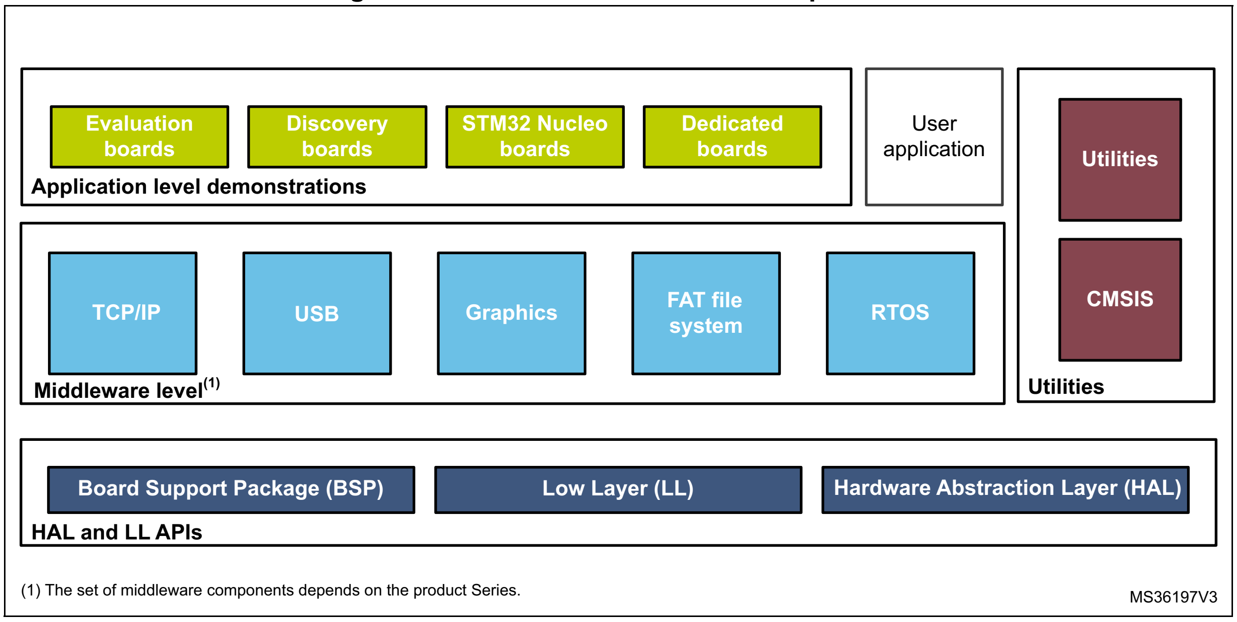
AN4724
Application note

STM32Cube firmware examples for STM32F1 Series

Introduction

The STM32CubeF1 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. STM32CubeF1 firmware components



Contents

1 Reference documents

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

- Latest release of STM32CubeF1 firmware package
- *Getting started with the STM32CubeF1 firmware package for the STM32F1 series* (UM1847)
- *STM32CubeF1 Nucleo demonstration firmware* (UM1853)
- *Description of STM32F1xx HAL and low-layer drivers* (UM1850)
- *STM32Cube USB Device library* (UM1734)
- *STM32Cube USB host library* (UM1720)
- *Developing applications on STM32Cube with FatFs* (UM1721)
- *Developing Applications on STM32Cube with RTOS (*UM1722)
- *Developing applications on STM32Cube with LwIP TCP/IP stack* (UM1713)
- *STM32Cube Ethernet IAP example* (UM1709)

2 STM32CubeF1 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 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 board.
- **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 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 quickly building a firmware application on a given board.

The examples are located under *STM32Cube_FW_F1_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 STM32CubeF1 firmware package.

Table 1. STM32CubeF1 firmware examples

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM32210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Templates_LL	-	Starter project	This projects provides a reference template based 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	-	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_DualMode Interleaved	This example provides a short description of how to use two ADC peripherals to perform conversions in interleaved dual-mode.	-	-	-	X
		ADC_Regular_injected_ groups	This example provides a short description of how to use the ADC peripheral to perform conversions using the two 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	-	-	X
		ADC_Sequencer	This example provides a short description of how to use the ADC peripheral with sequencer, to convert several channels.	-	X	-	-
	CAN	CAN_Networking	This example shows how to configure the CAN peripheral to send and receive CAN frames in normal mode.	-	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	X	X	X

