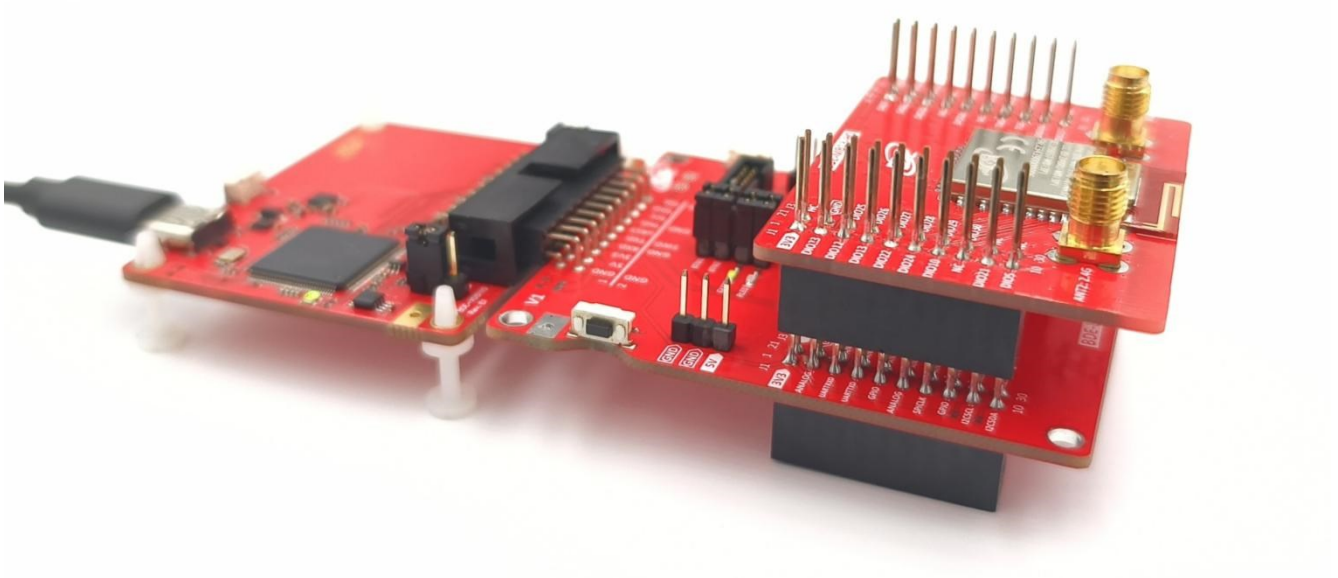
BDE-MB1354P101 Module User Guide

V1.0



Contents

1. General Description

This user guide is intended to assist the users in evaluating the module by providing instructions on how to effectively use the evaluation kit.

The BDE-MB1354P101 is a multi-band Sub-1GHz and 2.4GHz wireless module supporting Thread, Zigbee®, Bluetooth® 5.2 low Energy, IEEE 802.15.4g, IPv6-enabled smart objects (6LoWPAN), mioty, Wi-SUN, Amazon Sidewalk, proprietary systems including TI 15.4-Stack (Sub-1GHz and 2.4GHz), and concurrent multiprotocol through a Dynamic Multiprotocol Manager (DMM) driver.

The kit includes a breakout board with module (board number: BDE-MB13-BO), a Launchpad™ evaluation module which can connect with the breakout board through the BoosterPack™ interface (board number: BDE-LPEM), and also a XDS110 debugger used for debugging BDE-MB1354P101 device (board number: BDE-XDS110).

2. Launchpad Evaluation Module: BDE-LPEM

The BDE-LPEM is an evaluation module designed as an interface module with BoosterPack interface which can connect with any boards that come with the standard BoosterPack interface, including the BDE-MB13-BO board. For more details on the BoosterPack pinout standard, please refer to this link.

The board overview is shown as Figure 1.

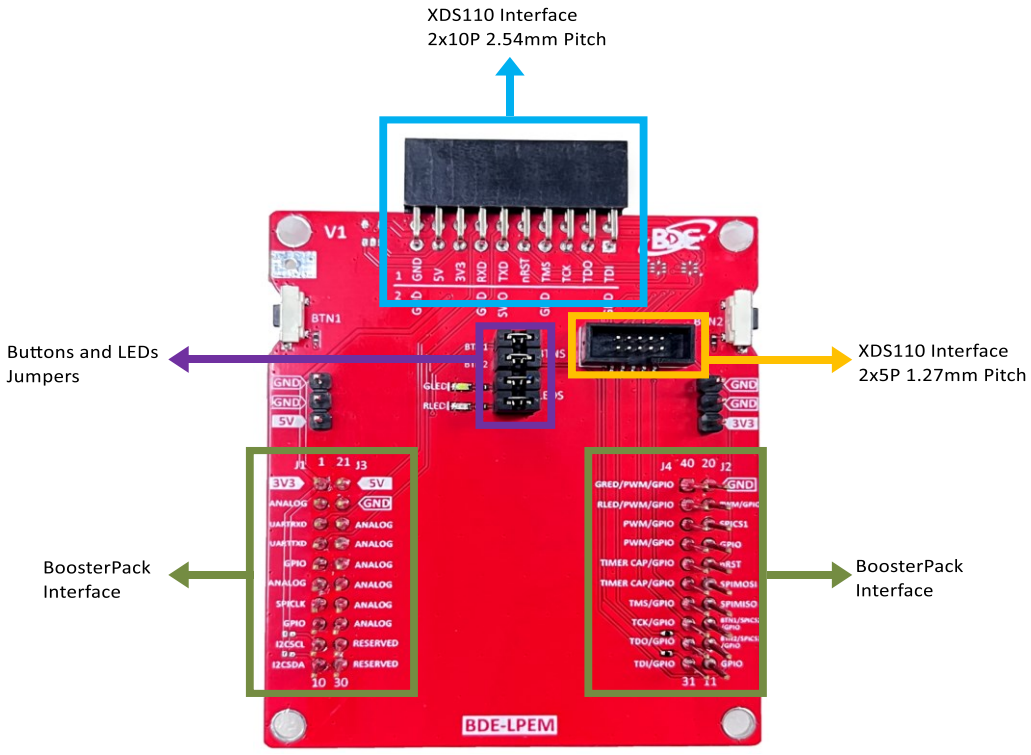


Figure 1. Board Overview of BDE-LPEM

3. Breakout Board: BDE-MB13-BO

We provide a breakout board for module to enable easy access to every exposed pad on module with the 2.54mm-pitch pins. The breakout board can be stacked to the BDE-LPEM board for evaluation.

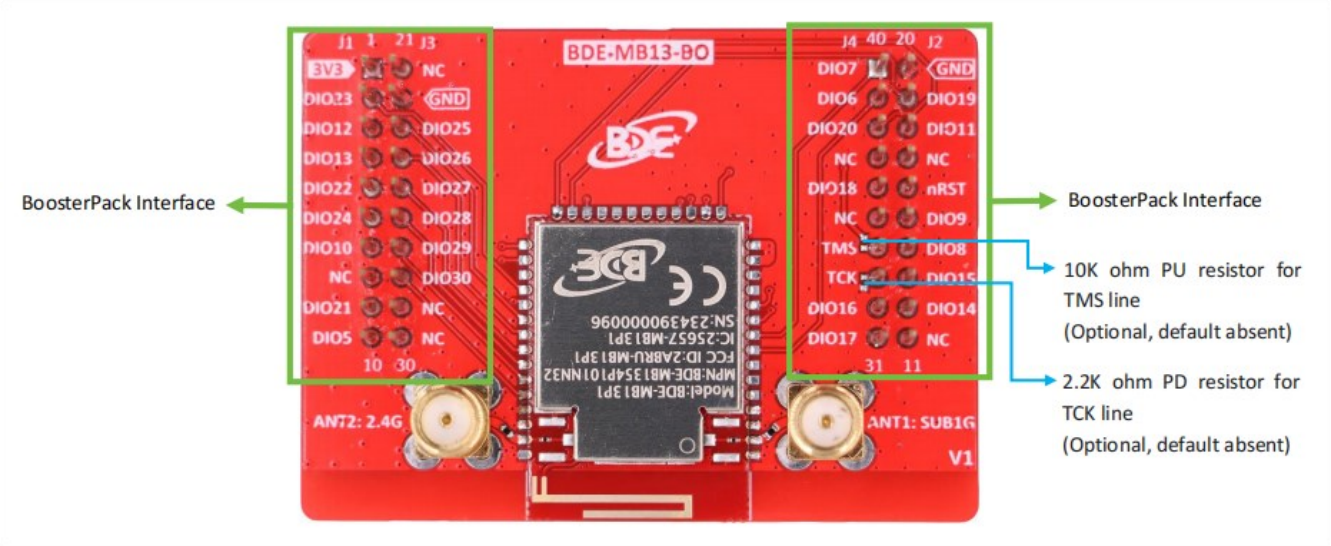


Figure 2. Board Overview of BDE-MB13-BO

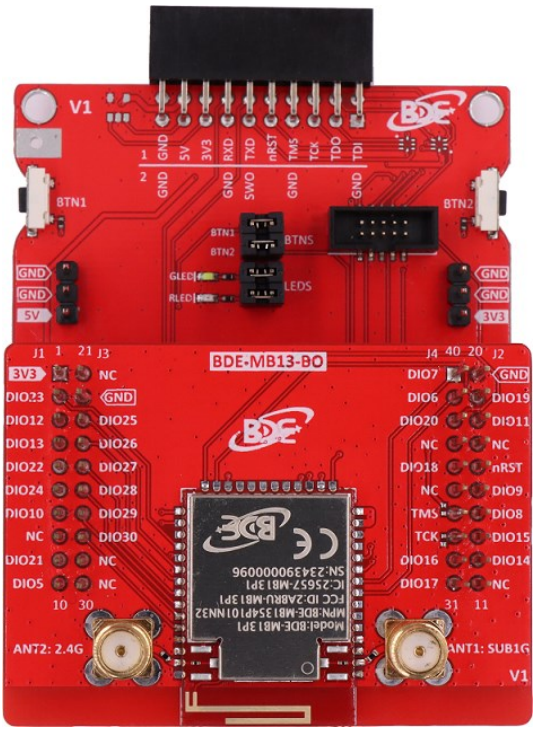


Figure 3. The Photo of BDE-LPEM Stacked with the Breakout Board with Module

3.1. Breakout Board Header Pin Assignment

Table 1 shows the pin assignment for breakout board header J1, J2, J3 and J4.

Table 1. Breakout Board Headers Pin Assignment

Pin Number	Pin Name	Type/Direction	Description
1	3V3	Power	Board power supply, connect with module VDD through a ferrite bead
2	DIO23	I/O	GPIO
3	DIO12	I/O	GPIO, module RX in BoosterPack pin
4	DIO13	I/O	GPIO, module TX in BoosterPack pin
5	DIO22	I/O	GPIO
6	DIO24	I/O	GPIO
7	DIO10	I/O	GPIO, SPI_CLK in BoosterPack pin
8	NC	-	No connect
9	DIO21	I/O	GPIO, I2C_SCL in BoosterPack pin
10	DIO5	I/O	GPIO, I2C_SDA in BoosterPack pin
11	NC	-	No connect
12	DIO14	I/O	GPIO, Button 2 in BoosterPack pin
13	DIO15	I/O	GPIO, Button 1 in BoosterPack pin
14	DIO8	I/O	GPIO, SPI_MISO in BoosterPack pin
15	DIO9	I/O	GPIO, SPI_MOSI in BoosterPack pin
16	nRST	Input	Reset pin, active low
17	NC	-	No connect
18	DIO11	I/O	GPIO, SPI_CS in BoosterPack pin
19	DIO19	I/O	GPIO
20	GND	GND	Board ground
21	NC	-	No connect
22	GND	GND	Board ground
23	DIO25	I/O	GPIO
24	DIO26	I/O	GPIO
25	DIO27	I/O	GPIO
26	DIO28	-	No connect
27	DIO29	-	No connect
28	DIO30	I/O	GPIO
29	NC	-	No connect
30	NC	-	No connect
31	DIO17	I/O	GPIO
32	DIO16	I/O	GPIO
33	TCK	-	TCK for JTAG/XDS110
34	TMS	-	TMS for JTAG/XDS110
35	NC	-	No connect
36	DIO18	I/O	GPIO
37	NC	-	No connect
38	DIO20	I/O	GPIO, PWM/GPIO in BoosterPack pin
39	DIO6	I/O	GPIO, RED LED in BoosterPack pin
40	DIO7	I/O	GPIO, Green LED in BoosterPack pin

3.2. Debug Header Pin Assignment

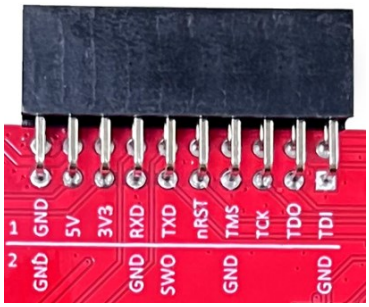


Figure 4. Debug Header

Table 2 shows the pin assignment for the debug header.

Table 2. Debug Header Pin Assignment

Pin Number	Pin Name	Type/Direction	Description
1, 2, 8, 14, 20	GND	GND	Board ground
3	5V	Power	5V supply from XDS110
4, 6, 12, 16, 18	NC	-	No connect
5	3V3	Power	Module power supply
7	RXD	Input	Module UART RXD
9	TXD	Output	Module UART TXD
10	SWO	I/O	SWO for JTAG
11	nRST	Input	Reset pin, active low
13	TMS	I/O	TMS for JTAG/XDS110, or SWDIO for SWD debug
15	TCK	I/O	TCK for JTAG/XDS110, or SWDCK for SWD debug
17	TDO	I/O	TDO for JTAG/XDS110
19	TDI	I/O	TDI for JTAG/XDS110

BDE recommends using XDS110 for the debug. J-Link or other debuggers support SWD could also work.

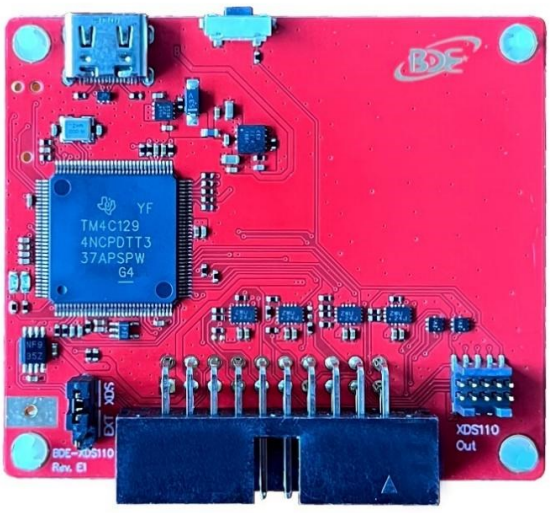
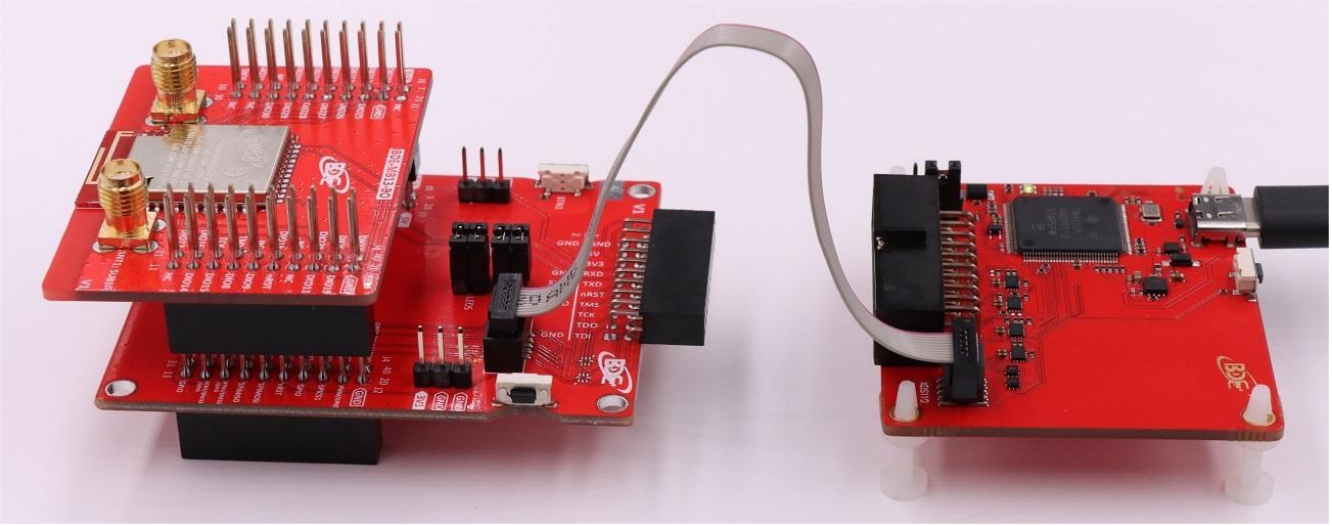


Figure 5. BDE-XDS110

The board can also be connected via a 2x5P 1.27-pitch ribbon cable to the XDS110 debugger.



4. Getting Started

Below section describes how to set up the development environment of BDE-MB1354P101xx module, including the hardware and software.

4.1. Prerequisites

4.1.1. Hardware

- BDE-MB13-BO, breakout board with module BDE-MB1354P101xxx, depending on what module variant you are choosing;
- BDE-LPEM, Launchpad evaluation interface module;
- BDE-XDS110, XDS110 debugger;
- USB type C cable;
- PC

4.1.2. Software and Tools

- Code Composer Studio
- SIMPLELINK-LOWPOWER-F2-SDK
- SimpleLink Connect mobile app
- UniFlash
- SmartRF Studio
- PuTTY

4.2. Hardware Setup

1. Plug BDE-MB13-BO into the BDE-LPEM evaluation module and connect BDE-LPEM to BDE-XDS110, as shown in Figure 6;
2. Connect the jumpers as shown in Figure 6;
3. Connect the BDE-XDS110 to the PC via a USB type C cable. The XDS110 will power the module with 3.3V if the jumper on BDE-XDS110 is plugged to XDS.

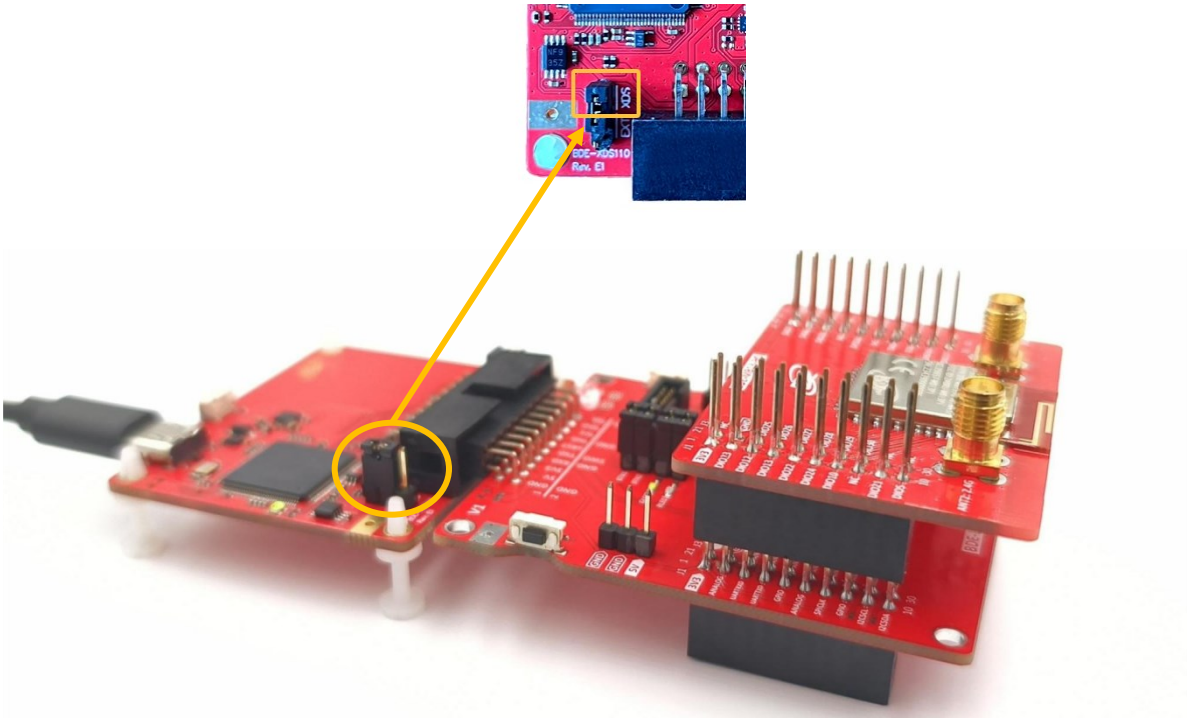
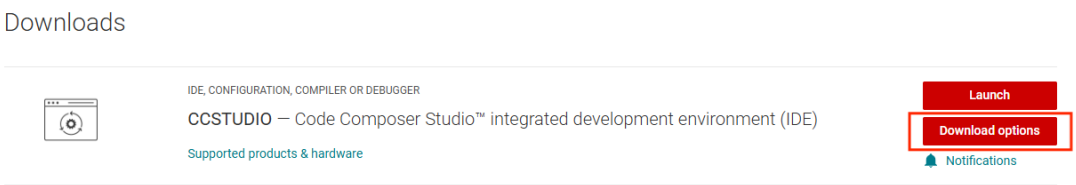


Figure 6. Hardware Setup

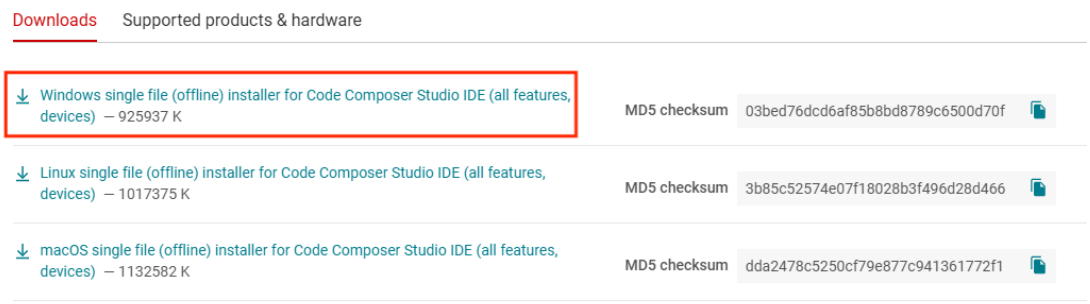
4.3. Software Setup

4.3.1. Download and Install CCS

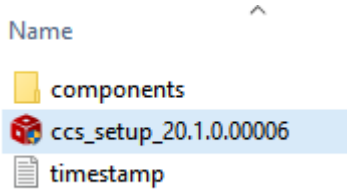
1. Visit the following link.
   https://www.ti.com/tool/CCSTUDIO
2. Find CCSTUDIO item, and click “Download options”.



