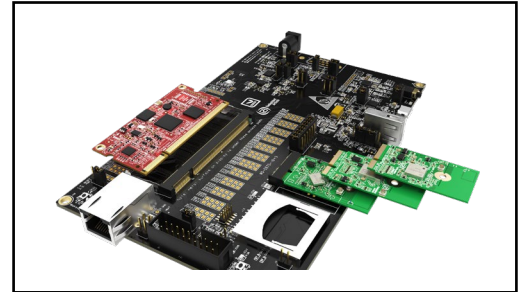


**Murata Wi-Fi/BT
(CYW) Solution for i.MX**

FreeRTOS User Guide



Revision History

Revision	Date	Author	Change Description
1.0	Nov 17, 2020	TF	Initial Release. NOTE: Material moved from previous Quick Start Guide. Added alternative option to download SDK in Section 6 . Added more sample application example in Section 7 . Removed i.MX RT1052 references.
1.1	Jan 28, 2021	TF	Added sample application example in Section 7 .

Table of Contents

Revision History	1
Table of Contents	2
1 Introduction	4
1.1 Acronyms	4
1.1 References	5
1.1.1 Murata Wi-Fi/BT (CYW) Solution for i.MX FreeRTOS Quick Start Guide	5
1.1.2 Murata Wi-Fi/BT Solution for i.MX Hardware User Manual	5
1.1.3 Murata's Community Forum Support	5
1.1.4 Murata uSD-M.2 Adapter Datasheet (Rev B1)	5
1.1.5 Murata uSD-M.2 Adapter Datasheet (legacy Rev A)	5
1.1.6 Murata's uSD-M.2 Adapter Landing Page	5
1.1.7 Embedded Artists' Reference Documentation	5
1.1.8 Murata's i.MX Wireless Solutions Landing Page	5
1.1.9 Murata's i.MX Wireless Solutions Landing Page	5
1.2 Hardware Options	6
2 Hardware Setup for NXP EVKs with uSD-M.2 Adapter	7
3 Hardware Setup for Embedded Artists Dev Kits via M.2 interface	9
4 Software Setup for NXP EVKs	10
5 Software Setup for Embedded Artists Dev Kits	11
6 SDK Setup for MCUXpresso	12
6.1 Drag and drop SDK in the IDE	12
6.2 Install SDK directly from MCUXpresso IDE	16
7 Running Sample Application	18
7.1 Example wiced_iperf_4343W	18
7.2 Example wiced_iperf_43012	26
7.3 Example wiced_mfg_test_4343W	34
7.4 Example wiced_mfg_test_43012	39
7.5 Example wiced_bt_passthrough_4343W	43
7.6 Example wiced_ble_4343W	49
7.7 Example wiced_webconfig_4343W	53
8 Murata's uSD-M.2 Adapter	62
8.1 Connecting the Wi-Fi/BT M.2 EVB to uSD-M.2 Adapter	62
8.2 Configuring uSD-M.2 Adapter Jumpers for Correct VIO Signaling	62
8.3 Securing uSD-M.2 Adapter to NXP i.MX RT10xx EVK	63
8.4 uSD-M.2 Adapter High-Level Description	64
9 Technical Support Contact	67
10 Additional Useful Links	67

LIST OF FIGURES

Figure 1: Connecting the EVB to the EVK	7
Figure 2: Connected setup close up	8
Figure 3: NXP i.MX RT 1050-EVKB with Type 1DX M.2 & uSD-M.2 Adapter.....	8
Figure 4: Embedded Artists i.MX RT 1062 Developer's Kit	9
Figure 5: NXP Getting Started Web Page	10
Figure 6: COM Port of NXP IMXRT1050-EVKB	10
Figure 7: Embedded Artists Resource Web.....	11
Figure 8: COM Port and JTAG of Embedded Artists Kit.....	11
Figure 9: NXP i.MX RT 1050 EVK hardware settings to run BT passthrough.....	43
Figure 10: NXP i.MX RT 1050 EVK hardware settings to run BLE.....	49
Figure 11: Additional cabling pins	53
Figure 12: Additional cabling.....	53
Figure 13: Connecting the Wi-Fi/BT M.2 EVB to uSD-M.2 Adapter	62
Figure 14: Host/M.2 IO Voltage Level Shift Options on Rev B1 Adapter.....	63
Figure 15: Securing uSD-M.2 Adapter to i.MX RT10xx EVK.....	63
Figure 16: uSD-M.2 Adapter Features (Top View)	65
Figure 17: uSD-M.2 Adapter Features (Bottom View).....	66

LIST OF TABLES

Table 1: Acronyms used in Quick Start Guide	4
Table 2: Embedded Artists Documentation Listing	6
Table 3: Murata module support on NXP and Embedded Artists' platforms	6
Table 4: Files to be modified on Embedded Artists' Dev Kit	20
Table 5: Additional connections to run BT passthrough	43
Table 6: Additional connections to run BLE	49
Table 7: uSD-M.2 Adapter Features	64
Table 8: List of Support Resources.....	67
Table 9: NXP links.....	67
Table 10: Embedded Artists' Landing Pages.....	68
Table 11: Embedded Artists' Datasheets and Schematics.....	68
Table 12: Embedded Artists' User Manuals and Software	68

1 Introduction

This document provides detailed instructions on adding wireless functionality with Embedded Artists' Wireless [M.2 Modules](#) on [NXP Semiconductor's](#) i.MX RT 1020/1050/1060/1064 Evaluation Kit and [Embedded Artists AB's](#) i.MX RT 1062 Developer's kit. NXP's latest SDK supports 1DX and 1LV M.2 modules. The SDK supports both Wi-Fi and Bluetooth for all the i.MX RT boards (except i.MX RT 1064 EVK where BLE is not functional). Murata provides a patch release for supporting Type 1MW – please go to Murata's Forum. This chapter explains how to set up the hardware and the steps required for software installation.

1.1 Acronyms

Table 1: Acronyms used in Quick Start Guide

Acronym	Meaning
BT	Bluetooth
CYW	Cypress
EA	Embedded Artists designs, manufactures and distributes current Wi-Fi/BT M.2 EVB's (link here). EA also have enhanced i.MX developer kits which provide comprehensive support for Murata modules (link here).
EULA	End User License Agreement
EVB	Evaluation Board (Embedded Artists' Wi-Fi/BT module)
EVK	Evaluation Kit
FTDI	Future Technology Devices International
IDE	Integrated Development Environment
JTAG	Joint Test Action Group
M.2	Formerly known as the Next Generation Form Factor (NGFF), is a specification for internally mounted computer expansion cards and associated connectors. The M.2 specification is defined by PCI-SIG (www.pcisig.com).
PC	Personal Computer
RF	Radio Frequency
RTOS	Real-time Operating System
RX	Receive
SD	Secure Digital
SDIO	Secure Digital Input Output
SDK	Software Development Kit
TX	Transmit
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
uSD	Micro SD
uSD-M.2	Micro SD to M.2 Adapter
Wi-Fi	Wireless LAN: "Wi-Fi" is a registered trademark of Wi-Fi Alliance
WLAN	Wireless Local Area Network

1.1 References

1.1.1 Murata Wi-Fi/BT (CYW) Solution for i.MX FreeRTOS Quick Start Guide

This [Quick Start Guide](#) provides quick steps to get started with Murata Wi-Fi/BT Cypress chipset-based solution with the help of an example.

1.1.2 Murata Wi-Fi/BT Solution for i.MX Hardware User Manual

This [manual](#) describes the Murata uSD-M.2 Adapter hardware. All interface signals to the NXP i.MX RT, 6, 7, and 8 EVK's are described. Specifics on interfacing each i.MX EVK to Murata uSD-M.2 Adapter are provided.

1.1.3 Murata's Community Forum Support

Murata's Community provides online support for the Murata Wi-Fi/Bluetooth modules on various i.MX platforms. Refer to [this link](#) for main Forum Wi-Fi and Bluetooth landing page.

1.1.4 Murata uSD-M.2 Adapter Datasheet (Rev B1)

This [datasheet](#) documents the current version of the Murata' latest uSD-M.2 adapter hardware and its interfacing options.

1.1.5 Murata uSD-M.2 Adapter Datasheet (legacy Rev A)

This [datasheet](#) documents the current version of the Murata's legacy uSD-M.2 adapter hardware and its interfacing options. This adapter version is no longer manufactured.

1.1.6 Murata's uSD-M.2 Adapter Landing Page

This [website landing page](#) provides latest/comprehensive information on Murata's adapter including links to where it can be purchased.

1.1.7 Embedded Artists' Reference Documentation

Embedded Artists designed the 1DX/1MW/1LV M.2 EVB's in close collaboration with Murata. It is **important to note** that Embedded Artists manufactures and distributes the Wi-Fi/BT M.2 EVB's. Refer to this main landing page for more information: www.embeddedartists.com/m2. **Table 2** lists some relevant documents published by Embedded Artists.

1.1.8 Murata's i.MX Wireless Solutions Landing Page

This [website landing page](#) provides latest/comprehensive information on Murata's i.MX Wireless solutions which use the uSD-M.2 Adapter as a key enabler so customers can easily evaluate Murata's modules on i.MX processors.

1.1.9 Murata's i.MX Wireless Solutions Landing Page

This [website landing page](#) provides latest/comprehensive information on Murata's i.MX Wireless solutions which use the uSD-M.2 Adapter as a key enabler so customers can easily evaluate Murata's modules on i.MX processors.

Table 2: Embedded Artists Documentation Listing

Documentation Filename	Note
Wi-Fi/BT M.2 EVB Primer	Introduction and drill-down on M.2 interface
M.2 SDIO Interface Schematic	Reference schematic for customers designing in WLAN-SDIO M.2 EVB.
M.2 PCIe Interface Schematic	Reference schematic for customers designing in WLAN-PCIe M.2 EVB.
1DX M.2 Module Datasheet	Comprehensive details on 1DX Wi-Fi/BT M.2 Module.
1MW M.2 Module Datasheet	Comprehensive details on 1MW Wi-Fi/BT M.2 Module.
1LV M.2 Module Datasheet	Comprehensive details on 1LV Wi-Fi/BT M.2 Module.

1.2 Hardware Options

This section describes how to configure the hardware correctly, for both NXP and Embedded Artists i.MX RT series platforms. The following table lists the different platforms supported by Murata modules.

Table 3: Murata module support on NXP and Embedded Artists' platforms

i.MX EVK / Dev kit	Manufacturer	Part number	Murata modules supported	Interconnect
i.MX RT1064 EVK	NXP	MIMXRT1064-EVK	1DX , 1MW ¹ , 1LV ²	uSD-M.2 Adapter
i.MX RT1060 EVK	NXP	MIMXRT1060-EVK	1DX , 1MW ¹ , 1LV ²	uSD-M.2 Adapter
i.MX RT1050 EVK	NXP	IMXRT1050-EVKB	1DX , 1MW ¹ , 1LV ²	uSD-M.2 Adapter
i.MX RT1020 EVK	NXP	MIMXRT1020-EVK	1DX , 1MW ¹ , 1LV ²	uSD-M.2 Adapter
i.MX RT106A	NXP	SLN-ALEXA-IOT	1DX	Soldered down
iMX RT1062 Dev Kit	Embedded Artists	EAK00310	1DX , 1MW , 1LV	M.2

NOTE: The Embedded Artists' iMX RT1062 Developer's Kit can be used instead of previous (now discontinued) iMX RT1052 Dev Kit for development/evaluation purposes. The main difference between the iMX RT1052 and the iMX RT1062 is that the iMX RT1062 has 1 MB of SRAM while the iMX RT1052 has 0.5 MB SRAM. The iMX RT1052 OEM COM is available for order [here](#).

¹ Driver support is present, example code to be added. Post on Murata Community Forum for assistance.

² Currently 1LV only supports Wi-Fi examples. Bluetooth example is not included yet. Go to Forum for assistance.

Several toolchains are supported by NXP as below, but MCUXpresso IDE is the primary focus in this document:

- NXP supports MCUXpresso IDE
- GNU toolchain for Arm® Cortex® -M with Cmake build system
- IAR Embedded Workbench
- Keil™ MDK-Arm

2 Hardware Setup for NXP EVKs with uSD-M.2 Adapter

To enable Murata's wireless solution on NXP's i.MX RT 1020/1050/1060/1064 Evaluation Kits, Embedded Artists' Wi-Fi/BT M.2 EVB's (Murata module onboard) connected to Murata's uSD-M.2 Adapter. The on-board debug adapter is supported. Refer to **Figure 1**, **Figure 2** and **Figure 3** for example of i.MX RT1050 EVK. Murata's uSD-M.2 Adapter plugs in directly to the EVK's microSD connector. The micro-USB connector (J28) is used for USB-UART/JTAG.

Refer to **Section 8** on how to correctly connect Embedded Artists' Wi-Fi/BT M.2 EVB to the Murata Adapter and how to properly jumper the Adapter for default 1.8V VIO operation (not 3.3V override mode). Now insert the Murata Adapter into the microSD slot (J20) until you hear the click sound (push-push connector). Per **Section 8.3**, it is best to **tape** the uSD Adapter-microSD connection. Make sure the green LED (LED1) on the adapter board is illuminated when powered. Also, the blue LED (LED2) should **not be** illuminated. Repeating the Murata uSD-M.2 Adapter jumper settings:

- For rev B1 adapter, J12 is in 1-2 pos & J13 is in 1-2 pos.
- For (legacy) rev A adapter, J12 is open.

Figure 1: Connecting the EVB to the EVK

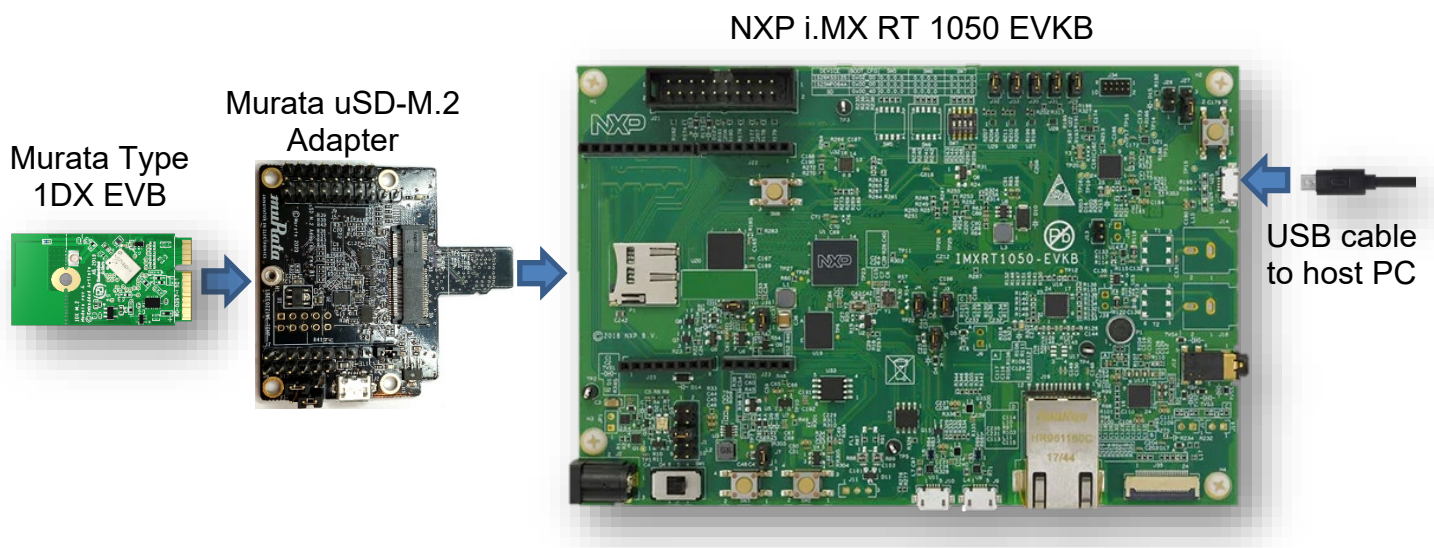


Figure 2: Connected setup close up

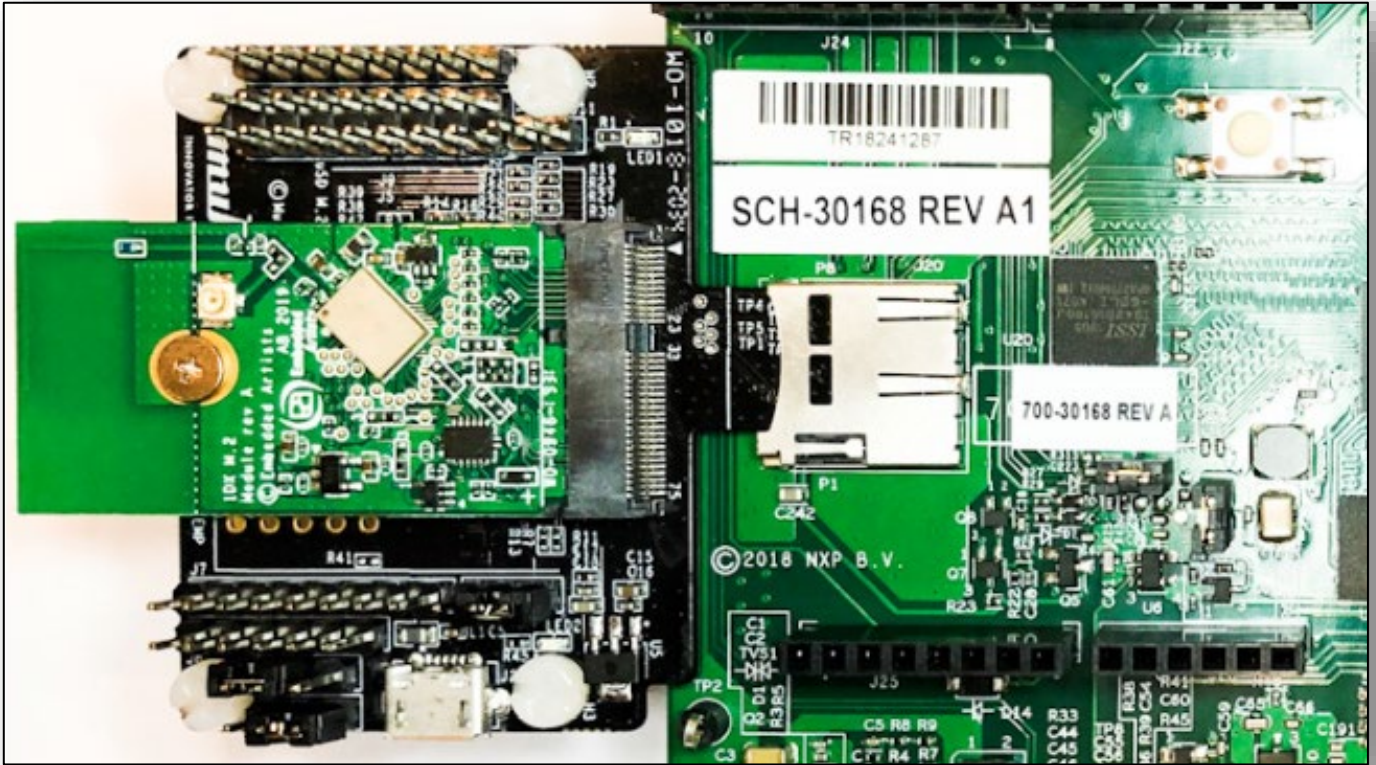
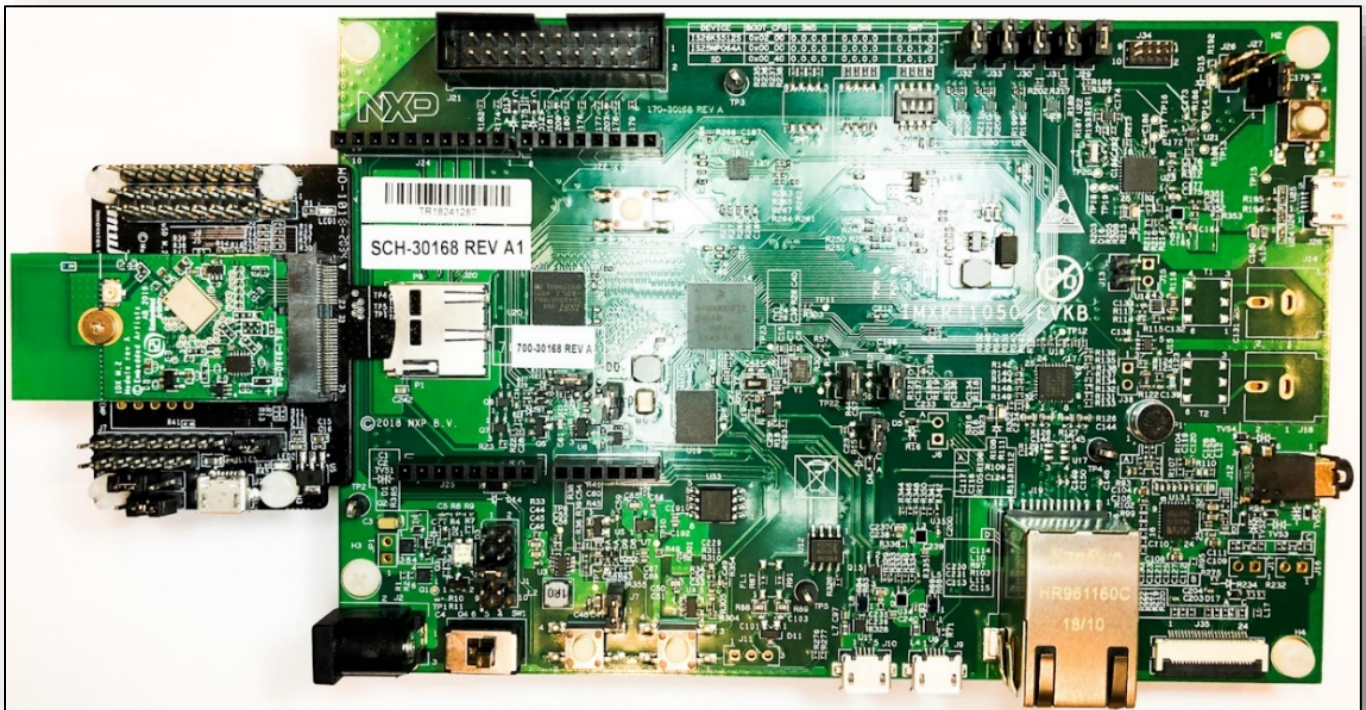


Figure 3: NXP i.MX RT 1050-EVKB with Type 1DX M.2 & uSD-M.2 Adapter

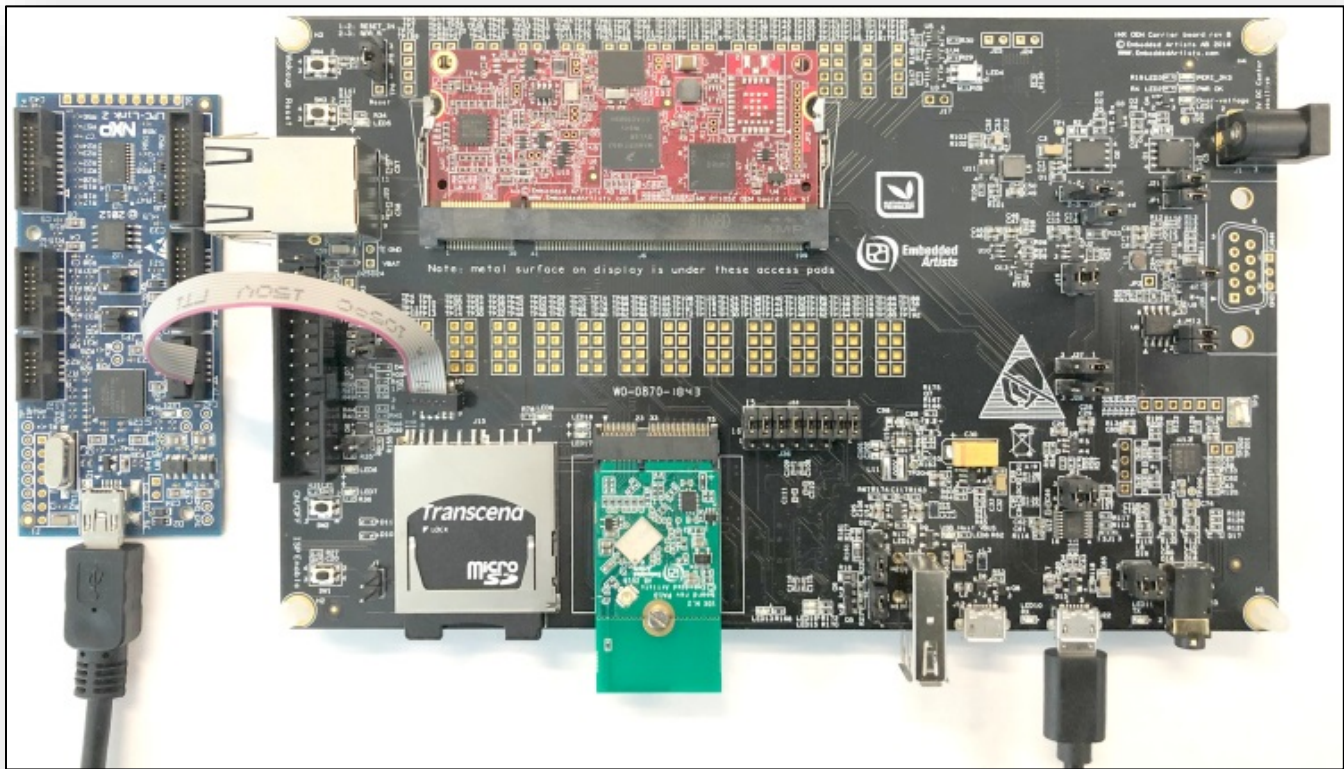


3 Hardware Setup for Embedded Artists Dev Kits via M.2 interface

Embedded Artists' i.MX RT 1062 Developer Kit has a M.2 connector onboard for direct connection to the M.2 EVB (no adapter required). LPC-Link2 is recommended for the debug adapter. Embedded Artists' website provides support package. **Figure 4** shows the full connection of developer's kit with M.2 and debug probe. Micro-USB connector (J22) is used for USB-UART. J10 is used for LPC-Link2 connection.

NOTE: The red line in the flex cable used to connect the debug probe to the developer's kit should align with the arrow at J10 (i.e., pin 1 of J10).

Figure 4: Embedded Artists i.MX RT 1062 Developer's Kit



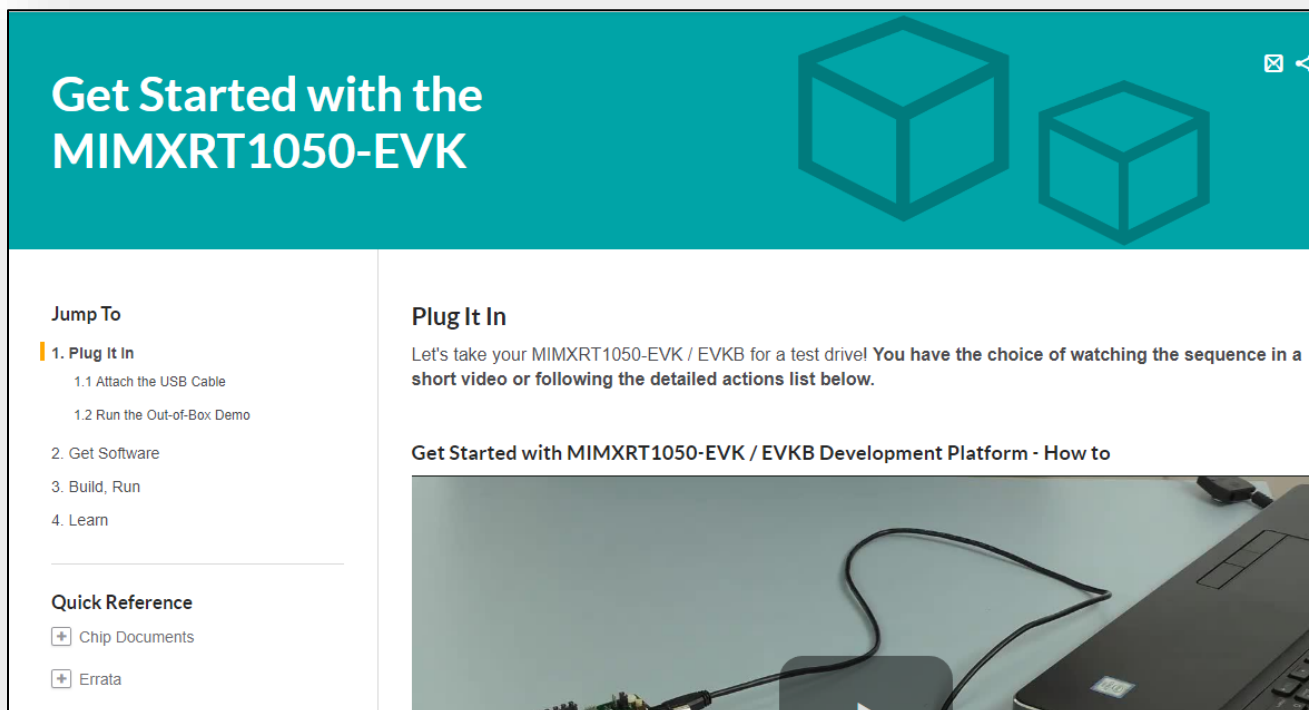
4 Software Setup for NXP EVKs

If you are using Embedded Artists Board, please skip this section and go to **Section 5** of this document.

Click [here](#) to go the NXP landing page as shown in **Figure 5**. Follow the steps described in NXP web to install these tools:

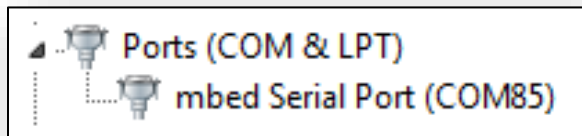
- MCUXPresso IDE
- mbed Virtual COM Port Driver
- Terminal application (TeraTerm, Putty, etc.)

Figure 5: NXP Getting Started Web Page



You should now be able to see mbed Serial Port on the Device Manager as shown in **Figure 6** if you completed software installation successfully.

Figure 6: COM Port of NXP IMXRT1050-EVKB



For installing the required SDK(s) for MCUXpresso IDE, please refer to **Section 6**.

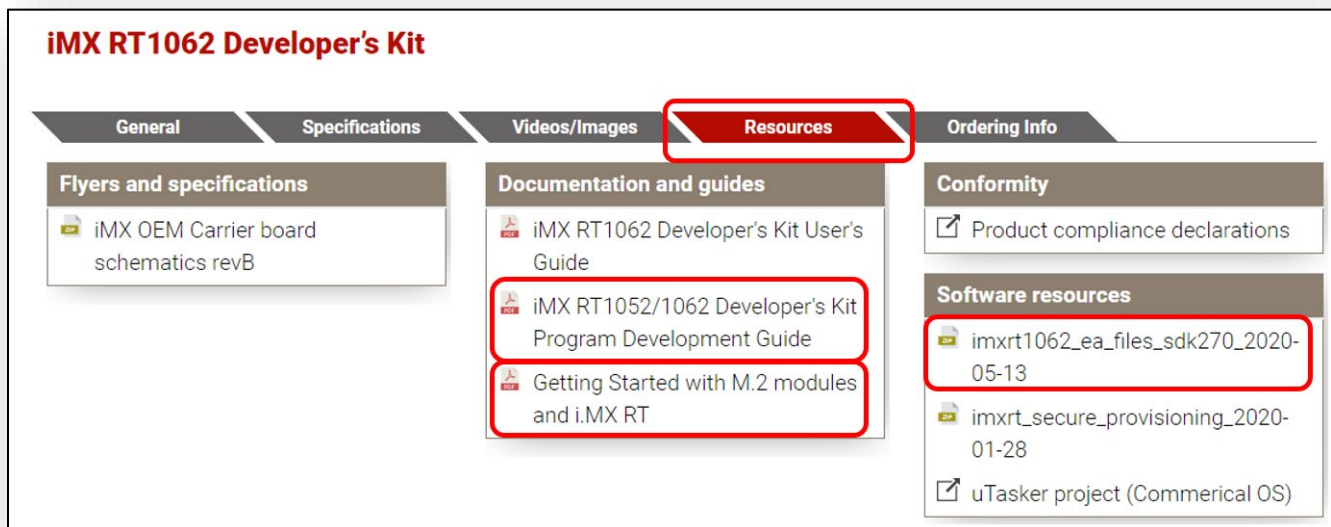
5 Software Setup for Embedded Artists Dev Kits

If you are using NXP Board, please skip this section and go back to **Section 4** of this document.

Click [here](#) to go the EA landing page as shown in **Figure 7**. Download the document [iMX RT1052/1062 Developer's Kit Program Development Guide](#) from the resource tab for detailed instructions about downloading and setting up the SDK. Download “imxrt10xx_ea_files_sdk270_yyyy-mm-dd.zip” for the SDK setup. Go to **Section 6** for more detail. You can also download “Getting Started with M.2 modules and i.MX RT” to run the sample example for the wiced_iperf. Also install all the tools mentioned below:

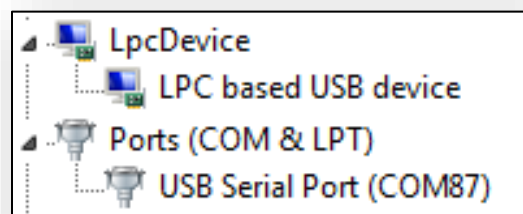
- MCUXpresso IDE
- FTDI Virtual COM Port Driver
- Terminal application (TeraTerm, Putty, etc.)

Figure 7: Embedded Artists Resource Web



You should see LPC based USB device and USB Serial Port on the Device Manager as below if you completed software installation successfully. The driver for LPC-Link2 is included in the MCUXpresso.