Table 1. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Examples	Cortex	CORTEXM_MPU	This example presents the MPU feature. The example purpose is to configure a memory region 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 a simple conversion.	-	X	-	-
	DMA	DMA_FLASHToRAM	This example provides a description of how to use a DMA channel to transfer a word data buffer from Flash memory to embedded SRAM through the HAL API.	-	X	-	-
	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. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Examples	FSMC	FSMC_NAND	This example describes how to configure the FSMC controller to access the NAND memory.	-	X	-	-
		FSMC_NOR	This example describes how to configure the FSMC controller to access the NOR memory.	-	X	-	-
		FSMC_NOR_Code Execute	This example describes how to build an application to be loaded into the NOR memory mounted on board and then execute it from internal Flash memory.	-	X	-	-
		FSMC_SRAM	This example describes how to configure the FSMC controller to access the SRAM.	-	X	-	-
		FSMC_SRAM_Data Memory	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_RTC_ALARM	This example describes how to customize the HAL timebase using the RTC Alarm instead of the SysTick as main timebase source. The discovery board user button (connected to EXTI Line0) will be used to Suspend or Resume tick increment.	X	X	X	X
		HAL_TimeBase_TIM	This example describes how to customize the HAL timebase using a general purpose timer instead of the SysTick as main timebase source.	X	X	X	X

Table 1. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-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	-
		I2C_TwoBoards_ComDMA	This example describes how to perform I2C data buffer transmission/reception between two boards, via DMA.	-	X	X	-
		I2C_TwoBoards_ComIT	This example describes how to perform I2C data buffer transmission/reception between two boards using an interrupt.	-	X	X	-
		I2C_TwoBoards_ComPolling	This example describes how to perform I2C data buffer transmission/reception between two boards in Polling mode.	-	X	X	-
		I2C_TwoBoards_RestartAdvComIT	This example describes how to perform I2C data buffer sequential transmission/reception between two boards using an interrupt.	-	X	X	-
		I2C_TwoBoards_RestartComIT	This example describes how to perform I2C data buffer sequential transmission/reception between two boards using an interrupt.	-	X	X	-
	I2S	I2S_Audio	This example provides basic implementation of audio features.	-	-	-	X
	IWDG	IWDG_Example	This example describes how to reload the IWDG counter and to simulate a software fault by generating an MCU IWDG reset when a programmed time period has elapsed.	-	-	-	X

Table 1. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Examples	PWR	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	-	-
		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 enters the system to Standby mode and wake up from this mode using external RESET or WKUP pin.	-	-	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
	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	-
		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	-	-
		RTC_LSI	This example demonstrates and explains how to use the LSI clock source auto-calibration to get a precise RTC clock.	-	-	-	X
		RTC_LowPower_ STANDBY	This example shows how to enter Standby mode and wake up from this mode using RTC Alarm Event connected to EXTI_Line17.	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 demonstrate the tamper detection feature.	-	X	-	-
	SMARTCARD	SMARTCARD_T0	This example describes a firmware Smartcard Interface based on the USART peripheral. The main purpose of this firmware example is to provide resources that facilitate the development of an application using the USART peripheral in Smartcard mode.	-	X	-	X

Table 1. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Examples	SPI	SPI_FullDuplex_ComDMA	This example shows how to perform SPI data buffer transmission/reception between two boards via DMA.	X	-	X	-
		SPI_FullDuplex_ComIT	This example shows how to ensure SPI data buffer transmission/reception between two boards by using an interrupt.	X	-	X	-
		SPI_FullDuplex_ComPolling	This example shows how to ensure SPI data buffer transmission/reception in Polling mode between two boards.	X	-	X	-
	TIM	TIM_ComplementarySignals	This example shows how to configure the TIM1 peripheral to generate three complementary TIM1 signals, insert a defined dead time value, and use the break feature and lock the desired parameters.	-	X	-	-
		TIM_DMA	This example provides a description of how to use DMA with TIM1 Update request to transfer data from memory to TIM1 Capture Compare Register 3 (TIM1_CCR3).	-	X	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 1 second timebase with the corresponding Interrupt request.	X	X	X	X

Table 1. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Examples	UART	UART_HyperTerminal_ DMA	This example shows how to perform UART data buffer transmission and reception with DMA. The communication is done with the HyperTerminal PC application.	X	-	X	X
		UART_Printf	This example shows how to reroute the C library printf function to the UART. It outputs a message sent by the UART to the HyperTerminal.	X	X	X	X
		UART_TwoBoards_ ComDMA	This example describes an UART transmission (transmit/receive) in DMA mode between two boards.	X	-	X	X
		UART_TwoBoards_ ComIT	This example describes an UART transmission (transmit/receive) in Interrupt mode between two boards.	X	-	X	X
		UART_TwoBoards_ ComPolling	This example describes an UART transmission (transmit/receive) in Polling mode between two boards.	X	-	X	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: 98			**18**	**34**	**27**	**19**

