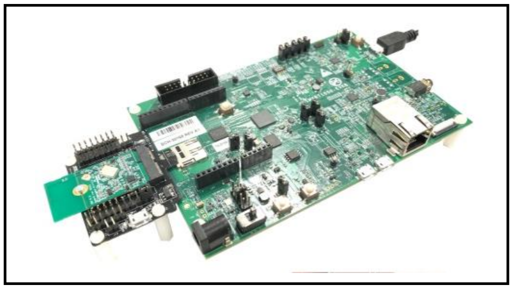
E2B-93-1097

Murata Wi-Fi/BT Solution for i.MX

Quick Start Guide (FreeRTOS)



Revision History

Revision	Date	Author	Change Description
1.0	May 29, 2019	T. Inoue P. Sah	Initial Release

Table of Contents

LIST OF FIGURES

1 Introduction

This document provides detailed instructions on adding wireless functionality with Embedded Artists' Wireless M.2 Modules on NXP's i.MX RT 1050/1060 Evaluation Kit and Embedded Artists' i.MX RT 1052/1062 Developer's kit. NXP's latest SDK supports 1DX M.2 module and Wi-Fi support only. Other M.2 modules (i.e. 1MW and 1LV) and the Bluetooth support are scheduled to be included in the upcoming release. This chapter explains how to set up the hardware and the steps required for software installation.

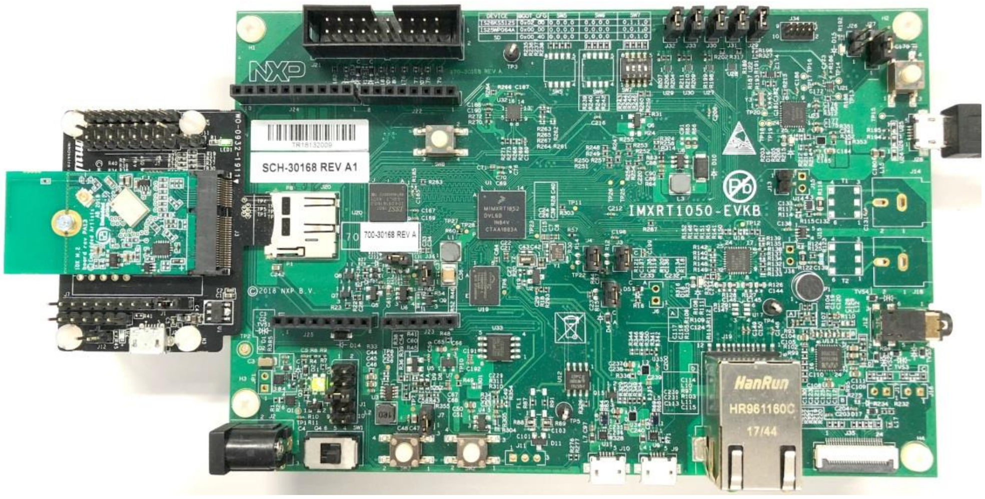
1.1 Hardware Options

This section describes how to configure the hardware correctly. There are two hardware suppliers for i.MX RT series, NXP and Embedded Artists (EA).

1.1.1 NXP with uSD-M.2 Adapter

NXP's i.MX RT1050/1060 Evaluation Kit supports the M.2 Wireless Module family with uSD-M.2 adapter board. On-board debug adapter is supported. Refer to **Figure 1** for the right connection of the uSD-M.2 adapter board and the NXP's i.MX RT board.

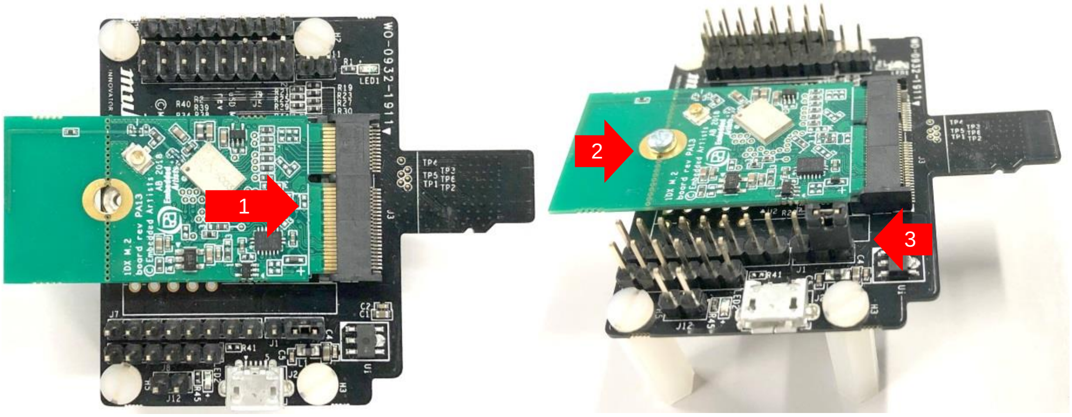
Figure 1: NXP IMXRT1050-EVKB with uSD-M.2 Adapter



Micro USB connector (J28) is used for USB-UART/JTAG. To use uSD-M.2 Adapter, attach the stand-offs (H2, H3, H4, H5), insert M.2 board into M.2 connector (J3), then tighten the screw (H1). Make sure J1 has a jumper on position 2-3 as shown in **Figure 2 (details in Figure 10)**. Refer to **Figure 2** for the correct orientation of the uSD-M.2 adapter board. And then insert it into uSD slot (J20) until you hear the click sound. If you want to detach the adapter, push it until you hear the click sound, then you can pull out from uSD slot. Make sure the green light i.e. LED1 on the adapter board is ON.

NOTE: Make sure J12 is open and there is no blue light on the adapter board. This indicates the board is operating on 1.8V.

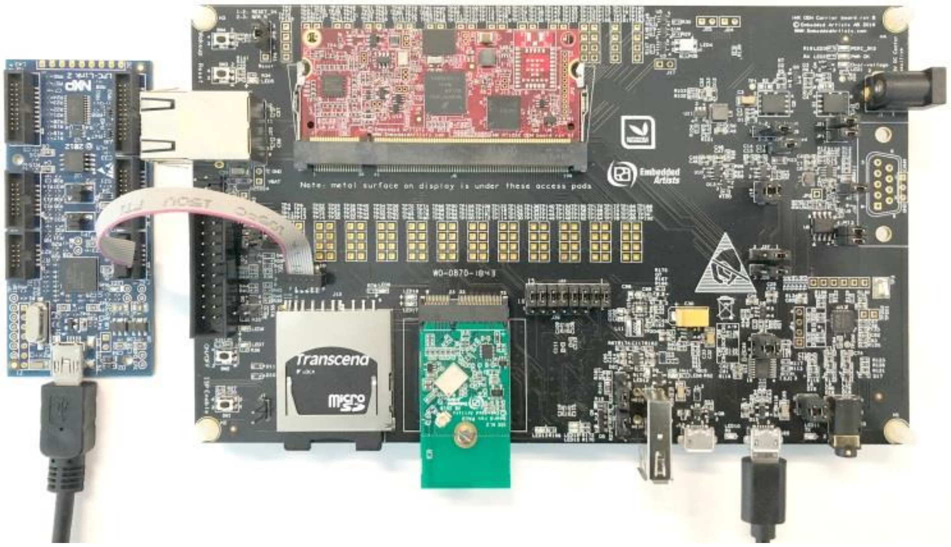
Figure 2: Setup for M.2 Board with uSD-M.2 Adapter



1.1.2 Embedded Artists

Embedded Artists' i.MX RT 1052/1062 Developer's Kit have M.2 connector built on the board for the M.2 Wireless Module family. LPC-Link2 is recommended for the debug adapter. Embedded Artists web site provides support package. **Figure 3** shows the full connection of developer's kit with M.2 and debug probe.

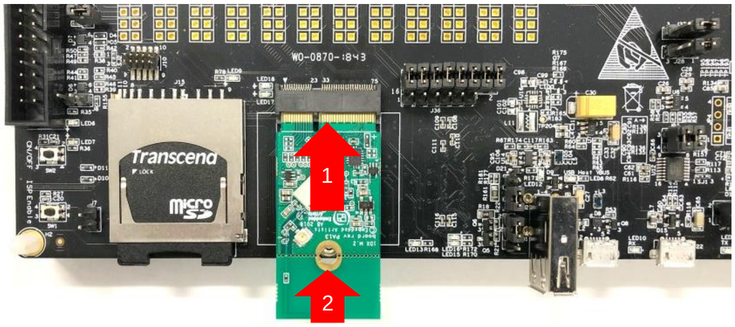
Figure 3: Embedded Artists iMX RT 1062 Developer's Kit



Micro USB connector (J22) is used for USB-UART. J10 is for LPC-Link2 connection. To use uSD-M.2 Adapter, insert M.2 board into M.2 connector (J38), then tighten the screw (H1). **Figure 4** shows how to mount the M.2 module into the connector.

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: How to Attach M.2 Board with EA Kit



1.2 Software Options

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

1.3 Acronyms

Table: Acronyms used in Quick Start Guide

Acronym	Meaning
BT	Bluetooth
EA	Embedded Artists (The company developed M.2 module with Murata)
EULA	End User License Agreement
EVB	Evaluation Board (Embedded Artists' Wi-Fi/BT module)
EVK	Evaluation Kit (includes EVB + Adapter)
FTDI	Future Technology Devices International
IDE	Integrated Development Environment
PC	Personal Computer
RF	Radio Frequency
RX	Receive
SD	Secure Digital
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

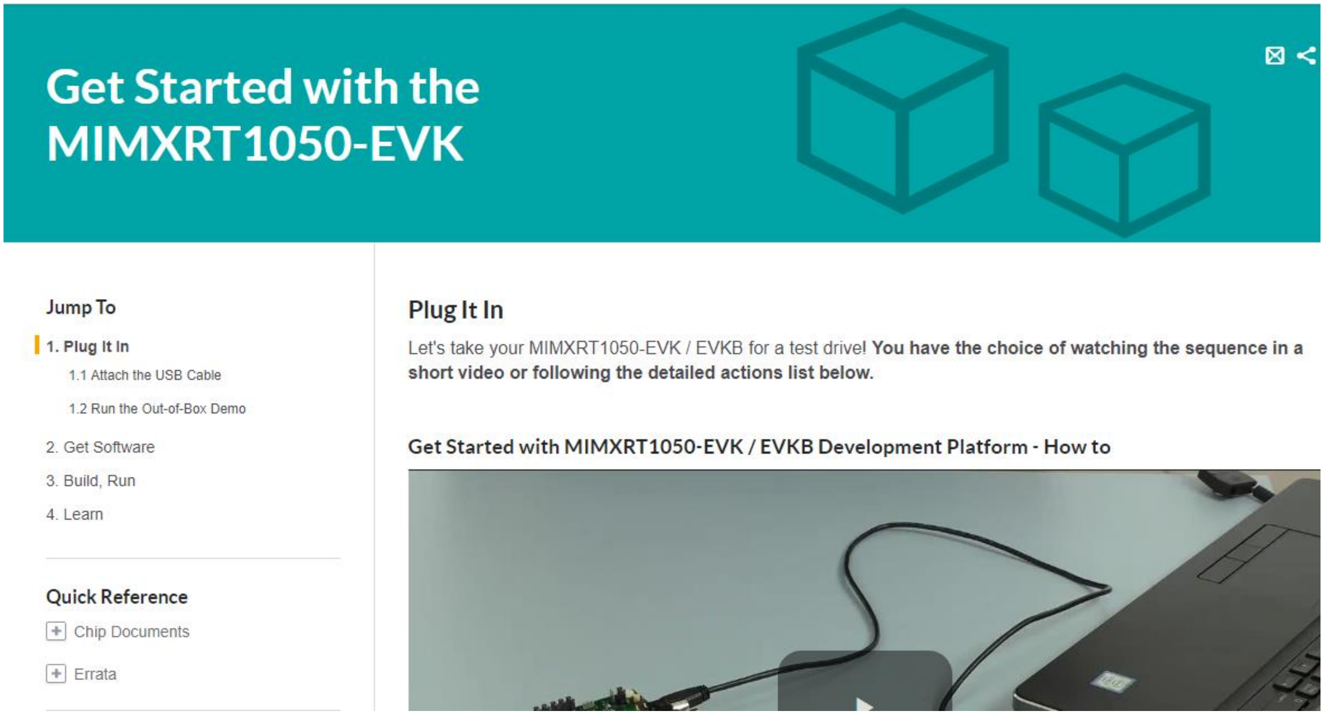
2 Setup for NXP Board

If you are using Embedded Artists Board, please skip this section and go to **Section 3** 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



Get Started with the MIMXRT1050-EVK

Jump To

1. Plug It In

1.1 Attach the USB Cable

1.2 Run the Out-of-Box Demo

2. Get Software

3. Build, Run

4. Learn

Quick Reference

Chip Documents

Errata

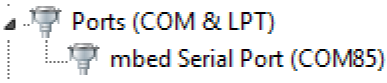
Plug It In

Let's take your MIMXRT1050-EVK / EVKB for a test drive! **You have the choice of watching the sequence in a short video or following the detailed actions list below.**

Get Started with MIMXRT1050-EVK / EVKB Development Platform - How to

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



Ports (COM & LPT)

mbed Serial Port (COM85)

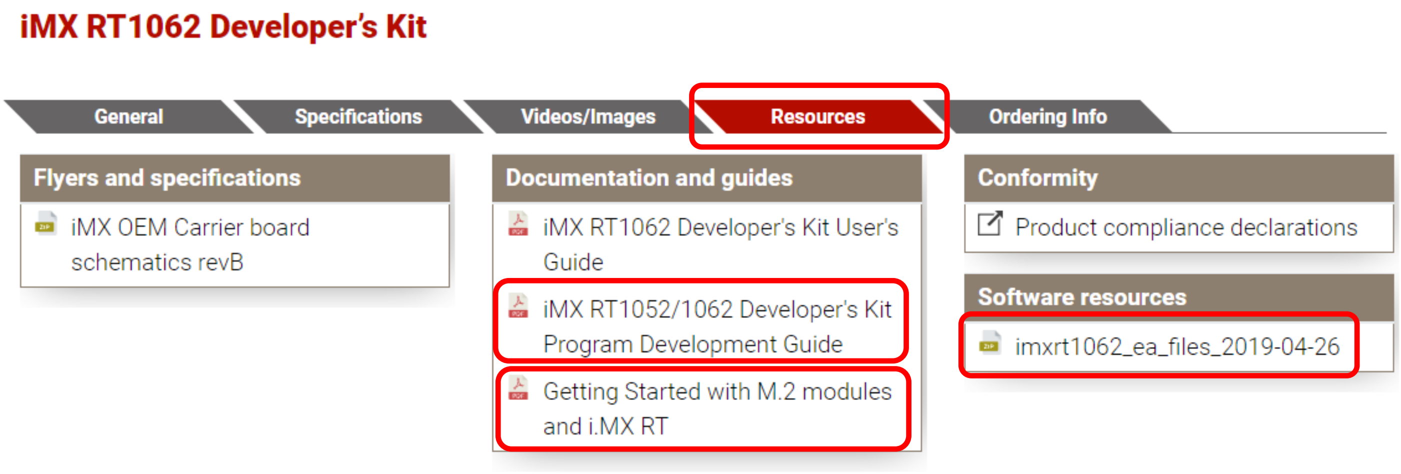
3 Setup for Embedded Artists Board

If you are using NXP Board, please skip this section and go back to **Section 2** 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_yyyy-mm-dd.zip” for the SDK setup. Go to **Section 4** 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



iMX RT1062 Developer’s Kit

Flyers and specifications

- iMX OEM Carrier board schematics revB

Documentation and guides

- iMX RT1062 Developer's Kit User's Guide
- iMX RT1052/1062 Developer's Kit Program Development Guide
- Getting Started with M.2 modules and i.MX RT

