UM12298

Getting Started with IW610 module on FRDM i.MX 91 development board running Linux OS

Rev. 1.0 — 3 July 2025

User manual

Document information

Information	Content
Keywords	FRDM i.MX 91 development board, FRDM-IMX91 board, production software, Wi-Fi, Bluetooth LE, 802.15.4, Linux OS, on-board MAYA-W476-00B (IW610 u-blox module)
Abstract	Provides the steps to set up IW610 module to FRDM i.MX 91 development board running Linux OS, transfer the software files, and bring up Wi-Fi, Bluetooth LE, OpenThread.

1 About this document

This document explains how to bring up Wi-Fi, Bluetooth LE and 802.15.4 with IW610 module placed on the FRDM i.MX 91 development board running Linux OS.

The user manual covers the following:

- How to set up IW610 module on FRDM i.MX 91 development board.
- How to connect FRDM i.MX 91 development board to a PC.
- How to download and install IW610 software.
- How to bring up Wi-Fi over SDIO.
- How to bring up Bluetooth LE over UART.
- How to bring up 802.15.4 over SPI.

2 FRDM-IMX91 evaluation kit content

The evaluation kit for FRDM-IMX91 includes:

- FRDM-IMX91 evaluation board
- Two USB 2.0 type-C male to type-A male assembly cables
- On-board MAYA-W476-00B (IW610 u-blox module)
- FRDM-IMX91 Quick Start Guide

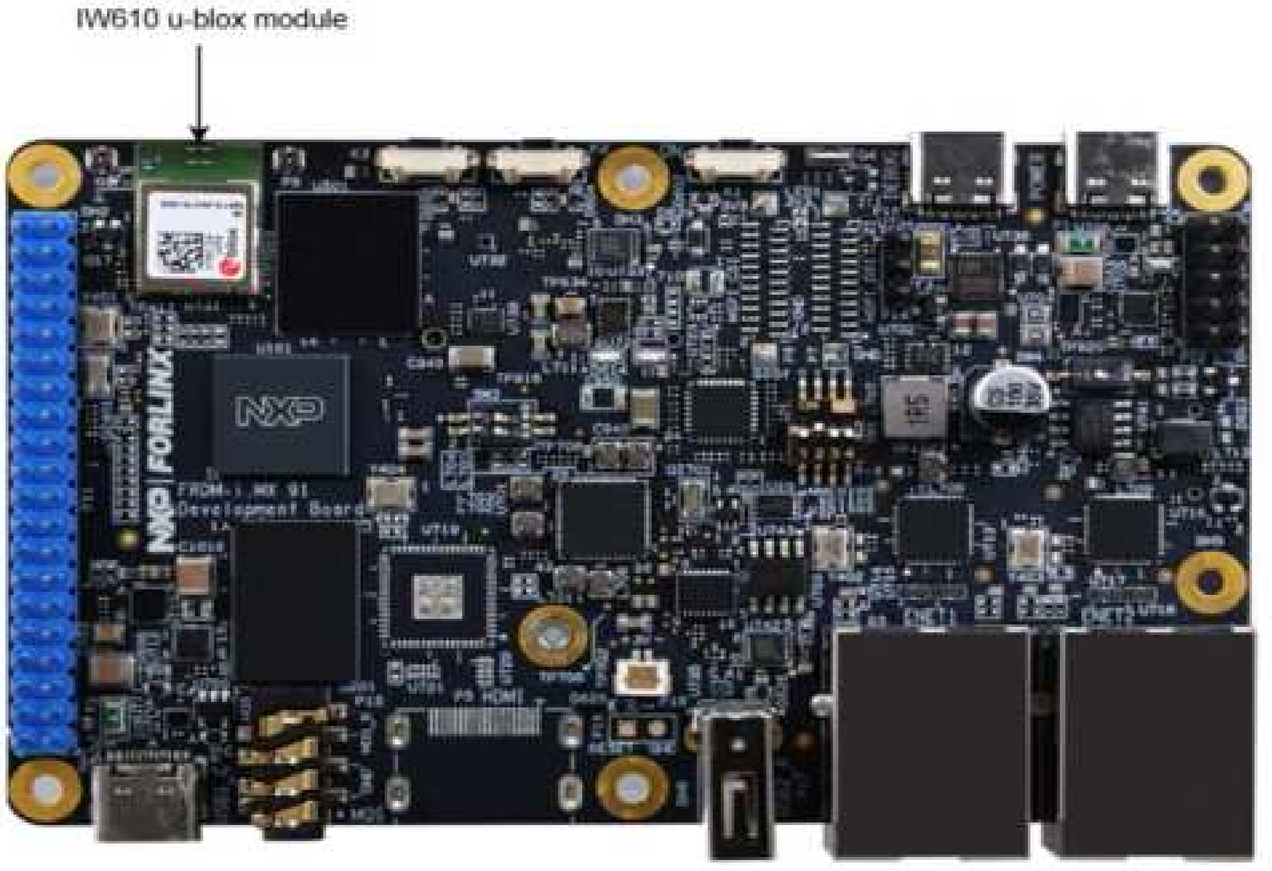


Figure 1. FRDM i.MX 91 development board

3 IW610 u-blox module – MAYA-W476-00B

This section provides an overview of IW610 u-blox module.

3.1 Antenna configuration

The two antenna pins (RF_ANT0 and RF_ANT1) of the module connect to U.FL connectors P9 and P10 (DNP by default). The module is supplied with VPCIe_3V3, VEXT_1V8, and VDD_1V8.

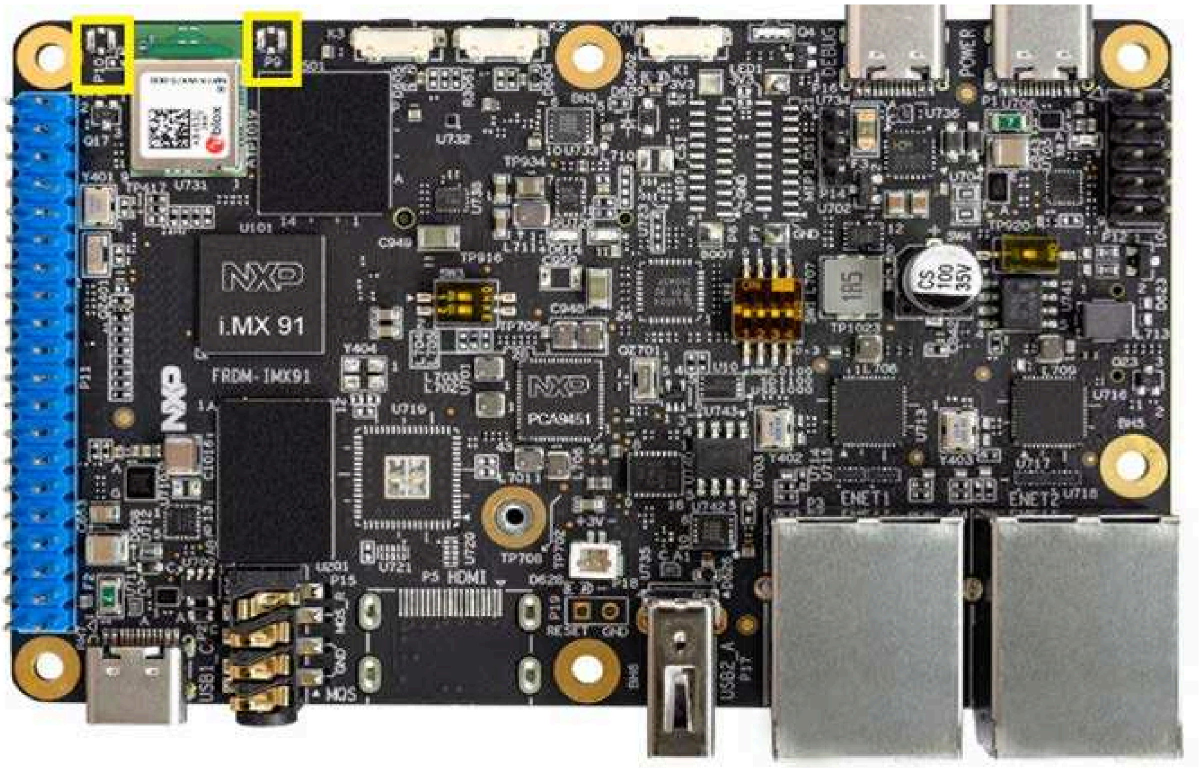


Figure 2. Antenna configuration of IW610 module

3.2 Host interface configuration

The module MAYA-W476-00B is pre-configured to boot-up via an M.2 connector with the following host interfaces:

- Wi-Fi: SDIO
- Bluetooth LE: UART
- 802.15.4: SPI

The MAYA-W476-00B module and M.2 connector share several interface lines on the FRDM i.MX 91 development board. Zero-ohm resistors enable the signal selection between the components.