Table 1. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Examples_ LL	ADC	ADC_Analog Watchdog	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 STM32F1xx ADC LL API. The peripheral initialization is 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 STM32F1xx ADC LL API. The peripheral initialization is 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 STM32F1xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	X	-
		ADC_GroupsRegular Injected	This example describes how to use a ADC peripheral with both ADC groups (ADC group regular and ADC group injected) in their intended use case. This example is based on the STM32F1xx ADC LL API. The peripheral initialization is 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 STM32F1xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	X	-
		ADC_MultimodeDual Interleaved	This example describes how to use several ADC peripherals in multimode mode interleaved. This example is based on the STM32F1xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	X	-

Table 1. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Examples_ LL	ADC	ADC_Single Conversion_Trigger SW	This example describes how to use a ADC peripheral to perform a single ADC channel conversion at each software start. It uses the polling programming model (for details on interrupt or DMA transfer programming models, refer to other examples). This example is based on the STM32F1xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	X	-
		ADC_Single Conversion_Trigger SW_DMA	This example describes how to use an ADC peripheral to perform a single ADC channel conversion at each software start. It uses the DMA transfer programming model (for details on polling or interrupt programming models, refer to other examples); This example is based on the STM32F1xx ADC LL API. The 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 an ADC peripheral to perform a single ADC channel conversion at each software start. It uses the interrupt programming model (for details on polling or DMA transfer programming models, refer to other examples).This example is based on the STM32F1xx ADC LL API. The 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 an ADC peripheral to perform a single ADC channel conversion at each timer trigger event. Converted data are indefinitely transferred by DMA into a table (circular mode); This example is based on the STM32F1xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	X	-

Table 1. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Examples_ LL	ADC	ADC_Temperature Sensor	This example describes how to use an ADC peripheral to perform a single ADC conversion of the internal temperature sensor and to calculate the temperature in Celsius degrees. This example using the polling programming model (for details on interrupt or DMA transfer programming models, refer to other examples). This example is based on the STM32F1xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	X	-
	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_CalculateAnd Check	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 STM32F1xx DAC LL API. The peripheral initialization is 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 waveform voltage from a digital data stream transferred by DMA. This example is based on the STM32F1xx DAC LL API. The peripheral initialization is 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 waveform voltage from a digital data stream transfered by DMA. This example is based on the STM32F1xx DAC LL API. The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	X	-	-

Table 1. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Examples_ LL	DMA	DMA_CopyFromFlashTo Memory	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_CopyFromFlashTo Memory_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. It is based on the STM32F1xx 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 STM32F1xx LL API. The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	X	-
	GPIO	GPIO_InfiniteLed Toggling	This example describes how to configure and use GPIOs to toggle the user LEDs available on the board every 250 ms. This example is based on the STM32F1xx LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	X	-
		GPIO_InfiniteLed Toggling_Init	This example describes how to configure and use GPIOs to toggle the user LEDs available on the board every 250 ms. This example is based on the STM32F1xx LL API. The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	X	-

Table 1. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Examples_LL	I2C	I2C_OneBoard_AdvCommunication_DMAAndIT	This example describes how to exchange data between an I2C Master device in DMA mode and an I2C Slave device in 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 one 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 one 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	-
		I2C_TwoBoards_MasterRx_SlaveTx_IT	This example describes how to receive one 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	-

Table 1. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Examples_LL	I2C	I2C_TwoBoards_MasterTx_SlaveRx_DMA	This example describes how to transmit data bytes from an I2C Master device using DMA mode to an I2C Slave device using DMA mode.The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	X	-
	IWDG	IWDG_RefreshUntilUserEvent	This example describes how to configure the IWDG to ensure period 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	-
	PWR	PWR_EnterStandbyMode	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_MAINREGU mode.	-	-	X	-
	RCC	RCC_OutputSystemClockOnMCO	This example describes how to configure MCO pin (PA8) to output the system clock.	-	-	X	-
		RCC_UseHSEasSystemClock	This example describes how to use the RCC LL API to start the HSE and use it as system clock.	-	-	X	-
		RCC_UseHSI_PLLasSystemClock	This example shows how to modify the PLL parameters in run time.	-	-	X	-

Table 1. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Examples_ LL	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. 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 ensure Alarm configuration and generation 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_Tamper	This example guides you through the different configuration steps by mean of LL API to ensure Tamper configuration using the RTC peripheral. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	X	-

