User Manual

DA16200 DA16600 Linux Driver Getting Started Guide

UM-WI-059

Abstract

The DA16200 is a highly integrated ultra-low power Wi-Fi system on chip (SoC) that allows users to develop a complete Wi-Fi solution on a single chip. The DA16600 also includes the DA14531 BLE solution. This document is a DA16200/DA16600 getting started guide intended to help new or existing developers quickly get started using Linux driver source codes to develop Wi-Fi and BLE applications with the DA16200/DA16600 chipset on RZ/G2L Linux system.

Contents

Figures

Tables

1 Terms and Definitions

AP	Access Point
BSP	Board Support Package
CMD	Command
FW	Firmware
GPIO	General Purpose Input Output
MAC	Medium Access Control
OTP	One-Time Programmable Memory
RX	Receive
SDIO	Secure Digital Input Output
SPI	Serial Peripheral Interface
STA	Station
TCP	Transport Control Protocol
TX	Transmit
UART	Universal Asynchronous Receiver-Transmitter
UDP	User Datagram Protocol

2 References

[1] UM-WI-046, DA16200 DA16600, FreeRTOS SDK Programmer Guide, User Manual, Renesas Electronics

3 Overview

The DA16200 is a highly integrated Wi-Fi system on chip (SoC) that allows users to develop Wi-Fi solutions using a single chip. The DA16600 also includes DA14531 BLE solution. The DA16200/DA16600 provides the Linux driver for the SPI interface and BLE HCI for the UART interface, so it can be used as a MAC-only chip on the RZ/G2L Linux system.

This document is a getting started guide to help new or existing developers develop Wi-Fi and BLE systems with the DA16200/DA16600 chipset on RZ/G2L Linux systems using the DA16200 and DA16600 Linux driver source codes quickly.

4 Linux Driver for Wi-Fi

4.1 Required Preparations

This section describes how to erase the flash memory. The flash memory should be erased before flash programming.

4.1.1 How to Erase Flash

After connecting to DA16200 or DA166600 via UART, run the command below in the command prompt.

[combo] reset

[MROM] sflash erase 0 1000

[MROM] sflash read 0 100 (Check if ff ff ff ff is printed)

```
        ******************************************
        *      FCI   FC9K MaskROM BootLoader
        *  Cortex-M4 (XTAL 40000 KHz, SYS 120000 KHz)
        *  Console Baud Rate : 0 (00000000)
        *  HW Version Num.   : fc905010
        *  Build Option      : RomALL
        *  RoSDK Date & Time : Mar 13 2019 13:05:45
        *  Build Date & Time : Mar 13 2019 13:11:24
        *           http://www.fci.co.kr
        ******************************************


[BOOT] reset

        ******************************************
        *      FCI   FC9K MaskROM BootLoader
        *  Cortex-M4 (XTAL 40000 KHz, SYS 120000 KHz)
        *  Console Baud Rate : 0 (00000000)
        *  HW Version Num.   : fc905010
        *  Build Option      : RomALL
        *  RoSDK Date & Time : Mar 13 2019 13:05:45
        *  Build Date & Time : Mar 13 2019 13:11:24
        *           http://www.fci.co.kr
        ******************************************


[MROM] sflash erase 0 1000
[MROM] sflash read 0 100
[00000000] : FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
[00000010] : FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
[00000020] : FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
[00000030] : FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
[00000040] : FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
[00000050] : FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
[00000060] : FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
[00000070] : FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
[00000080] : FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
[00000090] : FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
[000000A0] : FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
[000000B0] : FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
[000000C0] : FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
[000000D0] : FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
[000000E0] : FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
[000000F0] : FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ................
[MROM]
```

Figure 1: Example of Erasing Flash

4.2 RZ/G2L Linux System

4.2.1 SD Card

The Wi-Fi Linux driver is based on the Renesas RZ/G2L Linux SD card image for Linux kernel and file systems. The SD card image is written to the SD card using the MS Windows' **Win32 Disk Imager** tool.

4.2.2 NFS

Ethernet cable is required to use NFS mount function. The supported kernel and image are built using the RZ/G Verified Linux Package V3.0.3. This package can be found in Renesas website (https://www.renesas.com/us/en/products/microcontrollers-microprocessors/rz-mpus/rzg-linux-platform/rzg-marketplace/verified-linux-package/rzg-verified-linux-package). Search for RZ/G Verified Linux Package v3.0.3, and then download RTK0EF0045Z9006AZJ-v3.0.3.zip.

4.3 Kernel Config

The wireless network stack is not enabled by default in the Linux kernel and must be enabled using menuconfig.

1. Run bitbake for kernel config.

```
build$ bitbake -c menuconfig virtual/kernel
```

2. To use mac80211 as a module driver, RFKill should be enabled.
   Networking support -> RF switch subsystem support<*> -> GPIO RFKILL driver <*>

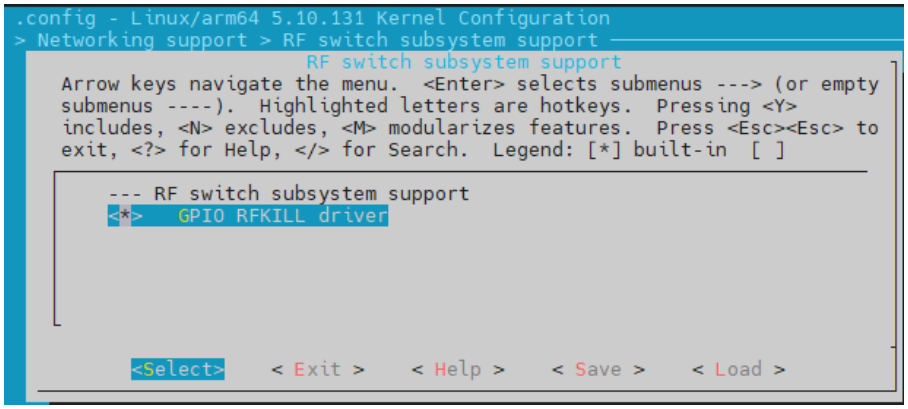

Figure 2: RFKILL Configuration

3. To enable the Wireless drivers, go to Networking support -> Wireless, and choose **Select**.

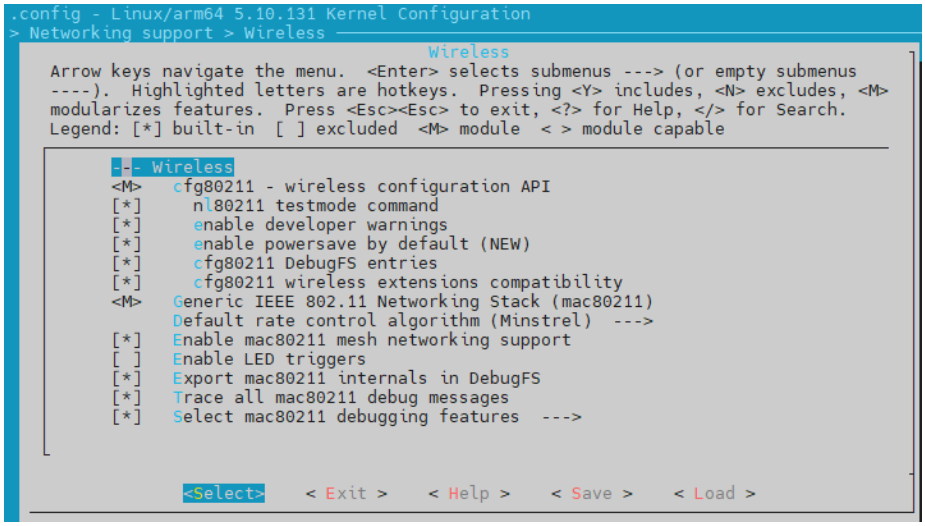

Figure 3: Wireless Configuration

4. Build the kernel.

```
build$ bitbake core-image-weston
```

4.3.1 SD Card

Copy `build/tmp/deploy/images/smarc-rzg2l/Image` to the SD card boot partition.

4.3.2 NFS

Copy `build/tmp/deploy/images/smarc-rzg2l/Image` to NFS server.

