UM11483

Getting Started with NXP-based Wireless Modules on i.MX 8M Quad EVK Running Linux OS

Rev. 4 — 21 September 2021 User manual

Document information

Information	Content
Keywords	i.MX 8M Quad Evaluation Kit (EVK), 88W8987-based wireless modules, 88W8997-based wireless modules, 88W9098-based wireless module, and IW416-based wireless module
Abstract	Details the bring-up of Wi-Fi and Bluetooth on NXP-based wireless modules connected with i.MX 8M Quad EVK running Linux OS

Revision history

Rev	Date	Description
v.1	20201015	Initial version
v.2	20210121	**Modifications** • Extended the scope to 88W8987- and 88W8997-based wireless modules • Updated the document title and overall document structure
v.3	20210612	**Modifications** • Section 1.2 "References": updated • Extended the scope to 88W9098-based wireless module: – Section 3.9 "88W9098-based Murata module LBEE6ZZ1": added – Section 4.4.3 "Enabling PCIe on M.2 (88W9098-based Murata module LBEE6ZZ1)": added – Section 5.4 "Bring-up of 88W9098-based Murata module LBEE6ZZ1": added • Extended the scope to 88W8997-based wireless modules – Section 3.8 "88W8997-based AzureWave module AW-CM276-uSD": added – Section 3.7 "88W8997-based AzureWave module AW-CM276MA-SUR": added – Section 5.3 "Bring-up of 88W8997-based AzureWave module (AW-CM276MA-SUR and AW-CM276-uSD)": added • Section 3.5.6 "Enabling SDIO on M.2 (AW-CM358MA and AW-CM276MA-SUR)": updated • Section 6 "Bring-up of Bluetooth interfaces": updated
v.4	20210921	**Modifications** Extended the scope to IW416-based wireless module • Section 1.1 "Purpose and scope": updated • Section 1.2 "References": updated • Section 3 "NXP-based wireless modules": updated • Section 3.10 "IW416-based AzureWave AW-AM510MA module": added • Section 4.4.2 "Enabling SDIO on M.2 connector": added • Section 5.5 "Bring-up of IW416-based AzureWave module AW-AM510MA": added • Section 6 "Bring-up of Bluetooth interfaces": updated Other modifications • Section 1.3 "Wi-Fi driver and firmware support on past Linux BSP releases": added • Replaced the sections *Setting up the host* and *Building the image* with Section 4.2 "Building the image from source" • Section 4.3 "Flashing the image to eMMC": updated *UUU usage* paragraph • Section 4.4.3 "Enabling PCIe on M.2 (88W9098-based Murata module LBEE6ZZ1)": updated

1 About this document

1.1 Purpose and scope

This document details the enabling of wireless solutions on i.MX 8M Quad evaluation kit. The i.MX 8M Quad EVK is powered by Linux and the NXP Linux drivers are used for NXP-based wireless modules. The manual covers the bring-up of i.MX 8M Quad EVK, the configurations for the BSP image and the hardware connection with NXP-based wireless modules. The later chapters describe how to bring up the Wi-Fi and Bluetooth interfaces.

This document specifies the hardware interconnection and software support for the i.MX 8M Quad evaluation kit and NXP-based wireless modules to enable Wi-Fi/Bluetooth functionality. Note that the AzureWave modules AW-CM358-uSD and AW-CM276-uSD only support Wi-Fi with the i.MX 8M Quad EVK.

Once you have enabled the Wi-Fi and/or Bluetooth interfaces, refer to the user manual UM11490 that details the Wi-Fi and Bluetooth features and configurations for i.MX 8M Quad EVK.

Note: *This document does not provide a detailed description of the i.MX 8M Quad BSP nor how to generate an image and rootfs as these topics are covered in* i.MX Yocto Project User's Guide*.*

1.2 References

Table 1. References

Reference type	Description
Datasheet	NXP - 88W8987 - 2.4/5 GHz Dual-Band 1x1 Wi-Fi 5 and Bluetooth 5 Solution - Short data sheet (public) (link)
Datasheet	NXP - 88W8997 - Dual-band 2x2 Wi-Fi 5 and Bluetooth 5 Solution - Data sheet (confidential) (link)
Datasheet	NXP - 88W9098 - Wi-Fi 6 2x2 Concurrent Dual Wi-Fi (CDW) and Bluetooth 5.1 Combo SoC - Data sheet (confidential) (link)
Datasheet	NXP - IW416 - Dual-band 1x1 Wi-Fi 4 and Bluetooth 5.1 Combo SoC - Data sheet (public) (link)
Datasheet	AzureWave - AW-CM358SM - IEEE 802.11a/b/g/n/ac WLAN with Bluetooth 5 Combo Stamp LGA Module (link)
Datasheet	AzureWave - AW-CM276MA-PUR - IEEE 802.11a/b/g/n/ac Wireless LAN 2T2R and Bluetooth 5.0 Combo Module (M.2 2230) (link)
Data sheet	AzureWave - AW-CM276MA-SUR - IEEE 802.11a/b/g/n/ac Wireless LAN 2T2R and Bluetooth 5.0 Combo Module (M.2 2230) (link)
Data sheet	AzureWave - AW-AM510MA - IEEE 802.11a/b/g/n Wireless LAN + Bluetooth 5. 1 Combo Module (M.2 2230) (link)
Data sheet	Murata - LBEE6ZZ1 module specification (link)
Fact sheet	NXP - FS - 88W8997-802.11ac wave 2 2x2 Wi-Fi Dual-band with Bluetooth 5 SoC (link)
Fact sheet	NXP - Evaluation Kit Based on i.MX 8M Quad Application Processors (link)
User guide	AzureWave - AW-CM358-uSD - uSD-1212 Adapter Board for AW-AM281-uSD and AW-CM358-uSD (link)
User manual	NXP - UM - i.MX Yocto Project User's Guide (link).
User manual	NXP - UM - i.MX Linux® User's Guide (link).
User manual	NXP - UM11490 - Feature Configuration Guide for NXP-based Modules with i.MX 8M Quad EVK (link)
Web page	NXP - 88W8987: 2.4/5 GHz Dual-Band 1x1 Wi-Fi® 5 (802.11ac) + Bluetooth® 5 Solution (link)
Web page	NXP - 88W8997: 2.4/5 GHz Dual-Band 2x2 Wi-Fi® 5 (802.11ac) + Bluetooth® 5 Solution (link)
Web page	NXP - 88W9098: 2.4/5 GHz Dual-Band 2x2 Wi-Fi® 6 (802.11ax) + Bluetooth® 5.1 (link)
Web page	NXP - IW416: 2.4/5 GHz Dual-Band 1x1 Wi-Fi® 4 (802.11n) + Bluetooth® 5.1 Solution (link)

1.3 Wi-Fi driver and firmware support on past Linux BSP releases

Table 2. Wi-Fi driver and firmware support on past Linux BSP releases

Linux BSP release	Wi-Fi driver version	Firmware version	Chipset support	Host interfaces
Linux 5.10.35_2.0.0	MM5X17247.p5-MGPL	17.92.5.p3	88W9098	PCIE (Wi-Fi) UART (Bluetooth)
		16.92.10.p213	88W8997	PCIE (Wi-Fi) UART (Bluetooth)
		16.92.10.p218	88W8997	SDIO (Wi-Fi) UART (Bluetooth)
		16.92.10.p210	88W8987	SDIO (Wi-Fi) UART (Bluetooth)
Linux 5.10.9_1.0.0	MXM5x16210-GPL	16.92.10.p118	88W8997	PCIE (Wi-Fi) UART (Bluetooth)
		16.92.10.p208	88W8987	SDIO (Wi-Fi) UART (Bluetooth)
Linux 5.4.70_2.3.0	MXM5x16210-GPL	16.92.10.p118	88W8997	PCIE(Wi-Fi) UART (Bluetooth)
		16.92.10.p208	88W8987	SDIO (Wi-Fi) UART (Bluetooth)

2 i.MX 8M Quad evaluation kit (EVK)

2.1 i.MX 8M Quad EVK overview

NXP i.MX 8M Quad Evaluation Kit (EVK) provides a platform for rapid evaluation of the i.MX 8MQuad, i.MX 8MDual and i.MX 8MQuadLite application processors, utilizing dual to quad Arm Cortex-A53s and single Cortex-M4 cores.

The EVK includes the hardware design files, tools and board support packages (BSPs) for Linux where:

- The i.MX 8M Quad Linux Board Support Package (BSP) supports the Linux Operating System (OS) on the i.MX 8M Quad application processors. The purpose of this software package is to enable Linux support on the i.MX 8M Quad family of Integrated Circuits (ICs) and their associated platforms. It provides the necessary software to interface the standard open-source Linux kernel to the i.MX 8M Quad hardware.
- The i.MX 8M Quad BSP is based on the latest long-term stable (LTS) version of the Linux kernel, which is enhanced with the features provided by NXP and can be accommodate customized Linux kernel configurations.

2.2 i.MX 8M Quad evaluation board

The i.MX 8M Quad evaluation board is based on the NXP i.MX 8M Quad application processor. The i.MX 8M Quad processor features an advanced implementation of the Quad Arm Cortex-A53+ ARM Cortex-M4 core which operates at speeds up to 1.5 GHz and 266 MHz respectively. The i.MX 8M Quad features an integrated power management module that reduces the complexity of an external power supply and simplifies the power sequencing. Each processor provides a 32-bit LVDDR3L/DDR4/ LPDDR4 memory interface and other interfaces to connect peripherals such as HDMI, LCD, Wi-Fi, Bluetooth, and camera sensors.

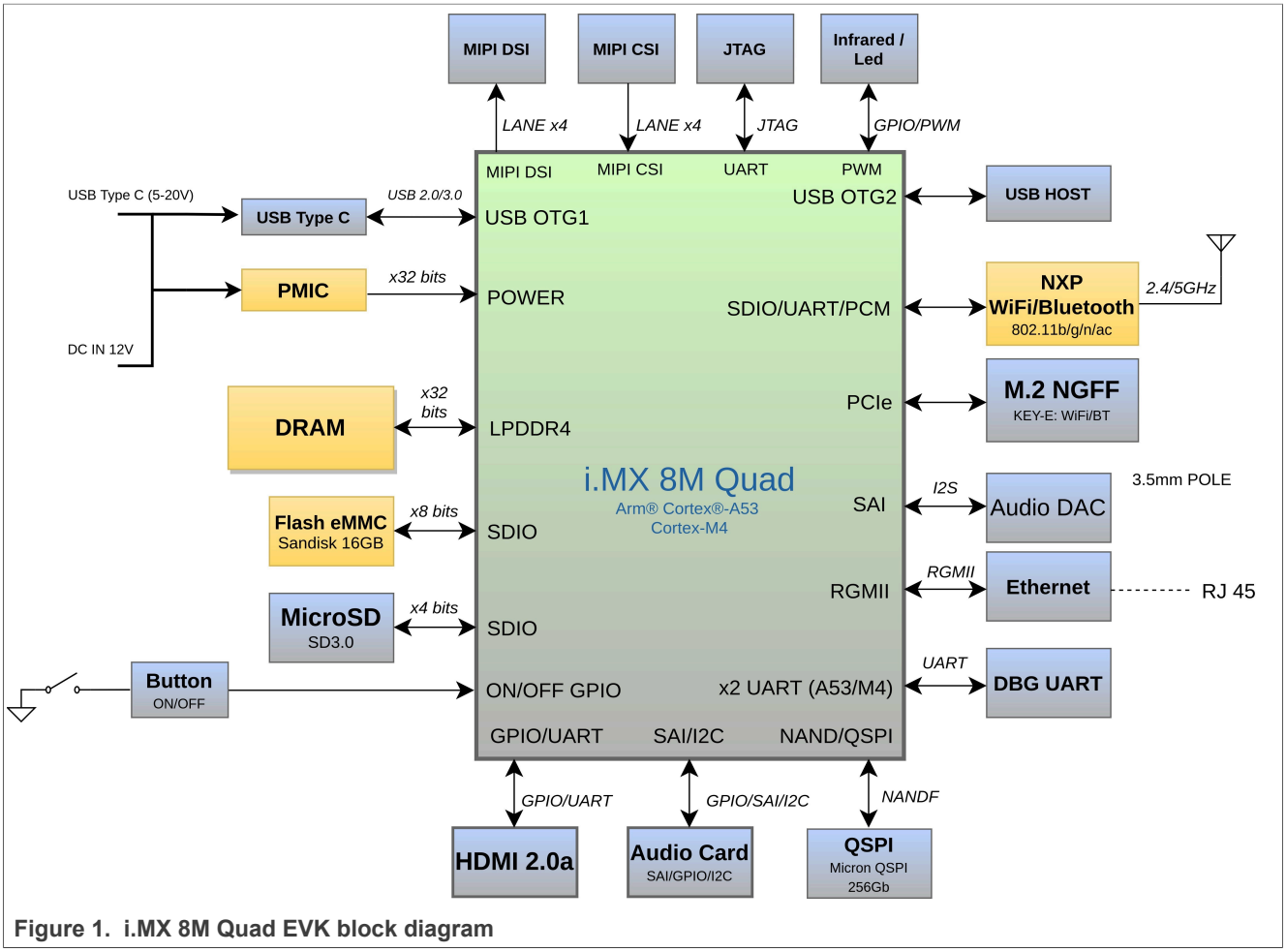


Figure 1. i.MX 8M Quad EVK block diagram

For more information about the application processor, please refer to the data sheet and reference manual on www.nxp.com.

Table 3 lists the features of i.MX 8M Quad EVK.

Table 3. Features of i.MX 8M Quad

Features	Features (continued)
i.MX 8M Quad applications processor with five cores (4×Arm® Cortex® -A53 and 1× Cortex-M4)	M.2 connector for Wi-Fi/Bluetooth (PCIe, USB, UART, I2C and I2S)
3 GB, 32-bit LPDDR4 with 1.6 GHz clock	USB3.0 Type-A connector
eMMC 5.0, 16 GB	HDMI2.0a Type-A connector
32 MB Octal SPI NOR flash	1 Gbit/s Ethernet
Micro SD card connector	Mini-SAS MIPI-DSI connector
USB3.0 Type-C connector with PD support	2x mini-SAS MIPI-CSI connectors for Camera
USB to serial converter for debug	Infrared receiver LEDs for power indication and general-purpose use
3.5 mm audio jack for amplified speakers	JTAG 10-pin connector

2.3 i.MX 8M Quad evaluation board interfaces

Figure 2 shows the front view of i.MX 8M Quad evaluation board with pointers to the interfaces.

