f

Getting started with the STMicroelectronics X-CUBE-BLE1 software package for STM32CubeMX

Introduction

This document provides the guidelines to configure and use the X-CUBE-BLE1 software package V6.2.2 for STM32CubeMX (minimum required version V6.3.0). The document contains a description of the provided sample applications, a description of the steps required to configure a generic project using the BLE middleware, 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 ST BlueNRG-M0 network processor, the X-NUCLEO-IDB05A2 expansion board and the X-CUBE-BLE1 expansion software for Bluetooth Low Energy are available on www.st.com.

IMPORTANT: *The X-CUBE-BLE1 software package also supports the X-NUCLEO-IDB05A1 expansion board equipped with the BlueNRG-MS network processor.*

Contents

List of figures

1 Acronyms and abbreviations

Table 1: list of acronyms

Acronym	Description
BLE	Bluetooth Low Energy
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 SLA0077.

4 Sample Applications Description

In this section a short overview of the sample applications included in the X-CUBE-BLE1 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-MS/BlueNRG-M0 chip;
- show the users how to use the BLE APIs, provided by the BlueNRG-MS middleware, for correctly initialize and use a BLE device.

4.1 SensorDemo_BLESensor-App

This sample application contains an example that shows how to implement the Sensor Demo 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_ms.c*, the environmental and the motion data can be sent automatically (with 1 sec period) or when the User Button is pressed.

4.2 SampleApp

This sample application shows how to simply use the BLE Stack.
To test this application you need two STM32 Nucleo boards with their respective BlueNRG-M0 STM32 expansion boards. One board needs to be configured as Server-Peripheral role, while the other needs to be configured as Client-Central role. Before flashing the boards, please make sure to use the right configuration by enabling/disabling the *#define SERVER_ROLE* in file *app_bluenrg_ms.c*.
Once the two STM32 Nucleo boards have been configured (one as Client and the other as Server) and flashed, the connection between the two boards establishes (when the LED2 on the CLIENT turns off).
By pressing the USER button on one board, the LD2 LED on the other one gets toggled and viceversa.
If you have only one STM32 Nucleo board, you can program it as SERVER and use as CLIENT the BLE IOT app for Android devices.

4.3 Beacon

This example application shows how to use the BlueNRG Bluetooth Low Energy (BLE) expansion board to implement an Eddystone Beacon device.
An Eddystone Beacon 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.

Eddystone is an open beacon format from Google that works with Android and iOS.
Two different kinds of devices can be selected through *#define EDDYSTONE_BEACON_TYPE* in file *app_bluenrg_ms.c*:

- EDDYSTONE_UID_BEACON_TYPE: a UID beacon broadcasts a unique ID that provides proximity and general location information.
- EDDYSTONE_URL_BEACON_TYPE: a URL beacon broadcasts a packet containing an URL code usable by compatible applications.

To locate the beacon, it is necessary to have a scanner application running on a BLE-capable smartphone, such as one of the following ones for Android:

- Physical Web
- iBeacon & Eddystone Scanner
- Beacon Radar

An alternative is to use a *Physical Web* compatible browser like Google Chrome (version ≥ 44).

4.4 VirtualCOMPort

VirtualCOMPort is the application to be used for updating the BlueNRG-M0/BlueNRG-MS firmware on the X-NUCLEO-IDB05A2/ X-NUCLEO-IDB05A1 expansion boards.
It must be used along with the ST BlueNRG GUI available on st.com.
The User can also use this sample application to port the VCOM application to his specific BlueNRG-M0/BlueNRG-MS PCB (assuming that the customer PCB has a USB or RS232 I/O port available for PC connection).
This application provides an interface compliant with the Bluetooth Low Energy DTM test commands. Anyway, this application is not a reference application to be used for BlueNRG-M0/BlueNRG-MS application development and evaluation.

5 Installing the X-CUBE-BLE1 pack in STM32CubeMX

After downloading (from www.st.com), installing and launching the STM32CubeMX (V≥5.0.0), the X-CUBE-BLE1 pack can be installed in few steps.

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

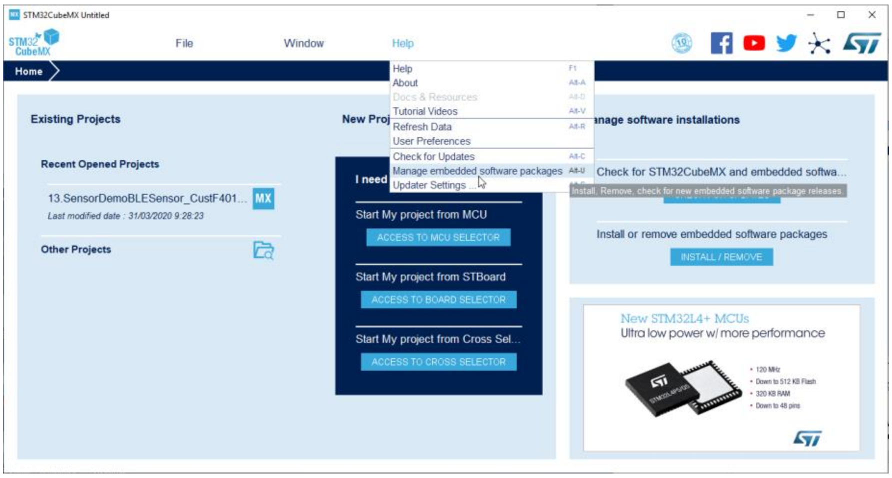
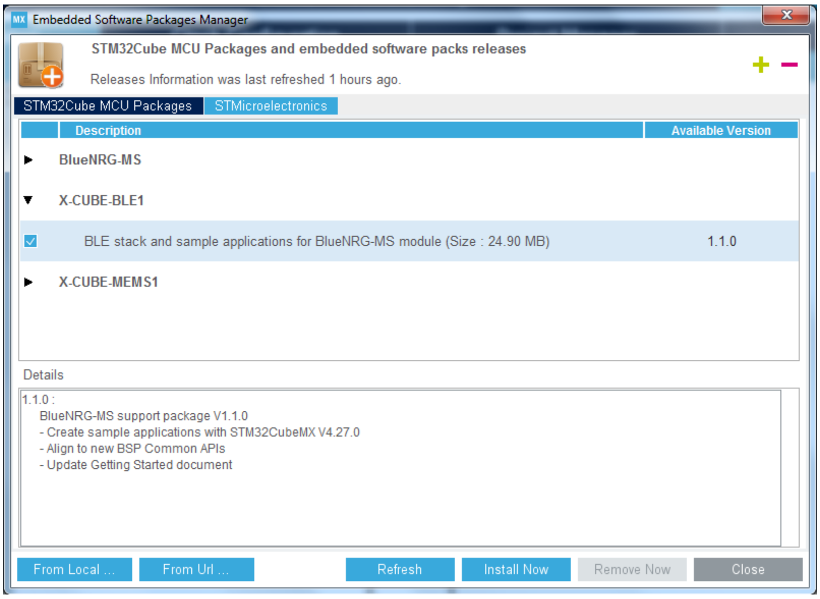

STM32CubeMX Untitled
STM32 CubeMX
File
Window
Help
Home
Help F1
About Alt-A
Docs & Resources Alt-D
Tutorial Videos Alt-V
Refresh Data Alt-R
User Preferences
Check for Updates Alt-C
Manage embedded software packages Alt-U
Updater Settings ...
Existing Projects
Recent Opened Projects
13.SensorDemoBLESensor_CustF401...
Last modified date : 31/03/2020 9:28:23
MX
Other Projects
New Proj
I need
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
nage software installations
Check for STM32CubeMX and embedded softwa...
Install, Remove, check for new embedded software package releases
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 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-BLE1 pack.



STM32Cube MCU Packages and embedded software packs releases

Releases Information was last refreshed 1 hours ago.

