Fetch Type	FLASH/REGON	~	2	- ADC3		CAN1 CAN2 CRC DAC DCMI
V _{DD}	3.3	~		BKPSRAM		DMA1 DMA2 DMA2D DSI
Voltage Source	Battery	~		─ ✓ BusMatrix ─ ✓ CAN1		ETH FMC GPIOB GPIOC
	Clocks		_			GPIOD GPIOE GPIOF GPIOG
CPU Frequency	User-defined	~	7	- 🗹 CRC		GPIOH GPIOI GPIOJ GPIOK
Interpolation Ranges	150 MHz 168 MHz	~	1	>- DAC		HASH (2C1) (2C2) (2C3) (WDG)
User Choice (Hz)	16000000		1	– ✓ DCMI > DMA1		LTDC PVD/BOR PWR
Clock Configuration	HSE PLL			> DMA2		QUADSPI RNG RTC SAI1
Clock Source Frequenc		-		— 🗹 DMA2D		SDIO SPI1 SPI2/I2S2 SPI3/I2S3
Clock Source Frequency		~		— 🗹 DSI — 🗸 ETH		SPI4 SPI5 SPI6 TIM1 TIM10
	Optional Settings			- FMC		(TIM11) (TIM12) (TIM13) (TIM14)
Step Duration	1	ms ~	4	— 🔽 GPIOA		(TIM2) (TIM3) (TIM4) (TIM5) (TIM6)
Additional Consumption	0	mA V	1	- 🗹 GPIOB		(TIM7) (TIM8) (TIM9) (UART4)
	Results			- V GPIOC		UARTS UART7 UART8
	0.04 mA		-			USART1 USART2 USART3
Without Peripherals 46.			4	- 🗹 GPIOF		USART6 USB_OTG_FS
	26 mA (A: 5.6 mA - D: 57.66	5 mA)	4	C GPIOG		USB_OTG_HS WWDG
Ta Max (°C) 94.	47		-	GPIOH		
			-Wa	arnings		
Available use cases: 0	Max: 60					OK Cancel

Figure 337. 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 338*). 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 339*). The user can then select a different SMPS model for the current sequence (see *Figure 340*, *Figure 341* and *Figure 342*)
- 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 343*).

Configuration of SMPS mode for each step (see *Figure 344*)
 If the SMPS checker is enabled, only the SMPS modes valid for the current step are proposed.



i igui	e 556. Selecting Sivil		
	STM32L496RGTxP		>
	T _A 25°C / V _{DD} 3.0V		\sim
	TAmbient	25°C	
	V _{DD}	3.0	\sim
PCC	Battery Selection		\sim
		Select	
	SMPS1_ST	Ļ	~
	Use SMPS	Help	
	Change		
	V _{IN(SMPS)}	3.0 V	
	V _{OUT(SMPS)}	1.1 V	
	OffCurrent	250 nA	
	QCurrent	500 nA	
	Efficiency	85 %	
	Туре	External	

Figure 338. Selecting SMPS for the current project



1.94		atabaoo	/ aamg m		incucio	-
STM32CubeMX PCC: SMP	S Database Managem	ent				×
User SMPS-						
Add User SMPS	Edit					
		CHIDG T				
Used North	0#0	SMPS T		T#	Tures	Detebury A
Used Name V _№ ⊗ SMPS1_ST 3.0	(SMPS) OffCurrent 250.0	QCurrent 500.0	V _{OUT(SMPS)}	Efficiency 85	Type External	Database 单 Default
ONI 01_01 0.0	230.0	500.0	1.1	05	External	Deladit
	Edit SMPS		×			
	User	SMPS				
	Name	SMPS2_User				
	Vin (V)	2.5				
	OffCurrent (nA)	10				
	Quiescent Current (nA)	10				
	V _{OUT(SMPS)}	1.2				
	Efficiency (%)	85				
	Туре	External				
		ок (Cancel			
	L					
					OK	Cancel

Figure 339. SMPS database - Adding new SMPS models

Figure 340. SMPS database - Selecting a different SMPS model

User SMPS								
Add User SMPS 🔟 Edit								
				SMP S T	able			
lsed	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
	SMPS1 ST	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 341. Current project configuration updated with new SMPS model



_	User	SMPS						
<u>A</u>	dd User SMPS	Ē	Edit					
					Table ———			
	Name	V _{IN(SMPS)}	OffCurrent	QCurrent	V _{OUT(SMPS)}	Efficiency	Туре	Database 单
sec	Name							
sec 9		2.5	10.0	10.0	1.2	85	External	User



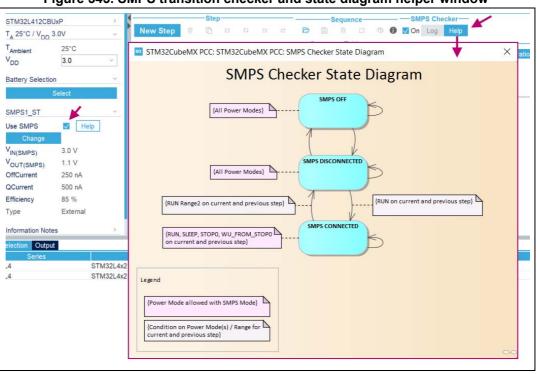


Figure 343. SMPS transition checker and state diagram helper window

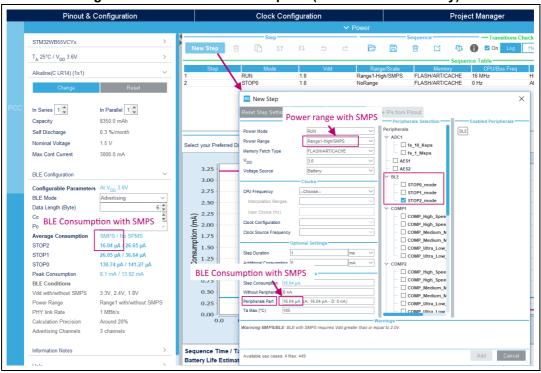


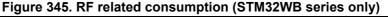
	.gg		•
🔤 New Step			×
Reset Step Settings	Enable All IPs Disable All IPs E	nable IPs from Pinout	
	ower/Memory	Peripherals Selection	Enabled Peripherals
Power Mode	RUN V	Peripherals	Enabled Forphoraio
Power Range	Range1-High 🗸	↓ ↓ ADC1	
Memory Fetch Type	FLASH V	_	
V _{DD}	3.0 ~	─ □ fs_1_Msps	
Voltage Source	Battery ~	─ □ fs_5_Msps	
_	SMPS	Y-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 V	☐ fs_5_Msps	
Interpolation Ranges		AHB_APB1_Bridge	
User Choice (Hz)		AHB_APB2_Bridge	
Clock Configuration	~	- CAN1	
Clock Source Frequency	V		
	otional Settings	→ DAC1	
Step Duration 1	ms V	- OUT1+OUT2-Buffer	
Additional Consumption 0		- OUT1+OUT2-Buffer	
	Results	- OUT1+OUT2-Buffe	
Step Consumption 10 nA		- OUT1-Buffer_OFF-	
Without Peripherals 10 nA		- OUT1-Buffer_ON-N	
Peripherals Part 0 nA	(A: 0 nA - D: 0 nA)		
		-Warnings	
Available use cases: 18 M	ax: 856		Add Cancel
, training and cards. To im	ans www		Vida

Figure 344. Configuring the SMPS mode for each step



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.





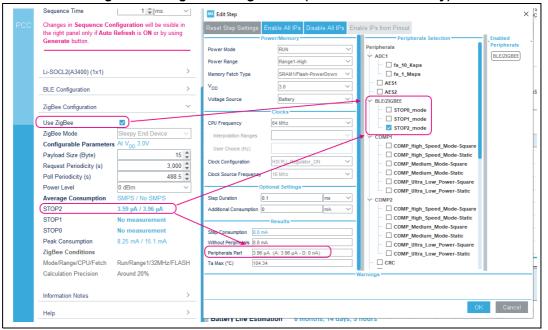
The Bluetooth Low-Energy mode can be selected from the left panel and configured to reflect the application relevant settings. For each new step enabling BLE, the peripheral consumption part is updated accordingly (see *Figure 346*). A similar approach is used for ZigBee (see *Figure 347*).



	Pinout & (Configuration	Clock Configuration
			✓ Power
	STM32WB55VCYx	>	Step-
	T _A 25°C / V _{DD} 3.6V	>	
	Alkaline(C LR14) (1x1)	~	Step Mode Vdd Rang
	Change	Reset	
PCC	In Series 1 🌲	In Parallel 1 韋	
	Capacity	8350.0 mAh	
	Self Discharge	0.3 %/month	Display Choices
	Nominal Voltage	1.5 V	Select your Preferred Display Plot: All Steps
	Max Cont Current	3000.0 mA	
	BLE Configuration	~	Select a BLE mode
	Configurable Parameters	At V _{DD} 3.6V	
	BLE Mode	Advertising \sim	Advertising ~
	Data Length (Byte)	6 🌲	Choose
	Connection Interval (ms)	1,000 🌲	Advertising Advertising Non Connectable
	Power Level	0 dBm ∨	Connected Master
	Average Consumption	SMPS / No SPMS	Connected Slave
	STOP2	16.04 μA / 26.65 μA	
	STOP1 STOP0	26.05 μΑ / 36.64 μΑ 130.74 μΑ / 141.27 μΑ	Configure the parameters
	Peak Consumption	130.74 μA / 141.27 μA 8.1 mA / 13.92 mA	for the selected mode
	BLE Conditions	0.1 mA7 13.52 mA	
	DEE CONULIONS		

Figure 346. RF Bluetooth Low-Energy mode configuration (STM32WB series only)



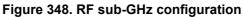




5.3.8 Sub-GHz support (STM32WL series only)

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 348*).

		L proiz pio prorango rei priz pizz_o
STOP2 Consumption 8	85 nA	I Edit Step
Low Power Mode	STOP2 ~	Reset Step Settings Enable All IPs Disable All IPs Enable IPs from Pinout
RUN Step 0.1 ms / ST	OP2 Step 0.9 ms	Power/Memory Peripherals Selection Enabled
0 20 4	0 60 80 100	Power Mode RUN CLUBCHZ
Sequence Time	1 🛊 ms 🗸 🗸	Power Range Range1-Medium SMPS-ON - LPUART1
Auto Refreeb ON auto	matically update sequence table.	Memory Fetch Type
display and results in t		
		Voltage Source Battery Voltage Source Spin
		Clocks SPI2/12S2
Li-SOCL2(AAA700) (1x	1) >	CPU Brequency = 🗌 sRAM1
		Interpolation Ranges
SUBGHZ Configuration		User Choice (Hz)
 Configurable Param 	eters at 25 °C / 3.0 V	Clock Configuration
Frequency Band	High ~	
SMPS	On V	
•Tx Parameters		Optional Settings - Imt2
Power Amplifier	Low Power V	
Output Power	10 dBm 🗸 🗸	Additional Consumption 0 mA
Optimal Settings		Results — USART1
Tx Current	17.5 mA	Step Consumption 5.1 mA
		Without Peripherals 0 nA Peripherals Part 5.1 mA - D: 0 nA)
Modulation	FSK V	Ta Max (°C) 125
Rx Boosted Rx Current	✓ 5.1 mA	
TX Current	5. T 10A	· · · · · · · · · · · · · · · · · · ·
Information Notes	>	
Help	>	OK Cance
Tielp		



5.3.9 Example feature (STM32MPUs and STM32H7 dual-core only)

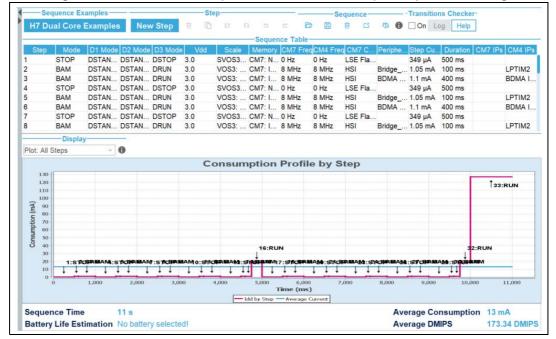
Under the section Sequence Examples, the PCC tool allows to access examples: each of them comes with an explanatory slide set and a ready-made sequence to load in PCC (see *Figure 349*).



× 😭	X Untitled: STM32H745IGKx				
2 To and a second se	File	Wind	ow Help		🎯 🖪 🖻 🎽 🛧 /
ome > STI	M32H745IGKx 〉 Untit	tled - Tools >			GENERATE CODE
Pinout	& Configuration	Clo	ck Configuration	Project Manager	Tools
			~ Power		
STM3	2H745IGKx	>Se	quence Examples	Step	Sequence
Т _Ј 25°	C / V _{DD} 3.0V	U H7 Di	al Core Examples New	Step 🕆 🗈 🗈 🖻	🖻 🛱 🕆 🖓 🕦
TJunct	ion (°C) 25	~ Step	Mart STM32CubeMX PCC: H7	Dual Core Examples	×
V _{DD}	3.0	~ Otep	WOO	H7 Dual Core Examples	
			Load Example 1	Example 1 Presentation	I2C transmission + data acquisition
Battery	y Selection		Load Example 2 Load Example 3	Example 2 Presentation Example 3 Presentation	I2C transmission + data acquisition I2C transmission
	Select			Benefits of H7 Dual Core	Short Presentation
Inform	ation Notes	> Plot: All s	Steps	Example 1	
Help		>		Low-power application of	example 3 🚥
				s example completes the example 2 by adding a play phase.	new data
			• The STI	purpose is to highlight the smart power manage avg2h7 Series (STM32H7x5 or STM32H7x7 Lines ver domains.	ment of s) using three
			• The STI	purpose is to highlight the smart power manager M32H7 Series (STM32H7x5 or STM32H7x7 Lines	ment of s) using three #Toutenes
			• The STI pow	yurpose is to highlight the smart power manages W32H7 Series (STM32H7x5 or STM32H7x7 Line: er domains.	s) using three

Figure 349. Power Consumption Calculator – Example set

Clicking "Load Example N" loads the sequence corresponding to example N (see *Figure 350*).





Clicking "Example N Presentation" displays the explanations for that example.



UM1718 Rev 46

The example can be changed anytime: the new sequence can be added to the current sequence, or replace it (see *Figure 351*).

Figure 351. Power Consumption Calculator – Example sequence new selection

- 5	equence Examples				-Step-						Seque	ence-		-	-Transi	itions (hecke
H7 C	Oual Core Examp	les Ne	w Step	8	Ō	Lf Il	5	2	6		Ø	C	5 1 3	0	On	Log	Help
Step	Mode D1	X STM32Cub	eMX PCC: H	17 Dual	Core Exa	mples										×	ration
Step	STOP DST					-H7 I	Jual Co	re Exan	nnles-								State in case of the local division of the l
2		ad Example	e 1		F	xample 1			ipico	13	2C tr	ansmis	sion +	data	a acquis	sition	ms
3		ad Example				xample 2									a acquis		ms
4	STOP DST	ad Example	e 3			xample 3						ansmis					ms
4	BAN DST	the second pro-				enefits of					Short	Preser	tation				
				×	_			nole 2-									ms
varning: load	Example 2 with existing	steps:		^			200										ms ms
Before load	ling Example 2, keep	or remove	existing ste	eps.									-				ms
Kanan C		ps Car			LOW	-pow	er ap	plica	tion	exa	am	pie 4		1			1115
Keep S	teps Remove Ste	ps Car	icei														
Plot: Al	Steps			_													
-			• T	his exa	mple co	mpletes th	ne examp	ole 1 by a	adding	a new	data						
1000			а	cquisiti	ion phase	Э.											
13				-		o highlight	-										
11						(STM32						ree					
10				ower d		. (
	3				omains.												
(An a					omains.												
(mA) no	0 - 0 -				omains.												
(hm) notion (mA)	0 - 0 - 0 -				omains.												
sumption (mA)	D =		57		omains.												
m) uption (m	0		£77		omains.		() tais (it										
Consumption (mA)	0		F1		omains.												
3	0 0 0 0 1:5 ř cijæs				omains.		-Slides	Show-									-
31 22 14			< Previous		omains.		-Slides						Ne	xt >			- 1
31 22 14	0 0 0 0 1:5 ř cijæs	2,000		<u>.</u>	omains. 4,00	0	-Slides	s Show- 1 of 9 6	,000		7,000		Ne 8,000	000	9,0	000	9 93 4
31 22 14	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		< Previous	<u>.</u>			Slide	s Show- 1 of 9	ms)		7,000			000	9,0	200	3 9 933 10,0

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.



UM1718

DDR Test suite testing and tuning features are available from the Tools view.

The DDR suite relies on two important concepts:

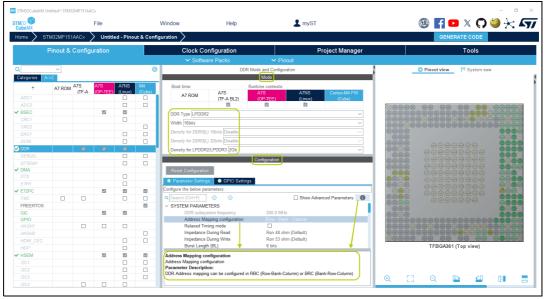
- the DDR timings as key inputs for the configuration of the DDR Controller and PHY
- the tuning of DDR signals to compensate board design imperfections.

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 352* 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.

	Figure 3	53. DDR3 confi	guration	
	DDF	R Mode and Configu	ration	
		Mode		
Boot time:		Runtime contexts:		
A7 ROM	A7S	A7S	A7NS	Cortex-M4 FW
	(TF-A BL2)	(OP-TEE)	(Linux)	(Cube)
		~		
DDR Type DDR3 / [DDR3L			~
Width 16bits				~
Density for DDR3(<mark>I</mark> .)	16bits 1Gb			~
		Configuration		
Reset Configuration				
🛛 🛛 Parameter Settin 🔉	s 🛛 🥺 Registers	OPIO Settings		
Configure the below para	ameters :			
Q Search (Ctrl+F)	© ()		Show Adva	anced Parameters 🛛 🕦
✓ SYSTEM PARAME	TERS			
· · · · · · · · · · · · · · · · · · ·	tem frequency	400.0 MH		
Speed Bin G			6G / 8-8-8	
Impedance D	-		nm / ODT = 80 ohr	
Impedance D			nm / ODT = 60 ohr	n (Default)
Relaxed Tim	pping configuration	Row - Bar	nk - Column	
	case over 85°C sup			
Burst Length		8		
Speed Bin Grade Speed Bin Grade				
Parameter Description	on:			
JEDEC Standard for D				
Set value of each timin	-		Bin Grade' selecti	on.
Tick 'Show Advanced F	Parameters' to see th	ie involved timings.		

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 354.

32 VT DeMX	File	Window	Help	L myST	<u>(19</u>)	🗗 🔼 X 🖓 🍏 🔆 🖊
	57CACx Vintitled - Tools	>				GENERATE CODE
Pinout	& Configuration	Clock C	Configuration	Project Manager		Tools
PCC Con	DR Interactive Port sele M3 Connect DDR utili DDR utili	ction	Loading 🖲 💿	Tarç	et Information	
	Not Connected Select	file stm32mp157c-ev1-ba	asic.stm32mp157c-ev1-basic.			
R Test Suite	arget DDR Tests					
		STM22Cube		- 1	×	
		STM 32Cube			×	
		STM 32Cube	Programmer I 32CubeProgrammer has been ma		× E	
		STM 32Cube ⊘ The STM	Programmer //32CubeProgrammer has been ma File Path			
		STM 32Cube	Programmer //32CubeProgrammer has been ma File Path	nually selected		
		STM 32Cube The STM Executable Browse (Version	Programmer //32CubeProgrammer has been ma File Path	nually selected		
		STM 32Cube The STM Executable Browse (Version	Programmer	nually selected		
		STM 32Cube The STM Executable Browse (Version	Programmer	nually selected STM32Cube/STM32CubeProgrammer/binISTM32_Programmer_		

Figure 354. 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 355).

Figure 355. DDR Suite - Target connected

III STM32CubeMX Unti	tled*: STM32MP157CACx						– ø ×
STM32 CubeMX	File	Window	Help	L myST		📑 🖻 🗶 💭	🍑 🔆 🏹
	32MP157CACx $>$ Untitled - Tool	• >				GENERATE CODE	
Pi	nout & Configuration	Clock C	Configuration	Project Manager		Tools	
PCC	COM3 V Disconnect	USB UAR	Loading () () T Select COM ~) asic.stm32mp157c-ev1-basic.	Config name: DDR3-1066/888 bin G 2 DDR Size: 8 GBits DDR Frequency: 533.0MHz	Target Information ()		
DDR Test Suite				-Boot SPL will then be re-started and f ectly on the configuration you have edit			

Output/Log messages

STM32CubeMX outputs DDR suite related activity logs (see *Figure 356*) and interactive protocol communication logs (see *Figure 357*). They are displayed by enabling outputs from the Window menu.

Figure 356. DDR activity logs

reating: STM32MP151CACx	
initializing: STM32MP151CACx	
Creating: STM32MP157CACx	
Initializing: STM32MP157CACx	
YYSRAM succesfuly loaded with: C:\Data\stm32\ddr\eval_board\Revc\same_port\02_07_2019\u-boot-spl.stm32-stm32mp157c-ev1-basic	
DDR Test Suite connected to target board	
arget board configuration name: DDR3-1066/888 bin G 2x4Gb 533MHz v1.45	
arget board DDR size: 8 GBits	
arget board DDR frequency: 533.0MHz	

Figure 357. DDR interactive logs

MCUs Se	lection	Output	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.

STM32CubeM	X Untitled": STM32MP157CACx						- 0 ×
STM32 CubeMX	File	Window	Help	L myST	(19)	f 🕨 X 🗘	🍏 🔆 🏹
Home >	STM32MP157CACx > Untitled - To	ols >				GENERATE CODE	
	Pinout & Configuration	Clock C	Configuration	Project Manager		Tools	
PCC	COM3 ~ Disconnect	t selection	Loading 🖲 🔕	Config name: DDR3-1066/888 bin G 2x4Gb v1.45 DDR Size: 8 GBits	Target Information ()		
CAD		select file stm32mp157c-ev1-ba	isic.stm32mp157c-ev1-basic.	DDR Frequency: 400.0MHz			
DDR Test S	Use the Load Registers' comh	lue.You can then conduct tes	t and tuning operations dir Load Registers Progress	I-Boot SPL will then be re-started and DDR C ectly on the configuration you have edited in th			
		6	Loading DDR Registers to 27 out of 96 registers Cancel	target			

Figure 358. 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 359).



132 TO IbeMX	File	Window	Help	💄 myST	🐵 F 🖻 🗴 🖓 🌽 🖌
ome 🔪 ST	FM32MP157CACx > Untitle	rd - Tools 🔪			GENERATE CODE
	Pinout & Configuration	Clock	Configuration	Project Manager	Tools
	DDR Interactive		I Loading 🔀 🚳	т	arget Information 🕕
	Target DDR Tests				
	ld(s) Test type(s)	Test name(s)	Execution		
	0 All	All	_	Run t	est 🕕
	1 Basic	Simple DataBus			
	2 Basic	DataBusWalking0		Parameter(s)	Value(s)
	3 Basic	DataBusWalking1	Address		0x C0000000
	4 Basic	AddressBus MemDevice	Loop		1
	5 Intensive 6 Intensive	SimultaneousSwitchingOutput	Loop		
	7 Intensive	Noise	Verdict		
	8 Intensive	NoiseBurst	verdict		
	o intensive	Random	_		
	10 Intensive with Stress	FrequencySelectivePattern		ltem(s)	Info
	Conditions	requirey delectiver allern	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		1110	
	Purpose	Verifies each data bus signal can be	driven low		
	Test Sequence			111111111111111111111111111110' at given addr	ess, read back given address and check the pattern is OK. Write then
	root ooquurub			en address and check the pattern is OK, and so on.	cos, read back ground and cost and check the patient is one time then
	Param1				ddress should be located in the DDR memory region [DDR base addre
	Param2		verdict. Same test is repeated [] our) times. Verdict OK if all tests are OK, KO otherwis	e.
	Restriction Limitations			you will need to perform the data bus test at multiple	
	Interest	Very basic test to be executed first to			,
	Failure Type	Catastrophic failure			

Figure 359. 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 360*). The test history is available in the output and Logs panels (see *Figure 361*).



Figure 360. DDR test suite results

Execution	
	Run test
Parameter(s)	Value(s)
Address	0xC0000000
Loop	1
Verdict	
ltem(s)	Info
Address	0xC0000000
Loop(s)	1
Result	Pass
Result details	no error for 1 loops

Figure 361. DDR tests history

MCUs Se	ection	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:DataBusWalking1
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	ooard d	configura	tion name: DDR3-1066/888 bin G 2x4Gb 400.0.0.0.0MHz v1.45
Target 1	board l	DDR size:	8 GBits
Target 1	board l	DDR frequ	lency: 400.0MHz
Current	config	guration	DDR registers loaded to the target board
DDR test	c #2 (1	DataBusWa	(lking0) triggered with parameters: [loop] l [addr] 0xC0000000
DDR test	c #3 (1	DataBusWa	(lkingl) triggered with parameters: [loop] 1 [addr] 0xC0000000
DDR test	t #4 (1	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.



UM1718 Rev 46

The MMT key features are:

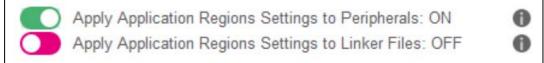
- Memory map display
- Application regions management
- Linker file generation

MMT interacts with peripherals starting from the moment the user enters its interface:

- Checks their settings
- Updates other peripherals involved in memory map configuration

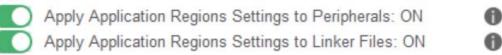
The peripherals are updated only when the first toggle button is ON.

Figure 362. Regions settings to peripherals ON



MMT updates the linker scripts only when the second toggle button is ON.

Figure 363. Regions settings to linker files ON



The applicative regions are saved into the user project even if the first toggle button is OFF.

Figure 364. Regions settings to peripherals OFF



O	v
	-

5.5.1 STM32U5, STM32H5, STM32WBA, and STM32WBAM with TrustZone activated

Feature: MMT usage, pinout, and configuration user interface

When the first toggle button is ON (see *Figure 362*), 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.



SAU Mode and Configuration								
	Mode							
Runtime contexts:								
Cortex-M33 secure		Cortex-M33 non secure						
۲								
✓ Enable SAU								
	Configuration							
Reset Configuration								
Parameter Settings								
onfigure the below parameters :								
Search (Ctrl+F) ③ ③								
Region 0								
Enable this region	Yes							
Start Address	0×08000000							
Block Size	0×04000000							
Secure Attribute	Non-Secure							
✓ Region 1								
Enable this region	Yes							
Start Address	0x0C1FE000							
Block Size	0x23E02000							
Secure Attribute	Non-Secure							
Region 2								
Enable this region	Yes							
Start Address	0×94000000							
Block Size	0×04000000							
Secure Attribute	Non-Secure							
Region 3								
Enable this region	No							
✓ Region 4								
Enable this region	No							
Region 5								
Enable this region	No							
Region 6								
Enable this region	No							
Region 7								
Enable this region	No							

Figure 365. 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
 Image: Constraint of the setting of the s

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.



a

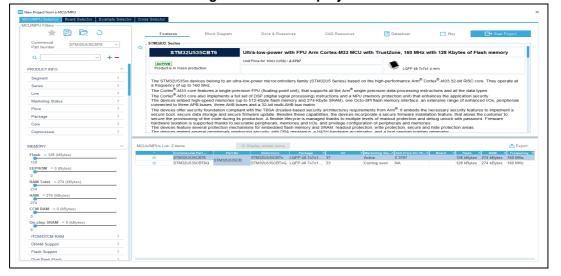
	File Window I	Halp 💄 myST		🐵 🖪 😐 🏏 🗛 🔆
> stina	2U5A82JTx0 🔪 mmt2.los - Tools 🔪 LPBAM Scenario & Co	efiguration >	GENERATE CODE	CHECK LPBAM DESIGN
	Pinout & Configuration	Clock Configuration	Project Manager	Tools
	Code Generation Configuration	Memory viewed by ARM Cotex-M33	Application Regions	
-	Apply application Regions settings to peripherals: ON		Appretation Regions	
	Apply Application Regions settings to linker files: ON		NA	
y ement	Search an Application Region		MyExternalRAM (PS)	
-	Search an Application Region			
	Show selected Application Region	Castonocon External RAM (OCTOBPI1) Remapped (64 MB)	MyRemappedRAM (3) Reserved Alea Repon	
		Cwt0000000 FMC Bank3 (256 MB)		
		0w1000000 Octo@Pt2 (256 MB)	NA	
		Caska Soccoce FMC (Bank 1 (240 MID)	NA	0
		cautoococococococococococococococococococo	MyExternaFlash (S)	Display Settings Reset View
:c		ox5co3eccoReserved		Hide security beside regions names
		0x50034400 Backup SRAM (5) (2 KB)		Hide Reserved Allased regions names
		0x40030CooReserved		8
		0x40034400 Backup SRAM (NS) (2 KB)		
		Cw38004000 Reserved		8
		(w18000000 SRAM4 Smart Run Domain (S) (16 KB)		
		(witi2?toro Reserved		A
0		(WEELLOOD) SRAMS (5) (832 KB)	MyThmad00Heap (NS) Reserved Alias Region	A
~		(with Databased SRAM3 (5) (832 KB)		
		083000000	MyBuffer (NS) Reserved Alias Region	
		SRAM2 (S) (S4 KB)	RAM (NS) Reserved Alias Region	
		6x30000000 SRAM1 (3) (769 K8)		
			RAM (5)	
		Cw28004000 Reserved		÷
		0x21000000 SRAMI Smart Run Domain (NS) (16 KB)		
		owarazhooce Reserved		ê .
		0821 LA0000 SRAMS (NS) (832 KB)	MyThread/Heap (NS)	Logend
		0x20020000 SRAM3 (NS) (832 K8)		
			MyBuffer (NS) Reserved Alias Region	Region allowing different types of security.
		SRAM2 (NS) (54 KB)	RAM (NS)	Secure region (3).
		(#2000000 SRAM1 (NS) (768 KB)	RAM (S) Reserved Alias Region	A .
		Cwot270000 Reserved		Non Secure region (%S).
		Cw01120000 SRAMS Code (S) (832 NE)	MyThread00Heap (NS) Reserved Alias Region	Non Secure Callable region (NSC).
		CALEDOCCO SRAMI Code (5) (822 KB)		Region accessible by Secure and Non-Secure
			MyOuffer (NS) Reserved Alias Region	A.
		SRAN2 Code (5) (64 KB)	RAM (IVS) Reserved Alias Region	Reserved region.
		faxtEscore faxtEscore SRAM1 Code (S) (768 KB)	RAM (3) Reserved Alias Region	A new application region can be added here.
			KONE (3) RESERVED AND REGION	
		0x0C400000Reserved		NIA IP Configuration is expected. See tooltip info for furthe
		CestC200000 Flash Bank2 memory (S) (2 MB)	FLASH (NS) Reserved Alias Region	
		Flash Bank1 memory (S) (2 MB)	FLASH_NSC (NSC)	
		6w0000000	FLASH (S)	
		oxxiaz hooce Reserved		

Figure 366. MMT view

5.5.2 An end-to-end usage example

Choose a supported MCU (STM32U585x in this example).

Figure 367. Start a project





Press the "Start Project" button, and then choose the "with TrustZone activated ?" option.

Figure 368. Use TrustZone

TrustZone feature available \times	
Do you want to create a new project : O without TrustZone activated ?	
• with TrustZone activated ?	
OK	

Choose the "Tools" tab followed by the "Memory Management" option to display the Memory Management Tool (see *Figure 369*).

	File Window	Help	💄 myST		🚳 🖪 🗖 🎽 🖓 🔆
∑ stiv	132U535CBTx 🔰 Untitled - Tools 🔪 LPBAW Scenario :	& Configuration	\rangle	GENERATE CODE	CHECK LPBAM DESIGN
<i>.</i>	Pinout & Configuration		Clock Configuration	Project Manager	Tools
	Code Generation Configuration	~	Memory viewed by ARM Contex-M33	Application Regions	+ 1
		0	No. Reserved		â
y	Apply Application Regions settings to linker files: OFF	0 000000	0ctxSP1 (256 MB)	NA	0
ement	Search an Application Region	>	con Received		â
		> 0x50036	Backup SRAM (S) (2 KB)		
	Show selected Application Region		coo Reserved		â
			Backup SRAM (NS) (2 KB)		
		0x38004			â
			SRAM4 Smart Run Domain (S) (16 KB)		Display Settings Reset View
			our Reserved		Hide security beside regions names
С		0x30040	500 505 SRAM2 (5) (64 KB)	RAM (NS) Reserved Alias Region	A
		0230033	SRAM1 (S) (192 KB)	RAM (S)	Hide Reserved Aliased regions names
					A
		0x28004	₀₀₀ Reserved ₀₀₀ SRAM4 Smart Run Domain (NS) (16 KB)		
			oco Reserved 	RAM (NS)	
		0x20033	000 ORAMIC (FILS) (192 KB)	RAM (S) Reserved Alias Region	
D				nair (3) meenver aliso migen	
		0x08141	no Reserved	All 400 December 41 December 41	
		(m(E13)	505 SRAM2 Code (S) (64 KB)	RAM (NS) Reserved Alias Region	
			558AM1 Code (S) (192 KB)	RAM (S) Reserved Alias Region	
		0x00023	200 ^{Reserved}		
				FLASH (NS) Reserved Alas Region	
			Flash Bank1 memory (S) (128 KB)	FLASH_NSC (NSC)	
		0x0000		FLASH (S)	
		0008.047	200 ^{Reserved}		Legend
		00004.033		RAM (NS) Reserved Alias Region	
		0008.007	000 SRAM1 Code (NS) (192 KB)	RAM (S) Reserved Alias Region	Region allowing different types of security.
		0x06627	ou Reserved		B Secure region (S).
				FLASH (NS)	Non Secure region (NS).
			Flash Bank1 memory (NS) (128 KB)	FLASH_NSC (S) Reserved Alias Region	ů,
		0x06666		FLASH (S) Reserved Alias Region	Non Secure Callable region (NSC).
		0000000	Reserved		Region accessible by Secure and Non Secure.
					Reserved region.
					A new application region can be added here.
					NA IP Configuration is expected. See tooltip into for furt
					The second secon

Figure 369. 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).

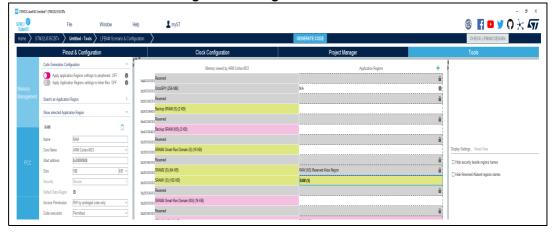


Figure 370. Region information

Code generation configuration

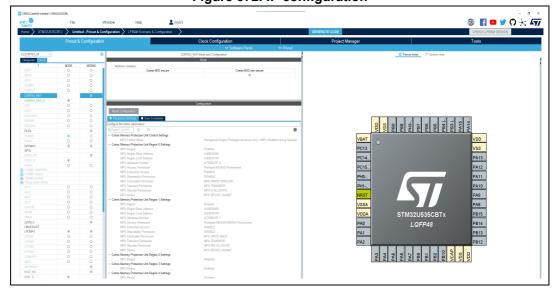
The application regions settings can be applied to peripherals on the left of the screen. The concerned peripherals are shown on the associated tooltip. This can impact their availability on the pinout screen configuration.

STM32Cube/VX Unit	titled": STM32U535CBTi						- 0
TM32 CubeNX	Fi	e Window		Help	💄 myST		🚳 👖 🗅 🎽 🗘 🛧 /
Home 🔪 STN	W32U535CBTx 🔪 L	Intitled - Tools 🔪 LPBAM :	Scenario & Conf	iguration	>	GENERATE CODE	CHECK LPBAM DESIGN
	Pino	ut & Configuration			Clock Configuration	Project Manager	Tools
	Code Generation Confi	pration	×		Memory viewed by ARM Cartex-N33	Application Regions	+ 1
		on Regions settings to peripherals: on Regions settings to linker files		(12)(10)	10 Reserved		8
/emory	- Abil Abicat	on regione assessé la IRAR IRA.	This	toggle button in	Scales whether the Application Regions settings are applied PU), FLASH, GTZC (MPCBBs), SAU	N/4	0
lanagement	Search an Application	Region	Whe		ton is CN, these peripherals become (partially) readonly		Ĥ
	Show selected Applica	tion Region	¥		tion pere- Watching on the last terms		
	RAM		n		₀₀ Reserved ₀₀ Backup SRAM (NS) (2 NB)		
	Name	RAM			₁₀ desenved		A
	Core Name	ARM Cortex-M33	v		oo oo SRAMA Smart Run Domain (S) (15 KB)		Display Settings Reset View
PCC	Start address	0.300000			ov Reserved		Hde security beside regions names
	Size	192	KB ~	0x300303	00 SRAM2 (S) (64 KB)		Hide Reserved Aliased regions names
	Security	Secure	v		00 SRAM1 (S) (192 KB)	RAM	
	Default Data Region	2		0x280043	00Reserved		
	Access Permission	RW by privileged code only	×		₂₀ SRAMA Smart Run Domain (NS) (16 KB)		A.
	Code execution	Permitted	~		₁₀ Reserved ₁₀ SRAM2 (NS) (64 KB)	RAM	
CAD	Shareability Cacheability	Non-Shareable Write-Back Read Write Allocate	Y		00 STAM1 (NS) (192 KB)		

Figure 371. 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* 373.

M32	File	Wi	ndow Help	L myST			
ome 🔪 STM32U535	CBTx 🔰 Untitle	ed - Pinout & Conf	iguration > LPBAM Scenario &	Configuration			
	Pinout &	Configuration			Clock Configurati	ion	
					~	Software Packs	🗸 Pinout
CORTEX_M ~		۲		GTZC_S	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				Configuration		
	0	0			Conliguration		
	0	0	Reset Configuration				
DCACHE1	0	0	⊘ TrustZone Illegal	access Controller	📀 User Ci	onstants 📀	NVIC Settings
	0	0		oller - Memory Protection Control		Block-Based Memory Pr	
FDCAN1	0	0		stZone Security Controller - Secur		TrustZone Security Controller - Pr	
FILEX		۲	Configure the below parameters :				
FLASH	۲	0	Q Search (Ctrl+F) ③ ④				6
FMAC	0	0	MPCWM1 (OCTOSPI1)				
GPDMA1	۲	۲	Configure Memory		from full Secured		
GPIO			Area 1 Start Address		0x90000000		
GTZC_NS		۲	Area 1 Size		0×0		
GTZC_S			Area 1 Secure Attribute		secured		
HASH Global TrustZon	e® Controller	0	Area 1 Privilege Attribute		privileged		
I-CUBE I-CUBE			Lock the configuration of MP	CWM1 Area 1 until next reset	MPCWM1 Area 1 is	not locked	
I-CUBE: Not available: I-CUBE: IP under MMT C	antrol		Status of MPCWM1 Area 1		disabled		
II UNDER WINT G	umentation (Ctrl+d)		Area 2 Start Address		0×90000000		
12C1	<u> </u>	0	Area 2 Size Area 2 Secure Attribute		0×0		
	0	0	Area 2 Secure Attribute Area 2 Privilege Attribute		secured		
	0	0	Lock the configuration of MP	CWM1 Area 2 until next reset	MPCWM1 Area 2 is	not locked	
I2C4	0	0	Status of MPCWM1 Area 2		disabled		
ICACHE	0	0	> MPCWM4 (BKPSRAM)				
	0	0	. /				
IRTIM	0	0					

Figure 373. IP under control



UM1718 Rev 46

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.



ubeNX	Fi	io V	ńndow	Help	1 myst			🌕 🛞 🛐 🗖 🎽 🖉
ame 🔪 STN	ABQUEBECETX 🔪 I	Untitled - Tools 🔪 🗆	PBAM Scenario & C	antiguration	\rangle	GENERATE CODE		CHECK LPBAM DESIGN
	Pinc	out & Configuration			Clock Configuration	Project Manage	r	Tools
	Code Generation Conf	iguiation	~	1	Memory viewed by ARM Contex-M33		Application Regions	1
		ion Regions settings to perio		(10.000)	Reserved		0	
етоту	Apply Applicat	ion Regions settings to link	mas on 🙂	0000000	OctoSP11 (266 MD)	NA	6	
anagement	Search an Application	Region	>		Reserved		ê	
	Show selected Applica	ation Region	~	Cu:50036	400 Backup SRAM (S) (2 KB)			
	DAM		•	0x40030	Reserved		ê ;	
	KOM				400 Backup SRAM (NS) (2 KB)			
	Name	RAM		0x38034	Reserved		ê ;	
	Core Name	ARM Cortex-M33		0x38000	SRAWA Smart Run Domain (S) (16 KB)			Display Settings Reset View
PCC	Start address	0x30000000		0x30140	Reserved		£	Hide security beside regions names
	Skze	192	KB ~	0x30130	SR4W2 (S) (64 KB)		ê ;	Vide Reserved Alased regions names
	Security	Secure		0420000	SRAM1 (S) (192 KB)	RAM		
	Default Data Region	8			000 Reserved		â	
	Access Pernission	RW by privileged code a	w v	0421000	SRAWA Smart Run Domain (NS) (16 KB)			
	Code execution	Permitted			ocoReserved		ê :	
	Shareability	Non Shareable			SRAM2 (NS) (64 KB)	RAM		1

Configuring an external memory

This example uses the FMC. Go to the Pinout & Configuration window (see *Figure 375*) and enable the IP.



32 beMX		File	Window	Help	💄 myST		
<u> </u>	J5A9ZJTx0	ຊ 🔪 Untitl	ed - Pinout & Configuration	> LPBAM Scenario &	Configuration >		
		Pinout &	Configuration			Clock Configuration	
	_					✓ Software Packs	\checkmark
	~	٢				nd Configuration	
egories A->Z					M	ode	
System Core		>	Runtime contexts:	Cortex-M33 secure		Cortex-M33 non secure	
nalog		>		0		 Outor max account 	
naiog			VNOR Flash/PSRAM/SRAM/RC	DM/LCD 1			
mers		>	Chip Select NE1				\sim
onnectivity		~	Memory type NOR Flash				~
			Address 24 bits				Max: 26 bits
¢	M33S	M33NS	LCD Register Select Disab	le			\sim
FDCAN1 FMC	0	0	Data 8 bits				~
12C1	0	0	Data/Address Disable				Max: Disable
	0	0	Clock Disable				×
	0	0	Address valid				
12C4	0	0					~
12C5	0	0	Wait Disable				~
12C6	0	0	Byte enable				
IRTIM LPUART1	0	0	✓ NOR Flash/PSRAM/SRAM/RC	DM/LCD 2			
	0	0	Chip Select Disable				~
	0	0	Memory type Disable				\sim
SDMMC1	0	0	Address Disable				Max: Disable
	0	0	LCD Register Select Disab	le			\sim
	0	0	Data Disable				\sim
	0	0	Data/Address Disable				🌲 Max: Disable
SPI3 UART4	0	0	Clock Disable				\sim
UART5	0	0			Config	uration	
	0	0	Reset Configuration				
	0	0		Constants 🛛 🕑 GPIO Sett			
	0	0	NOR/PSRAM 1 User C Configure the below parameters :	onstants OFIO Sett	1195		
	0	0		ত			(
USART6 USB OTG HS	0	0	✓ NOR/PSRAM control	9			
	0	0	Memory type			NOR Flash	
			Bank			Bank 1 NOR/PSRAM 1	
fultimedia		>	Write operation			Disabled	
			Write FIFO Extended mode			Enabled Disabled	
Security		>	 NOR/PSRAM timing 			Lisabieu	
Computing		>	Address setup time in H			15	
liddleware and Soft	ware Packs	>	Data setup time in HCL Data hold time in HCLK			255 0	
	nare i ueno		Bus turn around time in			15	
race and Debug		>					
Power and Thermal		>					

Figure 375. Configure an external memory

When going back to the MMT, a new region corresponding to the added FMC is created.

teNX	File Window	Help	🛓 myST			🥸 🚹 🕑) Q 🔆 🖌
me 🔪 STI	N 32U549ZJTxQ 🔰 Untitled - Tools 🔪 LPBAM Scena	irio & Configuration	\rangle	GENERATE CODE		CHECK LPBAM DES	IGN
	Pinout & Configuration		Clock Configuration	Project Manager		Tools	
	Code Generation Configuration	~ F	Memory viewed by ARM Contex-MI33		Application Regions	+	1
mory	Apply application Regions settings to peripherals: ON Apply Application Regions settings to linker files: ON		cccook Reserved ccccook Octos SP11 (255 MB)	NA		Ê	
inagement	Search an Application Region	> 0x500	CCCCC FMC Bank3 (256 MB)	NA		0	
	Show selected Application Region	> 0x700	00000 DebsSPI2 (256 MB) 00000 FMC Bavkt (240 MB)	NA NA		0	
		0:0600	cccool External Flash (FMC) (16 MB)				
			SECO) Reserved SECO) Backup SRAM (S) (2 KB)			ů.	Display Settings R
PCC			See a second sec			â	Hide security bes
		0,8400	1640) Backup SRAM (NS) (2 KB)				Hide Reserved Al
			C4100 Reserved			ê	
			CCCCC SRAM4 Smart Run Domain (S) (16 KB)				
		0x302	TEOD Reserved				
			20100 SRAMS (S) (82 KB) 00100 SRAMS (S) (82 KB)			<u>م</u>	
CAD		0x3000	C0000 GRAM2 (S) (64 KB)				
CHU		0x300	ссоо SRAM1 (S) (768 КВ)	RAM			
			Reserved				

Figure 376. New region created



STM82CubeMX U	ntitled": STM32USA92/Ti-Q							- đ
NC2 abeNX	File Window	H	Help 💄 myST				🕸 🚹 🖸	y () 🕆 🗳
lame 🔪 ST	IN 32U549ZITXQ 🔰 Untitled - Tools 🔪 LPBAM Scen	iario & Corri	figuration 🔪	GENE	ATE CODE		CHECK LPBAM DE	ESIGN
	Pinout & Configuration		Clock Configu	ration	Pr	roject Manager	Tools	
	Code Generation Configuration	× 1		lemony viewed by ARM Contex-M33		Application Regions	+	1
	Apply application Regions settings to peripherals: ON Apply Application Regions settings to linker files: ON	0	exaccores Reserved			Harrow days	â	
	Search an Application Region	\rightarrow	Lascoccess DetuSP11 (256 MB) Lascoccess		NA NA		0	
	Show selected Application Region	\rightarrow	Lx70000000 OctoSPI2 (256 MB)		NA		0	
			Institucion FMC Bank1 (240 MB)		NA		0	
			tx60000000 External Flash (FMC) (16 MB) tx50000000 Reserved		Adda	an application region X	â	
			1x50056101 Ex50056400 Backup SRAM (S) (2 KB)		Name	e MjEdenalFlash		Display Settings Res
			tx40036C0) Reserved		Addr	es 160000	â	Hide security beside
			(x40036400 Backup SRAM (NS) (2 KB)		Size			Hide Reserved Aliase
			(x300400) Reserved		3428		â	
		_	(x3800000) SRAMA Smart Run Domain (S) (16 KB)			Cancel Add		
			0x30270000 Reserved				â	
			1x3012A(000 <mark>8RAM5 (S) (832 KB)</mark>					
			(1x30000000) SRAM3 (S) (832 KB)				ê	
CAD							1	1

Figure 377. Adding a new region

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.