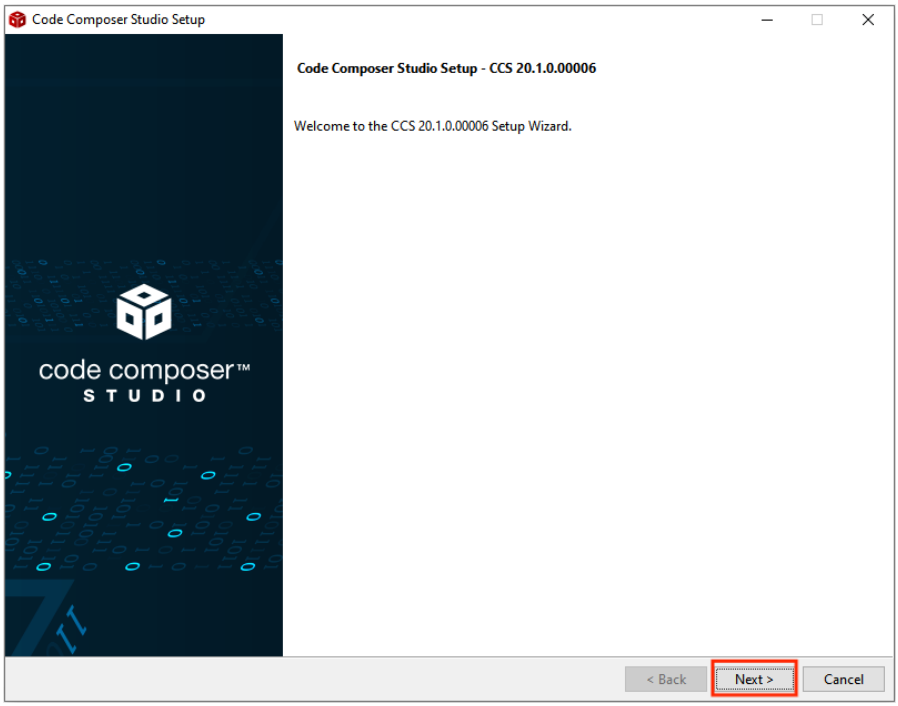
3. Choose the target platform, in this case, Windows is selected.



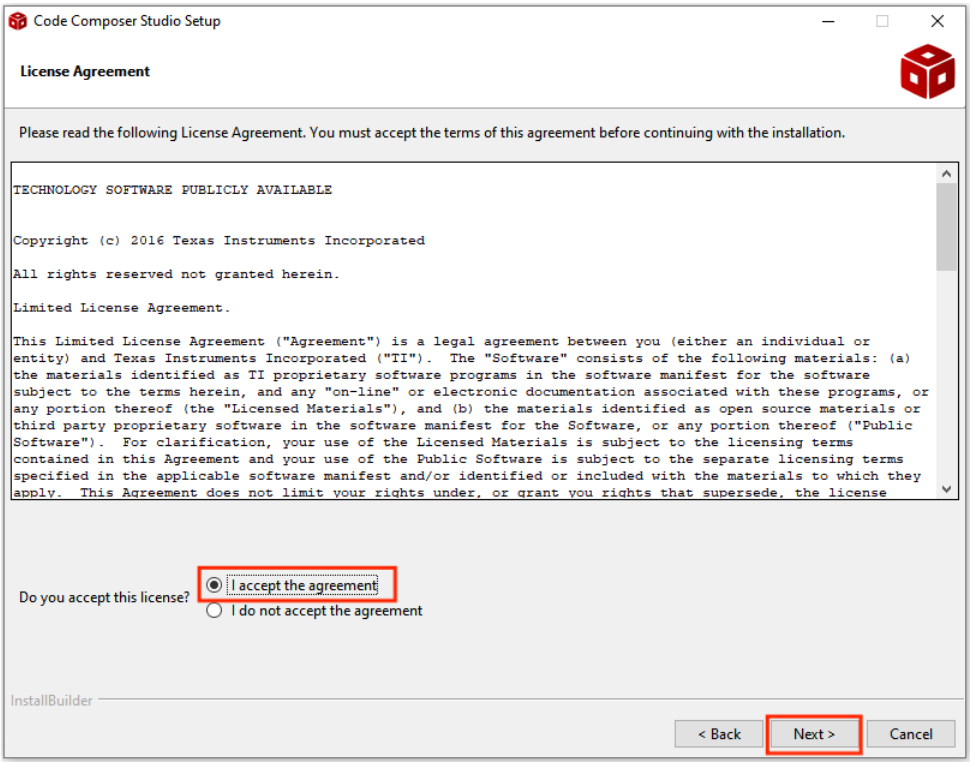
4. Decompress the zip file, and double-click “ccs_setup_xx.exe”.



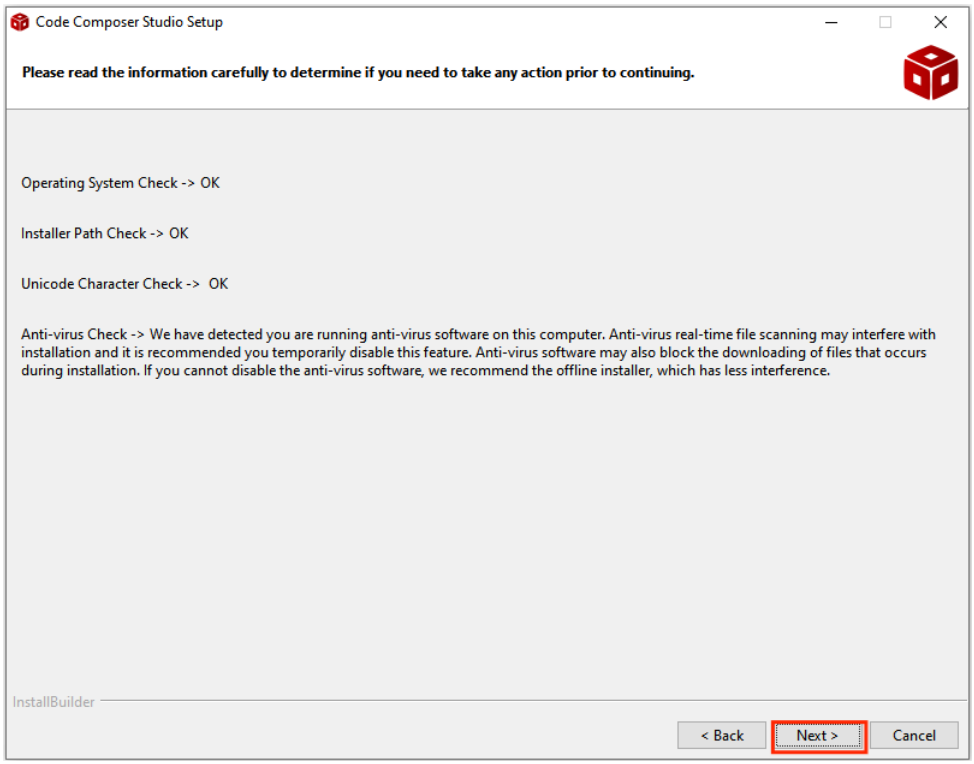
5. Click “Next”.



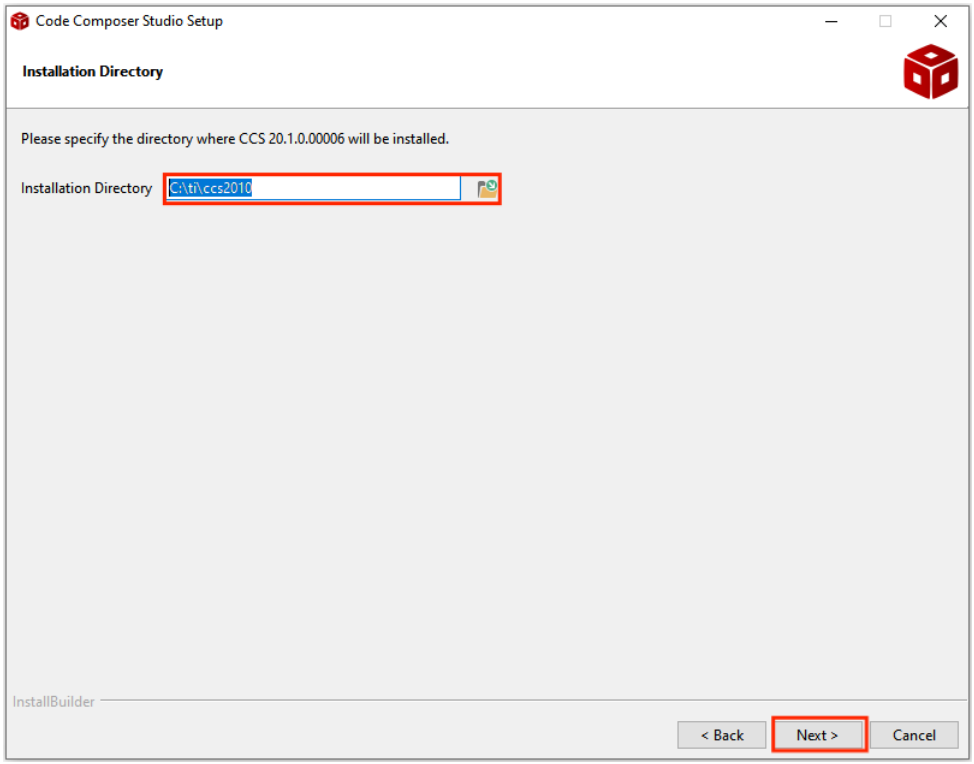
6. Select “I accept the agreement”, and then click “Next”.



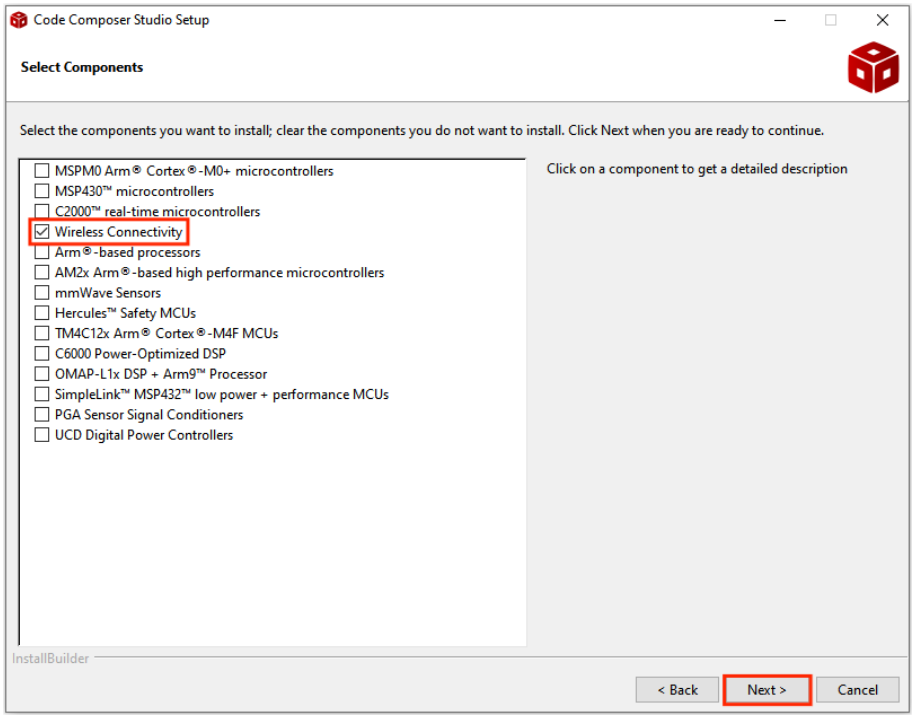
7. Click “Next”.



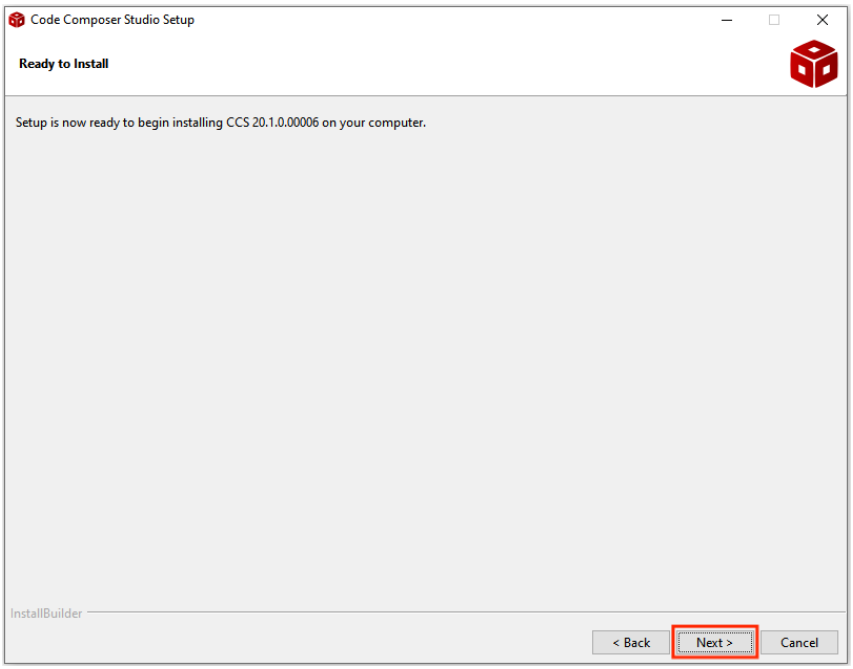
8. Choose the installation directory, and then click “Next”.



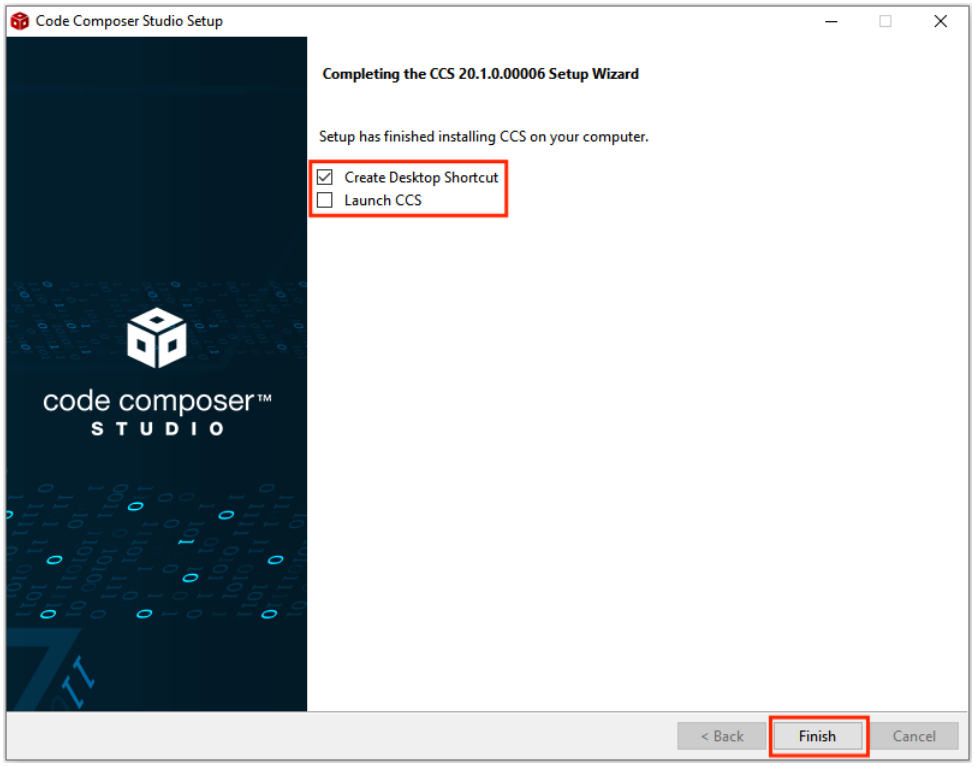
9. Select “Wireless connectivity” components, and then click “Next”.



