User Guide

Type 2BP UWB Module EVK

Software Development Startup Guide - Rev. 4.5

Table of Contents

Figures

Tables

About This Document

This document provides steps to start software development on Murata Type 2BP EVB (Rev 4.1).

> Enabling 3D AoA or applying calibration values are out of scope of this document.

Audience & Purpose

This guide is for developers and RF engineers who will develop software on Murata Type 2BP EVB (Revision 4.1).

Document Conventions

Table 1 describes the document conventions.

Table 1: Document Conventions

Conventions	Description
	**Warning Note** Indicates very important note. Users are strongly recommended to review.
	**Info Note** Intended for informational purposes. Users should review.
	**Menu Reference** Indicates menu navigation instructions. **Example**: Insert➔Tables➔Quick Tables➔Save Selection to Gallery
	**External Hyperlink** This symbol indicates a hyperlink to an external document or website. **Example**: Type 2BP Product Page  Click on the text to open the external link.
	**Internal Hyperlink** This symbol indicates a hyperlink within the document. **Example**: Prerequisites  Click on the text to open the link.
`Console input/output or code snippet`	**Console I/O or Code Snippet** This text ***Style*** denotes console input/output or a code snippet.
`# Console I/O comment` `// Code snippet comment`	**Console I/O or Code Snippet Comment** This text ***Style*** denotes a console input/output or code snippet comment. • Console I/O comment (preceded by "#") is for informational purposes only and does not denote actual console input/output. • Code Snippet comment (preceded by "//") may exist in the original code.

1 Prerequisites

In this guide, it is assumed that you have gone through Type 2BP EVK Quick Start Guide and installed USB-UART driver. It is also assumed that you have a terminal application (Tera Term, PuTTY, etc.).

The following SDK Package for SR150 is required.

- UWBIOT_SR150_v04.06.00_MCUx.zip

The following hardware are required:

- Type 2BP EVB Rev 4.x and 3.x.

> EVK Rev3.4 or earlier revision needs H/W rework to use with SDK v04.02.01 and later version.
> Please see Type 2BP EVK Rework Guide for Rev 3.

- Micro USB cable
- SWD Adapter: J-Link (SEGGER) or MCU-Link (NXP)
- 10 pin SWD cable (in case of J-Link (SEGGER), SEGGER 9-Pin Cortex-M Adapter or Olimex ARM-JTAG-20-10)

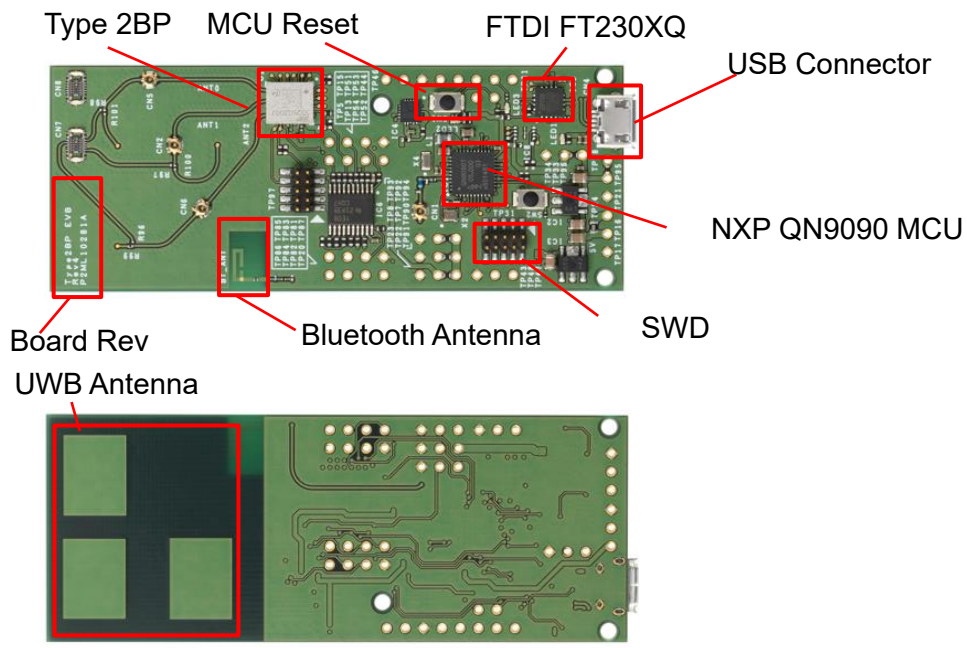
2 Evaluation Board

Type 2BP EVB is designed to be compatible with NXP RhodesV4 evaluation board.

USB Connector provides UART communication between QN9090 MCU and PC via FTDI FT230XQ USB-UART converter. USB Connector is used for power supply as well.

Figure 1 shows the various parts of the evaluation board.

Figure 1: Evaluation Board



3 Hardware Setup

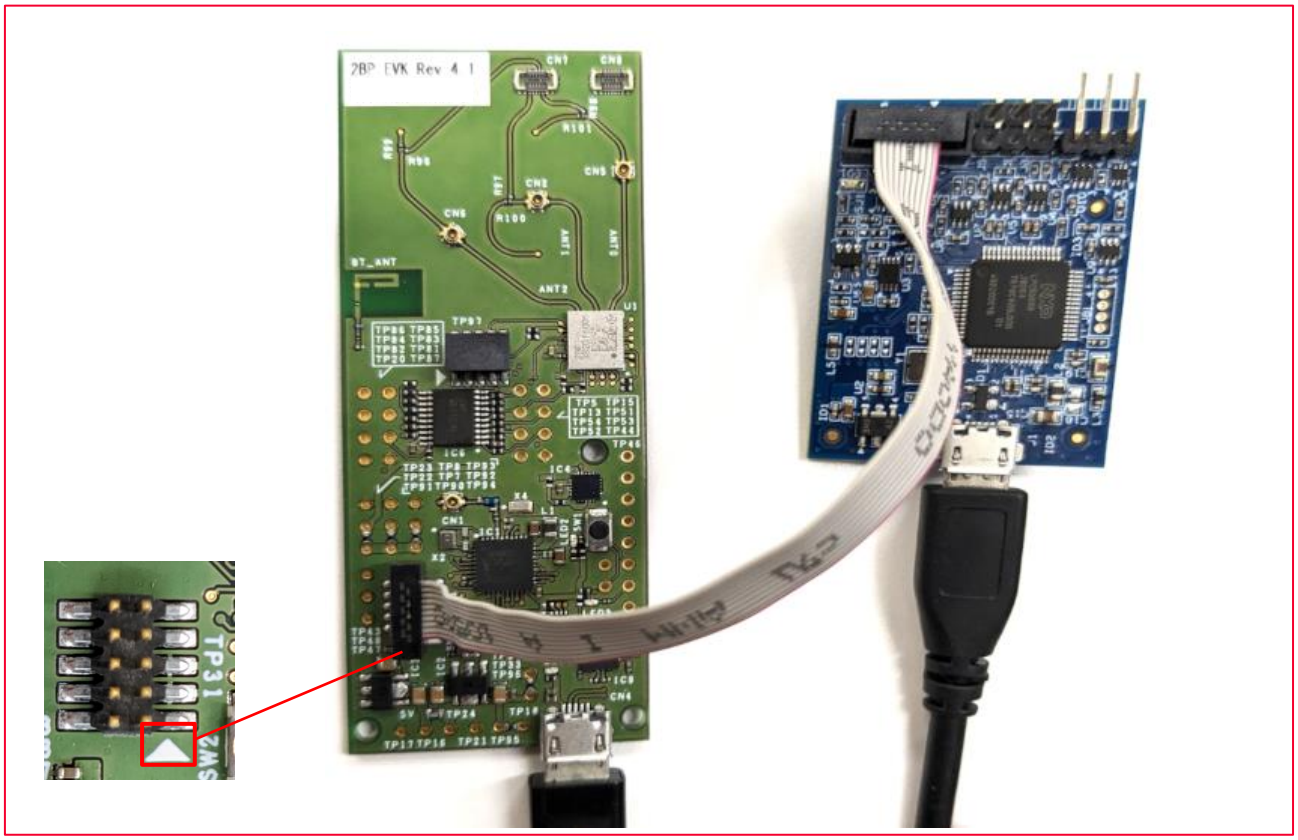
Find pin#1 marking of SWD connector and attach SWD adapter using 10-pin SWD cable as following figure. Attach Type 2BP EVB with your PC using micro-USB cable. Attach SWD adapter with your PC.

When performing debugging (e.g., step instruction) in software development, a debugger such as MCU-Link Debug Probe is required.

Simply rewriting software (SW) is possible with DK6 Programmer tool. This tool is console application program for Windows.

The hardware setup using J-Link debugger is shown in **Figure 2**.

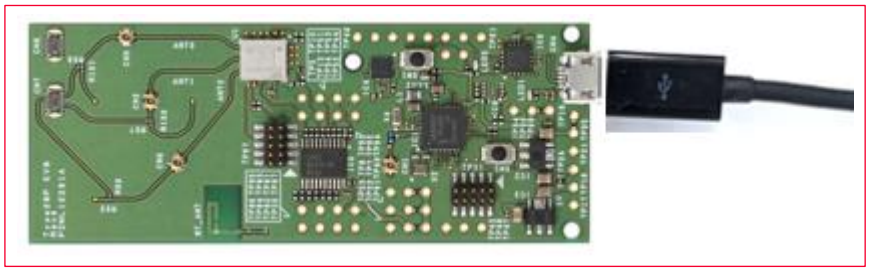
Figure 2: Hardware Setup Using MCU-Link Debug Probe Debugger



Connect the EVK and MCU-Link to PC through USB cable.

Connect the EVK with PC through USB cable using DK6Programmer as shown in **Figure 5**.

Figure 3: Using DK6Programmer



4 Installation

This section describes the installation process.

4.1 Terminal Application

Install Tera Term if you don't have any from:
OSDN - Develop and Download Open-Source Software

4.2 Python

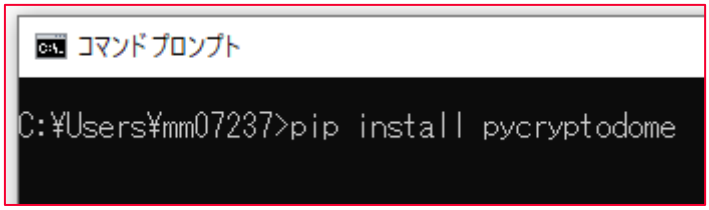
Python 3.7 or later is required. Download from: Python Download

4.3 Python pycryptodome

To develop QN9090 application, pycryptodome library is required. Open Command Prompt and run the command below to install, as shown in **Figure 4**:

```
pip install pycryptodome
```

Figure 4: Install Python Library



4.4 MCUXpresso IDE

Download MCUXpresso IDE 11.3 or later and run the installer. You may need to create NXP account to download the installer from:

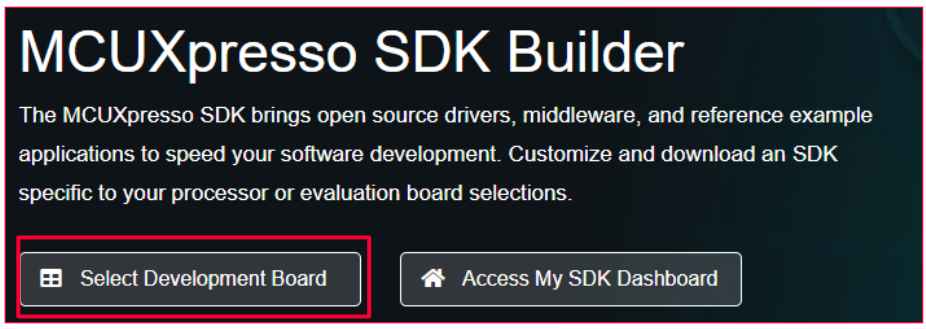
MCUXpresso Integrated Development Environment (IDE)

4.5 Download QN9090 SDK

QN9090 SDK is the basic SDK for MCUXpresso.