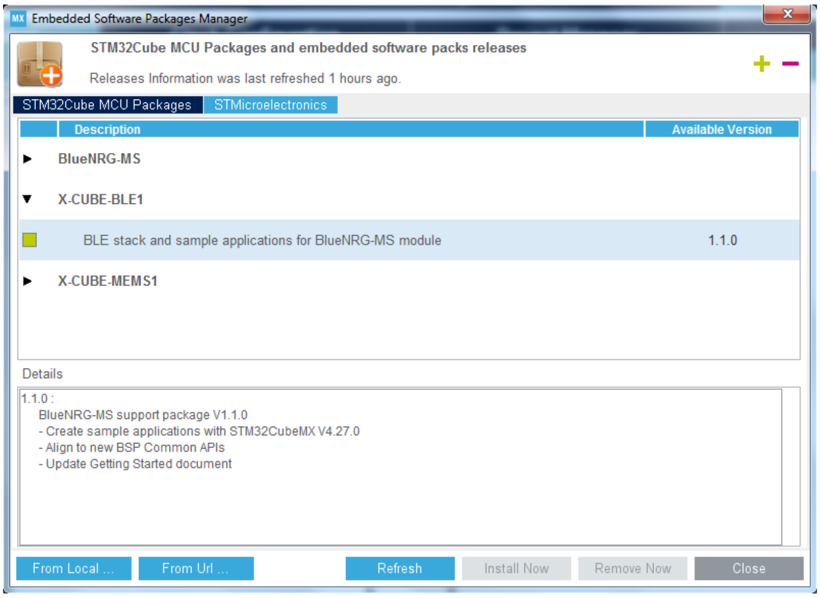
	Description	Available Version
▶	BlueNRG-MS	
▼	X-CUBE-BLE1	
☑	BLE stack and sample applications for BlueNRG-MS module (Size : 24.90 MB)	1.1.0
▶	X-CUBE-MEMS1	

Details

```
1.1.0 :
    BlueNRG-MS support package V1.1.0
    - Create sample applications with STM32CubeMX V4.27.0
    - Align to new BSP Common APIs
    - Update Getting Started document
```

Figure 2 Installing the X-CUBE-BLE1 pack in STM32CubeMX

3. Select the X-CUBE-BLE1 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 1 hours ago.

	Description	Available Version
▶	BlueNRG-MS	
▼	X-CUBE-BLE1	
🟩	BLE stack and sample applications for BlueNRG-MS module	1.1.0
▶	X-CUBE-MEMS1	

Details

```
1.1.0 :
    BlueNRG-MS support package V1.1.0
    - Create sample applications with STM32CubeMX V4.27.0
    - Align to new BSP Common APIs
    - Update Getting Started document
```

Figure 3 The X-CUBE-BLE1 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.

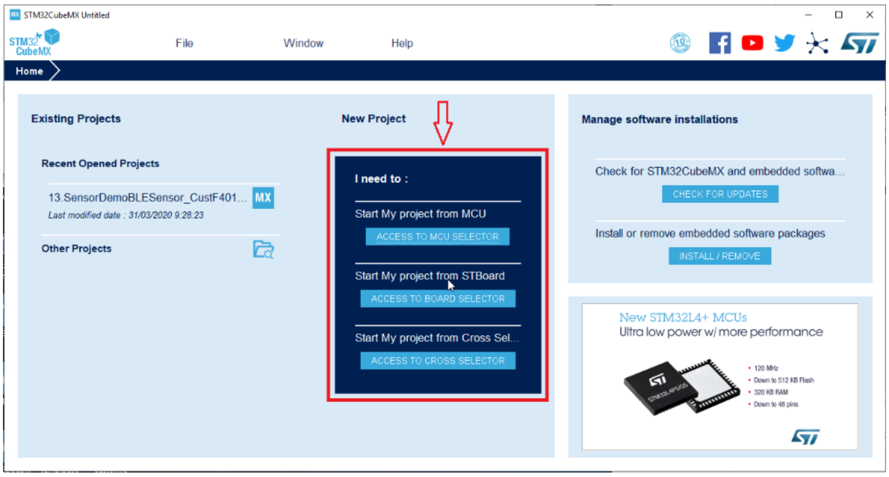

STM32CubeMX Untitled
STM32 CubeMX
File
Window
Help
Home
Existing Projects
Recent Opened Projects
13.SensorDemoBLESensor_CustF401...
Last modified date : 31/03/2020 9:28:23
Other Projects
New Project
I need to :
Start My project from MCU
ACCESS TO MCU SELECTOR
Start My project from STBoard
ACCESS TO BOARD 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.

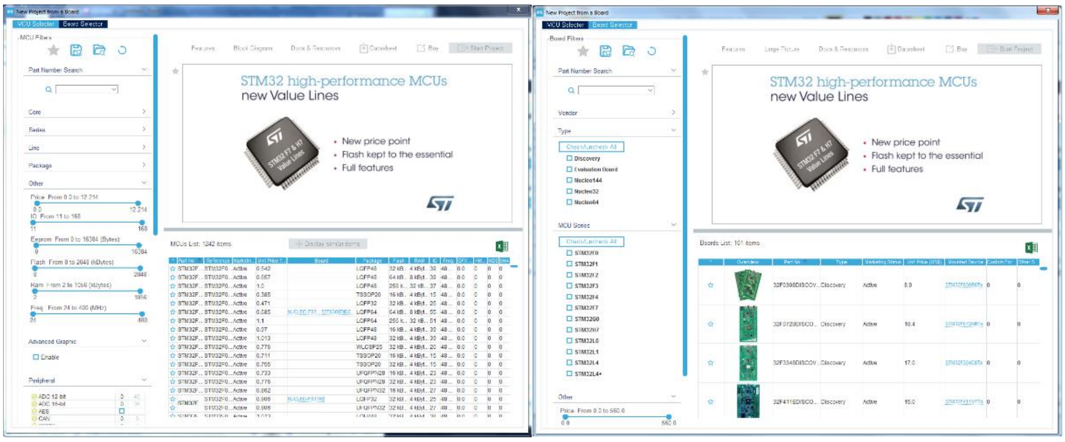


Figure 5 STM32CubeMX MCU/Board Selector windows

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 Pack and peripherals and configuring the clock.

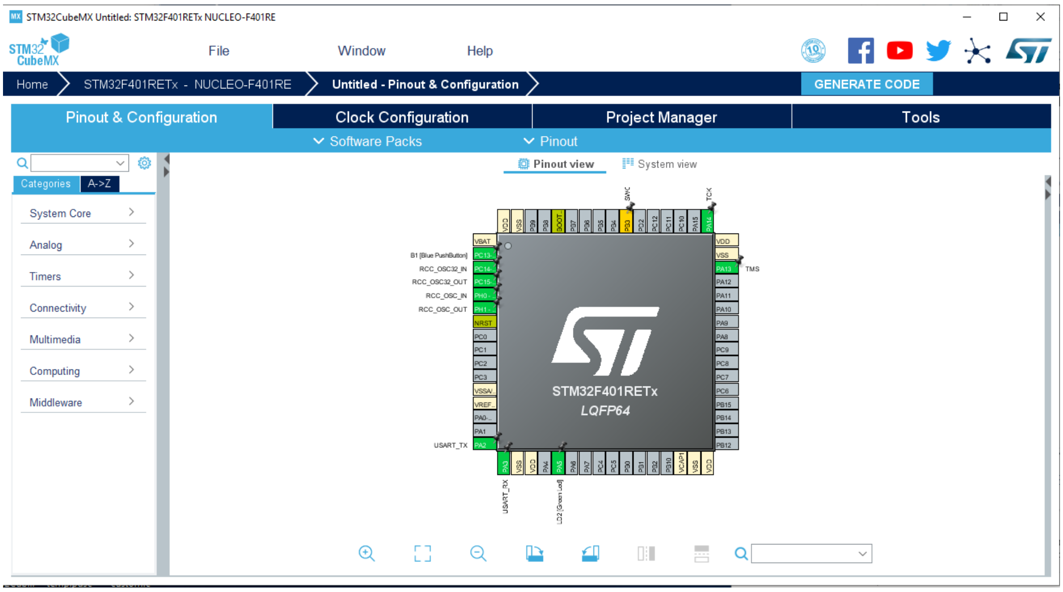
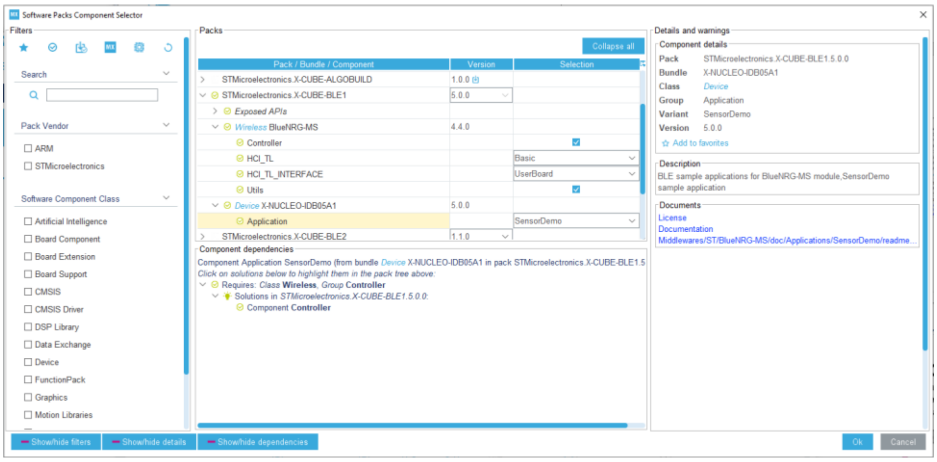


