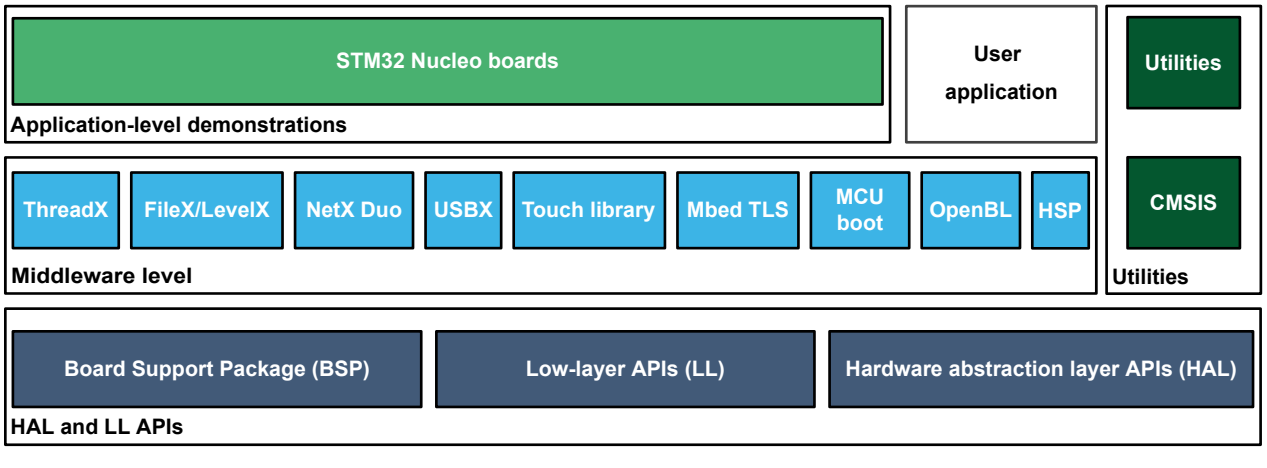
STM32Cube MCU package examples for STM32U3 series

Introduction

The STM32CubeU3 MCU Package comes with a rich set of examples running on STMicroelectronics boards. The examples are organized by board, and are provided with preconfigured projects for the main supported toolchains (see figure below).

Figure 1. STM32CubeU3 firmware components



1 STM32CubeU3 examples

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

- **Examples**
  The examples use only the HAL and BSP drivers (middleware components are not used). Their objective is to demonstrate the product/peripherals features and usage. They are organized per peripheral (one folder per peripheral, for example, TIM). Their complexity level ranges from the basic usage of a given peripheral (for example, PWM generation using timer) to the integration of several peripherals (for example, how to use DAC for signal generation with synchronization from TIM6 and DMA). The usage of the board resources is reduced to the strict minimum.
- **Examples_LL**
  These examples only use the LL drivers (HAL drivers and middleware components are not used). They offer an optimum implementation of typical use cases of the peripheral features and configuration sequences. The LL examples are organized per peripheral (one folder for each peripheral, for example, TIM) and run exclusively on the Nucleo board.
- **Examples_MIX**
  These examples only use 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 to combine the advantages of both APIs:
  - HAL drivers offer high-level function-oriented APIs, which have a high level of portability since they hide product/IP complexity to end users.
  - LL drivers offer low-level APIs at register level with better optimization.

  The examples are organized per peripheral (one folder for each peripheral, for example, TIM) and run exclusively on the Nucleo board.
- **Applications**
  The applications demonstrate the product performance and how to use the available middleware stacks. They are organized either by middleware (one folder per middleware, for example USB host) or by product feature that requires high-level firmware bricks (for example, audio). The integration of applications that use several middleware stacks is also supported.

The examples are located under `STM32Cube_FW_U3_VX.Y.Z\Projects\`. They all have the same structure:

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

To run the example, proceed as follows:

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

Note: Refer to sections “Development Toolchains and Compilers” and “Supported Devices and EVAL, Nucleo, and Discovery boards” of the firmware package release notes to know about the software/hardware environment used for the firmware development and validation. The correct operation of the provided examples is not guaranteed in some environments, for example when using different compiler or board versions.

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

Table 1 contains the list of examples provided with the STM32U3 MCU Package.

Note: STM32CubeMX-generated examples are highlighted with the MX STM32CubeMX icon. TrustZone indicates that the example is Arm® TrustZone® enabled.

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

The Arm word and logo are trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved.

Table 1. STM32CubeU3 firmware examples

STM32CubeMX-generated examples are highlighted with the icon MX.

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW(1)	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Templates	-	Starter project	This project provides a reference template that can be used to build any firmware application.	2	-	-	X	X	-	-	-	-	-
		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).	5	-	-	-	MX	MX	MX	MX	MX	MX
		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).	5	-	-	-	MX	MX	MX	MX	MX	MX
		TrustZoneEnabled_NoIsolation	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) and all MCU resources (memories, peripherals...) are configured as secure.	1	-	-	-	-	-	-	X	-	-
	ROT	OEMiROT_Appli	This project provides a OEMiROT boot path Full Secure reference template. Boot is performed through OEMiROT boot path after authenticity and the integrity checks of the project firmware and data images.	5	-	-	-	X	X	X	X	X	X
		OEMiROT_Appli_TrustZone	This project provides a OEMiROT boot path reference template. Boot is performed through OEMiROT boot path after authenticity and integrity checks of the project firmware and project data images.	4	-	-	-	X	X	X	X	X	-
	Total number of templates			**22**	**6**	**0**	**1**	**1**	**4**	**4**	**5**	**4**	**3**
Templates_LL	-	TrustZoneDisabled	This project provides a reference template through the LL API that can be used to build any firmware application.	5	-	-	-	MX	MX	MX	MX	MX	MX
	Total number of templates_ll			**5**	**1**	**0**	0	**1**	**1**	1	**1**	**1**	**1**
Examples	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.	1	-	-	-	-	-	-	MX	-	-
		ADC_ContinuousConversion_TriggerSW_ LowPower	Use ADC to convert a single channel with ADC low power features auto wait and auto power-off.	2	-	-	-	-	MX	-	MX	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples	ADC	ADC_DiscontinuousConversion_TriggerSW	This example describes how to use an ADC peripheral to perform multiple conversion from different ADC channel, one at a time after each software trigger.	1	-	-	-	-	-	-	MX	-	-
		ADC_FixedTriggerLatency	How to use an ADC peripheral to perform a single ADC conversion on a channel at each trigger event from a timer without any uncertainty (fixed trigger latency).	1	-	-	-	-	-	-	MX	-	-
		ADC_SingleConversion_TriggerSW_IT	How to use ADC to convert a single channel at each SW start, conversion performed using programming model: interrupt.	2	-	-	-	-	MX	-	MX	-	-
		ADC_SingleConversion_TriggerTimer_DMA	How to use an ADC peripheral to perform a single ADC conversion on a channel at each trigger event from a timer. Converted data is transferred by DMA into a table in RAM memory.	2	-	-	-	-	MX	-	MX	-	-
		ADC_TemperatureSensor	How to use an ADC peripheral to perform a single ADC conversion on the internal temperature sensor and calculate the temperature in degrees Celsius.	1	-	-	-	-	-	-	MX	-	-
	ADF	ADF_AudioRecorder	This project demonstrates how to configure ADF to perform PCM record in DMA circular mode.	1	-	-	-	-	-	-	MX	-	-
		ADF_AudioRecorder_LowPower	This project demonstrates how to configure ADF to perform PCM record in low power Stop 1 mode.	1	-	-	-	-	-	-	MX	-	-
		ADF_AudioSoundDetector	This project demonstrates how to configure ADF to perform audio sound detection.	1	-	-	-	-	-	-	MX	-	-
		ADF_AudioSoundDetector_LowPower	This project demonstrates how to configure ADF to perform audio sound detection in low power Stop 1 mode.	1	-	-	-	-	-	-	MX	-	-
	CCB	CCB_Protected_ECCScalarMul_BlobCreation	How to use the CCB peripheral to create a blob for ECC scalar multiplication.	1	-	-	-	-	-	-	MX	-	-
		CCB_Protected_ECCScalarMul_BlobUse	How to use the CCB peripheral to compute scalar multiplication k x P, using a special blob called "ECC key blob".	1	-	-	-	-	-	-	MX	-	-
		CCB_Protected_ECDSA_BlobCreation	How to use the CCB peripheral to create a blob for Elliptic curve digital signature algorithm (ECDSA) usage.	1	-	-	-	-	-	-	MX	-	-
		CCB_Protected_ECDSA_PublicKeyComputation	How to use the CCB peripheral to compute public Key, using a special blob called "ECDSA key blob".	1	-	-	-	-	-	-	MX	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples	CCB	CCB_Protected_ECDSA_Signature	How to use the CCB peripheral to compute a signed message regarding the Elliptic curve digital signature algorithm using a special blob called "ECDSA key blob".	1	-	-	-	-	-	-	X	-	-
		CCB_Protected_RSAModularExp_BlobCreation	How to use the CCB peripheral to create a blob for RSA modular exponentiation.	1	-	-	-	-	-	-	MX	-	-
		CCB_Protected_RSAModularExp_BlobUse	How to use the CCB peripheral to create and use an RSA modular exponentiation blob.	1	-	-	-	-	-	-	MX	-	-
	COMP	COMP_CompareGpioVsVrefInt_IT	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.	1	-	-	-	-	-	-	MX	-	-
		COMP_CompareGpioVsVrefInt_IT_ SystemStopMode	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 and during stop2 mode.	1	-	-	-	-	-	-	MX	-	-
		COMP_CompareGpioVsVrefInt_OutputGpio	How to use a comparator peripheral to compare a voltage level applied on a GPIO pin to the internal voltage reference (VrefInt) with comparator output connected to a GPIO pin.	1	-	-	-	-	-	-	MX	-	-
		COMP_CompareGpioVsVrefInt_Window_IT	How to use a pair of comparator peripherals to compare a voltage level applied on a GPIO pin to two thresholds: the internal voltage reference (VrefInt) and a fraction of the internal voltage reference (VrefInt/2), in interrupt mode.	1	-	-	-	-	-	-	MX	-	-
	CORTEX	CORTEXM_MPU	Presentation of the MPU features. This example configures MPU attributes of different MPU regions then configures a memory area as privileged read-only, and attempts to perform read and write operations in different modes.	1	-	-	-	-	-	-	MX	-	-
		CORTEXM_ModePrivilege	How to modify the Thread mode privilege access and stack. Thread mode is entered on reset or when returning from an exception.	1	-	-	-	-	-	-	MX	-	-
		CORTEXM_ProcessStack	How to modify the Thread mode stack. Thread mode is entered on reset, and can be entered as a result of an exception return.	1	-	-	-	-	-	-	MX	-	-
		CORTEXM_SysTick	How to use the default SysTick configuration with a 1 ms timebase to toggle LEDs.	5	-	-	-	-	MX	MX	MX	MX	MX

