



How to make STM32MP1 configuration easy thanks to ST ecosystem

Bossen WU



STM32 performance continuum

Microprocessors

Dual Cortex-A7 @ 650MHz (2470 DMIPS)
Cortex-M4 @ 209MHz (261 DMIPS)



Microcontrollers



Cortex-M3 @ 120MHz (150 DMIPS)



Cortex-M4 @ 180MHz (225 DMIPS)



Cortex-M7 @ 216MHz (462 DMIPS)



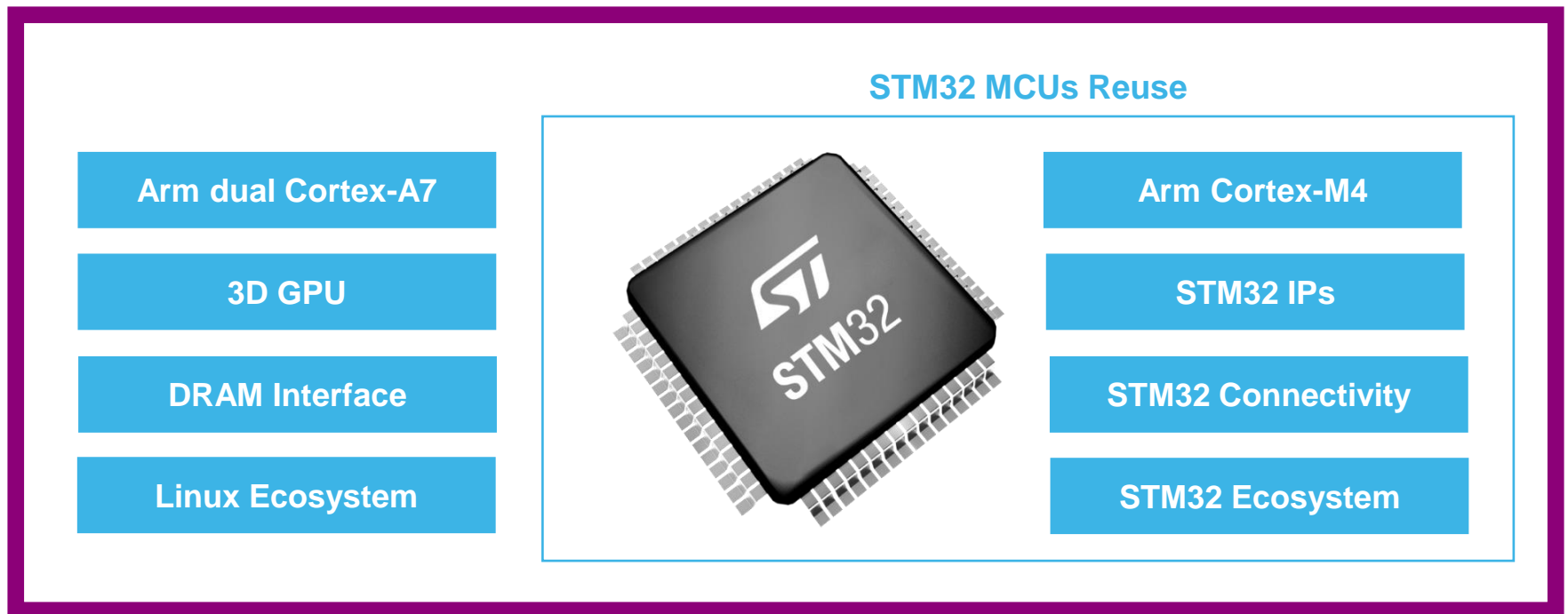
Cortex-M7 @ 400MHz (856 DMIPS)
Cortex-M4 @ 200MHz (250 DMIPS)

STM32 High Performance Portfolio



STM32MP1 introduction

STM32MP1





STM32MP157 Block Diagram





STM32MP1 hardware solutions

Speed-up evaluation, prototyping and design



EV1



DK2

DK1



Evaluation Board

Full feature STM32MP1 evaluation

- STM32MP157A-EV1
- STM32MP157C-EV1

2 Versions

Discovery Board

Flexible prototyping & demo

- STM32MP157A-DK1
 - STM32MP157C-DK2
- + MIPI DSI WVGA display
+ Wi-Fi/BT combo module

2 Versions

Boards & SoM*s

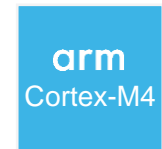
3rd Parties Boards for prototyping and production

- Board Specification from Linaro (96boards.org)
- Commercial SoM w/ different forms

