UM11350

Getting Started with IW416 Evaluation Boards Running Linux OS

Rev. 2 — 15 June 2021 User manual

Document information

Information	Content
Keywords	RD-IW416-QFN-WIB3-1A, RD-IW416-QFN-WIB3-2A, RD-IW416-CSP-WIB3-1A, RD-IW416-CSP-WIB3-2A, IW416 reference board, WIB3 adapter board, IW416 evaluation kit, production software, firmware/driver type
Abstract	Describes the boot and jumper configurations for IW416 reference boards, WIB3 adapter board, how to power-up IW416 evaluation board and install the software

Revision history

Rev	Date	Description
v.1	20200218	Initial version
v.2	20210615	**Modifications**: • Changed the document title and structure • Section 2 "IW416 evaluation boards": added the evaluation boards for IW416 in CSP package • Section 5 "Installing the software": added the section

1 About this document

1.1 Purpose

This document explains how to set up and power up IW416 evaluation boards with a host computer running on Linux OS, how to load the drivers, and how to bring up and test the Wi-Fi and Bluetooth.

1.2 References

IW416 product summary page on NXP website (link) is the entry point to access the released documentation and the software and tools.

- Documentation tab: access to datasheet, user manuals, and application notes
- Tools & Software tab: access to the reference design packages and software releases

2 IW416 evaluation boards

2.1 IW416 evaluation kit contents

This section lists the content of each IW416 evaluation kit.

2.1.1 RD-IW416-QFN-WIB3-1A evaluation kit

- RD-IW416-QFN-WIB3-1A reference board
- WIB3 adapter board
- 5 V DC wall power adapter supply
- SDIO ribbon cable and adapter
- Dual-band dipole antenna with U.FL connector
- USB cable type B

2.1.2 RD-IW416-QFN-WIB3-2A evaluation kit

- RD-IW416-QFN-WIB3-2A reference board
- WIB3 adapter board
- 5 V DC wall power adapter supply
- SDIO ribbon cable and adapter
- Dual-band dipole antenna with U.FL connector
- USB cable type B

2.1.3 RD-IW416-CSP-WIB3-1A evaluation kit

- RD-IW416-CSP-WIB3-1A reference board
- WIB3 adapter board
- 5 V DC wall power adapter supply
- SDIO ribbon cable and adapter
- Dual-band dipole antenna with U.FL connector
- USB cable type B

2.1.4 RD-IW416-CSP-WIB3-2A evaluation kit

- RD-IW416-CSP-WIB3-2A reference board
- WIB3 adapter board
- 5 V DC wall power adapter supply
- SDIO ribbon cable and adapter
- Dual-band dipole antenna with U.FL connector
- USB cable type B

Figure 1 shows the content of IW416 evaluation kit.

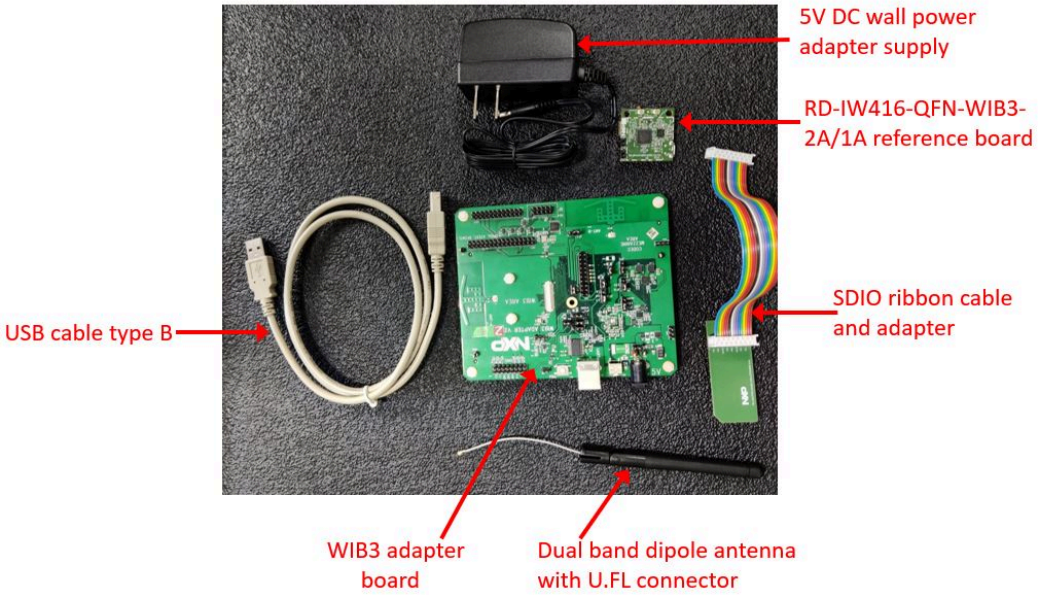


Figure 1. Content of IW416 evaluation kit

2.2 IW416 reference boards

IW416 reference boards are available with two configurations to accommodate Wi-Fi and Bluetooth either on a shared antenna or on separate antennas:

- One-antenna configuration – Wi-Fi and Bluetooth on a shared antenna with one RF switch
- Two-antenna configuration – Wi-Fi and Bluetooth on separate antennas

Both RD-IW416-QFN-WIB3-1A and RD-IW416-CSP-WIB3-1A boards use the one-antenna configuration.

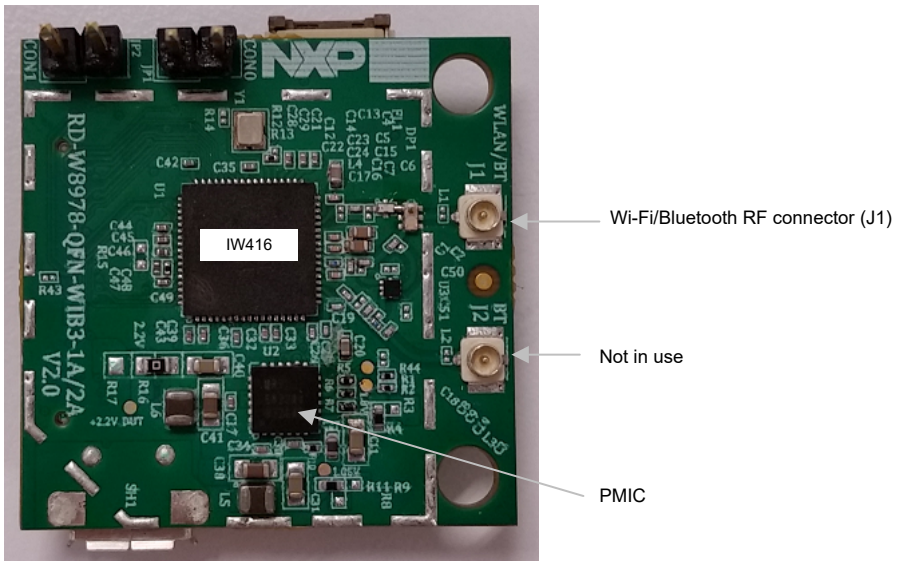


Figure 2. RD-IW416-QFN-WIB3-1A board

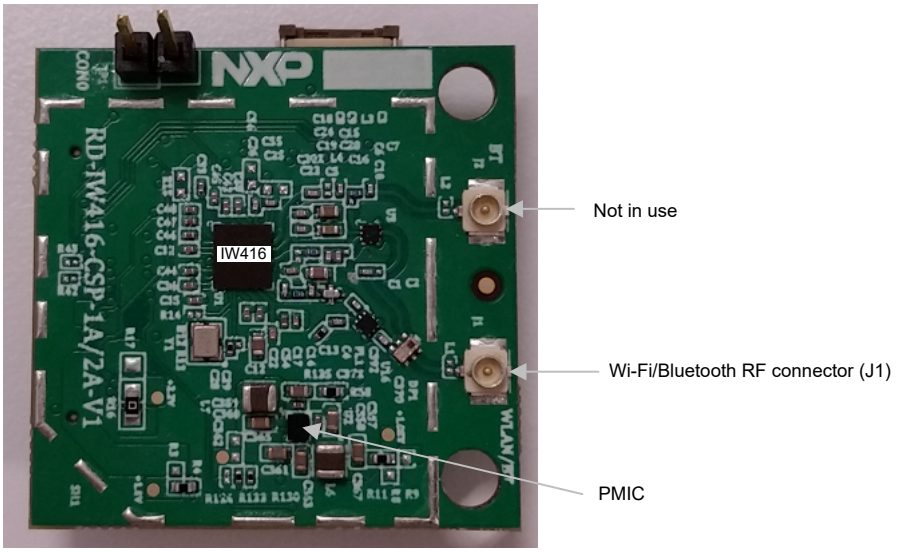


Figure 3. RD-IW416-CSP-WIB3-1A board

Both RD-IW416-QFN-WIB3-2A and RD-IW416-CSP-WIB3-2A boards use the two-antenna configuration.

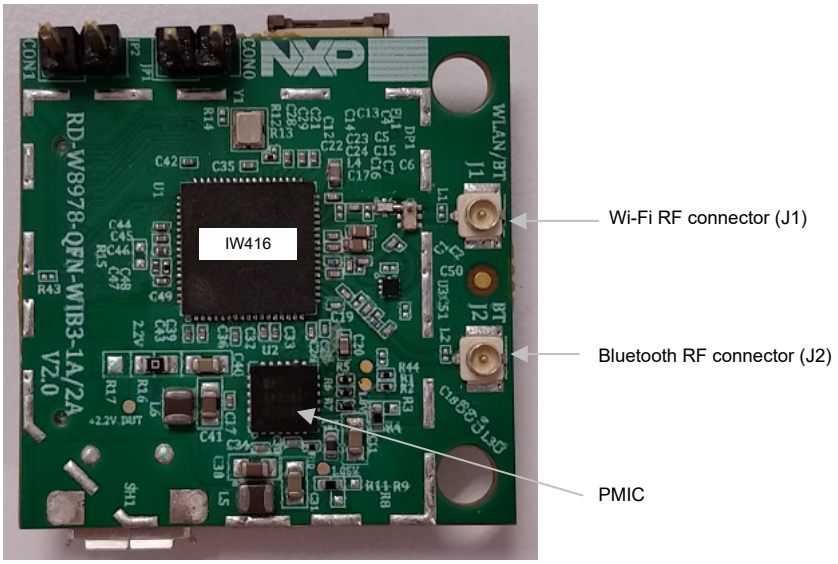


Figure 4. RD-IW416-QFN-WIB3-2A board

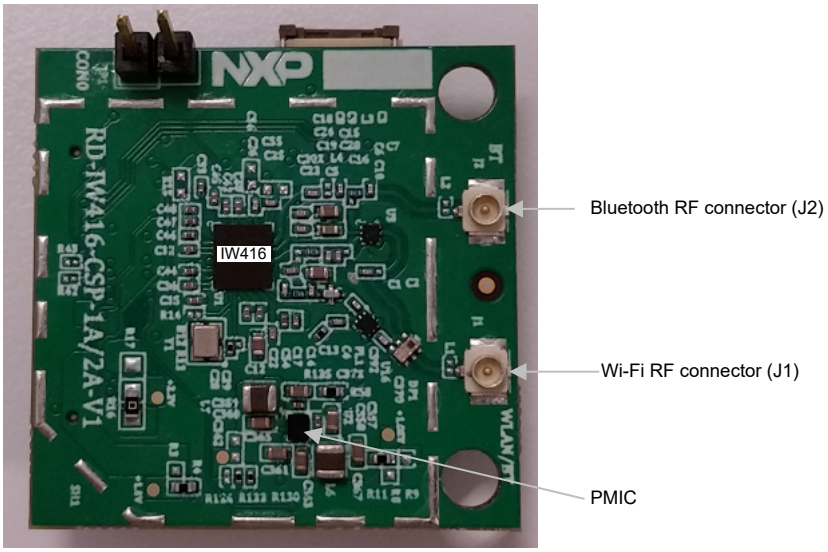


Figure 5. RD-IW416-CSP-WIB3-2A board

The two antennas need to be connected to the two separate RF connectors J1 and J2.