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
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). The user-defined generating polynomial is manually set to 0x65, that is, X^7 + X^6 + X^5 + X^2 + 1, as used in the Train Communication Network, IEC 60870-5[17].	1	-	-	-	-	-	-	MX	-	-
		CRC_Data_Reversing_16bit_CRC	How to configure the CRC using the HAL API. The CRC (cyclic redundancy check) calculation unit computes a 16-bit CRC code derived from a buffer of 32-bit data (words). Input and output data reversal features are enabled. The user-defined generating polynomial is manually set to 0x1021, that is, X^16 + X^12 + X^5 + 1 which is the CRC-CCITT generating polynomial.	1	-	-	-	-	-	-	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).	1	-	-	-	-	-	-	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.	1	-	-	-	-	-	-	MX	-	-
	CRYP	CRYP_AES_GCM	How to use the CRYPTO peripheral to encrypt and decrypt data using AES with Galois/Counter mode (GCM).	1	-	-	-	-	-	-	MX	-	-
		CRYP_DMA	How to use the AES peripheral to encrypt and decrypt data using AES 128 Algorithm with ECB chaining mode in DMA mode.	1	-	-	-	-	-	-	MX	-	-
		CRYP_GCM_GMAC_CCM_Modes	How to use the CRYP peripheral to encrypt data and generate authentication tags using GCM/ GMAC/CCM modes.	1	-	-	-	-	-	-	MX	-	-
		CRYP_GCM_GMAC_CMAC_Suspension	How to use the CRYP AES peripheral to suspend then resume the AES GCM and GMAC CMAC processing of a message in order to carry out the encryption, decryption or authentication tag computation of a higher-priority message(CMAC).	1	-	-	-	-	-	-	MX	-	-
		CRYP_SAES_ECB_CBC	How to use the Secure AES co-processor (SAES) peripheral to encrypt and decrypt data using AES ECB and CBC algorithms.	1	-	-	-	-	-	-	MX	-	-
		CRYP_SAES_SharedKey	How to use the Secure AES co-processor (SAES) peripheral to share application keys with AES peripheral.	1	-	-	-	-	-	-	MX	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples	CRYP	CRYP_SAES_WrapKey	How to use the Secure AES co-processor (SAES) peripheral to wrap application keys using hardware secret key DHUK then use it to encrypt in polling mode.	1	-	-	-	-	-	-	MX	-	-
	DAC	DAC_ContinuousConversionDMA_TriggerEXTI	How to use the DAC to generate a signal using the DMA and triggered by EXTI signal.	1	-	-	-	-	-	-	MX	-	-
		DAC_GenerateConstantSignal_SystemStopMode	How to use the DAC peripheral to do a simple conversion with the MCU in STOP1 mode.	1	-	-	-	-	-	-	MX	-	-
		DAC_SignalsGeneration	How to use the DAC peripheral to generate several signals using the DMA controller and the DAC internal wave generator.	1	-	-	-	-	-	-	MX	-	-
		DAC_SimpleConversion	How to use the DAC peripheral to do a simple conversion.	1	-	-	-	-	-	-	MX	-	-
	DMA	DMA_DataHandling	This example describes 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.	1	-	-	-	-	-	-	MX	-	-
		DMA_FLASHToRAM	This example describes how to use DMA to transfer a word data buffer from Flash memory to embedded SRAM through the HAL API.	2	-	-	-	-	MX	-	MX	-	-
		DMA_LinkedList	This example describes 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).	2	-	-	-	-	MX	-	MX	-	-
		DMA_RepeatedBlock	How to configure and use the DMA HAL API to perform repeated block transactions.	1	-	-	-	-	-	-	MX	-	-
		DMA_Trigger	How to configure and use the DMA HAL API to perform DMA triggered transactions.	1	-	-	-	-	-	-	MX	-	-
	DMA2D	DMA2D_MemToMemWithBlending	This example demonstrates DMA2D Pixel Format Conversion (PFC) and DMA2D Memory-to-Memory blending on STM32U3A6H-DK, then displays the RGB565 result on the external ST77916 TFT through the BSP LCD2 interface (FMC bus path).	1	-	-	MX	-	-	-	-	-	-
		DMA2D_MemoryToMemory	This example provides a description of how to configure the DMA2D peripheral in Memory_to_Memory transfer mode.	1	-	-	-	-	-	MX	-	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples	FDCAN	FDCAN_Adaptive_Bitrate_Receiver	This example describes how to configure the FDCAN peripheral to adapt to different CAN bit rates using restricted mode.	1	-	-	-	-	-	-	MX	-	-
		FDCAN_Adaptive_Bitrate_Transmitter	This example describes how to configure the FDCAN peripheral to adapt to different CAN bit rates using restricted mode.	1	-	-	-	-	-	-	MX	-	-
		FDCAN_Classic_Frame_Networking	This example describes how to configure the FDCAN peripheral to send and receive Classic CAN frames between two FDCAN units.	1	-	-	-	-	-	-	MX	-	-
		FDCAN_Com_IT	This example describes how to configure the FDCAN peripheral to achieve interrupt process communication between two FDCAN units.	1	-	-	-	-	-	-	MX	-	-
		FDCAN_Com_Polling	This example describes how to configure the FDCAN peripheral to achieve polling process communication between two FDCAN units.	1	-	-	-	-	-	-	MX	-	-
		FDCAN_Loopback	This example describes how to configure the FDCAN peripheral to operate in loopback mode.	1	-	-	-	-	-	-	MX	-	-
		FDCAN_Power_Down	This example describes the FDCAN operation in power-down mode.	1	-	-	-	-	-	-	MX	-	-
	FLASH	FLASH_ChangeOptionBytes	How to configure and use the FLASH HAL API to change option bytes values.	1	-	-	-	-	-	-	MX	-	-
		FLASH_EraseProgram	How to configure and use the FLASH HAL API to erase and program the internal At the beginning of the main program the HAL_Init() function is called to reset all the peripherals, initialize the Flash interface and the systick.	5	-	-	-	-	MX	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)**.	2	-	-	-	-	MX	-	MX	-	-
		FLASH_HideProtection_TrustZone	How to configure and use the FLASH HAL API for the secure hide protection and extended secure hide protection of internal Flash memory.	2	-	-	-	-	-	X	X	-	-
		FLASH_SwapBanks	Guide through the configuration steps to program internal Flash memory bank 1 and bank 2, and to swap between both of them by mean of the FLASH HAL API.	2	-	-	-	-	X	-	X	-	-
	FMC	FMC_LCD	This example demonstrates how to drive the ST77916 external TFT LCD (360x390 pixels, RGB565) mounted on the STM32U3A6H-DK board through the FMC asynchronous SRAM memory interface.	1	-	-	MX	-	-	-	-	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples	GFXPAND	GFXPAND_DMA	This example provides a description of how to use the GFXPAND peripheral in DMA mode to decode a 128x128 image encoded in QOI format into an ARGB8888 output buffer on STM32U3A6ZH devices.	1	-	-	-	-	-	MX	-	-	-
	GPIO	GPIO_EXTI	How to configure external interrupt lines.	5	-	-	-	-	MX	MX	MX	MX	MX
		GPIO_IOToggle	How to configure and use GPIOs through the HAL API.	6	-	-	-	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).	1	-	-	-	-	-	-	MX	-	-
	GTZC	GTZC_TZSC_MPCBB_TrustZone	How to use HAL GTZC MPCBB to build any example with illegal Access detection when TrustZone security is activated **(Option byte TZEN=0xB4)**.	1	-	-	-	-	-	-	MX	-	-
	HAL	HAL_TimeBase_RTC_WKUP	How to customize HAL using RTC wakeup as main source of time base, instead of Systick.	1	-	-	-	-	-	-	MX	-	-
		HAL_TimeBase_TIM	How to customize HAL using a general-purpose timer as main source of time base instead of Systick.	1	-	-	-	-	-	-	MX	-	-
	HASH	HASH_HMAC_SHA224SHA1_DMA_Suspension	This example describes how to suspend the HMAC digest computation when data are fed to the HASH IP by DMA.	1	-	-	-	-	-	-	MX	-	-
		HASH_SHA256	This example provides a short description of how to use the HASH peripheral to hash data using SHA_256 Algorithm.	1	-	-	-	-	-	-	MX	-	-
		HASH_SHA384	This example provides a short description of how to use the HASH peripheral to hash data using SHA_384 Algorithm.	1	-	-	-	-	-	-	MX	-	-
		HASH_SHA512	This example provides a short description of how to use the HASH peripheral to hash data using SHA512 Algorithms.	1	-	-	-	-	-	-	MX	-	-
	I2C	I2C_Sensor_Private_Command_IT	How to handle I2C data buffer transmission/ reception between STM32U3xx Nucleo and X-NUCLEO-IKS4A1, using an interrupt.	2	-	-	-	-	MX	-	MX	-	-
		I2C_TwoBoards_AdvComIT	How to handle several I2C data buffer transmission/reception between a master and a slave device, using an interrupt.	1	-	-	-	-	-	-	MX	-	-
		I2C_TwoBoards_ComDMA	How to handle an I2C to perform a data buffer transmission/reception between two boards, using DMA.	1	-	-	-	-	-	-	MX	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples	I2C	I2C_TwoBoards_ComDMA_Autonomous_LowPower_Master	How to handle an I2C3 as a Master to perform a data buffer transmission/reception in autonomous mode between two boards, using DMA while entering Stop 2 low-power mode.	1	-	-	-	-	-	-	MX	-	-
		I2C_TwoBoards_ComDMA_Autonomous_LowPower_Slave	How to handle an I2C3 as a Slave to perform a data buffer transmission/reception in autonomous mode between two boards, using DMA while entering Stop 2 low-power mode.	1	-	-	-	-	-	-	MX	-	-
		I2C_TwoBoards_ComDMA_Autonomous_Master	How to handle an I2C as a Master to perform a data buffer transmission/reception in autonomous mode between two boards, using DMA.	3	-	-	-	-	-	-	MX	MX	MX
		I2C_TwoBoards_ComDMA_Autonomous_Slave	How to handle an I2C as a Slave to perform a data buffer transmission/reception in autonomous mode between two boards, using DMA.	3	-	-	-	-	-	-	MX	MX	MX
		I2C_TwoBoards_ComIT	How to handle I2C data buffer transmission/ reception between two boards, using an interrupt.	1	-	-	-	-	-	-	MX	-	-
		I2C_TwoBoards_MultiMasterIT_Master	How to handle I2C data buffer communication between two boards, using an interrupt and two Masters and one Slave.	2	-	-	-	-	MX	-	MX	-	-
		I2C_TwoBoards_MultiMasterIT_Slave	How to handle I2C data buffer communication between two boards, using an interrupt and two Masters and one Slave.	2	-	-	-	-	MX	-	MX	-	-
		I2C_TwoBoards_RestartAdvComIT	How to perform multiple I2C data buffer transmission/reception between two boards, in interrupt mode and with restart condition.	1	-	-	-	-	-	-	MX	-	-
		I2C_TwoBoards_RestartComIT	How to handle single I2C data buffer transmission/reception between two boards, in interrupt mode and with restart condition.	1	-	-	-	-	-	-	MX	-	-
		I2C_WakeUpFromStop	How to handle I2C data buffer transmission/ reception between two boards, using an interrupt when the device is in Stop1 mode.	1	-	-	-	-	-	-	MX	-	-
	I3C	I3C_Controller_Direct_Command_DMA	How to handle a Direct Command procedure between an I3C Controller and an I3C Target, using DMA.	1	-	-	-	-	-	-	MX	-	-
		I3C_Controller_ENTDAA_IT	How to handle an ENTDAA procedure between an I3C Controller and one or more I3C Targets.	1	-	-	-	-	-	-	MX	-	-
		I3C_Controller_HotJoin_IT	How to handle an I3C as Target to perform a HOTJOIN procedure between an I3C Controller and I3C Targets, using interrupt.	1	-	-	-	-	-	-	MX	-	-
		I3C_Controller_I2C_ComDMA	How to handle I2C communication as I3C Controller data buffer transmission/reception between two boards, using DMA.	1	-	-	-	-	-	-	MX	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples	I3C	I3C_Controller_IBI_Wakeup_IT	How to handle an I3C as Controller to perform an In-Band-Interrupt event between an I3C Controller in Low Power Mode and I3C Targets.	1	-	-	-	-	-	-	MX	-	-
		I3C_Controller_InBandInterrupt_IT	How to handle an I3C as a Controller to perform an In-Band-Interrupt event between an I3C Controller and I3C Targets.	1	-	-	-	-	-	-	MX	-	-
		I3C_Controller_Private_Autonomous	How to handle the controller to transmit private message in autonomous mode - Board: NUCLEO-U385RG (embeds a STM32U385RG device) - SCL Pin: PB13 (Morpho CN10 pin 30).	1	-	-	-	-	-	-	MX	-	-
		I3C_Controller_Private_Command_IT	How to handle an I3C as a Controller to perform data buffer transmission/reception between two boards, using an interrupt.	1	-	-	-	-	-	-	MX	-	-
		I3C_Controller_ResetPattern_RSTACT	How to handle an I3C as Controller to perform the configuration of a chosen reset pattern after a Direct RSTACT Command procedure between an I3C Controller and an I3C Target, using interrupt.	1	-	-	-	-	-	-	MX	-	-
		I3C_Controller_ResetPattern_ WakeUpFromLowPower	How to handle I3C as Controller to send reset pattern, when Target is in low power mode.	3	-	-	-	-	MX	-	MX	-	MX
		I3C_Controller_Switch_To_Target	How to handle an I3C as a Controller to perform a Controller Role Request Direct Command procedure between an I3C Controller and an I3C Target, using Interrupt.	1	-	-	-	-	-	-	MX	-	-
		I3C_Controller_WakeUpFromStop	How to handle an I3C as a Controller to perform data buffer transmission/reception between two boards, using an interrupt, when the Target is in Stop mode.	3	-	-	-	-	-	MX	MX	-	MX
		I3C_Sensor_Direct_Command_DMA	How to handle a Direct Command procedure between STM32U3xx Nucleo and X-NUCLEO-IKS4A1, using DMA.	1	-	-	-	-	-	-	MX	-	-
		I3C_Sensor_Private_Command_IT	How to handle I3C as Controller data buffer transmission/reception between STM32U3xx Nucleo and X-NUCLEO-IKS4A1, using interrupt.	1	-	-	-	-	-	-	MX	-	-
		I3C_Target_Direct_Command_DMA	How to handle a Direct Command procedure between an I3C Controller and an I3C Target, using Controller in DMA.	1	-	-	-	-	-	-	MX	-	-
		I3C_Target_ENTDAA_IT	How to handle an ENTDAA procedure between an I3C Controller and one or more I3C Targets.	1	-	-	-	-	-	-	MX	-	-
		I3C_Target_HotJoin_IT	How to handle an I3C as Controller to perform a HOTJOIN procedure between an I3C Controller and I3C Targets, using interrupt.	1	-	-	-	-	-	-	MX	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples	I3C	I3C_Target_I2C_ComDMA	How to handle I2C data buffer transmission/ reception between two boards, using DMA.	1	-	-	-	-	-	-	MX	-	-
		I3C_Target_IBI_Wakeup_IT	How to handle an I3C as Target to perform an In Band Interrupt event between an I3C Controller in Stop Mode and an I3C Target.	1	-	-	-	-	-	-	MX	-	-
		I3C_Target_InBandInterrupt_IT	How to handle an I3C as a Target to perform an In-Band-Interrupt event between an I3C Controller and I3C Targets.	1	-	-	-	-	-	-	MX	-	-
		I3C_Target_Private_Autonomous	How to handle I3C as Target data buffer transmission/reception between two boards, using autonomous mode.	1	-	-	-	-	-	-	MX	-	-
		I3C_Target_Private_Command_IT	How to handle an I3C as a Target to perform data buffer transmission/reception between two boards, using an interrupt.	1	-	-	-	-	-	-	MX	-	-
		I3C_Target_ResetPattern_RSTACT	How to handle an I3C as Target to perform the configuration of a chosen reset pattern after a Direct RSTACT Command procedure between an I3C Controller and an I3C Target, using interrupt.	1	-	-	-	-	-	-	MX	-	-
		I3C_Target_ResetPattern_WakeUpFromStandBy	How to handle I3C as Target to wake up from standby, when the Target receives a reset pattern.	3	-	-	-	-	MX	-	MX	-	MX
		I3C_Target_ResetPattern_WakeUpFromStop	How to handle I3C as Target to wake up from Stop 3, when Target receives a reset pattern.	2	-	-	-	-	MX	-	MX	-	-
		I3C_Target_Switch_To_Controller	How to handle an I3C as a Target to perform a Controller Role Request procedure to an I3C Controller.	1	-	-	-	-	-	-	MX	-	-
		I3C_Target_WakeUpFromStop	How to handle an I3C as a Target to perform data buffer transmission/reception between two boards, using an interrupt, when the Target is in Stop 1 mode.	2	-	-	-	-	-	MX	MX	-	MX
	ICACHE	ICACHE_Memory_Remap	- How to execute code from a remapped region configured through the ICACHE HAL driver.	1	-	MX	-	-	-	-	-	-	-
	IWDG	IWDG_WakeUpFromStop2	How to use the IWDG Early Wakeup Interrupt to wake up the STM32U385RGTxQ device from STOP 2 mode while keeping the application configured for low-power consumption.	1	-	-	-	-	-	-	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 laps of time.	1	-	-	-	-	-	-	MX	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples	LCD	LCD_Blink_Frequency	How to use the embedded LCD glass controller and how to set the LCD blink mode and blinking frequency.	2	-	-	MX	MX	-	-	-	-	-
		LCD_Contrast_Control	How to use the embedded LCD glass controller and how to set the LCD Contrast level.	1	-	-	-	MX	-	-	-	-	-
		LCD_SegmentsDrive	-	1	-	-	-	MX	-	-	-	-	-
	LPTIM	LPTIM_PWM_LSE	How to configure and use, through the HAL LPTIM API, the LPTIM peripheral using LSE as counter clock, to generate a PWM signal, in a low-power mode.	1	-	-	-	-	-	-	MX	-	-
		LPTIM_PulseCounter	How to configure and use, through the LPTIM HAL API, the LPTIM peripheral to count pulses.	2	-	-	-	-	MX	-	MX	-	-
		LPTIM_Timeout	How to implement, through the HAL LPTIM API, a timeout with the LPTIMER peripheral, to wake up the system from a low-power Stop 1 mode.	2	-	-	-	-	-	-	MX	MX	-
	OPAMP	OPAMP_Follower	How to configure the OPAMP peripheral in follower mode interconnected with DAC and COMP.	1	-	-	-	-	-	-	MX	-	-
		OPAMP_PGA	This example shows how to use the built-in PGA mode (OPAMP programmable gain).	1	-	-	-	-	-	-	MX	-	-
	PKA	PKA_ECDSA_Sign	How to compute a signed message regarding the Elliptic curve digital signature algorithm (ECDSA).	1	-	-	-	-	-	-	MX	-	-
		PKA_ECDSA_Verify	How to determine if a given signature is valid regarding the Elliptic curve digital signature algorithm (ECDSA).	1	-	-	-	-	-	-	MX	-	-
		PKA_ModularExponentiation	How to use the PKA peripheral to execute modular exponentiation. This allows ciphering/ deciphering a text.	1	-	-	-	-	-	-	MX	-	-
	PWR	PWR_ModesSelection	This program shows how to configure the system to measure the current consumption in different low-power modes.	4	-	-	-	-	MX	MX	MX	MX	MX
		PWR_RUN_SMPS	How to use the SMPS PWR regulator.	1	-	-	-	-	-	-	MX	-	-
		PWR_SHUTDOWN	How to enter the system in SHUTDOWN mode and wake-up from this mode using external RESET or WKUP line.	1	-	-	-	-	-	-	MX	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples	PWR	PWR_SHUTDOWN_RTC	This example shows how to enter the system in SHUTDOWN mode and wake-up from this mode using the RTC Internal wake-up interrupt.	2	-	-	-	-	MX	-	MX	-	-
		PWR_SLEEP	How to enter the Sleep mode and wake up from this mode by using an interrupt.	1	-	-	-	-	-	-	MX	-	-
		PWR_STANDBY	How to enter the Standby mode and wake up from this mode by using an external reset or the WKUP pin.	1	-	-	-	-	-	-	MX	-	-
		PWR_STANDBY_RTC	How to enter the Standby mode and wake-up from this mode by using an external reset or the RTC wakeup timer.	1	-	-	-	-	-	-	MX	-	-
		PWR_STOP0	This example shows how to enter Stop 0 mode and wake up from this mode using an interrupt.	1	-	-	-	-	-	-	MX	-	-
		PWR_STOP0_RTC	This example shows how to enter Stop 0 mode and wake up from this mode using an interrupt from RTC Wake-up Timer.	1	-	-	-	-	-	-	MX	-	-
		PWR_STOP1	This example shows how to enter Stop 1 mode and wake up from this mode using an interrupt.	1	-	-	-	-	-	-	MX	-	-
		PWR_STOP1_RTC	How to enter the Stop 1 mode and wake-up from this mode by using the RTC wakeup timer.	1	-	-	-	-	-	-	MX	-	-
		PWR_STOP2	This example shows how to enter the system in Stop 2 mode and wake-up from this mode using external RESET or wake-up interrupt.	1	-	-	-	-	-	-	MX	-	-
		PWR_STOP2_RTC	How to enter the Stop 2 mode and wake-up from this mode using an external reset or RTC wakeup timer.	1	-	-	-	-	-	-	MX	-	-
		PWR_STOP3	This example shows how to enter the system in Stop 3 mode and wake-up from this mode using external RESET or wake-up interrupt.	1	-	-	-	-	-	-	MX	-	-
		PWR_STOP3_RTC	This example shows how to enter the system in Stop 3 mode and wake-up from this mode using the RTC Internal wake-up interrupt.	2	-	-	-	-	MX	-	MX	-	-
	RAMCFG	RAMCFG_Parity_Error	How to configure and use the RAMCFG HAL API to enable parity error detection and generate parity error interruption.	1	-	-	-	-	-	-	MX	-	-
		RAMCFG_WriteProtection	How to configure and use the RAMCFG HAL API to configure RAMCFG SRAM write protection page.	1	-	-	-	-	-	-	MX	-	-
	RCC	RCC_CRS_Synchronization_IT	Configuration of the clock recovery system (CRS) in Interrupt mode, using the RCC HAL API.	1	-	-	-	-	-	-	MX	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples	RCC	RCC_ClockConfig	Configuration of the system clock (SYSCLK) and modification of the clock settings in Run mode, using the RCC HAL API.	5	-	-	-	-	MX	MX	MX	MX	MX
		RCC_LSEConfig	Enabling/disabling of the low-speed external(LSE) RC oscillator (about 32 KHz) at run time, using the RCC HAL API.	1	-	-	-	-	-	-	MX	-	-
		RCC_LSIConfig	How to enable/disable the low-speed internal (LSI) RC oscillator (about 32 KHz) at run time, using the RCC HAL API.	1	-	-	-	-	-	-	MX	-	-
		RCC_MSIConfig	How to change the multi-speed internal (MSI) RC oscillator frequency at run time, using the PWR and RCC HAL API.	1	-	-	-	-	-	-	MX	-	-
		RCC_MSIPLLMode	How to change the multi-speed internal (MSI) RC oscillator frequency at run time with or without PLL mode, using the PWR and RCC HAL API. MSI use LSE as it's PLL input.	1	-	-	-	-	-	-	X	-	-
	RNG	RNG_MultiRNG_IT	Configuration of the RNG using the HAL API. This example uses RNG interrupts to generate 32-bit long random numbers.	1	-	-	-	-	-	-	MX	-	-
	RTC	RTC_Alarm	Configuration and generation of an RTC alarm using the RTC HAL API.	1	-	-	-	-	-	-	MX	-	-
		RTC_LowPower_STANDBY_WUT	How to periodically enter and wake up from STANDBY mode thanks to the RTC Wake-Up Timer (WUT).	1	-	-	-	-	-	-	MX	-	-
		RTC_Tamper	Configuration of the tamper detection with backup registers erase.	1	-	-	-	-	-	-	MX	-	-
		RTC_TimeStamp	Configuration of the RTC HAL API to demonstrate the timestamp feature.	1	-	-	-	-	-	-	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 the others are non-secure.	1	-	-	-	-	-	-	MX	-	-
	SAI	SAI_AudioLoopback	This project demonstrates how to configure SAI to perform audio loopback (record then playback).	1	-	-	-	-	-	-	MX	-	-
		SAI_AudioLoopback_LowPower	This project demonstrates how to configure SAI to perform audio loopback in low power sleep mode.	1	-	-	-	-	-	-	MX	-	-
		SAI_AudioPlay	This project demonstrates how to configure SAI to perform audio playback in DMA circular mode.	1	-	-	-	-	-	-	MX	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples	SAI	SAI_AudioRecord_PDM	This project demonstrates how to configure SAI to perform PDM record in DMA circular mode.	1	-	-	-	-	-	-	MX	-	-
		SAI_AudioRecord_PDM_LowPower	This project demonstrates how to configure SAI to perform PDM record in low power sleep mode.	1	-	-	-	-	-	-	MX	-	-
		SAI_DigitalLoopback	This project demonstrates how to configure SAI to perform transmission/reception in DMA circular mode.	1	-	-	-	-	-	-	MX	-	-
	SD	SD_ReadWrite_DMA	This example performs some write and read transfers to SD Card with SDMMC IP internal DMA mode.	1	-	-	-	-	-	-	MX	-	-
	SMBUS	SMBUS_TwoBoards_ComIT_Autonomous_ Master	How to handle SMBUS data buffer transmission/ reception between two boards, using autonomous mode.	1	-	-	-	-	-	-	MX	-	-
		SMBUS_TwoBoards_ComIT_Autonomous_Slave	How to handle SMBUS data buffer transmission/ reception between two boards, using autonomous mode.	1	-	-	-	-	-	-	MX	-	-
	SPI	SPI_FullDuplex_ComDMA_Autonomous_Master	How to handle a SPI as a Master to perform a data buffer transmission/reception in autonomous mode between two boards, using DMA in autonomous mode.	1	-	-	-	-	-	-	MX	-	-
