Radicom Research, Inc.

Preliminary Designer's Guide for the

RA6000 Series

Universal Communication Module EVK

RA6000-EVK

RA6001-EVK

RA6002-EVK

RoHS Compliant

SDK Packages: RA6000-EVK.RRI.001
RA6001-EVK.RRI.001
RA6002-EVK.RRI.001

Oct 15, 2021

Table of Contents

Introduction

Thank you for purchasing Radicom RA6000 series Universal Communication Module Extension EVK. We are committed to providing you quality service and technical support.

Radicom RA6000 series provides a reference design from AI camera to wire Ethernet, wireless WiFi and Bluetooth communication. The Evaluation Kit integrates a powerful low power CPU and an ultra-low power camera module into WiFi, Ethernet LAN, Bluetooth modules. With many I/O ports, RA6000 series can support Ardunio I/O interface plus I2C, SPI and UARTs ports at plug and play level which is very easy for developers to expand new features and capability. Featuring self-contained communication subsystem, streamlined and simple connectivity and complete software development environment, the Evaluation Kit allows embedded system integrators to incorporate all communication plus camera and sensors technology into one develop platform that previously may need many different platforms.

The RA6000 series is designed to fully support Radicom many FCC/IC/CE certified communication modules for supporting short range Bluetooth wireless, middle range WiFi wireless, or with wire Ethernet to reach worldwide communications, which can save developers' time and certification efforts.

Depending on modules supported in the RA6000 series Evaluation Kit can be divided into the following categories

- RA6000-EVK
    - RA6000MB
    - RA6000C-M
    - RBW6100-RA

- RA6001-EVK
    - RA6000MB
    - RA6000C-M
    - RW8300M-a
    - RB8762C-RA

- RA6002-EVK
    - RA6000MB
    - RA6000C-M
    - RW8300M-a-B3

Features

- One Serial and one SPI interface ports to different modules
- Support Host WiFi, Embedded WiFi, Ethernet LAN, Bluetooth Low Energy dual mode modules, Contact ID SIA alarm analog modem and Arduino various interface cards
- Arduino interface
- JTAG or SPI programming

Approvals

- Module 1: RTL8711AM 802.11b/g/n
  FCC/IC/CE, TELEC, KC, SRRC, CB, NCC, AS/NZS
- Module 2: RW8300M series
  FCC/IC/CE, TELEC
- Module 3: RB8762CMF-a, RB8762C-RA
  FCC/IC/CE, TELEC

Specifications

Dimensions	RA6000 series EVK:102.2 x 100 x 59.8 mm [4.02” x 3.94” x 2.35”]
Device Type	RA6000-EVK/ RA6001-EVK/ RA6002-EVK
Interface	USB, I2C, LAN, UART, J-TAG, SPI, ADC, GPIO
Supply Voltage: VDD	+5.0V
Operating Temperature	-40℃ to +85℃
Environmental	RoHS 3 compliant

Recommended System Configuration

Supported System / Module	BLE	WiFi	Ethernet
RBW6100-RA	○	○	
RW8300M-a (WiFi/Ethernet/AP/switch/Router/repeater)		○	○
RW8300M-a-B3 (WiFi/BT/Ethernet/AP/switch/Router/repeater)	○	○	○
RB8762C-RA	○		

**Depending module availability, please contact manufacture*

Module Optional Supported Interface

	I2C	UART	SPI	USB
RBW6100-RA	○	○		
RW8300M-a		○		
RW8300M-a-B3				○
RB8762C-RA	○	○		

RA6000 series EVK Electronic Characteristics

Description				Minimum	Typical	Maximum	Unit
Category	Module	Interface	State				
RA6000 series EVK (without any module)			Power-On	150	180	210	mA
WiFi /Ethernet	RTL8711AM	UART	Power-On	135	140	145	mA
			Active	180	200	220	mA
	RW8300	UART	Power-On	160	180	200	mA
			Active	230	250	270	mA
Bluetooth	RB8762CMF-a	UART/I2C	Power-On	3	4	5	mA
			Active	15	25	30	mA

Model and Ordering Information

Model Number	Description
RA6000-EVK 	Universal Communication Module EVK. Model: RA6000-EVK Included accessories: 1ea ~ RA6000MB Board 1ea ~ RA6000C-M Board 1ea ~ RBW6100-RA module 1ea ~ 6 feet mini USB cable (A to mini B)
RA6001-EVK 	Universal Communication Module EVK. Model: RA6001-EVK Included accessories: 1ea ~ RA6000MB Board 1ea ~ RA6000C-M Board 1ea ~ RW8300M-a 1ea ~ RB8762C-RA 1ea ~ 6 feet mini USB cable (A to mini B)

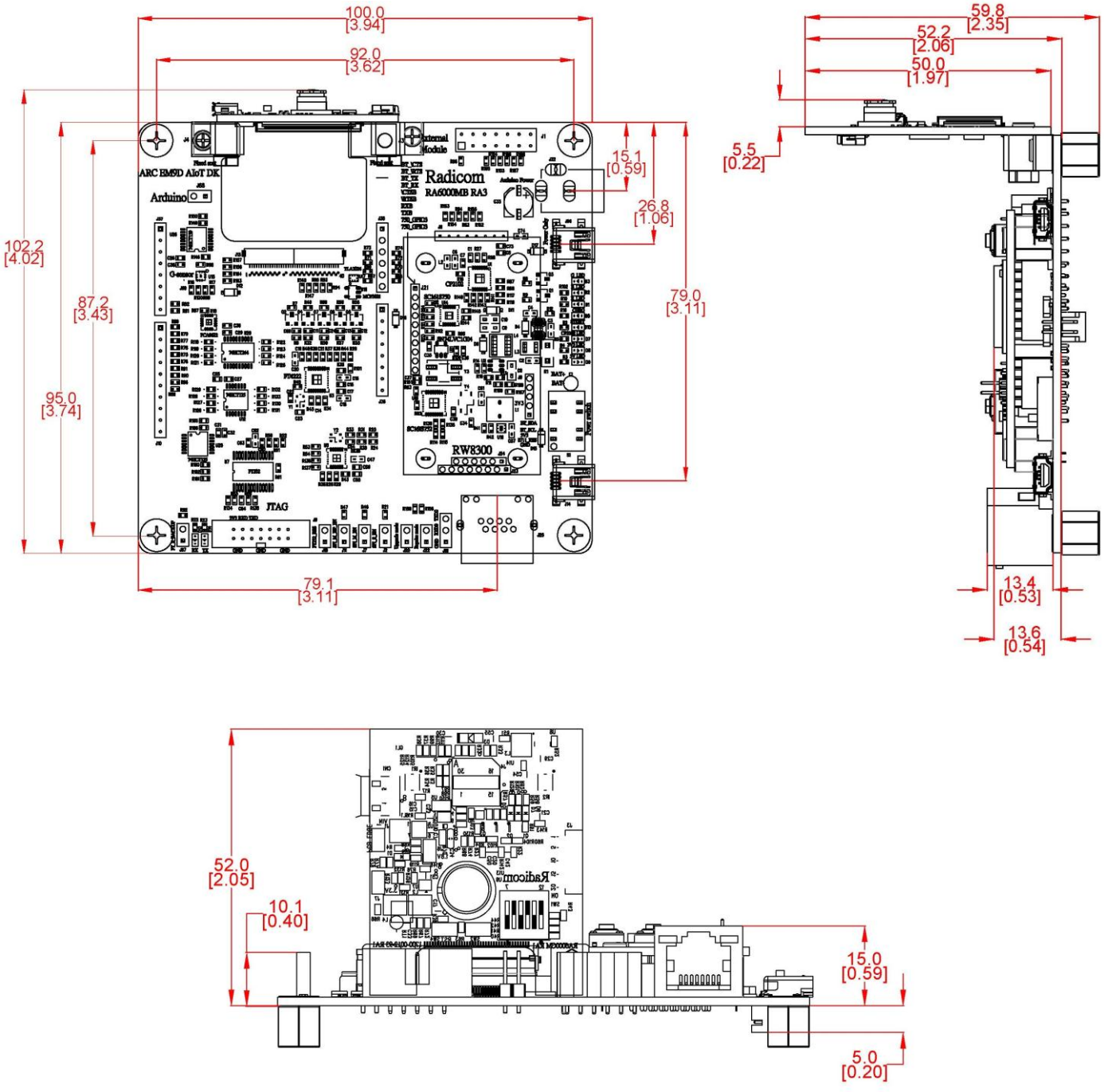
RA6002-EVK 	Universal Communication Module EVK. Model: RA6002-EVK Included accessories: 1ea ~ RA6000MB Board 1ea ~ RA6000C-M Board 1ea ~ RW8300M-a-B3 1ea ~ 6 feet mini USB cable (A to mini B)
RA6000MB 	MB board with through hole (DIP) pins for module Models: RA6000MB
RA6000C-M 	Camera module board Models: RA6000C-M
RBW6100-RA 	BT 5.1 + WiFi module Models: RBW6100-RA

RB8762C-RA 	RB8762C-RA Bluetooth Module added 2.54 Double rows pin header Models: RB8762C-RA
RW8300M-a 	Universal Adapter WiFi DIP Modules with antennas. Models: RW8300M-a
RW8300M-a-B3 	Universal Adapter WiFi/BT DIP Module Models with antennas Models: RW8300M-a-B3
Mini USB Cable 	Accessories- 6 feet USB A type to Mini USB B type cable

RJ-45 cable 	Accessories- RJ-45 Cat 5e Ethernet cable

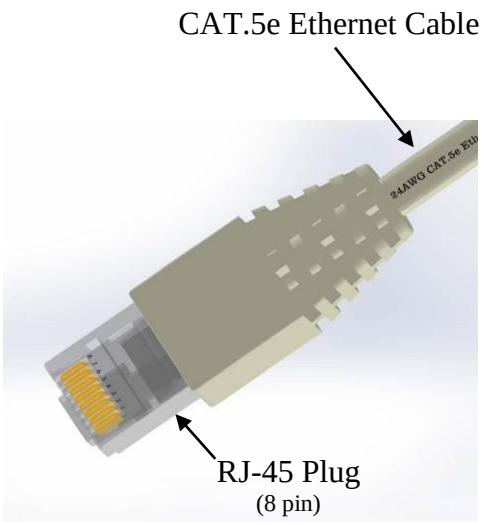
RA6000-EVK Mechanical Diagram

Top View: 102.2 x 100 x 59.8 mm [4.02” x 3.94” x 2.35”] Unit: mm[inch]



RA6000 series EVK RJ45 Ethernet Cable Description

The following table shows the RJ45 connector pin assignments for Ethernet Cable. PHY data rate is 100Mbps.

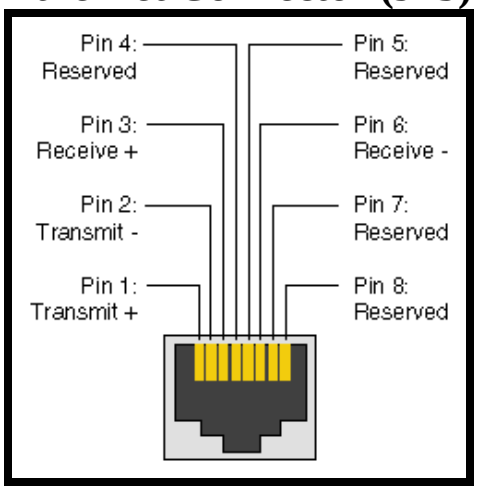


Contact	10/100 Base-T Signal
1	TD+ (Transmit Data)
2	TD- (Transmit Data)
3	RD+ (Receive Data)
4	Not used
5	Not used
6	RD- (Receive Data)
7	Not Used
8	Not Used

Standard RJ45 10/100 BASE-T Port

A RJ45 connector is used for Ethernet twisted pair links. The RJ45 connector has 8-pins, and may also be referred to as an **8-pin Modular Connector**. A male RJ45 plug is mounted on each end of the twisted pair cable. A female RJ45 Jack or Receptacle is integrated into the Ethernet hub or NIC.

Ethernet Connector (J25)



RA6000 series EVK System block

In order to increase I/O on system, Radicom provides a solution for controlling external I/O card and communication module. Increase 3 UART interface, 1 ADC interface, and some GPIOs through chipset. These chipsets use I2C or SPI to control. The system block diagram is following Figure 1.

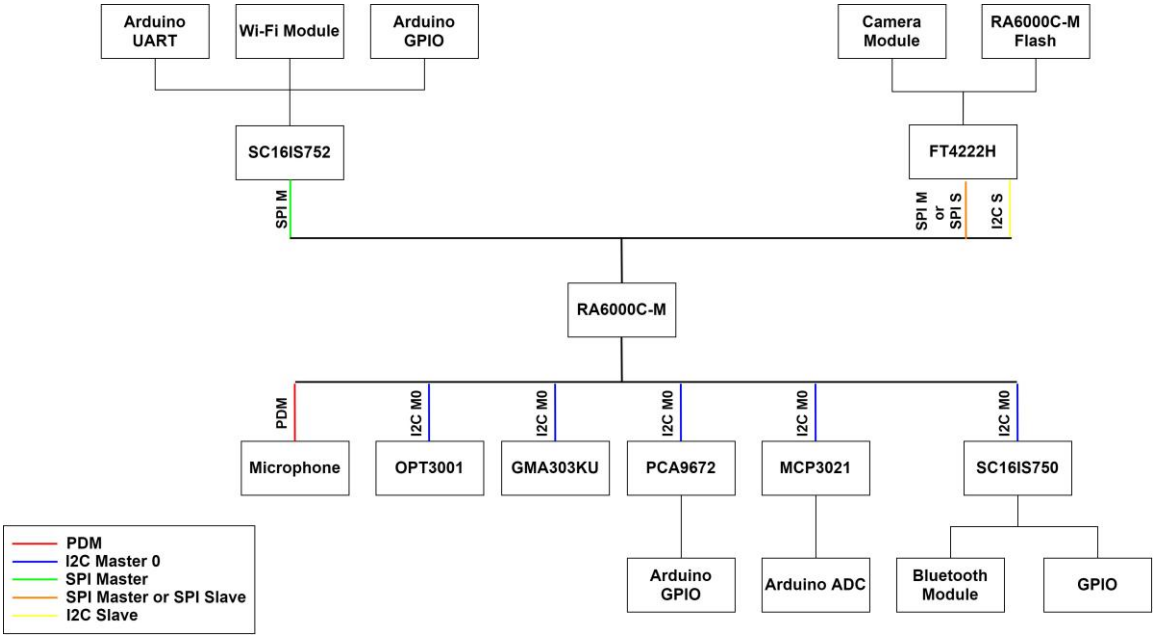


Figure 1 System block diagram

The control interfaces of chipsets are following.

Table 1 Chipset interface list

Chipset	Control interface	Output interface	Function	Note
OPT3001	RA6000C-M I2C Master 0	N/A	Ambient Light Sensor	
GMA303KU	RA6000C-M I2C Master 0	N/A	G Sensor	
PCA9672	RA6000C-M I2C Master 0	GPIO	I2C to GPIO.	For Arduino socket.
SC16IS750	RA6000C-M I2C Master 0	UART, GPIO	I2C to UART	For Bluetooth module.
SC16IS752	RA6000C-M SPI Master	UART, GPIO	SPI to UART	For Arduino UART socket and WiFi module.
FT4222H	1. RA6000C-M SPI Master or RA6000C-M SPI Slave 2. RA6000C-M I2C slave	N/A	1. SPI is for camera 2. I2C is for i2c command. (ex: OTA)	

RA6000C-M I2C master 0 connect to six devices, and I2C master 1 connect to Arduino socket only. I2C bus connection is following Figure 2. The Table 2 is device address list.
Note: MCP3021 device address is from vendor.

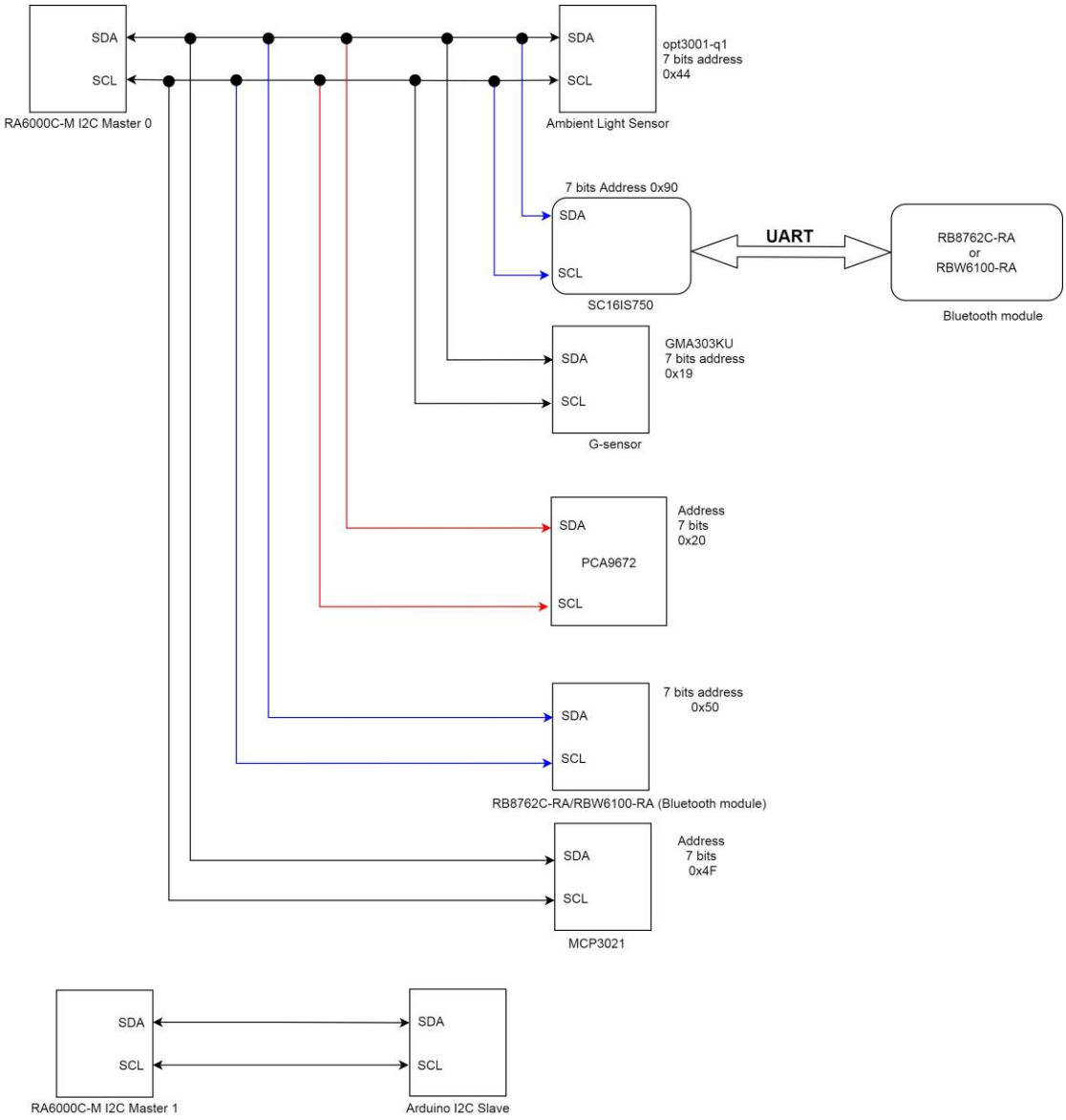


Figure 2 I2C bus connection

Table 2 I2C device address list

Device	I2C 7 bits Address	Note
opt3001	0x44	
SC16IS750	0x90	
GMA303KU	0x19	
PCA9672	0x20	
RB8762C-RA	0x50	
RBW6100-RA	0x50	Bluetooth module
MCP3021	0x4F	

RA6000C-M SPI master share to three devices. The FT4222H SPI modes are selected by jumper. SPI connection is following Figure 3.

Note: In this version, SPI clock limit on 10M Hz.

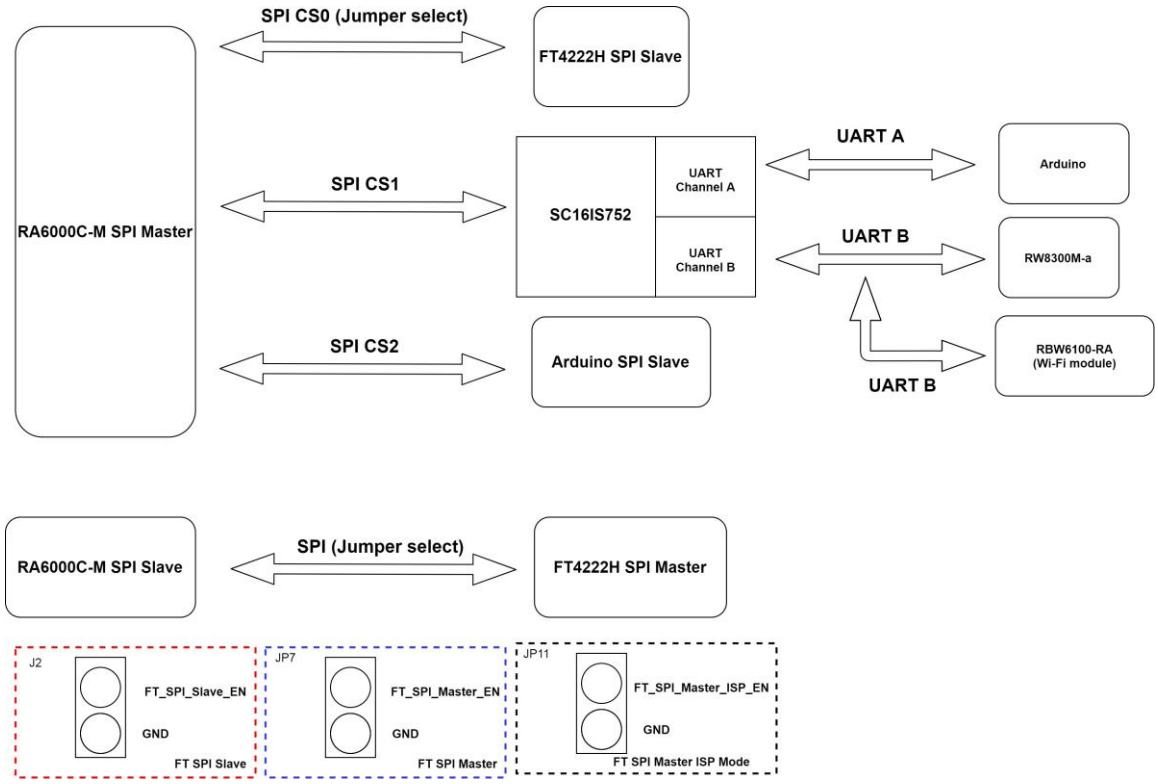


Figure 3 SPI bus connection

In this version, the user changes "chip select" by decoder. Decoder change SPI CS by PGPIO9 and PGPIO12 of RA6000C-M. Decoder state list is following Table 3.

Table 3 Decoder and chip select list

PGPIO9	PGPIO12	CS	Note
Low	Low	CS0	
High	Low	CS1	
Low	High	CS2	
High	High	CS0	

RA6000 series EVK USB Connection

RA6000 series EVK provides three USB devices. When user plug-in mini USB to PC/NB, the PC/NB will create two COM port and two interface A/B. These devices provide 2 UART interface, 1 I2C and 1 SPI interface. The UART interface connect to RA6000C-M UART0 and UART1, SPI interface connects to RA6000C-M for cream module, and I2C interface connect RA6000C-M I2C slave for I2C command. The USB block diagram is following Figure 4.

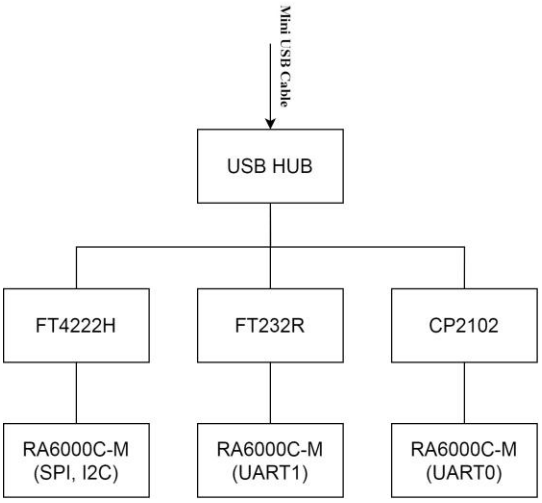


Figure 4 USB HUB block diagram

RA6000 series EVK provides two mini USB connector, one is connected to USB hub, the other provide external power input. The mini USB connector location is following Figure 5.

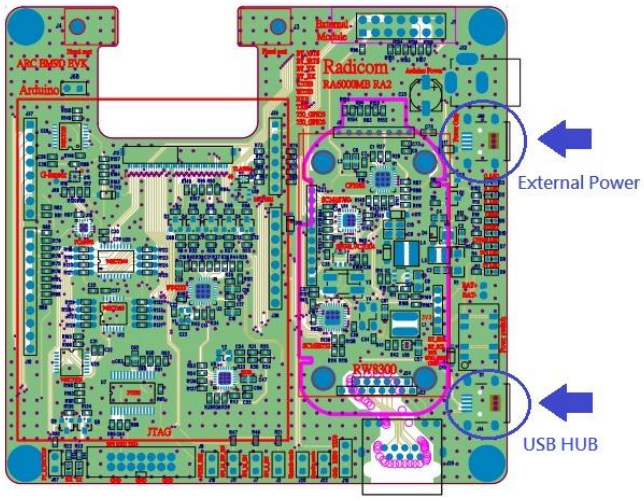


Figure 5 Mini USB connector Location

RA6000 series EVK FPC

RA6000MB connects to RA6000C-M by FFC (Flexible Flat Cable). They provide some interfaces connection for RA6000 series EVK from RA6000C-M, such as GPIO, SPI, I2C and UART. The FPC (Flex Printed Circuit) connector pin define of RA6000 series EVK is following Table 4.

Table 4 RA6000 series EVK FFC Pin define

Pin No.	Pin define	Pin No	Pin define
1	3.3V	21	I2C_MASTER_1_SCL
2	PGPIO6	22	I2C_MASTER_1_SDA
3	PSPI_SLV_SCLK	23	NC
4	PSPI_SLV_CS	24	I2C_MASTER_0_SCL
5	PSPI_SLV_SDI (MOSI)	25	I2C_MASTER_0_SDA
6	PSPI_SLV_SDO (MISO)	26	TESTMODE
7	SPI_MASTER_CS0	27	NC
8	SPI_MASTER_SDIO0 (MOSI)	28	NC
9	SPI_MASTER_SDIO1 (MISO)	29	NC
10	SPI_MASTER_SCK	30	NC
11	I2C_SLV_SDA	31	NC
12	I2C_SLV_SCL	32	NC
13	PGPIO4	33	PGPIO14
14	PGPIO5	34	BOOT_OPT0
15	GND	35	PGPIO8
16	UART1 TX	36	PGPIO9
17	UART1 RX	37	PGPIO12
18	GND	38	PGPIO13
19	UART0 TX	39	BOOT_OPT1
20	UART0 RX	40	BOOT_OPT2

Pin 3 to pin 6 is SPI slave, pin 7 to pin 10 is SPI master. Pin 26, pin 34, pin 39 and pin 40 are boot option pin.

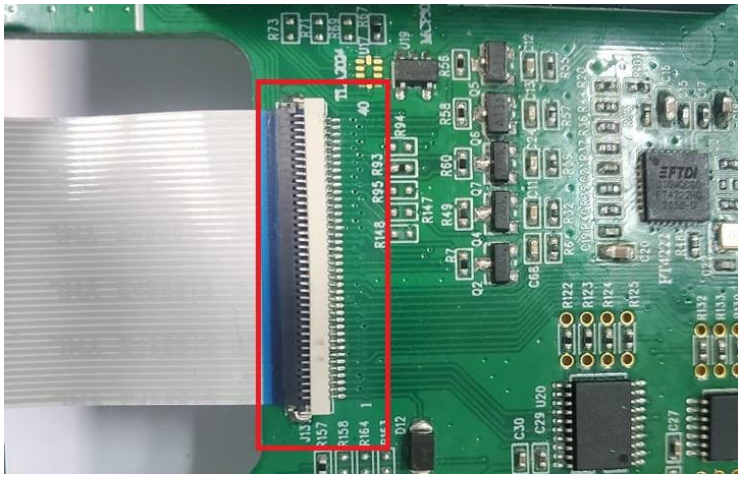


Figure 6 RA6000 series EVK FPC connector location

RA6000 series EVK GPIO

RA6000 series EVK has some GPIO from RA6000C-M, they through FFC connection. These GPIO are used for motion pin of camera module, resetting Arduino I/O chipset and connecting to Arduino I/O chipset interrupt pin. These GPIO define are following Table 5

Table 5 RA6000 series EVK GPIO define

GPIO	Connect to …	Descript
PGPIO0	AND Gate (U14)	
PGPIO1	NC	
PGPIO2	NC	
PGPIO3	Light sensor interrupt	
PGPIO4	FT4222H	For camera motion
PGPIO5	FT4222H	
PGPIO6	1. LED_R(RA6000C-M) 2. FT4222H	When work in ISP mode, it is chip select.
PGPIO7	NC	
PGPIO8	1. LED_G(RA6000C-M) 2. SC16IS752 RESET 3. SC16IS750 RESET 4. LED_R(RA6000MB)	For 752, 750 resetting. Low reset.
PGPIO9	1. LED_B(RA6000C-M) 2. 74HCT139 1A0 3. LED_B(RA6000MB)	For SPI master chip select
PGPIO10	NC	
PGPIO11	NC	
PGPIO12	1. 74HCT139 1A1 2. LED_G(RA6000MB)	For SPI master chip select
PGPIO13	SC16IS752 IRQ	For SC16IS752 interrupt pin
PGPIO14	SC16IS750 IRQ	For SC16IS750 interrupt pin

When user want to reset SC16IS752 and SC16IS750, user sets PGPIO8 to low. When SC16IS752 and SC16IS750 have interrupt, user can get interrupt state from PGPIO13, PGPIO14.

RA6000 series EVK has other GPIO from SC16IS750, SC16IS752 and PCA9672. These GPIO provide Arduino GPIO, RBW6100-RA WiFi reset pin, RBW6100-RA WiFi module GPIO, RB8762C-RA reset pin, PCA9672 interrupt pin and some backup GPIO pin. These GPIO define are following Table 6.

When user want to use Arduino GPIO, user can use 752 GPIO or PCA9672 GPIO by I2C or SPI. Please refer to P42 to use them.

Table 6 RA6000 series EVK other GPIO define

GPIO	Connect to …	Descript
9672 P0	Arduino	
9672 P1	Arduino	
9672 P2	Arduino	
9672 P3	Arduino	
9672 P4	Arduino	
9672 P5	Arduino	
9672 P6	NC	
9672 P7	NC	
750 GPIO0	NC	
750 GPIO1	NC	
750 GPIO2	RB8762C-RA RESET	For RB8762C-RA reset pin. Low reset.
750 GPIO3	RBW6100-RA WiFi GPIOC 4	For RBW6100-RA WiFi module GPIOC4
750 GPIO4	RBW6100-RA WiFi GPIOC 5	For RBW6100-RA WiFi module GPIOC5
750 GPIO5	RBW6100-RA WiFi Reset	For RBW6100-RA WiFi module reset pin. Low reset
750 GPIO6	J1 (External module)	
750 GPIO7	J1 (External module)	
752 GPIO0	Arduino	
752 GPIO1	Arduino	
752 GPIO2	1. Arduino 2. PCA9672 RESET	For PCA9672 resetting. Low reset.
752 GPIO3	PCA9672 INT	
752 GPIO4	NC	
752 GPIO5	Arduino	
752 GPIO6	Arduino	
752 GPIO7	Arduino	

RA6000 series EVK Arduino Pin Define

RA6000 series EVK provides similar Arduino socket for some external I/O card or module. The user can mount some I/O card or module to Arduino socket by oneself. The Arduino socket pins define is following Figure 7. RA6000MB provides a DC jack for Android Vin (7V). This pin is output pin to provide I/O card power (7V). Arduino Vin don't provide power (3.3V and 5V) to RA6000MB. The power (3.3V and 5V) of RA6000MB is from mini USB.

