



---

## Getting started with the STMicroelectronics X-CUBE-SUBG2 software package for STM32CubeMX

---

### Introduction

This document provides the guidelines to configure and use the X-CUBE-SUBG2 software package V5.0.0 for STM32CubeMX (minimum required version V6.5.0). The document contains a description of the provided sample application, a description of the steps required to configure a generic project using the X-NUCLEO-S2868A1 or X-NUCLEO-S2868A2 or X-NUCLEO-S2915A1 expansion board with a Nucleo board, as well as a description of the steps to configure and use the sample application provided in the package.

Information and documentation related to the S2LP components, the X-NUCLEO-S2868A1 or X-NUCLEO-S2868A2 or X-NUCLEO-S2915A1 expansion board and the ST expansion software for S2LP are available on [www.st.com](http://www.st.com).

## Contents

<b>Introduction .....</b>	<b>1</b>
<b>Contents.....</b>	<b>2</b>
<b>List of figures.....</b>	<b>3</b>
<b>1 Acronyms and abbreviations.....</b>	<b>5</b>
<b>2 What is STM32Cube? .....</b>	<b>6</b>
<b>3 License .....</b>	<b>6</b>
<b>4 Sample Application Description .....</b>	<b>7</b>
<b>4.1 P2P (Point to Point) Application.....</b>	<b>7</b>
<b>4.2 Contiki-NG based Applications.....</b>	<b>7</b>
<b>4.3 Command Line Interface Application .....</b>	<b>7</b>
<b>4.4 FIFO RX/TX Applications.....</b>	<b>7</b>
<b>5 Installing the X-CUBE-SUBG2 pack in STM32CubeMX.....</b>	<b>7</b>
<b>6 Starting a new project .....</b>	<b>9</b>
<b>7 STM32 Configuration Steps.....</b>	<b>11</b>
<b>7.1 P2P Application based on X-NUCLEO-S2868A1 or X-NUCLEO-S2868A2 or X-NUCLEO-S2915A1 .....</b>	<b>13</b>
<b>7.2 Contiki-NG applications based on X-NUCLEO-S2868A1 or X-NUCLEO-S2868A2 or X-NUCLEO-S2915A1 .....</b>	<b>20</b>
<b>7.3 Command Line Interface application based on X-NUCLEO-S2868A1 or X-NUCLEO-S2868A2 or X-NUCLEO-S2915A1 .....</b>	<b>25</b>
<b>7.4 FIFO TX and FIFO RX Applications for Custom Board.....</b>	<b>29</b>
<b>7.5 Advanced configuration .....</b>	<b>33</b>
<b>8 Generated Folders Structure .....</b>	<b>34</b>
<b>9 Known Limitations and workarounds .....</b>	<b>35</b>
<b>10 References .....</b>	<b>38</b>
<b>11 Revision history .....</b>	<b>39</b>

## List of figures

<b>Figure 1</b> Managing embedded software packs in STM32CubeMX.....	8
<b>Figure 2</b> Installing the X-CUBE-SUBG2 pack in STM32CubeMX.....	8
<b>Figure 3</b> The X-CUBE-SUBG2 pack in STM32CubeMX.....	9
<b>Figure 4</b> STM32CubeMX main page.....	9
<b>Figure 5</b> STM32CubeMX <b>MCU/Board Selector</b> windows .....	10
<b>Figure 6</b> STM32CubeMX <b>Pinout &amp; Configuration</b> window .....	10
<b>Figure 7</b> STM32CubeMX <b>Software Pack Component Selector</b> window .....	11
<b>Figure 8</b> STM32 Nucleo 64 & Nucleo 144 with X-NUCLEO-S2868A1 .....	12
<b>Figure 9</b> X-NUCLEO-S2868A1 & X-NUCLEO-S2915A1 .....	12
<b>Figure 10</b> X-NUCLEO-S2868A1 pinout.....	13
<b>Figure 11</b> X-NUCLEO-S2915A1 pinout.....	13
<b>Figure 12</b> STM32CubeMX Software Pack Component Selector window example for S2915A1 .....	14
<b>Figure 13</b> STM32CubeMX Additional Software settings for P2P application: Mode and Configuration view a) for X-NUCLEO-S2915A1 b) for X-NUCLEO-S2868A2 (or X-NUCLEO-S2868A1) .....	16
<b>Figure 14</b> STM32CubeMX NVIC Configuration.....	19
<b>Figure 15</b> STM32CubeMX SPI Configuration.....	19
<b>Figure 16</b> S2-LP EXTI edge configuration .....	20
<b>Figure 17</b> Contiki-NG configuration for UDP applications.....	21
<b>Figure 18</b> STM32CubeMX Additional Software settings for Contiki-NG application: Mode and Configuration view.....	22
<b>Figure 19</b> STM32CubeMX TIM Configuration (left) and RTIMER parameter (right) .....	22
<b>Figure 20</b> STM32CubeMX USART Configuration.....	23
<b>Figure 21</b> Contiki-NG configuration for Border Router application.....	23
<b>Figure 22</b> Contiki-NG configuration for Serial Sniffer application .....	24
<b>Figure 23</b> Contiki-NG configuration not linked to any application.....	25
<b>Figure 24</b> CLI Application components.....	25
<b>Figure 25</b> CLI Application Platform Settings (X-NUCLEO-S2868A2 left, X-NUCLEO-S2915A1 right) .....	26
<b>Figure 26</b> DMA USART configuration for CLI Application, RX left, TX right .....	26
<b>Figure 27</b> U5 family: GPDMA1 configuration for USART for CLI Application, CH0 (RX) left, CH1 (TX) right.....	27
<b>Figure 28</b> CLI Application clock configuration .....	27
<b>Figure 29</b> S2-LP DK GUI Transmission Test .....	28
<b>Figure 30</b> S2-LP DK GUI Transmission Test between NUCLEO-F401RE and NUCLEO-L053R8.....	28
<b>Figure 31</b> CLI Application: Buffers sizes customization .....	29
<b>Figure 32</b> Component Selector configuration for FIFO RX and FIFO TX Applications or the Custom Board case .....	29
<b>Figure 33</b> Custom Board: radio frequency selection .....	30
<b>Figure 34</b> Default Parameter Settings for FIFO RX Application .....	31
<b>Figure 35</b> FIFO RX Platform Settings .....	31
<b>Figure 36</b> Modified Parameter Settings for FIFO RX Application (one GPIO).....	32
<b>Figure 37</b> FIFO RX Application: GPIOs Edge detection settings: IRQ Line left, AF Line right .....	32
<b>Figure 38</b> Preemption Priority/Sub Priority for FIFO RX Application: line 4 is used for AF .....	33
<b>Figure 39</b> Advanced settings.....	34
<b>Figure 40</b> STM32CubeMX <b>Application Structure</b> Configuration.....	34

<b>Figure 41</b> Generate Under Root option to be disabled .....	35
<b>Figure 42</b> Library to be enabled for Contiki-NG based projects from STM32CubeIDE 1.6.x.....	35
<b>Figure 43</b> Pack option to be disabled to flash some L0 based board.....	36
<b>Figure 44</b> S2-LP DK GUI: possible not precise information on NUCLEO or X-NUCLEO.....	36

# 1 Acronyms and abbreviations

*Table 1 List of acronyms*

Acronym	Description
GHz	Giga Hertz
P2P	Point-to-Point communication
RF	Radio frequency
HAL	Hardware Abstraction Layer
SPI	Serial peripheral interface
NVIC	Nested Vectored Interrupt Controller
WSN	Wireless sensors network
BSP	Board support package
LED	Light emitting diode
IPv6	Internet Protocol vers. 6
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
6LoWPAN	Ipv6 over Low -Power Wireless Personal Area Networks
RPL	Routing Protocol for Low-Power and Lossy Networks
MAC	Medium Access Control
CSMA	Carrier-sense multiple access
USART	Universal Synchronous Asynchronous Receiver Transmitter

## 2

# What is STM32Cube?

STM32Cube™ represents an original initiative by STMicroelectronics to ease developers' life by reducing development effort, time and cost. STM32Cube covers the STM32 portfolio. Version 1.x of STM32Cube includes:

- STM32CubeMX, a graphical software configuration tool that allows the generation of C initialization code using graphical wizards.
- A comprehensive embedded software platform, delivered per series (such as the STM32CubeF4 for STM32F4 series).
  - STM32Cube HAL, an STM32 abstraction layer embedded software, ensuring maximized portability across the STM32 portfolio;
  - a consistent set of middleware components, such as RTOS, USB, TCP/IP, graphics;
  - all embedded software utilities, including a full set of examples.

## 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-SUBG2 pack is provided. For more info kindly refer User Manual of X-CUBE-SUBG2 available on [www.st.com](http://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-S2868A1 or X-NUCLEO-S2868A2 or X-NUCLEO-S2915A1 expansion board.
- show the users how to use the APIs to correctly initialize and use the S2-LP

## 4.1

### P2P (Point to Point) Application

This application provides a Point-to-Point communication example which involves sending a buffer from one node to another and acknowledgments using the features in S2LP.

## 4.2

### Contiki-NG based Applications

Contiki-NG [2] is a middleware that provides 6LoWPAN communication on top of the S2-LP radio. This involves MAC layers, RPL routing protocols, 6LoWPAN adaptation sub-layer, IPv6 network layer, UDP/TCP and above protocols.

- UDP Client: sends periodic UDP messages to a UDP Server and receives a UDP response (if any).
- UDP Server: receives UDP messages and replies with another UDP message.
- Border Router: connects the wireless 6LoWPAN with the Ipv6 protocol stack of a host PC.
- Serial Sniffer: captures RF packets and sends them via serial line to Wireshark application running on a host PC.

## 4.3

### Command Line Interface Application

This example is ported from S2-LP DK [3], it can be used together with the GUI delivered in the S2-LP DK to configure and test the S2-LP radio.

## 4.4

### FIFO RX/TX Applications

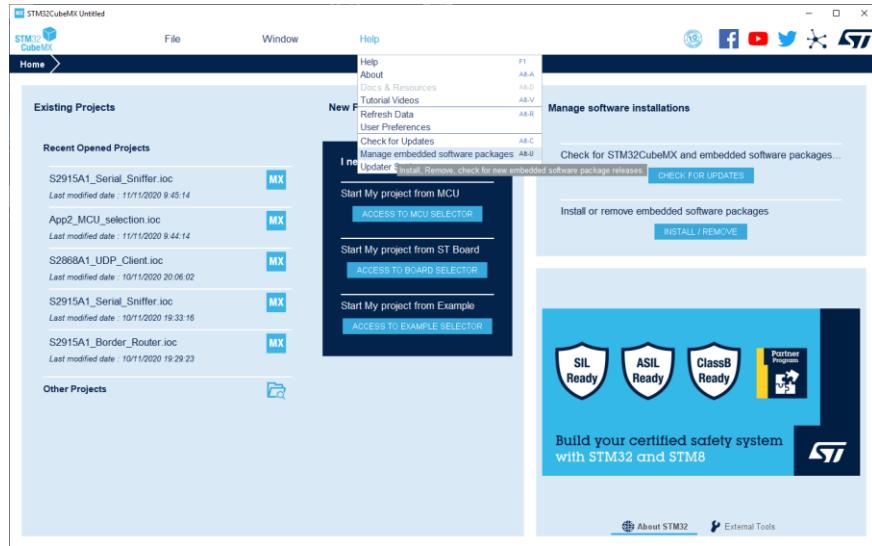
These examples are ported from S2-LP DK [3], they can be used together to demonstrate transmission and reception of packets of any length (more than the 128 bytes radio payload), and usage of two GPIOs configured in EXTI mode (FIFO RX). Currently they are supported only in the Custom Board case.

## 5

# Installing the X-CUBE-SUBG2 pack in STM32CubeMX

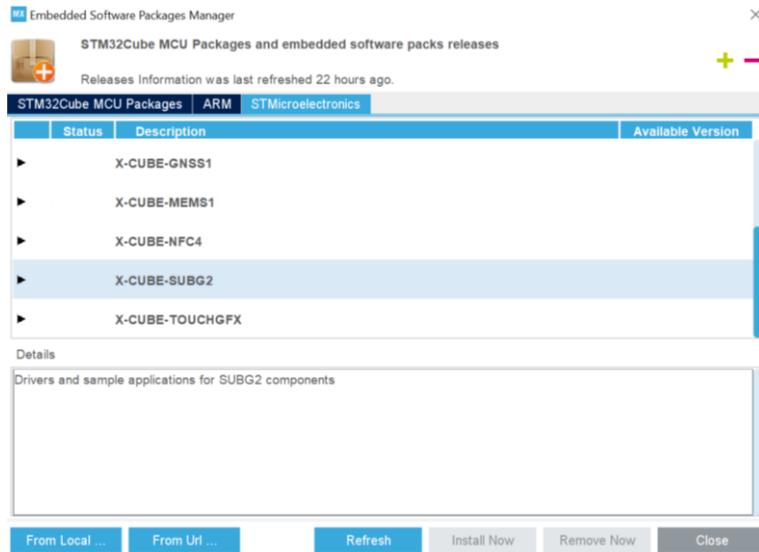
After downloading (from [www.st.com](http://www.st.com)), installing and launching the STM32CubeMX (V $\geq$ 6.5.0), the X-CUBE-SUBG2 pack can be installed in few steps.

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



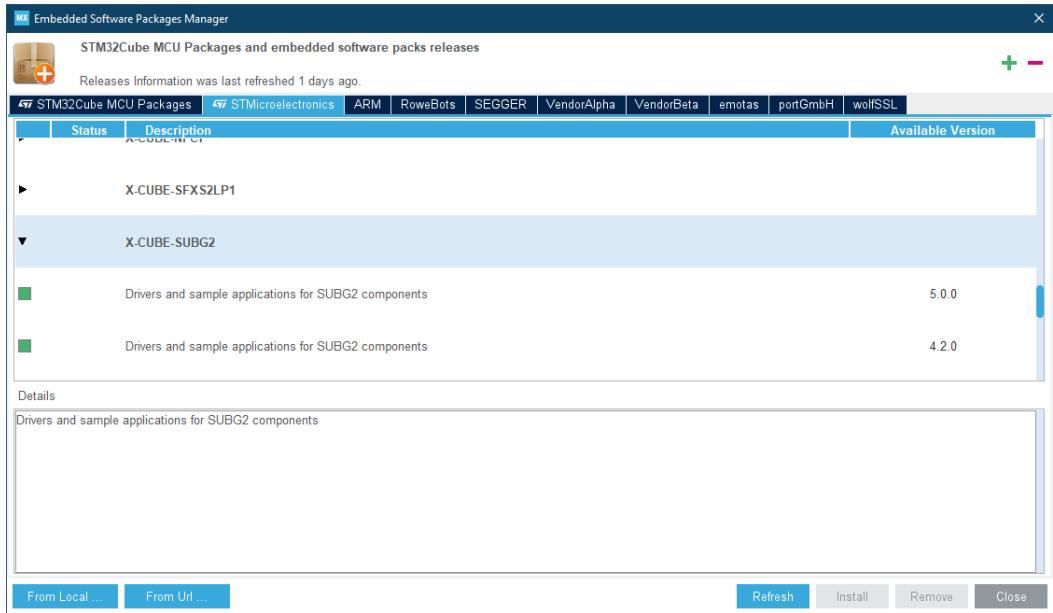
**Figure 1** Managing embedded software packs in STM32CubeMX

- From the Embedded Software Packages 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-SUBG2 pack.



