

Renesas RZ/G1M
iW-RainboW-G20M
Jorjin WiLink 8 Porting Guide
V.0.1

Date: 2017/03/07

Version History

Date	Version	Remark
2017/03/07	Ver.0.1	The initial version

1. Preparation

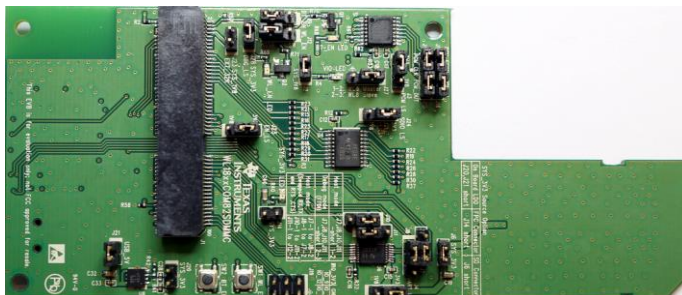
This document provides information on integrating WiLink8 on Linux 3.10.31 release for RZ/G1M Borad + Jorjin WG78xx module.

Software requirements:

- Host OS: Ubuntu14.04 64bit
- BSP Version:
iW-RainboW-G20M-Q7-R3.0-REL2.0-Linux3.10.31-YoctoDaisy_Deliverables
- GCC Version: gcc4.8.3 arm-poky-linux-gnueabi-
- Kernel Version: Linux 3.10.31
- Wi-Fi driver Version: R8.6_SP1
- Bluez Vesion : 5.15
- Bluetooth Firmware Version : 18xx_BT_Service_Pack_3.9

Hardware requirements:

- Hardware platforms: iW-RainboW-G20D RZ/G1 platform
- 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. Setup your Hardware

This section will walk you through setting-up your Renesas iWG-20 board for use with the WiLink8 Demos:

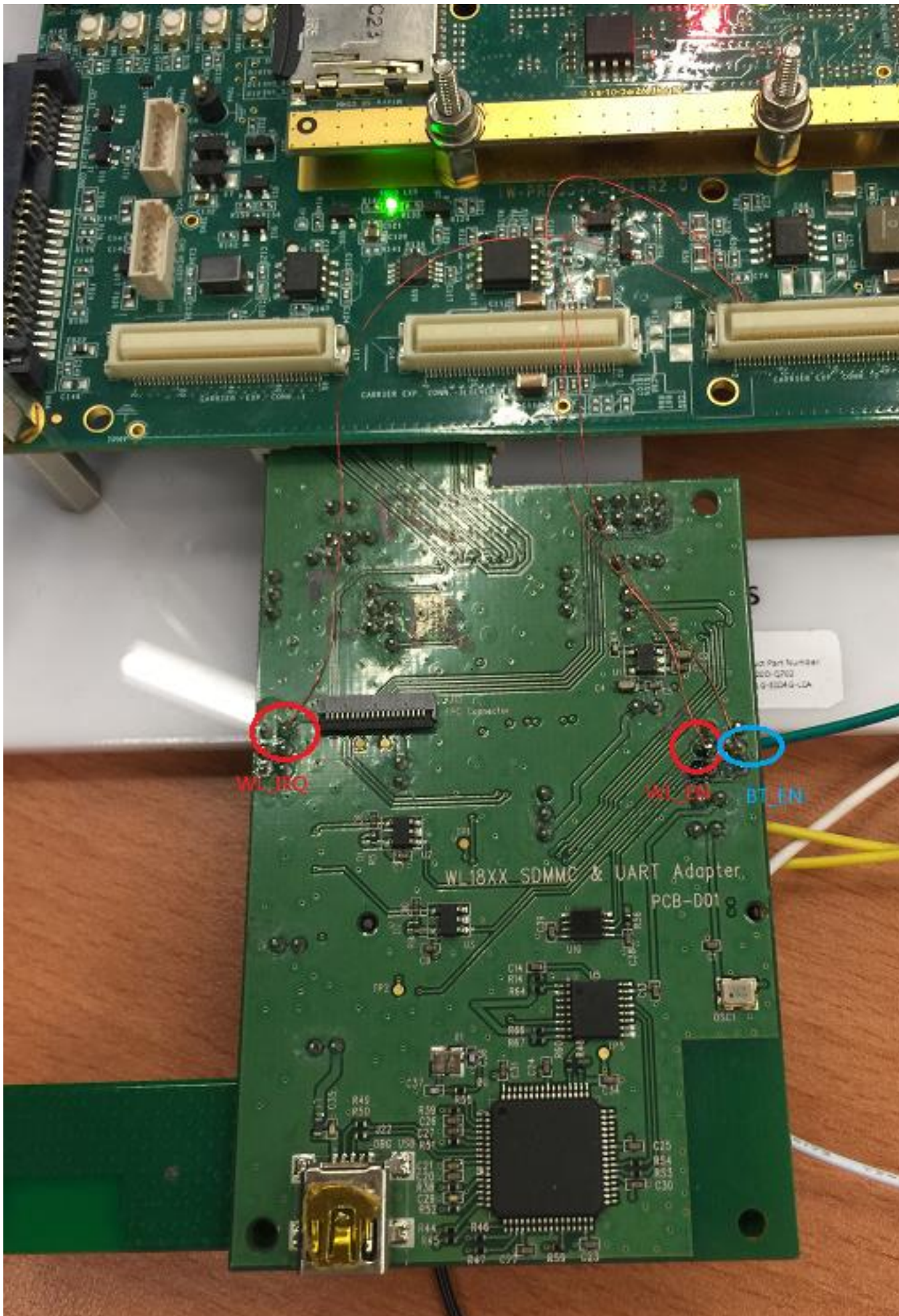
■ **Wi-Fi Part**

1. Plug the WL18xx SDIO Board into the SD slot of the iWG-20 board



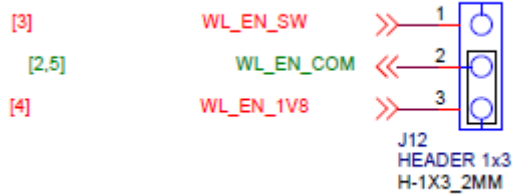
Sl.No.	Carrier board connection	G1M CPU GPIO	VIO	CPU Ball
1	Expansion 2 Pin-1	GP0_15	3.3V	W1
2	Expansion 2 Pin-3	GP0_14	3.3V	W2
3	Expansion 2 Pin-5	GP0_6	3.3V	AA1

2. The GP0_15 need to wire on wl18xxcom8sd2mmc board for wl_irq pin.
3. The GP0_14 need to wire on wl18xxcom8sd2mmc board for bt_en pin.
4. The GP0_6 need to wire on wl18xxcom8sd2mmc board for wl_en pin.



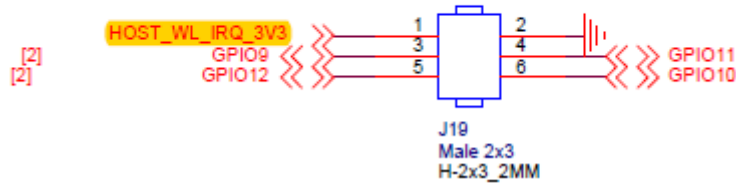
WL18XX SDIO Board:

WL_EN



WL_IRQ

DBG Header



■ **Bluetooth Part**

WL18xx HOST_BT_EN_LS → GP0_14

WL18xx HCI_RX_LS → J1 Pin4

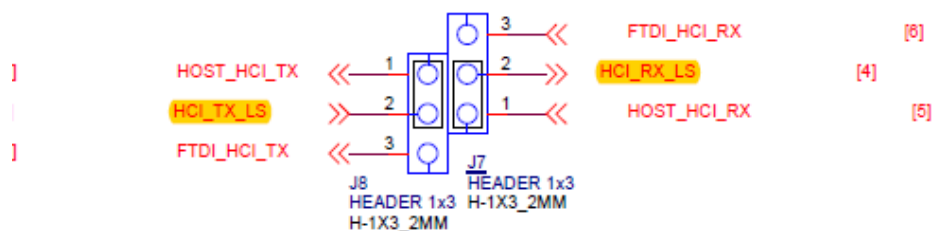
WL18xx HCI_TX_LS → J1 Pin5

WL18xx HCI_RTS_LS → GND

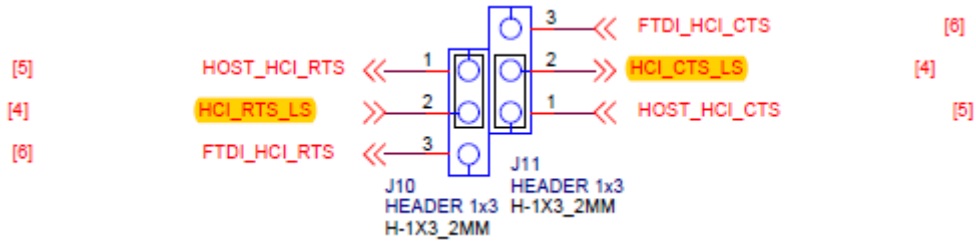
WL18xx HCI_CTS_LS → GND

WL18XX SDIO Board – HCI_TX_LS , HCI_RX_LS

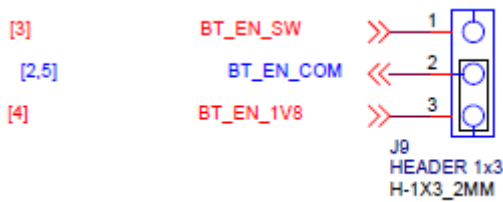
Host/FTDI UART Option



WL18XX SDIO Board – HCI_RTS_LS , HCI_CTS_LS



WL18XX SDIO Board – HOST_BT_EN_LS

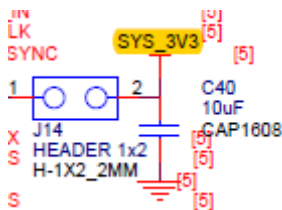


iWG20 board – J1 pin(Data Uart)

