

iMX8 EVM_8MQ
android_O8.1.0_1.3.0_8M

WL8 Porting Guide

V.0.1

Date: 2018/08/13

Version History

Date	Version	Remark
2018/08/13	Ver.0.1	The initial version

1. Preparation

This document provides information on integrating WiLink8 on Linux Android Oreo release for iMX8 EVK Borad + Jorjin WG78xx module.

Software requirements:

Host OS: Ubuntu14.04 64bit

BSP Version: android_O8.1.0_1.3.0_8M

Kernel Version: Linux 4.9

WiFi driver Version: Built-in in kernel.

WiFi firmware Version: Rev 8.9.0.0.78

BT firmware Version: 18xx_BT_Service_Pack_4.2

Get BSP from NXP's web site

[BSP for the i.MX 8M](#)

Please follow document to set build environment and build image to boot the board.

WiLink8 Release Notes/R8.7_SP3

http://software-dl.ti.com/ecs/WiLink8/SP/R8_7_SP3/exports/release_notes_R8_7_SP3.html

Hardware requirements:

WiFi Interface: SD2

Bluetooth Interface: UART3

External GPIO: WL_EN = GPIO5_IO12

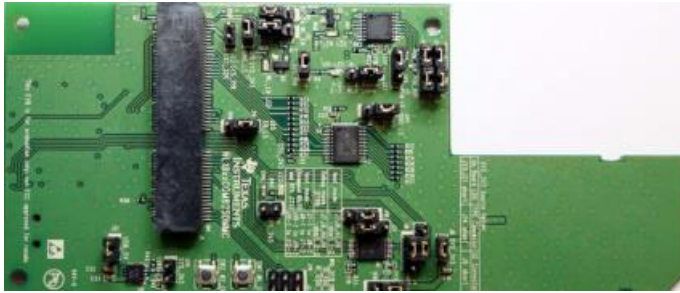
WL_IRQ = GPIO2_IO20

BT_EN = GPIO3_IO2

Hardware platforms: Evaluation Kit for the i.MX 8M Applications Processor

TI WiLink™ WL18xx WLAN SDIO/BT UART adapter Board: [wl18xxcom82sdmmc](#)

[Adapter kit](#)



