# ISP5261
## Application note AN240601

# Use of ISP5261 Evaluation Kit

## Introduction

### Scope

This document gives details on hardware and software for using and testing Insight SiP Bluetooth Low Energy and Wi-Fi 6 module ISP5261.

### Contents


1. **Recommended Documentation** ..... **3**
2. **Hardware kit content** ..... **4**
2.1. Global description ..... 4
2.2. ISP5261 module ..... 4
2.3. ISP5161-WX-TB test board ..... 5
2.4. ISP130603H interface board ..... 6
3. **Getting started with the kit** ..... **7**
3.1. J-Link drivers installation ..... 7
3.2. Setup the boards ..... 8
4. **Use of the kit with MCUXpresso SDK** ..... **9**
4.1. Software download & installation ..... 9
4.2. Setup SDK examples ..... 10
4.3. Build and run a Wi-Fi application ..... 14
4.4. Build and run a BLE application ..... 18
4.5. Loading protocol binary images (monolithic feature disabled) ..... 21
4.6. Flash memory partition table ..... 24
5. **Use of the kit with Zephyr OS** ..... **25**
5.1. Software download & installation ..... 25
5.2. Setup NXP Zephyr repository ..... 27
5.3. Build and run a Wi-Fi application ..... 29
5.4. Build and run a BLE application ..... 32





The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

## Document Revision History

| Revision | Date | Ref | Change Description |
|---|---|---|---|
| R0 | 12/11/2024 | jf, cb | Initial release |
| R1 | 02/04/2025 | Jf pg | Added module variants<br>Added monolithic feature<br>Added Zephyr OS chapter |


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

## 1. Recommended Documentation

The following documents are required to understand the complete setup and programming methods.

### NXP Documents

The following documents are available on NXP website:

- RW612 datasheet
  Link: https://www.nxp.com/docs/en/data-sheet/RW612.pdf
- UM12035: Getting Started with RW61x Running Zephyr OS
  Link: https://www.nxp.com/docs/en/user-manual/UM12035.pdf
- UM11799: NXP Wi-Fi and Bluetooth Demo Applications for RW61x
  Link: https://www.nxp.com/docs/en/user-manual/UM11799.pdf
- UM11798: Getting Started with Wireless on RW61x Evaluation Board Running FreeRTOS:
  Link: https://www.nxp.com/docs/en/user-manual/UM11798.pdf

### Insight SiP documents

The following documents are available on Insight SIP website or/and on request:

- AN240601 App Note – this document.
- ISP5261 data sheet.
- ISP5261-WX-TB schematic.
- ISP130603H Interface Board schematic.

You can access this document at https://www.insightsip.com/products/combo-smart-modules/isp5261.


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

## 2. Hardware kit content

### 2.1. Global description

The picture below shows the Evaluation Kit hardware content for the ISP5261 module.



*Figure 1; Interface board (on the left) and test board (on the right)*

A detailed description of the interface board ISP130603H is given in section 2.4.

### 2.2. ISP5261 module

The ISP5261 module is based on NXP RW61x single-chip Wi-Fi/Bluetooth LE Wireless MCU System on Chip (SoC).




*Figure 2: ISP5261 module*

It integrates a 32-bit ARM CortexTM-M33 CPU running at 260MHz, 4MB of QSPI flash memory, 1.2MB SRAM as well as analog and digital peripherals. Despite the small size of 12 x 12 x 1.8 mm, the module integrates decoupling capacitors, 40 MHz crystal and 32.768kHz crystal for low power timing, DC-DC converters, RF matching circuits and a dual-band antenna. Low power consumption and advanced power management enable battery lifetimes up to several months on AA batteries.

The ISP5261 is available in 2 variants:

- WX: includes RW610, providing Wi-Fi and BLE features
- WT: includes RW612, providing Wi-Fi, BLE and 802.15.4 features

For more details, see the ISP5261 datasheet.


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

## 2.3. ISP5161-WX-TB test board

The ISP5261-WX-TB test board consists of a module mounted on a PCB for prototyping and testing purposes. It has dimensions of 47 x 45 mm².

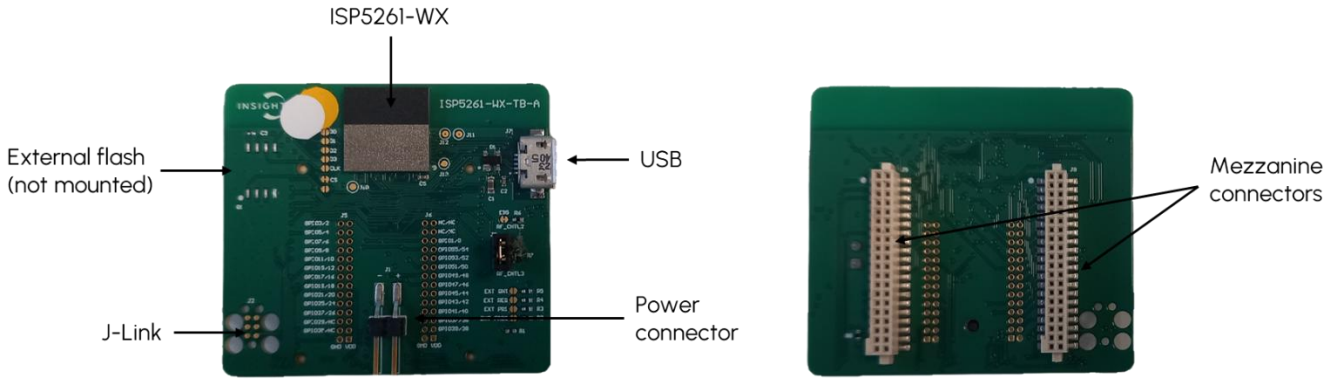



*Figure 3: ISP5261-WX test board*

It encloses:

- An ISP5161-WX module.
- 2 mezzanine connectors at the bottom side of the board (for connection to an interface board).
- 2 footprints for optional 2x14 pin/1.27mm pitch connector for access to the module pins.
- Footprint for programming using 6 pin Segger J-Link Adapter.
- https://www.segger.com/products/debug-probes/j-link/accessories/adapters/6-pin-needle-adapter/
- 2-pin header for external power supply.
- USB connector connected to the module's USB interface.
- Footprint for optional external flash memory.


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

## 2.4. ISP130603H interface board

The ISP130603H board is an interface board that has dimensions of 100x70mm². It can interface with test boards and is responsible for providing:

- Power supply.
- Programming and debugging capabilities.
- Virtual COM port.
- Access to all general-purpose input/outputs of the module.
- Buttons & LEDs.

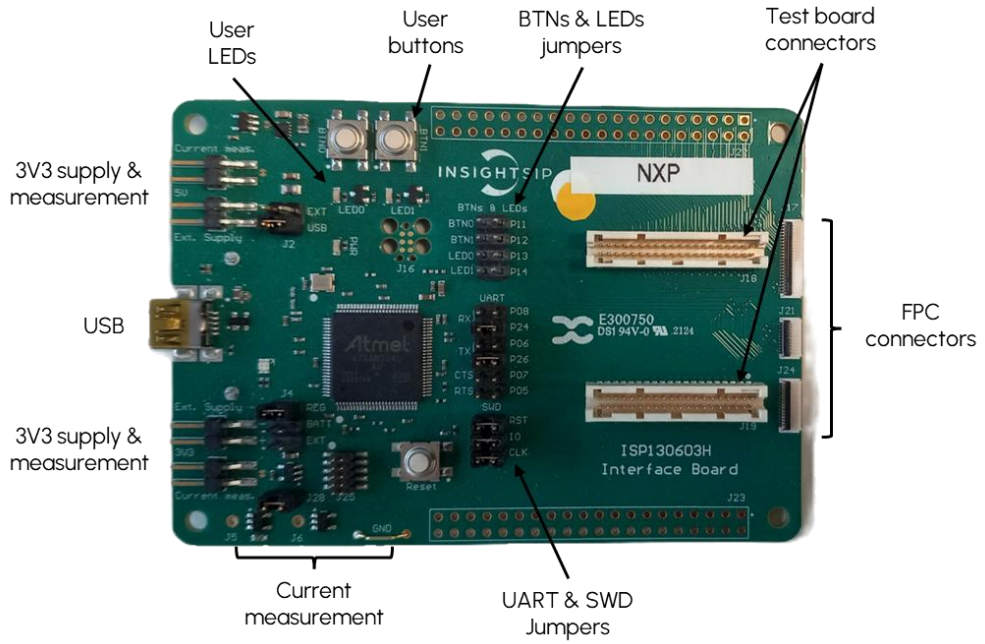



*Figure 4: ISP130603H Interface board*


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

## 3. Getting started with the kit

### 3.1. J-Link drivers installation

Flashing & debugging the ISP5261 module is done using a J-Link probe. An onboard version of J-Link probe is available on the ISP130603H Interface board.

Download the latest version of J-Link software at: https://www.segger.com/downloads/jlink

Once the J-Link software has been installed set a Path in the Environment variables to the location of JLink.exe for the latest version of the software:

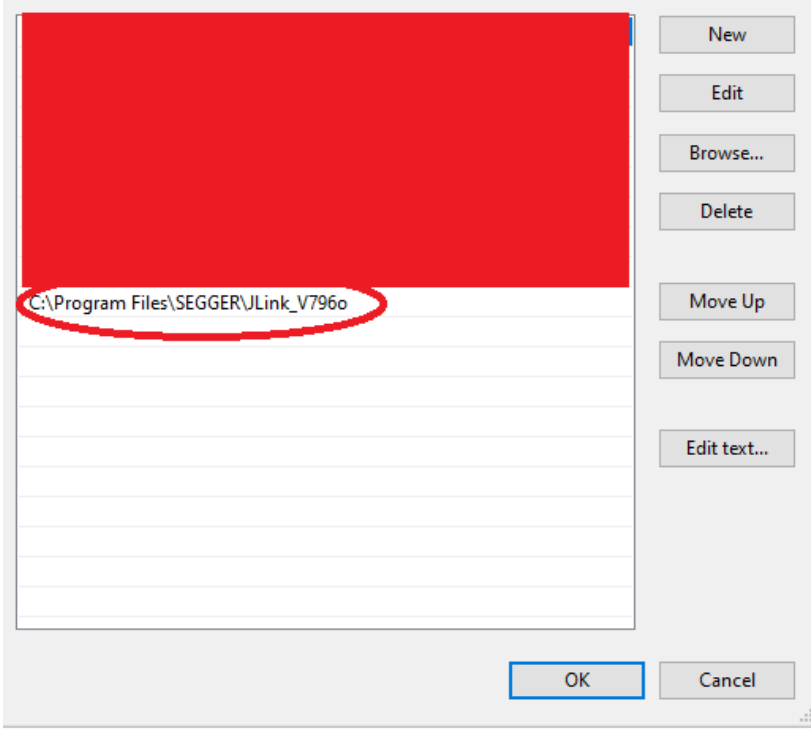



*Figure 5: Setting an environment variable*


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

### 3.2. Setup the boards

To start working with the ISP5261-WX module, plug the test board (ISP5162-WX-TB) on the interface board (ISP130603H) using the two mezzanine connectors.

