User Guide

Wi-Fi® + Bluetooth® (Infineon) for STM32 Nucleo

RTOS User Guide - Rev. 1.0

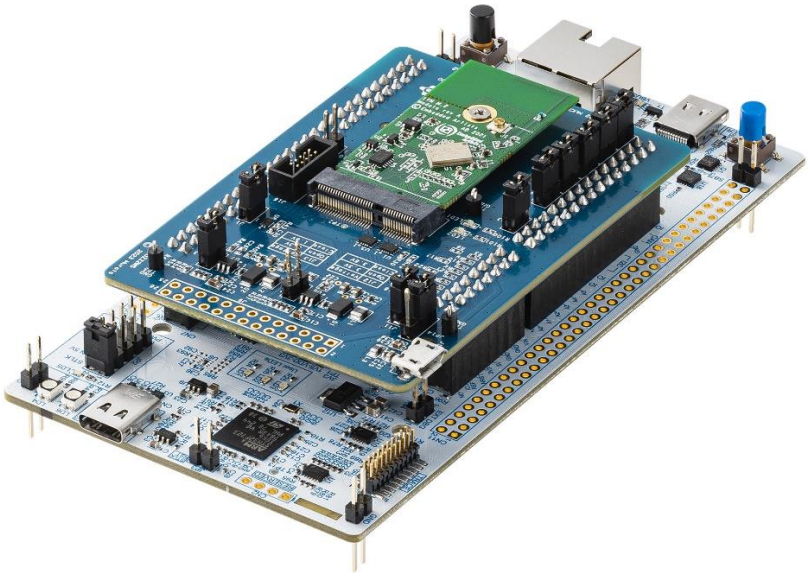


Table of Contents

1 Prerequisite 4
1.1 Supported STM32 Boards 4
1.2 Supported Connectivity Modules 4
1.3 Compatible Software 4
2 Download/Install/Import Expansion Pack 4
2.1 Downloading the Pack 4
2.2 Installing/Importing the Pack 4
3 Hardware Setup 8
3.1 Using NUCLEO-H563ZI Board 8
3.1.1 NUCLEO-H563ZI Board 10
3.1.2 Murata STM32 NUCLEO-M.2 Adapter Board 11
4 Example Projects 12
4.1 Wi-Fi Scan 12
4.1.1 Example Project Start/Import 12
4.1.2 Change Connectivity Module 13
4.1.3 Generate Code 14
4.1.4 Delete the Include Path Setting 16
4.1.5 Build the Project 18
4.1.6 Run Application 20
4.1.7 Serial Terminal Setup 21
4.1.8 Application Output 22
4.2 Bluetooth Hello Sensor 22
4.2.1 Example Project Start/Import 22
4.2.2 Change Connectivity Module to Serial Terminal Setup 23
4.2.3 Application Output 23
4.3 Tester - Wi-Fi Bluetooth Console 24
4.3.1 Example Project Start/Import 24
4.3.2 Change default Wi-Fi Connection Setting 24
4.3.3 Change Connectivity Module to Serial Terminal Setup 25
4.3.4 Application Output 25
4.3.5 Command List 25
4.3.6 iPerf Measurement 27
4.4 Other Example in the Pack 28
Revision History 29

Figures

Figure 1: STM32CubeMX Importing the Pack 5
Figure 2: STM32CubeMX Select Pack 5
Figure 3: STM32CubeMX Install Pack 6
Figure 4: STM32CubeMX Check Installed Pack 7
Figure 5:Jumper Settings (in Particular, JP4) on the NUCLEO-H563ZI Board 8
Figure 6: Combine the Three Boards 9
Figure 7: NUCLEO-H563ZI Board Features 10
Figure 8: Murata STM32 NUCLEO-M.2 Adapter Board Features 11
Figure 9: STM32CubeMX Select Component 13
Figure 10: STM32CubeMX Change Device and Module 14
Figure 11: STM32CubeMX Generate Code 15
Figure 12: STM32CubeMX Open Project 15
Figure 13: STM32CubeMX Select Workspace 16
Figure 14: STM32CubeIDE Select Properties 17
Figure 15: STM32CubeIDE Delete Include Paths 18
Figure 16: STM32CubeIDE Build the Project 19
Figure 17: STM32CubeIDE Build Successful Results 20
Figure 18: STM32CubeIDE Run Application 21
Figure 19: iPerf Setup 28

Tables

Table 1: Document Conventions 3
Table 2: NUCLEO Jumper Configuration 10
Table 3: Adapter Pinout Description 11
Table 4: Example List 12

About This Document

This document describes the steps for using Murata's Wi-Fi Module Type 1DX, Type 1YN, Type 1LV, Type 2AE, and Type 2BC on NUCLEO-H563ZI board with Infineon AIROC Wi-Fi/Bluetooth STM32 Expansion Pack 1.5.1.

The set-up process for STM32CubeMX and STM32CubeIDE is not described in this document.

Audience & Purpose

This document is intended for users who have previously used the STM32Cube. It is also intended for users with basic knowledge of Wi-Fi, Bluetooth, and C language.

Document Conventions

Table 1: Document Conventions

Conventions	Description
	**Warning Note** Indicates very important note. Users are strongly recommended to review.
	**Info Note** Intended for informational purposes. Users should review.
	**Menu Reference** Indicates menu navigation instructions. **Example**: Insert➔Tables➔Quick Tables➔Save Selection to Gallery 
	**External Hyperlink** This symbol indicates a hyperlink to an external document or website. **Example**: Embedded Artists AB  Click on the text to open the external link.
	**Internal Hyperlink** This symbol indicates a hyperlink within the document. **Example**: Prerequisite  Click on the text to open the link.
`Console input/output or code snippet`	**Console I/O or Code Snippet** This text ***Style*** denotes console input/output or a code snippet.
`# Console I/O comment` `// Code snippet comment`	**Console I/O or Code Snippet Comment** This text ***Style*** denotes a console input/output or code snippet comment. • Console I/O comment (preceded by "#") is for informational purposes only and does not denote actual console input/output. • Code Snippet comment (preceded by "//") may exist in the original code.

1 Prerequisite

1.1 Supported STM32 Boards

Infineon STM32 Connectivity Expansion Pack supports the following STM32 boards:

- NUCLEO-H563ZI

1.2 Supported Connectivity Modules

- Type 1DX (CYW4343W)
- Type 1YN (CYW43439)
- Type 1LV (CYW43012)
- Type 2AE (CYW4373E)
- Type 2BC (CYW4373)

1.3 Compatible Software

- STM32CubeMX 6.8.0
- STM32CubeIDE 1.12.0

2 Download/Install/Import Expansion Pack

2.1 Downloading the Pack

Download the expansion pack from GitHub.

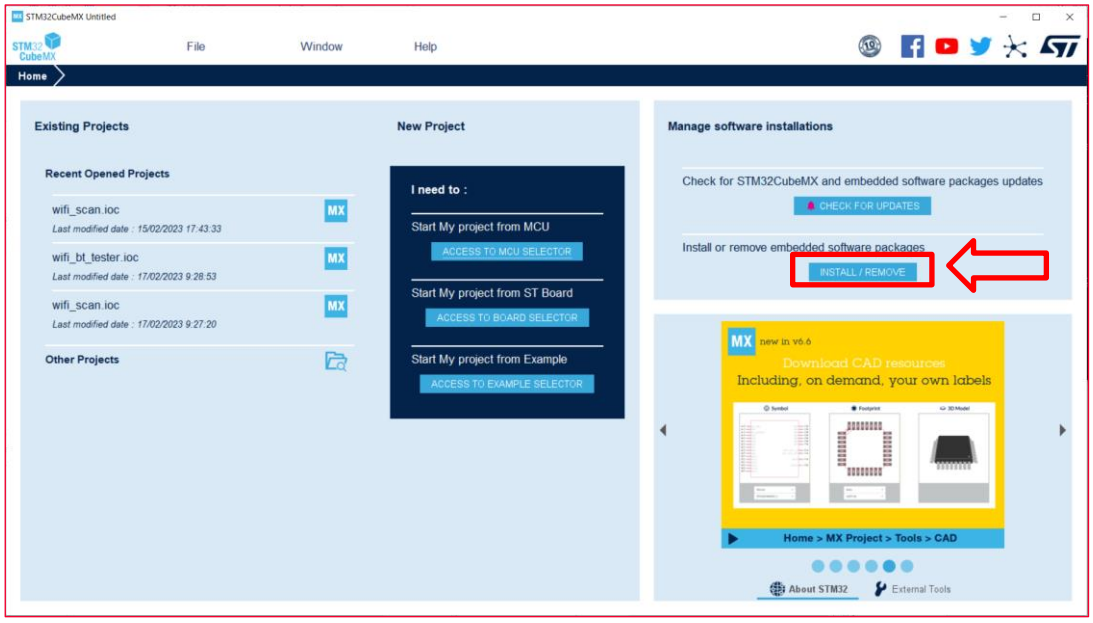
2.2 Installing/Importing the Pack

To add the expansion pack to the STM32 development environment:

1. Run the STM32CubeMX tool.
2. Select Home➔Manage Software Installations➔Install/Remove .

Figure 1 shows the STM32CubeMX importing pack option.

