User Manual

DA16200 DA16600 FreeRTOS Getting Started Guide

UM-WI-056

Abstract

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

Contents

Figures

Tables

1 Terms and Definitions

AP	Access Point
BSS	Basic Service Set
BUFP	Buffering Probe
COM	Communication Port
DDPS	DPM Dynamic Period Setting
DPM	Dynamic Power Management
EVK	Evaluation Kit
IDE	Integrated Development Environment
JDK	Java Development Kit
OTA	Over the Air
RTOS	Real Time Operating System
SDK	Software Development Kit
SFDP	Serial Flash Discoverable Parameter
SSID	Service Set Identifier
TIM	Traffic Indication Map
UART	Universal Asynchronous Receiver/Transmitter
UC	Unicast Packet

2 References

[1] DA16200, Datasheet, Renesas Electronics

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

[3] UM-B-114, DA14531, Devkit Pro Hardware, User Manual, Renesas Electronics

[4] UM-WI-012, DA16200, SPI SFlash Downloader, User Manual, Renesas Electronics

[5] UM-WI-003, DA16200 DA16600, Host Interface and AT-Command, User Manual, Renesas Electronics

[6] UM-WI-038, DA16200 DA16600, Getting Started with AWS-IoT Using AT-Command, User Manual, Renesas Electronics

[7] UM-WI-039, DA16200 DA16600, Multi Downloader, Renesas Electronics

[8] UM-WI-042, DA16200 DA16600, Provisioning Mobile App, User Manual, Renesas Electronics

[9] DA16200 DEVKT Electric Schematic, Renesas Electronics

[10] DA16600 DEVKT Electric Schematic, Renesas Electronics

3 Overview

The DA16200/DA16600 is a highly integrated ultra-low power Wi-Fi system on chip (SoC) that allows users to develop Wi-Fi solutions using a single chip.

Evaluation Kits and SDKs are provided to simplify the process of starting a project with the DA16200 or the DA16600.

This document provides step by step guides to start using the Evaluation Kit (EVK) and SDK by providing details on:

- The DA16200 and DA16600 EVK hardware (see Section 4)
- How to connect to the Evaluation board (EVB) (see Section 4.4)
- How to program firmware images to the DA16200/DA16600 (see Section 4.5)
- How to provision and test the Wi-Fi connection (see Section 4.6)
- How to install the development tools, build an application, program firmware images in e^{2}studio (see Section 5)
- How to debug an application (see Section 5.6)
- How to perform various test procedures to demonstrate the capabilities of the DA16200/DA16600 (see Section 6)

Once completing these steps, the EVK and development environment are ready for developing a complete Wi-Fi solution.

If the Wi-Fi solution being developed is based on a host MCU that uses the DA16200/DA16600 only as a Wi-Fi communication interface through AT Commands, then a prebuilt firmware image can be used to develop that solution. For details on how to use AT Commands see the following documents:

- UM-WI-003 DA16200 DA16600 Host Interface and AT Command User Manual Ref. [5]
- UM-WI-038 DA16200 DA16600 Getting Started with AT Command for AWS-IoT Ref. [6]

If the Wi-Fi solution being developed requires special functions or it is a standalone solution using only the DA16200/DA16600, then the SDK can be used to develop a new firmware image for that solution. For more details on how to use the SDK to develop an application, see the following documents:

- UM-WI-046 DA16200 DA16600 FreeRTOS SDK Programmer Guide Ref. [2]

4 Evaluation Kits

4.1 Introduction

The DA16200 and DA16600 EVKs are designed to allow a customer to test and develop Wi-Fi applications very quickly.

The EVK contains:

- Evaluation Board: The specific board with either DA16200MOD or DA16600MOD installed
- USB cable

4.2 DA16200 EVB Ver 11.0

Figure 1 shows the hardware configuration of the DA16200 EVB.

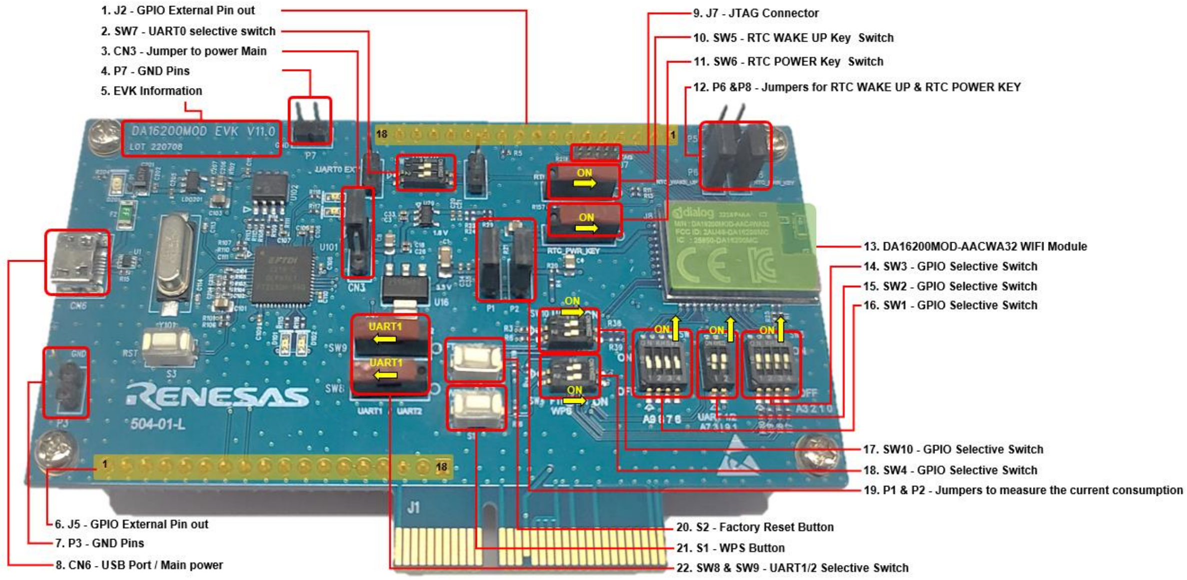
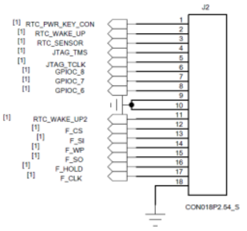

1. J2 - GPIO External Pin out
2. SW7 - UART0 selective switch
3. CN3 - Jumper to power Main
4. P7 - GND Pins
5. EVK Information
9. J7 - JTAG Connector
10. SW5 - RTC WAKE UP Key Switch
11. SW6 - RTC POWER Key Switch
12. P6 &P8 - Jumpers for RTC WAKE UP & RTC POWER KEY
13. DA16200MOD-AACWA32 WIFI Module
14. SW3 - GPIO Selective Switch
15. SW2 - GPIO Selective Switch
16. SW1 - GPIO Selective Switch
17. SW10 - GPIO Selective Switch
18. SW4 - GPIO Selective Switch
19. P1 & P2 - Jumpers to measure the current consumption
20. S2 - Factory Reset Button
21. S1 - WPS Button
22. SW8 & SW9 - UART1/2 Selective Switch
6. J5 - GPIO External Pin out
7. P3 - GND Pins
8. CN6 - USB Port / Main power
DA16200MOD EVK V11.0
ON
UART1
RENESAS
504-01-L
18
1
J1

Figure 1: DA16200 EVB Hardware Configuration

DA16200 EVB Ver 11.0 has the following components:

Table 1: Components on DA16200 EVB

No	Name	Description
1	**J2 – GPIO External pin-out**	 J2 [1] RTC_PWR_KEY_CON [1] RTC_WAKE_UP [1] RTC_SENSOR [1] JTAG_TMS [1] JTAG_TCLK [1] GPIOC_8 [1] GPIOC_7 [1] GPIOC_6 [1] RTC_WAKE_UP2 [1] F_CS [1] F_SI [1] F_WP [1] F_SO [1] F_HOLD [1] F_CLK 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 CON018P2.54_S
2	**SW7– UART0 selection switch**	UART0 selection switch, default ON, see Table 2.
3	**CN3 – Jumper to POWER**	5V Power Selection from USB or External
4	**P7 – GND Pins**	GND for Test
5	**Board Information**	Board version and manufacturing date

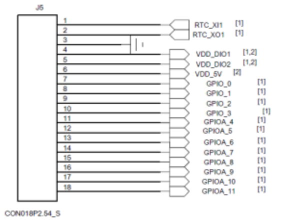
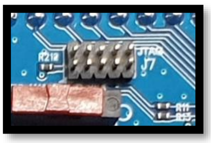
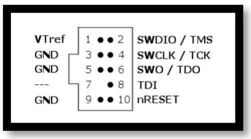
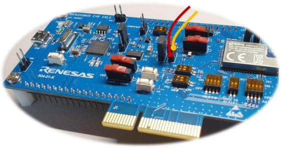
No	Name	Description
6	**J5 – GPIO External pin-out**	 J5 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 RTC_XI1 [1] RTC_XO1 [1] VDD_DIO1 [1,2] VDD_DIO2 [1,2] VDD_5V [2] GPIO_0 [1] GPIO_1 [1] GPIO_2 [1] GPIO_3 [1] GPIOA_4 [1] GPIOA_5 [1] GPIOA_6 [1] GPIOA_7 [1] GPIOA_8 [1] GPIOA_9 [1] GPIOA_10 [1] GPIOA_11 [1] CON018P2.54_S
7	**P3 – GND Pins**	GND for Test
8	**CN6 – USB Port**	Provide UART0 for debug, UART1/2 for test and SPI.
9	**J7 – JTAG connector**	Connector for the IAR I-jet JTAG Debugger   VTref GND GND --- GND 1 ● ● 2 3 ● ● 4 5 ● ● 6 7 ● 8 9 ● ● 10 SWDIO / TMS SWCLK / TCK SWO / TDO TDI nRESET Note: Pin 7 of the I-Jet debugger cable is keyed with a white plug so pin 7 must be removed from the EVB.
10	**SW5 – RTC_WAKE_UP switch**	Switch to wake up the board from sleep mode.
11	**SW6 – RTC_PWR_KEY switch**	Switch to turn the EVB ON and OFF.
12	**P6 – Jumpers for RTC_WAKE_UP1**	Pin for connecting the external control signal with MCU. For normal operation, this jumper should be shorted.
13	**DA16200MOD-AACWA32**	Renesas **Wi-Fi Module.**
14	**SW3 – GPIO selection switch**	GPIO selection switch, default OFF, see Table 2.
15	**SW2 – GPIO selection switch**	GPIO selection switch, default ON, see Table 2.
16	**SW1 – GPIO selection switch**	GPIO selection switch, default OFF, see Table 2.
17	**SW10 – GPIO selection switch**	GPIO selection switch, default ON, see Table 2.
18	**SW4 – GPIO selection switch**	GPIO selection switch, default ON, see Table 2.
19	**P1 – External VDD**	External VDD (3.3 V) can be supplied to the P1 pins. For normal operation, this jumper should be shorted.
	P2 – Jumper to measure current consumption of the DA16200	Pin for current measurement. For normal operation, this pin should be shorted. - Pull out the Short Pin cap and use jumper wires to connect to the measuring equipment. 
20	**S2 – Factory reset button**	Factory reset button using GPIOA_7. To enable this button, set Pin 2 of SW4 to ON and press the button for more than 5 seconds to initialize `nvram` data.
21	**S1 – WPS button**	WPS button using GPIOA_6. To enable this button, set Pin 1 of SW4 to ON and press the button to start WPS mode.
22	**SW8, SW9 – UART1/2 selection switch**	UART1/2 selection switch, default LEFT (UART1), see Table 2.

Table 2: DA16200 EVB Switch Description

Switch	Pin	Relevant GPIO	ON	OFF	Default
SW1	1	GPIOA6	Connects FT2232H to SPI_CSB (Note 1)	WPS	OFF
	2	GPIOA7	Connects FT2232H to SPI_CLK (Note 1)	Factory Reset	
	3	GPIOA8	Connects FT2232H to SPI_MISO (Note 1)	External pin-out (J2/J5) only	
	4	GPIOA9	Connects FT2232H to SPI_MOSI (Note 1)		
SW2	1	GPIOA4	Connects FT2232H to UART1_TXD	External pin-out (J2/J5) only	ON
	2	GPIOA5	Connects FT2232H to UART1_RXD		
SW3	1	GPIOA0	Connects FT2232H to SPI_CSB (Note 1)	External pin-out (J2/J5) only	OFF
	2	GPIOA1	Connects FT2232H to SPI_CLK (Note 1)		
	3	GPIOA2	Connects FT2232H to SPI_MISO (Note 1)		
	4	GPIOA3	Connects FT2232H to SPI_MOSI (Note 1)		
SW4	1	GPIOA6	WPS	External pin-out (J2/J5) only	ON
	2	GPIOA7	Factory Reset		
SW7	1	UART0_TXD	Connects FT232H to UART0_TXD	NC	ON
	2	UART0_RXD	Connects FT232H to UART0_RXD		
SW8	-	GPIOA4	Not available on DA16200 EVB V11.0 (Right)	Connects SW2 to SW10 (Left)	LEFT
SW9	-	GPIOA5	Not available on DA16200 EVB V11.0 (Right)	Connects SW2 to SW10 (Lett)	LEFT
SW10	1	GPIOA4	Connects SW8 to UART1_TXD	External pin-out (J2/J5) only	ON
	2	GPIOA5	Connects SW9 to UART1_RXD		

Note 1 By default, the SPI interface is configured to support firmware download through the FT2232H. See section 4.7 for details on how to configure the EVB to support AT Command processing from an MCU directly connected to the SPI interface.

The current consumption can be measured at P2 jumper with current measuring equipment. See Section 6.5 for details on the test setup. When DA16200 EVK Pro is used for measuring current consumption, the P1 jumper must be removed. For more information on DA16200 EVK Pro, see 6.5.2.

For more details on the DA16200 EVB, see the schematic Ref. [9].

4.3 DA16600 EVB Ver 6.0

Figure 2 shows the hardware configuration of the DA16600 EVB.

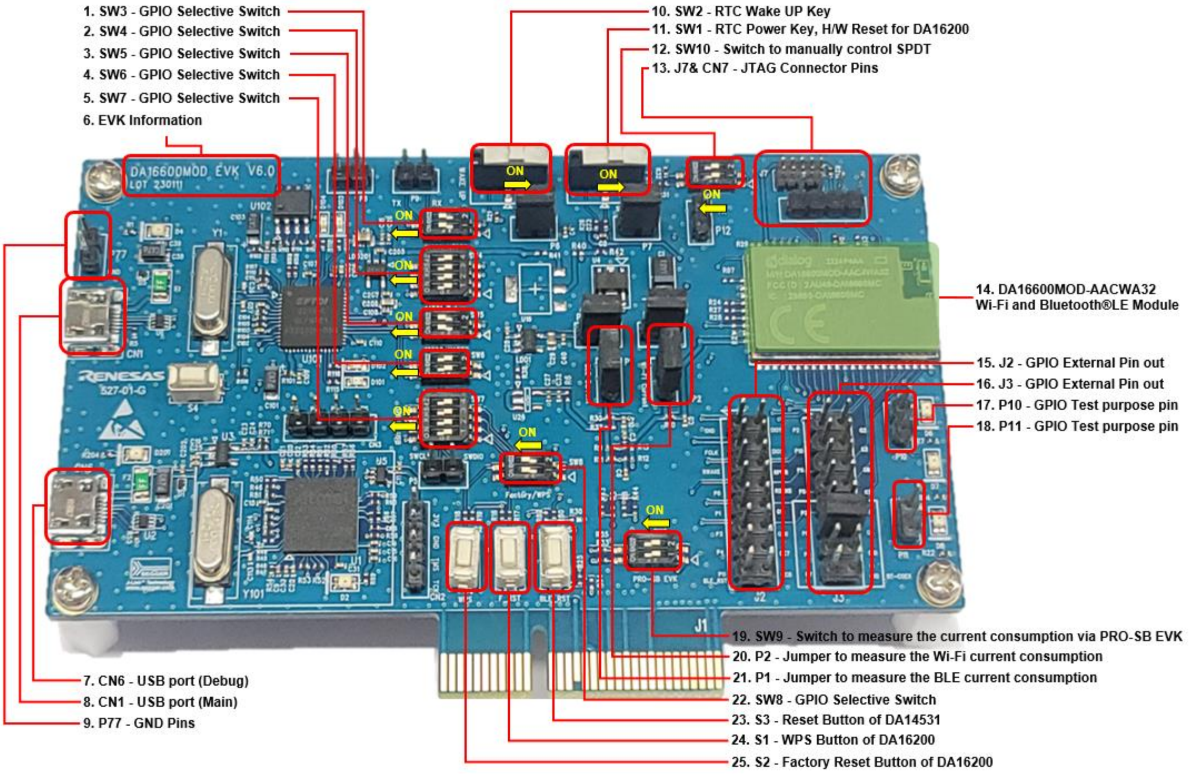
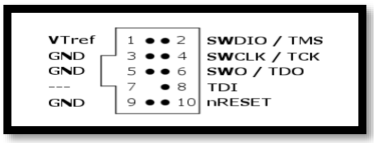
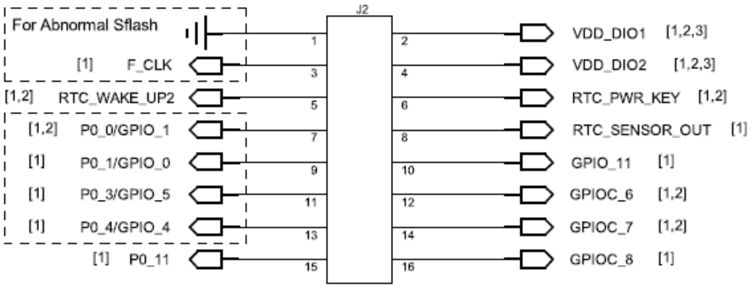
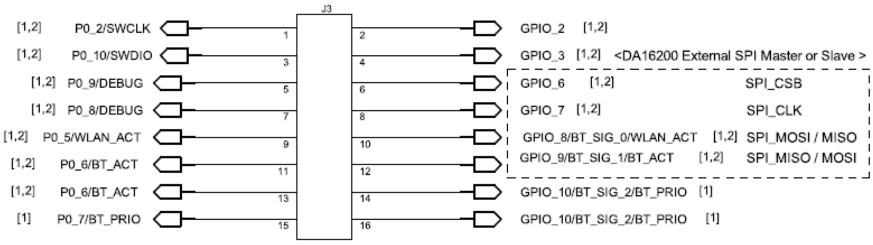

1. SW3 - GPIO Selective Switch
2. SW4 - GPIO Selective Switch
3. SW5 - GPIO Selective Switch
4. SW6 - GPIO Selective Switch
5. SW7 - GPIO Selective Switch
6. EVK Information
7. CN6 - USB port (Debug)
8. CN1 - USB port (Main)
9. P77 - GND Pins
10. SW2 - RTC Wake UP Key
11. SW1 - RTC Power Key, H/W Reset for DA16200
12. SW10 - Switch to manually control SPDT
13. J7& CN7 - JTAG Connector Pins
14. DA16600MOD-AACWA32 Wi-Fi and Bluetooth®LE Module
15. J2 - GPIO External Pin out
16. J3 - GPIO External Pin out
17. P10 - GPIO Test purpose pin
18. P11 - GPIO Test purpose pin
19. SW9 - Switch to measure the current consumption via PRO-SB EVK
20. P2 - Jumper to measure the Wi-Fi current consumption
21. P1 - Jumper to measure the BLE current consumption
22. SW8 - GPIO Selective Switch
23. S3 - Reset Button of DA14531
24. S1 - WPS Button of DA16200
25. S2 - Factory Reset Button of DA16200
ON

Figure 2: DA16600 EVB Hardware Configuration

DA16600 EVB has the following components:

Table 3: Components on DA16600 EVB

No	Name	Description
1	**SW3 – GPIO selective Switch**	GPIO selection switch, default ON, see Table 4.
2	**SW4 – GPIO selective Switch**	GPIO selection switch, default OFF, see Table 4.
3	**SW5 – GPIO selective Switch**	GPIO selection switch, default ON, see Table 4.
4	**SW6 – GPIO selective Switch**	GPIO selection switch, default OFF, see Table 4.
5	**SW7 – GPIO selective Switch**	GPIO selection switch, default OFF, see Table 4.
6	**Board information**	Board version and manufacturing date
7	**CN6 – USB Port (BLE)**	Connect directly to DA14531 for debug only. Note: Do not use this port during normal operation.
8	**CN1 – USB Port (Wi-Fi)**	Provides UART0 for debug and UART1 for test.
9	**P77 – GND Pins**	GND for test
10	**SW2 – RTC Wake up2 key**	Switch to wake up the board from sleep mode.
11	**SW1 – RTC Power key**	Switch to turn the EVB on and off.
12	**SW10 – Switch to control the SPDT**	Switch to control RF switch in DA16600MOD at test mode, default OFF.
13	**J7 & CN7 – JTAG Connector**	Connector for IARs I-jet JTAG Debugger.

No	Name	Description
		 VTref GND GND --- GND 1 2 3 4 5 6 7 8 9 10 SWDIO / TMS SWCLK / TCK SWO / TDO TDI nRESET Note: Pin 7 on the I-Jet debugger cable is keyed with a white plug so pin 7 must be removed from the EVB.
14	**DA16600MOD-AACWA32**	Renesas **Wi-Fi and Bluetooth® LE Module.**
15	**J2 – External pin-out**	 J2 For Abnormal Sflash [1] F_CLK [1,2] RTC_WAKE_UP2 [1,2] P0_0/GPIO_1 [1] P0_1/GPIO_0 [1] P0_3/GPIO_5 [1] P0_4/GPIO_4 [1] P0_11 1 3 5 7 9 11 13 15 2 4 6 8 10 12 14 16 VDD_DIO1 [1,2,3] VDD_DIO2 [1,2,3] RTC_PWR_KEY [1,2] RTC_SENSOR_OUT [1] GPIO_11 [1] GPIOC_6 [1,2] GPIOC_7 [1,2] GPIOC_8 [1]
16	**J3 – External pin-out**	 J3 [1,2] P0_2/SWCLK [1,2] P0_10/SWDIO [1,2] P0_9/DEBUG [1,2] P0_8/DEBUG [1,2] P0_5/WLAN_ACT [1,2] P0_6/BT_ACT [1,2] P0_6/BT_ACT [1] P0_7/BT_PRIO 1 3 5 7 9 11 13 15 2 4 6 8 10 12 14 16 GPIO_2 [1,2] GPIO_3 [1,2] <DA16200 External SPI Master or Slave > GPIO_6 [1,2] SPI_CSB GPIO_7 [1,2] SPI_CLK GPIO_8/BT_SIG_0/WLAN_ACT [1,2] SPI_MOSI / MISO GPIO_9/BT_SIG_1/BT_ACT [1,2] SPI_MISO / MOSI GPIO_10/BT_SIG_2/BT_PRIO [1] GPIO_10/BT_SIG_2/BT_PRIO [1] P0_6 (Pin #11) should be connected to GPIOA9 (Pin #12) or GPIOA10 (Pin #14) externally for BLE-Wi-Fi COEX.
17	**P10 – Jumper to test GPIO**	GPIO test pin. Add jumpers from J2/J3 to P10 to control the two LEDs using GPIOs.
18	**P11 – Jumper to test GPIO**	GPIO test pin. Add jumpers from J2/J3 to P11 to control the two LEDs using GPIOs.
19	**SW9 – Switch to test current consumption**	Switch to test current consumption using a pro board kit, default OFF.
20	**P2 – Jumper to measure current consumption of DA16200**	Jumper to measure current used by the Wi-Fi device. For normal operation, this jumper must be shorted.
21	**P1 – Jumper to measure current consumption of DA14531**	Jumper to measure current used by the Bluetooth® LE device. For normal operation, this jumper must be shorted.
22	**SW8 – GPIO selective Switch**	GPIO selection switch, default ON, see Table 4.
23	**S3 – DA14351 Reset Button**	Reset button of DA14531 in test mode.
24	**S1 – WPS Button**	WPS button using GPIOA6. To enable this button, set Pin 1 of SW7 to ON and press the button to start WPS mode.
25	**S2 – Factory Reset Button**	Factory reset button using GPIOA7. To enable this button, set Pin 2 of SW7 to ON and press the button for more than 5 seconds to initialize `nvram` data.

DA16600 EVB includes several switches, see description for each switch below.

Table 4: DA16600 EVB Switch Description

Switch	Pin	Relevant pin		ON	OFF	Default
SW3	1	DA16200	UART0_RXD	Connects FT232H to UART0_RXD	NC	ON
	2		UART0_TXD	Connects FT232H to UART0_TXD		
SW4	1	DA14531	P0_8	Connects FT2232H to P0_8 (UART_RXD)	External pin-out (J2/J3) only	OFF
	2		P0_9	Connects FT2232H to P0_9 (UART_TXD)		
	3		P0_2	Enables Keil toolchain debug of DA14531 (SWCLK)		
	4		P0_10	Enables Keil toolchain debug of DA14531 (SWDIO)		
SW5	1	DA16200	GPIOC6	Connects FT2232H to UART2_TXD	External pin-out (J2/J3) only	ON
	2		GPIOC7	Connects FT2232H to UART2_RXD		
SW6	1	DA14531	P0_5	Connect FT2232H to P0_5 (1-wire UART)	External pin-out (J2/J3) only	OFF
	2			Connect FT2232H to P0_5 (1-wire UART)		
SW7	1	DA16200	GPIOA6	Not available on DA16600 EVB V5.0	External pin-out (J2/J3) only	OFF
	2		GPIOA7			
	3		GPIOA8			
	4		GPIOA9			
SW8	1	DA16200	GPIOA6	WPS	External pin-out (J2/J3) only	ON
	2		GPIOA7	Factory Reset		
SW9	1	DA16200	Measuring current consumption	Enables to measure current consumption of DA16200 with EVK Pro. Need to remove jumper P2.	Normal operation / Need to connect P2	OFF
	2	DA14531	Measuring current consumption	Enables to measure current consumption of DA14531 with EVK Pro. Need to remove jumper P1.	Normal operation / Need to connect P1	
SW10	1	DA14531	P0_6	Manual control of the internal RF SPDT. Pin1 OFF & Pin2 OFF: internally controlled. Pin1 ON & Pin2 OFF: DA14531 RF path ON Pin1 OFF & Pin2 ON: DA16200 RF path ON		OFF
	2					

The current consumption can be measured at jumpers P1 and P2 with current measuring equipment. See Section 6.5 for details on the test setup.

When DA16600 EVK Pro is used for measuring current consumption, remove jumpers P1 and P2 and set pins 1 and 2 of SW9 to the ON position. To measure the current consumption of DA16200, remove the jumper on P2 and turn ON pin1 of SW9. To measure the current consumption of DA14531, remove the jumper on P1 and turn ON pin2 of SW9. The current consumption of the DA16200 and the DA14531 can be measured simultaneously.

For more information on the DA16600 EVB, see the DA16600 DEVKT Electric Schematic Ref. [10].

4.4 Connecting to EVB

4.4.1 Configuring DA16200/DA16600 Serial Debug Interface

The DA16200/DA16600 provides a command/debug interface on UART0 for performing configuration and diagnostic functions. When the EVB is connected to the USB port (**CN6** on the DA16200 EVB or **CN1** on the DA16600 EVB), two virtual COM ports are created.

- On Windows, the two COM ports are displayed in the device manager (see Figure 3).

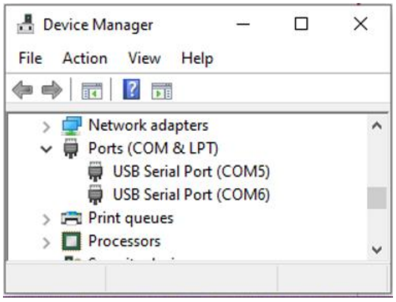


Figure 3: Windows - COM Ports

NOTE
On Windows, if the FTDI serial driver was not installed automatically, the driver can be downloaded from the following URL and install manually: http://www.ftdichip.com/Drivers/CDM/CDM21224_Setup.zip

- On Linux, the COM ports are created in the `/dev` directory as `ttyUSBx` devices.

```
$
$ ls -l /dev/ttyU*                              List the available ttyUSB serial
ports.
crw-rw---- 1 root dialout 188, 0 Aug 25 10:26 /dev/ttyUSB0
crw-rw---- 1 root dialout 188, 1 Aug 25 10:26 /dev/ttyUSB1
$
```

The lower numbered COM port (COM5 in Figure 3) is for the DA16200/DA16600 debug interface. The higher numbered COM port is for the DA16200/DA16600 ATCMD interface.

NOTE
On the DA16600 the higher numbered COM port can also be configured as the DA14531 serial debug interface. See Section 4.4.2.

A serial terminal application can be used for debugging DA16200/DA16600.

For Windows, the **Tera Term** terminal emulator program is recommended and can be downloaded from: https://ttssh2.osdn.jp. For Linux, the **minicom** terminal emulation program is recommended and can be installed using:

```
$
$ sudo apt install minicom                      Command to install minicom.
$
```

Once the terminal emulation application has been installed, Connect the USB cable to the EVB (**CN6** on DA16200 EVB or **CN1** on DA16600 EVB) and start the terminal emulation program.

In the terminal emulation program, go to the **Serial Port Setup** and select the lower number COM port and configure it as follows:

Table 5: Serial Port Configuration Values

Settings	Value
Baud Rate	230400
Data Bits	8
Parity	None
Stop Bits	1
Flow Control (HW/SW)	None

Turn ON the EVB (**SW6** on the DA16200 EVB or **SW2** on the DA16600 EVB) and check for output similar to the following:

```
Wakeup source is 0x4
[dpm_init_retmemory] DPM INIT CONFIGURATION(1)

        ******************************************************
        *              DA16200 SDK Information
        * ---------------------------------------------------
        *
        * - CPU Type         : Cortex-M4 (120MHz)
        * - OS Type          : FreeRTOS 10.4.3
        * - Serial Flash     : 4 MB
        * - SDK Version      : V3.1.3.0 GEN
        * - F/W Version      : FRTOS-GEN01-01-15129-000000
        * - F/W Build Time   : Aug 26 2021 22:58:01
        * - Boot Index       : 0
        *
        ******************************************************

System Mode : Station Only (0)
>>> DA16x Supp Ver2.7 - 2020_07
>>> MAC address (sta0) : d4:3d:39:10:a2:48
>>> sta0 interface add OK
>>> Start STA mode...
[/DA16200] #
```

Commands can be entered at the `[/DA16200]` prompt. A full list of the available debug interface commands can be found in Appendix B.

4.4.2 Configuring DA14531 Serial Debug Interface

This section is for DA16600 only and is required for special cases where access to the DA14531 BLE devices debug terminal is required.

The DA16600 EVB contains the DA14531 BLE device which is used for provisioning of the DA16600 Wi-Fi interface using a mobile application. On the DA16600 EVB, the DA14531 debug interface can be connected to the higher numbered COM port that is created when the USB cable is connected to **CN1**.

To enable the DA14531 debug interface, follow the settings below for **SW4, SW5, SW6** and **SW7**.

Table 6: Debug Switch Settings

Switch	Pin	Relevant pin		ON	Default
SW4	1	DA14531	P0_8	Connects FT2232H to P0_8 (UART_RXD)	**ON**
	2		P0_9	Connects FT2232H to P0_9 (UART_TXD)	

Switch	Pin	Relevant pin		ON	Default
	3		P0_2	Enables Keil toolchain debug of DA14531 (SWCLK)	Don't care
	4		P0_10	Enables Keil toolchain debug of DA14531 (SWDIO)	
SW5	1	DA16200	GPIOC6	Connects FT2232H to UART2_TXD	OFF
	2		GPIOC7	Connects FT2232H to UART2_RXD	
SW6	1	DA14531	P0_5	Connect FT2232H to P0_5 (1-wire UART)	OFF
	2			Connect FT2232H to P0_5 (1-wire UART)	
SW7	1	DA16200	GPIOA6	Not available on DA16600 EVB V5.0	OFF
	2		GPIOA7		
	3		GPIOA8		
	4		GPIOA9		

Once the switches are set properly, open a serial terminal application, and select the higher numbered COM port. Configure the higher numbered COM port as follows:

Table 7: Debug Console Settings

Settings	Value
Baud Rate	115200
Data Bits	8
Parity	None
Stop Bits	1
Flow Control (HW/SW)	None

Open a second serial terminal application and connect to the DA16200 debug console and run the `reboot` command. The following output will appear in the DA14531 debug console:

```
user_on_init
```

NOTE

The uart_program_da16200 in followed section can be used for debug console of DA16200 but it cannot be used with DA14531. Other console program like tera term or minicom cab be used for DA14531 debug console.

4.5 Programming Firmware Images

When using an EVB for the first time, the firmware must be updated to the latest version.

Prebuilt versions of the firmware for DA16200 and DA16600 can be downloaded from the Renesas Electronics' website. Go to the Renesas website (https://www.renesas.com/us/en/products/wireless-connectivity/wi-fi/low-power-wi-fi) and scroll down to the Software Downloads section. Find "DA16200 DA16600 FreeRTOS SDK Image" or type it in the search box, and then select the firmware and download. Alternately, the DA16200/DA16600 SDK can be used to rebuild the firmware images as described in Section 5.

There are two firmware images for the DA16200:

```
DA16200_FBOOT-GEN01-01-834dea5b8_W25Q32JW.img
DA16200_FRTOS-GEN01-01-866facc56-000000.img
```

There are three firmware images for the DA16600 since it also requires the DA14531 firmware image:

```
DA16600_FBOOT-GEN01-01-834dea5b8_AT25SL321.img
DA16600_FBOOT-GEN01-01-834dea5b8_W25Q32JW.img
```

```
DA16600_FRTOS-GEN01-01-866facc56-000000.img
DA14531_1/da14531_multi_part_proxr.img
```

NOTE

There are two versions of the DA16600 EVB: the Adesto AT25SL321 serial flash and the Winbond W25Q32JW serial flash.

Check the flash version by running `sflash info` command at the `[MROM]` prompt:

```
[/DA16600] # reset                               Switch to MROM command mode.
Reset BLE ...

[MROM] sflash info                               Display the flash information.
SFLASH:ef601615
Density:01ffffff
```

`ef601615`: Winbond W25Q32JWSNIQ

`1f421615`: Adesto AT25SL321

Choose the appropriate firmware image based on the flash version.

These firmware images are stored in the following locations in flash:

Table 8: Serial Flash Memory Map

Item	4 MB		Notes
	Address	Size	
2nd Bootloader	0x0000_0000	136 kB	BOOT firmware image
Boot Index	0x0002_2000	4 kB	Selects RTOS #0 or RTOS #1 as he active firmware
RTOS #0	0x0002_3000	1788 kB	RTOS firmware image
RTOS #1	0x001E_2000	1792 kB	Alternate RTOS firmware image (Used during OTA firmware update)
Certificate, NVRAM	0x003A_2000	44 kB	
User Area #01	0x003A_D000	332 kB	For DA16600, the DA14531 firmware image is stored here

NOTE

See Section 4.5.4 for details on how to select which firmware image is active (RTOS #0 or RTOS #1).

Each firmware image is downloaded individually through the debug serial port of the DA16200/DA16600 using the **Y-Modem** file transfer protocol.

4.5.1 Using uart_program_da16200 in Console Terminal

This tool is a console program which can be used in Windows, Linux, or python environment. The tool is in each folder of `<sdk_root_directory>/utility/j-link/scripts/qspi/win, linux or python)`. `uart_program_da16200.exe` in `win` folder is a prebuilt console program in Windows, `uart_program_da16200` in linux folder is a prebuilt console program in Linux, `uart_program_da16200.py` and `ymodem` folder in python folder is a python program based on python environment.

The steps below are how to program firmware images to DA16200/DA16600.

1. Copy the console program to the folder which has the images for DA16200/DA166000.
2. Connect the EVB to a laptop using a Micro USB cable.
3. Power ON the DA16200/DA16600 EVB.

4. Type `uart_program_da16200(.exe) -i <start address> <image name>` and click **Enter**.
   a. Example for FBOOT firmware image: uart_program_da16200.exe -i 0 DA16600_FBOOT-GEN01-01-c7f4c6cc22_AT25SL321.img
   b. Example for FRTOS firmware image: uart_program_da16200.exe -i 23000 DA16600_FRTOS-GEN01-01-0561372b7c-006529.img
   c. Example for DA14531 firmware image: uart_program_da16200.exe -i 3AD000 da14531_multi_part_proxr.img
5. Select **number** in the list of COM port and click **Enter** as follows.

```
d:\download>uart_program_da16200.exe -i 0 DA16200_FBOOT-GEN01-01-c7f4c6cc22_W25Q32JW.img
uart_program_da16200 Version 1.0.5
0. COM75 - USB Serial Port (COM75)
1. COM76 - USB Serial Port (COM76)
Please enter number in the list of your COM port and click enter.
--> 0
```

6. Programming is done automatically, and confirm the programming is completed successfully. Repeat steps 4~6 to program the FRTOS firmware image and the DA14531 firmware image (if using DA16600MOD) as follows.

```
d:\download>uart_program_da16200.exe -i 0 DA16200_FBOOT-GEN01-01-c7f4c6cc22_W25Q32JW.img
uart_program_da16200 Version 1.0.5
0. COM75 - USB Serial Port (COM75)
1. COM76 - USB Serial Port (COM76)
Please enter number in the list of your COM port and click enter.
--> 0
Selected COM port =  COM75 - USB Serial Port (COM75)
Entering download mode...
Preparing for download...
Read DA16200_FBOOT-GEN01-01-c7f4c6cc22_W25Q32JW.img
Ready for download.
Download file 1: DA16200_FBOOT-GEN01-01-c7f4c6cc22_W25Q32JW.img : 100.00% : 8.12s
Done successfully.

d:\download>uart_program_da16200.exe -i 23000 DA16200_FRTOS-GEN01-01-c4ca8087e8-006537.img
uart_program_da16200 Version 1.0.5
0. COM75 - USB Serial Port (COM75)
1. COM76 - USB Serial Port (COM76)
Please enter number in the list of your COM port and click enter.
--> 0
Selected COM port =  COM75 - USB Serial Port (COM75)
Entering download mode...
Preparing for download...
Ready for download.
Download file 1: DA16200_FRTOS-GEN01-01-c4ca8087e8-006537.img : 100.00% : 72.76s
Done successfully.
```

7. Confirm logs from DA16200/DA16600 for checking DA16200/DA16600 works properly as follows:
    a. Type `uart_program_da16200(.exe)`
    b. Select **number** in the list of COM port and click **Enter**
    c. Type `reboot`

```
d:\download>uart_program_da16200.exe
uart_program_da16200 Version 1.0.5
0. COM75 - USB Serial Port (COM75)
1. COM76 - USB Serial Port (COM76)
Please enter number in the list of your COM port and click enter.
--> 0
Selected COM port =  COM75 - USB Serial Port (COM75)
Entering console mode...
Ready for console mode. Input anything.


[/DA16200] #  reboot
reboot

Wakeup source is 0x0
[dpm_init_retmemory] DPM INIT CONFIGURATION(1)


        ******************************************************
        *             DA16200 SDK Information
        * ---------------------------------------------------
        *
        * - CPU Type        : Cortex-M4 (120MHz)
        * - OS Type         : FreeRTOS 10.4.3
        * - Serial Flash    : 4 MB
        * - SDK Version     : V3.2.8.0 GEN
        * - F/W Version     : FRTOS-GEN01-01-c4ca8087e8-006537
        * - F/W Build Time  : Jul 31 2023 14:07:09
        * - Boot Index      : 0
        *
        ******************************************************



System Mode : Station Only (0)
>>> Start DA16X Supplicant ...
>>> DA16x Supp Ver2.7 - 2022_03
>>> MAC address (sta0) : d4:3d:39:11:34:fc
>>> sta0 interface add OK
>>> Start STA mode...
>>> Hello World #1 ( Non network dependent application ) !!!
```

8. Reset to factory defaults:
   If it is the first time using the EVB or to return the EVB to a clean uninitialized state, run the factory reset command.