Make sure that:

- J2 jumper is on ***USB*** position.
- J4 jumper is on ***REG*** position.
- All 3 ***SWD*** jumpers (***RST***, ***IO*** and ***CLK***) are present.
- The UART jumpers should be set to **RX –> P24**, **TX –> P26, CTS et RTS NC**

Plug the USB connector to your computer

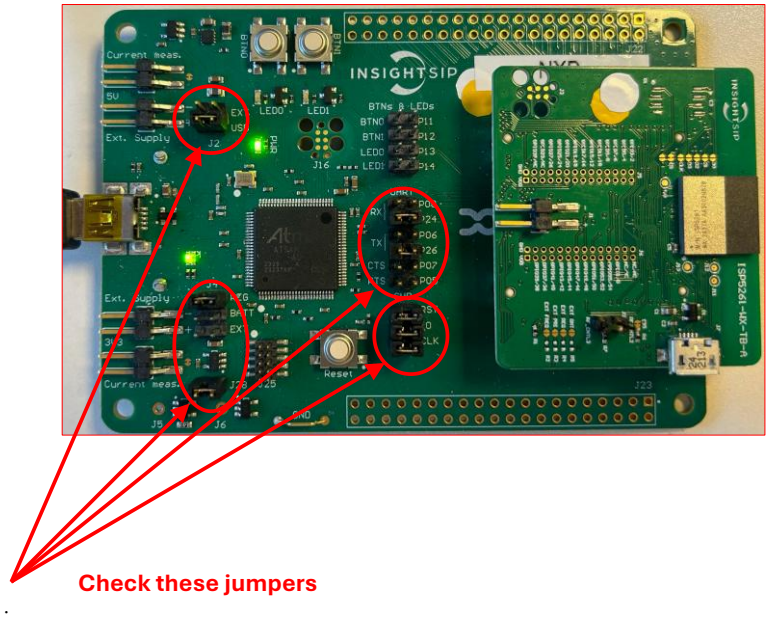



*Figure 6: Setup on Interface Board*

On the computer open ***Device Manager*** and check that the following devices appear in the list:

- J-Link driver in Universal Serial Bus Controllers.
- J-Link CDC UART Port in Ports.

If they do not appear install that J-Link drivers on the computer. They can be downloaded here: https://www.segger.com/downloads/jlink

The boards have been pre-programmed with the Wi-Fi command line interface program from the SDK and the network protocol images. See chapter 4.4.3. Test to learn how to run this application.


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

# 4. Use of the kit with MCUXpresso SDK

## 4.1. Software download & installation

### 4.1.1. MCUXpresso Integrated Development Environment (IDE)

The ***MCUXpresso IDE*** is the official development Environment of NXP. Other IDEs are compatible with the ISP5261-WX such as Visual Code Studio, Keil uVision and IAR.

***MCUXpresso IDE*** can be downloaded at:
https://www.nxp.com/design/design-center/software/development-software/mcuxpresso-software-and-tools-/mcuxpresso-integrated-development-environment-ide:MCUXpresso-IDE

### 4.1.2. MCUXpresso SDK

The SDK can be directly downloaded from ***MCUXpresso IDE***.
This document is written with MCUXpresso IDE v24.12.48 in mind. If using a later version follow the guidelines in the IDE.

When opening the IDE the Welcome page will be shown:

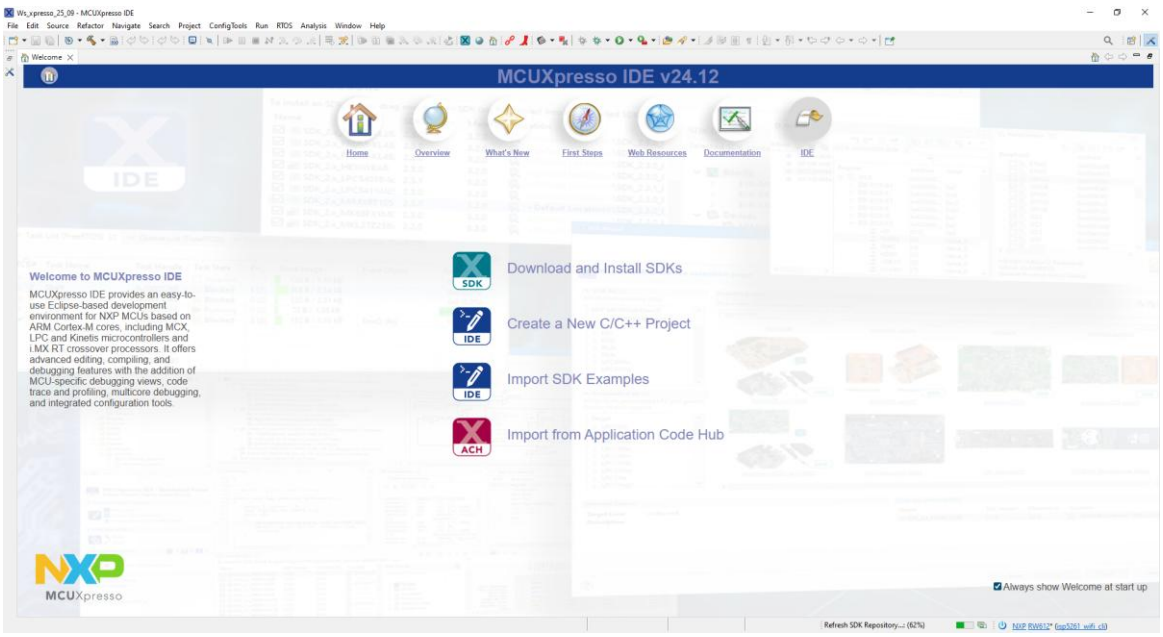



*Figure 7: MCUXpresso IDE Welcome page*

Select the option Import "Download and Install SDKs":


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

Choose RW610 in the Processors tab and Install:

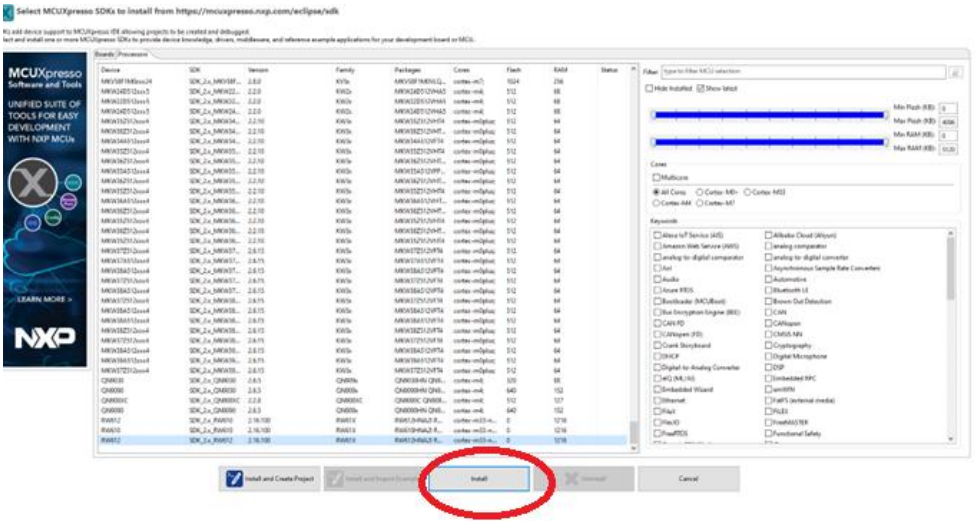


*Figure 8: Install SDK*

## 4.2. Setup SDK examples

After the SDK is installed select "Install SDK Examples" on the home page




*Figure 9: Select examples using Wizard*

Use the wizard to select examples.


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

You should see a page on which you can select a board. Currently there are two boards that can be selected. In this note we choose the frdmrw612 board as a starting point:

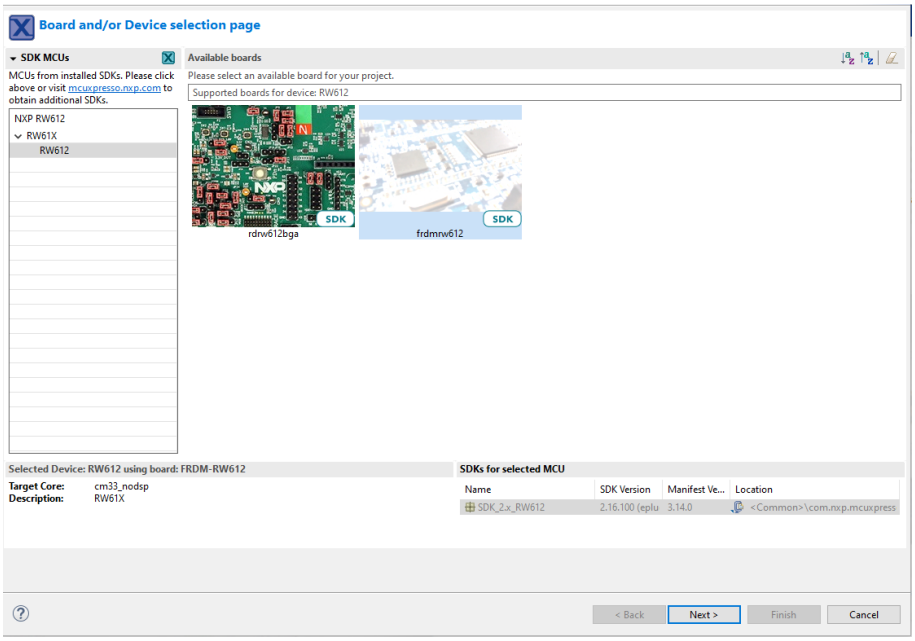



*Figure 10: Select a board*

Choose the board and enter Next.

