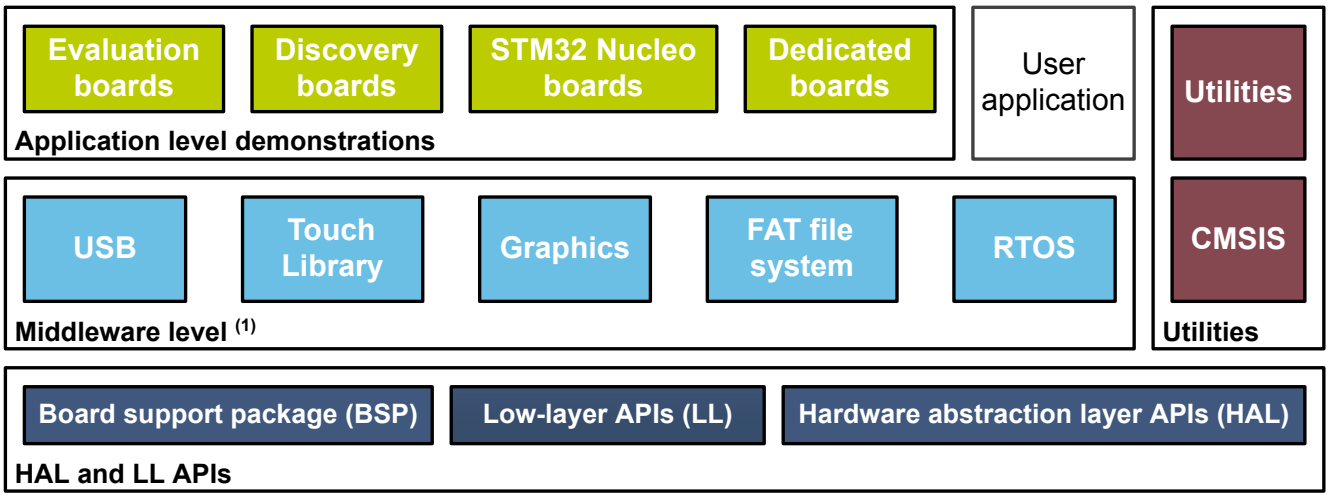
STM32Cube firmware examples for STM32L4 Series and STM32L4+ Series

Introduction

The STM32CubeL4 MCU Package is delivered with a rich set of examples running on STMicroelectronics boards. The examples are organized by board and provided with preconfigured projects for the main supported toolchains (refer to Figure 1).

Figure 1. STM32CubeL4 firmware components



(1) The set of middleware components depends on the product Series.

1 Reference documents

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

- Latest release of the STM32CubeL4 MCU Package for the 32-bit microcontrollers in the STM32L4 Series and STM32L4+ Series based on the Arm® Cortex®-M processor
- *Getting started with the STM32CubeL4 firmware package for STM32L4 Series and STM32L4+ Series* (UM1860)
- *Description of STM32L4/L4+ HAL and low-layer drivers* (UM1884)
- *STM32Cube USB Host library* (UM1720)
- *STM32Cube USB Device library* (UM1734)
- *Developing Applications on STM32Cube with FatFS* (UM1721)
- *Developing applications on STM32Cube with RTOS* (UM1722)
- *STM32CubeL4 Nucleo demonstration firmware* (UM1916)
- *STM32CubeL4 demonstration firmware for 32L476GDISCOVERY discovery kit* (UM1919)
- *STM32CubeL4 demonstration firmware for STM32L476G-EVAL board* (UM1937)
- *STM32CubeL4 demonstration firmware for 32L496GDISCOVERY kit* (UM2145)

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

2 STM32CubeL4 examples

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

- **Examples**

  These examples use only the HAL and BSP drivers (middleware not used). Their objective is to demonstrate the product/peripherals features and usage. They are organized per peripheral (one folder per peripheral, such as TIM). Their complexity level ranges from the basic usage of a given peripheral (such as PWM generation using timer) to the integration of several peripherals (such as 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 use only the LL drivers (HAL drivers and middleware components not used). They offer an optimum implementation of typical use cases of the peripheral features and configuration sequences. The examples are organized per peripheral (one folder for each peripheral, such as TIM) and run exclusively on Nucleo board.

- **Examples_MIX**

  These examples use only HAL, BSP and LL drivers (middleware components are not used). They aim at demonstrating how to use both HAL and LL APIs in the same application to combine the advantages of both APIs:

  – HAL offers high-level function-oriented APIs with high portability level by hiding product/IPs complexity for end users.

  – LL provides low-level APIs at register level with better optimization.

  The examples are organized per peripheral (one folder for each peripheral, such as TIM) and run exclusively on 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, such as USB Host) or by product feature that require high-level firmware bricks (such as Audio). The integration of applications that use several middleware stacks is also supported.

- **Demonstrations**

  The demonstrations aim at integrating and running the maximum number of peripherals and middleware stacks to showcase the product features and performance.

- **Template project**

  The template project is provided to allow the user to quickly build a firmware application using HAL and BSP drivers on a given board.

- **Template_LL project**

  The template LL projects are provided to allow the user to quickly build a firmware application using LL drivers on a given board.

The examples are located under `STM32Cube_FW_L4_VX.Y.Z\Projects\`. They all have the same structure:

- `\Inc` folder, containing all header files
- `\Src` folder, containing the sources code
- `\EWARM`, `\MDK-ARM`, `\SW4STM32` 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 "Development toolchains and compilers" and "Supported devices and evaluation boards" sections of the firmware package release notes to know more about the software/hardware environment used for the MCU Package development and validation. The correct operation of the provided examples is not guaranteed in other environments, for example when using different 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, pushbuttons, and others). 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 STM32CubeL4 MCU Package.

Table 1. STM32CubeL4 firmware examples

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Templates	-	Starter project	Reference template based on the STM32Cube HAL API that can be used to build any firmware application.	X	X	X	X	New	X	X	X	X	X	X	X	X	X	X	X	New	X	X
	Total number of templates: 6			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Templates_LL	-	Starter project	Reference template based on the STM32Cube LL API that can be used to build any firmware application.	X	X	X	X	New	X	X	X	X	X	X	X	X	X	X	X	New	X	X
	Total number of templates_LL: 6			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Examples	-	BSP	How to use the different BSP drivers of the board.	X	X	-	-	-	-	-	-	-	-	-	-	-	-	X	X	-	-	X
	ADC	ADC_AnalogWatchdog	How to use the ADC peripheral to perform conversions with an analog watchdog and out-of-window interrupts enabled.	-	X	-	-	New	-	X	-	-	-	-	-	X	-	-	-	-	-	-
		ADC_DMA_Transfer	How to configure and use the ADC to convert an external analog input and get the result using a DMA transfer through the HAL API.	-	X	-	X	New	-	X	-	-	X	-	X	X	-	-	-	-	-	-
		ADC_DifferentialMode	How to use an ADC peripheral to perform a conversion in differential mode between 2 ADC channels.	-	-	-	-	New	-	-	-	-	-	-	-	X	-	-	-	-	-	-
		ADC_DualModeInterleaved	How to use two ADC peripherals to perform conversions in dual interleaved mode.	-	X	-	-	-	-	X	-	-	-	-	-	X	-	-	-	-	-	-
		ADC_LowPower	How to use the ADC peripheral to perform conversions in ADC auto-wait low-power mode.	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-
		ADC_OverSampler	How to use an ADC peripheral in oversampling mode to increase resolution.	-	X	-	X	-	-	X	-	-	X	-	X	-	-	-	-	-	-	-
		ADC_Oversampling	This example describes how to use a ADC peripheral with ADC oversampling.	-	-	-	-	New	-	-	-	-	-	-	-	X	-	-	-	-	-	-
		ADC_RegularConversion_Interrupt	How to use the ADC in interrupt mode to convert data through the HAL API.	X	-	-	X	-	-	-	-	-	X	-	X	-	-	-	X	-	-	-
		ADC_RegularConversion_Polling	How to use the ADC in Polling mode to convert data through the HAL API.	-	X	-	X	-	-	X	-	-	X	-	X	-	-	-	-	-	-	-
		ADC_Regular_injected_groups	How to use the ADC peripheral to perform conversions using the two ADC groups: regular group for ADC conversions on the main stream, and injected group for ADC conversions limited to specific events (conversions injected into the main conversion stream).	-	X	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples	ADC	ADC_Sequencer	How to use the ADC peripheral with a sequencer to convert several channels.	-	X	-	X	New	-	X	-	-	-	-	-	X	-	-	-	-	-	-
	CAN	CAN_Networking	How to configure the CAN peripheral to send and receive CAN frames in normal mode.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	COMP	COMP_AnalogWatchdog	How to use a comparator peripheral to compare a voltage level to a threshold: the internal voltage reference (VREFINT), in interrupt mode.	-	X	-	-	-	-	X	-	-	-	-	-	X	-	-	-	-	-	-
		COMP_Interrupt	How to use a comparator peripheral to compare a voltage level applied on a GPIO pin to the the internal voltage reference (VREFINT), in interrupt mode.	-	X	-	-	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-
		COMP_PWMSignalControl	How to configure a comparator peripheral to automatically hold the TIMER PWM output in the safe state (low level) as soon as the comparator output is set to a high level.	-	-	-	-	-	-	-	-	-	-	-	-	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].	X	-	-	X	New	-	X	X	-	-	-	X	X	X	-	-	-	-	-
		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 8-bit data (bytes). 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.	X	-	-	X	New	-	X	X	-	-	-	X	X	X	-	-	-	-	-
		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).	X	X	-	X	New	-	X	X	-	X	-	X	X	X	-	-	-	-	X
		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).	X	X	-	X	New	-	X	X	-	X	-	X	X	X	-	-	-	-	X