Figure 8: COM Port and JTAG of Embedded Artists Kit

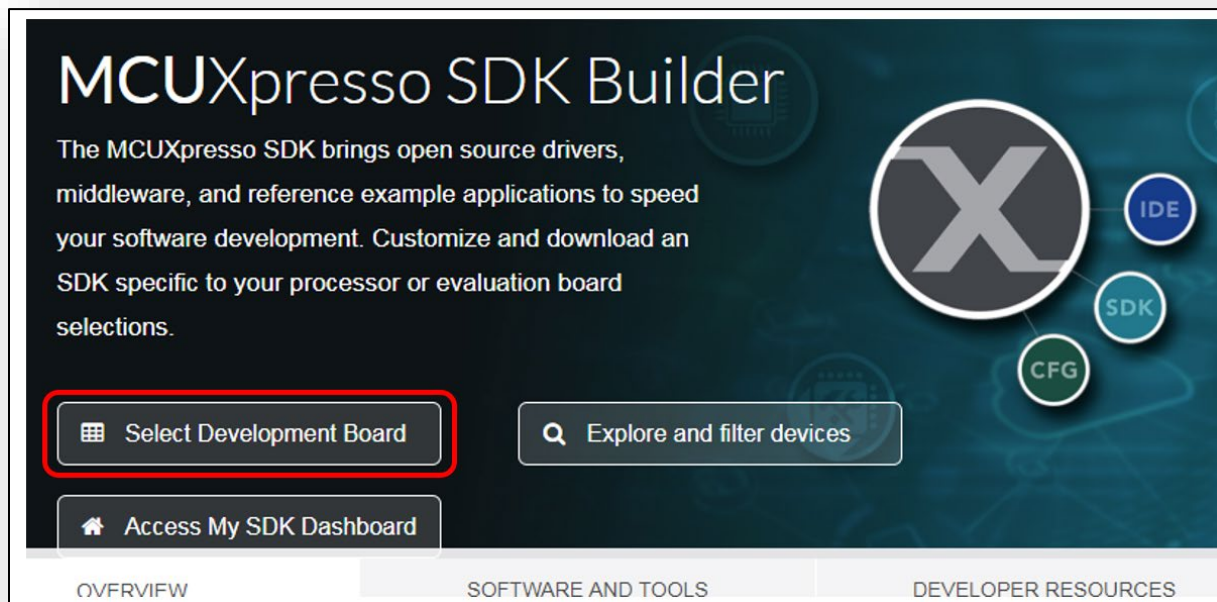


6 SDK Setup for MCUXpresso

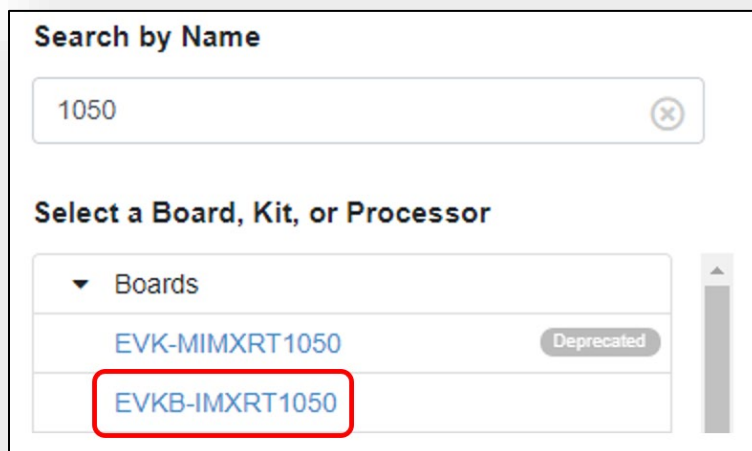
MCUXpresso supports various processors, so it requires appropriate SDK for i.MX RT. To support Embedded Artists' Wireless M.2 Modules, additional components (wifi_wiced and other related components) are required. There are two ways to install SDK in the new MCUXpresso IDE. Follow the steps below carefully to install right components.

6.1 Drag and drop SDK in the IDE

1. First download the MCUXpresso SDK by following this URL: <https://mcuxpresso.nxp.com>. You will need to login to your NXP account and then click "Select Development Board".



2. Type "1050", then select "EVKB-IMXRT1050" for i.MX RT 1050 EVKB.



3. Scroll down, then click “Build MCUXpresso SDK”

Hardware Details

Board	EVKB-IMXRT1050
Device	MIMXRT1052
Core Type / Max Freq	Cortex-M7F / 600MHz
Device Memory Size	0 KB Flash 512 KB RAM

Actions

- Build MCUXpresso SDK**
- Explore selection with Pins tool
- Explore selection with Clocks tool

4. Click “Select All” and then select “Download SDK” to download the SDK.

SDK Builder

Generate a downloadable SDK archive for use with desktop MCUXpresso Tools.

Developer Environment Settings
Selections here will impact files and examples projects included in the SDK and Generated Projects

SDK Version: 2.7.0 2019-12-19 | Toolchain / IDE: MCUXpresso IDE | Host OS: Windows

Search Name, Category, or Description... **Select All** **Unselect All**

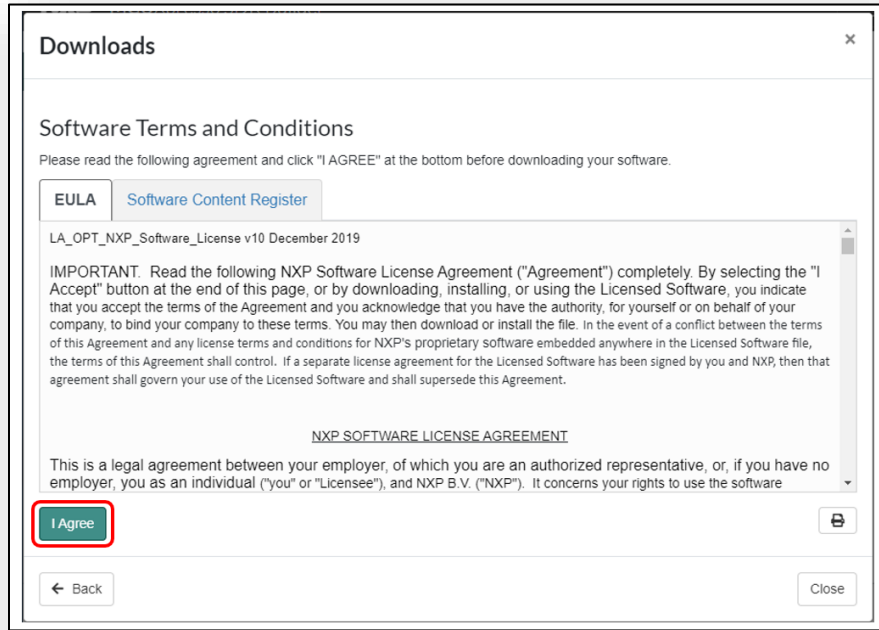
Name	Category	Description	Dependencies
<input checked="" type="checkbox"/> CMSIS DSP Library		CMSIS DSP Software Library	
<input checked="" type="checkbox"/> AWS IoT	Middleware	AWS IoT	
<input type="checkbox"/> Azure IoT	Middleware	Azure IoT SDK	
<input type="checkbox"/> canopen	Middleware	CANopen Stack - MicroCANopen Plus	
<input type="checkbox"/> cJSON	Middleware	cJSON library	
<input type="checkbox"/> Crank Storyboard GUI	Middleware	Crank Storyboard GUI Engine	
<input type="checkbox"/> eiQ	Middleware	eiQ machine learning SDK containing the ARM CMSIS-NN library (neural network ker... (more)	
<input type="checkbox"/> Embedded	Middleware	Embedded Wizard GUI	

This MCUXpresso SDK configuration is available for direct download

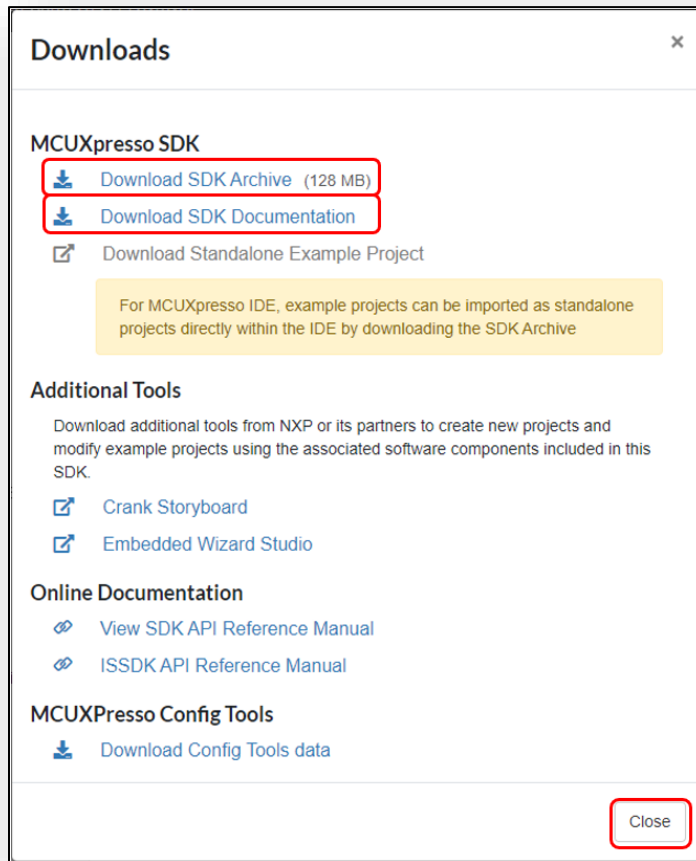
Download SDK

Archive Name: SDK_2.7.0_EVKB-IMXRT1050
Don't use: <> : ; / , | ? * * in the name of your SDK

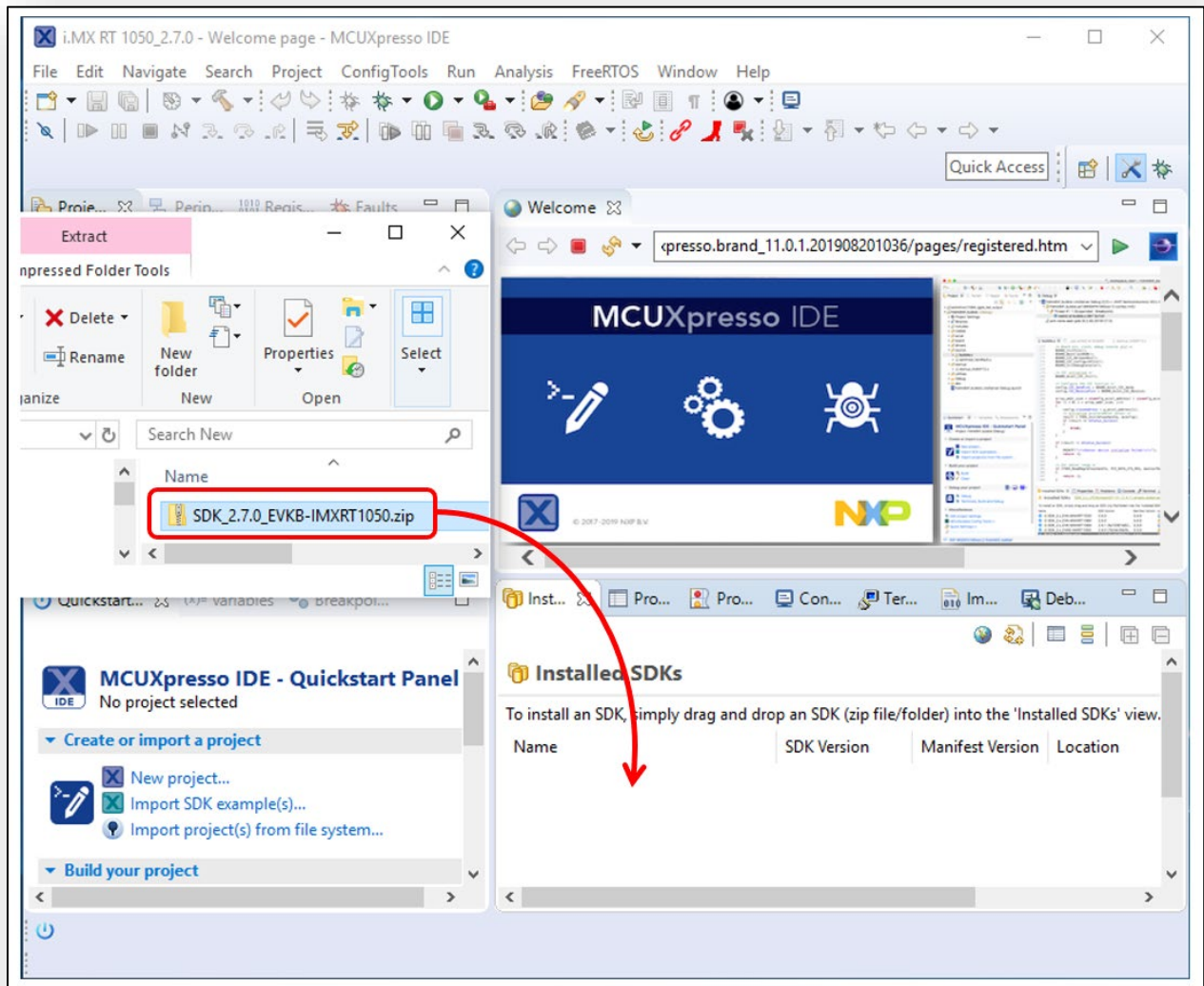
5. Agree to the EULA to start the SDK download.



6. Click "Download SDK Archive" if download does not start automatically. You can also download SDK Documentation, then click "Close".

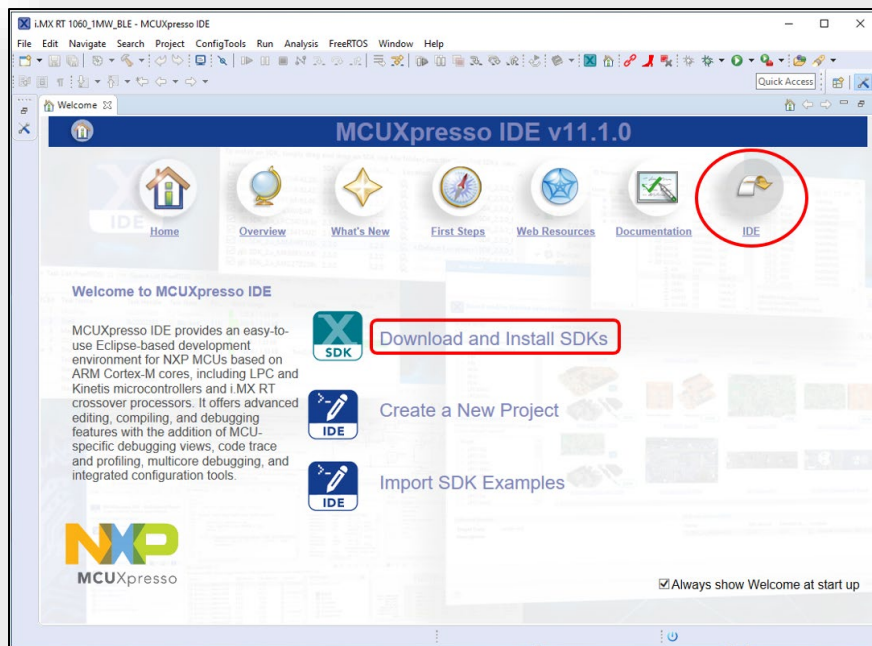


7. To install the SDK in the MCUXpresso, drag and drop the SDK Archive file on “Installed SDKs window”.

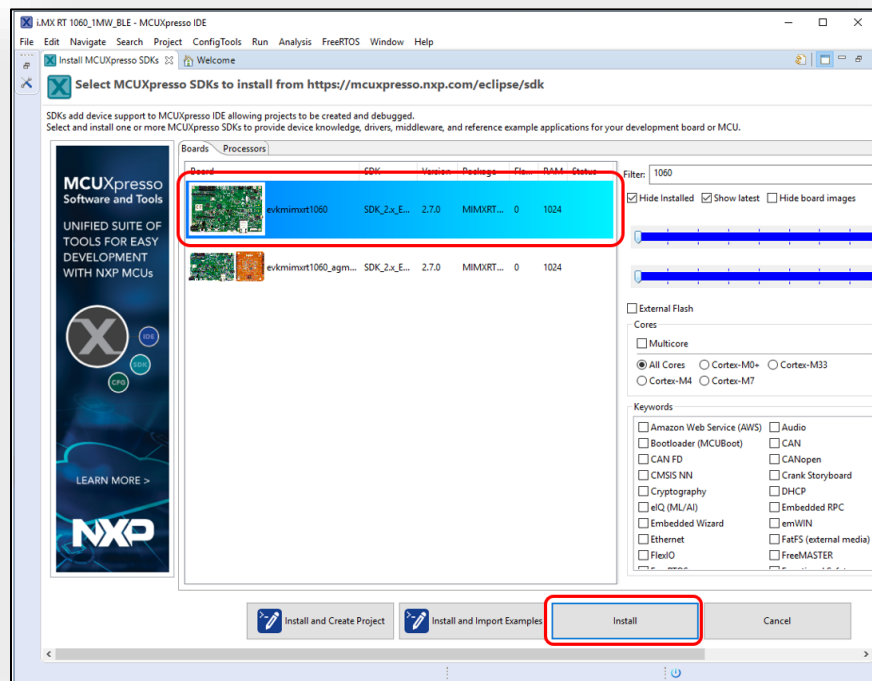


6.2 Install SDK directly from MCUXpresso IDE

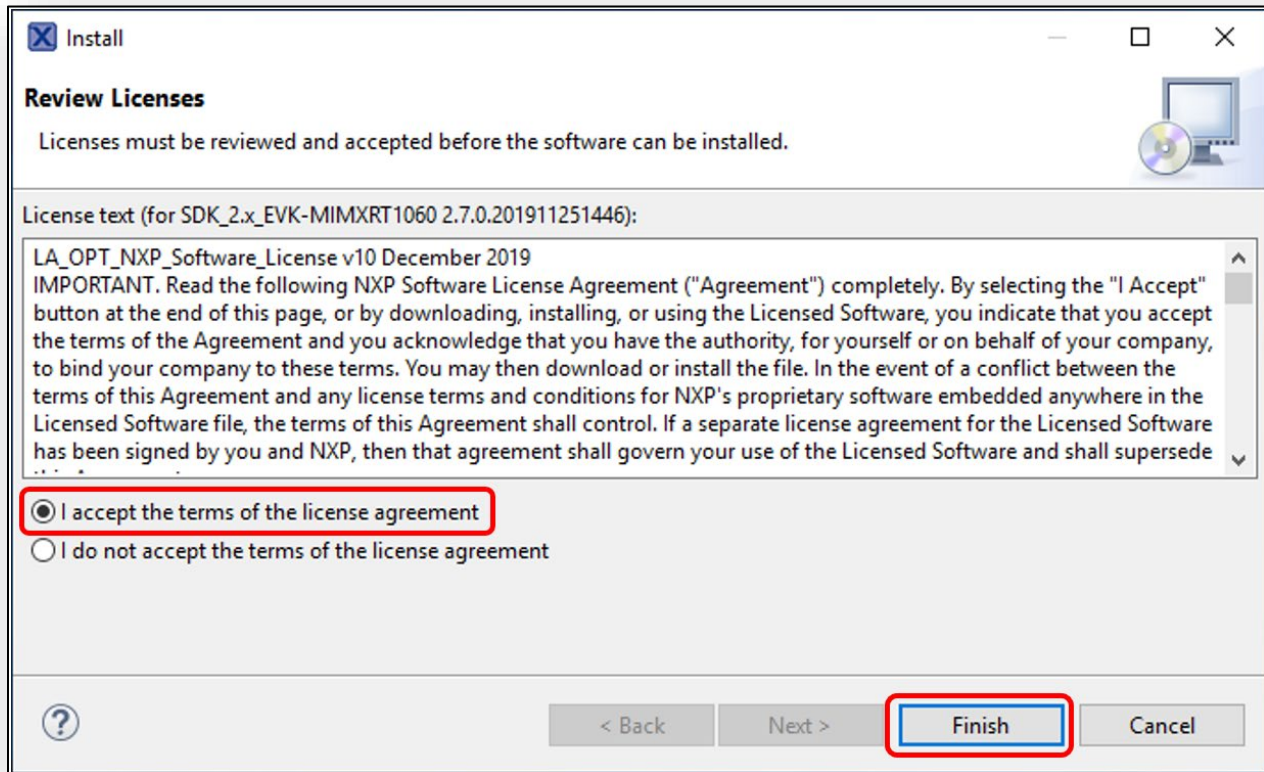
1. To install the SDK in the MCUXpresso IDE, click on “Download and Install SDKs”. To switch to normal IDE, click on IDE.



2. Type “1060” in the filter box. Click on “evkmimxrt1060” and click on “install” to download and install the SDK for i.MX RT 1060 EVK.



3. Accept the license agreement and click on finish to start the download.



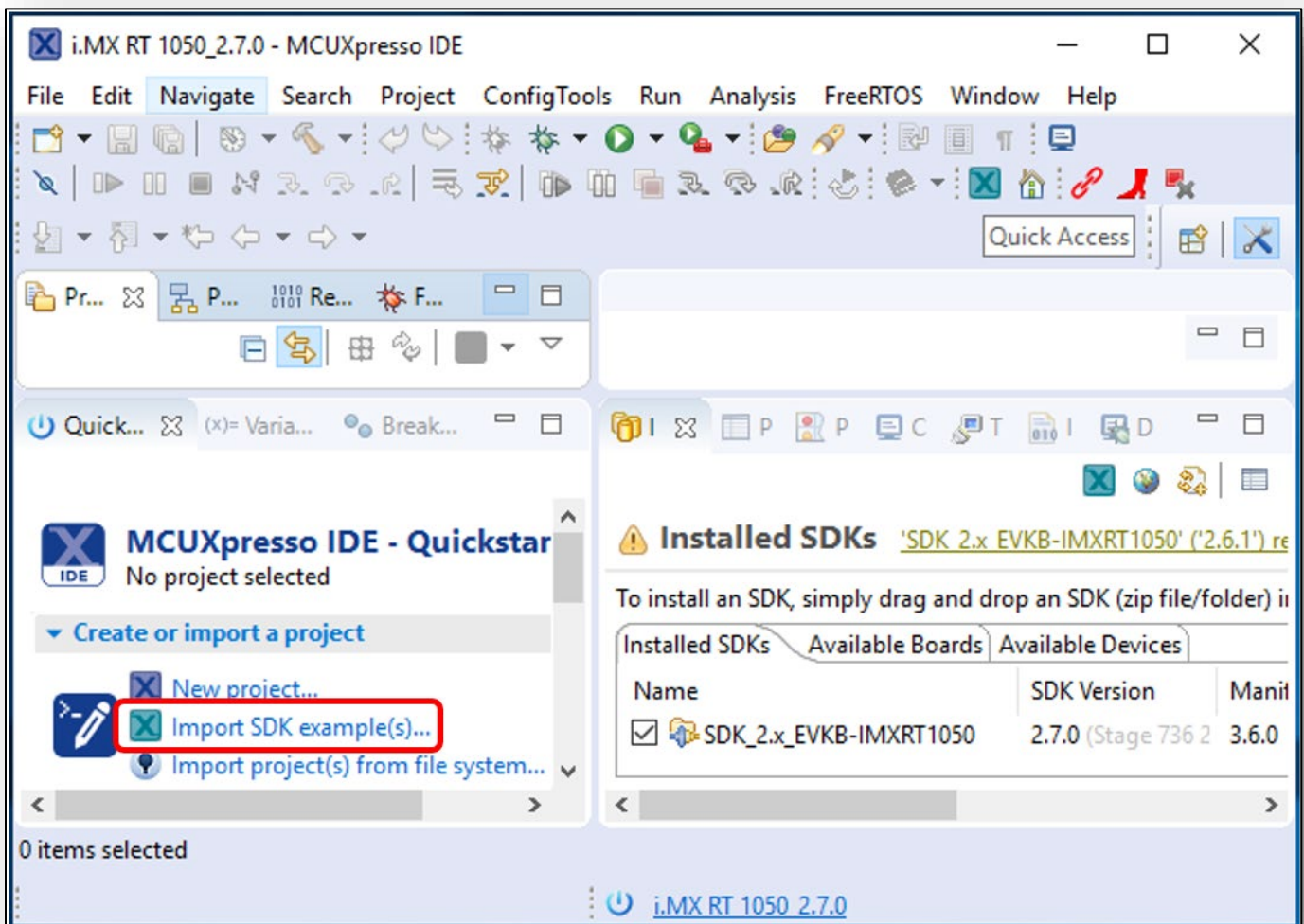
7 Running Sample Application

Various sample applications are provided by the SDK. There are four Wi-Fi examples included in the latest SDK:

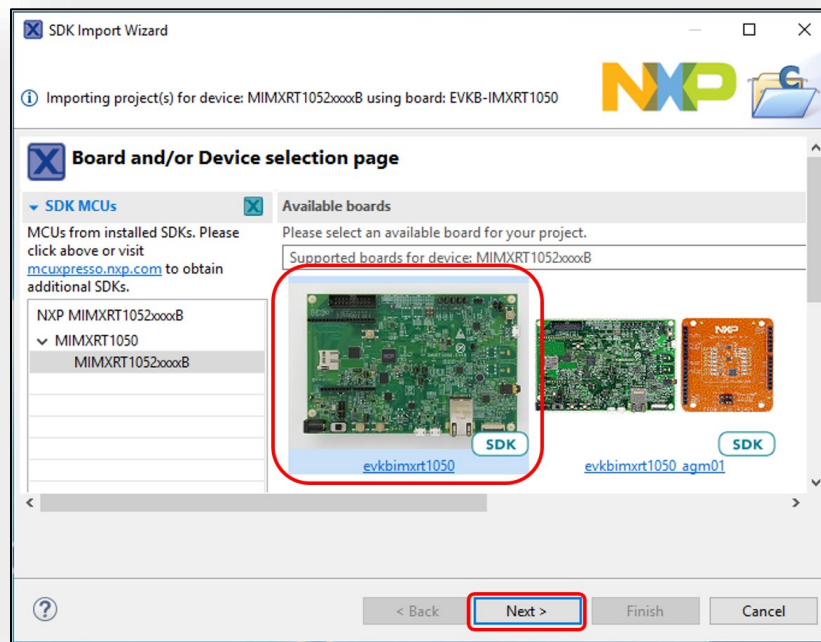
- wiced_iperf_4343W: this is for performance test for 1DX (CYW4343W).
- wiced_iperf_43012: this is for performance test for 1LV (CYW43012).
- wiced_mfg_test_4343W: this is for RF measurements and Regulatory testing for 1DX.
- wiced_mfg_test_43012: this is for RF measurements and Regulatory testing for 1LV.
- wiced_bt_passthrough_4343W: this is for Bluetooth testing for 1DX.
- wiced_ble_4343W: this is for BLE testing for 1DX.
- wiced_webconfig_4343W: this is for STA + SoftAP with a web server testing for 1DX.

7.1 Example wiced_iperf_4343W

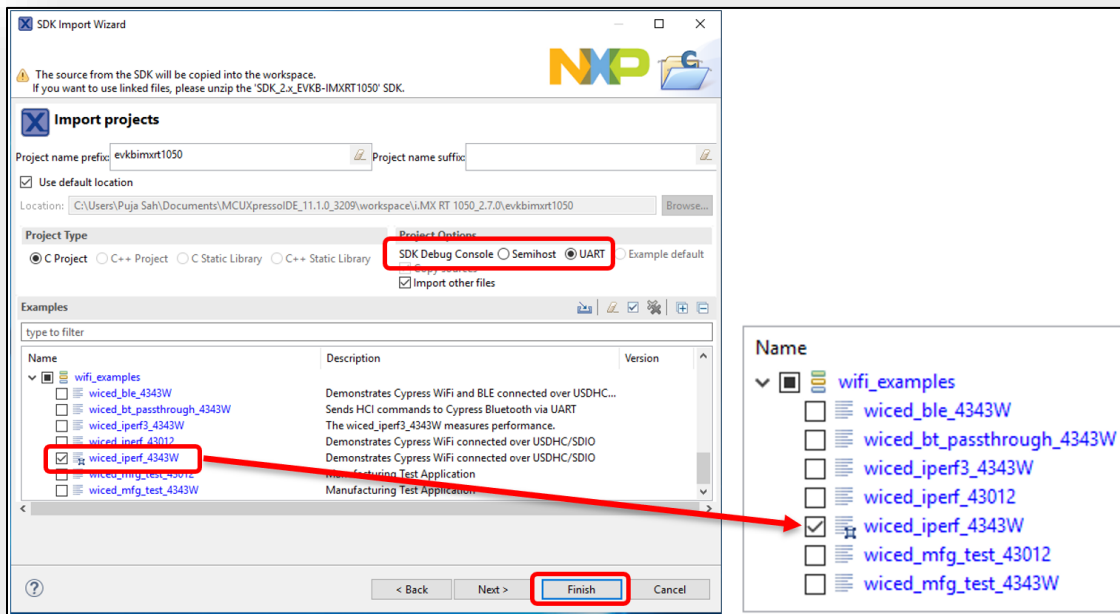
1. Click on “Import SDK example(s)...” in the Quickstart Panel.



2. Select “evkbimxrt1050” board and click Next button.

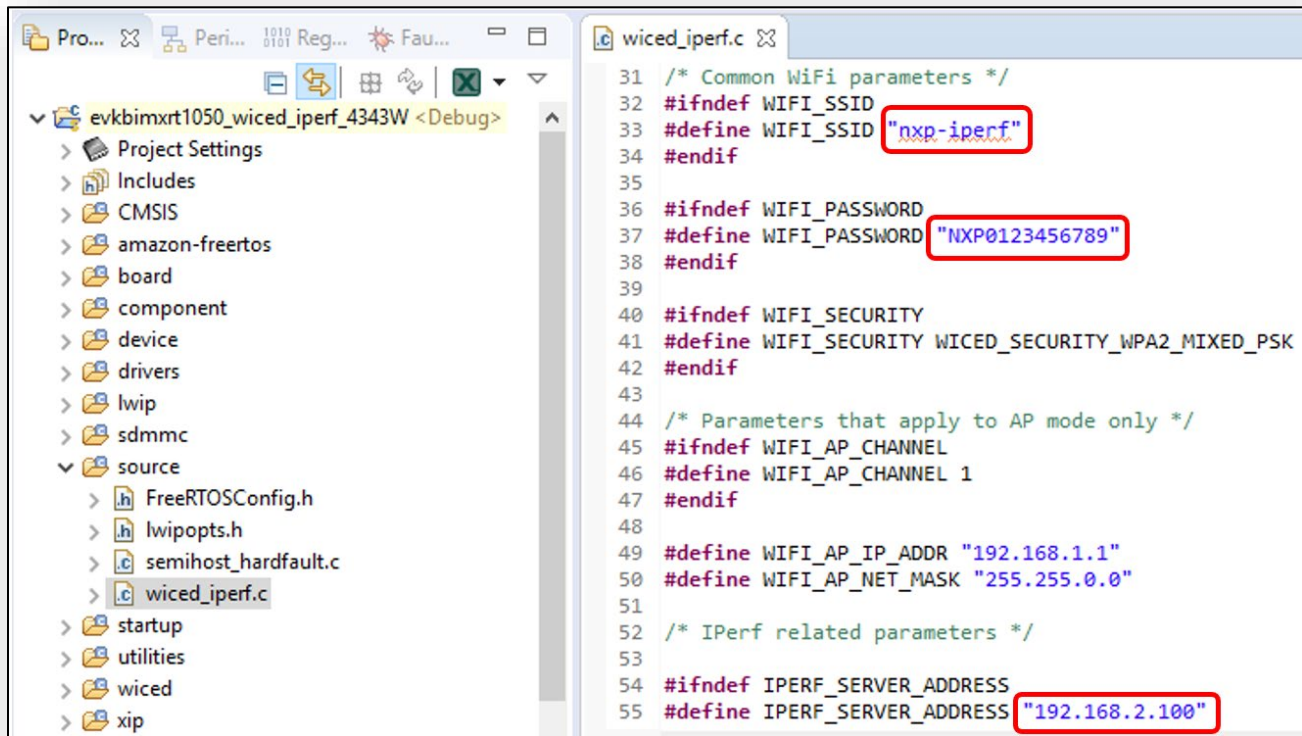


3. Expand wifi_examples and select wiced_iperf_4343W. Select UART for SDK Debug Console, then click Finish button.



NOTE: If you are using Embedded Artists board, modify flash memory size from 0x800000 to 0x400000 and select “MIMXRT1050-EcoXiP_ATXP032.cfx” as the driver. For details and to run the wiced_iperf_4343W example in the EA’s i.MX RT 1062 Developer’s Kit, please refer to the [Getting Started with M.2 modules and i.MX RT](#).

- Open evkbimxrt1050_wiced_iperf_4343W/source/wiced_iperf.c to modify WIFI_SSID, WIFI_PASSWORD and IPERF_SERVER_ADDRESS.

