## STM32Cube firmware examples for STM32H7 Series

Introduction

The STM32CubeH7 firmware 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. STM32CubeH7 firmware components

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

MS45432V1
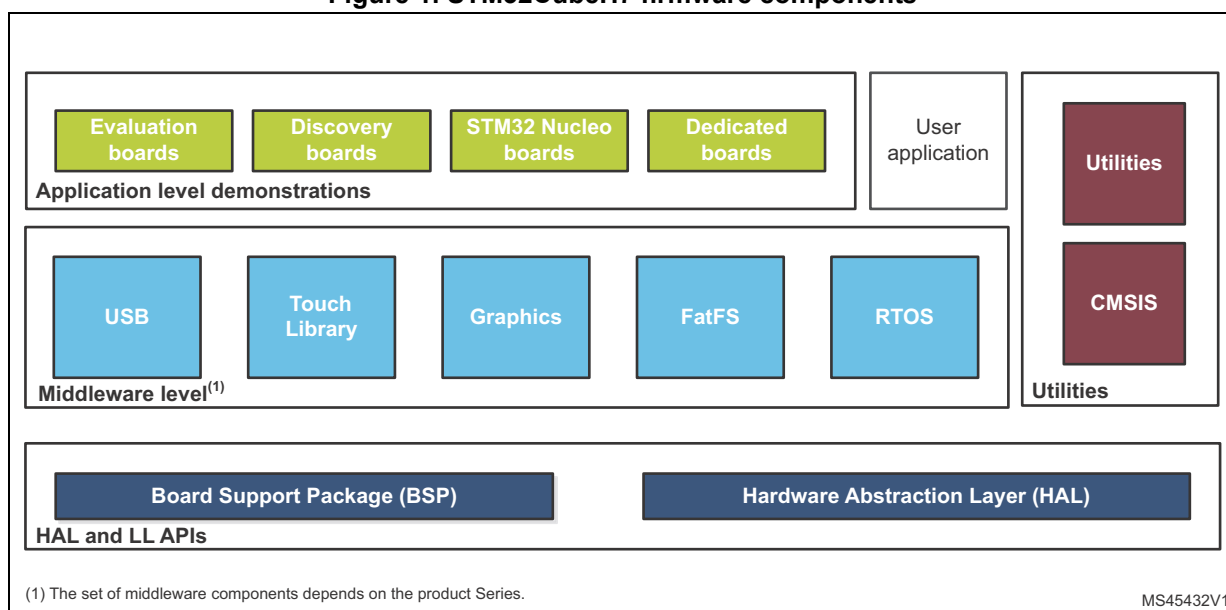
Contents

1   Reference documents

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

- Latest release of STM32CubeH7 firmware package
- *Getting started with STM32CubeH7 for STM32H7 Series* (UM2204)
- *STM32CubeH7 demonstration platform* (UM2222)
- *Description of STM32H7 HAL drivers* (UM2217)
- *STM32Cube USB Device library* (UM1734)
- *STM32Cube USB host library* (UM1720)
- *Developing applications on STM32Cube with FatFS* (UM1721)
- *Developing applications on STM32Cube with RTOS* (UM1722)
- *Developing applications on STM32Cube with LwIP TCP/IP stack* (UM1713)
- *STM32Cube Ethernet IAP example* (UM1709)

The microcontrollers of the STM32H7 Series are based on Arm® cores.

arm

2 STM32CubeH7 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.

- **Applications:** the applications demonstrate the product performance and how to use the available middleware stacks. They are organized either by middleware (a folder per middleware, e.g. USB Host) or by product feature that require high-level firmware bricks (e.g. Audio). The integration of applications that use several middleware stacks is also supported.

- **Demonstrations:** the demonstrations aim at integrating and running the maximum number of peripherals and middleware stacks to showcase the product features and performance.

- **Template project:** the template project is provided to allow the user to quickly build a firmware application using HAL and BSP drivers on a given board.

The examples are located under *STM32Cube_FW_STM32CubeH7_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 1 contains the list of examples provided with STM32CubeH7 firmware package.

