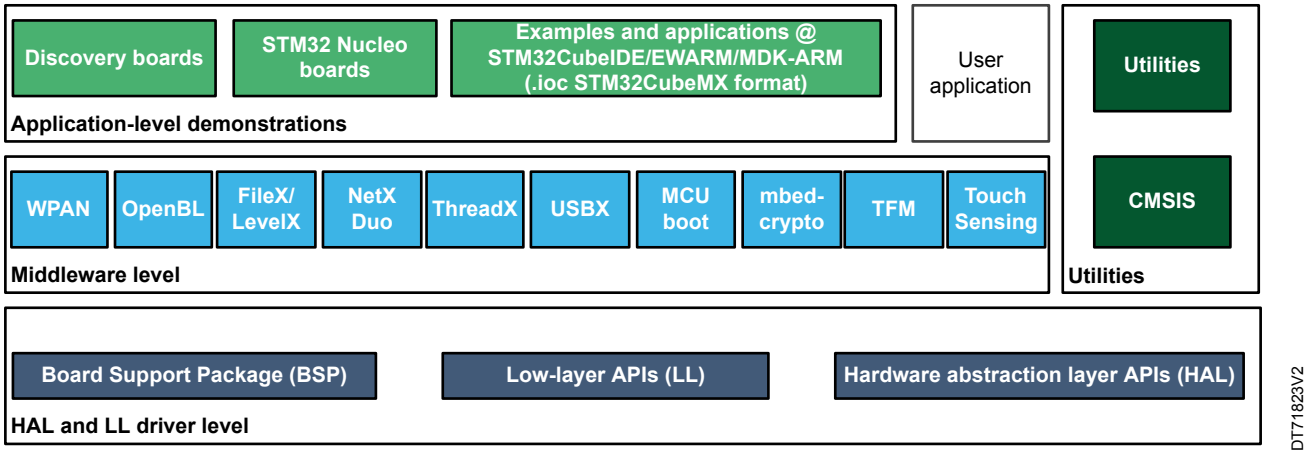
STM32Cube MCU package examples for STM32WBA series

Introduction

The STM32CubeWBA 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. STM32CubeWBA firmware components



1 STM32CubeWBA examples

The examples are classified depending on the STM32Cube level 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_WBA_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.txt 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 your preferred toolchain.
2. Rebuild all files and load the image into target memory.
3. Run the example by following the readme.txt instructions.

Note: Refer to section "Development Toolchains and Compilers" and "Supported Devices and EVAL, Nucleo, and Discovery boards" of the firmware package release notes to know about the SW/HW environment used for the firmware development and validation. The correct operation of the provided examples is not guaranteed on 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 your board, provided it has the same hardware functions (LED, LCD display, push-buttons, etc.). The BSP is based on a modular architecture that can be easily ported to any hardware by implementing the low-level routines.

Table 1. STM32CubeWBA firmware examples contains the list of examples provided with STM32CubeWBA MCU Package.

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

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

Reference materials available on www.st.com/stm32cubefw.