		SPI_FullDuplex_ComDMA_Autonomous_Slave	How to handle a SPI as a Slave to perform a data buffer transmission/reception in autonomous mode between two boards, using DMA in autonomous mode.	1	-	-	-	-	-	-	MX	-	-
		SPI_FullDuplex_ComDMA_LowPower_Master	A data buffer transmission/reception via SPI using DMA wakes up slave board from low power mode.	3	-	-	-	-	MX	-	MX	-	MX
		SPI_FullDuplex_ComDMA_LowPower_Slave	A data buffer transmission/reception via SPI using DMA wakes up slave board from low power mode.	3	-	-	-	-	MX	-	MX	-	MX
		SPI_FullDuplex_ComDMA_Master	Data buffer transmission/reception between two boards via SPI using DMA.	2	-	-	-	-	MX	-	MX	-	-
		SPI_FullDuplex_ComDMA_Slave	Data buffer transmission/reception between two boards via SPI using DMA.	2	-	-	-	-	MX	-	MX	-	-
		SPI_FullDuplex_ComIT_Master	How to handle a SPI as a Master to perform Data buffer transmission/reception between two boards using Interrupt mode.	1	-	-	-	-	-	-	MX	-	-
		SPI_FullDuplex_ComIT_Slave	How to handle a SPI as a Slave to perform Data buffer transmission/reception between two boards using Interrupt mode.	1	-	-	-	-	-	-	MX	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples	SPI	SPI_FullDuplex_ComPolling_Master	How to handle a SPI as a Master to perform Data buffer transmission/reception between two boards using Polling mode.	1	-	-	-	-	-	-	MX	-	-
		SPI_FullDuplex_ComPolling_Slave	How to handle a SPI as a Slave to perform Data buffer transmission/reception between two boards using Polling mode.	1	-	-	-	-	-	-	MX	-	-
	TIM	TIM_InputCapture_DMA	This example demonstrates how to measure the LSI clock frequency thanks to the DMA interface of the TIM16 timer instance.	1	-	-	-	-	-	-	MX	-	-
		TIM_OCToggle	Configuration of the TIM peripheral to generate four different signals at four different frequencies.	1	-	-	-	-	-	-	MX	-	-
		TIM_PWMInput	How to use the TIM peripheral to measure the frequency and duty cycle of an external signal.	2	-	-	-	-	MX	-	MX	-	-
		TIM_PWMOutput	This example shows how to configure the TIM peripheral in PWM (Pulse Width Modulation) mode.	2	-	-	-	-	MX	-	MX	-	-
		TIM_WakeUpFromSleep	How to enter the Sleep mode and wake up from this mode by using the timer's update interrupt.	3	-	-	-	-	MX	-	MX	-	MX
	UART	LPUART_TwoBoards_ComIT	LPUART transmission (transmit/receive) in Interrupt mode between two boards.	2	-	-	-	-	MX	-	MX	-	-
		LPUART_WakeUpFromStop	Configuration of an LPUART to wake up the MCU from Stop 2 mode when a given stimulus is received.	3	-	-	-	-	MX	-	MX	-	MX
		LPUART_WakeUpFromStop2_autonomus	LPUART communication in DMA mode between two boards, with a one-shot autonomous echo performed by the receiver board while it uses Stop2 mode.	1	-	-	-	-	MX	-	-	-	-
		UART_Console	How to use the HAL UART API for UART transmission (printf/getchar) via console with user interaction.	2	-	-	-	-	MX	-	MX	-	-
		UART_HyperTerminal_DMA	UART transmission (transmit/receive) in DMA mode between a board and an HyperTerminal PC application.	1	-	-	-	-	-	-	MX	-	-
		UART_HyperTerminal_IT	UART transmission (transmit/receive) in Interrupt mode between a board and an HyperTerminal PC application.	1	-	-	-	-	-	-	MX	-	-
		UART_LowPower_HyperTerminal_DMA	This example describes an low power UART transmission (transmit/receive) in a low-power DMA mode between a board and an Hyperterminal PC application.	2	-	-	-	-	MX	-	MX	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW(1)	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples	UART	UART_Printf	Re-routing of the C library printf function to the UART.	2	-	-	-	-	-	-	MX	MX	-
		UART_ReceptionToIdle_CircularDMA	How to use the HAL UART API for reception "TO IDLE" event in circular DMA mode.	1	-	-	-	-	-	-	MX	-	-
		UART_TwoBoards_ComDMA	UART transmission (transmit/receive) in DMA mode between two boards.	2	-	-	-	-	MX	-	MX	-	-
		UART_TwoBoards_ComDMA_stop1_autonomous	USART1 transmission and reception between two NUCLEO-U3C5ZI-Q boards using DMA.	1	-	-	-	-	MX	-	-	-	-
		UART_TwoBoards_ComDMAlinkedlist	UART transmission (transmit/receive) in DMA mode using linkedlist between two boards.	1	-	-	-	-	-	-	MX	-	-
		UART_TwoBoards_ComIT	UART transmission (transmit/receive) in Interrupt mode between two boards.	1	-	-	-	-	-	-	MX	-	-
		UART_TwoBoards_ComPolling	UART transmission (transmit/receive) in Polling mode between two boards.	1	-	-	-	-	-	-	MX	-	-
	USART	USART_SlaveMode	This example describes an USART-SPI communication (transmit/receive) between two boards where the USART is configured as a slave.	1	-	-	-	-	-	-	MX	-	-
		USART_SlaveMode_DMA	This example describes an USART-SPI communication (transmit/receive) with DMA between two boards where the USART is configured as a slave.	1	-	-	-	-	-	-	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.	1	-	-	-	-	-	-	MX	-	-
	XSPI	XSPI_EEPROM_MemoryMapped	- This example describes how to erase part of the QSPI EEPROM memory, write data in IT mode and access to QSPI EEPROM memory in memory-mapped mode to check the data in an infinite loop.	1	-	MX	-	-	-	-	-	-	-
		XSPI_EEPROM_ReadWrite_DMA	- This example describes how to erase part of a QSPI EEPROM memory, write data in DMA mode, read data in DMA mode and compare the result in an infinite loop.	1	-	MX	-	-	-	-	-	-	-
		XSPI_MemoryToolbox_M2_IS25LP032D	How to use a XSPI ISSI IS25LP032D memory in memory-mapped mode.	1	-	-	-	-	MX	-	-	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples	XSPI	XSPI_MemoryToolbox_M2_M95P32	How to use a XSPI ST M95P32 memory in memory-mapped mode.	1	-	-	-	-	MX	-	-	-	-
		XSPI_MemoryToolbox_M2_MX25LM51245G	How to use a XSPI Macronix MX25LM51245G memory in memory-mapped mode.	1	-	-	-	-	MX	-	-	-	-
		XSPI_MemoryToolbox_M2_MX25UW25645G	How to use a XSPI Macronix MX25UW25645G memory in memory-mapped mode.	1	-	-	-	-	MX	-	-	-	-
		XSPI_MemoryToolbox_M2_W25Q16JV	How to use a XSPI Winbond W25Q16JV memory in memory-mapped mode.	1	-	-	-	-	MX	-	-	-	-
		XSPI_NOR_AutoPolling_DTR	This example demonstrates direct use of the STM32U3 HAL XSPI driver with the IS25LX512M NOR memory mounted on the STM32U3A6H-DK. No BSP memory access API is used. The BSP is used only for board LED control and board configuration headers.	1	-	-	MX	-	-	-	-	-	-
		XSPI_NOR_ExecuteInPlace	- This example describes how to execute a part of the code from the XSPI memory. To do this, a section is created where the function is stored.	1	-	MX	-	-	-	-	-	-	-
		XSPI_NOR_MemoryMapped	- This example describes how to erase part of the QSPI memory, write data in IT mode and access to QSPI memory in memory-mapped mode to check the data in an infinite loop.	1	-	MX	-	-	-	-	-	-	-
		XSPI_NOR_MemoryMapped_DTR	This example demonstrates how to use the STM32U3 XSPI peripheral with an external NOR memory in memory-mapped octal DTR mode.	1	-	-	MX	-	-	-	-	-	-
		XSPI_NOR_ReadWrite_DMA_DTR	This example demonstrates how to use the STM32U3 XSPI peripheral with an external NOR memory and GPDMA transfers in octal DTR mode.	1	-	-	MX	-	-	-	-	-	-
	Total number of examples			**280**	**209**	**5**	**6**	**4**	**41**	**11**	**187**	**10**	**16**
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.	1	-	-	-	-	-	-	MX	-	-
		ADC_MultiChannelSingleConversion_Init	How to use an ADC peripheral to convert several channels. ADC conversions are performed successively in a scan sequence.	1	-	-	-	-	-	-	MX	-	-
		ADC_Oversampling_Init	How to use an ADC peripheral with oversampling.	1	-	-	-	-	-	-	MX	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples_LL	ADC	ADC_SingleConversion_TriggerSW_IT_Init	How to use ADC to convert a single channel at each SW start, conversion performed using programming model: interrupt.	1	-	-	-	-	-	-	MX	-	-
		ADC_SingleConversion_TriggerTimer_DMA_Init	How to use an ADC peripheral to perform a single ADC conversion on a channel at each trigger event from a timer. Converted data is transferred by DMA into a table in RAM memory.	1	-	-	-	-	-	-	MX	-	-
	COMP	COMP_CompareGpioVsVrefInt_IT_Init	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.	1	-	-	-	-	-	-	MX	-	-
		COMP_CompareGpioVsVrefInt_OutputGpio_Init	How to use a comparator peripheral to compare a voltage level applied on a GPIO pin to the internal voltage reference (VrefInt) with comparator output connected to a GPIO pin.	1	-	-	-	-	-	-	MX	-	-
		COMP_CompareGpioVsVrefInt_Window_IT_Init	How to use a pair of comparator peripherals to compare a voltage level applied on a GPIO pin to two thresholds: the internal voltage reference (VrefInt) and a fraction of the internal voltage reference (VrefInt/2), in interrupt mode.	1	-	-	-	-	-	-	MX	-	-
	CORTEX	CORTEX_MPU	Presentation of the MPU features. This example configures MPU attributes of different MPU regions then configures a memory area as privileged read-only, and attempts to perform read and write operations in different modes.	1	-	-	-	-	-	-	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).	1	-	-	-	-	-	-	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).	1	-	-	-	-	-	-	MX	-	-
	CRS	CRS_Synchronization_IT	How to configure the clock recovery system in IT mode through the STM32U3xx CRS LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	1	-	-	-	-	-	-	MX	-	-
		CRS_Synchronization_Polling	How to configure the clock recovery system in polling mode through the STM32U3xx CRS LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	1	-	-	-	-	-	-	MX	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples_LL	DMA	DMA_CopyFromFlashToMemory_Init	How to use a DMA channel to transfer a word data buffer from Flash memory to embedded SRAM. The peripheral initialization uses LL initialization functions to demonstrate LL init usage.	1	-	-	-	-	-	-	MX	-	-
	DMA2D	DMA2D_MemoryToMemory	This example describes how to configure the DMA2D peripheral in Memory-to-Memory transfer mode. The example is based on the STM32U3xx DMA2D LL API. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	1	-	-	-	-	-	MX	-	-	-
	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 STM32U3xx LL API. Peripheral initialization is done using LL initialization function to demonstrate LL init usage.	1	-	-	-	-	-	-	MX	-	-
	GPIO	GPIO_InfiniteLedToggling_Init	How to configure and use GPIOs to toggle the on-board user LEDs every 250 ms. This example is based on the STM32U3xx LL API. The peripheral is initialized with LL initialization function to demonstrate LL init usage.	1	-	-	-	-	-	-	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 LL initialization function to demonstrate LL init usage.	1	-	-	-	-	-	-	MX	-	-
		I2C_OneBoard_Communication_ PollingAndIT_Init	How to transmit data bytes from an I2C master device using polling mode to an I2C slave device using interrupt mode. The peripheral is initialized with LL unitary service functions to optimize for performance and size.	1	-	-	-	-	-	-	MX	-	-
		I2C_TwoBoards_MasterRx_SlaveTx_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 LL unitary service functions to optimize for performance and size.	1	-	-	-	-	-	-	MX	-	-
		I2C_TwoBoards_MasterTx_SlaveRx_DMA_Init	How to transmit data bytes from an I2C master device using DMA mode to an I2C slave device using DMA mode. The peripheral is initialized with LL unitary service functions to optimize for performance and size.	1	-	-	-	-	-	-	MX	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples_LL	I2C	I2C_TwoBoards_MasterTx_SlaveRx_Init	How to transmit data bytes from an I2C master device using polling mode to an I2C slave device using interrupt mode. The peripheral is initialized with LL unitary service functions to optimize for performance and size.	1	-	-	-	-	-	-	MX	-	-
		I2C_TwoBoards_WakeUpFromStop2_IT_Init	How to handle the reception of a data byte from an I2C slave device in Stop 2 mode by an I2C master device, both using interrupt mode. The peripheral is initialized with LL unitary service functions to optimize for performance and size.	1	-	-	-	-	-	-	MX	-	-
		I2C_TwoBoards_WakeUpFromStop_IT_Init	How to handle the reception of a data byte from an I2C slave device in Stop0 mode by an I2C master device, both using interrupt mode. The peripheral is initialized with LL unitary service functions to optimize for performance and size.	1	-	-	-	-	-	-	MX	-	-
	I3C	I3C_Controller_Direct_Command_IT	How to handle a Direct Command procedure between an I3C Controller and an I3C Target, using IT.	1	-	-	-	-	-	-	MX	-	-
		I3C_Controller_Direct_Command_Polling	How to handle a Direct Command procedure between an I3C Controller and an I3C Target, using Polling.	1	-	-	-	-	-	-	MX	-	-
		I3C_Controller_Private_Command_IT	How to handle I3C as Controller data buffer transmission/reception between two boards, using interrupt.	1	-	-	-	-	-	-	MX	-	-
		I3C_Target_Direct_Command_IT	How to handle a Direct Command procedure between an I3C Controller and an I3C Target, using Controller in Interrupt.	1	-	-	-	-	-	-	MX	-	-
		I3C_Target_Direct_Command_Polling	How to handle a Direct Command procedure between an I3C Controller and an I3C Target, using Controller in Polling.	1	-	-	-	-	-	-	MX	-	-
		I3C_Target_Private_Command_IT	How to handle I3C as Target data buffer transmission/reception between two boards, using interrupt.	1	-	-	-	-	-	-	MX	-	-
	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.	1	-	-	-	-	-	-	MX	-	-
	LPUART	LPUART_WakeUpFromStop2_Init	Configuration of GPIO and LPUART peripherals to allow characters received on LPUART_RX pin to wake up the MCU from low-power mode. This example is based on the LPUART LL API. The peripheral initialization uses LL initialization function to demonstrate LL init usage.	1	-	-	-	-	-	MX	-	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples_LL	OPAMP	OPAMP_Follower_Init	How to use the OPAMP peripheral in follower mode interconnected with DAC.	1	-	-	-	-	-	-	MX	-	-
		OPAMP_PGA_Init	How to use the OPAMP peripheral in PGA mode (OPAMP programmable gain) with DAC.	1	-	-	-	-	-	-	MX	-	-
	PKA	PKA_ECDSA_Sign	How to use the low-layer PKA API to generate an ECDSA signature.	1	-	-	-	-	-	-	MX	-	-
	RCC	RCC_OutputSystemClockOnMCO	Configuration of MCO pin (PA8) to output the system clock.	1	-	-	-	-	-	-	MX	-	-
		RCC_UseHSEasSystemClock	Use of the RCC LL API to start the HSE and use it as system clock.	1	-	-	-	-	-	-	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).	1	-	-	-	-	-	-	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).	1	-	-	-	-	-	-	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.	1	-	-	-	-	-	-	MX	-	-
		RTC_Calendar_Init	Configuration of the LL API to set the RTC calendar. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	1	-	-	-	-	-	-	MX	-	-
		RTC_ExitStandbyWithWakeUpTimer_Init	How to periodically enter and wake up from STANDBY mode thanks to the RTC Wakeup Timer (WUT).	1	-	-	-	-	-	-	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).	1	-	-	-	-	-	-	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).	1	-	-	-	-	-	-	MX	-	-
	SPI	SPI_OneBoard_FullDuplex_IT_Init	This example shows how to perform SPI data buffer transmission/reception between two instances in the same board by using interruptions.	1	-	-	-	-	-	-	MX	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples_LL	SPI	SPI_OneBoard_HalfDuplex_DMA_Init	Configuration of GPIO and SPI peripherals to transmit bytes from an SPI Master device to an SPI Slave device in DMA mode. This example is based on the STM32U3xx SPI LL API. The peripheral initialization uses the LL initialization function to demonstrate LL init usage.	1	-	-	-	-	-	-	MX	-	-
		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 STM32U3xx SPI LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	1	-	-	-	-	-	-	MX	-	-
		SPI_TwoBoards_FullDuplex_DMA_Master_Init	Data buffer transmission and reception via SPI using DMA mode. This example is based on the STM32U3xx SPI LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	1	-	-	-	-	-	-	MX	-	-
		SPI_TwoBoards_FullDuplex_DMA_Slave_Init	Data buffer transmission and reception via SPI using DMA mode. This example is based on the STM32U3xx SPI LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	1	-	-	-	-	-	-	MX	-	-
		SPI_TwoBoards_FullDuplex_IT_Master_Init	Data buffer transmission and receptionvia SPI using Interrupt mode. This example is based on the STM32U3xx SPI LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	1	-	-	-	-	-	-	MX	-	-
		SPI_TwoBoards_FullDuplex_IT_Slave_Init	Data buffer transmission and receptionvia SPI using Interrupt mode. This example is based on the STM32U3xx SPI LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	1	-	-	-	-	-	-	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.	1	-	-	-	-	-	-	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. This example is based on the STM32U3xx TIM LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	1	-	-	-	-	-	-	MX	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples_LL	TIM	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. This example is based on the STM32U3xx TIM LL API. The peripheral initialization uses LL initialization function to demonstrate LL Init.	1	-	-	-	-	-	-	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 STM32U3xx TIM LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	1	-	-	-	-	-	-	MX	-	-
		TIM_PWMOutput_Init	Use of a timer peripheral to generate a PWM output signal and update the PWM duty cycle. This example is based on the STM32U3xx TIM LL API. The peripheral initialization uses LL initialization function to demonstrate LL Init.	1	-	-	-	-	-	-	MX	-	-
		TIM_TimeBase_Init	Configuration of the TIM peripheral to generate a timebase. This example is based on the STM32U3xx TIM LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	1	-	-	-	-	-	-	MX	-	-
	USART	USART_Communication_Rx_IT_Continuous_Init	This example shows how to configure GPIO and USART peripheral for continuously receiving characters from HyperTerminal (PC) in Asynchronous mode using Interrupt mode. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	1	-	-	-	-	-	-	MX	-	-
		USART_Communication_Rx_IT_Continuous_VCP_Init	This example shows how to configure GPIO and USART peripheral for continuously receiving characters from HyperTerminal (PC) in Asynchronous mode using Interrupt mode. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	1	-	-	-	-	-	-	MX	-	-
		USART_Communication_Rx_IT_Init	This example shows how to configure GPIO and USART peripheral for receiving characters from HyperTerminal (PC) in Asynchronous mode using Interrupt mode. Peripheral initialization is done using LL initialization function to demonstrate LL init usage.	1	-	-	-	-	-	-	MX	-	-
		USART_Communication_Rx_IT_VCP_Init	This example shows how to configure GPIO and USART peripheral for receiving characters from HyperTerminal (PC) in Asynchronous mode using Interrupt mode. Peripheral initialization is done using LL initialization function to demonstrate LL init usage.	1	-	-	-	-	-	-	MX	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples_LL	USART	USART_Communication_Tx_IT_Init	This example shows how to configure GPIO and USART peripheral to send characters asynchronously to HyperTerminal (PC) in Interrupt mode. This example is based on STM32U3xx USART LL API. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	1	-	-	-	-	-	-	MX	-	-
		USART_Communication_Tx_IT_VCP_Init	This example shows how to configure GPIO and USART peripheral to send characters asynchronously to HyperTerminal (PC) in Interrupt mode. This example is based on STM32U3xx USART LL API. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	1	-	-	-	-	-	-	MX	-	-
		USART_Communication_Tx_Init	This example shows how to configure GPIO and USART peripherals to send characters asynchronously to an HyperTerminal (PC) in Polling mode. If the transfer could not be completed within the allocated time, a timeout allows to exit from the sequence with a Timeout error code. This example is based on STM32U3xx USART LL API. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	1	-	-	-	-	-	-	MX	-	-
		USART_Communication_Tx_VCP_Init	This example shows how to configure GPIO and USART peripherals to send characters asynchronously to an HyperTerminal (PC) in Polling mode. If the transfer could not be completed within the allocated time, a timeout allows to exit from the sequence with a Timeout error code. This example is based on STM32U3xx USART LL API. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	1	-	-	-	-	-	-	MX	-	-
		USART_WakeUpFromStop1_Init	Configuration of GPIO and USART1 peripherals allows characters received on the USART_RX pin to wake up the MCU from Stop 1 low-power mode.	1	-	-	-	-	-	-	-	-	MX
	UTILS	UTILS_ConfigureSystemClock	Use of UTILS LL API to configure the system clock using MSIS as source clock.	1	-	-	-	-	-	-	MX	-	-
		UTILS_ReadDeviceInfo	This example reads the UID, Device ID and Revision ID and saves them into a global information buffer.	1	-	-	-	-	-	-	MX	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Examples_LL	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).	1	-	-	-	-	-	-	MX	-	-
	Total number of examples_ll			**69**	**69**	**0**	**0**	**0**	**0**	**2**	**66**	**0**	**1**
Examples_MIX	ADC	ADC_SingleConversion_TriggerSW_IT	How to use ADC to convert a single channel at each SW start, conversion performed using programming model: interrupt.	1	-	-	-	-	-	-	MX	-	-
	CRC	CRC_PolynomialUpdate	How to use the CRC peripheral through the STM32U3xx CRC HAL and LL API.	1	-	-	-	-	-	-	MX	-	-
	DMA	DMA_FLASHToRAM	How to use a DMA to transfer a word data buffer from Flash memory to embedded SRAM through the STM32U3xx DMA HAL and LL API. The LL API is used for performance improvement.	1	-	-	-	-	-	-	MX	-	-
	SPI	SPI_FullDuplex_ComPolling_Master	Data buffer transmission/reception between two boards via SPI using Polling mode.	1	-	-	-	-	-	-	MX	-	-
		SPI_FullDuplex_ComPolling_Slave	Data buffer transmission/reception between two boards via SPI using Polling mode.	1	-	-	-	-	-	-	MX	-	-
		SPI_HalfDuplex_ComPollingIT_Master	Data buffer transmission/reception between two boards via SPI using Polling (LL driver) and Interrupt modes (HAL driver).	1	-	-	-	-	-	-	MX	-	-
		SPI_HalfDuplex_ComPollingIT_Slave	Data buffer transmission/reception between two boards via SPI using Polling (LL driver) and Interrupt modes (HAL driver).	1	-	-	-	-	-	-	MX	-	-
	TIM	TIM_PWMInput	Use of the TIM peripheral to measure an external signal frequency and duty cycle.	1	-	-	-	-	-	-	MX	-	-
	Total number of examples_mix			**8**	**8**	**0**	**0**	**0**	**0**	**0**	**8**	**0**	**0**
Applications	-	MbedTLS_HW_ALT	The NUCLEO-U3C5ZI-Q MbedTLS_HW_ALT project is based on a recursive copy of NUCLEO-U385RG-Q/Applications/MbedTLS_HW_ALT/ content and few changes.	4	-	-	-	X	X	-	-	X	X
		MbedTLS_HW_KWE	The NUCLEO-U3C5ZI-Q MbedTLS_HW_KWE project is based on a recursive copy of NUCLEO-U385RG-Q/Applications/MbedTLS_HW_KWE/ content and few changes.	3	-	-	-	X	X	-	-	X	-
		MbedTLS_SW	The NUCLEO-U3C5ZI-Q MbedTLS_SW project is based on a recursive copy of NUCLEO-U385RG-Q/Applications/MbedTLS_SW/ content and few changes.	4	-	-	-	X	X	-	-	X	X