Table 5: Data UART Header Pin Out

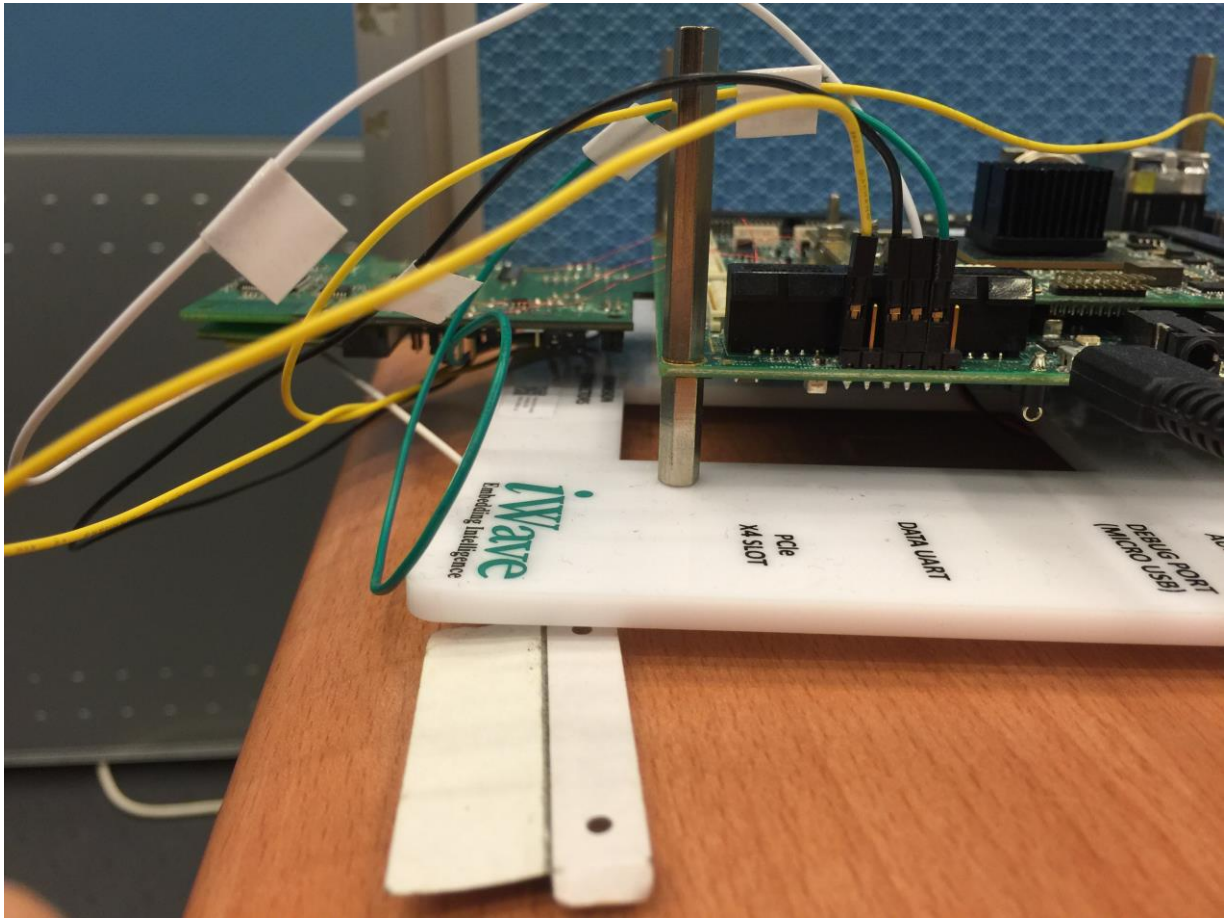
Pin No	Pin Name	Signal Name	Signal Type/ Termination	Description
1	GND	GND	Power	Ground.
2	UART_CTS#	SCIFB1_CTS#(GP7_8)	O, 3.3V CMOS	SCIFB1 interface Clear to Send signal.
3	VCC_3V3	VCC_3V3	O, 3.3V Power	3.3V Supply Voltage.
4	UART_RXD	SCIFB1_RXD(GP7_10)	I, 3.3V CMOS	SCIFB1 interface Receive signal.
5	UART_TXD	SCIFB1_TXD(GP7_12)	O, 3.3V CMOS	SCIFB1 interface Transmit signal.
6	UART_RTS#	SCIFB1_RTS#(GP7_9)	I, 3.3V CMOS	SCIFB1 interface Ready To Send signal.

J1 Pin3 need to wire to wl18xxcom8sd2mmc board SYS_3V3



Please note that we use only 2 pins(Tx, Rx) for Uart transmit since 3M baud rate would cause failed.

So WL18xx HCI_RTS_LS and WL18xx HCI_CTS_LS need to connect to GND.



2. Prepare Linux Kernel, Yocto And Driver For WiFi & BT

2.1. Get BSP from iWave

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

2.2. Adding WL8 related support to the kernel and yocto

Please applying the kernel patch and yocto patch.

Replace wpa-supPLICANT for R8.6_SP1

```
$ cd ${YOUR_PATH}/iwg20m-release-bsp/meta/recipes-connectivity
$ rm -rf wpa-supPLICANT
$ tar -xvf wpa-supPLICANT_R8.6_SP1.tar.bz2 -C
${YOUR_PATH}/iwg20m-release-bsp/meta/recipes-connectivity
```

Rebuild kernel and yocto

```
$ cd ${YOUR_PATH}/iwg20m-release-bsp
$ source poky/oe-init-build-env
$ bitbake -f -c compile linux-iwg20m
$ bitbake -c deploy linux-iwg20m -DDD
$ bitbake core-image-x11
```

2.3. Copy BT firmware to root filesystem

Copy the TI BT firmware to the target rootfs

```
$ cp BT_firmware/*.bts ${TARGET_ROOTFS}/lib/firmware/ti-connectivity/
```