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples	CRYP	CRYP_AESModes	How to use the CRYP peripheral to encrypt and decrypt data using AES in chaining modes (ECB, CBC, CTR).	-	X	-	-	-	-	X	-	-	X	-	-	-	-	-	-	-	-	-
		CRYP_AESModes_Suspension	How to use the CRYP AES peripheral to suspend then resume the AES ECB, CBC and CTR processing of a message in order to carry out the encryption or decryption of a higher-priority message.	-	X	-	-	-	-	X	-	-	X	-	-	-	-	-	-	-	-	-
		CRYP_DMA	How to use the CRYP peripheral to encrypt and decrypt data using the AES-128 algorithm with ECB chaining mode in DMA mode.	-	X	-	-	-	-	X	-	-	X	-	-	-	-	-	-	-	-	-
		CRYP_GCM_GMAC_CMAC_Modes	How to encrypt and decrypt data, and compute an authentication tag with GCM, GMAC, and CMAC AES algorithms.	-	X	-	-	-	-	X	-	-	X	-	-	-	-	-	-	-	-	-
		CRYP_GCM_GMAC_CMAC_Suspension	How to use the CRYP AES peripheral to suspend then resume the AES GCM, GMAC and CMAC processing of a message in order to carry out the encryption, decryption or authentication tag computation of a higher-priority message.	-	X	-	-	-	-	X	-	-	X	-	-	-	-	-	-	-	-	-
	Cortex	CORTEXM_MPU	Presentation of the MPU feature. This example configures a memory area as privileged read-only, and attempts to perform read and write operations in different modes.	-	X	-	-	-	-	X	X	-	X	-	X	X	-	-	-	-	-	-
		CORTEXM_ModePrivilege	How to modify the Thread mode privilege access and stack. Thread mode is entered on reset or when returning from an exception.	-	X	-	-	-	-	X	X	-	X	-	X	X	-	-	-	-	-	-
		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.	-	-	-	-	-	-	X	-	-	X	-	-	X	-	-	-	-	-	-
		CORTEXM_SysTick	How to use the default SysTick configuration with a 1 ms timebase to toggle LEDs.	-	X	-	-	-	-	X	X	-	X	-	X	X	-	-	-	-	-	-
	DAC	DAC_SignalsGeneration	How to use the DAC peripheral to generate several signals using the DMA controller.	X	X	-	-	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-
		DAC_SimpleConversion	How to use the DAC peripheral to do a simple conversion.	-	X	-	-	-	-	X	X	-	X	-	-	-	-	-	-	-	-	-
	DCMI	DCMI_CaptureMode	How to use the DCMI to interface with a camera module to continuously capture RGB565 images, crop them from size 320x240 to 240x240 then display the video stream on the LCD.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples	DCMI	DCMI_Preview	How to use the DCMI to interface with a camera module to continuously capture RGB565 images, crop them from size 320x240 to 240x240 then display the video stream on the LCD.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-
		DCMI_SnapshotMode	How to use the DCMI to interface with a camera module to capture a single RGB565 image and crop it from size 320x240 to 240x240, and once a full frame camera image is captured, display it on a 240x240 LCD in RGB565 format.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-
	DFSDM	DFSDM_AudioRecord	How to use the DFSDM HAL API to perform stereo audio recording. This example uses two MP34DT01 digital microphones mounted on the board.	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	X
		DFSDM_Thermometer	How to use the DFSDM HAL API to perform temperature measurements. This example uses the PT100 (thermistor) and STPMS2 (sigma-delta modulator) mounted on the board. The STPMS2 allows voltage and current values to be obtained from the PT100. The temperature value is thus deduced.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	DMA	DMAMUX_RequestGen	How to use the DMA with the DMAMUX request generator to generate DMA transfer requests upon an External line 13 rising edge signal.	-	-	-	X	New	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		DMAMUX_SYNC	How to use the DMA with the DMAMUX to synchronize a transfer with the LPTIM1 output signal. LPUART1 is used in DMA synchronized mode to send a countdown from 10 to 00, with a period of 2 seconds.	-	-	-	X	New	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		DMA_FLASHToRAM	This example provides a description of how to use a DMA channel to transfer a word data buffer from Flash memory to embedded SRAM through the HAL API.	-	X	-	X	New	-	-	-	-	X	-	X	X	X	-	-	-	X	-
	DMA2D	DMA2D_MemToMemWithBlending	How to configure the DMA2D peripheral in Memory-to-memory with blending transfer mode.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-
		DMA2D_MemToMemWithLCD	How to configure DMA2D peripheral in Memory-to-memory transfer mode and display the result on the LCD.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-
		DMA2D_MemToMemWithPFC	How to configure the DMA2D peripheral in Memory-to-memory transfer mode with pixel format conversion (PFC) mode.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-
		DMA2D_MemoryToMemory	How to configure the DMA2D peripheral in Memory-to-memory transfer mode.	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	New	X	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples	DMA2D	DMA2D_RegToMemWithLCD	How to configure DMA2D peripheral in Register-to-memory transfer mode and display the result on the LCD.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-
	DSI	DSI_CmdMode_SingleBuffer	How to use the embedded LCD DSI controller (using the LTDC and DSI Host IPs) to drive the round LCD mounted on-board.	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-
		DSI_ULPM_Data	How to use the embedded LCD DSI controller (using the LTDC and DSI Host IPs) to drive the KoD LCD mounted on-board and manage entry and exit in DSI ULPM mode on the data lane only. In this mode, the DSI PHY state machine enters a low-power state on the data lane, allowing some power saving when the LCD does not need to display. When the display is needed again, the DSI ULPM on the data lane is exited, and the display operates as before.	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-
		DSI_ULPM_DataClock	How to use the embedded LCD DSI controller (using the LTDC and DSI Host IPs) to drive the KoD LCD mounted on-board and manage entry and exit in DSI ULPM mode on data and clock lanes.	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-
	FIREWALL	FIREWALL_VolatileData_Executable	How to use the Firewall Peripheral to protect a volatile data segment and to define it as executable.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		FIREWALL_VolatileData_Shared	How to use the Firewall Peripheral to protect a code segment as well as volatile and non-volatile data segments.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
	FLASH	FLASH_DualBoot	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.	-	X	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	X	X
		FLASH_EraseProgram	How to configure and use the FLASH HAL API to erase and program the internal Flash memory.	X	X	-	X	New	-	X	X	-	X	-	X	X	X	-	X	-	X	X
		FLASH_FastProgram	How to configure and use the FLASH HAL API to erase and fast program the internal Flash memory.	X	X	-	X	New	-	X	X	-	X	-	-	-	-	-	X	-	X	X
		FLASH_WriteProtection	How to configure and use the FLASH HAL API to enable and disable the write protection of the internal Flash memory.	X	X	-	X	New	-	X	X	-	X	-	X	X	X	-	-	-	X	X
	FMC	FMC_NOR	How to configure the FMC controller to access the NOR memory.	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		FMC_SRAM	How to configure the FMC controller to access the SRAM memory.	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples	GPIO	GPIO_EXTI	How to configure external interrupt lines.	X	X	-	-	New	-	X	X	-	X	-	-	-	-	-	-	New	X	X
		GPIO_IOToggle	How to configure and use GPIOs through the HAL API.	X	X	-	-	New	-	X	X	-	X	-	X	-	X	-	-	New	X	X
	HAL	HAL_RegisterCallbacks_TIM	Register a callback function called every second based on TIM peripheral configuration to generate a timebase of one second with the corresponding interrupt request.	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-
		HAL_TimeBase_TIM	How to customize HAL using a general-purpose timer as main source of time base instead of Systick.	X	X	-	-	-	-	X	X	-	X	-	-	-	-	-	-	-	-	-
	HASH	HASH_HMAC_SHA1MD5	How to use the HASH peripheral to hash data with HMAC SHA-1 and HMAC MD5 algorithms.	-	-	-	X	-	-	X	-	-	-	-	-	-	-	-	-	New	-	-
		HASH_HMAC_SHA224MD5_IT_DMA_Suspension	How to suspend the HMAC digest computation when data are fed either under interruption or by DMA.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	New	-	-
		HASH_HMAC_SHA224SHA1_DMA_Suspension	How to suspend the HMAC digest computation when data are fed to the HASH unit with DMA.	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	New	-	-
		HASH_HMAC_SHA224SHA256_MultiBuffer_DMA	How to handle text messages larger than the maximum DMA transfer length. The input data are split into several buffers with sizes within the DMA limit, then fed successively to the HASH peripheral.	-	-	-	X	-	-	X	-	-	-	-	-	-	-	-	-	New	-	-
		HASH_HMAC_SHA256MD5_IT_Suspension	How to suspend the HMAC digest computation when data are fed in interrupt mode.	-	-	-	X	-	-	X	-	-	-	-	-	-	-	-	-	New	-	-
		HASH_SHA1MD5	How to use the HASH peripheral to hash data with SHA-1 and MD5 algorithms.	-	-	-	X	-	-	X	-	-	-	-	-	-	-	-	-	New	-	-
		HASH_SHA1MD5_DMA	How to use the HASH peripheral to hash data using SHA-1 and MD5 algorithms when data are fed to the HASH unit with DMA.	-	-	-	X	-	-	X	-	-	-	-	-	-	-	-	-	New	-	-
		HASH_SHA1SHA224_IT_Suspension	How to suspend the HASH peripheral when data are fed in interrupt mode.	-	-	-	X	-	-	X	-	-	-	-	-	-	-	-	-	New	-	-
		HASH_SHA1SHA256_IT_DMA_Suspension	How to suspend the HASH peripheral when data are fed either under interruption or with DMA.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	New	-	-
		HASH_SHA224SHA256_DMA	How to use the HASH peripheral to hash data with SHA224 and SHA256 algorithms.	-	-	-	X	-	-	X	-	-	-	-	-	-	-	-	-	New	-	-
		HASH_SHA256MD5_DMA_Suspension	How to suspend the HASH peripheral when data are fed to the HASH unit with DMA.	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	New	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples	I2C	I2C_EEPROM	How to handle I2C data buffer transmission/reception with DMA. In the example, the device communicates with an I2C EEPROM memory.	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		I2C_IOExpander	How to handle I2C data communication with the I/O expander device mounted on the evaluation board.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		I2C_TwoBoards_AdvComIT	How to handle I2C data buffer transmission/reception between two boards, using an interrupt.	-	X	-	X	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-
