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 kits.

The BDE-BW33xx module series is built on TI's CC33xx devices (CC3301/CC3300 and CC3351/CC3350), offering a variety of connectivity options for users: 2.4GHz Wi-Fi 6 with or without BLE, 2.4GHz and 5GHz dual-band Wi-Fi 6 with or without BLE. These modules also come in multiple variants with different antenna options and operating temperature ranges. For detailed information on the available variants, please refer to the module datasheet.

This module series is ideal for cost-sensitive embedded applications with a Linux or RTOS host running TCP/IP, where the peak application throughput requirement is a maximum of 50 Mbps and includes BLE capabilities. It is an excellent choice for bringing the efficiency of Wi-Fi 6 to embedded device applications, featuring a small PCB footprint and a highly optimized bill of materials for lower costs. Additionally, the module is backward compatible with Wi-Fi 4 (802.11 b/g/n) and Wi-Fi 5 (802.11 ac).

We offer an evaluation module, BDE-BW33-EM, along with breakout boards specific to each module, which can be interfaced with the BDE-BW33-EM for easy testing. Additionally, M.2 cards with a Key E form factor are available for certain modules, allowing for evaluation with boards that have an M.2 interface.

2. Evaluation Module - BDE-BW33-EM

The BDE-BW33-EM is a plug-in evaluation module, it can be the test and development board for different modules when stacked with breakout boards with the target modules. It can be connected to TI LaunchPad directly or processor boards through an adaptor board for rapid software development.

This kit can be used in below configurations:

(1) MCU and RTOS evaluation:
- BDE-BW33-EM + BDE-BW3301NP1-BO (Breakout board with BDE-BW3301NP1 module, can be different depending on the target module) + LP-AM243 (LaunchPad with the MCU running TCP/IP);

(2) Processor and Linux evaluation:
- BDE-BW33-EM + BDE-BW3301NP1-BO (Breakout board with BDE-BW3301NP1 module, can be different depending on the target module) + BDE-BW33-BBB (BeagleBone Black adaptor board) + BEAGL-BONE-BLACK;

(3) RF-testing with PC tools:
- BDE-BW33-EM + BDE-BW3301NP1-BO (Breakout board with BDE-BW3301NP1 module, can be different depending on the target module) + BDE-XDS110;

(4) Other MCU or MPU platform:
- This kit can also be used with other MCU/MPU boards for evaluation. The adaptor board might be needed for connecting the board. And also, software porting will be needed for other MCU/MPU platforms.

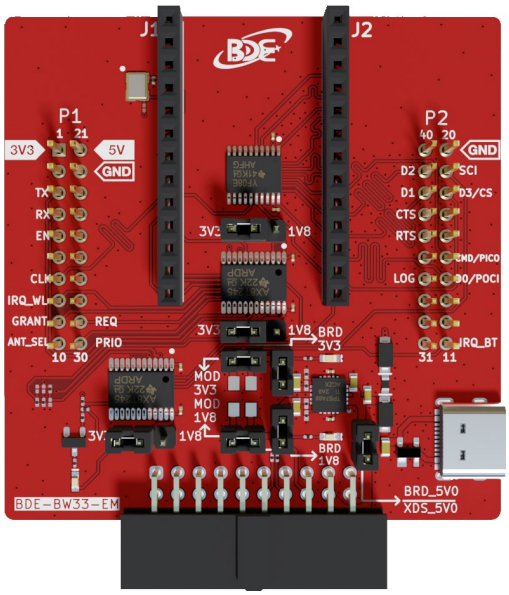


Figure 1. The Photo of BDE-BW33-EM

Figure 2. The Photo of the Breakout Boards with Module

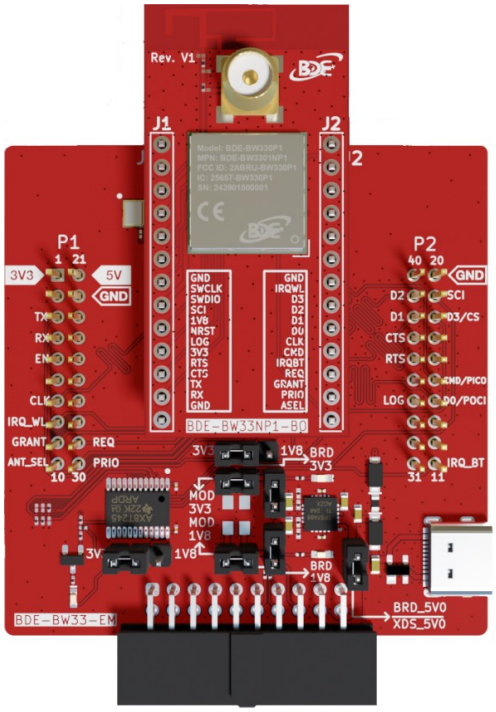


Figure 3. The Photo of BDE-BW33-EM Stacked with the Breakout Board with Module

2.1. Key Features

- Two 20-pin stackable connectors (TI BoosterPack™ standard)
- Power from on-board dual-rail (3.3V and 1.8V) LDO using USB or LaunchPad or both together
- Three level shifters for voltage translation (3.3V to 1.8V)
- JTAG header pins for SWD interface with XDS1110 debugger
- Jumpers for current measurement on both power supplies (3.3V and 1.8V) with provision to mount
- 0.1-ohm (0805) resistors for measurement with voltmeter
- 32kHz oscillator for lower power evaluation

2.2. Block Diagram

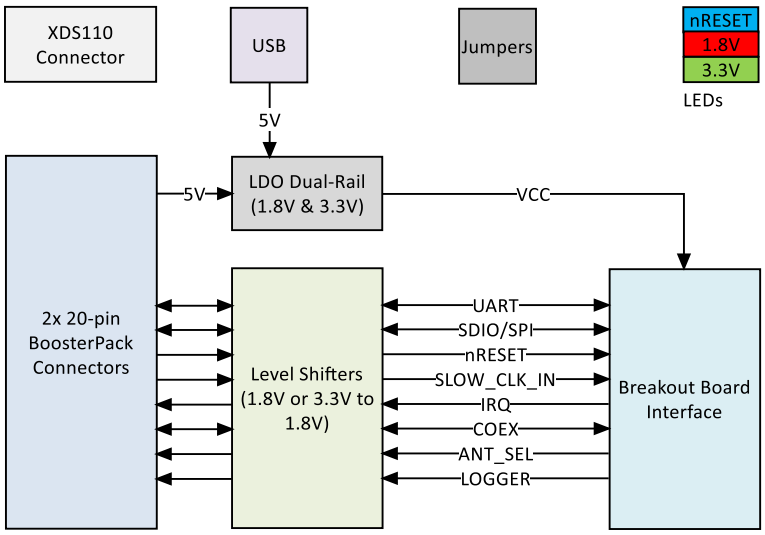


Figure 4. The Block Diagram of BDE-BW33-EM

2.3. Hardware Overview

Figure 5 shows the overview of the BDE-BW33-EM.

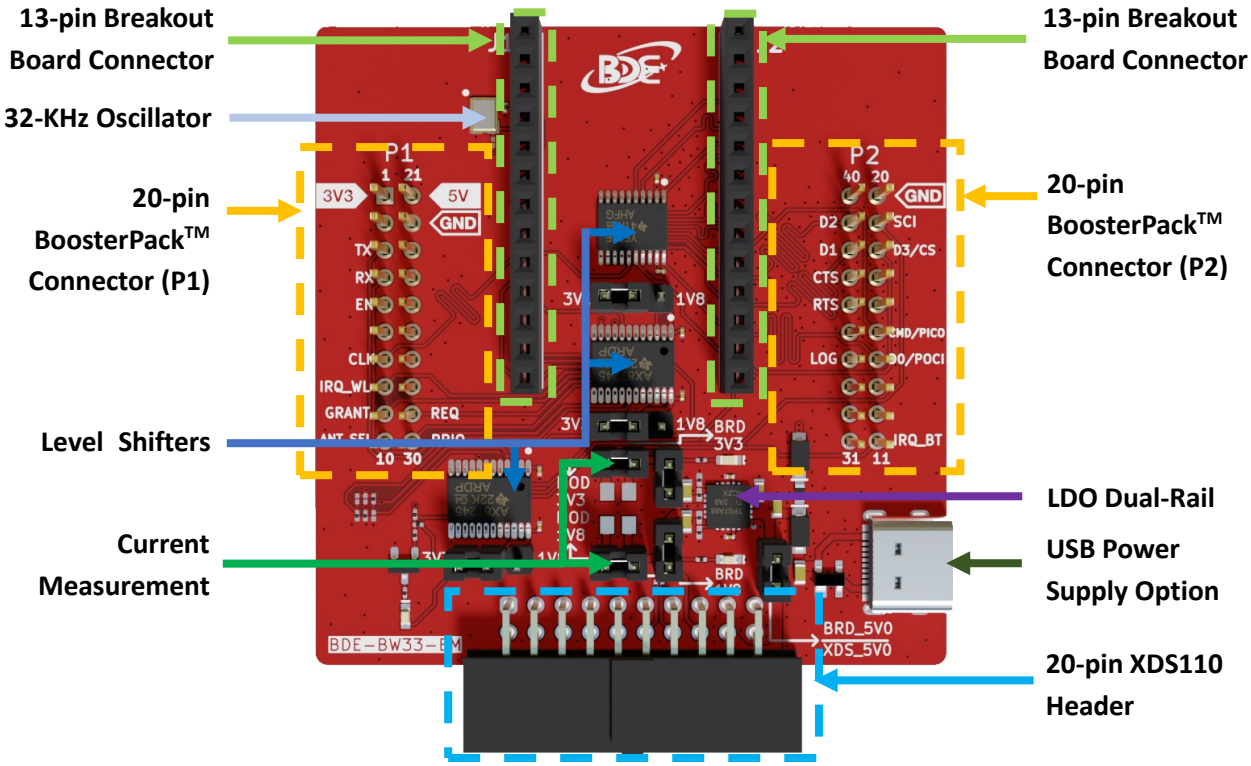


Figure 5. BDE-BW33-EM Overview

2.4. LED Indicators

Table 1 lists the LED descriptions.

Table 1. LED Descriptions

Reference	Color	Usage	Comments
LED1	Red	BRD_1V8 power indication	On: BRD_1V8 power rail is up Off: no BRD_1V8 power supplied
LED2	Green	BRD_3V3 power indication	On: BRD_3V3 power rail is up Off: no BRD_3V3 power supplied
LED3	Blue	nRESET indication	The LED indicates the state of the nRESET pin. If that LED is on, the device is functional which means the nRESET is high

2.5. Jumper Settings

Figure 6 shows default jumper settings.

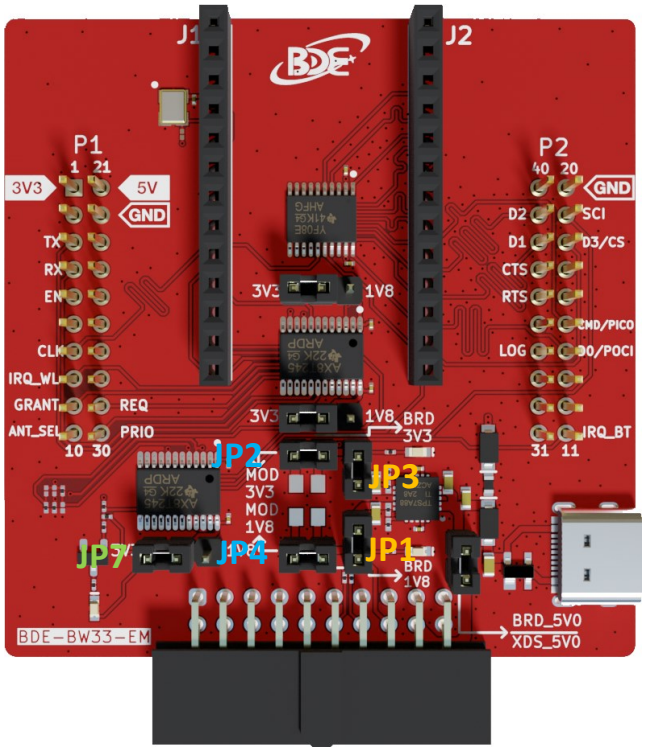


Figure 6. BDE-BW33-EM Default Jumper Settings

Table 2 shows jumper setting descriptions.

Table 2. Jumper Setting Descriptions

Reference	Usage	Comments
JP1, JP3	Power to board	Used to enable power to board for both supplies. See Section 2.9
JP2, JP4	Current measurement	Used to measure power to module only. See Section 2.9
JP8	USB power	Power the board with USB
JP5, JP6, JP7	Level shifter host voltage	Set to 3.3V or 1.8V to enable relevant level shifters to translate to correct host voltage level

2.6. BoosterPack Header Assignment

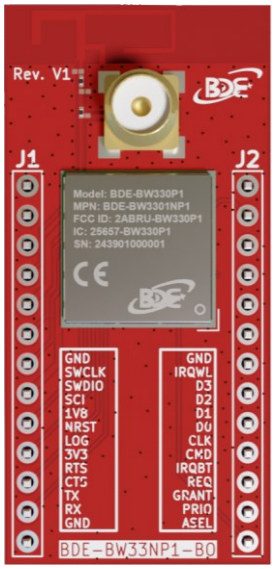
Table 3 shows the pin assignment for BoosterPack headers P1 and P2.

Table 3. BoosterPack Headers Pin Assignment

Pin Number	Pin Name	Type/Direction	Description
P1.1	VCC_MCU_3V3	Input	No functional purpose
P1.2	Reserved	N/A	N/A
P1.3	UART_TX_3V3 (from module)	Output	Module UART TX to host for BLE host controller interface
P1.4	UART_RX_3V3 (to module)	Input	Module UART RX from host for BLE host controller interface
P1.5	LP_RESET	Input	Reset line for module, used to enable/disable (active low). Driven by host through LaunchPad pins
P1.6	Reserved	N/A	N/A
P1.7	SDIO_CLK_3V3 (CLK)	Input	SDIO clock or SPI clock. Must be driven by host

Pin Number	Pin Name	Type/Direction	Description
P1.8	IRQ_WL_3V3	Output	Interrupt request from module to host for Wi-Fi activity
P1.9	COEX_GRANT_3V3	Output	External coexistence interface - grant (reserved for future use)
P1.10	ANT_SEL_3V3	Output	Antenna select control, not available on module variants with antenna diversity
P1.21	VCC_MCU_5V	Power	5V supply to board
P1.22	GND	GND	Board ground
P1.23	Reserved	N/A	N/A
P1.24	Reserved	N/A	N/A
P1.25	Reserved	N/A	N/A
P1.26	Reserved	N/A	N/A
P1.27	Reserved	N/A	N/A
P1.28	Reserved	N/A	N/A
P1.29	COEX_REQ_3V3	Input	External coexistence interface – request (reserved for future use)
P1.30	COEX_PRIORITY_3V3	Input	External coexistence interface - priority (reserved for future use)
P2.11	IRQ_BLE_3V3	Output	Interrupt request from module to host for BLE activity in shared SDIO mode
P2.12	Reserved	N/A	N/A
P2.13	Reserved	N/A	N/A
P2.14	SDIO_D0_3V3 (POCI)	Input/Output	SDIO data D0 or SPI POCI
P2.15	SDIO_CMD_3V3 (PICO)	Input/Output	SDIO command or SPI PICO
P2.16	Reserved	N/A	N/A
P2.17	Reserved	N/A	N/A
P2.18	SDIO_D3_3V3 (CS)	Input/Output	SDIO data D3 or SPI CS
P2.19	SLOW_CLK_IN_3V3	Input	Input for external RTC clock 32.768kHz
P2.20	GND	GND	Board ground
P2.31	Reserved	N/A	N/A
P2.32	Reserved	N/A	N/A
P2.33	Reserved	N/A	N/A
P2.34	LOGGER_3V3	Output	Tracer from module (UART TX debug logger)
P2.35	Reserved	N/A	N/A
P2.36	UART_RTS_3V3 (from module)	Output	UART RTS from module to host for BLE HCI flow control
P2.37	UART_CTS_3V3 (to module)	Input	UART CTS to module from host for BLE HCI flow control
P2.38	SDIO_D1_3V3	Input/Output	SDIO data D1
P2.39	SDIO_D2_3V3	Input/Output	SDIO data D2
P2.40	Reserved	N/A	N/A

2.7. Breakout Board Header Assignment



1: GND
2: SWCLK_1V8
3: SWDIO_1V8
4: SLOW_CLK_IN_1V8
5: MOD_1V8
6: nRESET_1V8
7: LOGGER_1V8
8: MOD_3V3
9: UART_RTS_1V8
10: UART_CTS_1V8
11: UART_TX_1V8
12: UART_RX_1V8
13: GND

1: GND
2: IRQ_WL_1V8
3: SDIO_D3_1V8 (CS)
4: SDIO_D2_1V8
5: SDIO_D1_1V8
6: SDIO_D0_1V8 (POCI)
7: SDIO_CLK_1V8 (CLK)
8: SDIO_CMD_1V8 (PICO)
9: IRQ_BLE_1V8
10: COEX_REQ_1V8
11: COEX_GRANT_1V8
12: COEX_PRIORITY_1V8
13: ANT_SEL_1V8

Figure 7. Breakout Board Headers (J1, J2)

Table 4 shows the pin assignment for the breakout board headers J1 and J2.

Table 4. Breakout Board Headers Pin Assignment

Pin Number	Pin Name	Type/Direction	Description
J1.1	GND	GND	Board ground
J1.2	SWCLK_1V8	Input	Serial wire clock
J1.3	SWDIO_1V8	Input/Output	Serial wire data input/output
J1.4	SLOW_CLK_IN_1V8	Input	Input for external RTC clock 32.768kHz
J1.5	MOD_1V8	Power	1.8V module power supply
J1.6	nRESET_1V8	Input	Reset line for module, used to enable/disable (active low)
J1.7	LOGGER_1V8	Output	Tracer from module (UART TX debug logger)
J1.8	MOD_3V3	Power	3.3V module power supply
J1.9	UART_RTS_1V8	Output	UART RTS from module to host for BLE HCI flow control
J1.10	UART_CTS_1V8	Input	UART CTS to module from host for BLE HCI flow control
J1.11	UART_TX_1V8	Output	Module UART TX to host for BLE host controller interface
J1.12	UART_RX_1V8	Input	Module UART RX from host for BLE host controller interface
J1.13	GND	GND	Board ground
J2.1	GND	GND	Board ground
J2.2	IRQ_WL_1V8	Output	Interrupt request from module to host for Wi-Fi activity
J2.3	SDIO_D3_1V8 (CS)	Input/Output	SDIO data D3 or SPI CS
J2.4	SDIO_D2_1V8	Input/Output	SDIO data D2
J2.5	SDIO_D1_1V8	Input/Output	SDIO data D1
J2.6	SDIO_D0_1V8 (POCI)	Input/Output	SDIO data D0 or SPI POCI

Pin Number	Pin Name	Type/Direction	Description
J2.7	SDIO_CLK_1V8 (CLK)	Input	SDIO clock or SPI clock. Must be driven by host
J2.8	SDIO_CMD_1V8 (PICO)	Input/Output	SDIO command or SPI PICO
J2.9	IRQ_BLE_1V8	Output	Interrupt request from module to host for BLE activity in shared SDIO mode
J2.10	COEX_REQ_1V8	Input	External coexistence interface – request
J2.11	COEX_GRANT_1V8	Output	External coexistence interface - grant
J2.12	COEX_PRIORITY_1V8	Input	External coexistence interface - priority
J2.13	ANT_SEL_1V8	Output	Antenna select control, not available on antenna diversity module variants

2.8. XDS110/JTAG Header Pin Assignment

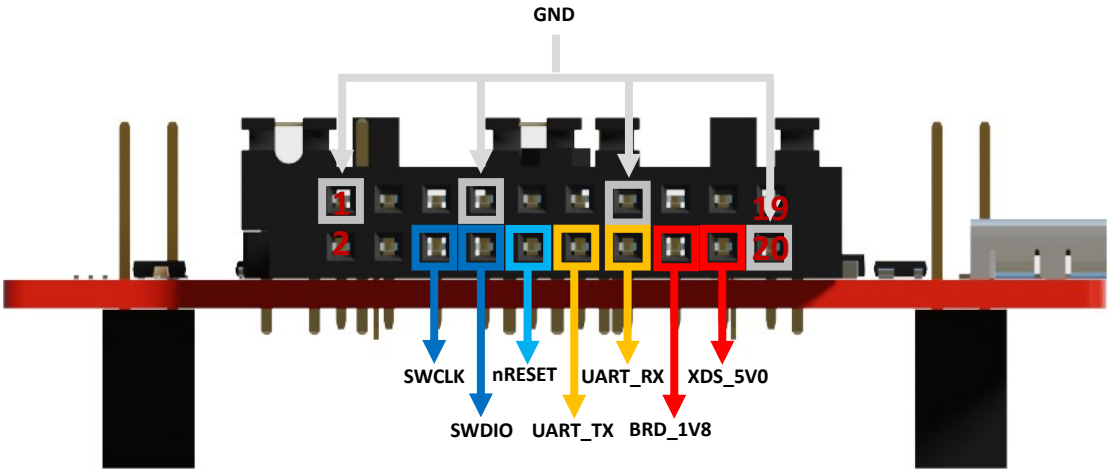


Figure 8. XDS110/JTAG Headers (J3)

Table 5 shows the pin assignment for the XDS110/JTAG header J3.

Table 5. XDS110/JTAG Header Pin Assignment

Pin Number	Pin Name	Type/Direction	Description
J3.1, J3.7, J3.13, J3.20	GND	GND	Board ground
J3.6	SWCLK_1V8	Input	Serial wire clock
J3.8	SWDIO_1V8	Input/Output	Serial wire data input/output
J3.10	nRESET_1V8	Input	Reset line for module, used to enable/disable (active low)
J3.12	UART_TX_1V8	Output	Module UART TX to host for BLE host controller interface, no connect to this pin by default
J3.14	UART_RX_1V8	Input	Module UART RX from host for BLE host controller interface, no connect to this pin by default
J3.16	BRD_1V8	Power	1.8V supply for reference voltage to connector, it tells the debugger the voltage level for debug signal is 1.8V
J3.18	XDS_5V0	Power	5V supply from XDS110 debugger

2.9. Power

The board is designed to accept power from a connected LaunchPad kit. Some LaunchPad kits cannot source the peak current requirements for Wi-Fi, which can be as high as 500mA. In such cases, the USB connector (USB1) on the BDE-BW33-EM can be used to aid in extra current. The use of Schottky diodes make sure that load sharing occurs between the USB connectors on the LaunchPad kit and the BoosterPack module without any board modifications. The jumpers labeled JP1 (BRD_1V8) and JP3 (BRD_3V3) can be used to measure the total current consumption of the board from the onboard LDO.

2.9.1. Measure the Module Current Draw

2.9.1.1. Low Current Measurement (LPDS)

To measure the current draw of the module for both power supplies (3.3V or 1.8V), a jumper labeled JP2 (for 3.3V supply) and a jumper labeled JP4 (for 1.8V supply) is provided on the board. By removing JP2, users can place an ammeter into this path to observe the current on the 3.3V supply. The same process can be used for observing the current on the 1.8V supply with JP4. BDE recommends this method for measuring the LPDS.

