I.MX8MM & IW416 WIFI & BT INTEGRATION AND TEST PROCEDURES

CAS MAGGIE JIANG

2022 NOV

Topics

This document describe how to use NXP WiFi & BT in customer i.MX board, how to test it , as well as modify some parameters in SW.

1. i.MX connect with different NXP WiFi & BT module.
2. NXP WiFi & BT integration in Linux & Android.
3. Some tips when debug WiFi & BT in customer i.MX board.
4. How to test WiFi & BT module using two different way.
5. Real case example on WiFi & BLT power adjustment through SW.

i.MX BSP supported NXP WiFi driver lists

- In i.MX current Yocto(L5.xx later version) and Android BSP, it has already included following NXP WiFi & BT driver and firmware:

NXP 88W8987 with SDIO interface

NXP 88W9098 with PCIe and SDIO interfaces

NXP 88W8997 with PCIe and SDIO interfaces

NXP IW416 with SDIO interface (NXP internal name is 8978,so some file name is 8978)

NXP 88W8801 with SDIO interface

NXP IW612 with SDIO interface.

WIFI & BT INTEGRATION

Use different WiFi & BT module in i.MX board

- For example : i.MX8MM EVK CPU2 board have CM358 module on board with WM8987 wifi inside.
- We can also use another uSD module LBEE5CJ2XK to test on i.MX8MM EVK.
- HW: i.MX8MM LPDDR4 EVK SDIO2 + LBEE5CJ2XK uSD interface module
- According to module datasheet, set the jumper to 1.8V SDIO. then connect with i.MX8MM EVK MicroSD port(SDIO2)

Integrate different NXP WiFi & BT into Yocto

In Yocto platform, if customer use other NXP WiFi list which is different from i.MX EVK default used, For example, if i.MX8MM EVK use PCIE 8987, while customer board use SDIO IW416, only dts file need to check. In most cases, default SW maybe OK.

- 1. Check dts file to confirm that PCIE or SDIO pin is set correctly. Sometimes they may need to add no-1-8-v to decrease SDIO timing requirement

```
&usdhc2 {
        keep-power-in-suspend;
        no-1-8-v;
        non-removable;
        wakeup-source;
        fsl,sdio-async-interrupt-enabled;
};
```

2. wifi driver will detect wifi type and interface, then it load corresponding wifi firmware automatically according to wifi_mod_para.conf. Load the modules in the kernel：

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

Integrate different NXP WiFi & BT into Yocto(continue)

For example, if it detect SDIO interface 8997, then it will use following structure to load 8997 firmware sdiouart8997_combo_v4.bin

wifi_mod_para.conf:

```
SD8997 = {
        cfg80211_wext=0xf
        max_vir_bss=1
        cal_data_cfg=none
        ps_mode=1
        auto_ds=1
        host_mlme=1
        fw_name=nxp/sdiouart8997_combo_v4.bin
}
```

wifi_mod_para.conf is located in following directory, No need to modify wifi_mod_para.conf

imx-yocto-bsp/imx8mp_xwayland/tmp/work/all-poky-linux/linux-firmware/1_20210818-r0/imx-firmware/nxp

Enable WiFi command line in Yocto board

modprobe moal mod_para=nxp/wifi_mod_para.conf

$connmanctl

connmanctl> enable wifi

connmanctl> scan wifi

connmanctl> services /* This should list of the network. For example wifi_c0e4347f5053_4a62726f_managed_psk*/

connmanctl> agent on

connmanctl> connect wifi_c0e4347f5053_4a62726f_managed_psk /*

……

connmanctl> quit

Enable BT command line in Yocto board

- WiFi and BT/BLE firmware are combined together, so it need to load WiFi & BT firmware firstly before running BT:
- modprobe moal mod_para=nxp/wifi_mod_para.conf

```
# For PCIe9098 and SDIO9098:
hciattach <device> any -s 3000000 3000000 flow
hciconfig hci0 up
# For all the other modules:
hciattach <device> any 115200 flow
hciconfig hci0 up
hcitool -i hci0 cmd 0x3f 0x0009 0xc0 0xc6 0x2d 0x00
killall hciattach
hciattach <device> any -s 3000000 3000000 flow
hciconfig hci0 up
```

Integrate different NXP WiFi & BT into Android

In Android platform, if customer use other NXP WiFi list which is different from i.MX EVK default used, For example, if i.MX8MM EVK use PCIE 8987, while customer i.MX8MM board use SDIO IW416, the modification is different with Yocto.