		I2C_TwoBoards_ComDMA	How to handle I2C data buffer transmission/reception between two boards, via DMA.	-	X	-	X	-	-	X	X	-	X	-	X	X	-	-	-	-	-	-
		I2C_TwoBoards_ComIT	How to handle I2C data buffer transmission/reception between two boards, using an interrupt.	-	X	-	X	-	-	X	X	-	X	-	X	X	-	-	-	-	-	-
		I2C_TwoBoards_ComPolling	How to handle I2C data buffer transmission/reception between two boards, in polling mode.	-	X	-	X	-	-	X	X	-	X	-	-	X	-	-	-	-	-	-
		I2C_TwoBoards_RestartAdvComIT	How to perform multiple I2C data buffer transmission/reception between two boards, in interrupt mode and with restart condition.	-	X	-	X	-	-	X	X	-	X	-	X	X	-	-	-	-	-	-
		I2C_TwoBoards_RestartComIT	How to handle single I2C data buffer transmission/reception between two boards, in interrupt mode and with restart condition.	-	X	-	X	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-
		I2C_WakeUpFromStop	How to handle I2C data buffer transmission/reception between two boards, using an interrupt when the device is in Stop mode.	-	X	-	-	-	-	X	X	-	-	-	X	-	-	-	-	-	-	-
		I2C_WakeUpFromStop2	How to handle I2C data buffer transmission/reception between two boards, using an interrupt when the device is in Stop 2 mode.	-	X	-	-	-	-	X	X	-	X	-	-	X	-	-	-	-	-	-
	IWDG	IWDG_Reset	How to handle the IWDG reload counter and simulate a software fault that generates an MCU IWDG reset after a preset laps of time.	X	X	-	-	-	-	X	X	-	X	-	-	X	-	-	-	-	-	-
		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.	X	X	-	-	-	-	X	X	-	X	-	-	X	-	-	-	-	-	-
	LCD	LCD_Blink_Frequency	How to use the embedded LCD glass controller and how to set the LCD blink mode and blinking frequency.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		LCD_SegmentsDrive	How to use the embedded LCD glass controller to drive the on-board LCD glass by Pacific Display Devices.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples	LPTIM	LPTIM_PWMExternalClock	How to configure and use, through the HAL LPTIM API, the LPTIM peripheral using an external counter clock, to generate a PWM signal at the lowest power consumption.	X	X	-	-	-	-	X	X	-	X	-	-	-	-	-	-	-	-	-
		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.	X	X	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-
		LPTIM_PulseCounter	How to configure and use, through the LPTIM HAL API, the LPTIM peripheral to count pulses.	X	X	-	-	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-
		LPTIM_RepetitionCounter	How to configure and use LPTIM repetition counter to update the autoreload counter upon an update event.	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	New	-	-
		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 mode.	X	X	-	-	-	-	X	-	-	X	-	X	X	-	-	X	New	-	-
	LPUART	LPUART_TwoBoards_ComIT	LPUART transmission (transmit/receive) in Interrupt mode between two boards.	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-
		LPUART_WakeUpFromStop	Configuration of an LPUART to wake up the MCU from Stop mode when a given stimulus is received.	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-
	LTDC	LTDC_ColorKeying	How to enable and use the LTDC color keying functionality.	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		LTDC_Display_1Layer	How to configure the LTDC peripheral to display a 480x272 RGB565 (16 bits/pixel) image on LCD using only one layer.	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		LTDC_Display_2Layers	How to configure the LTDC peripheral to display two layers at the same time.	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	OPAMP	OPAMP_PGA	How to use the built-in PGA mode (OPAMP programmable gain).	X	X	-	-	-	-	X	X	-	-	-	X	-	-	-	-	-	-	-
		OPAMP_STANDALONE	How to configure the OPAMP peripheral in standalone mode. The gain in this mode can be set externally (external gain setting mode).	-	X	-	-	-	-	X	-	-	-	-	X	-	-	-	-	-	-	-
	OSPI	OSPI_NOR_ExecuteInPlace	How to execute code from OSPI memory after code loading.	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		OSPI_NOR_ExecuteInPlace_DTR	How to execute code from OSPI memory after code loading.	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		OSPI_NOR_MemoryMapped	How to use a OSPI NOR memory in memory-mapped mode.	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	New	-	-
		OSPI_NOR_MemoryMapped_DTR	How to use a OSPI NOR memory in memory-mapped mode.	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples	OSPI	OSPI_NOR_ReadWrite_DMA	How to use a OSPI NOR memory in DMA mode.	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	New	-	-
		OSPI_NOR_ReadWrite_DMA_DTR	How to use a OSPI NOR memory in DMA mode.	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-
		OSPI_RAM_ExecuteInPlace	How to execute code from OSPI memory after code loading.	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	New	-	-
		OSPI_RAM_MemoryMapped	How to use a OSPI HyperRAM memory in memory-mapped mode.	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	New	-	-
		OSPI_RAM_ReadWrite_DMA	How to use a OSPI HyperRAM memory in DMA mode.	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	New	-	-
	PKA	PKA_ECCscalarMultiplication	How to use the PKA peripheral to execute ECC scalar multiplication. This allows generating a public key from a private key.	-	-	-	-	New	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		PKA_ECCscalarMultiplication_IT	How to use the PKA peripheral to execute ECC scalar multiplication. This allows generating a public key from a private key in interrupt mode.	-	-	-	-	New	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		PKA_ECDSA_Sign	How to compute a signed message regarding the Elliptic curve digital signature algorithm (ECDSA).	-	-	-	-	New	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		PKA_ECDSA_Sign_IT	How to compute a signed message regarding the Elliptic curve digital signature algorithm (ECDSA) in interrupt mode.	-	-	-	-	New	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		PKA_ECDSA_Verify	How to determine if a given signature is valid regarding the Elliptic curve digital signature algorithm (ECDSA).	-	-	-	-	New	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		PKA_ECDSA_Verify_IT	How to determine if a given signature is valid regarding the Elliptic curve digital signature algorithm (ECDSA) in interrupt mode.	-	-	-	-	New	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		PKA_ModularExponentiation	How to use the PKA peripheral to execute modular exponentiation. This allows ciphering/deciphering a text.	-	-	-	-	New	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		PKA_ModularExponentiationCRT	How to compute the Chinese Remainder Theorem (CRT) optimization.	-	-	-	-	New	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		PKA_ModularExponentiationCRT_IT	How to compute the Chinese Remainder Theorem (CRT) optimization in interrupt mode.	-	-	-	-	New	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		PKA_ModularExponentiation_IT	How to use the PKA peripheral to execute modular exponentiation. This allows ciphering/deciphering a text in interrupt mode.	-	-	-	-	New	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples	PKA	PKA_PointCheck	How to use the PKA peripheral to determine if a point is on a curve. This allows validating an external public key.	-	-	-	-	New	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		PKA_PointCheck_IT	How to use the PKA peripheral to determine if a point is on a curve. This allows validating an external public key.	-	-	-	-	New	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	PSSI	PSSI_Transmit	How to configure the PSSI using the HAL API to transmit data in blocking mode.	-	-	-	-	New	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		PSSI_Transmit_DMA	How to configure the PSSI using the HAL API to transmit data in DMA mode.	-	-	-	-	New	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	PWR	PWR_LPRUN	How to enter and exit the Low-power run mode.	-	-	-	X	New	-	X	X	-	X	-	X	-	X	-	-	-	-	-
		PWR_LPRUN_SRAM1	How to enter and exit the Low-power run mode.	-	-	-	X	New	-	X	X	-	X	-	X	-	-	-	-	-	-	-
		PWR_LPSLEEP	How to enter the Low-power sleep mode and wake up from this mode by using an interrupt.	-	-	-	X	New	-	X	X	-	X	-	X	-	X	-	-	-	-	-
		PWR_ModesSelection	How to configure the system to measure the current consumption in different low-power modes.	-	-	-	X	New	-	X	X	-	X	-	X	-	-	-	-	-	-	-
		PWR_RUN_SMPS	How to use SMPS in Run mode and access the power consumption gain obtained when the SMPS feature is used.	-	-	X	-	-	X	-	-	X	-	X	-	X	-	-	-	New	-	-
		PWR_SHUTDOWN	How to enter the Shutdown mode and wake up from this mode by using an external reset or the WKUP pin.	-	-	-	X	New	-	X	X	-	X	-	X	-	X	-	-	-	-	-
		PWR_SLEEP	How to enter the Sleep mode and wake up from this mode by using an interrupt.	-	-	-	X	New	-	X	X	-	X	-	X	-	X	-	-	-	-	-
		PWR_STANDBY	How to enter the Standby mode and wake up from this mode by using an external reset or the WKUP pin.	-	-	-	X	New	-	X	X	-	X	-	X	-	X	-	-	-	-	-
		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.	-	-	-	X	New	-	X	X	-	X	-	X	-	X	-	-	-	-	-
		PWR_STANDBY_SMPS	How to enter SMPS Standby mode and wake up from this mode using the wake-up pin.	-	-	X	-	-	X	-	-	X	-	X	-	X	-	-	-	New	-	-
		PWR_STOP0_SMPS	How to enter Stop 0 mode with or without SMPS enabled and to wake up from this mode using an interrupt.	-	-	X	-	-	X	-	-	X	-	X	-	X	-	-	-	New	-	-
		PWR_STOP1	How to enter the Stop 1 mode and wake up from this mode by using an interrupt.	-	-	-	X	New	-	X	X	-	X	-	X	-	X	-	-	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples	PWR	PWR_STOP1_RTC	How to enter the Stop 1 mode and wake up from this mode by using an interrupt from RTC wakeup timer.	-	-	-	X	New	-	X	X	-	X	-	X	-	X	-	-	-	-	-
		PWR_STOP2	How to enter the Stop 2 mode and wake up from this mode by using external reset or wakeup interrupt.	-	-	-	X	New	-	X	X	-	X	-	X	-	X	-	-	-	-	-
		PWR_STOP2_RTC	How to enter the Stop 2 mode and wake-up from this mode using an external reset or RTC wakeup timer.	-	-	-	X	New	-	X	X	-	X	-	X	-	X	-	-	-	-	-
	QSPI	QSPI_ExecuteInPlace	How to execute code from QSPI memory after code loading.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X
		QSPI_MemoryMapped	How to use a QSPI memory in memory-mapped mode.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X
		QSPI_PreInitConfig	How to configure the QSPI Peripheral in order to have access to external memory just after reset.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X
		QSPI_ReadWrite_DMA	How to use a QSPI memory in DMA mode.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X
		QSPI_ReadWrite_IT	How to use a QSPI memory in interrupt mode.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X