Conformity

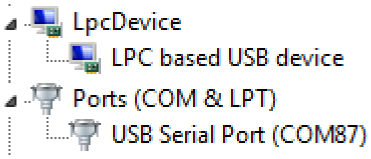
- Product compliance declarations

Software resources

- imxrt1062_ea_files_2019-04-26

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

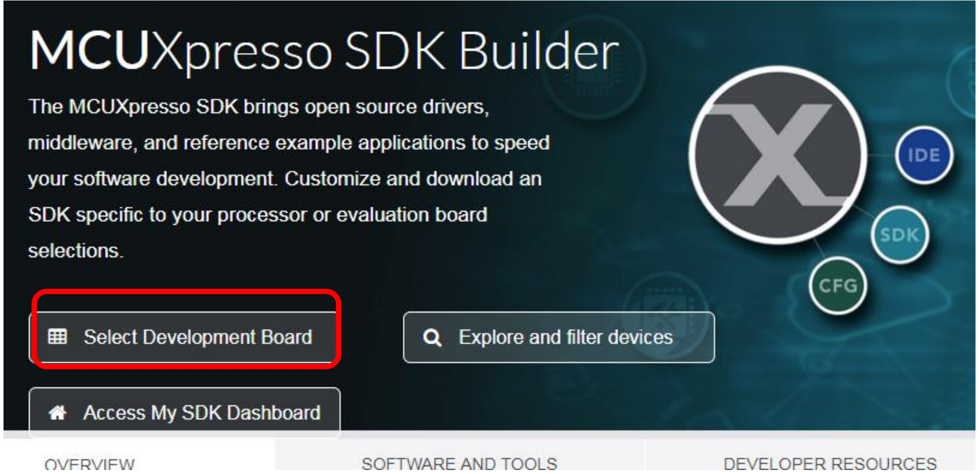


LpcDevice
LPC based USB device
Ports (COM & LPT)
USB Serial Port (COM87)

4 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. Follow the steps below carefully to have right components installed.

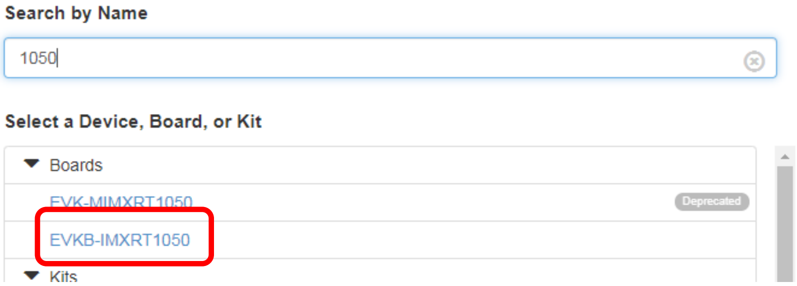
1. First download the MCUXpresso SDK by following this URL: https://mcuxpresso.nxp.com. You will need to login to your NXP account and then click "Select Development Board".



MCUXpresso SDK Builder

The MCUXpresso SDK brings open source drivers, middleware, and reference example applications to speed your software development. Customize and download an SDK specific to your processor or evaluation board selections.

2. Type "1050", then select "EVKB-IMXRT1050".



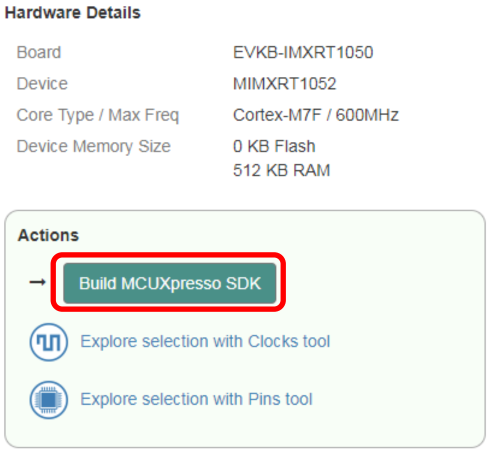
Search by Name

1050

Select a Device, Board, or Kit

- Boards
  - EVK-MIMXRT1050 Deprecated
  - EVKB-IMXRT1050
- Kits

3. Scroll down, then click "Build MCUXpresso SDK"

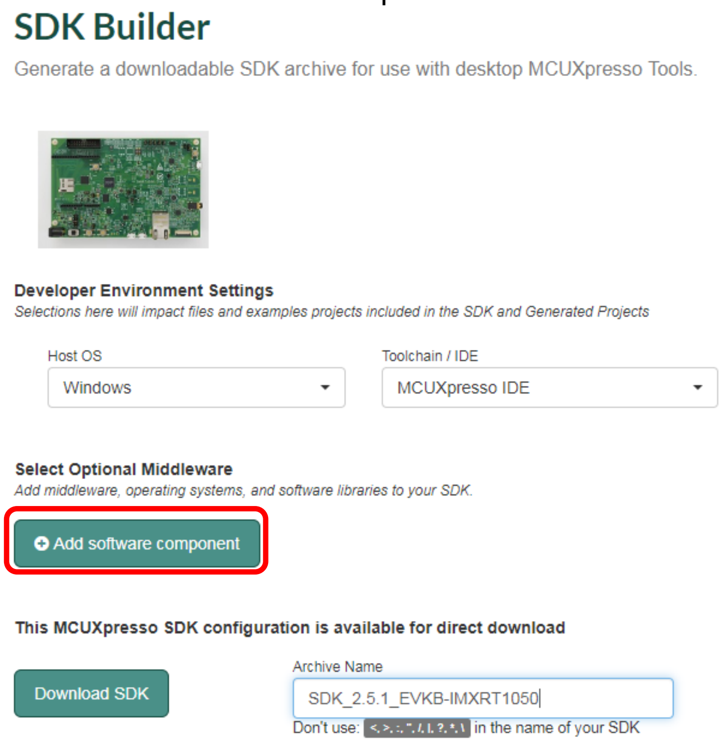


Hardware Details

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 Clocks tool
Explore selection with Pins tool

4. Click “Add software component”.



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

Host OS: Windows

Toolchain / IDE: MCUXpresso IDE

Select Optional Middleware

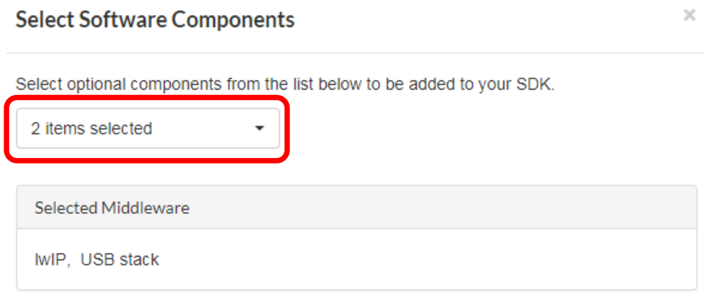
Add middleware, operating systems, and software libraries to your SDK.

This MCUXpresso SDK configuration is available for direct download

Archive Name: SDK_2.5.1_EVKB-IMXRT1050

Don't use: <, >, :, ", /, |, ?, *, \ in the name of your SDK

5. Click “2 items selected”.



Select Software Components

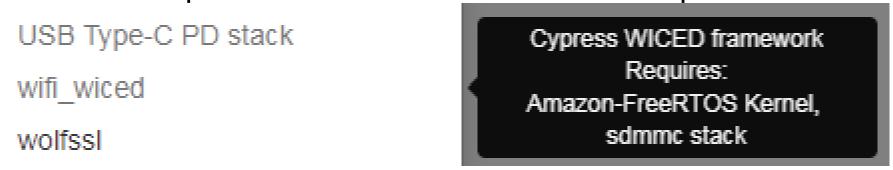
Select optional components from the list below to be added to your SDK.

2 items selected

Selected Middleware
lwIP, USB stack

6. Select components at least Amazon-FreeRTOS Kernel, sdmmc stack, wifi_wiced, mbedtls, AWS IoT, or you can click “Select All”. After selecting those components, then click “Save changes”.

AWS IoT and wifi_wiced have required components, so need to select those required components first. You can put the mouse cursor over the component to know required components.

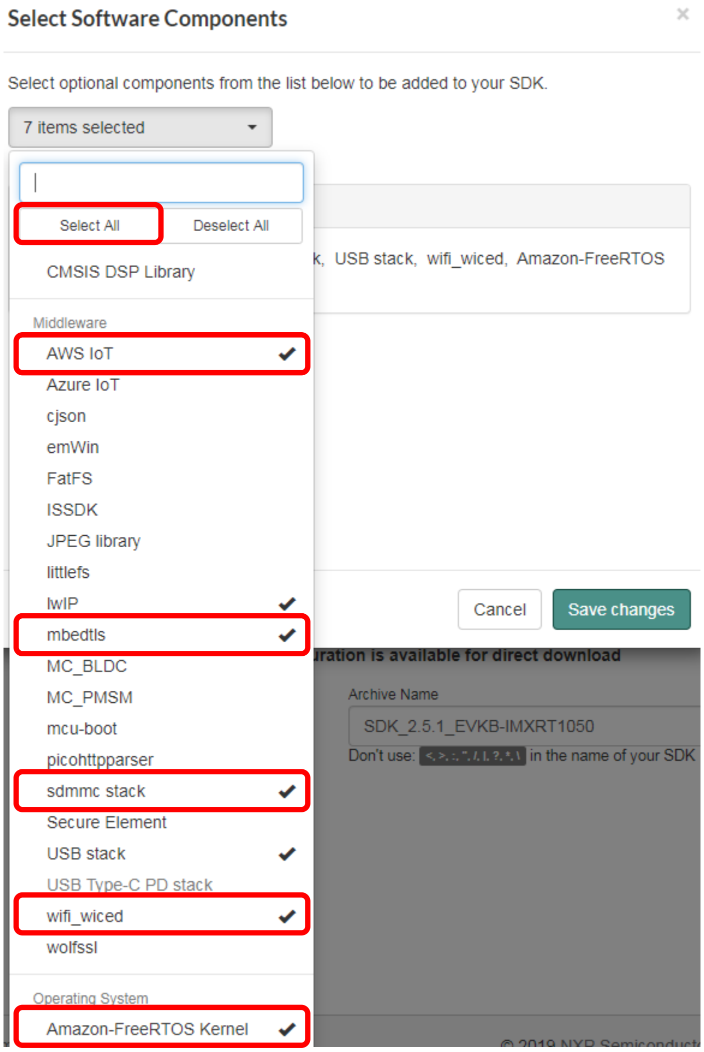


USB Type-C PD stack

wifi_wiced

wolfssl

Cypress WICED framework
Requires:
Amazon-FreeRTOS Kernel,
sdmmc stack



Select Software Components ×

Select optional components from the list below to be added to your SDK.

7 items selected ▾

Select All | Deselect All

- CMSIS DSP Library

Middleware

- AWS IoT ✓
- Azure IoT
- cjson
- emWin
- FatFS
- ISSDK
- JPEG library
- littlefs
- lwIP ✓
- mbedtls ✓
- MC_BLDC
- MC_PMSM
- mcu-boot
- picohttpparser
- sdmmc stack ✓
- Secure Element
- USB stack ✓
- USB Type-C PD stack
- wifi_wiced ✓
- wolfssl

Operating System

- Amazon-FreeRTOS Kernel ✓

k, USB stack, wifi_wiced, Amazon-FreeRTOS

Cancel | Save changes

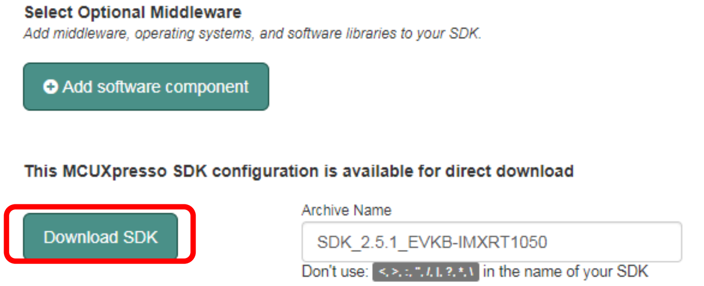
ration is available for direct download

Archive Name: SDK_2.5.1_EVKB-IMXRT1050

Don't use: <, >, :, ", /, |, ?, *, \ in the name of your SDK

© 2019 NXP Semiconducto

7. Click “Download SDK”, then agree the EULA to download the SDK.



Select Optional Middleware

Add middleware, operating systems, and software libraries to your SDK.

⊕ Add software component

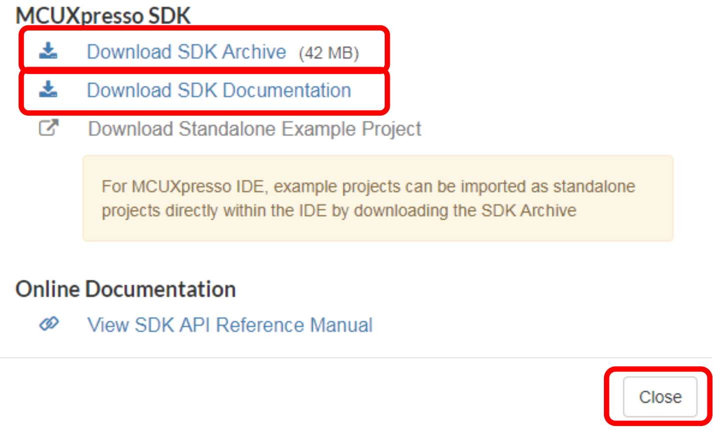
This MCUXpresso SDK configuration is available for direct download

Download SDK

Archive Name: SDK_2.5.1_EVKB-IMXRT1050

Don't use: <, >, :, ", /, |, ?, *, \ in the name of your SDK

8. Click “Download SDK Archive” if download doesn’t start automatically. You can also download SDK Documentation, then click “Close”.



