Getting started with the STMicroelectronics X-CUBE-TCPP software package for STM32CubeMX

Introduction

This document provides the guidelines to configure and use the X-CUBE-TCPP software package V4.2.0 for STM32CubeMX (minimum required version V6.13.0). The document contains a description of the provided sample applications, a description of the steps required to configure generic projects for USBPD Sink, Source or Dual-Role application using or not the X-NUCLEO-SNK1M1 or the X-NUCLEO-SRC1M1 or the X-NUCLEO-DRP1M1 expansion board with a Nucleo board. This package is compatible with Cortex-M0, M0+, M3, M4, M33 and M7. And, except for applications USB type C – 5V only, it is dedicated to MCU embedding the UCPD peripheral such as some MCU in following families STM32G0xx, STM32G4xx, STM32H5xx, STM32U5xx, STM32L5xx or STM32H7xx.

Information and documentation related to the TCPP0x components in USBPD applications, the X-NUCLEO-SNK1M1 or X-NUCLEO-SRC1M1 or X-NUCLEO-DRP1M1 expansion board and the ST expansion software for TCPP0x are available on www.st.com.

Contents

List of figures

1 Acronyms and Abbreviations

Acronym	Description
GUI	Graphic User Interface
PDO	Power Data Object
USB	Universal Serial Bus
UCPD	USB-C and PD controller
USBPD	Universal Serial Bus Power Delivery

2 What is STM32Cube?

STMCube™ initiative was originated by STMicroelectronics to ease developers' life by reducing development efforts, time and cost. STM32Cube covers STM32 portfolio.

STM32Cube Version 1.x includes:

- The STM32CubeMX, a graphical software configuration tool that allows to generate C initialization code using graphical wizards.
- A comprehensive embedded software platform, delivered per series (such as STM32CubeF4 for STM32F4 series)
  - ✓ The STM32Cube HAL, an STM32 abstraction layer embedded software, ensuring maximized portability across STM32 portfolio
  - ✓ A consistent set of middleware components such as RTOS, USB, TCP/IP, Graphics
  - ✓ All embedded software utilities coming with a full set of examples.

Information about STM32Cube are available on www.st.com at:
http://www.st.com/stm32cube

3 License

The software provided in this package is licensed under Software License Agreement SLA0095

4 Sample application description

In this section a short overview of the sample applications and examples included in the X-CUBE-TCPP pack is provided. For more information, kindly refer User Manual of X-CUBE-TCPP available on www.st.com.

The sample applications/examples:

- Are ready-to-use projects that can be generated through the STM32CubeMX for any Nucleo board and using the X-NUCLEO-SNK1M1 or the X-NUCLEO-SRC1M1 or the X-NUCLEO-DRP1M1 expansion board.
- Show the users how to use the APIs to correctly initialize and use the TCPP0x in USBPD applications.

- USB Power-Delivery Sink application
- USB Power-Delivery Source application
- USB Power-Delivery Dual Role application
- USB no Power-Delivery Sink application (5V only 1.5A or 3.0A)
- USB no Power-Delivery Source application (5V only)

5 System Setup Guide

5.1 Hardware Description

This section describes the hardware components needed for developing an USBPD application.

The following sub-sections describe the individual components.

5.1.1 STM32 Nucleo platform

The STM32 Nucleo boards provide an affordable and flexible way for users to try out new ideas and build prototypes with any STM32 microcontroller lines. The Arduino™ connectivity support and ST morpho headers make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide choice of specialized expansion boards. The STM32 Nucleo board does not require any separate probe as it integrates the ST-LINK/V2-1 debugger/programmer. The STM32 Nucleo board comes with the STM32 comprehensive software HAL library together with various packaged software examples.

Information about the STM32 Nucleo boards is available on www.st.com at

http://www.st.com/stm32nucleo

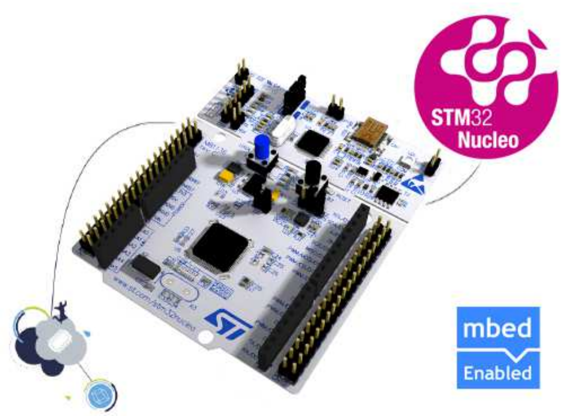


Figure 1 Firmware Architecture

5.1.2 X-NUCLEO expansion boards

The X-NUCLEO-SNK1M1 or X-NUCLEO-SRC1M1 or X-NUCLEO-DRP1M1 expansion board with a STM32 Nucleo 64 pins board (e.g., a Nucleo-G071RB) no particular hardware modification must be done.

Figure 2 STM32 Nucleo 64 with X-NUCLEO-SNK1M1

Figure 3 STM32 Nucleo 64 with X-NUCLEO-SRC1M1

Figure 4 STM32 Nucleo 64 with X-NUCLEO-DRP1M1

Figure 5 X-NUCLEO-SNK1M1 pinout

Figure 6 X-NUCLEO-SRC1M1 pinout

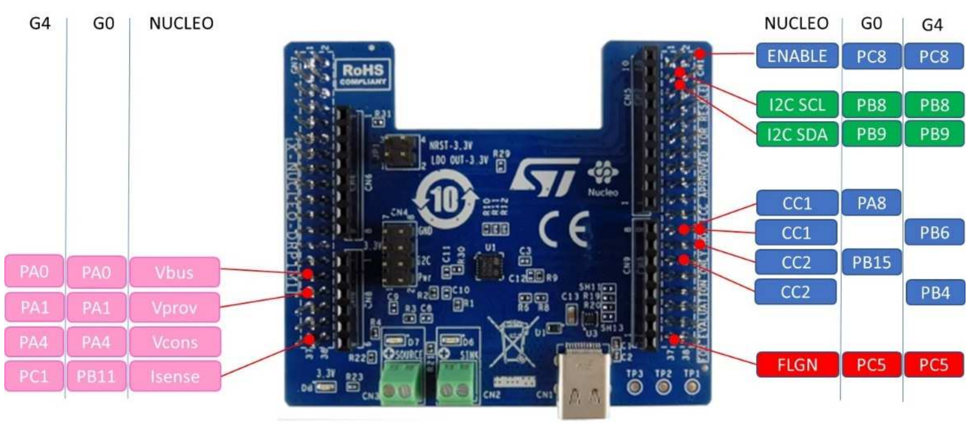


Figure 7 X-NUCLEO-DRP1M1 pinout

5.2 Software Description

The following software components are needed in order to setup the suitable development environment for creating USBPD applications for the STM32 Nucleo:

- X-CUBE-TCPP: an expansion for STM32Cube dedicated to USBPD applications development using TCPP series components. The X-CUBE-TCPP firmware and related documentation is available on st.com.
- Development tool-chain and Compiler: The STM32Cube expansion software supports the three following environments:
  - IAR Embedded Workbench for ARM® (EWARM) toolchain + ST-Link
  - RealView Microcontroller Development Kit (MDK-ARM) toolchain + ST-LINK
  - System Workbench for STM32 + ST-LINK

5.3 Hardware and Software Setup

This section describes the hardware and software setup procedures. It also describes the system setup needed for the above.

5.3.1 Hardware Setup

The following hardware components are needed:

1. One STM32 Nucleo Development platform (suggested order code: either NUCLEO-G071RB or NUCLEO-G474R for applications with UCPD or NUCLEO-L412RB in non-USBPD Sink application and NUCLEO-F446RE for non-USBPD Source application 5V only)
2. One TCPP expansion board (order code: X-NUCLEO-SNK1M1 for sink application, X-NUCLEO-SRC1M1 for Source application or X-NUCLEO-DRP1M1 for Dual-Role application)
3. One USB type A to Micro-B USB cable to connect the STM32 Nucleo to the PC

5.3.2 Software Setup

This section lists the minimum requirements for the developer to setup the SDK, run the sample testing scenario based on the GUI utility and customize applications.

5.3.2.1 Development Tool-chains and Compilers

Please select one of the Integrated Development Environments supported by the STM32Cube expansion software.

Please read the system requirements and setup information provided by the selected IDE provider.

5.3.2.2 PC Utility

The *STM32CubeMonitor-UCPD* utility can be downloaded from st.com. It is a Multi-OS : Windows®, Linus®, macOS®.) is a free software analyzer to monitor and configure USB Type-C™ and Power Delivery applications.

5.4 System Setup Guide

This section describes how to setup different hardware parts before writing and executing an application on the STM32 Nucleo board with the X-NUCLEO expansion board.

5.4.1 Installing the X-CUBE-TCPP pack in STM32CubeMX

After downloading (from www.st.com), installing and launching the STM32CubeMX (V>=6.11.0), the ‘X-CUBE-TCPP’ pack can be installed in few steps.

- From the menu, select Help -> Manage embedded software packages.

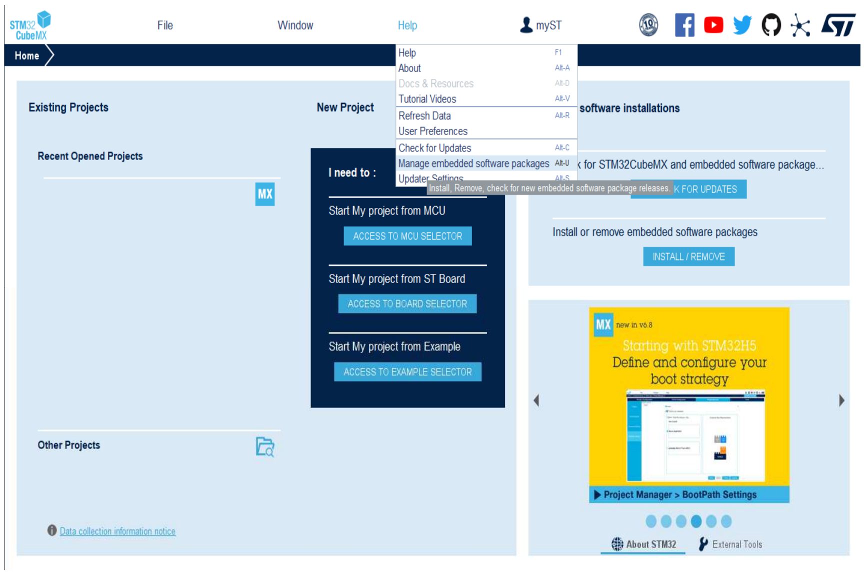


Figure 8 Managing embedded software packs in STM32CubeMX

- From the embedded Software Package Manager window, press the ‘Refresh’ button to get an updated list of the add-on packs. Go to the STMicroelectronics’ tab to find the X-Cube-TCPP pack.

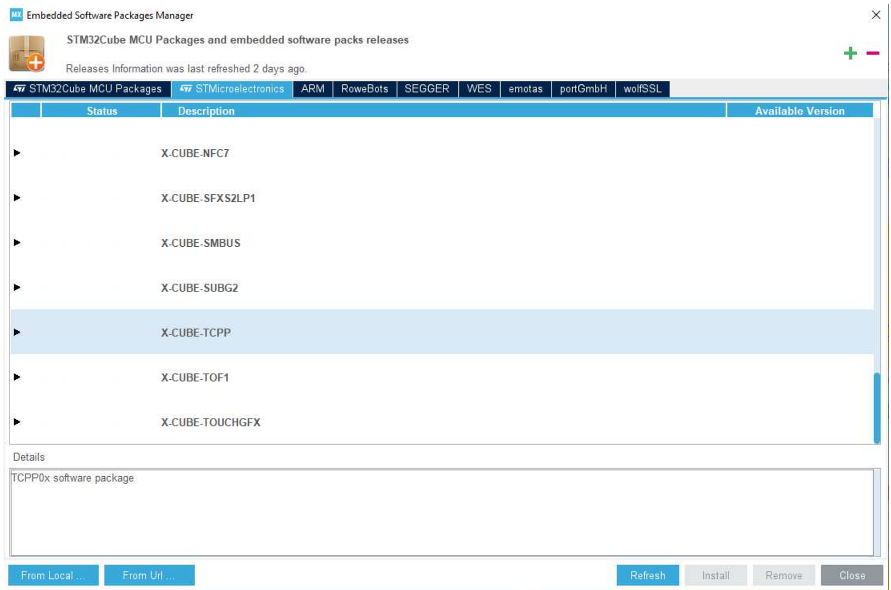


Figure 9 Installing the X-Cube-TCPP in STM32CubeMX

- Select it checking the corresponding box and install it pressing the 'Install Now' button. Once the installation is completed, the corresponding box will become green, the 'Close' button can be pressed, and the configuration of a new project can start.

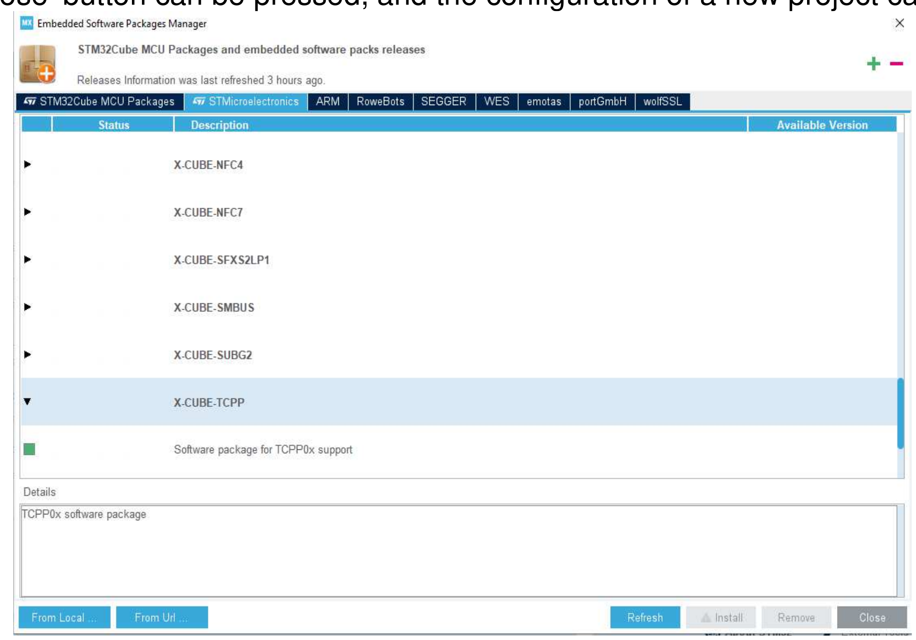


Figure 10 Installing the X-Cube-TCPP in STM32CubeMX

6 Starting a new project

After launching the STM32CubeMX, you can choose if starting a New Project from the MCU Selector if from the Board Selector.

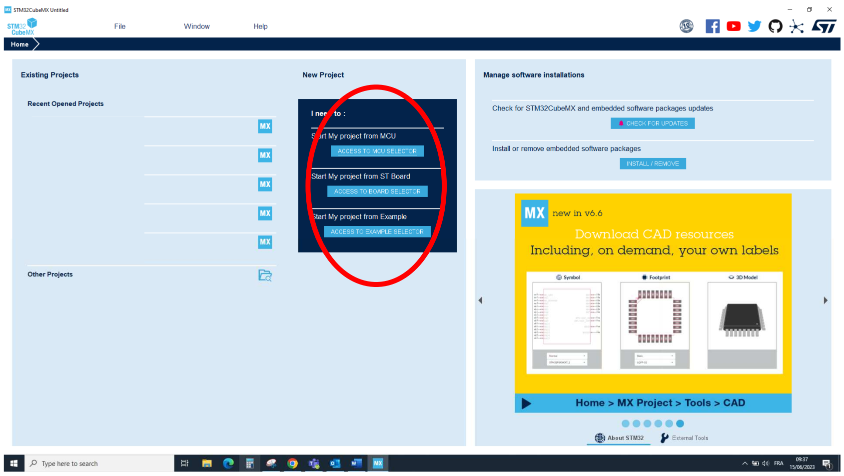


Figure 11 STM32CubeMX main page

The MCU/Board selector window will pop up. From this window, the STM32 MCU of platform can be selected.

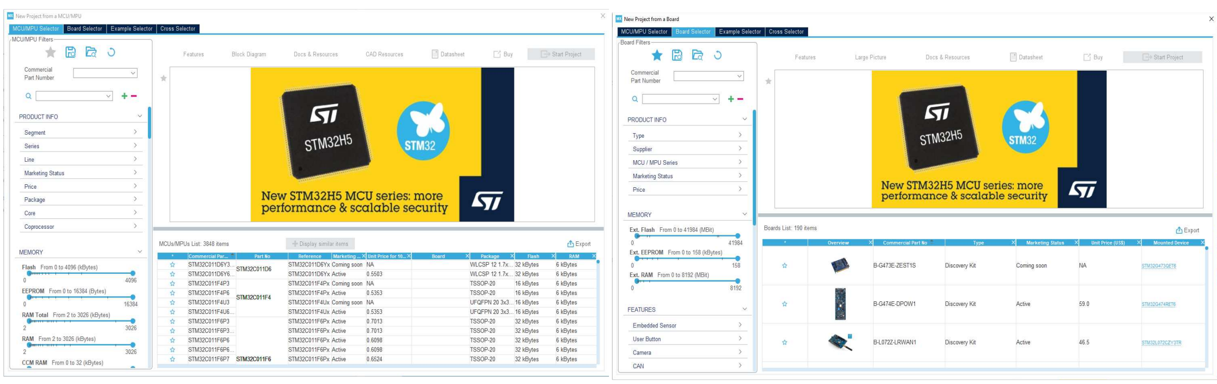


Figure 12 STM32CubeMX MCU/Board Selector windows

After selecting the MCU or the Board, the selected STM32 pinout will appear. From this window the user can set up the project, by adding one or more Additional software and peripherals and configuring the clock.

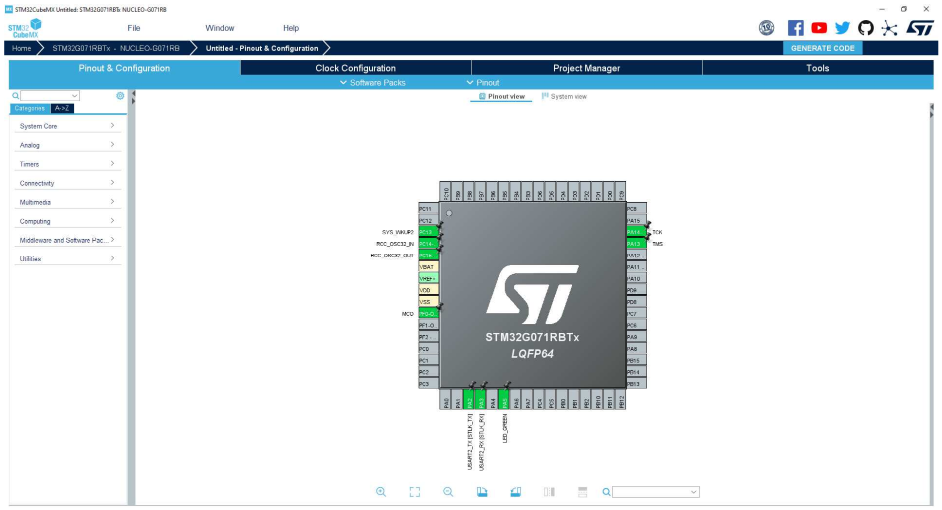


Figure 13 STM32CubeMX Pinout & Configuration window

To add the *'X-CUBE-TCPP'* additional software to the project, the *'Software Packs'* and then '*Select Components'* button must be checked.

From the Software Pack Component Selector window, the user can either choose to generate, for the selected MCU/Board, one of the enclosed sample applications or a new project, in this latter case, the user must just implement the main application logic, without bothering with the pinout and peripherals configuration code that will be automatically generated by STM32CubeMX.

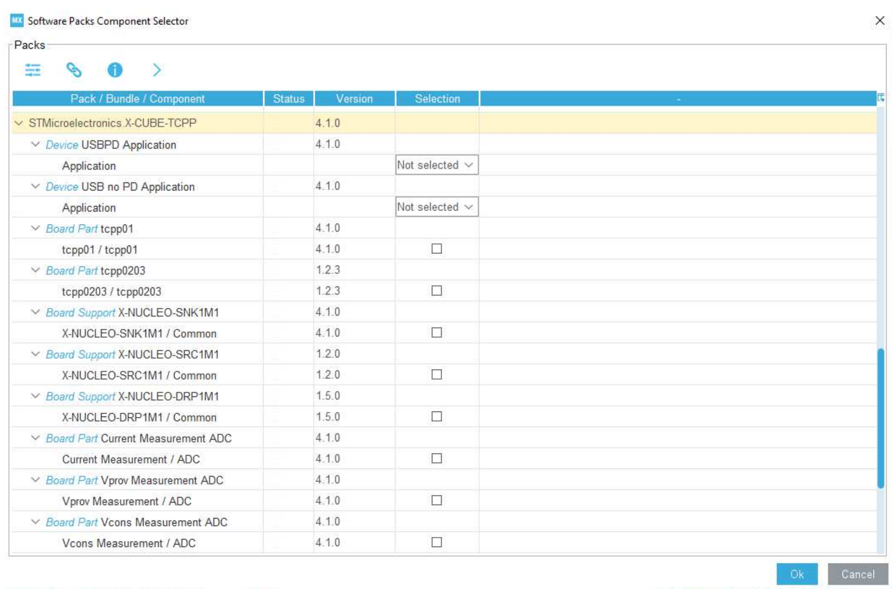


Figure 14 STM32CubeMX Software Packs Component Selector window

Note: in the following, each step description is followed by 3 illustrations, one for each of the applications : Sink, Source and Dual-Role

6.1 USBPD Application using X-Nucleo Extension Boards

The purpose of the chapter is to implement the USBPD Sink, Source or Dual Role solutions using respectively the X-Nucleo SNK1M1, SRC1M1 or DRP1M1 associated with the G0xx or G4xx series Nucleo. This is the quickest and easiest way to start these solutions.

The X-NUCLEO-SNK1M1 or X-NUCLEO-SRC1M1 or X-NUCLEO-DRP1M1 expansion board with a STM32 Nucleo 64 pins board (e.g., a Nucleo-G071RB) no particular hardware modification must be done.

USB Power-Delivery application needs several resources.

- An UCPD peripheral.
- The USBPD middleware.
- The FreeRTOS middleware.
- Optionally, an LPUART peripheral, the Tracer Utility and the GUI Utility will help for debug and protocol understanding.

First, clear the pinout:

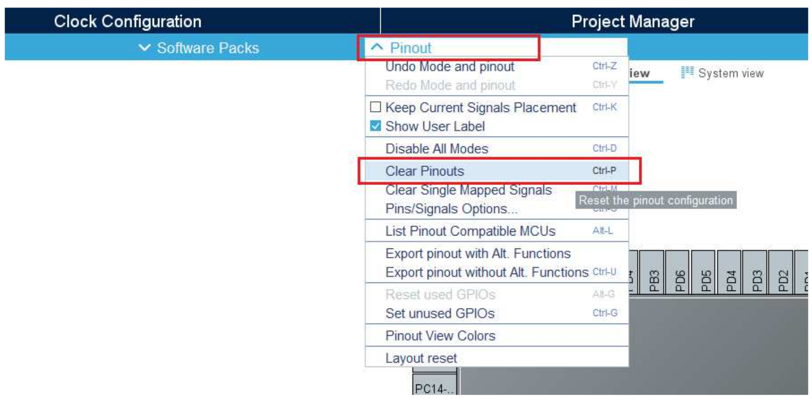


Figure 15 Clear pre-defined pinout

6.1.1 Software pack selection

Then select the Software Pack components: From the Software Packs category, press the ‘STMicroelectronics-CUBE-TCPP’ item and enable the chosen application (Sink, Source or Dual-Role).

A warning appears: click on the warning on the ‘STMicroelectronics.X-CUBE-TCPP’ row to show missing items.

Select the related board part using its checkbox:

- ‘TCPP01’ for a sink application.
- ‘TCPP0203’ for a source or a dual-role application.

Select the related board extension using its checkbox:

- ‘X-NUCLEO-SNK1M1’ for a sink application.
- ‘X-NUCLEO-SRC1M1’ for a source application.
- ‘X-NUCLEO-DRP1M1’ for a dual-role application.

Remaining items: ‘USBPD Middleware’, ‘RTOS Middleware’, ‘ADC HAL’ and ‘I2C HAL’ will be selected later. Press ‘OK’ to close this window.

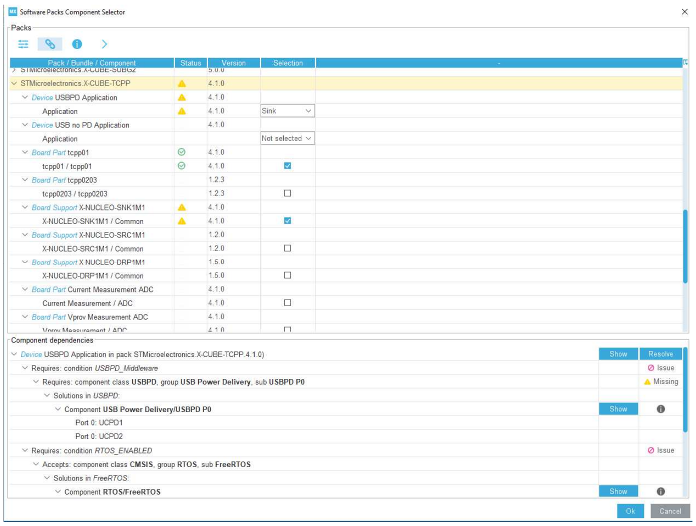


Figure 16 Sink - Software Pack components selection

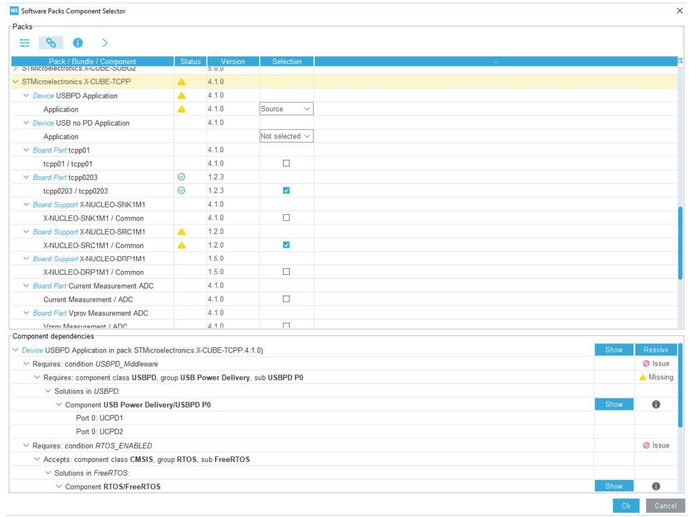


Figure 17 Source - Software Pack components selection.

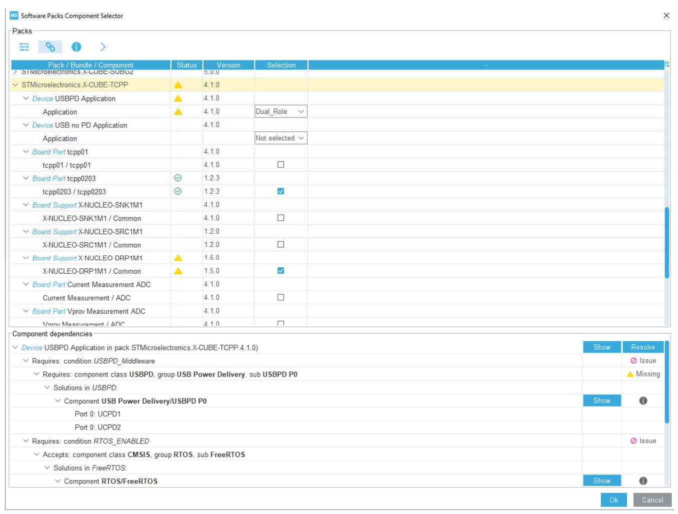


Figure 18 Dual-Role - Software Pack components selection.

Select the ‘Middleware and Software packs’ category. Select the ‘X-CUBE-TCPP’ item. Depending on your application:

- Check the ‘Sink’ application, the ‘TCPP01’ board part and the ‘X-NUCLEO-SNK1M1’ board extension.
- Check the ‘Source’ application, the ‘TCPP0203’ board part and the ‘X-NUCLEO-SRC1M1’ board extension.
- Check the ‘Dual-Role’ application, the ‘TCPP0203’ board part and the ‘X-NUCLEO-DRP1M1’ board extension.

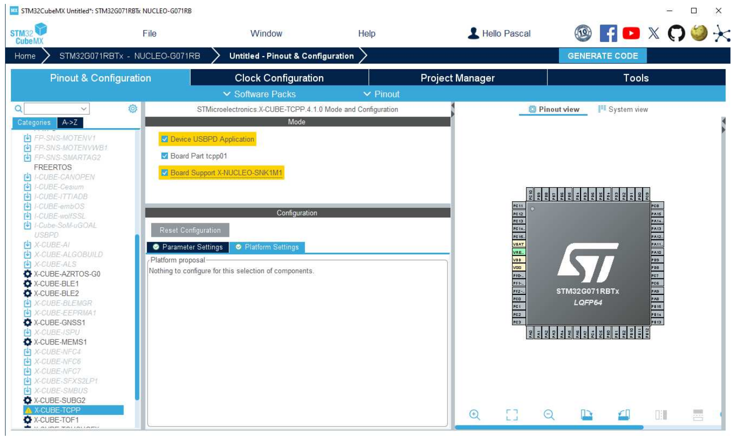


Figure 19 Sink - Software Pack components activation.