Figure 6 STM32CubeMX Configuration window

To add the X-CUBE-BLE1 additional software to the project, the button Software Packs → Select Components must be clicked.

From the Software Packs Component Selector window, the user can either chose 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.



Pack / Bundle / Component	Version	Selection
STMicroelectronics.X-CUBE-ALGOBUILD	1.0.0	
STMicroelectronics.X-CUBE-BLE1	5.0.0	
Exposed APIs		
Wireless BlueNRG-MS	4.4.0	
Controller		☑
HCI_TL		Basic
HCI_TL_INTERFACE		UserBoard
Utils		☑
Device X-NUCLEO-IDB05A1	5.0.0	
Application		SensorDemo
STMicroelectronics.X-CUBE-BLE2	1.1.0	

Component dependencies

Component Application SensorDemo (from bundle Device X-NUCLEO-IDB05A1 in pack STMicroelectronics.X-CUBE-BLE1.5

Click on solutions below to highlight them in the pack tree above:

Requires: *Class* **Wireless**, *Group* **Controller**

Solutions in *STMicroelectronics.X-CUBE-BLE1.5.0.0:*

Component **Controller**

Details and warnings

Component details

Pack	STMicroelectronics.X-CUBE-BLE1.5.0.0
Bundle	X-NUCLEO-IDB05A1
Class	Device
Group	Application
Variant	SensorDemo
Version	5.0.0

Add to favorites

Description

BLE sample applications for BlueNRG-MS module,SensorDemo sample application

Documents

License
Documentation
Middlewares/ST/BlueNRG-MS/doc/Applications/SensorDemo/readme...

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.

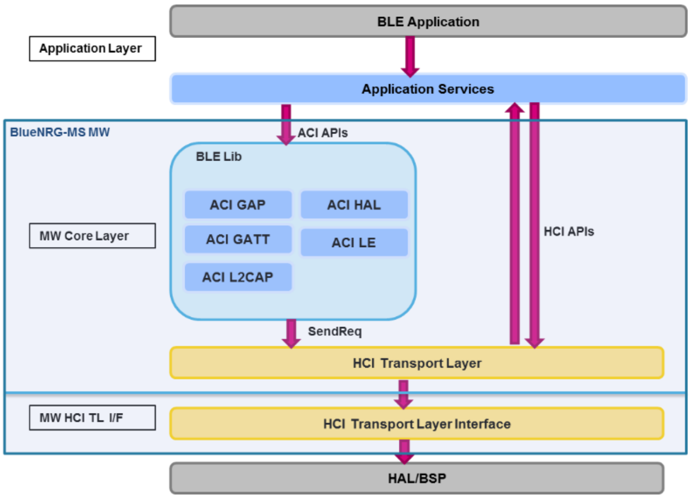


Figure 8 BlueNRG-M0/BlueNRG-MS software block scheme

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-IDB05A2 interfaces with the STM32 microcontroller via the SPI pin. Hence, assuming a user wants to interface the ST X-NUCLEO-IDB05A2 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.

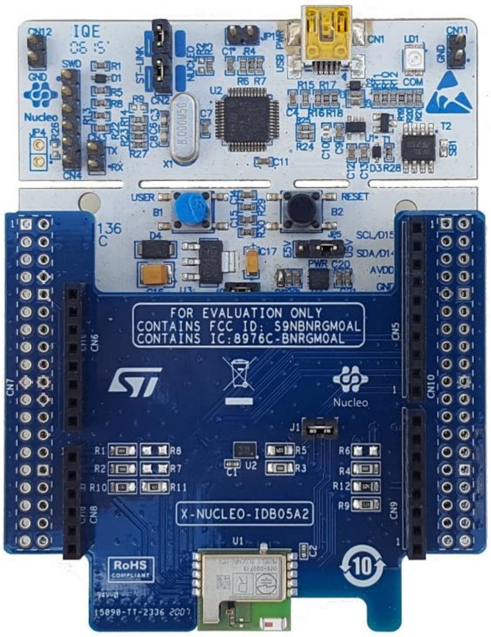


Figure 9 STM32 Nucleo 64 pins and X-NUCLEO-IDB05A2

If a Nucleo 144 pins is used, to correctly set the SPI clock on pin D13, the D3 pin and the D13 pin of the Arduino connector on the X-NUCLEO-IDB05A2 expansion board must be bridged (alternatively the resistor R9 must be open and a 0 Ohm resistor must be soldiered on R6). However, some Nucleo 144 pins does not require this modification. For instance, for the Nucleo-L496ZG-P, Nucleo-L496ZG, Nucleo-L4R5ZI and Nucleo-L4R5ZI-P, it is possible to assign the SPI clock to pin D3 without, consequently, the need to create a bridge between D3 pin and D13 pin.