*System on Module



A fully integrated design suite leveraging the stm32cube environment



STM32MPU Embedded Software Distribution

Cortex-m4 : stm32cubemp1 package

Full re-use of STM32 MCU Cube firmware on Arm Cortex-M



Several APIs to access peripherals



Collection of Middleware components for Cortex-M



Hundreds of Examples



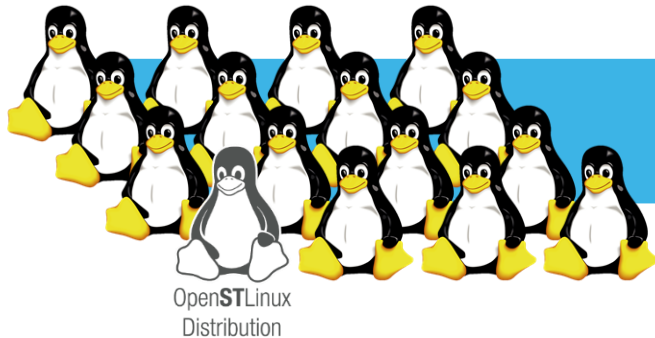
Production-ready Quality



Business-friendly license terms

Cortex-a7 : simplify your linux development

Fully mainlined open source Linux distribution for Arm Cortex-A7



STM32MP1 SoC drivers
already adopted by the Linux community

STM32MP1 supported in Linux 4.19 LTS

Fully compliant
with
open-source
standards



yocto
PROJECT



Pre-integrated
Secure OS

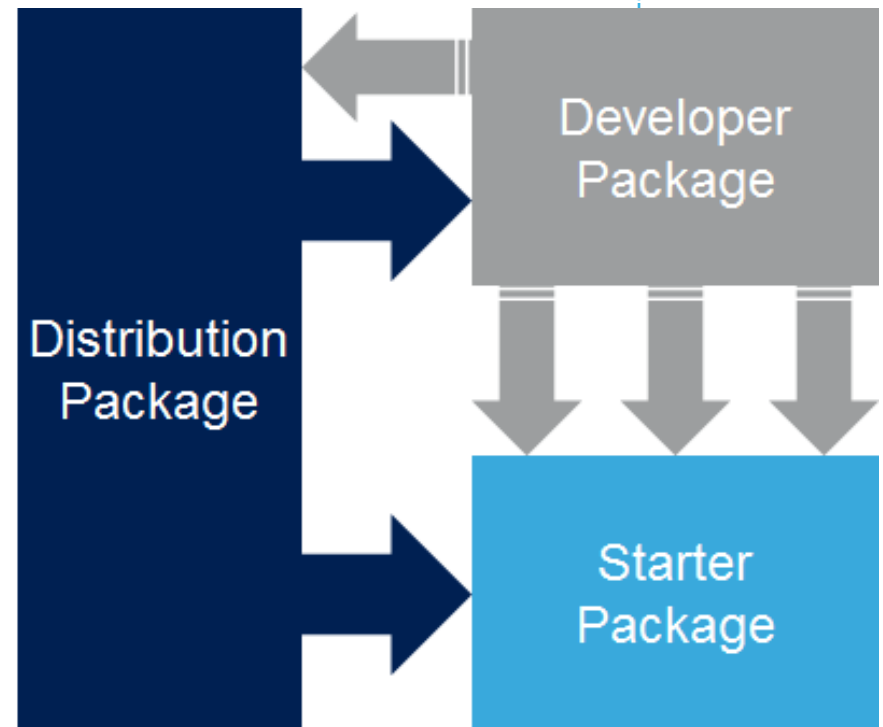


OP-TEE
.org

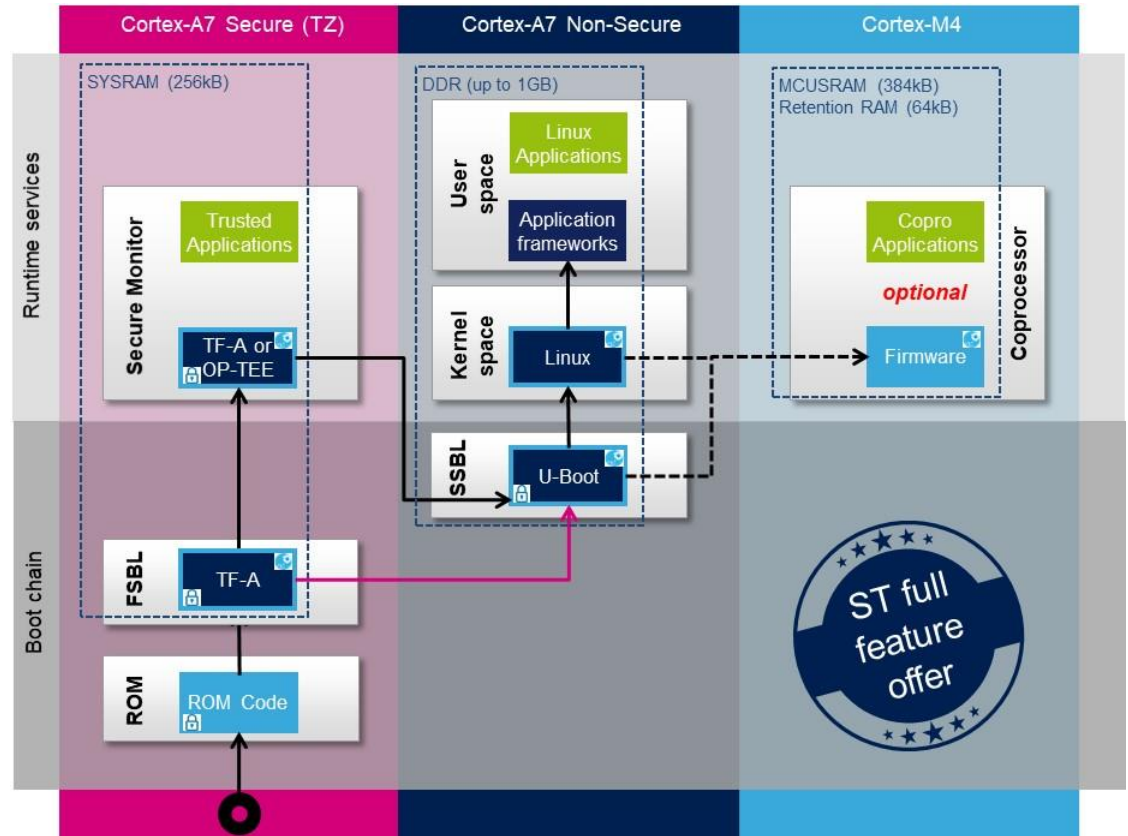
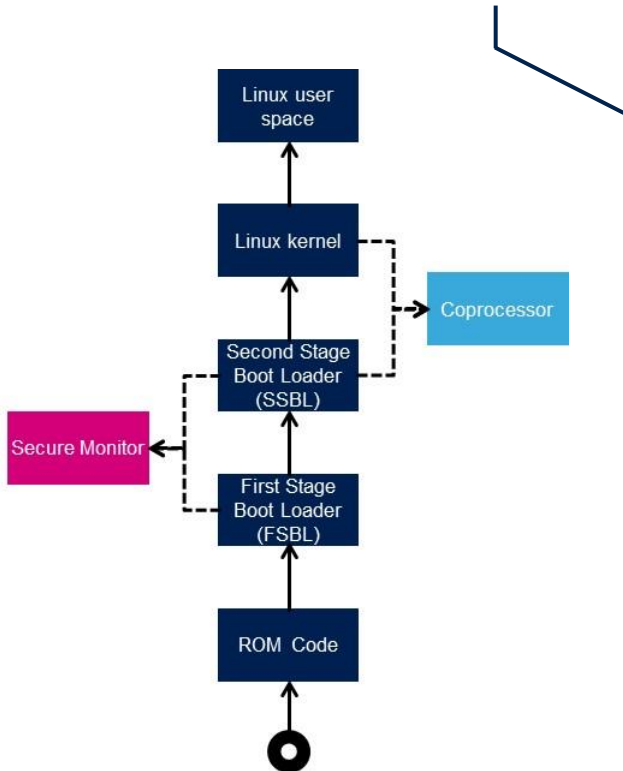
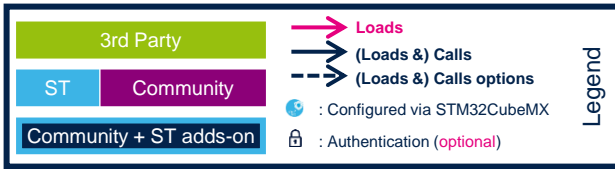
One distribution, three packages

Today's workshop

- **Starter Package**
 - To quickly and easily start with any STM32MP microprocessor device
- **Developer Package**
 - To add your own developments on top of the STM32MPU Embedded Software distribution
- **Distribution Package**
 - To create your own Linux® distribution, Developer Package



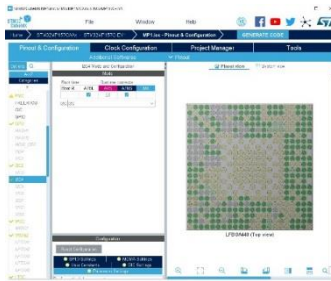
A trusted boot chain



Note : a Basic boot chain is also available, fully relying on U-Boot (instead of TF-A + U-Boot)

STM32MP1 software tools

Complete support of Arm Cortex-A + Cortex-M architecture



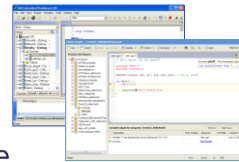
STM32CubeMX

STM32CubeMX enhanced for MPU

- Configure and generate Code
- DRAM interface tuning tool
- Device Tree generation

Today's
workshop

arm KEIL
MC6



IDEs Compile and Debug

Multi-Core Solutions

- Partners IDE
- Free IDE based on Eclipse
- Multi-core debugging

All-in-one STM32 programming tool
Multi-mode, user-friendly



STM32 Programming Tool

STM32CubeProgrammer

- Flash, DRAM and/or system memory
- OTP programming
- Signing tool & Keys generation