**Figure 2** Installing the X-CUBE-SUBG2 pack 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.

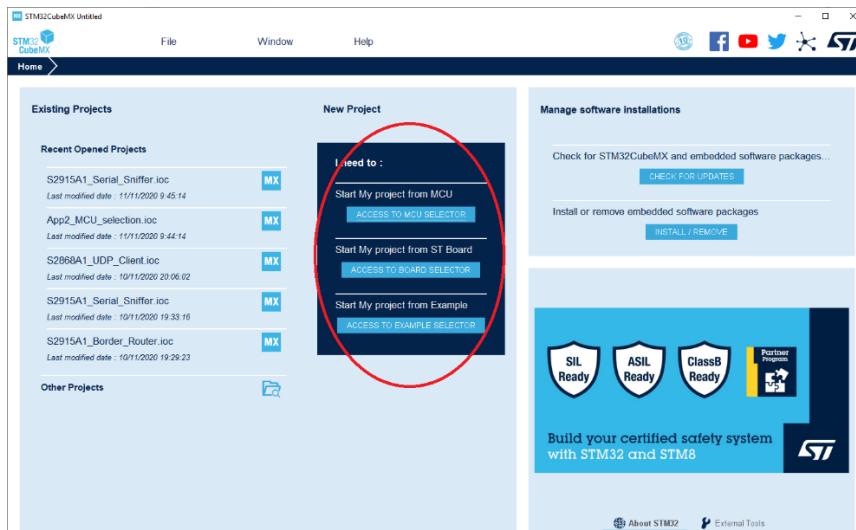


**Figure 3** The X-CUBE-SUBG2 pack in STM32CubeMX

## 6

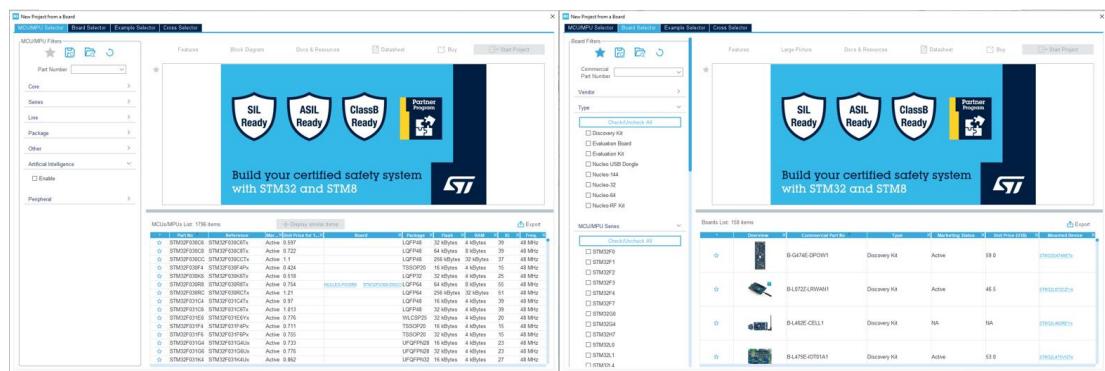
## Starting a new project

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



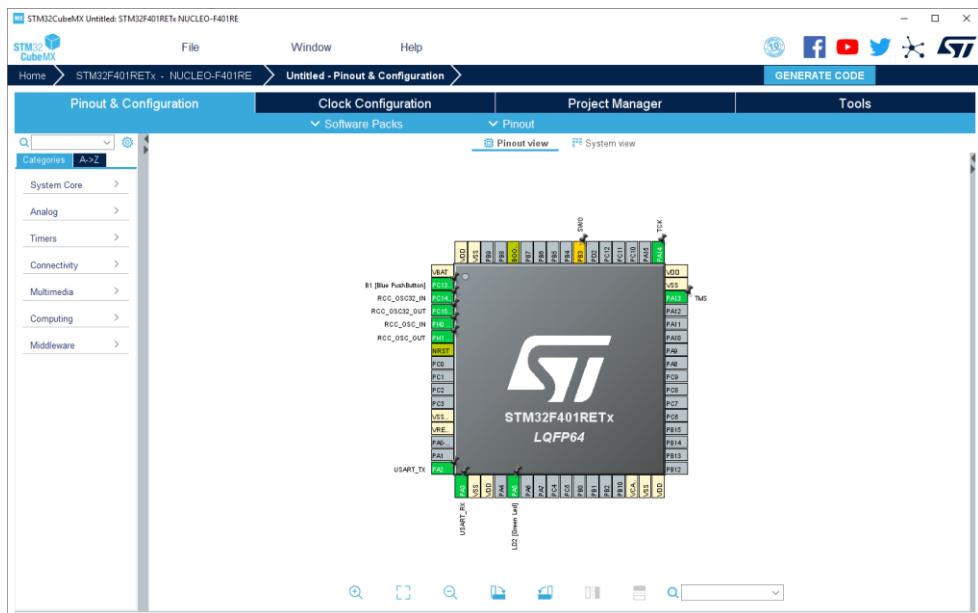
**Figure 4** STM32CubeMX main page

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



**Figure 5 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 6 STM32CubeMX Pinout & Configuration window**

To add the X-CUBE-SUBG2 additional software to the project, the “Software Packs” and then “Select Components” button must be clicked.

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.

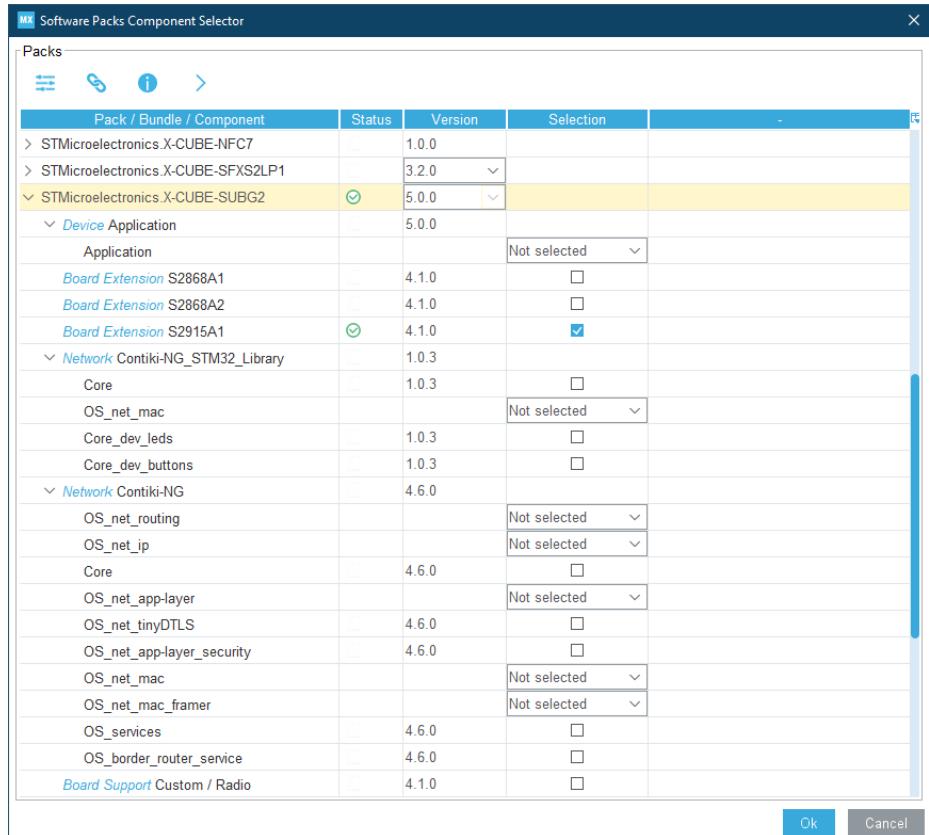


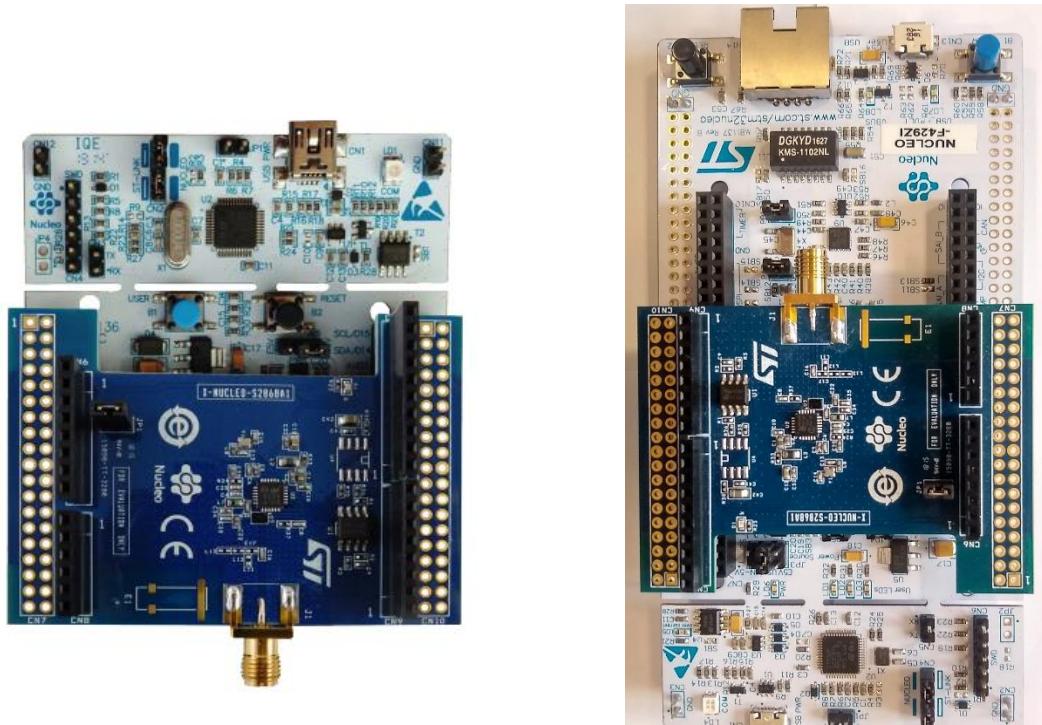
Figure 7 STM32CubeMX Software Pack Component Selector window

## 7

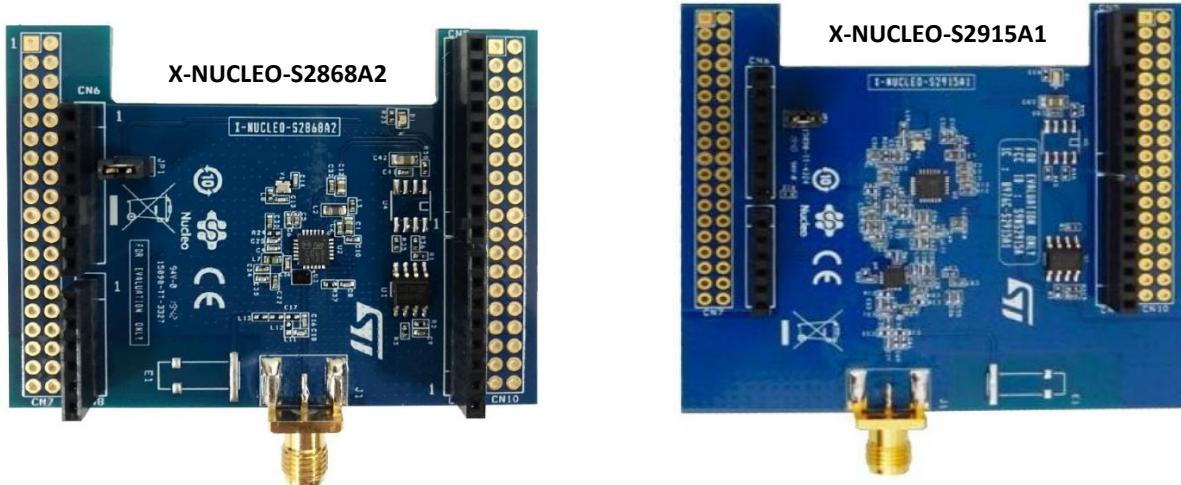
## STM32 Configuration Steps

The X-NUCLEO-S2868A1 or X-NUCLEO-S2868A2 or X-NUCLEO-S2915A1 interfaces with the STM32 microcontroller via the SPI bus.

Hence, assuming a user wants to interface the ST X-NUCLEO-S2868A1 or X-NUCLEO-S2868A2 or X-NUCLEO-S2915A1 expansion board with a STM32 Nucleo 64 pins board (e.g. a Nucleo-F401RETx) no particular hardware modification must be done.



**Figure 8** STM32 Nucleo 64 & Nucleo 144 with X-NUCLEO-S2868A1



**Figure 9** X-NUCLEO-S2868A1 & X-NUCLEO-S2915A1

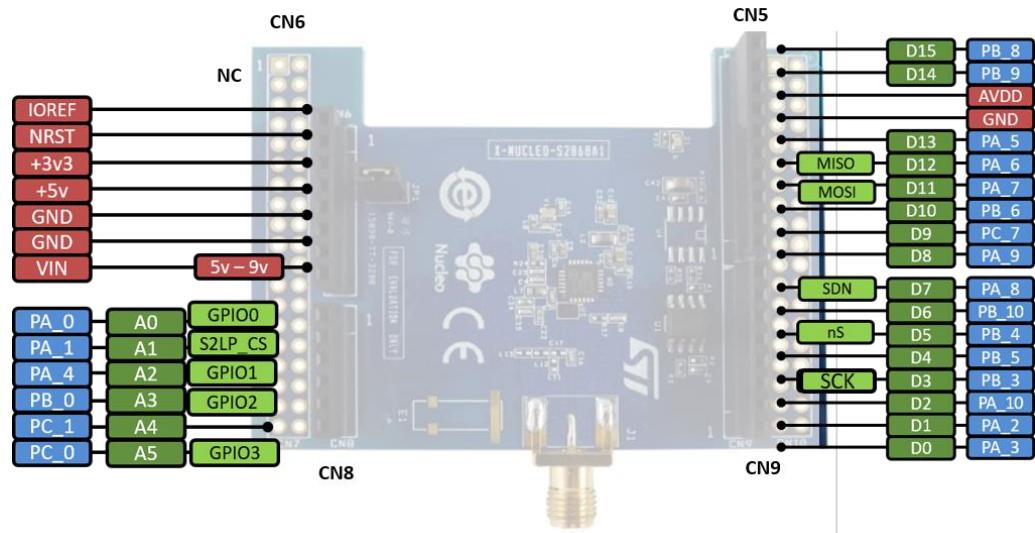


Figure 10 X-NUCLEO-S2868A1 pinout

Note that the X-NUCLEO-S2868A1 and X-NUCLEO-S2868A2 have the same pinout. However, the pinout of the X-NUCLEO-S2915A1 are different and depicted in Figure 11.