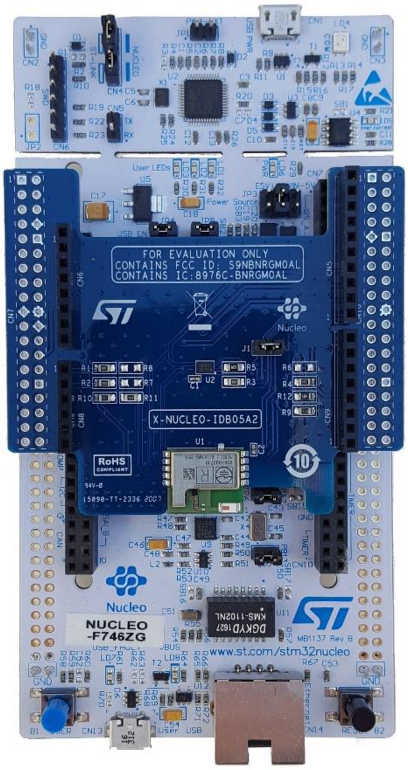


Figure 10 STM32 Nucleo 144 pins and X-NUCLEO-IDB05A2

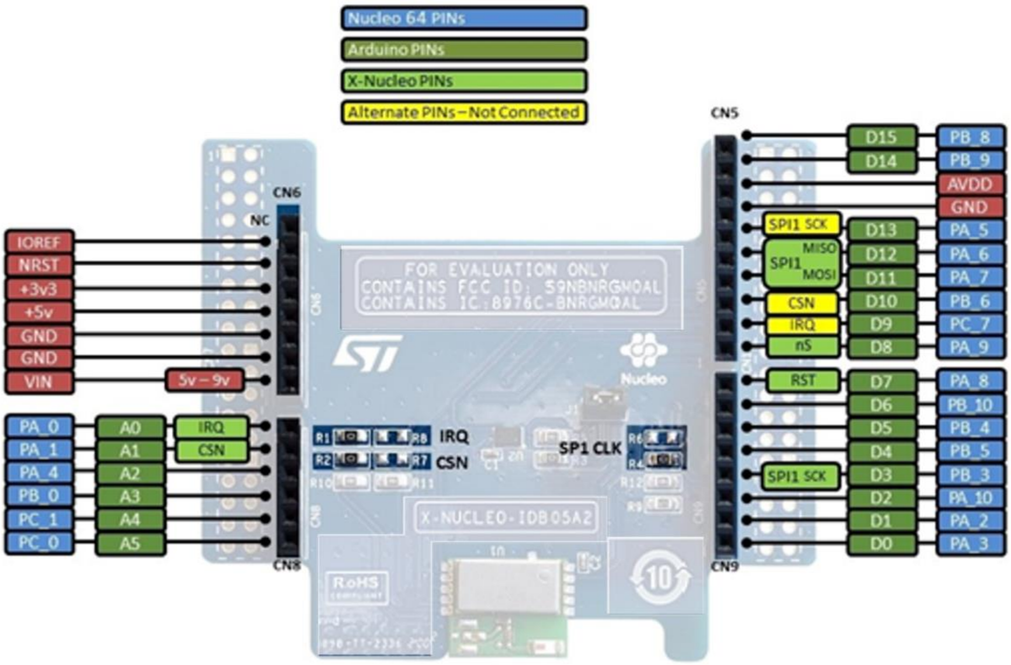


Figure 11 X-NUCLEO-IDB05A2 pinout

8.1 Use of Expansion Software without sample applications

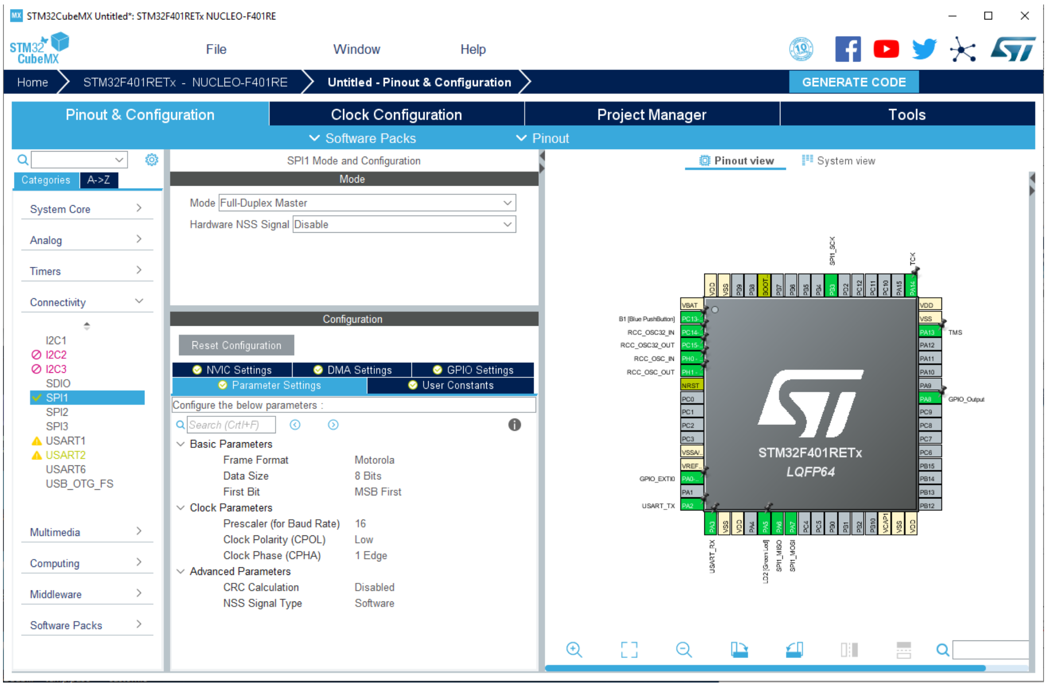
This section outlines how to configure STM32CubeMX when the use of the sample applications is not required. With such setup, only middleware and driver layers will be configured. This setup is useful when the user does not intend to leverage the sample application provided in the package.

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;
  - 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).

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



STM32CubeMX Untitled*: STM32F401RETx NUCLEO-F401RE

Home > STM32F401RETx - NUCLEO-F401RE > Untitled - Pinout & Configuration

SPI1 Mode and Configuration

Mode

Mode: Full-Duplex Master

Hardware NSS Signal: Disable

Configuration

Configure the below parameters :

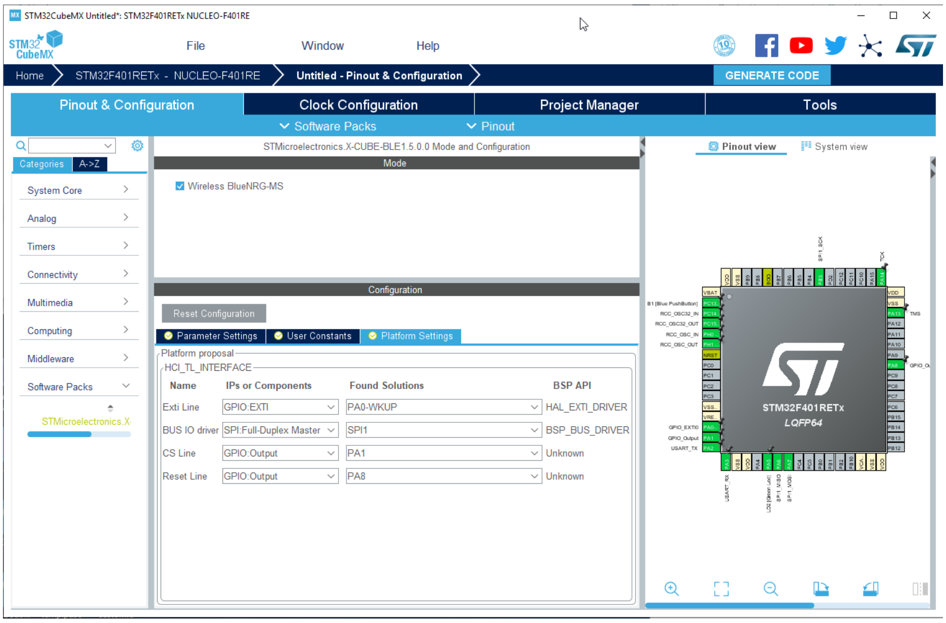
- Basic Parameters
  - Frame Format: Motorola
  - Data Size: 8 Bits
  - First Bit: MSB First
- Clock Parameters
  - Prescaler (for Baud Rate): 16
  - Clock Polarity (CPOL): Low
  - Clock Phase (CPHA): 1 Edge
- Advanced Parameters
  - CRC Calculation: Disabled
  - NSS Signal Type: Software

Figure 12 Pinout view

From the Software Packs > menu, click on the STMicroelectronics.X-CUBE-BLE1.x.y.z pack. Check the Wireless BlueNRG-MS 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-BLE1.5.0.0 Mode and Configuration

Mode

☑ Wireless BlueNRG-MS

Configuration

Platform proposal

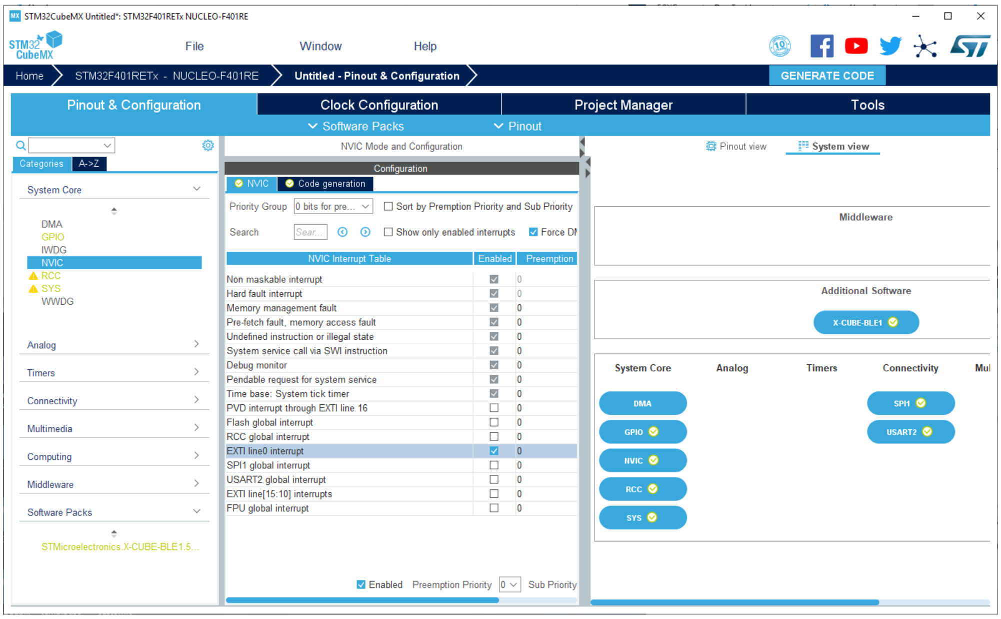
HCI_TL_INTERFACE

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-BLE1 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