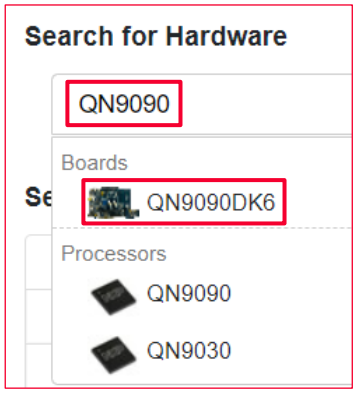
1. Visit the SDK site, and login with NXP credentials.
2. In case you see welcome page, click Select Development Board button, as shown in **Figure 5**.

Figure 5: MCUXpresso SDK Builder



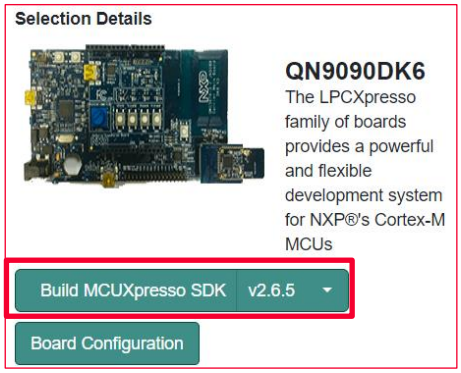
3. Type “QN9090” in the Search for Hardware area, then you can find Boards - QN9090DK6.

Figure 6: Search for the Hardware



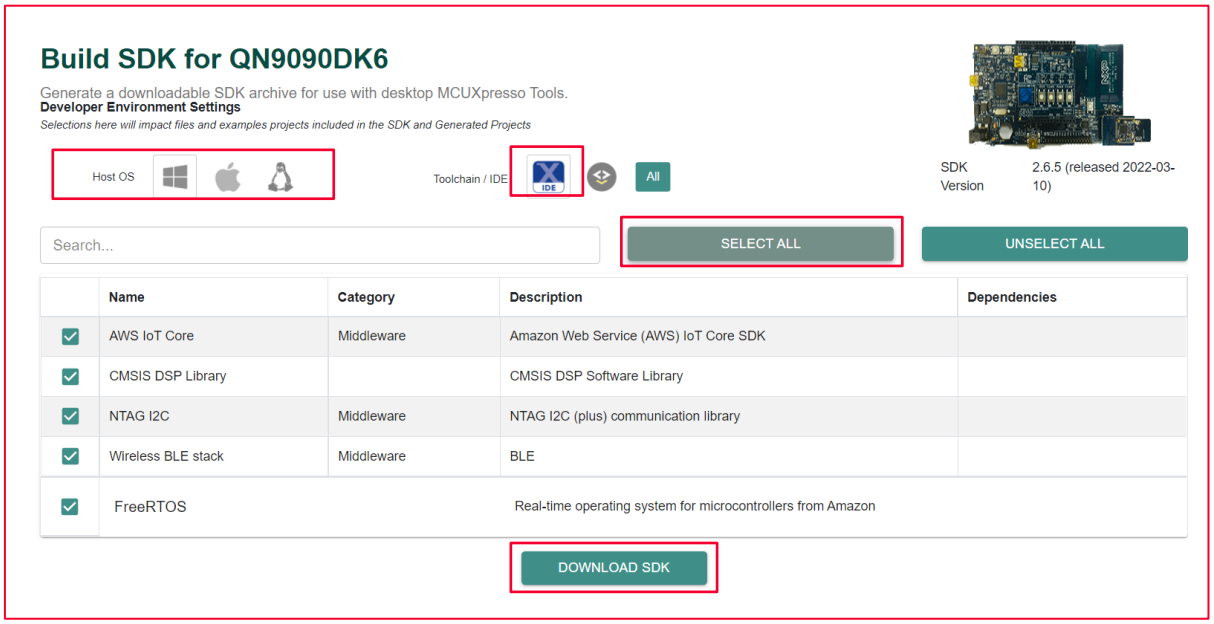
4. Build MCUXpresso SDK button should appear on the right-hand side. Click the button.

Figure 7: Build MCUXpresso SDK



5. Make sure the Host OS is correct and Toolchain is MCUXpresso IDE. Click **Select All** button, then click Download SDK button (**Figure 8**).

Figure 8: Download SDK for QN9090DK6

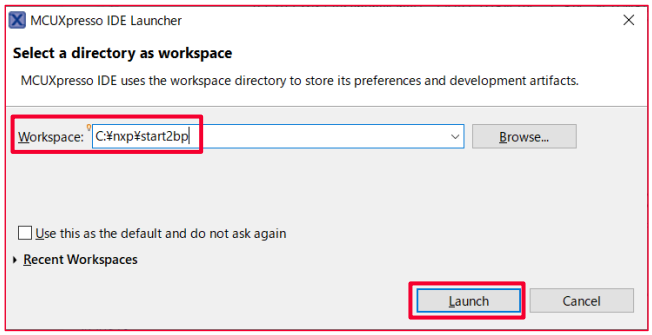


6. EULA may appear. Read carefully and follow the instructions.
7. Download will start automatically, and you will get SDK_x.x.x_QN9090DK6.zip.

4.6 Install QN9090DK SDK

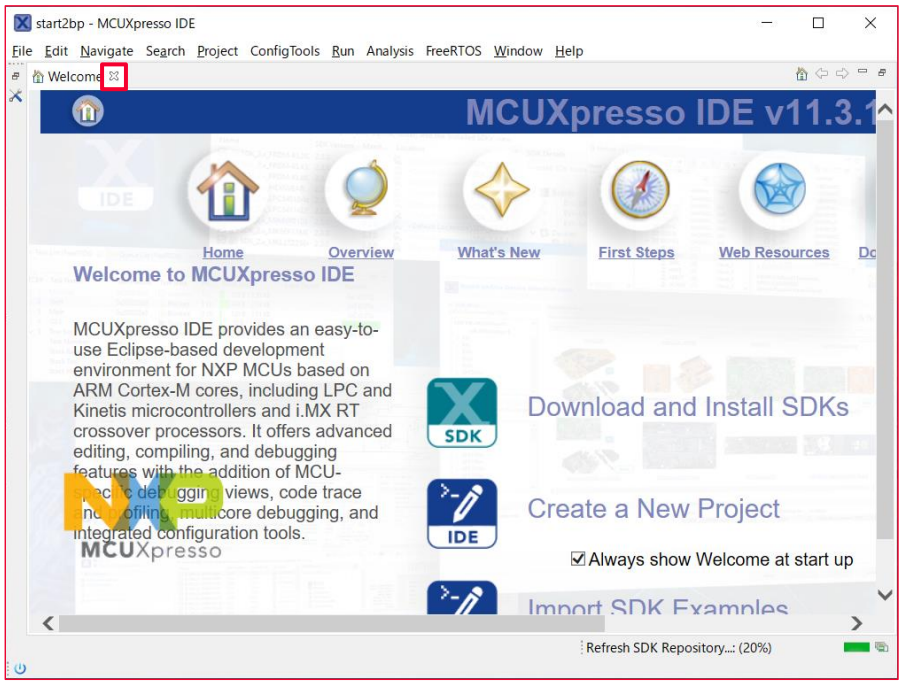
1. Launch MCUXpresso IDE.
2. Type Workspace directory as you want and click Launch button (C:\nxp\start2bp in this case) as shown in **Figure 9**.

Figure 9: Select Workspace Directory



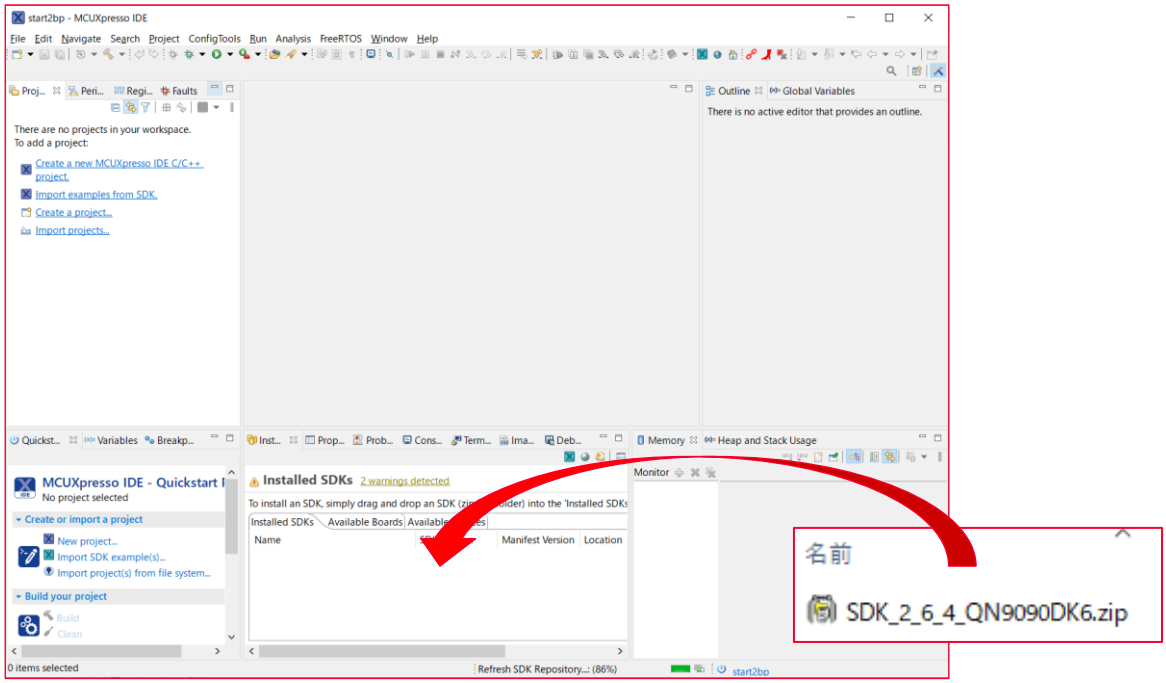
3. If you see the Welcome page, click "**X**" button on the Welcome tab (**Figure 10**).

Figure 10: MCUXpresso Welcome Page



4. Find SDK_x.x.x_QN9090DK6.zip on explorer and drag and drop the zip file into Installed SDKs tab (**Figure 11**). (No need to do this again if you see the SDK in the Installed SDKs window).

Figure 11: Install the SDK

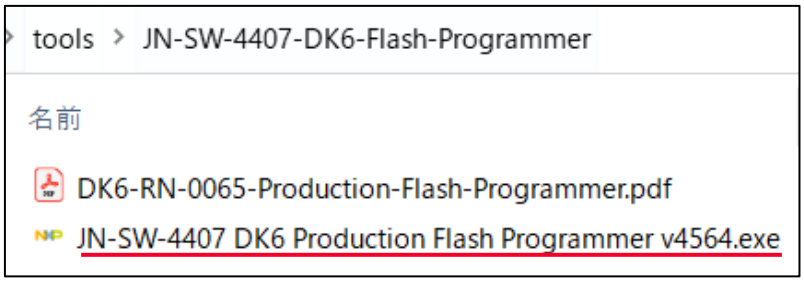


4.7 Install DK6Programmer

DK6Programmer supports EVK Rev3 and later.

