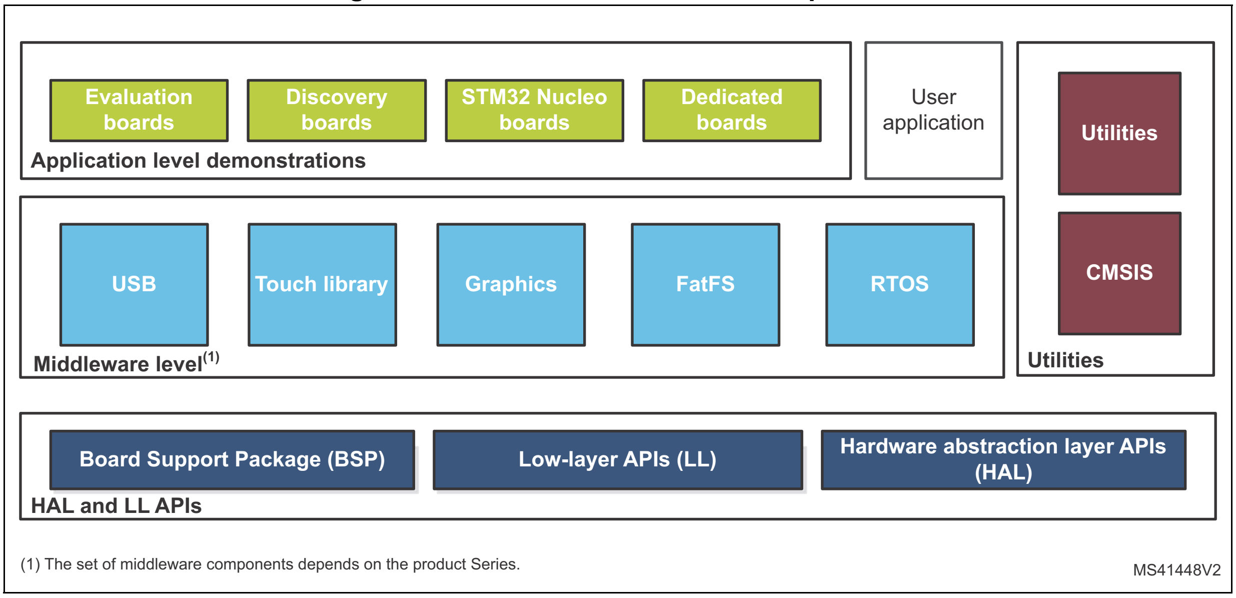
AN4706
Application note

STM32Cube firmware examples for STM32L1 Series

Introduction

The STM32CubeL1 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. STM32CubeL1 firmware components



Contents

1 Reference documents

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

- Latest release of STM32CubeL1 firmware package
- *Getting started with STM32CubeL1 for STM32L1 Series* user manual (UM1802)
- *STM32CubeL1 Nucleo demonstration firmware* user manual (UM1804)
- *Description of STM32L1 HAL and low-layer drivers* user manual (UM1816)
- *STM32Cube USB device library* user manual (UM1734)
- *Developing Applications on STM32Cube with RTOS* user manual *(*UM1722)
- *Developing applications on STM32Cube with FatFs* user manual (UM1721)

2 STM32CubeL1 examples

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

- **Examples**

  The examples use only the HAL and BSP drivers (middleware not used). Their objective is to demonstrate the product/peripherals features and usage. They are organized per peripheral (one folder per peripheral, e.g. TIM). 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 and Middleware not used), offering optimum implementation of typical use cases of the peripheral features and configuration procedures. They are organized per peripheral (a folder for each peripheral, e.g. TIM), and run exclusively on Nucleo board.
- **Examples_MIX**

  These examples use only HAL, BSP and LL drivers (Middleware are not used), having as objective to demonstrate how to use both HAL and LL APIs in the same application, to combine the advantages of both APIs (HAL offers high level and functionalities oriented APIs, with high portability level and hide product or IPs complexity to end user (Examples_LL offer low-level APIs at registers level with better optimization). The examples are organized per peripheral (a folder for each peripheral, e.g. 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 (a 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 built a firmware application using LL drivers on a given board.