Today's
workshop

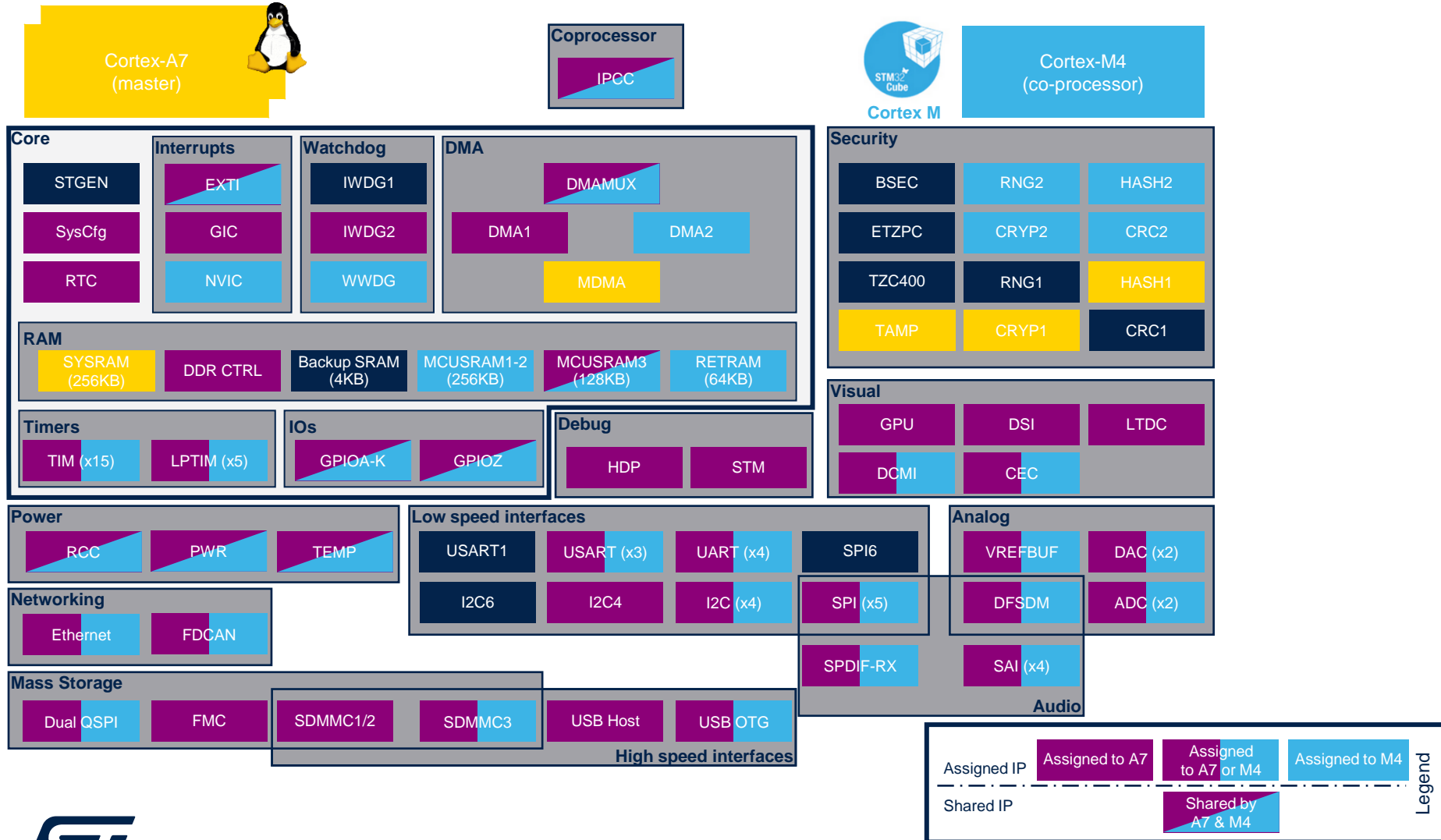




Cortex-a7 / cortex-m4 relationship



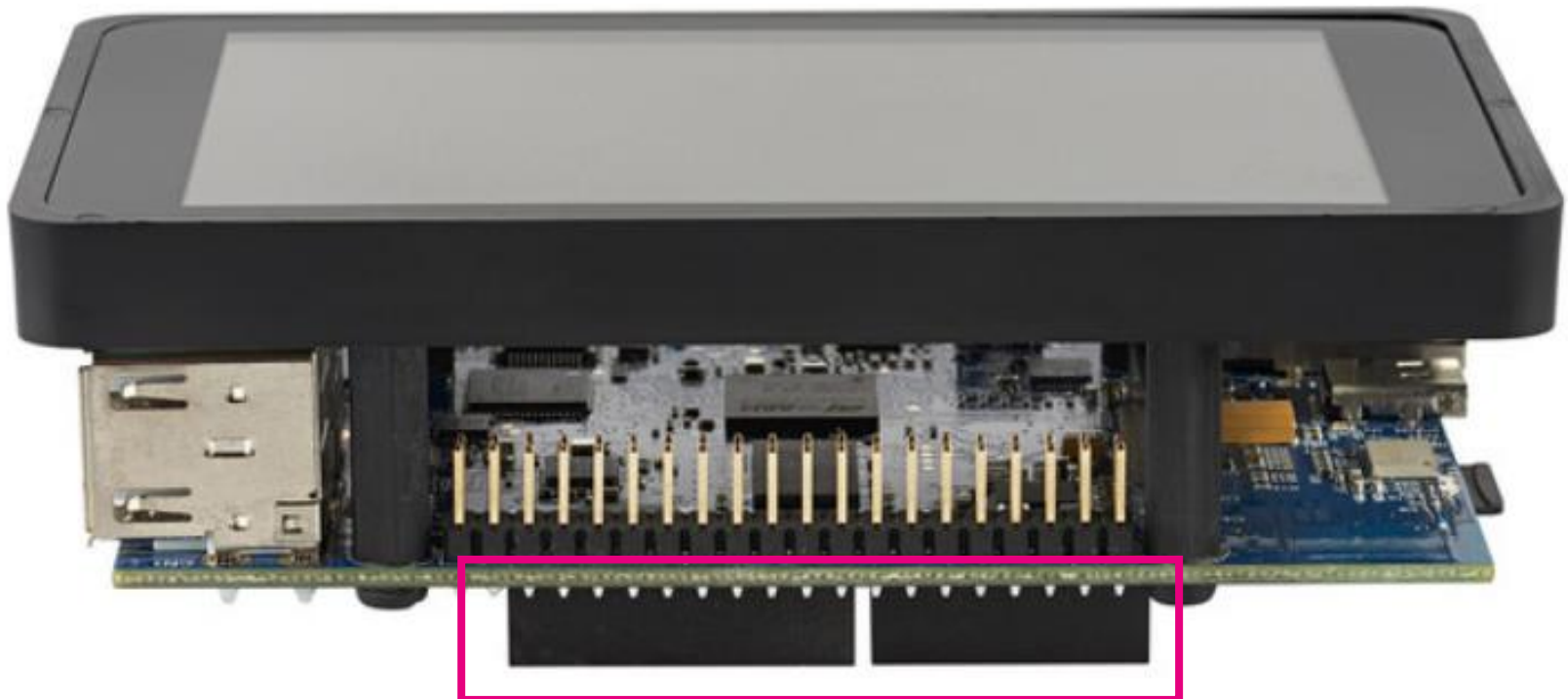
A7 / M4 IP Sharing





Easy configuration example thanks to STM32MP1 ecosystem

- Use the UART7 from the GPIO expander of the discovery board to communicate with a laptop.