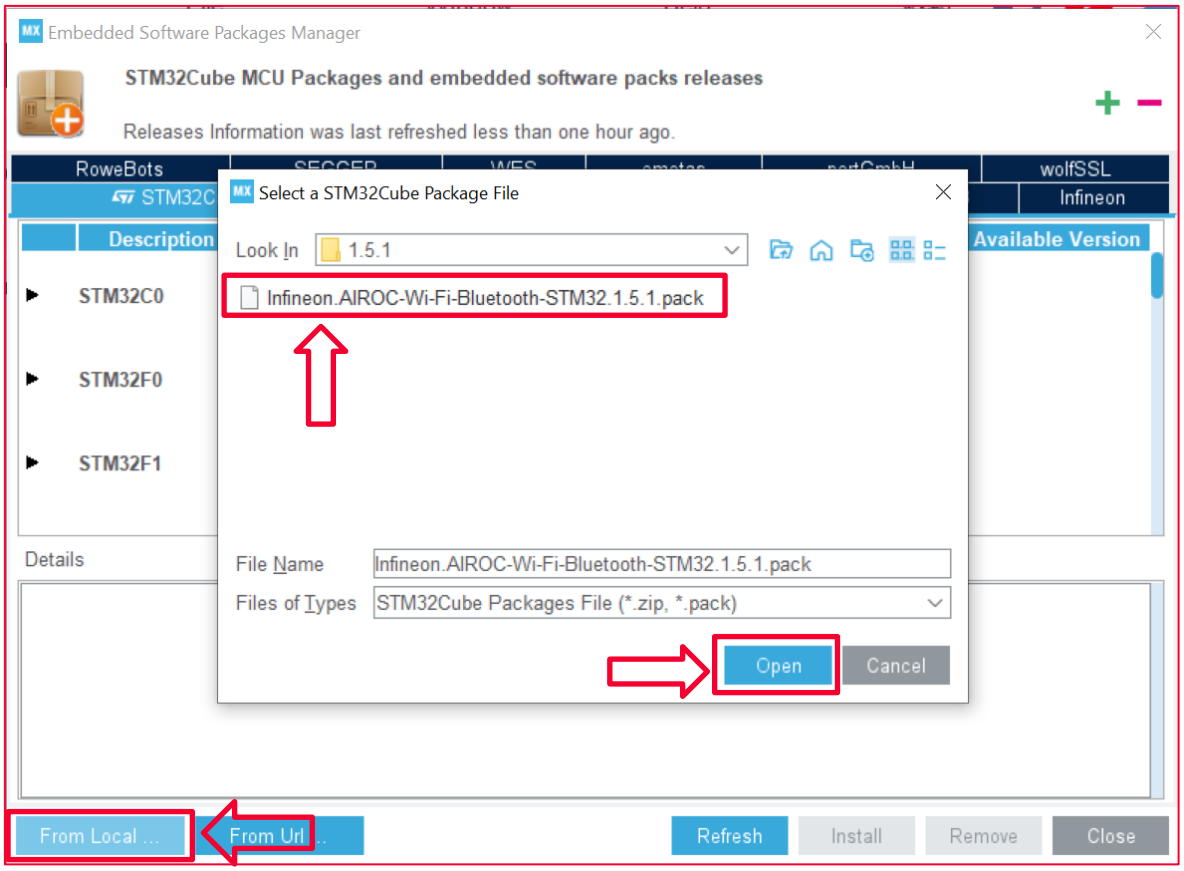
Figure 1: STM32CubeMX Importing the Pack



3. Select **From Local…**, navigate to the downloaded pack file, and select **Open**.

Figure 2 shows the STM32CubeMX pack selection.

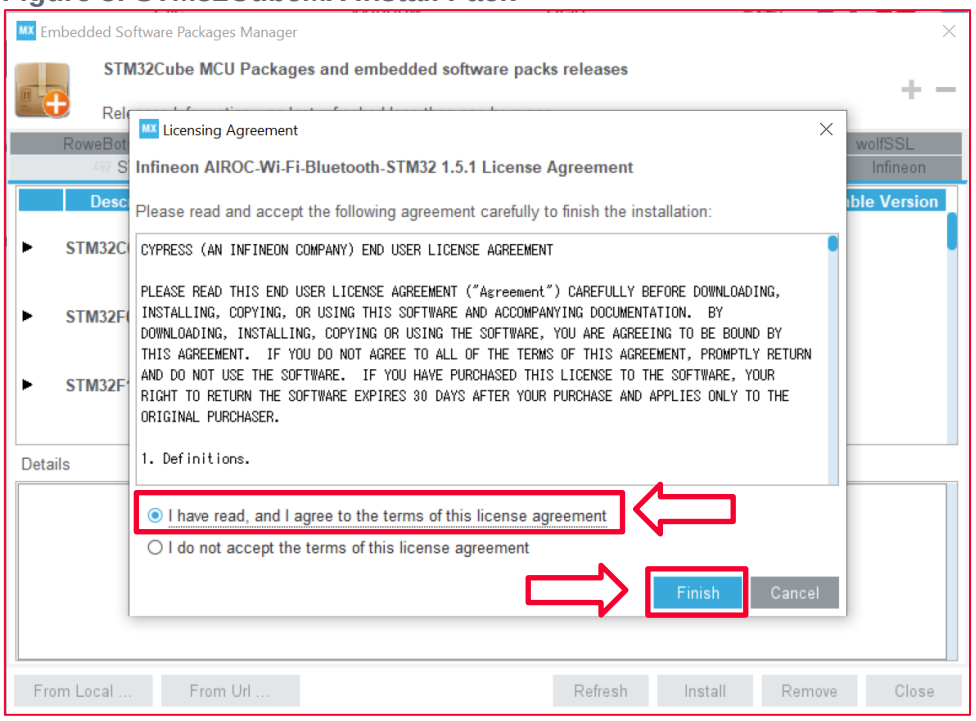
Figure 2: STM32CubeMX Select Pack



4. Accept the license agreement and select **Finish**.

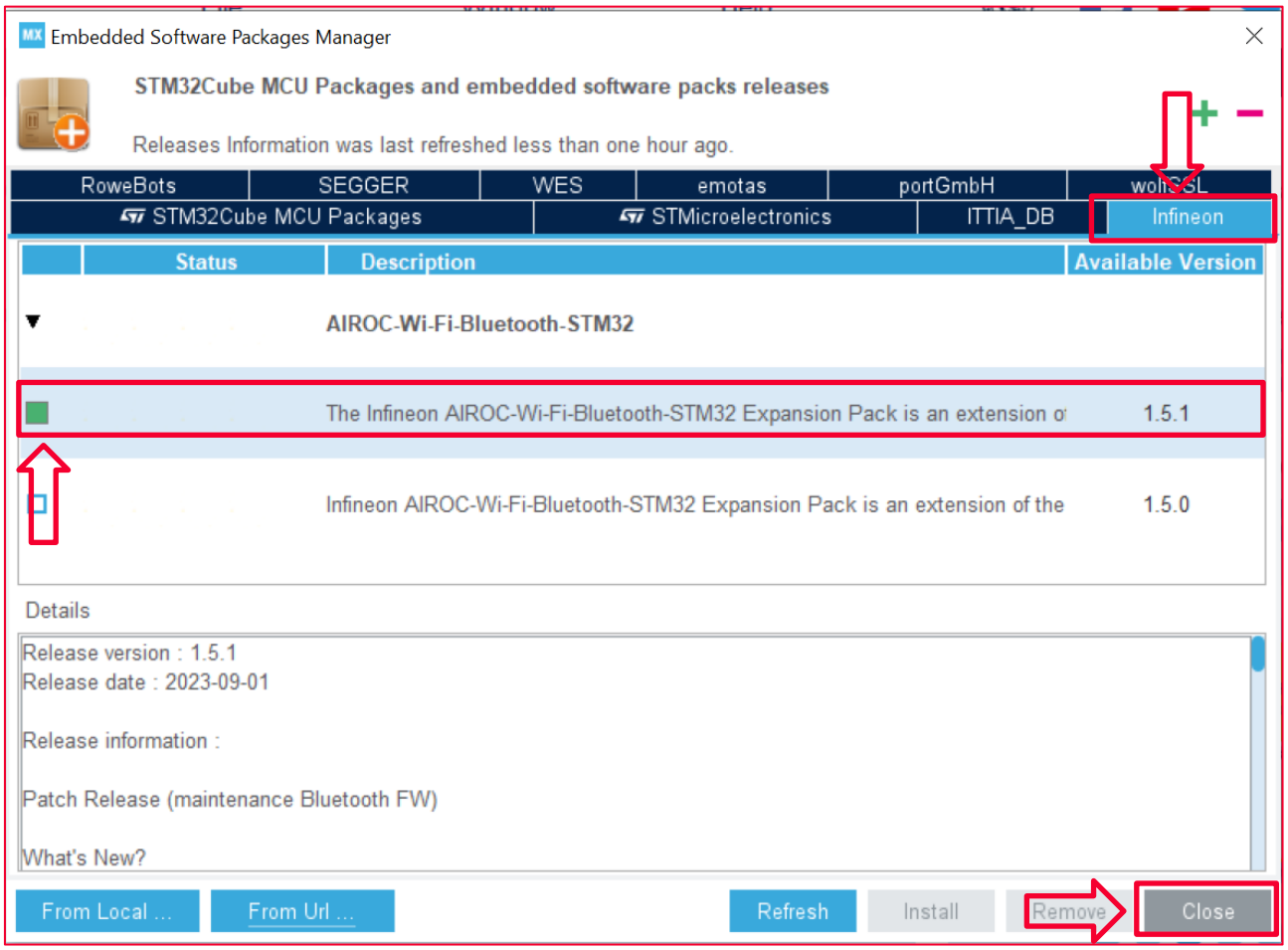
Figure 3 shows the STM32CubeMX pack installation completion options.

Figure 3: STM32CubeMX Install Pack



5. The tool shows an **Infineon** tab showing the installed Expansion Pack (**Figure 4**). Click **Close**.

Figure 4: STM32CubeMX Check Installed Pack



If you want to remove unwanted packs, click on the left green checkbox of the pack which you want to delete and click **Remove** as marked with the arrow in **Figure 4** and click **Close**.