Table 1. STM32CubeWBA firmware examples

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
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).	MX	X	MX	X	New	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).	MX	X	MX	X	New	X
	ROT	OEMiROT_Appli	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	-	-	-	-	-
		OEMiROT_Appli_TrustZone	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	-	-	-	-	-
	Total number of templates			**4**	**2**	**2**	**2**	**2**	**2**
Templates_LL	-	TrustZoneDisabled	This project provides a reference template based on the STM32Cube LL API that can be used to build any firmware application.	MX	X	MX	X	New	X
	Total number of templates_ll			**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.	-	-	MX	MX	-	-
		ADC_ContinuousConversion_TriggerSW_Low Power	How to use an ADC to convert a single channel using the ADC low-power feature, the autowait, and the automatic power-off.	-	-	MX	-	-	-
		ADC_MultiChannelSingleConversion	How to use an ADC peripheral to convert several channels. ADC conversions are performed successively in a scan sequence.	-	-	MX	MX	-	-
		ADC_Oversampling	How to use an ADC peripheral with oversampling.	-	-	-	MX	-	-
		ADC_SingleConversion_TriggerSW_IT	How to use an ADC to convert a single channel at each software start. ADC conversions are performed using the interrupt programming model.	-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Examples	ADC	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. The converted data is transferred by DMA into a table in RAM.	-	-	-	MX	-	-
	BSP	BSP_Example	How to use the different BSP drivers of the board.	-	X	-	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 (V_{REFINT}), in interrupt mode.	-	-	-	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 (V_{REFINT}) and a fraction of the internal voltage reference ($V_{REFINT}/2$), in interrupt mode.	-	-	-	MX	-	-
	CORTEX	CORTEXM_InterruptSwitch_TrustZone	How to first use an interrupt in a secure application, and then assign it to the nonsecure application when TrustZone® security is activated (option bit TZEN = 1).	-	-	-	MX	-	-
		CORTEXM_MPU	This example presents the MPU features. It configures the MPU attributes of different MPU regions, then a memory area as privileged read only, and attempts to perform read and write operations in different modes.	-	-	-	MX	-	-
		CORTEXM_ModePrivilege	How to modify the Thread mode privilege access and stack. The Thread mode is entered upon reset or when returning from an exception.	-	MX	MX	MX	-	-
		CORTEXM_ProcessStack	How to modify the Thread mode stack. The Thread mode is entered upon 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	-	-
		CORTEXM_SysTick_LSE	How to use the LSE as SysTick clock source with a 1 ms timebase to toggle LEDs.	-	-	-	X	-	-
	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].	-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Examples	CRC	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.	-	-	-	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	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_AESModes	How to use the CRYP peripheral to encrypt and decrypt data using the AES in chaining modes (ECB, CBC, CTR).	-	-	MX	MX	-	-
		CRYP_DMA	How to use the AES peripheral to encrypt and decrypt data using the AES128 algorithm with ECB chaining mode in DMA mode.	-	-	-	MX	-	-
		CRYP_GCM_GMAC_CMAC_Suspension	How to use the CRYP AES peripheral to suspend and then resume the AES GCM, GMAC, or CMAC processing of a message for encryption, decryption, or authentication tag computation of a higher-priority message (CMAC).	-	-	MX	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	-	-
	DMA	DMA_FLASHToRAM	How to use a DMA to transfer a word data buffer from flash memory to the embedded SRAM through the HAL API.	-	MX	MX	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Examples	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 reloads the next transfer parameters and starts it (without CPU intervention).	-	MX	-	-	-	-
		DMA_MemToMem_TrustZone	How to use the DMA HAL API to perform memory-to-memory data transfers over secure and nonsecure DMA channels, when TrustZone® security is activated (option bit TZEN = 1).	-	-	-	MX	-	-
	FLASH	FLASH_BlockBased_TrustZone	How to configure and use the FLASH HAL API to manage the block-based security of internal flash memory between secure and nonsecure applications, when TrustZone® security is activated (option bit TZEN = 1).	-	-	-	MX	-	-
		FLASH_DualBoot	How to configure and use the FLASH HAL API to program bank1 and bank2 of the STM32WBAxx internal flash memory mounted on the NUCLEO-WBA65RI, and swap between them.	-	-	MX	-	-	-
		FLASH_EraseProgram	How to configure and use the FLASH HAL API to erase and program the internal flash memory. At the beginning of the main program, the HAL_Init() function is called to reset all the peripherals, and initialize the flash memory interface and the SysTick.	-	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).	-	MX	-	MX	-	-
		FLASH_WriteProtection	How to configure and use the FLASH HAL API to enable and disable the write protection of the internal flash memory.	-	-	-	MX	-	-
	GPIO	GPIO_EXTI	How to configure external interrupt lines.	-	-	-	MX	-	-
		GPIO_IOToggle	How to configure and use GPIOs through the HAL API.	-	MX	MX	MX	-	-
		GPIO_IOToggle_TrustZone	How to use the GPIO HAL API to toggle secure and nonsecure I/Os when TrustZone® security is activated (option bit TZEN = 1).	-	-	-	MX	-	-
	GTZC	GTZC_TZSC_MPCBB_TrustZone	How to use GTZC MPCBB HAL API to build any example with SecureFault detection when TrustZone® security is activated (option bit TZEN = 1).	-	MX	-	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Examples	HAL	HAL_TimeBase	How to customize the HAL by using a general-purpose timer as the main source of time base, instead of the SysTick.	-	-	-	MX	-	-
		HAL_TimeBase_RTC_WKUP	How to customize the HAL by using an RTC wake-up as the main source of time base, instead of the SysTick.	-	-	-	MX	-	-
		HAL_TimeBase_TIM	How to customize the HAL by using a general-purpose timer as the main source of time base instead of the SysTick.	-	-	-	MX	-	-
	HASH	HASH_HMAC_SHA1MD5	How to use the HASH peripheral to hash data with HMAC SHA-1 and HMAC MD5 algorithms.	-	-	-	MX	-	-
		HASH_HMAC_SHA256MD5_IT_Suspension	How to suspend the HMAC digest computation when data are fed in interrupt mode.	-	-	-	MX	-	-
		HASH_SHA1MD5	How to use the HASH peripheral to hash data using SHA-1 and MD5 algorithms.	-	-	-	MX	-	-
		HASH_SHA1MD5_DMA	How to use the HASH peripheral to hash data using SHA-1 and MD5 algorithms.	-	-	-	MX	-	-
		HASH_SHA1SHA224_IT_Suspension	How to suspend the HASH peripheral when data is fed in interrupt mode.	-	-	-	X	-	-
		HASH_SHA1_DMA_TrustZone	How to use a secure HASH SHA-1 computation service based on a secure DMA channel when TrustZone® security is activated (option bit TZEN = 1).	-	-	-	MX	-	-
		HASH_SHA224SHA256_DMA	How to use the HASH peripheral to hash data with SHA224 and SHA256 algorithms.	-	-	-	MX	-	-
	HSEM	HSEM_ProcessSync	How to use a hardware semaphore to synchronize two processes.	-	-	MX	MX	-	-
		HSEM_ReadLock	How to enable, take then release a semaphore using two different processes.	-	-	MX	MX	-	-
	I2C	I2C_TwoBoards_AdvComIT	How to handle several I2C data buffer transmission/ reception between a controller and a target device, using an interrupt.	-	-	-	MX	-	-
		I2C_TwoBoards_ComDMA	How to handle I2C data buffer transmission/reception between two boards, via DMA.	-	-	MX	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Examples	I2C	I2C_TwoBoards_ComDMA_Autonomous_ Master	How to handle I2C data buffer transmission/reception between two boards, via DMA.	-	-	-	MX	-	-
		I2C_TwoBoards_ComDMA_Autonomous_ Slave	How to handle I2C data buffer transmission/reception between two boards, via DMA.	-	-	-	MX	-	-
		I2C_TwoBoards_ComIT	How to handle I2C data buffer transmission/reception between two boards, using an interrupt.	-	-	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 using a restart condition.	-	-	-	MX	-	-
		I2C_TwoBoards_RestartComIT	How to handle single I2C data buffer transmission/ reception between two boards, in interrupt mode and using a 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	-	-
	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	-	-
	LPTIM	LPTIM_PulseCounter	How to use the LPTIM HAL API to configure and use the LPTIM peripheral to count pulses.	-	-	-	MX	-	-
		LPTIM_Timeout	How to use the LPTIM HAL API to implement a timeout with the LPTIMER peripheral, and wake up the system from a low-power mode.	-	-	-	MX	-	-
	PKA	PKA_ECCscalarMultiplication	How to use the PKA peripheral to execute ECC scalar multiplication. This enables the generation of a public key from a private key.	-	-	-	MX	-	-
		PKA_ECCscalarMultiplication_IT	How to use the PKA peripheral to execute ECC scalar multiplication. This enables the generation of a public key from a private key in interrupt mode.	-	-	-	MX	-	-
		PKA_ECDSA_Sign	How to compute a signed message using the elliptic curve digital signature algorithm (ECDSA).	-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Examples	PKA	PKA_ECDSA_Sign_IT	How to compute a signed message using the elliptic curve digital signature algorithm (ECDSA) in interrupt mode.	-	-	-	MX	-	-
		PKA_ECDSA_Verify	How to determine if a given signature is valid using the elliptic curve digital signature algorithm (ECDSA).	-	-	-	MX	-	-
		PKA_ECDSA_Verify_IT	How to determine if a given signature is valid using the elliptic curve digital signature algorithm (ECDSA) in interrupt mode.	-	-	-	MX	-	-
		PKA_ModularExponentiation	How to use the PKA peripheral to execute modular exponentiation. This enables the encryption and decryption of a text.	-	-	-	MX	-	-
		PKA_ModularExponentiationCRT	How to compute the Chinese remainder theorem (CRT) optimization.	-	-	-	MX	-	-
		PKA_ModularExponentiationCRT_IT	How to compute the Chinese remainder theorem (CRT) optimization in interrupt mode.	-	-	-	MX	-	-
		PKA_ModularExponentiation_IT	How to use the PKA peripheral to execute modular exponentiation. This enables the encryption and decryption of a text in interrupt mode.	-	-	-	MX	-	-
		PKA_PointCheck	How to use the PKA peripheral to determine if a point is on a curve. This enables the validation of an external public key.	-	-	-	MX	-	-
		PKA_PointCheck_IT	How to use the PKA peripheral to determine if a point is on a curve. This enables the validation of an external public key.	-	-	-	MX	-	-
	PWR	PWR_RUN_SMPS	How to use the SMPS PWR regulator.	-	-	-	MX	-	-
		PWR_SLEEP	How to enter Sleep mode and wake up from this mode by using an interrupt.	-	-	-	MX	-	-
		PWR_STANDBY	How to enter Standby mode and wake up from this mode by using an external reset or the WKUP pin.	-	-	-	MX	-	-
		PWR_STANDBY_RTC	How to enter Standby mode and wake up from this mode by using an external reset or the RTC wake-up timer.	-	MX	MX	MX	-	-
		PWR_STOP1	How to enter Stop 1 mode and wake up from this mode using an interrupt.	-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Examples	PWR	PWR_STOP1_RTC	How to enter Stop 1 mode and wake up from this mode by using the RTC wake-up timer.	-	MX	-	MX	-	-
		PWR_STOP2	How to enter Stop 2 mode and wake up from this mode using an external reset or a wake-up interrupt.	-	-	MX	-	-	-
	RAMCFG	RAMCFG_Parity_Error	How to configure and use the RAMCFG HAL API to enable parity error detection and generate a parity error interrupt.	-	-	-	MX	-	-
		RAMCFG_WriteProtection	How to configure and use the RAMCFG HAL API to configure RAMCFG SRAM write-protection page.	-	-	-	MX	-	-
