STM32CubeMP1 Package release note - v1.1.0

STM32CubeMP1 Package release note - v1.1.0

Stable: 18.12.2019 - 14:34 / Revision: 18.12.2019 - 14:28

Contents

1 Release information

This article aims to describe the content of the **software** release included in the STM32CubeMP1 Package, version **v1.1.0**

2 Delivery scope and purpose

The **STM32CubeMP1 Package** is a software package running on **Arm® Cortex® M4** processors and is a fundamental part of the STM32MPU Embedded Software distribution.

This release includes:

- The STM32Cube HAL, STM32 abstraction layer embedded software ensuring maximized portability across the STM32 portfolio. HAL APIs are available for all peripherals.
- Low-layer APIs (LL) offering a fast lightweight expert-oriented layer that is closer to the hardware than the HAL . LL APIs are available only for a set of peripherals.
- A consistent set of middleware components such as FreeRTOS and OpenAMP.
- All embedded software utilities delivered with a full set of examples.

It also includes:

- BSP for the STM32MP157C-EV1 Evaluation board and the STM32MP157C-DK2 Discovery kit (button and LEDs only)
- Multi-core components
  - OpenAMP
  - ResourceManager
  - CoproSync
- A new ST Eclipse is introduced with this release : **STM32CubeIDE**.
  SW4STM32 can be still used, but the reference is now STM32CubeIDE. In case of support, ST will focus and recommand STM32CubeIDE tool.
- IDE ready projects (examples, applications and demonstration firmware) available through the System Workbench for STM32 toolchain -SW4STM32 (inherited from the first release), but the SW4STM32 projects can be imported inside STM32CubeIDE.
  - EWARM (version 8.32.3 and laters) and MDK-ARM(Pack Keil.STM32MP1xx_DFP.1.1.0.pack + MDK-ARM 5.27 and laters) are now compliant with STM32MP15.

3 Supported devices

The drivers provided within this package support the following devices :

- STM32MP157Cxx, STM32MP157Axx
- STM32MP153Cxx, STM32MP153Axx
- STM32MP151Cxx, STM32MP151Axx

4 Supported hardware

This software delivery is applicable to the following boards:

- STM32MP157C-EV1 Evaluation board (RevC). For information about this board, read the article STM32MP157C-EV1 - hardware description.
- STM32MP157C-DK2 Discovery board (RevC). For information about this board, read the article STM32MP157X-DKX - hardware description.

5 Known issues and limitations

- TIM Break source from DFSDM badly defined in header files - see patch upper V1.1.0 in github [1]
- **OpenAMP compilation issues with EWARM and MDK-ARM when code generated through CubeMx** :
  - OpenAMP Middleware update - patch is available in v1.1.1 (upper V1.1.0 in github)([2]

- Linker Template to update for IAR and KEIL: new OpenAMP section is added but commented (if OpenAMP used, the linker script should be updated to uncomment this section) - patch is available in v1.1.1 (upper V1.1.0 in github)[3]
  - Limitation with v1.1.1:
    - To avoid compiling errors in OpenAMP when compiling in MDK-ARM IDE, you have to disable the « Use MicroLIB » in « Target » tab.
    - To avoid linking errors in OpenAMP when compiling in EWARM.
      In the linker files (.icf), you have to uncomment those 4 lines :

```
/* Create region for OPENAMP */
/* define symbol __OPENAMP_region_start__ = 0x10040000; */
/* define symbol __OPENAMP_region_size__   = 0x8000;    */
/* export symbol __OPENAMP_region_start__;              */
/* export symbol __OPENAMP_region_size__;               */
```

    - To avoid linking errors in OpenAMP when compiling in MDK-ARM IDE.
      In the linker files (.sct), you have to uncomment those 4 lines :

```
; ***   Create region for OPENAMP                        ***
;  .resource_table +0 ALIGN 4 {         ; resource table
;    *(.resource_table)
;  }
;  __OpenAMP_SHMEM__ 0x10040000 EMPTY 0x8000 {} ; Shared Memory area used by OpenAMP
```