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Applications	-	OpenBootloader	This application exploits OpenBootloader Middleware to demonstrate how to develop an IAP application and how use it.	1	-	-	-	-	-	-	X	-	-
	FileX	Fx_File_Edit_Standalone	This application provides an example of FileX stack usage on NUCLEO-U385RG board, running in standalone mode (without ThreadX). It demonstrates how to create a Fat File system on the internal SRAM memory using FileX.	1	-	-	-	-	-	-	MX	-	-
		Fx_NOR_Write_Read_File	This application provides an example of Azure RTOS FileX and LevelX stacks usage on STM32U3A6-DK board, it demonstrates how to create a Fat File system on the NOR flash using FileX alongside LevelX. The application is designed to execute file operations on the MX25LM51245G NOR flash device, the code provides all required software code for properly managing it.	1	-	-	MX	-	-	-	-	-	-
		Fx_NoR_Write_Read_File	This application provides an example of Azure RTOS FileX and LevelX stacks usage on custom board. By default, it does not run on the STM32U366E-DK board, the user must add a custom board with NOR Flash support.	3	-	-	-	MX	MX	-	-	-	MX
	MbedTLS_HW_ALT	Cipher_AES_CBC_EncryptDecrypt_HAL	This application describes how to use the PSA Ref API to perform encryption and decryption using the AES CBC algorithm.	1	-	-	-	-	-	-	X	-	-
		Cipher_AES_CCM_Encrypt_Decrypt_HAL	This application describes how to use the PSA Ref API to perform authenticated encryption and verified decryption using the AES CCM algorithm.	1	-	-	-	-	-	-	X	-	-
		Cipher_AES_GCM_Encrypt_Decrypt_HAL	This application describes how to use the PSA Ref API to perform authenticated encryption and verified decryption using the AES GCM algorithm.	1	-	-	-	-	-	-	X	-	-
		ECC_ECDH_SharedSecretGeneration_HAL	This application describes how to use the Cryptographic Ref API to establish a shared secret using the ECDH algorithm over SECP256 curve.	1	-	-	-	-	-	-	X	-	-
		ECC_ECDSA_SignVerify_HAL	This application describes how to use the PSA Ref API to sign and verify a message using the ECDSA algorithm over SECP256 curve.	1	-	-	-	-	-	-	X	-	-
		Encrypted_ITS_KeyImport	This application describes how to use the PSA ITS alternative encrypted implementation to import AES-CBC key and store it in user persistent storage, the key is encrypted by the ITS.	1	-	-	-	-	-	-	X	-	-
		Hash_SHA2_Digest_HAL	This application describes how to use the PSA Ref API to digest a message using the SHA256 algorithm.	1	-	-	-	-	-	-	X	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Applications	MbedTLS_HW_ALT	MAC_HMAC_SHA2_AuthenticateVerify_HAL	This application describes how to use the PSA Ref API to authenticate and verify a message using the HMAC SHA256 algorithm.	1	-	-	-	-	-	-	X	-	-
		RSA_PKCS1v1.5_SignVerifyCRT_HAL	This application describes how to use the PSA Ref API to sign and verify a message using the RSA PKCS#1 v1.5 compliant algorithm.	1	-	-	-	-	-	-	X	-	-
		RSA_PKCS1v1.5_SignVerify_HAL	This application describes how to use the PSA Ref API to sign and verify a message using the RSA PKCS#1 v1.5 compliant algorithm.	1	-	-	-	-	-	-	X	-	-
		RSA_PKCS1v2.2_EncryptDecryptCRT_HAL	This application describes how to use the PSA Ref API to encrypt and decrypt a message using the RSA PKCS#1 v2.2 compliant algorithm.	1	-	-	-	-	-	-	X	-	-
		RSA_PKCS1v2.2_EncryptDecrypt_HAL	This application describes how to use the PSA Ref API to encrypt and decrypt a message using the RSA PKCS#1 v2.2 compliant algorithm.	1	-	-	-	-	-	-	X	-	-
		RSA_PKCS1v2.2_SignVerifyCRT_HAL	This application describes how to use the PSA Ref API to sign and verify a message using the RSA PKCS#1 v2.2 compliant algorithm.	1	-	-	-	-	-	-	X	-	-
		RSA_PKCS1v2.2_SignVerify_HAL	This application describes how to use the PSA Ref API to sign and verify a message using the RSA PKCS#1 v2.2 compliant algorithm.	1	-	-	-	-	-	-	X	-	-
	MbedTLS_HW_KWE	AES_WrapKey_KWE	This application describes how to use the PSA Crypto opaque driver based on STM32 Key Wrape Engine to wrap AES-GCM, AES-ECB, and AES-CBC keys.	1	-	-	-	-	-	-	X	-	-
		Cipher_AES_CBC_EncryptDecrypt_KWE	This application describes how to use the PSA Crypto opaque driver based on STM32 Key Wrape Engine to wrap the AES-CBC private key and use the wrapped key to perform encryption and verified the decryption using the AES CBC algorithm.	1	-	-	-	-	-	-	X	-	-
		Cipher_AES_CCM_Encrypt_Decrypt_KWE	This application describes how to use the PSA Crypto opaque driver based on STM32 Key Wrape Engine to wrap the AES-CCM private key and use the wrapped key to perform authenticated encryption and verified the decryption using the AES CCM algorithm.	1	-	-	-	-	-	-	X	-	-
		Cipher_AES_ECB_EncryptDecrypt_KWE	This application describes how to use the PSA Crypto opaque driver based on STM32 Key Wrape Engine to wrap the AES-ECB private key and use the wrapped key to perform encryption and verified the decryption using the AES CBC algorithm.	1	-	-	-	-	-	-	X	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Applications	MbedTLS_HW_KWE	Cipher_AES_GCM_Encrypt_Decrypt_KWE	This application describes how to use the PSA Crypto opaque driver based on STM32 Key Wrape Engine to wrap the AES-GCM private key and use the wrapped key to perform authenticated encryption and verified the decryption using the AES GCM algorithm.	1	-	-	-	-	-	-	X	-	-
		ECC_ECDH_GenerateWrappedKey_KWE	This application describes how to use the PSA Crypto opaque driver based on STM32 Key Wrape Engine to generate the ECDH private key.	1	-	-	-	-	-	-	X	-	-
		ECC_ECDH_SharedSecretGeneration_KWE	This application describes how to use the PSA Crypto opaque driver based on STM32 Key Wrape Engine to wrap the ECDH private key and use the wrapped key to generate the shared secret for ECDH key agreement.	1	-	-	-	-	-	-	X	-	-
		ECC_ECDH_WrapKey_KWE	This application describes how to use the PSA Crypto opaque driver based on STM32 Key Wrape Engine to wrap the ECDH private key.	1	-	-	-	-	-	-	X	-	-
		ECC_ECDSA_ExportPublicKey_KWE	This application describes how to use the PSA Ref API to sign and verify a message using the ECDSA algorithm over SECP256 curve.	1	-	-	-	-	-	-	X	-	-
		ECC_ECDSA_GenerateWrappedKey_KWE	This application describes how to use the PSA Crypto opaque driver based on STM32 Key Wrape Engine to generate the ECDSA private key.	1	-	-	-	-	-	-	X	-	-
		ECC_ECDSA_Sign_KWE	This application describes how to use the PSA Crypto opaque driver based on STM32 Key Wrape Engine to wrap the ECDSA private key and use the wrapped key for ECDSA signature.	1	-	-	-	-	-	-	X	-	-
		ECC_ECDSA_WrapKey_KWE	This application describes how to use the PSA Crypto opaque driver based on STM32 Key Wrape Engine to wrap the ECDSA private key.	1	-	-	-	-	-	-	X	-	-
		Initial_Attestation_KWE	This application describes how to use the PSA Crypto opaque driver based on STM32 Key Wrape Engine to sign an entity attestation token (ETA) for initial attestation using the device unique authentication key (DUA) in wrapped form.	1	-	-	-	-	-	-	X	-	-
		RSA_PKCS1v1.5_Sign_KWE	This application describes how to use the PSA Crypto opaque driver based on STM32 Key Wrape Engine to wrap the RSA private key and use the wrapped key to compute RSA PCKS1v1.5 signature.	1	-	-	-	-	-	-	X	-	-
		RSA_PKCS1v2.2_Decrypt_KWE	This application describes how to use the PSA Crypto opaque driver based on STM32 Key Wrape Engine to wrap the RSA private key and use the wrapped key for RSA PKCS1v2.2 decryption.	1	-	-	-	-	-	-	X	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Applications	MbedTLS_HW_KWE	RSA_PKCS1v2.2_Sign_KWE	This application describes how to use the PSA Crypto opaque driver based on STM32 Key Wrape Engine to wrap the RSA private key and use the wrapped key to compute RSA PKCS1v2.2 signature.	1	-	-	-	-	-	-	X	-	-
		RSA_WrapKey_KWE	This application describes how to use the PSA Crypto opaque driver based on STM32 Key Wrape Engine to wrap user RSA key.	1	-	-	-	-	-	-	X	-	-
	MbedTLS_SW	Cipher_AES_CBC_EncryptDecrypt_MBED	This application describes how to use the PSA Ref API to perform encryption and decryption using the AES CBC algorithm.	1	-	-	-	-	-	-	X	-	-
		Cipher_AES_CCM_Encrypt_Decrypt_MBED	This application describes how to use the PSA Ref API to perform authenticated encryption and verified decryption using the AES CCM algorithm.	1	-	-	-	-	-	-	X	-	-
		Cipher_AES_GCM_Encrypt_Decrypt_MBED	This application describes how to use the PSA Ref API to perform authenticated encryption and verified decryption using the AES GCM algorithm.	1	-	-	-	-	-	-	X	-	-
		Cipher_ChachaPoly_AuthEnc_VerifDec_MBED	This application describes how to use the PSA Ref API to perform authenticated encryption and verified decryption using the Chacha-Poly1305 algorithm.	1	-	-	-	-	-	-	X	-	-
		DRBG_RandomGeneration_MBED	This application describes how to use the PSA Ref API to generate random numbers using the DRBG module.	1	-	-	-	-	-	-	X	-	-
		ECC_ECDH_SharedSecretGeneration_MBED	This application describes how to use the PSA Ref API to establish a shared secret using the ECDH algorithm over SECP256 curve.	1	-	-	-	-	-	-	X	-	-
		ECC_ECDSA_SignVerify_MBED	This application describes how to use the PSA Ref API to sign and verify a message using the ECDSA algorithm over SECP256 curve.	1	-	-	-	-	-	-	X	-	-
		Hash_SHA2_Digest_MBED	This application describes how to use the PSA Ref API to digest a message using the SHA256 algorithm.	1	-	-	-	-	-	-	X	-	-
		MAC_AES_CMAC_AuthenticateVerify_MBED	This application describes how to use the PSA Ref API to authenticate and verify a message using the AES CMAC algorithm.	1	-	-	-	-	-	-	X	-	-
		MAC_HMAC_SHA2_AuthenticateVerify_MBED	This application describes how to use the PSA Ref API to authenticate and verify a message using the HMAC SHA256 algorithm.	1	-	-	-	-	-	-	X	-	-
		RSA_PKCS1v1.5_SignVerifyCRT_MBED	This application describes how to use the PSA Ref API to sign and verify a message using the RSA PKCS#1 v1.5 compliant algorithm.	1	-	-	-	-	-	-	X	-	-