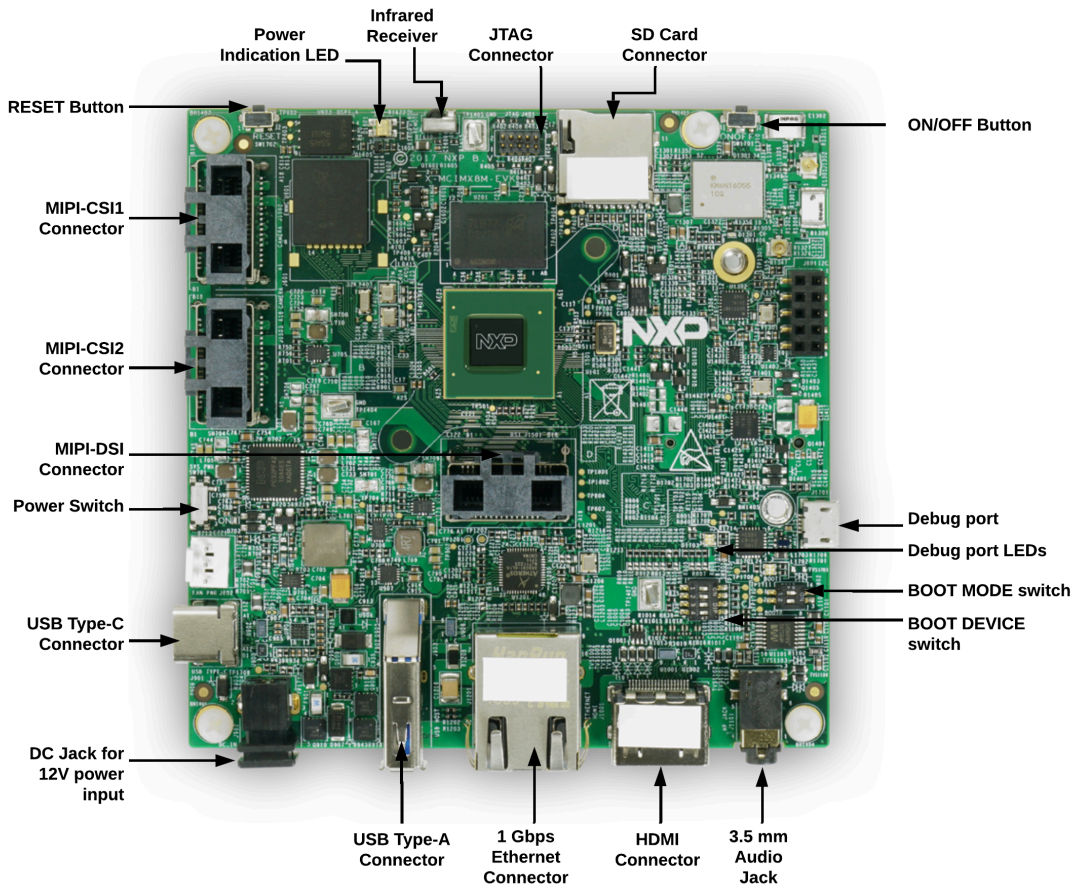


Figure 2. i.MX 8M Quad evaluation board interfaces - Front view

Figure 3 shows the back view of i.MX 8M Quad evaluation board with pointers to the interfaces.

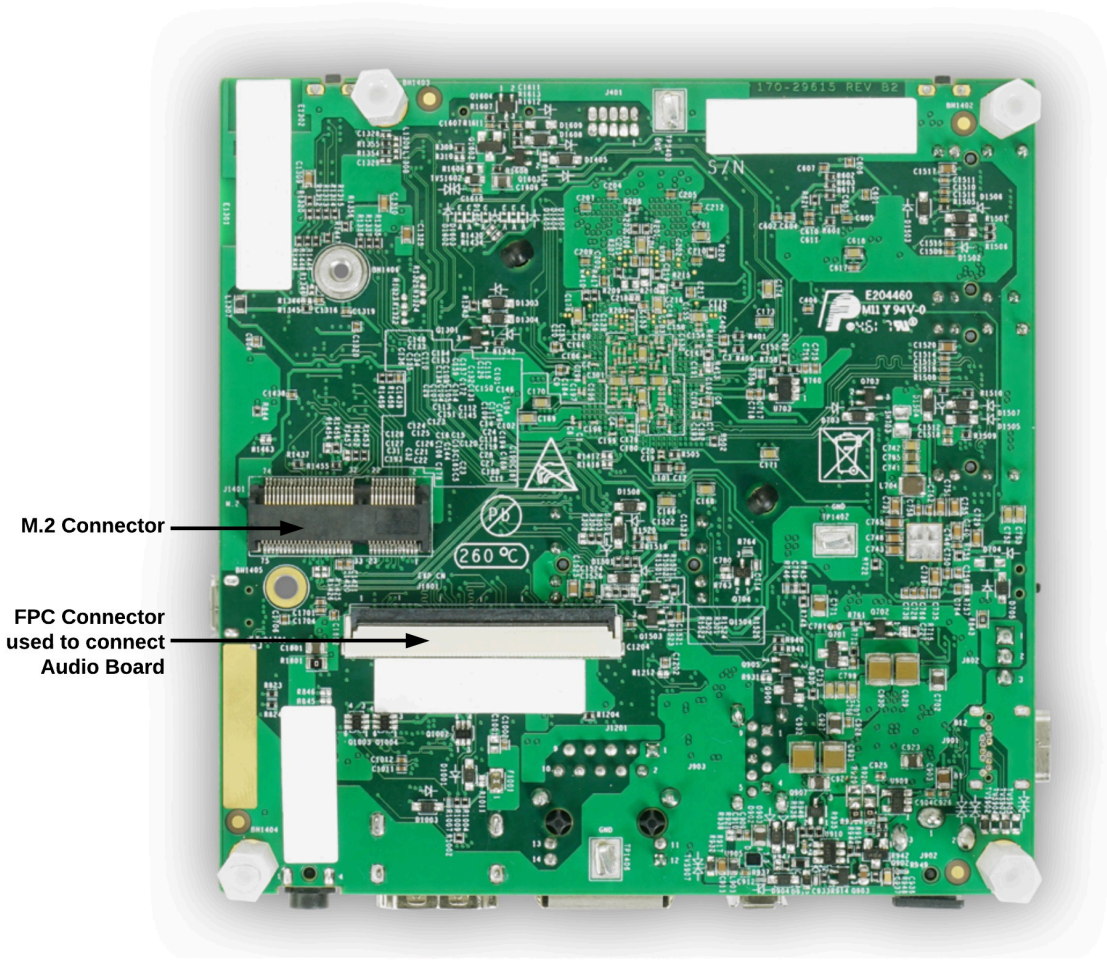


Figure 3. i.MX 8M Quad evaluation board interfaces - Back view

2.4 i.MX 8M Quad switch settings

Figure 4 shows the two switches on i.MX 8M Quad evaluation board. The *Boot Device Switch* is used to select the boot drive while the *Boot Mode Switch* is used to set the boot mode.

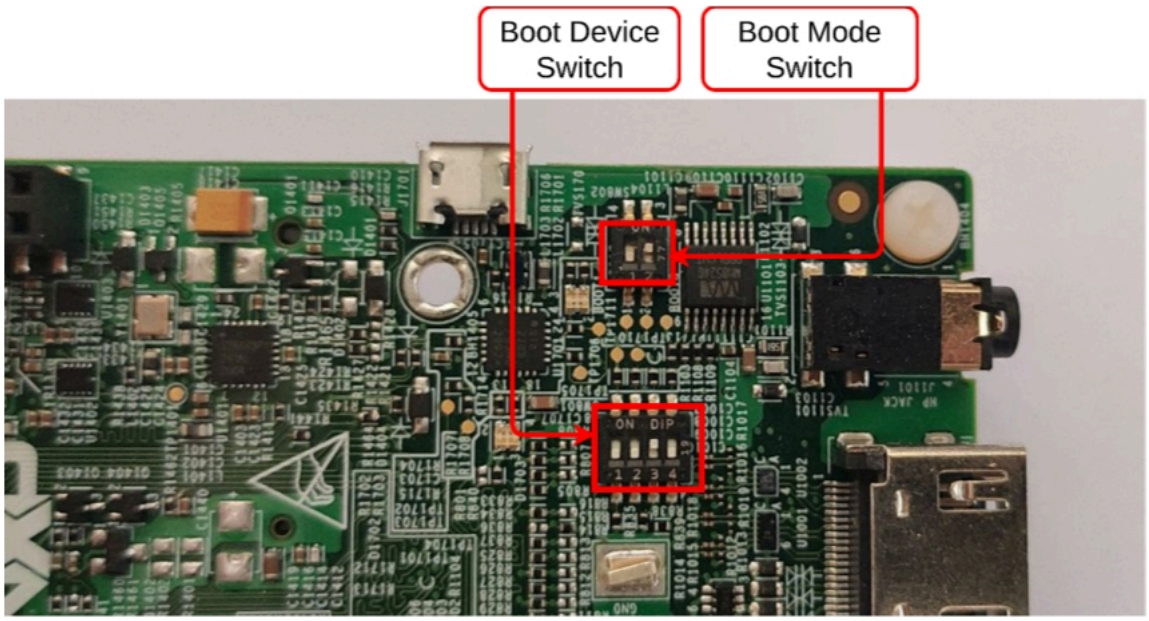


Figure 4. Boot device switch and boot mode switch on i.MX 8M Quad evaluation board

Table 4 shows the settings of the boot mode switch.

Table 4. Boot mode switch settings

D1	D2	Boot mode
0	1	Serial downloader
1	0	Internal boot

Table 5 shows the settings of the boot device switch to boot from eMMC.

Table 5. Switch settings to boot from eMMC

D1	D2	D3	D4
OFF	OFF	ON	OFF

3 NXP-based wireless modules

This document revision addresses the following NXP-based wireless modules.

- **88W8987**: refer to 88W8987 short datasheet and 88W8987 product overview on NXP website.
- **88W8997**: refer to 88W8997 fact sheet and 88W8997 product overview on NXP website.
- **88W9098**: refer to 88W9098 data sheet and 88W9098 product overview on NXP website.
- **IW416**: refer to IW416 data sheet and IW416 product overview on NXP website.

3.1 Interface with i.MX 8M Quad application processor

Figure 5 shows the high-level block diagram of i.MX 8M Quad application processor with the Wi-Fi (SDIO/PCIe) and Bluetooth (UART) hardware interfaces used to communicate with NXP-based wireless module.

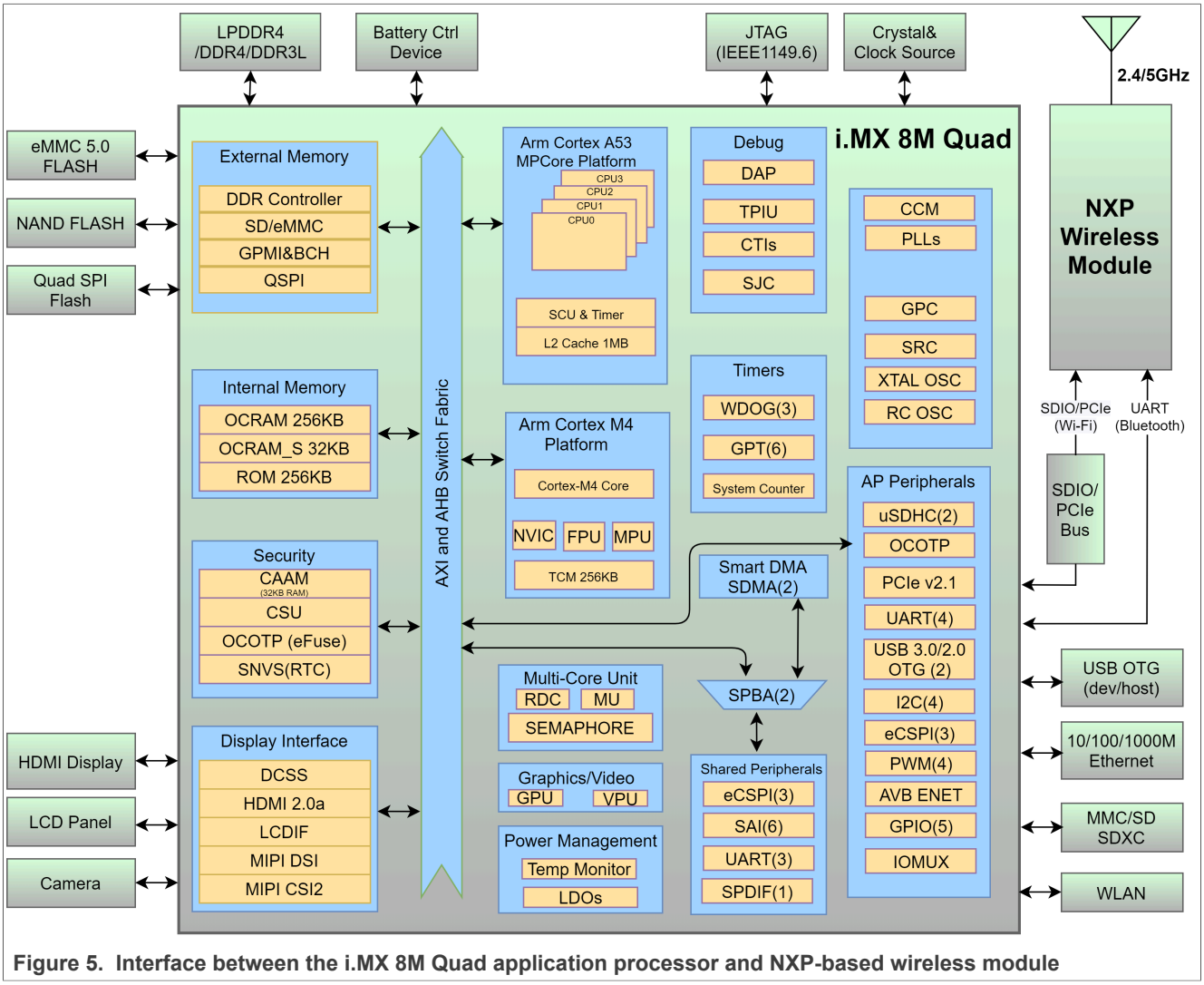


Figure 5. Interface between the i.MX 8M Quad application processor and NXP-based wireless module

3.2 Wi-Fi layer interfaces

The wireless module requires a kernel driver loaded on the i.MX 8M Quad host system and a firmware running on NXP-based wireless modules. When the interface bus driver detects the NXP-based wireless module, the MLAN module downloads the firmware binary via the interface adapter. NXP Wi-Fi driver is loaded between the bus driver and the network stack from the cfg80211 subsystem in the kernel. NXP kernel driver includes a set of controls and configurations to communicate with the user space through one of the following interfaces:

- Input/output control (IOCTL)
- Wireless Extension (Wext)
- CFG80211

The IOCTL provides a path to the user space applications *iwconfig* and *iwpriv* whereas cfg80211 interface provides a different path to the user space applications *wpa_supplicant*, *hostapd* and *iw*.

