AN5701

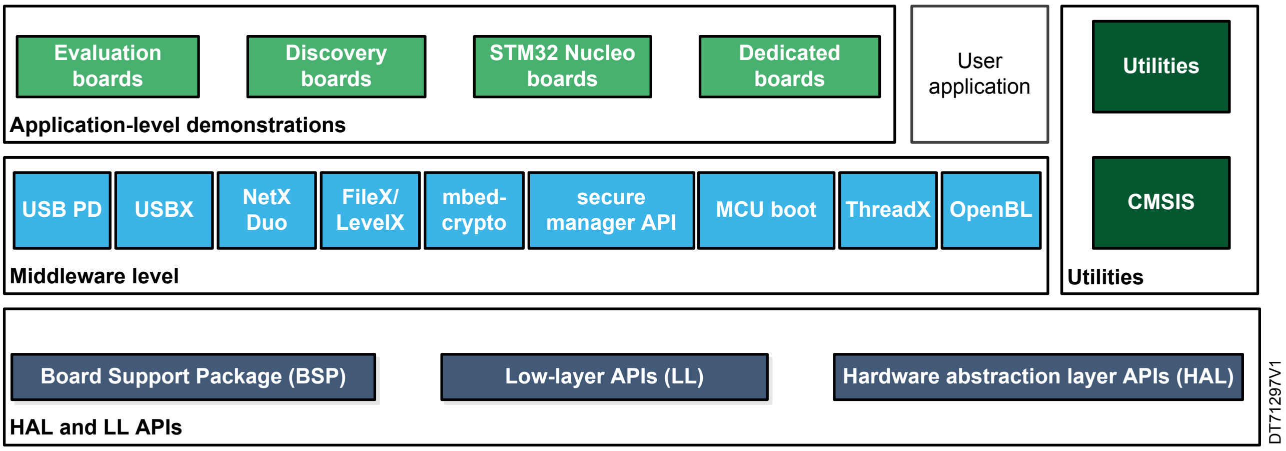
Application note

Introduction to STM32Cube MCU Package examples for STM32U5 MCUs

Introduction

The STM32CubeU5 MCU Package is delivered with a rich set of examples running on STMicroelectronics boards. The examples are organized by boards. They are provided with preconfigured projects for the main supported toolchains (refer to Figure 1. STM32CubeU5 firmware components).

Figure 1. **STM32CubeU5 firmware components**



1 Reference documents

The following items make up a reference set for the examples presented in this application note:

- The latest release of the STM32CubeU5 MCU Package for the 32-bit microcontrollers in the STM32U5 series based on the Arm® Cortex®-M processor with Arm®TrustZone®
- *Getting started with STM32CubeU5 for STM32U5 Series* (UM2883)
- *Description of STM32U5 HAL and low-layer drivers* (UM2911)
- *Getting started with STM32CubeU5 TFM application* (UM2851)
- *Overview of secure boot and secure firmware update solution on Arm® TrustZone® STM32 series microcontrollers* (AN5447)

Note: *Arm and TrustZone are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.*

2 STM32CubeU5 examples

The examples are classified depending on the STM32Cube level that they apply to. They are named as follows:

- **Examples**
  These examples use only the HAL and BSP drivers (middleware not used). Their objective is to demonstrate the product or peripheral features and usage. They are organized per peripheral (one folder per peripheral, such as TIM). Their complexity level ranges from the basic usage of a given peripheral, such as PWM generation using a timer, to the integration of several peripherals, such as how to use DAC for a signal generation with synchronization from TIM6 and DMA. The usage of the board resources is reduced to the strict minimum.
- **Examples_LL**
  These examples use only the LL drivers (HAL drivers and middleware components not used). They offer an optimum implementation of typical use cases of the peripheral features and configuration sequences. The examples are organized per peripheral (one folder for each peripheral, such as TIM) and are principally deployed on Nucleo boards.
- **Examples_MIX**
  These examples use only HAL, BSP, and LL drivers (Middleware components are not used). They aim at demonstrating how to use both HAL and LL APIs in the same application in order to combine the advantages of both APIs:
  - HAL offers high-level function-oriented APIs with high portability level by hiding product/IPs complexity for end-users.
  - LL provides low-level APIs at the register level with better optimization.

  The examples are organized per peripheral (one folder for each peripheral, such as TIM) and are exclusively deployed on Nucleo boards.
- **Applications**
  The applications demonstrate product performance and how to use the available middleware stacks. They are organized either by middleware (one folder per middleware, such as Azure® RTOS ThreadX) or product feature that requires high-level firmware bricks (such as LPBAM). The integration of applications that use several middleware stacks is also supported.
- **Demonstrations**
  The demonstrations aim at integrating and running the maximum number of peripherals and middleware stacks to showcase the product features and performance.
- **Template project**
  The template project is provided to enable the user to quickly build a firmware application using HAL and BSP drivers on a given board.
- **Template_LL project**
  The template LL projects are provided to enable the user to quickly build a firmware application using LL drivers on a given board.

The examples are located under `STM32Cube_FW_U5_VX.Y.Z\Projects\`.

The examples in the default product configuration with the Arm® TrustZone® disabled have the same structure:

- `*\Inc` folder, containing all header files
- `*\Src` folder, containing the sources code
- `*\EWARM`, `*\MDK-ARM`, and `*\STM32CubeIDE` folders, containing the preconfigured project for each toolchain
- `*\README.md` and `*\readme.html` file, describing the example behavior and the environment required to run the example

The examples with the Arm® TrustZone® enabled are suffixed with "_TrustZone" (except TFM applications) and have the same structure:

- `*\Secure\Inc` folder, containing all secure project header files
- `*\Secure\Src` and `*\Secure_nsclib\` folders, containing all secure project sources code
- `*\NonSecure\Inc` folder, containing all nonsecure project header files
- `*\NonSecure\Src` folder, containing all nonsecure project sources code
- `*\EWARM`, `*\MDK-ARM`, and `*\STM32CubeIDE` folders, containing the preconfigured project for each toolchain
- `*\README.md` and `*\readme.html` files, describing the example behavior and the environment required to run the example

To run the example, proceed as follows:

1. Open the example using your preferred toolchain.
2. Rebuild all files and load the image into target memory.
3. Run the example by following the `*\README.md` and `*\readme.html` instructions.

Note: *Refer to “Development toolchains and compilers” and “Supported devices and evaluation boards” sections of the firmware package release notes to know more about the software/hardware environment used for the MCU Package development and validation. The correct operation of the provided examples is not guaranteed in other environments, for example, when using different compilers or board versions.*

The examples can be tailored to run on any compatible hardware: simply update the BSP drivers for your board, provided it has the same hardware functions (LED, LCD, pushbuttons, and others). The BSP is based on a modular architecture that can be easily ported to any hardware by implementing low-level routines.

Table 1. STM32CubeU5 firmware examples contains the list of examples provided with the STM32CubeU5 MCU Package.

In this table, the label MX means that the projects are created using STM32CubeMX, the STM32Cube initialization code generator. Those projects can be opened with this tool to modify the projects themselves. The other projects are manually created to demonstrate the product features. In this table, the label TrustZone® means that the projects are created for devices with Arm® TrustZone® enabled. Read the project `*\README.md` and `*\readme.html` files for user option bytes configuration.

Table 1. STM32CubeU5 firmware examples

STM32CubeMX-generated examples are highlighted with the icon MX.