	RCC	RCC_CRS_Synchronization_IT	Configuration of the clock recovery service (CRS) in Interrupt mode, using the RCC HAL API.	-	-	-	-	-	-	X	-	-	X	-	X	-	X	-	-	-	-	-
		RCC_CRS_Synchronization_Polling	Configuration of the clock recovery service (CRS) in Polling mode, using the RCC HAL API.	-	-	-	-	-	-	X	-	-	X	-	X	-	X	-	-	-	-	-
		RCC_ClockConfig	Configuration of the system clock (SYSCLK) and modification of the clock settings in Run mode, using the RCC HAL API.	X	X	-	X	New	-	X	X	-	X	-	-	X	-	-	X	New	-	X
		RCC_LSEConfig	How to enable/disable at run time the propagation of the low-speed external(LSE) RC oscillator (about 32 KHz) to peripherals other than RTC, using the RCC HAL API.	-	-	-	-	New	-	-	-	-	-	-	-	X	-	-	-	-	-	-
		RCC_LSIConfig	How to enable/disable the low-speed internal (LSI) RC oscillator (about 32 KHz) at run time, using the RCC HAL API.	-	-	-	-	New	-	-	-	-	-	-	-	X	-	-	-	-	-	-
	RNG	RNG_Config	Configuration of the RNG using the HAL API. This example uses the RNG to generate 32-bit long random numbers.	-	-	-	-	New	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		RNG_MultiRNG	Configuration of the RNG using the HAL API. This example uses the RNG to generate 32-bit long random numbers.	X	X	-	X	New	-	X	-	-	X	-	X	X	-	-	-	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples	RNG	RNG_MultiRNG_IT	Configuration of the RNG using the HAL API. This example uses RNG interrupts to generate 32-bit long random numbers.	X	X	-	X	New	-	X	-	-	X	-	X	X	-	-	-	-	-	-
	RTC	RTC_Alarm	Configuration and generation of an RTC alarm using the RTC HAL API.	X	X	-	X	New	-	-	X	-	X	-	X	X	-	-	-	-	X	-
		RTC_Calendar	Configuration of the calendar using the RTC HAL API.	X	X	-	-	New	-	-	-	-	X	-	-	X	-	-	-	-	-	-
		RTC_InternalTimeStamp	Demonstration the internal timestamp feature using the RTC HAL API.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		RTC_LSI	Use of the LSI clock source autocalibration to get a precise RTC clock.	X	X	-	X	New	-	X	X	-	X	-	X	X	-	-	-	-	-	-
		RTC_LowPower_STANDBY	How to enter STANDBY mode and wake up from this mode using the RTC alarm event.	-	-	-	-	New	-	-	-	-	X	-	-	X	-	-	-	-	-	-
		RTC_Tamper	Configuration of the RTC HAL API to write/ read data to/from RTC Backup registers.	X	X	-	X	New	-	X	X	-	X	-	-	X	-	-	-	-	-	-
		RTC_TimeStamp	Configuration of the RTC HAL API to demonstrate the timestamp feature.	X	X	-	X	New	-	X	X	-	X	-	-	X	-	-	X	-	-	-
	SAI	SAI_AudioPlay	Use of the SAI HAL API to play an audio file in DMA circular mode and handle the buffer update.	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	X
	SMART CARD	SMARTCARD_T0	Firmware smartcard interface based on USART. The main purpose of this firmware example is to provide resources that ease the development of applications using the USART in Smartcard mode.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	SPI	SPI_FullDuplex_ComDMA	Data buffer transmission/reception between two boards via SPI using DMA.	-	-	-	X	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-
		SPI_FullDuplex_ComIT	Data buffer transmission/reception between two boards via SPI using Interrupt mode.	-	-	-	X	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-
		SPI_FullDuplex_ComPolling	Data buffer transmission/reception between two boards via SPI using Polling mode.	-	-	-	X	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-
		SPI_HalfDuplex_ComPolling	Data buffer half-duplex transmission/reception between two boards via SPI using Polling mode.	-	-	-	-	-	-	X	X	-	X	-	-	-	-	-	-	-	-	-
	SWPMI	SWPMI_Session	Configuration of the SWPMI peripheral to open a communication session with a SWP compliant card in no software buffer mode.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	TIM	TIM_DMA	Use of the DMA with TIMER Update request to transfer data from memory to TIMER Capture Compare Register 3 (TIMx_CCR3).	-	X	-	-	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples	TIM	TIM_DMABurst	How to update the TIMER channel 1 period and duty cycle using the TIMER DMA burst feature.	-	X	-	-	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-
		TIM_ExtTriggerSynchro	Synchronization of TIM peripherals in Cascade mode with an external trigger.	-	X	-	-	-	-	X	X	-	X	-	-	-	-	-	-	-	-	-
		TIM_InputCapture	How to use the TIM peripheral to measure an external signal frequency.	-	X	-	-	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-
		TIM_OCActive	Configuration of the TIM peripheral in Output Compare Active mode (when the counter matches the capture/compare register, the corresponding output pin is set to its active state).	-	X	-	-	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-
		TIM_OCInactive	Configuration of the TIM peripheral in Output Compare Inactive mode with the corresponding Interrupt requests for each channel.	-	X	-	-	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-
		TIM_OCToggle	Configuration of the TIM peripheral to generate four different signals at four different frequencies.	-	X	-	-	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-
		TIM_OnePulse	Use of the TIM peripheral to generate a single pulse when an external signal rising edge is received on the timer input pin.	-	X	-	-	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-
		TIM_PWMInput	How to use the TIM peripheral to measure the frequency and duty cycle of an external signal.	-	X	-	-	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-
		TIM_PWMOutput	Configuration of the TIM peripheral in PWM (pulse width modulation) mode.	X	X	-	-	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-
		TIM_TimeBase	Configuration of the TIM peripheral to generate a timebase of one second with the corresponding interrupt request.	-	X	-	-	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-
	TSC	TSC_BasicAcquisition_Interrupt	Use of he TSC to perform continuous acquisitions of one channel in Interrupt mode.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	UART	LPUART_WakeUpFromStop	Configuration of an LPUART to wake up the MCU from Stop mode when a given stimulus is received.	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		UART_HyperTerminal_DMA	UART transmission (transmit/receive) in DMA mode between a board and an HyperTerminal PC application.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		UART_LowPower_HyperTerminal_DMA	LPUART transmission (transmit/receive) in DMA mode between a board and an HyperTerminal PC application.	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		UART_Printf	Re-routing of the C library printf function to the UART.	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples	UART	UART_TwoBoards_ComDMA	UART transmission (transmit/receive) in DMA mode between two boards.	-	-	-	-	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-
		UART_TwoBoards_ComIT	UART transmission (transmit/receive) in Interrupt mode between two boards.	-	-	-	-	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-
		UART_TwoBoards_ComPolling	UART transmission (transmit/receive) in Polling mode between two boards.	-	-	-	-	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-
		UART_WakeUpFromStop	Configuration of an UART to wake up the MCU from Stop 1 mode when a given stimulus is received.	-	-	-	-	-	-	X	X	-	X	-	X	-	-	-	-	-	-	-
	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.	X	X	-	-	-	-	X	X	-	X	-	-	X	-	-	-	-	-	-
	Total number of examples: 716			**54**	**87**	**3**	**52**	**55**	**3**	**97**	**70**	**3**	**82**	**3**	**62**	**43**	**20**	**1**	**16**	**25**	**23**	**17**
Examples_LL	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.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		ADC_ContinuousConversion_TriggerSW	How to use an ADC peripheral to perform continuous ADC conversions on a channel, from a software start.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		ADC_ContinuousConversion_TriggerSW_Init	How to use an ADC peripheral to perform continuous ADC conversions on a channel, from a software start.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		ADC_ContinuousConversion_TriggerSW_LowPower	How to use an ADC peripheral with ADC low-power features.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		ADC_GroupsRegularInjected	How to use an ADC peripheral with both ADC groups (regular and injected) in their intended use cases.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		ADC_MultiChannelSingleConversion	How to use an ADC peripheral to convert several channels. ADC conversions are performed successively in a scan sequence.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		ADC_MultimodeDualInterleaved	How to use several ADC peripherals in multimode and interleaved mode.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		ADC_Oversampling	How to use an ADC peripheral with ADC oversampling.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		ADC_SingleConversion_Stop2_LP	How to use an ADC peripheral with a system configuration optimized for power consumption, to perform an acquisition at low frequency (between 1Hz and 1kHz).	-	-	-	-	-	-	X	X	-	-	-	-	-	X	-	-	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples_LL	ADC	ADC_SingleConversion_TriggerSW	How to use an ADC peripheral to perform a single ADC conversion on a channel at each software start. This example uses the polling programming model (for interrupt or DMA programming models, refer to other examples). This example is based on the STM32L4xx ADC LL API. The peripheral initialization is done using LL unitary service functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		ADC_SingleConversion_TriggerSW_DMA	How to use an ADC peripheral to perform a single ADC conversion on a channel, at each software start. This example uses the DMA programming model (for polling or interrupt programming models, refer to other examples). This example is based on the STM32L4xx ADC LL API. The peripheral initialization is done using LL unitary service functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		ADC_SingleConversion_TriggerSW_IT	How to use an ADC peripheral to perform a single ADC conversion on a channel, at each software start. This example uses the interrupt programming model (for polling or DMA programming models, refer to other examples). This example is based on the STM32L4xx ADC LL API. The peripheral initialization is done using LL unitary service functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		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 are indefinitely transferred by DMA into a table (circular mode). This example is based on the STM32L4xx ADC LL API. The peripheral initialization is done using LL unitary service functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		ADC_SingleConversion_TriggerTimer_DMA_LP	How to use an ADC peripheral with a system configuration optimized for power consumption, to perform an acquisition at medium frequency (between 10 Hz and 100 kHz).	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-