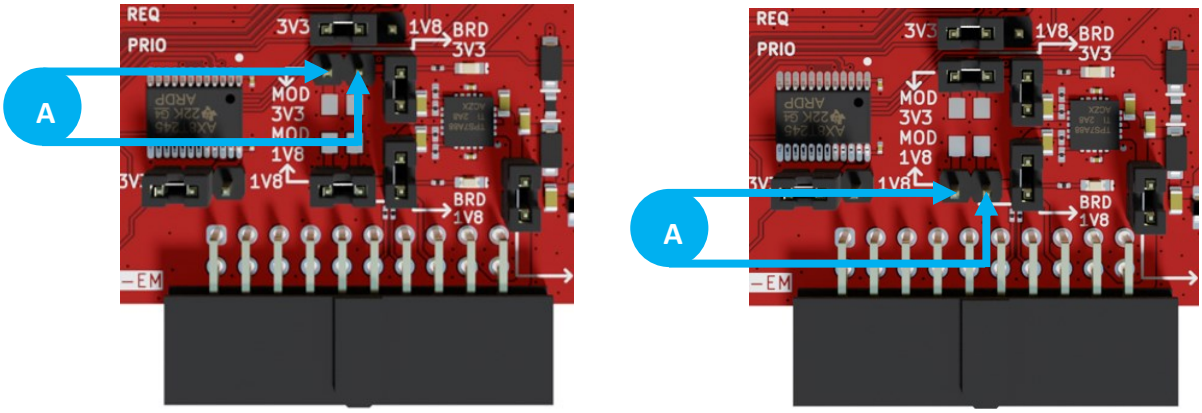


Figure 9. Measure LPDS Current with Ammeter

2.9.1.2. Active Current Measurement

To measure active current in a profile form, BDE recommends using a 0.1Ω 1% 0805 resistor on the board, and measuring the differential voltage across the resistor. This can be done using a voltmeter or an oscilloscope for measuring the current profile for both power supplies (3.3V or 1.8V). Jumper JP4 shunt is removed and a 0.01 resistor is populated in parallel to measure the active currents on the 1.8V supply. Similar operation with JP2 and 3.3V supply.

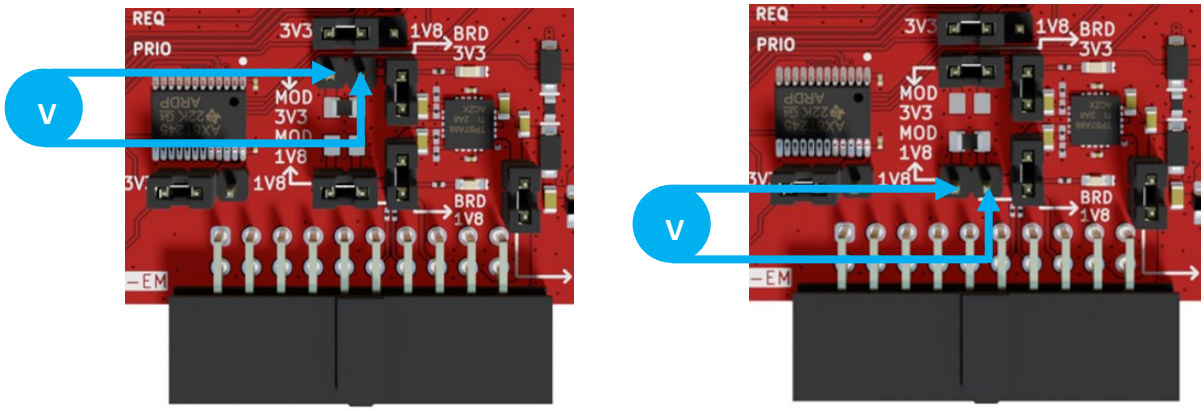


Figure 10. Measure Active Current with Voltmeter

3. M.2 Card

Besides BDE-BW33-EM, we also offer M.2 cards with Key E form factor for certain modules to enable evaluation with boards that have M.2 interface.

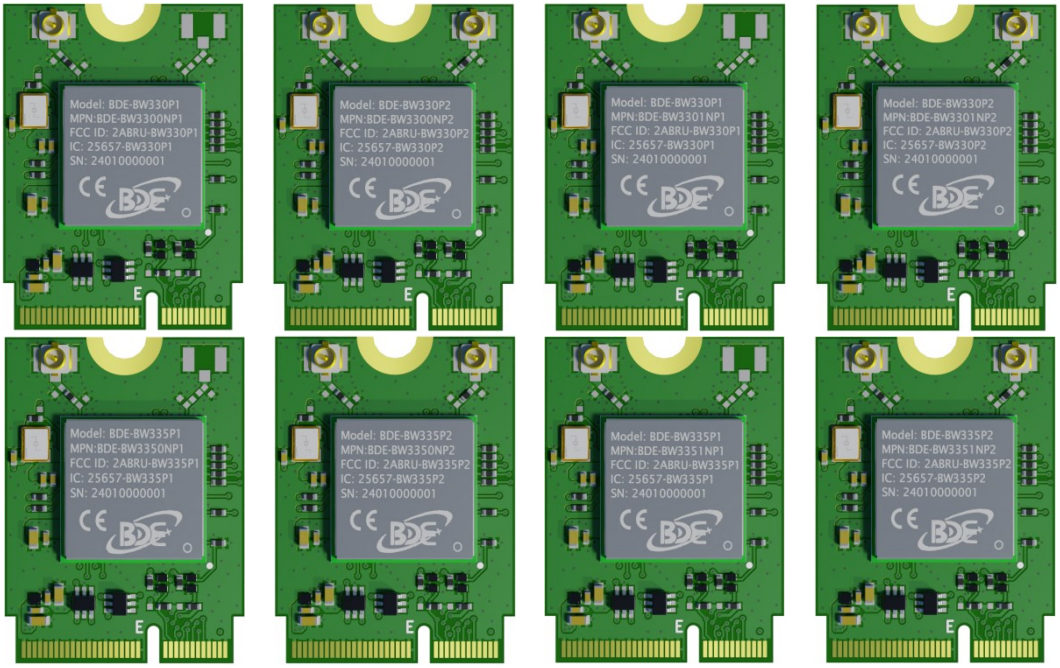


Figure 11. The Photo of M.2 Cards

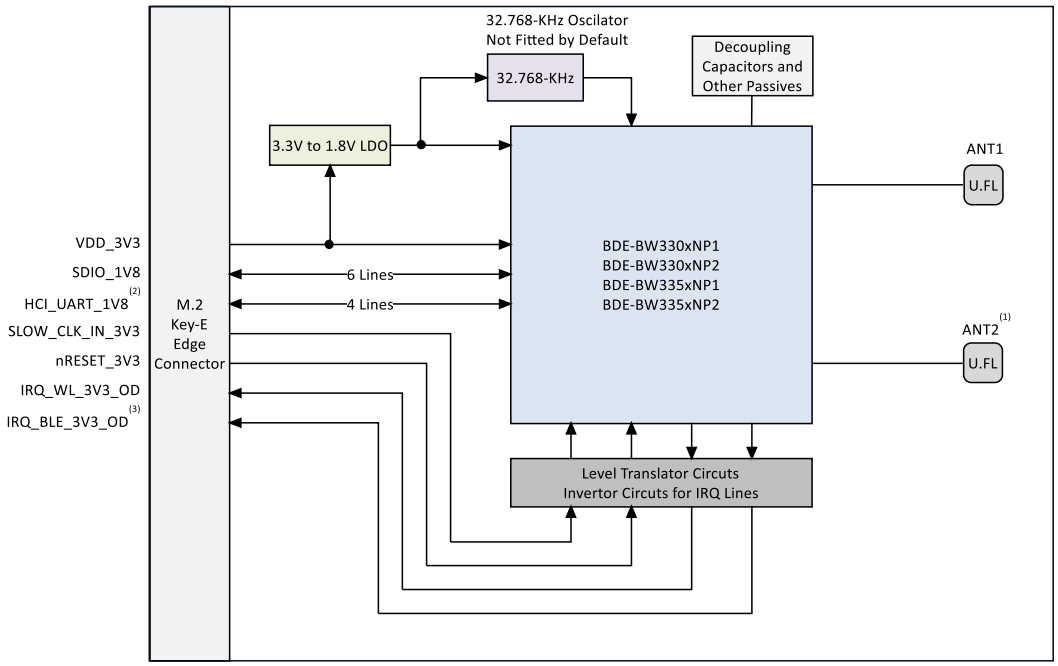
Table 6 lists the available M.2 cards for the modules.

Table 6. Available M.2 Cards

Orderable Part Number	Descriptions
BDE-BW3301NP1M2	2.4G Wi-Fi 6 and BLE Combo, single antenna port, assembled with BDE-BW3301NP1 module
BDE-BW3300NP1M2	2.4G Wi-Fi 6, single antenna port, assembled with BDE-BW3300NP1 module
BDE-BW3301NP2M2	2.4G Wi-Fi 6 and BLE Combo, dual antenna port with antenna diversity, assembled with BDE-BW3301NP2 module
BDE-BW3300NP2M2	2.4G Wi-Fi 6, dual antenna port with antenna diversity, assembled with BDE-BW3300NP2 module

Orderable Part Number	Descriptions
BDE-BW3351NP1M2	2.4G & 5G dual-band Wi-Fi 6 and BLE Combo, single antenna port, assembled with BDE-BW3351NP1 module
BDE-BW3350NP1M2	2.4G & 5G dual-band Wi-Fi 6, single antenna port, assembled with BDE-BW3350NP1 module
BDE-BW3351NP2M2	2.4G & 5G dual-band Wi-Fi 6 and BLE Combo, dual antenna port with antenna diversity, assembled with BDE-BW3351NP2 module
BDE-BW3350NP2M2	2.4G & 5G dual-band Wi-Fi 6, dual antenna port with antenna diversity, assembled with BDE-BW3350NP2 module

3.1. Block Diagram



(1) ANT2 is only available for module variants with antenna diversity feature;
(2) HCI_UART_1V8 is only available for module variants with BLE;
(3) IRQ_BLE_3V3_OD is only needed for BLE in shared SDIO mode.

Figure 12. The Block Diagram of M.2 Card

3.2. Pin Assignment

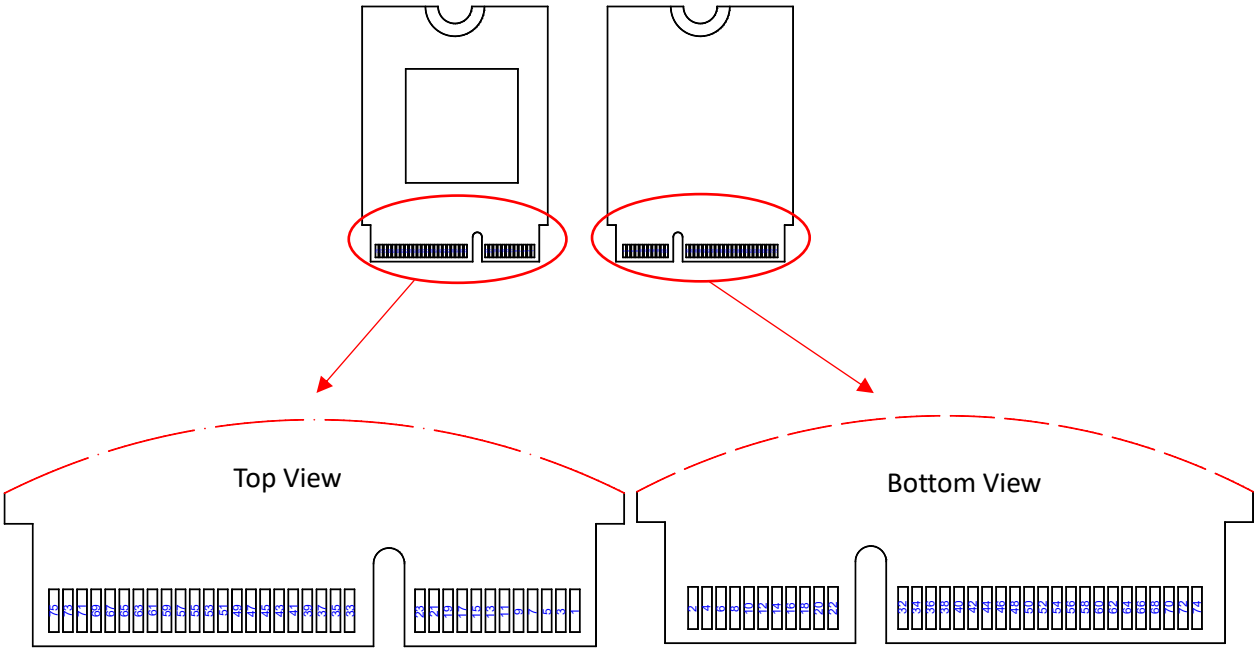


Figure 13. Pin Diagram of the M.2 Card

Table 7 shows the pin assignment of the M.2 card.

Table 7. M.2 Card Pin Assignment

Pin Number	Pin Name	Type/Direction	Description
1	GND	GND	Board ground
2	VDD_3V3	Power	3.3V power supply
3	NC	N/A	No connect
4	VDD_3V3	Power	3.3V power supply
5, 6	NC	N/A	No connect
7	GND	GND	Board ground
8	NC	N/A	No connect
9	SDIO_CLK_1V8	Input	SDIO_CLK_WL (SPI_CLK), 1.8V level, must be driven by host
10	NC	N/A	No connect
11	SDIO_CMD_1V8	Input	SDIO_CMD_WL (SPI_DIN), 1.8V level
12	NC	N/A	No connect
13	SDIO_D0_1V8	Input/Output	SDIO_D0_WL (SPI_DOUT), 1.8V level
14	NC	N/A	No connect
15	SDIO_D1_1V8	Input/Output	SDIO_D1_WL, 1.8V
16	NC	N/A	No connect
17	SDIO_D2_1V8	Input/Output	SDIO_D2_WL, 1.8V
18	GND	Ground	Board ground
19	SDIO_D3_1V8	Input/Output	SDIO_D3_WL (SPI_CS), 1.8V level

Pin Number	Pin Name	Type/Direction	Description
20	IRQ_BLE_3V3_OD	Output	IRQ_BLE to host (in shared SDIO mode), open drain, need pull-up resistor to 3.3V
21	IRQ_WL_3V3_OD	Output	IRQ_WL to host, open drain, need pull-up resistor to 3.3V
22	UART_TX_1V8	Output	UART TX for BLE HCI, 1.8V level
32	UART_RX_1V8	Input	UART RX for BLE HCI, 1.8V level
33	GND	Ground	Board ground
34	UART_RTS_1V8	Output	UART RTS for flow control for BLE HCI, 1.8V level
35	NC	N/A	No connect
36	UART_CTS_1V8	Input	UART CTS for flow control for BLE HCI, 1.8V level
37, 38	NC	N/A	No connect
39	GND	Ground	Board ground
40, 41, 42, 43, 44	NC	N/A	No connect
45	GND	Ground	Board ground
46, 47, 48, 49	NC	N/A	No connect
50	SLOW_CLK_IN_3V3	Input	External 32.768-KHz slow clock input, 3.3V level
51	GND	Ground	Board ground
52, 53, 54, 55	NC	N/A	No connect
56	nRESET_3V3	Input	Reset line for enabling or disabling device (active low), 3.3V level
57	GND	Ground	Board ground
58, 59, 60, 61, 62	NC	N/A	No connect
63	GND	Ground	Board ground
64, 65, 66, 67, 68	NC	N/A	No connect
69	GND	Ground	Board ground
70, 71, 72, 73	NC	N/A	No connect
74	VDD_3V3	Power	3.3V power supply
75	GND	Ground	Board ground

4. Standalone RF Testing

The BDE-BW33-EM along with the breakout board with module, can be used standalone (without a host) to test RF capabilities, using SimpleLink Wi-Fi Toolbox.

SimpleLink Wi-Fi Toolbox is a GUI-based tool for RF evaluation and testing of BDE-BW33xx modules during development and certification. The tool enables low-level radio testing capabilities by manually setting the radio

into transmit or receive modes. Usage of the tool requires familiarity and knowledge of radio circuit theory and radio test methods.

Below section describes the hardware setup and how to use XDS110 and SimpleLink Wi-Fi Toolbox to perform conducted RF testing.

4.1. Prerequisites

- Windows 10 64-bit/Ubuntu 18 (or higher) 64-bit operation system;
- Latest Chrome web browser;
- Installation of Simplelink Wi-Fi Toolbox;
- BDE-BW33-EM + BDE-BW3301NP1-BO (can be different depending on the target module);
- BDE-XDS110 debugger for SWD communication.

The LP-XDS110 enables direct communication to the module via the SWD interface. This allows external tools, such as the Simplelink Wi-Fi Toolbox, to send commands directly to the device without the use of an embedded host.

4.2. Test Setup

Figure 14 shows the test setup for using BDE-XDS110 to perform RF conducted testing for the module.

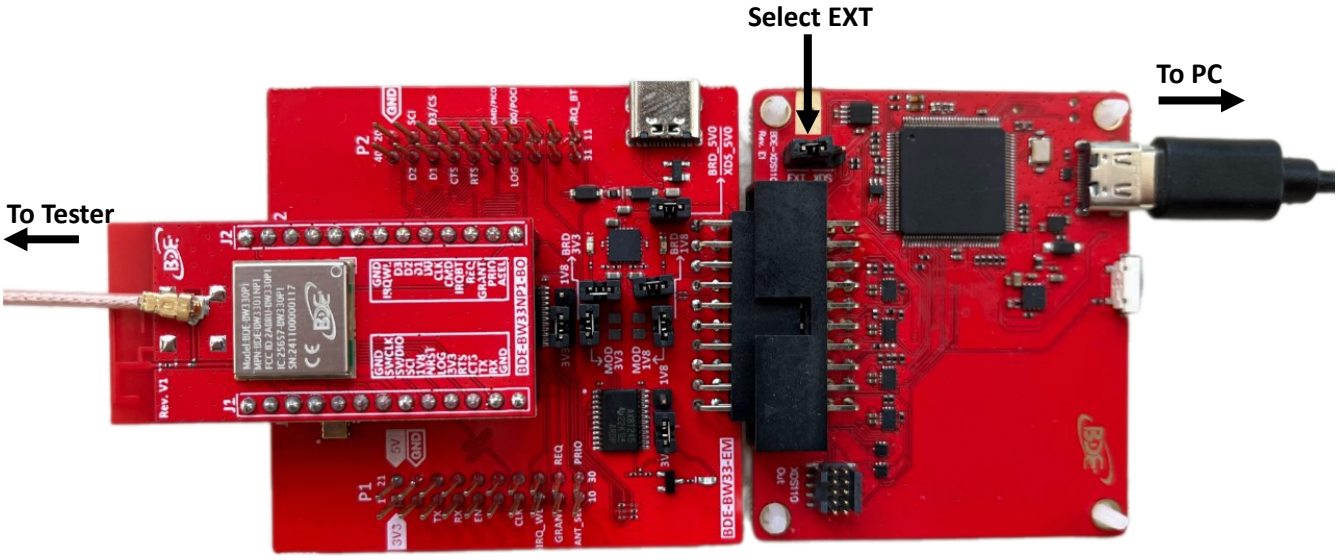
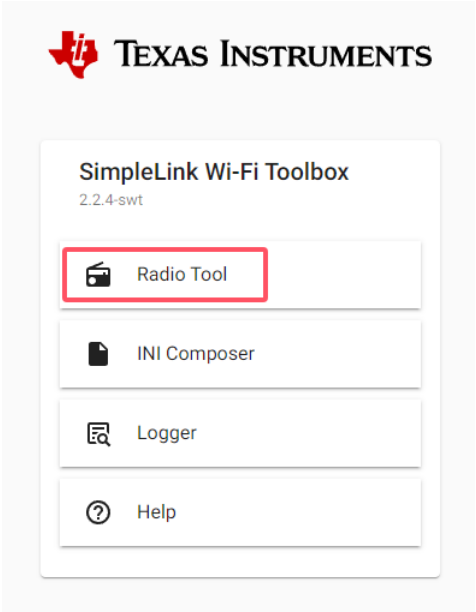


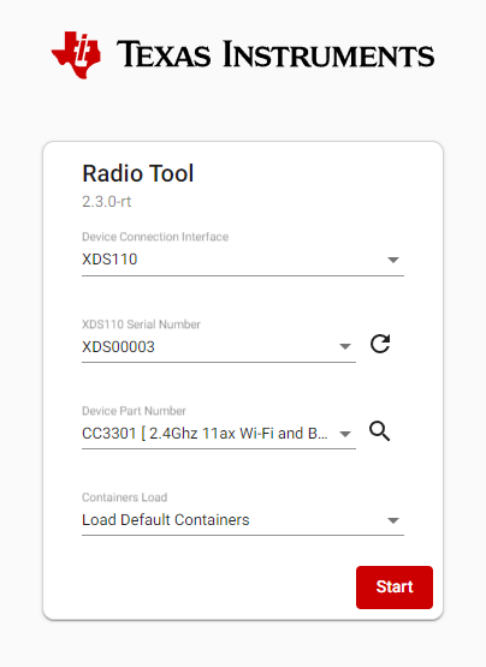
Figure 14. Standalone RF Testing Test Setup

4.3. Using Wi-Fi Toolbox

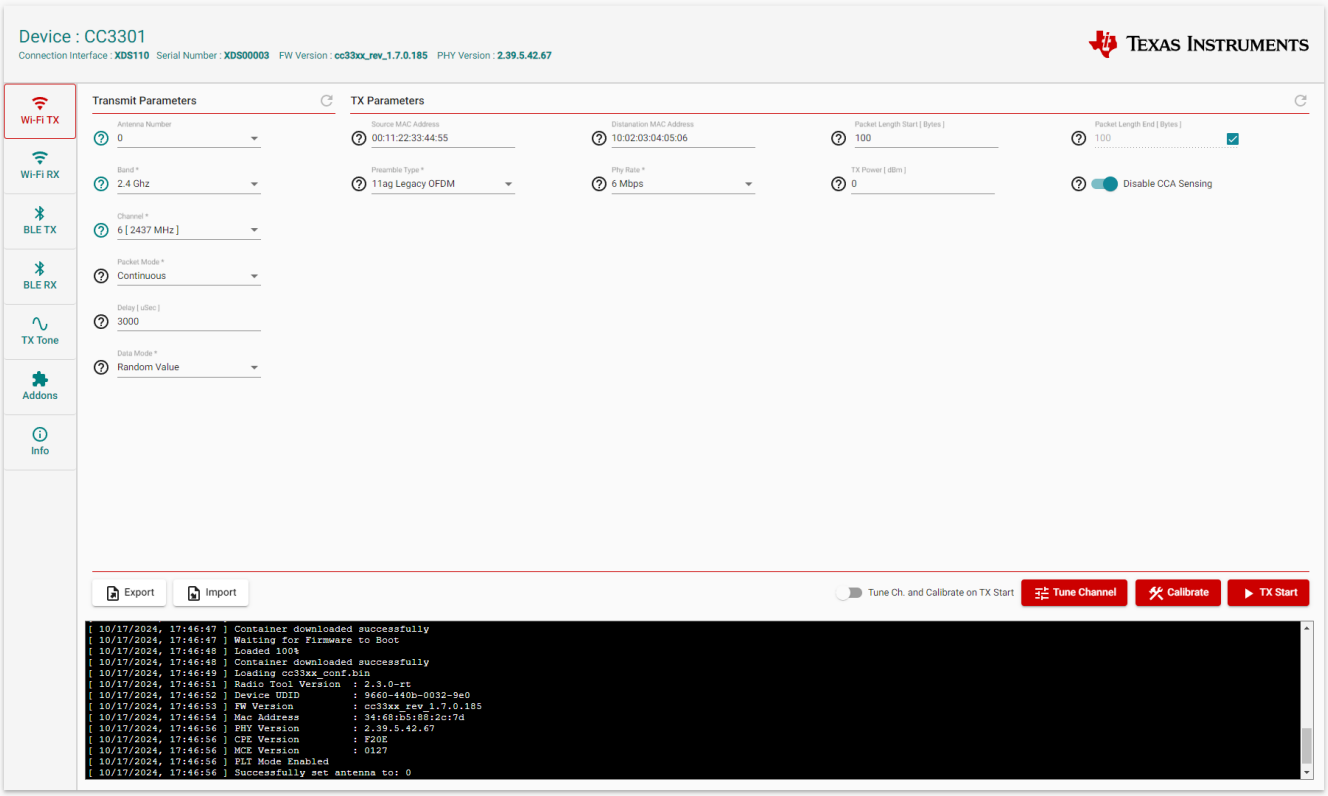
(1) Follow above setup and run Wi-Fi Toolbox, select “Radio Tool”;



(2) Select XDS110 interface and target device;



(3) Start performing the conducted RF testing.



For more information on how to use Wi-Fi Toolbox for testing CC33xx devices, you can refer to the user guide of the Toolbox: “CC33XX-Wifi-toolbox-User-Guide.html”.