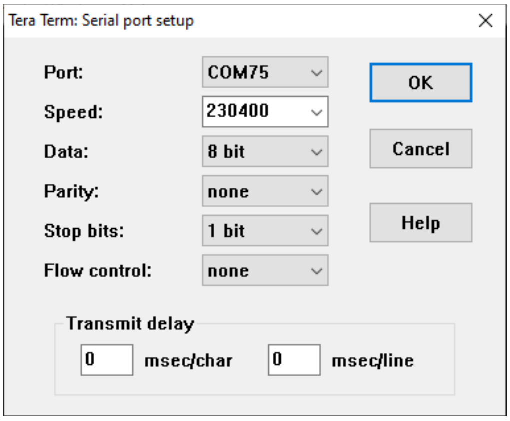
```
[/DA16200] #  factory    Use the "factory" command to do a factory reset
factory
FACTORY RESET [N/y/?]  y
y



←[31mStart Factory-Reset ...
←[0;39m
Rebooting....←[0m



Wakeup source is 0x0
[dpm_init_retmemory] DPM INIT CONFIGURATION(1)



        ******************************************************
        *             DA16200 SDK Information
        * ---------------------------------------------------
        *
        * - CPU Type        : Cortex-M4 (120MHz)
        * - OS Type         : FreeRTOS 10.4.3
        * - Serial Flash    : 4 MB
        * - SDK Version     : V3.2.8.0 GEN
        * - F/W Version     : FRTOS-GEN01-01-c4ca8087e8-006537
        * - F/W Build Time  : Jul 31 2023 14:07:09
        * - Boot Index      : 0
        *
        ******************************************************




System Mode : Station Only (0)
>>> Start DA16X Supplicant ...
>>> DA16x Supp Ver2.7 - 2022_03
>>> MAC address (sta0) : d4:3d:39:11:34:fc
>>> sta0 interface add OK
>>> Start STA mode...
>>> Hello World #1 ( Non network dependent application ) !!!
```

Once the firmware images are programmed successfully, the device can be configured by doing the Wi-Fi provisioning process. See Section 4.6 for details on how provision the Wi-Fi communication interface.

Please refer to `README.md` in python folder for more detail about how to use `uart_program_da16200(.exe/py)`.

4.5.2 Using Macro Script of Tera Term

This tool is for Windows only and the programming can be done automatically using a macro script of tera term. This macro files are available in the SDKROOT\apps\da16200\get_started\projects\da16200 or da16600\img\ folder in SDK package or each image folder of image package.
The serial port should be set up as shown in Figure 4. For DA16600, the lower port number of 2 available ports from DA16600 EVB must be selected for DA16200 debug interface.



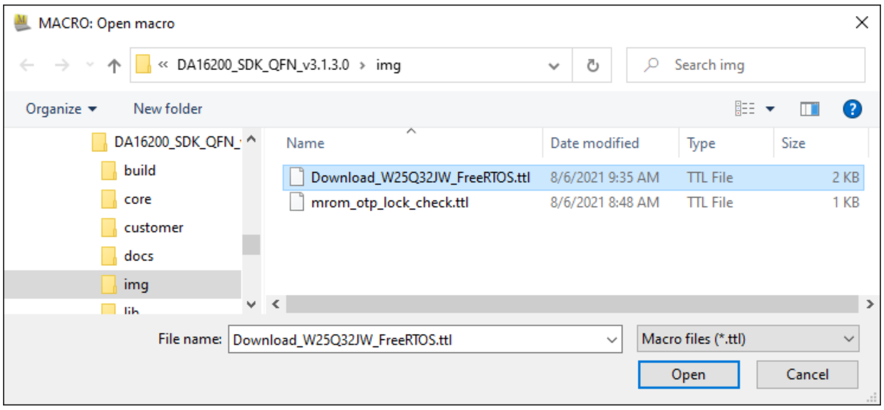
Port: COM75
Speed: 230400
Data: 8 bit
Parity: none
Stop bits: 1 bit
Flow control: none

Transmit delay
0 msec/char 0 msec/line

Figure 4: Serial Port Setup in Tera Term

The macro can be run as follows.

1. Once tera term is running and connected to the DA16200/DA16600, open the **Control** tab, and select the **Macro** menu item.
2. When the **MACRO: Open Macro** file selection window opens, navigate to the directory where the firmware images are stored and select the **.ttl** file as Figure 5.

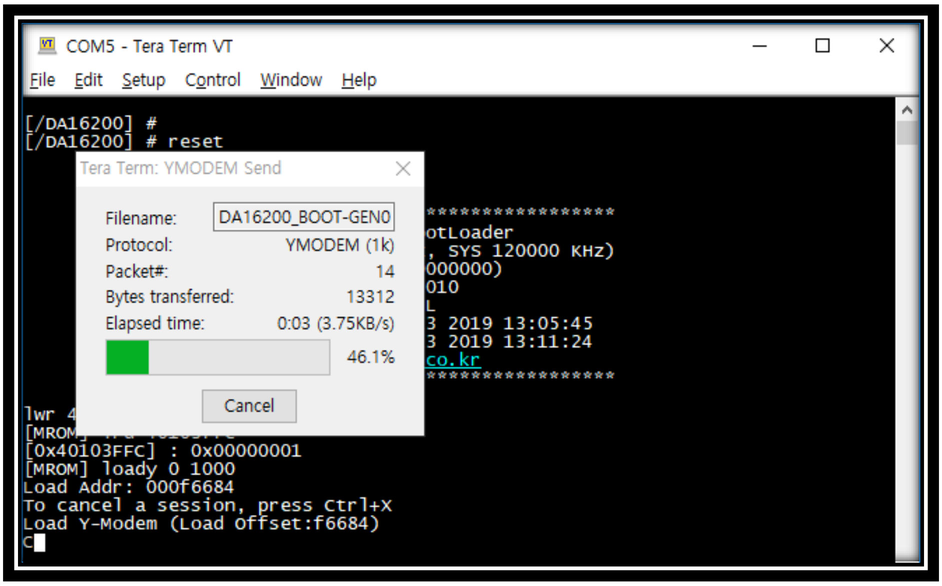


Name	Date modified	Type	Size
Download_W25Q32JW_FreeRTOS.ttl	8/6/2021 9:35 AM	TTL File	2 KB
mrom_otp_lock_check.ttl	8/6/2021 8:48 AM	TTL File	1 KB

File name: Download_W25Q32JW_FreeRTOS.ttl
Macro files (*.ttl)

Figure 5: Open Macro File

3. After opening the macro file, the FBOOT firmware image is programmed as shown in Figure 6.

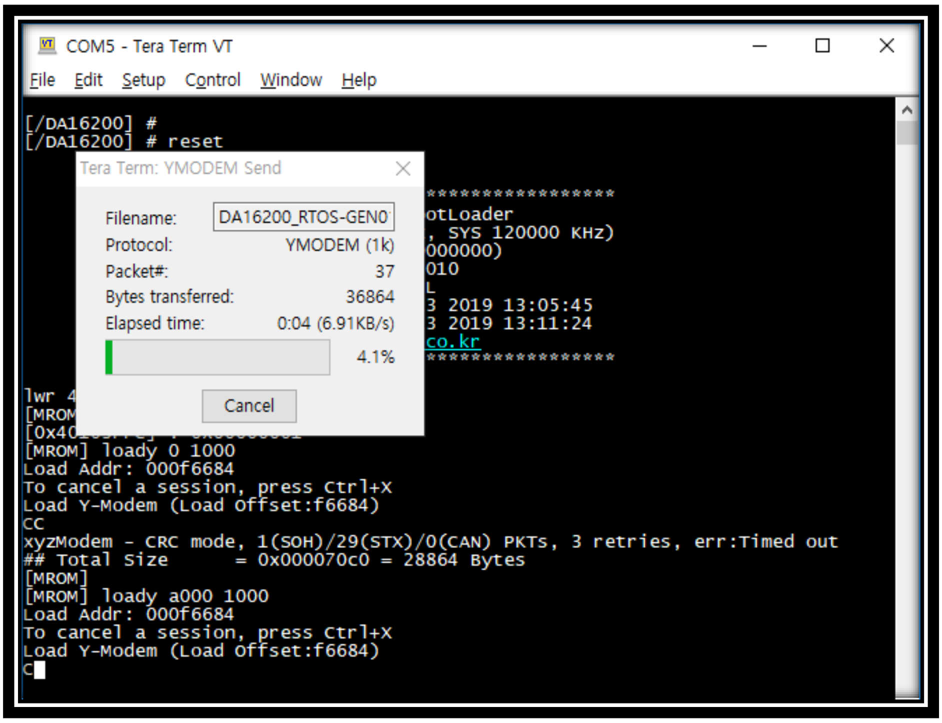


```
[/DA16200] #
[/DA16200] # reset
                                    *****************
                                    otLoader
                                    , SYS 120000 KHz)
                                    000000)
                                    010
                                    L
                                    3 2019 13:05:45
                                    3 2019 13:11:24
                                    co.kr
                                    *****************
lwr 4
[MROM]
[0x40103FFC] : 0x00000001
[MROM] loady 0 1000
Load Addr: 000f6684
To cancel a session, press Ctrl+X
Load Y-Modem (Load Offset:f6684)
C
```

Filename: DA16200_BOOT-GEN0
Protocol: YMODEM (1k)
Packet#: 14
Bytes transferred: 13312
Elapsed time: 0:03 (3.75KB/s)
46.1%

Figure 6: Programming FBOOT image

4. Program RTOS image as shown in Figure 7.



```
[/DA16200] #
[/DA16200] # reset
                                    *****************
                                    otLoader
                                    , SYS 120000 KHz)
                                    000000)
                                    010
                                    L
                                    3 2019 13:05:45
                                    3 2019 13:11:24
                                    co.kr
                                    *****************
lwr 4
[MROM
[0x40
[MROM] loady 0 1000
Load Addr: 000f6684
To cancel a session, press Ctrl+X
Load Y-Modem (Load Offset:f6684)
CC
xyzModem - CRC mode, 1(SOH)/29(STX)/0(CAN) PKTs, 3 retries, err:Timed out
## Total Size      = 0x000070c0 = 28864 Bytes
[MROM]
[MROM] loady a000 1000
Load Addr: 000f6684
To cancel a session, press Ctrl+X
Load Y-Modem (Load Offset:f6684)
C
```

Filename: DA16200_RTOS-GEN0
Protocol: YMODEM (1k)
Packet#: 37
Bytes transferred: 36864
Elapsed time: 0:04 (6.91KB/s)
4.1%

Figure 7: Programming FRTOS Image

5. DA16200 will reboot automatically after all images are programmed.

4.5.3 Using Multi-Download Tool

The Multi-Download Tool can be used to program the firmware images to multiple devices at the same time in Windows. See the user manual, Ref. [7] for details. uart_program_da16200(.exe) can be used for programing the firmware images to multiple devices by running the program on multiple terminals.

4.5.4 Changing Boot Index

Two versions of the main firmware can be stored in the flash at location RTOS #0 and RTOS #1. The **boot index** stored in serial flash selects which version of the firmware is used during the boot process. The command can be input through the UART download tool or other UART terminals.

The **boot index** can be changed by using the `boot_idx n` command:

```
boot_idx 0      // sets RTOS #0 as the firmware to boot
boot_idx 1      // sets RTOS #1 as the firmware to boot
```

After running the `boot_idx` command, run the `reboot` command to boot the firmware that was selected:

```
[/DA16200] # boot_idx 1  // or boot_idx 0
[/DA16200] # reboot

>>> P.TIM is relocated to RETMEM (0x20f835c0, 3)
[dpm_init_retmemory] DPM INIT CONFIGURATION(1)

Wakeup source is 0x0

        ******************************************************
        *             DA16200 SDK Information
        *
        * - CPU Type        : Cortex-M4 (120MHz)
        * - OS Type         : FreeRTOS 10.4.3
        * - Serial Flash    : 4 MB
        * - SDK Version     : V3.2.3.0 GEN
        * - F/W Version     : FRTOS-GEN01-01-6cc84b86f-003714
        * - F/W Build Time  : Apr 13 2022 10:20:02
        * - Boot Index      : 1
        *
        ******************************************************
```

After the reboot, verify the **Boot Index** and **F/W Version** matches the selected one.

4.6 Provisioning Wi-Fi

4.6.1 Setup for Station Mode

The Wi-Fi functions of the DA16200/DA16600 can easily be configured by running the **Easy Setup** Wi-Fi configuration wizard.

To configure the DA16200/DA16600 to operate in Station Mode, open the DA16200 or DA16600 debug console and run the `setup` command at the `[/DA16xxx]` prompt and then answer the questions to complete the setup as follows:

```
[/DA16200] setup                                              Start the Easy Setup Wizard.

Stop all services for the setting.
 Are you sure ? [Yes/No] : Y                                  Enter Y to stop the services.

[ DA16200 EASY SETUP ]

Country Code List:
AD  AE  AF  AI  AL  AM  AR  AS  AT  AU  AW  AZ  BA  BB  BD  BE  BF  BG  BH  BL
BM  BN  BO  BR  BS  BT  BY  BZ  CA  CF  CH  CI  CL  CN  CO  CR  CU  CX  CY  CZ
DE  DK  DM  DO  DZ  EC  EE  EG  ES  ET  EU  FI  FM  FR  GA  GB  GD  GE  GF  GH
GL  GP  GR  GT  GU  GY  HK  HN  HR  HT  HU  ID  IE  IL  IN  IR  IS  IT  JM  JO
JP  KE  KH  KN  KP  KR  KW  KY  KZ  LB  LC  LI  LK  LS  LT  LU  LV  MA  MC  MD
ME  MF  MH  MK  MN  MO  MP  MQ  MR  MT  MU  MV  MW  MX  MY  NG  NI  NL  NO  NP
NZ  OM  PA  PE  PF  PG  PH  PK  PL  PM  PR  PT  PW  PY  QA  RE  RO  RS  RU  RW
SA  SE  SG  SI  SK  SN  SR  SV  SY  TC  TD  TG  TH  TN  TR  TT  TW  TZ  UA  UG
UK  US  UY  UZ  VA  VC  VE  VI  VN  VU  WF  WS  YE  YT  ZA  ZW  ALL

 COUNTRY CODE ? [Quit] (Default KR) : US                      Enter the country code.

SYSMODE(WLAN MODE) ?
        1. Station
        2. Soft-AP
 MODE ? [1/2/Quit] (Default Station) : 1               Enter 1 for Station mode.

[ STATION CONFIGURATION ]
==============================================================================
[NO] [SSID]                                          [SIGNAL] [CH] [SECURITY]
------------------------------------------------------------------------------
[ 1] AndroidHotspot0568                                  -25    6        WPA2
[ 2] U+Net0208                                           -37    2        WPA2
[ 3] wonhome                                             -49   11        WPA2
[ 4] SK_WiFiGIGA1200                                     -53    3  WPA / WPA2
[ 5] [Hidden] BSSID-0a:5d:dd:de:12:03                    -54    3        WPA2
[ 6] DIRECT-GWM2020 Series                               -56    3        WPA2
[ 7] AT_303_WBEH2GT_a879                                 -66   11        WPA2
[ 8] olleh_WiFi_B602                                     -67   10  WPA / WPA2
[ 9] KT_GiGA_2G_Wave2_04A0                               -70    6  WPA / WPA2
[10] KT_GiGA_2G_Wave2_99EB                               -79    6  WPA / WPA2
[11] anyppi                                              -81    8        WPA2
------------------------------------------------------------------------------
[M] Manual Input
[Enter] Rescan
==============================================================================

 Select SSID ? (1~11/Manual/Quit) : 1                  Enter the SSID NO from the list.

 PSK-KEY(ASCII characters 8~63 or Hexadecimal characters 64) ? [Quit]
[123456789|123456789|123456789|123456789|123456789|123456789|1234]
:**********                                            Enter the password for the AP.

 Do you want to set advanced WiFi configuration ? [No/Yes/Quit] (Default No) : N
                                                       Enter N to skip this step.
==============================================
SSID       : AndroidHotspot0568
AUTH       : WPA/WAP2
ENCRYPTION: TKIP/AES(CCMP)
PSK KEY    : 123abc456
KEY TYPE   : ASCII
Hidden AP  : Not connect


 WIFI CONFIGURATION CONFIRM ? [Yes/No/Quit] : Y        Enter Y to confirm the configuration.

 IP Connection Type ? [Automatic IP/Static IP/Quit] : A Enter A for automatic DHCP IP address.

IP Connection Type: Automatic IP

 IP CONFIGURATION CONFIRM ? [Yes/No/Quit] : Y          Enter Y to confirm the configuration.

 SNTP Client enable ? [Yes/No/Quit] : N                Enter N to disable time sync.

 Dialog DPM (Dynamic Power Management) ? [Yes/No/Quit] : N  Enter N to disable DPM.
                                                 See section 6.4.1 for more information about DPM.

Configuration OK
. done
```

The configuration is now complete and stored in NVRAM. The system will reboot automatically and be connected to the selected network.

```
Reboot...

Wakeup source is 0x0
[dpm_init_retmemory] DPM INIT CONFIGURATION(1)

        ******************************************************
        *             DA16200 SDK Information
        * ---------------------------------------------------
        *
        * - CPU Type        : Cortex-M4 (120MHz)
        * - OS Type         : FreeRTOS 10.4.3
        * - Serial Flash    : 4 MB
        * - SDK Version     : V3.2.3.0 GEN
        * - F/W Version     : FRTOS-GEN01-01-6cc84b86f-003714
        * - F/W Build Time  : Apr 13 2022 10:20:02
        * - Boot Index      : 0
        *
```

```
         ******************************************************

System Mode : Station Only (0)
>>> DA16x Supp Ver2.7 - 2020_07
>>> MAC address (sta0) : d4:3d:39:10:a2:48
>>> sta0 interface add OK
>>> Start STA mode...
>>> Network Interface (wlan0) : UP
>>> Associated with ae:2d:be:27:70:7d

Connection COMPLETE to ae:2d:be:27:70:7d

-- DHCP Client WLAN0: SEL(6)

[/DA16200] # -- DHCP Client WLAN0: REQ(1)
-- DHCP Client WLAN0: CHK(8)
-- DHCP Client WLAN0: BOUND(10)
         Assigned addr   : 192.168.0.65
               netmask   : 255.255.255.0
               gateway   : 192.168.0.98
               DNS addr  : 192.168.0.98

         DHCP Server IP  : 192.168.0.98
         Lease Time      : 00h 59m 59s
         Renewal Time    : 00h 29m 59s
```

4.6.2 Setup for Soft AP Mode

Soft AP mode allows the DA16200 to be provisioned through the Wi-Fi interface using a mobile application. Setup for Soft AP Mode is almost the same as for Station Mode and can easily be configured by running the **Easy Setup** Wi-Fi configuration wizard.

To configure the DA16200/DA16600 to operate in Soft AP Mode, open the DA16xxx debug console and run the `setup` command at the `[DA16xxx]` prompt and then answer the questions to complete the setup as follows:

```
[/DA16200] setup                                    Start the Easy Setup Wizard.

Stop all services for the setting.
 Are you sure ? [Yes/No] : Y                        Enter Y to stop the running services.

[ DA16200 EASY SETUP ]

Country Code List:
AD  AE  AF  AI  AL  AM  AR  AS  AT  AU  AW  AZ  BA  BB  BD  BE  BF  BG  BH  BL
BM  BN  BO  BR  BS  BT  BY  BZ  CA  CF  CH  CI  CL  CN  CO  CR  CU  CX  CY  CZ
DE  DK  DM  DO  DZ  EC  EE  EG  ES  ET  EU  FI  FM  FR  GA  GB  GD  GE  GF  GH
GL  GP  GR  GT  GU  GY  HK  HN  HR  HT  HU  ID  IE  IL  IN  IR  IS  IT  JM  JO
JP  KE  KH  KN  KP  KR  KW  KY  KZ  LB  LC  LI  LK  LS  LT  LU  LV  MA  MC  MD
ME  MF  MH  MK  MN  MO  MP  MQ  MR  MT  MU  MV  MW  MX  MY  NG  NI  NL  NO  NP
NZ  OM  PA  PE  PF  PG  PH  PK  PL  PM  PR  PT  PW  PY  QA  RE  RO  RS  RU  RW
SA  SE  SG  SI  SK  SN  SR  SV  SY  TC  TD  TG  TH  TN  TR  TT  TW  TZ  UA  UG
UK  US  UY  UZ  VA  VC  VE  VI  VN  VU  WF  WS  YE  YT  ZA  ZW  ALL

 COUNTRY CODE ? [Quit] (Default KR) : US            Enter the country code.

SYSMODE(WLAN MODE) ?
        1. Station
        2. Soft-AP
 MODE ?  [1/2/Quit] (Default Station) : 2           Enter 2 for Station mode.

[ SOFT-AP CONFIGURATION ]

 SSID ? (Default DA16200_10A249) : TEST AP          Enter the SSID name.

 CHANNEL ? [1~13, Auto:0/Quit] (Default Auto) :     Enter to select Auto channel selection.

AUTHENTICATION ?
         1. OPEN
         2. WEP(Unsupported)
         3. WPA-PSK
         4. WPA2-PSK (Recommend)
        5. WPA/WPA2-PSK
 AUTHENTICATION ? [1/3/4/5/Quit] : 4                Enter 4 to select WPA2-PSK authentication.

ENCRYPTION ?
        1. TKIP (CAUTION: Unsupported 802.11N Mode)
        2. AES(CCMP)
        3. TKIP/AES(CCMP)
 ENCRYPTION ? [1/2/3/Quit] : 2                      Enter 2 to select AES encryption.

 PSK-KEY(ASCII characters 8~63 or Hexadecimal characters 64) ? [Quit]
[123456789|123456789|123456789|123456789|123456789|123456789|1234]
:*********                                          Enter the password for the AP.

 Do you want to set advanced WiFi configuration ? [No/Yes/Quit] (Default No) : N
                                                    Enter N to skip this step.
==================================================
SSID      : TEST AP
CHANNEL   : AUTO(ACS)
AUTH      : WPA2
ENCRYPTION: AES(CCMP)
PSK KEY   : 123abc456
KEY TYPE  : ASCII
WIFI MODE : 11b/g/n
==================================================
 WIFI CONFIGURATION CONFIRM ? [Yes/No/Quit] : Y     Enter Y to confirm the WIFI
                                                    configuration.

 IP ADDRESS ? [Quit] (Default 10.0.0.1) :           Enter to select the default IP Address.

 SUBNET ? [Quit] (Default 255.255.255.0) :          Enter to select the default Subnet.

 GATEWAY ? [Quit] (Default 10.0.0.1) :              Enter to select the default Gateway.
==================================================
[WLAN1]
IP ADDRESS: 10.0.0.1
SUBNET    : 255.255.255.0
GATEWAY   : 10.0.0.1
==================================================
 IP CONFIGURATION CONFIRM ? [Yes/No/Quit] : Y       Enter Y to confirm the IP configuration.

 DHCP SERVER CONFIGURATION ? [Yes/No/Quit] : Y      Enter Y to confirm the DHCP Server
                                                    configuration.


 DHCP SERVER LEASE IP Count(MAX 10) ? [Quit] (Default 10) :
                                                    Enter to select the default Lease Count.

 DHCP SERVER LEASE TIME(60 ~ 86400 SEC) ? [Quit] (Default 1800) :
                                                    Enter to select the default Lease Time.
==================================================
[DHCP SERVER]
Start IP  : 10.0.0.2
END IP    : 10.0.0.11
LEASE TIME: 1800
==================================================

 DHCP SERVER CONFIGURATION CONFIRM ? [Yes/No/Quit] : Y       Enter Y to confirm the configuration.
Configuration OK
. done
```

Once all settings are made, the configuration is saved, and the system reboots. A message is printed that Soft AP mode started successfully:

```
Reboot...

Wakeup source is 0x0
[dpm_init_retmemory] DPM INIT CONFIGURATION(1)

        ******************************************************
        *            DA16200 SDK Information
        * ---------------------------------------------------
        *
        * - CPU Type        : Cortex-M4 (120MHz)
        * - OS Type         : FreeRTOS 10.4.3
        * - Serial Flash    : 4 MB
        * - SDK Version     : V3.1.3.0 GEN
        * - F/W Version     : FRTOS-GEN01-01-15129-000000
        * - F/W Build Time  : Aug 26 2021 22:58:01
        * - Boot Index      : 0
        *
        ******************************************************

System Mode : Soft-AP (1)

>>> DHCP Server Started
>>> DA16x Supp Ver2.7 - 2020_07
>>> Add SoftAP Inteface (softap1) ...
>>> MAC address (softap1) : d4:3d:39:10:a2:49
>>> softap1 interface add OK
>>> AP Operating Channel: AUTO
>>> Soft-AP ACS : ideal ch is 4

>>> Network Interface (wlan1) : UP
BSS Isolate Disabled

Soft-AP is Ready (d4:3d:39:10:a2:49)
[/DA16200] #
```

After Soft AP mode is configured, a mobile application can be used to remotely provision the Wi-Fi interface to operate in Station Mode. See Ref. [8] for details on how to use the mobile application.

4.6.3 Setup for Wi-Fi Provisioning Using Bluetooth® LE

Bluetooth® LE is available on DA16600 only. The DA16600 module can be used in a product such as "Wi-Fi door-lock" where Wi-Fi is the main connection used during normal operation and Bluetooth® LE is a support connection used to do the Wi-Fi Provisioning during the product's initial "Out-of-the-Box" setup.

A Bluetooth® LE peer application such as an Android/IOS mobile App provides an interface to provision the Wi-Fi interface of the DA16600 device by providing configuration information such as a Wi-Fi Home router's SSID, password, server info, etc.

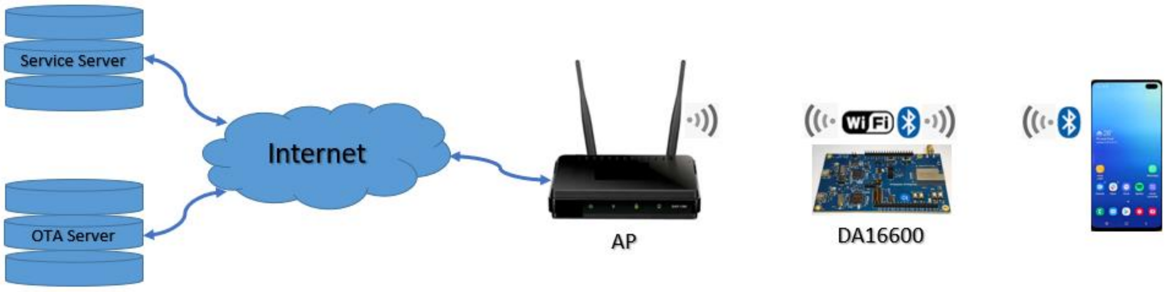


Figure 8: Diagram of Provisioning via Bluetooth® LE

See Ref. [8] for details on how to use the mobile application to provision the Wi-Fi connection on the DA16600.

4.7 Configuring UART/SPI as AT Command Interface

This section describes how to configure the DA16200 or DA16600 EVB for testing AT Command. DA16200 supports AT Command via UART or SPI. For more detailed information for AT Command, see Ref.[5].

Various prebuilt AT Command firmware images can be downloaded from the Renesas Electronics' website. Go to the Renesas website (https://www.renesas.com/us/en/products/wireless-connectivity/wi-fi/low-power-wi-fi) and scroll down to the Software Downloads section. Find "DA16200 DA16600 FreeRTOS SDK Image" or type it in the search box, and then select the firmware and download. Choose DA16200_IMG_FreeRTOS_ATCMD_QFN or DA16600_IMG_FreeRTOS_ATCMD_QFN for your device.

Table 9 shows the default GPIO pin settings of the UART and SPI interfaces of the DA16200 SDK and DA16600 SDK. These GPIOs need to be connected to the MCU.

Table 9: Default GPIO Pin Settings of UART and SPI

Interface	DA16200 SDK	DA16600 SDK
UART	GPIOA4 (UART1_TXD) GPIOA5 (UART1_RXD)	GPIOC6 (UART2_TXD) GPIOC7 (UART2_RXD)
SPI	GPIOA2 (SPI_CSB) GPIOA3 (SPI_CLK) GPIOA8 (SPI_MISO) GPIOA9 (SPI_MOSI) GPIOC6 (INT)	
MCU Wake-up on AT-CMD	GPIOA11 (MCU Wake-up)	

4.7.1 Configuration for Testing AT Command with EVB Only

Both the DA16200 and DA16600 EVBs include an FT2232H which is used to test AT Command over the UART interface. Figure 9 and Table 10 shows how to configure the DA16200 EVB for AT Command processing over the UART. This configuration connects UART1 of the DA16200 (GPIOA4/5) to the FT2232H.

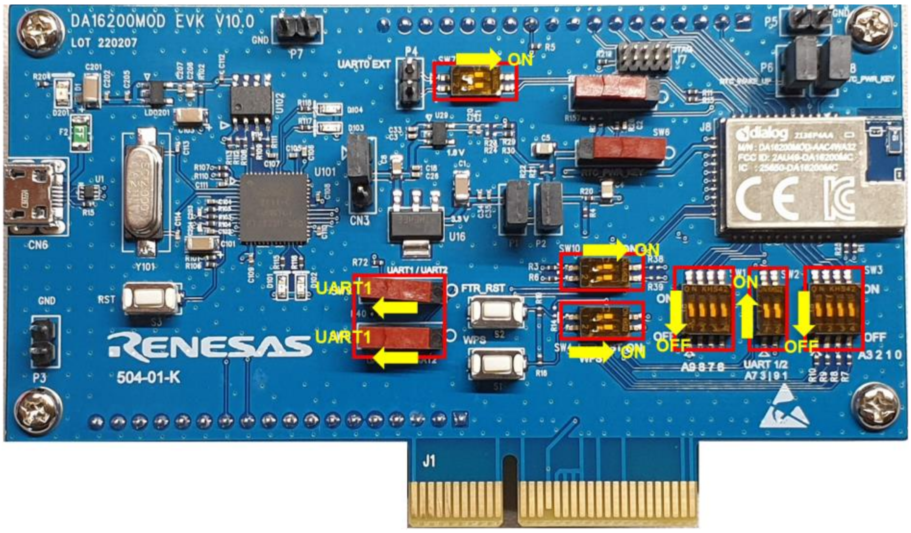


Figure 9: DA16200 EVB Configuration for AT-CMD over UART to FT2232H

Table 10: DA16200 EVB Switch Configuration for AT-CMD over UART of FT2232H

SW1	SW2	SW3	SW4	SW7	SW8	SW9	SW10
All OFF	All ON	All OFF	All ON	All ON	LEFT	LEFT	All ON

Figure 10 and Table 11 show how to configure the DA16600 EVB for AT Command processing over the UART. This configuration connects UART2 of the DA16200 (GPIOC6/7) to the FT2232H.

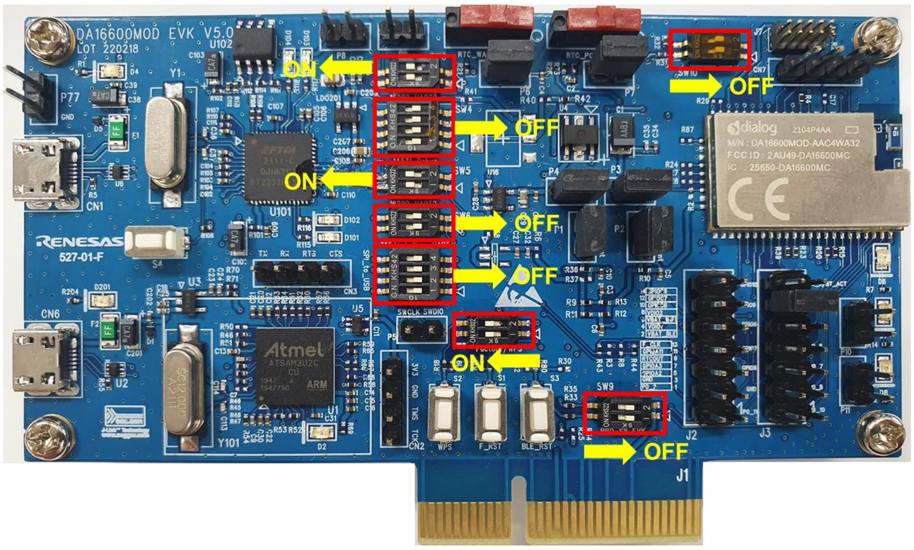


Figure 10: DA16600 EVB Configuration for AT-CMD over UART to FT2232H

Table 11: DA16600 EVB Switch Configuration for AT-CMD over UART of FT2232H

SW3	SW4	SW5	SW6	SW7	SW8	SW9	SW10
All ON	All OFF	All ON	All OFF	All OFF	All ON	All OFF	All OFF

See Ref. [5] for further steps.

4.7.2 Configuration for Testing AT Command Testing with External MCU

Both the DA16200 and DA16600 EVBs have external pins which allows users to test AT Command over the UART or SPI interface using an external MCU.

Figure 11, Table 12 and Table 13 show how to configure the DA16200 EVB for AT Command processing over the UART or SPI interface using an external MCU. This configuration connects UART1 or SPI of the DA16200 (GPIOA4/5 for UART or GPIOA2/3/8/9 for SPI) to the external pin out connectors.

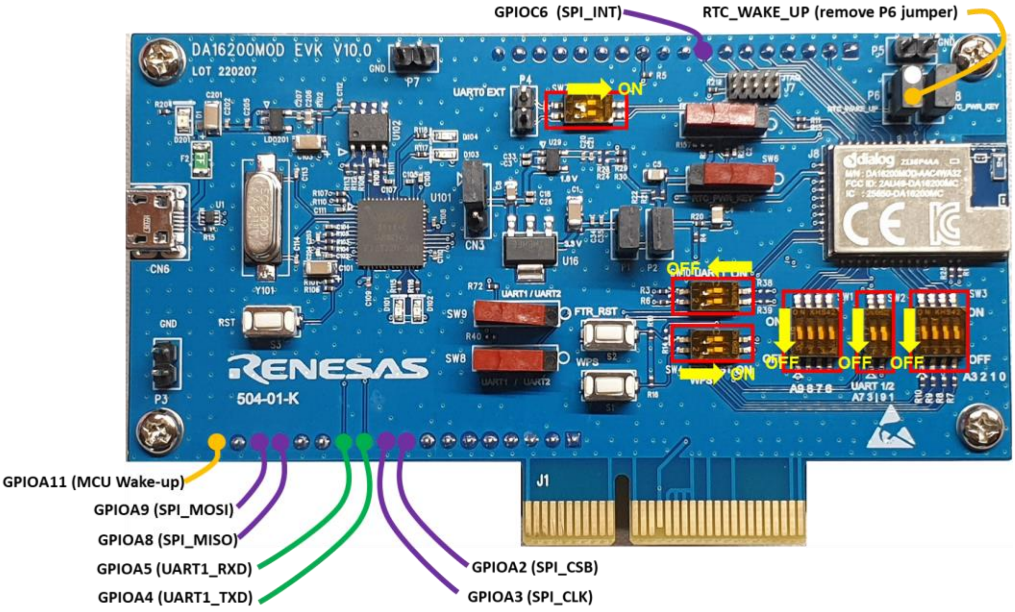


Figure 11: DA16200 EVB Configuration for AT-CMD over UART/SPI to External MCU

Table 12: DA16200 EVB Switch Configuration for AT-CMD over UART/SPI with External MCU

SW1	SW2	SW3	SW4	SW7	SW8	SW9	SW10
All OFF	All OFF	All OFF	All ON	All ON	Don't care	Don't care	All OFF

Table 13: DA16200 EVB Connector Configuration for AT-CMD over UART/SPI with External MCU

Interface	Connector	Pin	Relevant GPIO	Usage
AT Command over UART	P6	1	RTC_WAKE_UP	DA16200 Wake-up from MCU
	J5	11	GPIOA4	UART1_TXD
		12	GPIOA5	UART1_RXD
		18	GPIOA11	MCU Wake-up from DA16200
AT Command over SPI	P6	1	RTC_WAKE_UP	DA16200 Wake-up from MCU
	J2	8	GPIOC6	SPI_INT
	J5	9	GPIOA2	SPI_CSB
		10	GPIOA3	SPI_CLK
		15	GPIOA8	SPI_MISO
		16	GPIOA9	SPI_MOSI
		18	GPIOA11	MCU Wake-up from DA16200

Figure 12, Table 14 and Table 15 show how to configure the DA16600 EVB for AT Command processing over the UART or SPI interface using an external MCU. This configuration connects UART2 or SPI of the DA16600 (GPIOC6/7 for UART or GPIOA2/3/8/9 for SPI) to the external pin out connectors.

When using an AT Command over UART, Pin 11 of J3 and Pin 12 of J3 should be connected by a jumper pin for BLE-Wi-Fi COEX. Also, when using an AT Command over SPI, Pin 13 of J3 and Pin 14 of J3 should be connected.