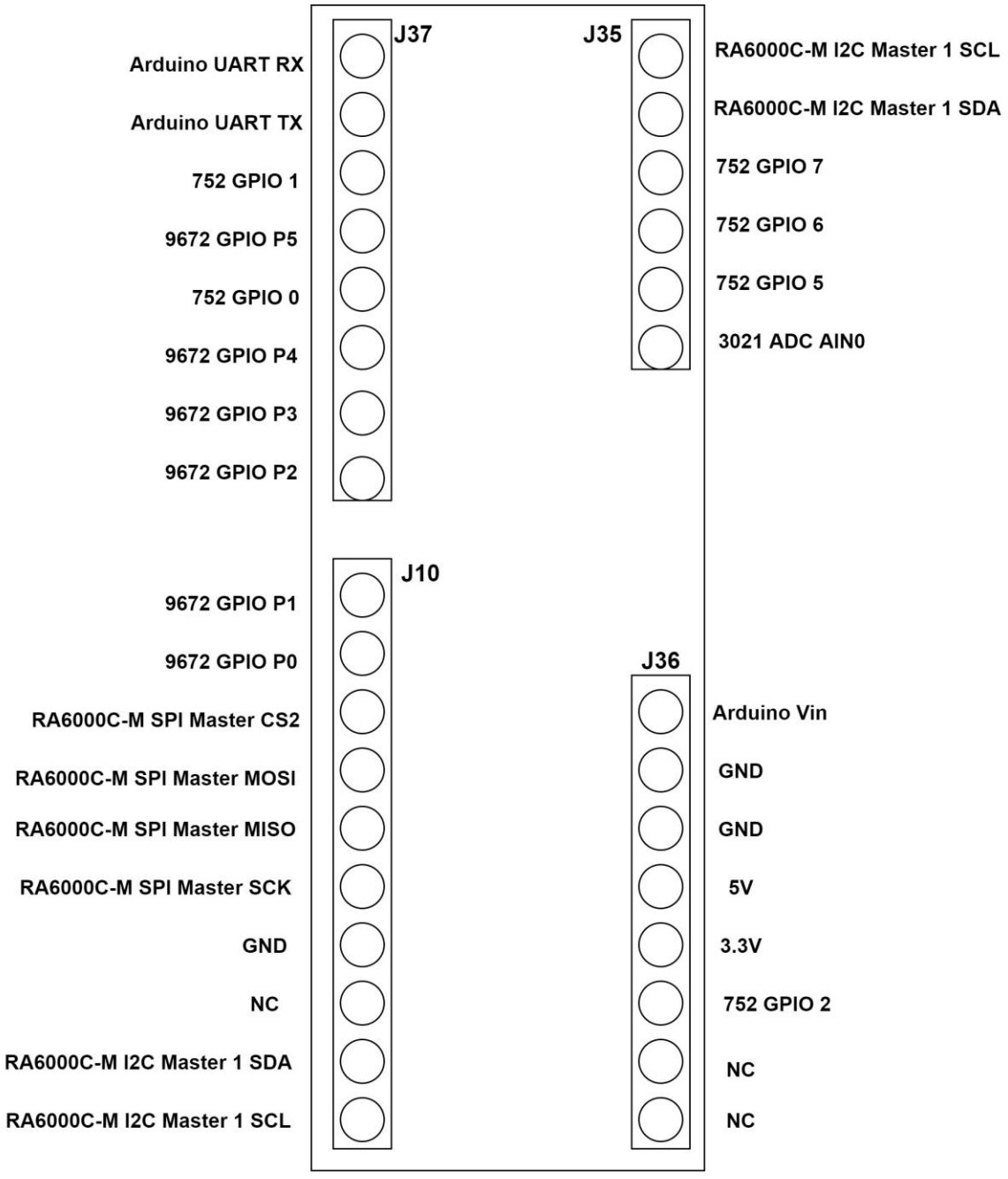


Figure 7 Arduino socket pin define

RA6000 series EVK LED Operation

RA6000 series EVK has ten LED to show some device state. This chapter will explain these LED states. There LED location is following Figure 8 to Figure 10.

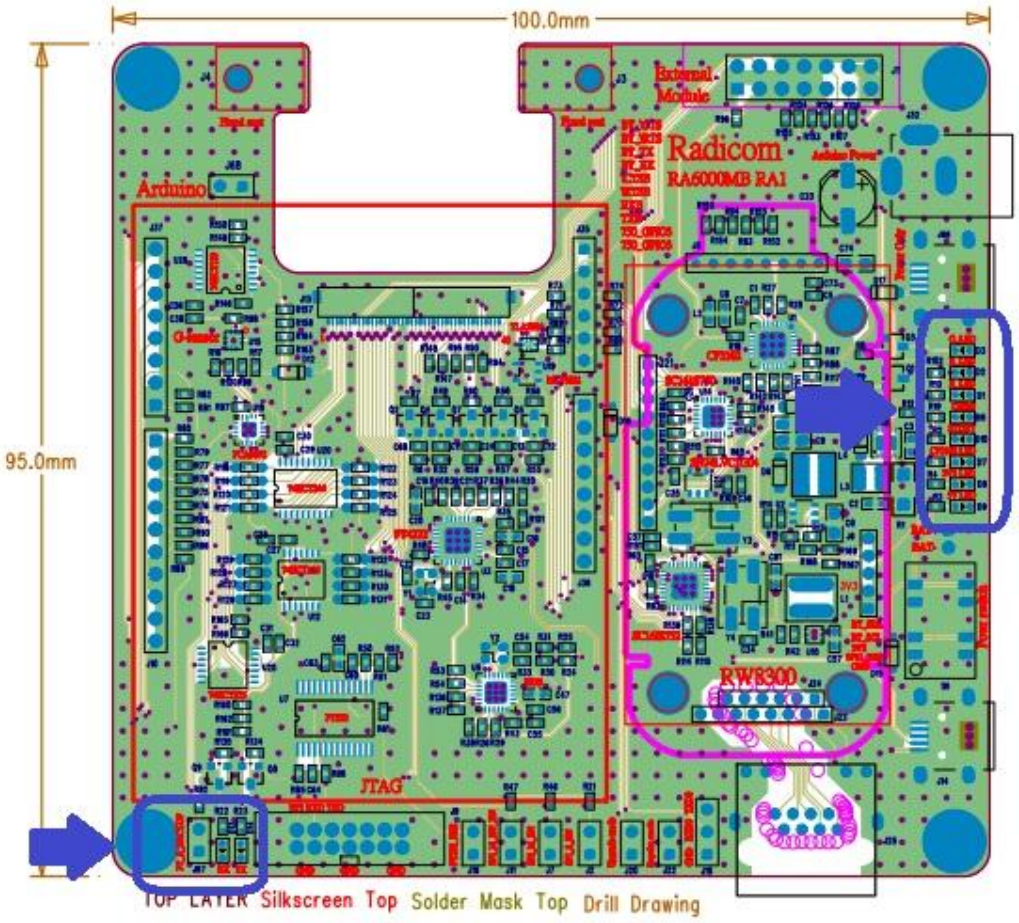


Figure 8 LED location-1

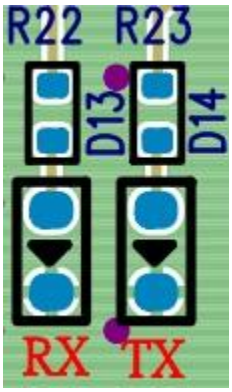


Figure 9 LED location-2

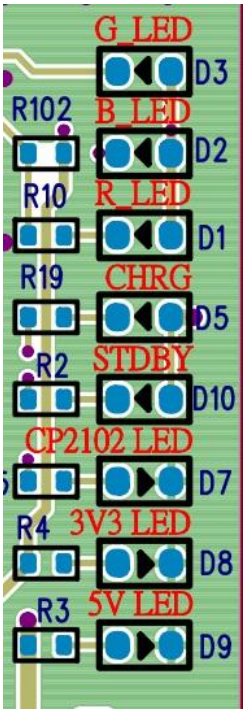


Figure 10 LED location-3

Table 7 Boot option switch setting list

LED	Color	Describe
D9 (5V LED)	Green	5V Power LED state.
D8 (3V3 LED)	Green	3.3V Power LED state.
D7 (CP2102LED)	Red	CP2102 Suspend state.
D10 (STDBY)	Blue	Battery is stand-by.
D15 (CHRG)	Red	Battery is charging.
D1 (R_LED)	Red	RA6000 series EVK PGPIO8 State
D2 (B_LED)	Blue	RA6000 series EVK PGPIO9 State
D3 (G_LED)	Green	RA6000 series EVK PGPIO12 State
D13 (RX)	Green	FT232R UART RX state
D14 (TX)	Green	FT232R UART TX state

RA6000 series EVK Jumper Setting

RA6000MB provides some jumpers to change mode or some backup solution. The jumpers are total eight. There are two jumpers to be backup solution, others are change mode. The jumper location is following Figure 11.

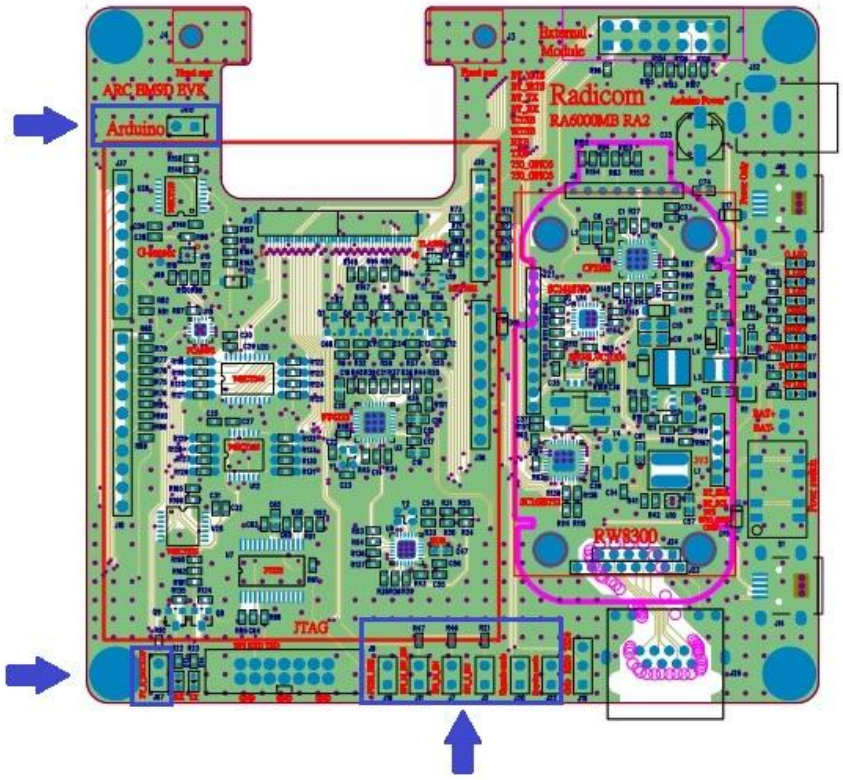


Figure 11 Jumper location

The Table 8 describes jumper function.

Table 8 Jumper setting list

Jumper	Describe
J68	Backup solution.
J67	Backup solution.
J19	For JTAG.
J11	SPI connection for upgrading firmware
J7	For connecting to FT4222H SPI master.
J2	For connecting to FT4222H SPI slave.
J20	For enabling upgrade mode.
J22	For enabling repair mode.

RA6000 series external module socket

RA6000 series EVK provides a socket for external module. This socket supports GPIO, UART and I2C. Radicom provides a RB8762C-RA BLE module for this socket. When others modules support this socket pin define, user can also mount others module by this socket. The socket pin define is following Table 9.

Figure 12 J1 location-1

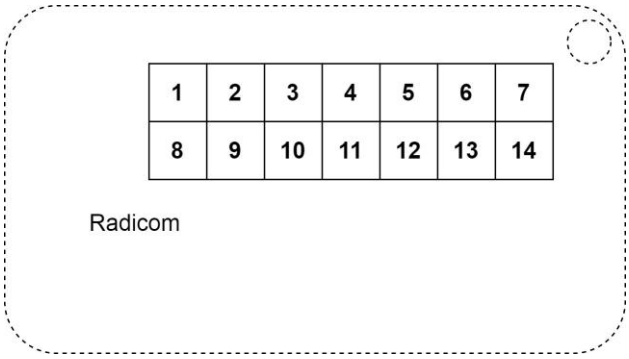


Figure 13 J1 location-2

Table 9 J1 pin define

Pin No.	Pin define	Pin No.	Pin define
1	750 UART CTS	8	3.3V
2	750 UART TX	9	750 GPIO2
3	750 UART RX	10	GND
4	750 UART RTS	11	GND
5	I2C M0 SDA	12	I2C M0 SCL
6	NC	13	NC
7	750 GPIO7	14	750 GPIO6

RA6000 series EVK Boot Option

The RA6000 series EVK boot options include upgrade mode, repair mode and normal mode. Radicom provides 2 method to change boot option. One reserves boot option switch of CPU main board from original design. The other easily changes boot option by jumper. This chapter will be shown how to change jumper to change boot option.

A. **Boot option switch from original design**

The boot option switch is on CPU main board, as Figure 14.

Figure 14 Boot switch location

Table 10 Boot option switch setting list

Boot Mode	Switch
Upgrade mode	Switch 1, 2, 4 keep on. Switch 3, 5, 6 keep off
Repair mode	Switch 2, 3 keep on. Switch 1, 4, 5, 6 keep off
Normal mode	All keep off

B. **Boot option jumper of EVK**

When user want to change boot option by boot option jumper, user need to check RA6000C-M boot option switch to keep normal mode (all off) first, as Figure 15. The location of boot option jumper is following Figure 16 and Figure 17

Figure 15 Normal mode

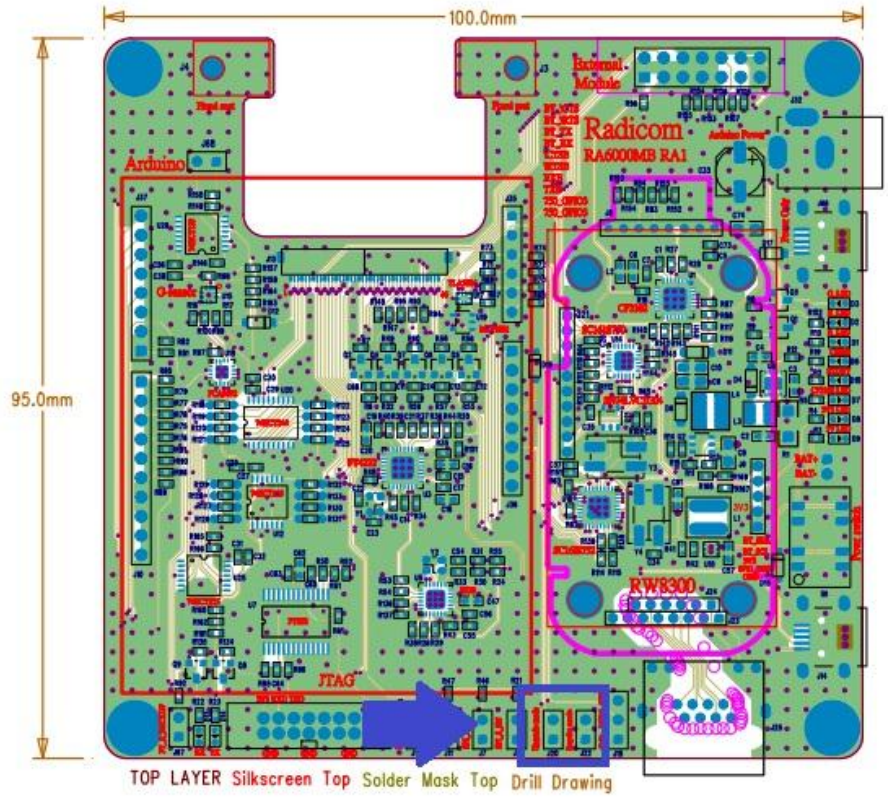


Figure 16 Jumper location-1

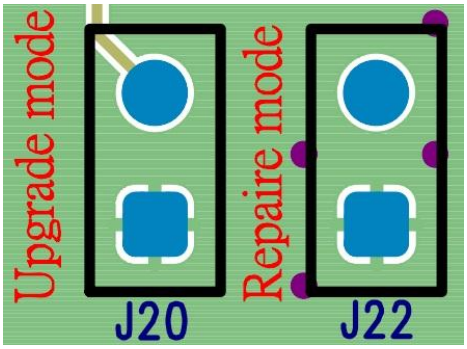


Figure 17 Jumper location-2

Table 11 Boot option jumper setting list

Boot Mode	Switch
Upgrade mode	J20 is shorted. J22 is opened.
Repair mode	J22 is shorted. J20 is opened.
Normal mode	J20 is opened. J22 is opened.

RA6000 series EVK Debug UART

RA6000 series EVK provides two methods about accessing debug UART. One use mini USB cable to access by CP2102, the other use USB to UART dongle to access. This chapter will be shown how to access debug UART.

A. **Through mini USB cable**

The PC/NB need to install CP2102 driver first. When user plug-in USB to PC/NB, the PC/NB will create a CP2102 COM port.
(download url: https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers)

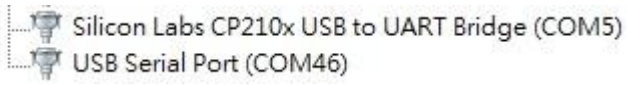


Figure 18 CP2102 COM port

B. **Through USB to UART dongle**

The PC/NB need to install driver for USB dongle first. When user plug-in USB dongle, the PC/NB will create a COM port. Please connect USB dongle to debug UART socket. The Debug UART socket location is following Figure 19.

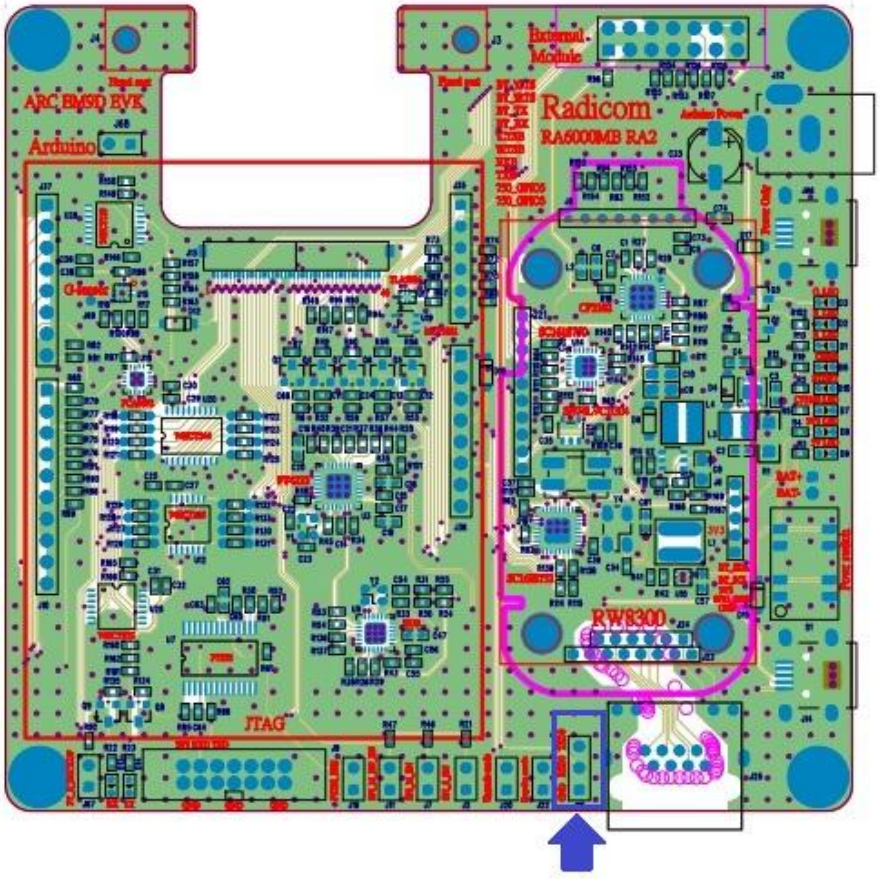


Figure 19 Debug UART socket location

The socket connection is following Figure 20

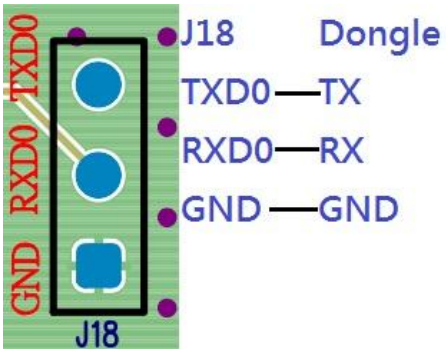


Figure 20 USB dongle connection

C. **How to access**

C.1 Windows

Please follow as below steps to access debug UART under windows system.
Step 1: Plug-in mini USB to PC/NB.

Step 2: Open a terminal application (ex: putty, teraterm…)

Step 3: Select COM port.

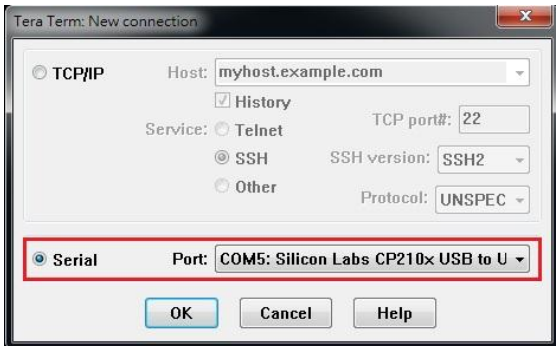


Figure 21 Teraterm select COM port

Step 4: Click "Setup > Serial port…" to open serial setting window.

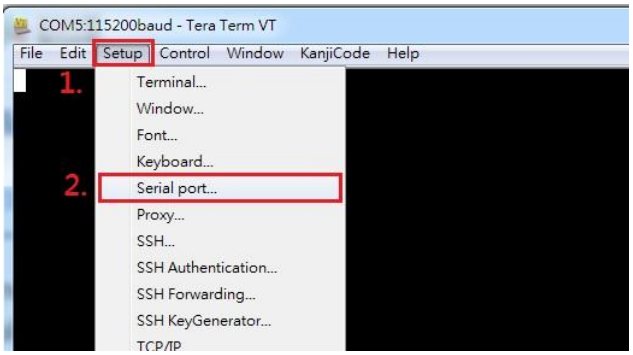


Figure 22 Click serial port setting

Step 5: Configure serial setting (115200, 8, n, 1), and click “OK” button.

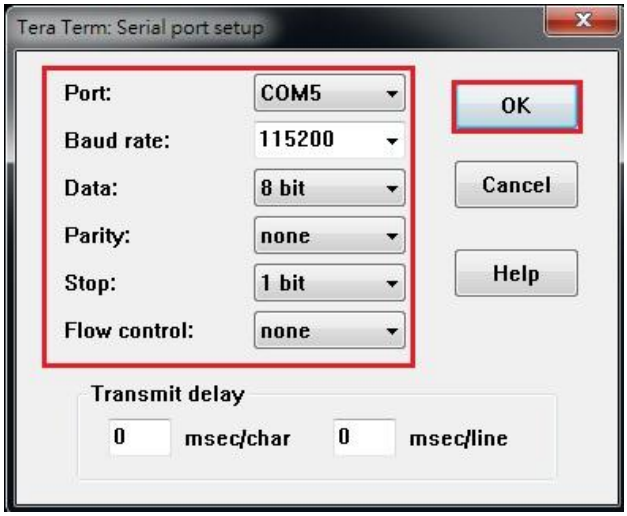


Figure 23 serial port setting

Step 5: Now, terminal application will show debug message.

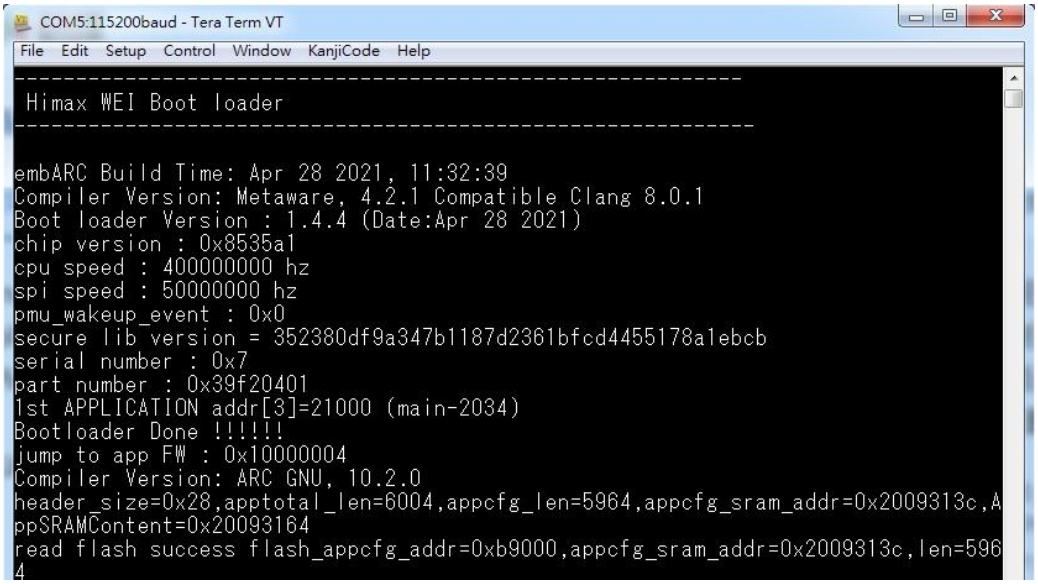


Figure 24 Debug message

C.1 Linux

Please follow as below steps to access debug UART under linux system.

Step 1: Plug-in mini USB to PC/NB

Step 2: Open a terminal, and open a serial application (ex: minicom).

Command: "**minicom -D /dev/ttyUSB0 -b 115200**"

Figure 25 "minicom" command

Step 3: type "ctrl + p" to check serial setting. (115200, 8, n, 1)

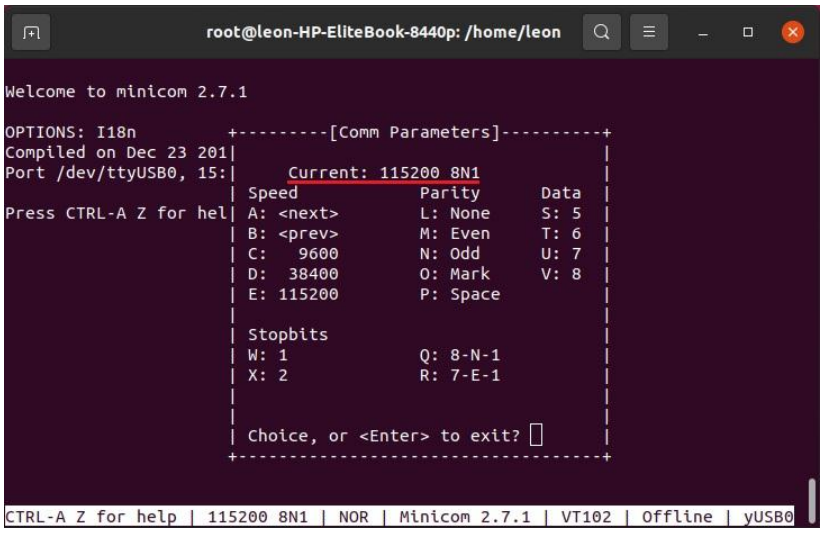


Figure 26 "minicom" serial setting

Step 4: Now, terminal application will show debug message.

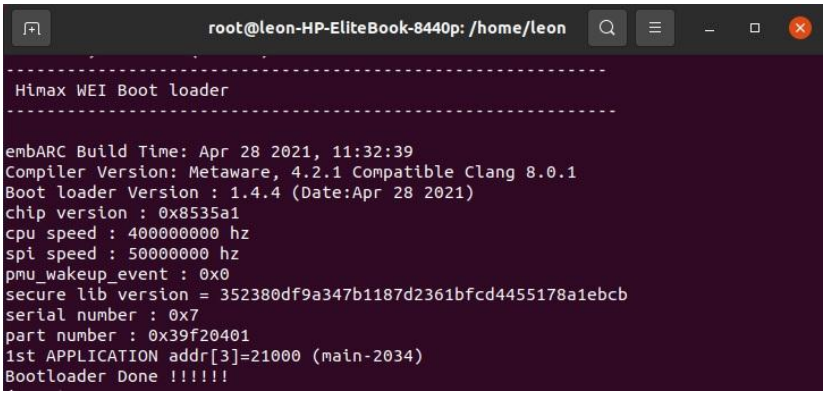


Figure 27 Debug message

RBW6100-RA Product Introduction

RBW6100-RA combine BLE with WiFi, it includes a RB8762CMF-a (BLE) module and a RTL8711AM (WiFi) module. The user can control these modules by UART AT command. RBW6100-RA has two buttons for resetting RTL8711AM and RB8762CMF-a.

Figure 28 RBW6100-RA hardware introduction

RTL8711AM can be AP router or station role, RTL8711AM is changed role by AT command. RTL8711AM provide some commands to open TCP server and TCP client. RTL8711AM can simply send and receive data with TCP server or client.

RB8762CMF-a can be BLE master or BLE slave role, RB8762CMF-a is changed role by AT command. RB8762CMF-a default is data passthrough mode. RB8762CMF-a builds connection with external BLE master and slave. The RB8762CMF-a can simply send and receive data with external BLE device.

RBW6100-RA AT Command

RBW6100-RA module combines BT and WiFi function. it includes a RB8762CMF-a module (BLE) and a RTL8711AM module (WiFi). This chapter will be shown these modules AT command.

A. **RTL8711AM AT command list**

RTL8711AM provides to change serial parameters, network setting and network function by AT command. The AT command list is following Table 12

Table 12 RTL8711AM AT command list

AT Command	Description
SYSTEM	
AT	**Test AT command ready**
ATS?	**Show all AT command**
ATSR	**Restart module**
ATSV	**Show firmware version**
ATSE	**Set AT command echo mode**
ATSY	**Factory Reset**
ATSU	**UART configuration**
ATSU?	**Show UART configuration**
ATSO	**OTA upgrade**
ATSC	**Choose activated image**
ATSG	**GPIO control**
WLAN	
ATPW	**Set WiFi mode**
ATPN	**Connect to AP (Station mode)**
ATWD	**Disconnect from AP**
ATWS	**Scan AP**
ATPA	**Set AP mode**
ATW?	**Show wlan currently status**
ATPH	**Set DHCP mode**
ATPE	**Set static IP for station mode**
ATPF	**Set static IP for AP, and DHCP rule**
TCPIP	
ATP0	**Get errno**
ATPS	**Open a TCP Server**
ATPC	Open a TCP Client
ATPD	Close TCP Server or Client
ATPT	Send Data to Server or Client in TCP connection
ATPR	Receive Data from TCP Server or Client of connection
ATPK	Enable auto receive data mode
ATPI	**Check network connectio status**
ATPP	Ping test
ATPU	Enter TCP Pass Through Mode
++++	Exit TCP Pass Through Mode

A.1 System

A.1.1 'AT ' Test AT command ready

Description: This command is used to test system boot successful.
Command Format:AT<CR>
Response: "[AT] OK" is success. Other response are fail.

A.1.2 'ATS?' Show all AT command

Description: This command will list all usable AT command.
Command Format:ATS?<CR>
Response: "[ATS?] <command list> [ATS?] OK" is success. Other response are fail.
Error number: [1: get command list fail]

A.1.3 'ATSR' Restart module

Description: This command is used to restart the module.
Command Format:ATSR<CR>
Response: "[ATSR] OK" is success. Other response are fail.

A.1.4 'ATSV' Show firmware version

Description: This command is used to show module firmware version.
Command Format:ATSV<CR>
Response: "Firmware version:XX" is success. Other response are fail.