1. Check dts file on PCIE or SDIO setting which is same with Yocto

2. Modify following evk_8mm.mk file. It use W8987 in i.MX8MM EVK, need to modify to IW416(internal name is also 8978, so the config file name is wifi_mod_para_sd8978.conf here)

/android-12_2.0.0/android_build/device/nxp/imx8m/evk_8mm/evk_8mm.mk

```
PRODUCT_COPY_FILES += \
-    vendor/nxp/imx-firmware/nxp/FwImage_8987/sdiouart8987_combo_v0.bin:vendor/firmware/sdiouart8987_combo_v0.bin \
-    vendor/nxp/imx-firmware/nxp/android_wifi_mod_para.conf:vendor/firmware/wifi_mod_para_sd8987.conf
+     vendor/nxp/imx-firmware/nxp/FwImage_IW416_SD/sdiouartiw416_combo_v0.bin:vendor/firmware/sdiouartiw416_combo_v0.bin \
+     vendor/nxp/imx-firmware/nxp/android_wifi_mod_para.conf:vendor/firmware/wifi_mod_para_sd8978.conf
```

Enable WiFi & BT in Android

- In Linux, kernel load WiFi driver manually:

  modprobe moal mod_para=nxp/wifi_mod_para.conf
- In Android, during bootup stage, it will detect 416 WiFi in HW, so it automatically load WiFi & BT firmware, following log mean driver and firmware is loaded successfully.

```
[    8.420496][  T243] fw_name=sdiouartiw416_combo_v0.bin
[    8.425737][  T243] SDIO: max_segs=128 max_seg_size=65535
[    8.431151][  T243] rx_work=1 cpu_num=4
[    8.435030][  T243] Attach mlan adapter operations.card_type is 0x108.
[    8.442142][  T243] wlan: Enable TX SG mode
[    8.446348][  T243] wlan: Enable RX SG mode
[    8.459849][  T243] Request firmware: sdiouartiw416_combo_v0.bin
[    8.797279][  T243] Wlan: FW download over, firmwarelen=492076 downloaded 492076
[   10.108298][  T243] WLAN FW is active
……
[   10.174332][  T243] wlan: version = SDIW416---16.92.21.p41.1-MM5X16322.p3-(FP92)
```

Some tips when enable NXP WiFi & BT in customer board

1. When the interface bus driver detects the NXP-based wireless module, the MLAN module downloads the firmware binary via the interface adapter. So it is the first step to detect WiFi in HW.

if SD2 connect with WiFi, then it should have correct MMC information under 1.8V HW status:

evk_8mm:/ # **cat /sys/kernel/debug/mmc1/ios**

clock: 50000000 Hz
actual clock: 50000000 Hz
vdd: 21 (3.3 ~ 3.4 V)
bus mode: 2 (push-pull)
chip select: 0 (don't care)
power mode: 2 (on)
bus width: 2 (4 bits)
timing spec: 7 (sd uhs DDR50)
signal voltage: 1 (1.80 V)
driver type: 0 (driver type B)

If there is no such information, it need to check dts setting and HW interface. It also need to check WIFI module hardware status, such as voltage, reset pin status, etc.

Some tips when enable NXP WiFi & BT in customer board(conti)

2. When load firmware, it must print out following detail log, when it means that WiFi firmware is loaded successfully.

When board bootup, it have detected 416 WiFi, driver and firmware is loaded successfully

```
[    8.420496][  T243] fw_name=sdiouartiw416_combo_v0.bin
[    8.425737][  T243] SDIO: max_segs=128 max_seg_size=65535
[    8.431151][  T243] rx_work=1 cpu_num=4
[    8.435030][  T243] Attach mlan adapter operations.card_type is 0x108.
[    8.442142][  T243] wlan: Enable TX SG mode
[    8.446348][  T243] wlan: Enable RX SG mode
[    8.459849][  T243] Request firmware: sdiouartiw416_combo_v0.bin
[    8.797279][  T243] Wlan: FW download over, firmwarelen=492076 downloaded 492076
[   10.108298][  T243] WLAN FW is active
[   10.112083][  T243] on_time is 10108364500
[   10.137488][  T168] fw_cap_info=0x181c0f03, dev_cap_mask=0xffffffff
[   10.143830][  T168] max_p2p_conn = 8, max_sta_conn = 8
[   10.174332][  T243] wlan: version = SDIW416---16.92.21.p41.1-MM5X16322.p3-(FP92)
```