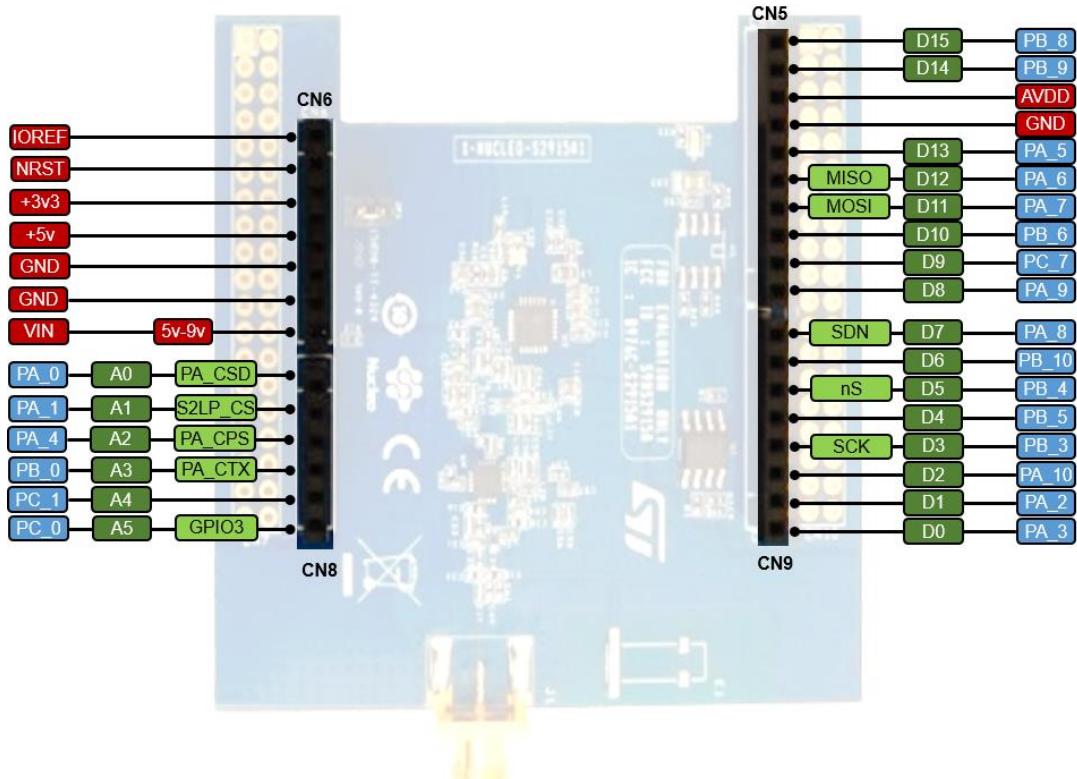


Figure 11 X-NUCLEO-S2915A1 pinout

## 7.1 P2P Application based on X-NUCLEO-S2868A1 or X-NUCLEO-S2868A2 or X-NUCLEO-S2915A1

This section outlines how to configure STM32CubeMX with X-NUCLEO-S2868A1 or X-NUCLEO-S2868A2 or X-NUCLEO-S2915A1 when the use of the P2P sample example is required. With such setup, only driver layers will be configured.

To add the X-CUBE-SUBG2 additional software to the project, the “Software Packs” and then

“Select Components” button must be clicked. From the “Software Pack Component Selector” window, the user has to select the example from the “Device” class and “Board Extension” class as shown in the figure below.

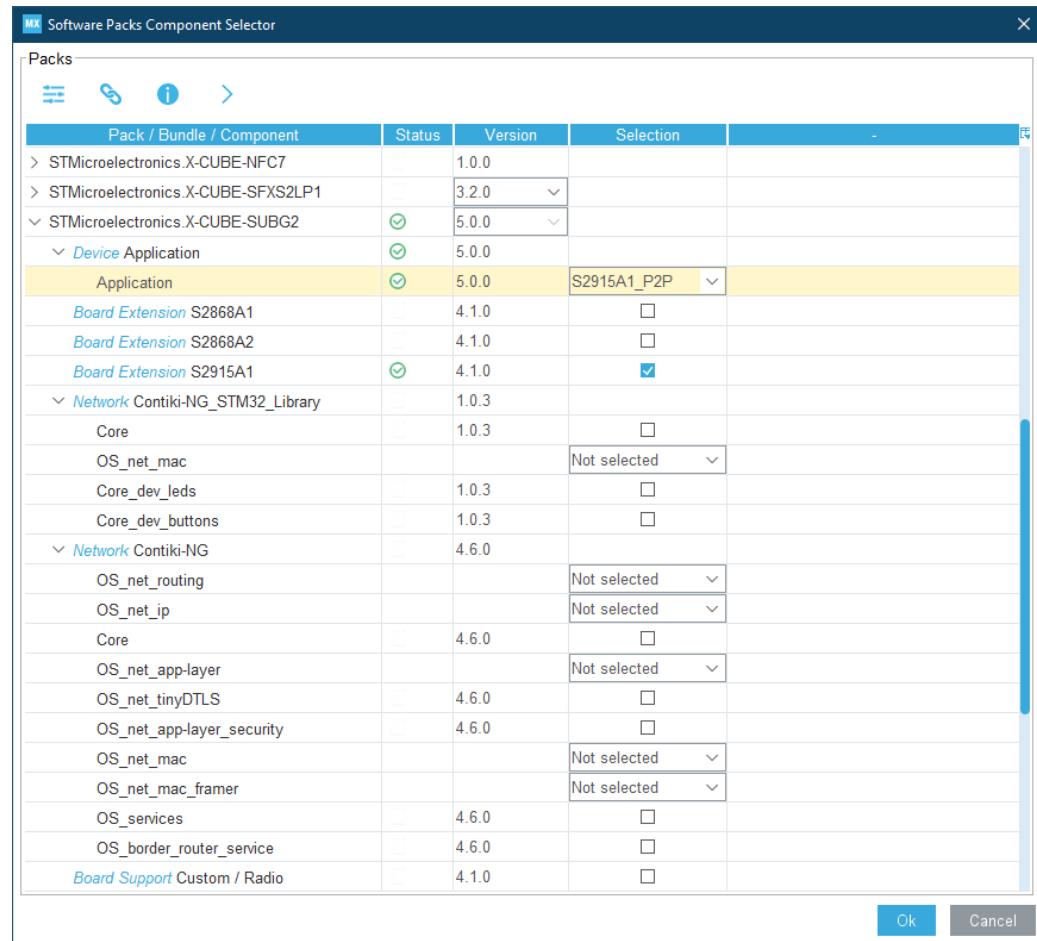


Figure 12 STM32CubeMX Software Pack Component Selector window example for S2915A1

From the **Pinout & Configuration** tab, set the SPI according to Table 2.

Group	Nucleo boards	SPI_SCK	SPI_MISO	SPI_MOSI	SPI_Instance
<b>BG1</b>	F030R8, F070RB, F072RB, F091RC, F334R8, F303RE, F401RE, F411RE, F446RE, F410RB, L010RB, L053R8, L073RZ, L152RE, L452RE, L476RG, G070RB, G071RB, G431RB, G474RE	“PB3”	“PA6”	“PA7”	SPI1
<b>BG2</b>	F103RB	“PA5”	“PA6”	“PA7”	SPI1
<b>BG3</b>	F302R8 L412RB-P, L452RE-P,	“PB13”	“PB14”	“PB15”	SPI2

Group	Nucleo boards	SPI_SCK	SPI_MISO	SPI莫斯I	SPI_Instance
	L433RC-P				
<b>BG4</b>	F429ZI, F446ZE, F207ZG, F746ZG, F767ZI, H743ZI, F722ZE, F303ZE, F412ZG, F413ZH, L552ZE-Q, WB-55	“PA5”	“PA6”	“PA7”	SPI1
<b>BG5</b>	L496ZG-P, L496ZG, L4R5ZI, L4R5ZI-P	“PE13”	“PA6”	“PA7”	SPI1
<b>BG6</b>	H753ZI, H743ZI2 H745ZI-Q, H755ZI-Q	“PA5”	“PA6”	“PB5”	SPI1

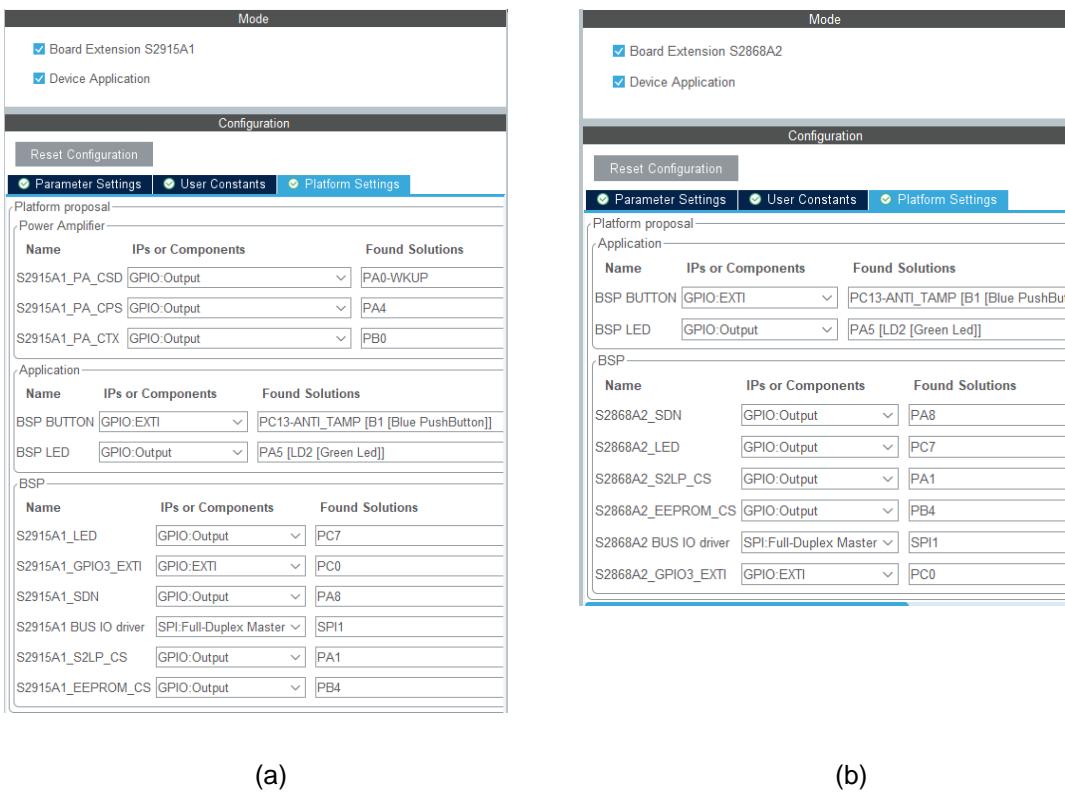
*Table 2 SPI Configurations for different Nucleo Boards*

For example, using the Nucleo F401RE,

- from the **Pinout** scheme, click on PA6 and set it as SPI1\_MISO;
- from the **Pinout** scheme, click on PA7 and set it as SPI1\_MOSI;
- from the **Pinout** scheme, click on PB3 and set it as SPI1\_SCK;
- enable the SPI1 as SPI from the “Connectivity” category;

While using X-NUCLEO-S2868A1 or X-NUCLEO-S2915A1 or X-NUCLEO-S2868A2 with boards under “**BG2**”, “**BG3**”, “**BG4**” and “**BG6**” groups presented in Table 2 (i.e., F103RB, F302R8, L412RB-P, L452RE-P, L433RC-P, F429ZI, F446ZE, F207ZG, F746ZG, F767ZI, H743ZI, F722ZE, F303ZE, F412ZG, F413ZH, L552ZE-Q, WB-55, H753ZI, H743ZI2, H745ZI-Q and H755ZI-Q), the SPI clock (i.e., SPI\_SCK) is available on pin D13 instead of D3. In these cases, the following modification are required to correctly set the SPI clock.

- Remove R11 and Connect R6.



(a)

(b)

**Figure 13** STM32CubeMX Additional Software settings for P2P application: Mode and Configuration view a) for X-NUCLEO-S2915A1 b) for X-NUCLEO-S2868A2 (or X-NUCLEO-S2868A1)

As shown from Figure 12, the power amplifier configurations are only needed for the X-NUCLEO-S2915A1.

From the **Software Packs** category, press the ‘STMicroelectronics.X-CUBE-SUBG2.5.0.0’ item, enable the “Board Extension” checkbox from the “Mode” view and set the following Platform Settings from the “Configuration” view (take into account that according the example chosen some settings can appear or not) based on Table 3.

**Table 3** Platform Settings configurations

Name	BSP_API	Supported IPs	Nucleo 64	Nucleo 144
<b>S2868A1</b>	<b>S2868A2</b>	<b>S2915A1</b>		
S2868A1 BUS IO driver	S2868A2 BUS IO driver	S2915A1 BUS IO driver	BSP_BUS_DRIVER	SPI: SPI
S2868A1_GPIO3_EXTI	S2868A2_GPIO3_EXTI	S2915A1_GPIO3_EXTI	HAL_EXTI_DRIVER	GPIO:EXTI
S2868A1_S2LP_CS	S2868A2_S2LP_CS	S2915A1_S2LP_CS		GPIO:Output
				A1

Name			BSP_API	Supported Ips	Nucleo 64	Nucleo 144
S2868A1	S2868A2	S2915A1				
S2868A1_SDN	S2868A2_SDN	S2915A1_SDN		GPIO:Output	D7	
BSP BUTTON	BSP BUTTON	BSP BUTTON	BSP_COMMON_D_RIVER	GPIO: EXTI	PC13	
BSP LED	BSP LED	BSP LED		GPIO:Output	PA5(PB13)	PB7
S2868A1_EEPROM_CS	S2868A2_EEPROM_CS	S2915A1_EEPROM_CS	BSP_COMMON_D_RIVER	GPIO:Output	D5	
S2868A1_LED	S2868A2_LED	S2915A1_LED		GPIO:Output	D9	
\		S2915A1_PA_CSD		GPIO:Output	A0	
/		S2915A1_PA_CPS		GPIO:Output	A2	
X		S2915A1_PA_CTX		GPIO:Output	A3	

Table 4 presents the “BSP LED” pin names for different Nucleo Boards.

*Table 4 BSP LED pin names for different Nucleo Boards*

Nucleo Board	BSP LED
F030R8, F070RB, F072RB, F091RC, F103RB, F303RE, F334R8, F401RE, F410RB, F411RE, F446RE, L010RB, L053R8, L073RZ, L152RE, L452RE, L476RG	“PA5”
F302R8, L452RE-P, L433RC-P, L412RB-P	“PB13”
WB-55	“PB0”
Nucleo-144 boards	“PB7”

For the P-NUCLEO-WB55 board, the “BSP BUTTON” should be set to “PC4” and not to “PC13”.

