Getting started with the STMicroelectronics X-CUBE-BLE2 software package for STM32CubeMX

Introduction

This document provides the guidelines to configure and use the X-CUBE-BLE2 software package V3.3.0 for STM32CubeMX (minimum required version V6.6.0). The document contains the description of the provided sample applications, the description of the steps required to configure a generic project using the BLE middleware as well as of the steps to configure and use the sample application provided in the package.

Information and documentation related to the ST BlueNRG-2 network processor, the X-NUCLEO-BNRG2A1 expansion board and the X-CUBE-BLE2 expansion software for Bluetooth Low Energy are available on www.st.com.

Contents

List of figures

1 Acronyms and abbreviations

Table 1: list of acronyms

Acronym	Description
BLE	Bluetooth Low Energy
FOTA	Firmware Over The Air
HAL	Hardware Abstraction Layer
HID	Human Interface Device
IOT	Internet Of Things
IP	Internet Protocol
LAN	Local Area Network
NVIC	Nested Vectored Interrupt Controller
PCB	Printed Circuit Board
RTC	Real Time Clock
RTOS	Real Time Operating System
SPI	Serial Peripheral Interface
UID	Unique Identifier
URL	Uniform Resource Locator
U(S)ART	Universal (Synchronous) Asynchronous Receiver Transmitter
USB	Universal Serial BUS
TCP	Transmission Control Protocol

2 What is STM32Cube?

STM32Cube is a combination of a full set of PC software tools and embedded software blocks running on STM32 microcontrollers and microprocessors:

- STM32CubeMX configuration tool for any STM32 device; it generates initialization C code for Cortex-M cores and the Linux device tree source for Cortex-A cores
- STM32CubeIDE integrated development environment based on open-source solutions like Eclipse or the GNU C/C++ toolchain, including compilation reporting features and advanced debug features
- STM32CubeProgrammer programming tool that provides an easy-to-use and efficient environment for reading, writing and verifying devices and external memories via a wide variety of available communication media (JTAG, SWD, UART, USB DFU, I2C, SPI, CAN, etc.)
- STM32CubeMonitor family of tools (STM32CubeMonRF, STM32CubeMonUCPD, STM32CubeMonPwr) to help developers customize their applications in real-time
- STM32Cube MCU and MPU packages specific to each STM32 series with drivers (HAL, low-layer, etc.), middleware, and lots of example code used in a wide variety of real-world use cases
- STM32Cube expansion packages for application-oriented solutions

3 License

The software provided in this package is licensed under Software License Agreement SLA0055.

4 Sample Applications Description

In this section a short overview of the sample applications included in the X-CUBE-BLE2 pack is provided.
The sample applications:

- are ready-to-use projects that can be generated through the STM32CubeMX for any board equipped with an STM32 MCU and using the BlueNRG-2 chip;
- show the users how to use the BLE APIs, provided by the BlueNRG-2 middleware, for correctly initialize and use a BLE device.

4.1 SensorDemo_BLESensor-App

This application is an example showing how to implement an application tailored for interacting with the "ST BLE Sensor" app for Android/iOS devices.
The "ST BLE Sensor" app is freely available on both Play Store and iTunes.
The source code of the "ST BLE Sensor" app is also available on GitHub for both iOS and Android devices.
After establishing the connection between the STM32 board and the smartphone:

- the temperature and the pressure emulated values are sent by the STM32 board to the mobile device and are shown in the ENVIRONMENTAL tab;
- the emulated sensor fusion data sent by the STM32 board to the mobile device reflects into the cube rotation showed in the app's MEMS SENSOR FUSION tab
- the plot of the emulated data (temperature, pressure, sensor fusion data, accelerometer, gyroscope and magnetometer) sent by the board are shown in the PLOT DATA tab;
- in the RSSI & Battery tab the RSSI value is shown.

According to the value of the `#define USE_BUTTON` in file `app_bluenrg_2.c`, the environmental and the motion data can be sent automatically (with 1 sec period) or when the User Button is pressed.
Projects generated for the STM32L476RG MCU contain the code for the FOTA (Firmware-Over-The-Air) support. In such a case there are some modifications to the code and the project settings that must be manually introduced. The required information is contained hereafter in this document and in the readme.txt file generated by the STM32CubeMX along with the application code.

4.2 SampleApp

This sample application shows how to simply use the BLE Stack.
It provides the user with a complete example on how to perform an ATT MTU exchange procedure, in order the server and the client can agree on the supported max MTU.
To test this application two STM32 Nucleo boards with their respective X-NUCLEO-BNRG2A1 STM32 expansion boards should be used. After flashing both the STM32 board, one board configures itself as BLE Server-Peripheral device, while the other as BLE Client-Central device.
After the connection between the two boards is established (signaled by the LD2 LED blinking on the Client-Central device), pressing the USER button on one board, the LD2 LED on the other one gets toggled and vice versa.
If you have only one STM32 Nucleo board, you can the “BLE Scanner” app (available for both the Android and the iOS devices) as BLE Client-Central device.

4.3 Beacon

This example application shows how to use the BlueNRG-2 STM32 expansion board to implement a Beacon device.

A Beacon device is a smart Bluetooth Low Energy device that transmits a small data payload at regular intervals using Bluetooth advertising packets.
Beacons are used to mark important places and objects. Typically, a beacon is visible to a user's device from a range of a few meters, allowing for highly context-sensitive use cases.
To locate the beacon, it is necessary to have a scanner application running on a BLE-capable smartphone, such as “BLE Scanner” app (available for both the Android and the iOS devices).

4.4 Central

This sample application shows how to implement a BLE Central device.
The BLE Central device can scan the BLE network, connect to a BLE device (the peripheral), discover its services and characteristics and update the characteristic properties enabling the data notifications, the reading and the writing. The user can communicate with the BLE central device through a serial console, setting the Speed to 115200 baud/sec, where both the command menu and the peripheral information are printed.
This application requires the following linker settings:

- CSTACK minimum size 0x600
- HEAP minimum size 0x200

4.5 Virtual_COM_Port

Virtual_COM_Port is the application to be used with the BlueNRG GUI SW package (STSW-BNRGUI).
The BlueNRG GUI is a PC application that can be used to interact and evaluate the capabilities of the BlueNRG-2 device both in master and slave role. The BlueNRG GUI allows standard and vendor specific HCI commands to be sent to the device controller and events to be received from.

4.6 IFRStack_Updater

The IFRStack_Updater sample application can be used to update the BlueNRG-2 firmware stack to the latest version or to change the IFR configuration.
To change the IFR configuration, the `#define BLUENRG_DEV_CONFIG_UPDATER` must be defined (see file `app_bluenrg_2.h`).
The IFR parameters can be changed editing the file `Middlewares\ST\BlueNRG-2\hci\bluenrg1_devConfig.c`.

WARNING *Be sure you know what you are doing when modifying the IFR settings. This operation should be performed only by expert users.*

To update the BlueNRG-2 firmware stack, the `#define BLUENRG_STACK_UPDATER` must be defined (see file `app_bluenrg_2.h`).
The firmware image used by the application is contained in the `FW_IMAGE` array defined in file `update_fw_image.c`. In the same file, the information on the DTM SPI and the FW versions is reported.
For STM32 MCUs with low flash size, the DTM image can be updated using a serial terminal (e.g. HyperTerminal, TeraTerm, …) and its transfer feature based on the YMODEM.
In this case the `#define BLUENRG_STACK_UPDATER_YMODEM` must be defined (see file `app_bluenrg_2.h`).
If, for instance, the TeraTerm is used as serial terminal application on the PC, to enable the YMODEM feature the following steps must be executed:

- open a connection
- go to `File --> Transfer --> YMODEM --> Send`
- select the DTM SPI binary file to load from the PC's file system (e.g. the `DTM_SPI_NOUPDATER.bin` contained in the installation folder of the ST's BlueNRG GUI, `C:\Users\<username>\ST\BlueNRG GUI X.Y.Z\Firmware\BlueNRG2\DTM`)

This application requires the following linker settings:

- CSTACK minimum size 0xD00
- HEAP minimum size 0x200

5 Installing the X-CUBE-BLE2 pack in STM32CubeMX

After downloading (from www.st.com), installing and launching the STM32CubeMX (V≥6.0.0), the X-CUBE-BLE2 pack can be installed in few steps.

1. From the top menu bar, select Help → Manage embedded software packages

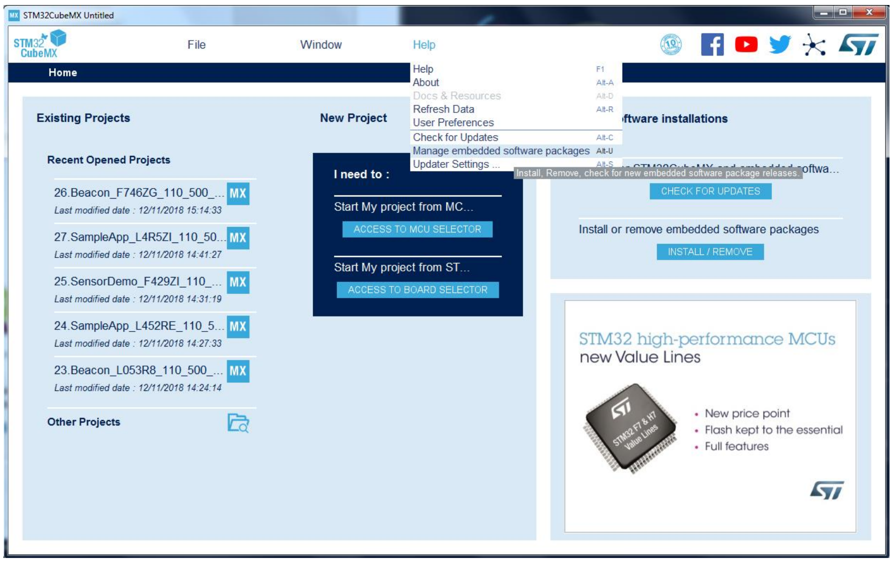
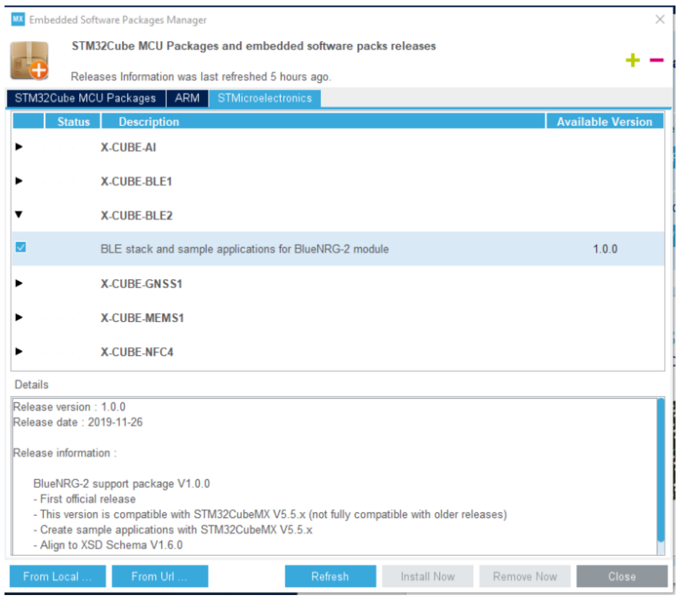

STM32CubeMX Untitled
File
Window
Help
Home
Help F1
About Alt-A
Docs & Resources Alt-D
Refresh Data Alt-R
User Preferences
Check for Updates Alt-C
Manage embedded software packages Alt-U
Updater Settings ... Alt-S
Install, Remove, check for new embedded software package releases
Existing Projects
New Project
ftware installations
Recent Opened Projects
26.Beacon_F746ZG_110_500_...
Last modified date : 12/11/2018 15:14:33
27.SampleApp_L4R5ZI_110_50...
Last modified date : 12/11/2018 14:41:27
25.SensorDemo_F429ZI_110_...
Last modified date : 12/11/2018 14:31:19
24.SampleApp_L452RE_110_5...
Last modified date : 12/11/2018 14:27:33
23.Beacon_L053R8_110_500_...
Last modified date : 12/11/2018 14:24:14
Other Projects
I need to :
Start My project from MC...
ACCESS TO MCU SELECTOR
Start My project from ST...
ACCESS TO BOARD SELECTOR
CHECK FOR UPDATES
Install or remove embedded software packages
INSTALL / REMOVE
STM32 high-performance MCUs
new Value Lines
New price point
Flash kept to the essential
Full features

Figure 1 Managing embedded software packs in STM32CubeMX

2. 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-BLE2 pack.



