User Guide



# Wi-Fi<sup>®</sup> + Bluetooth<sup>®</sup> (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.
i	Info Note Intended for informational purposes. Users should review.
F.	Menu Reference Indicates menu navigation instructions. Example: Insert-Tables-Quick Tables-Save Selection to Gallery
□7	External Hyperlink This symbol indicates a hyperlink to an external document or website. Example: Embedded Artists AB ご Click on the text to open the external link.
Ľ	Internal Hyperlink         This symbol indicates a hyperlink within the document.         Example:       Prerequisite □ <sup>ℓ</sup> Click on the text to open the link.
Console input/output or code snippet	Console I/O or Code Snippet This text <i>Style</i> denotes console input/output or a code snippet.
<pre># Console I/O comment // Code snippet comment</pre>	<ul> <li>Console I/O or Code Snippet Comment</li> <li>This text Style denotes a console input/output or code snippet comment.</li> <li>Console I/O comment (preceded by "#") is for informational purposes only and does not denote actual console input/output.</li> <li>Code Snippet comment (preceded by "//") may exist in the original code.</li> </ul>



# 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) ⊑7
- Type 1YN (CYW43439) **□**
- Type 1LV (CYW43012) ⊑7
- Type 2AE (CYW4373E) ⊑
- Type 2BC (CYW4373) ⊑7

# 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  $\Box$ <sup>2</sup>.

# 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 **Γ**.

Figure 1 shows the STM32CubeMX importing pack option.



#### Figure 1: STM32CubeMX Importing the Pack

STM32CubeMX Untitled					- 🗆 ×
STM32	File	Window	Help		🐵 🖪 🗖 🎽 😽 🖅
Home >					
Existing Projects			New Project	Man	age software installations
Recent Opened Pr	rojects		I need to :	с	Check for STM32CubeMX and embedded software packages updates
Last modified date :	15/02/2023 17:43:33	MIX	Start My project from MCU		
wifi_bt_tester.ioc Last modified date :	) 17/02/2023 9:28:53	MX	ACCESS TO MCU SELECTOR	. Ir	Install or remove embedded software packages
wifi_scan.ioc Last modified date :	17/02/2023 9:27:20	MX	ACCESS TO BOARD SELECTOR		
Other Projects		Ca.	Start My project from Example ACCESS TO EXAMPLE SELECTOR		Download CAD resources Including, on demand, your own labels
				4	
					Home > MX Project > Tools > CAD
					About STM32
					Change of the construction

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

Embedded Software I	Packages Manager	×
STM32Cut Releases In	be MCU Packages and embedded software packs releases	+ -
RoweBots	SECCED WES emotor part C	X wolfSSL
Description	Look In 📙 1.5.1 🗸 🔽 🔂 🔓	Available Version
► STM32C0	Infineon.AIROC-Wi-Fi-Bluetooth-STM32.1.5.1.pack	
► STM32F0	Î	
STM32F1		
Details	File Name     Infineon.AIROC-Wi-Fi-Bluetooth-STM32.1.5.1.pack       Files of Types     STM32Cube Packages File (*.zip, *.pack)	Cancel
From Local	From Url Refresh Insta	all Remove Close



4. Accept the license agreement and select Finish.

Figure 3 shows the STM32CubeMX pack installation completion options.

#### Figure 3: STM32CubeMX Install Pack

🚾 Embedded So	🚾 Embedded Software Packages Manager 🛛 🕹				
STN	132Cube MCU Packages and embedded software packs releases	+ -			
Rele	Licensing Agreement X	wolfSSL			
in S	Infineon AIROC-Wi-Fi-Bluetooth-STM32 1.5.1 License Agreement	Infineon			
Desc	Please read and accept the following agreement carefully to finish the installation:	ble Version			
STM32C	CYPRESS (AN INFINEON COMPANY) END USER LICENSE AGREEMENT				
<ul><li>► STM32F(</li><li>► STM32F(</li></ul>	PLEASE READ THIS END USER LICENSE AGREEMENT ("Agreement") CAREFULLY BEFORE DOWNLOADING, INSTALLING, COPYING, OR USING THIS SOFTWARE AND ACCOMPANYING DOCUMENTATION. BY DOWNLOADING, INSTALLING, COPYING OR USING THE SOFTWARE, YOU ARE AGREEING TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, PROMPTLY RETURN AND DO NOT USE THE SOFTWARE. IF YOU HAVE PURCHASED THIS LICENSE TO THE SOFTWARE, YOUR RIGHT TO RETURN THE SOFTWARE EXPIRES 30 DAYS AFTER YOUR PURCHASE AND APPLIES ONLY TO THE ORIGINAL PURCHASER.				
Details	1. Definitions.				
	I have read, and I agree to the terms of this license agreement I do not accept the terms of this license agreement Finish Cancel				
From Local	From Url Refresh Install Remove	Close			