This user guide is located in the installation folder of the Toolbox: “*C:\TI\simplelink_wifi_toolbox_2_2_4\docs*”, depending on where you install it.

5. Integration Options

The BDE-BW33xx module can be integrated in to different MCUs running RTOS or MPUs running Linux. BDE provides different evaluation kits for different platforms.

5.1. MCU and RTOS Evaluation: AM243 Platform

5.1.1. Description

This guide describes how to enable the BDE-BW3301NP1 module with TI’s Sitara™ high performance MCU AM243x series. Below are the steps for building and running the Wi-Fi example (named Network Terminal) from the CC33xx MCU package. The MCU package is a plugin which is based on content (such as TI drivers for peripherals, LWIP network stack, FreeRTOS OS adaptation) available in the base SDK (MCU-PLUS-AM243X-SDK).

5.1.2. Prerequisites

5.1.2.1. Hardware

- BDE-BW33-EM, evaluation plug-in module for BDE-BW33xx module series;
- BDE-BW3301NP1-BO, breakout board with BDE-BW3301NP1 module (can be different depending on the target module);
- LP-AM243, TI's Sitara™ high performance MCU AM243x Launchpad.

5.1.2.2. Software and Tools

- MCU-PLUS-AM243X-SDK (version 08.05.00.24)
- cc33xx_mcu_package (version cc33xx_mcu_package_R5)
- CCS Software
- TI_Clang 1.3.0 LTS
- Python 3.7 (see instructions in Sitara™ LP Manual)

5.1.3. MCU Package Folder Structure

- **ccs_projects/** - Projects to build required libraries
  - **ble_host_nimble**
  - **ble_interface**
  - **lwip**
  - **wifi_host_driver**
  - **wifi_interface**
  - **wifi_platform_cc33xx**
- **docs/** - User guides and API references
- **examples/** - Example applications
  - **CC3xx_ble_wifi_provisioning**
  - **CC3xx_thick_mac_network_terminal**
- **source/** - Drivers and libraries source code
  - **Wi-Fi Stack API** is under source/ti/net/wifi_interface
  - **cc33xx Driver** is under source/ti/drivers/net/wifi
  - **NimBLE Stack** is under source/third_party/nimble
- **tools/** -
  - **programing** - contains flash programming (Python) scripts
  - **wifi_fw** - contains CC33xx .bin files

Note: In the following sections, <CC33xx_MCU_package> represents the SDK root folder, such as C:/ti/cc33xx_mcu_package_R5.

5.1.4. Hardware Setup

For more information on the Sitara™ LaunchPad please refer to LP-AM243_EVM-SETUP.

Plug the BDE-BW33-EM+BDE-BW3301NP1-BO to the Sitara™ LaunchPad LP-AM243. When connecting the boards please remember to check the alignment of the 5V and GND pins of both boards to ensure right connection.

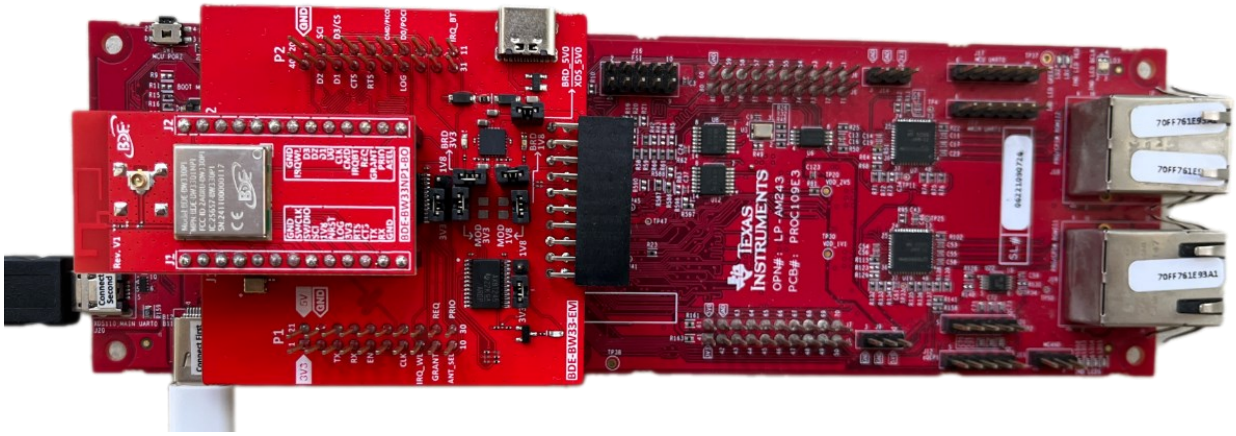


Figure 15. AM234x Hardware Setup

Table 8 shows the connections being made between the AM243x Launchpad and the BDE-BW33-EM.

Table 8. Pin Connection Between AM234x and BDE-BW33-EM

LP-AM243 Pin Number	BDE-BW33-EM Pin Number	Pin Descriptions
J3.21	P1.21	5V
J3.22	P1.22	GND
J1.3	P1.3	LP-AM243: RX to BDE-BW33-EM: TX
J1.4	P1.4	LP-AM243: TX to BDE-BW33-EM: RX
J1.5	P1.5	nRESET
J1.7	P1.7	SPI CLK
J1.8	P1.8	WLAN IRQ
J2.14	P2.14	SPI MISO
J2.15	P2.15	SPI MOSI
J2.18	P2.13	SPI CS
J4.36	P2.36	BDE-BW33-EM: CTS
J4.37	P2.37	BDE-BW33-EM: RTS

Power Connections:

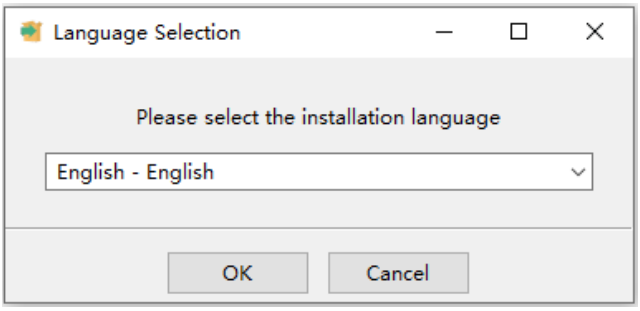
- Connect the LP-AM243 to its power supply and to the PC;
- When Disconnecting/Connecting the LP-AM243, always make sure you follow sequence:
  (1) Plug In: Connect power cable (USB-C), then Connect data cable (Micro USB);
  (2) Plug Out: Disconnect data cable (Micro USB), then Disconnect power cable (USB-C).

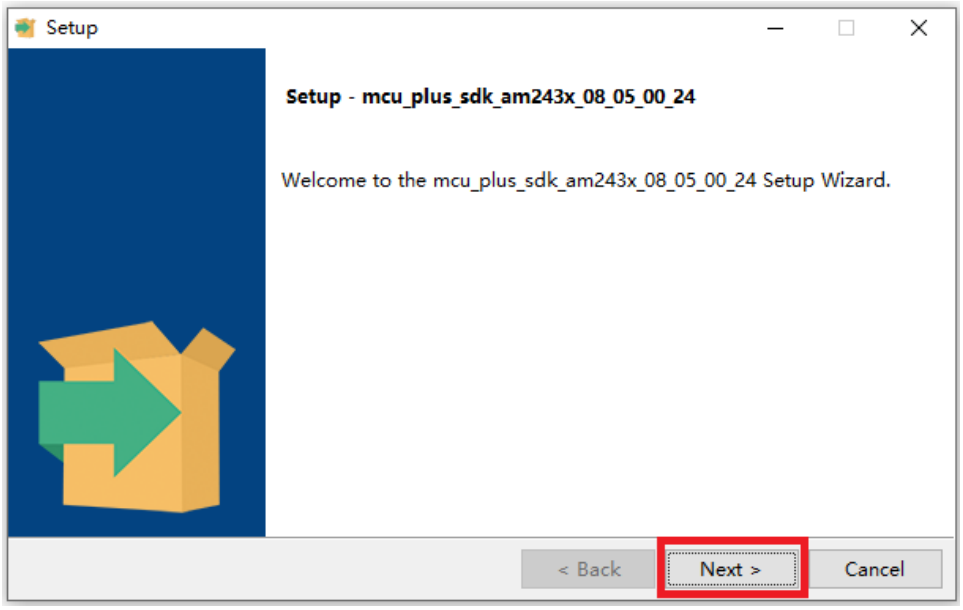
5.1.5. Install AM243X SDK & CCS

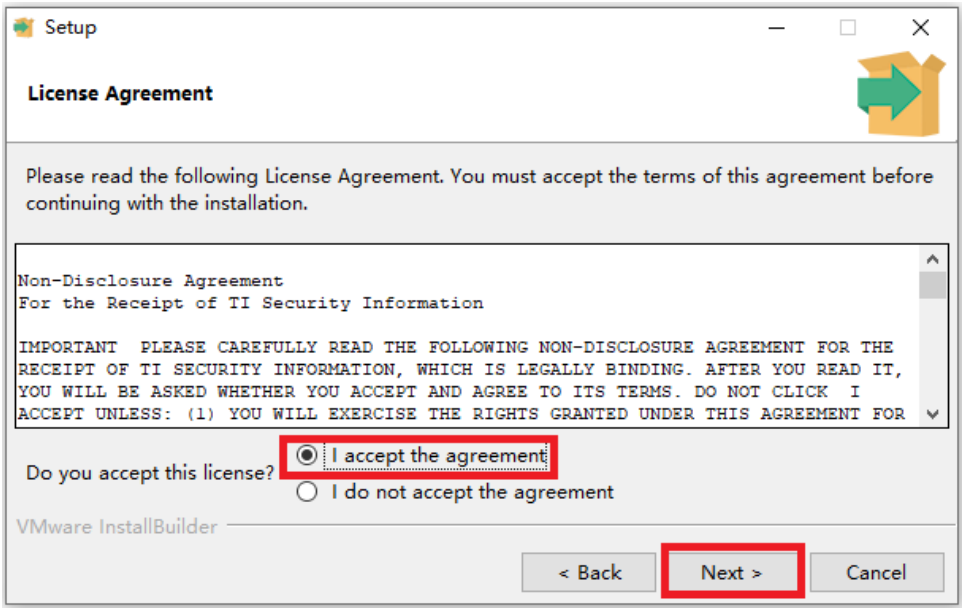
5.1.5.1. Download and Install AM243x SDK

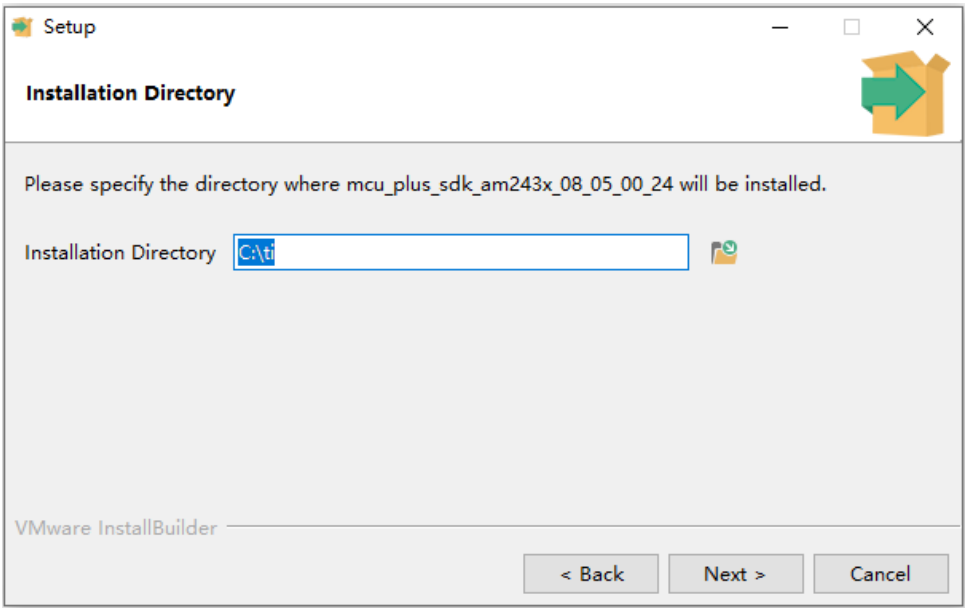
(1) Link to MCU-PLUS-AM243X-SDK resources, select mcu_plus_xxx-installer.exe in version 08.05.00.24 and download it;

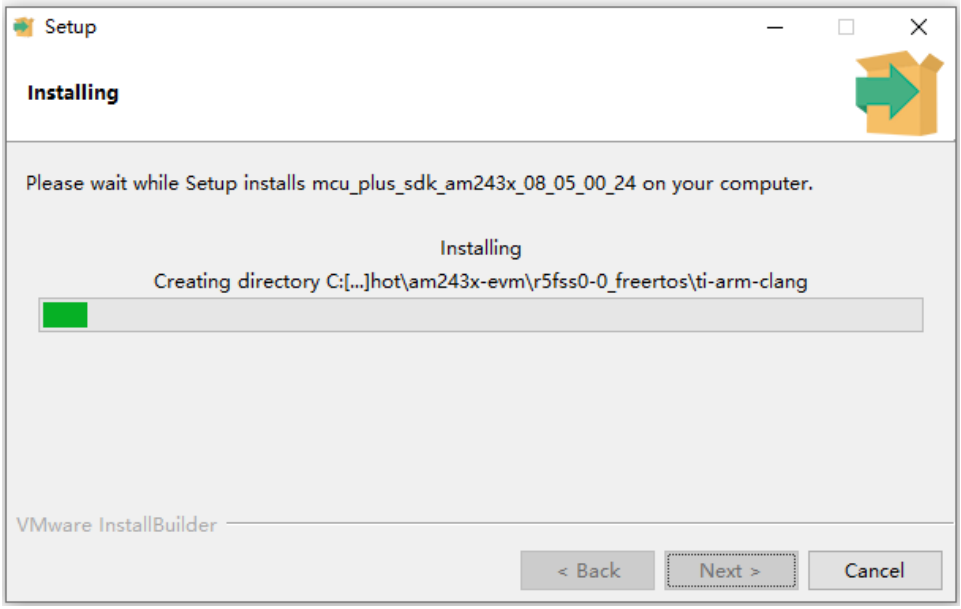
(2) Install the SDK.

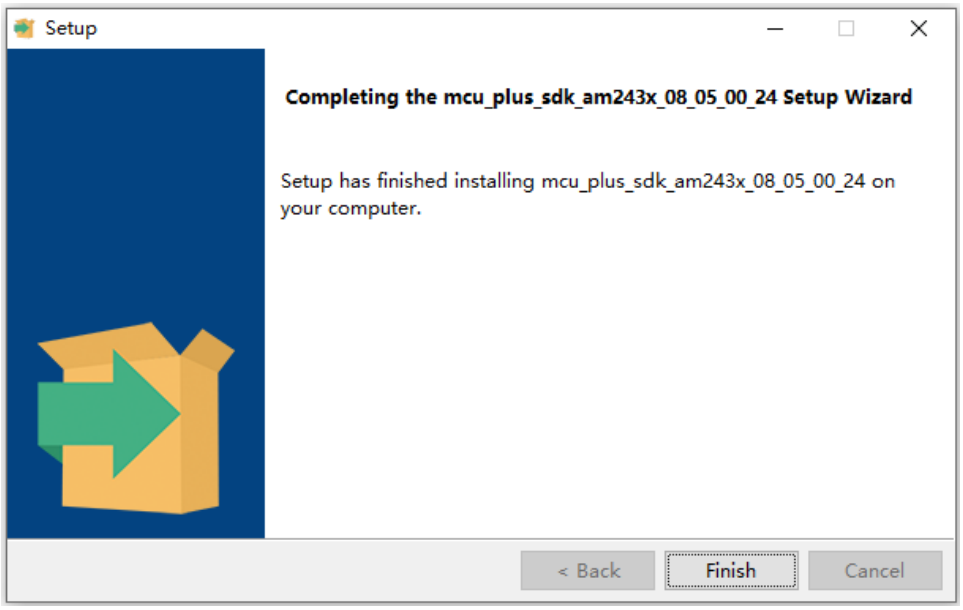



Setup
Setup - mcu_plus_sdk_am243x_08_05_00_24
Welcome to the mcu_plus_sdk_am243x_08_05_00_24 Setup Wizard.
< Back
Next >
Cancel


Setup
License Agreement
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.
Non-Disclosure Agreement
For the Receipt of TI Security Information
IMPORTANT PLEASE CAREFULLY READ THE FOLLOWING NON-DISCLOSURE AGREEMENT FOR THE RECEIPT OF TI SECURITY INFORMATION, WHICH IS LEGALLY BINDING. AFTER YOU READ IT, YOU WILL BE ASKED WHETHER YOU ACCEPT AND AGREE TO ITS TERMS. DO NOT CLICK I ACCEPT UNLESS: (1) YOU WILL EXERCISE THE RIGHTS GRANTED UNDER THIS AGREEMENT FOR
Do you accept this license?
I accept the agreement
I do not accept the agreement
VMware InstallBuilder
< Back
Next >
Cancel

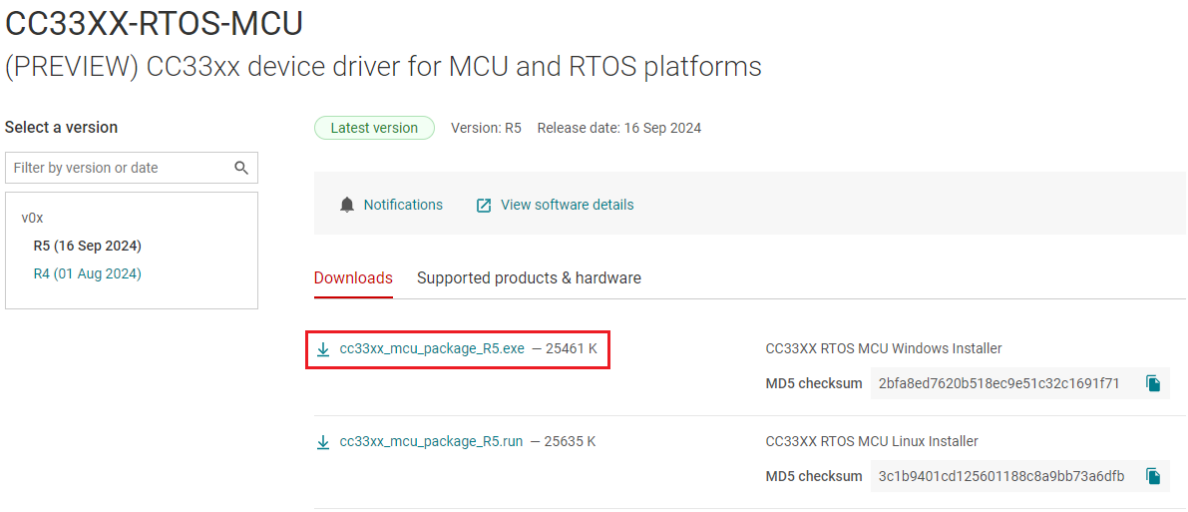

Setup
Installation Directory
Please specify the directory where mcu_plus_sdk_am243x_08_05_00_24 will be installed.
Installation Directory
C:\ti
VMware InstallBuilder
< Back
Next >
Cancel


Setup
Installing
Please wait while Setup installs mcu_plus_sdk_am243x_08_05_00_24 on your computer.
Installing
Creating directory C:[...]hot\am243x-evm\r5fss0-0_freertos\ti-arm-clang
VMware InstallBuilder
< Back
Next >
Cancel

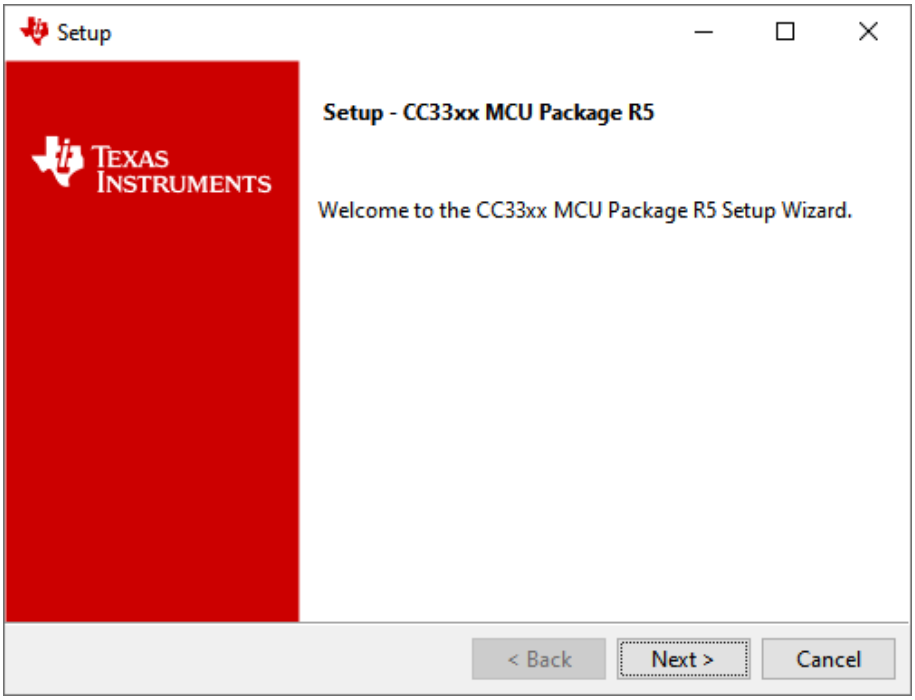


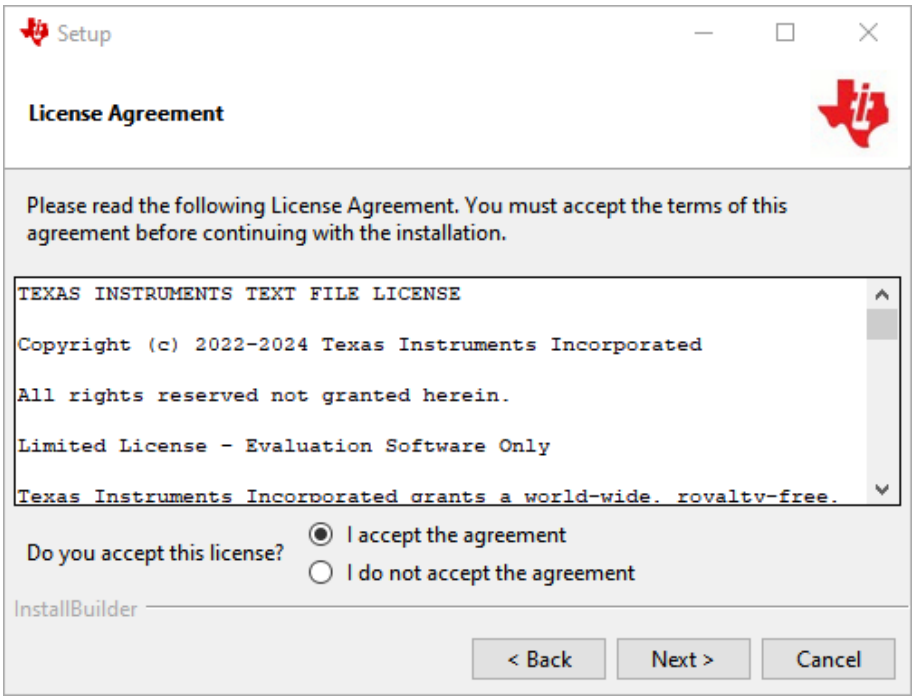
5.1.5.2. Download and Install CC33xx MCU Package