- **Projects imported inside STM32CubeIDE** : fix .project files to be aligned with official STM32CubeIDE - see patch upper V1.1.0 in github [4]
- **SystemClock generated by STM32CubeMx**:
  By default, the STM32CubeMx generated code is **compliant with the Engineering mode**.
  Therefore, the call of the following **clocks functions** (SystemClock_Config(), PeriphCommonClock_Config(), HAL_RCCEx_PeriphCLKConfig()) will have **to be removed in Production mode** since these clocks are managed by linux in this case.
  System part (SystemClock_Config() and PeriphCommonClock_Config()) can be removed in STM32CubeMX by selecting **"Not Generate function call** *for RCC in the Project Manager tab and then Advanced Settings tab.*
  PeriphCLKConfig *{{Highlight |should be removed manually} from stm32mp1xx_hal_msp.c (STM32CubeMX generated code) .*
  To make your code compatible with both engineering and production mode above RCC functions can be put under dynamic condition ***if(IS_ENGINEERING_BOOT_MODE())***

6 Contents

Note:

- For detailed information, read file **Release_Notes.html** delivered with the **STM32CubeMP1** package software.

6.1 Released projects

The **STM32CubeMP1 Firmware Package** comes with a rich set of examples running on **STMicroelectronics** boards, organized by board and provided with preconfigured projects for the main supported toolchain (**System Workbench for STM32**). The exhaustive list of projects is provided in the table List of Projects.

- **IDE ready projects**

Projects	STM32MP157C-DK2	STM32MP157C-EV1
Availables Firmware	27	28

6.2 Released components

- **Drivers**

Component	Version	Notes
Cortex-M CMSIS	V5.4.0	
STM32MP1xx CMSIS	V1.1.0	
STM32MP1xx HAL	V1.1.0	
BSP STM32MP15xx_EVAL	V1.1.0	
BSP STM32MP15xx_DISCO	V1.1.0	

- **Middleware**

Component	Version	Notes
FreeRTOS	V10.0.1 ST modified 20190329	
OpenAMP	v2018.10 ST modified 20190329 and ST interface 20190729	

- **Utilities**

Component	Version	Notes
Resourcemanager	V1.9.0	

6.3 Available drivers

Please find below the list of HAL and LL drivers available:

6.3.1 HAL drivers

STM32CubeMP1 HAL Driver items	Description
ADC	* This driver provides firmware functions to manage the following * functionalities of the Analog to Digital Convertor (ADC) * peripheral: * + Initialization and de-initialization functions * ++ Initialization and Configuration of ADC * + Operation functions * ++ Start, stop, get result of conversions of regular * group, using 3 possible modes: polling, interruption or DMA. * + Control functions * ++ Channels configuration on regular group * ++ Analog Watchdog configuration * + State functions * ++ ADC state machine management * ++ Interrupts and flags management * + Operation functions * ++ Start, stop, get result of conversions of ADC group injected, * using 2 possible modes: polling, interruption. * ++ Calibration * +++ ADC automatic self-calibration * +++ Calibration factors get or set * ++ Multimode feature when available * + Control functions * ++ Channels configuration on ADC group injected * + State functions * ++ ADC group injected contexts queue management
	* This driver provides firmware functions to manage the following * functionalities of the High Definition Multimedia Interface

STM32CubeMP1 HAL Driver items	Description
CEC	* Consumer Electronics Control Peripheral (CEC). * + Initialization and de-initialization function * + IO operation function * + Peripheral Control function
CRC	* This driver provides firmware functions to manage the following * functionalities of the Cyclic Redundancy Check (CRC) peripheral: * + Initialization and de-initialization functions * + Peripheral Control functions * + Peripheral State functions * + Extended features functions
CORTEX	* This driver provides firmware functions to manage the following * functionalities of the CORTEX (MPU, Cache, …): * + Initialization and de-initialization functions * + Peripheral Control functions
CRYP	* This driver provides firmware functions to manage the following * functionalities of the Cryptography (CRYP) peripheral: * + Initialization and de-initialization functions * + AES processing functions * + DES processing functions * + TDES processing functions * + DMA callback functions * + CRYP IRQ handler management * + Peripheral State functions * + Extended AES processing functions