The examples are located under *STM32Cube_FW_L1_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 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 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 STM32CubeL1 firmware package.

Table 1. STM32CubeL1 firmware examples

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO-L152RE
Templates_LL	-	Starter project	This projects provides a reference template through the LL API that can be used to build any firmware application.	X	X	X	X
	Total number of templates_ll: 4			**1**	**1**	**1**	**1**
Templates	-	Starter project	This projects provides a reference template that can be used to build any firmware application.	X	X	X	X
	Total number of templates: 4			**1**	**1**	**1**	**1**
Examples	-	BSP	This example provides a description of how to use the different BSP drivers.	-	-	X	-
	ADC	ADC_Analog Watchdog	This example provides a short description of how to use the ADC peripheral to perform conversions with analog watchdog and out-of-window interruptions enabled.	X	-	-	-
		ADC_LowPower	This example provides a short description of how to use the ADC peripheral to perform conversions with the following ADC low-power modes: auto-wait and auto-power off.	-	-	X	-
		ADC_Regular_injected_groups	This example provides a short description of how to use the ADC peripheral to perform conversions using the 2 ADC groups: regular group for ADC conversions on main stream and injected group for ADC conversions limited to specific events (conversions injected within main conversions stream).	-	X	-	-
		ADC_Sequencer	This example provides a short description of how to use the ADC peripheral with sequencer to convert several channels.	-	-	-	X

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO-L152RE
Examples	COMP	COMP_AnalogWatchdog	This example shows how to make an analog watchdog using the COMP peripherals in Window mode.	-	-	X	-
		COMP_Interrupt	This example shows how to configure the COMP peripheral to compare the external voltage applied to a specific pin with the Internal Voltage Reference. When the comparator input crosses (either rising or falling edges) the internal reference voltage V_{REFINT} (1.22 V), the comparator generates an interrupt.	X	-	-	-
		COMP_PWMSignalControl	This example shows how to configure the COMP peripheral to automatically hold TIMER PWM output on safe state (low level) as soon as the comparator output is High.	-	-	-	X
		COMP_PulseWidthMeasurement	This example shows how to configure the COMP peripheral to measure pulse widths. This method (measuring signal pulses using a comparator) is interesting when the external signal does not respect the V_{IL} and V_{IH} levels.	-	X	-	-
	CRC	CRC_Example	This example guides you through the different configuration steps by means of 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	-

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO -L152RE
Examples	Cortex	CORTEXM_MPU	This example presents the MPU features. Its purpose is to configure a memory area as privileged read-only region and attempt to perform read and write operations in different modes.	-	-	X	-
		CORTEXM_Mode Privilege	This example shows how to modify Thread mode privilege access and stack. Thread mode is entered on reset or when returning from an exception.	-	-	X	-
		CORTEXM_ SysTick	This example shows how to use the default SysTick configuration with a 1 ms timebase to toggle LEDs.	-	-	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	-
		DAC_Simple Conversion	This example provides a short description of how to use the DAC peripheral to perform the simple conversion of the 0xFF value in 8-bit right aligned mode. The result of the conversion can be retrieved by connecting PA4 (DAC channel1) to an oscilloscope.	-	-	-	X
	DMA	DMA_FLASH ToRAM	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	-
	FLASH	FLASH_Erase Program	This example describes how to configure and use the FLASH HAL API to erase and program the internal Flash memory.	-	-	X	-
		FLASH_Write Protection	This example describes how to configure and use the FLASH HAL API to enable and disable the write protection of the internal Flash memory.	-	-	-	X