Level	Module name	Project name	Description	STM32U5A9J-DK STM32U5G9J-DK1	STM32U5G9J-DK2	STM32U575I-EV	NUCLEO-U5A5ZJ-Q	NUCLEO-U575ZI-Q	NUCLEO-U545RE-Q	B-U585I-IOT02A
Templates	-	TrustZoneDisabled	This project provides a reference template based on the STM32Cube HAL API that can be used to build any firmware application when security is not enabled (TZEN=0).	X	X	X	X	X	X	X
		TrustZoneEnabled	This project provides a reference template based on the STM32Cube HAL API that can be used to build any firmware application when TrustZone® security is activated **(option bit TZEN=1)**.	X	X	X	X	X	X	X
	Total number of templates: 14			2	2	2	2	2	2	2
Templates_LL	-	TrustZoneDisabled	This project provides a reference template through the LL API that can be used to build any firmware application.	X	X	X	X	X	X	X
	Total number of templates_ll: 7			1	1	1	1	1	1	1
Examples	-	BSP	How to use the different BSP drivers of the board.	X	X	X	-	-	-	X
	ADC	ADC_AnalogWatchdog	How to use an ADC peripheral with an ADC analog watchdog to monitor a channel and detect when the corresponding conversion data is outside the window thresholds.	-	-	-	-	MX	-	-
		ADC_DMA_Transfer	How to configure and use the ADC2 to convert an external analog input and get the result using a DMA transfer through the HAL API.	-	-	-	-	MX	-	-
		ADC_DifferentialMode	How to configure and use the ADC1 to convert an external analog input in differential mode, difference between external voltage on VinN and VinP.	-	-	-	-	MX	-	-
		ADC_SingleConversion_TriggerSW_IT	How to use ADC to convert a single channel at each SW start, conversion performed using a programming model: interrupt.	-	-	-	-	MX	-	-
	COMP	COMP_Interrupt	How to use a comparator peripheral to compare a voltage level applied on a GPIO pin to the internal voltage reference (VREFINT), in interrupt mode.	-	-	-	-	MX	-	-
	CORDIC	CORDIC_Sin_DMA	How to use the CORDIC peripheral to calculate an array of sines in DMA mode.	-	-	-	-	MX	-	-
	CORTEX	CORTEXM_ModePrivilege	How to modify the thread mode privilege access and stack. Thread mode is entered on reset or when returning from an exception.	-	-	-	-	MX	-	-
		CORTEXM_SysTick	How to use the default SysTick configuration with a 1 ms timebase to toggle LEDs.	-	-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32U5A9J-DK	STM32U5G9J-DK1	STM32U5G9J-DK2	STM32U575I-EV	NUCLEO-U5A5ZJ-Q	NUCLEO-U575ZI-Q	NUCLEO-U545RE-Q	B-U585I-IOT02A
Examples	CRC	CRC_Bytes_Stream_7bit_CRC	How to configure the CRC using the HAL API. The CRC (cyclic redundancy check) calculation unit computes 7-bit CRC codes derived from buffers of 8-bit data (bytes).	-		-	-	-	MX	-	-
		CRC_Example	How to configure the CRC using the HAL API. The CRC (cyclic redundancy check) calculation unit computes the CRC code of a given buffer of 32-bit data words, using a fixed generator polynomial (0x4C11DB7).	-		-	-	-	MX	-	-
		CRC_UserDefinedPolynomial	How to configure the CRC using the HAL API. The CRC (cyclic redundancy check) calculation unit computes the 8-bit CRC code for a given buffer of 32-bit data words, based on a user-defined generating polynomial.	-		-	-	-	MX	-	-
	CRYP	CRYP_AES_GCM	How to use the CRYPTO peripheral to encrypt and decrypt data using AES with Galois/Counter mode (GCM).	-		-	-	-	-	-	MX
		CRYP_SAES_ECB_CBC	How to use the SecureAES coprocessor (SAES) peripheral to encrypt and decrypt data using AES, ECB, and CBC algorithms when security is not enabled (TZEN=0).	-		-	-	-	-	-	MX
		CRYP_SAES_SharedKey	How to use the secure AES coprocessor (SAES) peripheral to share application keys with the AES peripheral.	-		-	-	-	-	-	MX
		CRYP_SAES_WrapKey	How to use the secure AES coprocessor (SAES) peripheral to wrap application keys using the hardware secret key DHUK then use it to encrypt in polling mode.	-		-	-	-	-	-	MX
	DAC	DAC_SignalsGeneration_DMA	How to use the DAC peripheral to generate sine signal using the DMA controller.	-		-	-	-	MX	-	-
		DAC_SimpleConversion	How to use the DAC peripheral to do a simple conversion.	-		-	-	-	MX	-	-
	DCACHE	DCACHE_Maintenance	How to do data cache maintenance on a shared memory buffer accessed by 2 masters (CPU and DMA).	-		-	MX	-	-	-	-
	DCMI	DCMI_ContinousCap_ EmbeddedSynchMode	This example provides a description of how to configure the DCMI peripheral in Continuous Mode and EmbeddedSynchronization mode, with suspend and resume of the frame capture based on the STM32Cube HAL API when security is not enabled (TZEN=0).	-		-	MX	-	-	-	-
	DLYB	DLYB_OSPI_NOR_FastTuning	How to use delay block (DLYB) with a fast tuning.	-		-	MX	-	-	-	-
		DLYB_OSPI_PSRAM_ExhaustiveTuning	How to use delay block (DLYB) with an exhaustive tuning.	-		-	-	-	-	-	MX
	DMA	DMA_DataHandling	How to use the DMA controller to do data handling between transferred data from the source and transfer to the destination through the HAL API.	-		-	-	-	MX	MX	-

Level	Module name	Project name	Description	STM32U5A9J-DK	STM32U5G9J-DK1	STM32U5G9J-DK2	STM32U575I-EV	NUCLEO-U5A5ZJ-Q	NUCLEO-U575ZI-Q	NUCLEO-U545RE-Q	B-U585I-IOT02A
Examples	DMA	DMA_FLASHToRAM	How to use a DMA to transfer a word data buffer from flash memory to embedded SRAM through the HAL API.	MX		MX	-	MX	MX	MX	-
		DMA_LinkedList	How to use the DMA to perform a list of transfers. The transfer list is organized as linked-list, each time the current transfer ends the DMA automatically reloads the next transfer parameters, and starts it (without CPU intervention).	-		-	-	-	MX	-	-
		DMA_RepeatedBlock	How to configure and use the DMA HAL API to perform repeated block transactions.	-		-	MX	-	-	-	-
		DMA_Trigger	How to configure and use the DMA HAL API to perform DMA triggered transactions.	-		-	MX	-	-	-	-
	DMA2D	DMA2D_BlendingWithAlphaInversion	This example provides a description of how to configure the DMA2D peripheral in Memory_to_Memory with blending transfer and alpha inversion mode based on the STM32Cube HAL API when security is not enabled (TZEN=0).	-		-	MX	-	-	-	-
	DSI	DSI_ULPM_DataClock	This example provides a description of how to use the embedded LCD DSI controller (using IPs LTDC and DSI host) to drive the **J025FICN0201W (480x480)** display mounted on an MB1892 board and manage entry and exit in DSI ULPM mode on data lane and clock lane.	MX		-	-	-	-	-	-
		DSI_VideoMode_SingleBuffer	This example provides a description of how to use the embedded LCD DSI controller (using IPs LTDC and DSI host) to drive the **J025FICN0201W (480x480)** display mounted on an MB1892 board.	MX		-	-	-	-	-	-
	FDCAN	FDCAN_Loopback	How to configure the FDCAN to operate in loopback mode.	-		-	MX	-	-	-	-
	FLASH	FLASH_ChangeOptionBytes	How to configure and use the FLASH HAL API to change the STM32U5 devices option bytes.	-		-	-	-	X	-	-
		FLASH_EraseProgram	How to configure and use the FLASH HAL API to erase and program the internal flash memory .	MX		MX	-	-	MX	MX	-
		FLASH_EraseProgram_TrustZone	How to configure and use the FLASH HAL API to erase and program the internal flash memory when TrustZone® security is activated **(option bit TZEN=1)**.	-		-	-	-	X	-	-
	FMAC	FMAC_IIR_PollingToDMA	How to use the FMAC peripheral to perform an IIR filter from polling mode to DMA mode.	-		-	-	-	MX	-	-
	FMC	FMC_SRAM	This example describes how to configure the FMC controller to access the SRAM memory based on the STM32Cube HAL API that can be used to build any firmware application when security is not enabled (TZEN=0).	-		-	MX	-	-	-	-

