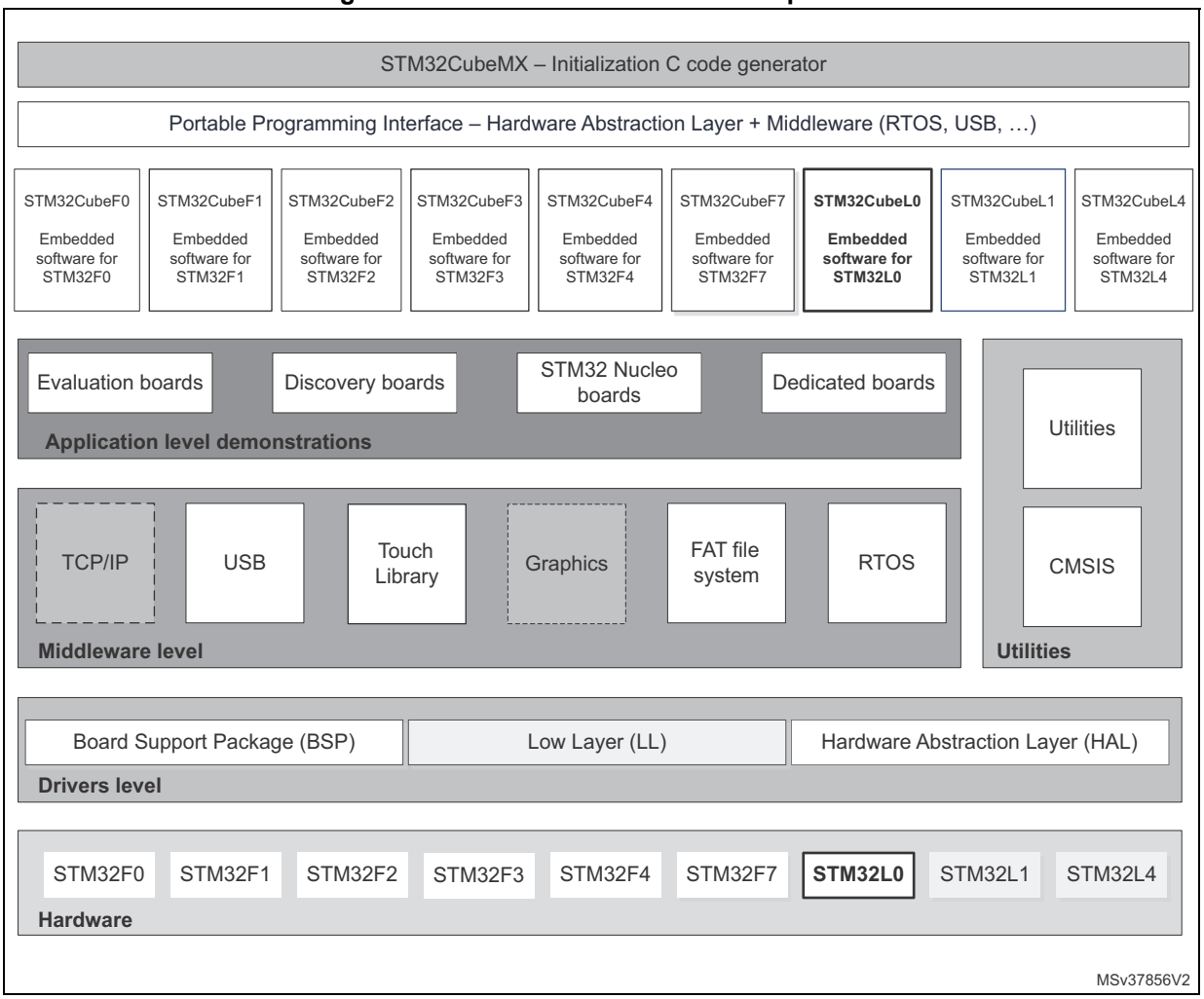
AN4725
Application note

STM32Cube firmware examples for STM32L0 Series

Introduction

The STM32CubeL0 firmware package comes 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 (see *Figure 1*).

Figure 1. STM32CubeL0 firmware components



Contents

1 Reference documents

The following user manuals are available on www.st.com/stm32cubefw:

- Latest release of STM32CubeL0 firmware package
- *Getting started with the STM32CubeL0 firmware package for the STM32L0 series* (UM1754)
- *Description of STM32L0xx HAL drivers* (UM1749)
- *STM32Cube USB Device library* (UM1734)
- *Developing applications on STM32Cube with FatFs* (UM1721)
- *Developing Applications on STM32Cube with RTOS (*UM1722)

2 STM32CubeL0 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 (the middleware is not used). Their objective is to demonstrate the product/peripherals features and usage. They are organized per peripheral (one folder for each peripheral, e.g. TIMER). Their complexity level ranges from the basic usage of a given peripheral (e.g. PWM generation using timer) to the integration of several peripherals (e.g. 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, e.g. TIM) and run exclusively on Nucleo boards.

- **Examples_MIX**

  These examples use only HAL, BSP and LL drivers (middleware components 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, e.g. TIM) and run exclusively on Nucleo boards.

- **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, e.g. USB Host) or by product feature that require high-level firmware bricks (e.g. 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 on a given board.

- **Template_LL project**

  The template LL project is 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_L0_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 *TrueSTUDIO* 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 an 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 firmware 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, 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 contains the list of examples provided within STM32CubeL0 firmware package.

Table 1. STM32CubeL0 firmware examples

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Templates_LL	-	Starter project	This project provides a reference template through the LL API that can be used to build any firmware application.	New	New	New	New	New	New
	Total number of templates: 6			**1**	**1**	**1**	**1**	**1**	**1**
Templates	-	Starter project	This project provides a reference template that can be used to build any firmware application.	X	X	X	X	X	X
	Total number of templates: 6			**1**	**1**	**1**	**1**	**1**	**1**
Examples	-	BSP	This example provides a description of how to use the different BSP drivers.	-	-	-	X	-	X
	ADC	ADC_AnalogWatchdog	This example provides a short description of how to use the ADC peripheral to perform conversions with analog watchdog and interruptions.	X	-	-	-	-	X
		ADC_DMA_Transfer	This example describes 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	X	X	X	X
		ADC_LowPower	This example provides a short description of how to use the ADC peripheral to perform conversions with ADC low power modes: auto-wait and auto-power off.	X	X	X	X	X	X
		ADC_OverSampler	This example describes how to configure and use ADC to convert an external analog input (PA0) combined with oversampling feature to increase resolution, through the STM32L0xx HAL API.	-	X	X	X	-	-
		ADC_Regular Conversion_Interrupt	This example describes how to use ADC1 and an interrupt to convert continuously data from ADC1 channel0, through the STM32L0xx HAL API.	X	X	X	X	-	X
		ADC_Regular Conversion_Polling	This example describes how to use ADC1 and the Polling mode to convert data from ADC1 channel0.	X	X	-	X	-	X
		ADC_Sequencer	This example provides a short description of how to use the ADC peripheral with sequencer to convert several channels.	-	-	-	New	-	-
	AES	AES_DMA	This example provides a short description of how to use the AES peripheral to encrypt and decrypt data using AES Algorithm with ECB chaining mode.	-	-	-	X	-	-
		AES_Modes	This example provides a short description of how to configure the AES hardware accelerator to encrypt then to decrypt text with the different modes ECB, CBC and CTR.	-	-	-	X	-	-