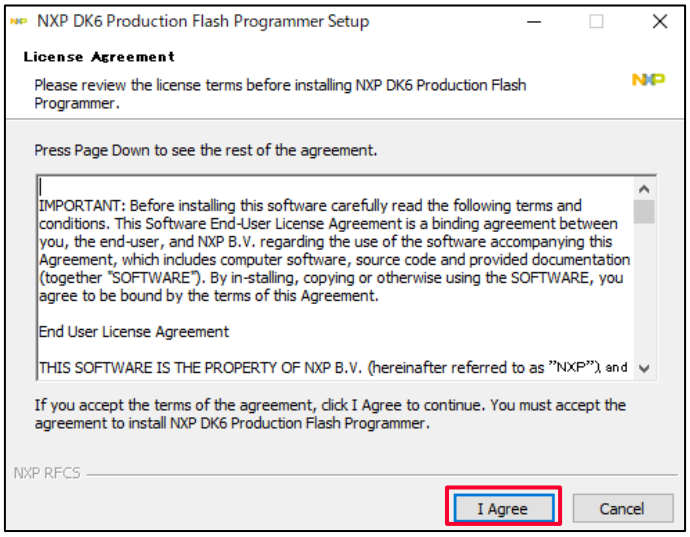
1. Unzip the SDK_x.x.x_QN9090DK6.zip file.
2. Run the installer (JN-SW-4407 DK6 Production Flash Programmer v4564.exe) in tools/JN-SW-4407-DK6-Flash-Programmer (**Figure 12**).

Figure 12: Run DK6Programmer Installer



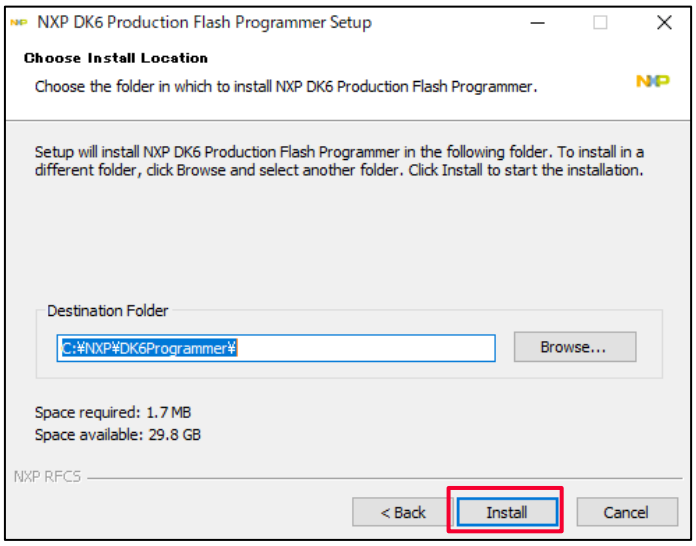
3. Click **I Agree** as shown in **Figure 13**.

Figure 13: License Agreement



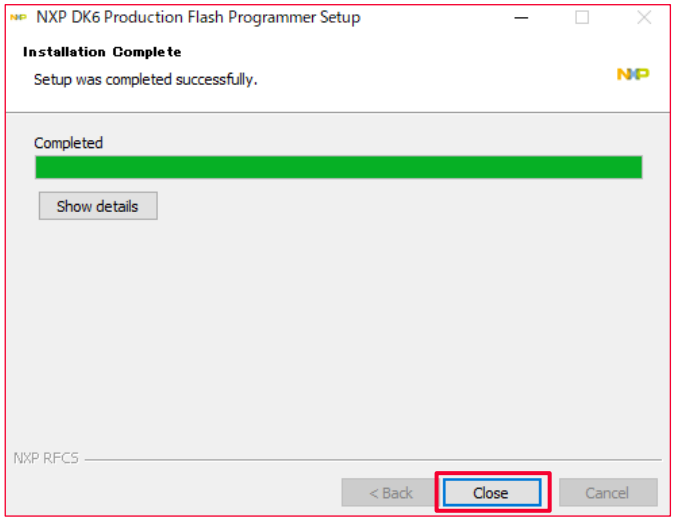
4. Click **Install** (**Figure 14**).

Figure 14: Install DK6Programmer



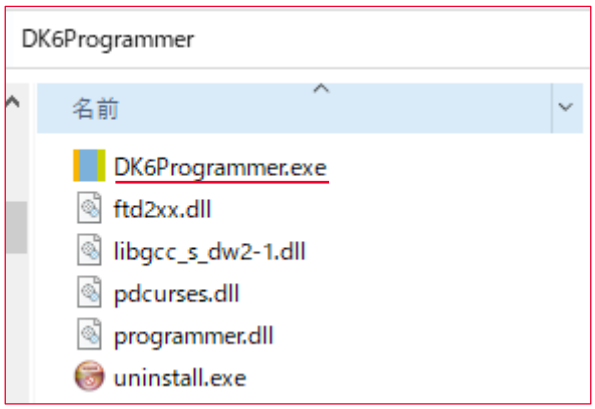
5. Installation is completed (**Figure 15**).

Figure 15: DK6Programmer Installation Complete



6. DK6Programmer application is installed (**Figure 16**).

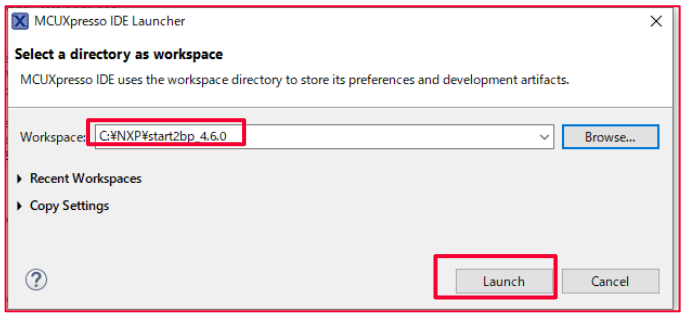
Figure 16: DK6Programmer Installed



5 Import Sample Project

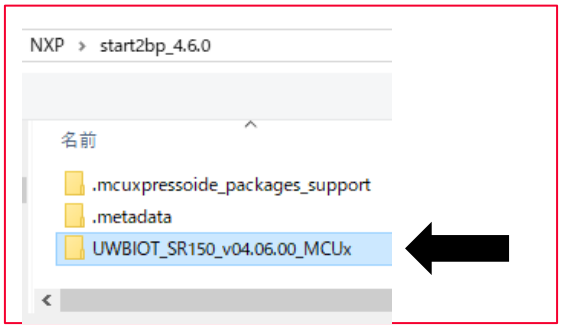
1. Launch MCUXpresso IDE.
2. Type Workspace directory as you want and click Launch button (C:\nxp\start2bp_4.6.0 in this case). The launch button is marked in **Figure 17**.

Figure 17: Launch MCUXpresso IDE



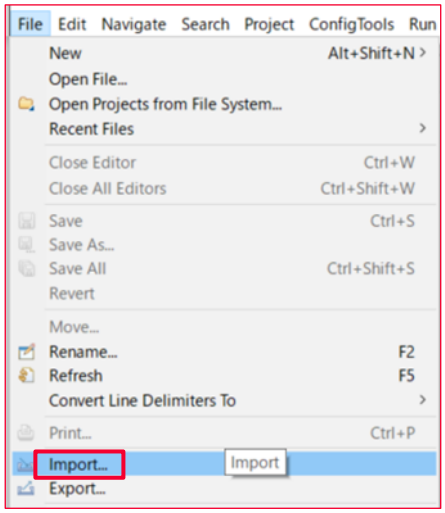
3. Unzip UWBIOT_SR150_v04.06.00_MCUx.zip into the directory you selected above (C:\nxp\start2bp_4.6.0 in this case).

Figure 18: Unzip the Project



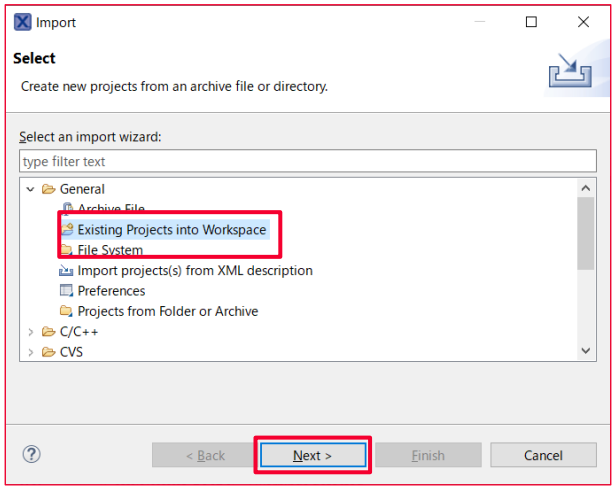
4. From menu, select File ➔ Import as marked in **Figure 19**.

Figure 19: Start Importing the Project



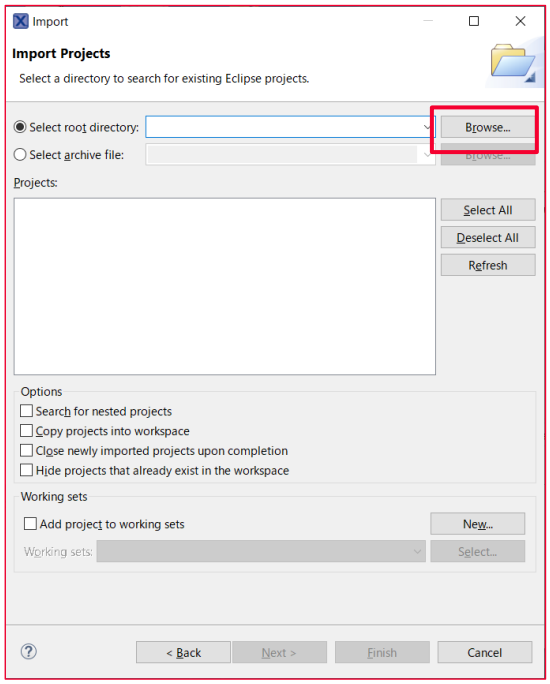
5. Select General ➔ Existing Projects into Workspace and click the **Next** button (**Figure 20**).

Figure 20: Select to Import Existing Project into Workspace



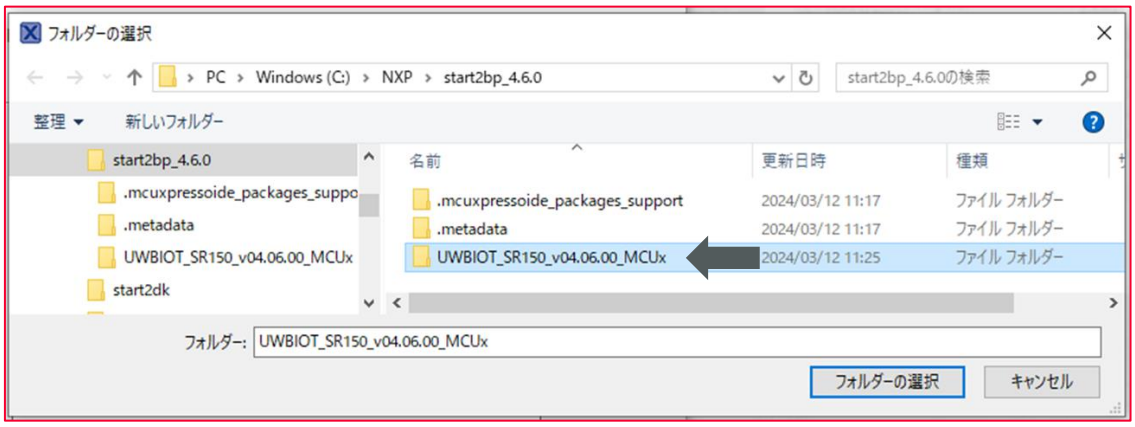
6. Click the **Browse …** button (**Figure 21**).