- All the boards information are available on st.com and/or wiki.st.com.
 - For this example we have to get the UART7 information of the Discovery board GPIO expansion on the wiki.
- https://wiki.st.com/stm32mpu/wiki/STM32MP157x-DKx_-_hardware_description#Arduino_Uno_connector

Pin names	Signal name	STM32 pin	Comment
1	ARD_D0	PE7	UART7_RX
2	ARD_D1	PE8	UART7_TX
3	ARD_D2	PE1	IO
4	ARD_D3	PD14	TIM4_CH3
5	ARD_D4	PE10	IO
6	ARD_D5	PD15	TIM4_CH4
7	ARD_D6	PE9	TIM1_CH1
8	ARD_D7	PD1	IO

Socket 8x1 (CN14)



Boot the starter package

- As soon as the starter package has been flashed you can boot the board and enjoy Linux world.
- Check the number of UART used by the system.
- The UART7 is not configured in the system.

```
root@stm32mp1:/sys/devices/platform/soc# ls
40004000.timer          48000000.dma          50000000.rcc          50092000.etf          58005000.sdmmc        5a002000.watchdog    modalias
4000b000.audio-controller 48001000.dma          50001000.pwr          50093000.tpiu        58007000.sdmmc        5a006000.usbphyc     of_node
4000e000.serial          48002000.dma-router  5000d000.interrupt-controller 500a0000.stm          5800a000.ethernet    5c002000.i2c         power
40010000.serial          48003000.adc          50020000.syscon       500dc000.etm          5800d000.usbh-ehci    5c004000.rtc         soc:pin-controller-z@54004000
40012000.i2c             49000000.usb-otg      50025000.vrefbuf      500dd000.etm          59000000.gpu          5c005000.nvmm        soc:pin-controller@50002000
40016000.cec             4c000000.hwspinlock  50028000.thermal      54003000.rng          5a000000.dsi          5c00a000.tamp        subsystem
4400b000.sai             4c001000.mailbox     50091000.funnel       58000000.dma          5a001000.display-controller driver_override       uevent
```

- 0x4000E000 → Uart2 /*Bluetooth*/
- 0x40010000 → Uart4 /*Console*/

```
root@stm32mp1:~# ls /dev/ttySTM*
/dev/ttySTM0
root@stm32mp1:~#
```

0x40018400 - 0x40018FFF	3 KB	Reserved	-
0x40018000 - 0x400183FF	1 KB	UART7	<i>USART registers</i>
0x40017400 - 0x40017FFF	3 KB	Reserved	-



- The STM32MP1 is supported in CubeMx since the version 5.1.0 on Linux and Windows.
- CubeMx is used for :
 - configure pin assignments, the clock tree, or internal peripherals
 - configure and tune DDR parameters
 - generate HAL initialization code for Cortex-M4
 - generate the Device Tree for a Linux kernel, TF-A and U-Boot firmware for Cortex-A7
- Steps used to add a new UART in CubeMx are :
 - Select the board
 - Add the peripheral
 - Check/modify the pin use for the peripheral
 - Check/modify the clock configuration





Cubemx : board selection

New Project from a MCU

MCU Selector Board Selector

MCU Filters

Part Number Search

Core

Check/Uncheck All

☒ ARM Cortex-A7 + ARM Cortex-M4

☐ ARM Cortex-M0

☐ ARM Cortex-M0+

☐ ARM Cortex-M3

☐ ARM Cortex-M33

☐ ARM Cortex-M4

☐ ARM Cortex-M7

Series

Line

Package

Other

Price From 4.848 to 9.5

IO From 98 to 176

Eeprom = 0 (Bytes)

Flash = 0 (kBytes)

Ram = 448 (kBytes)

Freq. = 650 (MHz)

Features Block Diagram Docs & Resources Datasheet Buy Start Project

New multicore STM32MP1 Series for Industrial and IoT applications

STM32MP1

OpenSTLinux Distribution

MCUs List: 24 items

Display similar items

*	Part No	Ref.	Marketing St.	Unit Price for 10kU	Board	Package	Flash	RAM	IO	Freq.	GFX Score	DDR	DEBUG	HDP	PKA	PWR	RF	TAMP
☆	STM32MP153A	S...	Active	5.328		TFBGA257 0	kBytes	448 kBytes	98	650 MHz	0.0	1	1	1	0	1	0	1
☆		S...	Active	7.5		LFPGA448 0	kBytes	448 kBytes	176	650 MHz	0.0	1	1	1	0	1	0	1
☆		S...	Active	7.5		LFPGA354 0	kBytes	448 kBytes	98	650 MHz	0.0	1	1	1	0	1	0	1
☆		S...	Active	7.5		TFBGA361 0	kBytes	448 kBytes	148	650 MHz	0.0	1	1	1	0	1	0	1
☆		S...	Active	7.5		TFBGA257 0	kBytes	448 kBytes	98	650 MHz	0.0	1	1	1	0	1	0	1
☆	STM32MP153C	S...	Active	8.1		LFPGA448 0	kBytes	448 kBytes	176	650 MHz	0.0	1	1	1	0	1	0	1
☆		S...	Active	8.1		LFPGA354 0	kBytes	448 kBytes	98	650 MHz	0.0	1	1	1	0	1	0	1
☆		S...	Active	8.1		TFBGA361 0	kBytes	448 kBytes	148	650 MHz	0.0	1	1	1	0	1	0	1
☆		S...	Active	8.1		TFBGA257 0	kBytes	448 kBytes	98	650 MHz	0.0	1	1	1	0	1	0	1
☆	STM32MP157A	S...	Active	8.9	STM32MP157A-EV1	LFPGA448 0	kBytes	448 kBytes	176	650 MHz	0.0	1	1	1	0	1	0	1
☆		S...	Active	8.9		LFPGA354 0	kBytes	448 kBytes	98	650 MHz	0.0	1	1	1	0	1	0	1
☆		S...	Active	8.9		TFBGA361 0	kBytes	448 kBytes	148	650 MHz	0.0	1	1	1	0	1	0	1
☆		S...	Active	8.9		TFBGA257 0	kBytes	448 kBytes	98	650 MHz	0.0	1	1	1	0	1	0	1
☆	STM32MP157C	S...	Active	9.5	STM32MP157C-EV1	LFPGA448 0	kBytes	448 kBytes	176	650 MHz	0.0	1	1	1	0	1	0	1
☆		S...	Active	9.5		LFPGA354 0	kBytes	448 kBytes	98	650 MHz	0.0	1	1	1	0	1	0	1
☆		S...	Active	9.5	STM32MP157C-DK2	TFBGA361 0	kBytes	448 kBytes	148	650 MHz	0.0	1	1	1	0	1	0	1
☆		S...	Active	9.5		TFBGA257 0	kBytes	448 kBytes	98	650 MHz	0.0	1	1	1	0	1	0	1