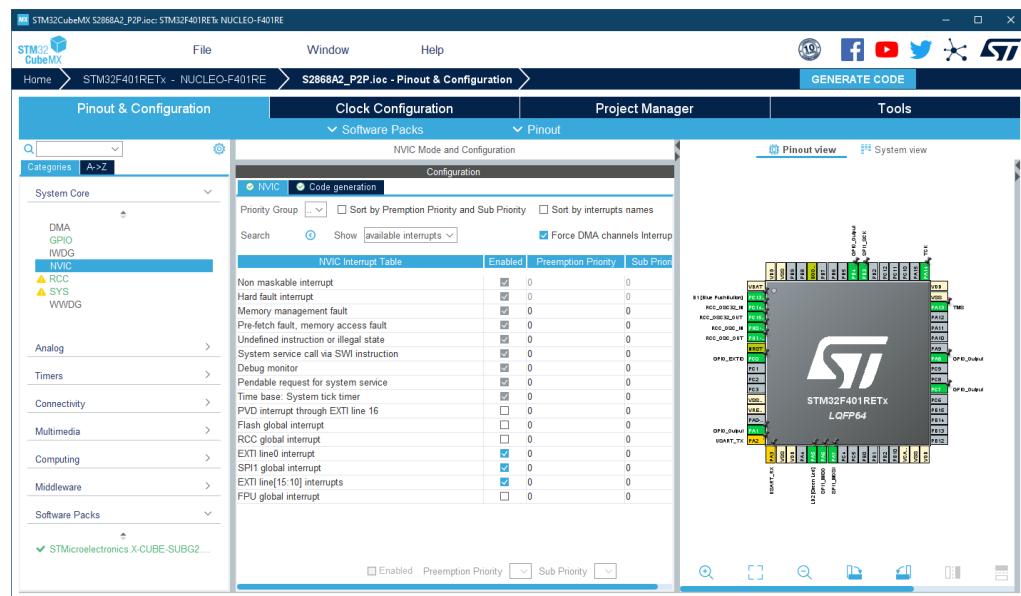
The A0, A1, A2, A3, A5, D9, D7 and D5 pins depend on the Nucleo Board and their appropriate names are recapitulated in Table 5.

*Table 5 A0, A1, A2, A3, A5, D9, D7 and D5 pin names for different Nucleo Boards*

Boards	A0	A1	A2	A3	A5	D9	D7	D5
1) F030R8								
2) F070RB								
3) F072RB								
4) F091RC								
5) F303RE								
6) F334R8								
7) F401RE								
8) F410RB	PA0	PA1	PA4	PB0	PC0	PC7	PA8	PB4
	PA0-WKUP							

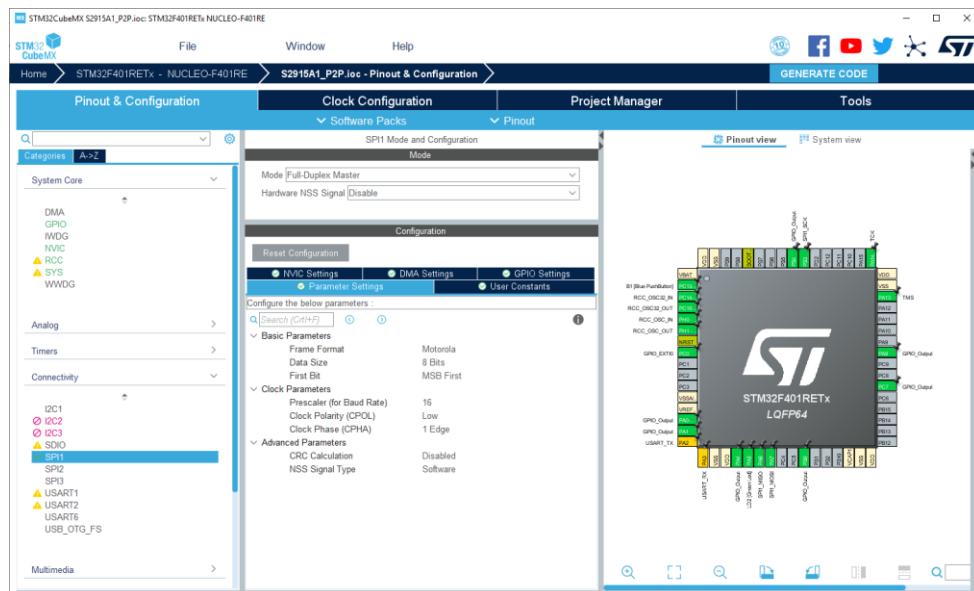
<b>9) F411RE</b>									
<b>10) F446RE</b>									
<b>11) L053R8</b>	PA0								
<b>12) L073RZ</b>									
<b>13) L152RE</b>	PA0-WKUP1								
<b>14) L452RE</b>	PA0								
<b>15) L476RG</b>									
<b>16) F302R8</b>	PA0	PA1	PA4	PB0	PC0	PC7	PA8	PB4	
<b>17) L010RB</b>									
<b>18) G070RB</b>	PA0	PA1	PA4	PB1	PB12	PC7	PA8	PB4	
<b>19) G071RB</b>									
<b>20) G431RB</b>									
<b>21) G474RE</b>									
<b>22) L412RB-P</b>	PA0	PA1	PC3	PC2	PC0	PA8	PC7	PA15 (JTDI)	
<b>23) L452RE-P</b>									
<b>24) L433RC-P</b>									
<b>25) F103RB</b>	PA0-WKUP	PA1	PA4	PB0	PC0	PC7	PA8	PB4	
<b>26) F207ZG</b>	PA3	PC0	PC3	PF3	PF10	PD15	PF13	PE11	
<b>27) F429ZI</b>									
<b>28) F446ZE</b>									
<b>29) F722ZE</b>									
<b>30) F746ZG</b>									
<b>31) F767ZI</b>									
<b>32) H743ZI</b>			PC3_C						
<b>33) L496ZG</b>	PA3	PC0	PC3	PC1	PC5	PD15	PF13	PE11	
<b>34) L496ZG-P</b>									
<b>35) L4R5ZI</b>									
<b>36) L4R5ZI-P</b>									
<b>37) F412ZG</b>									
<b>38) F413ZH</b>									
<b>39) F303ZE</b>	PA3	PC0	PC3	PD11	PD13	PD15	PF13	PE11	
<b>40) H753ZI</b>	PA3	PC0	PC3_C	PB1	PF10	PD15	PG12	PE11	
<b>41) L552ZE-Q</b>	PA3	PA2	PC3	PB0	PC0	PD15	PF13	PE11	
<b>42) WB55</b>	PC0	PC1	PA1	PA0	PC2	PA9	PC13	PA15	

From the **Pinout & Configuration** tab, click on “System Core” category and then on NVIC item to enable the EXTI line interrupts.



**Figure 14** STM32CubeMX NVIC Configuration

From the **Pinout & Configuration** tab, click on “Connectivity” category and then select SPI instance and set the SPI Prescaler (for Baud Rate), SPI instance and prescaler will vary depending on the pinout and Clock configuration of STM32. Maximum SPI Clock speed supported by S2LP is 10 MHz. Hence, the prescaler (for Baud rate) should be such that Clock Speed on STM32 NUCLEO is less than 10Mhz. In Figure 13, NUCLEO-F401RE is taken into account, since HCLK is 84 MHZ and so the prescaler set for SPI is 16.



**Figure 15** STM32CubeMX SPI Configuration

From the **Pinout & Configuration** tab, click on “System view” / “GPIO” and select the PIN used for main S2-LP radio interrupt (S2868A1\_GPIO3\_EXTI/ S2868A2\_GPIO3\_EXTI/ S2915A1\_GPIO3\_EXTI by default). Change GPIO mode to be “External Interrupt Mode with Falling edge trigger detection”. This is required because from X-CUBE-SUBG2 5.0.0 (for

STM32CubeMX 6.5.0) all the GPIO settings are retrieved from the configuration done in STM32CubeMX, without any internal re-initialization like in previous versions of X-CUBE-SUBG2.

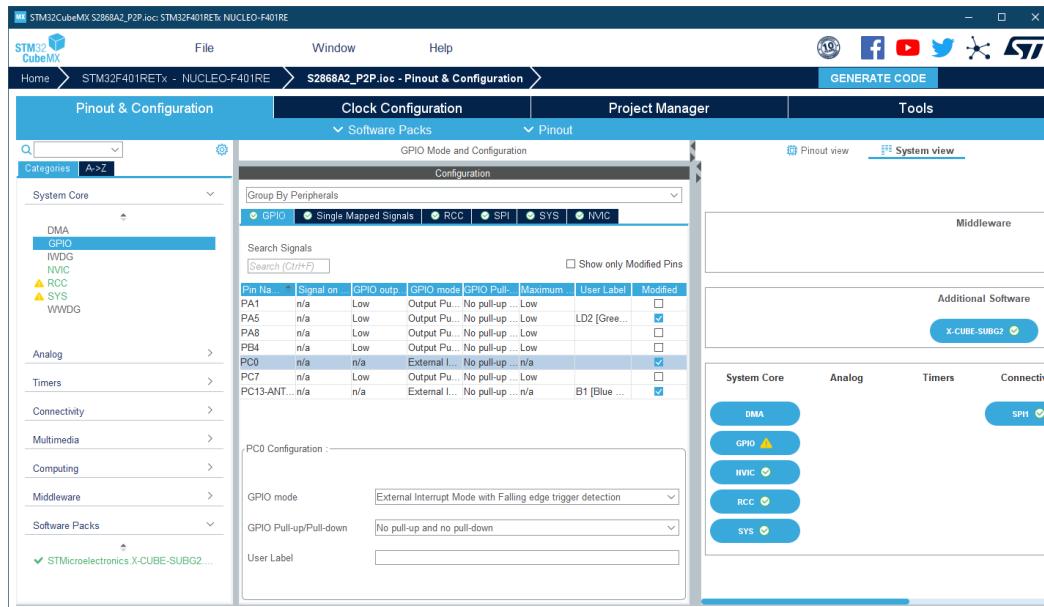


Figure 16 S2-LP EXT1 edge configuration

Once all the above described steps have been performed, the source code of the project using the **STMicroelectronics.X-CUBE-SUBG2** software can be generated clicking the “GENERATE CODE” button.

## 7.2 Contiki-NG applications based on X-NUCLEO-S2868A1 or X-NUCLEO-S2868A2 or X-NUCLEO-S2915A1

All the configurations steps described in the previous section apply as the base also for the Contiki-NG based application, for what concerns the Power Amplifier and the BSP.

In this case, more components can be selected in order to have the applications working.

For any Contiki-NG based application to work, three components **must** always be selected:

- Contiki-NG\_STM32\_Library / Core
- Contiki-NG / Core
- Contiki-NG / OS\_services

There are other components that are optional or can be configured. We can highlight a common suggested configuration for UDP Client and UDP Server. In **Figure 17** this configuration is reported:

- Contiki-NG / OS\_net\_routing configured as **rpl\_lite**
- Contiki-NG / OS\_net\_ip configured as **Ipv6**
- Contiki-NG / OS\_net\_mac configured as **csma**
- Contiki-NG / OS\_net\_mac\_framer configured as **framer\_802\_15\_4**

This is the common configuration meaning that Ipv6 protocol is enabled, “lite” version of the RPL routing protocol is used, Contiki-NG CSMA protocol is chosen for mac layer and standard 802.15.4 framer layer is selected.

This configuration can be used also to build other applications based on Contiki-NG.

It is important to note that the configuration of these 4 components must match in all the nodes to make them able to communicate.

The user can optionally choose to change the RPL routing protocol version by using:

- Contiki-NG / OS\_net\_routing configured as **rpl\_classic**  
but the same choice must be done for all the nodes to communicate.

The configuration of the following components (that enable usage of LED and Button):

- Contiki-NG\_STM32\_Library / Core\_dev\_leds
  - Contiki-NG\_STM32\_Library / Core\_dev\_buttons
- is optional for UDP applications.

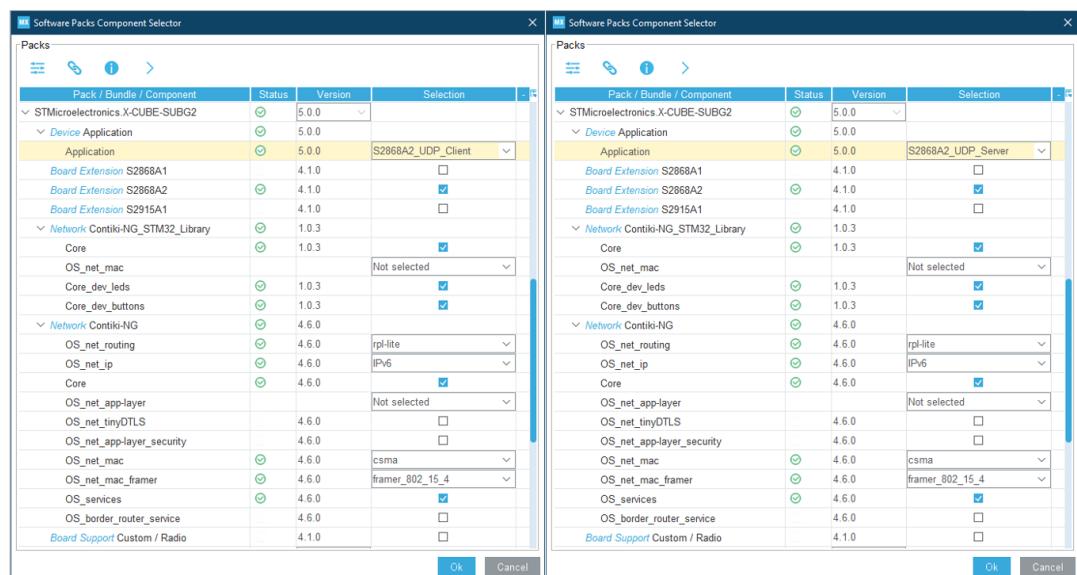


Figure 17 Contiki-NG configuration for UDP applications

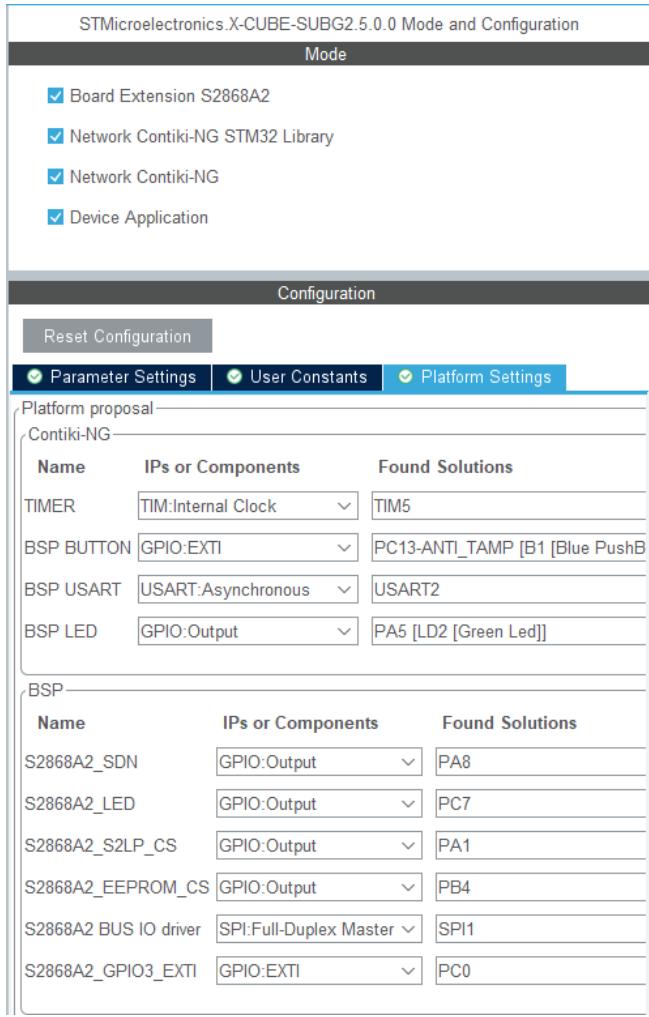
From the **Software Packs** category, press the ‘STMicroelectronics.X-CUBE-SUBG2.5.0.0’ item, enable the “Board Extension”, “Contiki-NG” related and “Application” checkboxes from the “Mode” view. For the BSP parameters to be set Platform Settings from the “Configuration” view, you can check previous section (related to P2P Example).