NVIC | Code generation

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

Search: Sear... ☐ Show only enabled interrupts ☑ Force DI

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

☑ Enabled Preemption Priority 0 Sub Priority

Figure 14 NVIC Mode and Configuration view

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

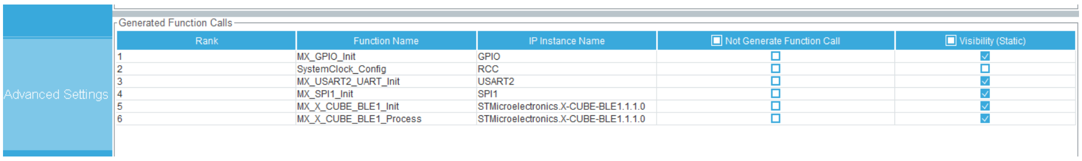
- check that the Data size is 8 Bits;
- set the Prescaler (for Baud Rate) to a value so that the HClock/Prescaler is less or equal to 8MHz (the maximum supported SPI speed)

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

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

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

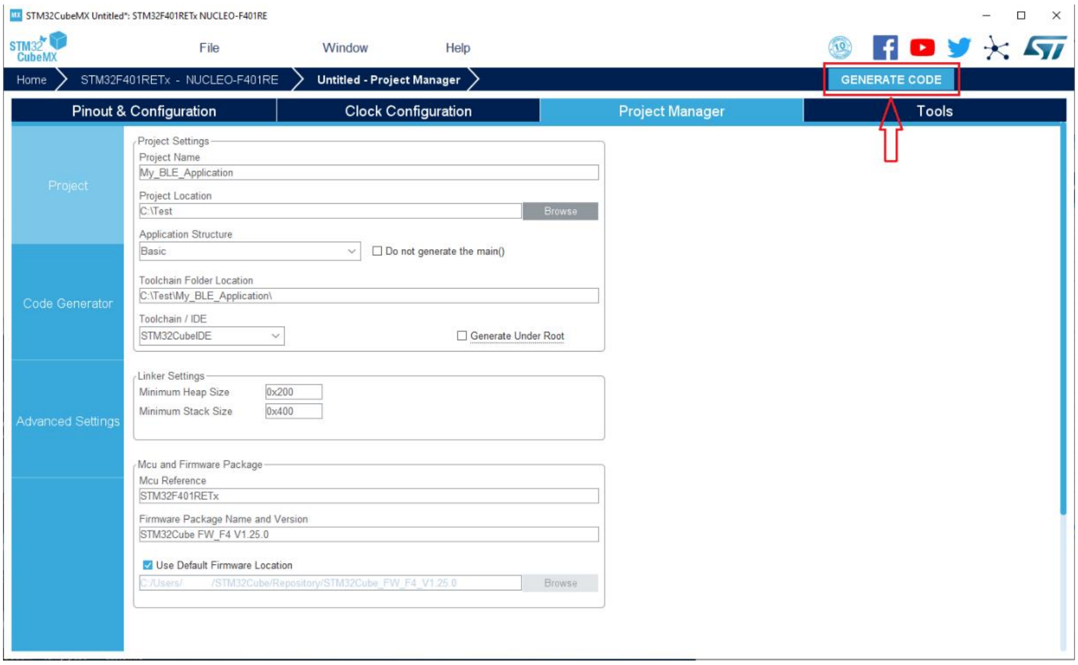


Generated Function Calls

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	☐	☑
4	MX_SPI1_Init	SPI1	☐	☑
5	MX_X_CUBE_BLE1_Init	STMicroelectronics.X-CUBE-BLE1.1.1.0	☐	☑
6	MX_X_CUBE_BLE1_Process	STMicroelectronics.X-CUBE-BLE1.1.1.0	☐	☑

Figure 15 Advanced Settings

the source code of the project using the **STMicroelectronics X-CUBE-BLE1** software can be generated clicking the GENERATE CODE button.



Project Settings

Project Name: My_BLE_Application

Project Location: C:\Test

Application Structure: Basic

☐ Do not generate the main()

Toolchain Folder Location: C:\Test\My_BLE_Application\

Toolchain / IDE: STM32CubeIDE

☐ 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/ /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:

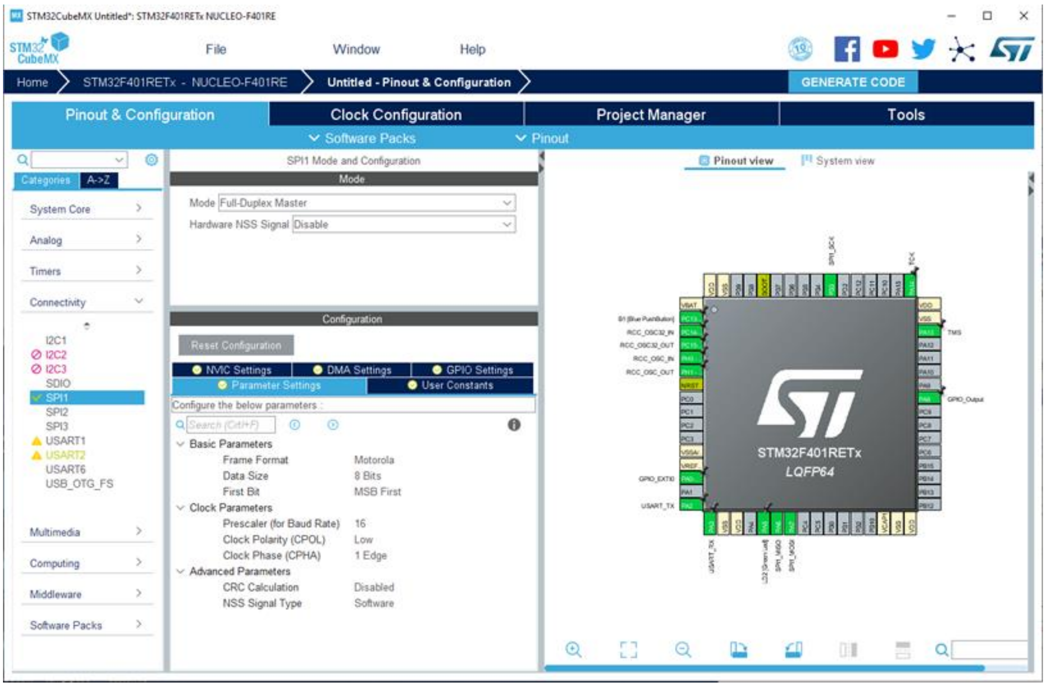
- from the Pinout view, if PB3 pin is already assigned, click on it and reset its state;
- from the ˅ Pinout → Clear Pinouts menu option reset the state of all pins (only for Nucleo 144);
- from the Connectivity > menu:
  - check that the ETH is disabled (only for Nucleo 144);
  - enable the SPI1 in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

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 and SampleApp 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 :