A.1.5 'ATSE' Set UART echo mode and debug mode

Description: This command is used to enable / disable UART echo
Command Format:'ATSE=<echo>'
Parameter:
<echo>: 0 is disable , 1 is enable.
Response: "[ATSE] OK" is success. Other response are fail.
Error number: [1 , 2: parameter number error]
[3:echo should be '0' or '1' only]

A.1.6 'ATSY' Factory Reset

Description: This command is used to clean flash data, module will restore to factory setting.
Command Format:ATSY<CR>
Response: "[ATSY] OK" is success. Other response are fail.
Error number: [1: restore default data fail]
[2: restore default image fail]

A.1.7 'ATSU' UART configuration

Description: This command is used to set up UART parameter.

Command Format:ATSU=<baudrate>,<databits>,<stopbits>,<parity>
,<flowcontrol>,<configmode><CR>

Parameter: <baudrate> 2400,4800,9600,19200,38400,57600,
115200(default),921600

<databits> 5(5 bit data), 6(6 bit data), 7(7 bit data)
8(8 bit data default)

<stopbits> 1(1 bit stop default), 2(2 bit stop)

<parity> 0 (none parity default), 1(Odd parity), 2(Even parity)

<flowcontrol> 0: disable flowcontrol (default) , 1: enable flowcontrol

<configmodee> 0: set the current configuration and will not save to flash.

1: save configuration to flash and take effect immediately.

2: save configuration to flash and take effect after reboot.

Response: "[ATSU] OK" is success. Other response are fail.

Error number: [1: command format error]
[2: command parameter error]

A.1.8 'ATSU?' Show UART configuration

Description: This command is used to show UART currently setting.

Command Format:ATSU?<CR>

Response: "<UART setting> [ATSU?] OK" is success. Other response are fail.

A.1.9 'ATSO' OTA upgrade

Description: This command is used to upgrade firmware.

Command Format:ATSO= <Download server ip address> , <Download server port>
<CR>

Response: "[ATSO] OK" is success. Other response are fail.

Error number: [1: command format error]
[2: command parameter error]

(Download Sever shoud run first and module should connect to the same network as Download server)

A.1.10'ATSC' Choose Activated Image

Description: This command is used to choose the activated image

Command Format:ATSC=<image ID><CR>

Parameter: <image ID> 0: default image
1: OTA upgrade image

Response: "[ATSC] OK" is success. Other response are fail.

Error number: [1: command format error]
[2: command parameter error]

A.1.11'ATSG' GPIO control

Description: This command is used to control GPIO
Command
Format:ATSG=<R/W>,<PORT>,<DATA>,<DIR>,<PULL><CR>
Parameter: <R/W> R: read GPIO
W: write GPIO
<PORT> Px_x, ex:PA_0
<DATA> 0 or 1 when write gpio
<DIR> 0 : Input PIN
1: Output PIN
<PULL> 0: Pull None / Pull default
1: Pull Up
2: Pull Down
3: Open Drain
Response: "[ATSG] OK:<value>" is success. Other response are fail.
Error number: [1: command format error]
[2: command parameter error]
[3: invalid pin name]

A.2 WLAN

A.2.1 'ATPW' Set WiFi mode

Description: This command is used to set WiFi mode, when executing APTN and ATPA
command must check mode first.
Command Format:ATPW=<mode><CR>
Parameter: <mode> 1: Station mode (default)
2: AP mode
Response: "[ATPW] OK" is success. Other response are fail.
Error number: [1: command format error]
[2: command parameter error]

A.2.2 'ATPN' Connect to AP

Description: This command is used to connect to AP for station.
Command Format:ATPN=<ssid>,<pwd><CR>
Parameter: <ssid> This parameter can't be empty. Must add prefix '\' for special
character(',' , '\' , '"', '[', ']')
<pwd> 1. WPA/WP2: length is 8~64.
2. WEP: length is 5 or 13.
Response: "[ATPN] OK" is success. Other response are fail.
Error number: [1: command format error]
[2: command parameter error]
[3: WiFi initial error]
[4:connect to AP failed]
[5:WiFi mode error]
[6: get ap security type failed]
[7:dhcp timeout, use static ip 192.168.1.80]
(1.Execute ATPW first, must be station mode. 2.If no password, remain the parameter <pwd> NULL)

A.2.3 'ATWD' Disconnect from AP

Description: This command is used to disconnect with AP for station.
Command Format:ATPWD<CR>
Response: "[ATWD] OK" is success. Other response are fail.
Error number: [3: operation failed]
[4: disconnect timeout]

A.2.4 'ATWS' Scan AP

Description: This command is used to scan AP in the air.
Command Format:ATPWS<CR>
Response: "AP: <num>,<ssid>,<channel>,<security>,<rssi>,<bssid>
[ATWD] OK" is success. Other response are fail.

A.2.5 'ATPA' Set AP mode

Description: This command is used to configuration AP mode.
Command Format:ATPA=<ssid>,<pwd>,<channel>,<hidden><CR>
Parameter: <ssid> This parameter can't be empty. Must add prefix '\' for special
character(',' , '\' , '"', '[', ']')
<pwd> WPA/WP2: length is 8~64.
<channel> The channel range is 1~ 11.
<hidden> 0: Not hidden ssid.
1:hidden ssid.
Response: "[ATPA] OK" is success. Other response are fail.
Error number: [1: command format error]
[2: command parameter error]
[3: WiFi initial error]
[4:start AP failed]
[5:WiFi mode error]
(1.Execute ATPW first, must be station mode. 2.If no password, remain the parameter <pwd> NULL)

A.2.6 'ATW?' Show wlan currently status

Description: This command is used to list WiFi information.
Command Format:ATW?<CR>
Response: "<WLAN status >[ATW?] OK" is success. Other response are fail.

A.2.7 'ATPH' Set DHCP mode

Description: This command is used to set DHCP function for both mode.
Command Format:ATPH=<mode>,<enable><CR>
Parameter: <mode> 1: AP mode
2:Station mode
<enable> 1:DHCP
2:Static IP
Response: "[ATPA] OK" is success. Other response are fail.
Error number: [1: command format error]
[2: command parameter error]
(1.Default is DHCP for both mode.
2. Use ATPE to set static IP for station.
3.Use ATPF to set DHCP rule for AP.)

A.2.8 'ATPE' Set static IP for Station

Description: This command is used to set static IP for station.
Command Format:ATPE=<IP>,<gateway>,<mask><CR>
Parameter: <IP> Static station IP
<gateway>Set gateway IP(optional)
<mask> Set mask IP (optional)
Response: "[ATPE] OK" is success. Other response are fail.
Error number: [1: command format error]
[2: command parameter error]
(Default static IP of station is 192.168.1.80)

A.2.9 'ATPF' Set DHCP rule and gateway

Description: This command is used to set DHCP rule and gateway for AP.
Command Format:ATPF=<start_ip>,<end_ip>,<gateway><CR>
Parameter: <start_ip> Set the start IP for client.
<end_ip> Set the end IP for client.
<gateway>Set gateway IP.
Response: "[ATPF] OK" is success. Other response are fail.
Error number: [1: command format error]
[2: command parameter error]
(Default gateway IP is 192.168.43.1)

A.3 TCPIP

A.3.1 'ATP0' Get LWIP errno

Description: This command is used to get errno in LWIP.
Command Format:ATP0<CR>
Response: "[ATP0] OK:<errno>" is success. Other response are fail.

A.3.2 'ATPS' Open a TCP Server

Description: This command is used to open a TCP server.

Command Format:ATPS=0,<Local Port><CR>

Response: "[ATPS] OK [ATPS] con_id= X" is success. Other response are fail.

Error number: [1: parameter number error]
[2: local port should be 1~65535]
[3:create con_id error]
[4: create server task error]
[5:create socket error]
[6: set socket option error]
[7: bind error]
[8: listen error]
[9: tcp server already exists error]
[10:accept error]
[11:create con_id for seed error]
[13:server can't start under TCP Pass Through mode]

(Auto receive mode will enable when TCP client connected with TCP server)

A.3.3 'ATPC' Open a TCP Client

Description: This command is used to open a TCP client.

Command Format:ATPC=0,<Remote IP ><Remote Port><CR>

Response: "[ATPC] OK [ATPC] con_id= X" is success. Other response are fail.

Error number: [1: parameter number error]
[2: remote IP format or host unfound error]
[3: remote port should be 1~65535 error]
[4: create con_id error(none available)]
[5: create client task error]
[6:inet_ntoa_r remote address error]
[7: create socket error]
[8: hang node error for tcp client]
[9: connect error for tcp client]
[11:local port should be 1~65535]
[12:bind local port error]
[13:connection already exists for TCP Pass Through mode]
[14: set broadcast on socket failed]
[15:set muticast add membership on socket failed]
[16:set muticast interface failed]

(Auto receive and Data pass through mode will enable when TCP client connected with TCP server)

A.3.4 'ATPD' Close a TCP Server or Client

Description: This command is used to close TCP Server or Client.
Command Format:ATPD<CR>
Response: "[ATPD] OK" is success. Other response are fail.
Error number: [1: command format error]
[2: command parameter error]
[3: no con_id is found]

A.3.5 'ATPT' Send data

Description: This command is used to send data to a specific connection.
Command Format: ATPT= <data_size>, <con_id>:<data> <CR>
Parameter: <data_size> Data length
<con_id> 1~9
<data> Payload data
Response: "[ATPT] OK, <con_id>" is success. Other response are fail.
Error number: [1: parameter number error]
[2: <Buffer Size> exceeds ATPT send buffer size]
[3: con_id is not found]
[7: TCP server should send data to the seed]
[8: write error for tcp client/server]

A.3.6 'ATPR' Receive data

Description: This command is used to receive data from a specific connection id
Command Format: ATPR= <Buffer Size>, <con_id>, <data> <CR>
Parameter: <Buffer Size > Buffer Size
<con_id> 1~9
<data> Payload data
Response: "[ATPR] OK, <con_id>" is success. Other response are fail.
Error number: [1: command format error]
[2: <Buffer Size> error(should be 1~ MAX_BUFFER(default))]
[3: con_id is not found]
[7: TCP server should send data to the seed]
[8: write error for tcp client/server]

A.3.7 'ATPK' Set auto receive data mode

Description: This command is used to set auto receive data mode.
Command Format: ATPK= <enable> <CR>
Parameter: <enable > 0: disable auto receive data mode (default)
1: enable auto receive data mode

Response: "[ATPK] OK" is success. Other response are fail.
Error number: [1: command parameter error]
[2:start auto receive task fail]

A.3.8 'ATPI' Check network connection status

Description: This command is used to print network connection status.
Command Format: ATPI<CR>
Response: "[ATPI] OK" is success. Other response are fail.

A.3.9 'ATPP' Ping test

Description: This command is used to ping a specific connection id, or ping a specific
network address
Command Format: ATPP = <Remote IP>, <Count/loop> <CR>
Parameter: <Remote IP >
<Count/loop > Loop: loop, no count
Count: loop with count

Response: "[ATPP] OK" is success. Other response are fail.
Error number: [1: command format error]
[2: con_id is not found]

A.3.10 'ATPU' Set TCP Pass Through mode

Description: This command is used to set TCP Pass Through mode in TCP client
Command Format: ATPU = 1<CR>
Response: "[ATPU] OK" is success. Other response are fail.
Error number: [1: command parameter error]
[2: no connection found when try to enter TCP Pass Through mode]
[3:can not enter TCP Pass Through mode if it's server connection]
[4:more than one connection when try to enter TCP Pass Through mode]
[5:start TCP Pass Through mode task failed]
(TCP Pass Through mode only support TCP client)

A.3.11 '++++' Exit TCP Pass Through mode

Description: This command is used to exit TCP Pass Through mode when TCP client enter TCP Pass Through mode.
Command Format: ++++
Response: None

B. **RB8762CMF-a AT command list**

RW8762CMF-a provides to change serial parameters, BLE setting and BLE function by AT command. The AT command list is following Table 13, Table 14.

Table 13 RW8762CMF-a AT command list-1

AT Commands	Slave Side	Master Side	Description
at	√	√	Test UART channel
at+laddr	√	√	Read BD address
at+name	√	√	Read device name
at+name=<Parameter>	√	√	Set device name (Max. length=15 byte) Default name is “Radicom”.
at+baud	√	√	Check current baud rate. Default value: 115200bit/s
at+baud=< Parameter>	√	√	Set Baud rate. 2400bit/s 4800bit/s 9600bit/s 19200bit/s 38400bit/s 57600bit/s 115200bit/s 921600bit/s (USB only)
at+role	√	√	Check the role of device.
at+role=< Parameter>	√	√	Role setting. Parameter range(0,1): 0: slave device 1: master device
at+inq	Invalid	√	Get slave device’s BD address (Max.=10 slave, sequence number 0-9) After this command “at+sinq” must be the next command
at+sinq	Invalid	√	Stop searching for BT device. *Must-do step, or master device will keep searching.
at+conn=< Parameter>	Invalid	√	Connect a slave device. *Input parameter from “AT+INQ”.
at+reset	√	√	Reboot the device
at+version	√	√	Read version number: patch version (8 characters) + app version (8 characters) Patch and app versions are both 32 bits, in the form of a string

Table 14 RW8762CMF-a AT command list-2

AT Commands	Slave Side	Master Side	Description
at+advmod=< Parameter>	√	Invalid	Parameter (0,1): 0: Manual broadcast 1: Automatic broadcast Default value: 1
at+advmod	√	Invalid	Check advmod< Parameter> settings
at+adven=< Parameter>	√	Invalid	Parameter (0,1): 0: turn off the broadcast 1: Turn on broadcasting Default value: 0
at+advint=< Parameter>	√	Invalid	Default value: 0x0140(200ms) Parameter : XXXX (HEX mode) Range: 0020 <= XXXX <= 4000
at+advint	√	Invalid	Check advint< Parameter> settings
at+power=< Parameter>	√	√	Parameter（-20,0,3,4,8） -20：-20dBm 0： 0dBm (default) 3： 3dBm 4： 4dBm 8： 8dbm
at+power	√	√	Check power< Parameter> settings
at+default	√	√	Restored default settings (1) Module name: "Radicom" (2) Broadcast mode: automatic (Slave only) (3) Serial port baud rate: 9600

RB8762CMF-a supports BLE master and slave mode, user change mode by AT command. Default BLE function is data pass through mode. If user want to use other BLE function, please contact to vender. Below steps will be shown how to BLE by AT command.

B.1 BLE Master

Step 1: Type "**at+role=1<CR><LF>**" to set as master.

Step 2: Type "**at+inq<CR><LF>**". Get slave's BD address.
OK. At+inq<CR><LF>
+INQ<CR><LF>
0:0X0012A1123456

Step 3: Type "**at+sinq<CR><LF>**". Stop the master from searching BT device.
+INQE<CR><LF>

Step 4: Type "**at+conn0<CR><LF>**". Connect to slave device.
OK<CR><LF>
+CONNS<CR><LF>
+CONNECTED>>0X12A1123456<CR><LF>

B.2 BLE Slave

Step 1: Enter "**at+role=0<CR><LF>**" to set as slave.

Step 2: Enter "**at+name<CR><LF>**" to read device name.
+Name=Radicom

Step 3: Enter "**at+laddr<CR><LF>**" to read BD address.
+LADDR=0012A1123456

※Device will enter data mode as long as connect. If you want access command mode again, please send "+++ath" to the connecting device. The device will leave data mode and enter command mode.

How to test each module

RA6000 series EVK provides Bluetooth and WiFi module, and Arduino socket for Arduino I/O card. The chapter will be shown you how to test these module and interface.

A. Testing with Arduino module

A.1 Lab2_SPI_Arduino_SC16IS752_GPIO

This section will be shown how to use 752 GPIO. The test environment of sample is following Figure 29.

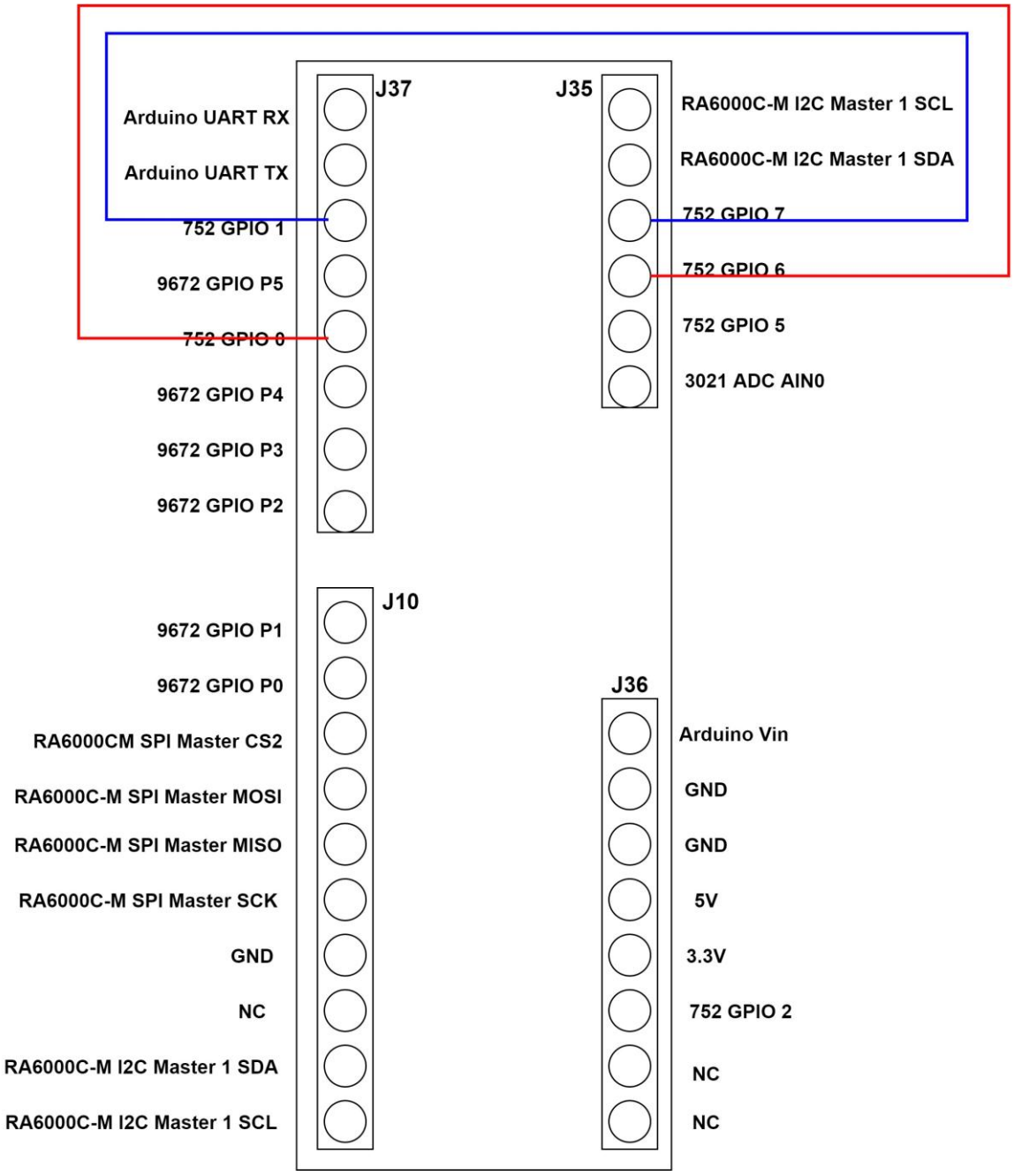


Figure 29 Arduino GPIO test wiring diagram

752 GPIO 0 and 752 GPIO 1 are output pin, 752 GPIO 6 and 752 GPIO 7 are input pin. 752 GPIO 6 set to interrupt.

Follow as below steps to test Arduino752 GPIO.

Step 1: Refer Lab1_ArduinoGPIO752 sample from SDK.

Step 2: Type "**make**" and "**make flash**" to build firmware.

Step 3: Upgrade firmware to RA6000 series EVK.

Step 4: Open debug message, 752 GPIO state will be shown

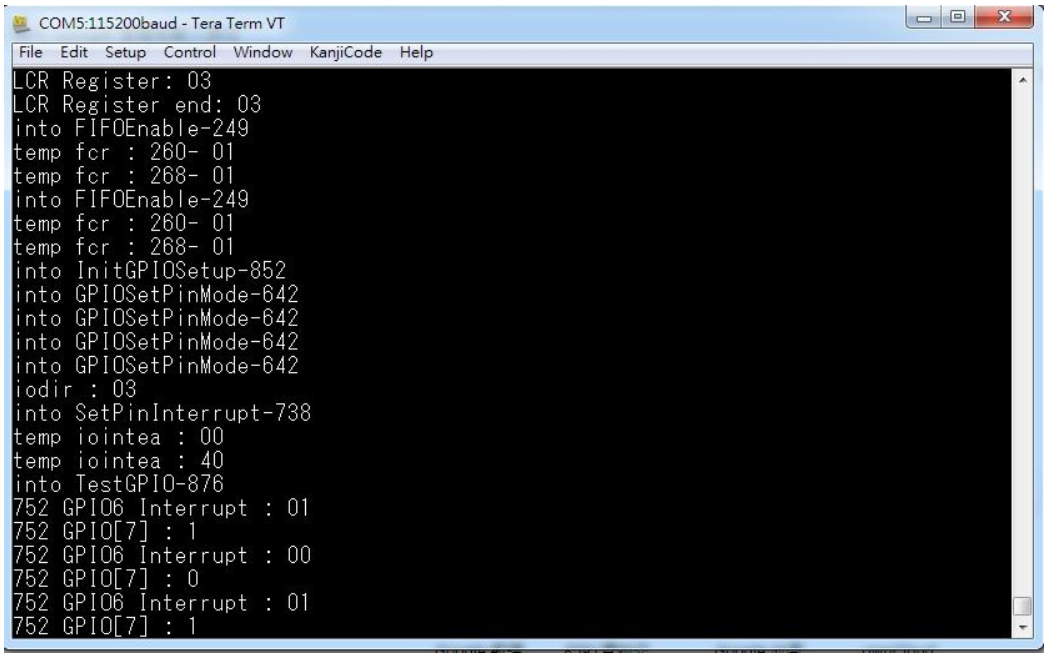


Figure 30 GPIO State

A.2 Lab1_ArduinoGPIO752_LEDBtn

This section will be shown how to control LED by button. The test environment of sample is following Figure 31.

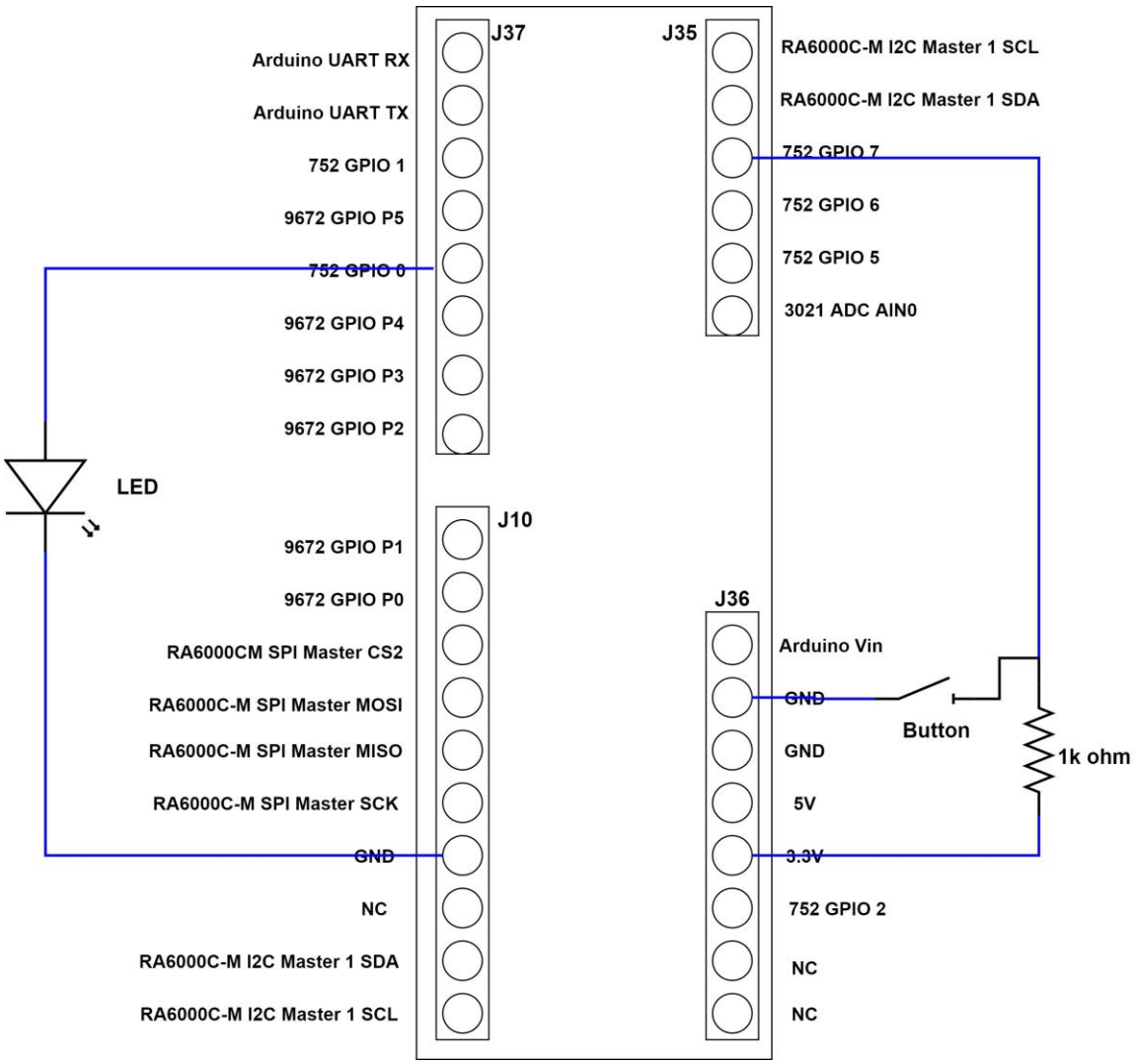


Figure 31 Arduino LED and Button wiring diagram

752 GPIO 0 is output pin, 752 GPIO 7 is input pin. The LED will show 752 GPIO 7 state.

Follow as below steps to test LED and button sample.

Step 1: Refer Lab1_ArduinoGPIO752_LEDBtn sample from SDK.

Step 2: Type "**make**" and "**make flash**" to build firmware.

Step 3: Upgrade firmware to RA6000 series EVK, then power on RA6000 series EVK.

Step 4: When user pushes button, LED will be dark. And user don't push button, LED will be glowing.

A.3 Lab1_ArduinoADC

RA6000 series EVK provides a 12 bits ADC port under Arduino socket. The ADC AIN0 voltage range is **0V to 3.3V**. This section will be shown how to get ADC value by I2C. The test environment of sample is following Figure 32.

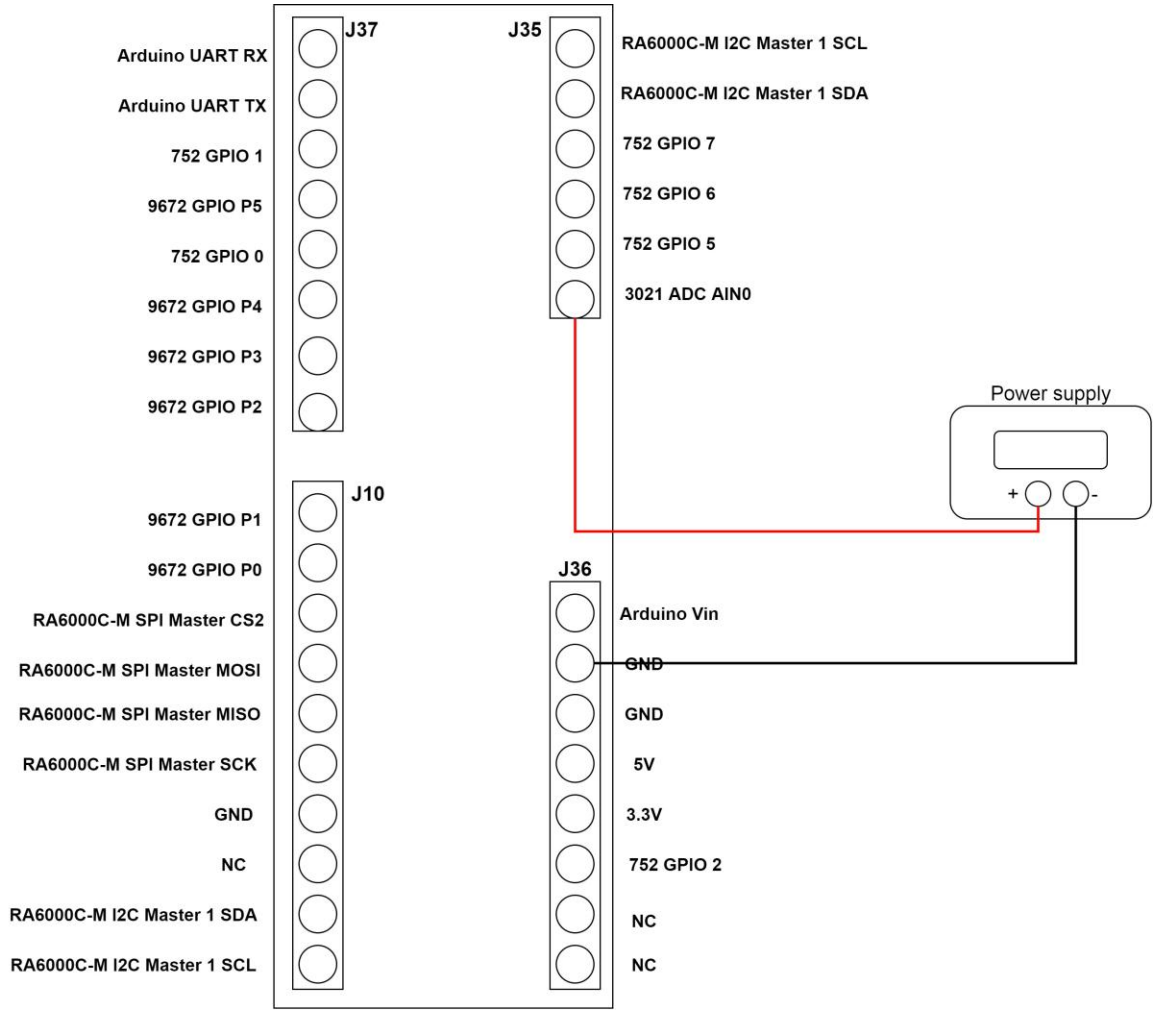


Figure 32 Arduino ADC test environment

Follow as below steps to test Arduino ADC.

Step 1: Refer Lab1_ArduinoADC sample from SDK.

Step 2: Type "**make**" and "**make flash**" to build firmware.

Step 3: Upgrade firmware to RA6000 series EVK.

Step 4: Open debug message, ADC value will be shown.

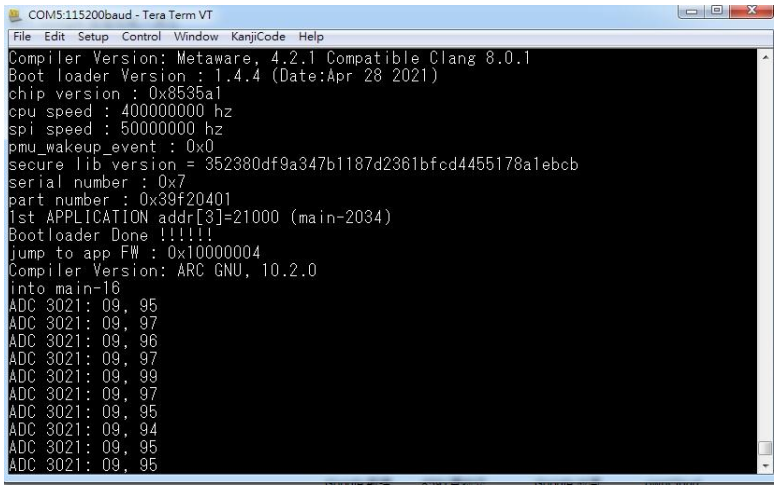


Figure 33 ADC test result

The equation of ADC value converting to voltage is following.

$$ADC\ AIN0 = \frac{ADC\ Value(0x0000 \sim 0x0fff)}{1241} \pm 0.01\ (V)$$

A.4 Lab1_ArduinouUART

RA6000 series EVK provides a UART under Arduino socket. This section will be shown how to test Arduino UART. The test environment of sample is following Figure 34.