	RCC	RCC_ClockConfig	How to configure the system clock (SYSCLK) and modify the clock settings in Run mode, using the RCC HAL API.	-	MX	MX	MX	-	-
		RCC_ClockConfig_TrustZone	How to configure the system clock (SYSCLK) in Run mode from the secure application upon request from the nonsecure application, using the RCC HAL API, when TrustZone® security is activated (option bit TZEN = 1).	-	-	-	MX	-	-
		RCC_LSEConfig	How to enable/disable the low-speed external (LSE) RC oscillator (approximately 32 KHz) at runtime, using the RCC HAL API.	-	-	-	MX	-	-
		RCC_LSIConfig	How to enable/disable the low-speed internal (LSI) RC oscillator (approximately 32 KHz) at runtime, using the RCC HAL API.	-	-	-	MX	-	-
	RNG	RNG_MultiRNG	How to configure the RNG using the HAL API. This example uses the RNG to generate 32-bit long random numbers.	-	-	-	MX	-	-
		RNG_MultiRNG_IT	How to configure the RNG using the HAL API. This example uses RNG interrupts to generate 32-bit long random numbers.	-	-	-	MX	-	-
	RTC	RTC_ActiveTamper	How to configure the active tamper detection with backup register erase.	-	MX	-	-	-	-
		RTC_Alarm	How to configure and generate an RTC alarm using the RTC HAL API.	-	MX	-	MX	-	-
		RTC_Calendar	How to configure the calendar using the RTC HAL API.	-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Examples	RTC	RTC_LSI	How to use the LSI clock source autocalibration to achieve a precise RTC clock.	-	-	-	MX	-	-
		RTC_LowPower_STANDBY_WUT	How to periodically enter and wake up from Standby mode using the RTC wake-up timer (WUT).	-	MX	-	-	-	-
		RTC_Tamper	How to configure the tamper detection with backup register erase.	-	-	-	MX	-	-
		RTC_TimeStamp	How to configure the RTC HAL API to demonstrate the timestamp feature.	-	MX	-	MX	-	-
		RTC_TrustZone	How to configure the TrustZone®-aware RTC peripheral when TrustZone® security is activated (option bit TZEN = 1). Some RTC features can be secure while others are nonsecure.	-	-	-	MX	-	-
	SAI	SAI_AudioPlay	How to use the SAI HAL API to play an audio file using the DMA circular mode, and handle the buffer update.	MX	MX	-	-	-	-
	SPI	SPI_FullDuplex_ComDMA_Autonomous_Master	How to perform data buffer transmission/reception between two boards via SPI, using the DMA.	-	-	-	MX	-	-
		SPI_FullDuplex_ComDMA_Autonomous_Slave	How to perform data buffer transmission/reception between two boards via SPI, using the DMA.	-	-	-	MX	-	-
		SPI_FullDuplex_ComDMA_Master	How to perform data buffer transmission/reception between two boards via SPI, using the DMA.	-	-	MX	MX	-	-
		SPI_FullDuplex_ComDMA_Slave	How to perform data buffer transmission/reception between two boards via SPI, using the DMA.	-	-	MX	MX	-	-
		SPI_FullDuplex_ComIT_Master	How to perform data buffer transmission/reception between two boards via SPI, in interrupt mode.	-	-	MX	MX	-	-
		SPI_FullDuplex_ComIT_Slave	How to perform data buffer transmission/reception between two boards via SPI, in interrupt mode.	-	-	MX	MX	-	-
		SPI_FullDuplex_ComPolling_Master	How to perform data buffer transmission/reception between two boards via SPI, in polling mode.	-	-	-	MX	-	-
		SPI_FullDuplex_ComPolling_Slave	How to perform data buffer transmission/reception between two boards via SPI, in polling mode.	-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Examples	TIM	TIM_DMA	How to use the DMA with timer update request to transfer data from memory to the timer capture compare register 3 (TIMx_CCR3).	-	-	MX	MX	-	-
		TIM_DMABurst	How to update the TIMER channel 1 period and duty cycle using the TIMER GPDMA burst feature.	-	-	-	MX	-	-
		TIM_ExtTriggerSynchro	How to synchronize TIMER peripherals in cascade mode using an external trigger.	-	-	MX	MX	-	-
		TIM_InputCapture	How to use the TIMER peripheral to measure an external signal frequency.	-	-	-	MX	-	-
		TIM_OCActive	How to configure the TIMER 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	How to configure the TIMER peripheral in output compare inactive mode with the corresponding interrupt request for each channel.	-	-	-	MX	-	-
		TIM_OCToggle	How to configure the TIMER peripheral to generate four different signals at four different frequencies.	-	-	-	MX	-	-
		TIM_OnePulse	How to use the TIMER peripheral to generate a single pulse when a rising edge of an external signal is received on the TIMER input pin.	-	-	-	MX	-	-
		TIM_PWMInput	How to use the TIMER peripheral to measure the frequency and duty cycle of an external signal.	-	-	-	MX	-	-
		TIM_PWMOutput	How to configure the TIMER peripheral in PWM (pulse width modulation) mode.	-	-	MX	MX	-	-
		TIM_TimeBase	How to configure the TIMER peripheral to generate a time base of one second with the corresponding interrupt request.	-	-	-	MX	-	-
	UART	UART_HyperTerminal_DMA	How to perform UART transmission (transmit/receive) in DMA mode between a board and a HyperTerminal PC application.	-	-	-	MX	-	-
		UART_HyperTerminal_IT	How to perform UART transmission (transmit/receive) in interrupt mode between a board and a HyperTerminal PC application.	-	-	MX	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Examples	UART	UART_LowPower_HyperTerminal_DMA	How to perform low-power UART transmission (transmit/receive) in DMA mode between a board and a HyperTerminal PC application.	-	-	MX	-	-	-
		UART_Printf	How to reroute the C library printf function to the UART.	-	-	MX	MX	-	-
		UART_TwoBoards_ComIT	How to perform UART transmission (transmit/receive) in interrupt mode between two boards.	-	-	-	MX	-	-
		UART_TwoBoards_ComPolling	How to perform UART transmission (transmit/receive) in polling mode between two boards.	-	-	-	MX	-	-
	USART	USART_SlaveMode	How to perform USART-SPI communication (transmit/receive) between two boards where the USART is configured as a slave.	-	-	-	MX	-	-
	WWDG	WWDG_Example	How to configure 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	-	-
	Total number of examples			**1**	**16**	**28**	**116**	**0**	**0**
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_ContinuousConv_TriggerSW_LowPower_Init	How to use an ADC to convert a single channel with ADC low-power features auto wait and auto power-off.	-	-	-	MX	-	-
		ADC_ContinuousConversion_TriggerSW_Init	How to use an ADC peripheral to convert a single channel continuously, from a software start.	-	-	-	MX	-	-
		ADC_Oversampling_Init	How to use an ADC peripheral with oversampling.	-	-	-	MX	-	-
		ADC_SingleConversion_TriggerSW_DMA_Init	How to use an ADC peripheral to perform a single ADC conversion on a channel at each software start. The converted data is transferred by DMA into a table in RAM.	-	-	-	MX	-	-
		ADC_SingleConversion_TriggerSW_IT_Init	How to use ADC to convert a single channel at each software start. The conversion is performed using the interrupt programming model.	-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Examples_LL	ADC	ADC_SingleConversion_TriggerSW_Init	How to use ADC to convert a single channel at each software start. The conversion is performed using the polling programming model.	-	-	-	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. The converted data is transferred by DMA into a table in RAM.	-	-	-	MX	-	-
		ADC_TemperatureSensor_Init	How to use an ADC peripheral to perform a single ADC conversion on the internal temperature sensor and calculate the temperature in degrees Celsius.	-	-	-	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 (V_{REFINT}), in interrupt mode.	-	-	-	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 (V_{REFINT}) with a comparator output connected to a GPIO pin.	-	-	-	MX	-	-
	CORTEX	CORTEX_MPU	Presentation of the MPU features. This example first configures the 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.	-	-	-	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 to optimize performance and size.	-	-	MX	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.	-	-	-	MX	-	-
	DMA	DMA_CopyFromFlashToMemory_Init	How to use a DMA channel to transfer a word data buffer from flash memory to the embedded SRAM. The peripheral initialization uses LL initialization functions to demonstrate LL init usage.	-	MX	-	MX	-	-
	EXTI	EXTI_ToggleLedOnIT_Init	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 STM32WBAxx LL API. The peripheral initialization is done using the LL initialization function to demonstrate LL init usage.	-	MX	-	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Examples_LL	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 STM32WBAxx LL API. The peripheral is initialized using the LL initialization function to demonstrate LL init usage.	-	MX	MX	MX	-	-
	HSEM	HSEM_DualProcess	How to use the low-layer HSEM API to initialize, lock, and unlock hardware semaphore in the context of two processes accessing the same resource.	-	-	-	MX	-	-
		HSEM_DualProcess_IT	How to use the low-layer HSEM API to initialize, lock, and unlock hardware semaphore in the context of two processes accessing the same resource.	-	-	-	MX	-	-
	I2C	I2C_OneBoard_AdvCommunication_ DMAAndIT_Init	How to exchange data between an I2C controller device in DMA mode and an I2C target device in interrupt mode. The peripheral is initialized with LL unitary service functions to optimize performance and size.	-	-	-	MX	-	-
		I2C_OneBoard_Communication_ DMAAndIT_Init	How to transmit data bytes from an I2C controller device using DMA mode to an I2C target device using interrupt mode. The peripheral is initialized with LL unitary service functions to optimize performance and size.	-	-	-	MX	-	-
		I2C_OneBoard_Communication_IT_Init	How to handle the reception of one data byte from an I2C target device by an I2C controller device. Both devices operate in interrupt mode. The peripheral is initialized with the LL initialization function to demonstrate LL init usage.	-	-	-	MX	-	-
		I2C_OneBoard_Communication_PollingAndIT_ Init	How to transmit data bytes from an I2C controller device using polling mode to an I2C target device using interrupt mode. The peripheral is initialized with LL unitary service functions to optimize performance and size.	-	-	-	MX	-	-
		I2C_TwoBoards_MasterRx_SlaveTx_IT_Init	How to handle the reception of one data byte from an I2C target device by an I2C controller device. Both devices operate in interrupt mode. The peripheral is initialized with LL unitary service functions to optimize performance and size.	-	-	-	MX	-	-
		I2C_TwoBoards_MasterTx_SlaveRx_DMA_Init	How to transmit data bytes from an I2C controller device using DMA mode to an I2C target device using DMA mode. The peripheral is initialized with LL unitary service functions to optimize performance and size.	-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Examples_LL	I2C	I2C_TwoBoards_MasterTx_SlaveRx_Init	How to transmit data bytes from an I2C controller device using polling mode to an I2C target device using interrupt mode. The peripheral is initialized with LL unitary service functions to optimize performance and size.	-	-	-	MX	-	-
		I2C_TwoBoards_WakeUpFromStop_IT_Init	How to handle the reception of a data byte from an I2C target device in Stop 0 mode by an I2C controller device, both using interrupt mode. The peripheral is initialized with LL unitary service functions to optimize performance and size.	-	-	-	MX	-	-
	LPTIM	LPTIM_PulseCounter_Init	How to use the LPTIM peripheral in counter mode to generate a PWM output signal and update its duty cycle. This example is based on the STM32WBAxx LPTIM LL API. The peripheral is initialized with the LL initialization function to demonstrate LL init usage.	-	-	-	MX	-	-
	PKA	PKA_ECDSA_Sign	How to use the low-layer PKA API to generate an ECDSA signature.	-	-	-	MX	-	-