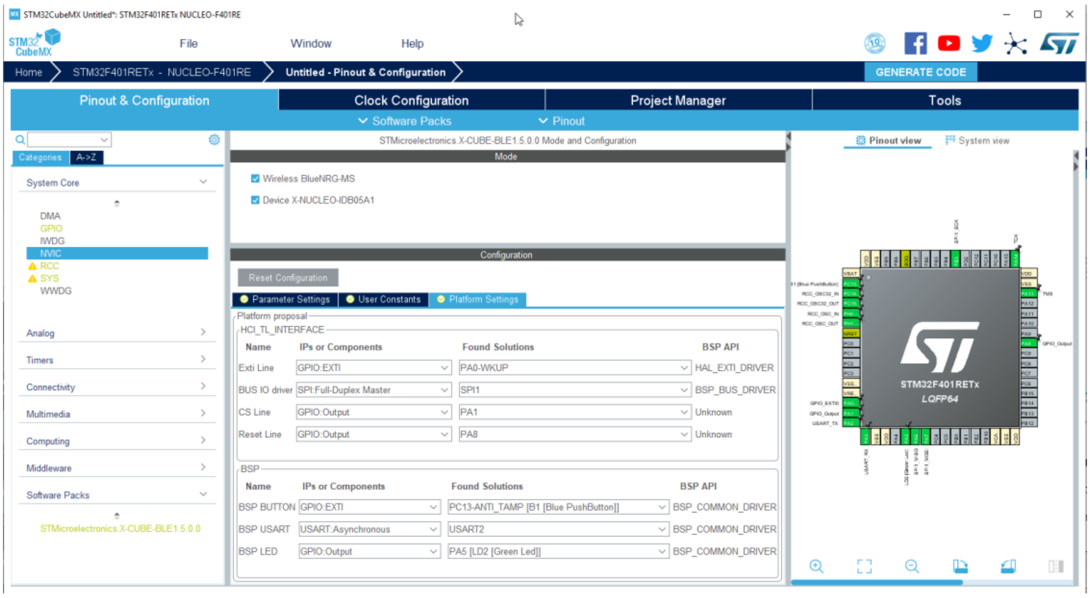
- Basic Parameters
  - Frame Format: Motorola
  - Data Size: 8 Bits
  - First Bit: MSB First
- Clock Parameters
  - Prescaler (for Baud Rate): 16
  - Clock Polarity (CPOL): Low
  - Clock Phase (CPHA): 1 Edge
- Advanced Parameters
  - CRC Calculation: Disabled
  - NSS Signal Type: Software

Figure 17 Pinout view

From the Software Packs > menu, click on the STMicroelectronics.X-CUBE-BLE1.x.y.z pack. Check the Wireless BlueNRG-MS 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 and SampleApp sample applications.



STMicroelectronics.X-CUBE-BLE1.5.0.0 Mode and Configuration

Mode

☑ Wireless BlueNRG-MS

☑ Device X-NUCLEO-IDB05A1

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

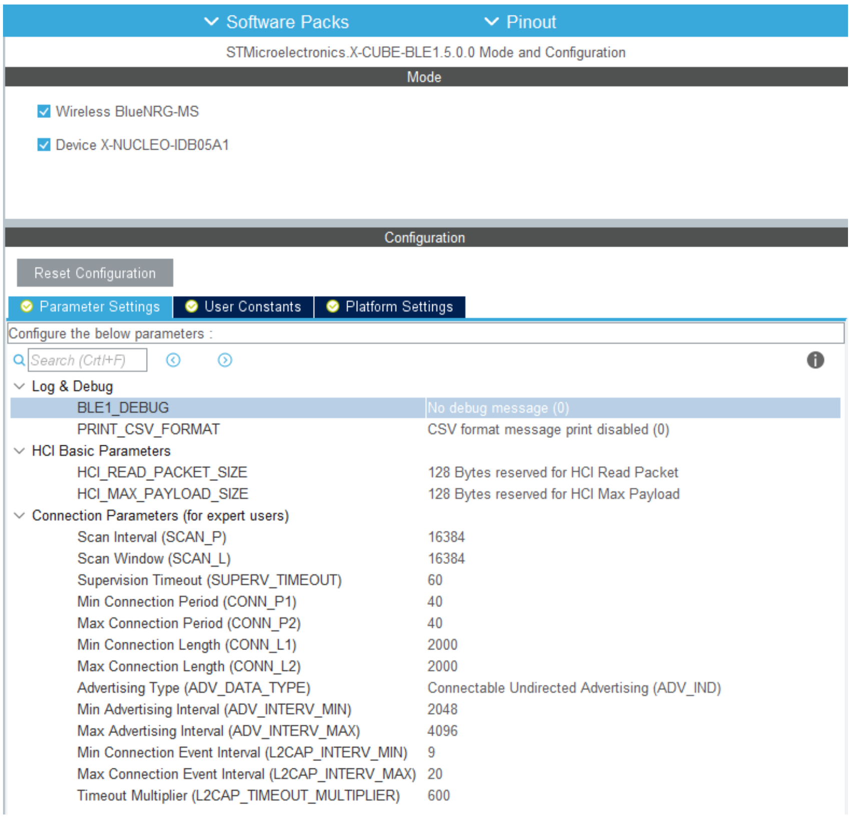
BSP

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-BLE1 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, apart from the Beacon one, the default parameters can be used. For the Beacon sample application, the Advertising Type and the Minimum and Maximum Advertising Intervals can be set as in the following table:

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



☑ Wireless BlueNRG-MS

☑ Device X-NUCLEO-IDB05A1

Configure the below parameters :

Log & Debug

BLE1_DEBUG	No debug message (0)
PRINT_CSV_FORMAT	CSV format message print disabled (0)

HCI Basic Parameters

HCI_READ_PACKET_SIZE	128 Bytes reserved for HCI Read Packet
HCI_MAX_PAYLOAD_SIZE	128 Bytes reserved for HCI Max Payload

Connection Parameters (for expert users)

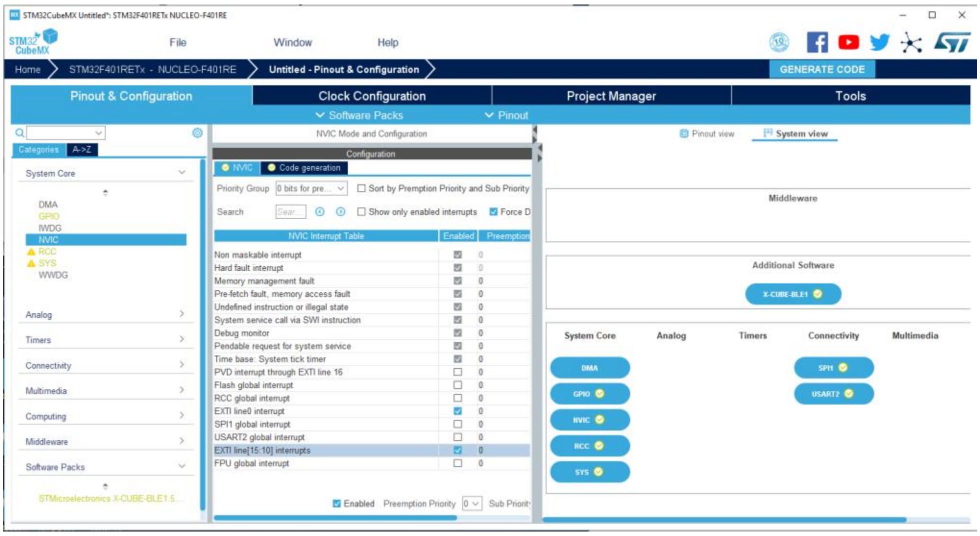
Scan Interval (SCAN_P)	16384
Scan Window (SCAN_L)	16384
Supervision Timeout (SUPERV_TIMEOUT)	60
Min Connection Period (CONN_P1)	40
Max Connection Period (CONN_P2)	40
Min Connection Length (CONN_L1)	2000
Max Connection Length (CONN_L2)	2000
Advertising Type (ADV_DATA_TYPE)	Connectable Undirected Advertising (ADV_IND)
Min Advertising Interval (ADV_INTERV_MIN)	2048
Max Advertising Interval (ADV_INTERV_MAX)	4096
Min Connection Event Interval (L2CAP_INTERV_MIN)	9
Max Connection Event Interval (L2CAP_INTERV_MAX)	20
Timeout Multiplier (L2CAP_TIMEOUT_MULTIPLIER)	600

Figure 19 BLE Connection 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.



