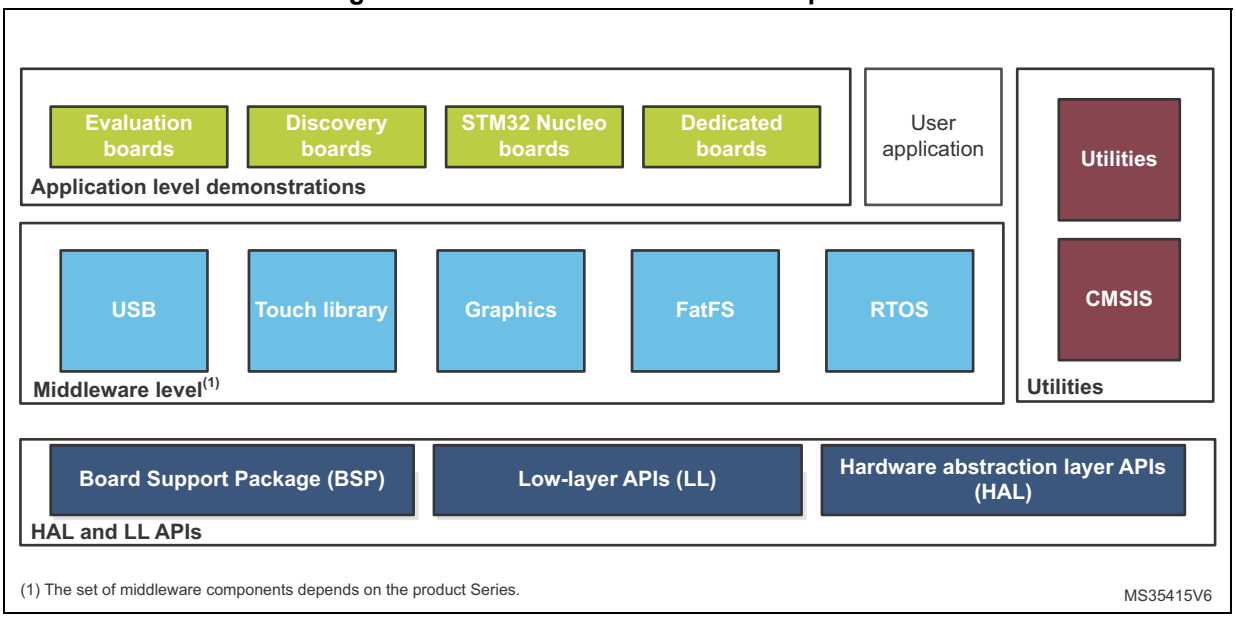
AN4735
Application note

STM32Cube firmware examples for STM32F0 Series

Introduction

The STM32CubeF0 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* and *Table 2*).

Figure 1. STM32CubeF0 firmware components



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

MS35415V6

Contents

1 Reference documents

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

- Latest release of STM32CubeF0 firmware package
- *Getting started with the STM32CubeF0 firmware package for STM32F0 Series* (UM1779)
- *Description of STM32F0xx HAL drivers* (UM1785)
- *STM32CubeF0 Nucleo demonstration firmware* (UM1787)
- *Demonstration firmware for STM32091C-EVAL board* (UM1819)
- *STM32Cube USB Device library (UM1734)*
- *Developing Applications on STM32Cube with FatFS (*UM1721)
- *Developing Applications on STM32Cube with RTOS (*UM1722)

2 STM32CubeF0 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 boards.
- **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 by hiding product/IPs complexity to the end user
  - LL provides 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 boards.
- **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 requiring 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 performances.
- **Template project:** the template project is provided to allow the user to quickly build a firmware application using HAL and BSP drivers on a given board.
- **Template_LL project:** the template LL project is provided to allow the user to quickly build a firmware application using LL drivers on a given board.

The examples are located under *STM32Cube_FW_F0_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 and \SW4STM32 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 2 contains the list of examples provided with STM32CubeF0 firmware package. The board mnemonics used in the column headers are also used in the firmware package.

The correspondence with STMicroelectronics board references is detailed in *Table 1*.

Table 1. Board references

Reference	RPN	Mnemonic used in *Table 2* and in the firmware
STM32F0308-Discovery	STM32F0308DISCOVERY	32F0308DISCOVERY
STM32F072B-Discovery	STM32F072BDISCOVERY	32F072BDISCOVERY
STM32F030R8-Nucleo	NUCLEO-F030R8	NUCLEO-F030R8
STM32F031K6-Nucleo	NUCLEO-F031K6	NUCLEO-F031K6
STM32F042K6-Nucleo	NUCLEO-F042K6	NUCLEO-F042K6
STM32F070RB-Nucleo	NUCLEO-F070RB	NUCLEO-F070RB
STM32F072RB-Nucleo	NUCLEO-F072RB	NUCLEO-F072RB
STM32F091RC-Nucleo	NUCLEO-F091RC	NUCLEO-F091RC
STM32072B_EVAL	STM32072B-EVAL	STM32072B-EVAL
STM32091C_EVAL	STM32091C-EVAL	STM32091C-EVAL