MCUXpresso SDK

Download SDK Archive (42 MB)

Download SDK Documentation

Download Standalone Example Project

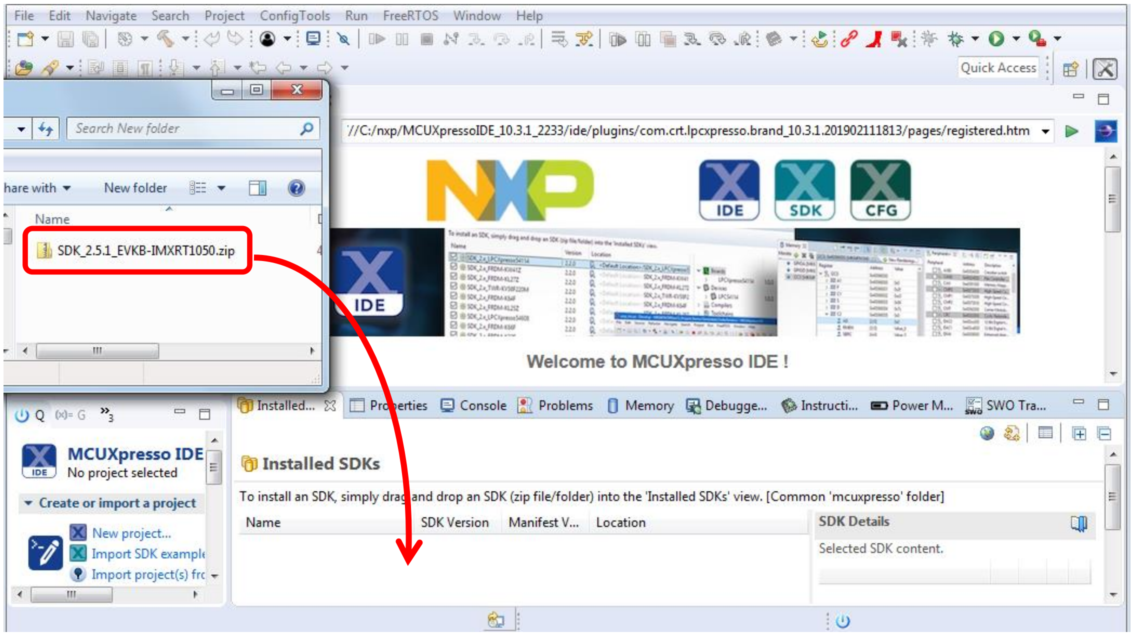
For MCUXpresso IDE, example projects can be imported as standalone projects directly within the IDE by downloading the SDK Archive

Online Documentation

View SDK API Reference Manual

Close

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


File Edit Navigate Search Project ConfigTools Run FreeRTOS Window Help
Quick Access
Search New folder
hare with
New folder
Name
SDK_2.5.1_EVKB-IMXRT1050.zip
//C:/nxp/MCUXpressoIDE_10.3.1_2233/ide/plugins/com.crt.lpcxpresso.brand_10.3.1.201902111813/pages/registered.htm
IDE
SDK
CFG
Welcome to MCUXpresso IDE !
MCUXpresso IDE
No project selected
Create or import a project
New project...
Import SDK example
Import project(s) fro
Installed...
Properties
Console
Problems
Memory
Debugge...
Instructi...
Power M...
SWO Tra...
Installed SDKs

To install an SDK, simply drag and drop an SDK (zip file/folder) into the 'Installed SDKs' view. [Common 'mcuxpresso' folder]

Name	SDK Version	Manifest V...	Location

SDK Details

Selected SDK content.

NOTE: If you are using EA board, before installing the SDK, unzip “imxrt10xx_ea_files_yyyy-mm-dd.zip” file downloaded in **Section 3** in some directory. Copy imxrt10xx_ea_files_yyyy-mm-dd\evkmimxrt10xx_flexspi_nor_config.c
to
SDK_2.5.1_EVK-MIMXRT10xx EA\boards\evkmimxrt10xx\xip\
replacing the current file. After the file is copied successfully you can now install the SDK in the MCUXpresso. You can drag and drop the unzipped and modified SDK file into the IDE to install the SDK.

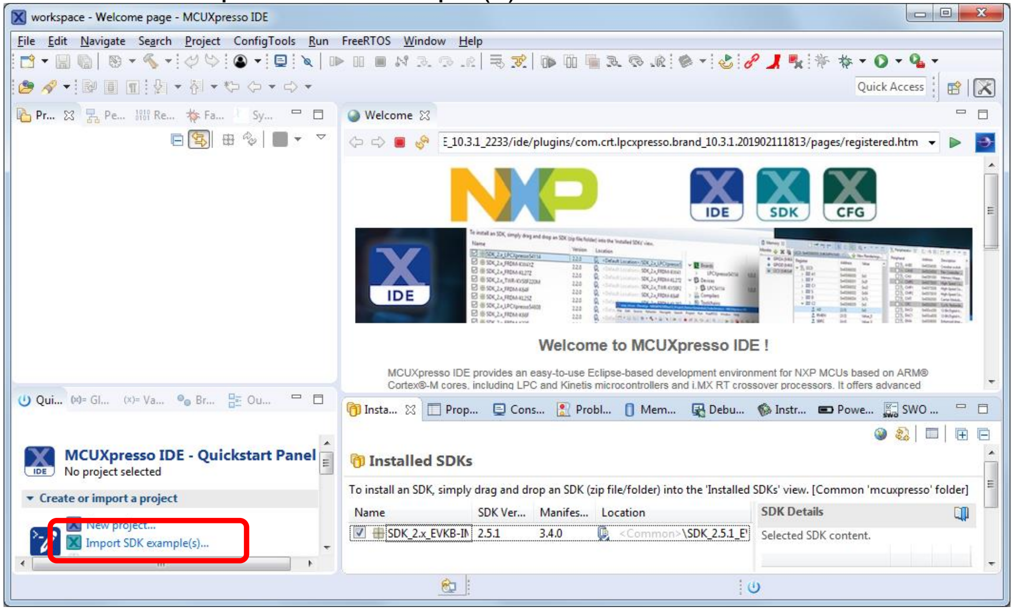
5 Running Sample Application

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

- wiced_iperf: this is for performance test.
- wiced_mfg_test: this is for RF measurements and Regulatory testings.

5.1 Example wiced_iperf

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


workspace - Welcome page - MCUXpresso IDE
File Edit Navigate Search Project ConfigTools Run FreeRTOS Window Help
Quick Access
Welcome
E_10.3.1_2233/ide/plugins/com.crt.lpcxpresso.brand_10.3.1.201902111813/pages/registered.htm
IDE
SDK
CFG
Qui... Gl... Va... Br... Ou...
MCUXpresso IDE - Quickstart Panel
No project selected
Create or import a project
New project...
Import SDK example(s)...
Insta... Prop... Cons... Probl... Mem... Debu... Instr... Powe... SWO ...

Welcome to MCUXpresso IDE !

MCUXpresso IDE provides an easy-to-use Eclipse-based development environment for NXP MCUs based on ARM® Cortex®-M cores, including LPC and Kinetis microcontrollers and i.MX RT crossover processors. It offers advanced

Installed SDKs

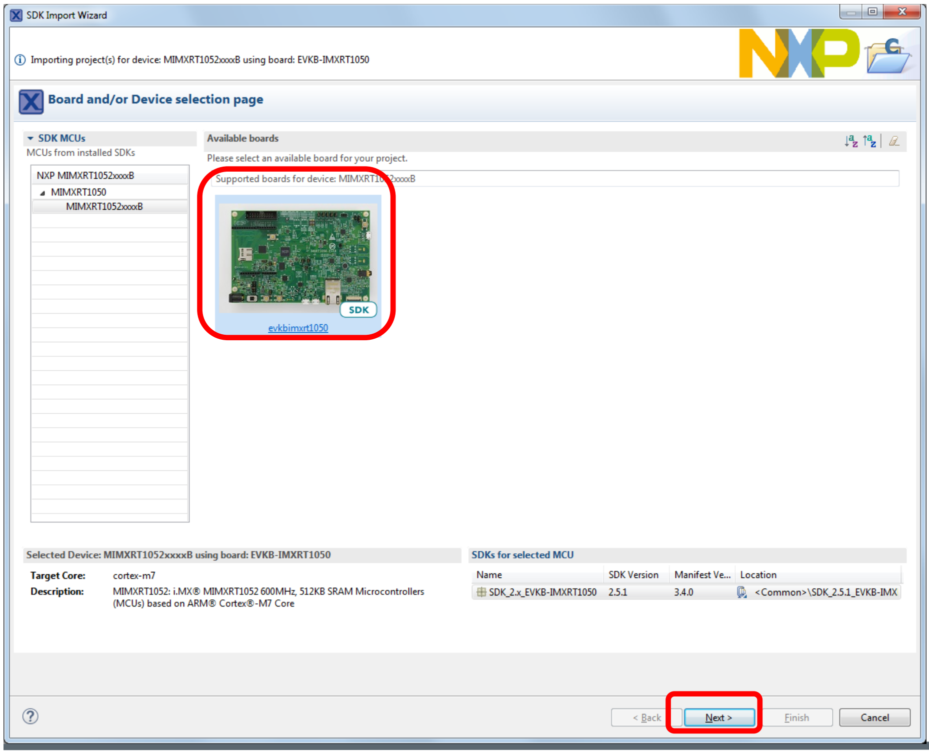
To install an SDK, simply drag and drop an SDK (zip file/folder) into the 'Installed SDKs' view. [Common 'mcuxpresso' folder]

Name	SDK Ver...	Manifes...	Location
☑ SDK_2.x_EVKB-IN	2.5.1	3.4.0	<Common>\SDK_2.5.1_E'

SDK Details

Selected SDK content.

2. Select “evkimxrt1050” board and click Next button.



Importing project(s) for device: MIMXRT1052xxxxB using board: EVKB-IMXRT1050

Please select an available board for your project.

Supported boards for device: MIMXRT1052xxxxB

Selected Device: MIMXRT1052xxxxB using board: EVKB-IMXRT1050

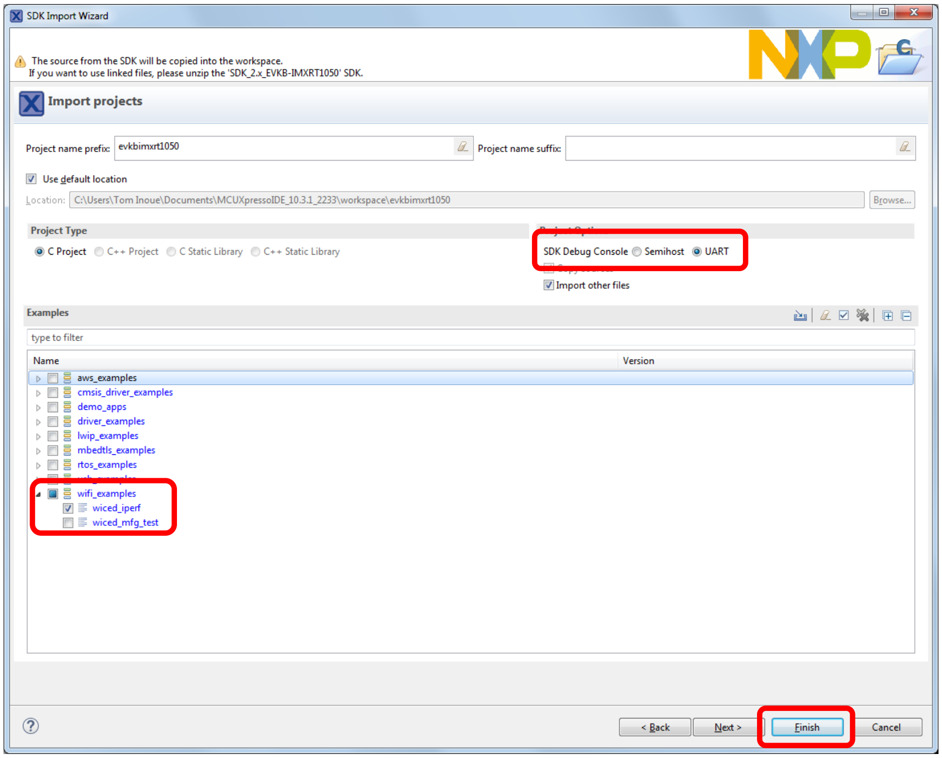
Target Core: cortex-m7

Description: MIMXRT1052: i.MX® MIMXRT1052 600MHz, 512KB SRAM Microcontrollers (MCUs) based on ARM® Cortex®-M7 Core

SDKs for selected MCU

Name	SDK Version	Manifest Ve...	Location
SDK_2.x_EVKB-IMXRT1050	2.5.1	3.4.0	<Common>\SDK_2.5.1_EVKB-IMX

3. Select wifi_examples - wiced_iperf. Select UART for SDK Debug Console, then click Finish button.



SDK Import Wizard

The source from the SDK will be copied into the workspace.
If you want to use linked files, please unzip the 'SDK_2.x_EVKB-IMXRT1050' SDK.

Import projects

Project name prefix: evkbimxrt1050 Project name suffix:

☑ Use default location

Location: C:\Users\Tom Inoue\Documents\MCUXpressoIDE_10.3.1_2233\workspace\evkbimxrt1050 Browse...