		PKA_ModularExponentiation	How to use the low-layer PKA API to execute RSA modular exponentiation.	-	-	-	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	-	-
		PWR_OptimizedRunMode	How to increase/decrease frequency and V_{CORE} and enter/exit the Low-power Run mode.	-	-	-	MX	-	-
		PWR_SMPS_16MHZ_HSI	How to use the SMPS PWR regulator with system clock at 16 MHz with HSI.	-	-	-	MX	-	-
		PWR_SMPS_64MHZ_HSI_PLL	How to use the SMPS PWR regulator with system clock at 64 MHz with HSI and PLL.	-	-	-	MX	-	-
	RCC	RCC_OutputSystemClockOnMCO	How to configure the MCO pin (PA8) to output the system clock.	-	-	-	MX	-	-
		RCC_UseHSEasSystemClock	How to use of the RCC LL API to start the HSE and use it as system clock.	-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Examples_LL	RCC	RCC_UseHSI_PLLasSystemClock	How to modify the PLL parameters at runtime.	-	MX	MX	MX	-	-
	RNG	RNG_GenerateRandomNumbers	How to configure the RNG to generate 32-bit long random numbers.	-	-	-	MX	-	-
		RNG_GenerateRandomNumbers_IT	How to configure the RNG to generate 32-bit long random numbers using interrupts.	-	-	-	MX	-	-
	RTC	RTC_Alarm_Init	How to configure the RTC LL API to configure and generate an alarm using the RTC peripheral. The peripheral initialization uses the LL initialization function.	-	MX	-	MX	-	-
		RTC_Calendar_Init	How to configure the LL API to set the RTC calendar. The peripheral initialization uses LL unitary service functions to optimize performance and size.	-	-	-	MX	-	-
		RTC_ExitStandbyWithWakeUpTimer_Init	How to periodically enter and wake up from Standby mode using to the RTC wake-up timer (WUT).	-	MX	-	MX	-	-
		RTC_Tamper_Init	How to configure the tamper using the RTC LL API. The peripheral initialization uses LL unitary service functions to optimize performance and size.	-	MX	-	MX	-	-
		RTC_TimeStamp_Init	How to configure the timestamp using the RTC LL API. The peripheral initialization uses LL unitary service functions to optimize performance and size.	-	MX	-	MX	-	-
	SPI	SPI_OneBoard_HalfDuplex_DMA_Init	How to configure the 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 STM32WBAxx SPI LL API. The peripheral initialization uses the LL initialization function to demonstrate LL init usage.	-	-	-	MX	-	-
		SPI_OneBoard_HalfDuplex_IT_Init	How to configure the 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 STM32WBAxx SPI LL API. The peripheral initialization uses LL unitary service functions to optimize performance and size.	-	-	-	MX	-	-
		SPI_TwoBoards_FullDuplex_DMA_Master_Init	How to perform data buffer transmission and reception via SPI using DMA mode. This example is based on the STM32WBAxx SPI LL API. The peripheral initialization uses LL unitary service functions to optimize performance and size.	-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Examples_LL	SPI	SPI_TwoBoards_FullDuplex_DMA_Slave_Init	How to perform data buffer transmission and reception via SPI using DMA mode. This example is based on the STM32WBAxx SPI LL API. The peripheral initialization uses LL unitary service functions to optimize performance and size.	-	-	-	MX	-	-
		SPI_TwoBoards_FullDuplex_IT_Master_Init	How to perform data buffer transmission and reception via SPI using interrupt mode. This example is based on the STM32WBAxx SPI LL API. The peripheral initialization uses LL unitary service functions to optimize performance and size.	-	-	-	MX	-	-
		SPI_TwoBoards_FullDuplex_IT_Slave_Init	How to perform data buffer transmission and reception via SPI using interrupt mode. This example is based on the STM32WBAxx SPI LL API. The peripheral initialization uses LL unitary service functions to optimize performance and size.	-	-	-	MX	-	-
	TIM	TIM_BreakAndDeadtime_Init	How to configure the TIMER 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 deadtime configuration.	-	-	-	MX	-	-
		TIM_InputCapture_Init	How to use of the TIMER 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 STM32WBAxx TIM LL API. The peripheral initialization uses LL unitary service functions to optimize performance and size.	-	-	-	MX	-	-
		TIM_OnePulse_Init	How to configure a TIMER to generate a positive pulse in output compare mode with a length of t_{PULSE} and after a delay of t_{DELAY}. This example is based on the STM32WBAxx TIM LL API. The peripheral initialization uses the LL initialization function to demonstrate LL Init.	-	-	-	MX	-	-
		TIM_OutputCompare_Init	How to configure the TIMER peripheral to generate an output waveform in different output compare modes. This example is based on the STM32WBAxx TIM LL API. The peripheral initialization uses LL unitary service functions to optimize performance and size.	-	-	-	MX	-	-
		TIM_PWMOutput_Init	How to use of a TIMER peripheral to generate a PWM output signal and update the PWM duty cycle. This example is based on the STM32WBAxx TIM LL API. The peripheral initialization uses the LL initialization function to demonstrate LL Init.	-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Examples_LL	TIM	TIM_TimeBase_Init	How to configure the TIMER peripheral to generate a timebase. This example is based on the STM32WBAxx TIM LL API. The peripheral initialization uses LL unitary service functions to optimize performance and size.	-	-	-	MX	-	-
	USART	USART_Communication_Rx_IT_ Continuous_Init	How to configure GPIO and USART peripherals to continuously receive characters from a HyperTerminal (PC) in asynchronous mode, using interrupt mode. The peripheral initialization is done using LL unitary services functions to optimize performance and size.	-	-	-	MX	-	-
		USART_Communication_Rx_IT_ Continuous_VCP_Init	How to configure GPIO and USART peripherals to continuously receive characters from a HyperTerminal (PC) in asynchronous mode, using interrupt mode. The peripheral initialization is done using LL unitary services functions to optimize performance and size.	-	-	-	MX	-	-
		USART_Communication_Rx_IT_Init	How to configure GPIO and USART peripherals to receive characters from a HyperTerminal (PC) in asynchronous mode, using interrupt mode. The peripheral initialization is done using the LL initialization function to demonstrate LL init usage.	-	-	-	MX	-	-
		USART_Communication_Rx_IT_VCP_Init	How to configure GPIO and USART peripherals to receive characters from a HyperTerminal (PC) in asynchronous mode using interrupt mode. The peripheral initialization is done using the LL initialization function to demonstrate LL init usage.	-	-	-	MX	-	-
		USART_Communication_TxRx_DMA_Init	How to configure GPIO and USART peripherals to send characters asynchronously to/from a HyperTerminal (PC) in DMA mode. This example is based on the the STM32WBAxx USART LL API. The peripheral initialization is done using LL unitary services functions to optimize performance and size.	-	-	-	MX	-	-
		USART_Communication_Tx_IT_Init	How to configure GPIO and USART peripherals to send characters asynchronously to a HyperTerminal (PC) in interrupt mode. This example is based on the STM32WBAxx USART LL API. The peripheral initialization is done using LL unitary services functions to optimize performance and size.	-	-	-	MX	-	-
		USART_Communication_Tx_IT_VCP_Init	How to configure GPIO and USART peripherals to send characters asynchronously to a HyperTerminal (PC) in interrupt mode. This example is based on the STM32WBAxx USART LL API. The peripheral initialization is done using LL unitary services functions to optimize performance and size.	-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Examples_LL	USART	USART_Communication_Tx_Init	How to configure GPIO and USART peripherals to send characters asynchronously to a HyperTerminal (PC) in polling mode. If the transfer cannot complete within the allocated time, a timeout allows exiting from the sequence with a timeout error code. This example is based on the STM32WBAxx USART LL API. The peripheral initialization is done using LL unitary services functions to optimize performance and size.	-	-	-	MX	-	-
		USART_Communication_Tx_VCP_Init	How to configure GPIO and USART peripherals to send characters asynchronously to a HyperTerminal (PC) in polling mode. If the transfer cannot complete within the allocated time, a timeout allows exiting from the sequence with a timeout error code. This example is based on the STM32WBAxx USART LL API. The peripheral initialization is done using LL unitary services functions to optimize performance and size.	-	-	-	MX	-	-
		USART_HardwareFlowControl_Init	How to configure GPIO and USART peripherals to receive characters asynchronously from a HyperTerminal (PC) in interrupt mode with the hardware flow control feature enabled. This example is based on the STM32WBAxx USART LL API. The peripheral initialization uses LL unitary service functions to optimize performance and size.	-	-	-	MX	-	-
		USART_SyncCommunication_FullDuplex_DMA_Init	How to configure GPIO, USART, DMA, and SPI peripherals to transmit bytes between a USART and an SPI (in slave mode) in DMA mode. This example is based on the STM32WBAxx USART LL API. The peripheral initialization uses LL unitary service functions to optimize performance and size.	-	-	-	MX	-	-
		USART_SyncCommunication_FullDuplex_IT_Init	How to configure GPIO, USART, DMA, and SPI peripherals to transmit bytes between a USART and an SPI (in slave mode) in interrupt mode. This example is based on the STM32WBAxx USART LL API (the SPI uses the DMA to receive/transmit the characters sent from/received by the USART). The peripheral initialization uses LL unitary service functions to optimize performance and size.	-	-	-	MX	-	-
	UTILS	UTILS_ConfigureSystemClock	How to use the UTILS LL API to configure the system clock using PLL with HSI as the source clock.	-	-	MX	-	-	-
	WWDG	WWDG_RefreshUntilUserEvent_Init	How to configure 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 to optimize performance and size.	-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Examples_LL	**Total number of examples_ll**			**0**	**8**	**4**	**70**	-	**0**
Examples_MIX	ADC	ADC_SingleConversion_TriggerSW_IT	How to use an ADC to convert a single channel at each software start. The conversion is performed using the interrupt programming model.	-	-	-	MX	-	-
	CRC	CRC_PolynomialUpdate	How to use the CRC peripheral through the STM32WBAxx CRC HAL and LL API.	-	-	-	MX	-	-
	DMA	DMA_FLASHToRAM	How to use a DMA to transfer a word data buffer from flash memory to embedded SRAM through the STM32WBAxx DMA HAL and LL API. The LL API is used for performance improvement.	-	-	-	MX	-	-
	I2C	I2C_OneBoard_ComSlave7_10bits_IT	How to perform I2C data buffer transmission/reception between one controller and two targets with different address sizes (7 bits or 10 bits). This example uses the STM32WBAxx I2C HAL and LL API (LL API usage for performance improvement) and an interrupt.	-	-	-	MX	-	-
	PWR	PWR_STOP1	How to enter Stop 1 mode and wake up from this mode by using an external reset or wake-up interrupt (all the RCC function calls use RCC LL API for minimizing footprint and maximizing performance).	-	-	-	MX	-	-
	SPI	SPI_FullDuplex_ComPolling_Master	How to perform data buffer transmission/reception between two boards via SPI using polling mode.	-	-	-	MX	-	-
		SPI_FullDuplex_ComPolling_Slave	How to perform data buffer transmission/reception between two boards via SPI using polling mode.	-	-	-	MX	-	-
		SPI_HalfDuplex_ComPollingIT_Master	How to perform data buffer transmission/reception between two boards via SPI using polling (LL driver) and interrupt modes (HAL driver).	-	-	-	MX	-	-
