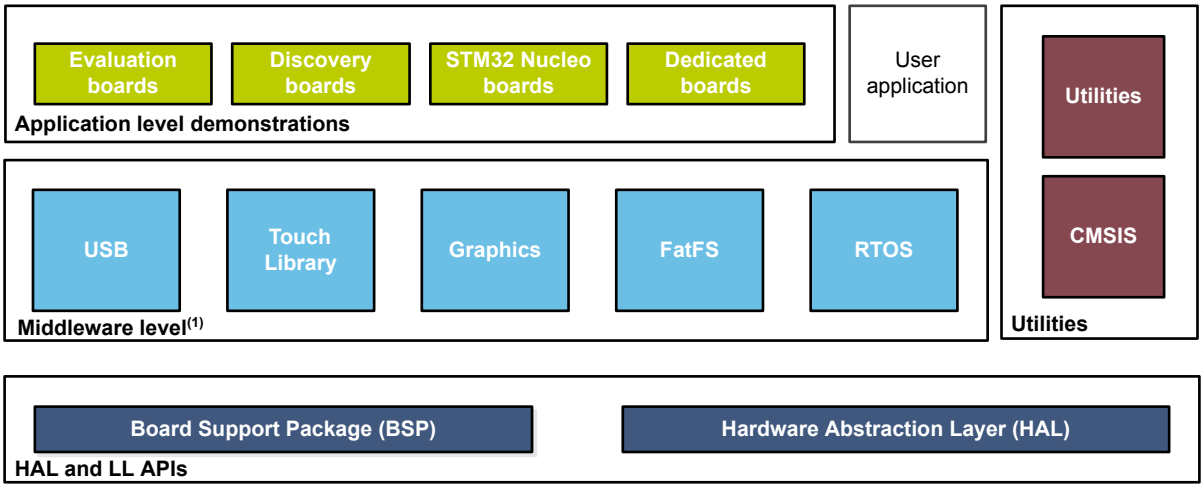
STM32Cube MCU Package examples for STM32H7 Series

Introduction

The STM32CubeH7 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 below).

Figure 1. STM32CubeH7 firmware components



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

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® Cortex® cores.

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

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 MCU Package 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. STM32CubeH7 firmware examples contains the list of examples provided with STM32CubeH7 MCU Package.

Table 1. STM32CubeH7 firmware examples

Level	Module Name	Project Name	Description	NUCLEO-H743ZI	STM32H743I-EVAL
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**
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 modules 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
	Total number of demonstration: 1			**0**	**1**
Examples	-	BSP	This example provides a description of how to use the different BSP drivers.	X	X
	ADC	ADC_AnalogWatchdog	This example provides a short description of how to use the ADC peripheral to perform conversions with analog watchdog and out-of-window interruptions enabled.	X	X
		ADC_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 external voltage on V_{INN} and V_{INP}.	X	X
		ADC_DualModeInterleaved	This example provides a short description of how to use two ADC peripherals to perform conversions in Interleaved dual-mode.	X	X
		ADC_InternalChannelConversion	This example describes how to configure and use the ADC to retreive 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_RegularConversion_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 to 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
	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

Level	Module Name	Project Name	Description	NUCLEO-H743ZI	STM32H743I-EVAL
Examples	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_UserDefinedPolynomial	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
	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 with swapping.	-	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
	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 sec 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 the 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 the DMA to transfer a word data buffer from Flash memory to embedded SRAM through the HAL API.	-	X

Level	Module Name	Project Name	Description	NUCLEO-H743ZI	STM32H743I-EVAL
Examples	DMA2D	DMA2D_BlendingWithAlphaInversion	This example provides a description of how to configure DMA2D peripheral in Memory_to_Memory with blending transfer and alpha inversion mode.	-	X
		DMA2D_MemToMemWithBlending	This example provides a description of how to configure DMA2D peripheral in Memory_to_Memory with blending transfer mode.	-	X
		DMA2D_MemToMemWithBlendingAndCLUT	This example provides a description of how to configure DMA2D peripheral in Memory_to_Memory with blending transfer mode and indexed 256 color images (L8).	-	X
		DMA2D_MemToMemWithPFCandRedBlueSwap	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_MemoryToMemory	This example provides a description of how to configure the DMA2D peripheral in Memory_to_Memory transfer mode.	-	X
		DMA2D_RegToMemWithLCD	This example provides a description of how to configure DMA2D peripheral in Register_to_Memory transfer mode and display the result on LCD.	-	X
	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 FDCAN units.	-	X
		FDCAN_Com_polling	This example shows how to achieve Polling Process Communication between two FDCAN units.	-	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 how to swap execution between bank1 and bank2 of the STM32H7xxxx 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