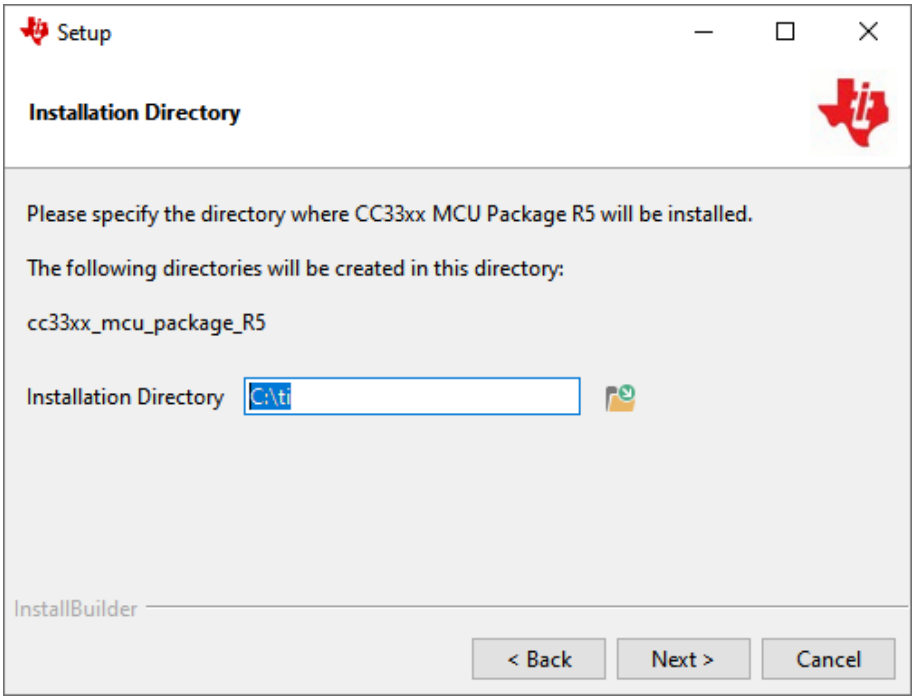
(1) Link to cc33xx_mcu_package resources, select cc33_mcu_package_R5.exe and download it;

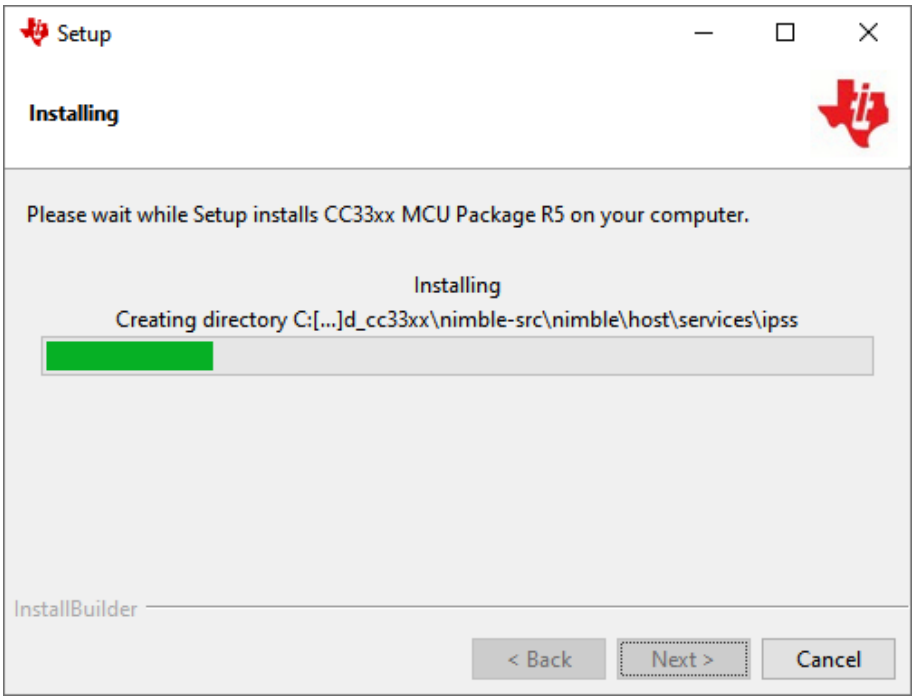


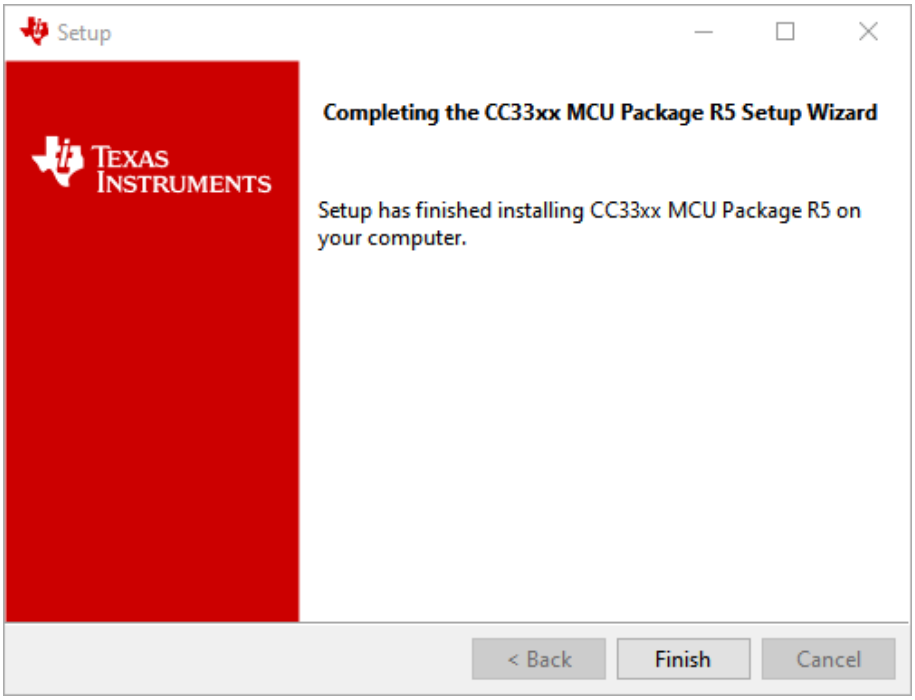
(2) Install the CC33xx MCU Package.


Setup
Setup - CC33xx MCU Package R5
Welcome to the CC33xx MCU Package R5 Setup Wizard.
< Back
Next >
Cancel


Setup
License Agreement
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.
TEXAS INSTRUMENTS TEXT FILE LICENSE
Copyright (c) 2022-2024 Texas Instruments Incorporated
All rights reserved not granted herein.
Limited License - Evaluation Software Only
Texas Instruments Incorporated grants a world-wide, royalty-free,
Do you accept this license?
I accept the agreement
I do not accept the agreement
InstallBuilder
< Back
Next >
Cancel


Setup
Installation Directory
Please specify the directory where CC33xx MCU Package R5 will be installed.
The following directories will be created in this directory:
cc33xx_mcu_package_R5
Installation Directory
C:\ti
InstallBuilder
< Back
Next >
Cancel

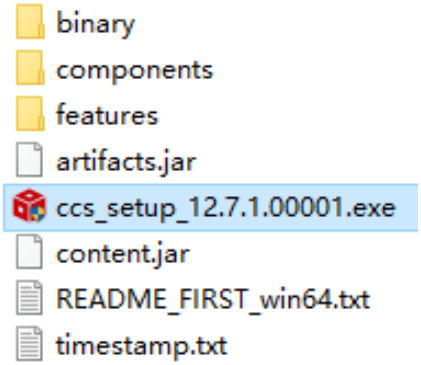

Setup
Installing
Please wait while Setup installs CC33xx MCU Package R5 on your computer.
Installing
Creating directory C:[...]d_cc33xx\nimble-src\nimble\host\services\ipss
InstallBuilder
< Back
Next >
Cancel

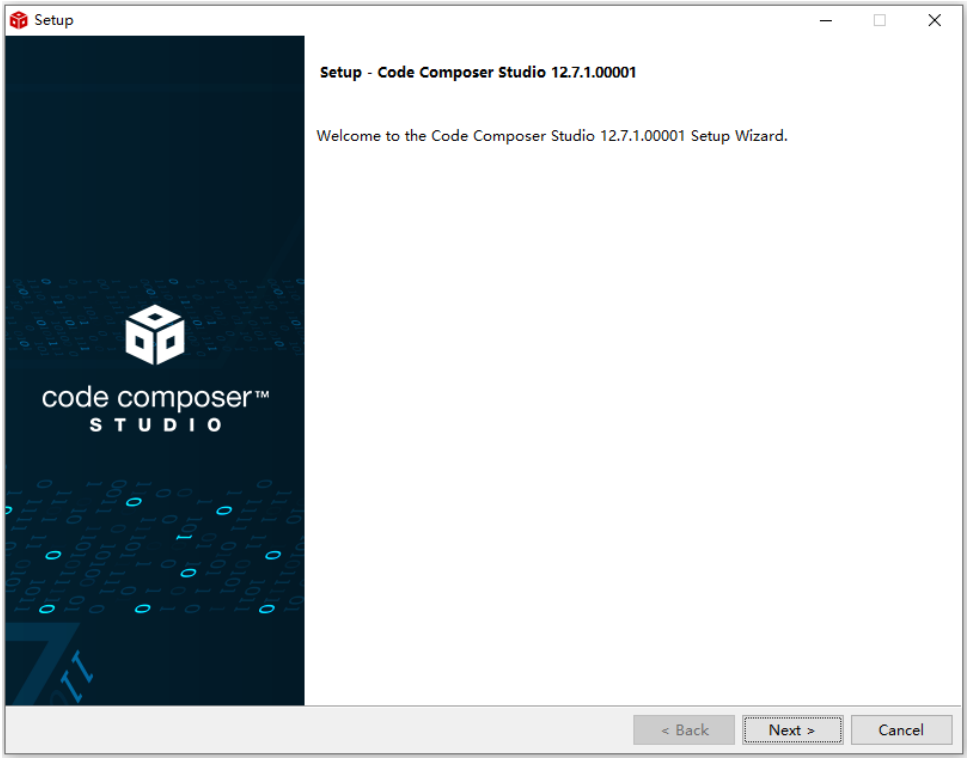


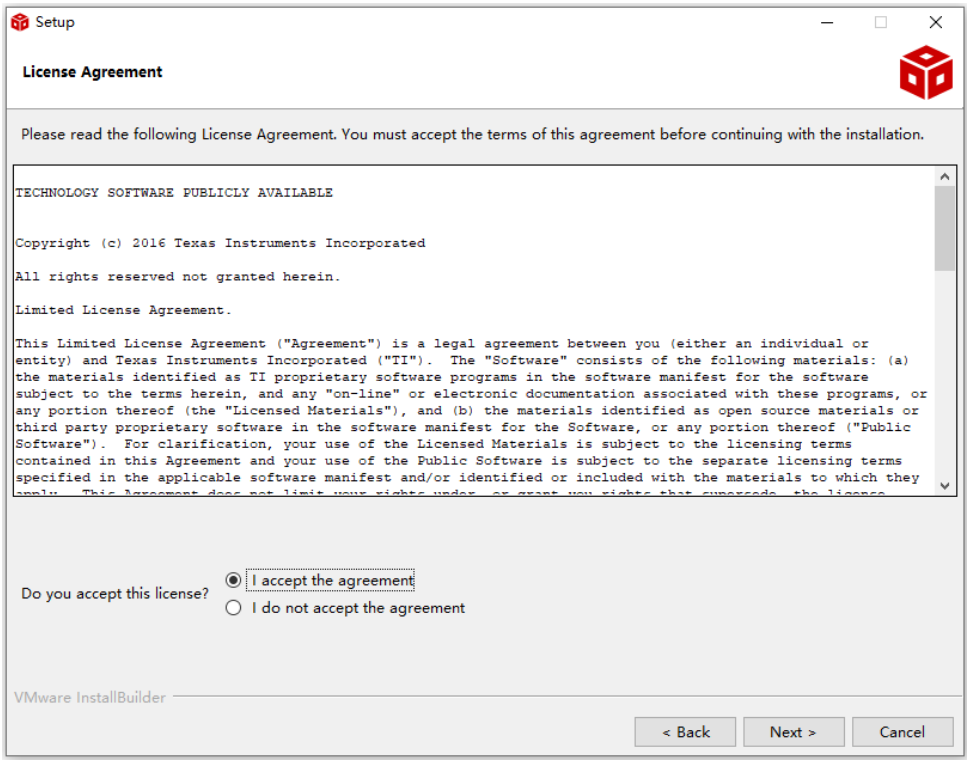
5.1.5.3. Download and Install CCS

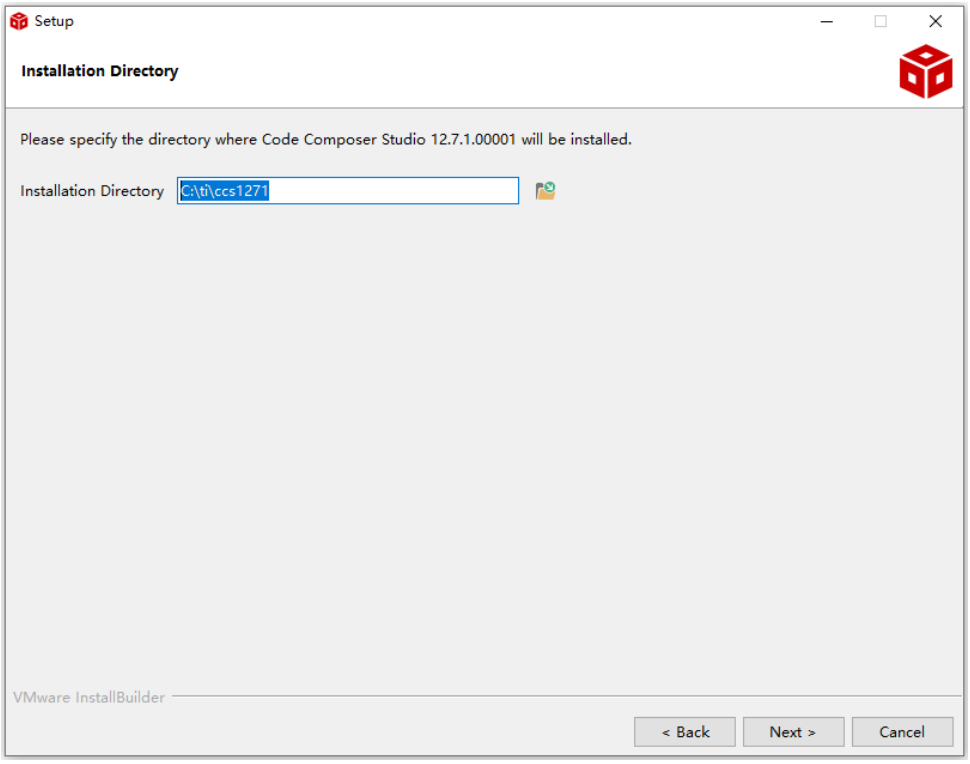
(1) Link to CCS Download resources and select the newest version;

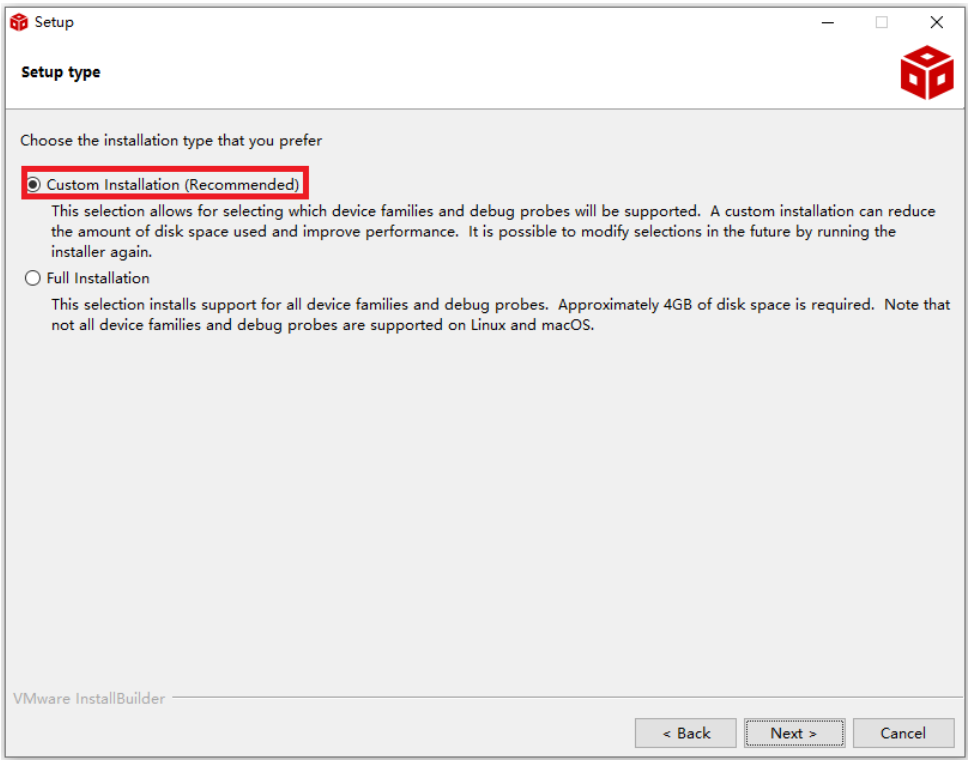
(2) Unzip and install the CCS;

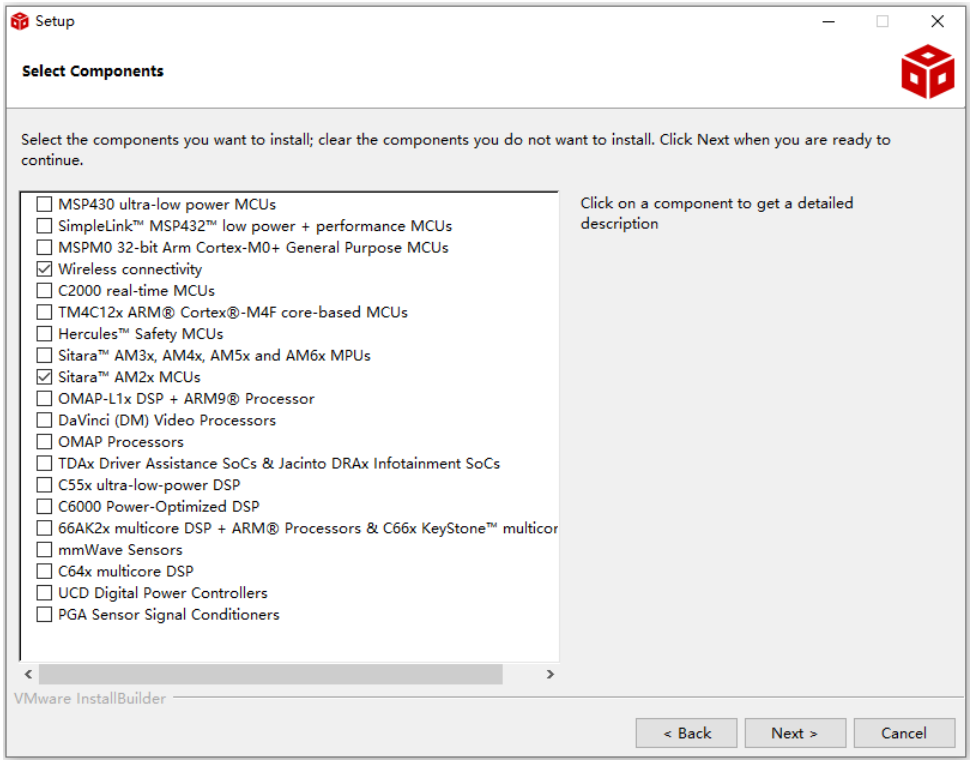

binary
components
features
artifacts.jar
ccs_setup_12.7.1.00001.exe
content.jar
README_FIRST_win64.txt
timestamp.txt

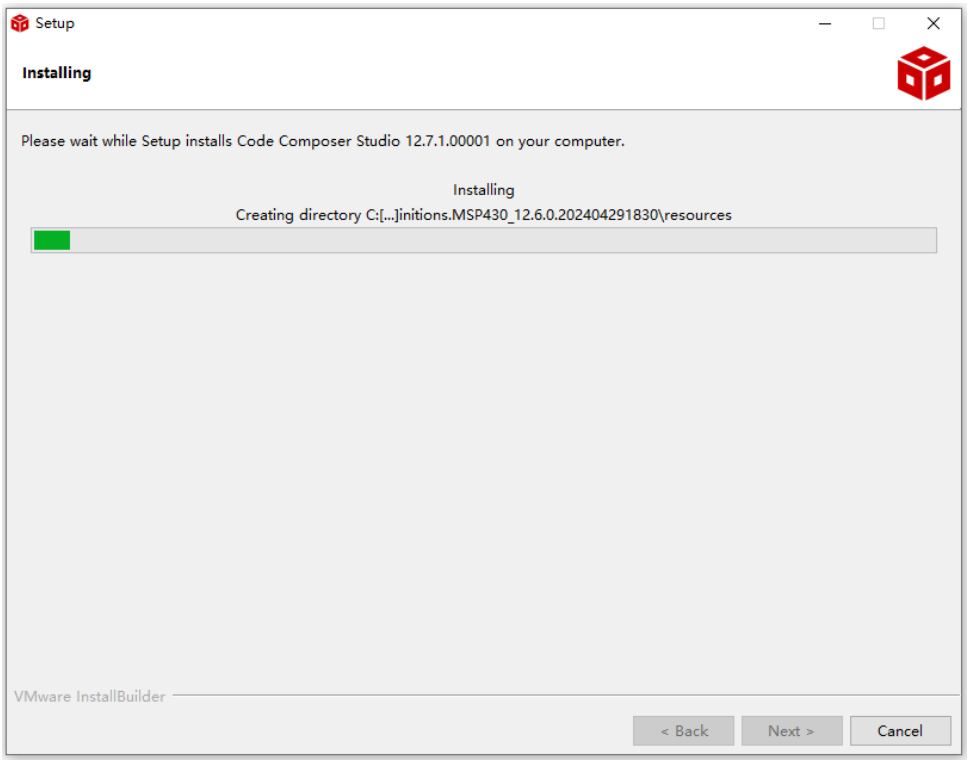

Setup
Setup - Code Composer Studio 12.7.1.00001
Welcome to the Code Composer Studio 12.7.1.00001 Setup Wizard.
code composer™
STUDIO
< Back
Next >
Cancel

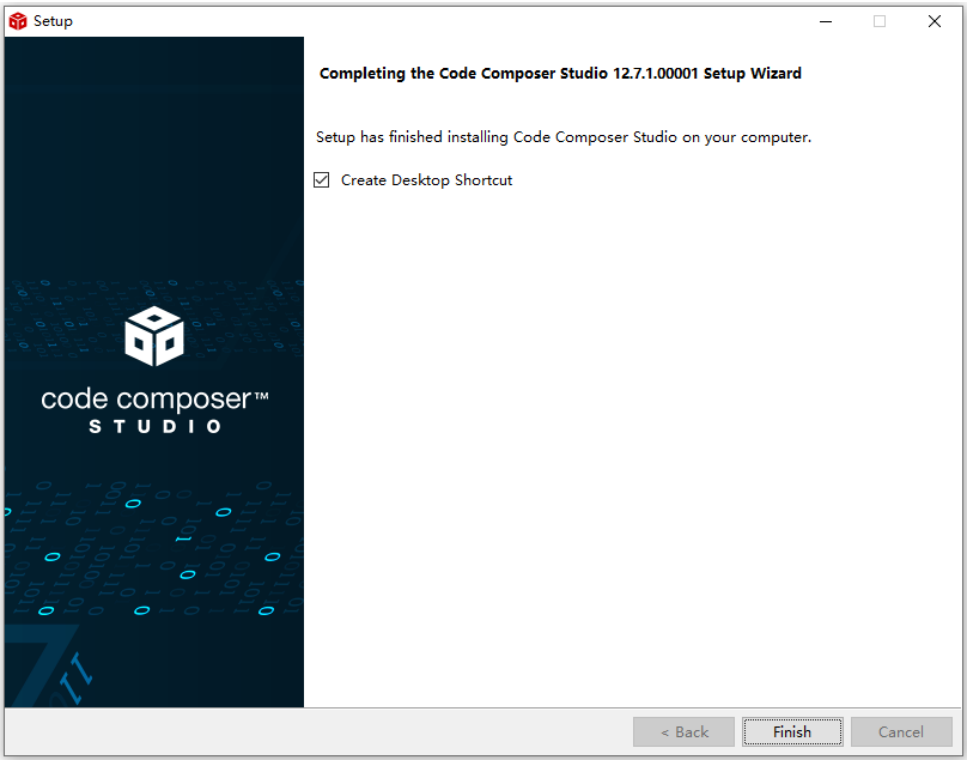

Setup
License Agreement
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.
TECHNOLOGY SOFTWARE PUBLICLY AVAILABLE
Copyright (c) 2016 Texas Instruments Incorporated
All rights reserved not granted herein.
Limited License Agreement.
This Limited License Agreement ("Agreement") is a legal agreement between you (either an individual or entity) and Texas Instruments Incorporated ("TI"). The "Software" consists of the following materials: (a) the materials identified as TI proprietary software programs in the software manifest for the software subject to the terms herein, and any "on-line" or electronic documentation associated with these programs, or any portion thereof (the "Licensed Materials"), and (b) the materials identified as open source materials or third party proprietary software in the software manifest for the Software, or any portion thereof ("Public Software"). For clarification, your use of the Licensed Materials is subject to the licensing terms contained in this Agreement and your use of the Public Software is subject to the separate licensing terms specified in the applicable software manifest and/or identified or included with the materials to which they
Do you accept this license?
I accept the agreement
I do not accept the agreement
VMware InstallBuilder
< Back
Next >
Cancel