Project Type: ● C Project ○ C++ Project ○ C Static Library ○ C++ Static Library

SDK Debug Console ○ Semihost ● UART

☑ Import other files

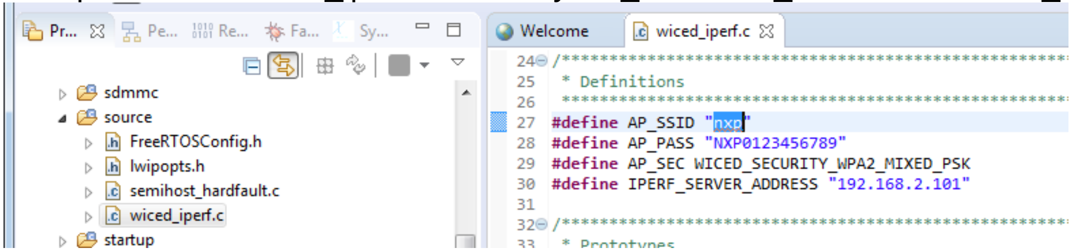
Examples

type to filter

Name	Version
▷ □ aws_examples	
▷ □ cmsis_driver_examples	
▷ □ demo_apps	
▷ □ driver_examples	
▷ □ lwip_examples	
▷ □ mbedtls_examples	
▷ □ rtos_examples	
▷ □ usb_examples	
◢ ■ wifi_examples	
☑ wiced_iperf	
□ wiced_mfg_test	

If you are using Embedded Artists board, modify flash memory size from 0x800000 to 0x400000 and select MIMXRT1050-EcoXiP_ATXP032.cfx as the driver. Same driver and flash memory size is used for both i.MX RT 1052 and i.MX RT 1062 Developer's kit. Please refer to "iMX RT1052/1062 Developer's Kit Program Development Guide" downloaded in **Section 3** for the detail. To run the wiced_iperf example in the EA's iMX RT 1052/1062 Developer's Kit, please refer to the "Getting Started with M.2 modules and i.MX RT" downloaded in **Section 3.**

4. Open source/wiced_iperf.c to modify AP_SSID, AP_PASS and IPERF_SERVER_ADDRESS.

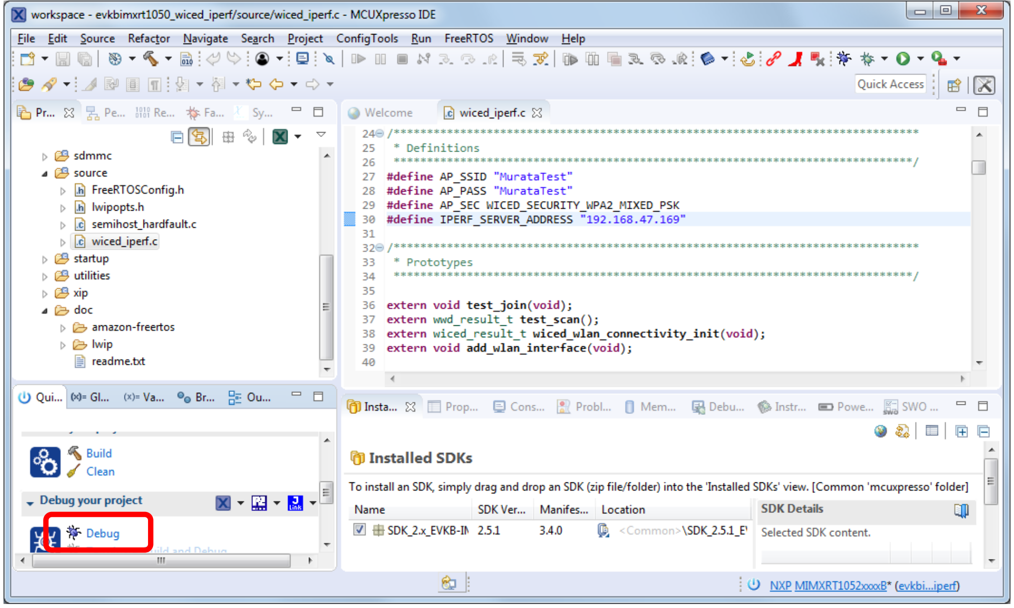


```
/*******************************************************
 * Definitions
 *******************************************************
#define AP_SSID "nxp"
#define AP_PASS "NXP0123456789"
#define AP_SEC WICED_SECURITY_WPA2_MIXED_PSK
#define IPERF_SERVER_ADDRESS "192.168.2.101"

/*******************************************************
 * Prototypes
```

In case of EA board, you need to add some files before starting debug.

File to copy	Destination
pin_mux.c	board/
fsl_lpi2c.c	drivers/
fsl_lpi2c.h	drivers/
pca6416.c	source/
pca6416.h	source/
wwd_platform.c	43xxx_wifi/WICED/platform/MCU/LPC/WWD
wwd_SDIO.c	43xxx_wifi/WICED/platform/MCU/LPC/WWD

5. Click Debug in the QuickStart Panel.



workspace - evkbimxrt1050_wiced_iperf/source/wiced_iperf.c - MCUXpresso IDE

```
/*******************************************************************************
 * Definitions
 ******************************************************************************/
#define AP_SSID "MurataTest"
#define AP_PASS "MurataTest"
#define AP_SEC WICED_SECURITY_WPA2_MIXED_PSK
#define IPERF_SERVER_ADDRESS "192.168.47.169"

/*******************************************************************************
 * Prototypes
 ******************************************************************************/

extern void test_join(void);
extern wwd_result_t test_scan();
extern wiced_result_t wiced_wlan_connectivity_init(void);
extern void add_wlan_interface(void);
```

Installed SDKs

To install an SDK, simply drag and drop an SDK (zip file/folder) into the 'Installed SDKs' view. [Common 'mcuxpresso' folder]

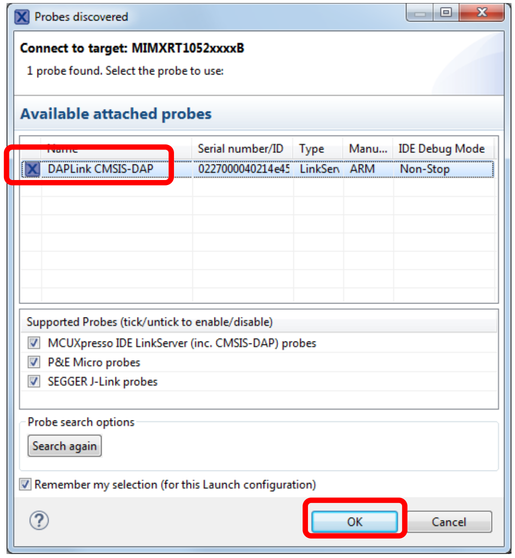
Name	SDK Ver...	Manifes...	Location
☑ SDK_2.x_EVKB-IM	2.5.1	3.4.0	<Common>\SDK_2.5.1_E

SDK Details

Selected SDK content.

NXP MIMXRT1052xxxxB* (evkbi...iperf)

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.



Probes discovered

Connect to target: MIMXRT1052xxxxB

1 probe found. Select the probe to use:

Available attached probes

Name	Serial number/ID	Type	Manu...	IDE Debug Mode
DAPLink CMSIS-DAP	0227000040214e45	LinkSen	ARM	Non-Stop

Supported Probes (tick/untick to enable/disable)

☑ MCUXpresso IDE LinkServer (inc. CMSIS-DAP) probes

☑ P&E Micro probes

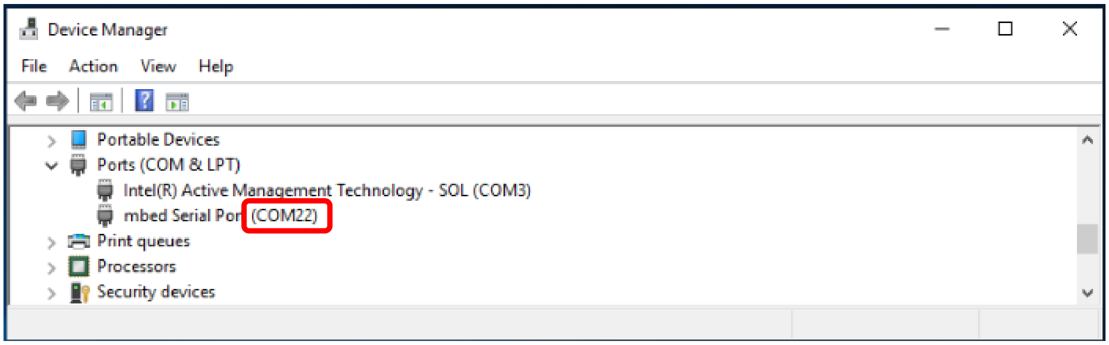
☑ SEGGER J-Link probes

Probe search options

Search again

☑ Remember my selection (for this Launch configuration)

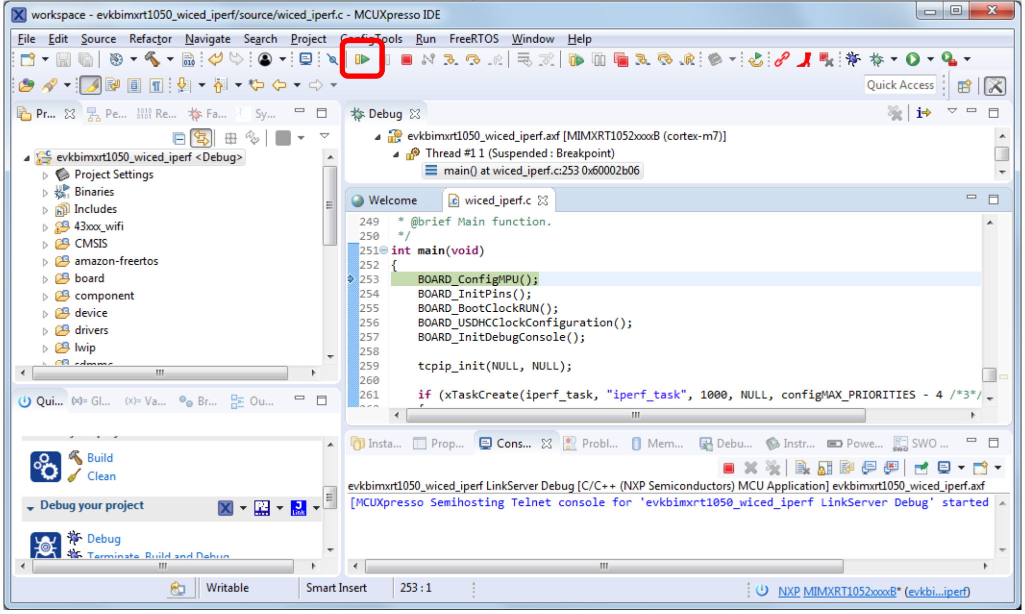
7. After the Debug process is complete, open the “Device Manager” in windows and go to “Ports (COM & LPT)” to check the mbed COM port number. In this case, COM 22 is our COM port number.



Device Manager

- Portable Devices
- Ports (COM & LPT)
  - Intel(R) Active Management Technology - SOL (COM3)
  - mbed Serial Port (COM22)
- Print queues
- Processors
- Security devices

8. Now the iperf example is ready to run. Open TeraTerm on the appropriate COM port (i.e. COM 22 in this case) and set 115200bps as the baudrate, then click resume button in MCUXpresso.

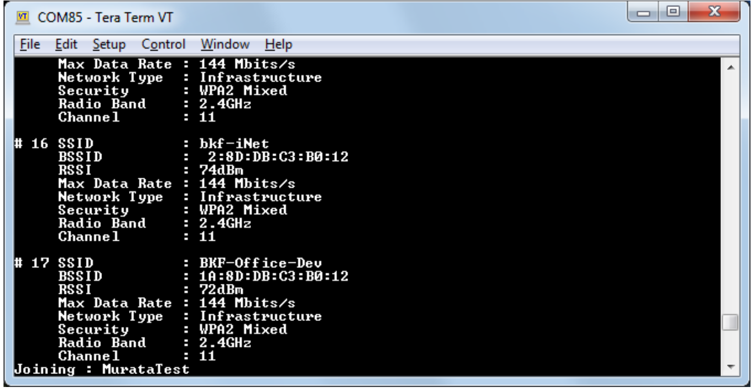

workspace - evkbimxrt1050_wiced_iperf/source/wiced_iperf.c - MCUXpresso IDE
File Edit Source Refactor Navigate Search Project ConfigTools Run FreeRTOS Window Help
Quick Access
Pr... Pe... Re... Fa... Sy...
evkbimxrt1050_wiced_iperf <Debug>
Project Settings
Binaries
Includes
43xxx_wifi
CMSIS
amazon-freertos
board
component
device
drivers
lwip
Qui... Gl... Va... Br... Ou...
Build
Clean
Debug your project
Debug
Terminate, Build and Debug
Debug
Welcome
wiced_iperf.c
Insta... Prop... Cons... Probl... Mem... Debu... Instr... Powe... SWO ...
Writable
Smart Insert
253 : 1
NXP MIMXRT1052xxxxB* (evkbi...iperf)

```
evkbimxrt1050_wiced_iperf.axf [MIMXRT1052xxxxB (cortex-m7)]
  Thread #1 1 (Suspended : Breakpoint)
    main() at wiced_iperf.c:253 0x60002b06
```

```
 * @brief Main function.
 */
int main(void)
{
    BOARD_ConfigMPU();
    BOARD_InitPins();
    BOARD_BootClockRUN();
    BOARD_USDHCClockConfiguration();
    BOARD_InitDebugConsole();

    tcpip_init(NULL, NULL);

    if (xTaskCreate(iperf_task, "iperf_task", 1000, NULL, configMAX_PRIORITIES - 4 /*3*/
```

```
evkbimxrt1050_wiced_iperf LinkServer Debug [C/C++ (NXP Semiconductors) MCU Application] evkbimxrt1050_wiced_iperf.axf
[MCUXpresso Semihosting Telnet console for 'evkbimxrt1050_wiced_iperf LinkServer Debug' started
```

You should see the output from i.MX RT.

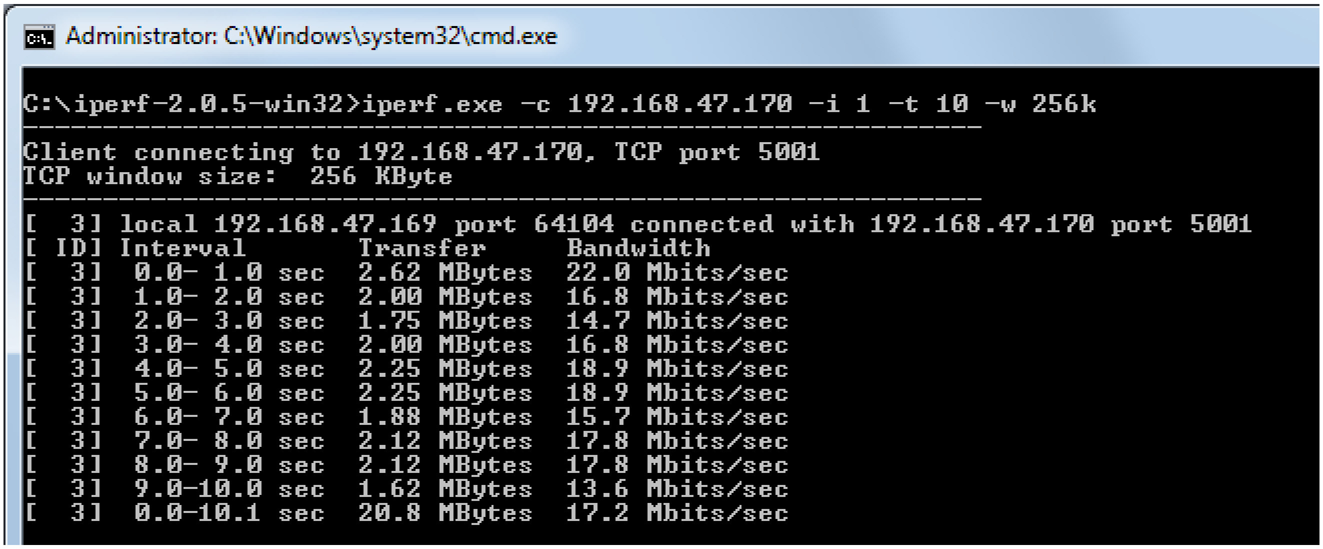


```
     Max Data Rate : 144 Mbits/s
     Network Type  : Infrastructure
     Security      : WPA2 Mixed
     Radio Band    : 2.4GHz
     Channel       : 11

# 16 SSID          : bkf-iNet
     BSSID         :  2:8D:DB:C3:B0:12
     RSSI          : 74dBm
     Max Data Rate : 144 Mbits/s
     Network Type  : Infrastructure
     Security      : WPA2 Mixed
     Radio Band    : 2.4GHz
     Channel       : 11

# 17 SSID          : BKF-Office-Dev
     BSSID         : 1A:8D:DB:C3:B0:12
     RSSI          : 72dBm
     Max Data Rate : 144 Mbits/s
     Network Type  : Infrastructure
     Security      : WPA2 Mixed
     Radio Band    : 2.4GHz
     Channel       : 11
Joining : MurataTest
```