STM32Cube MCU Packages and embedded software packs releases

Releases Information was last refreshed 5 hours ago.

	Status	Description	Available Version
▶		X-CUBE-AI	
▶		X-CUBE-BLE1	
▼		X-CUBE-BLE2	
✓		BLE stack and sample applications for BlueNRG-2 module	1.0.0
▶		X-CUBE-GNSS1	
▶		X-CUBE-MEMS1	
▶		X-CUBE-NFC4	

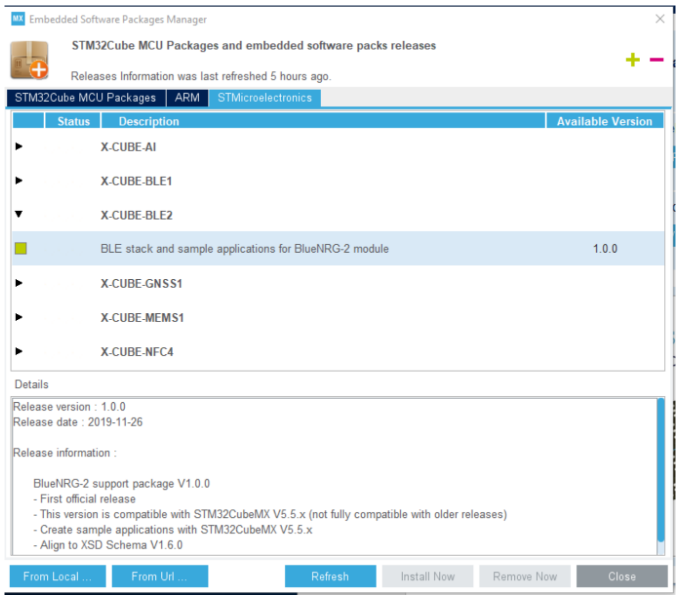
Release version : 1.0.0
Release date : 2019-11-26

Release information :

BlueNRG-2 support package V1.0.0
- First official release
- This version is compatible with STM32CubeMX V5.5.x (not fully compatible with older releases)
- Create sample applications with STM32CubeMX V5.5.x
- Align to XSD Schema V1.6.0

Figure 2 *Installing the X-CUBE-BLE2 pack in STM32CubeMX*

3. Select the X-CUBE-BLE2 pack checking the corresponding box and install it pressing the Install Now button. After accepting the license terms and 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.



STM32Cube MCU Packages and embedded software packs releases

Releases Information was last refreshed 5 hours ago.

	Status	Description	Available Version
▶		X-CUBE-AI	
▶		X-CUBE-BLE1	
▼		X-CUBE-BLE2	
🟩		BLE stack and sample applications for BlueNRG-2 module	1.0.0
▶		X-CUBE-GNSS1	
▶		X-CUBE-MEMS1	
▶		X-CUBE-NFC4	

Release version : 1.0.0
Release date : 2019-11-26

Release information :

BlueNRG-2 support package V1.0.0
- First official release
- This version is compatible with STM32CubeMX V5.5.x (not fully compatible with older releases)
- Create sample applications with STM32CubeMX V5.5.x
- Align to XSD Schema V1.6.0

Figure 3 *The X-CUBE-BLE2 pack in STM32CubeMX*

6 Starting a new project

After launching the STM32CubeMX, click either the ACCESS TO MCU SELECTOR or the ACCESS TO BOARD SELECTOR button in the GUI to start a project for your MCU or board.

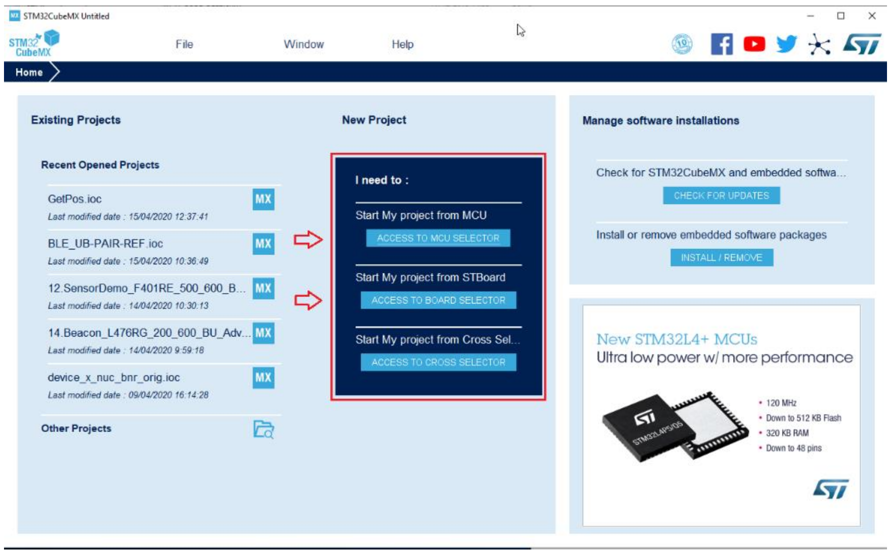

STM32CubeMX Untitled
File
Window
Help
Home
Existing Projects
Recent Opened Projects
GetPos.ioc
Last modified date : 15/04/2020 12:37:41
BLE_UB-PAIR-REF.ioc
Last modified date : 15/04/2020 10:36:49
12.SensorDemo_F401RE_500_600_B...
Last modified date : 14/04/2020 10:30:13
14.Beacon_L476RG_200_600_BU_Adv...
Last modified date : 14/04/2020 9:59:18
device_x_nuc_bnr_orig.ioc
Last modified date : 09/04/2020 16:14:28
Other Projects
New Project
I need to :
Start My project from MCU
ACCESS TO MCU SELECTOR
Start My project from STBoard
ACCESS TO BOARD SELECTOR
Start My project from Cross Sel...
ACCESS TO CROSS SELECTOR
Manage software installations
Check for STM32CubeMX and embedded softwa...
CHECK FOR UPDATES
Install or remove embedded software packages
INSTALL / REMOVE
New STM32L4+ MCUs
Ultra low power w/ more performance
120 MHz
Down to 512 KB Flash
320 KB RAM
Down to 48 pins

Figure 4 *STM32CubeMX main page*

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

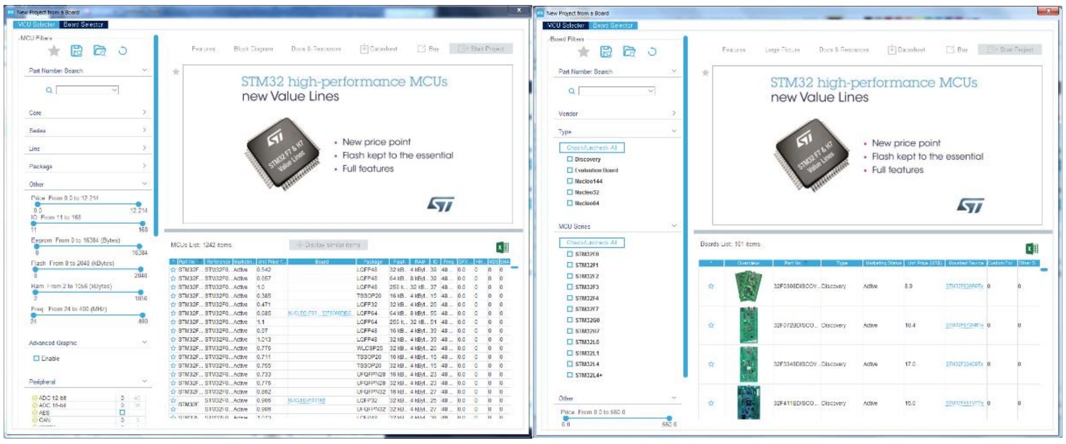


Figure 5 *STM32CubeMX* ***MCU/Board Selector*** *windows*

After selecting the MCU or the Board, the selected STM32 pinout will appear (the user can either choose to Initialize all peripherals with their default Mode or not). From this window the user can set up the project, by adding one or more Software Packs and peripherals and configuring the clock.

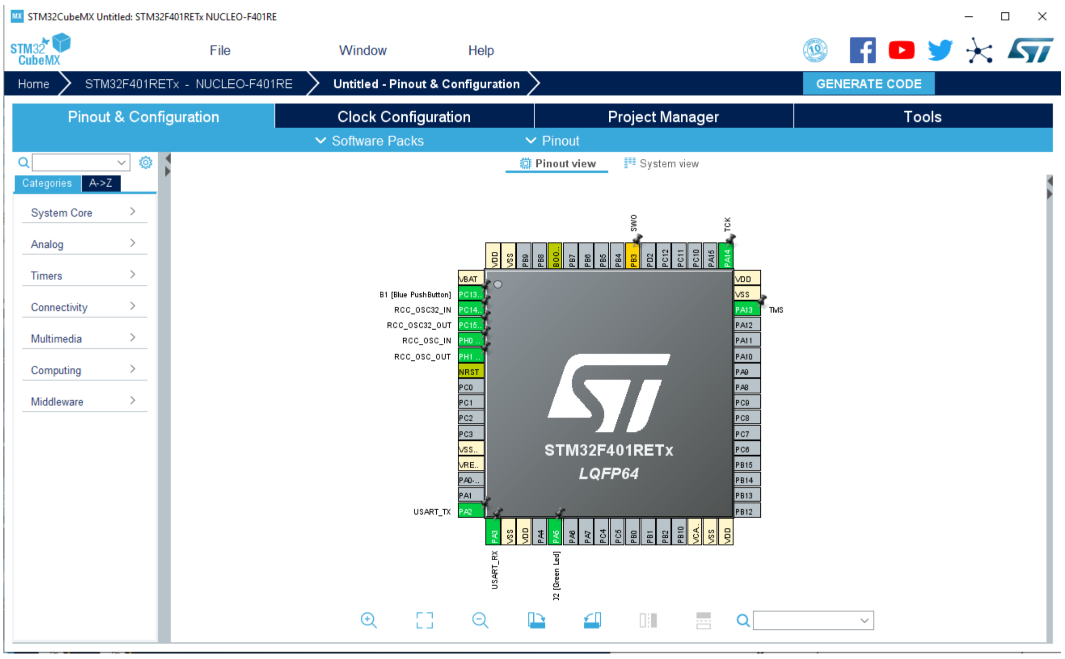
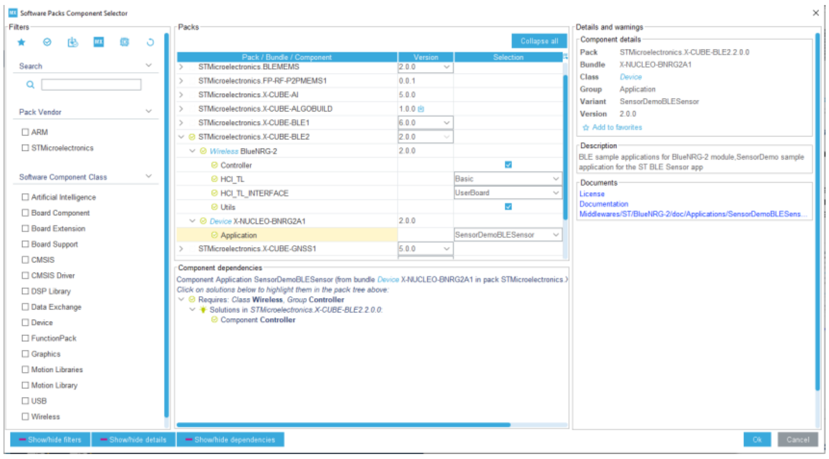


Figure 6 STM32CubeMX **Configuration** window

To add the X-CUBE-BLE2 additional software to the project, the Software Packs → Select Components menu item must be selected.
From the Software Packs 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.