On the next page you can select one or more examples to load. For the purposes of this note we have selected 2 Bluetooth examples:

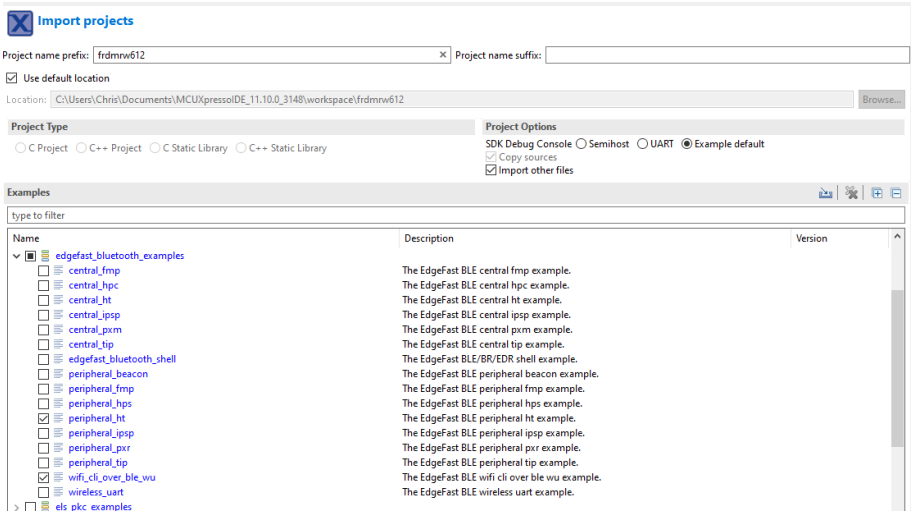



*Figure 11: Selct BLE examples*


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

And then we have selected a series of Wi-Fi examples:

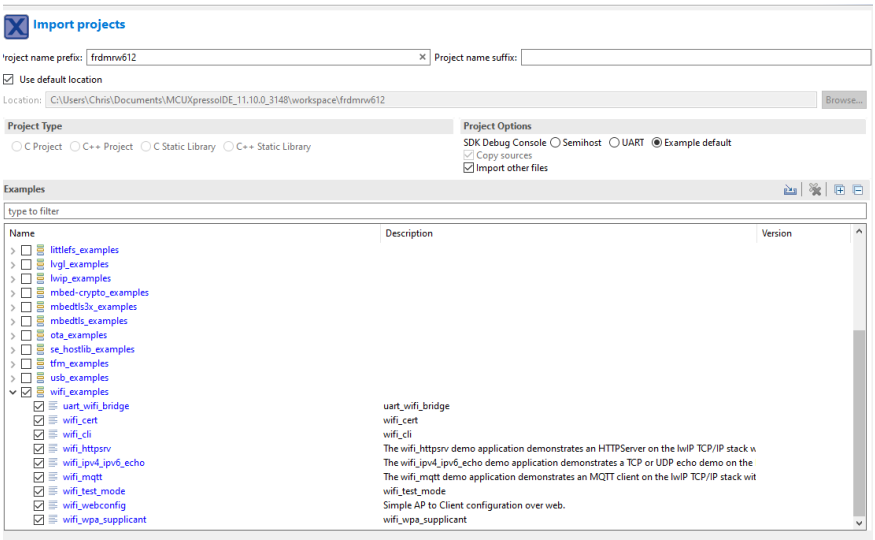



*Figure 12: Select WiFi examples*

In the "Installed SDKs" panel right click and Select "Download and install SDKs".
The workspace will open with the examples selected previously loaded:

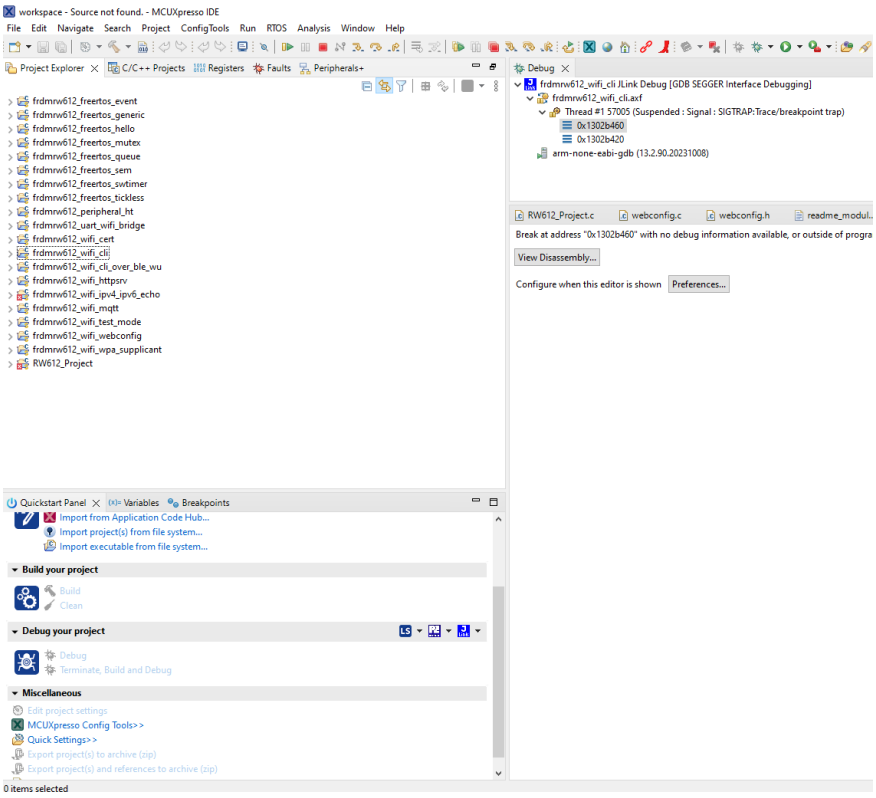



*Figure 13: Selected examples*


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

The FRDM-RW612 board uses an external flash with 64 MB. The ISP5261 module is fitted with a 4MB flash memory. Thus, to run the examples on the ISP521 module, modifications must be made to a few files.

For all projects the "flash_config.c" file must be modified. It can be found at <project directory>flash_config/ as shown below for the wifi command line example:

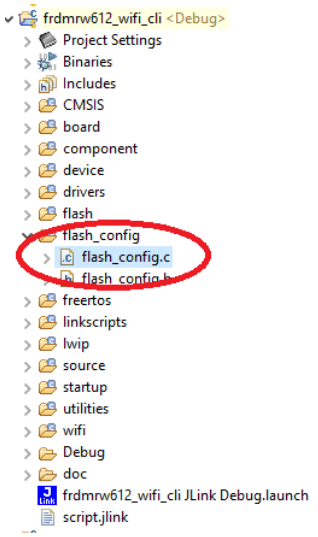



*Figure 14: flash_config.c position*

For applications that use LittleFS File system it is also necessary to change 2 files in each such project that are under <project directory>/flash/mflash/frdmrw612/:
"mflash_drv.c"
"mflash_drv.h"

All the required files are provided together in a zip file on Insight SiP website. "ISP5261_memory_config.zip" and should be used to replace the files in each project.


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

## 4.3. Build and run a Wi-Fi application

### 4.3.1. Build the project

A simple Wi-Fi example (***wifi_cli***) was pre-loaded on the test board. This chapter explains how to build this application, starting from the SDK and re-load it.

In the workspace that was created earlier select the ***wifi_cli*** example.

Modify the ***flash_config.c*** file as described in Setup SDK examples chapter.

To build the application click on "build" in the Quickstart panel.
The application should be generated in the Debug / Release sub-folder of the project.

**Note:**
**wifi_cli** use monolithic feature, this means than the protocol binary file will be included in the project and no additional step is necessary.
To check if monolithic feature is enabled check that CONFIG_MONOLOTHIC_WIFI symbol is enabled.

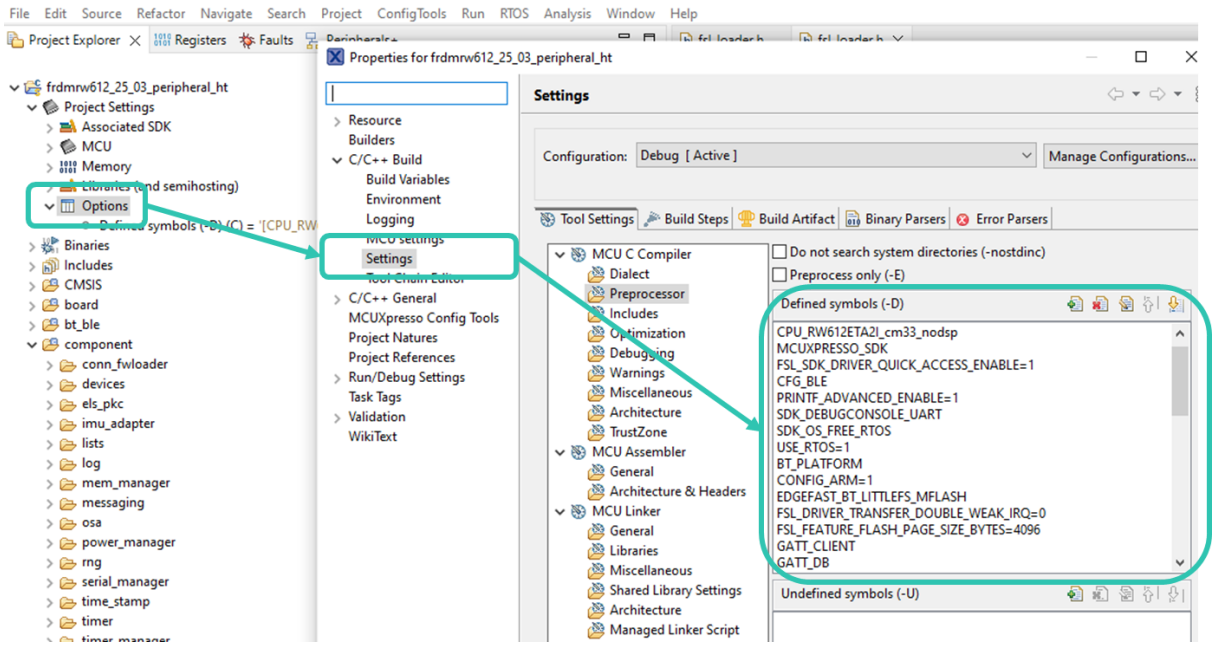



*Figure 15: Monolithic symbol*


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

### 4.3.2. Load the application

To load the application, click on the J-Link icon in the Quickstart panel, then click "Program flash action using SEGGER J-Link probes".