Table 2. STM32CubeF0 firmware examples

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
Templates_LL	-	Starter project	This project provides a reference template through the LL API that can be used to build any firmware application.	X	X	X	X	X	X	X	X	X	X
Total number of templates_ll: 10				**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Templates	-	Starter project	This project provides a reference template that can be used to build any firmware application.	X	X	X	X	X	X	X	X	X	X
Total number of templates: 10				**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Examples	-	BSP	This example provides a description of how to use the different BSP drivers.	-	-	-	-	-	X	-	-	-	X
	ADC	ADC_AnalogWatchdog	This example provides a short description of how to use the ADC peripheral to perform conversions with analog watchdog and out-of-window interruptions enabled.	-	-	X	-	-	-	-	X	-	-
		ADC_DMA_Transfer	This example describes how to configure and use the ADC peripheral to convert an external analog input and get the result using a DMA transfer, through the HAL API.	X	X	-	X	X	X	X	-	X	X
		ADC_LowPower	This example provides a short description of how to use the ADC peripheral to perform conversions with ADC low power modes: auto-wait and auto-power off.	-	-	-	-	-	X	-	-	-	X
		ADC_Regular Conversion_Polling	This example describes how to use the ADC in Polling mode to convert data through the HAL API.	X	-	-	-	-	X	-	-	-	X
		ADC_Sequencer	This example provides a short description of how to use the ADC peripheral with sequencer to convert several channels.	-	-	X	-	-	-	-	X	-	-
		ADC_TriggerMode	This example describes how to use the ADC1 and TIM2 to convert continuously data from ADC channel. Each time an external trigger is generated by TIM2 a new conversion is started by ADC.	X	-	-	-	-	X	-	-	-	X
	CAN	CAN_Networking	This example shows how to configure the CAN peripheral to send and receive CAN frames in normal mode. The sent frames are used to control LEDs by pressing Tamper push-button.	-	-	-	-	-	X	-	-	-	X

Table 2. STM32CubeF0 firmware examples (continued)

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
Examples	CEC	CEC_DataExchange	This example shows how to configure and use the CEC peripheral to receive and transmit messages.	-	-	-	-	-	X	-	-	-	X
		CEC_ListenMode	This example shows how to configure and use the CEC peripheral to receive and transmit messages between two boards while a third one (the spy device) listens but doesn't acknowledge the received messages.	-	-	-	-	-	X	-	-	-	X
		CEC_MultiAddress	This example shows how to configure and use the CEC peripheral to receive and transmit messages in the case where one device supports two distinct logical addresses at the same time.	-	-	-	-	-	X	-	-	-	X
	COMP	COMP_Analog Watchdog	This example shows how to make an analog watchdog using the COMP peripherals in window mode.	-	-	X	-	-	X	-	X	-	X
		COMP_Interrupt	This example shows how to configure the COMP peripheral to compare the external voltage applied on 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	-	X	-	-	X	-	X	-	X
	CRC	CRC_Bytes_Stream_ 7bit_CRC	Guides the user through the different configuration steps by means of the HAL API. The CRC (Cyclic Redundancy Check) calculation unit computes 7-bit long CRC codes derived from buffers of 8-bit data (bytes).	X	-	X	-	-	-	-	X	-	-
		CRC_Data_Reversing_ 16bit_CRC	Guides the user through the different configuration steps by means of the HAL API. The CRC calculation unit computes a 16-bit long CRC code derived from a buffer of 8-bit data (bytes).	X	-	X	-	-	-	-	X	-	-
		CRC_Example	Guides the user through the different configuration steps by mean of HAL API to ensure the use of the CR calculation unit to get a CRC code of a given buffer of data word (32 bits), based on a fixed generator polynomial (0x4C11DB7).	X	X	X	X	X	X	X	X	X	X
		CRC_UserDefined Polynomial	Guides the user through the different configuration steps by mean of HAL API to ensure the use of the CRC calculation unit to get a CRC code of a given buffer of data word (32 bits), based on a user defined generator polynomial. In this example, the polynomial is set manually to 0x9B.	X	-	X	-	-	X	-	X	-	X

Table 2. STM32CubeF0 firmware examples (continued)

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
Examples	Cortex	CORTEXM_ ProcessStack	This example shows how to modify Cortex®-M0 Thread mode stack.	X	X	X	X	-	-	X	X	-	-
		CORTEXM_SysTick	This example shows how to use the default configuration of SysTick with a time base equal to 1 ms in order to insert a delay between LEDs toggling.	X	X	X	X	-	X	X	X	-	X
	DAC	DAC_Signals Generation	This example provides a description of how to use the DAC peripheral to generate several signals using DMA controller.	X	-	X	-	-	X	-	X	-	X
		DAC_Simple Conversion	This example provides a short description of how to use the DAC peripheral to do a simple conversion.	X	-	X	-	-	X	-	X	-	X
	DMA	DMA_FLASHToRAM	This example provides a description of how to use a DMA to transfer a word data buffer from Flash memory to embedded SRAM through the HAL API.	X	X	X	X	-	X	X	X	-	X
	FLASH	FLASH_EraseProgram	This example describes how to configure and use the FLASH HAL API to erase and program the internal Flash memory.	X	X	X	X	X	X	-	X	X	X
		FLASH_ WriteProtection	This example provides a description of how to enable and disable the write protection for the STM32F091RCT6 Flash memory. To enable the Write Protection, uncomment the line "#define WRITE_PROTECTION_ENABLE" in main.c file.	X	X	X	X	-	X	-	X	-	X
	GPIO	GPIO_EXTI	This example shows how to configure external interrupt lines.	X	X	X	X	-	X	X	X	-	X
		GPIO_IOToggle	This example describes how to configure and use GPIOs through the HAL API.	X	X	X	X	X	X	X	X	X	X
	HAL	HAL_TimeBase_ RTC_ALARM	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	-	X	-	X	-	-
		HAL_TimeBase_ RTC_WKUP	This example describes how to customize the HAL time base using RTC wakeup instead of Systick as main source of time base. The User push-button is used to Suspend or Resume tick increment.	-	-	-	X	-	X	-	X	-	-
		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	-	X	-	X	-	-