3.3 SD3 interface

The SD3 interface lines are shared between the MAYA-W476-00B module and the M.2 connector. Zero-ohm resistors select either the MAYA-W476-00B module (default setting) or the M.2 connector (Figure 3).

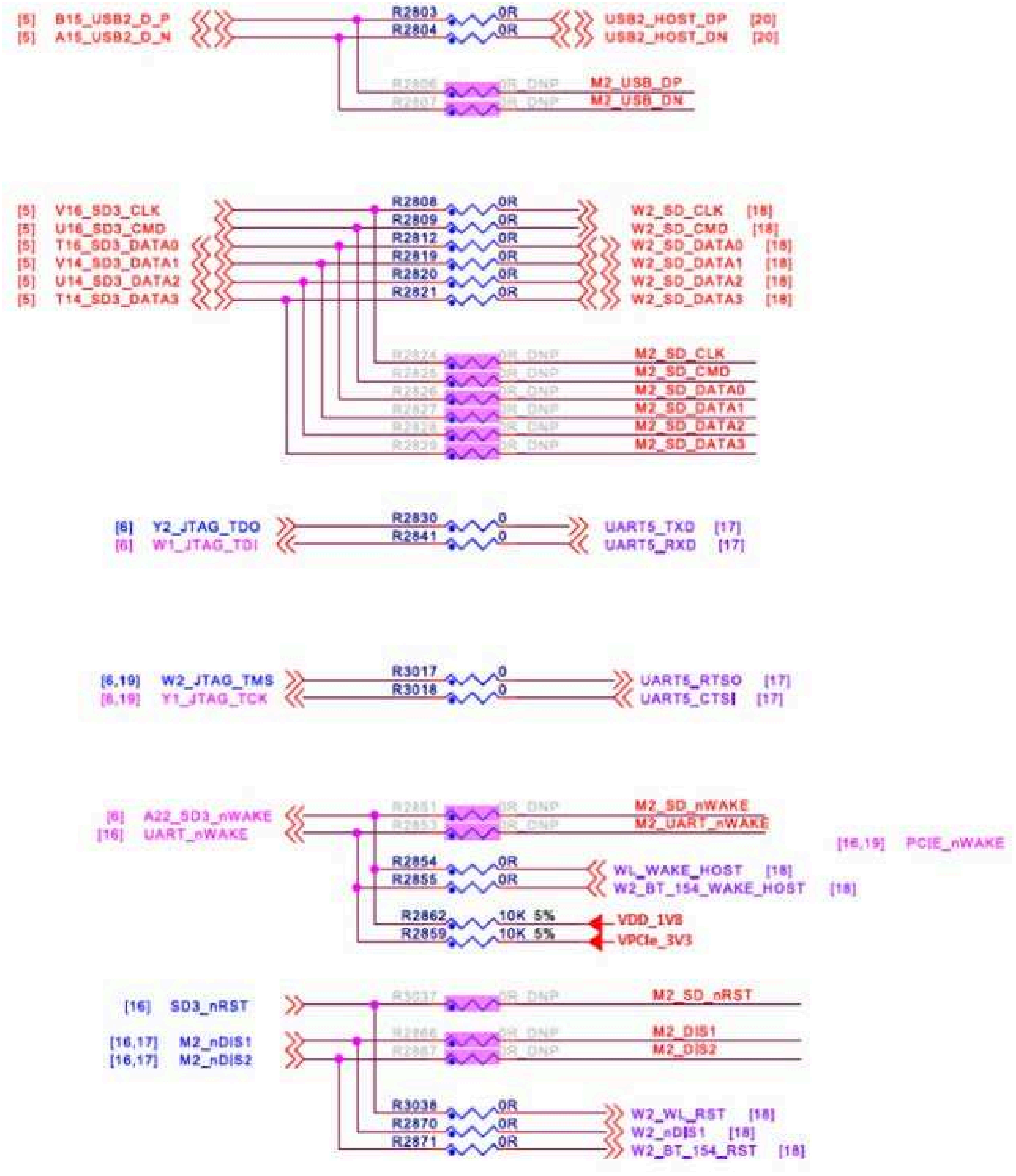


Figure 3. Resistors configuration for SD3

3.4 UART5, SAI1, and SPI3 interfaces

UART5 interface

Similar to SD3 lines, the UART5 interface lines are shared between the MAYA-W476-00B module and the M.2 connector. Zero-ohm resistors select either the MAYA-W476-00B module (default setting) or the M.2 connector (Figure 4).

SAI1 interface

The SAI1 interface lines are shared between the MAYA-W476-00B module and the M.2 connector. Zero-ohm resistors select either the MAYA-W476-00B module (default setting) or the M.2 connector for the 1.8 V translated signals which are generated using the 74AVC4T3144 bidirectional voltage translator (U728) (Figure 4).

SPI3 interface

The SPI3 signals (CLK, MOSI, MISO, and CS0) are multiplexed with GPIO_IO[08, 09, 10, 11] signals, respectively. The SPI3 signals are shared between the MAYA-W476-00B module and the M.2 connector. Zero-ohm resistors select either the MAYA-W476-00B module (default setting) or the M.2 connector for the 1.8 V translated signals which are generated using the 74AVC4T3144 bidirectional voltage translator (U729). See (Figure 4).