4.4 U-Boot Environment

4.4.1 SD Card

Set the environment variables using the following commands at the U-boot prompt to boot from the microSD card on the SMARC carrier board.

```
=> env default -a
=> setenv sd_boot1 'mmc dev 1 ; fatload mmc 1:1 0x48080000 Image ; fatload mmc 1:1
0x48000000 /Image-r9a07g044l2-smarc.dtb'
=> setenv sd_boot2 'setenv bootargs 'root=/dev/mmcblk1p2 rootwait' ; booti
0x48080000 - 0x48000000'
=> setenv bootcmd 'run sd_boot1 sd_boot2'
=> saveenv Saving Environment to MMC... Writing to MMC(0)….OK
```

4.4.2 NFS

Set the environment variables using the following commands at the U-boot prompt to boot to NFS on the SMARC carrier board.

```
=> setenv bootargsnfs 'setenv bootargs nfsrootdebug root=/dev/nfs rootfstype=nfs
ip=172.16.31.160:172.16.31.14:pi:255.255.224.0:pi:eth0:off
nfsroot=172.16.31.14:/nfsroot/rzg2lc,tcp,vers=3 vmalloc=384M'
=> setenv nfsload 'nfs 0x48080000 172.16.31.14:/nfsroot/rzg2lc/Image;nfs 0x48000000
172.16.31.14:/nfsroot/rzg2lc/Image-r9a07g044c2-smarc.dtb'
=> setenv serverip '172.16.31.14'
=> setenv ipaddr '172.16.31.160'
=> setenv netmask '255.255.224.0'
```

```
=> setenv ethact 'ethernet@11c20000'
=> setenv ethaddr '5A:54:E4:4D:03:9D'
=> setenv nfsboot 'run nfsload;run bootargsnfs;booti 0x48080000 - 0x48000000'
=> setenv bootcmd 'run nfsboot'
=> saveenv Saving Environment to MMC... Writing to MMC(0)….OK
```

The IP address, folder name, and ethaddr used above should be changed according to the user environment.

4.5 DA16200 Driver Module

4.5.1 Compile Driver Module (Optional)

DA16200 provides `da16200_linux_driver.tar.gz`, the Linux driver source code in the SDK. The driver is cross-compiled using "RZ/G Verified Linux Package V3.0.3" included in the RZ/G2L SDK. To compile the driver source code, the RZ/G2L SDK should be installed in advance. For how to build Yocto environment, see **r01us0553ej0107-rz-g (Release Note)** from the downloaded package (https://www.renesas.com/us/en/products/microcontrollers-microprocessors/rz-mpus/rzg-linux-platform/rzg-marketplace/verified-linux-package/rzg-verified-linux-package).

To build the driver, users need to change the path in KDIR with the one in the kernel build directory. The path can be found in the build folder which was generated when the BSP included in the RZ/G Verified Linux Package V3.0.3 was compiled.

```
vi Makefile
# rzg_bsp_v3.0.3 released by Renesas
KDIR ?= <RZ VLP directory>/build/tmp/work/smarc_rzg2l-poky-linux/linux-
renesas/5.10.158-cip22+gitAUTOINC+4f3d2d21ad-r1/linux-smarc_rzg2l-standard-build/
```

To build the driver, users need to enable the interface as shown below. For example, if a user is using a given interface, CONFIG_MODULE_TYPE should be set sdio or spi.

```
vi Makefile
# If using SDIO interface (default)
CONFIG_MODULE_TYPE ?= sdio

# If using SPI interface
CONFIG_MODULE_TYPE ?= spi
```

```
$ cd rswlan
$ source /opt/poky/3.1.17/environment-setup-aarch64-poky-linux
$ make
make -C  /home/liam/work/RZG_VLP_v3/v303/build/tmp/work/smarc_rzg2l-poky-
linux/linux-renesas/5.10.158-cip22+gitAUTOINC+4f3d2d21ad-r1/linux-smarc_rzg2l-
standard-build M=/home/work/da16200_linux_driver/rswlan modules
make[1]: Entering directory
'/home/liam/work/RZG_VLP_v3/v303/build/tmp/work/smarc_rzg2l-poky-linux/linux-
renesas/5.10.158-cip22+gitAUTOINC+4f3d2d21ad-r1/linux-smarc_rzg2l-standard-build'
  CC [M]  /home/work/da16200_linux_driver/rswlan/rs_main.o
  LD [M]  /home/work/da16200_linux_driver/rswlan/rswlan.o
  MODPOST /home/work/da16200_linux_driver/rswlan/Module.symvers
  LD [M]  /home/work/da16200_linux_driver/rswlan/rswlan.ko
make[1]: Leaving directory
'/home/liam/work/RZG_VLP_v3/v303/build/tmp/work/smarc_rzg2l-poky-linux/linux-
renesas/5.10.158-cip22+gitAUTOINC+4f3d2d21ad-r1/linux-smarc_rzg2l-standard-build'
```

When compile is completed, the “rswlan.ko” kernel module is created in the rswlan folder:

```
/work/da16200_linux_driver/rswlan$ ls rswlan.ko
rswlan.ko
```

4.5.1.1 Kernel 5.17 or Later

```
// Example of linux kernel 5.17
// makeFile
// change KERNELDIR
KERNELDIR ?= /home/liam/work/linux-stable/.out
// install gcc-arm-10.3
$ wget https://developer.arm.com/-/media/Files/downloads/gnu-a/10.3-
2021.07/binrel/gcc-arm-10.3-2021.07-x86_64-aarch64-none-elf.tar.xz
$ tar xvf gcc-arm-10.3-2021.07-x86_64-aarch64-none-elf.tar.xz -C /opt/arm/

$ cd rswlan
$ PATH=/opt/arm/gcc-arm-10.3-2021.07-x86_64-aarch64-none-elf/bin:$PATH ; export
CROSS_COMPILE=aarch64-none-elf-; export ARCH=arm64
$ make
make M=/home/liam/work/da16200_linux_driver/rswlan modules
make[1]: Entering directory '/home/liam/work/da16200_linux_driver/rswlan'
make -C  /home/liam/work/linux-stable/.out
M=/home/liam/work/da16200_linux_driver/rswlan modules
make[2]: Entering directory '/home/liam/work/linux-stable/.out'
  CC [M]  /home/work/da16200_linux_driver/rswlan/rs_main.o
  LD [M]  /home/work/da16200_linux_driver/rswlan/rswlan.o
  MODPOST /home/work/da16200_linux_driver/rswlan/Module.symvers
  LD [M]  /home/work/da16200_linux_driver/rswlan/rswlan.ko
make[2]: Leaving directory '/home/liam/work/linux-stable/.out'
make[1]: Leaving directory '/home/liam/work/da16200_linux_driver/rswlan'
$
```

4.5.2 Installing Driver as Linux Module

The DA16200 firmware is divided into `lmacfw_sdio.bin` and `lmacfw_spi.bin`, and either of them can be used depending on the interface supported by the platform. The firmware should be located in the /lib/firmware directory and is downloaded to the DA16200 when the kernel module is loaded. The kernel module driver can be loaded using the `insmod` command.

The rswlan.ko depends on mac80211 and cfg80211 module, which should be built into kernel. Or install the module (mac80211.ko, cfg80211.ko) before installing the rswlan.ko.

```
root@smarc-rzg2l:~# lsmod
Module                  Size  Used by
rswlan                212992  0
mac80211              589824  1 rswlan
libarc4                16384  1 mac80211
cfg80211              335872  2 mac80211,rswlan
```

4.5.2.1 SPI Interface

STA and AP mode are supported by SPI interface. The steps to install and configure the driver are as follows.

STA Mode

```
# insmod rswlan.ko
# wpa_supplicant -i wlan0 -c /etc/wpa_supplicant.conf &
# dhcpcd -q -b
```

AP Mode

```
# insmod rswlan.ko
```

```
# hostapd -iwlan0 -t -B /etc/hostapd_wlan0.conf
# ifconfig wlan0 10.0.0.1 netmask 255.255.255.0
# systemctl stop dnsmasq
# systemctl start dnsmasq

※ Create the followin
 to /etc/hostapd_wlan0.conf
interface=wlan0
hw_mode=g
channel=11
country_code=KR
ieee80211n=1
wmm_enabled=1
ssid=TEST-AP
driver=nl80211
ctrl_interface=/var/run/hostapd
ctrl_interface_group=0
# WPA2
auth_algs=1
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_passphrase=N12345678
rsn_pairwise=CCMP
wpa_pairwise=CCMP
group_cipher=CCMP

※ Add the following to /etc/dnsmasq.conf
resolv-file=/etc/resolv.dnsmasq
interface=wlan0
listen-address=10.0.0.1
bind-interfaces
domain-needed
server=8.8.8.8
bogus-priv
dhcp-range=10.0.0.2,10.0.0.20,255.255.255.0,24h
```

4.5.2.2 SDIO Interface

SDIO interface supports STA, AP, and Concurrent modes. For detailed information, see the following sections.