Table 1. STM32CubeL0 firmware examples (continued)

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Examples	COMP	COMP_Analog Watchdog	This example shows how to make an analog watchdog using the STM32L0xx COMP peripherals in Window mode: – The upper threshold is set to VREFINT = 1.22 V. – The lower threshold is set to VREFINT/4 = 1.22 V/4 = 0.305 V. – The input voltage is configured to be connected to PB4 pin. – LEDs on STM32L073Z-EVAL can be used to monitor the voltage level compared to comparators thresholds: LED3 and LED1 are turned ON and the MCU wakes up from Stop mode if the input voltage is above the higher threshold.	X	-	X	X	-	X
		COMP_Interrupt	This example shows how to configure the COMP peripheral to compare the external voltage applied on PA1 with the Internal Voltage Reference. When the comparator input which is connected to PA1 crosses (either rising or falling edges) the internal reference voltage VREFINT (1.22 V), the comparator 1 generates an interrupt and toggles LED2.	-	-	-	X	-	-
		COMP_PWMSignalControl	This example shows how to configure COMP2 peripheral to automatically hold TIM21 PWM output on safe state (low level) as soon as comparator output is set High.	X	X	X	X	-	X
		COMP_PulseWidth Measurement	This example shows how to configure the COMP1 peripheral to measure pulse width.	X	-	X	X	-	X
	CRC	CRC_Data_Reversing_16bit_CRC	This example guides you through the different configuration steps by means of the HAL API. The CRC (Cyclic Redundancy Check) calculation unit computes a 16-bit long CRC code derived from a buffer of 8-bit data (bytes).	X	-	-	-	-	-
		CRC_Example	This example guides you through the different configuration steps by mean of HAL API to ensure the use of the CRC (Cyclic Redundancy Check) calculation unit to get a CRC code of a given buffer of data words (32-bit long), based on a fixed generator polynomial (0x4C11DB7).	X	X	X	X	X	X
		CRC_bytes_stream_7bit_CRC	This example guides you through the different configuration steps by means of the HAL API. The CRC (Cyclic Redundancy Check) calculation unit computes 7-bit long CRC codes derived from buffers of 8-bit data (bytes).	X	-	-	-	-	-

Table 1. STM32CubeL0 firmware examples (continued)

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Examples	CRYP	CRYP_AESModes	This example provides a short description of how to use the CRYP peripheral to encrypt and decrypt data using AES in chaining modes (ECB, CBC, CTR).	X	X	-	-	X	-
		CRYP_DMA	This example provides a short description of how to use the CRYP peripheral to encrypt and decrypt data using AES 128 Algorithm with ECB chaining mode in DMA mode.	X	X	-	-	X	-
	Cortex	CORTEXM_MPU	This example presents the MPU features. Its purpose is to configure a memory area as privileged read-only area and attempt to perform read and write operations in different modes.	X	-	X	X	-	X
		CORTEXM_ModePrivilege	This example shows how to modify Cortex®-M0+ Thread mode privilege access and stack.	-	-	X	X	-	-
		CORTEXM_SysTick	This example shows how to use the default SysTick configuration with a 1 ms timebase to toggle LEDs.	X	-	X	X	-	X
	DAC	DAC_Signals Generation	This example provides a description of how to use the DAC peripheral to generate several signals using the DMA controller.	X	-	X	X	-	X
		DAC_Simple Conversion	This example provides a short description of how to use the DAC peripheral to perform a simple conversion of the 0xFF value in 8-bit right aligned mode. The conversion result can be retrieved by connecting an oscilloscope to PA4 (DAC channel1).	X	-	X	X	-	X
	DMA	DMA_FLASHToRAM	This example provides a description of how to use a DMA to transfer a word data buffer from Flash memory to embedded SRAM through the HAL API.	X	X	X	X	X	X
		DMA_RAMToDAC	This example provides a description of how to use a DMA channel to transfer data buffer from RAM to DAC.	X	-	X	X	-	-
	FIREWALL	FIREWALL_Volatile Data_Executable	This example shows how to use the Firewall IP to protect a volatile data segment located in SRAM and to define it as executable.	-	-	-	-	-	X
		FIREWALL_Volatile Data_Shared	This example shows how to use the Firewall IP to protect a code segment as well as volatile and non-volatile data segments.	-	-	-	-	-	X

Table 1. STM32CubeL0 firmware examples (continued)

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Examples	FLASH	FLASH_DualBoot	This example guides you through the different configuration steps by mean of HAL API. It shows how to program Bank 1 and Bank 2 of the STM32L0xx internal Flash memory mounted on STM32L073Z-EVAL and swap between both of them.	-	-	-	-	-	X
		FLASH_DualBoot_ Workaround	This example guides you through the different configuration steps by mean of HAL API. It shows how to program Bank 1 and Bank 2 of the STM32L0xx internal Flash memory mounted on STM32L073Z-EVAL and boot from one of them.	-	-	-	-	-	X
		FLASH_EraseProgram	This example guides you through the different configuration steps by mean of HAL API how. It shows how to erase and program the STM32L0xx internal Flash memory mounted on STM32L073Z-EVAL board.	X	X	X	X	X	X
		FLASH_ WriteProtection	This example describes how to configure and use the FLASH HAL API to enable and disable the write protection on the internal Flash memory.	X	X	X	X	X	X
	GPIO	GPIO_EXTI	This example shows how to configure external interrupt lines.	X	-	X	X	-	X
		GPIO_IOToggle	This example describes how to configure and use GPIOs through the STM32L0xx HAL API.	X	X	-	X	X	X
		GPIO_IOToggle_ MaxFrequency	This example describes how to configure and use GPIOs through the STM32L0xx HAL API.	X	X	X	X	X	X
		GPIO_IOToggle_ VariableFreq	This example describes how to make LED3 toggle at different frequencies. It configures and uses the GPIOs connected to LEDs and virtual button through the STM32L0xx HAL API.	-	X	-	-	-	-
	HAL	HAL_TimeBase	This example describes how to customize the HAL timebase using a general purpose timer (TIM6) instead of SysTick as main timebase source.	-	-	X	X	-	-