Figure 6 illustrates the Wi-Fi layer interface.

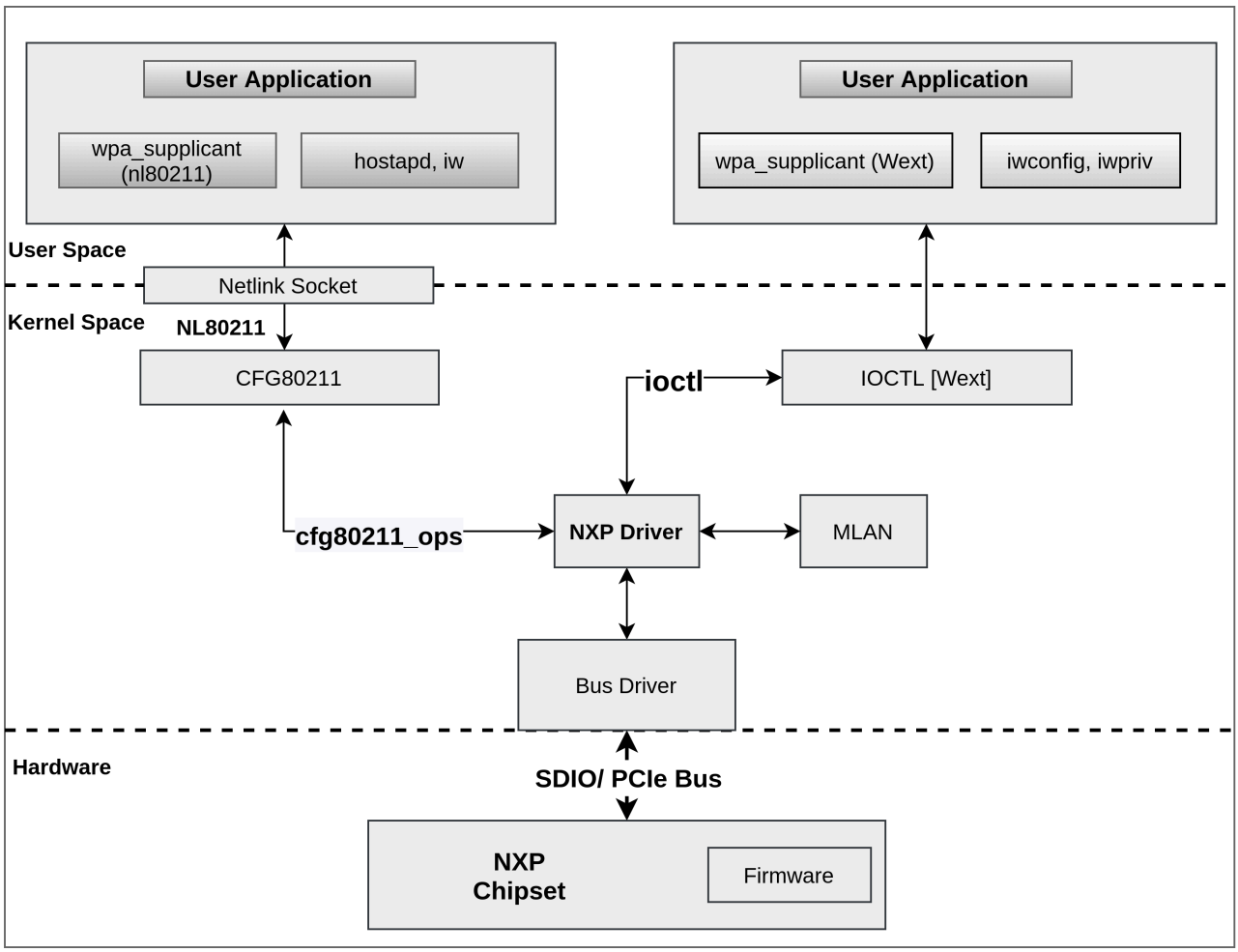


Figure 6. Wi-Fi layer interface

3.3 Bluetooth layer interfaces

Figure 7 illustrates the layers between the user applications and the NXP-based Bluetooth module. The NXP-based wireless module requires a kernel driver loaded on the i.MX 8M Quad host system and a firmware running on NXP SoC. The Wi-Fi driver loads the combo firmware. The *hci_uart* driver provides the HCI interface between the firmware and user application.

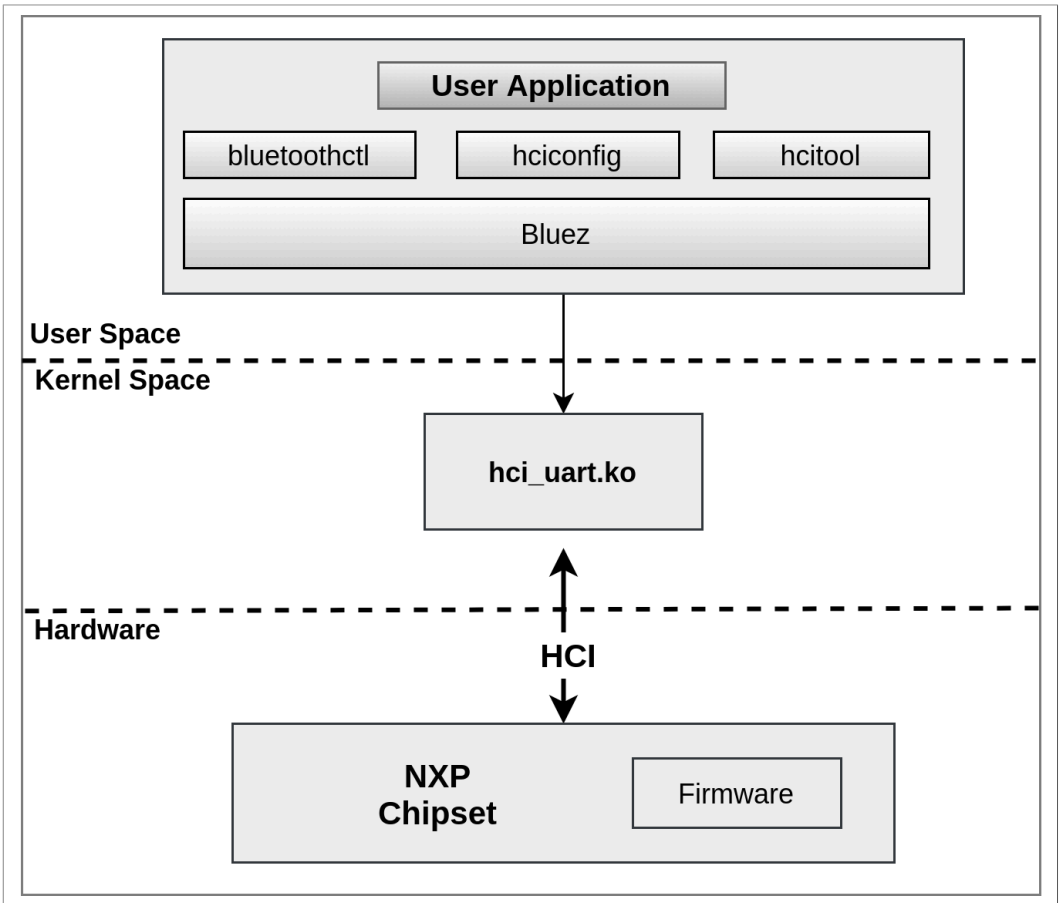


Figure 7. Bluetooth layer interface

3.4 88W8987-based Azurewave AW-CM358-uSD module

Azurewave provides a micro-SD card interface adapter (uSD-1212) with AW-CM358SM module that is compatible with the i.MX 8M Quad EVK. The AW-CM358-uSD supports Wi-Fi through a uSD device interface that conforms to the industry SDIO Full-Speed card specification and allows a host controller using the SDIO bus protocol to access the Wireless SoC device. The AW-CM358-uSD acts as the device on the SDIO bus, and features the following:

- SDIO 3.0 standard
- On-chip memory used for CIS
- Supports 1-bit SDIO and 4-bit SDIO transfer modes
- Special interrupt register for information exchange

3.4.1 Recommended antenna part

MAG.LAYERS: MSA-4008-25GC1-A2

3.4.2 Supported I/O signal level

- SDIO (3.0/2.0) supports 1.8V for I/O signal

3.4.3 Supported RF standard

Table 6. Supported RF standard

I/O voltage level	Wi-Fi	Bluetooth
AW-CM358-uSD	1x1 Wi-Fi 5 (2.4/5 GHz)	5.0

3.4.4 Supported Wi-Fi features

AW-CM358-uSD and AW-CM358MA modules share the same Wi-Fi feature set. Refer to Azurewave AW-CM358SM Datasheet.

Note: *Azurewave AW-CM358-uSD supports both Wi-Fi and Bluetooth RF standards. But as i.MX 8M Quad EVK does not include the FFC connector for Bluetooth interface, only Wi-Fi is supported for i.MX 8M Quad platform with AW-CM358-uSD module combination.*

3.4.5 Azurewave AW-CM358-uSD Wi-Fi module interface

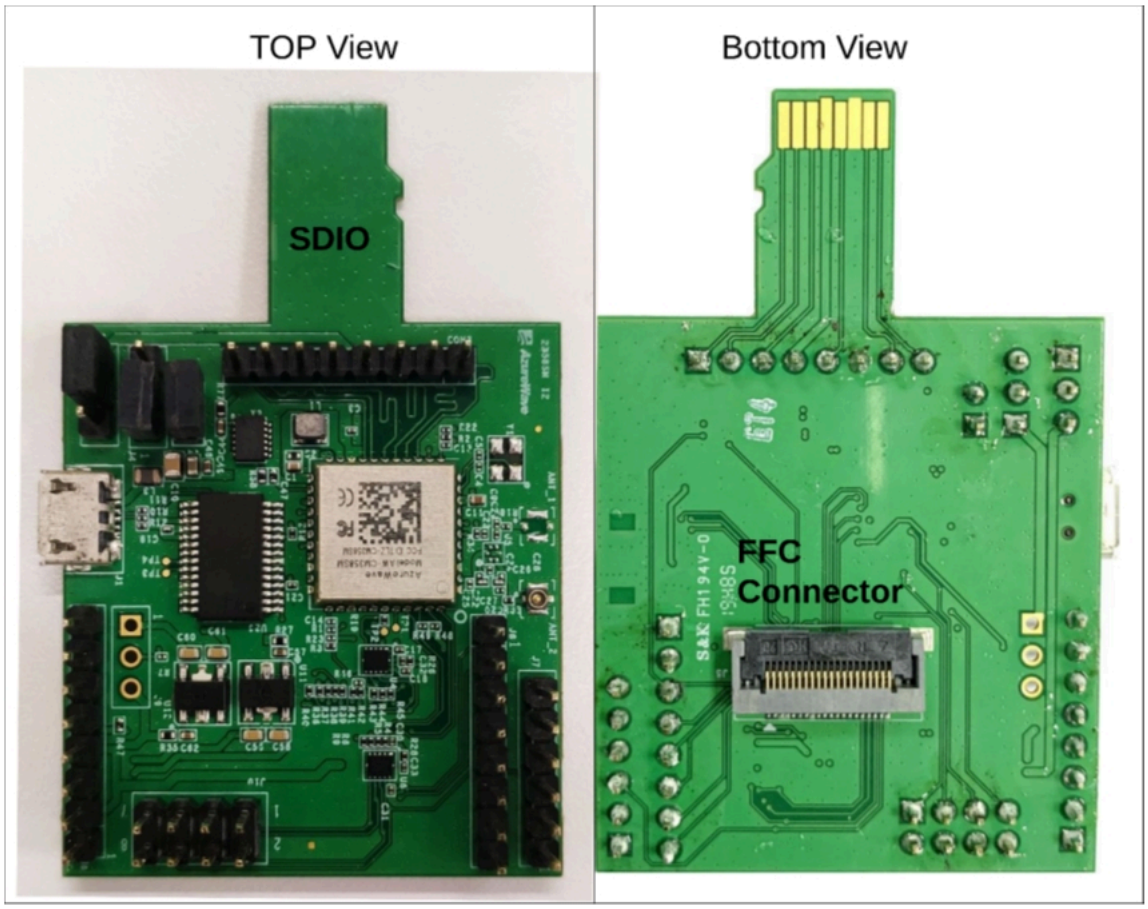


Figure 8. AzureWave AW-CM358-uSD module interface

3.4.6 Azurewave AW-CM358-uSD module jumpers and power supply

This section provides the jumper settings to configure the module with 1.8V SDIO voltage level for Wi-Fi. Connect the jumper J2 between the 1 and 2 header pins to select the SDIO module power source and connect the jumper J4 between the 1 and 2 header pins for 1.8V SDIO voltage as shown in Figure 9.