Software Packs Component Selector

Filters

Search

Pack Vendor
- ☐ ARM
- ☐ STMicroelectronics

Software Component Class
- ☐ Artificial Intelligence
- ☐ Board Component
- ☐ Board Extension
- ☐ Board Support
- ☐ CMSIS
- ☐ CMSIS Driver
- ☐ DSP Library
- ☐ Data Exchange
- ☐ Device
- ☐ FunctionPack
- ☐ Graphics
- ☐ Motion Libraries
- ☐ Motion Library
- ☐ USB
- ☐ Wireless

Packs

Pack / Bundle / Component	Version	Selection
STMicroelectronics.BLEMEMS	2.0.0	
STMicroelectronics.FP-RF-P2PMEMS1	0.0.1	
STMicroelectronics.X-CUBE-AI	5.0.0	
STMicroelectronics.X-CUBE-ALGOBUILD	1.0.0	
STMicroelectronics.X-CUBE-BLE1	6.0.0	
STMicroelectronics.X-CUBE-BLE2	2.0.0	
Wireless BlueNRG-2	2.0.0	
Controller		✓
HCI_TL		Basic
HCI_TL_INTERFACE		UserBoard
Utils		✓
Device X-NUCLEO-BNRG2A1	2.0.0	
Application		SensorDemoBLESensor
STMicroelectronics.X-CUBE-GNSS1	5.0.0	

Component dependencies
Component Application SensorDemoBLESensor (from bundle *Device* X-NUCLEO-BNRG2A1 in pack STMicroelectronics.)
Click on solutions below to highlight them in the pack tree above:
Requires: *Class* **Wireless**, *Group* **Controller**
Solutions in *STMicroelectronics.X-CUBE-BLE2.2.0.0:*
Component **Controller**

Details and warnings

Component details

Pack	STMicroelectronics.X-CUBE-BLE2.2.0.0
Bundle	X-NUCLEO-BNRG2A1
Class	*Device*
Group	Application
Variant	SensorDemoBLESensor
Version	2.0.0

Add to favorites

Description
BLE sample applications for BlueNRG-2 module,SensorDemo sample application for the ST BLE Sensor app

Documents
License
Documentation
Middlewares/ST/BlueNRG-2/doc/Applications/SensorDemoBLESens...

Figure 7 STM32CubeMX ***Software Packs Component Selector*** *window*

7 HCI_TL and HCI_TL_INTERFACE Configuration

The HCI_TL (Host Controller Interface Transport Layer) and the HCI_TL_INTERFACE (Host Controller Interface Transport Layer Interface) are the interfaces between the HAL/BSP layer and both the Middleware Core Layer and the Application Layer.

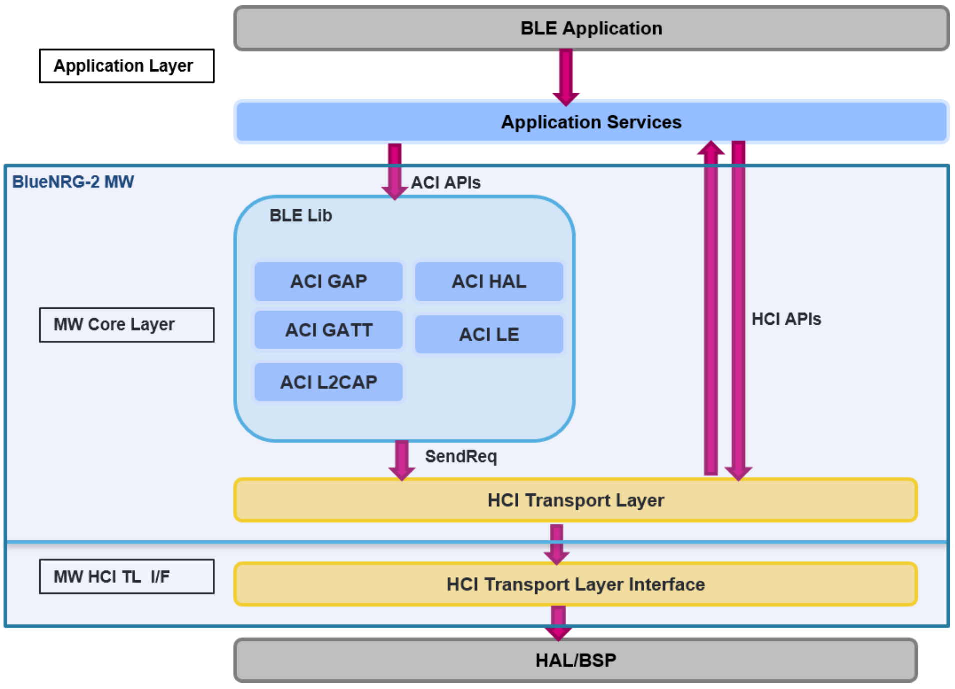


Figure 8 *BlueNRG-2 software block scheme*

Two different configurations may be used for both the components.

- HCI_TL
  - *Basic* the user can use a basic set of already implemented APIs
  - *Template* the user can implement his own APIs for building his own customized HCI TL
- HCI_TL_INTERFACE
  - *UserBoard* the user can use a basic set of already implemented APIs
  - *Template* the user can implement his own APIs for building his own customized HCI TL Interface.

For generating a ready to work sample application, the *Basic* and *UserBoard* configurations must be selected.

8 STM32 Configuration Steps

The X-NUCLEO-BNRG2A1 interfaces with the STM32 microcontroller via the SPI pins. Hence, assuming a user wants to interface the ST X-NUCLEO-BNRG2A1 expansion board with a STM32 Nucleo 64 pins board (e.g. a Nucleo-F401RETx) or a STM32 Nucleo 144 pins board (e.g. a Nucleo F429ZITx), the following steps must be executed in STM32CubeMX before generating a project.

NOTE

- *To correctly set the RESET on pin D7 a 0 Ohm resistor must be soldiered on R117. Alternatively, the D7 pin of the Arduino connector and the pin #5 of the J12 on the X-NUCLEO-BNRG2A1 expansion board must be bridged.*
- *On some STM32 Nucleo 64/144 pins, it could be required to bridge the D13 and D3 pins of the Arduino connector to correctly set the SPI SCK.*

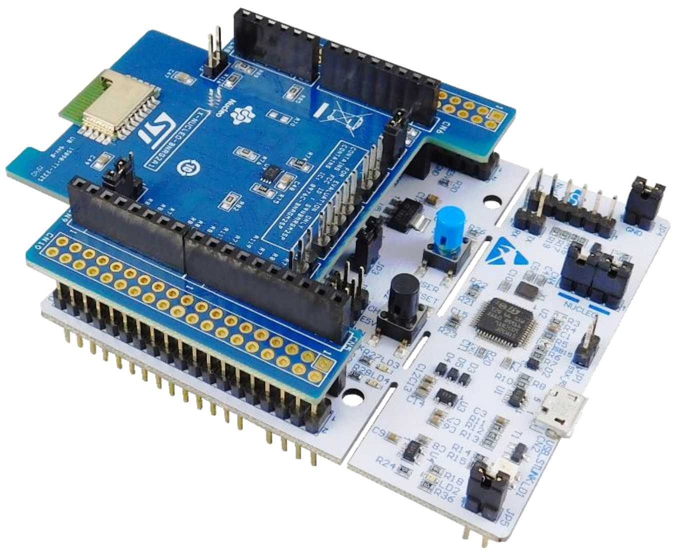


Figure 9 *STM32 Nucleo 64 pins and X-NUCLEO-BNRG2A1*

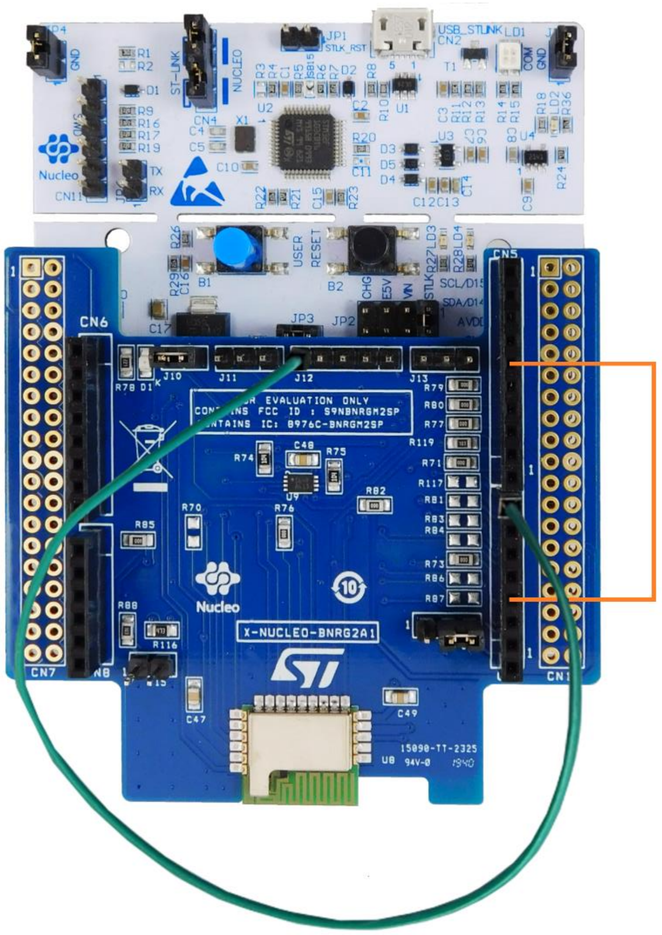


Figure 10 *STM32 Nucleo 64 pins and X-NUCLEO-BNRG2A1 (with bridge for RESET and the indication for the SPI SCK bridge)*