3 Hardware Setup

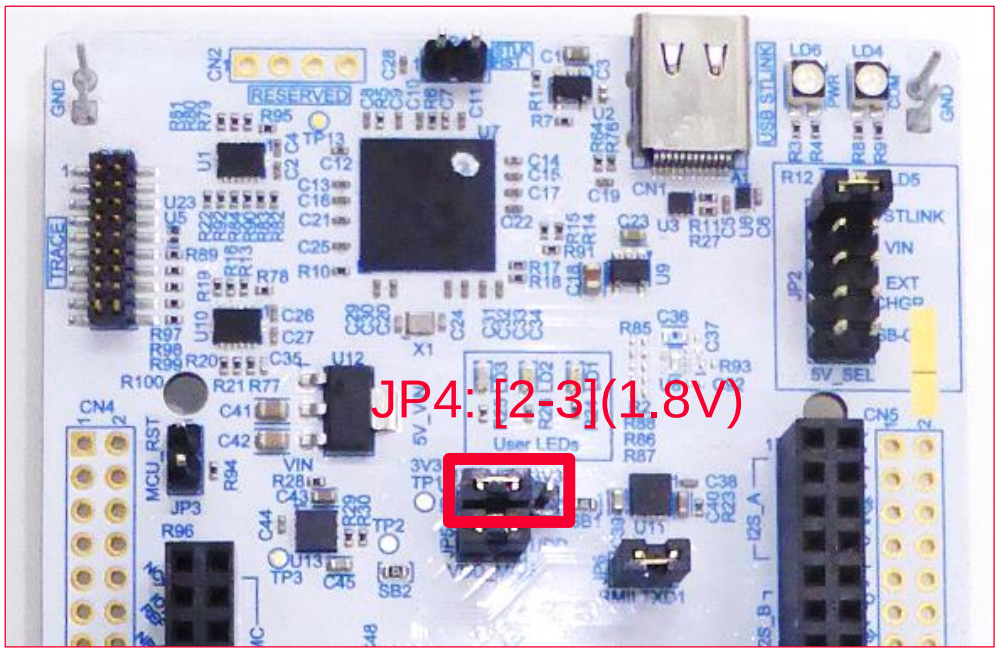
3.1 Using NUCLEO-H563ZI Board

The NUCLEO-H563ZI board setup requires three discrete boards to enable the STM32H5xx board to host Infineon's CYW43xxx connectivity device. The three boards and links are:

- NUCLEO-H563ZI board
- Murata STM32 NUCLEO-M.2 adapter board
- Murata M.2 module (Type 1LV, 1DX, 1YN, 2AE, 2BC)

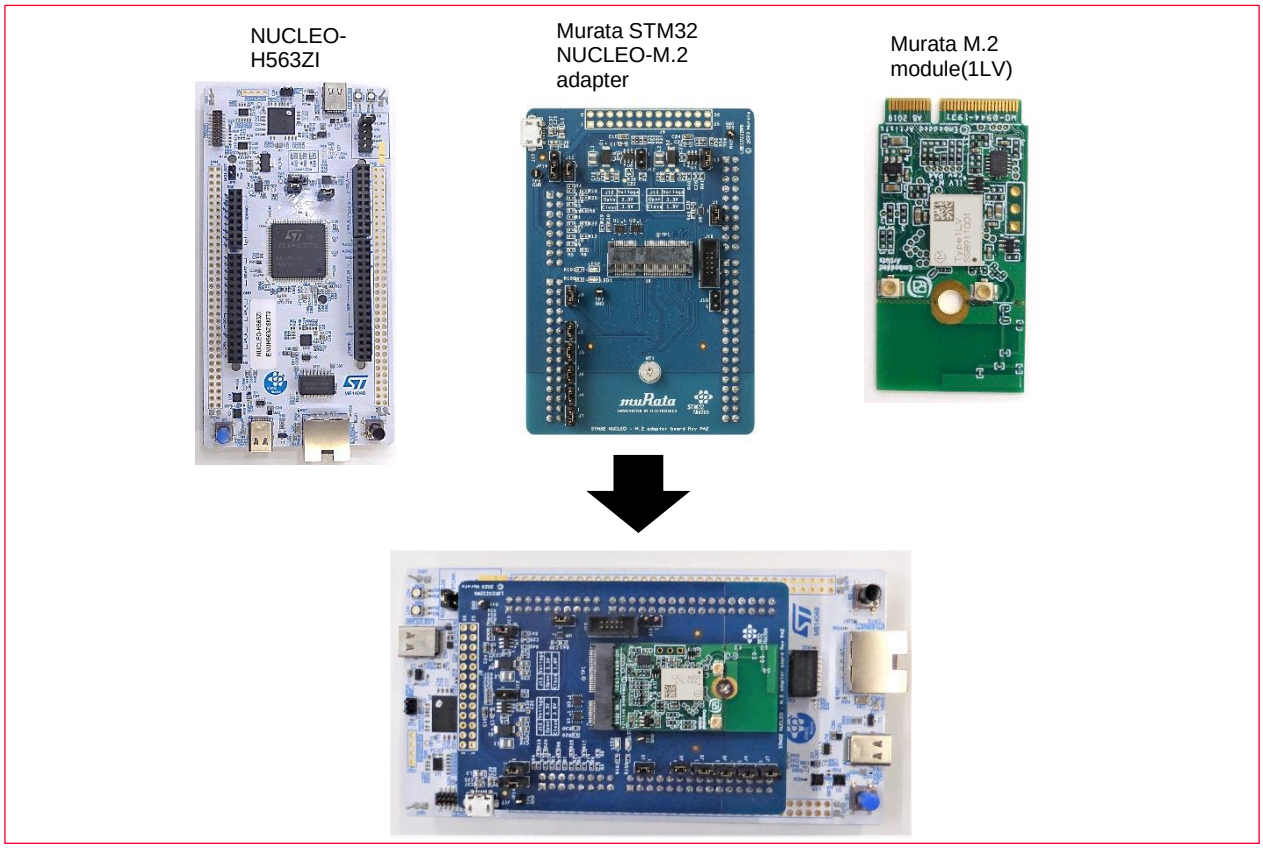
After boards are combined, it is very difficult to change the jumper setting of NUCLEO-H563ZI board. Be sure to check the jumper settings (in particular, JP4) on the NUCLEO-H563ZI board before combining boards as shown in **Figure 5**.

Figure 5:Jumper Settings (in Particular, JP4) on the NUCLEO-H563ZI Board



The combination of the three boards is shown in **Figure 6**.

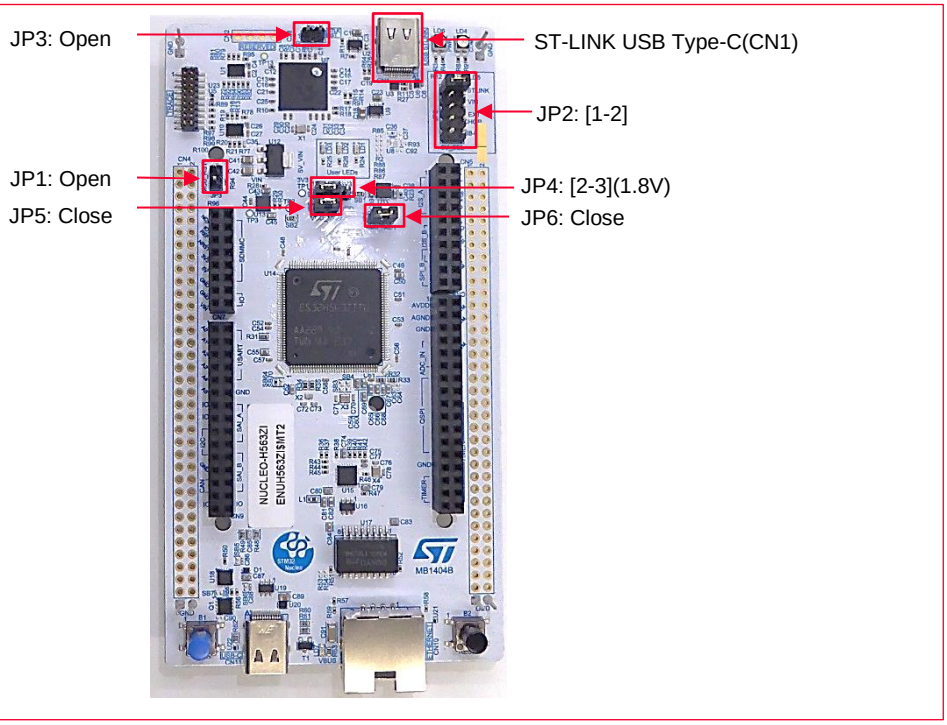
Figure 6: Combine the Three Boards



3.1.1 NUCLEO-H563ZI Board

Figure 7 shows the NUCLEO-H563ZI Board features.

Figure 7: NUCLEO-H563ZI Board Features



The pinout description are listed in **Table 2**.

Table 2: NUCLEO Jumper Configuration

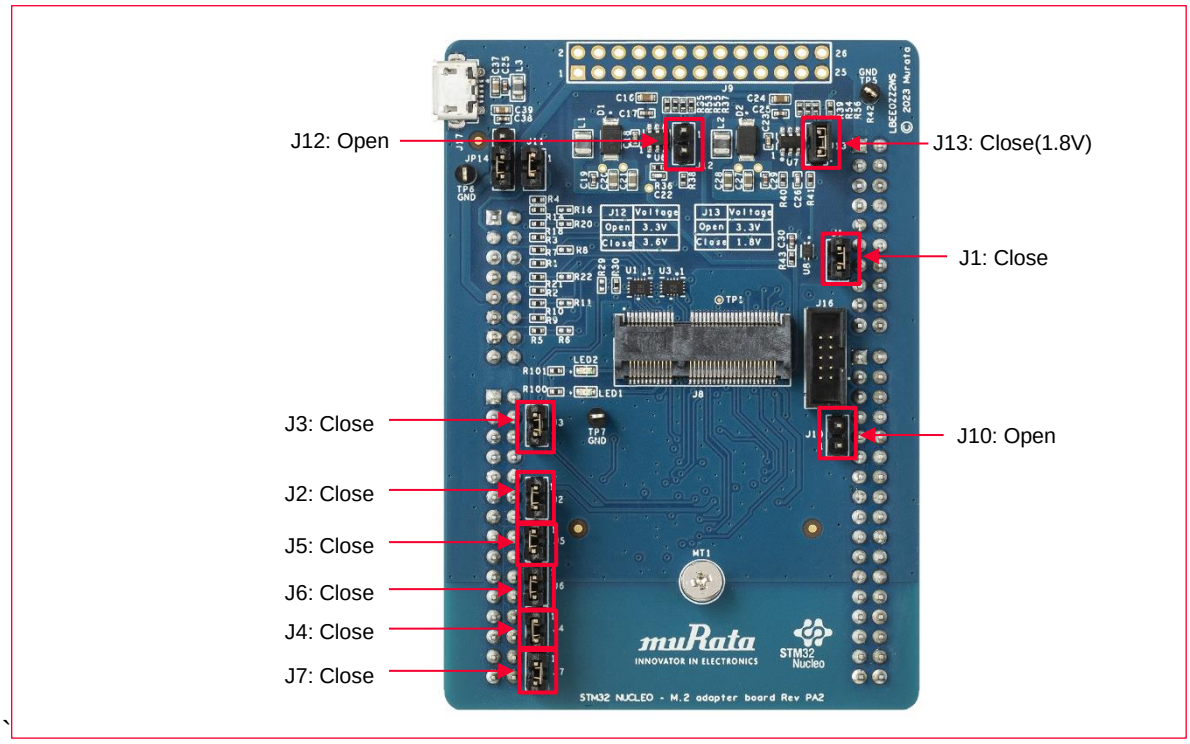
Definition	NUCLEO-H563ZI board	Note
External debug	JP1	• **Open:** The embedded STLINK-V3EC is selected • **Close**: An external debugger connected to the MIPI20 connector (CN5) can be used. The level shifter (U1) is in high impedance (HZ). STLINK-V3EC no longer drives the embedded STM32
Power source selection	JP2	• **[1-2]**: 5V from STLINK-V3EC USB connector with PC • [**3-4**]: 5V source from VIN 7-12V • **[5-6]**: 5V source from 5V_EXT • **[7-8]**: 5V source from USB_CHGR. From STLINK-V3EC USB connector (CN1) with a USB charger. • [**9-10]**: 5V source from USB user connector. • **Open**: NO 5V power source, configuration when external 3V3 is used.
STLK_RST	JP3	• **Open**: STLINK-V3EC is active. • **Close**: STLINK-V3EC is in Reset mode.
VDD voltage selection	JP4	• **[1-2]**: VDD_MCU supplied with 3V3_VDD. • **[2-3]**: VDD_MCU supplied with 1V8_VDD.
IDD measurement	JP5	• **Open**: VDD_MCU is not connected to VDD (the MCU is not power supplied). • **Close**: VDD_MCU is connected to VDD.
Ethernet transmit data1	JP6	• **Open**: Configuration when using ST Zio or ST morpho connector

Definition	NUCLEO-H563ZI board	Note
		• **Close**: using Ethernet

3.1.2 Murata STM32 NUCLEO-M.2 Adapter Board

The requirement for preparation on Murata STM32 NUCLEO-M.2 adapter board is shown in **Figure 8**.