Level	Module name	Project name	Description	STM32U5A9J-DK	STM32U5G9J-DK1	STM32U5G9J-DK2	STM32U575I-EV	NUCLEO-U5A5ZJ-Q	NUCLEO-U575ZI-Q	NUCLEO-U545RE-Q	B-U585I-IOT02A
Examples	FMC	FMC_SRAM_ReadWrite_DMA	This example describes how to configure the FMC controller and the DMA to access the SRAM memory based on the STM32Cube HAL API when security is not enabled (TZEN=0).	-		-	MX	-	-	-	-
	GFXTIM	GFXTIM_InterruptOnAbsoluteLine	Use of a GFXTIM peripheral to generate interrupts for specific absolute lines.	-		MX	-	-	-	-	-
	GPIO	GPIO_EXTI	How to configure external interrupt lines.	-		-	-	-	MX	-	MX
		GPIO_IOToggle	How to configure and use GPIOs through the HAL API.	MX		MX	MX	MX	MX	MX	MX
		GPIO_IOToggle_TrustZone	How to use HAL GPIO to toggle secure and unsecure IOs when TrustZone® security is activated (option bit TZEN=1).	-		-	-	-	MX	-	MX
	GTZC	GTZC_MPCWM_IllegalAccess_TrustZone	How to use GTZC MPCWM-TZIC to build any example when TrustZone® security is activated **(option bit TZEN=1)**.	-		-	MX	-	-	-	-
		GTZC_TZSC_MPCBB_TrustZone	How to use HAL GTZC MPCBB to build any example with illegal access detection when TrustZone® security is activated **(option bit TZEN=1)**.	-		-	-	-	MX	-	-
	HAL	HAL_TimeBase_RTC_ALARM	How to customize HAL using RTC alarm as main source of time base, instead of Systick.	-		-	-	-	MX	-	-
		HAL_TimeBase_RTC_WKUP	How to customize HAL using RTC wake-up as the main source of time base, instead of Systick.	-		-	-	-	MX	-	-
		HAL_TimeBase_TIM	How to customize HAL using a general-purpose timer as the main source of time base, instead of Systick.	-		-	-	-	MX	-	-
	HASH	HASH_HMAC_SHA1MD5	How to use the HASH peripheral to hash data with HMAC SHA-1 and HMAC MD5 algorithms.	-		-	-	-	MX	-	-
	I2C	I2C_TwoBoards_AdvComIT	How to handle several I²C data buffer transmissions/receptions between a master and a slave device using interrupts.	-		-	-	-	MX	-	-
		I2C_TwoBoards_ComDMA	How to handle I²C data buffer transmission/reception between two boards in DMA mode.	-		-	-	-	MX	-	-
		I2C_TwoBoards_ComDMA_ Autonomous_Master	How to handle I²C data buffer transmission/reception autonomously between two boards in DMA mode through the GPDMA1_Channel_3 trigger.	-		-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32U5A9J-DK	STM32U5G9J-DK1	STM32U5G9J-DK2	STM32U575I-EV	NUCLEO-U5A5ZJ-Q	NUCLEO-U575ZI-Q	NUCLEO-U545RE-Q	B-U585I-IOT02A
Examples	I2C	I2C_TwoBoards_ComDMA_Autonomous_Slave	How to handle I²C data buffer transmission/reception autonomously between two boards in DMA mode through the GPDMA1_Channel_3 trigger.	-		-	-	-	MX	-	-
		I2C_TwoBoards_ComDMA_LowPower	How to handle I²C data buffer transmission/reception in low-power mode between two boards in DMA mode.	-		-	-	-	MX	-	-
		I2C_TwoBoards_ComIT	How to handle I²C data buffer transmission/reception between two boards using interrupts.	-		-	-	-	MX	MX	-
		I2C_TwoBoards_ComPolling	How to handle I²C data buffer transmission/reception between two boards, in polling mode.	-		-	-	-	MX	-	-
		I2C_TwoBoards_RestartAdvComIT	How to perform multiple I²C data buffer transmissions/receptions between two boards in interrupt mode and with restart condition.	-		-	-	-	MX	-	-
		I2C_TwoBoards_RestartComIT	How to handle single I²C data buffer transmission/reception between two boards in interrupt mode and with restart condition.	-		-	-	-	MX	-	-
		I2C_WakeUpFromStop	How to handle I²C data buffer transmission/reception between two boards using an interrupt when the device is in Stop mode.	-		-	-	-	MX	MX	-
	ICACHE	ICACHE_Memory_Remap	How to execute code from a remapped region configured through the ICACHE HAL driver.	-		-	MX	-	-	-	-
	IWDG	IWDG_Reset	How to handle the IWDG reload counter and simulate a software fault that generates an MCU IWDG reset after a preset lapse of time.	-		-	-	-	MX	-	-
		IWDG_WindowMode	How to periodically update the IWDG reload counter and simulate a software fault that generates an MCU IWDG reset after a preset lapse of time.	-		-	-	-	MX	-	-
	JPEG	JPEG_DecodingFromFLASH_DMA	This example demonstrates how to decode a JPEG image stored in the internal flash memory using the JPEG hardware decoder in DMA mode. The decoded image is stored in the internal flash memory.	-		MX	-	-	-	-	-
		JPEG_EncodingFromFLASH_DMA	This example demonstrates how to encode an RGB image stored in the internal flash memory using the JPEG hardware encoder in DMA mode and save it in the internal flash memory.	-		MX	-	-	-	-	-
	LPTIM	LPTIM_IC_LSE	This example shows how to use the LPTIM peripheral to measure the frequency of an external signal in low-power mode using the LSE as a counter clock, through the HAL LPTIM API.	-		-	-	-	MX	-	-
		LPTIM_PWM_LSE	This example describes how to configure and use LPTIM to generate a PWM in low power mode using the LSE as a counter clock, through the HAL LPTIM API.	-		-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32U5A9J-DK	STM32U5G9J-DK1	STM32U5G9J-DK2	STM32U575I-EV	NUCLEO-U5A5ZJ-Q	NUCLEO-U575ZI-Q	NUCLEO-U545RE-Q	B-U585I-IOT02A
Examples	MDF	ADF_AudioRecorder	This example shows how to use the MDF HAL API (ADF instance) to perform mono audio recording.	-		-	MX	-	-	-	-
		ADF_AudioSoundDetector	This example shows how to use the MDF HAL API (ADF instance) to use audio sound activity detection.	-		-	-	-	-	-	MX
	OCTOSPI	OSPI_HyperRAM_MemoryMapped	How to use an OSPI HyperRAM memory in memory-mapped mode.	-		-	MX	-	-	-	-
		OSPI_HyperRAM_ReadWrite_IT	How to use an OSPI HyperRAM memory in indirect mode.	-		-	MX	-	-	-	-
		OSPI_NOR_AutoPolling_DTR	How to use an OSPI NOR memory in automatic polling mode.	-		-	-	-	-	-	MX
		OSPI_NOR_MemoryMapped	How to use an OSPI NOR memory in memory-mapped mode.	-		-	MX	-	-	-	-
		OSPI_NOR_ReadWrite_DMA_DTR	How to use an OSPI NOR memory in DMA mode.	-		-	-	-	-	-	MX
		OSPI_PSRAM_ExecuteInPlace	How to execute code from OSPI memory after code loading.	-		-	-	-	-	-	MX
		OSPI_PSRAM_MemoryMapped	How to use an OSPI PSRAM memory in memory-mapped mode.	-		-	MX	-	-	-	MX
	OPAMP	OPAMP_Follower	How to configure the OPAMP peripheral in follower mode interconnected with DAC and COMP.	-		-	-	-	MX	-	-
	OSPI	OSPI_NOR_MemoryMapped	How to use an OSPI NOR memory in memory-mapped mode.	X		-	-	-	-	-	-
	OTFDEC	OTFDEC_Data_Decrypt	This example describes how to decrypt data (encrypted with the CRYPT peripheral) located on the OCTOSPI external flash memory using the OTFDEC peripheral. The led lights of the B_U585I_IOT02 board are used to monitor the status as following: LED_GREEN is ON when the checked data is correct.	-		-	-	-	-	-	MX
	PKA	PKA_ECCDoubleBaseLadder	How to use the PKA to run an ECC double base ladder operation. This project is targeted to run on STM32U585AIIxQ devices on a B-U585I-IOT02A board from STMicroelectronics.	-		-	-	-	-	-	MX
		PKA_ECCProjective2Affine	How to use the PKA to run ECC projective to affine operation. This project is targeted to run on STM32U585xx devices on a B-U585I-IOT02A board from STMicroelectronics.	-		-	-	-	-	-	MX