10. Click “Next”, and the installation will start. Wait until the installation is complete.

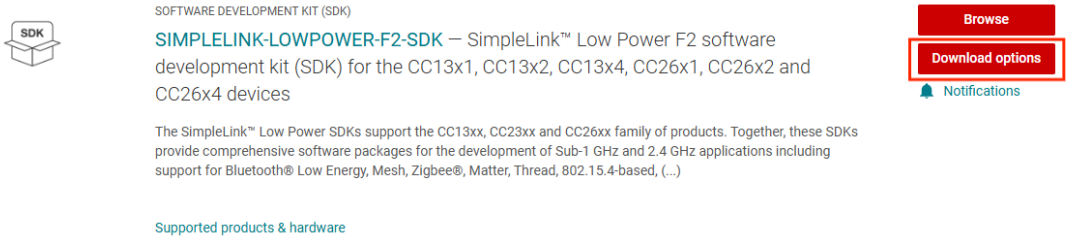


11. Click “Finish”.

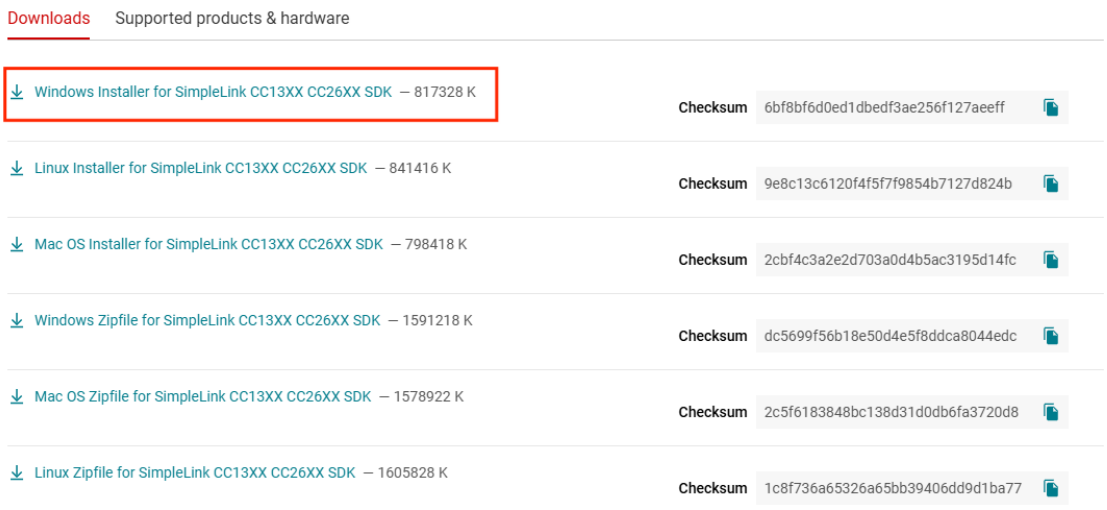


4.3.2. Download and Install SDK

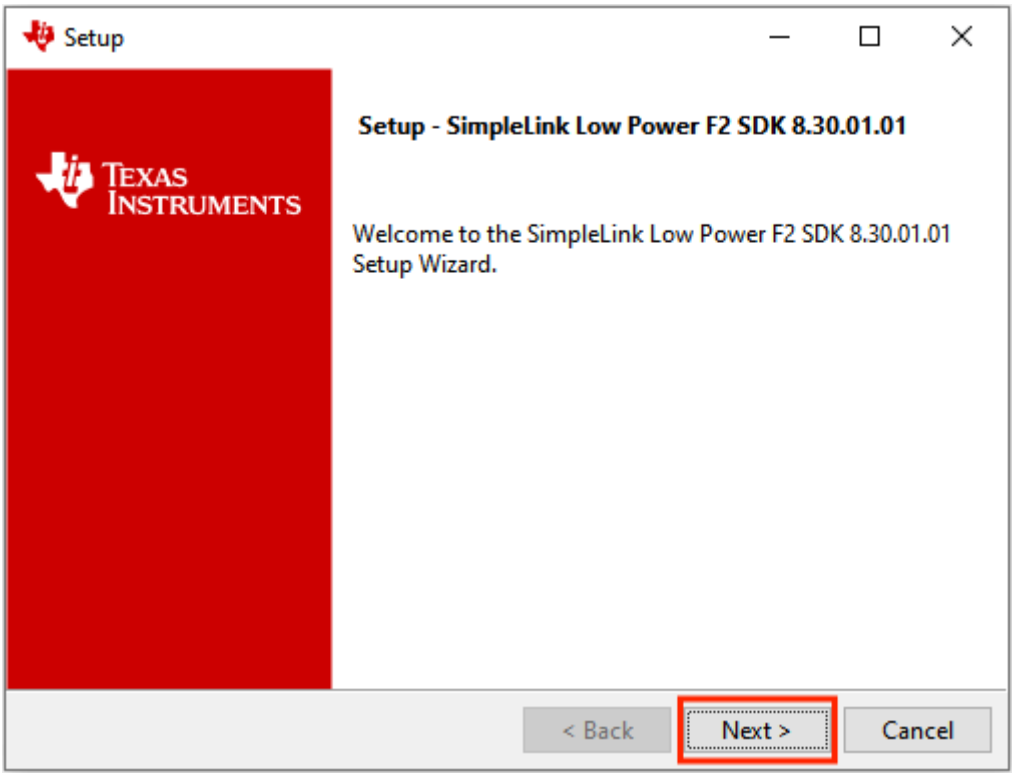
1. Visit the following link.
   https://www.ti.com/tool/download/SIMPLELINK-LOWPOWER-F2-SDK
2. Find “SIMPLELINK-LOWPOWER-F2-SDK” item, and click “Download options”.



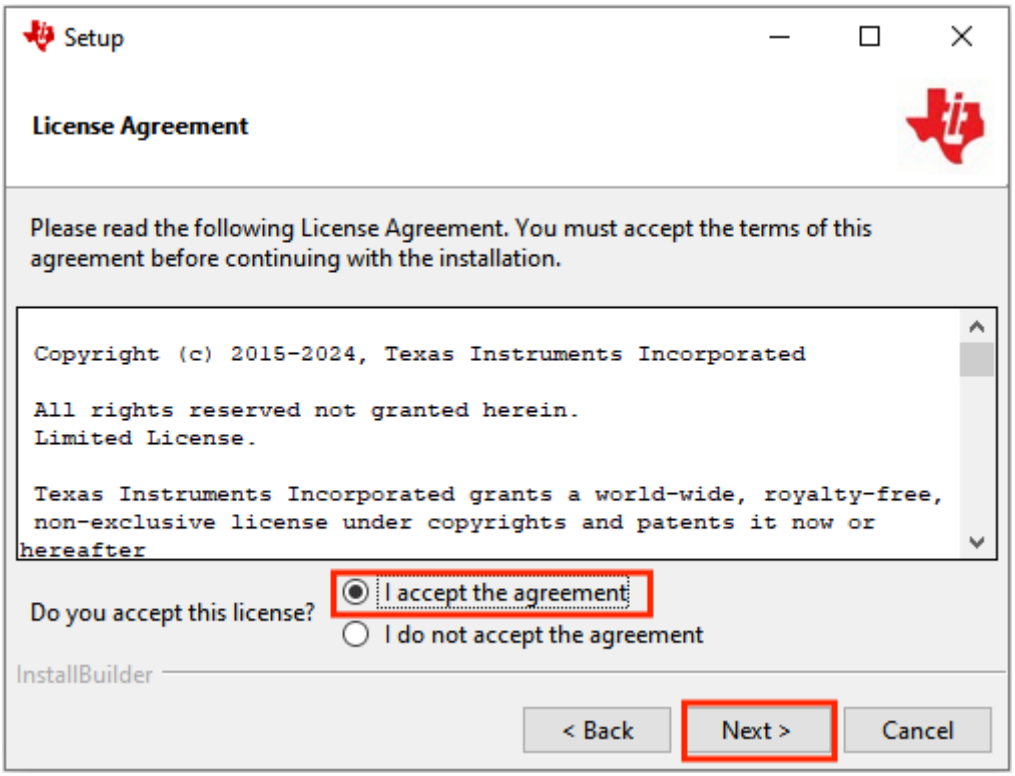
3. Choose the target platform, in this case, Windows is selected.



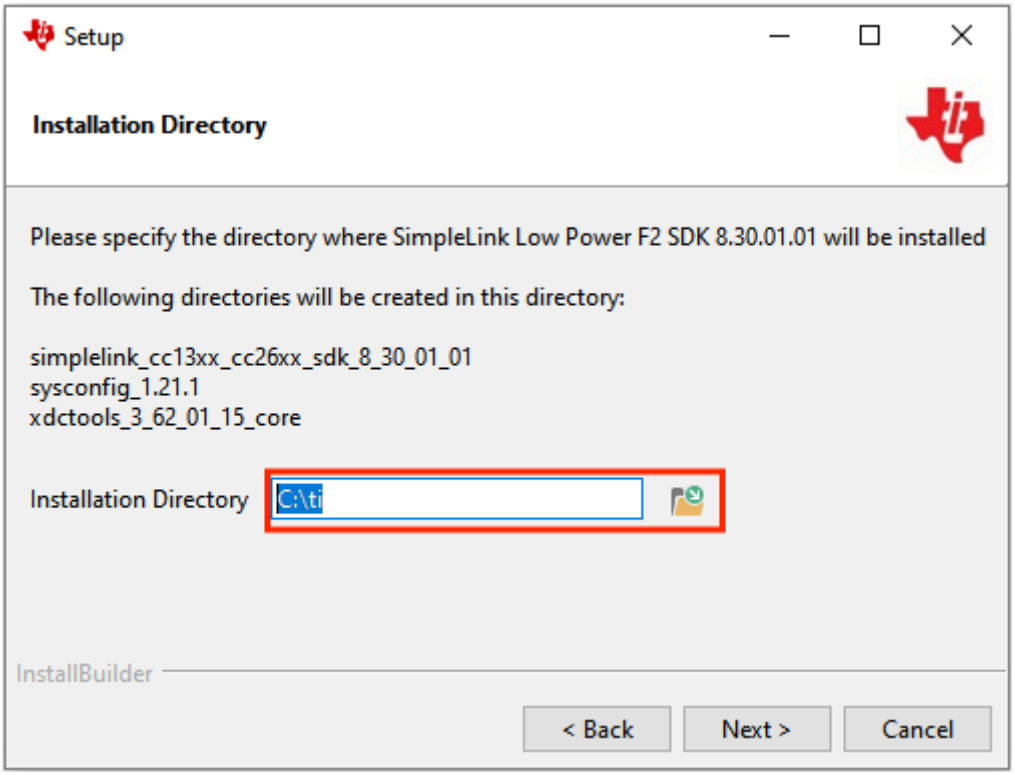
4. Double-click the installation package.
5. Click “Next”.



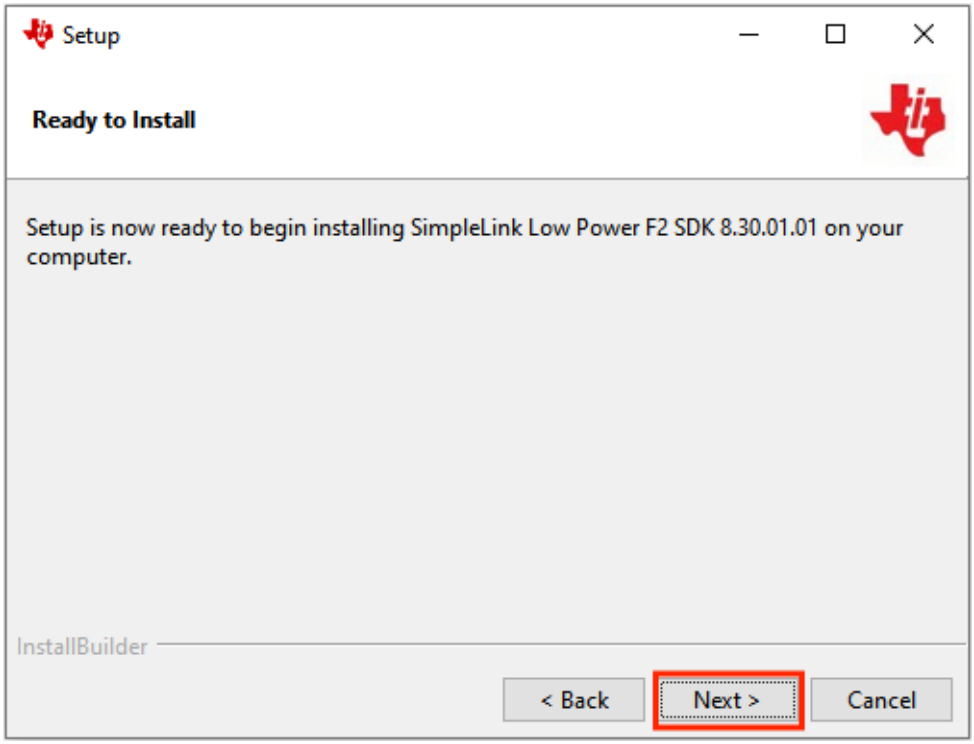
6. Select “I accept the agreement”, and then click “Next”.



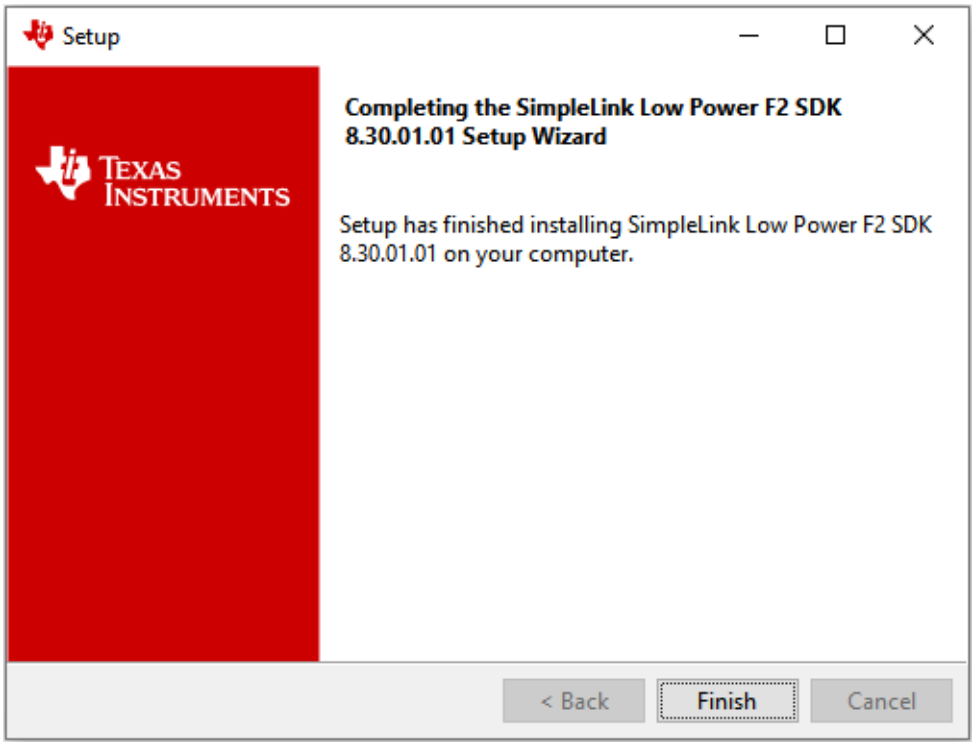
7. Choose the installation directory, and then click “Next”. Make sure the installation directory does not include non-ASCII characters or spaces.



8. Click “Next”, and the installation will start. Wait until the installation is complete.



9. Click “Finish”.



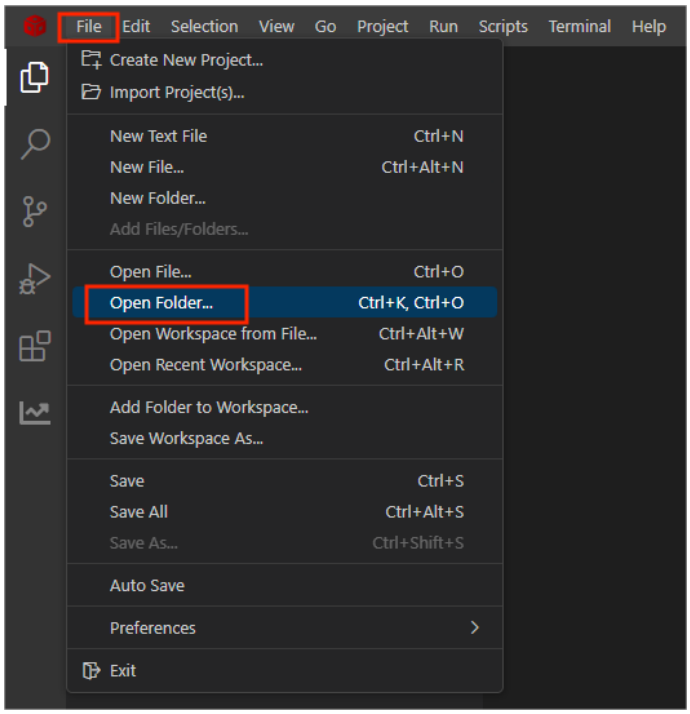
4.4. BLE5-STACK

This section describes how to use the BLE5-Stack projects with CCS.

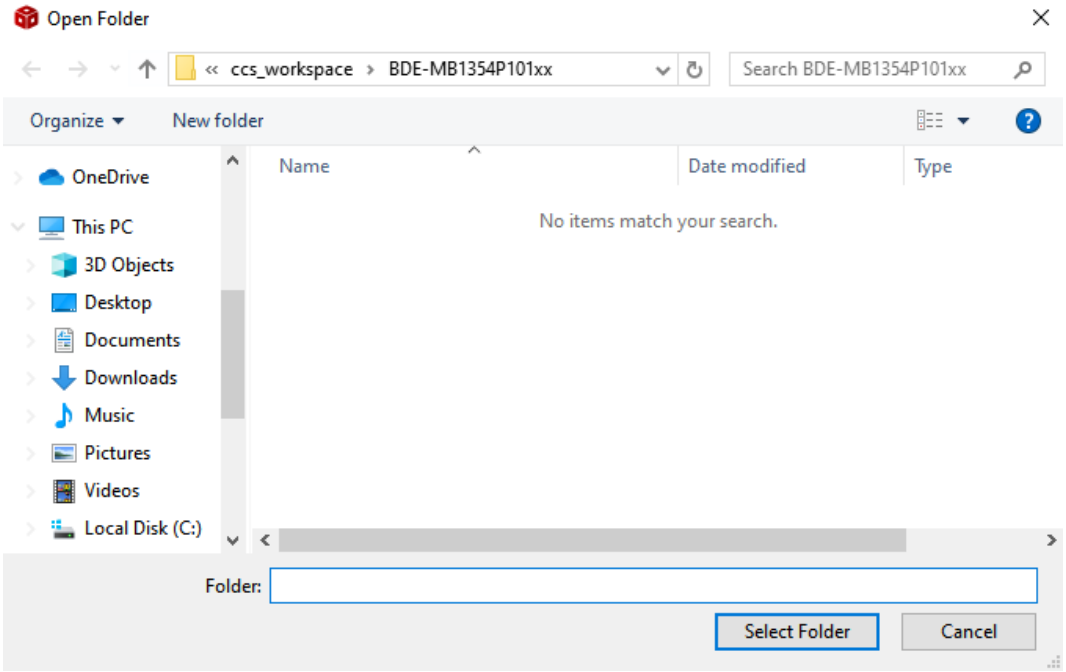
The BLE5-Stack examples are located at
C:\ti\simplelink_cc13xx_cc26xx_sdk_8_30_01_01\examples\rtos\LP_EM_CC1354P10_1\ble5stack

4.4.1. Import Project into CCS

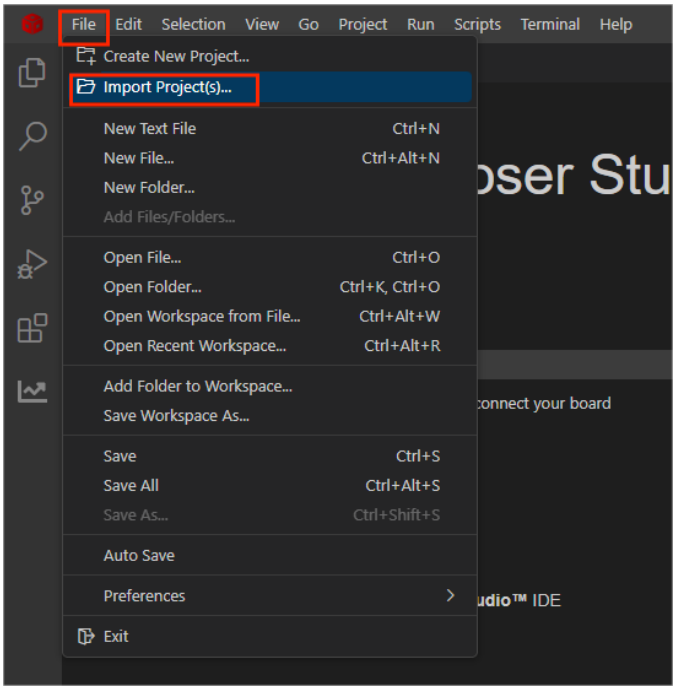
1. Open the CCS IDE.
2. Go to **File** -> **Open Folder**.



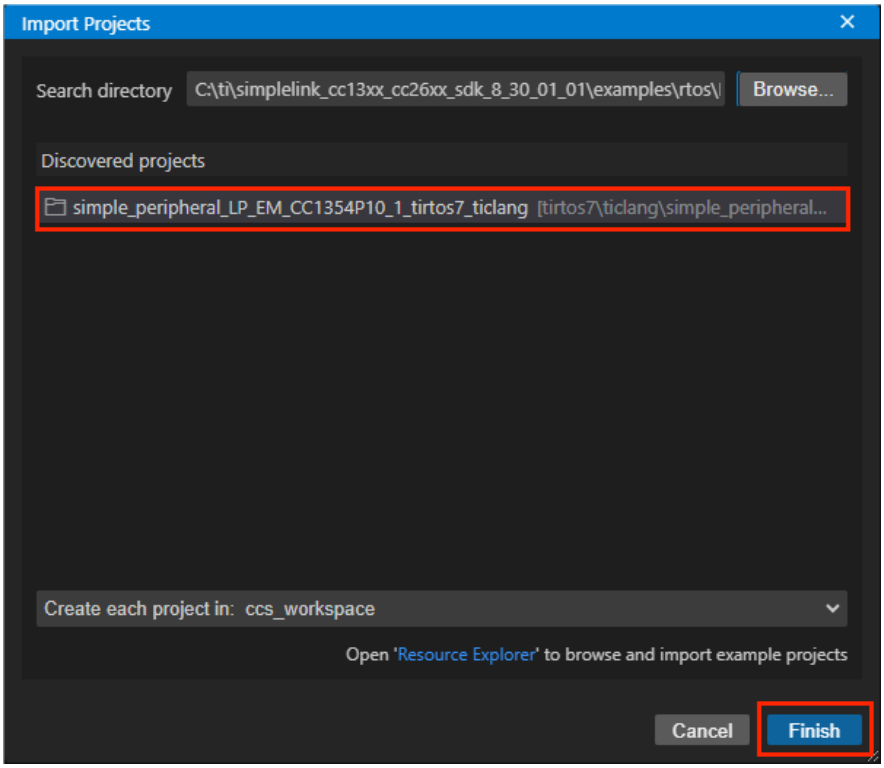
3. Select a custom path as the workspace.