Figure 8: Murata STM32 NUCLEO-M.2 Adapter Board Features



Switch between 1.8 V and 3.3 V depending on whether the jumper of the J13 is set.

- Set the jumper to 1.8V power (Type 1LV, Type 1DX, Type 1YN, Type 2AE, and Type 2BC).
- Do not set the jumper to 3.3V power (Type 1DX, Type 1YN, Type 2AE, and Type 2BC).

Use Pinout description as described in **Table 3**:

Table 3: Adapter Pinout Description

Connection	Operation	Murata STM32 NUCLEO-M.2 Adapter Board	Note
VDDIO Power Selection	VCC	J13	• **Open**: 3.3V • **Close**: 1.8V
VBAT Power Selection	VCC	J12	• **Open**: 3.3V • **Close**: 3.6V
SPI_A_REQ		J1	
WL_REG_ON	Wi-Fi	J2	**Enables/Disables WLAN core**: Active High
WL_HOST_WAKE	Wi-Fi	J4	**WLAN Host Wake**: Active Low (OOB IRQ)
WL_DEV_WAKE	Wi-Fi	J6	
BT_REG_ON	Bluetooth	J3	**Enables/Disables Bluetooth core**: Active High

Connection	Operation	Murata STM32 NUCLEO-M.2 Adapter Board	Note
BT_HOST_WAKE	Bluetooth	J5	
BT_DEV_WAKE	Bluetooth	J7	
BT disable	Bluetooth	J10	• **Open**: BT Enable • **Close**: BT Disable

4 Example Projects

All the examples demonstrate how an NUCLEO can be used to host CYW43012 (Type1LV) connectivity devices.

The expansion pack implements the demonstrates described in **Table 4**.

Table 4: Example List

Folder Name on the Expansion Pack	Description
ble_hello_sensor ble_hello_sensor_threadx	Start a simple Bluetooth on Peripheral role
ble_wifi_onboarding bre_wifi_onboarding_threadx	1. Start Bluetooth on Peripheral role. 2. Provide Wi-Fi connection information (SSID/password) from the combo device. 3. Connect to the AP with received information.
bt_mfg_tester	Use WMBT tool on a Windows PC
wifi_bt_tester	Start the command console which can use Wi-Fi and Bluetooth
wifi_join_wpa3	Start Wi-Fi connection with WPA3
wifi_mfg_tester wifi_mfg_tester_threadx	Use WL tool on a Windows PC
wifi_netxduo	Start Wi-Fi connection and UDP server on Azure RTOS NetX/NetXDuo stack usage
wifi_scan	Start Wi-Fi scan and show results

4.1 Wi-Fi Scan

This example:

- Demonstrates how to configure different scan filters provided in the Wi-Fi Connection Manager (WCM) middleware and scan for the available Wi-Fi networks.
- Initializes the Wi-Fi device and starts a Wi-Fi scan without any filter and prints the results on the serial terminal. The example starts a scan every three seconds after the previous scan is completed.
- Demonstrates how NUCLEO-H563ZI can be used to host CYW43xxx connectivity devices.

4.1.1 Example Project Start/Import

You can open the Wi-Fi Scan example by copying the example from the Pack to an appropriate location. Once you have copied the example, you can then open it in STM32CubeMX and export to your IDE using the following steps:

1. Copy the code example from the pack directory to your local directory.
   The default path for installed packs is:

   C:\Users\<USER>\STM32Cube\Repository\Packs

Copy the wifi_scan example from the appropriate directory. For instance, for NUCLEO-H563ZI:

C:\Users\<USER>\STM32Cube\Repository\Packs\Infineon\AIROC-Wi-Fi-Bluetooth-STM32\1.5.1\Projects\NUCLEO-H563ZI\Applications\wifi_scan

2. Paste into your working folder. For example:
   C:\Users\<USER>\STM32Cube\Example
3. Open *wifi_scan.ioc* file in the copied working folder.
   C:\Users\<USER>\STM32Cube\Example\wifi_scan\wifi_scan.ioc
4. Click **OK** to accept.

4.1.2 Change Connectivity Module

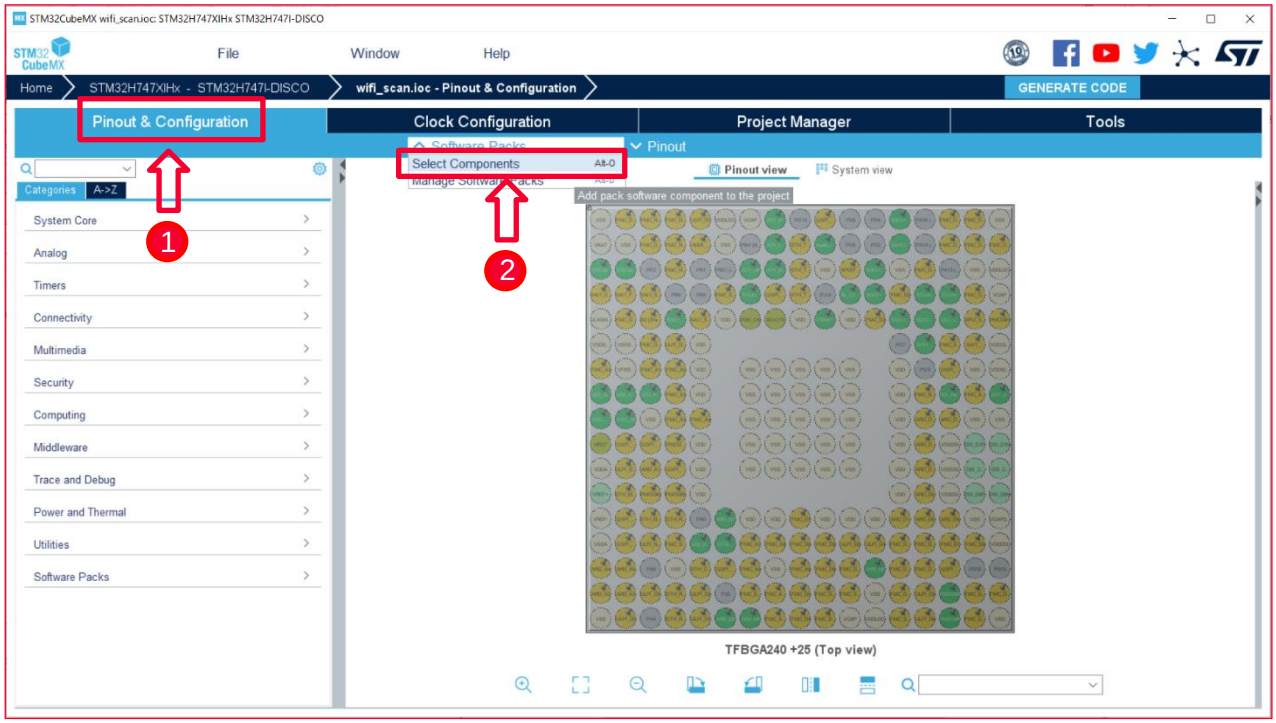
- The default module setting of the expansion pack is Type 1LV. The module setting must be changed when using the other module. Skip this section if you are using the default module (Type 1LV).

Follow these steps to change the connectivity module:

1. Select the **Pinout & Configuration** tab.
2. Select the **Select Components.**

Figure 9 shows the component selection.

Figure 9: STM32CubeMX Select Component

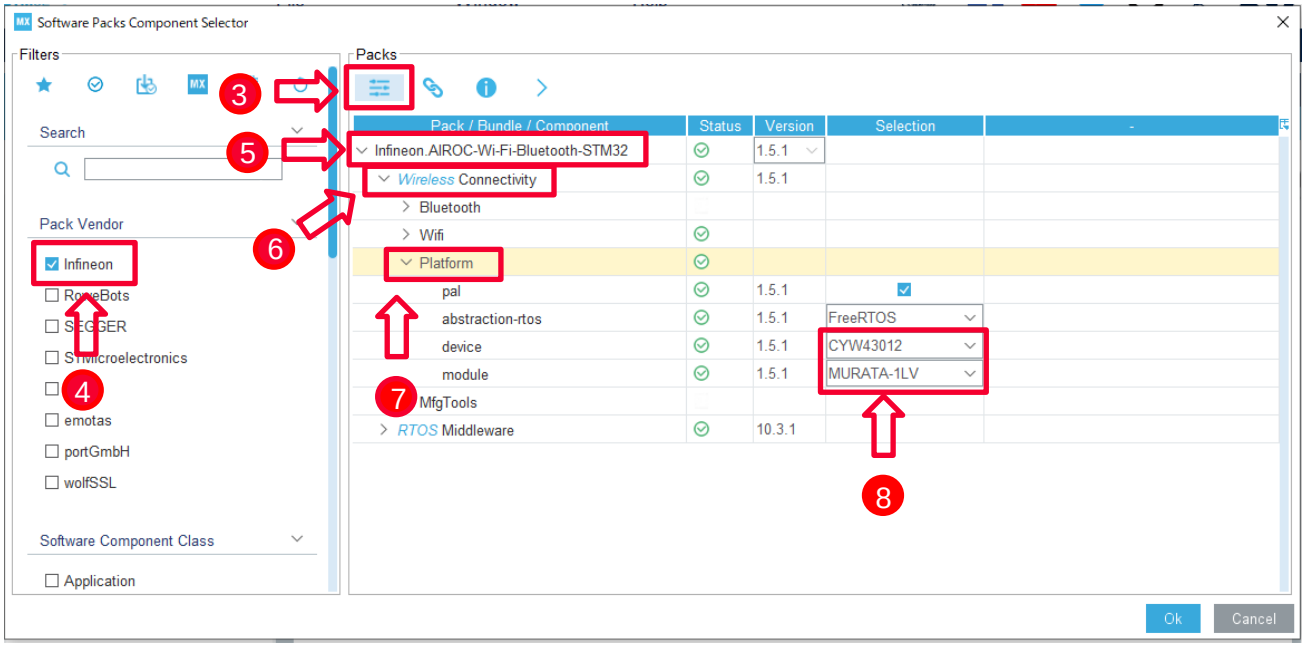


3. Select the **filter panel**.
4. Select **Infineon** under Pack vendor.
5. Open **Infineon AIROC-Wi-Fi-Bluetooth-STM32**.
6. Open **Wireless Connectivity**.