Level	Module Name	Project Name	Description	NUCLEO-H743ZI	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 to access the PC28F128M29EWLA NOR memory mounted on the STM32H743I-EVAL evaluation board.	-	X
		FMC_SDRAM	This example describes how to configure the FMC controller to access the SDRAM.	-	X
		FMC_SDRAM_DataMemory	This example describes how to configure the FMC controller to access the SDRAM including heap and stack.	-	X
		FMC_SDRAM_LowPower	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
	HRTIM	HRTIM_Arbitrary_Waveform	This example shows how to configure the HRTIM1 peripheral to generate arbitary 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 as source 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 an use the Fault event.	X	X
		HRTIM_MultiplePWM	This example shows how to configure the HRTIM1 peripheral to generate up to 5 PWM signals with different duty cycle for each HRTIM output.	X	X
		HRTIM_PWM_DifferentFrequencies	This example shows how to configure the HRTIM1 peripheral to generate up to 6 PWM signals with different timebase configuration for each slave timer.	X	X

Level	Module Name	Project Name	Description	NUCLEO-H743ZI	STM32H743I-EVAL
Examples	HSEM	HSEM_ProcessSync	This example shows how to use a hardware semaphore to synchronize two processes.	X	X
		HSEM_ReadLock	This example shows how to enable, take then release a 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.	-	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
	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_DecodingFromFLASH_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_DecodingUsingFs_DMA	This example 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 on the LCD-TFT screen.	-	X
		JPEG_DecodingUsingFs_Interrupt	This example demonstrates how to read jpeg file from SDCard 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_DecodingUsingFs_Polling	This example demonstrates how to read jpeg file from SDCard 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_EncodingFromFLASH_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 SDCard.	-	X
		JPEG_EncodingUsingFs_DMA	This example demonstrates how to read bmp file from SDCard memory using FatFs, encode it using the JPEG hardware encoder in DMA mode and save it on an SDCard.	-	X
		JPEG_MJPEG_VideoDecoding	This example demonstrates how to use the JPEG hardware decoder to decode an MJPEG video file and display it on the LCD-TFT screen.	-	X
		JPEG_MJPEG_VideoDecodingFromQSPI	This example demonstrates how to use the JPEG hardware decoder to decode an MJPEG video file located in the external QSPI Flash memory and display it on the LCD-TFT screen.	-	X

Level	Module Name	Project Name	Description	NUCLEO-H743ZI	STM32H743I-EVAL
Examples	LPTIM	LPTIM_Encoder	This example shows how to configure the LPTIM peripheral in Encoder mode.	X	X
		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 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 LTDC peripheral to display an RGB image of size 480x272 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
	MDMA	MDMA_DMA2D_Triggering	This example describes how to use the MDMA with its hardware trigger set to the DMA2D transfer complete flag.	-	X
		MDMA_GPDMA_Triggering	This example describes how to use the MDMA with its hardware trigger set to D2 Domain GP-DMA transfer complete flag.	-	X
		MDMA_LTDC_Triggering	This example describes how to use the MDMA with its 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 relaods the next transfer parameters and launches it without CPU intervention.	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 subimage (with a decimation factor of 1/2), and display the resulting red/green/blue decimated subimages on the LCD screen.	-	X
		MDMA_RepeatBlock_Rotation	This example describes 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 describes 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