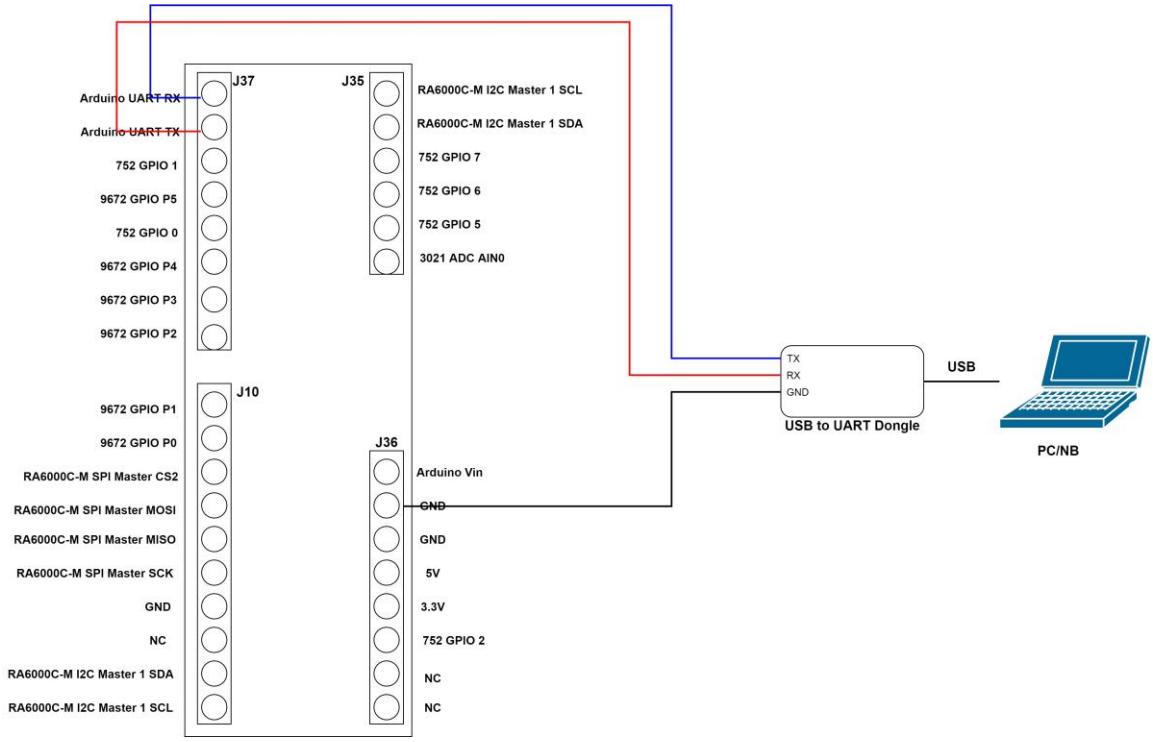


Figure 34 Arduino UART test environment

Follow as below steps to test Arduino UART.

Step 1: Refer Lab1_ArduinoUART sample from SDK.

Step 2: Type "**make**" and "**make flash**" to build firmware.

Step 3: Upgrade firmware to RA6000 series EVK.

Step 4: Open a terminal for USB to UART dongle.

Step 5: key in character under terminal, the character will be shown under terminal.

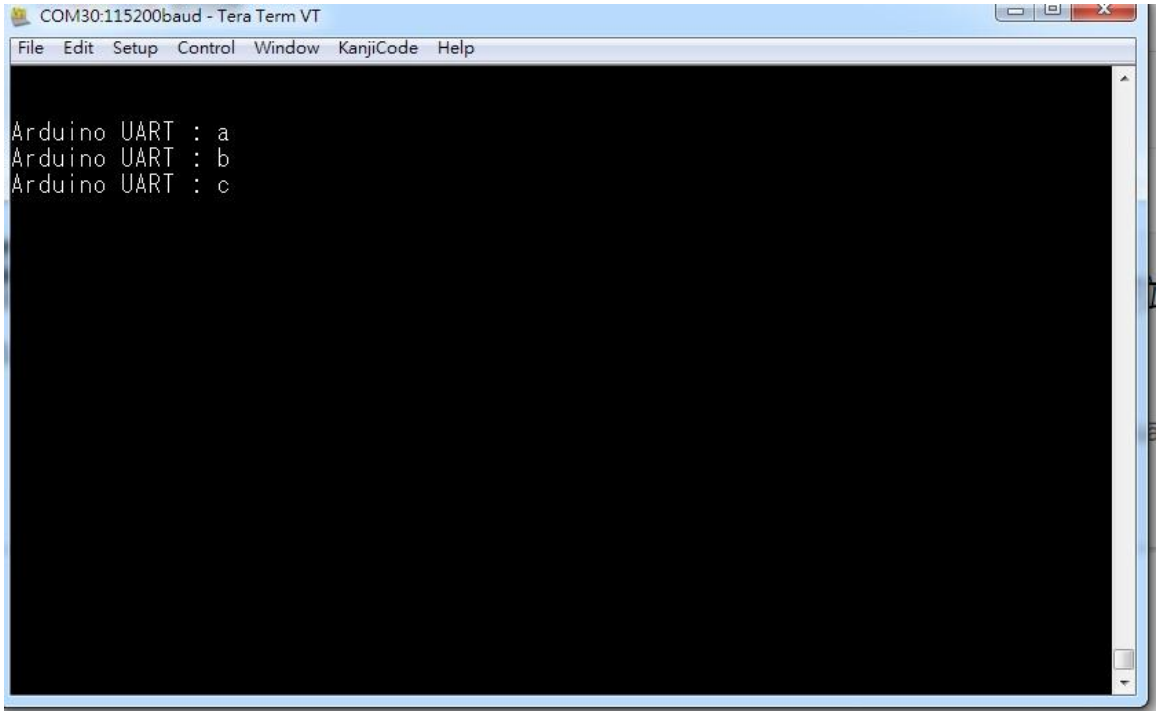


Figure 35 UART test result

B. Testing WiFi module with RBW6100-RA
B.1 Lab1_GetWiFiInfo

Follow as below steps to get WiFi module information.

Step 1: Refer Lab1_GetWiFiInfo sample from SDK.

Step 2: Type "**make**" and "**make flash**" to build firmware.

Step 3: Upgrade firmware to RA6000 series EVK.

Step 4: Open debug message, WiFi information will be shown.

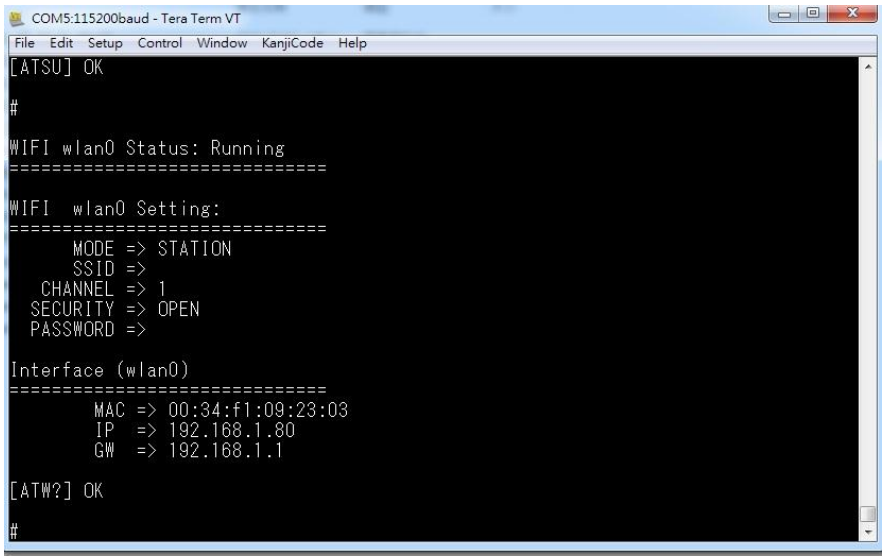


Figure 36 WiFi information

B.2 Lab2_WiFi_TCP_S_Passthrough

RBW6100-RA provides “open AP router” and “open socket” function by AT command. The user opens TCP server or TCP Client socket by AT command. And transfer data by data pass through mode. The test environment is following Figure 37.

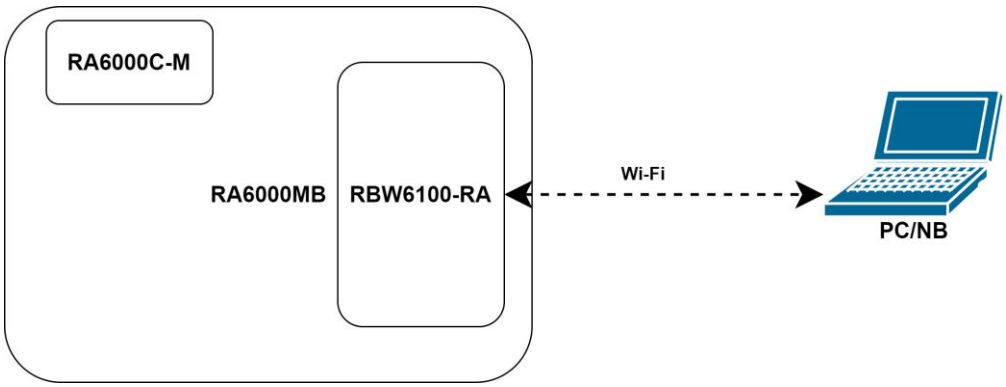


Figure 37 WiFi TCP server test environment

Note: this sample is worked under Ubuntu 20.04.

Follow as below steps to test WiFi data passthrough.

Step 1: Refer Lab2_WiFi_TCP_S_Passthroguh sample from SDK.

Step 2: Type “**make**” and “**make flash**” to build firmware.

Step 3: Upgrade firmware to RA6000 series EVK.

Step 4: Open debug UART (UART0), WiFi information will be shown.

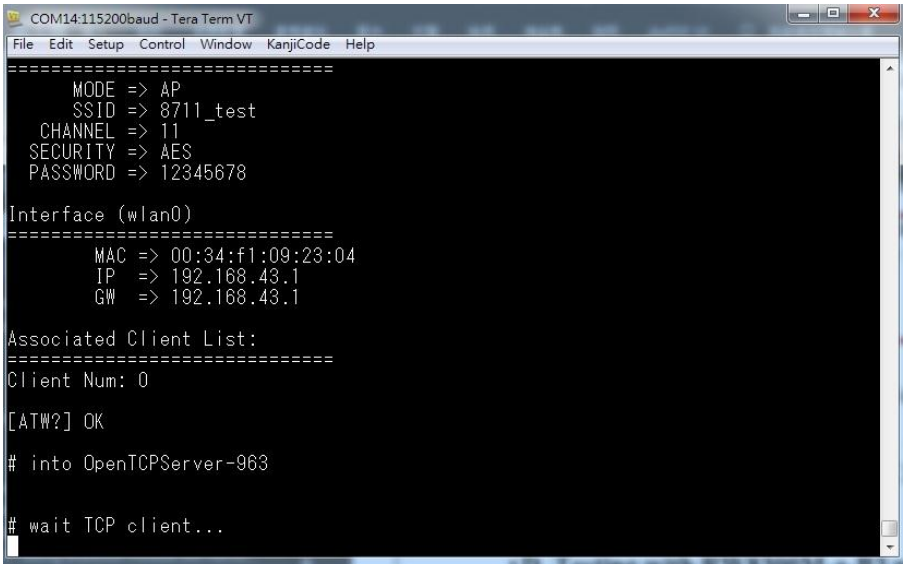


Figure 38 WiFi information

Step 5: Connect to RBW6100-RA WiFi under PC/NB.
(SSID: "**8711_test**", Password: "**12345678**")

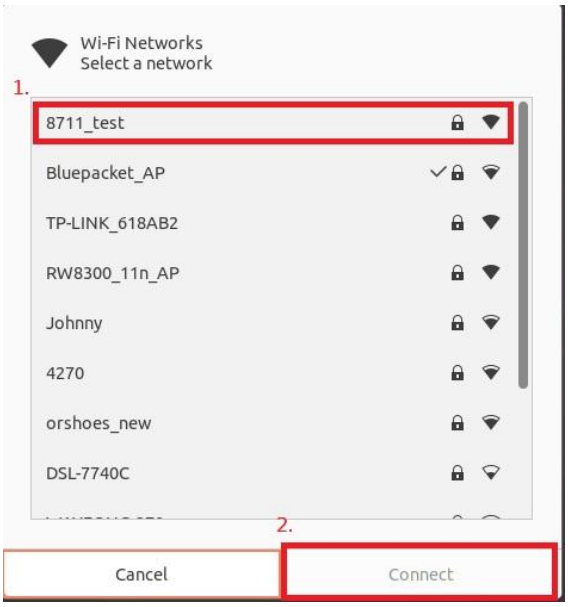


Figure 39 Connect to RBW6100-RA AP

Step 6: Open a terminal, and type "**nc 192.168.43.1 8000**" to connect RBW6100-RA server.
(command format: "**nc <server ip> <server port>**")

Step 7: Type some characters and push "Enter" key under terminal. Then the data will be sent to RA6000 series EVK by TCP. The data will be shown under debug UART (UART0).

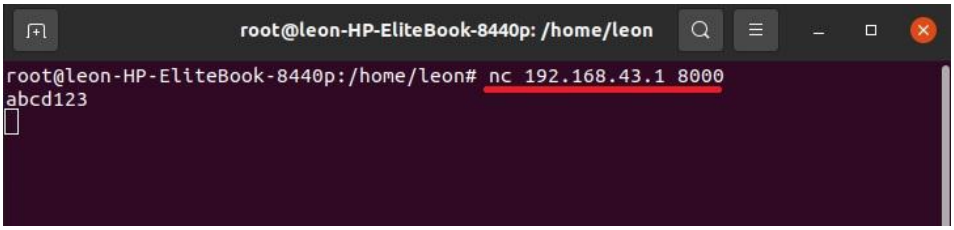


Figure 40 Connect to TCP server and send data.

Figure 41 RA6000 debug UART (UART0) message.

Step 8: Type some characters under debug UART (UART0) terminal. Then the data will be sent to PC/NB by TCP. The data will be shown under Ubuntu terminal.

Figure 42 type data under RA6000 debug UART (UART0)

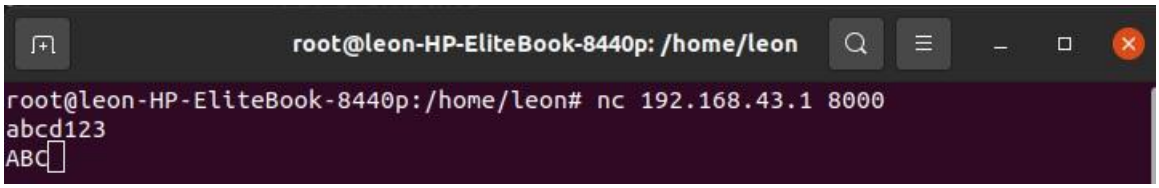


Figure 43 Ubuntu terminal show data.

C. Testing RBW6100-RA or RB8762C-RA Bluetooth module
C.1 Lab1_GetBLEInfo

Follow as below steps to get Bluetooth module information.

Step 1: Refer Lab1_GetBLEInfo sample from SDK.

Step 2: Type "**make**" and "**make flash**" to build firmware.

Step 3: Upgrade firmware to RA6000 series EVK.

Step 4: Open debug message, Bluetooth information will be shown.

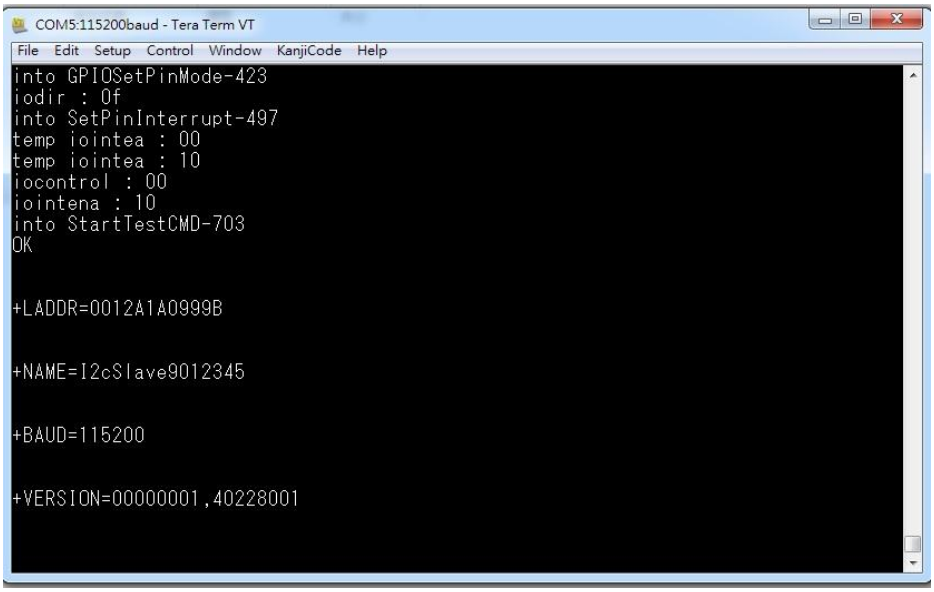


Figure 44 BLE information

C.2 Lab2_BLE_Passthrough

The default mode of RBW6100-RA and RB8762C-RA are data pass through mode, BLE role is slave. The Lab2_BLE_Passthrough sample is through UART0 and external BLE master communication. The test environment is following Figure 45.

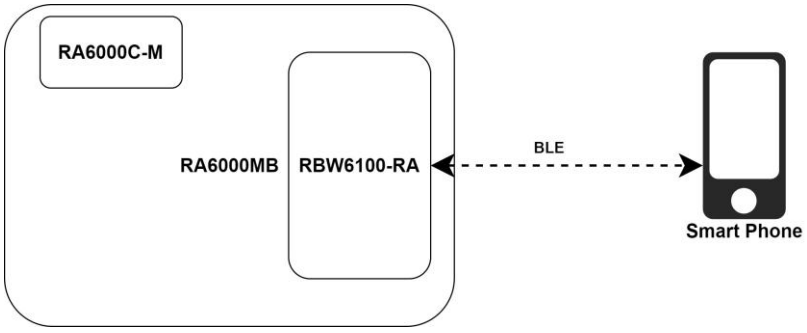


Figure 45 BLE data passthrough environment

Follow as below steps to get Bluetooth module information.

Step 1: Refer Lab1_GetBLEInfo sample from SDK.

Step 2: Type "**make**" and "**make flash**" to build firmware.

Step 3: Upgrade firmware to RA6000 series EVK.

Step 4: Power on RA6000 series EVK, the debug UART will show BLE information.

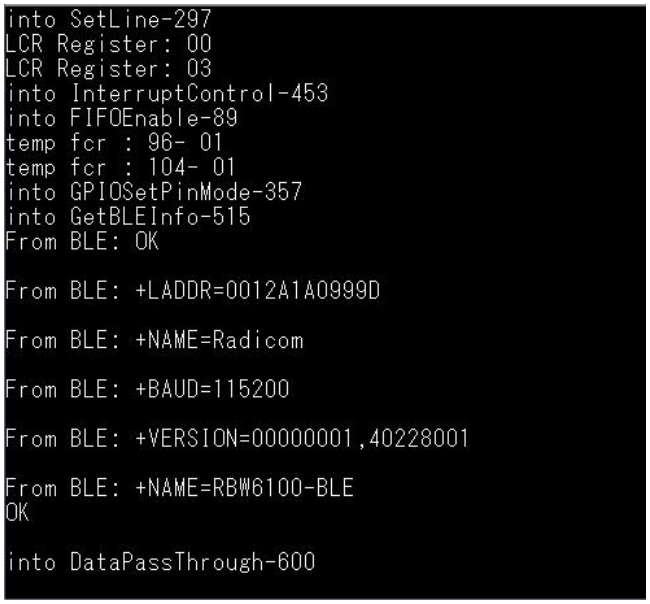


Figure 46 BT information

Step 5: Install "BLE Scanner" application under smart phone.

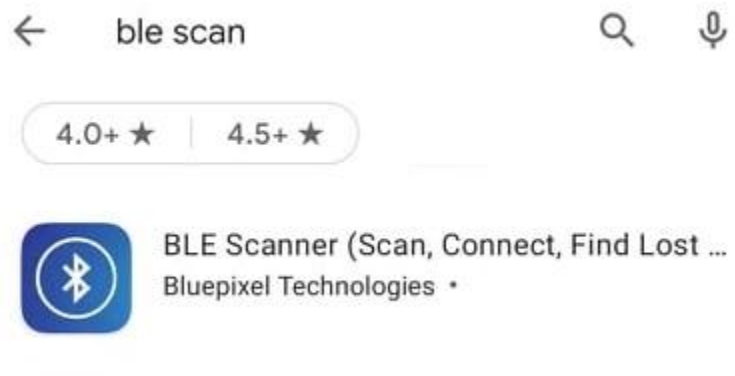


Figure 47 Install BLE Scanner

Step 6: Click “CONNECT” of button to connect to RBW6100-RA-BLE.

Figure 48 Connect to BLE

Step 6: Click “CUSTOM SERVICE” and Click “N” of UUID (000FFE2). Then click “W” of UUID (000FFE1) to send data.

Figure 49 CUSTOM SERVICE

Step 7: Type “abc” under textbox, and click “OK” button.

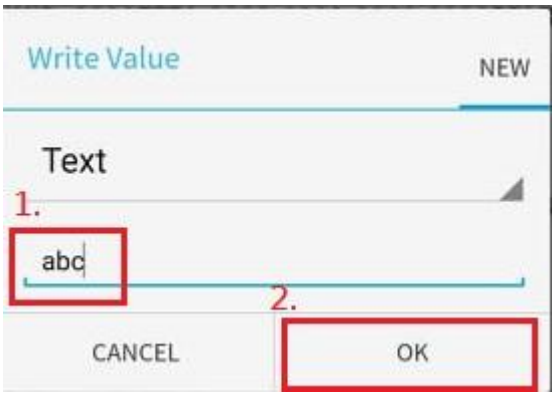


Figure 50 Send data

Step 7: Now, debug UART will show BLE data. Then type “a” under debug UART.

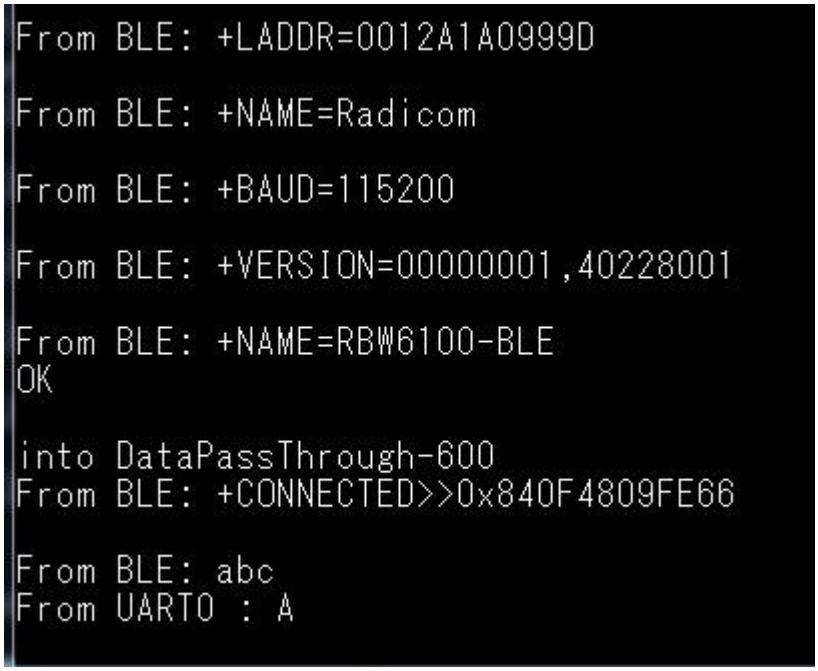


Figure 51 Debug UART(UART0) receive data

Step 8: Now, BLE Scanner will show debug UART data.

Figure 52 BLE receive data

D. Testing with RW8300M-a module

(Reserved)

E. Testing Ethernet

(Reserved)

F. Testing FT4222H SPI Master to RA6000 series EVK SPI Slave to open camera

Follow as below steps to test "**aito_bodydetect**" sample.

Step 1: Follow as below figure to setup jumper.

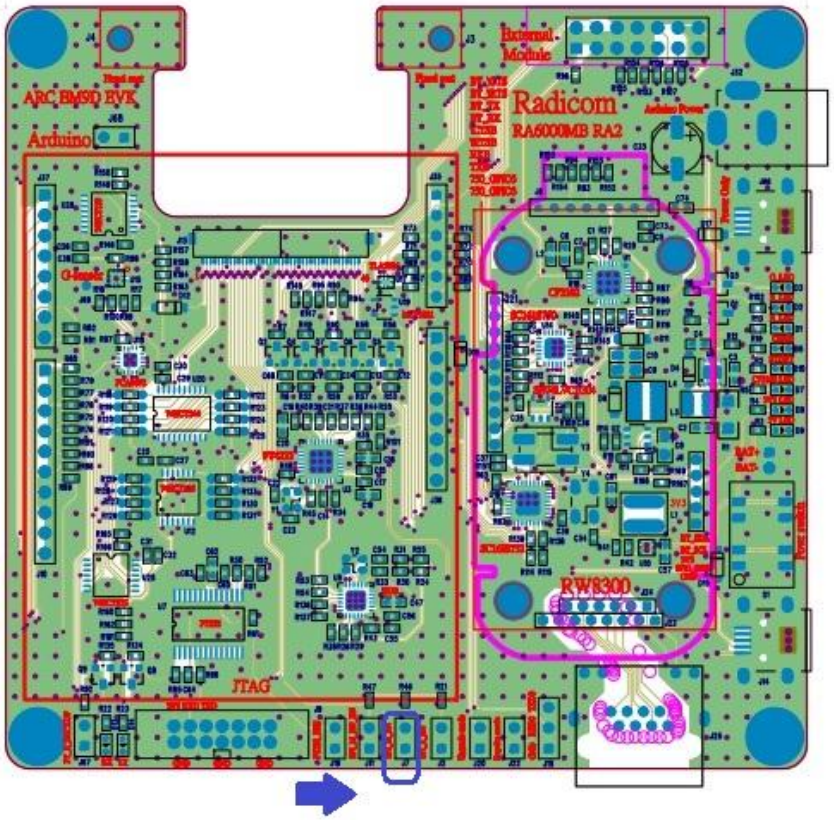


Figure 53 J2 location

Figure 54 J2 is shorted

Figure 55 Camera for FT4222 SPI slave physical diagram

Step 2: Refer Lab5_tflm_original sample from SDK.

Step 3: Type "**make**" and "**make flash**" to build firmware.

Step 4: Upgrade firmware to RA6000 series EVK.

Step5: Open GUI Tool, and open camera.
Note: In this version, SPI clock limit on 10 MHz.

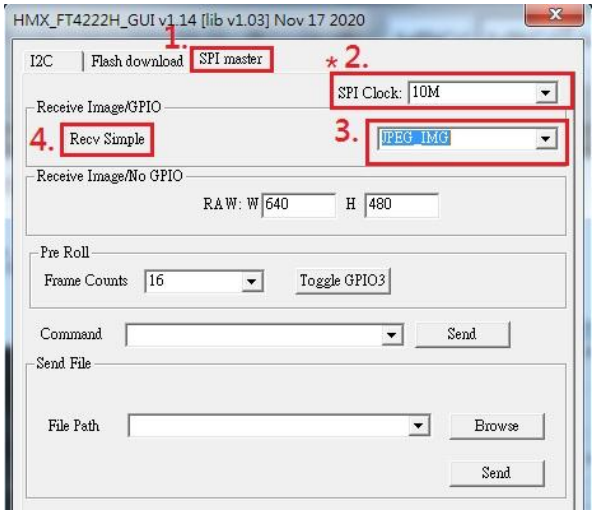


Figure 56 GUI Tool open camera -1

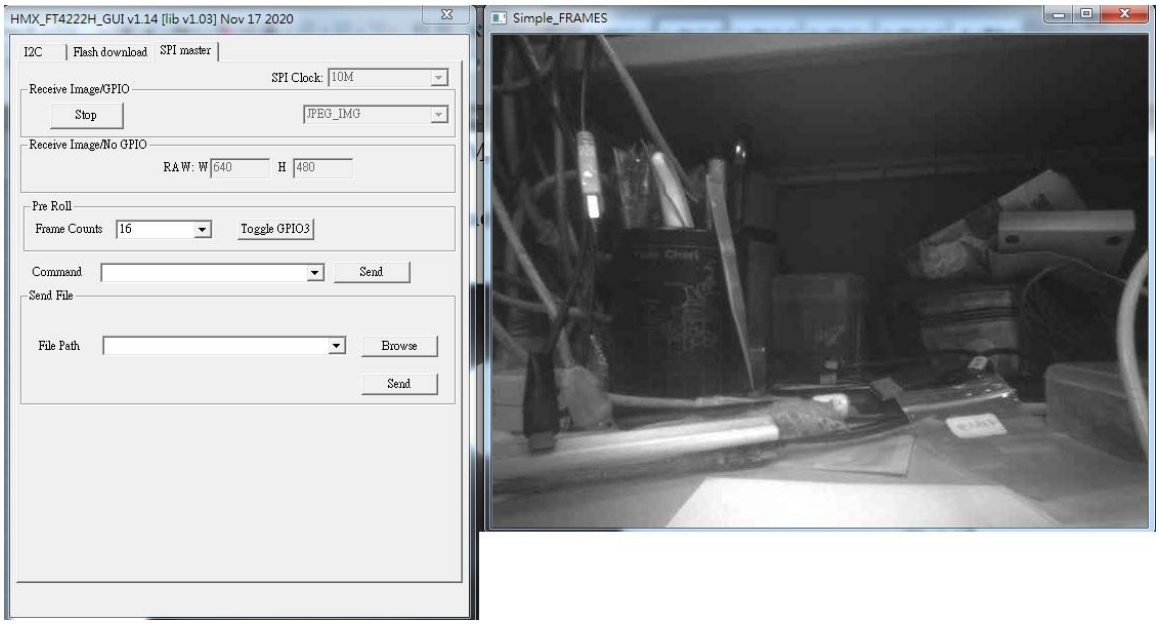


Figure 57 GUI Tool open camera -2

G. Testing RA6000 series EVK SPI Master to FT4222H SPI Slave to open camera

Follow as below steps to test “aito_bodydetect” sample.

Step 1: Follow as below figure to setup jumper.