Table 2. STM32CubeF0 firmware examples (continued)

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
Examples	I2C	I2C_EEPROM	This example describes how to perform I2C data buffer transmission/reception via DMA. The communication uses an I2C EEPROM.	X	-	-	-	-	X	-	-	-	X
		I2C_TwoBoards_ AdvComIT	This example describes how to perform I2C data buffer transmission/reception between two boards, using an interrupt	X	X	X	X	-	X	X	X	-	-
		I2C_TwoBoards_ ComDMA	This example describes how to perform I2C data buffer transmission/reception between two boards, via DMA.	X	X	X	X	-	-	X	X	-	X
		I2C_TwoBoards_ ComIT	This example describes how to perform I2C data buffer transmission/reception between two boards, using an interrupt.	X	X	X	X	-	-	X	X	-	-
		I2C_TwoBoards_ ComPolling	This example describes how to perform I2C data buffer transmission/reception between two boards in Polling mode.	X	X	X	X	-	-	X	X	-	-
		I2C_TwoBoards_ RestartAdvComIT	This example describes how to perform a multiple I2C data buffer transmission/reception with restart condition 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 with restart condition between two boards in Interrupt mode and with a restart condition.	-	-	X	-	-	-	-	-	-	-
		I2C_WakeUpFromStop	This example describes how to ensure I2C data buffer transmission and reception using Interrupt when the device is in Stop mode.	X	-	X	-	-	X	-	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	X	X	X	X	X	X	X	X	X
		IWDG_WindowMode	This example shows how to periodically update the IWDG reload counter and simulate a reload outside the window that generates an MCU IWDG reset.	X	X	X	X	X	X	X	X	X	X
	PWR	PWR_Current Consumption	This example shows how to configure the system to measure the current consumption in different low power modes.	X	X	X	X	-	-	X	X	-	-
		PWR_PVD	This example shows how to configure the programmable voltage detector using an external interrupt line. External DC supply has to be used to power V_{dd}.	X	-	X	-	-	-	-	X	-	-
		PWR_STANDBY	This example shows how to enter the system to Standby mode and wake-up from this mode using: external RESET, RTC Alarm A or WKUP pin.	-	-	-	-	-	X	-	-	-	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	-	-	-	X

Table 2. STM32CubeF0 firmware examples (continued)

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
Examples	RCC	RCC_CRS_ Synchonization_IT	This example describes how to use the RCC HAL API to configure Clock Recovery Service using IT mode.	-	-	-	-	-	X	-	-	-	X
		RCC_CRS_ Synchonization_Polling	This example describes how to use the RCC HAL API to configure Clock Recovery Service using Polling mode.	-	-	-	-	-	X	-	-	-	X
		RCC_ClockConfig	This example describes how to use the RCC HAL API to configure the system clock (SYSCLK) and modify the clock settings on run time.	X	-	X	X	-	X	X	X	-	X
	RTC	RTC_Alarm	Guides the user through the different configuration steps by mean of HAL API to ensure Alarm configuration and generation using the RTC peripheral.	X	X	X	X	X	-	X	X	X	-
		RTC_Calendar	Guides the user through the different configuration steps by mean of HAL API to configure the RTC Calendar.	-	-	-	-	-	X	-	-	-	X
		RTC_Tamper	Guides the user through the different configuration steps by mean of RTC HAL API to write/read data to/from RTC Backup registers. It also demonstrates the Tamper detection feature.	X	-	X	-	X	X	-	X	X	X
	SMBUS	SMBUS_TSENSOR	This example shows how to ensure SMBUS Data buffer transmission and reception with IT. The communication is done with a SMBUS temperature sensor.	-	-	-	-	-	X	-	-	-	X
	SPI	SPI_FullDuplex_ ComDMA	This example shows how to ensure SPI Data buffer transmission and reception with DMA. The communication is done with two boards through SPI.	X	X	X	X	X	-	X	X	X	-
		SPI_FullDuplex_ComIT	This example shows how to ensure SPI Data buffer transmission and reception using Interrupt. The communication is done with two boards through SPI.	X	X	X	X	X	-	X	X	X	-
		SPI_FullDuplex_ ComPolling	This example shows how to ensure SPI Data buffer transmission and reception using Polling. The communication is done with two boards through SPI.	X	X	X	X	X	-	X	X	X	-

Table 2. STM32CubeF0 firmware examples (continued)