Table 1. STM32CubeH7 firmware examples

Level	Module name	Project Name	Description	STM32H743ZI -Nucleo	STM32H743I -EVAL
Demonstration	-	-	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 modular architecture allowing re-using them separately in standalone applications. All these modules are managed by the STM32Cube demonstration kernel allowing to dynamically add new modules and access common resources (storage, graphical components and widgets, memory management, real-time operating system). The STM32Cube demonstration platform is built around the powerful graphical STemWin library and the FreeRTOS real-time operating system. It uses almost the whole STM32 capability to offer a large scope of usage based on the STM32Cube HAL BSP and several middleware components.	-	X
Total number of demonstration: 1				**0**	**1**
Templates	-	Starter project	This projects provides a reference template that can be used to build any firmware application.	X	X
Total number of templates: 2				**1**	**1**

Table 1. STM32CubeH7 firmware examples (continued)

Level	Module name	Project Name	Description	STM32H743ZI -Nucleo	STM32H743I -EVAL
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_DAC_Interconnect	This example describes how to configure and connect DAC output to ADC input and use the analog watchdog to monitor signal behavior.	X	X
		ADC_DMA_Transfer	This example describes how to configure and use the ADC to convert an external analog input and get the result using a DMA transfer through the HAL API.	X	X
		ADC_DifferentialMode	This example describes how to configure and use the ADC to convert an external analog input in Differential mode (difference between the external voltage on VINN and VINP).	X	X
		ADC_DualMode Interleaved	This example provides a short description of how to use two ADC peripherals to perform conversions in Interleaved dual-mode.	X	X
		ADC_InternalChannel Conversion	This example describes how to configure and use the ADC to retrieve the system internal voltage reference.	X	X
		ADC_OverSampler	This example describes how to configure and use the ADC to convert an external analog input combined with oversampling feature to increase resolution through the HAL API.	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
		ADC_Regular_injected_ groups	This example provides a short description of how to use the ADC peripheral to perform conversions using the two ADC groups: regular group for ADC conversions on main stream and injected group for ADC conversions limited on specific events (conversions injected within main conversions stream).	X	X
	CEC	CEC_DataExchange	This example shows how to configure and use the CEC peripheral to receive and transmit messages.	-	X

Table 1. STM32CubeH7 firmware examples (continued)

Level	Module name	Project Name	Description	STM32H743ZI -Nucleo	STM32H743I -EVAL
Examples	COMP	COMP_AnalogWatchdog	This example shows how to make an analog watchdog using the COMP peripherals in Window mode.	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.	X	X
		COMP_OutputBlanking	This example shows how to use the output blanking feature of COMP peripheral.	X	X
	CRC	CRC_Bytes_Stream_7bit _CRC	This example guides you through the different configuration steps by means of the HAL API. The CRC (cyclic redundancy check) calculation unit computes 7-bit long CRC codes derived from buffers of 8-bit data (bytes).	X	X
		CRC_Example	This example guides you through the different configuration steps by means of the HAL API. The CRC (cyclic redundancy check) calculation unit computes the CRC code of a given buffer of 32-bit data words, using a fixed generator polynomial (0x4C11DB7).	X	X
		CRC_UserDefined Polynomial	This example guides you 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

Table 1. STM32CubeH7 firmware examples (continued)

Level	Module name	Project Name	Description	STM32H743ZI -Nucleo	STM32H743I -EVAL
Examples	CRYP	CRYP_AESCCM_IT	This example provides a short description of how to use the CRYP peripheral to encrypt/decrypt data (Plaintext/Ciphertext) in Interrupt mode using AES with Combined Cipher Machine (CCM), then generate the authentication TAG.	-	X
		CRYP_AESGCM	This example provides a short description of how to use the CRYP peripheral to encrypt/decrypt data (Plaintext/Ciphertext) using AES Galois/counter mode (GCM) and generate the authentication TAG.	-	X
		CRYP_AESModes	This example provides a short description of how to use the CRYP peripheral to encrypt/decrypt data (Plaintext/Ciphertext) using AES ECB, CBC and CTR algorithm.	-	X
		CRYP_AESModes_DMA	This example provides a short description of how to use the CRYP peripheral to encrypt/decrypt data (Plaintext/Ciphertext) using AES ECB algorithm in DMA mode.	-	X
		CRYP_TDESModes	This example provides a short description of how to use the CRYP peripheral to encrypt/decrypt data (Plaintext/Ciphertext) using TDES ECB and CBC algorithm.	-	X
	DAC	DAC_DualConversion	This example provides a short description of how to use the DAC peripheral to perform a dual conversion.	X	X
		DAC_SignalsGeneration	This example provides a description of how to use the DAC peripheral to generate several signals using the DMA controller.	X	X
		DAC_SimpleConversion	This example provides a short description of how to use the DAC peripheral to perform a simple conversion.	X	X
	DFSDM	DFSDM_AudioRecord	This example shows how to use the DFSDM HAL API to perform stereo audio recording.	-	X

Table 1. STM32CubeH7 firmware examples (continued)