Level	Module Name	Project Name	Description	NUCLEO-H743ZI	STM32H743I-EVAL
Examples	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_ExternalBias	This example shows how to configure the OPAMP peripheral in PGA mode with bias voltage for the Non-inverting mode.	X	X
	PWR	PWR_Domain3SystemControl	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 the system in Standby mode and wake up from this mode using external RESET or WKUP pin connected to the user button.	X	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 Wake-up Timer.	X	X
		PWR_STOP_DataRetain	This example shows how to retain data in D3SRAM when the system enters Stop mode while the D1 domain remains 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
	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_MemoryMappedDual	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 QUADSPI interface in Dual mode. It erases part of the QUADSPI 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 QUADSPI memory, and 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

Level	Module Name	Project Name	Description	NUCLEO-H743ZI	STM32H743I-EVAL
Examples	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
	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 in DMA circular mode and how to handle the buffer update.	-	X
		SAI_AudioPlayback	This example shows how to use the SAI to play back audio data coming from two microphones.	-	X
	SD	SD_ReadWrite_DMA	This example shows how to support UHS-I SDCard and achieve a frequency of 100 MHz.	-	X
		SD_ReadWrite_DMADoubleBuffer	This example shows how to support UHS-I SDCard and achieve a frequency of 100 MHz.	-	X
		SD_ReadWrite_DMA_HS	This example shows how to support UHS-I SDCard and achieve a frequency of 145 MHz.	-	X
		SD_ReadWrite_IT	This example shows how to support UHS-I SDCard and achieve a frequency of 100 MHz.	-	X
	SPI	SPI_FullDuplex_ComDMA	This example shows how to perform SPI data buffer transmission/reception between two boards via DMA.	X	-
		SPI_FullDuplex_ComIT	This example shows how to ensure SPI data buffer transmission/reception between two boards by using an interrupt.	X	-
		SPI_FullDuplex_ComPolling	This example shows how to ensure SPI data buffer transmission/reception in Polling mode between two boards.	X	-
Examples	TIM	TIM_6Steps	This example shows how to configure TIM1 to generate 6 steps.	X	X
		TIM_Asymetric	This example shows how to configure the TIMER peripheral to generate an asymmetric signal.	X	X
		TIM_Combined	This example shows how to configure TIM1 to generate 3 PWM combined signals with TIM1 channel5.	X	X
		TIM_ComplementarySignals	This example shows how to configure TIM1 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 (TIMx_CCR3).	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_InputCapture	This example shows how to use the TIMER peripheral to measure the frequency of an external signal.	X	X
		TIM_OCToggle	This example shows how to configure the TIMER peripheral to generate four different signals at 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 TIMER 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 TIMER peripheral to generate a timebase of one second with the corresponding interrupt request.	X	X

Level	Module Name	Project Name	Description	NUCLEO-H743ZI	STM32H743I-EVAL
Examples	UART	LPUART_WakeUpFromStop	This example shows how to configure a LPUART to wake up the microcontroller from Stop mode when a given stimulus is received.	-	X
		UART_HyperTerminal_DMA	This example describes a UART transmission (transmit/receive) in DMA mode between a board and an HyperTerminal PC application.	-	X
		UART_HyperTerminal_IT	This example describes a 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 that is sent by the UART on the HyperTerminal.	-	X
		UART_TwoBoards_ComDMA	This example describes a UART transmission (transmit/receive) in DMA mode between two boards.	X	-
		UART_TwoBoards_ComIT	This example describes a UART transmission (transmit/receive) in Interrupt mode between two boards.	X	-
		UART_TwoBoards_ComPolling	This example describes a UART transmission (transmit/receive) in Polling mode between two boards.	X	-
		UART_WakeUpFromStopUsingFIFO	This example shows how to use UART HAL API to wake up the microcontroller 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 the 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 update and simulate a software fault that generates a microcontroller WWDG reset when a predefined time period has elapsed.	X	X
	Total number of examples: 206			**75**	**131**
Applications	Audio	Audio_playback_and_record	This application shows how to use the different SAI ((serial audio interface) functionalities to perform audio record and playback via ST MEMS microphones (MP34DT01).	-	X
	Display	LTDC_AnimatedPictureFromSDCard	This application describes how to display an animated picture saved on a microSD card to the LCD screen.	-	X
		LTDC_Paint	This application describes how to configure the LCD touch screen, assign an action related to a configured touch zone, and save a BMP picture on a USB disk.	-	X
		LTDC_PicturesFromSDCard	This application describes how to display pictures saved on an SDCard to the LCD screen.	-	X
	EEPROM	EEPROM_Emulation	This application shows how to substitute standalone EEPROM by software by emulating the EEPROM mechanism using STM32H743xx on-chip Flash memory.	X	X
	ExtMem_CodeExecution	ExtMem_Application\FreeRTOS	This application shows how to create threads using CMSIS RTOS API,with execution from external memory.	-	X
		ExtMem_Application\LedToggling	This application provides a sample LED toogling program, with execution from external memory.	-	X
		ExtMem_Boot	This directory contains a set of source files and preconfigured projects that describes how to build an application for execution from external memory using the ExtMem_Boot firmware.	-	X
	FPU	FPU_Fractal	This application explains how to use the STM32H7xxxx 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