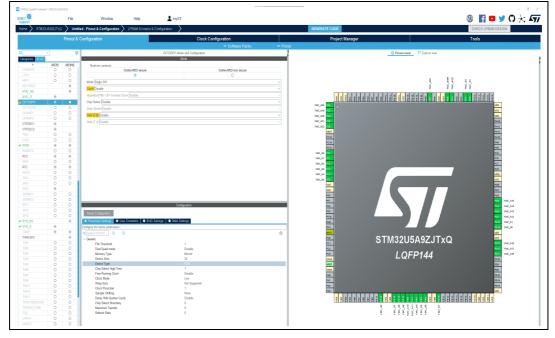


Figure 378. Adding a new memory

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.



ACC D	File	Window	н	Help 💄 myST				🛞 📑 🖻	y () 🔆 🏹
ame 🔪 ST	W32U548ZJTxQ 🔪 Untit	led-Tools 🔪 LPBAM :	Scenario & Corfi	figuration		GENERATE CODE		CHECK LPBAM D	SIGN
	Pinout &	Configuration			onfiguration	Project N	lanager	Tools	
	Code Generation Configuration	n	× \$	(AT)	Memory viewed by ARM Cortex-M33		Application Regions	+	1
	Apply application Rep			TEACOCCOD Reserved				A	
	Apply Application Re	jons settings to linker files: O	0 M	tasooon External RAM (OCTOSPI1) (256 MB)					
	Search an Application Region		>	tastoonen FMC Bank3 (256 MB)		NA	Add an application region X	0	
	Show selected Application R			1x7000000 OctoSPI2 (256 MB)		NA	Name MyExternaRAM	0	
	Show selected Application H	iĝos		Instituction FMC Bank1 (240 MB)		NA	Address 0x6000000	0	
	WyExternalFlash		0	Execoccess External Flash (FMC) (16 MB)		NyExternalFla			
	Name My6	xternalFlash		DeSCODECTOR Reserved				â	
	Core Name ARI	Contex-M33	v	Ex50036400 Backup SRAM (S) (2 KB)			Cancel Add		Display Settings Res
	Start address Dx9	1000000		Is40036CD0 Reserved				â	Hide security besid
	Size 16		WB v	Ix40056400 Backup SRAM (NS) (2 KB)					Hide Reserved Alian
	Security Sec	ane	v	1x38004000 Reserved				â	
	Access Permission RO	by privileged code only	v	1x3000000 SRAM4 Smart Run Demain (S) (16 KB)					
	Code execution Per	nitied	v	1x30270000 Reserved				â	
	Shareability Nor	Shareable	v	1x301A/(00) SRAM5 (S) (832 KB)					
	Cacheability Whit	e-Through Read Allocate for P	ROM/FL. V	1x30000000 SRAM3 (S) (832 KB)				ê	
				1x50000000 SRAM2 (S) (64 KB)					
				1x50000000 SRAM1 (S) (768 KB)		RAM			
				1x20004100 Reserved				â	
				1x2000000 SRAM4 Smart Run Domain (NS) (16 KB)					_
				0x20270000Reserved				â	
				0x201A0000 SRAM5 (NS) (832 KB)					
				0x20000000 SRAM3 (NS) (832 KB)		RAM			
				1x20000000 SRAM2 (NS) (64 KB)					1

Figure 379. Memory assignment

Configuring a memory region using the left panel

With the left panel (see *Figure 380*) 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 381*). The color is now pink (nonsecure), and the region starts in the middle of the RAM (OctoSPI1).

CENT.											y () 🔆 🖻
me > STN	W32U549ZJTxQ 🔪	Untitled - Tools	 LPBAM Scenar 	io & Corfigural	tion /	>	GENERATE COD			CHECK LPBAM DE	SIGN
	Pinc	out & Configurati	on	14.7		Clock Configuration		Project Manager		Tools	
	Code Generation Con	figuration		~		Nemory viewed by ARM Cortex-M33			Application Regions	+	1
	Apply applicat Apply Applicat	tion Regions settings to tion Regions settings to	peripheralis: ON linker files: ON	0	12A000000	Reserved				ê	
mory inagement	Search an Application	Region		· .	129000000	External RAM (OCTOSPI1) (256 MB)		WExternalRAM			
	Show selected Applic					FMC Bank3 (256 MB)		NA		0	
					127000000	OctoSPI2 (256 MB)		NA		0	
	MyExternalRAM		ť			FMC Bank1 (240 MB)		NA		0	
	Name	MyExternalRAM				External Flash (FMC) (16 MB)		NyExternaFlash			Display Settings Ret
	Core Name	ARM Cortex-M33 0x90000000		_	1850036000	Reserved Backup SRAM (S) (2 KB)					Hide security beside
PCC	Start address Size	64	100			Beserved				â	
	Security	Secure			Maccosecut	Backup SRAM (NS) (2 KB)					Hide Reserved Alias
	Access Permission	RW by prinleged co	de only			Reserved				â	
	Code execution	Permitted		× .	x3800000	SRAM4 Smart Run Domain (S) (16 KB)					
	Shareability	Non-Shareable		v (1:00270000	Reserved				â	
	Cacheability	Write-Back Read W	rite Allocate for RA			SRAM5 (5) (832 KB)					
CAD					x3000000	SRAM3 (5) (832 KB) SRAM2 (5) (64 KB)				•	
					1230000000	SRAM1 (S) (TEE KE)		RAM			
						Reserved				â	
					*2000000	SRAM4 Smart Run Domain (NS) (16 KB)					A.Y.

Figure 380. Left panel configuration



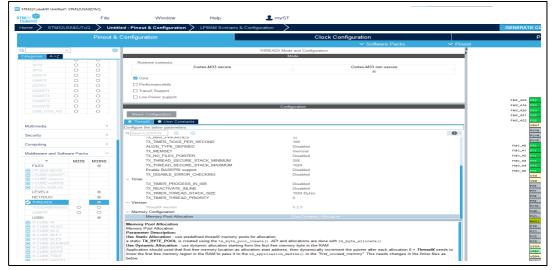
Figure	381.	Allocating	a region
--------	------	------------	----------

eWX	File Window		1 myst			🗖 🗡 🗘 💆
ne∕S∏	VS2USA9ZJTxQ 🔰 Untitled - Tools 🔪 LPBAM	Scenario & Configuration	n >	GENERATE CODE	CHECK LP	BAM DESIGN
	Pinout & Configuration	(A.T.	Clock Configuration	Project Manager	Tools	
	Code Generation Configuration	~	Memory viewed by ARM Cottex M33		Application Regions	+ 1
	Apply application Regions settings to perpherals: Apply Application Regions settings to Inker files:	ON 0 (sa	1000000 Reserved			â
nory nagement	Search an Application Region) (18	External RAM (OCTOSP11) (256 MB)	MyExternalRAM		
	Show selected Application Region		FNC Bank3 (256 MB)	NA		0
	WExternaRAM	(a)	Detection Detection (256 MB)	NA		0
		(0.6	1100000 FMC Bank1 (240 MB)	NA		
	Name MyExternalRAM		Edemail Flash (FMC) (16 MB)	MyEstamaFlash		Display Settings Res
	Core Name ARM Cottex-M33	(a5	10130000 Reserved 10136000 Backup SRAM (S) (2 KB)			
PCC	Start address 0x9000000					
	Siza 64	MB ~ (a4	10150CCC Reserved 10150CCC Backup SRAM (NS) (2 KB)			Hide Reserved Alia
	Security Non-Secure					
	Secure Read/Write Not Allowed		18 Decoce Researed			
	Access Permission RW by privileged code only		1270000 Reserved			
	Code execution Permitted		1270000 SRAWS (S) (832 KB)			
CAD	Shareability Non-Shareable		STAND (S) (82 KB)			A
00	Cacheability Write-Back Read Write Allocate	for RA. v (m3	SRAM2 (S) (64 KB)			-
		03	SRAM1 (S) (768 KB)	RAM		
			10000000 Reserved			
		082	BIOLOGIE SRAMA Smart Run Domain (NS) (16 KB)			

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.

Figure 382. Middleware memory allocation



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.



File Window	Help	💄 myST		® f	🗅 🄰 🗘 🔆 🖊
N 32U5442ZITxQ 🔰 Untitled - Tools 🔪 LPBAM Scen	aria & Configuration	\rangle	GENERATE CODE	CHECK LPB	
 Pinout & Configuration		Clock Configuration	Project Manager	Tools	
Code Generation Configuration	~	Nemory viewed by ARM Contex-M33		Application Regions	- 1
Apply application Regions settings to peripherals: ON Apply Application Regions settings to linker files: ON	0 1x200	Reserved			A 1
Appy Application Hegistis settings to invertiles: On	0	External RAM (OCTOSPI1) (256 M8)			-
Search an Application Region	> 00000	2000	MyExtenaRAM		
Show selected Application Region		FMC Bank3 (256 MB)	NA		0
	0x700	00105PI2 (256 MB)	NA		0
WyThreadXHeap	0 00010	PMC Bank1 (240 MB)	NA		0
Name MyThreadOHeap		Constant Flash (FMC) (16 MB)	MyExternalFlash		
Core Name ARM Cotex-M33	~ 0x500	Reserved			Display Settings
Start address (In201A0000		Backup SRAM (S) (2 KB)			Hide security be
Size (832	(B v 0x400	ecos Reserved			E Hide Reserved A
Security Non-Secure		R400 Backup SRAM (NS) (2 KB)			
Threadt Heap section	0x300	NEOD Reserved			8
Secure Read/Write Not Allowed		SRAM4 Smart Run Domain (S) (16 KB)			
Access Permission RW by privileged code only	× 0x302	Reserved			<u> </u>
Code execution Permitted		SRAMS (S) (832 KB)			ê.
Shareability Non-Shareable	~ 0x3000	xcoo ^{graan} 3 (5) (832 kB)			
Cacheability Write-Back Read Write Allocate for RA	. V	SRAM2 (S) (64 KB)			
	0x300				
		SEAMIN (5) (TEE KE)	RAM		
	0x280	Reserved			8
		SRAM4 Smart Run Domain (NS) (16 KB)			
	0x202	rcoop Reserved			<u>.</u>
	0x2013	(100) SRAMS (NS) (832 KB)	MyThreadXHeap		
	0x2000	50000 SRAM3 (HS) (832 KB)			Legend
		SRAM2 (NS) (64 KB)			Region allow
	0x200	2000 2000 SRAM1 (NS) (768 KB)	ruui		
					Secure regio
	0.0022	rccopReserved uccopSRAMAS Code (S) (832 KB)			Non Secure
	0.00013	Access Brivato Cubia (5) (522 KB) Const SRAM3 Cubia (5) (522 KB)			Non Secure

Figure 383. 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

Figure 384. Remapping the memory

W32		File	Window Help	L myST			
	U5A9ZJT×Q	Untitle	ed - Pinout & Configuration > LPBAM Scenar	io & Configuration			GENERATE CODE
		Pinout & C	Configuration	Clock	Configuration		Proje
					✓ Software Packs	✓ Pinout	
	~	۵		ICACHE Mode and Configuration			
ategories A->Z				Mode			
÷	M33S	M33NS	Runtime contexts:				
	-	۲	Cortex-M33 secure		Cortex-M33 non secure		
	۲				0		
	0	0	Memory address remap				
		0	Mode Disable			~	
			Secure Attribute not secured			~	
							8
							FMC_A23 PE2
	0	0		Configuration			FMC_A19 PE2
	0	0		Comganation			FMC_A20 PE4
	0	0	Reset Configuration				FMC_A21 PES
	0	0	Parameter Settings Subser Constants No	VIC Settings			PMC_A22 PEG
	0	0	Configure the below parameters :				VBAT
206	0	0				0	PC13
		•	Q Search (Clrl+F) ③ ④			0	PC14.
	0	0	✓ Region 0				PC15
	0	0	Region Size	Enable 2MB			FMC_AD PED
EVELX		۲	Base Address Range	[0x0,0x07FFFFFF]			FMC_A1 PE1
INKEDLIST			Base Address	0x0			FMG_A2 PF2
PDMA1	۲	۲	Remap Address	0x0000000			FMC_A3 PF3
	0	0	Traffic Route	Master1 port			FMC_A4
	0	0	Output Burst Type	WRAP			FMC_AS PFS
	0	0	~ Region 1				VSS
	0	0	Region	Disable			VDD
	0	0	✓ Region 2				PF6 PF7
	0	0	Region	Disable			PF7
	0	0	✓ Region 3				PFO
ETXDUO		۲	Region	Disable			PFB
VIC_NS		۲					PF 10
VIC_S	۲						PHO
CTOSPI1	۲	0					PH1
	0	0					NRST
	0	0					P C0 P C1 P C2
	0	0					PC1
OTFDEC1	۲						

• Go back to the Memory management Tool tab. Region 0x9000 0000 is named with Remapped, with the amount of RAM previously selected.



UM1718 Rev 46

	File Window	He			🤒 📑 🖣
ie 🗲 STP		icenario & Corfi		GENERATE CODE	CHECK LPB/
	Pinout & Configuration		Clock Configuration	Project Manager	Tools
	Code Generation Configuration	~ 1	Memory viewed by ARM Cortex M	33 Application Region	ف ا
	Apply application Regions settings to peripherals: Of Apply Application Regions settings to linker Res: ON	0	e+Lcooppie Reserved		
nory	 Apply Approximited proceedings to main life. On 		04140000000 Okto8P11 (128 MB)	NVA.	
agement	Search an Application Region	>	(154000)10 External RAM (OCTOSP11) (64 MB)	MyEsternaRAM	
	Show selected Application Region	~	(+10000000 External RAM (OCTOSP11) Remapped (54 MB)		
	brow sences Appearen regen		FMC Bank3 (256 MB)	NA	
	MyThreadOlleap	•	(110000)10 OctuSPI2 (256 MB)	NA	
	Name MyThread/Heap		(wilconner PMC Bankt (240 MB)	NA	
	Core Name ARM Cortex-M33	~	External Flash (FMC) (16 MB)	MyExternalFlash	
-oc	Start address 0x201A8000		eascosecre Reserved		
~~	Size 032	KD V	(15003641) Backup SRAM (5) (2 KB)		
	Security Non-Secure	~	de40034020 Reserved		
	Thread/ Heap section		0x40026410 Backup SRAM (NS) (2 KB)		
	Secure Read/Write Nat Allowed		Ex30004010 Reserved		
	Access Permission RW by privileged code only	~	(#380000000 SRAM4 Smart Run Demain (5) (15 K8)		
	Code execution Permitted		sa30270000 Reserved		
CAD	Shareability Non-Shareable		1x301A0310 SRAMS (S) (832 KB)		
	Cacheability Write-Back Read Write Allocate for	RA V	8+30000000 SRAMS (5) (832 KB)		
		104	SRAM2 (5) (51 KD)		
			(#10000010		
			(#X0000000 SRAM1 (S) (768 KB)	RAM	
			0+22004000 Reserved		
			SRAM4 Smart Run Demain (NS) (16 KB)		
			(#202700110 Reserved		
			1+20120210 SRAMS (NS) (032 KB)	MyThreadOlleap	
			0#20000000 SRAM3 (NS) (832 KB)		
			SRAM2 (NS) (64 KB)		
			8+20000318	RAM	
			9420000000 SRAM1 (NS) (768 KB)		
			(x(f270))) Reserved		
			0x(6(30)10 SRAMS Code (5) (832 KB)		
			04.08000000 SRAM3 Code (5) (832 KB)		

Figure 385. Remapped region is renamed

• There is also a Remap – External RAM(OCTOSPI1) added at address 0x0000 0000.

Figure 386. Remapped start address

0x08400000	
0x08200000 Flash Bank2 memory (NS) (2 MB)	FLASH
Flash Bank1 memory (NS) (2 MB)	â
0x08000000	â
0x04000000 Reserved	â
0x00000000 Remap - External RAM (OCTOSPI1) (64 MB)	

• Add a new region named "MyRemappedRAM" at that address.

Figure 387. New region remapped	
	-

Code Generation Config	guration	×	Memory view	ved by ARM Cortex-M33	Application Regions
Apply applicatio Apply Applicatio	on Regions settings to peripherals: ON on Regions settings to linker files: ON	0	SRAM2 Code (S) (64 KB) 0x0E000000		, pproducer region
Search an Application F	Region	\rightarrow	0x0E0000000 <mark>SRAM1 Code (S) (768 KB)</mark> 0x0E0000000 <mark>Reserved</mark>		<u></u>
Show selected Applicat	tion Region	~	0x0C200000 Flash Bank2 memory (S) (2 MB)		
MyRemappedRAM		Û	Flash Bank1 memory (S) (2 MB) 0x0C000000	FLA: FLA:	SH_NSC SH
Name	MyRemappedRAM		0x0A270000 Reserved		â
Core Name	ARM Cortex-M33	V	Ox03130000 SRAM5 Code (NS) (832 KB)		â
Start address	0x00000000		0x0A0D0000 SRAM3 Code (NS) (832 KB)		
Size Security	64 Secure	MB v	SRAM2 Code (NS) (64 KB)	MyB	Suffer
Access Permission	RW by privileged code only	\vee	0x0A000000 SRAM1 Code (NS) (768 KB)		â
Code execution	Permitted	V	0x08400000 Reserved		â
Shareability	Non-Shareable	V	0x08200000 Flash Bank2 memory (NS) (2 MB)	FLA:	SH
Cacheability	Write-Back Read Write Allocate for R/	€ ∨	Flash Bank1 memory (NS) (2 MB) 0x08000000		â â
			0x04000000 Reserved		â
			0x00000000 Remap - External RAM (OCTOSPI1) (64 MB)	MyF	RemappedRAM

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.



me 🔪 Si	M32USA9ZJTXQ 🔪 mi	mt2.ioc - Tools 🔰 LPBAM Scen	ario & Configuration		GENERATE CODE	CHECK LPBAM DESIGN
	Pinout & Config	juration		Clock Configuration	Project Manager	Tools
	Code Generation Configu	ration	F	Memory viewed by ARM (Cortex-M33 Application Regions	+
mory nagement	Apply application Apply Application	Regions settings to peripherals: ON Regions settings to linker files: ON	0	0x0E0C0000		a
	Search an Apolication Re	nina	>	0x0E000000 SRAM1 Code (S) (768 KB)		Display Sattings Reset View
PCC	Show selected Applicatio		· · ·	0x0C400000 Reserved 0x0C200000 Flash Bank2 memory (5) (2 MB)		Hide security beside regions names
	FLASH		Û	Flash Bank1 memory (S) (2 MB)	FLASH_NSC FLASH	Hide Reserved Allased regions names
CAD	Name	FLASH		0x0C000000 0x0A270000		a
0,70	Core Name Start address	ARM Cortex-M33	~	0x0A1A0000 SRAM5 Code (NS) (832 KB)		a
	Size	2048	KB V	0x0A0D0000 SRAM3 Code (NS) (832 KB)	+	Legend
	Security	Non-Secure	\sim	SRAM2 Code (NS) (64 KB)		Region allowing different types of security.
	Default Code Region			0x0A0C0000 0x0A0000000 SRAM1 Code (NS) (768 KB)		Secure region (5).
	Access Permission	RO by privileged code only	~	0x08400000 0x08400000 Reserved		Non Secure region (NS).
	Code execution	Permitted	~	0x08200000 Flash Bank2 memory (NS) (2 MB)	FLASH	Non Secure Callable region (NSC).
	Shareability Cacheability	Non-Shareable Write-Through Read Allocate for I	~ ROM/ ~	Flash Bank1 memory (NS) (2 MB)		Region accessible by Secure and Non-Sec
				0x04000000 Reserved		Reserved region.
				0x00000000 Remap - External RAM (OCTOSPI1) (64 MB)	MyRemappedRAM	A new application region can be added he

Figure 388. Resizing default 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	<u> </u>	Memory viewed by ARM Cortex-M33	Application Regions	+
		Regions settings to peripherals: ON	0 0x280040			
mory nagement	Apply Application	Regions settings to linker files: ON		00 ^{SRAM4} Smart Run Domain (NS) (16 KB)		
nagement	Search an Application Re	gion	> 0x202700			â
			0x201A00	00 SRAM5 (NS) (832 KB)	MyThreadXHeap (NS)	
	Show selected Application	n Region		00 SRAM3 (NS) (832 KB)		
	MyBuffer	ť	J		MyBuffer (NS) Reserved Alias Region	۵
PCC	Name	MyBuffer	0x200C00	SRAM2 (NS) (64 KB)	RAM (NS)	
FUU	Core Name	ARM Cortex-M33		00 90 SRAM1 (NS) (768 KB)	RAM (S) Reserved Alias Region	6
	Start address	0x0A0C4400	0x0E2700			a
	Size	47 KI		00 90 SRAM5 Code (S) (832 KB)	MyThreadXHeap (NS) Reserved Alias Region	6
	Security	Non-Secure	V OXOETADO	00 00 SRAM3 Code (S) (832 KB)		
	ThreadX Heap section		OXOEODOO	00	MyBuffer (NS) Reserved Alias Region	â
CAD	Secure Read/Write	Not Allowed	~	SRAM2 Code (S) (64 KB)	RAM (NS) Reserved Alias Region	
	Access Permission	RW by privileged code only	0x0E0C00			
	Code execution	Permitted		00 SRAM1 Code (S) (768 KB)	RAM (S) Reserved Alias Region	6
	Shareability	Non-Shareable	v 0x0C4000			<u> </u>
	Cacheability	Write-Back Read Write Allocate for RA	V 0x0C2000	00 Flash Bank2 memory (S) (2 MB)	FLASH (NS) Reserved Alias Region	<u> </u>
				Flash Bank1 memory (S) (2 MB)	FLASH_NSC (NSC)	
			0x0C0000		FLASH (S)	
			0x0A2700	00 Reserved		6
				00 SRAM5 Code (NS) (832 KB)	MyThreadXHeap (NS) Reserved Alias Region	â
				SRAM3 Code (NS) (832 KB)		
			01010200		MyBuffer (NS)	
			0x0A0C00	SRAM2 Code (NS) (64 KB)	RAM (NS) Reserved Alias Region	<u> </u>
				00 00 ^{SRAM1 Code (NS) (768 KB)}	RAM (S) Reserved Alias Region	
			0x084000	00 Flash Bank2 memory (NS) (2 MB)	FLASH (NS)	
			0x082000	00 ^{- 100} 0000 (100) (100) (100)		
				Flash Bank1 memory (NS) (2 MB)	FLASH_NSC (S) Reserved Alias Region	<u> </u>
			0x080x0		FLASH (S) Reserved Alias Region	<u> </u>
			0x040000	00 ^{Reserved}		<u><u></u></u>
				00 Remap - External RAM (OCTOSPI1) (64 MB)	MyRemappedRAM (S)	

Figure 389. 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.

_Min_Heap_Size = 0x200; /* required amount of heap */ ^	Pinout & Co	nfiguration Clock Configuration	Project Manager
_Min_Stack_Size = 0x400; /* required amount of stack */	\$	Memory viewed by ARM Cortex-M33	Application Regions
/* Memories definition */			
MEMORY		0x98000000	NA
FLASH (rx) : ORIGIN = 0x0c000000, LENGTH = 2040K	Memory	0x94000000 External RAM (OCTOSPI1) (64 MB)	MyExternalRAM (NS)
FLASH NSC (rx) : ORIGIN = 0x0c1fe000, LENGTH = 8K	Memory Management	0x90000000 External RAM (OCTOSPI1) Remapped (64 MB)	MyRemappedRAM (S) Reserved Alias Region
MyExternalFlash (rx) : ORIGIN = 0x60000000, LENGTH = 16384K		0x80000000 FMC Bank3 (256 MB)	N/A
RAM (xrw) : ORIGIN = 0x30000000, LENGTH = 768K			NA
MyRemappedRAM (xrw) : ORIGIN = 0m00000000, LENGTH = 65536K		0x70000000 OctoSPI2 (256 MB)	
· · · · · · · · · · · · · · · · · · ·		0x61000000 FMC Bank1 (240 MB)	NA
/* Sections */		0x60000000 External Flash (FMC) (16 MB)	MyExternalFlash (S)
SECTIONS		0x50036C00	í
/* The startup code into "FLASH" Rom type memory */		0x50036400 Backup SRAM (S) (2 KB)	
.isr_vector :	PCC		
		0x40036C00 ^{Reserved}	É
<pre>KEEP(*(.isr_vector)) /* Startup code */ -) >FLASH</pre>		0x40036400 Backup SRAM (NS) (2 KB)	
		0x38004000 Reserved	é
/* The program code and other data into "FLASH" Rom type memory */		0x360000000 SRAM4 Smart Run Domain (S) (16 KB)	
.text :			
(.text) / .text sections (code) */		0x30270000 Reserved	
(.text) /* .text* sections (code) */		0x301A0000 SRAM5 (S) (832 KB)	MyThreadXHeap (NS) Reserved Alias Region
(.glue_7) / glue arm to thumb code */		0x300D0000 SRAM3 (S) (832 KB)	
<pre>*(.glue_7t) /* glue thumb to arm code */ *(.eh frame)</pre>	CAD		MyBuffer (NS) Reserved Alias Region
(.en_rione)		SRAM2 (S) (64 KB)	
KEEP (*(.init))		0x300C0000	RAM (NS) Reserved Alias Region
KEEP (*(.fini))		0x30000000 SRAM1 (S) (768 KB)	RAM (S)
_etext = .; /* define a global symbols at end of code */		0x28004000 Reserved	6
- } >FLASH		0x28000000 SRAM4 Smart Run Domain (NS) (16 KB)	
<pre>/* Constant data into "FLASH" Rom type memory */ .rodata :</pre>		0x20270000	
E (0x201A0000 SRAM5 (NS) (832 KB)	MyThreadXHeap (NS)
(.rodata) / .rodata sections (constants, strings, etc.) *		0x200D0000 SRAM3 (NS) (832 KB)	
<pre>*(.rodata*) /* .rodata* sections (constants, strings, etc.) >>FLASH</pre>			MyButter (NS) Reserved Alias Region
-) >FLASH		SRAM2 (NS) (64 KB)	RAM (NS)
.ARM.extab :		0x200C0000	
白 (0x20000000 SRAM1 (NS) (768 KB)	RAM (S) Reserved Alias Region
<pre>*(.ARM.extab* .gnu.linkonce.armextab.*) } >FLASH</pre>		0x0E270000 Reserved	6
) / ELKON		0x0E1A0000 SRAM5 Code (S) (832 KB)	MyThreadXHeap (NS) Reserved Alias Region
.ARM :		0x0E0D0000 SRAM3 Code (S) (832 KB)	
		0x0E0D0000	
exidx_start = .; *(.ARM.exidx*)		SRAM2 Code (S) (64 KB)	MyBuffer (NS) Reserved Alias Region
exidx_end = .;		0x0E0C0000	RAM (NS) Reserved Alias Region
-) >FLASH		0x0E000000 SRAM1 Code (S) (768 KB)	RAM (S) Reserved Alias Region
.preinit_array :		0x0C400000 Reserved	6
.preinit_array :		0x0C400000	
<pre>PROVIDE_HIDDEN (preinit_array_start = .);</pre>		0x0C200000 Flash Bank2 memory (S) (2 MB)	FLASH (NS) Reserved Alias Region
<pre>KEEP (*(.preinit_array*))</pre>		Flash Bank1 memory (S) (2 MB)	FLASH_NSC (NSC)
<pre>PROVIDE_HIDDEN (preinit_array_end = .); >>FLASH</pre>		0x0C000000	FLASH (S)
,		0x0A270000 Reserved	6
.init_array :		0x0A270000	
		0x0A1A0000 SRAM5 Code (NS) (832 KB)	MyThreadXHeap (NS) Reserved Alias Region
PROVIDE HIDDEN (init array start = .);		0x0A0D0000 SRAM3 Code (NS) (832 KB)	
length : 5 076 lines : 194 Ln : 51 Col : 53 Sel : 48 L1 Unix (LF) UTE-8 UNS			16.Q. Mer (810)

Figure 390. 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 391* and *Figure 392*): modes and parameters become read-only.

The middle panel (see, respectively, *Figure 393* and *Figure 394* 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.

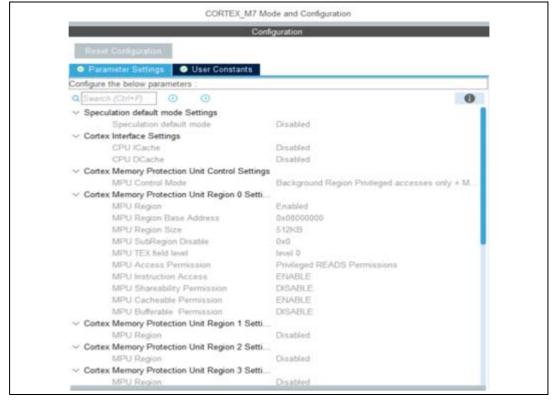
For the new project created under STM32CubeMX the tool creates the default application region to generate a valid project.



Figure	391.	ММТ	usage	(STM32U5)
--------	------	-----	-------	-----------

CORTEX_M	33 Mode and Configuration
	Configuration
Reset Configuration	
Parameter Settings Settings Settings Settings Settings Settings Settings Settings Set User Constants	\$
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	0×0800000
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	etti
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 46

Memory viewed by ARM Cortex-M33	Application Regions	+
xA0000000 Reserved		6
x90000000 OctoSPI1 (256 MB)	N/A	0
x40036C00 Reserved		6
x40036400 Backup SRAM (2 KB)		
x28004000 Reserved		6
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 Alice Design	
x0A000000 SRAM1 Code (192 KB)	RAM Reserved Alias Region	6
x08040000 Reserved		É
x08000000 Flash Bank1 memory (256 KB)	FLASH	
x0000000Reserved		é

Figure 393. MMT view for U5 without TrustZone

Figure 394. MMT view for H7 single core

	, , , , , , , , , , , , , , , , , , ,				
F	Pinout & Configuration		Clock Configuration	Project Manager	Tools
	Code Generation Configuration	\sim	Memory viewed by Arm	Cortex-M7	Application Regions
	Apply Application Regions Settings to Peripherals: ON		0x90000000 OCUSPTI (200 MD)	IWA	
	Apply Application Regions Settings to Linker Files: OFF	0	0x80000000 FMC Bank3 (256 MB)	N/A	
	Search an Application Region	>	0x70000000 OctoSPI2 (256 MB)	N/A	
			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	
FCC			0x30008000Reserved		
			0x30004000 AHB SRAM2 (16 KB)	RAM_D2	
			0x30000000 AHB SRAM1 (16 KB)	-	
			0x24020000 Reserved		
			0x24000000 AXI SRAM (128 KB)	RAM	
			0x20020000 Reserved		
CAD			0x20000000 DTCM memory (128 KB)	DTCMRAM	
			0x1FF20000Reserved		
			0x1FF00000 System memory (128 KB)		
			0x10008000Reserved		
			0x10004000 AHB SRAM2 Code (16 KB)		
			0x10000000 AHB SRAM1 Code (16 KB)		
R Test Suite			0x08080000 Reserved		
			0x08000000 Flash memory (512 KB)	FLASH	
			0x00010000 Reserved		
			0x00000000 ITCM 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 395*).



ime 🔪 STM3	32H723ZGIx 〉 U	ntitled - Tools >					GENERATE CODE
F	Pinout & Configu	ration	с	lock Configuration	Project Manager		Tools
	Code Generation Co	onfiguration	>	Memory viewed	by Arm Cortex-M7	Application Regions	-
		ation Regions Settings to Peripheral ation Regions Settings to Linker File		0x90000000 FMC Bank3 (256 MB)	N/A		
	Search an Application	on Region	>	0x70000000 OctoSPI2 (256 MB)	N/A		
	Show selected Appl	ication Region	~	_{0x60000000} FMC Bank1 (256 MB) _{0x38801000} Reserved	N/A		
	RAM		÷	0x38800000 Backup SRAM (4 KB)			
	Name	RAM		0x38004000 Reserved			
	Core Name	Arm Cortex-M7	~	0x38000000 AHB SRAM Domain 3 (16 KB)	RAM_D3		
	Start address	0×24000000		0x30008000 0x30004000 AHB SRAM2 (16 KB)			
	Size	128	KB 🗸	0x30004000 AHB SRAM1 (16 KB)	RAM_D2		
	Default Data Region			0x30000000 Reserved			
	Access Permission	RW by any privilege level	~	0x24020000 0x24000000 AXI SRAM (128 KB)	RAM		
	Code execution	Permitted	~	0x20020000 Reserved			
	Shareability	Non-Shareable	~	0x200000000 DTCM memory (128 KB)	DTCMRAM		
	Cacheability	Write-Back Read Write Allocate for	RW ~	0x1FF20000 Reserved			
				0x1FF00000 System memory (128 KB)			
				0x10008000 Reserved			
				0x10004000 AHB SRAM2 Code (16 KB)			
				0x10000000 AHB SRAM1 Code (16 KB)			
R Test Suite				0x08100000 Reserved			
				0x08000000 Flash memory (1 MB)	FLASH		
				0x00010000 Reserved			
				0x00000000 ITCM memory (64 KB)	ITCMRAM		

Figure 395. Default data region

FMC impact on MMT

When activating FMC and SDRAM Bank1, a tab mapping (see *Figure 396*) is displayed, with three options:

- 1. Default mapping (see *Figure 397*): 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 398* and *Figure 399*)
- 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 400* and *Figure 401*)



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		
✓ 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	
	Congulation	
Reset Configuration		
NOR/PSRAM 1 Bank Mapping NOR/PSRAM 2 SDRAM 1 SDRAM 2 User Constant	nts O NVIC Settings O GPIO Settings	
Configure the below presenters		
Q Search (Ctrl+F) 0 0		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	
	NORPSRAM bank and SURAM bank 1/bank2 are swapped SDRAM Bank2 remapped on FMC bank2 and still accessible at default mapping	
	por over came remapped on rind dame and add accessible at deadlic mapping	



Figure	397.	Default	mapping
--------	------	---------	---------

Memory viewed by ARM Cortex-M7		Application Regions	+
xE000000 Reserved			6
xD0000000 FMC SDRAM Bank2 (256 MB)	N/A		
xC0000000 FMC SDRAM Bank1 (256 MB)	N/A		
xA000000 Reserved			6
0x90000000 QuadSPI (256 MB)	N/A		
0x80000000 FMC Bank3 (256 MB)	N/A		
x7000000 Reserved			6
Dx64000000 FMC Bank1 (192 MB)	N/A		0
Dx60000000 External Flash (FMC) (64 MB)			
x38801000 Reserved			6
0x38800000 Backup SRAM (4 KB)		+	
x38010000 Reserved			6
0x38000000 AHB SRAM Domain 4 (64 KB)	RAM_D3		
x30048000 Reserved			6



	Memory viewed by Arm Cortex-M7		Application Regions	+
0xE0000000	Reserved			6
0xD0000000	FMC SDRAM Bank2 (256 MB)	N/A]
	FMC SDRAM Bank1 (256 MB)	N/A		
0xA0000000	Reserved			ê ;
0x9000000	OctoSPI1 (256 MB)	N/A		
	FMC Bank3 (256 MB)	N/A		
	OctoSPI2 (256 MB)	N/A		
	FMC Bank1 (192 MB)	N/A		0
0x60000000	External Flash (FMC) (64 MB)			
0x38801000				Ô (
0x38800000	Backup SRAM (4 KB)			
0x38004000				â
0x38000000	AHB SRAM Domain 3 (16 KB)	RAM_D3		
0x30008000				â



Figure 399. After the swap

Memory viewed by Arm Cortex-M7	Aţ	pplication Regions	+
xE0000000Reserved			â
xD0000000 FMC SDRAM Bank2 (256 MB)	N/A		
xC4000000 FMC Bank1 (192 MB)	N/A		0
xC0000000 External Flash (FMC) (64 MB)			
xA0000000Reserved			6
x90000000 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			â
x38800000 Backup SRAM (4 KB)			
x38004000 Reserved			â
x38000000 AHB SRAM Domain 3 (16 KB)	RAM_D3		
x30008000Reserved			â

Figure 400. Before remapping

	Memory viewed by ARM Cortex-M7	Application Regions	+
0xE0000000 Reserved			6
0xD0000000 FMC SD	RAM Bank2 (256 MB)	N/A	
	RAM Bank1 (256 MB)	N/A	
0xA0000000 Reserved			â
0x90000000QuadSPI	(256 MB)	N/A	
0x80000000 FMC Bar	nk3 (256 MB)	N/A	
0x70000000 Reserved			â
0x64000000 FMC Bar	uk1 (192 MB)	N/A	0
	Flash (FMC) (64 MB)		
_{0x38801000} Reserved			â
0x38800000Backup	SRAM (4 KB)	+	
0x38010000 Reserved			â
	AM Domain 4 (64 KB)	RAM_D3	
0x30048000 Reserved			â
0x30040000 AHB SR	AM3 (32 KB)		



Memory viewed by ARM Cortex-M7	Application Regions	+
Economo Reserved		â
FMC SDRAM Bank2 (256 MB)	N/A	0
	AppReg2 Reserved Alias Region	â
CO800000 FMC SDRAM Bank1 (248 MB)	N/A	0
cooooooo External RAM (FMC) (8 MB)	AppReg1	
KA0000000 Reserved		â
490000000 QuadSPI (256 MB)	N/A	
480000000 FMC Bank3 (256 MB)	N/A	
r70400000 FMC SDRAM Bank2 (252 MB)	N/A	0
x70000000 External RAM (FMC) (4 MB)	AppReg2	
x60000000 FMC Bank1 (256 MB)	N/A	
x38801000 Reserved		6
K38800000 Backup SRAM (4 KB)	AppReg0	
x38010000 Reserved		6
AHB SRAM Domain 4 (64 KB)	RAM_D3	
x30048000 Reserved		6
430040000 AHB SRAM3 (32 KB)		
430020000 AHB SRAM2 (128 KB)	RAM_D2	
430000000 AHB SRAM1 (128 KB)		
x24080000 Reserved		â
x24000000 AXI SRAM (512 KB)	RAM	
x20020000 Reserved		â
20000000 DTCM memory (128 KB)	DTCMRAM	

Figure 401. After remapping

5.5.4 STM32WBxx

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 402*).





The user must select the Core and the STM32Cube firmware from a list. It is possible to choose any STM32Cube firmware version (see *Figure 403*).

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.

In this example, we loaded an STM32WB5x MCU, so the list must contain only stm32wb5x_x binaries. the button "Refresh" is used to refresh the binaries list version existing in the repository of STM32Cube firmware (see *Figure 404*).



ne 🔪 STM32WB55CCUx 🔪 Untitled - Tools 🔪			GENERATE CODE
Pinout & Configuration	Clock Configuration	Project Manager	Tools
	Please Select a WB Cortex-M0+ firmware	e version from the list. Select Refresh	
mory nagement	□ stm32wb5x_BLE_HC	ClLayer_fw.bin	
nagement	stm32wb6x_BLE_St	ack_fw.bin	
	stm32wb6x_BLE_Th	read_fw.bin 1.6.0 V	
	stm32wb6x_Mac_80	2_15_4_fw.bin 1.19.0 V	
PCC	stm32wb5x_rfmonito	r_phy802_15_4_fw.bin 1.1.0 \	
	stm32wb5x_Thread_	FTD_fw.bin 1.19.0 V	
	stm32wb5x_Thread_	MTD_fw.bin 1.19.0 ~	
	□ stm32wb6x_BLE_HC	Cl_AdvScan_fw.bin	
CAD	□ stm32wb5x_BLE_LLI	D_fw.bin 1.18.0 ~	
	stm32wb5x_BLE_St	ack_full_fw.bin	
	□ stm32wb5x_BLE_Sta	ack_light_fw.bin 1.19.0 V	
	stm32wb5x_BLE_Th	read_dynamic_fw.bin	
R Test Suite	stm32wb6x_BLE_Th	read_static_fw.bin 1.19.0 V	
	stm32wb6x_BLE_Zig	bee_FFD_dynamic_fw.bin 1.19.0 V	
	stm32wb5x_BLE_Zig	bee_FFD_static_fw.bin 1.19.0 V	

Figure 402. MMT usage

Figure 403. Firmware version

Please Select a WB Cortex-M0+ firmware version from the list.	Select	Refresh
✓ stm32wb5x_BLE_HClLayer_fw.bin	1.19.0 ~	



lome 🔰 STM3	2WB55CCUx 🔰 Untitled - Tools 🔪			GENERATE CODE
F	Pinout & Configuration	Clock Configuration	Project Manager	Tools
	Code Generation Configuration		nd by ARM Cortex-M4 Ag	oplication Regions +
lemory	Apply Application Regions Settings to Peripherals Apply Application Regions Settings to Linker Files		N/A	Â
anagement	Search an Application Region	> 0x20040000 Reserved		â
	Show selected Application Region	<u>></u>	CM0 Firmware Ram2b	<u> </u>
		SRAM2 (64 KB)	CM0 Firmware Ram2a	Ê
PCC		0x20030000	RAM_SHARED	
		0x20010000 Reserved		
		SRAM1 (64 KB)	RAM	
		0x20000000 0x10010000 Reserved		8
CAD			CM0 Firmware Ram2b Reserve	id Alias Region 🔒
		SRAM2 Code (64 KB)	CM0 Firmware Ram2a Reserve	d Alias Region 🔒
		0x1000000	RAM_SHARED Reserved Aliat	
		0x08040000 Reserved	CM0 Fernware Flash	
R Test Suite		Flash memory (256 KB) 0x08000000	CMD Firmware Flash FLASH	
		0x00000000 Reserved		6

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.



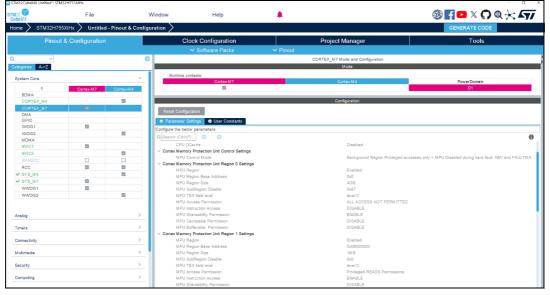


Figure 405. CORTEX_M7 Mode and Configuration



l igure 400.	CORTEX_I	4 Mode and Co	Ingulation					
CORTEX_M4 Mode and Configuration								
	м	ode						
Runtime contexts:								
Cortex-M7	Cort	ex-M4	PowerDomain					
		✓	D2					
	Config	guration						
Reset Configuration								
Settings ♀ Parameter Settings	User Co	nstants						
Configure the below parame								
Search (Ctrl+F)	\odot		•					
Cortex Memory Protecti								
MPU Control Mo		Background R	egion Privileged acces					
Cortex Memory Protecti	on Unit R							
MPU Region		Enabled						
MPU Region Bas	e Address	0x08100000						
MPU Region Size	Э	1MB						
MPU SubRegion	Disable	0x0						
MPU TEX field le	vel	level 0						
MPU Access Per	mission	Privileged REA	DS Permissions					
MPU Instruction	Access	ENABLE						
MPU Shareability	Permission	DISABLE						
MPU Cacheable	Permission	ENABLE						
MPU Bufferable	Permission	DISABLE						
Cortex Memory Protecti	on Unit R							
MPU Region		Enabled						
MPU Region Bas	e Address	0x30000000						
MPU Region Size		128KB						
MPU SubRegion		0x0						
MPU TEX field le		level 1						

Figure 406, CORTEX M4 Mode and Configuration

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:

First configuration:



Apply Application Regions Settings to Peripherals: ON Apply Application Regions Settings to Linker Files: OFF



UM1718 Rev 46

69

6

Second configuration:

Apply Application Regions Settings to Peripherals: ON (1) Apply Application Regions Settings to Linker Files: ON (1)

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

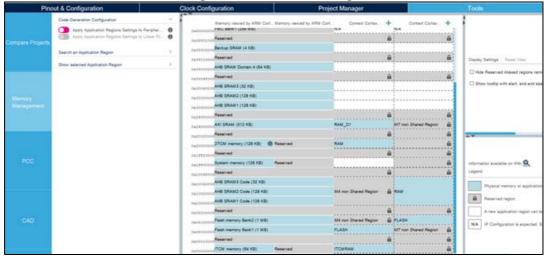


Figure 407. Default settings

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.