Setup
Installation Directory
Please specify the directory where Code Composer Studio 12.7.1.00001 will be installed.
Installation Directory
C:\ti\ccs1271
VMware InstallBuilder
< Back
Next >
Cancel


Setup
Setup type
Choose the installation type that you prefer
Custom Installation (Recommended)
This selection allows for selecting which device families and debug probes will be supported. A custom installation can reduce the amount of disk space used and improve performance. It is possible to modify selections in the future by running the installer again.
Full Installation
This selection installs support for all device families and debug probes. Approximately 4GB of disk space is required. Note that not all device families and debug probes are supported on Linux and macOS.
VMware InstallBuilder
< Back
Next >
Cancel


Setup
Select Components
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
MSP430 ultra-low power MCUs
SimpleLink™ MSP432™ low power + performance MCUs
MSPM0 32-bit Arm Cortex-M0+ General Purpose MCUs
Wireless connectivity
C2000 real-time MCUs
TM4C12x ARM® Cortex®-M4F core-based MCUs
Hercules™ Safety MCUs
Sitara™ AM3x, AM4x, AM5x and AM6x MPUs
Sitara™ AM2x MCUs
OMAP-L1x DSP + ARM9® Processor
DaVinci (DM) Video Processors
OMAP Processors
TDAx Driver Assistance SoCs & Jacinto DRAx Infotainment SoCs
C55x ultra-low-power DSP
C6000 Power-Optimized DSP
66AK2x multicore DSP + ARM® Processors & C66x KeyStone™ multicor
mmWave Sensors
C64x multicore DSP
UCD Digital Power Controllers
PGA Sensor Signal Conditioners
Click on a component to get a detailed description
VMware InstallBuilder
< Back
Next >
Cancel





5.1.6. Import and Build the NetworkTerminal Demo

(1) Go to **File->Import**

(2) Choose **C/C++ -> CCS Projects**

(3) Browse… -> **<CC33xx_MCU_package>/**

(4) Choose all - **wifi_host_driver**, **wifi_platform_cc33xx**, **wifi_stack_interface**, **lwip**, **ble_host_nimble**, **ble_interface and CC3xxx_network_terminal**

(5) First compile the Wi-Fi and BLE projects, then compile the network_terminal application

---

Note: If an error like "subdir_rules.mk:xx: recipe for target 'build-xxx' failed" occurs during the compilation process, please replace it with a higher version of sysConfig.

5.1.7. Program the Flash

Before running the example, the RAM bootloader and CC33xx Firmware need to be programmed to the flash on the AM243. The content of the 3 files can be found under **<CC33xx_MCU_package>/tools/wifi_fw/**

(1) The image is loaded and executed from the AM243x MSRAM. To save RAM space the binaries are stored in the flash and read during the FW init;

(2) The binaries should be programmed to a pre-defined offsets in the flash, as shown below table;

Table 9. Binary and Flash Offset

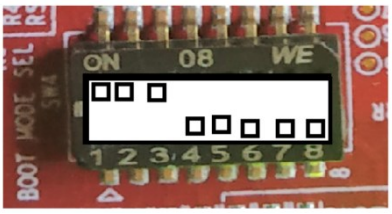
Binary Name	Description	Location	Offset
cc3xxx_network_terminal.appimage	Application Image	under app project output folder	0x00080000
cc33xx_fw.bin	CC33XX Firmware	under tools/cc33xx_firmware/	0x00800000
cc33xx_2nd_loader.bin	RAM Bootloader	under tools/cc33xx_firmware/	0x00900000

(3) After the successful compilation of the application, the binary image (.appimage or .appimage.hs_fs (for secure devices) file, e.g. cc3xxx_network_terminal.appimage) can be found in the Release/ folder inside the project.

5.1.8. Programming Instructions

Follow below steps for programming the flash of the evaluation board.

(1) Configure LP-AM243 to UART boot mode according to AM243x UART Boot Mode;



(2) If you are using a secure MCU (LP-AM243 PROC1093B or newer), open the <cc33xx_mcu_package>/tools/secured_programming folder (in a command line window). If you are using an older LP version - move to the <cc33xx_mcu_package>/tools/programming/ folder;

(3) Set up the programmer's configuration file;

In the **tools/programing** folder, there is a pre-configured file:
program_application_and_wifi_fw.cfg - contains both the CC3301firmware binaries and the application image. The configuration file will just work when the programming script is executed from the **tools/programing** folder.

All the paths are based on default locations of the files inside the SDK folder structure. When changing any of the default configuration (e.g. using an application other than the reference one or trying to use a special firmware binary that was not part of the SDK), you may need to manually edit the **program_application_and_wifi_fw.cfg** file and update the file path. Below is an example for the cfg file:

```
# --flash-writer=./sbl_uart_uniflash.release.tiimage
# First point to sbl_uart_uniflash binary, which function's as a server to flash one or more
files
--flash-writer=./sbl_uart_uniflash.release.tiimage

# Now send one or more files to flash or flashverify as needed. The order of sending files
does not matter

# Program the OSPI PHY tuning attack vector
--operation=flash-phy-tuning-data

# --file=./sbl_ospi.release.tiimage --operation=flash --flash-offset=0x0
# When sending bootloader make sure to flash at offset 0x0. ROM expects bootloader at offset
0x0
--file=./sbl_ospi.release.tiimage --operation=flash --flash-offset=0x0

#Network Terminal appimage file (cc3xxx_network_terminal.appimage)
--
file=C:/ti/cc33xx_mcu_package_R5/examples/CC3xx_thick_mac_network_terminal/build_cc3xx_rel
ease/cc3xxx_network_terminal.appimage --operation=flash --flash-offset=0x80000

#Wifi FW container (cc33xx_fw.bin)
--file=../wifi_fw/cc33xx_fw.bin --operation=flash --flash-offset=0x800000

#Wifi RAM Bootloafer container (cc33xx_2nd_loader.bin)
--file=../wifi_fw/cc33xx_2nd_loader.bin --operation=flash --flash-offset=0x900000

#cc33xx Conf configurations file (cc33xx-conf.bin)
--file=../wifi_fw/cc33xx-conf.bin --operation=flash --flash-offset=0xa00000
```

(4) Open CMD window, go to **<CC33xx_MCU_package>/tools/programing** folder;

(5) Run the uart_uniflash.py python script.

a) In Windows:

```
python uart_uniflash.py -p COM[XDS uart number] --
cfg=program_application_and_wifi_fw.cfg
```

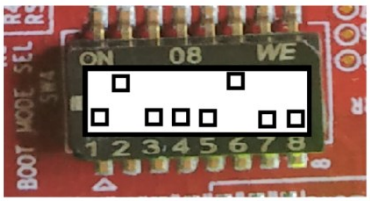
b) In Linux:

```
python3 uart_uniflash.py -p /dev/ttyACM[XDS uart number] --
cfg=program_application_and_wifi_fw.cfg
```

```
C:\ti\cc33xx_mcu_package_R5\tools\programing>python uart_uniflash.py -p COM18 --cfg=program_application_and_wifi_fw.cfg

Parsing config file ...
Parsing config file ... ERROR. 1 error(s).

[ERROR] Parsing error found on line 29 of program_application_and_wifi_fw.cfg
[ERROR] File not found !!!


C:\ti\cc33xx_mcu_package_R5\tools\programing>python uart_uniflash.py -p COM18 --cfg=program_application_and_wifi_fw.cfg

Parsing config file ...
Parsing config file ... SUCCESS. Found 7 command(s) !!!

Executing command 1 of 7 ...
Found flash writer ... sending ./sbl_uart_uniflash.release.tiimage
Sent flashwriter ./sbl_uart_uniflash.release.tiimage of size 294775 bytes in 28.23s.

Executing command 2 of 7 ...
Command arguments : --operation=flash-phy-tuning-data
Sent flash phy tuning data in 1.6s.
[STATUS] SUCCESS !!!

Executing command 3 of 7 ...
Command arguments : --file=./sbl_ospi.release.tiimage --operation=flash --flash-offset=0x0
Sent ./sbl_ospi.release.tiimage of size 299668 bytes in 29.92s.
[STATUS] SUCCESS !!!

Executing command 4 of 7 ...
Command arguments : --file=C:/ti/cc33xx_mcu_package_R5/examples/CC3xx_thick_mac_network_terminal/build_cc3xx_release/cc3xxx_network_terminal.appimage --operation=flash --flash-offset=0x80000
Sent C:/ti/cc33xx_mcu_package_R5/examples/CC3xx_thick_mac_network_terminal/build_cc3xx_release/cc3xxx_network_terminal.appimage of size 464708 bytes in 44.75s.
[STATUS] SUCCESS !!!

Executing command 5 of 7 ...
Command arguments : --file=../wifi_fw/cc33xx_fw.bin --operation=flash --flash-offset=0x800000
Sent ../wifi_fw/cc33xx_fw.bin of size 521716 bytes in 49.97s.
[STATUS] SUCCESS !!!

Executing command 6 of 7 ...
Command arguments : --file=../wifi_fw/cc33xx_2nd_loader.bin --operation=flash --flash-offset=0x900000
Sent ../wifi_fw/cc33xx_2nd_loader.bin of size 70212 bytes in 9.37s.
[STATUS] SUCCESS !!!

Executing command 7 of 7 ...
Command arguments : --file=../wifi_fw/cc33xx-conf.bin --operation=flash --flash-offset=0xa00000
Sent ../wifi_fw/cc33xx-conf.bin of size 1129 bytes in 3.17s.
[STATUS] SUCCESS !!!

All commands from config file are executed !!!
```

5.1.9. Run the Example from Flash

When loading the code from flash, configure the LP for AM243x QSPI Boot Mode.



(1) Open a serial terminal;

(2) Press the reset button on the LP-AM243 and trigger the application. The following menu should appear on the terminal;

```
DMSC Firmware Version 8.5.3--v08.05.03 (Chill Capybar
DMSC Firmware revision 0x8
DMSC ABI revision 3.1

[BOOTLOADER_PROFILE] Boot Media       : NOR SPI FLASH
[BOOTLOADER_PROFILE] Boot Media Clock : 100.000 MHz
[BOOTLOADER_PROFILE] Boot Image Size  : 453 KB
[BOOTLOADER_PROFILE] Cores present    :
r5f0-0
[BOOTLOADER PROFILE] SYSFW init                        :      10850us
[BOOTLOADER PROFILE] System_init                       :    5367700us
[BOOTLOADER PROFILE] Drivers_open                      :        268us
[BOOTLOADER PROFILE] Board_driversOpen                 :     140494us
[BOOTLOADER PROFILE] Sciclient Get Version             :      10027us
[BOOTLOADER PROFILE] CPU load                          :       6193us
[BOOTLOADER PROFILE] SBL Total Time Taken              :     166828us

Image loading done, switching to application ...
********************************************************************
******************   Osprey  Network Terminal   ********************
******************   Version 1.1.0.4            ********************
********************************************************************
================================================================================
Available commands:

help                clear               wlan_ap_role_up     wlan_ap_role_down
wlan_sta_role_up    wlan_sta_role_down  wlan_connect        wlan_disconnect
wlan_scan           wlan_get_mac        wlan_set_mac        wlan_get_ps
wlan_set_ps         wlan_set_pm         wlan_set_LSI        wlan_start
wlan_stop           ble_adv_cfg         ble_adv_enable      ble_scan_cfg
ble_scan_enable     ble_connect         ble_disconnect      ble_peers
ble_start           ble_stop            ble_test_mode       calibrator
send                recv                socket_show         kill
================================================================================

user:
```

(6) Type **help** to show this help menu;
(7) Type the name of a command to show the required parameters;
(8) The role (AP or station) should be started before any other command, e.g. with **wlan_sta_role_up** or **wlan_ap_role_up** (only one role can be up at a single time);
(9) Starting an access-point will enable mobile devices to connect to the module;
(10) Starting a station role will enable commands such as **scan** and **wlan_connect**;
(11) Once connected to an access point, you can use **send** and **receive** to send or receive data.

By now, you should have successfully brought up the module on AM243x.

5.2. MPU and Linux Evaluation: AM335 Platform

5.2.1. Description

This section describes the step-by-step instruction on bringing up the module on TI's microprocessor AM335x.

5.2.2. Prerequisites

5.2.2.1. Hardware

- BDE-BW33-EM+BDE-BW3301NP1-BO (can be different depending on the target module);
- BDE-BW33-BBB, adaptor board for BDE-BW33-EM to BeagleBone Black;
- BEAGL-BONE-BLACK, BeagleBone® Black Board based on AM335;
- Micro-SD card (16GB or larger);
- 5V 3A power supply for BeagleBone® Black;
- FTDI or other USB to Serial Converter for BeagleBone® Black console logs.

5.2.2.2. Software and Tools

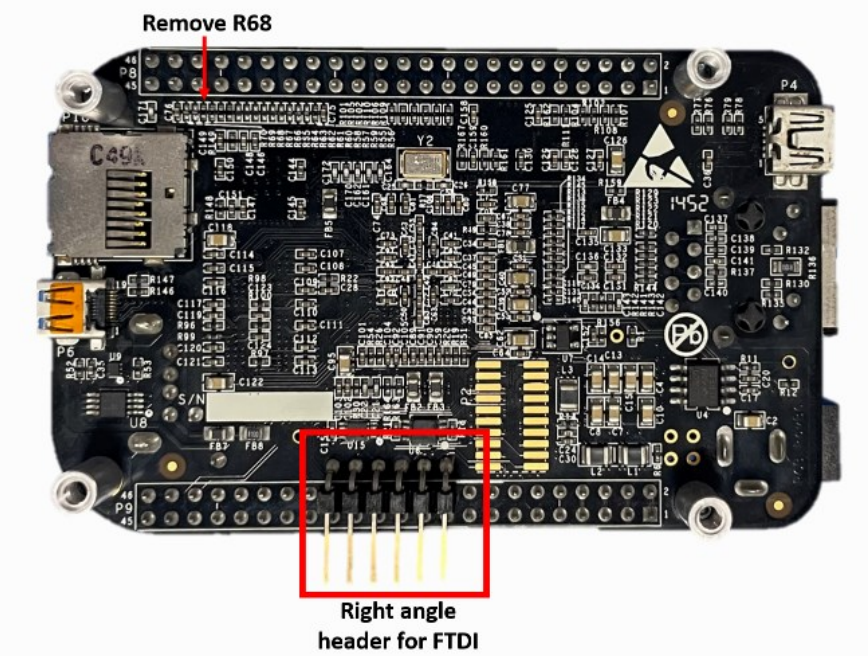
- Ubuntu 22.04;
- Processor SDK Linux for AM335x (version 09.01.00.001);
- SD card image (09.02.01.10);
- balenaEtcher;
- CC33xx device driver source (version 1.0.0.7).

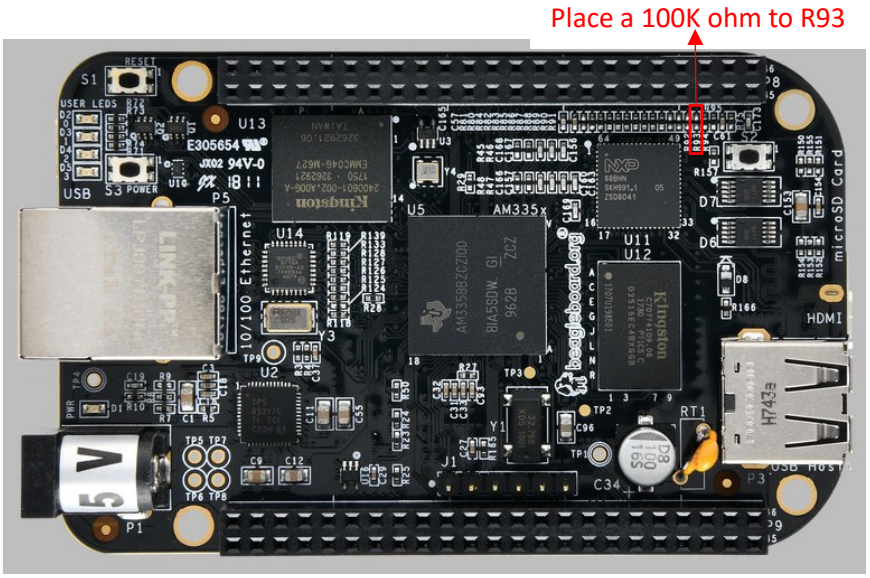
5.2.2.3. Hardware Rework

Rework is required to the BeagleBone® Black board to work with the module.

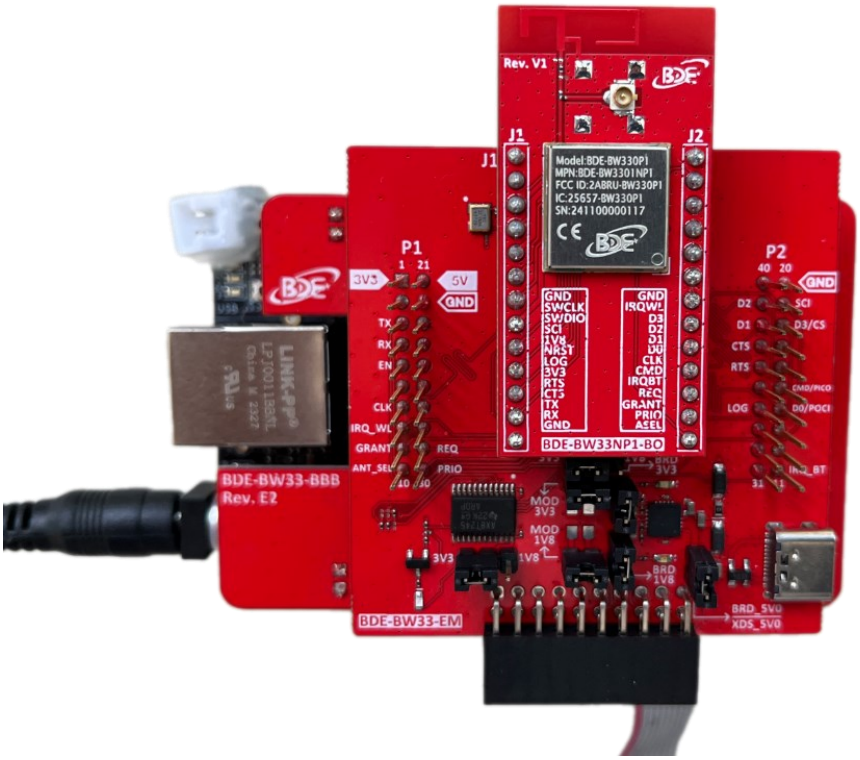
(1) Remove R68 and place it to R93 position;

(2) Add a header for UART communication with FTDI or other USB to UART bridge.





5.2.3. Hardware Setup



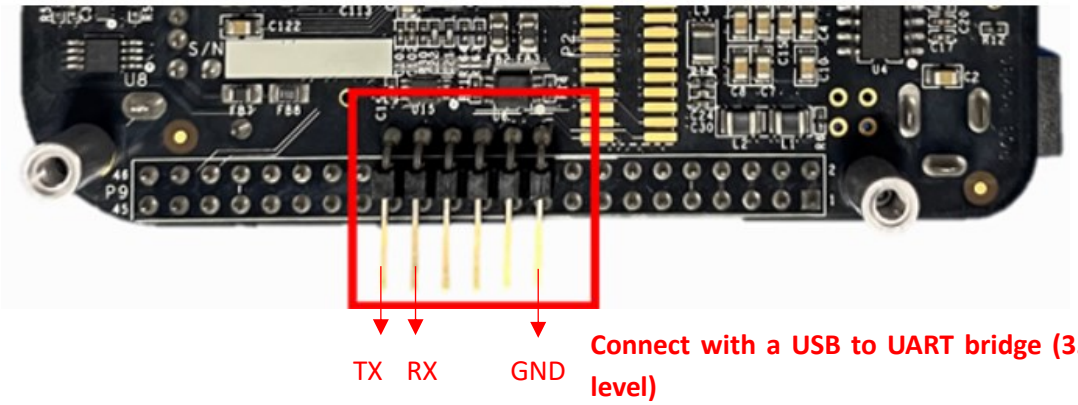


Figure 16. AM335x Hardware Setup

5.2.4. Steps

TI provides a Linux Installer allowing a user to re-build and install the CC33xx Linux drivers, firmware and related binaries. This installer is used in conjunction with TI AM335 Processor SDK.

Follow below steps to bring up the hardware.