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO -L152RE
Examples	FSMC	FSMC_NOR	This example describes how to configure the FSMC controller to access the NOR memory.	-	-	X	-
		FSMC_SRAM	This example describes how to configure the FSMC controller to access the SRAM.	-	-	X	-
		FSMC_SRAM_ DataMemory	This example describes how to configure the FSMC controller to access the SRAM including heap and stack.	-	-	X	-
	GPIO	GPIO_EXTI	This example shows how to configure external interrupt lines.	-	-	-	X
		GPIO_IOToggle	This example describes how to configure and use GPIOs through the HAL API.	X	X	X	X
	HAL	HAL_TimeBase_ TIM	This example describes how to customize the HAL time base using a general purpose timer instead of Systick as main source of time base.	-	-	X	-
	I2C	I2C_EEPROM	This example describes how to perform I2C data buffer transmission/reception via DMA. The communication uses an I2C EEPROM.	-	-	X	-
		I2C_TwoBoards_ AdvComIT	This example describes how to perform I2C data buffer transmission/reception between two boards, using an interrupt.	-	-	-	X
		I2C_TwoBoards_ ComDMA	This example describes how to perform I2C data buffer transmission/reception between two boards, via DMA.	-	-	-	X
		I2C_TwoBoards_ ComIT	This example describes how to perform I2C data buffer transmission/reception between two boards using an interrupt.	-	-	-	X
		I2C_TwoBoards_ ComPolling	This example describes how to perform I2C data buffer transmission/reception between two boards in Polling mode.	-	-	-	X

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO -L152RE
Examples	I2C (continued)	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.	-	-	-	X
		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.	-	-	-	X
	IWDG	IWDG_Reset	This example describes how to ensure IWDG reload counter and simulate a software fault that generates an MCU IWDG reset when a programmed time period has elapsed.	-	X	-	-
	LCD	LCD_Blink_ Frequency	This example provides a description of how to use the embedded LCD glass controller and configure the LCD Blink mode and Blink frequency.	X	-	-	-
		LCD_Contrast_ Control	This example provides a description of how to use the embedded LCD glass controller and configure the LCD contrast.	-	-	X	-
		LCD_Segments Drive	This example provides a description of how to use the embedded LCD controller to drive the Pacific Display LCD glass mounted on the board.	X	-	-	-
	OPAMP	OPAMP_ CALIBRATION	This example shows how to calibrate the OPAMP.	-	-	X	-
		OPAMP_Internal Follower	This example provides a short description of how to configure the OPAMP in Internal follower mode (unity gain). The signal applied on the OPAMP non-inverting input is reproduced on the OPAMP output.	-	-	-	X

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO -L152RE
Examples	PWR	PWR_LPRUN	This example shows how to enter Low-power run mode and exit from this mode using a key pushbutton.	-	-	-	X
		PWR_LPSLEEP	This example shows how to enter Low-power sleep mode and wake up from this mode using an interrupt.	-	-	-	X
		PWR_PVD	This example shows how to configure the programmable voltage detector using an external interrupt line.	-	-	X	-
		PWR_SLEEP	This example shows how to enter Sleep mode and wake up from this mode by using an interrupt.	-	-	-	X
		PWR_STANDBY	This example shows how to enter Standby mode and wake up from this mode using external RESET or WKUP pin.	-	-	X	-
		PWR_STANDBY_RTC	This example shows how to enter Standby mode and wake up from this mode using external RESET or RTC Wakeup Timer.	-	-	-	X
		PWR_STOP	This example shows how to enter Stop mode and wake up from this mode by using the RTC Wakeup timer event or an interrupt.	-	X	-	-
		PWR_STOP_RTC	This example shows how to enter Stop mode and wake up from this mode by using the RTC wakeup timer event connected to an interrupt.	-	-	-	X
	RCC	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	X	X