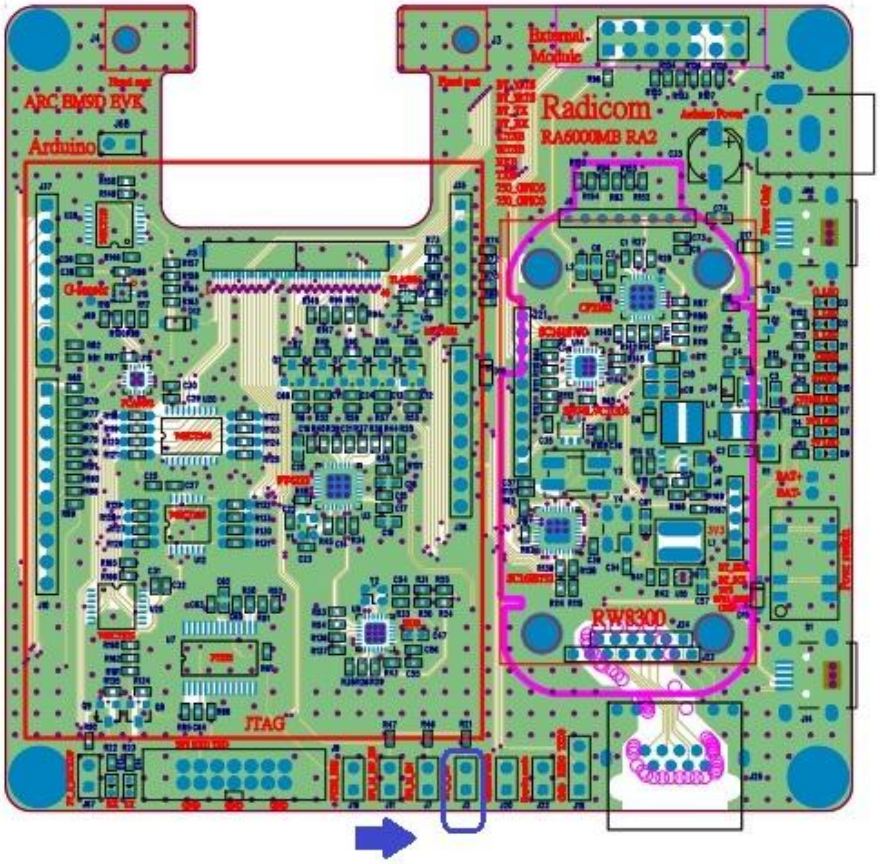


Figure 58 J2 location

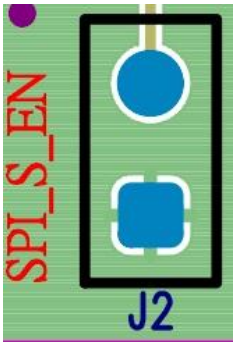


Figure 59 J2 is shorted

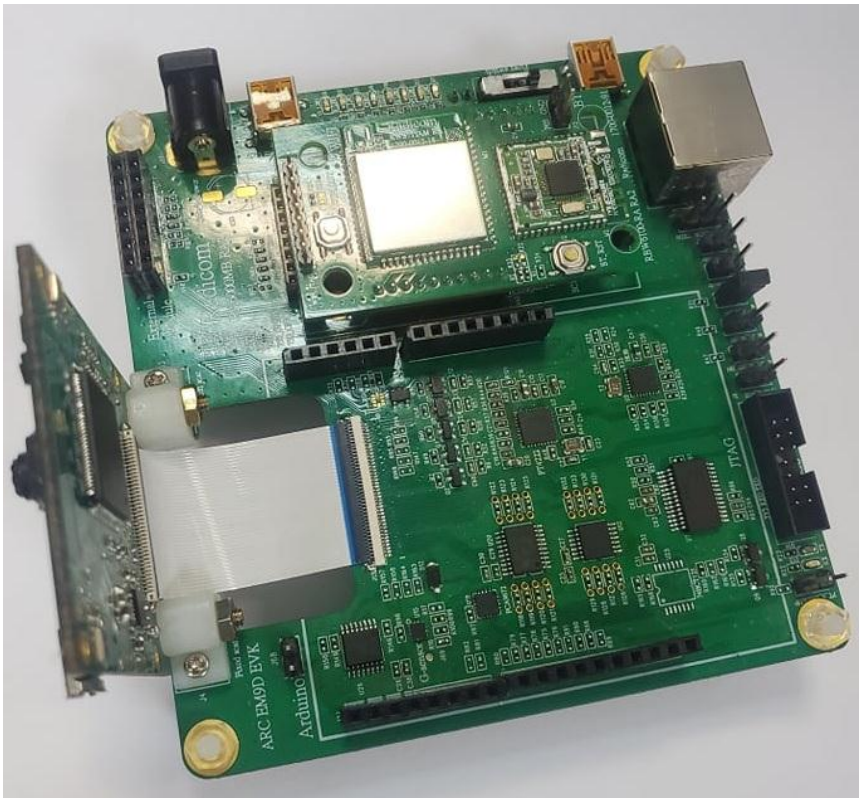


Figure 60 Camera for FT4222 SPI slave physical diagram

Step 2: Refer Lab5_tflm_original sample from SDK, and follow as below figure to modify app.mk. (Default is SPI slave)

```
EVENTHANDLER_SUPPORT = event_handler
EVENTHANDLER_SUPPORT_LIST += evt_i2ccomm evt_datapath evt_peripheral_cmd
APPL_DEFINES += -DI2C_COMM
APPL_DEFINES += -DEVT_I2CCOMM
APPL_DEFINES += -DEVT_DATAPATH
APPL_DEFINES += -DEVENT_HANDLER_LIB
APPL_DEFINES += -DSPI_MASTER_SEND
#for tflm end

include $(EMBARC_ROOT)/app/scenario_app/scenario_app.mk
~
~
```

Figure 61 Change to SPI master

Step 3: Type "**make clean**", "**make**" and "**make flash**" to rebuild firmware.

Step 4: Upgrade firmware to RA6000 series EVK.

Step 5: Open GUI Tool, and open camera.

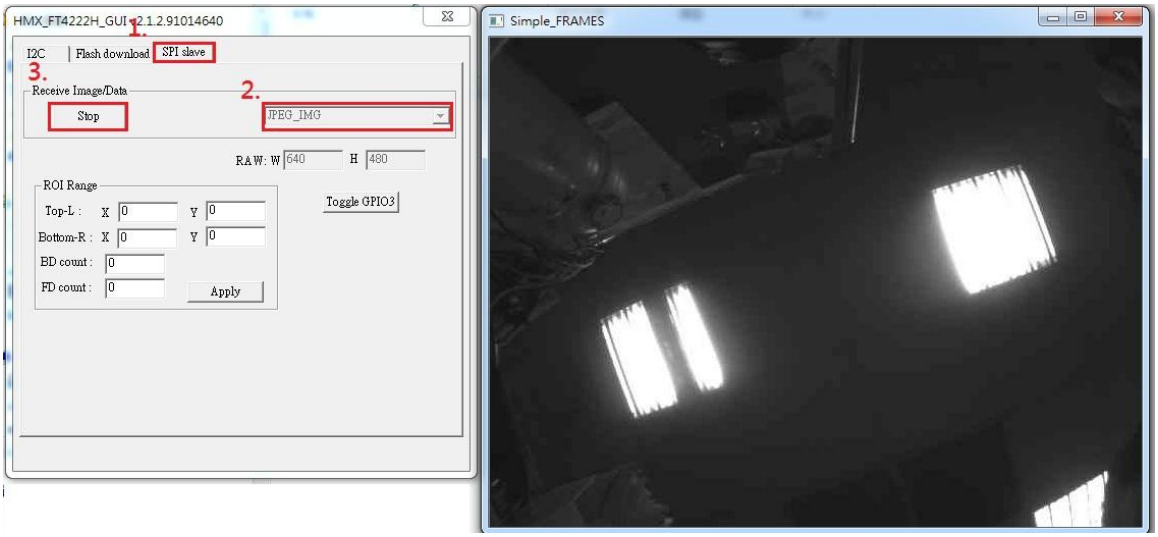


Figure 62 GUI Tool open camera

Software Development – GNU SDK

RA6000 series EVK SDK supports GNU and Mateware toolchain to build project, this section will show you how to build project by GNU.

Note: this sample is worked under Ubuntu 20.04.

GNU Development Toolkit

Download GNU tool chain, and GNU version is "**GNU tool chain for ARC Processors, 2021.03**".
(url: https://github.com/foss-for-synopsys-dwc-arc-processors/toolchain/releases/)
Note: The official package loss "tcftool". Please copy "tcftool" to 2021.03 package bin path from 2020.09 version.

Figure 63 Git Hub Download GNU Development Toolkit

After download and extract toolkit to local space, please remember to add it to environment PATH.
Command：
Type "***export PATH=[location of your ARC_GNU_ROOT]/bin:$PATH***" to set environment PATH under terminal.
Type "***echo $PATH***" to check environment PATH under terminal.

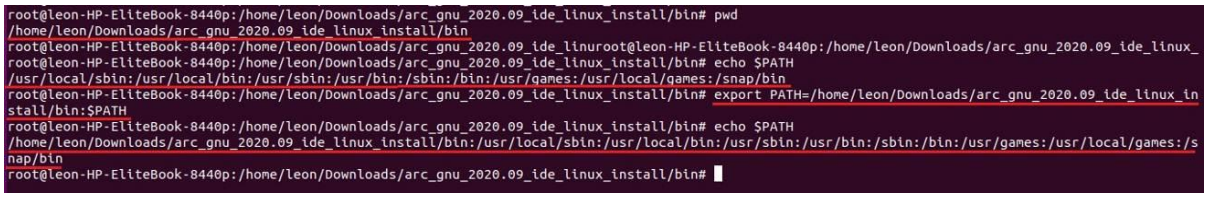


Figure 64 set and check environment PATH

Type "vim /etc/bash.bashrc" to edit script to auto set PATH.

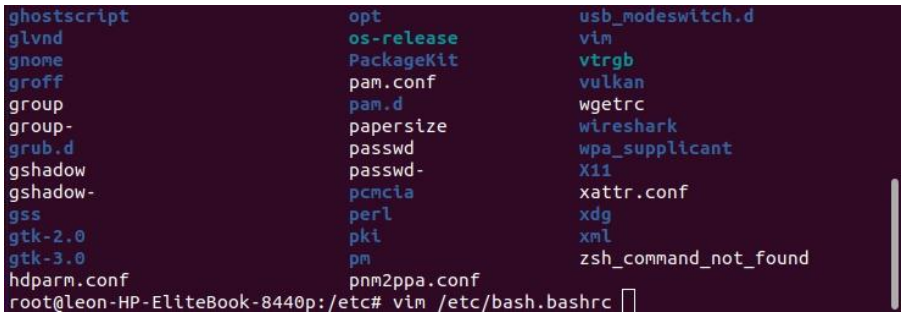


Figure 65 vim bash.bashrc

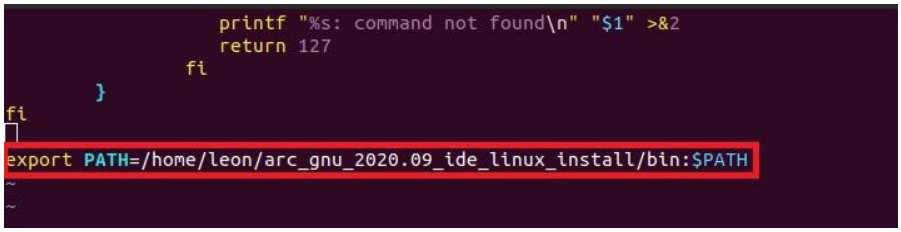


Figure 66 Add export command

SDK download and build project

Radicom will provide GNU SDK for RA6000 series EVK, please copy SDK file to Linux system, then decompression. Following as below steps to build project.

Step 1: open terminal and open SDK directory.

Step 2: Go to "***[location of your ARC_GNU_SDK_ROOT]/Example_Project***", to select sample project or create a new project in this directory.

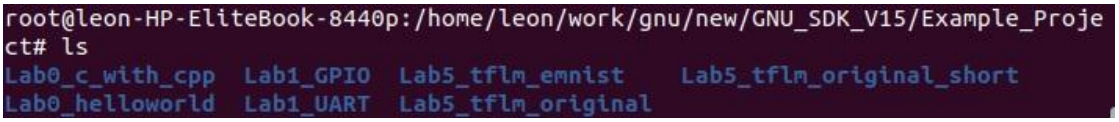


Figure 67 Open SDK directory

Step 3: type "***make clean***" to clean project.

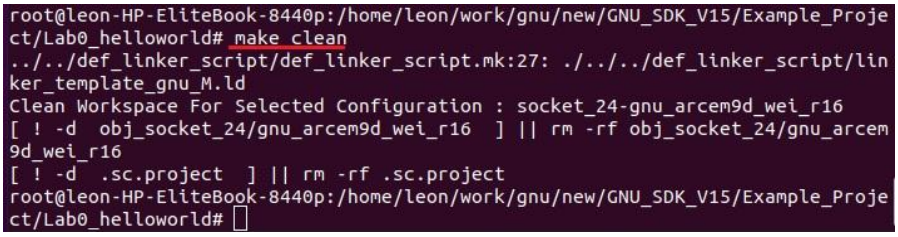


Figure 68 make clean

Step 4: type "***make***" to build project.

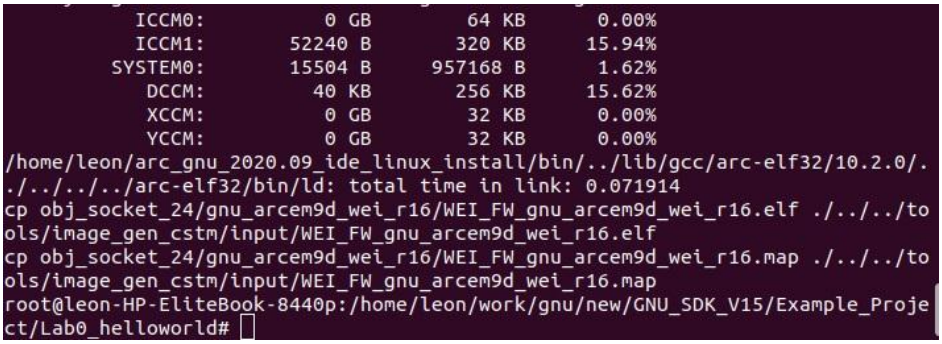


Figure 69 make

Step 5: type "***make flash***" to generate "output.img".

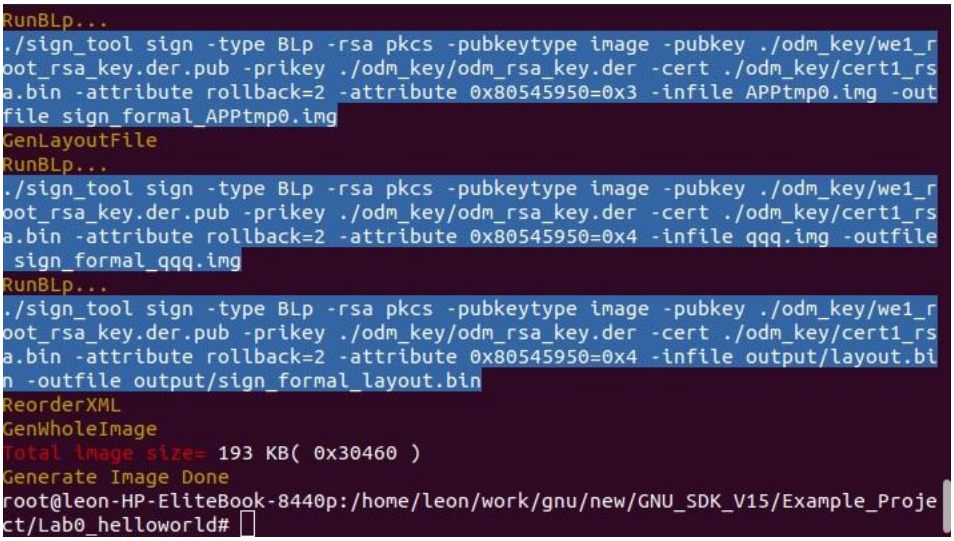


Figure 70 make flash

Step 6: Open directory "***$(SDK_ROOT_DIR)/tool/image_gen_cstm/output***" to copy **output.img** file.

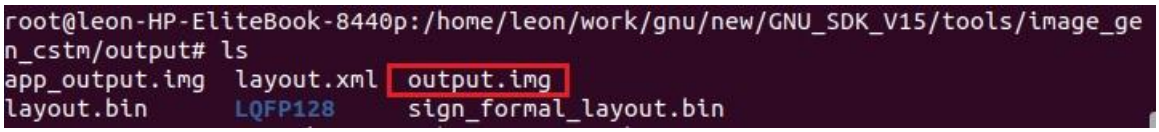


Figure 71 output.img.

Main CPU Programming

This chapter is shown you how to program, debug and repair. For program firmware, there are two ways to program main CPU module, one is using OTA tool and the other is using GUI tool of FT4222. For repair mode, user will can repair bootloader by OTA tool. For debugging, user will can debug by JTAG.

A. OTA Programming

The chapter will be shown how to OTA firmware by OTA tool. Please check EM9D firmware support OTA function.

Please follow as below figure to setup jumper.
(OTA mode work in normal mode.)

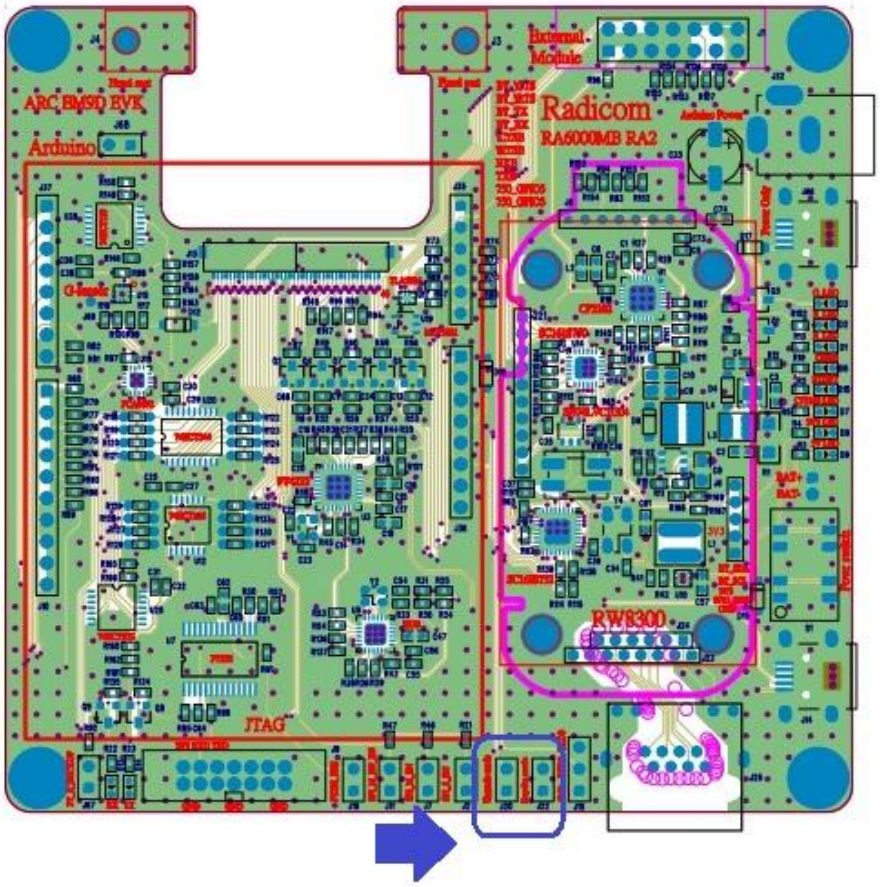


Figure 72 J20 and J22 location

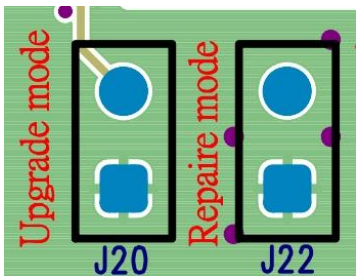


Figure 73 J20 and J22 don’t short

Figure 74 RA6000C-M SW1 all turn off

Figure 75 OTA mode physical diagram

Please follow as below steps to upgrade firmware by OTA.

Step 1: Copy layout.bin and output.img to "**$(OTA_TOOL_PATH)/img/**"
Note: layout.bin and output.img files are in "$(SDK_PATH)/tools/image_gen_cstm/output/"

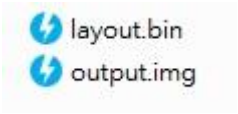


Figure 76 Copy files to img path

Step 2: Open ota tool.

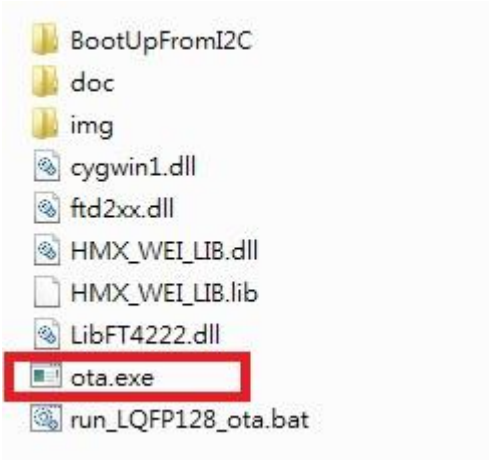


Figure 77 Execute OTA tool

Step 3: Type "**1**" and push "Enter" key, then Type "1" and push "Enter" key.

Figure 78 Type 1 to enter to Command item

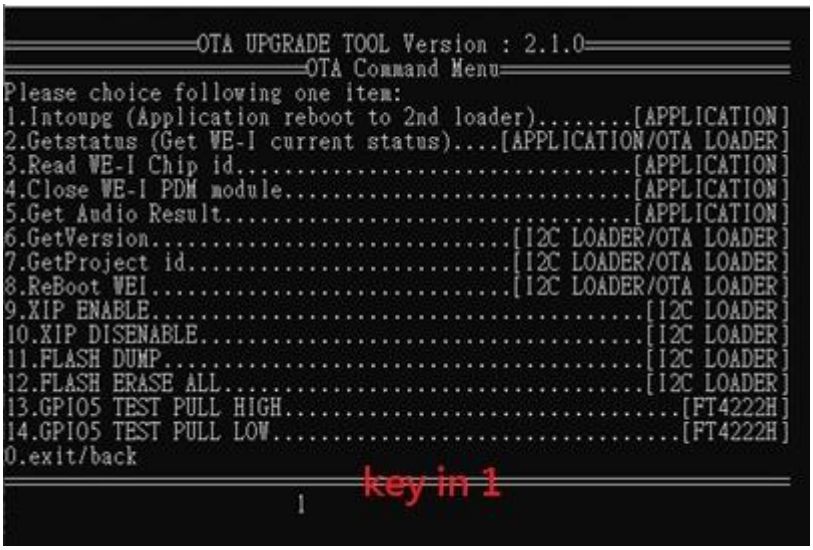


Figure 79 Type 1 to Intoupg

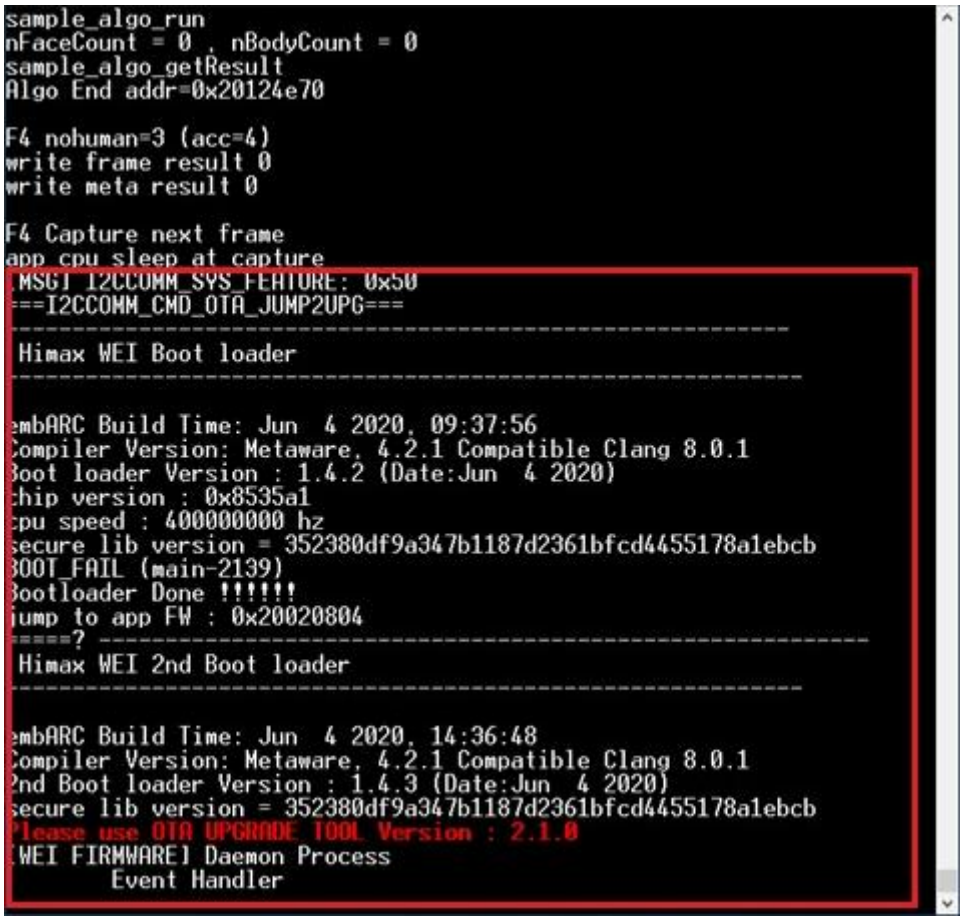


Figure 80 Debug message

Step 4: Type “0” and push “Enter” key, then type “2” and push “Enter” key.

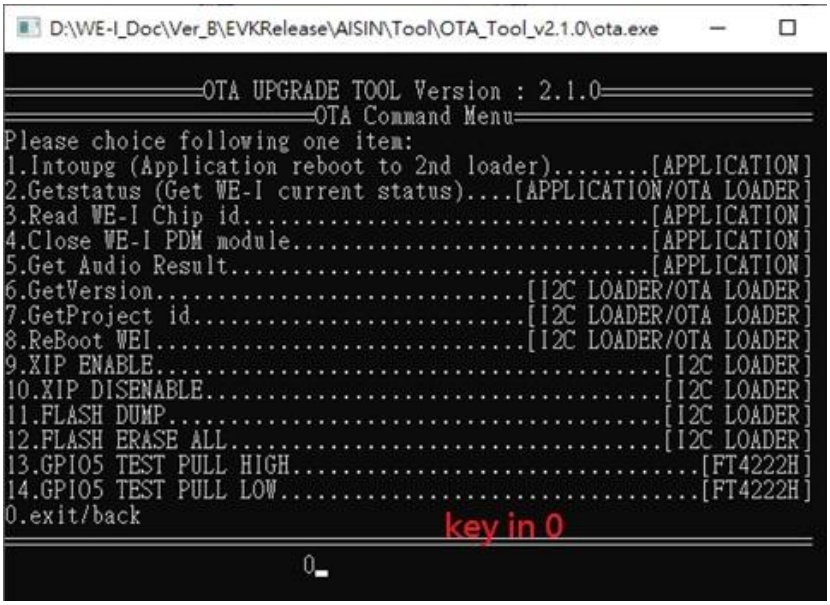


Figure 81 Type “0” to exit Command menu

Figure 82 Type “2” to into Upgrade menu

Step 5: Type “2” and push “Enter” key to OTA firmware.

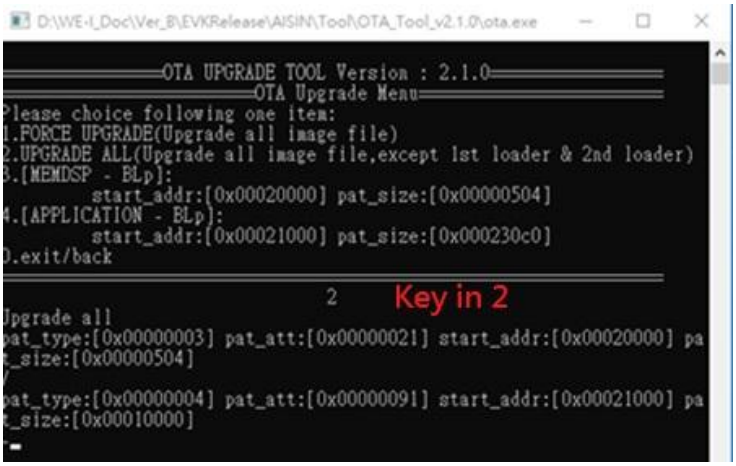


Figure 83 Type “2” to upgrade firmware

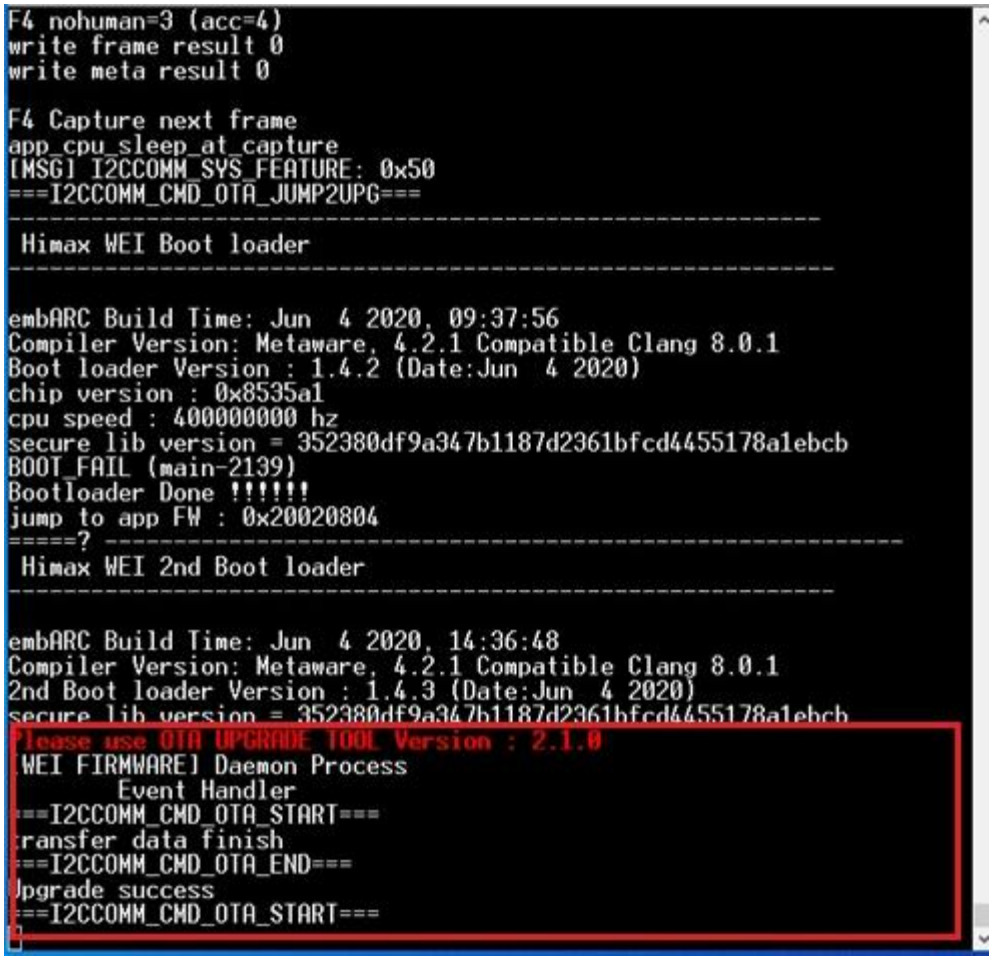


Figure 84 Debug message of OTA

B. FT4222 Programming

RA6000 series EVK upgrade firmware by FT4222H, please follow Figure 85 to setup upgrade firmware environment. The operating system of PC/NB is Windows (Windows 7 or Windows 10)

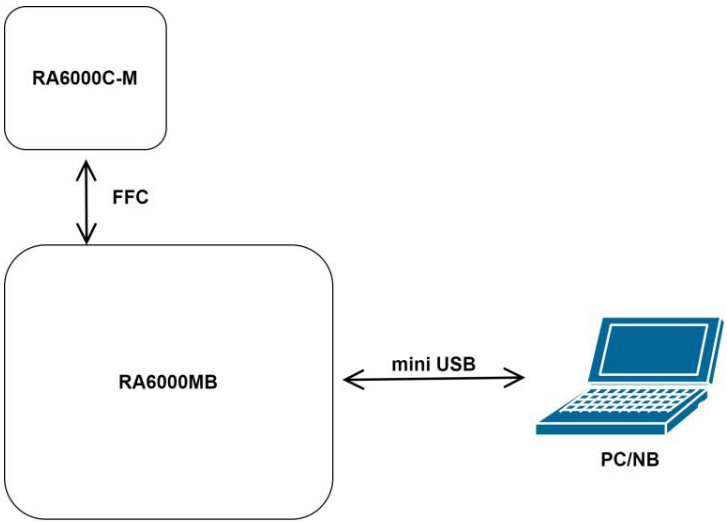


Figure 85 upgrade firmware environment

Please follow steps to upgrade firmware

Step 1: Check RA6000C-M SW1 to be all turn off. And RA6000 series EVK J20 to be shorted.