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



#### Figure 4: STM32CubeMX Check Installed Pack

🚾 Embed	ded Software Packag	es Manager			Ann M. Sand I	×
	STM32Cube MCU Packages and embedded software packs releases					
Boy	NoRoto			amotan	nortCmhH	uno Sel
Rov	STM32Cube M	CU Packages		STMicroelectronics		DB Infineon
	Status	Description				Available Version
•		AIROC-Wi-Fi-Blu	ietooth-STM32			
	1 - 1 - 1	The Infineon AIRC	C-Wi-Fi-Bluetoo	th-STM32 Expansion	Pack is an extensio	on or 1.5.1
Ê		Infineon AIROC-V	/i-Fi-Bluetooth-S	TM32 Expansion Pac	k is an extension of	the 1.5.0
Details						
Release v Release d	ersion : 1.5.1 late : 2023-09-01					
Release ii	nformation :					
Patch Rel	ease (maintenance	Bluetooth FW)				
What's Ne	ew?					
From Lo	ocal From	n Url		Refresh	Install	Remove Close



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) ロ



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 Config	guration
-------------------------------	----------

Definition	NUCLEO- H563ZI board	Note
External debug	JP1	<ul> <li>Open: The embedded STLINK-V3EC is selected</li> <li>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</li> </ul>
Power source selection	JP2	<ul> <li>[1-2]: 5V from STLINK-V3EC USB connector with PC</li> <li>[3-4]: 5V source from VIN 7-12V</li> <li>[5-6]: 5V source from 5V_EXT</li> <li>[7-8]: 5V source from USB_CHGR. From STLINK-V3EC USB connector (CN1) with a USB charger.</li> <li>[9-10]: 5V source from USB user connector.</li> <li>Open: NO 5V power source, configuration when external 3V3 is used.</li> </ul>
STLK_RST	JP3	<ul> <li>Open: STLINK-V3EC is active.</li> <li>Close: STLINK-V3EC is in Reset mode.</li> </ul>
VDD voltage selection	JP4	<ul> <li>[1-2]: VDD_MCU supplied with 3V3_VDD.</li> <li>[2-3]: VDD_MCU supplied with 1V8_VDD.</li> </ul>
IDD measurement	JP5	<ul> <li>Open: VDD_MCU is not connected to VDD (the MCU is not power supplied).</li> <li>Close: VDD_MCU is connected to VDD.</li> </ul>
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:

Connection	Operation	Murata STM32 NUCLEO- M.2 Adapter Board	Note
VDDIO Power Selection	VCC	J13	<ul> <li>Open: 3.3V</li> <li>Close: 1.8V</li> </ul>
VBAT Power Selection	VCC	J12	<ul> <li>Open: 3.3V</li> <li>Close: 3.6V</li> </ul>
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

#### **Table 3: Adapter Pinout Description**



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	Start a simple Bluetooth on Peripheral role
ble_hello_sensor_threadx	
ble_wifi_onboarding	<ol> <li>Start Bluetooth on Peripheral role.</li> </ol>
bre_wifi_onboarding_threadx	<ol> <li>Provide Wi-Fi connection information (SSID/password) from the combo device.</li> </ol>
	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	Use WL tool on a Windows PC
wifi_mfg_tester_threadx	
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

rs	Packs				
· · · 🗄 · 🔤 🖪 😅					
earch	Pack / Bundle / Component	Status	Version	Selection	
_ <b>5 </b>	✓ Infineon.AIROC-Wi-Fi-Bluetooth-STM32	$\odot$	1.5.1 🗸		
α	✓ Wireless Connectivity	$\odot$	1.5.1		
	> Bluetooth				
ack Vendor	> Wifi	$\odot$			
Infineon	✓ Platform	$\odot$			
RomeBots	pal	$\odot$	1.5.1	✓	
	abstraction-rtos	$\odot$	1.5.1	FreeRTOS	~
	device	$\odot$	1.5.1	CYW43012	~
	module	$\odot$	1.5.1	MURATA-1LV	~
4	7 MfgTools				
_ emotas	> RTOS Middleware	$\odot$	10.3.1	11	
] portGmbH				<b>U</b>	
wolfSSL					
oftware Component Class					
Application					

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.