Choose a supported MCU. 1.

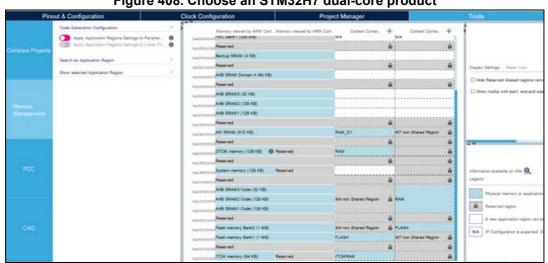


Figure 408. Choose an STM32H7 dual-core product

Click on the Start Project button, then choose Yes on the "Memory Protection Unit for 2. Cortex-M7" dialog box.

		Fe Block Di Do	cs & Res CAD Reso	Data 🖬 🕞 Start		
Commercial STM32H755XIH6TR	~	STM32H7 Series				
	+-	STM32H755XIH6TR		CU with 2MBytes of Flash MHz CPU, Art Accelerator, ry interface, large set of		
Segment Series Line	>	ACTIVE Product is in mass production	Unit Price for 10kU (US\$) : 12.0232	TFBGA 240+25 14x14x1 2		
Marketing Status > Price >		STM32H755xI devices are based on the high-performance Arm [®] Cortex [®] -M7 and Cortex [®] -M4 32- bit RISC cores. The Cortex [®] -M7 core operates at up to 480 MHz and the Cortex [®] -M4 core at up to				
Package	2		a fination point unit (EDII) which ex			
Core						
Coprocessor	> M	CUs/MPUs List: 1 item		🛆 Expor		
MEMORY	~	STM32H755XIHETR STM	32. Active TFBG	Flash Non Post Press 2048 kB 1024 kB 172 480 M		
Flash = 2048 (kBytes)						

Figure 409. Choose an STM32H7 dual-core product

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.



Figure	410.	Region	0	added
--------	------	--------	---	-------

		CORTEX,N	Fixed view 1" Syllett cau			
Canalana West			0	Nate		
Speen Gare			Restore particular	Constant PosseConstr.		
+	Context #17	Consulta	12	Concession (197)		
BOMA						
CONTENJAR		8			NAME OF TAXABLE PARTY AND ADDRESS OF TAXABLE PARTY.	
CONTRA INF		12				
OMA.					FC6697688888888888	
040					668666888288288288	
18051						
recol		19		- Contraction of the Contraction	11日日日日二日日 二日二日日日日日日	
MONA MUNICI				Companyar		
			Read Cardgesters			
10/102					日本市長ら とうこうち 日間着き	
100000	-	0	· # Parandar Sellings		8946	
RCC		8	Configure the below percentence		BB-BB ANANA ABB-	
W 212,64			Ritsent Stren	0	8888	
100 Jan 10 Jan 1	8		SPU1Carlw	Creatived	Legal Local Legal	
WWDG/	8		CPU DEwite	Contract		
WWDG3			Contex Marrory Protection Unit Centrel Satings		CREASECTRONCERS	
			WPU Control Mude	Raikgraund Region Privileged anianeau arty + http://Deablest .	と使用を用きった思いたと言語で	
			- Contex Memory Protection Unit Region 1 Settings		因因在世界的目标的影响和影响	
Aneropi		(i)	MPL Region	E-stad	A B B C B B B B B B B B B B B B B B B B	
			NPU Report Sale Address	040		
Times		1.1	MPU Regist Stee	40340	-00000000000000000000000000000000000000	
			NPU Susfager Deate	6-87		
Cereachily			MP12 36X Send level	incel 0	TF9GA340+25 (Top view)	
			970 Access Permission	AU, ADDESS NOT PERMITTED		
N.Briefle			MPG Indication Access	CIGARLE		
20.24			MPU Shareability Permakan	6%48.Z		
Decarty			MPU Cacheeline Permission	CHLARLE		
Careading			VPU Buffwable Permann	OGABLE	1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
and the second s			 Galles Merring Polacion Unit Region 1 Settings 	Destint		
			MIPJ Region	CM05042		

- 3. Select the "Tools" in the toolbar
 - Choose Memory Management.
 - Activate the Memory Management Tool support by clicking the radio button "Apply Application Regions Settings to Peripherals".

STRUCCLEAVE LANSER I	File	Window	Help				8 f •	× n • +	<
Home > STW32H	75530Hx 🔪 Untitled - Tools 🔪						_	RATE CODE	
Pino	out & Configuration		Clock Configuration		Project Manager			Tools	
Compare Projecte	Cole Generation Configuration Cole Generation Configuration Apply Application Regions Settings Apply Application Region Search an Application Region Q. Image: Configuration Region		Marroy danad by Attil C tubrescent Reserved tubrescent DeadOn (DB W0) tubrescent PMC Bank3 (DB W0) tubrescent Reserved	rfan-367	Memory viewed by AMM Contex-914	N.A. N.A.	 hen. hen.	Contact Dorbert14	4
Memory Management	Elson salacted Application Region	~	Instances PAC Bank (2014 M3) Instances Instances Instances Instances Instances Instances Instances Instances Instances Instances Instances Instances Instances Instances	,		N.A.	інл. Ф.		â
PCC			162010000 (MH2 SDAAS (32 KB) 162012000 (MH2 SDAAS (128 KB) 162112000 (MH2 SDAAS (128 KB) 162112000 (MH2 SDAAS (128 KB) 162112000 (KE SBAAS (512 KB)				iii http://www.com/article/	e Stand Region	â
			Record Assessed Record OTOM memory (128 KB) Record Reserved	0	Reserved		 		-

Figure 411. Activate Memory Management support

The default application regions are in exclusive mode (context sharing is unselected).

A reserved region in the other context will be created and mentioned as "Mx non-shared region".



Code Generation Confi	guration	Y
Apply Applicatio	on Regions Settings to Peripher	6
Apply Application	on Regions Settings to Linker Fil	0
Search an Application F	Region	>
Show selected Applicat	ion Region	~
Name	FLASH	
Core Name	ARM Cortex-M4	~
Start address	0x08100000	
Size	1024 KE	~
Default Code Region		
Access Permission	RO by privileged code only	~
Code execution	Permitted	\sim
Shareability	Non-Shareable	~
Cacheability	Non-Cacheable	~

Figure 412. 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.

M32	File	Window	Help			10	f 🕒 X 🔿 🍳	$\times 5$
ome > STM32H7	755XIHx > Untitled - Tools	\rangle					GENERATE CODE	
Pinout &	Configuration	Clock Conf	iguration	Project N	lanager		Tools	
	Memory view	ed by ARM Cortex-M7	Memory viewed by ARM Cortex-M4		Context CortexM7	+	Context CortexM4	+
mpare Projects	0xD4000000 Reserved			Add an applicati	ion region	×		â
	0xD0000000 FMC SDRAM Bank	2 (64 MB)		Name	new_app_reg		N/A	
	0xc0000000 FMC SDRAM Bank			Address	0x0000000		N/A	
	0xA0000000 Reserved			Size		OMB		â
	0x90000000 QuadSPI (256 MB))		Context	CortexM7	0.00	N/A	
	0x80000000 FMC Bank3 (256 N						N/A	
	0x70000000 Reserved			Context sharing	g 🔲 (M7,M4)			â
_	0x60000000 FMC Bank1 (256 N	IB)			Cancel	Add	N/A	
	0x38801000 Reserved			-		â		â
PCC	0x38800000 Backup SRAM (4 H	(B)						
	0x30010000 Reserved					â		â
	0x38000000 AHB SRAM Domai	n 4 (64 KB)					****************	
	0x30048000					â		â
CAD	0x30040000 AHB SRAM3 (32 K	B)						*******

Figure 413. Adding a new region

5. Select "Context sharing (M7, M4)", automatically another region is created with the same name, start address, and size

30048000Reserved	_			
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.



OPENAMP activation when using H7 dual-core:

- 8. OPENAMP activation
 - Configure the NVIC1 and 2 and select their related HSEM global interrupts.
 - Activate the Middleware OPENAMP_M4.
 - MMT will create 2 application regions for each core. The Master regions are defined by attribute mode.

Figure 414. Configure the NVIC1, NVIC2 and select their HSEM global interrupt

NYIC2 Mode and C	onsiguration		
Mode			
Runtime contexts:			
Cortex-M7 Cortex-M4		Power	Domain
5		1	02
	_		
Configuratio	20		
NVIC Ode generation			
Priority Group 💟 🗆 Sort by Premption Priority and	Sub Priority	Sort by	interrupts names
Search (C) Show available interrupts ~		Force D	MA channels Interru
NVIC2 Interrupt Table	Enable	d Preemptio	n Priority Sub Prio
von maskable interrupt	22	0	0
lard fault interrupt	53	0	0
lemory management fault	22	0	0
Pre-fetch fault, memory access fault	51	0	0
Indefined instruction or illegal state	23	0	0
System service call via SWI instruction	53	0	0
Debug monitor	23	0	0
Pendable request for system service	23	0	0
Time base: System tick timer	59	15	0
PVD and AVD interrupts through EXTI line 16		0	0
Flash global interrupt		0	0
CM7 send event interrupt for CM4		0	0
PU global interrupt		0	0
ISEM2 global interrupt	2	0	0
RAM ECC diagnostic global interrupt		0	0
fold core interrupt		0	0

Figure 415. OPENAMP_M7 parameters settings

4		× Ø	OPENA	MP_M7 Mode and Configuration		Pinout view System view
Categories A->Z				Mode		
Middleware and Software Packs		~	Runtime contexts:			
	Cortex-M7	Cortex-M4	Cortex-M7	Cortex-M4	PowerDomain D1	
AIROC-WI-FI-Bluetooth-STM35			-		D1	
FATES_M4		52	Enabled			
FATES_M7	2					
FP-SNS-MOTENVWB1						
E FP-SNS-SMARTAG2						
E FP-SNS-STBOX1						
FREERTOS_M4		2				
FREERTOS_M7	21					~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
- I-CUBE-CANOPEN				Configuration		\$\$\$\$\$\$ \$ \$\$ \$ \$ \$ \$ \$\$\$\$\$\$\$
1-CUBE-Cealum			Reset Configuration			────────────────────────────────────
I-CUBE-FS-RTOS						
1-CUBE-ITTIADB			Parameter Settings User Constants			
L-CUBE-Mangoose			Configure the below parameters :			
🗄 I-CUBE-embOS					0	
I-CUBE-wolfMQTT			Q Search (Ctrl+F) ③ ③		0	
L-CUBE-wol/SSH			~			
I-CUBE-wolfSSL			Instance	OPENAMP_M7		000000000000000000000000000000000000000
I-CUBE-wolfTPM I-Cube-SoM-uGOAL			Version			
			OPENAMP version	v2018.10		
LIBJPEG			Communication Mode			
LWIP			Mode	MASTER		
MBEDTLS			Configuration			TFBGA240 +25 (Top view)
V OPENAMP_M4		2	METAL_MAX_DEVICE_REGIONS	2		in bouche its (rep new)
OPENAMP M7	2	10 11	RPMSG_BUFFER_SIZE	512		
PDM2PCM_M4		53	NUM_RESOURCE_ENTRIES	2		
	5		VRING_COUNT	2		
		2	VDEV_ID	OXFF		
	5	14	VRING0_ID	0		
USB_DEVICE_M7	21					
USB_HOST_M4		12				
	5					0 53 0 55 43
X-CUBE-AI						Q [] Q 🕒 🚄
(X.CURE. M GORUND						



		1 19			lictors setting:	5
Pinout & Conf	iguration		Clock Configuration	1 P	roject Manager	Tools
٩		~	0	OPENAMP_M4 Mode and Configuration		Pinout view System view
Categories A->Z				Mode		
Middleware and Software Packs		~	 Runtime contexts: 			
	_		Cortex-M7	Cortex-M4	PowerDomain	
AIROC-Wi-Fi-Bluetooth-STM32	Cortex-M7	Cortex-M4		2	D2	
FATFS_M4		5	Enabled			
FATFS_M7	5					
FP-SNS-MOTENVWB1						
FP-SNS-SMARTAG2						
FP-SNS-STBOX1						
FREERTOS_M4		1				
FREERTOS_M7	1					
I-CUBE-CANOPEN				Configuration		
🔁 I-CUBE-Cesium			Reset Configuration			<u>୍ର୍କ୍ର୍କ୍ର୍କ୍ର୍ର୍ର୍ର୍କ୍ର୍ର୍କ୍ର୍ର୍କ୍ର୍ର୍</u>
I-CUBE-FS-RTOS						
I-CUBE-ITTIADB			Parameter Settings Subser Constitution	stants		
I-CUBE-Mongoose I-CUBE-embOS			Configure the below parameters :			
I-CUBE-wolfMQTT			Q Search (Ctrl+F) (3)		0	
I-CUBE-wolfSSH			Instance	OPENAMP M4		
I-CUBE-wolfSSL			Version	an ann an San		
I-CUBE-wolfTPM			OPENAMP version	v2018.10		
I-Cube-SoM-uGOAL			Communication Mode			
LIBJPEG			Mode	SLAVE		***************
LWIP			Configuration			
MBEDTLS			METAL_MAX_DEVICE_REGI	ONS 2		TFBGA240 +25 (Top view)
OPENAMP_M4		2	RPMSG_BUFFER_SIZE	512		
✓ OPENAMP_M7	1		NUM_RESOURCE_ENTRIES	2		
PDM2PCM_M4		52	VRING_COUNT	2		
PDM2PCM_M7	1		VDEV_ID	OxFF		
USB_DEVICE_M4		1	VRING0_ID	D		
USB_DEVICE_M7	5		VRING1_ID	1		
USB HOST MA		102				

Figure 416. OPENAMP_M4 parameters settings

Figure 417. The reserved memory regions for OPENAMP using MMT

0x30010000 Heserved	D ·	0	
	OPENAMP	OPENAMP	
AHB SRAM Domain 4 (64 KB) 0x3000000	OPENAMP_RSC_TAB	OPENAMP_RSC_TAB	
extopesood Reserved	e	â	

9. Press the Generate Code button to generate 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.



1	
2	//*###ICF### Section handled by ICF editor, don't touch! ****/ /*-Editor annotation file-*/
4	/*locidation=file='/ /*locidatorile='StockIT_DIRS\config\ide\lofEditor\cortex_vi_0.xml* *//*-Specials-*/
5	//
7 8	(*-Sizes-*/
9 10	define symbolICFEDIT_intvec_start = 0x00000000;
11	define symbolICFEDIT_size_cstack = 0x400;
12 13	define symbolICFEDIT_size_heap = 0x200;
14 15	/*-MMT symbols-Auto-generated By STM32CubeMX*/
16	define symbolICFEDIT_region_ITCHRAM_start = 0x00000000;
17 18	define symbolICFEDIT_region_ITCHRAM_end = 0xFFFF; define symbolICFEDIT_region_RAM_D1_start = 0x24000000;
19	define symbol _ICFEDIT_region_RAM_D1_end = 0x2407FFF; define symbol _ICFEDIT_region_OPENAMP_RSC_TAB_start = 0x38000000; define symbol _ICFEDIT_region_OPENAMP_RSC_TAB_end = 0x3800003FF; define symbol _ICFEDIT_region_OPENAMP_RSC_TAB_end = 0x3800003FF;
20	define symbolICFEDIT_region_OPENAMP_RSC_TAB_startexi30000000; define_symbolICFEDIT_region_OPENAMD_RSC_TAB_endexi3000000;
22	define symbolICFEDIT_region_OPENAMP_start = 0x30000400; define symbolICFEDIT_region_OPENAMP_end = 0x3000FFFF;
23 24	define symbolICFEDIT_region_OPENAMP_end = 8x3800FFFF;
25	/*-End of KX Symbols*/
26	/*-Symbols-*/
2.8	define symbolICFEDIT_region_RAM_start = 0x200000000;
2:9 3:0	define symbolICFEDIT_region_RAM_end = 0x2001FFFF; define symbolICFEDIT_region_FLASH_start = 0x00000000;
31	define symbolICFEDIT_region_FLASH_end = 0x80FFFFF;
32 33	/**** End of ICF editor section. ###ICF###*/
34	define memory mem with size = 46;
35 36	/*-MENCRV Regions-*/
37 38	define region RAM_region = mem:[fromICFEDIT_region_RAM_starttoICFEDIT_region_RAM_end];
39	perine region rubbn_region = mes.[rrum_tureui_region_rubbn_stanttotureui_region_rubbn_r
48 41	/*-MMT Regions-*/
42	define region ITCHRAM_region = mem:[fromICFEDIT_region_ITCHRAM_starttoICFEDIT_region_ITCHRAM_end]; define region RAM_D1_region = mem:[fromICFEDIT_region_RAM_D1_starttoICFEDIT_region_RAM_D1_end];
43	define region OPENAMP_RSC_TAB_region = mem:[fromICFEDIT_region_OPENAMP_RSC_TAB_start toICFEDIT_region_OPENAMP_RSC_TAB_end];
44 45	define region OPENAMP_region = mem:[from _ICFEDIT_region_OPENAMP_start_ to _ICFEDIT_region_OPENAMP_end_];
46 47	/*:End of MMIT Regions-*/
48	/*-Blocks-*/
49 50	<pre>define block CSTACK with alignment = 8, size =ICFEDIT_size_cstack();</pre>
50	define block HEAP with alignment = 8, size =ICFEDIT_size_heap {};
52	/*-Initialization_strategies-*/
53 54	initialize by copy { readwrite }; do not initialize { section .noinit };
55	
56 57	/*-Sections placements-*/ place at address mem:ICFEDIT_intvec_start { readonly section .intvec };
5.8	
59	/*-WHT Sections-*/
61	place in OPENAMP_region (section OPENAMP_section);
62	place in ITCHRAM_region (section ITCHRAM_section); place in RAM_D1_region (section RAM_D1_section);
64	place in RAM_UI_region (section RAM_UI_section); place in OPENAMP_RSC_TAI_region (section OPENAMP_RSC_TAB_section);
65	/*-end of MMT Sections-*/

Figure 418. Linker files update(stm32h755xxx_flash_cm4.icf)



	//*###ICF### Section handled by ICF editor, don't touch! ****/ /*-Editor annotation file-*/
	/* IcfEditorFile="\$TOOLKIT_DIR\$\config\ide\icfEditor\contex_v1_0.xml" *//*-Specials-*/
	/••••••
	/*-Sizes-*/ define symbolICFEDIT_intvec_start = 0x00000000;
	define symbolLCFEDIT_size_cstack = 0x400; define symbolLCFEDIT_size_heap = 0x200;
	/*-MMT symbols-Auto-generated By STH32CubeMX*/ define symbolICFEDIT_region_ITCMRAM_start = 0x00000000; define symbolICFEDIT_region_ITCMRAM_end = 0x7400000; define symbolICFEDIT_region_RAM_D1_end = 0x2407FFF; define symbolICFEDIT_region_OPENAMP_RSC_TAB_start = 0x3800000; define symbolICFEDIT_region_OPENAMP_RSC_TAB_end = 0x3800000; define symbolICFEDIT_region_OPENAMP_RSC_TAB_end = 0x3800000; define symbolICFEDIT_region_OPENAMP_RSC_TAB_end = 0x380000400; define symbolICFEDIT_region_OPENAMP_end = 0x38000400; define symbolICFEDIT_region_OPENAMP_end = 0x3800FFFF;
	/*-End of MX Symbols*/
	/*-Symbols.*// define symbolICFEDIT_region_RAM_start = 0x20000000; define symbolICFEDIT_region_FLASH_start = 0x0000000; define symbolICFEDIT_region_FLASH_end = 0x0000000; define symbol _ICFEDIT_region_FLASH_end = 0x0000000; /**** End of ICF editor section. ###ICF####//
	define memory mem with size = 46;
	/*-HEMORY Regions-*/ define region RAM_region - mem:[from _ICFEDIT_region_RAM_startto _ICFEDIT_region_RAM_end]; define region_FLASH_region - mem:[from _ICFEDIT_region_FLASH_starttoICFEDIT_region_FLASH_end];
	/*-WHT Regions-*/ define region IICHRAM_region = mem:[from _ICFEDIT_region_IICHRAM_startto _ICFEDIT_region_IICHRAM_end_]; define region RAM_Di_region = mem:[from _ICFEDIT_region_RAM_Di_startto _ICFEDIT_region_RAM_Di_end_]; define region OPENAMP_RSC_TAB_region = mem:[from _ICFEDIT_region_OPENAMP_RSC_TAB_startto _ICFEDIT_region_OPENAMP_RSC_TAB_end define region OPENAMP_region = mem:[from _ICFEDIT_region_OPENAMP_startto _ICFEDIT_region_OPENAMP_Startto _ICFEDIT_region_OPENAMP_Startto _ICFEDIT_region_OPENAMP_end_];
L	/*:End of MMI 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-*/ place at address mem:_ICFEDIT_intvec_start { readonly section .intvec };
	/*-MMT Sections-*/ place in OPENAMP_region { section OPENAMP_section }; place in ITCMRAM_region { section ITCMRAM_section }; place in RAM_D1_region { section RAM_D1_section }; place in OPENAMP_RSC_TAB_region { section OPENAMP_RSC_TAB_section };

Figure 419. Linker files update(stm32h755xxx_flash_cm7.icf)

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.



An example of MMT configuration of the ETH IP on the STM32H755XIH6TR MCU

1. Activate the IP ETH:

			Figure	e 420. Co	onfigura	atio	ו ח	ETH	IP				
STM32CubeMX U STM32 Cube MX	ntitled*: STM32H755XIHx	File	Window	Help							® f	D X 🔿	× ₹₹
Home S	тмз2н755хінх >	Untitled - P	inout & Configuration								GEN	IERATE CODE	
	Pinout & Config	uration	Clock	Configuration			Projec	t Manager				Tools	
				Software Packs	✓ Pinout								
Q			· ETH Mode	and Configuration	1				📮 Pinout v	iew 💾 S	ystem view		
Categories A	↓->Z			Mode									
Connectivity		~	Runtime contexts:										
	Cortex-M7	Cortex-M4			erDomain		-						
S ETH					D2		1		(m) (max) (may) (may) (
FDCAN1			Mode MII		~			an (va) (m) (m)		1 de 1	(m) (m) (m)	and an and	
FDCAN2			Activate Rx Err signal					ăăăă	čěč	ăăă	čěč	MAGA	
FMC			Activate Tx Err signal				1			3 3 3			
12C1 12C2													
1202							1) (m) (m) (m)) ((m) (m) (m)		
1204			Cc	nfiguration			1						
LPUART1			Reset Configuration					and frame (m) (m)	(m) (m)	(m) (m)	(10) (10)	AND (100) (100)	
MDIOS				IVIC Settings	GPIO Settings			a a a a		ăăă			
QUADSPI				ameter Settings	GPIO Settings								
SDMMC1			Configure the below parameters :				1			900			
SDMMC2			Q Search (Ctrl+F) () ()		0				(m) (m)((m) (m) (m)	(11) (11)		
SPI1 SPI2			 General : Ethernet Configuration 						(m) (m)	(m) (m) (m)	()		
SP12 SP13			Warning	The ETH can work only w	vhen RAM is poin				in in		(m)	A A A A	
SPI3			Ethernet MAC Address	00:80:E1:00:00:00									
SPI5			Tx Descriptor Length	4			-						
SPI6			First Tx Descriptor Address Rx Descriptor Length	0x30000080 4			-						
SWPMI1			First Rx Descriptor Address	0x30000000				101,5 (K2,8 (M) (M)					
UART4			Rx Buffers Address	0x30000100				an (an (an 🖉		in in in			
UART5			Rx Buffers Length	1524				-			A. 10	AAAA	
UART7							2		0000	000	Con Ca	0000	
UART8									TFBGA	240 + 25 (Toj	p view)		
USART1 USART2													
USART2 USART3													
USART6						Q	53	Q L	a 🕰		Ξ α		~ Uni
USB OTO						~	6.3	~ •					* U III

- MMT is going to create 3 application regions for each Core.

Figure 421. ETH MMT region

x30020000 AHB SRAM2 (128 KB)		
	RxPool	RxPool
AHB SRAM1 (128 KB)	TxDescriptor	TxDescriptor
x30000000	RxDescriptor	RxDescriptor
Reserved		a :

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" then ETH will be partially under MMT control.
- 3. Press the Generate Code button to generate code for both applications.



l igi	✓ 8	Software Packs
ETH	Mode and Configura	ation
	Mode	
Runtime contexts:		
Cortex-M7	Cortex-M4	PowerDomain
		D2
Mode MII		~
Activate Rx Err signal		
Activate Tx Err signal		
	Conferentian	
	Configuration	
Reset Configuration	Configuration	
Reset Configuration	Configuration	
NVIC Settings		GPIO Settings
 NVIC Settings Parameter Setting 	gs	 GPIO Settings User Constants
NVIC Settings	gs	
 NVIC Settings Parameter Setting 	gs	
 NVIC Settings Parameter Setting onfigure the below parameter 	gs rs:	User Constants
 NVIC Settings Parameter Setting onfigure the below parameter Search (Ctrl+F) 	gs (3) (3) Iration	User Constants
 NVIC Settings Parameter Setting onfigure the below parameter Search (Ctrl+F) General : Ethernet Configure 	gs (3) Irration The ETH (User Constants Can work only when RAM i.
 NVIC Settings Parameter Setting onfigure the below parameter Search (Ctrl+F) General : Ethernet Configu Warning 	gs (s: () () (ration The ETH () (ress 00:80:E1:	 User Constants Can work only when RAM i.
 NVIC Settings Parameter Setting onfigure the below parameter Search (Ctrl+F) General : Ethernet Configu Warning Ethernet MAC Address 	gs rs: The ETH ress 00:80:E1: th 4	User Constants Image: Constant s Ima
 NVIC Settings Parameter Setting onfigure the below parameter Search (Ctrl+F) General : Ethernet Configu Warning Ethernet MAC Added Tx Descriptor Lengt 	gs rs: The ETH of ress 00:80:E1: th 4 Address 0x300000	User Constants Image: Constant s Ima
 NVIC Settings Parameter Setting onfigure the below parameter Search (Ctrl+F) General : Ethernet Configu Warning Ethernet MAC Added Tx Descriptor Lengt First Tx Descriptor 	gs rs: The ETH of ress 00:80:E1: th 4 Address 0x300000 th 4	User Constants Image: Constants Image: Constants Can work only when RAM i. 00:00:00 80
 NVIC Settings Parameter Setting onfigure the below parameter Search (Ctrl+F) General : Ethernet Configu Warning Ethernet MAC Added Tx Descriptor Lengt First Tx Descriptor Lengt First Tx Descriptor Lengt 	gs (3) (3) (4) (5) (5) (5) (5) (5) (5) (5) (5) (5) (5	 User Constants Can work only when RAM i. 00:00:00 80 00

Figure 422. 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.



	rigure 425. The defined memories under the inter me (Contex-Mr)
2	//*###ICF### Section handled by ICF editor, don't touch! ****/
3.4	/*-Editor annotation file-*/ /* IcfEditorFile=*\$TOOLKIT_DIR\$\config\ide\IcfEditor\cortex_v1_0.xml* *//*-Specials-*/
5 6 7	/······/
8.9	/*-Sizes-*/ define symbolICFEDIT_intvec_start = 0x00000000;
10 11	define symbolICFEDIT_size_cstack_ = 0x400;
12 13	define symbolICfEDIT_size_heap = 0x200;
56789012345	/*-NMT symbols-Auto-generated By STM32CubeHX*/ define symbolICFEDIT_region_ITCMBAM_start = 0x00000000; define symbolICFEDIT_region_RAM_D1_start = 0x240700000; define symbolICFEDIT_region_RAM_D1_start = 0x30000000; define symbolICFEDIT_region_RADescriptor_start = 0x3000000; define symbolICFEDIT_region_RADescriptor_start
26 27	/*-End of MX Symbols*/
29	/#.Sumbols.#/
30 31 32 33 34	define symbolICFEDIT_region_RAM_start = 0x200000008; define symbolICFEDIT_region_RAM_end = 0x2001FFFF; define symbolICFEDIT_region_FLASM_start = 0x00000008; define symbolICFEDIT_region_FLASM_end = 0x000FFFFF; /**** End of ICF editor section. ###ICF####/
35 36 37	define memory mem with size = 46;
38 39	/*-HEHORY Regions-*/
48	define region RAM_region = mem:[from _ICFEDIT_region_RAM_starttoICFEDIT_region_RAM_end_]; define region FLASH_region = mem:[from _ICFEDIT_region_FLASH_starttoICFEDIT_region_FLASH_end_];
423445678	<pre>/*-WMT Regions-*/ define region ITCHRAM_region = mem:[from _ICFEDIT_region_ITCHRAM_start to _ICFEDIT_region_ITCHRAM_end_]; define region RAM_D0_region = mem:[from _ICFEDIT_region_RAM_D1_start to _ICFEDIT_region_RAM_D1_end_]; define region RADescriptor_region = mem:[from _ICFEDIT_region_RADescriptor_start_ to _ICFEDIT_region_RADescriptor_end_]; define region RADescriptor_region = mem:[from _ICFEDIT_region_RADescriptor_start_ to _ICFEDIT_region_RADescriptor_end_]; define region RADescriptor_region = mem:[from _ICFEDIT_region_RADescriptor_start_ to _ICFEDIT_region_RADescriptor_end_]; define region RADescriptor_region = mem:[from _ICFEDIT_region_RADescriptor_start_ to _ICFEDIT_region_RADescriptor_end_];</pre>
49	/*-End of MMT Regions-*/
51 52 53 54	/*-Blocks-*/ define block CSTACK with alignment = 8, size =ICFEDIT_size_cstack {}; define block HEAP with alignment = 8, size =ICFEDIT_size_heap {};
55 56 57 58	/*-Initialization_strategies-*/ initialize by copy { readwrite }; do not initialize { section .noinit };
59 60 61 62	/*-Sections placements-*/ place at address mem:ICFEDIT_intvec_start { readonly section .intvec };
63 64 65 66 67 68 69 70	/*-NMT Sections-*/ place in RxDescriptor_region { section RxDescriptor_section }; place in TxDescriptor_region { section TxDescriptor_section }; place in TrORAM_region { section TXDRAM_section }; place in RxDPool_region { section RxDD section }; place in RxPool_region { section RxDool_section }; /*-end of MMT Sections-*/
72	/*-user Sections-*/
73 74	/-user sections-// place in RAM_region { readonly }; place in RAM_region { readwrite, block HEAP, block CSTACK };

Figure 423. The defined memories under the linker file (Cortex-M7)



	Figure 424. The defined memories under the linker file (Cortex-M4)
4 11 14 15	//*###ICF### Section handled by ICF editor, don't touch! ****/ /*-Editor annotation file-*/ /* IcfEditorFile="\$TOOLKIT_DIR\$\config\ide\IcfEditor\cortex_v1_0.xml" *//*-Specials-*/
6 7 8 9	/*-Sires-*/ define symbolICFEDIT_intvec_start = @x08100000;
10 11 12 13	define symbolICFEDIT_size_cstack = 0x400; define symbolICFEDIT_size_heap = 0x200;
15 16 17 18 19 20 21	<pre>/*-MMT symbols-Auto-generated By STM32CubeMX*/ define symbolICFEDIT_region_RKDescriptor_start = 0x30000000; define symbolICFEDIT_region_RKDescriptor_end = 0x300000000; define symbolICFEDIT_region_TXDescriptor_end = 0x300000000; define symbolICFEDIT_region_RKPool_start = 0x300000000; define symbolICFEDIT_region_RKPool_end = 0x30000000;</pre>
23 24 25 26 27 28 29 30 31	/*-End of MX Symbols*/ /*-Symbols.*/ define symbolICFEDIT_region_RAM_start = 0x10000000; define symbolICFEDIT_region_RAM_end = 0x00100000; define symbolICFEDIT_region_FLASH_tart = 0x00100000; define symbolICFEDIT_region_FLASH_end = 0x01FFFFF; /**** End of ICF editor section. ###ICF###*/
32 33 34 35 36 37	define memory mem with size = 4G; /*-HENOKY Regions-*/ define region RAM_region = mem:[from _ICFEDII_region_RAM_startto _ICFEDII_region_RAM_end _]; define region FLASH_region = mem:[from _ICFEDII_region_FLASH_startto _ICFEDII_region_FLASH_end _];
38 39 40 41 42	/*-WMT Regions-*/ define region RxDescriptor_region = mem:[fromICFEDIT_region_RxDescriptor_start toICFEDIT_region_RxDescriptor_end_]; define region TxDescriptor_region = mem:[fromICFEDIT_region_TxDescriptor_start toICFEDIT_region_TxDescriptor_end_]; define region RxPool_region = mem:[fromICFEDIT_region_RxPool_start toICFEDIT_region_RxPool_end_];
43	/*-End of NNT Regions-*/
45 46 47 48	/*-Blocks-*/ define block CSTACK with alignment = 8, size =ICFEDIT_size_cstack {}; define block HEAP with alignment = 8, size =ICFEDIT_size_heap {};
49 50 51 52	/*-Initialization strategies-*/ initialize by copy { readwrite }; do not initialize { section .noinit };
53 54 55	/*-Sections placements-*/ place at address mem:_ICFEDIT_intvec_start { readonly section .intvec };
57 58 59 60 61 62	/*-NMMT Sections-*/ place in RxDescriptor_region { section RxDescriptor_section }; place in RxDescriptor_region { section TxDescriptor_section }; place in RxPool_region { section RxPool_section }; /*-end of MMT Sections-*/
64 65 66	/*-user Sections-*/ place in FLASH_region { readonly }; place in RAM_region { readwrite, block HEAP, block CSTACK };

Figure 424. The 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	0
Apply Application Regions Settings to Linker Files: OFF	0



CORTEX_M	17 Mode and Configuration
	Configuration
	a and a second
Reset Configuration	
Parameter Settings O User Constants	
Configure the below parameters :	
Q Search (Ctrl+F) ① ①	0
 Speculation default mode Settings 	
Speculation default mode	Disabled
Cortex Interface Settings	
CPU ICache	Disabled
CPU DCache	Disabled
Cortex Memory Protection Unit Control Set	ttings
MPU Control Mode	Background Region Privileged accesses only + M.
Cortex Memory Protection Unit Region 0 S	ietti
MPU Region	Enabled
MPU Region Base Address	0x08000000
MPU Region Size	512KB
MPU SubRegion Disable	OxO
MPU TEX field level	Invel 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 S	letti
MPU Region	Disabled
Cortex Memory Protection Unit Region 2 S	ietti
MPU Region	Disabled
Cortex Memory Protection Unit Region 3 S	letti
MPU Region	Disabled

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

UM1718 Rev 46

Δ7/

A

0

0

0

0

6

Only "Boot" and "Appli" contexts are managed by the MMT. Each context has its own application region (AppReg0 and AppReg1, respectively).

Figure 426. Default settings

			g_	
Home > S	TM32U535CBTx 🔰 Untitled - Tools 🔪 UPBAM Scenario & Corf	iguatan 🔪	GENERATE CODE	CHECK LPBAM DESIGN
	Pinout & Configuration	Clock Configuration	Project Manager	Tools
	Code Generation Configuration	Memory viewed by ARM Cortex-863	Application Regions	+ 1
	Apply application Regions settings to peripherals: OFF OFF OFF OFF	(valioscope Reserved		â.
Memory		0x80000000 OctoSP11 (256 MB)	NA	0
Management	Search an Application Region	Custopacco		ê ;
	Show selected Application Region	0x50036400 Backup SRAM (S) (2 KB)		
		0xe005ccoReserved		<u>iii</u>
		0x40036430 Backup SRAM (NS) (2 KB)		
		0x30104400 Reserved SRAMA Smart Run Domain (S) (15 KB)		Display Settings Reset View
		Chevron Contract Cont		
PCC		Cadobaccos SRAM2 (S) (64 KB)	RAM (NS) Reserved Alias Region	Hide security beside regions names
		All states and a state state	RAM (S)	Hide Reserved Alased regions names
		ce3000000 Sevent (5) (12 Ke) 0x2004000 Reserved		
		0420004000 0420000000 SRA4M Smart Run Domain (NS) (16 KB)		
		0x27040000 Reserved		ê.
		0x20030000 SRAM2 (NS) (64 KB)	RAM (NS)	
CAD		0x20000000 SRAM1 (NS) (192 KB)	RAW (S) Reserved Alias Region	ê :
		Ox10240000 Reserved		ê ;
		CatED30000 SRAM2 Code (S) (54 KB)	RAM (NS) Reserved Alias Report	a ;
		CatEDOCCCO SRAM1 Code (S) (192 KB)	RAM (S) Reserved Alas Region	â ;
		fastcs2coce Reserved		û 🗸
			FLASH (NS) Reserved Alias Region	ê.
		Flash Bank1 memory (5) (128 KB)	FLASH_NSC (NSC)	
		Cm0000000	PLASH (S)	
		Oxta040000 Reserved		iii Legend
		OstA030000 SRAMZ Code (NS) (64 KB)	RAM (NS) Reserved Alias Region	ů ;
		Cardaccourd SRAM1 Code (NS) (152 KB)	RAM (S) Reserved Alas Region	Region allowing different types of security.
		Oxtenzooor Reserved	0 MM AND	Secure region (S).
		Pash Bank1 memory (NS) (128 KB)	FLASH (NS) FLASH NSC (S) Reserved Alias Repon	Non Secure region (VS).
			FLASH (S) Reserved Alias Region	Non Secure Callable region (NSC).
		0x8000000	- A real for second some selfan.	
		0x80000000 Missened		Region accessible by Secure and Non-Secure.
				🛱 Reserved region.
				A new application region can be added here.
				NA IP Configuration is expected. See tooltip into far further details.

User interface

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 Gbytes 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).

New Project from a MCU/MPU									
CU/MPU Selector Board Selector Exa	ample Selector	Cross	Selector						
CU/MPU Filters)							
★ 🗟 🛱 ১			Features Block Diagram	Docs & Resources	CAD Resources	latasheet 📑 E	Buy 🕞 Start Project		
Commercial Part Number	\sim	☆	STM32H7 Series	-					
۹ 🔍 🗸	+-		STM32H7R3A8I6	Flash memory, 620 KE	I DSP with DP-FPU, Arm 8 SRAM, 550 MHz CPU, L faces, USB High Speed I	1 cache, graphic	accelerations,		
PRODUCT INFO	~			Unit Price for 10kU (US\$): NA	-				
Segment	>		COMING SOON Stay tuned !			UFBGA 169 7x7x0	.6 P 0.5 mm		
Series	>								
Line	~		High-performance and DSP with graphic accelerations, external				50 MHz CPU, L1 cache,		
			graphic accelerations, external	memory interfaces, USB High a	speed PHT and large set of pe	eriprierais			
Check/Uncheck All (2)	Check/Uncheck All (2)								
			Easturee						
	A		Features						
	Aa [ab]		• N/A						
STM32H7B0 Value line	Aa (ab)								
STM32H7B0 Value line	Aa (ab)	MCU	• N/A						
	Aa [əb]	мси					🕂 Export		
STM32H7R3/7S3	Aa (ab)	MCU	N/A s/MPUs List: 34 items Commerce	ial Part No 🔶 🔰 Part N		ADC 12-bit (c	converters) × Marketing Status		
 STM32H7R3/7S3 STM32H7R7/7S7 	Aa [ab]	мси	• N/A s/MPUs List: 34 items	A8I6 STM32H7R3A8	STM32H7R3A8lx	0	converters) × Marketing Status Coming soon		
STM32H7R3/7S3 STM32H7R7/7S7	Aa [ab]	MCU	NA s/MPUs List: 34 items Commerce STM3247R33 ☆ STM3247R33	A8I6 STM32H7R3A8 8K6 STM32H7R3I8	STM32H7R3A8lx STM32H7R3I8Kx	0	Converters) X Marketing Status Coming soon Coming soon		
STM32H7R3/7S3 STM32H7R7/7S7 STM32L0x0 Value Line STM32L0x1 STM32L0x1 STM32L0x2	Aa (ab)	MCU	• N/A s/MPUs List: 34 items • Commerc ☆ STM32H7R33 ☆ STM32H7R3 ☆ STM32H7R3	A8I6 STM32H7R3A8 8K6 8T6 STM32H7R3I8	STM32H7R3A8lx STM32H7R3I8Kx STM32H7R3I8Tx	0 0 0	converters) X Marketing Status Coming soon Coming soon Coming soon		
 STM32H7R3/7S3 STM32H7R7/7S7 STM32L0x0 Value Line STM32L0x1 STM32L0x2 STM32L0x3 	Aa (ab)	MCU	• NA s/MPUs List: 34 items ☆ STM32H7R3/ ☆ STM32H7R3/ ☆ STM32H7R3/ ☆ STM32H7R3/	A8I6 STM32H7R3A8 8K6 STM32H7R3I8 8T6 STM32H7R3I8 .8H6 STM32H7R3I 8	STM32H7R3A8Ix STM32H7R3I8Kx STM32H7R3I8Tx STM32H7R3L8Hx	0 0 0 2	Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon		
STM32H7R3/7S3 STM32H7R7/7S7 STM32L0x0 Value Line STM32L0x1 STM32L0x2 STM32L0x2 STM32L0x3 STM32L100 Value Line	Aa [ab]	MCU	NA SIMPUS List: 34 items Commerce Commerce SIM32HTR34 Commerce SIM32HTR31 SIM32HTR31	A8I6 STM32H7R3A8 8K6 STM32H7R3I8 8T6 STM32H7R3I8 8H6 STM32H7R3L8	STM32H7R3A8Ix STM32H7R3I8Kx STM32H7R3I8Tx STM32H7R3I8Tx STM32H7R3L8Hx STM32H7R3L8HxH	0 0 2 2	converters) × Marketing Status Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon		
2 STM32H7R3/7S3 2 STM32H7R3/7S7 STM32L0x0 Value Line STM32L0x1 STM32L0x2 STM32L0x3 STM32L100 Value Line STM32L151/152	Aa [ab]	MCU	• NA • Commerc ☆ STM32H7R3	A886 STM32H7R3A8 8K6 STM32H7R38 8T6 STM32H7R38 .8H6 STM32H7R3L8 .8H6H STM32H7R3R8	STM32H7R3A8ix STM32H7R3i8Kx STM32H7R3i8Tx STM32H7R3i8Tx STM32H7R3L8Hx STM32H7R3L8HxH STM32H7R3R8Vx	0 0 2 2 0	converters) X Marketing Status Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon		
STM32H7R3/7S3 STM32H7R7/7S7 STM32L0x0 Value Line STM32L0x1 STM32L0x2 STM32L0x2 STM32L0x3 STM32L100 Value Line	Aa (ab)	MCU	NVA SIM2VIS List: 34 items ☆ STIM32HTR3	A886 STM32H7R3A8 8K6 STM32H7R38 8T6 STM32H7R38 .8H6 STM32H7R3L8 .8H6H STM32H7R3R8 .8H6H STM32H7R3R8 .8H6H STM32H7R3R8	STM32H7R3A8bx STM32H7R318bx STM32H7R318bx STM32H7R3L8Hx STM32H7R3L8HxH STM32H7R3L8HxH STM32H7R3R8Vx STM32H7R3V8Hx	0 0 2 2 0 0	converters) >>> Marketing Status Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon		
2 STM32H7R3/7S3 2 STM32H7R3/7S7 STM32L0x0 Value Line STM32L0x1 STM32L0x2 STM32L0x3 STM32L100 Value Line STM32L151/152	Aa [ab]	MCU	NA Continerc Continerc Continerc Continerc SIM32H783 C SIM32H783 SIM34H783 SIM32H783 SIM3	A886 STM32H7R3A8 8K6 STM32H7R3A8 8K6 STM32H7R3I8 8H6 STM32H7R3L8 88H6H STM32H7R3L8 88V6 STM32H7R3R8 V8H6 STM32H7R3V8	STM32H7R3A8bx STM32H7R3I8Kx STM32H7R3I8Tx STM32H7R3L8Hxh STM32H7R3L8Hxh STM32H7R3R8Vx STM32H7R3X9Hx STM32H7R3V8Hx	0 0 2 2 0 0 0 0	Converters) Marketing Status Corning soon Corning soon Corning soon Corning soon Corning soon Corning soon Corning soon Corning soon Corning soon		
2 STM32H7R3/7S3 2 STM32H7R7/7S7 - STM32Lox0 Value Line - STM32Lox1 - STM32Lox3 - STM32Lox3 - STM32L150 Value Line - STM32L151/152 - STM32L162	Aa (ab)	MCU	N/A N/A SIM2015 List: 34 items	A886 STM32H7R3A8 BK6 STM32H7R318 BT5 STM32H7R318 BH6 STM32H7R318 L8H6H STM32H7R318 V8H6 STM32H7R318 V8H6 STM32H7R318 V8H6 STM32H7R318 V8H6 STM32H7R318 V8H6 STM32H7R318 V8H6 STM32H7R318	STM32H7R3A8bx STM32H7R306Xx STM32H7R306Xx STM32H7R308tx STM32H7R3L8HxH STM32H7R3L8HxH STM32H7R3V8Hx STM32H7R3V8Hx STM32H7R3V8Hx STM32H7R3V8Hx STM32H7R3V8Hx	0 0 2 2 0 0 0 0	Converters) X Marketing Status Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon		
STM32H7R3/7S3 STM32H7R7/7S7 STM32L0x2 STM32L0x2 STM32L0x2 STM32L0x2 STM32L0x3 STM32L0x3 STM32L160 STM32L150/152 STM32L150/152 STM32L162	Aa [əb]	MCU	NA Continerc Continerc Continerc Continerc SIM32H783 C SIM32H783 SIM34H783 SIM32H783 SIM3	A886 STM32H7R3A8 BK6 STM32H7R318 BT5 STM32H7R318 BH6 STM32H7R318 L8H6H STM32H7R318 V8H6 STM32H7R318 V8H6 STM32H7R318 V8H6 STM32H7R318 V8H6 STM32H7R318 V8H6 STM32H7R318 V8H6 STM32H7R318	STM32H7R3A8bx STM32H7R3I8Kx STM32H7R3I8Tx STM32H7R3L8Hxh STM32H7R3L8Hxh STM32H7R3R8Vx STM32H7R3X9Hx STM32H7R3V8Hx	0 0 2 2 0 0 0 0	converters) × Marketing Stat Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon Coming soon		

Figure 427, 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 428. Initialization dialogue							
Memory Protection Unit for Cortex-M7 ×								
?	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							

Figure 428 Initialization dialogue

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.