The sample wiced_iperf support several options. Let’s try iperf server. Type ‘1’ on TeraTerm.

```
IPv4 Address got from DHCP : 192.168.47.170

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

    1: server mode (RX only test)
    2: client mode (TX only test)
    3: client dual mode (TX and RX in parallel)
    4: client tradeoff mode (TX and RX sequentially)

Enter mode number: Initializing WiFi Connection...
```

9. Run iperf on your computer, then you should see the result like this.
iperf.exe -c 192.168.47.170 -i 1 -t 10 -w 256k

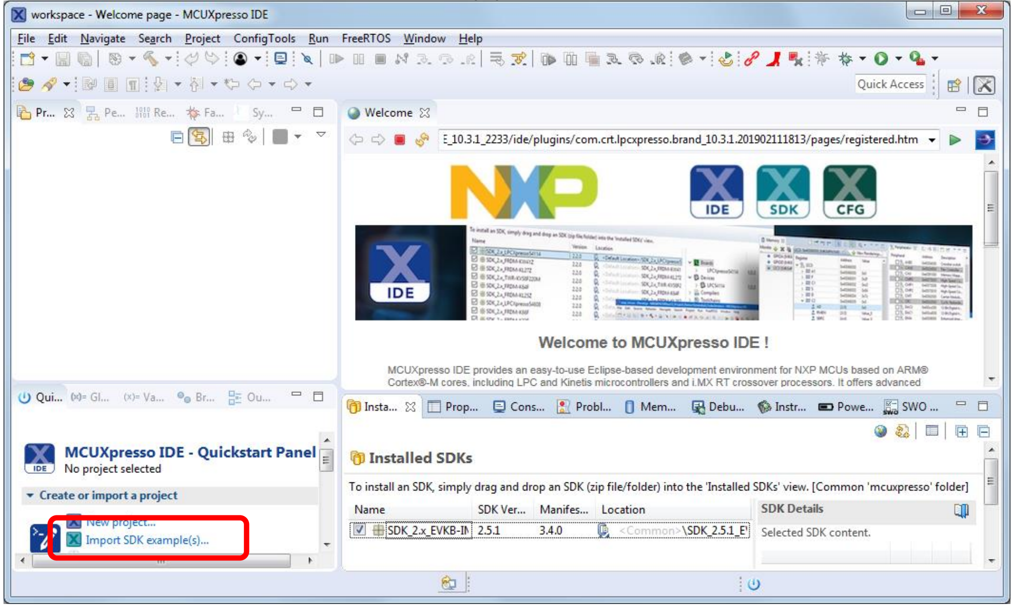


```
C:\iperf-2.0.5-win32>iperf.exe -c 192.168.47.170 -i 1 -t 10 -w 256k
------------------------------------------------------------
Client connecting to 192.168.47.170, TCP port 5001
TCP window size:  256 KByte
------------------------------------------------------------
[  3] local 192.168.47.169 port 64104 connected with 192.168.47.170 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec  2.62 MBytes  22.0 Mbits/sec
[  3]  1.0- 2.0 sec  2.00 MBytes  16.8 Mbits/sec
[  3]  2.0- 3.0 sec  1.75 MBytes  14.7 Mbits/sec
[  3]  3.0- 4.0 sec  2.00 MBytes  16.8 Mbits/sec
[  3]  4.0- 5.0 sec  2.25 MBytes  18.9 Mbits/sec
[  3]  5.0- 6.0 sec  2.25 MBytes  18.9 Mbits/sec
[  3]  6.0- 7.0 sec  1.88 MBytes  15.7 Mbits/sec
[  3]  7.0- 8.0 sec  2.12 MBytes  17.8 Mbits/sec
[  3]  8.0- 9.0 sec  2.12 MBytes  17.8 Mbits/sec
[  3]  9.0-10.0 sec  1.62 MBytes  13.6 Mbits/sec
[  3]  0.0-10.1 sec  20.8 MBytes  17.2 Mbits/sec
```

To find more details about what version of iperf to use and where to download it from, you can refer to the readme file in the SDK example. You can find the readme file in: evkbimxrt1050_wiced_iperf/doc/readme.txt.

5.2 Example wiced_mfg_test

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


workspace - Welcome page - MCUXpresso IDE
File Edit Navigate Search Project ConfigTools Run FreeRTOS Window Help
Quick Access
Pr... Pe... Re... Fa... Sy...
Welcome
E_10.3.1_2233/ide/plugins/com.crt.lpcxpresso.brand_10.3.1.201902111813/pages/registered.htm
NXP
IDE
SDK
CFG
IDE
Qui... Gl... Va... Br... Ou...
MCUXpresso IDE - Quickstart Panel
No project selected
Create or import a project
New project...
Import SDK example(s)...
Insta... Prop... Cons... Probl... Mem... Debu... Instr... Powe... SWO ...
Installed SDKs
SDK Details
Selected SDK content.

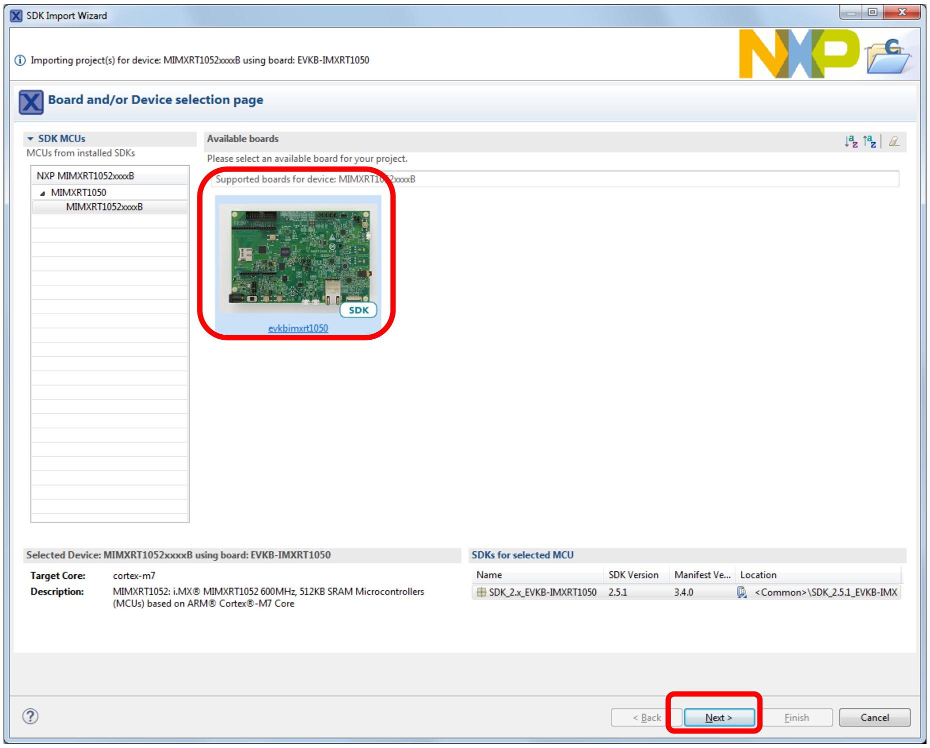
Welcome to MCUXpresso IDE !

MCUXpresso IDE provides an easy-to-use Eclipse-based development environment for NXP MCUs based on ARM® Cortex®-M cores, including LPC and Kinetis microcontrollers and i.MX RT crossover processors. It offers advanced

To install an SDK, simply drag and drop an SDK (zip file/folder) into the 'Installed SDKs' view. [Common 'mcuxpresso' folder]

Name	SDK Ver...	Manifes...	Location
☑ SDK_2.x_EVKB-IN	2.5.1	3.4.0	<Common>\SDK_2.5.1_E

2. Select “evkimxrt1050” board and click Next button.



SDK Import Wizard

Importing project(s) for device: MIMXRT1052xxxxB using board: EVKB-IMXRT1050

Board and/or Device selection page

SDK MCUs

MCUs from installed SDKs

- NXP MIMXRT1052xxxxB
  - MIMXRT1050
    - MIMXRT1052xxxxB

Available boards

Please select an available board for your project.

Supported boards for device: MIMXRT1052xxxxB

evkbimxrt1050

Selected Device: MIMXRT1052xxxxB using board: EVKB-IMXRT1050

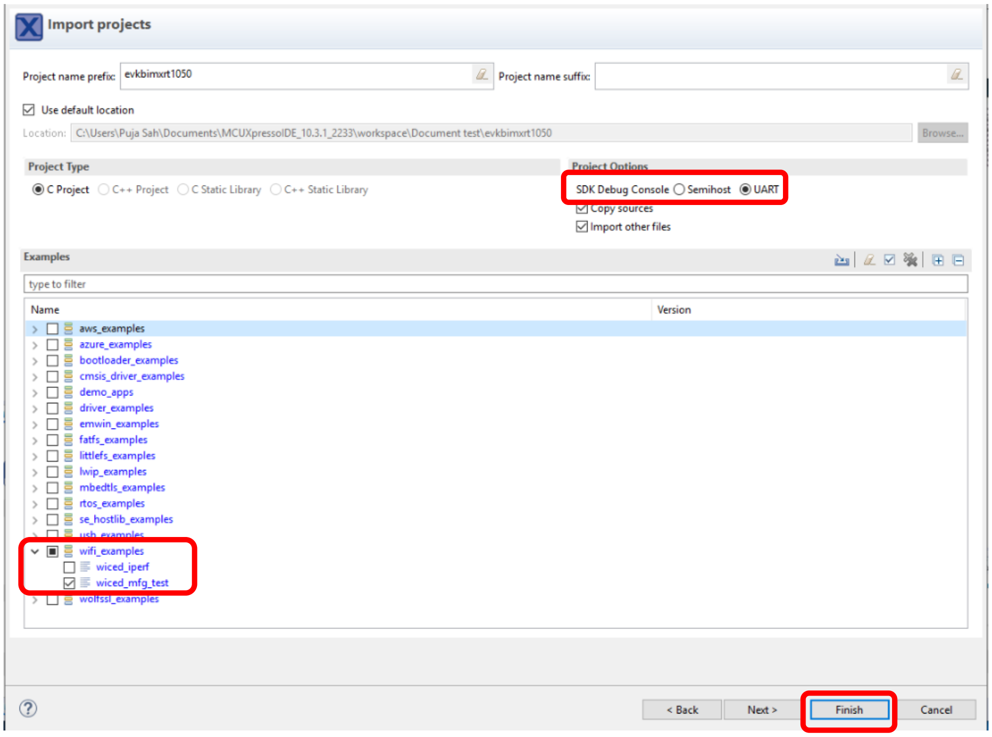
Target Core: cortex-m7

Description: MIMXRT1052: i.MX® MIMXRT1052 600MHz, 512KB SRAM Microcontrollers (MCUs) based on ARM® Cortex®-M7 Core

SDKs for selected MCU

Name	SDK Version	Manifest Ve...	Location
SDK_2.x_EVKB-IMXRT1050	2.5.1	3.4.0	<Common>\SDK_2.5.1_EVKB-IMX

3. Select wifi_examples - wiced_mfg_test. Select UART for SDK Debug Console, then click Finish button.



Import projects

Project name prefix: evkbimxrt1050

Project name suffix:

☑ Use default location

Location: C:\Users\Puja Sah\Documents\MCUXpressoIDE_10.3.1_2233\workspace\Document test\evkbimxrt1050

Project Type

● C Project ○ C++ Project ○ C Static Library ○ C++ Static Library

Project Options

SDK Debug Console ○ Semihost ● UART

☑ Copy sources

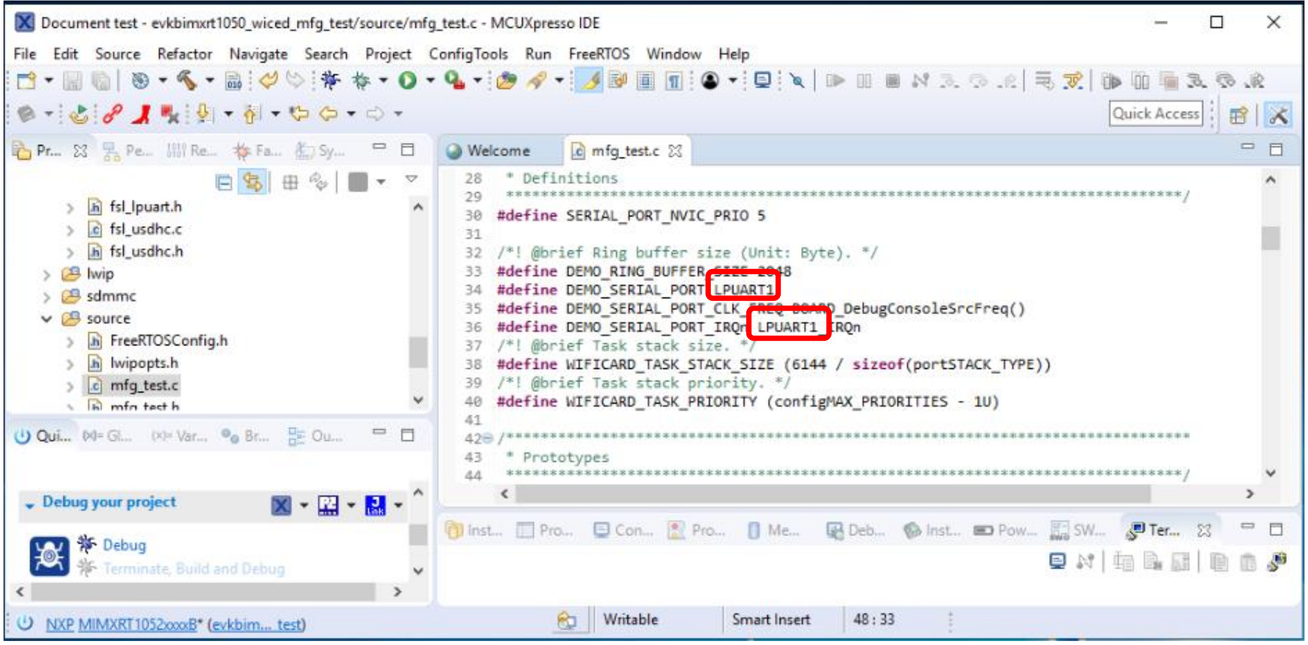
☑ Import other files

Examples

type to filter

Name	Version
□ aws_examples	
□ azure_examples	
□ bootloader_examples	
□ cmsis_driver_examples	
□ demo_apps	
□ driver_examples	
□ emwin_examples	
□ fatfs_examples	
□ littlefs_examples	
□ lwip_examples	
□ mbedtls_examples	
□ rtos_examples	
□ se_hostlib_examples	
□ usb_examples	
■ wifi_examples	
□ wiced_iperf	
☑ wiced_mfg_test	
□ wolfssl_examples	