In the Contiki-NG case, User Button and LED (if enabled) can be found in the Contiki-NG section.

As shown in **Figure 18** the two main differences are the TIMER and the USART, that must be selected for Contiki-NG based applications.

From the **Pinout & Configuration** tab, click on “Timers” category and then select TIM instance and enable the Internal Clock checkbox. No other configuration is needed since the actual initialization will be done from the application (taking into account the RTIMER\_ARCH\_SECOND parameter to compute the prescaler: default value should be left). **Figure 19** shows the TIM final configuration.

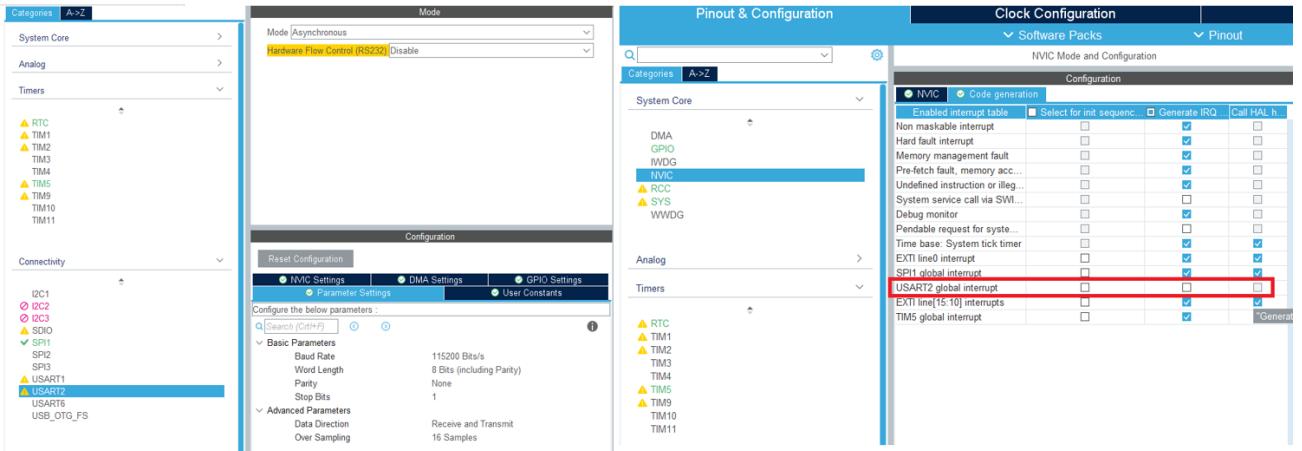
From the **Pinout & Configuration** tab, click on “Connectivity” category and then select USART instance and enable it by choosing **Asynchronous** mode. Baud Rate=115200 bps, Word Length=8 bits, Parity=None and Stop Bits=1 parameters must be chosen. It is important to go in “System Core” category and then on NVIC, “Code Generation” tab and ensure no code generation is selected for the chosen USART, since the application will take care of it. **Figure 20** shows the USART final configuration (left) and the NVIC settings (right).



**Figure 18** STM32CubeMX Additional Software settings for Contiki-NG application: Mode and Configuration view

This figure displays two screenshots of the STM32CubeMX configuration interface. The left screenshot shows the 'Timers' configuration for TIM5, with various modes and channel settings. The right screenshot shows the 'RTIMER' parameter configuration, specifically for the 'Basic Settings' of the S2868A2 timer, including GPIO3 EXT1 line, GPIO1 Configuration, and GPIO2 Configuration.

**Figure 19** STM32CubeMX TIM Configuration (left) and RTIMER parameter (right)



**Figure 20 STM32CubeMX USART Configuration**

For the Border Router application the same base configuration of the UDP applications is used, adding the following component:

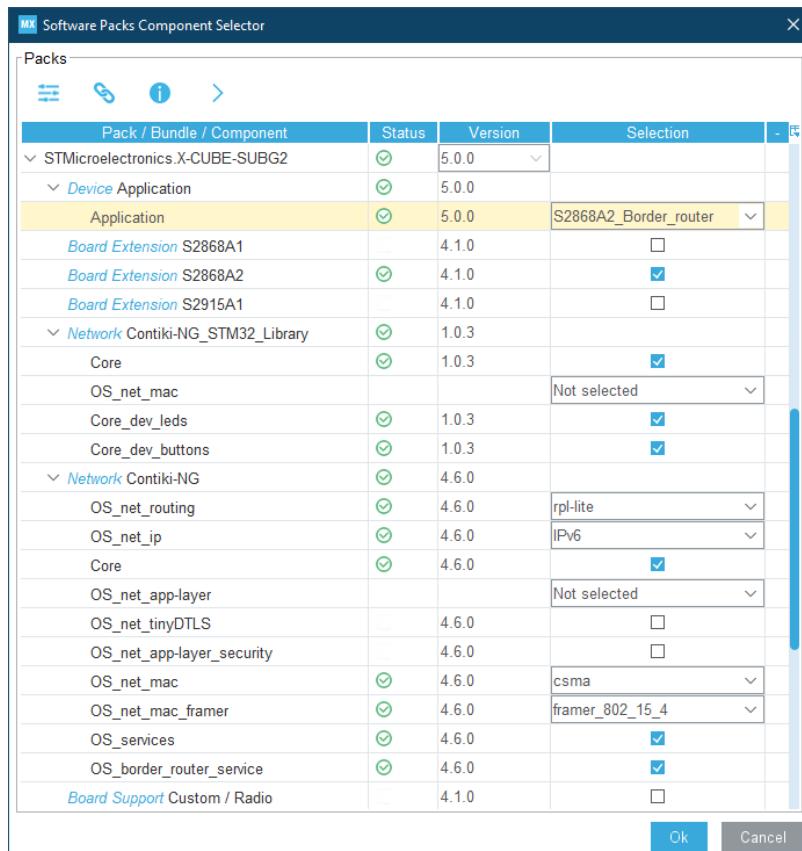
- Contiki-NG / OS\_border\_router\_services

For this application the Button component:

- Contiki-NG\_STM32\_Library / Core\_dev\_buttons

is also required.

The Border Router configuration is shown in **Figure 21**.



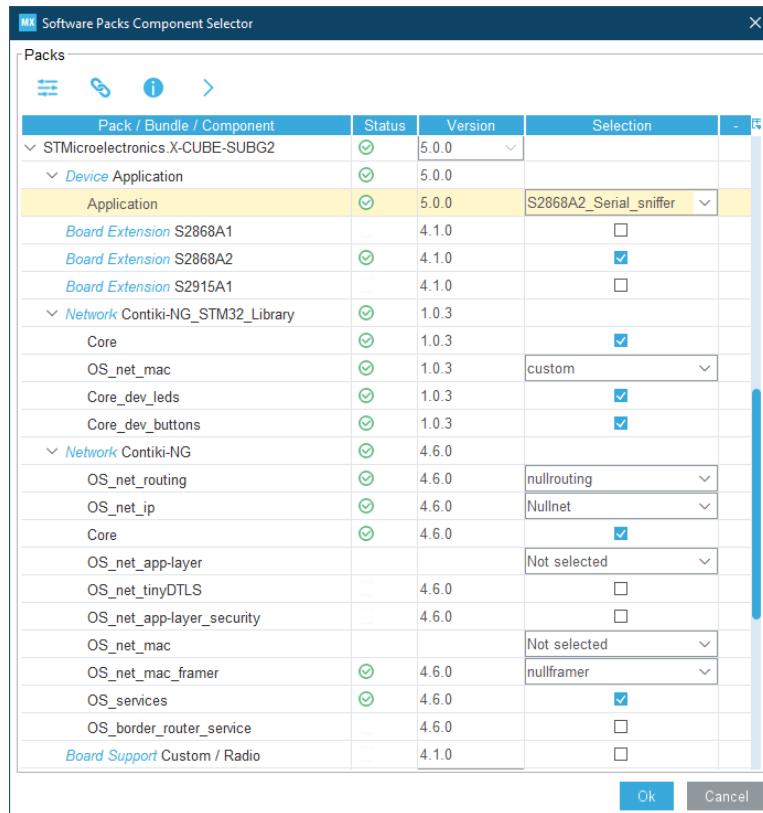
**Figure 21 Contiki-NG configuration for Border Router application**

The Serial Sniffer application differs in term of basic configuration since there is no need of a full protocol stack to process packets that just need to be captured and sent to Wireshark

application.

So, the configuration for the Serial Sniffer application must be as follows (and as summarized in **Figure 22**):

- Contiki-NG / OS\_net\_routing configured as **nullrouting**
- Contiki-NG / OS\_net\_ip configured as **nullnet**
- Contiki-NG / OS\_net\_mac **Not selected**
- Contiki-NG\_STM32\_Library / OS\_net\_mac configured as **custom**
- Contiki-NG / OS\_net\_mac\_framer configured as **nullframer**



**Figure 22** Contiki-NG configuration for Serial Sniffer application

The Contiki-NG\_STM32\_Library / OS\_net\_mac component implements an almost empty MAC layer that simply forward the captured packets.

For this application the Button component:

- Contiki-NG\_STM32\_Library / Core\_dev\_buttons is also required.

To experiment with different settings, not allowed in the provided applications, the user can choose not to select any application, like shown in **Figure 23**. It is important to notify that these configuration options are not fully tested so it might be difficult to get a fully working scenario.

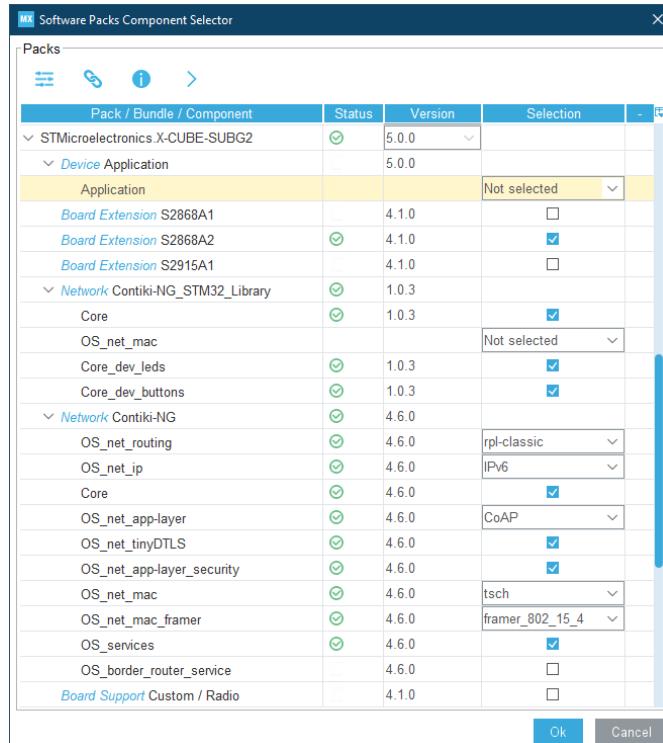


Figure 23 Contiki-NG configuration not linked to any application

For all the Contiki-NG based applications the source code can be generated using the “GENERATE CODE” button like in the P2P Example case.

### 7.3 Command Line Interface application based on X-NUCLEO-S2868A1 or X-NUCLEO-S2868A2 or X-NUCLEO-S2915A1

All the configurations steps described in the previous section 7.1 related to P2P Application apply as the base also for the Command Line Interface (CLI) application, for what concerns the Power Amplifier and the BSP. In this case, no specific components are needed except the BSP and the Application itself.

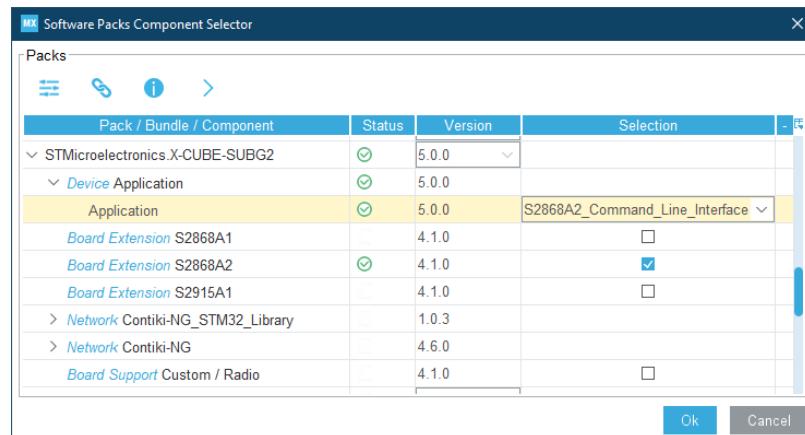


Figure 24 CLI Application components

For the CLI Application, all 4 S2-LP GPIOx (x=0..3) must be configured: in the usual way for

the PA Amplifier in the X-NUCLEO-S2915A1 case, or in Input mode for X-NUCLEO-S2868A1/X-NUCLEO-S2868A2. The PINs to be selected can be retrieved from the tables reported in same section 7.1. GPIO0 corresponds to S2915\_PA\_CSD, GPIO1 corresponds to S2915\_PA\_CPS, GPIO2 corresponds to S2915\_PA\_CTX.

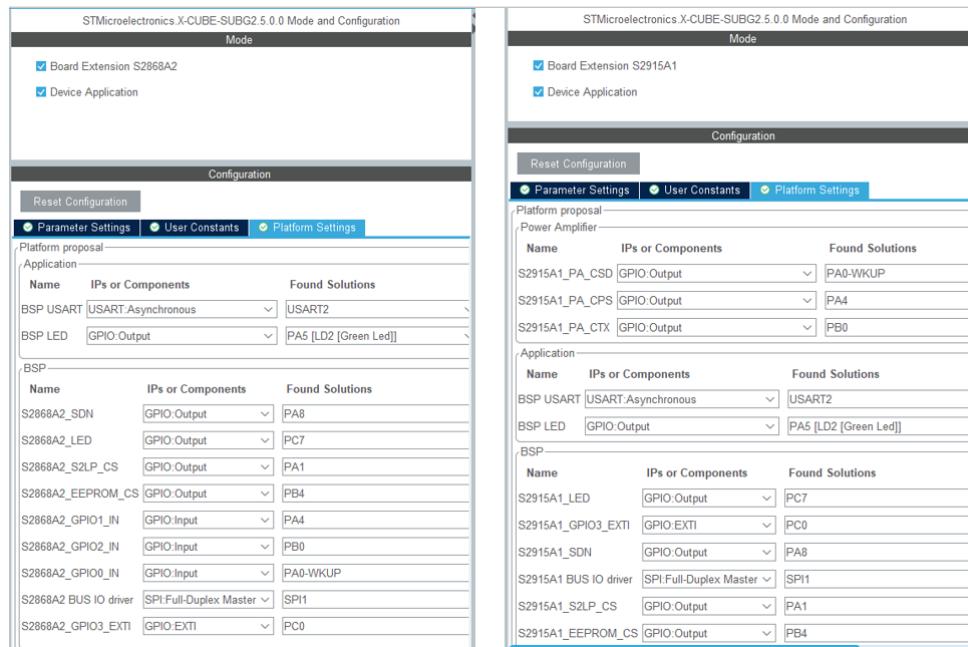


Figure 25 CLI Application Platform Settings (X-NUCLEO-S2868A2 left, X-NUCLEO-S2915A1 right)

The USART for the CLI Application must be configured with DMA enabled for both TX (normal mode) and RX (circular mode), like shown in Figure 26.