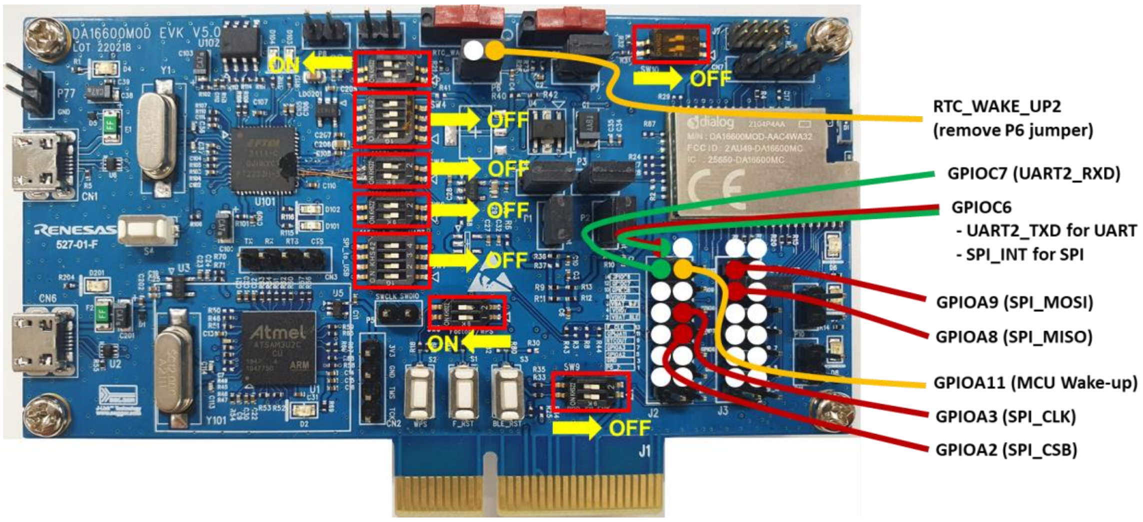


Figure 12: DA16600 EVB Configuration for AT-CMD over UART/SPI with External MCU

Table 14: DA16600 EVB Switch Configuration for AT-CMD over UART/SPI with External MCU

SW3	SW4	SW5	SW6	SW7	SW8	SW9	SW10
All ON	All OFF	All OFF	All OFF	All OFF	All ON	All OFF	All OFF

Table 15: DA16600 EVB Connector Configuration for AT-CMD over UART/SPI with External MCU

Interface	Connector	Pin	Relevant GPIO	Usage
AT Command over UART	P6	1	RTC_WAKE_UP2	DA16200 Wake-up from MCU
	J2	11	GPIOA11	MCU Wake-up from DA16200
		12	GPIOC7	UART2_RXD
		14	GPIOC6	UART2_TXD
		Pin 11 (P0_6) and Pin 12 (GPIOA9) should be connected.		
AT Command over SPI	P6	1	RTC_WAKE_UP2	DA16200 Wake-up from MCU
	J2	5	GPIOA2	SPI_CSB
		7	GPIOA3	SPI_CLK
		11	GPIOA11	MCU Wake-up from DA16200
		14	GPIOC6	SPI_INT
		Pin 13 (P0_6) and Pin 14 (GPIOA10) should be connected.		
	J3	10	GPIOA8	SPI_MISO
		12	GPIOA9	SPI_MOSI

5 Software Development Kit

5.1 Introduction

The DA16200 is a highly integrated ultra-low power Wi-Fi system on chip (SoC) that allows users to develop Wi-Fi solutions using a single chip. Wi-Fi applications can be developed for the DA16200 using the DA16200 FreeRTOS SDK and the Renesas e^{2}studio IDE on either a Windows 10 or Linux based development system.

To start developing applications for the DA16200, follow the steps below:

- Install and configure the e^{2}studio IDE
- Import the DA16200 SDK into the e^{2}studio IDE and build an application
- Download and test the application
- Use J-Link debugger to debug the application

The following describes each of these steps.

5.2 System Requirements

- OS: Windows 10 64bit or Ubuntu 20.04 LTS
- GNU Arm GCC 10.3
- Renesas e^{2}studio 2023-07 (23.7.0) or later
- J-Link Debug Probe

NOTE
For Linux OS ● Install mandatory library for executing e² studio ○ libpython3.10 Open a terminal window and enter the commands below: – sudo add-apt-repository ppa:deadsnakes/ppa – sudo apt-get update – sudo apt-get install libpython3.10

5.3 Installing e^{2}studio IDE

To install the e^{2}studio, download and run the e^{2}studio installer for either Windows or Linux from the e^{2}studio website: https://www.renesas.com/us/en/software-tool/e-studio

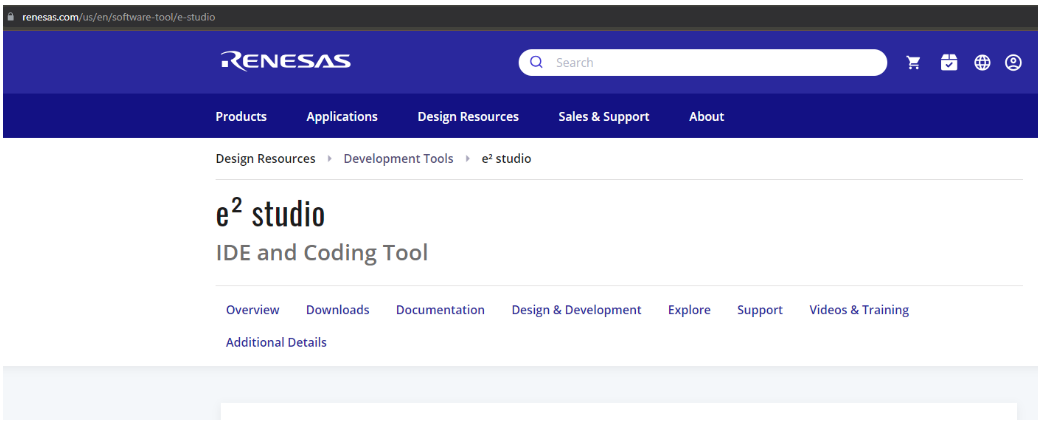
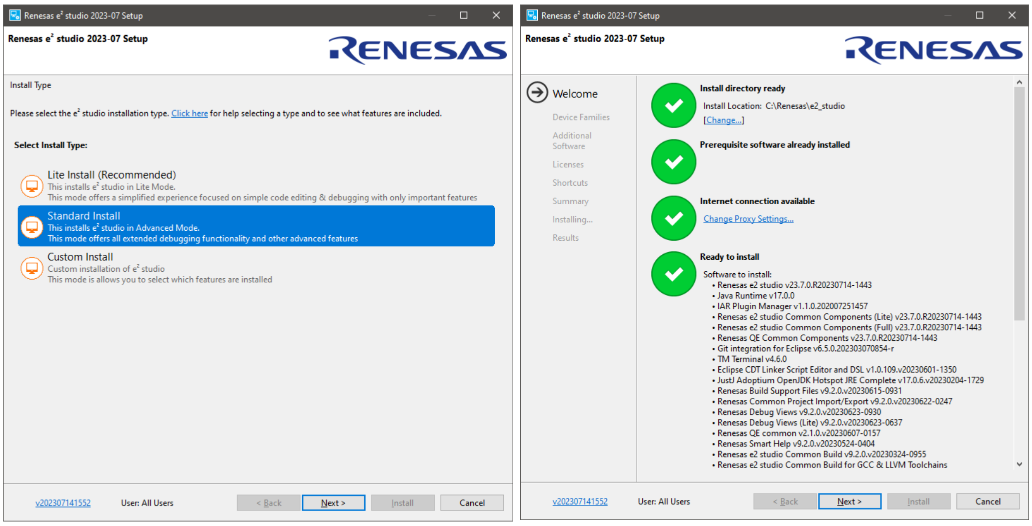


Figure 13: Download e2studio Installer

NOTE
Currently only Windows and Linux have been verified to work with the DA16200 and DA16600 SDKs.

After running the installer, select the **Standard Install** and then select **Next**.



Renesas e^2 studio 2023-07 Setup

Install Type

Please select the e^2 studio installation type. Click here for help selecting a type and to see what features are included.

Select Install Type:

- Lite Install (Recommended)
  This installs e^2 studio in Lite Mode.
  This mode offers a simplified experience focused on simple code editing & debugging with only important features
- Standard Install
  This installs e^2 studio in Advanced Mode.
  This mode offers all extended debugging functionality and other advanced features
- Custom Install
  Custom installation of e^2 studio
  This mode is allows you to select which features are installed

v202307141552 User: All Users

Renesas e^2 studio 2023-07 Setup

Install directory ready
Install Location: C:\Renesas\e2_studio
[Change...]

Prerequisite software already installed

Internet connection available
Change Proxy Settings...

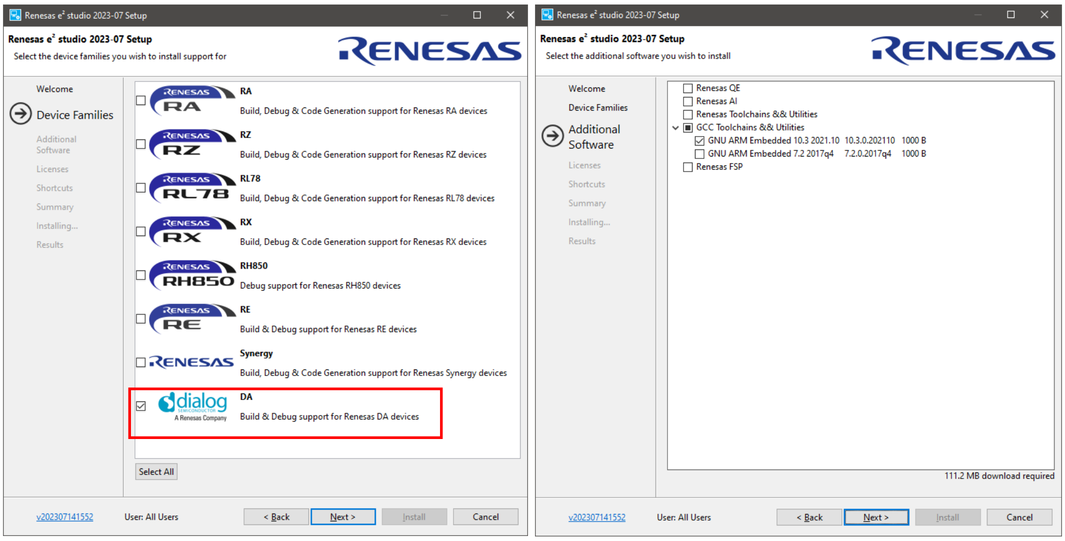
Ready to install
Software to install:

- Renesas e2 studio v23.7.0.R20230714-1443
- Java Runtime v17.0.0
- IAR Plugin Manager v1.1.0.202007251457
- Renesas e2 studio Common Components (Lite) v23.7.0.R20230714-1443
- Renesas e2 studio Common Components (Full) v23.7.0.R20230714-1443
- Renesas QE Common Components v23.7.0.R20230714-1443
- Git integration for Eclipse v6.5.0.202303070854-r
- TM Terminal v4.6.0
- Eclipse CDT Linker Script Editor and DSL v1.0.109.v20230601-1350
- JustJ Adoptium OpenJDK Hotspot JRE Complete v17.0.6.v20230204-1729
- Renesas Build Support Files v9.2.0.v20230615-0931
- Renesas Common Project Import/Export v9.2.0.v20230622-0247
- Renesas Debug Views v9.2.0.v20230623-0930
- Renesas Debug Views (Lite) v9.2.0.v20230623-0637
- Renesas QE common v2.1.0.v20230607-0157
- Renesas Smart Help v9.2.0.v20230524-0404
- Renesas e2 studio Common Build v9.2.0.v20230324-0955
- Renesas e2 studio Common Build for GCC & LLVM Toolchains

v202307141552 User: All Users

Figure 14: Run e^2studio Installer

NOTE
The e^2studio Installer will automatically install the required JRE and embedded development extensions required by the DA16200/DA16600 FreeRTOS SDK.



Renesas e^2 studio 2023-07 Setup

Select the device families you wish to install support for

- ☐ RA: Build, Debug & Code Generation support for Renesas RA devices
- ☐ RZ: Build, Debug & Code Generation support for Renesas RZ devices
- ☐ RL78: Build, Debug & Code Generation support for Renesas RL78 devices
- ☐ RX: Build, Debug & Code Generation support for Renesas RX devices
- ☐ RH850: Debug support for Renesas RH850 devices
- ☐ RE: Build & Debug support for Renesas RE devices
- ☐ Synergy: Build, Debug & Code Generation support for Renesas Synergy devices
- ☑ DA: Build & Debug support for Renesas DA devices

v202307141552 User: All Users

Renesas e^2 studio 2023-07 Setup

Select the additional software you wish to install

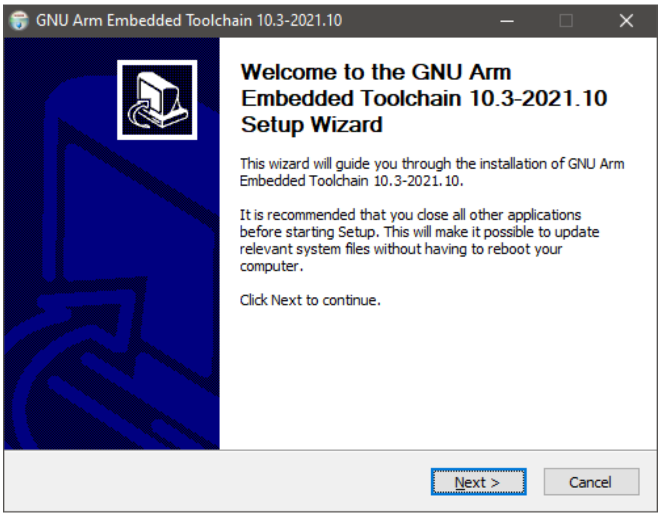
- ☐ Renesas QE
- ☐ Renesas AI
- ☐ Renesas Toolchains && Utilities
- ■ GCC Toolchains && Utilities
  - ☑ GNU ARM Embedded 10.3 2021.10 10.3.0.202110 1000 B
  - ☐ GNU ARM Embedded 7.2 2017q4 7.2.0.2017q4 1000 B
- ☐ Renesas FSP

111.2 MB download required

v202307141552 User: All Users

Figure 15: e^2studio Setup

Select the DA family in Device Families. The GCC toolchain is selected by default as GNU ARM Embedded 10.3 2021.10.



Welcome to the GNU Arm Embedded Toolchain 10.3-2021.10 Setup Wizard

This wizard will guide you through the installation of GNU Arm Embedded Toolchain 10.3-2021.10.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Figure 16: Install GNU Arm Embedded Toolchain

The GNU Arm Embedded Toolchain 10.3 must be installed. The Toolchain installation starts automatically during the e^{2}studio installation process. After installing the e^{2}studio and the GNU Arm Embedded Toolchain, run e^{2}studio and open the workspace.

5.4 Importing DA16200 FreeRTOS SDK Project into e^{2}studio

Download the DA16200 FreeRTOS SDK from the official website. Go to the Renesas website (https://www.renesas.com/us/en/products/wireless-connectivity/wi-fi/low-power-wi-fi) and scroll down to the Software Downloads section. Find “DA16200 DA16600 FreeRTOS SDK” or type it in the search box, and then select the firmware and download.

Create a workspace directory for the SDK. For example:

```
..../projects/da16xxx_workspace
```

Extract the SDK zip file into a directory under that workspace directory. For example:

```
..../projects/da16xxx_workspace/DA16200_DA16600_SDK_V3.2.x.x
```

This directory is known as the `<sdk_root_directory>`.

Run e^{2}studio and open the newly created workspace.

NOTE
For Linux systems, certain files used during the build process need to be set as executable. After extracting the SDK files into a directory, the `<sdk_root_directory>/tools/util` file permissions must be changed by running the `set_linux_perm.sh` script in the `<sdk_root_directory>/tools/util/` directory:

```
~$
~$ cd <sdk_root_diretory>/tools/util
~$ chmod 755 set_linux_perm.sh
~$ sh ./set_linux_perm.sh
~$
```

Various projects are contained within the SDK directory structure. All of these projects are stored in the `<sdk_root_directory>/apps/` directory.

These include the following:

- Default SDK build for the DA16200 and DA16600
  - ○ `<sdk_root_directory>/apps/da16200/get_started/`
  - ○ `<sdk_root_directory>/apps/da16600/get_started/`

- Various example projects
  - `<sdk_root_directory>/apps/common/examples/Crypto/`
  - `<sdk_root_directory>/apps/common/examples/DPM/`
  - `<sdk_root_directory>/apps/common/examples/ETC/`
  - `<sdk_root_directory>/apps/common/examples/Network/`
  - `<sdk_root_directory>/apps/common/examples/Periphial/`

The example project directories all have a similar structure, containing a project folder for either the DA16200 or DA16600:

- `<sdk_root_directory>/apps/common/examples/Crypto/Crypto_AES/projects/da16200`

`<sdk_root_directory>/apps/common/examples/Crypto/Crypto_AES/projects/da16600`

Import the da16200/da16600 project into the e²studio workspace as follows:

1. Under the **File** menu select **Import...** to open the Import dialog box and then select **General** > **Dialog SDK Project** and click **Next**.

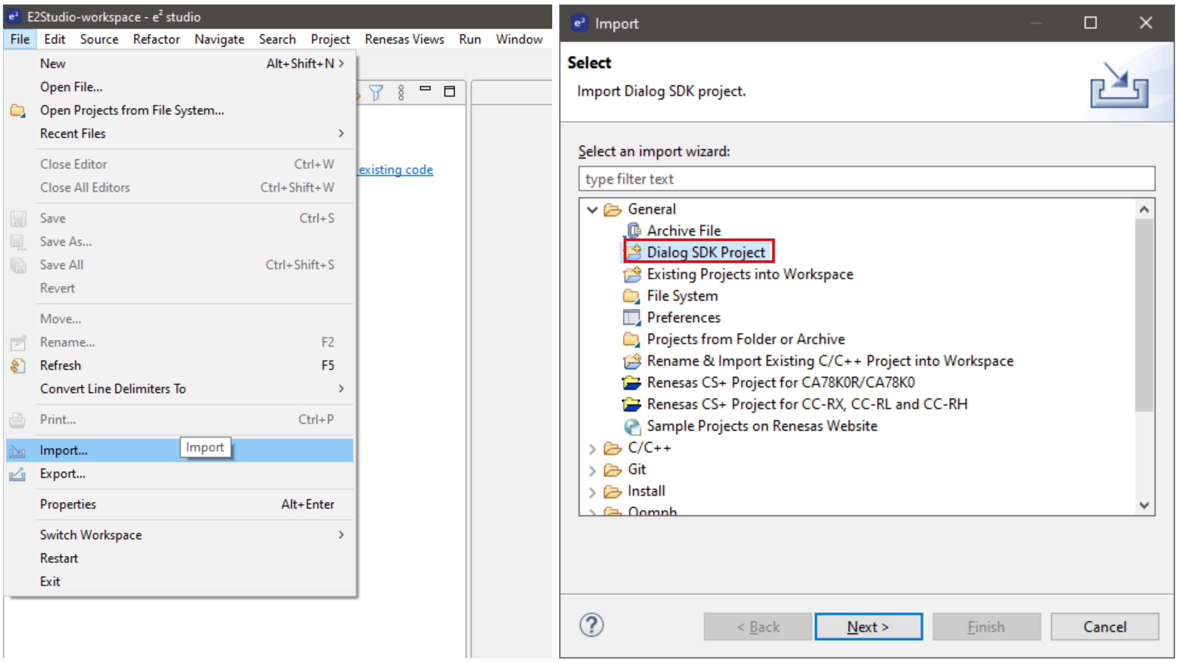
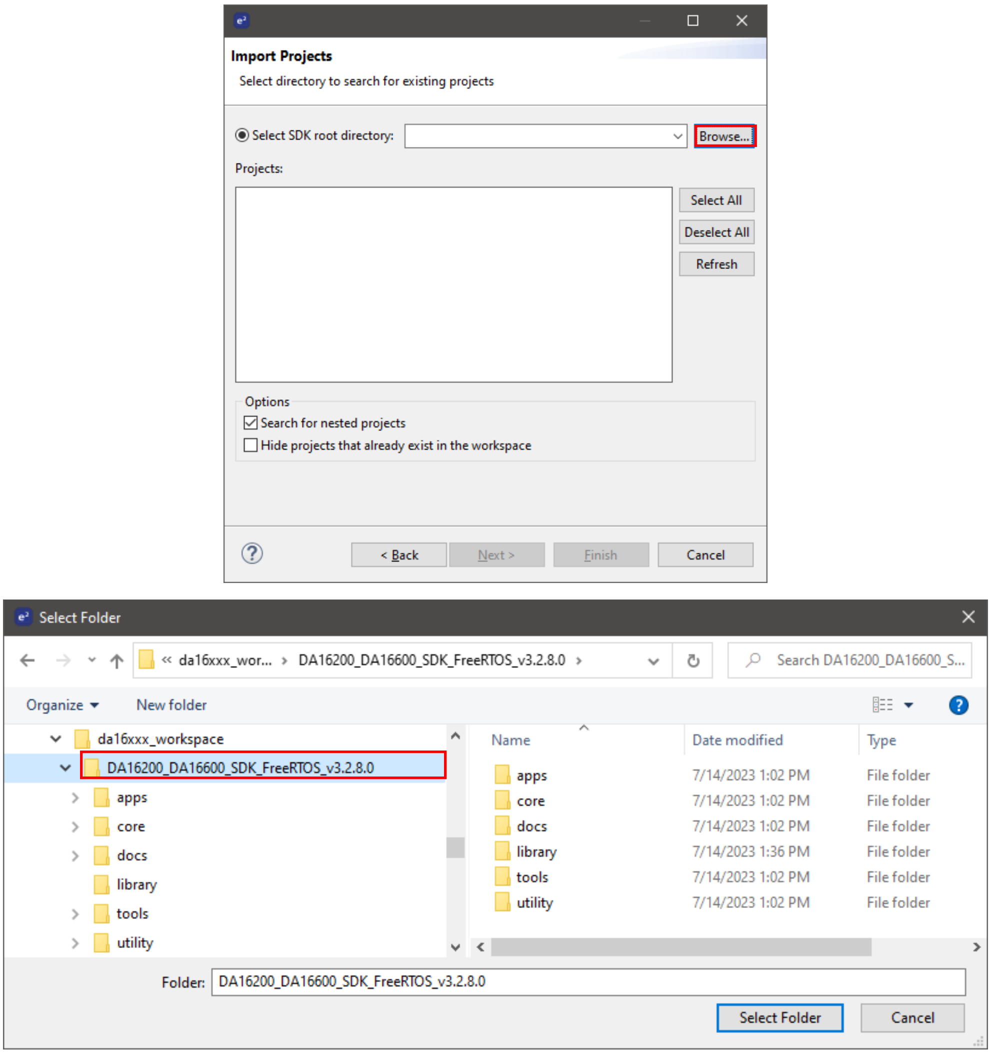

E2Studio-workspace - e² studio
File Edit Source Refactor Navigate Search Project Renesas Views Run Window
New Alt+Shift+N
Open File...
Open Projects from File System...
Recent Files
Close Editor Ctrl+W
Close All Editors Ctrl+Shift+W
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Revert
Move...
Rename... F2
Refresh F5
Convert Line Delimiters To
Print... Ctrl+P
Import...
Import
Export...
Properties Alt+Enter
Switch Workspace
Restart
Exit
existing code
Import
Select
Import Dialog SDK project.
Select an import wizard:
type filter text
General
Archive File
Dialog SDK Project
Existing Projects into Workspace
File System
Preferences
Projects from Folder or Archive
Rename & Import Existing C/C++ Project into Workspace
Renesas CS+ Project for CA78K0R/CA78K0
Renesas CS+ Project for CC-RX, CC-RL and CC-RH
Sample Projects on Renesas Website
C/C++
Git
Install
Oomph
< Back
Next >
Finish
Cancel

Figure 17: Import SDK Project to e²studio IDE

NOTE
Windows 10 has a path length limitation of 260 characters. The `<sdk_root_directory>` path length must be less than 70 characters.

2. In the **Import** dialog box, select the **Select SDK root directory** and click the **Browse** button. Use the file manager to navigate to the `<sdk_root_directory>/` directory and then click **Select Folder**.



Import Projects

Select directory to search for existing projects

● Select SDK root directory:

Projects:

Options

☑ Search for nested projects

☐ Hide projects that already exist in the workspace

Select Folder

« da16xxx_wor... › DA16200_DA16600_SDK_FreeRTOS_v3.2.8.0 ›

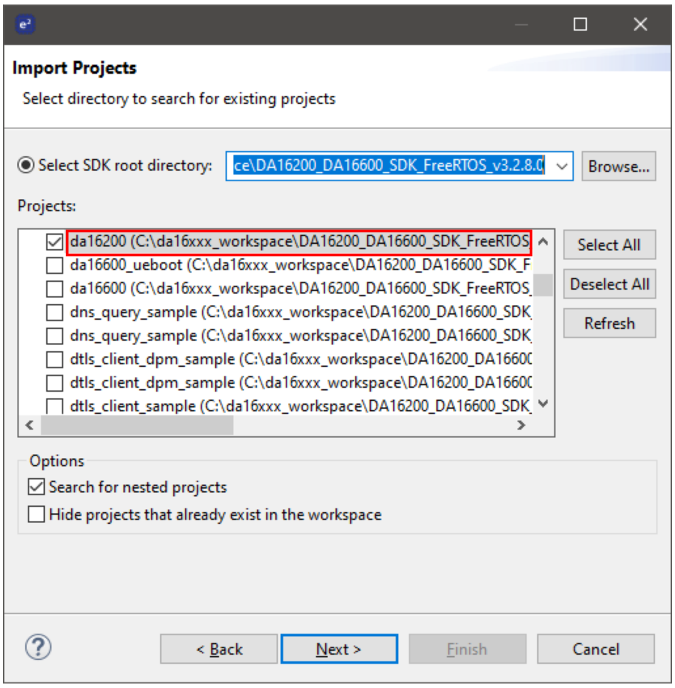
Search DA16200_DA16600_S...

Name	Date modified	Type
apps	7/14/2023 1:02 PM	File folder
core	7/14/2023 1:02 PM	File folder
docs	7/14/2023 1:02 PM	File folder
library	7/14/2023 1:36 PM	File folder
tools	7/14/2023 1:02 PM	File folder
utility	7/14/2023 1:02 PM	File folder

Folder: DA16200_DA16600_SDK_FreeRTOS_v3.2.8.0

Figure 18: Select Folder to Import SDK Project into e^{2}studio IDE

3. Various projects will appear in the **Projects** list. Select the **da16200 or da16600 project**. The second project to select is the **SDKJFlash project** which provides an absolute path for certain scripts to use and click **Next**.



Import Projects

Select directory to search for existing projects

● Select SDK root directory: ce\DA16200_DA16600_SDK_FreeRTOS_v3.2.8.0 | Browse...

Projects:

- ☑ da16200 (C:\da16xxx_workspace\DA16200_DA16600_SDK_FreeRTOS
- □ da16600_ueboot (C:\da16xxx_workspace\DA16200_DA16600_SDK_F
- □ da16600 (C:\da16xxx_workspace\DA16200_DA16600_SDK_FreeRTOS_
- □ dns_query_sample (C:\da16xxx_workspace\DA16200_DA16600_SDK_
- □ dns_query_sample (C:\da16xxx_workspace\DA16200_DA16600_SDK_
- □ dtls_client_dpm_sample (C:\da16xxx_workspace\DA16200_DA1660(
- □ dtls_client_dpm_sample (C:\da16xxx_workspace\DA16200_DA1660(
- □ dtls_client_sample (C:\da16xxx_workspace\DA16200_DA16600_SDK_

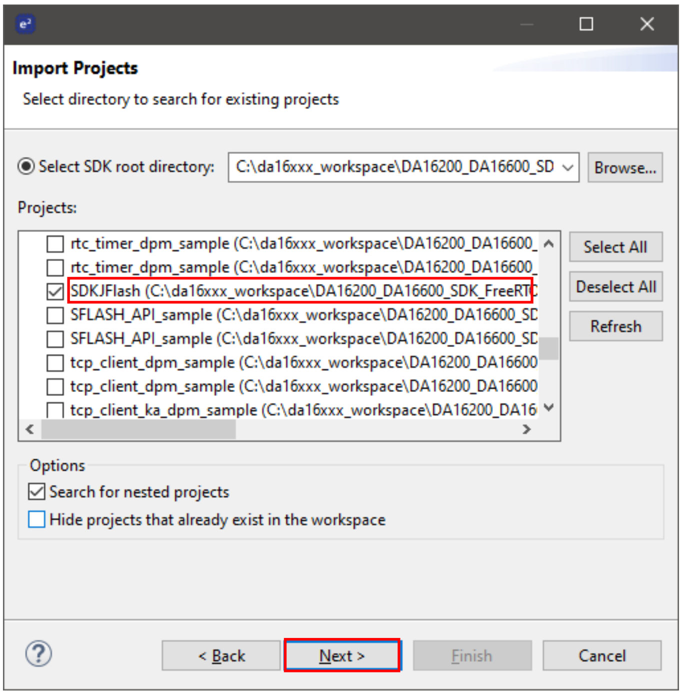
Select All | Deselect All | Refresh

Options

- ☑ Search for nested projects
- □ Hide projects that already exist in the workspace

< Back | Next > | Finish | Cancel

Figure 19: Import DA16200/DA16600 Project to e^{2}studio IDE



Import Projects

Select directory to search for existing projects

● Select SDK root directory: C:\da16xxx_workspace\DA16200_DA16600_SD | Browse...

Projects:

- □ rtc_timer_dpm_sample (C:\da16xxx_workspace\DA16200_DA16600_
- □ rtc_timer_dpm_sample (C:\da16xxx_workspace\DA16200_DA16600_
- ☑ SDKJFlash (C:\da16xxx_workspace\DA16200_DA16600_SDK_FreeRT(
- □ SFLASH_API_sample (C:\da16xxx_workspace\DA16200_DA16600_SD
- □ SFLASH_API_sample (C:\da16xxx_workspace\DA16200_DA16600_SD
- □ tcp_client_dpm_sample (C:\da16xxx_workspace\DA16200_DA16600
- □ tcp_client_dpm_sample (C:\da16xxx_workspace\DA16200_DA16600
- □ tcp_client_ka_dpm_sample (C:\da16xxx_workspace\DA16200_DA16

Select All | Deselect All | Refresh

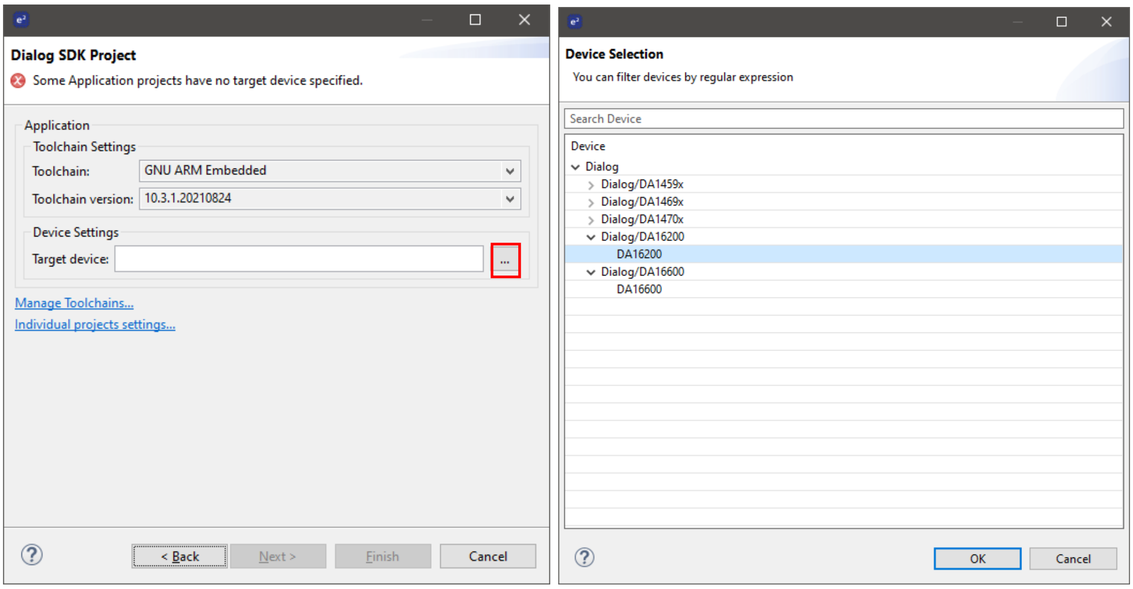
Options

- ☑ Search for nested projects
- □ Hide projects that already exist in the workspace

< Back | Next > | Finish | Cancel

Figure 20: Import SDKJFlash Project to e^{2}studio IDE

4. Dialog SDK Project window will appear as shown below. Select Target device DA16200 or DA16600 by selecting the ... button and then click **Finish**.



Dialog SDK Project

Some Application projects have no target device specified.

Application

Toolchain Settings

Toolchain: GNU ARM Embedded

Toolchain version: 10.3.1.20210824

Device Settings

Target device:

Device Selection

You can filter devices by regular expression

Search Device

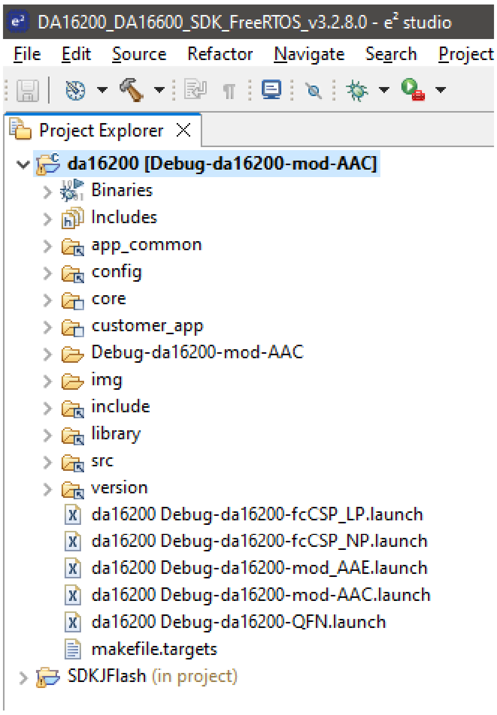
Device
- Dialog
  - Dialog/DA1459x
  - Dialog/DA1469x
  - Dialog/DA1470x
  - Dialog/DA16200
    - DA16200
  - Dialog/DA16600
    - DA16600

Figure 21: Dialog SDK Project

The basic setup for importing the SDK project is now complete. The next step is to build that project.

5.5 Build Project

1. Once the target projects are imported, it appears in the e^{2}studio Project Explorer. Build the DA16200/DA16600 project by right-clicking on the imported project and select **Build Project**.



DA16200_DA16600_SDK_FreeRTOS_v3.2.8.0 - e² studio

- da16200 [Debug-da16200-mod-AAC]
  - Binaries
  - Includes
  - app_common
  - config
  - core
  - customer_app
  - Debug-da16200-mod-AAC
  - img
  - include
  - library
  - src
  - version
  - da16200 Debug-da16200-fcCSP_LP.launch
  - da16200 Debug-da16200-fcCSP_NP.launch
  - da16200 Debug-da16200-mod_AAE.launch
  - da16200 Debug-da16200-mod-AAC.launch
  - da16200 Debug-da16200-QFN.launch
  - makefile.targets
- SDKJFlash (in project)

Figure 22: e^{2}studio Project Explorer

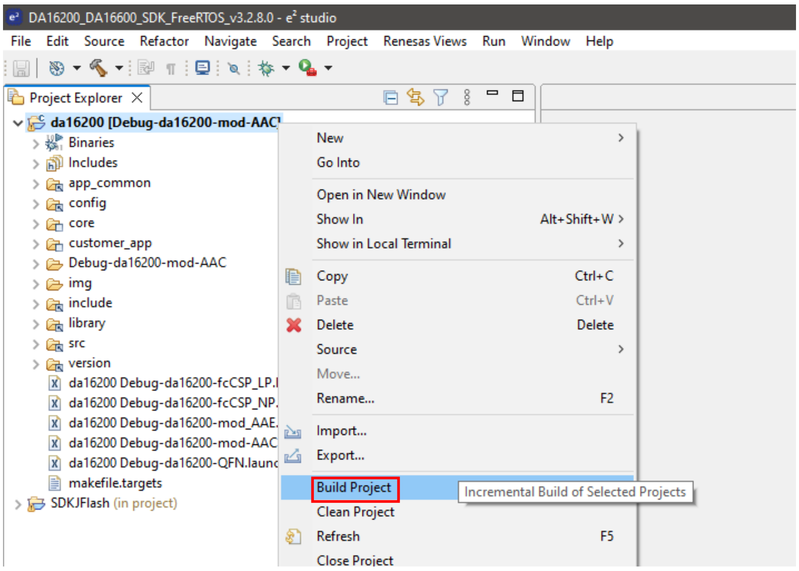
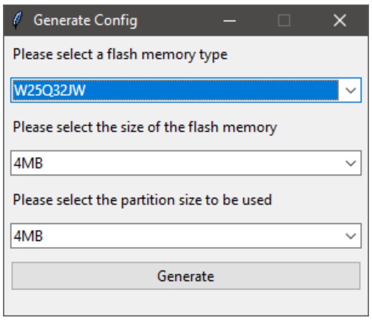

DA16200_DA16600_SDK_FreeRTOS_v3.2.8.0 - e² studio
File Edit Source Refactor Navigate Search Project Renesas Views Run Window Help
Project Explorer
da16200 [Debug-da16200-mod-AAC]
Binaries
Includes
app_common
config
core
customer_app
Debug-da16200-mod-AAC
img
include
library
src
version
da16200 Debug-da16200-fcCSP_LP.l
da16200 Debug-da16200-fcCSP_NP.
da16200 Debug-da16200-mod_AAE.
da16200 Debug-da16200-mod-AAC
da16200 Debug-da16200-QFN.launc
makefile.targets
SDKJFlash (in project)
New
Go Into
Open in New Window
Show In Alt+Shift+W
Show in Local Terminal
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Source
Move...
Rename... F2
Import...
Export...
Build Project
Incremental Build of Selected Projects
Clean Project
Refresh F5
Close Project

Figure 23: Build Project

NOTE
The SDK building project process may take several minutes depending on your computer.

2. If this build process is performed for the first time, the Generate Configuration pop-up window is displayed automatically so that the flash memory type can be selected. Select the flash type and click **Generate** to create the appropriate flash configuration file required to build the firmware.



Please select a flash memory type

W25Q32JW

Please select the size of the flash memory

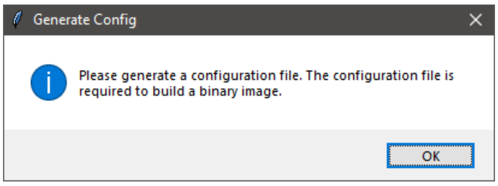
4MB

Please select the partition size to be used

4MB

Figure 24: Generate Configuration

If the pop-up window is closed without selecting and generating a configuration file, a warning pop-up will be displayed.



Please generate a configuration file. The configuration file is required to build a binary image.