Cubemx : board selection

MX New Project from a MCU

MCU Selector Board Selector

Board Filters

Part Number Search

STM32MP157C-DK2

Vendor

Type

MCU Series

Check/Uncheck All

STM32MP1

Other

Price = 99.0

Oscillator Freq. = 0 (MHz)

Peripheral

Accelerometer

Analog I/O

Arduino Form Factor

Audio Line In

Audio Line Out

Button

CAN

Camera

Compass

Custom Form Factor

Digital I/O

Ethernet

Gyroscope

IrDA

Joystick

LCD Display (Graphics)

LCD Display (Segment)

Features

Large Picture

Docs & Resources

Datasheet

Buy

Start Project

STM32MP157C-DK2

STM32 MP1


PREVIEW

Preview

Product is in design stage

Unit Price (US\$) : 99.0

Mounted device: STM32MP157CACx



The STM32MP157A-DK1 and STM32MP157C-DK2 Discovery kits leverage the capabilities of STM32MP1 Series microprocessors to allow users easily develop applications using STM32 MPU OpenSTLinux Distribution software for the main processor and STM32CubeMP1 software for the co-processor.

They include an ST-LINK embedded debug tool, LEDs, push-buttons, one Ethernet 1-Gbps connector, one USB Type-C OTG connector, four USB Type-A Host connectors, one HDMI transceiver, one stereo headset jack with analog microphone, and one microSD connector.


To expand the functionality of the STM32MP157A-DK1 and STM32MP157C-DK2 Discovery kits, two GPIO expansion connectors are also available for Arduino and Raspberry Pi shields.


Additionally, the STM32MP157C-DK2 Discovery kit features an LCD display with a touch panel, and Wi-Fi and Bluetooth Low Energy capability.

Features

- 1 x DDR3L 16 bits (4GB)
- 10M/100M/1G Ethernet interface
- USB host hub
- USB type-C
- Audio codec
- DSI 720p display
- HDMI
- Speaker / Headset
- Connector ext. (GPIO expansion)

Boards List: 1 item

*	Overview	Part No	Type	Marketing Status	Unit Price (US\$)	Mounted Device	Kit Contents	Included in Kit
☆		STM32MP157C-DK2	Discovery	Preview	99.0	STM32MP157CACx		