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO -L152RE
Examples	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	-
		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. It also demonstrates the tamper detection feature.	X	-	X	-
	SPI	SPI_FullDuplex_ComDMA	This example shows how to perform SPI data buffer transmission/reception between two boards via DMA.	-	-	-	X
		SPI_FullDuplex_ComIT	This example shows how to perform SPI data buffer transmission/reception between two boards by using an interrupt.	-	-	-	X
		SPI_FullDuplex_ComPolling	This example shows how to perform SPI data buffer transmission/reception in Polling mode between two boards.	-	-	-	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
		TIM_InputCapture	This example shows how to use the TIM peripheral to measure the frequency of an external signal.	-	-	X	-
		TIM_PWMOutput	This example shows how to configure the TIM peripheral in PWM (Pulse Width Modulation) mode.	-	-	-	X
		TIM_TimeBase	This example shows how to configure the TIM peripheral to generate a timebase of one second with the corresponding Interrupt request.	-	-	X	-

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO -L152RE
Examples	UART	UART_HyperTerminal_DMA	This example describes a UART transmission (transmit/receive) in DMA mode between a board and an HyperTerminal PC application.	-	-	X	-
		UART_Printf	This example shows how to reroute the C library printf function to the UART. This implementation outputs the printf message on the HyperTerminal using UART.	-	-	X	-
		UART_TwoBoards_ComDMA	This example describes a UART transmission (transmit/receive) in DMA mode between two boards.	-	-	-	X
		UART_TwoBoards_ComIT	This example describes a UART transmission (transmit/receive) in interrupt mode between two boards.	-	-	-	X
		UART_TwoBoards_ComPolling	This example describes a UART transmission (transmit/receive) in Polling mode between two boards.	-	-	-	X
	WWDG	WWDG_Example	This example guides you through the different configuration steps by means of the HAL API to perform periodic WWDG counter update and simulate a software fault that generates an MCU WWDG reset when a predefined time period has elapsed.	-	-	X	-
	Total number of examples: 69			**7**	**7**	**28**	**27**

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152D-EVAL	NUCLEO-L152RE
Examples_LL	ADC	ADC_Analog Watchdog	This example describes how to use an 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 STM32L1xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X
		ADC_Continuous Conversion_ TriggerSW	This example describes how to use an ADC peripheral to perform continuous ADC channel conversions, from a software start. This example is based on the STM32L1xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X
		ADC_Continuous Conversion_ TriggerSW_Init	This example describes how to use an ADC peripheral to perform continuous ADC channel conversions, from a software start. This example is based on the STM32L1xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X
		ADC_Continuous Conversion_ TriggerSW_ LowPower	This example describes how to use an ADC peripheral with ADC low-power features. This example is based on the STM32L1xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X
		ADC_Groups RegularInjected	This example describes how to use an ADC peripheral with both ADC groups (ADC group regular and ADC group injected) in their intended use case. This example is based on the STM32L1xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152D-EVAL	NUCLEO-L152RE
Examples_LL	ADC (continued)	ADC_MultiChannelSingleConversion	This example describes how to use an ADC peripheral to convert several channels. ADC conversions are performed successively in a scan sequence. This example is based on the STM32L1xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X
		ADC_SingleConversion_TriggerSW	This example describes how to use an ADC peripheral to perform a single ADC channel conversion at each software start. It uses the polling programming model (for programming models in Interrupt or DMA mode, refer to other examples). This example is based on the STM32L1xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X
		ADC_SingleConversion_TriggerSW_DMA	This example describes how to use an ADC peripheral to perform a single ADC channel conversion at each software start. It uses DMA transfer programming model (for programming models in Polling or Interrupt mode, refer to other examples). This example is based on the STM32L1xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO -L152RE
Examples_LL	ADC (continued)	ADC_SingleConversion_TriggerSW_IT	This example describes how to use an ADC peripheral to perform a single ADC channel conversion at each software start. It uses the interrupt programming model (for programming models in Polling or DMA mode, refer to other examples). This example is based on the STM32L1xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X
		ADC_SingleConversion_TriggerTimer_DMA	This example describes how to use an ADC peripheral to perform a single ADC channel conversion at each trigger event from timer. Conversion data are indefinitely transferred by DMA into a table (circular mode). This example is based on the STM32L1xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X
		ADC_TemperatureSensor	This example describes how to use an ADC peripheral to perform a single ADC conversion of the internal temperature sensor and calculate the temperature in Celsius degrees. It uses the Polling programming model (for programming models in Interrupt or DMA mode, refer to other examples). This example is based on the STM32L1xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO -L152RE
Examples_ LL	COMP	COMP_Compare GpioVsVrefInt_IT	This example describes how to use a comparator peripheral to compare, in Interrupt mode, a voltage level applied on a GPIO pin to the internal voltage reference (V_{REFINT}). This example is based on the STM32L1xx COMP LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X
		COMP_Compare GpioVsVrefInt_IT_ Init	This example describes how to use a comparator peripheral to compare, in interrupt mode, a voltage level applied on a GPIO pin to the internal voltage reference (V_{REFINT}). This example is based on the STM32L1xx COMP LL API. The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	-	X
	COMP (continued)	COMP_Compare GpioVsVrefInt_ Window_IT	This example describes how to use a pair of comparator peripherals to compare,, in interrupt mode, a voltage level applied on a GPIO pin to 2 thresholds: the internal voltage reference (V_{REFINT}) and a fraction of the internal voltage reference (V_{REFINT}/2). This example is based on the STM32L1xx COMP LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO -L152RE
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_Calculate AndCheck	This example shows how to configure the CRC calculation unit to get the CRC code of a given data buffer, based on a fixed generator polynomial (default value 0x4C11DB7). The peripheral initialization is 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 STM32L1xx DAC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X
		DAC_Generate Waveform_Trigger HW	This example describes how to use the DAC peripheral to generate a waveform voltage from a digital data stream transferred by DMA. This example is based on the STM32L1xx DAC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X
		DAC_Generate Waveform_Trigger HW_Init	This example describes how to use the DAC peripheral to generate a waveform voltage from a digital data stream transferred by DMA. This example is based on the STM32L1xx DAC LL API. The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	-	X