Figure 21: Start Browsing for the Project



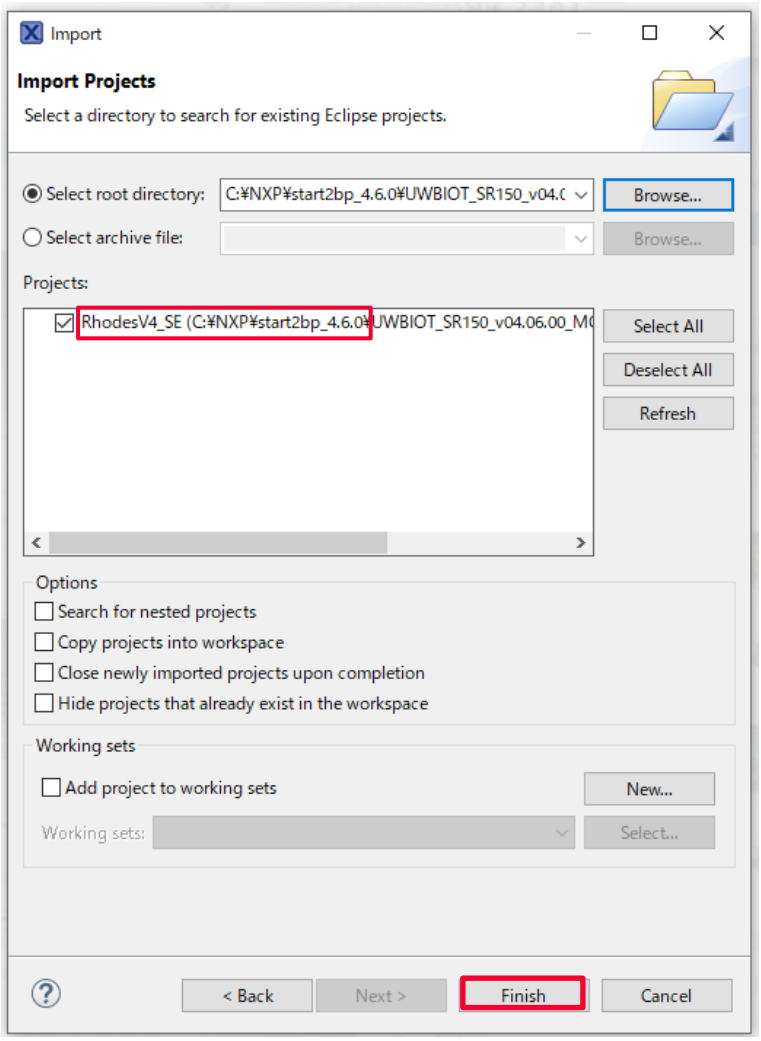
7. Select UWBIOT_SR150_v04.06.00_MCUx folder you unzipped in the previous step (**Figure 22**).

Figure 22: Locate the Project on Disk



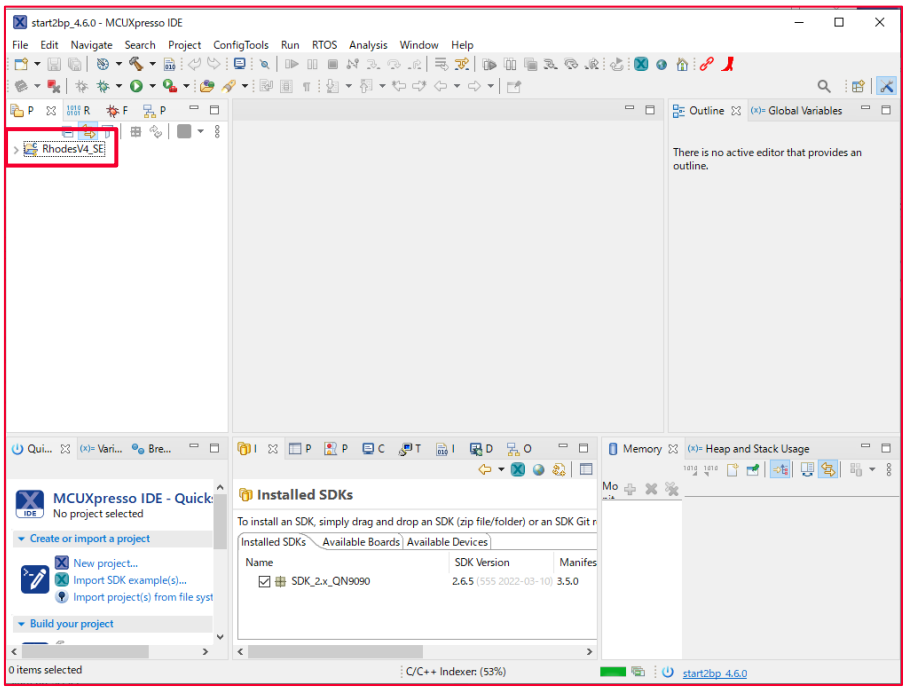
8. Make sure RhodesV4_SE is checked as shown in **Figure 23**. Click **the Finish** button.

Figure 23: Import Project - Finish



9. You will see imported projects in Project Explorer window (**Figure 24**).

Figure 24: Project Explorer Window



6 Applying Patch File

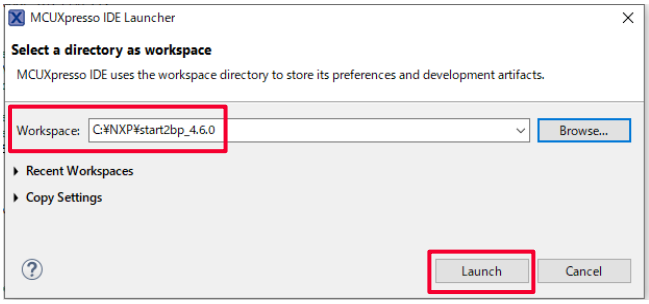
NXP SDKs are available on my Murata's SDK site. We also publish pre-built binary files for Type 2BP. We recommend applying the patch files to create this pre-built binary file. Please refer to the guide (How to make Pre-build binary) for details.

7 Build and Program Sample Application (demo_ranging_controller)

The SDK includes several sample applications. Select one sample application at a time.

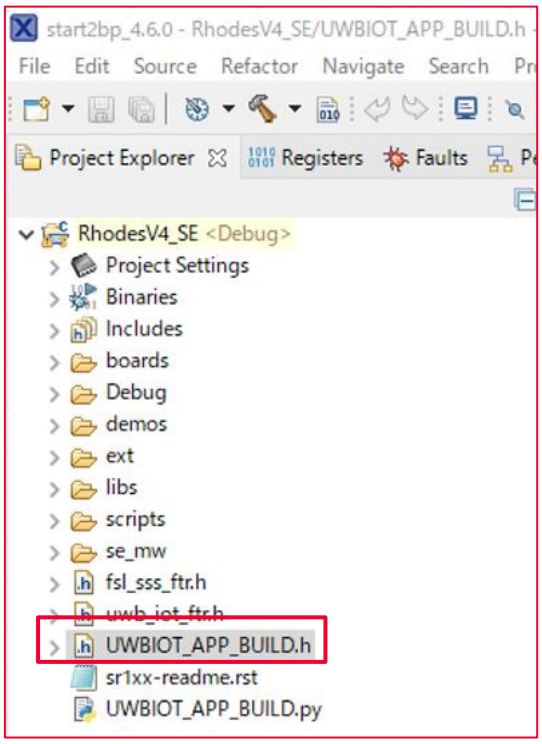
1. Launch MCUXpresso IDE.
2. Type a Workspace directory as you want and click Launch button (C:\nxp\start2bp_4.6.0 in this case) as shown in **Figure 25**.

Figure 25: Launch MCUXpresso IDE



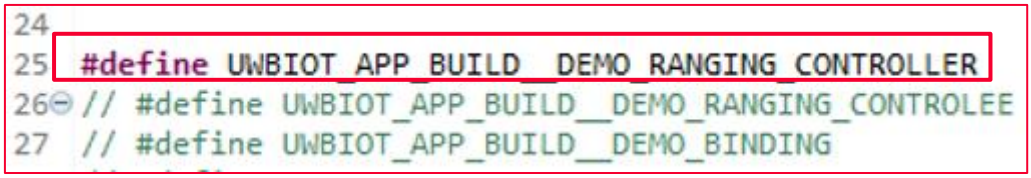
3. Expand RhodesV4_SE and double click UWBIOT_APP_BUILD.h as shown in **Figure 26**.

Figure 26: Select the UWBIOT_APP_BUILD.h File to Edit



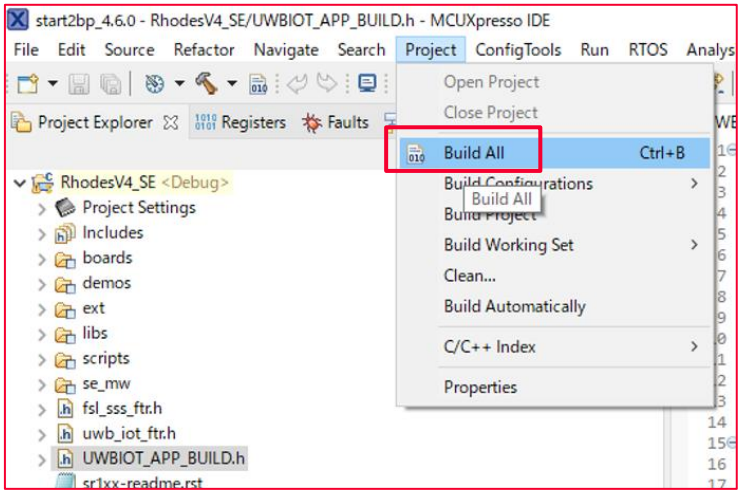
4. Comment out (disable define) UWBIOT_APP_BUILD__DEMO_BINDING and uncomment (enable define) UWBIOT_APP_BUILD__DEMO_RANGING_CONTROLLER as shown in **Figure 27**, then save it (Ctrl+S).

Figure 27: Edit the UWBIOT_APP_BUILD.h File



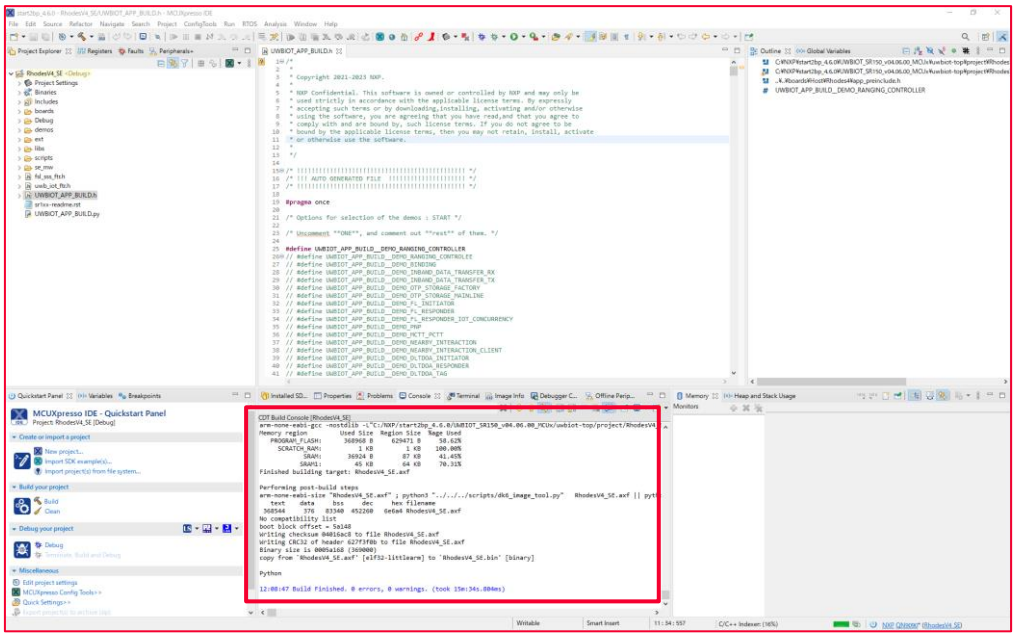
5. From the menu, select Project ➔ Build All (**Figure 28**).