Level	Module name	Project Name	Description	STM32H743ZI -Nucleo	STM32H743I -EVAL
Examples	DMA	DMAMUX_RequestGen	This example shows how to use the DMA with the DMAMUX request generator to generate DMA transfer requests upon LPTIM2 output signal, knowing that the LPTIM2 is configured in PWM with 2 s period.	X	X
		DMAMUX_SYNC	This example shows how to use the DMA with the DMAMUX to synchronize a transfer with LPTIM1 output signal.	-	X
		DMA_FIFOMode	This example provides a description of how to use a DMA to transfer a word data buffer from Flash memory to embedded SRAM with FIFO mode enabled through the HAL API.	-	X
		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
	DMA2D	DMA2D_BlendingWith AlphaInversion	This example provides a description of how to configure DMA2D peripheral in Memory-to-Memory transfer mode with blending transfer and alpha inversion mode.	-	X
		DMA2D_MemToMem WithBlending	This example provides a description of how to configure DMA2D peripheral in Memory-to-Memory transfer mode with blending transfer mode.	-	X
		DMA2D_MemToMem WithBlendingAndCLUT	This example provides a description of how to configure DMA2D peripheral in Memory-to-Memory transfer mode with blending transfer mode and indexed 256 color images (L8).	-	X
		DMA2D_MemToMem WithPFCandRedBlue Swap	This example provides a description of how to configure 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_MemoryTo Memory	This example provides a description of how to configure the DMA2D peripheral in Memory-to-Memory transfer mode.	-	X
		DMA2D_RegToMem WithLCD	This example provides a description of how to configure DMA2D peripheral in Register-to-Memory transfer mode and display the result on LCD.	-	X

Table 1. STM32CubeH7 firmware examples (continued)

Level	Module name	Project Name	Description	STM32H743ZI -Nucleo	STM32H743I -EVAL
Examples	FDCAN	FDCAN_Classic_Frame _Networking	This example shows how to configure the FDCAN peripheral to send and receive Classic CAN frames in normal mode.	-	X
		FDCAN_Clock_ calibration	This example shows how to achieve clock calibration on an FDCAN unit.	-	X
		FDCAN_Com_IT	This example shows how to achieve Interrupt Process Communication between two FDCANs.	-	X
		FDCAN_Com_polling	This example shows how to achieve Polling Process Communication between two FDCANs.	-	X
		FDCAN_Image_ transmission	This example shows the gain in time obtained by the activation of the Bit Rate Switching (BRS) feature.	-	X
		FDCAN_Loopback	This example shows how to configure the FDCAN to operate in Loopback mode.	-	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
		FLASH_SwapBank	This example guides you through the different configuration steps by mean of HAL API to swap execution between bank 1 and bank 2 of the STM32H7xx internal Flash memory.	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

Table 1. STM32CubeH7 firmware examples (continued)

Level	Module name	Project Name	Description	STM32H743ZI -Nucleo	STM32H743I -EVAL
Examples	FMC	FMC_NOR	This example guides you through the different configuration steps by mean of HAL API to configure the FMC controller in order to access the PC28F128M29EWLA NOR memory mounted on STM32H743I-EVAL evaluation board.	-	X
		FMC_SDRAM	This example describes how to configure the FMC controller to access the SDRAM.	-	X
		FMC_SDRAM_Data Memory	This example describes how to configure the FMC controller to access the SDRAM including heap and stack.	-	X
		FMC_SDRAM_Low Power	This example describes how to configure the FMC controller to access the SDRAM in low-power mode (SDRAM Self-refresh mode).	-	X
		FMC_SRAM	This example describes how to configure the FMC controller to access the SRAM.	-	X
	GPIO	GPIO_EXTI	This example shows how to configure external interrupt lines.	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
		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
		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
		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

Table 1. STM32CubeH7 firmware examples (continued)

Level	Module name	Project Name	Description	STM32H743ZI -Nucleo	STM32H743I -EVAL
Examples	HRTIM	HRTIM_Arbitrary_ Waveform	This example shows how to configure the HRTIM1 peripheral to generate an arbitrary signals.	X	X
		HRTIM_DAC_ADC_ Interconnect	This example shows how to use the interconnection feature between HRTIM, DAC and ADC.	X	X
		HRTIM_ExternalEvents	This example shows how to use an External event to set and reset the HRTIM.	X	X
		HRTIM_FaultEvent	This example shows how to configure the HRTIM peripheral in PWM mode, and how to configure and use the Fault event.	X	X
		HRTIM_MultiplePWM	This example shows how to configure the HRTIM1 peripheral to generate up to five PWM signals with different duty cycle for each HRTIM output.	X	X
		HRTIM_PWM_Different Frequencies	This example shows how to configure the HRTIM1 peripheral to generate up to six PWM signals with different timebase configuration for each slave timer.	X	X
	HSEM	HSEM_ProcessSync	This example shows how to use a HW semaphore to synchronize two processes.	X	X
		HSEM_ReadLock	This example shows how to enable, take then release semaphore using two different processes.	X	X
	I2C	I2C_EEPROM_fast_ mode_plus	This example describes how to perform I2C data buffer transmission/reception via DMA. The communication uses an I2C EEPROM memory.	-	X
		I2C_TwoBoards_ ComDMA	This example describes how to perform I2C data buffer transmission/reception between two boards via DMA.	X	-
		I2C_TwoBoards_ComIT	This example describes how to perform I2C data buffer transmission/reception between two boards using an interrupt.	X	-
		I2C_TwoBoards_ ComPolling	This example describes how to perform I2C data buffer transmission/reception between two boards in Polling mode.	X	-
		I2C_WakeUpFromStop	This example describes how to perform I2C data buffer transmission/reception between two boards using an interrupt when the device is in Stop mode.	X	X