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO-L152RE
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. The peripheral initialization is 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. The peripheral initialization is 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 to toggle the user LEDs available on the board when a user button is pressed. This example is based on the STM32L1xx LL API. The 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 to toggle the user LEDs available on the board when a user button is pressed. This example is based on the STM32L1xx LL API. The 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 user LEDs available on the board each 250 ms. The 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 user LEDs available on the board each 250 ms. The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	-	X

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO -L152RE
Examples_ LL	I2C	I2C_OneBoard_ AdvCommunication_DMAAndIT	This example describes how to exchange data between an I2C Master device using DMA mode and an I2C Slave device using Interrupt mode. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X
		I2C_OneBoard_ Communication_ DMAAndIT	This example describes how to transmit data bytes from an I2C Master device using DMA mode to an I2C Slave device using Interrupt mode. The 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 a data byte from an I2C Slave device to an I2C Master device. Both devices operate in Interrupt mode. The 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 a data byte from an I2C Slave device to an I2C Master device. Both devices operate in Interrupt mode. The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	-	X
		I2C_OneBoard_ Communication_ 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. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO-L152RE
Examples_ LL	I2C (continued)	I2C_TwoBoards_ MasterRx_SlaveTx _IT	This example describes how to receive a data byte from an I2C Slave device to an I2C Master device. Both devices operate in Interrupt mode. The 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 data bytes from an I2C Master device using Polling mode to an I2C Slave device using Interrupt mode. The 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 to an I2C Slave device. Both devices operate in DMA mode. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X
	IWDG	IWDG_Refresh UntilUserEvent	This example describes how to configure the IWDG to ensure periodic counter update and generate an MCU IWDG reset when a user button is pressed. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO -L152RE
Examples_ LL	OPAMP	OPAMP_Follower	This example describes how to use a operational amplifier peripheral in Follower mode. To test the OPAMP, a voltage waveform is generated by the DAC peripheral and feeds the OPAMP input. This example is based on the STM32L1xx OPAMP LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X
		OPAMP_Follower_ Init	This example describes how to use a operational amplifier peripheral in PGA mode (programmable gain amplifier). To test the OPAMP, a voltage waveform is generated by the DAC peripheral and feeds the OPAMP input. This example is based on the STM32L1xx OPAMP LL API. The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	-	X
	PWR	PWR_Enter StandbyMode	This example shows how to enter Standby mode and wake up from this mode using an external RESET or a wakeup interrupt.	-	-	-	X
		PWR_Enter StopMode	This example shows how to enter Stop mode.	-	-	-	X
		PWR_LPRun Mode_SRAM1	This example shows how to execute code (Low-power run mode) from SRAM1.	-	-	-	X
		PWR_Optimized RunMode	This example shows how to increase/decrease frequency and V_{CORE} and how to enter/exit Low-power run Mode.	-	-	-	X
	RCC	RCC_Output SystemClock OnMCO	This example describes how to configure MCO pin (PA8) to output the system clock.	-	-	-	X
		RCC_UseHSE asSystemClock	This example describes how to use the RCC LL API how to start the HSE and use it as system clock.	-	-	-	X
		RCC_UseHSI_PLL asSystemClock	This example shows how to modify the PLL parameters in Run mode.	-	-	-	X

