AN4731

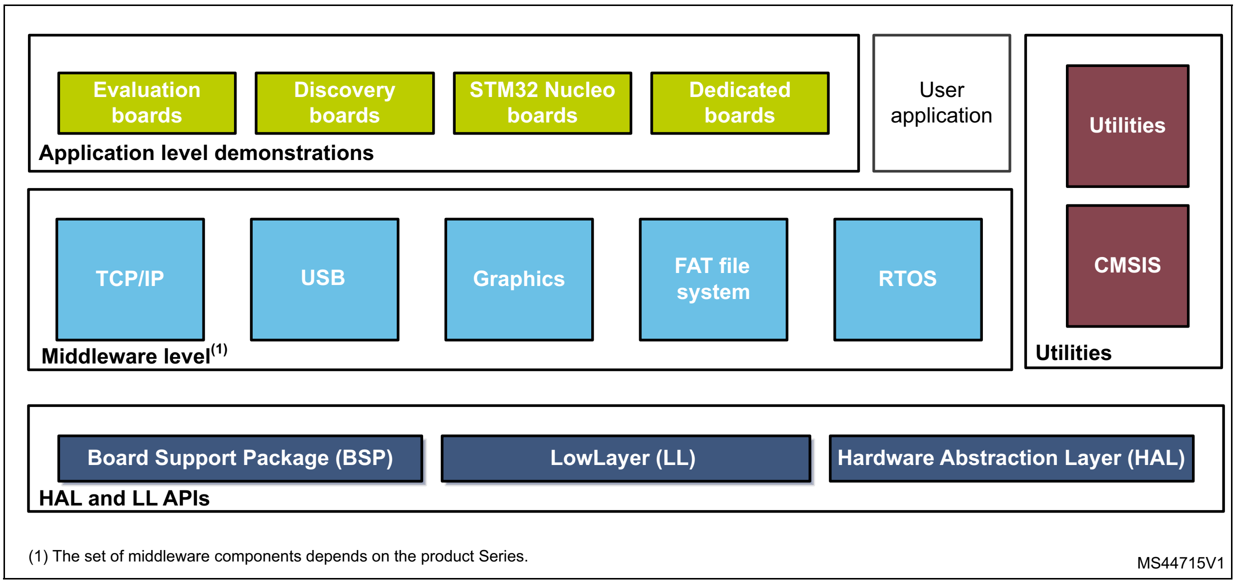
Application note

STM32Cube MCU Package examples for STM32F7 Series

Introduction

The STM32CubeF7 MCU Package comes with a rich set of examples running on STMicroelectronics boards. The examples are organized by board, and are provided with preconfigured projects for the main supported toolchains (see *Figure 1*).

Figure 1. STM32CubeF7 firmware components



Contents

1 Reference documents

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

- Latest release of STM32CubeF7 MCU Package for the STM32F7 32-bit Arm®[(a)] Cortex®-M7 microcontrollers
- *Getting started with STM32CubeF7 MCU Package for STM32F7 Series* user manual (UM1891)
- *STM32F7CubeF7 demonstration platform* user manual (UM1906)
- *Description of STM32F7xx HAL and LL drivers* user manual (UM1905)
- *STM32Cube USB Device library* user manual (UM1734)
- S*TM32Cube USB host library* user manual (UM1720)
- Developing applications on STM32Cube with FatFS user manual (UM1721)
- *Developing Applications on STM32Cube* with RTOS user manual *(*UM1722)
- *Developing Applications on STM32Cube with LwIP TCP/IP stack* user manual (UM1713)
- *STM32Cube Ethernet IAP example* user manual (UM1709)

arm

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

2 STM32CubeF7 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, for example TIM). Their complexity level ranges from the basic usage of a given peripheral (for example PWM generation using timer) to the integration of several peripherals (for example 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, for example TIM) and run exclusively on Nucleo board.
- **Examples_MIX**

  These examples use only the 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:
  - The HAL offers high-level function-oriented APIs with high portability level by hiding product/IPs complexity for end users.
  - The LL provides low-level APIs at register level with better optimization.

  The examples are organized per peripheral (one folder for each peripheral, for example 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, for example USB Host) or by product feature that require high-level firmware bricks (for example Audio). The integration of applications that use several middleware stacks is also supported.
- **Demonstrations:** the demonstrations aim to integrate and run the maximum number of peripherals and middleware stacks to showcase the product features and performance.
- **Template project:** the template project is provided to allow to quickly build a firmware application on a given board.

The examples are located under *STM32Cube_FW_STM32CubeF7_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 the 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 MCU 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, push-buttons, 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 the STM32CubeF7 MCU Package.

The total numbers of templates, templates_LL, demonstrations, examples, examples_LL, examples_MIX and applications are highlighted in gray in the table.