Level	Module Name	Project Name	Description	NUCLEO-H743ZI	STM32H743I-EVAL
Applications	FatFs	FatFS_uSD_Standalone	This application provides a description on how to use STM32Cube MCU Package 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_MultiDrives	This application provides a description on how to use STM32Cube MCU Package 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 multiple drives (USB disks and microSD).	-	X
		FatFs_RAMDisk	This application provides a description on how to use STM32Cube MCU Package with FatFs middleware component as a generic FAT file system module. The objective is to develop an application making the most of FatFs to configure an SRAM disk drive.	-	X
		FatFs_SDRAMDisk	This application provides a description on how to use STM32Cube MCU Package 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
		FatFs_USBDisk_RTOS	This application provides a description on how to use STM32Cube MCU Package 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. The objective is to develop an application making the most of the features offered by FatFs to configure a USB disk drive.	-	X
		FatFs_USBDisk_Standalone	This application provides a description on how to use STM32Cube MCU Package 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. The objective is to develop an application making the most of the features offered by FatFs to configure a USB disk drive.	-	X
		FatFs_uSD_DMA_RTOS	This application provides a description on how to use STM32Cube MCU Package 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 in RTOS mode .	-	X
		FatFs_uSD_DMA_Standalone	This application provides a description on how to use STM32Cube MCU Package 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 MCU Package 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
	FreeRTOS	FreeRTOS_MPU	This application aims at describing how to use the MPU FreeRTOS feature.	X	X
		FreeRTOS_SemaphoreFromISR	This application shows how to use a semaphore from ISR with CMSIS RTOS API .	X	X
		FreeRTOS_ThreadCreation	This application shows how to create a thread using CMSIS RTOS API.	X	X

Level	Module Name	Project Name	Description	NUCLEO-H743ZI	STM32H743I-EVAL
Applications	IAP	IAP_Binary_Template	This directory contains a set of source files that builds the application to be loaded into Flash memory using in-application programming (IAP) through USART.	-	X
		IAP_Main	This directory contains a set of source 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 guides STM32Cube HAL API users to run a http server application based on Netconn API of LwIP TCP/IP stack. It communicates with a web browser application hosted on a remote PC.	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. It communicates with a web browser application hosted on a remote PC.	-	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. It communicates with a web browser application hosted on a remote PC.	-	X
		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, open a command prompt window on the remote PC.	-	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, open a command prompt window on the remote PC.	-	X
		LwIP_TFTP_Server	This application guides STM32Cube HAL API users to run a tftp server application for STM32H7xxxx 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, open a command prompt window on the remote PC.	-	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, open a command prompt window on the remote PC.	-	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, open a command prompt window on the remote PC.	-	X