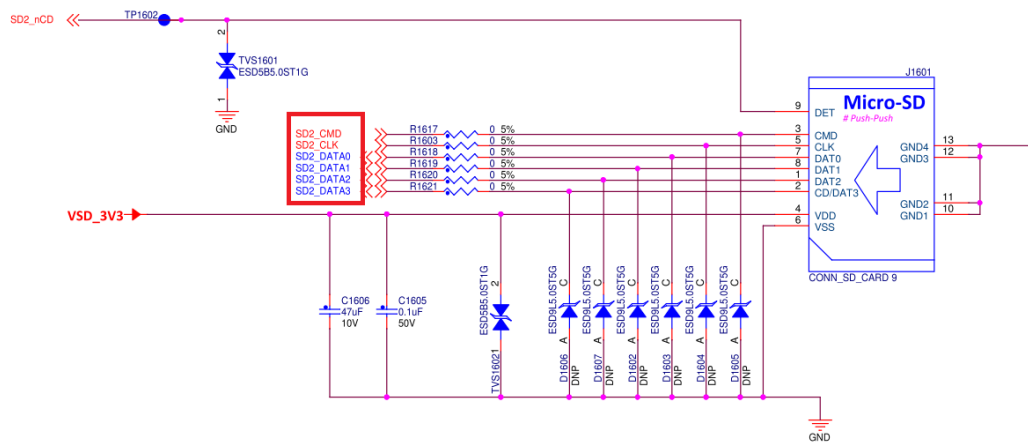
Jorjin WG78xx module family

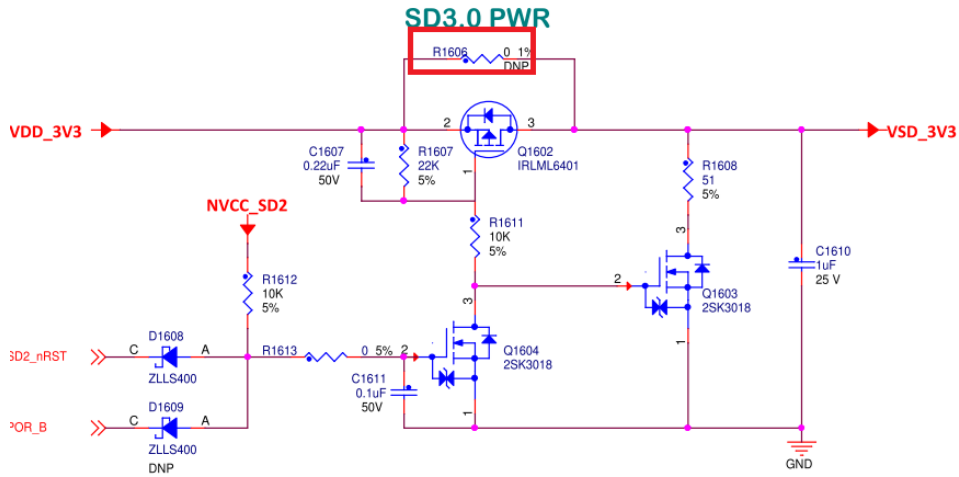
http://www.jorjin.com/products_2.php?id=11

1.1. Hardware Information

-WiFi-

iMX8 EVK side:

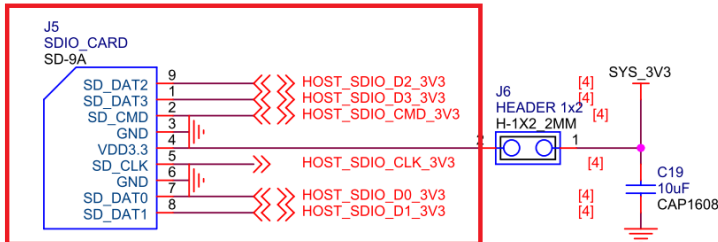




Note R1606 must be shorted.

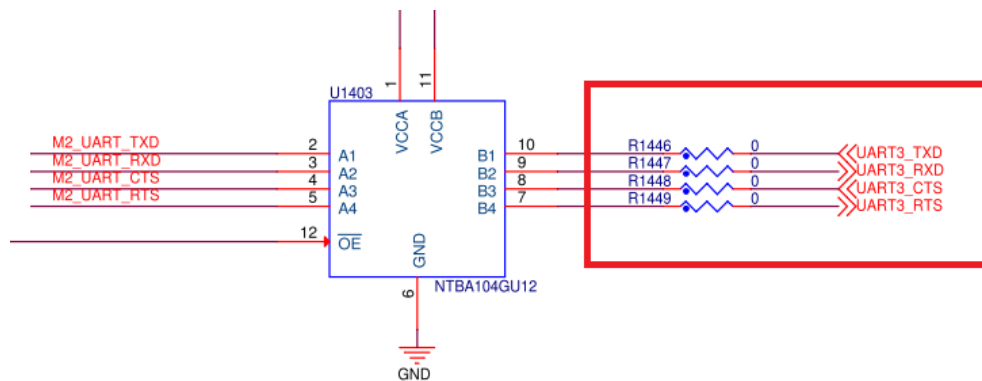
COM8SDMMC Side:

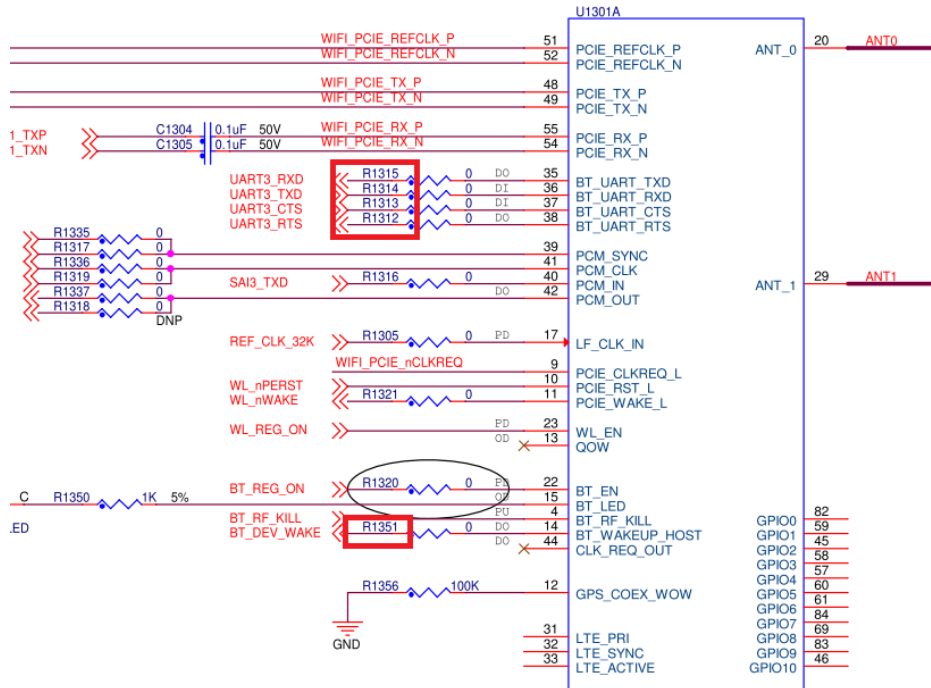
SD-MMC Connect



-BT-

iMX8 EVK Side:

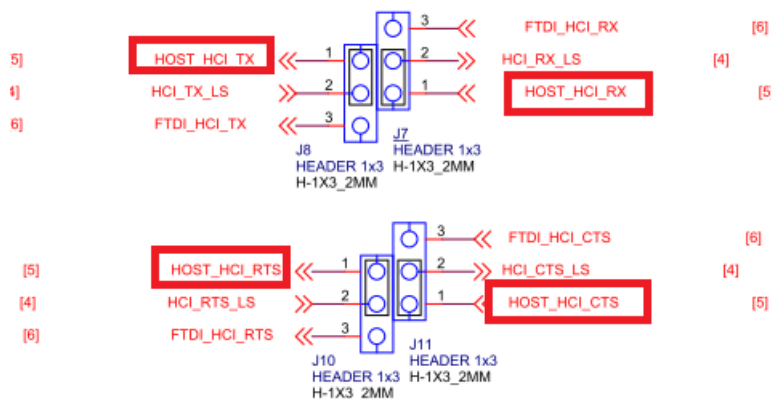




Note - We ported the TI WL18xx module on UART3, which will conflict with the original wireless module, so we removed R1312.R1313.R1314.R1315.R1351 resistance.

COM8SDMMC Side:

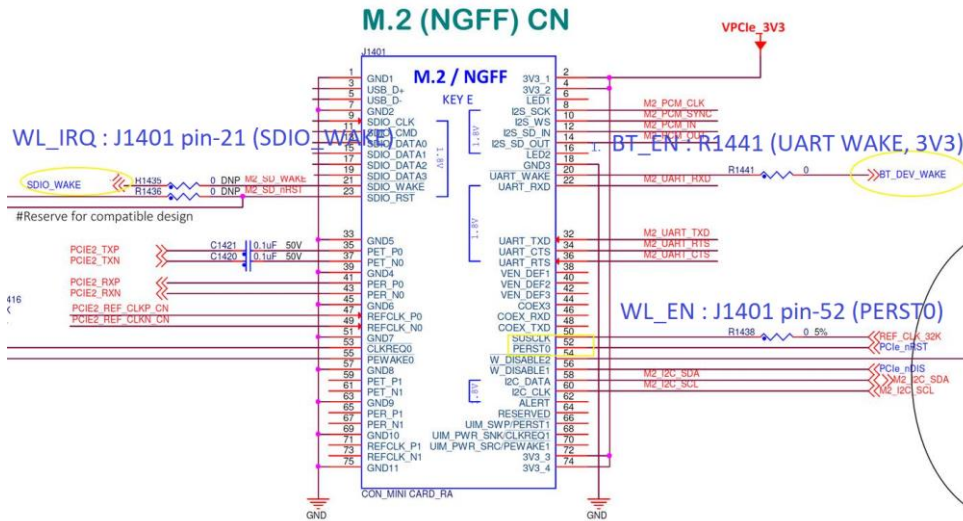
Host/FTDI UART Option



Host mode	Debug mode (FTDI)	Host mode (Swapped RX, TX and RTS, CTS)
J7, J8, J10, J11 -(short 1-2)	J7, J8, J10, J11 -(short 2-3)	J7-1 to J8-2 / J8-1 to J1-2 J10-1 to J11-2 / J11-1 to J12-2