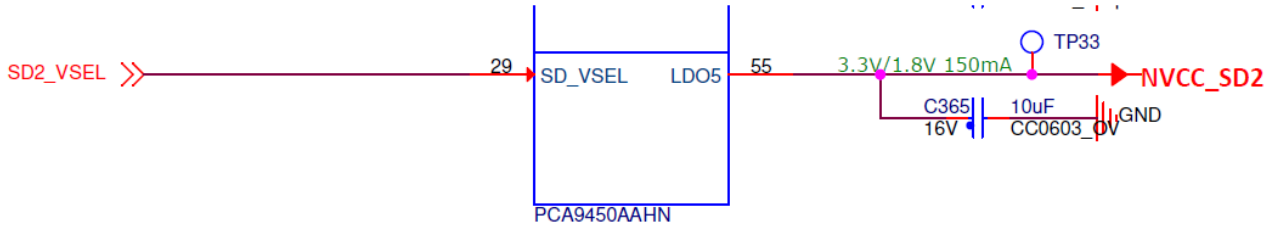
Some tips when enable NXP WiFi & BT in customer board(conti)

3. Normally WiFi module can work at 1.8V or 3.3V in HW, PMIC VSELECT pin input status define PMIC I.MX GPIO high/low input to PMIC SD2_VSEL , then NVCC_SD2 output 1.8V or 3.3V.

Check SD2_VSEL high/low status to make sure PMIC 1.8V/3.3V output to NVCC_SD2.

- For example, if customer HW design is 3.3V, then we set GPIO1_4(input to PCA9450 VSELECT pin) to low, then PCA9450 force output 3.3V to SDIO2
- Add no-1-8-v in i.MX side

```
&usdhc2 {
        pinctrl-assert-gpios = <&gpio1 4 GPIO_ACTIVE_LOW>;
        /delete-property/ cd-gpios;
        keep-power-in-suspend;
        no-1-8-v;
        non-removable;
        wakeup-source;
        fsl,sdio-async-interrupt-enabled;
};
```



Some tips when enable NXP WiFi & BT in customer board (conti)

4. When debug BT, WiFi &BT firmware is also need to load firstly.

5. NXP default BT firmware set flow control. So in i.MX side, RTS & CTS must be connected correctly and config in dts. If customer disable RTS/CTS in dts to expect it working without flow control, this debug way is wrong.

Customer need to check i.MX & module datasheet, to confirm RTS and CTS HW connection way. Otherwise, BT can’t work without flow control setting.

i.MX input – module output; i.MX output- module input. Normally DCE mode is set in i.MX side.

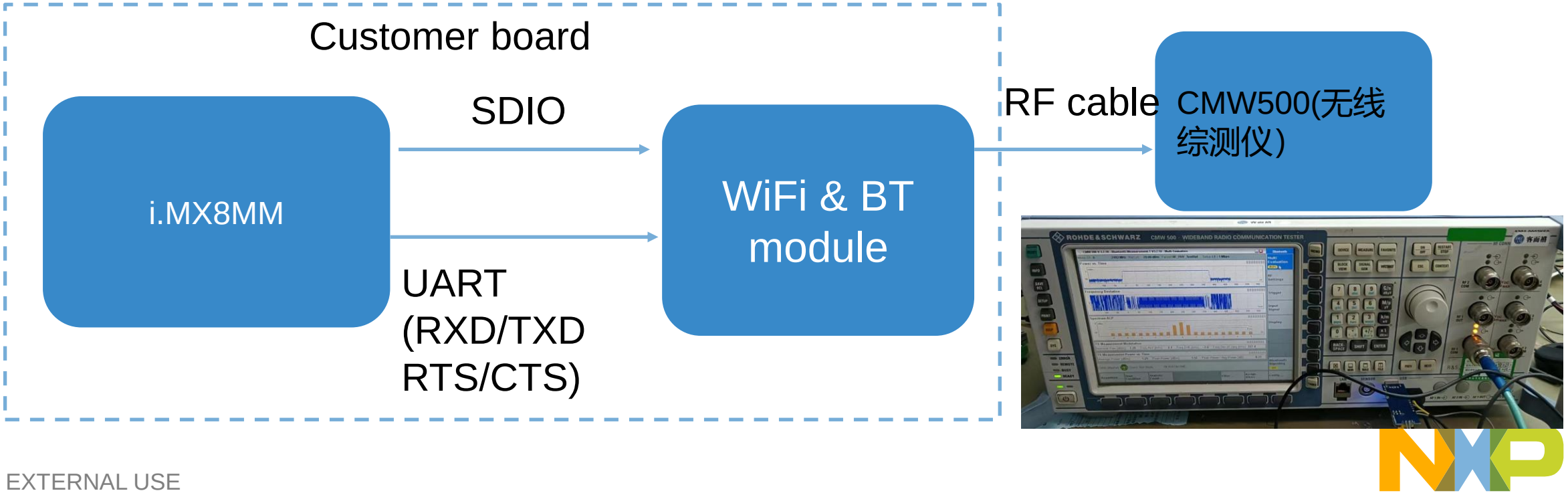
Table 67. UART I/O configuration vs. mode