Figure 28: Build the Application



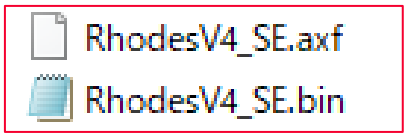
6. You can see build log in Console window. Wait for a while to see the Build Finished (**Figure 31**).

Figure 29: Build Finished



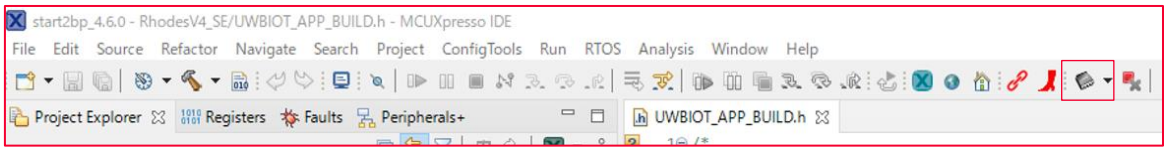
7. You can find binary images in uwbiot-top\project\RhodesV4_SE\Debug.

Figure 30: Built Binary Images



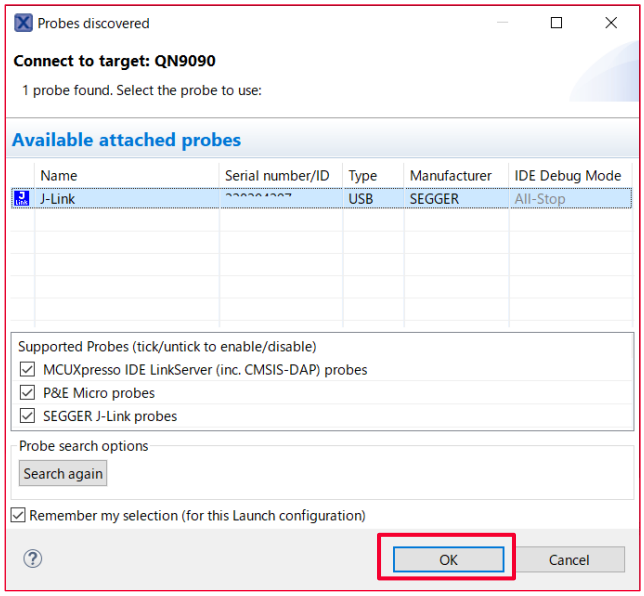
8. Use debuggers such as the MCU-Link case.
   (If you use DK6Programmer, please go to step 11)
   Click GUI Flash Tool button (**Figure 31**) to download the binary you just built into QN9090.

Figure 31: GUI Flash Tool Button



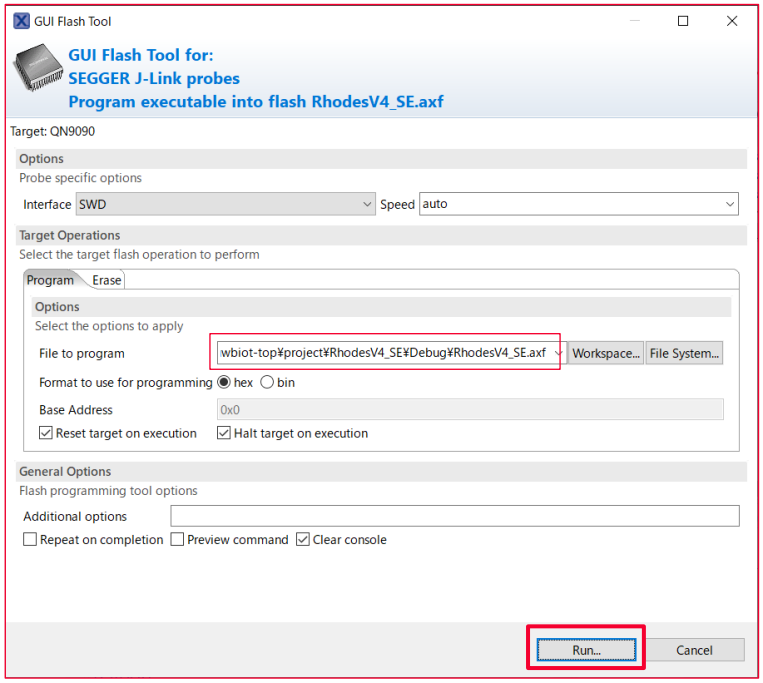
9. Confirm your SWD Adaptor and click the **OK** button (**Figure 32**).

Figure 32: Confirm Your SWD Adaptor



10. The file, RhodesV4_SE.axf, is automatically selected. Click the **Run**… button as marked in **Figure 33**.

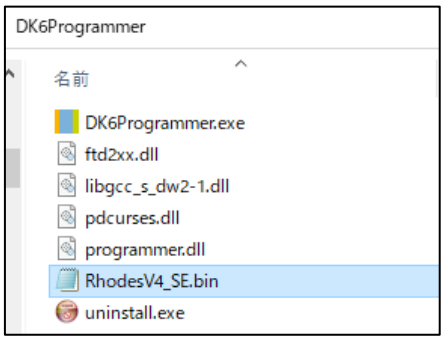
Figure 33: Start Flashing and Run



If the flash program is successful, please go to step 14.

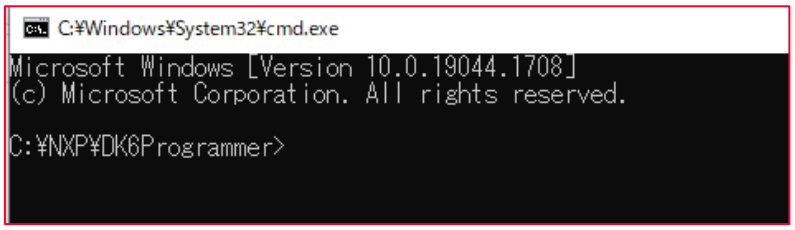
11. Use DK6Programmer case.
    Copy the bin file (uwbiot-top\project\RhodesV4_SE\Debug) generated by the build to the folder where the DK6Programmer was installed.

Figure 34: Copy the Binary File



12. Open the DK6Programmer path in Command Prompt.

Figure 35: Open Console and Go to Path



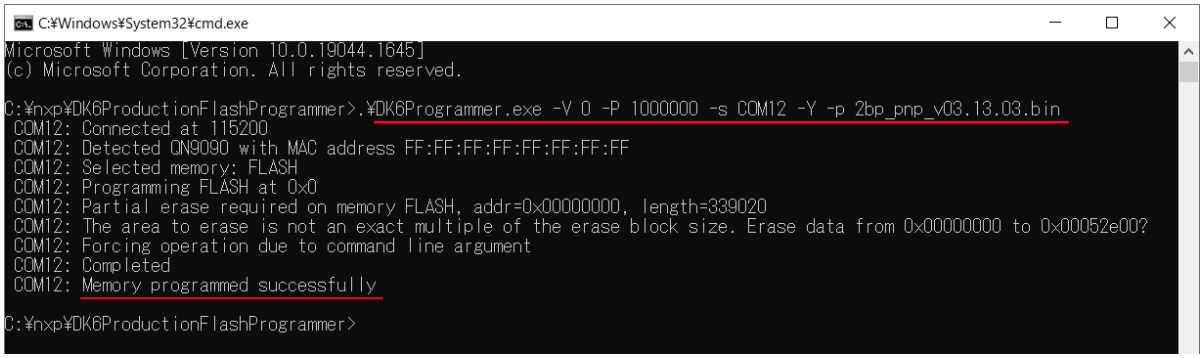
13. Execute the DK6Programmer application.
    Flashing command:

```
./DK6Programmer.exe -V 0 -P 1000000 -s COMx -Y -p xxxx.bin
```

> COM port number depends on your environment, please confirm on Device Manager.
> Please select the bin file name.

Example of execution is shown in **Figure 36**.

Figure 36: DK6Programmer Execution



14. Launch Tera Term and open the COM port. If you use SDK v04.02.01 or earlier, then the baud rate is 115200 bps. If you use SDK v04.04.03 or later then the baud rate is 3000000 bps. You will see log output on the terminal (**Figure 37**). With this build, appropriate calibration values are not loaded. Please refer to Applying Calibration Values Guide to apply calibration values.

Figure 37: Log Output Terminal

Table 3 shows the baud rates per SDK versions.

Table 2: Default Baud Rate Per SDK Versions

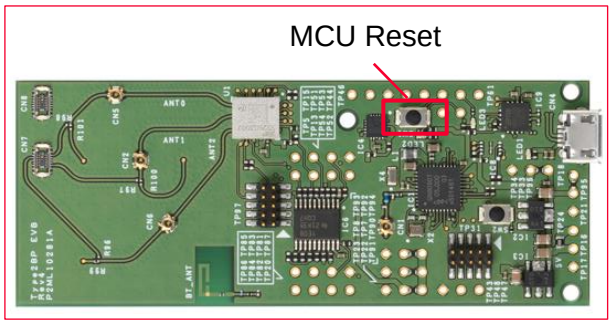
SDK Version	Default Baud Rate
v04.02.01 or earlier	115200 bps
v04.04.03 or later	3000000 bps

8 Trouble Shooting

8.1 Nothing Comes Out on Terminal (1)

Press **MCU Reset** button as marked in **Figure 38**.

Figure 38: MCU Reset Button



8.2 Nothing Comes Out on Terminal (2)

Make sure pycryptodome is installed on your PC. Try clean build from menu, Project➔Clean….

8.3 Cannot Click GUI Flash Tool Button

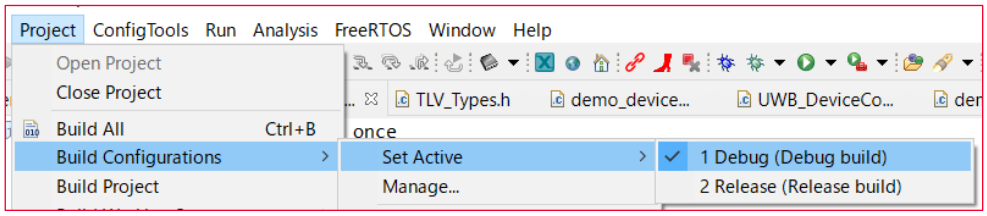
Sometimes, the GUI Flash Tool button is grayed out and you cannot click it even if you have completed building a demo application.

Click somewhere on Project Explorer window (for instance, RhodesV4_SE on top), then you should be able to click the button.

8.4 Debug Build and Release Build

You can select Debug build or Release build from menu Project➔ Build Configurations➔ Set Active as shown in **Figure 39**.

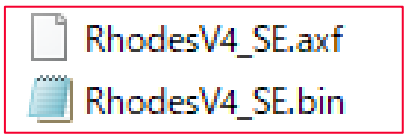
Figure 39: Build Configuration – Set Active



8.5 Binary Image Path

You can find binary images in uwbiot-top\project\RhodesV4_SE\Debug or uwbiot-top\project\RhodesV4_SE\Release depending on Build Configurations (**Figure 40**).