4. Open source/mfg_test.c and modify the LPUART3 to LPUART1.

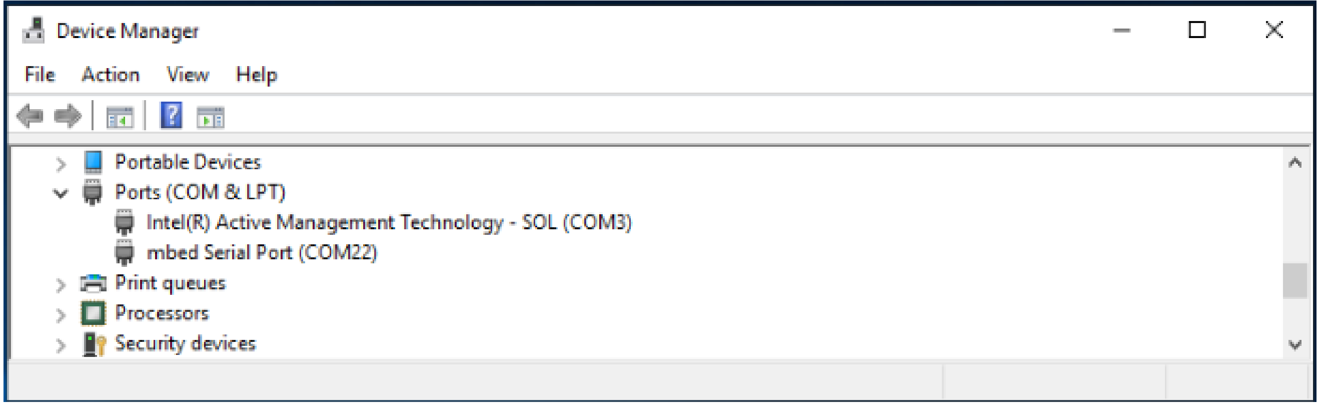


Document test - evkbimxrt1050_wiced_mfg_test/source/mfg_test.c - MCUXpresso IDE
File Edit Source Refactor Navigate Search Project ConfigTools Run FreeRTOS Window Help
Quick Access
Pr... Pe... Re... Fa... Sy...
fsl_lpuart.h
fsl_usdhc.c
fsl_usdhc.h
lwip
sdmmc
source
FreeRTOSConfig.h
lwipopts.h
mfg_test.c
mfg_test.h
Qui... Gl... Var... Br... Ou...
Debug your project
Debug
Terminate, Build and Debug
Welcome
mfg_test.c
Inst... Pro... Con... Pro... Me... Deb... Inst... Pow... SW... Ter...
NXP MIMXRT1052xxxxB* (evkbim... test)
Writable
Smart Insert
48 : 33

```
 * Definitions
 ******************************************************************************/
#define SERIAL_PORT_NVIC_PRIO 5

/*! @brief Ring buffer size (Unit: Byte). */
#define DEMO_RING_BUFFER_SIZE 2048
#define DEMO_SERIAL_PORT LPUART1
#define DEMO_SERIAL_PORT_CLK_FREQ BOARD_DebugConsoleSrcFreq()
#define DEMO_SERIAL_PORT_IRQn LPUART1_IRQn
/*! @brief Task stack size. */
#define WIFICARD_TASK_STACK_SIZE (6144 / sizeof(portSTACK_TYPE))
/*! @brief Task stack priority. */
#define WIFICARD_TASK_PRIORITY (configMAX_PRIORITIES - 1U)

/*******************************************************************************
 * Prototypes
 ******************************************************************************/
```

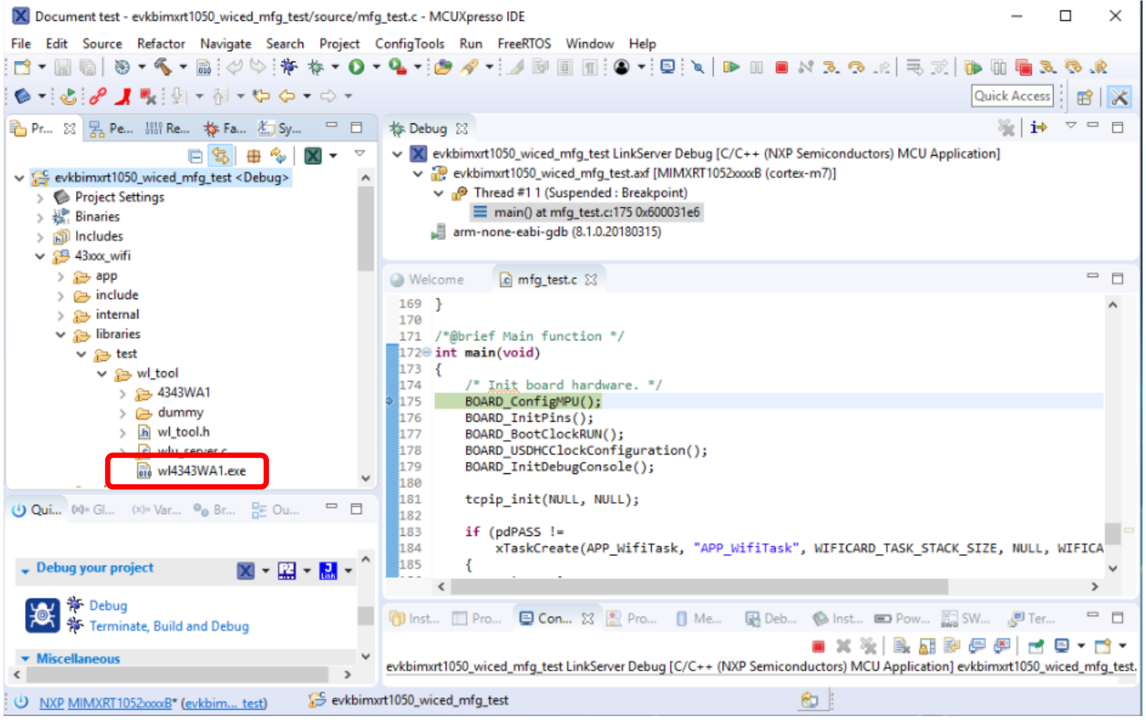
5. Save the file and then click on “Debug” button in the Quickstart Panel.

6. 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.



- Portable Devices
- Ports (COM & LPT)
  - Intel(R) Active Management Technology - SOL (COM3)
  - mbed Serial Port (COM22)
- Print queues
- Processors
- Security devices

7. Open 43xx_wifi/libraries/test/wl_tool and right click on wl4343WA1.exe.



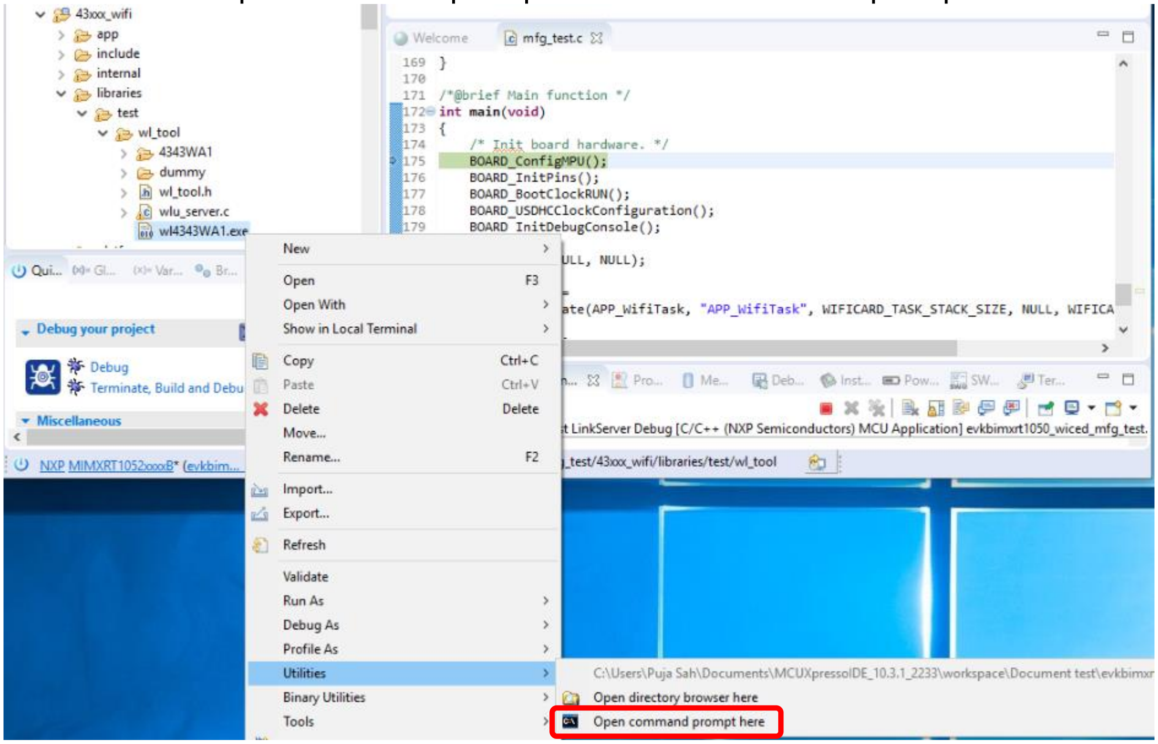
Document test - evkbimxrt1050_wiced_mfg_test/source/mfg_test.c - MCUXpresso IDE
File Edit Source Refactor Navigate Search Project ConfigTools Run FreeRTOS Window Help
Quick Access
Pr... Pe... Re... Fa... Sy...
evkbimxrt1050_wiced_mfg_test <Debug>
Project Settings
Binaries
Includes
43xx_wifi
app
include
internal
libraries
test
wl_tool
4343WA1
dummy
wl_tool.h
wlu_server.c
wl4343WA1.exe
Qui... Gl... Var... Br... Ou...
Debug your project
Debug
Terminate, Build and Debug
Miscellaneous
NXP MIMXRT1052xxxxB* (evkbim... test)
evkbimxrt1050_wiced_mfg_test
Debug
Welcome
mfg_test.c
Inst... Pro... Con... Pro... Me... Deb... Inst... Pow... SW... Ter...

```
evkbimxrt1050_wiced_mfg_test LinkServer Debug [C/C++ (NXP Semiconductors) MCU Application]
  evkbimxrt1050_wiced_mfg_test.axf [MIMXRT1052xxxxB (cortex-m7)]
    Thread #1 1 (Suspended : Breakpoint)
      main() at mfg_test.c:175 0x600031e6
  arm-none-eabi-gdb (8.1.0.20180315)
```

```
}

/*@brief Main function */
int main(void)
{
    /* Init board hardware. */
    BOARD_ConfigMPU();
    BOARD_InitPins();
    BOARD_BootClockRUN();
    BOARD_USDHCClockConfiguration();
    BOARD_InitDebugConsole();

    tcpip_init(NULL, NULL);

    if (pdPASS !=
        xTaskCreate(APP_WifiTask, "APP_WifiTask", WIFICARD_TASK_STACK_SIZE, NULL, WIFICA
    {
```

evkbimxrt1050_wiced_mfg_test LinkServer Debug [C/C++ (NXP Semiconductors) MCU Application] evkbimxrt1050_wiced_mfg_test.

Select Utilities/Open command prompt here and a command prompt window will come up.



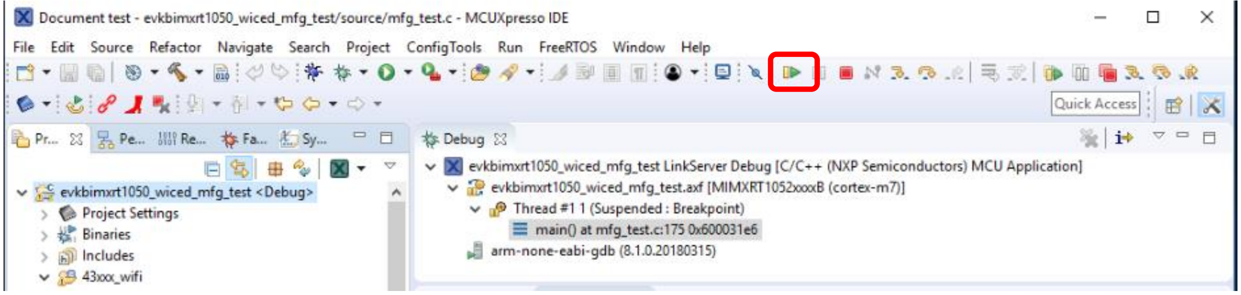
```
}

/*@brief Main function */
int main(void)
{
    /* Init board hardware. */
    BOARD_ConfigMPU();
    BOARD_InitPins();
    BOARD_BootClockRUN();
    BOARD_USDHCClockConfiguration();
    BOARD_InitDebugConsole();

ULL, NULL);

ate(APP_WifiTask, "APP_WifiTask", WIFICARD_TASK_STACK_SIZE, NULL, WIFICA
```

t LinkServer Debug [C/C++ (NXP Semiconductors) MCU Application] evkbimxrt1050_wiced_mfg_test.