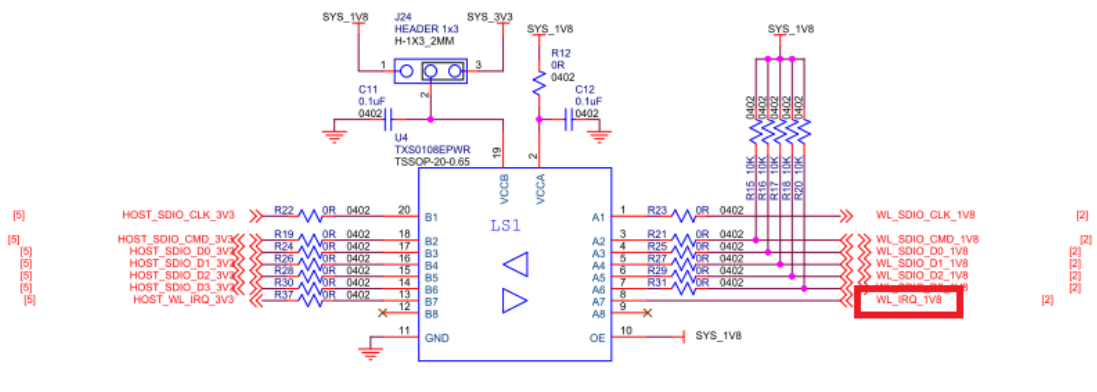
-External GPIO-

IMX8 EVK Side:

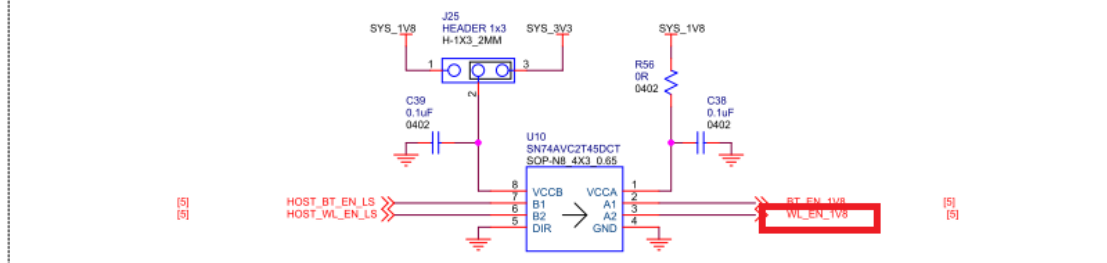


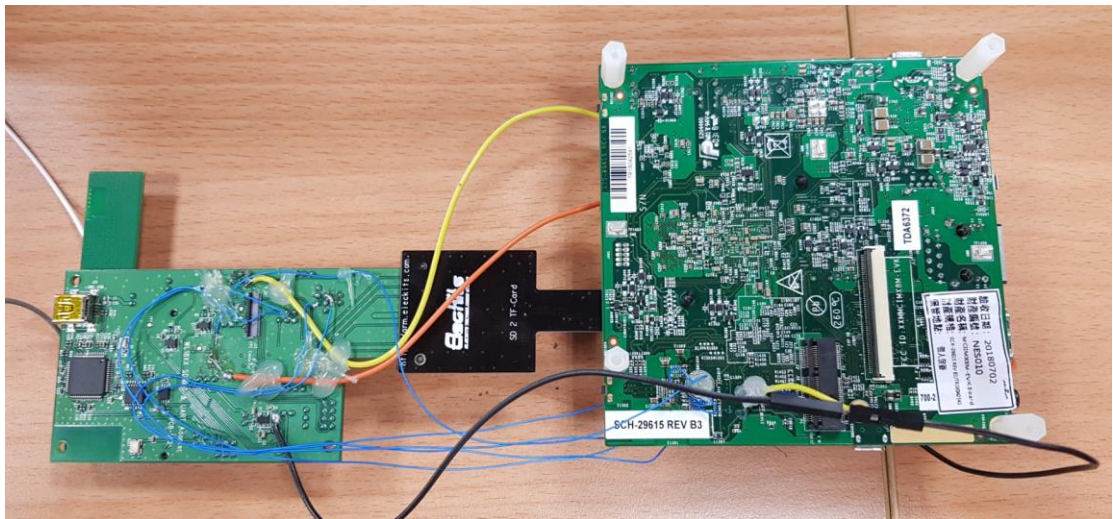
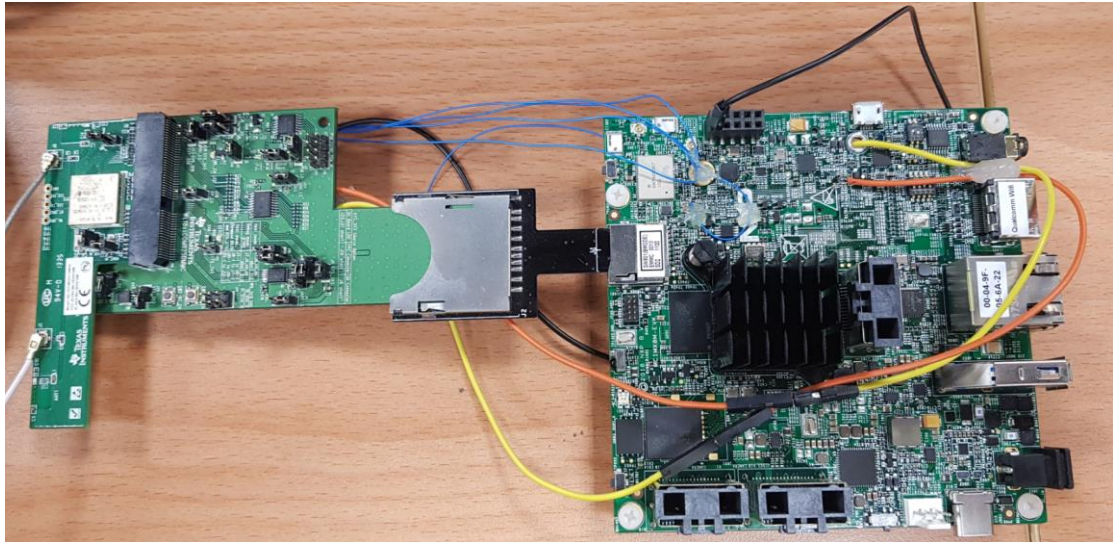
COM8SDMMC Side:

WiFi SDIO 1.8V to 3.3V Level Shifters



Enable 1.8V to 1.8V/3.3V Level Shifters





2. Prepare Linux Kernel and Android layer

2.1. Adding wilink8 related support to the android kernel

See:

[0001-Supported-TI-WL18XX-WiFi-with-Kernel-4.9-on-evk_8mq.patch](#)

[0002-Supported-TI-WL18XX-BT-with-Kernel-4.9-on-evk_8mq.patch](#)

Rebuild the android kernel after applying the kernel patches using the following sequence:

```
$ cd ${MY_ANDROID}/vendor/nxp-opensource/kernel-imx
$ export ARCH=arm64
$ export
CROSS_COMPILE=${MY_ANDROID}/prebuilts/gcc/linux-x86/aarch64/aarch64-linux-
android-4.9/bin/aarch64-linux-android-
$ make android_defconfig
$ make KCFLAGS=-mno-android
```

2.2. Integration of the wl18xx related package into the android

2.2.1 Get wifi/bt package

```
cd $MYDROID/hardware/
tar xvf ti.tar.bz2
```

2.2.2 Replace the BT stack .

```
rm $MYDROID/system/bt -rf
tar xvf bt.tar.bz2 -C $MYDROID/system
```

2.2.3 Patching the evk_8mq for enabling wl18xx

See:

[0001-Supported-TI-WL18XX-WIFI-in-device-fsl.patch](#)

[0002-Supported-TI-WL18XX-BT-in-device-fsl.patch](#)

[0003-Added-wpa-hostapd-config-file.patch](#)

[0004-Added-BDT-in-device-mk.patch](#)

2.2.4 Patching the Sepolicy.

See:

[0001-Allow-read-write-open-access-to-generic-devices.patch](#)

2.3. Rebuild Android final image

```
$ cd ${MYDROID}
$ source build/envsetup.sh
$ lunch evk_8mq-userdebug
$ make
```

3. Function Test

3.1. Station mode

1. Go to Settings → Network & Internet → Wi-Fi
2. Turn on wifi
3. Scan and choose an AP to connect
4. Browse to internet

3.2. AP mode

1. Go to Settings → Network & Internet → Hotspot & tethering
2. Enable Wi-Fi hotspot
3. Use another device connect to i.MX8 as access point

3.3. P2P mode

1. Go to Settings → Network & Internet → Wi-Fi → Wi-Fi Preferences → Wi-Fi Direct
2. Scan and connect to another wifi direct peer

3.4. Bluetooth

1. Go to Settings → Wireless & networks → Bluetooth
2. Enable Bluetooth
3. Scan and pair to another bluetooth device