Level	Module name	Project name	Description	STM32U5A9J-DK STM32U5G9J-DK1	STM32U5G9J-DK2	STM32U575I-EV	NUCLEO-U5A5ZJ-Q	NUCLEO-U575ZI-Q	NUCLEO-U545RE-Q	B-U585I-IOT02A
Examples	PKA	PKA_ECDSA_Sign	How to compute a signed message regarding the elliptic curve digital signature algorithm (ECDSA).	-	-	-	-	-	-	MX
		PKA_ModExpProtected_IT	How to use the PKA to run a protected modular exponentiation operation. This project is targeted to run on STM32U585AIIxQ devices on a B-U585I-IOT02A board from STMicroelectronics.	-	-	-	-	-	-	MX
	PWR	PWR_LPMODE_RTC	How to enter the system in different available low power modes and wake up from these modes by using an interrupt from the RTC wake-up timer.	MX	MX	-	-	MX	-	-
		PWR_ModesSelection	How to configure the system using HAL drivers to measure the current consumption in different low-power modes.	-	-	-	-	MX	MX	-
		PWR_SLEEP	How to enter Sleep mode and wake up from this mode by using an interrupt.	-	-	-	MX	MX	MX	MX
		PWR_STANDBY	How to enter Standby mode and wake up from this mode by using an external reset or the WKUP pin.	-	-	-	-	MX	MX	MX
	RAMCFG	RAMCFG_ECC_Error_Generation	How to configure and use the RAMCFG HAL API to manage ECC errors via the RAMCFG peripheral.	MX	MX	-	-	MX	-	-
		RAMCFG_WriteProtection	How to configure and use the RAMCFG HAL API to configure the RAMCFG SRAM write protection page.	-	-	MX	-	-	-	-
	RCC	RCC_ClockConfig	Configuration of the system clock (SYSCLK) and modification of the clock settings in Run mode, using the RCC HAL API.	MX	MX	MX	MX	MX	-	MX
		RCC_LSEConfig	Enabling/disabling of the low-speed external (LSE) RC oscillator (about 32 kHz) at runtime , using the RCC HAL API.	-	-	-	-	MX	-	-
		RCC_LSIConfig	How to enable/disable the low-speed internal (LSI) RC oscillator (about 32 kHz) at runtime , using the RCC HAL API.	-	-	-	-	MX	-	-
	RNG	RNG_MultiRNG	Configuration of the RNG using the HAL API. This example uses the RNG to generate 32-bit long random numbers.	-	-	-	-	MX	-	-
		RNG_MultiRNG_IT	Configuration of the RNG using the HAL API. This example uses RNG interrupts to generate 32-bit long random numbers.	-	-	-	-	MX	-	-
	RTC	RTC_ActiveTamper	Configuration of the active tamper detection with backup registers erase.	-	-	-	-	MX	MX	-