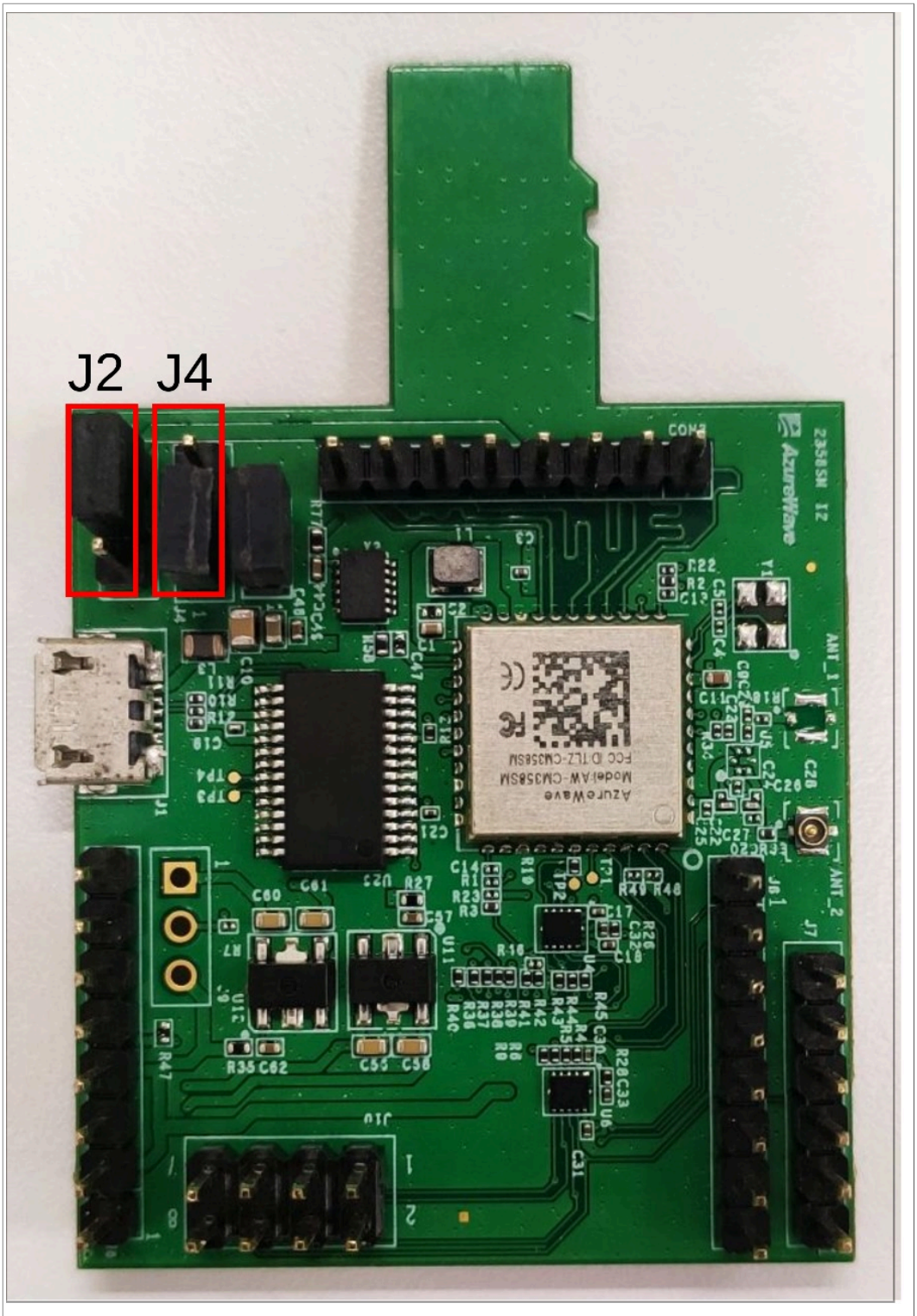


Figure 9. AzureWave AW-CM358-uSD module header positions

3.4.7 Plugging the wireless module

Plug the module into the connector slots of i.MX 8M Quad board:

- For AW-CM358MA, connect the module into the M.2 connector of the i.MX 8M Quad board and screw.
- For AW-CM358-uSD, plug the module into the SDIO card slot of the i.MX 8M Quad board.

```
[ 3632.632050] mmc1: new ultra high speed SDR104 SDIO card at address
 0001
```

3.4.8 AW-CM358-uSD module setup with i.MX 8M Quad

Plug AW-CM358-uSD module into the SDIO card slot of the i.MX 8M Quad board.

```
[ 3632.632050] mmc1: new ultra high speed SDR104 SDIO card at address
 0001
```

Connect the antenna.

Use a Micro USB to USB cable to connect i.MX 8M Quad EVK to the host computer running on Linux OS.

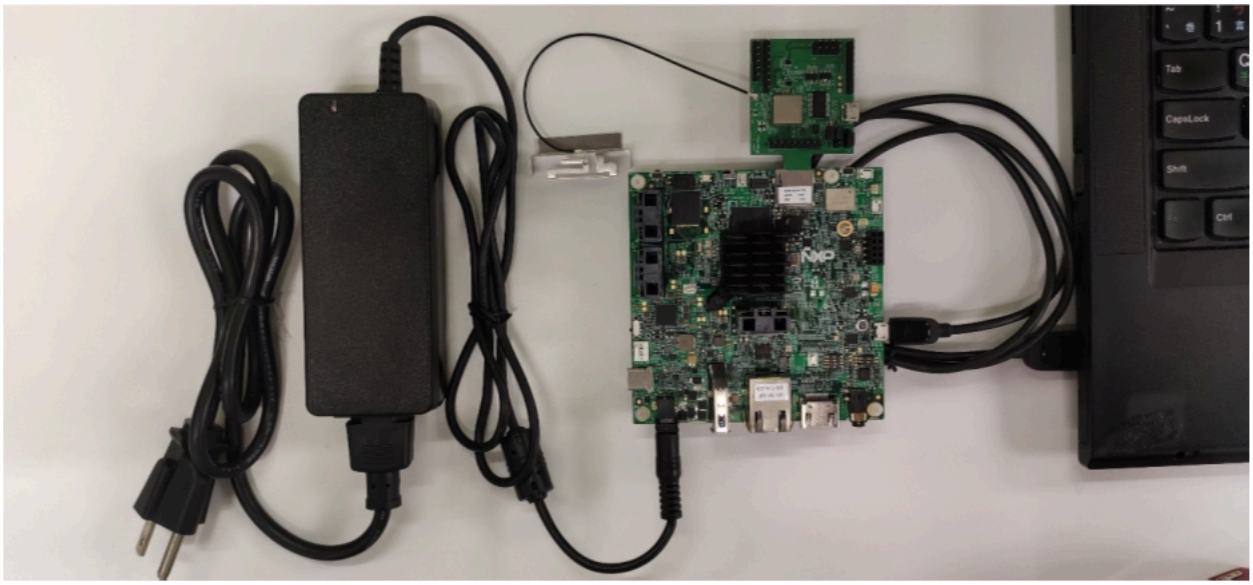


Figure 10. AzureWave AZ-CM358-uSD module and i.MX 8M Quad EVK setup

3.5 88W8987-based AzureWave AW-CM358MA module

The AW-CM358MA module supports a SDIO device interface that conforms to the industry standard SDIO Full-Speed card specification and allows a host controller using the SDIO bus protocol to access the Wireless SoC device.

The AW-CM358MA is a Wi-Fi 5 and Bluetooth 5 combo stamp module with the M.2 adaptor board that features:

- SDIO 3.0 standard
- On-chip memory used for CIS
- 1-bit SDIO and 4-bit SDIO transfer modes
- A special interrupt register for information exchange

3.5.1 Recommended antenna part

MAG.LAYERS: MSA-4008-25GC1-A2

3.5.2 Supported RF standards

Table 7. AW-CM358MA supported RF standards

Part number	Wi-Fi	Bluetooth
AW-CM358MA	1x1 Wi-Fi 5 (2.4/5GHz)	5.0

3.5.3 Supported Wi-Fi features

Refer to Azurewave AW-CM358SM Datasheet.

3.5.4 Supported Bluetooth features

Refer to Azurewave AW-CM358SM Datasheet.

3.5.5 AW-CM358MA module view

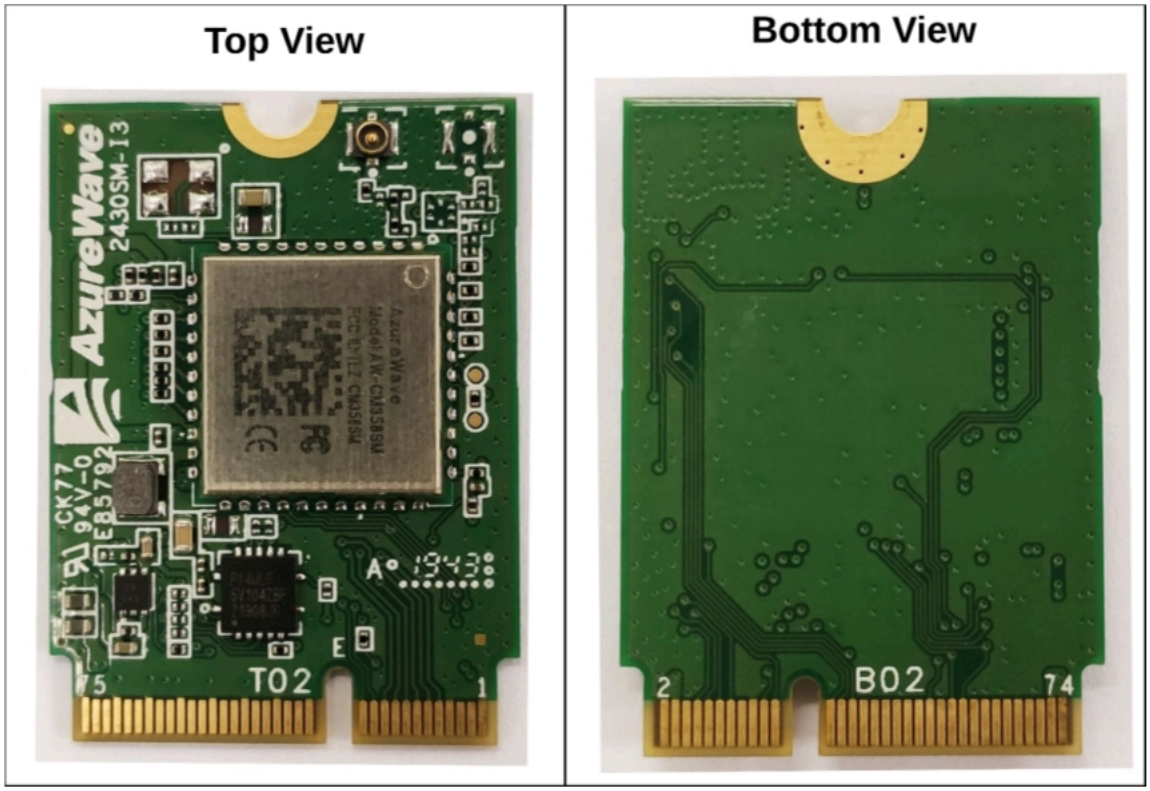


Figure 11. AzureWave AW-CM358MA module

3.5.6 Enabling SDIO on M.2 (AW-CM358MA and AW-CM276MA-SUR)

The command below sets as default the DTB file that enables the SDIO interface on the M.2 connector, and reboots the EVK to load the updated DTB file.

```
root@imx8mqevk:~# mv /run/media/mmcblk0p1/imx8mq-evk-usdhc2-m2.dtb /
run/media/mmcblk0p1/imx8mq-evk.dtb
root@imx8mqevk:~# reboot
```

Note: *This section only applies to AW-CM358MA and AW-CM276MA-SUR modules based on M.2 connector. The SDIO on M.2 DTB is necessary to enable the SDIO interface on the M.2 connector. By default the SDIO support is enabled for the SD Card slot connector. The hardware rework instructions are given in* Section 3.5.7 "i.MX 8M Quad rework for SDIO support on M.2".

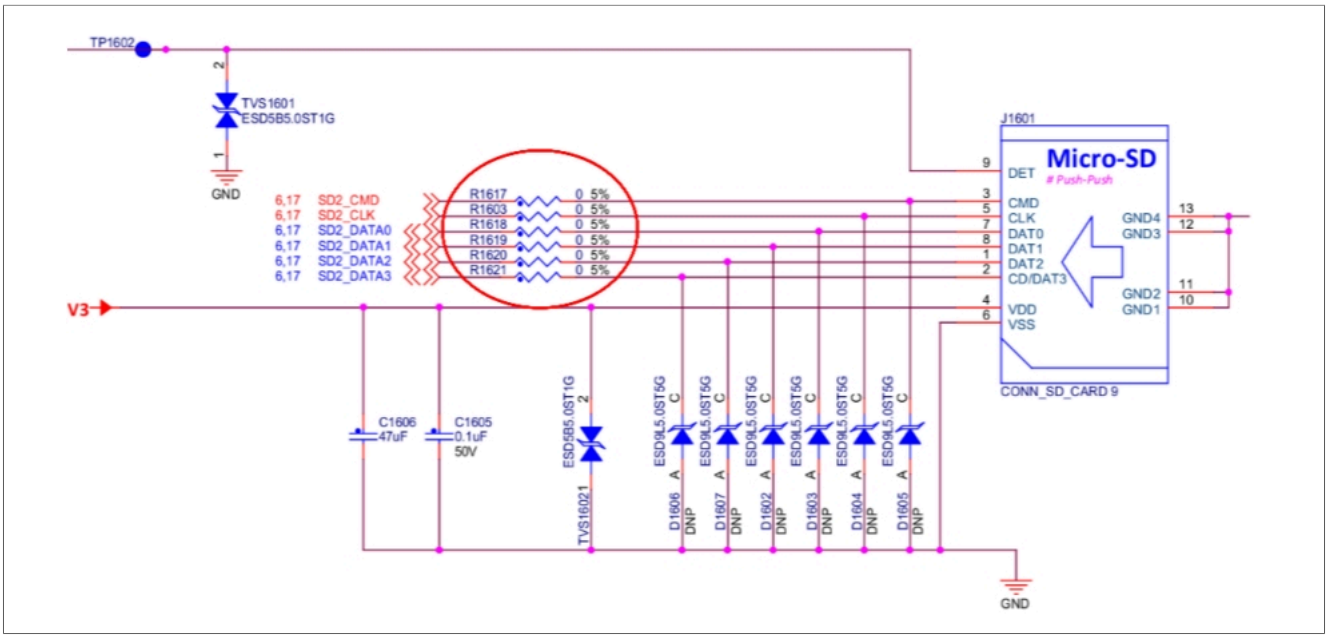
3.5.7 i.MX 8M Quad rework for SDIO support on M.2

The M.2 connector on AW-CM358MA wireless module supports SDIO interface for Wi-Fi whereas the i.MX 8M Quad M.2 connector supports PCIe interface by default. This section shows how to rework i.MX 8M Quad resistors to support SDIO on M.2 connector for the AW-CM358MA module.