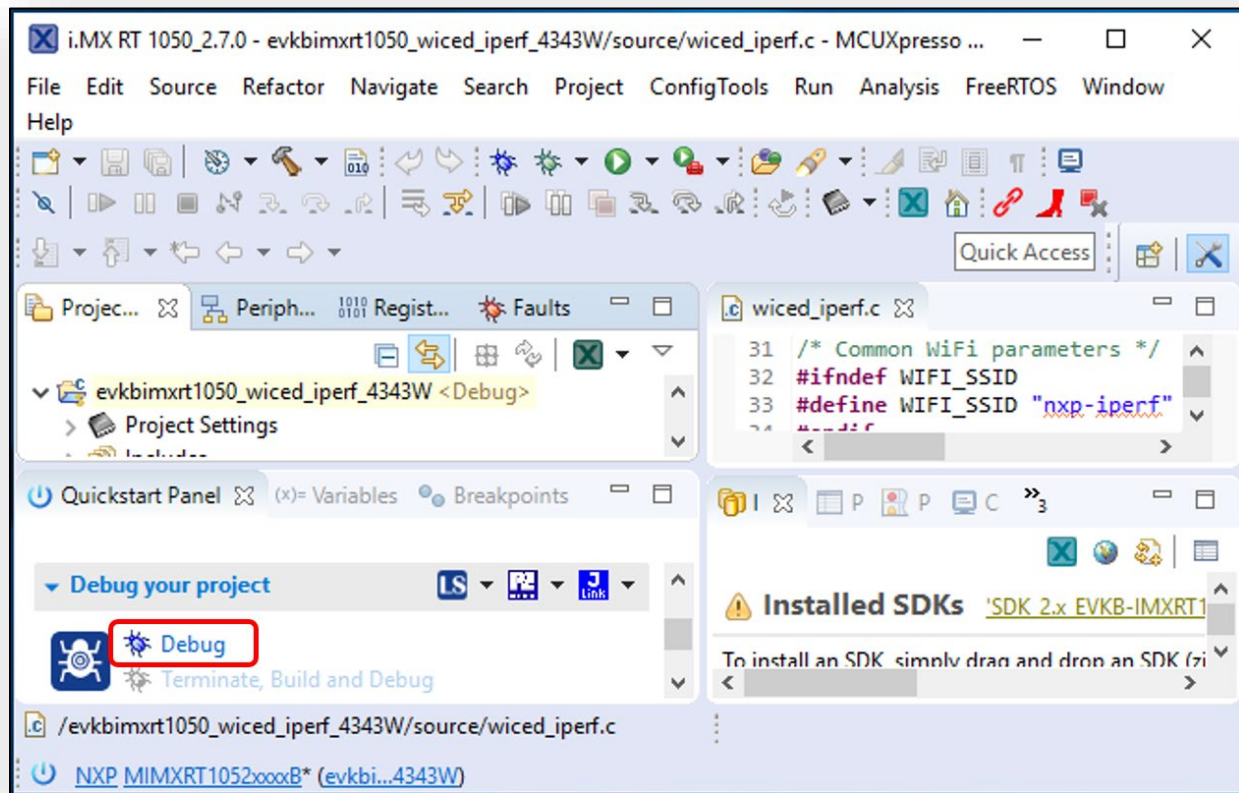


In case of Embedded Artists' board, you need to add or modify some files in the example before starting debug. You can get these files from "imxrt10xx_ea_files_sdk270_yyyy-mm-dd.zip" downloaded in **Section 5**.

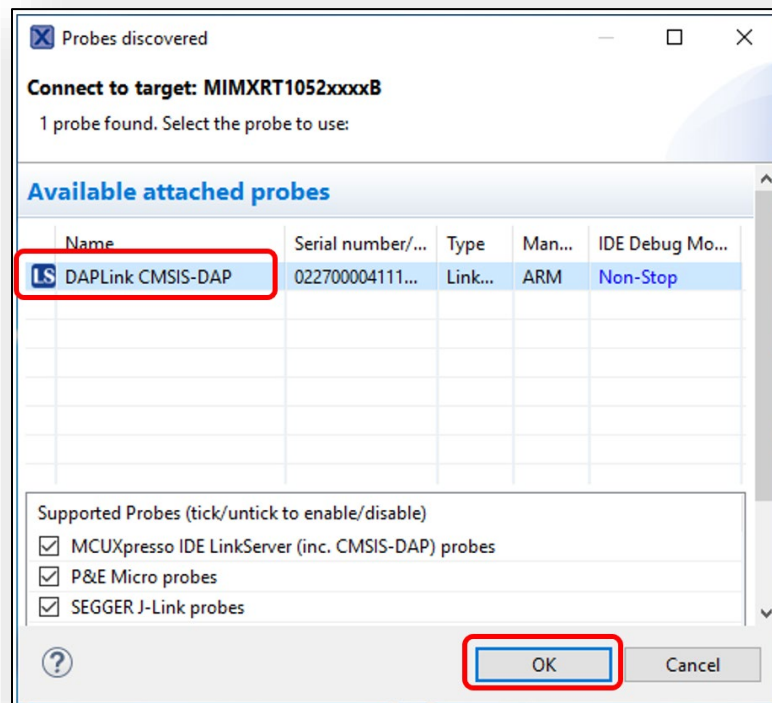
Table 4: Files to be modified on Embedded Artists' Dev Kit

File to copy	Destination	Comment
pin_mux.c	board/	Replace existing file
fsl_lpi2c.c	drivers/	New file
fsl_lpi2c.h	drivers/	New file
pca6416.c	source/	New file
pca6416.h	source/	New file
wwd_platform.c	43xxx_wifi/WICED/platform/MCU/LPC/WWD	Replace existing file
wwd_SDIO.c	43xxx_wifi/WICED/platform/MCU/LPC/WWD	Replace existing file
evkmimxrt10xx_flexspi_nor_config.c	xip/	Replace existing file

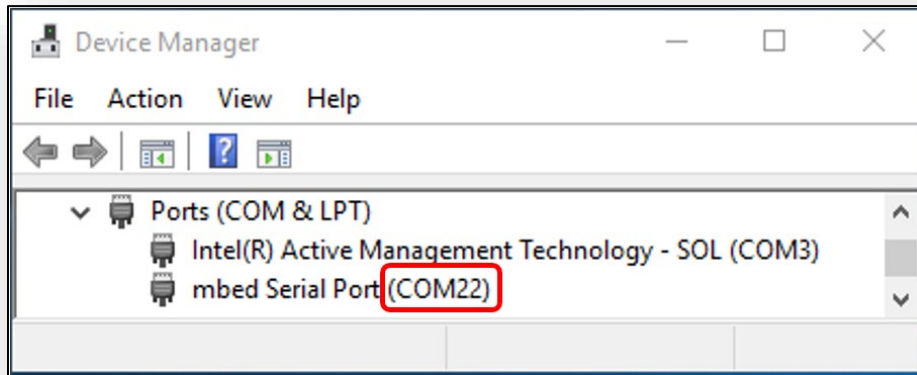
5. Click Debug in the QuickStart Panel.



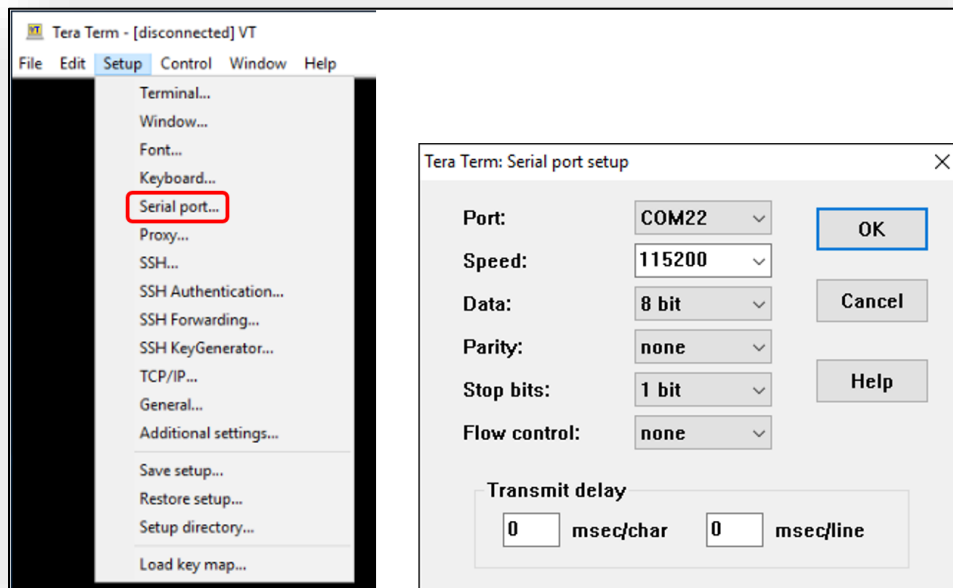
6. For the very first attempt, you need to select the appropriate JTAG adapter. Select the adapter and click OK, and then wait for a while for the build. You can see the log on console window.



7. After the Debug process is complete, open the “Device Manager” in windows and go to “Ports (COM & LPT)” to check the mbed COM port number. In this case, COM 22 is our COM port number.



8. Now the iperf example is ready to run. Open Tera Term on the appropriate COM port (i.e. COM 22 in this case). Configure port for 115200 bps, 8 bits data, no parity, and 1 stop bit (115200/8/N/1).



9. To download the iperf, follow the following link:

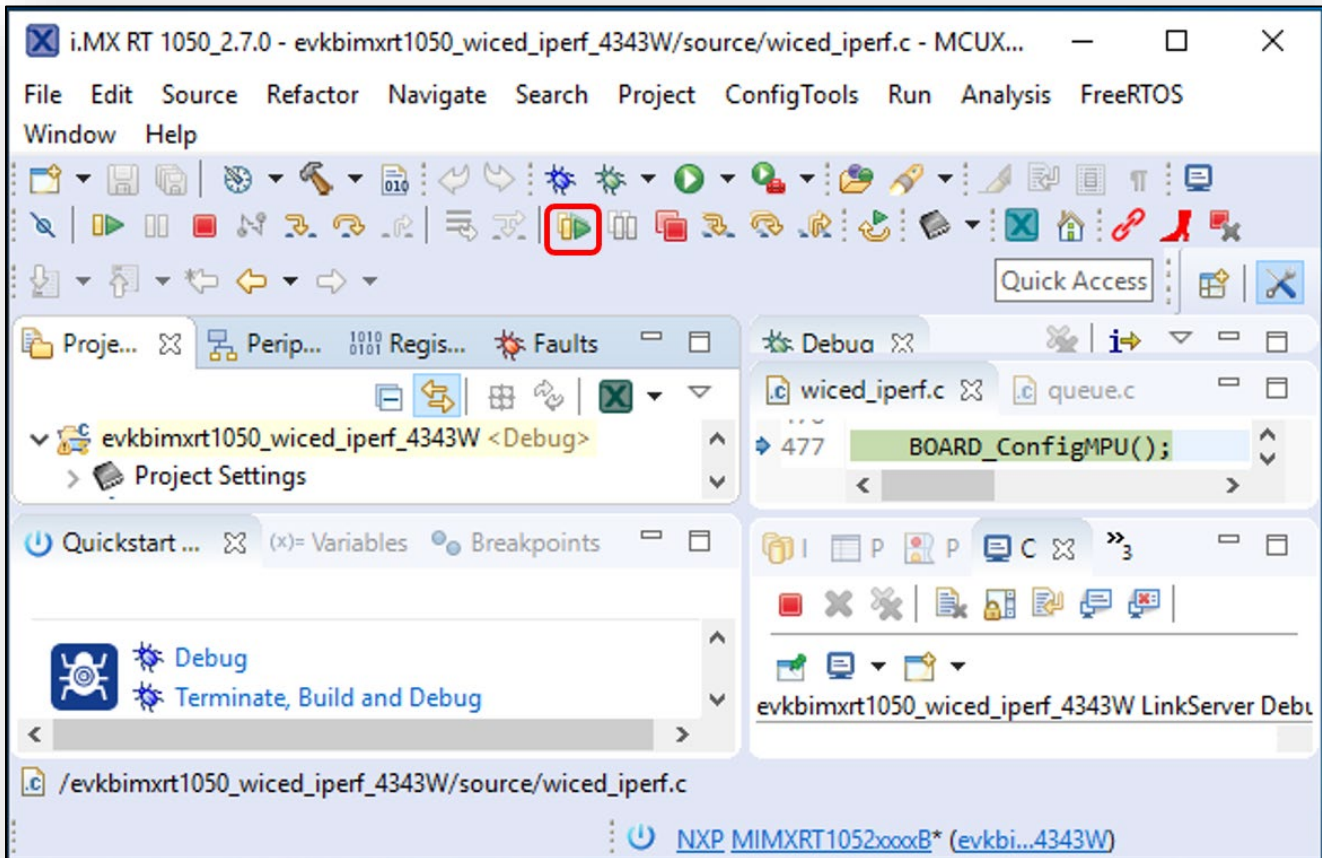
For Windows: <https://iperf.fr/download/windows/iperf-2.0.5-win32.zip>

For Linux: Follow the steps to install iperf.

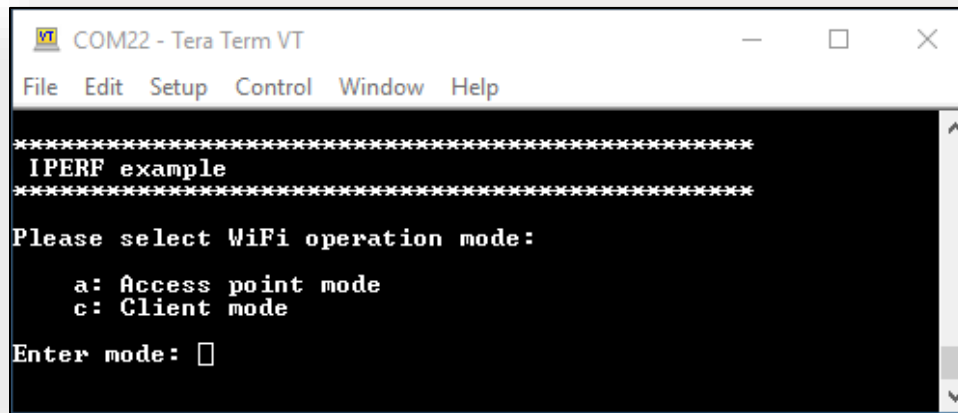
1. Download the .deb file for iperf 2.05 via https://iperf.fr/download/ubuntu/iperf_2.0.5+dfsg1-2_amd64.deb
2. Change directory to where the .deb file is.
3. Run the following commands to install it.

```
$ dpkg -I iperf_2.0.5+dfsg1-2_amd64.deb
$ sudo dpkg -i iperf_2.0.5+dfsg1-2_amd64.deb
$ sudo apt install iperf
```

10. Click resume button in MCUXpresso.



11. You should see this output from i.MX RT.



12. Type “c” to run as client mode. The sample wiced_iperf support several options. Let’s try iperf server. Type ‘1’ in Tera Term.

```

COM22 - Tera Term VT
File Edit Setup Control Window Help
*****
IPERF example
*****

Please select WiFi operation mode:
  a: Access point mode
  c: Client mode

Enter mode: c
Initializing WiFi connection...
Asynchronous Interrupt is not supported
WLAN MAC Address : 00:9D:68:8B:61:47
WLAN Firmware   : wl0: Feb 12 2018 04:08:14 version 7.79.2 (r683798 CY) FWID 01-27b63357
WLAN CLM       : API: 12.2 Data: 9.10.37 Compiler: 1.29.4 Cimport: 1.36.3 Creation: 20
18-02-12 04:00:50
Successfully initialized WiFi module
Joining: nxp-iperf
Successfully joined: nxp-iperf
Getting IP address from DHCP server
IPv4 Address got from DHCP : 192.168.2.130

Please select one of the following modes to run IPERF with:

  1: TCP server mode (RX only test)
  2: TCP client mode (TX only test)
  3: TCP client dual mode (TX and RX in parallel)
  4: TCP client tradeoff mode (TX and RX sequentially)
  5: UDP server mode (RX only test)
  6: UDP client mode (TX only test)
  7: UDP client dual mode (TX and RX in parallel)
  8: UDP client tradeoff mode (TX and RX sequentially)

Enter mode number: █
  
```

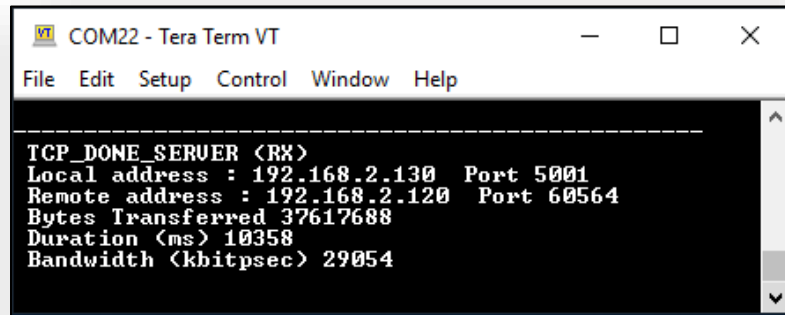
13. To see the throughput numbers, type iperf command in another laptop with iperf2 installed in it: “iperf -c <ip address> -w 256k -i 1 -P 1”. You will see below throughput.

Laptop iPerf Results:

```

skerr@SDK-W520: ~
skerr@SDK-W520:~$ iperf -c 192.168.2.130 -w 256k -i 1 -P 1
-----
Client connecting to 192.168.2.130, TCP port 5001
TCP window size: 416 KByte (WARNING: requested 256 KByte)
-----
[ 3] local 192.168.2.120 port 60564 connected with 192.168.2.130 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3]  0.0- 1.0 sec  4.12 MBytes  34.6 Mb/s
[ 3]  1.0- 2.0 sec  3.38 MBytes  28.3 Mb/s
[ 3]  2.0- 3.0 sec  3.25 MBytes  27.3 Mb/s
[ 3]  3.0- 4.0 sec  3.88 MBytes  32.5 Mb/s
[ 3]  4.0- 5.0 sec  3.00 MBytes  25.2 Mb/s
[ 3]  5.0- 6.0 sec  3.88 MBytes  32.5 Mb/s
[ 3]  6.0- 7.0 sec  3.38 MBytes  28.3 Mb/s
[ 3]  7.0- 8.0 sec  3.38 MBytes  28.3 Mb/s
[ 3]  8.0- 9.0 sec  4.00 MBytes  33.6 Mb/s
[ 3]  9.0-10.0 sec  3.50 MBytes  29.4 Mb/s
[ 3]  0.0-10.1 sec  35.9 MBytes  29.9 Mb/s
skerr@SDK-W520:~$ █
  
```

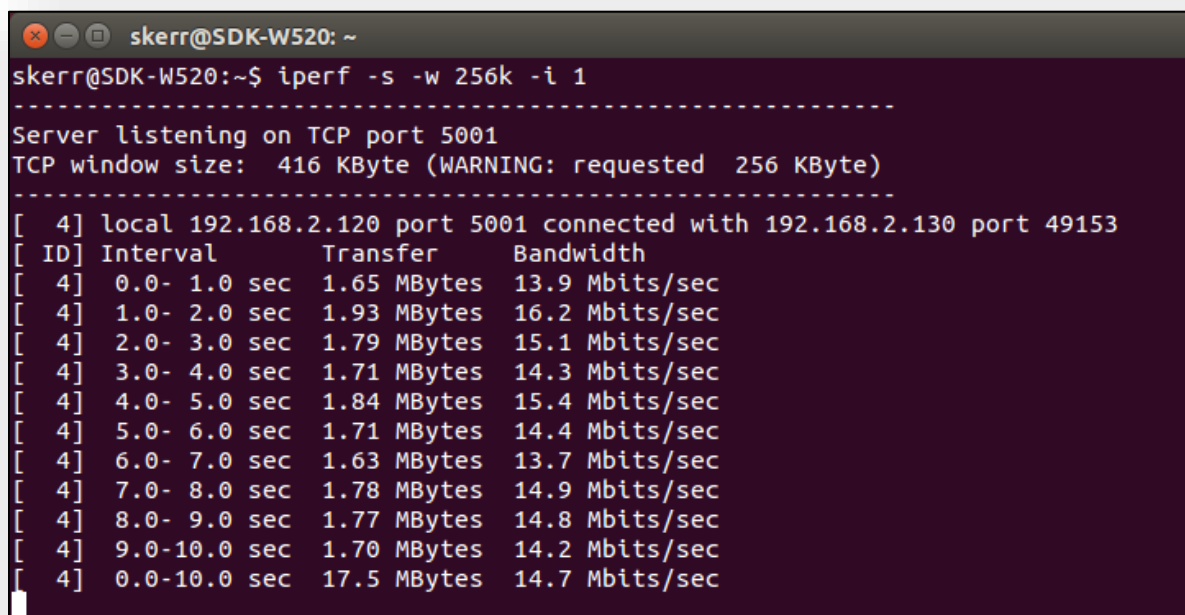

Console Output – Tera Term:



```
COM22 - Tera Term VT
File Edit Setup Control Window Help
-----
TCP_DONE_SERVER <RX>
Local address : 192.168.2.130 Port 5001
Remote address : 192.168.2.120 Port 60564
Bytes Transferred 37617688
Duration <ms> 10358
Bandwidth <kbitpsec> 29054
```

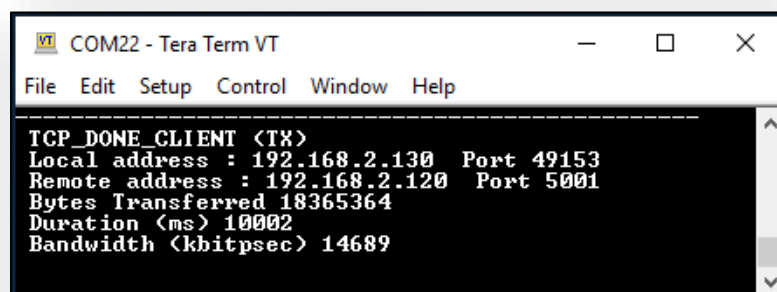
14. Type 2 in Tera Term to run TCP client mode (TX only). Before typing 2, make sure to run server on another Laptop (“iperf -s -w 256k -i 1”). You will see below outputs:

Laptop iPerf Results:



```
skerr@SDK-W520: ~
skerr@SDK-W520:~$ iperf -s -w 256k -i 1
-----
Server listening on TCP port 5001
TCP window size: 416 KByte (WARNING: requested 256 KByte)
-----
[ 4] local 192.168.2.120 port 5001 connected with 192.168.2.130 port 49153
[ ID] Interval      Transfer      Bandwidth
[ 4] 0.0- 1.0 sec  1.65 MBytes  13.9 Mbits/sec
[ 4] 1.0- 2.0 sec  1.93 MBytes  16.2 Mbits/sec
[ 4] 2.0- 3.0 sec  1.79 MBytes  15.1 Mbits/sec
[ 4] 3.0- 4.0 sec  1.71 MBytes  14.3 Mbits/sec
[ 4] 4.0- 5.0 sec  1.84 MBytes  15.4 Mbits/sec
[ 4] 5.0- 6.0 sec  1.71 MBytes  14.4 Mbits/sec
[ 4] 6.0- 7.0 sec  1.63 MBytes  13.7 Mbits/sec
[ 4] 7.0- 8.0 sec  1.78 MBytes  14.9 Mbits/sec
[ 4] 8.0- 9.0 sec  1.77 MBytes  14.8 Mbits/sec
[ 4] 9.0-10.0 sec  1.70 MBytes  14.2 Mbits/sec
[ 4] 0.0-10.0 sec  17.5 MBytes  14.7 Mbits/sec
```

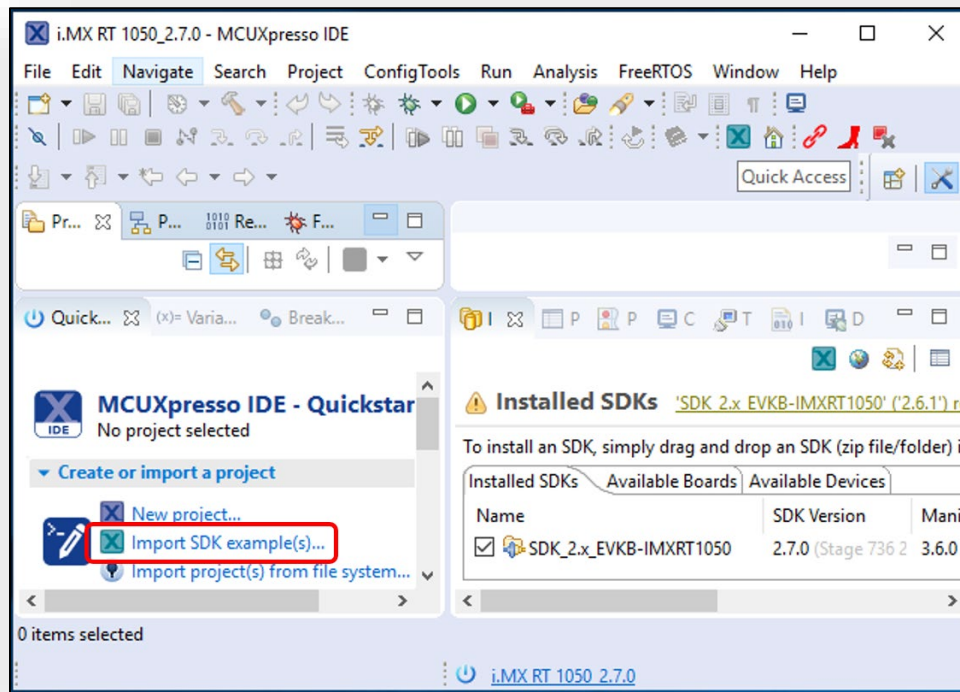
Console Output – Tera Term:



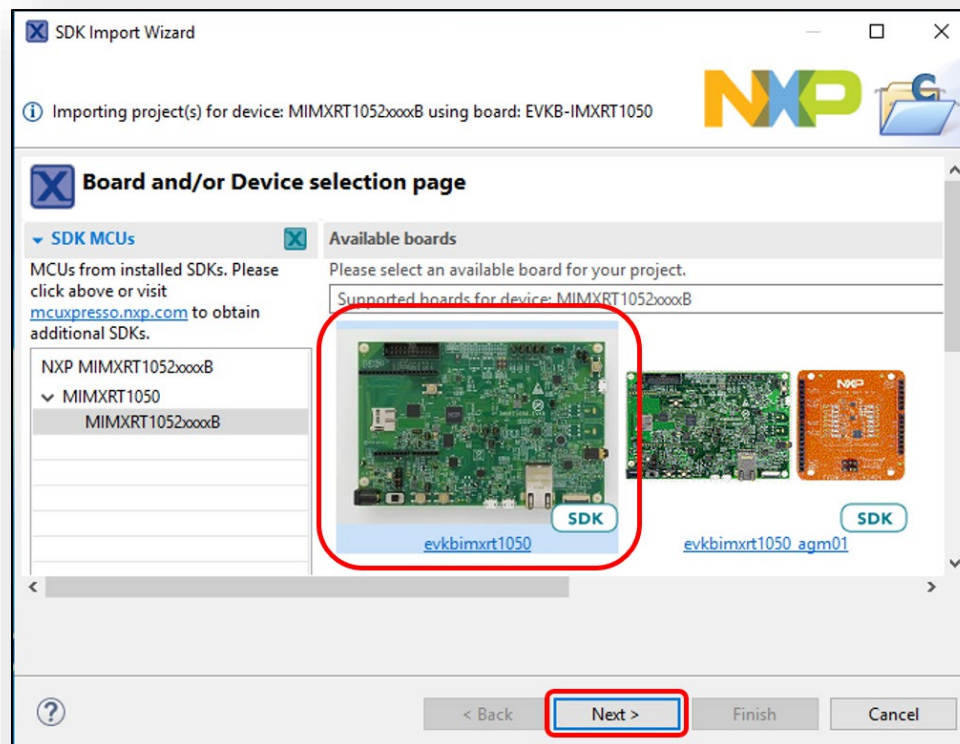
```
COM22 - Tera Term VT
File Edit Setup Control Window Help
-----
TCP_DONE_CLIENT <TX>
Local address : 192.168.2.130 Port 49153
Remote address : 192.168.2.120 Port 5001
Bytes Transferred 18365364
Duration <ms> 10002
Bandwidth <kbitpsec> 14689
```

7.2 Example wiced_iperf_43012

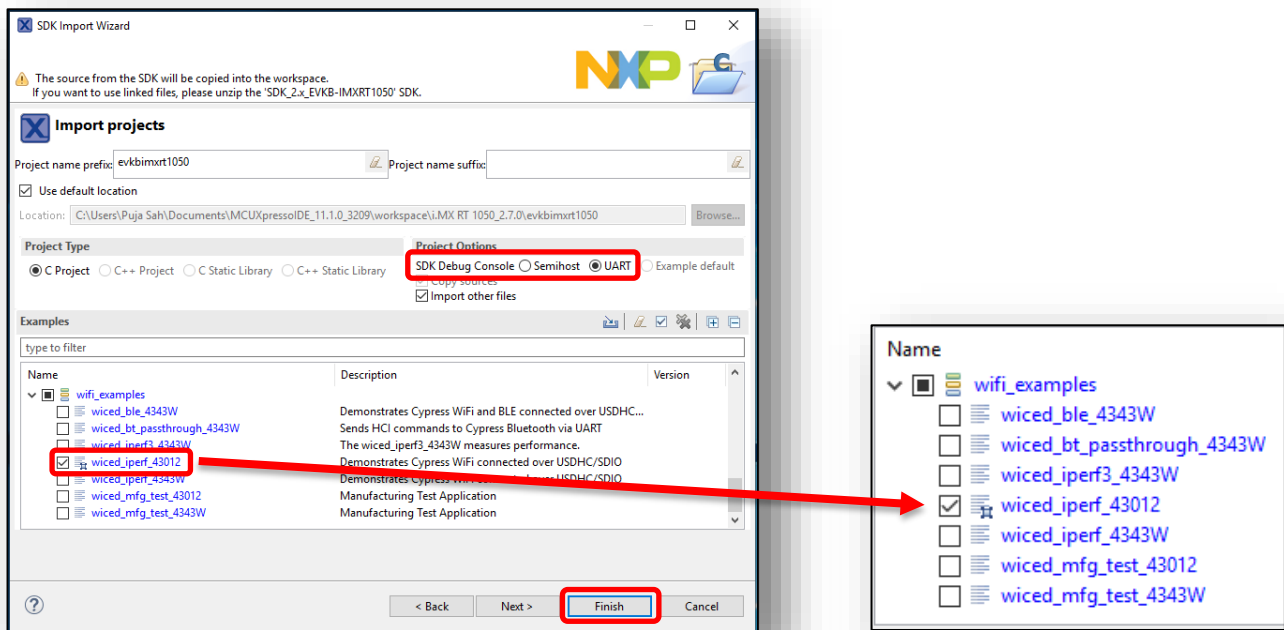
1. Click on “Import SDK example(s)...” in the Quickstart Panel.



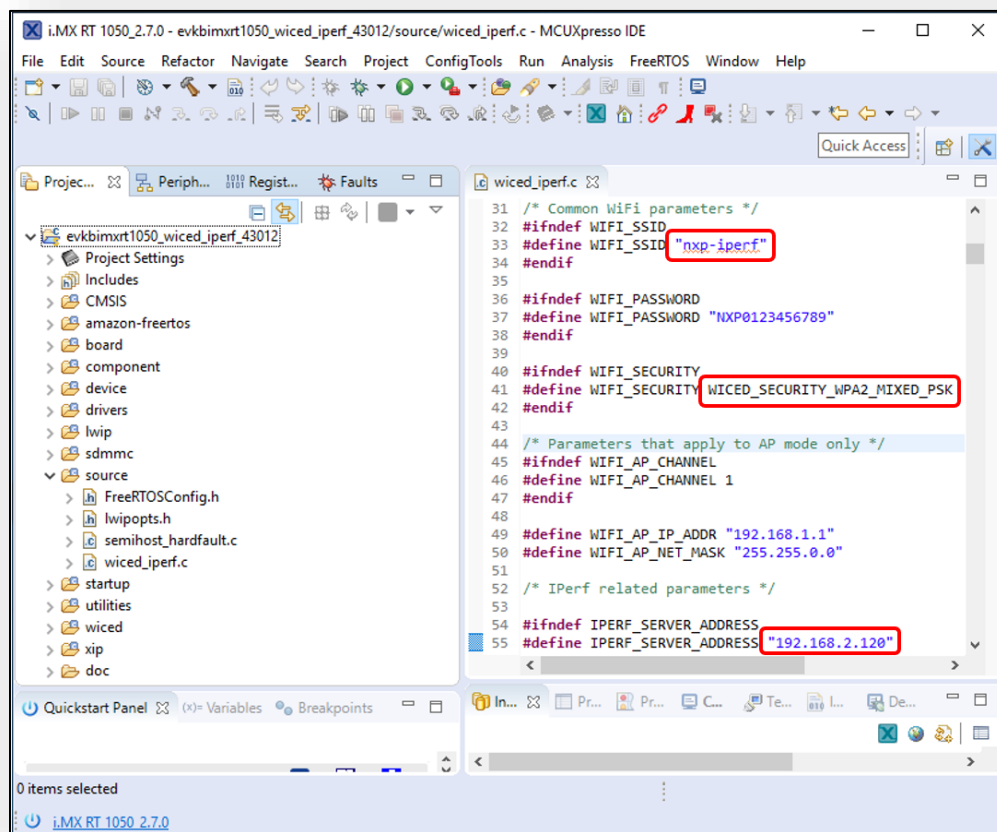
2. Select “evkbimxrt1050” board and click Next button.