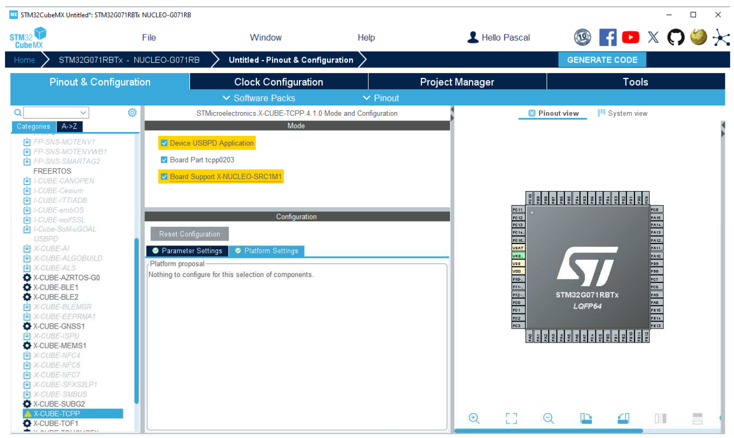


Figure 20 Source - Software Pack components activation.

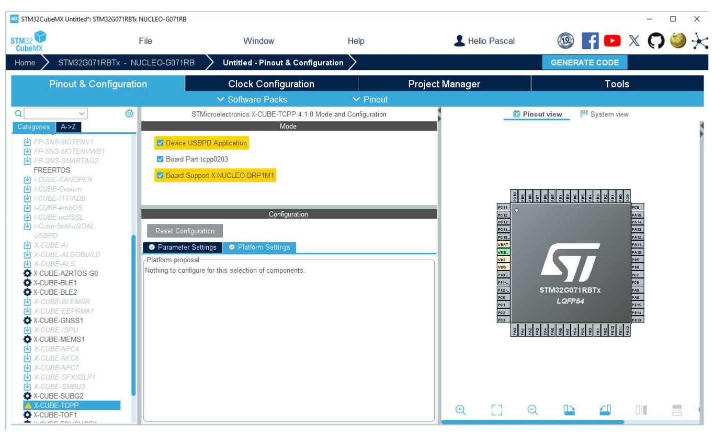


Figure 21 Dual-Role - Software Pack components activation.

6.1.2 UCPD Peripheral activation

Back to STM32CubeMX pinout and configuration window, click on ‘Connectivity’ category and then select an UCPD instance and set it to the chosen mode:

- Sink
- Source
- Dual-Role

In its ‘NVIC Settings’ tab, enable the ‘UCPD global interrupts’.

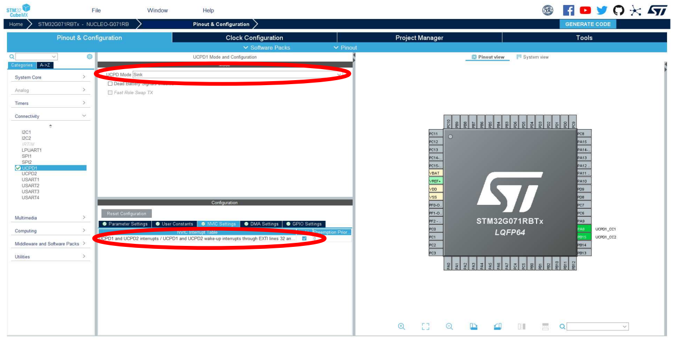


Figure 22 Sink UCPD peripheral selection and configuration.

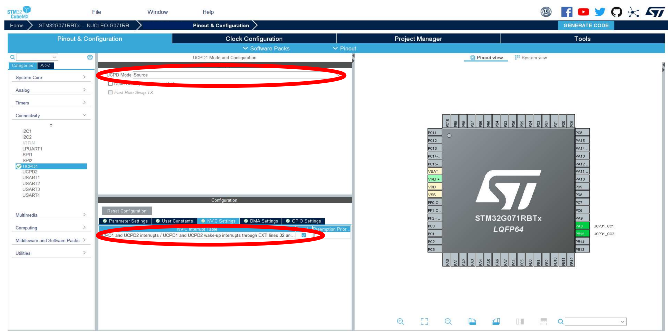


Figure 23 Source UCPD peripheral selection and configuration.

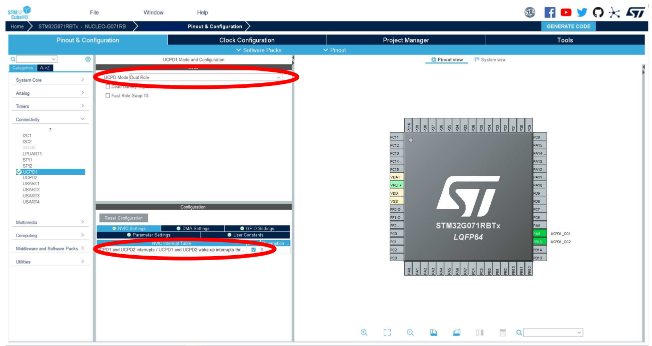


Figure 24 Dual-Role UCPD peripheral selection and configuration.

Note: Select UCPD1 when using a NUCLEO-G071.

6.1.3 DMA Configuration

In its ‘DMA Setting’ tab, add ‘UCPD_TX’ (Channel 4) and ‘UCPD_RX’ (Channel 2) DMA requests.

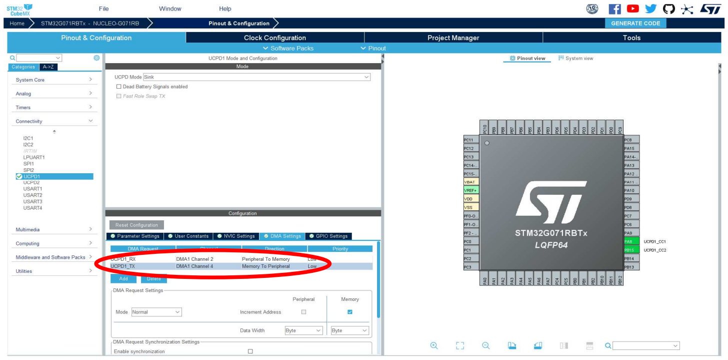


Figure 25 Sink UCPD peripheral DMA configuration

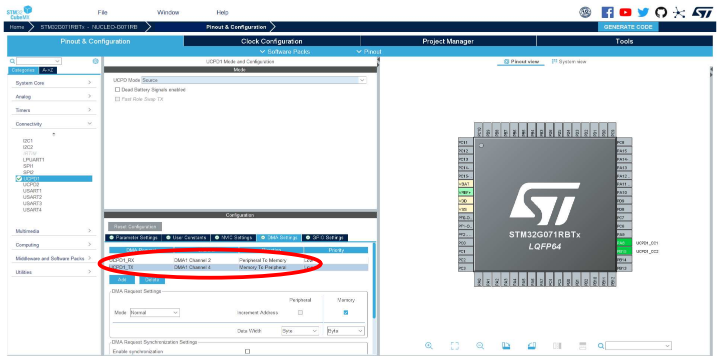


Figure 26 Source UCPD peripheral DMA configuration

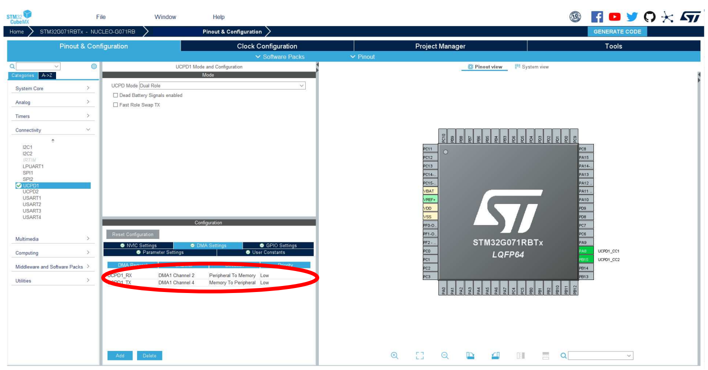


Figure 27 Dual-Role peripheral DMA configuration

6.1.4 USBPD Middleware

From the Pinout & Configuration tab, click on the ‘Middleware and software packs’ category and then select the ‘USBPD Middleware’. Enable it.

And under its ‘PDO General definition’ tab define the Sink, Source or DRP PDOs. Sink and Source PDO definition are in the following tables.

For Sink PDO, request at least 10mA current.

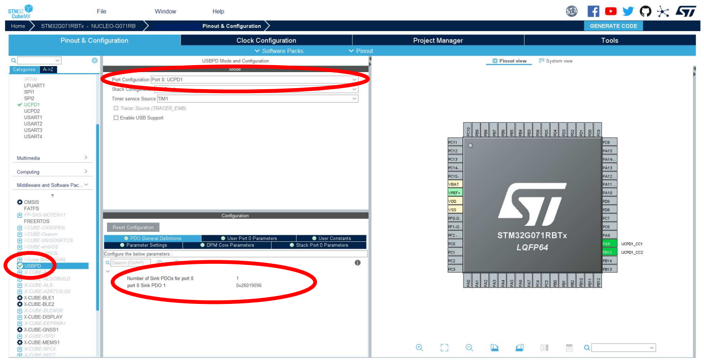


Figure 28 Sink - USBPD Middleware selection, PDO definition.

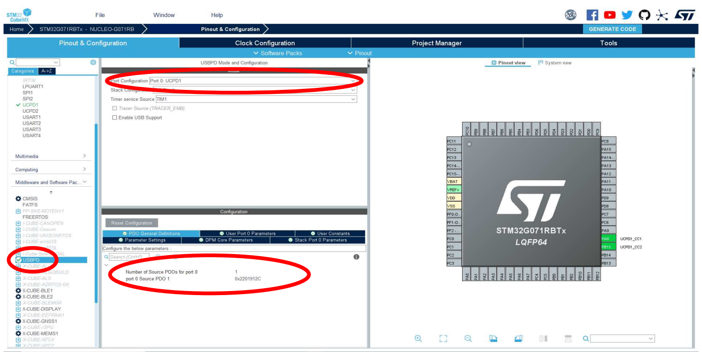


Figure 29 Source - USBPD Middleware selection, PDO definition.

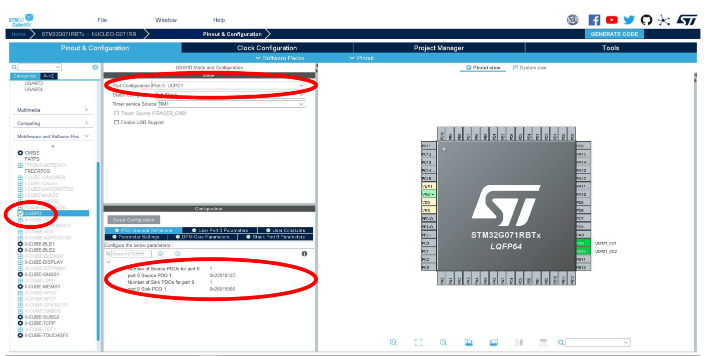


Figure 30 Dual-Role - USBPD Middleware selection, PDO definition.

Bit(s)	Description	Used value	Decoding
B31..30	Fixed supply	00b	Fixed
B29	Dual-Role Power	0b	No
B28	USB Suspend Supported	0b	No
B27	Unconstrained Power	0b	No
B26	USB Communications Capable	0b	No
B25	Dual-Role Data	1b	Yes
B24	Unchunked Extended Messages Supported	0b	No
B23	EPR Mode Capable	0b	No
B22	Reserved – Shall be set to zero	0b	No
B21..20	Peak Current	00b	Peak Equals
B19..10	Voltage in 50mV units	0001100100b	5V
B9..0	Maximum current in 10mA units	0100101100b	3A

Table 1 Sink - Fixed Supply PDO

Bit(s)	Description	Used value	Decoding
B31..30	Fixed supply	00b	Fixed
B29	Dual-Role Power	0b	No
B28	USB Suspend Supported	0b	No
B27	Unconstrained Power	0b	No
B26	USB Communications Capable	0b	No
B25	Dual-Role Data	0b	No
B24..22	Reserved – Shall be set to zero	0b	No
B21..20	Peak Current	00b	Peak Equals
B19..10	Voltage in 50mV units	0001100100b	5V
B9..0	Maximum current in 10mA units	0100101100b	3A

Table 2 Source – Fixed Supply PDO

For a **Dual-Role solution only**, in the Stack Port 0 parameters section, enable the CAD role toggle: CAD role toggle: Supported

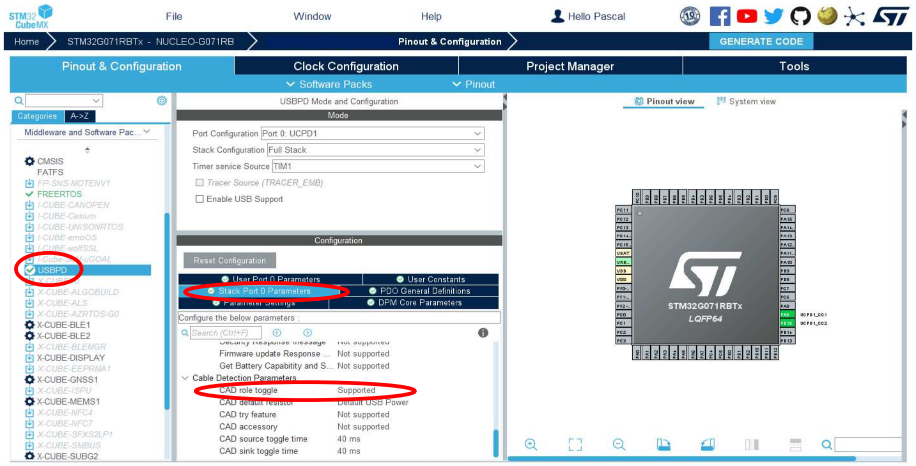


Figure 31 Dual-Role - USBPD Middleware selection, CAD role Toggle.

From the Pinout & Configuration tab, select the ‘FreeRTOS’ Middleware in the ‘Middleware and Software packs’ category. Enable it in ‘CMSIS_V1’ mode, and change ‘TOTAL HEAP SIZE’ to 7000.

Note: If an STM32G4 is used of a G0, LIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY needs to be set to 3 instead of CubeMX's default value 5. In some cases with STM32G4, leaving it to 5 will get the code execution stuck in vPortValidateInterruptPriority function.

6.1.5 RTOS

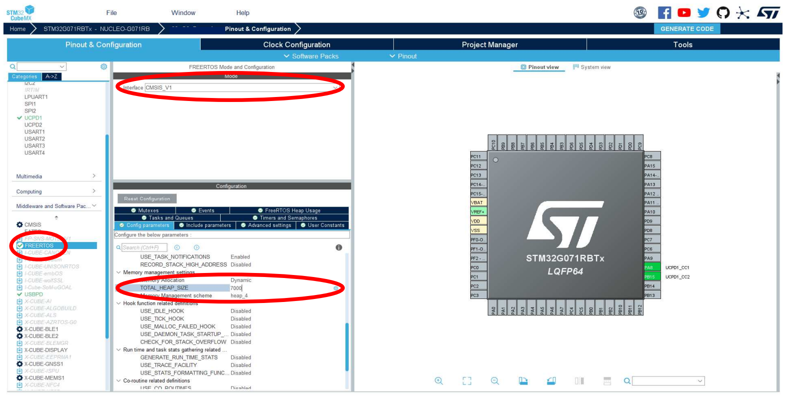


Figure 32 FreeRTOS Middleware selection.

Then under the ‘Include parameters’ tab, if it not the default case, enable ‘eTaskGetState’ in the included definitions.

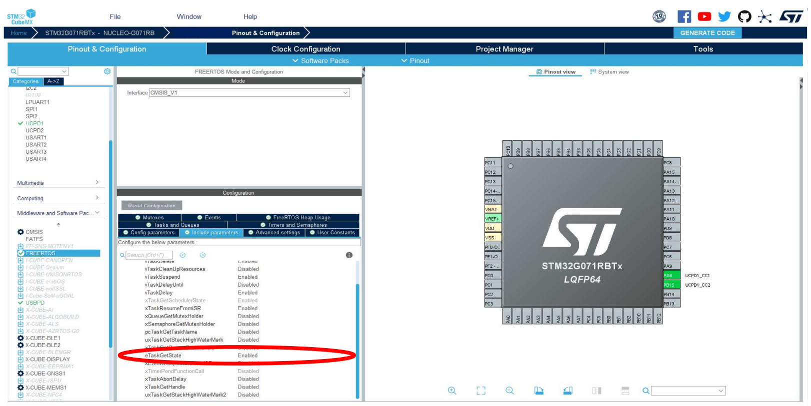


Figure 33 FreeRTOS Middleware configuration.

6.1.6 ADC and I2C Peripheral activation

To download the ADC and I2C HAL library used in the project, activate and ADC and an I2C, no matter the configuration or the instance.

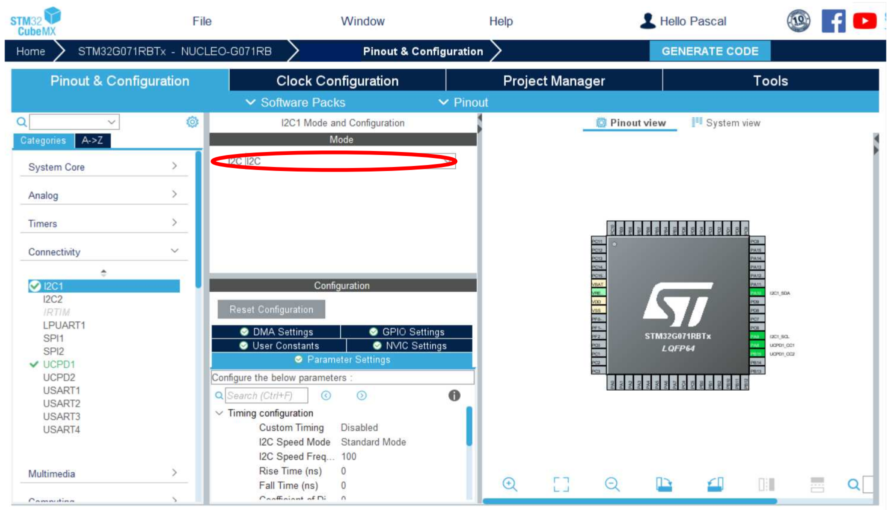


Figure 34 I2C Peripheral selection

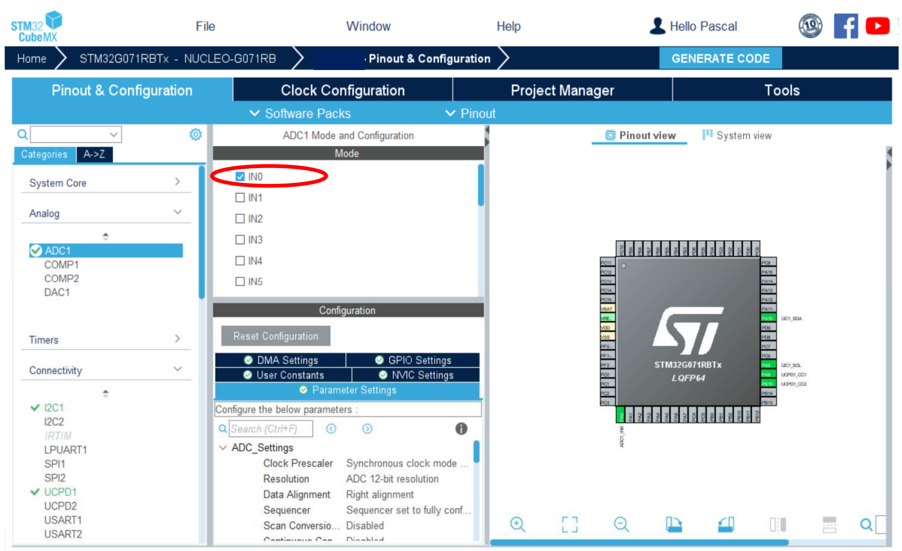


Figure 35 ADC Peripheral selection

6.1.7 Clock configuration

Under Clock Configuration main tab, change system clock mux to PLLCLK. It will set HCLK clock to 64MHz.

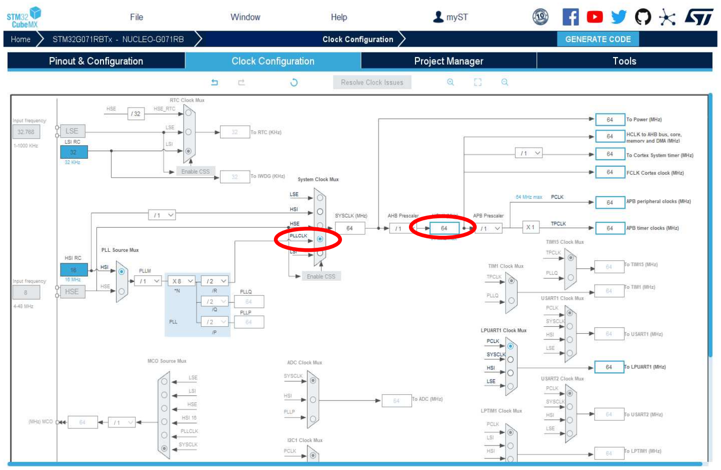


Figure 36 Clock configuration.

6.1.8 Project configuration

Under the Project Manager main tab, configure the minimum stack size to 0xC00 under the Project tab. This is a first value, which can be tuned later, depending on application needs.

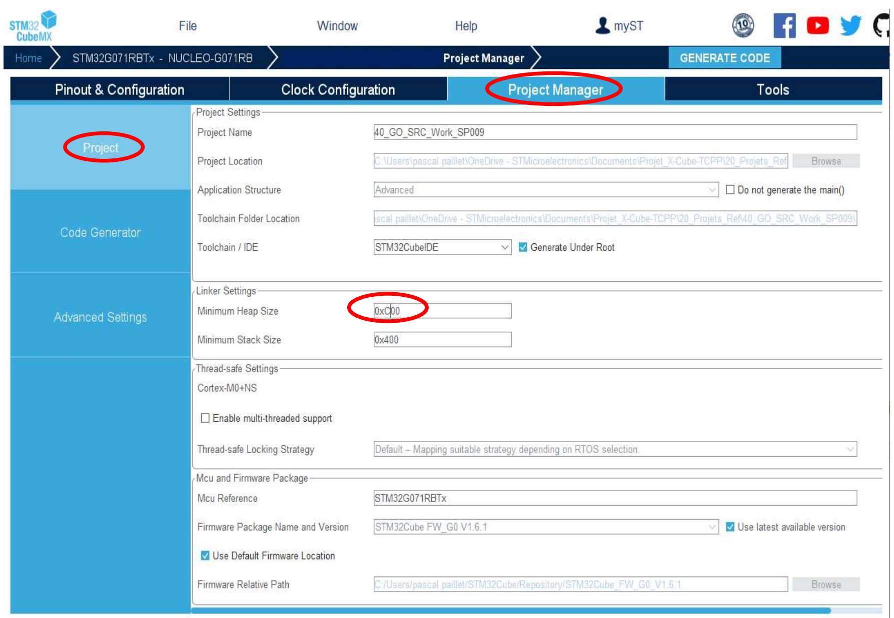


Figure 37 Heap Size configuration.

6.1.9 Tracer and GUI configuration

Following configuration is optional, dedicated to Tracer and GUI, which is useful for debug, and requires a serial interface to USB such as a ST-LINK

From the ‘Connectivity’ tab, enable the LPUART, set its baud rate to 921600 and its word length to 7bits.

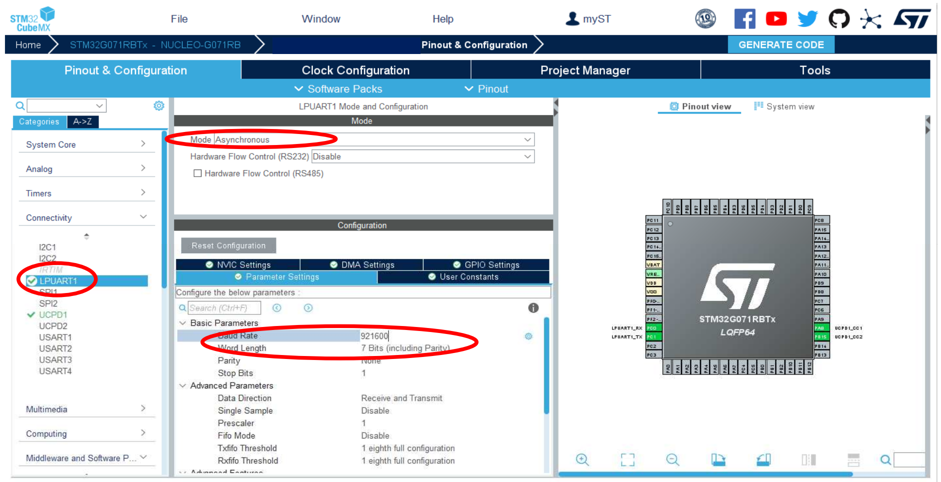


Figure 38 LPUART Parameter Settings

Then select the ‘DMA Settings’ tab and add the ‘LPUART_TX’ DMA, with an empty channel, for example Channel 3.

Note: Channel 1 is already used by the automatic configuration for ADC.

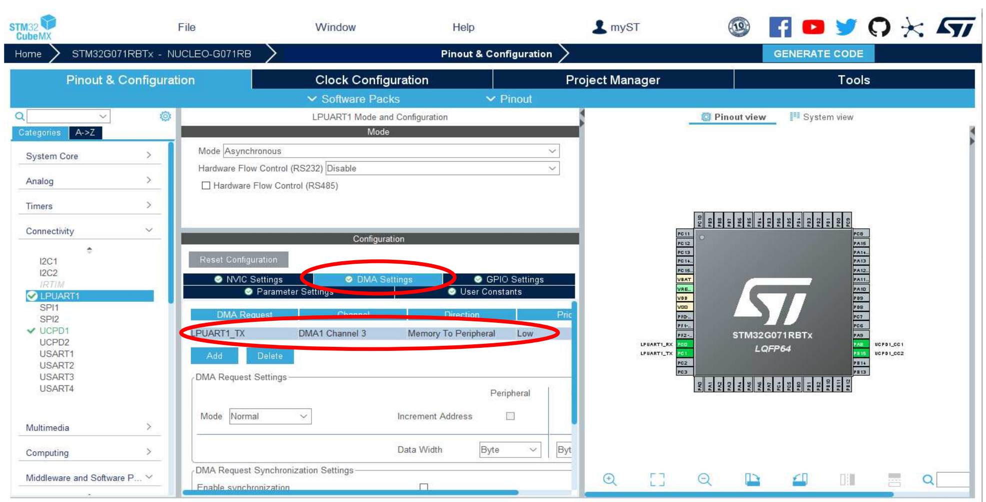


Figure 39 LPUART DMA Settings

Finally, in the ‘NVIC Settings’ enable the LPUART global interrupt.

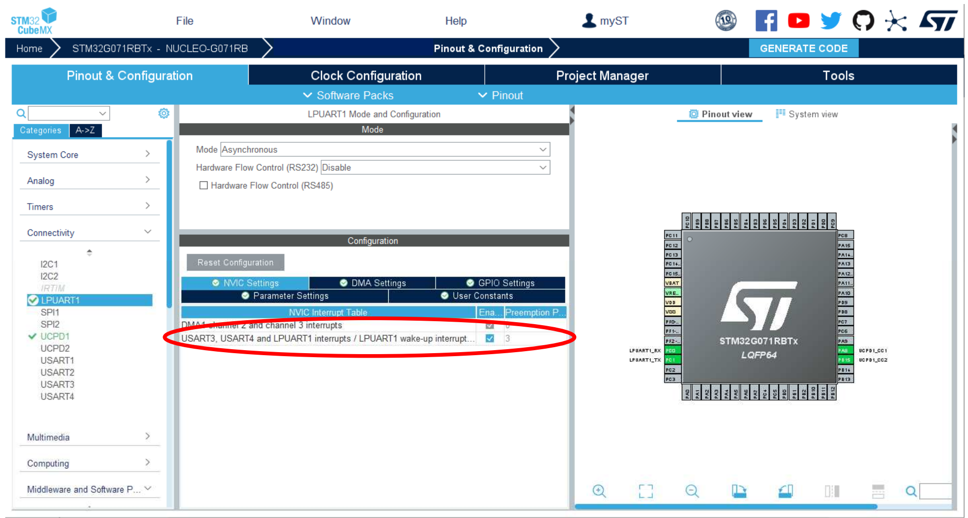


Figure 40 LPUART NVIC Settings.

In the pinout view, click left on PA2 and PA3 to remap them as ‘LPUART1_TX’ and ‘LPUART1_RX’.