Table 1. STM32CubeL0 firmware examples (continued)

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Examples	I2C	I2C_TwoBoards_AdvComIT	This example describes how to perform I2C data buffer transmission/reception between two boards, using an interrupt.	X	X	X	X	X	X
		I2C_TwoBoards_ComDMA	This example describes how to perform I2C data buffer transmission/reception between two boards, via DMA.	X	X	X	X	X	X
		I2C_TwoBoards_ComIT	This example describes how to perform I2C data buffer transmission/reception between two boards using an interrupt.	-	X	X	X	X	-
		I2C_TwoBoards_ComPolling	This example describes how to perform I2C data buffer transmission/reception between two boards in Polling mode.	-	X	-	X	X	-
		I2C_TwoBoards_RestartAdvComIT	This example describes how to perform a multiple I2C data buffer transmission/reception between two boards in Interrupt mode and with a restart condition.	New	-	-	-	-	-
		I2C_TwoBoards_RestartComIT	This example describes how to perform a single I2C data buffer transmission/reception between two boards in Interrupt mode and with a restart condition.	New	-	-	-	-	-
		I2C_WakeUpFromStop	This example describes how to perform I2C data buffer transmission/reception between two boards using an interrupt when the device is in STOP mode.	X	X	-	-	X	X
	IWDG	IWDG_Reset	This example guides you through the different configuration steps by mean of HAL API to reload IWDG counter and to simulate a software fault that generates an MCU IWDG reset on expiry of a programmed time period.	New	New	X	X	New	New
		IWDG_WindowMode	This example uses HAL IWDG API to periodically update the IWDG reload counter and simulate a software fault that generates an MCU IWDG reset when a programmed time period has elapsed.	X	-	X	X	-	X
	LCD	LCD_Blink_Frequency	This example provides a description of how to use the STM32L0xx embedded LCD GLASS controller and how to configure the LCD Blink mode and Blink frequency.	-	-	-	X	-	X
		LCD_SegmentsDrive	This example provides a description of how to use the STM32L0xx embedded LCD controller to drive the Pacific Display LCD glass mounted on STM32L152C-Discovery board.	-	-	-	X	-	-

Table 1. STM32CubeL0 firmware examples (continued)

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Examples	LPTIM	LPTIM_PWM ExternalClock	This example describes how to configure and use LPTIM to generate a PWM at the lowest power consumption, using an external counter clock, through the HAL LPTIM API.	-	-	X	X	-	X
		LPTIM_PWM_LSE	This example describes how to configure and use LPTIM to generate a PWM in low power mode using the LSE as a counter clock, through the HAL LPTIM API	-	-	X	X	-	X
		LPTIM_PulseCounter	This example describes how to configure and use LPTIM to count pulses through the LPTIM HAL API.	X	X	X	X	X	X
		LPTIM_Timeout	This example describes how to implement a low power timeout to wake-up the system using the LPTIMER, through the HAL LPTIM API.	X	X	X	X	-	X
	PWR	PWR_LPRUN	This example shows how to enter Low-power run mode and exit from this mode using WKUP/TAMPER pushbutton.	X	X	X	X	X	X
		PWR_LPSLEEP	This example shows how to enter Low-power sleep mode and wake up from this mode using WKUP/TAMPER pushbutton EXTI lines 4 to 15.	X	X	X	X	X	X
		PWR_PVD	This example shows how to configure the programmable voltage detector using an external interrupt line. External DC supply has to be used to power V_{DD}.	X	X	X	X	X	X
		PWR_SLEEP	This example shows how to enter Sleep mode and wake up from this mode using WKUP/TAMPER pushbutton EXTI lines 4 to 15.	X	X	X	X	X	X
		PWR_STANDBY	This example shows how to enter Standby mode and wake up from this mode using external RESET or WKUP pin.	X	X	X	X	X	X
		PWR_STANDBY_RTC	This example shows how to enter Standby mode and wake up from this mode using RTC Wakeup Timer Event connected to EXTI_Line20.	X	X	X	X	X	X
		PWR_STOP	This example shows how to enter Stop mode and wake up from this mode using Key push button EXTI_Line4-15.	X	X	X	X	X	X
		PWR_STOP_RTC	This example shows how to enter Stop mode and wake up from this mode using RTC Wakeup Timer Event connected to EXTI_Line17.	X	X	X	X	X	X

Table 1. STM32CubeL0 firmware examples (continued)

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Examples	RCC	RCC_CRS_ Synchonization_IT	This example describes how to use the RCC HAL API to configure Clock Recovery Service in IT mode.	-	-	-	-	-	New
		RCC_CRS_ Synchonization_ Polling	This example describes how to use the RCC HAL API to configure Clock Recovery Service in polling mode.	-	-	-	-	-	New
		RCC_ClockConfig	This example describes how to use the RCC HAL API to configure the system clock (SYSCLK) and modify the clock settings in Run mode.	-	-	-	X	-	X
		RCC_LSIConfig	This example describes how to use the RCC HAL API to enable or disable the low-speed internal (LSI) RC oscillator (about 40 KHz) in Run mode.	X	X	-	-	X	X
	RNG	RNG_MultiRNG	This example guides you through the HAL API different configuration steps to generate 32-bit long random numbers using the RNG peripheral.	X	-	X	X	-	X
	RTC	RTC_Alarm	This example guides you through the different configuration steps by means of the RTC HAL API to configure and generate an RTC alarm.	X	X	X	X	X	X
		RTC_Calendar	This example guides you through the different configuration steps by mean of HAL API to ensure Calendar configuration using the RTC peripheral.	-	X	X	-	X	X
		RTC_LSI	This example demonstrates and explains how to use the LSI clock source auto-calibration to get an accurate RTC clock.	X	X	X	X	X	X
		RTC_LowPower_ STANDBY	This example shows how to enter Standby mode and wake up from this mode using the RTC Alarm Event.	-	-	-	-	-	X
		RTC_Tamper	This example guides you through the different configuration steps by means of the RTC HAL API to write/read data to/from RTC Backup registers and to demonstrate the tamper detection feature.	X	X	X	X	X	X
		RTC_TimeStamp	This example guides you through the different configuration steps by means of the RTC HAL API to demonstrate the timestamp feature.	X	X	X	X	X	X
	SMBUS	SMBUS_TSENSOR	This example guides you through the different configuration steps by mean of HAL API to ensure SMBUS data buffer transmission and reception in Interrupt mode.	-	-	-	-	-	X