5.2.4.1. Download SD Card Image

Download the SD card image with below link.

https://www.ti.com/tool/download/PROCESSOR-SDK-LINUX-AM335X/09.01.00.001

PROCESSOR-SDK-LINUX-AM335X

Linux Processor SDK for AM335x

Select a version

Filter by version or date

v09x
09.01.00.001 (31 Oct 2023)
v08x
08.02.00.24 (19 May 2022)
v07x
07.03.00.005 (29 May 2021)
v06x
06.03.00.106 (18 Apr 2020)
06.01.00.08 (19 Oct 2019)
06.00.00.07 (07 Jul 2019)
v05x
05.03.00.07 (06 Apr 2019)
05.02.00.10 (19 Dec 2018)
05.01.00.11 (05 Oct 2018)
05.00.00.15 (25 Jul 2018)
v04x

Latest version Version: 09.01.00.001 Release date: 31 Oct 2023

Release notes View software details

Downloads Supported products & hardware

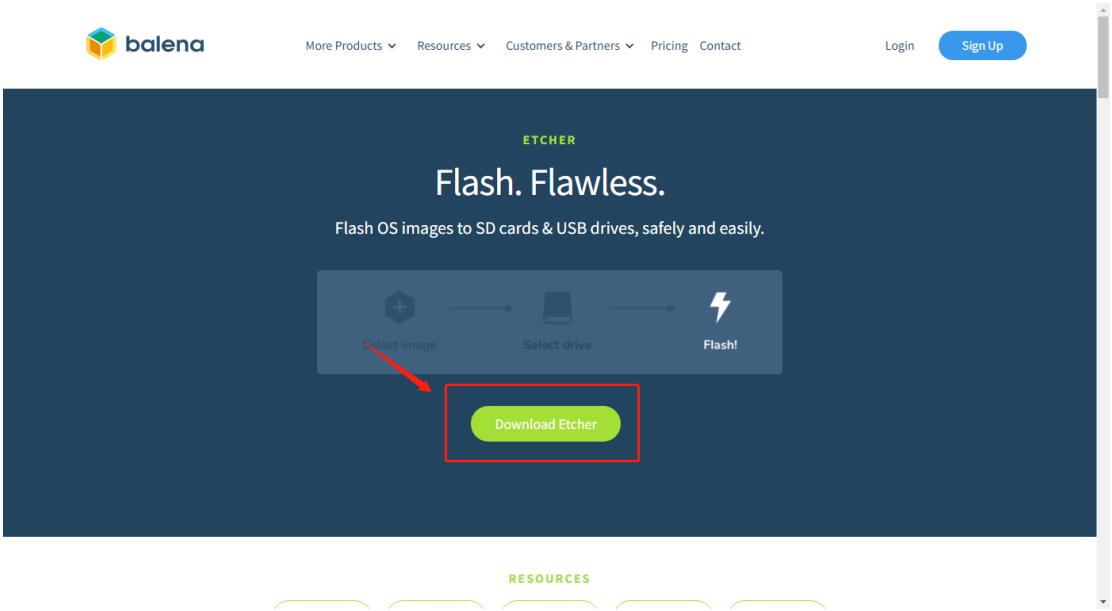
ti-processor-sdk-linux-am335x-evm-09.01.00.001-Linux-x86-Install.bin — 4087939 K
Processor SDK LINUX AM335x - Linux Installer for sources, pre-built binaries and file system images generated using Yocto build environment
MD5 checksum d63b24b043504b3a1656ce5d9b2d023b

tisdk-default-image-am335x-evm.wic.xz — 482219 K
Processor SDK LINUX AM335x Yocto - SD card image
MD5 checksum bc6d028c4e6490bcc533ee1be73fd70a

AM335x Linux CI/CD Snapshot for developers — 0 K
Get access to latest bug fixes and feature enhancements with Yocto build environment

5.2.4.2. Install SD Card Programming Utility

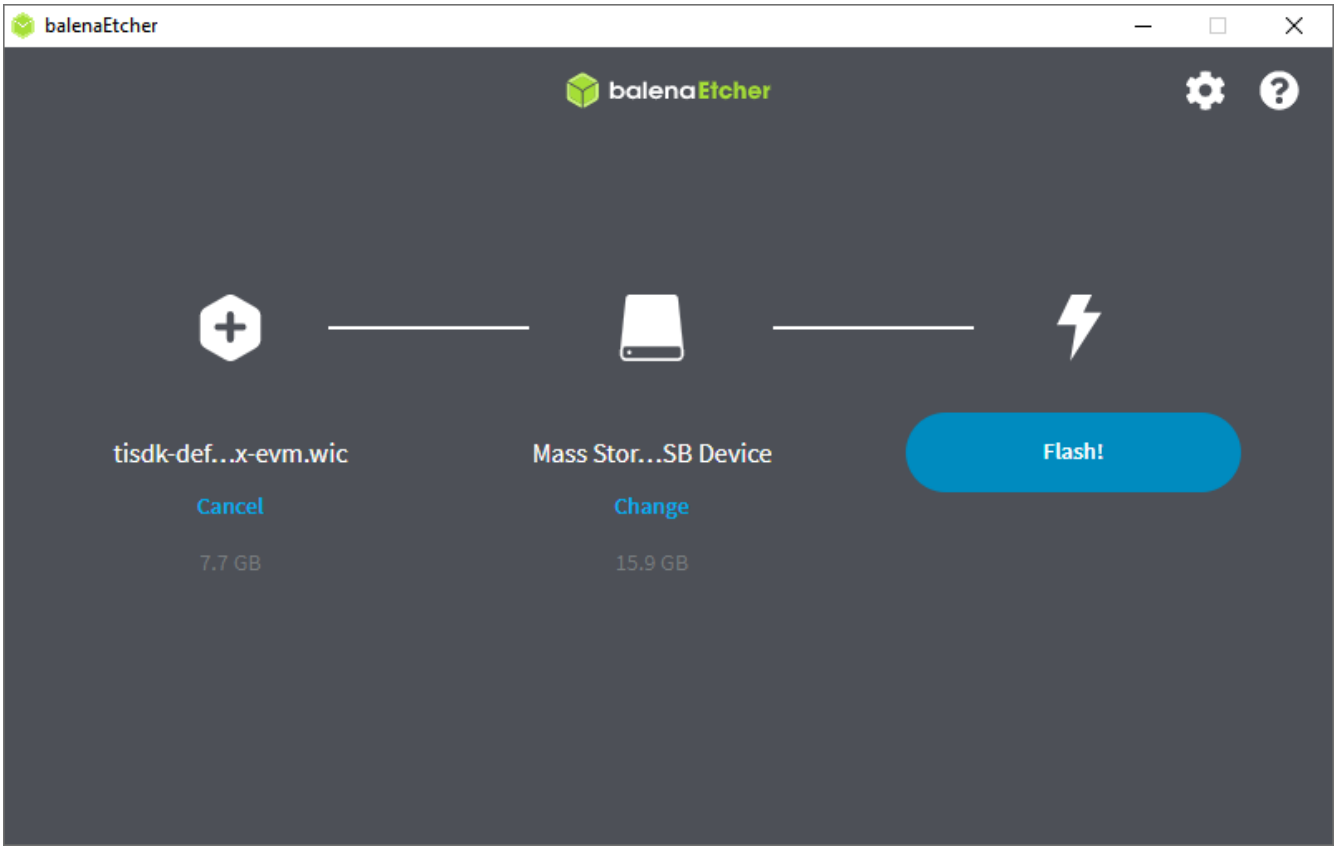
Download and install balenaEtcher.



5.2.4.3. Flash prebuilt AM335x WIC image on the SD Card

Use your computer's SD slot or a USB adapter to connect the SD card to your computer and write the image to your SD card. Etcher will transparently decompress the image on-the-fly before writing it to the SD card.

Select the image downloaded in the previous step, then select the target disk and click Flash.



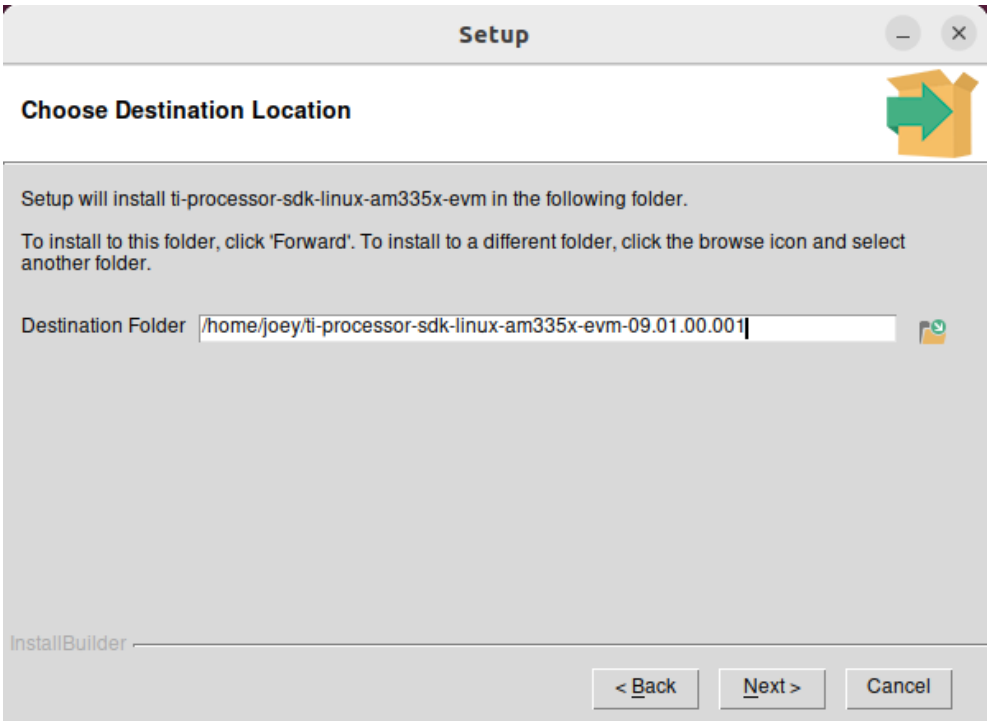
5.2.4.4. Install the processor Linux SDK for AM335x

Install the ti-processor-sdk-linux-am335x-evm-09.01.00.001-Linux-x86-Install.bin from:
https://www.ti.com/tool/download/PROCESSOR-SDK-LINUX-AM335X/09.01.00.001

Run the SDK installer.

```
joey@joey-vm:~$ ./ti-processor-sdk-linux-am335x-evm-09.01.00.001-Linux-x86-Install.bin
```

Install with the default Settings.

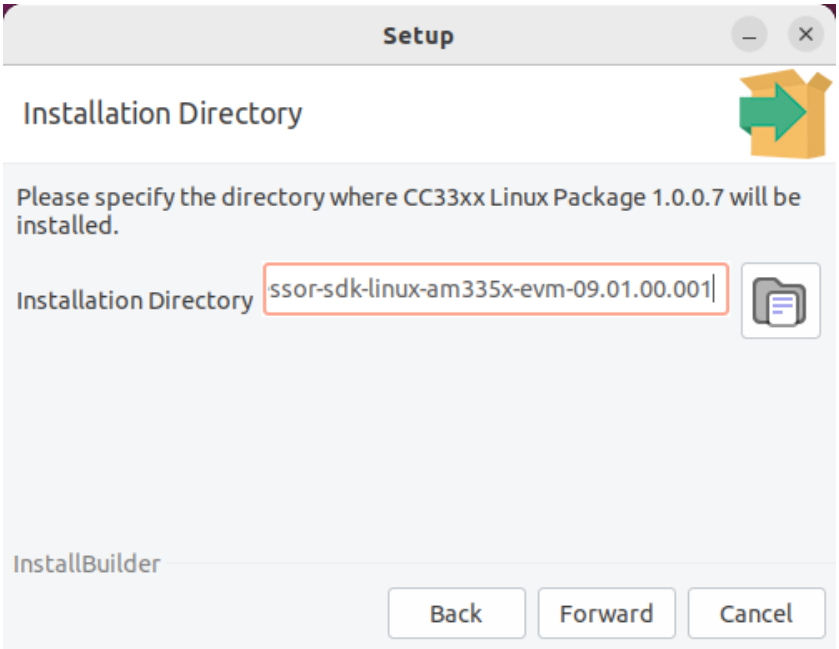


5.2.4.5. Run the CC33xx Linux Installer

Make sure you have execution permission for the file. Install the contents at the processor SDK root folder (i.e. where the Processor SDK is installed).

- Execute the following commands:

```
$ ./cc33xx-linux-package-1.0.0.7-linux-x64-installer.run
```



5.2.4.6. Apply the kernel patches, build the kernel and then install the binaries

- Execute the following commands.

```
$ apt-get install libgmp3-dev libmpc-dev
$ cd <processor SDK root folder>/cc33xx
$ ./build_cc33xx.sh  patch_kernel
$ ./build_cc33xx.sh  build_kernel
$ ./build_cc33xx.sh  install_kernel
```

5.2.4.7. Copy the rootfs to the SD Card

Un-plug and re-plug micro SD card. Mount rootfs/ partition of the SD card, then copy and paste rootfs/ binaries with contents of cc33xx/cc33xx_rootfs.

- Execute the following commands.

```
$ cd <processor SDK root folder>/cc33xx
$ sudo cp -r cc33xx_rootfs/* /media/<user>/root/
$ sudo cp -r cc33xx_rootfs/boot/* /media/<user>/boot/
$ sync
```