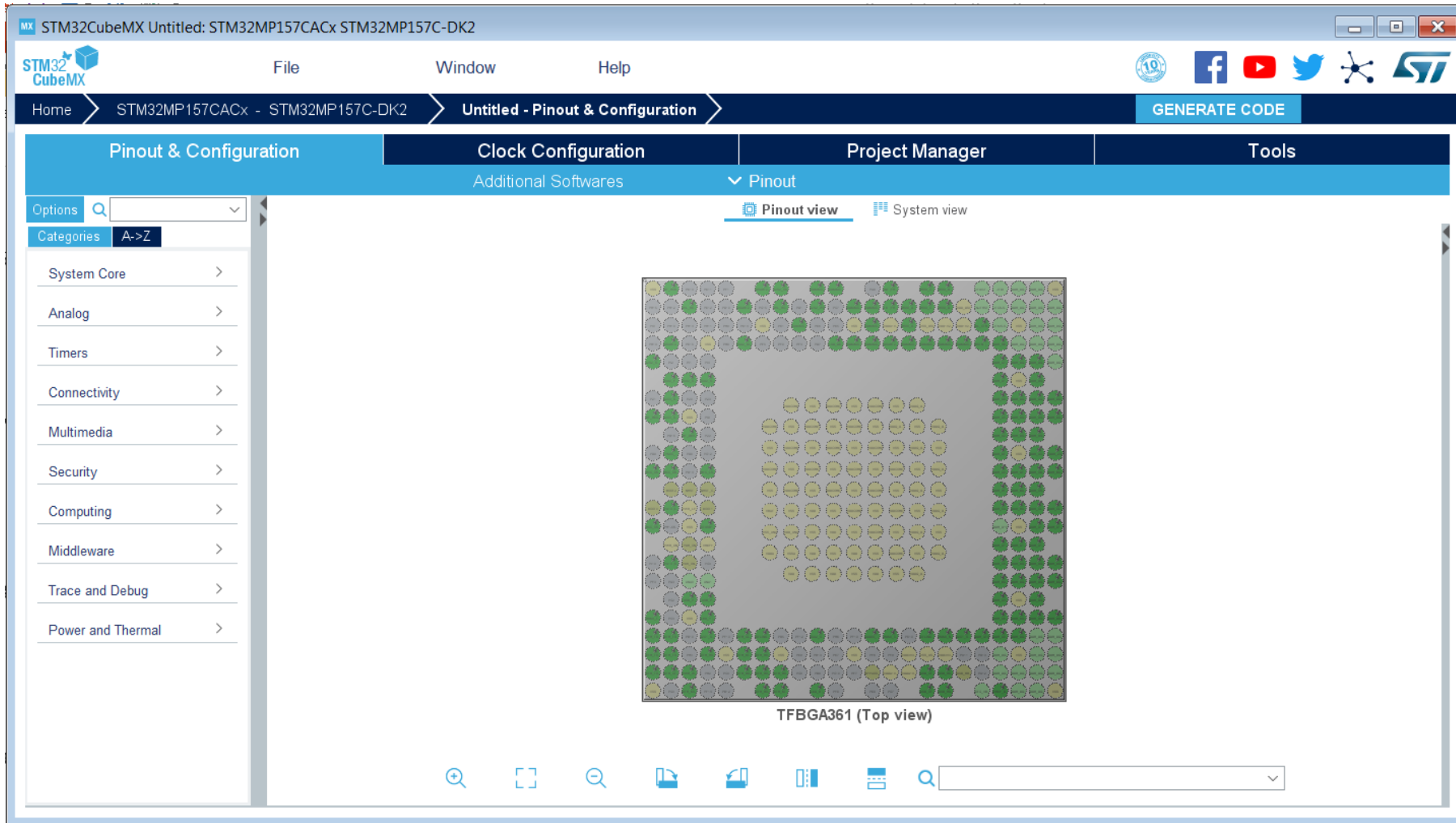


 life.augmented

25

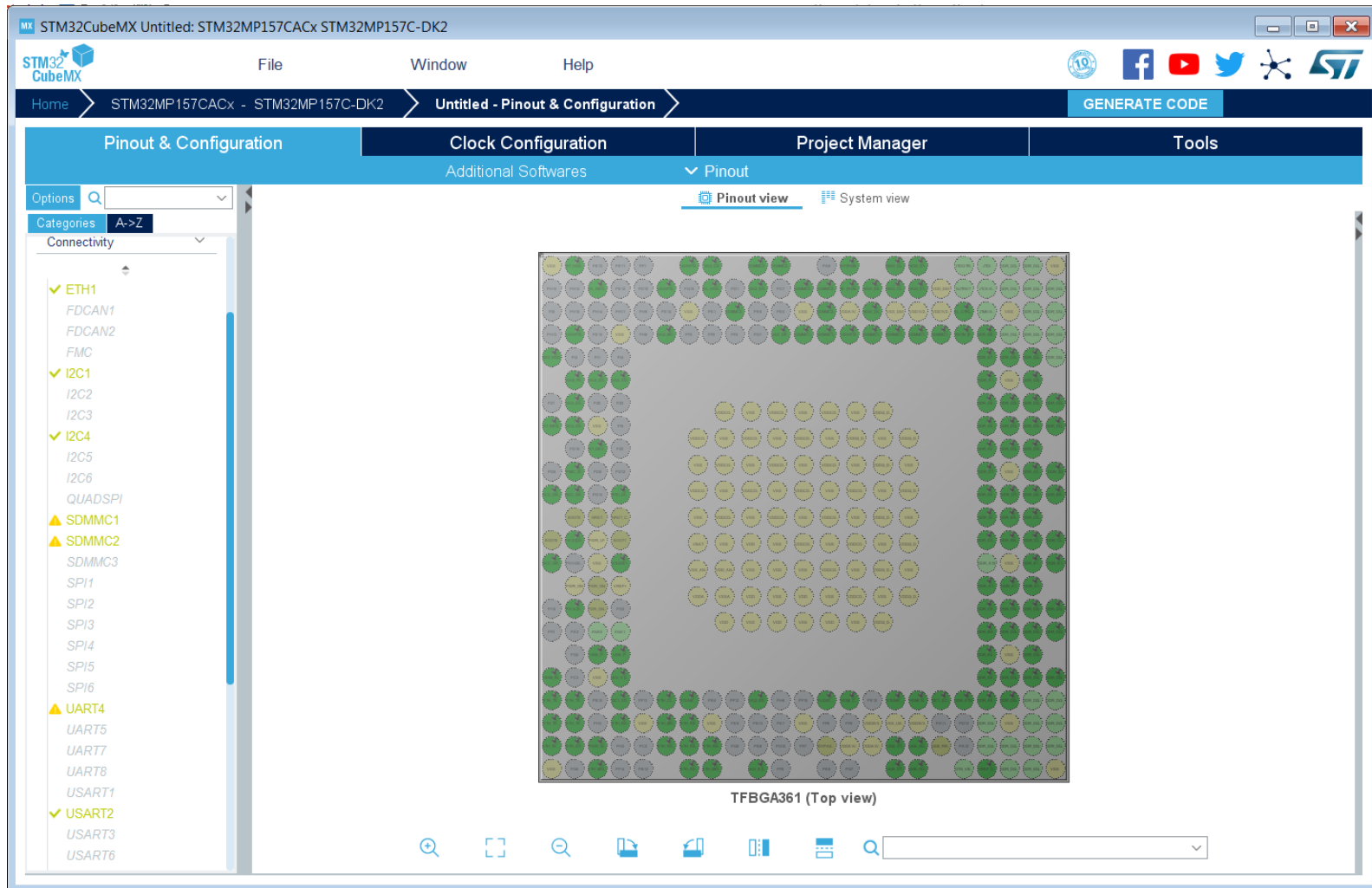


Cubemx : same for MCU or MPU





Cubemx : peripheral selection





Cubemx : peripheral selection

Home > STM32MP157CACx - STM32MP157C-DK2 > Untitled - Pinout & Configuration > GENERATE CODE

Pinout & Configuration | Clock Configuration | Project Manager | Tools

Software Packs | Pinout

UART7 Mode and Configuration

Mode

Boot time: Runtime contexts:

Boot ROM	Boot loader	A7S	A7NS	Cortex-M4
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Mode: Asynchronous

Hardware Flow Control (RS232): Disable

☐ Hardware Flow Control (RS485)

Configuration

Reset Configuration

GIC Settings | DMA Settings | GPIO Settings

Parameter Settings | User Constants

Search Signals

Search (Ctrl+F)

Pin Name	Signal on Pin	GPIO mode	GPIO Pull-up/...	Maximu
PA8	UART7_RX	Alternate funct...	No pull-up and...	n/a
PF7	UART7_TX	Alternate Func...	No pull-up and...	Low

Pinout view | System view

TFBGA361 (Top view)



Cubemx : pin configuration

- Default pin set in cubeMx are not the one selected on the GPIO expansion.
- PA8 & PF7 use as default
- PE7 & PE8 used on discovery board
- Change the UART7 pin is very easy on CubeMx

Pin names	Signal name	STM32 pin	Comment
1	ARD_D0	PE7	UART7_RX
2	ARD_D1	PE8	UART7_TX
3	ARD_D2	PE1	IO
4	ARD_D3	PD14	TIM4_CH3
5	ARD_D4	PE10	IO
6	ARD_D5	PD15	TIM4_CH4
7	ARD_D6	PE9	TIM1_CH1
8	ARD_D7	PD1	IO

Socket 8x1 (CN14)



Cubemx pin configuration

MX STM32CubeMX Untitled*: STM32MP157CACx STM32MP157C-DK2

File Window Help

Home STM32MP157CACx - STM32MP157C-DK2 Untitled - Pinout & Configuration GENERATE CODE

Pinout & Configuration Clock Configuration Project Manager Tools

Software Packs Pinout

Pinout view System view

Categories A->Z

UART7 Mode and Configuration

Mode

Boot time: Runtime contexts:

Boot ROM	Boot loader	A7S	A7NS	Cortex-M4
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Mode Asynchronous

Hardware Flow Control (RS232) Disable

☐ Hardware Flow Control (RS485)

Configuration

Reset Configuration

GIC Settings DMA Settings GPIO Settings

Parameter Settings User Constants

Search Signals

Search (Ctrl+F)