Table 1. STM32CubeL0 firmware examples (continued)

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Examples	SPI	SPI_FullDuplex_AdvCom	This example guides you through the different configuration steps by mean of HAL API to transmit/receive SPI data buffer in Polling mode and in an advanced communication mode: the master board always sends the command to the slave before any transmission is performed.	-	-	-	X	-	-
		SPI_FullDuplex_ComDMA	This example shows how to perform SPI data buffer transmission/reception between two boards via DMA.	X	X	X	X	X	X
		SPI_FullDuplex_ComIT	This example shows how to ensure SPI data buffer transmission/reception between two boards using an interrupt.	X	X	X	X	X	X
		SPI_FullDuplex_ComPolling	This example shows how to ensure SPI data buffer transmission/reception in Polling mode between two boards.	X	X	-	X	X	X
	TIM	TIM_DMA	This example provides a description of how to use DMA with TIMER Update request to transfer data from memory to TIMER Capture Compare Register 3 (TIMx_CCR3).	X	X	X	X	X	X
		TIM_DMABurst	This example shows how to update the TIMER channel1 period and the duty cycle using the TIMER DMA burst feature.	X	X	X	X	X	X
		TIM_ExtTrigger Synchro	This example shows how to synchronize TIM peripherals in cascade mode with an external trigger.	-	-	-	X	-	-
		TIM_InputCapture	This example shows how to use the TIM peripheral to measure an external signal frequency.	X	X	X	X	X	X
		TIM_OCActive	This example shows how to configure the TIM peripheral in Output Compare Active mode (when the counter matches the capture/compare register, the concerned output pin is set to its active state).	X	X	X	X	X	X
		TIM_OCInactive	This example shows how to configure the TIM peripheral in Output Compare Inactive mode with the corresponding Interrupt requests for each channel.	X	X	-	X	X	X
		TIM_OCToggle	This example shows how to configure the TIM peripheral to generate four different signals with four different frequencies.	X	X	X	X	X	X
		TIM_OnePulse	This example shows how to use the TIM peripheral to generate a One pulse mode after a rising edge of an external signal is received in Timer Input pin.	X	X	-	X	X	X
		TIM_PWMInput	This example shows how to use the TIM peripheral to measure the frequency and duty cycle of an external signal.	X	-	-	-	-	-
		TIM_PWMIntput	This example shows how to use the TIM peripheral to measure the frequency and duty cycle of an external signal.	-	-	-	X	-	-
		TIM_PWMOutput	This example shows how to configure the TIM peripheral in PWM (Pulse Width Modulation) mode.	X	X	X	X	X	X

Table 1. STM32CubeL0 firmware examples (continued)

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Examples	TIM	TIM_TimeBase	This example shows how to configure the TIM peripheral to generate a timebase of one second with the corresponding Interrupt request.	X	X	X	X	X	X
	TSC	TSC_BasicAcquisition_Interrupt	This example describes how to use the TSC to perform continuous acquisitions of two channels in Interrupt mode.	-	-	-	-	-	X
		TSC_BasicAcquisition_Polling	This example describes how to use the TSC to perform continuous acquisitions of one channel in Polling mode.	-	-	-	-	-	X
	UART	LPUART_TwoBoards_ComIT	This example shows an LPUART transmission (transmit/receive) between two STM32L0538-DISCO boards in Interrupt mode, the two LPUARTs are clocked with LSI.	-	-	X	-	-	-
		LPUART_WakeUpFromStop	This example shows how to configure the LPUART to wake up the MCU from Stop mode when the proper stimulus is received.	-	-	X	-	-	-
		UART_HyperTerminal_DMA	This example shows how to ensure UART data buffer transmission and reception with DMA. The communication is performed with the HyperTerminal PC application.	-	X	-	-	X	X
		UART_LowPower_HyperTerminal_DMA	This example guides you through the different configuration steps by mean of HAL API to ensure UART data buffer transmission and reception with DMA.	-	-	-	-	-	X
		UART_Printf	This example shows how to re-target the C library printf function to the UART.	-	X	-	-	New	X
		UART_TwoBoards_ComDMA	This example describes an UART transmission (transmit/receive) in DMA mode between two STM32L073Z-EVAL boards.	X	X	X	X	X	X
		UART_TwoBoards_ComIT	This example describes an UART transmission (transmit/receive) in Interrupt mode between two STM32L073Z-EVAL boards.	X	X	X	X	X	X
		UART_TwoBoards_ComPolling	This example describes a UART transmission (transmit/receive) in Polling mode between two STM32L073Z-EVAL boards.	X	X	-	X	X	X
		UART_WakeUpFromStop	This example shows how to configure an UART to wake up the MCU from Stop mode when the proper stimulus is received.	-	X	-	-	X	X
	WWDG	WWDG_Example	This example guides you through the different configuration steps by mean of HAL API to periodically update WWDG counter and simulate a software fault that generates an MCU WWDG reset when a programmed time period has elapsed.	X	X	X	X	X	X
	Total number of examples: 375			**64**	**56**	**57**	**72**	**50**	**76**

Table 1. STM32CubeL0 firmware examples (continued)

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Examples_LL	ADC	ADC_AnalogWatchdog	This example describes how to use a ADC peripheral with ADC analog watchdog to monitor a channel and detect when the corresponding conversion data is out of window thresholds; This example is based on the STM32L0xx ADC LL API; peripheral initialization done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		ADC_Continuous Conversion_Trigger SW	This example describes how to use a ADC peripheral to perform continuous ADC conversions of a channel, from a software start. This example is based on the STM32L0xx ADC LL API; peripheral initialization done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		ADC_Continuous Conversion_Trigger SW_Init	This example describes how to use a ADC peripheral to perform continuous ADC conversions of a channel, from a software start; This example is based on the STM32L0xx ADC LL API; peripheral initialization done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		ADC_Continuous Conversion_Trigger SW_LowPower	This example describes how to use a ADC peripheral with ADC low-power features. This example is based on the STM32L0xx ADC LL API; peripheral initialization done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		ADC_MultiChannel SingleConversion	This example describes how to use a ADC peripheral to convert several channels. ADC conversions are performed successively in a scan sequence. This example is based on the STM32L0xx ADC LL API; peripheral initialization done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		ADC_Oversampling	This example describes how to use a ADC peripheral with ADC oversampling. This example is based on the STM32L0xx ADC LL API; peripheral initialization done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		ADC_Single Conversion_ TriggerSW	This example describes how to use a ADC peripheral to perform a single ADC channel conversion 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 STM32L0xx ADC LL API; peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-