		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.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples_LL	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. This example is based on the STM32L4xx COMP LL API. The peripheral initialization is done using LL unitary service functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		COMP_CompareGpioVsVrefInt_IT_Init	How to use a comparator peripheral to compare a voltage level applied on a GPIO pin to the the internal voltage reference (VREFINT), in interrupt mode. This example is based on the STM32L4xx COMP LL API. The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		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).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		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. This example is based on the STM32L4xx COMP LL API. The peripheral initialization is done using LL unitary service functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
	CORTEX	CORTEX_MPU	Presentation of the MPU feature. This example configures a memory area as privileged read-only, and attempts to perform read and write operations in different modes.	-	-	-	-	-	-	X	X	-	-	-	-	X	-	-	-	-	-	-
	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).	-	-	-	-	-	-	X	X	-	-	-	-	X	X	-	-	-	-	-
		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).	-	-	-	-	New	-	X	X	-	-	-	-	X	X	-	-	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples_LL	CRS	CRS_Synchronization_IT	How to configure the clock recovery service in IT mode through the STM32L4xx CRS LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-
		CRS_Synchronization_Polling	How to configure the clock recovery service in polling mode through the STM32L4xx CRS LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-
	DAC	DAC_GenerateConstantSignal_TriggerSW	How to use the DAC peripheral to generate a constant voltage signal. This example is based on the STM32L4xx DAC LL API.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		DAC_GenerateConstantSignal_TriggerSW_LP	How to use the DAC peripheral to generate a constant voltage signal with the DAC low-power feature sample-and-hold. To be effective, a capacitor must be connected to the DAC channel output and the sample-and-hold timings must be tuned depending on the capacitor value. This example is based on the STM32L4xx DAC LL API. The peripheral initialization is done using LL unitary service functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		DAC_GenerateWaveform_TriggerHW	How to use the DAC peripheral to generate a voltage waveform from a digital data stream transfered by DMA. This example is based on the STM32L4xx DAC LL API. The peripheral initialization is done using LL unitary service functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		DAC_GenerateWaveform_TriggerHW_Init	How to use the DAC peripheral to generate a voltage waveform from a digital data stream transfered by DMA. This example is based on the STM32L4xx DAC LL API. The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
	DMA	DMA_CopyFromFlashToMemory	How to use a DMA channel to transfer a word data buffer from Flash memory to embedded SRAM. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size)..	-	-	-	-	-	-	X	X	-	-	-	-	X	X	-	-	-	-	-
		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.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples_LL	DMA2D	DMA2D_MemoryToMemory	How to configure the DMA2D peripheral in Memory-to-memory transfer mode. The example is based on the STM32L4xx DMA2D LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	New	X	-
	EXTI	EXTI_ToggleLedOnIT	How to configure the $moduleName$ and use GPIOs to toggle the user LEDs available on the board when a user button is pressed. It is based on the STM32L4xx LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		EXTI_ToggleLedOnIT_Init	How to configure the $moduleName$ and use GPIOs to toggle the user LEDs available on the board when a user button is pressed. This example is based on the STM32L4xx LL API. The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
	GPIO	GPIO_InfiniteLedToggling	How to configure and use GPIOs to toggle the on-board user LEDs every 250 ms. This example is based on the STM32L4xx LL API. The peripheral initialization is done using LL unitary service functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	X	-	-	-	-	-
		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 STM32L4xx LL API. The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	-	-	-	-	X	X	-	-	-	-	-	X	-	-	-	-	-
	I2C	I2C_OneBoard_AdvCommunication_DMAAndIT	How to exchange data between an I2C master device in DMA mode and an I2C slave device in interrupt mode. The peripheral is initialized with LL unitary service functions to optimize for performance and size.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		I2C_OneBoard_Communication_DMAAndIT	How to transmit data bytes from an I2C master device using DMA mode to an I2C slave device using interrupt mode. The peripheral is initialized with LL unitary service functions to optimize for performance and size.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		I2C_OneBoard_Communication_IT	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.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples_LL	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.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		I2C_OneBoard_Communication_PollingAndIT	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.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		I2C_TwoBoards_MasterRx_SlaveTx_IT	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.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		I2C_TwoBoards_MasterTx_SlaveRx	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.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		I2C_TwoBoards_MasterTx_SlaveRx_DMA	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.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		I2C_TwoBoards_WakeUpFromStop2_IT	How to handle the reception of a data byte from an I2C slave device in Stop2 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.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
	IWDG	IWDG_RefreshUntilUserEvent	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.	-	-	-	-	-	-	X	X	-	-	-	-	X	-	-	-	-	-	-
	LPTIM	LPTIM_PulseCounter	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 STM32L4xx LPTIM LL API. The peripheral initialization is done using LL unitary service functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples_LL	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 STM32L4xx LPTIM LL API. The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
	LPUART	LPUART_WakeUpFromStop2	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 unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		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.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
	OPAMP	OPAMP_PGA	How to use the OPAMP peripheral in PGA mode (programmable gain amplifier). To test the OPAMP, a voltage waveform is generated by the DAC and feeds the OPAMP input. This example is based on the STM32L4xx OPAMP LL API. The peripheral initialization is done using LL unitary service functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		OPAMP_PGA_Init	How to use the OPAMP peripheral in PGA mode (programmable gain amplifier). To test the OPAMP, a voltage waveform is generated by the DAC and feeds the OPAMP input. This example is based on the STM32L4xx OPAMP LL API. The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
	PKA	PKA_ECDSA_Sign	How to use the low-layer PKA API to generate an ECDSA signature.	-	-	-	-	New	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		PKA_ModularExponentiation	How to use the low-layer PKA API to execute RSA modular exponentiation.	-	-	-	-	New	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	PWR	PWR_EnterStandbyMode	How to enter the Standby mode and wake up from this mode by using an external reset or a wakeup interrupt.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		PWR_EnterStopMode	How to enter Stop 2 mode.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples_LL	PWR	PWR_LPRunMode_SRAM1	How to execute code in Low-power run mode from SRAM1..	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		PWR_OptimizedRunMode	How to increase/decrease frequency and VCORE and how to enter/exit the Low-power run mode.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
	RCC	RCC_HWAutoMSICalibration	How to use of the MSI clock source hardware autocalibration and LSE clock (PLL mode) to obtain a precise MSI clock.	-	-	-	-	New	-	X	X	-	-	-	-	X	-	-	-	-	-	-
		RCC_OutputSystemClockOnMCO	Configuration of MCO pin (PA8) to output the system clock.	-	-	-	-	New	-	X	X	-	-	-	-	X	-	-	-	-	-	-
		RCC_UseHSEasSystemClock	Use of the RCC LL API to start the HSE and use it as system clock.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		RCC_UseHSI_PLLasSystemClock	Modification of the PLL parameters in run time.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
	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).	-	-	-	-	-	-	X	X	-	-	-	-	X	-	-	-	-	-	-
		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).	-	-	-	-	-	-	X	X	-	-	-	-	X	-	-	-	-	-	-
	RTC	RTC_Alarm	Configuration of the RTC LL API to configure and generate an alarm using the RTC peripheral. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	New	-	X	X	-	-	-	-	X	-	-	-	-	-	-
		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.	-	-	-	-	New	-	X	X	-	-	-	-	X	-	-	-	-	-	-
		RTC_Calendar	Configuration of the LL API to set the RTC calendar. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	New	-	X	X	-	-	-	-	X	-	-	-	-	-	-
		RTC_ExitStandbyWithWakeUpTimer	Configuration of the RTC to wake up from Standby mode using the RTC Wakeup timer. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	New	-	X	X	-	-	-	-	X	-	-	-	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples_LL	RTC	RTC_ProgrammingTheWakeUpTimer	Configuration of the RTC to use the WUT. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-