STM32CubeMP1 HAL Driver items	Description
DAC	* This driver provides firmware functions to manage the following * functionalities of the Digital to Analog Converter (DAC) peripheral: * + Initialization and de-initialization functions * + IO operation functions * + Peripheral Control functions * + Peripheral State and Errors functions * + Extended features functions
DCMI	* This driver provides firmware functions to manage the following * functionalities of the Digital Camera Interface (DCMI) peripheral: * + Initialization and de-initialization functions * + IO operation functions * + Peripheral Control functions * + Peripheral State and Error functions
DFSDM	* This driver provides firmware functions to manage the following * functionalities of the Digital Filter for Sigma-Delta Modulators * (DFSDM) peripherals: * + Initialization and configuration of channels and filters * + Regular channels configuration * + Injected channels configuration * + Regular/Injected Channels DMA Configuration * + Interrupts and flags management * + Analog watchdog feature * + Short-circuit detector feature * + Extremes detector feature * + Clock absence detector feature * + Break generation on analog watchdog or short-circuit event * + Set and get pulses skipping on channel.

STM32CubeMP1 HAL Driver items	Description
DMA	* This driver provides firmware functions to manage the following * functionalities of the Direct Memory Access (DMA) peripheral: * + Initialization and de-initialization functions * + IO operation functions * + Peripheral State and errors functions * + Extended features functions
EXTI	* This driver provides firmware functions to manage the following * functionalities of the General Purpose Input /Output (EXTI) peripheral: * + Initialization and de-initialization functions * + IO operation functions
FDCAN	* This driver provides firmware functions to manage the following * functionalities of the Flexible DataRate Controller Area Network * (FDCAN) peripheral: * + Initialization and de-initialization functions * + IO operation functions * + Peripheral Configuration and Control functions * + Peripheral State and Error functions
GPIO	* This driver provides firmware functions to manage the following * functionalities of the General Purpose Input /Output (GPIO) peripheral: * + Initialization and de-initialization functions * + IO operation functions * + Extended Peripheral Control functions

STM32CubeMP1 HAL Driver items	Description
HAL	* This driver provides firmware functions to manage the following * functionalities of HAL, Tick, SYSCFG, DBGMCU: * + Initialization and de-initialization functions * + HAL Initialization and de-initialization functions * + Configure the source of the time base * + HAL Control functions * + Tick management (get/set/inc/priority /suspend/resume) * + Get HAL revision, the device revision identifier, the device identifier * + Enable/Disable DBG wake up on AIEC * + Enable/Disable the Debug Module during Domain1 SLEEP mode * + Enable/Disable the Debug Module during Domain1 STOP mode * + Enable/Disable the Debug Module during Domain1 STANDBY mode * + Configure the internal voltage reference buffer voltage scale * + Configure the internal voltage reference buffer high impedance mode * + Tune the Internal Voltage Reference buffer (VREFBUF) * + Enable/Disable the Internal Voltage Reference buffer (VREFBUF) * + Ethernet PHY Interface Selection either MII or RMII * + Analog Switch control for dual analog pads * + Enable/Disable the booster to reduce the total harmonic distortion of the analog * + Enable/Power-down the I/O Compensation Cell * + To Enable/Disable optimize the I/O speed when the product voltage is low * + Code selection for the I/O Compensation cell
	* This driver provides firmware functions to manage the following * functionalities of the HASH peripheral: * + Initialization and de-initialization methods * + HASH or HMAC processing in polling mode * + HASH or HMAC processing in interrupt mode

STM32CubeMP1 HAL Driver items	Description
HASH	* + HASH or HMAC processing in DMA mode * + Peripheral State methods * + HASH or HMAC processing suspension /resumption * Additionally, this driver provides functions to manage HMAC * multi-buffer DMA-based processing for MD-5, SHA-1, SHA-224 * and SHA-256.
HSEM	* This driver provides firmware functions to manage the following * functionalities of the semaphore peripheral: * + Semaphore Take function (2-Step Procedure) , non blocking * + Semaphore FastTake function (1-Step Procedure) , non blocking * + Semaphore Status check * + Semaphore Clear Key Set and Get * + Release and release all functions * + Semaphore notification enabling and disabling and callnack functions * + IRQ handler management
I2C	* This driver provides firmware functions to manage the following * functionalities of the Inter Integrated Circuit (I2C) peripheral: * + Initialization and de-initialization functions * + IO operation functions * + Peripheral State and Errors functions * + Extended features functions
IPCC	* This driver provides firmware functions to manage the following * functionalities of the Inter-Processor communication controller * peripherals (IPCC). * + Initialization and de-initialization functions * + Configuration, notification and interrupts handling * + Peripheral State and Error functions