Table 1. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Examples_ LL	SPI	SPI_OneBoard_Half Duplex_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. The example is based on the STM32F1xx SPI LL API.The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	X	-
		SPI_OneBoard_Half Duplex_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. The example is based on the STM32F1xx SPI LL API.The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	X	-
		SPI_OneBoard_Half Duplex_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. The example is based on the STM32F1xx SPI LL API.The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	X	-
		SPI_TwoBoards_Full Duplex_DMA	This example shows how to ensure SPI data buffer transmission and reception in DMA mode. The example is based on the STM32F1xx SPI LL API.The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	X	-
		SPI_TwoBoards_Full Duplex_IT	This example shows how to ensure SPI Data buffer transmission and reception in Interrupt mode. The example is based on the STM32F1xx SPI LL API.The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	X	-

Table 1. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Examples_LL	TIM	TIM_BreakAndDeadtime	This example shows how to configure the timer to perform the following operations: – generate three center-aligned PWM and complementary PWM signals – insert a defined dead time value – use the break feature – lock the desired parameters This example is based on the STM32F1xx TIM LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	X	-
		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 uses the STM32F1xx 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 timer peripheral to measure the frequency of a periodic signal provided either by an external signal generator or by another timer instance. This example uses the STM32F1xx 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}. This example is based on the STM32F1xx TIM LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	X	-

Table 1. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Examples_ LL	TIM	TIM_OutputCompare	This example shows how to configure the timer peripheral to generate an output waveform in different output compare modes. This example uses the STM32F1xx 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 uses the STM32F1xx TIM LL API. The 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 using the STM32F1xx 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 timer peripheral to generate a time base. This example uses the STM32F1xx TIM LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	X	-

Table 1. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Examples_ LL	USART	USART_ Communication_Rx_IT	This example shows how to configure GPIO and USART peripheral to receive characters from an HyperTerminal (PC) in Asynchronous mode using Interrupt mode.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 using Interrupt mode.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 using Interrupt mode.The 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 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 STM32F1xx 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 STM32F1xx 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 an HyperTerminal (PC) in Interrupt mode. This example is based on STM32F1xx USART LL API.The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	X	-

Table 1. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Examples_LL	USART	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 STM32F1xx USART LL API.The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	X	-
		USART_Sync Communication_Full Duplex_DMA	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 DMA mode through the STM32F1xx USART LL API.The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	X	-
		USART_Sync Communication_Full Duplex_IT	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 Interrupt mode through the STM32F1xx USART LL API (the SPI uses DMA to receive/transmit the 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_ReadDevice Info	This example describes how to read UID, Device ID and Revision ID and save them into a global information buffer.	-	-	X	-
	WWDG	WWDG_RefreshUntil UserEvent	This example describes how to configure the WWDG, periodically update the 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: 65			0	4	61	0

Table 1. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Examples_ MIX	ADC	ADC_Single Conversion_Trigger SW_IT	This example describes how to use the ADC to perform a single ADC channel conversion at each software start. This example uses the interrupt programming model (for programming models in Polling or DMA mode, refer to other examples). This example is based on the STM32F1xx ADC HAL and LL API (LL API usage for performance improvement).	-	X	-	-
	CRC	CRC_CalculateAnd Check	This example provides a description of how to use CRC peripheral through the STM32F1xx CRC HAL & LL API (LL API used for performance improvement); Fixed generator polynomial used in CRC IP is CRC-32 (Ethernet) polynomial: 0x4C11DB7.	-	-	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 STM32F1xx DMA HAL and LL API (LL API used for performance improvement).	-	-	X	-
	I2C	I2C_OneBoard_Com Slave7_10bits_IT	This example describes how to perform I2C data buffer transmission/reception between one master and two slaves with different address sizes (7-bit or 10-bit) and at different maximum speeds (400 KHz or 100 KHz). This example uses the STM32F1xx I2C HAL and LL API (LL API usage for performance improvement) and an interrupt.	-	-	X	-
	PWR	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	-
	SPI	SPI_FullDuplex_Com Polling	This example shows how to ensure SPI data buffer transmission/reception in Polling mode between two boards.	-	-	X	-
		SPI_HalfDuplex_Com PollingIT	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	-