STA Mode

```
# insmod rswlan.ko
# wpa_supplicant -i wlan0 -c /etc/wpa_supplicant.conf &
# dhcpcd -q -b
```

AP Mode

```
# insmod rswlan.ko
# hostapd -iwlan0 -t -B /etc/hostapd_wlan0.conf
# ifconfig wlan0 10.0.0.1 netmask 255.255.255.0
# systemctl stop dnsmasq
# systemctl start dnsmasq

※ Create the following to /etc/hostapd_wlan0.conf
interface=wlan0
hw_mode=g
channel=11
```

```
country_code=KR
ieee80211n=1
wmm_enabled=1
ssid=TEST-AP
driver=nl80211
ctrl_interface=/var/run/hostapd
ctrl_interface_group=0
# WPA2
auth_algs=1
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_passphrase=N12345678
rsn_pairwise=CCMP
wpa_pairwise=CCMP
group_cipher=CCMP

※ Add the following to /etc/dnsmasq.conf
resolv-file=/etc/resolv.dnsmasq
interface=wlan0
listen-address=10.0.0.1
bind-interfaces
domain-needed
server=8.8.8.8
bogus-priv
dhcp-range=10.0.0.2,10.0.0.20,255.255.255.0,24h
```

Concurrent Mode

The AP channel should be the same as the STA connection channel.

```
# insmod rswlan.ko
# wpa_supplicant -i wlan0 -c /etc/wpa_supplicant.conf &
# dhcpcd -q -b

# iw dev wlan0 interface add wlan1 type managed
# hostapd -iwlan1 -t -B /etc/hostapd_wlan1.conf
# ifconfig wlan1 10.0.0.1 netmask 255.255.255.0
# systemctl stop dnsmasq
# systemctl start dnsmasq

※ Create the following to /etc/hostapd_wlan1.conf
interface=wlan1
hw_mode=g
# The channel should be the same as the STA connection(wlan0) channel
channel=11
country_code=KR
ieee80211n=1
wmm_enabled=1
ssid=TEST-AP
driver=nl80211
ctrl_interface=/var/run/hostapd
ctrl_interface_group=0
# WPA2
auth_algs=1
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_passphrase=N12345678
rsn_pairwise=CCMP
wpa_pairwise=CCMP
group_cipher=CCMP
```

```
※ Add the following to /etc/dnsmasq.conf
resolv-file=/etc/resolv.dnsmasq
interface=wlan1
listen-address=10.0.0.1
bind-interfaces
domain-needed
server=8.8.8.8
bogus-priv
dhcp-range=10.0.0.2,10.0.0.20,255.255.255.0,24h
```

4.6 SPI Pin Connection for Wi-Fi

The SPI pin connections between the DA16200 and the RZ/G2L are defined as follows.

Table 1: SPI Pin Connection for Wi-Fi

DA16200 Module	Signal Name	RZ/G2L PMOD0/1 Pin Name	RZ/G2L PMOD Pin Number
Reset	Reset	GPIO9_PMOD	PMODE1_PIN8
GPIO [2]	Interrupt to Host	GPIO5	PMODE0_PIN8
GPIO [9]	SPI MOSI	SPI1 DO	PMODE0_PIN2
GPIO [8]	SPI MISO	SPI DIN	PMODE0_PIN3
GPIO [7]	SPI CLK	SPI1 CK	PMODE0_PIN4
GPIO [6]	SPI CS	SPI1 CS0	PMODE0_PIN1

4.6.1 Set Up Interrupt Pin in Device Tree

The DMA (dmas), compatible, reg, spi-max-frequency, reset-gpios, irq0-gpios, and reset-gpios can be set in the dts file. Available values for spi_clk are 25, 16.7, 12.5, 10, and 5 Mhz. The following is the value set on the RZG2L EVK board, and DMA, reset-gpios and irq0-gpios should be changed appropriately for each board.

```
arch/arm64/boot/dts/renesas/rz-smarc-common.dtsi

&spi1 {
        pinctrl-0 = <&spi1_pins>;
        pinctrl-names = "default";
        dmas = <&dmac 0x2e99>, <&dmac 0x2e9a>;
        dma-names = "tx", "rx";
        status = "okay";
        da16xxx@0 {
                compatible = "renesas,da16xxx";
                reg = <0>;
                spi-max-frequency = <25000000>;
                reset-gpios = <&pinctrl RZG2L_GPIO(42, 3) GPIO_ACTIVE_LOW>;
                irq0-gpios = <&pinctrl RZG2L_GPIO(43, 2) GPIO_ACTIVE_LOW>;
        };
 };
```

4.7 SDIO Pin Connection for Wi-Fi

The SDIO pin connections between the DA16200 and the RZ/G2L are defined as follows:

Table 2: SDIO Pin Connection for Wi-Fi

DA16200 Module	Signal Name	RZ/G2L MicroSD Connector Pin Name	RZ/G2L MicroSD Connector Pin Number
Reset	Reset	GPIO9_PMOD	PMODE1_PIN8
GPIOA [4]	SDIO_CMD	CMD	3
GPIOA [5]	SDIO_CLK	CLK	5
GPIOA [6]	SDIO_D3	DATA3	2
GPIOA [7]	SDIO_D2	DATA2	1
GPIOA [8]	SDIO_D1	DATA1	8
GPIOA [9]	SDIO_D0	DATA0	7

4.7.1 Set Up Card Detection and Reset Pin for Host

DA16200 SDIO Linux Driver requires one GPIO in Host MCU for its proper SDIO probe operation. The GPIO acts as a Card Detect (CD) and Reset pin of SDIO. For RZ/G2L platform, the following changes are needed for CD and Reset pin configuration in dts file. In particular, to use the Reset pin, the Linux kernel CONFIG_PWRSEQ_SIMPLE=y must be set.

The GPIO for CD should be kept as ground state (or pulled-down state) for proper operation.

```
// rzg2l-smarc-som.dtsi (in case RZG2L only)

&pinctrl {
         sdhi1_pins: sd1 {
                  sd1_data {
                           pins = "SD1_DATA0", "SD1_DATA1", "SD1_DATA2",
"SD1_DATA3";
                                                    power-source = <3300>;
                  };

                  sd1_ctrl {
                           pins = "SD1_CLK", "SD1_CMD";
                           power-source = <3300>;
                  };

-                 sd1_mux {
-                          pinmux = <RZG2L_PORT_PINMUX(19, 0, 1)>; /* SD1_CD */
-                 };
+                 //sd1_mux {
+                 //            pinmux = <RZG2L_PORT_PINMUX(19, 0, 1)>; /* SD1_CD */
+                 //};
           };

// rz-smarc-common.dtsi
/ {
    aliases {
    …
    };
…
+    wireless_POR: wireless_POR {
+        compatible = "mmc-pwrseq-simple";
+        reset-gpios = <&pinctrl RZG2L_GPIO(42, 3) GPIO_ACTIVE_LOW>;
+        reset-duration-ms = <10>;
+        post-reset-delay-ms = <100>;
+    };
}
```

```
…
&sdhi1 {
        pinctrl-0 = <&sdhi1_pins>;
        pinctrl-1 = <&sdhi1_pins_uhs>;
        pinctrl-names = "default", "state_uhs";
+       cd-gpios = <&pinctrl RZG2L_GPIO(19, 0) GPIO_ACTIVE_LOW>;
        vmmc-supply = <&reg_3p3v>;
        vqmmc-supply = <&vccq_sdhi1>;
-       non-removable;
        bus-width = <4>;
        sd-uhs-sdr50;
        sd-uhs-sdr104;
+       mmc-pwrseq = <&wireless_POR>;
        status = "okay";
};
```

4.8 BT COEX Configuration

The BT COEX feature is disabled by default. To enable the feature, run the following command with the required options. For a detailed description of the BT COEX options, see Ref. [1].

```
$ insmod rswlan.ko bt_coex=1
```

Options:

```
0: bt coex off (default)
1: bt > wi-fi priority
2: bt = wi-fi priority
3: bt < wi-fi priority
```

Pin configuration of Bluetooth® LE coexistence is defined as follows:

Table 3: Pin Configuration of Bluetooth® LE Coexistence

DA16200 Module	Signal Name	Description	Note
GPIO [10]	iBtAct	BT active signal	For DA16600 or DA14531, the pin should be connected to GPIO P0_6.

4.9 MAC Address

The Wi-Fi MAC address of the DA16200 is stored in the OTP memory area.