L Driver items	Description
LPTIM	* This driver provides firmware functions to manage the following * functionalities of the Low Power Timer (LPTIM) peripheral: * + Initialization and de-initialization functions. * + Start/Stop operation functions in polling mode. * + Start/Stop operation functions in interrupt mode. * + Reading operation functions. * + Peripheral State functions.
MDIOS	* This driver provides firmware functions to manage the following * functionalities of the MDIOS Peripheral. * + Initialization and de-initialization functions * + IO operation functions * + Peripheral Control functions
MDMA	* This driver provides firmware functions to manage the following * functionalities of the Master Direct Memory Access (MDMA) peripheral: * + Initialization/de-initialization functions * + I/O operation functions * + Peripheral State and errors functions
PWR	* This driver provides firmware functions to manage the following * functionalities of the Power Controller (PWR) peripheral: * + Initialization and de-initialization functions * + Peripheral Control functions * + Peripheral Extended features functions

STM32CubeMP1 HAL Driver items	Description
QUADSPI	* This driver provides firmware functions to manage the following * functionalities of the QuadSPI interface (QSPI). * + Initialization and de-initialization functions * + Indirect functional mode management * + Memory-mapped functional mode management * + Auto-polling functional mode management * + Interrupts and flags management * + MDMA channel configuration for indirect functional mode * + Errors management and abort functionality
RCC	* This driver provides firmware functions to manage the following * functionalities of the Reset and Clock Control (RCC) peripheral: * + Initialization and de-initialization functions * + Peripheral Control functions * + Extended Peripheral Control functions
RNG	* This driver provides firmware functions to manage the following * functionalities of the Random Number Generator (RNG) peripheral: * + Initialization and configuration functions * + Peripheral Control functions * + Peripheral State functions
SAI	* This driver provides firmware functions to manage the following * functionalities of the Serial Audio Interface (SAI) peripheral: * + Initialization/de-initialization functions * + I/O operation functions * + Peripheral Control functions * + Peripheral State functions * + Modify PDM microphone delays.

L Driver Items	
SD	* This driver provides firmware functions to manage the following * functionalities of the Secure Digital (SD) peripheral: * + Initialization and de-initialization functions * + IO operation functions * + Peripheral Control functions * + Peripheral State functions * + Extended features functions
SMBUS	* This driver provides firmware functions to manage the following * functionalities of the System Management Bus (SMBus) peripheral, * based on I2C principles of operation : * + Initialization and de-initialization functions * + IO operation functions * + Peripheral State and Errors functions
SPDIFRX	* This driver provides firmware functions to manage the following * functionalities of the SPDIFRX audio interface: * + Initialization and Configuration * + Data transfers functions * + DMA transfers management * + Interrupts and flags management
SPI	* This driver provides firmware functions to manage the following * functionalities of the Serial Peripheral Interface (SPI) peripheral: * + Initialization and de-initialization functions * + IO operation functions * + Peripheral Control functions * + Peripheral State functions * + IO operation functions * + Peripheral Control functions