		SPI_HalfDuplex_ComPollingIT_Slave	How to perform data buffer transmission/reception between two boards via SPI using polling (LL driver) and interrupt modes (HAL driver).	-	-	-	MX	-	-
	TIM	TIM_PWMInput	How to use the TIMER peripheral to measure an external signal frequency and duty cycle.	-	-	-	MX	-	-
	UART	UART_HyperTerminal_IT	How to use a UART to transmit data (transmit/receive) between a board and a HyperTerminal PC application in interrupt mode. This example describes how to use the USART peripheral through the STM32WBAxx UART HAL and LL API, the LL API being used for performance improvement.	-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Examples_MIX	UART	UART_HyperTerminal_TxPolling_RxIT	How to use a UART to transmit data (transmit/receive) between a board and a HyperTerminal PC application both in polling and interrupt modes. This example describes how to use the USART peripheral through the STM32WBAxx UART HAL and LL API, the LL API being used for performance improvement.	-	-	-	MX	-	-
	Total number of examples_mix			**0**	**0**	**0**	**12**	**0**	**0**
Applications	-	OpenBootloader	This application exploits the OpenBootloader middleware to demonstrate how to develop an IAP application and how to use it.	-	-	-	X	-	-
		ROT	The STM32WBA55G-DK1 ROT project is based on a recursive copy of `STM32WBA65I-DK1/Applications/ROT/` content and few changes.	-	X	X	X	-	-
	BLE	BLE_ApplicationInstallManager	The BLE_ApplicationInstallManager application, associated with a Bluetooth® LE application embedding OTA service, manages the over-the-air firmware update of a Bluetooth® LE application.	-	-	MX	MX	-	-
		BLE_Audio_GMAP_Central	How to use the Gaming Audio Profile profile in Unicast Client role (Unicast Game Gateway) and Broadcast Source role (Broadcast Game Sender) as specified by the Bluetooth® SIG.	MX	-	-	-	-	-
		BLE_Audio_GMAP_Peripheral	How to use the Gaming Audio Profile profile in Unicast Server role (Unicast Game Terminal) and Broadcast Sink role (Broadcast Game Receiver) as specified by the Bluetooth® SIG.	MX	-	-	-	-	-
		BLE_Audio_HAP_Central	How to use the Hearing Access Profile (HAP) in Unicast Client role (Hearing Aid Unicast Client / Hearing Aid Remote Controller / Immediate Alert Client) as specified by the Bluetooth® SIG.	MX	MX	-	-	-	-
		BLE_Audio_HAP_Peripheral	How to use the Hearing Access Profile in Unicast Server role (Hearing Aid role) as specified by the Bluetooth® SIG.	MX	MX	-	-	-	-
		BLE_Audio_PBP_Sink	How to use the Public Broadcast profile in Sink role as specified by the Bluetooth® SIG.	MX	MX	-	-	-	-
		BLE_Audio_PBP_Source	How to use the Public Broadcast profile in Source role as specified by the Bluetooth® SIG.	MX	MX	-	-	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Applications	BLE	BLE_Audio_TMAP_Central	How to use the Telephony and Media Audio Profile (TMAP) in Unicast Client role (Call Gateway and/or Unicast Media Sender) and Broadcast Source role (Broadcast Media Sender) as specified by the Bluetooth® SIG.	MX	MX	-	-	-	-
		BLE_Audio_TMAP_Peripheral	How to use the Telephony and Media Audio Profile (TMAP) in Unicast Server role (Call Terminal and/or Unicast Media Receiver), Broadcast Sink role (Broadcast Media Receiver) and Scan Delegator as specified by the Bluetooth® SIG.	MX	MX	-	-	-	-
		BLE_Beacon	How to advertise four types of beacon (tlm, uuid, url, iBeacon).	-	-	MX	MX	-	-
		BLE_BeaconHCI	How to advertise five types of beacon (Eddystone uid, Eddystone url, Eddystone uid+tlm, Eddystone url+tlm, iBeacon) using HCI.	-	-	MX	MX	-	-
		BLE_DataThroughput_Client	How to demonstrate point-to-point communication using the Bluetooth® LE component (as GATT server or GATT client).	-	-	MX	MX	-	-
		BLE_DataThroughput_Server	How to demonstrate point-to-point communication using the Bluetooth® LE component (as GATT server or GATT client).	-	-	MX	MX	-	-
		BLE_GenericHealth_ECG_ota	How to use the Generic Health Sensor profile with the over-the-air firmware update feature.	-	-	MX	-	-	-
		BLE_HID_Mouse	How to use the Human Interface Device over GATT profile for a mouse, as specified by the Bluetooth® LE SIG.	-	MX	-	-	-	-
		BLE_HR_P2PServer	How to use the Bluetooth® LE Heart Rate and Bluetooth® LE peer-to-peer server application.	-	-	-	-	-	MX
		BLE_HealthThermometer	How to use the Health Thermometer Sensor is a GATT server to measure the temperature and expose it via the Health Thermometer Service.	-	-	MX	MX	-	-
		BLE_HeartRate	How to use the Heart Rate Profile, widely used in fitness applications, to define the communication process between a GATT-server of a Heart Rate Sensor device (such as a wrist band) and a GATT-client Collector device (such as a smartphone or tablet).	-	MX	MX	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Applications	BLE	BLE_HeartRateFreeRTOS	How to use the Heart Rate Profile, widely used in fitness applications, to define the communication process between a GATT-server of a Heart Rate Sensor device (such as a wrist band) and a GATT-client Collector device (such as a smartphone or tablet).	-	-	MX	MX	-	-
		BLE_HeartRateThreadX	How to use the Heart Rate Profile, widely used in fitness applications, to define the communication process between a GATT-server of a Heart Rate Sensor device (such as a wrist band) and a GATT-client Collector device (such as a smartphone or tablet).	-	-	MX	MX	-	-
		BLE_HeartRate_ota	How to use the Bluetooth® LE Heart Rate with the over-the-air firmware update feature.	-	-	MX	MX	-	-
		BLE_Power_Central	How to measure the central power consumption using the Bluetooth® LE component.	-	-	MX	MX	-	-
		BLE_Power_Peripheral	How to measure the peripheral power consumption using the Bluetooth® LE component.	-	-	MX	MX	-	-
		BLE_Sensor	How to use the Bluetooth® LE sensor application.	-	-	-	-	New	MX
		BLE_SerialCom_Central	How to demonstrate point-to-point communication using the Bluetooth® LE component.	-	-	MX	MX	-	-
		BLE_SerialCom_Peripheral	How to demonstrate point-to-point communication using the Bluetooth® LE component.	-	-	MX	MX	-	-
		BLE_TransparentMode	How to communicate with the STM32CubeMonitor-RF tool using the transparent mode.	-	-	MX	MX	-	-
		BLE_TransparentMode_Ux_CDC	How to communicate with the STM32CubeMonitor-RF Tool using the transparent mode through USB Communication Device Class (CDC) ### __Keywords__ Connectivity, BLE, BLE protocol.	-	-	New	-	-	-
		BLE_p2pClient	How to use an STM32WBA device as a Bluetooth® LE central and GATT client.	-	-	MX	MX	-	-
		BLE_p2pClient_Ext	How to demonstrate a Bluetooth® LE scanner with connections from extended and legacy advertising.	-	-	MX	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Applications	BLE	BLE_p2pRouter	How to use an STM32WBA device acting simultaneously as a Bluetooth® LE central, a Bluetooth® LE peripheral, a GATT server, and a client.	-	-	MX	MX	-	-
		BLE_p2pServer	How to use the Bluetooth® LE Peer-to-Peer profile. This is a Generic Attribute Profile (GATT) based low-energy profile defined by STMicroelectronics with proprietary UUIDs(128 bits).	-	-	MX	MX	-	-
		BLE_p2pServerThreadX	How to use the Peer-to-Peer profile. This is a Generic Attribute Profile (GATT) based on low-energy profile defined by STMicroelectronics with proprietary UUIDs (128 bits).	-	-	MX	MX	-	-
		BLE_p2pServer_Ext	How to demonstrate how to use an STM32WBA device with the Bluetooth® LE basic plus stack version to utilize several extended advertising sets.	-	-	MX	MX	-	-
		BLE_p2pServer_Simplest	How to demonstrate point-to-point communication using Bluetooth® LE as simply as possible.	-	-	MX	MX	-	-
		BLE_p2pServer_TZ	This example is similar to the BLE_p2pServer with the TrustZone® being activated and the blue LED being connected to the secure side of the MCU.	-	-	MX	MX	-	-
		BLE_p2pServer_ota	How to use the Bluetooth® LE peer-to-peer server with the over-the-air firmware update feature.	-	-	MX	MX	-	-
	BLE_ Thread	BLE_HeartRate_Thread	How to use the concurrency mode Bluetooth® LE/ Thread with the Bluetooth® LE Heart Rate profile and COAP messages transmission over Thread.	-	-	X	-	-	-
		BLE_HeartRate_Thread_FreeRTOS	How to use the concurrency mode Bluetooth® LE/ Thread with the Bluetooth® LE Heart Rate profile and COAP messages transmission over Thread.	X	-	MX	-	-	-
		BLE_HeartRate_Thread_SED	How to use the Concurrency mode BLE/Thread with BLE Heart Rate profile and COAP messages transmission over Thread for a Sleepy End Device.	-	-	New	-	-	-
		BLE_HeartRate_Thread_SED_FreeRTOS	How to use the Concurrency mode BLE/Thread with BLE Heart Rate profile and COAP messages transmission over Thread for a Sleepy End Device.	New	-	New	-	-	-
	BLE_ Zigbee	BLE_HeartRate_Zigbee	How to use the concurrency mode Bluetooth® LE/ Zigbee® with the Bluetooth® LE Heart Rate profile and Zigbee® on/off toggle transmission.	-	-	X	-	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Applications	BLE_ Zigbee	ZigbeeDirect_OnOff	How to use Zigbee® Direct. This is a communication protocol designed to enable direct communication between Zigbee® devices and smartphones or tablets without the need for a dedicated Zigbee® hub or gateway.	-	-	X	-	-	-
		ZigbeeDirect_OnOff_Server_Router_Identify	How to use Zigbee® Direct. This is a communication protocol designed to enable direct communication between Zigbee® devices and smartphones or tablets, without the need for a dedicated Zigbee® hub or gateway.	-	-	X	-	-	-
	FileX	Fx_File_Edit_Standalone	This application provides an example of FileX stack usage on the NUCLEO-WBA65RI board, running in standalone mode (without ThreadX®). It demonstrates how to create a FAT file system on the internal SRAM using FileX.	-	-	MX	MX	-	-
	LPM	Tiny_lpm_3modes	This example is based on the tiny lpm utility. It shows how the sequencer handles the task waiting for an event.	-	-	-	X	-	-
	MbedTLS_ HW_ALT	Cipher_AES_CBC_EncryptDecrypt_HAL	How to use the PSA reference API to perform encryption and decryption using the AES CBC algorithm.	X	-	-	-	-	-
		Cipher_AES_GCM_AuthEncrypt_VerifDecrypt_ HAL	How to use the PSA reference API to perform authenticated encryption and verified decryption using the AES GCM algorithm.	X	-	-	-	-	-
		ECC_ECDH_SharedSecretGeneration_HAL	How to use the cryptographic reference API to establish a shared secret using the ECDH algorithm over SECP256 curve.	X	-	-	-	-	-
		ECC_ECDSA_SignVerify_HAL	How to use the PSA reference API to sign and verify a message using the ECDSA algorithm over SECP256 curve.	X	-	-	-	-	-
		Encrypted_ITS_KeyImport	How to use the PSA ITS alternative encrypted implementation to import an AES-CBC key and store it in user persistent storage. The key is encrypted by the ITS.	X	-	-	-	-	-
		Hash_SHA2_Digest_HAL	How to use the PSA reference API to digest a message using the SHA256 algorithm.	X	-	-	-	-	-
		MAC_HMAC_SHA2_AuthenticateVerify_HAL	How to use the PSA reference API to authenticate and verify a message using the HMAC SHA256 algorithm.	X	-	-	-	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Applications	MbedTLS_ HW_ALT	RSA_PKCS1v1.5_SignVerifyCRT_HAL	How to use the PSA reference API to sign and verify a message using the RSA PKCS#1 v1.5 compliant algorithm.	X	-	-	-	-	-
		RSA_PKCS1v1.5_SignVerify_HAL	How to use the PSA reference API to sign and verify a message using the RSA PKCS#1 v1.5 compliant algorithm.	X	-	-	-	-	-
		RSA_PKCS1v2.2_EncryptDecryptCRT_HAL	How to use the PSA reference API to encrypt and decrypt a message using the RSA PKCS#1 v2.2 compliant algorithm.	X	-	-	-	-	-
		RSA_PKCS1v2.2_EncryptDecrypt_HAL	How to use the PSA reference API to encrypt and decrypt a message using the RSA PKCS#1 v2.2 compliant algorithm.	X	-	-	-	-	-