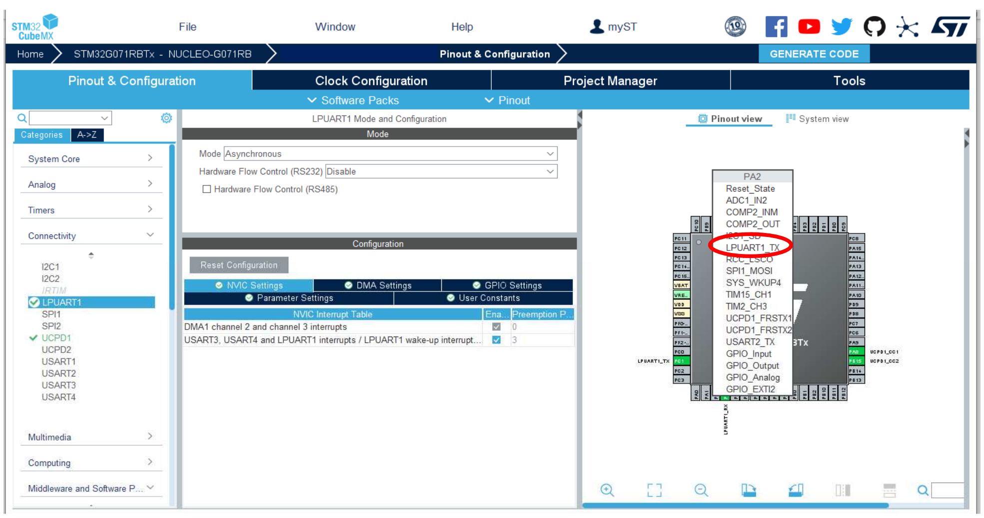


Figure 41 LPUART re-map.

Then in the ‘Utility’ category, enable the ‘Tracer’ and select the LPUART for source.

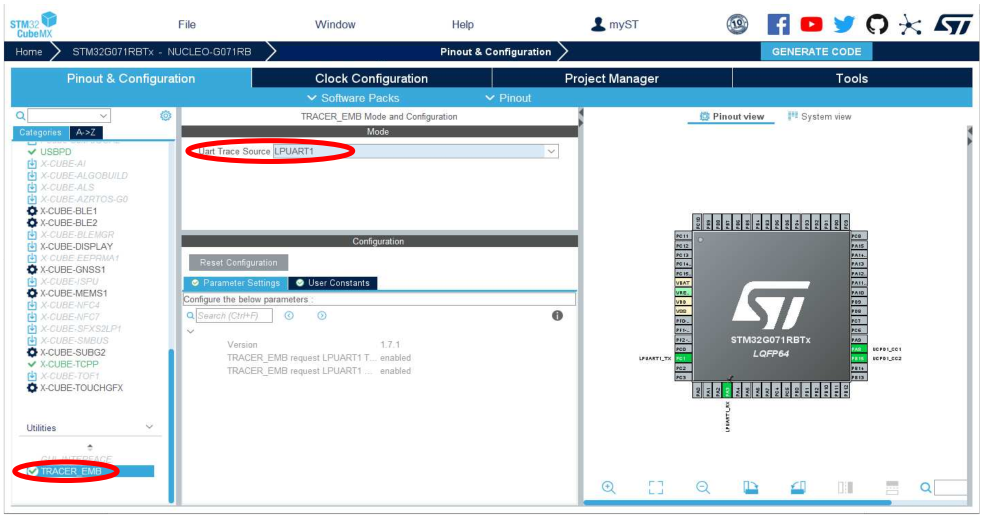


Figure 42 Tracer configuration.

Then in the ‘USBPD Middleware’ check ‘Tracer Source’

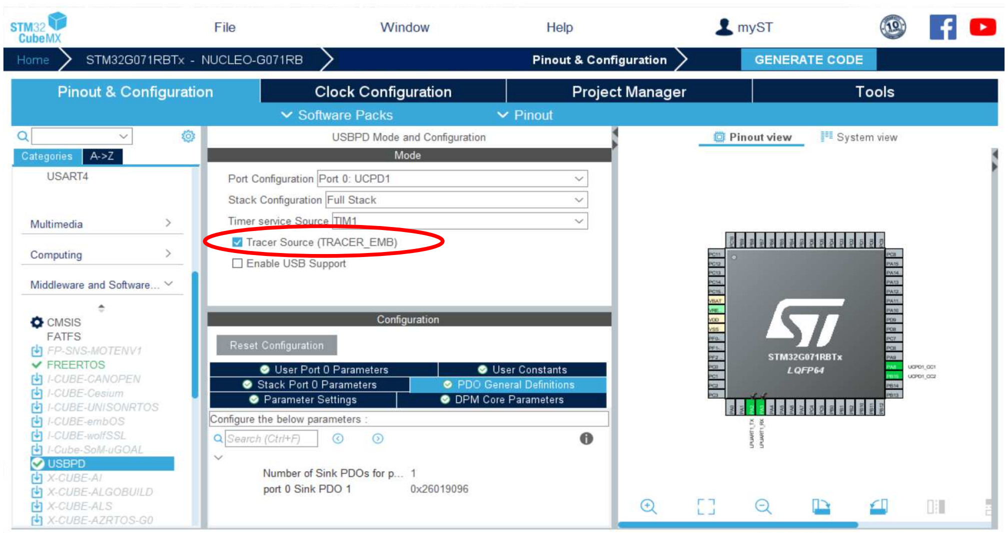


Figure 43 Sink – Tracer source configuration.

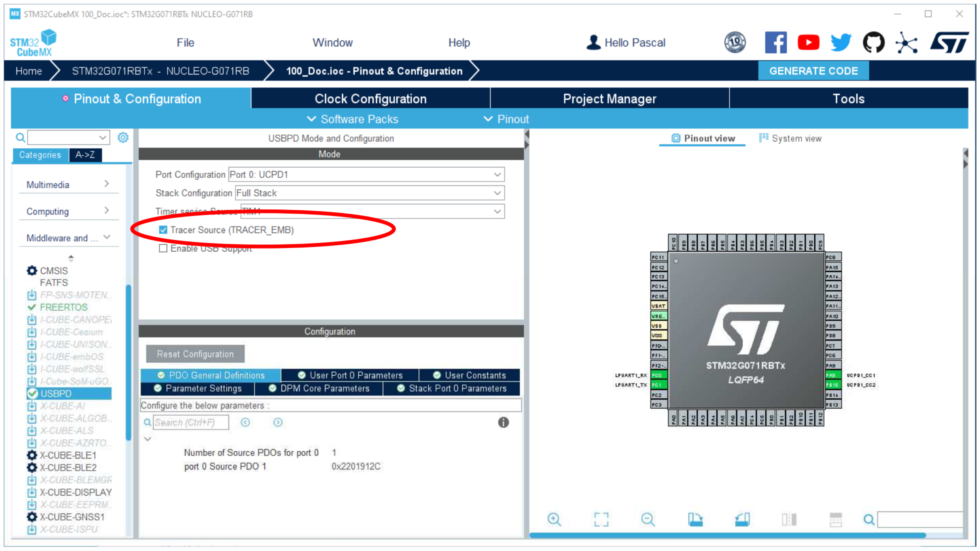


Figure 44 Source – Tracer source configuration.

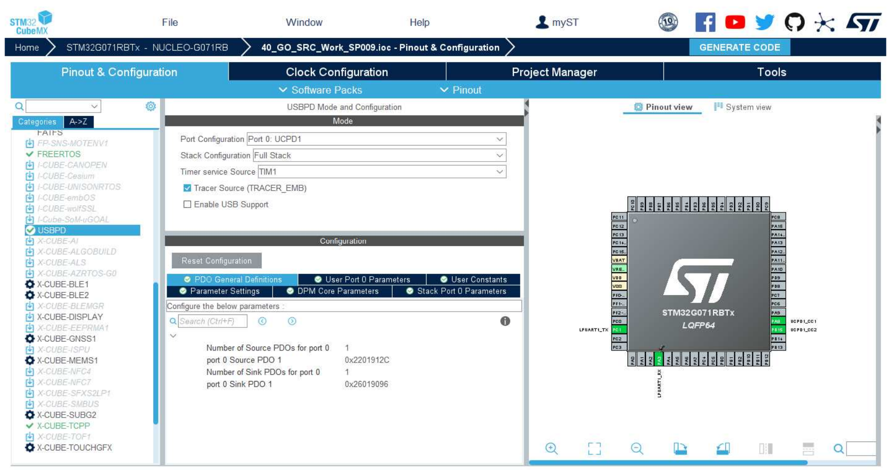


Figure 45 Dual-Role – Tracer source configuration.

Finally, back to the ‘Utility’ category, enable the ‘GUI_Interface’

Set a HWBoardVersionName and a PDTypeName. For example, ‘NUCLEO-G071RB’ and ‘SNK1M1’ (Name of the Sink solution and the NUCLEO-G071RB Board but all names are convenient).

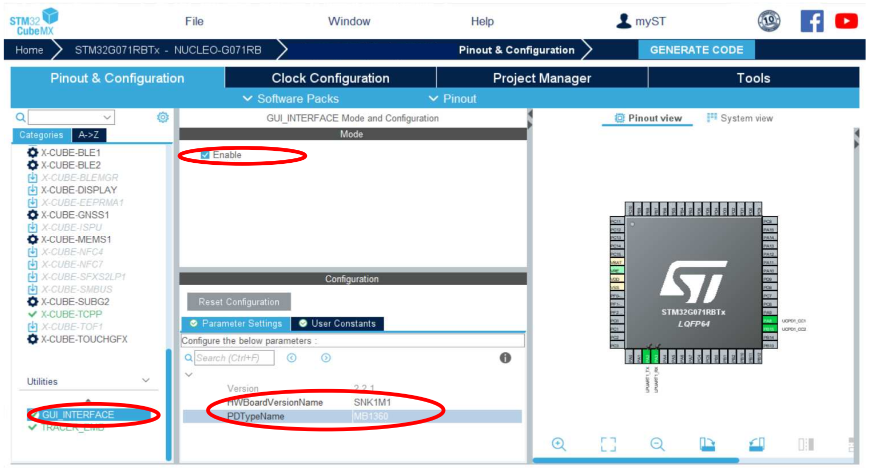


Figure 46 Sink – GUI_Interface configuration

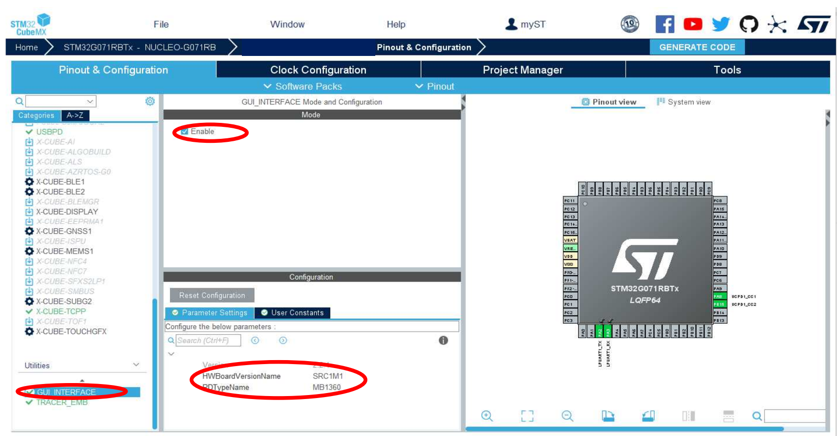


Figure 47 Source – GUI_Interface configuration

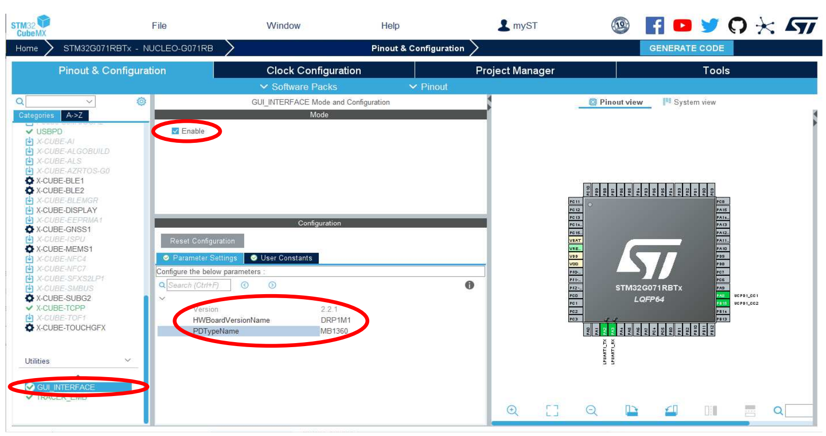


Figure 48 Dual-Role – GUI_Interface configuration

6.1.10 Project build

Generate the code for your IDE, for example STM32CubeIDE:

In the advanced Settings of the “Project Manager” tab, as we do not need ADC and I2C initialization functions (handled by the BSP drivers), uncheck Generate Code for the MX_I2C2_Init and MX_ADC1_Init functions.

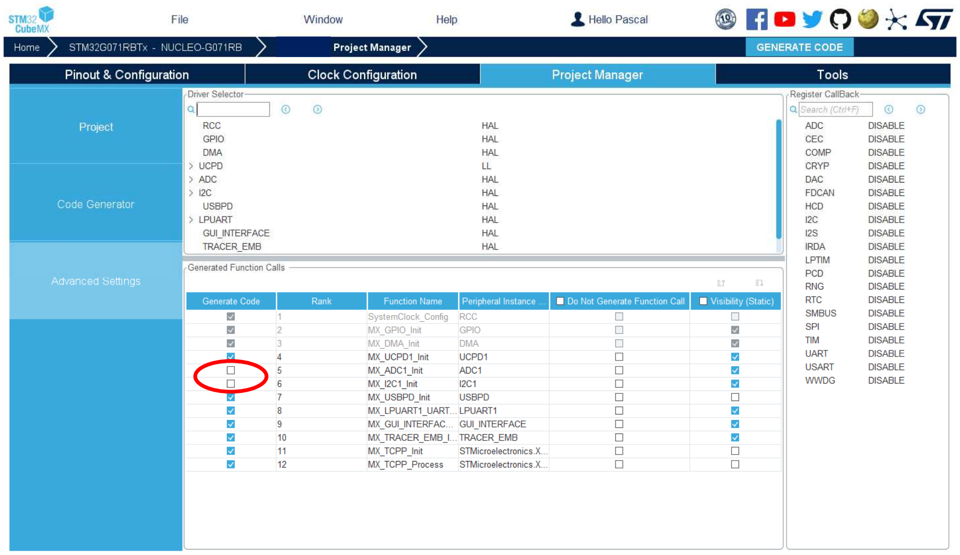


Figure 49 Uncheck ADC and I2C Initialization functions generation

Save your file with Ctrl+s and select generate code if prompted. You can also generate code from the STM32CubeIDE menu, clicking on Project/Generate Code, or by pressing Alt+K.

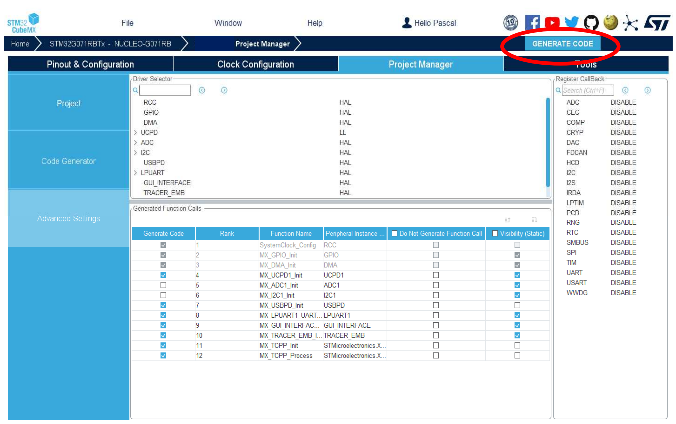


Figure 50 Code Generation

Following warning is usual, click on “Yes”.

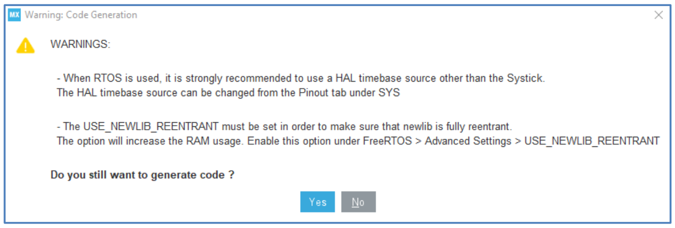


Figure 51 Warning

In your IDE, compile the code. It should be without any warning or error. And flash it in your application.

- The compilation must be performed without error or warnings.
- Build the application by clicking on the button (or select Project/Build Project).
- Run the application by clicking on the button (or select Run/Run)

6.1.11 Application test

If Tracer and GUI have been configured, using STCubeMonitor for debug:

With your application running on the board, launch the STM32CubeMonitor-UCPD application. The user's board must appear in the list when clicking ‘Refresh list of connected boards’, so double-click on the corresponding line (or click ‘NEXT’).

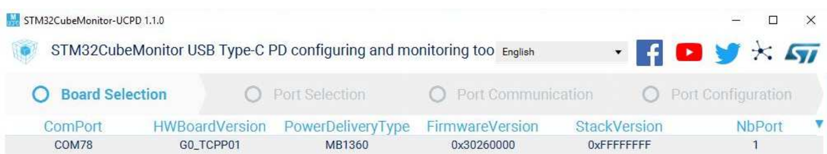


Figure 52 Sink - STM32CubeMonitor Connection

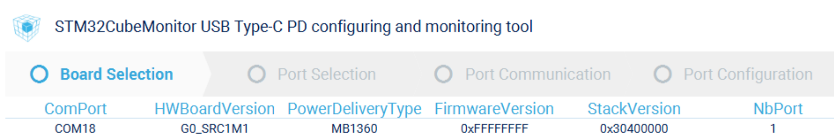


Figure 53 Source - STM32CubeMonitor Connection

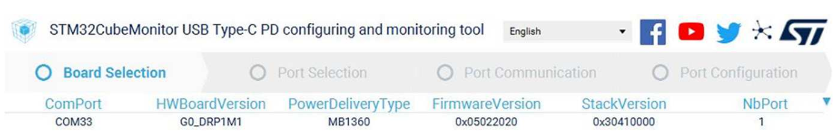


Figure 54 Dual-Role - STM32CubeMonitor Connection

Note: The ComPort may be different. It depends on the number of boards installed on the computer. Then double-click on the desired UCPD port, here Port 0, or select it and click ‘NEXT’.

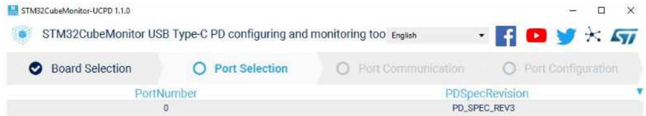


Figure 55 STM32CubeMonitor Port Selection

Click on the TRACES button in the bottom right corner to get protocol traces. Here below is the example of a Dual-Role application, that covers all 3 cases: Sink, Source and DRP:

You can then plug a power delivery sink into the USB Type-C® receptacle of the X-NUCLEO-DRP1M1 shield. The screen may look like this:

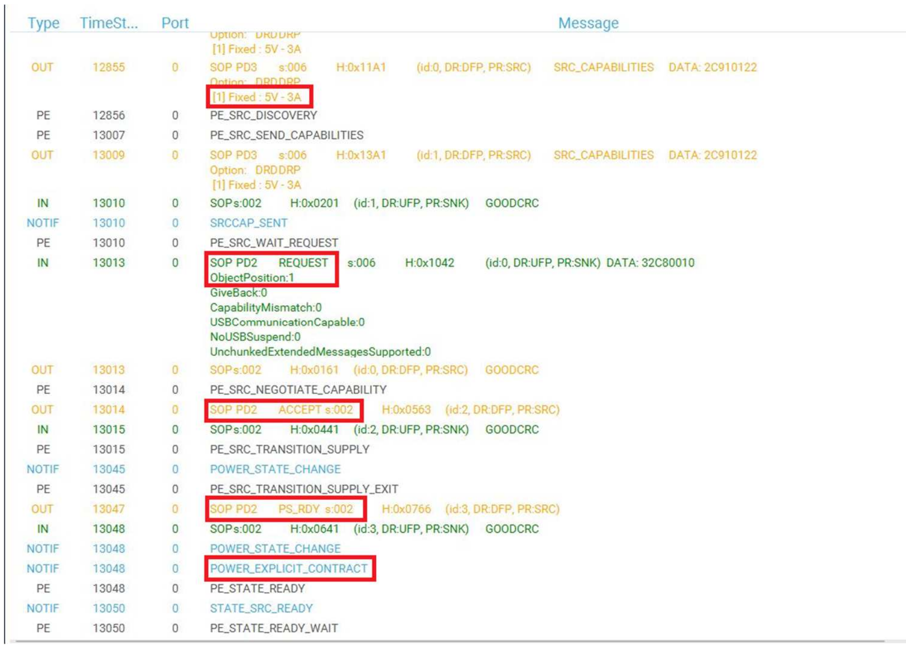
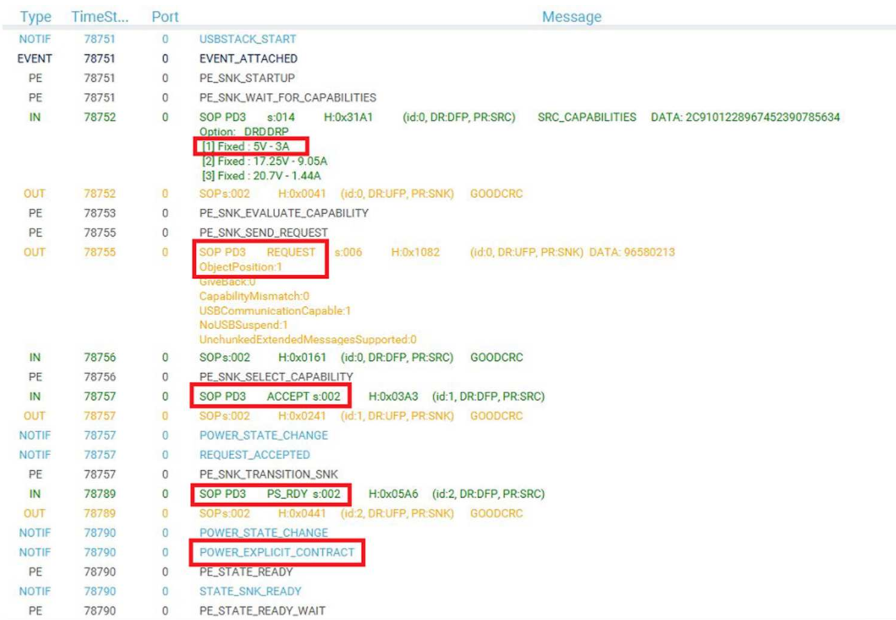


Figure 56 Dual-Role - STM32CubeMonitor Communication

The figure above shows the communication between the STM32G0 and the power delivery sink on the right panel. It is possible to verify the correct sequence to reach an explicit contract:

- The capabilities are sent by the STM32G0 DRP (OUT orange message).
- The request is sent by the sink (IN green message).
- The ACCEPT and the PS_RDY are sent by the STM32G0 source (OUT orange message).
- The contract negotiation ends by the POWER_EXPLICIT_CONTRACT notification (blue message).



Type	TimeSt...	Port	Message
NOTIF	78751	0	USBSTACK_START
EVENT	78751	0	EVENT_ATTACHED
PE	78751	0	PE_SNK_STARTUP
PE	78751	0	PE_SNK_WAIT_FOR_CAPABILITIES
IN	78752	0	SOP PD3 s:014 H:0x31A1 (id:0, DR:DFP, PR:SRC) SRC_CAPABILITIES DATA: 2C9101228967452390785634 Option: DRDDRP [1] Fixed : 5V - 3A [2] Fixed : 17.25V - 9.05A [3] Fixed : 20.7V - 1.44A
OUT	78752	0	SOPs:002 H:0x0041 (id:0, DR:UFP, PR:SNK) GOODCRC
PE	78753	0	PE_SNK_EVALUATE_CAPABILITY
PE	78755	0	PE_SNK_SEND_REQUEST
OUT	78755	0	SOP PD3 REQUEST s:006 H:0x1082 (id:0, DR:UFP, PR:SNK) DATA: 96580213 ObjectPosition:1 GiveBack:0 CapabilityMismatch:0 USBCommunicationCapable:1 NoUSBSuspend:1 UnchunkedExtendedMessagesSupported:0
IN	78756	0	SOPs:002 H:0x0161 (id:0, DR:DFP, PR:SRC) GOODCRC
PE	78756	0	PE_SNK_SELECT_CAPABILITY
IN	78757	0	SOP PD3 ACCEPT s:002 H:0x03A3 (id:1, DR:DFP, PR:SRC)
OUT	78757	0	SOPs:002 H:0x0241 (id:1, DR:UFP, PR:SNK) GOODCRC
NOTIF	78757	0	POWER_STATE_CHANGE
NOTIF	78757	0	REQUEST_ACCEPTED
PE	78757	0	PE_SNK_TRANSITION_SNK
IN	78789	0	SOP PD3 PS_RDY s:002 H:0x05A6 (id:2, DR:DFP, PR:SRC)
OUT	78789	0	SOPs:002 H:0x0441 (id:2, DR:UFP, PR:SNK) GOODCRC
NOTIF	78790	0	POWER_STATE_CHANGE
NOTIF	78790	0	POWER_EXPLICIT_CONTRACT
PE	78790	0	PE_STATE_READY
NOTIF	78790	0	STATE_SNK_READY
PE	78790	0	PE_STATE_READY_WAIT

Figure 57 Dual-Role - STM32CubeMonitor Communication

The figure above shows the communication between the STM32G0 and the power delivery source on the right panel. It is possible to verify the correct sequence to reach an explicit contract:

- The capabilities are sent by the source (IN green message).
- The request is sent by the STM32G0 DRP (OUT orange message).
- The ACCEPT and the PS_RDY are sent by the source (IN green message).
- The contract negotiation ends by the POWER_EXPLICIT_CONTRACT notification (blue message).

For more details on how to use this tool, refer to UM2468. And for more details on the protocol, refer to UM2552. Note that this trace is very helpful for debugging and application development.

You can also use the Measurement window in STM32CubeMonitor-UCPD to display a graph of the measured VBUS voltage and delivered current. Set the sampling period and click start.

Figure 58 STM32CubeMonitor Measurement

6.1.12 List of resources used

As some resources are configured by the software pack, here is the list of all resources used:

- For Sink solution
    - UCPD1 Peripheral with DMA Channels 2 and 4
    - ADC1
    - 1 ADC Channel
        - PB1 for STM32G0xx
        - PB0 for STM32G4xx
    - 2 GPIO Outputs
        - PB6 and PC10 for STM32G0xx
        - PB1 and PC10 for STM32G4xx
- For Source and Dual-Role solutions
    - UCPD1 Peripheral with DMA Channels 2 and 4
    - ADC1 with DMA Channel 1
    - 3 ADC Channels for Source solution
        - PA0, PA1 and PB11 for STM32G0xx
        - PA0, PA1 and PC1 for STM32G4xx
    - 4 ADC Channels for Dual-Role solution
        - PA0, PA1, PA4 and PB11 for STM32G0xx
        - PA0, PA1, PA4 and PC1 for STM32G4xx
    - 1 GPIO Output (PC8)
    - 1 EXTI (EXTI5 – PC5)
    - I2C1 (re-map on PB8 and PB9) for Source and Dual-Role solutions
- For Optional Tracer and GUI
    - LPUART1 (re-map on PA2 and PA3) with DMA TX Channel

6.2 Custom board - Free resources assignment

The objective of this chapter is to extend the previous implementation of the USBPD Sink, Source or Dual Role solutions to all STM32s with the UCPD peripheral, without pinout restriction, therefore with any board.

The USB Power-Delivery Source application needs several resources.

- An UCPD Peripheral
- The USBPD middleware
- An RTOS middleware
- 2 GPIO outputs for TCPP01 (Sink) or 1 GPIO output for TCPP02 & TCPP03 (Source and Dual Role)

- 1 EXTI for TCPP02 & TCPP03 (Source and Dual Role)
- 1 I2C bus for TCPP02 & TCPP03 (Source and Dual Role)
- At least 1 ADC input for Vbus voltage

If you are using a Nucleo, first, clear the pre-defined pinout:

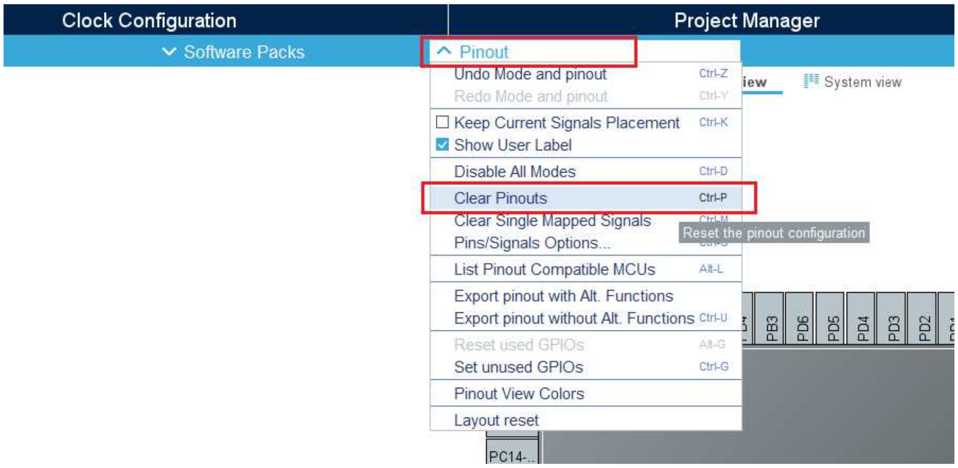


Figure 59 Clear pre-defined pinout

6.2.1 Software pack selection

Then select the Software Pack components: From the Software Packs category, press the 'STMicroelectronics-CUBE-TCPP' item and enable the chosen application (Sink, Source or Dual-Role).