Level	Module name	Project name	Description	STM32U5A9J-DK	STM32U5G9J-DK1	STM32U5G9J-DK2	STM32U575I-EV	NUCLEO-U5A5ZJ-Q	NUCLEO-U575ZI-Q	NUCLEO-U545RE-Q	B-U585I-IOT02A
Examples	RTC	RTC_Alarm	Configuration and generation of an RTC alarm using the RTC HAL API.	MX		MX	-	-	MX	MX	-
		RTC_Calendar	Configuration of the calendar using the RTC HAL API.	-		-	MX	-	-	-	-
		RTC_LSI	Use of the LSI clock source autocalibration to get a precise RTC clock.	-		-	-	-	MX	-	-
		RTC_LowPower_STANDBY_WUT	How to periodically enter and wake up from Standby mode thanks to the RTC wake-up timer (WUT).	-		-	-	-	MX	-	-
		RTC_Tamper	Configuration of the tamper detection with backup registers erase.	-		-	-	-	-	-	MX
		RTC_TimeStamp	Configuration of the RTC HAL API to demonstrate the timestamp feature.	-		-	-	-	MX	-	-
		RTC_TrustZone	How to configure the TrustZone-aware RTC peripheral when TrustZone® security is activated (option bit TZEN=1): some features of the RTC can be secure while others are non-secure.	-		-	-	-	MX	-	-
	SAI	SAI_AudioPlay	How to play an audio file via SAI using DMA circular mode.	-		-	MX	-	-	-	-
	SD	SD_ReadWrite_DMALinkedList	This example performs some write and read transfers to an SD card with SDMMC IP internal DMA mode based on the linked list feature.	-		-	MX	-	-	-	-
	SMARTCARD	SMARTCARD_ComDMA	This example aims to show how to communicate with a smartcard using DMA mode.	-		-	MX	-	-	-	-
	SMBUS	SMBUS_TwoBoards_ComIT_Autonomous_Master	How to handle SMBUS data buffer transmission/reception between two boards, using autonomous mode.	-		-	-	-	MX	-	-
		SMBUS_TwoBoards_ComIT_Autonomous_Slave	How to handle SMBUS data buffer transmission/reception between two boards, using autonomous mode.	-		-	-	-	MX	-	-
		SMBUS_TwoBoards_ComIT_Master	How to handle SMBUS data buffer transmission/reception between two boards, in interrupt mode.	-		-	-	-	MX	-	-
		SMBUS_TwoBoards_ComIT_Slave	How to handle SMBUS data buffer transmission/reception between two boards, in interrupt mode.	-		-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32U5A9J-DK	STM32U5G9J-DK1	STM32U5G9J-DK2	STM32U575I-EV	NUCLEO-U5A5ZJ-Q	NUCLEO-U575ZI-Q	NUCLEO-U545RE-Q	B-U585I-IOT02A
Examples	SPI	SPI_FullDuplex_ComDMA_Autonomous_Master	How to handle data buffer transmission/reception autonomously between two boards via SPI using DMA mode.	-		-	-	-	MX	-	-
		SPI_FullDuplex_ComDMA_Autonomous_Slave	How to handle data buffer transmission/reception autonomously between two boards via SPI using DMA mode.	-		-	-	-	MX	-	-
		SPI_FullDuplex_ComDMA_LowPower_Master	How to handle data buffer transmission/reception in low-power mode between two boards via SPI using DMA mode.	-		-	-	-	MX	-	-
		SPI_FullDuplex_ComDMA_LowPower_Slave	How to handle data buffer transmission/reception in Low Power mode between two boards via SPI using DMA mode.	-		-	-	-	MX	-	-
		SPI_FullDuplex_ComDMA_Master	How to handle data buffer transmission/reception between two boards via SPI using DMA mode.	-		-	-	-	MX	-	-
		SPI_FullDuplex_ComDMA_Slave	How to handle data buffer transmission/reception between two boards via SPI using DMA mode.	-		-	-	-	MX	-	-
		SPI_FullDuplex_ComIT_Master	How to handle data buffer transmission/reception between two boards via SPI using Interrupt mode.	-		-	-	-	MX	MX	-
		SPI_FullDuplex_ComIT_Slave	How to handle data buffer transmission/reception between two boards via SPI using Interrupt mode.	-		-	-	-	MX	MX	-
		SPI_FullDuplex_ComPolling_Master	How to handle data buffer transmission/reception between two boards via SPI using polling mode.	-		-	-	-	MX	-	-
		SPI_FullDuplex_ComPolling_Slave	How to handle data buffer transmission/reception between two boards via SPI using polling mode.	-		-	-	-	MX	-	-
	TIM	TIM_InputCapture	How to use the TIM peripheral to measure an external signal frequency.	-		-	-	-	MX	-	-
		TIM_OCActive	Configuration of the TIM peripheral in Output Compare Active mode (when the counter matches the capture/compare register, the corresponding output pin is set to its active state).	-		-	-	-	MX	-	-
		TIM_OCInactive	Configuration of the TIM peripheral in Output Compare Inactive mode with the corresponding Interrupt requests for each channel.	-		-	-	-	MX	-	-
		TIM_OCToggle	Configuration of the TIM peripheral to generate four different signals at four different frequencies.	-		-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32U5A9J-DK	STM32U5G9J-DK1	STM32U5G9J-DK2	STM32U575I-EV	NUCLEO-U5A5ZJ-Q	NUCLEO-U575ZI-Q	NUCLEO-U545RE-Q	B-U585I-IOT02A
Examples	TIM	TIM_PWMInput	How to use the TIM peripheral to measure the frequency and duty cycle of an external signal.	-		-	-	-	MX	-	-
		TIM_PWMOutput	This example shows how to configure the TIM peripheral in PWM (pulse width modulation) mode.	-		-	-	-	MX	-	-
	TSC	TSC_BasicAcquisition	Use of the TSC to perform continuous acquisitions of one channel in polling mode.	-		-	MX	-	-	-	-
		TSC_BasicAcquisition_Interrupt	Use of the TSC to perform continuous acquisitions of one channel in Interrupt mode.	-		-	MX	-	-	-	-
	UART	UART_Printf	Rerouting of the C library printf function to the UART.	-		-	MX	-	-	-	-
		UART_ReceptionToIdle_CircularDMA	How to use the HAL UART API for reception to IDLE an event in circular DMA mode.	-		-	-	-	MX	-	-
		UART_TwoBoards_ComDMA	UART transmission (transmit/receive) in DMA mode between two boards.	-		-	-	-	MX	-	-
		UART_TwoBoards_ComDMAlinkedlist	UART transmission (transmit/receive) in DMA mode using linkedlist between two boards.	-		-	-	-	MX	-	-
		UART_TwoBoards_ComIT	UART transmission (transmit/receive) in Interrupt mode between two boards.	-		-	-	-	MX	-	-
		UART_TwoBoards_ComPolling	UART transmission (transmit/receive) in polling mode between two boards.	-		-	-	-	MX	-	-
		UART_WakeUpFromStopUsingFIFO	Configuration of an UART to wake up the MCU from Stop mode with a FIFO level when a given stimulus is received.	-		-	-	-	MX	-	-
	USART	USART_SlaveMode	This example describes a USART-SPI communication (transmit/receive) between two boards where the USART is configured as a slave.	-		-	-	-	MX	MX	-
		USART_SlaveMode_DMA	This example describes a USART-SPI communication (transmit/receive) with DMA between two boards where the USART is configured as a slave.	-		-	-	-	MX	MX	-
	WWDG	WWDG_Example	Configuration of the HAL API to periodically update the WWDG counter and simulate a software fault that generates an MCU WWDG reset when a predefined time period has elapsed.	-		-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32U5A9J-DK STM32U5G9J-DK1	STM32U5G9J-DK2	STM32U575I-EV	NUCLEO-U5A5ZJ-Q	NUCLEO-U575ZI-Q	NUCLEO-U545RE-Q	B-U585I-IOT02A
Examples	Total number of examples: 179			11	11	27	4	88	15	23
Examples_LL	ADC	ADC_AnalogWatchdog_Init	How to use an ADC peripheral with an ADC analog watchdog to monitor a channel and detect when the corresponding conversion data is outside the window thresholds.	-	-	-	-	MX	-	-
		ADC_ContinuousConversion_TriggerSW_LowPower_Init	Use ADC to convert a single channel with ADC low power features auto wait and auto power-off.	-	-	-	-	MX	-	-
		ADC_Oversampling_Init	How to use an ADC peripheral with oversampling.	-	-	-	-	MX	-	-
		ADC_SingleConversion_TriggerSW_IT_Init	How to use ADC to convert a single channel at each SW start, conversion performed using the programming model: interrupt.	-	-	-	-	MX	-	-
		ADC_SingleConversion_TriggerSW_Init	How to use ADC to convert a single channel at each SW start, conversion performed using the programming model: polling.	-	-	-	-	MX	-	-
	CRC	CRC_CalculateAndCheck	How to configure the CRC calculation unit to compute a CRC code for a given data buffer, based on a fixed generator polynomial (default value 0x4C11DB7). The peripheral initialization is done using LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	MX	-	-
		CRC_UserDefinedPolynomial	How to configure and use the CRC calculation unit to compute an 8-bit CRC code for a given data buffer, based on a user-defined generating polynomial. The peripheral initialization is done using LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	MX	-	-
	DMA	DMA_LinkedList	How to use the DMA to perform a list of transfers. The transfer list is organized as linked-list, each time the current transfer ends the DMA automatically reload the next transfer parameters, and starts it (without CPU intervention).	-	-	-	-	X	-	-
	EXTI	EXTI_ToggleLedOnIT_Init	This example describes how to configure the EXTI and use GPIOs to toggle the user LEDs available on the board when a user button is pressed. This example is based on the STM32U5xx LL API. Peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	-	-	MX	-	-
	GPIO	GPIO_InfiniteLedToggling_Init	How to configure and use GPIOs to toggle the on-board user LEDs every 250 ms.	-	-	-	-	MX	-	-
	I2C	I2C_OneBoard_Communication_IT_Init	How to handle the reception of one data byte from an I²C slave device by an I²C master device. Both devices operate in interrupt mode. The peripheral is initialized with the LL initialization function to demonstrate LL init usage.	-	-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32U5A9J-DK	STM32U5G9J-DK1	STM32U5G9J-DK2	STM32U575I-EV	NUCLEO-U5A5ZJ-Q	NUCLEO-U575ZI-Q	NUCLEO-U545RE-Q	B-U585I-IOT02A
Examples_LL	IWDG	IWDG_RefreshUntilUserEvent_Init	How to configure the IWDG peripheral to ensure periodical counter update and generate an MCU IWDG reset when a user push-button is pressed. The peripheral is initialized with LL unitary service functions to optimize for performance and size.	-		-	-	-	MX	-	-
	PWR	PWR_EnterStandbyMode	How to enter Standby mode and wake up from this mode by using an external reset or a wake-up pin.	-		-	-	-	MX	-	-
		PWR_EnterStopMode	How to enter STOP 0 mode.	-		-	-	-	MX	-	-
	RCC	RCC_OutputSystemClockOnMCO	Configuration of the MCO pin (PA8) to output the system clock.	-		-	-	-	MX	-	-
		RCC_UseHSI_PLLasSystemClock	Modification of the PLL parameters in runtime .	-		-	-	-	MX	-	-
	RNG	RNG_GenerateRandomNumbers	Configuration of the RNG to generate 32-bit long random numbers. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-		-	-	-	MX	-	-
		RNG_GenerateRandomNumbers_IT	Configuration of the RNG to generate 32-bit long random numbers using interrupts. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-		-	-	-	MX	-	-
	RTC	RTC_Alarm_Init	Configuration of the RTC LL API to configure and generate an alarm using the RTC peripheral. The peripheral initialization uses the LL initialization function.	-		-	-	-	MX	-	-
		RTC_ExitStandbyWithWakeUpTimer_Init	How to periodically enter and wake up from Standby mode thanks to the RTC wake-up timer (WUT).	-		-	-	-	MX	-	-
		RTC_Tamper_Init	Configuration of the tamper using the RTC LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-		-	-	-	MX	-	-
		RTC_TimeStamp_Init	Configuration of the timestamp using the RTC LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-		-	-	-	MX	-	-
	SPI	SPI_OneBoard_HalfDuplex_IT_Init	Configuration of GPIO and SPI peripherals to transmit bytes from an SPI Master device to an SPI Slave device in Interrupt mode. This example is based on the STM32U5xx SPI LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-		-	-	-	MX	-	-
		SPI_TwoBoards_FullDuplex_IT_Master_Init	Data buffer transmission and reception via SPI using Interrupt mode. This example is based on the STM32U5xx SPI LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-		-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32U5A9J-DK	STM32U5G9J-DK1	STM32U5G9J-DK2	STM32U575I-EV	NUCLEO-U5A5ZJ-Q	NUCLEO-U575ZI-Q	NUCLEO-U545RE-Q	B-U585I-IOT02A
Examples_LL	SPI	SPI_TwoBoards_FullDuplex_IT_Slave_Init	Data buffer transmission and reception via SPI using Interrupt mode.	-		-	-	-	MX	-	-
	TIM	TIM_BreakAndDeadtime_Init	Configuration of the TIM peripheral to generate three center-aligned PWM and complementary PWM signals, insert a defined deadtime value, use the break feature, and lock the break and dead-time configuration.	-		-	-	-	MX	-	-
		TIM_InputCapture_Init	Use of the TIM peripheral to measure a periodic signal frequency provided either by an external signal generator or by another timer instance.	-		-	-	-	MX	-	-
		TIM_OnePulse_Init	Configuration of a timer to generate a positive pulse in Output Compare mode with a length of tPULSE and after a delay of tDELAY.	-		-	-	-	MX	-	-
		TIM_OutputCompare_Init	Configuration of the TIM peripheral to generate an output waveform in different output compare modes. This example is based on the STM32U5xx TIM LL API.	-		-	-	-	MX	-	-
		TIM_PWMOutput_Init	Use of a timer peripheral to generate a PWM output signal and update the PWM duty cycle.	-		-	-	-	MX	-	-
		TIM_TimeBase_Init	Configuration of the TIM peripheral to generate a timebase.	-		-	-	-	MX	-	-
	USART	USART_Communication_Rx_IT_Continuous_Init	This example shows how to configure the GPIO and USART peripherals for continuously receiving characters from HyperTerminal (PC) in asynchronous mode using Interrupt mode. Peripheral initialization is done using LL unitary services functions for optimization purposes (performance and size).	-		-	-	-	MX	-	-
		USART_Communication_Rx_IT_Continuous_VCP_Init	This example shows how to configure the GPIO and USART peripherals for continuously receiving characters from HyperTerminal (PC) in asynchronous mode using Interrupt mode. Peripheral initialization is done using LL unitary services functions for optimization purposes (performance and size).	-		-	-	-	MX	-	-
		USART_Communication_Rx_IT_Init	This example shows how to configure the GPIO and USART peripherals for receiving characters from HyperTerminal (PC) in asynchronous mode using Interrupt mode. Peripheral initialization is done using the LL initialization function to demonstrate LL init usage.	-		-	-	-	MX	-	-
		USART_Communication_Rx_IT_VCP_Init	This example shows how to configure the GPIO and USART peripherals for receiving characters from HyperTerminal (PC) in asynchronous mode using Interrupt mode. Peripheral initialization is done using the LL initialization function to demonstrate LL init usage.	-		-	-	-	MX	-	-
		USART_Communication_Tx_IT_Init	This example shows how to configure the GPIO and USART peripherals to send characters asynchronously to HyperTerminal (PC) in Interrupt mode. This example is based on an STM32U5xx USART LL API. Peripheral initialization is done using the LL unitary service functions for optimization purposes (performance and size).	-		-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32U5A9J-DK	STM32U5G9J-DK1	STM32U5G9J-DK2	STM32U575I-EV	NUCLEO-U5A5ZJ-Q	NUCLEO-U575ZI-Q	NUCLEO-U545RE-Q	B-U585I-IOT02A
Examples_LL	USART	USART_Communication_Tx_IT_VCP_Init	This example shows how to configure the GPIO and USART peripherals to send characters asynchronously to HyperTerminal (PC) in Interrupt mode. This example is based on the STM32U5xx USART LL API. Peripheral initialization is done using the LL unitary service functions for optimization purposes (performance and size).	-		-	-	-	MX	-	-
		USART_Communication_Tx_Init	This example shows how to configure the GPIO and USART peripherals to send characters asynchronously to a HyperTerminal (PC) in Polling mode. If the transfer cannot be completed within the allocated time, a timeout allows an exit from the sequence with a timeout error code. This example is based on the STM32U5xx USART LL API. Peripheral initialization is done using the LL unitary service functions for optimization purposes (performance and size).	-		-	-	-	MX	-	-
		USART_Communication_Tx_VCP_Init	This example shows how to configure GPIO and USART peripherals to send characters asynchronously to a HyperTerminal (PC) in Polling mode. If the transfer could not be completed within the allocated time, a timeout allows an exit from the sequence with a timeout error code. This example is based on the STM32U5xx USART LL API. Peripheral initialization is done using LL unitary service functions for optimization purposes (performance and size).	-		-	-	-	MX	-	-
	UTILS	UTILS_ConfigureSystemClock	Use of the UTILS LL API to configure the system clock using PLL with HSI as the source clock.	-		-	-	-	MX	-	-
		UTILS_ReadDeviceInfo	This example reads the UID, device ID, and revision ID and saves them into a global information buffer.	-		-	-	-	MX	-	-
	WWDG	WWDG_RefreshUntilUserEvent_Init	Configuration of the WWDG to periodically update the counter and generate an MCU WWDG reset when a user button is pressed. The peripheral initialization uses the LL unitary service functions for optimization purposes (performance and size).	-		-	-	-	MX	-	-
	Total number of examples_ll: 42			**0**		**0**	**0**	**0**	**42**	**0**	**0**
Examples_MIX	ADC	ADC_SingleConversion_TriggerSW_IT	How to use ADC to convert a single channel at each SW start, conversion performed using the programming model: interrupt.	-		-	-	-	MX	-	-
	CRC	CRC_PolynomialUpdate	How to use the CRC peripheral through the STM32U5xx CRC HAL and LL API.	-		-	-	-	MX	-	-
	SPI	SPI_FullDuplex_ComPolling_Master	Data buffer transmission/reception between two boards via SPI using Polling mode.	-		-	-	-	MX	-	-
		SPI_FullDuplex_ComPolling_Slave	Data buffer transmission/reception between two boards via SPI using Polling mode.	-		-	-	-	MX	-	-
	TIM	TIM_PWMInput	Use of the TIM peripheral to measure an external signal frequency and duty cycle.	-		-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32U5A9J-DK STM32U5G9J-DK1	STM32U5G9J-DK2	STM32U575I-EV	NUCLEO-U5A5ZJ-Q	NUCLEO-U575ZI-Q	NUCLEO-U545RE-Q	B-U585I-IOT02A
Examples_MIX	Total number of examples_mix: 5			0	0	0	0	5	0	0
Applications	-	OpenBootloader	This application exploits the OpenBootloader middleware to demonstrate how to develop an IAP application and how to use it.	X	-	-	-	-	-	X
		SBSFU	The SBSFU provides a Root of Trust solution, including secure boot and secure firmware update functionalities, that is used before executing the application. It provides an example of secure service (GPIO toggle) that is isolated from the non-secure application, but can be used by the nonsecure application at runtime.	-	-	-	-	-	-	X
		TFM	The TFM provides a Root of Trust solution including secure boot and secure firmware update functionalities that is used before executing the application. It provides TFM secure services that are isolated from the nonsecure application but can be used by the nonsecure application at runtime.	-	-	-	-	-	-	X
	BLE	BLE_AT_Client	This example demonstrates Bluetooth® Low Energy connectivity on a STM32WB5MMG module for the B-U585-IOT02A board.	-	-	-	-	-	-	X
	FileX	Fx_Dual_Instance	This application provides a working example of two storage media managed by two independent instances of FileX/LevelX running on an STM32U575I-EV board.	-	-	MX	-	-	-	-
		Fx_File_Edit_Standalone	This application provides an example of FileX stack usage on an STM32U5G9J-DK2 board, running in standalone mode (without ThreadX). It demonstrates how to create a FAT file system on the internal SRAM memory using the FileX API.	-	MX	MX	MX	-	MX	-
		Fx_MultiAccess	This application provides an example of Azure RTOS FileX stack usage on an STM32U575I-EV board. It demonstrates the FileX's concurrent file access capabilities. The application is designed to execute file operations on the SD card device; the code provides all required software code for handling SD card I/O operations.	-	-	MX	-	-	-	-
		Fx_NoR_Write_Read_File	This application provides an example of Azure RTOS FileX and LevelX stacks usage on a STM32U5A9J-DK board. It demonstrates how to create a FAT file system on the NOR flash memory using FileX alongside LevelX. The application is designed to execute file operations on the MX25LM51245G NOR flash memory device; the code provides all required software code for properly managing it.	MX	MX	-	-	-	-	MX
		Fx_uSD_File_Edit	This application provides an example of Azure RTOS FileX stack usage on an STM32U575I-EV board; it shows how to develop a basic SD card file operations application.	-	-	MX	-	-	-	-
	LPBAM	LPBAM_ADC_IntExtChannelSwitch	How to handle an ADC switch between internal and external channel configurations, then convert data using the DMA linked-list feature in low power mode through the LPBAM utility.	-	-	-	-	MX	-	-
		LPBAM_ADC_TempSense	How to handle the ADC temperature sensor monitoring switch between internal and external channel configurations, then convert data using the DMA linked-list feature in low power mode through the LPBAM utility.	-	-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32U5A9J-DK	STM32U5G9J-DK1	STM32U5G9J-DK2	STM32U575I-EV	NUCLEO-U5A5ZJ-Q	NUCLEO-U575ZI-Q	NUCLEO-U545RE-Q	B-U585I-IOT02A
Applications	LPBAM	LPBAM_COMP_InSwitch	How to handle COMP switch inputs and read compared values using the DMA linked-list feature in low power mode through the LPBAM utility.	-		-	-	-	MX	-	-
		LPBAM_DAC_OPAMP_SigAmpli	How to handle DAC continuous conversion and OPAMP switching configuration using the DMA linked-list feature in low power mode through the LPBAM utility.	-		-	-	-	MX	-	-
		LPBAM_DMA_MultiQExec	How to handle multiqueue execution using the DMA linked-list feature in low power mode through the LPBAM utility.	-		-	-	-	MX	-	-
		LPBAM_I2C_TransmitReceive	How to handle I²C sequential transmission/reception with data reload between two boards using the DMA linked-list feature in low power mode through the LPBAM utility.	-		-	-	-	MX	-	-
		LPBAM_LPGPIO_IOToggle	How to toggle a LPGPIO pin with the DMA linked-list feature in low power mode through the LPBAM utility every 1s.	-		-	-	-	MX	-	-
		LPBAM_LPTIM_PWMGen	How to handle LPTIM PWM generation with the DMA linked-list feature in low power mode through the LPBAM utility.	-		-	-	-	MX	-	-
		LPBAM_LPUART_TransmitReceive	How to handle LPUART transmission/reception between two boards with the DMA linked-list feature in low power mode through the LPBAM utility.	-		-	-	-	MX	-	-
		LPBAM_SPI_TransmitReceive	How to handle SPI two consecutive communications (full duplex, then simplex) between two boards with the DMA linked-list feature in low power mode through the LPBAM utility.	-		-	-	-	MX	-	-
	NetXDuo	Nx_HTTP_SSE	This application provides an example of Azure RTOS NetX/NetXDuo stack usage.	-		-	-	-	-	-	MX
		Nx_IAP_Client	This application provides an example of Azure RTOS NetX/NetXDuo stack usage.	-		-	-	-	-	-	MX
		Nx_MDNS	This application demonstrates mDNS on a MXCHIP EMW3080 module for the STM32U585AII6Q board.	-		-	-	-	-	-	MX
		Nx_MQTT_Client	This application provides an example of Azure RTOS NetX/NetXDuo stack usage.	-		-	-	-	-	-	MX
		Nx_Network_Basics	This application demonstrates WiFi connectivity on an MXCHIP EMW3080 module for the STM32U585AII6Q board.	-		-	-	-	-	-	MX
		Nx_SNTP_Client	This application provides an example of Azure RTOS NetX/NetXDuo stack usage.	-		-	-	-	-	-	MX