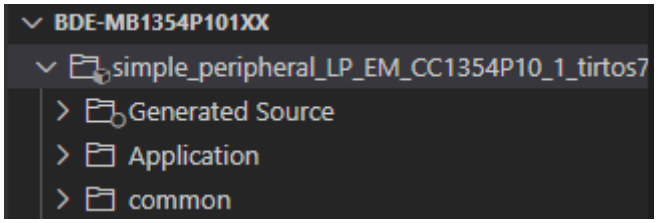
4. Go to **File** -> **Import Project(s)**.



5. Click “Browse”, navigate to the “simple_peripheral” project folder. The default path is: C:\ti\simplelink_cc13xx_cc26xx_sdk_8_30_01_01\examples\rtos\LP_EM_CC1354P10_1\ble5stack\simple_peripheral
6. Select the project, and click “Finish”.



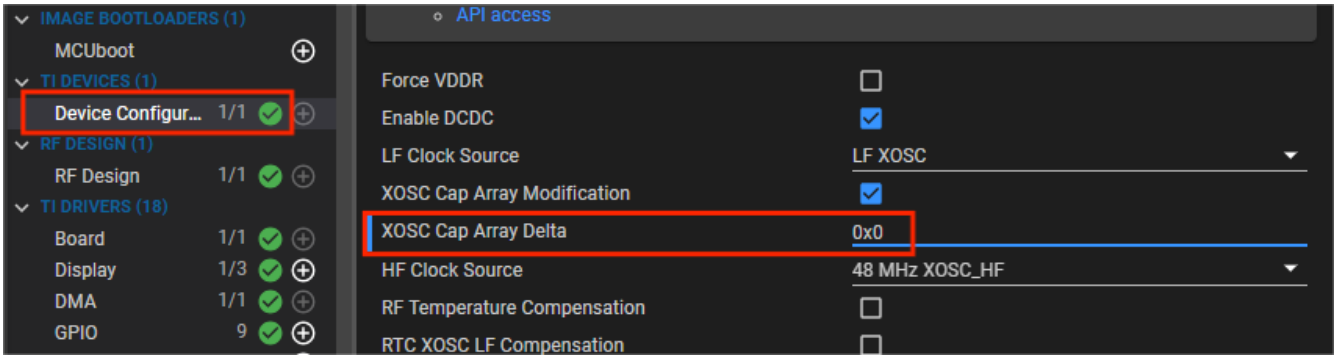
7. The “simple_peripheral” project will be imported into the IDE.



4.4.2. Modify XOSC Cap Setting

The “XOSC Cap Array Delta” parameter will impact the radio frequency offset. An inappropriate value can affect the RF performance. Follow the steps below to set the value that fits the module.

1. Open “simple_peripheral.syscfg” file by double-click it.
2. Select the “Device Configuration” item. Modify “XOSC Cap Array Delta” to 0x00.

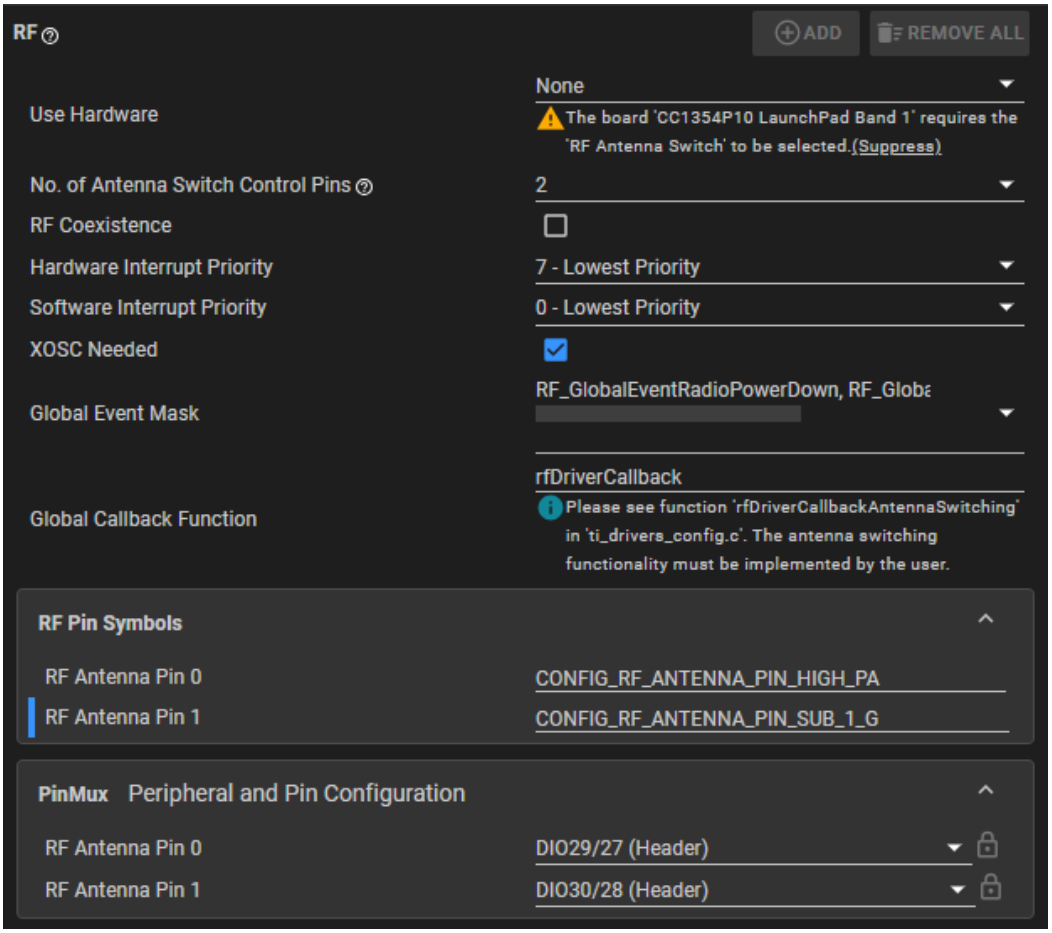


3. Save the changes.

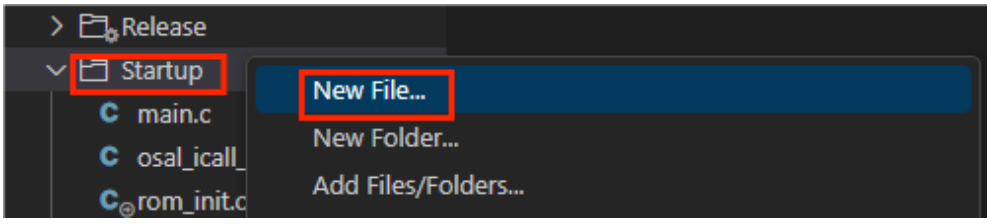
4.4.3. Modify RF Switch Configuration

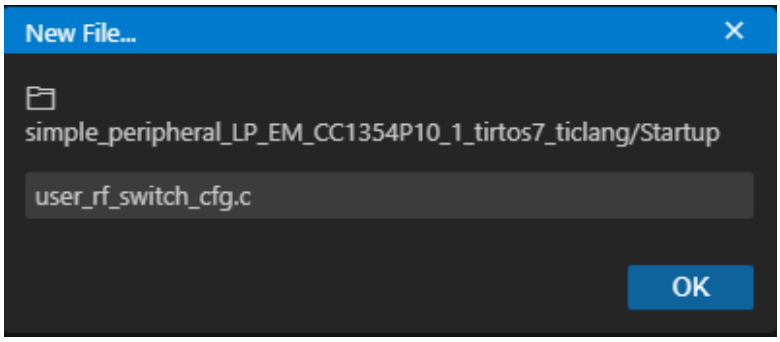
The BDE-MB1354P101 module uses a different RF switch design compared to TI LP-EM-CC1354P10-1. Follow the steps below to adjust the RF configuration to fit the module.

1. Open syscfg file.
2. Select “RF” item under “TI DRIVERS”. Modify the RF settings according to the following diagram.



3. Save the changes.
4. Right-click the “Startup” folder in the project and create a new C file named “user_rf_switch_cfg.c”.





5. Copy the following source code to user_rf_switch_cfg.c.

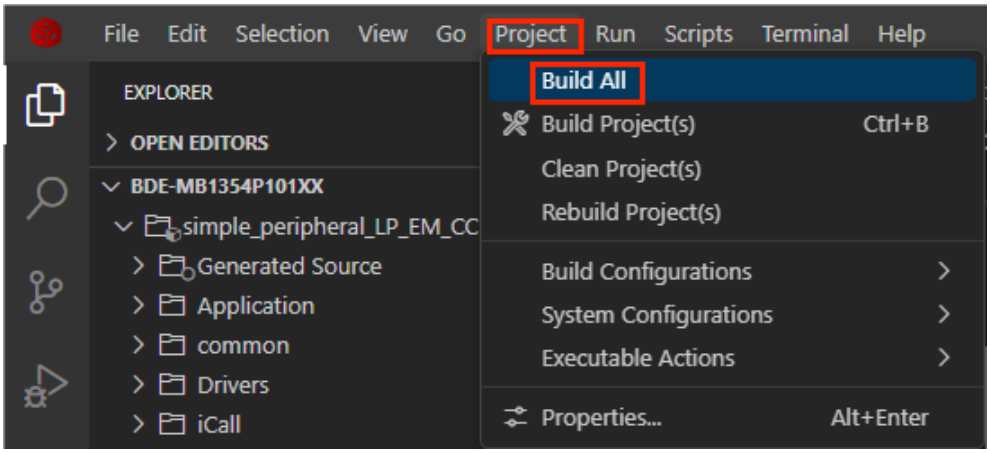
```
#include <stddef.h>
#include <stdint.h>
#ifndef DeviceFamily_CC13X4
#define DeviceFamily_CC13X4
```

```
#endif
#include <ti/devices/DeviceFamily.h>
#include "ti_drivers_config.h"
#include <ti/drivers/GPIO.h>
#include <ti/drivers/rf/RF.h>

/*
 * ======== Antenna switching ========
 */
/*
 * ======== rfDriverCallbackAntennaSwitching ========
 * Sets up the antenna switch depending on the current PHY configuration.
 * Fit BDE-MB13P2 module's switch
 *
 * Truth table:
 *
 * Path       DIO29 DIO30
 * ========== ===== =====
 * Off        0     0
 * TX         1     0
 * RX         0     1
 */
void rfDriverCallbackAntennaSwitching(RF_Handle client, RF_GlobalEvent events, void *arg)
{

    if (events & RF_GlobalEventRadioSetup) {
        bool    sub1GHz   = false;
        uint8_t loDivider = 0;

        /* Switch off all paths. */
        GPIO_write(CONFIG_RF_ANTENNA_PIN_SUB_1_G, 0);
        GPIO_write(CONFIG_RF_ANTENNA_PIN_HIGH_PA, 0);

        /* Decode the current PA configuration. */
        RF_TxPowerTable_PAType paType = (RF_TxPowerTable_PAType)RF_getTxPower(client).paType;

        /* Decode the generic argument as a setup command. */
        RF_RadioSetup* setupCommand = (RF_RadioSetup*)arg;

        switch (setupCommand->common.commandNo) {
            case (CMD_RADIO_SETUP):
            case (CMD_BLE5_RADIO_SETUP):
                    loDivider = RF_LODIVIDER_MASK & setupCommand->common.loDivider;
```

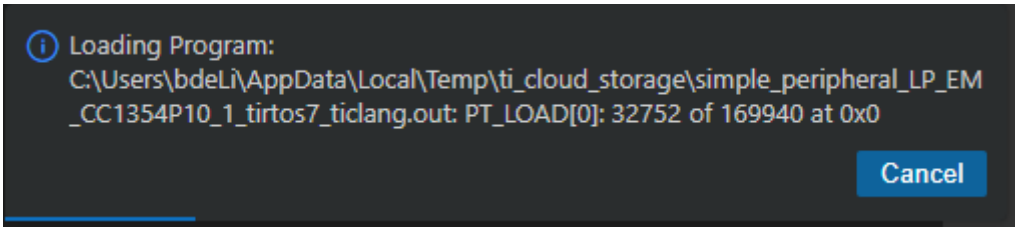
```
                    /* Sub-1GHz front-end. */
                    if (loDivider != 0) {
                        sub1GHz = true;
                    }
                    break;
            case (CMD_PROP_RADIO_DIV_SETUP):
                    loDivider = RF_LODIVIDER_MASK & setupCommand->prop_div.loDivider;

                    /* Sub-1GHz front-end. */
                    if (loDivider != 0) {
                        sub1GHz = true;
                    }
                    break;
            default:break;
        }

        if (sub1GHz) {
            /* Sub-1 GHz */
            if (paType == RF_TxPowerTable_HighPA) {
                /* PA enable --> HIGH PA
                 * LNA enable --> Sub-1 GHz
                 */
                /* Note: RFC_GPO3 is a work-around because the RFC_GPO1 (PA enable signal) is sometimes not
                         de-asserted on CC1352 Rev A. */
                GPIO_setConfigAndMux(CONFIG_RF_ANTENNA_PIN_SUB_1_G, GPIO_CFG_OUTPUT, IOC_PORT_RFC_GPO0);
                GPIO_setConfigAndMux(CONFIG_RF_ANTENNA_PIN_HIGH_PA, GPIO_CFG_OUTPUT, IOC_PORT_RFC_GPO3);
            } else {
                /* RF core active --> Sub-1 GHz */
                GPIO_setConfigAndMux(CONFIG_RF_ANTENNA_PIN_HIGH_PA, GPIO_CFG_OUTPUT, IOC_PORT_GPIO);
                GPIO_setConfigAndMux(CONFIG_RF_ANTENNA_PIN_SUB_1_G,    GPIO_CFG_OUTPUT    |    GPIO_CFG_OUT_HIGH,
IOC_PORT_GPIO);
            }
        }
    }
    else if (events & RF_GlobalEventRadioPowerDown) {
        /* Switch off all paths. */
        GPIO_write(CONFIG_RF_ANTENNA_PIN_SUB_1_G, 0);
        GPIO_write(CONFIG_RF_ANTENNA_PIN_HIGH_PA, 0);

        /* Reset the IO multiplexer to GPIO functionality */
        GPIO_setConfigAndMux(CONFIG_RF_ANTENNA_PIN_SUB_1_G, GPIO_CFG_OUTPUT, IOC_PORT_GPIO);
        GPIO_setConfigAndMux(CONFIG_RF_ANTENNA_PIN_HIGH_PA, GPIO_CFG_OUTPUT, IOC_PORT_GPIO);
    }
}
```

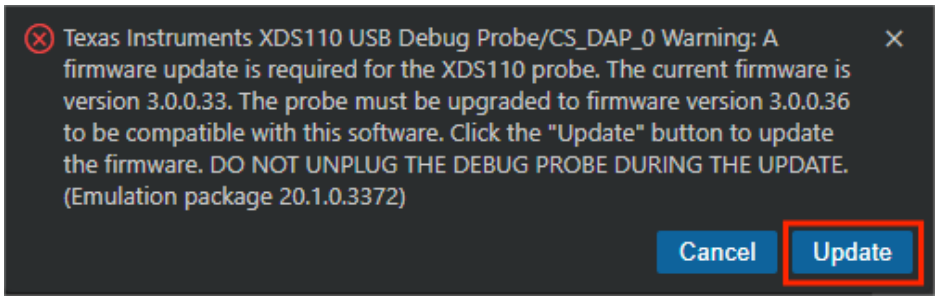
4.4.4. Build and Download Project

1. Select **Project** -> **Build All** to build the application project.

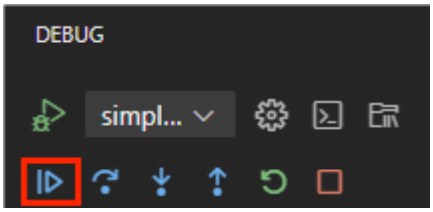


2. After building successfully, select **Run** -> **Debug** to download the application to the module. If the following message occur, click “Update”.



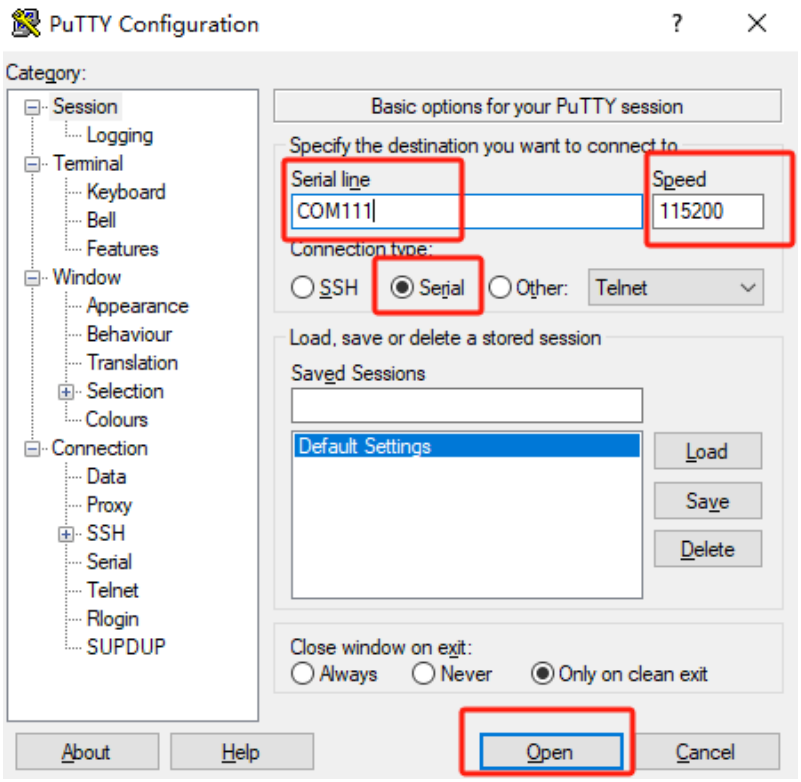


3. After the update process is finished, click “Continue”.

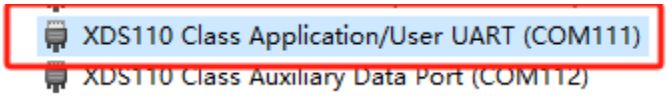


4.4.5. Running the Demo

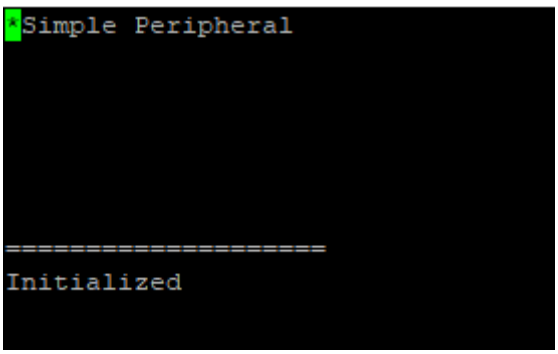
1. Open PuTTY, open the serial port with the following settings:



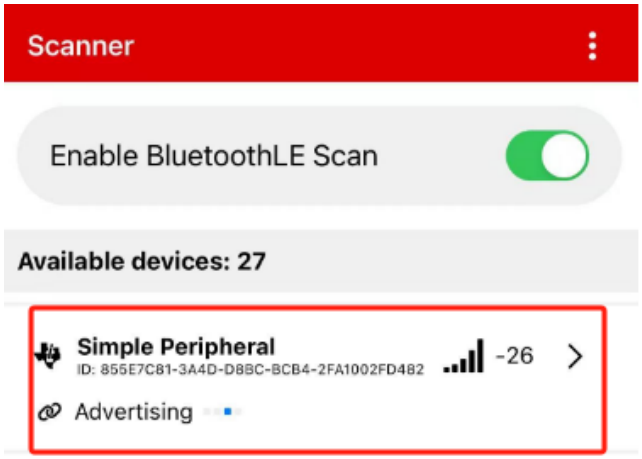
The COM number can be found in the System Device Manager:



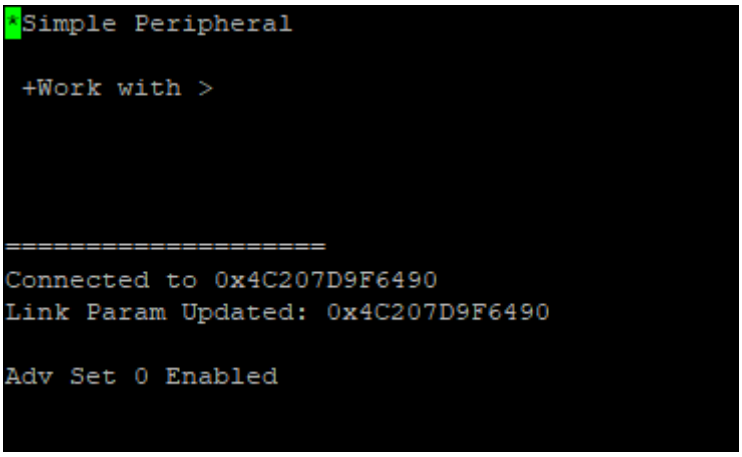
2. Debug the project and click “Continue” in CCS. The demo will begin execution. The PuTTY will display the menu as shown below:



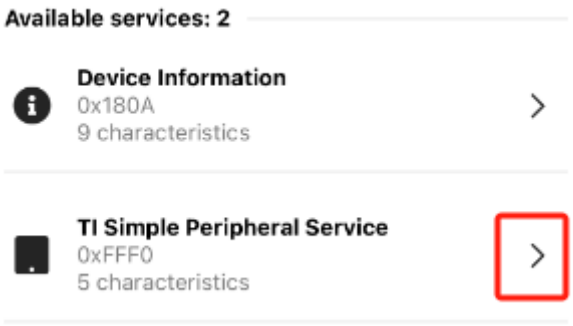
3. Open SimpleLink Connect mobile app. Click “Enable BluetoothLE Scan”, and you will see that the module is advertising.