STM32CubeMX wifi_so	an.ioc: STM32H747XIHx STM32H7	47I-DISCO				- 🗆 ×
STM32 CubeMX	File	Window	Help		<b>(19)</b>	🖪 🖸 🎽 🔆 🖅
Home 🔰 STM32	H747XIHx - STM32H747I-D	IISCO 🔰 wifi_scan.ioe	c - Project Manager >		GEI	NERATE CODE
Pino	ut & Configuration	C	lock Configuration	Project Manage	er	Tools
Proj	ect Project Project Project Project	Settings ame re Boot do 2 Location	wifi_scan Both CPUs booting at once STM32H747I-DISCO/Applications	Û		1 5
Code Ge	Applicat Toolcha Toolcha	ion Structure in Folder Location in / IDE	Advanced STM32H747I-DISCO\Application STM32CubeIDE	stwifi scant	○ Do not generate the main()	
Advanced	Settings Minimur	n Heap Size n Stack Size	M7 0x8000 0x400	M4 U 0x200 4 0x200 4		
	Cortexi Cortexi En: Thread	ate Settings A7	FreeRTOS Strategy #5 - Deny Ic	ck usage from interrupts	v]	
	Cortext	<sup>A4</sup> able multi-threaded support safe Locking Strategy	Default - Mapping suitable strat	egy depending on RTOS selection.	~	
	/Mcu and Mcu Re Firmwar	I Firmware Package ference e Package Name and Version	STM32H747XIHx STM32Cube FW_H7 V1.11.0			
	<b>☑</b> Use Firmwar	Default Firmware Location e Relative Path	C:/Users/M12076H/STM32Cube/	Repository/STM32Cube_FW_H7_V1.11.0	Browse	

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

K Code Generation	×
The Code is successfully generated under:     Applications/wifi_scan     Project language : C     Open Folder Open Project Close	

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

IDE STM32CubelDE Launcher	$\times$
Select a directory as workspace STM32CubeIDE uses the workspace directory to store its preferences and development artifacts.	
Workspace: C:¥Users¥ STM32CubelDE¥workspace_1.12.0 $\vee$ Browse	·
Use this as the default and do not ask again     7       Recent Workspaces     1       Launch     Cance	:1

## 4.1.4 Delete the Include Path Setting

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

IDE worksp	ace_1	I.12.0 - STM32CubelDE														
File Edit	So	urce Refactor Navigate Sear	ch Project Run	Win	dow	Help								-		-
1 🗖 🗕 🖫	C	🛞 🔻 🍆 🖛 🔝 🕨 🛛		_R	₹	V O	<u>í</u>	<u>8</u> * -	Ċ	• 6	* *	ş. ▼	•	<b>9</b> -	9	S
Project	Expl	orer X	🗖 🔁 🗖 🕴													
: 📭 wifi	scan	(in STM32CubelDE)	1													
		New	>													
		Go Into														
		Open in New Window														
		Show In	Alt+Shift+W >													
	D	Сору	Ctrl+C													
	Ē	Paste	Ctrl+V													
	×	Delete	Delete													
		Source	>													
		Move														
		Rename	F2													
	è	Import														
	പ്പ	Export														
		Build Project														
		Clean Project														
	\$	Refresh	F5													
		Close Project				Prob	lems .	🗊 Tacke	E	Conse			Droper	tion		
		Close Unrelated Project				No conso	oles to d	isplay at	t this	time.			riopei	ues		
		Build Configurations	>													
		Build Targets	>													
		Index	>													
	0	Run As	>													
	*	Debug As	>													
		Team	>													
		Compare With	>													
		Restore from Local History														
	\$	Generate Code														
	C.+	Convert to C++														
	*	Run C/C++ Code Analysis														
		Contigure	>													
		Properties	2 Alt+Enter													
	_															

- 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.