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Applications	MbedTLS_SW	RSA_PKCS1v1.5_SignVerify_MBED	This application describes how to use the PSA Ref API to sign and verify a message using the RSA PKCS#1 v1.5 compliant algorithm.	1	-	-	-	-	-	-	X	-	-
		RSA_PKCS1v2.2_EncryptDecryptCRT_MBED	This application describes how to use the PSA Ref API to encrypt and decrypt a message using the RSA PKCS#1 v2.2 compliant algorithm.	1	-	-	-	-	-	-	X	-	-
		RSA_PKCS1v2.2_EncryptDecrypt_MBED	This application describes how to use the PSA Ref API to encrypt and decrypt a message using the RSA PKCS#1 v2.2 compliant algorithm.	1	-	-	-	-	-	-	X	-	-
		RSA_PKCS1v2.2_SignVerifyCRT_MBED	This application describes how to use the PSA Ref API to sign and verify a message using the RSA PKCS#1 v2.2 compliant algorithm.	1	-	-	-	-	-	-	X	-	-
		RSA_PKCS1v2.2_SignVerify_MBED	This application describes how to use the PSA Ref API to sign and verify a message using the RSA PKCS#1 v2.2 compliant algorithm.	1	-	-	-	-	-	-	X	-	-
	ROT	OEMiROT_Appli	This project provides a OEMiROT boot path Full Secure application example. Boot is performed through OEMiROT boot path after authenticity and the integrity checks of the project firmware and project data images.	4	-	-	-	-	X	X	X	X	X
		OEMiROT_Appli_TrustZone	This project provides a OEMiROT boot path application example. Boot is performed through OEMiROT boot path after authenticity and the integrity checks of the project firmware and project data images.	4	-	-	-	-	X	X	X	X	X
		OEMiROT_Boot	This project provides an OEMiROT example. OEMiROT boot path performs authenticity and the integrity checks of the project firmware and data images.	4	-	-	-	-	X	X	X	X	X
	ThreadX	Tx_LowPower	This application provides an example of Azure RTOS ThreadX stack usage, it shows how to develop an application using ThreadX LowPower feature.	5	-	-	-	MX	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.	1	-	-	-	-	-	-	X	-	-
		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.	5	-	-	MX	-	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.	4	-	-	-	MX	MX	-	MX	-	MX