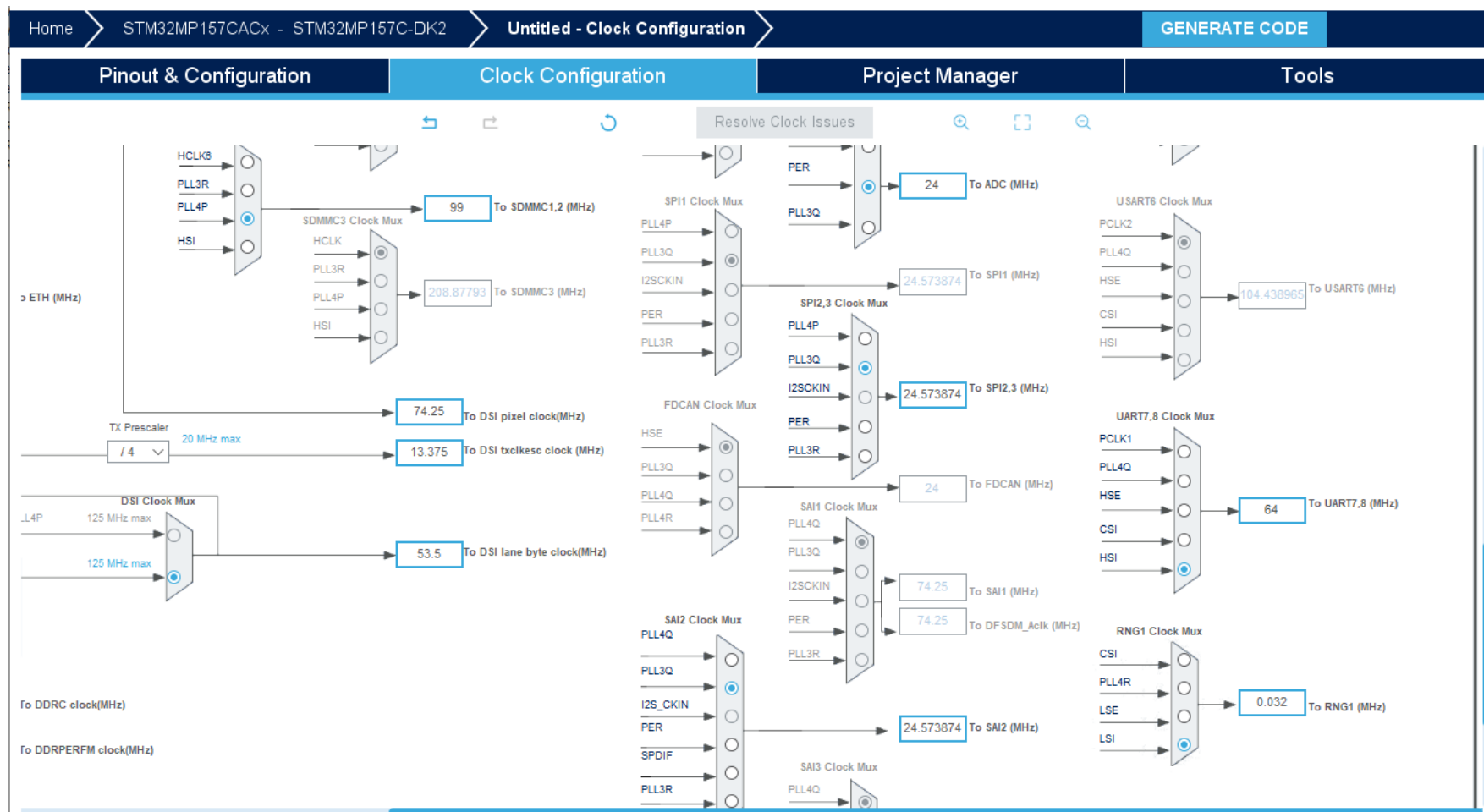
Pin Name	Signal on Pin	GPIO mode	GPIO Pull-up/...	Maximum
PE7	UART7_RX	Alternate funct...	No pull-up and...	n/a
PE8	UART7_TX	Alternate Func...	No pull-up and...	Low

TFBGA361 (Top view)

Reset_State
DFSDM1_CKIN2
FMC_D5
QUADSPI_BK2_IO1
TIM1_CH1N
UART7_TX
GPIO_Input
GPIO_Output
GPIO_Analog
EVENTOUT
GPIO_EXTI8