Figure 86 RA6000C-M SW1 all turn off

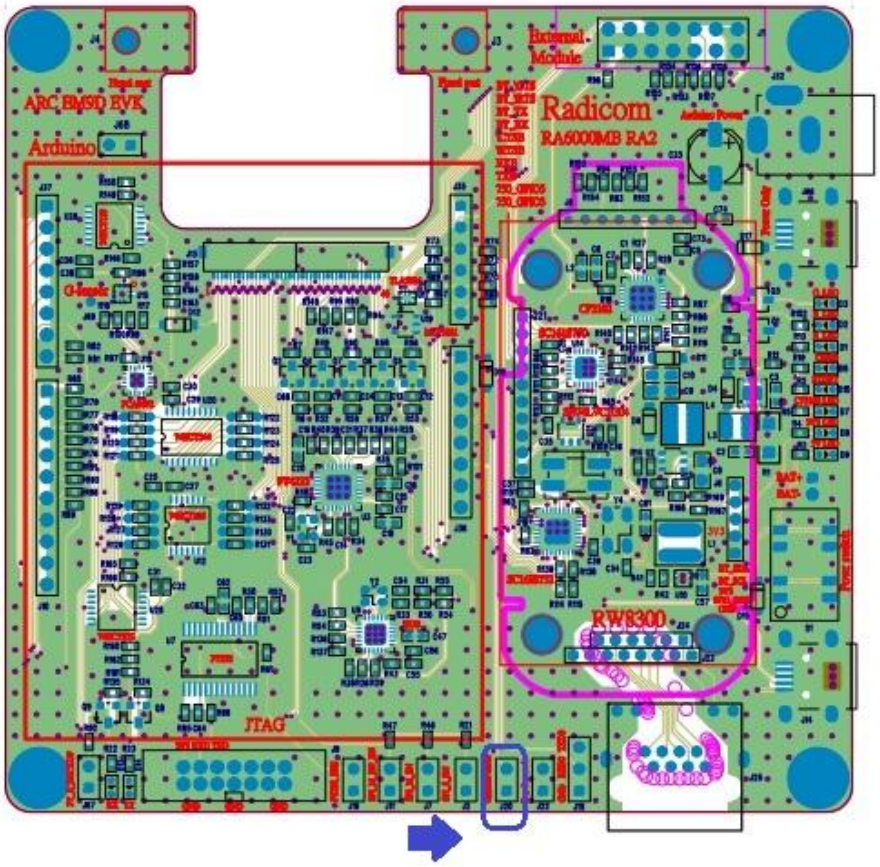


Figure 87 J20 physical location

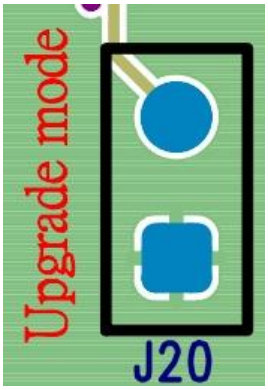


Figure 88 J20 jumper is shorted

Step 2: Check RA6000 series EVK J11 to be shorted.

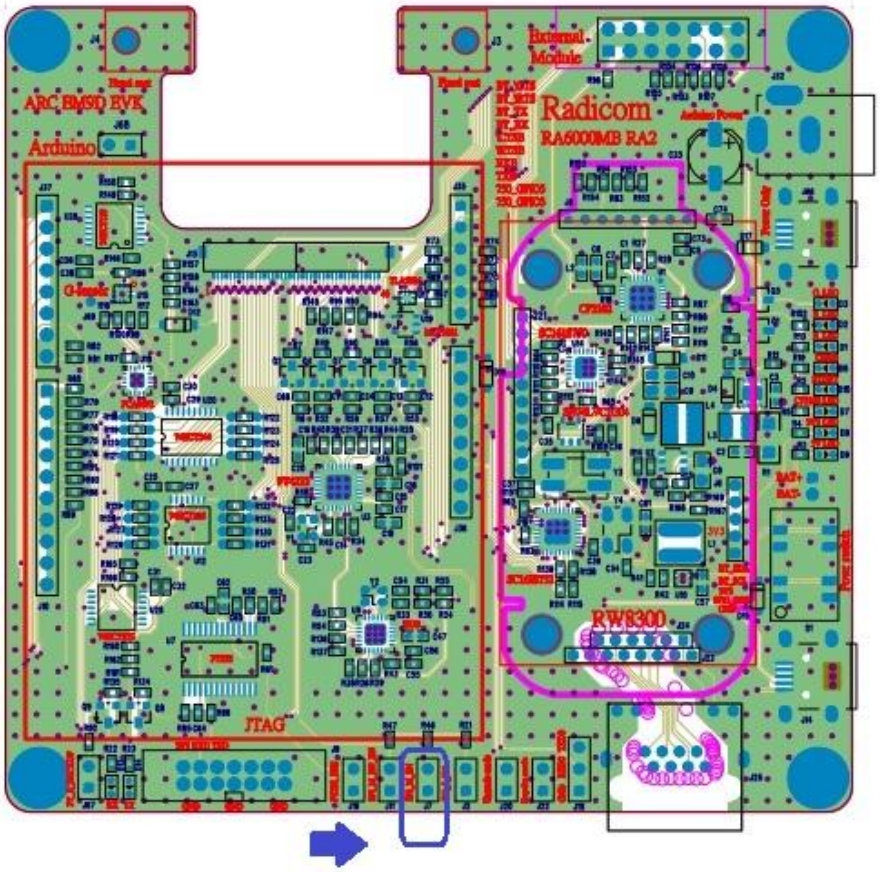


Figure 89 J11 physical location

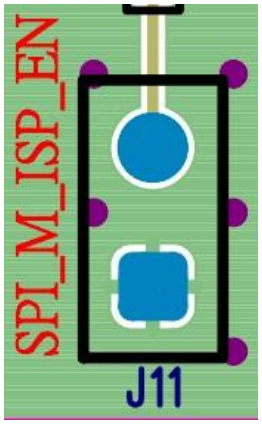


Figure 90 J11 jumper is shorted

Figure 91 upgrade mode physical diagram

Step 3: Plug-in micro USB to PC/NB, and check driver (FT4222H) to be installed.

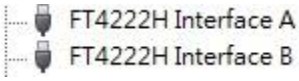


Figure 92 Check FT4222H Device

Note: On Windows 10, the PC/NB must install "vc_redist" package.

Figure 93 vc_redist.x86.exe

Step 4: Execute program "**HMX_FT4222H_GUI**"

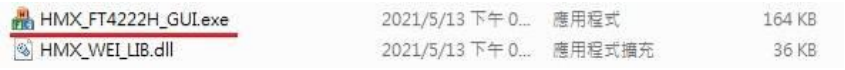


Figure 94 HMX_FT4222H_GUI

Step 5: Click "**Flash download"** label, and click "**Read ID**" button. Then check ID to be "**ef/60/15**".

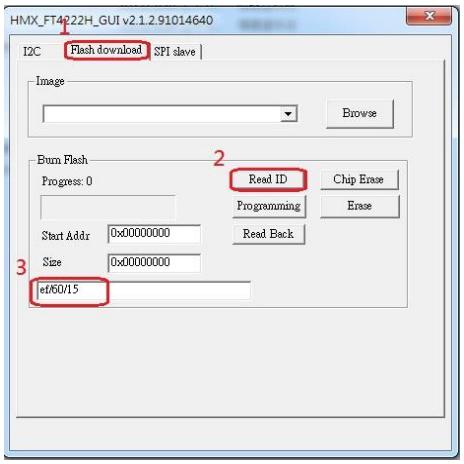


Figure 95 Check ID

Step 6: Click "Chip Erase" to erase flash.

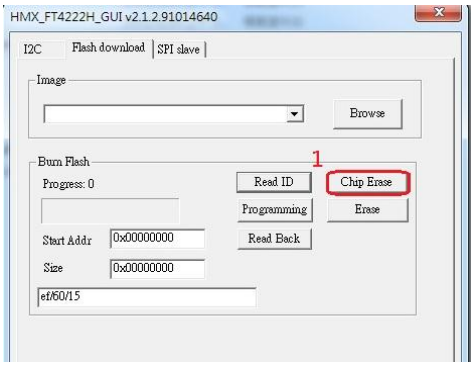


Figure 96 Chip Erase

Step 7 Click “Browse” to open image file (output.img). Then click Programming to upgrade firmware.

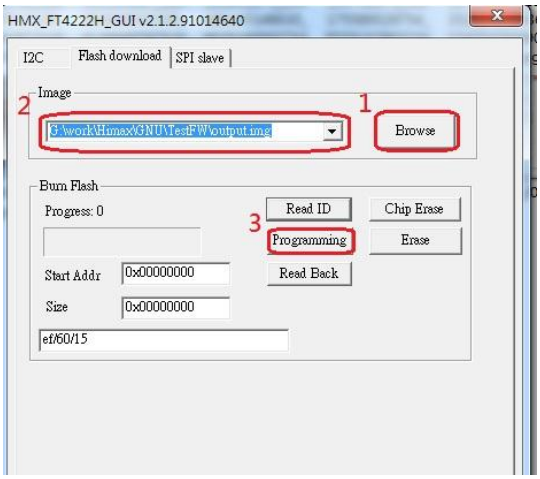


Figure 97 Upgrade firmware

C. **Repair mode**

RA6000 series EVK repair bootloader by FT4222H, please follow Figure 98 to setup repair environment. The operating system of PC/NB is Windows (Windows 7 or Windows 10)

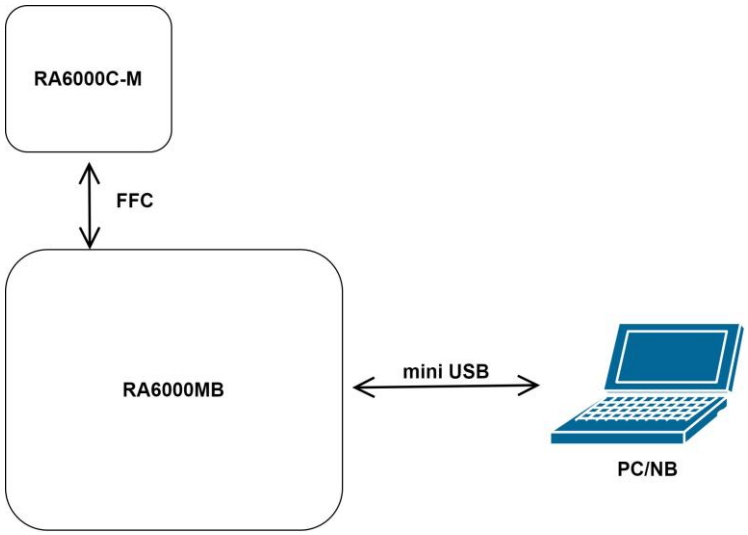


Figure 98 Repair environment

Please follow steps to repair bootloader

Step 1: Check RA6000C-M SW1 to be all turn off. And RA6000 series EVK J22 to be shorted.

Figure 99 RA6000C-M SW1 all turn off

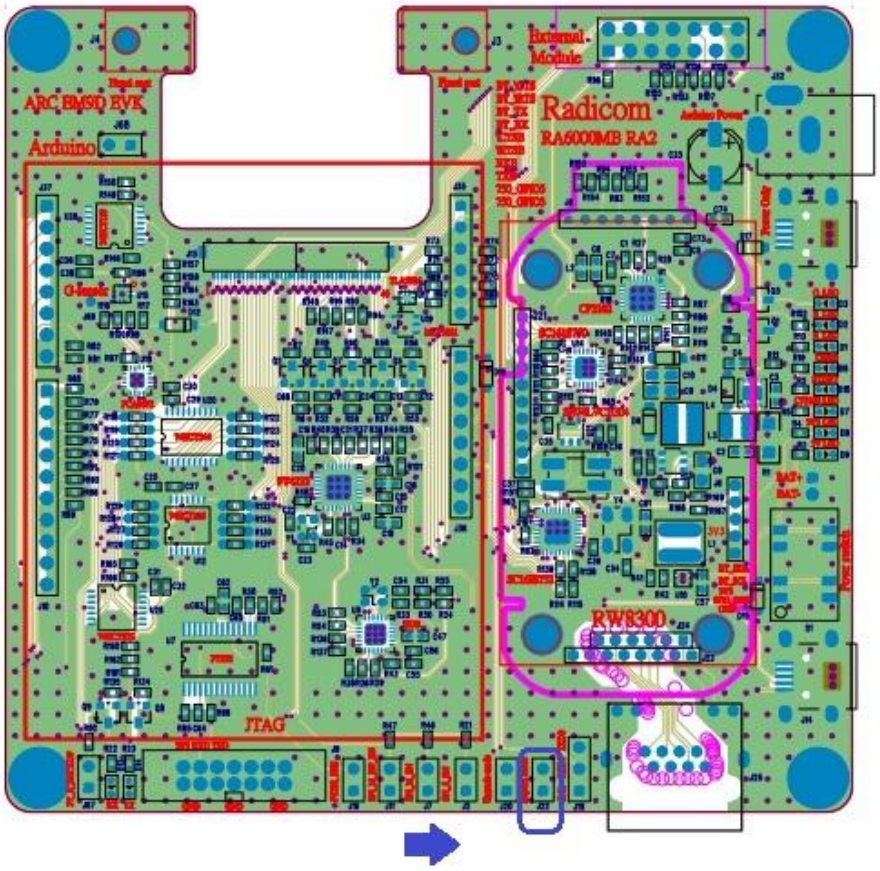


Figure 100 J22 physical location

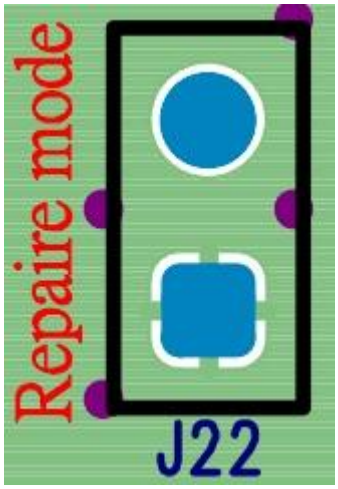


Figure 101 J22 jumper connection

Figure 102 Repair mode physical diagram

Step 2: Execute “ota.exe” tool. (OTA tool version is v2.1.0)

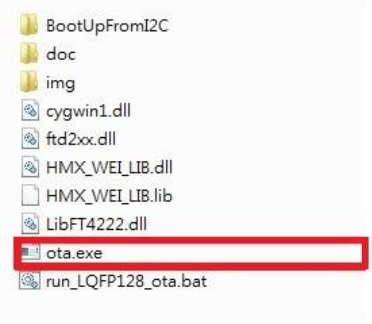


Figure 103 Execute ota tool

Step 3: Type “3” and push “**Enter**” key.

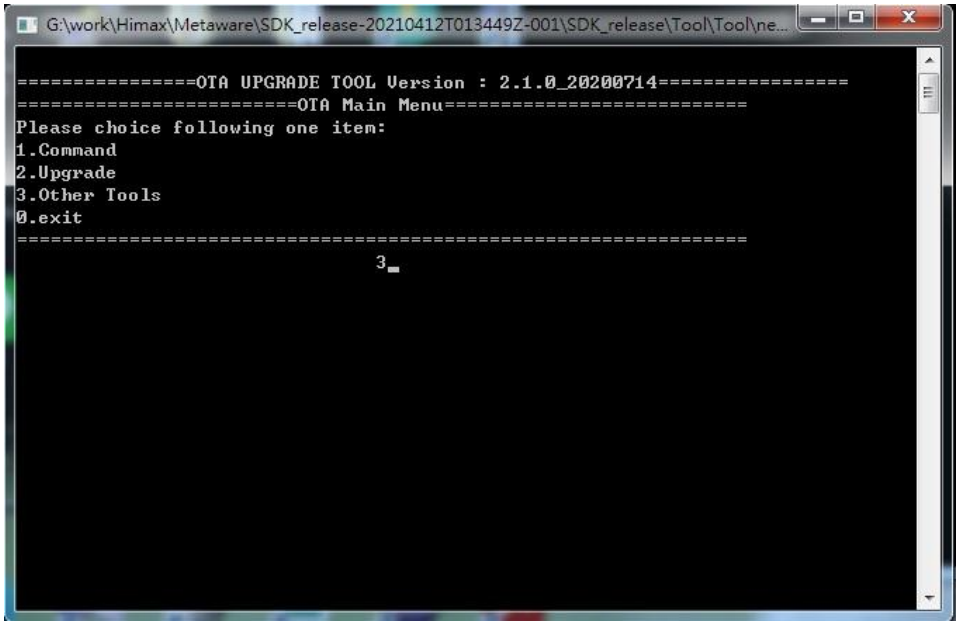


Figure 104 OTA Main Menu

Step 4: Type “1” and push “**Enter**” key.

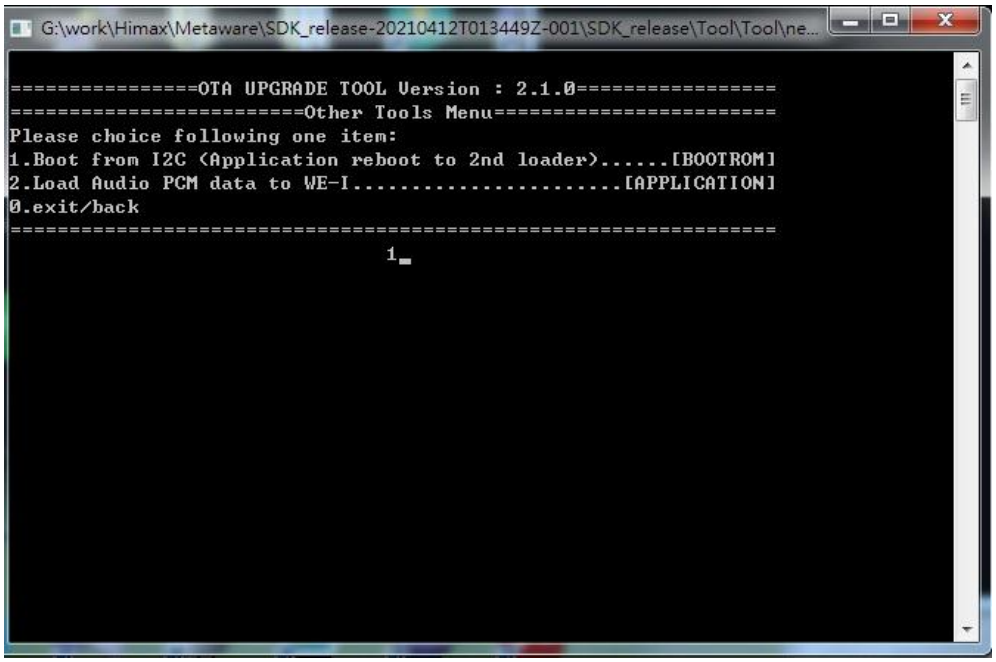


Figure 105 Other Tools Menu

Step 5: Type "**BootUpFromI2C\HX6539-A04TLDG-1111K_LQFP128_Debug\sign_formal_PA8530_EM9D_2nd_Bootloader.bin**" and push "**Enter**" key

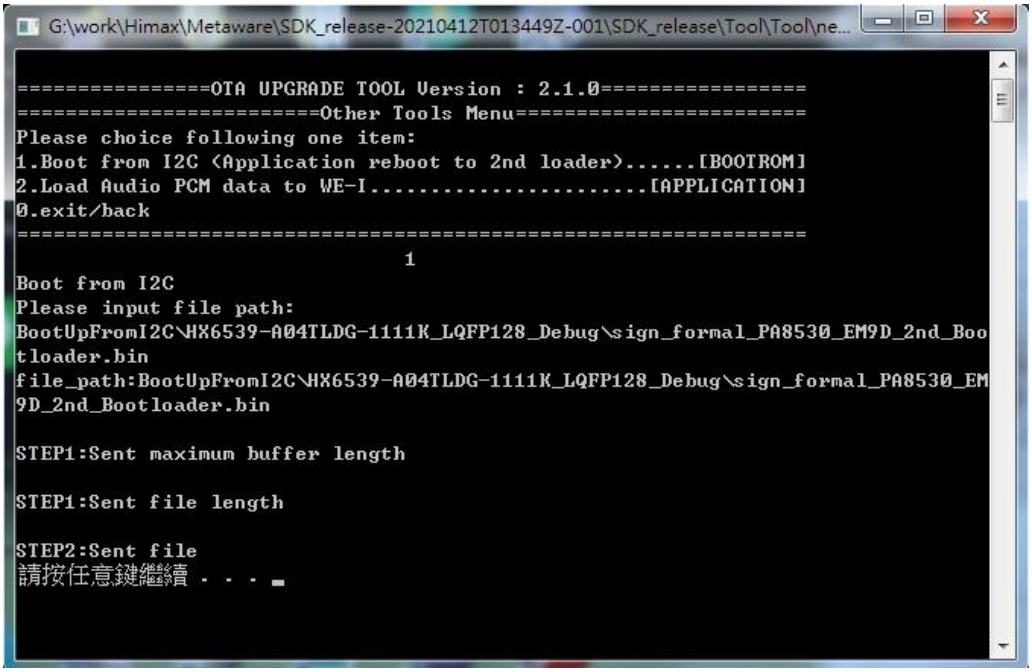


Figure 106 repair bootloader

Step 6: Debug UART will print message.

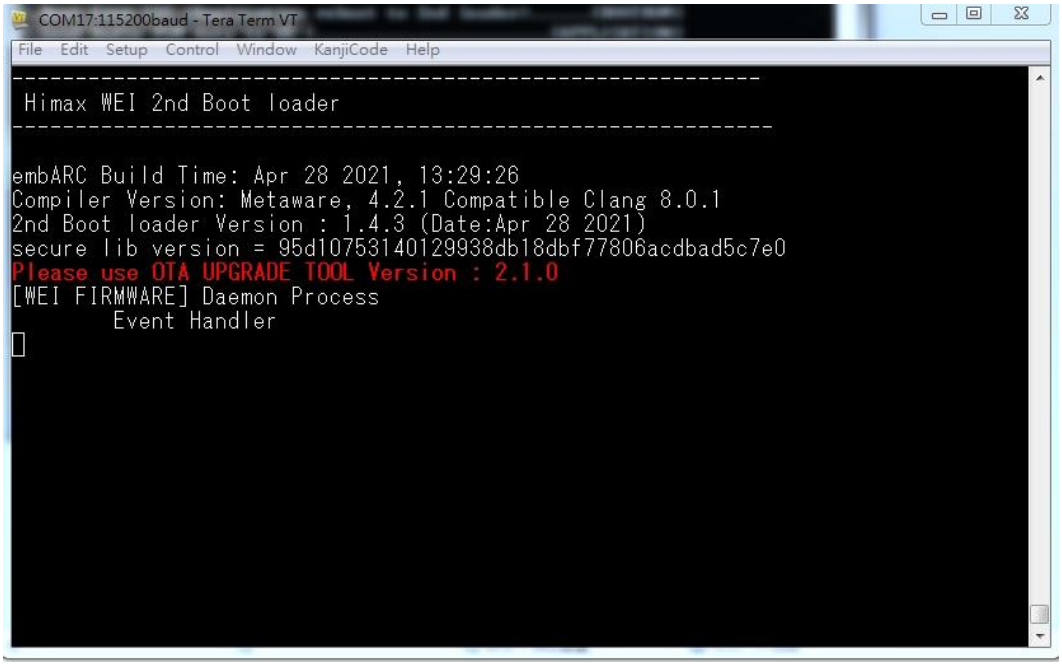


Figure 107 Debug UART message

D. JTAG Debug tool

The chapter will be shown how to use JTAG to debug by Metaware and GNU. For this example, the JTAG use HS2 JTAG as Figure 108. Please follow as below Figure 109 to setup JATG debug environment.

Figure 108 HS2 JTAG physical diagram

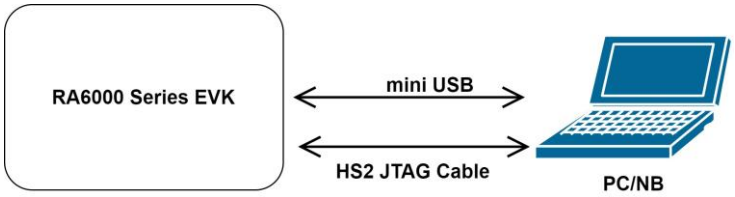


Figure 109 HS2 JTAG debug environment

Note: PC/NB need to install "Metaware" and "Digilent adept"
Please follow as below figure to setup JTAG and jumper.
(JTAG work in normal mode.)

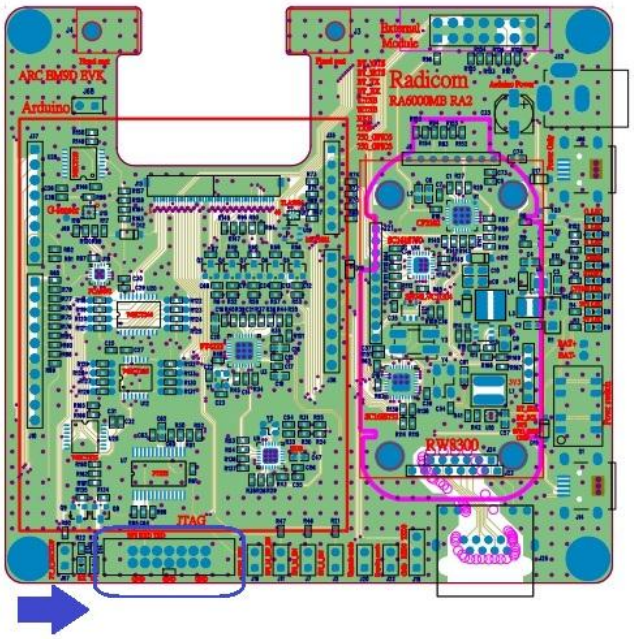


Figure 110 Jumper and socket location

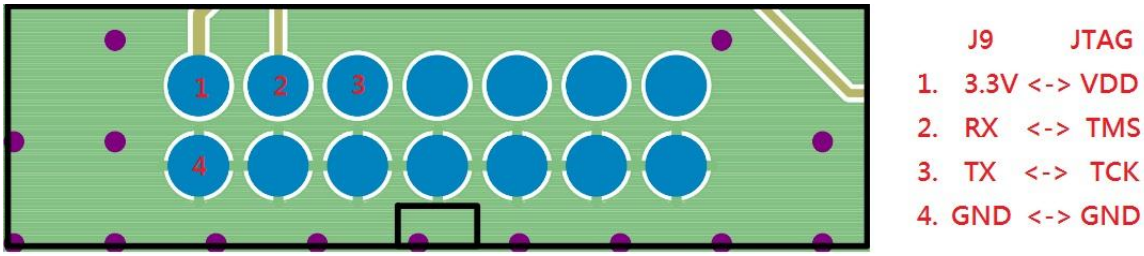


Figure 111 J9 pin define

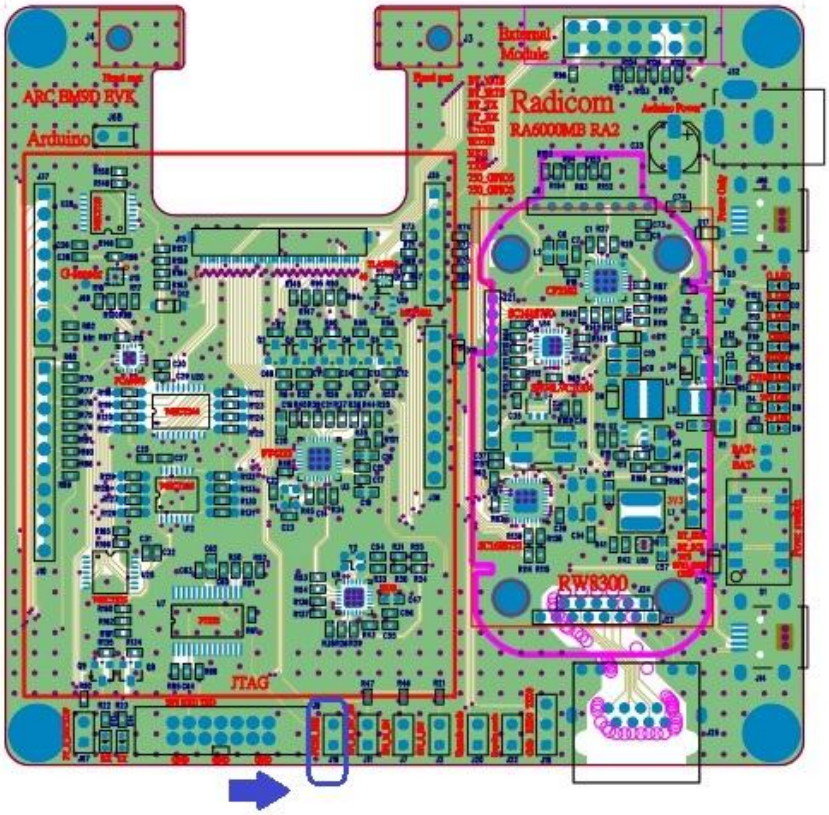


Figure 112 J19 location

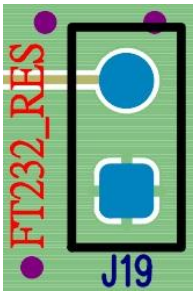


Figure 113 J19 pin is shorted

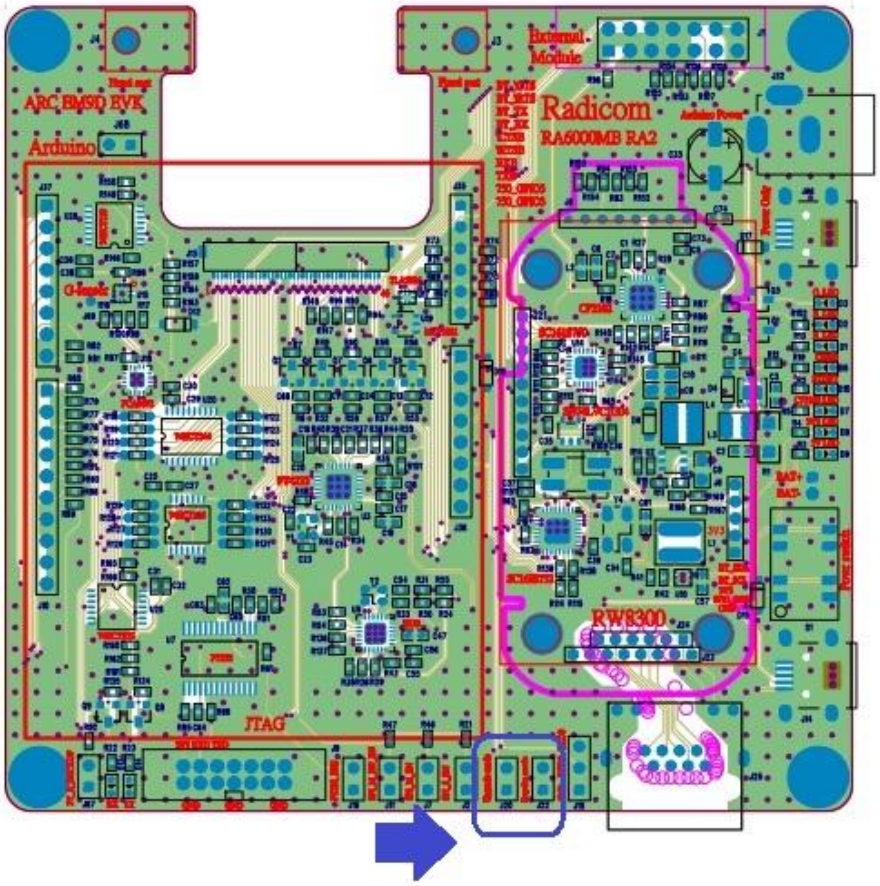


Figure 114 J20 and J22 location

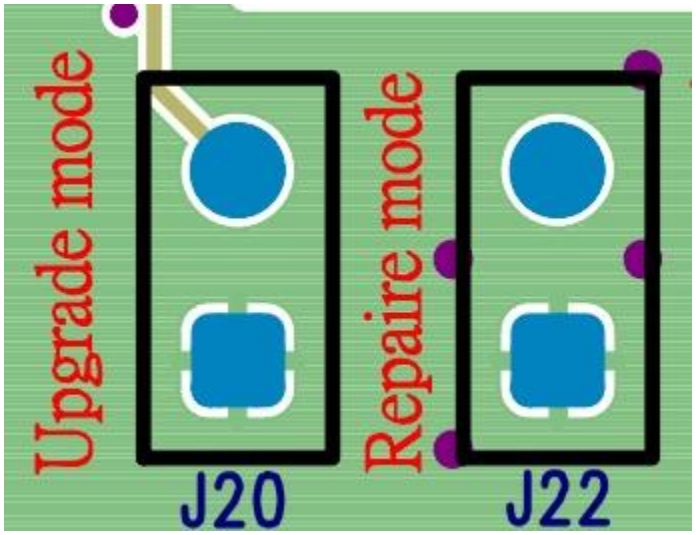


Figure 115 J20 and J22 don't be shorted

Figure 116 RA6000C-M SW1 all turn off

Figure 117 JTAG setting physical diagram

D.1 JTAG by Metaware

Please follow as below steps to build and run JTAG.