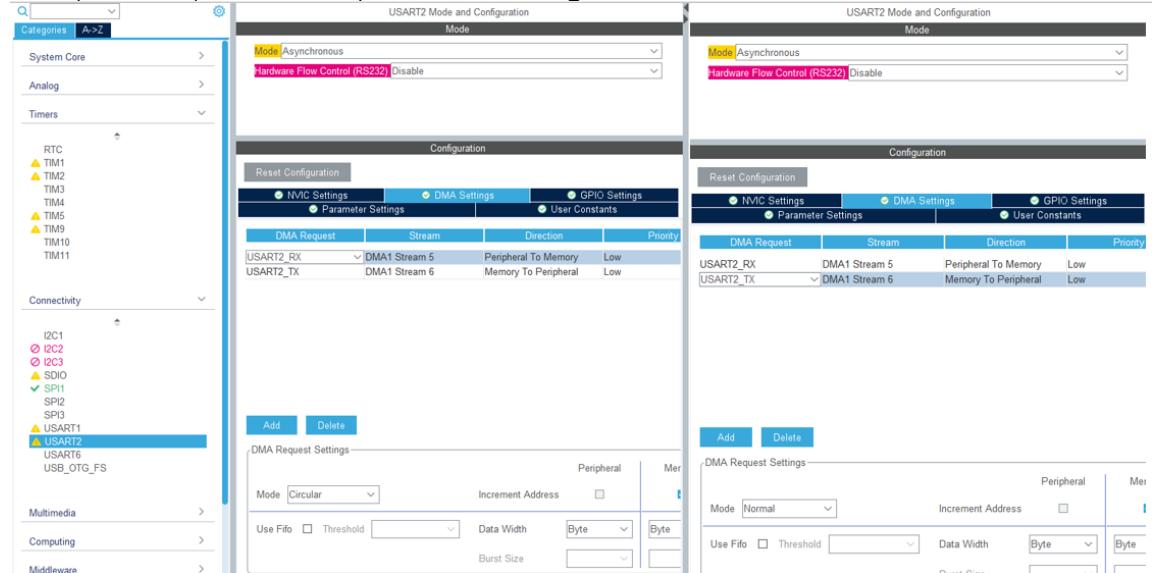


Figure 26 DMA USART configuration for CLI Application, RX left, TX right

When configuring the CLI Application for the U5 family, the DMA for the USART must be enabled using GPDMA1, in this Circular Mode must be disabled for both TX and RX, like shown in Figure 27.

Figure 27 U5 family: GPDMA1 configuration for USART for CLI Application, CH0 (RX) left, CH1 (TX) right

For the multi-packets transmission test that can be done with the CLI Application firmware to properly work, when using different MCUs, it is important to select a Clock source the most precise as possible, because the applicative timer used in that test is implemented using the Systick, a small difference in the Clock oscillator can make the test fail because transmitter and receiver lose synchronization.

When available, it is suggested to use HSE in the PLL Source MUX, as reported in **Figure 28**.

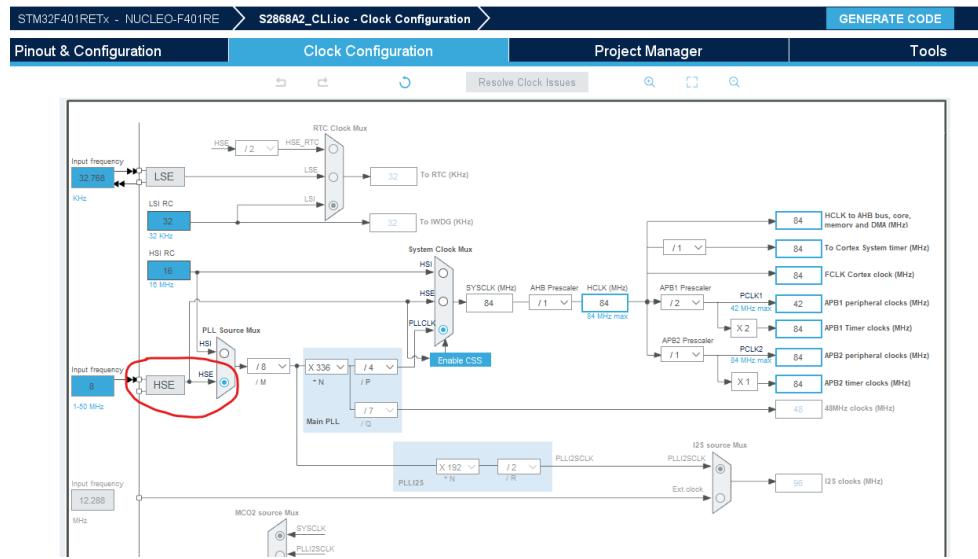


Figure 28 CLI Application clock configuration

It is possible to adjust relative timing also by changing the “Ref timer” setting (by default set to 500ms) in the Transmission Test TX tab of the S2-LP DK GUI.

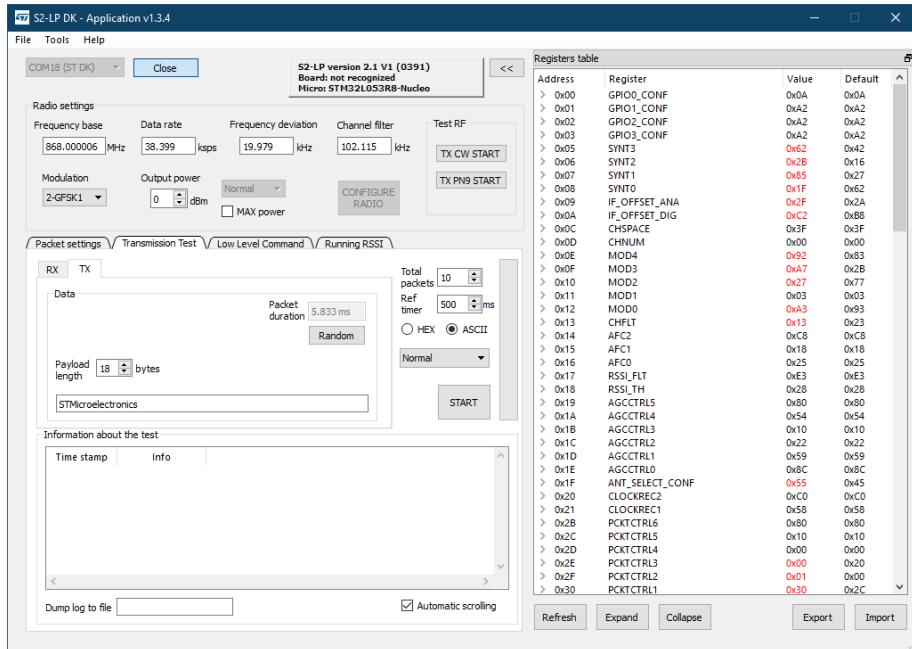


Figure 29 S2-LP DK GUI Transmission Test

If the reference timers are set the same in the different boards, Transmission Test should be completely successful (all packets received, 0 packets discarded) like shown in **Figure 30**.

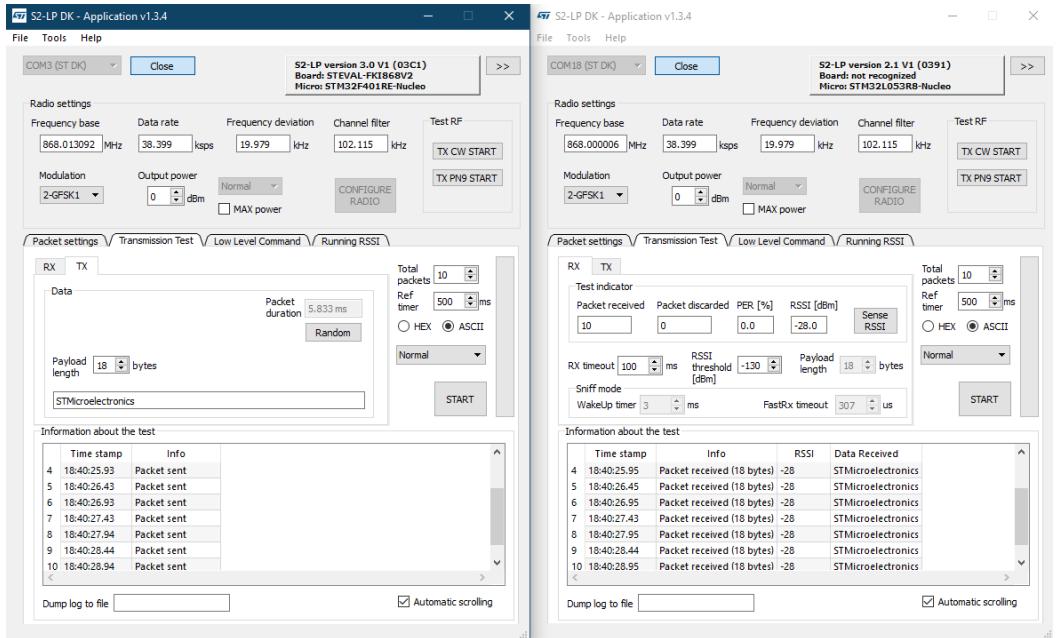
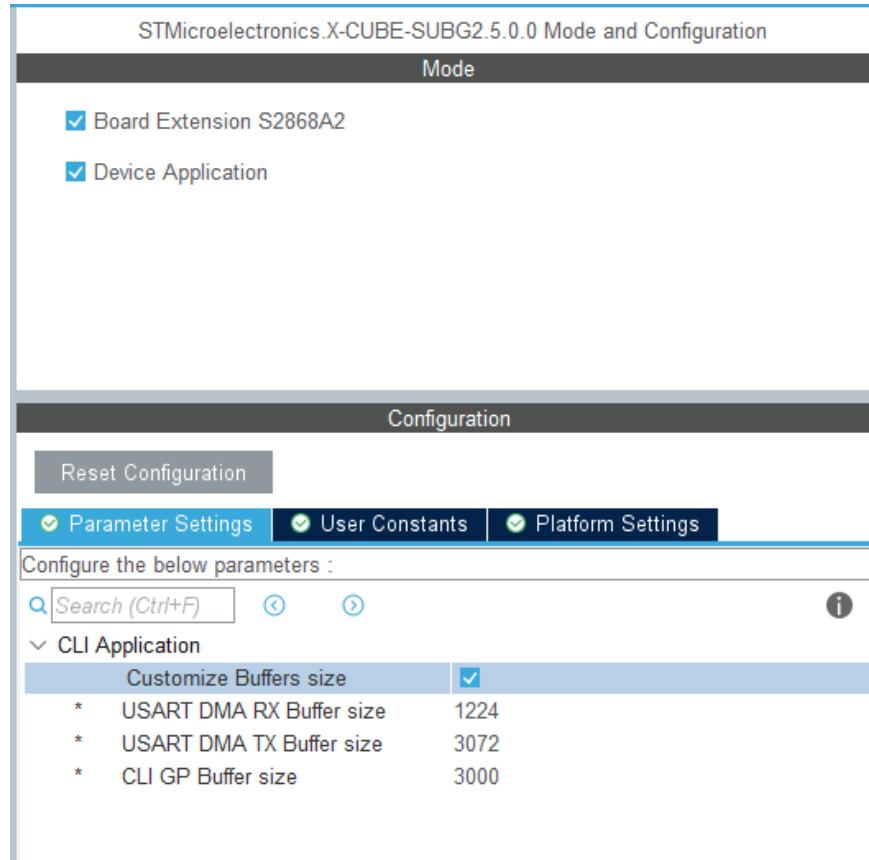


Figure 30 S2-LP DK GUI Transmission Test between NUCLEO-F401RE and NUCLEO-L053R8

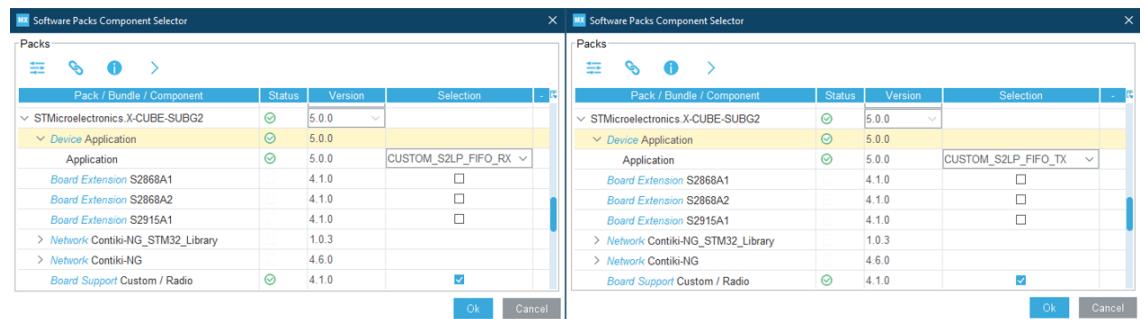
In the Parameter Settings tab it is possible to tweak the size of some applicative buffers. By default the Customize Buffers size checkbox is unchecked and these buffer sizes will be automatically chosen depending on the MCU Family. By checking the above mentioned option, it is possible to change the size of these buffers, in order to reduce them for boards with small memory, or even to increase if possible. This configuration option is illustrated in **Figure 31**.



**Figure 31** CLI Application: Buffers sizes customization

## 7.4 FIFO TX and FIFO RX Applications for Custom Board

Most of the configurations steps described in the previous section 7.1 related to P2P Application apply as the base also for the FIFO TX and FIFO RX application, with the main difference being that no real BSP must be selected in this case, but the “Board Support Custom / Radio”, and the Application itself, of course.

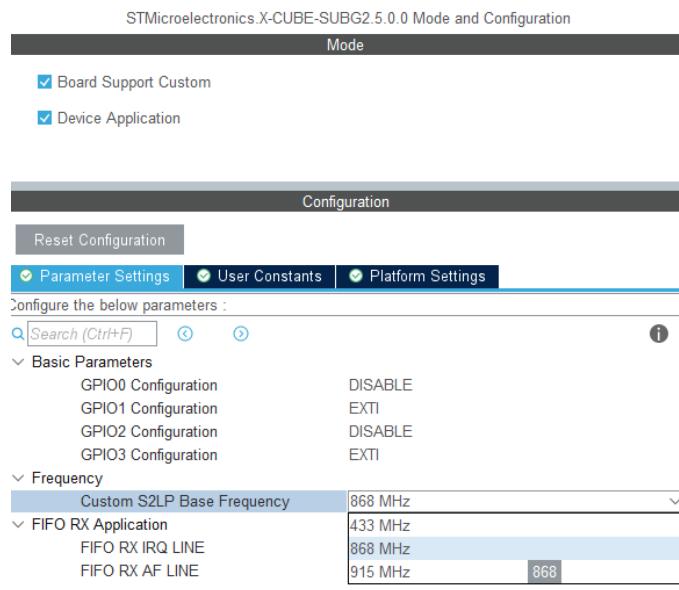


**Figure 32** Component Selector configuration for FIFO RX and FIFO TX Applications or the Custom Board case

The Custom Board support is intended for expansion boards that use the S2-LP transceiver but with wiring different from the standard X-NUCLEO supported in the three classical BSP. For example, with standard X-NUCLEO-S2868A2 only two different GPIOs can really be used in EXTI mode because the four GPIOs are put on two different EXTI lines (0 and 4). Using a custom board, this limitation can be superseded.