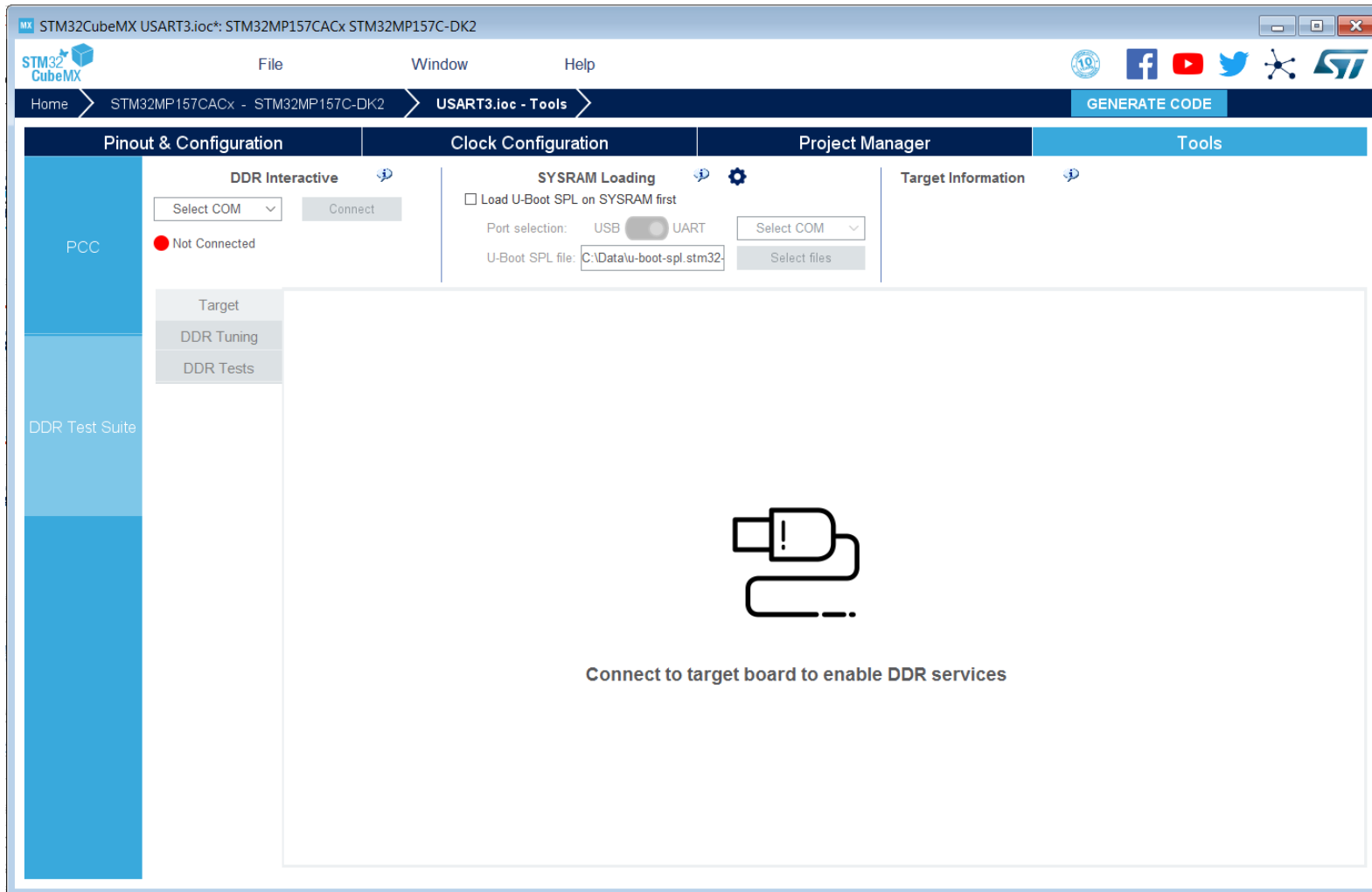


Cubemx clock configuration



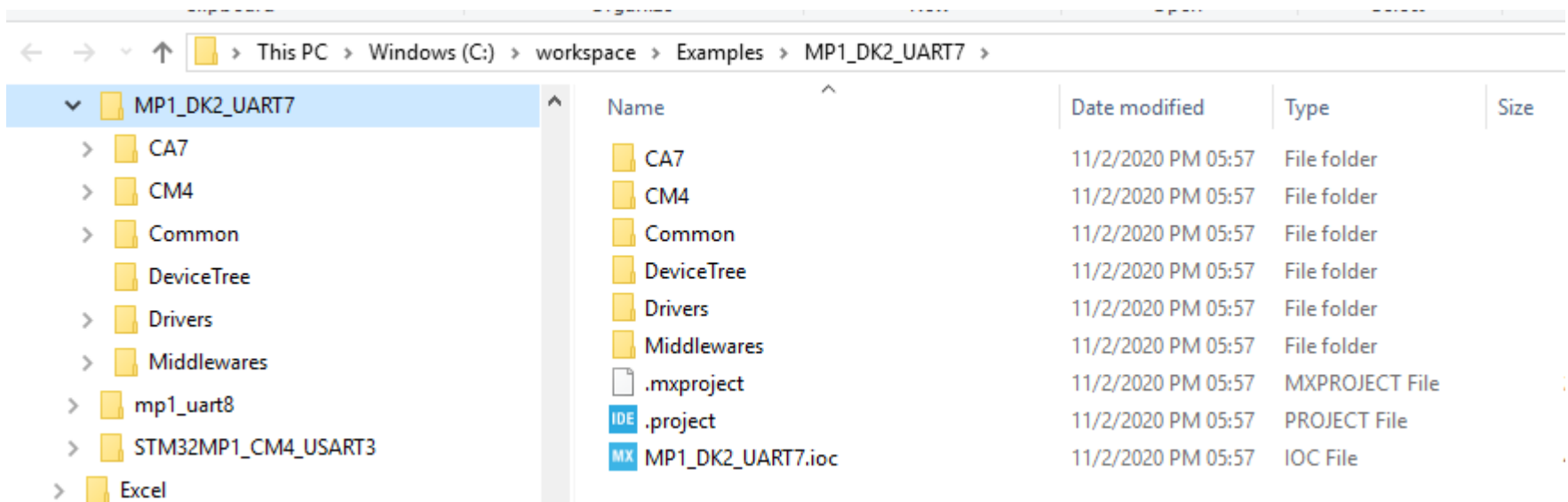


Cubemx : DDR plugin





Cubemx : generate code





Cubemx : device tree

```
uart7_pins_mx: uart7_mx-0 {
    pins1 {
        pinmux = <STM32_PINMUX('E', 7, AF7)>; /* UART7_RX */
        bias-disable;
    };
    pins2 {
        pinmux = <STM32_PINMUX('E', 8, AF7)>; /* UART7_TX */
        bias-disable;
        drive-push-pull;
        slew-rate = <0>;
    };
};

uart7_sleep_pins_mx: uart7_sleep_mx-0 {
    pins {
        pinmux = <STM32_PINMUX('E', 7, ANALOG)>; /* UART7_RX */
        pinmux = <STM32_PINMUX('E', 8, ANALOG)>; /* UART7_TX */
    };
};
```

```
&uart7{
    pinctrl-names = "default", "sleep";
    pinctrl-0 = <&uart7_pins_mx>;
    pinctrl-1 = <&uart7_sleep_pins_mx>;
    status = "okay";

    /* USER CODE BEGIN uart7 */
    /* USER CODE END uart7 */
};
```

- This is an extract of the Linux device tree generated by CubeMx, as the Discovery board configuration is already existing we have to report only the USART7 configuration
- There is the pin control configuration with the PE7 & PE8.
- And the peripheral configuration with the « okay » status for the activation



Cubemx : device tree

```
st,pkcs = <
    CLK_CKPER_HSE
    CLK_ETH_PLL4P
    CLK_SDMMC12_PLL4P
    CLK_DSI_DSIPLL
    CLK_STGEN_HSE
    CLK_USBPHY_HSE
    CLK_SPI2S1_DISABLED
    CLK_SPI2S23_PLL3Q
    CLK_SPI45_DISABLED
    CLK_SPI6_DISABLED
    CLK_I2C46_HSI
    CLK_SDMMC3_DISABLED
    CLK_USBO_USBPHY
    CLK_ADC_CKPER
    CLK_CEC_LSE
    CLK_I2C12_HSI
    CLK_I2C35_DISABLED
    CLK_UART1_DISABLED
    CLK_UART24_HSI
    CLK_UART35_DISABLED
    CLK_UART6_DISABLED
    CLK_UART78_HSI
    CLK_SPDIF_DISABLED
    CLK_SAI1_DISABLED
    CLK_SAI2_PLL3Q
    CLK_SAI3_DISABLED
    CLK_SAI4_DISABLED
    CLK_RNG1_LSI
    CLK_LPTIM1_DISABLED
    CLK_LPTIM23_DISABLED
    CLK_LPTIM45_DISABLED
```

- This is an extract of the Uboot device tree generated by cubeMx, we keep only the clock configuration
- In the system the clock device tree can be managed by TF-A or u-boot.
- There isn't UART7 peripheral configuration in this device tree like we select only A7NS



Developer package

- The developer package is available on wiki.st.com:
 - https://wiki.st.com/stm32mpu/wiki/STM32MP1_Developer_Package
- With the developer package you can rebuild each binary of each software layer (file system partition can be generated only by distribution kit).
- We have to modify the existing device tree adding the UART7 device tree configuration.
- In this example we have to rebuild u-boot (the device tree is included inside the binary) and the Linux DTS (no need to rebuild the kernel binary).
 - `make ARCH=arm dtbs LOADADDR=0xC2000040 O="$PWD/../build"`



Update the firmware and test the UART7

- To update u-boot you can use the STM32Programmer (flash u-boot binary only not needed to flash the all system).
- To update the Linux device tree you can simply copy it on the bootfs file system.(through Ethernet, via an usb key, putting the sdcard on you laptop...)

```
root@stm32mp1:~# ls /sys/devices/platform/soc/
40004000.timer          48001000.dma          50020000.syscon       54003000.rng          5a002000.watchdog    power
4000b000.audio-controller 48002000.dma-router   50025000.vrefbuf     58000000.dma          5a006000.usbphyc     soc:pin-controller-z@54004000
4000e000.serial         48003000.adc          50028000.thermal     58005000.sdmmc        5c002000.i2c         soc:pin-controller@50002000
4000f000.serial         49000000.usb-otg      50091000.funnel      58007000.sdmmc        5c004000.rtc         subsystem
40010000.serial         4c000000.hwspinlock  50092000.etf         5800a000.ethernet     5c005000.nvmem       uevent
40012000.i2c            4c001000.mailbox     50093000.tpiu        5800d000.usbh-ehci    5c00a000.tamp
40016000.cec            50000000.rcc          500a0000.stm          59000000.gpu          driver_override
4400b000.sai            50001000.pwr          500dc000.etm          5a000000.dsi          modalias
48000000.dma            5000d000.interrupt-controller 500dd000.etm          5a001000.display-controller of_node
```

```
root@stm32mp1:~# ls /dev/ttySTM*
/dev/ttySTM0 /dev/ttySTM1
root@stm32mp1:~#
```



Check result

- Check result on board

```
root@stm32mp1:~# ls /dev/ttySTM*  
/dev/ttySTM0 /dev/ttySTM2
```

- Say Hello... Nothing happen

```
root@stm32mp1:~# echo Hello > /dev/ttySTM2
```

- Check serial port setting

```
root@stm32mp1:~# stty -a -F /dev/ttySTM2  
speed 9600 baud; rows 0; columns 0; line = 0;  
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
```

- Change serial port setting

```
root@stm32mp1:~# stty -F /dev/ttySTM2 115200
```

- Say Hello again

```
root@stm32mp1:~# echo Hello > /dev/ttySTM2
```



Releasing your creativity



www.st.com/STM32MP1