Table 1. STM32CubeH7 firmware examples (continued)

Level	Module name	Project Name	Description	STM32H743ZI -Nucleo	STM32H743I -EVAL
Examples	IWDG	IWDG_WindowMode	This example describes how to periodically update the IWDG reload counter and simulate a software fault that generates an MCU IWDG reset when a programmed time period has elapsed.	X	X
	JPEG	JPEG_DecodingFrom FLASH_DMA	This example demonstrates how to decode a JPEG image stored in the internal Flash memory using the JPEG hardware decoder in DMA mode, and display the final ARGB8888 image on an LCD mounted on the board.	-	X
		JPEG_DecodingUsing Fs_DMA	This example demonstrates how to read JPEG file from an SD Card memory using FatFS, decode it using the JPEG hardware decoder in DMA mode, and display the final ARGB8888 image on the LCD-TFT screen.	-	X
		JPEG_DecodingUsing Fs_Interrupt	This example demonstrates how to read JPEG file from an SD Card memory using FatFS, decode it using the JPEG hardware decoder in Interrupt mode, and display the final ARGB8888 image on the LCD-TFT screen.	-	X
		JPEG_DecodingUsing Fs_Polling	This example demonstrates how to read JPEG file from an SD Card memory using FatFS, decode it using the JPEG hardware decoder in Polling mode, and display the final ARGB8888 image on the LCD-TFT screen.	-	X
		JPEG_EncodingFrom FLASH_DMA	This example demonstrates how to read an RGB image stored in the internal Flash memory, encode it using the JPEG hardware encoder in DMA mode, and save it on an SD Card.	-	X
		JPEG_EncodingUsing Fs_DMA	This example demonstrates how to read BMP file from an SD Card memory using FatFS, encode it using the JPEG hardware encoder in DMA mode, and save it back on the SD Card.	-	X
		JPEG_MJPEG_Video Decoding	This example demonstrates how to use the hardware JPEG decoder to decode an MJPEG video file and display it on the LCD-TFT screen.	-	X
		JPEG_MJPEG_Video DecodingFromQSPI	This example demonstrates how to use the hardware JPEG decoder to decode an MJPEG video file located in the external QuadSPI Flash memory and display it on the LCD-TFT screen.	-	X

Table 1. STM32CubeH7 firmware examples (continued)

Level	Module name	Project Name	Description	STM32H743ZI-Nucleo	STM32H743I-EVAL
Examples	LPTIM	LPTIM_Encoder	This example shows how to configure the LPTIM peripheral in Encoder mode.	X	X
		LPTIM_PWMExternal Clock	This example describes how to configure and use the 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 the 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 the 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 LPTIM, through the HAL LPTIM API.	X	X
	LTDC	LTDC_ColorKeying	This example describe how to enable and use the LTDC color keying functionality.	-	X
		LTDC_Display_1Layer	This example provides a description of how to configure the LTDC peripheral to display a 480x272 RGB image and format RGB565 (16 bits/pixel) on the LCD using only one layer.	-	X
		LTDC_Display_2Layers	This example describes how to configure the LTDC peripheral to display two layers simultaneously.	-	X

Table 1. STM32CubeH7 firmware examples (continued)