4.10 Architecture

The DA16200 Linux driver architecture is as follows:

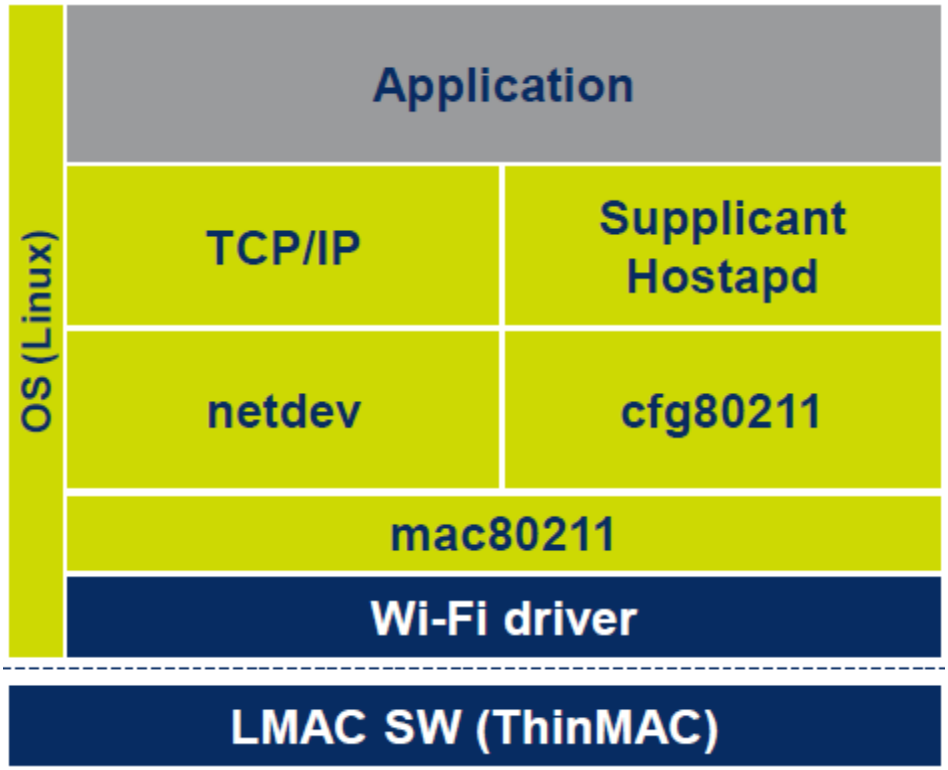

Figure 4: Linux Driver Architecture

- LMAC SW (`lmacfw_sdio.bin or lmacfw_spi.bin`) is the DA16200 firmware stored in the "`/lib/firmware`" folder
- Wi-Fi driver (`rswlan.ko`) is compiled from the DA16200 Wi-Fi driver source code
- rs_spi.c provides the SPI interface for the LMAC SW
- rs_sdio.c provides the SDIO interface for the LMAC SW

4.11 H/W Reset

RG2L+DA16200 supports H/W reset function for DA16200 module in SDIO and SPI interface. Use hw_reset_fw() function for hardware reset.

Table 4: Pin Configuration of HW Reset

DA16200 Module	Signal Name	RZ/G2L MicroSD Connector Pin Name	RZ/G2L MicroSD Connector Pin Number
RTC_PWR_KEY	HW reset	GPIO5	8

4.12 Throughput Test

This section describes the iperf3 throughput test using the TCP/UDP client/server protocol. RZG2L+DA16200 supports the `iperf3` command to measure packet transmission performance.

4.12.1 Test Setup

Install the iperf3 tool as follows:

1. Download Iperf3 from https://iperf.fr/iperf-download.php.
   Renesas recommends using **Iperf3 Version 3.1.3**.
2. Create a folder called **iperf3** in path **C:**.

3. Unzip the downloaded file and move the contents to the iperf3 folder.
4. Use the iperf3 or iperf3 -h command to check the options available for iperf3.

4.12.2 iperf3 Test with Client Mode

Set up the iperf3 test for Client mode as follows:

1. Connect the server (the host PC) to the AP.
2. In the command prompt, use the command `ipconfig` to find the IP address.

```
Ethernet adapter Ethernet 4:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::dc2a:9b1:5fc0:e744%21
   IPv4 Address. . . . . . . . . . . : 192.168.0.66
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1
```

Figure 5: Check IP Address

NOTE
The IP address can be different depending on the home AP settings.

For stable iperf3 testing, run the Windows Security APP to turn off the network firewall. Renesas recommends to disable all network firewalls on the host PC before running the test.

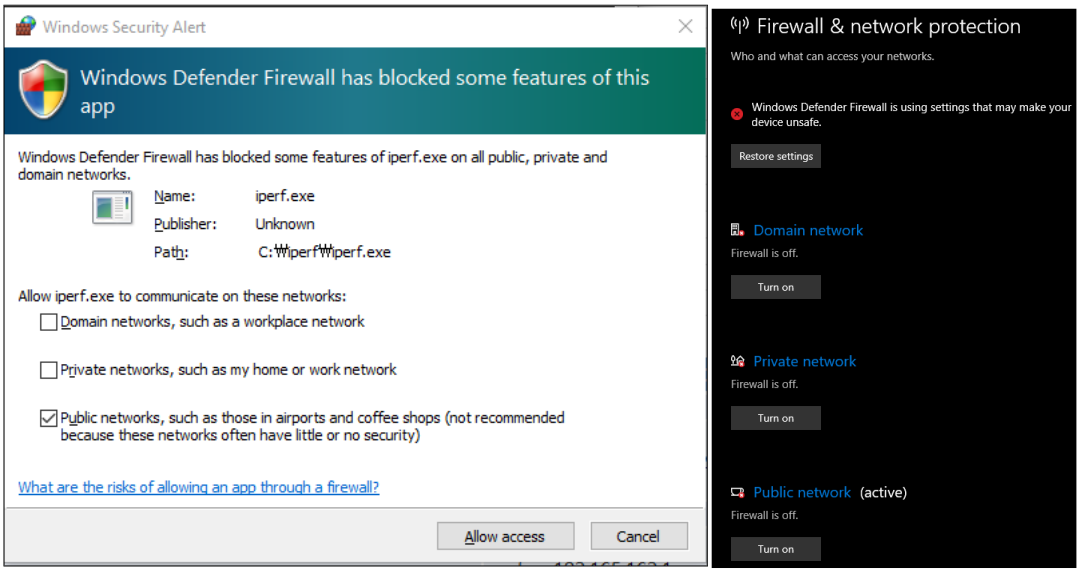

Figure 6: Disable Firewall for iperf3 Test

3. In the command prompt, move to the directory where iperf3 is installed, and type `iperf3 -s -i1` to configure the TCP server.

```
$ iperf3 -s -i1
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

Figure 7: Run iperf3 Server on Host PC

NOTE
When the message appears as shown in Figure 7, the iperf3 test is ready to start.

4. In the RZG2L+DA16200 console window, run the iperf3 test with Client mode.
   - For example: `[/DA16200/NET] #iperf3 -c 10.0.0.1 -t 10 -i 1`
     - The format of the command type is:
       `iperf3 -c [DESTINATION IP] -i [INTERVAL TIME] -t [TEST TIME]`

```
root@smarc-rzg2l:~# iperf3 -c 192.168.0.245 -t 10 -i 1
Connecting to host 192.168.0.245, port 5201
[  5] local 192.168.0.98 port 34946 connected to 192.168.0.245 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  1.13 MBytes  9.45 Mbits/sec    4   37.1 KBytes
[  5]   1.00-2.00   sec  1.07 MBytes  8.95 Mbits/sec    0   54.2 KBytes
[  5]   2.00-3.00   sec  1.16 MBytes  9.76 Mbits/sec    0   65.6 KBytes
[  5]   3.00-4.00   sec  1.10 MBytes  9.25 Mbits/sec    0   72.7 KBytes
[  5]   4.00-5.00   sec  1.10 MBytes  9.25 Mbits/sec    0   78.4 KBytes
[  5]   5.00-6.00   sec  1004 KBytes  8.22 Mbits/sec   28   9.98 KBytes
[  5]   6.00-7.00   sec   878 KBytes  7.19 Mbits/sec   24   8.55 KBytes
[  5]   7.00-8.00   sec  1004 KBytes  8.22 Mbits/sec   13   25.7 KBytes
[  5]   8.00-9.00   sec  1004 KBytes  8.22 Mbits/sec   25   8.55 KBytes
[  5]   9.00-10.00  sec  1004 KBytes  8.22 Mbits/sec   11   28.5 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  10.3 MBytes  8.67 Mbits/sec  105             sender
[  5]   0.00-10.00  sec  10.1 MBytes  8.51 Mbits/sec                  receiver

iperf Done.
```

Figure 8: Iperf3 Test Result on TCP Tx

5. In the command prompt, move to the directory where iperf3 is installed, and type `iperf3 -s -i1` to configure the UDP server.

6. In the RZG2L+DA16200 console window, run the iperf3 test in Client mode.
   - For example: `[/DA16200/NET] #iperf3 -c 10.0.0.1 -t 10 -i 1 -u -b 20m`
     - The format of the command is:
       `iperf3 -c [DESTINATION IP] -i [INTERVAL TIME] -t [TEST TIME] -u -b [bandwidth]`