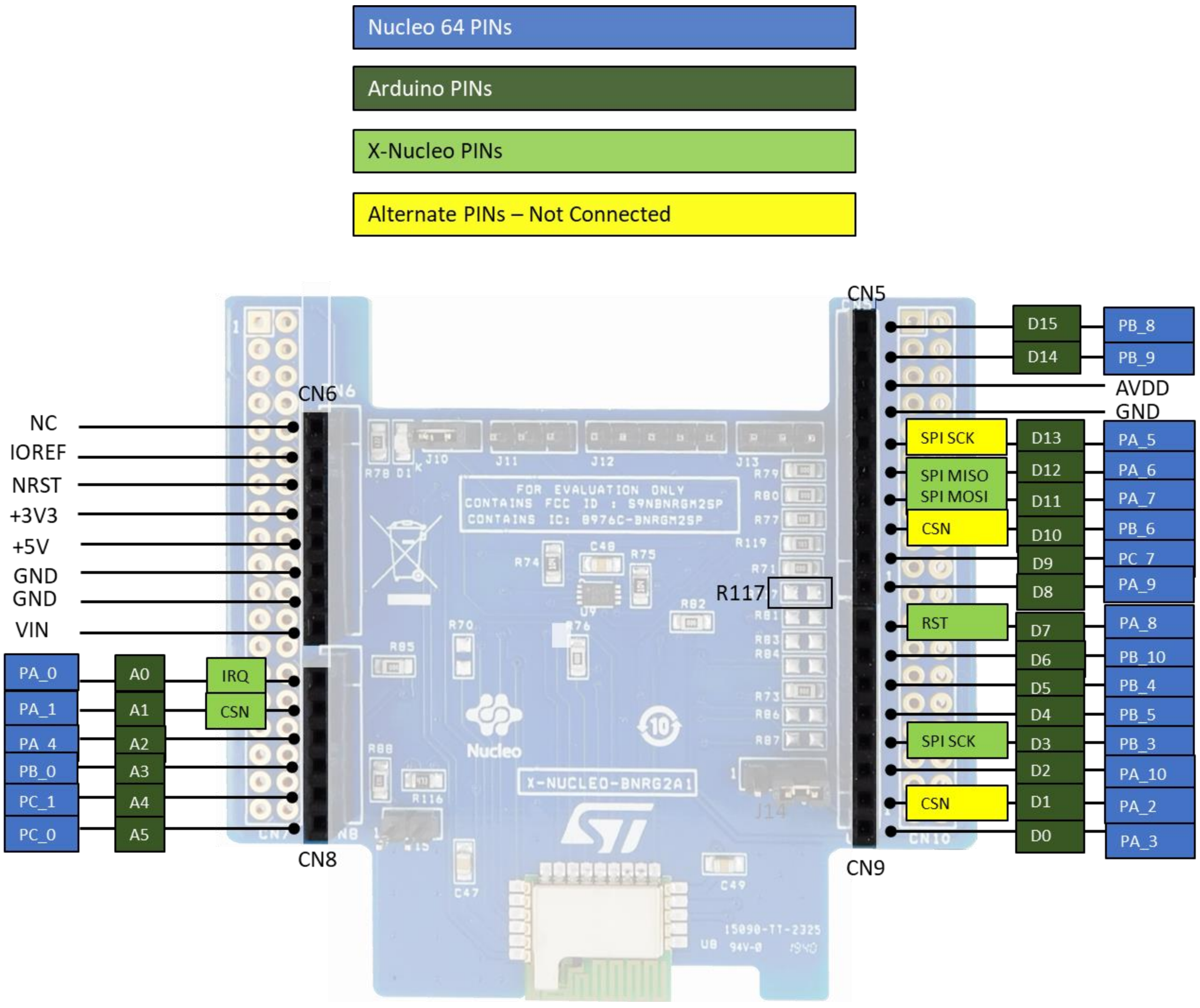


***Figure 11** X-NUCLEO-BNRG2A1 pinout*

8.1 Use of Expansion Software without sample applications

This section outlines how to configure STM32CubeMX when the use of the sample applications is not required. With such setup, only middleware and driver layers will be configured. This setup is useful when the user does not intend to leverage the sample applications provided in the package.

From the Pinout & Configuration tab:

- from the Pinout view, if PB3 pin is already assigned, click on it and reset its state;
- from the ˅ Pinout → Clear Pinouts menu option reset the state of all pins (only for Nucleo 144);
- from the Connectivity > menu:
  - check that the ETH is disabled (only for Nucleo 144)
  - enable the SPI1 in Full-Duplex Master Mode

  Check that the enabled SPI uses the following pins:

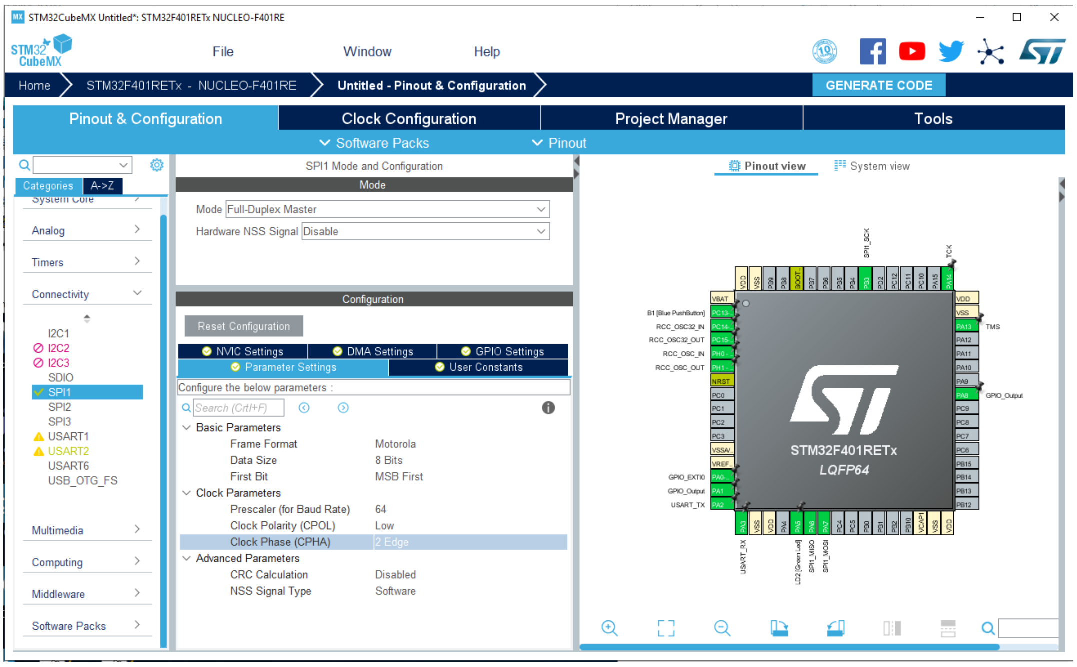
	Arduino PINs
SPI_MISO	D12

SPI_MOSI	D11
SPI_SCK	D3

In case different pins have been enabled by default, change them using the Alternative pins (if available) or bridge the enabled ones and the ones reported in the table above.

From the Pinout view set:

Nucleo 64		Nucleo 144	
PA0	GPIO_EXTI0	PA3	GPIO_EXTI3
PA1	GPIO_Output	PC0	GPIO_Output
PA8	GPIO_Output	PF13	GPIO_Output



SPI1 Mode and Configuration

Mode

Mode: Full-Duplex Master

Hardware NSS Signal: Disable

Configuration

Reset Configuration

NVIC Settings | DMA Settings | GPIO Settings | Parameter Settings | User Constants

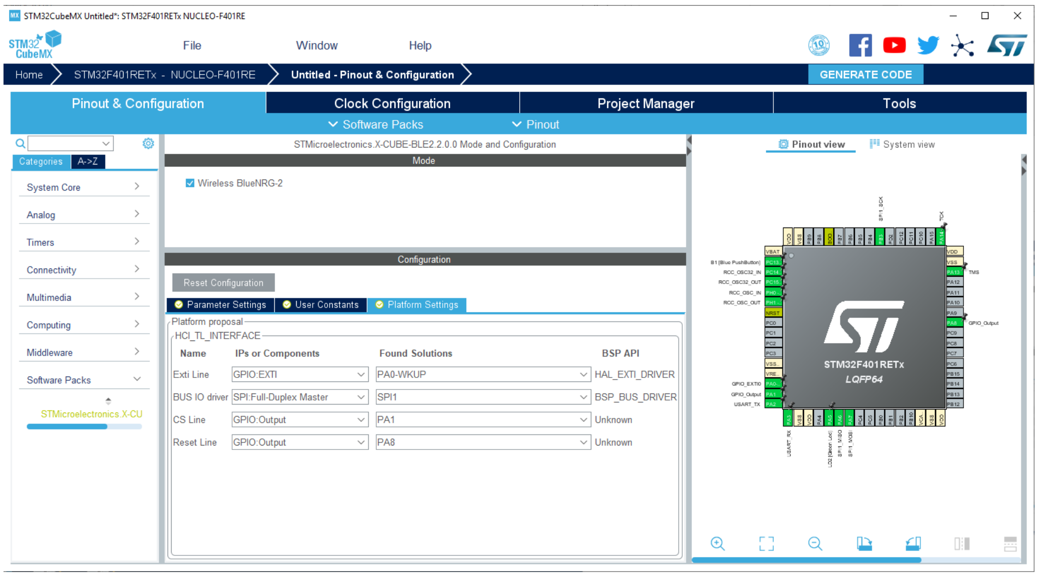
Configure the below parameters :

Basic Parameters	
Frame Format	Motorola
Data Size	8 Bits
First Bit	MSB First
Clock Parameters	
Prescaler (for Baud Rate)	64
Clock Polarity (CPOL)	Low
Clock Phase (CPHA)	2 Edge
Advanced Parameters	
CRC Calculation	Disabled
NSS Signal Type	Software

Figure 12 Pinout view

From the Software Packs category item, click on the STMicroelectronics.X-CUBE-BLE2.x.y.z pack. Check the Wireless BlueNRG-2 box and set the following Platform Settings:

Name	IPs or Components	Found solutions		BSP Api
		Nucleo 64	Nucleo 144	
Exti Line	GPIO:EXTI	PA0	PA3	HAL_EXTI_DRIVER
BUS IO driver	SPI:Full-Duplex Master	SPI1	SPI1	BSP_BUS_DRIVER
CS Line	GPIO:Output	PA1	PC0	Unknown
Reset Line	GPIO:Output	PA8	PF13	Unknown



STMicroelectronics.X-CUBE-BLE2.2.0.0 Mode and Configuration

Mode

✓ Wireless BlueNRG-2

Configuration

Platform proposal

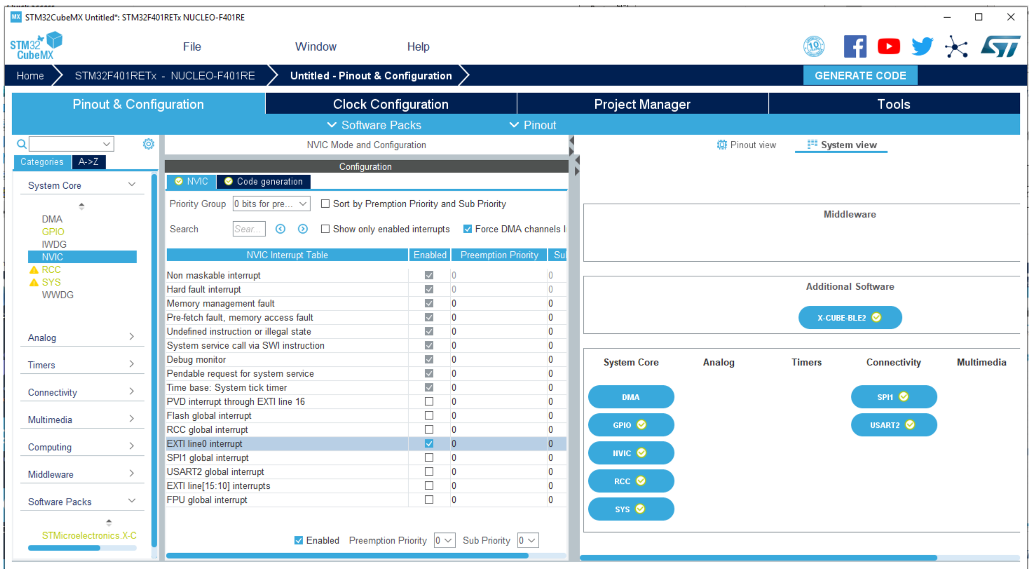
HCI_TL_INTERFACE

Name	IPs or Components	Found Solutions	BSP API
Exti Line	GPIO:EXTI	PA0-WKUP	HAL_EXTI_DRIVER
BUS IO driver	SPI:Full-Duplex Master	SPI1	BSP_BUS_DRIVER
CS Line	GPIO:Output	PA1	Unknown
Reset Line	GPIO:Output	PA8	Unknown

Figure 13 STMicroelectronics.X-CUBE-BLE2 ***Mode and Configuration*** view

From the System view, click on NVIC button under System Core category to enable the EXTI line interrupt:

Nucleo 64	Nucleo 144
EXTI line 0 interrupt	EXTI line 3 interrupt



NVIC Mode and Configuration

Configuration

Priority Group: 0 bits for pre...