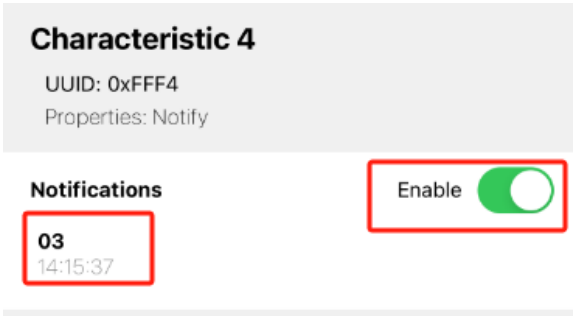
4. Click the right arrow on the right of “Simple Peripheral” item to connect to the module. After a successful connection, you will see PuTTY update the information.



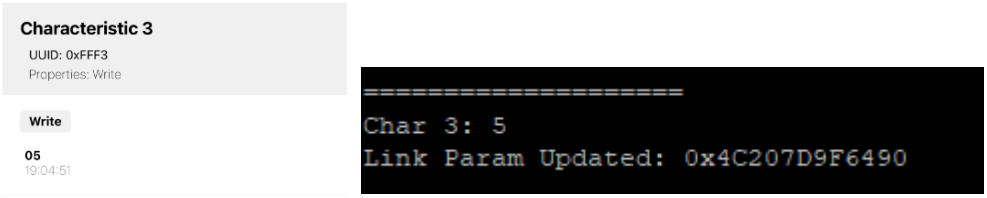
5. Select “TI Simple Peripheral Service”.



6. Enable the notification of “Characteristic 4”, the app will receive notification sent by the module.



7. Write “05” to “Characteristic 3”. PuTTY will print the write information.



4.5. Host_test Project & Btool Software

TI provides a PC tool called “btool” to help developers simplify development. Developers can use Btool to scan and connect to peripheral devices, or to send raw HCI commands.

The directory for Btool is:
C:\ti\simplelink_cc13xx_cc26xx_sdk_8_30_01_01\tools\ble5stack\btool.

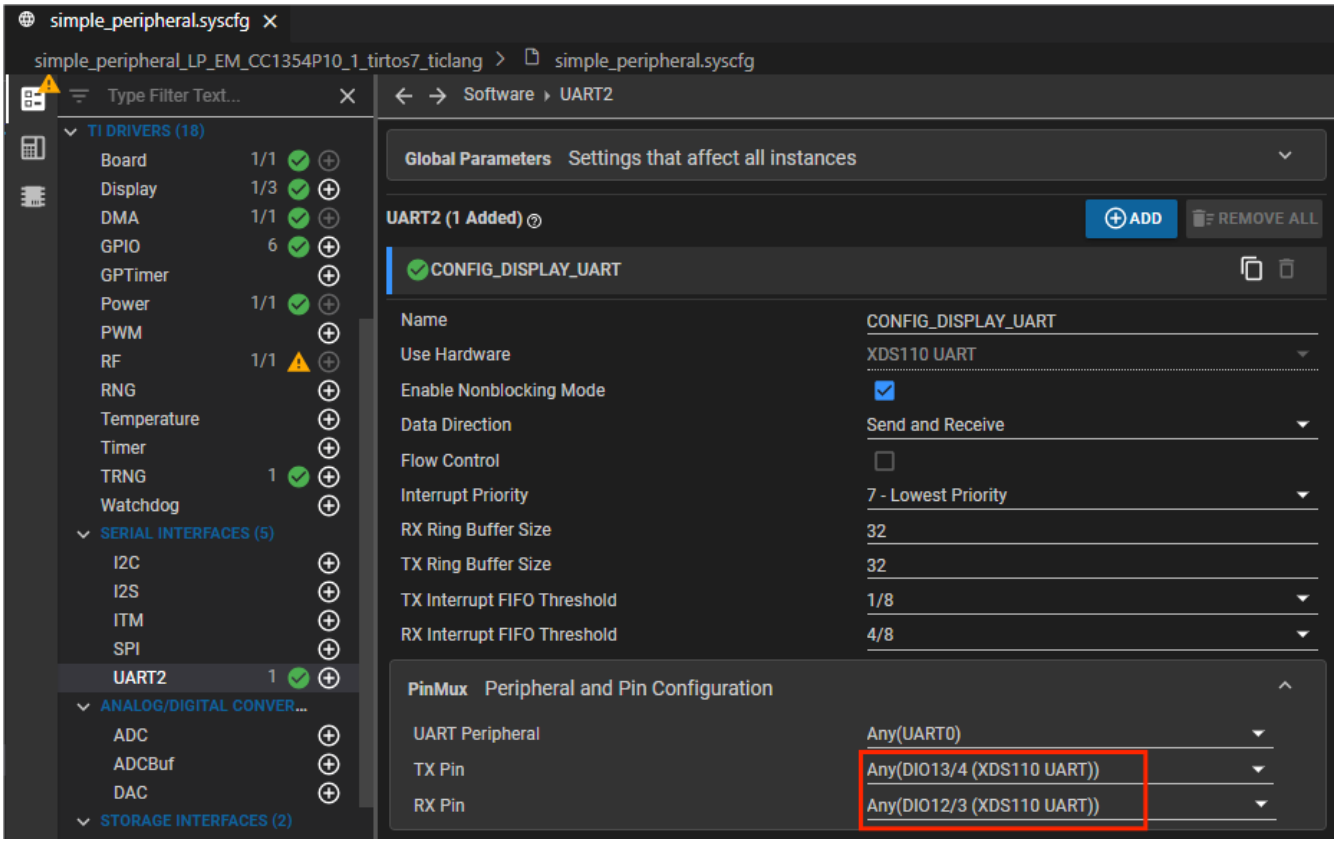
Before using Btool, the module must have the host_test application flashed onto it first.

The project directory for host_test is:
C:\ti\simplelink_cc13xx_cc26xx_sdk_8_30_01_01\examples\rtos\LP_EM_CC1354P10_1\ble5stack\host_test.

Before building the host_test project, developers should perform the additional steps mentioned in sections 4.4.2 and 4.4.3.

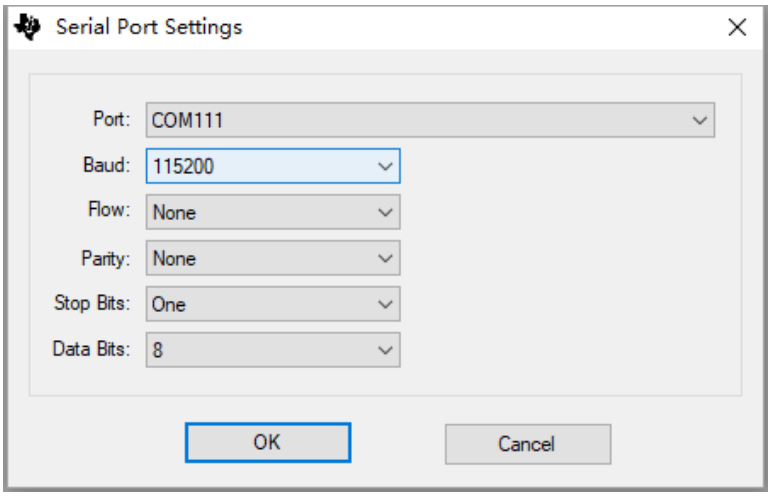
The module will enter DTM mode once the host_test application is flashed. Btool can connect to the module through UART interface.

The pins used by UART can be found in the following diagram:



The following steps demonstrate how to operate Btool.

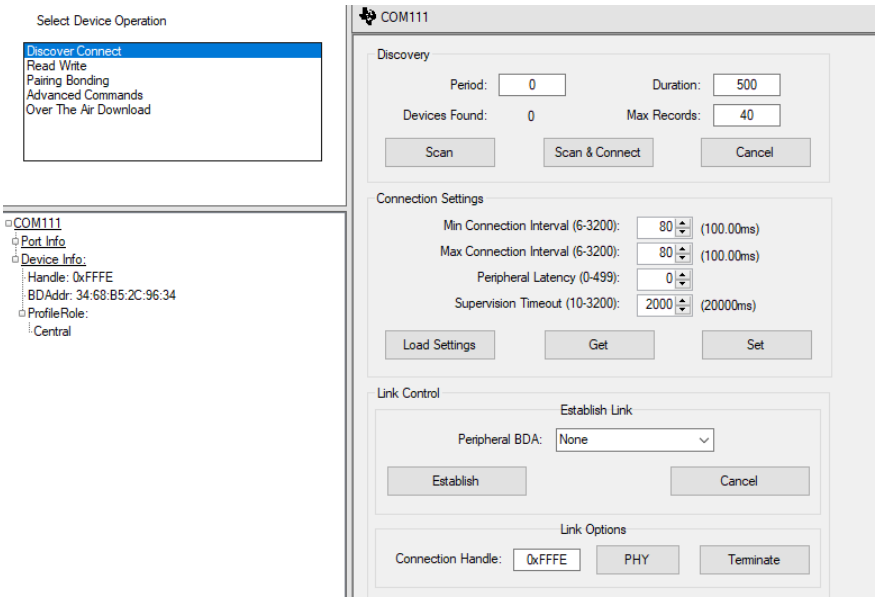
1. Open Btool.
2. Open the serial port with the following settings, and make sure the correct port is selected.



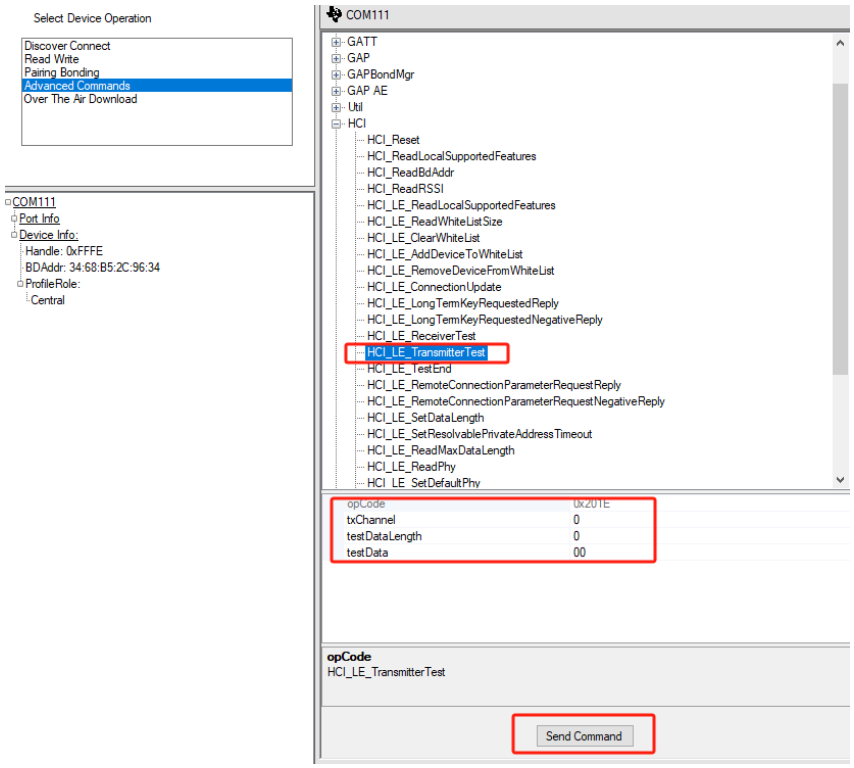
3. If the hardware functions properly, the following UART logs will be printed.

```
[1] : <Info> - 02:40:25.158
Port opened at 2024/10/22 14:40:25
------------------------------------------------------------------
[2] : <Tx> - 02:40:25.237
-Type            : 0x01 (Command)
-OpCode          : 0xFC1D (HCIExt_ResetSystemCmd)
-Data Length     : 0x01 (1) byte(s)
 Type            : 0x00 (0) (Chip Reset)
Dump(Tx):
0000:01 1D FC 01 00                                        .....
------------------------------------------------------------------
[3] : <Rx> - 02:40:25.384
-Type            : 0x04 (Event)
-EventCode       : 0x00FF (HCI_LE_ExtEvent)
-Data Length     : 0x05 (5) bytes(s)
 Event           : 0x041D (1053) (HCIExt_ResetSystemCmdDone)
 Status          : 0x00 (0) (SUCCESS)
 CmdOpCode       : 0xFC1D (HCIExt_ResetSystemCmd)
Dump(Rx):
0000:04 FF 05 1D 04 00 1D FC                               ........
------------------------------------------------------------------
[4] : <Tx> - 02:40:25.903
-Type            : 0x01 (Command)
-OpCode          : 0xFE00 (GAP_DeviceInit)
-Data Length     : 0x08 (8) byte(s)
 ProfileRole     : 0x08 (8) (
                   Central)
 AddrMode        : 0x00 (0) (ADDRMODE_PUBLIC)
 RandomAddr      : 00:00:00:00:00:00
Dump(Tx):
0000:01 00 FE 08 08 00 00 00 00 00 00 00                   ............
------------------------------------------------------------------
[5] : <Rx> - 02:40:25.980
-Type            : 0x04 (Event)
-EventCode       : 0x00FF (HCI_LE_ExtEvent)
-Data Length     : 0x06 (6) bytes(s)
 Event           : 0x067F (1663) (GAP_HCI_ExtentionCommandStatus)
 Status          : 0x00 (0) (SUCCESS)
```

4. Developers can scan for and connect to peripheral devices on the “Discover Connect” page.



5. The “Advanced Commands” page provides a variety of HCI commands. For example, developers can send “HCI_LE_Transmitter_Test” command in the following diagram.



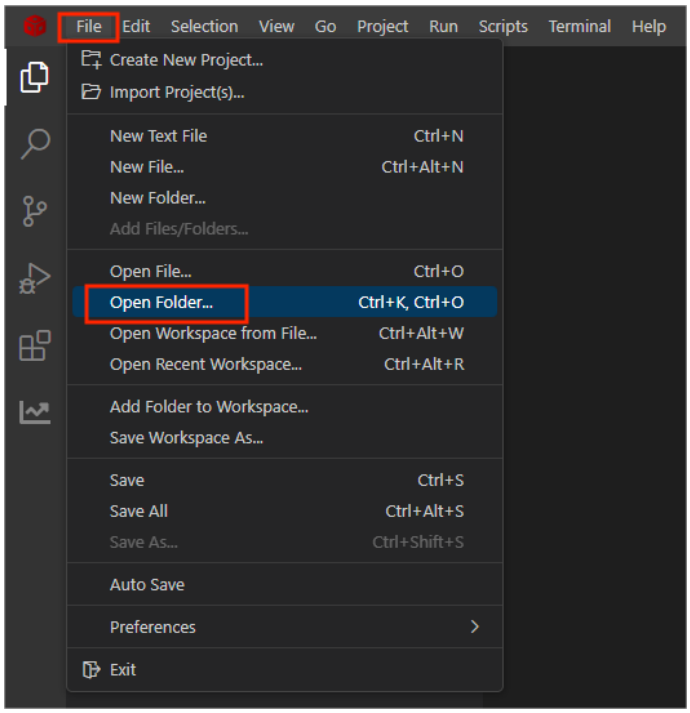
4.6. TI154STACK

TI’s royalty-free TI 15.4-Stack is a complete software platform for developing applications that require extremely low-power, long-range, reliable, robust and secure wireless star-topology-based networking solutions. This section describes how to use the TI 15.4-Stack projects with CCS.

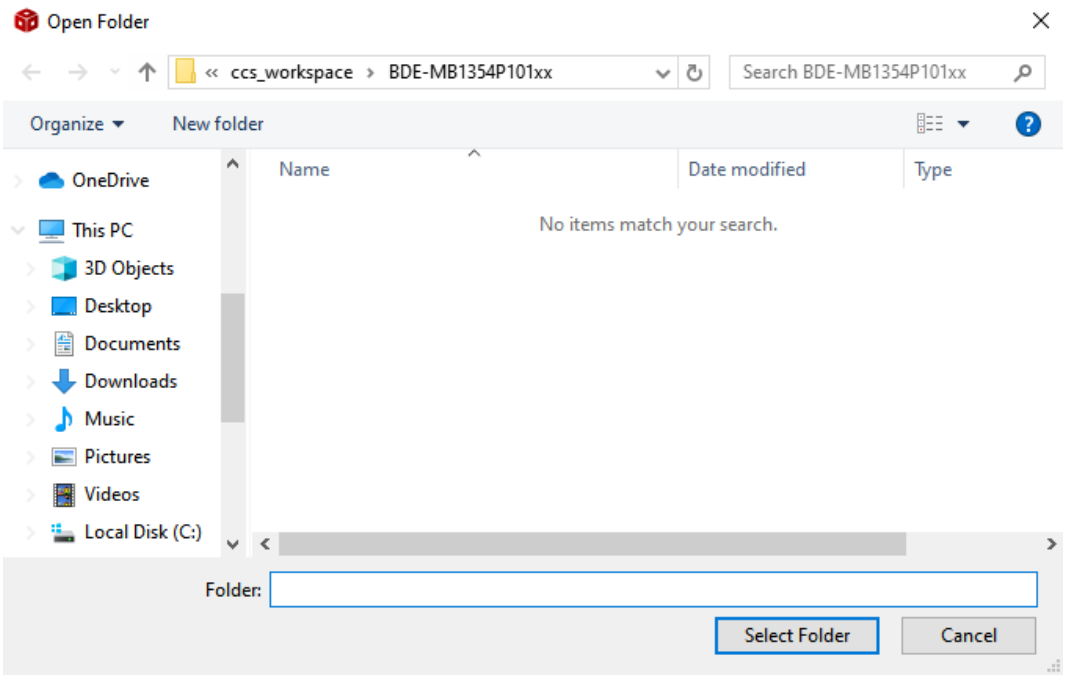
To demonstrate the TI 15.4-Stack demos, two modules are required.

4.6.1. Import Projects into CCS

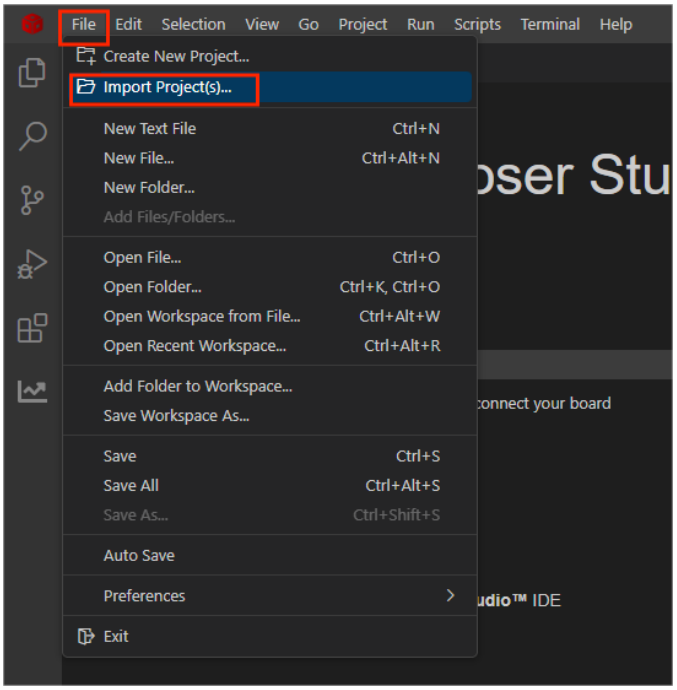
1. Open the CCS IDE.
2. Go to **File** -> **Open Folder**.