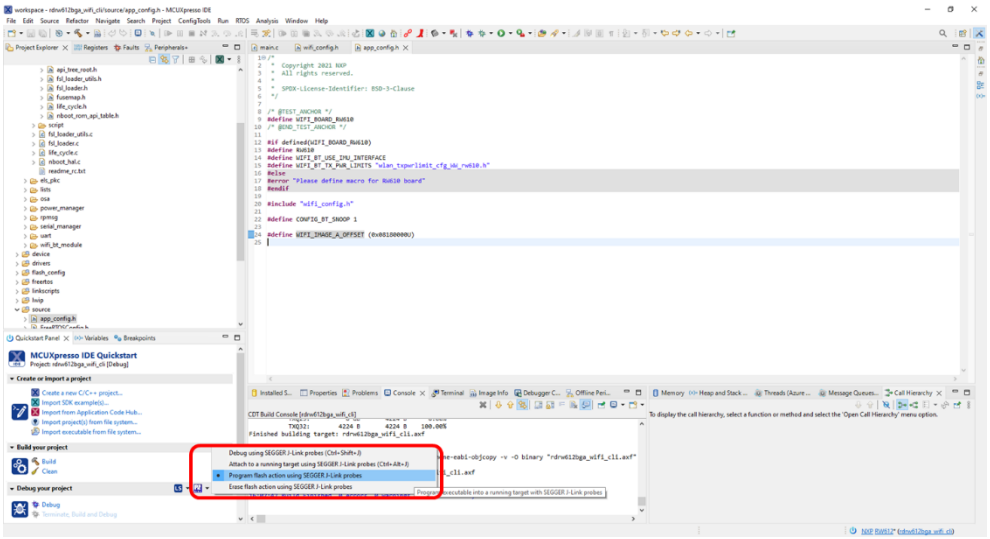



*Figure 16: Load Wi-Fi application using MCUXpresso IDE*

### 4.3.3. Test

Open a serial terminal. In this chapter we will use ***TeraTerm***.

Choose Serial and select the COM port of the ISP130603H board (J-Link CDC UART Port). The port number will differ from one system to another.

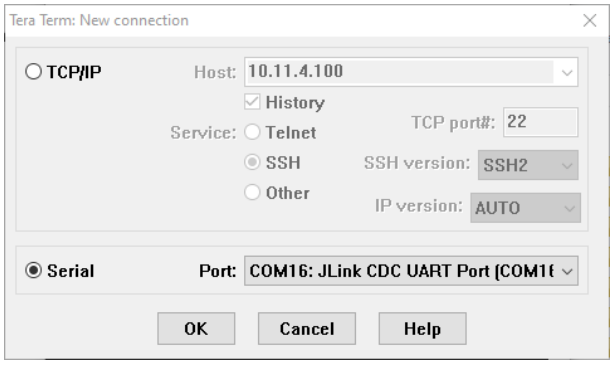



*Figure 17: Serial port selection in TeraTerm*

Go to Setup -> Serial port and set the speed to ***115200***.


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

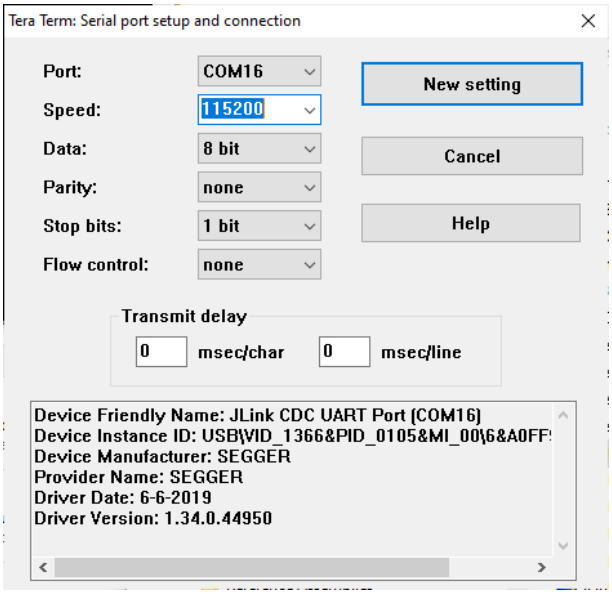



*Figure 18: Speed setting in TeraTerm*

Go to Setup -> Terminal and set New-line Receive to "AUTO".

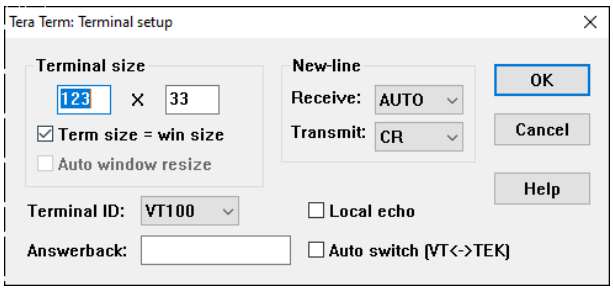



*Figure 19: Terminal setting in TeraTerm*

Press the Reset button on the ISP130603H Interface board and a welcome message and a list of available commands should appear on the terminal.


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

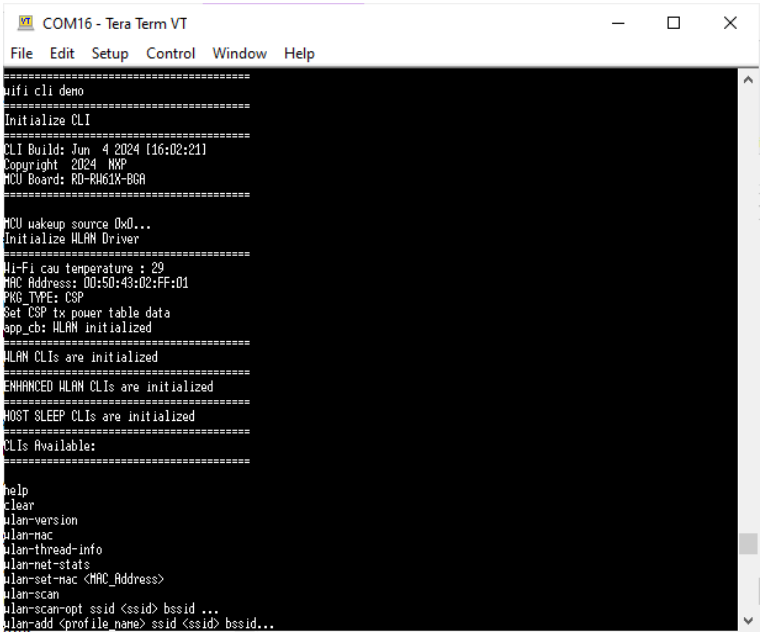



*Figure 20: WiFi cli command options in TeraTerm*

Now you have access to all Wi-Fi commands.

For example, write ***wlan-scan*** and the list of Access point around should be displayed.

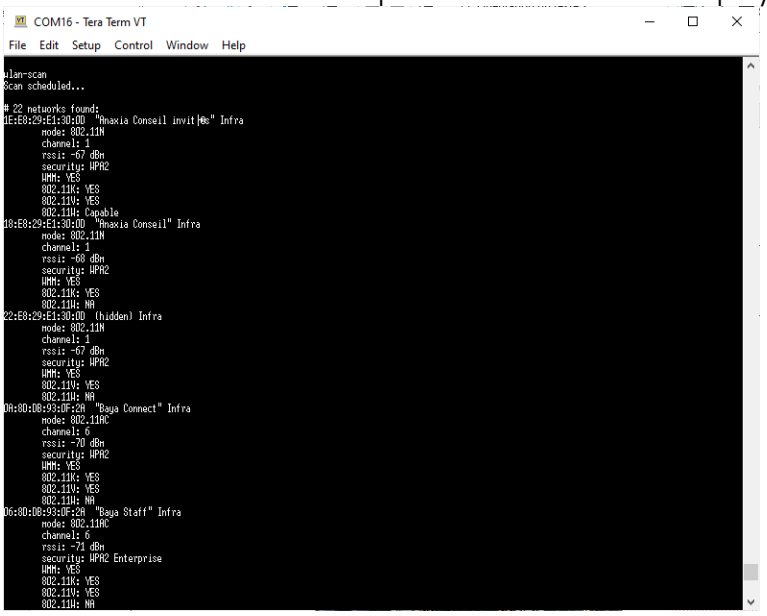



*Figure 21: wlan-scan command results in TeraTerm*


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

## 4.4. Build and run a BLE application

### 4.4.1. Build the project

This chapter will explain how to build a simple BLE example called ***peripheral_ht***. This example implements a peripheral role with the health thermometer service.

Laad and open ***peripheral_ht*** sample.

Modify the ***flash_config.c***, file as described in Setup SDK examples chapter.

Load protocol binaries.
As for SDK 24.12.48, ***peripheral_ht*** does not use monolithic feature, this means than the protocol binary files must be loaded manually at specific addresses.
See chapter Loading protocol binary images (monolithic feature disabled)

Before building the application, the memory partition table must be modified.
See chapter Flash memory partition table for more information.
This can be simply done in the ***fsl_loader.h*** file by adding the following line:

#define WIFI_IMAGE_A_OFFSET (0x08180000U)

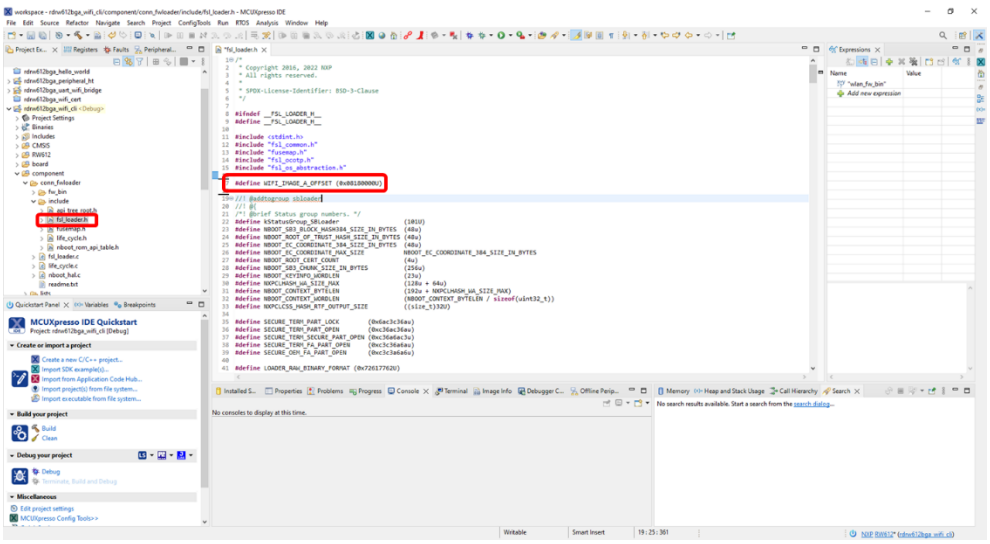


*Figure 22: Modify memory partitioning in MCUXpresso IDE*

The fsl_loader.h file is located at: <your_project_location>\component\conn_fwloader\include.