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW(1)	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Applications	ThreadX	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.	1	-	-	-	-	-	-	MX	-	-
	TouchSensing	TouchSensing_1touchkey	Use of the TSC to perform continuous acquisitions of one channel in Interrupt mode.	1	-	-	-	MX	-	-	-	-	-
		TouchSensing_1touchkey_IT	Use of the TSC to perform continuous acquisitions of one channel in Interrupt mode.	2	-	-	MX	X	-	-	-	-	-
		TouchSensing_1touchkey_LowPower	Use of the TSC to perform continuous acquisitions of one channel in Interrupt mode (low power mode).	1	-	-	-	MX	-	-	-	-	-
	USBX	Ux_Device_Audio20_MultiChannel	This application provides an example of Azure RTOS USBX device stack usage on the STM32U3A6H-DK board.	1	-	-	MX	-	-	-	-	-	-
		Ux_Device_Audio20_MultiChannel_Standalone	This application provides an example of standalone USBX device stack usage on the STM32U3A6H-DK board.	1	-	-	MX	-	-	-	-	-	-
		Ux_Device_Audio20_Recording	This application provides an example of Azure RTOS USBX stack usage on STM32U3A6H-DK board, it shows how to develop a USB Device AUDIO 2.0 class application for microphone recording (IN endpoint).	1	-	-	MX	-	-	-	-	-	-
		Ux_Device_CDC_ACM	This application provides an example of Azure RTOS USBX stack usage on NUCLEO-U3C5ZI-Q board, it shows how to develop USB Device communication Class "CDC_ACM" based application.	5	-	-	-	-	MX	MX	MX	MX	MX
		Ux_Device_HID	This application provides an example of Azure RTOS USBX stack usage on STM32U366xx board, it shows how to develop USB Device Human Interface "HID" mouse based application.	3	-	-	-	MX	MX	MX	-	-	-
		Ux_Device_HID_Standalone	This application provides an example of USBX stack usage on NUCLEO-U3C5ZI-Q board, It shows how to develop USB Device Human Interface "HID" mouse based bare metal application.	3	-	-	-	-	MX	MX	MX	-	-
		Ux_Host_Audio_20	This application provides an example of Azure RTOS USBX stack usage. It shows how to develop USB Host Audio able to enumerate and communicate with a device audio speaker 1.0/2.0.	1	-	-	MX	-	-	-	-	-	-
		Ux_Host_CDC_ACM	This application provides an example of Azure RTOS USBX stack usage .	3	-	-	MX	-	MX	-	-	-	MX