Level	Module name	Project Name	Description	STM32H743ZI -Nucleo	STM32H743I -EVAL
Examples	MDMA	MDMA_DMA2D_ Triggering	This example describes how to use the MDMA with hardware trigger set to the DMA2D transfer complete flag.	-	X
		MDMA_GPDMA_ Triggering	This example describes how to use the MDMA with hardware trigger set to D2 Domain GP-DMA transfer complete flag.	-	X
		MDMA_LTDC_ Triggering	This example describes how to use the MDMA with hardware trigger set to the LTDC Line Interrupt Flag.	-	X
		MDMA_LinkedList	This example shows how to use the MDMA to perform a list of transfers. The transfer list is organized as a linked list: each time the current transfer completes, the MDMA automatically reloads the next transfer parameters and starts it without the intervention of the CPU.	X	X
		MDMA_LinkedList_ ColorsComp	This example demonstrates how to use the MDMA in linked list mode to extract red/green and blue colors from an ARGB8888 image, resize each sub-image (with a decimation factor divided by 2), and display the resulting red/green/blue decimated sub-images on the LCD.	-	X
		MDMA_RepeatBlock_ Rotation	This example provides a description of how to use the MDMA in repeat block trigger mode in order to copy an RGB565 image to the LCD frame buffer.	-	X
		MDMA_RepeatBlock_ ZoomOut	This example provides a description of how to use the MDMA in repeat block trigger mode in order to decimate an RGB565 image and copy it to the LCD frame buffer.	-	X
	OPAMP	OPAMP_Calibration	This example shows how to calibrate the OPAMP.	X	-
		OPAMP_Follower	This example shows how to configure OPAMP peripheral in follower mode interconnected with DAC and COMP.	X	X
		OPAMP_PGA_External Bias	This example shows how to configure OPAMP peripheral in PGA mode with bias voltage for the non-inverting mode.	X	X

Table 1. STM32CubeH7 firmware examples (continued)

Level	Module name	Project Name	Description	STM32H743ZI -Nucleo	STM32H743I -EVAL
Examples	PWR	PWR_Domain3System Control	This example shows how to maintain a basic activity of the system in low-power mode with D3 Domain only by ensuring the communication with the D3SRAM, the BDMA and the I2C4 when the CPU is in Stop mode.	X	X
		PWR_STANDBY	This example shows how to enter Standby mode and wake up from this mode using an external reset or the WKUP pin connected to the user button.	X	X
		PWR_STANDBY_RTC	This example shows how to enter Standby mode and wake up from this mode using an external reset or the RTC wakeup timer.	X	X
		PWR_STOP_DataRetain	This example shows how to retain data in D3 SRAM when the system enter Stop mode with D1 domain in Standby mode to guarantee low-power consumption.	X	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	X

Table 1. STM32CubeH7 firmware examples (continued)

Level	Module name	Project Name	Description	STM32H743ZI -Nucleo	STM32H743I -EVAL
Examples	QSPI	QSPI_ExecuteInPlace	This example describes how to execute a part of the code from the QuadSPI memory. To do this, a section is created where the function is stored.	-	X
		QSPI_MemoryMapped	This example describes how to erase part of the QuadSPI memory, write data in DMA mode, and access the QuadSPI memory in memory-mapped mode to check the data in a forever loop.	-	X
		QSPI_MemoryMapped Dual	This example describes how to erase part of the QuadSPI memory, write data in DMA mode, and access the QuadSPI memory in memory-mapped dual mode to check the data in a forever loop.	-	X
		QSPI_ReadWriteDual_ DMA	This example describes how to use the QuadSPI interface in Dual mode. It erases part of the QuadSPI memory, writes data in DMA mode, reads back 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 QuadSPI memory, read and write data in DMA mode.	-	X
		QSPI_ReadWrite_IT	This example describes how to erase part of the QuadSPI memory, write data in Interrupt mode, read data in Interrupt mode, and compare the result in a forever loop.	-	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
	RNG	RNG_MultiRNG	This example guides you through the HAL API different configuration steps to ensure 32-bit long random numbers generation by the RNG peripheral.	X	X

Table 1. STM32CubeH7 firmware examples (continued)

Level	Module name	Project Name	Description	STM32H743ZI -Nucleo	STM32H743I -EVAL
Examples	RTC	RTC_Alarm	This example guides you through the different configuration steps by means of the RTC HAL API to configure and generate an RTC alarm.	X	X
		RTC_Tamper	This example guides you through the different configuration steps by means of the RTC HAL API to write/read data to/from RTC Backup registers. It also demonstrates the tamper detection feature.	X	X
		RTC_TimeStamp	This example guides you through the different configuration steps by means of the RTC HAL API to demonstrate the timestamp feature.	X	X
	SAI	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
		SAI_AudioPlayback	This example shows how to use the SAI to playback audio data coming from two Microphones.	-	X
	SD	SD_ReadWrite_DMA	This example shows how to support UHS-I SD Card and achieve a frequency of 100 MHz.	-	X
		SD_ReadWrite_DMA DoubleBuffer	This example shows how to support UHS-I SD Card and achieve a frequency of 100 MHz.	-	X
		SD_ReadWrite_DMA_ HS	This example shows how to support UHS-I SD Card and achieve a frequency of 145 MHz.	-	X
		SD_ReadWrite_IT	This example shows how to support UHS-I SD Card and achieve a frequency of 100 MHz.	-	X
	SPI	SPI_FullDuplex_Com DMA	This example shows how to perform SPI data buffer transmission/reception between two boards via DMA.	X	-
		SPI_FullDuplex_ComIT	This example shows how to ensure SPI data buffer transmission/reception between two boards by using an interrupt.	X	-
		SPI_FullDuplex_Com Polling	This example shows how to ensure SPI data buffer transmission/reception in Polling mode between two boards.	X	-