Table 1. STM32CubeL0 firmware examples (continued)

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Examples_LL	ADC	ADC_Single Conversion_Trigger SW_DMA	This example describes how to use a ADC peripheral to perform a single ADC channel conversion 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 STM32L0xx ADC LL API; peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		ADC_Single Conversion_Trigger SW_IT	This example describes how to use a ADC peripheral to perform a single ADC channel conversion 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 STM32L0xx ADC LL API; peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		ADC_Single Conversion_Trigger Timer_DMA	This example describes how to use a ADC peripheral to perform a single ADC channel conversion at each trigger event from time. Converted data are indefinitely transferred via DMA into a table (circular mode). This example is based on the STM32L0xx ADC LL API; peripheral initialization done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		ADC_Temperature Sensor	This example describes how to use a ADC peripheral to perform a single ADC conversion of the internal temperature sensor and calculate the temperature in °C. This example using the Polling programming model (for programming models in Interrupt or DMA mode, refer to other examples). This example is based on the STM32L0xx ADC LL API; peripheral initialization done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
	COMP	COMP_CompareWith InternalReference_IT	This example shows 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 STM32L0xx COMP LL API; peripheral initialization done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		COMP_CompareWith InternalReference_IT_ Init	This example shows 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 STM32L0xx COMP LL API; peripheral initialization done using LL initialization function to demonstrate LL init usage.	X	-	-	-	-	-

Table 1. STM32CubeL0 firmware examples (continued)

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Examples_LL	CORTEX	CORTEX_MPU	This example presents the MPU feature. Its purpose is to configure a memory area as privileged read-only area and attempt to perform read and write operations in different modes.	X	-	-	-	-	-
	CRC	CRC_ CalculateAndCheck	This example shows how to configure CRC calculation unit to get the CRC code of a given data buffer, based on a fixed generator polynomial (default value 0x4C11DB7). Peripheral initialization done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		CRC_UserDefined Polynomial	This example shows how to configure and use CRC calculation unit to get the 8-bit long CRC of a given data buffer, based on a user-defined generating polynomial. Peripheral initialization done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
	CRS	CRS_ Synchronization_IT	This example describes how to configure Clock Recovery Service in Interrupt mode through the STM32L0xx CRS LL API. Peripheral initialization done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		CRS_Synchronization_ Polling	This example describes how to configure Clock Recovery Service in Polling mode through the STM32L0xx CRS LL API. Peripheral initialization done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
	DAC	DAC_Generate ConstantSignal_ TriggerSW	This example describes how to use the DAC peripheral to generate a constant voltage signal. This example is based on the STM32L0xx DAC LL API; peripheral initialization done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		DAC_Generate Waveform_TriggerHW	This example describes how to use the DAC peripheral to generate a voltage waveform from digital data stream transfered by DMA. This example is based on the STM32L0xx DAC LL API; peripheral initialization done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		DAC_Generate Waveform_TriggerHW_ Init	This example describes how to use the DAC peripheral to generate a voltage waveform from digital data stream transfered by DMA. This example is based on the STM32L0xx DAC LL API; peripheral initialization done using LL initialization function to demonstrate LL init usage.	X	-	-	-	-	-

Table 1. STM32CubeL0 firmware examples (continued)

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Examples_LL	DMA	DMA_CopyFromFlashToMemory	This example describes how to use a DMA channel to transfer a word data buffer from Flash memory to embedded SRAM. Peripheral initialization done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		DMA_CopyFromFlashToMemory_Init	This example describes how to use a DMA channel to transfer a word data buffer from Flash memory to embedded SRAM. Peripherals initialization done using LL initialization function to demonstrate LL init usage.	X	-	-	-	-	-
	EXTI	EXTI_ToggleLedOnIT	This example describes how to configure the EXTI and use GPIOs using the STM32L0xx LL API to toggles the users LEDs available on the board when a user button is pressed. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		EXTI_ToggleLedOnIT_Init	This example describes how to configure the EXTI and use GPIOs using the STM32L0xx LL API to toggle the users LEDs available on the board when a user button is pressed. Peripheral initialization is done using LL initialization function to demonstrate LL init usage.	X	-	-	-	-	-
	GPIO	GPIO_InfiniteLedToggling	This example describes how to configure and use GPIOs through the LL API to toggle the users LEDs available on the board every 250 ms. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		GPIO_InfiniteLedToggling_Init	This example describes how to configure and use GPIOs through the LL API to toggle the users LEDs available on the board every 250 ms. Peripheral initialization done using LL initialization function to demonstrate LL init usage.	X	-	-	-	-	-

Table 1. STM32CubeL0 firmware examples (continued)

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Examples_LL	I2C	I2C_OneBoard_ AdvCommunication_ DMAAndIT	This example describes how to exchange some data between an I2C Master device using DMA mode and an I2C Slave device using IT mode. Peripheral initialization done using LL unitary services functions for optimization purpose (performance and size).	New	-	-	-	-	-
		I2C_OneBoard_ Communication_ DMAAndIT	This example describes how to transmit some data bytes from an I2C Master device using DMA mode to an I2C Slave device using Interrupt mode. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		I2C_OneBoard_ Communication_IT	This example describes how to receive data byte from an I2C Slave device using Interrupt mode to an I2C Master device using Interrupt mode. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		I2C_OneBoard_ Communication_IT_ Init	This example describes how to receive data byte from an I2C Slave device using Interrupt mode to an I2C Master device using Interrupt mode. Peripheral initialization is done using LL initialization function to demonstrate LL init usage.	X	-	-	-	-	-
		I2C_OneBoard_ ommunication_ PollingAndIT	This example describes how to transmit data bytes from an I2C Master device using Polling mode to an I2C Slave device using Interrupt mode. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		I2C_TwoBoards_ MasterRx_SlaveTx_IT	This example describes how to receive data byte from an I2C Slave device using Interrupt mode to an I2C Master device using Interrupt mode. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		I2C_TwoBoards_ MasterTx_SlaveRx	This example describes how to transmit some data bytes from an I2C Master device using Polling mode to an I2C Slave device using Interrupt mode. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		I2C_TwoBoards_ MasterTx_SlaveRx_ DMA	This example describes how to transmit some data bytes from an I2C Master device using DMA mode to an I2C Slave device using DMA mode. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		I2C_TwoBoards_Wake UpFromStop_IT	This example describes how to receive data byte from an I2C Slave device in Stop mode using Interrupt mode to an I2C Master device using Interrupt mode. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-