2.4. WiLink8 Driver Release Notes/R8.6_SP1

http://processors.wiki.ti.com/index.php/WiLink8_Release_Notes/R8.6_SP1

3. Function Test

3.1. WiFi Station mode

1. Enable Wi-Fi

```
$ ifconfig wlan0 up
$ wpa_supplicant -Dnl80211 -iwlan0 -c/etc/wpa_supplicant.conf -B
```

2. Scan for Wi-Fi access points

```
$ wpa_cli -i wlan0 scan
$ wpa_cli -i wlan0 scan_results
```

3. Connect an Wi-Fi AP

```
$ wpa_cli -iwlan0 disconnect
$ for i in `wpa_cli -iwlan0 list_networks | grep ^[0-9] | cut -f1`; do wpa_cli -iwlan0 remove_network $i; done
$ wpa_cli -iwlan0 add_network
$ wpa_cli -iwlan0 set_network 0 auth_alg OPEN
$ wpa_cli -iwlan0 set_network 0 key_mgmt WPA-PSK
$ wpa_cli -iwlan0 set_network 0 psk "guest123"
$ wpa_cli -iwlan0 set_network 0 proto RSN
$ wpa_cli -iwlan0 set_network 0 mode 0
$ wpa_cli -iwlan0 set_network 0 ssid "AP-Guest"
$ wpa_cli -iwlan0 select_network 0
$ wpa_cli -iwlan0 enable_network 0
$ wpa_cli -iwlan0 reassociate
$ wpa_cli -iwlan0 status
$ iw wlan0 link
```

4. Get DHCP IP

```
$ udhcpc -i wlan0
```

5. Disable Wi-Fi

```
$ killall wpa_supplicant
```

More details about wpa_supplicant as link below.

[http://processors.wiki.ti.com/index.php/Connect to Secure AP using WPA Supplic
ant](http://processors.wiki.ti.com/index.php/Connect_to_Secure_AP_using_WPA_Supplicant)

3.2. WiFi AP mode

1. Enable Wi-Fi

```
$ ifconfig wlan0 up  
$ hostapd /etc/hostapd.conf -B  
$ ifconfig wlan0 192.168.0.1
```

2. Disable Wi-Fi

```
$ killall hosaptd
```

More details about hostapd as below link.

[http://processors.wiki.ti.com/index.php/OMAP Wireless Connectivity NLCP WLAN
_AP Configuration Scripts](http://processors.wiki.ti.com/index.php/OMAP_Wireless_Connectivity_NLCP_WLAN_AP_Configuration_Scripts)

3.3. BT Pairing

1. Enable Bluetooth

```
root@iWave-G20M:~# cd /usr/share/wl8-demos/  
root@iWave-G20M:/usr/share/wl8-demos# ./BT_Init.sh
```

2. Start Bluetooth daemon

```
root@iWave-G20M:~# cd /usr/lib/bluez5/bluetooth/  
root@iWave-G20M:/usr/lib/bluez5/bluetooth# ./bluetoothd &
```

Start bluetoothctl:

```
user $bluetoothctl
```

List the available controllers:

```
[bluetooth]#list
```

Display information about a controller:

```
[bluetooth]#show controller_mac_address
```

Set the default controller:

```
[bluetooth]#select controller_mac_address
```

Power on the controller:

```
[bluetooth]#power on
```

Enable the agent and set it as default:

```
[bluetooth]#agent on
```

```
[bluetooth]#default-agent
```

Set the controller as discoverable (temporarily for 3 minutes) and pairable:

```
[bluetooth]#discoverable on
```

```
[bluetooth]#pairable on
```

Scan for devices:

```
[bluetooth]#scan on
```

Put the device into pairing mode. This generally involves pressing a button or a combinations of buttons, usually for several seconds.

Discover the device MAC address:

```
[bluetooth]#devices
```

Pair with the device:

```
[bluetooth]#pair device_mac_address
```

Enter the PIN if prompted:

```
[agent]PIN code: ####
```

Allow the service authorization if requested:

```
[agent]Authorize service service_uuid (yes/no): yes
```

Trust the device:

```
[bluetooth]#trust device_mac_address
```

Connect to the device:

```
[bluetooth]#connect device_mac_address
```

Display information about the device:

```
[bluetooth]#info device_mac_address
```

The device is now paired:

```
[bluetooth]#quit
```