		RTC_Tamper	Configuration of the Tamper using the RTC LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	New	-	X	X	-	-	-	-	X	-	-	-	-	-	-
		RTC_TimeStamp	Configuration of the Timestamp using the RTC LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
	SPI	SPI_OneBoard_HalfDuplex_DMA	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 STM32L4xx SPI LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		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 STM32L4xx SPI LL API. The peripheral initialization uses the LL initialization function to demonstrate LL init usage.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		SPI_OneBoard_HalfDuplex_IT	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 STM32L4xx SPI LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		SPI_TwoBoards_FullDuplex_DMA	Data buffer transmission and reception via SPI using DMA mode. This example is based on the STM32L4xx SPI LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		SPI_TwoBoards_FullDuplex_IT	Data buffer transmission and reception via SPI using Interrupt mode. This example is based on the STM32L4xx SPI LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples_LL	SWPMI	SWPMI_Loopback_MultiSWBuffer	Configuration of the SWPMI peripheral to start a communication using DMA multibuffers in Loopback mode. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		SWPMI_Loopback_MultiSWBuffer_Init	Configuration of the SWPMI peripheral to start a communication using DMA multibuffers in Loopback mode. The peripheral initialization uses the LL initialization function to demonstrate LL init usage.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		SWPMI_Loopback_NoSWBuffer	Configuration of the SWPMI peripheral to start a communication using No software buffer mode in Loopback mode. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
	TIM	TIM_BreakAndDeadtime	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.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		TIM_DMA	Use of the DMA with a timer update request to transfer data from memory to Timer Capture Compare Register 3 (TIMx_CCR3). This example is based on the STM32L4xx TIM LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		TIM_InputCapture	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 STM32L4xx TIM LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		TIM_OnePulse	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 STM32L4xx TIM LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples_LL	TIM	TIM_OutputCompare	Configuration of the TIM peripheral to generate an output waveform in different output compare modes. This example is based on the STM32L4xx TIM LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		TIM_PWMOutput	Use of a timer peripheral to generate a PWM output signal and update the PWM duty cycle. This example is based on the STM32L4xx TIM LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		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 STM32L4xx TIM LL API. The peripheral initialization uses LL initialization function to demonstrate LL init.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		TIM_TimeBase	Configuration of the TIM peripheral to generate a timebase. This example is based on the STM32L4xx TIM LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
	USART	USART_Communication_Rx_IT	Configuration of GPIO and USART peripherals to receive characters from an HyperTerminal (PC) in Asynchronous mode using an interrupt. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		USART_Communication_Rx_IT_Continuous	Configuration of GPIO and USART peripherals to continuously receive characters from an HyperTerminal (PC) in Asynchronous mode using an interrupt. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		USART_Communication_Rx_IT_Init	Configuration of GPIO and USART peripherals to receive characters from an HyperTerminal (PC) in Asynchronous mode using an interrupt. The peripheral initialization uses the LL initialization function to demonstrate LL init.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples_LL	USART	USART_Communication_Tx	Configuration of GPIO and USART peripherals to send characters asynchronously to an HyperTerminal (PC) in Polling mode. If the transfer could not be complete within the allocated time, a timeout allows to exit from the sequence with timeout error. This example is based on STM32L4xx USART LL API. The peripheral initialization uses LL unitary service functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		USART_Communication_TxRx_DMA	Configuration of GPIO and USART peripherals to send characters asynchronously to/from an HyperTerminal (PC) in DMA mode.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		USART_Communication_Tx_IT	Configuration of GPIO and USART peripheral to send characters asynchronously to HyperTerminal (PC) in Interrupt mode. This example is based on the STM32L4xx USART LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		USART_HardwareFlowControl	Configuration of GPIO and $COM_INSTANCE1_TYPE$ peripheral to receive characters asynchronously from an HyperTerminal (PC) in Interrupt mode with the Hardware Flow Control feature enabled. This example is based on STM32L4xx USART LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		USART_SyncCommunication_FullDuplex_DMA	Configuration of 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 STM32L4xx USART LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		USART_SyncCommunication_FullDuplex_IT	Configuration of 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 STM32L4xx USART LL API (the SPI uses the DMA to receive/transmit characters sent from/ received by the USART). The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples_LL	USART	USART_WakeUpFromStop1	Configuration of GPIO and STM32L4xx USART peripherals to receive characters on USART_RX pin and wake up the MCU from low-power mode. This example is based on the STM32L4xx USART LL API. The peripheral initialization uses LL unitary service functions for optimization purposes (performance and size).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
	UTILS	UTILS_ConfigureSystemClock	How to use UTILS LL API to configure the system clock using PLL with HSI as source clock.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		UTILS_ReadDeviceInfo	How to read UID, Device ID and Revision ID and save them into a global information buffer.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
	WWDG	WWDG_RefreshUntilUserEvent	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).	-	-	-	-	-	-	X	X	-	-	-	-	X	-	-	-	-	-	-
	Total number of examples_LL: 219			**0**	**0**	**0**	**0**	**10**	**0**	**94**	**91**	**0**	**0**	**0**	**0**	**15**	**7**	**0**	**0**	**1**	**1**	**0**
Examples_MIX	ADC	ADC_SingleConversion_TriggerSW_IT	How to use the ADC to perform a single ADC channel conversion at each software start. This example uses the interrupt programming model (for programming models in Polling or DMA mode, refer to other examples). This example is based on the STM32L4xx ADC HAL and LL API (LL API usage for performance improvement).	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-
	CRC	CRC_PolynomialUpdate	How to use the CRC peripheral through the STM32L4xx CRC HAL and LL API.	-	-	-	-	New	-	X	X	-	-	-	-	X	X	-	-	-	-	-
	DMA	DMA_FLASHToRAM	How to use a DMA to transfer a word data buffer from Flash memory to embedded SRAM through the STM32L4xx DMA HAL and LL API. The LL API is used for performance improvement.	-	-	-	-	-	-	X	X	-	-	-	-	X	X	-	-	-	-	-
	DMA2D	DMA2D_MemToMemWithLCD	How to configure the DMA2D peripheral in Memory-to-memory transfer mode and display the result on the LCD. The DMA2D LL APIs are used for performance improvement.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-
		DMA2D_MemToMemWithRBSwap	How to configure DMA2D peripheral in Memory-to-memory transfer mode with pixel format conversion and image blending, then display the result on LCD. The DMA2D LL APIs are used for performance improvement.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Examples_MIX	I2C	I2C_OneBoard_ComSlave7_10bits_IT	How to perform I2C data buffer transmission/ reception between one master and two slaves with different address sizes (7-bit or 10-bit). This example uses the STM32L4xx I2C HAL and LL API (LL API usage for performance improvement) and an interrupt.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
	OPAMP	OPAMP_CALIBRATION	How to calibrate and operate the OPAMP peripheral. This example is based on the STM32L4xx OPAMP HAL and LL API (LL API used for performance improvement).	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-
	PWR	PWR_STANDBY_RTC	How to enter Standby mode and wake up from this mode using an external reset or the RTC wakeup timer through the STM32L4xx RTC and RCC HAL and LL API. The LL API is used for performance improvement.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		PWR_STOP1	How to enter the system in Stop 1 mode and wake up from this mode using external reset or wakeup interrupt (all the RCC functions calls use RCC LL API for footprint and performance improvements).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
	SPI	SPI_FullDuplex_ComPolling	Data buffer transmission/reception between two boards via SPI using Polling mode.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		SPI_HalfDuplex_ComPollingIT	Data buffer transmission/reception between two boards via SPI using Polling (LL driver) and Interrupt modes (HAL driver).	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
	TIM	TIM_6Steps	Configuration of the TIM1 peripheral to generate six-step PWM signals.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
	UART	UART_HyperTerminal_IT	Use of a UART to transmit data (transmit/ receive) between a board and an HyperTerminal PC application in Interrupt mode. This example describes how to use the USART peripheral through the STM32L4xx UART HAL and LL API, the LL API being used for performance improvement.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
		UART_HyperTerminal_TxPolling_RxIT	Use of a UART to transmit data (transmit/ receive) between a board and an HyperTerminal PC application both in Polling and Interrupt modes. This example describes how to use the USART peripheral through the STM32L4xx UART HAL and LL API, the LL API being used for performance improvement.	-	-	-	-	-	-	X	X	-	-	-	-	-	-	-	-	-	-	-
Total number of examples_MIX: 29				**0**	**0**	**0**	**0**	**1**	**0**	**10**	**12**	**0**	**0**	**0**	**0**	**2**	**2**	**0**	**0**	**0**	**2**	**0**