Table 1. STM32CubeL0 firmware examples (continued)

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Examples_LL	IWDG	IWDG_RefreshUntil UserEvent	This example describes how to configure the IWDG, ensure periodic counter update and generate an MCU IWDG reset when a user button is pressed. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
	LPTIM	LPTIM_PulseCounter	This example describes how to use the LPTIM in counter mode to generate a PWM output signal and update PWM duty cycle, based on a trigger provided by an external function generator. This example is based on the STM32L0xx LPTIM LL API; peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		LPTIM_PulseCounter_Init	This example describes how to use the LPTIM in counter mode to generate a PWM output signal and update PWM duty cycle, based on a trigger provided by an external function generator. This example is based on the STM32L0xx LPTIM LL API; peripheral initialization is done using LL initialization function to demonstrate LL init usage.	X	-	-	-	-	-
	LPUART	LPUART_WakeUp FromStop	This example shows how to configure 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 STM32L0xx LPUART LL API; peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		LPUART_WakeUp FromStop_Init	This example shows how to configure 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 STM32L0xx LPUART LL API; peripheral initialization is done using LL initialization function to demonstrate LL init usage.	X	-	-	-	-	-
	PWR	PWR_EnterStandby Mode	This example shows how to enter Standby mode and wake up from this mode using external RESET or wakeup interrupt.	X	-	-	-	-	-
		PWR_EnterStopMode	This example shows how to enter Stop mode.	X	-	-	-	-	-
		PWR_LPRunMode_ SRAM1	This example shows how to execute code (Low-power-run mode) from SRAM1.	X	-	-	-	-	-
		PWR_OptimizedRun Mode	This example shows how to increase/decrease frequency and V_{CORE} and to enter/exit Low-power-run mode.	X	-	-	-	-	-
	RCC	RCC_OutputSystem ClockOnMCO	This example describes how to configure MCO pin (PA8) to output the system clock.	X	-	-	-	-	-
		RCC_UseHSEas SystemClock	This example describes how to use the RCC LL API how to start the HSE and use it as system clock.	X	-	-	-	-	-
		RCC_UseHSI_PLLasS ystemClock	This example shows how to modify the PLL parameters in Run mode.	X	-	-	-	-	-

Table 1. STM32CubeL0 firmware examples (continued)

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Examples_LL	RNG	RNG_Generate RandomNumbers	This example shows how to configure RNG peripheral to generate 32-bit long random numbers. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		RNG_Generate RandomNumbers_IT	This example shows how to configure RNG peripheral to generate of 32-bit long random numbers, using interrupts. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
	RTC	RTC_Alarm	This example guides you through the different configuration steps by mean of LL API to ensure Alarm configuration and generation using the RTC peripheral. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		RTC_Alarm_Init	This example guides you through the different configuration steps by mean of LL API to ensure Alarm configuration and generation using the RTC peripheral. Peripheral initialization is done using LL initialization function to demonstrate LL init usage.	X	-	-	-	-	-
		RTC_Calendar	This example guides you through the different configuration steps by mean of LL API to ensure Calendar configuration using the RTC peripheral. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		RTC_ExitStandby WithWakeUpTimer	This example shows how to configure the RTC to wake up from Standby mode in using RTC Wakeup Timer. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		RTC_ProgrammingThe WakeUpTimer	This example shows how to configure the RTC in order to work with the WUT. Peripheral initialization done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	New	-	-
		RTC_Tamper	This example guides you through the different configuration steps by mean of LL API to ensure Tamper configuration using the RTC peripheral. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		RTC_TimeStamp	This example guides you through the different configuration steps by mean of LL API to ensure Time Stamp configuration using the RTC peripheral. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-

Table 1. STM32CubeL0 firmware examples (continued)

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Examples_LL	SPI	SPI_OneBoard_HalfDuplex_DMA	This example shows how to configure GPIO and SPI peripherals to transmit bytes from an SPI Master device to an SPI Slave device in DMA mode through the STM32L0xx SPI LL API. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		SPI_OneBoard_HalfDuplex_DMA_Init	This example shows how to configure GPIO and SPI peripherals to transmit bytes from an SPI Master device to an SPI Slave device in DMA mode through the STM32L0xx SPI LL API. Peripherals initialization is done using LL initialization function to demonstrate LL init usage.	X	-	-	-	-	-
		SPI_OneBoard_HalfDuplex_IT	This example shows how to configure GPIO and SPI peripherals to transmit bytes from an SPI Master device to an SPI Slave device in Interrupt mode through the STM32L0xx SPI LL API. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		SPI_TwoBoards_FullDuplex_DMA	This example shows how to ensure SPI data buffer transmission and reception in DMA mode through the STM32L0xx SPI LL API. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		SPI_TwoBoards_FullDuplex_IT	This example shows how to ensure SPI data buffer transmission and reception in Interrupt mode through the STM32L0xx SPI LL API. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
	TIM	TIM_DMA	This example provides a description of how to use DMA with TIMER update request to transfer data from memory to TIMER Capture Compare Register 3 (TIMx_CCR3). This example is based on the STM32L0xx TIM LL API; peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		TIM_InputCapture	This example shows how to use the TIM peripheral to measure the frequency of a periodic signal provided either by an external signal generator or by another timer instance. This example is based on the STM32L0xx TIM LL API, peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		TIM_OnePulse	This example shows 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}.	X	-	-	-	-	-

Table 1. STM32CubeL0 firmware examples (continued)