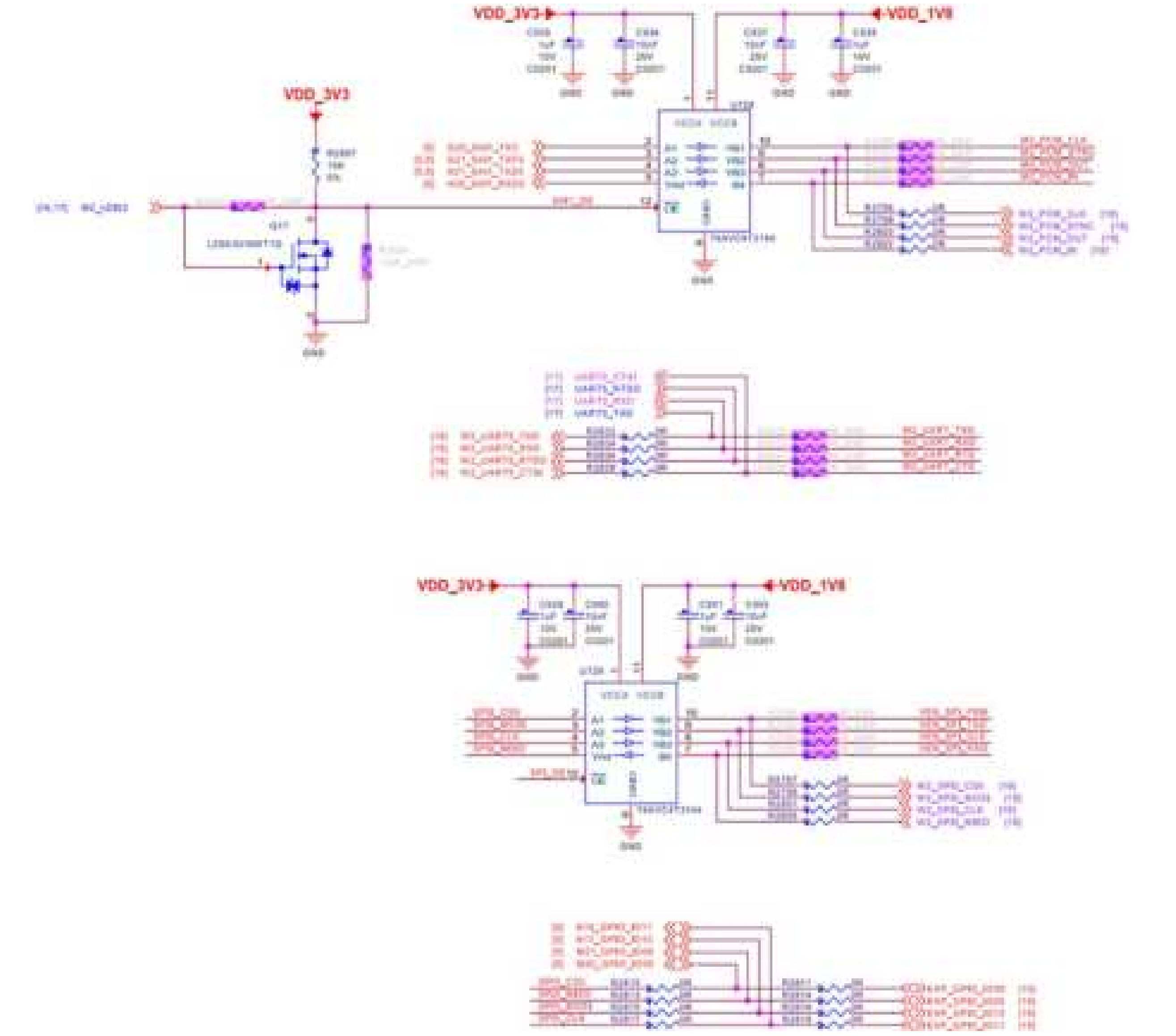
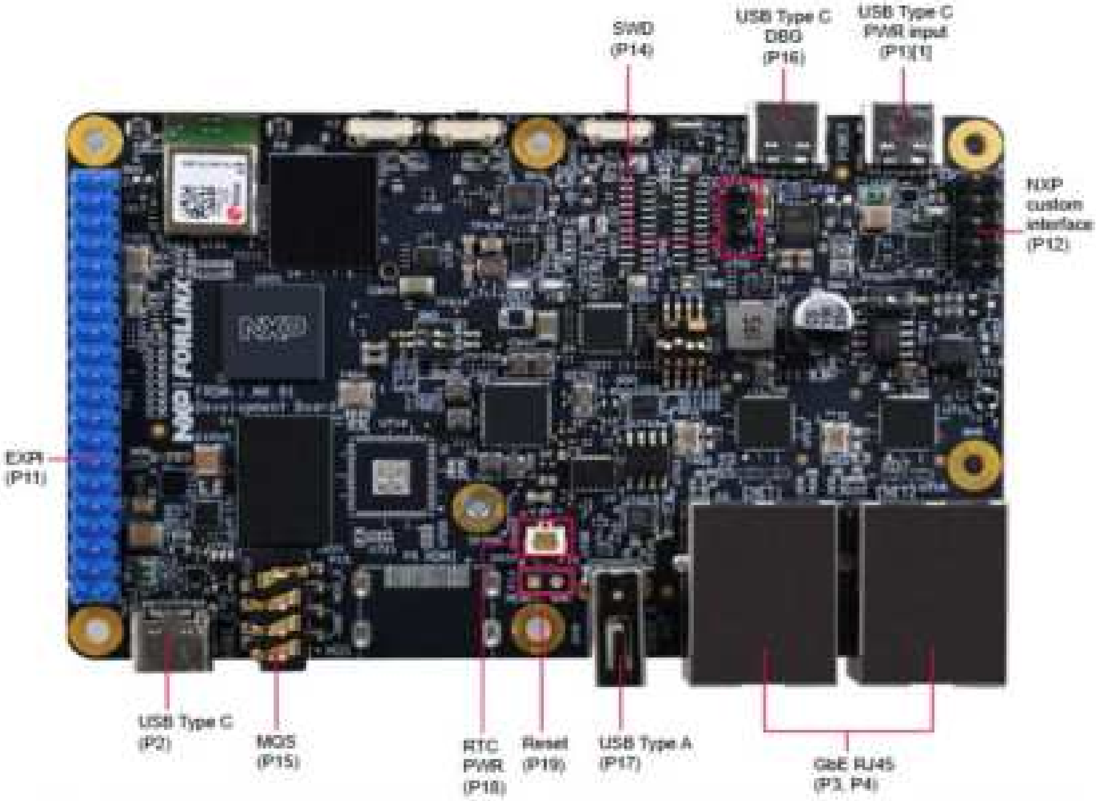


Figure 4. Resistor configurations for SAI1, UART5 and SPI3

4 FRDM-IMX91 development board overview

This section describes the peripheral sockets on FRDM-IMX91 development board.



[1] - USB Type C PWR input (P1) shown in the figure is the only power supply port, and must always be supplied for system running.

Figure 5. FRDM-i.MX 91 peripheral sockets

Figure 6 shows the back view of FRDM i.MX 91 development board.

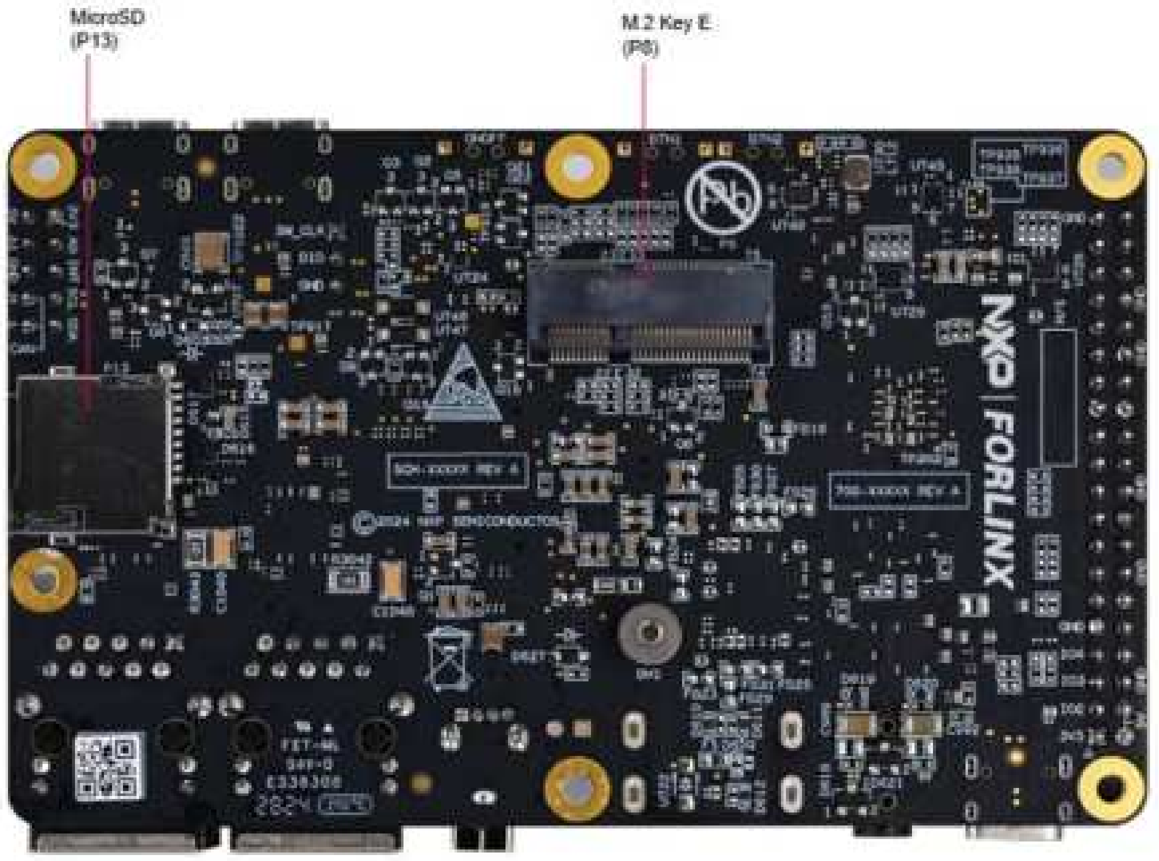


Figure 6. Back view of FRDM i.MX 91 development board

Table 1 details the use of the peripherals on FRDM i.MX 91 development board.

Table 1. Use of peripherals on FRDM i.MX 91 development board

Peripheral	Usage
Debug UART	Console access
USB-C download	Flash the platform with Linux
USB-C power	Power supply
Ethernet port (x2)	Ethernet access
Micro-SD slot	MicroSD card connector (SD3.0 supported)
Parallel display	Parallel display support
M.2 interface	One M.2/NGFF Key E mini card 75-pin connector, P8, supporting USB, SDIO, SAI, UART, I2C, and Vendor-defined SPI interfaces
EXPI	One 2x20-pin EXPI for I2S, UART, I2C, and GPIO expansion
MQS	Audio
Tri-radio module	U-blox MAYA-W476-00B module supporting Wi-Fi 6, Bluetooth Low Energy 5.4, and 802.15.4

Table 1. Use of peripherals on FRDM i.MX 91 development board*...continued*

Peripheral	Usage
I/O expanders	One 10-pin 2x5 2.54 mm connector P12 with: [question: what are the dimensions? 2 x 5 x 2.54 mm?] • One high-speed CAN transceiver TJA1051GT/3 connection • 3-pin header for I2C/I3C expansion • 2-channel ADC support
USB Type A	Connection to full-speed USB host and device controller

5 Board setup

This section details the setup of FRDM i.MX 91 development board with IW610 on-board module.