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

Search: Sear... □ Show only enabled interrupts ☑ Force D

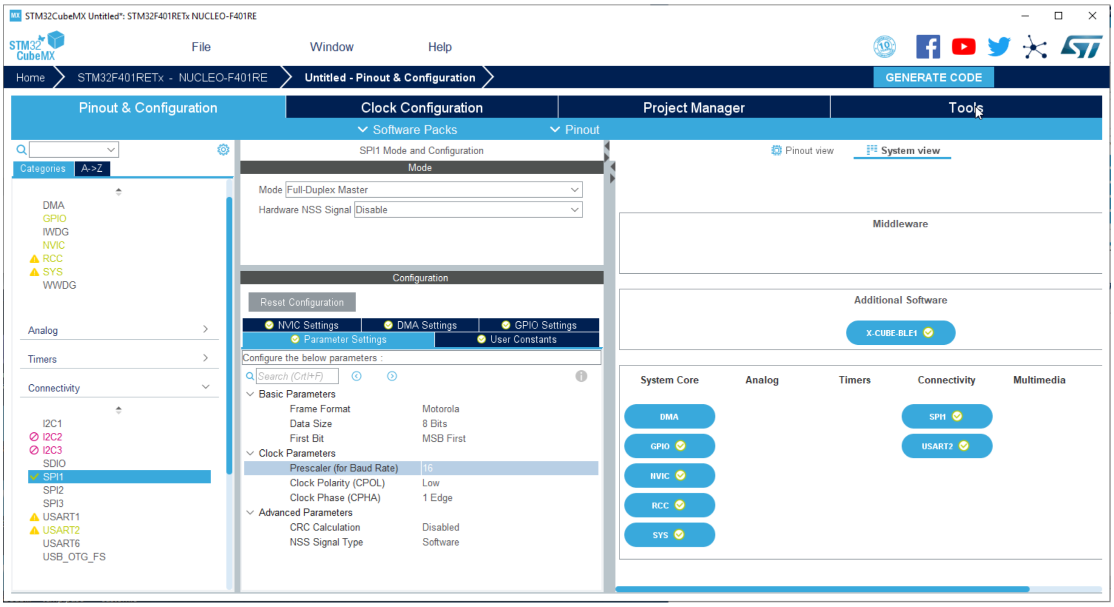
NVIC Interrupt Table	Enabled	Preemption
Non maskable interrupt	☑	0
Hard fault interrupt	☑	0
Memory management fault	☑	0
Pre-fetch fault, memory access fault	☑	0
Undefined instruction or illegal state	☑	0
System service call via SWI instruction	☑	0
Debug monitor	☑	0
Pendable request for system service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

☑ Enabled Preemption Priority 0 Sub Priority

Figure 20 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 8MHz (the maximum supported SPI speed).



Mode: Full-Duplex Master

Hardware NSS Signal: Disable

Configure the below parameters :

Basic Parameters

Parameter	Value
Frame Format	Motorola
Data Size	8 Bits
First Bit	MSB First

Clock Parameters

Parameter	Value
Prescaler (for Baud Rate)	16
Clock Polarity (CPOL)	Low
Clock Phase (CPHA)	1 Edge

Advanced Parameters

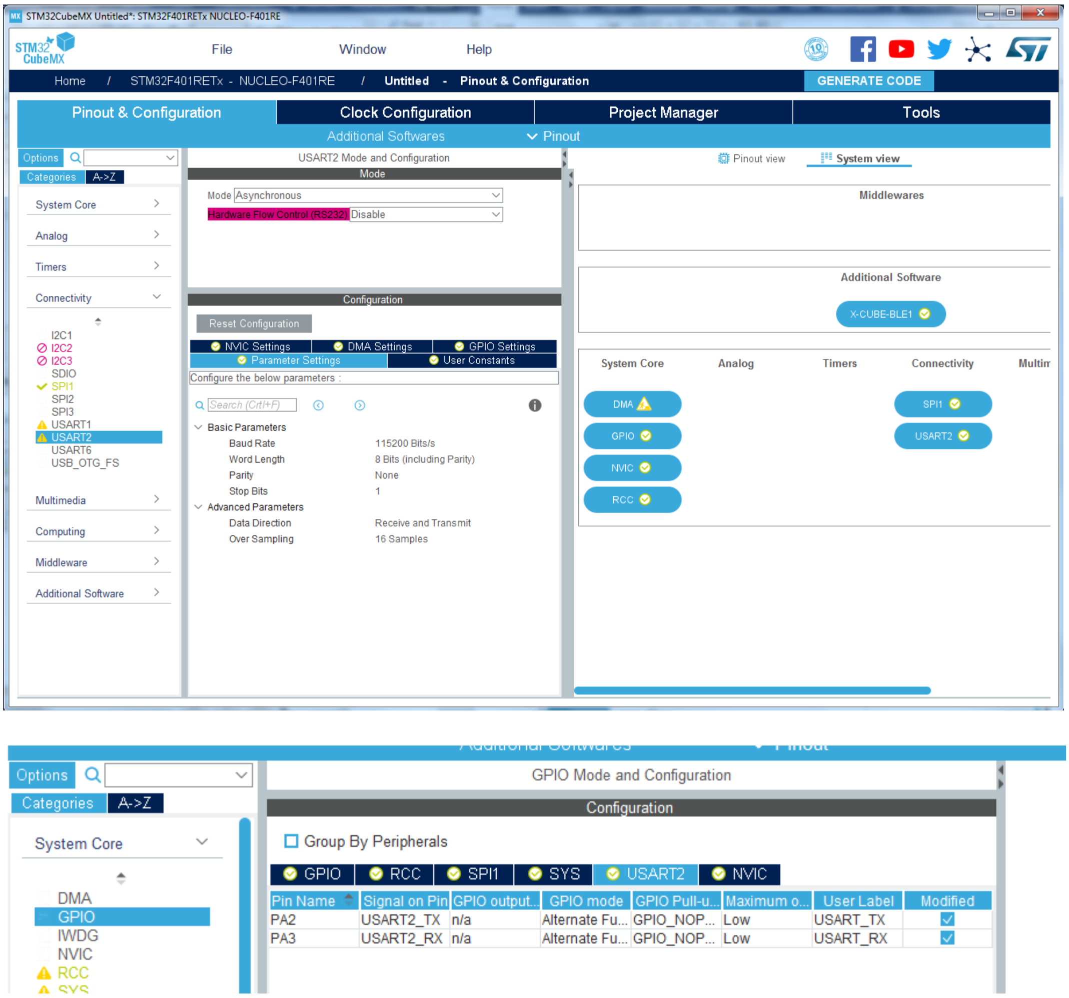
Parameter	Value
CRC Calculation	Disabled
NSS Signal Type	Software

Figure 21 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

Reset Configuration

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

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

GPIO Mode and Configuration

Configuration

□ Group By Peripherals

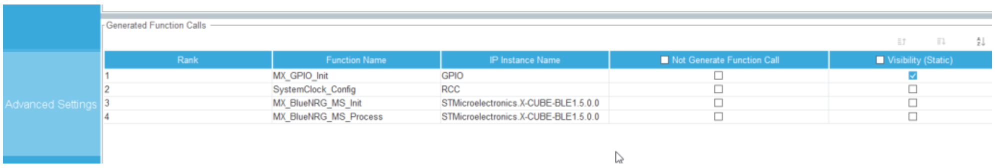
GPIO | RCC | SPI1 | SYS | USART2 | NVIC

Pin Name	Signal on Pin	GPIO output...	GPIO mode	GPIO Pull-u...	Maximum o...	User Label	Modified
PA2	USART2_TX	n/a	Alternate Fu...	GPIO_NOP...	Low	USART_TX	☑
PA3	USART2_RX	n/a	Alternate Fu...	GPIO_NOP...	Low	USART_RX	☑