Table 1. STM32CubeF7 firmware examples

Level	Module Name	Project Name	Description	NUCLEO -F722ZE	32F769IDI SCOVERY	NUCLEO -F746ZG	32F746GD ISCOVERY	STM3275 6G-EVAL	32F723EDI SCOVERY	NUCLEO -F767ZI	STM32F7 69I-EVAL
Templates	-	Starter project	This projects provides a reference template that can be used to build any firmware application.	X	X	X	X	X	X	X	X
	Total number of templates: 8			**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Templates_LL	-	Starter project	This projects provides a reference template through the LL API that can be used to build any firmware application.	X	X	X	X	X	X	X	X
	Total number of templates_LL: 8			**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	X	X	X	X	-	X
	ADC	ADC_DualModeInterleaved	This example provides a short description of how to use the ADC peripheral to convert a regular channel in dual interleaved mode.	-	-	-	-	X	-	-	X
		ADC_InjectedConversion_ Interrupt	This example describes how to interrupt a continuous ADC regular channel conversion using ADC injected channels and how to get the converted value of this conversion.	-	-	-	-	X	-	-	X
		ADC_RegularConversion_ DMA	This example describes how to use the ADC3 and DMA to transfer continuously converted data from ADC3 to memory.	X	-	X	X	X	X	X	X
		ADC_RegularConversion_ Interrupt	This example describes how to use the ADC in interrupt mode to convert data through the HAL API.	X	-	X	-	X	X	X	X
		ADC_RegularConversion_ Polling	This example describes how to use the ADC in Polling mode to convert data through the HAL API.	X	-	-	-	X	X	X	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO-F722ZE	32F769IDISCOVERY	NUCLEO-F746ZG	32F746GDISCOVERY	STM32756G-EVAL	32F723EDISCOVERY	NUCLEO-F767ZI	STM32F769I-EVAL
Examples	ADC	ADC_TemperatureSensor	This example describes how to use the ADC1 and the temperature sensor to calculate the junction temperature of the device.	-	X	-	-	-	-	-	X
		ADC_TriggerMode	This example describes how to use the ADC3 and TIM2 to convert continuously data from ADC channel. Each time an external trigger is generated by TIM2 a new conversion is started by the ADC.	-	-	-	-	X	-	-	X
		ADC_TripleModeInterleaved	This example provides a short description of how to use the ADC peripheral to convert a regular channel in triple interleaved mode.	-	-	-	-	X	-	-	X
	CAN	CAN_Loopback	This example provides a description of how to set up a communication with the CAN in loopback mode.	-	-	-	-	X	X	-	X
		CAN_Networking	This example shows how to configure the CAN peripheral to send and receive CAN frames in normal mode.	-	-	-	-	X	-	-	X
	CEC	CEC_DataExchange	This example shows how to configure and use the CEC peripheral to receive and transmit messages.	-	-	-	X	X	-	-	X
	CRC	CRC_Bytes_Stream_7bit_CRC	This example 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	X	X
		CRC_Example	This example guides the user 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	X	X	X
		CRC_UserDefinedPolynomial	This example guides the user through the different configuration steps by means of the HAL API. The CRC (Cyclic Redundancy Check) calculation unit computes the 8-bit long CRC code of a given buffer of 32-bit data words, based on a user-defined generating polynomial.	X	X	-	-	X	X	X	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO -F722ZE	32F769IDI SCOVERY	NUCLEO -F746ZG	32F746GD ISCOVERY	STM3275 6G-EVAL	32F723EDI SCOVERY	NUCLEO -F767ZI	STM32F7 69I-EVAL
Examples	CRYP	CRYP_AESModes	This example provides a short description of how to use the CRYPTO peripheral to encrypt and decrypt data using AES in chaining modes (ECB, CBC, CTR).	-	-	-	-	X	-	-	X
		CRYP_AES_CCM	This example provides a short description of how to use the CRYPTO peripheral to encrypt data using AES with Combined Cipher Machine (CCM).	-	-	-	-	X	-	-	X
		CRYP_AES_DMA	This example provides a short description of how to use the CRYPTO peripheral to encrypt and decrypt data using AES-128 algorithm with ECB chaining mode.	-	-	-	-	X	-	-	X
		CRYP_AES_GCM	This example provides a description of how to use the CRYPTO peripheral to encrypt and decrypt data using AES with Galois/Counter Mode (GCM).	-	-	-	-	X	-	-	X
		CRYP_DESTDESmodes	This example provides a short description of how to use the CRYPTO peripheral to encrypt and decrypt data using DES and TDES in all mode (ECB, CBC) algorithm.	-	-	-	-	X	-	-	X
		CRYP_TDES_DMA	This example provides a short description of how to use the CRYPTO peripheral to encrypt data using TDES algorithm.	-	-	-	-	X	-	-	X
	CORTEX	CORTEXM_MPU	This example presents the MPU feature. The example purpose is to configure a memory region as privileged read only region and tries to perform read and write operation in different mode.	X	-	X	-	X	X	X	X
		CORTEXM_MPU_Config	This example presents the MPU feature. The example purpose is to configure two memories areas in Write-through and Write-back modes using the MPU attributes.	-	-	-	-	-	-	-	X
		CORTEXM_ModePrivilege	This example shows how to modify the thread mode privilege access and stack. The thread mode is entered on reset or when returning from an exception.	X	-	-	-	X	-	X	X
		CORTEXM_ProcessStack	This example shows how to modify the thread mode stack. The thread mode is entered on reset, and can be entered as a result of an exception return.	X	-	-	-	-	X	X	-
		CORTEXM_SysTick	This example shows how to use the default SysTick configuration with a 1 ms time base to toggle LEDs.	X	-	X	-	X	X	X	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO-F722ZE	32F769IDISCOVERY	NUCLEO-F746ZG	32F746GDISCOVERY	STM32756G-EVAL	32F723EDISCOVERY	NUCLEO-F767ZI	STM32F769I-EVAL
Examples	DAC	DAC_SignalsGeneration	This example provides a description of how to use the DAC peripheral to generate several signals using DMA controller.	-	-	-	X	X	X	-	X
		DAC_SimpleConversion	This example provides a short description of how to use the DAC peripheral to do a simple conversion.	-	-	-	-	X	X	-	X
	DCMI	DCMI_CaptureMode	This example provides a short description of how to use the DCMI to interface with a camera module, to capture continuously camera images in a Camera Frame Buffer in external RAM and each time a full frame camera image is captured to display it on the LCD in ARGB8888 format.	-	-	-	-	X	-	-	X
		DCMI_SnapshotMode	This example provides a short description of how to use the DCMI to interface with a camera module, to capture a single image in Camera Frame Buffer (320x240 in RGB565) and once a full frame camera is captured to display it on the LCD in ARGB8888 format.	-	-	-	-	X	-	-	X
	DFSDM	DFSDM_AudioRecord	This example shows how to use the DFSDM HAL API to perform a stereo audio recording.	-	-	-	-	-	-	-	X
	DMA	DMA_FIFOMode	This example provides a description of how to use a DMA to transfer a word data buffer from the Flash memory to embedded SRAM memory with FIFO mode enabled through the HAL API.	X	-	-	-	X	-	X	X
		DMA_FLASHToRAM	This example provides a description of how to use a DMA to transfer a word data buffer from the Flash memory to embedded SRAM memory through the HAL API.	X	X	X	X	X	X	X	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO-F722ZE	32F769IDISCOVERY	NUCLEO-F746ZG	32F746GDISCOVERY	STM32756G-EVAL	32F723EDISCOVERY	NUCLEO-F767ZI	STM32F769I-EVAL
Examples	DMA2D	DMA2D_BlendingWithAlphaInversion	This example provides a description of how to configure the DMA2D peripheral in Memory_to_Memory with blending transfer and alpha inversion mode.	-	X	-	-	-	-	-	X
		DMA2D_MemToMemWithBlending	This example provides a description of how to configure the DMA2D peripheral in Memory_to_Memory with blending transfer mode.	-	X	-	X	X	-	-	X
		DMA2D_MemToMemWithBlendingAndCLUT	This example provides a description of how to configure the DMA2D peripheral in Memory_to_Memory with blending transfer mode and indexed 256 color images (L8).	-	-	-	-	-	-	-	X
		DMA2D_MemToMemWithLCD	This example provides a description of how to configure the DMA2D peripheral in Memory_to_Memory transfer mode and display the result on LCD.	-	-	-	-	X	-	-	X
		DMA2D_MemToMemWithPFC	This example provides a description of how to configure the DMA2D peripheral for transfer in Memory_to_Memory with Pixel Format Conversion (PFC) mode.	-	-	-	X	X	-	-	-
		DMA2D_MemToMemWithPFC_A8	This example provides a description of how to configure the DMA2D peripheral in Memory_to_Memory with Pixel Format Conversion and A8 color mode, and display the result on LCD.	-	-	-	-	-	-	-	X
		DMA2D_MemToMemWithPFCandRedBlueSwap	This example provides a description of how to configure the DMA2D peripheral in Memory_to_Memory transfer mode with Pixel Format conversion and Red and Blue swap, and display the result on LCD.	-	-	-	-	-	-	-	X
		DMA2D_MemoryToMemory	This example provides a description of how to configure the DMA2D peripheral in Memory_to_Memory transfer mode.	-	-	-	-	X	-	X	X
		DMA2D_RegToMemWithLCD	This example provides a description of how to configure the DMA2D peripheral in Register_to_Memory transfer mode and display the result on LCD.	-	-	-	-	X	-	-	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO-F722ZE	32F769IDISCOVERY	NUCLEO-F746ZG	32F746GDISCOVERY	STM32756G-EVAL	32F723EDISCOVERY	NUCLEO-F767ZI	STM32F769I-EVAL
Examples	FLASH	FLASH_DualBoot	This example guides the user through the different configuration steps by means of the HAL API how to program bank1 and bank2 of the STM32F7xx internal FLASH memory mounted on STM32F769I-EVAL and swap between both of them.	-	X	-	-	-	-	-	X
		FLASH_EraseProgram	This application 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
		FLASH_JumpBootloader	This example describes how to jump to Bootloader.	-	-	-	-	-	-	-	X
		FLASH_PcropProtection	This example describes how to configure and use the FLASH HAL API to enable and disable the PCROP protection of the internal Flash memory.	-	-	-	-	-	X	-	-
		FLASH_SwapBank	This example guides the user through the different configuration steps by measn of the HAL API how to swap execution between bank1 and bank2 of the STM32F7xx internal FLASH memory mounted on STM32F769I-EVAL.	-	X	-	-	-	-	X	X
		FLASH_WriteProtection	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	X	-	X
	FMC	FMC_NOR	This example guides the user through the different configuration steps by means of the HAL API to configure the FMC controller to access the PC28F128M29EWLA NOR memory mounted on STM32F769I-EVAL evaluation board.	-	-	-	-	X	-	-	X
		FMC_NOR_PreInitConfig	This example describes how to execute a part of the code from the NOR external memory.	-	-	-	-	-	-	-	X
		FMC_PSRAM	This example describes how to configure the FMC controller to access the PSRAM memory.	-	-	-	-	-	X	-	-
		FMC_PSRAM_PreInitConfig	This example describes how to execute a part of the code from the PSRAM external memory.	-	-	-	-	-	X	-	-
		FMC_SDRAM	This example guides the user through the different configuration steps by means of the HAL API to configure the FMC controller to access the IS42S32800G SDRAM memory mounted on STM32F769I-EVAL evaluation board.	-	X	-	X	X	-	-	X
		FMC_SDRAM_DataMemory	This example describes how to configure the FMC controller to access the SDRAM memory including heap and stack.	-	-	-	X	X	-	-	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO-F722ZE	32F769IDISCOVERY	NUCLEO-F746ZG	32F746GDISCOVERY	STM32756G-EVAL	32F723EDISCOVERY	NUCLEO-F767ZI	STM32F769I-EVAL
Examples	FMC	FMC_SDRAM_LowPower	This example describes how to configure the FMC controller to access the SDRAM memory in low-power mode (SDRAM Self Refresh mode).	-	-	-	X	X	-	-	X
		FMC_SDRAM_MemRemap	This example guides the user through the different configuration steps to use the IS42S32800G SDRAM memory (mounted on STM32F769I-EVAL evaluation board) as code execution memory.	-	-	-	-	X	-	-	X
		FMC_SRAM	This example describes how to configure the FMC controller to access the SRAM memory.	-	-	-	-	X	-	-	X
		FMC_SRAM_DataMemory	This example guides the user through the different configuration steps by means of the HAL API to configure the FMC controller to access the IS61WV102416BLL-10MLI SRAM memory mounted on STM327x6G-EVAL revB evaluation board (including heap and stack).	-	-	-	-	X	-	-	-
		FMC_SRAM_PreInitConfig	This example describes how to execute a part of the code from the SRAM external memory.	-	-	-	-	-	-	-	X
	GPIO	GPIO_EXTI	This example shows how to configure external interrupt lines.	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
	HAL	HAL_TimeBase_RTC_ALARM	This example describes how to customize the HAL time base using RTC alarm instead of Systick as main source of time base. The tamper push-button (connected to EXTI Line[15:10]) is used to Suspend or Resume tick increment.	X	X	X	X	X	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 tamper push-button (connected to EXTI Line[15:10]) is used to Suspend or Resume tick increment.	X	X	X	X	X	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	X	X	X	X	X
	HASH	HASH_HMAC_SHA1MD5	This example provides a short description of how to use the HASH peripheral to hash data using HMAC SHA-1 and HMAC MD5 algorithms.	-	-	-	-	X	-	-	X
		HASH_SHA1MD5	This example provides a short description of how to use the HASH peripheral to hash data using SHA-1 and MD5 algorithms.	-	-	-	-	X	-	-	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO-F722ZE	32F769IDISCOVERY	NUCLEO-F746ZG	32F746GDISCOVERY	STM32756G-EVAL	32F723EDISCOVERY	NUCLEO-F767ZI	STM32F769I-EVAL
Examples	HASH	HASH_SHA1MD5_DMA	This example provides a short description of how to use the HASH peripheral to hash data using SHA-1 and MD5 algorithms.	-	-	-	-	X	-	-	X
		HASH_SHA224SHA256_DMA	This example provides a short description of how to use the HASH peripheral to hash data using SHA224 and SHA256 algorithms.	-	-	-	-	X	-	-	X
	I2C	I2C_EEPROM	This example describes how to perform I2C data buffer transmission/reception via DMA.	-	-	-	-	X	-	-	-
		I2C_EEPROM_FM+	This example describes how to perform I2C data buffer transmission/reception in Fast Mode Plus via DMA. The communication uses an I2C EEPROM memory.	-	-	-	-	-	-	-	X
		I2C_TwoBoards_AdvComIT	This example describes how to perform I2C data buffer transmission/reception between two boards, using an interrupt.	-	-	-	X	-	-	-	-
		I2C_TwoBoards_ComDMA	This example describes how to perform I2C data buffer transmission/reception between two boards, via DMA.	X	X	-	X	-	X	-	-
		I2C_TwoBoards_ComIT	This example describes how to perform I2C data buffer transmission/reception between two boards using an interrupt.	X	X	-	X	-	X	-	-
		I2C_TwoBoards_ComPolling	This example describes how to perform I2C data buffer transmission/reception between two boards in Polling mode.	X	X	-	X	-	X	-	-
		I2C_TwoBoards_RestartAdvComIT	This example guides the user through the different configuration steps by means of the HAL API to ensure I2C Data buffer transmission and reception using Interrupt.	-	-	-	-	-	-	X	-
		I2C_TwoBoards_RestartComIT	This example guides the user through the different configuration steps by means of the HAL API to ensure I2C Data buffer transmission and reception using Interrupt.	-	-	-	-	-	-	X	-
	I2S	I2S_DataExchangeInterrupt	This example provides a description of how to set a communication between two SPIs in I2S mode using interrupts and performing a transfer from Master to Slave.	X	-	X	-	-	-	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	-	X	-	X	X	X	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO -F722ZE	32F769IDI SCOVERY	NUCLEO -F746ZG	32F746GD ISCOVERY	STM3275 6G-EVAL	32F723EDI SCOVERY	NUCLEO -F767ZI	STM32F7 69I-EVAL
Examples	JPEG	JPEG_DecodingFromFLASH_DMA	This application demonstrates how to decode a JPEG image stored in the internal Flash memory using the JPEG HW decoder in DMA mode and display the final ARGB8888 image on KoD LCD mounted on board or a HDMI monitor connected through the DSI-HDMI bridge board MB1232.A.	-	-	-	-	-	-	-	X
		JPEG_DecodingUsingFs_DMA	This application demonstrates how to read a JPEG file from SDCard memory using FatFS, decode it using the JPEG HW decoder in DMA mode and display the final ARGB8888 image KoD LCD mounted on board or a HDMI monitor connected through the DSI-HDMI bridge board MB1232.A.	-	X	-	-	-	-	-	X
		JPEG_DecodingUsingFs_Interrupt	This application demonstrates how to read a JPEG file from SDCard memory using FatFS, decode it using the JPEG HW decoder in interrupt mode and display the final ARGB8888 image on KoD LCD mounted on board or a HDMI monitor connected through the DSI-HDMI bridge board MB1232.A.	-	-	-	-	-	-	-	X
		JPEG_DecodingUsingFs_Polling	This application demonstrates how to read a JPEG file from SDCard memory using FatFS, decode it using the JPEG HW decoder in polling mode and display the final ARGB8888 image on KoD LCD mounted on board or a HDMI monitor connected through the DSI-HDMI bridge board MB1232.A.	-	-	-	-	-	-	-	X
		JPEG_EncodingFromFLASH_DMA	This example demonstrates how to read a bmp file from SDCard memory using FatFS, encode it using the JPEG HW encoder in DMA mode and save it in SDCard.	-	-	-	-	-	-	-	X
		JPEG_EncodingUsingFs_DMA	This example demonstrates how to read bmp file from SDCard memory using FatFS, encode it using the JPEG HW encoder in DMA mode and save it in SDCard.	-	-	-	-	-	-	-	X
		JPEG_MJPEG_VideoDecoding	This example demonstrates how to use the HW JPEG decoder to decode a MJPEG video file and display it on the DSI screen.	-	-	-	-	-	-	-	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO-F722ZE	32F769IDISCOVERY	NUCLEO-F746ZG	32F746GDISCOVERY	STM32756G-EVAL	32F723EDISCOVERY	NUCLEO-F767ZI	STM32F769I-EVAL
Examples	LCD_DSI	LCD_DSI_CmdMode_DoubleBuffer	This example provides a description of how to use the embedded LCD DSI controller (using IPs LTDC and DSI Host) to drive the KoD LCD mounted on board.	-	X	-	-	-	-	-	X
		LCD_DSI_CmdMode_PartialRefresh	This example provides a description of how to use the embedded LCD DSI controller (using IPs LTDC and DSI Host) to drive the KoD LCD mounted on board.	-	X	-	-	-	-	-	X
		LCD_DSI_CmdMode_SingleBuffer	This example provides a description of how to use the embedded LCD DSI controller (using IPs LTDC and DSI Host) to drive the KoD LCD mounted on board.	-	-	-	-	-	-	-	X
		LCD_DSI_CmdMode_TearingEffect	This example provides a description of how to use the embedded LCD DSI controller (using IPs LTDC and DSI Host) to drive the KoD LCD mounted on board.	-	-	-	-	-	-	-	X
		LCD_DSI_CmdMode_TearingEffect_ExtPin	This example provides a description of how to use the embedded LCD DSI controller (using IPs LTDC and DSI Host) to drive the KoD LCD mounted on board.	-	-	-	-	-	-	-	X
		LCD_DSI_ULPM_Data	This example provides a description of how to use the embedded LCD DSI controller (using IPs LTDC and DSI Host) to drive the KoD LCD mounted on board and manage entry and exit in DSI ULPM mode on data lane only. In this mode, the DSI PHY state machine is entering a low power state on data lane and allows to save some power when the LCD does not need to display.	-	-	-	-	-	-	-	X
		LCD_DSI_ULPM_DataClock	This example provides a description of how to use the embedded LCD DSI controller (using IPs LTDC and DSI Host) to drive the KoD LCD mounted on board and manage entry and exit in DSI ULPM mode on data lane and clock lane.	-	-	-	-	-	-	-	X
		LCD_DSI_VideoMode_DoubleBuffering	This example provides a description of how to use the embedded LCD DSI controller (using IPs LTDC and DSI Host) to drive the KoD LCD mounted on board or a HDMI monitor connected through the DSI-HDMI bridge board MB1232.A.	-	X	-	-	-	-	-	X
		LCD_DSI_VideoMode_SingleBuffer	This example provides a description of how to use the embedded LCD DSI controller (using IPs LTDC and DSI Host) to drive the KoD LCD mounted on board or a HDMI monitor connected through the DSI-HDMI bridge board MB1232.A.	-	X	-	-	-	-	-	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO-F722ZE	32F769IDISCOVERY	NUCLEO-F746ZG	32F746GDISCOVERY	STM32756G-EVAL	32F723EDISCOVERY	NUCLEO-F767ZI	STM32F769I-EVAL
Examples	LPTIM	LPTIM_PWMExternalClock	This example describes how to configure and use LPTIM to generate a PWM at the lowest power consumption, using an external counter clock, through the HAL LPTIM API.	-	-	-	-	-	-	X	X
		LPTIM_PWM_LSE	This example describes how to configure and use LPTIM to generate a PWM in low-power mode using the LSE as a counter clock, through the HAL LPTIM API.	-	-	-	-	-	-	X	X
		LPTIM_PulseCounter	This example describes how to configure and use LPTIM to count pulses through the LPTIM HAL API.	-	-	-	-	-	-	X	X
		LPTIM_Timeout	This example describes how to implement a low power timeout to wake up the system using the LPTIMER, through the HAL LPTIM API.	X	-	-	-	-	-	X	X
	LTDC	LTDC_ColorKeying	This example describes how to enable and use the LTDC color keying functionality.	-	-	-	-	X	-	-	-
		LTDC_Display_1Layer	This example provides a description of how to configure LTDC peripheral to display BMP image of size 480x272 and format RGB888 (24 bits/pixel) on LCD using only one layer.	-	-	-	X	X	-	-	-
		LTDC_Display_2Layers	This example describes how to configure the LTDC peripheral to display two layers at the same time.	-	-	-	X	X	-	-	-