```
root@smarc-rzg2l:~# iperf3 -c 192.168.0.245 -t 10 -i 1 -u -b 20m
Connecting to host 192.168.0.245, port 5201
[  5] local 192.168.0.98 port 55792 connected to 192.168.0.245 port 5201
[ ID] Interval           Transfer     Bitrate         Total Datagrams
[  5]   0.00-1.00   sec  1.26 MBytes  10.6 Mbits/sec  906
[  5]   1.00-2.00   sec  1.24 MBytes  10.4 Mbits/sec  892
[  5]   2.00-3.00   sec  1.26 MBytes  10.6 Mbits/sec  904
[  5]   3.00-4.00   sec  1.25 MBytes  10.5 Mbits/sec  897
[  5]   4.00-5.00   sec  1.24 MBytes  10.4 Mbits/sec  891
[  5]   5.00-6.00   sec  1.24 MBytes  10.4 Mbits/sec  888
[  5]   6.00-7.00   sec  1.25 MBytes  10.5 Mbits/sec  900
[  5]   7.00-8.00   sec  1.24 MBytes  10.4 Mbits/sec  892
[  5]   8.00-9.00   sec  1.24 MBytes  10.4 Mbits/sec  894
[  5]   9.00-10.00  sec  1.26 MBytes  10.6 Mbits/sec  904
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  5]   0.00-10.00  sec  12.5 MBytes  10.5 Mbits/sec  0.000 ms  0/8968 (0%)  sender
[  5]   0.00-10.00  sec  12.5 MBytes  10.5 Mbits/sec  1.876 ms  12/8967 (0.13%)  receiver

iperf Done.
```

Figure 9: Iperf3 Test Result on UDP Tx

4.12.3 iperf3 Test with Server Mode

For iperf3, there is no need to run RZG2L+DA16200 in the same server mode as iperf2. TCP or UDP Rx tests can be conducted by adding the -R option from the client's point of view while maintaining the server mode operated by the host PC.

1. In the RZG2L+DA16200 console window, run the iperf3 test in TCP Client mode.
   - For example: `[/DA16200/NET] #iperf3 -c 10.0.0.2 -i 1 -t 10 -R`

```
root@smarc-rzg2l:~# iperf3 -c 192.168.0.245 -t 10 -i 1 -R
Connecting to host 192.168.0.245, port 5201
Reverse mode, remote host 192.168.0.245 is sending
[  5] local 192.168.0.98 port 34956 connected to 192.168.0.245 port 5201
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-1.00   sec  1.26 MBytes  10.6 Mbits/sec
[  5]   1.00-2.00   sec  1.31 MBytes  11.0 Mbits/sec
[  5]   2.00-3.00   sec  1.30 MBytes  10.9 Mbits/sec
[  5]   3.00-4.00   sec  1.25 MBytes  10.5 Mbits/sec
[  5]   4.00-5.00   sec  1.21 MBytes  10.2 Mbits/sec
[  5]   5.00-6.00   sec  1.21 MBytes  10.2 Mbits/sec
[  5]   6.00-7.00   sec  1.21 MBytes  10.2 Mbits/sec
[  5]   7.00-8.00   sec  1.23 MBytes  10.4 Mbits/sec
[  5]   8.00-9.00   sec  1.23 MBytes  10.3 Mbits/sec
[  5]   9.00-10.00  sec  1.20 MBytes  10.1 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-10.00  sec  12.6 MBytes  10.6 Mbits/sec                  sender
[  5]   0.00-10.00  sec  12.4 MBytes  10.4 Mbits/sec                  receiver

iperf Done.
```

Figure 10: Iperf3 Test Result on TCP Rx

2. In the RZG2L+DA16200 console window, run the iperf3 test in UDP Client mode.
   - For example: `[/DA16200/NET] #iperf3 -c 10.0.0.2 -i 1 -t 10 -u -b 20m -R`

```
root@smarc-rzg2l:~# iperf3 -c 192.168.0.245 -t 10 -i 1 -u -b 20m -R
Connecting to host 192.168.0.245, port 5201
Reverse mode, remote host 192.168.0.245 is sending
[  5] local 192.168.0.98 port 56151 connected to 192.168.0.245 port 5201
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  5]   0.00-1.00   sec  1.29 MBytes  10.8 Mbits/sec  0.982 ms  0/923 (0%)
[  5]   1.00-2.00   sec  1.55 MBytes  13.0 Mbits/sec  0.810 ms  0/1110 (0%)
[  5]   2.00-3.00   sec  1.55 MBytes  13.0 Mbits/sec  0.792 ms  214/1325 (16%)
[  5]   3.00-4.00   sec  1.53 MBytes  12.9 Mbits/sec  0.813 ms  606/1707 (36%)
[  5]   4.00-5.00   sec  1.55 MBytes  13.0 Mbits/sec  0.784 ms  603/1719 (35%)
[  5]   5.00-6.00   sec  1.56 MBytes  13.0 Mbits/sec  1.052 ms  595/1712 (35%)
[  5]   6.00-7.00   sec  1.54 MBytes  13.0 Mbits/sec  1.118 ms  600/1709 (35%)
[  5]   7.00-8.00   sec  1.50 MBytes  12.6 Mbits/sec  1.064 ms  602/1679 (36%)
[  5]   8.00-9.00   sec  1.53 MBytes  12.8 Mbits/sec  0.889 ms  637/1736 (37%)
[  5]   9.00-10.00  sec  1.58 MBytes  13.2 Mbits/sec  0.767 ms  610/1742 (35%)
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Los[  479.134675] queue idx 3, cfm_status.ac 1
t/Total Datagrams
[  5]   0.00-10.00  sec  23.9 MBytes  20.0 Mbits/sec  0.000 ms  0/15362 (0%)  sender
[  5]   0.00-10.00  sec  15.2 MBytes  12.7 Mbits/sec  0.767 ms  4467/15362 (29%)  receiver