**Note**: We only need to modify the address of the Wi-Fi A image. The address of all other images will be shifted accordingly.

Furthermore, since this project uses LittleFS file system it is necessary to change the mflash_drv.c and mflash_drv.h files as indicated earlier.


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

The memory layout that is defined in the project options must also be changed to match the memory partition for 4MB.

This is done by "editing the Memory table as shown below:

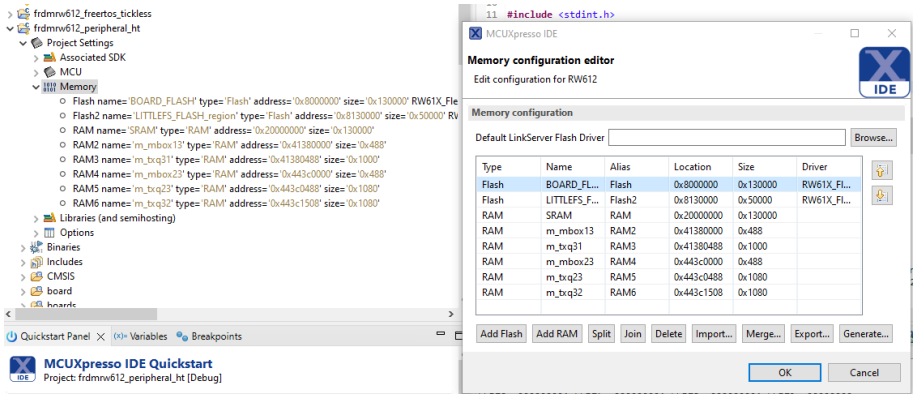



*Figure 23: Modify memory partition*

Note that Board Flash location is 0x8000000 with size 0x360000, LITTLE_FS Flash location is 0x83b0000 with size 0x50000.

To build the application click on build in the Quickstart panel.
The application should be generated in the Debug / Release sub-folder of the project.

### 4.4.2. Load the application

To load the application, click on the J-Link icon in the Quickstart panel, then click "Program flash action using SEGGER J-Link probes".

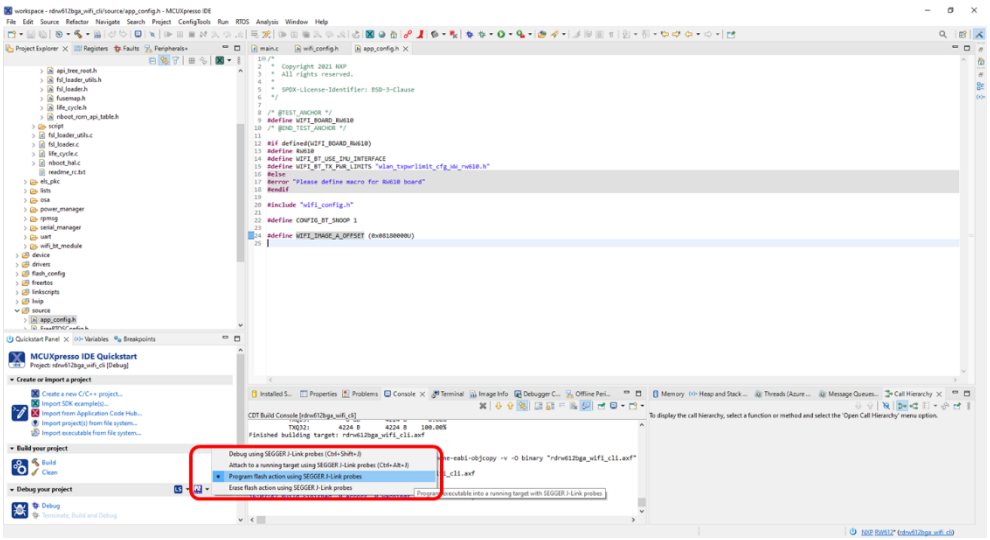



*Figure 24: Load BLE application using MCUXpresso IDE*


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

### 4.4.3. Test

Launch the ***IoT Toolbox*** application on your smartphone and select the Thermometer service. A device called ***peripheral_ht*** should be advertising, select it to connect.

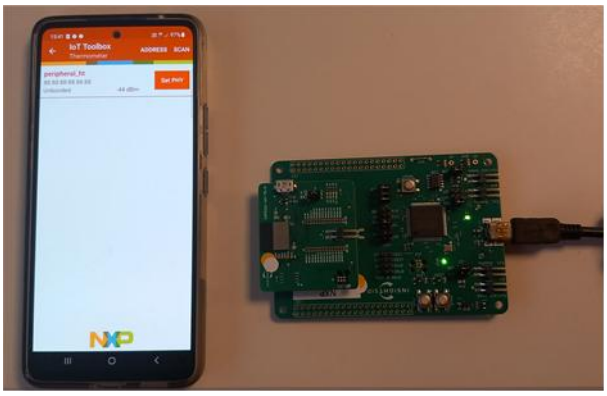


*Figure 25: Device advertising*

Once connected, temperature readings show on the smartphone.

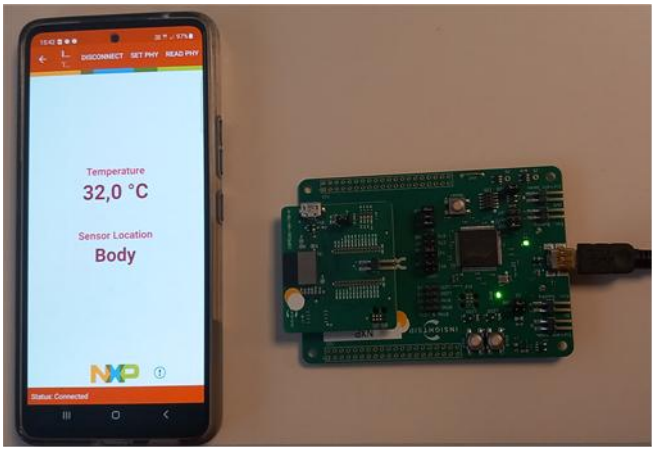



*Figure 26: Temperature readings on smartphone*


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

### 4.5. Loading protocol binary images (monolithic feature disabled)

Should you erase the complete memory of the module, follow the following steps to re-load the protocol binary images.

Depending on what radio the application needs, protocol binary images must be loaded onto the memory flash. These images need to be loaded at specific addresses in the memory flash. For more detail see Flash memory partition table annex chapter located at the end of the document.

The image files are located at:

<project folder>/component/con_fwloader/fw_bin/

The folder should contain:

- ***Rw61x_sb_wifi_a2.bin*** for the Wi-Fi protocol.
- ***Rw61x_sb_ble_a2.bin*** for the BLE protocol.
- ***rw61x_sb_ble_15d4_combo_a2.***bin for ble / 802.15.4 (WT variant only)

**Note:**
The image filenames might slightly differ depending on which SDK version is used.

Since the path for the JLink commander has been set as an environment variable the files can be loaded directly from the command prompt or windows power shell once the directory has been set correctly to the location of the binary files.

In windows power shell set directory to: <sdk folder>/components/conn_fwloader/fw_bin/
Start JLink
>Connect
Select device
> RW610
Target
>S (SWD)
Speed
>4000


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

Windows PowerShell

```
PS F:\> cd .\SDK_2_16_100_FRDM-RW612\
PS F:\SDK_2_16_100_FRDM-RW612> cd .\components\
PS F:\SDK_2_16_100_FRDM-RW612\components> cd .\conn_fwloader\
PS F:\SDK_2_16_100_FRDM-RW612\components\conn_fwloader> cd .\fw_bin\
PS F:\SDK_2_16_100_FRDM-RW612\components\conn_fwloader\fw_bin> Jlink
SEGGER J-Link Commander V7.94e (Compiled Jan 15 2024 15:20:36)
DLL version V7.94e, compiled Jan 15 2024 15:18:46

Connecting to J-Link via USB...O.K.
Firmware: J-Link OB-SAM3U128 V3 compiled Sep 18 2024 15:45:39
Hardware version: V3.00
J-Link uptime (since boot): 0d 05h 14m 49s
S/N: 483154944
USB speed mode: High speed (480 MBit/s)
VTref=3.300V


Type "connect" to establish a target connection, '?' for help
J-Link>connect
Please specify device / core. <Default>: RW612
Type '?' for selection dialog
Device>RW612
Please specify target interface:
  J) JTAG (Default)
  S) SWD
  T) cJTAG
TIF>S
Specify target interface speed [kHz]. <Default>: 4000 kHz
Speed>4000
Device "RW612" selected.


Connecting to target via SWD
InitTarget() start
SWD selected. Executing JTAG -> SWD switching sequence.
DAP initialized successfully.
  DMAP_REG_ID: DM-AP IDCODE detected: 0x002A0000
  DHCSR: 0x05100000
InitTarget() end - Took 4.27ms
Found SW-DP with ID 0x6BA02477
DPIDR: 0x6BA02477
CoreSight SoC-400 or earlier
Scanning AP map to find all available APs
AP[1]: Stopped AP scan as end of AP map has been reached
AP[0]: AHB-AP (IDR: 0x84770001)
Iterating through AP map to find AHB-AP to use
AP[0]: Core found
AP[0]: AHB-AP ROM base: 0xE00FF000
CPUID register: 0x410FD214. Implementer code: 0x41 (ARM)
Feature set: Mainline
Cache: No cache
Found Cortex-M33 r0p4, Little endian.
FPUnit: 8 code (BP) slots and 0 literal slots
Security extension: implemented
Secure debug: enabled
CoreSight components:
ROMTbl[0] @ E00FF000
[0][0]: E000E000 CID B105900D PID 000BBD21 DEVARCH 47702A04 DEVTYPE 00 Cortex-M33
[0][1]: E0001000 CID B105900D PID 000BBD21 DEVARCH 47701A02 DEVTYPE 00 DWT
[0][2]: E0002000 CID B105900D PID 000BBD21 DEVARCH 47701A03 DEVTYPE 00 FPB
[0][3]: E0000000 CID B105900D PID 000BBD21 DEVARCH 47701A01 DEVTYPE 43 ITM
[0][5]: E0041000 CID B105900D PID 002BBD21 DEVARCH 47724A13 DEVTYPE 13 ETM
[0][6]: E0042000 CID B105900D PID 000BBD21 DEVARCH 47701A14 DEVTYPE 14 CSS600-CTI
Memory zones:
  Zone: "Default" Description: Default access mode
Cortex-M33 identified.
```