Note: *The hardware rework instructions reroute the SDIO interface from the SD card connector to the M.2 connector. A software change is also required to enable the rerouted SDIO interface. Refer to* Section 3.5.6 "Enabling SDIO on M.2 (AW-CM358MA and AW-CM276MA-SUR)" *to enable the SDIO interface on the M.2 connector using a DTB file. Note that the SDIO on the Micro SD card slot will be disabled after the rework.*

- Silkscreen of PCBA SCH-38820 / PCBA SCH-29615

- Remove the following 0Ω 0402 resistors: R1603, R1617, R1618, R1619, R1620 and R1621 (micro SD card J1601)



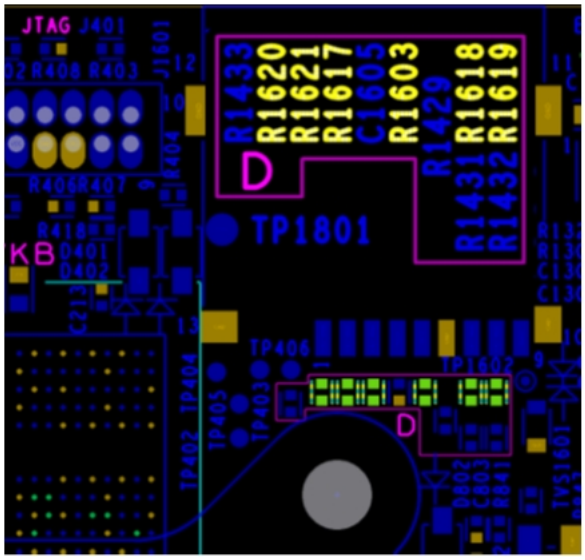


Figure 12. MicroSD Card J1601 resistors

- Install the following 0Ω 0402 resistors: R1429, R1430, R1431, R1432, R1433, R1434, R1435 and R1436 (M.2 J1401)

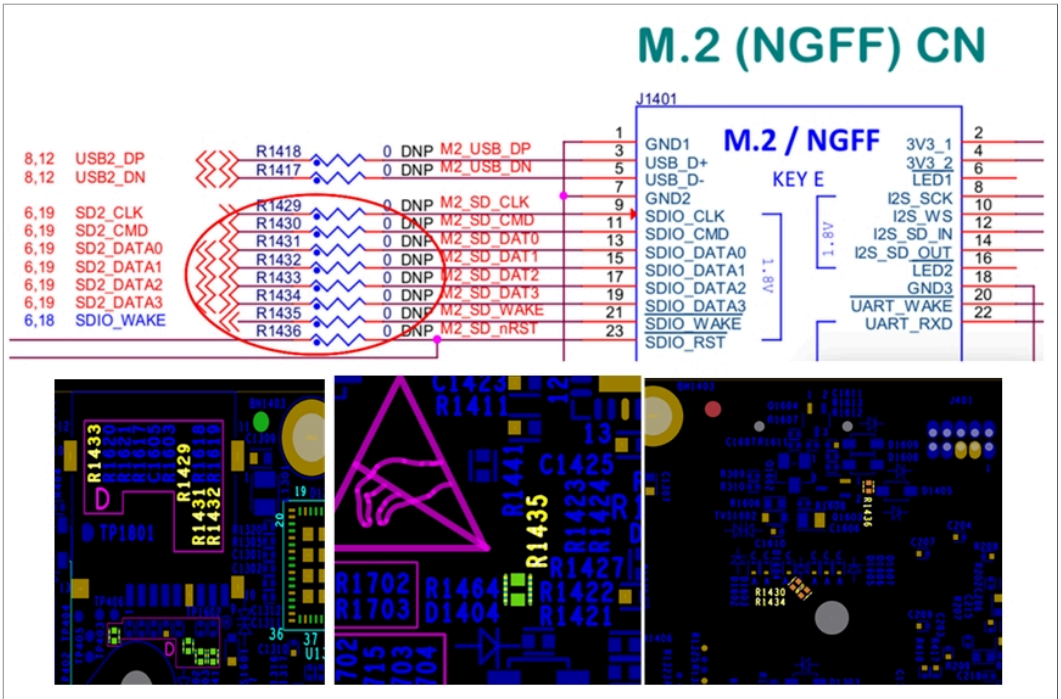


Figure 13. M.2 J1401 resistors

3.5.8 AW-CM358MA module setup with iMX 8M Quad

Connect AW-CM358MA module into the M.2 connector of the i.MX 8M Quad board and screw.

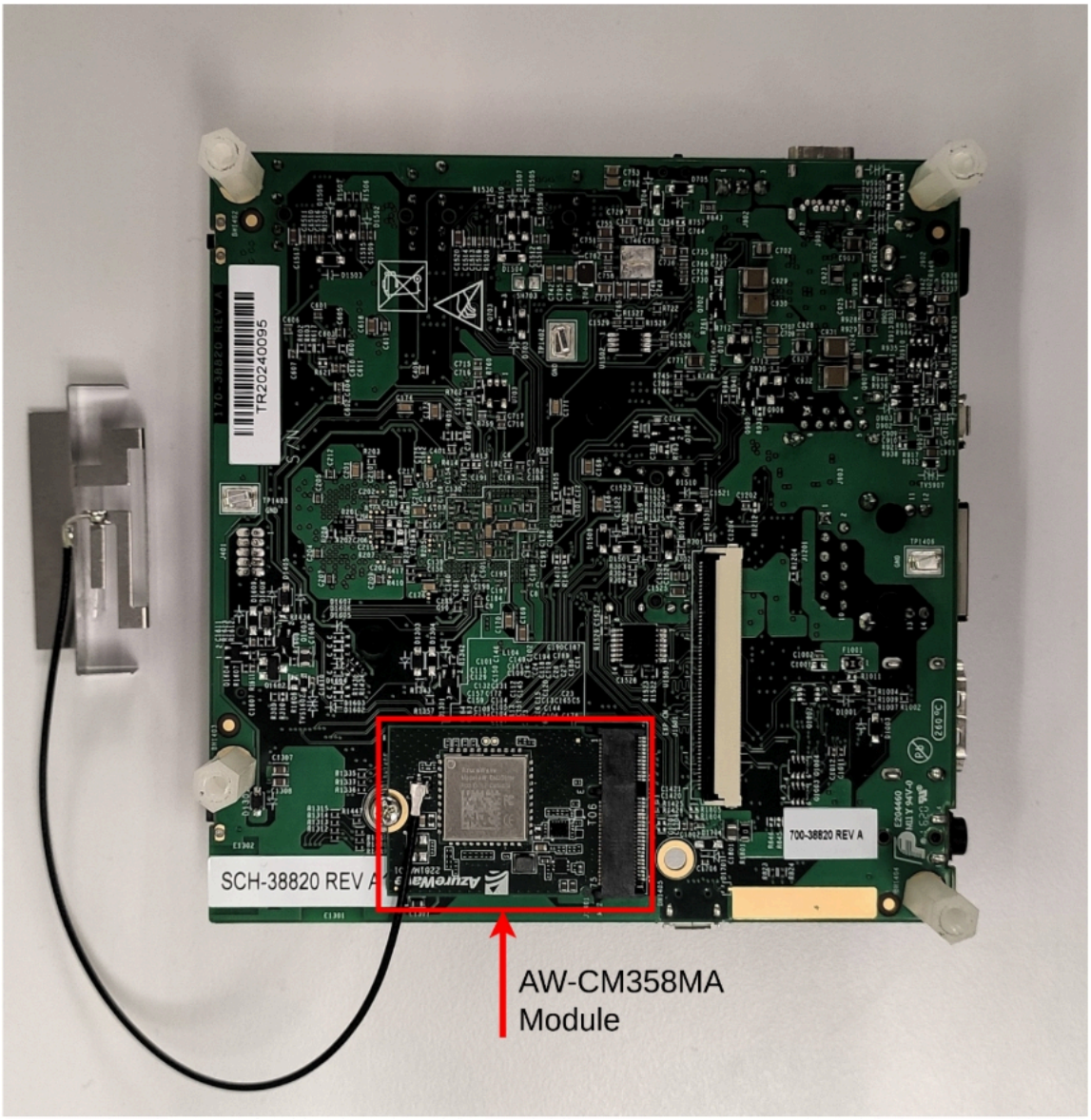


Figure 14. Azurewave AW-CM358MA module plugged into i.MX 8M Quad bottom side M.2 connector

Connect the antenna, and use a Micro USB to USB cable to connect i.MX 8M Quad EVK to the host computer running on Linux OS.

Figure 15. Azurewave AW-CM358MA module and i.MX 8M Quad setup

3.6 88W8997-based AzureWave AW-CM276MA-PUR module

AzureWave AW-CM276MA-PUR supports PCIe and high-speed UART interfaces to the host processor for Wi-Fi and Bluetooth with the M.2 adapter board. Refer to AzureWave AW-CM276MA-PUR Datasheet. AW-CM276MA-PUR module features the following:

- PCIe M.2 TYPE connector
- PCIe interface for Wi-Fi
- UART interface for Bluetooth

Figure 16. AW276MA-PUR module top and bottom views

3.6.1 Recommended antenna part

MAG.LAYERS: MSA-4008-25GC1-A2

3.6.2 Supported RF standards

Table 8. AW-CM276MA-PUR supported RF standards

Part number	Wi-Fi	Bluetooth
AW-CM276MA-PUR	2x2 Wi-Fi 5 (2.4 GHz/5 GHz)	5.0

3.6.3 Supported Wi-Fi features

Refer to AzureWave AW-CM276MA-PUR Datasheet.

3.6.4 Supported Bluetooth features

Refer to AzureWave AW-CM276MA-PUR Datasheet.

3.7 88W8997-based AzureWave module AW-CM276MA-SUR

The AzureWave AW-CM276MA-SUR module based on 88W8997 wireless device supports a SDIO device interface that conforms to the industry standard SDIO Full-Speed card specification and allows a host controller using the SDIO bus protocol to access the wireless SoC device. The AW-CM276MA-SUR is a Wi-Fi 5 and Bluetooth 5 combo stamp module with the M.2 adaptor board that features:

- SDIO 3.0 standard
- On-chip memory used for CIS
- 1-bit SDIO and 4-bit SDIO transfer modes
- Special interrupt register for information exchange
- Allows card to interrupt host

3.7.1 Recommended antenna part

- MAG.LAYERS: MSA-4008-25GC1-A2

3.7.2 Supported RF standard

Table 9. AW-CM276MA-SUR supported RF standards

Part number	Wi-Fi	Bluetooth
AW-CM276MA-SUR	Wi-Fi 5	Bluetooth 5.0

3.7.3 Supported Wi-Fi features

Refer to AW-CM276MA-SUR data sheet.

3.7.4 Supported Bluetooth features

Refer to AW-CM276MA-SUR data sheet.

3.7.5 AzureWave AW-CM276MA-SUR Wi-Fi module interface

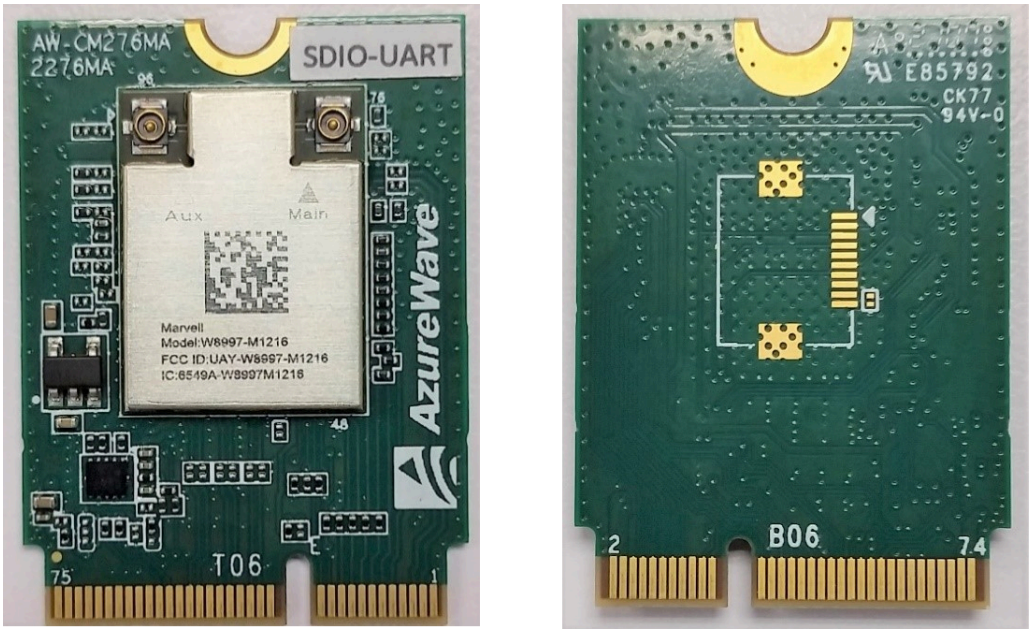


Figure 17. AzureWave AW-CM276MA-SUR Wi-Fi module interface

3.8 88W8997-based AzureWave module AW-CM276-uSD

Azurewave provides a micro-SD card interface adapter (uSD-1216) with AW-CM276NF module that is compatible with the i.MX 8M Quad EVK. The AW-CM276-uSD supports Wi-Fi through uSD device interface that conforms to the industry standard SDIO Full-Speed card specification and allows a host controller using the SDIO bus protocol to access the Wireless SoC device. The AW-CM276NF acts as the device on the SDIO bus, and features the following:

- SDIO 3.0 standard
- On-chip memory used for CIS
- Supports 1-bit SDIO and 4-bit SDIO transfer modes
- Special interrupt register for information exchange
- Allows card to interrupt host

3.8.1 Recommended antenna part

- MAG.LAYERS: MSA-4008-25GC1-A2

3.8.2 Supported I/O signal level

SDIO (3.0/2.0) supports 1.8V/3.3V for I/O signal.

3.8.3 Supported RF standard

Table 10. AW-CM276-uSD supported RF standards

Part number	Wi-Fi	Bluetooth
AW-CM276NF	2x2 Wi-Fi 5	Bluetooth 5.0

3.8.4 Supported Wi-Fi features

Refer to AzureWave AW-CM276NF data sheet.

Note: *AzureWave AW-CM276NF supports both Wi-Fi and Bluetooth RF standards. But as i.MX 8M Quad EVK does not include the FFC connector for Bluetooth interface, only Wi-Fi is supported for i.MX 8M Quad platform with AW-CM276-uSD module combination.*

3.8.5 AzureWave AW-CM276-uSD Wi-Fi module interface

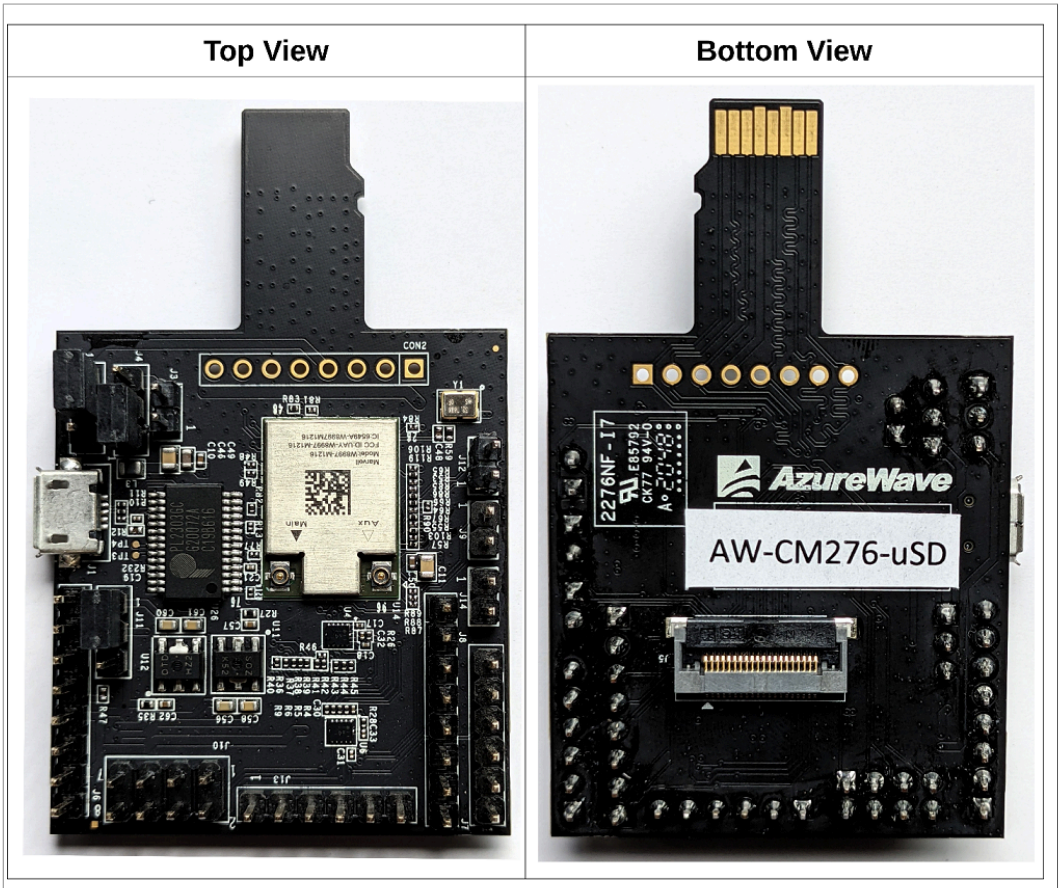


Figure 18. AzureWave AW-CM276-uSD Wi-Fi module interface

3.8.6 AzureWave AW-CM276-uSD module jumpers and power supply

This section provides the jumper settings to configure the module with 1.8 V SDIO voltage level for Wi-Fi.