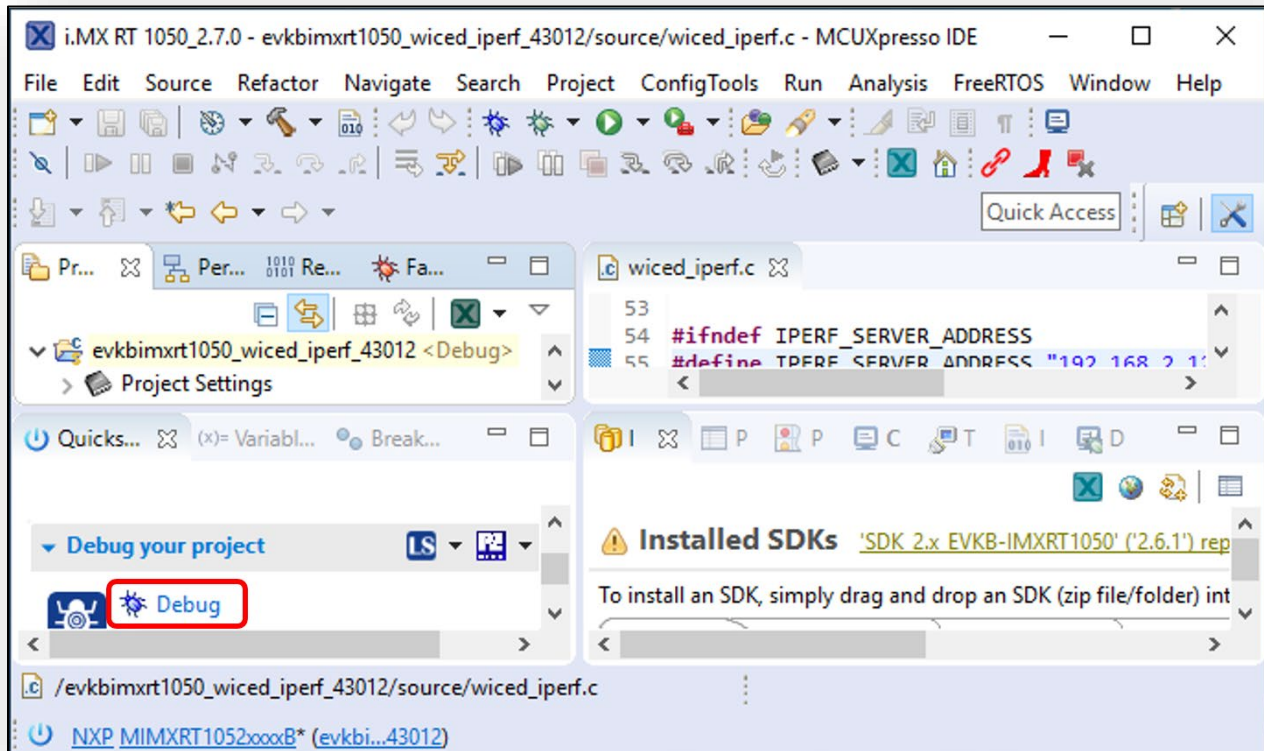
- Expand `wifi_examples` and select `wiced_iperf_43012`. Select UART for SDK Debug Console, then click Finish button.



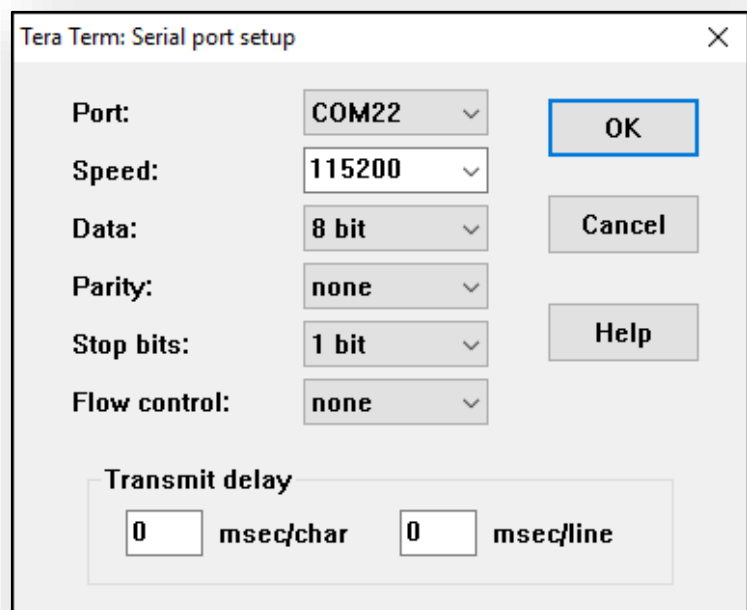
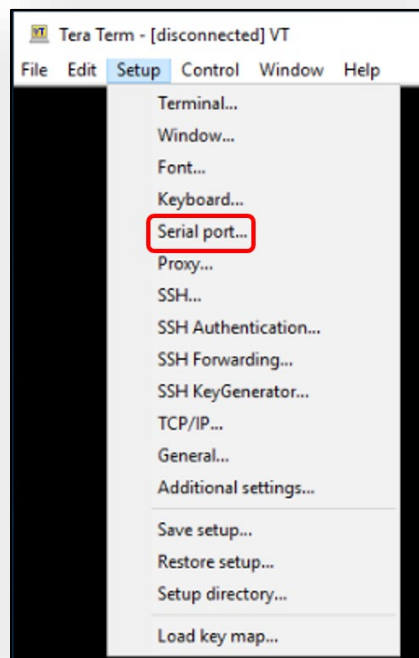
- Open `source/wiced_iperf.c` to modify `WIFI_SSID`, `WIFI_PASSWORD` and `IPERF_SERVER_ADDRESS`.



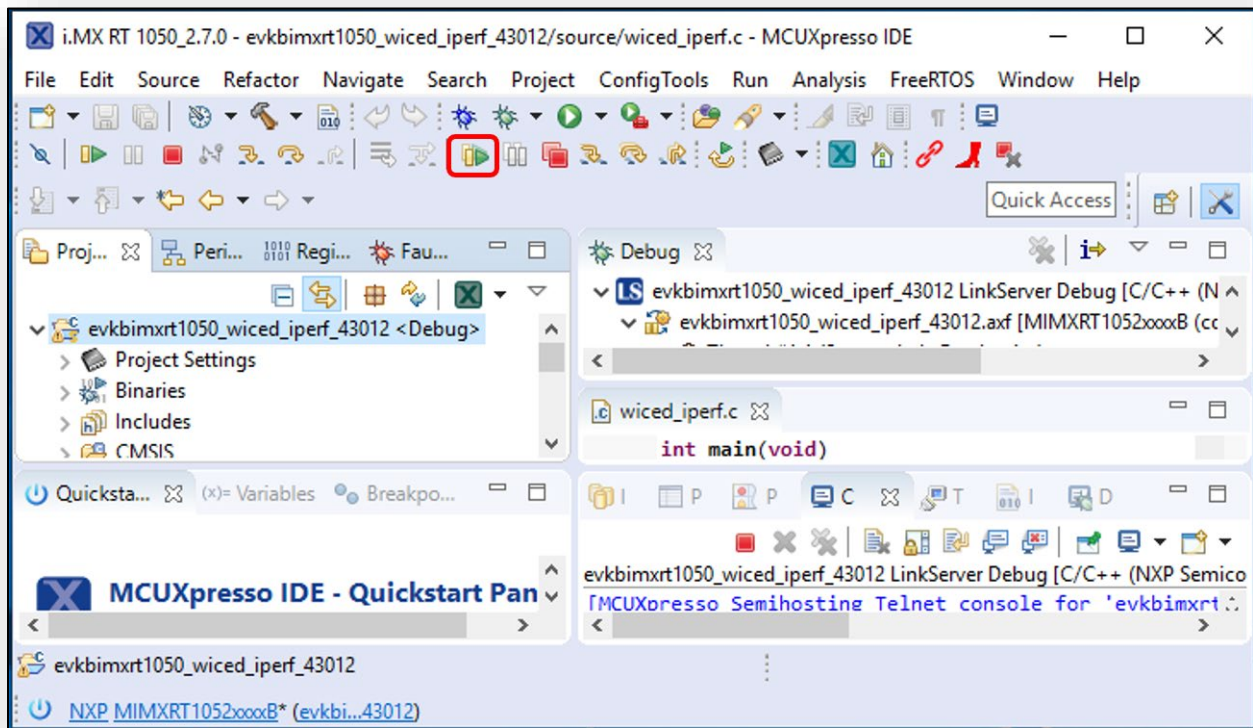
5. Click on “Debug” button in the Quickstart Panel.



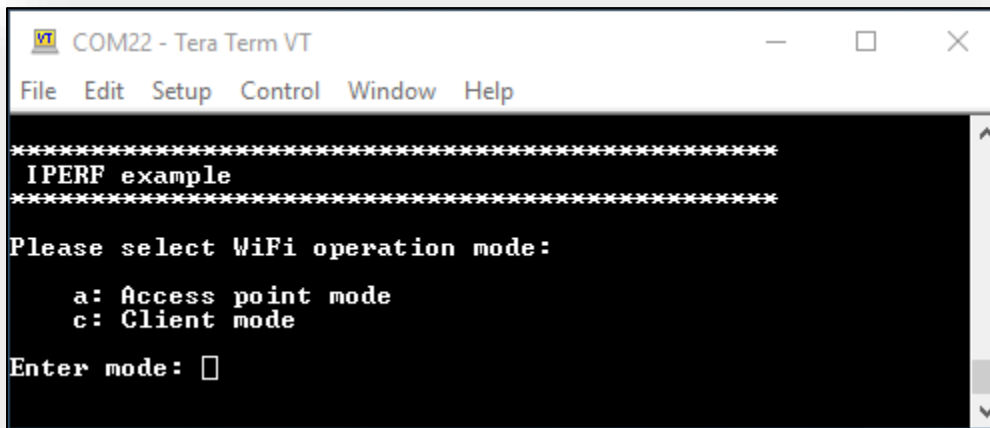
6. After the debug process is complete, the iperf example is ready to run. Open Tera Term on the appropriate COM port (i.e., COM 22 in this case). Configure port for 115200 bps, 8 bits data, no parity, and 1 stop bit (115200/8/N/1).



7. Click resume button in MCUXpresso.



8. You should see following output in Tera Term.



9. Type “c” to run as client mode. The sample wiced_iperf support several options. Let’s try iperf server. Type ‘1’ in Tera Term.

```

COM22 - Tera Term VT
File Edit Setup Control Window Help
*****
IPERF example
*****
Please select WiFi operation mode:

a: Access point mode
c: Client mode

Enter mode: c
Initializing WiFi connection
WLAN MAC Address : 10:98:C3:22:C4:B6
WLAN Firmware   : wl0: Mar 12 2019 23:51:54 version 13.10.271.154 (r712270) FWID 01-6b3ca8a7
WLAN CLM       : API: 18.2 Data: 9.10.0 Compiler: 1.36.1 Clmimport: 1.34.1 Creation: 2019-03-
12 23:42:29
Successfully initialized WiFi module
Joining: nxp-iperf
Successfully joined: nxp-iperf
Getting IP address from DHCP server
IPv4 Address got from DHCP : 192.168.2.144
Please select one of the following modes to run IPERF with:

1: TCP server mode (RX only test)
2: TCP client mode (TX only test)
3: TCP client dual mode (TX and RX in parallel)
4: TCP client tradeoff mode (TX and RX sequentially)
5: UDP server mode (RX only test)
6: UDP client mode (TX only test)
7: UDP client dual mode (TX and RX in parallel)
8: UDP client tradeoff mode (TX and RX sequentially)

Enter mode number: █
  
```

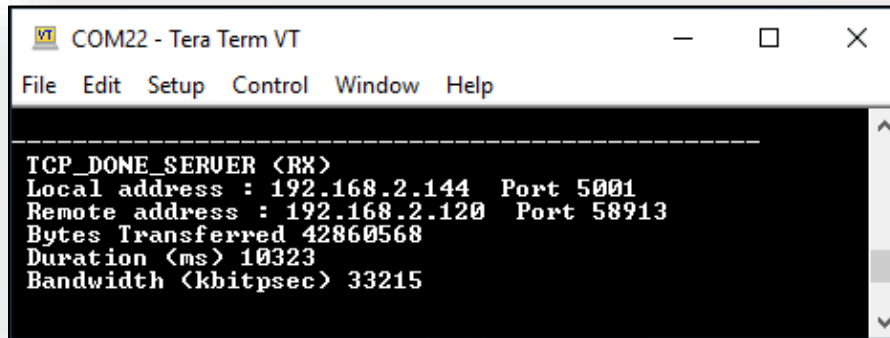
10. To see the throughput numbers, type iperf command in another laptop with iperf2 installed in it: “iperf -c <ip address> -w 256k -i 1 -P 1”. Throughput performance numbers are shown for both 2.4GHz and 5GHz below.

Laptop iPerf Results (2.4GHz):

```

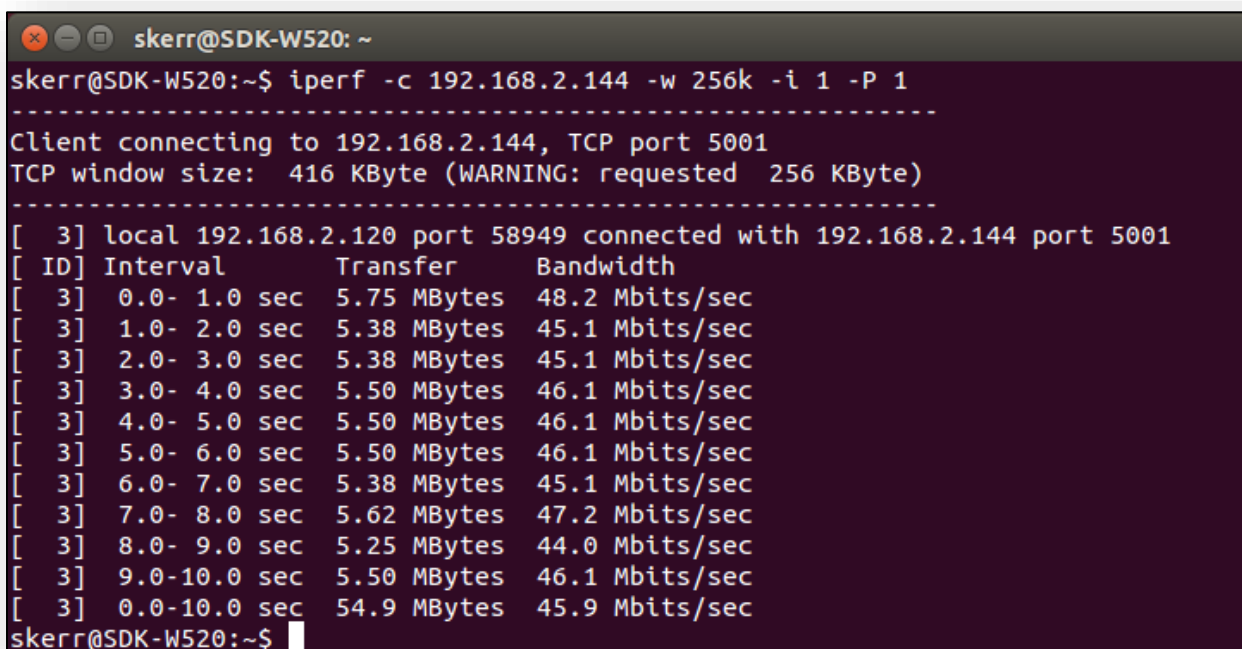
skerr@SDK-W520: ~
skerr@SDK-W520:~$ iperf -c 192.168.2.144 -w 256k -i 1 -P 1
-----
Client connecting to 192.168.2.144, TCP port 5001
TCP window size: 416 KByte (WARNING: requested 256 KByte)
-----
[ 3] local 192.168.2.120 port 58913 connected with 192.168.2.144 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0- 1.0 sec  4.38 MBytes  36.7 Mb/s
[ 3] 1.0- 2.0 sec  4.12 MBytes  34.6 Mb/s
[ 3] 2.0- 3.0 sec  3.62 MBytes  30.4 Mb/s
[ 3] 3.0- 4.0 sec  4.25 MBytes  35.7 Mb/s
[ 3] 4.0- 5.0 sec  4.12 MBytes  34.6 Mb/s
[ 3] 5.0- 6.0 sec  4.25 MBytes  35.7 Mb/s
[ 3] 6.0- 7.0 sec  3.88 MBytes  32.5 Mb/s
[ 3] 7.0- 8.0 sec  4.25 MBytes  35.7 Mb/s
[ 3] 8.0- 9.0 sec  4.12 MBytes  34.6 Mb/s
[ 3] 9.0-10.0 sec  3.75 MBytes  31.5 Mb/s
[ 3] 0.0-10.0 sec 40.9 MBytes  34.2 Mb/s
skerr@SDK-W520:~$ █
  
```

Console Output – Tera Term (2.4GHz):



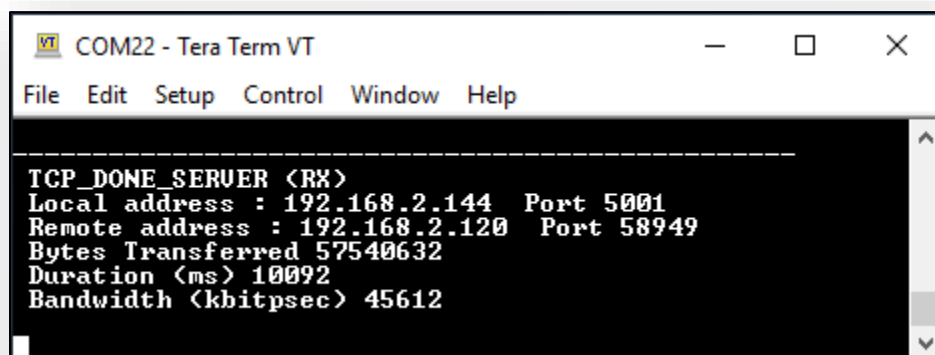
```
COM22 - Tera Term VT
File Edit Setup Control Window Help
-----
TCP_DONE_SERVER <RX>
Local address : 192.168.2.144 Port 5001
Remote address : 192.168.2.120 Port 58913
Bytes Transferred 42860568
Duration <ms> 10323
Bandwidth <kbitpsec> 33215
```

Laptop iPerf Results (5GHz):



```
skerr@SDK-W520: ~
skerr@SDK-W520:~$ iperf -c 192.168.2.144 -w 256k -i 1 -P 1
-----
Client connecting to 192.168.2.144, TCP port 5001
TCP window size: 416 KByte (WARNING: requested 256 KByte)
-----
[ 3] local 192.168.2.120 port 58949 connected with 192.168.2.144 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0- 1.0 sec  5.75 MBytes 48.2 Mbits/sec
[ 3] 1.0- 2.0 sec  5.38 MBytes 45.1 Mbits/sec
[ 3] 2.0- 3.0 sec  5.38 MBytes 45.1 Mbits/sec
[ 3] 3.0- 4.0 sec  5.50 MBytes 46.1 Mbits/sec
[ 3] 4.0- 5.0 sec  5.50 MBytes 46.1 Mbits/sec
[ 3] 5.0- 6.0 sec  5.50 MBytes 46.1 Mbits/sec
[ 3] 6.0- 7.0 sec  5.38 MBytes 45.1 Mbits/sec
[ 3] 7.0- 8.0 sec  5.62 MBytes 47.2 Mbits/sec
[ 3] 8.0- 9.0 sec  5.25 MBytes 44.0 Mbits/sec
[ 3] 9.0-10.0 sec  5.50 MBytes 46.1 Mbits/sec
[ 3] 0.0-10.0 sec 54.9 MBytes 45.9 Mbits/sec
skerr@SDK-W520:~$
```

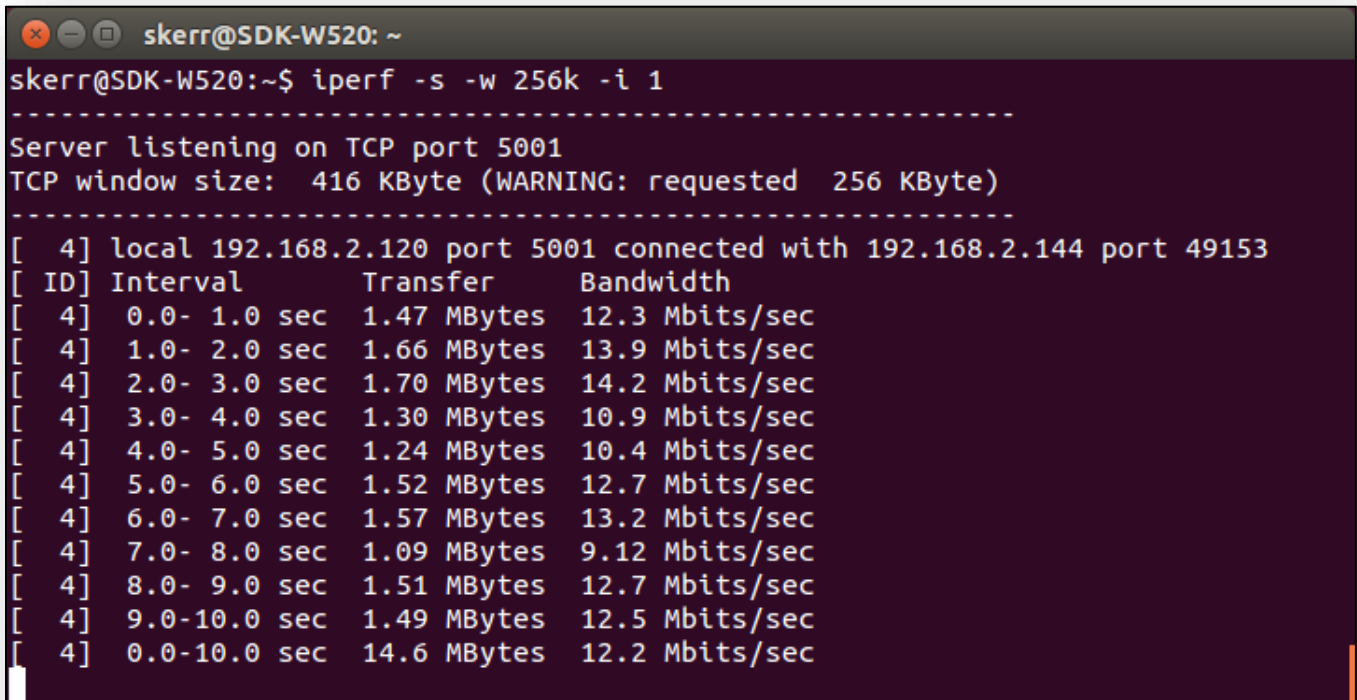
Console output – Tera Term (5GHz):



```
COM22 - Tera Term VT
File Edit Setup Control Window Help
-----
TCP_DONE_SERVER <RX>
Local address : 192.168.2.144 Port 5001
Remote address : 192.168.2.120 Port 58949
Bytes Transferred 57540632
Duration <ms> 10092
Bandwidth <kbitpsec> 45612
```

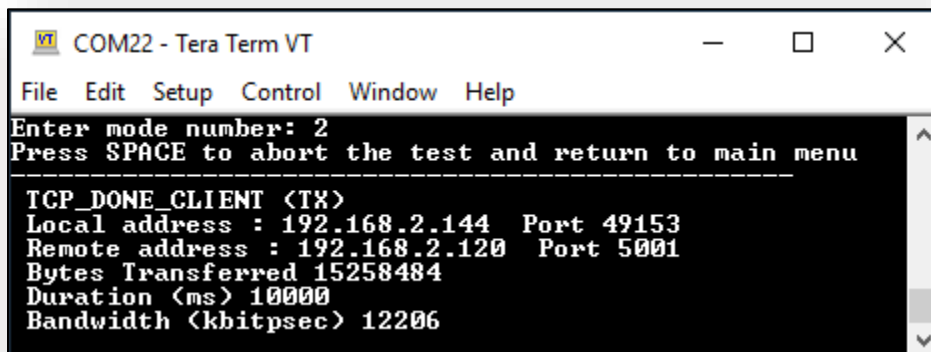
11. Type 2 in Tera Term to run TCP client mode (TX only). Before typing 2, make sure to run server on another Laptop (“iperf -s -w 256k -i 1”). You will see below outputs:

Laptop iPerf Results (2.4GHz):



```
skerr@SDK-W520: ~
skerr@SDK-W520:~$ iperf -s -w 256k -i 1
-----
Server listening on TCP port 5001
TCP window size: 416 KByte (WARNING: requested 256 KByte)
-----
[ 4] local 192.168.2.120 port 5001 connected with 192.168.2.144 port 49153
[ ID] Interval      Transfer      Bandwidth
[ 4] 0.0- 1.0 sec  1.47 MBytes  12.3 Mbits/sec
[ 4] 1.0- 2.0 sec  1.66 MBytes  13.9 Mbits/sec
[ 4] 2.0- 3.0 sec  1.70 MBytes  14.2 Mbits/sec
[ 4] 3.0- 4.0 sec  1.30 MBytes  10.9 Mbits/sec
[ 4] 4.0- 5.0 sec  1.24 MBytes  10.4 Mbits/sec
[ 4] 5.0- 6.0 sec  1.52 MBytes  12.7 Mbits/sec
[ 4] 6.0- 7.0 sec  1.57 MBytes  13.2 Mbits/sec
[ 4] 7.0- 8.0 sec  1.09 MBytes   9.12 Mbits/sec
[ 4] 8.0- 9.0 sec  1.51 MBytes  12.7 Mbits/sec
[ 4] 9.0-10.0 sec  1.49 MBytes  12.5 Mbits/sec
[ 4] 0.0-10.0 sec  14.6 MBytes  12.2 Mbits/sec
```

Console output – Tera Term (2.4GHz):



```
COM22 - Tera Term VT
File Edit Setup Control Window Help
Enter mode number: 2
Press SPACE to abort the test and return to main menu
-----
TCP_DONE_CLIENT <TX>
Local address : 192.168.2.144 Port 49153
Remote address : 192.168.2.120 Port 5001
Bytes Transferred 15258484
Duration <ms> 10000
Bandwidth <kbitpsec> 12206
```


Laptop iPerf Results (5GHz):

```
skerr@SDK-W520: ~
skerr@SDK-W520:~$ iperf -s -w 256k -i 1
-----
Server listening on TCP port 5001
TCP window size: 416 KByte (WARNING: requested 256 KByte)
-----
[ 4] local 192.168.2.120 port 5001 connected with 192.168.2.144 port 49153
[ ID] Interval          Transfer          Bandwidth
[ 4] 0.0- 1.0 sec      2.01 MBytes      16.9 Mb/s
[ 4] 1.0- 2.0 sec      1.75 MBytes      14.7 Mb/s
[ 4] 2.0- 3.0 sec      2.42 MBytes      20.3 Mb/s
[ 4] 3.0- 4.0 sec      2.47 MBytes      20.7 Mb/s
[ 4] 4.0- 5.0 sec      2.54 MBytes      21.3 Mb/s
[ 4] 5.0- 6.0 sec      2.51 MBytes      21.1 Mb/s
[ 4] 6.0- 7.0 sec      2.50 MBytes      20.9 Mb/s
[ 4] 7.0- 8.0 sec      2.56 MBytes      21.5 Mb/s
[ 4] 8.0- 9.0 sec      2.58 MBytes      21.7 Mb/s
[ 4] 9.0-10.0 sec      2.53 MBytes      21.2 Mb/s
[ 4] 0.0-10.0 sec      23.9 MBytes      20.0 Mb/s
```

Console output – Tera Term (5GHz):

```
COM22 - Tera Term VT
File Edit Setup Control Window Help
Enter mode number: 2
Press SPACE to abort the test and return to main menu
-----
TCP_DONE_CLIENT <TX>
Local address : 192.168.2.144 Port 49153
Remote address : 192.168.2.120 Port 5001
Bytes Transferred 25034644
Duration <ms> 10000
Bandwidth <kbitsec> 20027
```

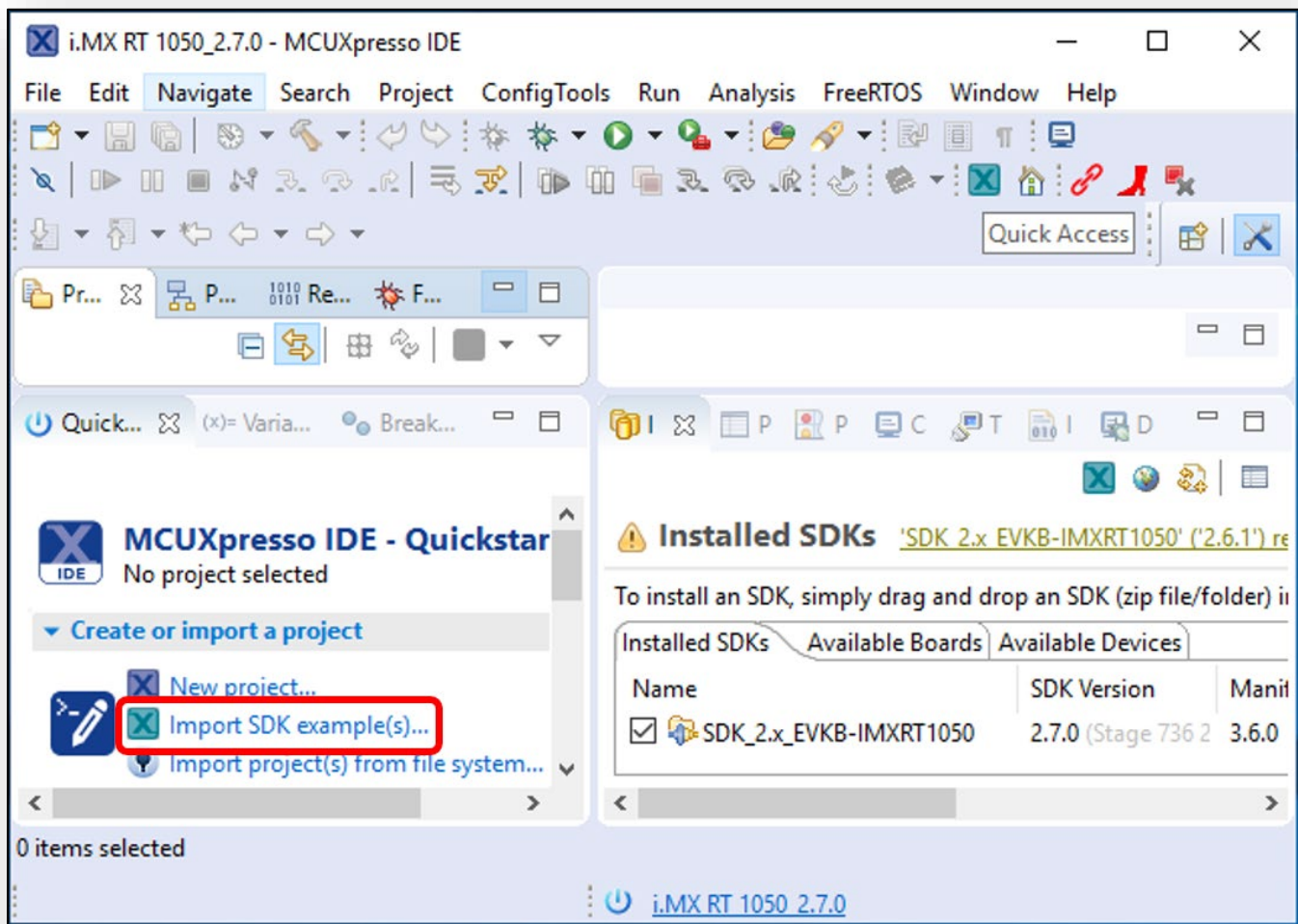
7.3 Example wiced_mfg_test_4343W

1. Hardware Requirement (Optional):

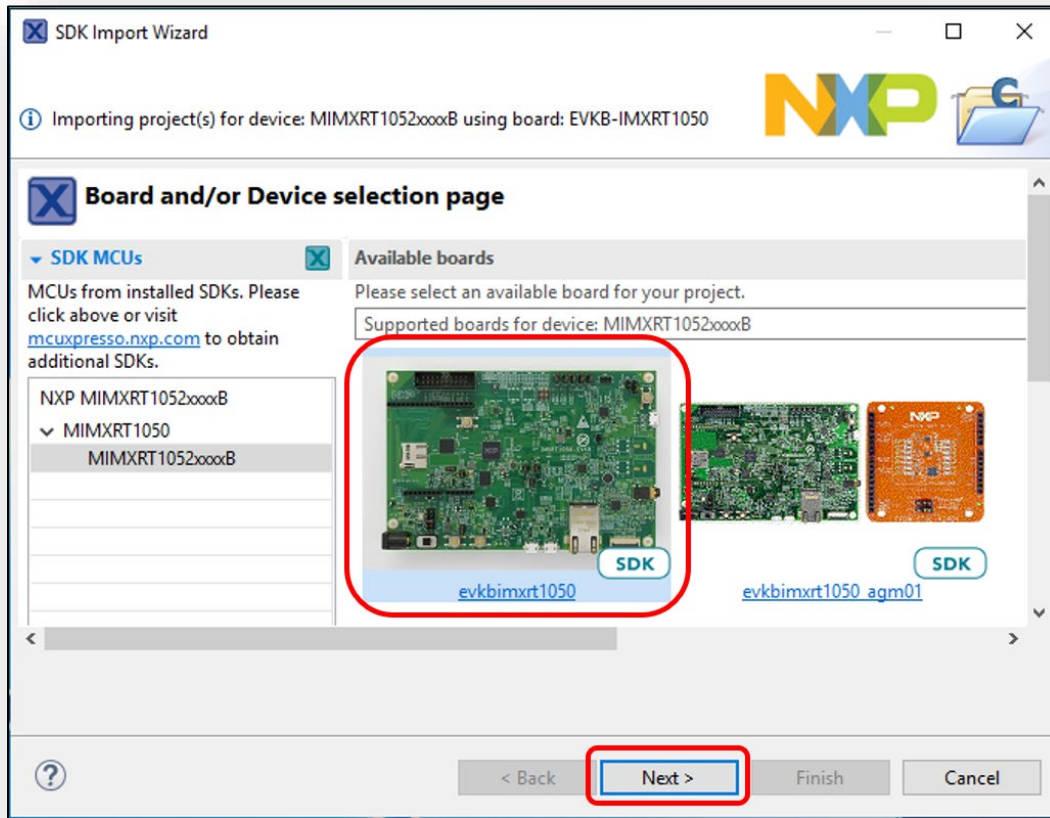
- a. evkbimxrt1050 board's connector J22, pin 3 to Murarta_uSD M.2 Adapter's connector J9, pin 3
- b. Connect the board with PC by using USB/UART converter:
 - board uart RX (pin 1 on J22) - connect it to TX pin on converter
 - board uart TX (pin 2 on J22) - connect it to RX pin on converter
 - board GND (pin 7 on J24)

2. Step 1 is only required if you want to see throughput in the terminal window. Otherwise just skip step 1 and go to step 3.