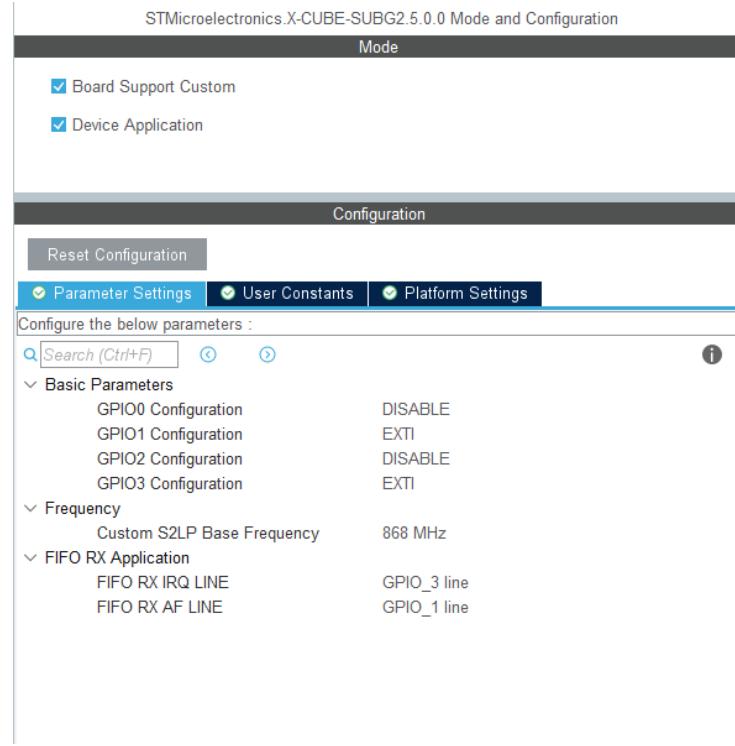
Preconfigured projects and default configuration are compatible with X-NUCLEO-S2868A2 (and X-NUCLEO-S2868A1), anyway.  
 EEPROM and Power Amplifier are not supported in the Custom Board case: user should implement relevant code, in case.

In the Custom Board case, since the radio frequency is not dictated by the preconfigured X-NUCLEO, it can be selected in the Parameter Settings section, like shown in **Figure 33**.



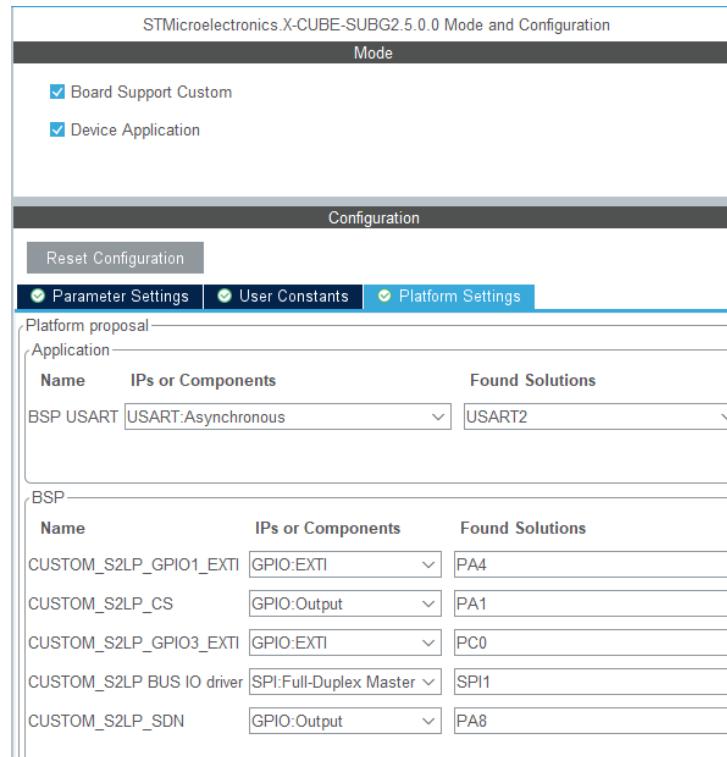
**Figure 33** Custom Board: radio frequency selection

Starting from X-CUBE-SUBG2 5.0.0, if allowed by the selected application, it is possible to configure any of the four S2-LP GPIOs in any possible mode (EXTI, Input, Output), or to disable it. The FIFO RX Application allows to select one line for the FIFO RX Almost Full IRQ, and one line for the standard S2-LP IRQ. This can be done by selecting two GPIOs in EXTI mode and assigning the IRQ Line and AF Line accordingly.



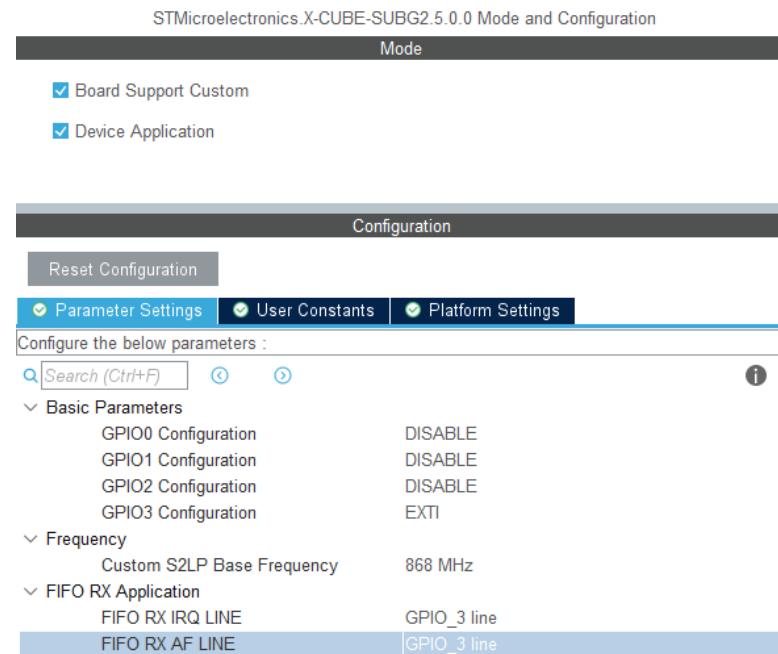
**Figure 34** Default Parameter Settings for FIFO RX Application

Relevant PINs must be assigned to the selected GPIOs in the Platform Settings tab like depicted in **Figure 35**



**Figure 35** FIFO RX Platform Settings

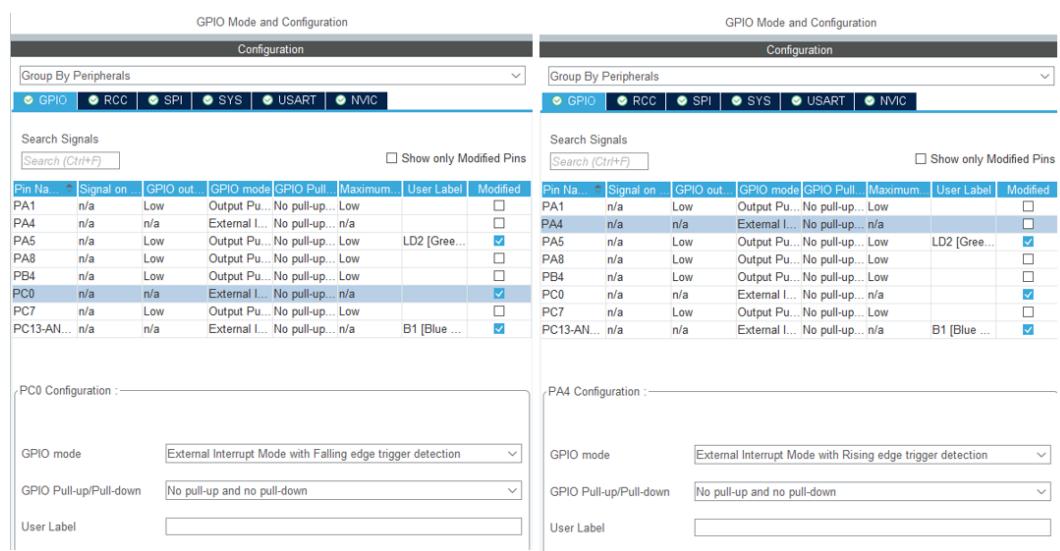
Alternatively, the user can choose to use only one GPIO, in this case he must configure the applicative lines accordingly, like reported in **Figure 36**



**Figure 36** Modified Parameter Settings for FIFO RX Application (one GPIO)

When using two different lines for IRQ Line and AF Line, it is important to:

- 1) Select different Edge selection for the two lines. Like illustrated in section 7.1, the GPIO associated to main IRQ Line must be configured to use “External Interrupt Mode with **Falling** edge trigger detection”, while the GPIO associated to AF Line must be left to default configuration “External Interrupt Mode with **Rising** edge trigger detection”.
- 2) Select a **higher** (i.e. **lower** number) Preemption Priority/Sub Priority for the AF Line compared to IRQ Line, like illustrated in **Figure 38**



**Figure 37** FIFO RX Application: GPIOs Edge detection settings: IRQ Line left, AF Line right

NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
Non maskable interrupt	<input checked="" type="checkbox"/>	0	0
Hard fault interrupt	<input checked="" type="checkbox"/>	0	0
Memory management fault	<input checked="" type="checkbox"/>	0	0
Pre-fetch fault, memory access fault	<input checked="" type="checkbox"/>	0	0
Undefined instruction or illegal state	<input checked="" type="checkbox"/>	0	0
System service call via SWI instruction	<input checked="" type="checkbox"/>	0	0
Debug monitor	<input checked="" type="checkbox"/>	0	0
Pendable request for system service	<input checked="" type="checkbox"/>	0	0
Time base: System tick timer	<input checked="" type="checkbox"/>	0	0
PVD interrupt through EXTI line 16	<input type="checkbox"/>	0	0
Flash global interrupt	<input type="checkbox"/>	0	0
RCC global interrupt	<input type="checkbox"/>	0	0
EXTI line0 interrupt	<input checked="" type="checkbox"/>	0	1
EXTI line4 interrupt	<input checked="" type="checkbox"/>	0	0
SPI1 global interrupt	<input type="checkbox"/>	0	0
USART2 global interrupt	<input type="checkbox"/>	0	0
EXTI line[15:10] interrupts	<input type="checkbox"/>	0	0
FPU global interrupt	<input type="checkbox"/>	0	0

Figure 38 Preemption Priority/Sub Priority for FIFO RX Application: line 4 is used for AF

## 7.5 Advanced configuration

Starting from X-CUBE-SUBG2 5.0.0 all four S2-LP GPIOs can be configured in any mode (EXTI, Input, Output), if a Power Amplifier is not used (X-NUCLEO-S2915A1 case), otherwise GPIO0..2 will be needed for this purpose. This is supported at BSP level (no Application case) and might be supported by the Applications. Currently:

- P2P and Contiki-NG based Applications allow the configuration of any GPIO, but they are not making applicative use of this possibility, user can only change the main S2-LP IRQ GPIO
- CLI Application blocks the possibility to configure GPIOs: GPIO3 must be in EXTI to serve as main S2-LP IRQ, the other GPIOs are forced to be configured in Input mode. S2-LP DK GUI may change these settings at runtime.
- FIFO RX/TX Applications allow the configuration of any GPIO and the FIFO RX Application, like shown in section 7.4, can use two different IRQs associated to two different GPIOs configured in EXTI.

In the Parameter Settings tab it is possible also to choose the main S2-LP GPIO, that obviously must be associated to a GPIO configured in EXTI mode.

Moreover, it is possible to exclude the X-NUCLEO LED from the BSP Platform Settings by disabling the corresponding checkbox in the Parameter Settings tab: this can be useful to build applications that need to use the PIN associated to the LED for other purposes.

Only the enabled GPIOs will be reported in the Platform Settings tab, with the selected mode. As an example, in **Figure 39** we show how the Platform Settings (on the right) will be presented if we disable, in the Parameter Settings tab (on the left), the X-NUCLEO LED and GPIO2, and set GPIO0 as Output, GPIO1 as EXTI, GPIO3 as Input, and we choose GPIO1 to be the main S2-LP EXTI Line.

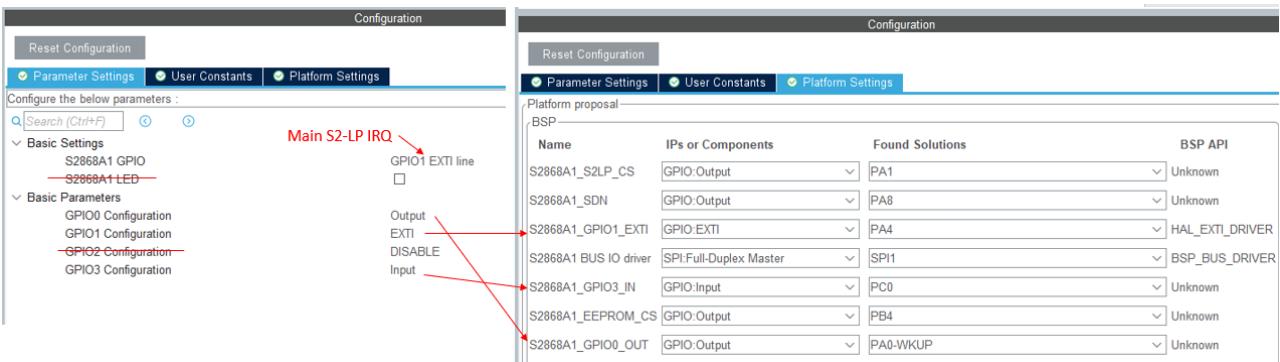


Figure 39 Advanced settings

## 8

## Generated Folders Structure

When generating a project, two models of folders structure can be adopted when using a high level firmware component (i.e. a middleware in the STM32Cube MCU package):

- **Basic Structure:** the basic structure is often used with HAL examples and single package projects. This structure consists of having the IDE configuration folder in the same level as the sources (organized in *Inc* and *Src* subfolders).
- **Advanced Structure:** the advanced structure provides a more efficient and organized folders model that allows ease middleware applications integration when several packages are used.

In the Advanced mode *Src* and *Inc* are generated under folder *Core*.