Port	DTE Mode		DCE Mode	
	Direction	Description	Direction	Description
UARTx_RTS_B	Output	UARTx_RTS_B from DTE to DCE	Input	UARTx_RTS_B from DTE to DCE
UARTx_CTS_B	Input	UARTx_CTS_B from DCE to DTE	Output	UARTx_CTS_B from DCE to DTE
UARTx_TX_DATA	Input	Serial data from DCE to DTE	Output	Serial data from DCE to DTE
UARTx_RX_DATA	Output	Serial data from DTE to DCE	Input	Serial data from DTE to DCE

WIFI & BT TEST

Non signaling test(非信令测试)

- Most customer use this way. It use command line to send test command to WiFi module. No special firmware is needed and standard firmware inside BSP is OK.
- Customer can use hcitool command to send data between i.MX & module side. i.MX and module UART is must connected.



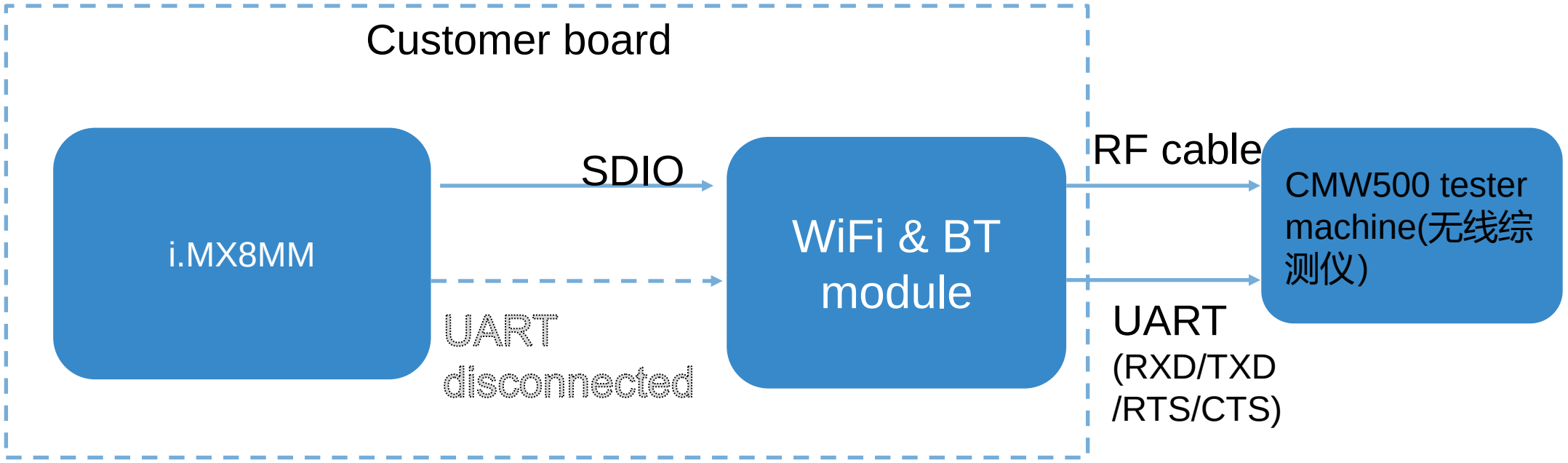
Non signaling test limitation

- When test BLE RX signal, machine side send data and i.MX8MM receive the data. But it can only test whether it received the data or not, while it can’t test delay time, sensitivity, etc. This is its limitation.
- In non signaling test, all the command is communicated between wifi module and tester machine, so i.MX isn’t involved in.
- NXP WiFi team provide Labtool tool as following, customer can refer to it on how to use command line to test.