		RSA_PKCS1v2.2_SignVerifyCRT_HAL	How to use the PSA reference API to sign and verify a message using the RSA PKCS#1 v2.2 compliant algorithm.	X	-	-	-	-	-
		RSA_PKCS1v2.2_SignVerify_HAL	How to use the PSA reference API to sign and verify a message using the RSA PKCS#1 v2.2 compliant algorithm.	X	-	-	-	-	-
	MbedTLS_ SW	Cipher_AES_CBC_EncryptDecrypt_MBED	How to use the PSA reference API to perform encryption and decryption using the AES CBC algorithm.	X	-	-	-	-	-
		Cipher_AES_GCM_AuthEncrypt_VerifDecrypt_ MBED	How to use the PSA reference API to perform authenticated encryption and verified decryption using the AES GCM algorithm.	X	-	-	-	-	-
		Cipher_ChachaPoly_AuthEnc_VerifDec_MBED	How to use the PSA reference API to perform authenticated encryption and verified decryption using the Chacha-Poly1305 algorithm.	X	-	-	-	-	-
		DRBG_RandomGeneration_MBED	How to use the PSA reference API to generate random numbers using the DRBG module.	X	-	-	-	-	-
		ECC_ECDH_SharedSecretGeneration_MBED	How to use the PSA reference API to establish a shared secret using the ECDH algorithm over SECP256 curve.	X	-	-	-	-	-
		ECC_ECDSA_SignVerify_MBED	How to use the PSA reference API to sign and verify a message using the ECDSA algorithm over SECP256 curve.	X	-	-	-	-	-
		Hash_SHA2_Digest_MBED	How to use the PSA reference API to digest a message using the SHA256 algorithm.	X	-	-	-	-	-
		MAC_AES_CMAC_AuthenticateVerify_MBED	How to use the PSA reference API to authenticate and verify a message using the AES CMAC algorithm.	X	-	-	-	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Applications	MbedTLS_SW	MAC_HMAC_SHA2_AuthenticateVerify_MBED	How to use the PSA reference API to authenticate and verify a message using the HMAC SHA256 algorithm.	X	-	-	-	-	-
		RSA_PKCS1v1.5_SignVerifyCRT_MBED	How to use the PSA reference API to sign and verify a message using the RSA PKCS#1 v1.5 compliant algorithm.	X	-	-	-	-	-
		RSA_PKCS1v1.5_SignVerify_MBED	How to use the PSA reference API to sign and verify a message using the RSA PKCS#1 v1.5 compliant algorithm.	X	-	-	-	-	-
		RSA_PKCS1v2.2_EncryptDecryptCRT_MBED	How to use the PSA reference API to encrypt and decrypt a message using the RSA PKCS#1 v2.2 compliant algorithm.	X	-	-	-	-	-
		RSA_PKCS1v2.2_EncryptDecrypt_MBED	How to use the PSA reference API to encrypt and decrypt a message using the RSA PKCS#1 v2.2 compliant algorithm.	X	-	-	-	-	-
		RSA_PKCS1v2.2_SignVerifyCRT_MBED	How to use the PSA reference API to sign and verify a message using the RSA PKCS#1 v2.2 compliant algorithm.	X	-	-	-	-	-
		RSA_PKCS1v2.2_SignVerify_MBED	How to use the PSA reference API to sign and verify a message using the RSA PKCS#1 v2.2 compliant algorithm.	X	-	-	-	-	-
	Phy_802_15_4	Phy_Cli	This dedicated application enables the control and test of the 802_15_4 radio via a command-line interface or the STM32CubeMonitor-RF.	-	-	X	X	-	-
	ROT	OEMiROT_Appli	This project provides an OEMiROT boot path fully secure application example. The boot is performed through the OEMiROT boot path after checking the authenticity and integrity of the project firmware and project data images.	X	-	-	-	-	-
		OEMiROT_Appli_TrustZone	This project provides an OEMiROT boot path application example. The boot is performed through the OEMiROT boot path after checking the authenticity and integrity of the project firmware and project data images.	X	-	-	-	-	-
		OEMiROT_Boot	This project provides an OEMiROT example. The OEMiROT boot path performs the authenticity and the integrity check of the project firmware and data images.	X	-	-	-	-	-
	Sequencer	Sequencer_gpio_toggle	This example is based on the sequencer utilities. It shows how to use a sequencer task to toggle a GPIO.	-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Applications	Sequencer	Sequencer_gpio_toggle_lowpower	This example is based on the sequencer utilities.	-	-	-	MX	-	-
		Sequencer_task_pauseresume	This example is based on the sequencer utilities. It shows how the sequencer handles the task pause/ resume mechanism.	-	-	-	MX	-	-
		Sequencer_task_prio	This example is based on the sequencer utility. It shows how the sequencer manages the task priority.	-	-	-	MX	-	-
		Sequencer_task_waitevent	This example is based on the sequencer utility. It shows how the sequencer handles the task waiting for an event.	-	-	-	MX	-	-
	Thread	Thread_Cli_Cmd_FTD	How to control the Thread stack via Cli commands.	-	-	New	-	-	-
		Thread_Cli_Cmd_LTD	How to control the Thread stack via Cli commands.	-	-	New	-	-	-
		Thread_Coap_Generic	How to build a Thread application based on Coap messages.	-	-	MX	MX	-	X
		Thread_Coap_Generic_FreeRTOS	How to build a Thread application based on Coap messages.	-	-	MX	-	-	-
		Thread_Coap_Generic_ThreadX	How to build a Thread application based on Coap messages.	-	-	MX	-	-	-
		Thread_Coap_OTA	How to build a Thread application based on Coap messages.	-	-	MX	-	-	-
		Thread_Commissioning	How to build a Thread application based on Coap messages.	-	-	X	-	-	-
		Thread_OTA_Client	How to update over-the -air (OTA) firmware application and Copro wireless binary using Thread (server side).	-	-	MX	-	-	-
		Thread_OTA_Server	How to update the over-the-air (OTA) firmware application and Copro wireless binary using Thread (server side).	-	-	MX	-	-	-
		Thread_SED_Coap_Multicast	How to build a Thread sleepy end device application based on Coap messages.	-	-	MX	-	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Applications	Thread	Thread_Udp	How to transfer data using UDP.	-	-	MX	-	-	-
	ThreadX	Tx_CMSIS_Wrapper	This application provides an example of CMSIS RTOS adaptation layer for Eclipse® ThreadX®. It shows how to develop an application using the CMSIS RTOS 2 APIs.	-	-	-	X	-	-
		Tx_FreeRTOS_Wrapper	This application provides an example of Eclipse® 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 Eclipse® ThreadX® stack usage. It shows how to develop an application using the ThreadX® low-power feature.	-	-	MX	MX	-	-
		Tx_MPU	This application provides an example of Eclipse® ThreadX® stack usage. It shows how to develop an application using the ThreadX® module feature.	-	-	X	X	-	-
		Tx_SecureLEDToggle_TrustZone	This application provides an example of Eclipse® ThreadX® stack usage. It shows how to develop an application using the ThreadX® when the TrustZone® feature is enabled (TZEN = 1).	-	-	MX	MX	-	-
		Tx_Thread_Creation	This application provides an example of Eclipse® ThreadX® stack usage. It shows how to develop an application using the ThreadX® thread management APIs.	-	-	MX	MX	-	-
		Tx_Thread_MsgQueue	This application provides an example of Eclipse® ThreadX® stack usage. It shows how to develop an application using the ThreadX® message queue APIs.	-	-	MX	MX	-	-
		Tx_Thread_Sync	This application provides an example of Eclipse® ThreadX® stack usage. It shows how to develop an application using the ThreadX® synchronization APIs.	-	-	MX	MX	-	-
	USBX	Ux_Device_CDC_ACM	This application provides an example of USBX stack usage on the NUCLEO-WBA65RI board. It shows how to develop a USB Device communication Class (CDC_ACM) application.	-	-	MX	-	-	-
		Ux_Device_HID_Standalone	This application provides an example of USBX stack usage on the NUCLEO-WBA65RI board. It shows how to develop a USB Device Human Interface (HID) mouse based bare metal application.	-	-	MX	-	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Applications	USBX	Ux_Host_HID	This application provides an example of USBX stack usage.	-	-	MX	-	-	-
		Ux_Host_HID_Standalone	This application provides an example of USBX stack usage.	-	-	MX	-	-	-
	Zigbee	Zigbee_Basic_Client_Coord	How to use the basic attributes of an on/off cluster on a device acting as a client within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_Basic_Server_Router	How to use the basic attributes of an on/off cluster on a device acting as a server within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_Diagnostic_Client_Coord	How to use the diagnostic attributes of an on/off cluster on a device acting as a server within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_Diagnostic_Server_Router	How to use the diagnostic attributes of an on/off cluster on a device acting as a client within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_DoorLock_Client_Router	How to use the door-lock cluster on a device acting as a client within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_DoorLock_Server_Coord	How to use the door-lock cluster on a device acting as a server within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_FindBind_Coord	How to use the finding and binding feature on a device acting as a server with coordinator role within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_FindBind_IAS_Router	How to use the finding-and-binding feature on a device using an IAS cluster, and acting as a client with router role within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_FindBind_OnOff_Router	How to use the finding-and-binding feature on a device acting as a client with router role within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_Messaging_Client_Coord	How to use the messaging cluster on a device acting as a coordinator within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_Messaging_Client_Coord_R22-SE	How to use the messaging cluster on a device acting as a coordinator within a centralized Zigbee® network.	-	-	-	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Applications	Zigbee	Zigbee_Messaging_Server_Router	How to use the messaging cluster on a device acting as a router within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_Messaging_Server_Router_R22-SE	How to use the messaging cluster on a device acting as a router within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_MeterId_Client_Coord	How to use the meter identification cluster on a device acting as a client within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_MeterId_Server_Router	How to use the meter identification cluster on a device acting as a server within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_OTA_Client_Router	How to use the OTA cluster on a device acting as a client within a centralized Zigbee® network.	-	-	-	X	-	-
		Zigbee_OTA_Server_Coord	How to use the OTA cluster on a device acting as a Server within a centralized Zigbee® network.	-	-	-	X	-	-
		Zigbee_OccupSensing_Client_Coord	How to use the occupancy sensing on a device acting as a client within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_OccupSensing_Server_Router	How to use the occupancy sensing on a device acting as a server within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_OnOff_Client_Distrib	How to use the on/off cluster on a device acting as a client within a distributed Zigbee® network.	-	-	MX	MX	-	-
		Zigbee_OnOff_Client_Router	How to use the on/off cluster on a device acting as a client within a centralized Zigbee® network.	MX	MX	MX	MX	-	MX
		Zigbee_OnOff_Client_Router_FreeRtos	How to use the on/off cluster on a device acting as a client within a centralized Zigbee® network.	-	MX	-	MX	-	-
		Zigbee_OnOff_Client_Router_OTA	How to use the OTA cluster on a device acting as a client within a centralized Zigbee® network.	-	-	-	X	-	-
		Zigbee_OnOff_Client_Router_ThreadX	How to use the on/off cluster on a device acting as a client within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_OnOff_Client_SED	How to use the on/off cluster on a device acting as an ED client within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_OnOff_Server_Coord	How to use the on/off cluster on a device acting as a server within a centralized Zigbee® network.	MX	MX	MX	MX	-	MX