Ao2 CORTE_VR_BOOT Mode and Configuration System Core Oo * Boot Application ExternalMemoryLoader * Boot Contract, MI, Dory Boot Application ExternalMemoryLoader * * Boot Contract, MI, Dory Boot	Image: Section		Pinout & Configuration			Clock Configuration	
State CONTEX_MU_BOOT Mease and Refiguration space Particles contracts * Bet Application * * Bet * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * <t< th=""><th>vite COTE UM Data yten Cos ************************************</th><th></th><th>Pinout & Configuration</th><th></th><th></th><th></th><th>ke N Disout</th></t<>	vite COTE UM Data yten Cos ************************************		Pinout & Configuration				ke N Disout
Spream Spream Spream Period ************************************	general ytan Cos * Dott Application Control of Joont Control of Joont Plottin Bat Plottin Plottin <th></th> <th></th> <th></th> <th><u></u></th> <th></th> <th></th>				<u></u>		
Bot Application Control (MA) (DOI 0 Destination (Application Bot Application Bot Application Control (MA) (DOI 0 Destination (MA) (DOI 0 Control (MA) (DOI 0 0	Application Application Bot Application Control of a point point of a poin			~	ø	CORTEX_M7_	BOOT Mode and Configuration
Bot Application ExternalMemoryLaster CORREL_04_000 0 <td>Optimum Optimum Application ExternalAtmonyLaster * Dot Application ExternalAtmonyLaster // Advin Image Image Image Image // Advin Image Image Image Image Image // Advin Image Image</td> <td>aregones Araz</td> <td></td> <td></td> <td></td> <td>Destination of the second seco</td> <td>Nioue</td>	Optimum Optimum Application ExternalAtmonyLaster * Dot Application ExternalAtmonyLaster // Advin Image Image Image Image // Advin Image Image Image Image Image // Advin Image	aregones Araz				Destination of the second seco	Nioue
Control MA APPU Image Image ALAY Image Image ALAY Image Image APDAAL Image Image Attimedia Imagee Attimage Imagee <td>Control MARY Image Image AAAN Image Image AANN Image Image AANN Image Image AANN Image Image AAANN Image Image AAAN Image Image AAAN Image Image AAAN Image Imagee AAAN Imagee Imagee AAANN Imagee Imagee Amany Imagee Imagee Am</td> <td>System Core</td> <td></td> <td></td> <td>~</td> <td></td> <td>Application ExternalMemoryLoader</td>	Control MARY Image Image AAAN Image Image AANN Image Image AANN Image Image AANN Image Image AAANN Image Image AAAN Image Image AAAN Image Image AAAN Image Imagee AAAN Imagee Imagee AAANN Imagee Imagee Amany Imagee Imagee Am	System Core			~		Application ExternalMemoryLoader
Control My (psort Image: Control My (psort Image: Control My (psort CPDAN1 Image: Control My (psort Image: Control My (psort CPDAN1 Image: Control My (psort Image: Control My (psort CPDAN1 Image: Control My (psort Image: Control My (psort CPDAN1 Image: Control My (psort Image: Control My (psort CPDAN1 Image: Control My (psort Image: Control My (psort Control My (psort Image: Control My (psort Image: Control My (psort Control My (psort Image: Control My (psort Image: Control My (psort Control My (psort Image: Control My (psort Image: Control My (psort Control My (psort Image: Control My (psort Image: Control My (psort Control My (psort Image: Control My (psort Image: Control My (psort Control My (psort Image: Control My (psort Image: Control My (psort Control My (psort My	Control of Dool Dool	0	Boot	Application	ExternalMemoryLoader	2	
Avan OPDAM1 OPDAM1 OPDAM1 OPDAM1 OPDAM1 OPDAM1 OPDAM1 OPDAM1 States Sompting Concepting States	AAAP OPQAHA OPQ OPQAHA OPQ OPQ OPQAHA OPQ OPQAHA <	CORTEX_M7_APPLI		1			
OPDAN1 Image Image <t< td=""><td>OrDAN Image <th< td=""><td>CORTEX_M7_BOOT</td><td>2</td><td></td><td></td><td></td><td></td></th<></td></t<>	OrDAN Image Image <th< td=""><td>CORTEX_M7_BOOT</td><td>2</td><td></td><td></td><td></td><td></td></th<>	CORTEX_M7_BOOT	2				
OPGO Image MPDAA1 Image MOC_APPU Image MOC_APPU Image MOC_APPU Image SYS Images Sys Images Sys Images Sys Images Sys Images Sys Images	orog	FLASH					
iPOAN1 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	iPUDAti I MOC I MOC,BOT I RCC I RCC I Sig I INMO,ROT I RCC I Sig I Inters I Someticity I terms I control I Someticity I terms I control I Someticity I terms I control I Someticity I terms I <	GPDMA1					
MOC Image Imagee <	ANO NOC, APPL NOC, APPL NOC, APPL NOC, APPL NOC, APPL RCC BYBO BYBO Immain Immai	GPIO					
MVC, BOC BC BCC B	NOC, APU_ Image NOC, BOOT Image Active Image	HPDMA1		¥			
NVE.goor B Image	INVE.BOOT BCC BYOO BYOO </td <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>						
NVE.goor B Image	INVE.BOOT BCC BYOO BYOO </td <td>NVIC_APPLI</td> <td></td> <td></td> <td></td> <td></td> <td></td>	NVIC_APPLI					
BC BC BC BC BC BC BC Contentially Contentially <t< td=""><td>BCC 999 1000 <!--</td--><td></td><td>2</td><td>_</td><td></td><td></td><td></td></td></t<>	BCC 999 1000 </td <td></td> <td>2</td> <td>_</td> <td></td> <td></td> <td></td>		2	_			
BY9 WNOO WNOO WNOO Walag Stans Somerskip Jonnedsig Security Somerskip Jonnedsig Somerskip Somerskip Somerskip Somerskip Jonnedsig Somerskip Jonnedsig Somerskip Jonnedsig Somerskip Jonnedsig Somerskip Somerskip Jonnedsig Somerskip Jonnedsig Somerskip Somerskip <td< td=""><td>0 0 0 0 nakig </td><td></td><td></td><td></td><td></td><td></td><td></td></td<>	0 0 0 0 nakig						
Inno	integendent image national image image image image<						
hvaig >> hv	nalig >> >> >> >> >> >> >> >> >> >> >> >> >>						
imers > imers > intersela > intersela<	imes > imes > <td< th=""><th></th><th></th><th></th><th></th><th></th><th></th></td<>						
imers > imers > intersela > intersela<	imes > imes > <td< td=""><td></td><td></td><td></td><td></td><td></td><td></td></td<>						
Zomeskily >> Security >> Security >> Computing >> Addeeare and Sobure Packs >> Frace and Sobure Packs >> Power and Thomail >> Attline deal >> States and Sobure Packs >> Prover and Thomail >> Attline deal >> States and Sobure Packs >> <	warrante Security Security <td< td=""><td>inalog</td><td></td><td></td><td>></td><td></td><td></td></td<>	inalog			>		
Addimedia Addimedia Security	winneda >> security >> security >> tide/ware and Software Packs >> ace and Debog >> towar and Thermal >> tibes >> tiber >> tiber >> tiber >> tiber >> View Contribute Databled Cottops tiber Addets Cottops tiber Addets tiber >> View Contribute Databled Cottops tiber Addets Cottops tiber Addets MEN (Region Background Region Phileged accesses only + MPU Dial MEN (Region Background Region Phileged accesses only + MPU Dial MEN (Region Background Region Phileged accesses only + MPU Dial MEN (Region Background Region Phileged accesses only + MPU Dial MEN (Region Background Region Phileged accesses only + MPU Dial MEN (Region Background Region Phileged accesses only + MPU Dial MEN (Region Background Region Phileged accesses only + MPU Dial MEN (Region Background Region Phileged accesses only + MPU Dial MEN (Region Baccesse accesse acce	imers			>		
Addimedia Addimedia Security	winneda >> security >> security >> tide/ware and Software Packs >> ace and Debog >> towar and Thermal >> tibes >> tiber >> tiber >> tiber >> tiber >> View Contribute Databled Cottops tiber Addets Cottops tiber Addets tiber >> View Contribute Databled Cottops tiber Addets Cottops tiber Addets MEN (Region Background Region Phileged accesses only + MPU Dial MEN (Region Background Region Phileged accesses only + MPU Dial MEN (Region Background Region Phileged accesses only + MPU Dial MEN (Region Background Region Phileged accesses only + MPU Dial MEN (Region Background Region Phileged accesses only + MPU Dial MEN (Region Background Region Phileged accesses only + MPU Dial MEN (Region Background Region Phileged accesses only + MPU Dial MEN (Region Background Region Phileged accesses only + MPU Dial MEN (Region Baccesse accesse acce	Connectivity			、 、		
Beconty >> Computing >> Addefeare and Solvare Packs >> Frace and Debug >> Power and Thomail >> Altifies >> State and Debug >> Other Of Brannelles Sellings) ● User Constants Constant Construction >> Altifies >> State and Debug >> Other >> Other Constant Construction User Constants >> Other Constant Construction Constru	ecuity >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	Jonnectivity					
Configuration Configuration AddRever and Schware Packs Configuration AddRever and Schware Packs Configuration Trace and Debug Configuration Power and Thermal Configuration Nities Configuration Configuration Configuration <t< td=""><td>moduling >> diddeware and Software Packs >> race and Debug >> twer and Thermal >> billes >> ther >> ther >> there >> <</td><td>Aultimedia</td><td></td><td></td><td>></td><td></td><td></td></t<>	moduling >> diddeware and Software Packs >> race and Debug >> twer and Thermal >> billes >> ther >> ther >> there >> <	Aultimedia			>		
Alleleware and Solvare Packs >> Crace and Debug >> Dower and Thermal >> Alleles >> Dither >> Other >>	Constraint Constraint Modeware and Software Packs >> Size and Debug >> wer and Themail >> Nillers >> Nillers >> Nillers >> Nillers >> View Constant Constant Weight parameters : Constant Mode Settings Constant Constant Constant Mode Settings Constant Constant Mode Settings Constant Constant Mode Settings Constant CPU Octach Dasaled CPU Octach Dasaled CPU Octach Dasaled MPU Region Dister Address Od MPU Region Dister Address Od MPU Region Dister Address Odd MPU Regio	Security			>		
Addeeare and Settware Packs > frace and Debug > Power and Thermal > Ablies > Statistics > Other and Thermal > Ablies > Statistics > Statistics > Other and Thermal > Ablies > Statistics Securitics default mode Settings GPU Catche Disabled OCT Catter Memory Protection Uke Region Disabled Onto MPU Region Base Address 0-0 MPU Region Base Address 0-0 MPU Region Disabled 0-0 MPU Region Disabled 0-07 MPU Region Disabled 0-07 MPU Region Disabled 0-01 MPU Region Disabled<	tidevare and Software Packs >> crea and Debug >> crea and Debug >> crea and Debug >> tities >>> tities >>> tities >>> tities >>> tities >>>> tities >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	Computing			>		Configuration
Waver and Thermal >> Waver and Thermal >> Nikes >> Status >> Operation and their production of the	Second control Context relation addatum rode straining Vexer and Thermal >> Vexer and Thermal <td>Addleware and Software Packs</td> <td></td> <td></td> <td>></td> <td></td> <td></td>	Addleware and Software Packs			>		
Were and Themal >> Atters >> Atters >> Other Or (answer/Citre) Or (answer/Citre) Station >> Other Or (answer/Citre) Or (answer/Citre) Station >> Other Or (answer/Citre) Or (answer/Citre) Other Or (answer/Citre) Or (answer/Citre) Or (answer/Citre) Or (answer/Citre) Or (answer/Citre) Other Or (answer/Citre) Or (answer/Citre) Or (answer/Citre) Or (answer/Citre) Or (answer/Citre) <t< td=""><td>were and Themal >> billes >> constantification data mode Settings Databled coll Cache Databled Background Region Philleged accesses only + MPU Disal MPU Region Base Address 0:0 MPU Region 0 Settings MPU Region Base Address 0:0 MPU Region 0 Settings MPU Region Base Address 0:0 0:0 MPU Region Base Address 0:0 0:0 MPU Region Base Address 0:0</td><td>race and Debug</td><td></td><td></td><td>></td><td></td><td></td></t<>	were and Themal >> billes >> constantification data mode Settings Databled coll Cache Databled Background Region Philleged accesses only + MPU Disal MPU Region Base Address 0:0 MPU Region 0 Settings MPU Region Base Address 0:0 MPU Region 0 Settings MPU Region Base Address 0:0 0:0 MPU Region Base Address 0:0 0:0 MPU Region Base Address 0:0	race and Debug			>		
All Constraints Constraints </td <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>							
Attilde > > Separation default mode Enabled 20ber > > Otark Netdice Settings Diaskied QPU Cache Diaskied QPU Cache Diaskied V Cotex Memory Protection Uet Coterd Settings MPU Cache Diaskied V Cotex Memory Protection Uet Region 0 Settings MPU Cache Diaskied V Cotex Memory Protection Uet Region 0 Settings MPU Cache Diaskied V Cotex Memory Protection Uet Region 0 Settings MPU Region Base Address Diad MPU Region Base Address MPU Region Base Address MPU Region MPU Region Base Address Diad MPU Region Base Address Diad MPU Region Base Memory Protection Uet Region 0 Settings MPU Region Base Head MPU Region Base Address Diad Diad	states > States Enabled ther > Cotex tetrices Settings CPU Ocche DisaMed CPU Ocche DisaMed Stateground Region Philogend accesses only + MPU Disa MPU Region Ost WPU Center Mode Background Region Philogend accesses only + MPU Disa MPU Region Disa Modes DisaMed Background Region Philogend accesses only + MPU Disa MPU Region Disate Address 0:00 MPU Region Disate Address 0	rower and Thermal			>		•
Date Cates Restricts Settings Cates Restricts Settings CULCache Cates Restricts Settings Disabled CPU Dache Cates Restricts Settings Disabled Cotter Mercy Precision UR Control Settings Eaclground Region Phileged accesses only + MPU Disabled Cotter Mercy Precision UR Control Settings Eaclground Region Phileged accesses only + MPU Disabled Cotter Mercy Precision UR Control Settings Eaclground Region Phileged accesses only + MPU Disabled V Cotter Mercy Precision UR Region Settings Eaclground Region Phileged accesses only + MPU Disabled MPU Region Base Adverse MPU Region Base Adverse MPU Region Disabled MPU Region Disabled 0x47 MPU Region Disabled MPU Region Disabled Intel 0 MPU Region Base Adverse Disabled MPU Region Disable MPU Region Size MPU Region Size MPU Region Restrict MPU Region Phileged accesses only + MPU Disabled MPU Region Disabled MPU Region Size MPU Region Restrict MPU Restrict	ther > Cotte, Interface Statuto Enabled ther Cotte, Interface Statuto Disabled CPU Occuba Disabled Cotte Interface Status MEU Control Mode MPU Region Disabled Enabled MPU Region Disabled Cottes	Jtilities			>		
Anni CPU Cache CPU Cache CosaMed CPU Cache CosaM	other CPU Ocche Disabled CPU Ocche Disabled CPU Ocche Disabled V Cottes Merroy Protection Unit Control Settings MPU Desited Background Region Privileged accesses only + MPU Disabled V Cottes Merroy Protection Unit Region 0 Settings MPU Region 0 Settings Enabled MPU Region 0 Settings MPU Region 0 Settings MPU Region 0 Settings MPU Region 0 Sets 408 MPU Region Disabled Oxf7 MPU Region Disabled Oxf7 MPU Region Disabled Not MPU Region Disabled DISABLE MPU Region Disabled OSABLE V Cottes Merroy Protection Unit Region 2 Settings MPU Region Disabled						Enabled
COTCAME DisaMed V< Cottex Memory Protection Unk Control Statings	COTOCATH Disabled V - Cotest Memory Protection Unit Control Settings Background Region Philliped accesses only + MPU Disable V - Cotest Memory Protection Unit Region 0 Settings Enabled MPU, Region Save Got MPU, Region Save 4GB MPU, Region Save 4GB MPU, Region Rave AGB MPU, Region Rave Disabled MPU, Region Rave Disabled MPU, Destable Premission DISABLE MPU, Destable Premission DISABLE MPU, Region Disabled V Cottest Memory Protection Unit Region 2 Settings	Other			>		Direction
MMPU Control Mode Background Region Printleged accesses only + MPU Databact Cottex Memory Protection Ush Region 0 Settings MPU Region Enabled MPU Region Base Addess 0x0 MPU Region Base Addess 0x0 MPU Region Base Addess 0x0 MPU Subjection Databa 0x0 MPU Subjection Databa 0x0 MPU Subjection Databas 0x0 MPU Subjection Databas 0x0 MPU Subjection Databas 0x10 MPU Subjection Databas MPU Subjection Databas 0x14 MPU Subjection Databas 0x14 MPU Subjection Databas 0x14 MPU Subjection Databas DSABLE MPU Subjection Database Permission DSABLE WPU Subjection Database Permission DSABLE WPU Subjection Database Permission Disabled	MPU Control Mode Background Ragion Philisged accesses only + MPU Disal Cottex Memory Philischen Unk Ragion Settings MPU Region MPU Region Settings MPU Region Set Advess Od MPU Region Set Advess Od MPU Region Set Advess Od MPU Region Data/te Odd7 MPU Efficie Ideal Ideal Point Ideal Point Ideal Point Ideal Point Ideal Point Ideal Point MPU Region Caccess Primission Ideal Point <li< td=""><td></td><td></td><td></td><td></td><td></td><td>Utsatiled</td></li<>						Utsatiled
Cottes Manapy Protection Unit Region 0 Settings MPU Region Enabled MPU Region Base Address 0x0 MPU Region Datable MPU Region Datable MPU Standard Howel Inen0 MPU Access Permission MARLE MPU Access Permission DISABLE MPU Distribution Access MPU Distribution Permission DISABLE MPU Bernings MPU Region 1 Settings MPU Region 1 Settings MPU Region 1 Settings MPU Region	Cottes Manay Protection Unit Region 0 Settings MPU Region Base Address Oxf MPU Region Base Address Oxf MPU Region Databa MPU Region Databa MPU Region Databa MPU Region Databa MPU Region Databas						Background Darion Drivilanad accasses only 4 MDI Disah
MPU Region Enabled MPU Region Base Addess 0x0 MPU Region Base 0x0 MPU Superground 0x87 MPU Access Permission OISARLE MPU Superground DISARLE MPU Region 1 Settings Metal Region 1 Settings	MPU Region Enabled MPU Region Base Address 0x0 MPU Region Base Address 0x0 MPU Selevices 0x0 MPU Selevices 0x0 MPU Tack field livel 1xx10 MPU Tack field livel 1xx10 MPU BaseAddress DISABLE MPU BaseAddres Permission DISABLE MPU Cacheade Permission DISABLE MPU Cacheade Permission DISABLE MPU Cacheade Permission DISABLE MPU Releation Unit Region 2 Settings MINE						secondentia region nimegea accesses only + MPO DISBD
MPU Region Base Addess 0x0 MPU Region State KGB MPU Region Deable 0x87 MPU TES Kild Kevel text0 MPU Access Permission ALL ACCESS NOT PERMITTED MPU Shaready Permission DISABLE MPU Shaready Permission DISABLE MPU Bifwaready Permission DISABLE MPU Bifwaready Permission DISABLE V Cates Merry Protection Unit Region 1 Settings MPU Region 1 Settings	MPU Region Bate Addess Oxt MPU Region State 6/09 MPU Region Drashe 0xt7 MPU TEX fold Howel level 0 MPU TEX fold Howel level 0 MPU TEX fold Howel level 0 MPU TEX fold Howel 1 level 0 MPU TEX fold Howel 2 Level 0 MPU TeX fold Howel 2 LEVEL 0 MPU TeX fold Howel 2 DISARLE MPU Cacheadre Permission DISARLE MPU Cacheadre Permission DISARLE MPU Cacheadre Permission DISARLE MPU Reprint Disabled Cates Memory Protection Unit Region 2 Settings MPU Reprint Medican Unit Region 2 Settings Disabled						Enabled
MPU Repain Size ILGB MPU SuR-Region Datatio 0x87 MPU SuR-Region Datatio 0x87 MPU Surface Networks 0x87 MPU Access Permission ALLACCESS NOT PERMITTED MPU Surface Networks DISARLE V Duffundes Permission DISARLE V Cette Menny Protection Unit Region 1 Settings MPU Region	MPU Report Size LGB MPU Sub-Region Davia 0x17 MPU Sub-Region Davia 0x17 MPU TEX Hild Invel Invel 0 MPU Access Permission ALL ACCESS NOT PERMITTED MPU Instruction Access DISABLE MPU Damaship Permission DISABLE MPU Cachabel Permission DISABLE MPU Cachabel Permission DISABLE MPU Cachabel Permission DISABLE MPU Cachabel Permission DISABLE VC Cates Memory Protection Unit Region 2 Settings Disabled						
MPU SuB-Rejon 0x87 MPU Test field field level 0 MPU Access Permission ALL ACCESS NOT PERMITTED MPU Instruction Access DISABLE MPU Instruction Access MPU Shareability Permission EIABLE MPU Shareability Permission DISABLE MPU Binner/MP Permission DISABLE MPU Binner/MP Permission DISABLE V Catek Merry Protection Unit Region 1 Settings MPU Region Disabled	MFU SuB-Rejon Ox87 MFU Text 64 Howel Invert0 MFU Text 64 Howel Invert0 MFU Access Permission ALL ACCESS NOT PERMITTED MFU Instruction Access DISARIE MFU Shareability Permission DISARIE MFU Cacheabile Permission DISARIE MFU Cacheabile Permission DISARIE Octotes Memory Protection Unit Region 1 Sterings MEU MFU Pareicol Unit Region 2 Sterings						
MFU Level 0 MFU Access Not PERMITTED MFU Access Not PERMITTED MFU Instruction Access MFU Status Permission Disabled MEU	MPU IEX faid fevel Ievel 0 MPU Access Permission ALLACCESS NOT PERMITTED MPU Instruction Access DISABLE MPU Shareability Permission DISABLE MPU Gacheabile Permission DISABLE MPU Gacheabile Permission DISABLE Cottes Memory Photection Unit Region 1 Setting MPU Region DISABLE Cottes Memory Photection Unit Region 2 Setting						
MPU Access Permission ALL ACCESS NOT PERMITTED MPU Instruction Access DISABLE MPU Stateability Permission EIMABLE MPU Gachado Permission DISABLE MPU Buffinde Permission DISABLE V Catte Memory Protection Unit Region 1 Settings MPU Region Disabled	MPU Access Permission ALL ACCESS NOT PERMITTED MPU Instruction Access DISARLE MPU Shareability Permission DISARLE MPU Cacheabel Permission DISARLE MPU Cacheabel Permission DISARLE Cottes Memory Protection Unit Region 1 Stating MPU Region Disabled Cottes Memory Protection Unit Region 2 Setings						
MPU Instruction Access DISABLE MPU Shaveabily Permission ENABLE MPU Cacheable Permission DISABLE MPU Biffunde Permission DISABLE - Conte Many Protection Unit Region 1 Settings MPU Region Disabled	MPU Instruction Access DISABLE MPU Stavability Permission EHABLE MPU Cacheable Permission DISABLE MPU Cacheable Permission DISABLE Control Memory Protection Unit Region 1 Setting Cottets Memory Protection Unit Region 1 Setting MPU Region Disabled Cottets Memory Protection Unit Region 2 Setting						
MPU Shareability Permission ENABLE MPU Cocheadre Permission DISABLE MPU Bufferable Permission DISABLE Cotter Kenny Protection Unit Region 1 Settings MPU Region Disabled	MPU Shareabily Permission EMARE MPU Cacheabe Permission DISARE MPU Cacheabe Permission DISARE Cottex Memory Photecon Unit Region 1 Statings MPU Region Disabled Cottex Memory Photecion Unit Region 2 Setings						
MPU Cacheade Permission DISARLE MPU Buffunde Permission DISARLE <-> Cotets Manny Protection Unit Region 1 Settings MPU Region Disabled	MPU Cachenable Permission DISABLE MPU Bulkerable Permission DISABLE Cottex Memory Protection Unit Region 1 Settings MPU Region 0 Disabled Cottex Memory Protection Unit Region 2 Settings						
MPU Buffenalie Permission DISABLE ~ Catex Memory Pratecian In Region 1 Settings MPU Region Disabled	MPU Deflexible Permission DISABLE Cottex Memory Protection Unit Region 1 Statings MPU Region Disabled Cottex Memory Protection Unit Region 2 Settings						
✓ Cottex Memory Protection Unit Region 1 Settings MPU Region Disabled	✓ Cortex Memory Protection Unit Region 1 Settings MPU Region Disabled ✓ Cortex Memory Protection Unit Region 2 Settings						
	 Cortex Memory Protection Unit Region 2 Settings 						
 Cortex Memory Protection Unit Region 2 Settings 							Disabled

Figure 429. 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		
	cwsteo1ccoReserved			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		
	0824020000 AXI SRAM2 (128 KB)					
	C#24000000 AXI SRAM1 (128 KB)					
	cx20010000 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 430. 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.

Figure 431. Warning message

Warning: Code Generation	×
WARNINGS:	
- Region Flash of context Boot	and region Flash of context Appli are overlapped.
Do you still want to generate of	code ?
X	es <u>N</u> o

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 432. Configure the XSPI

ome	> STM32H7R3A	.8lx 🗡 Men	noryMapfor	STM32H7R.io	c - Pinout & Configuration	n /		GENERATE CODE	
Pinout & Configuration Clock Cor				Clock Co	onfiguration	figuration Project Manager			
				 Software F 	Packs 🗸 🗸 F				
				~ 🔕		XSP	11 Mode and Configuration		
ategori	es A->Z						Mode		
	÷	Boot	Application	ExtMemLo	Runtime contexts:				
XSPI1					Boot		Application	ExternalMemoryLoader	
	E-TOUCHGFX								
	E-TOF1				Mode Single SPI				
	E-SUBG2								
	E-SMBUS				Port Port1 Single				\sim
	E-SFXS2LP1				HyperBus(TM) 1.8V Invert	ed Clock Disable			
	E-NFC7								
	E-NFC6				Chip Select Override Disa	ble			~
	E-NFC4 E-MEMS1						Configuration		
							oomigaration		
					Reset Configuration				
	E-EEPRMA1								
					Parameter Settings	MMT 🛛 📀 User Co	nstants 🛛 🥺 DMA Settings	GPIO Settings	
	E-BLE2				Configure the below paramete	rs :			
					Q Search (Ctrl+F)	0			
X-CUB	E-ALS					0			
X-CUB	E-ALGOBUILD				Fifo Threshold		1		
X-CUB	E-AI				Memory Mode		Disable		
WWDG	9				Memory Type		Micron		
	BUF				Memory Size		16 Bytes		
	DTG_HS				Chip Select High Ti	and Quela	to bytes		

- 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.



ne 🔪 STM32H7R3A8Ix	> Men	noryMapforS	TM32H7R.ioc	- Pinout & Configuration >		GENERATE CODE
Pinout & Configura	ition		Clock Co	nfiguration	Project Manager	Tools
		×	Software Page 1	acks 🗸 Pinout		
			~ 🔕		EXTMEM_MANAGER Mode and Configur	a
egories A->Z					Mode	
omputing			2	Runtime contexts:		
			~	Boot	Application	ExternalMemoryLoader
ddleware and Software Packs						×.
\$	Boot	Application	ExtMemL	Activate External Memory Manag	er	
AIROC-Wi-Fi-Bluetooth-ST						
EXTMEM_LOADER			V			
EXTMEM_MANAGER						
FATFS						
FP-SNS-MOTENVWB1						
FP-SNS-SMARTAG2					Configuration	
FREERTOS				Reset Configurati		
I-CUBE-CANOPEN				Reset Configurati		
I-CUBE-Cesium				🗢 Boot usecase 🛛 😔 Memory 1 🛛 0	Memory 2 Ser Constants	
I-CUBE-ITTIADB				Configure the below parameters :		
I-CUBE-embOS I-CUBE-wolfSSL				Q Search (Ctrl+F) (0) (0)		0
				✓ Boot	_	
I-Cube-SoM-uGOAL						
I-Cube-SoM-uGOAL LWIP				select boot code generation		
I-Cube-SoM-uGOAL LWIP USBPD				Selection of the boot system	✓ Execute In Place	
I-Cube-SoM-uGOAL LWIP					Execute In Place	

Figure 433. 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.

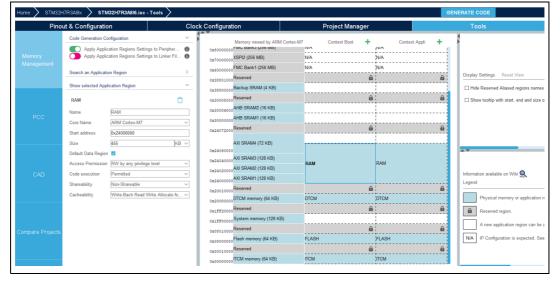


Figure 434. Tooltip

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	
			~ Ø	CORTEX_	M7_APPLI Mode and Configura	tion
tegories A->Z					Mode	
System Core			~	Runtime contexts: Boot	Application	ExternalMemoryLoader
÷	Boot	Application	ExtMemI	Boot	Application	ExternalMemoryLoader
CORTEX M7 AP	Door	S S	Extinent			
CORTEX M7 BO	2				Configuration	
FLASH						
GPDMA1	~			Reset Configuration		
GPIO		_		Parameter Settings Subser Constants		
HPDMA1	1	2		Configure the below parameters :		
				Q Search (Ctrl+F) ③ ④		
NVIC APPLI				Speculation default mode Settings		
NVIC BOOT				Speculation default mode Settings Speculation default mode	Disabled	
RCC	1			Cortex Interface Settings	Linsabled	
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 		
1000				MPU Region	Enabled	
Timers			>	MPU Region Base Address	0x0800000	
				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	
211 11 12 AT				MPU Cacheable Permission	ENABLE	
Computing			>	MPU Bufferable Permission	DISABLE	
Middle	Dealer		>	Cortex Memory Protection Unit Region 1 Settings		
Middleware and Software	e Macks			MPU Region	Disabled	
Trace and Debug			5	 Cortex Memory Protection Unit Region 2 Settings MPU Region 	Disabled	
made and Debug			<u></u>	MPU Region		

Figure 435. 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.

Management Code Generation Configuration Management Management Management Management Codes and Application Region Settings to Enripherals: OIL 0 Apply Application Region Management Management Management Apply Application Region Settings to Enripherals: OIL 0 Apply Application Region Management Manage	> STM32H7	17R318Kx 🔰 Untitled - Tools 🔪					GENERATE CODE
Management Apply Application Regions Settings to Persphereds: ON Apply Application Regions Settings to Linker Files: ON Show selected Application Region Control Linker Files: ON CADD Control Linker Files: ON Show selected Application Region Control Linker Files: ON Categories Control Linker Files: ON Categories Control Linker Files: ON Show selected Application Region Control Linker Files: ON Categories Control Linker Files: ON Categories Control Linker Files: ON Show selected Application Region Control Linker Files: ON Categories Control Linker File: ON Categories Control Linker Categories Control Linker File	Pinc	out & Configuration	Cloc	ck Configuration	Project Ma	nager	Tools
Mannory Management Appl Application Region Settings to Linker Files: CN Image method (not compared to the setting to Linker Files: CN) Image method (not compared to the setting to Linker Files: CN) Image method (not compared to the setting to Linker Files: CN) Image method (not compared to the setting to Linker Files: CN) Image method (not compared to the setting to Linker Files: CN) Image method (not compared to the setting to Linker Files: CN) Image method (not compared to the setting to Linker Files: CN) Image method (not compared to the setting to Linker Files: CN) Image method (not compared to the setting to Linker Files: CN) Image method (not compared to the setting to Linker Files: CN) Image method (not compared to the setting to the set	<u>[</u>	Code Generation Configuration	<u> </u>	Memory viewed by ARM	I Cortex-M7 C	Context Boot 🕂	Context Appli 🕂
Memory Management Corport spatialized scaling is data if all of all				0xE0000000 Reserved		â	â
Beach an Application Region > Generation (25 MB) NA NA Brow selected Application Region > Generation (25 MB) NA NA PCC Show selected Application Region > Generation (25 MB) NA NA CAD Show selected Application Region > Generation (25 MB) NA NA CAD Show selected Application Region > Generation (25 MB) NA NA CAD Show selected Application Region NA NA NA CAD Show selected Application Region Show selected Application Region NA NA CAD Show selected Application Region Show selected Application Region NA NA CAD Show selected Application Region Show selected Application Region NA NA CAD Show selected Application Region Show selected Application Region Show selected Application Region NA CAD Show selected Application Region CAD Show selected Application Region Show selected Application Region Show selected Application Region Show selected Application Region	ory	Appry Approaches regions detailings to blickel Files. Or		0xD00000000 FMC SDRAM Bank2 (256 MB)	N/A)	N/A
FCC Des 000000 (MC Bankl (25 MB)) NA NA FCC FMC Bankl (25 MB) NA NA FMC Bankl (25 MB) NA NA NA FMC Bankl (25 MB) NA NA NA FMC Bankl (25 MB) NA NA NA FMC Bankly SRAMI (16 KB) FMC Bankly (12 KB) <t< td=""><td></td><td>Search an Application Region</td><td>></td><td>0xC00000000 FMC SDRAM Bank1 (256 MB)</td><td>N/A</td><td>)</td><td>N/A</td></t<>		Search an Application Region	>	0xC00000000 FMC SDRAM Bank1 (256 MB)	N/A)	N/A
CAD State MA MA CAD MA MA MA CAD MAS SAMA (126 MB) MA MA CAD MAS SAMA (16 KB) MA MA CAD MAS SAMA (12 KB) MA MA SCALCORD MAS SAMA (12 KB) MA MA <td< td=""><td>s</td><td>Show selected Application Region</td><td>></td><td></td><td></td><td>â</td><td>a</td></td<>	s	Show selected Application Region	>			â	a
PCC NA NA 0x1000000 RAC Bank1 (256 MB) NA NA 0x2000000 RAC Bank1 (256 MB) NA NA 0x1000000 Researed 0 0 0 0x1000000 Researed 0 0 0 0 0x1000000 Researed 0 <td< td=""><td></td><td></td><td>_</td><td></td><td>N/A</td><td></td><td>N/A</td></td<>			_		N/A		N/A
PCC Dc Bank1 (256 MB) NA NA Bestrond Bestrond Calibration (256 MB) NA CAD Bestrond Bestrond Calibration (256 MB) Calibration (256 MB) CAD Bestrond Bestrond Calibration (256 MB) Calibration (256 M					N/A)	N/A
CAD 6x3500000 Reserved 6x360000 Reserved 6x300000 CAD 6x300000 Reserved 6x300000 Reserved 6x300000 MB SRAM1 (16 KB) 0 0 0 0 0 MB SRAM1 (16 KB) 0 <td< td=""><td></td><td></td><td></td><td>0x70000000 XSPI2 (256 MB)</td><td>N/A</td><td>)</td><td>N/A</td></td<>				0x70000000 XSPI2 (256 MB)	N/A)	N/A
CAD 6x1300000 (0x000000 ABBowned) 6x80000 (0x000000 ABB SAM2 (16 K6)) 0 0 CAD 0x1500000 (0x100000 ABB SAM2 (12 K6)) 0 0 0 AXI SRAM4 (12 K8) (0x100000 ABB SAM2 (12 K6)) 0 0 0 0 0x1500000 ABB SAM2 (12 K6)) 0 0 0 0 0 0x1500000 ABB SAM2 (12 K6)) 0 0 0 0 0 0 0x1500000 ABB SAM2 (12 K6) 0	PCC				N/A)	N/A
CAD 0x1000000 AHB SRAM2 (16 KB)						â	<u> </u>
CAD 6±2000000 FMB SRAM2 (16 K8) Image: Calibration of the Status (16 K8) Image: Calibration of the							
CAD Gestionnee HEI SRAM1 (16 KB) Image: Calibration of the served				0x30008000 Reserved		â	<u> </u>
CAD 6x24072000 Reserved Image: Calibratic							
AXI SRAM4 (72 KB) AXI SRAM4 (72 KB) AXI SRAM4 (72 KB) AXI SRAM4 (72 KB) 6x24040000 MI SRAM2 (128 KB) AXI SRAM2 (128 KB) RAM 0x2000000 MI SRAM2 (128 KB) RAM RAM 0x2000000 MI M memory (64 KB) DTCM DTCM 0x112 M memory (64 KB) RAM RAM RAM						-	
0x2400000 AVI SRAM3 (121 KG) RAM RAM 0x2400000 AVI SRAM2 (121 KG) RAM RAM 0x2400000 AVI SRAM1 (121 KG) RAM RAM 0x2000000 Reserved RAM RAM 0x2000000 REServed RAM RAM 0x2000000 ROM memory (45 KG) RAM RAM	CAD			0x24072000 Reserved		â	<u> </u>
ODD T Text Subte M3 SRAM3 (128 (0B) (01 SRAM2 (128 (0B)) (01 SRAM2 (128 (128 (128 (128 (128 (128 (128 (12				AXI SRAM4 (72 KB)			
Op/En Op/En <th< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td></th<>							
Citz 24000000 ANI SRAWT (128 KB) Image: City 2400000 DDR Test Suite City 2000000 DTCM DTCM City 2000000 DTCM DTCM DTCM City 2000000 City 2000000 City 2000000 City 2000000 Dit CM					RAM		RAM
DDR Test Suite descended d							
ODLR Fest suite Grazionosono ^{DTCM} (KB) DTCM (Sector) Grazionosono ^{Reserved}							
ox1FF20000Reserved	Fest Suite			0x20010000 Reserved	DTCM		
0x1FF20000							
Sustam mamory (129 VB)				0x1FF20000 Reserved 0x1FF00000 System memory (128 KB)			
621F00000 System mentor (Leo KD)						۵	<u>م</u>

Figure 436. Linker files update



ne 🔪 STI	M32H7S3IBKx 🔪 test.ioc - Tools 🔪		,			GENERATE CODE
	Pinout & Configuration	Clock Configuration		Project Manager		Tools
	Code Generation Configuration	Memory viewed by ARM (Cotex-M7	Context Boot	Context Appli	1
	Apply Application Regions Settings to Peripher.	O CEACODOCCO PARAGENER				
	Apply Application Regions Settings to Linker Fil	0x90000000 XSPI1 (256 MB)	N/A	¢VA		
	Search an Application Region	Oxteosococ FMC Bank3 (256 MB)	N/A	λ∧v		
		0x7000000 XSPI2 (256 MB)	N/A	7VA		Display Settings Reset View
	Show selected Application Region	> FMC Bank1 (256 MB)	N/A	₩A		THE REAL PROPERTY OF
		0x35501000 Reserved		â:	Ê.	Hide Reserved Aliased regions names
		0x38800000 Backup SRAM (4 KB)				Show tooltip with start, end and size of
		0x30000000 Reserved	Add an application regi	ion × Br:	£	
		0#30004000 AHB SRAM2 (15 KB)	Name	AppReg		
		0x30010000 AHB SRAM1 (15 KB)	Address 0x38800000	0		
		0x24072000 Reserved	Size 4	08 . KB 0 MB	â	
		A31 SRAM4 (72 KB) 0x24040000	Context Dont	Cancel Act		2.*
		0x24040000 AX SRAM3 (128 KB)	RAM	RAM		and the second second second
		0x24020000 AXI SRAM2 (128 KB)				Information available on Wiki 🧟
		0x24060000 AXI SRAM1 (128 KB)				Legend
		0x20010000Reserved		ê;	ê :	Physical memory or application re
		0x20000000 DTCM memory (64 KB)	DTCM	,DTCM		B Reserved region.
		Ox17720000 Reserved		ê ;	Ê.	C Reserved region.
		Ox1FF00000 System memory (128 KB)				A new application region can be ad
		Ox00010000 Reserved			â :	N/A IP Configuration is expected. See
		exosoecce Flash memory (54 KB)	FLASH	(FLASH		
		oxooolooce Reserved		â ;	â :	
		Ox00000000 ITCM memory (64 KB)	ITCM	TCM		

Figure 437. 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
				V Software Packs	 Pinout 		
			0	Ð	TMEM_MANAGER Mode and Cor	n-guration	Pinout view 🔛 System view
legories A>Z					Mode		
fiddleware and Software Packs			~	Runtime contexts:			
AIROC W 7-Busices STAD	Boot	Application	ExtMemLoader	Beet	Application	ExternalMemoryLoader 19	
EXTMEM LOADER			12	Activate External Memory Ma	inager		
EXTMEM MANAGER			2				
FAILS			1				
FF-SNS-MOTENVW81							
FP-SNS-SMARTAG2							
FREERTOS							
I CUBE CANOPEN							
							00000000000000000
	0						
	0	0					
X-CUBE-AI	<u>u</u>	U					
X-CUBE-ALGOBULD							
X-CUBE-ALS							
X-CUBE-BLE!					Configuration		
X-CUBE BLE2					Consiguration		000000000000000
X-CUBE-BLEMGR				Reset Configuration			
X-CUBE-EEPRMA1							
X-CUBE-GNSS1				Boot usecase Memory 1	Memory 2 User Cons	tants	
X-CUBE ISPU				Configure the below parameters :			
X-CURE-MEMS1				QSearch (Chi+F) 0 0		0	
X-CUBE-NEC4				Announcement of the second sec		0	000000000000000000000000000000000000000
X CUBE NFC6				~ Bost	-		
X-CUBE-NFC7				select boot code generation	on 🗌		
X-CUBE-SFXSR.Pt							UFBGA169 (Top view)
X-CUBE-SNBUS							
X-CUBE-SUBG2 X-CUBE-TOF (
X-CUBE-TOUCHGFX							

Figure 438. 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 Externalflemory.cader छ	Pinsut view
defector and Software Packs Boot All 1 AIRED NAL DOORD Boot All EXTRUDAL LONGRA C All EXTRUDAL LONGRA C All PARISON AND TANKA C All PARISON AND TANKA C All PARISON AND TANKA C C PARISON AND TAND TANKA C C PARISON AND TANKA 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 (Coll Coll All ERTMON (Jongen) Coll All ERTMON (Jongen) Coll All ERTMON (Jongen) Coll All Fill (Coll (Application E	ExtMemLoader	Runtime contexts: Boot	Mode Application	ExternalMemoryLoader	Pinaut view ¹⁰ System view
Idensity and Software Packs Boot A ALRCS (IN CACAR) C A ALRCS (IN CACAR) C A EXMODUL_MARKACK C A EXMODUL_MARKACK C A FREEXTOR C A ISABLE SAME AND		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
EXTINUE (LADEER EXTINUE (LADEER AT 73 0 TA 205 0 TA				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.MOTEVINET PLANS.MOTEVINET PLANS.MOTEVINET REFITOR REFIT						
PLANS.SMATAG2 PLANTAG2 PLANTAG						
Imperation Imperation CodeE CANOPER Imperation CodeE Comm Imperation CodeE Comment Imperation CodeE code Imperation CodeE code Imperation CodeE code Imperation CodeE code Imperation CodeE codE codE codE codE codE codE codE cod						000000000000000000000000000000000000000
Calle C-Assing A Caller Control Contro						
Calle C. Andre N Calle C. Calles m Calles C. Market Calles C. 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			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						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 CUEP A/ SCARE / SCAR						
						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 439. Execute In Place