A warning appears: click on the warning on the 'STMicroelectronics.X-CUBE-TCPP' row to show missing items.

Select the related board part using its checkbox:

- 'TCPP01' for a sink application.
- 'TCPP02' for a source application.
- 'TCPP03' for a dual-role application.

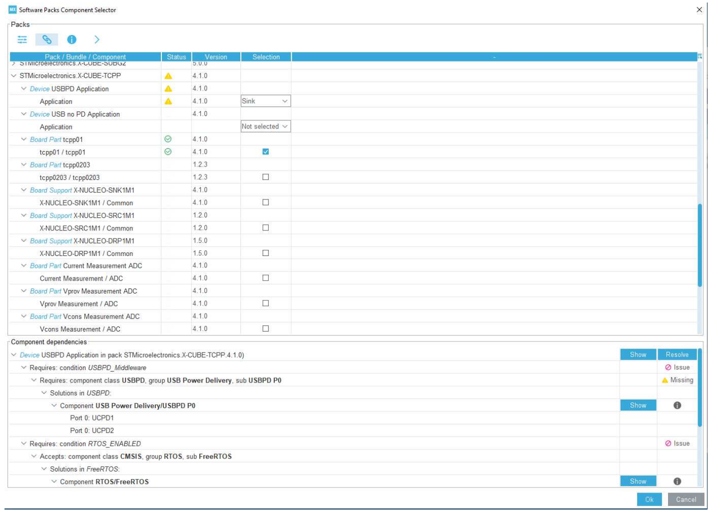


Figure 60 Sink – Software pack components selection

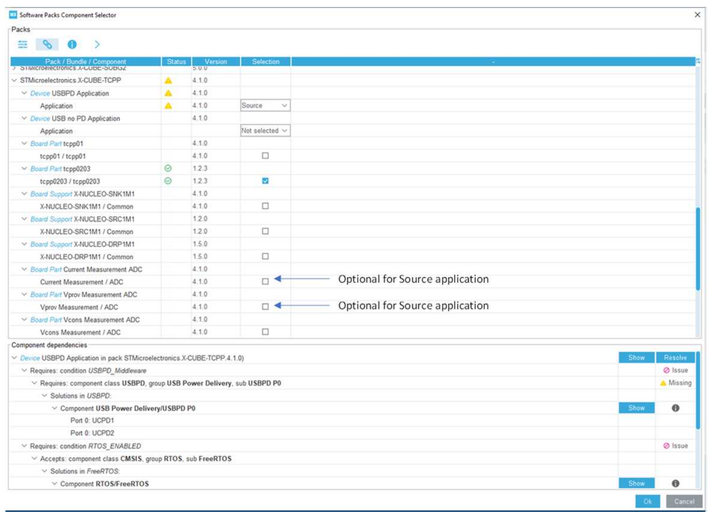


Figure 61 Source – Software pack components selection

Note 1: It is possible to monitor the Vbus current by selecting “Isense” ADC.

Note 2: It is possible to monitor the Vprovider voltage by selecting “Vprov” ADC.

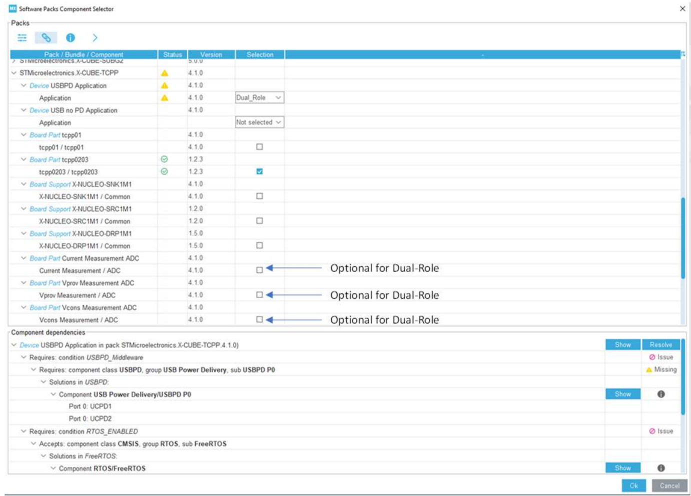


Figure 62 Dual-Role – Software pack components selection

Note 1: It is possible to monitor the Vbus current by selecting “Isense” ADC.

Note 2: It is possible to monitor the Vprovider voltage by selecting “Vprov” ADC.

Note 3: It is possible to monitor the Vconsumer voltage by selecting “Vcons” ADC.

Remaining items: ‘USBPD Middleware’ and ‘RTOS’ will be selected later. Press ‘OK’ to close this window.

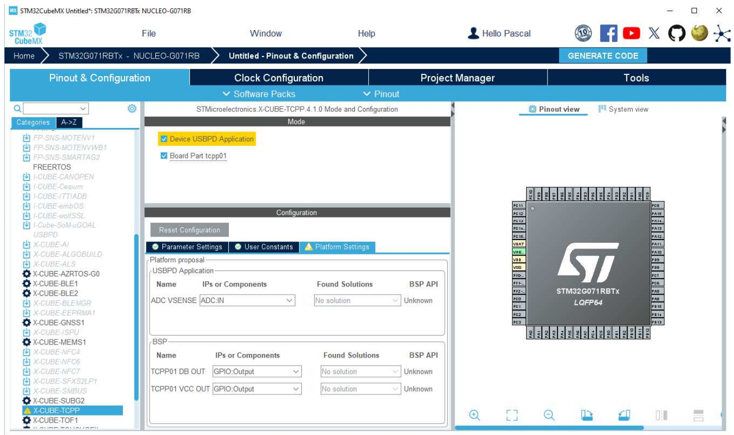


Figure 63 Sink – Software pack components selection

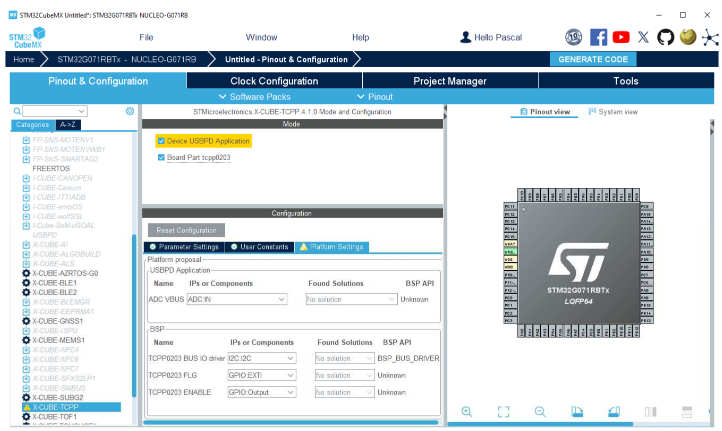


Figure 64 Source – Software pack components activation

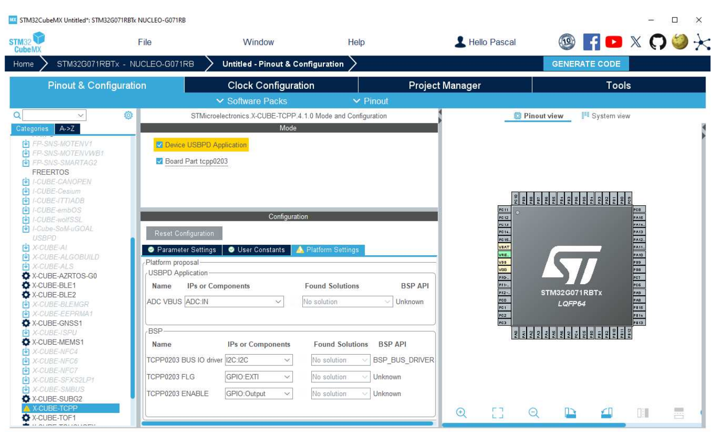


Figure 65 Dual-Role – Software pack components activation

The following chapters consist in these resources configuration to affect it in this ‘Platform Settings’ tab.

6.2.2 UCPD Peripheral activation

Back to STM32CubeMX pinout and configuration window, click on ‘Connectivity’ category and then select an UCPD instance and set it to the chosen mode:

- Sink
- Source
- Dual-Role

In its ‘NVIC Settings’ tab, enable the ‘UCPD global interrupts’.

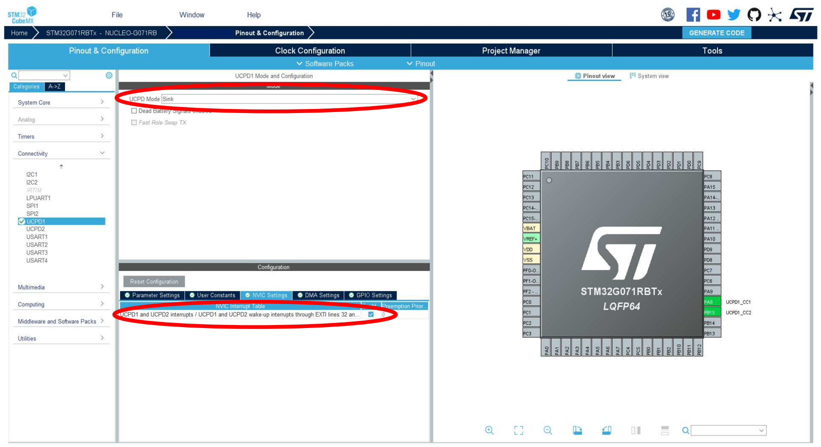


Figure 66 Sink - UCPD peripheral selection and configuration.

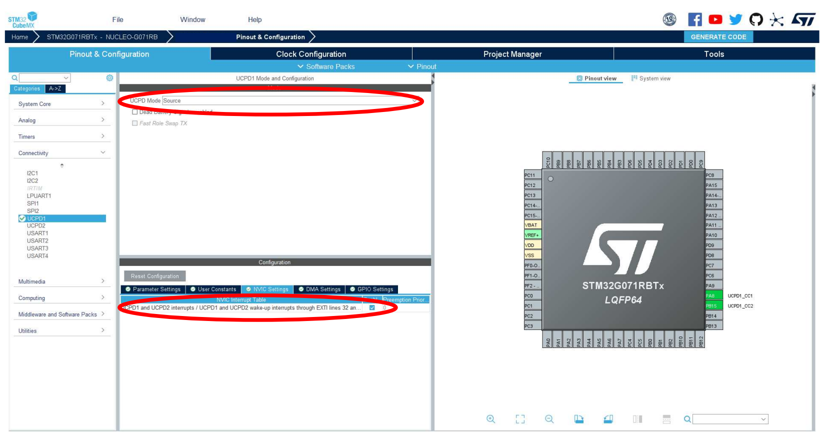


Figure 67 Source - UCPD peripheral selection and configuration.

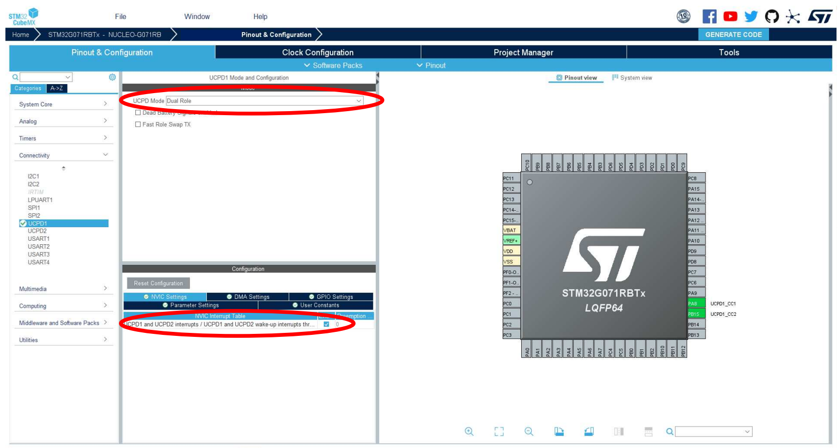


Figure 68 Dual-Role - UCPD peripheral selection and configuration.

Direct memory access (DMA) is required for UCPD, if more than 1 ADC Channel (Vbus) is used and if an optional USART for tracer and GUI is enabled. Depending on STM32 MCU it can be DMA or GPDMA. The following two chapters describe the configuration for each case.

6.2.3 DMA Configuration

In the UCPD '*DMA Setting*' tab, add '*UCPD_TX'* and '*UCPD_RX'* DMA requests.

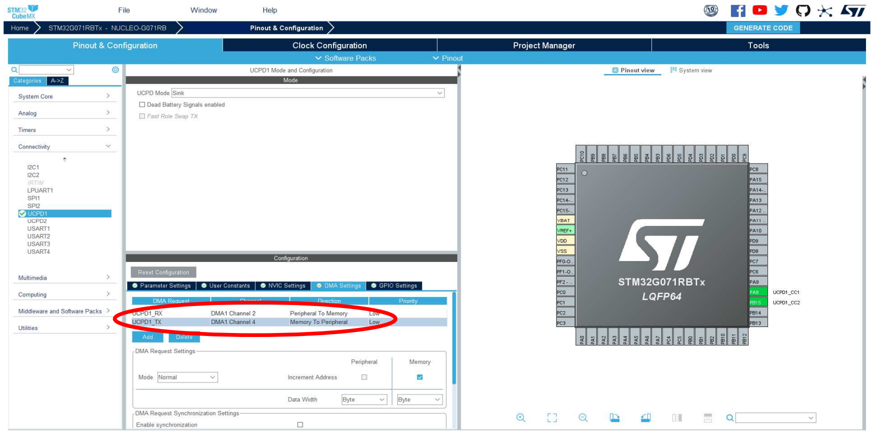


Figure 69 Sink UCPD peripheral DMA configuration.

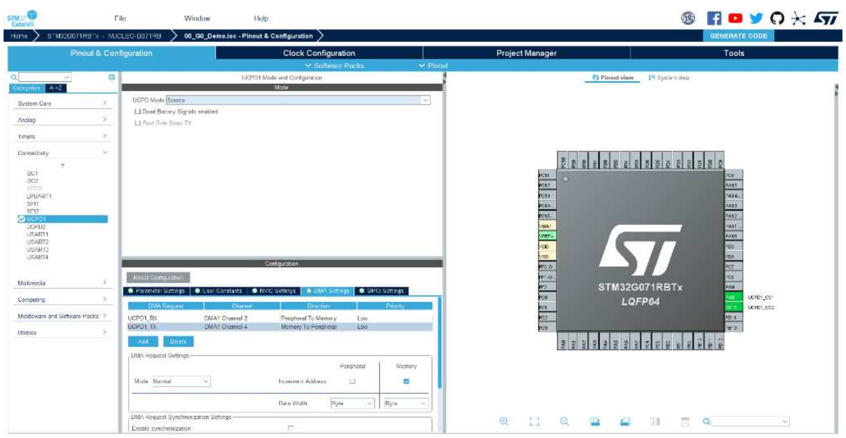


Figure 70 Source UCPD peripheral DMA configuration

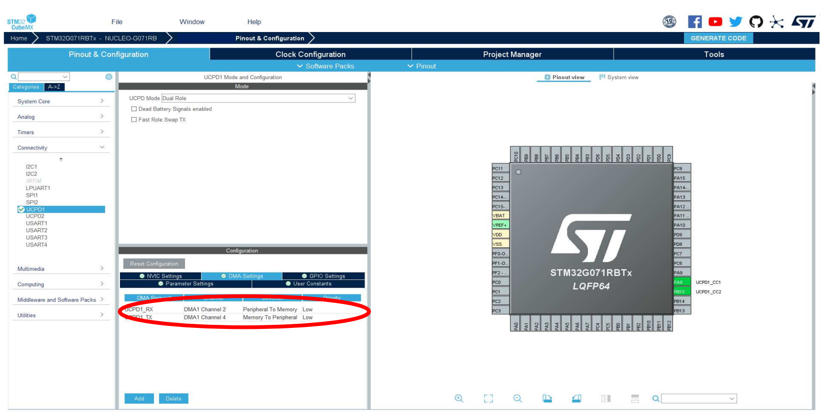


Figure 71 Dual-Role peripheral DMA configuration

6.2.4 GPDMA Configuration

In the GPDMA category, enable 2 ‘GPDMA Channels’ in ‘Standard Request Mode’

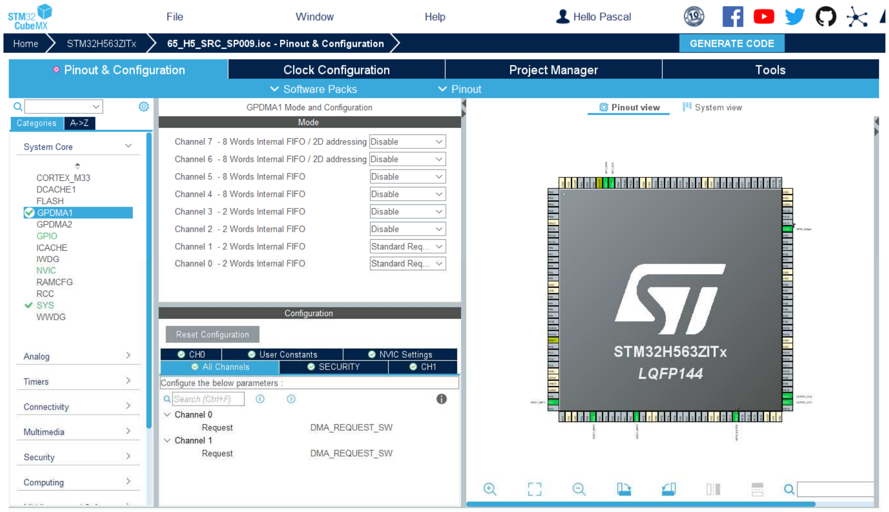


Figure 72 GPDMA Channel activation

Then select ‘Channel 0’ for example and assign it ‘UCPD1_TX’ in ‘Memory to Peripheral’ direction and enable ‘Source Address Increment’.

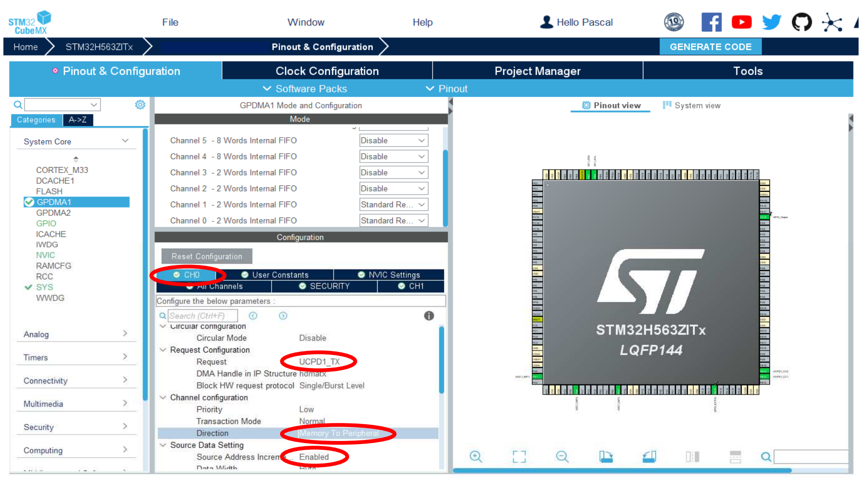


Figure 73 GPDMA Channel assignment to UCPD-TX

Repeat the operation for ‘Channel 1’ for example and assign it ‘UCPD1_RX’ but, in ‘Peripheral to Memory’ direction and enable ‘Destination Address Increment’.

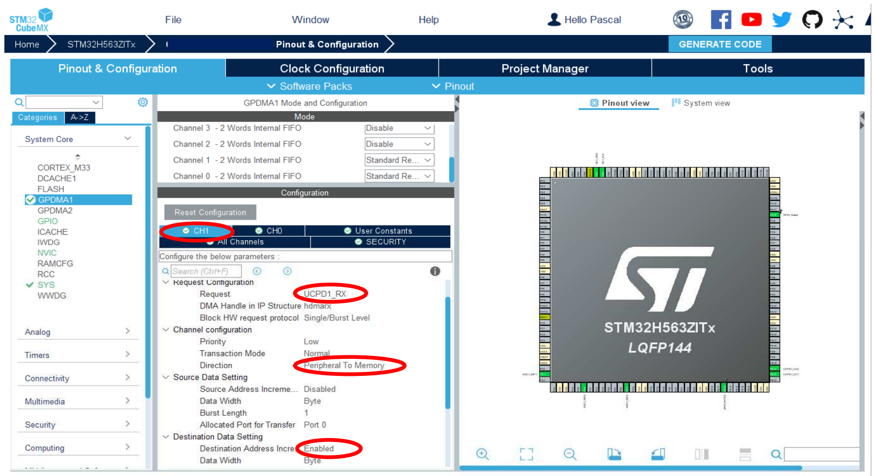


Figure 74 GPDMA Channel configuration

6.2.5 USBPD Middleware

From the Pinout & Configuration tab, click on the ‘Middleware and software packs’ category and then select the ‘USBPD Middleware’. Enable it.

And under its ‘PDO General definition’ tab define the Sink, Source or DRP PDOs. Sink and Source PDO definition are in the following tables.

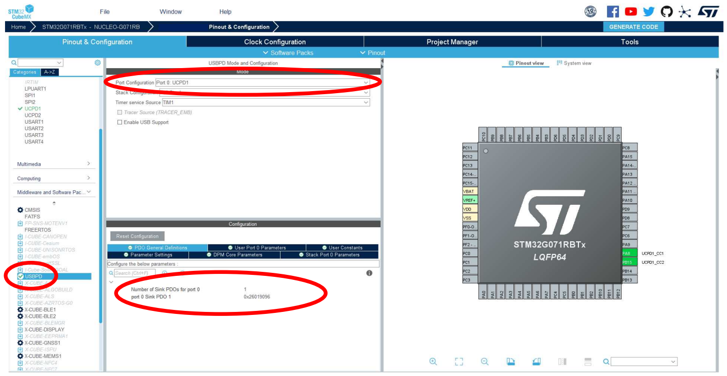


Figure 75 Sink - USBPD Middleware selection, PDO definition.

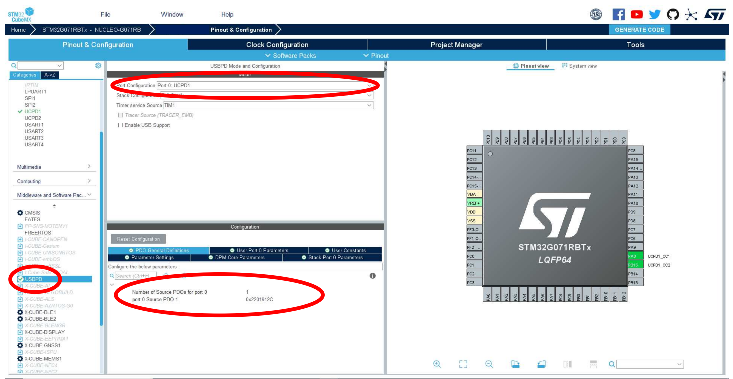


Figure 76 Source - USBPD Middleware selection, PDO definition.

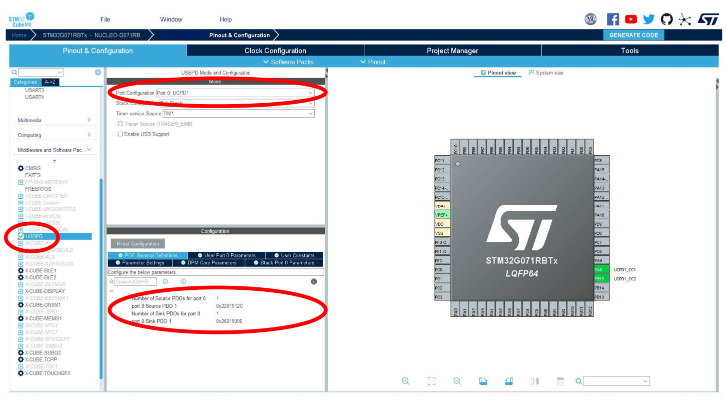


Figure 77 Dual-Role - Middleware selection, PDO definition.

Bit(s)	Description	Used value	Decoding
B31..30	Fixed supply	00b	Fixed
B29	Dual-Role Power	0b	No
B28	USB Suspend Supported	0b	No
B27	Unconstrained Power	0b	No
B26	USB Communications Capable	0b	No
B25	Dual-Role Data	1b	Yes
B24	Unchunked Extended Messages Supported	0b	No
B23	EPR Mode Capable	0b	No
B22	Reserved – Shall be set to zero	0b	No
B21..20	Peak Current	00b	Peak Equals
B19..10	Voltage in 50mV units	0001100100b	5V
B9..0	Maximum current in 10mA units	0100101100b	3A

Table 3 Sink - Fixed Supply PDO

Bit(s)	Description	Used value	Decoding
B31..30	Fixed supply	00b	Fixed
B29	Dual-Role Power	0b	No
B28	USB Suspend Supported	0b	No
B27	Unconstrained Power	0b	No
B26	USB Communications Capable	0b	No
B25	Dual-Role Data	0b	No
B24..22	Reserved – Shall be set to zero	0b	No
B21..20	Peak Current	00b	Peak Equals
B19..10	Voltage in 50mV units	0001100100b	5V
B9..0	Maximum current in 10mA units	0100101100b	3A

Table 4 Source – Fixed Supply PDO

For a Dual-Role solution only, in the Stack Port 0 parameters section, enable the CAD role toggle: CAD role toggle: Supported

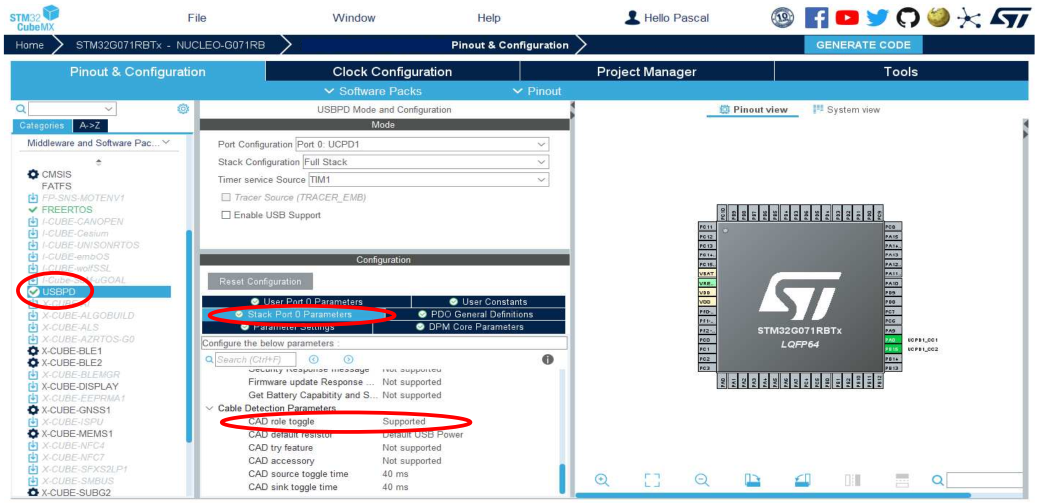


Figure 78 Dual-Role - USBPD Middleware selection, CAD role Toggle.

6.2.6 RTOS

Depending on MCU all RTOS may not be available, when available choose FreeRTOS as described below, else choose ThreadX from AzurRTOS, as described just after.

6.2.6.1 FreeRTOS

From the Pinout & Configuration tab, select the ‘FreeRTOS’ Middleware in the ‘Middleware and Software packs’ category. Enable it in ‘CMSIS_V1’ mode, and change ‘TOTAL HEAP SIZE’ to 7000.

Note: for STM32G4xx series, reduce the ‘LIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY’ from default value 5 down to 3.

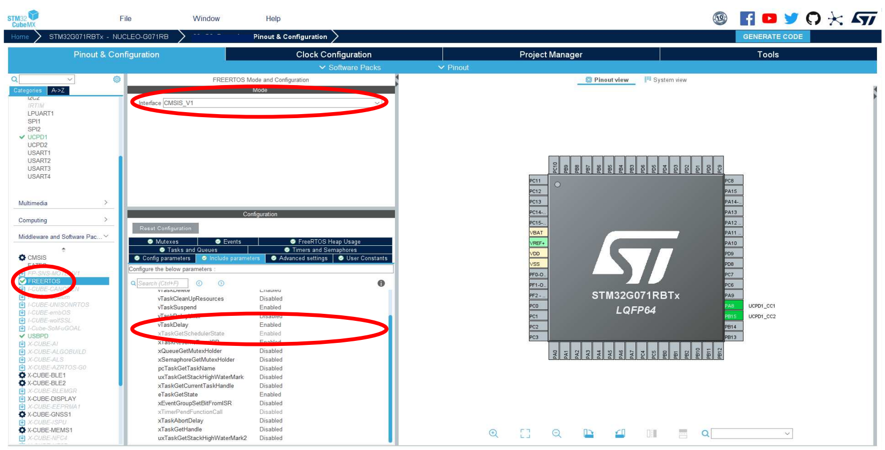


Figure 79 FreeRTOS Middleware selection.

Then under the ‘Include parameters’ tab, enable ‘eTaskGetState’ in the included definitions.