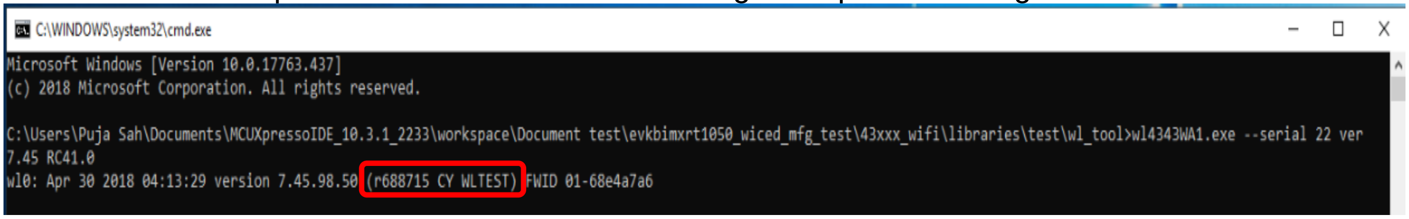
_test/43xx_wifi/libraries/test/wl_tool

C:\Users\Puja Sah\Documents\MCUXpressoIDE_10.3.1_2233\workspace\Document test\evkbimxr

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


Document test - evkbimxrt1050_wiced_mfg_test/source/mfg_test.c - MCUXpresso IDE
File Edit Source Refactor Navigate Search Project ConfigTools Run FreeRTOS Window Help
Quick Access
Pr... Pe... Re... Fa... Sy...
evkbimxrt1050_wiced_mfg_test <Debug>
Project Settings
Binaries
Includes
43xxx_wifi
Debug
evkbimxrt1050_wiced_mfg_test LinkServer Debug [C/C++ (NXP Semiconductors) MCU Application]
evkbimxrt1050_wiced_mfg_test.axf [MIMXRT1052xxxxB (cortex-m7)]
Thread #1 1 (Suspended : Breakpoint)
main() at mfg_test.c:175 0x600031e6
arm-none-eabi-gdb (8.1.0.20180315)

9. Type “wl4343WA1.exe - -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.



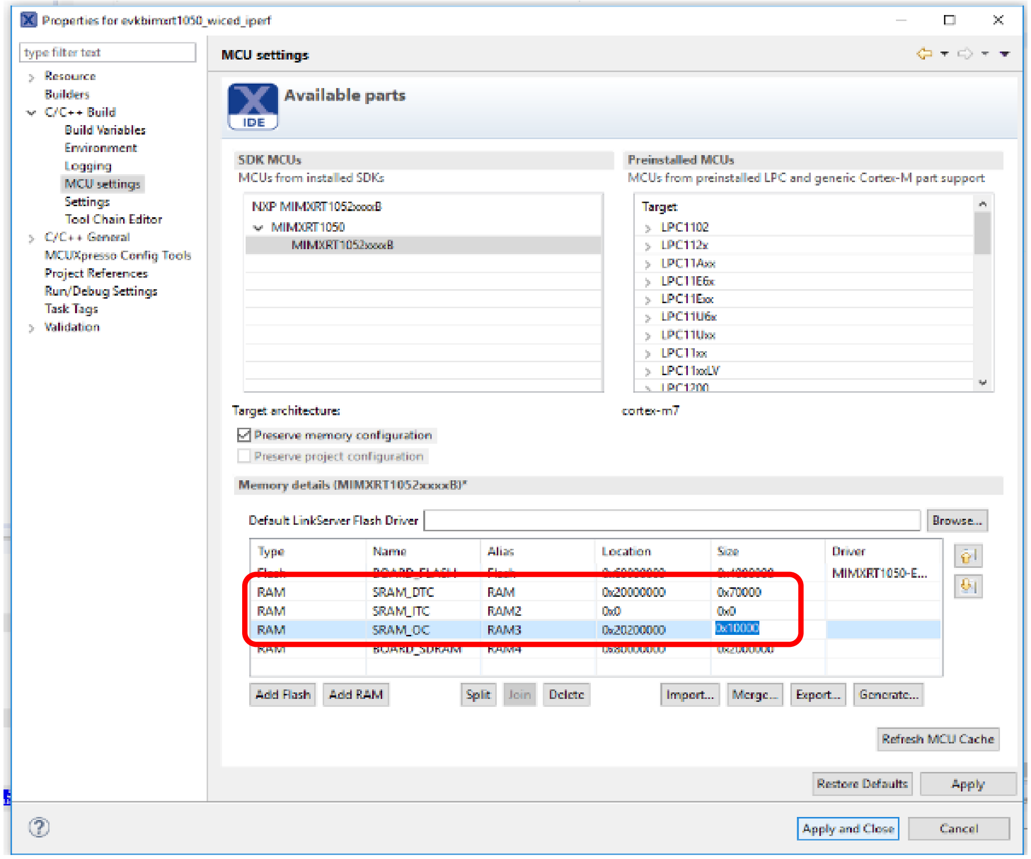
```
Microsoft Windows [Version 10.0.17763.437]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Puja Sah\Documents\MCUXpressoIDE_10.3.1_2233\workspace\Document test\evkbimxrt1050_wiced_mfg_test\43xxx_wifi\libraries\test\wl_tool>wl4343WA1.exe --serial 22 ver
7.45 RC41.0
wl0: Apr 30 2018 04:13:29 version 7.45.98.50 (r688715 CY WLTEST) FWID 01-68e4a7a6
```

6 Throughput Improvement

The throughput can be improved by making some changes in the wiced_iperf example. The changes include increasing TCP window size and number of buffers and changing memory size. This section describes the steps to increase throughput only for NXP's i.MX RT 1050 EVKB. To improve the throughput on EA's i.MX RT 1052/1062 Developer's kit, please refer to the "Getting Started with M.2 modules and i.MX RT" downloaded in **Section 3.**

1. First run wiced_iperf example successfully as described in **Section 5.1**.

2. Select project/properties/MCU settings and change the SRAM_DTC value to 0x70000, SRAM_ITC value to 0x0 and SRAM_OC value to 0x10000.



Target architecture: cortex-m7

☑ Preserve memory configuration

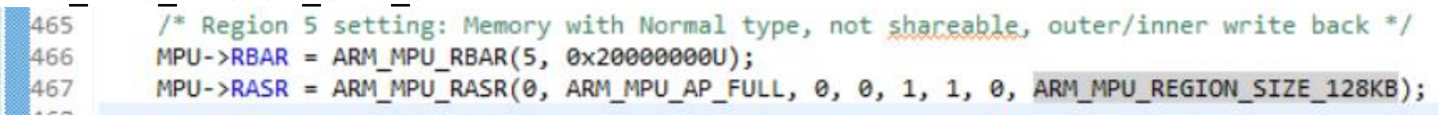
☐ Preserve project configuration

Memory details (MIMXRT1052xxxxB)*

Default LinkServer Flash Driver [ ] Browse...

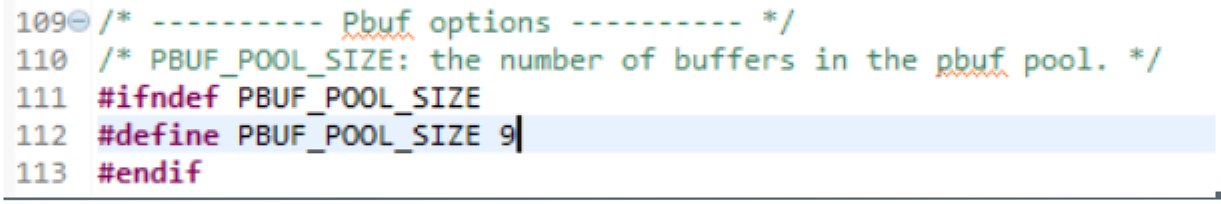
Type	Name	Alias	Location	Size	Driver
Flash	BOARD_FLASH	Flash	0x60000000	0x4000000	MIMXRT1050-E...
RAM	SRAM_DTC	RAM	0x20000000	0x70000	
RAM	SRAM_ITC	RAM2	0x0	0x0	
RAM	SRAM_OC	RAM3	0x20200000	0x10000	
RAM	BOARD_SDRAM	RAM4	0x80000000	0x2000000	

3. Open board/board.c and locate these lines. Replace ARM_MPU_REGION_SIZE_128KB to ARM_MPU_REGION_SIZE_512KB.



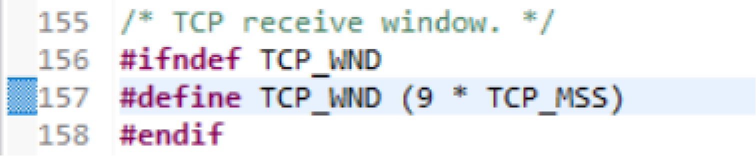
```
    /* Region 5 setting: Memory with Normal type, not shareable, outer/inner write back */
    MPU->RBAR = ARM_MPU_RBAR(5, 0x20000000U);
    MPU->RASR = ARM_MPU_RASR(0, ARM_MPU_AP_FULL, 0, 0, 1, 1, 0, ARM_MPU_REGION_SIZE_128KB);
```

4. Open source/lwipopts.h and locate these lines. Change from 9 to 44 buffers in the pool.



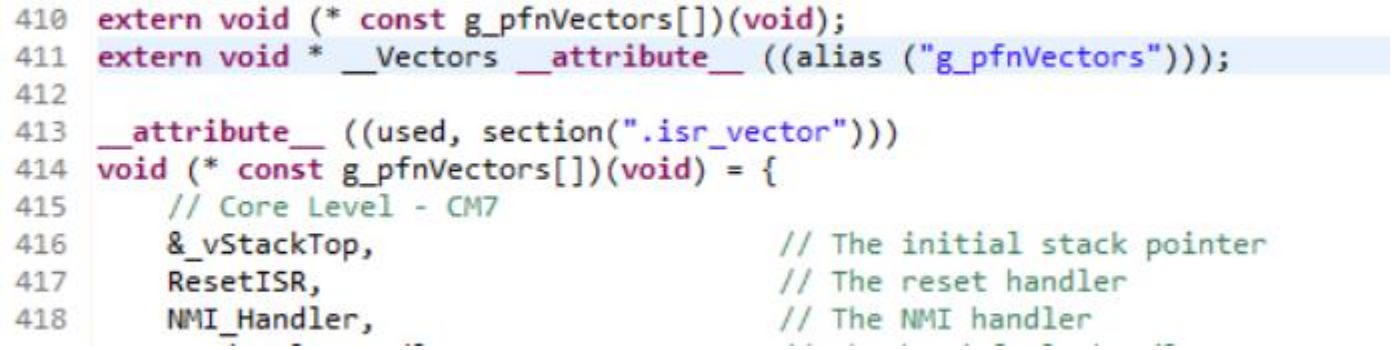
```
/* ---------- Pbuf options ---------- */
/* PBUF_POOL_SIZE: the number of buffers in the pbuf pool. */
#ifndef PBUF_POOL_SIZE
#define PBUF_POOL_SIZE 9
#endif
```

5. In the same file replace the constant 9 with 44 in TCP receive window.



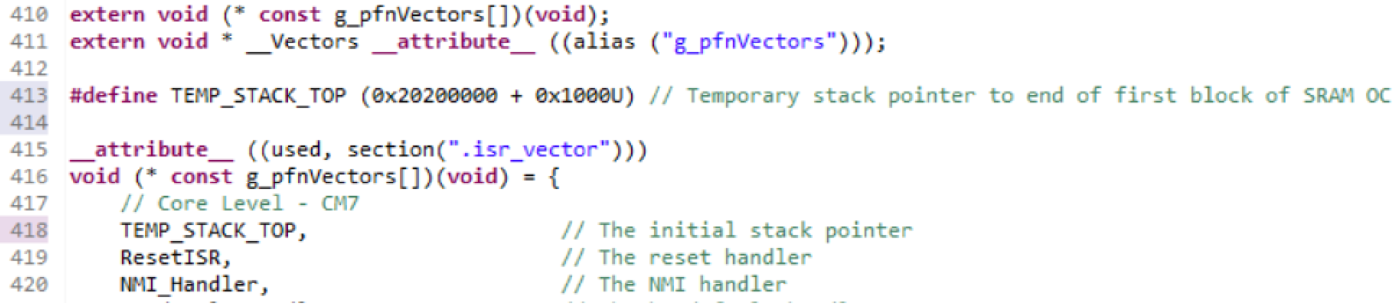
```
/* TCP receive window. */
#ifndef TCP_WND
#define TCP_WND (9 * TCP_MSS)
#endif
```

6. Open startup/startup_mimxrt1052.c and locate these lines.



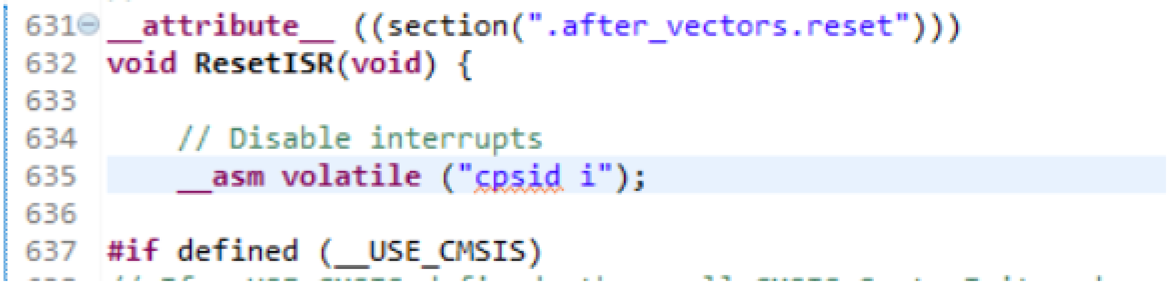
```
extern void (* const g_pfnVectors[])(void);
extern void * __Vectors __attribute__ ((alias ("g_pfnVectors")));

__attribute__ ((used, section(".isr_vector")))
void (* const g_pfnVectors[])(void) = {
    // Core Level - CM7
    &_vStackTop,                        // The initial stack pointer
    ResetISR,                           // The reset handler
    NMI_Handler,                        // The NMI handler
```

Add line #define TEMP_STACK_TOP (0x20200000 + 0x1000U). Replace &_vStackTop with TEMP_STACK_TOP.



```
extern void (* const g_pfnVectors[])(void);
extern void * __Vectors __attribute__ ((alias ("g_pfnVectors")));

#define TEMP_STACK_TOP (0x20200000 + 0x1000U) // Temporary stack pointer to end of first block of SRAM OC

__attribute__ ((used, section(".isr_vector")))
void (* const g_pfnVectors[])(void) = {
    // Core Level - CM7
    TEMP_STACK_TOP,                     // The initial stack pointer
    ResetISR,                           // The reset handler
    NMI_Handler,                        // The NMI handler
```

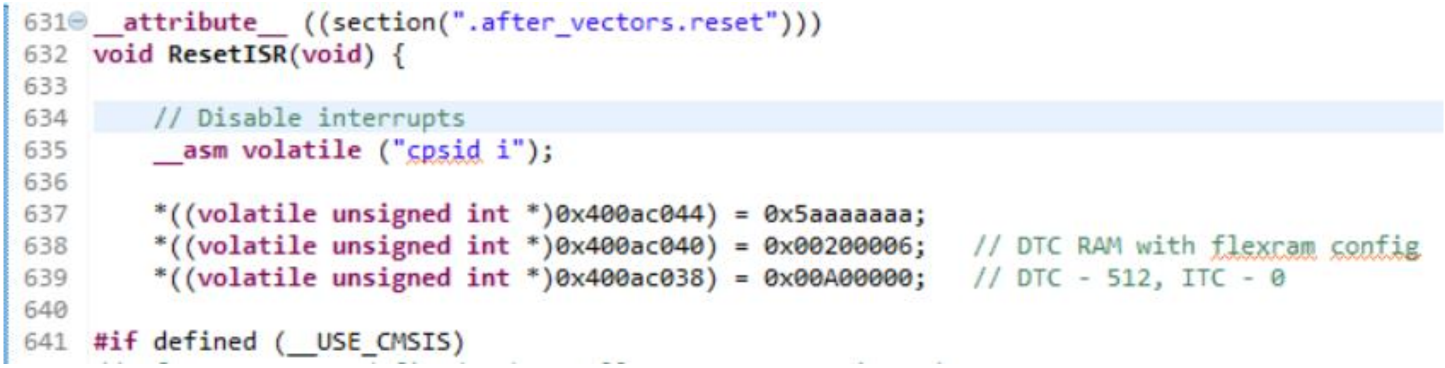
7. In the same file (startup/startup mimxrt1052.c) locate the ResetISR function.



```
__attribute__ ((section(".after_vectors.reset")))
void ResetISR(void) {

    // Disable interrupts
    __asm volatile ("cpsid i");

#if defined (__USE_CMSIS)
```