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Applications	USBX	Ux_Host_HID	This application provides an example of Azure RTOS USBX stack usage.	3	-	-	MX	-	MX	-	MX	-	-
		Ux_Host_HID_Standalone	This application provides an example of Azure USBX stack usage.	3	-	-	MX	-	MX	-	MX	-	-
	hsp_engine	biquad_acc	This application exploits the HSP Engine Middleware to demonstrate how to process a Biquad DF1 Cascade in Accelerator Mode.	2	-	-	-	-	MX	MX	-	-	-
		cfft_acc	This application exploits the HSP Engine Middleware to demonstrate how to process a Complex FFT in Accelerator Mode.	2	-	-	-	-	MX	MX	-	-	-
		fir_acc	This application exploits the HSP Engine Middleware to demonstrate how to process a FIR in Accelerator Mode.	2	-	-	-	-	MX	MX	-	-	-
		fir_rfft_seq	This application exploits the HSP Engine Middleware to demonstrate with Sequencer Mode with Ping/Pong buffer: - How to process a FIR + FFT with processing list.	2	-	-	-	-	MX	MX	-	-	-
		fir_seq	This application exploits the HSP Engine Middleware to demonstrate how to process a FIR in Sequencer Mode.	2	-	-	-	-	MX	MX	-	-	-
		rfft_acc	This application exploits the HSP Engine Middleware to demonstrate how to process a RFFT in Accelerator Mode.	2	-	-	-	-	MX	MX	-	-	-
	Total number of applications			**135**	**81**	**0**	**10**	**7**	**22**	**16**	**61**	**8**	**11**
ROT_ Provisioning	-	OEMiROT	This section provides an overview of the available scripts for OEMiROT boot path.	5	-	-	-	-	X	X	X	X	X
		OEMiRoT_OEMuRoT	This section provides an overview of the available scripts for OEMiROT_OEMuROT boot path.	4	-	-	-	-	X	X	X	X	-
		OEMiRoT_TFM	This section provides an overview of the available scripts for OEMiRoT_TFM boot path.	1	-	-	-	-	X	-	-	-	-
	Total number of rot_provisioning			**10**	**3**	**0**	**0**	**0**	**3**	**2**	**2**	**2**	**1**
Templates_ Board	-	Starter project	This project provides a reference template for the NUCLEO-U3C5ZI-Q board based on the STM32Cube HAL API and the BSP drivers that can be used to build any firmware application.	5	-	-	-	-	MX	MX	MX	MX	MX
	Total number of templates_board			**5**	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**