iperf Done.
```

Figure 11: Iperf3 Test Result on UDP Rx

4.13 Supported Commands for RF Tests

This section describes how to conduct RF tests in Linux driver along with available commands. The software package is provided so that users can build their products. To use RF test commands, enable the nl80211 testmode command in the kernel config. See Figure 3 for how to enable the command.

RF test can be done with the following command format:

- rf [interface] [command] <params>
- Interface: The network interface such as wlan0
- command: The command for RF test
- params: a parameter or parameters for the command

Available commands and parameters are described in Table 5 and Table 6.

Table 5: Common Command List

Command	Parameters	Description
rd	<reg>	Read digital registers in DA16x00
	Example Type ./rf wlan0 rd 60b000a0 when read 0xe8460 address Then get value 0x00000000 root@smarc-rzg2l:~# ./rf wlan0 rd e8460 ### do_read, argc = 1 ### do_read, argv[0] = e8460 ### do_read, addr = 000e8460 Reg value = 0x00000000	
ird		Read RF registers in DA16x00
	Example Type ./rf wlan0 ird f8 when read 0x7 address Then get value 0x00000100 root@smarc-rzg2l:~# ./rf wlan0 ird 7 Reg value = 0x00000100	
wr	<reg> <value>	Write value at digital registers in DA16x00
	Example Type ./rf wlan0 wr e8460 aa55aa55 when write 0xaa55aa55 value to 0xe8460 address Then read 0xe8460 to check write value root@smarc-rzg2l:~# ./rf wlan0 wr e8460 aa55aa55 root@smarc-rzg2l:~# ./rf wlan0 rd e8460 ### do_read, argc = 1 ### do_read, argv[0] = e8460 ### do_read, addr = 000e8460 Reg value = 0xaa55aa55	
iwr	<reg> <value>	Write value at RF registers in DA16x00
	Example Type ./rf wlan0 iwr 7 aa55 when write 0xaa55 value to 0x7 address Then read 0x7 to check write value root@smarc-rzg2l:~# ./rf wlan0 iwr 7 aa55 root@smarc-rzg2l:~# ./rf wlan0 ird 7 Reg value = 0x0000aa55	

Table 6: RF Test Command List

RF Command	Parameters	Description
tx	<Ch> <BW> <numFrames> <frameLen> <txRate> <txPower> <destAddr> <bssid> <htEnable> <GI> <greenField> <preambleType> <qosEnable> <ackPolicy>	Start RF Tx test. <Ch>: Carriere frequency (2412 ~ 2484MHz) <BW>: [0]: Fixed. Carrier bandwidth. 20 MHz fixed. <numFrames>: Number of frames to transmit. <frameLen>: Length of frame (bytes). <txRate>: Data rate b1: 11b DSSS 1 Mbps b2: 11b DSSS 2 Mbps b5_5: 11b DSSS 5.5 Mbps b11: 11b DSSS 11 Mbps g6: 11g 6 Mbps

	<scramnler> <aifsnVal> <ant>	g9: 11g 9 Mbps g12: 11g 12 Mbps g18: 11g 18 Mbps g24: 11g 24 Mbps g36: 11g 36 Mbps g48: 11g 48 Mbps g54: 11g 54 Mbps n6_5: 11n 6.5 Mbps (7.2 Mbps @ Short GI) n13: 11n 13 Mbps (14.4 Mbps @ Short GI) n19_5: 11n 19.5 Mbps (21.7 Mbps @ Short GI) n26: 11n 26 Mbps (28.9 Mbps @ Short GI) n39: 11n 39 Mbps (43.3 Mbps @ Short GI) n52: 11n 52 Mbps (57.8 Mbps @ Short GI) n58_5: 11n 58.5 Mbps (65 Mbps @ Short GI) n65: 11n 65 Mbps (72.2 Mbps @ Short GI) <txPower>: TX power (0 ~ 11), 0.8 dB step. <destAddr>: MAC address to send packet. <bssid>: BSSID <htEnable>: N/A <GI>: [short|long]. Guad interval. 11n mode only. <greenField>: [on|off]. Set greenfield mode on/off. <preambleType>: [short|long]. Preamble type @ DSSS mode <qosEnable>: [on|off]. MAC header QoS control. <ackPolicy>: [NO|NORM|BA|CBA] <scramnler>: N/A <aifsnVal>: [0 ~ 15]. Indicate the AIFS in units of slots after SIFS that HW should wait for before starting backoff, for access category. <ant>: [0]. Fixed.
	Example packet mode Tx test with 2412MHz and power grade as 0 `root@smarc-rzg2l:~# ./rf wlan0 tx start 2412 0 1000 n65 0`	
tx	stop	Stop RF TX test.
	Example packet mode Tx stop test `root@smarc-rzg2l:~# ./rf wlan0 tx stop`	
cont	<txRate> <txPower> <Ch>	Start RF continuous TX test. <txRate>: Data rate. Refer to tx command. <txPower>: TX power (0 ~ 11), 0.8 dB step. <Ch>: Carrier frequency (2412 ~ 2484 MHz).
	Example continuous mode Tx test with 2412MHz and power grade as 0 `root@smarc-rzg2l:~# ./rf wlan0 cont start n65 0 2412`	
cont	stop	STOP RF continuous TX test
	Example continuous mode Tx stop test `root@smarc-rzg2l:~# ./rf wlan0 cont stop`	
cw	<txPower> <Ch>	Start RF CW test. <txPower>: TX power (0 ~ 11), 0.8 dB step. <Ch>: Carrier frequency (2412 ~ 2484 MHz).

	Example CW Tx test with 2412MHz and power grade as 0 `root@smarc-rzg2l:~# ./rf wlan0 cw start 0 2412`	
cw	stop	STOP RF continuous TX test
	Example CW Tx stop test `root@smarc-rzg2l:~# ./rf wlan0 cw stop`	
ch	ch	Change RF channel frequency
	Example Change channel frequency to 2412 `root@smarc-rzg2l:~# ./rf wlan0 ch 2412`	
per	start	Start RF RX PER test

	Example
	PER test with ./rf wlan0 per run command
	`root@smarc-rzg2l:~# ./rf wlan0 per run`
	`phyerroracc = 0 , phyerror = 0`
	`errored acc = 0 , errored pkts = 0`
	`rxovflowacc = 0 , rxovflow = 0`
	`passed acc = 58 , passed pkts = 20`
	`rcvd acc = 58 , rcvd total = 20`
	`vga_gain = 0 , lna_gain = 0`
	`per acc = 0.000000 per = 0.000000`	

per	stop	STOP RF RX PER test
	Example `Press any button to stop per test`	

4.14 Test Example in RZ/G2L Evaluation Board

4.14.1 Test Setup

RF test application is verified on RZ/G2L evaluation board and DA16200 board using SDIO interface.

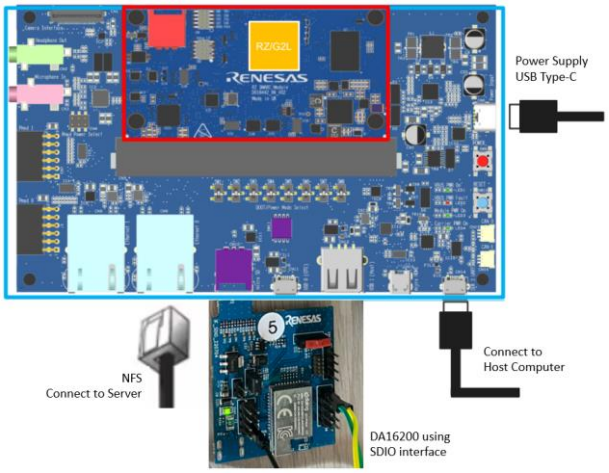

Figure 12: Test Setup

- Install driver module (insmod rswlan.ko) and set up the interface (ifconfig wlan0 up) in Linux host. See Figure 13 for details.

```
BSP: RZG2L/RZG2L-SMARC-EVK/3.0.1
LSI: RZG2L
Version: 3.0.1
smarc-rzg2l login: root
Last login: Tue Jan  3 04:48:24 UTC 2023
root@smarc-rzg2l:~# modprobe mac80211
[   37.364123] cfg80211: Loading compiled-in X.509 certificates for regulatory
 database
[   37.403383] cfg80211: Loaded X.509 cert 'sforshee: 00b28ddf47aef9cea7'
root@smarc-rzg2l:~# insmod rwnx_drv.ko
[   55.638027] rwnx v5.0.1.17 - build: Renesas Jan 31 2023 14:59:46
root@smarc-rzg2l:~# [   56.322721] mmc1: card 0001 removed
[   56.397556] mmc1: new high speed SDIO card at address 0001
[   56.521878] MACSW version 5.0.1.17
[   56.525316] ieee80211 phy0: RWNX_SDM (2)
[   56.530359] ieee80211 phy0: PHY features: [NSS=1][CHBW=20]
[   56.536442] ieee80211 phy0: FW features: [BCN][AUTOBCN][HWSCAN][CMON][MROLE
][PS][UAPSD][DPSM][AMPDU][CHNL_CTXT][P2P][P2P_GO]

root@smarc-rzg2l:~# lsmod
Module                  Size  Used by
rwnx_drv              245760  0
mac80211              610304  1 rwnx_drv
libarc4                16384  1 mac80211
cfg80211              335872  2 mac80211,rwnx_drv
vspm_if                49152  0
vspm                  102400  1 vspm_if
mmngrbuf               16384  0
mmngr                  24576  0
root@smarc-rzg2l:~# ifconfig wlan0 up
root@smarc-rzg2l:~#
```

Figure 13: Preparation for RF Test

- The help message for RF test application is shown in Figure 14.

```
root@smarc-rzg2l:~# ./rf wlan0
error! 0x3
USAGE:
        rf interface cmd [arg0] [arg1] ...
        interface ... Name of interface like wlan0
        cmd       ... RF command - rd, wr, ird, iwr, tx, cw, cont, ch & per
        [arg]     ... Optional parameters of RF command
root@smarc-rzg2l:~#
```

Figure 14: Help Message for RF Test

4.14.2 `tx` (Packet Mode)

This is the normal test mode with packet generation. The packet mode offers the possibility to adjust duty of RF Burst at time domain. The parameters for `tx` command can be shown with `./rf wlan0 tx` command.