*Figure 27: Navigate to FW_bin and prepare JLink to load protocol images*

To write protocol images type as below:

- loadbin rw61x_sb_wifi_a2.bin 0x08180000
- loadbin rw61x_sb_ble_a2.bin 0x082c0000
- loadbin rw61x_sb_ble_15d4_combo_a2.bin 0x08360000


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

```
J-Link>loadbin rw61x_sb_wifi_a2.bin 0x08180000
'loadbin': Performing implicit reset & halt of MCU.
ResetTarget() start
Reset via SYSRESETREQ and reset pin + halt after bootloader
ROM entered ISP command handling loop. Re-enable the debug access.
  MSPLIM cleared
ResetTarget() end - Took 118ms
Downloading file [rw61x_sb_wifi_a2.bin]...
J-Link: Flash download: Bank 0 @ 0x08000000: 1 range affected (589824 bytes)
J-Link: Flash download: Total: 8.965s (Prepare: 0.266s, Compare: 1.684s, Erase: 2.756s, Program: 3.244s, Verify: 0.866s,
 Restore: 0.146s)
J-Link: Flash download: Program speed: 177 KB/s
O.K.
J-Link>loadbin rw61x_sb_ble_a2.bin 0x082c0000
'loadbin': Performing implicit reset & halt of MCU.
ResetTarget() start
Reset via SYSRESETREQ and reset pin + halt after bootloader
ROM entered ISP command handling loop. Re-enable the debug access.
  MSPLIM cleared
ResetTarget() end - Took 118ms
Downloading file [rw61x_sb_ble_a2.bin]...
J-Link: Flash download: Bank 0 @ 0x08000000: 1 range affected (262144 bytes)
J-Link: Flash download: Total: 4.086s (Prepare: 0.256s, Compare: 0.792s, Erase: 1.205s, Program: 1.304s, Verify: 0.381s,
 Restore: 0.146s)
J-Link: Flash download: Program speed: 196 KB/s
O.K.
J-Link>loadbin rw61x_sb_ble_15d4_combo_a2.bin 0x08360000
'loadbin': Performing implicit reset & halt of MCU.
ResetTarget() start
Reset via SYSRESETREQ and reset pin + halt after bootloader
ROM entered ISP command handling loop. Re-enable the debug access.
  MSPLIM cleared
ResetTarget() end - Took 118ms
Downloading file [rw61x_sb_ble_15d4_combo_a2.bin]...
J-Link: Flash download: Bank 0 @ 0x08000000: 1 range affected (327680 bytes)
J-Link: Flash download: Total: 5.145s (Prepare: 0.252s, Compare: 0.882s, Erase: 1.524s, Program: 1.859s, Verify: 0.479s,
 Restore: 0.146s)
J-Link: Flash download: Program speed: 172 KB/s
O.K.
J-Link>
```

*Figure 28: Download protocol images*

Note that these memory locations have been chosen to fit the application and the 2 protocol images in the 4 MB memory.

These protocol image files only need to be loaded once on the module (unless a full chip erase is performed).


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

### 4.6. Flash memory partition table

The flash memory is divided into different blocks or partitions. Each partition contains either the user application a LittleFS file system or a radio protocol stack (Wi-Fi, BLE and/or 802.15.4).

The ***MCUXpresso SDK*** comes with default partitioning configured for 64 MB of flash. However, the ISP5261 module has an internal flash memory of 4MB and so the default partitioning does not fit the internal flash memory.

The following table is a suggested partitioning for a 4MB memory of the ISP5261:

*Table 1 : Flash memory partition*

| | Start address | Maximum size |
|---|---|---|
| **LittleFS** | 0x083B0000 | 0xA0000 (0.625MB) |
| **15.4 Image B** | 0x08360000 | 0x50000 (0.3125MB) |
| **15.4 Image A** | 0x08310000 | 0x50000 (0.3125MB) |
| **BLE Image B** | 0x082C0000 | 0x50000 (0.3125MB) |
| **BLE Image A** | 0x087C0000 | 0x50000 (0.3125MB) |
| **WIFI Image B** | 0x081D0000 | 0xA0000 (0.625MB) |
| **WIFI Image A** | 0x08130000 | 0xA0000 (0.625MB) |
| **Application** | 0x08000000 | 0x130000 (1.1875MB) |

This partitioning enables all 3 radios and provides 1.1875MB of flash memory for the user application and 0.625s MB of flash memory for the LittleFS file system.
This is an example of partitioning table that covers all features. If one these features is not required for the project this table can be updated accordingly in order to increase application memory size.


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

# 5. Use of the kit with Zephyr OS

## 5.1. Software download & installation

### 5.1.1. Visual Studio Code:

Download at https://visualstudio.microsoft.com/fr/downloads/.

### 5.1.2. MCUXpresso plug-in for Visual Studio Code

Open Visual Studio Code and go to ***Extension***
Search for ***MCUXpresso for VS Code*** and click Install

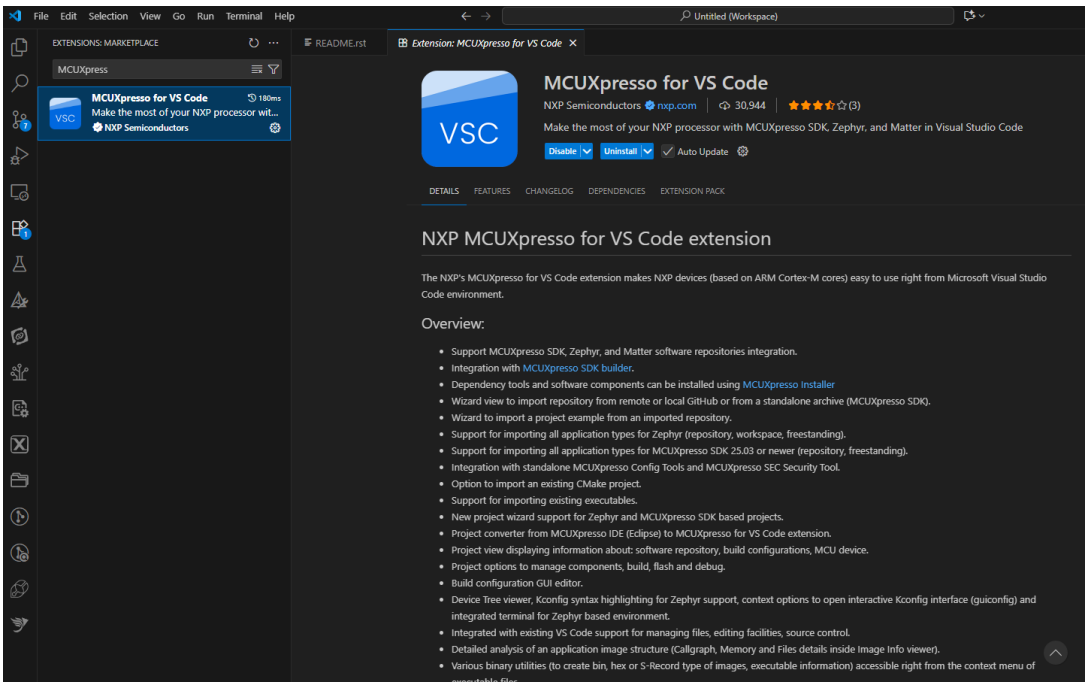



*Figure 29: MCUXpresso for VS Code Extesnion*

Once installed go the ***MCUXpresso for VS Code*** menu and click on Open ***MCUXpresso installer***. A new window will open, Select ***Zephyr Developer*** and click install.


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

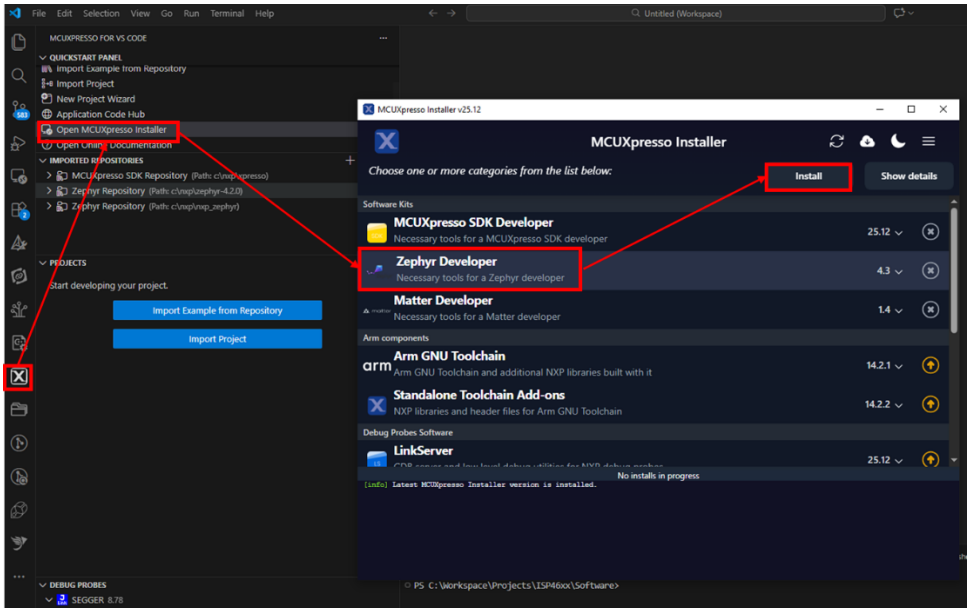



*Figure 30: MCUXpresso installer*

### 5.1.3. Zephyr software development kit (SDK)

Go to Zephyr SDK github page at https://github.com/zephyrproject-rtos/sdk-ng/releases and download the latest SDK full bundle for Windows (zephyr-sdk-0.X.Y_windows-x86_64.7z).