Figure 440. MMT Execute In Place

Pinout & Conf	iguration		Clock Co	onfiguration		Project Manager		Tools	
Code Generation C		~		Memory viewed by ARM Con	tex-M7	Context Boot	+	Context Appli	1
	cation Regions Settings to Pe cation Regions Settings to Lin		0xE000000	Reserved			a :	6	8
Арру Арри	cation Regions Settings to Lin	iker Piles: ON 😗	0x2000000	FMC SDRAM Bank2 (256 MB)	N/A		'N/A		1
Search an Applicat	ion Region	>	0xC000000	FMC SDRAM Bank1 (256 MB)	N/A		N/A		
Show selected App	lication Region	~	0x2000000	Reserved			â	ê	
RAM		Ċ	0#9000000	External Flash (XSPI1) (256 MB)			FLASH		Dis
Name	RAM		0x8000000	FMC Bank3 (256 MB)	N/A		'N/A		
Core Name	ARM Cortex-M7	~	0x7000000	XSPI2 (256 MB)	N/A		'N/A		
Start address	0×24000000			FMC Bank1 (256 MB)	N/A		'n/A		
Size	120	KB V	0x3880100	Reserved			a	6	
Default Data Region				Backup SRAM (4 KB)					5
Access Permission	RW by any privilege level	~	0x3000800	Reserved			<u> </u>	6	8
Code execution	Permitted	~		AHB SRAM2 (16 KB)					
Shareability	Non-Shareable	~		AHB SRAM1 (16 KB)					
Cacheability	Write-Back Read Write Allo	ocate for RA V	0x2407200	Reserved			A	£	
				AXI SRAM4 (72 KB)					
				AXI SRAM3 (128 KB)					
			0x2402000	AXI SRAM2 (128 KB)					4
				AXI SRAM1 (128 KB)	RA		RAM		
			0x2400000 0x2001000		KA	-		۵	Info
			0x2001000	DTCM memory (64 KB)	DT	3M	DTCM		Leg
			0x2000000 0x1FF2000				â	6	
			0x1FF2000	System memory (128 KB)					
			0x1FF0000 0x0801000				â	A	
			0x0501000	Flash memory (64 KB)	FU	ISH		rd FLASH(Boot)	
			0x0001000				0 '		N



e > STM32H7R3A8k >	Untitled - Pino	ut & Configural	tion >				GENERATE CODE
Pinout & Config	guration		c	lock Configuration	Project Ma	anager	Tools
					 Pinout 		
		Ŷ	G		MANAGER Mode and Configuration		Pinout view I ¹¹ System view
gories A->Z				17 million	Mode		
iddleware and Software Packs			· •	Runtime contexts:			
	P	Andresson	C. all and a set of	Boot	Application Exten	nalMemoryLoader	
AIROC-W-F-Bluetoch-STN32	Boot	Application	ExtMemLoader	2		53	
EXTMEM LOADER			53	Activate External Memory Manager			
EXTMEM MANAGER		1		Contraction of the second seco			
FATES		0					
		<u> </u>					
	0						
							000000000000000000000000000000000000000
							000000000000000000000000000000000000000
	0						0000000000000
					Configuration		
							0000000000000
				Reset Configuration			
				Boot usecase Memory 1 M	feman 2 Allers Constants		
					Constants		0000000000000000
				Configure the below parameters :			
				Q. Search (Chit+F) ③ ④		0	
				✓ Boot			00000000000000
				select boot code generation			
				Selection of the boot system	Load and Run		
				LRUN source			
				select the source memory	Memory 1		UFBGA169 (Top view)
				source address offset	0x00000000 hex		
				source code size	0x00000000 hex		
				V LRUN destination			
				selection of the memory	Internal Memory		
				destination address	0x24050000 hex		
ce and Debug			>				
wer and Thermal			>				
lities							
her							

Figure 441. Load and Run

Figure 442. MMT Load and Run

	Pinout & Conf	iguration		Clock Configuration	P	roject Manager	Tools	
	Code Generation Co		¥	Memory viewe	I by ARM Cortex-M7	Context Boot +	Context Appli 🕂	4
mory nagement	Apply Application Regions Settings to Peripherals: ON Apply Application Regions Settings to Linker Files: ON		0xE0000000 Reserved		6	:		
	Apply Appli	cation Regions Settings to Lin	ker Files: ON 🌘	0xD0000000 FMC SDRAM Bank2 (25	MB) N/A		N/A	
gomon	Search an Application Region >		EMC SDRAM Bank1 (25	MB) N/A		N/A		
			~	0x20000000 Reserved		<u></u>	A	
	Show selected App	lication Region		0x90000000 External Flash (XSPI1) (2	56 MB) Reserved L	OAD_REGION(Appli)	Reserved LOAD_REGION	
PCC	RAM		Û	0x10000000 FMC Bank3 (256 MB)	N/A		AVA	Display
	Name	RAM		0x70000000 XSPI2 (256 MB)	N/A			Пна
	Core Name	ARM Cortex-M7		0x700000000 FMC Bank1 (256 MB)	N/A		5VA	□ Sho
	Start address	0x24000000		0x38801000 Reserved		â		L) She
	Size	120	KB ~	0x38800000 Backup SRAM (4 KB)				
	Default Data Region			0x30000000 Reserved			A	
		RW by any privilege level	~	0x30004000 AHB SRAM2 (16 KB)				
CAD	Code execution	Permitted		0x30000000 AHB SRAM1 (16 KB)			1	
	Shareability	Non-Shareable		0x24072000 Reserved		A	A	
	Cacheability	Write-Back Read Write Allo	-	0x24060000 AXI SRAM4 (72 KB)				
	Gacrieadincy	THRE Dack Read Three Para	cate for fort	0424040000	Reserved F	(LASH(Appli)	FLASH	A.Y.
				AXI SRAM3 (128 KB) 0x24040000				
				0x24040000 AXI SRAM2 (128 KB)			L	
				0424020000			l	
				AXI SRAM1 (128 KB) 0x24000000	RAM		RAM	
				0x20010000 Reserved		ê		Informat Legend
				0x20000000 DTCM memory (64 KB)	DTCM		ртсм	Legend
				0x1FF20000Reserved		a	a	
				0x1FF00000 System memory (128 KE)			â
				0x08010000 Reserved		<u></u>	a	
				0x080000000 Flash memory (64 KB)	FLASH		Reserved FLASH(Boot)	
				0x00010000 Reserved		â		N/A
				0x00000000 ITCM memory (64 KB)	ITCM		атсм	

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 Point */ ENTRY (Reset Handler)				
ENIRI (Reset_Handler)				
/* Highest address of the user mode stack */				
_estack = ORIGIN(RAM) + LENGTH(RAM); /* end of Ram type memory */				
_Min_Heap_Size = 0x200; /* required amount of heap */				
Min_Stack_Size = 0x400; /* required amount of stack */				
/* Memories definition */				
REMORT 1				
FLASH (rx) : ORIGIN = 0x080000000, LENGTH = 64E				
ITCM (xrw) : ORIGIN = 0x00000000, LENGTH = 64K				
DTCM (xrw) : ORIGIN = 0x20000000, LENGTH = 64K				
RAM (xrw) : ORIGIN = 0x24000000, LENGTH = 455E				
-)				
/* Sections */				
SECTIONS				
/* The startup code into "FLASH" Rom type memory */	Memory viewed by ARM Cortex-M7	Context Bost	+ Context Apple	1
.isr vector :	catoooccus in the and	Correst Dots	T Uness Appl	T
1	THE ROOM OF LEVE	NA	NA	
. = ALIGN(4);	2880000000	74 A	74A	
<pre>KEEP(*(.isr_vector)) /* Startup code */</pre>	factorization Reserved		6	â
. = ALIGN(4);	Barders Official is with	***************************************		
-) >PLASH	antione			
	(x20000000 Reserved		():	â ;
/* The program code and other data into "FLASH" Rom type memory */ .text :	AHE SRAM2 (16 KB)		1	
.text :	1200400 Sec. 10			
= ALIGN(4);	(#3000000 AHB SRAM1 (16 KB)			
(.text) / .text sections (code) */	Reserved		6	â
(.text) /* .text* sections (code) */	sile to serve	******		
(.glue_7) / glue arm to thumb code */	AU SRAMA (72 KEI)			
<pre>*(.glue_7t) /* glue thumb to arm code */</pre>	0x24040000		1	
*(.eh_frame)	AU SRAMJ (128 KB)		204M	
<pre>KEEP (*(.init))</pre>	211 00 100 100 LONG LONG	RAM		
KEEP (*(.fini))	0x24020000 AU SRAM2 (128 KB)			
	(#24100000 AU SRAM1 (128 KB)			
. = ALIGN(4);	Reserved		6	A
_etext = .; /* define a global symbols at end of code */				
—) >FLASH	(s2000000 DTCM memory (54 KS)	DTCM	;DTCM	
/* Constant data into "FLASH" Rom type memory */	Contraction Reserved		6	â
.rodata :	Eastern manage (1710 p.D)	******		
	(stiff)())) option nemory (stand)		1	
. = ALIGN(4);	Currenzioner Reserved		():	ê :
	The American States	FLASH	FLASH	
(.rodata) / .rodata sections (constants, strings, etc.) */			and the second se	
<pre>*(.rodata) /* .rodata sections (constants, strings, etc.) */ *(.rodata*) /* .rodata* sections (constants, strings, etc.) *</pre>	dimension of the second s			
(.rodata) / .rodata sections (constants, strings, etc.) */	Central Little Reserved		6	â,

Figure 443. Linker files

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.

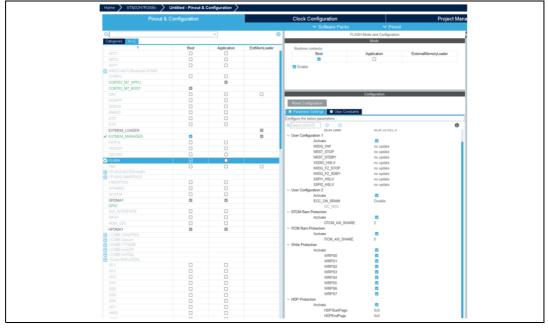


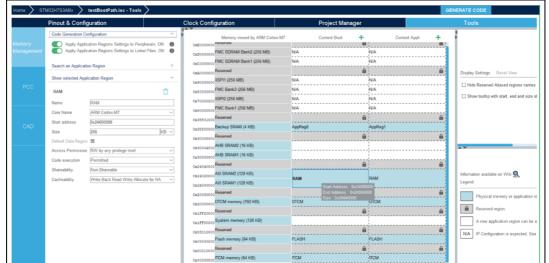
Figure 444. Flash option bytes



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 445. ECC_ON_SRAM enabled and DTCM_AXI_SHARED set to 2



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 446*).







UM1718 Rev 46

Figure	446.	MMT	usage
--------	------	-----	-------

	Configuration
Reset Configuration	
 Parameter Settings User Const. 	ants
Configure the below parameters :	
Q Search (Ctrl+F) ③ ③	0
Cortex Memory Protection Unit Control	Settings
MPU Control Mode	Background Region Privileged accesses only + M.
Cortex Memory Protection Unit Region	0 Setti
MPU Region	Enabled
MPU Region Base Address	0x10000000
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 447.	User interface

inout & Configuration	Clock C	Configuration	Project Manag	er	۱	Fools
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		20010000 Reserved				é
Apply Application Regions Settings to Enixer Files, OFF	0x2	2000C000 SRAM3 (16 KB)				
Search an Application Region	>	20008000 SRAM2 (16 KB)		RAM		
Show selected Application Region		20004000 SRAM1 (16 KB)				
	0x2	20000000 SRAM0 (16 KB)				
	0.81	100C0000 Reserved				é
	0x1	10040000 Flash memory (512 KB)		FLASH		
	0x1	10001800 Reserved				é
	0.81	10000000 ROM memory (6 KB)		REGION_ROM		
	0.00	00000000Reserved				í

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 448. Linker files update

Code Generation Configuration	~	Memory viewed by ARM Contex-M0+	Application Regions
Apply Application Regions Settings to Peripherals: ON Apply Application Regions Settings to Linker Files: ON		0x20010000Reserved	â
- the share country of the second	~ I	0x20000000 SRAM3 (16 KB)	
Search an Application Region	>	0x2000e000 SRAM2 (16 KB)	RAM
Show selected Application Region	- X -	0x20004000 SRAM1 (16 KB)	
		0x20000000 SRAM0 (16 KB)	
		Cx100C0000 Reserved	â
		0x10040000Flash memory (512 KB)	FLASH
	- 1	ox10001800 Reserved	â
	- 1	0x1000000 ROM memory (5 KB)	REGION_ROM
		exoseccose Reserved	â

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 449. Impact on RADIO (STM32WB09)

 Pinout & Configu	ration		Clock Config	uration	Project Ma	nager		Tools
Code Generation Co		×]		Memory v	iewed by ARM Cortex-M0+		Application Regions	+
	ation Regions Settings to Per ation Regions Settings to Link		0x2001000	Reserved				â
Search an Application	on Region	>	0x2000C00					
Show selected Appl	ication Region	~		SRAM2 (16 KB) SRAM1 (16 KB)		RAM		
RAM		Û	0x2000400	0.0000000				
Name	RAM			SRAM0 (16 KB)		Blue Core Config		â
Core Name	ARM Cortex-M0+	~	0x2000000					
Start address	0x200001A0		0x1000000	Reserved				6
Size	63	KB 🗸	0x1004000	Flash memory (512 KB)		FLASH		
Default Data Region	2		0x1000150	Reserved				â
Access Permission	RW by any privilege level	~	0x1000000	ROM memory (6 KB)		REGION_ROM		
Code execution	Permitted	Ý	0x0000000	Reserved				â
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
0x90000000			CODE_IN_EXT_FLASH
0x80000000	FMC Bank3 (256 MB)	N/A	N/A
0x70000000	XSPI2 (256 MB)	N/A	N/A
0x60000000	FMC Bank1 (256 MB)	N/A	N/A
0x38801000	Reserved	a	a
0x38800000	Backup SRAM (4 KB)		
0x30008000	Reserved	â	a
	AHB SRAM2 (16 KB)		
0x30004000			RAM
0x30000000	AHB SRAM1 (16 KB)		
0x24072000	Reserved	a	a
0x24060000	AXI SRAM4 (72 KB)		
0x24040000	AXI SRAM3 (128 KB)		
0x24020000	AXI SRAM2 (128 KB)		
0x24000000	AXI SRAM1 (128 KB)		
0x20010000	Reserved	a	a
0x20000000	DTCM memory (64 KB)		DTCM
0x1FF20000	Reserved	a	<u> </u>
	System memory (128 KB)		
0x08010000	Reserved	<u> </u>	
0x08000000	Flash memory (64 KB)		BOOT_PATH_EXEC
0x00010000	Reserved	6	a

Figure 450. MMT/boot path (STM32H7RS)



Memory viewed by ARM Cortex-M33	Application Regions	+
0x0C080000 Reserved		â
Flash Bank2 memory (S) (256 KB)		
0x0C040000	DOWNLOAD_NON_SECURE_CODE_REGION (NS) Reserve	d Alias R
	DOWNLOAD_SECURE_CODE_REGION	6
	FLASH (NS) Reserved Alias Region	â
	rLASH (NS) Reserved Allas Region	
Flash Bank1 memory (S) (256 KB)	FLASH_NSC (NSC)	
	FLASH (S)	
0x0C000000	ROT_REGION (NS) Reserved Alias Region	â
0x0A044000 Reserved		
0x0A034000 SRAM3 Code (NS) (64 KB)		
0x0A020000 SRAM2 Code (NS) (80 KB)		
SRAM1 Code (NS) (128 KB)	RAM (NS) Reserved Alias Region	6
0x0A00000	RAM (S) Reserved Alias Region	6
0x08080000 Reserved		â
Flash Bank2 memory (NS) (256 KB)		
0x08040000	DOWNLOAD NON SECURE CODE REGION	â
	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 452*)
- Look at the memory definition: check the "App_User" declaration in the Appli project in case of an OEM-iRoT boot path (see *Figure 453* and *Figure 454*).



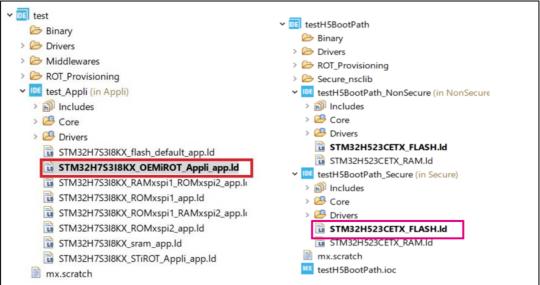
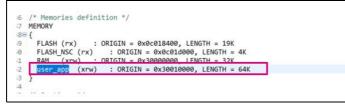


Figure 452. Linker files location (STM32H7RS on the left, STM32H5 on the right)



RAM RAM_NONC CONF FLA	ACHEABLEBUFFER (xrw) : ORIGIN =	RAM, LENGTH =RAM_SIZERAM_NONCACHEABLEBUFFER_SIZERAM_CONF_FLAG_SIZI RAM_NONCACHEABLEBUFFER, LENGTH =RAM_NONCACHEABLEBUFFER_SIZE RAM_CONF_FLAG, LENGTH =RAM_CONF_FLAG_SIZE
RAMECC_H		RAM_CONF_FEND, LENGTH =RAM_CONF_FEND_SIZE RAM_RAMECC_HANDLER, LENGTH =RAM_RAMECC_HANDLER_SIZE
ITCM	(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

Figure 454. 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 455* and *Figure 456* 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



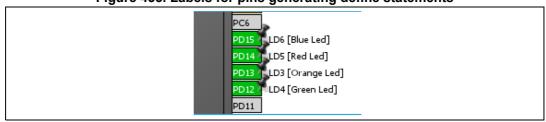


Figure 455. Labels for pins generating define statements



Co	onfiguration
Reset Configuration	
📀 NVIC Settings 🛛 📀 DMA Settings	GPIO Settings
Parameter Settings	Ser 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 457*, 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



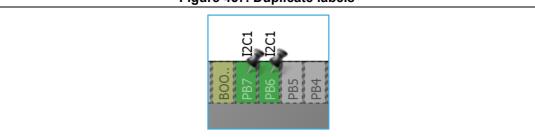


Figure 457. Duplicate labels

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.



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></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.



```
Figure 458. HAL-based peripheral initialization: usart.c code snippet
```

```
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. 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;
 GPI0_InitStruct.Mode = LL_GPI0_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);
```



*/ main.c HAL-based		*/ main.c LL-based
/* Includes	-	/* Includes
<pre>#include "main.h"</pre>		<pre>#include "main.h"</pre>
<pre>#include "stm3214xx_hal.h"</pre>	E 🗘	
#include "usart.h"		<pre>#include "usart.h"</pre>
#include "gpio.h"		<pre>#include "gpio.h"</pre>
<pre>void SystemClock_Config(void);</pre>		<pre>void SystemClock_Config(void);</pre>
<pre>void Error_Handler(void);</pre>		<pre>void Error_Handler(void);</pre>
int main(void)		<pre>int main(void)</pre>
{		{
<pre>/* Reset of all peripherals,</pre>		/* Reset of all peripherals,
Initializes the Flash interface and the Systick. **/		Initializes the Flash interface and the Systick. **/
HAL_Init();	4	LL_Init();
/* Configure the system clock */		/* Configure the system clock */
SystemClock_Config();		SystemClock_Config();
<pre>/* Initialize all configured peripherals */</pre>		<pre>/* Initialize all configured peripherals */</pre>
MX GPIO Init();		MX GPIO Init();
MX_USART1_UART_Init();		MX_USART1_UART_Init();



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 461).

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:

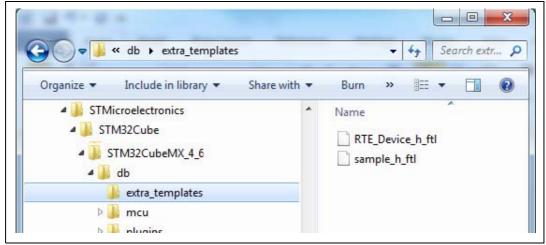
FreeMarker command in template:

```
${peripheralParams.get("RCC").get("LSI_VALUE")}
```

Resulting generated code:

```
LSI_VALUE : 32000 [peripheralParams.get("RCC").get("LSI_VALUE")]
```

Figure 461. Default content of the extra_templates folder



UM1718 Rev 46



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 462*).

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 463*).

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 464*).

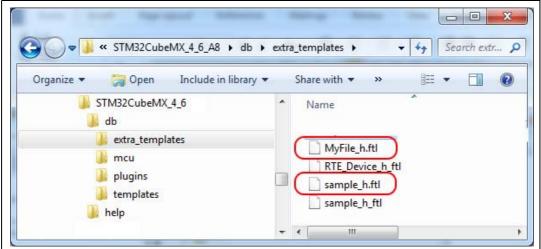


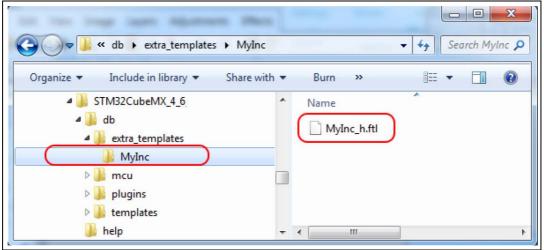
Figure 462. extra_templates folder with user templates



rigule 403. Project root loider with t	.01	responding custom generated mes
Organize Include in library Share with		eGen > • • • Search Cus > Burn New folder
CustomCodeGen Custo	* III	Name I Drivers 1 Inc 1 MyInc 1 MyInc 1 Src 1 Nmxproject 1 CustomCodeGen.ioc 1 MyFile.h 1 sample.h 1

Figure 463. Project root folder with corresponding custom generated files

Figure 464. User custom folder for templates





	J
Goo ♥ → CustomCodeGen ▶ MyInc	
Organize ▼ Include in library ▼ Share with ▼ Burn ≫	:= - 1 🔞
CustomCodeGen Drivers Inc MyInc Projects)
	•

Figure 465. Custom folder with corresponding custom generated files

6.4 Additional settings for C project generation

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 pre-processor define statements
 - Preprocessor define statements can be specified using the following syntax after the [Others] line:

Syntax

```
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=I2S; I2C
```

.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:

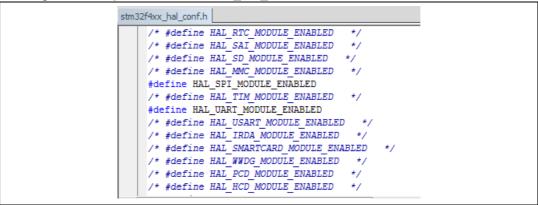
- the project MyF429IDiscoProject.ewp file in EWARM folder (see Figure 466)
- the stm32f4xx_hal_conf.h file in the project Inc folder (see *Figure 467*)
- the project view within EWARM user interface as shown in Figure 468 and Figure 469.



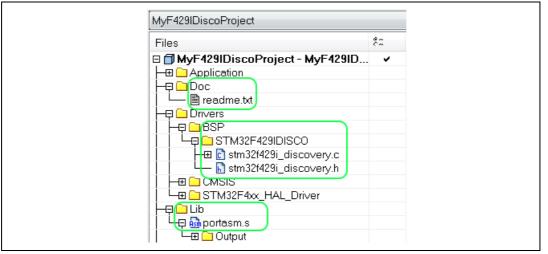
Figure 466. Update of the project .ewp file (EWARM IDE) for preprocessor define statements



Figure 467. Update of stm32f4xx_hal_conf.h file to enable selected modules









tions for node "test"		
Category:		Factory Settings
General Options Static Analysis	*	Discard Unused Publics
Runtime Checking		Language 1 Language 2 Code Optimizations Output List Preproce
C/C++ Compiler		
Output Converter Custom Build Build Actions Linker Debugger Simulator Angel CADI CADI CMSIS DAP		Ignore standard include directories Additional include directories: (one per line) \$PROJ_DIR\$//Inc \$PROJ_DIR\$//Inc \$PROJ_DIR\$//Drivers/STM32F4xx_HAL_Driver/Inc \$PROJ_DIR\$//Drivers/STM32F4xx_HAL_Driver/Inc/Legacy \$PROJ_DIR\$//Drivers/CMSIS/Device/ST/STM32F4xx/Include \$PROJ_DIR\$//Drivers/CMSIS/Include \$Prevention \$Proj_Dires//Drivers/CMSIS/Include
GDB Server		
IAR ROM-monitor		Defined symbols: (one per line)
I-jet/JTAGjet J-Link/J-Trace TI Stellaris Macraigor		USE_HAL_DRIVER STM32F429xx USE_STM32F429I_DISCO
PE micro		
RDI		
ST-LINK		
Third-Party Driver	*	OK Cancel

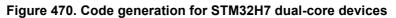
Figure 469. Preprocessor define statements in EWARM IDE



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	onfiguratio	on			📜 CN	4
۵ .	~		(Ö			📜 CM	17
Categories A->Z				<u> </u>		📕 Co	mmon
÷ HRTIM	Cortex-M7	Cortex-M4	Initializer			📕 Dri	vers
/ 12C1	\checkmark		Cortex-M7			📜 EW	ARM
12C212C3		\checkmark	Cortex-M4 Cortex-M7 ~			🗍 .m	project
12C4						_	M32H747 dualcore project1.ioc
12S1 12S2						511	
	iew <u>P</u> roject		Tools Window	v <u>H</u> elp	Ŧ	< Q	
Workspace				-	ņх		
							Solort the context to
STM32H747 d	ualcore proiect'	1 CM4			~		Select the context to
STM32H747_d	ualcore_project1	1_CM4			~		Select the context to work with
STM32H747_di STM32H747_di	ualcore_project1 ualcore_project1	1_CM4 1_CM7	vigoti - STM2		×		
STM32H747_d STM32H747_d STM32H747_d	ualcore_project ualcore_project ?H747_dua.lo	1_CM4 1_CM7	oject1 - STM32	2H7 ✓	~		
STM32H747_di STM32H747_di	ualcore_project ualcore_project ?H747_dualo lication	1_CM4 1_CM7	oject1 - STM32	2H7 ✓			
STM32H747_dd STM32H747_dd	ualcore_project1 ualcore_project1 ? H747_dual d lication WARM startup_stm1	1 CM4 1_CM7 core_pro 32h747xx	_CM4.s	2H7 🗸	-		
STM32H747 di STM32H747 di STM32H747 di STM32 	ualcore_project ualcore_project ?H747_duate ication WARM startup_stm startup_stm	1 CM4 1_CM7 core_pro 32h747xx	_CM4.s	2H7 🗸			
STM32H747_dt STM32H747_dt □ ● STM32 	ualcore_project ualcore_project ?H747_duald ication WARM startup_stm startup_stm ser	1 CM4 1_CM7 core_pro 32h747xx	_CM4.s	2H7 🗸	•		
STM32H747_dt STM32H747_dt C ● STM32 	ualcore_project ualcore_project ? H747_dual d ication WARM startup_stm startup_stm ser ers	1 CM4 1_CM7 core_pro 32h747xx	_CM4.s	2H7 🗸			
STM32H747_dt STM32H747_dt C ● STM32 C ● STM32 C ● C ● C 0 C 0 C 0 C 0 C 0 C 0 C 0 C 0 C 0 C 0	ualcore_project ualcore_project ? H747_dual d ication WARM startup_stm startup_stm ser ers	1_CM4 1_CM7 core_pro 32h747xx 32h747xx	CM4.s CM7.s	2H7 🗸			



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.

EWARM	^	Name
		📕 settings
		STM32H747_dualcore_project1_CM4
		Project.eww
		startup_stm32h747xx_CM4.s
		startup_stm32h747xx_CM7.s
		STM32H747_dualcore_project1.ewd
		📂 STM32H747_dualcore_project1.ewp
		stm32h747xx_dtcmram_CM7.icf
		stm32h747xx_flash_CM4.icf
		stm32h747xx_flash_CM7.icf
		stm32h747xx_flash_rw_sram1_CM7.icf
		stm32h747xx_flash_rw_sram2_CM4.icf
		stm32h747xx_sram1_CM7.icf
		stm32h747xx_sram2_CM4.icf
	100	

Figure 471. 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 $\mathsf{Cortex}^{\texttt{B}}\text{-}\mathsf{M33}$ 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 TrustZone[®] 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.

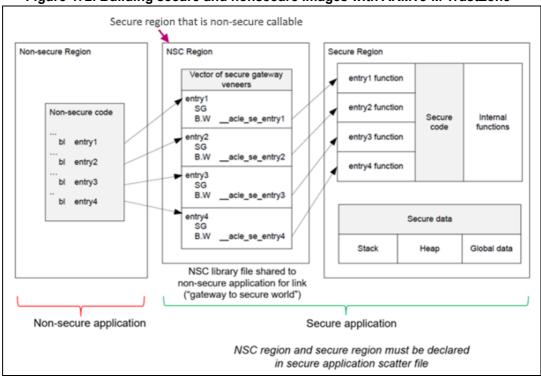


Figure 472. Building secure and nonsecure images with ARMv8-M TrustZone[®]



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* 473 (use TZ_BasicStructure_project_inCubeIDE.png) and *Figure* 474 (use STM32L5_STM32CubeMX_Project_settings_inCubeIDE.png).



Part and a second se	Project Explorer 🔀	🖻 🔄 🗸 🗖	
~	iii stm32l5_TZ_BasicStructure_project1		
	✓ DE stm32l5_TZ_BasicStructure project1 N	onSecure (in NonSecure)	
	> 🔊 Includes		
	> 🖓 Drivers		
	V 🖉 Src	iscal	
	> c main.c		
	stm32l5xx_hal_msp.c		
	> c stm32l5xx_it.c		
	> ic syscalls.c		
	> c sysmem.c		
	🔉 🖻 system_stm32l5xx_ns.c 🗡		
	V 📇 Startup		
	Startup_stm32l562cetx.s		
	V 🗁 Inc		
	⊨ ⊫ main.h		
	stm32I5xx_hal_conf.h		
	stm32l5xx_it.h		
	🗟 STM32L562CETX_FLASH.Id 🚩		
	🗟 STM32L562CETX_RAM.Id 🗡		
	✓ DE stm32I5_TZ_BasicStructure_project1_Se	ecure (in Secure)	
	> 🗊 Includes		
	> 🚑 Drivers		
	🗸 📴 Src		
	> 🖻 main.c		
	> 🖻 secure_nsc.c 🚩		
	> 1 stm32l5xx_hal_msp.c		
	> 🖻 stm32l5xx_it.c		
	> 🖻 syscalls.c		
	> c sysmem.c		
	> 🖻 system_stm32l5xx_s.c 🚩		
	🗸 🔁 Startup		
	> S startup_stm32l562cetx.s		
	🗸 🗁 Inc		
	📓 main.h		
	📓 partition_stm32l562xx.h 🚩		
	📓 stm32l5xx_hal_conf.h		
	📓 stm32l5xx_it.h		
	🗟 STM32L562CETX_FLASH.Id 🗲		
	🗟 STM32L562CETX_RAM.Id 🚩		
· · · · · · · · · · · · · · · · · · ·			



Figure 474. Project settings for STM32CubeIDE toolchain

Project Name	
stm32l5_TZ_BasicStructur	e_project1
Project Location	
C:\STM32CubeMX_Project	3
Application Structure	
Basic	✓ Do not generate the main()
	✓ □ Do not generate the main()
Toolchain Folder Location	
Toolchain Folder Location	Do not generate the main() s\stm32l5_TZ_BasicStructure_project1\
Toolchain Folder Location	

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.

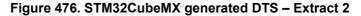
The device tree structure is very close to C language multiple level structures with the "root" (/) being the highest level then "peripherals" being sub-nodes described further in the hierarchy (see figures 475, 476 and 477).

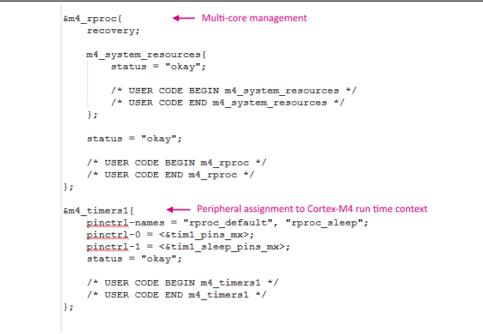
STM32CubeMX generation uses widely overloading mechanisms to complete or change some SOC devices definitions when user configurations require it.



```
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:
};
```