Step 1: Download and Install “digilent.adept.system” under PC/NB.
(url: https://reference.digilentinc.com/software/adept/start?redirect=1)

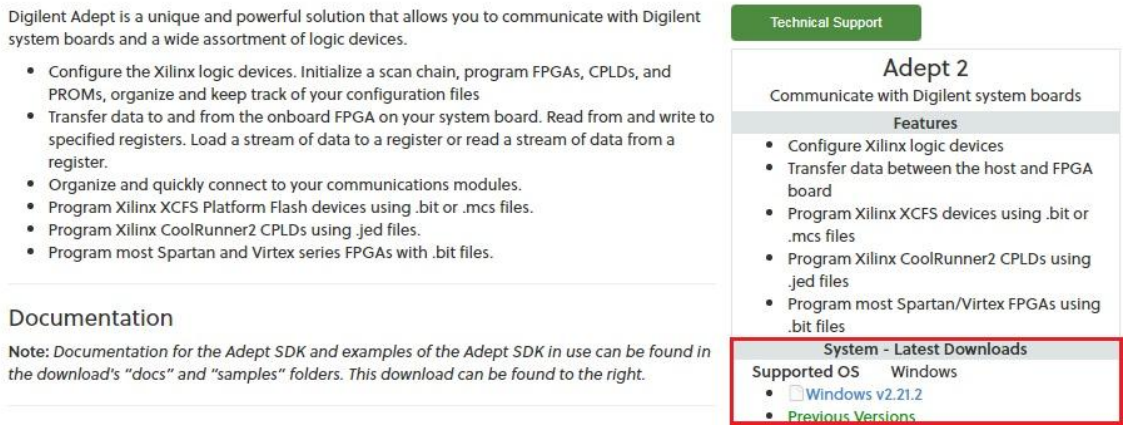


Figure 118 Click windows v2.21.2

Figure 119 Download

Figure 120 digilent.adept.system

Step 2: Open Metaware, and open project.

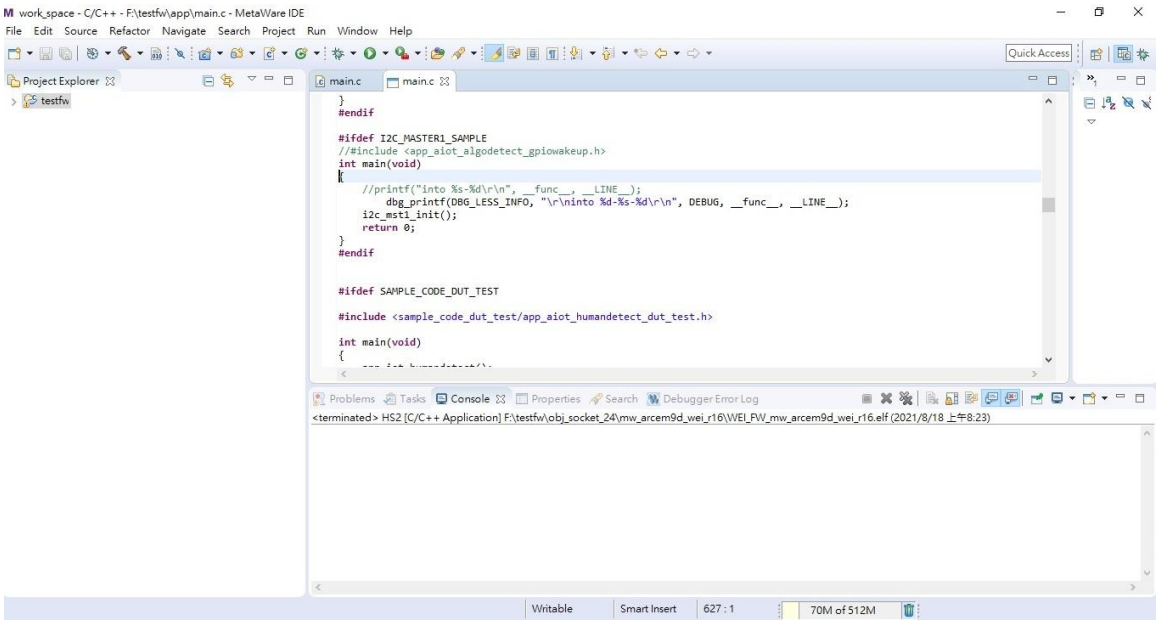


Figure 121 Open project

Step 2: Power on RA6000 series EVK and boot from normal mode.

Step 3: Open GUI Tool and Click "**Browse**" button.

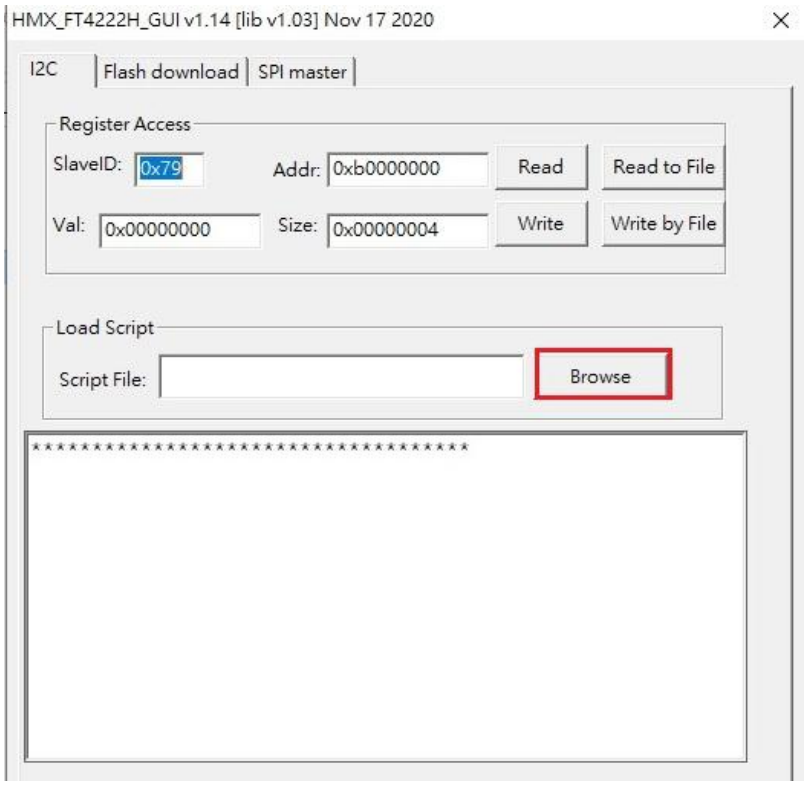


Figure 122 GUI Tool

Step 4: Select Script "**PLL_Script_24to400MHz_JTGA_B.txt**"

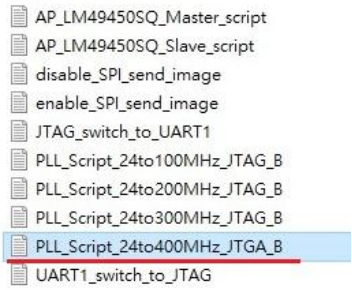


Figure 123 GUI Tool

Step 5: Read "**0xb0000044**" to check value to be "**0xc0000100**"
Note: If value is not "0xc0000100", please turn off EVK and return to step 2.

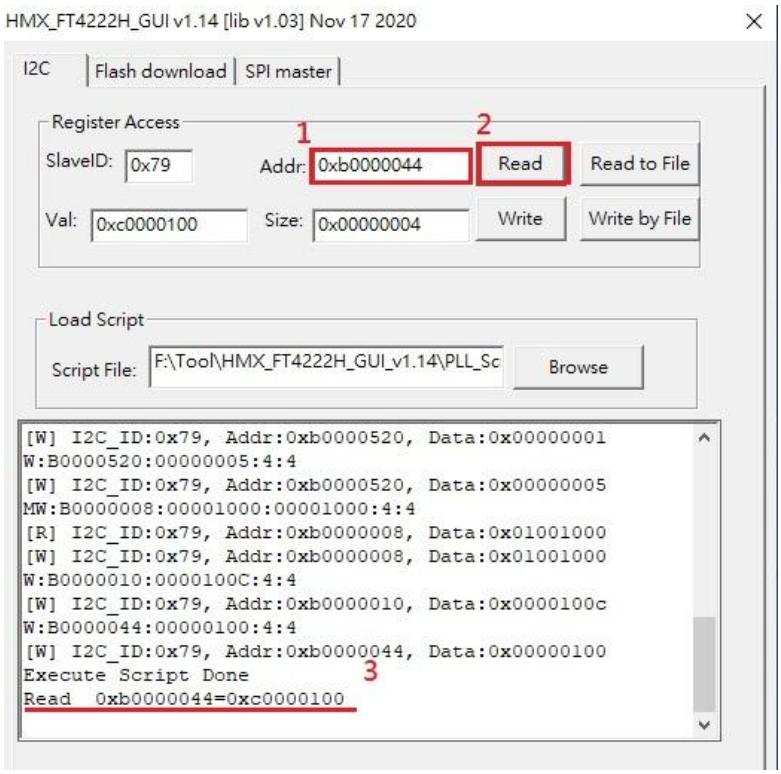


Figure 124 GUI Tool

Step 6:Right click project name, and click "**Debug As**" > "**Debug Configurations…**"

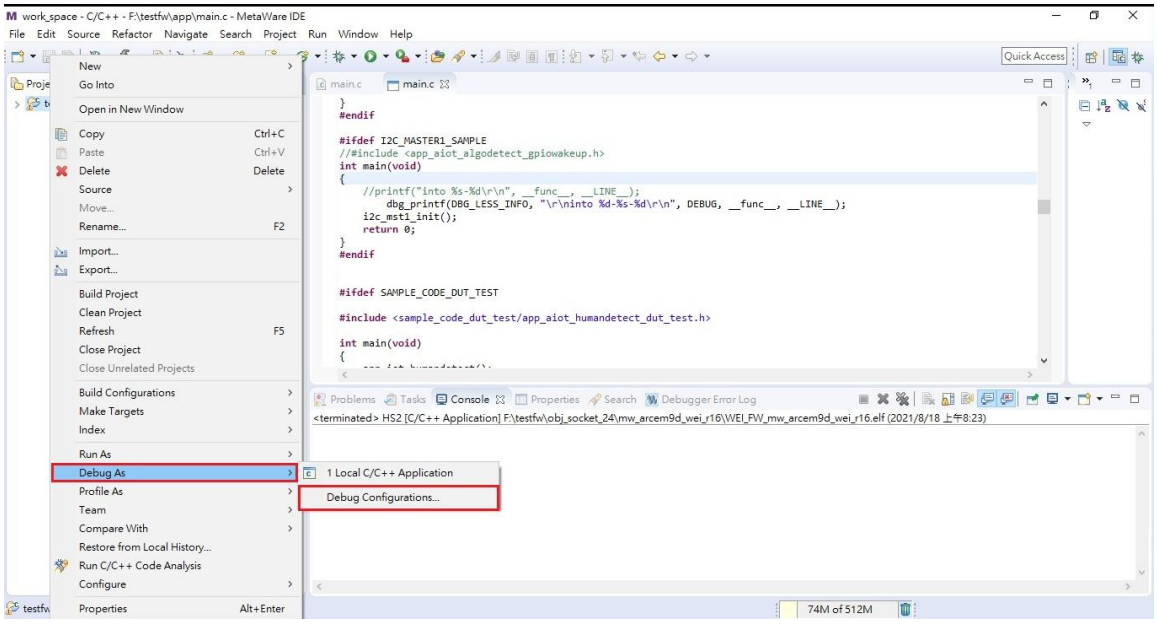


Figure 125 Open Debug Configurations…

Step 7: Right click "**C/C++ Application**" and click "**New**"

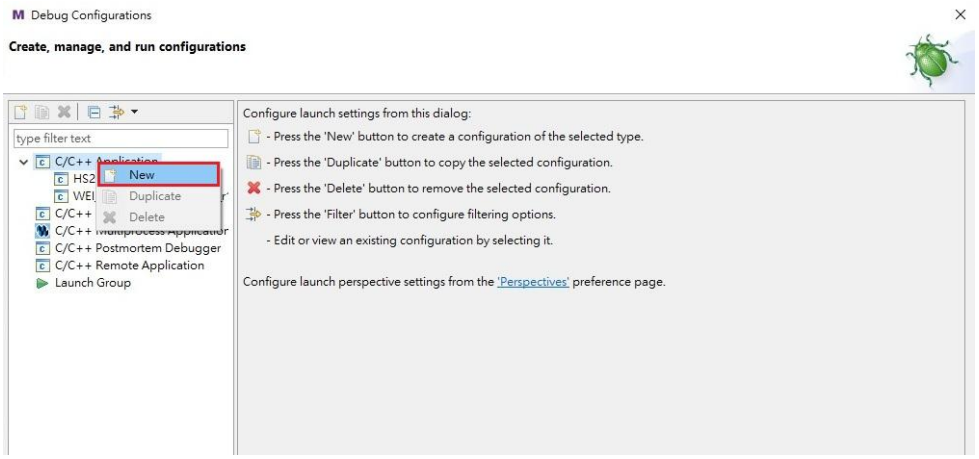


Figure 126 New Debug

Step 8: Click “Browse…” button and select elf file from project.

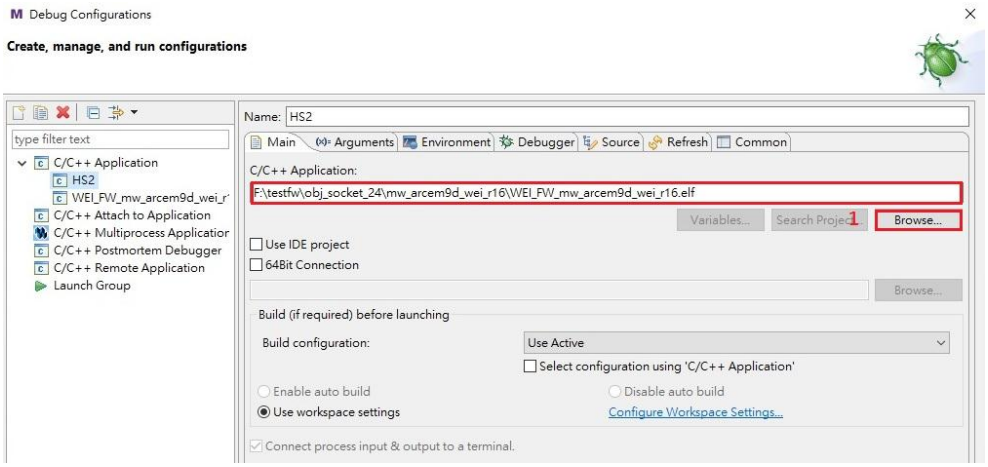


Figure 127 Set elf file path

Step 9: Click “**Debugger**” label and follow as figure to set parameters. Then click “Debug”.

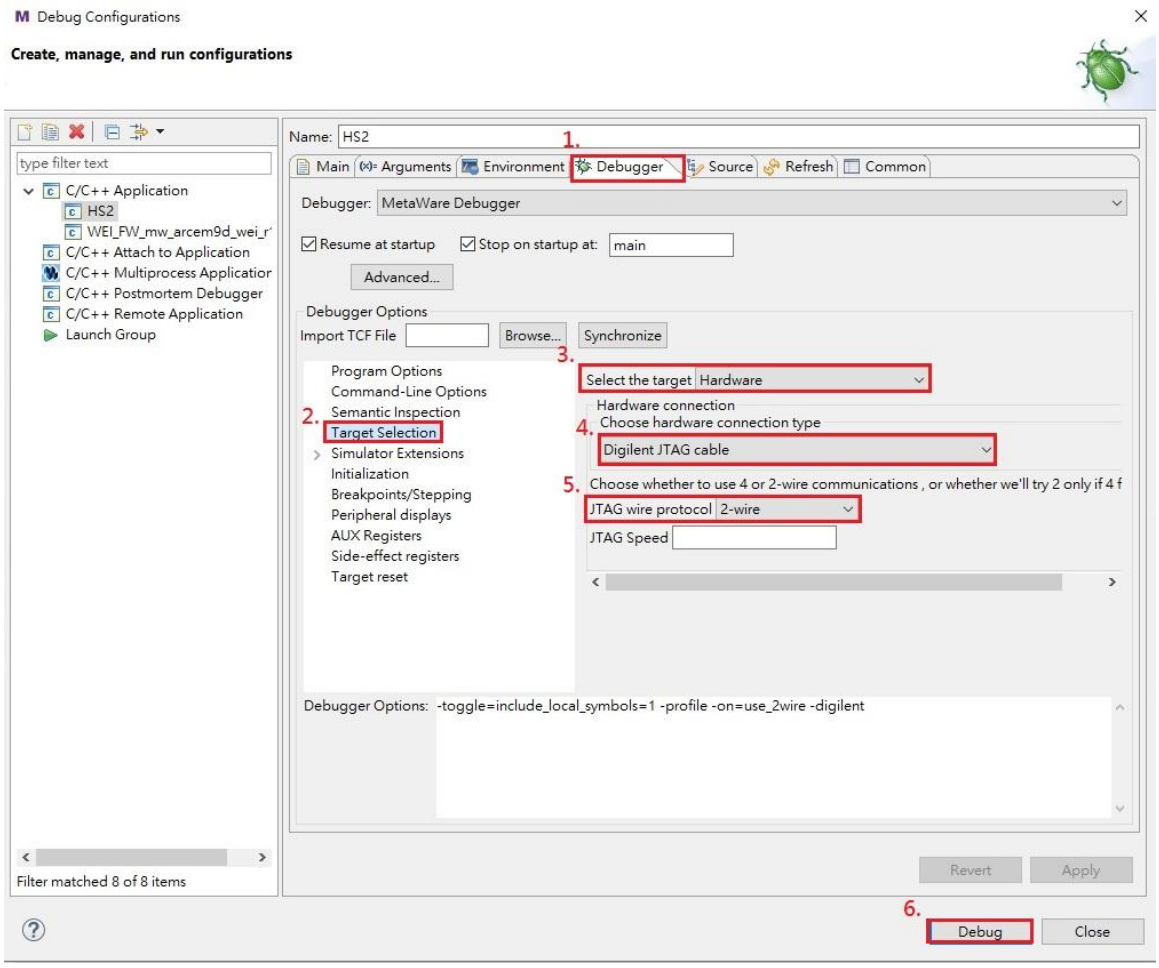


Figure 128 Set Target parameters

Step 10: Click “**Yes**” to open debug perspective.

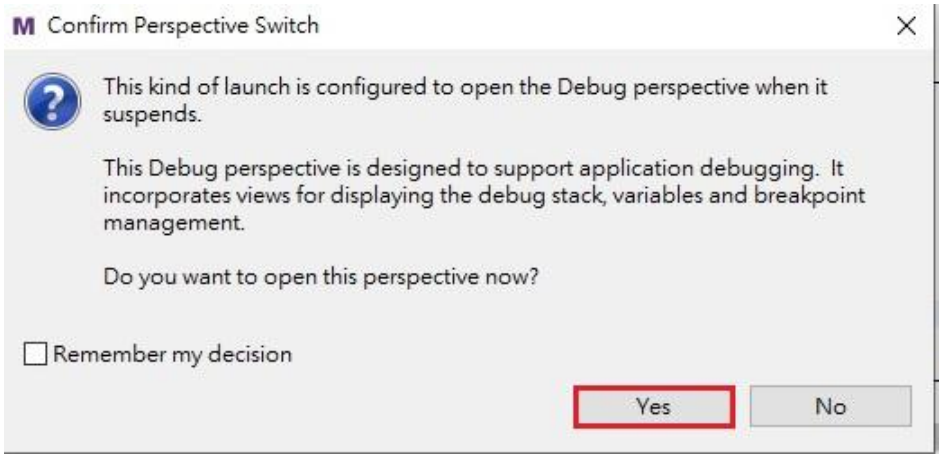


Figure 129 Confirm Perspective Switch

Step 11: Click “**Run**” > “**Step Into** “or “**Step Return**” to debug project.

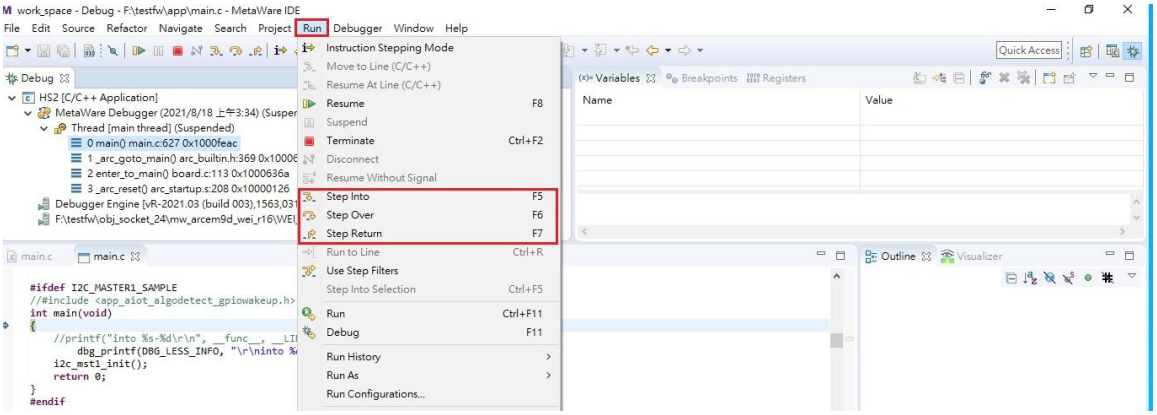


Figure 130 Confirm Perspective Switch

Step 12: Select variable and right click, then click “Add Watch Expression…” to observe variable.

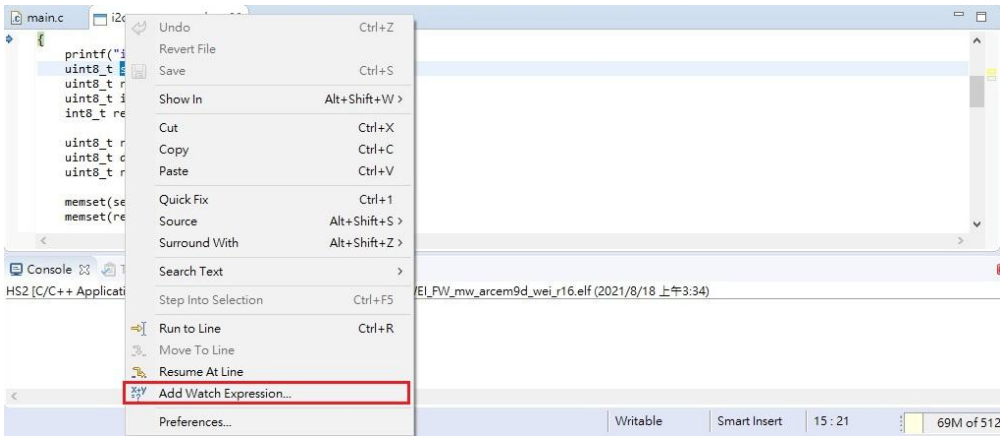


Figure 131 Confirm Perspective Switch

Figure 132 variable name

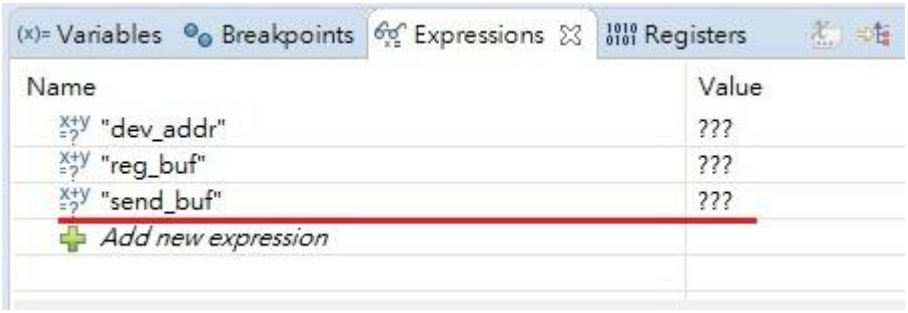


Figure 133 Expressions window

Note: If JTAG execute unsuccessfully or exit illegally, please reboot PC/NB and return to step 1.

D.2 JTAG by GNU

Please follow as below steps to build and run JTAG.
Note: this sample is worked under Ubuntu 20.04.

Step 1: Download and decompress GNU Toolchain package. (version: 2021.03)
(url: https://github.com/foss-for-synopsys-dwc-arc-processors/toolchain/releases/)

Figure 134 Download GNU Toolchain package

Step 2: Download and decompress FT4222 Linux driver. And copy "**a.out**" and "**UART1_switch_to_JTAG.txt**" to **<ft4222 driver location>/examples.**
Note: Radicom will provide "a.out" and "UART1_switch_to_JTAG.txt" files.
(url: https://ftdichip.com/software-examples/ft4222h-software-examples/)

Figure 135 Download FT4222 Linux driver

Step 3: Open a terminal, and type "**sudo apt-get install libncursesw5 libncurses5 libncurses5:i386 libmpfr-dev**" and "**sudo ln -s /usr/lib/x86_64-linux-gnu/libmpfr.so.6 /usr/lib/x86_64-linux-gnu/libmpfr.so.4**" to install library and create soft link.

Step 4: Modify SDK script file. The file path is "**/options/toolchain/toolchain_gnu.mk**". "**-Wl, --strip-debug**" will be removed. Then rebuild source code.

```
        LINK_OPT += -Wl,--print-memory-usage
        LINK_OPT += -fstack-usage
        #LINK_OPT += -Wl,-marcv2elfx -Wl,--strip-debug -Wl,--stats,--gc-section
s
        LINK_OPT += -Wl,-marcv2elfx -Wl,--stats,--gc-sections
```

Figure 136 Modify toolchain_gnu.mk

Step 5: Enter to <**ft4222 driver location**>, and type "**sudo ./install4222.sh**" to install FT4222 driver.

Step 6: Enter to **<arc ide location>/bin**, type "**cp arc-linux-gdb /usr/bin/.**" and "**cp openocd /usr/bin/.**".

Step 7: Type "**arc-linux-gdb --version**" and "**openocd --version**" to check tool work.

```
SDK3_1# openocd --version
Open On-Chip Debugger 0.9.0-dev-g842d4db-dirty (2020-10-20-18:59)
Licensed under GNU GPL v2
For bug reports, read
        http://openocd.sourceforge.net/doc/doxygen/bugs.html
```

Figure 137 openocd version

```
/SDK3_1# arc-linux-gdb --version
GNU gdb (ARCv2 ISA Linux uClibc toolchain 2020.09) 10.0.50.20200611-git
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html
>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

Figure 138 arc-linux-gdb version

Step 8: Power on EVK, and plug in HS2 JTAG. Type "**lsusb**" to check HS2 JTAG and FT4222 driver.

```
/SDK3_1# lsusb
Bus 002 Device 032: ID 0403:6014 Future Technology Devices International, L
td FT232H Single HS USB-UART/FIFO IC
Bus 002 Device 035: ID 0403:601c Future Technology Devices International, L
td
Bus 002 Device 034: ID 0403:6015 Future Technology Devices International, L
td Bridge(I2C/SPI/UART/FIFO)
Bus 002 Device 033: ID 04b4:6570 Cypress Semiconductor Corp. Unprogrammed C
/7C65632/34 hub HX2VL
Bus 002 Device 003: ID 046d:c52b Logitech, Inc. Unifying Receiver
```

Figure 139 Check JTAG and FT4222 driver

Step 9: Enter to **<FT4222 driver location>/examples,** type "**chmod 777 a.out**" and "**./a.out**".