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO -L152RE
Examples_LL	RTC	RTC_Alarm	This example guides you through the different configuration steps by mean of LL API to configure and generate an alarm using the RTC peripheral. The 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 configure and generate an alarm using the RTC peripheral.The 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 HAL API to configure the RTC calendar. The 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 the system from Standby mode using RTC Wakeup Timer. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X
		RTC_Tamper	This example guides you through the different configuration steps by mean of LL API to configure the RTC tamper. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X
	RTC	RTC_TimeStamp	This example guides you through the different configuration steps by mean of LL API to configure the RTC timestamp. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO -L152RE
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. This example is based on the STM32L1xx SPI LL API. The 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. This example is based on the STM32L1xx SPI LL API. The peripheral 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. This example is based on the STM32L1xx SPI LL API. The 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 perform SPI data buffer transmission and reception in DMA mode. This example is based on the STM32L1xx SPI LL API. The 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 perform SPI data buffer transmission and reception in Interrupt mode. This example is based on the STM32L1xx SPI LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO -L152RE
Examples_ LL	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 STM32L1xx TIM LL API. The 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 delivered either by an external signal generator or by another timer instance. This example is based on the STM32L1xx TIM LL API. The 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
		TIM_Output Compare	This example shows how to configure the TIM peripheral to generate an output waveform in different output compare modes. This example is based on the STM32L1xx TIM LL API. The 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 STM32L1xx TIM LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO -L152RE
Examples_ LL	TIM (continued)	TIM_PWMOutput_ Init	This example describes how to use a timer peripheral to generate a PWM output signal and update PWM duty cycle. It is based on the STM32L1xx TIM LL API. The 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 timebase. It is based on the STM32L1xx TIM LL API. The 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 an HyperTerminal (PC) in Asynchronous mode and using an interrupt. The 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 peripheral to continuously receive characters from an HyperTerminal (PC) in Asynchronous mode and using an interrupt. The 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 peripheral to receive characters from an HyperTerminal (PC) in Asynchronous mode and using an interrupt. The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	-	X