```
root@smarc-rzg2l:~# ./rf wlan0 tx
tx start/stop [frequency] [numFrames] [frameLen] [txRate] [txPower] [addrDest]
 [bssid] [GI] [greenField] [preambleType] [qosEnable] [ackPolicy] [aifsnVal]
start/stop      ... Start or Stop RF TX test
[frequency]     ... Channel Frequency (2412 ~ 2472), default 2412
[numFrames]     ... Number of frames to transmit, default 100
[frameLen]      ... Length of frame (bytes), default 100
[txRate]        ... Data rate (b1 b2 b5_5 b11
                              g6 g9 g12 g18 g24 g36 g48 g54
                              n6_5 n13 n19_5 n26 n39 n52 n58_5 n65 ), default
b1
[txPower]       ... TX power (0 ~ 15) 0.8 dB step, default 0
[addrDest]      ... DEST. address (xx:xx:xx:xx:xx:xx), default 70:80:90:a0:b0:c
0
[bssid]         ... BSSID (xx:xx:xx:xx:xx:xx), default 10:20:30:40:50:60
[GI]            ... Guard interval 11n mode only (short|long), default short
[greenField]    ... Greenfield mode  (on/off), default off
[preambleType]  ... Preamble type @DSSS (short|long), default short
[qosEnable]     ... MAC header QoS control (on|off), default off
[ackPolicy]     ... Ack Policy (NO|NORM|BA|CBA), default NO
[aifsnVal]      ... Indicate the AIFS in units of slots after SIFS (0~15), defa
ult 1
root@smarc-rzg2l:~#
```

Figure 15: `tx` Command

4.14.2.1 `tx` Command Example

The command sends parameters such as channel1,1000 byte-packet, and power grade 0 with 802.11n MCS7.

- Figure 16 shows Tx start /stop using the following command
- `./rf wlan0 tx start 2412 0 1000 n65 0`
- `./rf wlan0 tx stop`

```
root@smarc-rzg2l:~# ./rf wlan0 tx start 2412 0 1000 n65 0
root@smarc-rzg2l:~# ./rf wlan0 tx stop
root@smarc-rzg2l:~#
```

Figure 16: `tx` Start/Stop Command Example

The parameters are optional. If skipped, default value will be used.

4.14.3 `cont`

This command sends TX continuously and is for TX power test. In this mode, TX packet is generated continuously with more than 95% duty cycle. The parameters for `cont` command can be shown with `./rf wlan0 cont` command.

```
root@smarc-rzg2l:~# ./rf wlan0 cont
cont start/stop [txRate] [txPower] [frequency]
start/stop    ... Start or Stop RF continuous TX
[txRate]      ... Data rate
[txPower]     ... TX power (0 ~ 15) 0.8 dB step
[frequency]   ... Channel Frequency (2412 ~ 2472)
root@smarc-rzg2l:~#
```

Figure 17: `cont` Command

4.14.3.1 `cont` Command Example

Test setting with 802.11n MCS7, channel 1 and power grade 0.

- Figure 18 shows Tx start/stop using the following command
- ./rf wlan0 cont start 2412 n65 500 1000 n65 0 2412
- ./rf wlan0 cont stop

```
root@smarc-rzg2l:~# ./rf wlan0 cont start n65 0 2412
root@smarc-rzg2l:~# ./rf wlan0 cont stop
root@smarc-rzg2l:~#
```

Figure 18: `cont` Start/Stop Command Example

The parameters are optional. If skipped, default value will be used.

4.14.4 `cw`

This command is for transmitting continuous wave (sine wave) to measure frequency error. The parameters for `cw` command can be shown with `./rf wlan0 cw` command.

```
root@smarc-rzg2l:~# ./rf wlan0 cw
cw start/stop [txPower] [frequency]
start/stop    ... Start or Stop RF CW test
[txPower]     ... TX power (0 ~ 15) 0.8 dB step
[frequency]   ... Channel Frequency (2412 ~ 2472)
root@smarc-rzg2l:~#
```

Figure 19: `cw` Command

4.14.4.1 `cw` Command Example

The command sends parameters of channel1 and power grade 0.

- Figure 20 shows Tx start/stop using the following command

- `./rf wlan0 cw start 0 2412`
- `./rf wlan0 cw stop`

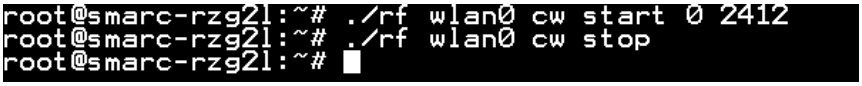

Figure 20: `cw` Start/Stop Command Example

The parameters are optional. If skipped, default value will be used.

4.14.5 `ch`

This command is for setting channel frequency. The parameters for `ch` command are shown in Figure 21.

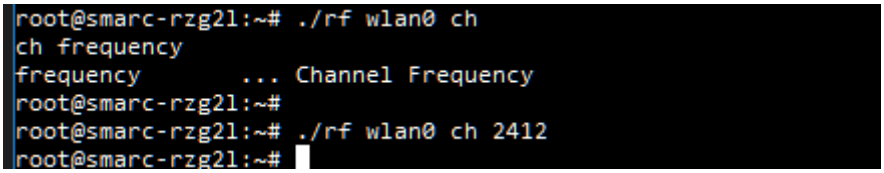

Figure 21: `ch` Command

4.14.6 `per`

This command is for measuring Packet Error Rate (PER). The parameters for `per` command are shown in Figure 22.

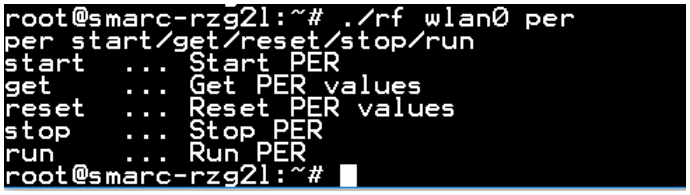

Figure 22: `per` Command

4.14.6.1 per Command Example

The `./rf wlan0 per run` command will display `per` information.

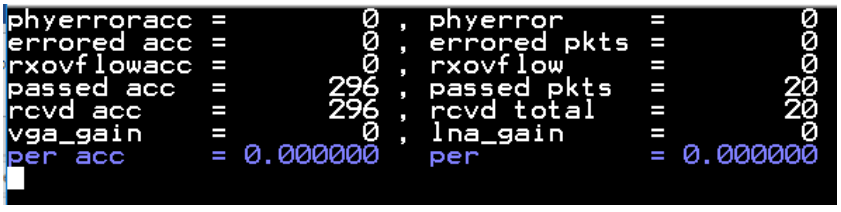

Figure 23: `per` Command Example

4.14.7 `rd` and `wr`

`rd` and `wr` commands are to read and write digital registers in DA16200 and DA16600.

4.14.7.1 rd and wr Command Example

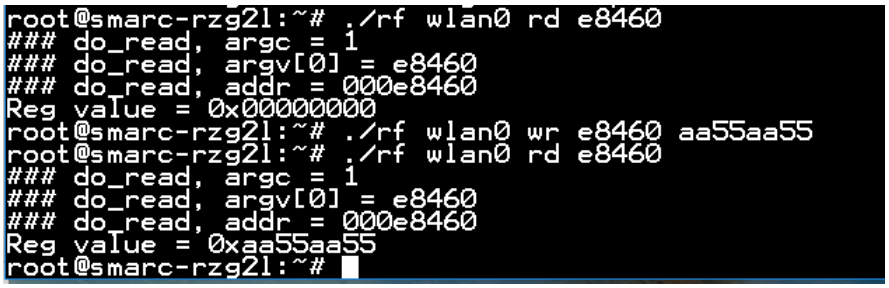

Figure 24: `rd` and `wr` Command Example

4.14.8 `ird` and `iwr`

The `ird` and `iwr` commands are to read and write RF registers in DA16200 and DA16600.

4.14.8.1 `ird` and `iwr` Command Example

Figure 25: `ird` and `iwr` Command Example

5 Linux Driver for BLE

5.1 RZ/G2L Linux System

5.1.1 SD Card

The Linux driver is based on the Renesas RZ/G2L Linux SD card image for Linux kernel and file systems. The SD card image is written to the SD card using the MS Windows “Win32 Disk Imager” tool.

5.1.2 NFS

Ethernet cable is required to use NFS mount function. The supported kernel and image were built using the RZ/G Verified Linux Package V3.0.3. This package can be downloaded from the Renesas website (https://www.renesas.com/us/en/products/microcontrollers-microprocessors/rz-mpus/rzg-linux-platform/rzg-marketplace/verified-linux-package/rzg-verified-linux-package).

5.2 Linux Driver

The meta-da16600 must be applied to the board support package (BSP). This meta-layer contains the patches required to allow Linux to deal with the DA16600. However, to get a design to use the DA16600, make sure the BSP provides the necessary changes to the device tree. The steps to apply the meta-layer into the BSP are as follows:

```
1. Extract meta-da16600.tar.gz from the device driver package into the BSP directory.
    $tar -zxvf meta-da16600.tar.gz -C /path/to/BSP directory

2. The below line should be added into build/conf/bblayers.conf of the BSP.
    ${TOPDIR}/../meta-da16600 \

3. The below lines should be added into build/conf/local.conf.
    MACHINE_FEATURES += "bluetooth"
    MACHINE_ESSENTIAL_EXTRA_RDEPENDS += " \
            linux-firmware-da14531 \
    "
    IMAGE_INSTALL_append = " kernel-modules bluez5"

4. Adding the bluetooth configuration to the device tree as shown in the example below.
   The pin assignment for the reset-gpios should be confirmed.