STM32CubeMP1 HAL Driver items	Description
TIMER	* This driver provides firmware functions to manage the following * functionalities of the Timer (TIM) peripheral: * + Time Base Initialization * + Time Base Start * + Time Base Start Interruption * + Time Base Start DMA * + Time Output Compare/PWM Initialization * + Time Output Compare/PWM Channel Configuration * + Time Output Compare/PWM Start * + Time Output Compare/PWM Start Interruption * + Time Output Compare/PWM Start DMA * + Time Input Capture Initialization * + Time Input Capture Channel Configuration * + Time Input Capture Start * + Time Input Capture Start Interruption * + Time Input Capture Start DMA * + Time One Pulse Initialization * + Time One Pulse Channel Configuration * + Time One Pulse Start * + Time Encoder Interface Initialization * + Time Encoder Interface Start * + Time Encoder Interface Start Interruption * + Time Encoder Interface Start DMA * + Commutation Event configuration with Interruption and DMA * + Time OCRef clear configuration * + Time External Clock configuration * + Time Hall Sensor Interface Initialization * + Time Hall Sensor Interface Start * + Time Complementary signal bread and dead time configuration * + Time Master and Slave synchronization configuration * + Time Output Compare/PWM Channel Configuration (for channels 5 and 6) * + Time OCRef clear configuration * + Timer remapping capabilities configuration
	* This driver provides firmware functions to manage the following * functionalities of the Universal (Synchronous) Asynchronous Receiver Transmitter * Peripheral (U(S)ART). * + Initialization and de-initialization

STM32CubeMP1 HAL Driver items	Description
U(S)ART	functions * + IO operation functions * + Peripheral Control functions * + Peripheral State and Error functions * + Peripheral Control functions
WWDG	* This driver provides firmware functions to manage the following * functionalities of the Window Watchdog (WWDG) peripheral: * + Initialization and Configuration functions * + IO operation functions

6.3.2 LL drivers

(Legend : LL drivers added since ecosystem release v1.1.0)

STM32CubeMP1 LL Driver items	Description
ADC	* This driver provides firmware functions to manage the following * functionalities of the ADC peripheral: * + Initialization/de-initialization functions * + Configuration functions (ADC instance, group regular, group injected, * channels, analog watchdog, oversampling, multimode) * + IT/FLAGS management functions
BUS	* This driver provides firmware functions to manage the following * functionalities of the BUS peripheral: * + Enable/disable/reset clocks for all system BUS (AHB2, AHB3, AHB4, AHB5, AHB6, * AXI, MLAHB, APB1, APB2, APB3, APB4, APB5)

STM32CubeMP1 LL Driver items	Description
DMA	* This driver provides firmware functions to manage the following * functionalities of the DMA peripheral: * + Initialization/de-initialization functions * + Configuration functions * + IT/FLAGS management functions
DMAMUX	* This driver provides firmware functions to manage the following * functionalities of the DMAMUX peripheral: * + Initialization/de-initialization functions * + IT/FLAGS management functions
EXTI	* This driver provides firmware functions to manage the following * functionalities of the EXTI peripheral: * + Initialization/de-initialization functions * + IT/FLAGS/Trigger management functions * + Configuration functions
GPIO	* This driver provides firmware functions to manage the following * functionalities of the GPIO peripheral: * + Initialization/de-initialization functions * + Data access functions * + Port configuration functions
HSEM	* This driver provides firmware functions to manage the following * functionalities of the HSEM peripheral: * + IT/FLAGS management functions * +Data management functions

STM32CubeMP1 LL Driver items	Description
I2C	* This driver provides firmware functions to manage the following * functionalities of the I2C peripheral: * + Initialization/de-initialization functions * + IT/FLAGS management functions * +Data management functions * + Configuration functions
IPCC	* This driver provides firmware functions to manage the following * functionalities of the IPCC peripheral: * + IT/FLAGS management functions * + Enable/disable transmit and receive channels functions
LPTIM	* This driver provides firmware functions to manage the following * functionalities of the LPTIM peripheral: * + Initialization/de-initialization functions * + IT/FLAGS management functions * + Configuration (Trigger / Clock / Encoder / LPTIM) functions
PWR	* This driver provides firmware functions to manage the following * functionalities of the PWR peripheral: * + Initialization/de-initialization functions * + FLAGS management functions * + Configuration functions
	* This driver provides firmware functions to manage the following * functionalities of the RCC peripheral: * + Clocks management functions (HSE/HSI/CSI/