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO -L152RE
Examples_ LL	USART (continued)	USART_ Communication_Tx	This example shows how to configure GPIO and USART peripherals to send characters asynchronously to an HyperTerminal (PC) in Polling mode. If the transfer could not be completed within the allocated time, a timeout allows to exit from the sequence with a Timeout error code. This example is based on STM32L1xx USART LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X
		USART_ Communication_TxRx_DMA	This example shows how to configure GPIO and USART peripheral to send characters asynchronously to/from an HyperTerminal (PC) in DMA mode. This example is based on STM32L1xx USART LL API. The 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 peripheral to send characters asynchronously to HyperTerminal (PC) in Interrupt mode. This example is based on STM32L1xx USART LL API. The 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 peripheral to receive characters asynchronously from an HyperTerminal (PC) in Interrupt mode with Hardware Flow Control feature enabled. This example is based on STM32L1xx USART LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO -L152RE
Examples_ LL	USART (continued)	USART_Sync Communication_ FullDuplex_DMA	This example shows how to configure GPIO, USART, DMA and SPI peripherals to transmit bytes between a USART peripheral and an SPI peripheral (in slave mode) by using DMA mode. This example is based on the STM32L1xx USART LL API. The peripheral initialization is 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 to transmit bytes between a USART peripheral and an SPI peripheral (in slave mode) in Interrupt mode. This example is based on the STM32L1xx USART LL API (SPI is using DMA for receiving/transmitting characters sent from/received by USART). The 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_Read DeviceInfo	This example describes how to Read UID, Device ID and Revision ID and save them into a global information buffer.	-	-	-	X
	WWDG	WWDG_Refresh UntilUserEvent	This example describes how to configure WWDG, periodically update counter and generate an MCU WWDG reset when a user button is pressed. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	X
	Total number of examples_ll: 73			0	0	0	73

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO -L152RE
Examples_ MIX	ADC	ADC_Single Conversion_ TriggerSW_IT	This example describes how to use an ADC peripheral to perform a single ADC channel conversion at each software start. It uses the interrupt programming model (for programming models in Polling or DMA mode, refer to other examples). This example is based on the STM32L1xx ADC HAL & LL API (LL API used for performance improvement).	-	-	-	X
	CRC	CRC_Calculate AndCheck	This example describes how to use the CRC peripheral through the STM32L1xx CRC HAL and LL API (LL API used for performance improvement). The CRC calculation unit computes the CRC code of a given buffer of 32-bit data words, using a fixed generator polynomial (0x4C11DB7).	-	-	-	X
	DMA	DMA_FLASH ToRAM	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 STM32L1xx DMA HAL and LL API (LL API used for performance improvement).	-	-	-	X
	I2C	I2C_OneBoard_ ComSlave7_10bits _IT	This example describes how to perform I2C data buffer transmission/reception between a master and 2 slaves with different address sizes (7-bit or 10-bit) and different maximum speeds (400 KHz or 100 KHz). This example is based on the STM32L1xx HAL and LL API (LL API used for performance improvement), using an interrupt.	-	-	-	X
	OPAMP	OPAMP_ CALIBRATION	This example describes how to use a operational amplifier peripheral with OPAMP calibration and operation. This example is based on the STM32L1xx OPAMP HAL and LL API (LL API used for performance improvement).	-	-	-	X

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO -L152RE
Examples _MIX	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. It is based on the STM32L1xx RTC and RCC HAL & LL API (LL API used 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 an external RESET or an wakeup interrupt (all the RCC functions calls use RCC LL API for footprint and performance improvements).	-	-	-	X
	SPI	SPI_FullDuplex_ ComPolling	This example shows how to perform SPI data buffer transmission/reception in Polling mode between two boards.	-	-	-	X
		SPI_HalfDuplex_ ComPollingIT	This example shows how to perform 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

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152D-EVAL	NUCLEO-L152RE
Examples_MIX	UART	UART_HyperTerminal_IT	This example describes how to use a UART to transmit data (transmit/receive) between a board and an HyperTerminal PC application in Interrupt mode. This example provides a description of how to use USART peripheral through the STM32L1xx UART HAL and LL API (LL API used for performance improvement).	-	-	-	X
		UART_HyperTerminal_TxPolling_RxIT	This example describes how to use a UART to transmit data (transmit/receive) between a board and an HyperTerminal PC application both in Polling and Interrupt mode. This example provides a description of how to use USART peripheral through the STM32L1xx UART HAL and LL API (LL API used for performance improvement).	-	-	-	X
	Total number of examples_mix: 12			**0**	**0**	**0**	**12**