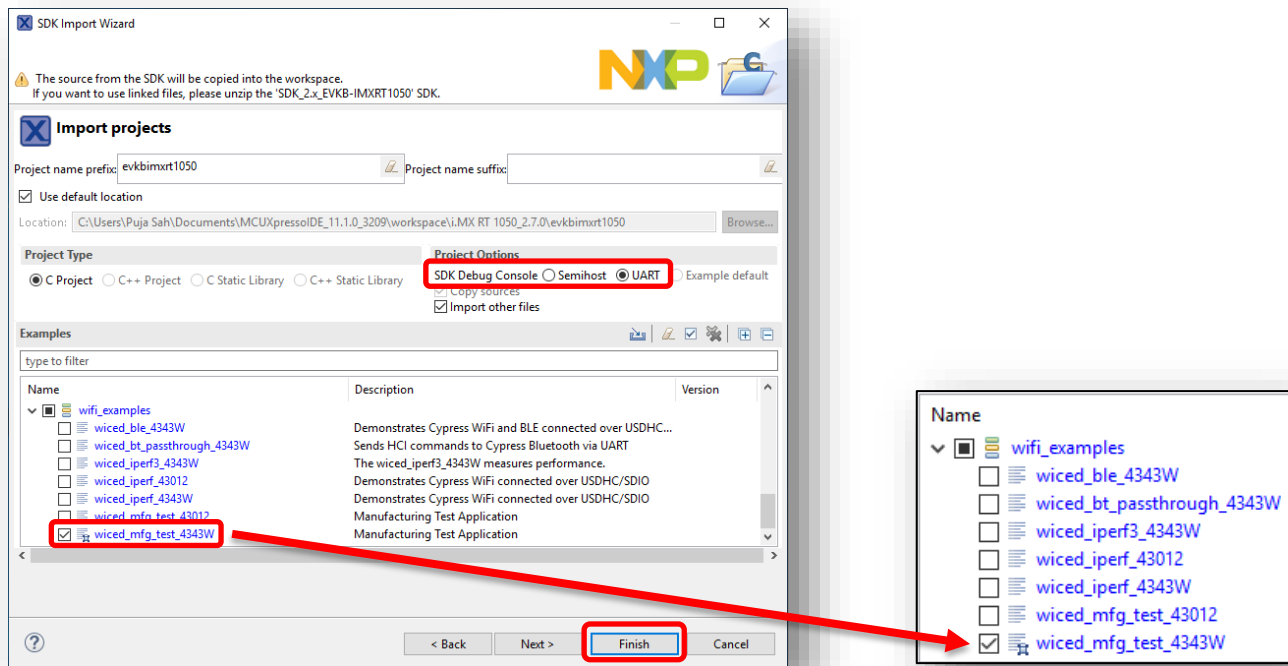
3. Click on “Import SDK example(s)...” in the Quickstart Panel.



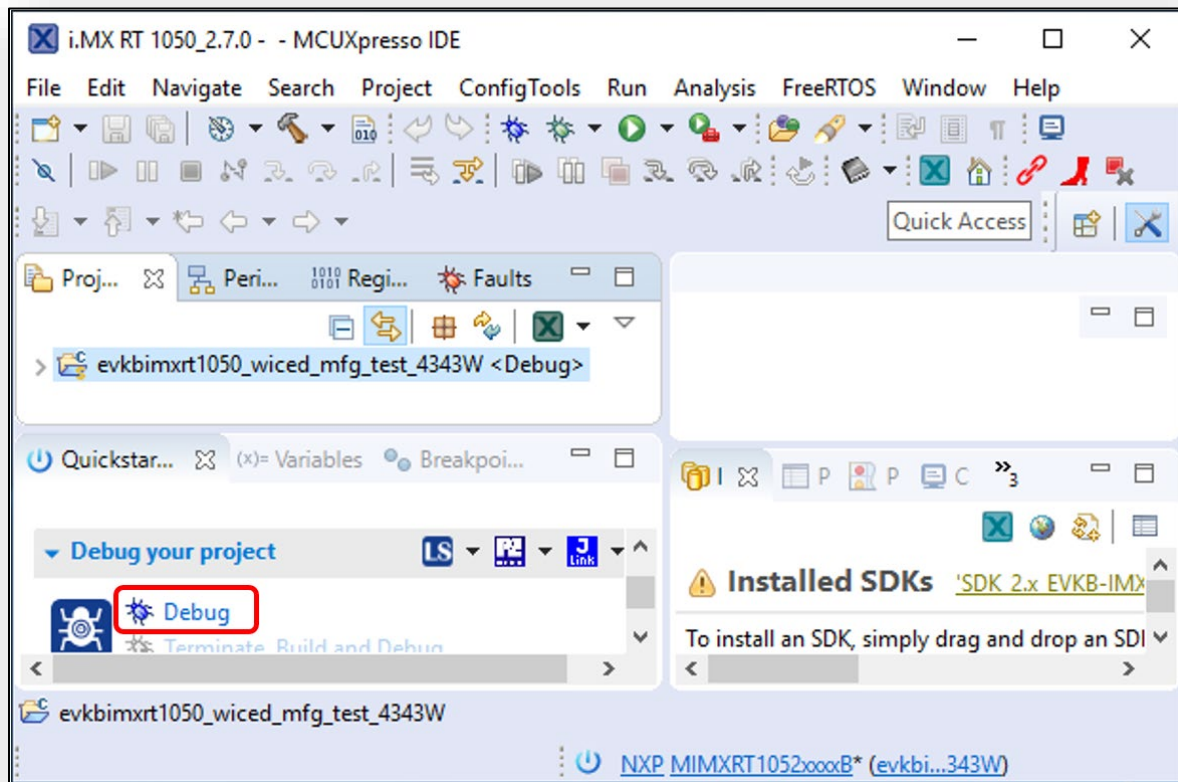
4. Select “evkbimxrt1050” board and click Next button.



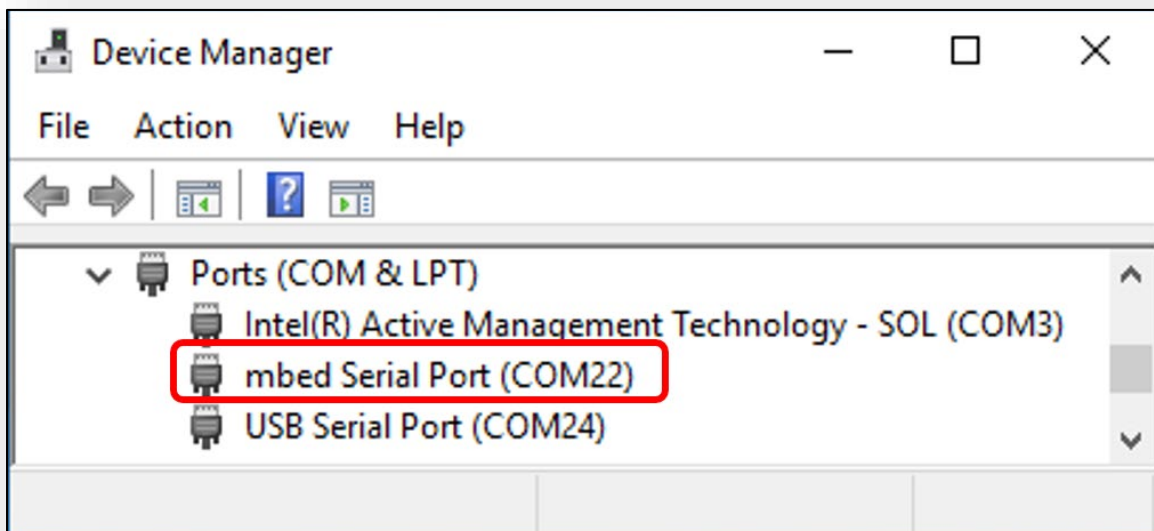
5. Expand wifi_examples and select wiced_mfg_test_4343W. Select UART for SDK Debug Console, then click Finish button.



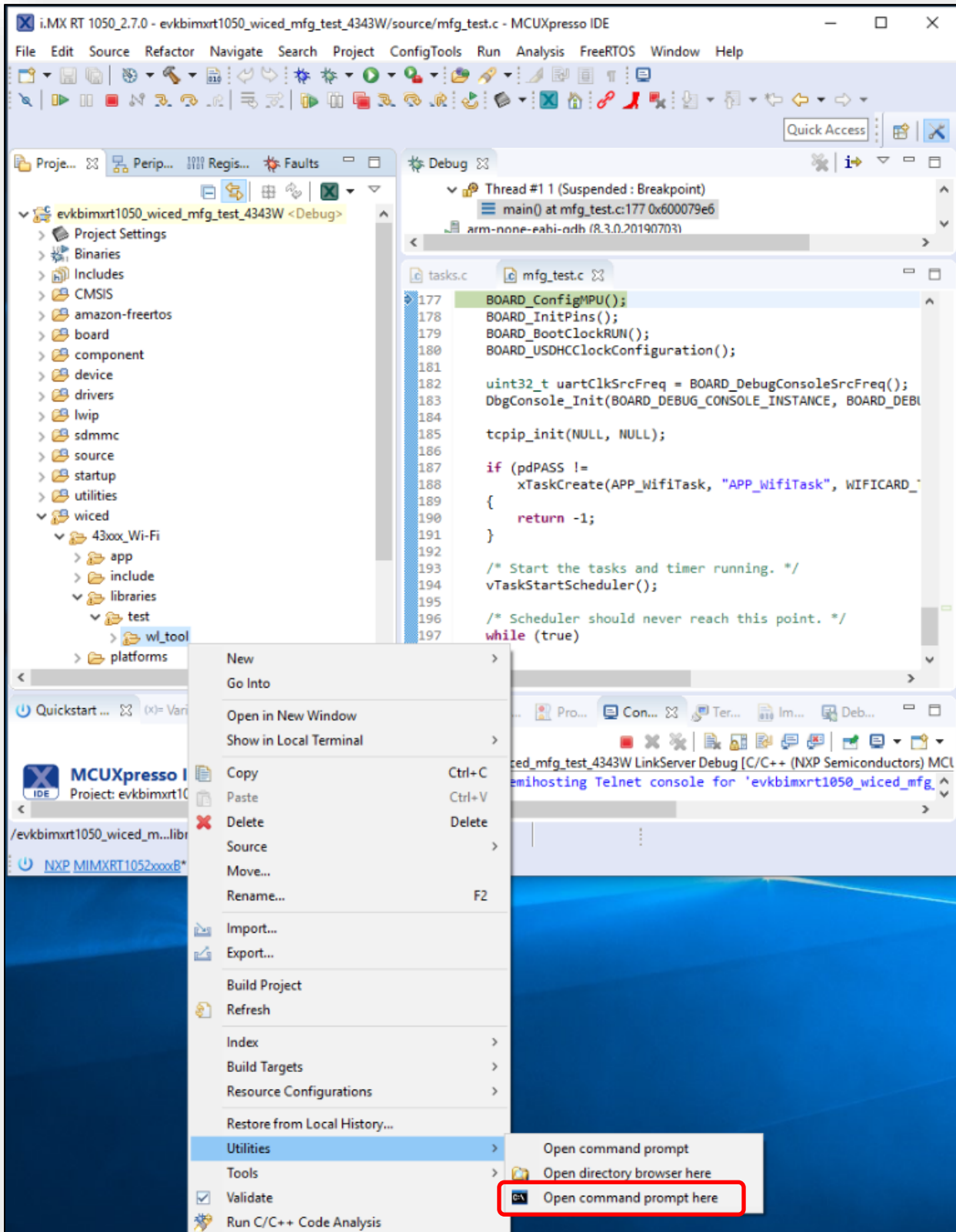
6. Click on “Debug” button in the Quickstart Panel.



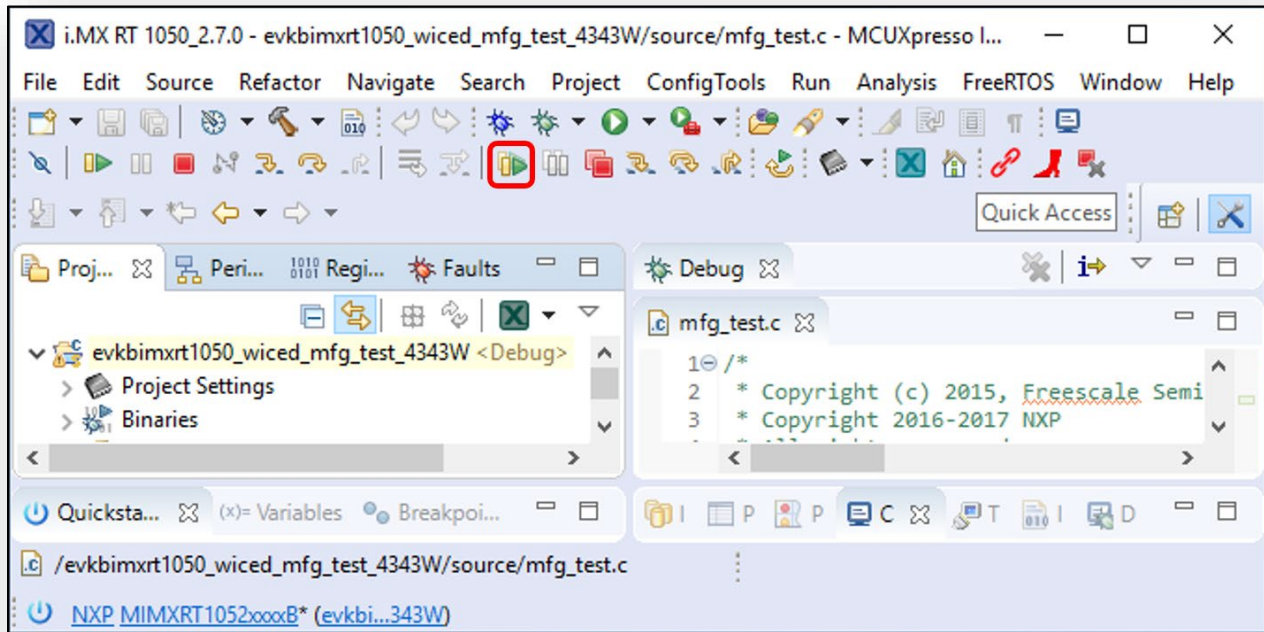
7. After the Debug process is complete, open the “Device Manager” in windows and go to “Ports (COM & LPT)” to check the mbed COM port number. In this case, COM 22 is our COM port number.



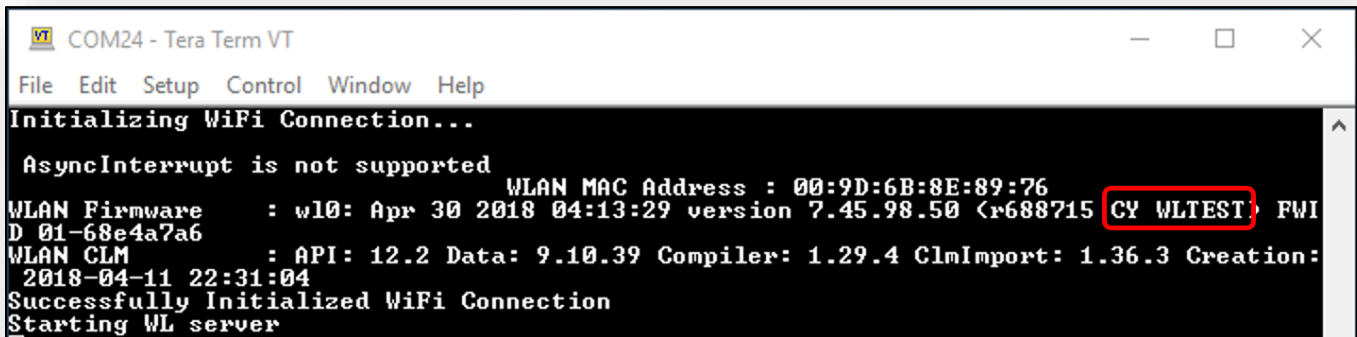
- Open evkbimxrt1050_wiced_mfg_test_4343W/wiced/43xxx_Wi-Fi/libraries/test/wl_tool and right click on wl_tool. Select Utilities/Open command prompt here and a command prompt window will come up.



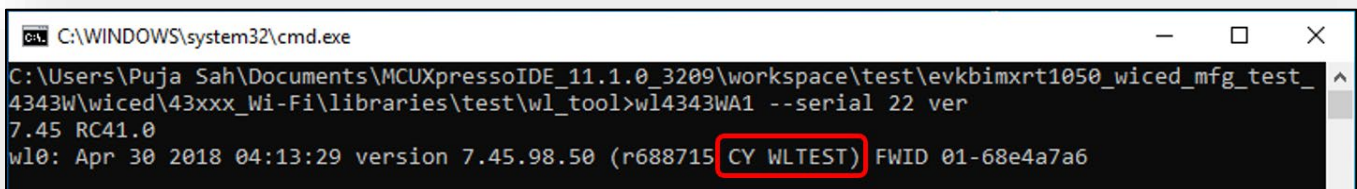
9. After the command prompt window shows up, click on “resume debug” in the MCUXpresso IDE.



10. To see logs from debug console, open the Tera Term with USB/UART converter port number (i.e., COM 24). You should see the output as below in the terminal window:

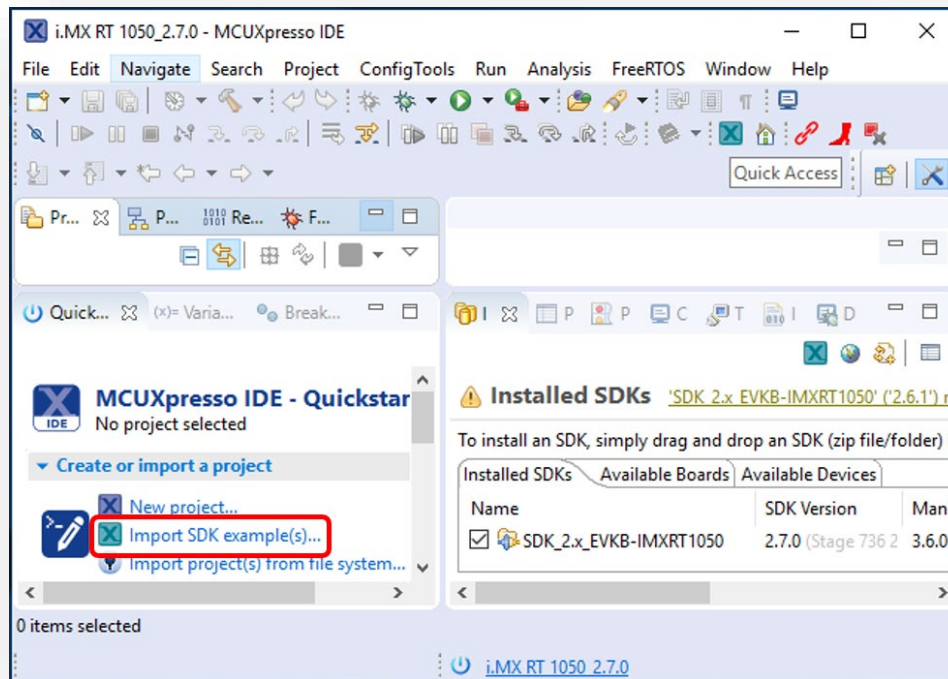


11. Type “wl4343WA1 --serial <COM port number> ver” and hit return. If you can see CY WLTEST in the output that means the manufacturing example is working fine.



7.4 Example wiced_mfg_test_43012

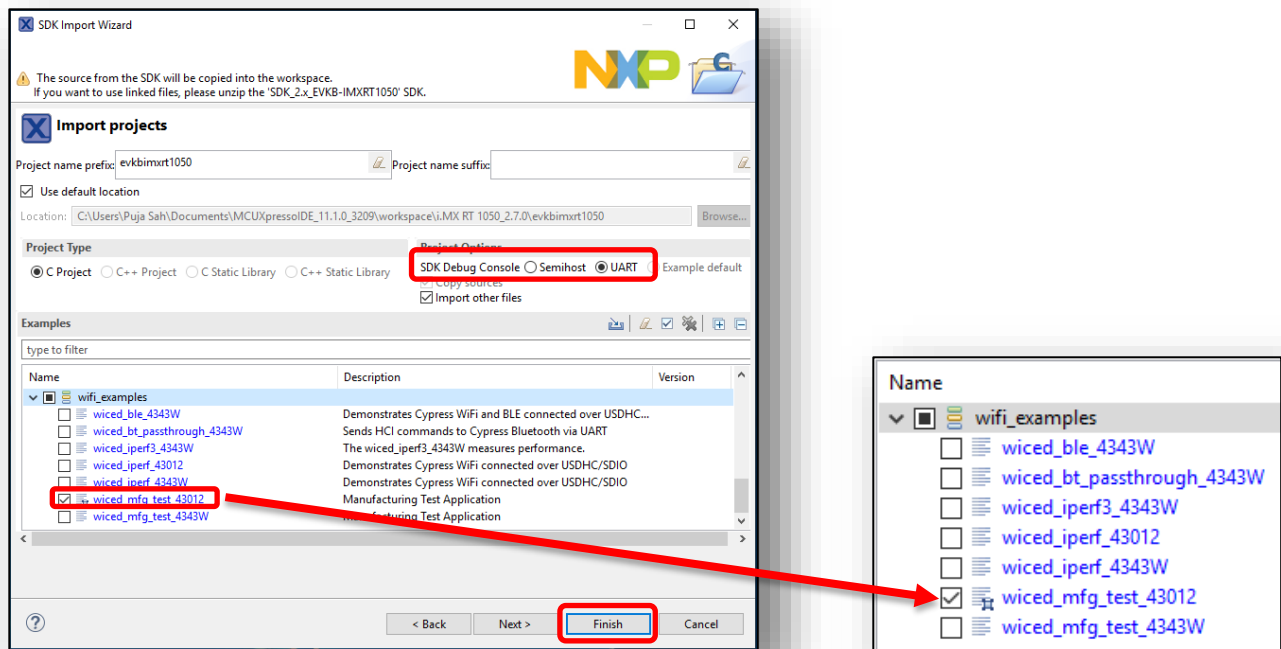
1. Click on “Import SDK example(s)...” in the Quickstart Panel.



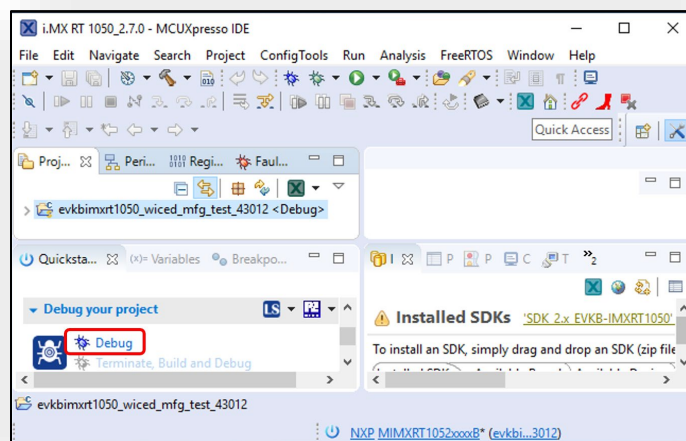
2. Select “evkbimxrt1050” board and click Next button.



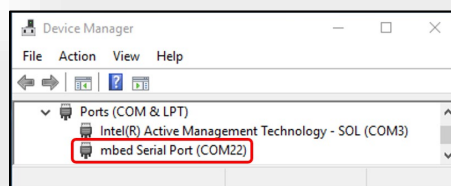
- Expand `wifi_examples` and select `wiced_mfg_test_43012`. Select UART for SDK Debug Console, then click Finish button.



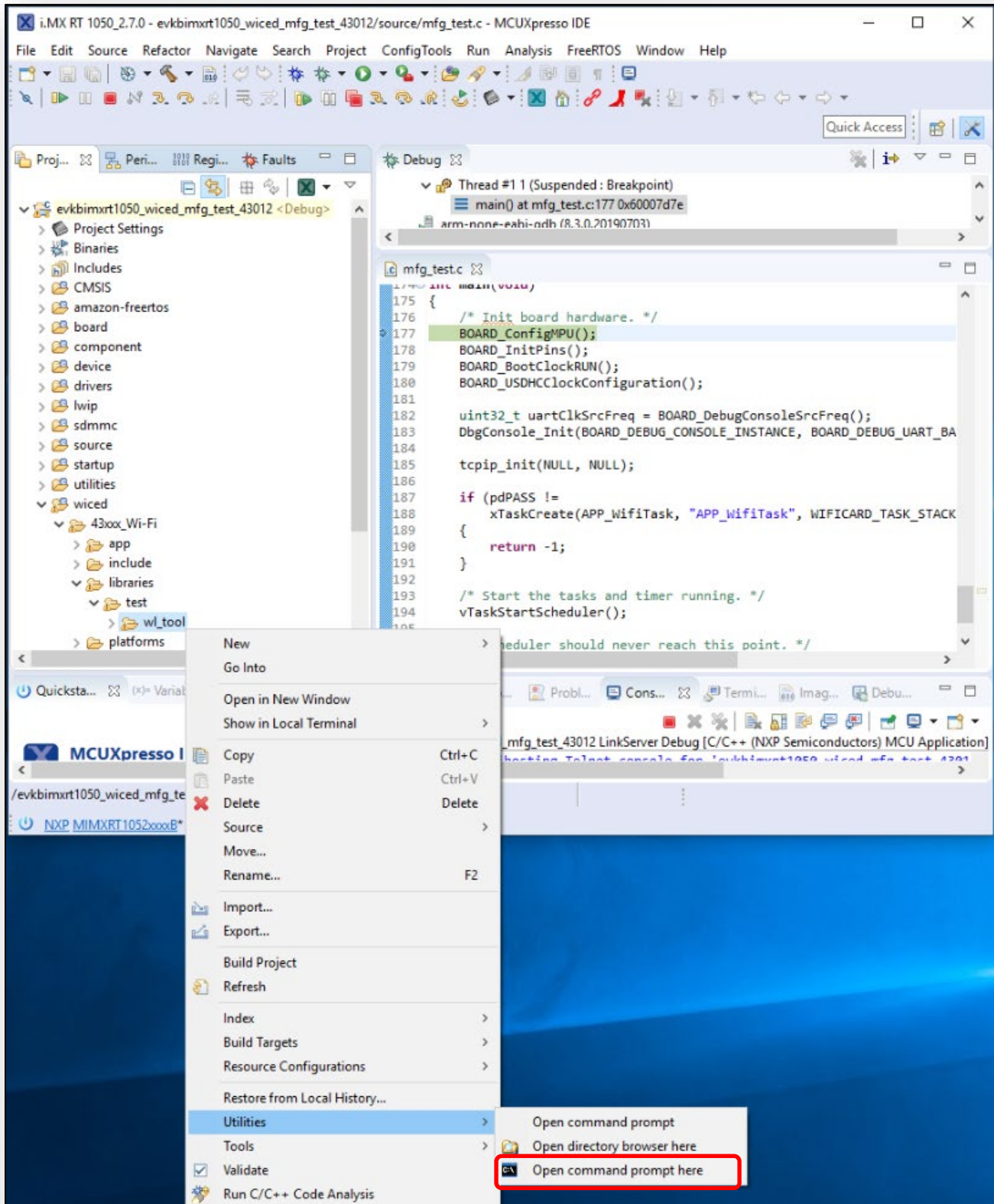
- Click on "Debug" button in the Quickstart Panel.



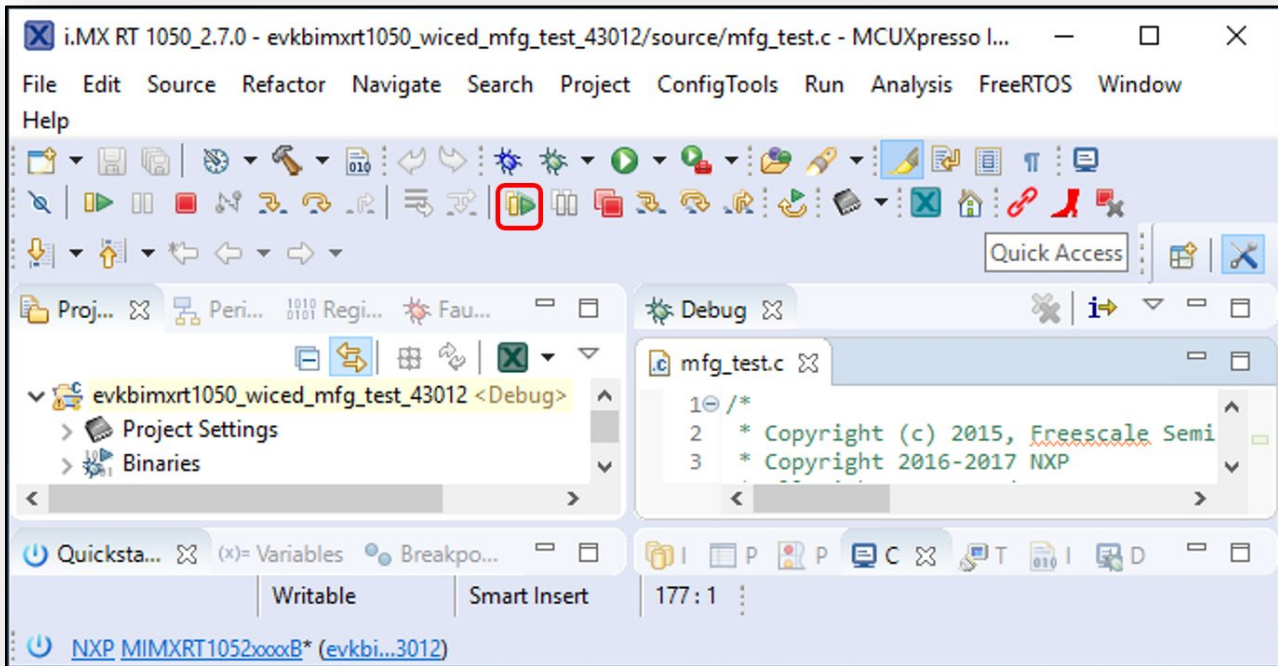
- After the Debug process is complete, open the "Device Manager" in windows and go to "Ports (COM & LPT)" to check the mbed COM port number. In this case, COM 22 is our COM port number.



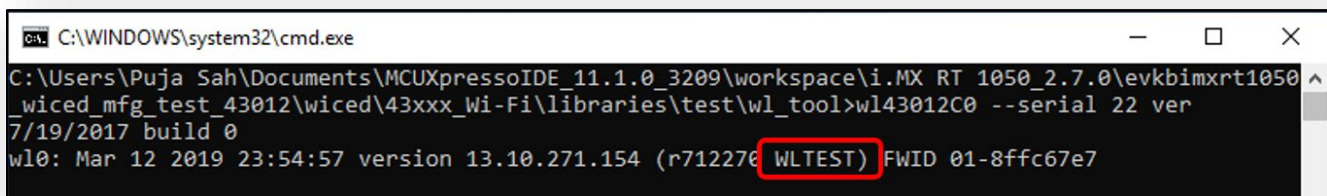
- Open evkbimxrt1050_wiced_mfg_test_43012/43xxx_Wi-Fi/libraries/test/wl_tool and right click on wl_tool. Select Utilities/Open command prompt here and a command prompt window will come up.



7. After the command prompt window shows up, click on “resume debug” in the MCUXpresso IDE.



8. Type “wl43012C0 --serial <COM port number> ver” and hit return. If you can see WLTEST in the output that means the manufacturing example is working fine.