5.1 FRDM i.MX 91 development board

To get started with FRDM i.MX 91 development board, see ref.[10].

Step 1 – Change the boot mode switch to Serial Download.

Table 2. Boot mode switch settings

SW1[1:4]	BOOT_MODE[3:0]	Boot mode
1100	0001	Serial downloader (USB)
0000	0010	uSDHC1 8-bit eMMC 5.1
0100	0011	uSDHC2 4-bit SD3.0

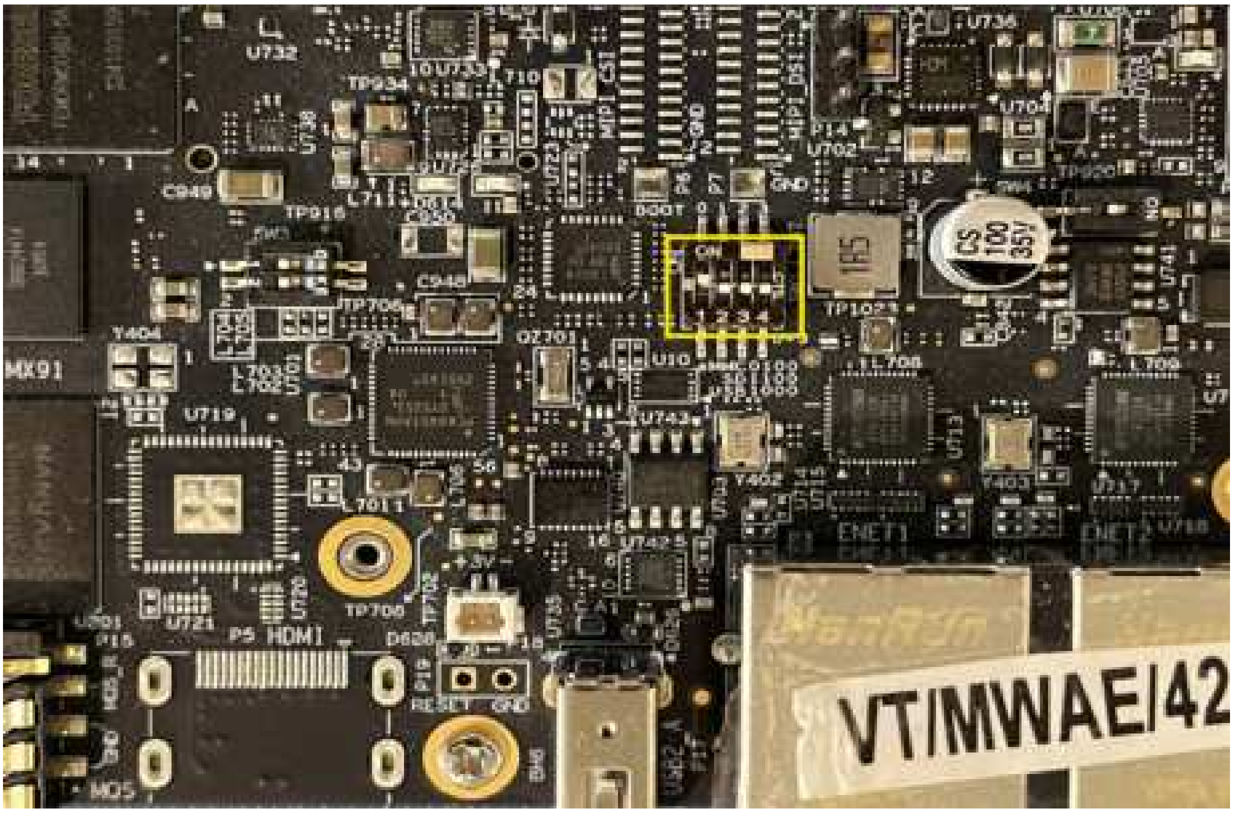


Figure 7. Boot mode selection switch

Step 2 – Connect the board to the PC.

Connect the two type-C cables to the power supply and to the USB port on FRDM i.MX 91 development board respectively (Figure 8).

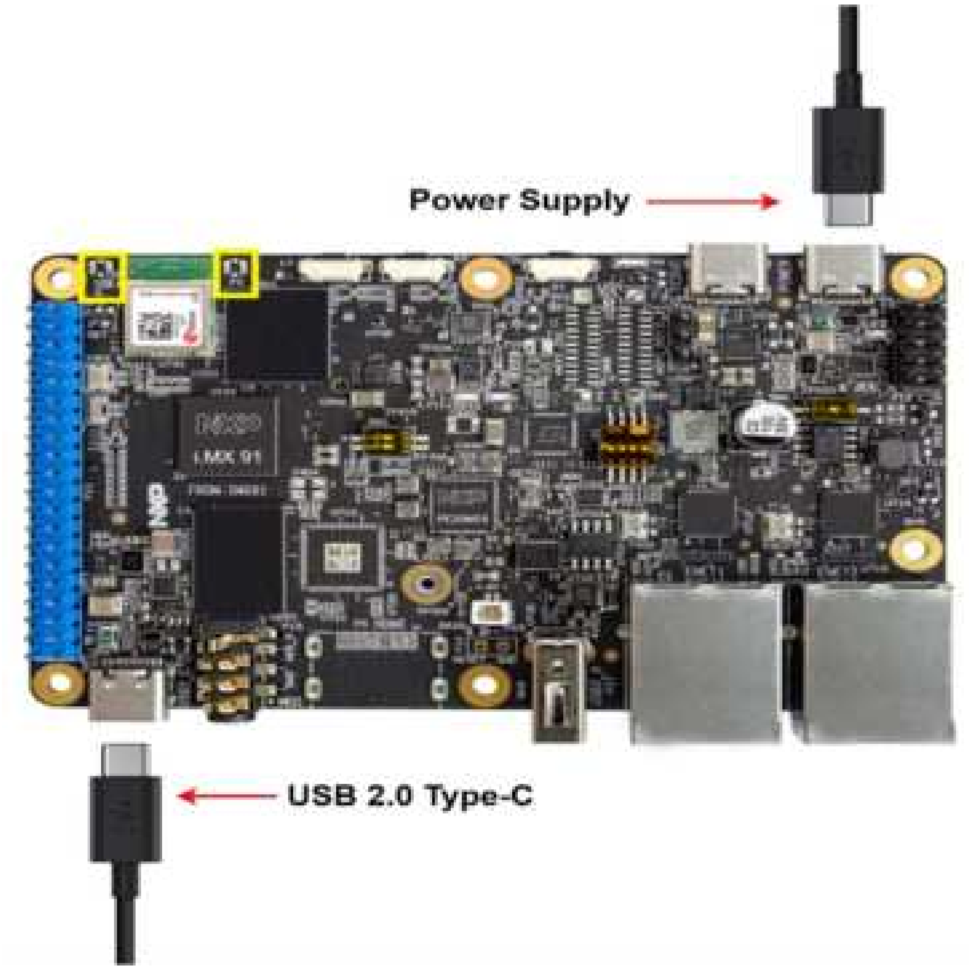


Figure 8. FRDM i.MX 91 development board connection setup

Step 3 – Download Linux BSP and *uuu.exe*.

- Get the latest pre-built Linux image for i.MX 91 EVK, QSB from ref.[8]. The file contains the demo Linux image.
- To bring up 802.15.4, follow the steps to build the BSP from ref.[2].
- Download the latest *uuu.exe* from ref.[3]. Save the executable in the same directory as the Linux BSP.
- Issue the command to flash the full pre-compiled BSP.

```
uuu.exe <Linux_BSP.zip>
```

The flashing takes around 5 minutes.

Example of command output:

```
uuu (Universal Update Utility) for nxp imx chips -- libuuu_1.5.182-0-gda3cd53
Success 1 Failure 0
1:2-FC3E75D9 8/ 8 [Done] FB: done
```

Step 4 – Turn OFF the board.