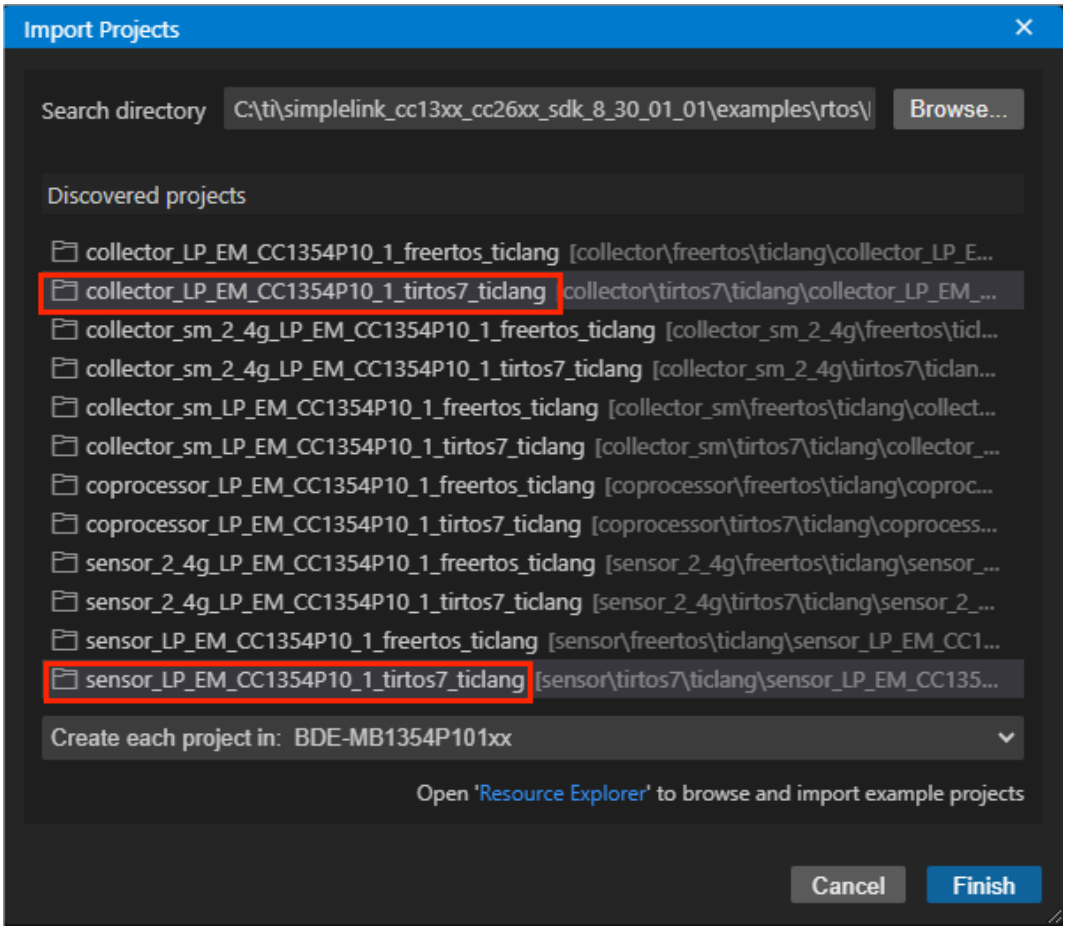
3. Select a custom path as the workspace.



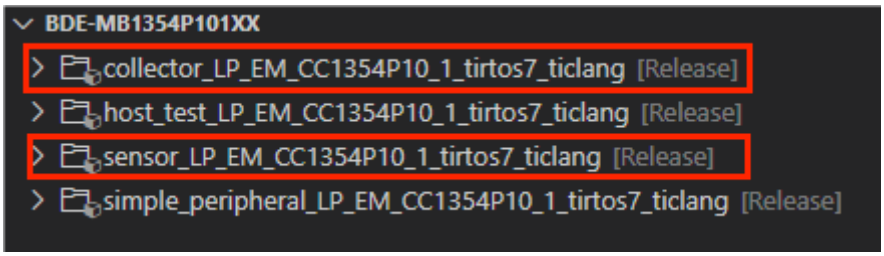
4. Go to **File** -> **Import Project(s)**.



5. Click “Browse”, navigate to the ti154stack folder. The default path is: C:\ti\simplelink_cc13xx_cc26xx_sdk_8_30_01_01\examples\rtos\LP_EM_CC1354P10_1\ti154stack.
6. Select “collector_LP_EM_CC1354P10_1_tirtos7_ticlang” and “sensor_LP_EM_CC1354P10_1_tirtos7_ticlang” projects, then click ‘Finish’.



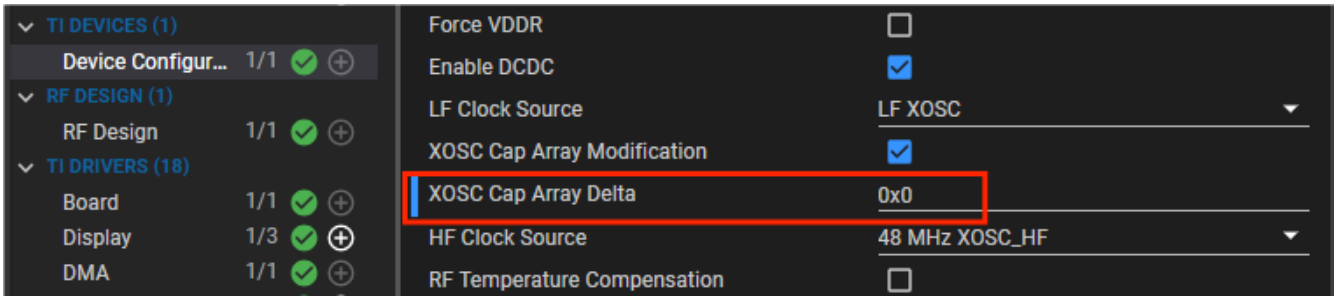
7. The two projects will be imported into the IDE.



4.6.2. Modify XOSC Cap Setting

The 'XOSC Cap Array Delta' parameter will impact the radio frequency offset. An inappropriate value can affect the RF performance. Follow the steps below to set the value that fits the module.

1. Open "collector.syscfg" file by double-click it.
2. Select the "Device Configuration" item. Modify "XOSC Cap Array Delta" to 0x00.

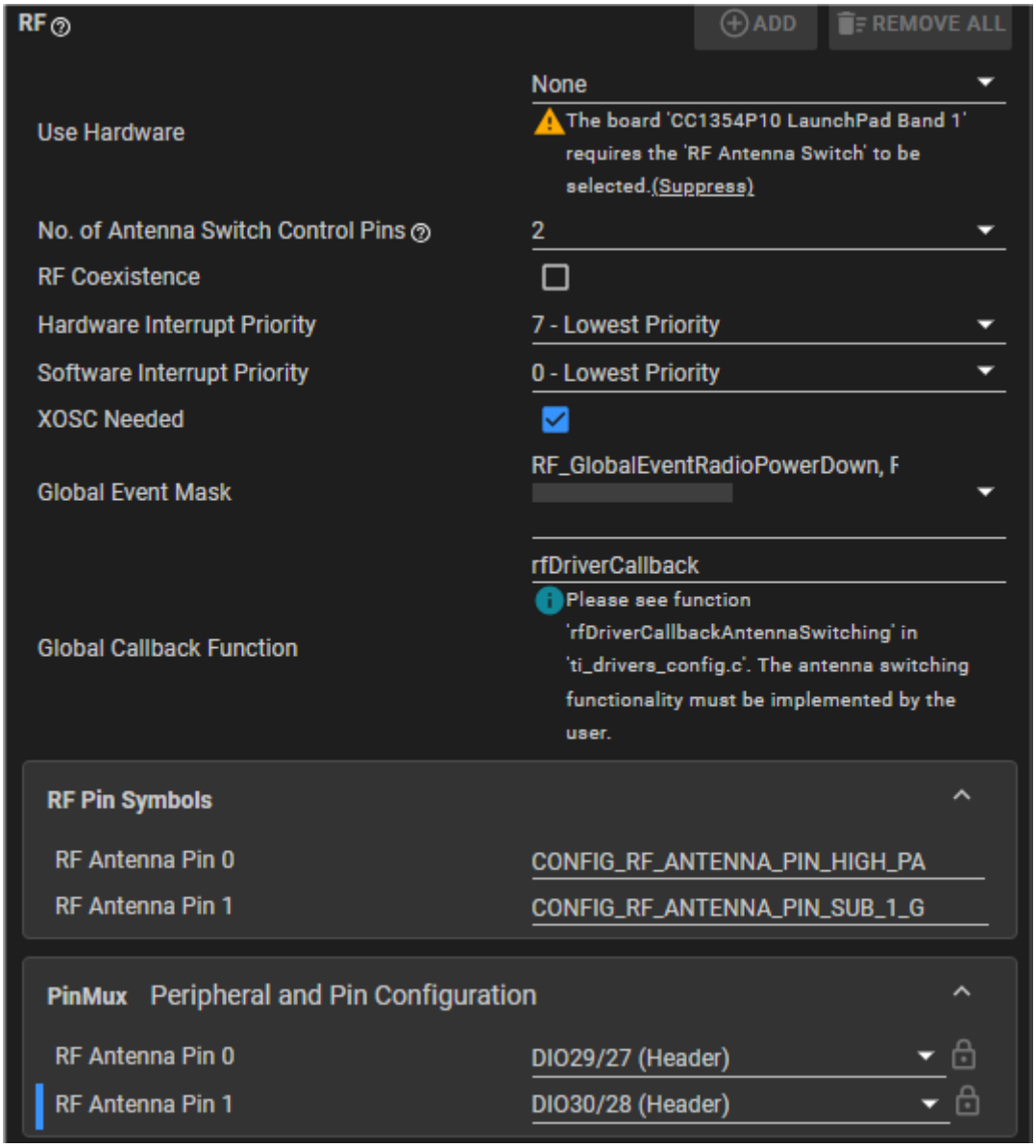


3. Save the changes.
4. Similarly, modify the "XOSC Cap Array Delta" value in sensor.syscfg.

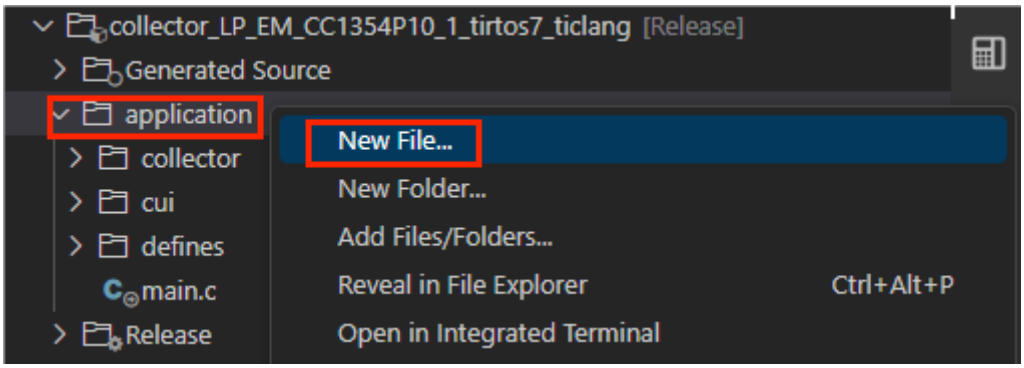
4.6.3. Modify RF Switch Configuration

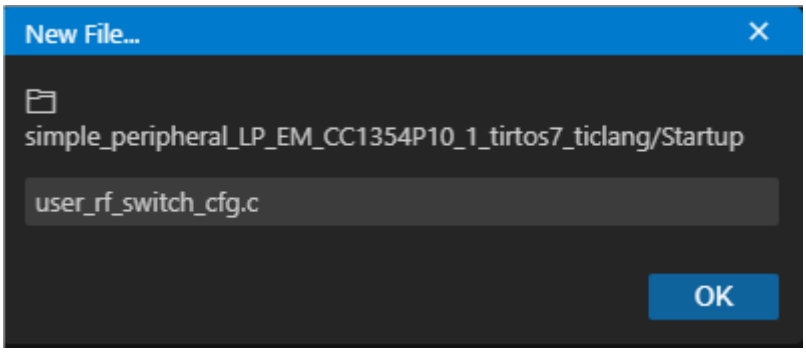
The BDE-MB1354P101 module uses a different RF switch design compared to TI LP-EM-CC1354P10-1. Follow the steps below to adjust the RF configuration to fit the module.

1. Open syscfg file.
2. Select 'RF' item under "TI DRIVERS". Modify the RF settings according to the following diagram.



3. Save the changes.
4. Right-click the “application” folder in the project and create a new C file named “user_rf_switch_cfg.c”.





5. Copy the following source code to user_rf_switch_cfg.c.

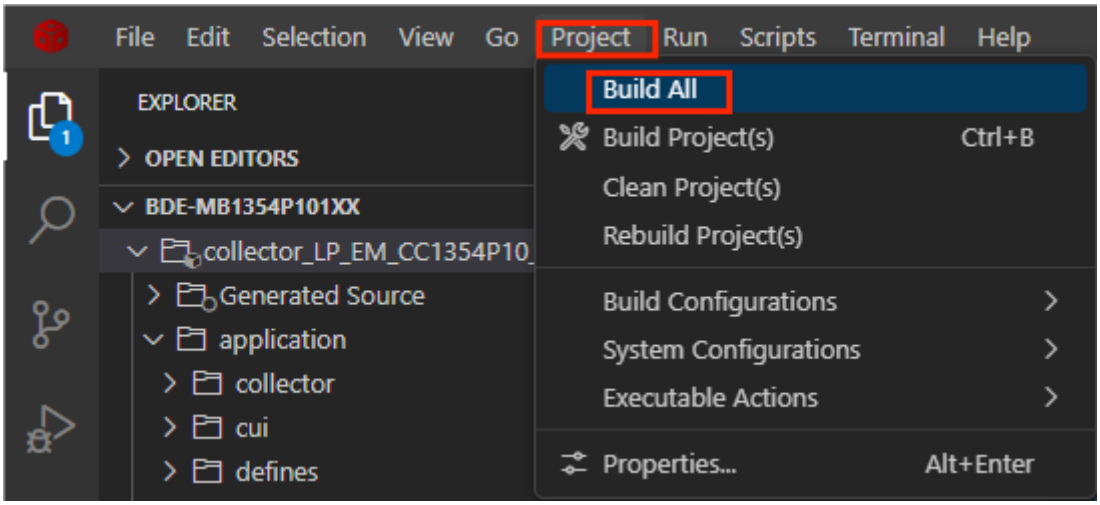
```
#include <stddef.h>
#include <stdint.h>
#ifndef DeviceFamily_CC13X4
#define DeviceFamily_CC13X4
#endif
#include <ti/devices/DeviceFamily.h>
#include "ti_drivers_config.h"
#include <ti/drivers/GPIO.h>
#include <ti/drivers/rf/RF.h>



/*
 * ======== Antenna switching ========
 */
/*
 * ======== rfDriverCallbackAntennaSwitching ========
 * Sets up the antenna switch depending on the current PHY configuration.
 * Fit BDE-MB13P2 module's switch
 *
 * Truth table:
 *
 * Path       DIO29 DIO30
 * ========== ===== =====
 * Off        0     0
 * TX         1     0
 * RX         0     1
 */
void rfDriverCallbackAntennaSwitching(RF_Handle client, RF_GlobalEvent events, void *arg)
{

    if (events & RF_GlobalEventRadioSetup) {
        bool    sub1GHz    = false;
        uint8_t loDivider = 0;
```

```
    /* Switch off all paths. */
    GPIO_write(CONFIG_RF_ANTENNA_PIN_SUB_1_G, 0);
    GPIO_write(CONFIG_RF_ANTENNA_PIN_HIGH_PA, 0);

    /* Decode the current PA configuration. */
    RF_TxPowerTable_PAType paType = (RF_TxPowerTable_PAType)RF_getTxPower(client).paType;

    /* Decode the generic argument as a setup command. */
    RF_RadioSetup* setupCommand = (RF_RadioSetup*)arg;

    switch (setupCommand->common.commandNo) {
        case (CMD_RADIO_SETUP):
        case (CMD_BLE5_RADIO_SETUP):
                loDivider = RF_LODIVIDER_MASK & setupCommand->common.loDivider;

                /* Sub-1GHz front-end. */
                if (loDivider != 0) {
                    sub1GHz = true;
                }
                break;
        case (CMD_PROP_RADIO_DIV_SETUP):
                loDivider = RF_LODIVIDER_MASK & setupCommand->prop_div.loDivider;

                /* Sub-1GHz front-end. */
                if (loDivider != 0) {
                    sub1GHz = true;
                }
                break;
        default:break;
    }

    if (sub1GHz) {
        /* Sub-1 GHz */
        if (paType == RF_TxPowerTable_HighPA) {
            /* PA enable --> HIGH PA
             * LNA enable --> Sub-1 GHz
             */
            /* Note: RFC_GPO3 is a work-around because the RFC_GPO1 (PA enable signal) is sometimes not
                     de-asserted on CC1352 Rev A. */
            GPIO_setConfigAndMux(CONFIG_RF_ANTENNA_PIN_SUB_1_G, GPIO_CFG_OUTPUT, IOC_PORT_RFC_GPO0);
            GPIO_setConfigAndMux(CONFIG_RF_ANTENNA_PIN_HIGH_PA, GPIO_CFG_OUTPUT, IOC_PORT_RFC_GPO3);
        } else {
            /* RF core active --> Sub-1 GHz */
            GPIO_setConfigAndMux(CONFIG_RF_ANTENNA_PIN_HIGH_PA, GPIO_CFG_OUTPUT, IOC_PORT_GPIO);
```

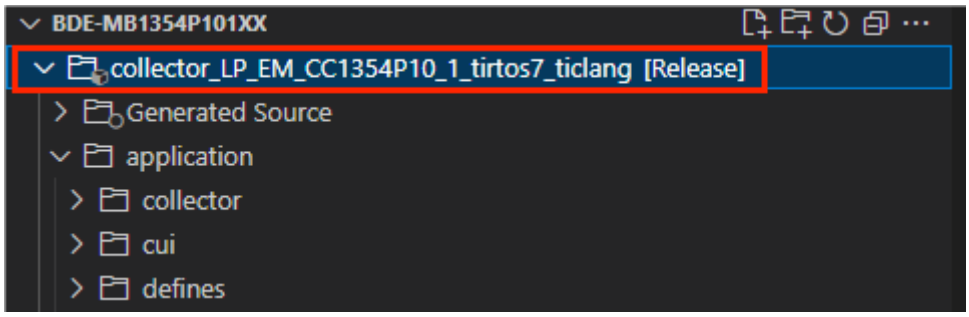
```
                GPIO_setConfigAndMux(CONFIG_RF_ANTENNA_PIN_SUB_1_G,    GPIO_CFG_OUTPUT    |    GPIO_CFG_OUT_HIGH,
IOC_PORT_GPIO);
            }
        }
    }
    else if (events & RF_GlobalEventRadioPowerDown) {
        /* Switch off all paths. */
        GPIO_write(CONFIG_RF_ANTENNA_PIN_SUB_1_G, 0);
        GPIO_write(CONFIG_RF_ANTENNA_PIN_HIGH_PA, 0);

        /* Reset the IO multiplexer to GPIO functionality */
        GPIO_setConfigAndMux(CONFIG_RF_ANTENNA_PIN_SUB_1_G, GPIO_CFG_OUTPUT, IOC_PORT_GPIO);
        GPIO_setConfigAndMux(CONFIG_RF_ANTENNA_PIN_HIGH_PA, GPIO_CFG_OUTPUT, IOC_PORT_GPIO);
    }
}
```

4.6.4. Build and Download Project

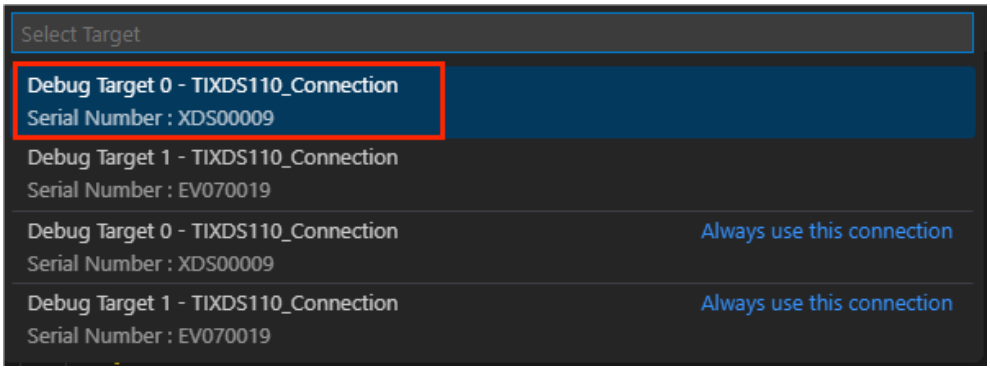
1. Select **Project** -> **Build All** in the toolbar menu. Both the “collect” and “sensor” projects will be built.