Connect the jumper J2 between the 1 and 2 header pins to select the SDIO module power source, connect the jumper J4 between the 1 and 2 header pins to select 1.8 V VIO voltage level, and connect the jumper J11 between the 1 and 2 header pins for 1.8V VIO_SD voltage level as shown in Figure 19.

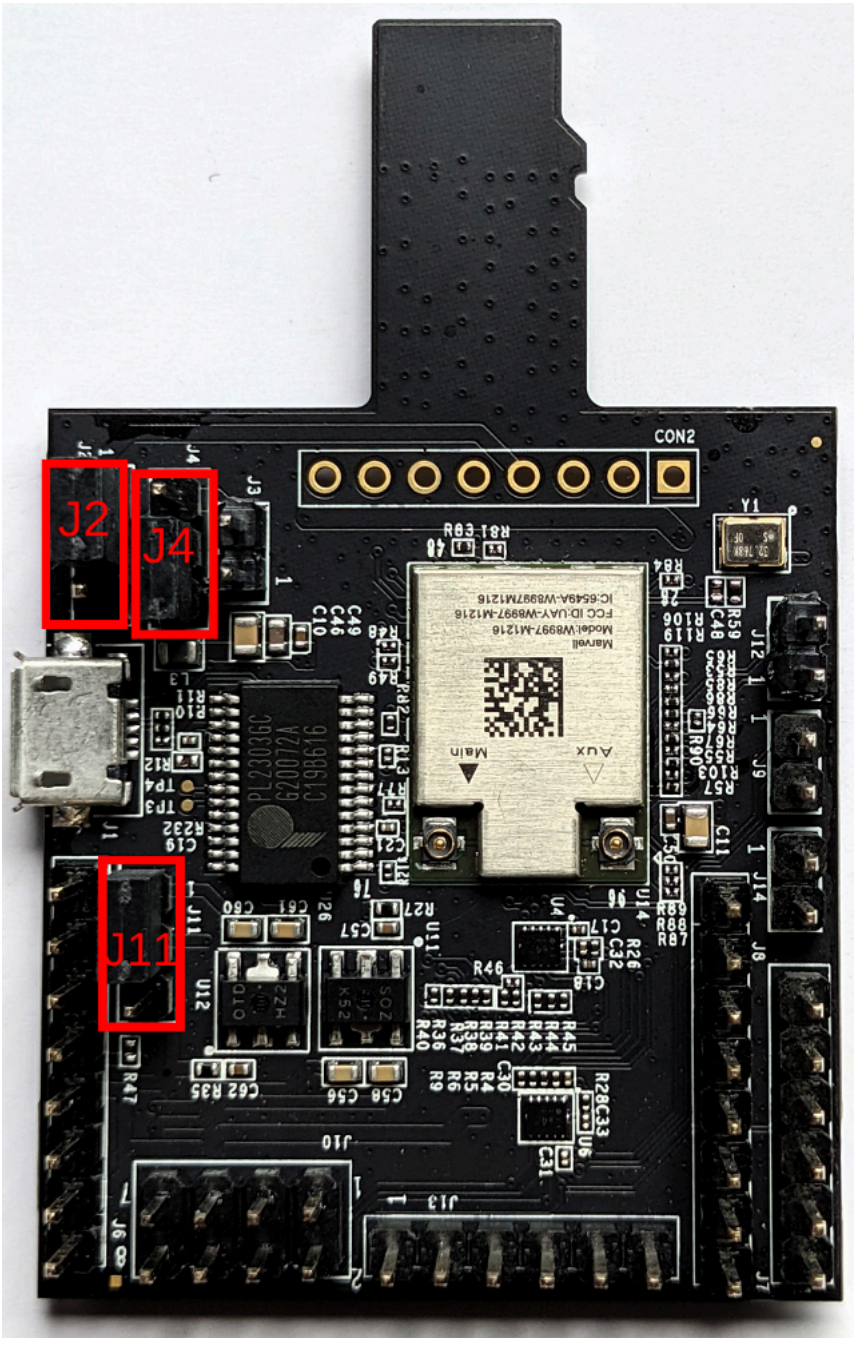


Figure 19. AzureWave AW-CM276-uSD module jumpers and power supply

3.9 88W9098-based Murata module LBEE6ZZ1

LBEE6ZZ1 is a wireless module from Murata based on NXP's 88W9098 wireless device. Murata provides an M.2 interface with LBEE6ZZ1 module that is compatible with the i.MX 8M Quad EVK. The LBEE6ZZ1 supports Wi-Fi through a PCIe device interface that conforms to the industry PCIe 2.0 specifications and allows a host controller using the PCIe bus protocol to access the wireless SoC device. The LBEE6ZZ1 acts as the device on the PCIe bus, and features the following:

- PCIe 2.0 interface for Wi-Fi
- UART interface for Bluetooth
- IEEE 802.11 a/b/g/n/ac/ax 2x2 MU-MIMO

3.9.1 Recommended antenna

- Dual-band antenna with u.FL connector

3.9.2 Supported RF standards

Table 11. LBEE6ZZ1 supported RF standards

Part number	Wi-Fi	Bluetooth
LBEE6ZZ1	Wi-Fi 6 2x2	5.1

3.9.3 Supported Wi-Fi features

Refer to 88W9098 data sheet.

3.9.4 Supported Bluetooth features

Refer to 88W9098 data sheet.

3.9.5 Murata LBEE6ZZ1 Wi-Fi module interface

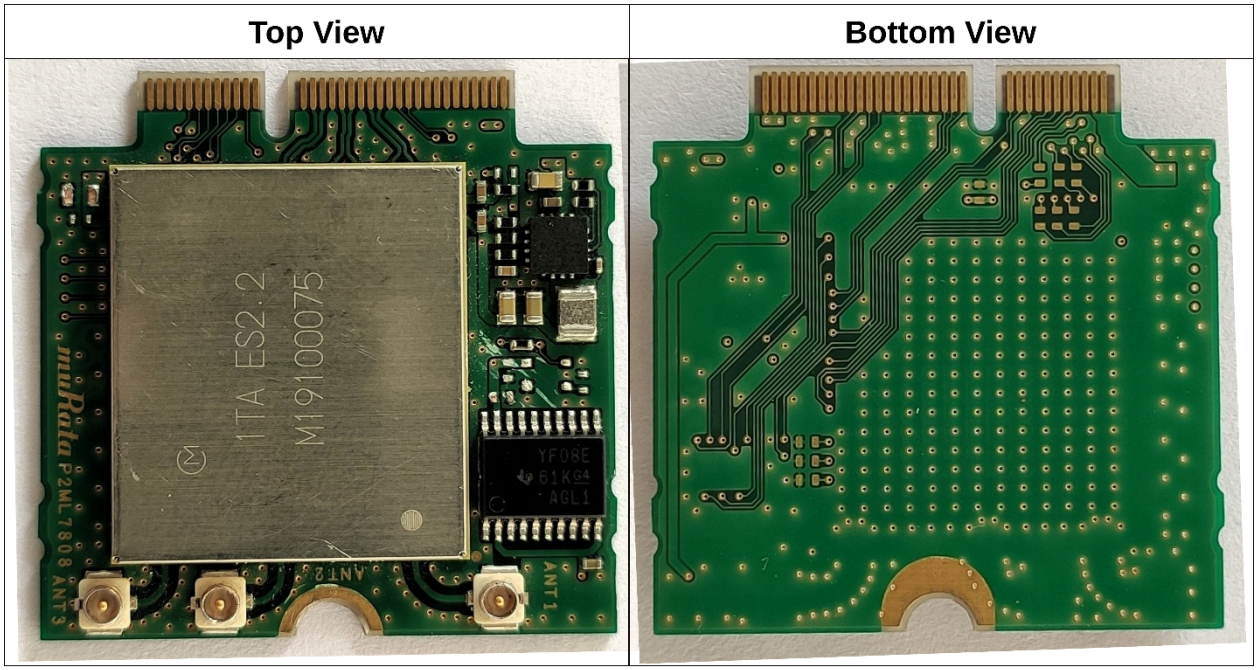


Figure 20. Murata LBEE6ZZ1 Wi-Fi module interface

3.10 IW416-based AzureWave AW-AM510MA module

The AW-AM510MA module supports a SDIO device interface that conforms to the industry standard SDIO Full-Speed card specification and allows a host controller using the SDIO bus protocol to access the Wireless SoC device. The AW-AM510MA is a Wi-Fi 4 and Bluetooth 5.1 combo stamp module with the M.2 adaptor board that features:

- SDIO 3.0 standard
- On-chip memory used for CIS
- 1-bit SDIO and 4-bit SDIO transfer modes
- One special interrupt register for information exchange

3.10.1 Recommended antenna part

MAG.LAYERS: MSA-4008-25GC1-A2

3.10.2 Supported RF standards

Table 12. AW-AM510MA supported RF standards

Part number	Wi-Fi	Bluetooth
AW-AM510MA	1x1 Wi-Fi 4 (2.4 GHz/5 GHz)	5.1

3.10.3 Supported Wi-Fi features

Refer to AzureWave AW-AM510MA data sheet.

3.10.4 Supported Bluetooth features

Refer to AzureWave AW-AM510MA data sheet.

3.10.5 AW-AM510MA module view

aw-am510ma-module-view.png

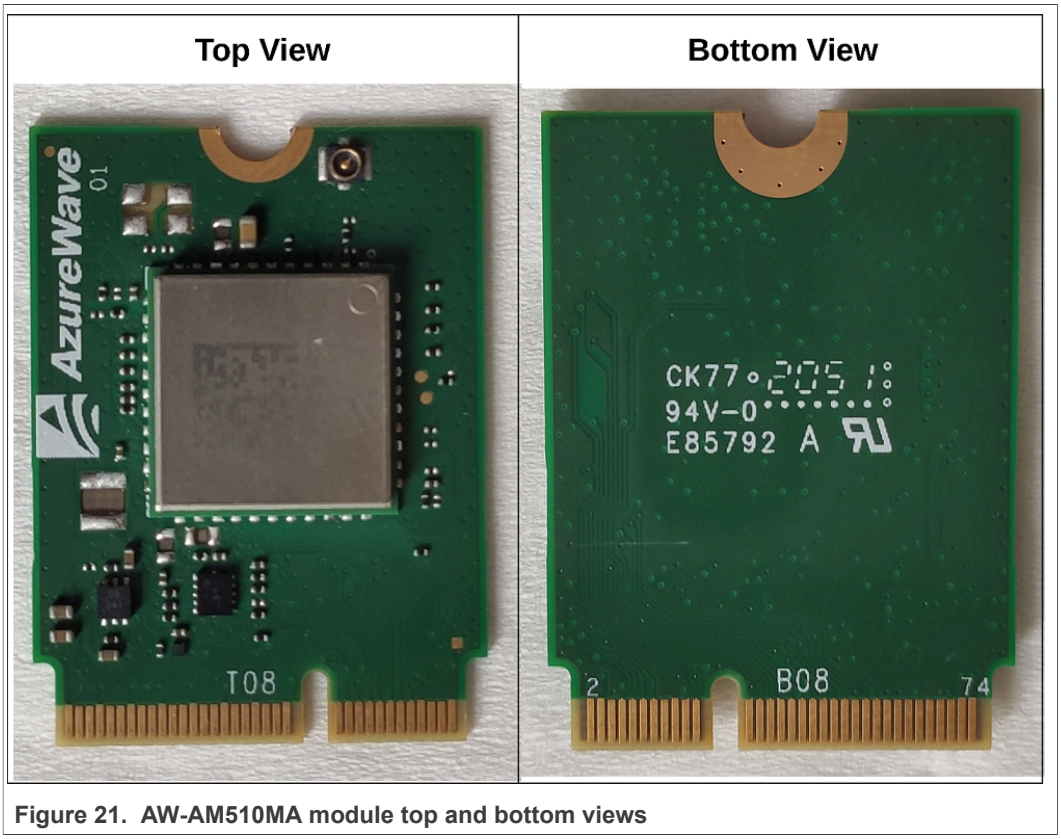


Figure 21. AW-AM510MA module top and bottom views

Note: *Refer to Section 3.5.7 for rework instructions to support SDIO interface on M.2.*

4 i.MX 8M EVK Linux image setup

The i.MX 8M Quad Linux BSP is a collection of binary files, source code, and support files used to create a U-Boot bootloader, a Linux kernel image, and a root file system for i.MX 8M Quad development platforms.

All the information on how to set up the Linux OS host, how to run and configure a Yocto Project, generate an image, and generate a rootfs, are covered in the i.MX Yocto Project User's Guide. On i.MX 8M Quad, four elements are needed:

- imx-boot (built by imx-mkimage), which includes SPL, U-Boot, Arm Trusted Firmware, DDR firmware, and HDMI firmware
- Linux kernel image
- A device tree file (*.dtb*) for the board being used
- A root file system (*rootfs*) for the specific Linux image

Read more in section 4.1 of *i.MX Linux® User's Guide*, document number “IMXLUG”.

4.1 Using the pre-built image

This section describes the steps to prepare eMMC to boot up an i.MX 8M Quad EVK using a Linux host machine. The pre-built image can be downloaded from the page Embedded Linux for I.MX Application Processors on NXP website. Accept NXP software license agreement when prompted.

The release contains a pre-built image that is built specifically for the board configuration. It does not run on other boards unless U-Boot, the device tree, and rootfs are changed.