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
Examples	TIM	TIM_Complementary Signals	This example shows how to configure the TIM1 peripheral to generate three complementary TIM1 signals, to insert a defined dead time value, to use the break feature and to lock the desired parameters.	X	X	X	X	-	-	X	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 (CCR3).	X	X	X	X	-	-	X	X	-	-
		TIM_InputCapture	This example shows how to use the TIM peripheral to measure the frequency of an external signal.	X	X	X	X	-	X	X	X	-	X
		TIM_PWMInput	This example shows how to use the TIM peripheral to measure the frequency and duty cycle of an external signal.	X	X	X	X	-	-	X	X	-	-
		TIM_PWMOutput	This example shows how to configure the TIM peripheral in PWM (Pulse Width Modulation) mode.	X	X	X	-	-	X	X	X	-	X
		TIM_TimeBase	This example shows how to configure the TIM peripheral to generate a time base of one second with the corresponding Interrupt request.	X	X	X	X	-	-	X	X	-	X
	TSC	TSC_Basic Acquisition_Interrupt	This example describes how to use the TSC to perform continuous acquisitions of two channels in interrupt mode.	X	-	-	-	-	X	-	-	-	X
		TSC_Basic Acquisition_Polling	This example describes how to use the TSC to perform continuous acquisitions of one channel in Polling mode.	X	-	-	-	-	X	-	-	-	X

Table 2. STM32CubeF0 firmware examples (continued)

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
Examples	UART	UART_HyperTerminal_ DMA	This example describes an UART transmission (transmit/receive) in DMA mode between a board and an Hyperterminal PC application.	-	-	-	-	X	X	-	-	X	X
		UART_OneBoards_ 8UART	Guides the user through the different configuration steps by mean of HAL API to ensure Data buffer transmission and reception At the beginning of the main program the HAL_Init() function is called to reset all the peripherals, initialize the Flash interface and the Systick.	-	-	-	-	-	X	-	-	-	-
		UART_TwoBoards_ ComDMA	This example describes an UART transmission (transmit/receive) in DMA mode between two boards.	X	X	X	X	X	-	X	X	X	-
		UART_TwoBoards_ ComIT	This example describes an UART transmission (transmit/receive) in interrupt mode between two boards.	X	X	X	X	X	-	X	X	X	-
		UART_TwoBoards_ ComPolling	This example describes an UART transmission (transmit/receive) in Polling mode between two boards.	X	X	X	X	X	-	X	X	X	-
		UART_WakeUpFrom Stop	This example shows how to configure an UART to wake up the MCU from Stop mode when a proper stimulus is received.	X	-	X	-	X	X	-	X	X	X
	WWDG	WWDG_Example	Guides the user through the different configuration steps by mean of HAL API to ensure WWDG counter update at regular period and simulate a software fault generating an MCU WWDG reset on expiry of a programmed time period.	X	X	X	X	X	X	X	X	X	X
Total number of Examples: 349				**46**	**30**	**45**	**33**	**17**	**45**	**29**	**46**	**17**	**41**

Table 2. STM32CubeF0 firmware examples (continued)

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
Examples_LL	ADC	ADC_AnalogWatchdog	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 STM32F0xx ADC LL API; peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		ADC_Continuous Conversion_TriggerSW	This example describes how to use an ADC peripheral to perform continuous ADC conversions of a channel, from a SW start. This example is based on the STM32F0xx ADC LL API; peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		ADC_Continuous Conversion_TriggerSW_ Init	This example describes how to use an ADC peripheral to perform continuous conversions of a channel, from a SW start. This example is based on the STM32F0xx ADC LL API; peripheral initialization done using LL unitary service functions for optimization purposes (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 STM32F0xx ADC LL API; peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		ADC_MultiChannel SingleConversion	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 STM32F0xx ADC LL API; peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		ADC_SingleConversion _TriggerSW	This example describes how to use an ADC peripheral to perform a single ADC conversion of a channel, at each software start. Example using programming model interrupt (for programming models polling or DMA transfer, refer to other examples). This example is based on the STM32F0xx ADC LL API; peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-

Table 2. STM32CubeF0 firmware examples (continued)

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
Examples_LL	ADC	ADC_SingleConversion _TriggerSW_DMA	This example describes how to use an ADC peripheral to perform a single ADC conversion of a channel, at each software start. Example using programming model DMA transfer (for programming models polling or interrupt, refer to other examples). This example is based on the STM32F0xx ADC LL API; peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		ADC_SingleConversion _TriggerSW_IT	This example describes how to use an ADC peripheral to perform a single ADC conversion of a channel, at each software start. Example using programming model interrupt (for programming models polling or DMA transfer, refer to other examples). This example is based on the STM32F0xx ADC LL API; peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		ADC_SingleConversion _TriggerTimer_DMA	This example describes how to use an ADC peripheral to perform a single ADC conversion of a channel, at each trigger event from timer; Conversion data are transferred by DMA into a table, indefinitely (circular mode). This example is based on the STM32F0xx ADC LL API; peripheral initialization done using LL unitary service functions for optimization purpose (performance and size).	-	-	X	-	-	-	-	-	-	-
		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. Example using programming model polling (for programming models interrupt or DMA transfer, refer to other examples). This example is based on the STM32F0xx ADC LL API; peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-

Table 2. STM32CubeF0 firmware examples (continued)