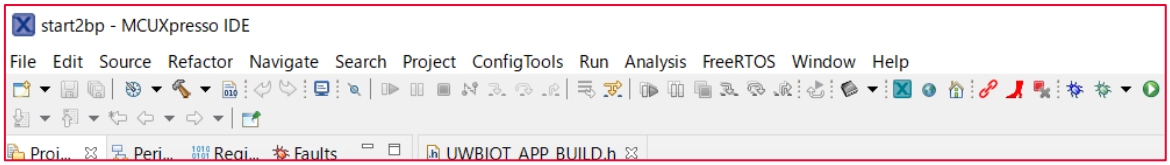
Figure 40: Binary Images



9 Debug Sample Application (demo_ranging_controller)

Click the **Debug** button and follow the instructions.

Figure 41: Debug Sample Application



10 Build Sample Application (demo_ranging_controlee)

To build demo_ranging_controlee, following changes are needed in UWBIOT_APP_BUILD.h. Other configurations are the same with demo_ranging_controller.

- Disable UWBIOT_APP_BUILD__DEMO_RANGING_CONTROLLER
- Enable UWBIOT_APP_BUILD__DEMO_RANGING_CONTROLEE

Figure 42: Build Sample Application

```
25  //#define UWBIOT_APP_BUILD__DEMO_RANGING_CONTROLLER
26  #define UWBIOT_APP_BUILD__DEMO_RANGING_CONTROLEE
```

If you run demo_ranging_controller and demo_ranging_controlee at the same time, you will be able to see the distance between the EVBs. With this build, appropriate calibration values are not loaded. Please refer to Applying Calibration Values Guide guide to apply calibration values.

Figure 43 shows the ranging result between the EVBs.

Figure 43: Ranging Distance Between EVBs

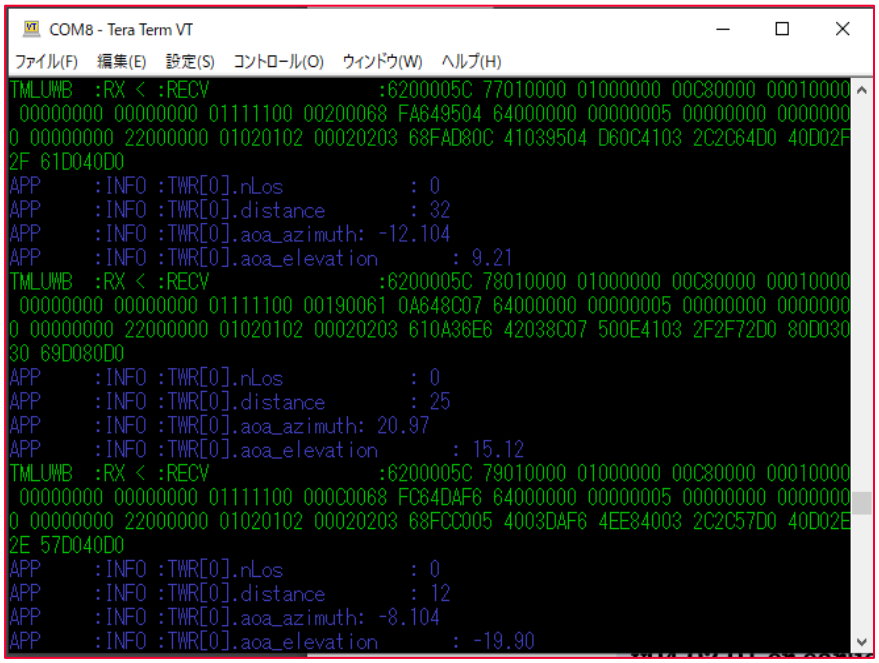


Table 3 shows the baud rates per SDK versions.

Table 3: Default Baud Rate Per SDK Versions

SDK Version	Default Baud Rate
v04.02.01 or earlier	115200 bps
v04.04.03 or later	3000000 bps

11 Build Sample Application (pnp)

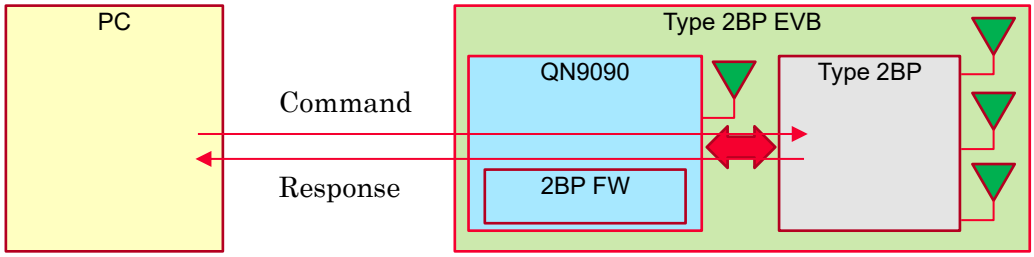
Demo_ranging_controller and controlee are called Standalone mode; demo application automatically starts running as programmed.

There is another type of demo application called PnP (Plug and Play) mode which means when you turn on the EVB, PnP mode application waits for commands from PC.

QN9090 MCU on the EVB is the host processor for Type 2BP. In PnP mode, QN9090 will pass through commands from PC to Type 2BP and response from Type 2BP to PC.

The sample application block diagram is shown in **Figure 44**.

Figure 44: Sample Application Block Diagram

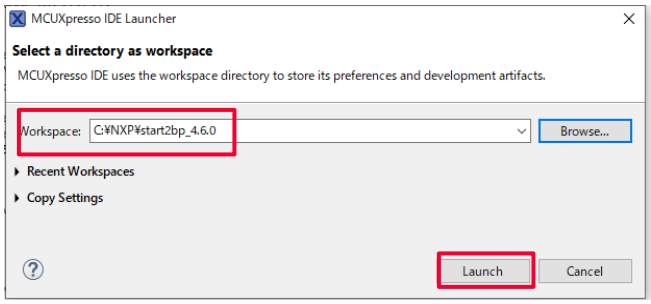


QN9090 has flash memory and can store Type 2BP firmware. When you turn on the EVB, QN9090 can automatically download the firmware into Type 2BP.

Here are the steps to build PnP mode application.

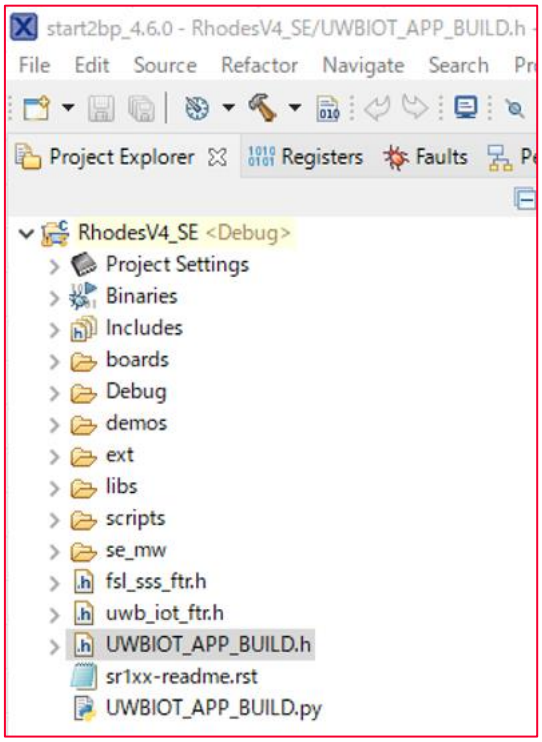
1. Launch MCUXpresso IDE.
2. Type Workspace directory as you want and click Launch button (C:\nxp\start2bp in this case). Refer to **Figure 45**.

Figure 45: Launch MCUXpresso IDE



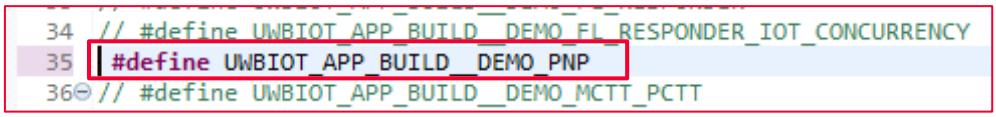
3. Expand RhodesV4_SE and double click UWBIOT_APP_BUILD.h (**Figure 46**).

Figure 46: Select the UWBIOT_APP_BUILD.h File to Edit



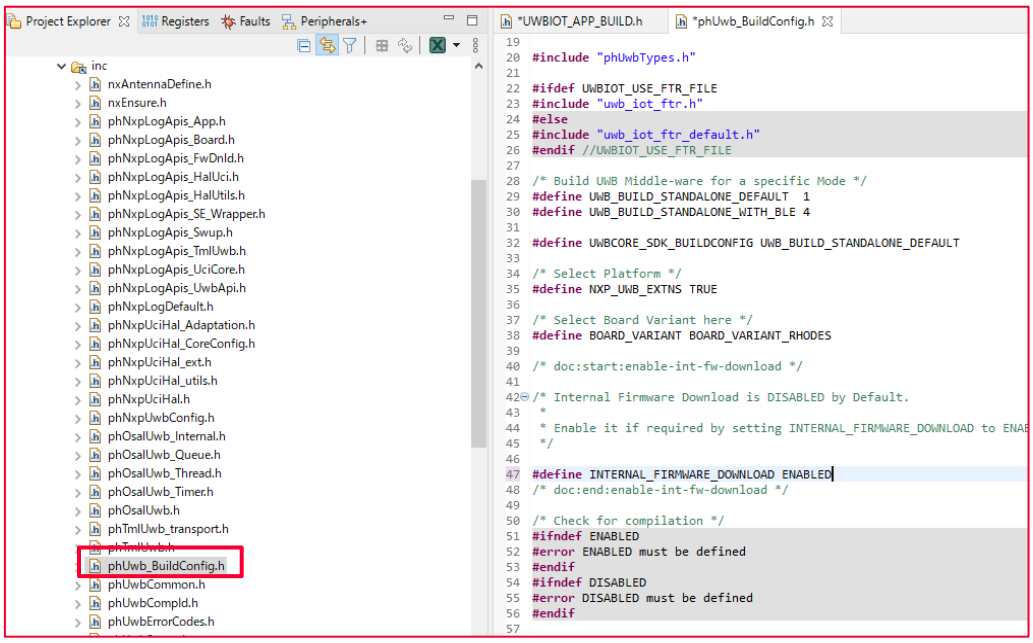
4. Uncomment (enable define) UWBIOT_APP_BUILD__DEMO_PNP and comment out (disable define) all of others as shown in **Figure 47**, then save it (Ctrl+S).