STM32CubeMP1 LL Driver items	Description
RCC	LSE/LSI/MCO/PLL) * + RTC/TIMERS functions * + IT/FLAGS management functions * + De-initialization functions * + Get system and peripherals clocks frequency functions
SPI	* This driver provides firmware functions to manage the following * functionalities of the SPI peripheral: * + Initialization/de-initialization functions * + IT/FLAGS management functions * + Data / DMA management functions * + Configuration functions
SYSTEM	* This driver provides firmware functions to manage the following * functionalities of the SYSTEM peripheral: * + SYSCFG and DBGMCU functions
TIM	* This driver provides firmware functions to manage the following * functionalities of the TIM peripheral: * + Initialization/de-initialization functions * + Configuration functions (Time base, Capture Compare, Output and Input Channel, * Timer Synchro, Break, DMA Burst Mode) * + Counter clock selection functions * + Timer input remapping functions * + IT/FLAGS management functions * + DMA management functions * + Event management functions
	* This driver provides firmware functions to manage the following * functionalities of the USART peripheral: * + Initialization/de-initialization functions * + Configuration functions (Irda,

STM32CubeMP1 LL Driver items	Description
USART	Smartcard, Half duplex, * SPI Slave, LIN, Driver enable) * + Advanced configurations services functions * + IT/FLAGS management functions * + DMA management functions * + Data management functions * + Execution functions
UTILS	* This driver provides firmware functions to manage the following * functionalities of the UTILS peripheral: * + Device Electronic Signature functions * + DELAY functions * + SYSTEM functions
WWDG	* This driver provides firmware functions to manage the following * functionalities of the WWDG peripheral: * + Enable / Disable functions * + Configuration functions * + IT/FLAGS management functions

6.4 Available projects

Please find below the list of projects available for STM32MP157C-EV1 and STM32MP157C-DK2 :

Note:

- Please refer to article Introduction to boot mode to get more information about **Production** and **Engineering** modes

The preferred supported toolchain for STM32MP15 is now **STM32CubeIDE** (All-in-one multi-OS development tool).
All the projects are available with System Workbench for STM32 (inherited from first release), but the projects can be imported inside STM32CubeIDE.
Moreover, other toolchains are now compliant with STM32MP15 :

- IAR Embedded Workbench for ARM (EWARM) toolchain,
- RealView Microcontroller Development Kit (MDK-ARM) toolchain,

By default, all the examples are available with SW4STM32. The migration on other toolchain is not yet finished, but some of them are available :

- Legend:
  - (*) : list of STM32MP157C-EV1 examples available with 4 IDEs (SW4STM32, IAR, KEIL and STM32CubeIDE).
  - (**) : GPIO_EXTI project is ported on SW4STM32 and STM32CubeIDE for STM32MP157C-EV1 and STM32MP157C-DK2.
  - Project Name : New project added since ecosystem release v1.1.0

Level Examples	Module Name	Project Name	Description	STM32MP157C-DK2	STM32MP157C-EV1	Mode	Core
	ADC	ADC_SingleConversion_TriggerTimer_DMA	Use ADC to convert a single channel at each trig from timer, conversion data are transferred by DMA into an array, indefinitely (circular mode).	x	x	Production & Engineering	Cortex-M4
	CRC	CRC_UserDefinedPolynomial	How to configure the CRC using the HAL API. The CRC (cyclic redundancy check) calculation unit computes the 8-bit CRC code for a given buffer of 32-bit data words, based on a user-defined generating polynomial.	x	x	Production & Engineering	Cortex-M4
	CRYP	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	Production & Engineering	Cortex-M4
							Cortex-

Level Examples	Module Name	Project Name	Description	STM32MP157C-DK2	STM32MP157C-EV1	Mode	Core
	CORTEX	CORTEXM_MPU	Presentation of the MPU feature. This example configures a memory area as privileged read-only, and attempts to perform read and write operations in different modes.	x	x	Production & Engineering	M4
	DAC	DAC_SimpleConversion	How to use the DAC peripheral to do a simple conversion.	-	x	Production & Engineering	Cortex-M4
	DMA	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	x	Production & Engineering	Cortex-M4
	FDCAN	FDCAN_Loopback	How to configure the FDCAN to operate in loopback mode.	x	x	Engineering Only	Cortex-M4
(**)	GPIO	GPIO_EXTI	How to configure external interrupt lines.	x	x	Production & Engineering	Cortex-M4
							Co