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO -F722ZE	32F769IDI SCOVERY	NUCLEO -F746ZG	32F746GD ISCOVERY	STM3275 6G-EVAL	32F723EDI SCOVERY	NUCLEO -F767ZI	STM32F7 69I-EVAL
Examples	PWR	PWR_CurrentConsumption	This example shows how to configure the STM32F7xx system to measure different Low-power mode current consumption. The Low-power modes are: - Sleep mode - Stop mode with RTC - Standby mode without RTC and BKPSRAM - Standby mode with RTC - Standby mode with RTC and BKPSRAM. To run this example, the user has to follow the following steps: 1. Select the Low-power modes to be measured by uncommenting the corresponding line inside the stm32f7xx_lp_modes.h file.	X	X	X	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	-	-	-
		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 wakeup timer.		-	-	-	-	-	-	X
		PWR_STOP	This example shows how to enter the system to Stop mode and wake up from this mode using RTC wakeup timer event connected to EXTI_Line22 or by pressing the tamper push-button connected to EXTI15_10.	-	-	-	-	X	-	-	-
		PWR_STOP_RTC	This example shows how to enter Stop mode and wake up from this mode by using the RTC wakeup timer event connected to an interrupt.	-	-	-	-	-	-	-	X
	QSPI	QSPI_ExecuteInPlace	This example describes how to execute a part of the code from the QSPI memory. To do this, a section is created where the function is stored.	-	-	-	X	X	-	-	X
		QSPI_MemoryMapped	This example describes how to erase part of the QSPI memory, write data in DMA mode and access to QSPI memory in memory-mapped mode to check the data in a forever loop.	-	-	-	-	X	-	-	X
		QSPI_PreInitConfig	This example describes how to execute a part of the code from the QSPI memory. To do this, a section is created where the function is stored.	-	-	-	X	X	-	-	X
		QSPI_ReadWrite	This example shows how to erase, to write to and to read from the external MX25R6435F Macronix Flash memory using QSPI communication.	-	-	-	-	-	X	-	-