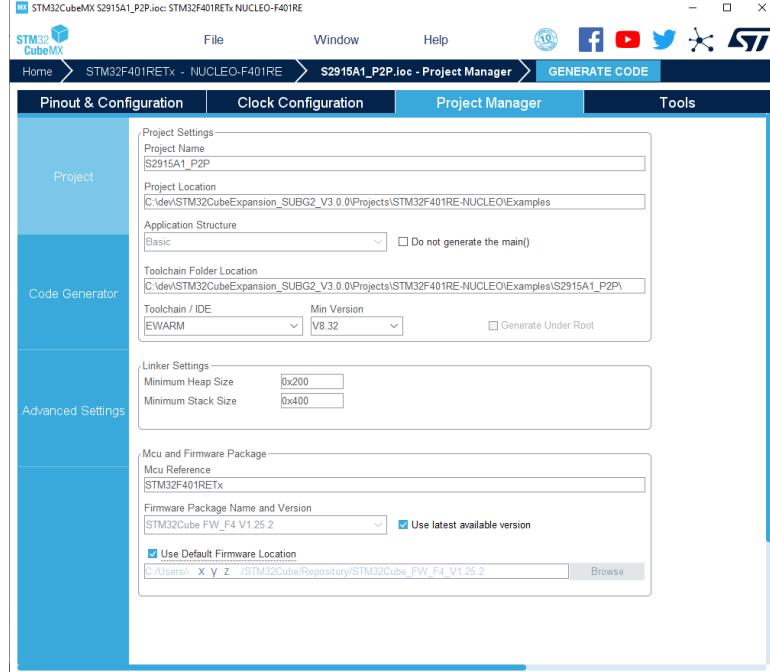
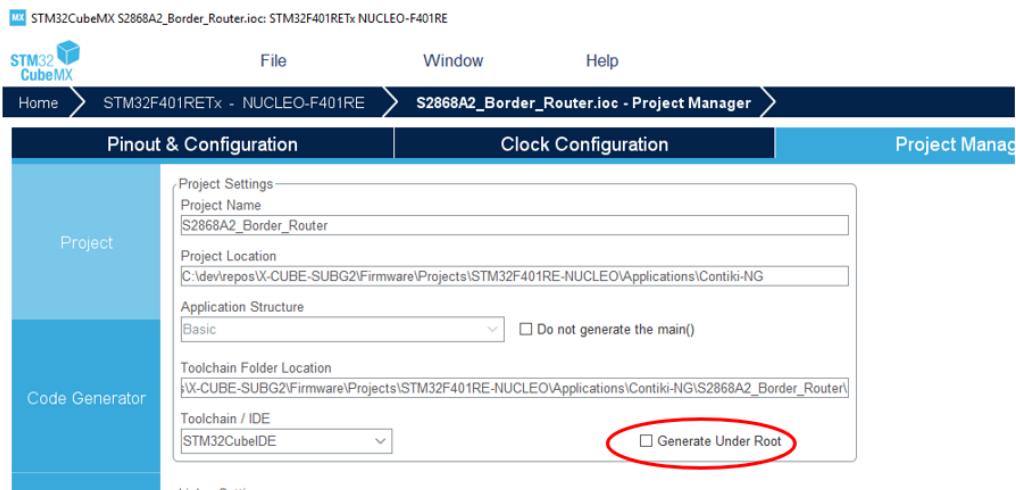


Figure 40 STM32CubeMX Application Structure Configuration

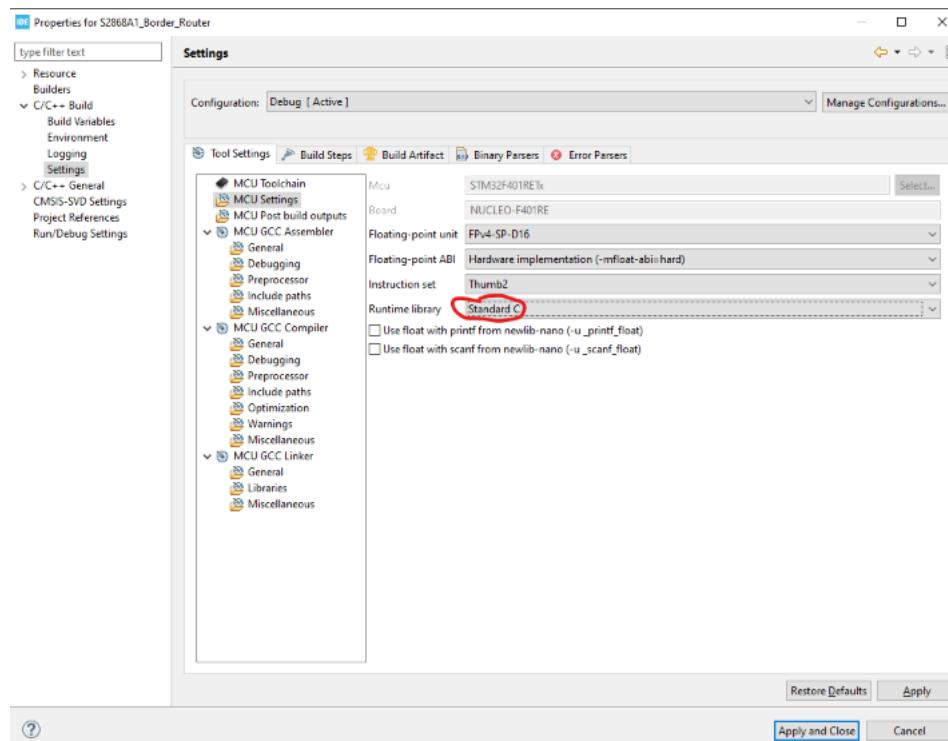
If using STM32CubeIDE please deselect the (by-default enabled) “Generate Under Root” option, like reported in **Figure 41**.



*Figure 41 Generate Under Root option to be disabled*

## 9 Known Limitations and workarounds

- The Contiki-NG based projects for STM32CubeIDE must use “Standard C” library instead of the default “Reduced C”: Properties, C/C++ Build, Settings, MCU Settings, Runtime library => Standard C, like shown in next figure.



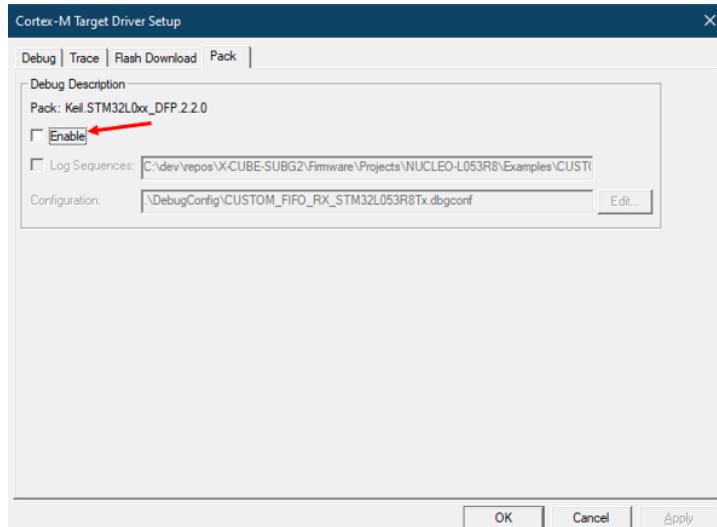
*Figure 42 Library to be enabled for Contiki-NG based projects from STM32CubeIDE 1.6.x*

- If a project has been generated for P2P, to modify it for a Contiki-NG based application that uses LED or Button, it is required to deselect any application and save the .ioc project file before choosing the new Application.
- For platforms with very little FLASH memory it may happen that the generated binary does not fit into it. If using STM32CubeIDE try the Release configuration instead of the

Debug one or enable -Os flag to optimize for size. This issue affects also the preconfigured CLI Application examples for NUCLEO-L053R8: STM32CubeIDE Debug projects have been modified to include -Os option to let them work.

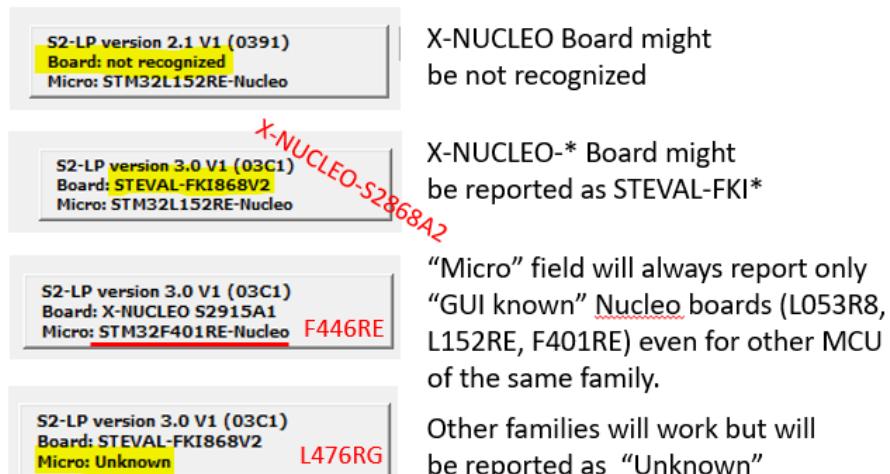
Other possibilities are to disable not required components (via STM32CubeMX) or to change some parameters at compile time (project-conf.h file and Contiki-NG configuration files).

- With ARM KEIL IDE, projects generated by STM32CubeMX for some STM32L0 core may not flash with default configuration: open target configuration, Debug/Pack and un-check “Enable” checkbox, like shown in next figure:



**Figure 43** Pack option to be disabled to flash some L0 based board.

- In this porting for STM32CubeMX, only X-NUCLEO-\* are supported, not STEVAL-FKI\*. NUCLEO or X-NUCLEO information displayed by S2-LP GUI when using CLI Example might not be correct, like detailed in **Figure 44**:



**Figure 44** S2-LP DK GUI: possible not precise information on NUCLEO or X-NUCLEO.

- On some Windows PC the boards used to run the Border Router and Serial Sniffer applications may need to be connected to a Serial Terminal application (like Teraterm) before using the user space utilities, to work correctly.
- If using older IAR System Workbench version (8.32.\*), “rtimer-arch.h” Contiki-NG file

causes an error.

- In MDK-ARM projects, mainly for STM32H7 series, if no log message is printed on the serial terminal, enable the Use MicroLib option in the project settings.
- On dual-core STM32 series this expansion software can be used on both cores but exclusively.
- For the **NUCLEO-F303ZE** Board, the Arduino Pin “A5”, corresponding to the “PD13” MCU pin, should be configured as “GPIO\_EXTI13” for the “S2868A1\_GPIO3”, “S2868A2\_GPIO3” and “S2915A1\_GPIO3” for S2868A1, S2868A2 and S2915A1, respectively. At the same time, the user button, corresponding to “PC13”, should be also configured as “GPIO\_EXTI13”. Given the fact that only one pin can be configured as “GPIO\_EXTI13”, we exclude this board from the list of boards, or you can use different S2868A1 or S2868A2 or S2915A1 GPIO pin for the Interrupt event.
- For the **NUCLEO-F302R8**, **NUCLEO-L412RB-P**, **NUCLEO-L452RE-P** and **NUCLEO-L433RC-P** Board, the user should configure “SPI2” with the following details:
  - a. SPI2\_MOSI ← D11 (PB15)
  - b. SPI2\_MISO ← D12 (PB14)
  - c. SPI2\_SCK ← D13 (PB13)

With this setting, we have the following problem. The “D13”, corresponding to “PB13” MCU pin, is also used by the P2P example as “GPIO\_Output” for the “BSP LED”. Due to the fact that we can configure “PB13” either as “GPIO\_Output” or as “SPI2\_SCK”. So, for LED indication, one can use LED of the X-NUCLEO (S2868A1 or S2868A2 or S2915A1) connected to PC7 by mounting R17 with 0-ohm resistor.

Following will be change in software, to reflect (S2868A1 or S2868A2 or S2915A1) LED instead of BSP LED on NUCLEO-F302R8 or NUCLEO-L412RB-P or NUCLEO-L452RE-P or NUCLEO-L433RC-P.

1. Replace `BSP_LED_Init (LED2)` with `S2868A1_LED_Init`, `S2868A2_LED_Init` and `S2915A1_LED_Init` for `S2868A1_LED`, `S2868A2_LED` and `S2915A1_LED`, respectively, in `MX_SUBG2_P2P_Init`.
  2. Replace `BSP_LED_Toggle (LED2)` with `S2868A1_LED_Toggle`, `S2868A2_LED_Toggle` and `S2915A1_LED_Toggle`, for `S2868A1_LED`, `S2868A2_LED_Toggle` and `S2915A1_LED_Toggle`, respectively.
  3. Replace `BSP_LED_On (LED2)` with `S2868A1_LED_On`, `S2868A2_LED_On` and `S2915A1_LED_On` for `S2868A1_LED`, `S2868A2_LED` and `S2915A1_LED`, respectively.
  4. Replace `BSP_LED_Off (LED2)` with `S2868A1_LED_Off`, `S2868A2_LED_Off` and `S2915A1_LED_Off` for `S2868A1_LED`, `S2868A2_LED` and `S2915A1_LED`, respectively.
- For the **NUCLEO-F103RB** Board, the user should configure “SPI1” with the following details:
    - a. SPI1\_MOSI ← D11 (PA7)
    - b. SPI1\_MISO ← D12 (PA6)
    - c. SPI1\_SCK ← D13 (PA5)

Following will be change in software, to reflect (S2868A1 or S2868A2 or S2915A1) LED instead of BSP LED on NUCLEO-F103RB.

1. Replace `BSP_LED_Init (LED2)` with `S2868A1_LED_Init`, `S2868A2_LED_Init`

and S2915A1\_LED\_Init for S2868A1\_LED, S2868A2\_LED and S2915A1\_LED, respectively, in MX\_SUBG2\_P2P\_Init.

2. Replace BSP\_LED\_Toggle (LED2) with S2868A1\_LED\_Toggle, S2868A2\_LED\_Toggle and S2915A1\_LED\_Toggle, for S2868A1\_LED, S2868A2\_LED\_Toggle and S2915A1\_LED\_Toggle, respectively.
3. Replace BSP\_LED\_On (LED2) with S2868A1\_LED\_On, S2868A2\_LED\_On and S2915A1\_LED\_On for S2868A1\_LED, S2868A2\_LED and S2915A1\_LED, respectively.
4. Replace BSP\_LED\_Off (LED2) with S2868A1\_LED\_Off, S2868A2\_LED\_Off and S2915A1\_LED\_Off for S2868A1\_LED, S2868A2\_LED and S2915A1\_LED, respectively.

## 10 References

- [1] [UM2669](#) – User Manual – *Getting started with X-CUBE-SUBG2, Sub-1 GHz RF software expansion for STM32Cube*
- [2] [Contiki-NG GitHub repository](#)
- [3] [Evaluation SW package based on S2-LP](#)

# 11 Revision history

**Table 6** Document revision history

Date	Version	Changes
21-Nov-2019	1	Initial release
04-Feb-2020	2	Add S2868A2 and S2915A1 support Improvement in the table of configuration of pins
11-Nov-2020	3	Added Contiki-NG middleware and related applications
11-Feb-2021	4	Updated info for v4.0.0
21-Apr-2021	5	Updated info for v4.1.0
14-June-2021	6	Updated info for v4.2.0
20-April-2022	7	Updated info for v5.0.0

---

## IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to [www.st.com/trademarks](http://www.st.com/trademarks). All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2020 STMicroelectronics – All rights reserved