Figure 47: Edit the UWBIOT_APP_BUILD.h File



5. Expand libs – halimp – inc and double click phUwb_BuildConfig.h **Figure 48**.

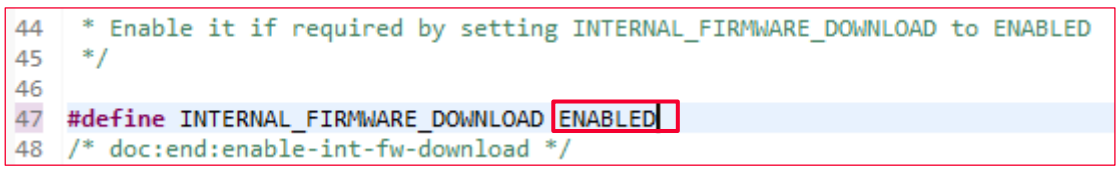
Figure 48: Select the phUwb_BuildConfig.h File to Edit



6. Find INTERNAL_FIRMWARE_DOWNLOAD and modify from DISABLED to ENABLED as shown in **Figure 49**.

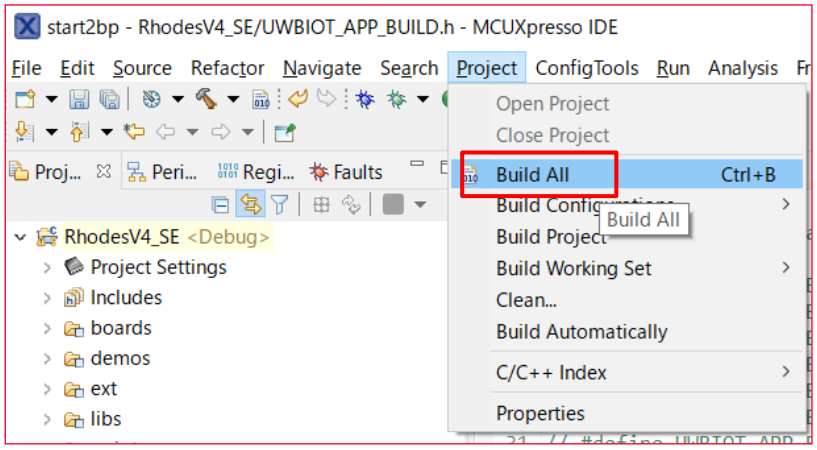
This configuration only affects the PnP mode and is not needed for other demos.

Figure 49: Edit the phUwb_BuildConfig.h File



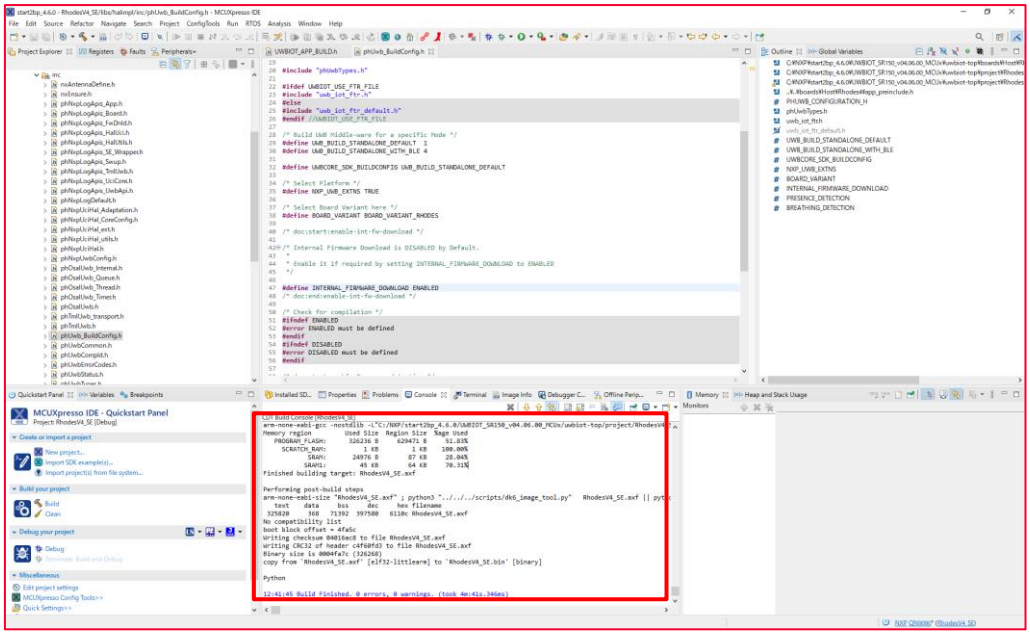
7. From menu, select Project➔Build All (**Figure 50)**.

Figure 50: Build Application



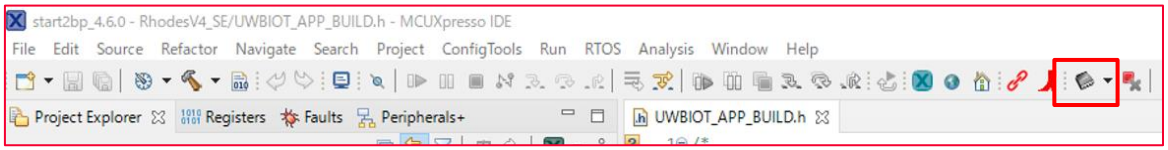
8. You can see build log in Console window. Wait for a while to see Build Finished (**Figure 51**).

Figure 51: Build Finished



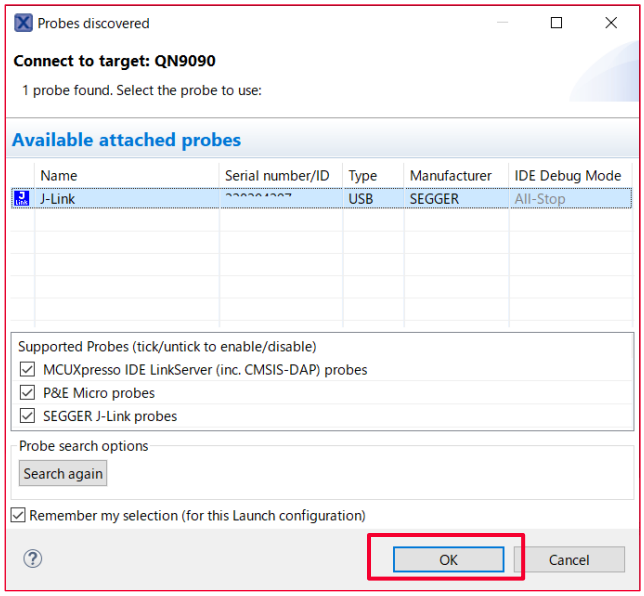
9. Click GUI Flash Tool button as marked in **Figure 52** to download the binary you just built into QN9090.

Figure 52: GUI Tool Button



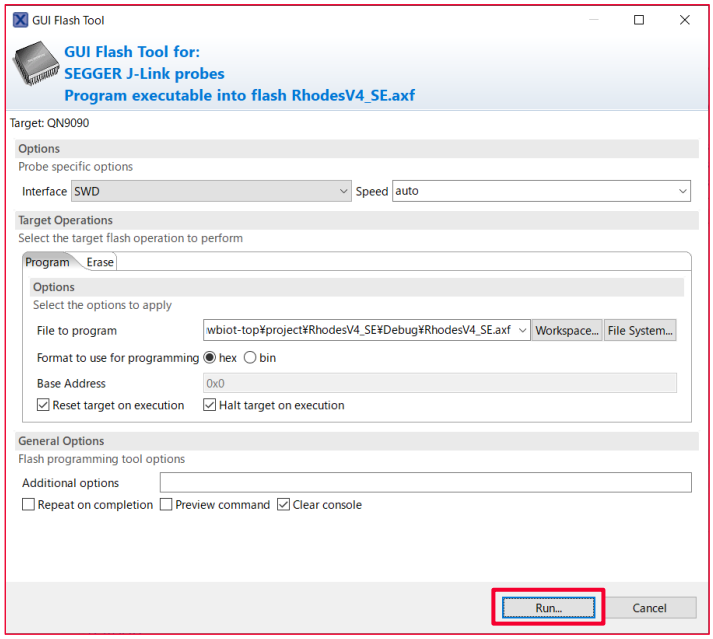
10. Confirm your SWD Adaptor and click the **OK** button (**Figure 53**).

Figure 53: Confirm SWD Adaptor



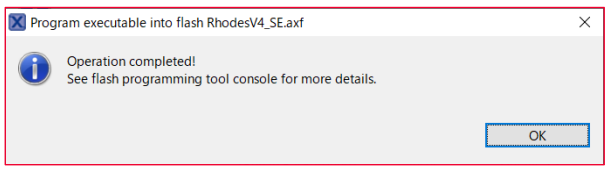
11. Click **Run…** button (**Figure 54**).

Figure 54: Run… Button



12. Wait for a while and you will see Operation completed window as shown in **Figure 55**.

Figure 55: Operation Completed



13. Your EVB is ready to interact in PnP mode. Please refer to the PnP Test Guide.

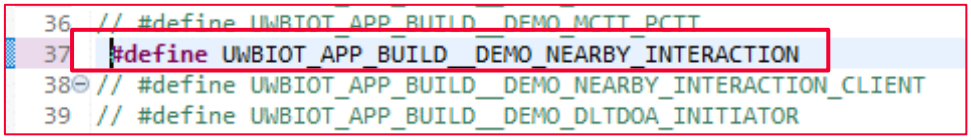
12 Build Sample Application (Nearby Interaction with Apple U1)

Apple announced Nearby Interaction Framework to utilize U1 chip and Implementing Spatial Interactions with Third-Party Accessories sample code for iPhone11 or later with U1 chip.

To build demo application for Apple U1, the following changes are needed in UWBIOT_APP_BUILD.h.

- Enable UWBIOT_APP_BUILD__DEMO_UWB_BLE_SR150I
- Disable all other defines.

Figure 56: Edit the UWBIOT_APP_BUILD.h File



After building and downloading the demo application, you will see the following output shown in **Figure 57**.