5.2.4.8. Power Up and Boot

Plug SD card into the BeagleBone Black and power on the BeagleBone Black.

```
root@am335x-evm:~# ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 90:06:f2:55:0a:99  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 195  bytes 249398 (243.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 195  bytes 249398 (243.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 34:68:b5:87:fc:99  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

By now, you should have successfully enabled the module on BeagleBone Black. For example scripts, refer to WiFi Fundamentals.

5.3. MPU and Linux Evaluation: AM62 Platform

5.3.1. Description

Below section describes step-by-step instructions on bringing up the module with TI's low-cost AM62x Sitara™ MPU.

5.3.2. Prerequisites

5.3.2.1. Hardware

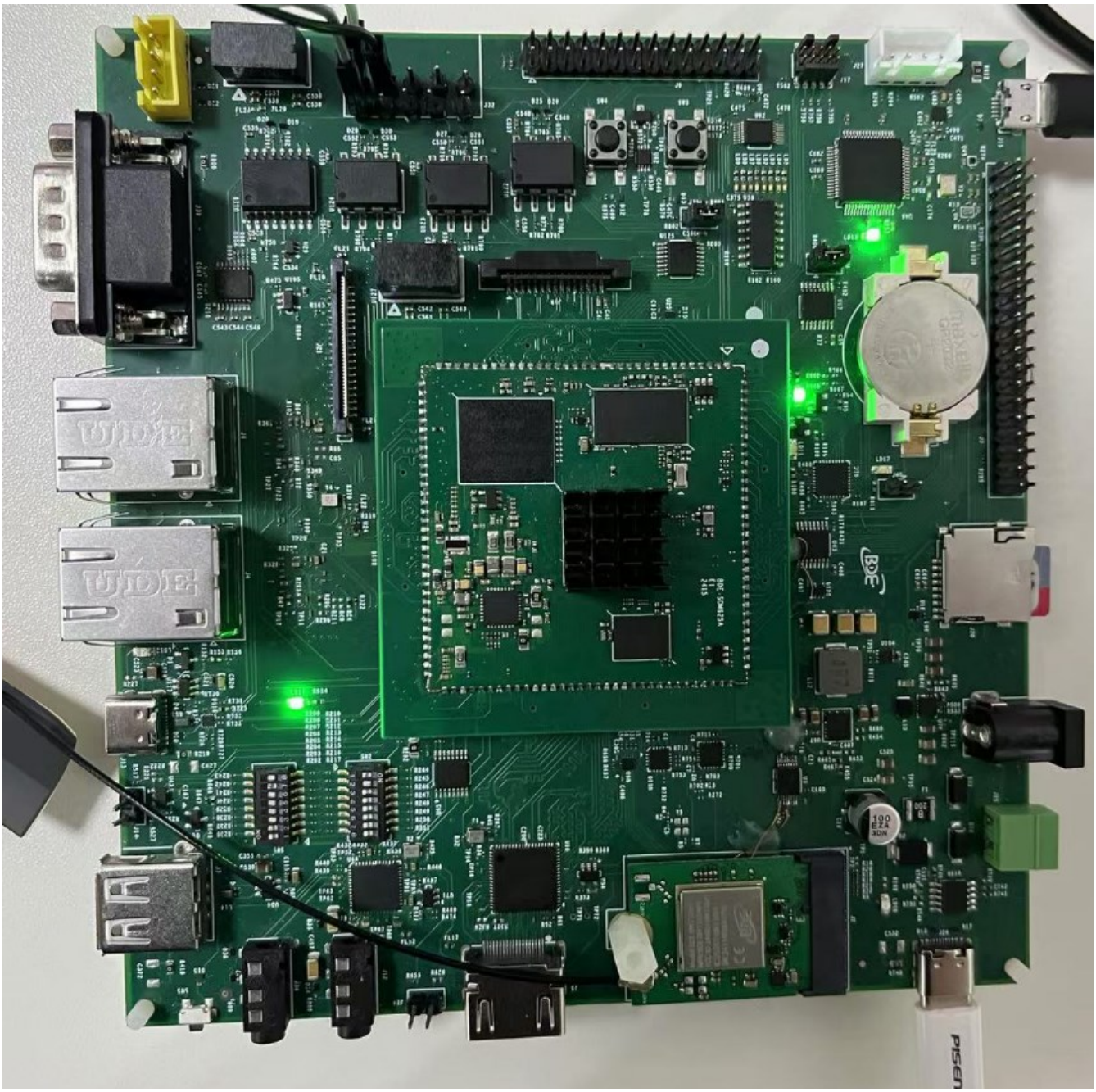
- BDE-BW3301NP1M2, the M.2 card with BDE-BW3301NP1 module (can be different depending on the target module);
- BDE-AM62x-EB, evaluation board for AM62x SOM;
- BDE-SOM62x, System-on-Module based on AM62x.

5.3.2.2. Software and Tools

- Ubuntu 22.04;
- Processor SDK Linux for AM62X (09.02.01.10);
- SD card image (09.02.01.10);
- balenaEtcher;
- CC33xx device driver source for MPUs running Linux OS (1.0.0.7).

5.3.2.3. Hardware Setup

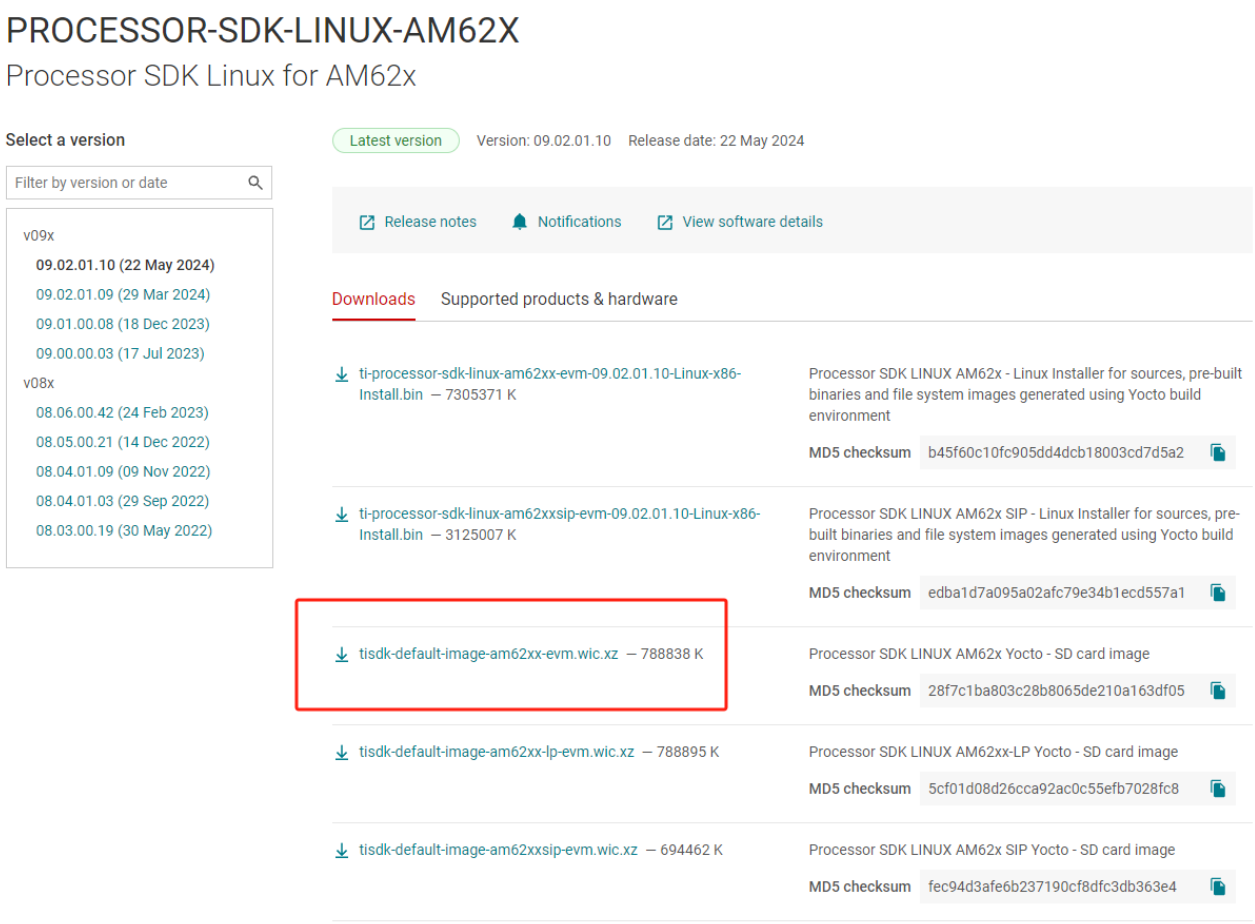
Plug the M.2 card to the M.2 card slot of the BDE-AM62x-EB board.



5.3.3. Steps

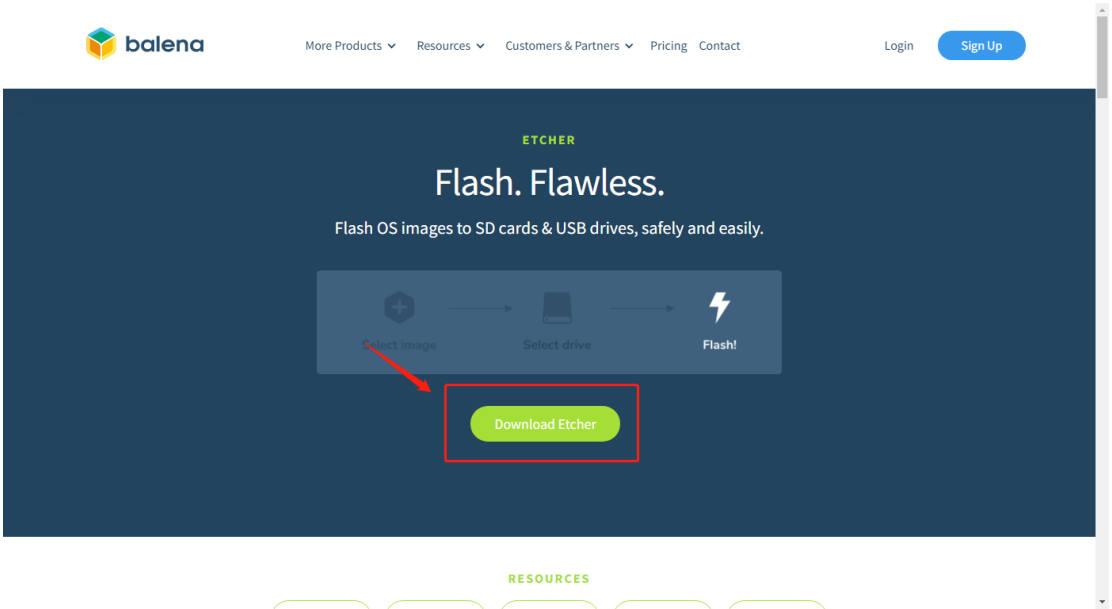
5.3.3.1. Download the SD Card Image

Download the SD card image with below link:
https://www.ti.com/tool/download/PROCESSOR-SDK-LINUX-AM62X/09.02.01.10



5.3.3.2. Install SD Card Programming Utility

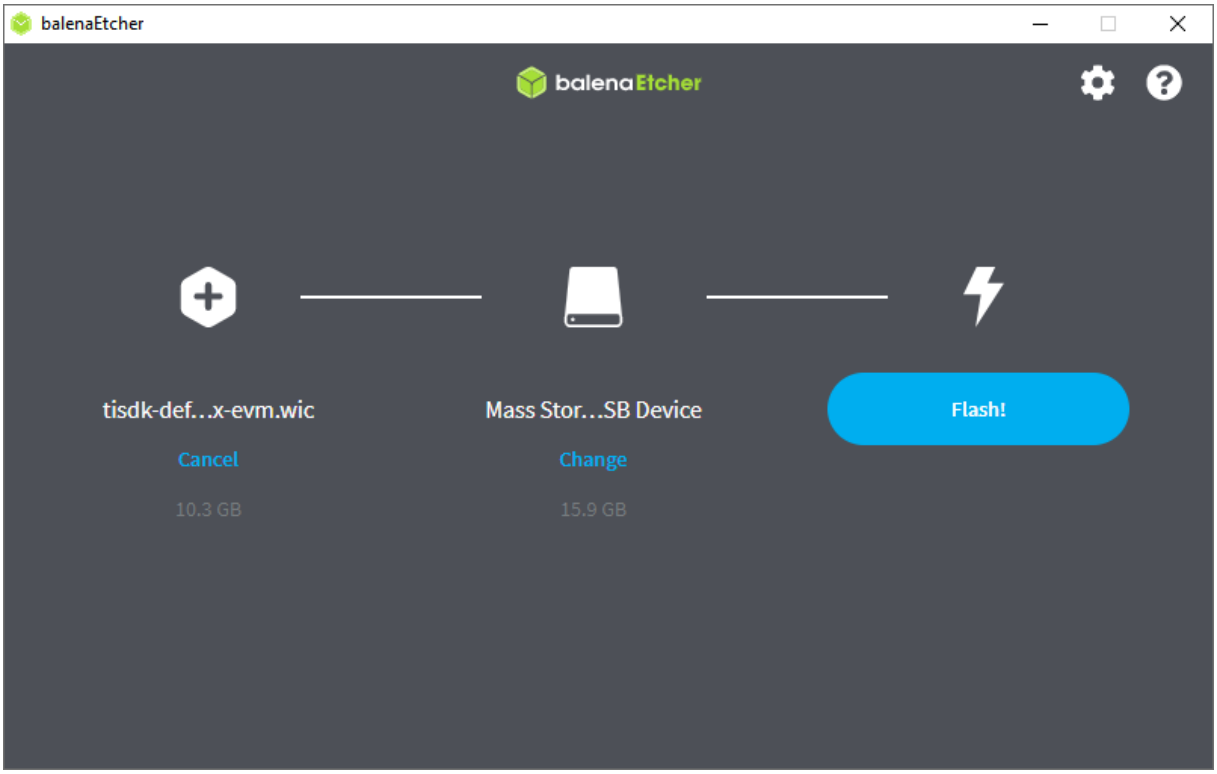
Download and install balenaEtcher.



5.3.3.3. Flash prebuilt AM62x WIC image to the SD Card

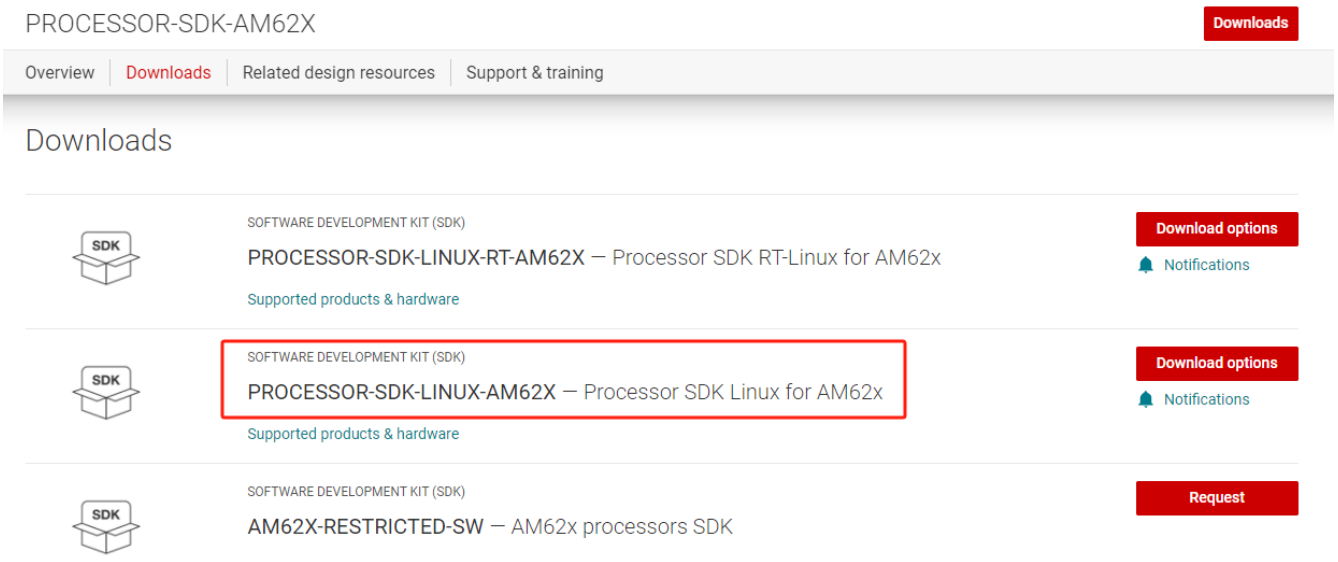
Use your computer's SD slot or a USB adapter to connect the SD card to your computer and flash the image to your SD card. Etcher will transparently decompress the image on-the-fly before writing it to the SD card.

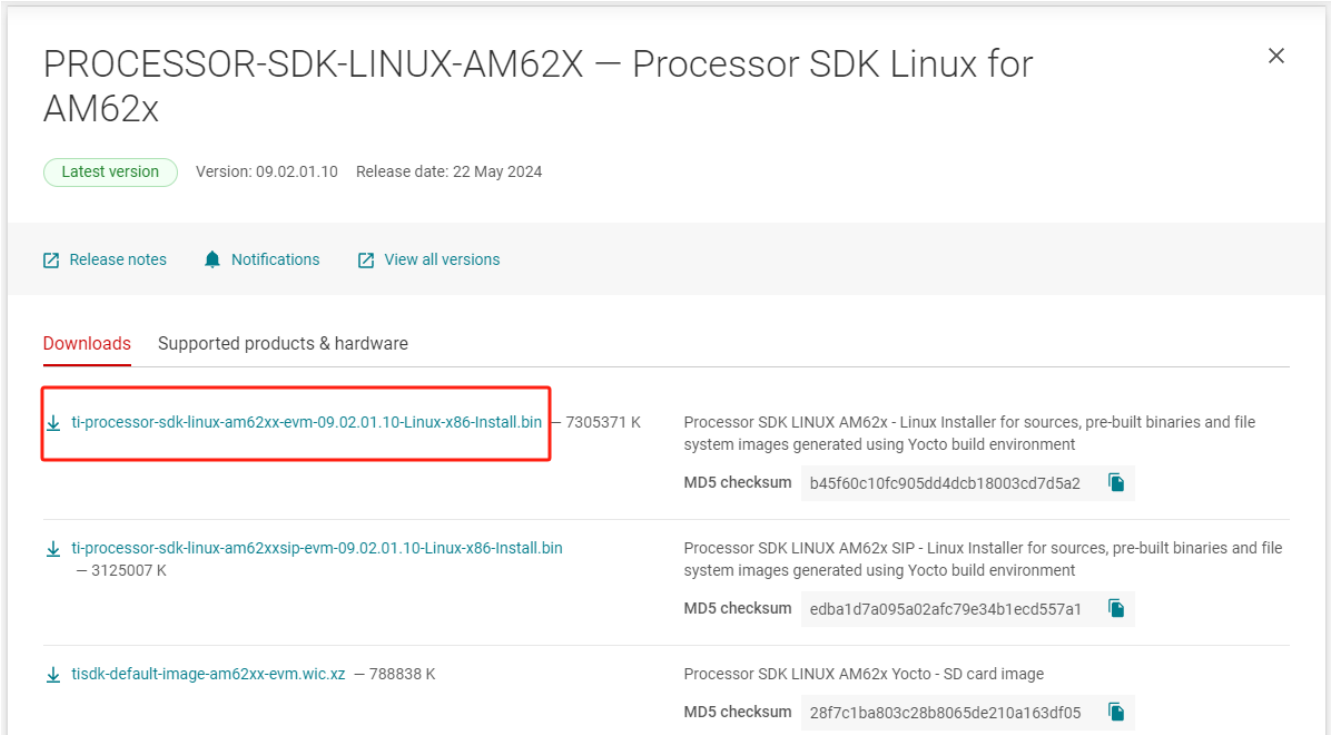
Select the image downloaded in the previous step, then select the target disk and click Flash.



5.3.3.4. Download and Install Linux SDK for AM62x

Download the SDK with below link and install: https://www.ti.com/tool/PROCESSOR-SDK-AM62X.

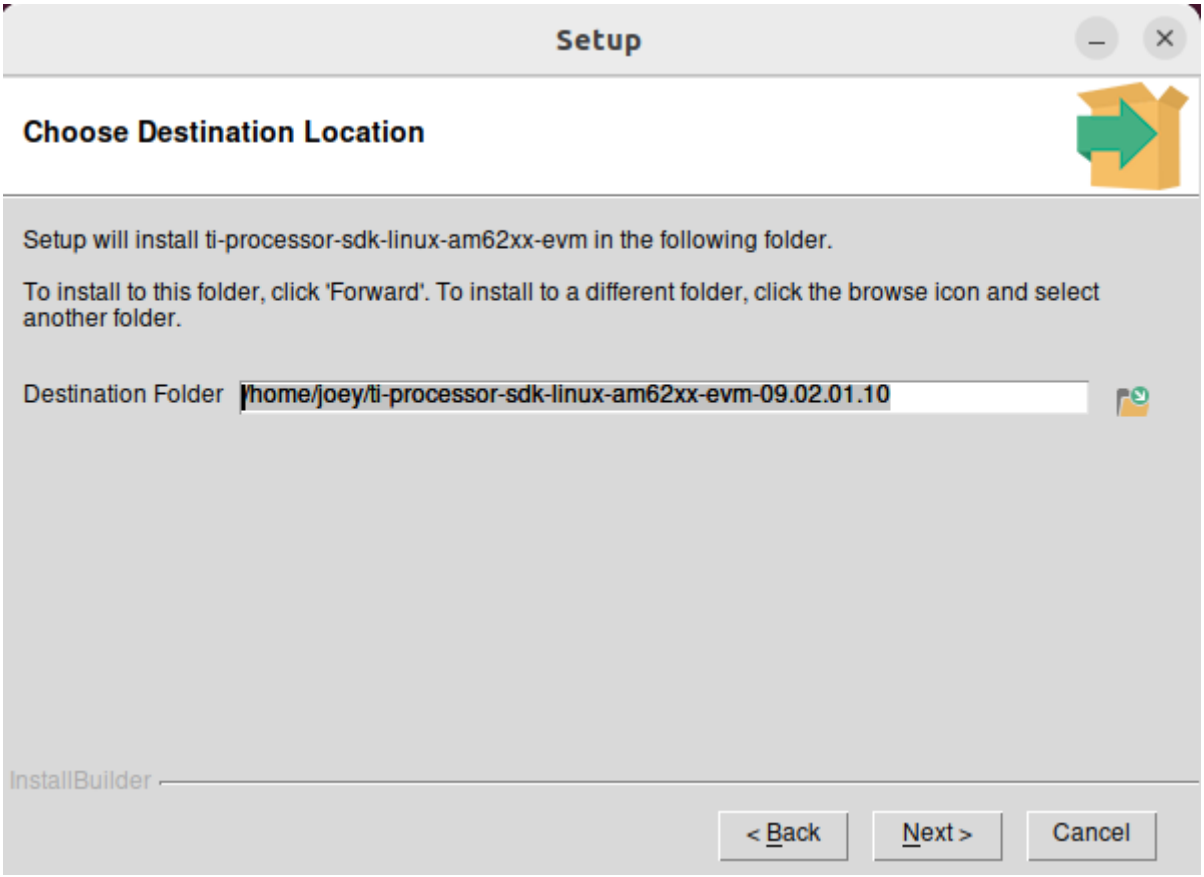




Run the SDK installer.

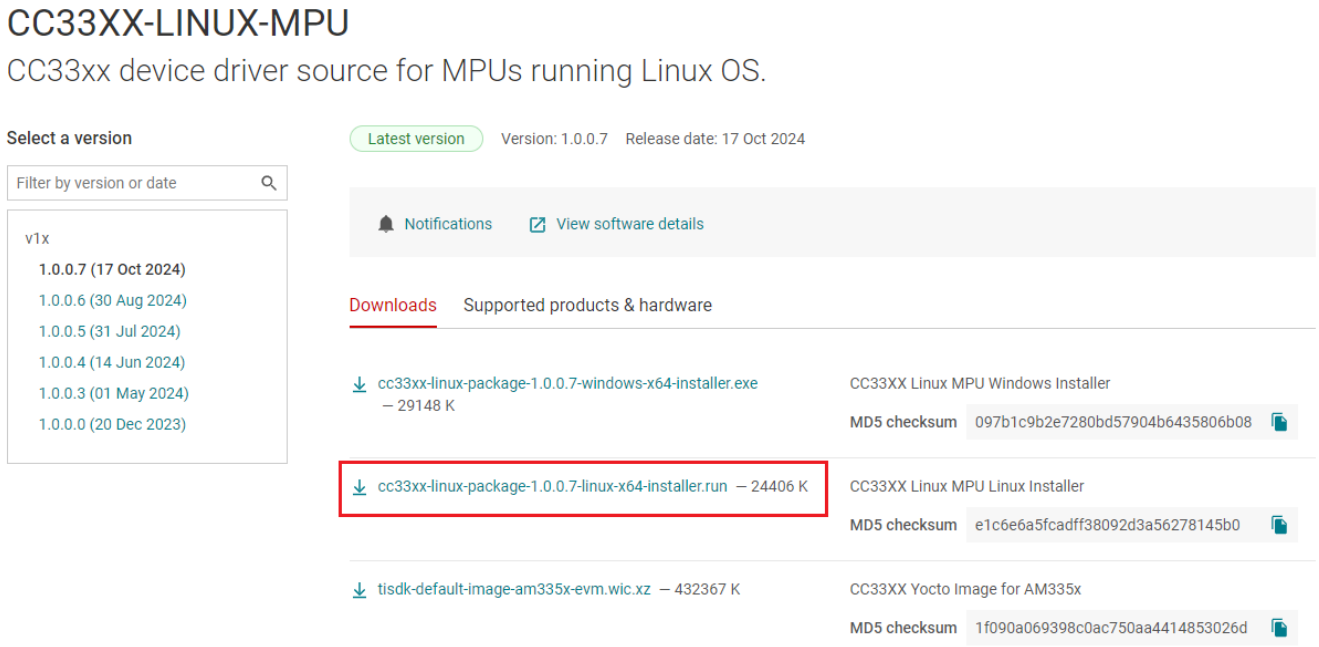
```
$ ./ti-processor-sdk-linux-am62xx-evm-09.02.01.10-Linux-x86-Install.bin
```

Install with the default Settings.



5.3.3.5. Download and Install CC33xx Device Driver Source

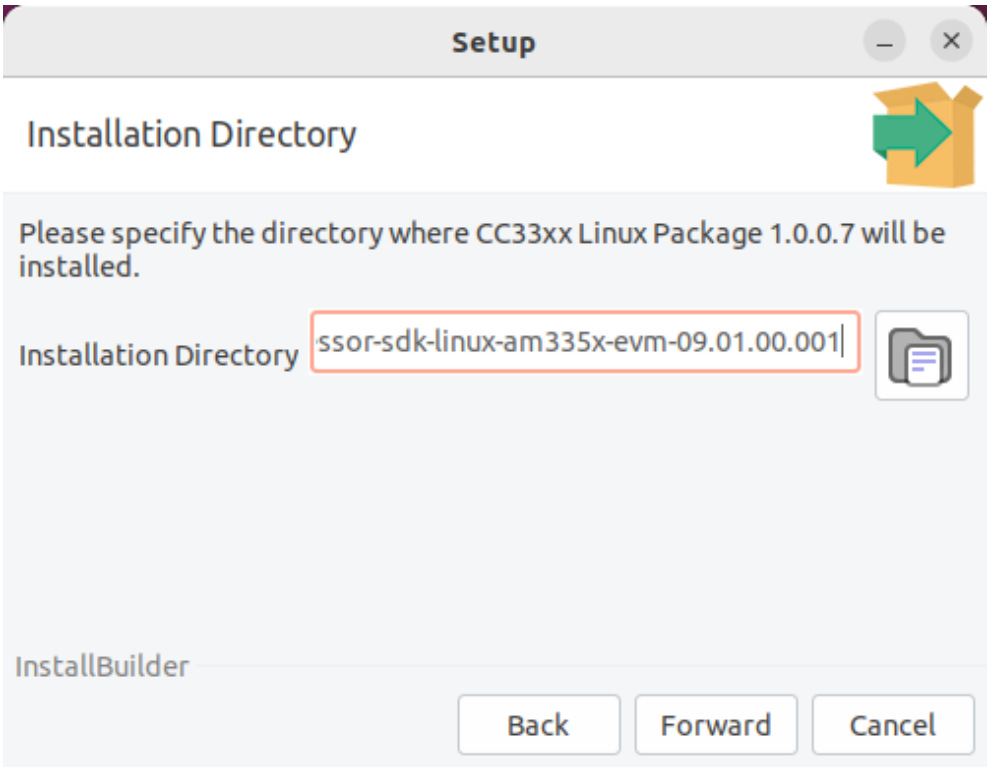
Download the CC33xx device drivers source with below link:
https://www.ti.com/tool/download/CC33XX-LINUX-MPU



Run the SDK installer.

```
$ ./cc33xx-linux-package-1.0.0.7-linux-x64-installer.run
```

Install with the default Settings.



5.3.3.6. Apply Patches and Build

(1) Navigate to the TI Linux kernel source directory.

```
$ cd ~/ti-processor-sdk-linux-am62xx-evm-09.02.01.10/board-support/ti-linux-kernel-6.1.83+gitAUTOINC+c1c2f1971f-ti/
```

(2) Patch the TI Linux kernel with the CC33XX patches.

```
$ patch -p0 < ~/ti-processor-sdk-linux-am335x-evm-09.01.00.001/cc33xx/patches/cc33xx_kernel.patch
$ patch -p0 < ~/ti-processor-sdk-linux-am335x-evm-09.01.00.001/cc33xx/patches/cc33xx_am625-sk_dts.patch
$ patch -p0 < ~/ti-processor-sdk-linux-am335x-evm-09.01.00.001/cc33xx/patches/cc33xx_ti_arm64_config.patch
```

(3) Add the SDK's toolchain to the PATH environment variable and clean the kernel sources.

```
$ export PATH=~/ti-processor-sdk-linux-am62xx-evm-09.02.01.10/linux-devkit/sysroots/x86_64-arago-linux/usr/bin/aarch64-oe-linux:$PATH
$ make ARCH=arm64 CROSS_COMPILE=aarch64-oe-linux- distclean
```

(4) Build the kernel, modules, and device tree.

```
$ make ARCH=arm64 CROSS_COMPILE=aarch64-oe-linux- defconfig ti_arm64_prune.config
$ make ARCH=arm64 CROSS_COMPILE=aarch64-oe-linux- Image modules dtbs -j8
```

5.3.3.7. Install Kernel Modules and dtbs

(1) Insert SD card into Host PC (Ubuntu). If the am62x MPU you are using is the GP version, perform following commands.

```
$ sudo cd /media/<user>/boot/
$ sudo cp tiboot3.bin tiboot3.bin.bak
$ sudo cp tiboot3-am62x-gp-evm.bin tiboot3.bin
```

(2) Navigate to the ti linux kernel source directory.

```
$ cd ~/ti-processor-sdk-linux-am62xx-evm-09.02.01.10/board-support/ti-linux-kernel-6.1.83+gitAUTOINC+c1c2f1971f-ti/
```

(3) Install the kernel, modules, and DTBs.

```
$ sudo cp arch/arm64/boot/Image /media/<user>/root/boot/
$ sudo make ARCH=arm64 modules_install INSTALL_MOD_PATH=/media/<user>/root
$ sudo cp arch/arm64/boot/dts/ti/k3-am625-sk.dtb /media/<user>/root/boot/dtb/ti/
```

5.3.3.8. Add FW and Scripts

(1) Copy contents under "cc33xx_rootfs" directory into SD card's root directory and merge. Execute the following commands.

```
$ cd ~/ti-processor-sdk-linux-am335x-evm-09.01.00.001/cc33xx/cc33xx_rootfs
$ sudo cp -r * /media/<user>/root/
$ sync
```

5.3.3.9. Power Up and Boot

Remove SD Card from SD card slot of PC or the adaptor, and insert it to the SD card slot of BDE-AM62-EB board, power on the board and wlan0 interface should now be up.

```
root@am62xx-evm:~# ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 1c:63:49:27:75:60  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 5a:42:fb:1a:2f:86  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 93  bytes 7714 (7.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 93  bytes 7714 (7.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 34:68:b5:87:f7:dd  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

By now, you should have successfully enabled the module on AM62x. For example scripts, refer to WiFi Fundamentals.

5.4. MPU and Linux Evaluation: iMX93 Platform

5.4.1. Description

This guide describes how to enable the BDE-BW3301NP1 module with BDE-SOM93V.

5.4.2. Get Ready

5.4.2.1. Hardware

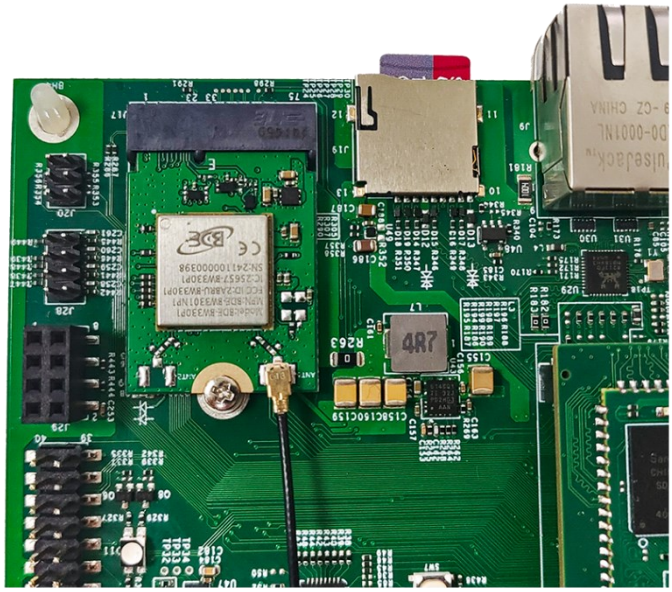
- BDE-BW3301NP1M2 M.2 card, the M.2 card with BDE-BW3301NP1 module (can be different depending on the target module);
- BDE-SOM93V-EB, evaluation board for SOM module BDE-SOM93V;
- BDE-SOM93V, System-on-Module based on NXP i.MX93.

5.4.2.2. Software and Tools

- Ubuntu 22.04;
- The i.MX family Linux Board Support Package (Linux 6.1.55_2.2.0);
- balenaEtcher;
- CC33xx device driver source for MPUs running Linux OS (1.0.0.4);
- Toolchain.

5.4.2.3. Hardware Setup

Plug the M.2 card to the BDE-SOM93V evaluation board.

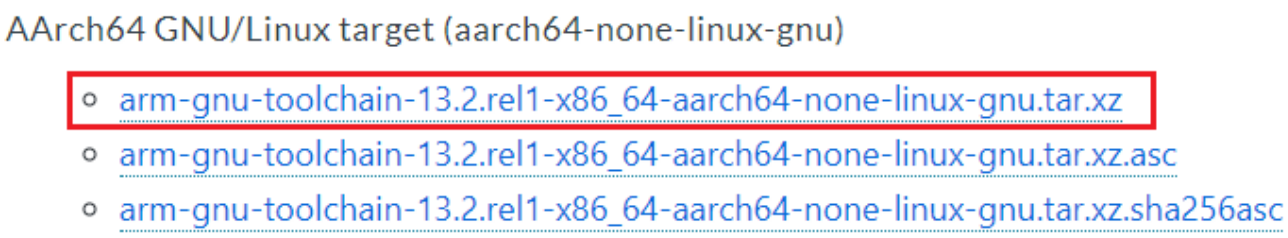


5.4.3. Steps

Follow below steps to bring up the hardware.