Document : IW416 Labtool User Guide (public) (nxp.com)

Software : MFG-W8978-MF-LABTOOL-Native-1.0.0.13.0-16.80.21.p51 (nxp.com)

Signaling test 信令测试



In signaling test, i.MX and module UART is disconnected and the tester machine （CMW500综测仪） connect four signal UART with module.

Signaling test procedure

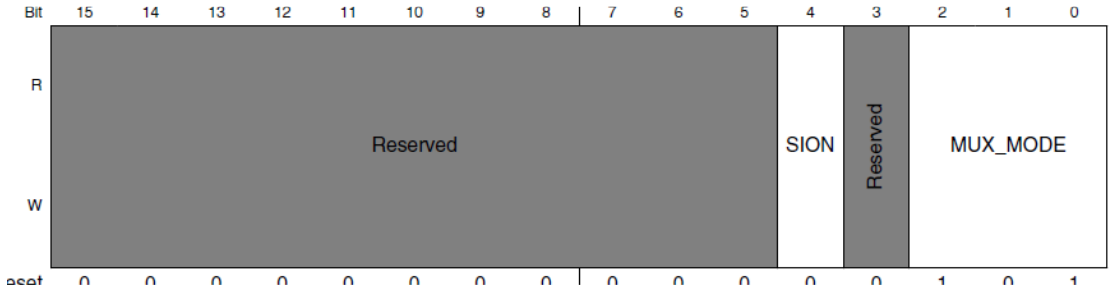
- i.MX8MM and module UART is disconnected in HW or set disable in dts.
- If customer use default HW & SW with i.MX UART working, it can also use following way:
- i.MX8MM and module UART is connected; module UART and CMW500(综测仪) is also connected.
- When bootup, i.MX load WiFi & BLE firmware to the module through SDIO. UART connection and setting is done in normal SW.
- After board bootup, we use memtool to set UART to GPIO. The purpose is to disconnect i.MX & module UART control and let CMW500(综测仪) to control module UART.
- Then CMW500(综测仪) send command to module through UART to get & receive test command freely. It can not only test throughtput, but also can test delay, sensitivity, etc.

Use memtool to disable i.MX & BT UART control

- ./memtool 0x300x303301F4 1

UART RXD signal is config in dts, so Bit0-Bit2 is 001 when reading, we need to set to 101 to set to GPIO

./memtool -32 0x303301F4=5



MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: ECSPI1_SCLK. 000 **ALT0** — Select signal ECSPI1_SCLK 001 **ALT1** — Select signal UART3_RX 101 **ALT5** — Select signal GPIO5_IO06

Set UART RXT/TXD/RTS/CTS four signal to GPIO config accordingly:

./memtool -32 0x303301F4=5

./memtool -32 0x303301F8=5

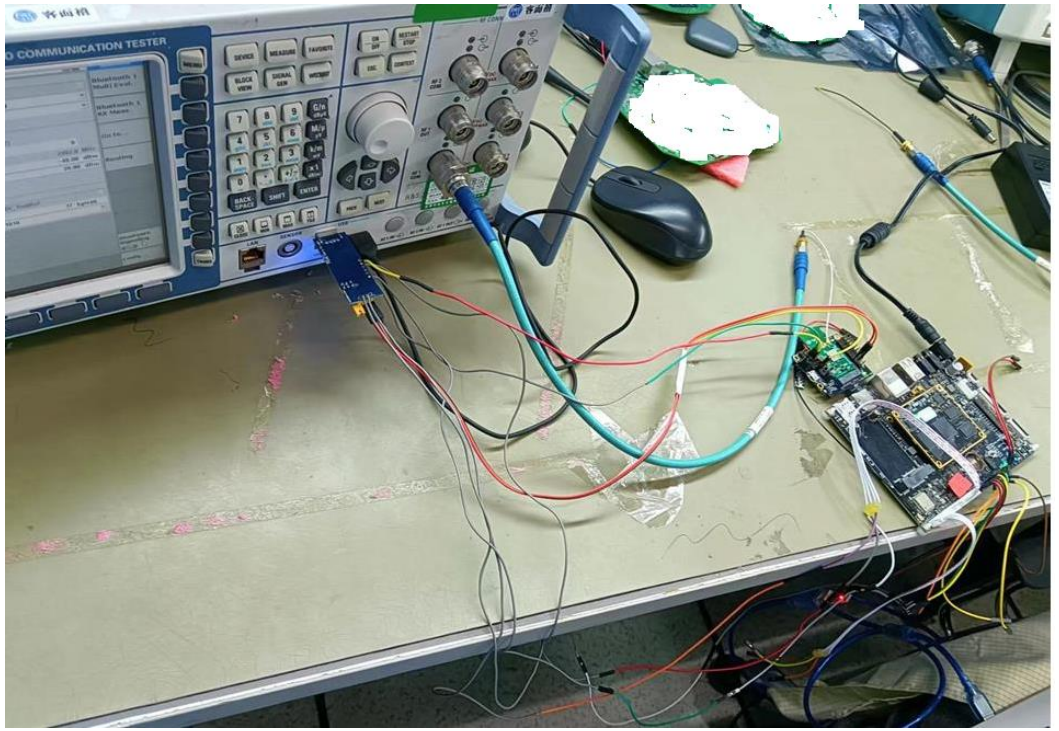
./memtool -32 0x303301FC=5

./memtool -32 0x30330200=5

Test environment

After fix following two points, then signaling test is realized in i.MX8MM board.

1. WiFi & BT module must connect
four signal line(RXD/TXD/RTS/CTS)
with the tester machine.



2.i.MX8MM and module UART
connection must be disconnected or
set to disable, or set to GPIO after bootup.

WIFI & BT PARAMETER ADJUSTMENT

Case example 1: Modify WiFi power table

All the firmware and config files are located in this directory:
/android-12_2.0.0/android_build/vendor/nxp/imx-firmware/nxp

FwImage_xxx directory is every wifi firmware directory
android_wifi_mod_para.conf is android config file.

```
nxa17678@lsv11191:~/Android/android-12_2.0.0/android_build/vendor/nxp/imx-firmware/nxp$
ls
android_wifi_mod_para.conf           FwImage_8997       FwImage_IW416_SD
android_wifi_mod_para_powersave.conf FwImage_8997_SD    SCR-nxp.txt
FwImage_8801_SD                      FwImage_9098_PCIE  wifi_mod_para.conf
FwImage_8987                         FwImage_9098_SD
```

Case example 1: Modify WiFi power table （continue)

1. **Get a new power table txpower_US.bin from module vendor and put it to IW416 firmware directory:**

/android-12_2.0.0/android_build/vendor/nxp/imx-firmware
/nxp/FwImage_IW416_SD/txpower_US.bin

2. Add txpower_US.bin to makefile:

/android-12_2.0.0/android_build/device/nxp/imx8m/evk_8mm/evk_8mm.mk

```
PRODUCT_COPY_FILES += \
+    vendor/nxp/imx-firmware/nxp/FwImage_IW416_SD/txpower_US.bin:vendor/firmware/txpower_US.bin \
```

3. **Add a line in config file SD8978 structure (8978 is NXP internal name for IW416)**

/android-12_2.0.0/android_build/vendor/nxp/android_wifi_mod_para.conf

```
  SD8978 = {
            auto_ds=1
            host_mlme=1
            fw_name=sdiouartiw416_combo_v0.bin
+       txpwrlimit_cfg=nxp/txpower_US.bin
```

Case example2: Modify BT power

- Background

IW416 silicon spec define the maximal BT & BLE power output value to 10db more, while Murata module side OTP setting is only 3db, so customer require to increase to 8db more to meet their product requirement.

Case example2: Modify BT power (continue)

- **Solution 1: Use hcitool command to set 8db after board bootup**

hcitool -i hci0 cmd 0x3f 0x00ee 0x01 0x08 （设定power为8dBm）
hcitool -i hci0 cmd 0x3f 0x00ee 0x01 （读取OTP的power设定值）
hcitool -i hci0 cmd 0x3 0x3 （Reset）

- **Solution 2: Use Labtool to rewrite OTP**

Command 53: Set calibration data to storage from text file
This command sets the calibration data to storage from the following calibration data text files:

- *CalWlanDataFile.txt*
- *CalBtDataFile.txt*
- *PwrTble_Path0.txt*

When the storage type is set to OTP, this command loads the calibration data generated from the text files into OTP.

Some documents reference

- IMX_LINUX_USERS_GUIDE
- Android-12.0.0_2.0.0_WiFi release notes
- IW416 Labtool user guide (UM11434) :

IW416 Labtool User Guide (public) (nxp.com)

- Getting Started with NXP-based Wireless Modules on i.MX 8M Quad EVK Running Linux OS user manual (UM11483)

Getting Started with NXP-based Wireless Modules on i.MX 8M Quad EVK Running Linux OS - User manual (restricted)

SECURE CONNECTIONS
FOR A SMARTER WORLD