- J1 is for Wi-Fi
- J2 is for Bluetooth

For two-antenna designs, make sure to isolate the two antennas as much as possible to limit the interference between coexisting radios.

2.3 Boot configuration

The host interface can be configured by installing or removing specific jumpers on the evaluation board.

Table 1 shows the host interface configuration.

Note: *Refer to Section 5.6 "Configuration pins" in IW416 datasheet for details on the host configuration options and configuration pins.*

Table 1. Host interface configuration

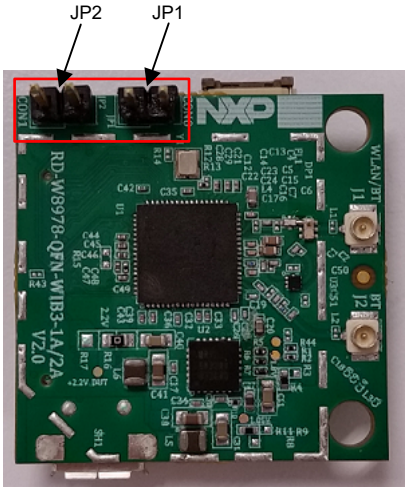
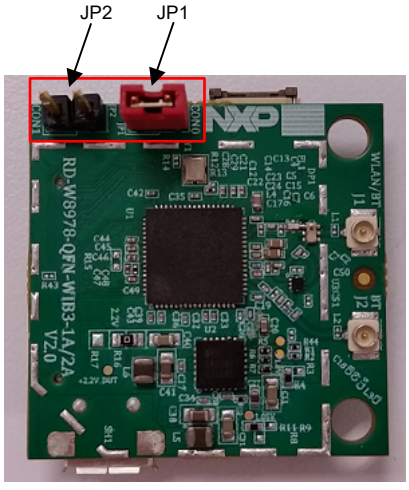
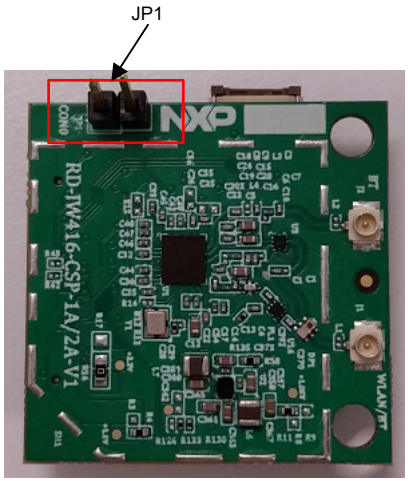
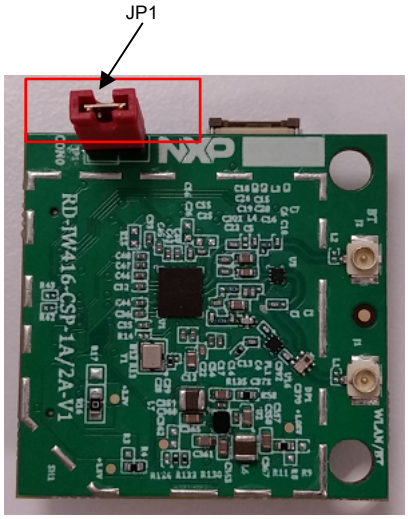
Host interface mode	Board configuration
Wi-Fi and Bluetooth on SDIO (default) QFN package	Remove both JP1 and JP2 
Wi-Fi on SDIO and Bluetooth on UART QFN package	Install JP1 and remove JP2 

Table 1. Host interface configuration...*continued*

Host interface mode	Board configuration
Wi-Fi and Bluetooth on SDIO (default) WLCSP package	Remove JP1 
Wi-Fi on SDIO and Bluetooth on UART WLCSP package	Install JP1 

2.4 Power supply for IW416 reference design boards

Figure 6 shows an overview of the power supply for IW416 reference design boards. The default jumper positions are shown in bold.

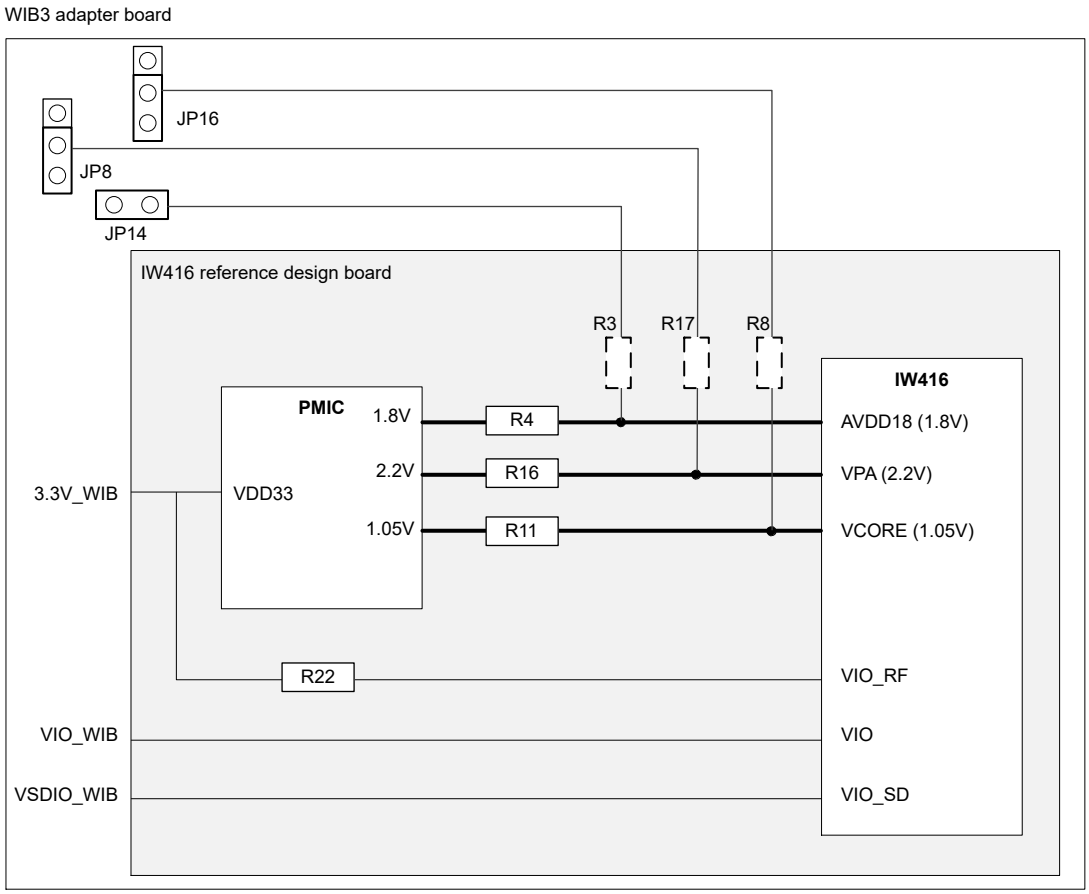


Figure 6. Power supply overview for IW416 reference design boards

2.5 WIB3 adapter board

IW416 reference design board is mounted on a WIB3 adapter board for connection to the host platform.

Figure 7 shows IW416 reference design with QFN package mounted on WIB3 adapter board.

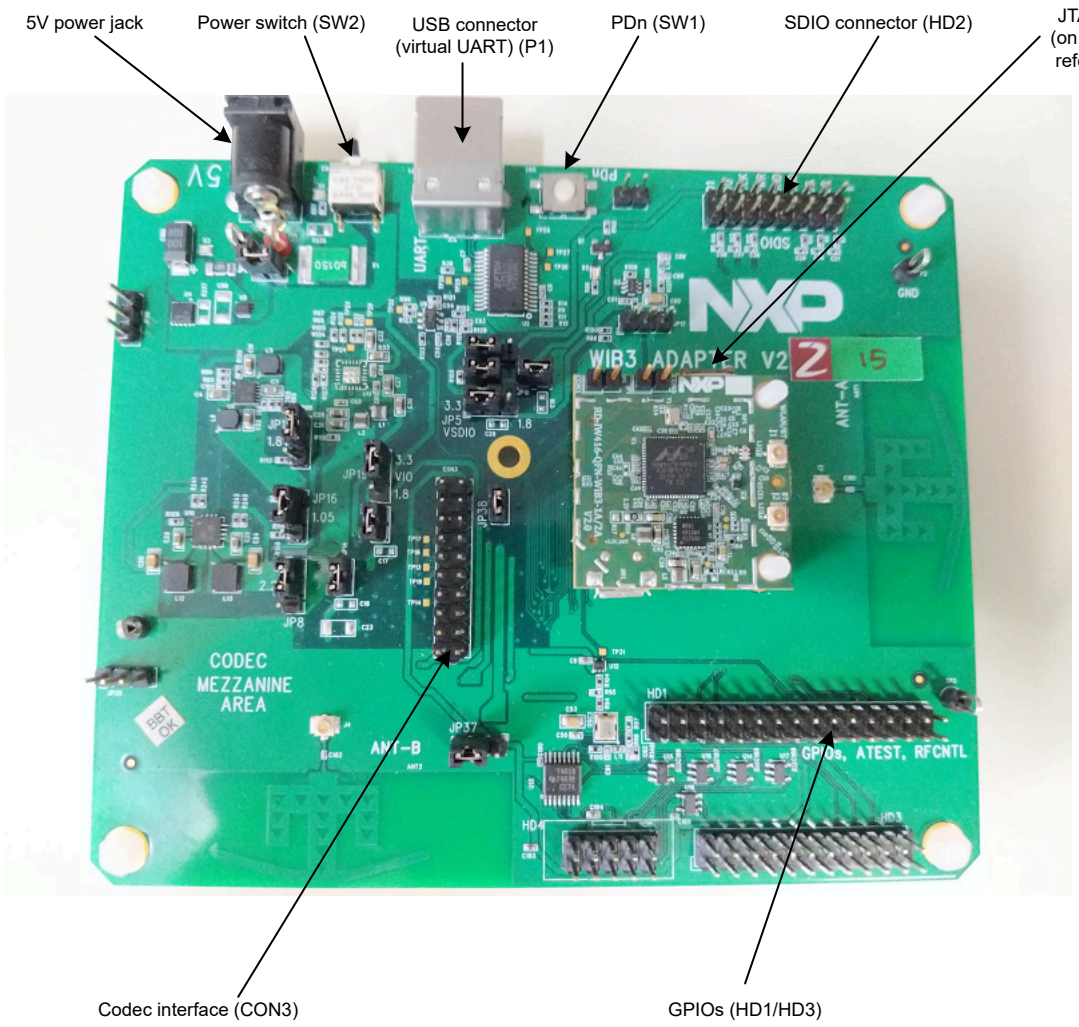


Figure 7. IW416 reference design board mounted on WIB3 adapter board - QFN package

Figure 8 shows IW416 reference design with WLCSP package mounted on WIB3 adapter board.