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
Examples_LL	COMP	COMP_CompareGpioVsVrefInt_IT	This example describes how to use a comparator peripheral to compare a voltage level applied on a GPIO pin versus the internal voltage reference (V_{refInt}), in Interrupt mode. This example is based on the STM32F0xx COMP LL API; peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		COMP_CompareGpioVsVrefInt_IT_Init	This example describes how to use a comparator peripheral to compare a voltage level applied on a GPIO pin versus the internal voltage reference (V_{refInt}), in Interrupt mode. This example is based on the STM32F0xx COMP LL API; peripheral initialization done using LL initialization function to demonstrate LL init usage	-	-	X	-	-	-	-	-	-	-
		COMP_CompareGpioVsVrefInt_OutputGpio	This example describes how to use a comparator peripheral to compare a voltage level applied on a GPIO pin versus the internal voltage reference (V_{refInt}), comparator output is connected to a GPIO. This example is based on the STM32F0xx COMP LL API; peripheral initialization done using LL unitary service functions for optimization purpose (performance and size).	-	-	X	-	-	-	-	-	-	-
		COMP_CompareGpioVsVrefInt_Window_IT	This example shows how to use a pair of comparator peripherals to compare a voltage level applied on a GPIO pin versus two thresholds: the internal voltage reference (V_{refInt}) and a fraction of the internal voltage reference ($V_{refInt}/2$), in Interrupt mode. This example is based on the STM32F0xx COMP LL API; peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
	CRC	CRC_CalculateAndCheck	This example shows how to configure CRC calculation unit to get a CRC code of a given data buffer, based on a fixed generator polynomial (default value 0x4C11DB7). Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		CRC_UserDefinedPolynomial	This example shows how to configure and use CRC calculation unit to get a 8-bit long CRC of a given data buffer, based on a user-defined generating polynomial. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-

Table 2. STM32CubeF0 firmware examples (continued)

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
Examples_LL	CRS	CRS_Synchronization_IT	This example describes how to configure Clock Recovery Service in IT mode through the STM32F0xx CRS LL API. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		CRS_Synchronization_Polling	This example describes how to configure Clock Recovery Service in Polling mode through the STM32F0xx CRS LL API. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
	DAC	DAC_GenerateConstantSignal_TriggerSW	This example describes how to use the DAC peripheral to generate a constant voltage signal. This example is based on the STM32F0xx DAC LL API; peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		DAC_GenerateWaveform_TriggerHW	This example describes how to use the DAC peripheral to generate a waveform voltage from digital data stream transfered by DMA. This example is based on the STM32F0xx DAC LL API; peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		DAC_GenerateWaveform_TriggerHW_Init	This example describes how to use the DAC peripheral to generate a voltage waveform from digital data stream transfered by DMA. This example is based on the STM32F0xx DAC LL API; peripheral initialization done using LL initialization function to demonstrate LL init usage.	-	-	X	-	-	-	-	-	-	-
	DMA	DMA_CopyFromFlashToMemory	This example describes how to use a DMA channel to transfer a word data buffer from Flash memory to embedded SRAM. Peripheral initialization done using LL unitary service functions for optimization purposes (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. Peripheral initialization done using LL initialization function to demonstrate LL init usage.	-	-	X	-	-	-	-	-	-	-

Table 2. STM32CubeF0 firmware examples (continued)

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
Examples_LL	EXTI	EXTI_ToggleLedOnIT	This example describes how to configure the EXTI and use GPIOs using the STM32F0xx LL API to toggle the available user LEDs on the board when User button is pressed. Peripheral initialization done using LL unitary service functions for optimization purpose (performance and size).	-	-	X	-	-	-	-	-	-	-
		EXTI_ToggleLedOnIT_Init	This example describes how to configure the EXTI and use GPIOs using the STM32F0xx LL API to toggle the available user LEDs on the board when User button is pressed. Peripheral initialization 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 available user LEDs on the board each 250 ms. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		GPIO_InfiniteLedToggling_Init	This example describes how to configure and use GPIOs through the LL API to toggle the available user LEDs on the board each 250 ms. Peripheral initialization done using LL initialization function to demonstrate LL init usage.	-	-	X	-	-	-	-	-	-	-

Table 2. STM32CubeF0 firmware examples (continued)

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
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 IT mode. Peripheral initialization done using LL unitary service functions for optimization purposes (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 IT mode. Peripheral initialization done using LL unitary service functions for optimization purpose (performance and size).	-	-	X	-	-	-	-	-	-	-
		I2C_OneBoard _Communication_IT	This example describes how to receive data bytes from an I2C Slave device using IT mode to an I2C Master device using IT mode. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		I2C_OneBoard _Communication_IT_Init	This example describes how to receive data bytes from an I2C Slave device using IT mode to an I2C Master device using IT mode. Peripheral initialization 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 IT mode. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		I2C_TwoBoards_Master Rx_SlaveTx_IT	This example describes how to receive data bytes from an I2C Slave device using IT mode to an I2C Master device using IT mode. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		I2C_TwoBoards_Master Tx_SlaveRx	This example describes how to transmit data bytes from an I2C Master device using Polling mode to an I2C Slave device using IT mode. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		I2C_TwoBoards_Master Tx_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. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-

Table 2. STM32CubeF0 firmware examples (continued)