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO -F722ZE	32F769IDI SCOVERY	NUCLEO -F746ZG	32F746GD ISCOVERY	STM3275 6G-EVAL	32F723EDI SCOVERY	NUCLEO -F767ZI	STM32F7 69I-EVAL
Examples	QSPI	QSPI_ReadWriteDual_DMA	This example describes how to use the QSPI interface in dual mode. It erases part of the QSPI memory, writes data in DMA mode, reads data in DMA mode and compares the result in a forever loop.	-	-	-	-	-	-	-	X
		QSPI_ReadWrite_DMA	This example describes how to erase part of the QSPI memory, write data in DMA mode, read data in DMA mode and compare the result in a forever loop.	-	-	-	-	X	-	-	X
		QSPI_ReadWrite_IT	This example describes how to erase part of the QSPI memory, write data in IT mode, read data in IT mode and compare the result in a forever loop.	-	-	-	X	X	-	-	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	X	X	X
	RNG	RNG_MultiRNG	This example guides the user through the different configuration steps by means of the HAL API to ensure RNG random 32-bit number generation.	X	X	-	-	X	X	X	X
	RTC	RTC_Alarm	This example guides the user through the different configuration steps by means of the HAL API to configure and generate an RTC alarm.	-	-	-	-	X	X	-	X
		RTC_Calendar	This example guides the user through the different configuration steps by means of the HAL API to ensure Calendar configuration using the RTC peripheral.	-	-	X	-	X	-	X	X
		RTC_Chronometer	This example illustrates how to simulate a precise chronometer with sub second feature.	-	-	-	-	-	-	-	X
		RTC_InternalTimeStamp	This example guides the user through the different configuration steps by means of the RTC HAL API to demonstrate the internal timestamp feature.	-	-	-	-	-	-	-	X
		RTC_Tamper	This example guides the user 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	-	X	-	-	X
		RTC_TimeStamp	This example guides the user through the different configuration steps by means of the RTC HAL API to demonstrate the timestamp feature.	X	-	-	-	X	-	-	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO-F722ZE	32F769IDISCOVERY	NUCLEO-F746ZG	32F746GDISCOVERY	STM32756G-EVAL	32F723EDISCOVERY	NUCLEO-F767ZI	STM32F769I-EVAL
Examples	SAI	SAI_Audio	This example provides basic implementation of audio features using BSP_AUDIO.	-	-	-	-	X	-	-	-
		SAI_AudioPlay	This example shows how to use the SAI HAL API to play an audio file using the DMA circular mode and how to handle the buffer update.	-	X	-	-	X	-	-	X
	SPDIFRX	SPDIFRX_Loopback	This example shows how to use the SPDIFRX HAL APIs to receive a data buffer in polling mode.	-	X	-	-	-	-	-	-
	SPI	SPI_FullDuplex_AdvComIT	This example guides the user through the different configuration steps by means of the HAL API to ensure the SPI data buffer transmission and reception using Interrupt, in an advance communication mode. The master board is always sending a command to the slave before any transmission and the slave board is sending an acknowledge before going further.	-	-	-	-	-	-	X	-
		SPI_FullDuplex_AdvComPolling	This example guides the user through the different configuration steps by means of the HAL API to ensure the SPI data buffer transmission and reception using Interrupt, in an advance communication mode. The master board is always sending a command to the slave before any transmission and the slave board is sending an acknowledge before going further.	-	-	-	-	-	-	X	-
		SPI_FullDuplex_ComDMA	This example shows how to perform SPI data buffer transmission/reception between two boards via DMA.	X	X	-	X	-	X	X	-
		SPI_FullDuplex_ComIT	This example shows how to ensure SPI data buffer transmission/reception between two boards by using an interrupt.	-	-	-	X	-	X	X	-
		SPI_FullDuplex_ComPolling	This example shows how to ensure SPI data buffer transmission/reception in Polling mode between two boards.	X	-	-	X	-	X	X	-
	TIM	TIM_6Steps	This example shows how to configure the TIM1 peripheral to generate 6 Steps.	-	-	-	-	X	-	-	X
		TIM_7PWMOutput	This example shows how to configure the TIM1 peripheral to generate 7 PWM signals with 4 different duty cycles (50%, 37.5%, 25% and 12.5%).	-	-	-	-	X	-	-	X
		TIM_CascadeSynchro	This example shows how to synchronize TIM2 and Timers (TIM3 and TIM4) in cascade mode.	-	-	-	-	X	-	-	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO-F722ZE	32F769IDISCOVERY	NUCLEO-F746ZG	32F746GDISCOVERY	STM32756G-EVAL	32F723EDISCOVERY	NUCLEO-F767ZI	STM32F769I-EVAL
Examples	TIM	TIM_ComplementarySignals	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
		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).	-	-	X	-	X	-	X	X
		TIM_DMABurst	This example shows how to update the TIMER channel1 period and the duty cycle using the TIMER DMA burst feature.	-	-	-	-	X	-	-	X
		TIM_ExtTriggerSynchro	This example shows how to synchronize TIM peripherals in cascade mode with an external trigger.	-	-	-	-	X	-	-	X
		TIM_InputCapture	This example shows how to use the TIM peripheral to measure the frequency of an external signal.	-	-	X	-	X	-	-	X
		TIM_OCActive	This example shows how to configure the TIM peripheral in Output Compare Active mode (when the counter matches the capture/compare register, the concerned output pin is set to its active state).	-	-	X	-	X	-	-	X
		TIM_OCInactive	This example shows how to configure the TIM peripheral in Output Compare Inactive mode with the corresponding Interrupt requests for each channel.	-	-	-	-	X	-	-	X
		TIM_OCToggle	This example shows how to configure the TIM peripheral to generate four different signals with four different frequencies.	-	-	X	-	X	-	-	X
		TIM_OnePulse	This example shows how to use the TIM peripheral to generate a One pulse mode after a Rising edge of an external signal is received in Timer Input pin.	-	-	X	-	X	-	-	X
		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
		TIM_PWMOutput	This example shows how to configure the TIM peripheral in PWM (Pulse Width Modulation) mode.	-	-	X	-	X	-	-	X
		TIM_ParallelSynchro	This example shows how to synchronize TIM2 and timers (TIM3 and TIM4) in parallel mode.	-	-	-	-	X	-	-	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO -F722ZE	32F769IDI SCOVERY	NUCLEO -F746ZG	32F746GD ISCOVERY	STM3275 6G-EVAL	32F723EDI SCOVERY	NUCLEO -F767ZI	STM32F7 69I-EVAL
Examples	TIM	TIM_PrescalerSelection	This example shows how to configure the TIM peripheral in PWM (Pulse Width Modulation) mode with clock prescaler selection feature activated using __HAL_RCC_TIMCLKPRESCALER() which allow to double the output frequency.	-	-	-	-	X	-	-	X
		TIM_Synchronization	This example shows how to synchronize TIM1 and Timers (TIM3 and TIM4) in parallel mode.	-	-	-	-	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
	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
		UART_HyperTerminal_IT	This example describes an UART transmission (transmit/receive) between a board and an Hyperterminal PC application by using an interrupt.	-	-	-	-	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 on the HyperTerminal.	-	-	X	-	X	-	X	X
		UART_TwoBoards_ComDMA	This example describes a UART transmission (transmit/receive) in DMA mode between two boards.	X	X	-	X	-	X	-	-
		UART_TwoBoards_ComIT	This example describes a UART transmission (transmit/receive) in interrupt mode between two boards.	-	-	-	X	-	X	-	-
		UART_TwoBoards_ComPolling	This example describes a UART transmission (transmit/receive) in polling mode between two boards.	-	-	-	X	-	X	-	-
	WWDG	WWDG_Example	This example guides the user 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	-	X	-	X	X	X	X
	Total number of examples: 421			**33**	**30**	**28**	**32**	**93**	**41**	**41**	**123**