388/557

UM1718 Rev 46



```
&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 477. 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 478*). For each Device tree STM32CubeMX generates Device tree source (DTS) files.

	File	Window	Help	
Home 🔰 STM32MI	₽151CAAx <mark>〉 STM32MP1_</mark>	Test1.ioc - Project Manager	\rangle	
Pinout 8	Configuration	Clock Config	uration	Project Manager
Advanced Settings	DeviceTree Root Location C:\STM32CubeMX_Projects\STM Manifest Version	M32MP15_Project\DeviceTree	Brow	
	openstlinux-4.19-thud-mp1-1 Manifest Content: Firmware Name Com	9-01-11 munity Version	STM32MP15_ DeviceTree Drivers	kernel
	Linux 4.19 Cube STM32Cube U-Boot 2018.11 TF-A 2.0	FW_MP1 V1.0.0	L Inc Src Sw4STM32	📙 tf-a 📜 u-boo

Figure 478. 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.

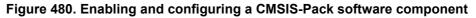
STM32CubeMX supports third-party and other STMicroelectronics embedded software solutions, delivered as software packs. STM32CubeMX enables to:

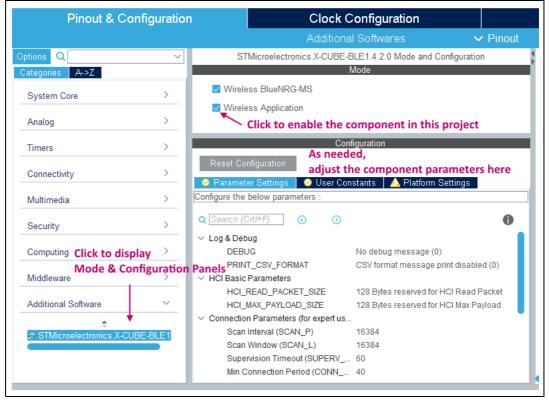
- 1. Install software packs and check for updates (see Section 3.4.5).
- 2. Select software components for the current project (see *Section 4.15*). Once this is done, the selected components appear in the tree view (see *Figure 479*).
- 3. Enable the software component from the tree view (see *Figure 480*). Use contextual help to get more details on the selection.
- 4. Configure software components (see *Figure 480*). This function is possible only for components coming with files in STM32CubeMX proprietary format.
- 5. Generate the C project for selected toolchains (see Figure 481).
 - 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.



Software Packs Component Selector				×
Packs	its for cont	ext: Cortex-M7	~	
Pack / Bundle / Component	Status	Version	Selection	C,
> RoweBots.I-CUBE-UNISONRTOS		5.5.0-4 也 😂	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 \checkmark	
BlueNRG-MS / Utils	\odot		✓	
✓ Device BLE1_Applications	\odot	6.1.0		
Application	\odot		SensorDemoBLESensor 🗸	
> STMicroelectronics.X-CUBE-BLE2		3.2.0 ~		
				Ok Cancel

Figure 479. Selecting a CMSIS-Pack software component







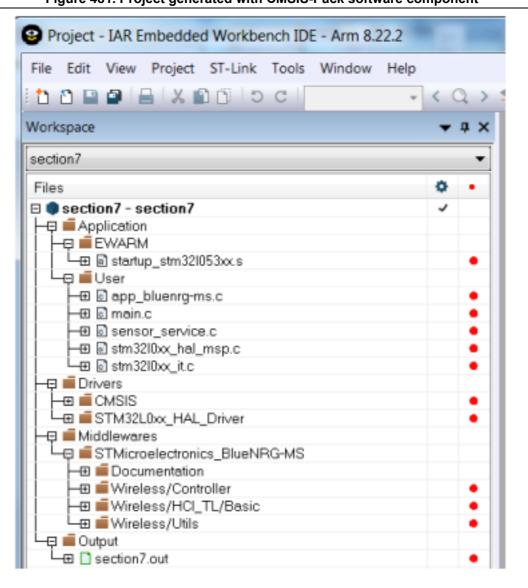


Figure 481. Project generated with CMSIS-Pack software component



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.
- 2. Select the **MCU Selector** tab and filter down the STM32 portfolio by selecting STM32F4 as 'Series', STM32F407 as 'Lines', and LQFP100 as 'Package' (see *Figure 482*).
- 3. Select the STM32F407VGTx from the MCU list and click **OK**.

New Project from a MCU Block Diagram Docs & Resources Datasheet Features T Buy Core STM32F407VG 1 High-performance foundation line, ARM Cortex-M4 core with DSP and FPU, 1 Mbyte Flash, 168 MHz CPU, ART Accelerator, Ethernet, FSMC STM32F4 Unit Price for 10kU (US\$): 6.57 ACTIVE Active LOEP100 Board: STM32F4DISCOVERY Check/Uncheck All The STM32F405xx and STM32F407xx family is based on the high-performance ARM® Cortex®-M4 32-bit RISC core operating at a frequency of up to 168 MHz. The Cortex-M4 core features a Floating point unit (FPU) single precision which supports all ARM single-precision data-processing instructions and data types. It also implements a full set of DSP instructions and a memory STM32F401 STM32F405/415 protection unit (MPU) which enhances application security STM32F407/417 protection unit (MPU) which enhances application security. The STM32F405xx and STM32F407xx training incorporates high-speed embedded memories (Flash memory up to 1 Mbyte, up to 192 Kbytes of SRAM), up to 4 Kbytes of backup SRAM, and an extensive range of enhanced I/Os and peripherals connected to two APB buses, three AHB buses and a 32-bit multi-AHB bus matrix. All devices offer three 12-bit ADCs, two DACs, a low-power RTC, twelve general-purpose 16-bit timers including two PWM timers for motor control, two general-purpose 32-bit timers. a true random number generator (RNG). They also feature standard and advanced communication interfaces. STM32F411 STM32F412 STM32F413/423 STM32F427/437 STM32F429/439 Features STM32F446 STM32F469/479 MCUs List: 4 items x∎ Package Active 5.644 STM32F407VE STM32F407VETx LQFP100 512 kBytes 192 kBytes Check/Uncheck All STM32F407VG STM32F407VGTx Active 6.57 LQFP100 1024 kBytes 192 kBytes 82 0 LQFP100 STM32F417VED STM32F417VE 5.991 LOFP100 512 kBytes 192 kBytes LQFP144 STM32F417VG STM32F417VGTx Active 6.917 LQFP100 1024 kBytes 192 kBytes LQFP176 UFBGA176 Other Price From 5.644 to 6.917 6.917 5.644 IO = 82

Figure 482. MCU selection

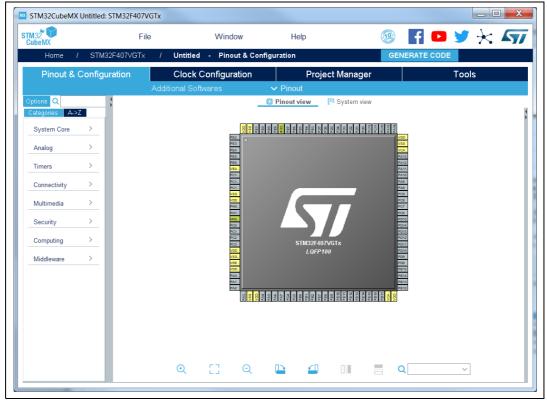
STM32CubeMX views are then populated with the selected MCU database (*Figure 483*). Optionally, remove the MCUs Selection bottom window by deselecting **Window > Outputs** submenu (see *Figure 484*).



STM32CubeMX U	Jntitled: STM32F407	/GTx							□ X
M32 D ubeMX	Fil	е	Window	Help		(19)	f 🗖	🗲 🕑	5
Home /	STM32F407VGTx	/ Untitled	d - Pinout & Con	figuration		GEN	ERATE COD	E	
Pinout & C	onfiguration	Clock	Configuration	Pro	ject Managei			Tools	
				✓ Pinout					
ptions Q	4 2		-	📋 Pinout view	System view				
Categories A->Z	-			-		-034			
System Core	<u>></u>		9 9 10 19 PE2	280 280 280 280 280 280 280 280 280 280		FC11 FC10 FA15 FA14			
Analog	>		PE3 PE4			VSS			
Timers	>		PES			PA1 PA1			
			PC1.			PAT			
Connectivity	>		PC1_ VSS			PAR			
Multimedia	>		PH0.			PC8			
Security	>		PH1_ NRS_ PC0			POI			
Computing	>		PC1 PC2			PD1	3		
			PC3	STM32F4 LQF1		PD1	0		
Middleware	>		VSS. VRE			PD6 PD6 PB1			
			PA0			PB1 PB1	4		
			PA2 3 8 8	704 745 747 747 747 748 181 181	e/ 69 611 612 612 613 613	8 5 <mark>5 8</mark>	2		
		Œ	[] Q					\sim	
/ICUs Selection Serie	Output	Lines		Mcu	P	ickage	F	Required Periphera	ls
STM32F4 STM32F4	STM32F	407/417	STM32F407V STM32F407V	ETx	LQFP100 LQFP100		None None	en priore	
STM32F4 STM32F4	STM32F		STM32F407V STM32F417V		LQFP100		None		

Figure 483. Pinout view with MCUs selection

Figure 484. Pinout view without MCUs selection window





UM1718 Rev 46

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 with the peripheral 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:

PD14 PD13 PD12 GPIO_Output PD11	
PD11 PD10	

Figure 485. GPIO pin configuration



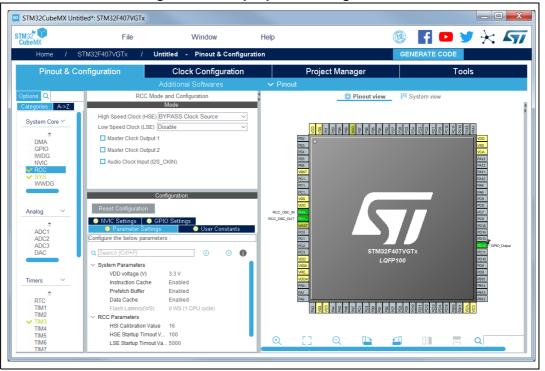
b) Enable a timer to be used as timebase for toggling the LED. This is done by selecting Internal Clock as TIM3 clock source from the peripheral tree (see *Figure 486*).

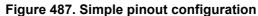
Pinout & Con	figuration	Clock Configuration	Proje
		Additional Softwares	✓ Pinout
Options Q		TIM3 Mode and Configuration	
Categories A->Z		Mode	
Analog >	Slave Mode Di	sable	\sim
	Trigger Source	Disable	\sim
Timers ~	Clock Source	Internal Clock	\sim
÷	Channel1 Dis	able	\sim
RTC TIM1	Channel2 Disa	able	\sim
TIM2	Channel3 Disa	able	\sim
✓ TIM3 TIM4	Channel4 Disa	able	\sim
TIM5	Combined Cha	annels Disable	\sim
TIM6 TIM7	Use ETR a	s Clearing Source	
TIM8	☐ XOR activa	ation	
TIM9 TIM10	One Pulse	Mode	
TIM11			
TIM12 TIM13		Configuration	
	Reset Configu	uration	
Connectivity >	🛛 📀 NVIC Setting		
			ser Constants
Multimedia >	Configure the belo	ow parameters :	
	Q Search (Crtl+	F) 💿 🕥	0
Security >	✓ Counter Setting	gs	-
Computing >			
Middleware >			
	L		





c) You can also configure the RCC to use an external oscillator as potential clock source (see *Figure 487*).





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 488. Save Project As window	Figure 488	3. Save	Project	As	window
------------------------------------	------------	---------	---------	----	--------

Save	Proje	ct As							X
Save In:	1.5	STM32Cube_s	simpleLedTo	oggle	\sim	G)	ഹ	C3	8.8. 8-
Folder n									
Files of	<u>T</u> ypes	STM32Cube	eMX project	Files					\sim
						Save		С	ancel
			_		_				

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 与 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 489*). An .ioc file is then generated for the project along with a .pdf and .txt reports with the same name.



STM32CubeMX Unt	titled: STM32F031E6	5Yx			
STM32 CubeMX	File)	Window	Help	(19)
Home /	STM32F031E6Yx	/ Untitled	- Project Manager	r	GENERA
Pinout & Co	nfiguration	Clock C	Configuration	Project Manage Generate Report	
	If the projec	name is generally u	v, you will be asked for a	no project is currently saved.	Browse

Answering **No** will require to provide a name and location for the report only. As shown in *Figure 490*, a confirmation message is displayed when the operation is successful.

Figure 490. Generate Project Report - Project successfully created

🔤 Ge	nerate Project Reports
0	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



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 491*).

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.

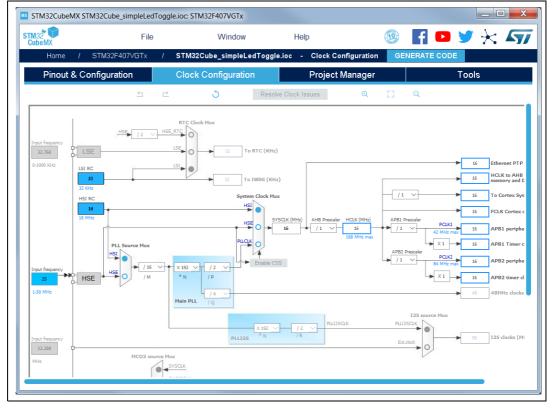
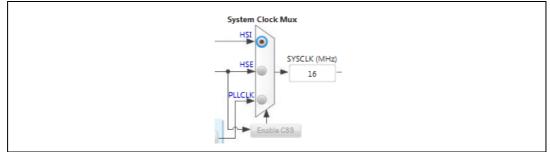


Figure 491. Clock tree view



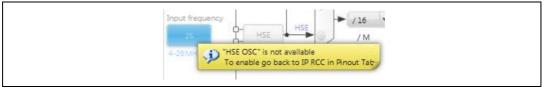
 Select the clock source (HSE, HSI or PLLCLK) that will drive the system clock. In the example taken for the tutorial, select HSI to use the internal 16 MHz clock (see *Figure 492*).





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 493*).

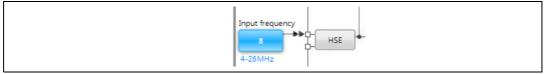
Figure 493. HSE clock source disabled



Other clock configuration options for the STM32F4DISCOVERY board:

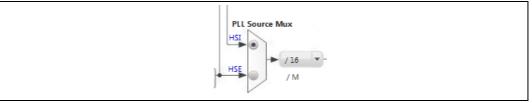
 Select the external HSE source and enter 8 in the HSE input frequency box since an 8 MHz crystal is connected on the discovery board:

Figure 494. HSE clock source enabled



 Select the external PLL clock source and the HSI or HSE as the PLL input clock source.

Figure 495	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 🗳 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 496*).

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.	10C-: STM152F407VGTX			_		
132 TO IbeMX	File	Window	Help		(19)	🗗 🖸 🎽	\times 5
Home / S	TM32F407VGTx / :	STM32Cube_simpleLedToggle.ioc	- Pinout & Confi	guration	G	ENERATE CODE	
Pinout & C	onfiguration	Clock Configuration		Project Manager		Tools	
otions Q		TIM3 Mode and Configuration		5	Pinout view	System view	
ategories A->Z		Mode					
	Slave Mode Disable		\sim				Middlewares
System Core >	Trigger Source Disal	ble	~				
Analog >	Clock Source Inter		~				
Analog	Channel1 Disable		~				
Timers ~	Channel2 Disable						
^	Channel3 Disable		~	System Core	Analog	Timers	Connectivity
RTC							
TIM1	Channel4 Disable		~	DMA 🔬		тімз 🥝	
TIM2	Combined Channels		\sim				
TIM4	🔲 Use ETR as Clea	ring Source		GPIO 😔			
TIM5	XOR activation						
TIM6 TIM7	One Pulse Mode						
TIM8							
TIM9 TIM10		Configuration)		
TIM11	Reset Configuration						
TIM12							
TIM13 TIM14		🗧 📀 User Constants 🛛 📀 NVIC Setting	gs 🥝 DMA Settings				
	Configure the below para	ameters :		-			
	Q Search (CrtI+F)	0	0				
Connectivity >	> Counter Settings						
	> Trigger Output (TRG0)	0) Parameters					
Multimedia >							
Constant N	- II.						
Security >							

Figure 496. 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.

Pinout & Cont	figuration	Clock Configuration	F
		Additional Softwares	✓ Pinout
Options Q		RNG Mode and Configuration	
Categories A->Z		Mode	
✓ RCC ✓ RNG RTC	Activated		
SDIO SPI1 SPI2 SPI3 SYS TIM1 TIM2 TIM3 TIM4 TIM5	Reset Configuratio	Configuration on gs I VIC Settings	
TIM6 TIM7 TIM8 TIM9 TIM10	🔺 w	arning: This IP has no parameters to be conf	igured.

Figure 497. 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 498. Timer 3 configuration window

Reset Configuration Parameter Settings 🛛 🤡 User Constants 🛛 🥥 NVIC Sett	tings 📔 🥝 DMA Settings	
nfigure the below parameters :	ings V DWA Settings	
Search (CrtI+F) ③ ③		6
Counter Settings		
Prescaler (PSC - 16 bits value)	0	
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)	
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 499. Timer 3 configuration

Co	onfiguration	
Reset Configuration		
OParameter Settings	tings 📔 🥝 DMA Settings	
Configure the below parameters :		
Q [Search (Crt1+F)] ③ ④		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		
Triana Fund Only dias		
Trigger Event Selection TIM_MasterOutputTrigger		
I'm_masterouputrigger		



- 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 500*.

- 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 500. Enabling Timer 3 interrupt

		Configuration			
Reset Configuration					
Parameter Settings	🥺 User Constants	😔 NVIC Se	ttings	🥺 DMA Settings	
NVIC Interrupt Table		Enabled	Pr	reemption 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.

		💭 Pino	ut view System	m view		
			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.		
RCC 🥑						

Figure 501. 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 502*).

Figure 502. GPIO mode configuration	
ODIO Mada and Operation	

			-		
		Con	figuration		
Peripherals					
2 RCC					
				Show o	only Modified Pins
				User Label	Modified
ation ·					
			1		
evel			Low		~
			Output Push Pull		~
Pull-down			No pull-up and no pull-down		~
put speed			Low		~
	s F) Signal on Pin n/a ation : evel Pull-down	≥ RCC s = Signal on Pin GPIO output level n/a Low ation :	Peripherals	Peripherals	Peripherals

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 503*):

- 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.



	D	MA Mode and	Configuration		
	_	Configur	ation		
DMA1 🥝 DMA2 🛛 🥺 MemToN	1em				
DMA Request	Stream		Direction		Priority
elect v elect M3_CH4/UP					
					Add Delete
MA Request Settings					Add Delete
MA Request Settings				Peripheral	Add Delete Memory
MA Request Settings			Increment Address	Peripheral	
		~	Increment Address Data Width		Memory

Figure 503. 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.

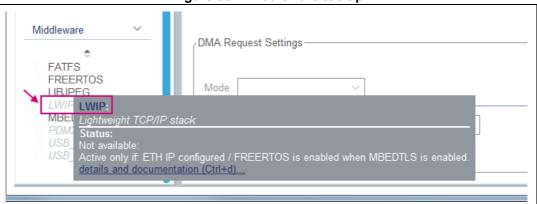


Figure 504. Middleware tooltip



1. Configure the USB peripheral from the **Pinout** view.

Figure 505. USB Host configuration	
------------------------------------	--

Options Q		USB_OTG_FS Mode and Configuration
Categories A->Z		Mode
USARTS USART6	Mode Host_Only	
USB_OTG 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 506. FatFs over USB mode enabled
FATFS Mode and Configuration
Mode
External SRAM
SD Card
USB Disk
User-defined
Configuration
Reset Configuration
😔 User Constants 🛛 😔 Platform Settings
Set Defines Advanced 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.

	IX ST	M32Cube_s	simpleLec	dToggle	e.ioc*: STM32F40	7VGTx									
M32 To ubeMX			File)	Wi	indow	He	lp		19	f	D	y	\star	$\overline{\mathbf{y}}$
Home	1	STM32F40	07VGTx	1	STM32Cube_si	mpleLedTog	gle.ioc -	Pinout &	Configurat	ion	GENER	ATE CO	ODE		
Pinout 8	& Co	onfiguratio	on		Clock Confi	guration		Project	Manager				Tool	s	
							🗸 Pinou								
ptions Q		A N				1	🕽 Pinout vie	w	System view						
Categories A	->Z	-						Middle	waree						
Timers	>							Middle	Wares						
0	>						FATF	s 🥝	USB_HO	ѕт 🕑					
Connectivity	/	-													
Multimedia	>		Systen	n Core	Analog	Т	imers	Conne	ctivity	Multin	nedia	5	Security		Comp
Security	>			•											
Computing	>		DMA	▲			13 😔	USB_F	-s 🖌			C F	rng 🔗		
	~		GPIC	0											
Middleware	~	-	NVIC	0											
V FATFS			RCC	0											
FREERTO	5		RUC	•											
MBEDTLS															
DM2PCM	ICE														
VISB_HOS	T														
			-												

Figure 507. 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	in	
Reset Configuration			
🥝 Advanced Settings	🥝 User Constants	🥺 Platform Settings	
	🥝 Set Defin		
Configure the below param	eters :		
Q Search (CrtI+F)		\odot	0
✓ Version			
FATFS version	R0.12	с	
✓ Function Parameters			
FS_READONLY (R	ead-only mode) Disabl	led	
	mization level) Disabl		
USE_STRFUNC (S	string functions) Enable	ed with LF -> CRLF conversio	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. ate() and f_rename() functions	

Figure 508. 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 509*.

- 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 509*).

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 V Min Vers	v9.30 v
	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 R	TOS selection.
	CortexM33NS		
	Enable multi-threaded support		
	Thread-safe Locking Strategy	Default – Mapping suitable strategy depending on R	TOS 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/STM32Cul	pe_FW_L5_V1.5.1

Figure 509. 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 510*, 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 proje	ct configuration file
Code Generator	□ Backup previo ✓ Keep User Co	oheral initialization as a pair of '.c/.h' files per pe usly generated files when re-generating de when re-generating sly generated files when not re-generated	eripheral
Advanced Settings	Template Settings		on) Settings

Figure 510. 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 511. Missing firmware package warning message

Project	: Manager Settings	J
▲	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 512. Error during download

Problem	n during Download and/or Unzip
	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.

W 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 Port	8080
Authentication	
Require Authentication V Remember my Credentials	
User Login JohnDoe	
Password ••••••	
Check Conn	ection
ОК	Cancel

Figure 513. Updater settings for download

4. Click **Check connection.** The check mark turns green once the connection is established.



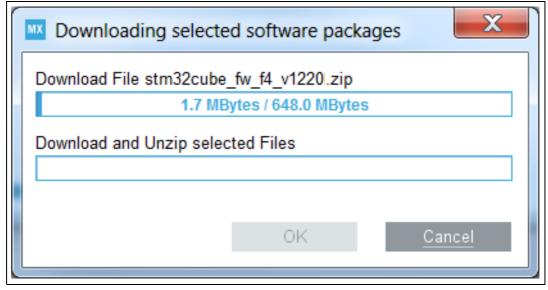


Updater Settings Connection Parameters
Proxy Server Type
O No Proxy
O Use System Proxy Parameters
Manual Configuration of Proxy Server
Manual Configuration of Proxy Server
Proxy HTTP do.it.mycompany.com Port 8080
Authentication
Require Authentication Remember my Credentials
User Login JohnDoe
Password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

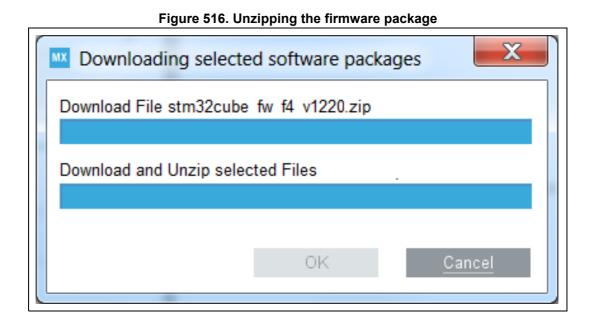
Figure 514. Updater settings with connection

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 515. Downloading the firmware package







6. Finally, a confirmation message is displayed to indicate that the C code generation has been successful.



e Generation
The Code is successfully generated under C:/STM32CubeMX_Projects/stm32f429_fatfs_sd_test
Open Folder <u>Open Project</u> 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	iry 🕶	» 🔠 • 🗍 🌘	2
 STM32Cube_SimpleLedToggle Drivers Inc Middlewares Projects Src 		Name Drivers Inc Middlewares Projects Src STM32Cube_SimpleLedToggle.ioc	

Figure 518. 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.

1. 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 517*).

File Edit View Tools Help Organize Include in library Share with ● STM32Cube_simpleLedToggle ● Drivers ● EWARM ● Inc ● MDK-ARM ● Svc ● SW4STM32 ● TrueSTUDIO	Burn New folder Image: The second secon	0
11 items	STM32Cube_simpleLedToggle_Configuration.pdf	

Figure 519. C code generation output: Projects folder



2. As an example, select .eww file to load the project in the IAR[™] EWARM IDE.

<u>File E</u> dit <u>V</u> iew <u>T</u> ools <u>H</u> elp		
Organize 🔹 🎇 Open 🔹 Burn New folde	er 🔠 🕶	
 STM32Cube_simpleLedToggle Drivers EWARM Inc MDK-ARM Src SW4STM32 TrueSTUDIO 	 Name settings STM32Cube_simpleLedToggle Configura Project.eww STM32Cube_simpleLedToggle.ewd STM32Cube_simpleLedToggle.ewp stm32f407xx_flash.icf stm32f407xx_sram.icf 	Date modifie 7/28/2015 2:4 7/28/2015 2:4 7/28/2015 2:3 7/28/2015 2:3 7/28/2015 2:3 7/16/2015 5:5 7/16/2015 5:5
Project.eww Date modified: 7/28/201	 • • • • • • • • • • • • • • • • • • •	۴

Figure 520. C code generation for EWARM



3. Select the main.c file to open in editor.

Project - IAR Embedded Workbench IDE	
ile <u>E</u> dit <u>V</u> iew <u>P</u> roject <u>T</u> ools <u>W</u> indow <u>H</u> elp	
) 🛩 🖬 🕼 🐇 🐘 🋍 🗠 🖂	- 🗸 🏷 🦮 🔁 🖪 🖻 🐡 🏟 🕼 🗟 👷 🥭 🕭
	x main.c * f0 ▼ >
6TM32Cube_simpleLedToggle Configuration	/* Includes
Files 😤 🕅	#include "stm32f4xx_hal.h"
	/* USER CODE BEGIN Includes */
I	/* USER CODE END Includes */
- 🛱 🔁 Application	
	/* Private variables
L-p 🗀 User	TIM_HandleTypeDef htim3;
Here 🔁 main.c	/* USER CODE BEGIN PV */
—⊞ 🖻 stm32f4xx_hal_msp.c	/* Private variables
└─⊞ 🖸 stm32f4∞_it.c	/* USER CODE END PV */
— 🕀 🧰 Drivers	
- 🖽 🧰 Output	/* Private function prototypes
	<pre>void SystemClock_Config(void);</pre>
	<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();
	<pre>/* Initialize all configured peripherals */</pre>
	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 */
STM32Cube_simpleLedToggle	
	↓ ← m → ·
ady	Ln 51, Col 1 System NL

Figure 521. 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	82	ð;
□ 🗇 STM32Cube	_SimpleLedToggle	× .	
- Application			
🗕 🕀 🗀 Drivers			
🕂 🕀 🗀 Example			*
🖵 🔁 Output			

Figure 522. IAR™ options

5. Click the ST-LINK category and make sure SWD is selected to communicate with the STM32F4DISCOVERY board. Click **OK**.

General Options 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 ST-LINK	ST-LINK Reset Normal Interface	Clock setup CPU clock: SWO clock:	Auto	▼ MHz kHz	
Third-Party Driver					

Figure 523. SWD connection

6. Select **Project > Rebuild all**. Check if the project building has succeeded.

Figure 524. Project building log

Messages
stm32f4xx_hal_tim.c stm32f4xx_hal_tim_ex.c stm32f4xx_it.c stm32f4xx_II_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 526. 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 527*).
- 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.

Import Pr	oject
mported Proje	act
:\STM32Cub	eMX_Projects\5_0_UM_Tuto1\STM32Cube_simpleLedToggle\STM32Cube_simpleLedToggle.ioc
mport MX Set	ttings
Import Pov	ver Consumption Calculator Settings
Import Pro	ject Settings
mport Pinout/	Clock Configuration/Configuration Settings
Automatic	Import
O Manual Im	port
	Pinning Status
	Peripherals Configuration
Periphera	
From STM FATES	To STM32F429ZITX
NVIC	
RCC	
RNG	V RNG
CVC	
Import Status	Try Import Show View Pinout ~
1.1	
	ing: STM32F407V(E-G)Ix alysis: C:\STM32CubeMX Projects\5 0 UM Tuto1\STM32Cube simpleLedToggle\STM32Cube simpleLedToggle.ioc project
	arysis Cristascument_projects_joum_intor(simscume_simplementoggie(simscume_simplementoggie) for project being imported is not the same as the Mou (STMS2F4920IIX) currently edited
Inc nou (Shortovis, Joana in one ribjeob being imported ib not one bame ab one nou (Dinbritishink) cartenory carted
Import Tr	у:
Importing	project completed
	OK Cancel

Figure 527. Import Project menu



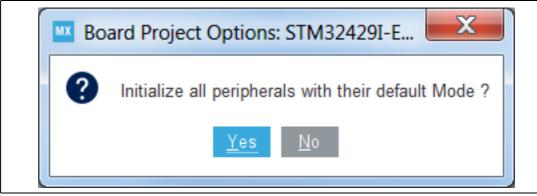
12 Tutorial 2 - Example of FatFs on an SD card using STM32429I-EVAL evaluation board

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 528*). The **Pinout** view is loaded, matching the MCU pinout configuration on the evaluation board (see *Figure 529*).

Figure 528. Board peripheral initialization dialog box



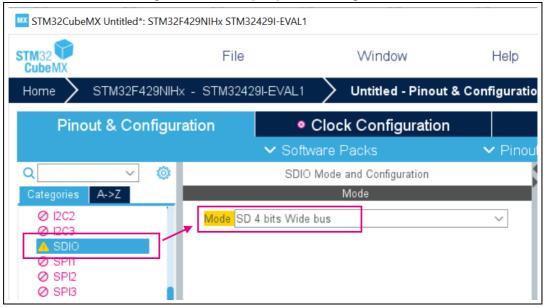


lew Project from a Board											
Board Filters	ა ~		Feat Larg	je Pi D	locs & Resol	u [🖵 Datas.		3	⊡ → St	art Pr
	√	H	C 3		electroni t and Exa		1324291	Evalua	tion Bo	ard	
Vendor	>		STM32 F4	M32 F4 Active		Unit F	rice (US	5):389.0			
Туре	~			Product is	s in mass pro	duction	Moun	ted devic	e: <u>STM32F</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	
Nucleo64		☆		STM3241G			STM32F			1	0
MCU Series	\sim										
Check/Uncheck All					· · · ·						
STM32F0		☆		STM32429I	Ev Active	389.0		STM32	0	0	1
			والمسطرة								
STM32F1											

Figure 529. Board selection

- 7. From the Peripheral tree on the left, expand the SDIO peripheral and select "SD 4 bits wide bus" (see *Figure 530*). 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 531*).

FATFS Mode and Configuration	
Mode	
External SDRAM	
External SRAM	
✓ SD Card	
USB Disk	
User-defined	

Figure 531. 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 532*).

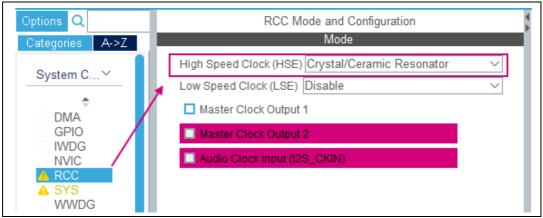


Figure 532. RCC peripheral configuration

b) Configure the clock tree from the clock tab (see *Figure 533*).

HSI RC	_		System Cloc	k Mux	
16	†		HSI	1	
16 MHz			HSE	SYSCLK (MHz) 168	AHB Prescaler HCLK (MHz)
			DUCK		180 MHz max
	PLL Source Mux		PLLCLK	l.	
	HSI		Enabl	le CSS	
25 HSE	HSE / 25 *	× 336 × /2 *N /P			
4-26MHz	/M	*N /P			
		L /7			
		Main PLL / Q			

Figure 533. 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.



Project STM32	Settings	Generate Report	
Project STM32	0		
C:\STM	Name F429NI-STM32F429I-EVAL Location 32CubeMX_Projects tion Structure		
Basic Toolcha	in Folder Location 32CubeMX_Projects\STM32F429NI-STM32F429 in / IDE	Do not generate the main() H-EVAL Generate Under Root	
Advanced Settings Minimu	m Heap Size 0x200 m Stack Size 0x400		

- 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 535*). 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 536*).

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 ST-LINK	Factory Settings Config Library Inker configuration file Inker configuration file PROJ_DIR\$\stm32f429xx_flash.icf Edit Configuration file symbol definitions: (one per line) Linker configuration file editor
Third-Party Driver TI XDS	Vector Table Memory Regions Stack/Heap Sizes CSTACK 0x800 HEAP 0x400

Figure 536. 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);
```

```
/*##-1- FatFS: Link the SD disk I/O driver \#\#\#\#\#\#\#\#\#\#\#/
```



```
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

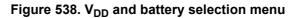
Follow the steps below to create the sequence (see Figure 537):

- 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 538*).
 - b) Select the Li-MnO2 (CR2032) battery. This step is optional. The battery type can be changed later on (see *Figure 538*).



M32 ubeMX		i*: STM32L476RGTx File		Mi	ndow		Help					6	10)		D y	- مد ا	G
ubeMX	<hr/>			vvi	luow		neip					5				\sim	
lome	STM32L476R	GTx 💙 Untitled	d - Tools	>									GENE	RATE C	ODE		
	Pinout & Conf	figuration		Cloc	k Configu	uration			Projec	t Manag	er				Tools		
						→ F	Power										
	STM32L476RGTx	~			_		tep —					- Sequen	ce —			nsitions (Checke
1	Series	STM32L4	Ne	w Step	Ū	D	ΞĴ	≣l	-		B	Ū	Ľ	ላይ	() ⊠ 0	n Log	Help
	Line	STM32L4 STM32L4x6								uence Tabl	le ——						
	Datasheet	025976 Rev4	5	Step	Mode		Vdd	Range/Sca	le (CPU/Bus Fre	q	Clock Co	nfig Pe	ripherals	Step Cu	rrent [Ouratior
		-	1		RUN	2.4		Range1-High				HSE		1:fs_5		1 m	
	T _A 25°C / V _{DD} 2.4V	· ~	2		STANDBY	2.4		NoRange	0 Hz			LSI RTC	RTC	•	464 nA	1 m	
	A 20 07 0D 2.40		3		WU_FROM			NoRange	4 MHz			MSI FAST			1.7 mA	20.1	
	TAmbient	25°C	4		RUN	2.4		Range1-High				ISE	RTC		2.16 mA	1 m	
		2.4 ~	5		RUN	2.4			16 MHz			HSE	ADC	1:fs_5	4.47 mA	1 m	
	V _{DD}	2.4 🗸	6		SLEEP	2.4			16 MHz			HSE			589 µA	1 m	
					RUN	2.4			16 MHz			HSE		1:fs_5		1 m	
	Li-MnO2(CR2032) (1x1) ~	8		STOP1	2.4		NoRange NoRange	0 Hz			ALL CLOC HSI16	K USA	RIT	6.65 µA 1.62 mA	1 m	
		,	10		WU_FROM_ RUN	2.4			16 MHz			HSE	DTC		1.62 mA 1.89 mA	6.3 1 m	
	Change		10		STANDBY	2.4		NoRange	0 Hz			LSI RTC	RIC	USARTI	464 nA	1 m	
					STANDDT	2.4		Norvange				LOINTO			404 114		•
					Dis	play Cho	ices —										
	In Series 1 🌲	In Parallel 1 🌲	Select	your Pr	eferred Displa	y Plot:	All Step	os 🚿	0								
	Capacity	225.0 mAh		·	· ·	·											
									Consumpt	ion Profile	by Step)					
	Self Discharge	0.12 %/month	6	1					5:RUN		7	RUN					
	Nominal Voltage	3.0 V	25		1:RUN				. 1		1						
			Consumption (mA)		T.KON			_				_					
	Max Cont Current	3.0 mA	<u></u> 4	+				4:RUN									
			i i i i i i i i i i i i i i i i i i i					4:RUN WSTDBY							10:RUN		
	Information Notes	>	Ē				1	***						9:W	TOP1		
			- <u>1</u> <u>2</u> 2				-			6:SLEEP				1			
	Help	>	8 1			2:STAN	IDBY						8:STO	1		11:ST/	NDBY
	P					ŧ				-			ŧ			ŧ	
			0		0.5 1.0	1.5	2.0	2.5 3.0	3.5 4.0	4.5 5	.0 5.5	6.0	6.5	7.0 7	.5 8.0	8.5	9.0
				0.0 (1.5 1.0	1.5	2.0	2.5 3.0	5.5 4.0			6.0	6.5	7.0 7	.5 8.0	8.5	9.0
										Time (m	5)	_					
									-Idd by St	ep — Avera	ge Current						
					ime / Ta M							_			Consum		

Figure 537. Power Consumption Calculation example



	\sim
STM32L4 STM32L4x6 025976_Rev4	
	\sim
25°C 2.4	\sim
(1)	\sim
Reset	
In Parallel 1 韋	
225.0 mAh	
0.12 %/month	
3.0 V	
3.0 mA	
	STM32L4x6 025976_Rev4 25°C 2.4 x1) Reset In Parallel 1 225.0 mAh 0.12 %/month 3.0 V

UM1718 Rev 46



- 5. Enable the **Transition checker** to ensure the sequence is valid (see *Figure 538*). 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 538.
 - 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 539*).
 - Some peripherals for which consumption is unavailable or negligible are highlighted with '*' (see *Figure 539*).

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 539. 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 540*).

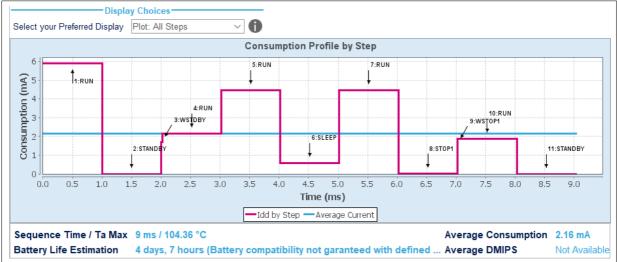


Figure 540. sequence results before optimization

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.



UM1718 Rev 46

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 541).

] 🕳
🔤 Edit Step			M Edit Step
Reset Step Settings Enal	ble All IPs Disable All IPs Ena	ble IPs from Pinout	Reset Step Settings Enable All IPs Disable All IPs Enable IPs from Pinout
Powe	er/Memory	Peripherals Selection	Power/Memory Peripherals Selection
Power Mode	RUN 🗸	Peripherals	
Power Range	Range1-High 🗸 🗸	✓ ADC1	Power Range Range1-High - IWDG
Memory Fetch Type	FLASH/ART/Cache V	fs 1 Msps	Memory Fetch Type FLASH/ART/Cache V Buffer OFF
V _{DD}	2.4 ~	✓ fs_5_Msps	V _{DD} 2.4 V - LPTIM1
		V-ADC2	Votage Source Battery
Voltage Source	Duttory	fs_10_ksps	
С	Clocks	Is_1_Msps ✓ fs 5 Msps	Clocks V OPAMP1
CPU Frequency 24	24 MHz 🗸	✓ ADC3	CPU Frequency 16 MHz Low_Power
Interpolation Ranges	\sim	_	Interpolation Ranges
User Choice (Hz)		- 🗌 fs_1_Msps	User Choice (Hz)
Clock Configuration	ISE V	✓ fs_5_Msps AHB APB1 Bridge	Clock Configuration
Clock Source Frequency 2	24 MHz 🗸	AHB_APB2_Bridge	Clock Source Frequency 16 1/z V PWB
			Optimal Settings QUADSPL
	hal Settings	- 🗆 CRC	Step Duration 0.1 ms V - RNG
Step Duration 1	ms ∨	V-DAC1	Additional Consumption 0 mA V RTC
Additional Consumption 0	mA 🗸	OUT1+OUT2-Buffer OUT1+OUT2-Buffer OUT1+OUT2-Buffer	Results SAI1
	esults	- OUT1+OUT2-Buffer	Step Consumption 2.16 mA
Step Consumption 5.9 mA		- OUT1-Buffer_OFF-	Without Peripherals 2.16 mA
Without Peripherals 3.18 mA		OUT1-Buffer_ON-N	Peripherals Part 0 nA (A: 0 nA - D: 0 nA)
	(A: 2.58 mA - D: 141.6 µA)	- OUT1-Buffer_ON-V	Ta Max (*C) 104.77
Ta Max (°C) 104.36			Warnings
The step consumption is higher t	than the max continuous current (3 mA)		
Available use cases: 1 Max: 52	20		Available use cases: 1 Max: 520

Figure 541. 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 542).

p 5 optimization
Optimized settings
Edit Step
Reset Step Settings Enable All IPs Disable All IPs Enable IPs from Pinout
Power/Memory Peripherals Selection
Power Range NoRange
Memory Fetch Type FLASH/ART/Cache
V _{DD} 2.4 V [fs_5_Msps
Voltage Source Battery V ADC2
fs_10_ksps
Clocks fs_1_Msps
CPU Frequency 2 MHz
Interpolation Ranges V ADC3
User Choice (Hz)
fs_1_Msps
Clock Configuration
Clock Source Frequency 2 MHz
Optional Settings — AHB_APB2_Bridge
Step Duration
Additional Consumption 0 mA V PAC1
Step Consumption 271 µA
Without Peripherals 271 µA
Peripherals Part 0 nA (A: 0 nA - D: 0 nA) OUT1-Buffer ON-N
Ta Max (°C) 104.97 — OUT1-Buffer ON-M
Warnings

Figure 542. 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 543).

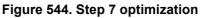
🚥 Edit Step	Optimized settings
Reset Step Settings Enable All IPs Disable All IPs Enable IP:	Ps from Pinout Edit Step
Power Mode SLEEP V Peri	Peripherals Selection Reset Step Settings Enable All IPs Disable All IPs Enable IPs from Pinout Power/Memory Power/Memory Peripherals
V _{DD} 2.4 Votage Source	Image: http://www.image.com/image
Interpolation Ranges	Image: base state s
	Image: bit is a figure of the second secon
Step Durationms	□ ous-matrix Optional Settings - AHB_APB2_Bridge □ CAN1
Results Step Consumption 589 µA Wthout Peripherals 589 µA Peripherals Part 0 nA (A: 0 nA - D: 0 nA)	COUTI+OUT2-Buffer OUTI+OUT2-Buffer OUTI+OUT2-Buffer OUTI+OUT2-Buffer OUTI+OUT2-Buffer



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 544*).



👯 New Step		_	_		Optimized	Settings	
Reset Step Settings Enable All IPs Disable	All IPs Enable			🚾 Edit Step			
Power/Memory		Peripherals Selection	Enal				
Power Mode RUN	~		USART1			nable IPs from Pinout	led Per
Power Range Range2-Medium	~	COMP_Ultra_Low_Pow		Power Mode	ower/Memory	UU12-Butter_UN-Wo	
Memory Fetch Type FLASH/ART/Cach	ie ~	COMP_Ultra_Low_Pow			LOWPOWER_RUN N		ARI1
V _{DD} 2.4	~	COMP_Ultra_Low_Pow		Power Range	NoRange	DMAT	
		- 🗆 TIM1		Memory Fetch Type	FLASH/ART/Cache V		
Voltage Source Battery	~	- 🗌 TIM2		VDD	2.4 ~	- FW	
Clocks		– 🗌 ТІМЗ		Voltage Source	Battery V	- GPIOA	
CPU Frequency 16 MHz	~	- TIM4			Clocks	— 🗌 дріов	
Interpolation Ranges	~	— 🔲 ТІМБ		CPU Frequency	User-defined V		
User Choice (Hz)		- TIM6 - TIM7		Interpolation Ranges	100 kHz - 2 MHz	- GPIOD	
		- TIM8		User Choice (Hz)	100000	- [12C1	
Clock Configuration HSE	~	- TIM15		Clock Configuration	MSI		
Clock Source Frequency 16 MHz	~	- TIM16		Clock Source Frequency	100 kHz	- 🗆 12C3	
Optional Settings		- TIM17				- 🗆 IWDG	
Step Duration 1	ms ~	- 🗆 TS			otional Settings	↓ LCD	
Additional Consumption 0	mA ~	- 🗌 TSC		Step Duration	ms >	Buffer_OFF	
-	ma ~	UART4		Additional Consumption 0	mA ~		
Results		UART5			Results		
Step Consumption 1.89 mA		USART1		Step Consumption 42.06	μΑ	- OPAMP1	
Without Peripherals 1.81 mA		USART2		Without Peripherals 41.8 µ	A	Low_Power	
Peripherals Part 84.8 µA (A: 0 nA - D: 84.8 µA)		USART3		Peripherals Part 260 n/	A (A: 0 nA - D: 260 nA)	Normal	
Ta Max (°C) 104.8		- WWDG		Ta Max (*C) 105		Y OPAMP2	
			1				



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 545).

Figure 545. Step 8 optimization

MX New Step		_	K Edit Step	Optimized Settings
	nable All IPs Disable All IPs Enab		Reset Step Settings E	Enable All IPs Disable All IPs Enable IPs from Pinout
	Power/Memory	Peripherals Selection Er	ab	wer/Memory Ena
Power Mode	STOP0 ~	- COMP_Medium_Power USART1*		
Power Range	NoRange	COMP_Medium_Power	Power Mode	STOPT *
	Norvange	COMP_Medium_Power	Power Range	NoRange
Memory Fetch Type	n/a 🗸	COMP_Medium_Power		
VDD	3.0 ~	COMP_Medium_Power	Memory Fetch Type	
		COMP_Medium_Power	Vpp	2.4
Voltage Source	Battery ~	COMP_Medium_Power		Battery V D 2021
	Clocks	COMP_Medium_Power	Voltage Source	1203
CPU Frequency	0 Hz ~	COMP_OFF_VREFBUF		- Clocks I IWDG
Interpolation Ranges		- COMP_OFF_VREFBUF	CPU Frequency	
merpolation realiges		COMP_OFF_VREFBUF		
User Choice (Hz)		COMP_Ultra_Low_Pow	Interpolation Ranges	
Clock Configuration	ALL CLOCKS OFF ~	COMP_Ultra_Low_Pow	User Choice (Hz)	− ✓ LPUART1*
-		- COMP_Ultra_Low_Pow		V OPAMP1
Clock Source Frequency	0 Hz Y	COMP_Ultra_Low_Pow	Clock Configuration	ALL CLOCKS OFF Low_Power
	ptional Settings	COMP_Ultra_Low_Pow	Clock Source Frequency	0 Hz V Normal
Step Duration 1	ms ~	- COMP_Ultra_Low_Pow		tional Settings
		COMP_Ultra_Low_Pow		Low_Power
Additional Consumption 0	mA ~	COMP_Ultra_Low_Pow	Step Duration 1	ms V Normal
	Results	UART4*	Additional Consumption 0	mA V PVD/BOR
Step Consumption 111 µ	A	- UART5*		Results RTC*
Without Peripherals 111 µ	Δ.	- 🗸 USART1*		- SWPMI1*
		- USART2*	Step Consumption 6.65 µ/	Y SYS-VREFBUF/COMP1
Peripherals Part 0 nA	(A: 0 nA - D: 0 nA)	- USART3*	Without Peripherals 6.65 µ/	A COMP_High_Spee
Ta Max (°C) 104.99	9	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 546*).

The example given in Figure 547 shows an average current consumption reduction of 155 μ A.

🚥 Edit Step	-	Edit Step Optimized S	ettings
	able IPs from Pinout	Reset Step Settings Enable All IPs Disable All IPs Ena	able IPs from Pinout
Power/Memory	Peripherals Selection — Enabled F	Power/Memory	Peripherals Selection — Enabled Periphe
Power Mode RUN V	Peripherals RTC USART1	Power Mode RUN ~	OUT2-Buffer_ON-V (RTC) (LPUART1)
Power Range Range2-Medium V		Power Range Range2-Medium V	DFSDM1
Memory Fetch Type FLASH/ART/Cache V	fs_1_Msps	Memory Fetch Type FLASH/ART/Cache V	- DMA1
V _{DD} 2.4 ~	fs_5_Msps	V _{DD} 2.4 V	
Votage Source Battery V	V- ADC2	Votace Source Battery	- GPIOA
	- ☐ fs_10_ksps - ☐ fs 1 Msps		— 🔲 GPIOB
Clocks	5.5_Msps	Clocks	- GPIOC - GPIOD
	V- ADC3	CPU Frequency	
Interpolation Ranges	fs_10_ksps	Interpolation Ranges	- 12C1
User Choice (Hz)	fs_1_Msps	User Choice (Hz)	- 🗌 12C2
Clock Configuration HSE ~	AHB_APB1_Bridge	Clock Configuration	- 12C3 - 1WDG
Clock Source Frequency 16 MHz ~	AHB_APB2_Bridge	Clock Source Frequency 1 MHz ~	✓ LCD
Optional Settings	- CAN1	Optional Settings	Buffer_OFF
Step Duration 1 ms V	- CRC	Step Duration 1 ms ~	- LPTIM1
Additional Consumption 0 mA	OUT1+OUT2-Buffer	Additional Consumption 0 mA ~	- CLPTIM2
Results	OUT1+OUT2-Buffei	Results	V OPAMP1
Step Consumption 1.89 mA	OUT1+OUT2-Buffet	Step Consumption 234.2 µA	- Low_Power
Without Peripherals 1.81 mA	OUT1-Buffer_OFF- OUT1-Buffer_ON-N	Without Peripherals 232 µA	Normal
Peripherals Part 84.8 µA (A: 0 nA - D: 84.8 µA)	OUT1-Buffer ON-V	Peripherals Part 2.2 µA (A: 0 nA - D: 2.2 µA)	✓ OPAMP2
	Narnings	Ta Max (°C) 104.97	Varnings

Figure 546. Step 10 optimization

See Figure 547 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.

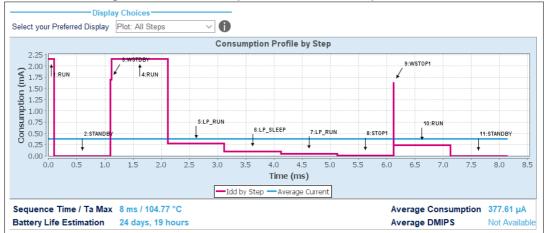


Figure 547. Power sequence results after optimizations



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 548*).





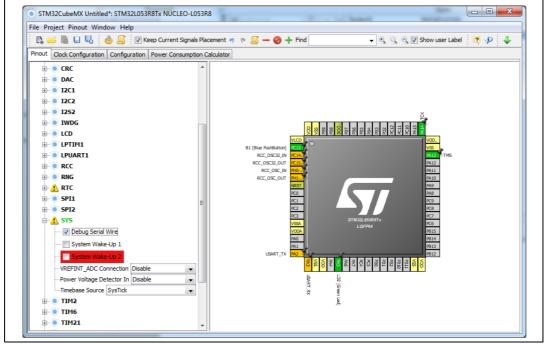
Figure 548. Selecting NUCLEO_L053R8 board



14.3 Selecting the features from the Pinout view

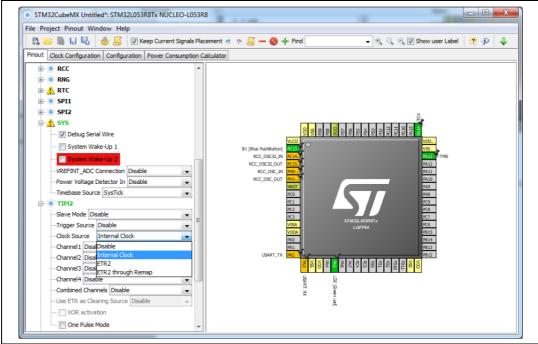
1. Select Debug Serial Wire under SYS (see *Figure 549*).





2. Select Internal Clock as clock source under TIM2 peripheral (see *Figure 550*).

Figure 550. Selecting TIM2 clock source



UM1718 Rev 46



3. Select the Asynchronous mode for the USART2 peripheral (see *Figure 551*).

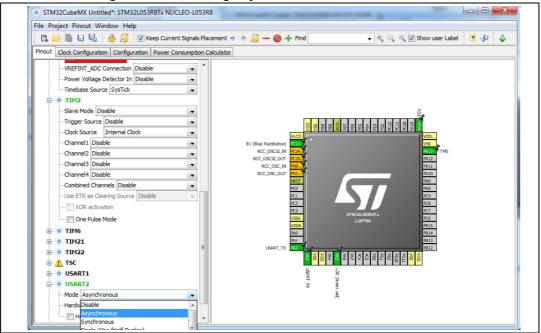
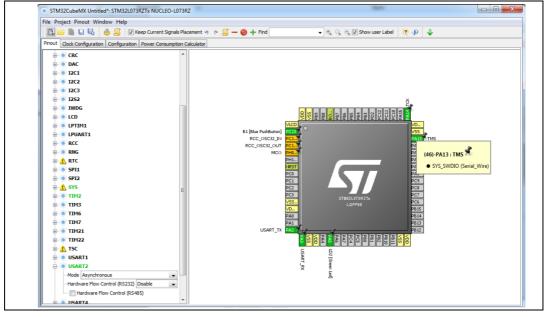


Figure 551. Selecting asynchronous mode for USART2

- 4. Check that the signals are properly assigned on pins (see *Figure 552*):
 - 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 553*).

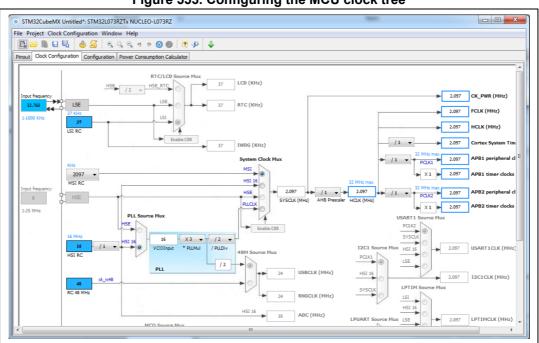


Figure 553. Configuring the MCU clock tree



Configuring the peripheral parameters from the 14.5 **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 554).
- 2. Click **OK** to apply the changes and close the window.

♥ USART2 Configuration ♥ Parameter Settings ♥ User Constants ♥ NVIC Settings ♥ DMA Settings Configure the below parameters : Search (CrtH+F) ♥ ● □ Basic Parameters ● ● ● □ Basic Parameters ● ● ● ● □ Basic Parameters ●
Configure the below parameters : Search : Search (CrtH+F) 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
Search (CrtI+F) Basic Parameters Baud Rate 9600 Bits/s Word Length 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
Search (CrtI+F) 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
Basic Parameters Baud Rate 9600 Bits/s Word Length 8 Bits (including Parity) Parity None Stop Bits 1 Advanced Parameters Data Direction Data Direction Receive and Transmit Over Sampling 16 Samples
Baud Rate 9600 Bits/s Word Length 8 Bits (including Parity) Parity None Stop Bits 1 Advanced Parameters Data Direction Over Sampling 16 Samples
Word Length 8 Bits (including Parity) Parity None Stop Bits 1 Advanced Parameters Data Direction Data Direction Receive and Transmit Over Sampling 16 Samples
Parity None Stop Bits 1 Advanced Parameters Data Direction Data Direction Receive and Transmit Over Sampling 16 Samples
Stop Bits 1 Advanced Parameters Data Direction Data Direction Receive and Transmit Over Sampling 16 Samples
Advanced Parameters Data Direction Receive and Transmit Over Sampling 16 Samples
Data Direction Receive and Transmit Over Sampling 16 Samples
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 Enable
MSB First Disable
Apply Ok Cancel



- UM1718
- 3. Click **TIM2** and change the prescaler to 16000, the Word Length to 8 bits and the Counter Period to 1000 (see *Figure 555*).

Search · Sea		
Search . Sea	arch (Crtl+F) 🕹 🔶	
Counter S	Gettings	
Pres	scaler (PSC - 16 bits value)	16000
Cou	nter Mode	Up
Cou	nter Period (AutoReload Register - 16 bits valu	1000
Inte	rnal Clock Division (CKD)	No Division
🗉 Trigger Ou	utput (TRGO) Parameters	
Mas	ter/Slave Mode	Disable (no sync between this TIM (Master) and its Slaves
Trigg	ger Event Selection	Reset (UG bit from TIMx_EGR)
Counter Pa	riod (AutoReload Register - 16 bits value))

Figure 555. Configuring TIM2 parameters



4. Enable TIM2 global interrupt from the **NVIC Settings** tab (see *Figure 556*).

TIM2 Configuration		x
Parameter Settings 🔗 User Constants 🔗 NVIC Settings	ntering DMA Settings	
Interrupt Table	Enabled	Preemption Priority
TIM2 global interrupt	V	0
		pply Ok Cancel
		pply Ok Cancel

Figure 556. 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 557*).

oject Code Generator A	Advanced Settings	
Project Settings		
Project Name		
Nucleo_L073_UART_Com	nm	
Project Location C:\STM32CubeMX_Projec	sta\Tu tarial	
C. priviszcubelitx_Projec		
Toolchain Folder Location	1	
C:\STM32CubeMX_Projec	cts\Tutoriel\Nucleo_L073_UART_Comm	
Toolchain / IDE		
EWARM	▼ Generate Under Root	
Linkin		
Linker Settings Minimum Heap Size Minimum Stack Size	0x200	
Minimum Stack Size	0×400	
	je	
Mcu and Firmware Package Mcu Reference		
Mcu Reference STM32L073RZTx	and Version	
Mcu Reference		
Mcu Reference STM32L073RZTx Firmware Package Name a		
Mcu Reference STM32L073RZTx Firmware Package Name a		
Mcu Reference STM32L073RZTx Firmware Package Name a		

Figure 557. 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 558*, and click **OK** to generate the code.

Project Settings	×
Project Code Generator Advanced Settings	
STM32Cube Firmware Library Package	
Copy all used libraries into the project folder	
 Copy 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 Image: Set all free pins as analog (to optimize the power consumption) Image: Enable Full Assert	
Template Settings	
Select a template to generate customized code	Settings
Ok	Cancel

Figure	558.	Generating	the	code
--------	------	------------	-----	------

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 */



UM1718 Rev 46