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO -L152RE
Applications	FatFs	FatFs_uSD	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 using most of the features offered by FatFs to configure a microSD drive.	-	-	X	-
	FreeRTOS	FreeRTOS_ LowPower	This application shows how to enter and exit low-power mode with CMSIS RTOS API.	-	-	X	-
		FreeRTOS_Mail	This application shows how to use mail queues with CMSIS RTOS API.	-	-	X	-
		FreeRTOS_ Mutexes	This application shows how to use mutexes with CMSIS RTOS API.	-	-	X	-
		FreeRTOS_ Queues	This application shows how to use message queues with CMSIS RTOS API.	-	-	X	-
		FreeRTOS_ Semaphore	This application shows how to use semaphores with CMSIS RTOS API.	-	-	X	-
		FreeRTOS_ SemaphoreFromISR	This application shows how to use semaphore from ISR with CMSIS RTOS API.	-	-	X	-
		FreeRTOS_Signal	This application shows how to use thread signaling using CMSIS RTOS API.	-	-	X	-
		FreeRTOS_Signal FromISR	This application shows how to use thread signaling from an interrupt using CMSIS RTOS API.	-	-	X	-
		FreeRTOS_Thread Creation	This application shows how to implement a thread creation using CMSIS RTOS API.	X	X	X	X
		FreeRTOS_Timers	This application shows how to use timers of CMSIS RTOS API.	-	-	X	-
	STemWin	STemWin_ HelloWorld	This application shows how to implement a simple "Hello World" example based on STemWin.	-	-	X	-

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO -L152RE
Applications	Touch Sensing	TouchSensing_ Linear_hwacq	This firmware is a basic example on how to use the STMTouch driver with one linear sensor using the hardware acquisition.	-	-	X	-
		TouchSensing_ Linear_swacq	This firmware is a basic example on how to use the STMTouch driver with one linear sensor using the software acquisition.	-	-	X	-
	USB_ Device	CDC_Standalone	This application shows how to use the USB device application based on the Device Communication Class (CDC) following the PSTN subprotocol, and using the USB Device and UART peripherals.	-	-	X	-
		CustomHID_ Standalone	This application shows how to use the USB device application based on the Custom HID Class.	-	-	X	-
		DFU_Standalone	This example is a part of the USB Device Library package, which uses STM32Cube firmware. It describes how to use USB device application based on the Device Firmware Upgrade (DFU) on the STM32L1xx devices.	-	-	X	X
		HID_Standalone	This example is a part of the USB Device Library package, which uses STM32Cube firmware. It describes how to use USB device application based on the Human Interface (HID) on the STM32L1xx devices.	-	-	X	X
		MSC_Standalone	This application shows how to use the USB device application based on the Mass Storage Class (MSC).	-	-	X	-
	Total number of applications: 24			**1**	**1**	**19**	**3**

Table 1. STM32CubeL1 firmware examples (continued)

Level	Module name	Project name	Description	32L152C DISCOVERY	32L100C DISCOVERY	STM32L152 D-EVAL	NUCLEO -L152RE
Demonstra-tions	-	Adafruit_LCD_1_8_SD_Joystick	This demonstration firmware is based on STM32Cube. It helps you to discover STM32 Cortex®-M devices that can be plugged on a STM32 Nucleo board.	-	-	-	X
		LED_Blinking	This demonstration uses the USER button as well as LED3 and LED4 available on the board.	-	X	-	-
	Total number of demonstrations: 2			**0**	**1**	**0**	**1**
Total number of projects: 188				**10**	**11**	**49**	**118**

3 Revision history

Table 2. Document revision history

Date	Revision	Changes
27-May-2015	1	Initial release.
11-Jul-2016	2	Added SW4STM32 firmware in *Section 2: STM32CubeL1 examples*. Added support for Low-layer drivers.
26-Apr-2017	3	Updated *Figure 1: STM32CubeL1 firmware components*. Added: Templates_LL on all boards. HAL_TimeBase_TIM for STM32L152D_EVAL. Removed HAL_TimeBase for STM32L152D_EVAL.
13-Sep-2017	4	Updated *Table 1: STM32CubeL1 firmware examples*.