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO -F722ZE	32F769IDI SCOVERY	NUCLEO -F746ZG	32F746GD ISCOVERY	STM3275 6G-EVAL	32F723EDI SCOVERY	NUCLEO -F767ZI	STM32F7 69I-EVAL
Examples_LL	ADC	ADC_AnalogWatchdog	This example describes how to use a ADC peripheral with the 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 STM32F7xx ADC LL API. The peripheral initialization is done using LL unitary service functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
		ADC_ContinuousConversion_TriggerSW	This example describes how to use a ADC peripheral to perform continuous ADC conversions of a channel, from a SW start. This example is based on the STM32F7xx ADC LL API. The peripheral initialization is done using LL unitary service functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
		ADC_ContinuousConversion_TriggerSW_Init	This example describes how to use a ADC peripheral to perform continuous ADC conversions of a channel, from a SW start. This example is based on the STM32F7xx ADC LL API. The peripheral initialization is done using LL unitary service functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
		ADC_GroupsRegularInjected	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 STM32F7xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
		ADC_MultiChannelSingleConversion	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 STM32F7xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO -F722ZE	32F769IDI SCOVERY	NUCLEO -F746ZG	32F746GD ISCOVERY	STM3275 6G-EVAL	32F723EDI SCOVERY	NUCLEO -F767ZI	STM32F7 69I-EVAL
Examples_LL	ADC	ADC_MultimodeDualInterleaved	This example describes how to use several ADC peripherals in multimode, mode interleaved. This example is based on the STM32F7xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
		ADC_SingleConversion_TriggerSW	This example describes how to use a ADC peripheral to perform a single ADC conversion of a channel, at each software start. The example is using the programming model: polling (for programming models interrupt or DMA transfer, refer to other examples). This example is based on the STM32F7xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
		ADC_SingleConversion_TriggerSW_DMA	This example describes how to use a ADC peripheral to perform a single ADC conversion of a channel, at each software start. The example is using the programming model: DMA transfer (for programming models polling or interrupt, refer to other examples). This example is based on the STM32F7xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
		ADC_SingleConversion_TriggerSW_IT	This example describes how to use a ADC peripheral to perform a single ADC conversion of a channel, at each software start. The example is using the programming model: interrupt (for programming models polling or DMA transfer, refer to other examples). This example is based on the STM32F7xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO -F722ZE	32F769IDI SCOVERY	NUCLEO -F746ZG	32F746GD ISCOVERY	STM3275 6G-EVAL	32F723EDI SCOVERY	NUCLEO -F767ZI	STM32F7 69I-EVAL
Examples_LL	ADC	ADC_SingleConversion_TriggerTimer_DMA	This example describes how to use a ADC peripheral to perform a single ADC conversion of a channel, at each trigger event from timer. The conversion data are transferred by DMA into a table, indefinitely (circular mode). This example is based on the STM32F7xx ADC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
		ADC_TemperatureSensor	This example describes how to use a ADC peripheral to perform a single ADC conversion of the internal temperature sensor and to calculate the temperature in Celsius degrees. The example is using the programming model: polling (for programming models interrupt or DMA transfer, refer to other examples);. This example is based on the STM32F7xx 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_CalculateAndCheck	This example shows how to configure the CRC calculation unit to get a 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	-
		CRC_UserDefinedPolynomial	This example shows how to configure and use the CRC calculation unit to get a 8-bit long CRC of a given data buffer, based on a user-defined generating polynomial. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO -F722ZE	32F769IDI SCOVERY	NUCLEO -F746ZG	32F746GD ISCOVERY	STM3275 6G-EVAL	32F723EDI SCOVERY	NUCLEO -F767ZI	STM32F7 69I-EVAL
Examples_LL	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 STM32F7xx DAC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (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 STM32F7xx DAC LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
		DAC_GenerateWaveform_TriggerHW_Init	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 STM32F7xx DAC LL API. The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	-	-	-	-	X	-
	DMA	DMA_CopyFromFlashToMemory	This example describes how to use a DMA to transfer a word data buffer from the Flash memory to embedded SRAM. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
		DMA_CopyFromFlashToMemory_Init	This example describes how to use a DMA to transfer a word data buffer from the Flash memory to embedded SRAM. The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	-	-	-	-	X	-
	DMA2D	DMA2D_MemoryToMemory	This example describes how to configure the DMA2D peripheral in Memory-to-Memory transfer mode. The example is based on the STM32F7xx DMA2D LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO -F722ZE	32F769IDI SCOVERY	NUCLEO -F746ZG	32F746GD ISCOVERY	STM3275 6G-EVAL	32F723EDI SCOVERY	NUCLEO -F767ZI	STM32F7 69I-EVAL
Examples_LL	EXTI	EXTI_ToggleLedOnIT	This example describes how to configure the EXTI and use GPIOs using the STM32F7xx LL API to toggles the available users LEDs on the board when The user button is pressed. 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 using the STM32F7xx LL API to toggles the available users LEDs on the board when the user button is pressed. The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	-	-	-	-	X	-
	GPIO	GPIO_InfiniteLedToggling	This example describes how to configure and use GPIOs through the LL API to toggles the available users LEDs on the board each 250 ms. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
		GPIO_InfiniteLedToggling_I nit	This example describes how to configure and use GPIOs through the LL API to toggles the available users LEDs on the board each 250 ms. The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	-	-	-	-	X	-
	I2C	I2C_OneBoard_AdvComm unication_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_Communic ation_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_Communic ation_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	-

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO-F722ZE	32F769IDISCOVERY	NUCLEO-F746ZG	32F746GDISCOVERY	STM32756G-EVAL	32F723EDISCOVERY	NUCLEO-F767ZI	STM32F769I-EVAL
Examples_LL	I2C	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	-
		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	-

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO-F722ZE	32F769IDISCOVERY	NUCLEO-F746ZG	32F746GDISCOVERY	STM32756G-EVAL	32F723EDISCOVERY	NUCLEO-F767ZI	STM32F769I-EVAL
Examples_LL	LPTIM	LPTIM_PulseCounter	This example describes how to use the LPTIM in counter mode to generate a PWM output signal and update PWM duty cycle, based on a trigger provided by an external function generator. This example is based on the STM32F7xx LPTIM LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
		LPTIM_PulseCounter_Init	This example describes how to use the LPTIM in counter mode to generate a PWM output signal and update PWM duty cycle, based on a trigger provided by an external function generator. This example is based on the STM32F7xx LPTIM LL API. The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	-	-	-	-	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_MAINREGU mode.	-	-	-	-	-	-	X	-
	RCC	RCC_OutputSystemClockOnMCO	This example describes how to configure MCO pins (PA8 and PC9) to output the system clock.	-	-	-	-	-	-	X	-
		RCC_UseHSEasSystemClock	This example describes how to use the RCC LL API how 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	-
	RNG	RNG_GenerateRandomNumbers	This example shows how to configure RNG peripheral to allow generation of 32-bit long Random Numbers. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
		RNG_GenerateRandomNumbers_IT	This example shows how to configure the RNG peripheral to allow generation of 32-bit long Random Numbers, using interrupts. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO -F722ZE	32F769IDI SCOVERY	NUCLEO -F746ZG	32F746GD ISCOVERY	STM3275 6G-EVAL	32F723EDI SCOVERY	NUCLEO -F767ZI	STM32F7 69I-EVAL
Examples_LL	RTC	RTC_Alarm	This example guides the user through the different configuration steps by means 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 the user through the different configuration steps by means 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 the user through the different configuration steps by means 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_ExitStandbyWithWakeUpTimer	This example shows how to configure the RTC in order to wake up from Standby mode using RTC Wakeup Timer. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
		RTC_Tamper	This example guides the user 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	-
		RTC_TimeStamp	This example guides the user through the different configuration steps by means of LL API to ensure Time Stamp configuration using the RTC peripheral. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO -F722ZE	32F769IDI SCOVERY	NUCLEO -F746ZG	32F746GD ISCOVERY	STM3275 6G-EVAL	32F723EDI SCOVERY	NUCLEO -F767ZI	STM32F7 69I-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 the DMA mode through the STM32F7xx SPI LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
		SPI_OneBoard_HalfDuplex _DMA_Init	This example shows how to configure GPIO and SPI peripherals for transmitting bytes from an SPI Master device to an SPI Slave device by using the DMA mode through the STM32F7xx SPI LL API. The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	-	-	-	-	X	-
		SPI_OneBoard_HalfDuplex _IT	This example shows how to configure GPIO and SPI peripherals for transmitting bytes from an SPI Master device to an SPI Slave device by using IT mode through the STM32F7xx SPI LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
		SPI_TwoBoards_FullDuple x_DMA	This example shows how to ensure the SPI data buffer transmission and reception in DMA mode. The example is based on the STM32F7xx SPI LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
		SPI_TwoBoards_FullDuple x_IT	This example shows how to ensure the SPI Data buffer transmission and reception in Interrupt mode. The example is based on the STM32F7xx SPI LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO-F722ZE	32F769IDISCOVERY	NUCLEO-F746ZG	32F746GDISCOVERY	STM32756G-EVAL	32F723EDISCOVERY	NUCLEO-F767ZI	STM32F769I-EVAL
Examples_LL	TIM	TIM_BreakAndDeadtime	This example shows how to configure the Timer to perform the following: to generate three center-aligned PWM and complementary PWM signals, to insert a defined dead time value, to use the break feature, to lock the desired parameters. This example is based on the STM32F7xx 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 the DMA with TIMER update request to transfer Data from the memory to the TIMER Capture Compare Register 3 (TIMx_CCR3). The example is using the STM32F7xx TIM LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
		TIM_InputCapture	This example shows how to use the TIM peripheral to measure the frequency of a periodic signal provided either by an external signal generator or by another timer instance. The example is using the STM32F7xx 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 tPULSE and after a delay of tDELAY. This example is based on the STM32F7xx TIM LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
		TIM_OutputCompare	This example shows how to configure the TIM peripheral to generate an output waveform in different output compare modes. The example is using the STM32F7xx 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 the PWM duty cycle. The example is using the STM32F7xx TIM LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO -F722ZE	32F769IDI SCOVERY	NUCLEO -F746ZG	32F746GD ISCOVERY	STM3275 6G-EVAL	32F723EDI SCOVERY	NUCLEO -F767ZI	STM32F7 69I-EVAL
Examples_LL	TIM	TIM_PWMOutput_Init	This example describes how to use a timer peripheral to generate a PWM output signal and update the PWM duty cycle. The example is using the STM32F7xx TIM LL API. The peripheral initialization is done using LL initialization function to demonstrate LL init usage.	-	-	-	-	-	-	X	-
		TIM_TimeBase	This example shows how to configure the TIM peripheral to generate a time base. The example is using the STM32F7xx TIM LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
	USART	USART_Communication_Rx_IT	This example shows how to configure the GPIO and USART peripherals for receiving characters from 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 the GPIO and USART peripherals for continuously receiving characters from 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 the GPIO and USART peripherals for receiving characters from 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 the 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 STM32F7xx USART LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO -F722ZE	32F769IDI SCOVERY	NUCLEO -F746ZG	32F746GD ISCOVERY	STM3275 6G-EVAL	32F723EDI SCOVERY	NUCLEO -F767ZI	STM32F7 69I-EVAL
Examples_LL	USART	USART_Communication_TxRx_DMA	This example shows how to configure the GPIO and USART peripherals to send characters asynchronously to/from an HyperTerminal (PC) in DMA mode. This example is based on STM32F7xx 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 the GPIO and USART peripherals to send characters asynchronously to HyperTerminal (PC) in Interrupt mode. This example is based on STM32F7xx USART LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
		USART_HardwareFlowControl	This example shows how to configure the GPIO and USART peripherals to receive characters asynchronously from HyperTerminal (PC) in Interrupt mode with Hardware Flow Control feature enabled. This example is based on STM32F7xx USART LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
		USART_SyncCommunication_FullDuplex_DMA	This example shows how to configure the 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 STM32F7xx USART LL API. The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-
		USART_SyncCommunication_FullDuplex_IT	This example shows how to configure the GPIO, USART, DMA and SPI peripherals for transmitting bytes from/to an USART peripheral to/from an SPI peripheral (in slave mode) by using Interrupt mode through the STM32F7xx USART LL API (SPI is using DMA for receving/transmitting characters sent from/received by USART). The peripheral initialization is done using LL unitary services functions for optimization purpose (performance and size).	-	-	-	-	-	-	X	-