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Examples_LL	TIM	TIM_OutputCompare	This example shows how to configure the TIM peripheral to generate an output waveform in different output compare modes. This example is based the STM32L0xx TIM LL API, peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		TIM_PWMOutput	This example describes how to use a timer peripheral to generate a PWM output signal and update PWM duty cycle. This example is based on the STM32L0xx TIM LL API; peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		TIM_PWMOutput_Init	This example describes how to use a timer peripheral to generate a PWM output signal and update PWM duty cycle. This example is based on the STM32L0xx TIM LL API; peripheral initialization is done using LL initialization function to demonstrate LL init usage.	X	-	-	-	-	-
		TIM_TimeBase	This example shows how to configure the TIM peripheral to generate a time base. This example is based on the STM32L0xx TIM LL API; peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
	USART	USART_ Communication_Rx_IT	This example shows how to configure GPIO and USART peripheral for receiving characters from a PC HyperTerminal in Asynchronous mode using an interrupt. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		USART_ Communication_Rx_IT _Continuous	This example shows how to configure GPIO and USART peripherals to continuously receive characters from a PC HyperTerminal in Asynchronous mode using an interrupt. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		USART_ Communication_Rx_IT _Init	This example shows how to configure GPIO and USART peripherals to continuously receive characters from a PC HyperTerminal in Asynchronous mode using an interrupt. Peripheral initialization is done using LL initialization function to demonstrate LL init usage.	X	-	-	-	-	-
		USART_ Communication_Tx	This example shows how to configure GPIO and USART peripherals to asynchronously send characters to a PC HyperTerminal in Polling mode. If the transfer has not been completed within the allocated time, a timeout allows to exit from the sequence with a Timeout error code. This example is based on STM32L0xx USART LL API; peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-

Table 1. STM32CubeL0 firmware examples (continued)

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Examples_LL	USART	USART_ Communication_TxRx_ DMA	This example shows how to configure GPIO and USART peripherals to asynchronously send characters to/from a PC HyperTerminal in DMA mode. This example is based on STM32L0xx USART LL API; peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		USART_ Communication_Tx_IT	This example shows how to configure GPIO and USART peripherals to asynchronously send characters to a PC HyperTerminal in Interrupt mode. This example is based on STM32L0xx USART LL API; peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		USART_Hardware FlowControl	This example shows how to configure GPIO and USART peripherals to asynchronously receive characters from a PC HyperTerminal in Interrupt mode with Hardware Flow Control feature enabled. This example is based on STM32L0xx USART LL API; peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		USART_Sync Communication_ FullDuplex_DMA	This example shows how to configure GPIO, USART, DMA and SPI peripherals to transmit bytes from/to an USART peripheral to/from an SPI peripheral (in slave mode) by using DMA mode through the STM32L0xx USART LL API. Peripheral initialization done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		USART_Sync Communication_ FullDuplex_IT	This example shows how to configure GPIO, USART, DMA and SPI peripherals for transmitting bytes from/to an USART peripheral to/from an SPI peripheral (in slave mode) by using IT mode through the STM32L0xx USART LL API (SPI is using DMA for receiving/transmitting characters sent from/received by USART). Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
		USART_WakeUp FromStop	This example shows how to configure GPIO and USART peripherals to allow characters received on USART RX pin to wake up the MCU from low-power mode. This example is based on STM32L0xx USART LL API. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
	UTILS	UTILS_Configure SystemClock	This example describes how to use UTILS LL API to configure the system clock using PLL with HSI as source clock. The user application just needs to calculate PLL parameters using STM32CubeMX and call the UTILS LL API.	X	-	-	-	-	-
		UTILS_ReadDevice Info	This example describes how to Read UID, Device ID and Revision ID and save them into a global information buffer.	X	-	-	-	-	-

Table 1. STM32CubeL0 firmware examples (continued)

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Examples_LL	WWDG	WWDG_RefreshUntil UserEvent	This example describes how to configure WWDG, periodically update the counter and generate an MCU WWDG reset when a user button is pressed. Peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	X	-	-	-	-	-
	Total number of examples_ll: 82			**82**	**0**	**0**	**0**	**0**	**0**
Examples_MIX	ADC	ADC_Single Conversion_Trigger SW_IT	This example describes how to use a ADC peripheral to perform a single channel ADC conversion at each software start. It uses the Interrupt programming model (for Polling or DMA programming models, refer to other examples). This example is based on the STM32L0xx ADC HAL and LL API (LL API usage for performance improvement).	X	-	-	-	-	-
	CRC	CRC_Polynomial Update	This example provides a description of how to use CRC peripheral through the STM32L0xx CRC HAL and LL API (LL API usage for performance improvement). The CRC (Cyclic Redundancy Check) calculation unit computes the 8-bit long CRC code of a given buffer of 32-bit data words, based on a user-defined generating polynomial. The polynomial is first set manually to 0x9B that is X^8 + X^7 + X^4 + X^3 + X + 1. The generated polynomial value and length are then updated (set to 0x1021 that is X^16 + X^12 + X^5 + 1) for new CRC calculation; These updates are performed using CRC LL API.	X	-	-	-	-	-
	DMA	DMA_FLASHToRAM	This example provides a description of how to use DMA to transfer a word data buffer from Flash memory to embedded SRAM through the STM32L0xx DMA HAL & LL API (LL API usage for performance improvement).	X	-	-	-	-	-
	I2C	I2C_OneBoard_ ComSlave7_10bits_IT	This example describes how to perform I2C data buffer transmission/reception between master and 2 slaves with different address size (7-bit or 10-bit) through the STM32L0xx HAL and LL API (LL API usage for performance improvement), using an interrupt.	X	-	-	-	-	-
	PWR	PWR_STANDBY_RTC	This example shows how to enter Standby mode and wake up from this mode using external RESET or RTC Wakeup Timer through the STM32L0xx RTC & RCC HAL and LL API (LL API usage for performance improvement).	X	-	-	-	-	-
		PWR_STOP	This example shows how to enter Stop with low-power regulator 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	-	-	-	-	-

Table 1. STM32CubeL0 firmware examples (continued)

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Examples_MIX	SPI	SPI_FullDuplex_ComPolling	This example shows how to ensure SPI data buffer transmission/reception in Polling mode between two boards.	X	-	-	-	-	-
		SPI_HalfDuplex_ComPollingIT	This example shows how to ensure SPI data buffer transmission/reception between two boards by using Polling (LL Driver) an interrupt mode (HAL Driver).	X	-	-	-	-	-
	TIM	TIM_PWMInput	This example shows how to use the TIM peripheral to measure the frequency and duty cycle of an external signal.	X	-	-	-	-	-
	UART	UART_HyperTerminal_IT	This example describes how to use an UART to transmit data (transmit/receive) between a board and an PC HyperTerminal application in Interrupt mode. This example provides a description of how to use USART peripheral through the STM32L0xx UART HAL and LL API (LL API usage for performance improvement).	X	-	-	-	-	-
		UART_HyperTerminal_TxPolling_RxIT	This example describes how to use an UART to transmit data (transmit/receive) between a board and an PC HyperTerminal application both in Polling and Interrupt modes. This example provides a description of how to use USART peripheral through the STM32L0xx UART HAL and LL API (LL API usage for performance improvement).	X	-	-	-	-	-
	Total number of examples_mix: 11			**11**	**0**	**0**	**0**	**0**	**0**
Applications	FatFs	FatFs_uSD	This example provides a description on how to use the STM32Cube firmware with FatFs middleware component as a generic FAT file system module. The objective is to develop an application that exploits the features offered by FatFs to configure a microSD drive.	-	-	-	X	-	X
		FatFs_uSD_RTOS	This example provides a description on how to use STM32Cube firmware with FatFs middleware component as a generic FAT file system module. The objective is to develop an application that exploits the features offered by FatFs to configure a microSD drive in RTOS mode The SD card available on the Adafruit 1.8" TFT shield and mounted on top of the STM32 Nucleo board is used.	-	-	-	X	-	-