5.4.3.1. Download the Toolchain

(1) Go to this link and select the toolchain of the following version;



(2) Copy the downloaded package into Ubuntu through the shared folder and execute the following commands to extract the toolchain to the /usr/local/arm directory. If the arm directory does not exist, create a new one;

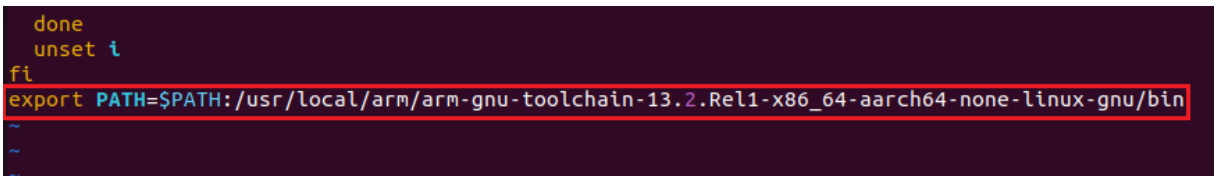
```
$ sudo mkdir /usr/local/arm
$ tar -xvf arm-gnu-toolchain-13.2.rel1-x86_64-aarch64-none-linux-gnu.tar.xz -C /usr/local/arm
```

(3) Add the path of the toolchain to the environment variables;

```
$ sudo vim /etc/profile
```

(4) Add the following path at the end of the file;

```
export PATH=$PATH:/usr/local/arm/arm-gnu-toolchain-13.2.Rel1-x86_64-aarch64-none-linux-gnu/bin
```



(5) After saving and exiting, use the following command to make the configuration take effect;

```
$ source /etc/profile
```

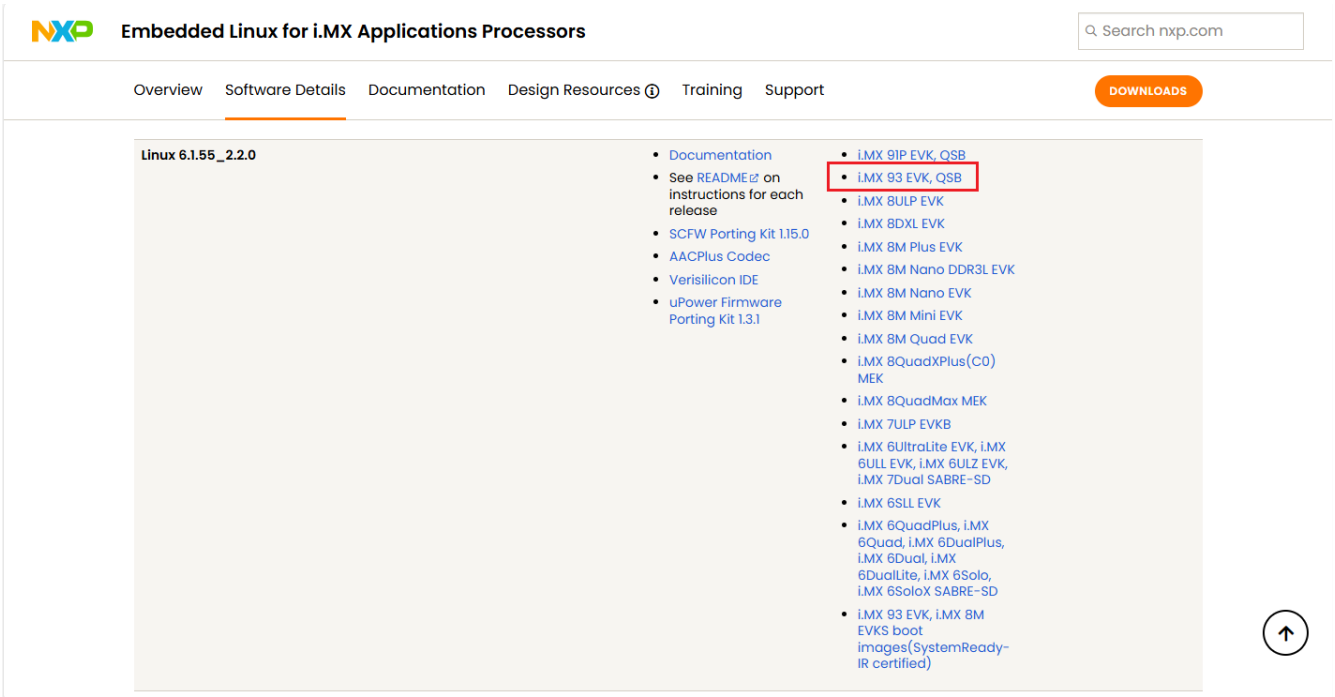
(6) Use the following command to check the version information of the toolchain.

```
$ aarch64-none-linux-gnu-gcc -v
```

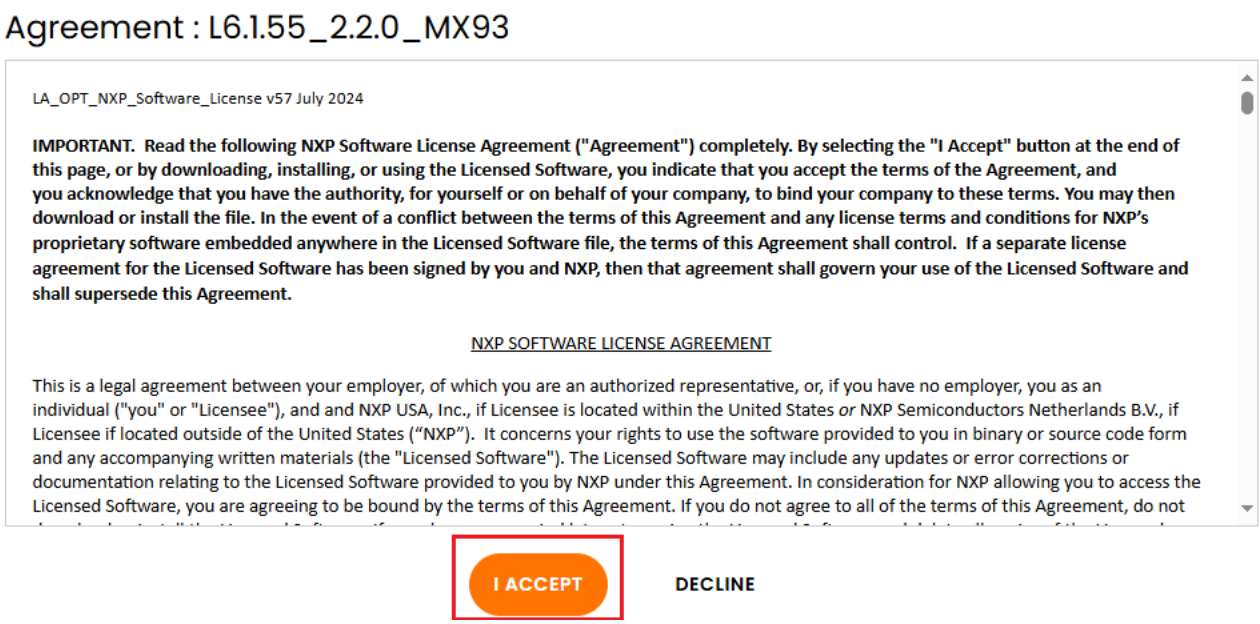
```
joey@joey-vm:~$ aarch64-none-linux-gnu-gcc -v
Using built-in specs.
COLLECT_GCC=aarch64-none-linux-gnu-gcc
COLLECT_LTO_WRAPPER=/usr/local/arm/arm-gnu-toolchain-13.2.Rel1-x86_64-aarch64-no
ne-linux-gnu/bin/../libexec/gcc/aarch64-none-linux-gnu/13.2.1/lto-wrapper
Target: aarch64-none-linux-gnu
Configured with: /data/jenkins/workspace/GNU-toolchain/arm-13/src/gcc/configure
--target=aarch64-none-linux-gnu --prefix= --with-sysroot=/aarch64-none-linux-gnu
/libc --with-build-sysroot=/data/jenkins/workspace/GNU-toolchain/arm-13/build-aa
rch64-none-linux-gnu/install//aarch64-none-linux-gnu/libc --with-bugurl=https://
bugs.linaro.org/ --enable-gnu-indirect-function --enable-shared --disable-libssp
 --disable-libmudflap --enable-checking=release --enable-languages=c,c++,fortran
 --with-gmp=/data/jenkins/workspace/GNU-toolchain/arm-13/build-aarch64-none-linu
x-gnu/host-tools --with-mpfr=/data/jenkins/workspace/GNU-toolchain/arm-13/build-
aarch64-none-linux-gnu/host-tools --with-mpc=/data/jenkins/workspace/GNU-toolcha
in/arm-13/build-aarch64-none-linux-gnu/host-tools --with-isl=/data/jenkins/works
pace/GNU-toolchain/arm-13/build-aarch64-none-linux-gnu/host-tools --enable-fix-c
ortex-a53-843419 --with-pkgversion='Arm GNU Toolchain 13.2.rel1 (Build arm-13.7)
'
Thread model: posix
Supported LTO compression algorithms: zlib
gcc version 13.2.1 20231009 (Arm GNU Toolchain 13.2.rel1 (Build arm-13.7))
```

5.4.3.2. Download the i.MX family Linux Board Support Package

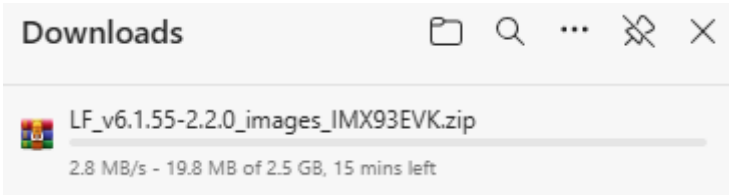
(1) Download the i.MX family Linux Board Support Package through the link;



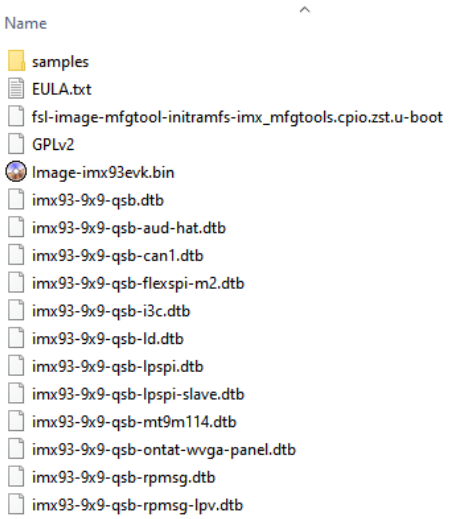
(2) Click on the "i.MX 93 EVK, QSB" link;



(3) Click on the "I ACCEPT";

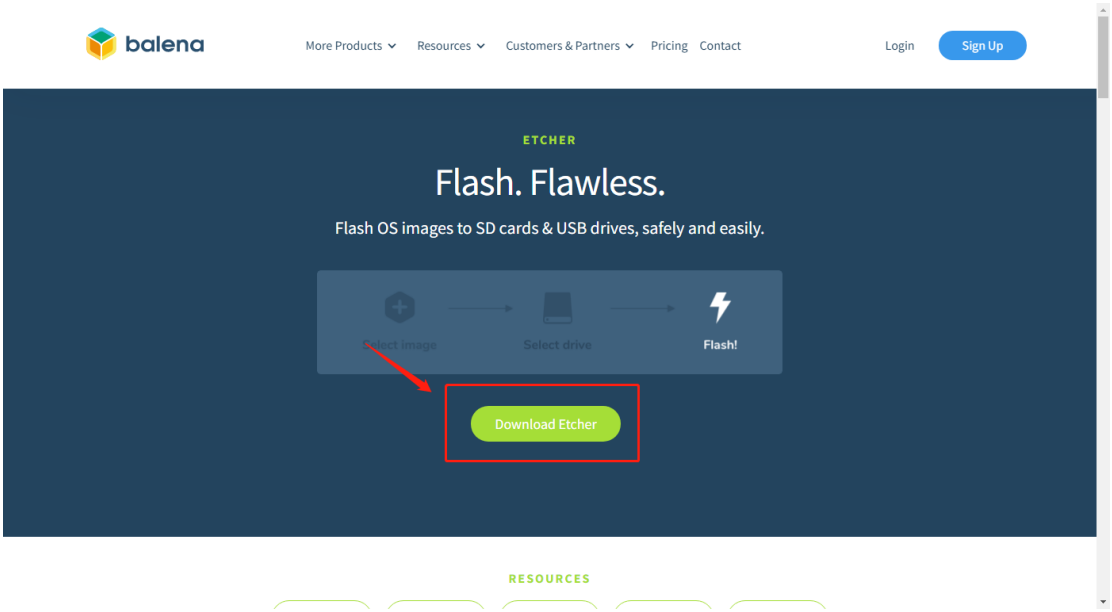


(4) Wait for the download. Extract the file after the download is completed.



5.4.3.3. Install SD Card Programming Utility

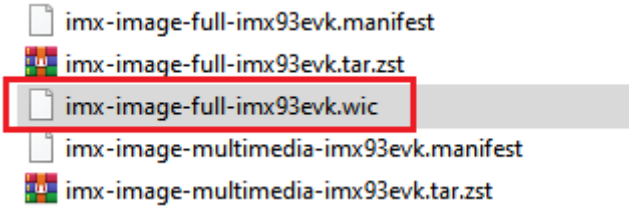
Download and install balenaEtcher.



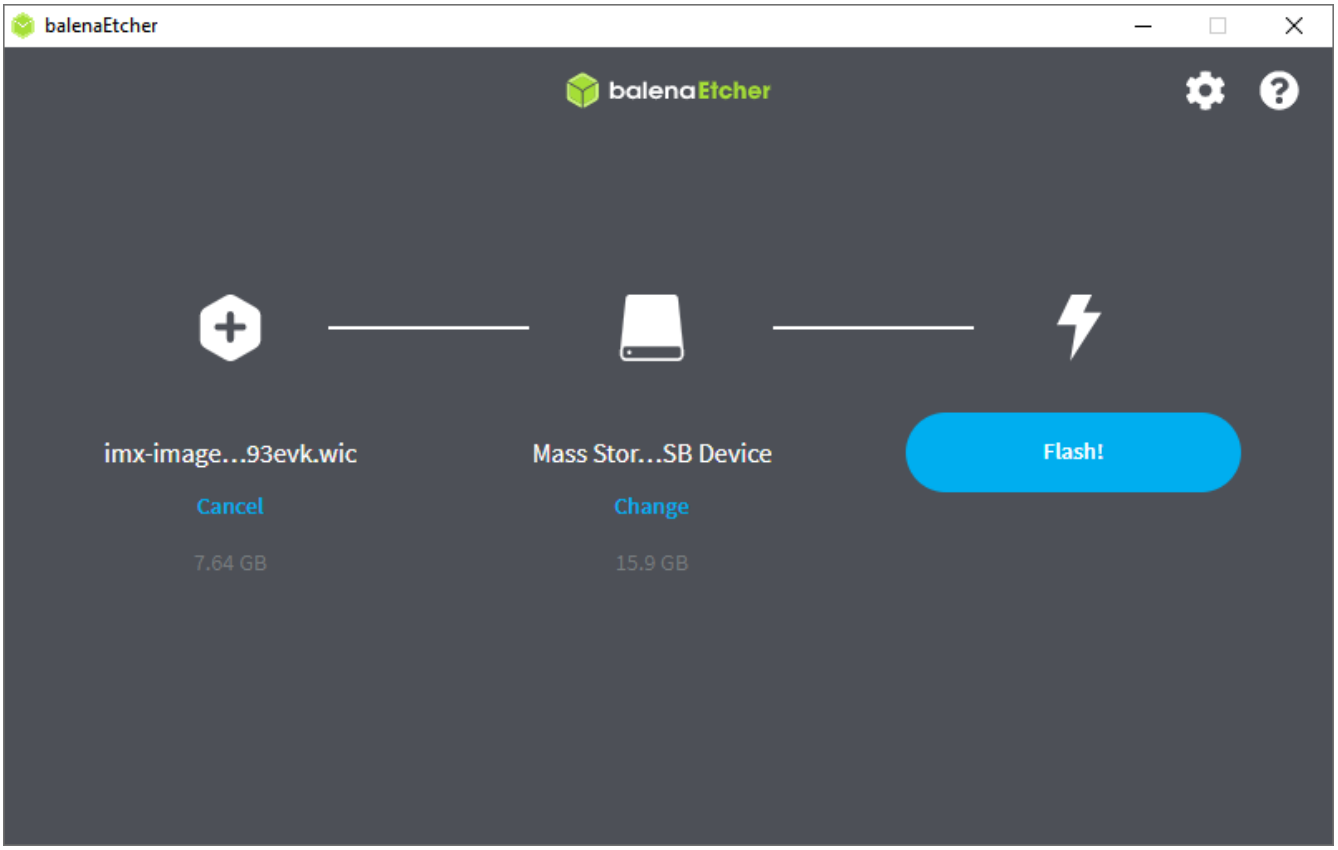
5.4.3.4. Flash prebuilt IMX93 WIC image on the SD Card

(1) Use your computer's SD slot or a USB adapter to connect the SD card to your computer and write the image to your SD card;

(2) Select the "imx-image-full-imx93evk.wic" file in the i.MX family Linux Board Support Package folder;

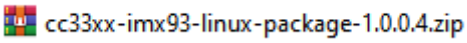


(3) Select the target disk and click the "Flash" button (Here "Flash" can be understood as "write" or similar meanings, depending on the actual function of the software interface).

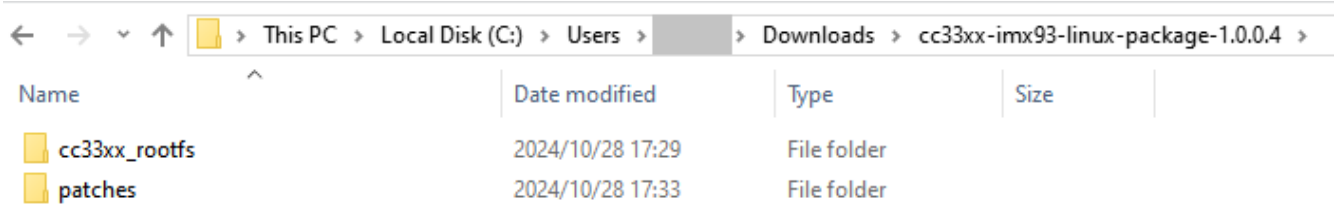


5.4.3.5. Download and Install CC33xx device driver source

(1) Download the CC33xx device driver source at this link;



(2) Extract to obtain the following folders;



(3) Copy the "cc33xx-imx93-linux-package-1.0.0.4" folder and all the files under it to Ubuntu.

```
joey@joey-vm:~$ ls cc33xx-imx93-linux-package-1.0.0.4/
cc33xx_rootfs  patches
```

5.4.3.6. Clone the i.MX Linux Kernel

Clone the i.MX Linux Kernel repository and checkout the lf-6.1.y branch using the following commands in Ubuntu.

```
$ git clone https://github.com/nxp-imx/uboot-imx -b lf-6.1.55-2.2.0
```

5.4.3.7. Apply Patches and Build

(1) Navigate to the ti linux kernel source directory;

```
$ cd linux-imx/
```

(2) Patch the i.MX linux kernel with the CC33XX driver and DTS patches;

```
$ patch -p0 < <path_to_cc33xx_folder>/patches/cc33xx_kernel.patch
$ patch -p0 < <path_to_cc33xx_folder>/patches/cc33xx_imx93_dts.patch
$ patch -p0 < <path_to_cc33xx_folder>/patches/imx_v8_defconfig.patch
```

(3) Build the kernel, modules, and devicetree.

```
$ make ARCH=arm64 CROSS_COMPILE=aarch64-none-linux-gnu- imx_v8_defconfig
$ make ARCH=arm64 CROSS_COMPILE=aarch64-none-linux-gnu- Image modules dtbs
-j8
```

5.4.3.8. Install Kernel Modules and dtbs

(1) Plug the SD card into the Ubuntu machine and install the kernel, modules, and DTBs on to the SD card;

```
$ sudo cp arch/arm64/boot/Image /media/<user>/root/boot/
$ sudo cp arch/arm64/boot/Image /media/<user>/boot/
$ sudo cp arch/arm64/boot/dts/freescale/imx93-11x11-evk.dtb
/media/<user>/root/boot/
$ sudo cp arch/arm64/boot/dts/freescale/imx93-11x11-evk.dtb
/media/<user>/boot/
$ sudo make ARCH=arm64 modules_install INSTALL_MOD_PATH=/media/<user>/root
```

(2) Install the CC33XX firmware and example scripts by copying the contents under “cc33xx_rootfs” into the SD card’s root directory.

```
$ sudo cp -r <path_to_cc33xx_folder>/cc33xx_rootfs/* /media/<user>/root
$ sync
```

5.4.3.9. Power Up and Boot

Eject the SD card from the host PC and plug it into the BDE-SOM93V evaluation board. Apply the USB-C power supply and open a serial terminal with baudrate 115200.

After the linux kernel boots, the CC33XX driver should now be loaded and the wlan0 interface should be shown in the output of ifconfig. For example scripts, refer to WiFi Fundamentals.

```
root@imx93evk:~# ifconfig wlan0
wlan0: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        ether 34:68:b5:87:f7:dd  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

6. Hardware Design Files

To access the hardware design files for BDE-BW33-EM, including the schematic, the PCB layout and the Bill of Materials, the user can submit a request on the bdecomm.com under the module product page. Request Now.

7. Ordering Information

Table 10. Ordering Information

Product Type	Orderable Part Number	Description	Note
Plug-in Evaluation Module	BDE-BW33-EM	Plug-in Evaluation Module for BDE-BW33xx Module Series	
Module Breakout Board	BDE-BW3301NP1-BO	Breakout Board with BDE-BW3301NP1 Module	
	BDE-BW3301UP1-BO	Breakout Board with BDE-BW3301UP1 Module	
	BDE-BW3301AP1-BO	Breakout Board with BDE-BW3301AP1 Module	
	BDE-BW3301NP2-BO	Breakout Board with BDE-BW3301NP2 Module	
	BDE-BW3301UP2-BO	Breakout Board with BDE-BW3301UP2 Module	
	BDE-BW3351NP1-BO	Breakout Board with BDE-BW3351NP1 Module	
	BDE-BW3351UP1-BO	Breakout Board with BDE-BW3351UP1 Module	
	BDE-BW3351AP1-BO	Breakout Board with BDE-BW3351AP1 Module	
	BDE-BW3351NP2-BO	Breakout Board with BDE-BW3351NP2 Module	
	BDE-BW3351UP2-BO	Breakout Board with BDE-BW3351UP2 Module	
	BDE-BW3301N1-BO	Breakout Board with BDE-BW3301N1 Module	
	BDE-BW3301U1-BO	Breakout Board with BDE-BW3301U1 Module	
	BDE-BW3301A1-BO	Breakout Board with BDE-BW3301A1 Module	
	BDE-BW3351N1-BO	Breakout Board with BDE-BW3351N1 Module	
	BDE-BW3351U1-BO	Breakout Board with BDE-BW3351U1 Module	
	BDE-BW3351A1-BO	Breakout Board with BDE-BW3351A1 Module	
M.2 Card	BDE-BW3301NP1M2	M.2 Card with BDE-BW3301NP1 Module	
	BDE-BW3300NP1M2	M.2 Card with BDE-BW3300NP1 Module	
	BDE-BW3301NP2M2	M.2 Card with BDE-BW3301NP2 Module	
	BDE-BW3300NP2M2	M.2 Card with BDE-BW3300NP2 Module	
	BDE-BW3351NP1M2	M.2 Card with BDE-BW3351NP1 Module	
	BDE-BW3350NP1M2	M.2 Card with BDE-BW3350NP1 Module	
	BDE-BW3351NP2M2	M.2 Card with BDE-BW3351NP2 Module	
	BDE-BW3350NP2M2	M.2 Card with BDE-BW3350NP2 Module	
Adaptor Board	BDE-BW33-BBB	Adaptor board for BDE-BW33-EM to BeagleBone Black	
MCU/MPU Evaluation Board	LP-AM243	TI's Sitara™ high performance MCU AM243x Launchpad	Purchased from TI.com
	BEAGL-BONE-BLACK	BeagleBone Black Board Based on AM335	Purchased from TI.com
	BDE-SOM62x-EB	Carrier Board for BDE SoM BDE-SOM625A	

Product Type	Orderable Part Number	Description	Note
	BDE-SOM62x	System on Module Based on AM625	
	BDE-SOM93V-EB	Carrier Board for BDE SoM BDE-SOM93V	
	BDE-SOM93V	System on Module Based on i.MX93	

8. Revision History

Table 11. Revision History

Revision	Date	Description
V1.0	2024-10-25	First Release

9. Additional Information

9.1. Trademarks

SimpleLink™, BoosterPack™, LaunchPad™ are trademarks of Texas Instruments.
Wi-Fi 6® and Wi-Fi® are registered trademarks of Wi-Fi Alliance.
Bluetooth® is a registered trademark of Bluetooth SIG, Inc.
Linux® is a registered trademark of Linux Foundation.
BeagleBoard.org® and BeagleBone® are registered trademarks of BeagleBoard.org Foundation.
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
Website: www.bdecomm.com Email: info@bdecomm.com

China: B2-403, 162 Science Avenue, Huangpu District, Guangzhou 510663, China
Tel: +86-20-28065335
Website: www.bdecomm.com Email: info@bdecomm.com