Figure 25: Generate Configuration Warning Pop-Up

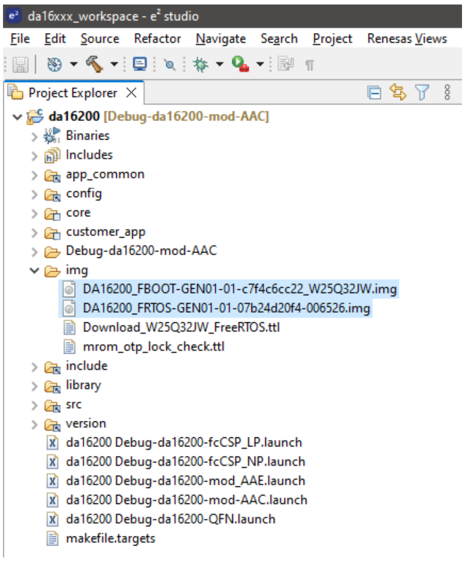
This step cannot be skipped due to a flash configuration file is required to build the firmware. When the build is complete, the following output is displayed in the e^{2}studio console window.

```
*-------------------------------------
*Image Generate success
*-------------------------------------

[CM.3.secuboot.bat] END
[.\util\mk_sboot_image.bat] END
*=====================================
*Post-Build Clean Start for Windows
*=====================================
Start mk_sboot_image_clean.bat
*=====================================
*Post-Build End for Windows
*=====================================

arm-none-eabi-size --format=berkeley "da16200.elf"
   text    data     bss     dec     hex filename
1145404   10224  141152 1296780  13c98c da16200.elf

15:56:21 Build Finished. 0 errors, 0 warnings. (took 4m:10s.749ms)
```

There are two firmware images created by the build process and they are stored in the `<sdk_root_directory>/<project_path>/img/` directory. For Example:

```
DA16200_FBOOT-GEN01-01-c7f4c6cc22_W25Q32JW.img
DA16200_FRTOS-GEN01-01-07b24d20f4-006526.img
```

The images can be found in the directory show in Figure 26.



da16xxx_workspace - e² studio

File Edit Source Refactor Navigate Search Project Renesas Views

Project Explorer

```
da16200 [Debug-da16200-mod-AAC]
  Binaries
  Includes
  app_common
  config
  core
  customer_app
  Debug-da16200-mod-AAC
  img
    DA16200_FBOOT-GEN01-01-c7f4c6cc22_W25Q32JW.img
    DA16200_FRTOS-GEN01-01-07b24d20f4-006526.img
    Download_W25Q32JW_FreeRTOS.ttl
    mrom_otp_lock_check.ttl
  include
  library
  src
  version
  da16200 Debug-da16200-fcCSP_LP.launch
  da16200 Debug-da16200-fcCSP_NP.launch
  da16200 Debug-da16200-mod_AAE.launch
  da16200 Debug-da16200-mod-AAC.launch
  da16200 Debug-da16200-QFN.launch
  makefile.targets
```

Figure 26: Location of Firmware Image Files

FBOOT is the bootloader image which is used to initialize the DA16200 and launch the main firmware.

- The bootloader image should be loaded first into the flash of a new device
- The bootloader image contains SFDP (flash specific) information. Note that the bootloader must be loaded into flash before loading any other images
- When updating the SDK, always load the bootloader image first

FRTOS is the main firmware image which includes the RTOS and user applications.

5.5.1 Build Configurations

There are several types of build configuration for DA16200 / DA16600 SDK project. Release and Debug configurations are almost the same except that the Release build configurations disable the console (UART0) input and output.

- DA16200 Build Configurations
  - Debug
    - Debug-da16200-mod-AAC : DA16200 AAC module type (default)
    - Debug-da16200-mod-AAE : DA16200 AAE module type Debug Build
    - Debug-da16200-QFN : DA16200 QFN chipset type Debug Build
    - Debug-da16200-fcCSP_LP : DA16200 fcCSP Low Power chipset type Debug Build
    - Debug-da16200-fcCSP_NP : DA16200 fcCSP Normal Power chipset Type Debug Build
  - Release
    - Release-da16200-mod-AAC : DA16200 AAC module type Release Build
    - Release -da16200-mod-AAE : DA16200 AAE module type Release Build
    - Release -da16200-QFN : DA16200 QFN chipset type Release Build
    - Release -da16200-fcCSP_LP : DA16200 fcCSP Low Power chipset type Release Build
    - Release -da16200-fcCSP_NP : DA16200 fcCSP Normal Power chipset Type Release Build

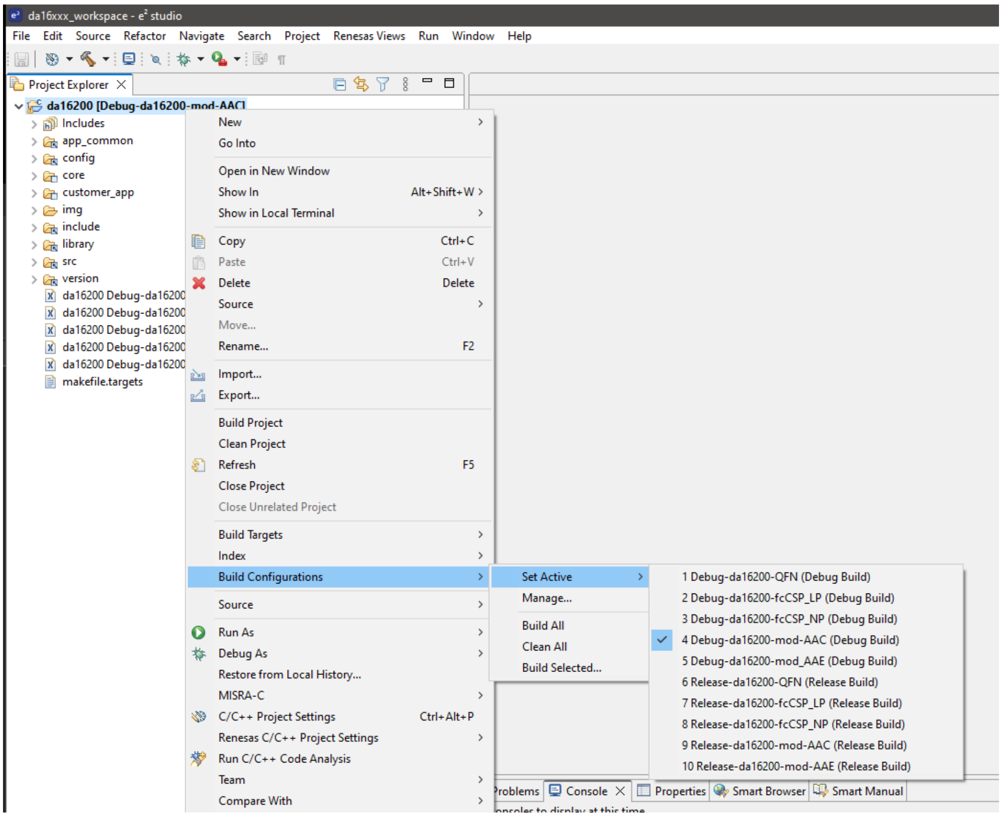

da16xxx_workspace - e² studio
File Edit Source Refactor Navigate Search Project Renesas Views Run Window Help
Project Explorer
da16200 [Debug-da16200-mod-AAC]
Includes
app_common
config
core
customer_app
img
include
library
src
version
da16200 Debug-da16200
da16200 Debug-da16200
da16200 Debug-da16200
da16200 Debug-da16200
da16200 Debug-da16200
makefile.targets
New
Go Into
Open in New Window
Show In Alt+Shift+W
Show in Local Terminal
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Source
Move...
Rename... F2
Import...
Export...
Build Project
Clean Project
Refresh F5
Close Project
Close Unrelated Project
Build Targets
Index
Build Configurations
Source
Run As
Debug As
Restore from Local History...
MISRA-C
C/C++ Project Settings Ctrl+Alt+P
Renesas C/C++ Project Settings
Run C/C++ Code Analysis
Team
Compare With
Set Active
Manage...
Build All
Clean All
Build Selected...
1 Debug-da16200-QFN (Debug Build)
2 Debug-da16200-fcCSP_LP (Debug Build)
3 Debug-da16200-fcCSP_NP (Debug Build)
✓ 4 Debug-da16200-mod-AAC (Debug Build)
5 Debug-da16200-mod_AAE (Debug Build)
6 Release-da16200-QFN (Release Build)
7 Release-da16200-fcCSP_LP (Release Build)
8 Release-da16200-fcCSP_NP (Release Build)
9 Release-da16200-mod-AAC (Release Build)
10 Release-da16200-mod-AAE (Release Build)
Problems Console Properties Smart Browser Smart Manual
onsoles to display at this time

Figure 27: DA16200 Build Configurations

- DA16600 Build Configurations
  - Debug
    - Debug-da16600-mod : DA16600 module type Debug Build (default)
  - Release
    - Release-da16600-mod : DA16600 module type Release Build

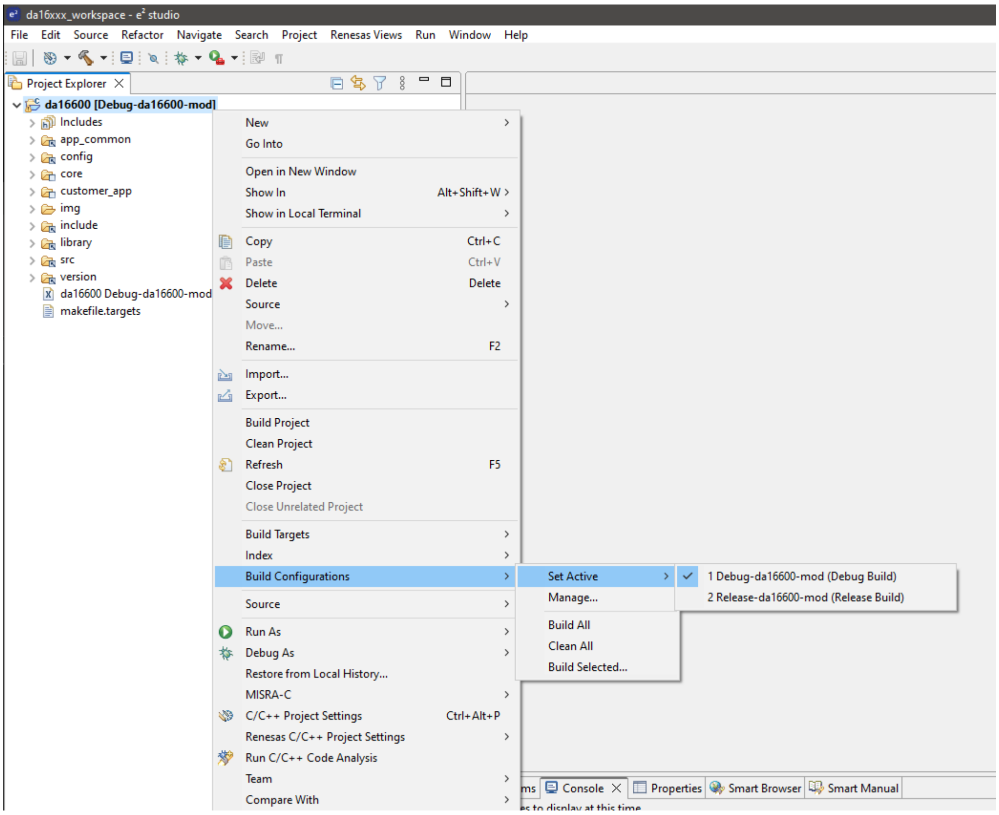


Figure 28: DA16600 Build Configurations

5.5.2 Changing Flash Configuration

The flash configuration can be changed by running the **Generate Configuration tool** which is provided by the SDK.

To use the Generate Configuration tool, import Launch Configurations:

- File>Import>Run/Debug/Launch Configurations

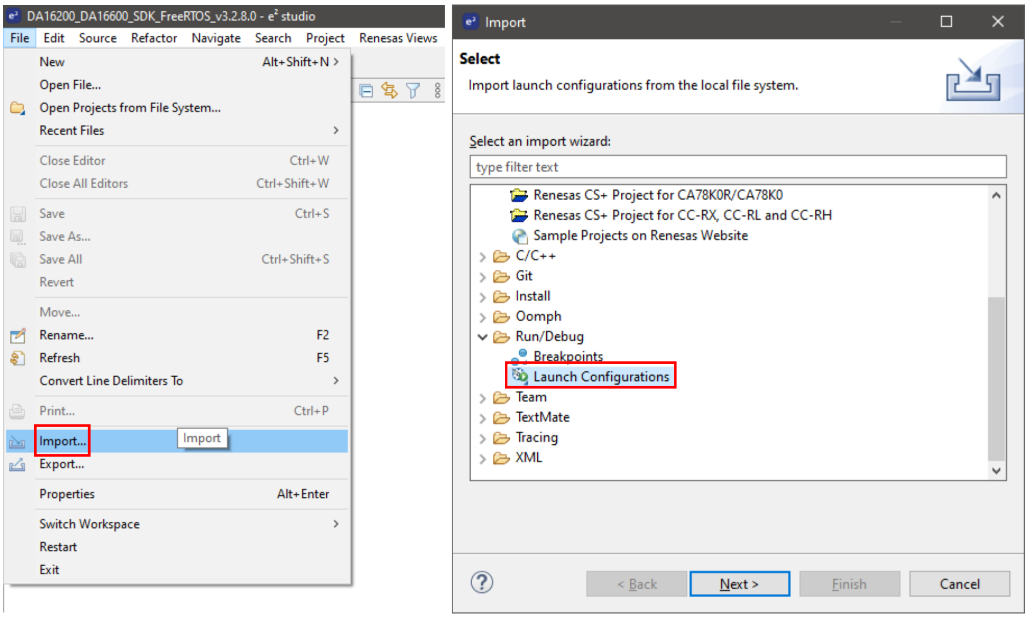


Figure 29: Import Launch Configuration

- Select **<sdk_root_directory>/utility/cfg_generator** directory using Browse button

- Select Generate Configuration launch file and then, click **Finish**

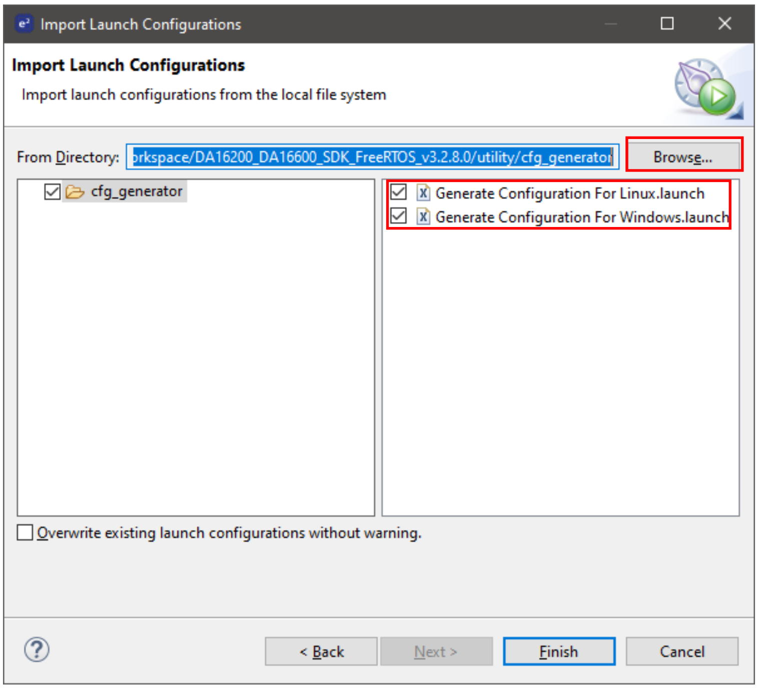

Import Launch Configurations
Import Launch Configurations
Import launch configurations from the local file system
From Directory:
orkspace/DA16200_DA16600_SDK_FreeRTOS_v3.2.8.0/utility/cfg_generator
Browse...
☑ cfg_generator
☑ Generate Configuration For Linux.launch
☑ Generate Configuration For Windows.launch
□ Overwrite existing launch configurations without warning.
< Back
Next >
Finish
Cancel

Figure 30: Import Generate Configuration

Generate Configuration can be run by going to the **Run > External Tools** menu and selecting either **Generate Configuration For Windows** or **Generate Configuration For Linux**

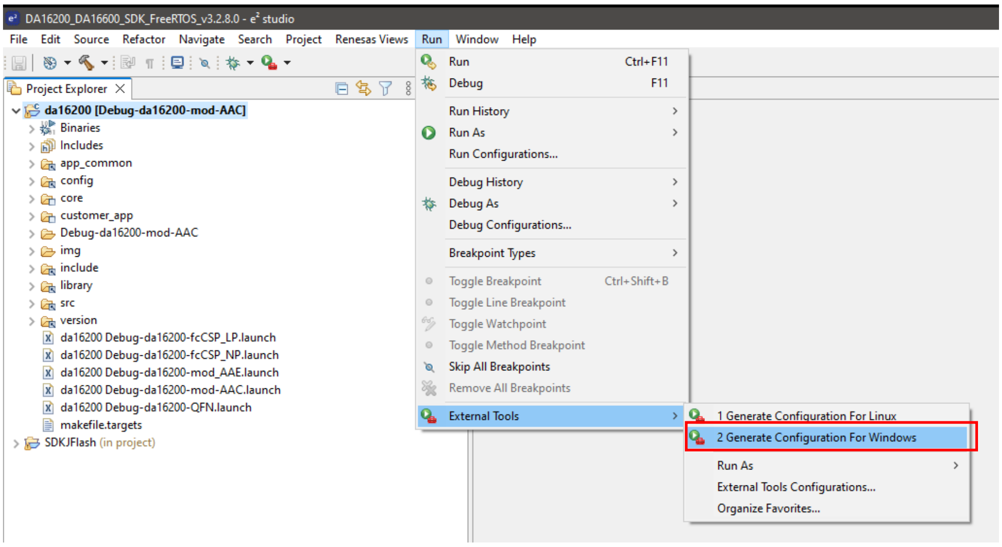

DA16200_DA16600_SDK_FreeRTOS_v3.2.8.0 - e² studio
File Edit Source Refactor Navigate Search Project Renesas Views Run Window Help
Project Explorer
da16200 [Debug-da16200-mod-AAC]
Binaries
Includes
app_common
config
core
customer_app
Debug-da16200-mod-AAC
img
include
library
src
version
da16200 Debug-da16200-fcCSP_LP.launch
da16200 Debug-da16200-fcCSP_NP.launch
da16200 Debug-da16200-mod_AAE.launch
da16200 Debug-da16200-mod-AAC.launch
da16200 Debug-da16200-QFN.launch
makefile.targets
SDKJFlash (in project)
Run Ctrl+F11
Debug F11
Run History
Run As
Run Configurations...
Debug History
Debug As
Debug Configurations...
Breakpoint Types
Toggle Breakpoint Ctrl+Shift+B
Toggle Line Breakpoint
Toggle Watchpoint
Toggle Method Breakpoint
Skip All Breakpoints
Remove All Breakpoints
External Tools
1 Generate Configuration For Linux
2 Generate Configuration For Windows
Run As
External Tools Configurations...
Organize Favorites...

Figure 31: Run Generate Configuration

The Generate Configuration pop-up window is displayed as shown Figure 24. Select the flash type, then click on generate to create the appropriate flash configuration file required to build the firmware.

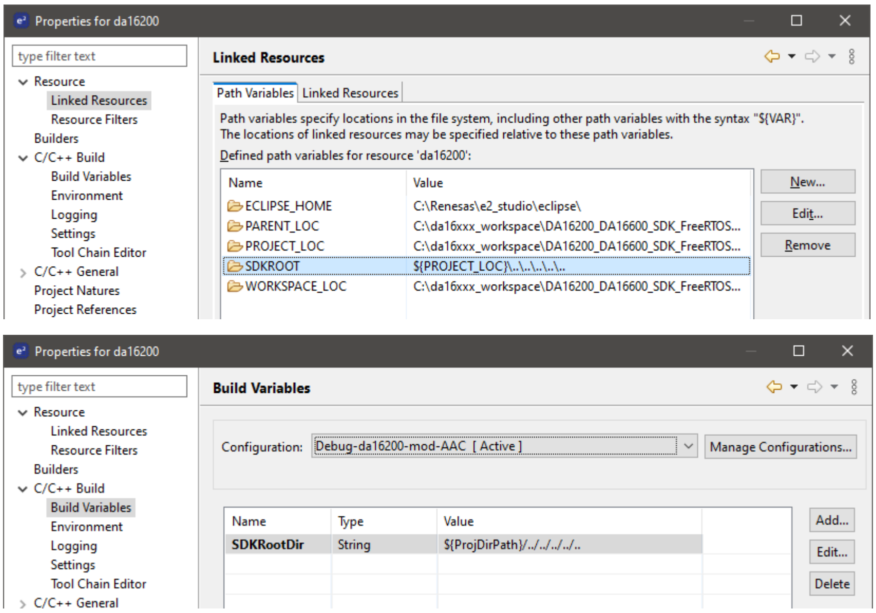
5.5.3 How to Move Project Location

Project location can be moved to another folder. In order to so do, SDKROOT and SDKRootDir variables in the e^{2}studio must be changed.

SDKROOT/SDKRootDir paths are the SDK Root path from the project location.(SDKROOT and SDKRootDir must be same path.)

e^{2}studio → Project → Properties → Resource → Linked Resources → SDKROOT

e^{2}studio → Project → Properties → C/C++ Build → Build Variable → SDKRootDir



Path variables specify locations in the file system, including other path variables with the syntax "${VAR}". The locations of linked resources may be specified relative to these path variables.

Defined path variables for resource 'da16200':

Name	Value
ECLIPSE_HOME	C:\Renesas\e2_studio\eclipse\
PARENT_LOC	C:\da16xxx_workspace\DA16200_DA16600_SDK_FreeRTOS...
PROJECT_LOC	C:\da16xxx_workspace\DA16200_DA16600_SDK_FreeRTOS...
SDKROOT	${PROJECT_LOC}\..\..\..\..\..
WORKSPACE_LOC	C:\da16xxx_workspace\DA16200_DA16600_SDK_FreeRTOS...

Configuration: Debug-da16200-mod-AAC [Active]

Name	Type	Value
SDKRootDir	String	${ProjDirPath}/../../../../..

Figure 32: SDK Root Path Variables

5.6 Debugging with J-Link Debug Probe

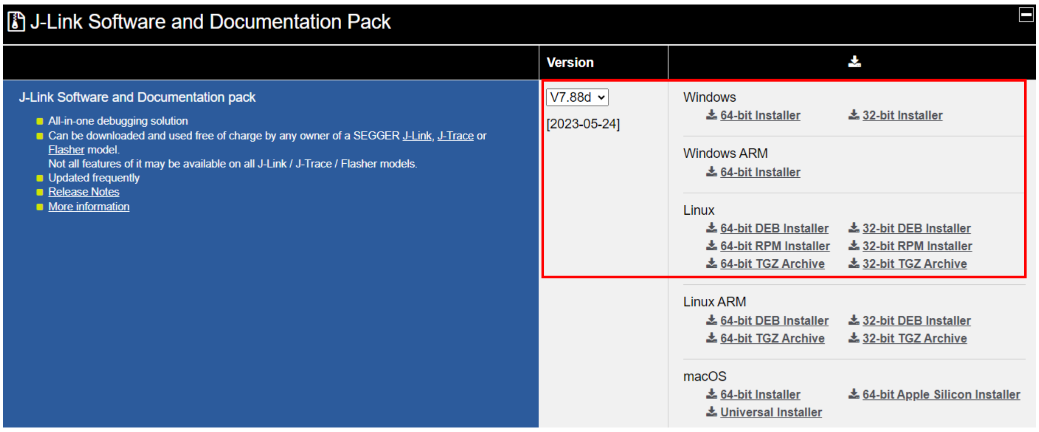
5.6.1 Installing J-Link

To debug the DA16200/DA16600, a J-Link debug probe and the J-Link software is required.

A list of the available debug probes can be found on the Segger website: https://www.segger.com/products/debug-probes/j-link/models/model-overview/

The J-link software can be downloaded from the Segger website: https://www.segger.com/downloads/jlink/

Download and install the version for your specific OS.



J-Link Software and Documentation pack

- All-in-one debugging solution
- Can be downloaded and used free of charge by any owner of a SEGGER J-Link, J-Trace or Flasher model.
  Not all features of it may be available on all J-Link / J-Trace / Flasher models.
- Updated frequently
- Release Notes
- More information

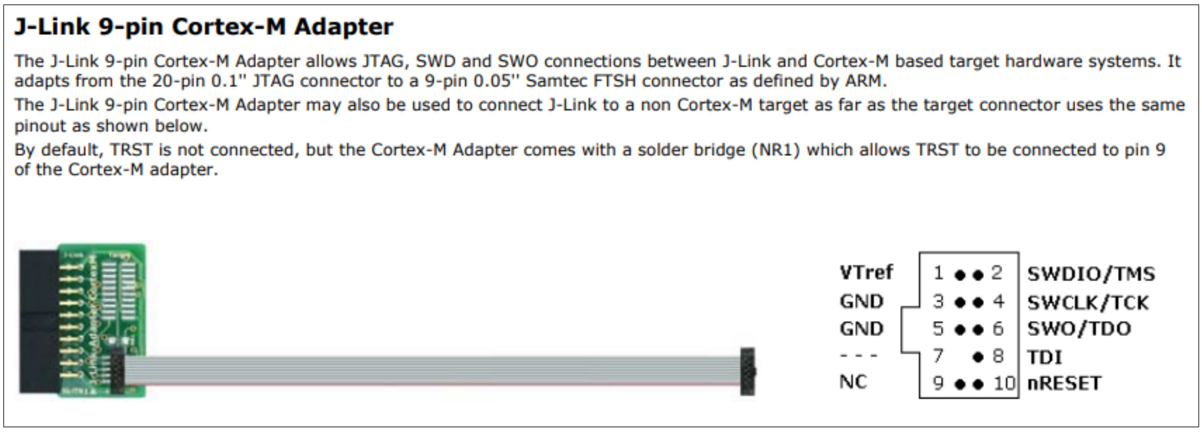
Figure 33: Download J-Link Software

The Windows version can be installed by running the downloaded installer. The Linux version can be installed using the following command:

```
$
$ sudo dpkg -i JLink_Linux_V788d_x86_64.deb
$
```

5.6.2 Connect J-Link

The following section describes how to connect the J-Link debug probe to the DA16200. Connect the 20-pin connector of the “J-Link 9-pin Cortex-M Adapter” to the J-Link debug probe and connect the 9-pin connector to the “JTAG connector” on the DA16200/DA16600 EVB board.



J-Link 9-pin Cortex-M Adapter

The J-Link 9-pin Cortex-M Adapter allows JTAG, SWD and SWO connections between J-Link and Cortex-M based target hardware systems. It adapts from the 20-pin 0.1'' JTAG connector to a 9-pin 0.05'' Samtec FTSH connector as defined by ARM.

The J-Link 9-pin Cortex-M Adapter may also be used to connect J-Link to a non Cortex-M target as far as the target connector uses the same pinout as shown below.

By default, TRST is not connected, but the Cortex-M Adapter comes with a solder bridge (NR1) which allows TRST to be connected to pin 9 of the Cortex-M adapter.

VTref	1 ●	● 2	SWDIO/TMS
GND	3 ●	● 4	SWCLK/TCK
GND	5 ●	● 6	SWO/TDO
- - -	7	● 8	TDI
NC	9 ●	● 10	nRESET

Figure 34: J-Link 9-pin Cortex-M Adapter

5.6.3 Run Debug Mode

To start debugging an application, right-click on the project in the project explorer and select **Debug As** > **Debug Configurations**.

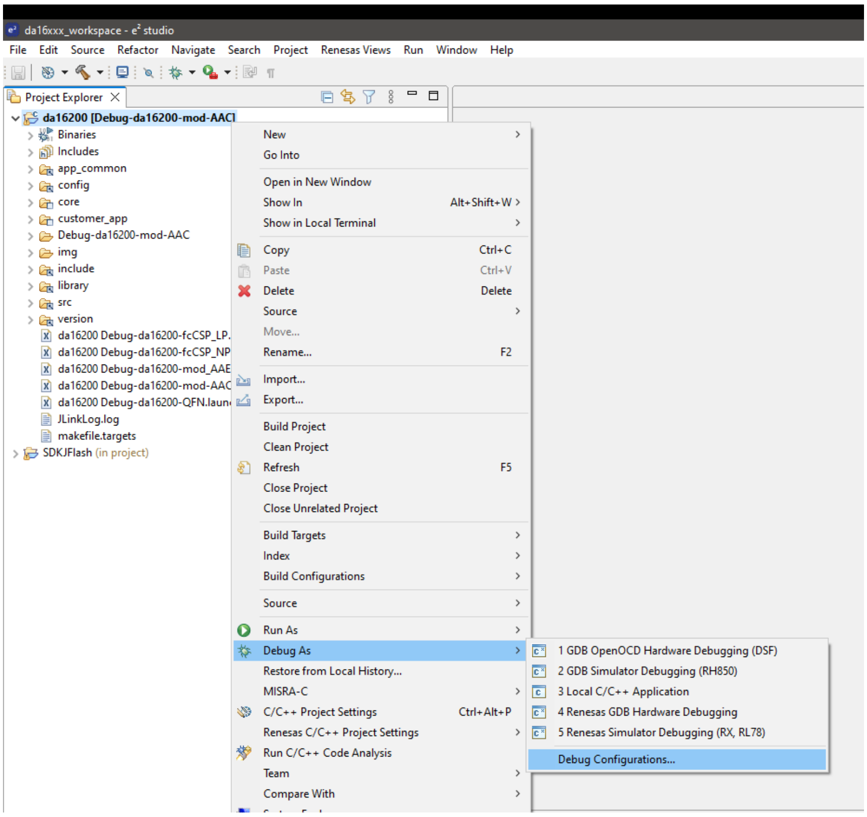
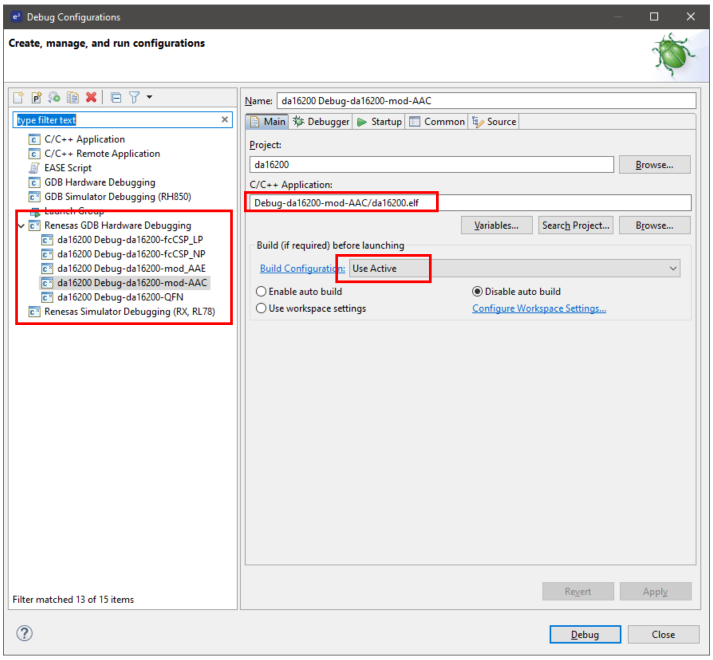

da16xxx_workspace - e² studio
File Edit Source Refactor Navigate Search Project Renesas Views Run Window Help
Project Explorer
da16200 [Debug-da16200-mod-AAC]
Binaries
Includes
app_common
config
core
customer_app
Debug-da16200-mod-AAC
img
include
library
src
version
da16200 Debug-da16200-fcCSP_LP.
da16200 Debug-da16200-fcCSP_NP
da16200 Debug-da16200-mod_AAE
da16200 Debug-da16200-mod-AAC
da16200 Debug-da16200-QFN.laun
JLinkLog.log
makefile.targets
SDKJFlash (in project)
New
Go Into
Open in New Window
Show In Alt+Shift+W
Show in Local Terminal
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Source
Move...
Rename... F2
Import...
Export...
Build Project
Clean Project
Refresh F5
Close Project
Close Unrelated Project
Build Targets
Index
Build Configurations
Source
Run As
Debug As
Restore from Local History...
MISRA-C
C/C++ Project Settings Ctrl+Alt+P
Renesas C/C++ Project Settings
Run C/C++ Code Analysis
Team
Compare With
1 GDB OpenOCD Hardware Debugging (DSF)
2 GDB Simulator Debugging (RH850)
3 Local C/C++ Application
4 Renesas GDB Hardware Debugging
5 Renesas Simulator Debugging (RX, RL78)
Debug Configurations...

Figure 35: Run Debug Mode

Open **the Renesas GDB Hardware Debugging** entry in the list and select one of debugging configurations and then click **Debug**.


Debug Configurations
Create, manage, and run configurations
type filter text
C/C++ Application
C/C++ Remote Application
EASE Script
GDB Hardware Debugging
GDB Simulator Debugging (RH850)
Launch Group
Renesas GDB Hardware Debugging
da16200 Debug-da16200-fcCSP_LP
da16200 Debug-da16200-fcCSP_NP
da16200 Debug-da16200-mod_AAE
da16200 Debug-da16200-mod-AAC
da16200 Debug-da16200-QFN
Renesas Simulator Debugging (RX, RL78)
Filter matched 13 of 15 items
Main
Debugger
Startup
Common
Source
Variables...
Search Project...
Browse...
Build (if required) before launching
Configure Workspace Settings...
Revert
Apply
Debug
Close

Name: da16200 Debug-da16200-mod-AAC

Project: da16200

C/C++ Application: Debug-da16200-mod-AAC/da16200.elf

Build Configuration: Use Active

○ Enable auto build ● Disable auto build

○ Use workspace settings

Figure 36: Select Debug Configuration

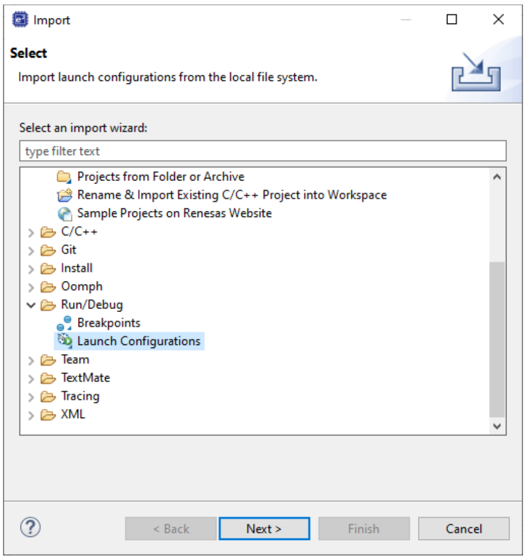
NOTE
The current FreeRTOS SDK does not support automatic downloading of the firmware image into flash through the e^{2}studio debug interface therefore the firmware must be loaded into SFLASH before starting to debug the application.

5.7 Programming Firmware Images in e^{2}studio

5.7.1 Import Launches for Programming Firmware Images

Firmware images can be programmed using launches in e^{2}studio. The launches can be imported as follows.

1. Click **File > Import** and Select **Run/Debug > Launch Configurations** as follows.



Import

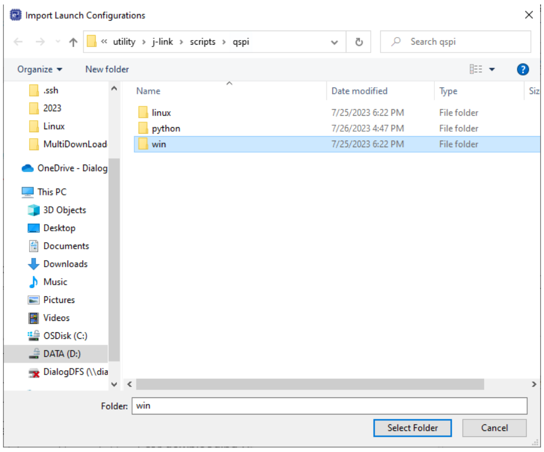
Select

Import launch configurations from the local file system.

Select an import wizard:

Figure 37: Import Launches

2. Click **Next**, browse `<sdk_root_directory>\utility\j-link\scripts\qspi` and select folder win, linux, or python in `<sdk_root_directory>\ulitil\j-link\scripts\qspi` according to environment.



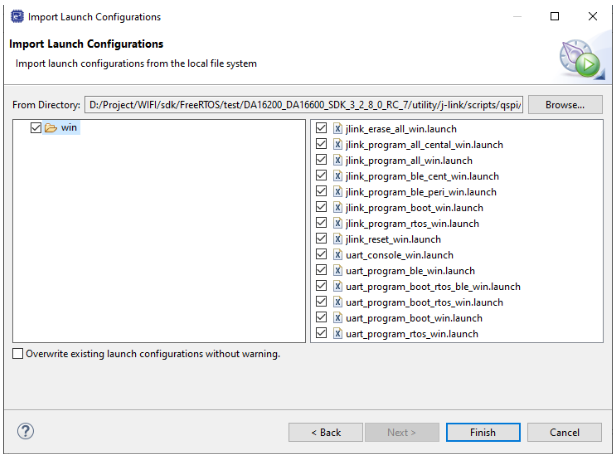
Import Launch Configurations

Name	Date modified	Type	Siz
linux	7/25/2023 6:22 PM	File folder	
python	7/26/2023 4:47 PM	File folder	
win	7/25/2023 6:22 PM	File folder	

Folder: win

Figure 38: Browse Folder

3. Click **Select Folder** and Select launch configurations.



Import Launch Configurations

Import Launch Configurations

Import launch configurations from the local file system

From Directory: D:/Project/WIFI/sdk/FreeRTOS/test/DA16200_DA16600_SDK_3_2_8_0_RC_7/utility/j-link/scripts/qspi/

☑ win

☑ jlink_erase_all_win.launch
☑ jlink_program_all_cental_win.launch
☑ jlink_program_all_win.launch
☑ jlink_program_ble_cent_win.launch
☑ jlink_program_ble_peri_win.launch
☑ jlink_program_boot_win.launch
☑ jlink_program_rtos_win.launch
☑ jlink_reset_win.launch
☑ uart_console_win.launch
☑ uart_program_ble_win.launch
☑ uart_program_boot_rtos_ble_win.launch
☑ uart_program_boot_rtos_win.launch
☑ uart_program_boot_win.launch
☑ uart_program_rtos_win.launch

□ Overwrite existing launch configurations without warning.

Figure 39: Select Launch Configurations

4. Click **Finish** and Confirm launch configurations in **Run > External Tools** menu.

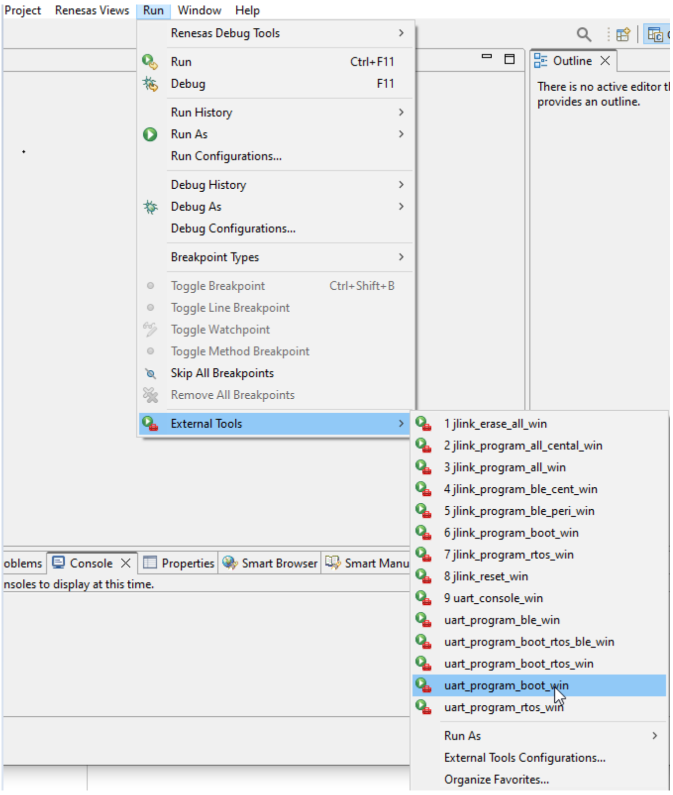
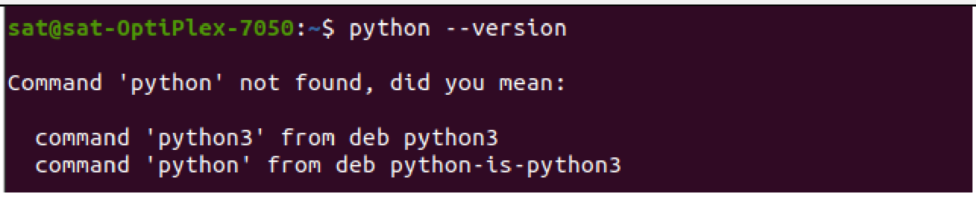


Figure 40: Launch Configurations for Programming Firmware Images

NOTE
SDKJFlash project must be imported before programming. (See Section 5.4 and Figure 20) ● **Python 3.8** must be installed to use jlink_xxx_py scripts. ● **For Linux OS** ○ Check python version in the terminal using the command as below. – python --version ○ If you get a **Command 'python' not found** error as shown below, you need to make a symbolic link for python command.