Table 1. STM32CubeL0 firmware examples (continued)

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Applications	FreeRTOS	FreeRTOS_LowPower	This application shows how to enter and exit low-power mode with CMSIS RTOS API.	X	X	X	X	-	X
		FreeRTOS_ LowPower_LPTIM	This application shows how to enter in Stop mode when all RTOS tasks are suspended.	-	-	-	New	-	-
		FreeRTOS_Mail	This application shows how to use mail queues with CMSIS RTOS API.	X	X	X	X	-	X
		FreeRTOS_Mutexes	This application shows how to use mutexes with CMSIS RTOS API.	X	X	X	X	-	X
		FreeRTOS_Queues	This application shows how to use message queues with CMSIS RTOS API.	X	X	X	X	-	X
		FreeRTOS_ Semaphore	This application shows how to use semaphores with CMSIS RTOS API.	X	X	X	X	-	X
		FreeRTOS_ SemaphoreFromISR	This application shows how to use semaphore from ISR with CMSIS RTOS API.	X	X	X	X	-	X
		FreeRTOS_Signal	This application shows how to use thread signaling using CMSIS RTOS API.	X	X	X	X	-	X
		FreeRTOS_ SignalFromISR	This application shows how to use thread signaling from an interrupt using CMSIS RTOS API.	X	X	X	X	-	X
		FreeRTOS_Thread Creation	This application shows how to implement a thread creation using CMSIS RTOS API.	X	X	X	X	-	X
		FreeRTOS_Timers	This application shows how to use timers of CMSIS RTOS API.	X	X	X	X	-	X
	IAP	IAP_Binary_Template	This directory contains a set of sources files to build an 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 preconfigured projects that describes how to build an application to be loaded into Flash memory using In-Application Programming (IAP, through USART).	-	-	-	-	-	X
	LCD	LCD_Display_Text	This example provides a description of how to use the STM32L0xx embedded LCD GLASS controller and how to configures the LCD to display a simple text and activate the different icons.	-	-	-	-	-	X
	Touch Sensing	TouchSensing_Linear	This firmware is a basic example on how to use the STMTouch driver with one linear sensor. The ECS and DTO are also used.	-	-	X	-	-	X

Table 1. STM32CubeL0 firmware examples (continued)

Level	Module name	Project name	Description	NUCLEO-L073RZ	NUCLEO-L031K6	STM32L053C8 DISCOVERY	NUCLEO-L053R8	NUCLEO-L011K4	STM32L073Z-EVAL
Applications	USB_Device	CDC_Standalone	This example is a part of the USB Device Library package using STM32Cube firmware. It describes how to use USB device application based on the Device Communication Class (CDC) following the PSTN sub-protocol in the STM32L0xx devices using the USB Device and UART peripherals.	-	-	-	-	-	X
		DFU_Standalone	This example is a part of the USB Device Library package using STM32Cube firmware. It describes how to use USB device application based on the Device Firmware Upgrade (DFU) on the STM32L0xx devices.	-	-	X	X	-	X
		HID_Standalone	This example is a part of the USB Device Library package using STM32Cube firmware.	-	-	-	X	-	X
		HID_Standalone_BCD	This example is a part of the USB Device Library package using STM32Cube firmware. This example describes how to use the BCD feature based on the USB HID device application.	-	-	New	New	-	New
		HID_Standalone_LPM	This example describes how to use USB device application based on the Human Interface (HID) with Link Power Management Protocol (LPM) on the STM32L073xx USB FS devices.	-	-	-	New	-	New
		HID_TSL_Standalone	This example is a part of the USB Device Library package using STM32Cube firmware.	-	-	X	-	-	-
		HID_TSL_Standalone_LPM	This example describes how to use USB device application based on the Human Interface (HID) with Link Power Management Protocol (LPM) on the STM32L053xx USB FS devices.	-	-	New	-	-	-
		MSC_Standalone	This example is a part of the USB Device Library package using STM32Cube firmware. It describes how to use USB device application based on the Mass Storage Class (MSC) on the STM32L0xx devices.	-	-	-	-	-	X
	Total number of applications: 73			**10**	**10**	**15**	**17**	**0**	**21**
Demonstrations	-	Adafruit_LCD_1_8_SD_Joystick	The provided demonstration firmware based on STM32Cube helps you to discover STM32 Cortex-M devices that can be plugged on a STM32 Nucleo board.	X	-	-	-	-	-
		Demo	The provided demonstration firmware based on STM32Cube helps you to discover STM32 Cortex-M devices that can be plugged on a STM32L073Z-EVAL board.	-	-	X	X	-	X
		Gravitech_4digits	The provided demonstration firmware based on STM32Cube helps you to discover STM32 Cortex-M devices that can be plugged on a STM32 Nucleo-32 board.	-	X	-	-	X	-
	Total number of demonstrations: 6			**1**	**1**	**1**	**1**	**1**	**1**
Total number of projects: 559				**170**	**69**	**75**	**92**	**53**	**100**

3 Revision history

Table 2. Document revision history

Date	Revision	Changes
06-Jul-2015	1	Initial release.
30-Nov-2015	2	Added SW4STM32 firmware in *Section 2: STM32CubeL0 examples*. Added NUCLEO-L073RZ, NUCLEO-L011K4, NUCLEO-L031K6, STM32L073Z-EVAL and 32L0538DISCOVERY in *Table 1: STM32CubeL0 firmware examples*.
13-Apr-2016	3	Added support for Low-Level Driver (LL).
15-Nov-2016	4	Updated *Section 2: STM32CubeL0 examples*. Updated *Table 1: STM32CubeL0 firmware examples*.