Level	Module name	Project name	Description	STM32U5A9J-DK	STM32U5G9J-DK1	STM32U5G9J-DK2	STM32U575I-EV	NUCLEO-U5A5ZJ-Q	NUCLEO-U575ZI-Q	NUCLEO-U545RE-Q	B-U585I-IOT02A
Applications	NetXDuo	Nx_TCP_Echo_Client	This application provides an example of Azure RTOS NetX/NetXDuo stack usage.	-		-	-	-	-	-	MX
		Nx_TCP_Echo_Server	This application provides an example of Azure RTOS NetX/NetXDuo stack usage.	-		-	-	-	-	-	MX
		Nx_UDP_Echo_Client	This application provides an example of Azure RTOS NetX/NetXDuo stack usage.	-		-	-	-	-	-	MX
		Nx_UDP_Echo_Server	This application provides an example of Azure RTOS NetX/NetXDuo stack usage.	-		-	-	-	-	-	MX
		Nx_WebServer	This application provides an example of Azure RTOS NetX/NetXDuo stack usage.	-		-	-	-	-	-	MX
	ThreadX	Tx_CMSIS_Wrapper	This application provides an example of the CMSIS RTOS adaptation layer for Azure RTOS ThreadX; it shows how to develop an application using the CMSIS RTOS 2 APIs.	-		-	-	-	-	-	X
		Tx_FreeRTOS_Wrapper	This application provides an example of Azure RTOS ThreadX stack usage; it shows how to develop an application using the FreeRTOS™ adaptation layer for ThreadX.	-		-	-	-	X	-	-
		Tx_LowPower	This application provides an example of Azure RTOS ThreadX stack usage; it shows how to develop an application using the ThreadX in LowPower mode.	-		MX	-	MX	MX	MX	-
		Tx_MPU	This application provides an example of Azure RTOS ThreadX stack usage; it shows how to develop an application using the ThreadX module feature.	X		X	-	-	X	X	-
		Tx_SecureLEDToggle_TrustZone	This application provides an example of Azure RTOS ThreadX stack usage; it shows how to develop an application using the ThreadX when the TrustZone® feature is enabled (TZEN=1).	-		MX	-	MX	MX	MX	-
		Tx_Thread_Creation	This application provides an example of Azure RTOS ThreadX stack usage; it shows how to develop an application using the ThreadX thread management APIs.	-		MX	-	MX	MX	MX	-
		Tx_Thread_MsgQueue	This application provides an example of Azure RTOS ThreadX stack usage; it shows how to develop an application using the ThreadX message queue APIs.	-		-	-	-	MX	-	-
		Tx_Thread_Sync	This application provides an example of Azure RTOS ThreadX stack usage; it shows how to develop an application using the ThreadX synchronization APIs.	-		-	-	-	MX	-	-
	USBPD	USBPD_SNK_UX_Device_HID_ CDC_ACM	This application is a USB Power Delivery type C consumer and USB device using the Azure RTOS USBX stack on a STM32U585xx board.	-		-	-	-	-	-	MX