NOTE



```
sat@sat-OptiPlex-7050:~$ python --version

Command 'python' not found, did you mean:

  command 'python3' from deb python3
  command 'python' from deb python-is-python3
```

- ○ You can make a symbolic link for python command as follows.
  - – sudo ln -s /usr/bin/python3 /usr/bin/python
- ○ If access to serial interface is denied, permission should be changed as follows.
  - – sudo usermod -a -G dialout <username>

5.7.2 Programming Firmware Images over Serial Interface

The firmware image can be programmed over serial interface. The requirements are below.

1. e^2studio 2023-07 (23.7.0) or later.
2. Python 3.8 and pyserial package (optional)

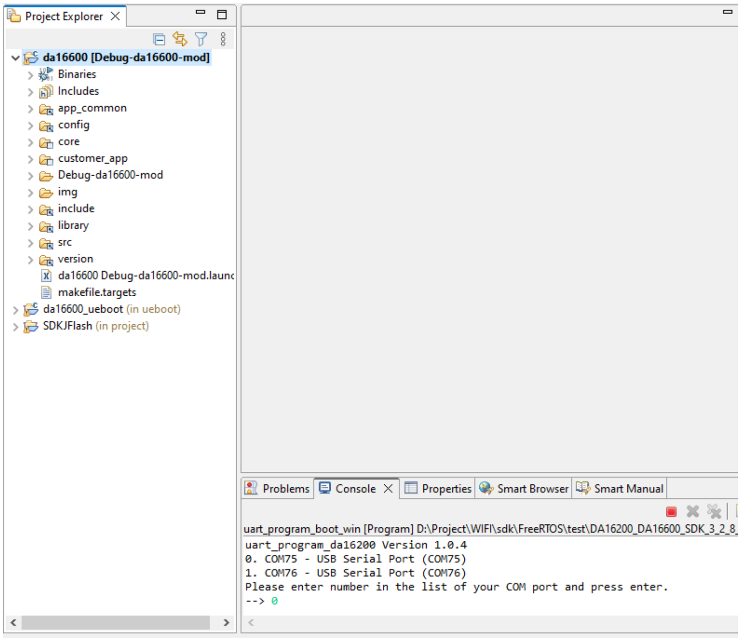
The following launches are included in the SDK.

- uart_program_boot_win(linux/py) : Program FBOOT image
- uart_program_rtos_win(linux/py) : Program FRTOS image
- uart_program_boot_rtos_win(linux/py) : Program FBOOT and FRTOS image
- uart_program_ble_win(linux/py) : Program BLE firmware image (only DA16600)
- uart_program_boot_rtos_ble_win(linux/py) : Program FBOOT, FRTOS and DA14531 firmware image (only DA16600)

The programming can be done as follows.

1. Select the **Project** and select proper **Launch** in **Run > External Tools** menu, then the image programming is invoked.

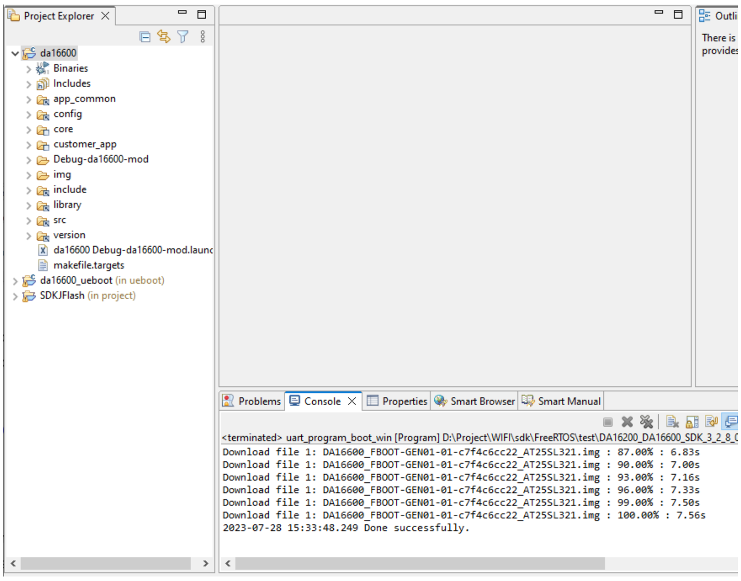
2. Select proper **number** in the list of serial interfaces at console terminal.



```
uart_program_boot_win [Program] D:\Project\WIFI\sdk\FreeRTOS\test\DA16200_DA16600_SDK_3_2_8_
uart_program_da16200 Version 1.0.4
0. COM75 - USB Serial Port (COM75)
1. COM76 - USB Serial Port (COM76)
Please enter number in the list of your COM port and press enter.
--> 0
```

Figure 41: Select Number from Serial Interface List

3. The programming is done automatically and confirm it is done successfully.

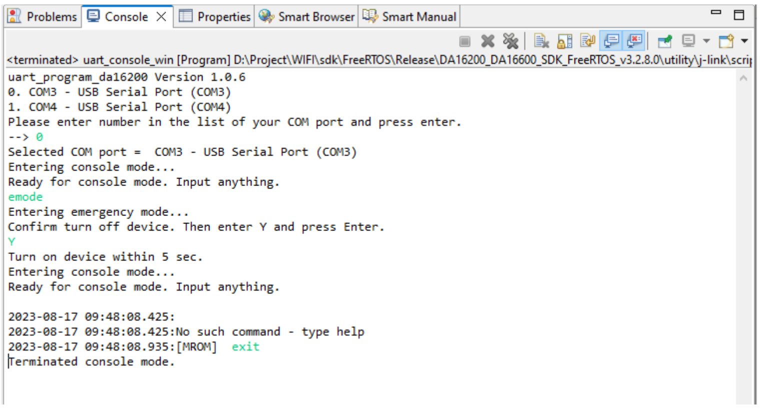


```
<terminated> uart_program_boot_win [Program] D:\Project\WIFI\sdk\FreeRTOS\test\DA16200_DA16600_SDK_3_2_8_(
Download file 1: DA16600_FBOOT-GEN01-01-c7f4c6cc22_AT25SL321.img : 87.00% : 6.83s
Download file 1: DA16600_FBOOT-GEN01-01-c7f4c6cc22_AT25SL321.img : 90.00% : 7.00s
Download file 1: DA16600_FBOOT-GEN01-01-c7f4c6cc22_AT25SL321.img : 93.00% : 7.16s
Download file 1: DA16600_FBOOT-GEN01-01-c7f4c6cc22_AT25SL321.img : 96.00% : 7.33s
Download file 1: DA16600_FBOOT-GEN01-01-c7f4c6cc22_AT25SL321.img : 99.00% : 7.50s
Download file 1: DA16600_FBOOT-GEN01-01-c7f4c6cc22_AT25SL321.img : 100.00% : 7.56s
2023-07-28 15:33:48.249 Done successfully.
```

Figure 42: Programming Firmware Image over Serial Interface

The console interface is disabled at the firmware image built in release build configuration or at DPM low power mode. In these cases, further programming the firmware image can be done as follows.

1. Programming firmware image using SEGGER J-Link as shown in Section 5.7.3.
2. Programing firmware image over serial interface as shown in Figure 43.
    a. Turn off device
    b. Run uart_console_win(linux, py) launch in **Run > External Tools** menu
    c. Select port number and enter `emode` for running emergency mode
    d. Follow the instructions in the Console and confirm the device enter MROM state
    e. Exit the uart_console_win(linux, py) by entering `Exit` or clicking **Terminate** icon
    f. Run uart_program_xxx launch for programming firmware image



```
<terminated> uart_console_win [Program] D:\Project\WIFI\sdk\FreeRTOS\Release\DA16200_DA16600_SDK_FreeRTOS_v3.2.8.0\utility\j-link\scri
uart_program_da16200 Version 1.0.6
0. COM3 - USB Serial Port (COM3)
1. COM4 - USB Serial Port (COM4)
Please enter number in the list of your COM port and press enter.
--> 0
Selected COM port =  COM3 - USB Serial Port (COM3)
Entering console mode...
Ready for console mode. Input anything.
emode
Entering emergency mode...
Confirm turn off device. Then enter Y and press Enter.
Y
Turn on device within 5 sec.
Entering console mode...
Ready for console mode. Input anything.

2023-08-17 09:48:08.425:
2023-08-17 09:48:08.425:No such command - type help
2023-08-17 09:48:08.935:[MROM]  exit
Terminated console mode.
```

Figure 43: Emergency Mode for Programming Firmware Image

NOTE
If uart_program_xxx runs without exit of uart_console_win(linux, py), the open of serial interface would be failed. In this case, the uart_console_win(linux, py) should be terminated by clicking **Remove all Terminated Launches** and **Terminate** icon in Console terminal.

* See Appendix D.1.2.

5.7.3 Programming Firmware Images Using SEGGER J-Link

The firmware image can be programmed using Segger J-Link also. The requirements are below.

1. J-Link Lite or higher.
   See: https://www.segger.com/products/debug-probes/j-link/models/model-overview/
2. e^{2}studio 2023-07 (23.7.0) or later.
3. Python 3.8 (optional).

The J-Link setup procedure required to work with J-Flash is described in chapter 2 of the J-Link / J-Trace User Guide (UM08001).The following scripts are included in the SDK:

- jlink_erase_all_win(linux/py) : Erase all area of the flash
- jlink_program_all_win(linux/py): Program all images
- jlink_program_boot_win(linux/py): Program FBOOT image
- jlink_program_rtos_win((linux/py): Program FRTOS image
- jlink_program_all_central_win(linux/py): Program all images for BLE central role.(For DA16600)
- jlink_program_ble_peri_win(linux/py): Program BLE image for a peripheral.(For DA16600)
  jlink_program_ble_cent_win(linux/py): Program BLE image for a central role.(For DA16600)

The programming can be done by selecting the **project** and proper **launch** in **Run > External Tools** menu as shown in Figure 44, then the programming is invoked automatically.

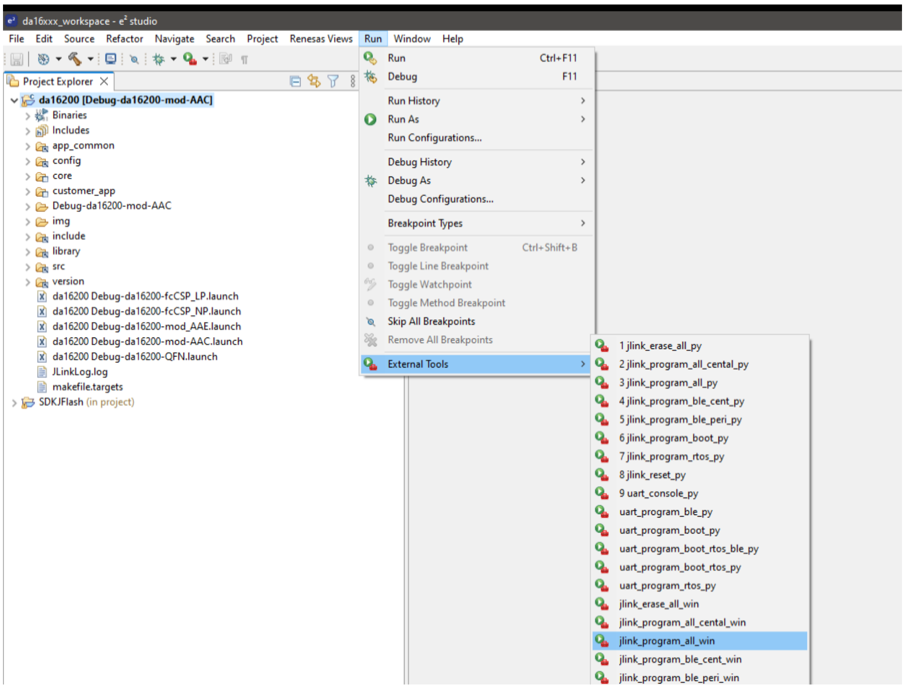
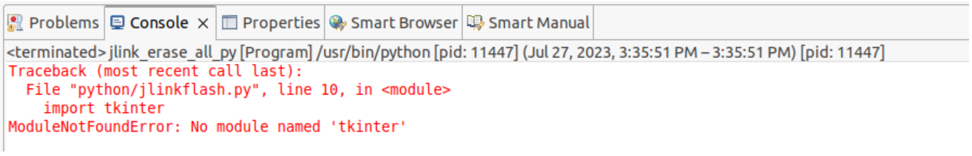


Figure 44: Run J-Link Launch

NOTE

- **For Linux OS**
  - If you get the following error when running jlink_xxx_py, you need to install the **tkinter** module for python



```
<terminated> jlink_erase_all_py [Program] /usr/bin/python [pid: 11447] (Jul 27, 2023, 3:35:51 PM – 3:35:51 PM) [pid: 11447]
Traceback (most recent call last):
  File "python/jlinkflash.py", line 10, in <module>
    import tkinter
ModuleNotFoundError: No module named 'tkinter'
```

  - You can install **tkinter** module using command in terminal as below.

NOTE
– sudo apt-get install python3-tk

Other methods for programming the firmware images are described in Section 4.5.

5.8 How to Migrate IDE from Eclipse to e^{2}studio for Old SDK

The old SDK (before SDK 3.2.8) can be used in the e^{2}studio also. The migration can be done as follows.

1. Remove `.project` in root folder of the old SDK.
2. Remove all folders and files in `<sdk_root_directory>\utility\j-link\scripts\qspi\`.
3. Copy all folders and files in `<sdk_root_directory>\utility\j-link\scripts\qspi\` of SDK 3.2.8 (or later) to `<sdk_root_directory>\utility\j-link\scripts\qspi\` of old SDK.
4. Follow installing e2studio/creating workspace/import projects/import launches as described in previous sections.

6 Test Procedures

6.1 Introduction

The following sections describe several tests that will verify the proper operation of the DA16200/DA16600 and demonstrates its various features.

The tests include:

- Ping Test – Verifies the connection between the DA16200/DA16600 and another device connected to the same AP
- Throughput Test – Demonstrates the Wi-Fi performance of the DA16200/DA16600
- DPM Test – Demonstrates the various power modes of the DA16200/DA16600
- Current Test – Demonstrates the amount of power used when the DA16200/DA16600 is in various sleep modes

6.2 Ping Test

Ping is a standard application that is used to verify if devices exist on a network. This test procedure will demonstrate that the DA16200/DA16600 will respond to ping commands while in DPM mode.

6.2.1 Test Setup

The ping communication test requires an access point (AP) and two stations consisting of a DA16200 or DA16600 EVB and a desktop. The two stations must be connected to the same sub-network AP as shown in Figure 45. The DA16200 must be connected to the AP via Wi-Fi, and the laptop must be connected to the AP with an Ethernet cable.

After the DA16200/DA16600 is configured, it will go into DPM low power mode and will only wake up from the low power mode when unicast packets are received.

In this test, a ping application which runs on the laptop acts as a network peer that sends a unicast packet to the DA16200/DA16600. This shows that when a DA16200/DA16600 is in DPM low power mode, it can successfully wake up and receive the unicast packets in real-time.

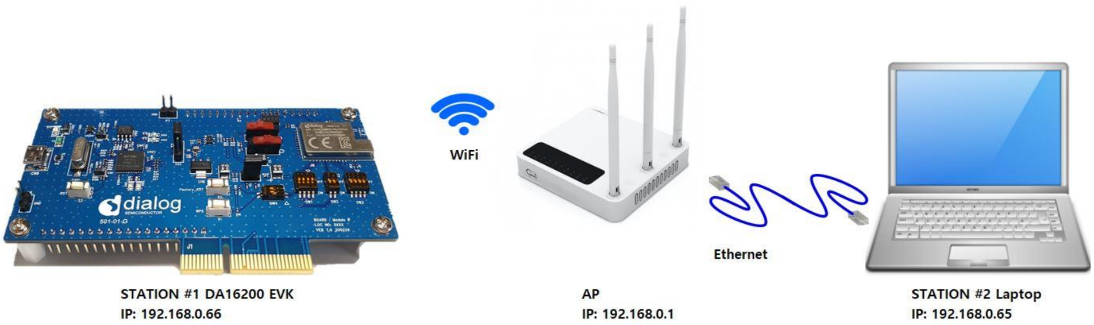


Figure 45: Ping Test Environment

1. On the laptop, open a Command Prompt as administrator.
2. Run the `ipconfig` command to list the IP address of the network adaptor:

```
C:\user\testuser>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet 4:

Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::4c8:3627:424b:1951%20
   IPv4 Address. . . . . . . . . . . : 192.168.0.65
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1
```

In this case, the IP Address is `192.168.0.65` and the Default Gateway is `192.168.0.1`.

3. Open a serial terminal and connect to the DA16200 serial debug port.

   Configure the DA16200/DA16600 to operate in Station Mode as described in Section 4.6.1. Once the DA16200/DA16600 is configured, reboot and check the output to get the assigned IP address:

```
Connection COMPLETE to 80:ca:4b:30:02:0a

-- DHCP Client WLAN0: SEL(6)
-- DHCP Client WLAN0: REQ(1)
-- DHCP Client WLAN0: CHK(8)
-- DHCP Client WLAN0: BOUND(10)
         Assigned addr   : 192.168.0.66
               netmask   : 255.255.255.0
               gateway   : 192.168.0.1
               DNS addr  : 61.41.153.2

         DHCP Server IP  : 192.168.0.1
         Lease Time      : 06h 00m 00s
         Renewal Time    : 03h 00m 00s
```

   In this case, the assigned IP for the DA16200/DA16600 is `192.168.0.66`.

4. Verify the setup.

   Using the IP address of the DA16200/DA16600 run a ping command on the laptop as follows:

```
C:\user\testuser>ping 192.168.0.66


Pinging 192.168.0.66 with 32 bytes of data:
Reply from 192.168.0.66: bytes=32 time=81ms TTL=255
Reply from 192.168.0.66: bytes=32 time=14ms TTL=255
Reply from 192.168.0.66: bytes=32 time=18ms TTL=255
Reply from 192.168.0.66: bytes=32 time=25ms TTL=255

Ping statistics for 192.168.0.66:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 14ms, Maximum = 81ms, Average = 34ms
```

6.2.2 Perform Ping Test with DPM Enabled

Once the environment is setup and simple ping is working, perform the ping test with DPM enabled as follows:

1. Enable DPM as described in Section 6.4.2.

   After DPM is enabled, the DA16200/DA16600 will go into DPM low power mode.

2. Run the ping command on the laptop.

Using the IP address of the DA16200/DA16600 run a ping command on the laptop as follows:

```
C:\user\testuser>ping 192.168.0.66


Pinging 192.168.0.66 with 32 bytes of data:
Reply from 192.168.0.66: bytes=32 time=81ms TTL=255
Reply from 192.168.0.66: bytes=32 time=14ms TTL=255
Reply from 192.168.0.66: bytes=32 time=18ms TTL=255
Reply from 192.168.0.66: bytes=32 time=25ms TTL=255

Ping statistics for 192.168.0.66:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 14ms, Maximum = 81ms, Average = 34ms
```

For each ping packet sent, the DA16200/DA16600 will wake up and print a message on the serial terminal and then return to the low power mode:

```
Wakeup source is 0x82

>>> TIM STATUS: 0x00000001
>>> TIM : UC

>>> Start DPM Power-Down !!!
rwnx_send_set_ps_mode PS TIME (us) 139351

Wakeup source is 0x82

>>> TIM STATUS: 0x00000001
>>> TIM : UC

>>> Start DPM Power-Down !!!
rwnx_send_set_ps_mode PS TIME (us) 179263

Wakeup source is 0x82

>>> TIM STATUS: 0x00000001
>>> TIM : UC

>>> Start DPM Power-Down !!!
rwnx_send_set_ps_mode PS TIME (us) 129354
```

To disable or change the DPM settings, see Section 6.4.3.

6.2.3 How to Add ARP Record

This section describes how to add a DHCP assigned IP address to the ARP table and change that IP address from a dynamic to a static IP address.

If the DA16200/DA16600 is in DPM low power mode and there are multiple network interfaces enabled on the laptop, an ARP entry must be added for the specific interface so that the laptop knows how to find the DA16200/DA16600.

Since retransmission logic is not included in the higher protocol (TCP / UDP), an additional ARP record is required for ping tests between the laptop and the DA16200/DA16600 which is operating in DPM low power mode.

1. Use the `arp -s` command to manually add an ARP record to the ARP cache and then use the `arp -a` command to view the ARP table for the network interface.

```
C:\user\testuser>arp -s 192.168.0.66 d4-3d-39-10-a2-48
C:\user\testuser>arp -a

Interface: 192.168.0.105 --- 0x12
  Internet Address      Physical Address      Type
  192.168.0.1           80-ca-4b-30-02-09     dynamic
  192.168.0.66          d4-3d-39-10-a2-48     dynamic
  192.168.0.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

The DA16200's IP address 192.168.0.66 is now added to the ARP table as a dynamic type.

NOTE

When setting up the ARP cache to static with the command `arp -s` on higher versions of Windows, an error like `Failed to add ARP entry, Access is denied.` will be occurred. Renesas recommends using command `netsh` to change the network settings.

2. Use the command `netsh interface show interface` to find the interface name.

```
C:\user\testuser>netsh interface show interface

Admin State    State          Type             Interface Name
-------------------------------------------------------------------------
Enabled        Connected      Dedicated        Ethernet
Enabled        Disconnected   Dedicated        Wi-Fi
```

3. Use the interface name found for DA16200 to set the ARP cache to static with the `netsh` command and then use `arp -a` to verify that it is now a static ARP entry.

```
C:\user\testuser>netsh interface ip add neighbors "Ethernet" "192.168.0.66" "d4-3d-39-10-a2-48"

C:\WINDOWS\system32>arp -a

Interface: 192.168.0.105 --- 0x12
  Internet Address      Physical Address      Type
  192.168.0.1           80-ca-4b-30-02-09     dynamic
  192.168.0.66          d4-3d-39-10-a2-48     static
  192.168.0.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

The DA16200's IP address 192.168.0.66 is now added to ARP table as a static type.

4. Use the command `arp -d` or `netsh interface ip delete arpcache` to initialize the ARP cache.

6.3 Throughput Test

This section describes how to perform the throughput test using the TCP client/server protocol.

DA16200 has a command `iperf` for measuring throughput performance. To do the throughput test, prepare the DA16200 to operate in Station Mode as described in Section 4.6.1 and disable DPM as described in Section 6.4.3.

6.3.1 Test Setup

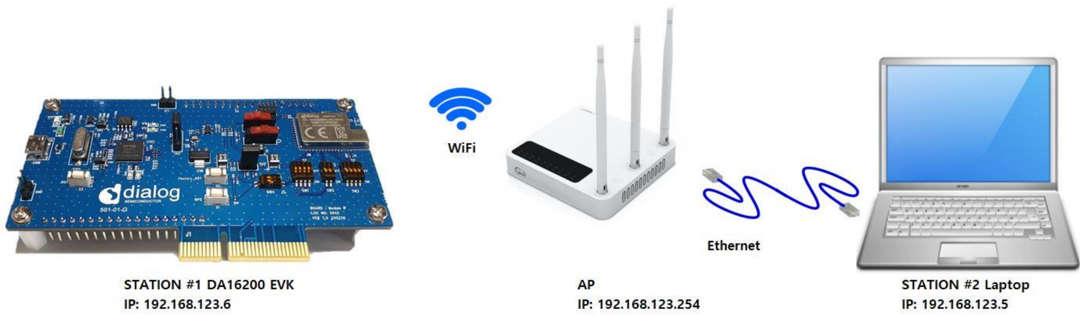


Figure 46: Throughput Test Environment

The Iperf tool should be installed on your laptop. **Iperf Version 2.0.9** is recommended.

NOTE
Iperf 3.x is not supported.

To setup Iperf tool, follow the steps below:

1. Download Iperf from https://iperf.fr/iperf-download.php.
2. Create a folder called **Iperf** in path **C:\**.

3. Unzip the downloaded file and move the contents to the Iperf folder.
4. Prepare the DA16200 to operate in Station Mode as described in Section 4.6.1.
5. Use command `iperf` or `iperf -h` to see the available options in `Iperf`.

Example:

```
[/DA16200] # net                                         Switch to network command mode.
    Command-List is changed, "NET"
[/DA16200/NET] # iperf                                   Run iperf command.
Usage:  iperf -I [WLAN0|WLAN1] [-s|-c host][options]
        iperf [-h] [-v]

Client/Server:
        -I      Interface [WLAN0|WLAN1]
        -i      seconds between periodic bandwidth reports
        -u      use UDP rather than TCP
        -p, #   server port to listen on/connect to
        -f, [kmKM]   format to report: Kbits, Mbits, KBytes, MBytes
        -d      finsh service
                ex) iperf -d -c -u : udp clinet
                    iperf -d -c    : tcp clinet
                    iperf -d -u    : udp server
                    iperf -d       : tcp server

Server specific:
        -s      run in server mode
        -T  #   Rx Time Out Min:1 sec. 'F' Forever

Client specific:
        -c      <host>   run in client mode, connecting to <host>
        -t  #   time in seconds to transmit for (default 10 secs)
        -x  #   tcp API mode default:basic tcp(API) 1:Altcp 2:Socket
        -y  #   Transmit delay, tick 1 ~ 100
        -l  #   PacketSize option (UDP default 1470, IPv6 1448 TCP 1000)
        -n  #   UDP Tx packet number
        -P, #   Pair Index (0,1,2)
                (default Max, Step 1~100 Mbps)
        -O      use Main Packet Pool

Miscellaneous:
        -h      print this message
        -v      print version
```

6.3.2 Throughput Test with DA16200/DA16600 as Client

To set-up the throughput test with the DA16200/DA16600 operating as a client and the laptop operating as a server, follow the steps below:

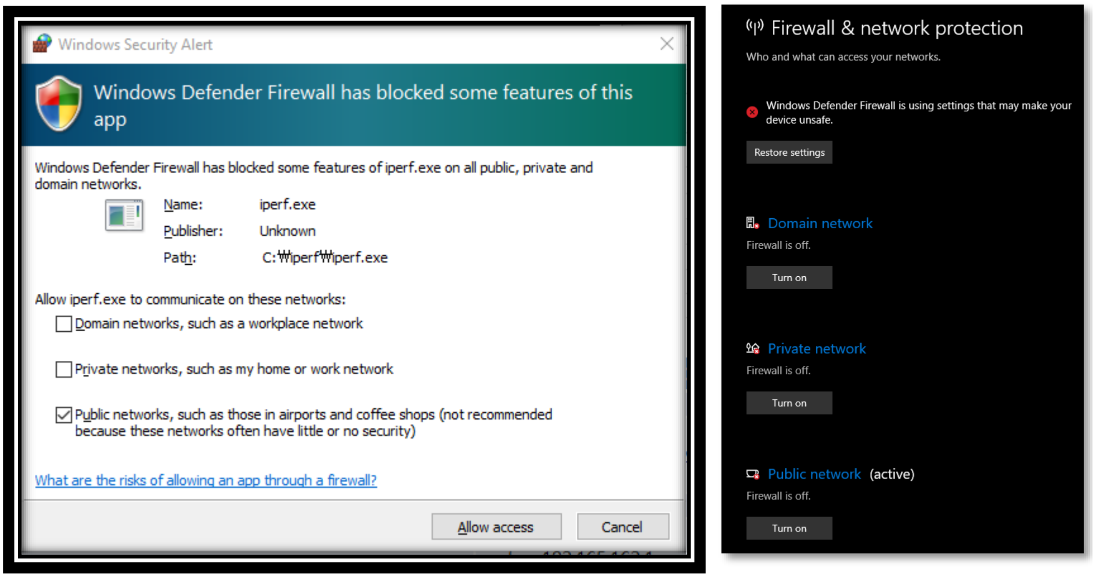
1. Connect the laptop which is the server to the AP.
2. In the CMD window, use the command `ipconfig /all` to find the IP address:

```
C:\user\testuser>ipconfig /all

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : Davolink
   Description . . . . . . . . . . . : Realtek USB GbE Family Controller #3
   Physical Address. . . . . . . . . : 50-E0-85-D7-1E-4B
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::b1e7:a692:a700:748a%13(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.0.103(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Monday, September 27, 2021 4:23:16 PM
   Lease Expires . . . . . . . . . . : Tuesday, September 28, 2021 4:24:33 PM
   Default Gateway . . . . . . . . . : 192.168.0.1
   DHCP Server . . . . . . . . . . . : 192.168.0.1
   DHCPv6 IAID . . . . . . . . . . . : 173072517
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-28-2E-4F-CA-98-E7-43-AD-7D-81
   DNS Servers . . . . . . . . . . . : 1.214.68.2
                                       61.41.153.2
```

```
    NetBIOS over Tcpip. . . . . . . . : Enabled
```

NOTE
The IP address is different depending on the AP settings.

3. To ensure stable Iperf testing, run the Windows Security APP and turn off the network firewall.
   ○ It is recommended to disable the laptop from all network firewalls before attempting a test as Figure 47.



Windows Security Alert

Windows Defender Firewall has blocked some features of this app

Windows Defender Firewall has blocked some features of iperf.exe on all public, private and domain networks.

Name: iperf.exe
Publisher: Unknown
Path: C:₩iperf₩iperf.exe

Allow iperf.exe to communicate on these networks:

□ Domain networks, such as a workplace network

□ Private networks, such as my home or work network

☑ Public networks, such as those in airports and coffee shops (not recommended because these networks often have little or no security)

What are the risks of allowing an app through a firewall?

Firewall & network protection

Who and what can access your networks.

Windows Defender Firewall is using settings that may make your device unsafe.

Domain network
Firewall is off.

Private network
Firewall is off.

Public network (active)
Firewall is off.

Figure 47: Disable Firewall for Iperf Test

4. In the CMD window, go to the directory where Iperf is installed, and type `iperf -s` to configure the TCP server:

```
c:\tools\iperf>iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size:  208 KByte (default)
------------------------------------------------------------
```

   Iperf is now running as a server on the laptop and waiting for the DA16200/DA16600 to start the client test app.

5. In the DA16200/DA16600 console window switch to “NET” command mode and run “`iperf -I wlan0 -c 192.168.0.103 -t 5 -i 1`” to run the Iperf client test on the DA16200/DA16600:

```
[/DA16200] # net                                          Switch to network command mode.
    Command-List is changed, "NET"

[/DA16200/NET] # iperf -I wlan0 -c 192.168.219.103 -t 5 -i 1     Run the Iperf client test.
[/DA16200/NET] #

[TCP] Tx Test  (Client)  ==> 192.168.219.103:5001
TCP_TX: [ No ]       [Interval]    [Transfer]        [Bandwidth]             [Dst IP:Port]
TCP_TX: [0001]       0.00- 1.00    2.070 MBytes     17.368 Mbits/sec 192.168.0.103:5001
TCP_TX: [0002]       1.00- 2.00    2.156 MBytes     18.092 Mbits/sec 192.168.0.103:5001
TCP_TX: [0003]       2.00- 3.00    2.142 MBytes     17.975 Mbits/sec 192.168.0.103:5001
TCP_TX: [0004]       3.00- 4.00    2.192 MBytes     18.396 Mbits/sec 192.168.0.103:5001
TCP_TX: [Total]      0.00- 5.00   10.801 MBytes     18.122 Mbits/sec 192.168.0.103:5001


[/DA16200/NET] #
```

   Where the format of the iperf command type is:

```
iperf -I [INTERFACE] [-s/-c] [DESTINATION IP] (-u) -i [INTERVAL TIME] -t [TEST TIME]
```

6.3.3 Throughput Test with DA16200/DA16600 as Server

To set-up the throughput test with the DA16200/DA16600 operating as a server and the laptop operating as a client, follow the steps below:

1. Power on the DA16200/DA16600 and check the assigned IP address in the DA16200/DA16600 console window:

```
Connection COMPLETE to 80:ca:4b:30:02:0a

-- DHCP Client WLAN0: SEL(6)
-- DHCP Client WLAN0: REQ(1)
-- DHCP Client WLAN0: CHK(8)
-- DHCP Client WLAN0: BOUND(10)
        Assigned addr   : 192.168.0.66
             netmask    : 255.255.255.0
             gateway    : 192.168.0.254
             DNS addr   : 61.41.153.2

        DHCP Server IP  : 192.168.0.254
        Lease Time      : 06h 00m 00s
        Renewal Time    : 03h 00m 00s


[/DA16200] #
```

2. In the DA16200/DA16600 console window switch to “NET” command mode and run “`iperf -I wlan0 -s`” to run the Iperf server:

```
[/DA16200] # net                                            Switch to network command mode.
   Command-List is changed, "NET"
[/DA16200/NET] # iperf -I wlan0 -s                          Run Iperf in server mode.

iPerf Server(TCP): Ready
[/DA16200/NET] #
```

   The DA16200/DA16600 is now running in iperf server mode.

3. In the CMD window, run “`iperf -I wlan0 -c 192.168.0.103 -t 5 -i 1`” to run the Iperf client test on the laptop:

```
C:\tools\iperf>iperf.exe -c 192.168.0.66 -t 5 -i 1          Run the Iperf client test.
------------------------------------------------------------
Client connecting to 192.168.0.66, TCP port 5001
TCP window size:  208 KByte (default)
------------------------------------------------------------
[  3] local 192.168.219.105 port 53916 connected with 192.168.0.66 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 1.0 sec  2.62 MBytes  22.0 Mbits/sec
[  3]  1.0- 2.0 sec  2.50 MBytes  21.0 Mbits/sec
[  3]  2.0- 3.0 sec  2.38 MBytes  19.9 Mbits/sec
[  3]  3.0- 4.0 sec  2.38 MBytes  19.9 Mbits/sec
[  3]  4.0- 5.0 sec  2.38 MBytes  19.9 Mbits/sec
[  3]  0.0- 5.0 sec  12.2 MBytes  20.5 Mbits/sec

C:\tools\iperf>
```

The following results will appear in the DA16200/DA16600 command window:

```
[TCP] Receive Test (Server)
TCP_RX:[ No ]      [Interval]    [Transfer]       [Bandwidth]
TCP_RX:[Total]     0.00- 5.13    12.250 MBytes    20.031 Mbits/sec 192.168.0.103:53916
```

6.4 DPM Setup

6.4.1 What is DPM

Dynamic Power Management (DPM) is a synthesis of breakthrough in ultra-low power technologies that enable extremely low power operation in the DA16200. DPM shuts down all microelements on the chip when not in use, which allows a near zero level of power consumption when not actively transmitting or receiving data. Such low-power consumption can provide a battery life of one year or more, depending on the application. DPM also enables ultra-low power transmit/receive modes when the SoC needs to be awake to exchange information with other devices. Advanced algorithms enable to stay asleep until the exact required moment to wake up to transmit or receive.

6.4.2 Enable DPM Mode

This section describes how to enable DPM mode.

1. Follow the steps in Section 4.6 to setup the DA16200/DA16600 in station mode.
   a. During the setup, enable DPM as follows:

```
Dialog DPM (Dynamic Power Management) ? [Yes/No/Quit] : y      Enter "y" to enable DPM.

 DPM factors : Defaults ? [Yes/No/Quit] : y                     Enter "y" to accept defaults.

==================================
DPM MODE           : Enable
Dynamic Period Set : Disable
Keep Alive Time    : 30000 ms
User Wakeup Time   : 0 sec.
TIM Wakeup Count   : 10 dtim
==================================
 DPM CONFIGURATION CONFIRM ? [Yes/No/Quit] : y                  Enter "y" to confirm the configuration.

Configuration OK
. done
```

NOTE
The default TIM wakeup count is 10 dtim: 10 dtim*102.4 = 1,024 ms = 1 sec @ DTIM = 1 (Assumed that AP DTIM = 3, 10 dtim is 921.6 ms) Wakeup from sleep state takes place every second to check if a packet has been received.

   b. Custom DPM factors can be defined during DPM setup as follows:

```
 Dialog DPM (Dynamic Power Management) ? [Yes/No/Quit] : y          Enter "y" to enable DPM.

 DPM factors : Defaults ? [Yes/No/Quit] : n             Enter "n" to provide custom DPM factors.

 DDPS Enable ? [No/Yes/Quit] (Default: No) :            Enter to accept the default of DDPS disabled.

 DPM Keep Alive Time(0~600000 ms) ? [Quit] (Default 30000 ms) :
                                                        Custom value or "enter" for default.

 DPM User Wakeup Time(0~86400 Sec.) ? [Quit] (Default 0 Sec.) :
                                                        Custom value or "enter" for default.

 DPM TIM Wakeup Count(1~65535 dtim) ? [Quit] (Default 10) :   Custom value or "enter" for default.
==================================
DPM MODE           : Enable
Dynamic Period Set : Disable
Keep Alive Time    : 30000 ms
User Wakeup Time   : 0 sec.
TIM Wakeup Count   : 10 dtim
==================================
 DPM CONFIGURATION CONFIRM ? [Yes/No/Quit] : y                  Enter "y" to confirm DPM factors.
```

```
Configuration OK
. done
```

2. After reboot, the DA16200/DA16600 will enter DPM low power mode. The following message is displayed:

```
Connection COMPLETE to 80:ca:4b:30:02:0a

-- DHCP Client WLAN0: SEL(6)
-- DHCP Client WLAN0: REQ(1)
-- DHCP Client WLAN0: CHK(8)
-- DHCP Client WLAN0: BOUND(10)
        Assigned addr   : 192.168.219.105
             netmask    : 255.255.255.0
             gateway    : 192.168.219.1
             DNS addr   : 61.41.153.2

        DHCP Server IP  : 192.168.219.1
        Lease Time      : 06h 00m 00s
        Renewal Time    : 05h 00m 00s
```

NOTE
When the DA16200/DA16600 is in DPM low power mode, user input is not accepted by the debug terminal. To reenable user input, see Section 6.4.3.