Insert the following lines after the _asm… line:

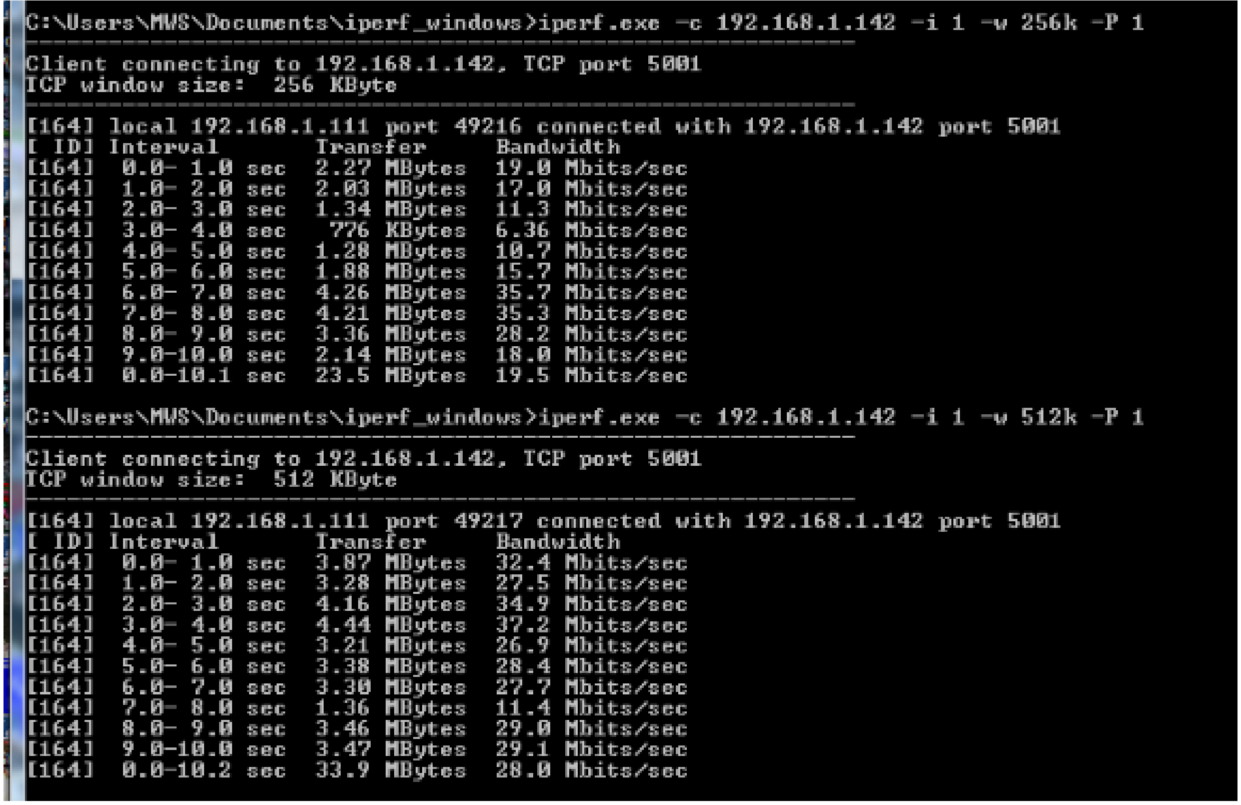
```
    *((volatile unsigned int *)0x400ac044) = 0x5aaaaaaa;
    *((volatile unsigned int *)0x400ac040) = 0x00200006; // DTC RAM with flexram config
    *((volatile unsigned int *)0x400ac038) = 0x00A00000; // DTC - 512, ITC - 0
```



```
__attribute__ ((section(".after_vectors.reset")))
void ResetISR(void) {

    // Disable interrupts
    __asm volatile ("cpsid i");

    *((volatile unsigned int *)0x400ac044) = 0x5aaaaaaa;
    *((volatile unsigned int *)0x400ac040) = 0x00200006;   // DTC RAM with flexram config
    *((volatile unsigned int *)0x400ac038) = 0x00A00000;   // DTC - 512, ITC - 0

#if defined (__USE_CMSIS)
```

8. Compile and run. Type “1” in the terminal window to start server mode.

9. Run the iperf command on the PC: “iperf - c 192.168.1.142 -i 1 -w 256k -P 1”. You will see the following output on the PC running iperf command.

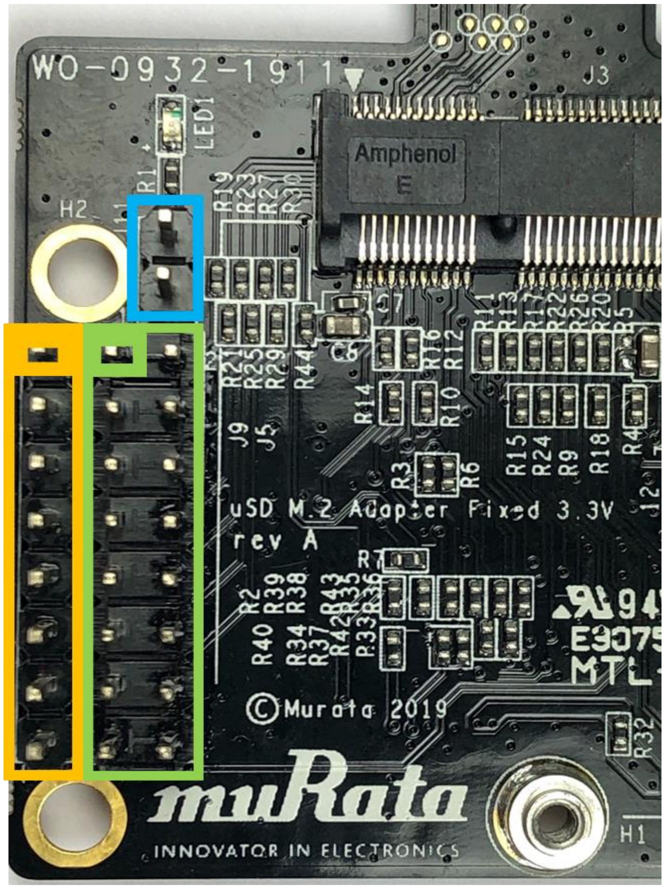


```
C:\Users\MWS\Documents\iperf_windows>iperf.exe -c 192.168.1.142 -i 1 -w 256k -P 1
------------------------------------------------------------
Client connecting to 192.168.1.142, TCP port 5001
TCP window size:  256 KByte
------------------------------------------------------------
[164] local 192.168.1.111 port 49216 connected with 192.168.1.142 port 5001
[ ID] Interval       Transfer     Bandwidth
[164]  0.0- 1.0 sec  2.27 MBytes  19.0 Mbits/sec
[164]  1.0- 2.0 sec  2.03 MBytes  17.0 Mbits/sec
[164]  2.0- 3.0 sec  1.34 MBytes  11.3 Mbits/sec
[164]  3.0- 4.0 sec   776 KBytes  6.36 Mbits/sec
[164]  4.0- 5.0 sec  1.28 MBytes  10.7 Mbits/sec
[164]  5.0- 6.0 sec  1.88 MBytes  15.7 Mbits/sec
[164]  6.0- 7.0 sec  4.26 MBytes  35.7 Mbits/sec
[164]  7.0- 8.0 sec  4.21 MBytes  35.3 Mbits/sec
[164]  8.0- 9.0 sec  3.36 MBytes  28.2 Mbits/sec
[164]  9.0-10.0 sec  2.14 MBytes  18.0 Mbits/sec
[164]  0.0-10.1 sec  23.5 MBytes  19.5 Mbits/sec

C:\Users\MWS\Documents\iperf_windows>iperf.exe -c 192.168.1.142 -i 1 -w 512k -P 1
------------------------------------------------------------
Client connecting to 192.168.1.142, TCP port 5001
TCP window size:  512 KByte
------------------------------------------------------------
[164] local 192.168.1.111 port 49217 connected with 192.168.1.142 port 5001
[ ID] Interval       Transfer     Bandwidth
[164]  0.0- 1.0 sec  3.87 MBytes  32.4 Mbits/sec
[164]  1.0- 2.0 sec  3.28 MBytes  27.5 Mbits/sec
[164]  2.0- 3.0 sec  4.16 MBytes  34.9 Mbits/sec
[164]  3.0- 4.0 sec  4.44 MBytes  37.2 Mbits/sec
[164]  4.0- 5.0 sec  3.21 MBytes  26.9 Mbits/sec
[164]  5.0- 6.0 sec  3.38 MBytes  28.4 Mbits/sec
[164]  6.0- 7.0 sec  3.30 MBytes  27.7 Mbits/sec
[164]  7.0- 8.0 sec  1.36 MBytes  11.4 Mbits/sec
[164]  8.0- 9.0 sec  3.46 MBytes  29.0 Mbits/sec
[164]  9.0-10.0 sec  3.47 MBytes  29.1 Mbits/sec
[164]  0.0-10.2 sec  33.9 MBytes  28.0 Mbits/sec
```

7 uSD-M.2 Adapter

Figures below show the details on headers and jumpers. Please refer uSD-M.2 Adapter Datasheet document for more detail. You can find the document here: https://wireless.murata.com/datasheet?/RFM/data/usd-m2_revA.pdf

Figure 9: uSD-M.2 Adapter: Left Headers/Jumpers



J11 = Optional BT Disable; Jumper for WLAN-Only Mode

- ➔ Jumper Installed = BT_REG_ON is Low (BT Core disabled)
- ➔ Not Installed = BT_REG_ON is driven active high by Adapter on-board circuitry (default); or driven by Host if Arduino cable installed (J9; Pin #4).

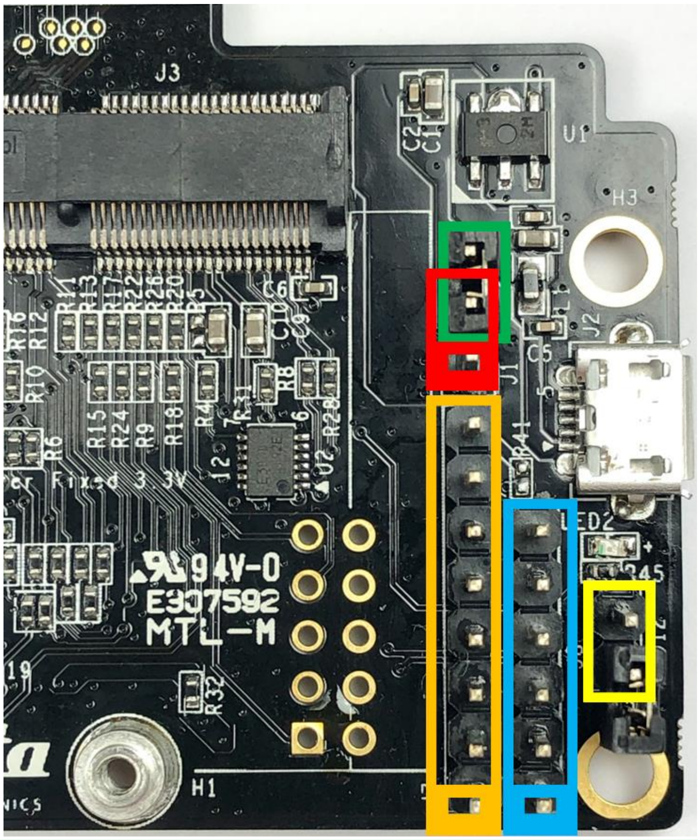
J9 = BT UART TX/RX and WLAN/BT CTRL Arduino Header

Pin#	J9 Signal		Pin#	J9 Signal
1	BT_UART_TXD_3V3		5	WL_HOST_WAKE_3V3
2	BT_UART_RXD_3V3		6	BT_HOST_WAKE_3V3
3	WL_REG_ON_3V3		7	WL_DEV_WAKE_3V3
4	BT_REG_ON_3V3		8	BT_DEV_WAKE_3V3

J5 = Optional BT PCM and WLAN/BT Debug Signals

Pin#	J5 Signal		Pin#	J5 Signal
1	BT_PCM_IN_3V3		14	LPO_IN_3V3
3	BT_PCM_OUT_3V3		15	GND
5	BT_PCM_SYNC_3V3		16	USD_3V3
7	BT_PCM_CLK_3V3			

Figure 10: uSD-M.2 Adapter: Right Headers/Jumpers



J1 = Power Supply Selector

Jumper must be installed to power Adapter

- ➔ 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

J12 = VDDIO Override Short for 3.3V VDDIO (open for 1.8V)

- ➔ Jumper Installed = VDDIO set to 3.3V; LED2 (Blue) illuminates
- ➔ Not Installed = VDDIO set to 1.8V (default)

J7 = Optional Arduino Header Power Supply

Pin#	J7 Signal		Pin#	J7 Signal
2	3V3		6	GND
4	3V3		7	GND
5	5V			

J8 = BT UART RTS/CTS Arduino Header

Pin#	J8 Signal		Pin#	J8 Signal
3	BT_UART_RTS_3V3		4	BT_UART_CTS_3V3

8 Links

8.1 NXP

i.MX RT Community
https://community.nxp.com/community/imxrt

MCUXPresso SDK Builder
https://mcuxpresso.nxp.com

i.MX RT 1050 Evaluation Kit Information
https://www.nxp.com/support/developer-resources/nxp-designs/i.mx-rt1050-evaluation-kit:MIMXRT1050-EVK

i.MX RT 1060 Evaluation Kit Information
https://www.nxp.com/support/developer-resources/nxp-designs/mimxrt1060-evk-i.mx-rt1060-evaluation-kit:MIMXRT1060-EVK

i.MX RT 1050 Getting Started Guide
https://www.nxp.com/document/guide/get-started-with-the-mimxrt1050-evk:GS-MIMXRT1050-EVK

i.MX RT 1060 Getting Started Guide
https://www.nxp.com/document/guide/get-started-with-the-mimxrt1060-evk:GS-MIMXRT1060-EVK

MCUXPresso
https://www.nxp.com/support/developer-resources/software-development-tools/mcuxpresso-software-and-tools/mcuxpresso-software-development-kit-sdk:MCUXpresso-SDK

LPC-Link2
https://www.nxp.com/support/developer-resources/software-development-tools/lpc-developer-resources-/lpc-microcontroller-utilities/lpc-link2:OM13054

8.2 Embedded Artists

Getting Started
https://www.embeddedartists.com/getting-started

iMX RT 1062 Developer’s Kit
https://www.embeddedartists.com/products/imx-rt1062-developers-kit

iMX RT 1052 Developer’s Kit
https://www.embeddedartists.com/products/imx-rt1052-developers-kit/

M.2 Module Family
https://www.embeddedartists.com/m2/