Figure 22 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

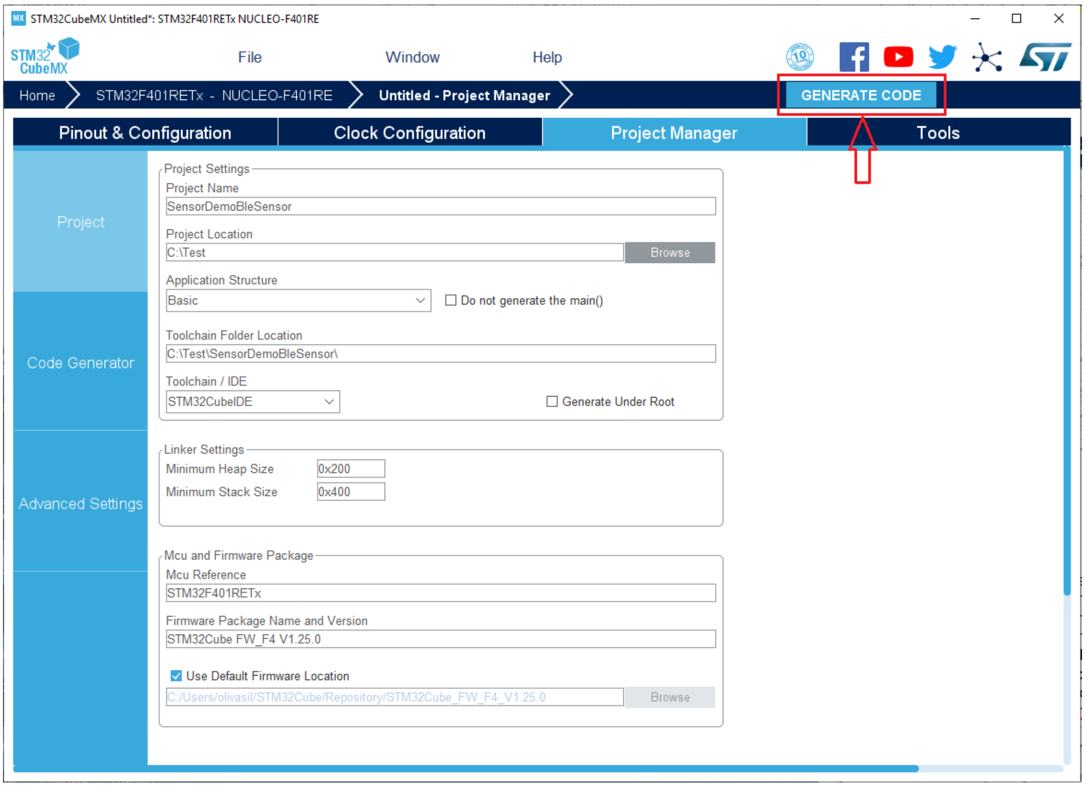


Generated Function Calls

Rank	Function Name	IP Instance Name	□ Not Generate Function Call	□ Visibility (Static)
1	MX_GPIO_Init	GPIO	□	☑
2	SystemClock_Config	RCC	□	□
3	MX_BlueNRG_MS_Init	STMicroelectronics.X-CUBE-BLE1.5.0.0	□	□
4	MX_BlueNRG_MS_Process	STMicroelectronics.X-CUBE-BLE1.5.0.0	□	□

Figure 23 Advanced Settings

the source code of the project using the **STMicroelectronics X-CUBE-BLE1** software can be generated clicking the GENERATE CODE button.



STM32CubeMX Untitled*: STM32F401RETx NUCLEO-F401RE

Project Settings

Project Name: SensorDemoBleSensor

Project Location: C:\Test [Browse]

Application Structure: Basic

☐ Do not generate the main()

Toolchain Folder Location: C:\Test\SensorDemoBleSensor\

Toolchain / IDE: STM32CubeIDE

☐ 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

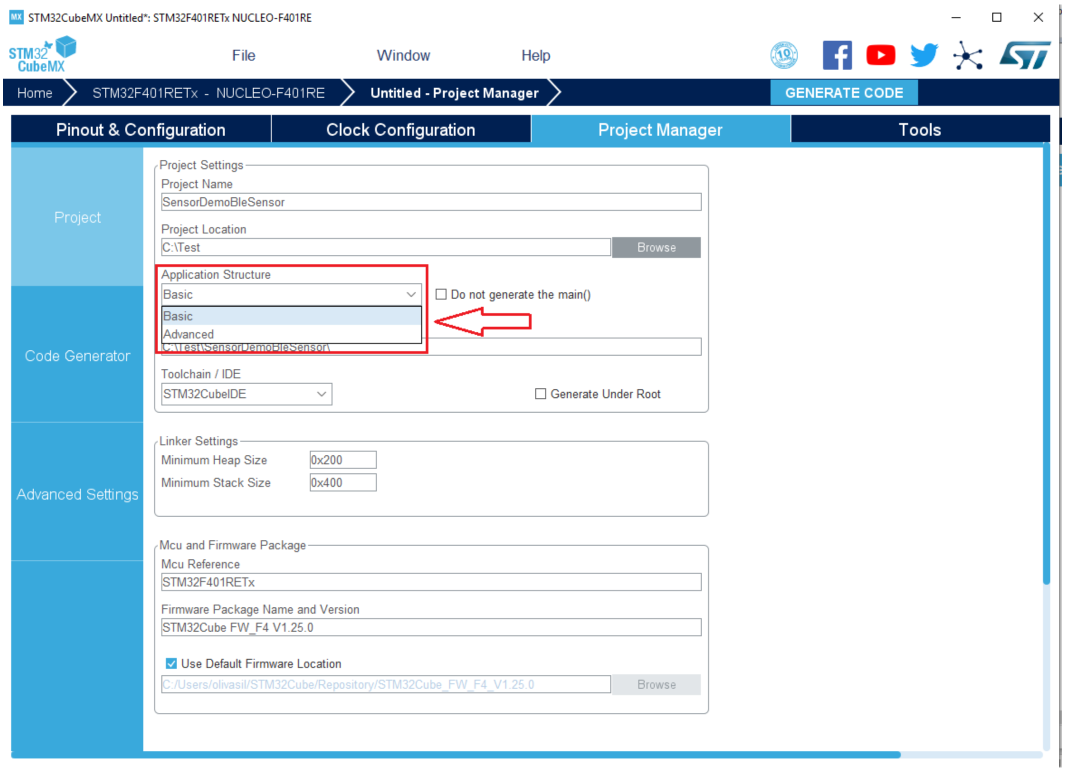
C:/Users/olivasil/STM32Cube/Repository/STM32Cube_FW_F4_V1.25.0 [Browse]

Figure 24 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 the application files are generated in folders *Core* (and subfolders *Src* and *Inc*) and *BlueNRG-MS* (and subfolders *App* and *Target*).



Project Settings

Project Name: SensorDemoBleSensor

Project Location: C:\Test

Application Structure: Basic (Basic / Advanced)

□ Do not generate the main()

C:\Test\SensorDemoBleSensor\

Toolchain / IDE: STM32CubeIDE

□ 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 25 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-M0/BlueNRG-MS chip.
- On dual-core STM32 series this expansion software can be used on both cores but exclusively.
- In MDK-ARM projects, if no log message is printed on the serial terminal or you face build errors mostly related to *fopen*, *fclose* and *fflush*, enable the Use MicroLib option in the project settings.

9 References

[1] UM1873 – User Manual - *Getting started with the X-CUBE-BLE1 Bluetooth Low Energy software expansion for STM32Cube* (see section 3.4 "Guide for writing applications")
[2] AN4642 – Application Notes – *Overview of the BLE Profiles application for X-CUBE-BLE1, expansion for STM32Cube*
[3] AN4979 – Application Notes – *Bluetooth Low Energy beacons with Eddystone*
[4] UM1724 – User Manual – *STM32 Nucleo-64 boards (MB1136)*
[5] X CUBE BLE1 for STM32CubeMX

10 Revision history

Table 2: Document revision history

Date	Version	Changes
15-Dec-2017	1	Initial release
25-Jun-2018	2	Add pack installation instructions
		Add HCI_TL and HCI_TL_INTERFACE configuration description
31-Aug-2018	3	Update pictures
12-Oct-2018	4	Add SensorDemo_BlueMS-App sample application description
13-Nov-2018	5	Align screenshots to STM32CubeMX v5.0.0 GUI
18-Feb-2019	6	Align to BlueNRG-MS pack v4.3.0
11-Jul-2019	7	Update sections: 8, 8.2
22-Apr-2020	8	Update screenshots to STM32CubeMX V6.0.0 Remove SensorDemo sample application description since no more supported
17-Mar-2021	9	Introduce references to BlueNRG-M0 network processor
27-Nov-2021	10	Remove reference to Flash Updater Tool since no more supported

IMPORTANT NOTICE – PLEASE READ CAREFULLY