Table 1. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Examples_ MIX	TIM	TIM_6Steps	This example shows how to configure the TIM1 peripheral to generate 6-steps PWM signal. The STM32F1xx TIM1 peripheral allows programming in advance the configuration for the next TIM1 output behavior (or step) and changing the configuration of all the channels simultaneously. This operation is possible when the COM (commutation) event is used. This example is based on the STM32F1xx TIM HAL and LL API (LL API usage for performance improvement).	-	-	X	-
		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 HyperTerminal PC application in Interrupt mode. This example provides a description of how to use USART peripheral through the STM32F1xx 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 HyperTerminal PC application both in Polling and Interrupt modes. This example provides a description of how to use USART peripheral through the STM32F1xx UART HAL and LL API (LL API usage for performance improvement).	-	-	X	-
	Total number of examples_mix: 11			**0**	**1**	**10**	**0**

Table 1. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Applications	EEPROM	EEPROM_Emulation	Please refer to AN2594 for further details regarding this application.	-	-	X	-
	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	-	X
	FreeRTOS	FreeRTOS_Mail	This application shows how to use mail queues with CMSIS RTOS API.	-	X	-	-
		FreeRTOS_Signal	This application shows how to use thread signaling using CMSIS RTOS API.	-	X	-	-
		FreeRTOS_SignalFromISR	This application shows how to use thread signaling from an interrupt using CMSIS RTOS API.	-	X	-	-
		FreeRTOS_Thread Creation	This example creates two threads with the same priority, which execute in a periodic cycle of 5 seconds for Thread 1 and 10 seconds for Thread 2.	-	X	X	X
	IAP	IAP_Binary_Template	This directory contains a set of sources files that build the application to be loaded into Flash memory using In-Application Programming (IAP) through the USART.	-	X	-	X
		IAP_Main	This directory contains a set of sources files and pre-configured projects that describes how to build an application to be loaded into Flash memory using In-Application Programming (IAP) through USART).	-	X	-	X

Table 1. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Applications	LwIP	LwIP_TCP_Echo_Client	This application shows how to run TCP Echo Client application based on Raw API of LwIP TCP/IP stack. To run this application, open a command prompt window on the remote PC.	-	-	-	X
		LwIP_TCP_Echo_Server	This application shows how to run TCP Echo Server application based on Raw API of LwIP TCP/IP stack. To run this application, open a command prompt window on the remote PC.	-	-	-	X
		LwIP_UDP_Echo_Client	This application shows how to run a UDP Echo Client application based on Raw API of LwIP TCP/IP stack. To run this application, open a command prompt window on the remote PC.	-	-	-	X
		LwIP_UDP_Echo_Server	This application shows how to run UDP Echo Server application based on Raw API of LwIP TCP/IP stack. To run this application, open a command prompt window on the remote PC.	-	-	-	X
	STemWin	STemWin_HelloWorld	This application shows how to implement a simple "Hello World" example based on STemWin.	-	X	-	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 using the USB Device and UART peripherals.	-	X	-	X
		CustomHID_Standalone	This application shows how to use the USB device application based on the Custom HID Class.	-	X	-	X
		DFU_Standalone	This application presents a compliant implementation of the Device Firmware Upgrade (DFU) capability for programming the embedded Flash memory through the USB peripheral.	-	X	-	X
		HID_Standalone	This application shows how to use the USB device application based on the Human Interface (HID).	-	X	X	X
		MSC_Standalone	This application shows how to use the USB device application based on the Mass Storage Class (MSC).	-	X	-	X

Table 1. STM32CubeF1 firmware examples (continued)

Level	Module Name	Project Name	Description	STM32VL DISCOVERY	STM3210E-EVAL	NUCLEO-F103RB	STM3210C-EVAL
Applications	USB_Host	CDC_Standalone	This application shows how to use the USB host application based on the CDC class.	-	-	-	X
		HID_RTOS	This application shows how to use the USB host application based on the HID class.	-	-	-	X
		HID_Standalone	This application shows how to use the USB host application based on the HID class.	-	-	-	X
		MSC_RTOS	This application shows how to use the USB host application based on the Mass Storage Class (MSC).	-	-	-	X
		MSC_Standalone	This application shows how to use the USB host application based on the Mass Storage Class (MSC).	-	-	-	X
	Total number of applications: 35			**0**	**13**	**3**	**19**
Demonstra-tions	-	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	-
	Total number of demonstrations: 1			**0**	**0**	**1**	**0**
Total number of projects: 218				**20**	**54**	**104**	**40**

3 Revision history

Table 2. Document revision history

Date	Revision	Changes
06-Jul-2015	1	Initial release.
20-Apr-2017	2	Updated *Figure 1: STM32CubeF1 firmware components* and *Section 2: STM32CubeF1 examples* to add Low Layer (LL).