Step 5 – Reset the boot mode switches to e-mmc boot.

Table 3. Boot mode switch settings

SW1[1:4]	BOOT_MODE[3:0]	Boot Mode
1100	0001	Serial downloader (USB)
0000	0010	uSDHC1 8-bit eMMC 5.1
0100	0011	uSDHC2 4-bit SD3.0

Step 6 – Set up debug access.

- Plug the Micro-USB cable to the UART-debug port on FRDM i.MX 91 development board.
- Plug the other end of the Micro-USB cable to the PC.
- Download the Serial Terminal program.
  - On Windows: install PUTTY (ref.[5]), teraterm (ref.[6]), or USB Device driver (ref.[7]).
  - On Linux: issue the command to download and install Minicom:

```
sudo apt-get install minicom
```

Step 7 – Check the baudrate and other serial port settings in the serial terminal window on the PC.

Command syntax for minicom:

```
ubuntu@ubuntu-desktop:/# sudo minicom -s -D /dev/<ttyACMx>
```

The `-s` option shows the configuration settings on the serial console.

Table 4. Command parameters

Parameter	Description
`ttyACMx`	ttyACM0 or ttyACM1 are the serial devices.

- Select the `serial port setup` to verify the configuration settings as shown below.

```
A - Serial Device : /dev/ttyACM0
E - Bps/Par/Bits : 115200 8N1
F - Hardware Flow Control : No
G - Software Flow Control : No
```

- Save and exit.

Step 8 – Enter the login information when prompted.

```
imx91evk login: root
```

6 IW610 software

6.1 Download

IW610 software is available for download in the Design resource section of IW610 product page at nxp.com.

- Go to IW610 product page (ref.[9]).
- Click **Design Resources**.

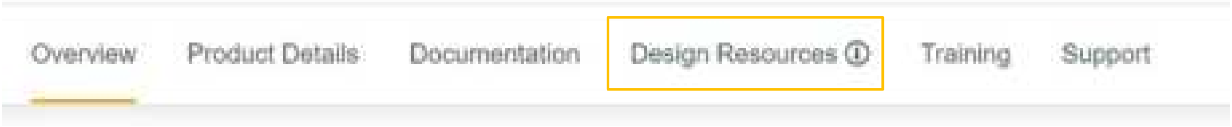
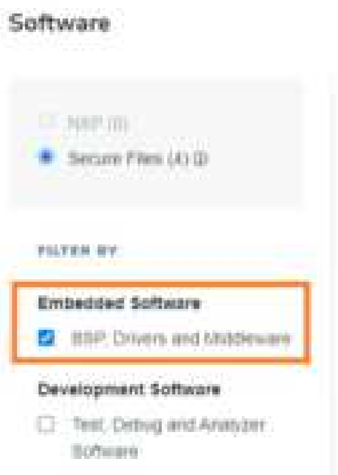


Figure 9. Design resources tab

- Sign in to access the secure files.
- Select **Design Resources** again.
- Scroll down to **Software**.
- Select **Embedded Software**.
- Select **BSP, Drivers, and Middleware** (Figure 10).


Software

○ NXP (0)
● Secure Files (4) ⓘ

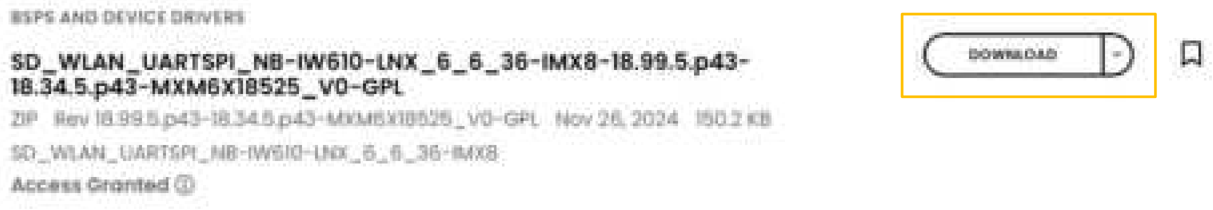
FILTER BY

Embedded Software
☑ BSP, Drivers and Middleware

Development Software
□ Test, Debug and Analyzer Software

Figure 10. Select BSP, Drivers and Middleware

- Look for the latest release of IW610 software and click **Download** (Figure 11).


BSPS AND DEVICE DRIVERS

SD_WLAN_UARTSPI_NB-IW610-LNX_6_6_36-IMX8-18.99.5.p43-18.34.5.p43-MXM6X18525_V0-GPL

ZIP Rev 18.99.5.p43-18.34.5.p43-MXM6X18525_V0-GPL Nov 26, 2024 150.2 KB

SD_WLAN_UARTSPI_NB-IW610-LNX_6_6_36-IMX8

Access Granted ⓘ

Figure 11. IW610 software download

6.2 Firmware type

Table 5 describes the files included in the *FwImage* directory of IW610 software package.

Table 5. Firmware types and file names

Firmware type	Firmware file name
Combo (Wi-Fi + Bluetooth LE + 802.15.4)	*sduartspi_iw610.bin.se*
Combo (Wi-Fi + Bluetooth LE)	*sduart_iw610.bin.se*
Standalone Bluetooth LE + 802.15.4	*uartspi_iw610.bin.se*
Standalone Wi-Fi	*sd_iw610.bin.se*
Standalone Bluetooth LE	*uartiw610_bt.bin.se*

7 Software installation

This section explains how to install IW610 software.

- The software is downloaded on the host PC and copied to FRDM i.MX 91 development board running Linux OS.
- The host interfaces between IW610 and FRDM i.MX 91 development board are used to install the software on IW610.

Note: *To compile drivers and utilities, follow the guidelines in ref.[4].*

7.1 File transfer to FRDM i.MX 91 development board

This section explains the two methods to transfer files from the host PC to FRDM i.MX 91 development board. Refer to Section 6 for details on the software release content.

- One method uses USB-C to USB adapter (recommended).
- The other method uses SCP utility.

7.1.1 USB-C to USB-A adapter

The procedure to use USB-C to USB adapter to transfer files is as follows:

Step 1 – Connect the hardware.

- Plug the USB-C to USB adapter to the Download slot on FRDM i.MX 91 development board.
- Plug the USB flash drive containing the downloaded IW610 software to the other end of the USB adapter.

Note: *Linux supports FAT32 and exFAT USB formats. Other formats may run into an error.*

Figure 12 shows USB-C to USB-A adapter connections.