Table 1. STM32CubeH7 firmware examples (continued)

Level	Module name	Project Name	Description	STM32H743ZI -Nucleo	STM32H743I -EVAL
Examples	TIM	TIM_6Steps	This example shows how to configure the TIM1 peripheral to generate six steps.	X	X
		TIM_Asymetric	This example shows how to configure the TIM peripheral to generate an asymmetric signal.	X	X
		TIM_Combined	This example shows how to configure the TIM1 peripheral to generate three PWM combined signals with TIM1 Channel 5.	X	X
		TIM_Complementary Signals	This example shows how to configure the TIM1 peripheral to generate three complementary TIM1 signals, insert a defined dead-time value, use the break feature, and 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 (TIMx_CCR3).	X	X
		TIM_DMABurst	This example shows how to update the TIMER channel 1 period and duty cycle using the TIMER DMA burst feature.	X	X
		TIM_InputCapture	This example shows how to use the TIM peripheral to measure the frequency of an external signal.	X	X
		TIM_OCToggle	This example shows how to configure the TIM peripheral to generate four different signals with four different frequencies.	X	X
		TIM_OnePulse	This example shows how to use the TIMER peripheral to generate a single pulse when a rising edge of an external signal is received on the TIMER Input pin.	X	X
		TIM_PWMOutput	This example shows how to configure the TIM peripheral in PWM (pulse width modulation) mode.	X	X
		TIM_Synchronization	This example shows how to synchronize TIM1 with TIM3 and TIM4 timers, in parallel mode.	X	X
		TIM_TimeBase	This example shows how to configure the TIM peripheral to generate a timebase of one second with the corresponding Interrupt request.	X	X

Table 1. STM32CubeH7 firmware examples (continued)

Level	Module name	Project Name	Description	STM32H743ZI -Nucleo	STM32H743I -EVAL
Examples	UART	LPUART_WakeUpFrom Stop	This example shows how to configure an LPUART to wake up the MCU from Stop mode when a given stimulus is received.	-	X
		UART_HyperTerminal_ DMA	This example describes an UART transmission (transmit/receive) in DMA mode between a board and an HyperTerminal PC application.	-	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
		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
		UART_TwoBoards_Com DMA	This example describes an UART transmission (transmit/receive) in DMA mode between two boards.	X	-
		UART_TwoBoards_Com IT	This example describes an UART transmission (transmit/receive) in interrupt mode between two boards.	X	-
		UART_TwoBoards_Com Polling	This example describes an UART transmission (transmit/receive) in polling mode between two boards.	X	-
		UART_WakeUpFromSto pUsingFIFO	This example shows how to use UART HAL API to wake up the MCU from Stop mode using the UART FIFO level.	-	X
	USART	USART_SlaveMode	This example describes an USART-SPI communication (transmit/receive) between two boards where he USART is configured as a slave.	X	-
	WWDG	WWDG_Example	This example guides you through the different configuration steps by means of the HAL API to perform periodic WWDG counter updates and simulate a software fault that generates an MCU WWDG reset when a predefined time period has elapsed.	X	X
Total number of examples: 206				**75**	**131**

Table 1. STM32CubeH7 firmware examples (continued)

Level	Module name	Project Name	Description	STM32H743ZI -Nucleo	STM32H743I -EVAL
Applications	Audio	Audio_playback_and_ record	This application shows how to use the different SAI (serial audio interface) features to ensure audio recording and playback via ST MEMS microphones (MP34DT01).	-	X
	Display	LTDC_AnimatedPicture FromSDCard	This application describes how to display an animated picture on an LCD saved under microSD.	-	X
		LTDC_Paint	This application describes how to configure an LCD touchscreen and attribute an action related to configured touch zone, and how to save BMP picture on an USB disk.	-	X
		LTDC_PicturesFrom SDCard	This application describes how to display pictures on an LCD saved under SD Card.	-	X
	EEPROM	EEPROM_Emulation	This application describes the software solution for substituting standalone EEPROM by emulating the EEPROM mechanism using STM32H743x on-chip Flash memory.	X	X
	FPU	FPU_Fractal	This application explains how to use STM32H7 floating-point unit (FPU) and demonstrates the benefits it brings. The ARM® Cortex®-M7 FPU is an implementation of the ARM FPv5 double-precision FPU.	-	X
	FatFs	FatFs_MultiDrives	This application describes how to use STM32Cube firmware with FatFS middleware component as a generic FAT file system module, in order to develop an application that exploits FatFS features with multidrive (USB Disk, microSD) configuration.	-	X
		FatFs_RAMDisk	This application describes how to use STM32Cube firmware with FatFS middleware component as a generic FAT file system module, in order to develop an application that exploits FatFS features with RAM disk (SRAM) drive configuration.	-	X
		FatFs_SDRAMDisk	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 making the most of the features offered by FatFS to configure a RAM disk drive.	-	X