```
root@leon-HP-EliteBook-8440p:/home/leon/Downloads/libft4222-linux/examples# ./a.
out

Device 1 is interface A of mode-0 FT4222H:
  0x0403601c    FT4222 A
Chip version: 42220400, LibFT4222 version: 0104042C
Packet Init Value OK
Invalid Command !!
Invalid Command !!
ID:79
W:b0000044:00000100:4:4
Write Addr=0xb0000044, Data=0x00000100
Invalid Command !!
Skipping interface B of mode-0 FT4222H.
root@leon-HP-EliteBook-8440p:/home/leon/Downloads/libft4222-linux/examples#
```

Figure 140 Execute a.out script

Step 10: Enter to **<arc ide location >/share/openocd/scripts**, type "**openocd -c 'gdb_port 4910' -f board/snps_em_sk_v2.3_cjtag.cfg**".

Figure 141 Execute a.out script

Step 11: Enter to **<sdk path>/obj_socket_24/gnu_arcem9d_wei_r16/**, type "**arc-linux-gdb WEI_FW_gnu_arcem9d_wei_r16.elf**".

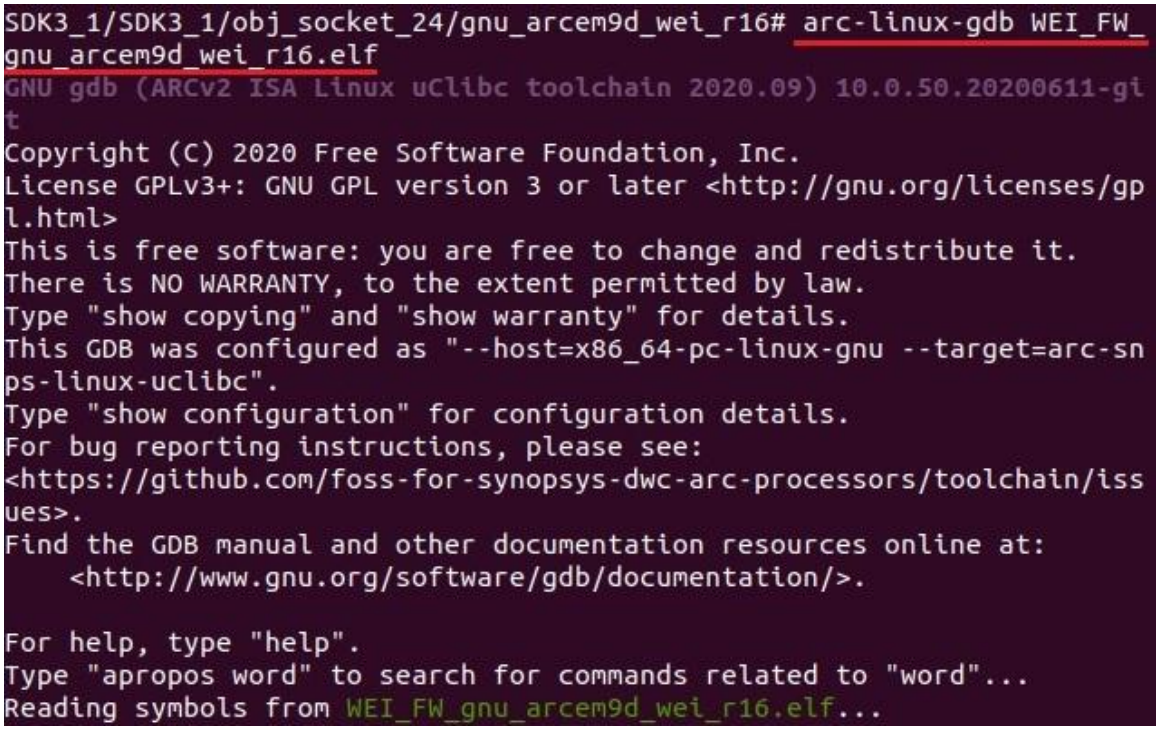


Figure 142 Execute a.out script

Step 11: And type gdb command:

(gdb) target remote :49101
(gdb) set remotetimeout 15
(gdb) directory <source path>
(gdb) load

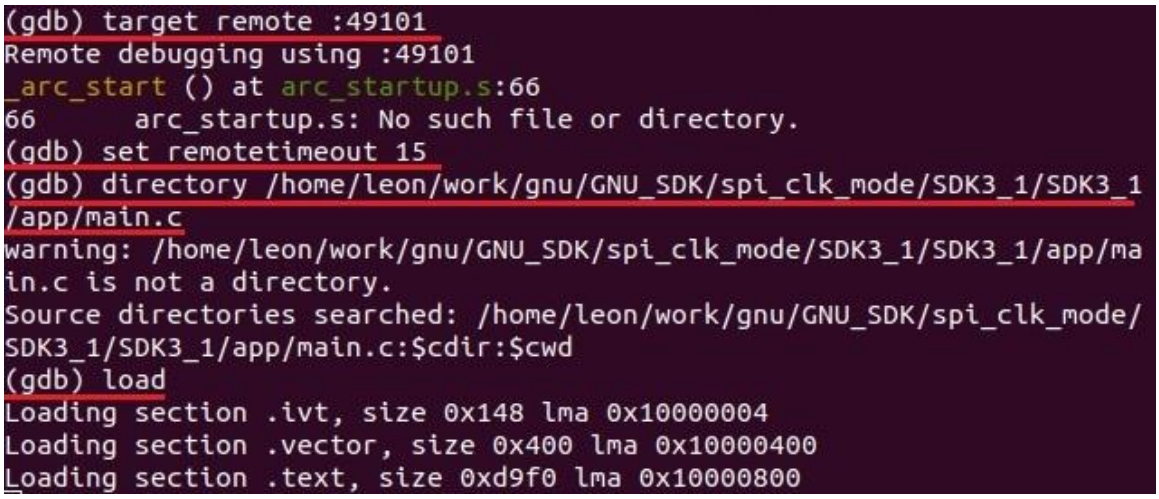


Figure 143 Execute a.out script

Step 12: Now, user can debug by gdb.

Table 15 openocd common command list

Command	Describe
gdb_port [number]	Set port number for GDB to connect OpenOCD.
interface <name>	Activate the driver of debug adapter.
adapter_khz <kHz>	Set debug adapter (JTAG) clock frequency.
interface_list	List all debug adapter that OpenOCD support.
targets [name]	List current target info.
target types	List all target device that OpenOCD support.
shutdown [error]	Close all OpenOCD connection.
debug_level [n]	Set debug level.
log_output [filename]	Store OpenOCD log message.
reset halt	Reset the target to halt status.

Table 16 gdb command list

Command	Describe
b	Puts a breakpoint at current line
b N	Puts a breakpoint at line N
d N	Deletes breakpoint number N
info break	list breakpoints
c	Continues running the program until the next breakpoint or error
step	Runs the next line of the program
n	Like s, but it does not step into functions
p var	Prints the current value of the variable "var"
display var	Prints the current value of the variable "var" every time
q	Quits gdb
directory	Load source code location
list	Show current line source code

There are 3 modules supported update firmware on RA6000 series EVK: RW8300, RB8762 and RTL8711AM

Check and upgrade RW8300 firmware:

Power on RA6000 series EVK, and connect the RJ45 cable to RW8300 and PC/NB. Please make sure the RW8300 is connected to Web server. Following as below figure to setup upgrade environment.

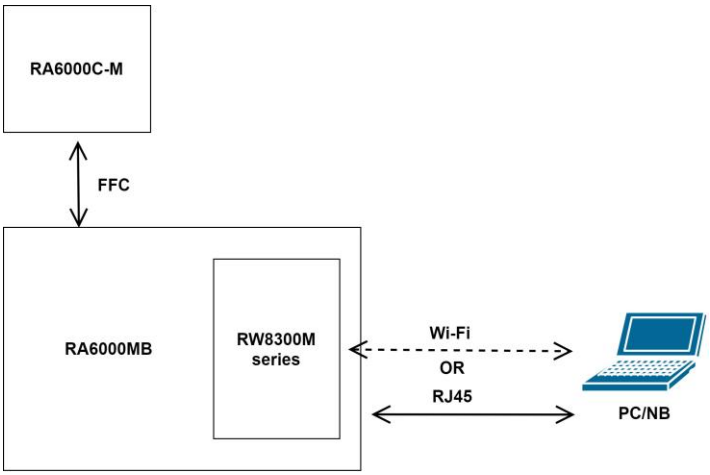


Figure 144 RW8300 upgrade firmware environment

Following as below steps to check and upgrade firmware.

Step 1: Open a browser and type the default gateway address (192.168.2.3).

Step 2: Go to **Management → Status** page to check current firmware version.

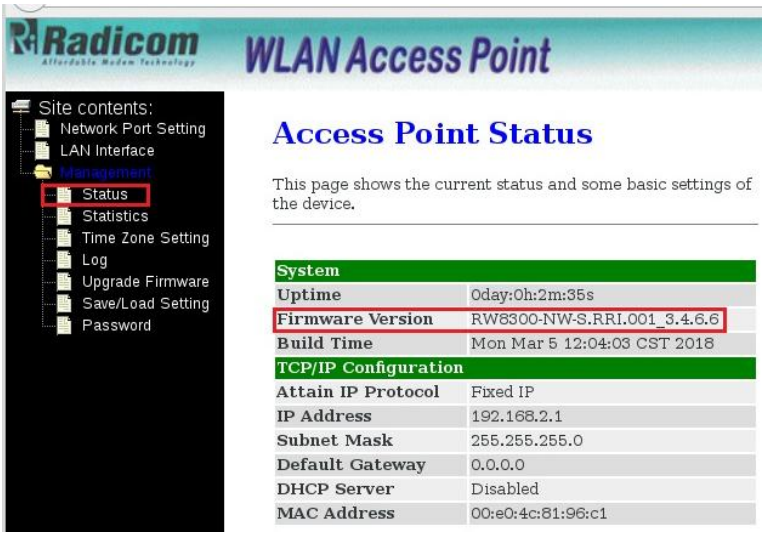


Figure 145 Network Status

Step 3: Go to **Management → Upgrade Firmware** page for upgrading firmware.

Click Browse to select the upgrading firmware location and click Upload.

Figure 146 Upgrade Firmware

Step 4: It will take approximately 95 seconds to complete the upgrading.

Warning: "DO NOT power down or interrupt the upgrading process."

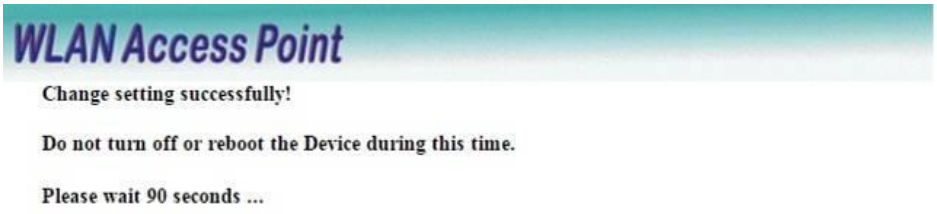


Figure 147 Countdown Page

Updating RTL8711AM firmware

OTA (Over-the-Air) programming is a methodology of updating device firmware remotely via TCP/IP network connection. In this version, RA6000C-M will send OTA command to RTL8711. Follow as Figure 148 to setup OTA environment.

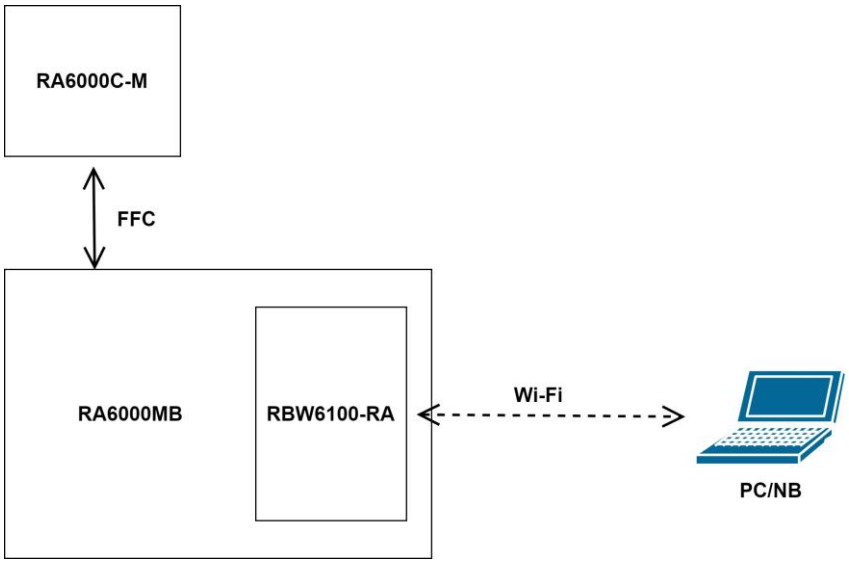


Figure 148 Setup OAT 8711 environment

Operate the following steps to update RTL8711AM firmware.

Step 1: Prepare DownloadServer.exe, start.bat and FW image for OTA under PC/NB.

Step 2: Refer Lab1_WiFiOTA sample from SDK.

Step 3: Type "**make**" and "**make flash**" to build firmware.

Step 4: Upgrade firmware to RA6000-EVK.

Step 5: Power on RA6000-EVK, and RBW6100-RA will open AP. The WiFi information will be shown in UART0 (debug UART).

```
      MODE => AP<CR><LF>
<CR>
      SSID => OTA<CR><LF>
<CR>
   CHANNEL => 11<CR><LF>
<CR>
  SECURITY => AES<CR><LF>
<CR>
  PASSWORD => 12345678<CR><LF>
<CR>
<CR><LF>
<CR>
Interface (wlan0)<CR><LF>
<CR>
============================<CR><LF>
<CR>
<HT>  MAC =>[redacted]<CR><LF>
<CR>
<HT>  IP  => 192.168.43.1<CR><LF>
<CR>
<HT>  GW  => 192.168.43.1<CR><LF>
```

Figure 149 WiFi information by UART0 (Debug UART)

```
# <CR>
<CR><LF>
Waiting command for OTA...<CR>
<CR><LF>
```

Figure 150 Waiting command by UART0 (Debug UART)

Step 6: Connect to 8711 WiFi under PC/NB.

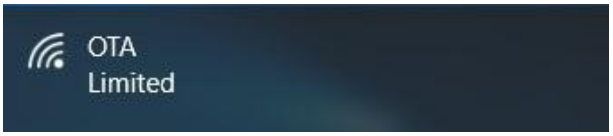


Figure 151 connect to 8711 WiFi

Step 7: Go to "**Network and Internet**" > "**Network Connections**", and Set IP address automatically.

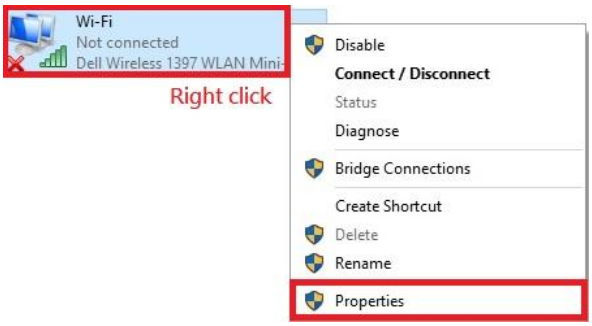


Figure 152 WiFi device

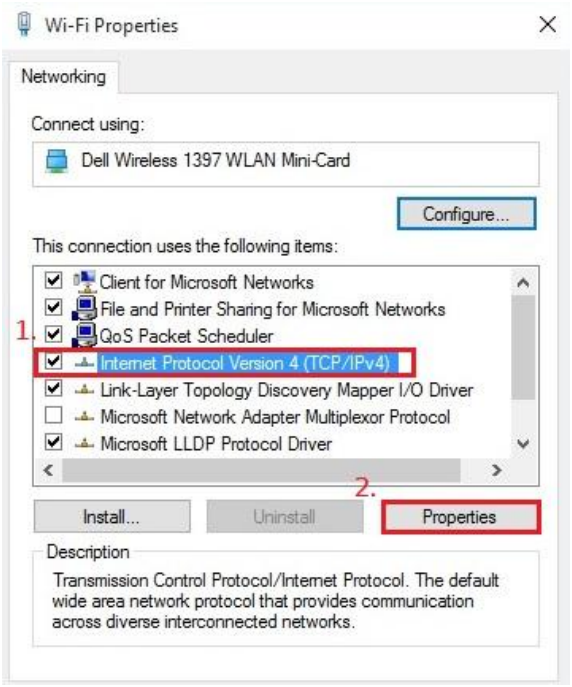


Figure 153 Properties

Figure 154 TCP/IPv4

Step 8: Open a command window, type “ipconfig” to check IP address.

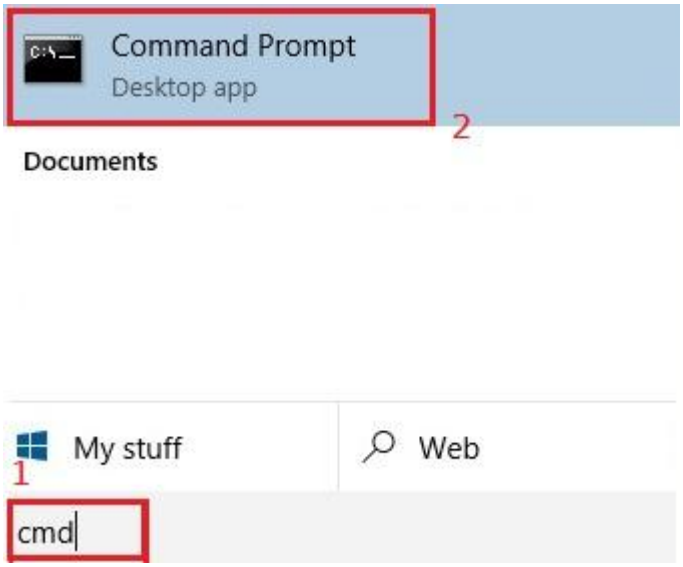


Figure 155 Open command prompt

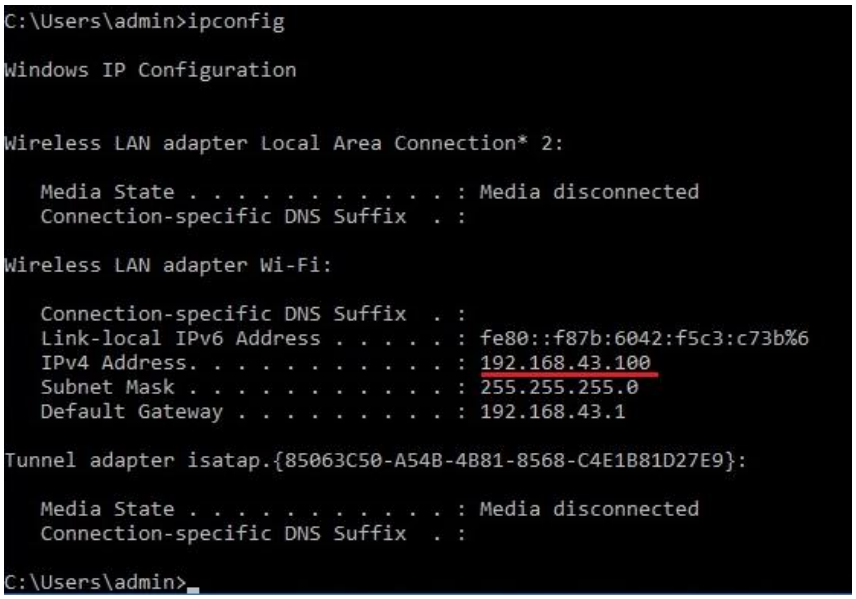


Figure 156 Check IP address

Step 9: Type "DownloadServer.exe 8082 RTL8711AM.CMD.RRI.1.3.5.O.4.0C.bin" to open "DownloadServer" under PC/NB. (Command format: **DownloadServer.exe <port> <File path>**)

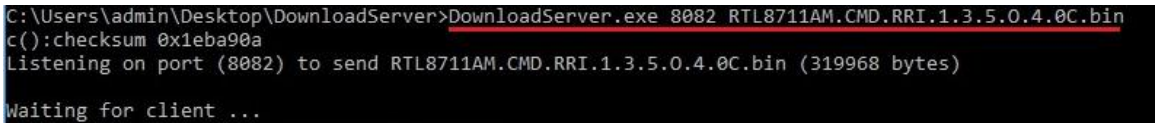


Figure 157 Open Download Server

Step 10: Open a "Docklight" application or other terminal application, and open UART0 (debug UART). Send "OTA" to UART0, RA6000 will send OTA command to RBW6100-RA. RBW6100-RA connect to "DownloadServer" and start to upgrade firmware.

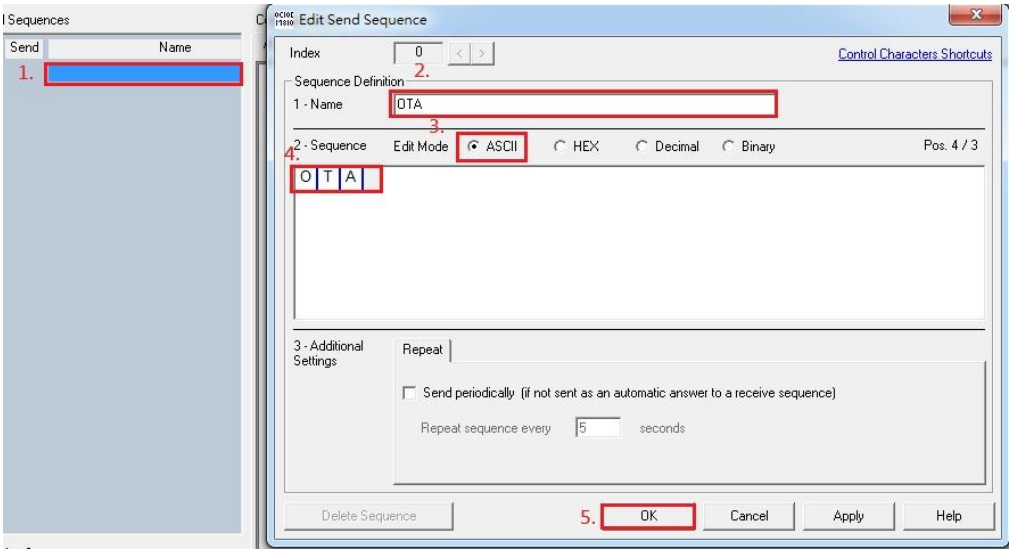


Figure 158 Docklight set command

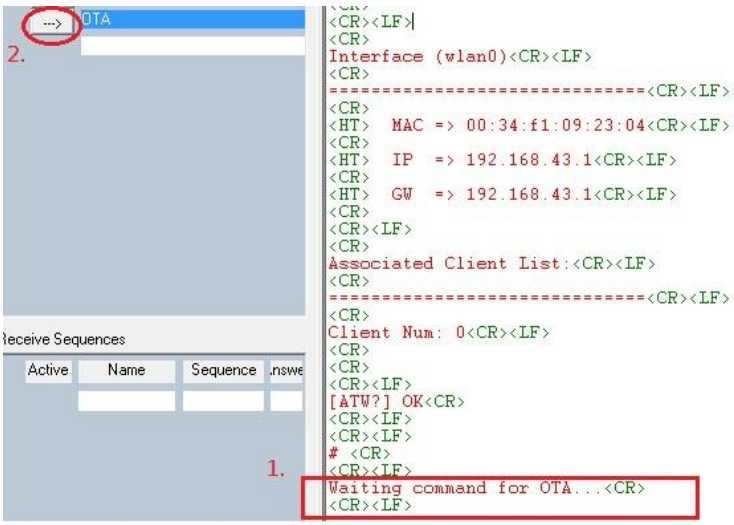


Figure 159 Docklight send command

```
Waiting command for OTA...<CR>
<CR><LF>
into StartOTA-894<CR>
<CR><LF>
<CR>
<CR><LF>
[ATSO] OK<CR>
<CR><LF>
<CR><LF>
# <CR>
<CR><LF>
Firmware version:v1.3.5<CR>
<CR><LF>
<CR><LF>
<CR>
Mac address: 00:34:f1:09:23:04<CR>
<CR><LF>
<CR><LF>
# <CR>
<CR><LF>
OTA Finish!!<CR>
<CR><LF>
```

Figure 160 Start OTA

Step 10: Download server will send firmware bin file to 8711 and upgrade firmware.

```
C:\Users\admin\Desktop\DownloadServer>DownloadServer.exe 8082 RTL8711AM.CMD.RRI.1.3.5.O.4.0C.bin
c():checksum 0x1eba90a
Listening on port (8082) to send RTL8711AM.CMD.RRI.1.3.5.O.4.0C.bin (319968 bytes)

Waiting for client ...
Accept client connection from 192.168.43.1
Send checksum and file size first
Send checksum byte 12
Sending file...
.....
.....
.....
Total send 319968 bytes
Client Disconnected.
Waiting for client ...
```

Figure 161 Download server message

Updating RB8762 firmware:

This chapter will show you how to use OTA to upgrade firmware.

Step 1. Upload the latest firmware file into your smart phone.

Step 2. Launch OTA APP and click "Function Test". Click "Device" to search for "Radicom " device. Then select the device to enter next step.

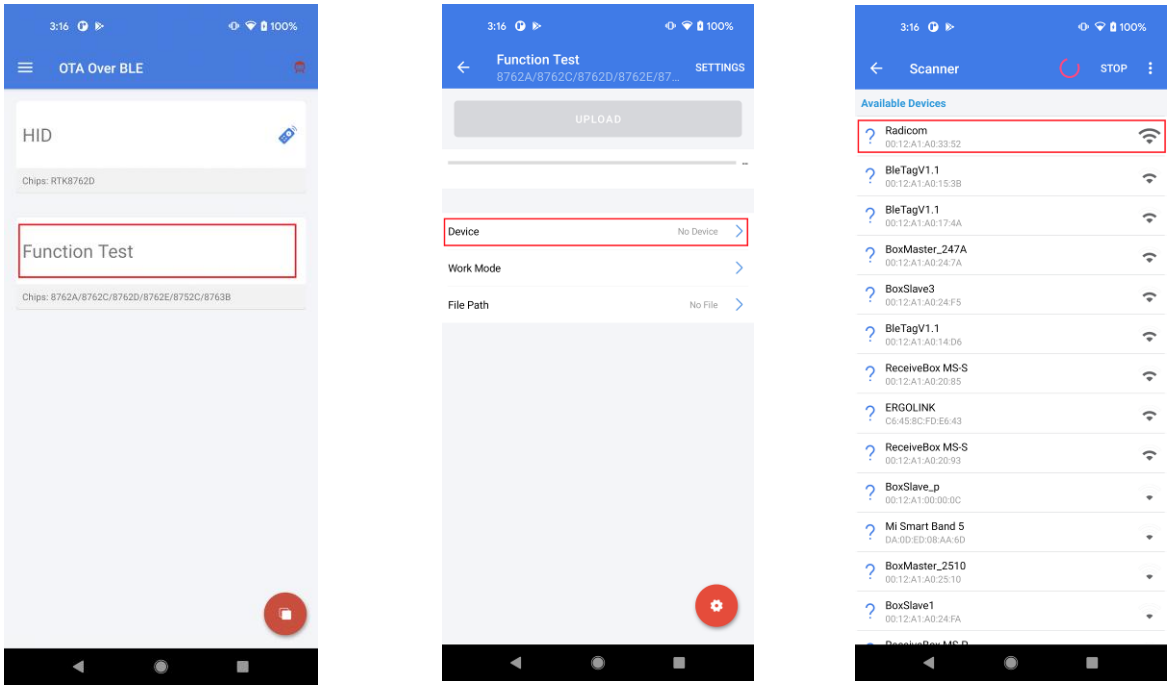


Figure 162 OTA Procedure-1

Step 3. Click "**File Path**" to select APP image file.. Then, click "**UPLOAD**".

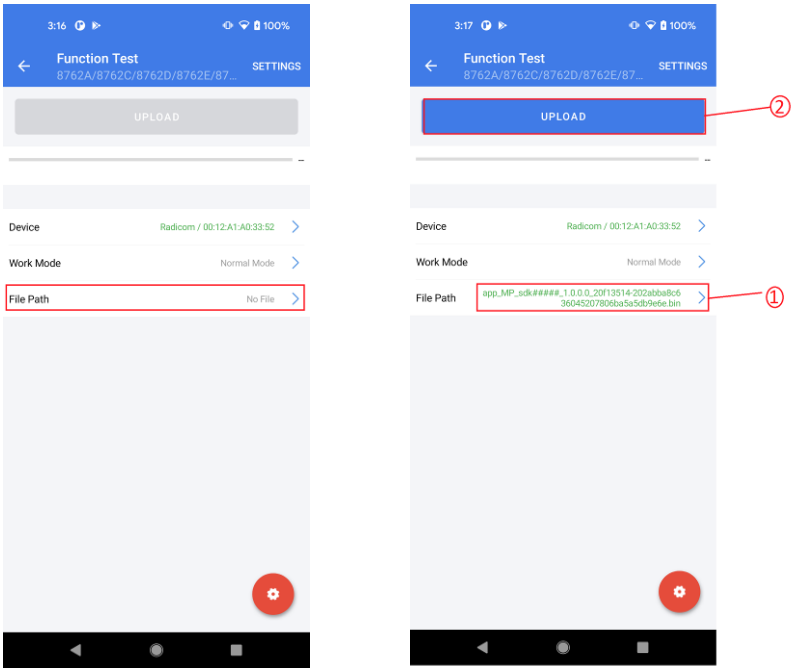


Figure 163 OTA Procedure-2

Step 4. The firmware upgrade will start immediately. Once the screen shows “image active success” means firmware upgrade successfully.

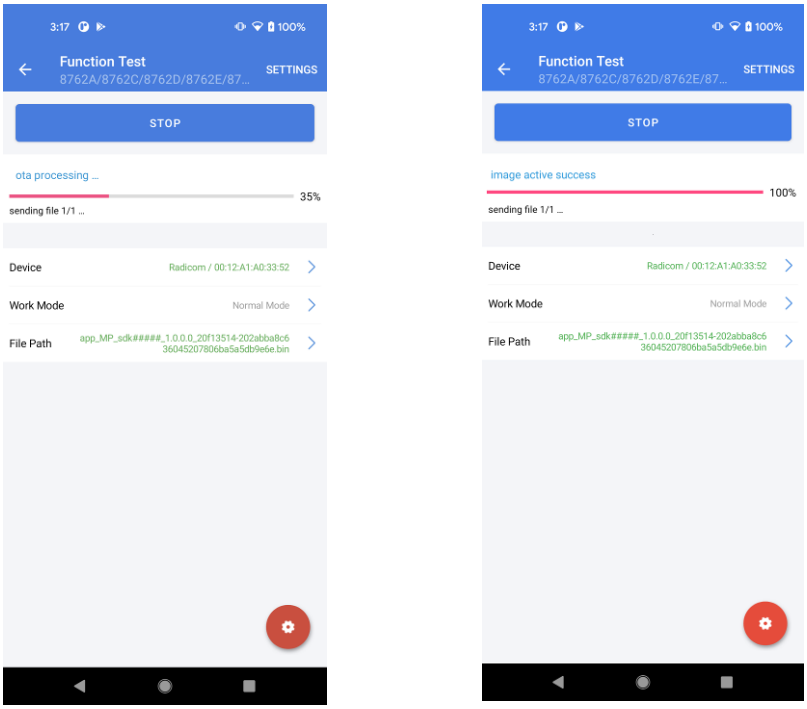


Figure 164 OTA Procedure-3

Limited Warranty

Warranty Coverage and Duration

Radicom Research, Inc. ("RRI") warrants to the original purchaser its RRI-manufactured products ("Product") against defects in material and workmanship under normal use and service for a period of one year from the date of delivery. During the applicable warranty period, at no charge, RRI will, at its option, either repair, replace or refund the purchase price of this Product, provided it is returned in accordance with the terms of this warranty to RRI. Repair, at the option of RRI, may include the replacement of parts, boards or other components with functionally equivalent reconditioned or new parts, boards or other components. Replaced parts, boards or other components are warranted for the balance of the original applicable warranty period. All replaced items shall become the property of RRI.

RRI MAKES NO GUARANTEE OR WARRANTY THAT THE PRODUCT WILL PREVENT OCCURRENCES, OR THE CONSEQUENCES THEREOF, WHICH THE PRODUCT IS DESIGNED TO DETECT.

This expressed limited warranty is extended by RRI to the original end-user purchaser only, and is not assignable or transferable to any other party. This is the complete warranty for the Product manufactured by RRI, and RRI assumes no obligation or liability for additions or modifications to this warranty. In no case does RRI warrant the installation, maintenance or service of the Product.

RRI is not responsible in any way for any ancillary equipment not furnished by RRI that is attached to or used in connection with the Product, or for operation of the Product with any ancillary equipment, and all such equipment is expressly excluded from this warranty. Because of wide variations in topographical and atmospheric conditions, which may require availability of repeater stations or of particular radio frequencies, RRI assumes no liability for range, coverage or suitability of the Product for any particular application. Buyer acknowledges that RRI does not know a particular purpose for which buyer wants the Product, and that buyer is not relying on RRI's skill and judgment to select or furnish suitable goods.

What this Warranty does NOT Cover:

(a) Defects or damage resulting from use of the Product in other than its normal and customary manner.

(b) Defects or damage from misuse, accident or neglect.

(c) Defects of damage from improper testing, operation, maintenance, installation, alteration, modification or adjustment.

(d) Disassembly or repair of the Product in such a manner as to adversely affect performance or prevent adequate inspection and testing to verify any warranty claim.

(e) Any Product that has had its serial number or date code removed or made illegible.

How to Receive Warranty Service:

To obtain warranty service, contact RRI by phone +1 (408) 383-9006 for your sales representative or email to sales@radi.com for an RMA (Return Merchandise Authorization) number. Deliver or send the Product, transportation and insurance prepaid to RRI, with the RMA number clearly marked on the outside of the package.

General Provision

This warranty sets forth the full extent of RRI's responsibilities regarding the Product. Repair, replacement or refund of the purchase price, at RRI's option, is the exclusive remedy.

THIS WARRANTY IS GIVEN IN LIEU OF ALL OTHER EXPRESSED WARRANTIES. ANY APPLICABLE IMPLIED WARRANTIES, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTY OF MERCHANTABILITY, ARE LIMITED TO THE DURATION OF THIS LIMITED WARRANTY. TO THE FULLEST EXTENT PERMITTED BY LAW, RRI DISCLAIMS ANY LIABILITY FOR DAMAGES IN EXCESS OF THE PURCHASE PRICE OF THE PRODUCT, FOR ANY LOSS OF USE, LOSS OF TIME, INCONVENIENCE, COMMERCIAL LOSS, LOST PROFITS OR SAVING OR OTHER INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE OR FAILURE OF SUCH PRODUCT.

Contacting Radicom Research

If more information or technical support is needed, please contact us:

671 E. Brokaw Road

San Jose, CA. 95112

Telephone: (408) 383 9006

Fax: (408) 383 9007

or

e-mail: sales@radi.com

http://www.radi.com/