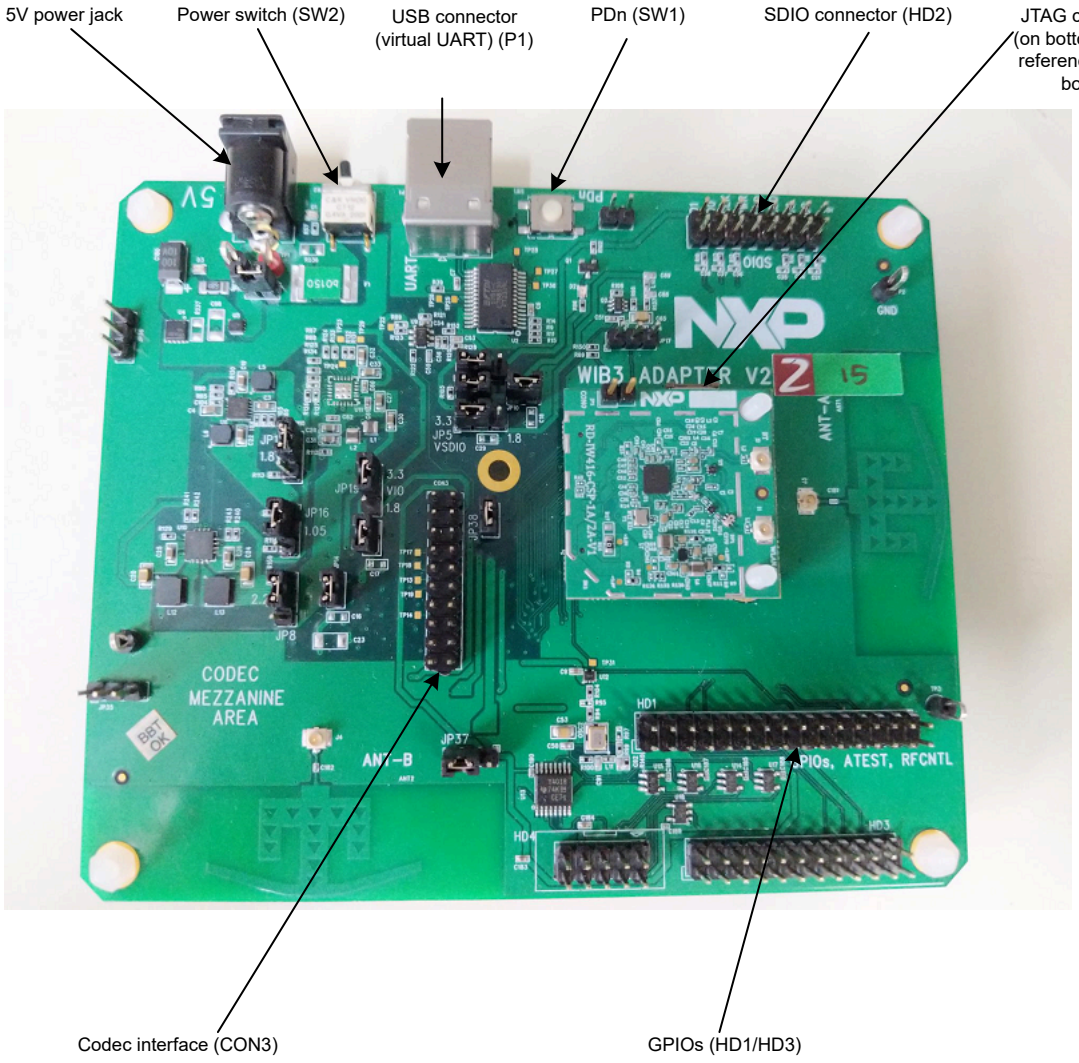


Figure 8. IW416 reference design board mounted on WIB3 adapter board - WLCSP package

2.6 Jumper configuration

Table 2 shows the default jumper configuration for the board power supply.

Table 2. Jumper configuration

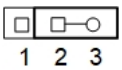
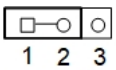
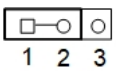
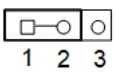
Function	Description	Header #	Default configuration	
+5V	External adapter DC 5V input	JP15	On	
3.3V	3.3V power supply output from DC-DC. Supply to 3.3V_WIB and VIO_WIB (option by JP19).	JP13	On	
3.3V_WIB	Source from 3.3V DC-DC. 3.3V supply to IW416 reference design board	JP10	On	
VIO selection	Select source for VIO_WIB supply level from on-board 3.3V DC-DC or 1.8V DC-DC: • 1-2: 3.3V • 2-3: 1.8V (default)	JP19	1.8V (default)	
VIO_WIB	VIO_WIB source based on VIO selection (JP19). VIO supply to IW416 reference design board.	JP12	On	
1.8V_WIB	External 1.8V (on-board DC-DC) supply to IW416 reference design board	JP14	On	
VSDIO selection	Select source for VSDIO_WIB from either VSDIO_HOST or on-board 3.3V DC-DC • 1-2: 3.3V (default) • 2-3: VSDIO_HOST	JP7	3.3V (default)	
VSDIO_WIB	VIO_SD (SDIO) supply to IW416 reference board • 1-2: 3.3V (default) • 2-3: 1.8V	JP5	3.3V (default)	
2.2V_WIB	Select external 2.2V from power supply A (on-board DC-DC) or power supply B (on-board PMIC).	JP8	Power supply A	

Table 2. Jumper configuration*...continued*

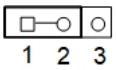
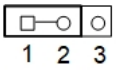
Function	Description	Header #	Default configuration	
1.8V_BRD	Select external 1.8V from power supply A (on-board DC-DC) or power supply B (on-board PMIC).	JP11	Power supply A	
1.05V_WIB	Select 1.05V from power supply A (on-board DC-DC) or power supply B (on-board PMIC)	JP16	Power supply A	

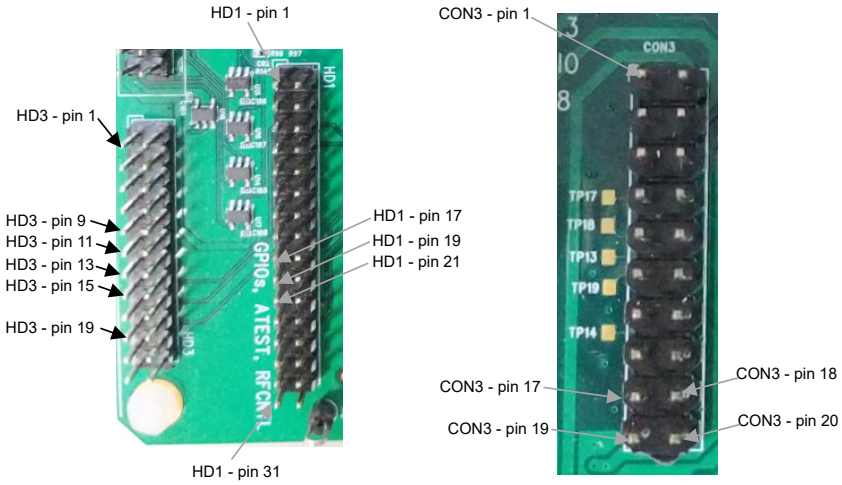
Table 3 shows the header information for the WIB3 adapter board.

Table 3. WIB3 adapter board header description

Header #	Description
J1	External 5V supply jack to power up the WIB adapter board.
P1	USB Type-B connector for virtual UART port to connect to the Bluetooth host.
HD2	SDIO interface header (need a ribbon cable to connect to the host).
J2	Connector used to connect IW416 reference design board.
JP10	Enables current measurement on 3.3V supply rail to IW416 reference board
JP12	Enables current measurement on VIO supply rail to IW416 reference board
JP15	Enables current measurement on 5V supply rail beyond the input protection circuit of J1/F1/SW2/U4/U5/D3.

Table 4 shows the pinout of the header connectors on the WIB3 adapter board.

Table 4. Header pinout

IW416 pin	Mapping on WIB adapter board header	Description
SLP_CLK_IN	HD3 - pin 21	
GPIO[15]	HD1 - pin 31	
GPIO[14]	HD1 - pin 29	
GPIO[13]	HD1 - pin 27	
GPIO[12]	HD1 - pin 25	
GPIO[11]	HD1 - pin 23	
GPIO[10]	HD1 - pin 17	
GPIO[9]	HD1 - pin 19	
GPIO[8]	HD1 - pin 21	
GPIO[7]	CON3 - pin 17	
GPIO[6]	CON3 - pin 18	
GPIO[5]	CON3 - pin 20	
GPIO[4]	CON3 - pin 19	
GPIO[3]	HD1 - pin 7	
GPIO[2]	HD1 - pin 5	
GPIO[1]	HD1 - pin 3	
GPIO[0]	HD1 - pin 1	
RF_CNTL3_P	HD3 - pin 15	
RF_CNTL2_N	HD3 - pin 13	
RF_CNTL1_P	HD3 - pin 11	
RF_CNTL0_N	HD3 - pin 9	
PDn	HD3 - pin 19	

2.7 Power supply for WIB3 adapter board

Figure 9 shows an overview of the power supply for the WIB3 adapter board. The default jumper positions are shown in bold.