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
Examples_LL	I2C	I2C_TwoBoards_WakeUpFromStop_IT	This example describes how to receive data byte from an I2C Slave device in Stop mode using IT mode to an I2C Master device using IT mode. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
	IWDG	IWDG _RefreshUntilUserEvent	This example describes how to configure the IWDG and insure counter update at regular period and generating an MCU IWDG reset at User Button pressed. Peripheral initialization done using LL unitary service functions for optimization purpose (performance and size).	-	-	X	-	-	-	-	-	-	-
	PWR	PWR _EnterStandbyMode	This example shows how to enter the system in Standby mode and wake-up from this mode using external RESET or wake-up interrupt.	-	-	X	-	-	-	-	-	-	-
		PWR_EnterStopMode	This example shows how to enter the system in STOP_LPREGU mode.	-	-	X	-	-	-	-	-	-	-
	RCC	RCC_OutputSystemClockOnMCO	This example describes how to configure MCO pin (PA8) to output the system clock.	-	-	X	-	-	-	-	-	-	-
		RCC_UseHSEas SystemClock	This example describes how to use the RCC LL API to start the HSE and use it as system clock.	-	-	X	-	-	-	-	-	-	-
		RCC_UseHSI_PLLas SystemClock	This example shows how to modify the PLL parameters in run time.	-	-	X	-	-	-	-	-	-	-

Table 2. STM32CubeF0 firmware examples (continued)

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
Examples_LL	RTC	RTC_Alarm	Guides the user through the different configuration steps by means of LL API to ensure Alarm configuration and generation using the RTC peripheral. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		RTC_Alarm_Init	Guides the user through the different configuration steps by means of LL API to ensure Alarm configuration and generation using the RTC peripheral. Peripheral initialization done using LL initialization function to demonstrate LL init usage.	-	-	X	-	-	-	-	-	-	-
		RTC_Calendar	Guides the user through the different configuration steps by means of HAL API to configure the RTC calendar. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		RTC_ExitStandbyWithWakeUpTimer	This example shows how to configure the RTC in order to wakeup system from Standby mode using RTC Wakeup Timer. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		RTC_Tamper	Guides the user through the different configuration steps by mean of LL API to ensure Tamper configuration using the RTC peripheral. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		RTC_TimeStamp	Guides the user through the different configuration steps by means of LL API to ensure Time Stamp configuration using the RTC peripheral. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-

Table 2. STM32CubeF0 firmware examples (continued)

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
Examples_LL	SPI	SPI_OneBoard _HalfDuplex_DMA	This example shows how to configure GPIO and SPI peripherals for transmitting bytes from an SPI Master device to an SPI Slave device by using DMA mode through the STM32F0xx SPI LL API. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		SPI_OneBoard _HalfDuplex_DMA_Init	This example shows how to configure GPIO and SPI peripherals for transmitting bytes from an SPI Master device to an SPI Slave device by using DMA mode through the STM32F0xx SPI LL API. Peripheral initialization 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 for transmitting bytes from an SPI Master device to an SPI Slave device by using IT mode through the STM32F0xx SPI LL API. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		SPI_TwoBoards _FullDuplex_DMA	This example shows how to ensure SPI Data buffer transmission and reception using DMA mode through the STM32F0xx SPI LL API. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		SPI_TwoBoards _FullDuplex_IT	This example shows how to ensure SPI Data buffer transmission and reception using Interrupt mode through the STM32F0xx SPI LL API. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-

Table 2. STM32CubeF0 firmware examples (continued)

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
Examples_LL	TIM	TIM_BreakAndDeadtime	This example shows how to configure the TIMER to: – 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 STM32F0xx TIM LL API; peripheral initialization is done using LL unitary service functions for optimization purposes (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 (CCR3). Example using the STM32F0xx TIM LL API; peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		TIM_InputCapture	This example shows how to use the TIM peripheral to measure the frequency of a periodic signal provided either by an external signal generator or by another timer instance. Example using the STM32F0xx TIM LL API; peripheral initialization done using LL unitary service functions for optimization purposes (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_OutputCompare	This example shows how to configure the TIM peripheral to generate an output waveform in different output compare modes. Example using the STM32F0xx TIM LL API, peripheral initialization done using LL unitary service functions for optimization purposes (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. Example using the STM32F0xx TIM LL API, peripheral initialization done using LL unitary service functions for optimization purposes (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. Example using the STM32F0xx TIM LL API, peripheral initialization done using LL initialization function to demonstrate LL init usage.	-	-	X	-	-	-	-	-	-	-

Table 2. STM32CubeF0 firmware examples (continued)

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
Examples_LL	TIM	TIM_TimeBase	This example shows how to configure the TIM peripheral to generate a time base. Example using the STM32F0xx TIM LL API, peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
	USART	USART_Communication_Rx_IT	This example shows how to configure GPIO and USART peripheral for receiving characters from HyperTerminal (PC) in Asynchronous mode using IT. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		USART_Communication_Rx_IT_Continuous	This example shows how to configure GPIO and USART peripheral for continuously receiving characters from HyperTerminal (PC) in Asynchronous mode using IT. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		USART_Communication_Rx_IT_Init	This example shows how to configure GPIO and USART peripheral for receiving characters from HyperTerminal (PC) in Asynchronous mode using IT. Peripheral initialization 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 cannot be completed within the allocated time, a timeout allows to exit from the sequence with a Timeout error code. This example is based on STM32F0xx USART LL API; peripheral initialization done using LL unitary service functions for optimization purpose (performance and size).	-	-	X	-	-	-	-	-	-	-
		USART_Communication_TxRx_DMA	This example shows how to configure GPIO and USART peripheral to asynchronously send characters to/from an HyperTerminal (PC) in DMA mode. Example based on STM32F0xx USART LL API. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		USART_Communication_Tx_IT	This example shows how to configure GPIO and USART peripheral to asynchronously send characters to HyperTerminal (PC) in Interrupt mode. This example is based on STM32F0xx USART LL API. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-