Note: *A pre-built image contains all the image elements and does not require to build the image using the Yocto setup.*

The content of the extracted downloaded release archive is shown hereafter.

```
├── EULA.txt
├── fsl-image-mfgtool-initramfs-imx_mfgtools.cpio.gz.u-boot
├── GPLv2
├── Image-imx8_all.bin
├── Image-imx8mqevk.bin
├── imx8mq-evk-ak4497.dtb
├── imx8mq-evk-audio-tdm.dtb
├── imx8mq-evk-dcss-adv7535.dtb
├── imx8mq-evk-dcss-rm67191.dtb
├── imx8mq-evk-dp.dtb
├── imx8mq-evk.dtb
├── imx8mq-evk-dual-display.dtb
├── imx8mq-evk-inmate.dtb
├── imx8mq-evk-lcdif-adv7535.dtb
├── imx8mq-evk-lcdif-rm67191.dtb
├── imx8mq-evk-pcie1-m2.dtb
├── imx8mq-evk-pcie-ep.dtb
├── imx8mq-evk-pdm.dtb
├── imx8mq-evk-root.dtb
├── imx8mq-evk-rpmsg.dtb
├── imx8mq-evk-usdhc2-m2.dtb
├── imx8mq-evk-usd-wifi.dtb
├── imx-boot-imx8mqevk-sd.bin-flash_dp_evk
├── imx-boot-imx8mqevk-sd.bin-flash_evk
├── imx-boot-imx8mqevk-sd.bin-flash_evk_no_hdmi
├── imx-image-full-imx8mqevk.manifest
├── imx-image-full-imx8mqevk.tar.bz2
├── imx-image-full-imx8mqevk.wic
├── imx-image-multimedia-imx8mqevk.manifest
├── imx-image-multimedia-imx8mqevk.tar.bz2
├── imx-image-multimedia-imx8mqevk.wic
├── imx_mcore_demos
├── QUALCOMM_ATHEROS_LICENSE_AGREEMENT.pdf
├── README.uuu
├── samples
├── SCR-5.4.47_2.2.0.txt
└── uuu.auto
```

4.2 Building the image from source

Refer to sections 3, 4 and 5 in i.MX Yocto Project User's Guide.

4.3 Flashing the image to eMMC

The *Universal Update Utility* (*UUU*) runs on a Windows or Linux OS host and is used to download images to different devices on an i.MX board.

Download *UUU*

Download UUU version 1.2.135 or later from https://github.com/NXPmicro/mfgtools/releases.

Copy *UUU*

Copy the uuu binary into the */usr/bin/* directory and change the permission to executable.

```
ubuntu@ubuntu-desktop:/# sudo chmod +x /usr/bin/uuu
```

Flash the image

Use the following format for the command:

```
ubuntu@ubuntu-desktop:/# sudo uuu -b emmc_all <bootloader> <image>
```

For example:

```
ubuntu@ubuntu-desktop:/# sudo uuu -b emmc_all imx-boot-
imx8mqevk-sd.bin-flash_evk imx-image-full-imx8mqevk.wic
```

Follow these instructions to use *UUU* for i.MX 8M Quad:

- Connect a USB cable from a computer to the USB OTG/TYPE C port on the board
- Connect a USB cable from the OTG-to-UART port to the computer for console output
- Open a Terminal emulator program (for example minicom)
- Set the boot pin to serial download mode (refer to Section 2.4 "i.MX 8M Quad switch settings")

UUU usage

```
ubuntu@ubuntu-desktop:/# sudo uuu -b emmc_all <bootloader> <image>
```

Using the pre-built image (Section 4.1 "Using the pre-built image"), the following command burns the complete image into eMMC:

```
ubuntu@ubuntu-desktop:/# sudo uuu <release package>.zip
```

4.4 Booting from eMMC

To boot the i.MX 8M Quad from eMMC, set the boot switch per the settings given in Section 2.4 "i.MX 8M Quad switch settings".

4.4.1 Serial console setup

Follow these steps to setup the serial console and access the i.MX 8M Quad device terminal:

- Open the serial console and login into the device

```
ubuntu@ubuntu-desktop:/# sudo minicom -s -D /dev/ttyUSBx
```

- *ttyUSB0* or *ttyUSB01* are the serial devices. The minicom setup configuration is given below:

```
A -    Serial Device      : /dev/ttyUSBx
E -    Bps/Par/Bits       : 115200 8N1
F - Hardware Flow Control : No
G - Software Flow Control : No
```

- Save and exit.

4.4.2 Enabling SDIO on M.2 connector

The procedure below sets as default the DTB file that enables the SDIO interface on the M.2 connector.

- **Step 1** - Power-on the board and hit any key within 3 seconds to halt u-boot from booting and enter the u-boot console
- **Step 2** - Set `fdtfile` with the DTB file to enable SDIO interface M.2 connector on bottom of the board:

```
u-boot=> setenv fdtfile imx8mq-evk-usdhc2-m2.dtb
u-boot=> saveenv
```

Command output example:

```
Saving Environment to MMC... Writing to MMC(0)... OK
```

- **Step 3** - Reset the board

```
u-boot=> reset
```

Command output example:

```
resetting ...
```

Note: *This section only applies to modules based on M.2 connector with SDIO interface for Wi-Fi. The SDIO on M.2 DTB is necessary to enable the SDIO interface on the M.2 connector. By default the SDIO support is enabled for the SD Card slot connector. The hardware rework instructions are given in* Section 3.5.7 "i.MX 8M Quad rework for SDIO support on M.2".

4.4.3 Enabling PCIe on M.2 (88W9098-based Murata module LBEE6ZZ1)

The procedure below sets as default the DTB file that enables the PCIe interface on the M.2 connector.

- **Step 1** - Power-on the board and hit any key within 3 seconds to halt u-boot from booting and enter the u-boot console
- **Step 2** - Set `fdtfile` with the DTB file to enable PCIe interface M.2 connector on bottom of the board:

```
u-boot=> setenv fdtfile imx8mq-evk-pcie1-m2.dtb
      u-boot=> saveenv
```

Command output example:

```
Saving Environment to MMC... Writing to MMC(0)... OK
```

- **Step 3** - Reset the board

```
u-boot=> reset
```

Command output example:

```
resetting ...
```

4.4.4 Linux OS login

The default login username for the i.MX Linux OS is *root*. There is no password.

4.4.5 Software package release version information

The software package release version information is provided in the release notes.

5 Bring-up of Wi-Fi interfaces

This section describes the bring-up steps for the Wi-Fi interfaces of the NXP-based wireless module connected to i.MX 8M Quad EVK.

5.1 Bring-up of 88W8987-based wireless module (AW-CM358MA)

Follow these instructions to load the driver modules and bring up the 88W8987-based wireless module

- Use the nano editor included in the pre-built image to edit and verify the module parameters in *wifi_mod_para.conf* configuration file

```
root@imx8mqevk:~# nano /lib/firmware/nxp/wifi_mod_para.conf
```

Content of the configuration file

```
SD8987 = {
      cfg80211_wext=0xf
      wfd_name=p2p
      max_vir_bss=1
      cal_data_cfg=none
      drv_mode=7
      ps_mode=2
      auto_ds=2
      fw_name=nxp/sdiouart8987_combo_v0.bin
}
```

- Load the modules in the kernel

```
root@imx8mqevk:~# modprobe moal mod_para=nxp/wifi_mod_para.conf
```

Verify the kernel debug messages in the command output

```
[ 2896.073550] wlan: Loading MWLAN driver
[ 2896.178399] vendor=0x02DF device=0x9149 class=0 function=1
[ 2896.183972] Attach moal handle ops, card interface type:0x105
[ 2896.189935] SD8987: init module param from usr cfg
[ 2896.194807] card_type: SD8987, config block: 0
[ 2896.199291] cfg80211_wext=0xf
[ 2896.202274] wfd_name=p2p
[ 2896.204855] max_vir_bss=1
[ 2896.207509] cal_data_cfg=none
[ 2896.210494] drv_mode = 7
[ 2896.213054] ps_mode = 2
[ 2896.215532] auto_ds = 2
[ 2896.218007] fw_name=nxp/sdiouart8987_combo_v0.bin
[ 2896.225747] SDIO: max_segs=128 max_seg_size=65535
[ 2896.230505] rx_work=1 cpu_num=4
[ 2896.233724] Attach mlan adapter operations.card_type is 0x105.
[ 2896.240142] wlan: Enable TX SG mode
[ 2896.243666] wlan: Enable RX SG mode
[ 2896.251307] Request firmware: nxp/sdiouart8987_combo_v0.bin
[ 2896.667066] Wlan: FW download over, firmwarelen=526996 downloaded
 526996
[ 2897.601246] WLAN FW is active
[ 2897.604233] on_time is 2897601629500
[ 2897.638511] fw_cap_info=0x181c3f03, dev_cap_mask=0xffffffff
[ 2897.644138] max_p2p_conn = 8, max_sta_conn = 8
[ 2897.671116] wlan: version = SD8987---16.92.10.p207-MXM4X16186.p6-
GPL-(FP92)
[ 2897.682184] wlan: Driver loaded successfully
```

- Verify the Wi-Fi interfaces

```
root@imx8mqevk:~# ifconfig -a
```

Command output example:

```
eth0  Link encap:Ethernet  HWaddr 00:04:9f:06:77:40
      UP BROADCAST MULTICAST DYNAMIC  MTU:1500  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
lo    Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      inet6 addr: ::1/128 Scope:Host
      UP LOOPBACK RUNNING  MTU:65536  Metric:1
      RX packets:90 errors:0 dropped:0 overruns:0 frame:0
      TX packets:90 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:6708 (6.5 KiB)  TX bytes:6708 (6.5 KiB)
mlan0 Link encap:Ethernet  HWaddr 00:50:43:24:83:c4
      BROADCAST MULTICAST  MTU:1500  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
p2p0  Link encap:Ethernet  HWaddr 02:50:43:24:83:c4
      BROADCAST MULTICAST  MTU:1500  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
uap0  Link encap:Ethernet  HWaddr 00:50:43:24:84:c4
      BROADCAST MULTICAST  MTU:1500  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

- Get the Wi-Fi driver version

```
root@imx8mqevk:~# cat /proc/mwlan/adapter0/uap0/info
```

Command output example

```
driver_name = "uap"
driver_version = SD8987---w8987o-V0, MXXX, FP92, 16.92.10.p206-
MXM5X16186.p6-GPL-(FP92)
interface_name="uap0"firmware_major_version=16.92.10
media_state="Disconnected"mac_address="70:66:55:26:8b:6b"
```

5.2 Bring-up of 88W8997-based wireless module (AW-CM276MA-PUR)

Follow these instructions to load the driver modules and bring up the 88W8987-based wireless module

- Use the nano editor included in the pre-built image to edit and verify the module parameters in *wifi_mod_para.conf* configuration file

```
root@imx8mqevk:~# nano /lib/firmware/nxp/wifi_mod_para.conf
```

Content of the configuration file

```
PCIE8997 = {
      cfg80211_wext=0xf
      wfd_name=p2p
      cal_data_cfg=none
      drv_mode=7
      fw_name=nxp/pcieuart8997_combo_v4.bin
}
```

- Load the modules in the kernel

```
root@imx8mqevk:~# modprobe moal mod_para=nxp/wifi_mod_para.conf
```

Verify the kernel debug messages in the command output

```
[  139.999861] wlan: Loading MWLAN driver
[  140.004586] wlan_pcie 0000:01:00.0: enabling device (0000 -> 0002)
[  140.010904] Attach moal handle ops, card interface type: 0x204
[  140.016763] No module param cfg file specified
[  140.021239] rx_work=1 cpu_num=4
[  140.024418] Attach mlan adapter operations.card_type is 0x204.
[  140.034659] Request firmware: mrvl/pcieuart8997_combo_v4.bin
[  140.993340] FW download over, size 627620 bytes
[  141.859194] WLAN FW is active
[  141.862165] on_time is 141859539976
[  142.039695] fw_cap_info=0x18fcffa3, dev_cap_mask=0xffffffff
[  142.045296] max_p2p_conn = 8, max_sta_conn = 8
[  142.070295] wlan: version = PCIE8997-16.68.10.p16-MXM5X16215-GPL-
(FP92)
[  142.077289] wlan: Driver loaded successfully
```

- Verify the Wi-Fi interfaces

```
root@imx8mqevk:~# ifconfig -a
```

Command output example:

```
eth0 Link encap:Ethernet  HWaddr 00:04:9f:06:c5:a5
      inet addr:169.254.130.90  Bcast:169.254.255.255
 Mask:255.255.0.0
      inet6 addr: fe80::204:9fff:fe06:c5a5/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:89 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 B)  TX bytes:30164 (29.4 KiB)

lo   Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      inet6 addr: ::1/128 Scope:Host
      UP LOOPBACK RUNNING  MTU:65536  Metric:1
      RX packets:458 errors:0 dropped:0 overruns:0 frame:0
      TX packets:458 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:29156 (28.4 KiB)  TX bytes:29156 (28.4 KiB)

mlan0 Link encap:Ethernet  HWaddr 70:66:55:9b:3a:95
      BROADCAST MULTICAST  MTU:1500  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

p2p0 Link encap:Ethernet  HWaddr 72:66:55:9b:3a:95
      BROADCAST MULTICAST  MTU:1500  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

uap0  Link encap:Ethernet  HWaddr 70:66:55:9b:3b:95
      BROADCAST MULTICAST  MTU:1500  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

- Get the Wi-Fi driver version

```
root@imx8mqevk:~# cat /proc/mwlan/adapter0/uap0/info
```

Command output example

```
driver_name = "uap"
driver_version = PCIE8997-w8997o-V4, RF878X, FP68_LINUX, 16.68.10.p16-
MXM5X16215-GPL-(FP92)
interface_name="uap0"
firmware_major_version=16.68.10
media_state="Disconnected"
mac_address="70:66:55:9b:3b:95"
```

5.3 Bring-up of 88W8997-based AzureWave module (AW-CM276MA-SUR and AW-CM276-uSD)

Follow these instructions to load the driver modules and bring up the 88W8997-based wireless module.