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO-F722ZE	32F769IDISCOVERY	NUCLEO-F746ZG	32F746GDISCOVERY	STM32756G-EVAL	32F723EDISCOVERY	NUCLEO-F767ZI	STM32F769I-EVAL
Examples_LL	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 just needs to calculate PLL parameters using STM32CubeMX and to 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 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: 73			**0**	**0**	**0**	**0**	**0**	**0**	**73**	**0**
Examples_MIX	ADC	ADC_SingleConversion_TriggerSW_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 STM32F7xx ADC HAL and LL API (LL API usage for performance improvement).	-	-	-	-	-	-	X	-
	CRC	CRC_PolynomialUpdate	This example provides a description of how to use CRC peripheral through the STM32F7xx CRC HAL & LL API (LL API used for performance improvement).	-	-	-	-	-	-	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 STM32F7xx DMA HAL and LL API (LL API used for performance improvement).	-	-	-	-	-	-	X	-

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO -F722ZE	32F769IDI SCOVERY	NUCLEO -F746ZG	32F746GD ISCOVERY	STM3275 6G-EVAL	32F723EDI SCOVERY	NUCLEO -F767ZI	STM32F7 69I-EVAL
Examples_MIX	DMA2D	DMA2D_MemToMemWithLCD	This example provides a description of how to configure the DMA2D peripheral in Memory_to_Memory transfer mode and display the result on LCD, in resorting to DMA2D LL APIs for performance improvement.	-	-	-	-	-	-	X	-
		DMA2D_MemToMemWithRBSwap	This example provides a description of how to configure DMA2D peripheral in Memory to Memory transfer mode with Pixel Format Conversion and images blending then display the result on LCD, in resorting to DMA2D LL APIs for performance improvement.	-	-	-	-	-	-	X	-
	I2C	I2C_OneBoard_ComSlave7_10bits_IT	This example describes how to perform I2C data buffer transmission/reception between master and 2 slaves with different Address size (7-bit or 10-bit) through the STM32F7xx 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 STM32F7xx 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	-

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO-F722ZE	32F769IDISCOVERY	NUCLEO-F746ZG	32F746GDISCOVERY	STM32756G-EVAL	32F723EDISCOVERY	NUCLEO-F767ZI	STM32F769I-EVAL
Examples_MIX	SPI	SPI_FullDuplex_ComPolling	This example shows how to ensure SPI data buffer transmission/reception in Polling mode between two boards.	-	-	-	-	-	-	X	-
		SPI_HalfDuplex_ComPollingIT	This example shows how to ensure SPI data buffer transmission/reception between two boards by using Polling (LL Driver) an interrupt mode (HAL Driver).	-	-	-	-	-	-	X	-
	TIM	TIM_6Steps	This example shows how to configure the TIM1 peripheral to generate 6 Steps PWM signal; The STM32F7xx TIM1 peripheral offers the possibility to program in advance the configuration for the next TIM1 outputs behaviour (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 STM32F7xx TIM HAL & LL API (LL API used 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 STM32F7xx 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 STM32F7xx UART HAL and LL API (LL API usage for performance improvement).	-	-	-	-	-	-	X	-
	Total number of examples_MIX: 14			**0**	**0**	**0**	**0**	**0**	**0**	**14**	**0**