7.5 Example wiced_bt_passthrough_4343W

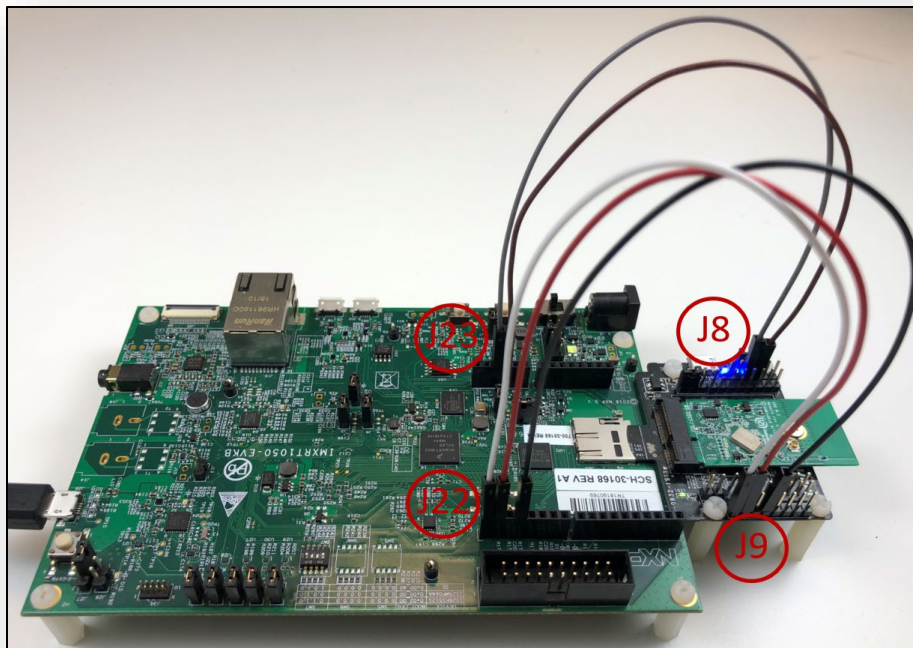
1. Hardware requirements:

- a. Connect the uSD-M.2 adapter with the NXP i.MX RT EVK as per **Table 5** and **Figure 9**.
- b. Set the uSD-M.2 adapter to run in 3.3V override mode:
 - i. For rev A adapter, J12 is closed for 3.3V VIO
 - ii. For rev B1 adapter, J12 is in 2-3 pos for 3.3V & J13 is in 1-2 pos for 3.3V VIO
- c. Connect the board with PC by using USB/UART converter (Optional):
 - board UART RX (pin 8 on J22) - connect it to TX pin on converter
 - board UART TX (pin 7 on J22) - connect it to RX pin on converter
 - board GND (pin 7 on J24)

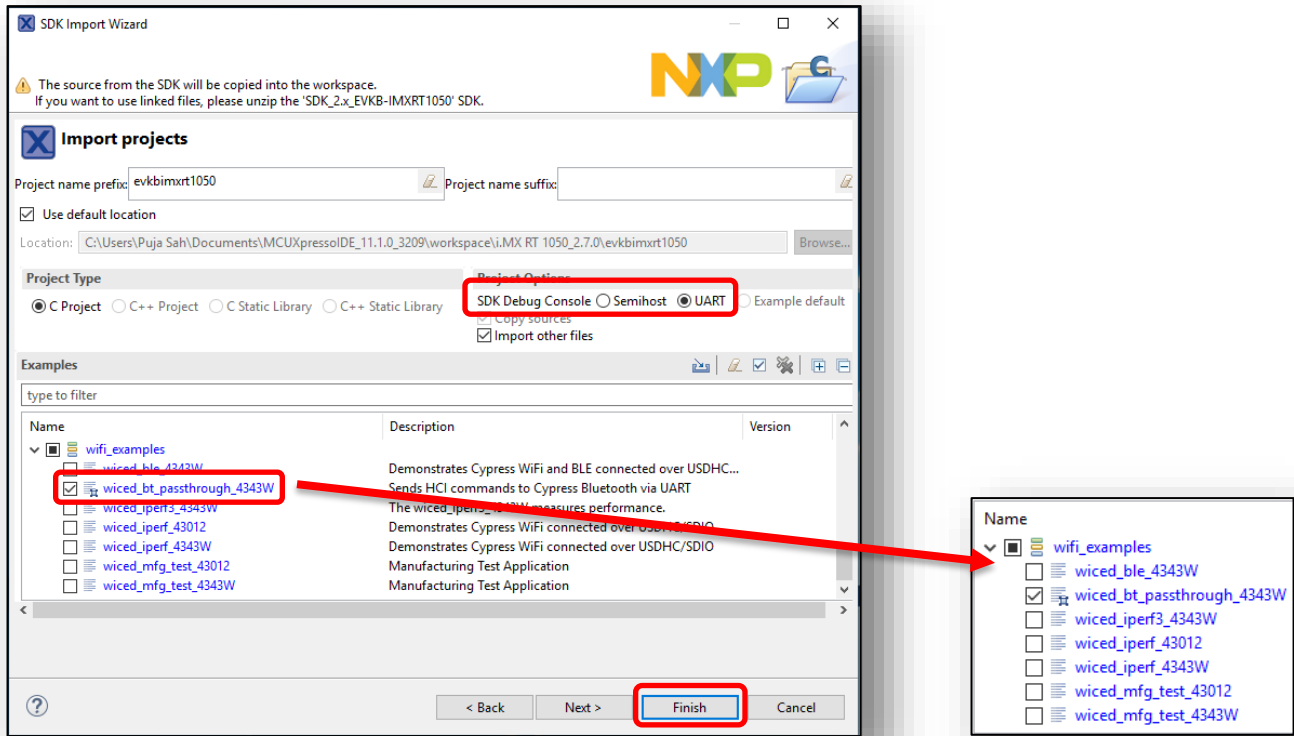
Table 5: Additional connections to run BT passthrough

Signal Name	uSD-M.2 Adapter Header/Pin	i.MX RT EVK Pin	i.MX RT EVK Signal
BT_UART_TXD_HOST	J9 / Pin 1	J22 / 1	LPUART3_RX
BT_UART_RXD_HOST	J9 / Pin 2	J22 / 2	LPUART3_TX
BT_REG_ON_HOST	J9 / Pin 4	J22 / 4	GPIO1_IO24
BT_UART_RTS_HOST	J8 / Pin 3	J23 / 3	LPUART3_CTS_B
BT_UART_CTS_HOST	J8 / Pin 4	J23 / 4	LPUART3_RTS_B

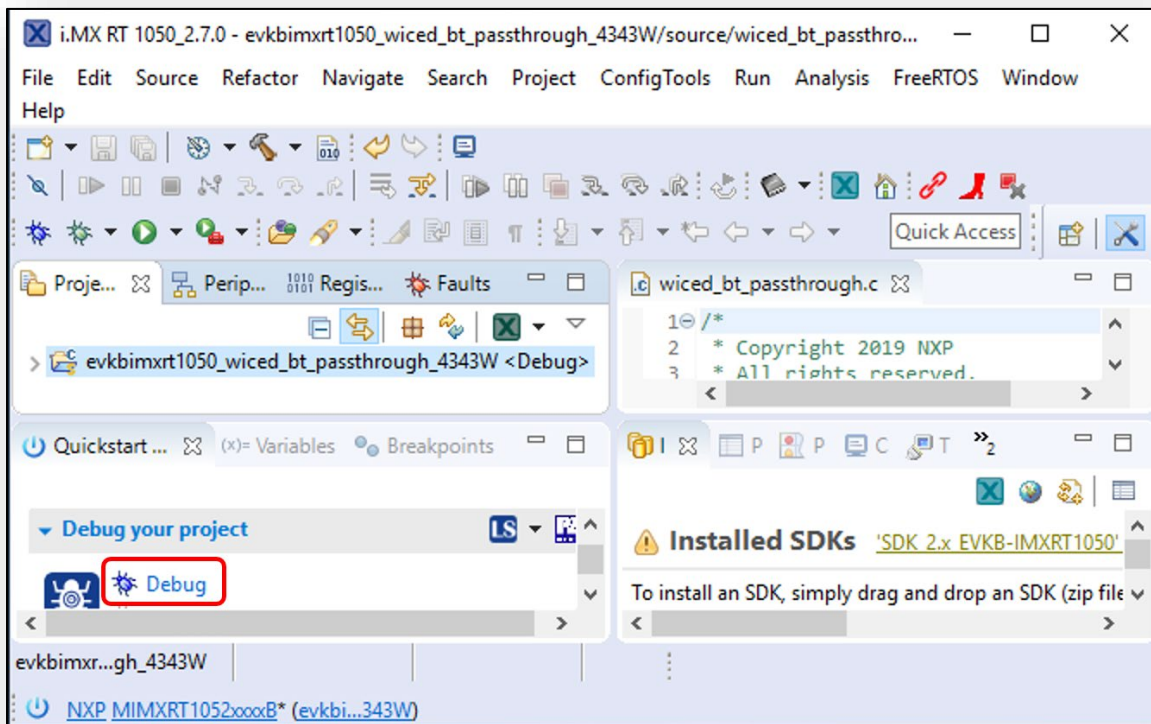
Figure 9: NXP i.MX RT 1050 EVK hardware settings to run BT passthrough



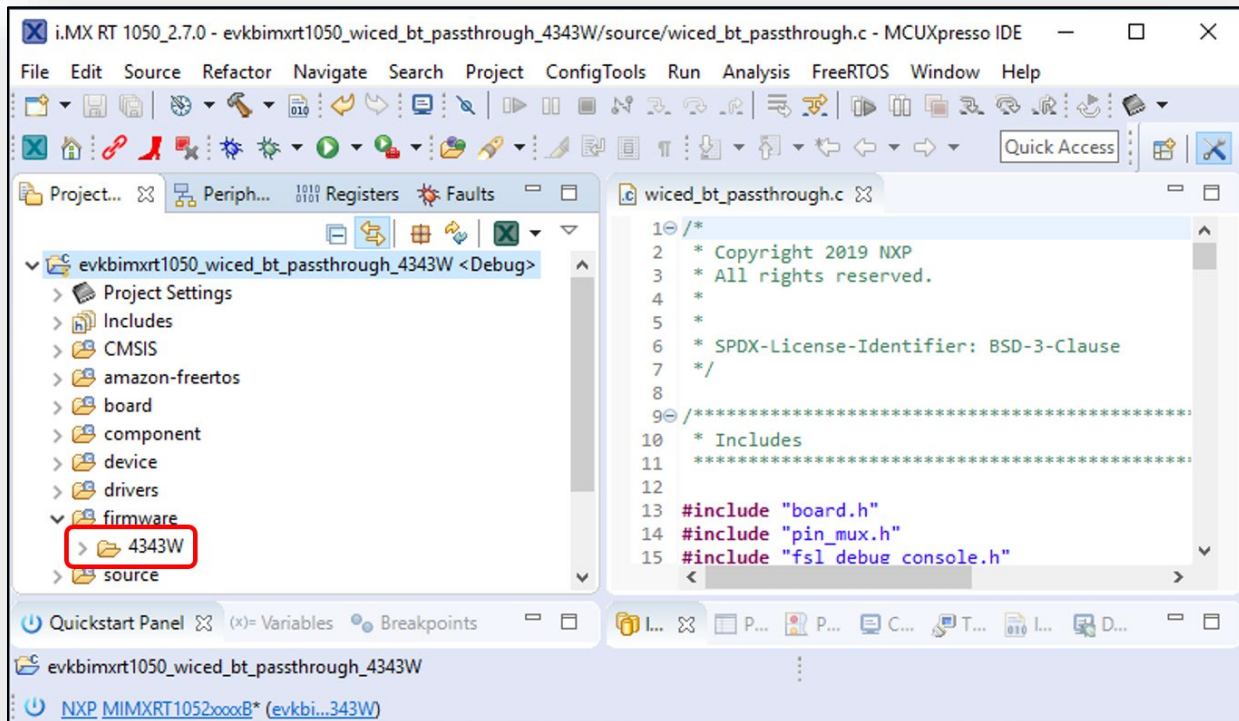
- Click on “Import SDK example(s)...” in the Quickstart Panel. Select “evkbimxrt1050” board and click Next button. Expand wifi_examples and select wiced_bt_passthrough_4343W. Select UART for SDK Debug Console, then click Finish button.



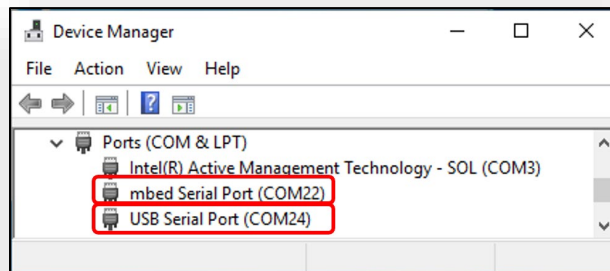
- Click on “Debug” button.



- Copy and paste the folder 4343W (evkbimxrt1050_wiced_bt_passthrough_4343W/firmware/4343W) in your desktop.

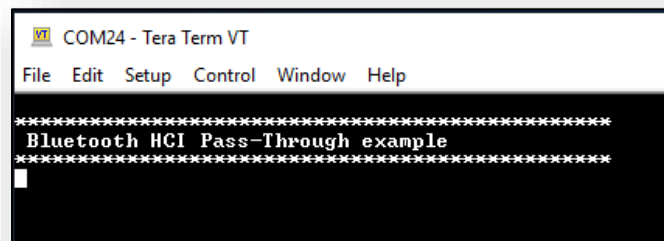


- Open the device manager and check for the port numbers.

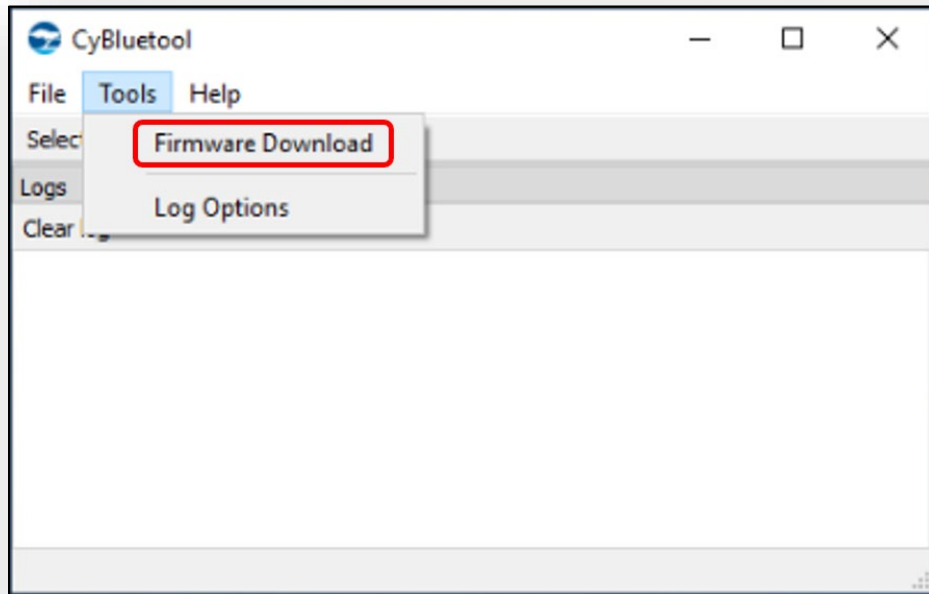


- Open Tera Term at the USB serial port number (i.e., COM 24). Run debug in the IDE and then do resume debug. You will see below output in the terminal window.

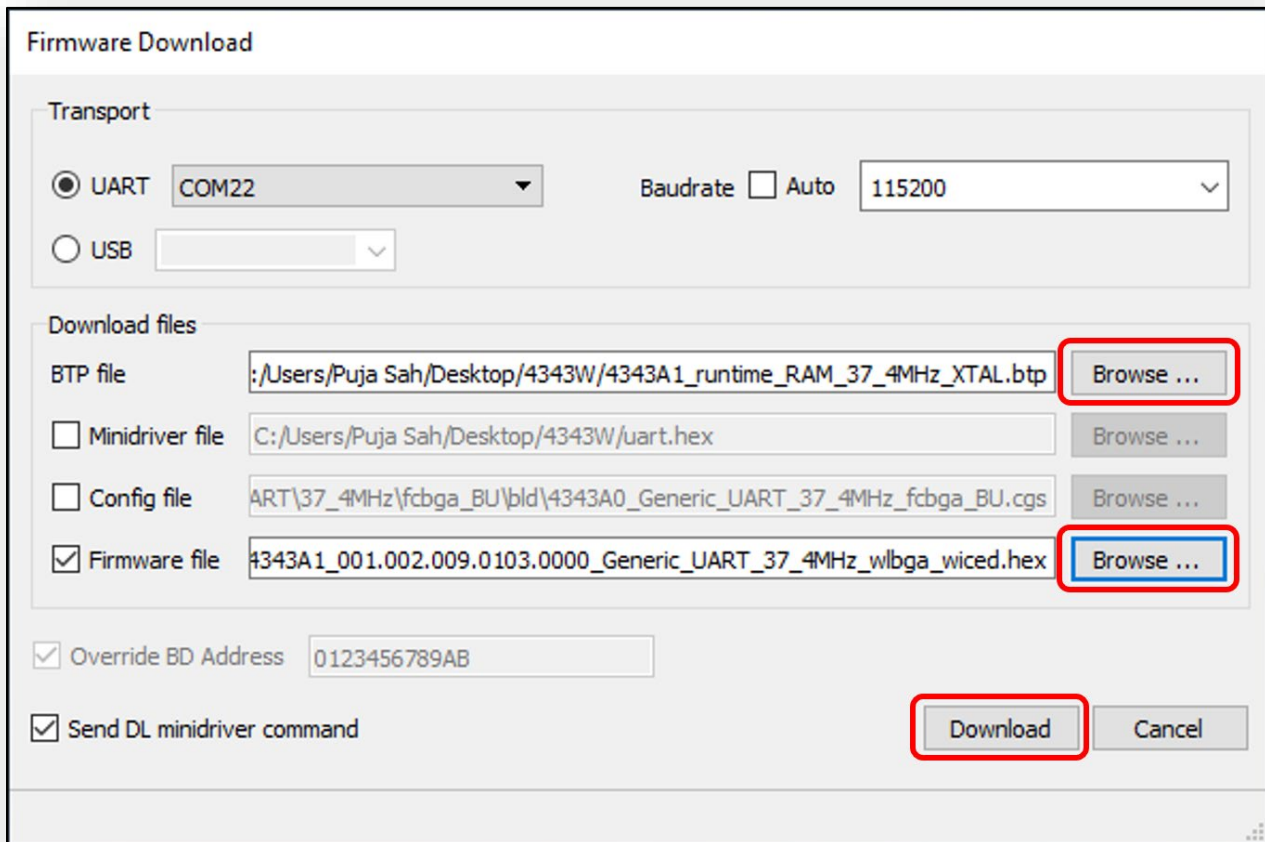
NOTE: This step is optional. This step is required only when USB/UART converter is used.



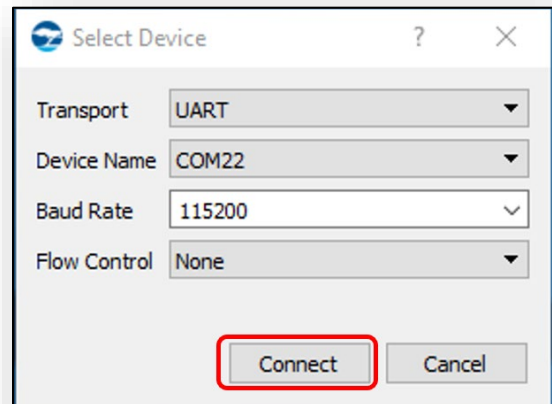
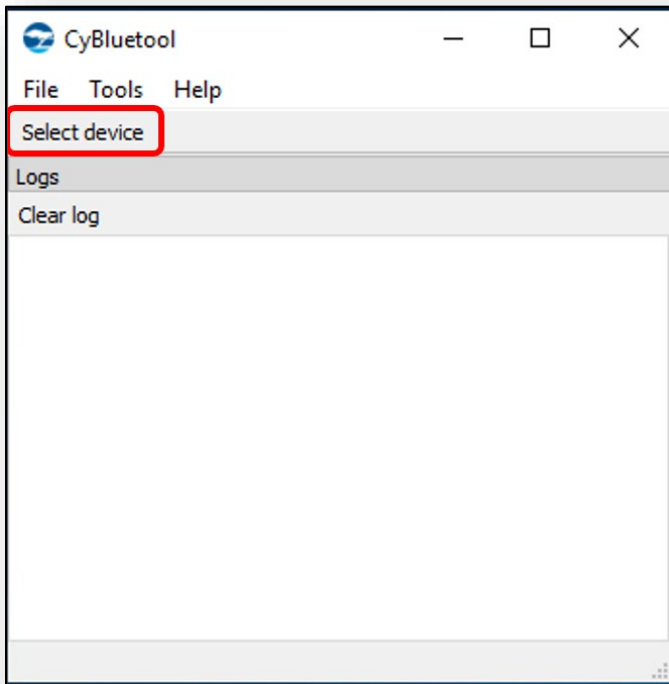
7. Open the CyBluetool and select Firmware Download.



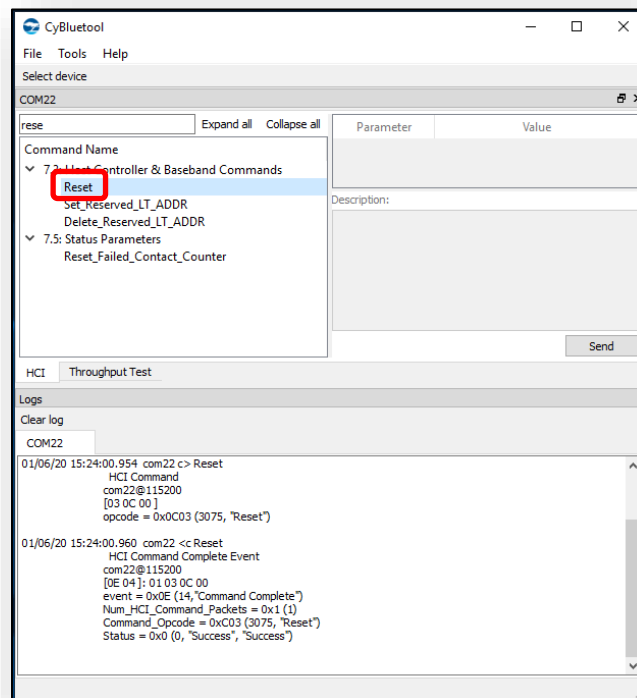
8. The below window will come up. Select the UART as mbed serial port number (i.e., COM 22). Browse for the BTP file and go to the 4343W folder location in Desktop. Also browse for the Firmware file and then Click on "Download".



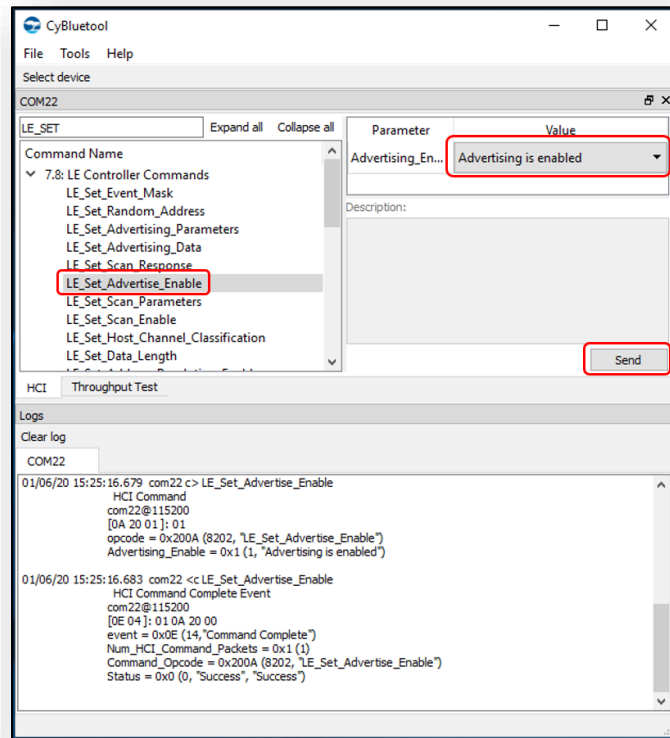
9. Click on "Select device" in CyBluetool. Use the following settings in the dialog window:
- Transport: UART
 - Device Name: the name of the mbed serial port
 - Baud Rate: 115200
 - Flow Control: None



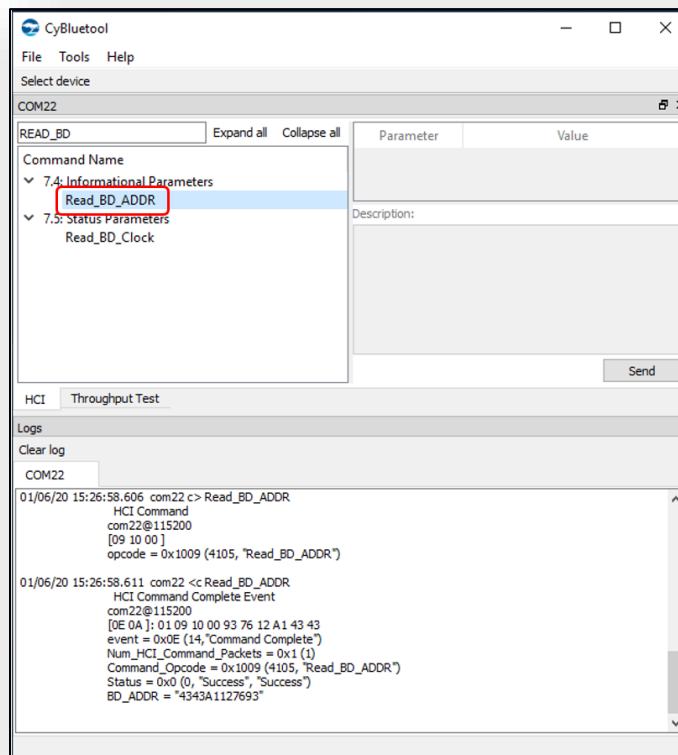
10. Using CyBluetool, send command Reset to verify its successful working.



11. Send command LE_Set_Advertise_Enable with parameter Advertising_Enable set to "Advertising is enabled".



12. Send command Read_BD_ADDR by double clicking on "Read_BD_ADDR".



7.6 Example wiced_ble_4343W

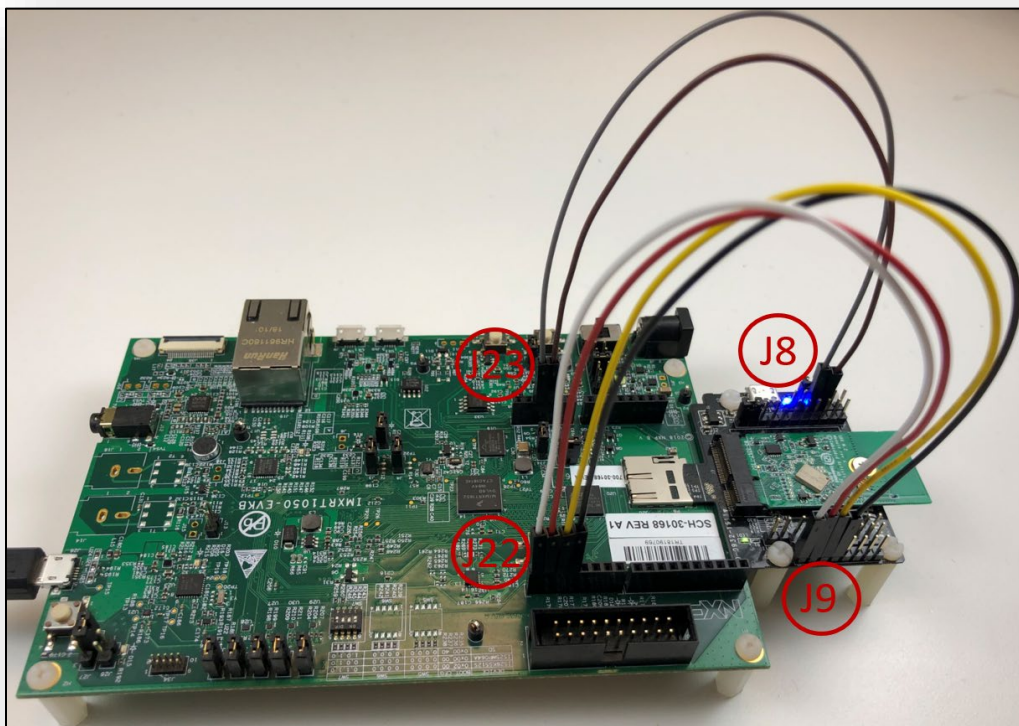
1. Hardware Requirements:

- a. Connect the uSD-M.2 adapter with the NXP i.MX RT EVK as per **Table 5** and **Figure 10**.
- b. Run at 3.3V:
 - i. For rev A adapter, J12 is closed for 3.3V VIO
 - ii. For rev B1 adapter, J12 is in 2-3 pos for 3.3V & J13 is in 1-2 pos for 3.3V VIO

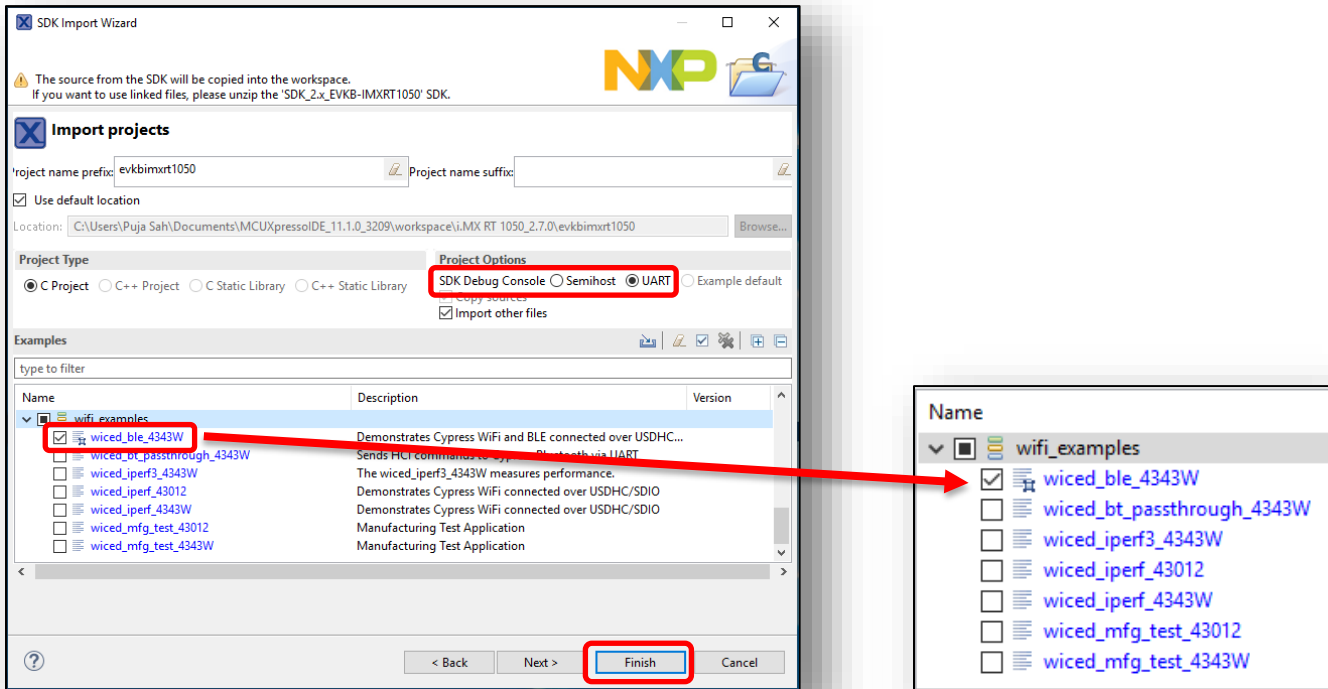
Table 6: Additional connections to run BLE

Signal Name	uSD-M.2 Adapter Header/Pin	i.MX RT EVK Pin	i.MX RT EVK Signal
BT_UART_TXD_HOST	J9 / Pin 1	J22 / 1	LPUART3_RX
BT_UART_RXD_HOST	J9 / Pin 2	J22 / 2	LPUART3_TX
WL_REG_ON_HOST	J9 / Pin 3	J22 / 3	GPIO1_IO11
BT_REG_ON_HOST	J9 / Pin 4	J22 / 4	GPIO1_IO24
BT_UART_RTS_HOST	J8 / Pin 3	J23 / 3	LPUART3_CTS_B
BT_UART_CTS_HOST	J8 / Pin 4	J23 / 4	LPUART3_RTS_B

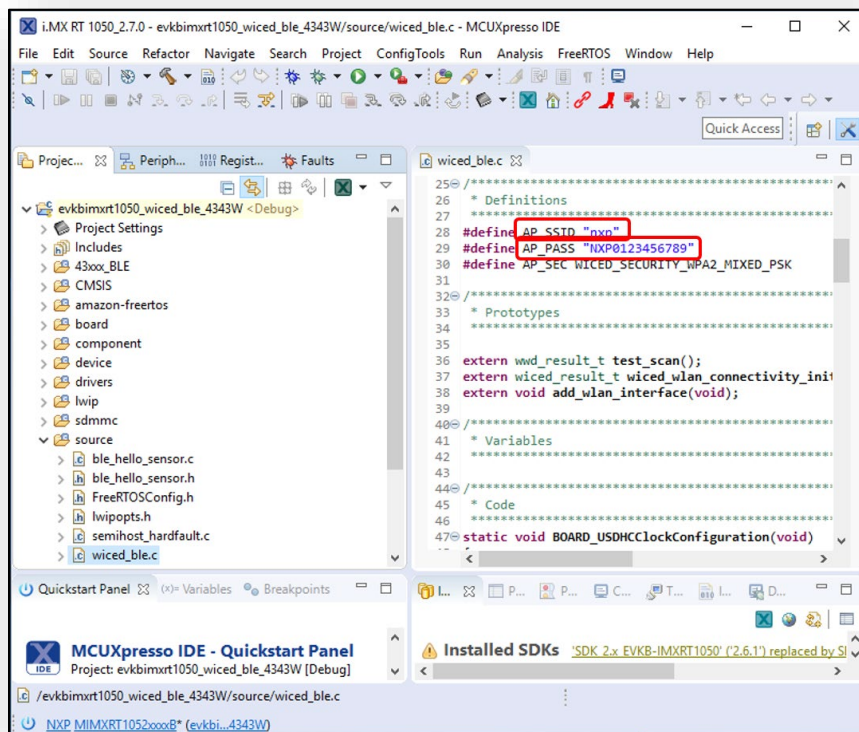
Figure 10: NXP i.MX RT 1050 EVK hardware settings to run BLE



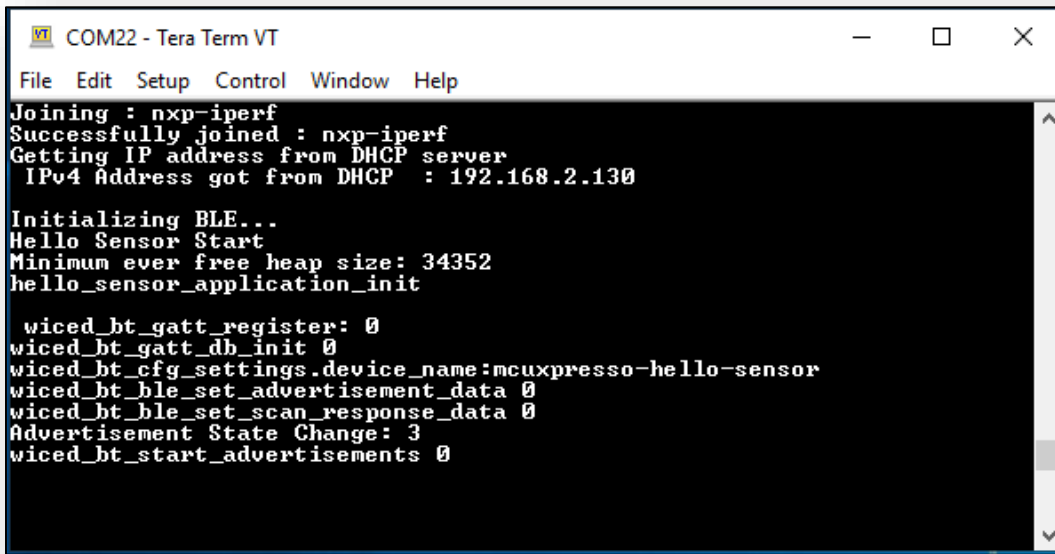
- Click on “Import SDK example(s)...” in the Quickstart Panel. Select “evkbimxrt1050” board and click Next button. Expand wifi_examples and select “wiced_ble_4343W”. Select UART for SDK Debug Console, then click Finish button.