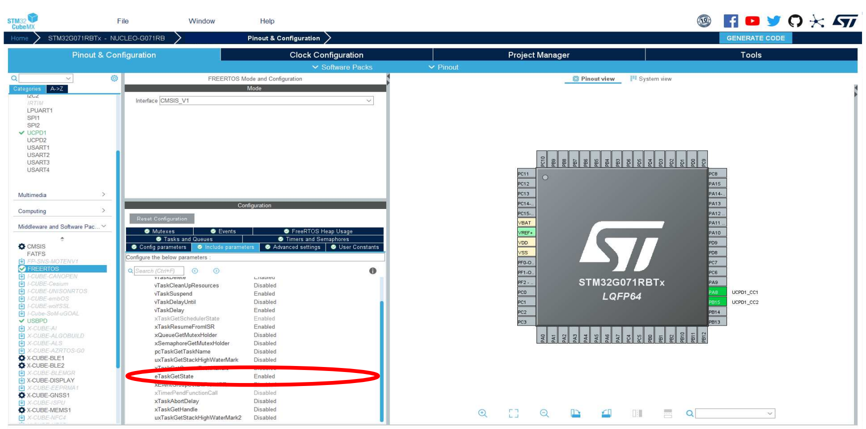


Figure 80 FreeRTOS Middleware configuration.

6.2.6.2 AzurRTOS

From the Pinout & Configuration tab, select the ‘ThreadX’ Middleware in the ‘Middleware and Software packs’ category. Enable its ‘Core’.

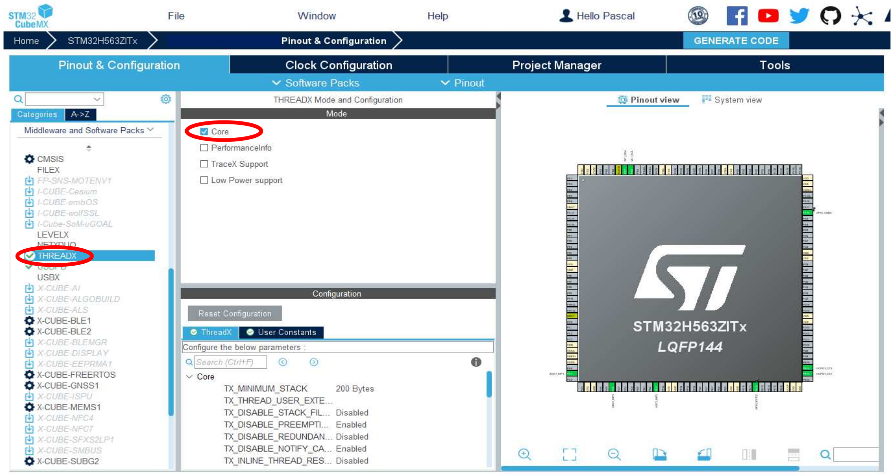


Figure 81 AzurRTOS Middleware configuration.

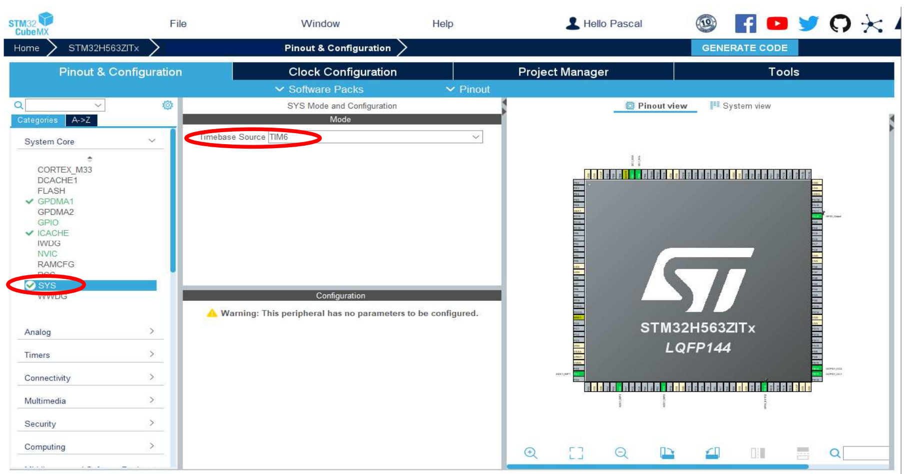


Figure 82 System Timebase Source modification.

6.2.7 I-Cache

If the MCU has the feature, feel free to enable ‘I-cache’ in the ‘System Core’ category for best performances.

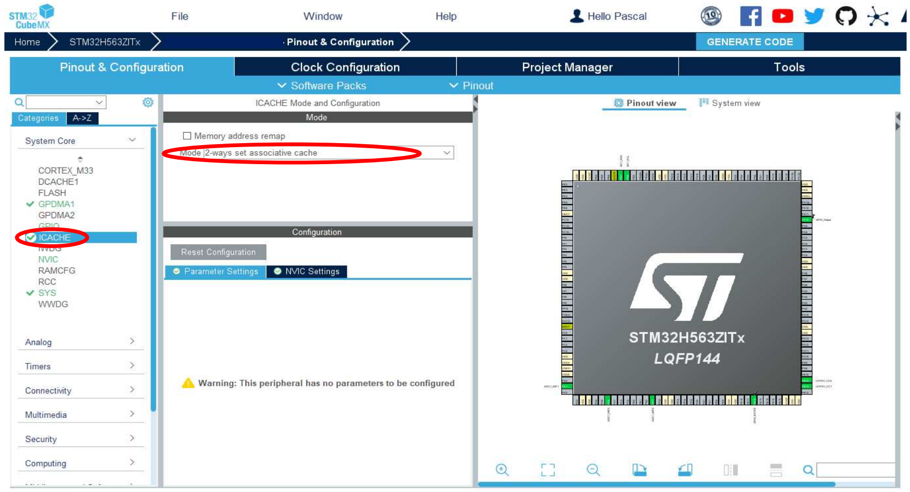


Figure 83 I-Cache Activation

6.2.8 Resources configuration for Sink Application (TCPP01)

6.2.8.1 GPIO Configuration

The TCPP01-M12 needs 2 GPIO Outputs for its ‘Vcc’ and ‘DB’ inputs.

On the pinout view, click left on your GPIO for DB and select ‘GPIO Output’ Then click right to name it as ‘DB_OUT’, repeat the same for ‘VCC_OUT’. Optionally you can set their names in the “User Label” field

In the GPIO Tab, set these 2 outputs as High level by default.

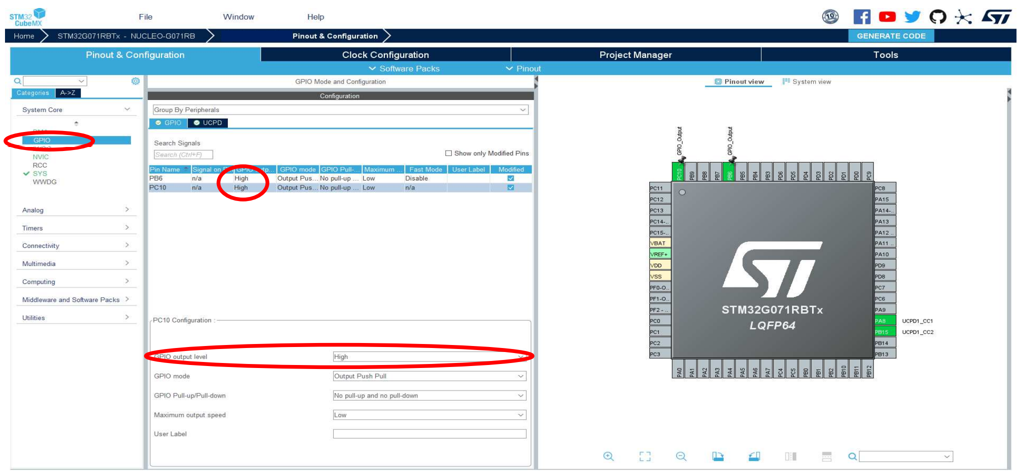


Figure 84 Sink - GPIO configuration

Note 1: Select PC10 (DB_OUT) and PB6 (VCC_OUT), for a SNK1M1 with a NUCLEO-G071.

Note 2: Select PC10 (DB_OUT) and PB1 (VCC_OUT), for a SNK1M1 with a NUCLEO-G474.

Note 3: If DB_OUT or VCC_OUT or both are directly tied to Vcc in the application, it is possible to pass this configuration.

6.2.8.2 ADC Configuration

Back to STM32CubeMX pinout and configuration window, click on ‘Analog’ category.

For the Power Delivery stack to work, VBUS needs to be monitored. An ADC needs to be configured. It will measure the VBUS voltage.

- Set the ‘Clock Prescaler’ to ‘Synchronous clock divided by 4’, adapt for other MCUs,
- For MCUs with higher resolution, keep 12 Bits resolution.
- Enable the ‘Continuous Conversion Mode’
- Set the overrun behavior to ‘Overrun data overwritten’.
- Select a medium SamplingTime.

Note: Adapt the configuration depending on the MCU capabilities.

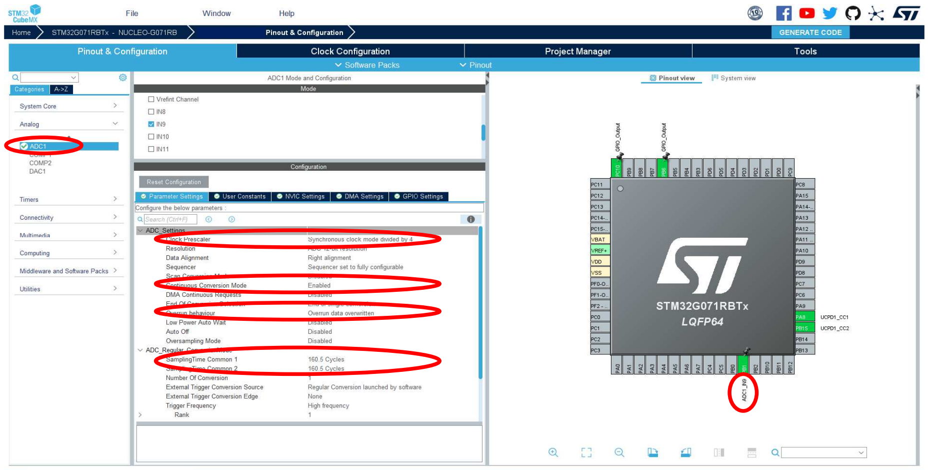


Figure 85 Sink - ADC configuration

Note 1: for a X-NUCLEO-SNK1M1 plugged on a NUCLEO-G0xx, select PB1-Ch9 for Vbus.

Note 2: for a X-NUCLEO-SNK1M1 plugged on a NUCLEO-G4xx, select PB0-Ch15 for Vbus.

6.2.8.3 Platform Settings configuration

Back in the Middleware and Software pack category. Select the X-CUBE-TCPP item. Select the ‘Platform Settings’ tab.

Affect resources depending on your choices:

For the application group, affect the ADC Channel for Vsense

For the BSP group, affect the “Vcc_Out” and the “DB_Out” GPIO Outputs.

Note: One or both outputs can be let as “undefined” if tied to Vcc.

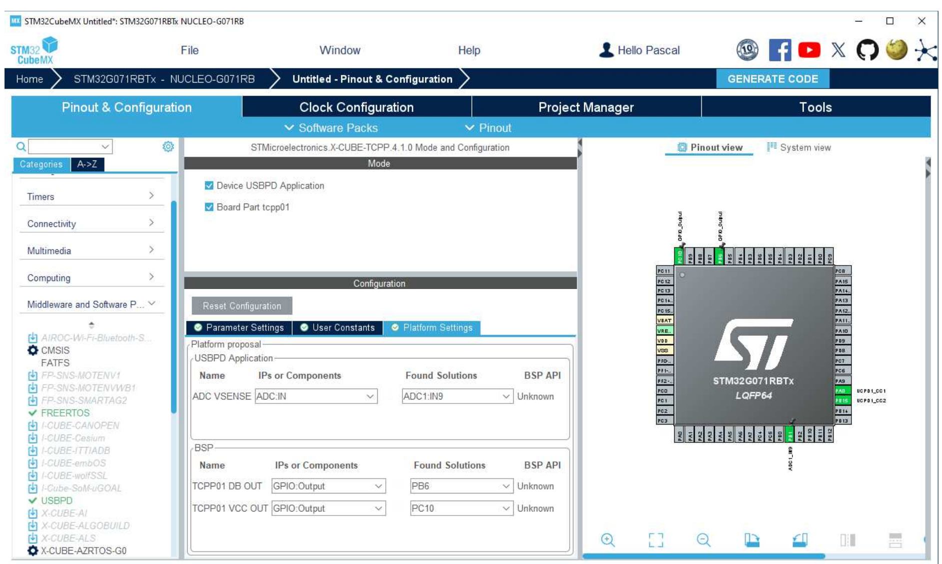


Figure 86 Sink – Software pack platform settings

6.2.9 Resources configuration for Source and Dual-Role Applications (TCPP02 & TCPP03)

6.2.9.1 I2C Configuration

As TCPP02-M18 and TCPP03-M20 communicate via I2C, enable the I2C peripheral in the 'Connectivity' section, enable I2C1 peripheral, in I2C mode. Set I2C Speed Mode to 'Fast speed'

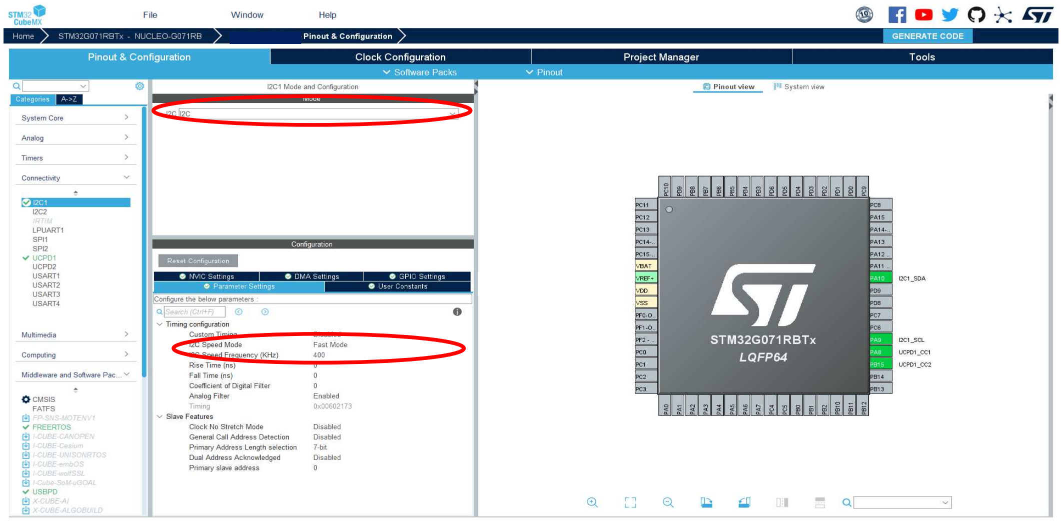


Figure 87 Source or Dual Role - I2C Selection and configuration

Note: Select I2C1 and re-map its pin on PB8 and PB9 for a SRC1M1 or DRP1M1 with a NUCLEO-G0xx or NUCLEO-G4xx.

Click left on these pins and select I2C1_SCL and I2C1_SDA.

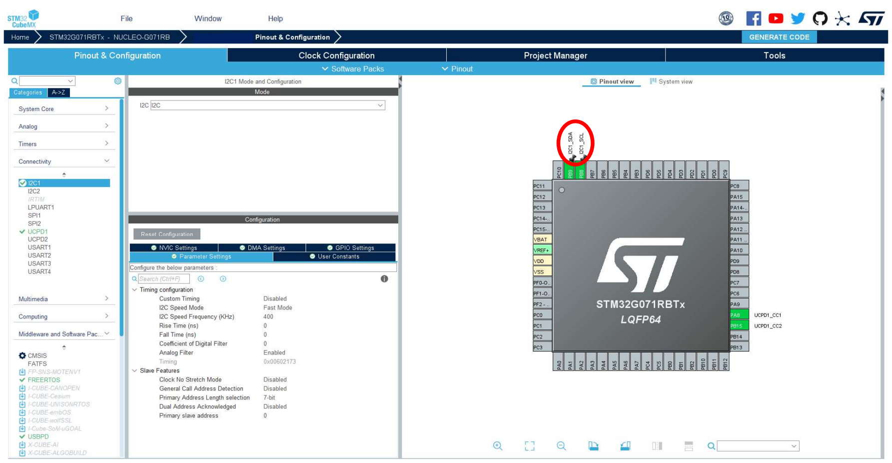


Figure 88 Source or Dual-Role – I2C GPIO Re-map

6.2.9.2 GPIO Configuration

TCPP02-M18 and TCPP03-M20 also need a GPIO Output for its ‘Enable’ input and an Interrupt input for its Alarm Flag ‘FLGN’ output.

On the pinout view, click left on your GPIO for Enable, and select GPIO Output. Then click right to name it as ‘ENABLE’.

Note: Select PC8 for a SRC1M1 or DRP1M1 with a NUCLEO-G0xx or NUCLEO-G4xx.

Note: The following step can be skipped if the FLGN input is not used.

Also on the pinout view, click left on your GPIO for ‘*FLGN*’ input, and select GPIO_EXTI mode, and with a right click name it as *‘FLGN’.*

In the system-core section, select GPIO, and then change this EXTI pin mode to ‘*External interrupt mode with falling edge detection*, and set a *Pull-Up.*

Note: Select PC5 (EXTI5) for a SRC1M1 of DRP1M1 with a NUCLEO-G0xx or NUCLEO-G4xx.

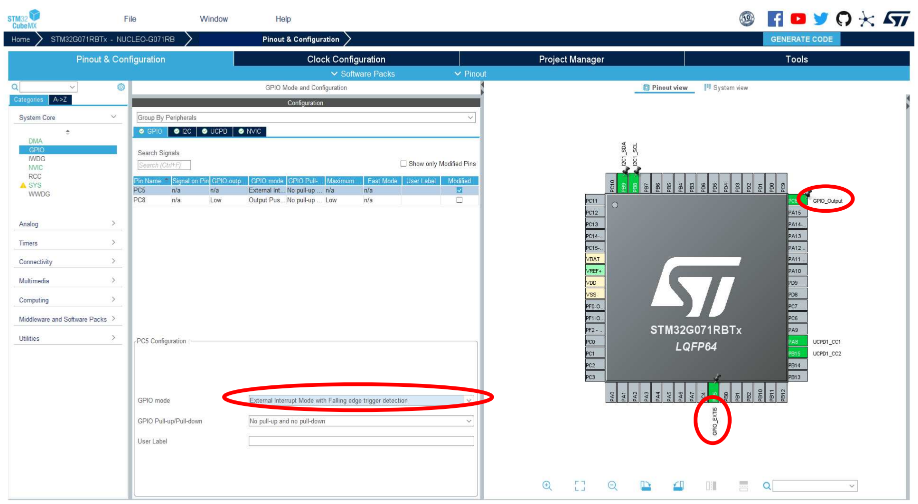


Figure 89 Source or Dual-Role – GPIO configuration

Finally, in the ‘NVIC’ tab, activate the related EXTI line interrupt.

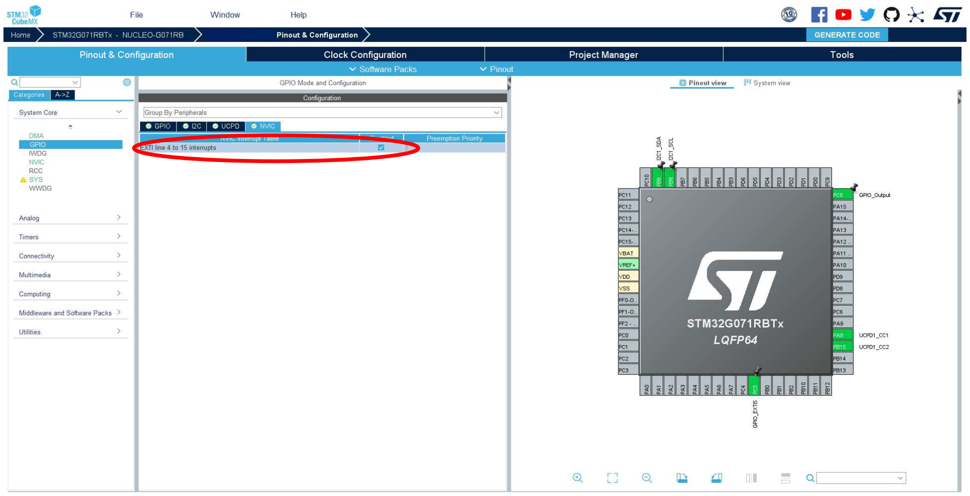


Figure 90 Source or Dual-Role – EXTI Interrupt activation

6.2.9.3 ADC Configuration

Back to STM32CubeMX pinout and configuration window, click on ‘Analog’ category. For the Power delivery stack to work, VBUS needs to be monitored. An ADC needs to be configured, it will measure the VBUS voltage, and optionally depending on your software pack selection: Vbus current, Vprovider, Vcons (for Dual-Role).

The ADC configuration differs and is simpler if only 1 ADC (Vbus) is selected. The following two chapters describe the configuration for both case:

6.2.9.3.1 ADC Configuration if only Vbus is selected

Back to STM32CubeMX pinout and configuration window, click on ‘Analog’ category.

For the Power Delivery stack to work, VBUS needs to be monitored. An ADC needs to be configured. It will measure the VBUS voltage.

- Set the ADC Channel as Single-ended
- Set the ‘Clock Prescaler’ to ‘Synchronous clock divided by 4’, adapt for other MCUs,
- For MCUs with higher resolution, keep 12 Bits resolution.
- Enable the ‘Continuous Conversion Mode’
- Set the overrun behavior to ‘Overrun data overwritten’.
- Select a medium SamplingTime.

Note: Adapt the configuration depending on the MCU capabilities.

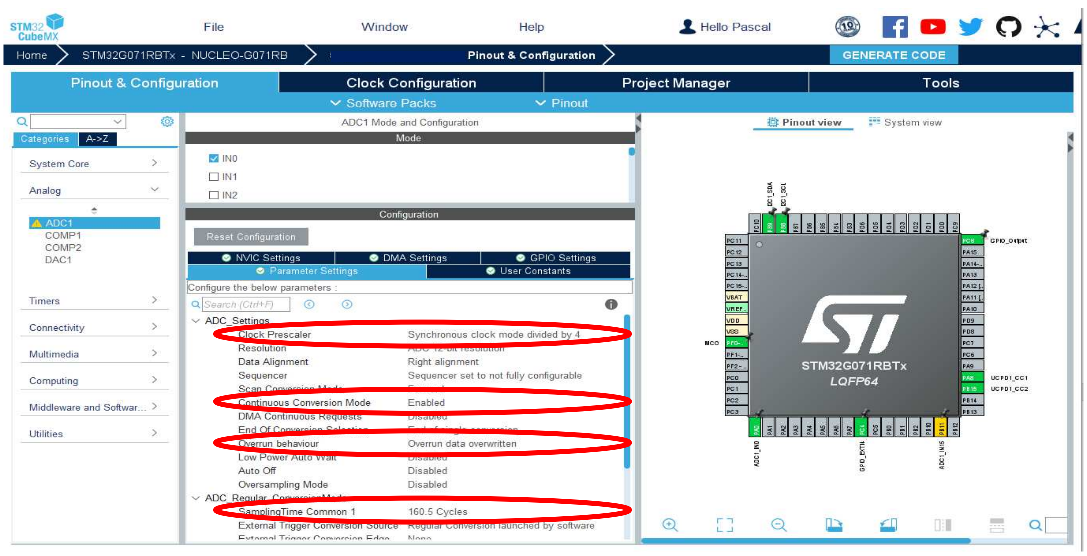


Figure 91 Source or Dual-Role – ADC Channel configuration

Note 1: for a SRC1M1 or DRP1M1 with a NUCLEO-G0xx, select PA0-Ch0 for Vbus.

Note 2: for a SRC1M1 or DRP1M1 with a NUCLEO-G4xx, select PA0-Ch1 for Vbus.

6.2.9.3.2 ADC Configuration if more than Vbus is selected

Back to STM32CubeMX pinout and configuration window, click on ‘Analog’ category. For the Power delivery stack to work, VBUS needs to be monitored. An ADC needs to be configured, it will measure the VBUS voltage, and optionally depending on your software pack selection: Vbus current, Vprovider, Vcons (for Dual-Role).

In the ‘Analog’ section, select as channels as necessary from the same ADC. ADC Channels must be configured as ‘Single-ended’.

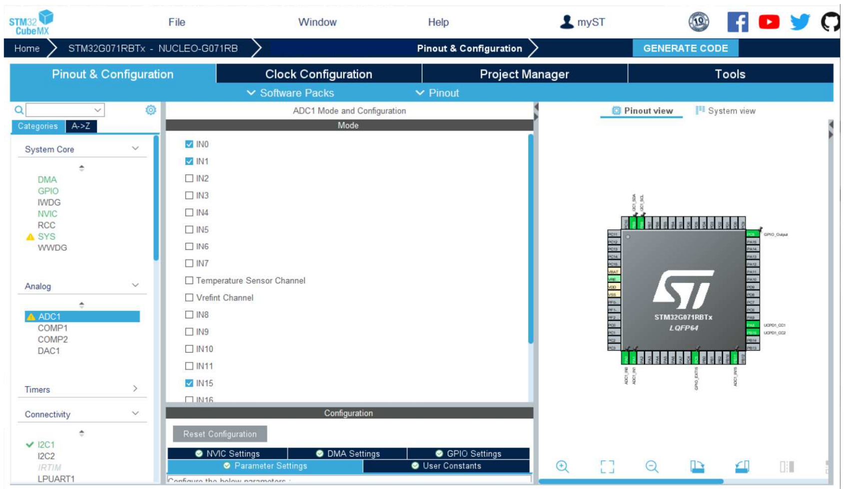


Figure 92 Source or Dual-Role – ADC Channels selection

Note: for a SRC1M1 or DRP1M1 with a NUCLEO-G0xx, select PA0-Ch0 for Vbus, PB11-Ch15 for Isense, PA1-Ch1 for Vprovider and PA4-Ch4 for Vcons (DRP)

Note: for a SRC1M1 or DRP1M1 with a NUCLEO-G4xx, select PA0-Ch1 for Vbus, PC1-Ch7 for Isense, PA1-Ch2 for Vprovider and PA4-Ch17 for Vcons (DRP)

In the ‘DMA Settings’ tab, click on ‘Add’ with a free DMA Channel. Set its mode to ‘Circular’.

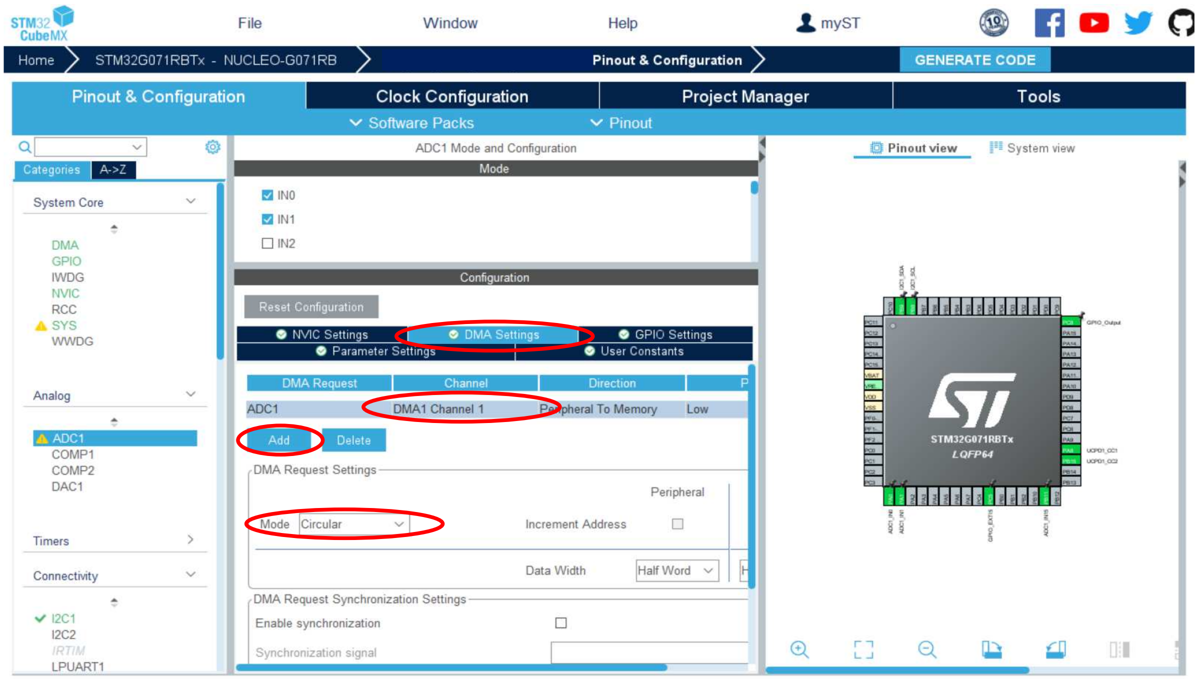


Figure 93 Source or Dual-Role – ADC DMA Configuration

Else in the ‘System Core’ category, enable a new GPDMA channel, in ‘Standard Request Mode’, assign it the ‘ADC’, in ‘Peripheral to Memory’ direction and enable ‘Destination Address Increment’.