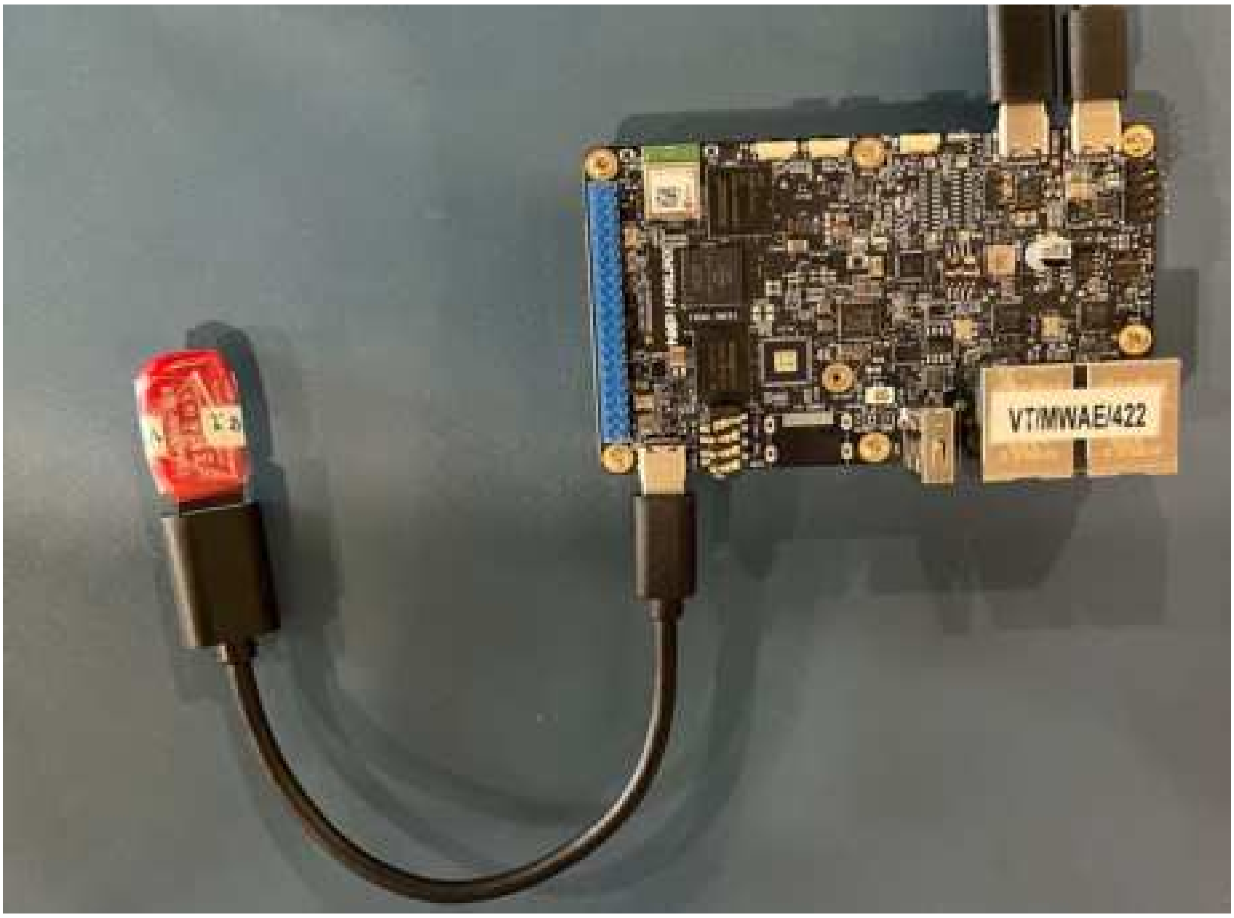


Figure 12. USB-C to USB-A adapter connections

Step 2 – Debug access.

- To bring up FRDM i.MX 91 development board, refer to Section 5.1.

Step 3 – Transfer the files.

- Issue the command to locate the USB flash drive content.

```
lsblk
```

Example of command output:

```
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
sda 8:0 1 59.6G 0 disk
`-sda1 8:1 1 59.6G 0 part /run/media/sda1
mtdblock0 31:0 0 32M 0 disk
mmcblk2 179:0 0 14.7G 0 disk
|-mmcblk2p1 179:1 0 83.2M 0 part /run/media/mmcblk2p1
`-mmcblk2p2 179:2 0 4.5G 0 part /
mmcblk2boot0 179:32 0 4M 1 disk
mmcblk2boot1 179:64 0 4M 1 disk
```

The content of the USB flash drive is located in the directory */run/media/sda1*.

- Go to USB flash drive directory.

  ```
  cd <USB directory>
  ```

- Copy the files to */home/root* directory.

  ```
  cp -r /Drivers /home/root
  cp -r /FWImage /home/root
  ```

7.1.2 SCP utility (optional)

This section explains how to use SCP utility to transfer files from the host PC to FRDM i.MX 91 development board via Ethernet. For details on the software release content, refer to Section 6.

Step 1 – Set up the Ethernet interface.

To connect the two interfaces on the same network, attach one end of the ethernet cable to FRDM-i.MX 91 Ethernet port and the other end to the host PC.

Note: *The ethernet port is used either to transfer files or for debug access (SSH).*

- Get the IP address of the connected PC.

```
Ifconfig
```

- Assign the IP address of FRDM-i.MX91 on eth0 interface.

```
imx91evk login:~# ifconfig eth0 192.168.1.100 up
```

- Verify that eth0 interface is up.

```
#  ifconfig eth0
```

Example of output for FRDM i.MX 91 board connected to the PC and matching IP addresses:

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.1.100 netmask 255.255.255.0 broadcast 192.168.1.255
inet6 fe80::204:9fff:fe07:1230 prefixlen 64 scopeid 0x20<link>
ether 00:04:9f:07:12:30 txqueuelen 1000 (Ethernet)
RX packets 1817 bytes 2605025 (2.4 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 272 bytes 32259 (31.5 KiB)
```

- **Step 2** – Copy the Wi-Fi driver module directory (*Drivers*) to FRDM i.MX 91 development board.

```
scp -r /Drivers root@192.168.1.100:/home/root
```

Note: *For Bluetooth over UART, use the native UART driver included in the Linux Kernel.*

Step 3 – Copy the firmware binaries

- Copy the *FWImage* directory to FRDM i.MX 91 development board:

```
scp -r /FWImage root@192.168.1.100:/home/root
```

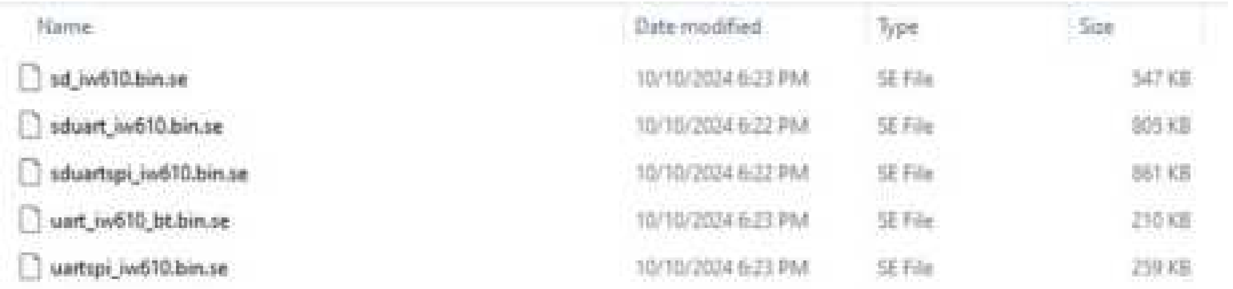
- List the content of *FwImage* directory.

```
~/FWImage# ls
```

Example of command output:

```
sd_iw610.bin.se sduart_iw610.bin.se sduartspi_iw610.bin.se uart_iw610_bt.bin.se
uartspi_iw610.bin.se
```

Figure 13 shows the content of *FwImage* directory.



Name	Date modified	Type	Size
sd_iw610.bin.se	10/10/2024 6:23 PM	SE File	547 KB
sduart_iw610.bin.se	10/10/2024 6:22 PM	SE File	805 KB
sduartspi_iw610.bin.se	10/10/2024 6:22 PM	SE File	861 KB
uart_iw610_bt.bin.se	10/10/2024 6:23 PM	SE File	210 KB
uartspi_iw610.bin.se	10/10/2024 6:23 PM	SE File	259 KB

Figure 13. Content of *FwImage* directory

8 Firmware download onto IW610

This section provides the commands to:

- Copy the firmware to FRDM i.MX 91 development board.
- Load the module drivers and firmware onto IW610.

8.1 Combo Wi-Fi, Bluetooth, and 802.15.4

Step 1 – Copy the firmware binary.

- Go to the firmware directory.

```
cd /lib/firmware
```

- Create nxp directory inside the firmware directory.

```
mkdir nxp
```

- Copy the combo firmware.

```
cp FwImage/sduartspi_iw610.bin.se /lib/firmware/nxp
```

Step 2 – Load the module drivers and firmware.

- Update the *wifi_mod_para.conf* file for IW610.

```
vi /lib/firmware/nxp/wifi_mod_para.conf
Sample driver parameters configurations for IW610:
SDIW610 = {
cfg80211_wext=0xf
max_vir_bss=1
ps_mode=1
auto_ds=1
host_mlme=1
fw_name=sduartspi_iw610.bin.se
}
```

- Use the modprobe command to load the drivers.

```
modprobe moal mod_para=nxp/wifi_mod_para.conf
```

Step 3 – Verify that the Wi-Fi interfaces are up.

```
ifconfig -a
```

Example of command output:

```
mlan0: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        ether b8:f4:4f:ab:41:9d  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 ba:f4:4f:ab:42:9d  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
wfd0: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        ether ba:f4:4f:ab:41:9d  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
```

Step 4 – Bring up the Bluetooth interface.

```
modprobe btnxpuart
hciconfig hci0 up
hciconfig –a
```

Example of command output:

```
hci0:   Type: Primary  Bus: UART
        BD Address: B8:F4:4F:AB:41:9C  ACL MTU: 256:8  SCO MTU: 0:0
        UP RUNNING
        RX bytes:615 acl:0 sco:0 events:52 errors:0
        TX bytes:346 acl:0 sco:0 commands:52 errors:0
        Features: 0x00 0x00 0x00 0x00 0x60 0x00 0x00 0x00
        Packet type: DM1 DH1 HV1
        Link policy:
        Link mode: PERIPHERAL ACCEPT
```

8.2 Standalone Wi-Fi

Step 1 – Copy the standalone Wi-Fi firmware binary.