Level Examples	Module Name	Project Name	Description	STM32MP157C-DK2	STM32MP157C-EV1	Mode	Core
	HASH	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	Production & Engineering	rtex-M4
	I2C	I2C_TwoBoards_ComDMA	How to handle I2C data buffer transmission /reception between two boards, via DMA.	-	x	Production & Engineering	Cortex-M4
		I2C_TwoBoards_ComIT	How to handle I2C data buffer transmission /reception between two boards, using an interrupt.	x	x	Production & Engineering	Cortex-M4
	LPTIM	LPTIM_PulseCounter	This example describes how to configure and use LPTIM to count pulses through the LPTIM HAL API.	x	x	Production (DK2) & Engineering (EV1 and DK2)	Cortex-M4
	PWR	PWR_STOP_CoPro	How to enter the CSTOP and STOP modes using CM4 core (aka coprocessor) and wake up from this mode by using external wakeup interrupt.	x	x	Production Only	Cortex-M4
							C

Level Examples	Module Name	Project Name	Description	STM32MP157C-DK2	STM32MP157C-EV1	Mode	Core
	QSPI	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	Production & Engineering	cortex-M4
	SPI	SPI_FullDuplex_ComDMA_Master	Data buffer transmission/reception between two boards via SPI in Polling.	x	x	Production & Engineering	Cortex-M4
		SPI_FullDuplex_ComDMA_Slave	Data buffer transmission/reception between two boards via SPI using DMA.	x	x	Production & Engineering	Cortex-M4
		SPI_FullDuplex_ComIT_Master	Data buffer transmission/reception between two boards via SPI using Interrupt mode.	x	-	Production & Engineering	Cortex-M4
		SPI_FullDuplex_ComIT_Slave	Data buffer transmission/reception between two boards via SPI using Interrupt mode.	x	-	Production & Engineering	Cortex-M4

Level Examples	Module Name	Project Name	Description	STM32MP157C-DK2	STM32MP157C-EV1	Mode	Core
	TIM	TIM_DMABurst	This example shows how to update the TIMER TIM2_CH4 period and the duty cycle using the TIMER DMA burst feature.	x	x	Production & Engineering	Cortex-M4
	UART	UART_TwoBoards_ComDMA	UART transmission (transmit/receive) in DMA mode between two boards.	x	-	Production & Engineering	Cortex-M4
		UART_TwoBoards_ComIT	UART transmission (transmit/receive) in Interrupt mode between two boards.	x	x	Production & Engineering	Cortex-M4
		UART_Receive_Transmit_Console	UART transmission (printf/getchar) via console with user interaction.	x	x	Production & Engineering	Cortex-M4
							Cortex-M

Level Examples	Module Name	Project Name	Description	STM32MP157C-DK2	STM32MP157C-EV1	Mode	Core
	WWDG	WWDG_Example	Configuration of the HAL API to periodically update the WWDG counter and simulate a software fault that generates an MCU WWDG reset when a predefined time period has elapsed.	x	x	Production & Engineering	4
Applications	Total number of examples: 40			20	20		
	CoproSync	CoproSync_ShutDown	Send of shutdown information to Cortex-M4 so that it is able to take necessary actions before going to reset state.	x	x	Production	Cortex-M4
(*)	FreeRTOS	FreeRTOS_ThreadCreation	How to implement thread creation using CMSIS RTOS API.	x	x	Production & Engineering	Cortex-M4
	OpenAMP	OpenAMP_Dynamic_ResMgr	How to use OpenAMP MW + Virtual UART to create an Inter-Processor Communication channel seen as TTY device in Linux OS.	-	x	Production	Cortex-M4
							C

Level Examples	Module Name	Project Name	Description	STM32MP157C-DK2	STM32MP157C-EV1	Mode	Core
		OpenAMP_TTY_echo	How to use OpenAMP MW + Virtual UART to create an Inter-Processor Communication channel seen as TTY device in Linux OS	x	x	Production	cortex-M4
		OpenAMP_TTY_echo_wakeup	How to use OpenAMP MW to enter in different power system operating mode (Run, Stop and Standby).	x	x	Production	Cortex-M4
(*)		OpenAMP_raw	How to use OpenAMP MW to create an Inter-Processor Communication channel	x	x	Production	Cortex-M4
Demonstrations	Total number of applications: 11			5	6		
							Cortex-