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

#### Figure 17: STM32CubeIDE Build Successful Results

Console      X	0) 10 15 15 15 15 15 15 15 15 15 15 15 15 15
CDT Build Console [wifi_scan_CM7]	
arm-none-eabi-gcc -o "wifi_scan_CM7.elf" @"objects.list"	-mcpu=cortex-m7 -T"C:\wifi_scan\STM32Cu ^
Finished building target: wifi_scan_CM7.elf	
arm-none-eabi-size wifi_scan_CM7.elf arm-none-eabi-objdump -h -S wifi_scan_CM7.elf > "wifi_sc arm-none-eabi-objcopy -O ihex wifi_scan_CM7.elf "wifi_s text data bss dec hex filename 683508 376 145808 829692 ca8fc wifi_scan_CM7.elf Finished building: default.size.stdout	can_CM7.list" scan_CM7.hex"
Finished building: wifi_scan_CM7.hex	
Finished building: wifi_scan_CM7.list	
14:53:34 Build Finished. 0 errors, 0 warnings. (took 1m:1s	s.851ms)
	~

## 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: STM32CubelDE Run Application

Edit Sou	rce Refactor Navigate Sear	ch Project Run	Window	Help							
- 8 6	🛞 <b>- 4</b> - 🔜 🛛 💌 🛛		r   🗟	7 💩 👩 🗸	· 🚳 - 🖻	- 🞯 - 🕸	🌾 🕶 🚺 🤜	• 💁 🖬	🔗 👏	- 12	I 1
roject Explo		🗖 🔁 🖓 🕴									
wifi_s		1									
淼 Bii 	Galata	,									
Ar	do into										
🗁 De	Open in New Window										
🗁 Dr	Show In	Alt+Shift+W >									
	Сору	Ctrl+C									
🔒 ST 🗅	Paste	Ctrl+V									
📉 wi X	Delete	Delete									
	Source	>									
	Move										
	Rename	F2									
2	Import										
4	Export										
	Build Project										
	Clean Project										
\$	Refresh	F5									
	Close Project										
	Close Unrelated Project			Problems	🚈 Tasks 🖳	Console $\times$	Prop	erties			
	Build Configurations	>		CDT Build Conso	ble [wifi_scan]		011/0301/1				
	Build Targets	>		arm-none-eab arm-none-eab	i-gcc "C:/ i-gcc -o '	Users/M11 wifi scan	082H/STM3 .elf" @"(	32Cube/E obiects.	xample/ list"	wifi_s -mcpu	can/( =cort
	Index	>		Finished bui	lding targ	get: wifi_	scan.elf				
0	Run As	> [	DE 1 STI	M32 C/C++ Appl	ication	_s an.	elf 🤅	3			
蓉	Aebug As	>	Run	Configurations		) ihex	wifi_scan.	an.elf	wifi_so	can.he	t x"
	Team	, 1		676452	aca U2	dec dec	hex	filenam	e an elf		
	Compare With	>		Finished bui	lding: def	ault.size	.stdout	w111_50	an.e11		
	2 pre from Local History			Finished bui	lding: wif	i scan he	~				
	Generate Code			1 Ini Sheu Dui	.10106. W1	1_scan.ne	~				
÷	Convert to C++			Finished bui	lding: wif	i_scan.li	st				
2	Kun C/C++ Code Analysis	,		15.20.51 8.4	ld Cinicha	d 0 0000			(took 1	m. 20 c	120-
		Alter		15.25.51 001	iu i misne	u. e erro	15, 0 wai	nings.	(1008 1		10000
	Droportion										

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)									
<pre>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</pre>									
# SSID Security	RSSI	Channel	MAC Address						
1 XXXXXXXXXXXXX WPA2_AES_PSK 2 YYYYYYYYYYYYYYYYYY	-78	36	BC:26:C7:43:DF:B5						
OPEN 3 Zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz	-49	36	A8:5E:45:64:2F:CC						

# 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 ⊑<sup>r</sup>.

## 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 : w10: 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

    -42
    1
    24:4B:FE:D1:40:18

    -60
    11
    D8:9B:3B:3E:06:D0

    -47
    11
    C4:41:1E:9D:EB:57

                      open

        YYYYYYYYYYYY
        wpa2

                                                 11
ZZZZZZZZZZZZZZZZZZZZZ wpa2 aes
                                                           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





Copyright © Murata Manufacturing Co., Ltd. All rights reserved. The information and content in this document are provided "as-is" with no warranties of any kind and are for informational purpose only. Data and information have been carefully checked and are believed to be accurate; however, no liability or responsibility for any errors, omissions, or inaccuracies is assumed.

Wi-Fi<sup>®</sup> is a registered trademark of Wi-Fi Alliance. The Bluetooth<sup>®</sup> word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. Other brand and product names are trademarks or registered trademarks of their respective owners.

Specifications are subject to change without notice.