6.4.3 Hold/Disable DPM Mode

When the DA16200/DA16600 is in DPM low power mode, user input is not accepted by the debug terminal. This is because the UART interfaces are powered down during DPM low power mode.

To exit this state and start over with `setup`, follow the steps below:

1. Copy the string `dpm hold` to the clipboard.
   a. For example: open Notepad, type `dpm hold`, and then copy (Ctrl + C) the command string.
2. Use RTC_PWR_KEY to power off (move to OFF position).
3. Use RTC_PWR_KEY to power on (move to ON position).
4. Before the message "`>>> Start DPM Power-Down !!!`" is printed on the console, quickly follow the steps below:
   a. Paste the `dpm hold` string in the terminal window and immediately click the **ENTER** key.
      i. For Windows tera term, use ALT-R or right click the window to paste the string.
      ii. For Linux minicom, use CTRL-V or right click the window to paste the string.
   b. Once this procedure is done quickly and successfully, the message "`- DPM Sleep Manager HOLD…`" is displayed.
   c. If DPM low power mode does not stop successfully, then try again.

```
         ******************************************************
         *            DA16200 SDK Information
         * ---------------------------------------------------
         *
         * - CPU Type        : Cortex-M4 (120MHz)
         * - OS Type         : FreeRTOS 10.4.3
         * - Serial Flash    : 4 MB
         * - SDK Version     : V3.1.3.0 GEN
         * - F/W Version     : FRTOS-GEN01-01-15129-000000
         * - F/W Build Time  : Aug 26 2021 22:58:01
         * - Boot Index      : 0
         *
         ******************************************************

System Mode : Station Only (0)
>>> DA16x Supp Ver2.7 - 2020_07
>>> Wi-Fi mode : b/g/n -> b/g (for DPM)
>>> MAC address (sta0) : d4:3d:39:10:a2:48
>>> sta0 interface add OK
>>> Start STA mode...
dpm hold

- DPM Sleep Manager HOLD ...
>>> Network Interface (wlan0) : UP
>>> Associated with 80:ca:4b:30:02:0a

Connection COMPLETE to 80:ca:4b:30:02:0a

-- DHCP Client WLAN0: SEL(6)
-- DHCP Client WLAN0: REQ(1)
-- DHCP Client WLAN0: CHK(8)
-- DHCP Client WLAN0: BOUND(10)
         Assigned addr   : 192.168.219.105
               netmask   : 255.255.255.0
               gateway   : 192.168.219.1
               DNS addr  : 61.41.153.2

         DHCP Server IP  : 192.168.219.1
         Lease Time      : 06h 00m 00s
         Renewal Time    : 05h 00m 00s

[/DA16200] #
```

5. Once the DPM Hold command is successfully applied, debug commands such as `setup` is available again.
6. DPM mode can be disabled by running the `dpm off` command at the command prompt.

   The DA16200/DA16600 will reboot and the command prompt is available again.

```
[/DA16200] # dpm off

Wakeup source is 0x1
[dpm_init_retmemory] DPM INIT CONFIGURATION(1)

        ******************************************************
        *             DA16200 SDK Information
        * ---------------------------------------------------
        *
        * - CPU Type        : Cortex-M4 (120MHz)
        * - OS Type         : FreeRTOS 10.4.3
        * - Serial Flash    : 4 MB
        * - SDK Version     : V3.1.3.0 GEN
        * - F/W Version     : FRTOS-GEN01-01-15129-000000
        * - F/W Build Time  : Aug 26 2021 22:58:01
        * - Boot Index      : 0
        *
        ******************************************************

System Mode : Station Only (0)
>>> DA16x Supp Ver2.7 - 2020_07
>>> MAC address (sta0) : d4:3d:39:10:a2:48
>>> sta0 interface add OK
>>> Start STA mode...

>>> Network Interface (wlan0) : UP
>>> Associated with 80:ca:4b:30:02:0a

Connection COMPLETE to 80:ca:4b:30:02:0a

-- DHCP Client WLAN0: SEL(6)
-- DHCP Client WLAN0: REQ(1)
-- DHCP Client WLAN0: CHK(8)
-- DHCP Client WLAN0: BOUND(10)
         Assigned addr   : 192.168.219.105
               netmask   : 255.255.255.0
               gateway   : 192.168.219.1
               DNS addr  : 61.41.153.2

         DHCP Server IP  : 192.168.219.1
         Lease Time      : 06h 00m 00s
         Renewal Time    : 03h 00m 00s
[/DA16200] #
```

NOTE
The **dpm hold** command should be used to check the status of DPM operation on the console. After entering dpm hold state of the DA16200, performing normal operations other than for debugging purposes is not permitted.

6.5 Current Measurement

To measure the current waveform, connect the EVB's current measurement point (P2) with the measurement instrument. For more detailed information on the Sleep modes, see the Low Power Operation Mode section in the DA16200 Datasheet Ref. [1].

6.5.1 Test Setup - KEYSIGHT 14585A

Figure 48 shows a typical test setup environment.

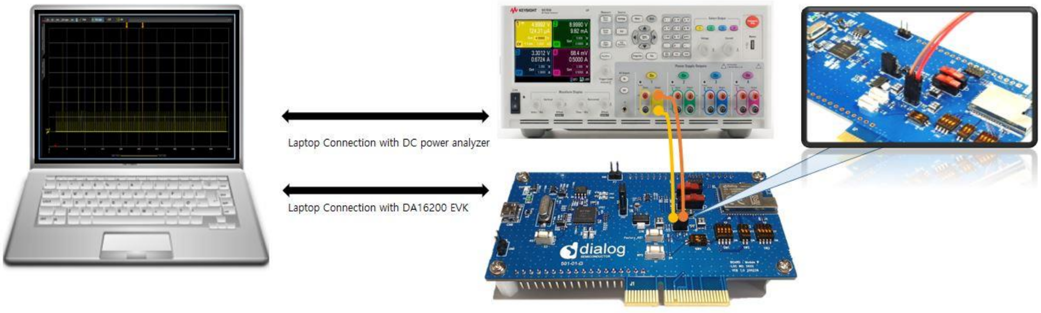

Laptop Connection with DC power analyzer
Laptop Connection with DA16200 EVK

Figure 48: Current Test Environment with 14585A

6.5.2 Test Setup with EVK PRO Board

EVK Pro board is a simple evaluation board without any additional devices to measure current.

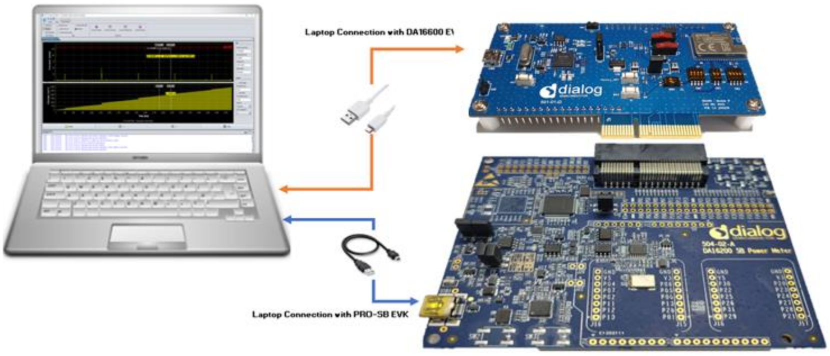

Laptop Connection with DA16600 E\
Laptop Connection with PRO-SB EVK

Figure 49: Current Test Environment with EVK PRO for DA16200

1. To measure current with DA16200 EVK PRO board, connect the two boards with the PCI connector.
2. Open the two jumpers (P2) of the power line on the DA16200 EVB as shown in Figure 50.
3. Power is supplied from the EVK PRO board, and current consumption of DA16200 can be measured.

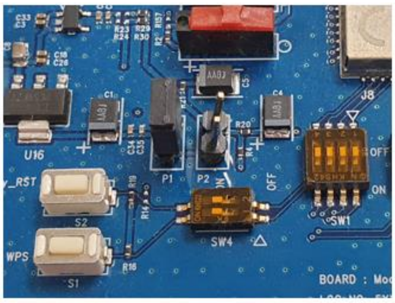


Figure 50: Setup DA16200 EVB Jumper

Figure 51 shows a typical test setup environment with DA16600 EVK PRO board and Wi-Fi IoT Power Profiler Tool.

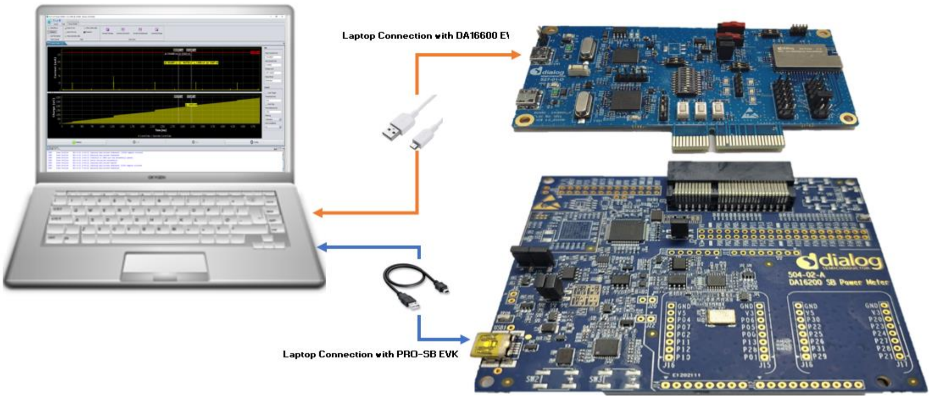

Laptop Connection with DA16600 EV
Laptop Connection with PRO-SB EVK
dialog
504-02-A
DA16200 SB Power Meter

Figure 51: Current Test Environment with EVK PRO for DA16600

1. To measure current with DA16600 EVK PRO board, connect the two boards with the PCI connector.
2. Change the jumper setting (P1 and P2) and switch (SW5) settings on the DA16600 EVB.
3. Open the DA16600 EVB's power two jumper caps as shown in Figure 52.

There is a selectable DIP Switch (SW5):

- If SW (1) moves to switch on, measure the current of Wi-Fi chipset
- If SW (2) moves to switch on, measure the current of the Bluetooth® LE chipset

By switching on both, users can measure all current consumed by Bluetooth® LE and Wi-Fi.

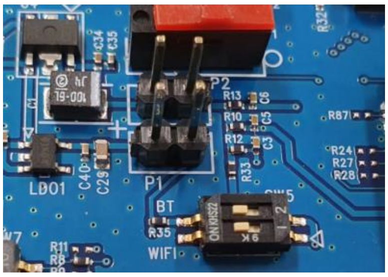


Figure 52: Setup DA16600 EVB Jumper

6.5.2.1 DA16200/DA16600 EVK PRO Board (504-02-A)

The actual component locations of the DA16200/DA16600 EVK PRO Board are shown in Figure 53.

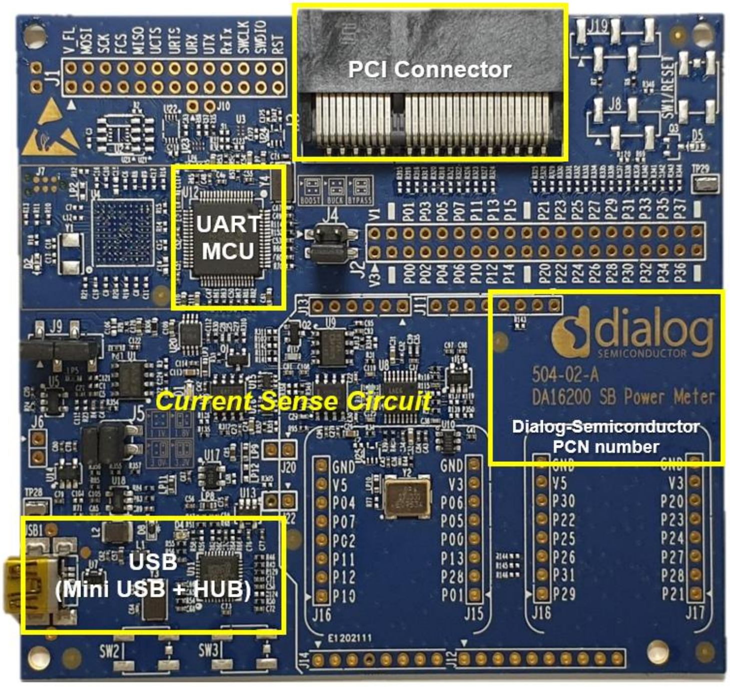


Figure 53: DA16200/DA16600 EVK PRO Board (504-02-A)

- PCI Connector: a connector on which to install the DA16200/DA16600 EVB board
- UART MCU: provides communication between DA16200/DA16600 EVK PRO board and PC. Also transfers the current measurement samples to the PC
- Current Sense Circuit: monitors the current of the DA16200/DA16600 EVB

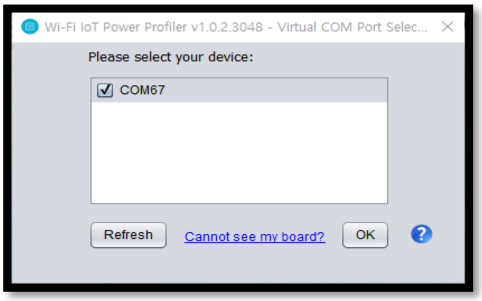
USB Hub: USB interfaces to the PC

6.5.2.2 Wi-Fi IoT Power Profiler

Wi-Fi IoT Power Profiler uses the SPI port of a device connected via USB for communication. The user needs to select the SPI port to connect before using the Power Profiler tool.

To select the SPI port, simply click the checkbox next to the SPI port. The port might be preselected already since the application remembers the last selected SPI port. If necessary, use the **Refresh** button to update the list of available SPI ports for the devices connected via USB. If the device is connected but not listed, it is better to use a different USB port, wait for a few seconds and then click the **Refresh** button again. Problems in identifying the FTDI device may indicate an invalid installation of the FTDI drivers.

The ? button can be used to access the application's user guide, release notes, license and about info. When the user clicks the **OK** button to launch the main application window as Figure 54, the SmartSnippets Wi-Fi IoT Power Profiler tool can be downloaded from the Renesas website (https://www.renesas.com/us/en/products/wireless-connectivity/wi-fi/low-power-wi-fi). Go to the website and scroll down to Software Downloads section to download it.


Wi-Fi IoT Power Profiler v1.0.2.3048 - Virtual COM Port Selec...
Refresh
Cannot see my board?
OK

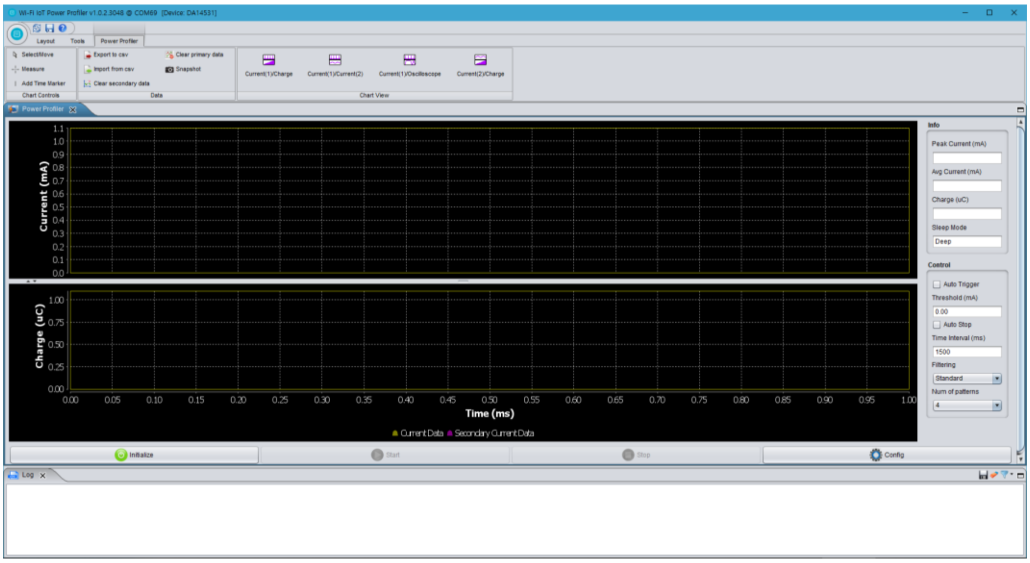
Please select your device:

☑ COM67

Figure 54: Select Virtual COM Port

6.5.2.3 Wi-Fi IoT Power Profiler Setup

When the proper COM port is selected as Figure 54, click the **OK** button. Then, the main window of the Power Profiler application in SmartSnippets Wi-Fi IoT Power Profiler Toolbox will pop up as shown in Figure 55. Click the **Config** button at the bottom right and check that the configuration is correct as shown in Figure 56.



Info

Peak Current (mA)

Avg Current (mA)

Charge (uC)

Sleep Mode: Deep

Control

□ Auto Trigger

Threshold (mA): 0.00

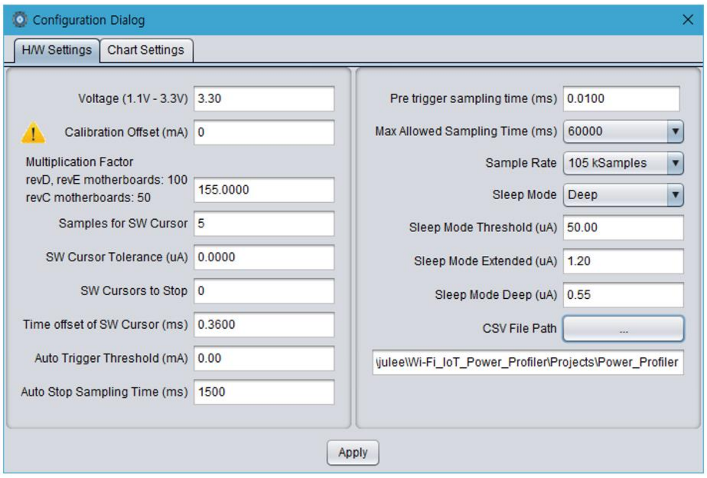
□ Auto Stop

Time Interval (ms): 1500

Filtering: Standard

Num of patterns: 4

Figure 55: Power Profiler of SmartSnippets Toolbox



Voltage (1.1V - 3.3V)	3.30
Calibration Offset (mA)	0
Multiplication Factor revD, revE motherboards: 100 revC motherboards: 50	155.0000
Samples for SW Cursor	5
SW Cursor Tolerance (uA)	0.0000
SW Cursors to Stop	0
Time offset of SW Cursor (ms)	0.3600
Auto Trigger Threshold (mA)	0.00
Auto Stop Sampling Time (ms)	1500

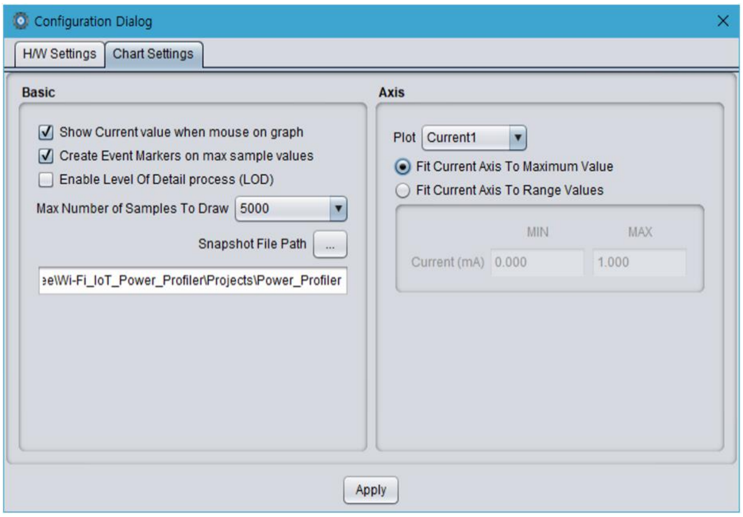
Pre trigger sampling time (ms)	0.0100
Max Allowed Sampling Time (ms)	60000
Sample Rate	105 kSamples
Sleep Mode	Deep
Sleep Mode Threshold (uA)	50.00
Sleep Mode Extended (uA)	1.20
Sleep Mode Deep (uA)	0.55
CSV File Path	...

\julee\Wi-Fi_IoT_Power_Profiler\Projects\Power_Profiler

Figure 56: Power Profiler Configuration

In the H/W Settings tab of the new profile, the **Voltage (1.1 V - 3.3V)** field is typically empty and the **Calibration Offset (mA)** field is zero. Apply the correct voltage as configured in the Power Meter LDO (typically, it is 3.3 V). The Calibration offset (mA) will be filled automatically with a calculated value. This value is good enough to measure with ±2% accuracy, so in most cases we can leave it as is. Check the multiplication factor. For DA16200/DA16600, it should be **155**.

The user can set a maximum scale for the waveform window optionally, since there are large peaks on wakeup from sleep (charging capacitors) generally. Thus, the useful part of the signal is compressed to a very small area. To set a maximum scale for the waveform window, open the **Chart Settings** tab as shown in Figure 57 and for **Plot** *Current1*, set the **MAX** scale of **Current (mA)** to 10 mA. Note that this does not affect anything to the accuracy of the measurement. It can only zoom the waveform, which is presented in the viewer. Click **Apply** and return to the measurement window.



☑ Show Current value when mouse on graph

☑ Create Event Markers on max sample values

□ Enable Level Of Detail process (LOD)

Max Number of Samples To Draw: 5000

Snapshot File Path: ...

e\Wi-Fi_IoT_Power_Profiler\Projects\Power_Profiler

Plot: Current1

● Fit Current Axis To Maximum Value

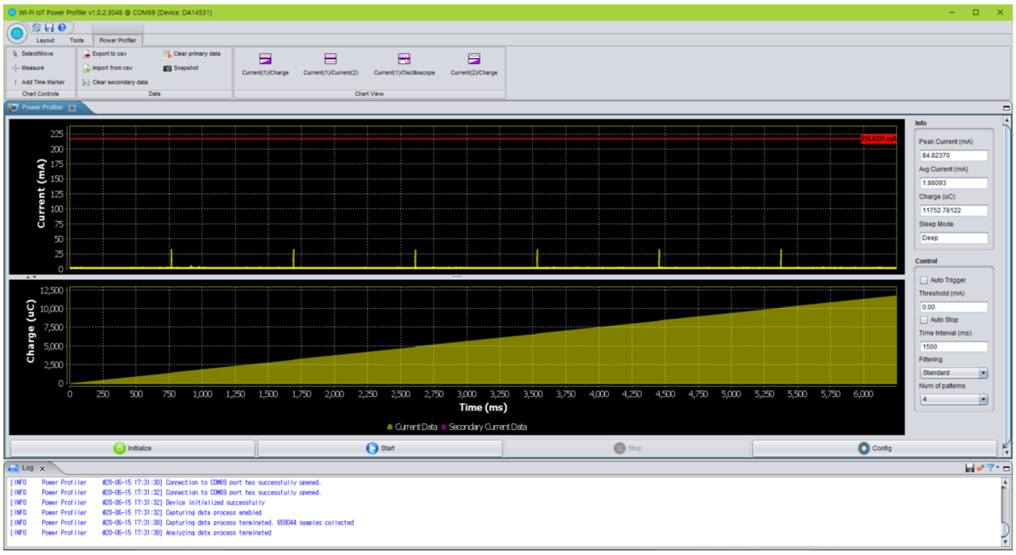
○ Fit Current Axis To Range Values

	MIN	MAX
Current (mA)	0.000	1.000

Figure 57: Power Profiler Chart Settings

6.5.2.4 Measurements

In the measurement window as shown in Figure 58, click **Initialize** at the bottom left (this step is only needed at the first communication with the A/D converter). Click **Start**.



Peak Current (mA): 84.82370

Avg Current (mA): 1.88093

Charge (uC): 11752.78122

Sleep Mode: Deep

□ Auto Trigger

Threshold (mA): 0.00

□ Auto Stop

Time Interval (ms): 1500

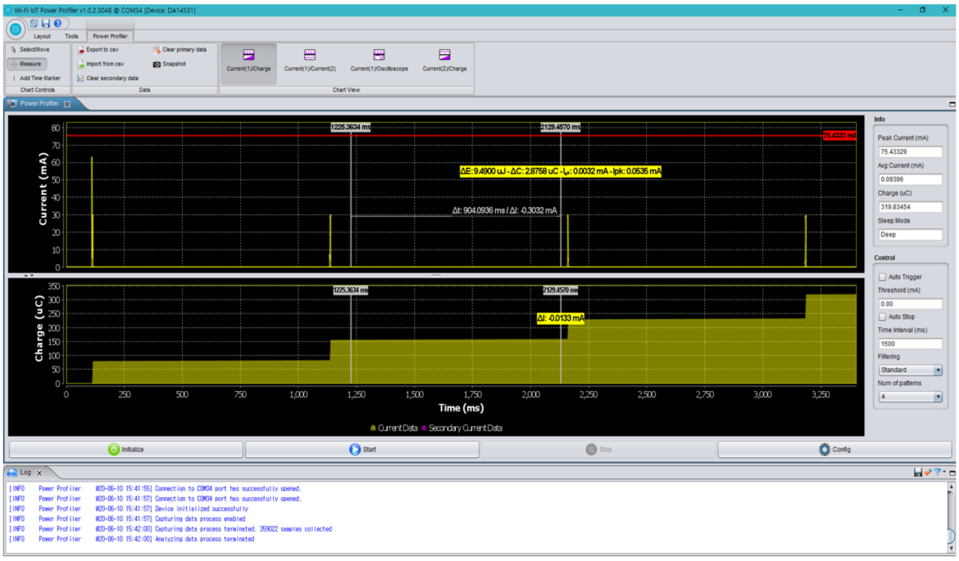
Filtering: Standard

Num of patterns: 4

```
[INFO    Power Profiler    @20-06-15 17:31:30] Connection to COM69 port has successfully opened.
[INFO    Power Profiler    @20-06-15 17:31:32] Connection to COM69 port has successfully opened.
[INFO    Power Profiler    @20-06-15 17:31:32] Device initialized successfully
[INFO    Power Profiler    @20-06-15 17:31:32] Capturing data process enabled
[INFO    Power Profiler    @20-06-15 17:31:38] Capturing data process terminated. 659044 samples collected
[INFO    Power Profiler    @20-06-15 17:31:38] Analyzing data process terminated
```

Figure 58: Free Running Capture

Select the **Power Profiler** > **Measure** menu on the window, users can measure the current consumption in the desired section as peak and average by using a mouse.



Info

Peak Current (mA): 75.43329

Avg Current (mA): 0.09396

Charge (uC): 319.83454

Sleep Mode: Deep

Control

□ Auto Trigger

Threshold (mA): 0.00

□ Auto Stop

Time Interval (ms): 1500

Filtering: Standard

Num of patterns: 4

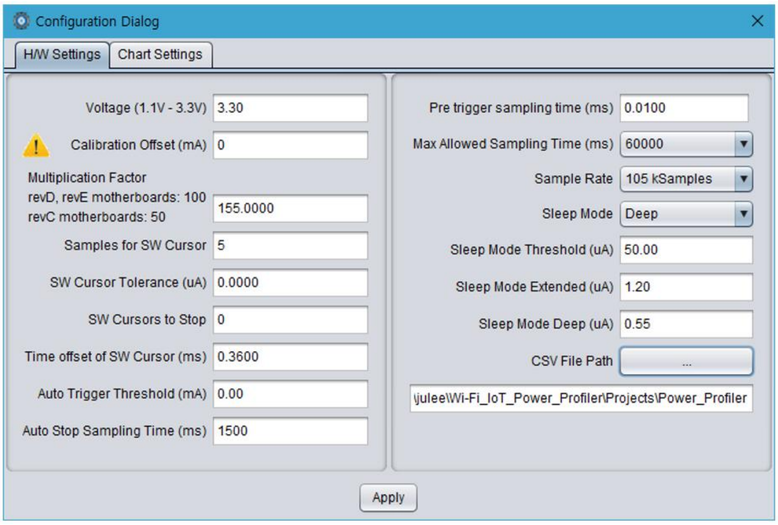
```
[INFO    Power Profiler    @20-06-10 15:41:55] Connection to COM34 port has successfully opened.
[INFO    Power Profiler    @20-06-10 15:41:57] Connection to COM34 port has successfully opened.
[INFO    Power Profiler    @20-06-10 15:41:57] Device initialized successfully
[INFO    Power Profiler    @20-06-10 15:41:57] Capturing data process enabled
[INFO    Power Profiler    @20-06-10 15:42:00] Capturing data process terminated. 359022 samples collected
[INFO    Power Profiler    @20-06-10 15:42:00] Analyzing data process terminated
```

Figure 59: Consumption Measurement by Using Chart Controls

6.5.2.5 Manual Calibration

For the most accurate measurement, follow the manual calibration steps below:

1. Remove the daughterboard, i.e., the load, by physically disconnecting the DA16200/DA16600 EVB from the motherboard.
2. In the main window of the Power Profiler, click **Config** and temporarily set the **Calibration Offset (mA)** field to zero as shown in Figure 60.



Voltage (1.1V - 3.3V): 3.30

Calibration Offset (mA): 0

Multiplication Factor
revD, revE motherboards: 100
revC motherboards: 50: 155.0000

Samples for SW Cursor: 5

SW Cursor Tolerance (uA): 0.0000

SW Cursors to Stop: 0

Time offset of SW Cursor (ms): 0.3600

Auto Trigger Threshold (mA): 0.00

Auto Stop Sampling Time (ms): 1500

Pre trigger sampling time (ms): 0.0100

Max Allowed Sampling Time (ms): 60000

Sample Rate: 105 kSamples

Sleep Mode: Deep

Sleep Mode Threshold (uA): 50.00

Sleep Mode Extended (uA): 1.20

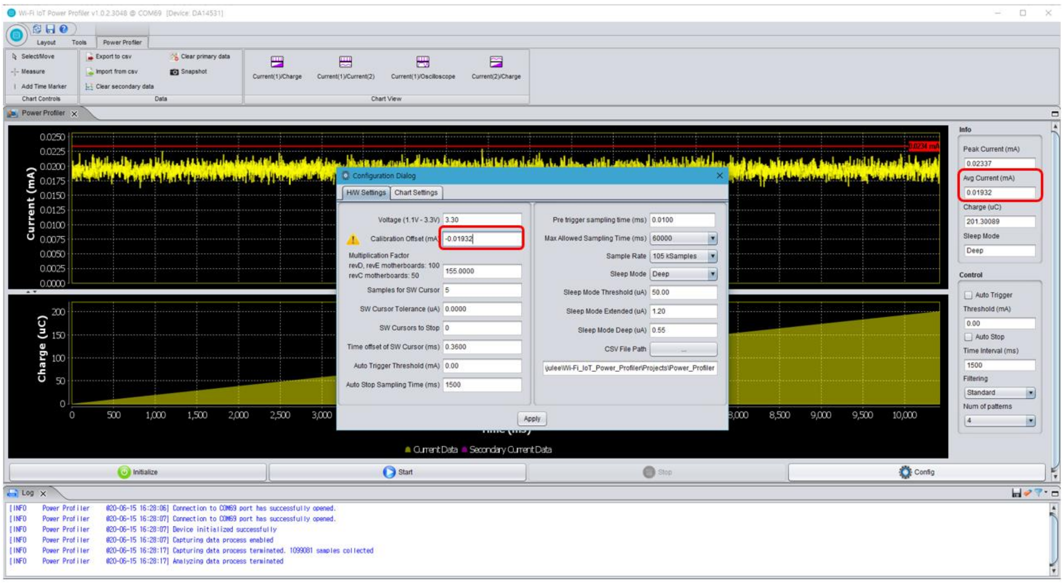
Sleep Mode Deep (uA): 0.55

CSV File Path: ...

\julee\Wi-Fi_IoT_Power_Profiler\Projects\Power_Profiler

Figure 60: Setting Offset to Zero

3. Run a free running capture for a few seconds and note down the number in the **Avg Current (mA)** field as Figure 61. Note, the last digits of the number keep changing slowly due to temperature drift. It is important to do the calibration steps periodically.
4. Finally, put this average value with a negative sign in the **Calibration Offset (mA)** field. CTRL-C and CTRL-V are also workable as Figure 61.



Configuration Dialog

H/W Settings | Chart Settings

Voltage (1.1V - 3.3V): 3.30
Calibration Offset (mA): -0.01932
Multiplication Factor revD, revE motherboards: 100 revC motherboards: 50: 155.0000
Samples for SW Cursor: 5
SW Cursor Tolerance (uA): 0.0000
SW Cursors to Stop: 0
Time offset of SW Cursor (ms): 0.3600
Auto Trigger Threshold (mA): 0.00
Auto Stop Sampling Time (ms): 1500

Pre trigger sampling time (ms): 0.0100
Max Allowed Sampling Time (ms): 60000
Sample Rate: 105 kSamples
Sleep Mode: Deep
Sleep Mode Threshold (uA): 50.00
Sleep Mode Extended (uA): 1.20
Sleep Mode Deep (uA): 0.55
CSV File Path: \julee\Wi-Fi_IoT_Power_Profiler\Projects\Power_Profiler

Apply

Info

Peak Current (mA): 0.02337
Avg Current (mA): 0.01932
Charge (uC): 201.30089
Sleep Mode: Deep

Control

□ Auto Trigger
Threshold (mA): 0.00
□ Auto Stop
Time Interval (ms): 1500
Filtering: Standard
Num of patterns: 4

```
[INFO    Power Profiler    020-06-15 16:28:06] Connection to COM69 port has successfully opened.
[INFO    Power Profiler    020-06-15 16:28:07] Connection to COM69 port has successfully opened.
[INFO    Power Profiler    020-06-15 16:28:07] Device initialized successfully
[INFO    Power Profiler    020-06-15 16:28:07] Capturing data process enabled
[INFO    Power Profiler    020-06-15 16:28:17] Capturing data process terminated. 1099081 samples collected
[INFO    Power Profiler    020-06-15 16:28:17] Analyzing data process terminated
```

Figure 61: Setting Calibration Offset

The user manual can be found in the local folder where the DA16200 DA16600 Evaluation Kit Pro Power Profiler.is downloaded.

6.5.3 Current Measurement at Sleep Mode 1

To measure current in Sleep 1, power off the EVB by setting **RTC_PWR_KEY** to the **OFF** position (**SW6** on the DA16200 EVB or **SW2** on the DA16600 EVB).

6.5.4 Current Measurement at Sleep Mode 2

To measure current in Sleep 2, use the CLI commands as follows.

1. Power on the EVB by setting **RTC_PWR_KEY** to the **ON** position (**SW6** on the DA16200 EVB or **SW2** on the DA16600 EVB).
2. Type the `factory` command to set the DA16200/DA16600 to its default settings.
3. After the EVB reboots, run the `sys.hal` command to enter 'hal' mode.
   - ○ `[/DA16200] # sys.hal`
4. Run the `sleep 2 <time(sec)>` command to enter sleep 2 mode for the specified time in seconds.
   - ○ `[/DA16200/SYS.HAL] # sleep 2 <time(sec)>`

   For instance, `[/DA16200/SYS.HAL] # sleep 2 10`

   It will make DA16200 stay at sleep 2 for a set amount of time (10 seconds) and wake up.

6.5.5 Current Measurement at Sleep Mode 3

To measure current in Sleep 3, use the CLI commands as follows.

5. Power on the EVK by setting **RTC_PWR_KEY** to the **ON** position (**SW6** on the DA16200 EVB or **SW2** on the DA16600 EVB).
6. Type the `factory` command to set the DA16200/DA16600 to its default settings.
7. After the EVK reboots, run the `sys.hal` command to enter 'hal' mode.
   - ○ `[/DA16200] # sys.hal`
8. Run the `sleep 3 <time(sec)>` command to enter sleep 3 mode for the specified time in seconds.
   - ○ `[/DA16200/SYS.HAL] # sleep 3 <time(sec)>`

For instance, `[/DA16200/SYS.HAL] # sleep 3 10`

It will make DA16200 stay at sleep 3 for a set amount of time (10 seconds) and wake up.

6.5.6 Current Measurement at DPM Low Power Mode

- Enable DPM mode as described in Section 6.4.2
  - When running the DA16200 with DPM settings, DA16200 will run `sleep 3 state` and `wake up for Beacon check` and `Keep Alive` according to the configured DTIM

For example: Figure 62 shows the current waveform with settings DTIM 10 (about 1sec @ AP DTIM=1) and Keep Alive 30s. Sleep 3 means sleep state in the interval between RXs or between RX and TX.

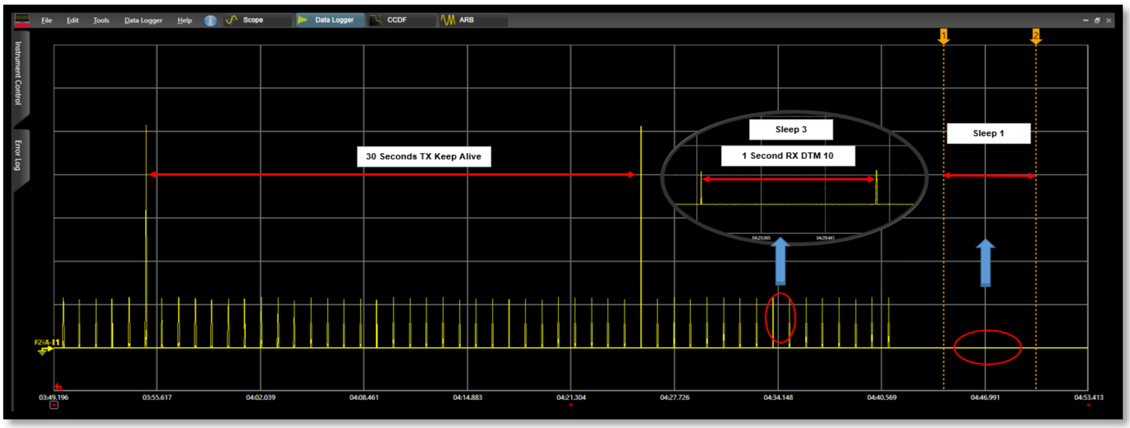


Figure 62: Current Measurement at DPM Low Power Mode

Appendix A Old Versions of EVB

A.1 DA16200 EVB Ver 10.0

Figure 63 shows the hardware configuration of the DA16200 EVB.

