Sona NX611 Evaluation Package Yocto Integration

Sona NX611 Series

Application Note *v0.1*

Introduction

This guide covers hardware and software setup for implementing our Sona NX611 Series Wi-Fi 6 module in Yocto Linux. The Sona NX611 is offered in three form factors: the SIP module, the M.2 1216 module and the M.2 2230 module. Since the M.2 2230 is a pluggable edge-connector module, it's easier to integrate for evaluation, and is recommended. This guide assumes integration of the M.2 2230 module.

Table 1: Sona NX611 variants

Form Factor	Wi-Fi	BT	Part Number
SIP	*SDIO*	*UART*	*453-00155 (2 RF trace pins)* *453-00156 (1 RF trace pin)*
M.2 2230	*SDIO*	*UART*	*453-00165 (1 RF connector)* *453-00166 (2 RF connectors)*
M.2 1216	*SDIO*	*UART*	*453-00157 (2 RF connectors)* *453-00158 (1 RF connector)* *453-00160 (Chip antenna)*

Requirement

- iMX8MP EVK (or any platform which has an M.2 interface with SDIO)
- Sona NX611 SDIO M.2 2230 card (either 453-00165 or 453-00166)
- Sona NX611 evaluation package (Please contact Ezurio FAE to get this evaluation package)
- Development PC with Yocto build environment
- Terminal software for console message dump
- AP/Router which is able to support 2.4G/5GHz band

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852-2762-4823

Yocto Integration: Build Environment Setup

1. Modify the bblayers.conf under imx-yacto-bsp/build/conf as below:

```
BBLAYERS = " \
  ${BSPDIR}/sources/poky/meta \
  ${BSPDIR}/sources/poky/meta-yocto \
  \
  ${BSPDIR}/sources/meta-openembedded/meta-oe \
  ${BSPDIR}/sources/meta-openembedded/meta-multimedia \
  \
  ${BSPDIR}/sources/meta-fsl-arm \
  ${BSPDIR}/sources/meta-fsl-arm-extra \
  ${BSPDIR}/sources/meta-fsl-demos \
  ${BSPDIR}/sources/meta-summit-radio-eng/meta-summit-radio \
```

2. Change kernel config and rebuild kernel:

```
cd ~/imx-yacto-bsp/
source setup-environment build/
bitbake -c menuconfig virtual/kernel
```

Kernel Config Modification:

Wi-Fi:

Set cfg80211 to 'm' instead of build-in.

```
Networking support --> Wireless
<m> cfg80211
```

Disable the firmware sysfs fallback mechanism.

```
Device drivers --> Generic Driver options --> Firmware loader -->
[ ] Enable the firmware sysfs fallback mechanism
```

BT:

Disable networking support for Bluetooth:

```
Networking support -->
 < > Bluetooth
```

In iMX8M platform, to integrate BT via UART, you may need to set "i.MX SDMA support" to M as kernel module instead of built-in. In newer i.MX BSP, the built-in module may be initialized before root file system loading and cause SDMA initialization failure.

```
Device drivers --> DMA Engine support -->
[M] i.MX SDMA support
```

3. To integrate Wi-Fi functionality, add the following packages in IMAGE_INSTALL in local.conf under imx-yacto-bsp/build/conf. Be sure to select the correct firmware recipe:

```
IMAGE_INSTALL:append = " kernel-module-nx-backports \
      nx61x-firmware \
      summit-supplicant-nx \
      summit-networkmanager-nx \
      summit-networkmanager-nx-nmcli \
```

Additional packages may be needed to add imx-sdma kernel module/firmware and packagegroup-tools-bluetooth in IMAGE_INSTALL in local.conf under imx-yacto-bsp/build:

```
IMAGE_INSTALL:append = " packagegroup-tools-bluetooth \
                         kernel-module-imx-sdma \
                         firmare-imx-sdma-imx7d"
```