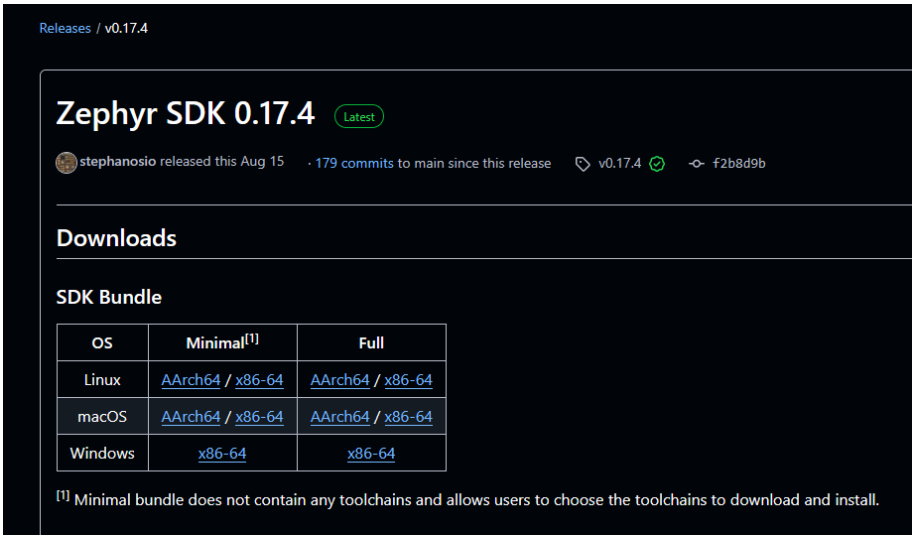



*Figure 31: Zephyr SDK download*

Extract the downloaded SDK and move in the extracted directory.
Open a command prompt and execute ***setup.cmd*** script.
Enter Y when prompted to accept installation of tools and packages.


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

### 5.1.4. Zephyr repository

Open *Visual Studio Code* and go to *MCUXpresso for VS Code* menu.
Click *import* repository and select ***NXP Zephyr*** repository.
This note is written with nxp zephyr revision 4.2.0 in mind.

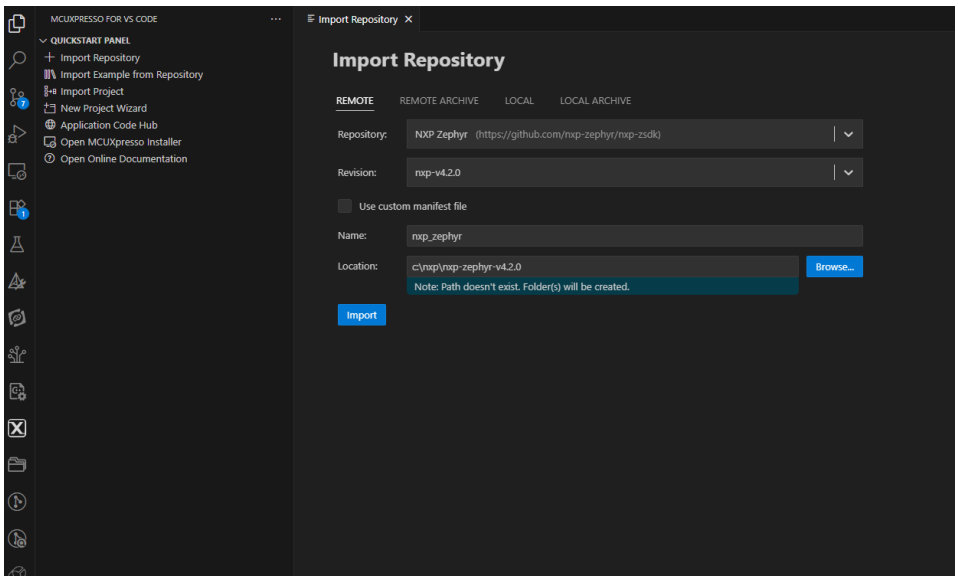



*Figure 32: Zephyr repository download*

Click ***import*** and wait for the installation to end.

## 5.2. Setup NXP Zephyr repository

We need to execute the following step in order to generate binaries compatible with ISP5261.

Go to https://www.insightsip.com/products/wi-fi-6-modules/isp5261 and download in Hardware & Firmware library section the following files:

- ISP5261_tb_device_tree_for_Zephyr.zip
- ISP5261_Zephyr_patch.zip

Extract ***isp5261_tb_device_tree_for_Zephyr.zip*** in <NXP repository Path>\zephyr\boards\nxp
It should look like this:


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

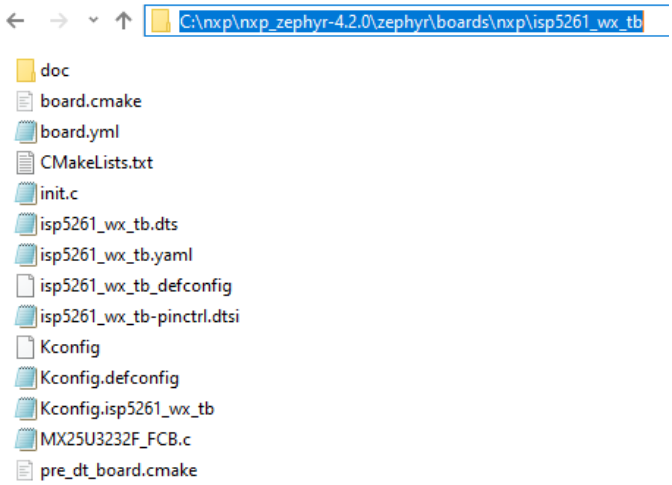



*Figure 33: ISP5261-WX-TB device tree*

Extract the file called ***module_hal_nx_patch_for_insightsip.patch*** from ISP5261_Zephyr_patch.zip and copy it to <NXP repository Path>\modules\hal\nxp.
Open a command prompt and execute: *git apply .\module_hal_nx_patch_for_insightsip.patch*

Extract the file called ***zephy_patch_for_insightsip.patch*** from ISP5261_Zephyr_patch.zip and copy it to <NXP repository Path>\zephyr.
Open a command prompt and execute: *git apply .\zephyr_patch_for_insightsip.patch*


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

## 5.3. Build and run a Wi-Fi application

This chapter explains how to build the Wi-Fi ***shell*** application. This application gives access to Wi-Fi functionalities through a serial terminal.

### 5.3.1. Build the project

Open *Visual Studio Code* and go to *MCUXpresso for VS Code* menu.
Click *Import Example from Repository*. Select in the new menu :

- **Repository** : Path where the Zephyr is installed
- **Board** : isp5261-wx-tb
- **Template** : zephyr/samples/net/wifi/shell

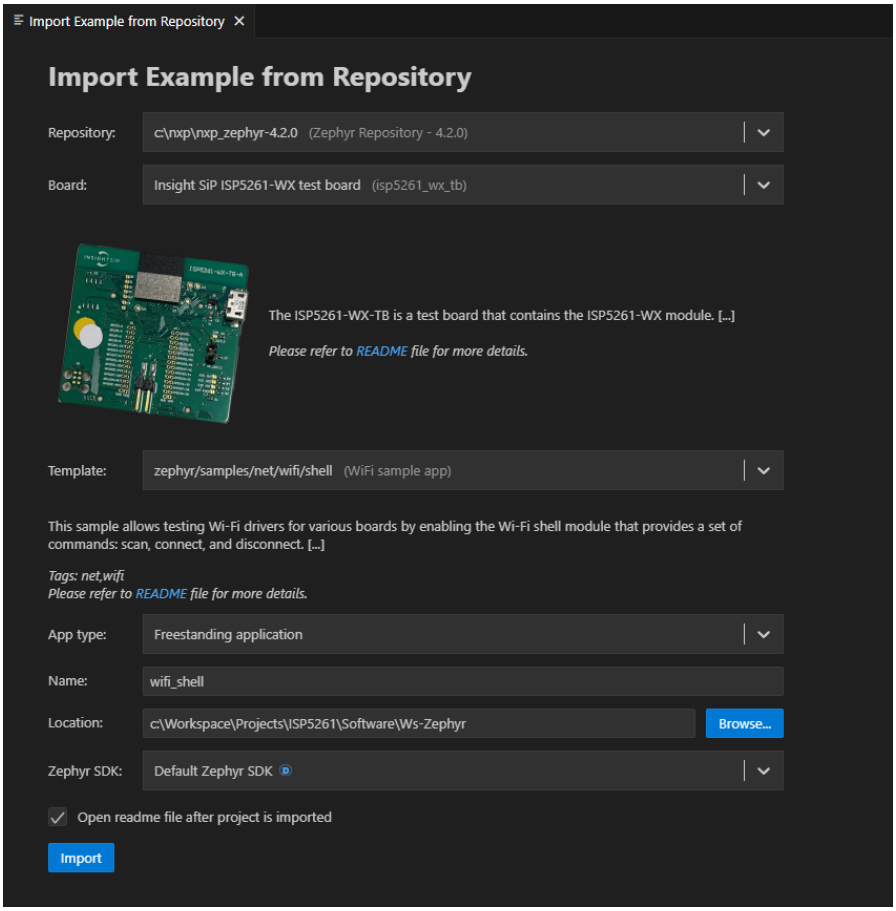



*Figure 34: Import Wi-Fi shell example*

Click ***Import*** and the project should be added to your project list.

Open the newly created shell project folder. For example, it can be located by right-clicking the shell project and then go to Project -> Reveal in File Explorer


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

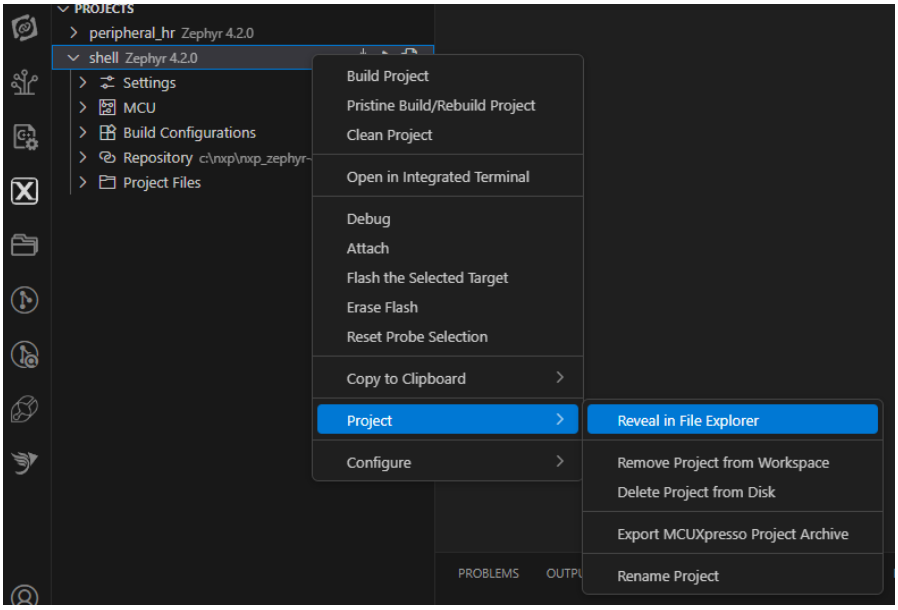



*Figure 35: Shell project location*

In the boards folder, rename the files ***frdm_rw612.conf*** to ***isp5261-wx-tb.conf*** and ***frdm_rw612.overlay*** to ***isp5261-wx-tb.overlay***.