☐ Sort by Premption Priority and Sub Priority

☐ Show only enabled interrupts

✓ Force DMA channels I

NVIC Interrupt Table	Enabled	Preemption Priority	Su
Non maskable interrupt	✓	0	0
Hard fault interrupt	✓	0	0
Memory management fault	✓	0	0
Pre-fetch fault, memory access fault	✓	0	0
Undefined instruction or illegal state	✓	0	0
System service call via SWI instruction	✓	0	0
Debug monitor	✓	0	0
Pendable request for system service	✓	0	0
Time base: System tick timer	✓	0	0
PVD interrupt through EXTI line 16	☐	0	0
Flash global interrupt	☐	0	0
RCC global interrupt	☐	0	0
EXTI line0 interrupt	✓	0	0
SPI1 global interrupt	☐	0	0
USART2 global interrupt	☐	0	0
EXTI line[15:10] interrupts	☐	0	0
FPU global interrupt	☐	0	0

✓ Enabled Preemption Priority 0 Sub Priority 0

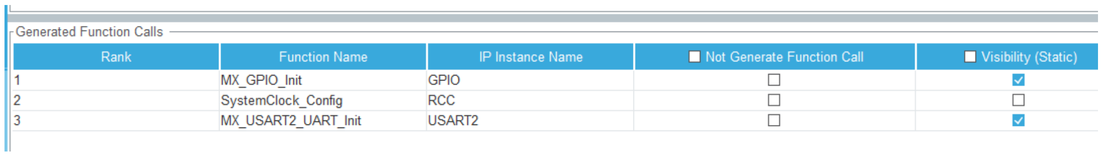
Figure 14 NVIC ***Mode and Configuration*** view

From the System view, click on SPIx button under Connectivity category and:

- check that the Data size is 8 Bits;
- set the Prescaler (for Baud Rate) to a value so that the HClock/Prescaler is less or equal to 1000.0 Kbits/s (the maximum supported SPI speed);
- set the Clock Phase (CPHA) to 2 Edge

Once all the above described steps have been performed, from the Project Manager tab the Project Name, the Project Location, the Toolchain/IDE, the Firmware Package Name and Version and so on can be set.

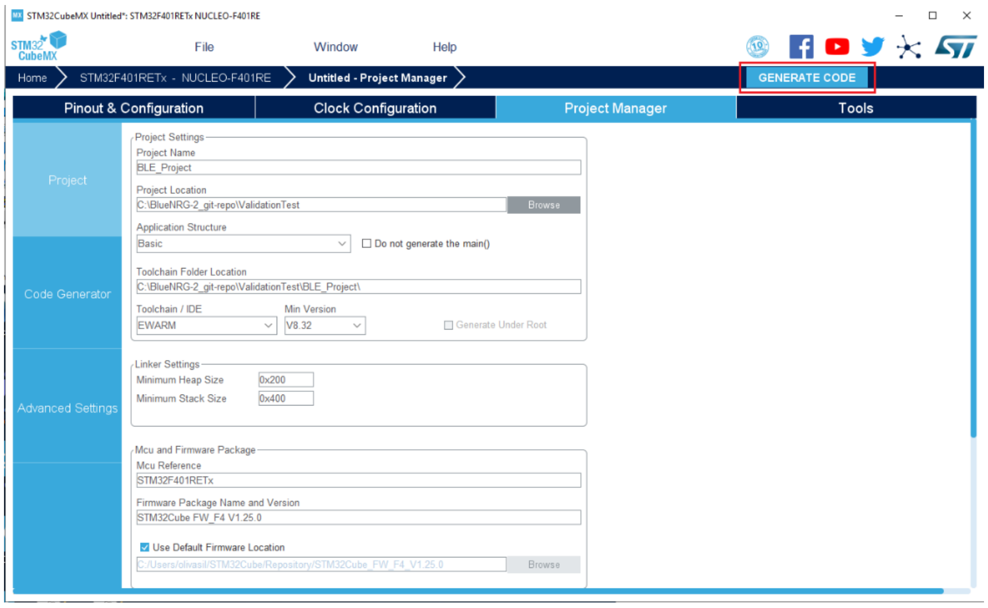
In the Advanced Settings tab, the following options are set by default. You can use or change them according to your needs.


Generated Function Calls

Rank	Function Name	IP Instance Name	☐ Not Generate Function Call	☐ Visibility (Static)
1	MX_GPIO_Init	GPIO	☐	✓
2	SystemClock_Config	RCC	☐	☐
3	MX_USART2_UART_Init	USART2	☐	✓

Figure 15 *Advanced Settings*

the source code of the project using the **STMicroelectronics X-CUBE-BLE2** software can be generated clicking the GENERATE CODE button.



Project Settings
Project Name: BLE_Project
Project Location: C:\BlueNRG-2_git-repo\ValidationTest
Application Structure: Basic ☐ Do not generate the main()
Toolchain Folder Location: C:\BlueNRG-2_git-repo\ValidationTest\BLE_Project\
Toolchain / IDE: EWARM Min Version: V8.32 ☐ Generate Under Root

Linker Settings
Minimum Heap Size: 0x200
Minimum Stack Size: 0x400

Mcu and Firmware Package
Mcu Reference: STM32F401RETx
Firmware Package Name and Version: STM32Cube FW_F4 V1.25.0
✓ Use Default Firmware Location
C:/Users/olivasil/STM32Cube/Repository/STM32Cube_FW_F4_V1.25.0

Figure 16 *Project Manager view*

8.2 Use of Expansion Software with sample applications

This section outlines how to configure STM32CubeMX when the use of the sample applications is required. With such setup, all the components of the expansion software package, including applications, will be properly configured.

From the Pinout & Configuration tab:

- from the Pinout view, if PB3 pin is already assigned, click on it and reset its state;
- from the ∨ Pinout → Clear Pinouts menu option reset the state of all pins (only for Nucleo 144);
- from the Connectivity > menu:
    - check that the ETH is disabled (only for Nucleo 144);
    - enable the SPI1 in Full-Duplex Master Mode;

- if not enabled yet, enable the USART2 in Asynchronous mode (for Nucleo 64)
- if not enabled yet, enable the USART3 in Asynchronous mode (for Nucleo 144)

Check that the enabled SPI uses the following pins:

	Arduino PINs
SPI_MISO	D12
SPI_MOSI	D11
SPI_SCK	D3

In case different pins have been enabled by default, change them using the Alternative pins (if available) or bridge the enabled ones and the ones reported in the table above. Also, the UART enabled above is used for the logs on a terminal via serial link. The correct USART to be enabled may change for some Nucleo board. Information on the correct USART to use are reported in the board User Manual available on st.com.

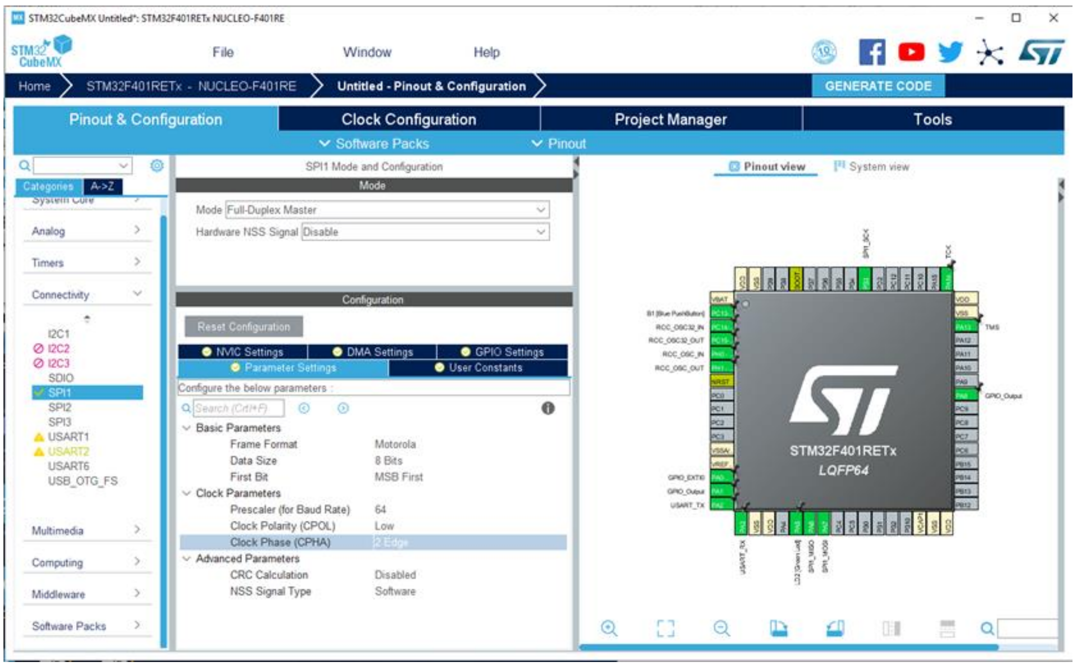
NOTE: Only for SensorDemo_BLESensor-App for STM32L476RG MCU:
from the Computing > menu enable the CRC checking the Activated checkbox (this module is required by the FOTA feature - available only for the STM32L476RG MCU -).

From the Pinout view set:

Nucleo 64			**Nucleo 144**		
PIN	Mode	Label	PIN	Mode	Label
PA0	GPIO_EXTI0		PA3	GPIO_EXTI3	
PA1	GPIO_Output		PC0	GPIO_Output	
PA8	GPIO_Output		PF13	GPIO_Output	
PA2	USART2_TX	USART_TX	PD8	USART3_TX	USART_TX
PA3	USART2_RX	USART_RX	PD9	USART3_RX	USART_RX
PA5⋆	GPIO_Output	LD2 [Green Led]	PB7	GPIO_Output	LD2[Blue]
PC13	GPIO_EXTI13	B1 [Blue PushButton]	PC13	GPIO_EXTI13	USER_Btn[B1]

Pins in the green rows are used only by the SensorDemo_BLESensor-App, SampleApp and Central sample applications.