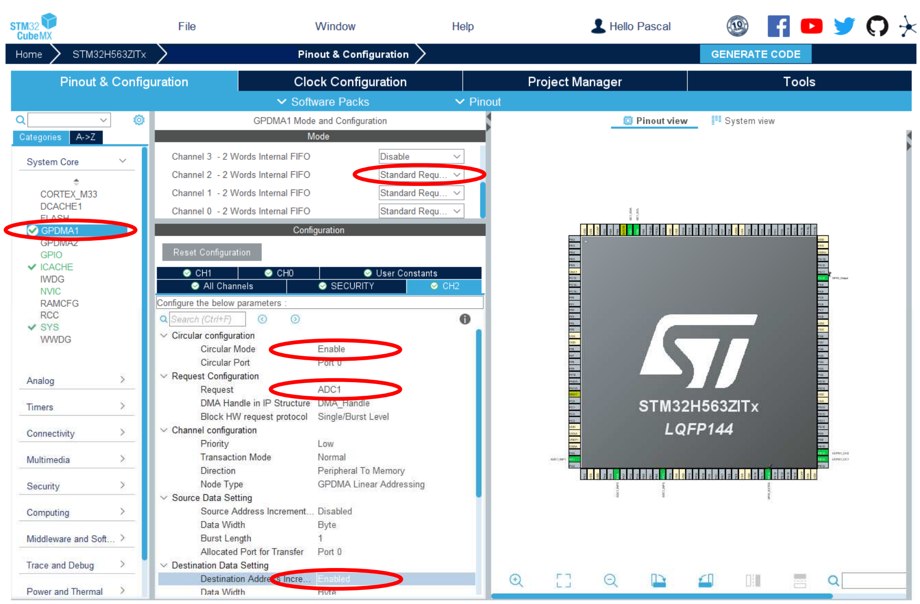


Figure 94 Source or Dual-Role – ADC DMA Configuration

In its ‘Parameter settings’ tab, set the ‘Clock Prescaler’ to ‘Synchronous clock mode divided by 4’, enable the ‘scan conversion’ mode, enable the ‘continuous conversion’ mode, also enable the ‘DMA Continuous Requests’, set the Overrun behavior to ‘Overrun data overwritten’ and increase the ‘SamplingTime’ to a medium value and set the ‘number of conversions’ to the number of selected channels.

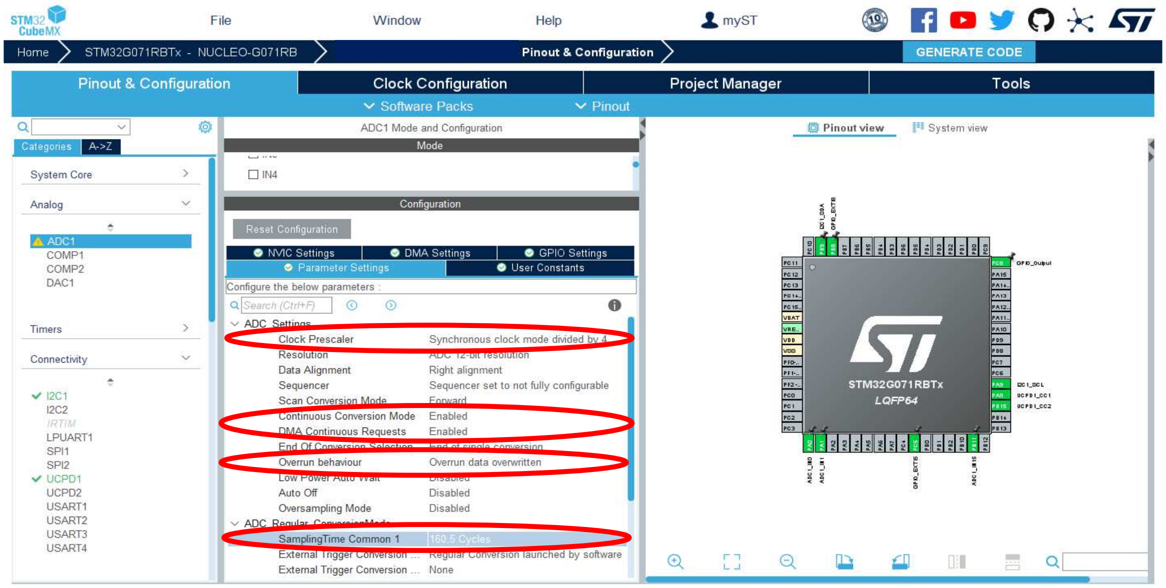


Figure 95 Source or Dual-Role – ADC configuration

Define Channels ranking: Respect imperatively Rank 1 for Channel corresponding to Vbus. And in case of GUI usage Rank 2 for the Vbus current.

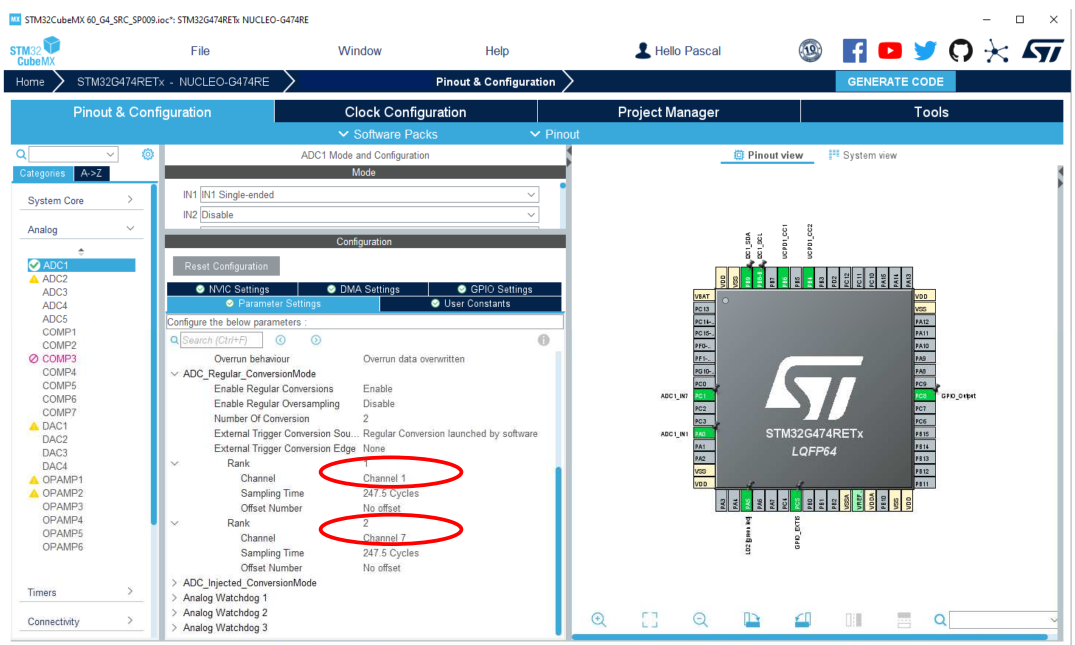


Figure 96 Source or Dual-Role – ADC configuration

6.2.9.4 Platform Settings Configuration

Back in the Middleware and Software pack category. Select the X-CUBE-TCPP item. Select the ‘Platform Settings’ tab.

Affect resources depending on your choices:

- For the application group, affect ADC Channels for Vbus, Isense, Vprov, Vcons
- For the BSP group, affect the I2C bus, the Enable Output and the FLG EXTI.

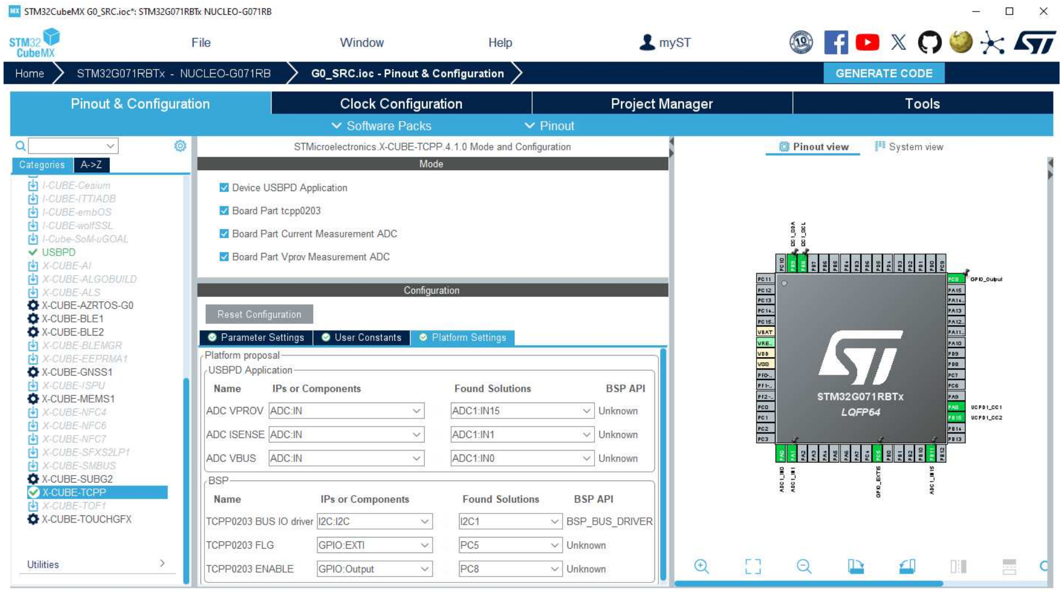


Figure 97 Source – Software pack platform settings

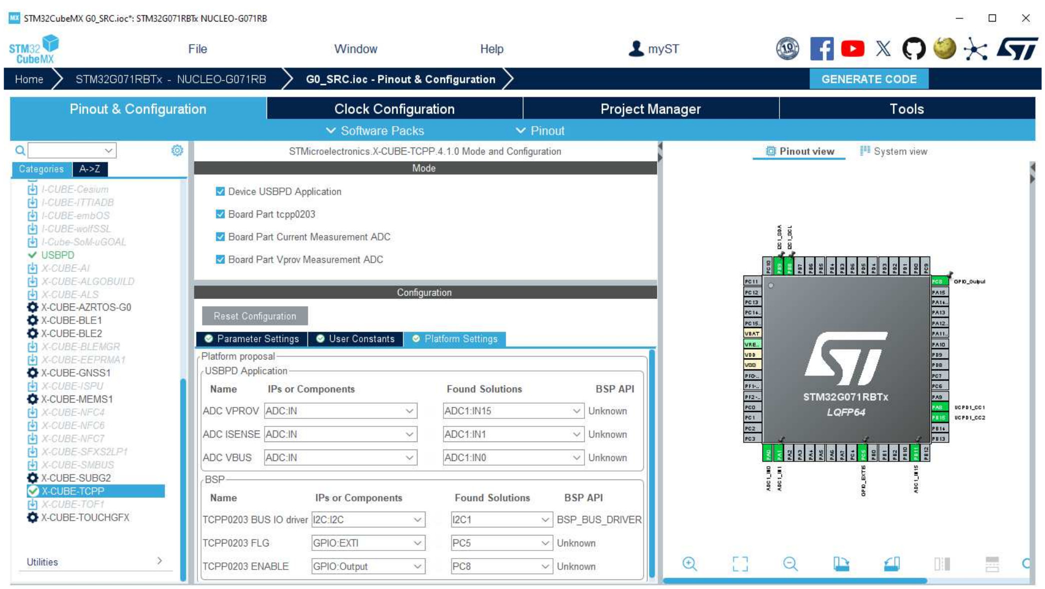


Figure 98 Dual-Role – Software pack platform settings

6.2.9.5 Parameters Settings Configuration

Still in the X-CUBE-TCPP Software pack item, select the ‘Parameters Settings’ tab.

Adjust resistor bridges and current sense resistors upon you board:

- For example, defaults settings are for SNK1M1, SRC1M1 or DRP1M1 X-NUCLEO with:
    - The analog reference voltage VDDA_APPLI of 3300 mV
    - A Vbus bridge divider of 200kOhms and 40kOhms
    - A Current sense resistor of 7mOhms
- As another example, for a STM32H573I-DK or B-U585AI-IOT02A these values should be:
    - The analog reference voltage VDDA_APPLI of 3300 mV
    - A Vbus bridge divider of 330kOhms and 50kOhms
    - A Current sense resistor of 47mOhms

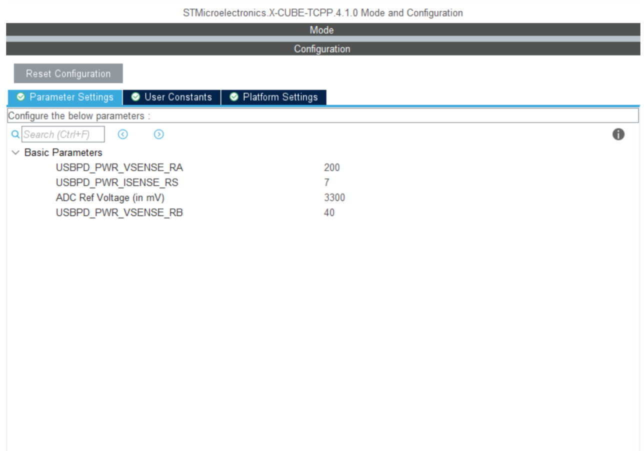


Figure 99 Software pack parameters settings

6.2.10 Option : set tracer and GUI for debug

Following configuration is optional, dedicated to Tracer and GUI, which is useful for debug, and requires a serial interface to USB such as a ST-LINK.

From the ‘Connectivity’ tab, enable an USART or LPUART, set its baud rate to 921600 and its word length to 7bits.

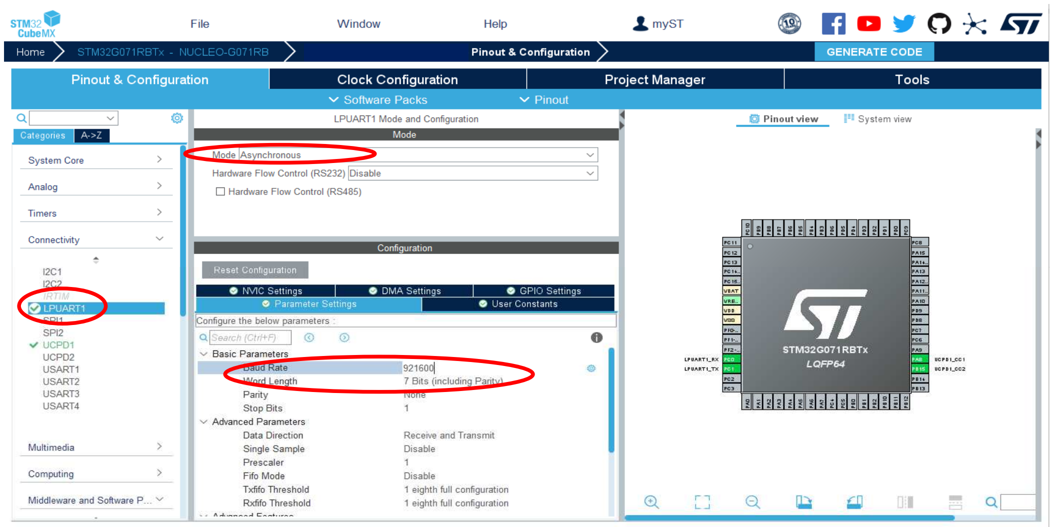


Figure 100 LPUART Parameter Settings

Then select the ‘DMA Settings’ tab and add the LPUART_TX DMA, with an empty channel, for example Channel 3.

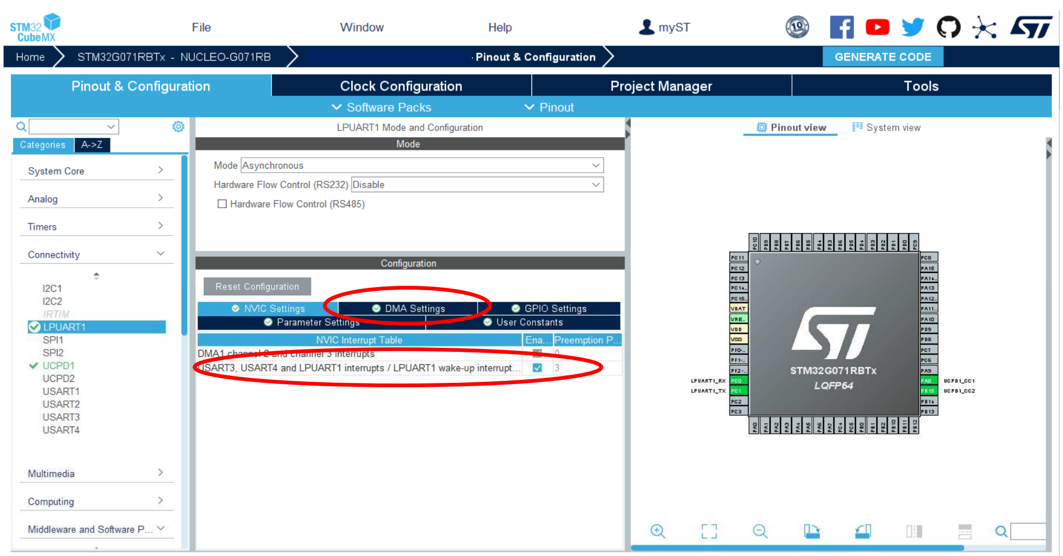


Figure 101 LPUART DMA Settings

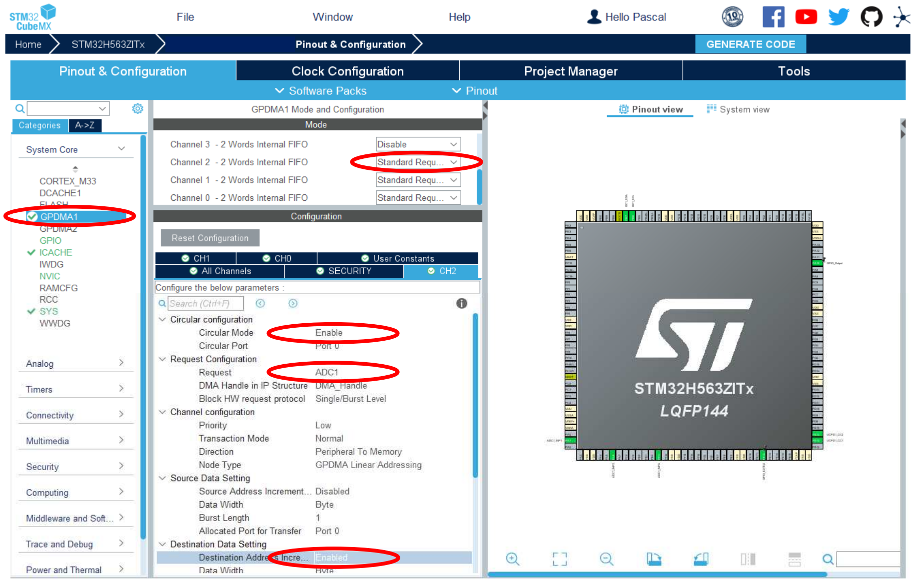


Figure 102 LPUART GPDMA Settings

Finally, in the ‘NVIC Settings’ enable the LPUART global interrupt.

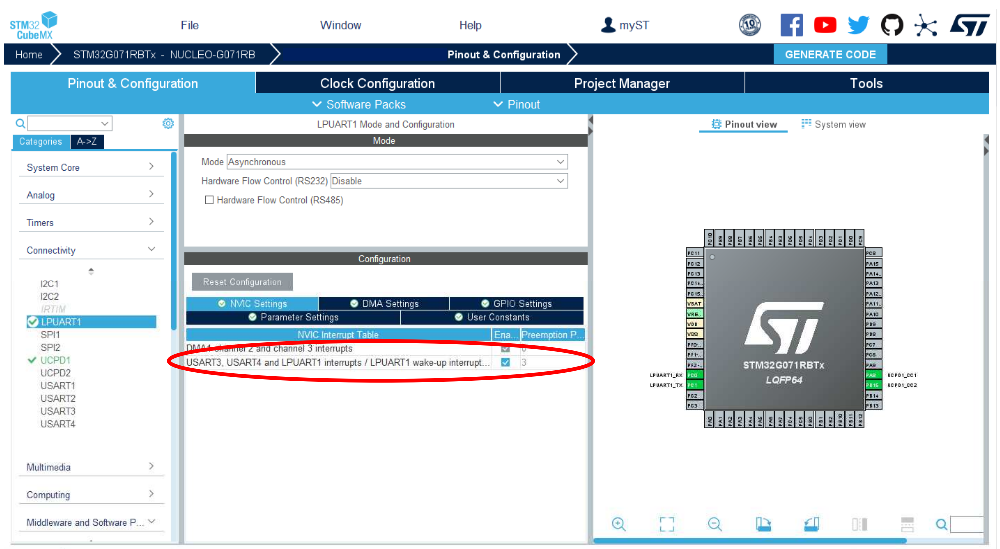


Figure 103 LPUART NVIC Settings

In case of a Nucleo, In the pinout view, click left on PA2 and PA3 to re-map them as 'LPUART1_TX' and 'LPUART1_RX'.

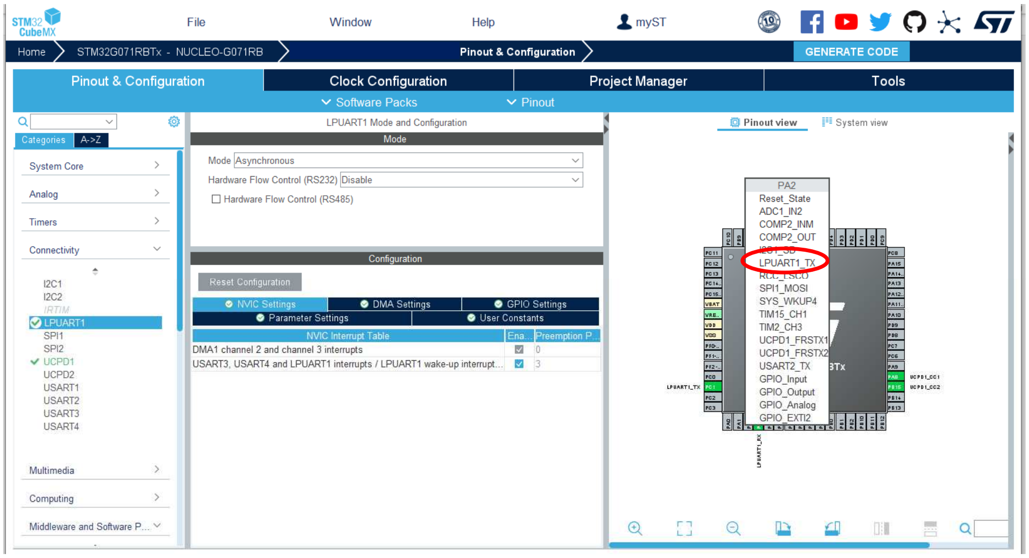


Figure 104 LPUART re-map

Then in the 'Utility' category, enable the 'Tracer' and select the LPUART for source.

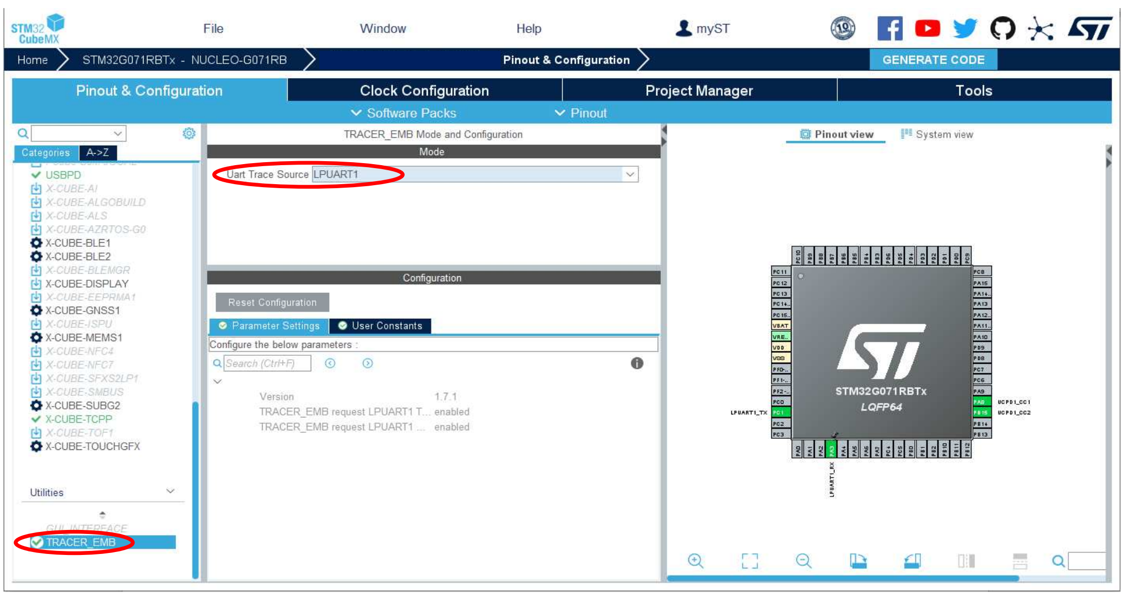


Figure 105 Tracer configuration

Then in the 'USBPD Middleware' check 'Tracer Source'

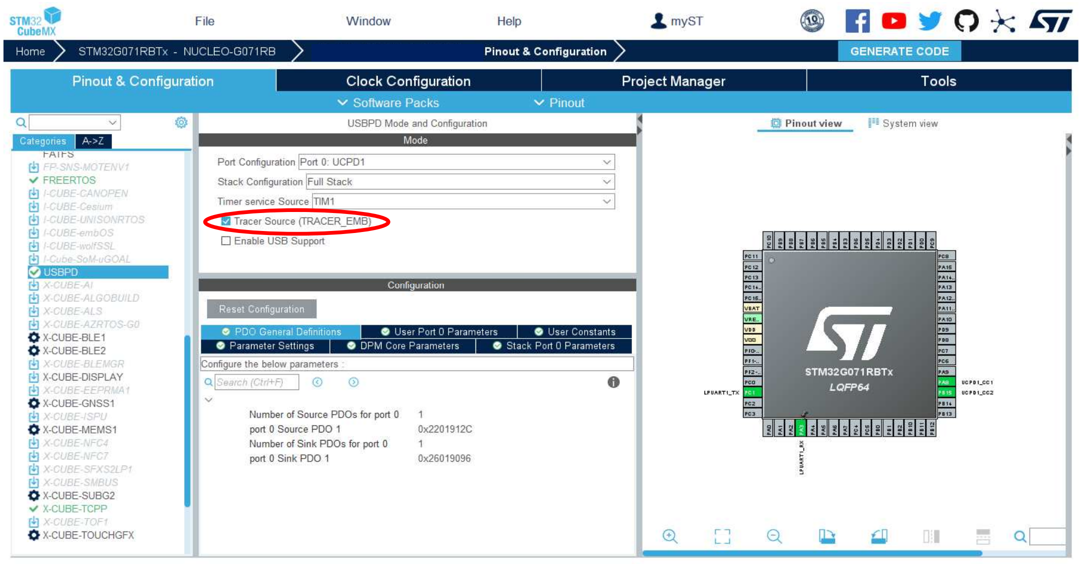


Figure 106 Tracer source configuration

Finally, back to the 'Utility' category, enable the 'GUI_Interface'

Set a HWBoardVersionName and a PDTypeName. For example, 'DRP1M1' and 'MB1360' (Name of the NUCLEO-G071RB Board but all names are convenient)

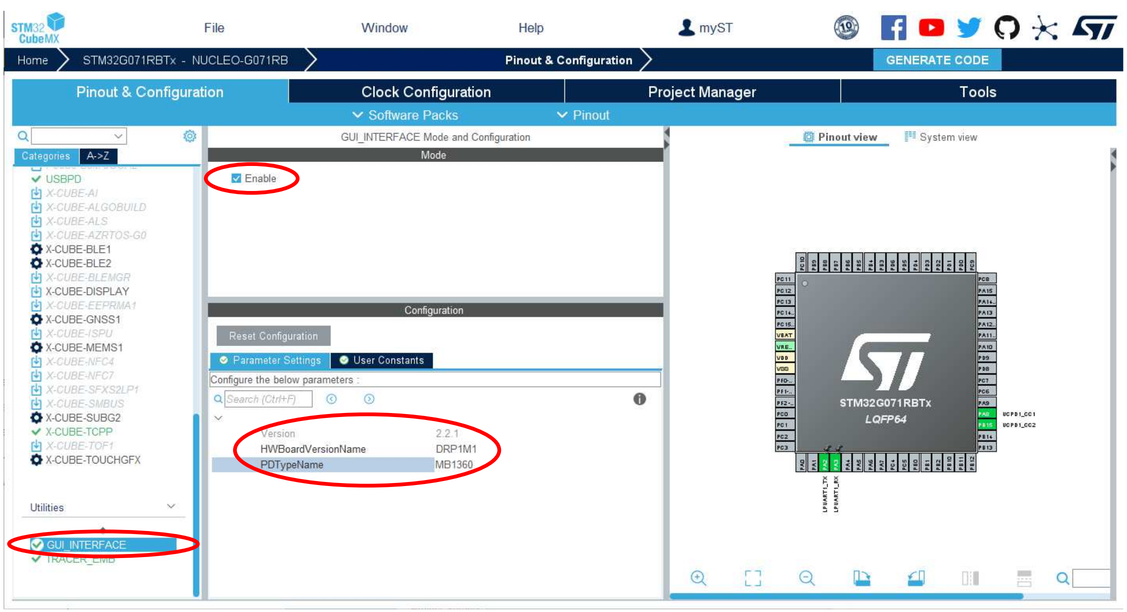


Figure 107 GUI_Interface configuration

6.2.11 Finalization and project generation

Adjust the clock frequency.