Figure 57: Output After Building and Downloading Demo Application

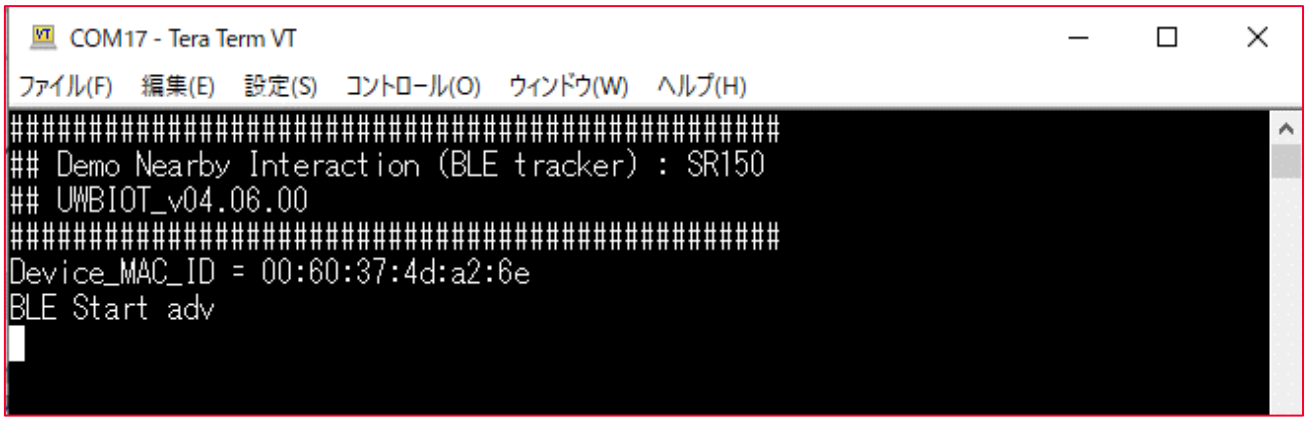


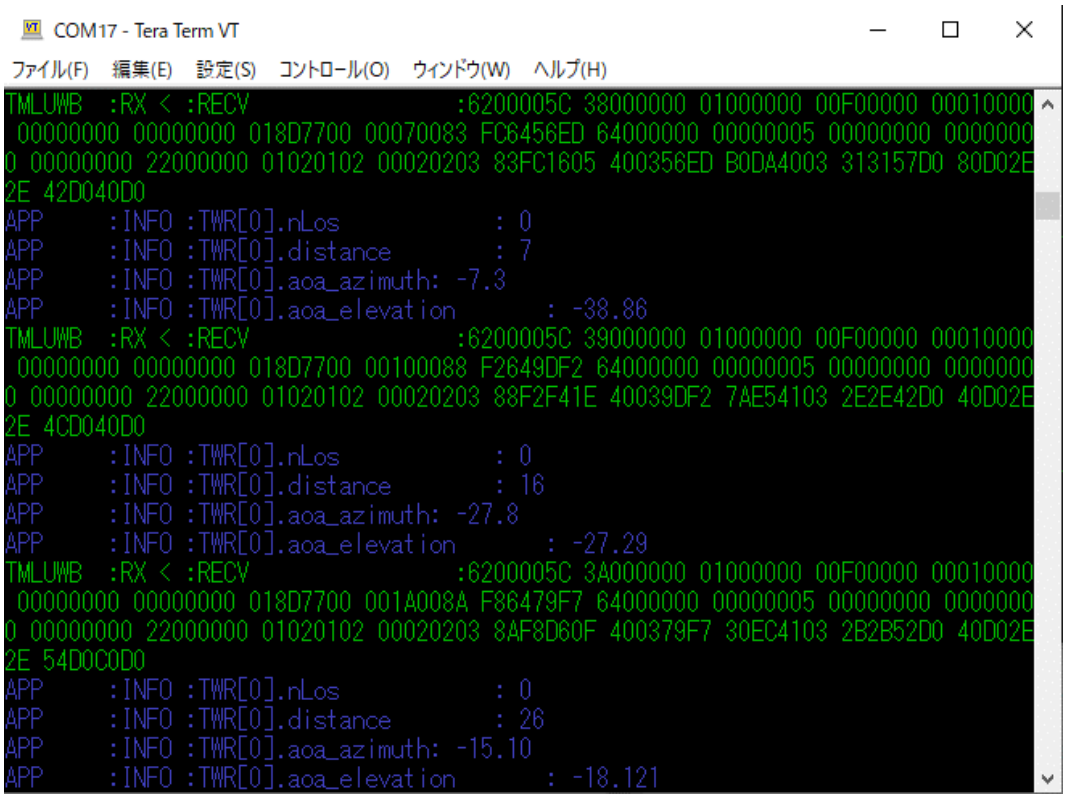
Table 4 describes the default baud rate per SDK version.

Table 4: Default Baud Rate Per SDK Versions

SDK Version	Default Baud Rate
v04.02.01 or earlier	115200 bps
v04.04.03 or later	3000000 bps

After establishing the connection, you will see the output a shown in **Figure 58**. With this build, appropriate calibration values are not loaded. Please refer to Applying Calibration Values Guide to apply calibration values.

Figure 58: Output After Establishing Connection



13 Appendix A - Flash/SRAM Memory Usage for Each of the Demos

The following table shows the Flash/SRAM memory usage for some demos in v04.06.00. By default, the compilation options are set to "-O1" for Debug and "-Os" for Release. See the MCUXPresso IDE documentation for more information on how to check out the options.

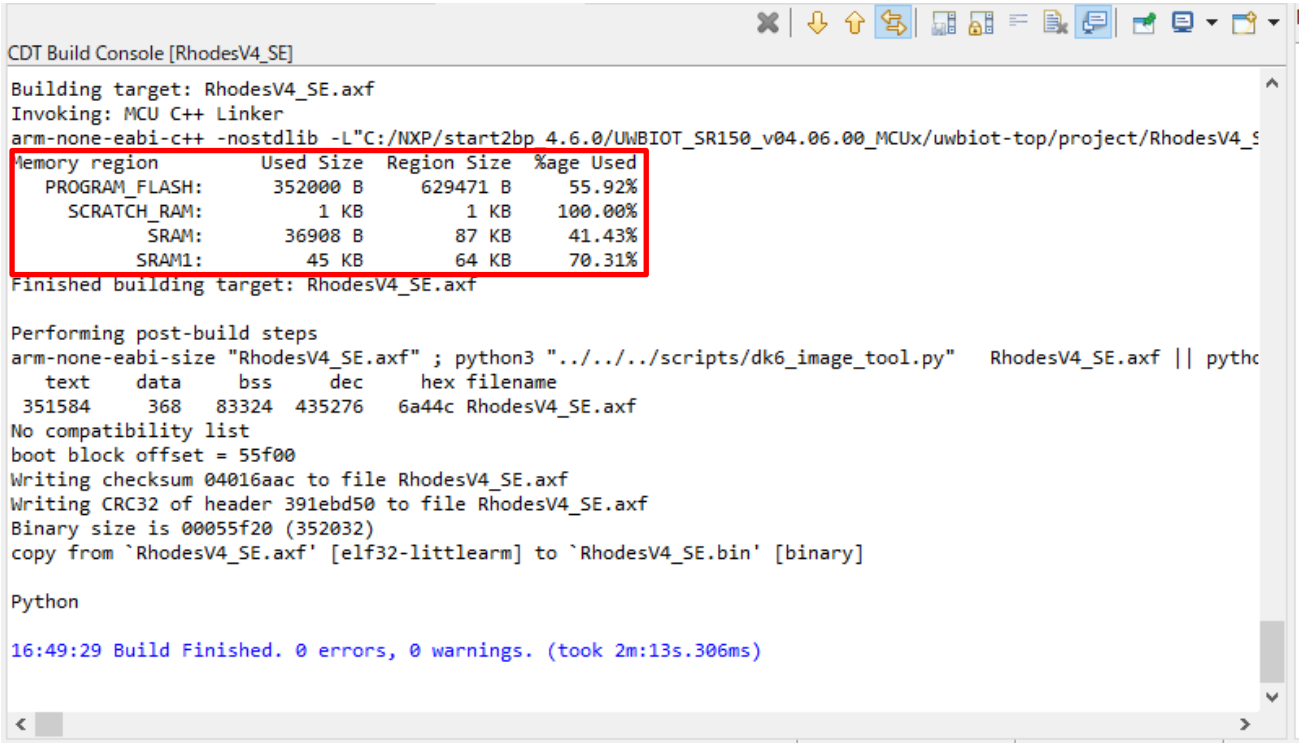
Table 5: Flash/SRAM Memory Usage for Demos in v04.06.00

SR150 v04.06.00	Flash Used Size		SRAM Used Size	
Demo	**Debug**	**Release**	**Debug**	**Release**
DEMO_RANGING_CONTROLLER	369 kB	352 kB	37 kB	37 kB
DEMO_RANGING_CONTROLEE	369 kB	352 kB	37 kB	37 kB
DEMO_PNP	326 kB	312 kB	25 kB	25 kB
DEMO_NEARBY_INTERACTION	545 kB	524 kB	82 kB	81 kB

If you want to know the memory usage of other demo, it will be displayed in the Build Console when the demo application is successfully built as shown in

Figure *59*.

Figure 59: Confirmation of Flash/SRAM Memory Usage



14 Appendix B – How to Change the Baud Rate

Starting with SDK v04.04.03 or later for SR150, the Baud rate for UART has been changed from 115200bps to 3Mbps. The default setting is 3Mbps, but if you want to change to 115200bps, change the settings for standalone mode and PnP mode.

14.1 Standalone Mode

Change the “BOARD_DEBUG_UART_BAUDRATE” in boards/Host/Rhodes4_SPI/board.h as shown in **Figure 60**

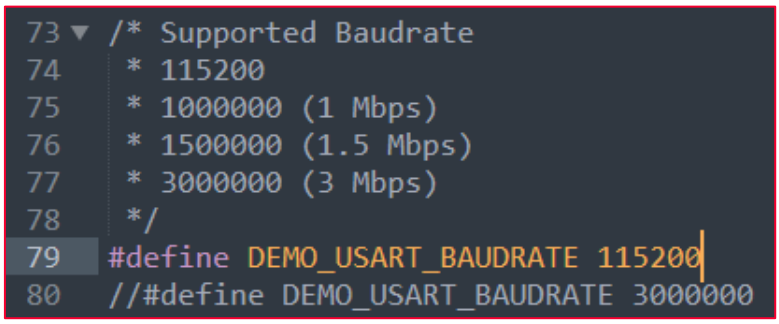
Figure 60: Change BOARD_DEBUG_UART_BAUDRATE

```
245  /* doc-start:uart_logging */
246  /* The UART to use for debug messages. */
247  /* when UWBIOT_LOG=Verbose, set the baudrate to 3Mbps for viewing logs in terminal emulator(ex. Tera term)*/
248  #define BOARD_DEBUG_UART_TYPE kSerialPort_Uart
249  #if (UWBIOT_LOG_VERBOSE == 1)
250  #define BOARD_DEBUG_UART_BAUDRATE 115200
251  //#define BOARD_DEBUG_UART_BAUDRATE 3000000U //3Mbps
252  #pragma message("Logging Baud rate is set to 3Mbits. Please change settings on Serial port GUI")
253  #else
254  #define BOARD_DEBUG_UART_BAUDRATE 115200
255  //#define BOARD_DEBUG_UART_BAUDRATE 3000000U
256  #endif
257  /* doc-end:uart_logging */
258  #endif
```

14.2 PnP Mode

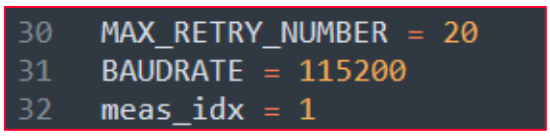
1. Change “DEMO_USART_BAUDRATE” in demos/pnp/Rhodes4/pnp_usart.c

Figure 61: Change DEMO_USART_BAUDRATE



2. Change “serial_port.baudrate” in MTD_SCP_102_A_DS_TWR_SR150_UART_interface_v040600.py.

Figure 62: Change serial_port.baudrate



Revision History

Revision	Date	Author	Change Description
2.1	Jul 19, 2021		Initial
3.0	Jul 30, 2021		Updated for EVB Rev 3.0 design
3.0A	Sep 10, 2021		• Updated for SDK v3.09.00. 1. QN9090 SDK is replaced from QN9090DK6 to QN9090 2. Importing project is RhodesV4_SE 3. Demo application selection is in header file 4. Configuration option for SE in Project Property • Removed 8.2. Nothing comes out on Terminal • Added 8.3. Cannot click GUI Flash Tool button • Added 10. Build Sample Application (pnp)
3.0B	Mar 15, 2022		• Updated for SDK v03.13.03 • Removed disabling SE step in Section 6. Build and Program Sample Application (demo_ranging_controller) • Added 12. Build Sample Application (Nearby Interaction with Apple U1)
3.0C	Mar 24, 2022		• Added 8.4 Debug Build and Release Build. • Added 8.5. Binary Image Path. • Corrected define in 12. Build Sample Application (Nearby Interaction with Apple U1).
4.0	Aug 10, 2022		Updated for EVB Rev 4.0 design
4.0A	Oct 12, 2022		• Added 4.7 Install DK6Programmer • Added 6.11 usage for DK6Programer
4.0B	Dec 6, 2022		Updated for SDK v04.02.01
4.0C	Dec 6, 2022		Revised 1. Prerequisites
4.0D	Jan 12, 2023		• Updated for SWD information. • Added 6 Applying Patch file
4.1	Jun 01, 2023		• Updated for EVB Rev 4.1 design • Updated for SDK v04.04.03
4.2	Sep 13, 2023		Added Appendix A. Flash/SRAM Memory Usage for Each of the Demos.
4.3	Oct 12, 2023		Added Appendix B. How to change the Baud rate
4.4	Nov 10, 2023		Update for debugger information
4.5	Mar 18, 2024		Updated for SDK v04.06.00 Document format changed