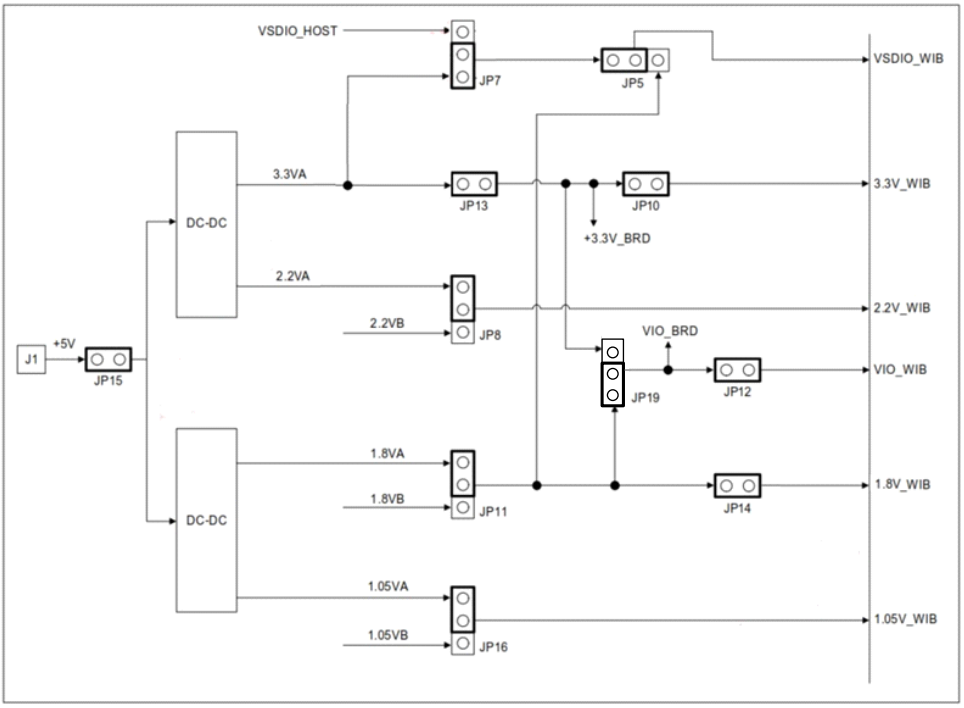


Figure 9. Power supply overview—WIB3 adapter board

3 Powering up the board

3.1 Handling recommendations

The board is sensitive to electrostatic discharge (ESD). Make sure to observe anti-static precautions when handling the boards.

3.2 Power-up sequence

Step 1 – Configure IW416 reference board

By default IW416 reference board is configured for SDIO-UART interface.

Configure IW416 reference board with the correct boot option as described in Section 2.3.

Step 2 – Mount IW416 reference board

Mount IW416 reference board onto the WIB3 adapter board.

Step 3 – Configure the WIB3 adapter board jumpers

Configure the board jumpers according to the power requirements detailed in Section 2.6. Use the default jumper setting for normal operation unless a different power supply configuration is required.

- Match IW416 SDIO voltage level with the HOST SDIO level using jumper JP7 or JP5
- Match VIO supply level with HOST IO level using jumper JP19

Step 4 – Connect power

- Connect the external 5 V DC power supply to jack J1 on the board and turn on the SW2 switch.

The next step depends on the host interface used (SDIO or UART).

4 Connecting the board to the host computer

This section shows how to connect SDIO, or UART host interface on IW416 evaluation board to the host computer.

4.1 SDIO host interface

To use SDIO as the host interface:

1. Connect SDIO ribbon cable and SDIO adapter to the WIB3 adapter board, as shown in Figure 10.
2. Insert the SDIO adapter into the host system, as shown in Figure 11.

Note:

A native SDIO interface is required on the host to use SDIO/SDIO mode. A USB-to-SDIO adapter will not work.

Figure 10 shows the SDIO interface cable connection with the red trace aligned to SD_DATA[2] (D2) signal.

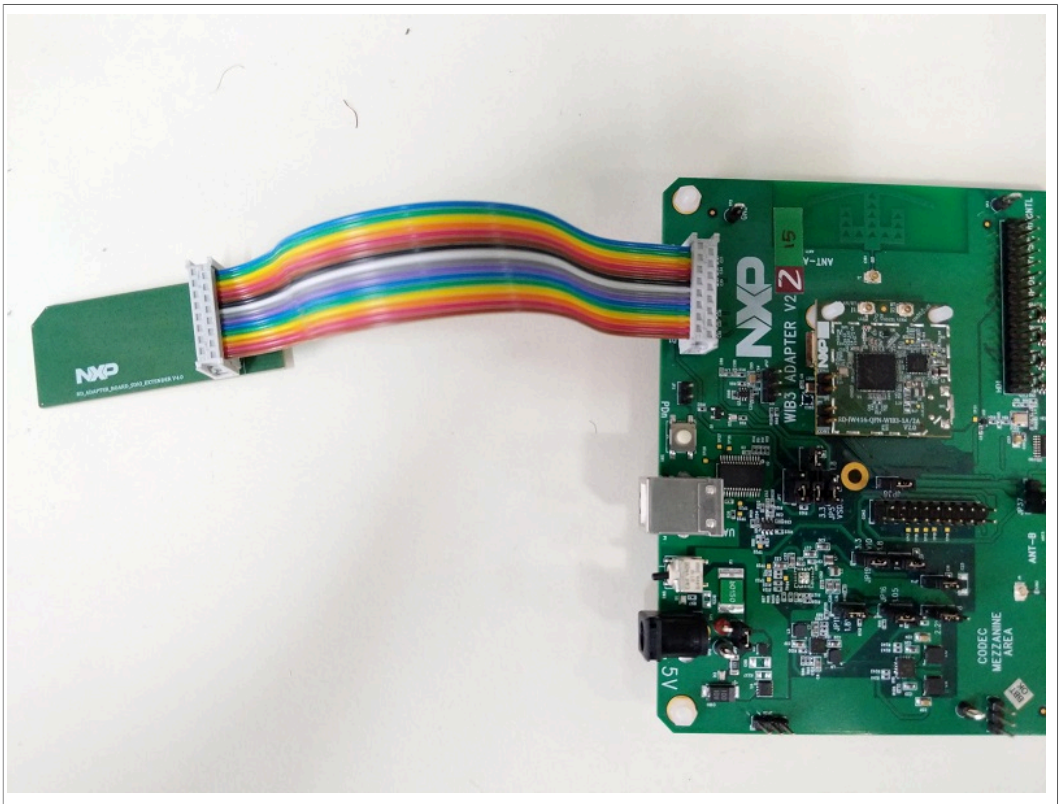


Figure 10. SDIO ribbon cable and adapter

Figure 11 shows the SDIO ribbon cable connected to the host.

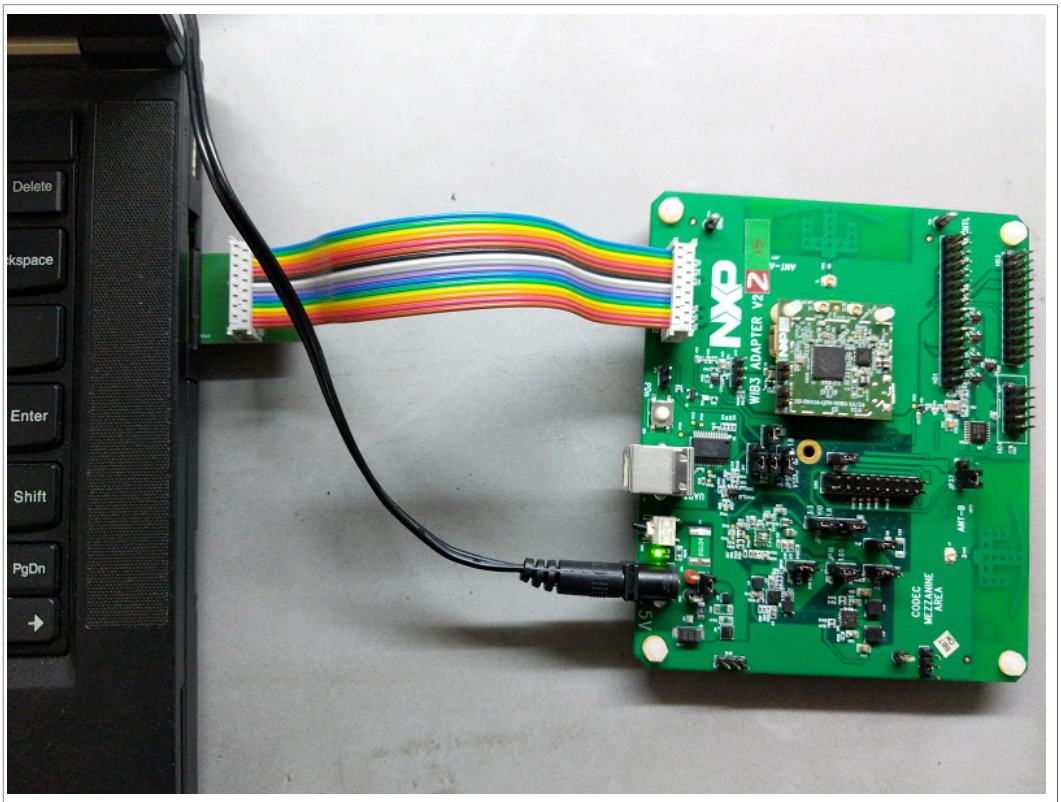


Figure 11. SDIO connection to host

4.2 UART host interface

UART may be used as the host interface for Bluetooth using a virtual UART port. To use UART as the host interface, connect a USB type B cable from the WIB3 adapter board to the host.

Figure 12 shows the connection to the host through USB (virtual UART port).

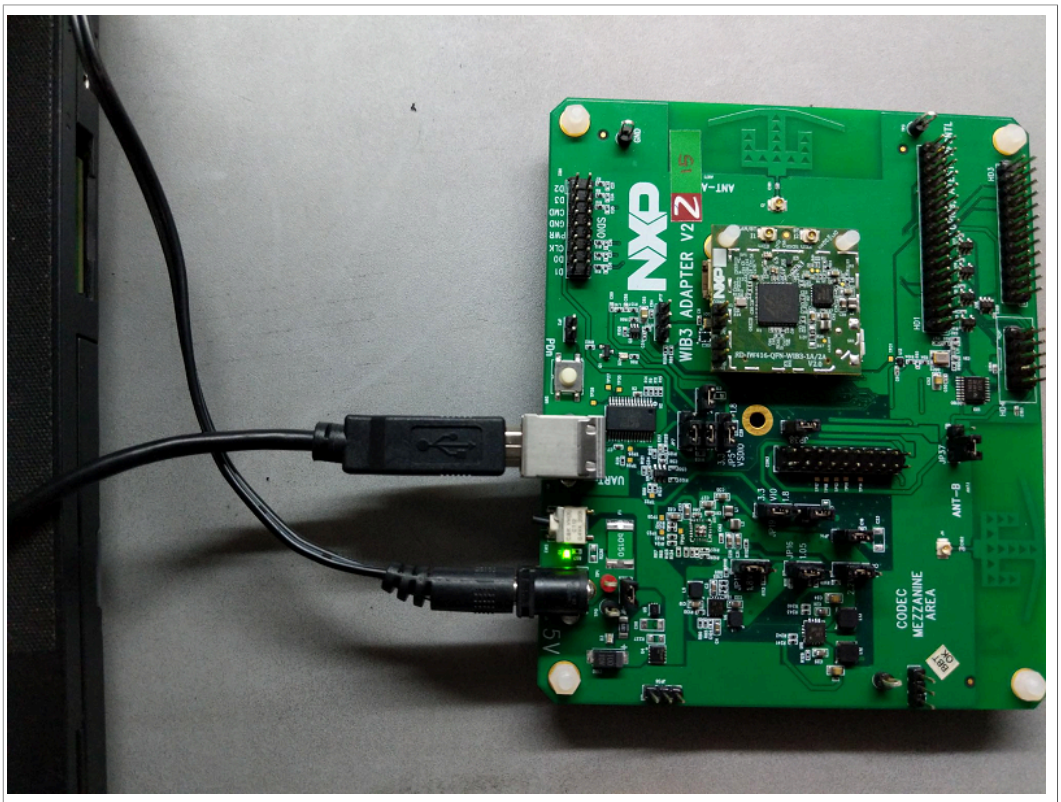


Figure 12. UART host interface (virtual UART port)

5 Installing the software

This section provides some guidance for the software download and installation, and for Wi-Fi and Bluetooth bring-up and testing. Section 5.9 "Appendix" explains how to disable Linux interface renaming, and how to install Linux Ubuntu 16.04 OS.