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Applications	-	Proximity	This application shows how to use the VL53L0X sensor mounted on the B-L475E-IOT01A board to provide proximity information.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-
	BLE	HeartRate	This application shows how to use BLE component for HeartRate profile application.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-
		P2P_LedButton	This example aims at demonstrating point-to-point communications using the BLE component.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-
	FatFS	FatFs_RAMDisk	How to use STM32Cube firmware with FatFs middleware component as a generic FAT file system module. This example develops an application that exploits FatFs features to configure a RAM disk (SRAM) drive.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		FatFs_USBDisk_Standalone	How to use STM32Cube firmware with FatFs middleware component as a generic FAT file system module and STM32 USB On-The-Go (OTG) host library, in both Full Speed (HS) mode. This example develops an application exploiting FatFs features, with USB disk drive configuration.	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		FatFs_eMMC_Standalone	How to use STM32Cube firmware with FatFs middleware component as a generic FAT file system module. This example develops an application that exploits FatFs features to configure a eMMC drive.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	New	-	-
		FatFs_uSD_DMA_RTOS	How to use STM32Cube firmware with FatFs middleware component as a generic FAT file system module. This example develops an application exploiting FatFs features, with a microSD drive in RTOS mode configuration.	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-
		FatFs_uSD_DMA_Standalone	How to use STM32Cube™ firmware with FatFs middleware component as a generic FAT file system module. This example develops an application making the most of FatFs features to configure a microSD drive.	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		FatFs_uSD_Standalone	How to use STM32Cube firmware with FatFs middleware component as a generic FAT file system module. This example develops an application that exploits FatFs features to configure a microSD drive.	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-
	FreeRTOS	FreeRTOS_LowPower	How to enter and exit low-power mode with CMSIS RTOS API.	X	X	-	X	-	-	X	-	-	X	-	-	X	-	-	X	-	-	-
		FreeRTOS_LowPower_LPTIM	How to enter and exit low-power mode with CMSIS RTOS API with LPTIM used as clock source for both RTOS and HAL ticks.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Applications	FreeRTOS	FreeRTOS_MPU	How to use the MPU feature of FreeRTOS.	X	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-
		FreeRTOS_Mail	How to use mail queues with CMSIS RTOS API.	-	X	-	-	-	-	X	-	-	X	-	-	X	-	-	-	-	-	-
		FreeRTOS_Mutexes	How to use mutexes with CMSIS RTOS API.	-	X	-	-	-	-	X	-	-	X	-	-	X	-	-	-	-	-	-
		FreeRTOS_Queues	How to use message queues with CMSIS RTOS API.	-	X	-	-	-	-	X	-	-	X	-	-	X	-	-	-	-	-	-
	FreeRTOS	FreeRTOS_Semaphore	How to use semaphores with CMSIS RTOS API.	-	X	-	-	-	-	X	-	-	X	-	-	X	-	-	-	-	-	-
		FreeRTOS_SemaphoreFromISR	How to use semaphore from ISR with CMSIS RTOS API.	-	X	-	-	-	-	X	-	-	X	-	-	X	-	-	-	-	-	-
		FreeRTOS_Signal	How to perform thread signaling using CMSIS RTOS API.	-	X	-	-	-	-	X	-	-	X	-	-	X	-	-	-	-	-	-
		FreeRTOS_SignalFromISR	How to perform thread signaling from an interrupt using CMSIS RTOS API.	-	X	-	-	-	-	X	-	-	X	-	-	X	-	-	-	-	-	-
		FreeRTOS_ThreadCreation	How to implement thread creation using CMSIS RTOS API.	X	X	-	-	-	-	X	X	-	X	-	X	X	X	-	-	-	-	X
		FreeRTOS_Timers	How to use timers of CMSIS RTOS API.	-	X	-	-	-	-	X	-	-	X	-	-	X	-	-	-	-	-	-
	IAP	IAP_Binary_Template	This directory contains a set of sources files that build the application to be loaded into Flash memory using In-Application Programming (IAP) through USART.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		IAP_Main	This directory contains a set of sources files and pre-configured projects that describes how to build an application to be loaded into Flash memory using In-Application Programming (IAP) through USART.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	NFC	WrAARtoRunBLEapp	This application aims at showing how to write an AAR NDEF message to an M24SR type 4 NFC tag so that the BLE STM32 Profiles application is launched on the smartphone when it comes near the NFC antenna .	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-
		WriteTag	This application aims at showing how to write NDEF messages to an M24SR type 4 NFC tag so that the associated application is launched on the smartphone when it comes near the NFC antenna.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-
	STemWin	HelloWorld	Simple "Hello World" example based on STemWin.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-
		STemWin_HelloWorld	Simple "Hello World" example based on STemWin.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Applications	STemWin	STemWin_SampleDemo	How to implement a sample demonstration example allowing to show some of the STemWin Library capabilities.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-
	SubGhz	P2P	This application aims at demonstrating point-to-point communication between two B-L475E-IOT01A boards with SubGhz module using Spirit1 driver and STM32Cube firmware.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-
	Touch Sensing	TouchSensing_1touchkey	Use of the STMTouch driver with 1 touchkey sensor.	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	USB_ Device	CDC_Standalone	Use of the USB device application based on the Device Communication Class (CDC) and following the PSTN subprotocol. This application uses the USB Device and UART peripherals.	-	X	-	-	-	-	-	-	-	X	-	-	X	-	-	-	-	X	-
		CustomHID_Standalone	Use of the USB device application based on the Custom HID Class.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		DFU_Standalone	Compliant implementation of the Device Firmware Upgrade (DFU) capability to program the embedded Flash memory through the USB peripheral.	X	X	-	X	-	-	-	-	-	X	-	X	X	-	-	X	-	X	X
		HID_Standalone	Use of the USB device application based on the Human Interface (HID).	X	X	-	X	-	-	-	-	-	X	-	X	X	-	X	X	-	X	X
		HID_Standalone_BCD	Use of the USB device application based on the Human Interface (HID) with Battery Charger Detection (BCD).	-	X	-	-	-	-	-	-	-	X	-	X	X	-	-	-	-	X	X
		HID_Standalone_LPM	Use of the USB device application based on the Human Interface (HID) with Link Power Management Protocol (LPM).	-	X	-	-	-	-	-	-	-	X	-	X	X	-	-	-	-	X	X
		MSC_Standalone	Use of the USB device application based on the Mass Storage Class (MSC).	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	New	-	-
	USB_Host	CDC_Standalone	Use of the USB host application based on the CDC class.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		HID_Standalone	Use of the USB host application based on the HID class.	-	X	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-
		MSC_Standalone	Use of the USB host application based on the Mass Storage Class (MSC).	X	X	-	X	-	-	-	-	-	-	-	-	-	-	X	X	-	X	-
	WiFi	WiFi_Client_Server	This application shows how to use the Es-WiFi module to perform a TCP client mode using STM32Cube HAL. It demonstrates how to set up a client program and connect it to a TCP server.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Applications	WiFi	WiFi_HTTP_Server	This application shows how to make HTTP requests using the Es-WiFi module based on STM32Cube HAL.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-
	Total number of applications: 128			**12**	**28**	**0**	**5**	**0**	**0**	**10**	**2**	**0**	**15**	**0**	**5**	**15**	**1**	**11**	**6**	**2**	**11**	**5**
Demonstrations	-	Adafruit_LCD_1_8_SD_Joystick	Demonstration firmware based on STM32Cube. This example helps you to discover STM32 Cortex-M devices that are plugged onto your STM32 Nucleo board.	-	-	-	X	-	-	X	X	-	X	-	-	X	-	-	-	-	-	-
		Demo		-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	X	X
		Gravitech_4digits	Demonstration of firmware based on STM32Cube. This example provides firmware to to help you to discover STM32 Cortex-M devices that are plugged onto an your STM32NUCLEO_32 board.	-	-	-	-	-	-	-	-	-	-	-	X	-	X	-	-	-	-	-
		MB1144	The STM32Cube demonstration platform comes on top of the STM32Cube as a firmware package that offers a full set of software components based on a modular architecture. All modules can be reused separately in standalone applications. All these modules are managed by the STM32Cube demonstration kernel that allows to dynamically add new modules and access common resources (storage, graphical components and widgets, memory management, real-time operating system). The STM32Cube demonstration platform is built around the powerful graphical STemWin library and the FreeRTOS realtime operating system. It uses almost all STM32 features and offers a large scope of use cases based on the STM32Cube HAL BSP and several middleware components.	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		MenuLauncher	The STM32Cube demonstration platform comes on top of the STM32Cube(TM) as a firmware package that offers a full set of software components based on a modular architecture.	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-
		Watermark	The software watermarks the PNG xxxxx.png files with logo.png.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	New	-	-
	Embedded Wizard	Binary	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-
	STemWin	Binary	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-