⁕ NOTE: In the User Manual UM1724 “STM32 Nucleo-64 boards (MB1136)” it is reported that *the green LED is a user LED connected to Arduino signal D13 corresponding to STM32 I/O PA5 (pin 21) or PB13 (pin 34) depending on the STM32 target*. That means the for some Nucleo-64 board (for instance the Nucleo-F302R8) the LD2[Green Led] may be connected to the PB13 pin instead of to the PA5 pin.



SPI1 Mode and Configuration

Mode

Mode Full-Duplex Master

Hardware NSS Signal Disable

Configuration

Reset Configuration

NVIC Settings | DMA Settings | GPIO Settings | Parameter Settings | User Constants

Configure the below parameters :

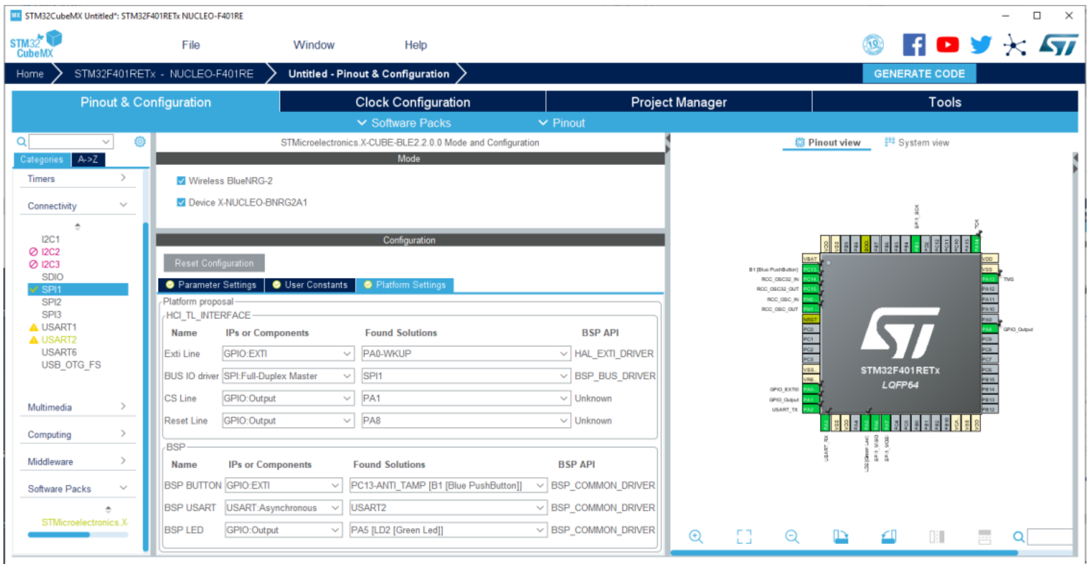
```
Basic Parameters
    Frame Format                Motorola
    Data Size                   8 Bits
    First Bit                   MSB First
Clock Parameters
    Prescaler (for Baud Rate)   64
    Clock Polarity (CPOL)       Low
    Clock Phase (CPHA)          2 Edge
Advanced Parameters
    CRC Calculation             Disabled
    NSS Signal Type             Software
```

***Figure 17 Pinout** view*

From the Additional Software > menu, click on the STMicroelectronics.X-CUBE-BLE2.x.y.z pack. Check the Wireless BlueNRG-2 box and set the following Platform Settings:

Name	Supported IPs	Found solutions		BSP Api
		Nucleo 64	Nucleo 144	
BUS IO driver	SPI:Full-Duplex Master	SPI1	SPI1	BSP_BUS_DRIVER
Exti Line	GPIO:EXTI	PA0	PA3	HAL_EXTI_DRIVER
CS Line	GPIO:Output	PA1	PC0	Unknown
Reset Line	GPIO:Output	PA8	PF13	Unknown
BSP LED	GPIO:Output	PA5	PB7	BSP_COMMON_DRIVER
BSP BUTTON	GPIO:EXTI	PC13	PC13	BSP_COMMON_DRIVER
BSP USART	USART:Asynchronous	USART2	USART3	BSP_COMMON_DRIVER

Pins in the green rows are used only by the SensorDemo_BLESensor-App, SampleApp and Central sample applications.



☑ Wireless BlueNRG-2

☑ Device X-NUCLEO-BNRG2A1

Platform proposal

HCI_TL_INTERFACE

Name	IPs or Components	Found Solutions	BSP API
Exti Line	GPIO:EXTI	PA0-WKUP	HAL_EXTI_DRIVER
BUS IO driver	SPI:Full-Duplex Master	SPI1	BSP_BUS_DRIVER
CS Line	GPIO:Output	PA1	Unknown
Reset Line	GPIO:Output	PA8	Unknown

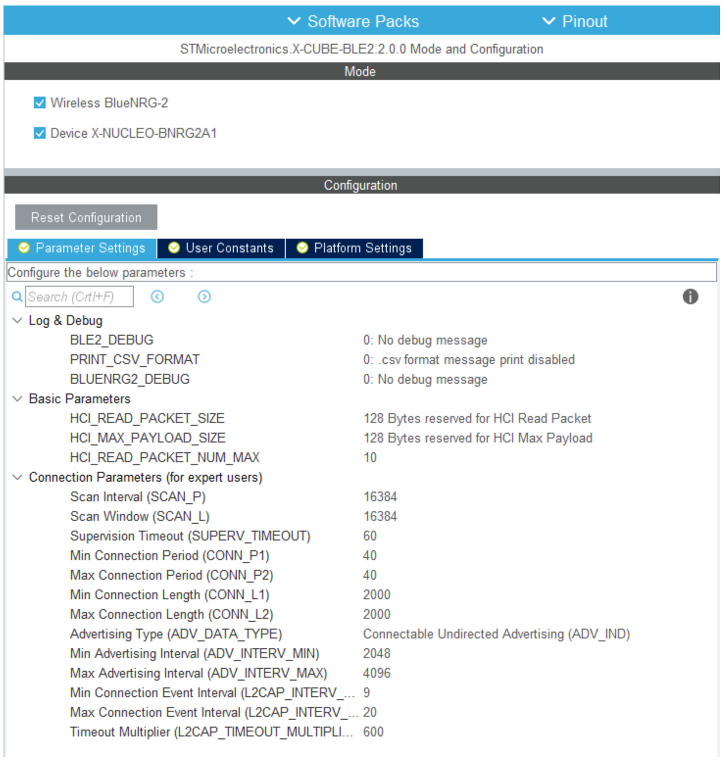
BSP

Name	IPs or Components	Found Solutions	BSP API
BSP BUTTON	GPIO:EXTI	PC13-ANTI_TAMP [B1 [Blue PushButton]]	BSP_COMMON_DRIVER
BSP USART	USART:Asynchronous	USART2	BSP_COMMON_DRIVER
BSP LED	GPIO:Output	PA5 [LD2 [Green Led]]	BSP_COMMON_DRIVER

Figure 18 *STMicroelectronics.X-CUBE-BLE2* ***Mode and Configuration*** *view*

From the Parameter Settings tab, some parameters for the data logging, the debugging and for the BLE scanning, advertising and connection can be set.
For all the sample applications, the default parameters can be used.
For the Beacon sample application, the Advertising Type and the Minimum and Maximum Advertising Intervals should be set as in the following table (but the application works also with the default parameters):

Beacon Sample Application	
Advertising Type (ADV_DATA_TYPE)	Non Connectable Undirected Advertising (ADV_NONCONN_IND)
Minimum Advertising Interval (ADV_INTERV_MIN)	1600
Maximum Advertising Interval (ADV_INTERV_MAX)	1600



STMicroelectronics.X-CUBE-BLE2.2.0.0 Mode and Configuration

Mode

☑ Wireless BlueNRG-2

☑ Device X-NUCLEO-BNRG2A1

Configuration

Configure the below parameters :

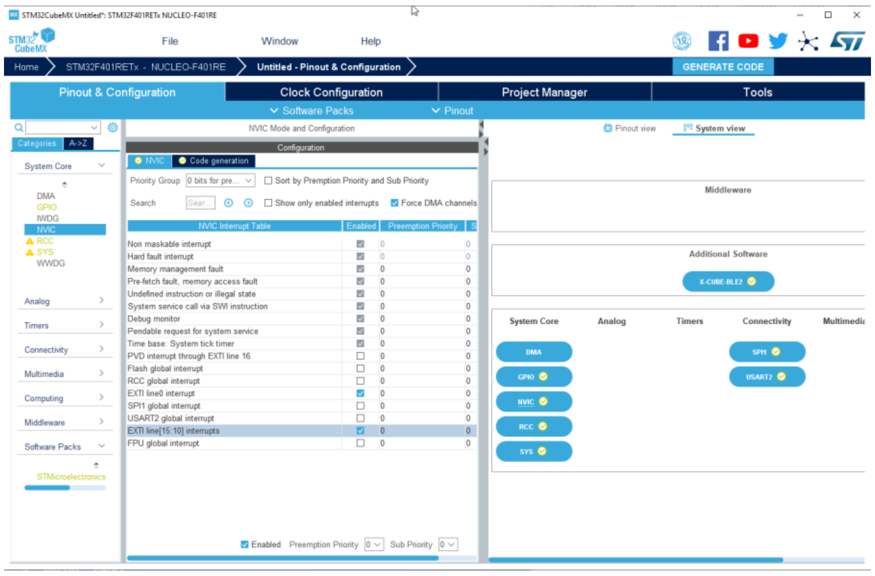
Log & Debug	
BLE2_DEBUG	0: No debug message
PRINT_CSV_FORMAT	0: .csv format message print disabled
BLUENRG2_DEBUG	0: No debug message
Basic Parameters	
HCI_READ_PACKET_SIZE	128 Bytes reserved for HCI Read Packet
HCI_MAX_PAYLOAD_SIZE	128 Bytes reserved for HCI Max Payload
HCI_READ_PACKET_NUM_MAX	10
Connection Parameters (for expert users)	
Scan Interval (SCAN_P)	16384
Scan Window (SCAN_L)	16384
Supervision Timeout (SUPERV_TIMEOUT)	60
Min Connection Period (CONN_P1)	40
Max Connection Period (CONN_P2)	40
Min Connection Length (CONN_L1)	2000
Max Connection Length (CONN_L2)	2000
Advertising Type (ADV_DATA_TYPE)	Connectable Undirected Advertising (ADV_IND)
Min Advertising Interval (ADV_INTERV_MIN)	2048
Max Advertising Interval (ADV_INTERV_MAX)	4096
Min Connection Event Interval (L2CAP_INTERV_...	9
Max Connection Event Interval (L2CAP_INTERV_...	20
Timeout Multiplier (L2CAP_TIMEOUT_MULTIPLI...	600

Figure 19 *BLE Parameter Settings*

From the **Configuration** tab, click on NVIC button under System to enable the EXTI line interrupts for both the SPI IRQ and the User Button (when used):

Name	Nucleo 64	Nucleo 144
Exti Line	EXTI line 0 interrupt	EXTI line 3 interrupt
BSP BUTTON	EXTI line 13 interrupt	EXTI line 13 interrupt

EXTI line interrupt in the green row must be enabled only for the SensorDemo_BLESensor-App and SampleApp sample applications.