Level Examples	Module Name	Project Name	Description	STM32MP157C-DK2	STM32MP157C-EV1	Mode	Core
	AI	AI_Character_Recognition	This project demonstrate a complex application that is running on both CPU1(CA7) and CPU2 (CM4)	x	x	Production	M4
Templates	Total number of demonstrations: 2			1	1		
(*)	-	Starter project	This projects provides a reference template that can be used to build any firmware application in mode on Cortex-M4	x	x	Engineering	Cortex-M4
Total number of projects: 55	Total number of templates: 2			1 27	1 28		

7 Minor release updates

STMicroelectronics regularly delivers corrections through github® components. You can decide to incorporate them into your developer package or distribution package.

- Please refer to STM32MP1 Developer Package or How to switch to github® mode in distribution package.

7.1 STM32CubeMP1 Package v1.1.1

STM32CubeMP1 updates github release 1.1.1

8 How to get started with STM32CubeMP1 Package

- Please refer to How to get software and start with this release

9 Associated tools

- Please refer to Referenced tools release notes to obtain more information on all available tools.

10 References

Hardware Abstraction Layer

Low layer of STM32Cube

Board support package

(Software)Integrated development/design/debugging environment

Digital Filter for Sigma-Delta Modulator

Reset and Clock Control

Cortex Microcontroller Software Interface Standard

Evaluation board

Discovery kit

Analog-to-digital converter. The process of converting a sampled analog signal to a digital code that represents the amplitude of the original signal sample.

Direct Memory Access

Consumer Electronics Control (HDMI standard)

input/output

Cyclic redundancy check calculation unit

Microprocessor Unit

Cryptographic processor

Advanced Encryption Standard

Data Encryption Standard

Triple Data Encryption Standard

Digital-to-analog converter (Electronic circuit that converts a binary number into a continuously varying value.)

Digital Camera Memory Interface

External Interrupt

General-Purpose Input/Output (A realization of open ended transmission between devices on an embedded level. These pins available on a processor can be programmed to be used to either accept input or provide output to external devices depending on user desires and applications requirements.)

System Configuration

voltage reference buffer (STM32 specific)

Hash-based Message Authentication Code

Secure Hash Algorithm

Hardware Semaphore

Inter-Integrated Circuit (Bi-directional 2-wire bus standard for efficient inter-IC control.)

Inter-Processor Communication Controller

low-power timer (STM32 specific)

Random Number Generator

Serial Audio Interface (Mechanism used to transfer non-buffered audio data between processors and/or audio converters.)

Secure digital

System Management Bus

Serial Peripheral Interface

Pulse Width Modulation

Android Runtime (see https://source.android.com/devices/tech/dalvik)

High Speed External oscillator (STM32 clock source)

High Speed Internal oscillator (STM32 clock source) or High Speed Synchronous Serial Interface (MIPI® Alliance standard)

Multi Speed Internal oscillator (STM32 clock source)

Low Speed External oscillator (STM32 clock source)

Low Speed Internal oscillator (STM32 clock source)

Real Time Clock

Universal Synchronous/Asynchronous Receiver/Transmitter

Local Interconnect Network

Operating System

Application programming interface

Flash memories combine high density and cost effectiveness of EPROMs with the electrical erasability of EEPROMs. For this reason, the Flash memory market is one of the most exciting areas of the semiconductor industry today and new applications requiring in system reprogramming, such as cellular telephones, automotive engine management systems, hard disk drives, PC BIOS software for Plug & Play, digital TV, set top boxes, fax and other modems, PC cards and multimedia CD-ROMs, offer the prospect of very high volume demand.

also known as

Universal Asynchronous Receiver/Transmitter

Microcontroller Unit (MCUs have internal flash memory and are intended to operate with a minimum amount of external support ICs. They commonly are a self-contained, system-on-chip (SoC) designs.)

Real Time Operating System

TeleTYpewriter

Artificial Intelligence