```
/* 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 */
```

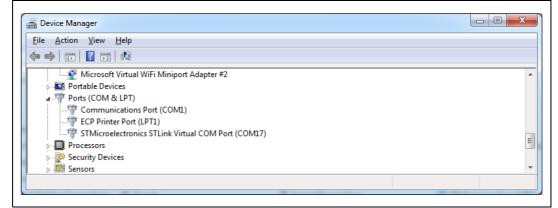
14.8 Compiling and running the project

- 1. Compile the project within your favorite IDE.
- 2. Download it to the board.
- 3. Run the program.

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 559*).

Figure 559. Checking the communication port





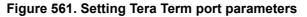
UM1718 Tutorial 4 - Example of UART communications with an STM32L053xx Nucleo board

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 560*).

Port:	COM17 • OK
<u>B</u> aud rate:	9600 -
<u>D</u> ata:	8 bit - Cancel
P <u>a</u> rity:	none 🔹
<u>S</u> top:	1 bit ▼ <u>H</u> elp
Elow control:	none 🔹
Transmit dela O mse	iy c <u>/c</u> har 0 msec <u>/l</u> ine

Figure 560. 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 561*).



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	2CubeMX 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:

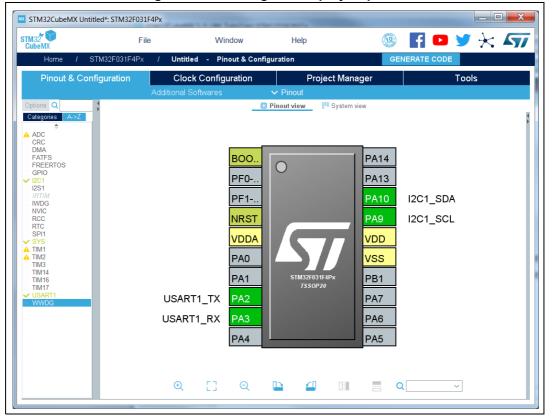


Figure 562. Existing or new project pinout

2. Go to the **Pinout** menu and select **List Pinout Compatible MCUs**. The **Pinout compatible** window pops up (see *Figure 563* and *Figure 564*).

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 564* 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 563* 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 564* for an example).

Figure 563. List of pinout compatible MCUs - Partial match with hardware compatibility

MCUs Filters	MCUs List: 104 Item						
Series :	MCU	Package	Flash	Ram	Signals to remap	Comments	Ē
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	
0	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	, remaps from Pir	1(8)-PA2 to	o Pin(30)-f	PA9	Need HW change	
Ignore System Pins	STM32FUSART1_RX					Need HW change	
	STM32F I2C1_SCL rer	maps from Pin(1	7)-PA9 to	Pin(6)-PF		Need HW change	
Search	STM32F US 15 SDA Te	maps from Pin(*	10)-PA 10 (0 Pin(5)-P		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	
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 Power Pins ✓ Ignore System Pins							

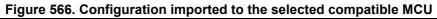


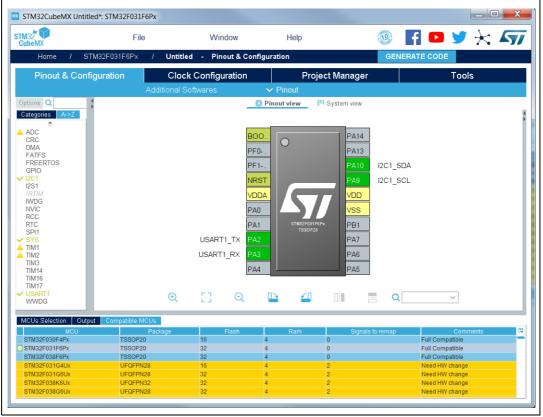
- UM1718
- 3. Then, select an MCU to import the current configuration to, and click **OK**, **Import**:

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
	CTM22E070DDTy	LOEDR4	100	10	4	Nood LIW obongo

Figure 565. 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:

- 1. 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 567*).

STM32CubeMX Untitle	ed*: STM32F401	CCFx							_ [X
STM32 CubeMX	File	е	Window	Help)	19	f 🗖	9	\star	57
Home / ST	M32F401CCFx	/ Untitled -	Pinout & Confi	iguration		GEN	IERATE COI	DE		
Pinout & Confi	iguration	Clock Con	ifiguration		Project Manage	r		Tools	s	
				✓ Pinout						
Options Q Categories A->Z	~ /×	c-Embedded.Middleware. Mo		nfigurat	🙆 Pi	nout view	Syste	n view		
System Core	>	VcloneCommon	CycloneCommon	•				Mic	ldlewar	es
Analog	>									
Timers	>							Additic	onal Sof	tware
Connectivity	>									
Multimedia	>							Mide	dleware (2
Computing	>	Config	uration	- II	System Core	Analog		Timers	(Connectiv
Middleware	>									
Additional Software	~									
 Oryx-Embedded.Mic 	ddleware									

Figure 567. 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.



UM1718

6. Check that no parameters can be configured in the **Configuration** tab (see *Figure 568*).

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
	Middleware 🥥

Figure 568. Pack software components: no configurable parameters

7. Select the **Project manager** project tab to specify project parameters (see *Figure 569*), 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:\STM32CubeMX Application Structt		Browse
Code Generator	Basic Toolchain Folder L	✓ □ Do no pocation _Projects\Oryx_project1\	it generate the main() erate Under Root
Advanced Settings	Linker Settings — Minimum Heap Siz Minimum Stack Si		
	STM32Cube FW_I	Name and Version -4 V1.22.0	

Figure 569. 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 570*).

ile Edit View Project ST-Link Tools Window Help		
1 1 🖬 🖬 🔚 🗶 🛍 🗇 I 5 C I 🚽 🗸 Q >	\$ ►E	< 🤤
orkspace	-	ąх
roject1		•
- Files	\$	•
🕽 🏶 project1 - project1 *	~	
–🖓 🛋 Application		
		٠
		•
—⊞ 🖬 Drivers —⊟ 🖬 Middlewares		•
Lep Oryx-Embedded.Middleware.1.7.8		
L ⊡ i cpu_endian.c		
-g ■ CycloneCommon/DateOoTimeOoHelperOoRoutines		
L → E o date_time.c		
http 📫 CycloneCommon/RTOSOoPort/None		
└─⊞ 🗈 os_port_none.c		٠
—⊞ 🛋 Output		

Figure 570. 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 571)
- 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 572*).
- Mobile application (see *Figure 573*): STMicroelectronics BlueNRG application for iOS[®] or Android[™]

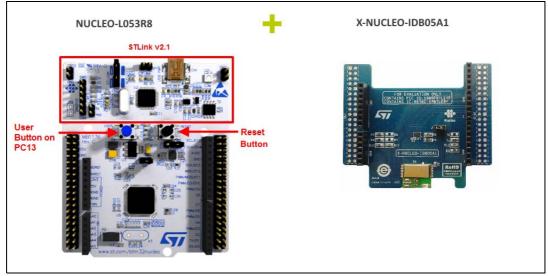


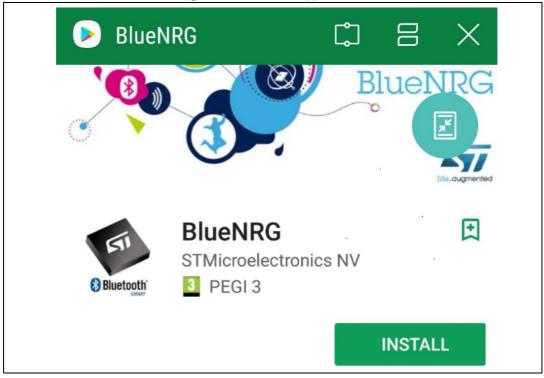
Figure 571. Hardware prerequisites



Embedded Sof	tware Packages Manager		X	
STM32	Cube MCU Packages and embedded software packs releases			
Release	es prormation was last refreshed 3 hours ago.			
TM32Cube MCU	Packages Alibaba Oryx-Embedded STMicroelectronics	IwIP		
Descriptio	n	Installed Version	Available Version	
STM32L0				
STM32C	ube MCU Package for STM32L0 Series	1.10.0	1.10.0	
STM32C	ube MCU Package for STM32L0 Series	1.9.0	1.9.0	
STM32	Embedded Software Packages Manager			×
	STM32Cube MCU Packages and embedded softv	vare packs releases	i i i i i i i i i i i i i i i i i i i	
etails	Releases Information was last refreshed 3 hours ago.	1		
	STM32Cube MCU Packages Alibaba Oryx-Embedded	TMicroelectronics	lwIP	
	Description		Av	ailable Version
	▼ X-CUBE-BLE1			
From Local	BLE stack and sample applications for BlueNRG-MS	nodule		1.1.0
	► X-CUBE-MEMS1			
	Details			
		/	/	
	From Local From Url Refr	sh Install N	low Remove Now	Close

Figure 572. Embedded software packages

Figure 573. 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 574):
 - 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 M	MCU Packages and embedded software packs releases			
	mation was last refreshed 3 hours ago.			-
STM32Cube MCU Packag	ges Alibaba Oryx-Embedded STMicroelectronics I	wIP		
Description		Installed Version	Available Version	
▼ STM32L0				
STM32Cube M0	CU Package for STM32L0 Series	1.10.0	1.10.0	
STM32Cube M0	CU Package for STM32L0 Series	1.9.0	1.9.0	
STM32 MX En	nbedded Software Packages Manager			×
Details	STM32Cube MCU Packages and embedded software Releases Information was last refreshed 3 hours ago.	are packs releases	;	+ -
STM		Microelectronics	IwIP	
	Description			vailable Version
•	X-CUBE-BLE1			
From Local	BLE stack and sample applications for BlueNRG-MS m	odule		1.1.0
	X-CUBE-MEMS1			
Deta	ils			
		/	/	
Fro	om Local From Url Refre	sh Install N	Now Remove Nov	v Close

Figure 574. Installing Embedded software packages

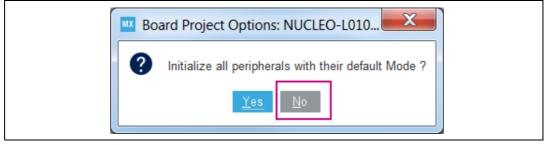
- 3. Start a new project:
 - a) Select New Project to open the new project window.
 - b) Select the **Board selector** tab.
 - c) Select Nucleo64 as board type and STM32L0 as MCU Series.
 - d) Select the NUCLEO-L053R8 from the resulting board list (see Figure 575).
 - e) Answer **No** when prompted to initialize all peripherals in their default mode (see *Figure* 576).



ew Project from a Board									
CU Selector Board Selector									
۹]	_	Feat Lar	ge Pic Docs	& Resour	🖵 Datas		G→	Start Pro
Vendor	>	☆ 「 ^N	UCLEO-L010RB						
Type Check/Uncheck All	~			STMicroelect Examples	ronics NUC	LEO-L01	ORB Board S	Support	and
Discovery			STM32 LO		Unit Pr	ice (US\$) : 0.()		
Evaluation Board					Mounte	ed device: <u>STN</u>	132L010RBTx		
Nucleo32									
Vucleo64									
MCU Series	~	Boards	List: 3 items						×
Check/Uncheck All		*	Overview	Part No 🌻 Typ	e Marketi Unit Pi	ice Mounted	. MCU SerCustom	For Mem	ory ROM
STM32F0									
STM32F1		\$		NUCLEO-L01Nuc	0.0		STM32L00	0	0
STM32F3			and the state						
STM32F4									
STM32G0		\$		NUCLEO-L0 Nuc	Active 13.0	STM321	STM32L00	0	0
STM32L0		м		NOOLLO-LO Nut		or moze	01m32E00	0	
V SIMJZLU									

Figure 575. Starting a new project - selecting the NUCLEO-L053R8 board

Figure 576. 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 577*)

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	\vee		
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	\sim	
BlueNRG-MS / Utils	\odot		✓		
✓ Device BLE1_Applications	\odot	6.1.0			
Application	\odot		SensorDemoBLESensor	\sim	
STMicroelectronics.X-CUBE-BLE2		3.2.0 ~			

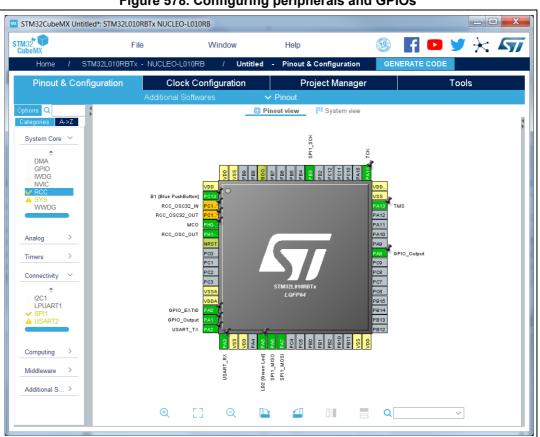
Figure 577. Selecting X-Cube-BLE1 components

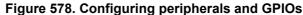
- 5. Enable peripherals and GPIOs from the **Pinout** tab (see *Figure 578*):
 - a) Configure USART2 in Asynchronous mode.
 - b) Configure SPI1 in Full-duplex master mode.
 - c) Left-click the following pins and configure them for the required GPIO settings:
 PA0: GPIO_EXTI0
 - PA1: GPIO_Output

PA8: GPIO_Output

d) Enable Debug Serial Wire under SYS peripheral.







- 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 579*).
 - 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



TM32 CubeMX	File	• Wind	low	Help	G	آ (\star \backsim
CubeMX	T He	vviik	1010	Пер	6	2 61		~ •
Home /	STM32L010RBTx ·	NUCLEO-L010RB	Untitled -	Pinout & Configurat	ion	GENERATE	CODE	
Pinout & C	onfiguration	Clock Configu	ration	Project Mana	ager		Тоо	ls
				Pinout				
Options Q			NVIC	Mode and Configuration				
Categories A->Z				Configuration				
System C Y	O NVIC O C	ode generation						
						Sort by P	remption Priority	y and Sub Priorit
DMA	Search Search (CrtI+F)			00	Show onl	y enabled interru	upts
GPIO							,	
= NVIC	Non maskable Inter		/IC Interrupt Table			Enabl V	ed Preem 0	ption Priority
V RCC	Hard fault interrupt	·				\checkmark	0	
WWDG	System service call Pendable request for					✓ ✓	0	
	Time base: System					V	0	
	Flash and EEPROM						0	
Analog >	RCC global interrup						0	
	EXTI line 0 and line EXTI line 4 to 15 inte						0	
Timers >	SPI1 global interrup					H	0	
		rrupt / USART2 wake-up interru	ot through EXTI line 2	26		Ē	0	
Connectivity~								
⊕ I2C1	•							
I2C1 LPUART1								
✓ SPI1								
USART2								
A USARIZ						ority 0 V		

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 580*).
- c) Leave the Parameter Settings Tab unchanged.
- d) Go the Platform settings tab, configure the connection with the hardware resources as indicated in *Figure 580* and *Table 26*.

Name	IPs or components	Found solutions
BUS IO driver	SPI in Full-duplex master mode	SPI1
EXTI Line	GPIO:EXTI	PA0
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



	9-			•• ===:		
STM32CubeMX Untitled*: STM3	2L010RBTx NUCLEO-L010RE	3				_ _ X
STM32	File	Window H	elp	(1)	F 🖸 🔰	* 🖌 🌆
Home / STM32L010F	RBTx - NUCLEO-L010RB	/ Untitled - Pir	nout & Configuration	G	ENERATE CODE	
Pinout & Configurati	ion Clo	ck Configuration	Projec	t Manager	Tools	;
Options Q	~	STMicroelectronics.X-	CUBE-BLE1.1.1.0 Mode and (Configuration	📋 Pinout view	System view
Categories A->Z			Mode		<u>}</u>	
System Core	>	eless BlueNRG-MS			Middlewares	
Analog	> Wir	eless Application				
			Configuration			
Timers	> Depart (Configuration	Conliguration		Additional Softwa	are
Connectivity		neter Settings 🥝 User Con	stants 🛛 📀 Platform Setting			
	Platform				X-CUBE-BLE1 😔	
Computing		NTERFACE				
Middleware	> Name	IPs or Components	Found Solutions	BSP API	Timers	Connectivity
	Exti Line	GPIO:EXTI	V PA0	✓ HAL_EXTI_DRIVER		connectivity
Additional Software	BUS IO d	river SPI:Full-Duplex Master	V SPI1	V BSP_BUS_DRIVER		SPI1 🔗
# STMicroelectronics X-CUBE-BL	E1 1 1 0 CS Line	GPIO:Output	V PA1	✓ Unknown		
	Reset Lin	e GPIO:Output	V PA8	✓ Unknown		USART2 🕑
	BSP					
	Name	IPs or Components	Found Solutions	BSP API		
	BSP BUT	TON GPIO:EXTI	PC13 [B1 [Blue PushBut	BSP_COMMON_DRIVER		
	BSP USA	RT USART:Asynchro ~	USART2	- BSP_COMMON_DRIVER		
	BSP LED	GPIO:Output ~	PA5 [LD2 [Green Led]]	V BSP_COMMON_DRIVER		

Figure 580. Enabling X-Cube-BLE1

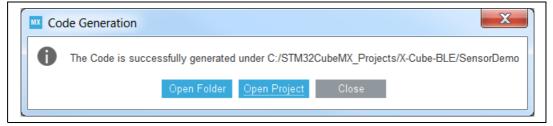
- 8. Generate the SensorDemo project:
 - a) Click GENERATE CODE to generate the code. The **Project Settings** window opens if the project has not yet been saved.
 - b) Click **GENERATE CODE** to generate the code once the project settings have been properly configured (see *Figure 581*). When the generation 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 582*).



Pinout & Cor	nfiguration	Clock Configurati		oject Manager enerate Report	Tools
Project	Project Settings Project Name SensorDemo Project Location C:\STM32CubeM>	_Projects\X-Cube-BLE		Browse	
Code Generator			✓ □ Do not generate the emo\	e main()	
	Toolchain / IDE TrueSTUDIO		✓ Generate Under R	loot	
Code Generator	Copy all used Copy only the	vare Library Package libraries into the project folder necessary library files / library files as reference in the t	oolchain project configuratio	on file	
	☐ Backup previo ✓ Keep User Co	heral initialization as a pair of '.c./ usly generated files when re-gene de when re-generating sly generated files when not re-ge	rating		
	Q Search (CrtI+F)		0		
dvanced Settings	RCC STMicroelectronics GPIO	S.X-CUBE-BLE1.1.1.0		HAL HAL HAL	
	2 3	Function Name MX_GPIO_Init C SystemClock_Config F MX_X_CUBE_BLE1_Init S	IP Instance Name SPIO RCC STMicroelectronics X-CUBE-B TMicroelectronics X-CUBE-B	LE1.1.1.0	Call Visibility (Stati

Figure 581. Configuring the SensorDemo project

Figure 582. 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.

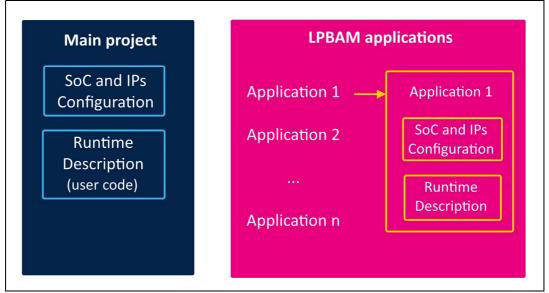


Figure 583. LPBAM project

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 584 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.



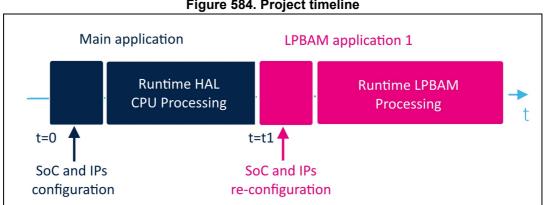
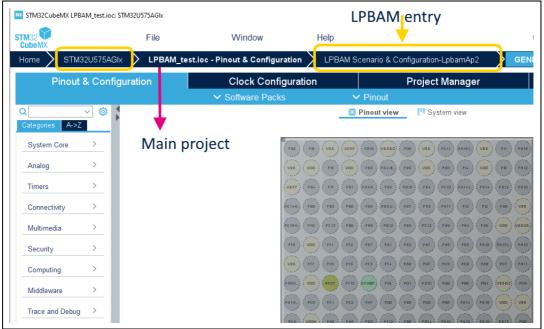


Figure 584. Project timeline

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 585*). 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.



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.

STM32CubeMX Untitled: STM32U575AG STM32 CubeMX Home STM32U575AGIx	Main project CLICK TC File Window Untitled - Pinout & Configuration	w Help		- P NERATE CODE	
	LPBAM S	Scenario & Configuration			
LPBAM Management ✓ ✓ IPBAM Manager └ ♣ Add Application		e create or activate a LPBAM appli	ication first		

Figure 586. 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".



UM1718 Rev 46

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.

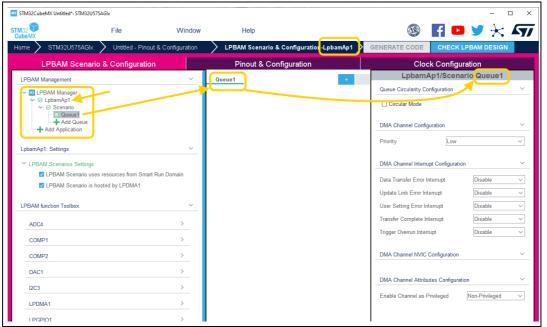


Figure 587. Adding an application

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.



				j		
ibeMX Untitled*: STM32U57	75AGIx					- 0
	File	Window	Help	LPBAM Applica	ation 🔞	F 🖸 🏏 🔆 🗸
STM32U575AGIX	VINTITIED - Pinout	& Configuration $>$	LPBAM Scenari	o & Configuration -LpbamAp1	GENERATE CODE	CHECK LPBAM DESIGN
LPBAM Scenario	o & Configuration		Pinout & C	configuration 🗡 🖂	Cloc	k Configuration
		,	✓ Pinout			
s A->Z			-	Pinout view III System view		
Core >		(PE) (VD)	P5 VD VCA)	PGIS (VODO) (PG) (VOD (PC1) (PAIS 16 (21.00) (PG) (VG) (PG) (PG) (PG)		
tivity >		(VEAT) PC140	PES (PE) (PE)		0 PHIS PHIS PHIS PIC PHIS PHIS PIC PHIS VD	

Figure 588. 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 589*, this panel is divided in three parts.

KX STM32CubeMX Untitled*: STM32U5	75AGIx		/	- C	×
STM32 CubeMX	File	Window	Help	🐵 F 🖻 🎽 🔆	57
Home > STM32U575AGIx	ntitled - Pinout & C	onfiguration	LPBAM Scenario & Configuration-LpbamAp1	GENERATE CODE CHECK LPBAM DESIGN	
LPBAM Scenari	o & Configuration		Pinout & Configuration	Clock Configuration	
LPBAM Management		~	Queue1 +	LpbamAp1/Scenario/Queue1	
✓ MX LPBAM Manager				Queue Circularity Configuration	~
✓ O LpbamAp1 ✓ O Scenario O Queue1			COMP1:Start_1	Circular Mode	
Add Queue			COMP1:OutputLevel 2	DMA Channel Configuration	~
			CONT 1. Output. Ever_2	Priority Low	\sim
LpbamAp1: Settings					
✓ LPBAM Scenarios Settings				DMA Channel Interrupt Configuration	~
LPBAM Scenario use LPBAM Scenario is h	es resources from Smart Run I	Domain		Data Transfer Error Interrupt Disable	\sim
CPDAW Scenario is n	losted by LPDMAT			Update Link Error Interret	\sim
LPBAM function Toolbox		~		User Setting Error Inter upt 🥊 Disable	\sim
ADC4		>		Transfer Complete Interrupt Disable	\sim
	\square	<u> </u>		Trigger Overrun Interrupt Disable	\sim
COMP1		~			
Start		+		DMA Channel NVIC Configuration	~
OutputLevel		+			
COMP2		、 、		DMA Channel Attributes Configuration	~
COMP2				Enable Channel as Privileged Non-Privileged	\sim
DAC4		>			



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.

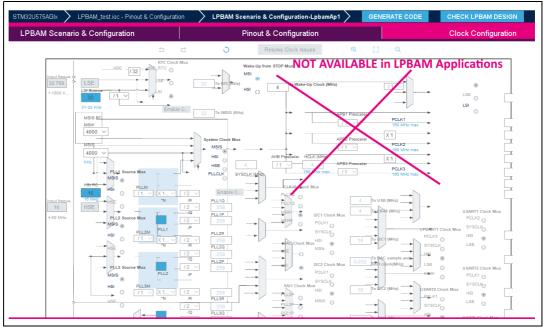


Figure 590. Clock tree configuration

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.



	Configuration		LPBAM Scenario & Configuration
Q ✓ € System Core ✓ OME 0 DMA 0 DMA 0 DMA 0 NDOG NOC RCC × VWVDG NOG Analog > Timers ✓ RCC * TIMI TIMI TIMI TIMI TIMI TIMI SPH USART1 USART2 Multimedia Computing CRC © Middeware and Software P > Trace and Debug >	Main project context	Q ✓ Ø System Care ✓ GPIO LPDMA1 NVC RCC Analog ✓ * Accla COMP1 COMP1 DMM1 VREFBUF Timers ✓ * LPTMA1 LPTMA1 LPTMA1 LPTMA1 LPTMA1 LPTMA1 VREFBUF Competing ✓ * LPTMA1 LPTMA4 LPTMA4 LPTMA5 ✓ * LPTMA5 LPTMA5 ✓ * LPTMA5 LPTMA4 ✓ * LPTMA5 LPTMA5 ✓ * ACF1 Power and Thermal ✓ * * PWR	LPBAM context

Figure 591. Available IPs

LPBAM Sce	nario & Configuration Pinout & Configuration	
	✓ Pinout	
Q ~ 1	LPTIM1 Mode and Configuration	
Categories A->Z	Mode	
	Mode Counts internal clock events	\sim
Analog ~	External Trigger	
\$	✓ Channel_1_Active	
ADC4 & COMP1	CH1 IO usage no IO used : only internal input/output connections	\sim
COMP2	Channel_2_Active	
DAC1 OPAMP1	CH2 IO usage Input	\sim
OPAMP2	☑ LPBAM Advanced Settings	
VREFBUF		
	Configuration	
Timers ~	Reset Configuration	
\$	Parameter Settings 📀 Advanced Settings for LPBAM 📀 User Constants 🛇 NVIC Settings	
LPTIM1	Configure the below parameters :	
LPTIM5 LPTIM4	Q Search (Ctrl+F) ③ ③	0
RTC	P IP Internal Interrupt	
TAMP	Capture/compare 1 over-capture interrupt Disable	
	Capture/compare 2 over-capture interrupt Disable	
Connectivity 🗸		

Figure 592. 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 LPBAM 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 Sce	nario & Configuration	Pinout & Configuration ✓ Pinout
Q ~	٢	PWR Mod	e and Configuration
Categories A->Z			Mode
System Core	>	 WakeUp from Standby configuration Debug Pins 	
Analog	>	Monitoring Low Power	
Timers	>	✓ Dead Battery Signals disabled	
Connectivity	>	 Power saving mode Security/Privilege attributes 	
Power and Thermal	~		

Figure 593. 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 594*).

To remove nodes from the diagram, click the cross on the node right-end-upper corner.

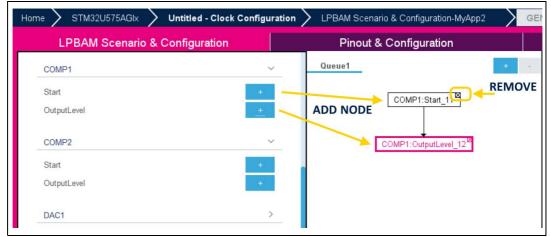


Figure 594. Adding nodes to a queue



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 595*).
- 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 LPBAM DES	IGN	
Pinout & Configuration	Clock Configuration			
Queue1 Queue2 +	MyAj	pp2/Scenario	e2	
	Queue Circularity	Configuration	~	
Conf ADC4:Conversion Config_13	Circular Mode	9		
	DMA Channel Co	onfiguration	~	
Conf Data ADC4:Conversion_data_14	Priority	Low	~	
Conf ADC4:Analog_W;tchdog_15	DMA Channel Int	errupt Configuration	~	
	Data Transfer Err	or Interrupt Disable	~	
	Update Link Erro	r Interrupt Disable	~	

Figure 595. 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 596*: 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.



i iguio ovoi quouo i		,		
on 🔰 LPBAM Scenario & Configuratio	on-MyAp	op2	GENERATE CODE	CHECK LPBAM
Pinout & Configuration			Clock Cor	nfiguration
Queue1 Queue2 Queue3	+		MyApp2/Sc	enario/Queue3
			Queue Circularity Configura	ation
Conf ADC4:Conversion_Config_16 [™]			Circular Mode	
Conf ADC4:Analog_Watchdog_17 [⊠]			DMA Channel Configuration	1
			Priority	_0W
Conf ADC4:Conversion_data_18			DMA Channel Interrupt Con	figuration
			Data Transfer Error Interrup	t Disable
			Hadata Link Error Interrupt	Disable

Figure 596. 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 597*).

MyApp2/Scenario/Queue	3	LpbamAp2/Scenario/Queue3	
		Queue Circularity Configuration	~
DMA Channel Configuration	~	Circular Mode	
Priority	~	DMA Channel Configuration	~
DMA Channel Interrupt Configuration	~	Priority	~
Data Transfer Error Interrupt Disable	~	DMA Channel Interrupt Configuration	~
Update Link Error Interrupt Disable	~	Data Transfer Error Interrupt Enable	~
User Setting Error Interrupt Disable	~	Update Link Error Interrupt Disable	~
Transfer Complete Interrupt Disable	~		~
Trigger Overrun Interrupt Disable	~		\sim
DMA Channel NVIC Configuration	~	DMA Channel NVIC Configuration	~
		Preemption Priority 0	∽
DMA Channel Attributes Configuration	~	Sub Priority 0	☑
Enable Channel as Privileged Non-Privileg	ed 🗸	DMA Channel Attributes Configuration	~
		Enable Channel as Privileged Non-Privileged	~

Figure 597. 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 597*). 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 598*).
- 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 :	<u>0</u>
	(+) Configure the ADC peripheral for conversion.	
Conv	(+) Starts the ADC conversion.	
Analo	 (+) Configure and starts the ADC conversion. (+) Configure and starts only the watchdog input signal monitoring v 	without transfor to SDAM
	Related documentation:	
	- LPBAM Utility Getting Started	
COM	- LPBAM Utility for ADC : Description, Usage and Constraints	
		•
COM	P2 >	

Figure 598. 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 599*).

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	ıt & Config	uration		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 599. 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 599*, 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 600*)



Pinout & Configuration				Clock Configuration			
Queue1	Queue1 Queue2 Queue3 +			MyApp2/Scenario/Queue1/COMP1:Start_11			
				Enter the Function	Name Start_11		
	COMP	l:Start_11 [™]		Start			
				Input Plus	PC5		
COMP1:OutputLevel_12 [⊠]				Input Minus	1/4 Internal Vref		
				Output Polarity	arity COMP Output on GPIO is not inverted		
				Trigger Configura	tion		
				The Function execution is not conditionned by a Trigger			

Figure 600. 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 601*), 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.



rigare cert El Bran nec	<u>~~</u>
Queue1 Queue2 + -	LpbamAp1/Scenario/Queue1/COMP1:Start_11
Gubucz Gubucz	Enter the Function Name Start_11
COMP1:Start_11	Start ~
	Input Plus PC5
COMP1:OutputLevel_12	Input Minus 1/4 Internal Vref ~
COMPT.OutputEver_12	Output Polarity COMP Output on GPIO is not inverted ~
	Trigger Configuration
	The Function execution is trigged on the Rising Edge of the Hardware Signal <
	Trigger Hardware Signal is EXTI Line 0
	EXTI Line 0
	EXII Line 1 Select signal
	EXTI Line 3
	EXTI Line 4 ATAMP TRG1
	TAMP TRG2
	TAMP TRG3

Figure 601. 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 602*).

Figure 602. 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 V		
	Input Plus PC5 ~		
COMP1:OutputLevel_12	Input Minus 1/4 Internal Vref 🗸		
	Output Polarity COMP Output on GPIO is not inverted ~		
	Trigger Configuration ~		
	The Function execution is $${\rm trigged}$$ on the Rising Edge of the Hardware Signal \sim		
	Trigger Hardware Signal is EXTI Line 0 🗸		
	TAMP TRG2		
	ITAMP TRG3		
	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 603*).

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.



Pinout & Configuration	Clock Configuration		
Queue1 + -	LpbamAp1/Scenario/Queue1/COMP1:OutputLevel_13		
	Enter the Function Name OutputLevel_13		
COMP1:OutputLevel_13	Data Buffer Name DataBufferName		
	Data Buffer Offset DataBufferOffset		
ADC4: Conversion_Config_15	Number of Data myDataSize		
	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 Disa	abled	
	Destination Address Increment After Transfer Enab	bled	
	Src Data Width Word	rd .	
	Dest Data Width Word	'd	
	Transfer Event Generation TC a	and HT generated on the last lir	

Figure 603. 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 603*).

Figure 604	Reconfiguring	DMA for data	transfer when	destination is	memory
------------	---------------	--------------	---------------	----------------	--------

Reconfigure DMA for Data Transfer / Default Setting is functional				
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 last	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 605*).

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).

STM32CubeMX Untitled*: STM32U575AGkQ – – – × Click to check					
TM32 File Window		^{ck} 🍳 🗗 🗖 🎽 🗘 🔆 🖅			
Home > STM32U575AGIxQ > Untitled - Project Manager >	LPBAM Scenario & Configuration-MyApp2 >	GENERATE COLL CHECK LPBAM DESIGN			
LPBAM Scenario & Configuration	Pinout & Configuration	Clock Configuration			
LPBAM Management V	Queue1 < > +	MyApp2/Scenario/Queue1			
 ✓ IPBAM Manager > LpbarnAp1 > Ø MyAp2 > Scenario > O Gueuent 	12C3:Master_Receive_Config_3	Queue Circularity Configuration V Gircular Mode			
Add Queue Add Application MyAc 2. Settings	↓ I2C3:Master_Receive_7 [®]	DMA Channel Configuration Priority Low			
		DMA Channel Interrupt Configuration			
Rubime contexts: Cortex-M33 secure Cortex-M33 non secure Cortex-M33 non secure		Data Transfer Error Interrupt Disable V Update Link Error Interrupt Disable V			
LPBAM Scenario uses resources from Smart Run Domain	Descrite	User Setting Error Interrupt Disable 🗸			
C LPBAM Scenario is hosted by LPDMA1	Results	Transfer Complete Interrupt Disable ~			
MCUS SY Edion Output LFBAM Output Log Show Attribute Warmin Messages LERAM Design Check log Lybe Application Scenario Scenario Output Configuration : Configuration of Queuel is the default configuration. Queue can be configured through this application "LFBAM scenario & NyApp2 projection" when focusing on the Queue in the central panel NyApp2 projection Scenario scenario of chipmeral Configuration: ICG3 is used through Queuel/'I2C3:Master_Receive_7' and is not configured. IF must be configured in this Application 'Prepheral Configuration' ICG3 is used through Queuel/'I2C3:Master_Receive_Config_3' and is not configured. IF must be configured in this Application 'Prepheral Configuration: ICG3 is used through Queuel/'I2C3:Master_Receive_Config_3' and is not configured. IF must be configured in this Application 'finout & Configuration' panel Prepheral Configuration: All Trigger Hardware Signals used in this scenario should be configured in this application 'Finout & Configuration' panel B Scenario Design: function I2C3:MASTERECEIVE_CONFIG is used Through Queuel/'I2C3:Master_Receive_Config_3'. For memory optimization purpose, when a reconfiguration : Configuration of Queuel is the default configuration. Queue can be configured through this application "LEBAM scenario & Configuration" when focusing on the Queue in the central panel					

Figure 605. Design check



18.5 Generating a project with LPBAM applications

Click Generate Code from the main project view. As exemplified in *Figure 605*, 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.

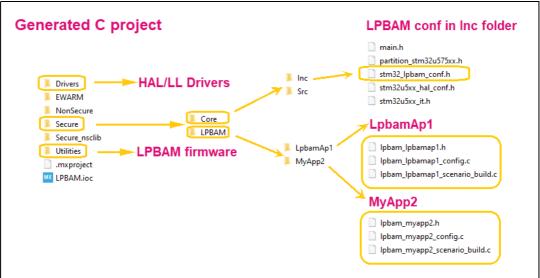


Figure 606. STM32CubeMX project generated with LPBAM applications

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 $^{\textcircled{R}}$ 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 607*) and the clock tree (*Figure 608*).

STM32CubeMX LPBAM view

In STM32CubeMX LPBAM Application configuration context, the peripherals and the clock tree do not come with dedicated security settings (see *Figure 609* and *Figure 610*). The choice of context, secure or nonsecure, is done at LPBAM application level (*Figure 611*).

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 612*)



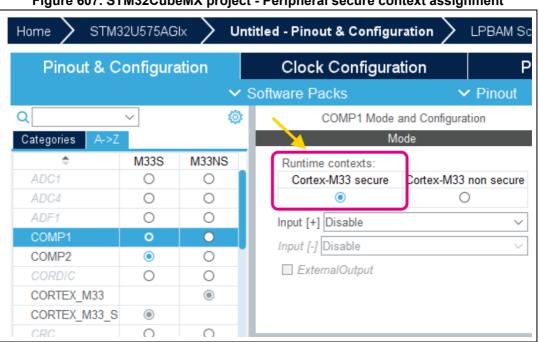
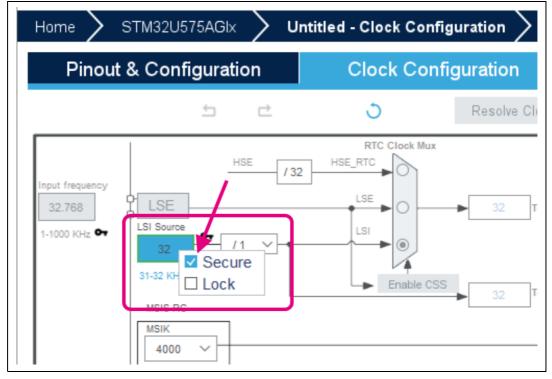


Figure 607. STM32CubeMX project - Peripheral secure context assignment

Figure 608. STM32CubeMX project - Clock source secure context assignment

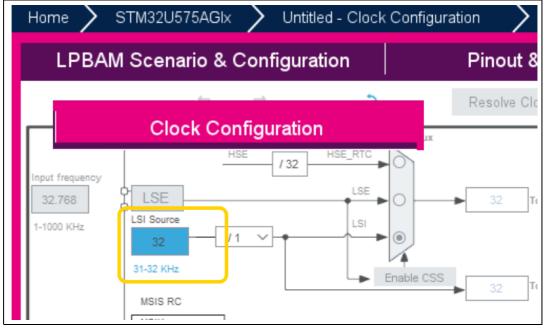




· · · gai • • • • • · · .	
Home 🔰 STM32U575AG	Ix $>$ Untitled - Clock Configuration $>$ LPBAM Scenario
LPBAM Scenario &	Configuration Pinout & Configuration
	✓ Pinout
Q 🔅	COMP1 Mode and Configuration
Categories A->Z	Input [+] Disable
COMP1 COMP2	Input [-] Disable ~
DAC1 GPIO	
I2C3 LPDMA1 LPTIM1	
LPTIM1 LPTIM3 LPTIM4	

Figure 609. LPBAM project - Peripheral no context assignment

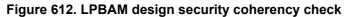


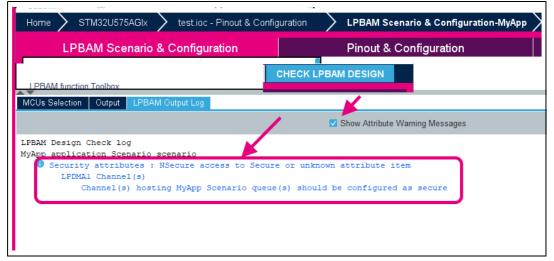




Home 🔪 STM32U575AGIx 🔪	 Untitled - Clock Configu 	ration	> LPBAM Scenario &
LPBAM Scenario & Co	nfiguration	Pinou	ut & Configuration
LPBAM Management		~	Queue1
 ✓ ILPBAM Manager ✓ MyApp ✓ Scenario ✓ Queue1 ✓ Add Queue ✓ Add Application 		~	
Runtime contexts:		<u>)</u>	
Cortex-M33 secure	Cortex-M33 non secure		
✓ LPBAM Scenarios Settings			
LPBAM Scenario uses re	esources from Smart Run Don	nain	
LPBAM Scenario is host	ed by LPDMA1		

Figure 611. LPBAM application - Secure context assignment







Appendix A STM32CubeMX pin assignment rules

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 613*).
- I2C1 peripheral mode is set to SMBus-Alert mode.



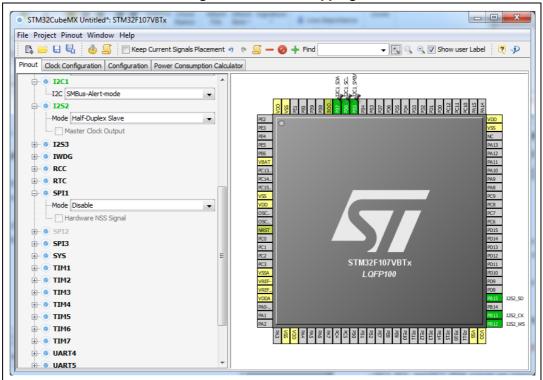


Figure 613. Block mapping

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 614*).



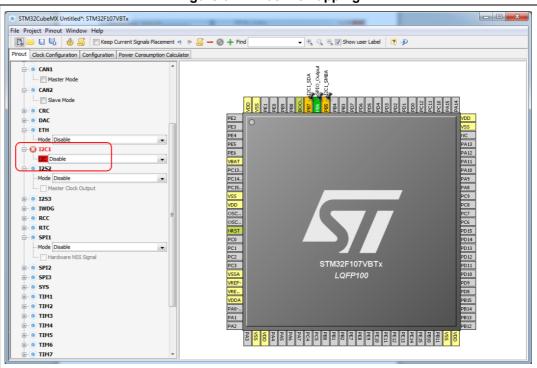


Figure 614. Block remapping

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 615* and *Figure 616*).



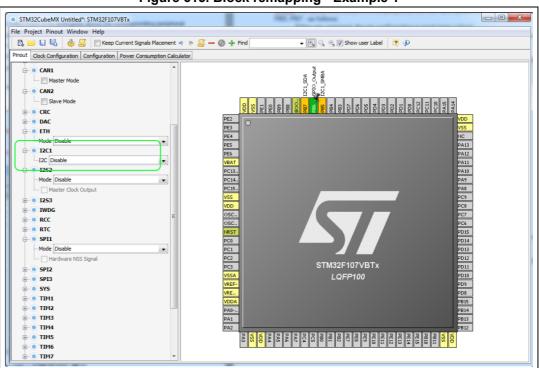
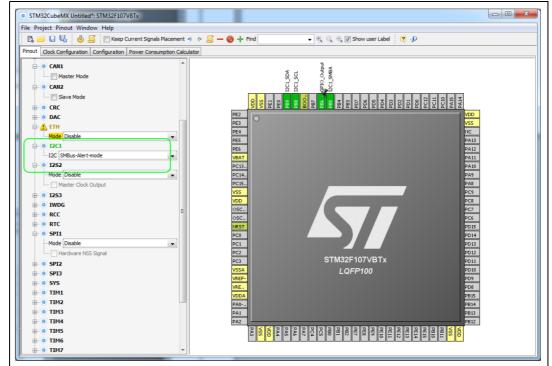


Figure 615. Block remapping - Example 1

Figure 616. Block remapping - Example 2



UM1718 Rev 46



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 617*).

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 618*).

(by pressing CTRL and clicking PB4 to show PA6 alternate function in blue, then drag and drop the signal to pin PA6)

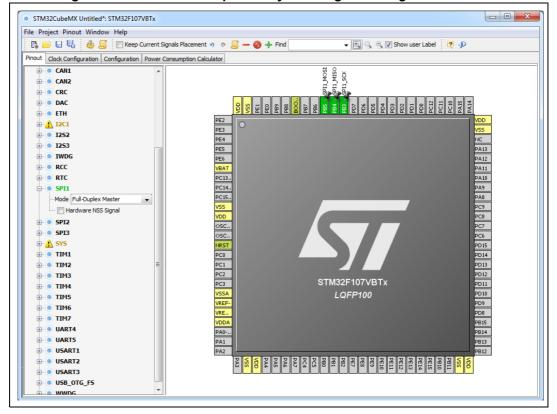


Figure 617. Block inter-dependency - SPI signals assigned to PB3/4/5



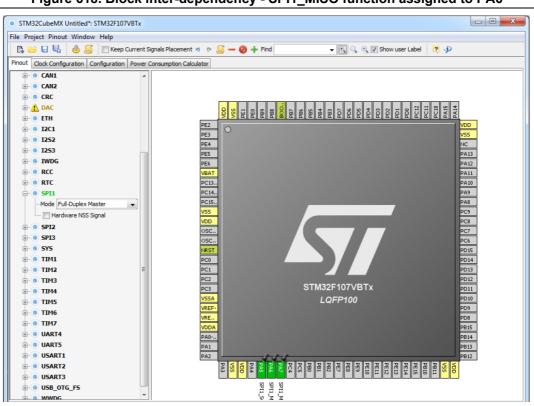


Figure 618. Block inter-dependency - SPI1_MISO function assigned to PA6



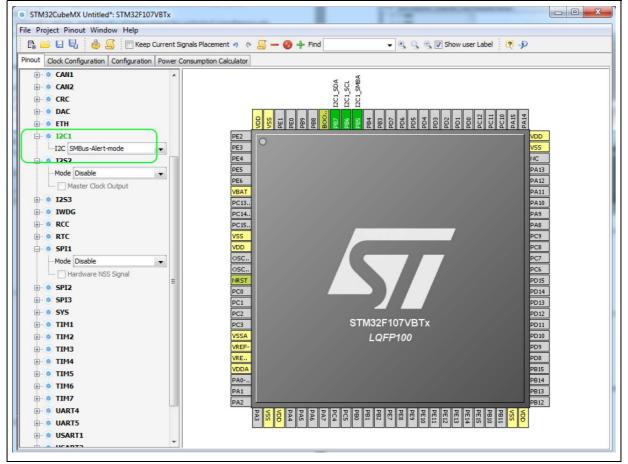
A.3 One block = one peripheral mode

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 619*).

Figure 619. 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.

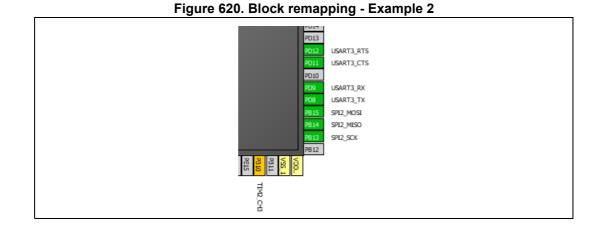
When setting a peripheral mode, if at least one pin in the default block is already used, STM32CubeMX tries to find an alternate block. If none can be found, it either selects the functions in a different sequence, or unchecks Keep Current Signals Placement, and remaps all the blocks to find a solution.



UM1718 Rev 46

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 620*).



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 621*).

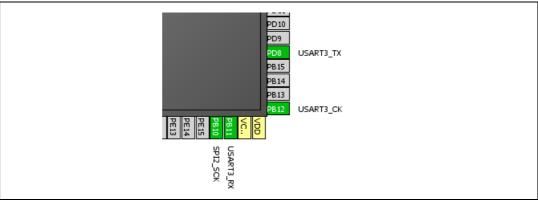


Figure 621. Function remapping example



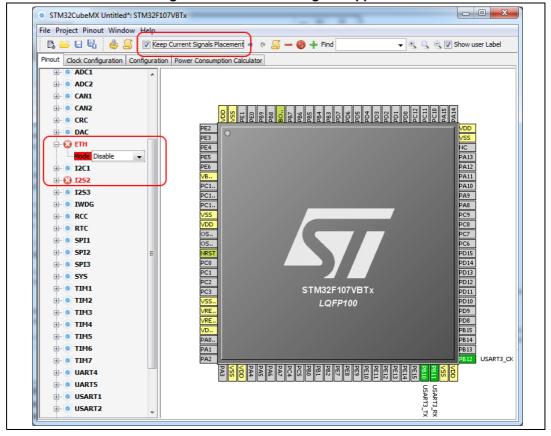
A.6 Block shifting (only for STM32F10x and when "Keep Current Signals placement" is unchecked)

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 622*).

Unchecking Keep Current Signals Placement allows STM32CubeMX shifting blocks around and freeing a block for the Ethernet MII mode. (see *Figure 623*).







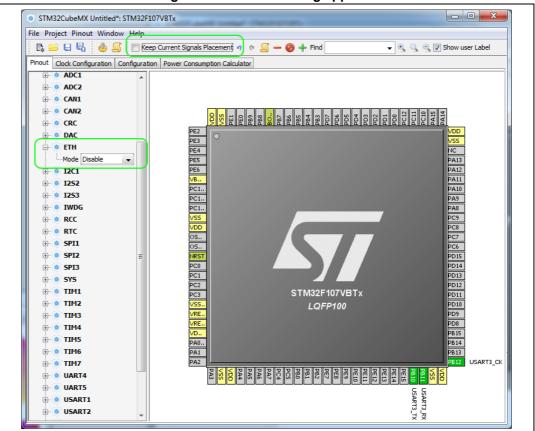


Figure 623. Block shifting applied

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.



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 624*).

Figure 624. FreeRTOS HOOK functions to be completed by user

-	e the following parameters:		
earch :	Search (Crtl+F) 🗢 🛧		
Versi	ions		-
	CMSIS-RTOS version	1.02	
	FreeRTOS version	8.2.3	
Kerne	el settings		
Hook	c function related definitions		
	USE_IDLE_HOOK	Disabled	
	USE_TICK_HOOK	Disabled	:
	USE_MALLOC_FAILED_HOOK	Disabled	
	CHECK_FOR_STACK_OVERFLOW	Disabled	
Runt	time and task stats gathering related definitions		
	USE_TRACE_FACILITY	Enabled	
	GENERATE_RUN_TIME_STATS	Disabled	
Co-ro	outine related definitions		
	USE_CO_ROUTINES	Disabled	
	MAX_CO_ROUTINE_PRIORITIES	2	
Softv	ware 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 626).

DHCP must be disabled, to configure a static IP address.

Infigure the below parameters : earch : Search (CrtI+F)	Statistics	Checksum	🗸 Debug	User Constants
earch : Search (CrtI+F)	Seneral Se	ettings	Key Options	Perf/Checks
I.WIP Version I.4.1 I.WIP Version (Version of LwIP sup I.4.1 DHCP Option Enabled I.WIP_DHCP (DHCP Module) Enabled RTOS Settings WITH_RTOS (Use FREERTOS ** WITH_RTOS (Use FREERTOS ** Disabled Protocols Options LWIP_ICMP (ICMP Module Activati LWIP_ICMP (IGMP Module Activati Enabled LWIP_DNS (DNS Module) Disabled LWIP_DNS (DNS Module) Enabled	Configure the below p	arameters :		
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_ICMP (ICMP Module Activati Enabled LWIP_ICMP (ICMP Module) Disabled LWIP_DNS (DNS Module) Disabled LWIP_UDP (UDP Module) Enabled	Search : Search (Crt	(+F)	-	
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		1	
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	(Version of LwIP sup	1.4.1	
RTOS Settings Disabled 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 FREERTOS ** .	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 Activat	i Enabled	
LWIP_UDP (UDP Module) Enabled	LWIP_IGMP	(IGMP Module)	Disabled	
	LWIP_DNS (DNS Module)	Disabled	
	LWIP_UDP (UDP Module)	Enabled	
MEMP NUM LUP PCR Number of 4	MEMP NUM	LIDP_PCB (Number o	f 4	

Figure 625. LwIP 1.4.1 configuration



Configure the be	Settings			Checksum	Debug	SNMP	Constants
onfigure the b	- I	Key Opti	ons	N INA	MILIPD	SINMP	SNIP
	elow parametel	rs :					
Search : Searc	:h (Crtl+F)		•				
LwIP Versio	n						
LwIP	Version (Versio	n of LwIP suppo	rted b	1.5.0_RC0_	20160211		
DHCP Optio	n						
LWIP.	DHCP (DHCP N	Module)		Enabled			
RTOS Settir	ngs						
WITH	_RTOS (Use FR	EERTOS ** Cu	beMX	Disabled			
Protocols O	ptions						E
LWIP_	_ICMP (ICMP M	odule Activatio	ר)	Enabled			
	_IGMP (IGMP M	-		Disabled			
	DNS (DNS Mod	-		Disabled			
-	UDP (UDP Mod			Enabled			
		B (Number of L	IDP Co				
LWIP_	TCP (TCP Mod	ule)		Enabled			

Figure 626. LwIP 1.5 configuration

STM32CubeMX generated C code will report 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.



Search : Search (Crtl+F)	Show Advanced Parameters	
Version		*
LIBJPEG version	8d	
MW configuration		
Data Stream management type	FatFS	
FREERTOS	Enabled	=
 General Settings 		
Use FREERTOS Memory Allocator	r Disabled	
JPEG basic settings		
BITS_IN_JSAMPLE	8 bits	
MAX_COMPONENTS	12	
JCOEF	short	
JPEG encoder and decoder common ca		
DCT_ISLOW_SUPPORTED		
DCT_IFAST_SUPPORTED		
DCT_FLOAT_SUPPORTED	Enabled	
JPEG encoder options		
C_ARITH_CODING_SUPPORTED		
C_MULTISCAN_FILES_SUPPORT.	Disabled	Ŧ
Data Stream management type		
LIBJPEG_FS_type		

Figure 627. Libjpeg configuration window

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 628: Mbed TLS without LwIP).

If LwIP stack is used, FreeRTOS must be enabled as well (see *Figure 629: Mbed TLS with LwIP and FreeRTOS*).



STM32CubeMX generates the following files, whose contents can be modified by the user through STM32CubeMX user interface (see *Figure 630: Mbed TLS configuration window*) 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 628. Mbed TLS without LwIP

MBEDTLS Configuration			X
Modules Modules	dules Configuration	< User Consta	ints
Version and modes	[®] Feature support	Alternate implement	ation
Configure the below parameters :			
Rearch (Octor)			
Search : Search (Crtl+F)	• •		
Version			
MBEDTLS version	2.4.0		
TCP/IP stack			
TCP/IP stack	None		
RNG dependency RNG IP	CW/ DNC		
Modes	SW RNG		
MBEDTLS_SSL_CLI_C	Not Defined		
MBEDTLS_SSL_CLL_C MBEDTLS_SSL_SRV_C	Not Defined		
MBEDTES_SSE_SKV_C	Not Defined		
Restore Default	Appl	ly Ok C	Cancel



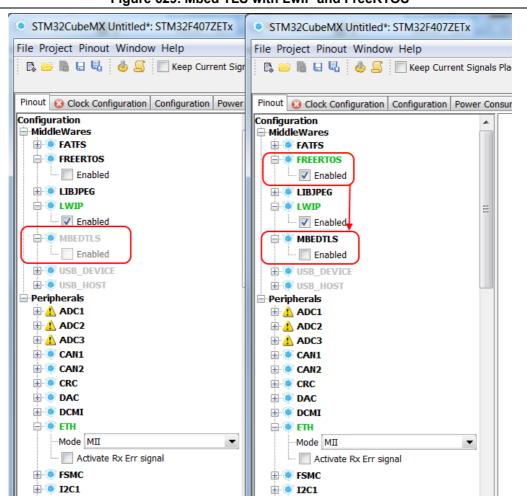


Figure 629. Mbed TLS with LwIP and FreeRTOS



	Configuration 🚽 💎 User Co	
Version and modes	ature support 🛛 🛷 Alternate impler	nentation
configure the below parameters :		
Connels (Cathar		
Search : Search (Crtl+F)	• •	
Version		
MBEDTLS version	2.4.0	
TCP/IP stack		
TCP/IP stack		
RNG dependency		
RNG IP	SW RNG	
Modes		
MBEDTLS_SSL_CLI_C	Defined	
MBEDTLS_SSL_SRV_C	Defined	