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO -F722ZE	32F769IDI SCOVERY	NUCLEO -F746ZG	32F746GD ISCOVERY	STM3275 6G-EVAL	32F723EDI SCOVERY	NUCLEO -F767ZI	STM32F7 69I-EVAL
Applications	Audio	Audio_playback_and_record	This application shows how to use the different functionalities of the audio device and ST MEMS microphones (MP34DT01), use the touch screen to enter into playback or record menu 1) Explorer Audio File menu.	-	X	-	X	X	X	-	X
	Camera	Camera_To_USBDisk	This application provides a short description of how to use the DCMI to interface with the camera module and display in continuous mode the picture on LCD and to save a picture in USB device.	-	-	-	-	X	-	-	X
	Display	LCD_AnimatedPictureFromSDCard	This application describes how to display an animated picture on LCD saved under microSD.	-	-	-	-	-	-	-	X
		LCD_DSI_ImagesSlider	This application aims to show the outstanding capability of Display Serial Interface (DSI) peripheral to display images with high resolution (800x480). With a simple movement of finger, the content of GRAM is directly updated and displayed on DSI LCD.	-	-	-	-	-	-	-	X
		LCD_PicturesFromSDCard	This application describes how to display pictures on LCD saved under SD card.	-	X	-	-	-	-	-	X
		LCD_PicturesFromUSB	This application describes how to display pictures on LCD saved under USB disk.	-	-	-	-	-	X	-	-
		LTDC_AnimatedPictureFromSDCard	This application describes how to display an animated picture on LCD saved under microSD.	-	-	-	-	X	-	-	-
		LTDC_Paint	This application describes how to configure LCD touch screen and attributes an action related to configured touch zone and how to save BMP picture in USB Disk.	-	-	-	-	X	-	-	-
		LTDC_PicturesFromSDCard	This application describes how to display pictures on LCD saved under SD card.	-	-	-	X	X	-	-	-
	EEPROM	EEPROM_Emulation	This application describes the software solution for substituting standalone EEPROM by emulating the EEPROM mechanism using the on-chip Flash memory of the STM32F77x devices.	X	X	X	X	X	X	X	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name		Description	NUCLEO -F722ZE	32F769IDI SCOVERY	NUCLEO -F746ZG	32F746GD ISCOVERY	STM3275 6G-EVAL	32F723EDI SCOVERY	NUCLEO -F767ZI	STM32F7 69I-EVAL
Applications	ExtMem_CodeExecution	ExtMem_ Application	FreeRTOS	How to implement thread creation using CMSIS RTOS API with execution from external memory.	-	-	-	-	X	X	-	-
			LedToggling	This application provides sample LED toogling program with execution from external memory.	-	-	-	-	X	X	-	-
		ExtMem_Boot		This directory contains a set of sources files and pre-configured projects that describes how to build an application for execution from external memory using the ExtMem_Boot firmware.	-	-	-	-	X	X	-	-
	FPU	FPU_Fractal		This application explains how to use, and demonstrates the benefits brought by, the STM32F7 floating-point units (FPU). The Cortex-M7 FPU is an implementation of the Arm FPv5-SP double-precision FPU.	-	-	-	-	X	-	-	X
	FatFS	FatFs_MultiDrives		This application provides a description on how to use STM32Cube firmware with FatFS middleware component as a generic FAT file system module, in order to develop an application exploiting FatFS offered features with multidrives (RAMDisk, uSD) configuration.	-	-	-	-	X	-	-	X
		FatFs_RAMDisk		This example provides a description on how to use STM32Cube firmware with FatFS middleware component as a generic FAT file system module, in order to develop an application exploiting FatFS offered features with RAM disk (SRAM) drive configuration.	-	-	-	-	X	-	-	X
		FatFs_RAMDisk_RTOS		This application provides a description on how to use STM32Cube firmware with FatFS middleware component as a generic FAT file system module, in order to develop an application exploiting FatFS offered features with RAM disk (SDRAM) drive in RTOS mode configuration.	-	-	-	-	X	-	-	X
		FatFs_USBDisk		This application provides a description on how to use STM32Cube firmware with FatFS middleware component as a generic FAT file system module and STM32 USB On-The-Go (OTG) host library, in Full Speed (FS), High Speed (HS) and High Speed in Full Speed (HS-IN-FS) modes, in order to develop an application exploiting FatFS offered features with the USB disk drive configuration.	X	-	X	-	X	X	X	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO-F722ZE	32F769IDISCOVERY	NUCLEO-F746ZG	32F746GDISCOVERY	STM32756G-EVAL	32F723EDISCOVERY	NUCLEO-F767ZI	STM32F769I-EVAL
Applications	FatFS	FatFs_USBDisk_MultipleAccess_RTOS	This application provides a description on how to use STM32Cube firmware with FatFS middleware component as a generic FAT file system module, FreeRTOS as an RTOS module based on using CMSIS-OS wrapping layer common APIs, and also STM32 USB On-The-Go (OTG) host library, in Full Speed (FS), High Speed (HS) and High Speed in Full Speed (HS-IN-FS) modes, in order to develop an application exploiting FatFS offered features with USB disk drive in RTOS mode configuration.	-	-	-	-	X	-	-	X
		FatFs_USBDisk_RTOS	This application provides a description on how to use STM32Cube firmware with FatFS middleware component as a generic FAT file system module, FreeRTOS as an RTOS module based on using CMSIS-OS wrapping layer common APIs, and also STM32 USB On-The-Go (OTG) host library, in Full Speed (FS), High Speed (HS) and High Speed in Full Speed (HS-IN-FS) modes, in order to develop an application exploiting FatFS offered features with USB disk drive in RTOS mode configuration.	-	-	-	-	X	-	-	X
		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 making the most of the features offered by FatFS to configure a microSD drive.	X	-	-	X	X	-	X	X
		FatFs_uSD_RTOS	This application provides a description on how to use STM32Cube firmware with FatFS middleware component as a generic FAT file system module, in order to develop an application exploiting FatFS offered features with microSD drive in RTOS mode configuration.	-	-	-	X	X	-	-	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO-F722ZE	32F769IDISCOVERY	NUCLEO-F746ZG	32F746GDISCOVERY	STM32756G-EVAL	32F723EDISCOVERY	NUCLEO-F767ZI	STM32F769I-EVAL
Applications	Free RTOS	FreeRTOS_DelayUntil	This directory contains a set of source files that implement thread delaying using osDelayUntil function.	-	-	-	X	X	-	-	-
		FreeRTOS_LowPower	This application shows how to enter and exit low-power mode with CMSIS RTOS API.	X	-	-	-	X	-	-	-
		FreeRTOS_MPU	This application shows how to use the MPU feature of FreeRTOS.	-	-	-	-	X	-	-	-
		FreeRTOS_Mail	This application shows how to use mail queues with CMSIS RTOS API.	X	X	-	X	X	X	-	X
		FreeRTOS_Mutexes	This application shows how to use mutexes with CMSIS RTOS API.	X	-	-	-	X	-	-	X
		FreeRTOS_Queues	This application shows how to use message queues with CMSIS RTOS API.	X	-	-	-	X	X	-	X
		FreeRTOS_Semaphore	This application shows how to use semaphores with CMSIS RTOS API	X	-	-	-	X	-	-	X
		FreeRTOS_SemaphoreFromISR	This application shows how to use semaphore from ISR with CMSIS RTOS API.	X	X	-	X	X	X	-	X
		FreeRTOS_Signal	This application shows how to use a thread signaling using CMSIS RTOS API.	X	-	-	-	X	-	-	X
		FreeRTOS_SignalFromISR	This application shows how to use a thread signaling from an interrupt using CMSIS RTOS API.	X	X	-	-	X	X	-	X
		FreeRTOS_ThreadCreation	This application shows how to implement a thread creation using CMSIS RTOS API.	X	X	-	-	X	X	-	X
		FreeRTOS_Timers	This application shows how to use timers of CMSIS RTOS API.	X	X	-	-	X	X	-	X
	GPS	GPS_Map_Tracker	This application shows how to use the Nano GPS Click module plugged on Fanout board connected to the STM32F723E-Discovery via STMOD+ to perform a real-time positioning using the STM32 Cube HAL.	-	-	-	-	-	X	-	-

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO-F722ZE	32F769IDISCOVERY	NUCLEO-F746ZG	32F746GDISCOVERY	STM32756G-EVAL	32F723EDISCOVERY	NUCLEO-F767ZI	STM32F769I-EVAL
Applications	IAP	IAP_Binary_Template	This directory contains a set of sources files that build the application to be loaded into the Flash memory using In-Application Programming (IAP, through USART).	-	-	-	-	X	-	-	-
		IAP_DualBank_Binary_Template	This directory contains a set of sources files that build the application to be loaded into the Flash memory using In-Application Programming (IAP, through USART).	-	-	-	-	-	-	-	X
		IAP_Main	This directory contains a set of source files and pre-configured projects that describes how to build an application to be loaded into the Flash memory using In-Application Programming (IAP, through USART).	-	-	-	-	X	-	-	X
		IAP_SingleBank_Binary_Template	This directory contains a set of sources files that build the application to be loaded into the Flash memory using In-Application Programming (IAP, through USART).	-	-	-	-	-	-	-	X
	LibJPEG	LibJPEG_Decoding	This application demonstrates how to read a jpeg file from SD card memory, decode it and display the final BMP image on the LCD.	-	-	-	X	X	X	-	-
		LibJPEG_Encoding	This example demonstrates how to read BMP file from micro SD, encode it, save the jpeg file in USB disk then decode the jpeg file and display the final BMP image on the LCD.	-	-	-	-	X	-	-	-
	LwIP	LwIP_HTTP_Server_Netconn_RTOS	This application guides STM32Cube HAL API users to run a http server application based on Netconn API of LwIP TCP/IP stack. The communication is done with a web browser application in a remote PC.	-	X	X	X	X	-	X	X
		LwIP_HTTP_Server_Raw	This application guides STM32Cube HAL API users to run a http server application based on Raw API of LwIP TCP/IP stack The communication is done with a web browser application in a remote PC.	-	-	-	-	X	-	-	X
		LwIP_HTTP_Server_Socket_RTOS	This application guides STM32Cube HAL API users to run a http server application based on Socket API of LwIP TCP/IP stack The communication is done with a web browser application in a remote PC.	-	X	-	-	X	-	-	X
		LwIP_IAP	This application guides STM32Cube HAL API users to run In-Application Programming (IAP) over Ethernet.	-	-	-	-	X	-	-	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO-F722ZE	32F769IDISCOVERY	NUCLEO-F746ZG	32F746GDISCOVERY	STM32756G-EVAL	32F723EDISCOVERY	NUCLEO-F767ZI	STM32F769I-EVAL
Applications	LwIP	LwIP_TCP_Echo_Client	This application guides STM32Cube HAL API users to run TCP Echo Client application based on Raw API of LwIP TCP/IP stack. To run this application, on the remote PC, open a command prompt window.	-	-	-	-	X	-	-	-
		LwIP_TCP_Echo_Server	This application guides STM32Cube HAL API users to run TCP Echo Server application based on Raw API of LwIP TCP/IP stack. To run this application, on the remote PC, open a command prompt window.	-	-	-	-	X	-	-	-
		LwIP_TFTP_Server	This application guides STM32Cube HAL API users to run a tftp server application for STM32F7xx devices.	-	-	-	-	X	-	-	-
		LwIP_UDPTCP_Echo_Server_Netconn_RTOS	This application guides STM32Cube HAL API users to run a UDP/TCP Echo Server application based on Netconn API of LwIP TCP/IP stack. To run this application, on the remote PC, open a command prompt window.	-	-	-	-	X	-	-	-
		LwIP_UDP_Echo_Client	This application guides STM32Cube HAL API users to run a UDP Echo Client application based on Raw API of LwIP TCP/IP stack. To run this application, on the remote PC, open a command prompt window.	-	-	-	-	X	-	-	-
		LwIP_UDP_Echo_Server	This application guides STM32Cube HAL API users to run UDP Echo Server application based on Raw API of LwIP TCP/IP stack. To run this application, on the remote PC, open a command prompt window.	-	-	-	-	X	-	-	-
	QSPI	QSPI_perfs	This application describes how to display pictures stored on QSPI flash memory on LCD and measures data transfer performance between QSPI Flash and SDRAM memory.	-	-	-	X	-	-	-	-