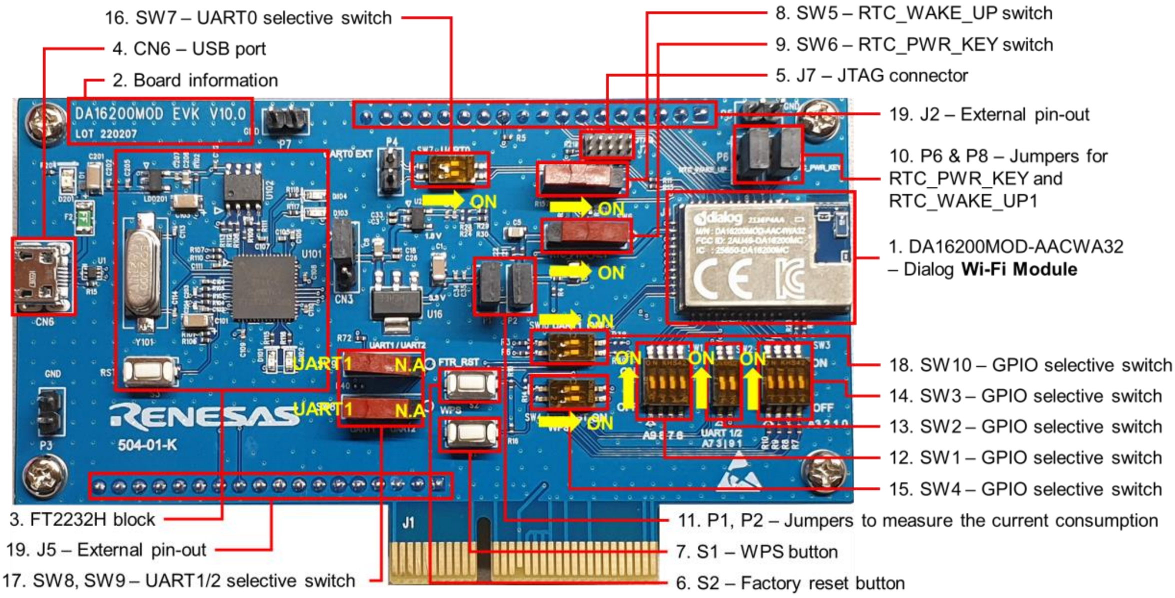
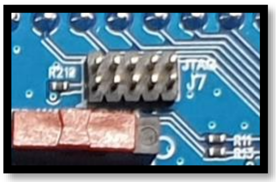
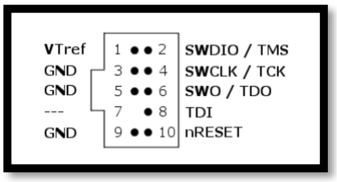

16. SW7 – UART0 selective switch
4. CN6 – USB port
2. Board information
8. SW5 – RTC_WAKE_UP switch
9. SW6 – RTC_PWR_KEY switch
5. J7 – JTAG connector
19. J2 – External pin-out
10. P6 & P8 – Jumpers for RTC_PWR_KEY and RTC_WAKE_UP1
1. DA16200MOD-AACWA32 – Dialog Wi-Fi Module
18. SW10 – GPIO selective switch
14. SW3 – GPIO selective switch
13. SW2 – GPIO selective switch
12. SW1 – GPIO selective switch
15. SW4 – GPIO selective switch
11. P1, P2 – Jumpers to measure the current consumption
7. S1 – WPS button
6. S2 – Factory reset button
3. FT2232H block
19. J5 – External pin-out
17. SW8, SW9 – UART1/2 selective switch
ON
ON
ON
ON
ON
ON
ON
ON
UART1
N.A
UART1
N.A
DA16200MOD EVK V10.0
LOT 220207
504-01-K

Figure 63: DA16200 EVB Hardware Configuration

DA16200 EVB Ver 10.0 has the following components:

Table 16: Components on DA16200 EVB

No	Name	Description
1	**DA16200MOD-AACWA32**	Renesas **Wi-Fi Module.**
2	**Board Information**	Board version and manufacturing date.
3	**FH2232H Block**	USB to UART/SPI interfaces.
4	**CN6 – USB Port**	Provides UART0 for debug, UART1/2 for test and SPI.
5	**J7 – JTAG connector**	Connector for the IAR I-jet JTAG Debugger   VTref 1 ●● 2 SWDIO / TMS GND 3 ●● 4 SWCLK / TCK GND 5 ●● 6 SWO / TDO --- 7 ● 8 TDI GND 9 ●● 10 nRESET Note: Pin 7 of the I-Jet debugger cable is keyed with a white plug so pin 7 must be removed from the EVB.
6	**S2 – Factory reset button**	Factory reset button using GPIOA_7. To enable this button, set Pin 2 of SW4 to ON and press the button for more than 5 seconds to initialize `nvram` data.
7	**S1 – WPS button**	WPS button using GPIOA_6. To enable this button, set Pin 1 of SW4 to ON and press the button to start WPS mode.
8	**SW5 – RTC_WAKE_UP switch**	Switch to wake up the board from sleep mode.

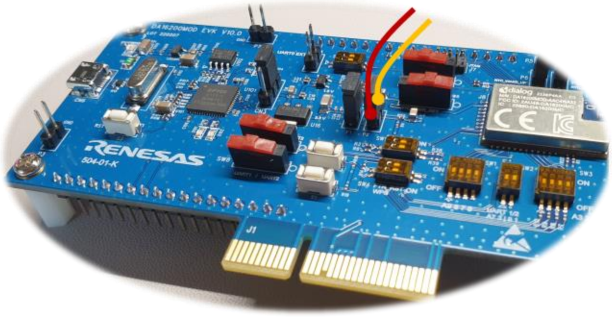
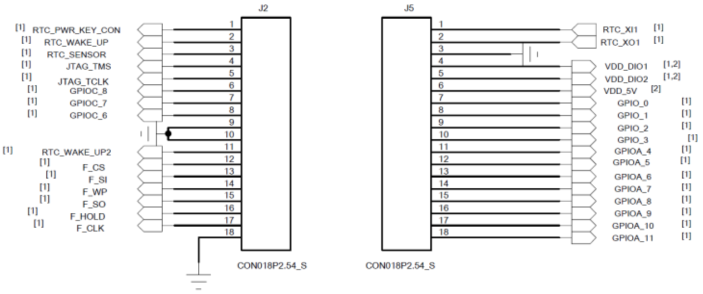
No	Name	Description
9	**SW6 – RTC_PWR_KEY switch**	Switch to turn the EVB ON and OFF.
10	**P6 – Jumpers for RTC_WAKE_UP1**	Pin for connecting the external control signal with MCU. For normal operation, this jumper should be shorted.
	P8 – Jumpers for RTC_PWR_KEY	Pin for connecting the external control signal with MCU. For normal operation, this jumper should be shorted.
11	**P1 – External VDD**	External VDD (3.3V) can be supplied to the P1 pins. For normal operation, this jumper should be shorted.
	P2 – Jumper to measure current consumption of the DA16200	Pin for current measurement. For normal operation, this pin should be shorted. - Pull out the Short Pin cap and use jumper wires to connect to the measuring equipment. 
12	**SW1 – GPIO selection switch**	GPIO selection switch, default OFF, see Table 17.
13	**SW2 – GPIO selection switch**	GPIO selection switch, default ON, see Table 17.
14	**SW3 – GPIO selection switch**	GPIO selection switch, default OFF, see Table 17.
15	**SW4 – GPIO selection switch**	GPIO selection switch, default ON, see Table 17.
16	**SW7 – UART0 selection switch**	UART0 selection switch, default ON, see Table 17.
17	**SW8, SW9 – UART1/2 selection switch**	UART1/2 selection switch, default LEFT (UART1), see
18	**SW10 – GPIO selection switch**	GPIO selection switch, default ON, see Table 17.
19	**J2, J5 – External pin-out**	GPIO connector.  J2 [1] RTC_PWR_KEY_CON [1] RTC_WAKE_UP [1] RTC_SENSOR [1] JTAG_TMS [1] JTAG_TCLK [1] GPIOC_8 [1] GPIOC_7 [1] GPIOC_6 [1] RTC_WAKE_UP2 [1] F_CS [1] F_SI [1] F_WP [1] F_SO [1] F_HOLD [1] F_CLK 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 CON018P2.54_S J5 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 CON018P2.54_S RTC_XI1 [1] RTC_XO1 [1] VDD_DIO1 [1,2] VDD_DIO2 [1,2] VDD_5V [2] GPIO_0 [1] GPIO_1 [1] GPIO_2 [1] GPIO_3 [1] GPIOA_4 [1] GPIOA_5 [1] GPIOA_6 [1] GPIOA_7 [1] GPIOA_8 [1] GPIOA_9 [1] GPIOA_10 [1] GPIOA_11 [1]

Table 17: DA16200 EVB Switch Description

Switch	Pin	Relevant GPIO	ON	OFF	Default
SW1	1	GPIOA6	Connects FT2232H to SPI_CSB (Note 1)	WPS	OFF
	2	GPIOA7	Connects FT2232H to SPI_CLK (Note 1)	Factory Reset	
	3	GPIOA8	Connects FT2232H to SPI_MISO (Note 1)	External pin-out (J2/J5) only	
	4	GPIOA9	Connects FT2232H to SPI_MOSI (Note 1)		

SW2	1	GPIOA4	Connects FT2232H to UART1_TXD	External pin-out (J2/J5) only	ON
	2	GPIOA5	Connects FT2232H to UART1_RXD		
SW3	1	GPIOA0	Connects FT2232H to SPI_CSB (Note 1)	External pin-out (J2/J5) only	OFF
	2	GPIOA1	Connects FT2232H to SPI_CLK (Note 1)		
	3	GPIOA2	Connects FT2232H to SPI_MISO (Note 1)		
	4	GPIOA3	Connects FT2232H to SPI_MOSI (Note 1)		
SW4	1	GPIOA6	WPS	External pin-out (J2/J5) only	ON
	2	GPIOA7	Factory Reset		
SW7	1	UART0_TXD	Connects FT232H to UART0_TXD	NC	ON
	2	UART0_RXD	Connects FT232H to UART0_RXD		
SW8	-	GPIOA4	Not available on DA16200 EVB V10.0 (Right)	Connects SW2 to SW10 (Left)	LEFT
SW9	-	GPIOA5	Not available on DA16200 EVB V10.0 (Right)	Connects SW2 to SW10 (Lett)	LEFT
SW10	1	GPIOA4	Connects SW8 to UART1_TXD	External pin-out (J2/J5) only	ON
	2	GPIOA5	Connects SW9 to UART1_RXD		

Note 1 By default, the SPI interface is configured to support firmware download through the FT2232H. See section 4.7 for details on how to configure the EVB to support AT Command processing from an MCU directly connected to the SPI interface.

The current consumption can be measured at P2 jumper with current measuring equipment. See Section 6.5 for details on the test setup.

When DA16200 EVK Pro is used for measuring current consumption, the P1 jumper must be removed. For more information on DA16200 EVK Pro, see 6.5.2.

For more details on the DA16200 EVB, see the schematic Ref. [9].

A.2 DA16200 EVB Ver 9.0

Figure 64 shows the hardware configuration of the DA16200 EVB Ver 9.0.

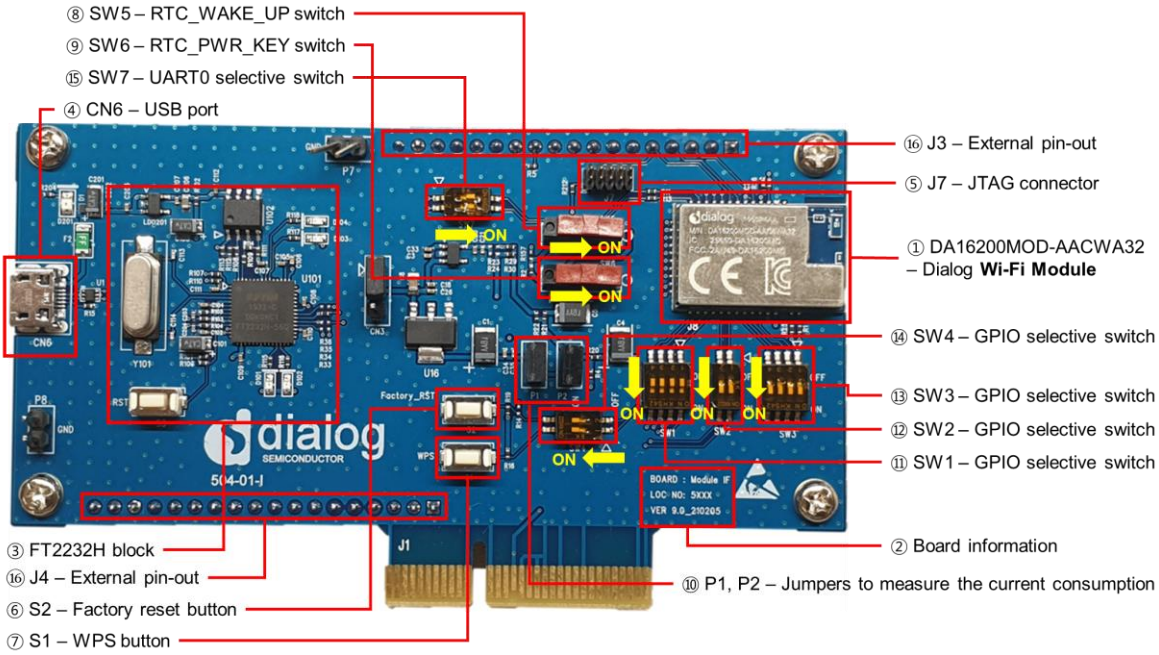
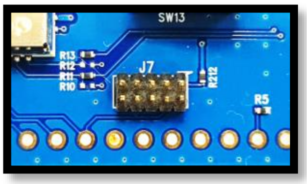
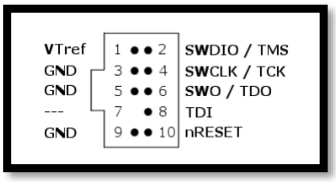

⑧ SW5 – RTC_WAKE_UP switch
⑨ SW6 – RTC_PWR_KEY switch
⑮ SW7 – UART0 selective switch
④ CN6 – USB port
⑯ J3 – External pin-out
⑤ J7 – JTAG connector
① DA16200MOD-AACWA32 – Dialog Wi-Fi Module
⑭ SW4 – GPIO selective switch
⑬ SW3 – GPIO selective switch
⑫ SW2 – GPIO selective switch
⑪ SW1 – GPIO selective switch
② Board information
⑩ P1, P2 – Jumpers to measure the current consumption
③ FT2232H block
⑯ J4 – External pin-out
⑥ S2 – Factory reset button
⑦ S1 – WPS button
ON
ON
ON
ON
ON
ON
ON
dialog SEMICONDUCTOR
504-01-I
BOARD : Module IF
LOC NO: 5XXX
VER 9.0_210205

Figure 64: DA16200 EVB Ver 9.0 Hardware Configuration

DA16200 EVB V9.0 has the following components:

Table 18: Components on DA16200 EVB

No	Name	Description
1	**DA16200MOD-AACWA32**	Renesas **Wi-Fi Module.**
2	**Board information**	Board version and manufacturing date.
3	**FH2232H block**	USB to UART/SPI interfaces.
4	**CN6 – USB port**	Provides UART0 for debug, UART1/2 for test and SPI.
5	**J7 – JTAG connector**	Connector for the IAR I-jet JTAG Debugger   VTref 1 ●● 2 SWDIO / TMS GND 3 ●● 4 SWCLK / TCK GND 5 ●● 6 SWO / TDO --- 7 ● 8 TDI GND 9 ●● 10 nRESET Note: Pin 7 of the I-Jet debugger cable is keyed with a white plug so pin 7 must be removed from the EVB.
6	**S2 – Factory reset button**	Factory reset button using GPIOA7. To enable this button, set Pin 2 of SW4 to ON and press the button for more than 5 seconds to initialize `nvram` data.
7	**S1 – WPS button**	WPS button using GPIOA6. To enable this button, set Pin 1 of SW4 to ON and press the button to start WPS mode.
8	**SW5 – RTC_WAKE_UP switch**	Switch to wake up the board from sleep mode.
9	**SW6 – RTC_PWR_KEY switch**	Switch to turn the EVB ON and OFF.
10	**P1 – External VDD**	External VDD (3.3V) can be supplied to the P1 pins. For normal operation, this jumper should be shorted.

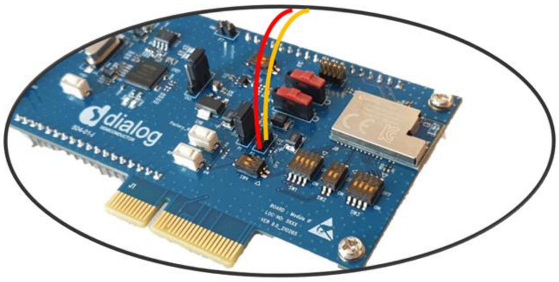
No	Name	Description
10	**P2 – Jumper to measure current consumption of the DA16200**	Pin for current measurement. For normal operation, this pin should be shorted. - Pull out the Short Pin cap and use jumper wires to connect to the measuring equipment. 
11	**SW1 – GPIO selection switch**	GPIO selection switch, default OFF, see Table 19.
12	**SW2 – GPIO selection switch**	GPIO selection switch, default ON, see Table 19.
13	**SW3 – GPIO selection switch**	GPIO selection switch, default ON, see Table 19.
14	**SW4 – GPIO selection switch**	GPIO selection switch, default ON, see Table 19.
15	**SW7 – UART0 selection switch**	UART0 selection switch, default ON, see Table 19.
16	**J3, J4 – External Connector**	Provides access to all pins of the DA16200MOD for testing and debugging.

Table 19: GPIO and SPI Selection Switch

Switch	Pin	Relevant GPIO	On	Off
SW1	1	GPIOA6	Connects FT2232H to SPI_CSB	WPS
	2	GPIOA7	Connects FT2232H to SPI_CLK	Factory Reset
	3	GPIOA8	Connects FT2232H to SPI_MISO	Ext Con (J3/J4) only
	4	GPIOA9	Connects FT2232H to SPI_MOSI	Ext Con (J3/J4) only
SW2	1	GPIOA4	Connects FT2232H to UART1_TXD	Ext Con (J3/J4) only
	2	GPIOA5	Connects FT2232H to UART1_RXD	Ext Con (J3/J4) only
SW3	1	GPIOA0	Connects FT2232H to SPI_CSB	Ext Con (J3/J4) only
	2	GPIOA1	Connects FT2232H to SPI_CLK	Ext Con (J3/J4) only
	3	GPIOA2	Connects FT2232H to SPI_MISO	Ext Con (J3/J4) only
	4	GPIOA3	Connects FT2232H to SPI_MOSI	Ext Con (J3/J4) only
SW4	1	GPIOA6	WPS	Ext Con (J3/J4) only
	2	GPIOA7	Factory Reset	Ext Con (J3/J4) only
SW7	1	UART0_TXD	Connects FT232H to UART0_TXD	NC
	2	UART0_RXD	Connects FT232H to UART0_RXD	NC

The current consumption can be measured at jumper P2 with current measuring equipment. See Section 6.5 for details on the test setup.

When DA16200 EVK Pro Board is used for measuring current consumption, the P1 jumper must be removed.

For more details on the DA16200 EVB, see the schematic Ref. [9].

A.3 DA16600 EVB Ver 5.0

Figure 65 shows the hardware configuration of the DA16600 EVB.

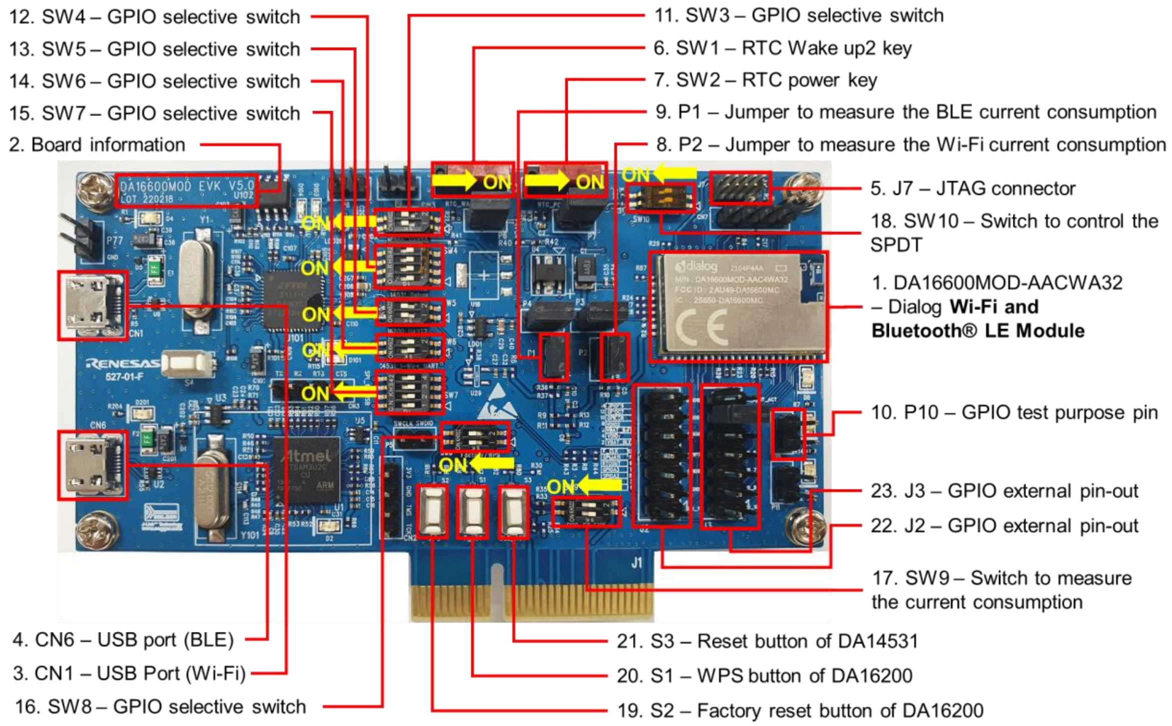
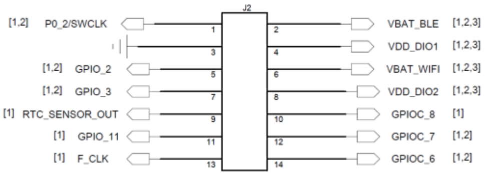
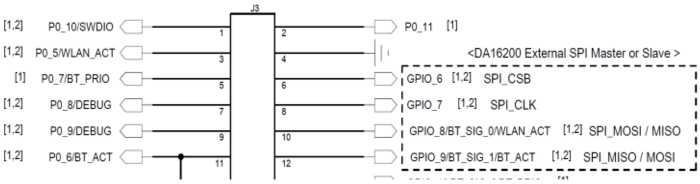

12. SW4 – GPIO selective switch
13. SW5 – GPIO selective switch
14. SW6 – GPIO selective switch
15. SW7 – GPIO selective switch
2. Board information
11. SW3 – GPIO selective switch
6. SW1 – RTC Wake up2 key
7. SW2 – RTC power key
9. P1 – Jumper to measure the BLE current consumption
8. P2 – Jumper to measure the Wi-Fi current consumption
5. J7 – JTAG connector
18. SW10 – Switch to control the SPDT
1. DA16600MOD-AACWA32 – Dialog Wi-Fi and Bluetooth® LE Module
10. P10 – GPIO test purpose pin
23. J3 – GPIO external pin-out
22. J2 – GPIO external pin-out
17. SW9 – Switch to measure the current consumption
4. CN6 – USB port (BLE)
3. CN1 – USB Port (Wi-Fi)
16. SW8 – GPIO selective switch
21. S3 – Reset button of DA14531
20. S1 – WPS button of DA16200
19. S2 – Factory reset button of DA16200
ON
DA16600MOD EVK V5.0
LOT 220218
RENESAS
Atmel
J1

Figure 65: DA16600 EVB Hardware Configuration

DA16600 EVB has the following components:

Table 20: Components on DA16600 EVB

No	Name	Description
1	**DA16600MOD-AACWA32**	Renesas **Wi-Fi and Bluetooth® LE Module.**
2	**Board information**	Board version and manufacturing date
3	**CN1 – USB Port (Wi-Fi)**	Provides UART0 for debug and UART1 for test.
4	**CN6 – USB Port (BLE)**	Connect directly to DA14531 for debug only. Note: Do not use this port during normal operation.
5	**J7 – JTAG Connector**	Connector for IARs I-jet JTAG Debugger VTref 1 ●● 2 SWDIO / TMS GND 3 ●● 4 SWCLK / TCK GND 5 ●● 6 SWO / TDO --- 7 ● 8 TDI GND 9 ●● 10 nRESET Note: Pin 7 on the I-Jet debugger cable is keyed with a white plug so pin 7 must be removed from the EVB.
6	**SW1 – RTC Wake up2 key**	Switch to wake up the board from sleep mode.
7	**SW2 – RTC Power key**	Switch to turn the EVB on and off.

No	Name	Description
8	**P2 – Jumper to measure current consumption of DA16200**	Jumper to measure current used by the Wi-Fi device. For normal operation, this jumper must be shorted.
9	**P1 – Jumper to measure current consumption of DA14531**	Jumper to measure current used by the Bluetooth® LE device. For normal operation, this jumper must be shorted.
10	**P10 – Jumper to test GPIO**	GPIO test pin. Add jumpers from J2/J3 to P10 to control the two LEDs using GPIOs.
11	**SW3 – GPIO selective Switch**	GPIO selection switch, default ON, see Table 21.
12	**SW4 – GPIO selective Switch**	GPIO selection switch, default OFF, see Table 21.
13	**SW5 – GPIO selective Switch**	GPIO selection switch, default ON, see Table 21.
14	**SW6 – GPIO selective Switch**	GPIO selection switch, default OFF, see Table 21.
15	**SW7 – GPIO selective Switch**	GPIO selection switch, default OFF, see Table 21.
16	**SW8 – GPIO selective Switch**	GPIO selection switch, default ON, see Table 21.
17	**SW9 – Switch to test current consumption**	Switch to test current consumption using a pro board kit, default OFF.
18	**SW10 – Switch to control the SPDT**	Switch to control RF switch in DA16600MOD at test mode, default OFF.
19	**S2 – Factory Reset Button**	Factory reset button using GPIOA7. To enable this button, set Pin 2 of SW7 to ON and press the button for more than 5 seconds to initialize `nvram` data.
20	**S1 – WPS Button**	WPS button using GPIOA6. To enable this button, set Pin 1 of SW7 to ON and press the button to start WPS mode.
21	**S3 – DA14351 Reset Button**	Reset button of DA14531 in test mode.
22	**J2 – External pin-out**	GPIO connector.  J2 [1,2] P0_2/SWCLK 1 3 [1,2] GPIO_2 5 [1,2] GPIO_3 7 [1] RTC_SENSOR_OUT 9 [1] GPIO_11 11 [1] F_CLK 13 2 VBAT_BLE [1,2,3] 4 VDD_DIO1 [1,2,3] 6 VBAT_WIFI [1,2,3] 8 VDD_DIO2 [1,2,3] 10 GPIOC_8 [1] 12 GPIOC_7 [1,2] 14 GPIOC_6 [1,2]
22	**J3 – External pin-out**	GPIO connector.  J3 [1,2] P0_10/SWDIO 1 [1,2] P0_5/WLAN_ACT 3 [1] P0_7/BT_PRIO 5 [1,2] P0_8/DEBUG 7 [1,2] P0_9/DEBUG 9 [1,2] P0_6/BT_ACT 11 2 P0_11 [1] 4 <DA16200 External SPI Master or Slave > 6 GPIO_6 [1,2] SPI_CSB 8 GPIO_7 [1,2] SPI_CLK 10 GPIO_8/BT_SIG_0/WLAN_ACT [1,2] SPI_MOSI / MISO 12 GPIO_9/BT_SIG_1/BT_ACT [1,2] SPI_MISO / MOSI P0_6 (Pin #11) should be connected to GPIOA9 (Pin #12) or GPIOA10 (Pin #14) externally for BLE-Wi-Fi COEX.

DA16600 EVB includes several switches, see description for each switch below.

Table 21: DA16600 EVB Switch Description

Switch	Pin	Relevant pin		ON	OFF	Default
SW3	1	DA16200	UART0_RXD	Connects FT232H to UART0_RXD	NC	ON
	2		UART0_TXD	Connects FT232H to UART0_TXD		
SW4	1	DA14531	P0_8	Connects FT2232H to P0_8 (UART_RXD)	External pin-out (J2/J3) only	OFF
	2		P0_9	Connects FT2232H to P0_9 (UART_TXD)		
	3		P0_2	Enables Keil toolchain debug of DA14531 (SWCLK)		
	4		P0_10	Enables Keil toolchain debug of DA14531 (SWDIO)		
SW5	1	DA16200	GPIOC6	Connects FT2232H to UART2_TXD	External pin-out (J2/J3) only	ON
	2		GPIOC7	Connects FT2232H to UART2_RXD		
SW6	1	DA14531	P0_5	Connect FT2232H to P0_5 (1-wire UART)	External pin-out (J2/J3) only	OFF
	2			Connect FT2232H to P0_5 (1-wire UART)		
SW7	1	DA16200	GPIOA6	Not available on DA16600 EVB V5.0	External pin-out (J2/J3) only	OFF
	2		GPIOA7			
	3		GPIOA8			
	4		GPIOA9			
SW8	1	DA16200	GPIOA6	WPS	External pin-out (J2/J3) only	ON
	2		GPIOA7	Factory Reset		
SW9	1	DA16200	Measuring current consumption	Enables to measure current consumption of DA16200 with EVK Pro. Need to remove jumper P2.	Normal operation / Need to connect P2	OFF
	2	DA14531	Measuring current consumption	Enables to measure current consumption of DA14531 with EVK Pro. Need to remove jumper P1.	Normal operation / Need to connect P1	
SW10	1	DA14531	P0_6	Manual control of the internal RF SPDT. Pin1 OFF & Pin2 OFF: internally controlled. Pin1 ON & Pin2 OFF: DA14531 RF path ON Pin1 OFF & Pin2 ON: DA16200 RF path ON		OFF
	2					

The current consumption can be measured at jumpers P1 and P2 with current measuring equipment. See Section 6.5 for details on the test setup.

When DA16600 EVK Pro is used for measuring current consumption, remove jumpers P1 and P2 and set pins 1 and 2 of SW9 to the ON position.

To measure the current consumption of DA16200, remove the jumper on P2 and turn ON pin1 of SW9. To measure the current consumption of DA14531, remove the jumper on P1 and turn ON pin2 of SW9. The current consumption of the DA16200 and the DA14531 can be measured simultaneously.

For more information on the DA16600 EVB, see the DA16600 DEVKT Electric Schematic Ref. [10].

A.4 DA16600 EVB Ver 4.0

Figure 66 shows the hardware configuration of the DA16600 EVB.

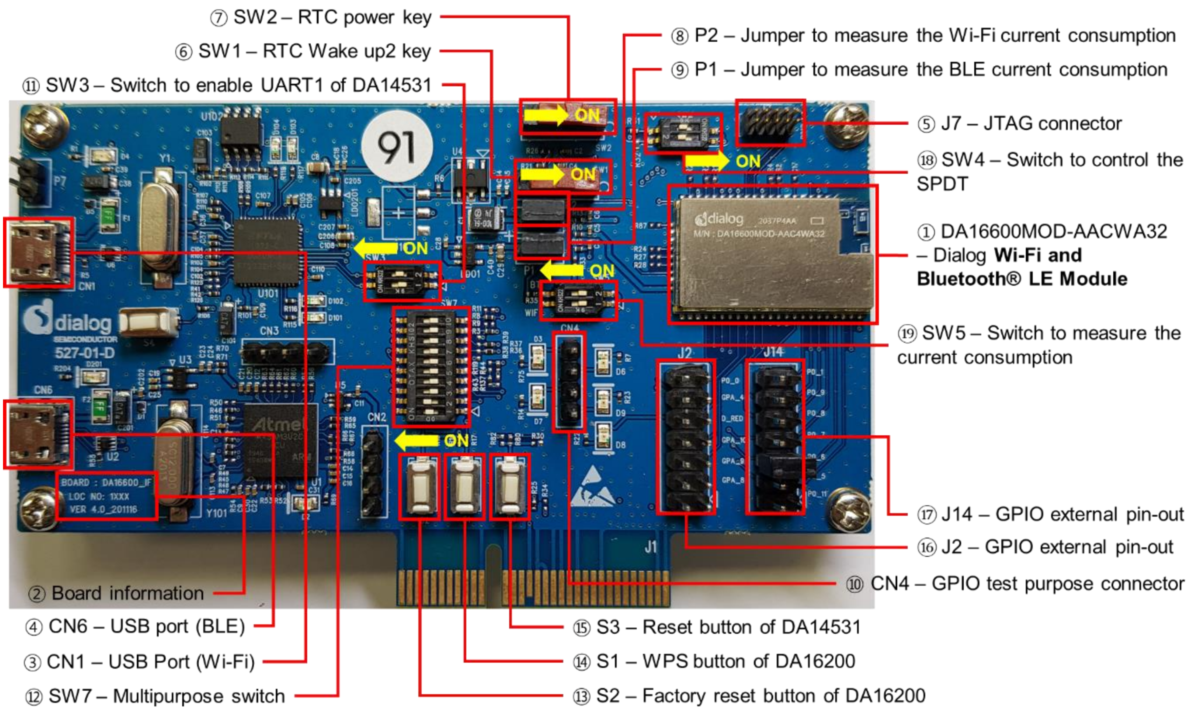
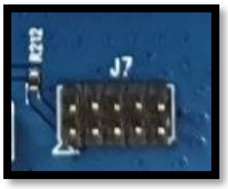
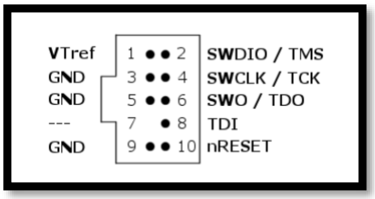
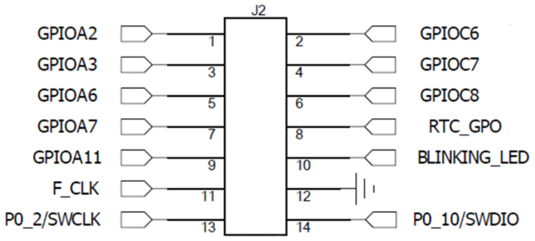
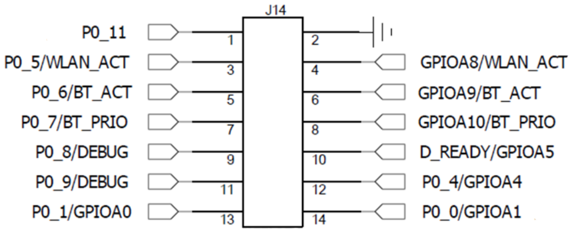

⑦ SW2 – RTC power key
⑥ SW1 – RTC Wake up2 key
⑪ SW3 – Switch to enable UART1 of DA14531
⑧ P2 – Jumper to measure the Wi-Fi current consumption
⑨ P1 – Jumper to measure the BLE current consumption
⑤ J7 – JTAG connector
⑱ SW4 – Switch to control the SPDT
① DA16600MOD-AACWA32 – Dialog Wi-Fi and Bluetooth® LE Module
⑲ SW5 – Switch to measure the current consumption
⑰ J14 – GPIO external pin-out
⑯ J2 – GPIO external pin-out
⑩ CN4 – GPIO test purpose connector
② Board information
④ CN6 – USB port (BLE)
③ CN1 – USB Port (Wi-Fi)
⑫ SW7 – Multipurpose switch
⑮ S3 – Reset button of DA14531
⑭ S1 – WPS button of DA16200
⑬ S2 – Factory reset button of DA16200
ON
ON
ON
ON
ON
ON

Figure 66: DA16600 EVB Ver 4.0 Hardware Configuration

DA16600 EVB Ver 5.0 has the following components:

Table 22: Components on DA16600 EVB

ID	Name	Description
1	**DA16600MOD-AACWA32**	Renesas **Wi-Fi and Bluetooth® LE Module.**
2	**Board information**	Board version and manufacturing date
3	**USB Port (Wi-Fi) CN1**	Provides UART0 for debug and UART1 for test.
4	**USB Port (BLE) CN6**	Connect directly to DA14531 for debug only. Note: Do not use this port during normal operation.
5	**JTAG Connector J7**	Connector for IARs I-jet JTAG Debugger   VTref 1 ●● 2 SWDIO / TMS GND 3 ●● 4 SWCLK / TCK GND 5 ●● 6 SWO / TDO --- 7 ● 8 TDI GND 9 ●● 10 nRESET Note: Pin 7 on the I-Jet debugger cable is keyed with a white plug so pin 7 must be removed from the EVB.
6	**RTC Wake up2 key SW1**	Switch to wake up the board from sleep mode.
7	**RTC Power key SW2**	Switch to turn the EVB on and off.
8	**Jumper P2**	Jumper to measure current used by the Wi-Fi device. For normal operation, this jumper must be shorted.
9	**Jumper P1**	Jumper to measure current used by the Bluetooth® LE device. For normal operation, this jumper must be shorted.

ID	Name	Description
10	**Connector CN4**	GPIO test connector. Add jumpers from J2/J4 to CN4 to control the four LEDs using GPIOs.
11	**Switch SW3**	Switch to connect directly to DA14531 and use UART to check Bluetooth® LE performance. Set this switch to off for normal operation.
12	**Switch SW7**	Multipurpose switch. Set this switch to off for normal operation.
13	**S2 – Factory Reset Button**	Factory reset button of DA16200 using GPIO_7. To enable this button, set Pin 2 of SW7 to on.
14	**S1 – WPS Button**	WPS button of DA16200 using GPIO_6. To enable this button, set Pin 1 of SW7 to on.
15	**S3 – DA14351 Reset Button**	Reset button of DA14531 in test mode.
16	**Connector J2**	GPIO connector.  J2 GPIOA2 GPIOA3 GPIOA6 GPIOA7 GPIOA11 F_CLK P0_2/SWCLK 1 3 5 7 9 11 13 2 4 6 8 10 12 14 GPIOC6 GPIOC7 GPIOC8 RTC_GPO BLINKING_LED P0_10/SWDIO
17	**Connector J14**	GPIO connector.  J14 P0_11 P0_5/WLAN_ACT P0_6/BT_ACT P0_7/BT_PRIO P0_8/DEBUG P0_9/DEBUG P0_1/GPIOA0 1 3 5 7 9 11 13 2 4 6 8 10 12 14 GPIOA8/WLAN_ACT GPIOA9/BT_ACT GPIOA10/BT_PRIO D_READY/GPIOA5 P0_4/GPIOA4 P0_0/GPIOA1
18	**Switch SW4**	Switch to control RF switch in DA16600MOD at test mode.
19	**Switch SW5**	Switch to check current consumption using a power meter kit.

DA16600MOD EVB includes several switches, see description for each switch below.

Table 23: Description of Switches

Switch	Pin	Description	Relevant pin		Default
SW7	1	Connects WPS button S1 to GPIOA6	DA16200	GPIOA6	ON
	2	Connects Factory Reset button S2 to GPIOA7	DA16200	GPIOA7	ON
	3	Enables Keil toolchain debug of DA14531 (SWCLK)	DA14531	P0_2/SWCLK	OFF
	4	Enables Keil toolchain debug of DA14531 (SWDIO)	DA14531	P0_10/SWDIO	OFF
	5	Enables UART debug of DA14531 (UART2 RX)	DA14531	P0_8/DEBUG	OFF
	6	Enables UART debug of DA14531 (UART2 TX)	DA14531	P0_9/DEBUG	OFF
	7	Does not use	-	-	OFF
	8	Does not use	-	-	OFF

Switch	Pin	Description	Relevant pin		Default
	9	Does not use	-	-	OFF
	10	Does not use	-	-	OFF
SW3	1	Enables UART1 of DA14531 (1-wire UART)	DA14531	P0_5/WLAN_ACT	OFF
	2	Enables UART1 of DA14531 (1-wire UART)	DA14531	P0_5/WLAN_ACT	OFF
SW4	1	Manual control of the internal RF SPDT. Pin1 OFF & Pin2 OFF: internally controlled. Pin1 ON & Pin2 OFF: DA14531 RF path ON Pin1 OFF & Pin2 ON: DA16200 RF path ON	DA14531	P0_6/BT_ACT	OFF
	2				OFF
SW5	1	Measures current consumption of DA16200 with DA16600 EVK Pro	-	-	OFF
	2	Measures current consumption of DA14531 with DA16600 EVK Pro	-	-	OFF

The current consumption can be measured at jumpers P1 and P2 with current measuring equipment. See Section 6.5 for details on the test setup.