- Open evkbimxrt1050_wiced_ble_4343W/source/wiced_ble.c to modify AP_SSID, AP_PASS if necessary.



- Run debug in the IDE. Once the app is booted, BLE advertising starts with the device name “mcuxpresso-hello-sensor”.

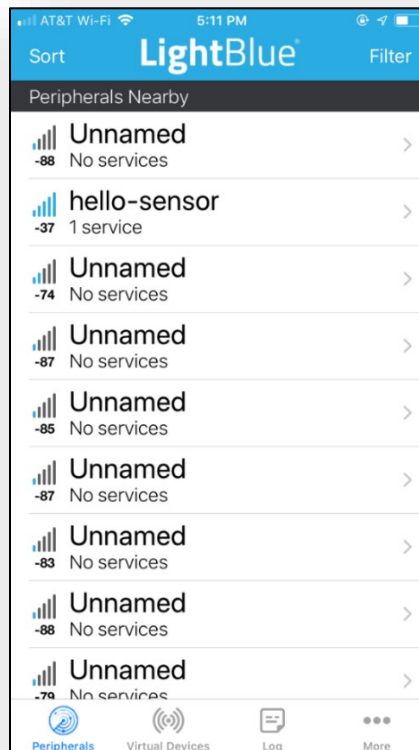


```
COM22 - Tera Term VT
File Edit Setup Control Window Help
Joining : nxp-iperf
Successfully joined : nxp-iperf
Getting IP address from DHCP server
IPv4 Address got from DHCP : 192.168.2.130

Initializing BLE...
Hello Sensor Start
Minimum ever free heap size: 34352
hello_sensor_application_init

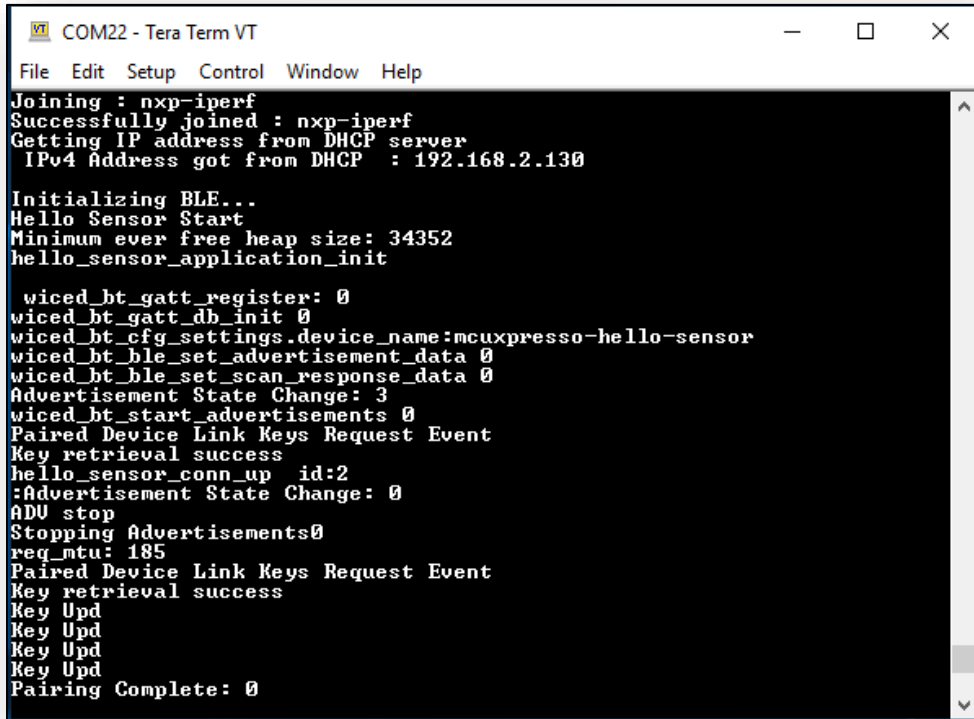
wiced_bt_gatt_register: 0
wiced_bt_gatt_db_init 0
wiced_bt_cfg_settings.device_name:mcuxpresso-hello-sensor
wiced_bt_ble_set_advertisement_data 0
wiced_bt_ble_set_scan_response_data 0
Advertisement State Change: 3
wiced_bt_start_advertisements 0
```

- Download and install any of the BLE scanner apps such as LightBlue / BLE Scanner / nRF Connect to the BLE enabled device.
- Open the app and scan for devices. Scan results shows “mcuxpresso-hello-sensor”. You will see below output in your phone.



7. Connect to mcuxpresso-hello-sensor to view its services. When paired to the device, you will see below outputs:

In Tera Term:

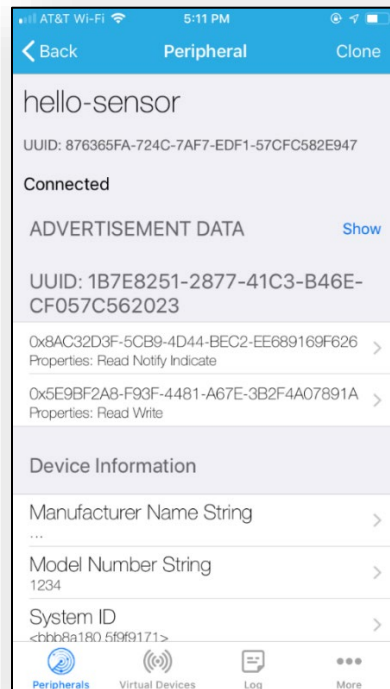


```
COM22 - Tera Term VT
File Edit Setup Control Window Help
Joining : nxp-iperf
Successfully joined : nxp-iperf
Getting IP address from DHCP server
IPv4 Address got from DHCP : 192.168.2.130

Initializing BLE...
Hello Sensor Start
Minimum ever free heap size: 34352
hello_sensor_application_init

wiced_bt_gatt_register: 0
wiced_bt_gatt_db_init 0
wiced_bt_cfg_settings.device_name:mcuxpresso-hello-sensor
wiced_bt_ble_set_advertisement_data 0
wiced_bt_ble_set_scan_response_data 0
Advertisement State Change: 3
wiced_bt_start_advertisements 0
Paired Device Link Keys Request Event
Key retrieval success
hello_sensor_conn_up id:2
:Advertisement State Change: 0
ADU stop
Stopping Advertisements0
req_mtu: 185
Paired Device Link Keys Request Event
Key retrieval success
Key Upd
Key Upd
Key Upd
Key Upd
Pairing Complete: 0
```

In Phone:

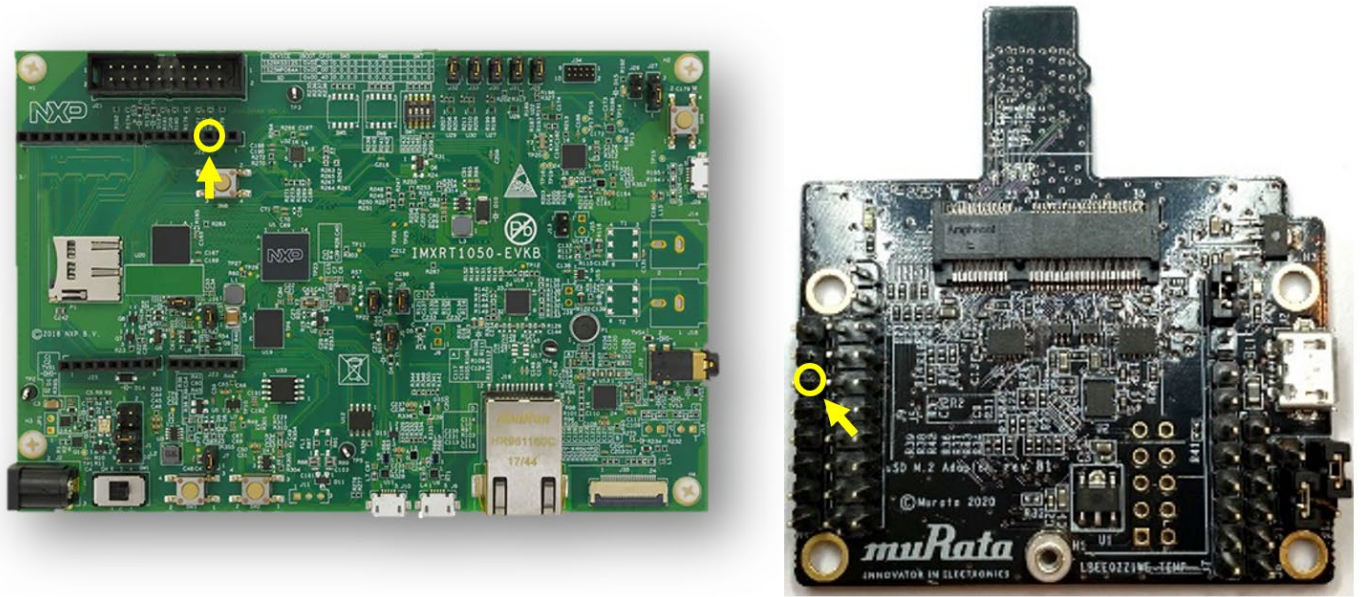


7.7 Example wiced_webconfig_4343W

1. Hardware Requirement:

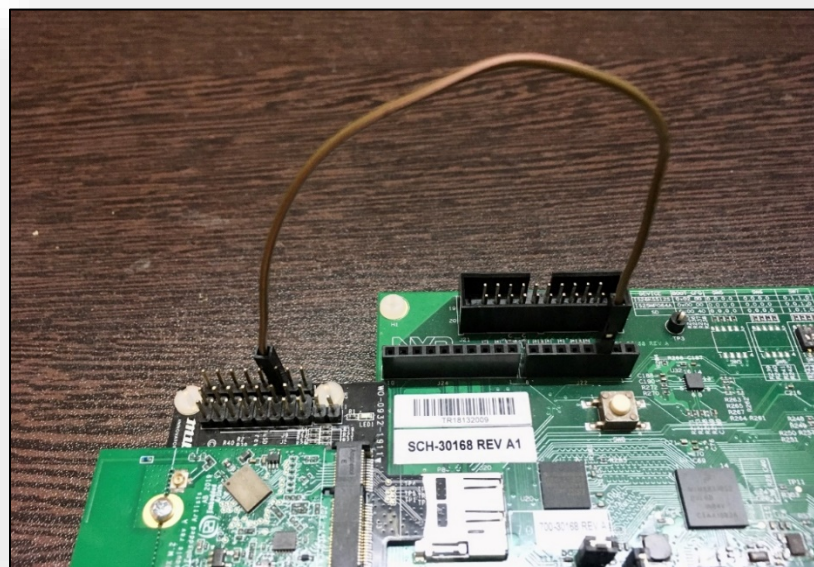
- a) Connect pin3 of evkbimxrt1050 board's connector J22 to pin 3 of Murata uSD M.2 Adapter's connector J9, using plug-to-receptable cables:

Figure 11: Additional cabling pins

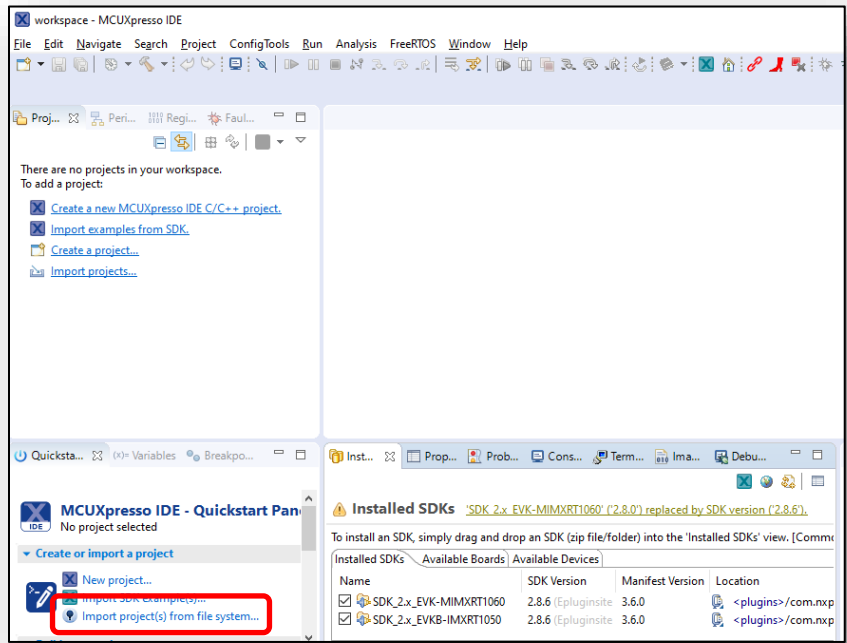


See the figure below for this additional cabling.

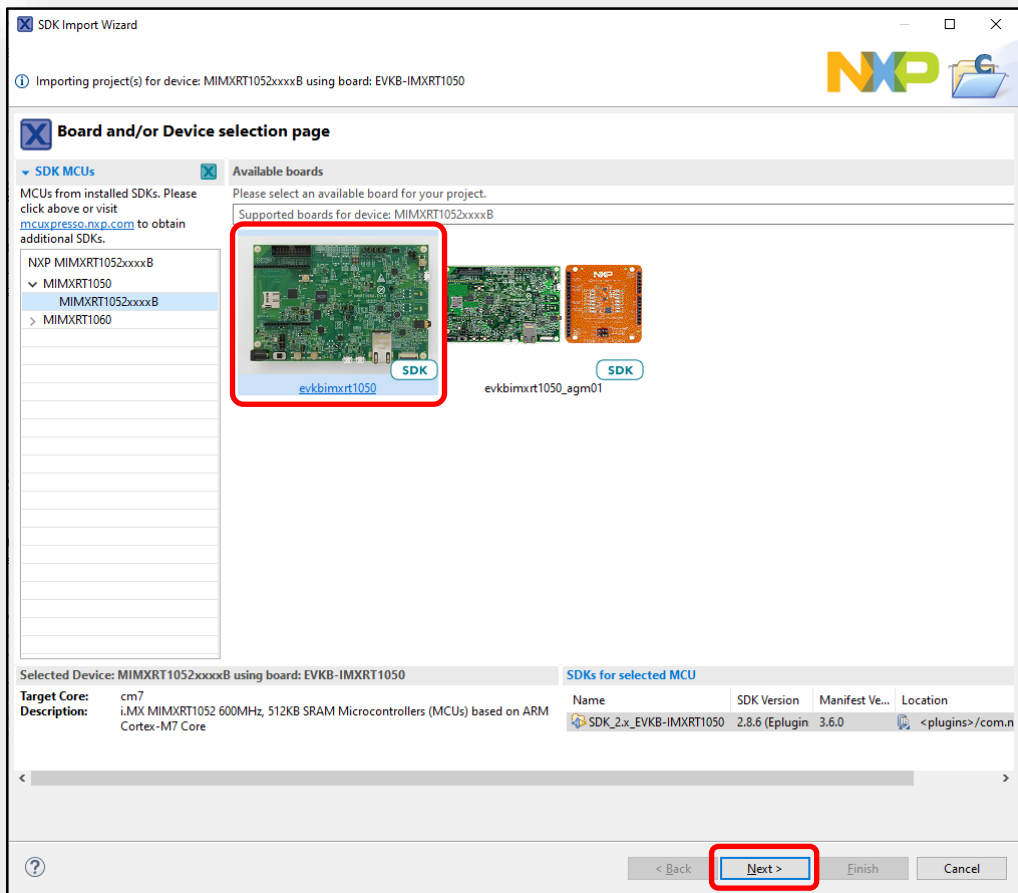
Figure 12: Additional cabling



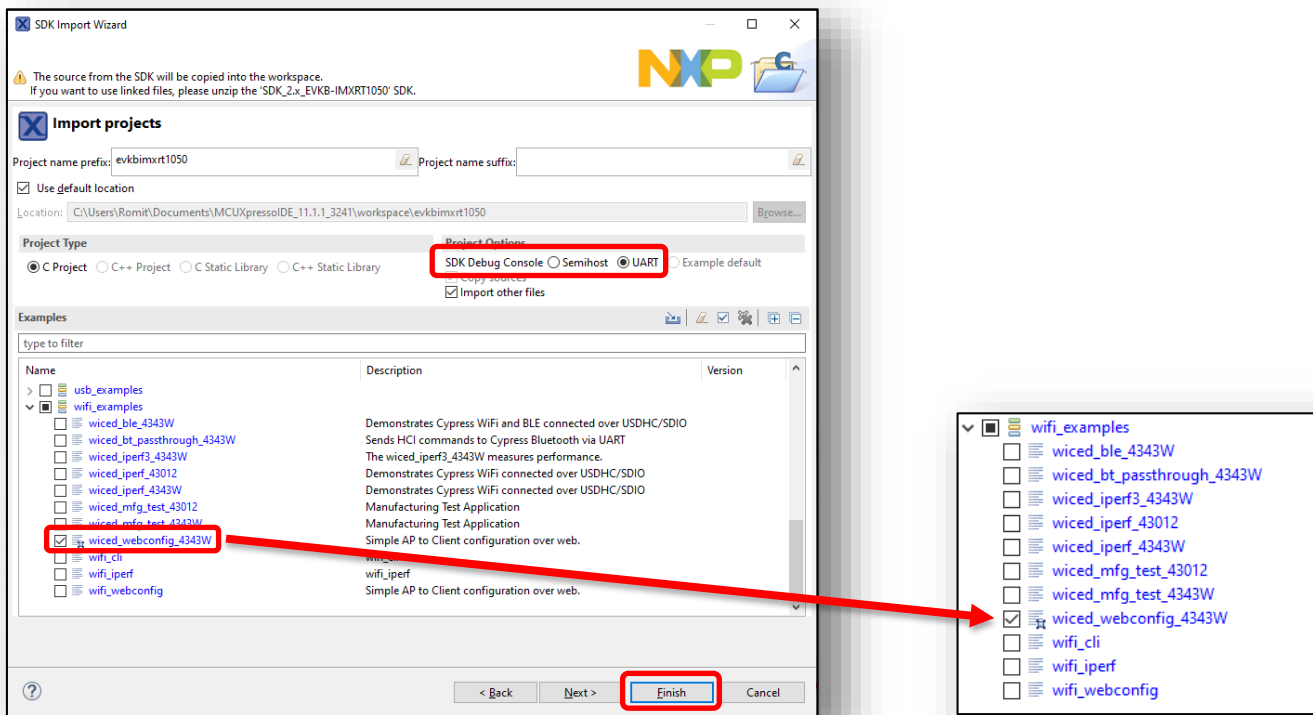
2. Click on “Import SDK example(s)...” in the Quickstart Panel.



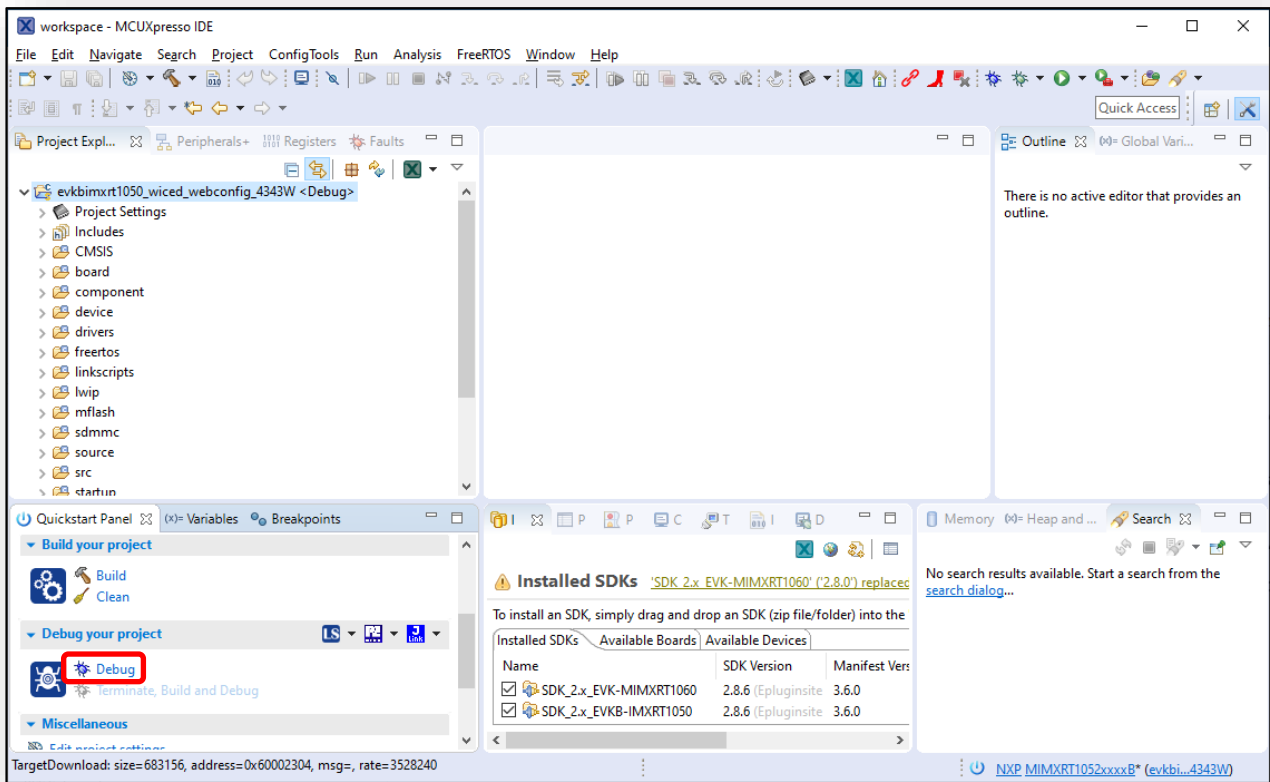
3. Select “evkbimxrt1050” board and click Next button.



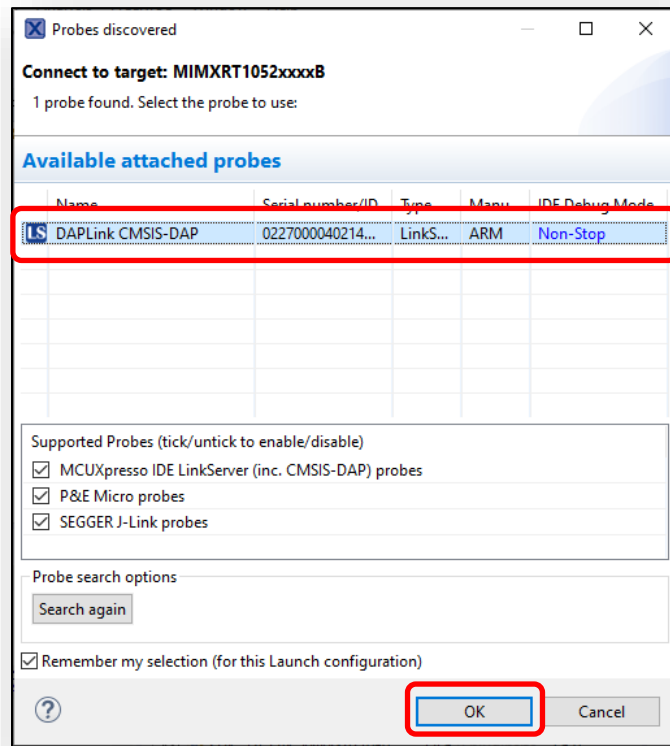
- Expand `wifi_examples` and select `wiced_webconfig_4343W`. Select `UART` for SDK Debug Console, then click `Finish` button.



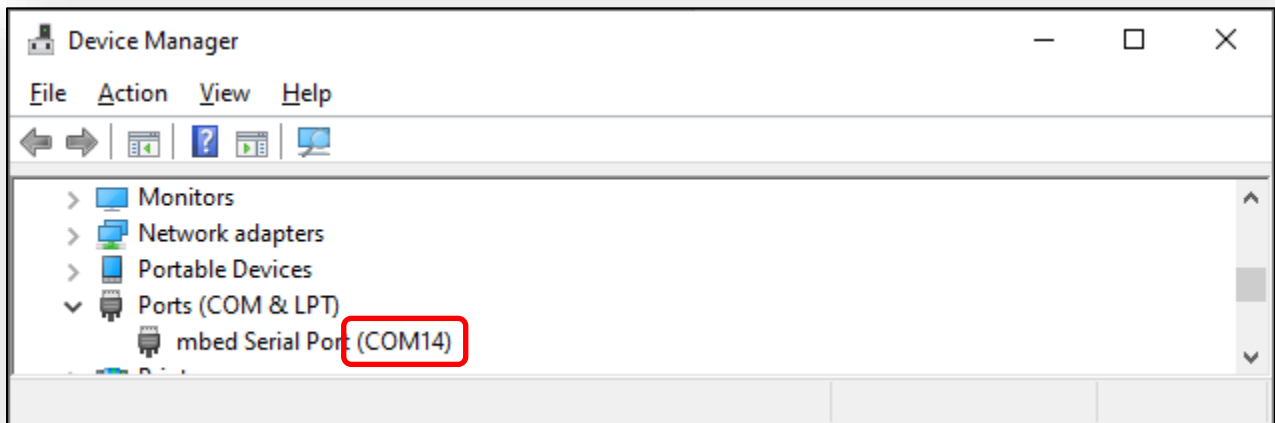
- Click `Debug` in the Quickstart Panel.



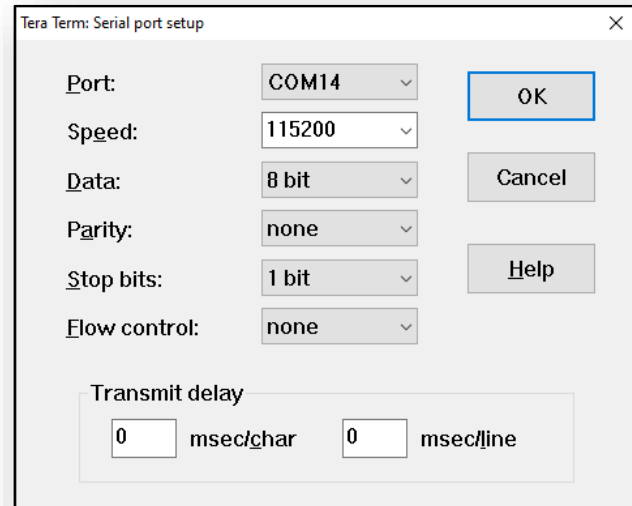
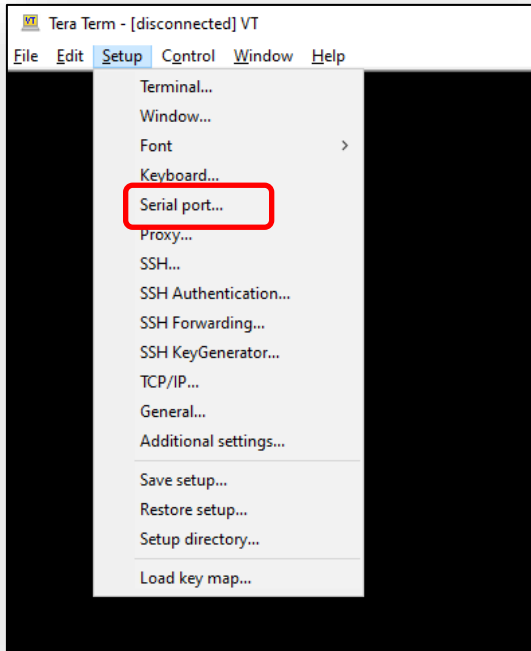
6. For the very first attempt, you need to select the appropriate JTAG adapter. Select the adapter and click OK, and then wait for a while for the build. You can see the log on console window.



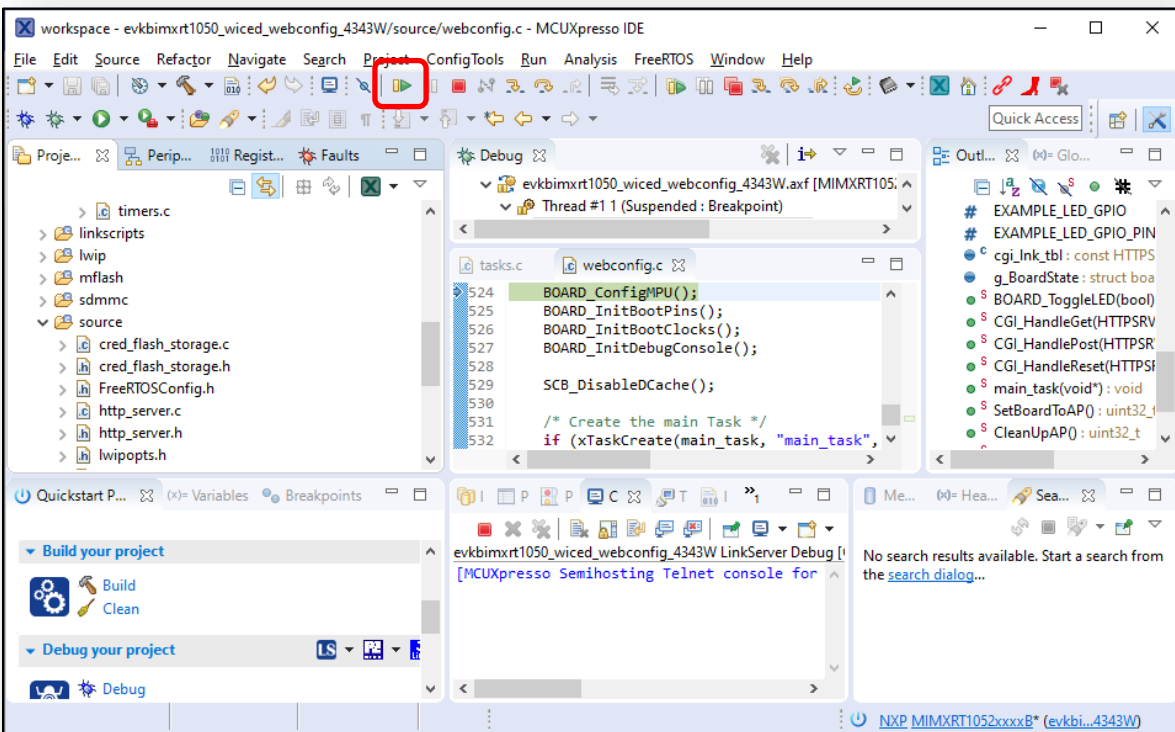
7. After the Debug process is complete, open the “Device Manager” in windows and go to “Ports (COM & LPT)” to check the mbed COM port number. In this case, COM 14 is our COM port number.



8. Now the webconfig example is ready to run. Open Tera Term on the appropriate COM port (i.e. COM 14 in this case). Configure port for 115200 bps, 8 bits data, no parity, and 1 stop bit (115200/8/N/1).