7. Open **Platform**.
8. Change **device** and **module.**

Changing the STM32CubeMX device and module process is marked in **Figure 10**.

Figure 10: STM32CubeMX Change Device and Module



9. Click **Ok** to save change.

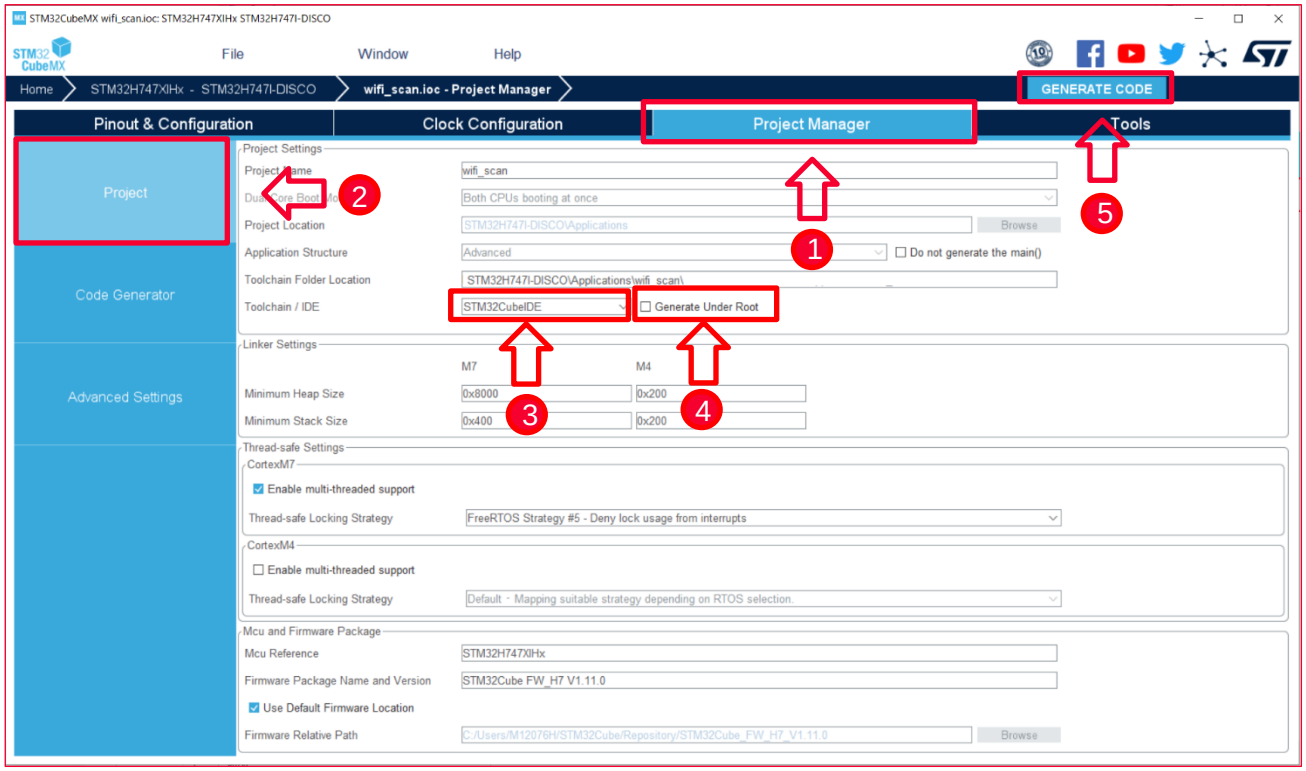
4.1.3 Generate Code

Follow these steps to generate code:

1. Select the **Project Manager tab**.
2. Select **Project**.
3. Select the appropriate option under **Toolchain/IDE**.
4. Unselect the **Generate Under Root** check box.
5. Click **GENERATE CODE**.

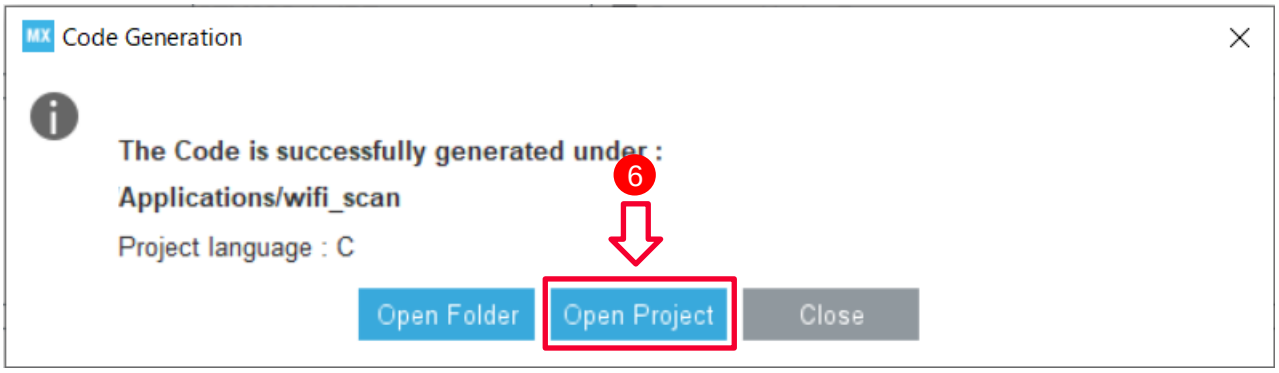
The STM32CubeMX code generation process is marked in **Figure 11**.

Figure 11: STM32CubeMX Generate Code



6. After the code is successfully generated, click **Open Project** to open this example with STM32Cube IDE. **Figure 12** shows the Open Project button.

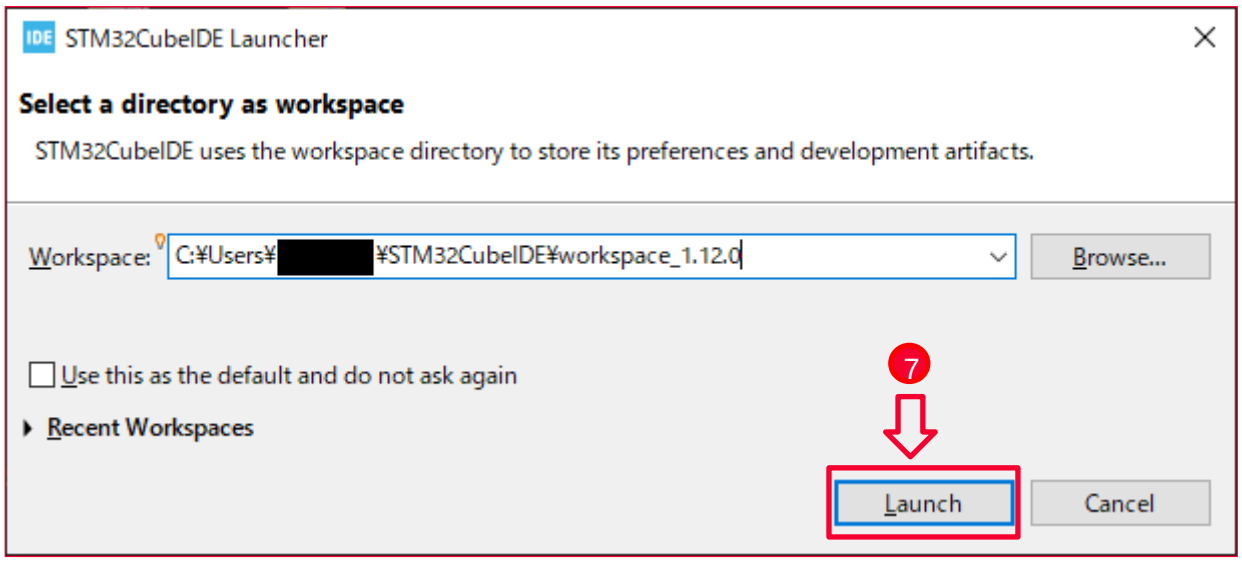
Figure 12: STM32CubeMX Open Project



7. Input the workspace directory of STM32CubeIDE in **Workspace** (Default settings are fine if there are no problems.), click **Launch**.

Figure 13 shows the Launch button.