Level	Module name	Project name	Description	STM32U5A9J-DK	STM32U5G9J-DK1	STM32U5G9J-DK2	STM32U575I-EV	NUCLEO-U5A5ZJ-Q	NUCLEO-U575ZI-Q	NUCLEO-U545RE-Q	B-U585I-IOT02A
Applications	USBPD	USBPD_SRC_UX_Host_MSC	This application is a USB Power Delivery type C provider and USB host using the Azure RTOS USBX stack. It shows how to develop a USB Power Delivery type C provider in the case of a USB host application based on Mass storage "MSC", which is able to enumerate and communicates with a removable USB flash memory disk.	-		-	MX	-	-	-	-
	USBX	Ux_Device_CDC_ACM	This application provides an example of Azure RTOS USBX stack usage on an STM32U5G9J-DK2 board; it shows how to develop a USB device communication class "CDC_ACM" based application.	-		MX	-	MX	MX	MX	-
		Ux_Device_CDC_ECM	This application provides an example of Azure RTOS CDC_EM stack usage on a STM32U575I-EV board; it shows how to run an HTTP web server based application stack via a USB interface. The application is designed to load files and web pages stored on an SD card using an HTTP web server through a USB interface, using the CDC_ECM class. The code provides all required features to build a compliant HTTP web server. The main entry function tx_application_define() is called by ThreadX during kernel start. At this stage, the USBX initializes the network layer through the USBx class (CDC_ECM); the FileX and the NetXDuo system are also initialized. The NX_IP instance and the HTTP web server are created and configured; then the application creates two main threads: app_ux_device_thread_entry (Prio: 10; PreemptionPrio: 10) used to initialize the USB OTG HAL PCD driver and start the device.	-		-	-	MX	-	-	-
		Ux_Device_DFU	This application provides an example of Azure RTOS USBX stack usage on an STM32U5G5xx board; it shows how to develop a USB device firmware upgrade "DFU" based application.	-		MX	-	MX	-	-	MX
		Ux_Device_HID	This application provides an example of Azure RTOS USBX stack usage on an STM32U585xx board; it shows how to develop a USB device human interface "HID" mouse based application.	-		-	-	-	-	-	MX
		Ux_Device_HID_CDC_ACM	This application provides an example of Azure RTOS USBX stack usage on an STM32U585xx board; it shows how to develop a composite USB device communication class "HID" and "CDC_ACM" based application.	-		-	-	-	-	-	MX
		Ux_Device_MSC	This application provides an example of Azure RTOS USBX stack usage on an STM32U575I-Eval board; it shows how to develop a USB device mass storage class based application.	-		-	MX	-	-	-	-
		Ux_Host_CDC_ACM	This application provides an example of Azure RTOS USBX stack usage.	MX		MX	MX	-	-	-	-
		Ux_Host_DualClass	This application provides an example of Azure RTOS USBX stack usage.	-		-	MX	-	-	-	-
		Ux_Host_HID	This application provides an example of Azure RTOS USBX stack usage.	MX		MX	MX	-	-	MX	-