Several device tree changes are required on the iMX8MP EVK for compatibility with the NX611 SDIO interface. The SDIO bus signal strength may need to reduce from the default value to improve signal integrity to get better performance result. The following is an example of device tree modification in the iMX8MP EVK: *imx8mp-evk-usdhc1-m2.dts.*

```
&iomuxc {
  pinctrl_usdhc1: usdhc1grp {
         fsl,pins = <
                MX8MP_IOMUXC_SD1_CLK__USDHC1_CLK      0x190
                MX8MP_IOMUXC_SD1_CMD__USDHC1_CMD      0x1d0
                MX8MP_IOMUXC_SD1_DATA0__USDHC1_DATA0  0x1d0
                MX8MP_IOMUXC_SD1_DATA1__USDHC1_DATA1  0x1d0
                MX8MP_IOMUXC_SD1_DATA2__USDHC1_DATA2  0x1d0
                MX8MP_IOMUXC_SD1_DATA3__USDHC1_DATA3  0x1d0
                MX8MP_IOMUXC_SD1_DATA4__GPIO2_IO06    0x10
        >;
  };

  pinctrl_usdhc1_100mhz: usdhc1-100mhzgrp {
        fsl,pins = <
                MX8MP_IOMUXC_SD1_CLK__USDHC1_CLK      0x190
                MX8MP_IOMUXC_SD1_CMD__USDHC1_CMD      0x1d0
                MX8MP_IOMUXC_SD1_DATA0__USDHC1_DATA0  0x1d0
                MX8MP_IOMUXC_SD1_DATA1__USDHC1_DATA1  0x1d0
                MX8MP_IOMUXC_SD1_DATA2__USDHC1_DATA2  0x1d0
                MX8MP_IOMUXC_SD1_DATA3__USDHC1_DATA3  0x1d0
                MX8MP_IOMUXC_SD1_DATA4__GPIO2_IO06    0x10
        >;
  };

  pinctrl_usdhc1_200mhz: usdhc1-200mhzgrp {
        fsl,pins = <
                MX8MP_IOMUXC_SD1_CLK__USDHC1_CLK      0x190
                MX8MP_IOMUXC_SD1_CMD__USDHC1_CMD      0x1d0
                MX8MP_IOMUXC_SD1_DATA0__USDHC1_DATA0  0x1d0
                MX8MP_IOMUXC_SD1_DATA1__USDHC1_DATA1  0x1d0
                MX8MP_IOMUXC_SD1_DATA2__USDHC1_DATA2  0x1d0
                MX8MP_IOMUXC_SD1_DATA3__USDHC1_DATA3  0x1d0
                MX8MP_IOMUXC_SD1_DATA4__GPIO2_IO06    0x10
        >;
  };
};
```

Hardware handshaking is required on the Bluetooth UART. Check if the UART port in use is configured with CTS/RTS handshaking enabled. The UART interface on the M.2 slot is uart1 (ttymxc0) on the iMX8MP EVK. The iMX8MP EVK device tree enables handshaking by default on uart1.

```
Imx8mp-evk.dts

&uart1    {  /*BT*/
  pinctrl-names = "default";
  …
      fsl,uart-has-ctsrts;
      status = "okay";
};
```

By default, the dtb file, *imx8mp-evk.dtb*, is only to enable PCIE interface. To enable the SDIO bus, you need to use *imx8mp-evk-usdhc-m2.dtb* instead.

4. Build core-image-minimal image:

```
bitbake core-image-minimal
```

5. Program the SD card with the whole image including kernel and root file system:

```
sudo dd if=xxxx.wic of=/dev/mmcblk0 bs=1M
```

Wi-Fi: Configuring the Supplicant or NetworkManager

The Wi-Fi connection can be configured either by NetworkManager or supplicant.

After driver loading, two virtual interfaces will come out, "mlan0" and "uap0" for STA and AP modes.

If you only want to run in either STA or AP mode, you may need to disable the other interface to reduce the RF sharing.

Here is the example while only run NX611 in STA mode.

```
ifconfig uap0 down
```

You can run either nmcli or sdcsupp on "mlan0" interface.

Configuration with NetworkManager

Here is an example to set up a connection with a WPA3-SAE network using NetworkManager.

```
nmcli conn add con-name NETGEAR84-5G ifname mlan0 type wifi ssid NETGEAR22-5G wifi.powersave 2 802-11-wireless-security.key-mgmt sae 802-11-wireless-security.psk miles123
```

Configuration with sdcsupp

You can also use the supplicant (sdcsupp) directly with a configuration file.

```
sdcsupp -Dnl80211 -c /etc/wpa_supplicant.conf -i mlan0 -B -dddddd

/etc/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
update_config=1
sae_groups=19
sae_pwe=2

network={
ssid="NETGEAR84-5G"
key_mgmt=SAE
sae_password="miles123"
ieee80211w=2
}
```

BT: Running btattach to bring up hci interface

The traditional btattach implementation for BT serial interface may be used for BT support.

Upon successfully loading this firmware by the Wi-Fi driver, btattach may be issued.

For example:

```
/usr/bin/btattach -B /dev/ttymxc0 -P h4 -S 3000000 2>&1 > /dev/null&
```

An alternative method to using btattach is to use the serdev driver btnxpuart.ko (available with kernel version 4.12 and higher). This mechanism is configured in the device tree.

Here is a sample device tree snippet configuring a serial port for use with the NX 611 radio

on the iMX8mp-evk evaluation board.

```
&uart1 {
    bluetooth {
            compatible = "nxp,88w8987-bt";
            fw-init-baudrate = <3000000>;
        };
};
```

Note - To eliminate race conditions that occur when loading the BT firmware via the Wi-Fi driver, the Wi-Fi driver, moal.ko, should be instructed to load just the Wi-Fi firmware, sd_w61x_v1.bin.se. This change results in the BT firmware always being loaded by the serdev driver.

Revision History

Version	Date	Notes	Contributor(s)	Approver
0.1	6 June 2024	Preliminary Release	Miles Chung	Bob Monroe

Ezurio's products are subject to standard Terms & Conditions.