Level	Module Name	Project Name	Description	NUCLEO-H743ZI	STM32H743I-EVAL
Applications	STemWin	STemWin_HelloWorld	This directory contains a set of source files that implements a simple "Hello World" application based on STemWin for STM32H743xx devices.	-	X
		STemWin_SampleDemo	This directory contains a set of source files that implements a sample demonstration application allowing to demonstrate some of the STemWin library capabilities on STM32H743xx devices.	-	X
	USB_Device	Audio_Standalone	This application is a part of the USB device library package and uses the STM32Cube MCU Package. It describes how implement STM32H7xxxx audio streaming capability (the output is a speaker/headset).	-	X
		CDC_Standalone	This application is a part of the USB device library package and is based on the STM32Cube MCU Package. It describes how to use the USB device application based on the device communication class (CDC) and following the PSTN subprotocol. This application uses STM32H743xx USB OTG and UART peripherals.	-	X
		CustomHID_Standalone	This application is a part of the USB device library package and uses the STM32Cube MCU Package. It describes how to use the USB device application based on the custom HID class on STM32H743xx devices.	-	X
		DFU_Standalone	This application is a part of the USB device library package and uses the STM32Cube MCU Package. It describes how to use the USB device application based on the device firmware upgrade (DFU) on STM32H7xxxx devices.	-	X
		DualCore_Standalone	This application is a part of the USB device library package and uses the STM32Cube MCU Package. It describes how to use the USB device application based on STM32H7xxxx multicore feature. It integrates the device communication class (CDC) as well as human interface (HID) in the same project.	-	X
		HID_LPM_Standalone	The STM32H7xxxx devices support the USB Link Power Management Protocol (LPM-L1) in compliance with USB 2.0 LPM-L1 ECN. hpcd.Init.lpm_enable in usbd_conf.c must be set to 1 to enable the support for LPM-L1 protocol in the USB stack.	-	X
		HID_Standalone	This application is a part of the USB Device Library package and uses the STM32Cube MCU Package. It describes how to use the USB device application based on the human interface (HID) on STM32H743xx devices.	-	X
		MSC_Standalone	This application is a part of the USB Device Library package and uses the STM32Cube MCU Package. It describes how to use the USB device application based on the mass storage class (MSC) on STM32H7xxxx devices.	-	X

Level	Module Name	Project Name	Description	NUCLEO-H743ZI	STM32H743I-EVAL
Applications	USB_Host	AUDIO_Standalone	This application is a part of the USB host library package and uses the STM32Cube MCU Package. It describes how to use USB host application based on the audio OUT class on STM32H7xxxx devices.	-	X
		CDC_Standalone	This application is a part of the USB host library package and uses the STM32Cube MCU Package. It describes how to use USB host application based on the communication class (CDC) on STM32H7xxxx devices.	-	X
		DualCore_Standalone	This application is a part of the USB host library package and uses STM32Cube MCU Package. It describes how to use USB host application based on the STM32H7xxxx multicore support feature. It integrates 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 and uses the STM32Cube MCU Package. It describes how to dynamically switch, on the same port, between available USB host applications, on STM32H7xxxx devices.	-	X
		FWupgrade_Standalone	This application is a part of the USB host library package and uses the STM32Cube MCU Package. It describes how to use the USB host application based on the in-application programming (IAP) on STM32H7xxxx devices.	-	X
		HID_RTOS	This application is a part of the USB host library package and uses the STM32Cube MCU Package. It describes how to use the USB host application based on the human interface class (HID) on STM32H7xxxx devices.	-	X
		HID_Standalone	This application is a part of the USB host library package and uses the STM32Cube MCU Package. It describes how to use the USB host application based on the human interface class (HID) on STM32H7xxxx devices.	-	X
		MSC_RTOS	This application is a part of the USB host library package and uses the STM32Cube MCU Package. It describes how to use the USB host application based on the mass storage class (MSC) on STM32H7xxxx devices.	-	X
		MSC_Standalone	This application is a part of the USB host library package and uses the STM32Cube MCU Package. It describes how to use the USB host application based on the mass storage class (MSC) on 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: 61			**6**	**54**
Demonstrations	-	Demo	The demonstration firmware is based on STM32Cube. It helps you discover the STM32 Arm Cortex-M core devices that can be plugged on a NUCLEO-H743ZI board.	X	-
	Total number of demonstrations: 1			**1**	**0**
Total number of projects: 270				**83**	**187**

Revision history

Table 2. **Document revision history**

Date	Version	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.
27-Jun-2018	4	Added Arm wordmark notice in Section 1 Reference documents. Replaced STM32Cube embedded firmware by STM32Cube MCU Package in the whole document. Updated Table 1. STM32CubeH7 firmware examples.

Contents