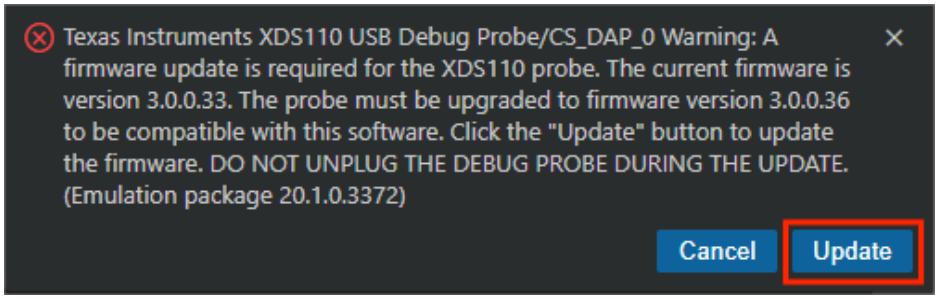
2. Select the “collector” project as the current active project by clicking the “collector” project root directory in the “Project Explorer” view.

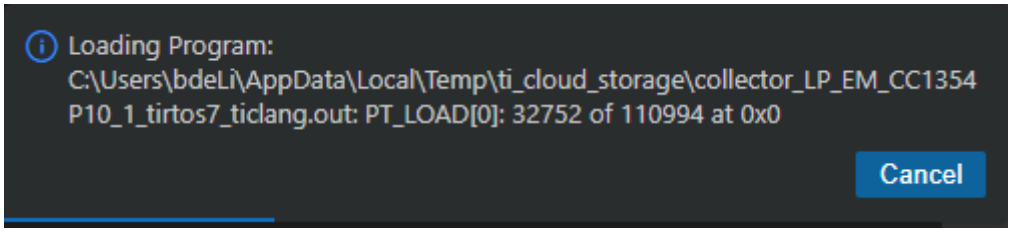


3. Select **Run** -> **Debug** to download the “collector” application to one of the two modules.

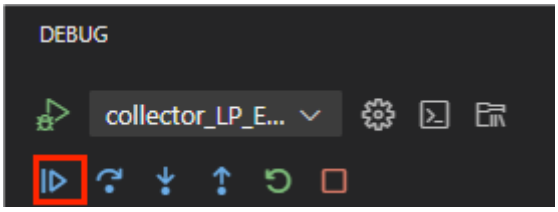


4. If the following message occur, click “Update”.

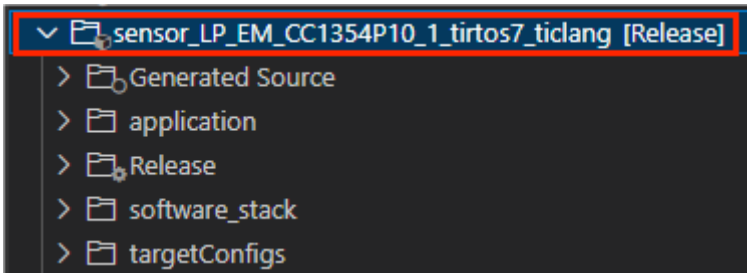




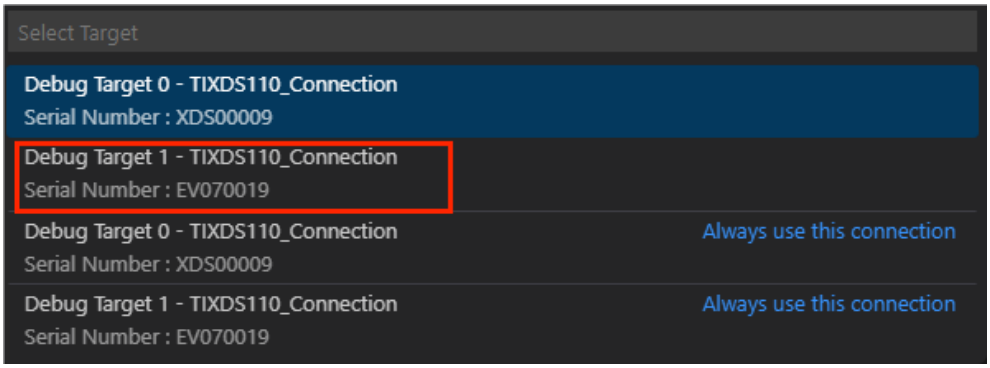
5. After the update process is finished, click “Continue”.



6. Select the “sensor” project as the current active project by clicking the “sensor” project root directory in the “Project Explorer” view.

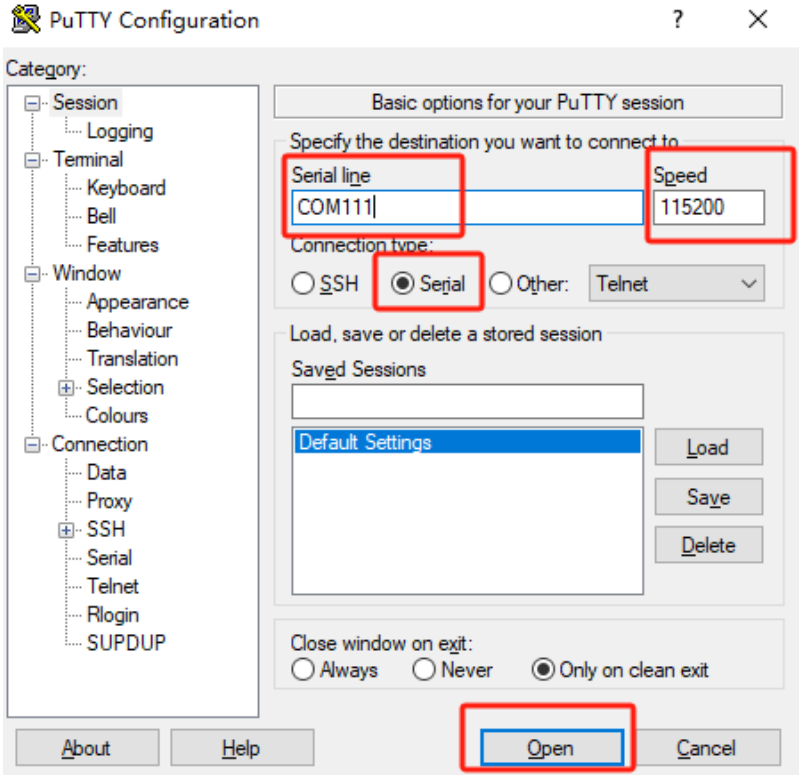


7. Select **Run** -> **Debug** to download the “sensor” application to the other module.

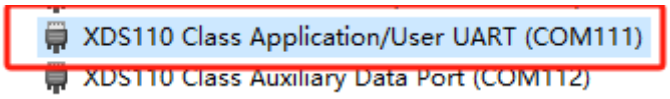


4.6.5. Running the demo

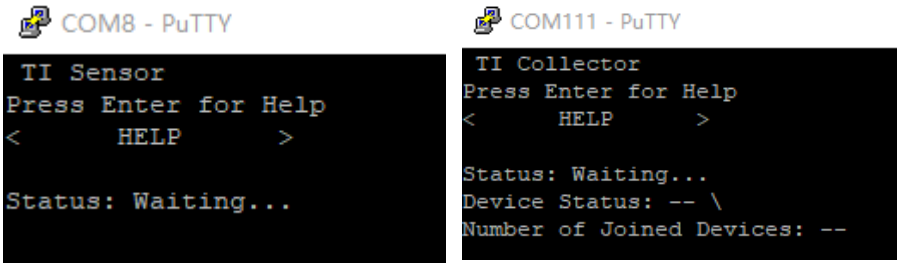
1. Open two instances of PuTTY. Open the correct serial ports separately using the following settings.



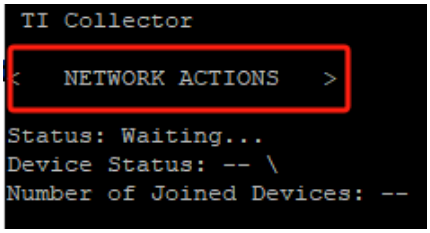
The COM number can be found in the System Device Manager:



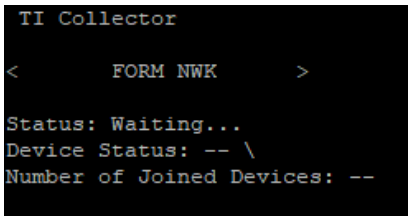
2. Reset the two modules by pressing the reset button on the BDE-XDS110 board. PuTTY will display the menu.



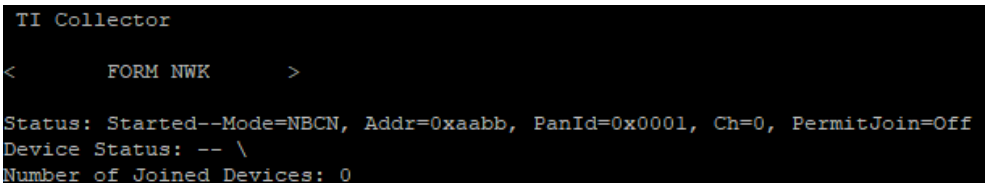
3. On the collector terminal side, press the right arrow key on the keyboard twice to switch to the “NETWORK ACTIONS” menu.



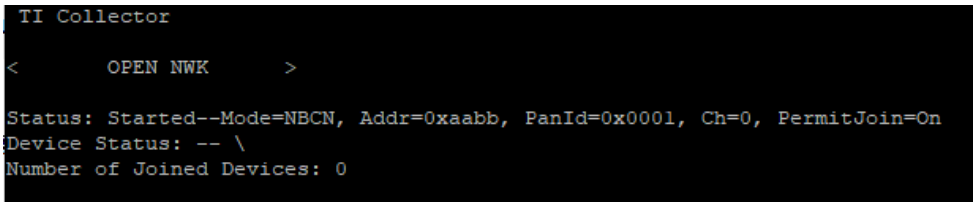
4. Press the enter key on the keyboard to open the “NETWORK ACTIONS” menu.



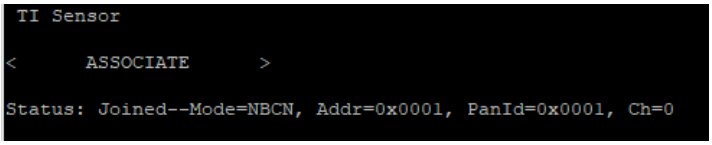
5. The “NETWORK ACTIONS” menu has three actions: “FORM NWK”, “OPEN NWK”, “CLOSE NWK”. You can switch between these actions by pressing the left/right arrow keys.
6. Switch to “FORM NWK”, then press the Enter key to form a TI 15.4 network. By default, the network forbids other devices from joining.

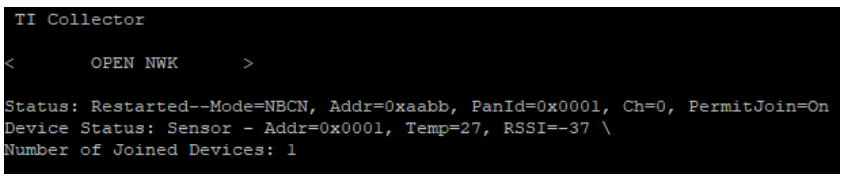


7. Switch to “OPEN NWK”, then press the Enter key to allow devices to join. At this time, the red LED on the collector board will blink.



8. On the sensor terminal side, open the “NETWORK ACTIONS” menu and perform the “ASSOCIATE” action. After a while, the sensor will join the network created by the collector. The red LED on the sensor board will turn on.



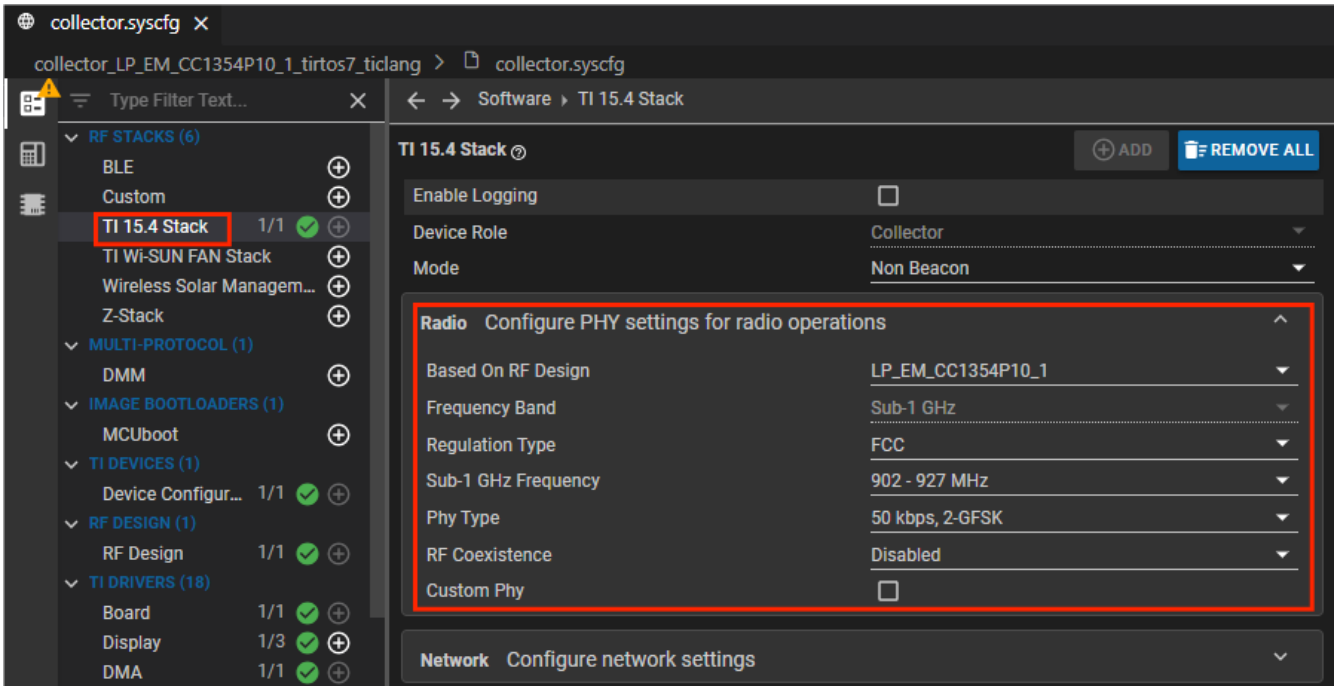


9. On the collector terminal side, press the Esc key to retun to the root menu, then switch to the “APP” menu and perform the “SEND TOGGLE” action. The red LED on the sensor board will toggle.

```
TI Collector

<      SEND TOGGLE      >

Status: Restarted--Mode=NBCN, Addr=0xaabb, PanId=0x0001, Ch=0, PermitJoin=On
Device Status: Sensor - Addr=0x0001, Temp=27, RSSI=-37 /
Number of Joined Devices: 1
```

4.6.6. Change to a Different PHY

The PHY settings are located under the ‘TI 15.4 Stack’ item in the syscfg file.



Developers can change the PHY settings according to their needs.

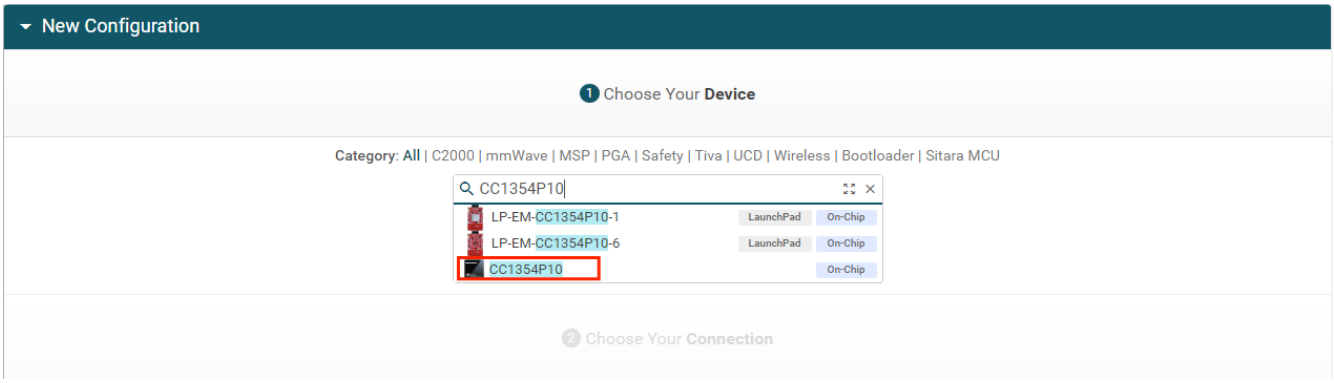
4.7. Use UniFlash to Download Firmware

In the section above, the method to flash the firmware is through debugging in the CCS IDE.

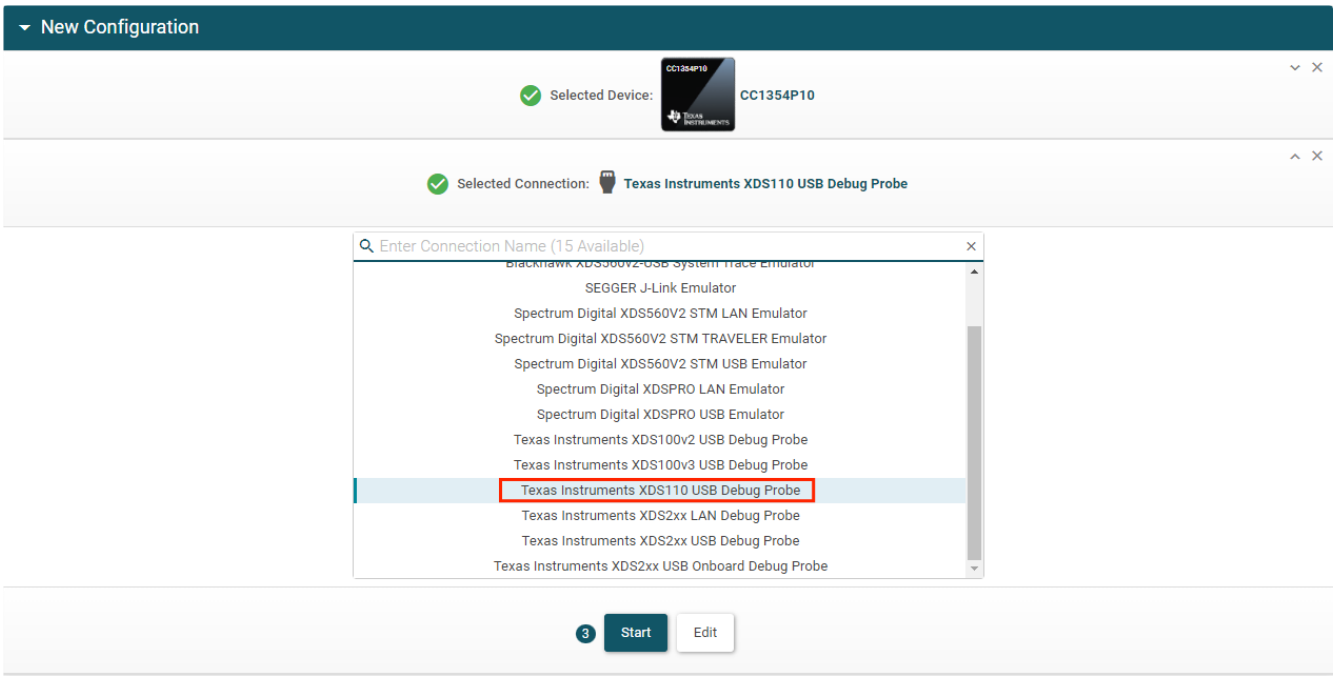
TI provides a software tool called UniFlash, which can be used for programming on-chip flash on TI microcontrollers and wireless connectivity devices.