    In the file rzg2l-smarc.dtsi,

     /*
     * To enable SCIF2 (SER0) on PMOD1 (CN7)
     * SW1 should be at position 2->3 so that SER0_CTS# line is activated
     * SW2 should be at position 2->3 so that SER0_TX line is activated
     * SW3 should be at position 2->3 so that SER0_RX line is activated
     * SW4 should be at position 2->3 so that SER0_RTS# line is activated
     */
    #if PMOD1_SER0
    &scif2 {
            pinctrl-0 = <&scif2_pins>;
            pinctrl-names = "default";
            uart-has-rtscts;
            status = "okay";
            bluetooth {
                    compatible = "renesas,DA14531";
                    reset-gpios = <&pinctrl RZG2L_GPIO(47, 2) GPIO_ACTIVE_HIGH>;
```

```
            };
    };
    #endif
```

5.3 U-Boot Environment

5.3.1 SD Card

To boot from the microSD card on the SMARC carrier board, set the environment variables using the following commands at the U-boot prompt:

```
=> env default -a
=> setenv sd_boot1 'mmc dev 1 ; fatload mmc 1:1 0x48080000 Image ; fatload mmc 1:1
0x48000000 /Image-r9a07g044l2-smarc.dtb'
=> setenv sd_boot2 'setenv bootargs 'root=/dev/mmcblk1p2 rootwait' ; booti 0x48080000
- 0x48000000'
=> setenv bootcmd 'run sd_boot1 sd_boot2'
=> saveenv Saving Environment to MMC... Writing to MMC(0)….OK
```

5.3.2 NFS

To boot to NFS on the SMARC carrier board, set the environment variables using the following commands at the U-boot prompt:

```
=> setenv bootargsnfs 'setenv bootargs nfsrootdebug root=/dev/nfs rootfstype=nfs
ip=172.16.31.160:172.16.31.14:pi:255.255.224.0:pi:eth0:off
nfsroot=172.16.31.14:/nfsroot/rzg2lc,tcp,vers=3 vmalloc=384M'
=> setenv nfsload 'nfs 0x48080000 172.16.31.14:/nfsroot/rzg2lc/Image;nfs 0x48000000
172.16.31.14:/nfsroot/rzg2lc/Image-r9a07g044c2-smarc.dtb'
=> setenv serverip '172.16.31.14'
=> setenv ipaddr '172.16.31.160'
=> setenv netmask '255.255.224.0'
=> setenv ethact 'ethernet@11c20000'
=> setenv ethaddr '5A:54:E4:4D:03:9D'
=> setenv nfsboot 'run nfsload;run bootargsnfs;booti 0x48080000 - 0x48000000'
=> setenv bootcmd 'run nfsboot'
=> saveenv Saving Environment to MMC... Writing to MMC(0)….OK
```

The IP address, folder name, and ethaddr used above should be changed according to the user environment.

5.4 Serial Port Connection for HCI

The serial port connections between the DA16600 Module and the RZ/G2L are listed in Table 7.

Table 7: Serial Port Connection for HCI

DA16600 Module	Signal Name	RZ/G2L PMOD1 Pin Name	RZ/G2L PMOD1 Pin Number
P0_0	UART TX	SER0_RX	3
P0_1	UART RX	SER0_TX	2
P0_3	UART RTS	SER0_CTS	1
P0_4	UART CTS	SER0_RTS	4
P0_0	RST	GPIO13	10

5.5 Test

The device information can be found using the '`hciconfig`' command.

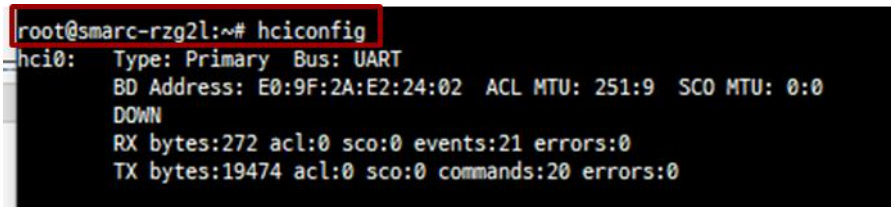

Figure 26: hiconfig

The scanned lists can be shown using the '`hcitool lescan`' command.

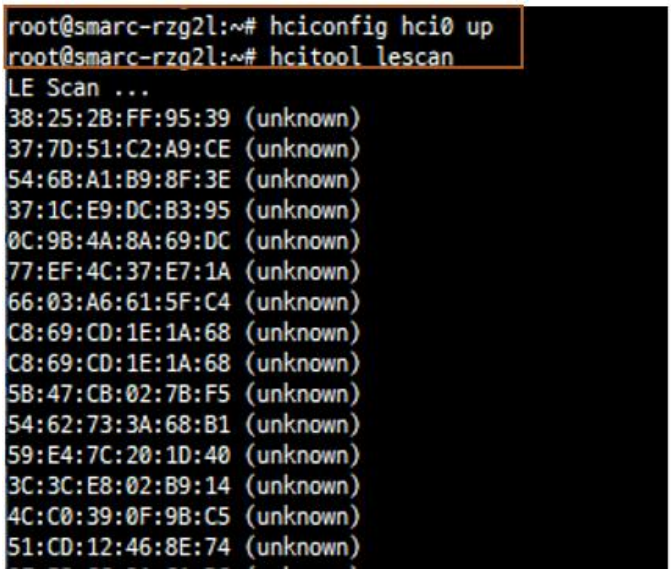

Figure 27: hcitool lescan Result

A connection with one of the peripheral devices in the scanned lists can be made using the '`hcitool lecc`' command.

Figure 28: Connection Result of hcitool lecc

Revision History

Revision	Date	Description
2.7	Mar. 13. 2024	● Revised STA/AP/Concurrent mode Linux network configure steps in Section 4.5.2.2 SDIO Interface
2.6	Nov. 08. 2023	Updated based on v5.2.x.x: ● Removed SPI IRQ1 Pin configuration because IRQ1 Pin not used ● Changed drwnx_drv to rswlan ● Added CONFIG_MODULE_TYPE for selecting module bus interface type sdio or spi
2.5	Oct. 27. 2023	● Changed how to download RZ/G Verified Linux Package v3.0.3 in 4.2.2 , 4.5.1 and 5.1.2
2.4	Aug. 18, 2023	● Updated SDIO GPIO Pin configuration for Reset pin in 4.7 ● Updated SDIO Linux kernel device tree for Reset pin in 4.7.1 ● Updated the description of section 5.2
2.3	Jul. 28, 2023	● Added ch command in Table 6 in Section 4.13 ● Added RF command (ch) in 4.14.5
2.2	Jun. 30, 2023	Changed the kernel path in Makefile in 4.5.1
2.1	Apr. 29, 2023	● Added separate firmware images for Host 32-bit and 64-bit ● Updated SPI GPIO Pin configuration
2.0	Apr. 14, 2023	Added how to set GPIO INPUT mode in 4.7.1
1.9	Mar. 30, 2023	● Added how to enable interface in use in 4.51 ● Added the section 4.13 and 4.14
1.8	Jan. 18, 2023	Changed the description of section 5.2 to add the meta layer for BLE
1.7	Nov. 25, 022	● Divided the firmware image to lmacfw_sdio.bin and lmacfw_spi.bin
1.6	Nov. 07, 2022	● Added AP/Concurrent mode execution method to SDIO ● Added AP mode execution method to SPI ● Changed image of Figure 3 ● Added method to set the interrupt and the reset gpio in dts
1.5	Oct. 06, 2022	● Added required preparations ● Added guide for over than kernel 5.10
1.4	Aug. 26, 2022	● Changed the compile module part to optional
1.3	Aug. 08, 2022	● Added HW Reset for SDIO interface
1.2	Jul. 05, 2022	● Changed iperf to iperf3 ● Added information about the SDIO interface ● Added information about the NFS
1.1	May 31, 2022	● Fixed GPIO notation ● Added BT COEX configuration ● Added figure captions
1.0	May 03, 2022	Initial release

Status Definitions

Status	Definition
DRAFT	The content of this document is under review and subject to formal approval, which may result in modifications or additions.
APPROVED or unmarked	The content of this document has been approved for publication.

RoHS Compliance

Renesas Electronics' suppliers certify that its products are in compliance with the requirements of Directive 2011/65/EU of the European Parliament on the restriction of the use of certain hazardous substances in electrical and electronic equipment. RoHS certificates from our suppliers are available on request.

Important Notice and Disclaimer

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu

Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:

https://www.renesas.com/contact/