Figure 630. 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 631: Enabling the TouchSensing peripheral, Figure 632: Touch-sensing sensor selection panel* and *Figure 633: TouchSensing configuration panel*) and/or using user sections in the code itself:

- touchsensing.c/.h
- tsl_user.c/.h
- tsl_conf.h



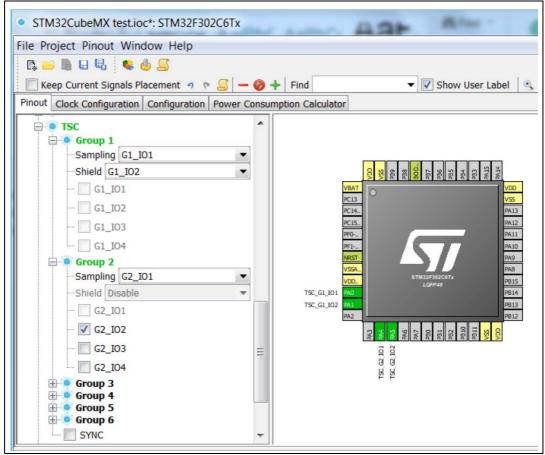


Figure 631. Enabling the TouchSensing peripheral



Sensors selection 🧹 Config par	ameters 😵 User Constants	
Configure the below parameters :		
Search : Search (Crtl+F)		
Summary		
TSLPRM_TOTAL_OBJECTS	5 1	
TSC_ACTIVE_CHANNELS =	= 1	
TSLPRM_TOTAL_CHANNEI	LS 1	
± Linear/Rotary sensors used		
TouchKey sensors		
TSLPRM_TOTAL_TOUCHK		
IO_TOUCHKEY1 TSLPRM_TOTAL_TOUCHK	G2_IO2	
TSLPRM_TOTAL_OBJECTS TSLPRM_TOTAL_OBJECTS must be : Parameter Description: Total number of sensors/objects in a Count all TouchKeys, Linear and Rota	application	

Figure 632. Touch-sensing sensor selection panel



• TOUCHSENSING Configuration		×
Sensors selection 🖋 Config parame	eters 🎻 User Constants	
Configure the below parameters :		
Search : Search (Crtl+F)	• •	
Version and modes		
TouchSensing version	2.2.0	
Optional features		
Acquisition limits		
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 Exclusion System (DXS)		
Miscellaneous parameters		
In miscenarie das parameters		
Restore Default	Apply Ok	Cancel

Figure 633. 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 BLE Disabled: Active only if RF, RTC &	HSEM are enabled & FreeRTOS is disabled
Analog >		
Timers >		
Connectivity >		
Multimedia		
Security >		
Computing >		
Middleware ~	Configuratio	n

Figure 634. 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.

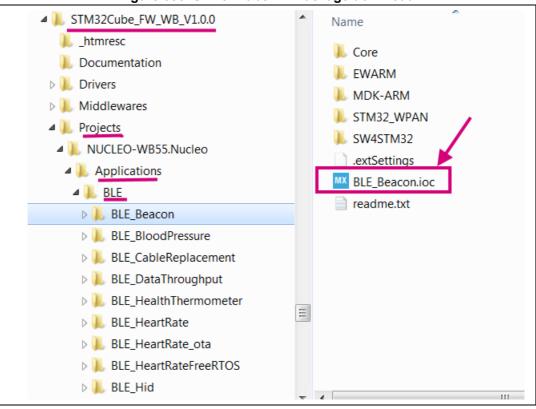


Figure 635. STM32CubeWB Package download



This package can be installed through STM32CubeMX following the standard procedure described in *Section 3.4.3: Installing STM32 MCU packages*.

		1 19		STWSZCUDEWE BLE applications folde	<u></u>
MX	Embed	ded Software Pa	ckages Ma	nager	X
		STM32Cube MC	U Package	s and embedded software packs releases	± -
	\bigcirc	Releases Informa	tion was la	t refreshed less than one hour ado	I
ST	M32Cul	oe MCU Packages			
	[Description		Installed Version	n Available Version
Þ	STN	132F7			
Þ	STN	132G0	ſ	Downloading selected software packages]
Þ	STN	132H7		Unzip File : stm32cube_fw_wb_v100.zip	1
Þ	STN	132L0		Download and Unzip selected Files	
Þ	STN	132L1			
Þ	STN	132L4		OK Cancel	
Þ	STN	132MP1			_
T	STN	132WB			
		STM32Cube MCU	Package fo	r STM32WB Series (Size : 70.8 MB)	1.0.0
De	tails				
	<mark>M32Cu</mark> ain Cha	beWB Firmware]	Package V	.0.0	
		-	32CubeWB	(STM32Cube for STM32WB Series) supporting STM32WB55xx d	levices.
				(
F	From Lo	cal From	ı Url	Refresh Install Now Remove 1	Now Close

Figure 636. STM32CubeWB 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.

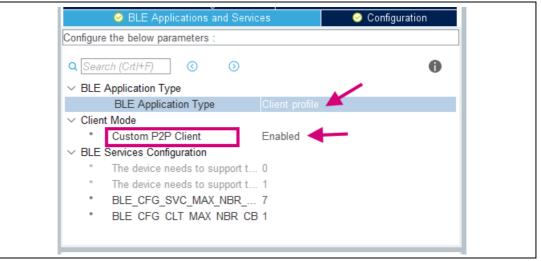
Note: The BLE Transparent mode and all Thread applications require either the USART or the LPUART peripheral to be configured as well.



	rigure our: DEE of	•		
	STM32_WPAN M	ode and Configu	ration	
	M	ode		
E	3LE			
	Confi	guration		
Dee	at Canfinumation			
Res	et Configuration			
	🥝 Parameter Settings	<u>(</u>	Oser Constants	
	BLE Applications and Servi	ces	😔 Configurat	ion
Configure	e the below parameters :			
			1	•
Q Sea	rch (CrtI+F) 🕜 🕥			U
\sim BLE	Application Type			
	BLE Application Type	Server profile		\sim
✓ Serve	er Mode			
	BT SIG Beacon	Enabled		
*	BT SIG Blood Pressure Sensor			
*	BT SIG Health Thermometer			
*	BT SIG Heart Rate Sensor	Disabled		
*	Custom P2P Server	Disabled		
	Custom Template	Disabled		
✓ BLE	Services Configuration The device needs to support	4		
	The device needs to support The device needs to support			
*	BLE_CFG_SVC_MAX_NBR			
	BLE_CFG_CLT_MAX_NBR			
		0		

Figure 637. BLE Server profile selection





520/557

UM1718 Rev 46



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.

	VPAN Mode a	nd Configuratio	n	
	Mode			
BLE				
THREAD				
	Configurati	on		
Reset Configuration				
🥺 Parameter Setting	gs	😔 U:	ser Constants	
🕑 THREAD Applicatio	ns and Service	es	😔 Configurat	ion
Configure the below parameters	:			
Q Search (CrtI+F)	\odot			0
 THREAD application type 		•		-
Thread Application	Thr	ead_Coap_Ger	neric	~
		ead_Cli_Cmd		
		ead_Coap_Ger		

Figure 639. 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 640*).
- 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 641*). 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 642*).

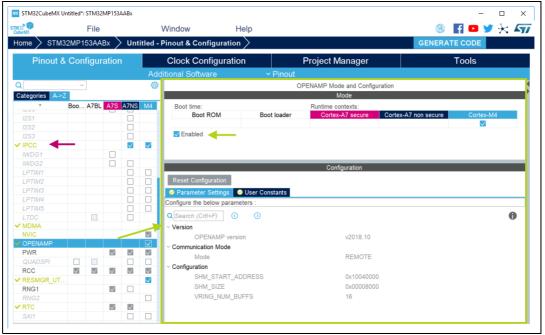


Figure 640. Enabling OpenAmp for STM32MPUs



Options Q ~	RESMGR_UTILITY Mode and Configuration
Categories A->Z	Mode
÷	Boot time: Runtime contexts:
LTDC	Boot ROM Boot loader A7S A7NS Cortex-M4
✓ MDMA	
NVIC	Peripheral assignment request
	✓ Dynamic system resources update
PWR	
QUADSPI	Configuration
RCC	
RESMGR_UTILITY	Reset Configuration
RNG1	📀 Parameter Settings 🛛 📀 Peripherals assignment 🛛 📀 User Constants
RNG2	Configure the below parameters :
✓ RTC	
SAI1	Q Search (CrtI+F) ③ ③
SAI2	✓ Version
SAI3	RESMGR_UTILITY version mp1/v1.4.0
SAI4	✓ configuration
SDMMC1	use the RPMSG/OpenAMP-based extension true
SDMMC2	
SDMMC3	
SPDIFRX	
SPI1	
SPI2	
SPI3	
SPI4	
SPI5	

Figure 641. Enabling the Resource Manager for STM32MPUs



Pinout	& Con	fiqur	atior	ı		Clock Configura	ation	Project Mar	nager	-
					Add	litional Software		Pinout	0	
Q	~]			Ô		RESMGF	R_UTILITY Mode and Config	uration	•
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 assign	ment request			
CRC1				\checkmark		✓ Dynamic system				
CRC2					\checkmark					
DAC1					\checkmark					
DCMI										
DDR				\checkmark						
DEBUG										
DFSDM1				\checkmark				Configuration		
✓ DMA						Reset Configuration				
DTS							J			
ETH1			_		_			s assignment 🛛 📀 User Con	stants	
ETZPC			\checkmark	\checkmark	\checkmark	Configure the below pa	rameters :			
FDCAN1						Q Search (CrtI+F)	\odot \bigcirc			0
FDCAN2	_	_			\checkmark	~ RESMGR_UTILITY	Common			
FMC							•	t for ADC1 in resource table	is assign	ned to Cortex-A7NS
FREERTOS GIC			\checkmark	\checkmark	\sim			t for ADC2 in resource table		ned to Cortex-A7NS
GPIO			\checkmark	$\mathbf{\nabla}$					0	led to contex-Arins
HASH1							-	t for CEC in resource table	no	11.0.1.1.17110
HASH2							5	t for CRC1 in resource table	0	ned to Cortex-A7NS
HDMI CEC							-	t for CRC2 in resource table	-	ned to Cortex-M4
HDP					-			t for CRYP1 in resource tab		
HSEM						request peri	pheral assignmen	t for CRYP2 in resource tab	ole no	
12C1						request peri	pheral assignmen	t for DAC1 in resource table	e is assign	ned to Cortex-M4
1202						request peri	pheral assignmen	t for DBGMCU in resource	table no	
12C3						request peri	oheral assignmen	t for DCMI in resource table	e no	
12C4						request peri	oheral assignmen	t for DFSDM1 in resource t	able is assign	ed to Cortex-A7NS
12C5		_				request peri	oheral assignmen	t for DLYB_QUADSPI in re	sourc no	
12C6					_		-	t for DLYB SDMMC1 in res		
10.04						i oquoor pon				

Figure 642. Resource Manager: peripheral assignment view

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 643 shows an example of STM32 microcontroller part numbering scheme.

Figure 643. STM32 microcontroller part	numbering scheme
--	------------------

Example:	STM32	F	439 V	1 1	6	xxx
Device family						
STM32 = ARM-based 32-bit microcontroller	<u>ı</u>					
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, Ethemet, 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						<u> </u>
TR = tape and reel						



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 μs 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 644*). 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								
Mide	lleware examples	ile)						
TCP/IP IwIP stack + Polar SSL Middleware level	RTOS FreeRTOS Utilities (time, string, file)							
Hardware Abstraction Layer APIs (H/	IAL examples AL) Board Support Package (BSP)	Utilities						
MCU Series (STM32F4, F1, F2, F3. Hardware	.) Evaluation boards, discovery boards, dedicated demonstration boards	MSv34720V2						

Figure 644. 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?.
			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
		3 4.3	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	3		Updated <i>Figure 323: Power Consumption Calculator default view</i> and added battery selection in <i>Section 5.3.1: Building a power consumption sequence</i> .
	-		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?.

Table 27. Document re	evision history
-----------------------	-----------------



Date	Revision	STM32CubeMX release number	Changes
			Added support of STM32CubeF0, STM32CubeF3, STM32F0 and STM32F3 Series in cover page, <i>Section 2.2: Key features</i> , <i>Section 2.3: Rules and limitations</i> ,
			Added board selection capability and pin locking capability in Section 2.2: Key features, Table 2: Home page shortcuts, Section 4.2: 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.
19-Jun-2014	4	4.4	Updated Figure 51: STM32CubeMX Main window upon MCU selection, Figure 183: Project Settings window, Figure 310: About window, Figure 140: STM32CubeMX Pinout view, Figure 120: Chip view, Figure 323: Power Consumption Calculator default view, Figure 324: Battery selection, Figure 87: Building a power consumption sequence, Figure 326: Power consumption sequence: New Step default view, Figure 333: Power Consumption Calculator view after sequence building, Figure 334: Sequence table management functions, Figure 88: PCC Edit Step window, Figure 83: Power consumption sequence: new step configured (STM32F4 example), Figure 331: ADC selected in Pinout view, Figure 332: Power Consumption Calculator configuration window: ADC enabled using import pinout, Figure 336: Description of the Results area, Figure 100: Peripheral power consumption tooltip, Figure 537: Power Consumption Calculation example, Figure 155: Sequence table and
		Figure 156: Power Consumption Calculation results. Updated Figure 142: STM32CubeMX Configuration 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 from Section 5.3.2: Configuring a step in the power sequence.
			Updated Section 8.2: Reviewing results.
			Updated appendix <i>B.3.4: FatFs</i> and <i>Appendix D: STM32</i> <i>microcontrollers power consumption parameters</i> . Added Appendix <i>D.1.3: STM32L0 series</i> and <i>D.2.3: STM32L0 series features three</i> <i>VCORE ranges</i> .

Table 27. Document revision history (continued)



Date	Revision	STM32CubeMX release number	Changes
19-Sep-2014	5	4.5	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 43: New Project window - Board selector. Updated Figure 191: 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 84: GPIO configuration window - GPIO selection; Section 4.5.12: GPIO configuration window and Figure 89: DMA MemToMem configuration. Updated introduction of Section 4.10: Clock Configuration view. Updated Section 4.10.1: Clock tree configuration functions and Section 4.10.3: Recommendations, Section 5.3: Power Consumption Calculator view, Figure 326: Power consumption calculator view after sequence building, Figure 83: Power consumption sequence: new step configured (STM32F4 example), and Figure 332: Power Consumption calculator configuration window: ADC enabled using import pinout. Added Figure 335: Power Consumption: Peripherals consumption colitip. Updated Section 5.3.4: Power sequence step parameters glossary. Updated Section 1.1: Creating a new STM32CubeMX Project and Section 1.2: Configuring the MCU pinout. Added Section 12: Tutorial 2 - Example of FatFs on an SD card using STM32429I-EVAL evaluation board and updated Section 8: Tutorial 3 - Generating GPIO initialization C code (STM32F1 Series only). Updated Section 5.3.2: Configuring a step in the power sequence.

Table 27. Document revision history (continued)



Date	Revision	STM32CubeMX release number	Changes
			Complete project generation, power consumption calculation and clock tree configuration now available on all STM32 Series.
			Updated Section 2.2: Key features and Section 2.3: Rules and limitations.
			Updated Eclipse IDEs in Section 3.1.3: Software requirements.
			Updated Figure 18: Updater Settings window, Figure 21: Embedded Software Packages Manager window and Figure 43: New Project window - Board selector, Updated Section 4.11: Project Manager view and Section 4.14: Update Manager windows.
			Updated Figure 310: About window.
			Removed Figure STM32CubeMX Configuration view - STM32F1 Series.
			Updated Table 17: STM32CubeMX Chip view - Icons and color scheme.
			Updated Section 5.14.1: Peripherals and Middleware Configuration window.
			Updated Figure 87: Adding a new DMA request and Figure 89: DMA MemToMem configuration.
			Updated Section 4.10.1: Clock tree configuration functions.
19-Jan-2015	6	4.6	Updated Figure 324: Battery selection, Figure 87: Building a power consumption sequence, Figure 88: PCC Edit Step window.
			Added Section 6.3: Custom code generation.
			Updated Figure 491: Clock tree view and Figure 496: Pinout & Configuration view.
			Updated peripheral configuration sequence and <i>Figure 498: Timer 3 configuration window</i> in <i>Section 11.6.2: Configuring the peripherals.</i> Removed Tutorial 3: Generating GPIO initialization C code (STM32F1 Series only).
			Updated Figure 502: GPIO mode configuration.
			Updated Figure 537: Power Consumption Calculation example and Figure 155: Sequence table.
			Updated Appendix <i>A.1: Block consistency</i> , <i>A.2: Block inter- dependency</i> and <i>A.3: One block</i> = one peripheral mode.
			Appendix <i>A.4: Block remapping</i> (STM32F10x only): updated Section : <i>Example</i> .
			Appendix A.6: Block shifting (only for STM32F10x and when "Keep Current Signals placement" is unchecked): updated Section : Example
			Updated Appendix <i>A.8: Mapping a function individually</i> .
			Updated Appendix <i>B.3.1: Overview</i> .
			Updated Appendix D.1.3: STM32L0 series.

Table 27. Document revision history (continued)



		STM32CubeMX	ument revision history (continued)	
Date	Revision	release number	Changes	
19-Mar-2015	7	4.7	 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 43: New Project window - Board selector. Updated IDE list in Section 4.11: Project Manager view and modified Figure 183: Project Settings window. Updated Section 4.10.1: Clock tree configuration functions. Updated Figure 179: STM32F469NIHx clock tree configuration view. Section 5.3: Power Consumption Calculator view: added transition checker option. Updated Figure 323: Power Consumption Calculator default view, Figure 324: Battery selection and Figure 87: Building a power consumption sequence. Added Figure 327: Enabling the transition checker option on an already configured sequence - All transitions valid, Figure 328: Enabling the transition checker option on an already configured 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 336: Description of the Results area. Updated Figure 537: Power Consumption Calculation example, Figure 155: Sequence table, Figure 156: Power Consumption Calculation results and Figure 158: Power consumption results - IP consumption chart. Updated Appendix B.3.1: Overview and B.3.5: FreeRTOS. 	
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 326: Power consumption sequence: New Step default view, Figure 88: PCC Edit Step window, Figure 83: Power consumption sequence: new step configured (STM32F4 example), Figure 332: 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 504: 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 STM32429I-EVAL.
16-Oct-2015	11	4.11	 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 537: 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 196: Automatic project import; modified Manual project import step and updated Figure 197: Manual project import and Figure 198: 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 537: Power Consumption Calculation example to Figure 546: Step 10 optimization. Updated Figure 547: Power sequence results after optimizations.



pdated Section 2.2: Key features: Information related to .ioc files. Clock tree configuration Automatic updates of STM32CubeMX and STM32Cube. pdated limitation related to STM32CubeMX C code generation in ection 2.3: Rules and limitations. dded Linux in Section 3.1.1: Supported operating systems and rchitectures. Updated Java Run Time Environment release number i Section 3.1.3: Software requirements. pdated Section 3.2.1: Installing STM32CubeMX standalone version, ection 3.2.3: Uninstalling STM32CubeMX plug-in installation ackage. pdated Section 3.3.1: Running STM32CubeMX as a standalone oplication. pdated Section 3.3.1: Running STM32CubeMX as a standalone oplication. pdated Section 5.15.1: Pinning and labeling signals on pins. dded Section 5.15.1: Pinning and labeling signals on pins. dded Section 5.16: Setting HAL timebase source pdated Figure 143: Configuration window tabs for GPIO, DMA and VIC settings (STM32F4 Series). dded note related to GPIO configuration in output mode in ection 4.5.12: GPIO configuration window; updated Figure 84: GPIO onfiguration window - GPIO selection. lodified Figure 323: Power Consumption Sequence, Figure 325: Step panagement functions, Figure 327: Enabling the transition checker otion on an already configured sequence - All transitions valid, igure 38: Enabling the transition checker option on an already onfigured sequence - At least one transition invalid. dded import pinout button icon in Section : Importing pinout. dded Section 1: Selecting/deselecting all peripherals. Modified igure 333: Power Consumption Calculator view after sequence uilding. Updated Section : Managing the whole sequence (load, ave and compare). Updated Figure 336: Description of the Results rea and Figure 483: Pinout view with MCUs selection and igure 549: Sequence table. pdated Section 1.3: Custom code generation. pdated Figure 483: Pinout view with MCUs selection and igure 548: Pinout view without MCUs selection and igure 548: Pinout view without MCUs selection and

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 <i>Section 2.2: Key features</i> . Added MacOS in <i>Section 3.1.1: Supported operating systems and</i> <i>architectures</i> , <i>Section 3.2.1: Installing STM32CubeMX standalone</i> <i>version</i> , <i>Section 3.2.3: Uninstalling STM32CubeMX standalone</i> <i>version</i> and <i>Section 3.4.3: Running STM32CubeMX plug-in from</i> <i>Eclipse IDE</i> . Added command lines allowing the generation of user files from templates in <i>Section 3.3.2: Running STM32CubeMX in command-line</i> <i>mode</i> . Updated new library installation sequence in <i>Section 3.4.2: Updater</i> <i>configuration</i> . Updated Figure 107: Pinout menus (Pinout tab selected) and Figure 108: Pinout menus (Pinout tab not selected) in <i>Section 5.7.3:</i> Pinout menu. Modified Table 16: Window menu. Updated Section 5.7: Output windows. Updated Figure 102: NVIC settings when using SysTick as HAL timebase, no FreeRTOS and Figure 103: NVIC settings when using FreeRTOS and SysTick as HAL timebase in Section 4.5.16: Setting HAL timebase source. Updated Figure 75: User Constants tab and Figure 76: 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 84: GPIO configuration window - GPIO selection, Figure 85: GPIO configuration grouped by peripheral and Figure 86: Multiple pins 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>



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 4.11.3: Advanced Settings tab 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 99: FreeRTOS</i> <i>heap usage.</i> Updated <i>Figure 84: 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: Tutorial 4 - Example of UART communications <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 183: Project Settings window. Added Restore Default feature: Updated Table 8: Peripheral and Middleware configuration window buttons and tooltips Updated Figure 77: Using constants for peripheral parameter settings.

Table 27. Document revision history (continued)



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 196: Automatic project import, Figure 197: Manual project import, Figure 198: Import Project menu - Try Import with errors and Figure 199: Import Project menu - Successful import after adjustments. Modified Appendix B.3.4: FatFs, B.3.5: FreeRTOS and B.3.6: LwIP.
			Added Appendix B.3.7: Libjpeg.
		4.20	Table 17: STM32CubeMX Chip view - Icons and color scheme:– Updated list of alternate function example.
	19		 Updated example and description corresponding to function mapping on a pin.
02-Mar-2017			 Added example and description for analog signals sharing the same pin.
			Updated Figure 87: Peripheral Configuration window (STM32F4 Series), Figure 75: User Constants tab, Figure 81: Consequence when deleting a user constant for peripheral configuration, Figure 82: Searching for a name in a user constant list and Figure 83: 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	4.21	Minor modifications in Section 1: STM32Cube overview. Updated Figure 41: New Project window - MCU selector and Figure 183: Project Settings window.
			Updated description of Project Settings in Section 4.11.1: Project tab. Updated Figure 194: 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 644: STM32Cube Embedded Software package</i> look- and-feel.



Date	Revision	STM32CubeMX release number	Changes
		4.22	Added STM32H7 to the list of supported STM32 Series. Added MCU data and documentation refresh capability in <i>Section 3.4:</i> <i>Getting updates using STM32CubeMX</i> and updated <i>Figure 18:</i> <i>Updater Settings window</i> .
			Added capability to identify close MCUs in Section 4.2: New Project window, updated Figure 41: 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 482: MCU selection.
06-Jul-2017	21		Updated Figure 51: STM32CubeMX Main window upon MCU selection.
			Added Rotate clockwise/Counter clockwise and Top/Bottom view in <i>Table 15: Pinout menu</i> .
			Added Section 4.1.4: Social links.
			Updated Figure 344: Configuring the SMPS mode for each step.
			Updated Section 6.2: STM32Cube code generation using Low Layer drivers.
			Updated Figure 509: Project Settings and toolchain selection.
	22	4.22.1	Added STM32L4+ Series in <i>Introduction</i> , Section 5.3: Power Consumption Calculator view and Section 6.2: STM32Cube code generation using Low Layer drivers.
05-Sep-2017			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?
	23	4.23	Added Section 1: General information.
			Renamed Display close button into Display similar items in <i>Section 4.2: New Project window</i> .
18-Oct-2017			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)



Date	Revision	STM32CubeMX 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 Figure 43: New Project window - Board selector. Updated Figure 52: STM32CubeMX Main window upon board selection (peripherals not initialized) and introductory sentence. Updated Figure 53: 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 4.14: Update Manager windows Added pin stacking function in Table 17: STM32CubeMX Chip view - lcons and color scheme. Section 6.2: STM32Cube code generation using Low Layer drivers: added STM32F0, STM32F3, STM32L0 in the list of product Series supporting low-level drivers. Section 12: Tutorial 2 - Example of FatFs on an SD card using STM32429I-EVAL evaluation board: updated Figure 529: Board selection and modified step 6 of the sequence for generating a project and running tutorial 2. Section 14: Tutorial 4 - Example of UART communications with an STM32L053xx Nucleo board: updated Figure 548: Selecting NUCLEO_LO53R8 board. Added Section 16: Tutorial 6 - Adding embedded software packs to user projects.
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.



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 41: New Project window - MCU selector, Figure 27: Enabling graphics choice in MCU selector, Figure 183: Project Settings window, Figure 188: 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 feature, Show favorite MCUs feature and Section 4.4.16: Graphics frameworks and simulator. Added Section 17: Tutorial 8 – Using STemWin 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		4.27	Updated STM32Cube logo on cover page. Replaced STMCube™ by STM32Cube™ in the whole document. Updated Section 1: STM32Cube overview.
			Updated <i>Figure 1: Overview of STM32CubeMX C code generation 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 Section 3.4 title into "Getting updates using STM32CubeMX". Suppressed figures Connection Parameters tab - No proxy and Connection Parameters tab - Use System proxy parameters. Updated Figure 22: Managing embedded software packages - Help menu.
	00		In Section 3.4.5: Installing embedded software packs, updated step 3f of the embedded software pack installation sequence and added Figure 27: License agreement acceptance.
	26		Section 4.2: New Project window: updated Figure 41: New Project window - MCU selector, Figure 42: Marking a favorite and Figure 43: New Project window - Board selector.
			Section 5.7.1: File menu: added caution note for New Project in Table 13: File menu functions. Updated Figure 107: Pinout menus (Pinout tab selected) and Figure 108: Pinout menus (Pinout tab not selected).
			Section 4.11: Project Manager view:
			 Added note related to project saving (step 3).
			 Updated Figure 183: Project Settings window Updated Section 4.11.1: Project tab and Figure 188: Selecting a different firmware location.
			Added Section 4.15.4: Component dependencies panel, Contextual help, Section 10: Support of additional software components using CMSIS-Pack standard and Section 17: Tutorial 7 – Using the X-Cube-BLE1 software pack.
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
			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.



Date Revision STM	I32CubeMX ase number	Changes
12-Nov-2018 27 (cont'd)	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-BLET. Updated Figure 1: Overview of STM32CubeMX C code generation flow, Figure 16: STM32Cube installation wizard, Figure 7: Closing STM32CubeMX perspective, Figure 9: Opening Eclipse plug-in, Figure 10: STM32CubeMX perspective, Figure 337: Overall peripheral consumption, Figure 450: User constant generating define statements, Figure 479: Selecting a CMSIS-Pack software component, Figure 480: Enabling and configuring a CMSIS-Pack software component, Figure 481: Project generated with CMSIS- Pack software component, Figure 482: MCU selection, Figure 483: Pinout view with MCUs selection, Figure 484: Pinout view without MCUs selection window, Figure 486: Timer configuration, Figure 487: Simple pinout configuration, Figure 488: Save Project As window, Figure 490: Generate Project Report - New project creation, Figure 491: Clock tree view, Figure 496: Pinout & Configuration view, Figure 491: Clock tree view, Figure 501: Infigure 501: GPIO configuration, Figure 503: DMA parameters configuration parameters, Figure 504: Middleware tooltip, Figure 505: USB Host configuration, Figure 507: System view with NatFs and USB enabled, Figure 508: FatFs define statements, Figure 509: Project Settings and toolchain selection, Figure 515: Downloading the firmware package, Figure 516: Unzipping the firmware package, Figure 517: System view with PatFs and USB enabled, Figure 507: System view with NatFatFs oorfiguration, Figure 507: Additional software components: no configurable parameters, Figure 576: Cade generation completion message, Figure 527: Import Project menu, Figure 575: Project Settings menu, Figure 5

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: 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 STM32MPUs, Section 4.6.2: Boot stages configuration, 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 183: Project Settings window, Figure 184: Project folder, Figure 188: Selecting a different firmware location, Figure 196: Automatic project import, Figure 197: Manual project import, Figure 200: Set unused pins window, Figure 470: STM32CubeMX generated DTS – Extract 3, Figure 479: Selecting a CMSIS-Pack software component, Figure 480: Enabling and configuring a CMSIS- Pack software component, Figure 534: FATFS tutorial - Project settings and Figure 535: 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 STM32MPUs, 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.



Date	Revision	STM32CubeMX release number	Changes
16-Apr-2019	29 (conťď)	5.1	Updated <i>Table 3: Window menu.</i> Updated figures 27 to 43, <i>Figure 194: Advanced Settings window</i> , figures 323 to 330, 332 to 335 and 337 to 346, <i>Figure 509: Project Settings and toolchain selection</i> and figures 537 to 547, Added <i>Figure 38: New Project window shortcuts, Figure 106:</i> <i>STM32MPUs: assignment options for GPIOs, Figure 642: Resource</i> <i>Manager: peripheral assignment view</i> and <i>Figure 644: 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 34: STM32CubeMX Home page, Figure 207: Selection of additional software components, Figure 208: Additional software components - Updated tree view, Figure 479: Selecting a CMSIS-Pack software component and Figure 577: 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 31		ChangesUpdated 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,
			STM32MPUs: assignment options for GPIOs, Figure 183: Project Settings window, Figure 354: DDR Suite - Connection to target, Figure 355: DDR Suite - Target connected, Figure 356: DDR activity logs, Figure 357: DDR interactive logs, Figure 358: DDR register loading, Figure 359: DDR test list from U-Boot SPL, Figure 360: DDR test suite results, Figure 361: 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 474: Project settings for STM32CubeIDE toolchain and Figure 509: Project Settings and toolchain selection. Added Figure 39: Enabling TrustZone.

Table 27. 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 feature, 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 34: STM32CubeMX Home page, Figure 38: New Project window shortcuts, Figure 43: New Project window - Board selector, Figure 46: Cross selector - Data refresh prerequisite, Figure 194: Advanced Settings window, Figure 204: Additional software window, Figure 200: Device tree generation for the Linux kernel, Figure 201: STM32CubeMX Device tree generation for TF-A, Figure 577: 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 33: Help menu: checking for updates, Figure 34: STM32CubeMX Home page, Figure 194: Advanced Settings window, Figure 204: Additional software window, Figure 311: ST Tools and Figure 530: 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 Table 6: Pinout menu and shortcuts. Added Figure 2: Full disk access for macOS and Figure 205: Component dependency resolution. Updated Figure 34: STM32CubeMX Home page, Figure 39: Enabling TrustZone, Figure 204: Additional software window. Removed former Figure 5: Auto-install command line and former Section 18.6: Why do I get the error "Java 8 update 45" when
	35	6.3	installing "Java 8 update 45" or a more recent version of the JRE?. 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 40: 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 479: Selecting a CMSIS-Pack software component and Figure 577: 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 90: NVIC configuration tab - FreeRTOS disabled.

Table 27. Document revision history (continued)



Date	Revision	STM32CubeMX release number	Changes
18-Feb-2022	37	6.5	Updated Introduction and Section 3.1.1: Supported operating systems and architectures.
			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.
14-Jun-2022	38	6.6	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. Updated Table 1: Command line summary. Updated Figure 310: About window.
			Added Section 4.17: CAD Resources view section 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.
17-Nov-2022	39	6.7	Updated Section 2.2: Key features and Section 17: Tutorial 7 – Using the X-Cube-BLE1 software pack. 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?. Updated Figure 41: New Project window - MCU selector, Figure 42: Marking a favorite, Figure 29: New Project window - MCU list with close function, Figure 30: New Project window - List showing close MCUs, and Figure 310: About window. Minor text edits across the whole document.
21-Feb-2023	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. Added Section 4.18: Boot path and its subsections. Removed former Section 5.3.4: DDR tuning and DDR tuning tab (read-only). Updated Figure 41: New Project window - MCU selector, Figure 183: Project Settings window, and Figure 605: 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 220 to 224 and Figure 272: Code generated with secure manager API. Added Section 4.18.6: 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 46: Cross selector - Data refresh prerequisite, Figure 216: Boot paths for STM32H57x devices, Figure 226: Select the STM32H5 device, Figure 228: Boot paths for STM32H56x devices, figures 231 to 244, figures 167 to 255, figures 257 to 259, figures 263 to 267, Figure 271: Secure manager API configuration, and Figure 310: 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 240, Figure 254, and Figure 266 titles Updated Figure 266: Boot path and Debug Authentication tab. Updated figures 216 to 224 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
			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.
		6.10.0	Updated Table 18: Boot paths without TrustZone [®] (TZEN = 0) and Table 19: Boot paths with TrustZone [®] (TZEN = 1).
20-Nov-2023	43		Updated Figure 220: Application boot path (OEM-uRoT assembled), Figure 221: Application boot path: ST-iRoT and uRoT secure/nonsecure project, Figure 223: Application boot path: ST-iRoT
			dual figure, Figure 238: Project provisioning, Figure 240: Boot path and debug authentication panel, Figure 247: IDE post build commands, Figure 258: IDE post build commands, Figure 270: IDE post build commands, figures 352 to 355, Figure 358: DDR register loading, and Figure 359: DDR test list from U-Boot SPL.
			Removed former <i>Figure 167: Selection of the OEMiRoT_Boot project</i> and <i>Figure 195: Generated project</i> .
			Minor text edits across the whole document.
	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 4.18.6: 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).
13-Mar-2024			Added Section 4.18.7: 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 221: Application boot path: ST- iRoT and uRoT secure/nonsecure project, Figure 223: Application boot path: ST-iRoT dual figure, Figure 233: Boot path selection, Figure 240: Boot path and debug authentication panel, Figure 244: Generate the code, Figure 254: Boot path and Debug Authentication tab, Figure 266: Boot path and Debug Authentication tab, and Figure 275: Boot path project. Minor text edits across the whole document.

Table 27. Document revision history (continued)



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 STM32MPUs, Section 4.18.7: 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 (STM32MPUs), Section 4.7: RIF configuration, Section 4.18.8: 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 Table 20: Boot paths for STM32H7RS devices. Added Figure 20: Connection failure and Figure 32: Updates are available. Updated Figure 45: Popup window - Starting a project from an example, Figure 280: Boot path project, Figure 509: Project Settings and toolchain selection, and Figure 591: 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: Memories protection for STM32MP2 series, Section 4.19: User authentication, and Section 5.5.1: STM32U5, STM32H5, STM32WBA, and STM32WBAM with TrustZone activated Added Section 4.7.10: Memories protection for STM32MP2 series, Section 4.7.11: Memories protection for STM32N6 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)



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.

© 2024 STMicroelectronics - All rights reserved



UM1718 Rev 46