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Applications	Zigbee	Zigbee_OnOff_Server_Coord_FreeRtos	How to use the on/off cluster on a device acting as a server within a centralized Zigbee® network.	-	MX	-	MX	-	-
		Zigbee_OnOff_Server_Coord_ThreadX	How to use the on/off cluster on a device acting as a cerver within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_OnOff_Server_Distrib	How to use the on/off cluster on a device acting as a client within a distributed Zigbee® network.	-	-	-	MX	-	-
		Zigbee_PowerConfig_Client_Coord	How to use the power configuration cluster on a device acting as a coordinator within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_PowerConfig_Server_SED	How to use the power configuration cluster on a device acting as a SED within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_PressMeas_Client_Coord	How to use the pressure measurement cluster on a device acting as a client within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_PressMeas_Client_Coord_Persist	How to use the pressure measurement cluster on a device acting as a client within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_PressMeas_Server_Router	How to use the pressure measurement cluster on a device acting as a server within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_PressMeas_Server_Router_Persist	How to use the pressure measurement cluster on a device acting as a server within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_TempMeas_Client_Coord	How to use the temperature measurement cluster on a device acting as a client within a centralized Zigbee® network.	-	-	MX	MX	-	-
		Zigbee_TempMeas_Server_Router	How to use the temperature measurement cluster on a device acting as a server within a centralized Zigbee® network.	-	-	-	MX	-	-
		Zigbee_TempMeas_Server_SED	How to use the temperature measurement cluster on a device acting as a server within a centralized Zigbee® network.	-	-	MX	MX	-	-
		Zigbee_Thermostat_Client_Coord	How to use the thermostat cluster on a device acting as a server within a centralized Zigbee® network.	MX	MX	-	MX	-	-