Table 1. STM32CubeH7 firmware examples (continued)

Level	Module name	Project Name	Description	STM32H743ZI-Nucleo	STM32H743I-EVAL
Applications	FatFs	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 and STM32 USB On-The-Go (OTG) host library in Full Speed (FS) and High Speed (HS) modes, in order to develop an application that exploits FatFS features with USB disk drive configuration.	-	X
		FatFs_USBDisk_ Standalone	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) and High Speed (HS) and modes, in order to develop an application that exploits FatFS features with USB disk drive configuration.	-	X
		FatFs_uSD_DMA_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 that exploits FatFS features with microSD drive in RTOS mode configuration.	-	X
		FatFs_uSD_DMA_ Standalone	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 making the most of the features offered by FatFS to configure a microSD drive.	-	X
		FatFs_uSD_Standalone	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 that makes the most of the features offered by FatFS to configure a microSD drive.	X	X
	FreeRTOS	FreeRTOS_MPU	This application aims to describe how to use the MPU feature of FreeRTOS.	X	X
		FreeRTOS_Semaphore FromISR	This application shows how to use semaphore from ISR with CMSIS RTOS API.	X	X
		FreeRTOS_Thread Creation	This application shows how to implement a thread creation using CMSIS RTOS API.	X	X

Table 1. STM32CubeH7 firmware examples (continued)

Level	Module name	Project Name	Description	STM32H743ZI -Nucleo	STM32H743I -EVAL
Applications	IAP	IAP_Binary_Template	This directory contains a set of sources files to 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 preconfigured projects that describes how to build an In-application programming (IAP) that uses an USART interface to load an application binary.	-	X
	LibJPEG	LibJPEG_Decoding	This application demonstrates how to use the libjpeg API to decode a JPEG file.	-	X
		LibJPEG_Encoding	This application demonstrates how to use the libjpeg API to encode and decode a BMP image into a JPEG file.	-	X
	LwIP	LwIP_HTTP_Server_ Netconn_RTOS	This application demonstrates how to run an http server application based on Netconn API of LwIP TCP/IP stack The communication is done with a web browser application hosted on a remote PC.	X	X
		LwIP_HTTP_Server_ Raw	This application demonstrates how to run an http server application based on Raw API of LwIP TCP/IP stack. The communication is done with a web browser application hosted on a remote PC.	-	X
		LwIP_HTTP_Server_ Socket_RTOS	This application demonstrates how to run a http server application based on Socket API of LwIP TCP/IP stack. The communication is done with a web browser application hosted on a remote PC.	-	X
		LwIP_TCP_Echo_Client	This application demonstrates how to run TCP Echo Client application based on Raw API of LwIP TCP/IP stack To run this application, open a command prompt window on the remote PC.	-	X
		LwIP_TCP_Echo_Server	This application demonstrates how to run TCP Echo Server application based on Raw API of LwIP TCP/IP stack To run this application, open a command prompt window on the remote PC.	-	X

Table 1. STM32CubeH7 firmware examples (continued)

Level	Module name	Project Name	Description	STM32H743ZI-Nucleo	STM32H743I-EVAL
Applications	LwIP	LwIP_TFTP_Server	This application demonstrates how to run a tftp server application for STM32H7xx devices.	-	X
		LwIP_UDPTCP_Echo_Server_Netconn_RTOS	This application demonstrates how to run a UDP/TCP Echo Server application based on Netconn API of LwIP TCP/IP stack. To run this application, open a command prompt window on the remote PC.	-	X
		LwIP_UDP_Echo_Client	This application demonstrates how to run a UDP Echo Client application based on Raw API of LwIP TCP/IP stack. To run this application, open a command prompt window on the remote PC.	-	X
		LwIP_UDP_Echo_Server	This application demonstrates how to run UDP Echo Server application based on Raw API of LwIP TCP/IP stack. To run this application, open a command prompt window on the remote PC.	-	X
	STemWin	STemWin_HelloWorld	This directory contains a set of source files that implement a simple "Hello World" application based on STemWin for STM32H743xx devices.	-	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 STM32H743xx devices.	-	X

Table 1. STM32CubeH7 firmware examples (continued)