Level	Module name	Project name	Description	STM32U5A9J-DK	STM32U5G9J-DK1	STM32U5G9J-DK2	STM32U575I-EV	NUCLEO-U5A5ZJ-Q	NUCLEO-U575ZI-Q	NUCLEO-U545RE-Q	B-U585I-IOT02A
Applications	USBX	Ux_Host_HID_CDC_ACM	This application provides an example of Azure RTOS USBX stack usage.	-		-	MX	-	-	-	-
		Ux_Host_HUB_HID_MSC	This application provides an example of Azure RTOS USBX stack usage.	MX		MX	-	-	-	-	-
		Ux_Host_MSC	This application provides an example of Azure RTOS USBX stack usage. It shows how to develop USB host Mass storage "MSC" able to enumerate and communicate with a removable USB flash memory disk.	-		-	MX	-	-	-	-
	Total number of applications: 81			6		11	12	6	18	7	21
Demonstrations	-	Demo	The STM32Cube demonstration platform comes on top of the STM32Cube as a firmware package that offers a full set of software components based on a modular architecture. All modules can be reused separately in standalone applications. All these modules are managed by the STM32Cube demonstration kernel, that allows for dynamically adding new modules and access common resources (storage, memory management, real-time operating system). The STM32Cube demonstration platform is built around a basic GUI interface. It is based on the STM32Cube HAL BSP and several middleware components.	-		-	X	-	-	-	-
		IOT_HTTP_WebServer	The STM32Cube demonstration platform comes on top of the STM32Cube as a firmware package. It is based on the STM32Cube HAL, BSP, and middleware components. It shows how to perform a web server demonstration using the MXCHIP wifi module.	-		-	-	-	-	-	X
	Total number of demonstrations: 2			0		0	1	0	0	0	1
Total number of projects: 330				20		25	43	13	156	25	48

Revision history

Table 2. Document revision history

Date	Version	Changes
22-Sep-2021	1	Initial release
15-Feb-2023	2	Updated Table 1. STM32CubeU5 firmware examples with examples supporting STM32U5A9xx, STM32U5A5xx and STM32U545xx devices
07-Jul-2023	3	• Replaced the image in Figure 1. STM32CubeU5 firmware components • Updated Table 1. STM32CubeU5 firmware examples with examples supporting STM32U5GXxx devices
19-Mar-2024	4	Updated Table 1. STM32CubeU5 firmware examples with one new project name in "Examples_LL" for ADC on NUCLEO-U575ZI-Q board

Contents

List of tables

IMPORTANT NOTICE – READ CAREFULLY