5.1 Conventions

This section employs the following conventions:

- Commands and examples of command outputs are shown in paragraphs with grey background color

```
This is an example of command
```

- Terms related to commands use a `monospace` font:
  - `parameter`
  - `option`
  - `command name`
- File names, directory names and paths are shown in *italics:*
  - *<file name>.<extension>*
  - *<directory>*
  - *path/to/directory/and/file*

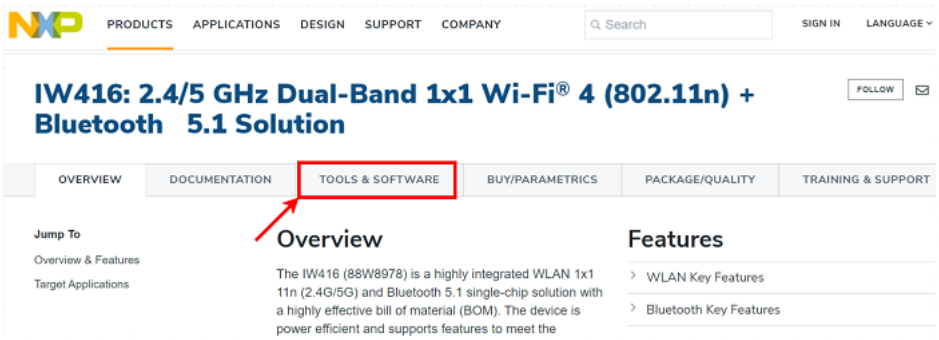
5.2 Hardware and software requirements

- PC with SDIO slot and USB port
- Ubuntu 16.04 open source operating system on Linux. See Section 5.9.2 "Install Ubuntu Linux 16.04 OS".
- Linux **root user** profile

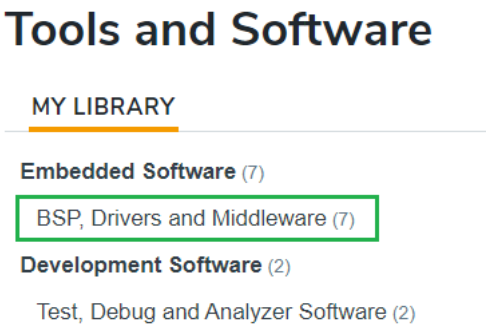
5.3 Downloading the production software

The production software release package is available for download on the TOOLS & SOFTWARE tab of the wireless device product page on NXP website.

- Go to the Wi-Fi and Bluetooth homepage on NXP website (link)
- Click on the name of the wireless device to open the product page
- Click on the TOOLS & SOFTWARE tab



- Select **MY LIBRARY**
- Select **Embedded Software > BSP, Drivers and Middleware** category



- Look for the production release software package release

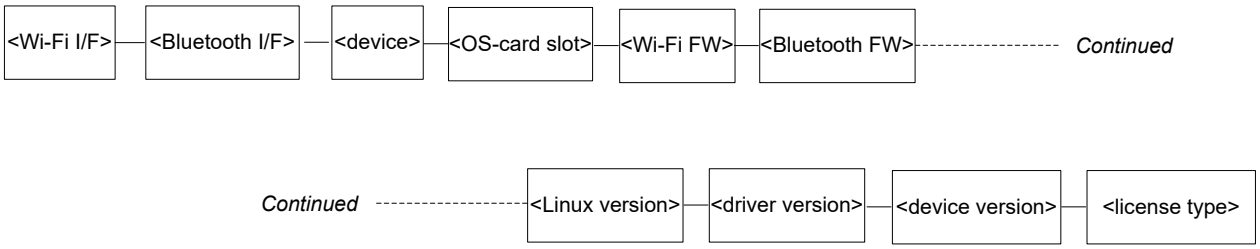
5.4 About the production software package

5.4.1 Production software package name

The production software package name includes the following information:

- Wi-Fi host interface
- Bluetooth host interface
- Wireless device part number
- OS (platform) and card slot
- Wi-Fi firmware version
- Bluetooth/Bluetooth LE firmware version
- Linux version, driver version and IW416 die version:
- License type - For example general public license (GPL), software license agreement (SLA)

Figure 13 illustrates the production software package release string.



<Wi-Fi host interface>-<Bluetooth host interface>-<device part number>-<OS name and version>-<card slot>-<Wi-Fi firmware version>-<Bluetooth firmware version>-<Linux version><driver version>_<wireless device version>-<license type>.tar

Figure 13. Production software package release string

5.4.2 Production software package content

The production software release package includes the following files:

- Software package tarball - *<production-software-release-reference>.tar*
- Release notes - *<product-software-release-reference>-ReleaseNotes.pdf*
- List of software features - *<software-release-version>-Features.pdf*

Table 5 shows an example of production software release package for IW416.

Table 5. Example of production software package release content for IW416

File name	Description
SD-WLAN-UART-BT-8978-U16-MMC-W16.68.10.p101-16.26.10.p101-C4X16679_V0-GPL.tar	Software package tarball
SD-WLAN-UART-BT-8978-U16-MMC-W16.68.10.p101-16.26.10.p101-C4X16679_V0-GPL-ReleaseNotes.pdf	Current version release notes
WLAN-BT-8978-FP68-FP26-Features.pdf	List of features

5.4.3 Combo and parallel firmware types

The software package includes two types of firmware: combo and parallel. The main features of each firmware type are listed below.

Combo firmware type

- A single firmware binary download activates both Wi-Fi and Bluetooth
- The Wi-Fi and Bluetooth functionalities are tied to a single firmware binary file

Parallel firmware type

- Separate FW binaries must be downloaded and each must be separately activated for Wi-Fi and Bluetooth through the respective interfaces
- Wi-Fi and Bluetooth functionality are tied to their respective firmware binary files
- Wi-Fi and/or Bluetooth can be reset independently without interfering with the operation of the other

Table 6 describes the firmware binary file included in IW416 production software package.

Table 6. Firmware binary files included in IW416 software package for SDIO-UART interfaces

File name	Description
sd8978_wlan_v0.bin	Wi-Fi only parallel firmware
sduart8978_combo_v0.bin	Wi-Fi and Bluetooth combo firmware
uart8978_bt_v0.bin	Bluetooth only parallel firmware

5.5 Installing the production software

5.5.1 Extract the software package release

When the download is complete, the *tar* file of the software package release is stored in the *Downloads* directory.

Create a directory at another location:

```
mkdir /path-to-location/<directoryName>
```

Move the *.tar* file to the newly created directory

```
mv <Release-String>.tar /path-to-location/<directoryName>
```

Go to the new directory

```
cd /path-to-location/<directoryName>
```

Extract the content of the software package

```
tar -xzf <file-name>
```

The software package tarball includes the *FwImage* directory and a set of *.tgz files*:

```
FwImage
<partialReleaseName>-app-src.tgz
<partialReleaseName>-GPL-src.tgz
<partialReleaseName>-mlan-src.tgz
UART-FW-LOADER-<version>-src.tgz
UART-<partialReleaseName>-GPL-src.tgz
```

Where

- The *.tgz* file names ending with *-app-src*, *GPL-src* and *mlan-src* are the Wi-Fi driver and Wi-Fi utilities
- The *.tgz* file names beginning with *UART-* are the Bluetooth driver and firmware download helper packages
- The *FwImage* directory includes the combo and parallel firmware binaries. See Section 5.4.3.

5.5.2 Extract the Wi-Fi driver and utilities

This section presents two methods to extract the Wi-Fi driver and utilities:

- The first method uses the exact file name of each *.tgz* file
- The second method looks for part of the file name of the *.tgz* file

Method 1 - Extract the *.tgz* files using the exact file name

- Go to the repository where you extracted the content of the software tarball
- Run the command to extract the content of the *.tgz* files.

```
tar -xvf <file-name>
```

Where `file-name` is the name of the Wi-Fi driver and utilities *.tgz* file:

- *<Wi-Fi-driver-release-name>-app-src.tgz*
- *<Wi-Fi-driver-release-name>-GPL-src.tgz*
- *<Wi-Fi-driver-release-name>-mlan-src.tgz*

For example, the three *.tgz* files for IW416 Wi-Fi driver and utilities have the following file names:

- *SD-UAPSTA-8978-U16-MMC-W16.68.10.p101-C4X16679_V0-app-src.tgz*
- *SD-UAPSTA-8978-U16-MMC-W16.68.10.p101-C4X16679_V0-GPL-src.tgz*
- *SD-UAPSTA-8978-U16-MMC-W16.68.10.p101-C4X16679_V0-mlan-src.tgz*

Method 2 - Extract the *.tgz* files using part of the file name

- Go to the repository where you extracted the content of the software tarball
- Run the command to extract the content of the **app-src.tgz* file.

```
tar -xvf *-WIFI-*app-src.tgz
```

- Run the command to extract the content of the **GPL-src.tgz* file.

```
tar -xvf *-WIFI-*GPL-src.tgz
```

- Run the command to extract the content of the **mlan-src.tgz* file.

```
tar -xvf *-WIFI-*mlan-src.tgz
```