4.7.1. Configure UniFlash

1. Open UniFlash.
2. Select “CC1354P10”.

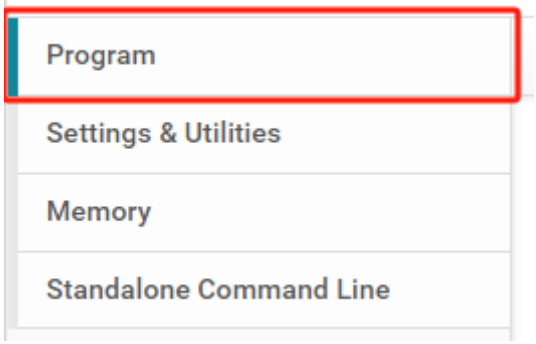


3. Select “Texas Instruments XDS110 USB Debug Probe”, then click the “Start” button.

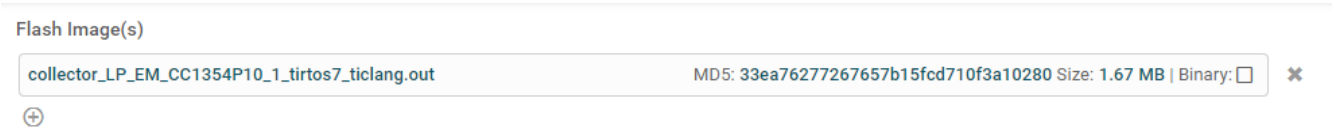


4.7.2. Programming

1. Select “Program” item.



2. Select the image file prepared for programming. By default, the image file is located in the Release/Debug folder within the project folder in the workspace.



3. Click the “Load Image” button to begin programming.

4. When programming is successful, the console will display the following information.

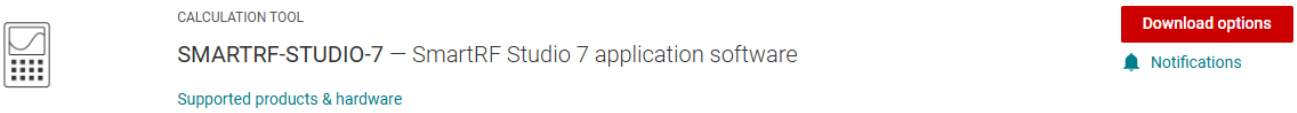
Please refer to the UniFlash quick start guide for more information about the UniFlash tool. Click the “Help” icon to open the quick start guide.

4.8. SmartRF™ Studio

SmartRF™ Studio is a Windows application that helps developers to easily evaluate the radio for all TI CC1xxx and CC2xxx low-power RF devices.

This tool can also be used for RF certification testing.

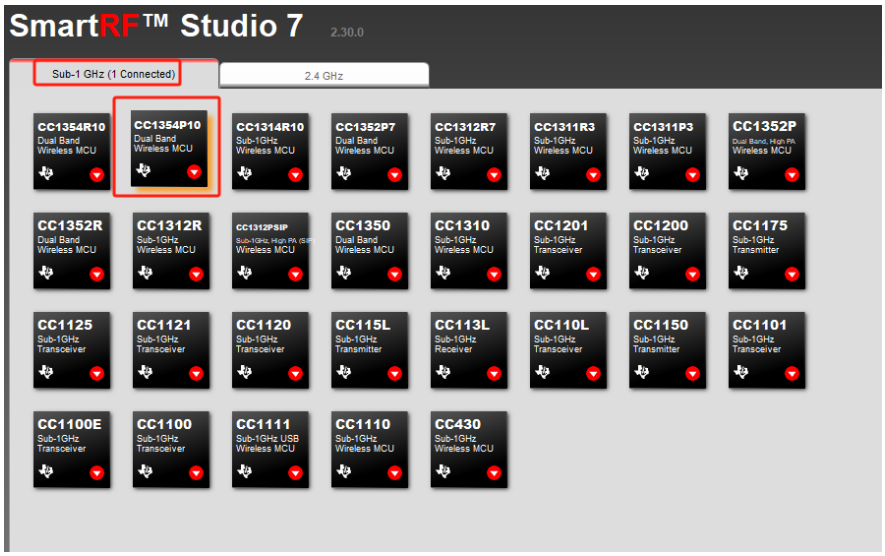
Visit the link below to download and install SmartRF™ Studio.
https://www.ti.com/tool/SMARTRFTM-STUDIO



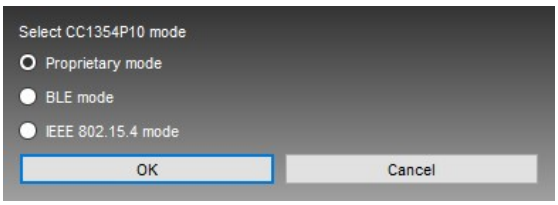
If the installer becomes locked during the download process, follow the instructions provided by TI.

4.8.1. Connect to the Module

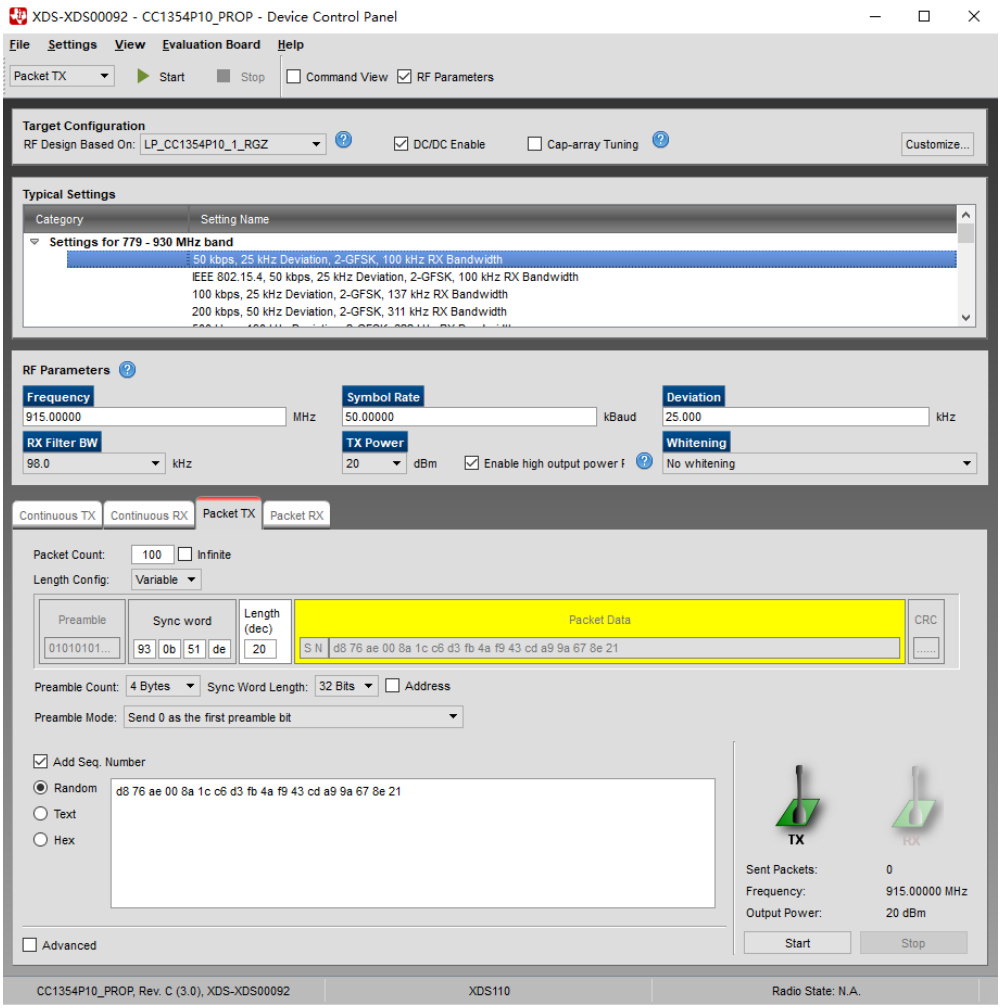
1. Connect the module to the PC using the XDS debug probe.
2. Open SmartRF Studio 7.
3. SmartRF Studio will show that one Sub-1 GHz device is connected.



4. Double-click the “CC1354P10” item.
5. Select the mode you want; in this example, the “Proprietary mode” is selected.

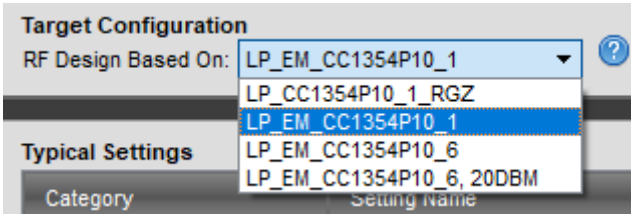


6. After clicking OK, SmartRF Studio will open a new window called “Device Control Panel”.

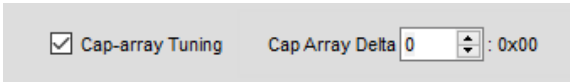


4.8.2. Change the Default Settings

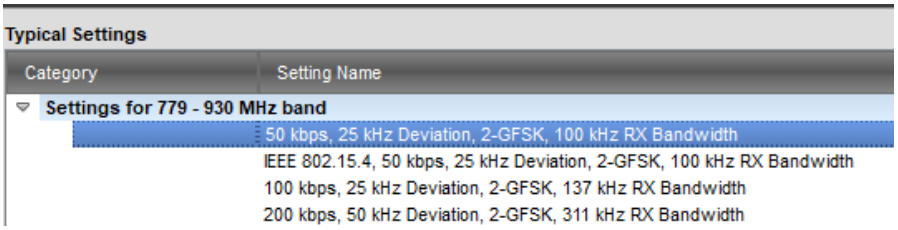
1. Select “LP_EM_CC1354P10_1”.



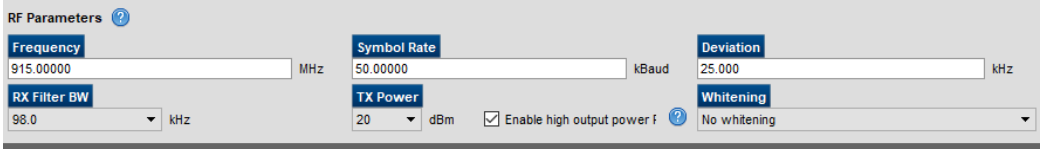
2. Select “Cap-array Tuning” and modify the “Cap Array Delta” to 0.



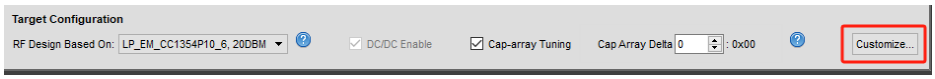
3. Select the PHY setting.



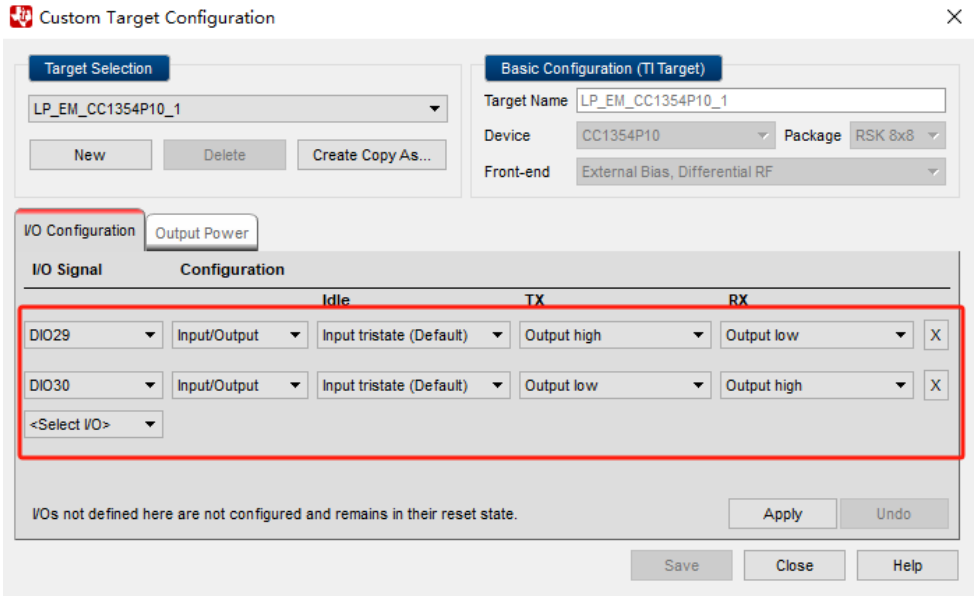
4. Select the “Frequency” and “TX Power”.



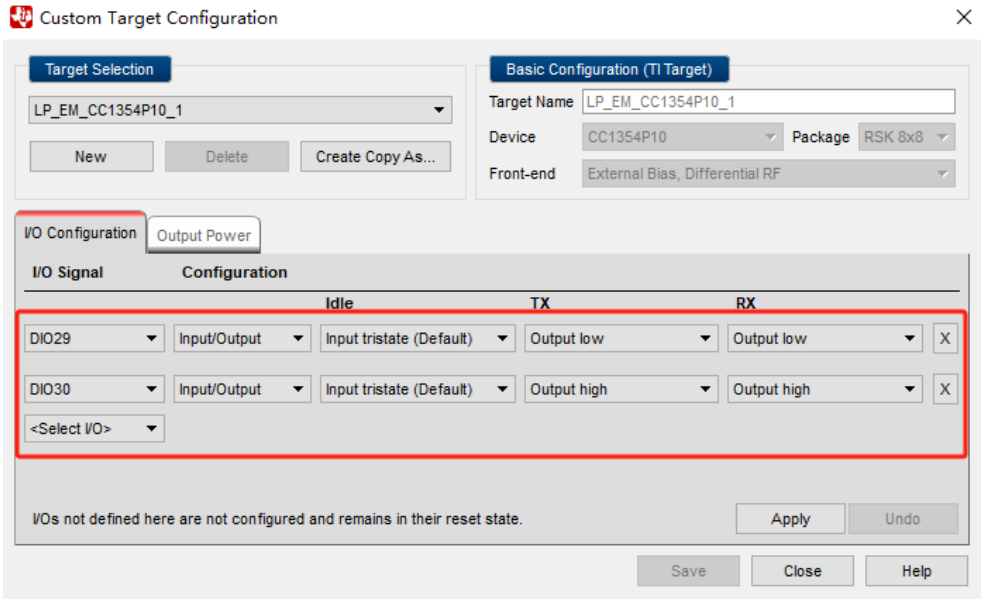
5. Click “Customize…” button to open the “Custom Target Configuration” window.



6. Modify the I/O configuration according to the following diagrams. Click “Apply” and then click “Close”.



I/O configuration with PA enabled

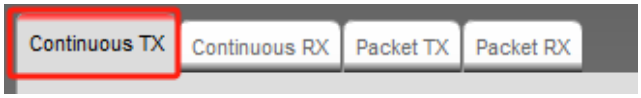


I/O configuration with PA disabled

4.8.3. Perform RF Test

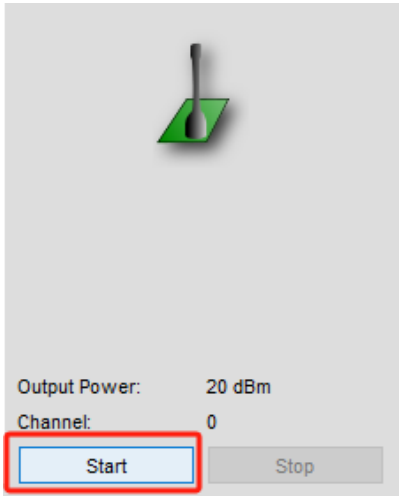
RF testing is divided into continuous TX/RX and packet TX/RX. The signal duty cycle is 100% when operating in continuous mode. The following steps use continuous TX as an example to demonstrate how to start the RF test.

1. Click “Continuous TX”.



2. Enable or disable modulation of the transmitted signal. When the “Unmodulated” radio box is selected, the signal is a carrier wave.

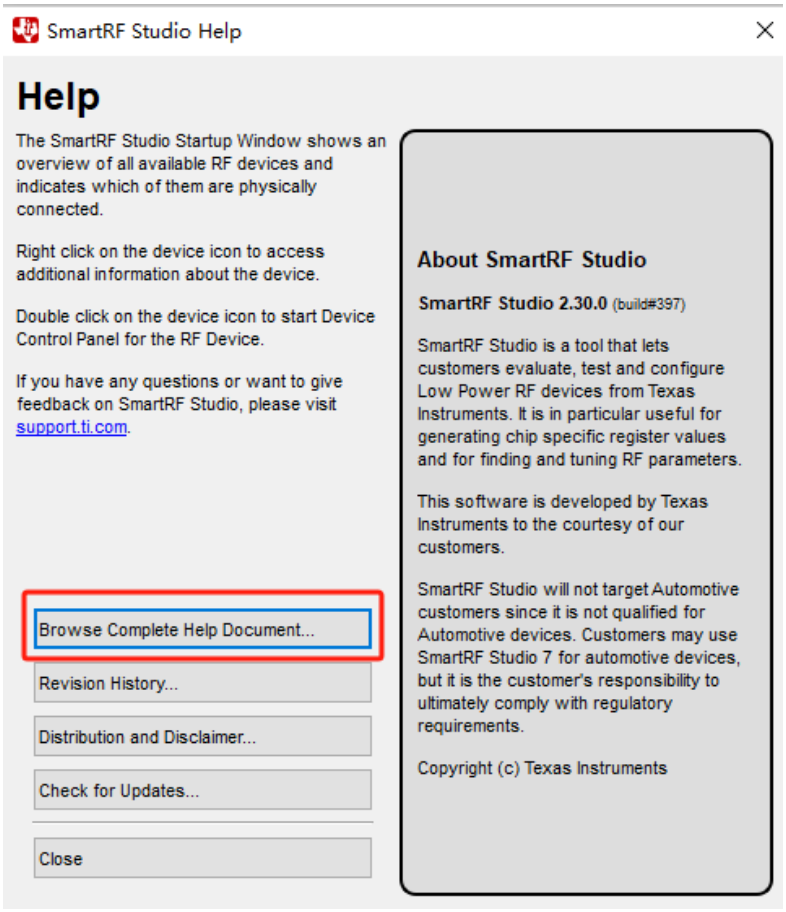
3. Click “Start” to start transmitting the RF signal.



NOTE: Each time the RF testing type or “Enable high output power PA” setting is changed, the I/O configuration must be reconfigured.

Please refer to the help document for more information about the SmartRF Studio tool. Click icon 

to open the “SmartRF Studio Help” window, then click the “Browse Complete Help Document…” button to access the help document.



5. Other Resources

Access more documents by visiting the TI CC1354P10 page at the link below.
https://www.ti.com/product/CC1354P10

Additionally, refer to the guides for different components for more information about the Simplelink SDK. The guides are located in the directory below:
C:\ti\simplelink_cc13xx_cc26xx_sdk_8_30_01_01\docs

For other related documentations of the module, please visit the module product page at bdecomm.com.

6. Ordering Information

Table 3. Ordering Information

Product Type	Orderable Part Number	Description
Evaluation Kit	BDE-MB1354P101xxx-K1	Evaluation kit 1 for module BDE-MB1354P101xxx, includes 1x BDE-MB13-BO, 1x BDE-LPEM, 1x BDE-XDS110, 1x USB type C cable. Refer to module datasheet for module variants. For example, BDE-MB1354P101UA0-K1 is the evaluation kit 1 for module BDE-MB1354P101UA0

Product Type	Orderable Part Number	Description
Module Breakout Board	BDE-MB13-BO	Breakout board with BDE-MB1354P101module
Interface Evaluation Module	BDE-LPEM	Launchpad evaluation module with BoosterPack interface
Debugger	BDE-XDS110	XDS110 debugger

7. Revision History

Table 4. Revision History

Revision	Date	Description
V1.0	2025-5-27	First release

8. Additional Information

8.1. Trademarks

Bluetooth® is a registered trademark of Bluetooth SIG, Inc.
Launchpad™ is a registered trademark of Texas Instruments.
BoosterPack™ is a registered trademark of Texas Instruments.
SmartRF™ is a registered trademark of Texas Instruments.
All trademarks are the property of their respective owners.

Important Notice and Disclaimer

The information contained herein is believed to be reliable. BDE makes no warranties regarding the information contain herein. BDE assumes no responsibility or liability whatsoever for any of the information contained herein. BDE assumes no responsibility or liability whatsoever for the use of the information contained herein. The information contained herein is provided "AS IS, WHERE IS" and with all faults, and the entire risk associated with such information is entirely with the user. All information contained herein is subject to change without notice. Customers should obtain and verify the latest relevant information before placing orders for BDE products. The information contained herein or any use of such information does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other intellectual property rights, whether with regard to such information itself or anything described by such information.

Contact

BDE Technology Inc.

USA: 67 E Madison St, # 1603A, Chicago, IL 60603, US
Tel: +1-312-379-9589

China: B2-403, 162 Science Avenue, Huangpu District, Guangzhou 510663, China
Tel: +86-20-28065335

Website: www.bdecomm.com Email: info@bdecomm.com