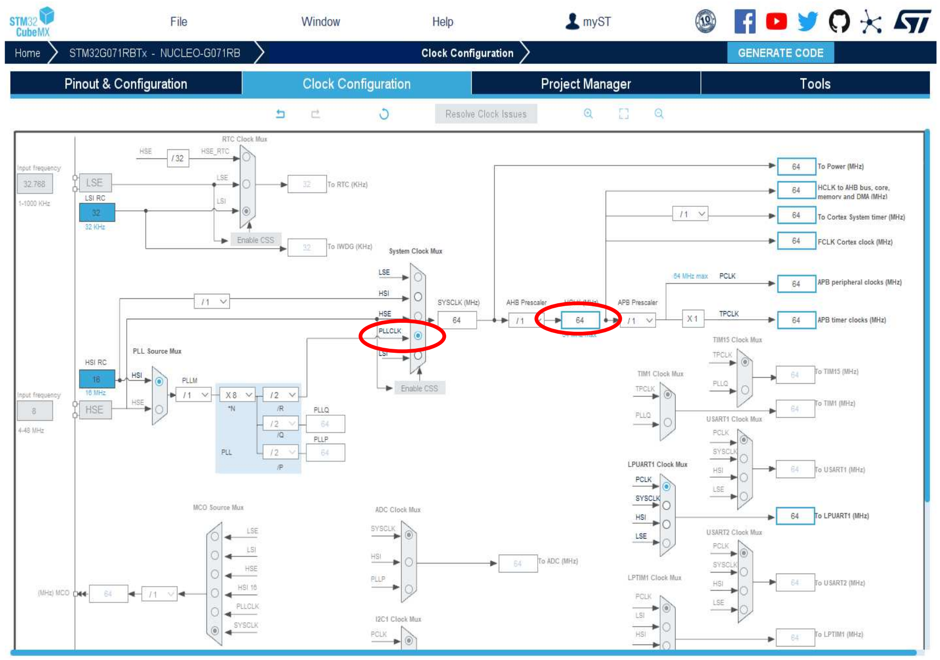


Figure 108 Clock configuration

Adjust the memory heap

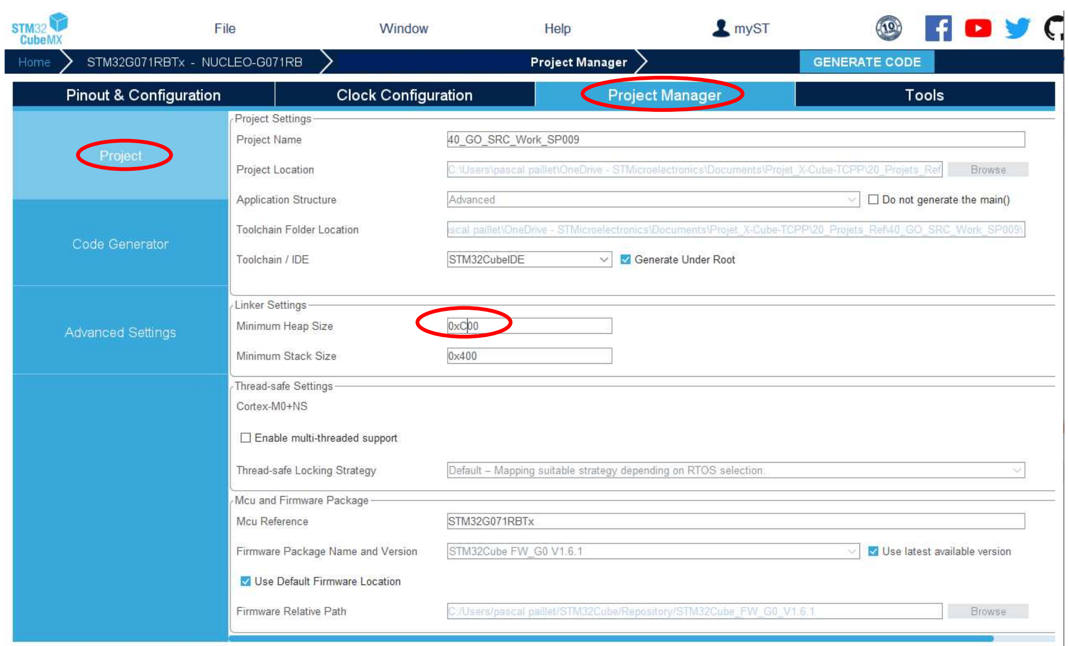


Figure 109 Heap size configuration

Generate the code for your IDE, for example STM32CubeIDE.

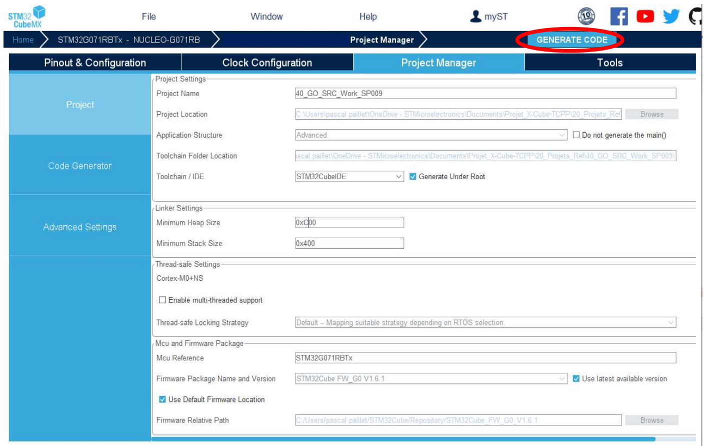


Figure 110 Code Generation

Following warning is usual, Click on “Yes”

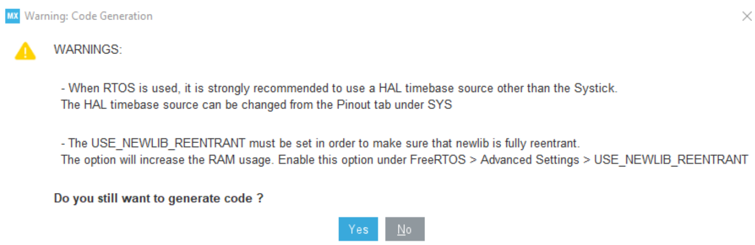


Figure 111 Source or Dual-Role – Code Generation

In your IDE, compile the code. It should be without any warning or error. And flash it in your application.

- The compilation must be performed without error or warnings.
- Build the application by clicking on the button (or select Project/Build Project).
- Run the application by clicking on the button (or select Run/Run)

If a Tracer and GUI has been configured: Use STCubeMonitor for debug:
With your application running on the board, launch the STM32CubeMonitor-UCPD application. The user's board must appear in the list when clicking ‘Refresh list of connected boards’, so double-click on the corresponding line (or click ‘NEXT’).

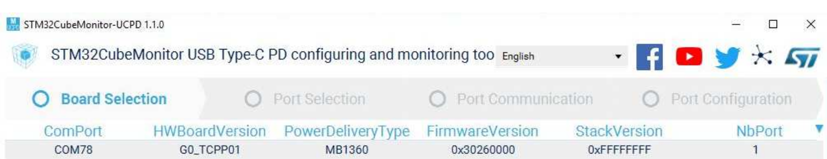


Figure 112 Sink - STM32CubeMonitor Connection

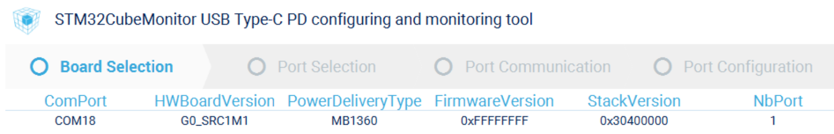


Figure 113 Source - STM32CubeMonitor Connection

Figure 114 Dual-Role - STM32CubeMonitor Connection

Note: The ComPort may be different. It depends on the number of boards installed on the computer. Then double-click on the desired UCPD port, here Port 0, or select it and click 'NEXT'.

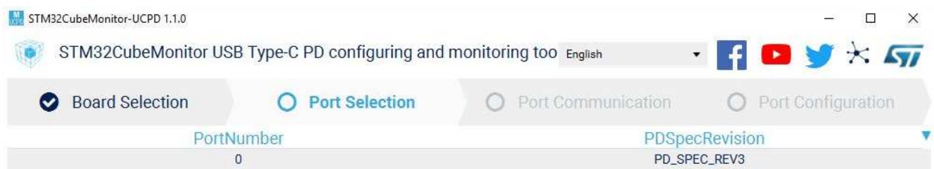


Figure 115 STM32CubeMonitor Port Selection

Click on the TRACES button in the bottom right corner to get protocol traces. Here below is the example of a Dual-Role application, that covers all 3 cases: Sink, Source and DRP:

You can then plug a power delivery sink into the USB Type-C® receptacle of the X-NUCLEO-DRP1M1 shield. The screen may look like this:

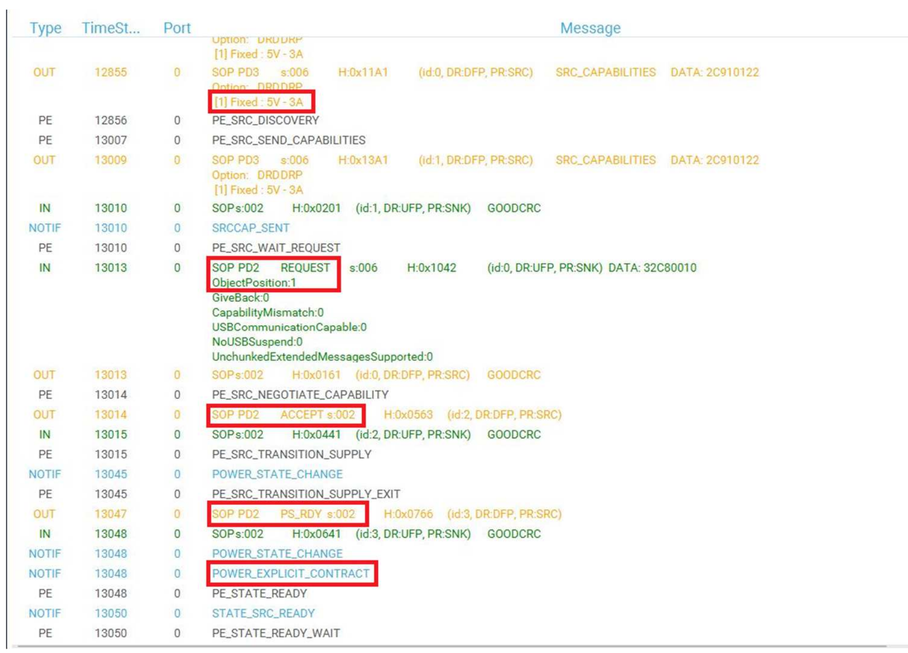
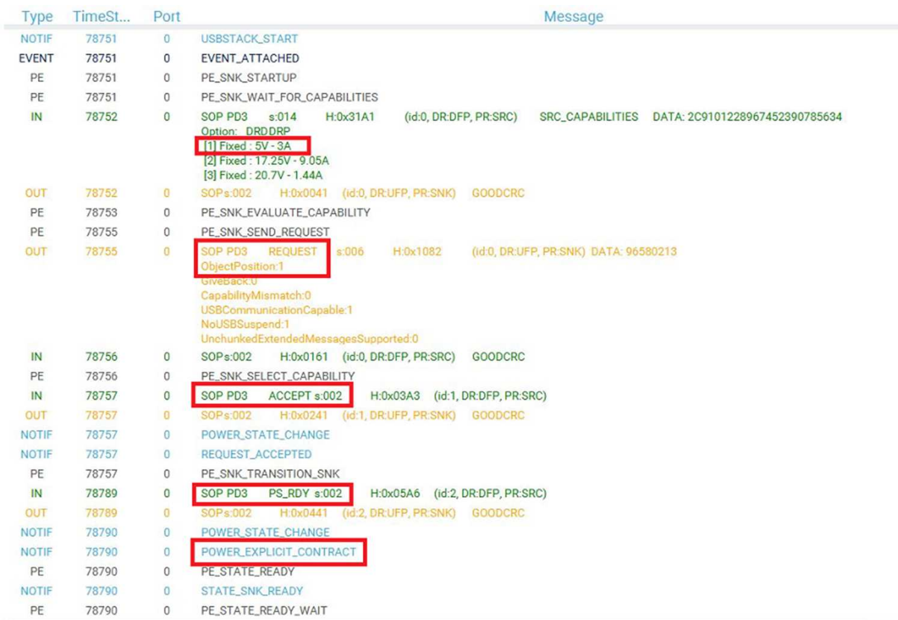


Figure 116 Dual-Role - STM32CubeMonitor Communication

The figure above shows the communication between the STM32G0 and the power delivery sink on the right panel. It is possible to verify the correct sequence to reach an explicit contract:

- The capabilities are sent by the STM32G0 DRP (OUT orange message).
- The request is sent by the sink (IN green message).
- The ACCEPT and the PS_RDY are sent by the STM32G0 source (OUT orange message).
- The contract negotiation ends by the POWER_EXPLICIT_CONTRACT notification (blue message).



Type	TimeSt...	Port	Message
NOTIF	78751	0	USBSTACK_START
EVENT	78751	0	EVENT_ATTACHED
PE	78751	0	PE_SNK_STARTUP
PE	78751	0	PE_SNK_WAIT_FOR_CAPABILITIES
IN	78752	0	SOP PD3 s:014 H:0x31A1 (id:0, DR:DFP, PR:SRC) SRC_CAPABILITIES DATA: 2C9101228967452390785634 Option: DRDDRP [1] Fixed : 5V - 3A [2] Fixed : 17.25V - 9.05A [3] Fixed : 20.7V - 1.44A
OUT	78752	0	SOPs:002 H:0x0041 (id:0, DR:UFP, PR:SNK) GOODCRC
PE	78753	0	PE_SNK_EVALUATE_CAPABILITY
PE	78755	0	PE_SNK_SEND_REQUEST
OUT	78755	0	SOP PD3 REQUEST s:006 H:0x1082 (id:0, DR:UFP, PR:SNK) DATA: 96580213 ObjectPosition:1 GiveBack:0 CapabilityMismatch:0 USBCommunicationCapable:1 NoUSBSuspend:1 UnchunkedExtendedMessagesSupported:0
IN	78756	0	SOPs:002 H:0x0161 (id:0, DR:DFP, PR:SRC) GOODCRC
PE	78756	0	PE_SNK_SELECT_CAPABILITY
IN	78757	0	SOP PD3 ACCEPT s:002 H:0x03A3 (id:1, DR:DFP, PR:SRC)
OUT	78757	0	SOPs:002 H:0x0241 (id:1, DR:UFP, PR:SNK) GOODCRC
NOTIF	78757	0	POWER_STATE_CHANGE
NOTIF	78757	0	REQUEST_ACCEPTED
PE	78757	0	PE_SNK_TRANSITION_SNK
IN	78789	0	SOP PD3 PS_RDY s:002 H:0x05A6 (id:2, DR:DFP, PR:SRC)
OUT	78789	0	SOPs:002 H:0x0441 (id:2, DR:UFP, PR:SNK) GOODCRC
NOTIF	78790	0	POWER_STATE_CHANGE
NOTIF	78790	0	POWER_EXPLICIT_CONTRACT
PE	78790	0	PE_STATE_READY
NOTIF	78790	0	STATE_SNK_READY
PE	78790	0	PE_STATE_READY_WAIT

Figure 117 Dual-Role - STM32CubeMonitor Communication

The figure above shows the communication between the STM32G0 and the power delivery source on the right panel. It is possible to verify the correct sequence to reach an explicit contract:

- The capabilities are sent by the source (IN green message).
- The request is sent by the STM32G0 DRP (OUT orange message).
- The ACCEPT and the PS_RDY are sent by the source (IN green message).
- The contract negotiation ends by the POWER_EXPLICIT_CONTRACT notification (blue message).

For more details on how to use this tool, refer to UM2468. And for more details on the protocol, refer to UM2552. Note that this trace is very helpful for debugging and application development.

You can also use the Measurement window in STM32CubeMonitor-UCPD to display a graph of the measured VBUS voltage and delivered current. Set the sampling period and click start.

Figure 118 STM32CubeMonitor Measurement

6.3 USB no Power-Delivery Application – Source Application (5V only) using X-NUCLEO-SRC1M1

The USB Type-C Source No Power Delivery – 5V only application needs several resources.

First, clear the pinout:

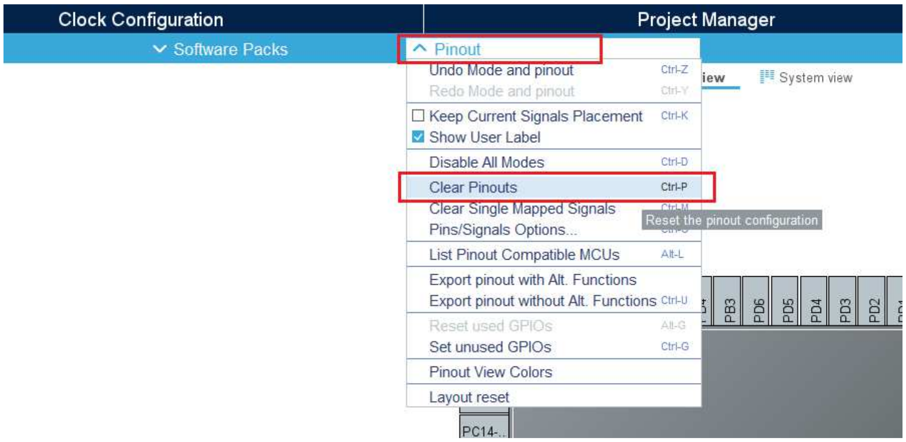


Figure 119 Source NoPD SRC1M1 – Clear pre-defined pinout

Then select the Software Pack components: From the Software Packs category, press the 'STMicroelectronics-CUBE-TCPP' item and enable the 'Source' application in the USB no Power Delivery section.

A warning appears: click on the warning on the 'STMicroelectronics-X-CUBE-TCPP' row to show missing items.

- Select the 'TCPP0203' board part using its checkbox.
- Select the 'X-NUCLEO-SRC1M1' board extension using its checkbox.

Its exists warning due to the lack of 'ADC_HAL' and 'I2C_HAL' that will be solved later.

Press 'OK' to close the window.

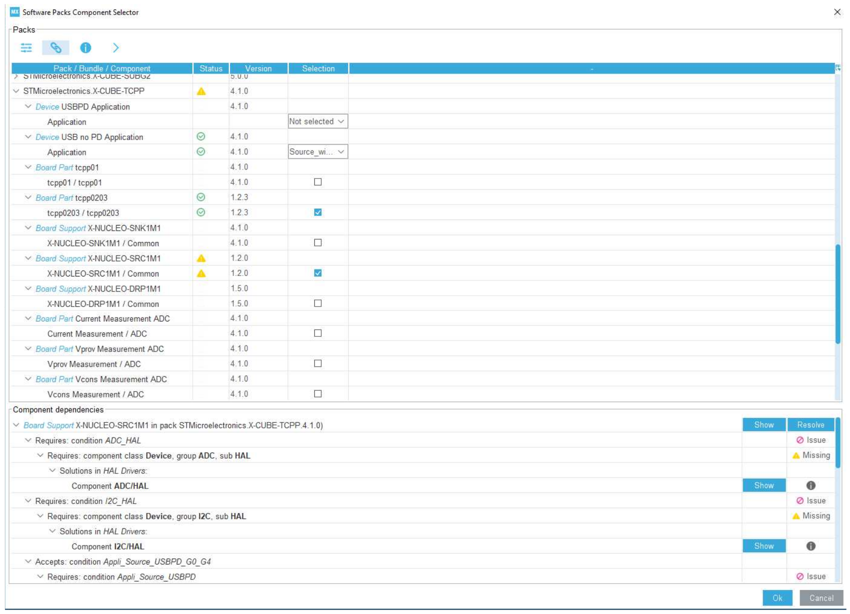


Figure 120 Source noPD SRC1M1 – Software Pack components selection

Select the ‘Middleware and software packs’ category. Select the ‘X-CUBE-TCPP’ item. Check the ‘Source’ application, the ‘TCPP0203’ board part and the ‘SRC1M1’ board extension.

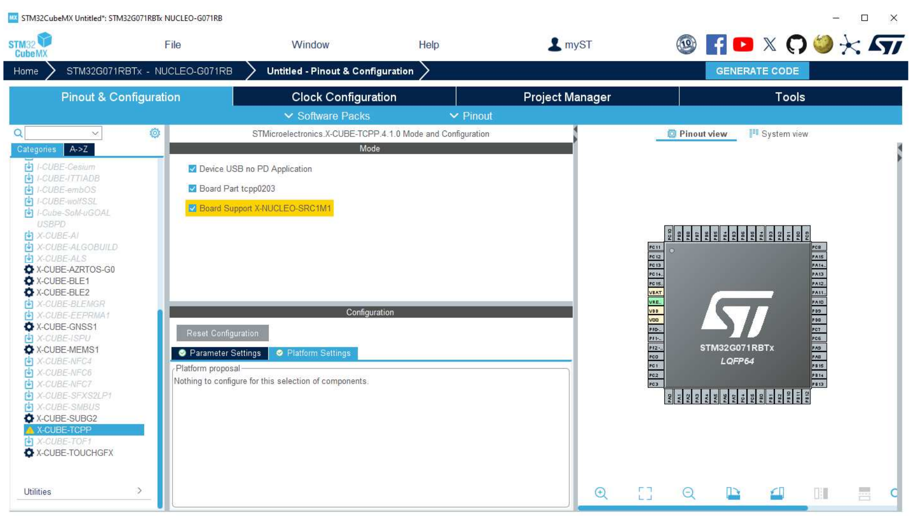


Figure 121 Source noPD SRC1M1 – Software Pack components activation

Under the Clock configuration main tab, change the system clock mux to PLLCLK, it will set HCLK clock to 84MHz.

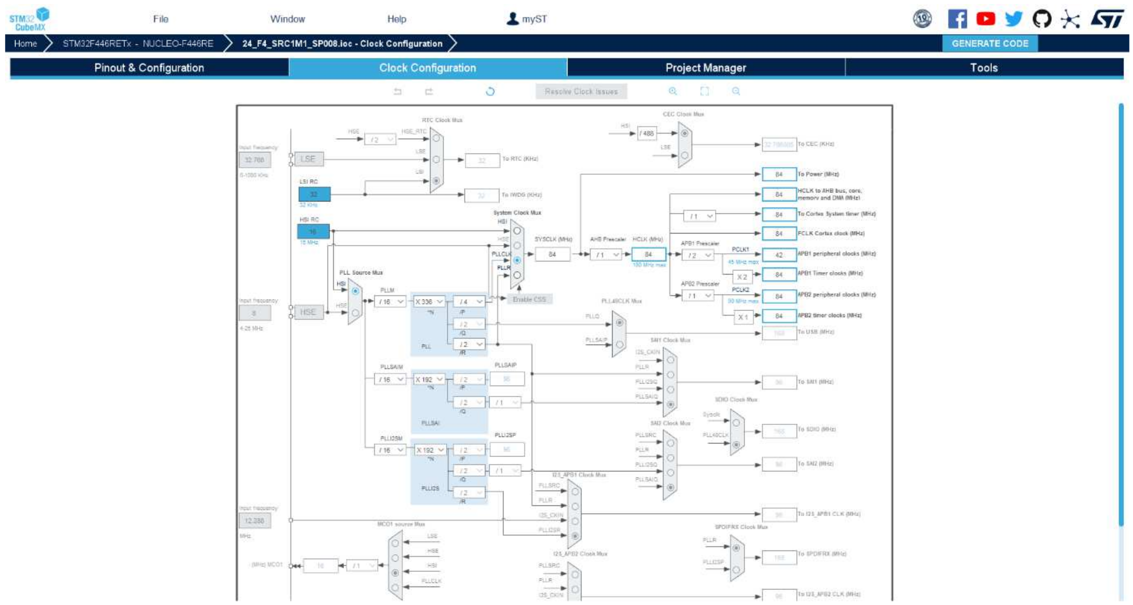


Figure 122 Source noPD SRC1M1 – Clock configuration

Generate the code for your IDE, for example STM32CubeIDE.

Save your file with Ctrl+s and select generate code if prompted. You can also generate code from the STM32CubeIDE menu, clicking on Project/Generate Code, or by pressing Alt+K.

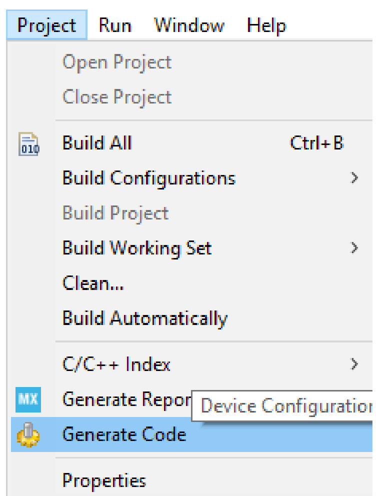


Figure 123 Source noPD SRC1M1 – Code Generation

In your IDE, compile the code. It should be without any warning or error. And flash it in your application.

- The compilation must be performed without error or warnings.
- Build the application by clicking on the button (or select Project/Build Project).
- Run the application by clicking on the button (or select Run/Run)

Note: When plugging a SRC1M1 X-NUCLEO on a NUCLEO-F446, the TCPP02 FLG output is routed on a NC pin of the NUCLEO. Then place a link between PA3 (CN10 - 37) and PC4 (CN10 - 34).

Please adapt in function of your NUCLEO board.

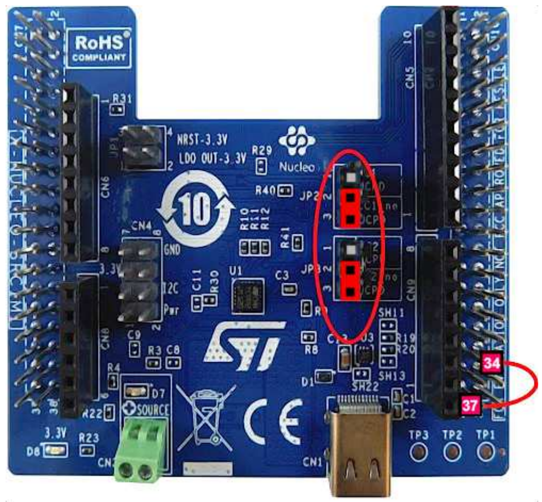


Figure 124 Source noPD SRC1M1 – X-NUCLEO-SRC1M1 configuration

This X-NUCLEO-SRC1M1 shield default configuration allows SINK to source up to 0.5A @ 5V.

Plug an external 5V source with current capability >0.6A into the green "source" connector.

The current sense resistor R4 is 7mOhms, then TCPP02 current protection level is 6A. Refer to TCPP02-M18 datasheet. [5]

With this configuration, the board is powered by the ST-Link of the Nucleo board.

If you want to power your system from the external power supply connected to the "source" terminal, and not from the ST-Link, add the JP1 jumpers between 1-2 and 3-4.

Note: To increase the solution current capability to 3A @ 5V,

- Remove R35 and place it on SH19
- Remove R39 and place it on SH21
- Replace R4 sense resistor (initially 7mOhms) with a 10mOhms resistor
- Next, plug an external 5V source with current capability > 4.5A into the green "source" connector.
- In SRC1M1_conf.h change SRC1M1_ISENSE_RS value from 7 milliohms to 10 milliohms:

```
#define SRC1M1_ISENSE_RS            10u  /* Current measure shunt resistor in milliohm
*/
```

6.4 USB no Power-Delivery Application - Source Application (5V only) Free resources assignment

The USB Type-C Source No Power Delivery – 5V only application needs several resources.

- An I2C peripheral for the TCPP02
- A GPIO Output for the TCPP02 Enable input
- An input interrupt for the TCPP02 FLG Alarm output.
- 5 ADC inputs for CC1, CC2, Vbus, Isense and VProvider

If you are using a Nucleo, first, clear the pinout:

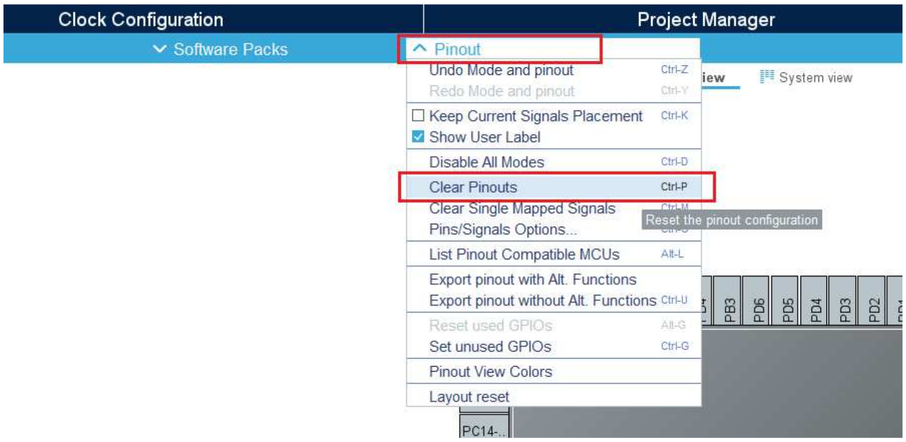


Figure 125 Source NoPD – Clear pre-defined pinout

Then select the Software Pack components: From the Software Packs category, press the ‘STMicroelectronics-CUBE-TCPP’ item and enable the ‘Source’ application in the USB no Power Delivery section.