Figure 13: **STM32CubeMX Select Workspace**



4.1.4 Delete the Include Path Setting

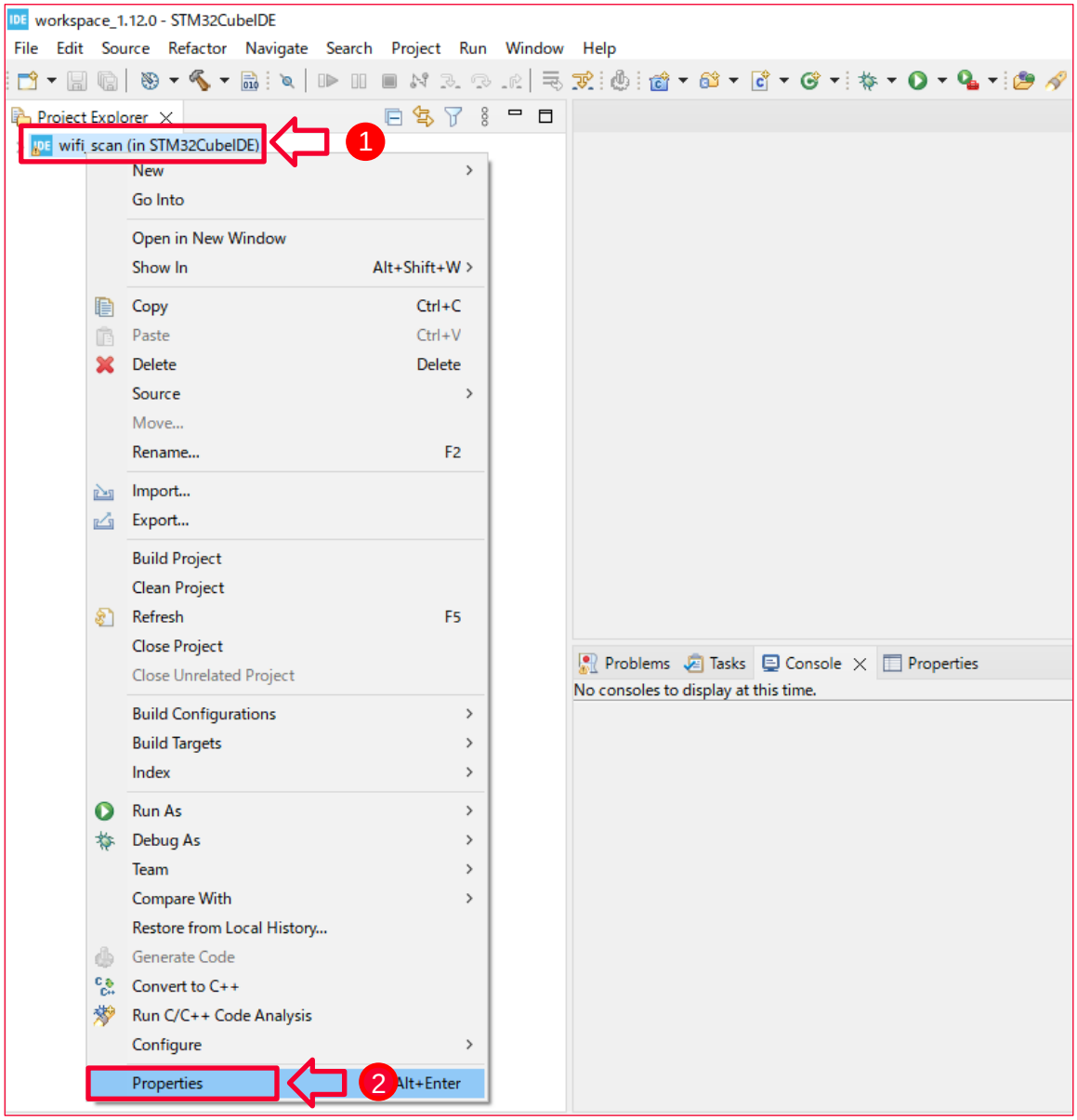
- If you change the module setting at Change Connectivity Module, the include path setting of default module remains in the system. Must delete the include path setting of the default module.
  Skip this section if you use the default module (Type 1LV).

Follow these steps to delete the include path setting:

1. Right-click on the project name.
2. Select **Properties**.

Select Properties steps are marked in **Figure 14.**

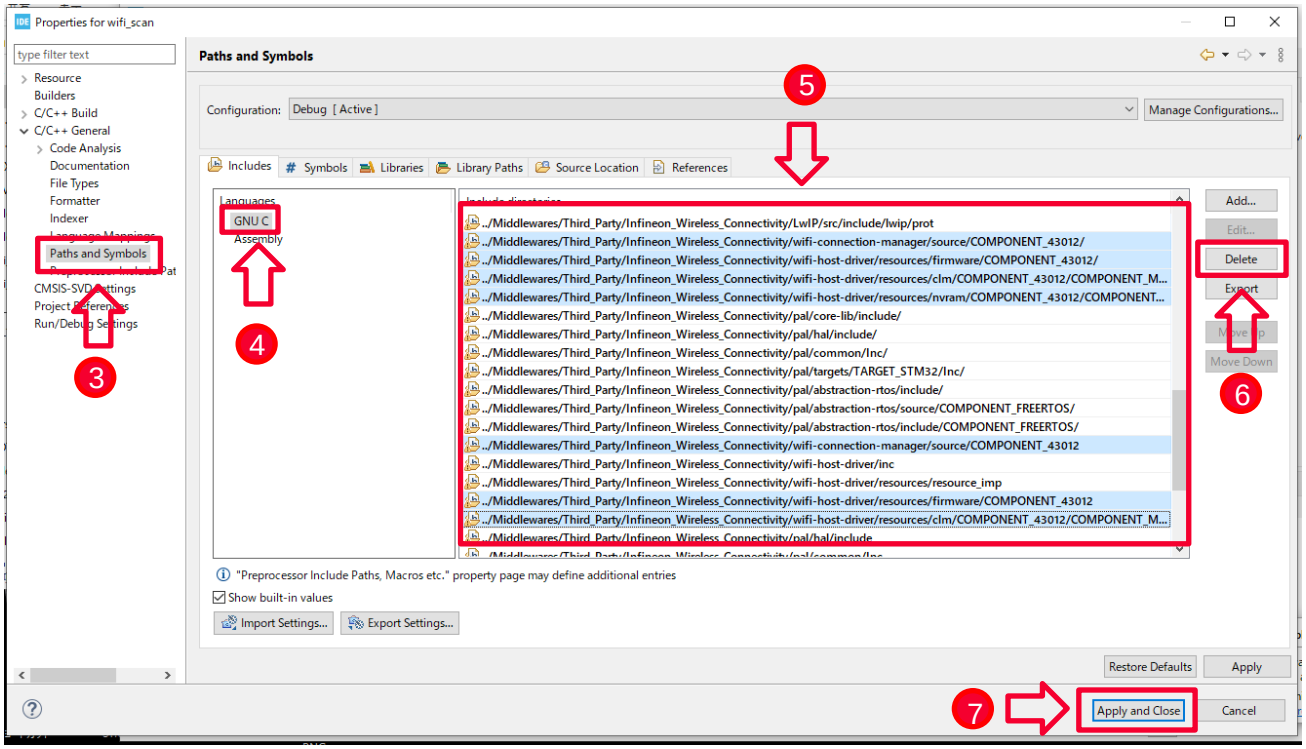
Figure 14: STM32CubeIDE Select Properties



3. Select **Paths and Symbols** under C/C++ General.
4. Select **GNU C** in the Includes tab.
5. Select all include path of default device (**COMPONENT_43012**).
6. Click **Delete** button.
7. Click **Apply and Close** button.

Figure 15 shows the STM32CubeIDE delete include paths.

Figure 15: STM32CubeIDE Delete Include Paths



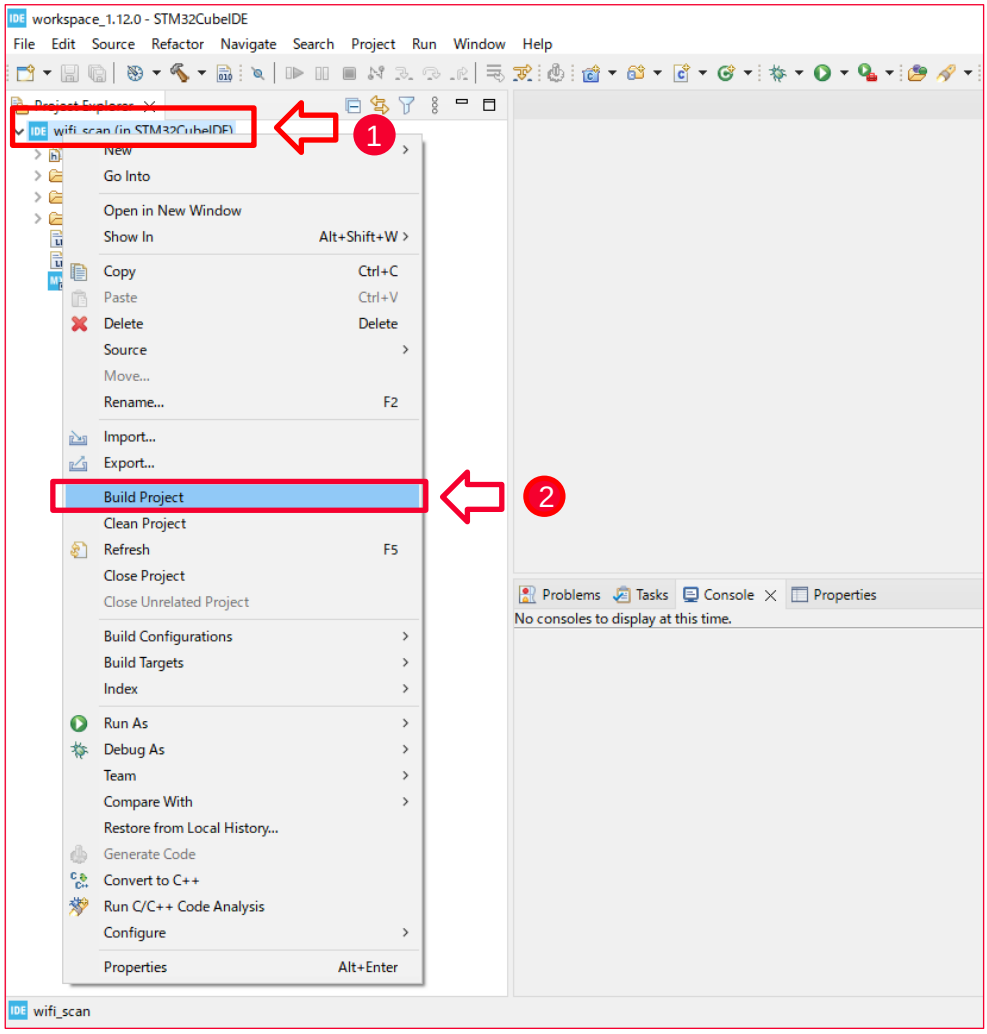
4.1.5 Build the Project

To build the project follow these steps:

1. Right click on the project name.
2. Select **Build Project**.

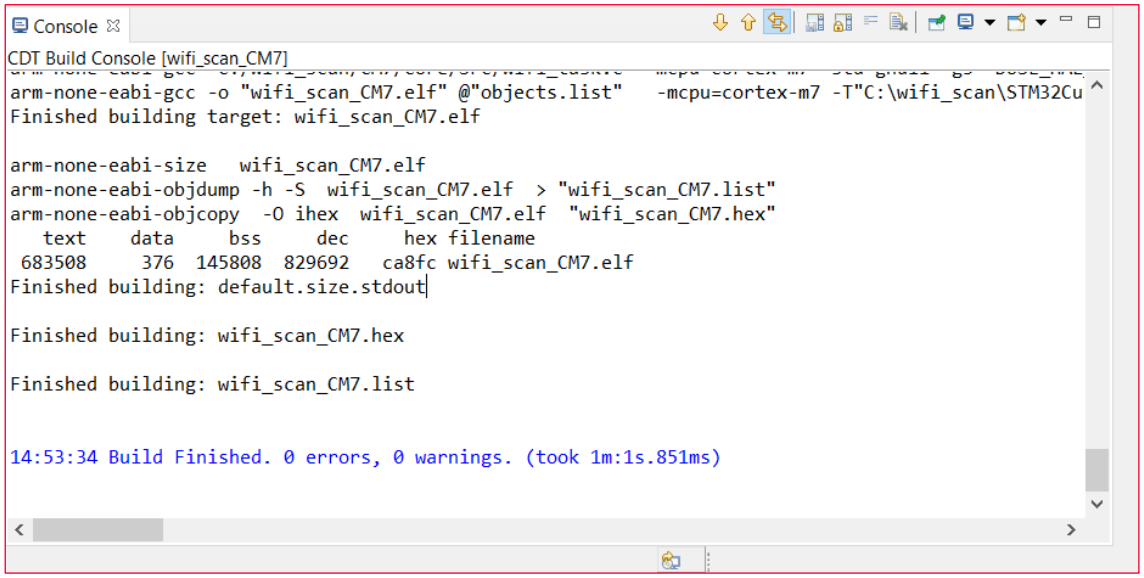
Figure 16 shows the STM32CubeIDE build project options.

Figure 16: STM32CubeIDE Build the Project



Example output from a successful build is shown in **Figure 17**.

Figure 17: STM32CubeIDE Build Successful Results



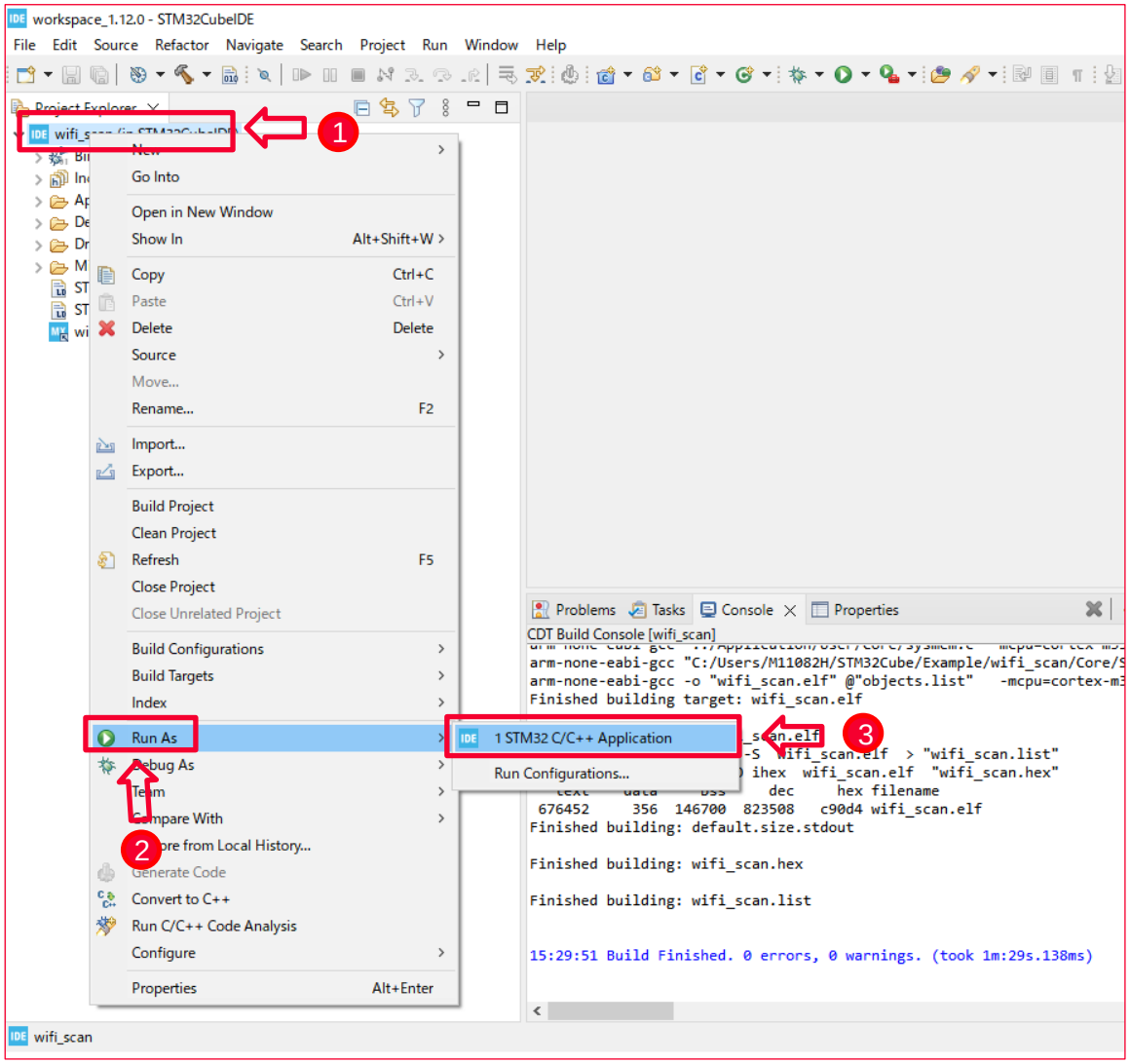
4.1.6 Run Application

To run application follow these steps:

1. Right click on the project name.
2. Select **Run As**.
3. Select **1 STM32 C/C++ Application**.

Figure 18 shows option for STM32CubeIDE run application.

Figure 18: STM32CubeIDE Run Application



4. Click the **OK** button to run application

4.1.7 Serial Terminal Setup

The terminal interface is a virtual COM port which is part of the ST-LINK (CN1) USB connection.
Terminal emulator configuration:

- **Baud Rate**: 115200
- **Data Length**: 8 Bits
- **Stop Bit(s)**: 1
- **Parity**: None
- **Flow control**: None

4.1.8 Application Output

```
WiFi-Scan app (FreeRTOS V10.4.6)

WLAN MAC Address : E8:E8:B7:9F:CF:5A
WLAN Firmware    : wl0: Dec 12 2022 18:42:34 version 13.10.271.293 (9974213 CY)
FWID 01-e2162f9b
WLAN CLM         : API: 18.2 Data: Murata.1LVindoorSTA Compiler: 1.36.1
ClmImport: 1.36.3 Customization: v2 191015 Creation: 2020-01-22 06:19:41
WHD VERSION      : 3.1.0.19622 : master dev-v3.1.0 : GCC 10.3 : 2023-05-25
20:42:44 +0800

--------------------------------------------------------------------------------
--------------------
  #                  SSID                  RSSI    Channel       MAC Address
Security
--------------------------------------------------------------------------------
--------------------
  1   xxxxxxxxxxxxxx                        -78      36      BC:26:C7:43:DF:B5
WPA2_AES_PSK
  2   YYYYYYYYYYYYYYYYY                     -55      36      00:11:32:B1:41:50
OPEN
  3   Zzzzzzzzzzzz                          -49      36      A8:5E:45:64:2F:CC
WPA2_AES_PSK
```

4.2 Bluetooth Hello Sensor

This code example demonstrates the implementation of a simple Bluetooth stack functionality in GAP peripheral role. During initialization, the app registers with LE stack to receive various notifications including bonding complete, connection status change and peer write. The peer device can also write to client configuration descriptor of the notification characteristic.

4.2.1 Example Project Start/Import

You can open the Bluetooth hello sensor example by copying the example from the pack to an appropriate location. Once you have copied the example, you can then open it in STM32CubeMX and export to your IDE using the following steps:

- Copy the code example from the pack directory to your local directory.
  The default path for installed packs is:

  C:\Users\<USER>\STM32Cube\Repository\Packs\

- Copy the ble_hello_sensor example from the appropriate directory. For instance, for NUCLEO-H563ZI:

  C:\Users\<USER>\STM32Cube\Repository\Packs\Infineon\AIROC-Wi-Fi-Bluetooth-STM32\1.5.1\Projects\NUCLEO-H563ZI\Applications\ble_hello_sensor

- Paste into your working folder. For example:

  C:\Users\<USER>\STM32Cube\Example

4. Open *ble_hello_sensor.ioc* file in the copied working folder.

   C:\Users\<USER>\STM32Cube\Example\ble_hello_sensor\ble_hello_sensor.ioc

5. Click **OK** to accept.

4.2.2 Change Connectivity Module to Serial Terminal Setup

The steps to change connectivity module, generate code, delete the include path setting, build the project, run application, and serial terminal setup are same as shown in Wi-Fi scan application example.
So, refer to Wi-Fi Scan.

4.2.3 Application Output

```
[0] Hello Sensor Start
[0] wiced_bt_stack_init()
[503] bt_post_reset_cback()
[505] bt_post_reset_cback(): Change baudrate (3000000) for FW downloading
[512] bt_update_controller_baudrate(): 3000000
[518] bt_baudrate_updated_cback(): Baudrate is updated for FW downloading
[524] bt_update_platform_baudrate(): 3000000
[1437] bt_patch_download_complete_cback(): status = 1
[1441] bt_fw_download_complete_cback(): Reset baudrate to 115200
[1447] bt_update_platform_baudrate(): 115200
[1651] bt_fw_download_complete_cback(): post-reset process is Done
[2086] hello_sensor_management_cback: 16

[2090] wiced_post_stack_init_cback
[2093] hello_sensor_management_cback: 0

[2097] hello_sensor_application_init

[2100] wiced_bt_gatt_register: 0

[2104] wiced_bt_gatt_db_init 0

[2107] hello_sensor_management_cback: 18

[2110] Advertisement State Change: 3

[2114] wiced_bt_start_advertisements 0

Local Bluetooth Address: E8:E8:B7:9F:CF:5B
[2121] wiced_post_stack_init_cback(): BT sleep mode is NOT enabled
[3108] hello_sensor_timeout: 1, ft:0

[4109] hello_sensor_timeout: 2, ft:0

[5109] hello_sensor_timeout: 3, ft:0

[6110] hello_sensor_timeout: 4, ft:0
```

The STM32 hardware starts to send the advertisement signal.
Turn on the Bluetooth on your device (Android or iPhone, etc), and execute scanning for Bluetooth by BLE application.
You can find the BLE device named by “hello”.