Structure of the extracted content

The three *.tgz* files are extracted into one directory with a name ending with *-GPL*. The new **-GPL* directory includes *wlan_src* sub directory:

```
FwImage
      |<partialReleaseName>_wlan_<version>.bin
      |<partialReleaseName>_combo_<version>.bin
      |<partialReleaseName>_bt_<version>.bin
<new-directory-name>-GPL
       |wlan_src
<partialReleaseName>-app-src.tgz
<partialReleaseName>-GPL-src.tgz
<partialReleaseName>-mlan-src.tgz
UART-FW-LOADER-<version>-src.tgz
UART-<partialReleaseName>-GPL-src.tgz
```

wlan_src directory includes four sub directories, the modules (*.mod*), and a set of files:

```
*-GPL
     |wlan_src
        | mapp
        | mlan
        | mlinux
        | script
        gpl-2.0.txt
        Makefile
        README
        README_MLAN
        README_RBC
        README_UAP
        README_WIFIDIRECT
        mlan.mod
        moal.mod
```

5.5.3 Extract the Bluetooth driver and UART FW loader helper source

- Go to the repository where you extracted the content of the software release
- Run the command to extract the content of the *.tgz* files.

```
tar -xvf <file-name>
```

Where `file-name` is the name of the Bluetooth driver and UART firmware loader Helper *.tgz* files:

- *<Bluetooth-driver-release-name>-GPL-src.tgz*
- *<UART-firmware-loader-release-name>-src.tgz*

For example, the *.tgz* files for IW416 Bluetooth driver and UART firmware loader helper source have the following file names:

- UART-BT-8978-U16-X86-16.26.10.p101-2.2-M4X14100-GPL-src.tgz
- UART-FW-LOADER-M100-src.tgz

The *<UART-firmware-loader-release-name>-src.tgz* is extracted into *uartfwloader_src* sub directory of **-GPL* directory, and *<Bluetooth-driver-release-name>-GPL-src.tgz* is extracted into *muart_src* sub directory of **-GPL* directory:

```
FwImage
        |<partialReleaseName>_wlan_<version>.bin
        |<partialReleaseName>_combo_<version>.bin
        |<partialReleaseName>_bt_<version>.bin
<new-directory-name>-GPL
        |muart_src
        |wlan_src
        |uartfwloader_src
<partialReleaseName>-app-src.tgz
<partialReleaseName>-GPL-src.tgz
<partialReleaseName>-mlan-src.tgz
UART-FW-LOADER-<version>-src.tgz
UART-<partialReleaseName>-GPL-src.tgz
```

5.6 Wi-Fi bring-up and testing

5.6.1 Compile the Wi-Fi drivers

- Use the `cd` command to go to the directory where you extracted the software package
- Use the `cd` command to go to **-GPL/wlan_src* directory

```
cd *-GPL/wlan_src
```

- Use the `make` command to compile the Wi-Fi driver source

```
make clean; make build
```

- Check the output of the compilation by moving to the parent directory

```
cd ..
```

- Look for the *bin_wlan* directory that was created during the compilation, at the same level as *wlan_src* directory
- Use the `cd` command to go to *bin_wlan* directory

```
cd bin_wlan
```

- Check that the directory includes the two driver modules: *mlan.ko* and *<module name>.ko*

For example, the name of the driver module for IW416 is *sd8978* as the release is for SDIO Wi-Fi host interface, and 88W8978 is the former product name of IW416.

The *bin_wlan* directory also includes utilities to send commands to the Wi-Fi radio in different modes of operation such as station (STA), micro AP (uAP) (hotspot) and Wi-Fi direct (WFD). `mlanutl` utility is used to configure station mode, whereas `uaputl` utility is used to configure uAP mode. Refer to *README_MLAN* for help on `mlanutl` usage, and *README_uAP* for help on `uaputl` usage.

IW416 Wi-Fi driver directory after the compilation has the following structure:

```
| bin_wlan
        |config
        |load
        |mlan2040coex
        |mlanevent.exe
        |mlan.ko
        |mlanutl
        |README
        |README_MLAN
        |README_UAP
        |README_WIFIDIRECT
        |sd8978.ko
        |uaputl.exe
        |unload
        |wifidirect
        |wifidirectutl
        |wifidisplay
| wlan_src
        |gpl-2.0.txt
        |Makefile
        |mapp
        |mlan
        |mlan.ko
        |mlan.mod.c
        |mlan.mod.0
        |mlan.o
        |mlinux
        |modules.order
        |Module.symvers
        |README
        |README_MLAN
        |README_OPENWRT
        |README_UAP
        |README_WIFIDIRECT
        |script
        |sd8xxx.ko
        |sd8xxx.mod.c
        |sd8xxx.mod.o
        |sd8xxx.o
```

5.6.2 Copy the Wi-Fi parallel firmware to the default directory

Note: *Skip this section if you are using the combo firmware.* Section 5.6.4 *provides the guidance for the combo firmware.*

The Wi-Fi firmware needs to be copied to the default location for firmware, that is */lib/firmware/nxp* directory.

- Use the `cd` command to go to the *firmware* directory

```
cd /lib/firmware/
```

- Create *nxp* directory:

```
mkdir nxp
```

- Go to the directory where the *FwImage* directory was extracted
- Use the `cd` command to go to *FwImage*
- Use the `ls` command to list the content of the directory
- Look for the **parallel** Wi-Fi firmware file to copy to */lib/firmware/nxp* directory. The file name include **_wlan_v#.bin* where v# is the version number of the Wi-Fi parallel firmware.
- Copy the Wi-Fi parallel firmware to the default */lib/firmware/nxp* directory

```
cp <Wi-Fi parallel firmware file name.bin> /lib/firmware/nxp
```

For example, the command to copy IW416 Wi-Fi parallel firmware to */lib/firmware/nxp* directory is:

```
cp sd8978_wlan_v0.bin  /lib/firmware/nxp
```

5.6.3 Load the Wi-Fi drivers (parallel firmware)

- Go to the *bin_wlan* directory

```
cd bin_wlan/
```

- Use the `insmod` command to load the drivers. Refer to the README files in *bin_wlan* directory for details of the command arguments.

```
insmod mlan.ko
insmod <module name>.ko fw_serial=0 fw_name=nxp/<Wi-Fi parallel
 firmware file name>.bin
```

For example, the commands to load IW416 Wi-Fi drivers using the Wi-Fi parallel firmware is:

```
insmod mlan.ko
insmod sd8978.ko fw_serial=0 fw_name=nxp/sd8978_wlan_v0.bin
```

5.6.4 Copy the combo firmware to the default directory

Note: *Skip this section if you are using the parallel firmware.* Section 5.6.2 *provides the guidance for the parallel firmware.*

The combo firmware needs to be copied to the default location for firmware, that is */lib/firmware/nxp* directory.

- Use the `cd` command to go to the *firmware* directory

```
cd /lib/firmware/
```

- Create *nxp* directory:

```
mkdir nxp
```

- Go to the directory where the *FwImage* directory was extracted
- Use the `cd` command to go to *FwImage*
- Use the `ls` command to list the content of the directory
- Look for the **combo** firmware file to copy to */lib/firmware/nxp* directory. The file name include **_combo_v#.bin* where v# is the version number of the combo firmware.
- Copy the combo firmware to the default */lib/firmware/nxp* directory

```
cp <combo firmware file name.bin> /lib/firmware/nxp
```

For example, the command to copy IW416 combo firmware to */lib/firmware/nxp* directory is:

```
cp sduart8978_combo_v0.bin  /lib/firmware/nxp
```

5.6.5 Load the Wi-Fi drivers (combo firmware)

- Go to the *bin_wlan* directory

```
cd bin_wlan/
```

- Use the `insmod` command to load the drivers. Refer to the README files in *bin_wlan* directory for details of the command arguments.

```
insmod mlan.ko
insmod <module name>.ko fw_serial=1 fw_name=nxp/<combo firmware file
 name>.bin
```

For example, the commands to load IW416 Wi-Fi drivers using the combo firmware is:

```
insmod mlan.ko
insmod sd8978.ko fw_serial=1 fw_name=nxp/sduart8978_combo_v0.bin
```

5.6.6 Download the firmware over SDIO

Refer to Section 4.1 to ensure IW416 evaluation board is connected to the host computer using the SDIO ribbon cable.

The host driver automatically detects the hardware version of IW416 evaluation board (EVB) and downloads the appropriate firmware.

The `dmesg` command is used to view the progress of the Wi-Fi bring-up.

The debug message (dmesg) snippet below illustrates the Wi-Fi device registration

```
wlan: Loading MWLAN driver
wlan: Driver loaded successfully
…
Request firmware: nxp/sduart8978_combo_v0.bin
Wlan: FW download over, firmwarelen=566724 downloaded 566724
…
WLAN FW is active
wlan: version = SD8978-16.68.10.p101-C4X16C679-GPL-(FP68)
```

The bring-up of the Wi-Fi interfaces follows the completion of the firmware download.

- Use the `iwconfig` command to view the details of the Wi-Fi interfaces.