Table 2. STM32CubeF0 firmware examples (continued)

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
Examples_LL	USART	USART _HardwareFlowControl	This example shows how to configure GPIO and USART peripheral to asynchronously receive characters from HyperTerminal (PC) in Interrupt mode with Hardware Flow Control feature enabled. This example is based on STM32F0xx USART LL API; Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		USART _SyncCommunication _FullDuplex_DMA	This example shows how to configure GPIO, USART, DMA and SPI peripherals to transmit bytes from/to an USART peripheral to/from an SPI peripheral (in slave mode) by using DMA mode through the STM32F0xx USART LL API. Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		USART _SyncCommunication _FullDuplex_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 IT mode through the STM32F0xx USART LL API (SPI is using DMA to receive/transmit characters sent from/received by USART). Peripheral initialization done using LL unitary service functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
		USART _WakeUpFromStop	This example shows how to configure GPIO and USART peripherals allowing characters received on USART RX pin to wake-up MCU from Low power mode, using STM32F0xx USART LL API. Peripheral initialization done using LL unitary service functions for optimization purpose (performance and size).	-	-	X	-	-	-	-	-	-	-
	UTILS	UTILS _ConfigureSystemClock	This example describes how to use UTILS LL API to configure the system clock using PLL with HSI as source clock. The user application only needs to calculate PLL parameters using STM32CubeMX, and call the UTILS LL API.	-	-	X	-	-	-	-	-	-	-
		UTILS_ReadDeviceInfo	This example describes how to Read UID, Device ID and Revision ID and save them into a global information buffer.	-	-	X	-	-	-	-	-	-	-
	WWDG	WWDG _RefreshUntilUserEvent	This example describes how to configure WWDG, update counter at regular periods, and generating an MCU WWDG reset at User Button pressed. Peripheral initialization done using LL unitary services functions for optimization purposes (performance and size).	-	-	X	-	-	-	-	-	-	-
Total number of Examples_LL: 74				**0**	**0**	**74**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Table 2. STM32CubeF0 firmware examples (continued)

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
Examples_MIX	ADC	ADC_SingleConversion _TriggerSW_IT	This example describes how to use an ADC peripheral to perform a single ADC conversion of a channel, at each software start. Example using programming model interrupt (for programming models polling or DMA transfer, refer to other examples). This example is based on the STM32F0xx ADC HAL & LL API (LL API used for performance improvement).	-	-	X	-	-	-	-	-	-	-
	CRC	CRC_PolynomialUpdate	This example provides a description of how to use CRC peripheral through the STM32F0xx CRC HAL & LL API (LL API used for performance improvement). The CRC (Cyclic Redundancy Check) calculation unit computes a 8-bit long CRC code of a given buffer of 32-bit data words, based on a user-defined generating polynomial. In this example, the polynomial is first set manually to 0x9B, that is $X^8 + X^7 + X^4 + X^3 + X + 1$. Then, in a second step, polynomial value and length are updated (set to 0x1021, that is $X^{16} + X^{12} + X^5 + 1$) for new CRC calculation. These updates are performed using CRC LL API.	-	-	X	-	-	-	-	-	-	-
	DMA	DMA_FLASHToRAM	This example provides a description of how to use a DMA to transfer a word data buffer from Flash memory to the embedded SRAM through the STM32F0xx DMA HAL & 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 master and two slaves with different address size (7-bit or 10-bit) through the STM32F0xx HAL & LL API (LL API used for performance improvement), using an interrupt.	-	-	X	-	-	-	-	-	-	-
	PWR	PWR_STANDBY_RTC	This example shows how to enter the system in Standby mode and wake-up from this mode using external RESET or RTC Wake-up Timer through the STM32F0xx RTC & RCC HAL & LL API (LL API used for performance improvement).	-	-	X	-	-	-	-	-	-	-
		PWR_STOP	This example shows how to enter the system in Stop with Low power regulator mode and wake-up from this mode using external RESET or wake-up 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 ensure SPI data buffer transmission/reception in Polling mode between two boards.	-	-	X	-	-	-	-	-	-	-
		SPI_HalfDuplex _ComPollingIT	This example shows how to ensure SPI data buffer transmission/reception between two boards by using Polling (LL Driver) and interrupt mode (HAL Driver).	-	-	X	-	-	-	-	-	-	-

Table 2. STM32CubeF0 firmware examples (continued)

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
Examples_MIX	TIM	TIM_6Steps	This example shows how to configure the TIM1 peripheral to generate 6 Steps PWM signal. The STM32F0xx TIM1 peripheral offers the possibility to program in advance the configuration for the next TIM1 outputs behavior (step) and change the configuration of all the channels at the same time. This operation is possible when the COM (commutation) event is used. Example using the STM32F0xx TIM HAL & LL API (LL API used for performance improvement).	-	-	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 STM32F0xx UART HAL & LL API (LL API used 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 STM32F0xx UART HAL & LL API (LL API used for performance improvement).	-	-	X	-	-	-	-	-	-	-
Total number of Examples_MIX: 11				**0**	**0**	**11**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Applications	EEPROM	EEPROM_Emulation	This application shows how to emulate EEPROM on internal Flash memory.	-	-	-	-	-	-	-	X	-	-
	FatFs	FatFs_uSD	This application 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	-	X

Table 2. STM32CubeF0 firmware examples (continued)