4.3 Tester - Wi-Fi Bluetooth Console

This application integrates the command console library including the Wi-Fi iPerf and Bluetooth Low Energy functionality. You can use this application to characterize the Wi-Fi/Bluetooth LE functionality and performance.

This example demonstrates how NUCLEO-H563ZI can be used to host CYW43xxx connectivity devices.

4.3.1 Example Project Start/Import

You can open the example by copying the example from the Pack to an appropriate location. Once you have copied the example, you can then open it in STM32CubeMX and export to your IDE using the following steps:

1. Copy the code example from the pack directory to your local directory.
   - The default path for installed packs is:

     C:\Users\<USER>\STM32Cube\Repository\Packs\
   - Copy the wifi_bt_tester example from the appropriate directory. For instance, for NUCLEO-H563ZI:

     C:\Users\<USER>\STM32Cube\Repository\Packs\Infineon\AIROC-Wi-Fi-Bluetooth-STM32\1.5.1\Projects\NUCLEO-H563ZI\Applications\wifi_bt_tester
   - Paste into your working folder. For example:

     C:\Users\<USER>\STM32Cube\Example
2. Open *wifi_bt_tester.ioc* file in the copied working folder.

   C:\Users\<USER>\STM32Cube\Example\wifi_bt_tester\wifi_bt_tester.ioc
3. Click **OK** to accept.

4.3.2 Change default Wi-Fi Connection Setting

When “wifi_bt_tester” application is started, it starts to connect to the access point of default setting. Until the connection successes or fails, you cannot operate the command console.
So, it is recommended to change the SSID/PASSWORD and retry count on failure.

1. Open wifi_bt_tester\Core\Src\console_task.c
2. Change the SSID/PASSWORD and retry count.

```
/* Private macro -------------------------------------------------------------
*/
#ifndef WIFI_SSID
#define WIFI_SSID                       "WIFI_SSID"
#endif
#ifndef WIFI_PASSWORD
#define WIFI_PASSWORD                   "WIFI_PASSWORD"
#endif
#define WIFI_BAND                       CY_WCM_WIFI_BAND_ANY
#define CMD_CONSOLE_MAX_WIFI_RETRY_COUNT 15
#define IP_STR_LEN                      16
```

4.3.3 Change Connectivity Module to Serial Terminal Setup

Steps to change connectivity module, generate code, delete the include path setting, build the project, run Application, and serial terminal setup are same as Wi-Fi scan application.
So, refer to Wi-Fi Scan.

However, in the delete the include path setting procedure, delete the **GNU C++** include path setting as well as **GNU C**.

4.3.4 Application Output

```
Command console application

WLAN MAC Address : E8:E8:B7:9F:CF:5A
WLAN Firmware    : wl0: Dec 12 2022 18:42:34 version 13.10.271.293 (9974213 CY)
FWID 01-e2162f9b
WLAN CLM         : API: 18.2 Data: Murata.1LVindoorSTA Compiler: 1.36.1
ClmImport: 1.36.3 Customization: v2 191015 Creation: 2020-01-22 06:19:41
WHD VERSION      : 3.1.0.19622 : master dev-v3.1.0 : GCC 10.3 : 2023-05-25
20:42:44 +0800
WCM Initialized
Successfully joined wifi network '************* , result = 0'
IP Address 192.168.0.107 assigned
executing command_console_add_remove_command
> Wi-Fi module initialized...
> scan
#### Scan Results ####

SSID                 Security Type  RSSI(dBm)  Channel BSSID
XXXXXXXXXXXX         open           -42        1       24:4B:FE:D1:40:18
YYYYYYYYYYYYYY       wpa2           -60        11      D8:9B:3B:3E:06:D0
ZZZZZZZZZZZZZZZZZZZZ wpa2_aes       -47         11      C4:41:1E:9D:EB:57
#### Scan Results END ####
```

4.3.5 Command List

This section provides that the list and the description of the commands which can be entered in command console.

Also, you can confirm this information by entering the “help” command in command console.

4.3.5.1 Command for Wi-Fi

- join <ssid> <open|wpa_aes|wpa_tkip|wpa2|wpa2_aes|wpa2_tkip|wpa2_fbt|wpa3|wpa3_wpa2> [password] [channel]
  - When any parameter has spaces, use quotes. E.g. "my ssid" "my wpa2 key"
  - Join an AP.

    (This command is deprecated, and it will be removed in the future. Please use wifi_join command)
- leave
  - Leave the connected AP.
    (This command is deprecated, and it will be removed in the future. Please use wifi_leave command)

- scan
  - Scan all the Wi-Fi AP in the vicinity.
    (This command is deprecated, and it will be removed in the future. Please use wifi_scan command)
- ping <IP address> [timeout(ms)]
  - ping to an IP address.
    (This command is deprecated, and it will be removed in the future. Please use wifi_ping command)
- get_rssi
  - Get the received signal strength of the AP (client mode only).
    (This command is deprecated, and it will be removed in the future. Please use wifi_get_rssi command)
- wifi_join <ssid> <open|wpa_aes|wpa_tkip|wpa2|wpa2_aes|wpa2_tkip|wpa2_fbt|wpa3|wpa3_wpa2> [password] [channel]
  - When any parameter has spaces, use quotes. E.g. "my ssid" "my wpa2 key"
  - Join an AP.
- wifi_leave

  - Leave the connected AP.
- wifi_scan
  - Scan all the Wi-Fi AP in the vicinity.
- wifi_ping <IP address> [timeout(ms)]
  - ping to an IP address
- wifi_get_rssi
  - Get the received signal strength of the AP (client mode only).

4.3.5.2 Command for Bluetooth

- bt_on
  - Turn On Bluetooth
- bt_off
  - Turn Off Bluetooth
- bt_get_device_address
  - Get Bluetooth Device Address
- ble_start_adv
  - Start BLE Advertisement.
- ble_stop_adv
  - Stop BLE Advertisement.
- ble_start_scan
  - Start BLE Scan

- ble_stop_scan
  - Stop BLE Scan
- ble_coc_init
  - Initializes LE COC with PSM 19 and MTU 100
- ble_coc_adv
  - Start LE COC advertisements.
- ble_coc_scan_connect
  - Scan and Connect to a LE COC server.
- ble_coc_disconnect
  - Disconnect LE COC
- ble_coc_send_start
  - Start Sending LE COC data.
- ble_coc_send_stop
  - Stop Sending LE COC data.
- ble_get_throughput
  - Get LE COC Throughput

4.3.5.3 Command for iPerf

- iperf -s
  - On the peer iPerf device (host PC), start iPerf as a TCP client to send the TCP data.
- iperf -c <server_ip_addr> -t <time in sec>
  - On the peer iPerf device (host PC), start iPerf as a TCP server.
  - Sample command: iperf -c 192.168.0.100 -t 60
- iperf -s -u
  - On the peer iPerf device (host PC), start iPerf as a UDP client to send the UDP data.
- iperf -c <server_ip_addr> -t <time in sec> -u -b <band width>
  - On the peer iPerf device (host PC), start iPerf as a UDP server.
  - Sample command: iperf -c 192.168.0.100 -t 60 -u -b 50M

4.3.6 iPerf Measurement

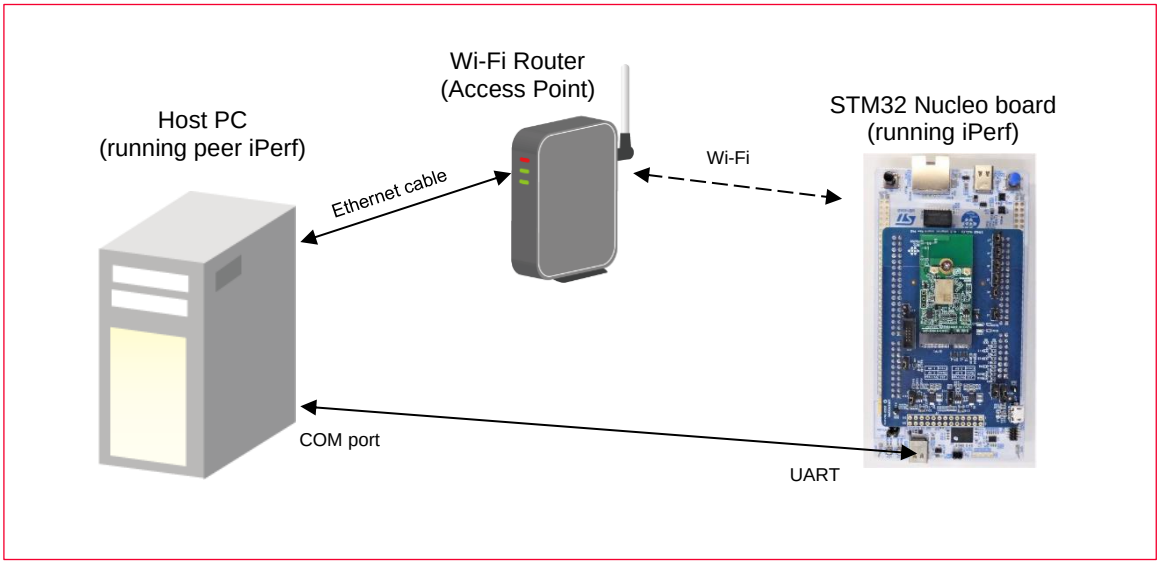
iPerf commands are used for measuring the Wi-Fi performance/throughput.

The iPerf sends TCP/UDP data between two peer devices to compute the Wi-Fi performance/throughput.

4.3.6.1 iPerf Setup

Figure 19 shows the exact setup that should be used for measuring the Wi-Fi performance/throughput of a STM32 device using iPerf.

Figure 19: iPerf Setup



4.3.6.2 iPerf Command

iPerf command example:

- TCP: iperf -c <ip> -i1 -t30
- TCP: iperf -s -i1
- UDP: iperf -c <ip> -i1 -t30 -u -b 50M
- UDP: iperf -s -i1 -u

4.4 Other Example in the Pack

With other example in the pack, please check this document (Infineon AIROC Wi-Fi/Bluetooth STM32 Expansion Pack User Guide).

Revision History

Revision	Date	Author	Change Description
1.0	Sep 11, 2023	K.Gouda	Initial Release