- Use the nano editor included in the pre-built image to edit and verify the module parameters in *wifi_mod_para.conf* configuration file

```
root@imx8mqevk:~# nano /lib/firmware/nxp/wifi_mod_para.conf
```

Content of the configuration file:

```
SD8997 = {
    ps_mode=2
    auto_ds=2
    cfg80211_wext=0xf
    cal_data_cfg=none
    fw_name=nxp/sdiouart8997_combo_v4.bin
    drv_mode=7
    max_vir_bss=1
    wfd_name=p2p
}
```

- Load the modules in the kernel

```
root@imx8mqevk:~# modprobe moal mod_para=nxp/wifi_mod_para.conf
```

Verify the kernel debug messages in the command output.

```
[   49.395686] wlan: Loading MWLAN driver
[   49.401313] vendor=0x02DF device=0x9141 class=0 function=1
[   49.409809] Attach moal handle ops, card interface type: 0x104
[   49.418621] SD8997: init module param from usr cfg
[   49.423457] card_type: SD8997, config block: 0
[   49.427951] ps_mode = 2
[   49.430397] auto_ds = 2
[   49.432860] cfg80211_wext=0xf
[   49.435843] cal_data_cfg=none
[   49.438811] fw_name=nxp/sdiouart8997_combo_v4.bin
[   49.443540] drv_mode = 7
[   49.446073] max_vir_bss=1
[   49.448714] wfd_name=p2p
[   49.451306] SDIO: max_segs=128 max_seg_size=65535
[   49.456036] rx_work=1 cpu_num=4
[   49.459243] Attach mlan adapter operations.card_type is 0x104.
[   49.466060] wlan: Enable TX SG mode
[   49.469570] wlan: Enable RX SG mode
[   49.479279] Request firmware: nxp/sdiouart8997_combo_v4.bin
[   50.201775] Wlan: FW download over, firmwarelen=543964 downloaded
 543964
[   51.071805] WLAN FW is active
[   51.074778] on_time is 51071607569
[   51.118478] fw_cap_info=0x181c3fa3, dev_cap_mask=0xffffffff
[   51.124075] max_p2p_conn = 8, max_sta_conn = 8
[   51.148460] wlan: version = SD8997----16.92.10.p216-MM5X16247.p1-
GPL-(FP92)
[   51.156963] usbcore: registered new interface driver usbxxx
[   51.162772] wlan: Driver loaded successfully
```

- Verify the Wi-Fi interfaces

```
root@imx8mqevk:~# ifconfig -a
```

Command output example:

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.0.103  netmask 255.255.255.0  broadcast 192.168.0.255
    inet6 fe80::204:9fff:fe06:c791  prefixlen 64  scopeid 0x20<link>
    ether 00:04:9f:06:c7:91  txqueuelen 1000  (Ethernet)
    RX packets 338  bytes 29157 (28.4 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 387  bytes 34568 (33.7 KiB)
    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 732  bytes 53888 (52.6 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 732  bytes 53888 (52.6 KiB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

mlan0: flags=4098<BROADCAST,MULTICAST>  mtu 1500
    ether 00:e9:3a:0d:c2:59  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

p2p0: flags=4098<BROADCAST,MULTICAST>  mtu 1500
    ether 02:e9:3a:0d:c2:59  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

uap0: flags=4098<BROADCAST,MULTICAST>  mtu 1500
    ether 00:e9:3a:0d:c3:59  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
```

- Get the Wi-Fi driver version

```
root@imx8mqevk:~# cat /proc/mwlan/adapter0/uap0/info
```

Command output example:

```
driver_name = "uap"
driver_version = SD8997----w8997o-V4, MXXX, FP92, 16.92.10.p216-
MM5X16247.p1-GPL-(FP92)
interface_name="uap0"
firmware_major_version=16.92.10
media_state="Disconnected"
mac_address="00:e9:3a:0d:c3:59"
```

5.4 Bring-up of 88W9098-based Murata module LBEE6ZZ1

Follow these instructions to load the driver modules and bring up the 88W9098-based wireless module.

- Use the *nano* editor included in the pre-built image to edit and verify the module parameters in *wifi_mod_para.conf* configuration file

```
root@imx8mqevk:~# nano /lib/firmware/nxp/wifi_mod_para.conf
```

Content of the configuration file:

```
PCIE9098_0 = {
      cfg80211_wext=0xf
      wfd_name=p2p
      max_vir_bss=1
      cal_data_cfg=none
      drv_mode=7
      ps_mode=2
      auto_ds=2
      fw_name=nxp/pcieuart9098_combo_v0.bin
}
```

- Load the modules in the kernel

```
root@imx8mqevk:~# modprobe moal mod_para=nxp/wifi_mod_para.conf
```

- Verify the kernel debug message in the command output

```
[ 1934.903485] wlan: Loading MWLAN driver
[ 1934.908682] usbcore: registered new interface driver usbxxx
[ 1934.914517] wlan_pcie 0001:01:00.0: enabling device (0000 -> 0002)
[ 1934.920862] Attach moal handle ops, card interface type: 0x206
[ 1934.926883] PCIE9098: init module param from usr cfg
[ 1934.931915] card_type: PCIE9098, config block: 0
[ 1934.936630] cfg80211_wext=0xf
[ 1934.939652] wfd_name=p2p
[ 1934.942186] max_vir_bss=1
[ 1934.944828] cal_data_cfg=none
[ 1934.947809] drv_mode = 7
[ 1934.950342] ps_mode = 2
[ 1934.952801] auto_ds = 2
[ 1934.955264] fw_name=nxp/pcieuart9098_combo_v1.bin
[ 1934.959993] rx_work=1 cpu_num=4
[ 1934.963148] Attach mlan adapter operations.card_type is 0x206.
[ 1934.971277] Request firmware: nxp/pcieuart9098_combo_v1.bin
[ 1935.403742] FW download over, size 617212 bytes
[ 1936.271780] WLAN FW is active
[ 1936.274751] on_time is 1936271521056
[ 1937.621201] fw_cap_info=0x81c3fa3, dev_cap_mask=0xffffffff
[ 1937.626749] max_p2p_conn = 8, max_sta_conn = 64
[ 1937.641095] wlan: version = PCIE9098--17.92.1.p55-MM5X17247-GPL-
(FP92)
[ 1937.650087] wlan_pcie 0001:01:00.1: enabling device (0000 -> 0002)
[ 1937.657779] Attach moal handle ops, card interface type: 0x206
[ 1937.665636] PCIE9098: init module param from usr cfg
[ 1937.670841] Configuration block, fallback processing
[ 1937.682944] Configuration fallback to, card_type: 0x206, blk_id:
 0x0
[ 1937.689503] rx_work=1 cpu_num=4
[ 1937.694195] Attach mlan adapter operations.card_type is 0x206.
[ 1937.707333] Request firmware: nxp/pcieuart9098_combo_v1.bin
[ 1937.713705] WLAN FW already running! Skip FW download
[ 1937.718879] WLAN FW is active
[ 1937.721903] on_time is 1937718673474
[ 1937.737820] fw_cap_info=0x81c3fa3, dev_cap_mask=0xffffffff
[ 1937.743389] max_p2p_conn = 8, max_sta_conn = 64
[ 1937.757214]wlan: version = PCIE9098--17.92.1.p55-MM5X17247-GPL-
(FP92)
[ 1937.765249] wlan: Driver loaded successfully
```

- Verify the Wi-Fi interfaces

```
root@imx8mqevk:~# ifconfig -a
```

Command output example

```
eth0: flags=-28669<UP,BROADCAST,MULTICAST,DYNAMIC>  mtu 1500
    ether 00:04:9f:06:c5:a5  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 380  bytes 25376 (24.7 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 380  bytes 25376 (24.7 KiB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

mlan0: flags=4098<BROADCAST,MULTICAST>  mtu 1500
    ether 2c:4c:c6:f4:67:b0  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

mmlan0: flags=4098<BROADCAST,MULTICAST>  mtu 1500
    ether 2c:4c:c6:f4:67:b1  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

muap0: flags=4098<BROADCAST,MULTICAST>  mtu 1500
    ether 2c:4c:c6:f4:68:b1  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

p2p0: flags=4098<BROADCAST,MULTICAST>  mtu 1500
    ether 2e:4c:c6:f4:67:b0  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

p2p1: flags=4098<BROADCAST,MULTICAST>  mtu 1500
    ether 2e:4c:c6:f4:67:b1  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

uap0: flags=4098<BROADCAST,MULTICAST>  mtu 1500
    ether 2c:4c:c6:f4:68:b0  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
```

Note: *NXP 88W9098-based wireless modules support Wi-Fi dual radios with dual MACs and dual independent Wi-Fi interfaces created by the driver for each mode of operation.*

- Get the Wi-Fi driver version

```
root@imx8mqevk:~# cat /proc/mwlan/adapter0/uap0/info
```

Command output example

```
driver_name = "uap"
driver_version = PCIE9098--w9098o-V1, MXXX, FP92, 17.92.1.p55-
MM5X17247-GPL-(FP92)
interface_name="uap0"
firmware_major_version=17.92.1
media_state="Disconnected"
mac_address="2c:4c:c6:f4:68:b0"
```

5.5 Bring-up of IW416-based AzureWave module AW-AM510MA

Follow these instructions to load the driver modules and bring up the IW416-based wireless module.

- Use the nano editor included in the pre-built image to edit and verify the module parameters in *wifi_mod_para.conf* configuration file

Content of the configuration file:

```
SD8978 = {
      cfg80211_wext=0xf
      wfd_name=p2p
      max_vir_bss=1
      cal_data_cfg=none
      drv_mode=7
      ps_mode=2
      auto_ds=2
      fw_name=nxp/sdiouartiw416_combo_v0.bin
}
```

- Load the modules in the kernel

```
root@imx8mqevk:~# modprobe moal mod_para=nxp/wifi_mod_para.conf
```

Verify the kernel debug messages in the command output:

```
[  558.903379] wlan: Loading MWLAN driver
[  559.018905] vendor=0x02DF device=0x9159 class=0 function=1
[  559.024535] Attach moal handle ops, card interface type: 0x108
[  559.030552] No module param cfg file specified
[  559.035015] SDIO: max_segs=128 max_seg_size=65535
[  559.039733] rx_work=1 cpu_num=4
[  559.042909] Attach mlan adapter operations.card_type is 0x108.
[  559.049601] wlan: Enable TX SG mode
[  559.053110] wlan: Enable RX SG mode
[  559.060287] Request firmware: nxp/sdiouart8978_combo_v0.bin
[  559.444220] Wlan: FW download over, firmwarelen=481164 downloaded 481164
[  560.855398] WLAN FW is active
[  560.858381] on_time is 560854191159
[  560.884715] fw_cap_info=0x181c0f03, dev_cap_mask=0xffffffff
[  560.890320] max_p2p_conn = 8, max_sta_conn = 8
[  560.915183] wlan: version = SDIW416---16.92.10.p233-MM5X16266.p1-GPL-(FP92)
[  560.924100] usbcore: registered new interface driver usbxxx
[  560.931152] wlan: Driver loaded successfully
```

- Verify the Wi-Fi interfaces

```
root@imx8mqevk:~# ifconfig -a
```

Command output example:

```
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
      ether 00:04:9f:06:c5:a5  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 524  bytes 34160 (33.3 KiB)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 524  bytes 34160 (33.3 KiB)
      TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

mlan0: flags=4098<BROADCAST,MULTICAST>  mtu 1500
      ether 48:e7:da:3c:79:d9  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

p2p0: flags=4098<BROADCAST,MULTICAST>  mtu 1500
      ether 4a:e7:da:3c:79:d9  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

uap0: flags=4098<BROADCAST,MULTICAST>  mtu 1500
      ether 48:e7:da:3c:7a:d9  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
```

- Get the Wi-Fi driver version

```
root@imx8mqevk:~# cat /proc/mwlan/adapter0/uap0/info
```

Command output example:

```
driver_name = "uap"
driver_version = SDIW416---w8978o-V0, MXXX, FP92, 16.92.10.p233-MM5X16266.p1-
GPL-(FP92)
interface_name="uap0"
firmware_major_version=16.92.10
media_state="Disconnected"
mac_address="48:e7:da:3c:7a:d9"
```

5.6 Bring up the Wi-Fi interface

Use the following steps to bring up the Wi-Fi interfaces

- Invoke the command to initialize *mlan0* interface

```
root@imx8mqevk:~# ifconfig mlan0 up
root@imx8mqevk:~# ifconfig mlan0
mlan0   Link encap:Ethernet  HWaddr 00:50:43:24:83:c4
        UP BROADCAST MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

- Invoke the command to initialize the *uap0* interface

```
root@imx8mqevk:~# ifconfig uap0 up
root@imx8mqevk:~# ifconfig uap0
uap0    Link encap:Ethernet  HWaddr 00:50:43:24:84:c4
        UP BROADCAST MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

6 Bring-up of Bluetooth interfaces

Note: *You must first load the combo firmware and initialize the Wi-Fi driver prior to initializing the Bluetooth interface. Refer to* Section 3.4.7 "Plugging the wireless module" *and* Section 5.6 "Bring up the Wi-Fi interface" *.*

Use the following steps to bring up the Bluetooth interfaces.

Note: *IW416-based and 88W9098-based wireless modules work on 3M baud rate, so skip to step no. 3 for IW416-based and 88W9098-based modules.*

- **Step 1** - Invoke the command to initialize *hci0* interface

```
root@imx8mqevk:~# hciattach /dev/ttymxc2 any 115200 flow
Setting TTY to N_HCI line discipline
Device setup complete
root@imx8mqevk:~# hciconfig hci0 up
root@imx8mqevk:~# hciconfig
hci0:   Type: Primary  Bus: UART
BD Address: 00:50:43:24:83:B7 ACL MTU: 1021:7  SCO MTU: 120:6
UP RUNNING
RX bytes:1451 acl:0 sco:0 events:84 errors:0
TX bytes:789 acl:0 sco:0 commands:84 errors:0
```

- **Step 2** - Invoke the HCI command to change the speed where `<device>` is */dev/ttymxc2*.

```
root@imx8mqevk:~# hcitool -i hci0 cmd 0x3f 0x0009 0xc0 0xc6 0x2d 0x00
```

- **Step 3** - Invoke the HCI command to initialize *hci0* interface on 3M baud rate

```
root@imx8mqevk:~# killall hciattach
root@imx8mqevk:~# hciattach /dev/ttymxc2 any 3000000 flow
root@imx8mqevk:~# hciconfig hci0 up
```

- **Step 4** - Invoke the HCI tool command to scan nearby Bluetooth devices

```
root@imx8mqevk:~# hcitool -i hci0 scan
Scanning ...
    40:49:0F:9E:66:FE       desktop
root@imx8mqevk:~#
```

7 Contact information

Use the following links for more product details, queries and support.

Home page: www.nxp.com

Web support: nxp.com/support

NXP community: community.nxp.com

iMX community: community.nxp.com/community/imx

8 Acronyms and abbreviations

Table 13. Acronyms and abbreviations

Acronyms	Definition
AP	Access point
BSP	Board support package
BT	Bluetooth
DTB	Device tree blob
EVK	Evaluation kit
FW	Firmware
STA	Station
uSD	Micro SD
WLAN	Wireless local area network
WPA	Wi-Fi protected access

9 Legal information

9.1 Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

9.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors. In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory. Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification. Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products. NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer. In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages. Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

9.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

Tables

Figures

Contents

UM11483

Getting Started with NXP-based Wireless Modules on i.MX 8M Quad EVK Running Linux OS

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 21 September 2021
Document identifier: UM11483