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
Applications	FreeRTOS	FreeRTOS_LowPower	This application shows how to enter and exit low power mode with CMSIS RTOS API.	-	-	-	-	-	X	-	-	-	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	-	-	-	X
		FreeRTOS_Queues	This application shows how to use message queues with CMSIS RTOS API.	-	-	-	-	-	X	-	-	-	X
		FreeRTOS_ Semaphore	This application shows how to use semaphores with CMSIS RTOS API .	-	-	-	-	-	X	-	-	-	X
		FreeRTOS_Semaphore FromISR	This application shows how to use semaphore from ISR with CMSIS RTOS API.	-	-	-	-	-	X	-	-	-	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_ ThreadCreation	This application shows how to implement a thread creation using CMSIS RTOS API.	X	X	X	X	-	X	X	X	-	X
		FreeRTOS_Timers	This application shows how to use timers of CMSIS RTOS API.	-	-	-	-	-	X	-	-	-	X
	IAP	IAP_Binary_Template	This directory contains a set of sources files that build the application to be loaded into Flash memory using In-Application Programming (IAP) through USART.	-	-	-	-	-	X	-	-	-	-
		IAP_Main	This directory contains a set of sources files and pre-configured projects that describes how to build an application to be loaded into Flash memory using In-Application Programming (IAP) through USART.	-	-	-	-	-	X	-	-	-	-
	STemWin	STemWin_HelloWorld	This directory contains a set of source files that implement a simple “Hello World” example based on STemWin for STM32F0xx devices.	-	-	-	-	-	X	-	-	-	-

Table 2. STM32CubeF0 firmware examples (continued)

Level	Module name	Project Name	Description	32F072B DISCOVERY	NUCLEO -F030R8	NUCLEO -F072RB	NUCLEO -F070RB	NUCLEO -F042K6	STM32091C -EVAL	32F0308 DISCOVERY	NUCLEO -F091RC	NUCLEO -F031K6	STM32072B -EVAL
Applications	Touch Sensing	TouchSensing_ 2touchkeys	This firmware is a basic example on how to use the STMTouch driver with two touchkey sensors. The ECS and DTO are also used.	-	-	-	-	-	X	-	-	-	X
		TouchSensing_Linear	This firmware is a basic example on how to use the STMTouch driver with one linear sensor. The ECS and DTO are also used.	X	-	-	-	-	-	-	-	-	-
		TouchSensing_Linear IT	This firmware is a basic example on how to use the STMTouch driver with one linear sensor. The ECS and DTO are also used.	X	-	-	-	-	-	-	-	-	-
	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
		CustomHID_Standalone	This application shows how to use the USB device application based on the Custom HID Class.	-	-	-	-	-	-	-	-	-	X
		DFU_Standalone	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
		MSC_Standalone	This application shows how to use the USB device application based on the Mass Storage Class (MSC).	X	-	-	-	-	-	-	-	-	X
Total number of Applications: 41				**5**	**1**	**1**	**1**	**0**	**15**	**1**	**3**	**0**	**14**
Demonstra-tions	-	Demo	The provided demonstration firmware based on STM32Cube helps user to discover STM32 Cortex®-M devices that can be plugged on a STM32NUCLEO board.	X	X	X	X	-	X	X	X	-	-
		Gravitech _4Digits_Counter	This demonstration shows how to use the Gravitech 7 segments 4 digits shield with a Nucleo 32 Board.	-	-	-	-	X	-	-	-	X	-
Total number of Demonstrations: 9				**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**
Total number of projects: 504				**52**	**32**	**132**	**33**	**18**	**59**	**31**	**48**	**18**	**55**

3 Revision history

Table 3. Document revision history

Date	Revision	Changes
06-Jul-2015	1	Initial release.
15-Sep-2015	2	Added NUCLEO-F042K6 and NUCLEO-F031K6.
01-Feb-2016	3	Added \SW4STM32 toolchain. Updated *Table 2: STM32CubeF0 firmware examples*: – Updated Nucleo boards supported by ADC_DMA_Transfer project. – Added CRC_Bytes_Stream_7bit_CRC and CRC_Data_Reversing_16bit_CRC projects. – Updated Nucleo boards supported by CRC_Example, FLASH_EraseProgram, IWDG_Example, RTC_Alarm, RTC_Tamper, SPI_FullDuplex_ComDMA, SPI_FullDuplex_ComIT, SPI_FullDuplex_ComPolling, UART_HyperTerminal_DMA, UART_TwoBoards_ComDMA, UART_TwoBoards_ComIT, UART_TwoBoards_ComPolling, UART_WakeUpFromStop and WWDG_Example – Added IAP application – Added TouchSensing_Linear_IT project.
16-May-2016	4	Added Examples_LL and Examples_MIX in *Section 2: STM32CubeF0 examples*. Updated *Table 2: STM32CubeF0 firmware examples*.
15-Nov-2016	5	Updated document title and *Introduction*. Added *Section 1: Reference documents*. Updated *Section 2: STM32CubeF0 examples*. Updated *Figure 1: STM32CubeF0 firmware components*. Updated *Table 2: STM32CubeF0 firmware examples*.
28-Jul-2017	6	Updated *Introduction* and *Section 2: STM32CubeF0 examples*. Updated *Figure 1: STM32CubeF0 firmware components*. Added *Table 1: Board references*. Updated *Table 2: STM32CubeF0 firmware examples*.