A warning appears: click on the warning on the ‘STMicroelectronics-X-CUBE-TCPP’ row to show missing items.

- Select the ‘TCPP0203’ board part using its checkbox.

Then all warnings should be cleared.

Press ‘OK‘ to close the window.

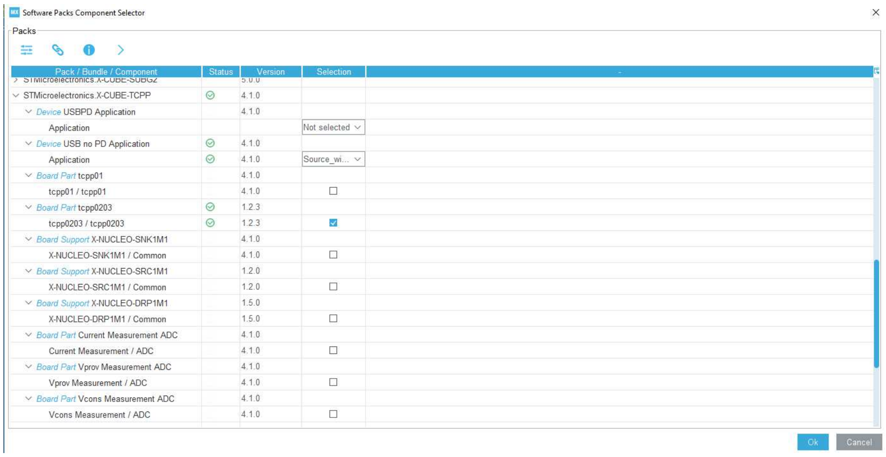


Figure 126 Source noPD – Software Pack components selection

Select the ‘Middleware and software packs’ category. Select the ‘X-CUBE-TCPP’ item. Check the ‘Source’ application, the ‘TCPP0203’ board part and the ‘X-NUCLEO-SRC1M1’ board extension.

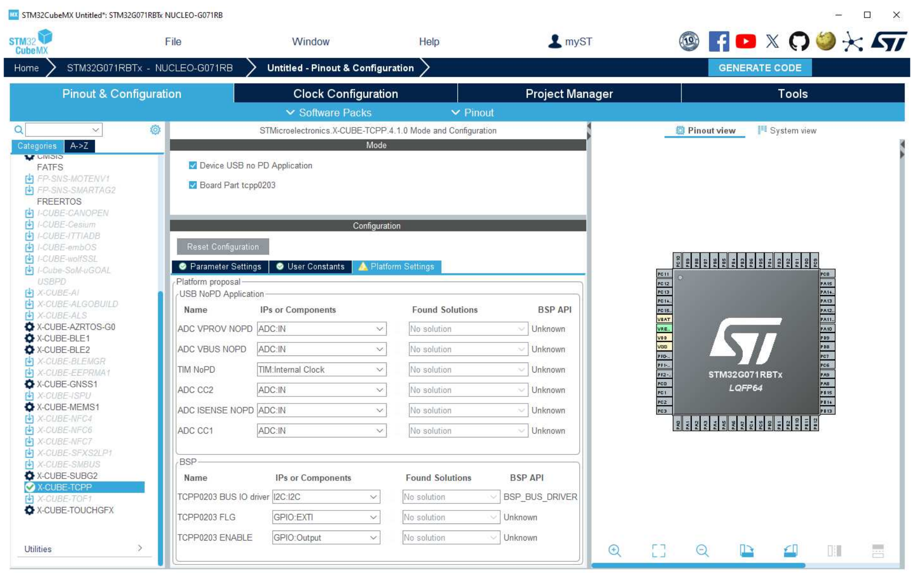


Figure 127 Source noPD – Software Pack components activation

In the ‘Platform Settings’ tab some resources must be affected.

The following chapters consist in these resources configuration to affect it in this ‘Platform Settings’ tab.

- Five analog signals must be monitored to ensure proper and safe USB 3A-5V delivery: CC1 and CC2 lines voltage, Vbus and Vprovider voltages and Iana, the current through Vbus.

  Back to STM32CubeMX pinout and configuration window, with a left click, select corresponding analog input pins and set it in ADCx-INy mode.

It is mandatory to use the same ADC for all 5 signals.

Note: for a X-NUCLEO-SRC1M1 plugged on a NUCLEO-F446, PA4 is configured as ADC1 Channel 4 (CC1), PC0 as ADC1 Channel 10 (CC2), PA0 as ADC1 Channel 0 (VBus), PA1 as ADC1 Channel 1(Vprov) and PC1 as ADC1 Channel 11 (IANA)
Note: it is not mandatory to set their name as each value is stored in a table by the DMA in function of its rank.

in the 'DMA' tab, click on 'Add' and select ADC1 in the DMA2 request column. Set the 'DMA' mode to: 'Circular'

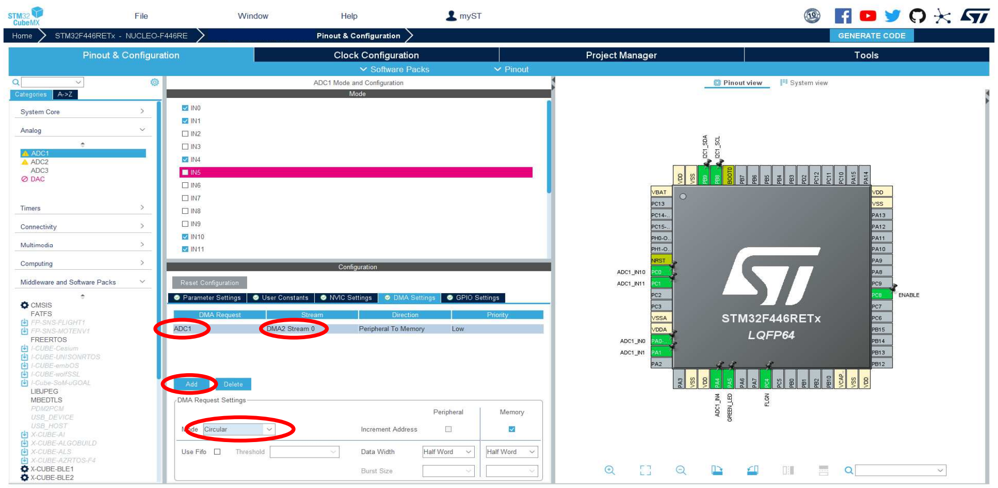


Figure 128 Source noPD – ADC Configuration with DMA

Adapt this method for MCU with GPDMA.

In the Analog section click on ADC and select, in the Parameter Settings tab:

- Scan conversion mode: Enabled
- Continuous conversion mode: Enabled
- DMA Continuous requests: Enabled
- Number of conversions: 5

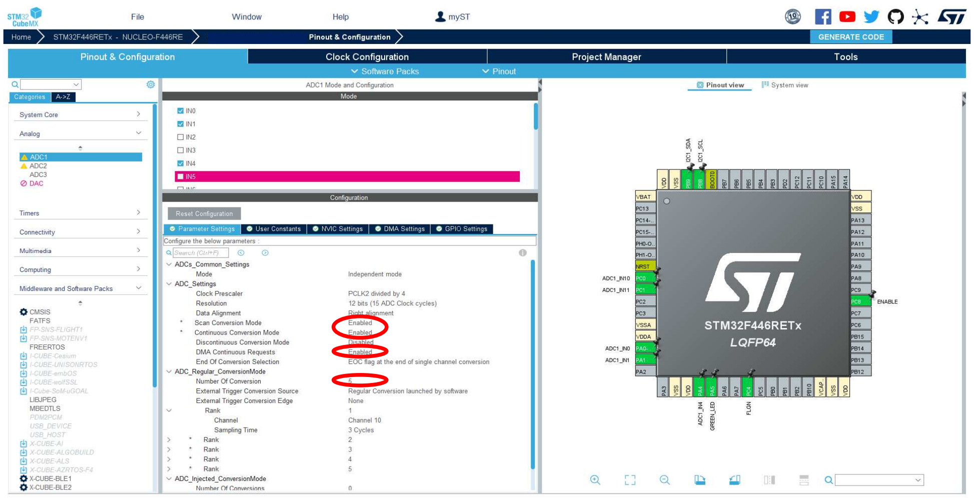


Figure 129 Source noPD – ADC Configuration

Define for each ADC input its channel and 84 cycles sampling time.

- Rank 1: CC2: Channel 10
- Rank 2: CC1: Channel 4
- Rank 3: Vbus: Channel 0
- Rank 4: Iana: Channel 11
- Rank 5: Vprov: Channel 1

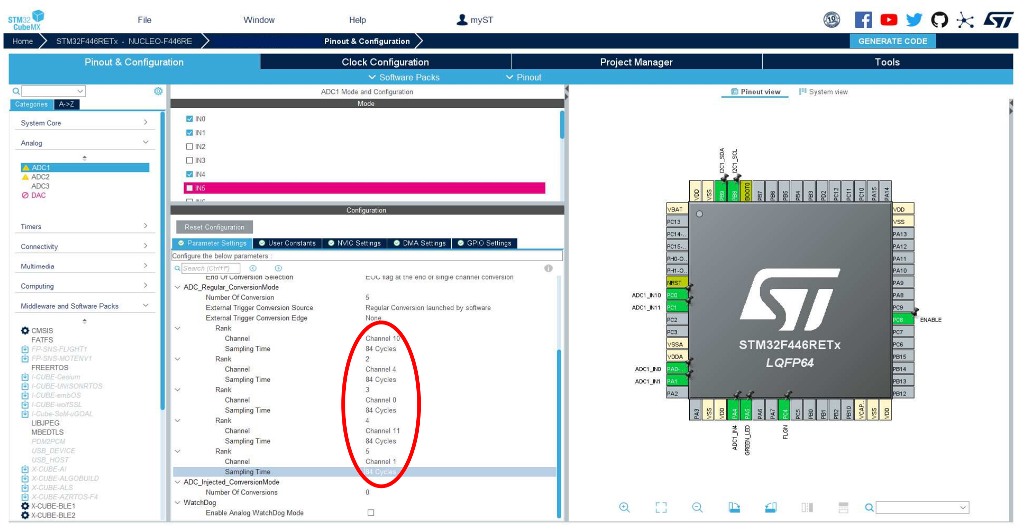


Figure 130 Source noPD – ADC Rank configuration

- As the TCPP02-M18 communicates via I2C, enable the I2C peripheral in the 'Connectivity' section, enable I2C1 peripheral, in I2C mode. Set I2C Speed Mode to 'Fast speed'

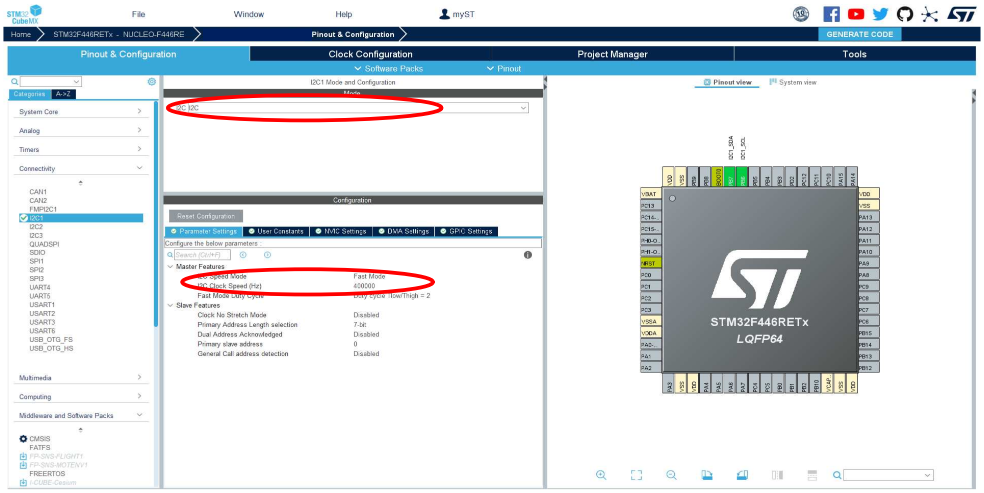


Figure 131 Source noPD – I2C configuration

Note: Select I2C1 and re-map its pin on PB8 and PB9 for a X-NUCLEO-SRC1M1 plugged on a NUCLEO-G071. Click left on these pins and select I2C1_SCL and I2C1_SDA.

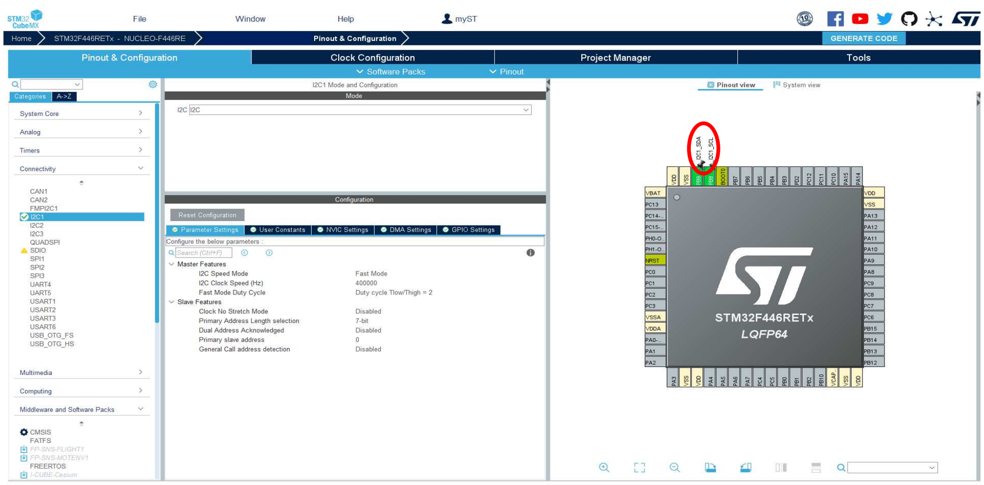


Figure 132 Source noPD – I2C re-map

- The TCPP02-M18 also needs a GPIO Output for its Enable input and an Interrupt input for its Alarm Flag FLGN output.

  On the pinout view, click left on your GPIO for Enable, and select GPIO Output. Then click right to name it as ‘ENABLE’.
  Note: Select PC8 for a X-NUCLEO-SRC1M1 plugged on a NUCLEO-G071.

  Also on the pinout view, click left on your GPIO for FLG input, and select GPIO_EXTI mode, and with a right click name it as ‘FLGN’.
  In the system-core section, select GPIO, and then change this EXTI pin mode to ‘External interrupt mode with falling edge detection’ with ‘Pull-up’.
  Note: Select PC4 (EXTI5) for a SRC1M1 plugged on a NUCLEO-F4XX.

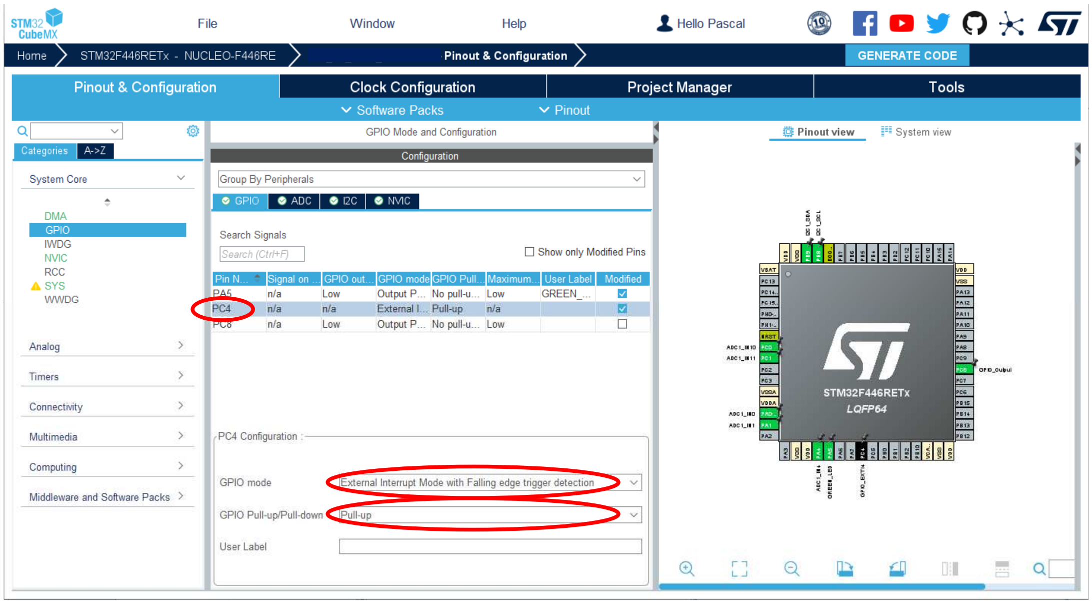


Figure 133 Source noPD - GPIO configuration

Back in the Middleware and Software pack category. Select the X-CUBE-TCPP item. Select the ‘Platform Settings’ tab.

Affect resources depending on your choices:

- For the application group, affect ADC Channels for CC1, CC2, Vbus, Vprov, ISense
- For the BSP group, affect the I2C bus, the Enable Output and the FLGN EXTI.

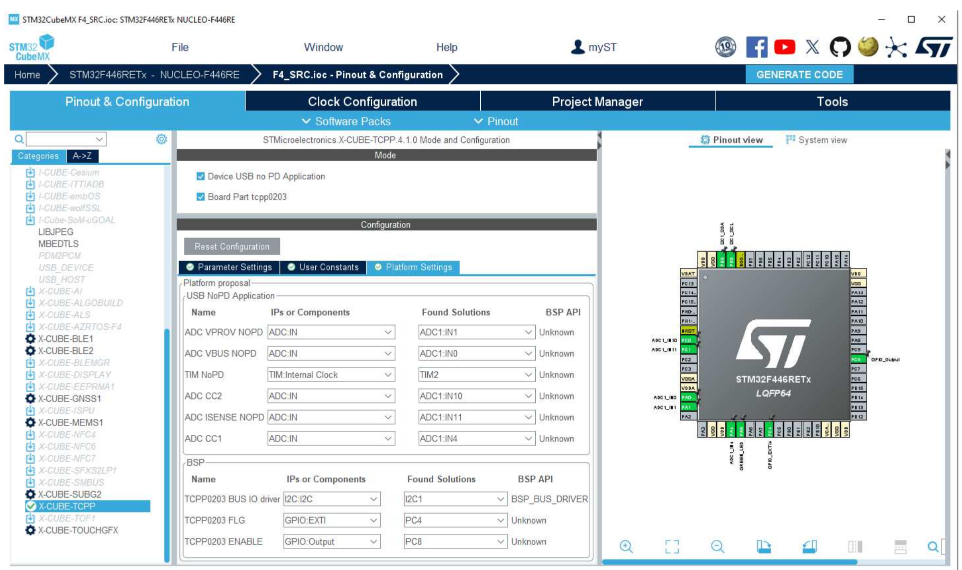


Figure 134 Source noPD – Platform settings

Under the Clock configuration main tab, change the system clock mux to PLLCLK, it will set HCLK clock to 84MHz.

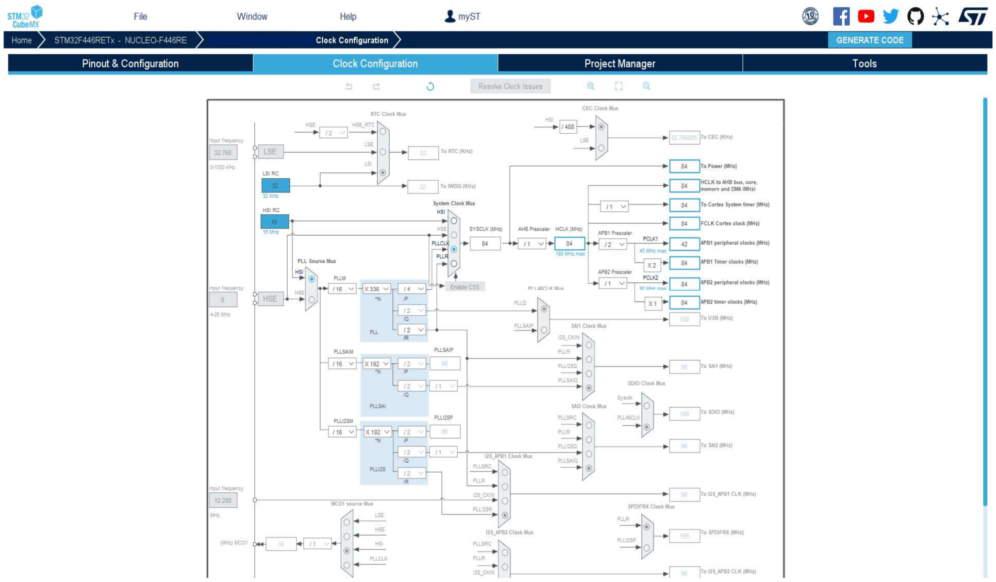


Figure 135 Source noPD – Clock configuration

in the Timers section, select timer2: "TIM2", affect the "Internal clock" as clock source. In the Parameter Settings tab:

- ‘The Internal clock division’ to No division; Then the timer peripheral frequency is 84MHz
- Set the ‘Prescaler’ value to 2099; Then the timer counter frequency is 84 / (2099+1) = 40kHz
- ‘The counter period’ to 39; Then the timer period is 40kHz / (39+1) = 1ms
- ‘Auto-reload’ preload to Enable;

Note: Adjust these settings for your MCU and its clock to obtain a 10ms timer period.

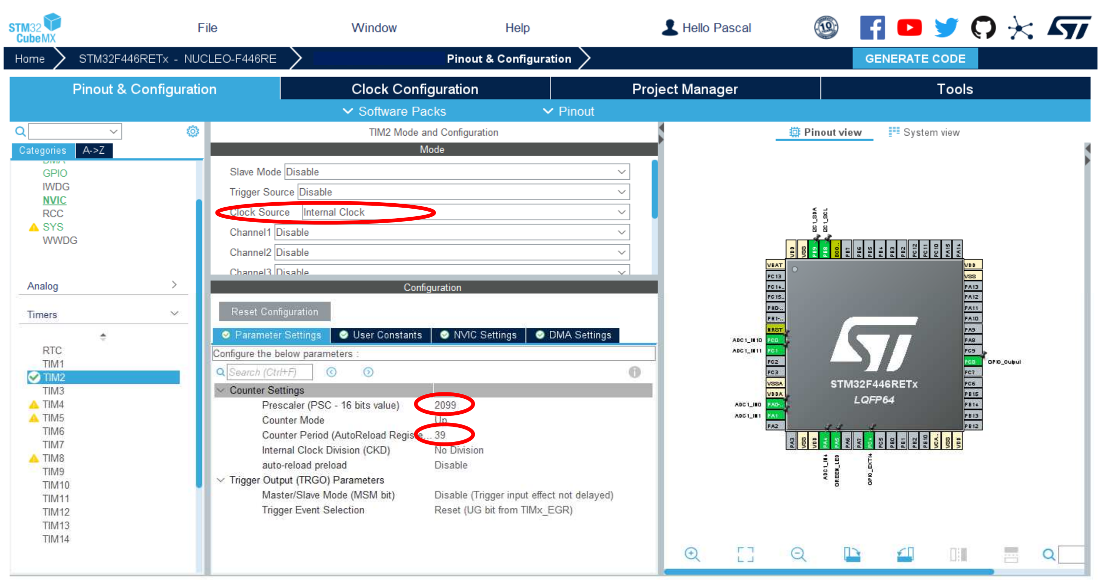


Figure 136 Source noPD – Timer configuration

Enable Tim2 interrupts

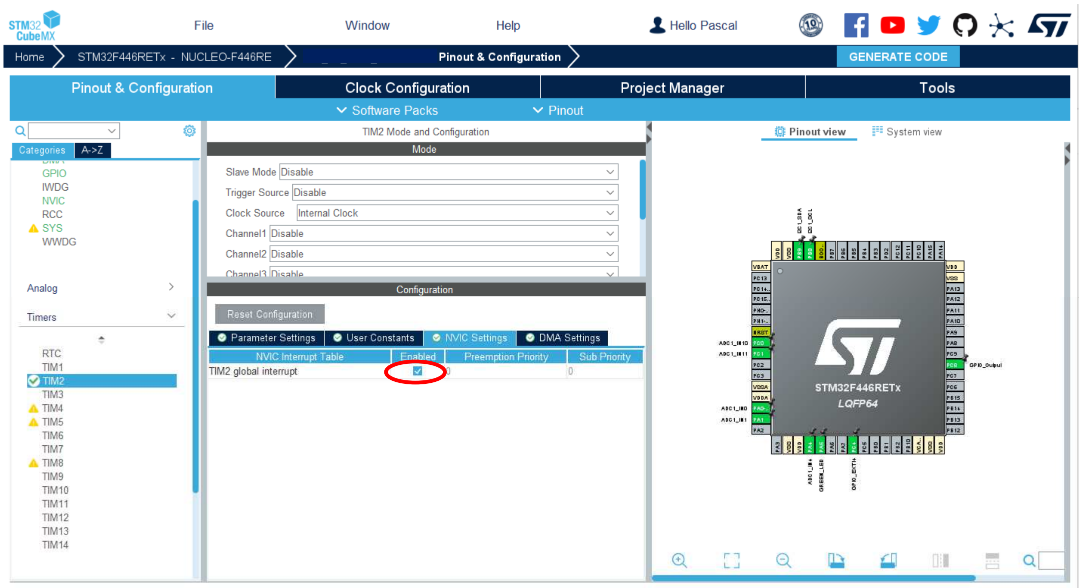


Figure 137 Source noPD – Timer interrupt activation

Generate the code for your IDE, for example STM32CubeIDE.

Save your file with Ctrl+s and select generate code if prompted. You can also generate code from the STM32CubeIDE menu, clicking on Project/Generate Code, or by pressing Alt+K.

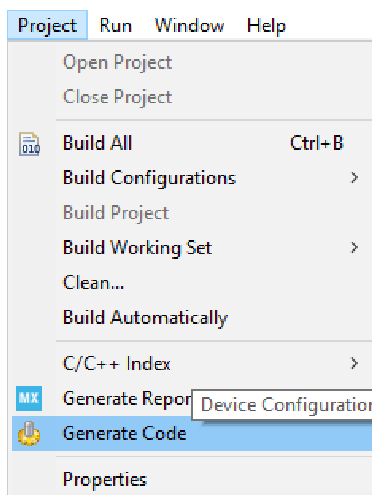


Figure 138 Source noPD – Code Generation

In your IDE, compile the code. It should be without any warning or error. And flash it in your application.

- The compilation must be performed without error or warnings.
- Build the application by clicking on the button (or select Project/Build Project).
- Run the application by clicking on the button (or select Run/Run)

Note: When plugging a SRC1M1 X-NUCLEO on a NUCLEO-F446, the TCPP02 FLG output is routed on a NC pin of the NUCLEO. Then place a link between PA3 (CN10 - 37) and PC4 (CN10 - 34).

Please adapt in function of your NUCLEO board.

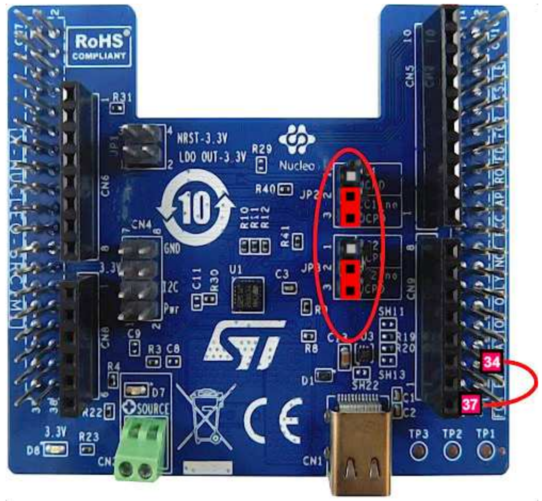


Figure 139 Source noPD – X-NUCLEO-SRC1M1 configuration

This X-NUCLEO-SRC1M1 shield default configuration allows SINK to source up to 0.5A @ 5V.

Plug an external 5V source with current capability >0.6A into the green "source" connector.

The current sense resistor R4 is 7mOhms, then TCPP02 current protection level is 6A. Refer to TCPP02-M18 datasheet. [5]

With this configuration, the board is powered by the ST-Link of the Nucleo board.

If you want to power your system from the external power supply connected to the "source" terminal, and not from the ST-Link, add the JP1 jumpers between 1-2 and 3-4.

Note: To increase the solution current capability to 3A @ 5V,

- Remove R35 and place it on SH19.
- Remove R39 and place it on SH21.
- Replace R4 sense resistor (initially 7mOhms) with a 10mOhms resistor.
- Next, plug an external 5V source with current capability > 4.5A into the green "source" connector.
- In SRC1M1_conf.h change SRC1M1_ISENSE_RS value from 7 milliohms to 10 milliohms:

  #define SRC1M1_ISENSE_RS 10u /* Current measure shunt resistor in milliohm */

7 Revision History

Table 5 Document revision history

Date	Version	Changes
16/01/23	1	Initial release

IMPORTANT NOTICE – PLEASE READ CAREFULLY