```
cp FwImage/sd_iw610.bin.se /lib/firmware/nxp/
```

Step 2 – Load the module drivers and firmware.

- Update the *wifi_mod_para.conf* file for IW610.

```
vi /lib/firmware/nxp/wifi_mod_para.conf
Sample driver parameters configurations for IW610: SDIW610 = {
cfg80211_wext=0xf
max_vir_bss=1
ps_mode=1
auto_ds=1
host_mlme=1
fw_name=nxp/sd_iw610.bin.se
}
```

- Use the `modprobe` command to load the drivers.

```
modprobe moal mod_para=nxp/wifi_mod_para.conf
```

Step 3 – Verify that the Wi-Fi interfaces are up.

```
ifconfig -a
```

Example of command output:

```
mlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether b8:f4:4f:ab:41:9d 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 ba:f4:4f:ab:42:9d 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
wfd0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether ba:f4:4f:ab:41:9d 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
```

8.3 Standalone Bluetooth

Step 1 – Copy the standalone Bluetooth firmware binary.

```
cp FwImage/uart_iw610_bt.bin.se  /lib/firmware/nxp
```

Step 2 – Load the module drivers and firmware.

- Update the *wifi_mod_para.conf* file for IW610.

```
vi /lib/firmware/nxp/wifi_mod_para.conf
```

Example of driver configuration for IW610:

```
SDIW610 = {
cfg80211_wext=0xf
max_vir_bss=1
ps_mode=1
auto_ds=1
host_mlme=1
fw_name=nxp/uart_iw610_bt.bin.se
}
```

Step 3 – Initialize the Bluetooth interface hci0.

```
modprobe btnxpuart
```

Step 4 – Set up the Bluetooth interface and verify the status.

```
hciconfig hci0 up
hciconfig -a
```

Example of command output:

```
hci0: Type: Primary Bus: UART
       BD Address: B8:F4:4F:AB:41:9C ACL MTU: 256:8 SCO MTU: 0:0
       UP RUNNING
       RX bytes:964 acl:0 sco:0 events:84 errors:0
       TX bytes:572 acl:0 sco:0 commands:83 errors:0
       Features: 0x00 0x00 0x00 0x00 0x60 0x00 0x00 0x00
       Packet type: DM1 DH1 HV1
       Link policy:
       Link mode: PERIPHERAL ACCEPT
```

9 Bring-up of Wi-Fi

This section shows how to bring up Wi-Fi in STA and uAP modes.

Ensure the combo firmware or standalone Wi-Fi firmware is downloaded before bringing up the Wi-Fi radio in STA or uAP mode. Refer to Section 8.1 and Section 8.3.

9.1 STA mode

Step 1 – Open the configuration file.

The configuration file */etc/wpa_supplicant.conf* includes the network details of the external-AP running on 5 GHz band. The external AP could be your home/office Wi-Fi router or Mobile hotspot.

Example of configuration file content *wpa_supplicant.conf* for WPA2 security:

```
nano /etc/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
update_config=1
network={
     ssid="NXP_Demo"
     key_mgmt=WPA-PSK
     psk="123456789"
}
```

Step 2 – To avoid bring-up failures of the Wi-Fi radio in STA mode, stop all earlier instances of the wpa_supplicant.

```
killall wpa_supplicant
wpa_supplicant: no process found
```

Step3 – Start wpa_supplicant in background.

```
wpa_supplicant -imlan0 -Dnl80211 -c/etc/wpa_supplicant.conf -B
```

Step 4 – To get the dynamic IP address from the external AP, start the udhcp client.

```
udhcpc -i mlan0
udhcpc: started, v1.36.1
udhcpc: broadcasting discover
udhcpc: broadcasting select for 192.168.86.74, server
192.168.86.214
udhcpc: lease of 192.168.86.74 obtained from 192.168.86.214,
lease time 3599
/etc/udhcpc.d/50default: Adding DNS 192.168.86.214
```

Step 5 – Verify the IP address assigned to the STA device interface.

```
ifconfig mlan0
```

Example of command output:

```
mlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.86.74 netmask 255.255.255.0 broadcast
192.168.86.255
inet6 fe80::baf4:4fff:feab:4361 prefixlen 64 scopeid
0x20<link>
ether b8:f4:4f:ab:43:61 txqueuelen 1000 (Ethernet)
RX packets 6 bytes 1004 (1004.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 35 bytes 5224 (5.1 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

9.2 AP mode

This section shows how to configure the Wi-Fi radio in AP mode on the FRDM i.MX 91 development board using the */etc/hostapd.conf* configuration file.

Step 1 – Create a backup version of the configuration file.

```
mv /etc/hostapd.conf /etc/hostapd-default.conf
```

Step 2 – Create the configuration file */etc/hostapd.conf* with WPA2 security and AP 5GHz band profile configuration. The file is a simple text-based configuration file.

```
nano /etc/hostapd.conf
interface=uap0
hw_mode=a
channel=0
country_code=US
ssid=NXP_Demo
auth_algs=1
ieee80211n=1
wpa=2
wpa_key_mgmt=WPA-PSK
rsn_pairwise=CCMP
wpa_passphrase=123456789
```

Step 3 – To avoid bring-up failures, stop any running instance of the hostapd.

```
killall hostapd
```

Step 4- Assign an IP address to the AP interface.

```
ifconfig uap0 192.168.1.2 netmask 255.255.255.0
```

Step 5 – Verify the IP address and netmask assigned to the uap0 interface.

```
ifconfig uap0
```

Command output example:

```
uap0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
inet 192.168.1.2 netmask 255.255.255.0 broadcast 192.168.1.255
ether ba:f4:4f:ab:44:61 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
```

Step 6 – To assign the IP address to the client devices to connect to the created AP, start the udhcp server.

```
udhcpd /etc/udhcpd.conf
```

Step 7 – Start hostapd in the background.

```
hostapd /etc/hostapd.conf -B
```

Example of command output:

```
uap0: interface state UNINITIALIZED->COUNTRY_UPDATE
[ 5228.355165] wlan: uap0 START SCAN
[ 5229.422504] wlan: SCAN COMPLETED: scanned AP count=21
[ 5229.423305] wlan: uap0 START SCAN
[ 5229.860649] wlan: SCAN COMPLETED: scanned AP count=21
[ 5229.861706] wlan: uap0 START SCAN
[ 5230.298647] wlan: SCAN COMPLETED: scanned AP count=21
[ 5230.299325] wlan: uap0 START SCAN
[ 5230.736356] wlan: SCAN COMPLETED: scanned AP count=21
[ 5230.737445] wlan: uap0 START SCAN
[ 5231.174323] wlan: SCAN COMPLETED: scanned AP count=21
[ 5231.195669] wlan: uap0 Starting AP
[ 5231.212550] wlan: uap0 AP started
[ 5231.215891] wlan: HostMlme uap0 send deauth/disassoc
[ 5231.218671] Set AC=3, txop=47 cwmin=3, cwmax=7 aifs=1
[ 5231.219520] Set AC=2, txop=94 cwmin=7, cwmax=15 aifs=1
[ 5231.220238] Set AC=0, txop=0 cwmin=15, cwmax=63 aifs=3
[ 5231.221340] Set AC=1, txop=0 cwmin=15, cwmax=1023 aifs=7
```

10 Bring-up of Bluetooth LE

This section provides the steps to bring up the Bluetooth LE radio of IW610.

Step 1 – Download the combo firmware or the standalone firmware for Bluetooth LE. Refer to Section 8.1 or Section 8.3.

Step 2 – Change the Bluetooth device (BD) address.

Example of `hcitool` command to change the BD address:

```
hcitool -i hci0 cmd 3f 22 fe 06 23 22 21 43 50 00
```

Command output example:

```
< HCI Command: ogf 0x3f, ocf 0x0022, plen 8
  FE 06 23 22 21 43 50 00
> HCI Event: 0x0e plen 4
  01 22 FC 00
```

Step 3 – Reset hci0 interface.

```
hciconfig hci0 reset
```

Step 4 – Check the configuration.

```
hciconfig
```

Command output example:

```
hci0:   Type: Primary  Bus: UART
        BD Address: 00:50:43:21:22:23  ACL MTU: 256:8  SCO MTU: 0:0
        UP RUNNING
        RX bytes:951 acl:0 sco:0 events:82 errors:0
        TX bytes:573 acl:0 sco:0 commands:82 errors:0
```

11 Bring-up of 802.15.4

This section includes the steps to bring up 802.15.4 radio of IW610 module onFRDM i.MX 91 development board.