Level	Module Name	Project Name	Description	STM32L4R9I-EVAL	STM32L476G-EVAL	NUCLEO-L4R5ZI-P	NUCLEO-L4R5ZI	NUCLEO-L4P5ZG	NUCLEO-L496ZG-P	NUCLEO-L496ZG	NUCLEO-L476RG	NUCLEO-L452RE-P	NUCLEO-L452RE	NUCLEO-L433RC-P	NUCLEO-L432KC	NUCLEO-L412RB-P	NUCLEO-L412KB	B-L475E-IOT01A	32L4R9IDISCOVERY	32L4P5GDISCOVERY	32L496GDISCOVERY	32L476GDISCOVERY
Demonstrations	TouchGFX	Binary	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-
	Total number of demonstrations: 17			1	1	0	1	0	0	1	1	0	1	0	1	1	1	1	4	1	1	1
Total number of projects: 1147				69	118	5	60	68	5	214	178	5	100	5	70	78	33	15	28	31	40	25

Revision history

Table 2. **Document revision history**

Date	Version	Changes
6-Jul-2015	1	Initial release.
15-Sep-2015	2	Added UM1916 and UM1919 in *Section 1: Reference documents*. Updated *Figure 1: STM32CubeL4 firmware components*. Updated *Section 2: STM32CubeL4 examples* to add Low Layer drivers.
26-Feb-2016	3	Added UM1937 in *Section 1: Reference documents*. *Table 1: STM32CubeL4 firmware examples*: updated HAL and LL examples.
10-Mar-2016	4	Added NUCLEO-L432KC board together with several examples, applications and Gravitech_4digits demonstration. Added I2C_TwoBoards_RestartAdvComIT and I2C_TwoBoards_RestartComIT examples for STM32L476G-EVAL and NUCLEO-L476RG boards. Added QSPI_PreInitConfig example for STM32L476GEVAL and 32L496GDISCOVERY boards.
14-Nov-2016	5	Updated description of template projects in *Section 2* introduction. Added NUCLEO-L452RE board with several examples, applications and demonstration with Adafruit® shield. Added Templates_LL new projects to allow the quick build of a firmware application on a given board with the LL API.
13-Dec-2016	6	Removed NUCLEO-L452RE board.
28-Feb-2017	7	Updated *Figure 1: STM32CubeL4 firmware components*. Added NUCLEO-L496ZG and 32L496GDISCOVERY boards as well as corresponding examples, applications and demonstrations (such as Adafruit® shield on the Nucleo board and graphic demonstration on the Discovery kit).
25-Jun-2017	8	Updated *Figure 1: STM32CubeL4 firmware components*. Modified LL APIs preferred spelling to ‘low-layer APIs. Added B-L475RE-IOT01A and NUCLEO-L452RE boards together with several examples, applications and project templates.
25-Jul-2017	9	Added NUCLEO-L433RC-P board together with several examples, applications and project templates.
1-Sep-2017	10	Added NUCLEO-L4R5ZI board together with several examples, applications and demonstration based on the Adafruit® shield.
3-Nov-2017	11	Added STM32L4R9I-EVAL and 32L4R9IDISCOVERY boards together with several examples, applications, demonstrations and project templates.
11-Oct-2018	12	Added NUCLEO-L412KB and NUCLEO-L412RB-P boards together with several examples, applications, demonstrations and project templates.
18-Dec-2019	13	Added NUCLEO-L4P5ZG and 32L4P5GDISCOVERY boards together with several examples, applications, demonstrations and project templates.

Contents

List of tables