9. Click Resume button in MCUXpresso.



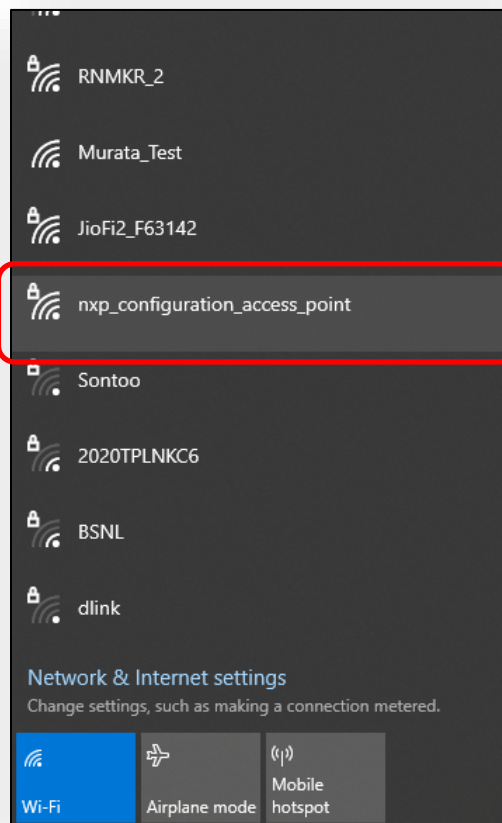
10. You should see this output from i.MX RT on Tera Term.

```
COM14 - Tera Term VT
File Edit Setup Control Window Help

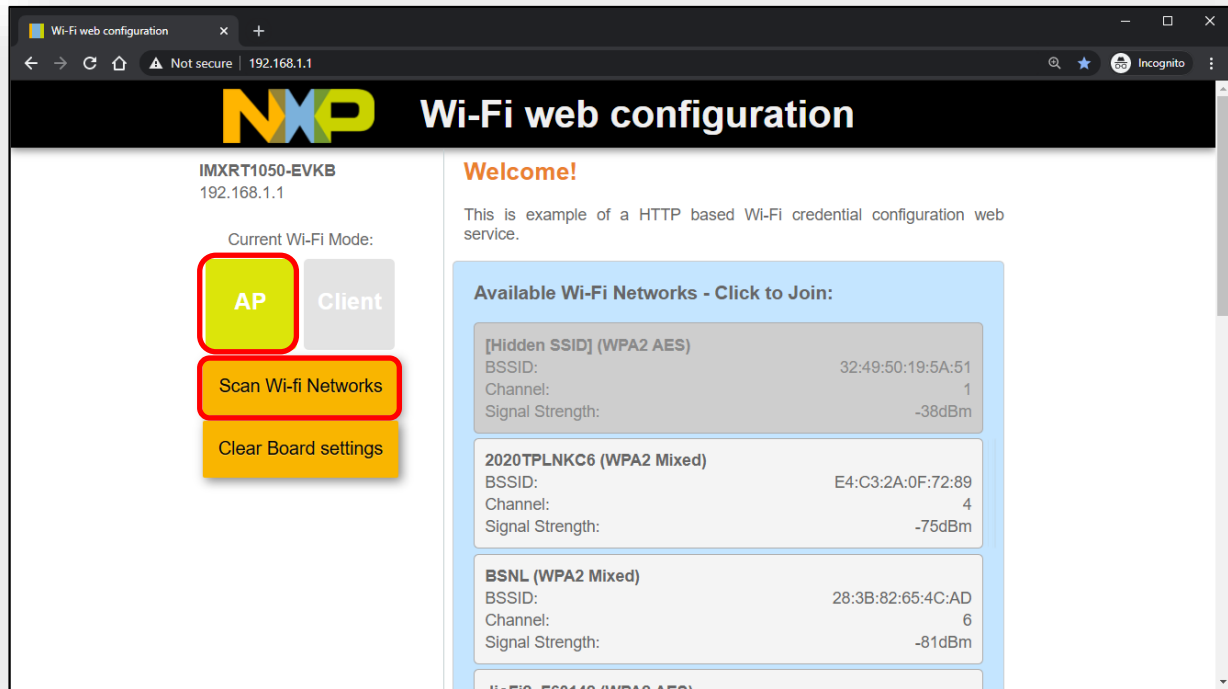
Starting webconfig DEMO
[i] Trying to load data from mflash.
[i] Nothing stored yet
[i] Initializing WiFi connection...

AsyncInterrupt is not supported
WLAN MAC Address : A0:C9:A0:3C:C5:64
WLAN Firmware   : wl0: Oct 22 2019 01:57:42 version 7.45.98.94 (r723000 CY) FWID 01-73a5ed62
WLAN CLM        : API: 12.2 Data: 9.10.39 Compiler: 1.29.4 ClmImport: 1.36.3 Creation: 2018-02-12 04:00:50
[i] Successfully initialized WiFi module
Starting Access Point: SSID: nxp_configuration_access_point, Chnl: 1
Now join that network on your device and connect to this IP: 192.168.1.1
```

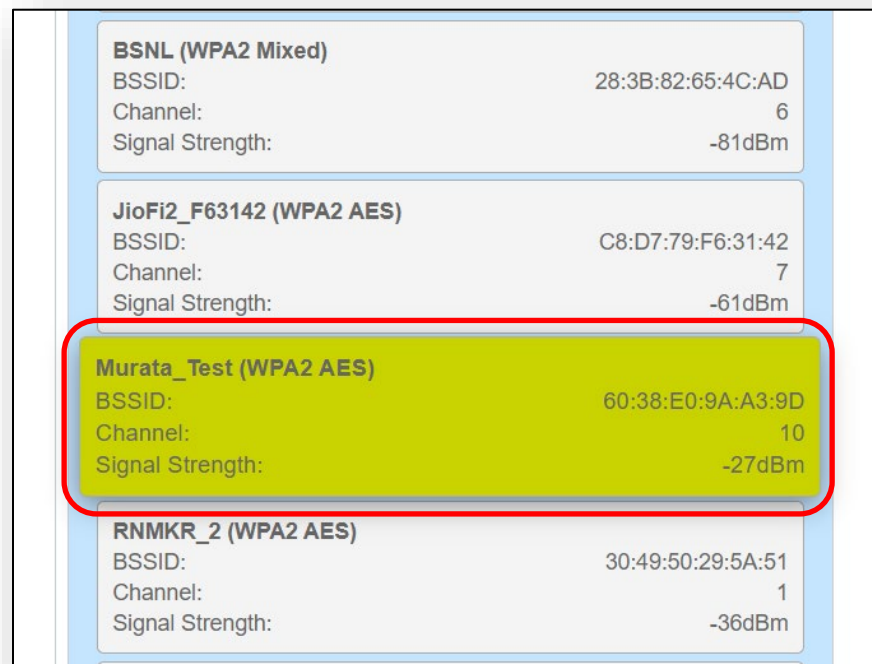
11. From a wireless client device (can be a laptop, or a phone), search for available wireless networks and connect to “nxp_configuration_access_point” SSID. The password is “NXP0123456789”.



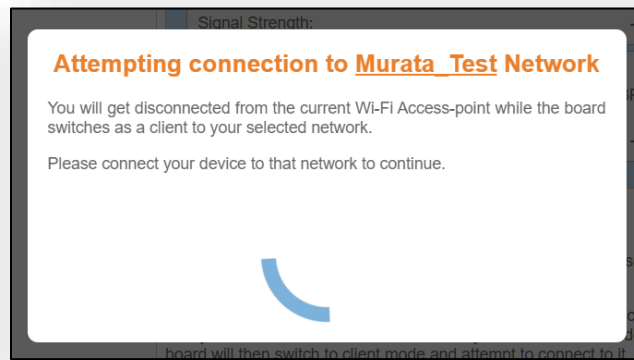
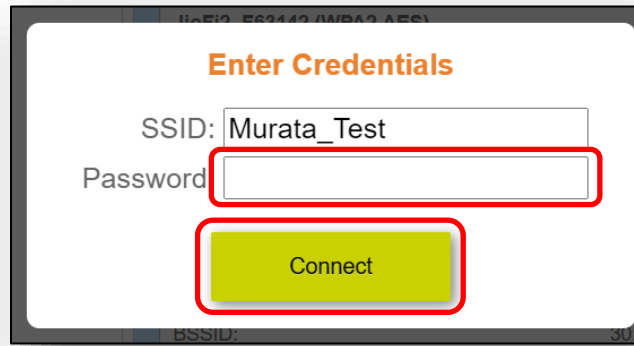
12. Open the web browser on the client device (Microsoft Internet Explorer is not supported) and go to “192.168.1.1” IP address. The wiced_webconfig_4343W example creates a web-based configuration interface to set up the Wi-Fi client configurations here. The EVK is currently set up as an AP, as can be seen in the UI. You can scan for available networks here.



13. Click on a network of your selection to connect to it.



14. Enter the network password and click on connect.



15. If connection is successful, the credential will be saved on the EVK mflash and will be used automatically after the EVK reboots. The AP will be turned off. You should see this output on Tera Term. Note the IP address shown.

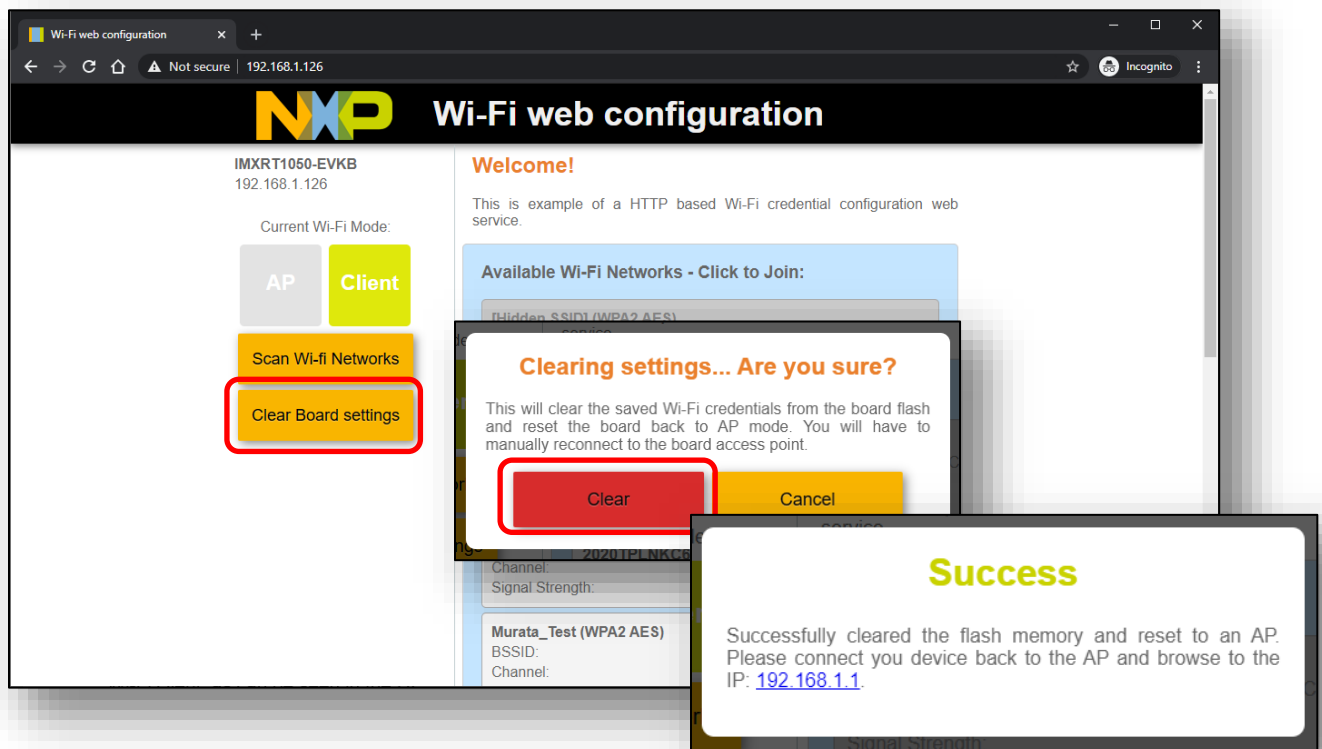
```
COM14 - Tera Term VT
File Edit Setup Control Window Help
Channel : 7
Romit
BSSID : 3C:84:6A:A9:A1:0E
RSSI : -85dBm
Max Data Rate : 216 Mbits/s
Radio Band : 2.4GHz
Channel : 7
Murata_Test
BSSID : 60:38:E0:9A:A3:9D
RSSI : -27dBm
Max Data Rate : 600 Mbits/s
Radio Band : 2.4GHz
Channel : 10
[i] Chosen ssid: Murata_Test
[i] Chosen passphrase: "password"
[i] Joining: Murata_Test
Getting IP address from DHCP server
[i] Successfully joined: Murata_Test
Now join that network on your device and connect to this IP: 192.168.1.126
[i] mflash_save_file success
[i] Stopping AP!
```

Upon reboot, the EVK will henceforth automatically connect to the saved network.

16. Connect the wireless client device to the same network as the EVK, open the web browser and go to the IP address shown on the Tera Term window (192.168.1.126 in this example).
17. The web-based configuration interface will be accessible here. The EVK is currently set up as a Wi-Fi client, as can be seen in the UI.



18. You can use the “Clear Board settings” button on the interface to remove the saved network settings.

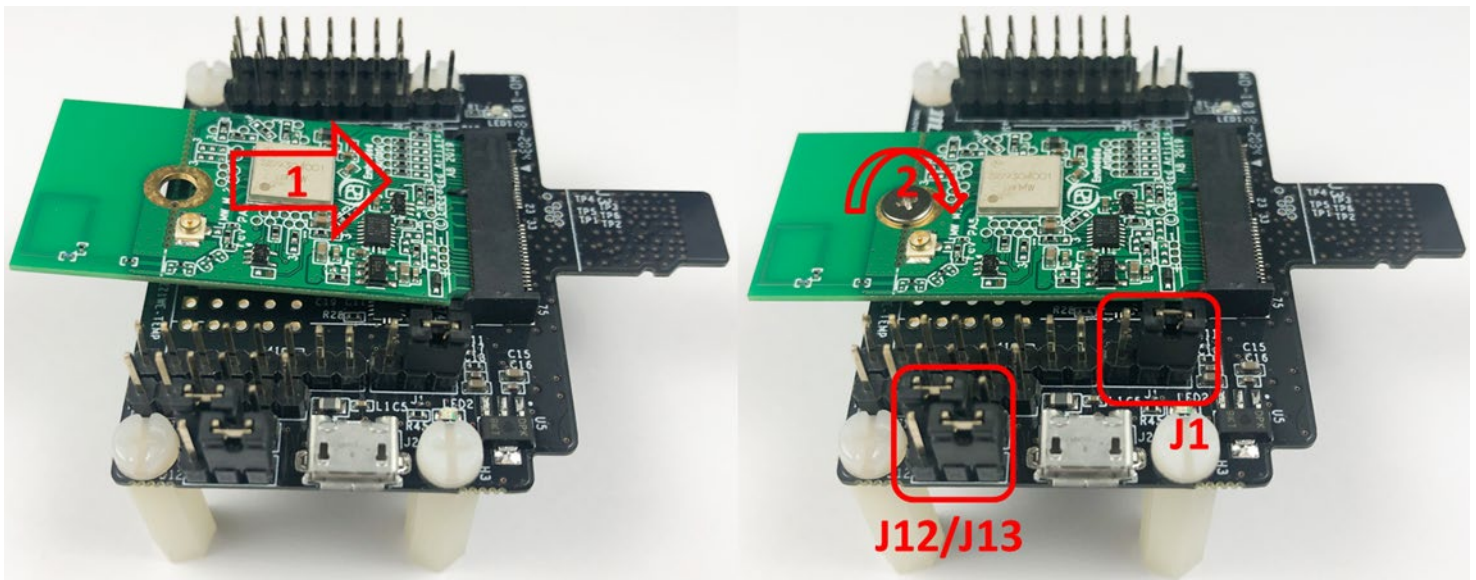


8 Murata's uSD-M.2 Adapter

8.1 Connecting the Wi-Fi/BT M.2 EVB to uSD-M.2 Adapter

When connecting the Wi-Fi/BT M.2 EVB to uSD-M.2 Adapter Rev B1 (**Figure 13**), make sure to (#1) firmly insert it before using M.2 screw to (#2) secure it in place. Important Jumpers (J12, J13, and J1) are highlighted.

Figure 13: Connecting the Wi-Fi/BT M.2 EVB to uSD-M.2 Adapter



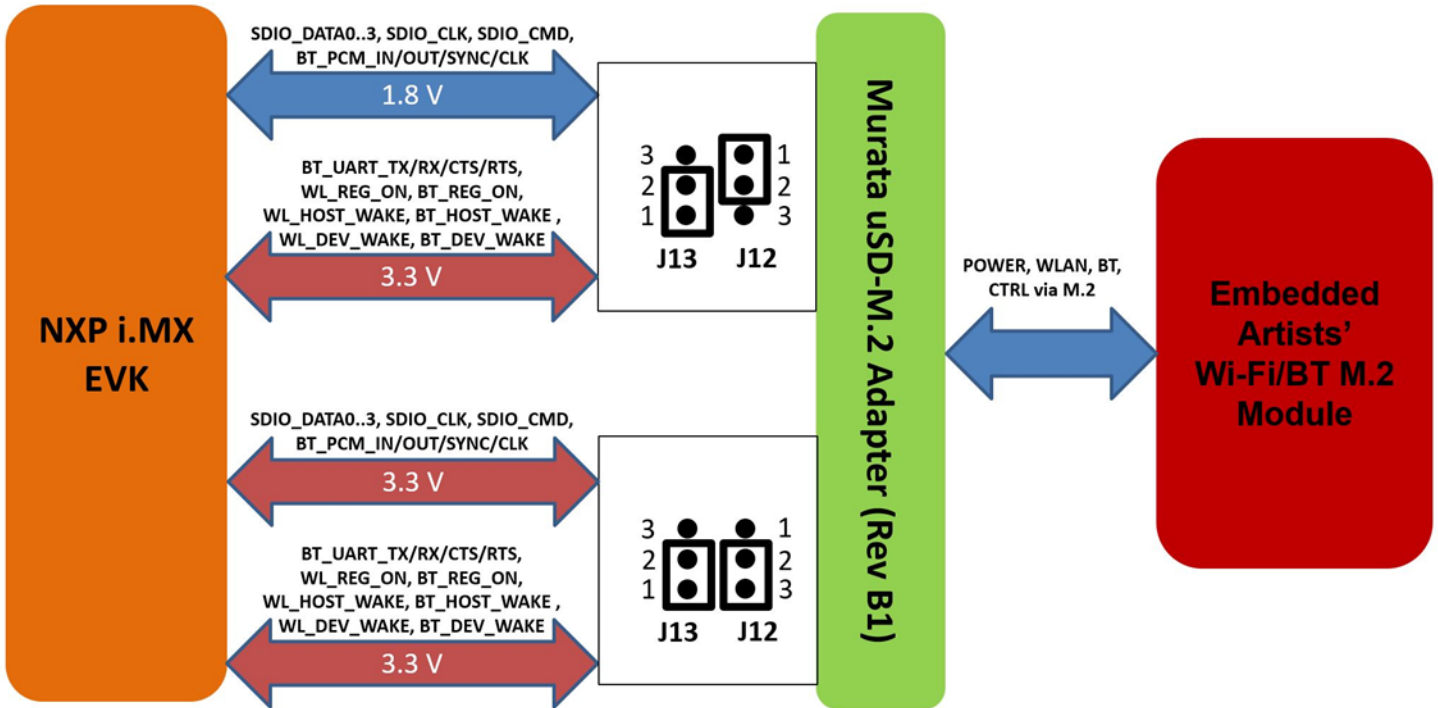
8.2 Configuring uSD-M.2 Adapter Jumpers for Correct VIO Signaling

Figure 14 shows a block diagram highlighting the Host (i.MX RT10xx EVK) and Wi-Fi/BT M.2 EVB VIO signaling voltages.

Default configuration is to have J13/J12 set to 1-2/1-2 positions respectively for the 1.8V VIO default configuration (WLAN-SDIO VIO at 1.8V VIO; BT-UART and WLAN/BT control signals at 3.3V VIO). Rev B1 Adapter level shifts the BT-UART signals and all WLAN/BT control signals except WL_REG_ON/BT_REG_ON which get level shifted on Wi-Fi/BT M.2 EVB.

In certain configurations where Host WLAN-SDIO VIO is 3.3V, we J13/J12 set to 1-2/2-3 positions respectively for the 3.3V VIO override mode setting (WLAN-SDIO, BT-UART, and WLAN/BT control signals all at 3.3V VIO). There is no level shifting on the Adapter of Wi-Fi/BT M.2 EVB in this case.

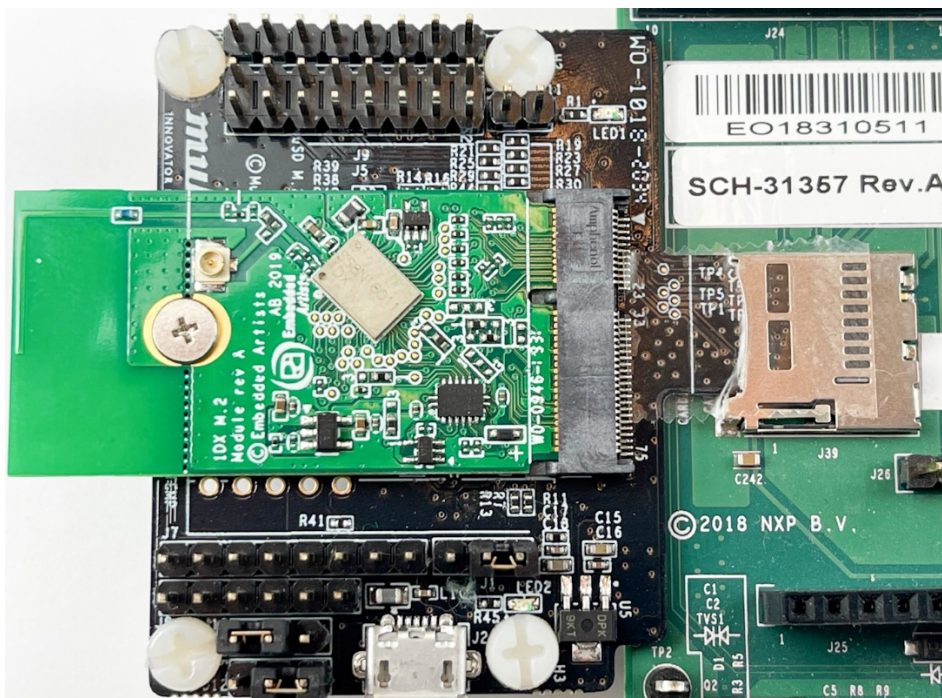
Figure 14: Host/M.2 IO Voltage Level Shift Options on Rev B1 Adapter



8.3 Securing uSD-M.2 Adapter to NXP i.MX RT10xx EVK

On NXP's i.MX RT10xx EVK's, one potential problem is an unreliable uSD/SD electrical connection when using Murata's uSD-M.2 Adapter. This is due to the "push push" uSD connector and lack of friction in the interface between Adapter and microSD connector.

Figure 15: Securing uSD-M.2 Adapter to i.MX RT10xx EVK



To properly secure the uSD-M.2 Adapter interconnect on the i.MX RT10xx EVK's, Murata **strongly recommends** to simply tape the uSD Adapter-EVK connection as shown in **Figure 15**. Note that taping the uSD Adapter-EVK connection makes the platform a little less flexible to work with. However, removing and re-applying clear tape is straightforward.

8.4 uSD-M.2 Adapter High-Level Description

Figure 16 and **Figure 17** show the features on the uSD-M.2 Adapter; with details in **Table 7**. The uSD-M.2 Adapter supports additional signals to WLAN-SDIO using either Arduino headers (J5, J8, and J9) or 20 pin FFC connector (J6). For more details on Murata's uSD-M.2 Adapter, refer to the [Hardware User Manual](#).

Table 7: uSD-M.2 Adapter Features

Char	Description
A	microSD connector provides Power (VBAT, GND) and WLAN-SDIO
B	SDIO bus test points (CLK, CMD, DAT0, DAT1, DAT2, DAT3)
C	Power LED Indicator (green): if not illuminated then no power applied to M.2 EVB
D	J11 = Optional BT Disable Jumper for WLAN-Only Mode (close this jumper to drive BT_REG_ON low and disable Bluetooth Core; thereby optimizing power consumption)
E	J9 = BT UART TX/RX and WLAN/BT Control Signals (8 pin header)
F	J5 = Optional BT PCM and WLAN/BT Debug Signals (2x8 pin header)
G	Threaded mount for M.2 screw: 30mm distance from M.2 connector
H	Regulator to step down optional 5V VBAT from USB or Arduino header to 3.3V
I	External sleep clock input (32.768kHz)
J	J7 = Optional Arduino Header Power Supply (8 pin header; 5V or 3.3V VBAT)
K	J8 = BT UART RTS/CTS Signals (6 pin header)
L	J13 = Host IO Voltage: J13 in 1-2 pos for 3.3V VDDIO (default); J13 in 2-3 pos for 1.8V
M	J12 = M.2 IO Voltage: J12 in 1-2 pos for 1.8V VDDIO (default); J12 in 2-3 pos for 3.3V
N	J2 = Optional 5V USB Power Supply via Micro-AB USB Connector
O	LED2 = 3.3V M.2 IO Voltage Indicator (Blue) – not illuminated in default configuration
P	Regulator to provide optional 1.8V VIO to M.2 interface (M.2 EVB's have own 1.8V onboard)
Q	J1 = Power Supply Selector Jumper must be installed to power Adapter (unless J5 Arduino Header Pins #15/16 are connected to external GND/3.3V VBAT). Position 1-2: 5V/3.3V VBAT supply from micro-USB (J2); or Arduino (J7) Position 2-3: VBAT supply (typical 3.1~3.3V) from microSD connector
R	M.2 Connector: type 2230-xx-E
S	microSD connector pins: provides Power (VBAT, GND) and WLAN-SDIO
T	WLAN JTAG header (header pins not populated)
U	20 pin FFC connector (BT UART, BT PCM, WLAN/BT Control signals)
V	Additional test points from 20pin flat/flex connector

Figure 16: uSD-M.2 Adapter Features (Top View)

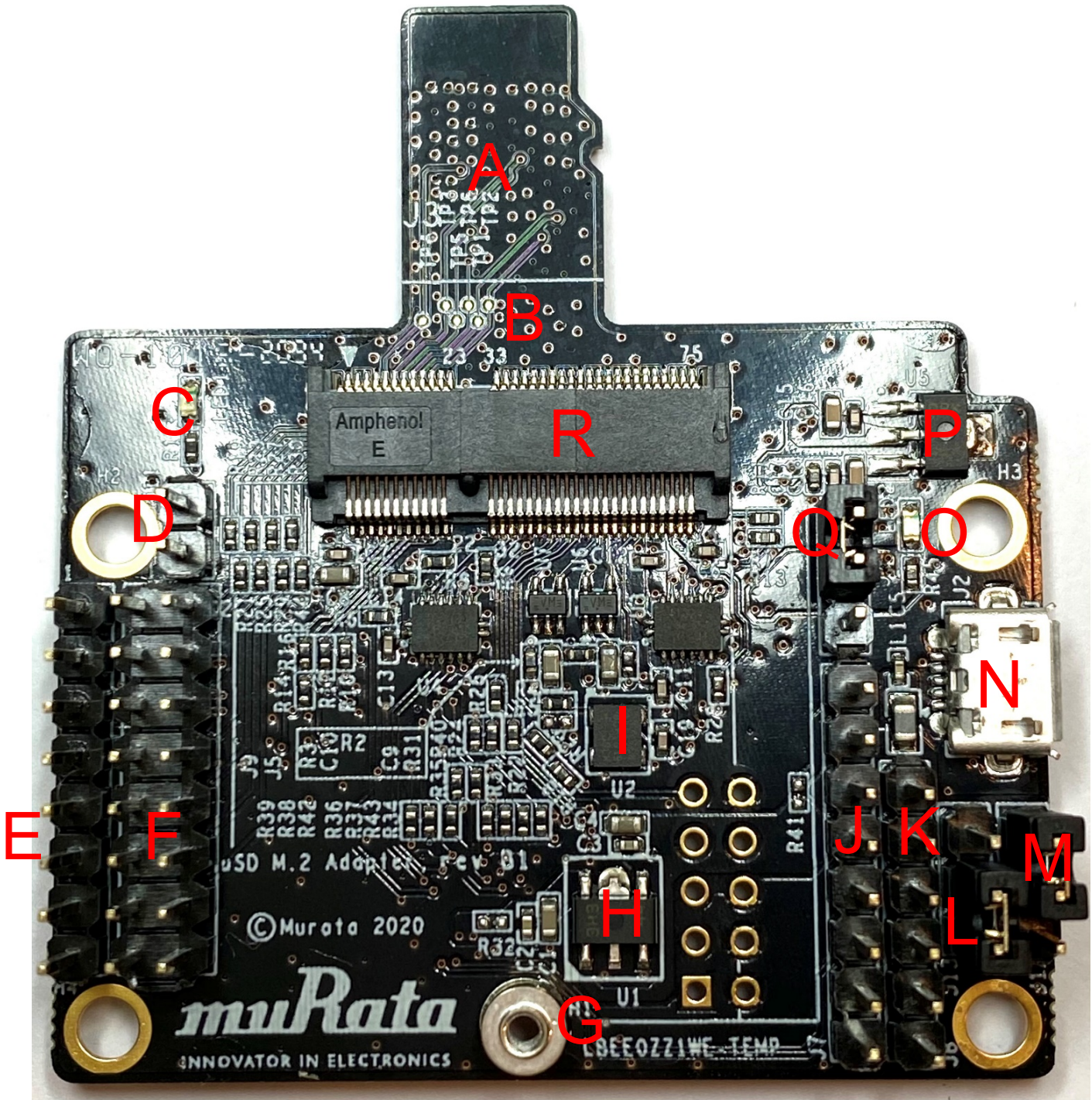
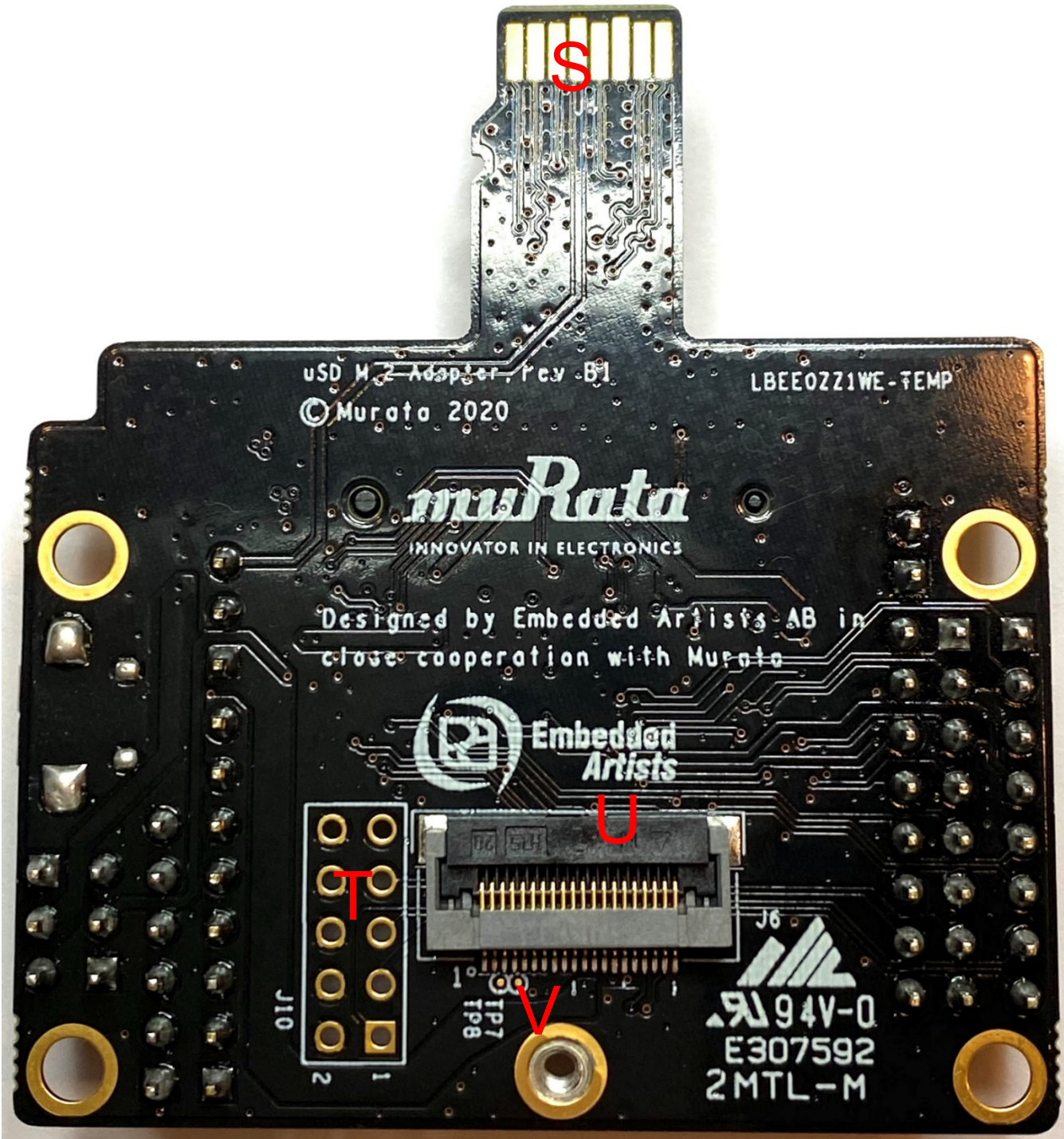


Figure 17: uSD-M.2 Adapter Features (Bottom View)



9 Technical Support Contact

Table 8 lists all the support resources available for the Murata Wi-Fi/Bluetooth solution.

Table 8: List of Support Resources

Support Site	Notes
Murata Community Forum	Primary support point for technical queries. This is an open forum for all customers. Registration is required.
Murata i.MX Landing Page	No login credentials required. Murata documentation covering hardware, software, testing, etc. is provided here.
Murata uSD-M.2 Adapter Landing Page	Landing page for uSD-M.2 Adapter. In conjunction with Murata i.MX Landing Page, this should provide the user with comprehensive getting started documentation.
Murata Module Landing Page	No login credentials required. Murata documentation covering all Cypress-based Wi-Fi/BT modules is provided here.

10 Additional Useful Links

In addition to Table 8 listings of support resources, Table 9, Table 10, Table 11 and Table 12 provides some useful links.

Table 9: NXP links

Link	Notes
MCUXpresso IDE	Landing page to download MCUXpresso IDE
MCUXpresso SDK	Comprehensive information of MCUXpresso SDK
MCUXpresso SDK Builder	Customize and build MCUXpresso SDKs
LPC-Link 2	Landing page of the debug probe for i.MX RT EVKs
i.MX RT 1050 EVK	Landing page of the i.MX RT 1050 EVK
i.MX RT 1060 EVK	Landing page of the i.MX RT 1060 EVK
i.MX RT 1050 Getting Started	Getting started guide for the i.MX RT 1050 EVK
i.MX RT 1056 Getting Started	Getting started guide for the i.MX RT 1056 EVK

Table 10: Embedded Artists' Landing Pages

Landing Pages	Notes
<u>Embedded Artists' Website</u>	The Art of Embedded Systems Development – made EASY™
<u>i.MX RT COM Boards</u>	Listing of Computer-on-Module boards.
<u>i.MX RT COM Carrier Board V2</u>	Main baseboard which all the COM boards plug into.
<u>M.2 Module Family</u>	Top level listing of 1DX, 1LV, 1MW M.2 EVBs.

Table 11: Embedded Artists' Datasheets and Schematics

Datasheets and Schematics	Notes
<u>i.MX RT COM Carrier Board V2 Datasheet</u>	Comprehensive definition of COM Carrier (baseboard).
<u>i.MX RT COM Carrier Board V2 Schematics</u>	Complete schematics including clear definition of uSD-M.2 Adapter.
<u>M.2 SDIO Interface Schematic</u>	Reference schematic for customers designing in WLAN-SDIO M.2 EVB.
<u>M.2 PCIe Interface Schematic</u>	Reference schematic for customers designing in WLAN-PCIe M.2 EVB.
<u>EACOM Board Specification Guide</u>	Comprehensive definition of Embedded Artists' Computer-On-Module's.
<u>1DX M.2 Module Datasheet</u>	Comprehensive details on 1DX Wi-Fi/BT M.2 Module.
<u>1LV M.2 Module Datasheet</u>	Comprehensive details on 1LV Wi-Fi/BT M.2 Module.
<u>1MW M.2 Module Datasheet</u>	Comprehensive details on 1MW Wi-Fi/BT M.2 Module.

Table 12: Embedded Artists' User Manuals and Software

User Manuals and Software	Notes
<u>Getting Started with M.2 modules and i.MX RT</u>	How to bring up Embedded Artists i.MX RT Dev Kits.
<u>Wi-Fi/BT M.2 EVB Primer</u>	Introduction and drill-down on M.2 interface.