NVIC Mode and Configuration

Configuration

NVIC | Code generation

Priority Group: 0 bits for pre... ☐ Sort by Premption Priority and Sub Priority

Search: Sear... ☐ Show only enabled interrupts ✓ Force DMA channels

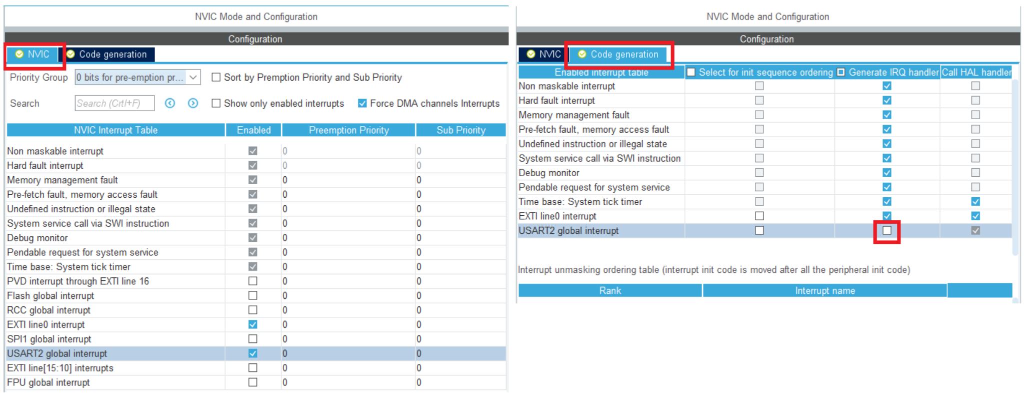
NVIC Interrupt Table	Enabled	Preemption Priority	S
Non maskable interrupt	✓	0	0
Hard fault interrupt	✓	0	0
Memory management fault	✓	0	0
Pre-fetch fault, memory access fault	✓	0	0
Undefined instruction or illegal state	✓	0	0
System service call via SWI instruction	✓	0	0
Debug monitor	✓	0	0
Pendable request for system service	✓	0	0
Time base: System tick timer	✓	0	0
PVD interrupt through EXTI line 16	☐	0	0
Flash global interrupt	☐	0	0
RCC global interrupt	☐	0	0
EXTI line0 interrupt	✓	0	0
SPI1 global interrupt	☐	0	0
USART2 global interrupt	☐	0	0
EXTI line[15:10] interrupts	✓	0	0
FPU global interrupt	☐	0	0

✓ Enabled Preemption Priority 0 Sub Priority 0

Figure 20 NVIC **Mode and Configuration** view

The IFRStack_Updater sample application requires this additional setting:

- enable the USART global interrupt from the `NVIC` view
- uncheck the Generate IRQ handler from `NVIC --> Code generation`



NVIC Mode and Configuration

Configuration

NVIC | Code generation

Priority Group: 0 bits for pre-emption pr... ☐ Sort by Premption Priority and Sub Priority

Search: Search (Crtl+F) ☐ Show only enabled interrupts ✓ Force DMA channels Interrupts

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

NVIC Mode and Configuration

Configuration

NVIC | Code generation

Enabled interrupt table	☐ Select for init sequence ordering	☐ Generate IRQ handler	Call HAL handler
Non maskable interrupt	☐	✓	☐
Hard fault interrupt	☐	✓	☐
Memory management fault	☐	✓	☐
Pre-fetch fault, memory access fault	☐	✓	☐
Undefined instruction or illegal state	☐	✓	☐
System service call via SWI instruction	☐	✓	☐
Debug monitor	☐	✓	☐
Pendable request for system service	☐	✓	☐
Time base: System tick timer	☐	✓	✓
EXTI line0 interrupt	☐	✓	✓
USART2 global interrupt	☐	☐	✓

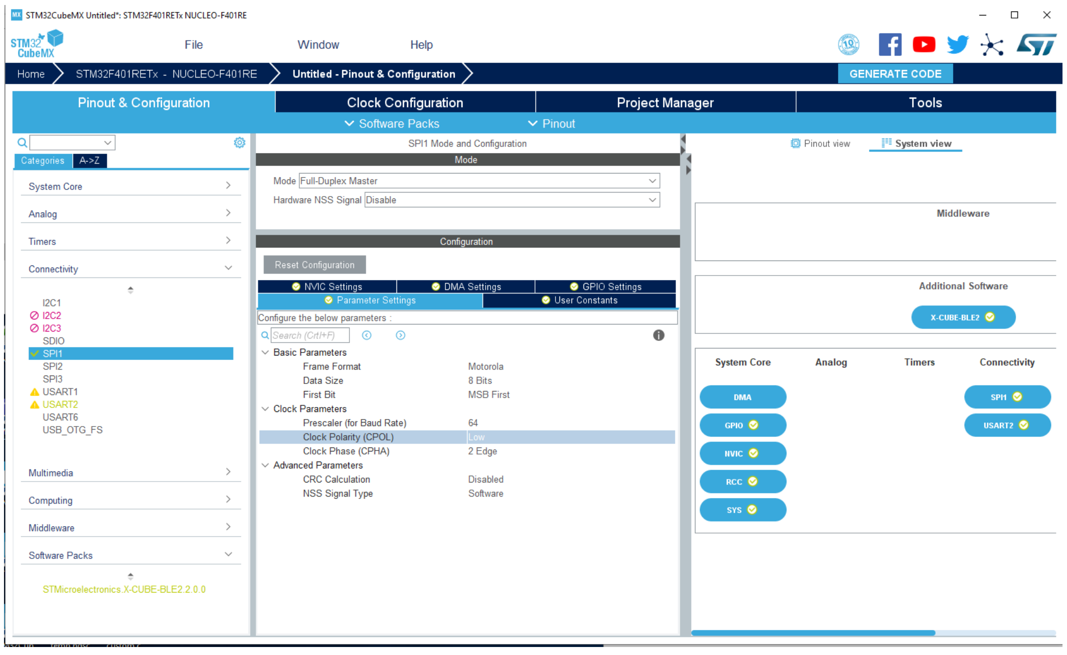
Interrupt unmasking ordering table (interrupt init code is moved after all the peripheral init code)

Rank	Interrupt name	

Figure 21 NVIC **settings for IFRStack_Updater**

From the System view, click on SPIx button under Connectivity category and:

- check that the Data size is 8 Bits;
- set the Prescaler (for Baud Rate) to a value so that the HClock/Prescaler is less or equal to 1000.0 KBits/s (the maximum supported SPI speed);
- set the Clock Phase (CPHA) to 2 Edge


SPI1 Mode and Configuration

Mode

Mode Full-Duplex Master

Hardware NSS Signal Disable

Configuration

Configure the below parameters :

Basic Parameters

Frame Format	Motorola
Data Size	8 Bits
First Bit	MSB First

Clock Parameters

Prescaler (for Baud Rate)	64
Clock Polarity (CPOL)	Low
Clock Phase (CPHA)	2 Edge

Advanced Parameters

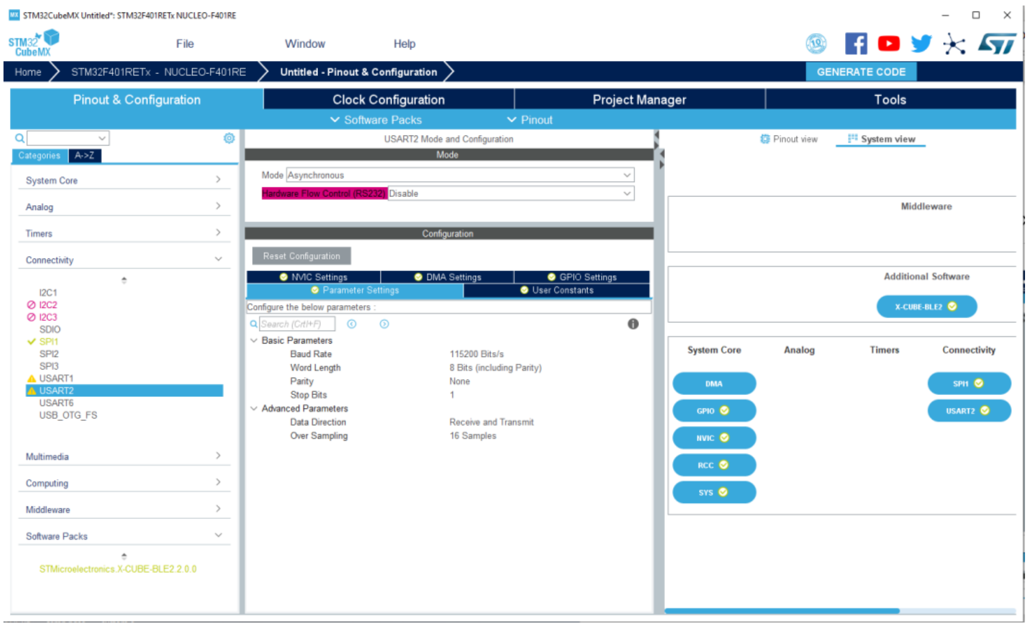
CRC Calculation	Disabled
NSS Signal Type	Software

Figure 22 *STM32CubeMX SPI Configuration*

From the System view, click on USARTx button under Connectivity category and check that the following configuration is set:

Baud Rate	115200 Bits/s
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1

Also, from the GPIO Settings tab, be sure the USART_TX and USART_RX labels are set.


USART2 Mode and Configuration

Mode

Mode Asynchronous

Hardware Flow Control (RS232) Disable

Configuration

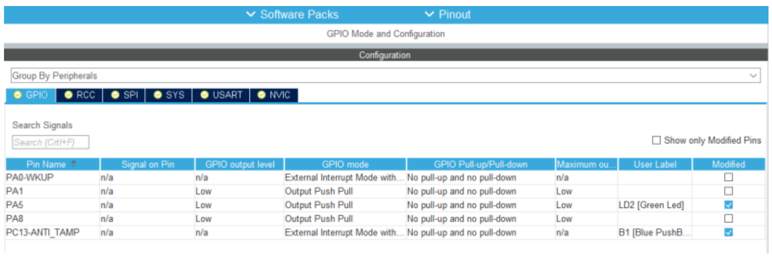
Configure the below parameters :

Basic Parameters

Baud Rate	115200 Bits/s
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1

Advanced Parameters

Data Direction	Receive and Transmit
Over Sampling	16 Samples

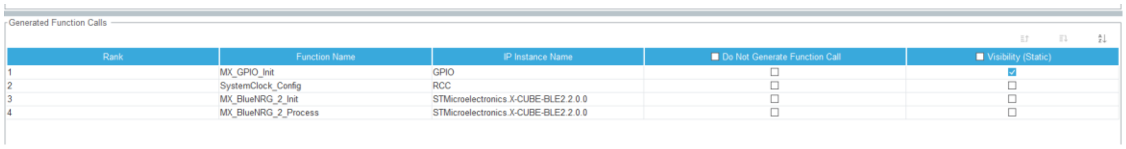


Pin Name	Signal on Pin	GPIO output level	GPIO mode	GPIO Pull-up/Pull-down	Maximum ou...	User Label	Modified
PA0-WKUP	n/a	n/a	External Interrupt Mode with...	No pull-up and no pull-down	n/a		☐
PA1	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low		☐
PA5	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low	LD2 [Green Led]	✓
PA8	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low		☐
PC13-ANTI_TAMP	n/a	n/a	External Interrupt Mode with...	No pull-up and no pull-down	n/a	B1 [Blue PushB...	✓

***Figure 23** USART Configuration*

Once all the above described steps have been performed, from the Project Manager tab the Project Name, the Project Location, the Toolchain/IDE, the Firmware Package Name and Version and so on can be set.