It should look like this:

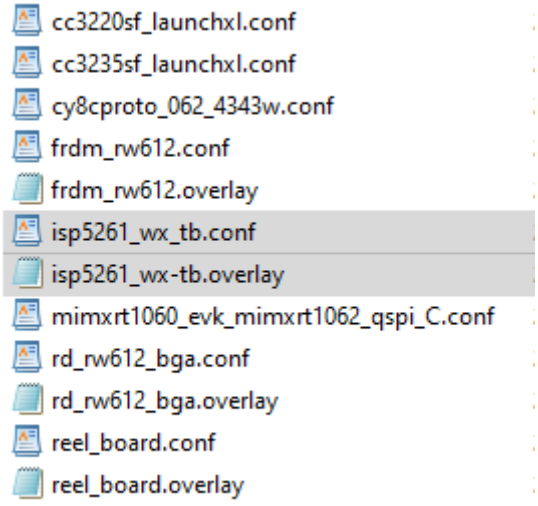



*Figure 36: shell/boards folder*

Now the project is ready to be built. Right-click the shell project and then select ***Build Project***.


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

### 5.3.2. Load and test the application

Right-click the shell project and then select ***Flash the selected target***.
With a serial terminal such as ***TeraTerm***, open a COM port with a baudrate of 115200.
You should see boot logs:

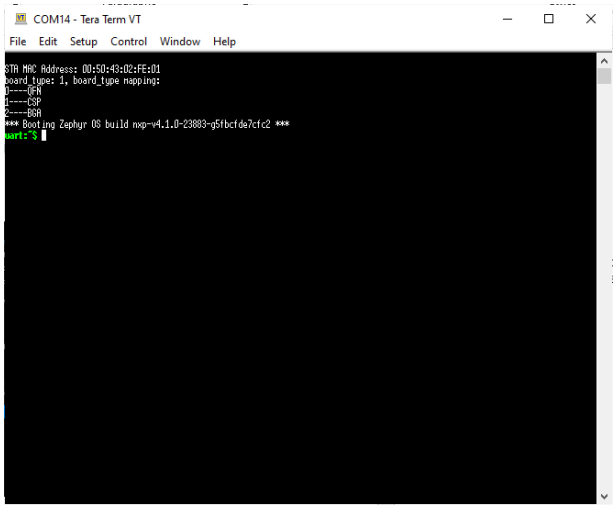



*Figure 37: Shell example boot logs*

If nothing appears try pressing the reset button on the ISP130603H board.
Type **?** to see a list of commands.
For example, type ***wifi scan*** to scan for access points.

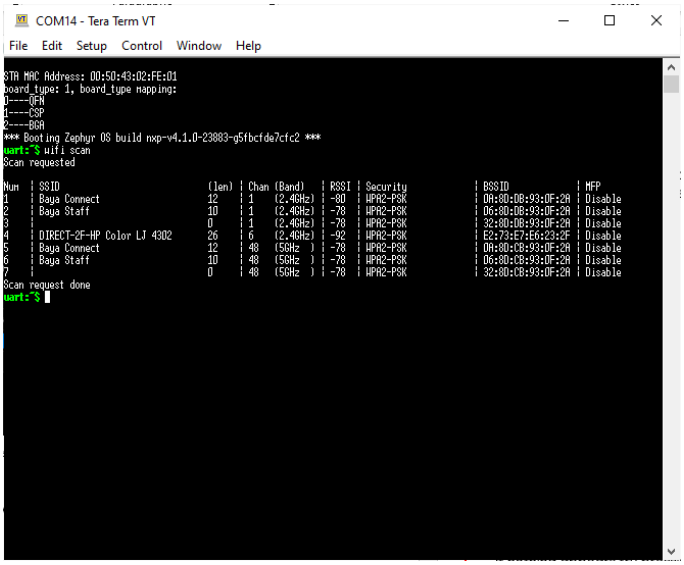



*Figure 38: scan with Shell example*


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

## 5.4. Build and run a BLE application

This chapter explains how to build the BLE ***peripheral_hr*** application. This application sends simulated heart rate value to a central device such as a smartphone.

### 5.4.1. Build the project

Open *Visual Studio Code* and go to *MCUXpresso for VS Code* menu.
Click *Import Example from Repository*. Select in the new menu :

- **Repository** : Path where the Zephyr is installed
- **Board** : isp5261-wx-tb
- **Template** : zephyr/samples/bluetooth/peripheral_hr

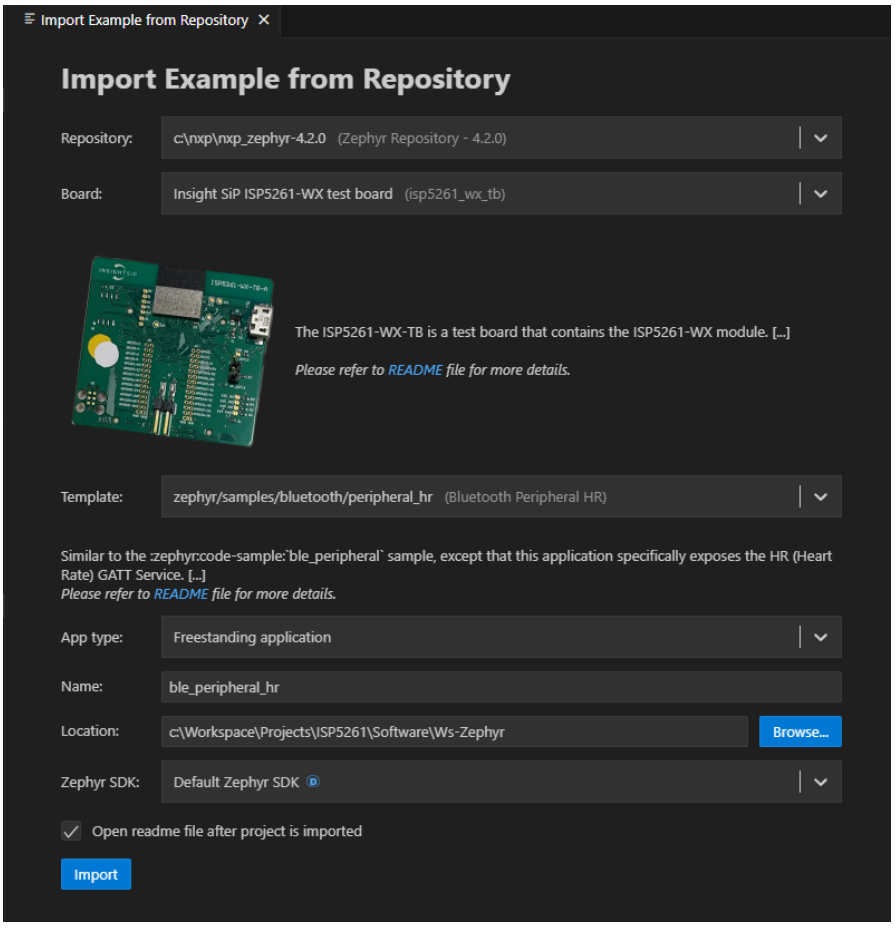



*Figure 39: Import BLE peripheral_hr example*

Click *Import* and the project should be added to your project list.

Now the project is ready to be built. Right-click the shell project and then select *Build Project*.


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.

### 5.4.2. Load and test the application

Right-click the shell project and then select ***Flash the selected target***.
With a serial terminal such as ***TeraTerm***, open a COM port with a baudrate of 115200.
You should see boot logs:

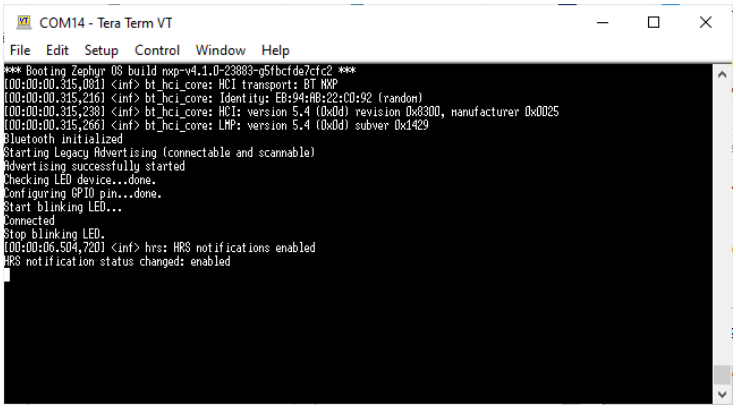



*Figure 40: peripheral_hr example boot logs*

Launch the ***IoT Toolbox*** application on your smartphone and select the Thermometer service. A device called ***peripheral_hr*** should be advertising, select it to connect.

Once connected, temperature readings show on the smartphone.

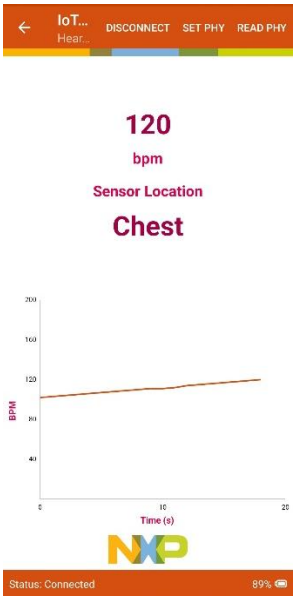



*Figure 41: HR application*


The information contained in this document is the property of Insight SiP and should not be disclosed to any third party without written permission. Specification subject to change without notice.