Output of `iwconfig` command:

```
uap0     IEEE 802.11-DS  ESSID:""
        Mode:Master  Frequency:2.437 GHz  Access Point: Not-Associated

        Encryption key:off
        Link Quality:0  Signal level:0  Noise level:0
        Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
        Tx excessive retries:0  Invalid misc:0   Missed beacon:0

mlan0    IEEE 802.11-DS  ESSID:""
        Mode:Managed  Access Point: Not-Associated   Bit Rate:1 Mb/s

        Tx-Power=24 dBm
        Retry limit:9   RTS thr=2347 B   Fragment thr=2346 B
        Encryption key:off
        Power Management:on
        Link Quality=0/5  Signal level=0 dBm  Noise level=0 dBm
        Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
        Tx excessive retries:0  Invalid misc:0   Missed beacon:0

wfd0     IEEE 802.11-DS  ESSID:""
        Mode:Managed  Access Point: Not-Associated   Bit Rate:1 Mb/s

        Tx-Power=24 dBm
        Retry limit:9   RTS thr=2347 B   Fragment thr=2346 B
        Encryption key:off
        Power Management:on
        Link Quality=0/5  Signal level=0 dBm  Noise level=0 dBm
        Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
        Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

Table 7 "Interface types and names" provides the details of the interfaces.

Table 7. Interface types and names

Interface type	Interface name
STA	mlan0
uAP	uap0
P2P	wfd0

5.6.7 Test the Wi-Fi operation

This section shows how to test the Wi-Fi STA and uAP interfaces.

Wi-Fi can be started in the following modes:

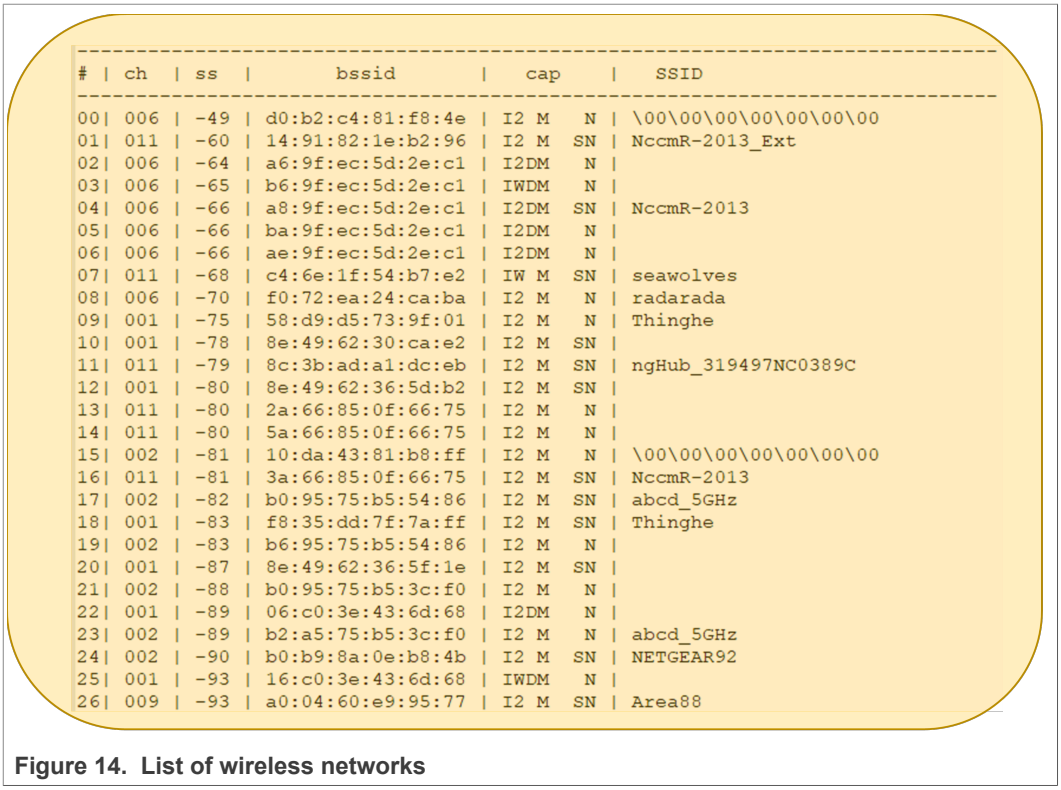
- Station (STA) mode: In this mode, the device under test (IW416) is started as a station. Following the scan for available wireless networks, the device connects to an Access Point.
- Micro access point (uAP) mode: In this mode, the device under test (IW416) is started as an Access Point.

STA bring-up

- Use the `mlanutl` utility and `setuserscan` command to scan for the wireless networks
- Refer to the *README_MLAN* available in the *wlan_src* directory for details on the parameters

```
./mlanutl mlan0 setuserscan
```

After 5 seconds, the command prints a list of available wireless networks and their details.



```
-------------------------------------------------------------------------------
# | ch  | ss  |       bssid       |   cap    |   SSID
-------------------------------------------------------------------------------
00| 006 | -49 | d0:b2:c4:81:f8:4e | I2 M   N | \00\00\00\00\00\00
01| 011 | -60 | 14:91:82:1e:b2:96 | I2 M  SN | NccmR-2013_Ext
02| 006 | -64 | a6:9f:ec:5d:2e:c1 | I2DM   N |
03| 006 | -65 | b6:9f:ec:5d:2e:c1 | IWDM   N |
04| 006 | -66 | a8:9f:ec:5d:2e:c1 | I2DM  SN | NccmR-2013
05| 006 | -66 | ba:9f:ec:5d:2e:c1 | I2DM   N |
06| 006 | -66 | ae:9f:ec:5d:2e:c1 | I2DM   N |
07| 011 | -68 | c4:6e:1f:54:b7:e2 | IW M  SN | seawolves
08| 006 | -70 | f0:72:ea:24:ca:ba | I2 M   N | radarada
09| 001 | -75 | 58:d9:d5:73:9f:01 | I2 M   N | Thinghe
10| 001 | -78 | 8e:49:62:30:ca:e2 | I2 M  SN |
11| 011 | -79 | 8c:3b:ad:a1:dc:eb | I2 M  SN | ngHub_319497NC0389C
12| 001 | -80 | 8e:49:62:36:5d:b2 | I2 M  SN |
13| 011 | -80 | 2a:66:85:0f:66:75 | I2 M   N |
14| 011 | -80 | 5a:66:85:0f:66:75 | I2 M   N |
15| 002 | -81 | 10:da:43:81:b8:ff | I2 M   N | \00\00\00\00\00\00
16| 011 | -81 | 3a:66:85:0f:66:75 | I2 M  SN | NccmR-2013
17| 002 | -82 | b0:95:75:b5:54:86 | I2 M  SN | abcd_5GHz
18| 001 | -83 | f8:35:dd:7f:7a:ff | I2 M  SN | Thinghe
19| 002 | -83 | b6:95:75:b5:54:86 | I2 M   N |
20| 001 | -87 | 8e:49:62:36:5f:1e | I2 M  SN |
21| 002 | -88 | b0:95:75:b5:3c:f0 | I2 M   N |
22| 001 | -89 | 06:c0:3e:43:6d:68 | I2DM   N |
23| 002 | -89 | b2:a5:75:b5:3c:f0 | I2 M   N | abcd_5GHz
24| 002 | -90 | b0:b9:8a:0e:b8:4b | I2 M  SN | NETGEAR92
25| 001 | -93 | 16:c0:3e:43:6d:68 | IWDM   N |
26| 009 | -93 | a0:04:60:e9:95:77 | I2 M  SN | Area88
```

Figure 14. List of wireless networks

Table 8 "Parameters of wireless networks" provides the meaning of the heading in the above figure.

Table 8. Parameters of wireless networks

Heading value	Definition
#	serial number
Ch	wireless channel
SS	signal strength
BSSID	basic service set ID - The MAC address of the wireless network
Cap	capabilities of the wireless network
SSID	service set ID - The name of the wireless network

- Command to connect to the secured network:

```
./mlanutl mlan0 passphrase "1;ssid=<SSID>;passphrase=<Passphrase>"
iwconfig mlan0 essid <SSID>
```

- Command to connect to the open network:

```
iwconfig mlan0 essid <SSID>
```

- Acquire the IP address using `dhclient` command:

```
dhclient mlan0 -v
```

- Confirm the communication link using `ping` command:

```
ping <IP-Address-of-AP>
```

The ping outcome confirms that the STA is connected to an external Access Point and working:

```
Pinging 192.168.29.1 with 32 bytes of data:
Reply from 192.168.29.1: bytes=32 time=2ms TTL=64
Reply from 192.168.29.1: bytes=32 time=4ms TTL=64
Reply from 192.168.29.1: bytes=32 time=4ms TTL=64
Reply from 192.168.29.1: bytes=32 time=2ms TTL=64

Ping statistics for 192.168.29.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

uAP bring-up

The uAP can be started in the following modes:

- Open mode: *uaputl* utility is used to start the uAP in open mode.
- WPA2 mode: *hostapd* utility is used to start the uAP in WPA2 mode. The configuration file (**.conf*) is used to pass the uAP configuration data such as passphrase, and ssid to *hostapd* utility.
- Run the command to start the uAP in open mode:

```
./uaputl.exe bss_start
```

Check the command output:

```
BSS start successful!
```

- Use `iwconfig` command to view the information about the AP:

```
iwconfig uap0
```

- Connect an external station to this uAP, and run the command to retrieve the list of connected stations:

```
./uaputl.exe sta_list
```

The command output confirms that uAP is working correctly:

```
Number of STA = 1
STA 1 information:
=====================
MAC Address: 38:e6:0a:c6:1a:ec
Power mfg status: power save
Mode: 2.4G_11n,
Rssi : -77 dBm
```

Stopping the AP

The command to stop the AP is as follows:

```
./uaputl.exe bss_stop
```

Starting the uAP in wpa2 mode

- Use the `cat` command to create the configuration file:

```
cat hostapd_wpa2.conf
```

- Edit *hostapd_wpa2.conf* file content:

```
interface=uap0
hw_mode=g
channel=6
country_code=US
ssid=AP_TEST
ieee80211n=1
auth_algs=1
wpa=2
wpa_key_mgmt=WPA-PSK
rsn_pairwise=CCMP
wpa_passphrase=123456789
```

- Use `hostapd` command to start the AP in wpa2 mode:

```
hostapd hostapd_wpa2.conf
```

Command output example:

```
Configuration file: hostapd_wpa2.conf
uap0: interface state UNINITIALIZED->COUNTRY_UPDATE
uap0: INTERFACE-DISABLED
Using interface uap0 with hwaddr 70:66:55:e5:51:6d and ssid "AP_TEST"
uap0: interface state COUNTRY_UPDATE->ENABLED
uap0: AP-ENABLED
uap0: INTERFACE-ENABLED
uap0: STA 38:e6:0a:c6:1a:ec IEEE 802.11: associated
uap0: AP-STA-CONNECTED 38:e6:0a:c6:1a:ec
uap0: STA 38:e6:0a:c6:1a:ec RADIUS: starting accounting session
 600AA545-00000003
uap0: STA 38:e6:0a:c6:1a:ec WPA: pairwise key handshake completed
 (RSN)
```

5.7 Bluetooth bring-up and testing

A common use-case for Bluetooth is the streaming of digital audio using the advanced audio distribution profiles (A2DP) universal protocol. Every A2DP connection has a source device (SRC) and a sink device (SNK). SRC device transmits local digital audio files to SNK device.

This section demonstrates IW416 EVB capability as A2DP SNK where a smart phone with digital music shall acts as A2DP SRC.

5.7.1 Hardware requirement

- A2DP source device, for example a device with audio streaming over Bluetooth capability such as a smart phone

5.7.2 Compile the Bluetooth drivers

- Use the `cd` command to go to the directory where you extracted the software package
- Use the `cd` command to go to **-GPL/muart_src* directory

```
cd *-GPL/muart_src
```

- Use the `make` command to compile the Bluetooth driver source

```
make clean; make build
```

- Check that there are no errors in the output of the compilation
- Use the `ls` command to list the content of the directory
- Look for the *.ko* driver file type, for example *hci_uart.ko* in muart_src directory

5.7.3 Load the Bluetooth drivers (parallel firmware)

Note: *Section 5.7.5 provides the guidance for the combo firmware.*

- Go to the **-GPL* directory where you compiled the Bluetooth drivers

```
cd *-GPL/
```

- Use the `cd` command to go to *uartfwloader/linux* directory

```
cd uartfwloader/linux/
```

- Use the `make` command to compile the source:

```
make clean; make make
TARGET=W8978
```

The *fw_loader_W8978* executable is created in the *linux* directory. It will be used to download the Bluetooth parallel firmware over UART to the development board.

- Use the `dmesg` command to get the list of connected USB devices

```
dmesg
```

Refer to Section 4.2 "UART host interface" to connect the board to the host computer using a virtual USB port.

- Identify the UART/USB port to which IW416 evaluation board is connected.