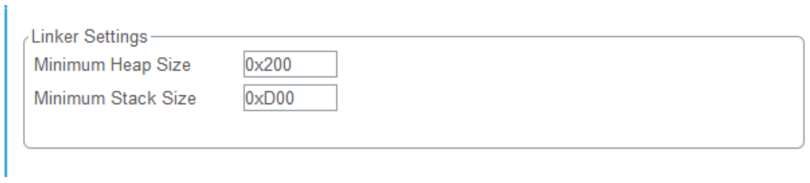
Hence, after checking that in the Advanced Settings tab the following options are set



Rank	Function Name	IP Instance Name	Do Not Generate Function Call	Visibility (Static)
1	MX_GPIO_Init	GPIO	☐	✓
2	SystemClock_Config	RCC	☐	☐
3	MX_BlueNRG_2_Init	STMicroelectronics.X-CUBE-BLE2.2.0.0	☐	☐
4	MX_BlueNRG_2_Process	STMicroelectronics.X-CUBE-BLE2.2.0.0	☐	☐

***Figure 24** Advanced Settings*

and that for the IFRStack_Updater sample application the following linker settings are used

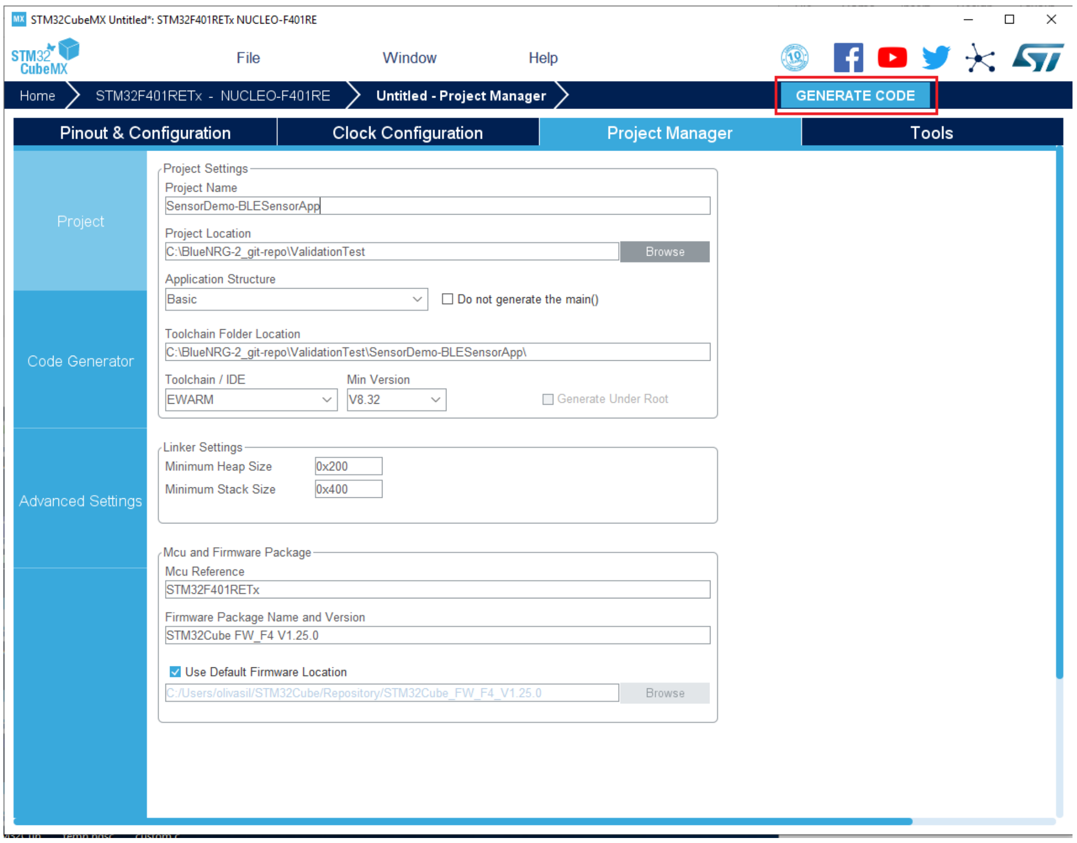


Linker Settings

Minimum Heap Size	0x200
Minimum Stack Size	0xD00

***Figure 25** IFRStack_Updater Linker Settings*

the source code of the project using the **STMicroelectronics X-CUBE-BLE2** software can be generated clicking the GENERATE CODE button.



Project Settings
Project Name: SensorDemo-BLESensorApp
Project Location: C:\BlueNRG-2_git-repo\ValidationTest
Application Structure: Basic
☐ Do not generate the main()
Toolchain Folder Location: C:\BlueNRG-2_git-repo\ValidationTest\SensorDemo-BLESensorApp\
Toolchain / IDE: EWARM
Min Version: V8.32
☐ Generate Under Root

Linker Settings
Minimum Heap Size: 0x200
Minimum Stack Size: 0x400

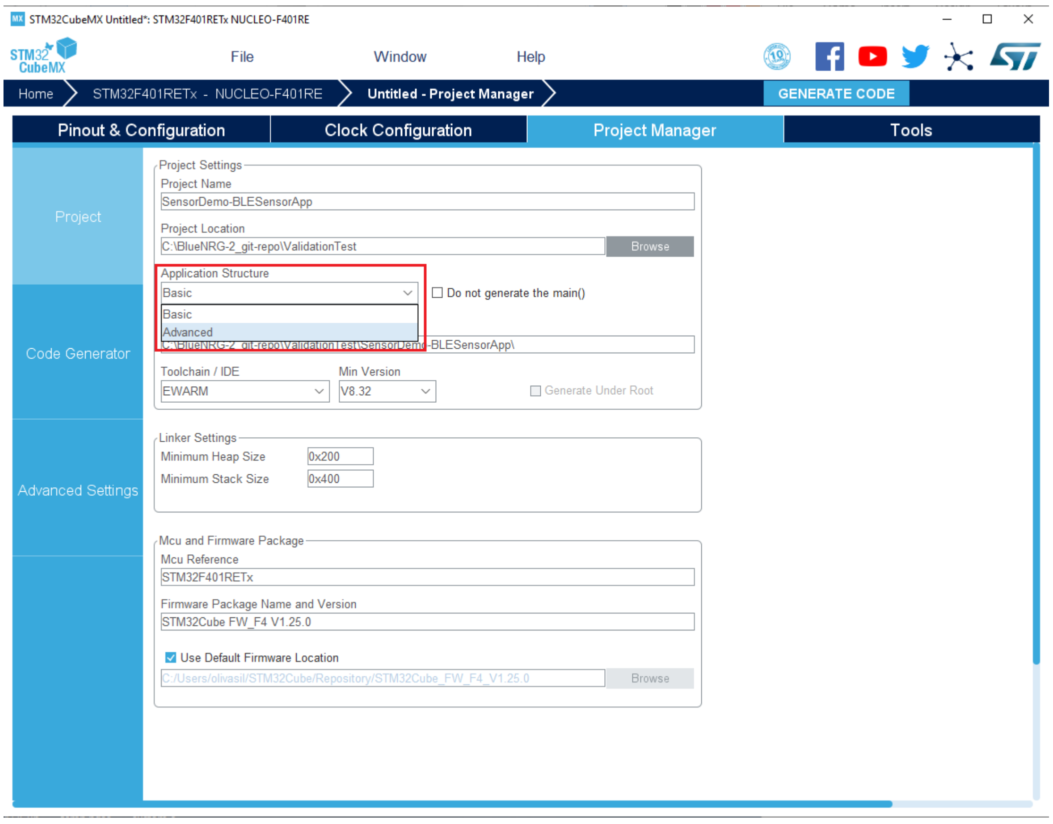
Mcu and Firmware Package
Mcu Reference: STM32F401RETx
Firmware Package Name and Version: STM32Cube FW_F4 V1.25.0
☑ Use Default Firmware Location
C:/Users/olivasil/STM32Cube/Repository/STM32Cube_FW_F4_V1.25.0

Figure 26 *Project Manager view*

9 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 middleware 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 middlewares are used.
  In the Advanced mode `Src` and `Inc` are generated under folder *Core*.
  For each middleware, the list of the generated files is under `<MW_Name>` (`BlueNRG-2` for the X-CUBE-BLE2 pack), at the same level as `Core` and either in `App` or in `Target` subfolder.



STM32CubeMX Untitled*: STM32F401RETx NUCLEO-F401RE

File Window Help

Home > STM32F401RETx - NUCLEO-F401RE > Untitled - Project Manager > GENERATE CODE

Pinout & Configuration | Clock Configuration | Project Manager | Tools

Project | Code Generator | Advanced Settings

Project Settings
Project Name: SensorDemo-BLESensorApp
Project Location: C:\BlueNRG-2_git-repo\ValidationTest [Browse]
Application Structure: Basic (Basic / Advanced) ☐ Do not generate the main()
C:\BlueNRG-2_git-repo\ValidationTest\SensorDemo-BLESensorApp\
Toolchain / IDE: EWARM Min Version: V8.32 ☐ Generate Under Root

Linker Settings
Minimum Heap Size: 0x200
Minimum Stack Size: 0x400

Mcu and Firmware Package
Mcu Reference: STM32F401RETx
Firmware Package Name and Version: STM32Cube FW_F4 V1.25.0
☑ Use Default Firmware Location
C:/Users/olivasil/STM32Cube/Repository/STM32Cube_FW_F4_V1.25.0 [Browse]

Figure 27 *Application Structure*

10 Known Limitations and workarounds

- For sample applications using any low power feature, such as **Beacon**, the ST-Link reset must be set in *Connect during reset* mode into the generated project configuration options.
- The Virtual_COM_Port sample application must be used with the following configuration for the HCI Transport Layer (HCI_TL) and the HCI Transport Layer Interface (HCI_TL_INTERFACE):
  - HCI_TL → Basic
  - HCI_TL_INTERFACE → UserBoard

  Other configurations using the template files are not supported yet.
- No support to **Low Level (LL) Driver** is provided yet for the SPI interface used by the BlueNRG-2 chip.
- To correctly set the RESET on pin D7 a 0 Ohm resistor must be soldiered on R117. Alternatively, the D7 pin of the Arduino connector and the pin #5 of the J12 on the X-NUCLEO-BNRG2A1 expansion board must be bridged.
- On dual-core STM32 series this expansion software can be used on both cores but exclusively.
- After generating the SensorDemo_BLESensor-App for the STM32L476RG MCU some additional steps must be manually executed for making the FOTA feature working. All the required instructions are contained in the readme.txt file that is generated by the STM32CubeMX in the application folder. Please, read it carefully.
- In MDK-ARM projects (e.g., for STM32F4, STM32H7 and STM32L4 series), if no log message is printed on the serial terminal, enable the Use MicroLib option in the project settings.

9 References

[1] UM2666 – User Manual - *Getting started with the X-CUBE-BLE2 Bluetooth Low Energy software expansion for STM32Cube*
[2] AN4979 – Application Notes – *Bluetooth Low Energy beacons with Eddystone*
[3] UM1724 – User Manual – *STM32 Nucleo-64 boards (MB1136)*

10 Revision history

Table 2: Document revision history

Date	Version	Changes
05-Dec-2019	1	Initial release
26-Jun-2020	2	Add Central sample application description. Update screenshots.
27-Jul-2020	3	Add IFRStack_Updater sample application description
03-Nov-2020	4	Add instructions for the FOTA feature in SensorDemo_BLESensor-App