Level	Module name	Project Name	Description	STM32H743ZI -Nucleo	STM32H743I -EVAL
Applications	USB_ Device	Audio_Standalone	This application is a part of the USB Device Library package included in the STM32Cube firmware. It describes how to use USB device application based on the Audio Class implementation of an audio streaming capability on the STM32H7xx devices. The output is the speaker headset.	-	X
		CDC_Standalone	This application is a part of the USB Device Library package included in the STM32Cube firmware. It describes how to use USB device application based on the Device Communication Class (CDC) following the PSTN subprotocol on the STM32H743xx devices. The OTG-USB and UART peripherals are used.	-	X
		CustomHID_Standalone	This application is a part of the USB Device Library package included in the STM32Cube firmware. It describes how to use USB device application based on the Custom HID Class on the STM32H743xx devices.	-	X
		DFU_Standalone	This application is a part of the USB Device Library package included in the STM32Cube firmware. It describes how to use USB device application based on the Device Firmware Upgrade (DFU) on the STM32H7xx devices.	-	X
		DualCore_Standalone	This application is a part of the USB Device Library package included in the STM32Cube firmware. It describes how to use USB device application based on the STM32H7xx multicore support feature integrating the Device Communication Class (CDC) and Human Interface (HID) in the same project.	-	X
		HID_LPM_Standalone	The STM32H7xx devices support the USB Link Power Management Protocol (LPM-L1) and complies with the USB 2.0 LPM-L1 ECN.	-	X
		HID_Standalone	This application is a part of the USB Device Library package included in the STM32Cube firmware. It describes how to use USB device application based on the Human Interface (HID) on the STM32H743xx devices.	-	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 STM32H7xx devices.	-	X

Table 1. STM32CubeH7 firmware examples (continued)

Level	Module name	Project Name	Description	STM32H743ZI -Nucleo	STM32H743I -EVAL
Applications	USB_Host	AUDIO_Standalone	This application is a part of the USB Host Library package included in the STM32Cube firmware. It describes how to use USB host application based on the Audio OUT class on the STM32H7xx devices.	-	X
		CDC_Standalone	This application is a part of the USB Host Library package included in the STM32Cube firmware. It describes how to use USB host application based on the Communication Class (CDC) on the STM32H7xx devices.	-	X
		DualCore_Standalone	This application is a part of the USB Host Library package included in the STM32Cube firmware. It describes how to use USB host application based on the STM32H7x multicore support feature integrating Mass Storage (MSC) and Human Interface (HID) in the same project.	-	X
		DynamicSwitch_ Standalone	This application is a part of the USB Host Library package included in the STM32Cube firmware. It describes how to dynamically switch, on the same port, between available USB host applications on the STM32H7xx devices.	-	X
		FWupgrade_Standalone	This application is a part of the USB Host Library package included in the STM32Cube firmware. It describes how to use USB host application based on the In-Application programming (IAP) on the STM32H7xx devices.	-	X
		HID_RTOS	This application is a part of the USB Host Library package included in the STM32Cube firmware. It describes how to use USB host application based on the Human Interface Class (HID) on the STM32H7x devices.	-	X
		HID_Standalone	This application is a part of the USB Host Library package included in the STM32Cube firmware. It describes how to use USB host application based on the Human Interface Class (HID) on the STM32H7x devices.	-	X
		MSC_RTOS	This application is a part of the USB Host Library package included in the STM32Cube firmware. It describes how to use USB host application based on the Mass Storage Class (MSC) on the STM32H7x devices.	-	X

Table 1. STM32CubeH7 firmware examples (continued)

Level	Module name	Project Name	Description	STM32H743ZI -Nucleo	STM32H743I -EVAL
Applications	USB_Host	MSC_Standalone	This application is a part of the USB Host Library package included in the STM32Cube firmware. It describes how to use USB host application based on the Mass Storage Class (MSC) on the STM32H7x devices.	-	X
	mbedTLS	SSL_Client	This application describes how to run an SSL client application based on mbedTLS cryptographic library and LwIP TCP/IP stack.	-	X
		SSL_Server	This application describes how to run an SSL server application based on mbedTLS cryptographic library and LwIP TCP/IP stack.	-	X
Total number of applications: 57				**6**	**51**
Demonstrations	-	Demo	The provided demonstration firmware based on STM32Cube helps you to discover the STM32 Cortex-M devices that can be plugged on a NUCLEO-H743ZI board.	X	-
Total number of demonstrations: 1				**1**	**0**
Total number of projects: 266				**83**	**184**

3 Revision history

Table 2. Document revision history

Date	Revision	Changes
12-May-2017	1	Initial release.
05-Sep-2017	2	Updated applications and demonstrations in *Table 1: STM32CubeH7 firmware examples*.
02-Jan-2018	3	Updated *Section 1: Reference documents*. Updated *Table 1: STM32CubeH7 firmware examples*.

IMPORTANT NOTICE – PLEASE READ CAREFULLY