Figure 15 shows an example of `dmesg` command output.

```
[1216296.702777] usb 3-1: new full-speed USB device number 6
using xhci_hcd
[1216296.849026] usb 3-1: New USB device found, idVendor=0403,
idProduct=6001
...
...
...
[1216296.852022] usb 3-1: Detected FT232RL
[1216296.852419] usb 3-1: FTDI USB Serial Device converter now
attached to ttyUSB0
```

Figure 15. Identification of connected USB devices

- Use the `ls` command in *dev* directory to confirm that ttyUSB0 is active.

```
ls /dev/ttyUSB0
```

5.7.4 Download the Bluetooth firmware over UART (parallel firmware)

The *fw_loader* executable is used to download the Bluetooth parallel firmware over UART.

- Use the `cd` command to go to the directory with the *fw_loader* executable:

```
cd *-GPL/uartfwloader/linux/
```

- Use the `--help` command for the detailed list of parameters to indicate in the command to download the firmware.

```
./fw_loader_W8978 --help
```

The output of the `--help` command is shown below:

```
fw_loader_W8978 <ComPort> <BaudRate> <FlowControl> <FileName>
```

The following is the example of command to download the Bluetooth parallel firmware over UART for IW416:

```
./fw_loader_W8978 /dev/ttyUSB1 115200 0 ../../../FwImage/
uart8978_bt_v0.bin 3000000
```

The following is an example of command output:

```
Protocol: NXP Proprietary
FW Loader Version: M305
ComPort : /dev/ttyUSB1
BaudRate: 115200
FlowControl: 0
Filename: ../../../FwImage/uart8978_bt_v0.bin
Second BaudRate: 3000000

ChipID is : 7201, Version is : 0
File downloaded:   137956:  137956
Download Complete
time:3933
```

5.7.5 Load the Bluetooth drivers (combo firmware)

Note: Section 5.7.3 *provides the guidance for the parallel firmware.*

- Go to *bin_muart* directory

```
cd *-GPL/bin_muart/
```

- Use the `insmod` command to load the drivers. Refer to the README file in *bin_muart* directory for details of the command arguments.

```
insmod hci_uart.ko
```

Command to load IW416 Bluetooth drivers using the combo firmware:

```
insmod hci_uart.ko fw_serial=1 fw_name=nxp/sduart8978_combo_v0.bin
```

At this point, the Bluetooth combo firmware gets activated.

5.7.6 Test Bluetooth A2DP procedure

Attach the Bluetooth UART host interface on the evaluation board to the virtual USB port on the host computer

The Bluetooth UART host interface is available on a hci interface, most often hci0.

- Use the `cd /` command to be at the root of the Linux host computer
- Use `hciattach` command to attach the hci interface of the evaluation board to */dev/ttyUSB0* socket:

```
hciattach /dev/ttyUSB0 any 3000000 flow
```

Wait for the command output:

```
Device setup complete
```

- Use the `cd /` command to move to the root of the Linux host computer
- Use the `hciconfig` command to print the information on the Bluetooth interface:

```
hciconfig -all
```

The following is an example of command output:

```
hci0: Type: BR/EDR  Bus: UART
BD Address: 20:4E:F6:25:F3:18  ACL MTU: 1021:7  SCO MTU: 60:12
UP RUNNING
RX bytes:551 acl:0 sco:0 events:54 errors:0
TX bytes:2763 acl:0 sco:0 commands:54 errors:0
```

- If the print out shows a few hci interfaces, run the command below to bring down the hci interfaces that could interfere with the A2DP Bluetooth operation.

```
hciconfig <hci-number> down
```

Start `dbus daemon`

- Run the command to start `dbus daemon` in the background:

```
./usr/local/bluez/dbus-1.8.6/bin/dbus-daemon --system          --
nopidfile
```

- Stop `bluetoothd` in case it is already running

```
kill -9 <process_id>
```

Start `bluetoothd` in the foreground

- Run the command to start `bluetoothd`:

```
/usr/local/bluez/bluez-tools/libexec/bluetooth/bluetoothd    -ndE --
compat
```

Open a new terminal and start `pulseaudio`

- Run the command to start `pulseaudio`:

```
#/usr/local/bluez/pulseaudio-8.0_for_bluez-5.42/bin/pulseaudio -vvv
```

Open another terminal and start `bluetoothctl`

- Run the command to start `bluetoothctl` to interact with the Bluetooth daemon:

```
#/usr/local/bluez/bluez-tools/bin/bluetoothctl
```

Command output example:

```
[NEW] Controller 20:4E:F6:25:F3:18 BlueZ 5.44 [default]
[bluetooth]#
```

Authenticate

- Register with an authentication agent that will handle the PIN prompt:

```
[bluetooth]# agent on
Agent registered
[bluetooth]# default-agent
Default agent request successful
```

Enable the discoverable mode

- Enable the discoverable mode so a peer Bluetooth device can scan and connect to this device:

```
[bluetooth]# discoverable on
Changing discoverable on succeeded
```

Initiate the connection from a peer Bluetooth device such as a smart phone.

The following shows an example of Bluetooth device pairing:

```
[CHG] Controller 20:4E:F6:25:F3:18 Discoverable: 0x01
[NEW] Device B4:F5:00:31:CB:4E Moto E
Request confirmation
[agent] Confirm passkey 888367 (yes/no): yes
[CHG] Device B4:F5:00:31:CB:4E Modalias: bluetooth:v000Fp1200d1436
...
[CHG] Device B4:F5:00:31:CB:4E ServicesResolved: yes
[CHG] Device B4:F5:00:31:CB:4E Paired: yes
```

Authorize A2DP service

- Enter `yes` when asked to authorize the A2DP service

```
Authorize service
[agent] Authorize service 0000110d-0000-1000-8000-00805f9b34fb (yes/
no): yes
...
```

Play some music on the paired smart phone

The music played on the paired smart phone should be routed to the speaker of the Linux machine.

5.8 MFG software bring-up

If you are using the evaluation board for RF testing, download the MFG software package from NXP website. Refer to IW416 Labtool User Guide (UM11434) to set up the test environment and start the testing.

5.9 Appendix

5.9.1 Disable Linux interface renaming

If Linux OS renames the interfaces, run the following to disable the renaming:

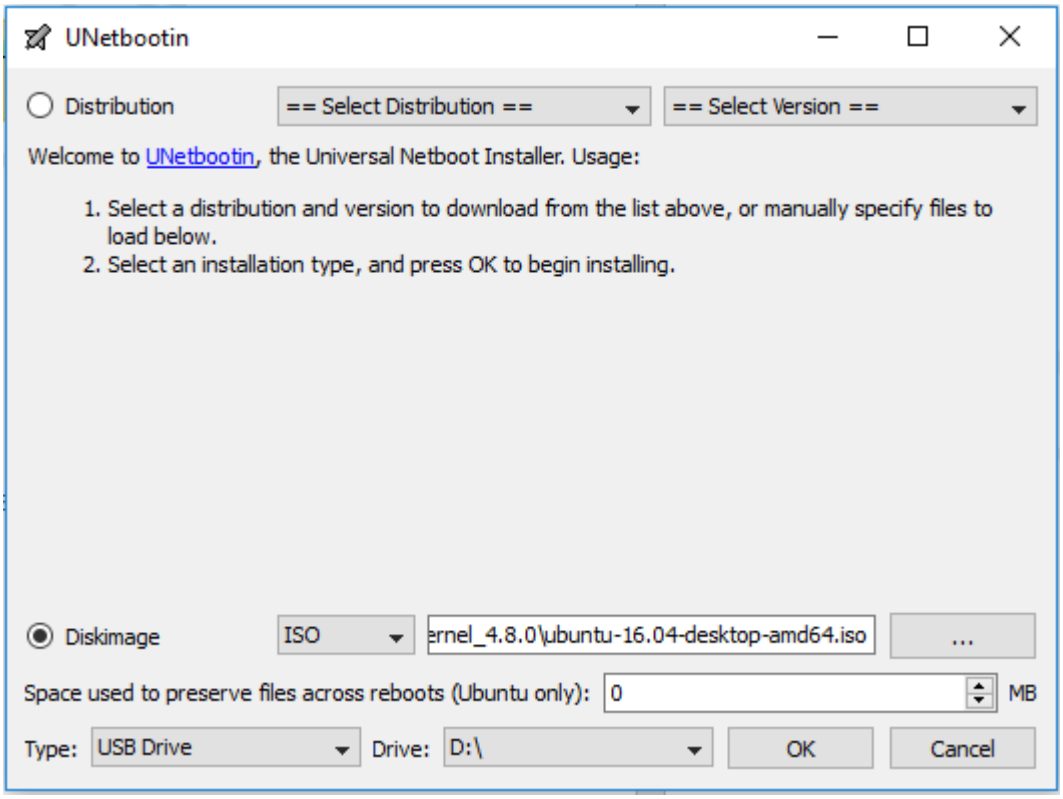
```
vi /etc/default/grub
```

Change the line `GRUB_CMDLINE_LINUX=""` into
`GRUB_CMDLINE_LINUX="net.ifnames=0 biosdevname=0"`

```
root# update-grub
root# reboot
```

5.9.2 Install Ubuntu Linux 16.04 OS

- Using another computer, **download** *Ubuntu 16.04 Long Term Support (LTS)* version from Ubuntu website https://releases.ubuntu.com/16.04/. Choose the OS compatible with your computer – 64-bit/32-bit. The recommendation is 64-bit.
- **Save** the ISO file in the computer and use tools such as `UNetBootIn` to create an installation USB. `UNetBootIn` can be downloaded from http://unetbootin.github.io/
- **Insert** an empty USB Flash drive with a minimum size of 4 GB. **Select** the previously downloaded *.iso* file in the *Disk Image Field*. **Click OK** and wait for the Installable USB to be created.



- **Plug in** the USB install disk into the computer on which Ubuntu 16.04 is to be installed
- **Power on** the computer.
- During boot, **enter** the *BIOS Setup Menu*. **Select** the USB drive as the primary Boot device. **Save** the settings and **exit** *BIOS Setup Menu*.
- Laptop should now boot from the installable USB.
- **Follow** the Ubuntu installation steps.
- **Create** a *root* password.
- **Reboot** the computer.
- **Open** a terminal.
- **Execute** `sudo su` command to enter into root mode. Enter root password.
- All the commands mentioned in the following sections are to be executed as root user.

6 Acronyms and abbreviations

Table 9. Acronyms and abbreviations

Acronym	Definition
A2DP	Advanced audio distribution profiles
AC	Alternating current
BT	Bluetooth
DC	Digital current
DUT	Device under test
ESD	Electrostatic discharge
EVB	Evaluation board
HCI	Human-computer interaction
MFG	Manufacturing
RD	Reference design
SDIO	Secure digital input output
UART	Universal asynchronous receiver transmitter
WIB	Wireless interposer board

7 Legal information

7.1 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.

7.2 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.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. 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.

7.3 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.

Tables

Figures

Contents

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

Date of release: 15 June 2021
Document identifier: UM11350