Level	Module name	Project name	Description	STM32WBA65I-DK1	STM32WBA55G-DK1	NUCLEO- WBA65RI	NUCLEO- WBA55CG	B-WBA6M-WPAN	B-WBA5M-WPAN
Applications	Zigbee	Zigbee_Thermostat_Server_Router	How to use the thermostat cluster on a device acting as a server within a centralized Zigbee® network.	MX	MX	-	MX	-	-
	Total number of applications			**45**	**15**	**61**	**83**	**1**	**5**
ROT_Provisioning	-	OEMiROT	This section provides an overview of the available scripts for OEMiROT boot path.	X	-	-	-	-	-
		OEMiRoT_OEMuRoT	This section provides an overview of the available scripts for OEMiROT_OEMuROT boot path.	X	-	-	-	-	-
	Total number of applications			**2**	**0**	**0**	**0**	**0**	**0**
Total number of projects				**53**	**42**	**96**	**284**	**4**	**8**

2 Reference documents

The reference documents are available on www.st.com/stm32cubefw:

- Latest release of STM32CubeWBA firmware package
- *Getting started with STM32CubeWBA for STM32WBA series* (UM3131)
- *Description of STM32WBAxx HAL and LL drivers* (UM3140)

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

Revision history

Table 2. **Document revision history**

Date	Version	Changes
6-Mar-2023	1	Initial release.
29-Jun-2023	2	Section 1 STM32CubeWBA examples updated.
28-Mar-2024	3	Section 1 STM32CubeWBA examples updated.
10-Jul-2024	4	Section 1 STM32CubeWBA examples updated.
27-Nov-2024	5	Section 1 STM32CubeWBA examples updated.
25-Feb-2025	6	Section 1 STM32CubeWBA examples updated.
24-Jun-2025	7	Section 1 STM32CubeWBA examples updated.
12-Nov-2025	8	Section 1: STM32CubeWBA examples updated.

Contents

1 STM32CubeWBA examples 2
2 Reference documents 35
Revision history 36

List of tables

Table 1. STM32CubeWBA firmware examples . . . 3
Table 2. Document revision history . . . 36