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO-F722ZE	32F769IDISCOVERY	NUCLEO-F746ZG	32F746GDISCOVERY	STM32756G-EVAL	32F723EDISCOVERY	NUCLEO-F767ZI	STM32F769I-EVAL
Applications	STemWin	STemWin_HelloWorld	This directory contains a set of source files that implement a simple "Hello World" application based on STemWin for STM32F7xx devices.	-	-	-	X	X	X	-	X
		STemWin_SampleDemo	This directory contains a set of source files that implement a sample demonstration application allowing to show some of the STemWin Library capabilities on STM32F7xx devices.	-	-	-	-	X	-	-	X
		STemWin_acceleration	This directory contains a set of source files that implement a simple "acceleration" application based on STemWin for STM32F7xx devices	-	X	-	X	X	-	-	X
		STemWin_animation	This directory contains a set of source files that implement a simple "animation" application based on STemWin for STM32F7xx devices	-	X	-	X	X	-	-	X
		STemWin_fonts	This directory contains a set of source files that implement a simple "fonts" application based on STemWin for STM32F7xx devices	-	X	-	X	X	-	-	X
		STemWin_helloworld	Simple "Hello World" example based on STemWin	-	X	-	-	-	-	-	-
		STemWin_memory_device	This directory contains a set of source files that implement a simple "memory device" application based on STemWin for STM32F7xx devices.	-	X	-	X	X	-	-	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO -F722ZE	32F769IDI SCOVERY	NUCLEO -F746ZG	32F746GD ISCOVERY	STM3275 6G-EVAL	32F723EDI SCOVERY	NUCLEO -F767ZI	STM32F7 69I-EVAL
Applications	USB_ Device	AUDIO_Standalone	This application is a part of the USB Device Library package using STM32Cube firmware. It describes how to use USB device application based on the AUDIO Class implementation of an audio streaming (Out: Speaker/Headset) capability on the STM32F7xx devices.	-	-	-	X	X	X	-	X
		CDC_Standalone	This application shows how to use the USB device application based on the Device Communication Class (CDC) compliant with the PSTN subprotocol. The USB Device and UART peripherals are used.	-	-	-	-	X	-	-	X
		CustomHID_Standalone	This application is a part of the USB Device Library package using STM32Cube firmware. It describes how to use USB device application based on the Custom HID Class on the STM32F7xx devices.	-	-	-	-	X	-	-	X
		DFU_Standalone	This application is a part of the USB Device Library package using STM32Cube firmware. It describes how to use USB device application based on the Device Firmware Upgrade (DFU) on the STM32F746xx devices.	-	-	X	X	X	X	-	-
		DualCore_Standalone	This application is a part of the USB Device Library package using STM32Cube firmware. It describes how to use USB device application based on the STM32F7xx multi core support feature integrating the Device Communication Class (CDC) and Human Interface (HID) in the same project.	-	-	-	X	X	-	-	X
		HID_LPM_Standalone	The STM32F7xx devices support the USB Link Power Management Protocol (LPM-L1) and comply with the USB 2.0 LPM-L1 ECN. The hpcd.Init.lpm_enable in the usbd_conf.c should be set to 1 to enable the support for LPM-L1 protocol in the USB stack.	-	-	-	X	X	-	-	X
		HID_Standalone	This application is a part of the USB Device Library package using STM32Cube firmware. It describes how to use USB device application based on the Human Interface (HID) on the STM32F7xx devices.	X	-	X	X	X	X	X	X
		MSC_Standalone	This application is a part of the USB Device Library package using STM32Cube firmware. It describes how to use USB device application based on the Mass Storage Class (MSC) on the STM32F7xx devices.	X	X	X	X	X	-	X	-

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO-F722ZE	32F769IDISCOVERY	NUCLEO-F746ZG	32F746GDISCOVERY	STM32756G-EVAL	32F723EDISCOVERY	NUCLEO-F767ZI	STM32F769I-EVAL
Applications	USB_Host	AUDIO_Standalone	This application is a part of the USB Host Library package using STM32Cube firmware. It describes how to use USB host application based on the Audio OUT class on the STM32F7xx devices.	-	-	-	-	X	-	-	-
		CDC_Standalone	This application is a part of the USB Host Library package using STM32Cube firmware. It describes how to use USB host application based on the Communication Class (CDC) on the STM32F7xx devices.	-	-	-	X	X	-	-	X
		DualCore_Standalone	This application is a part of the USB Host Library package using STM32Cube firmware. It describes how to use USB host application based on the STM32F7xx multi core support feature integrating Mass Storage (MSC) and Human Interface (HID) in the same project.	-	-	-	X	X	-	-	X
		DynamicSwitch_Standalone	This application is a part of the USB Host Library package using STM32Cube firmware. It describes how to use dynamically switch, on the same port, between available USB host applications on the STM32F7xx devices.	-	-	-	X	X	X	-	X
		FWupgrade_Standalone	This application is a part of the USB Host Library package using STM32Cube firmware. It describes how to use USB host application based on the In-Application programming (IAP) on the STM32F7x6 devices.	-	-	-	-	X	-	-	-
		HID_RTOS	This application is a part of the USB Host Library package using STM32Cube firmware. It describes how to use USB host application based on the Human Interface Class (HID) on the STM32F7xx devices.	-	-	-	X	X	X	-	X

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO-F722ZE	32F769IDISCOVERY	NUCLEO-F746ZG	32F746GDISCOVERY	STM32756G-EVAL	32F723EDISCOVERY	NUCLEO-F767ZI	STM32F769I-EVAL
Applications	USB_Host	HID_Standalone	This application is a part of the USB Host Library package using STM32Cube firmware. It describes how to use USB host application based on the Human Interface Class (HID) on the STM32F7xx devices.	X	-	X	X	X	X	X	X
		MSC_RTOS	This application is a part of the USB Host Library package using STM32Cube firmware. It describes how to use USB host application based on the Mass Storage Class (MSC) on the STM32F7x6 devices in RTOS mode configuration.	-	-	-	X	X	X	-	X
		MSC_Standalone	This application is a part of the USB Host Library package using STM32Cube firmware. It describes how to use USB host application based on the Mass Storage Class (MSC) on the STM32F7x6 devices.	X	X	X	X	X	X	X	X
		MTP_Standalone	This application is a part of the USB Host Library package using STM32Cube firmware. It describes how to use USB host application based on the Media Transfer Protocol (MTP) on the STM32F7xx devices.	-	-	-	-	X	-	-	X
	WiFi	Esp8266_IAP_Client	This application shows how to use the ESP8266 WiFi module to perform an IAP ("In Application Programming") using STM32 Cube HAL.	-	X	-	-	-	X	-	-
	mbedTLS	SSL_Client	This application describes how to run an SSL client application based on mbedTLS crypto library and LwIP TCP/IP stack.	-	X	-	-	X	-	-	-
		SSL_Server	This application guides the STM32Cube HAL API users to run an SSL Server application based on mbedTLS crypto library and LwIP TCP/IP stack.	-	X	-	-	X	-	-	-
	Total number of applications: 228			**17**	**20**	**8**	**29**	**70**	**25**	**8**	**51**

Table 1. STM32CubeF7 firmware examples (continued)

Level	Module Name	Project Name	Description	NUCLEO -F722ZE	32F769IDI SCOVERY	NUCLEO -F746ZG	32F746GD ISCOVERY	STM3275 6G-EVAL	32F723EDI SCOVERY	NUCLEO -F767ZI	STM32F7 69I-EVAL
Demonstrations	-	Demo	The provided demonstration firmware based on STM32Cube helps the user to discover STM32 Cortex-M devices that can be plugged on a NUCLEO-F767ZI board.	X	-	X	-	-	-	X	-
	-	STemWin	The STM32Cube demonstration platform comes on top of the STM32Cube as a firmware package that offers a full set of software components based on a module architecture allowing re-using them separately in standalone applications. All these modules are managed by the STM32Cube demonstration kernel allowing to dynamically adding new modules and access to common resources (storage, graphical components and widgets, memory management, Real-Time operating system). The STM32Cube demonstration platform is built around the powerful graphical library STemWin and the FreeRTOS real time operating system and uses almost the whole STM32 capability to offer a large scope of usage based on the STM32Cube HAL BSP and several middleware components.	-	X	-	X	X	X	-	X
	Total number of demonstrations: 8			**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Total number of projects: 760				**53**	**53**	**39**	**64**	**166**	**69**	**139**	**177**

3 Revision history

Table 2. Document revision history

Date	Revision	Changes
07-Jul-2015	1	Initial release.
26-Nov-2015	2	Updated *Table 1: STM32CubeF7 firmware examples* adding the list of examples provided with STM32F746ZG-Nucleo board.
02-May-2016	3	Updated *Table 1: STM32CubeF7 firmware examples* adding the list of examples, applications and demonstrations provided with the STM32F769I-EVAL, STM32F769I-Discovery and STM32F767ZI-Nucleo boards. Updated *Reference documents* list.
09-Feb-2017	4	Updated *Figure 1: STM32CubeF7 firmware components*. Updated *STM32CubeF7 examples* adding examples_LL and examples_MIX. Updated *Reference documents* list. Updated *Table 1: STM32CubeF7 firmware examples* adding the list of examples, examples_LL, examples_MIX, applications and demonstrations provided with the STM32F723E-Discovery and the STM32F722ZE-Nucleo boards.
02-Jul-2018	5	Updated *Section 1: Reference documents*. Updated *Table 1: STM32CubeF7 firmware examples* adding applications.