Level	Module name	Project name	Description	Overall	Unique	STM32U3_CUSTOM_HW[1]	STM32U3A6H-DK	STM32U366E-DK	NUCLEO-U3C5ZI-Q	NUCLEO-U3A6ZH-Q	NUCLEO-U385RG-Q	NUCLEO-U366RE-Q	NUCLEO-U345RC-Q
Demonstration	-	Demo	The STM32Cube demonstration platform comes on top of 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.	2	-	-	X	X	-	-	-	-	-
	Total number of demonstration			**2**	**1**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**0**
Total number of projects				**536**	**379**	**5**	**18**	**13**	**72**	**37**	**331**	**26**	**34**

1. *The firmware examples listed in this column are not supported by the STM32U366E-DK, NUCLEO-U3C5ZI-Q, NUCLEO-U385RG-Q, NUCLEO-U366RE-Q, or NUCLEO-U345RC-Q boards. They can be used with a custom board.*

2 Reference documents

The reference documents are available on www.st.com/stm32cubefw or from the STM32CubeU3 product web page:

- Latest release of the STM32U3 firmware package
- *Getting started with STM32CubeU3 for STM32U3 series* (UM3427)
- *Description of STM32U3 HAL and LL drivers* (UM3439)

Revision history

Table 2. **Document revision history**

Date	Revision	Changes
19-Feb-2025	1	Initial release.
23-Feb-2026	2	Added examples for board NUCLEO-U3C5ZI-Q.
03-Jun-2026	3	Added examples for boards NUCLEO-U345RC-Q, NUCLEO-U366RE-Q, and STM32U366E-DK.
26-Oct-2026	4	Added examples for boards STM32U3A6H-DK and NUCLEO-U3A6ZH-Q.

Contents

1 STM32CubeU3 examples 2
2 Reference documents 36
Revision history 37

List of tables

Table 1. STM32CubeU3 firmware examples 3
Table 2. Document revision history 37

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.

In the event of any conflict between the provisions of this document and the provisions of any contractual arrangement in force between the purchasers and ST, the provisions of such contractual arrangement shall prevail.

The 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.

The 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 the 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.

If the purchasers identify an ST product that meets their functional and performance requirements but that is not designated for the purchasers' market segment, the purchasers shall contact ST for more information.

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.

© 2026 STMicroelectronics – All rights reserved