Step 1 – Download the combo firmware. Refer to Section 8.1.

Step 2 – Download the pre-built meta-matter enabled image for FRDM i.MX 91 development board ref.[1].

Step 3 – Follow the steps to set up FRDM i.MX 91 development board (Section 5.1).

Step 4 – Issue the command to flash the image to FRDM i.MX 91 development board.

```
sudo uuu -b emmc_all imx-boot-imx91frdm-sd.bin-flash_singleboot imx-image-multimedia-
imx91frdm-iwxxx-matter.rootfs.wic.zst
```

Step 5 – Transfer the ot-ctl and ot-daemon utilities available in IW610 software to FRDM i.MX 91 development board.

The path to ot-ctl and ot-daemon utilities is:

SD_WLAN_UARTSPI_NB-IW610-LNX_6_x_xx-IMX8-18.99.5.p43-18.34.5.p43-MXM6X18525_V0-GPL/OT-Tools_LNX_6_x_xx-IMX8/imx-linux-othost-070-112024-9681690/SPI/usr/bin.

- To transfer ot-ctl and ot-daemon utilities to FRDM i.MX 91 development board, follow the steps in Section 7.1.

11.1 Create a Thread network

This section shows how to create a Thread network when the firmware is loaded. Two sets of FRDM i.MX 91 development board with IW610 module are used.

Figure 14 shows the leader becoming the commissioner of the Thread network.

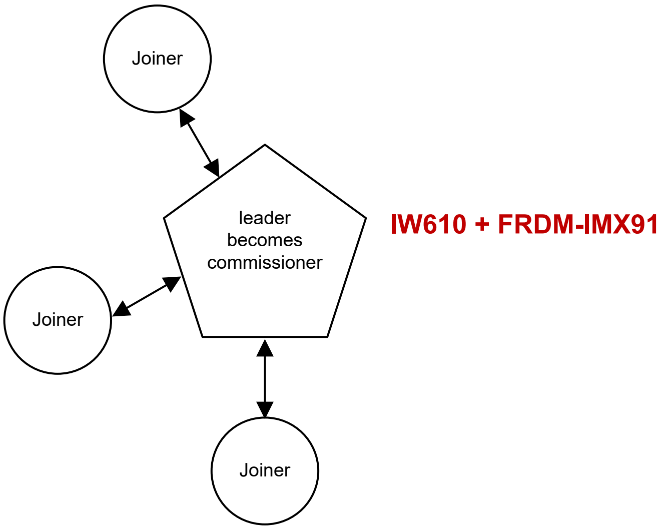


Figure 14. The leader becomes commissioner of the Thread network

Step 1 – Look for SPI interface enumeration. `/dev/gpiochip5` is the SPI interface for IW610.

```
root@ imx91frdm-iwxxx-matter:~# ls -l /dev/gpio*
```

Expected output:

```
crw------- 1 root root 254, 0 Feb 28 00:07 /dev/gpiochip0
crw------- 1 root root 254, 1 Feb 28 00:07 /dev/gpiochip1
crw------- 1 root root 254, 2 Feb 28 00:07 /dev/gpiochip2
crw------- 1 root root 254, 3 Feb 28 00:07 /dev/gpiochip3
crw------- 1 root root 254, 4 Feb 28 00:07 /dev/gpiochip4
crw------- 1 root root 254, 5 Feb 28 00:07 /dev/gpiochip5
```

Step 2 – Start ot-daemon in the background.

```
/usr/sbin/ot-daemon 'spinel+spi:///dev/spidev0.0?gpio-reset-device=/dev/gpiochip5&gpio-
int-device=/dev/gpiochip4&gpio-int-line=10&gpio-reset-line=1&spi-mode=0&spi-
speed=1000000&spi-reset-delay=0' &
```

Step 3 – Verify the ot-daemon process.

```
ps -ax | grep ot-daemon
```

Command output example:

```
560 ttyLP0    S      0:07 /usr/sbin/ot-daemon spinel+spi:///dev/spidev0.0?gpio-reset-
device=/dev/gpiochip5&gpio-int-device=/dev/gpiochip4&gpio-int-line=10&gpio-reset-
line=1&spi-mode=0&spi-speed=1000000&spi-reset-delay=0
711 ttyLP0    S+     0:00 grep ot-daemon
```

Step 4 – Open ot-ctl command line interface (CLI).

```
root@imx91frdm-iwxxx-matter:~# ot-ctl
> thread stop
Done
> ifconfig down
Done
> factoryreset
```

Step 5 – Generate the new network configuration.

```
> dataset init new
Done
```

Step 6 – Set the channel of operation.

```
> dataset channel 25
Done
```

Step 7 – Commit the dataset as active.

```
> dataset commit active
Done
```

Step 8 – Enable the interface.

```
> ifconfig up
Done
```

Step 9 – Start the Thread network and verify the state of the device. Ensure that the state of the device is leader (it may take a few seconds).

```
> thread start
Done
> state
leader
Done
> dataset active -x
0e08000000000001000035060004001fffe0020812060bf94440544b0708fd6cfb681e924a910510fdf29
435bb215e68b03b7cd77105e2f0030f4f70656e5468726561642d623636310102b6610410cd93e71a7273
1b454c4fb39d33abd5bc0c0402a0f7f80003000019
Done
```

Step 10 – Check the assigned IPv6 addresses

```
> ipaddr
fd6c:fb68:1e92:4a91:0:ff:fe00:fc00
fd6c:fb68:1e92:4a91:0:ff:fe00:8400
fd6c:fb68:1e92:4a91:4f6e:c817:320a:1574
fe80:0:0:0:3c7e:d333:fad8:818c
Done
```

Step 11 – Join the thread network from the end device (the second set of FRDM i.MX 91 development board with IW610 module).

Issue the commands below for the end device. The end device will join the thread network created in **step 9** by the leader.

```
> thread stop
Done
> ifconfig down
Done
> factoryreset
> dataset set active 0e08000000000001000035060004001fffe0020812060bf94440544b0708
fd6cfb681e924a910510fdf29435bb215e68b03b7cd77105e2f0030f4f70656e5468726561642d623
636310102b6610410cd93e71a72731b454c4fb39d33abd5bc0c0402a0f7f80003000019
Done
> dataset commit active
Done
> ifconfig up
Done
> thread start
Done
> state
child
Done
> ping fd6c:fb68:1e92:4a91:4f6e:c817:320a:157474
16 bytes from fd6c:fb68:1e92:4a91:4f6e:c817:320a:1574: icmp_seq=1 hlim=64 time=42ms
1 packets transmitted, 1 packets received. Packet loss = 0.0%. Round-trip min/avg/max =
 42/42.0/42 ms.
Done
```

12 Note about the source code in the document

The example code shown in this document has the following copyright and BSD-3-Clause license:

13 References

[1] FRDM-IMX91 Demo Images (link)
[2] GitHub – nxp-imx-support/meta-imx-frdm (link)
[3] GitHub – nxp-imx/mfg-tools/releases (link)
[4] User manual – UM11675: How to Download and Build NXP Wi-Fi Drivers (link)
[5] Webpage – Putty, SSH and telnet client for Windows (link)
[6] Webpage – Tera Term – Terminal emulator for Windows (link)
[7] Webpage – USB device driver CMD21228_Setup.exe (link)
[8] Webpage – i.MX Linux download page (link)
[9] Webpage – IW610: 1x1 Dual-band Wi-Fi 6 and Bluetooth Low Energy / 802.15.4 Solution Family (link)
[10] Webpage – Getting Started with FRDM-IMX91 Development Board (link)

14 Revision history

Table 6. Revision history

Document ID	Release date	Description
UM12298 v.1.0	3 July 2025	• Initial version

Legal information

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.

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.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

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.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

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

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

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

NXP — wordmark and logo are trademarks of NXP B.V.

Bluetooth — the Bluetooth wordmark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by NXP Semiconductors is under license.

Matter, Zigbee — are developed by the Connectivity Standards Alliance. The Alliance's Brands and all goodwill associated therewith, are the exclusive property of the Alliance.

Tables

Figures

Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.