When DA16600 EVK Pro is used for measuring current consumption, remove jumpers P1 and P2 and set pins 1 and 2 of SW5 to the ON position.

To measure the current consumption of DA16200, remove the jumper on P2 and turn ON pin1 of SW5. To measure the current consumption of DA14531, remove the jumper on P1 and turn ON pin2 of SW5. The current consumption of the DA16200 and the DA14531 can be measured simultaneously. For more details on the DA16600 EVB, see the DA16600 DEVKT Electric Schematic Ref. [10].

Appendix B DA16200 Debug Interface Commands

The DA16200 has various console commands to operate its functions. The UART0 interface connects the console with a serial terminal tool. Some commands in the following sections may be disabled according to the SDK's features configuration.

B.1 Console Commands

The DA16200 console commands are categorized as follows:

- root
  - `[/DA16200] #`
- mem
  - `[/DA16200/MEM] #`
- sys
  - `[/DA16200/SYS] #`
- nvram
  - `[/DA16200/NVRAM] #`
- net
  - `[/DA16200/NET] #`
- user
  - `[/DA16200/USER] #`

Use command `help or ? (Question mark)` to list the available commands and options.

There is a function to display the console command history, and up to 5 commands can be saved. Use the following keys and characters to access the history function:

- `↑ or ↓ (arrow key)` on your keyboard: show the command history one by one.
- `! (Exclamation mark)`: view the list of the command history.
- `! (Exclamation mark) + Number`: select and execute one previous command in the list.

It is possible to move between categories. Use these options:

- `top`: move to the highest-rank, root.
- `up`: move to one step upper rank category.
- Category command (for example `sys`, `nvram`, `net`): move to the category. To run each command of each category, go to the category first, or prefix the category name to the command as shown in the example:
  - `net`
  - `net.ifconfig`

B.1.1 Root Commands

Table 24: Root Commands

Command	Parameters	Description
`help /?`	(none)	Display help information for the corresponding category
`up`	(none)	Move up one rank category
`top`	(none)	Move to the Root category
`factory`	(none)	Factory reset for all settings

Command	Parameters	Description
ps	(none)	Display thread information
setup	(none)	DA16200 general function setting wizard (Easy Setup) Make step-by-step configuration settings for elements such as SYSMODE, WI-FI, and NETWORK
reboot	(none) [mode]	Reboot por: POR rebooting
reset	(none)	Reset to the Bootloader prompt
ver	(none)	Display SDK version & system information
time	[option]	Display or set the current time. time set [YYYY-MM-DD] [hh:mm:ss]: set date and time time zone [-hh:mm]: set time zone time boot: display booting time time uptime: display booting duration time help: display help
getwlanmac	(none)	Display the MAC address for network interfaces
setwlanmac	[xx:xx:xx:xx:xx:xx | xx-xx-xx-xx-xx-xx |xxxxxxxxxxxx]	Set up the MAC address for network interfaces. For example: setwlanmac aa:bb:cc:00:00:02 | aa-bb-cc-00-00-02 | aabbcc000002
dpm	[options]	Set DPM condition on | off: DPM feature enabled or disabled status: DPM status print rtm: view DPM backup data rtc: view DPM RTC timer debug [level]: turn DPM debug on / off level = 1(MSG_ERROR), 2(MSG_INFO), 3(MSG_DEBUG), 4(MSG_EXCESSIVE)

B.1.2 Network Commands

To move to the network command category, type the command net.

Table 25: Network Commands

Command	Parameter	Description
arp	[interface] [options]	Display the ARP table of a network interface ● a: display the ARP table of every interface ● d: delete all of ARP table Help: Help display
arpsend	[interface] [dst ipaddress]	Transmit the ARP request message of the target IP For example: arpsend wlan0 10.0.0.1
garpsend	[interface] [option]	Transmit a GARP message with option: ● 0: normal garp ● 1: check IP conflict For example: arpsend wlan0
Arping	[interface] [options]	Send ARP ping to target IP ● -I [wlan0|wlan1] : Interface name

Command	Parameter	Description
		● `-n or -c count` : Stop after sending count ARP REQUEST ● `-w timeout` : Specify a timeout, in milliseconds, before arping exists regardless of how many packets have been sent or received. (Min:10ms) ● `-i interval` : Interval in milliseconds to wait for each reply.(MIN:10ms) For example: arping 172.16.0.1 -n 10 -w 1000 -i 1000

`dhcpd`	`[interface]`
	`[options]`
		DHCP server setting (Only Soft AP mode)
		● `boot [on|off]`: automatic start setting with a certain interface
		● `range <Start IP ADDRESS> <END IP Address>`: IP lease band setting (max. 10)
		● `lease_time <Integer>`: lease time setting (min. 60 sec.)
		● `dns <IP Address>`: lease IP DNS server address setting
		● `response_delay <Integer>`: time of response delay
		● `status`: display DHCP Server status
		● `lease [0|1]`: display IP lease table
		Display tables including un-allotted tables when flag = 1

`ifconfig`	`(none)`
	`[interface wlan0|wlan1]`
	`[options]`
		Display or set the basic network setting and status
		● `ifconfig`: display basic network settings information
		● `ifconfig -a`: display details of all network interfaces
		● `ifconfig [wlan0|wlan1]`: display details of a network interface
		● `ifconfig [wlan0|wlan1] [ipaddress] [subnet] [gateway]`: set static IP addresses to a network interface
		● `ifconfig [wlan0|wlan1] dhcp`: enable/Disable DCHP to a network interface
		● `ifconfig [wlan0|wlan1] [up|down]`: go Up/Down a network interface
		● `ifconfig [wlan0|wlan1] [start|stop|renew|release]`: DHCP client command
		● `ifconfig [wlan0|wlan1] [dns] [DNS ServerIP]`: set DNS server address (static IP) to a network interface
		● `ifconfig help`: display help

`ping`	`-I [interface wlan0|wlan1] [domain|ip] -n [count] -l [size] -w [timeout] -i [interval]`	Ping test to the target address with a certain option ● `[interface wlan0|wlan1]`: ○ Network interface. With no designated interface, an interface for a subnet band of the same destination IP address is designated ● `[count]`: the count of ping tests ● `[size]`: the size of data to be transmitted (max.: 10000) ● `[timeout]`: waiting time for a response to the transmitted message (min.: 10 ms) ● `[interval]`: waiting time for a message transmission (min.: 10 ms) ● `[-6]`: ping test with an IPv6 address For example: `ping 172.16.0.1 -l 1024 -n 10 -w 1000 -i 1000` `ping -6 fe80::1:2 -I wlan0`
`sntp`	`[option]`	Show or configure for SNTP operation ● `status` : Show current SNTP configuration ● `enable | disable`

Command	Parameter	Description
		● `addr [server]` : First SNTP server address ● `addr_2 [server]` : Second SNTP server address ● `addr_3 [server]` : Third SNTP server address ● `period [second]` : Set SNTP renewal time ● `sync` : Run SNTP sync with current SNTP server

`iperf`	`-I [interface]`
	`[-s|-c host]`
	`[options]`
		Setup Iperf client/server

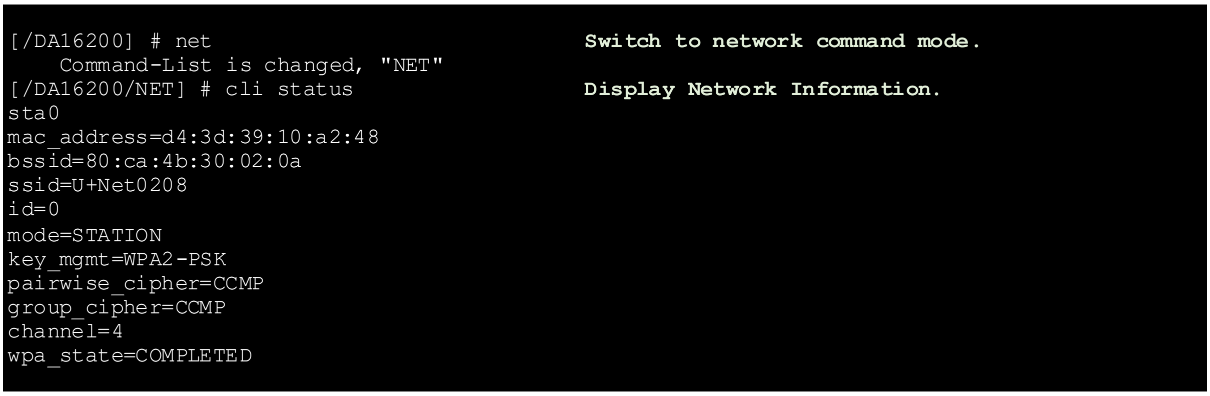
`mqtt_client`	`[options]`	Run or show the status of the MQTT Client ● `start` : start mqtt_client ● `stop` : stop mqtt_client ● `check` : check mqtt_client connection ● `-m {msg} [topic] publish {msg} with [topic]` : if [topic] is not specified the one stored in nvram is used ● `-l` : publish long message
`cli`	`[options]`	See the CLI section
`debug`	`[options]`	Execute various types of debug commands ● `arp [on|off]`: arp debug message output on/off ● `dhcpd [level]`: DHCP Server debug level setting (level=0~2 default 0) ● `dhcpc [level]`: DHCP Client debug level setting (level=0~5 default 1) ● `umac [on|off] mask`: debug umac 1 0x4
`rssi`	`[wlan0|wlan1]`	Show RSSI value for currently connected interface
`Getsysmode`	`[none]`	Show current Wi-Fi operation mode

B.2 CLI Command

B.2.1 Overview

The DA16200 supplicant plays a key role in providing users with Wi-Fi functionality. Major functions include IEEE 802.11 management frame, various security functions (WPA & RSN by IEEE 802.11i) and Command Line Interface (CLI) to control DA16200 Wi-Fi performance. The CLI in DA16200 can execute commands in the network command state.

For example, in Station Mode the network information of DA16200 can be displayed using the `cli status` command while in network command state:



```
[/DA16200] # net                              Switch to network command mode.
    Command-List is changed, "NET"
[/DA16200/NET] # cli status                   Display Network Information.
sta0
mac_address=d4:3d:39:10:a2:48
bssid=80:ca:4b:30:02:0a
ssid=U+Net0208
id=0
mode=STATION
key_mgmt=WPA2-PSK
pairwise_cipher=CCMP
group_cipher=CCMP
channel=4
wpa_state=COMPLETED
```

B.2.2 CLI Format

There are four CLI formats (Type A~D):

- Read/Write Parameter (Type A)
  - Read: `[/DA16200/NET] # cli [CLI]`
  - Write: `[/DA16200/NET] # cli [CLI] <VALUE>`
- Write Only Parameter (Type B)
  - `[/DA16200/NET] # cli [CLI] <VALUE>` or `cli [CLI] <OPTION> <VALUE>`
- Read Only Parameter (Type C)
  - `[/DA16200/NET] # cli [CLI]` or `cli [CLI] <OPTION>`
- Execution Parameter (Type D)
  - `[/DA16200/NET] # cli [CLI]` or `cli [CLI] < OPTION>`

B.2.3 Common Commands

Table 26: CLI Commands in Common Mode

CLI	Parameter	Description
status	(none)	Get the main information on the interface being operated at DA16200 For example: `[/DA16200/NET] # cli status`

save_config	(none)	Save all parameters modified through CLI, etc. in NVRAM
		(Saved values become applicable after a reboot) (D)
		For example: `[/DA16200/NET] # cli save_config`
		* Information saved in NVRAM may be inquired with the following command:
		For example:
		`[/DA16200/NVRAM] # printenv`
		`Total length (95)`
		`country_code (STR,03) ........ KR`
		`SYSMODE (STR,02) ............ 0`
		`0: NETMODE (STR,02) ........... 1`
		`N0_Profile (STR,02) .......... 1`
		`N0_ssid (STR,16) ............ "ACST_AC_TEST1"`

select_network	<mode>	Execute a motion in a certain mode (STA access, AP operation, etc.) (D) ● `<mode>` STA: 0 | AP: 1 For example: `[/DA16200/NET] # cli select_network 0` ○ Implement STA access * For a certain mode through the `select_network` CLI, the following tasks need to be carried out first: ○ `add_network` (profile generation) ○ SSID generation through `set_network` ○ For AP operation, set up the frequency and country code values with command `set_network` ○ For Security, generate WPA or WEP key values with command `set_network` (optional)
add_network	<mode>	Generate a specific mode (STA, AP) Profile (access information table) (D) `<mode>`: 0(STA) | 1(AP) For example: `[/DA16200/NET] # cli add_network 1` ○ Generate a profile for AP Mode
remove_network	<mode>	Delete a certain mode (STA, AP) profile (D) `<mode>`: 0(STA) | 1(AP)

CLI	Parameter	Description
		For example: [/DA16200/NET] # cli remove_network 1 ○ Delete a profile for AP Mode
set_network	<mode> <variable> <value>	Set parameter values for a specific mode (STA, AP) (B) <mode>: 0(STA) | 1(AP) <variable>: a specific parameter ● ssid: [STA] Operation SSID for AP SSID / [AP] AP interface to be connected ● psk: passphrase or PSK values ● proto: for WPA use, set up the version (<WPA(=WPA1)> | <RSN(=WPA2)> | <WPA RSN>) ● key_mgmt: key management mode (<NONE> | <WPA_PSK> | <WPA-EAP>) ● pairwise: unicast data message encryption mode (<TKIP> | <CCMP> | <TKIP CCMP>) ● group: broadcast data message encryption mode (<TKIP> | <CCMP> | <TKIP CCMP>) ● wep_key#: WEP key (#:0~3) values ● wep_tx_keyidx: WEP key index to be used ● frequency: [AP] Operation Frequency (MHz) ● mode: Operation Mode <0(STA)> | <2(AP)> ● Wi-Fi_mode: <0(BGN)> | <1(GN)> | <2(BG)> | <3(N)> | <4(G)> | <5(B)> ● beacon_int: [AP] Beacon transport interval ● dtim_period: [AP] DTIM interval ● ap_power: [AP] Output Power (dBm) ● isolate: 'Isolate' Use (<0(off)> | <1(on)>) ● -disabled: Prevent automatic profiling on reboot (<0(off)> | <1(on)>) <value>: settings for a certain variable For example: [/DA16200/NET] # cli set_network 1 ssid 'DA16200_AP' ○ For DA16200 AP operation, SSID= DA16200_AP setting For example: [/DA16200/NET] # cli set_network 1 beacon_int 200 ○ For DA16200 AP operation, Beacon interval 20 ms setting For example: [/DA16200/NET] # cli set_network 0 key_mgmt WPA_PSK ○ For DA16200 STA operation, access in the WPA PSK security mode * A profile needs to be generated with command add_network so that a profile can be set with command set_network (with no profile, 'FAIL')
get_network	<mode> <variable>	Get specific parameter values for a specific mode (STA, AP) (C) <mode>: 0(STA) | 1(AP) <variable>: a specific parameter For example: [/DA16200/NET] # cli set_network 0 ssid ○ Inquiry of an object subject to DA16200 STA access ("TEST_BED_AP") For example: [/DA16200/NET] # cli set_network 1 psk ○ For DA16200 AP operation, inquiry of the PSK password setting
country	<value>	Set a country related to channel operation (A) <value>: Country Code that meets ISO 3166-1 alpha-2 standards Default: KR For example: [/DA16200/NET] # cli country US ○ Set the Country Code to US For example: [/DA16200/NET] # cli country

CLI	Parameter	Description
		○ KR
flush	(none)	For every interface (STA, AP), DA16200 deletes the Profile and closes DA16200 service operation (D)

B.2.4 STA Commands

Table 27: CLI Commands on STA Mode

Command	Parameters	Description
scan	(none) or <freq>	Active scanning (Probe Request Broadcast) (D) For <freq> inputs, it is possible to scan APs of a certain frequency range (MHz) only (optional) For example: `[/DA16200/NET] # cli scan` ○ Scans all channels that correspond to the current country setting
disconnect	(none)	Disconnect the accessed AP (D) For example: `[/DA16200/NET] # cli disconnect` ○ OK (With no AP being accessed, 'FAIL')
roam	(none) or <oper>	Roaming On/Off and Roaming status inquiry (A) <oper> run: On | stop: Off Default: Roaming Off For example: `[/DA16200/NET] # cli roam` ○ Roaming=STOP, Threshold=-65 ○ Usage: cli roam [run/stop] For example: `[/DA16200/NET] # cli roam stop` ○ Roaming function-off
roam_threshold	<value>	Roaming triggering RSSI value (dBm) setting (B) <value>: Roaming threshold RSSI (dBm) Default: -65 (dBm) For example: `[/DA16200/NET] # cli roam_threshold -85` ○ Set the roaming threshold to -85 dBm

B.2.5 Soft AP Commands

Table 28: CLI Commands on Soft AP Mode

Command	Parameter	Description
ap	<option>	AP interface beginning/closing/restarting (Applicable with no reboot after main info. modification of AP interface SSID, PSK, etc.) (D) <option>: start | stop | restart For example: `[/DA16200/NET] # cli ap start` ○ AP interface initiating (If it is being operated, 'FAIL') For example: `[/DA16200/NET] # cli ap stop` ○ AP interface closing (If not being operated, 'FAIL') For example: `[/DA16200/NET] # cli set_network 1 ssid 'DA16200_AP2'` For example: `[/DA16200/NET] # cli ap restart` ○ Modify SSID of the interface of AP being operated For example: `[/DA16200/NET] # cli set_network 1 pairwise TKIP`

Command	Parameter	Description
		For example: `[/DA16200/NET] # cli ap restart` ○ Modify the AP interface encryption mode to TKIP
`ap_chan_switch`	`<Ch.>`	Modify the AP interface operation channel (B) `<Ch.>`: AP operation channel (1~13) or frequency (MHz) For example: `[/DA16200/NET] # cli ap_chan_switch 3` ○ Modify the AP interface channel to 3 (242 MHz) For example: `[/DA16200/NET] # cli ap_chan_switch 11 2462` ○ Modify the AP interface channel to 11 (2462 MHz)

`ap_status`	(none)	Get main information about the interface at DA16200 (C)
		For example:
		`[/DA16200/NET] # cli ap_status`
		`state=ENABLED`
		`phy=fc9k_phy0`
		`freq=2472`
		`num_sta_non_erp=0`
		`num_sta_no_short_slot_time=0`
		`num_sta_no_short_preamble=0`
		`olbc=0`
		`num_sta_ht_no_gf=0`
		`num_sta_no_ht=0`
		`num_sta_ht_20_mhz=0`
		`num_sta_ht40_intolerant=0`
		`olbc_ht=0`
		`ht_op_mode=0x0`
		`cac_time_seconds=0`
		`cac_time_l`

`all_sta`	(none)	Output the list information of STA being accessed to the AP interface (C)
		For example:
		`[/DA16200/NET] # cli all_sta`
		`50:77:05:DB:C4:3E`
		`flags=[AUTH][ASSOC][AUTHORIZED][SHORT_PREAMBLE][WMM`
		`aid=1`
		`capability=0x431`
		`listen_interval=10`
		`mode = 802.11n`
		`timeout_next=0`
		`rx_packets=632`
		`tx_packets=9`
		`rx_bytes=67451`
		`tx_bytes=4767`
		`connected_time=77`
		`sta_count=1`

`deauthenticate`	`<addr>`	The `deauthenticate` message is transmitted to the access STA with a certain MAC address to cancel the access (D) `<addr>`: MAC address of the access STA For example: `[/DA16200/NET] # cli deauthenticate aa:ff:01:00:00:00` ○ Transmit the de-authentication message to STA whose MAC address is AA:FF:01:00:00:00
`disassociate`	`<addr>`	The disassociation message is transmitted to the access STA with a certain MAC address to cancel the access (D) `<addr>`: MAC address of the access STA For example: `[/DA16200/NET] # cli disassociate aa:ff:01:00:00:00`

Command	Parameter	Description
		○ Transmit the disassociation message to STA whose MAC address is `AA:FF:01:00:00:00`
`wmm_enabled`	`<value>`	WMM function availability setting and inquiry (A) `<value>`: On: 1 | Off: 0 Default: Off For example: `[/DA16200/NET] # cli wmm_enabled 1` ○ Use the WMM function
`wmm_ps_enabled`	`<value>`	WMM-PS function availability setting and inquiry (A) `<value>`: On: 1 | Off: 0 Default: Off For example: `[/DA16200/NET] # cli wmm_ps_enabled 1` ○ Use the WMM-PS function

`wmm_params`	`<target>`
	`<category>`
	`<AIFS>`
	`<CWmin>`
	`<CWmax>`
	`<Burst (AP) or TxOP Limit (STA) >`
		Set up details of DA16200 AP or STA's certain category WMM parameters (B)
		`<target>`: ap | sta
		`<category>`: be(best-effort) | bk(background) | vi(video) | vo(voice)
		For example: `[/DA16200/NET] # cli wmm_params ap be 3 15 63 10`
		○ For WMM AP's best-effort category, AIFS=3, CWmin=15, CWmax=63, and Burst=10
		For example: `[/DA16200/NET] # cli wmm_params sta vo 4 7 15 60`
		○ For WMM STA's voice category, AIFS=4, CWmin=7, CWmax=15, TXOP_Limit=60

`all_wmm`	`(none)`	Inquiry of all parameters that can be set up by means of wmm_params CLI (C) (See example) For example: `[/DA16200/NET] # cli all_wmm`
`acl_mac`	`<addr>`	Add the MAC address to the Access Control Management List (B) `<addr>`: AP MAC Address For example: `[/DA16200/NET] # cli acl_mac AA:FF:01:00:00:06` ○ Add MAC address AA:FF:01:00:00:06 to ACL

`Acl`	`<oper>`
	`<addr>`
		Set up, delete, or inquire the use of ACL (A)
		`<oper>`: allow | deny | clear | delete (If none, inquire of it)
		`<addr>`: AP MAC Address (only when oper=”delete”)
		`[/DA16200/NET] # cli acl [allow/deny/clear/delete mac_address]`
		For example: `[/DA16200/NET] # cli acl`
		For example: `[/DA16200/NET] # cli acl allow`
		○ Access allowed only for AP Lists in ACL
		For example: `[/DA16200/NET] # cli acl deny`
		○ Access denied only for AP Lists in ACL
		For example: `[/DA16200/NET] # cli acl clear`
		○ Entire ACL clear
		For example: `[/DA16200/NET] # cli delete aa:ff:01:00:00:08`
		○ Delete AA:FF:01:00:00:08 from ACL

Command	Parameter	Description
`ap_max_inactivity`	`<value>`	If there is no data frame exchange of accessed STA during the time setting, disconnect the STA (A) `<value>`: inactivity timeout (sec) Default: 300 (sec.) For example: `[/DA16200/NET] # cli ap_max_inactivity 600` ○ Disconnect the access STA with no data frame exchange for 600 seconds For example: `[/DA16200/NET] # cli ap_max_inactivity 0` ○ Uncheck data frame exchange of the accessed STA For example: `[/DA16200/NET] # cli ap_max_inactivity` ○ Read `ap_max_inactivity` value
`ap_send_ka`	`<value>`	A function to send 'keep-alive' NULL packets to the accessed STA at intervals of 30 seconds and check ACK receipts (A) `<value>`: On: 1 | Off: 0 ● `On`: if the STA accessed to DA16200 AP interface is out of coverage or closed abnormally, disconnection occurs after the ‘ap_max_inactivity timeout’ passes ● `Off`: if there is no constant data frame exchange with the STA accessed to DA16200 AP interface for `ap_max_inactivity timeout`, then disconnect. Default: 0 (not used) For example: `[/DA16200/NET] # cli ap_send_ka 1` ○ ap_send_ka=1
`ap_rts`	`<value>`	For AP mode operation, set up the RTS Threshold value to be used (A) `<value>`: The standard for a size of packets that use the RTS Control Frame (bytes) Default: 2437 (bytes) For example: `[/DA16200/NET] # cli ap_rts 1000` ○ Use RTS for transmission of 1000 bytes or larger frames ○ ap_rts=1000 For example: `[/DA16200/NET] # cli ap_rts` ○ ap_rts=2437
`greenfield`	`<value>`	Enable/Disable use of Greenfield `<value>`: On: 1 | Off: 0 If Greenfield is on, DA16200 uses 11n HT mode only. In that case, 11b, 11g info. and STA access are not allowed Default: 0 (not used) For example: `[/DA16200/NET] # cli greenfield 1` ○ Use the Greenfield function ○ greenfield=1 For example: `[/DA16200/NET] # cli greenfield` ○ greenfield=0

B.2.6 Advanced Commands

Table 29: Advanced CLI Commands

Command	Parameters	Description
wps_pbc	(none)	Run WPS Push Button Configuration (PBC)
wps_pin	<pin>	Run WPS PIN method <pin> pin code (any: generate a random code) For example: [/DA16200/NET] # cli wps_pin 27833513 For example: [/DA16200/NET] # cli wps_pin any

B.3 MROM Commands

B.3.1 Common Commands

Table 30: Common Commands in MROM

Command	Parameters	Description
!	(none)	None
reboot	(none) [mode]	Reboot por: POR rebooting
reset	(none)	Reset to the Bootloader prompt
ver	(none)	Display version of MaskRom
help /?	(none)	Display help information for the corresponding category
boot	[address]	Booting address: booting address

B.3.2 Memory Access Commands

Table 31: Memory Access Commands in MROM

Command	Parameters	Description
brd	[addr] [length]	Byte read memory
bwr	[addr] [data] [length]	Byte write memory
wrd	[addr] [length]	Word read memory
wwr	[addr] [data] [length]	Word write memory
lrd	[addr] [length]	Long read memory
lwr	[addr] [data] [length]	Long write memory

B.3.3 Download Commands

Table 32: Download Commands in MROM

Command	Parameters	Description
loady	[addr] [sector_size] [format] boot	Download image to SFLASH boot : '0' address
ymodem	[addr] [size] sfdp	Download image to RAM sfdp : download sfdp to 0xf80040 (retention memory)

B.3.4 SFLASH Commands

Table 33: SFLASH Commands in MROM

Command	Parameters	Description
sflash	[op] [start] [length]	Sflash access op: read, write, erase start: start address length: length

B.3.5 Secure Commands

Table 34: Secure Commands in MROM

Command	Parameters	Description
sbrom	[target] [bootaddress] [debug]	Secure boot command target : sflash boot address : 0 (boot) or a000 (rtos) debug : anything
socid	(none)	Display socid (secure soc id)

B.3.6 Additional Commands

Table 35: Additional Commands in MROM

Command	Parameters	Description
bmcfg	[op] [bootmode]	Boot mode config Not used
oops	[op] [value]…	Oops configuration Not used
dbgt	[op] [op] [value0]…	Debug mode configuration Not used
floady	[addr] [sector_size] [format] [boot]	Fast download image to sflash Not used
rs485	[offset] [id]	Download image with rs485 Not used
clock	[new clock]	System clock setting
setsfl	[mode]	Select sflash Mode : 0 (stacked sflash), 1 (external sflash) Used '1'only

Appendix C SDK Memory Maps

C.1 Introduction

The DA16200/DA16600 SDK provide two kinds of memory maps: an external SFLASH device and an internal 512 KB SRAM.

This document describes the memory maps for the external SFLASH device and the internal SRAM. Also, the document describes how to change the SFLASH device and adjust the SRAM memory map used in the SDK.

NOTE

The DA16200/DA16600 SDK currently supports the following SFLASH types:

- **4-MB SFLASH**

: Adesto	- AT25SL321
: FMSC	- FM25W32
: GigaDevice	- GD25LE32E
: ISSI	- IS25LQ032B
: MXIC	- MX25L3233F, MX25L25635F, MX25R3235F, MX25U3232F, MX25U3235F
: Winbond	- W25Q32JV, W25Q32JW

To use a different type of SFLASH with the DA16200/DA16600, contact Renesas Electronics to confirm compatibility.

C.2 SFLASH Memory Map

The DA16200 and the DA16600 FreeRTOS SDK support 4 MB SFLASH only.

C.2.1 DA16200

Table 36: 4 MB SFLASH Map for DA16200

Address	Name		Size (kB)
0x0000_0000	2nd Bootloader		136
0x0002_2000	Boot Index		4
0x0002_3000	RTOS #0		1788
0x001E_2000	RTOS #1		1788
0x003A_1000	Reserved Area		4
0x003A_2000	Debug / RMA Certificate		4
0x003A_3000	TLS Certificate #0 (MQTT)	CA	4
0x003A_4000		Cert	4
0x003A_5000		Private key	4
0x003A_6000		Diffie-Hellmann key	4
0x003A_7000	TLS Certificate #1 (HTTPs / OTA)	CA	4
0x003A_8000		Cert	4
0x003A_9000		Private key	4
0x003A_A000		Diffie-Hellmann key	4
0x003A_B000	NVRAM #0		4
0x003A_C000	NVRAM #1 (Backup)		4
0x003A_D000	User Area		256
0x003E_D000	TLS Certificate Key #2 (WPA Enterprise)	CA	4
0x003E_E000		Cert	4
0x003E_F000		Private	4
0x003F_0000		Diffie-Hellmann key	4
0x003F_1000	TLS Certificate Key #3 (Reserved)	CA	4
0x003F_2000		Certificate	4
0x003F_3000		Private Key	4
0x003F_4000		Diffie-Hellmann key	4
0x003F_5000	NVRAM FOOTPRINT		4

Address	Name	Size (kB)
0x003F_6000	AT-CMD TLS Certificate Key #0 ~ #9 $^{(*)}$	40

NOTE
See the Secure Socket Command section of Ref. [5] for usage of “AT-CMD TLS Certificate Key #0 ~ #9” area.

C.2.2 DA16600

Table 37: 4 MB SFLASH Map for DA16600

Address	Name		Size (kB)
0x0000_0000	2^{nd} Bootloader		136
0x0002_2000	Boot Index		4
0x0002_3000	RTOS #0		1788
0x001E_2000	RTOS #1		1788
0x003A_1000	Reserved Area		4
0x003A_2000	Debug / RMA Certificate		4
0x003A_3000	TLS Certificate #0 (MQTT)	CA	4
0x003A_4000		Cert	4
0x003A_5000		Private key	4
0x003A_6000		Diffie-Hellmann key	4
0x003A_7000	TLS Certificate #1 (HTTPs / OTA)	CA	4
0x003A_8000		Cert	4
0x003A_9000		Private key	4
0x003A_A000		Diffie-Hellmann key	4
0x003A_B000	NVRAM #0		4
0x003A_C000	NVRAM #1 (Backup)		4
0x003A_D000	Bluetooth® LE Firmware area (Depends on BLE_IMG_SIZE, default: 64kB)		64 ~ 80
0x003B_D000 or 0x003C_1000	Bluetooth® LE Security DB area (Depends on __MULTI_BONDING_SUPPORT__, default: 0 kB)		0 or 4
0x003B_D000 ~ 0x003C_2000	User Area (Start address: 0x003B_D000, default: 192 kB)		172 ~ 192
0x003E_D000	TLS Certificate Key #2 (WPA Enterprise)	CA	4
0x003E_E000		Cert	4
0x003E_F000		Private	4
0x003F_0000		Diffie-Hellmann key	4
0x003F_1000	TLS Certificate Key #3 (Reserved)	CA	4
0x003F_2000		Certificate	4
0x003F_3000		Private Key	4
0x003F_4000		Diffie-Hellmann key	4
0x003F_5000	NVRAM FOOTPRINT		4
0x003F_6000	AT-CMD TLS Certificate Key #0 ~ #9 $^{(*)}$		40

NOTE
See the Secure Socket Command section of Ref. [5] for usage of “AT-CMD TLS Certificate Key #0 ~ #9” area.

The DA16200/DA16600 SDK contains a script to automatically select the SFLASH type. See section 5.5.2 for details on how the SFLASH is selected in e^{2}studio .

To change the SFLASH type without using the script, go to the `<sdk_root_directory>/tools/SBOOT/config` directory and copy the `da16xtpmconfig.cfg.xxxx(4MB)` file that matches the SFLASH device that is being used and copy it to `da16xtpmconfig.cfg`.

C.3 SRAM Memory Map

In the FreeRTOS SDK, the overall memory map is organized as shown in Figure 67.

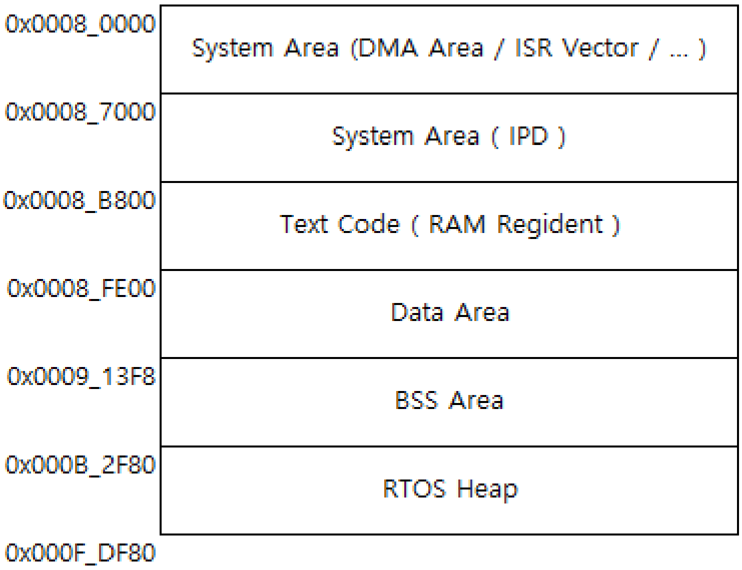


Figure 67: System Memory Map

Appendix D Troubleshooting

D.1 Boot Failure

During the development process, there may be times when there is an error in the code or a problem programming the firmware which could cause the DA16200/DA16600 to fail to boot properly.

This may happen because the Serial Flash Discoverable Parameters (SFDP) which are stored in Retention Memory become corrupt or erased. The DA16200/DA16600 requires the SFDP for the SFLASH to operate properly. The SFDP parameters are included in the FBOOT firmware image and can be recovered by programming the firmware again.

D.1.1 Check SFDP

Switch to [MROM] mode and use the `brd f80000 100` command to check if the SFPD information:

```
[/DA16200] # brd f80000 100

[00F80000] : 00 00 00 01 FF FF FF FF FF FF FF FF 00 00 00 40      ...............@
[00F80010] : A1 00 91 FC 00 00 00 00 00 00 00 00 00 00 00 00      ................
[00F80020] : 0B 00 00 00 C0 D4 01 80 00 00 00 00 89 34 6F 93      .............4o.
[00F80030] : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 64      ...............d
[00F80040] : 53 46 44 50 15 16 60 EF 00 00 40 00 F5 20 F1 FF      SFDP..`...@.. ..
[00F80050] : FF FF FF 01 44 EB 08 6B 08 3B 04 BB EE FF FF FF      ....D..k.;......
[00F80060] : FF FF 00 FF FF FF 44 EB 0C 20 0F 52 10 D8 00 FF      ......D.. .R....
[00F80070] : 11 34 1E F3 83 3A 00 CC CC 43 1C 62 7A 75 7A 75      .4...:...C.bzuzu
[00F80080] : 07 CD D5 5C 42 00 EF 40 FF 7F 00 01 07 02 36 39      ...\B..@......69
[00F80090] : 3C FF E1 E0 FF 00 00 00 32 68 68 68 32 64 00 04      <.......2hhh2d..
[00F800A0] : 00 04 00 00 00 00 00 00 90 9F AB 5A 05 01 04 06      ...........Z....
[00F800B0] : 50 66 99 00 00 B9 AB 38 FF B7 E9 00 00 02 32 00      Pf.....8......2.
[00F800C0] : 02 32 00 20 00 0B 60 35 00 00 00 00 00 00 00 00      .2. ..`5........
[00F800D0] : 00 00 00 00 28 50 78 A0 40 40 40 80 00 00 00 00      ....(Px.@@@.....
[00F800E0] : 00 00 00 00 FF FF FF FF FF FF FF FF FF FF FF FF      ................
[00F800F0] : AD B6 F3 D8 00 00 00 00 00 00 00 00 00 00 00 00      ................

[/DA16200] #
```

If the SFDP information is not similar to the above, then the both of FBOOT and FRTOS image must be programmed again without reset and reboot.

D.1.2 Recovery from Unresponsive Boot

If the DA16200/DA16600 does not boot and the serial command interface is unresponsive, then access can be recovered by following steps:

1. Copy uart_program_da16200 to the folder which has FBOOT and FRTOS images.
2. Power off the DA16200/DA16600.
3. Run the uart_program_da16200 and select **number** in list of serial interface.
4. Input `emode`.
5. Follow the instructions from uart_program_da16200.
6. Confirm the devices enter MROM state.
7. Then the FBOOT and FRTOS images can be programmed manually as follows.
8. Input `dload`.
9. Input **address** for FBOOT image. (0 at default image)
10. Select **number** of FBOOT image in lists which shows all images in the folder.
11. Programming is done automatically.
12. Input **y** for more programming.
13. Input **address** for FRTOS image. (23000 at default image)
14. Select **number** of FRTOS image in lists which shows all images in the folder.
15. Programming is done automatically.
16. Input **n** because no more programming is required.

17. Input **reboot** and confirm the programmed image works.

```
d:\download>uart_program_da16200.exe
uart_program_da16200 Version 1.0.5
0. COM75 - USB Serial Port (COM75)
1. COM76 - USB Serial Port (COM76)
Please enter number in the list of your COM port and click enter.
--> 0
Selected COM port =  COM75 - USB Serial Port (COM75)
Entering console mode...
Ready for console mode. Input anything.

2023-07-31 16:49:42.644:[/DA16200] #  emode
Entering emergency mode...
Confirm turn off device. Then enter Y and click Enter.
Y
Turn on device within 5 sec.
Entering console mode...
Ready for console mode. Input anything.

2023-07-31 16:50:02.527:
2023-07-31 16:50:02.527:No such command - type help
2023-07-31 16:50:03.033:[MROM]  dload
2023-07-31 16:50:13.519:[MROM]
2023-07-31 16:50:14.030:[MROM]  Please enter address (0 ~ 3FF000) and click enter: 0
0. da14531_multi_part_proxr.img
1. DA16200_FBOOT-GEN01-01-c7f4c6cc22_W25Q32JW.img
2. DA16200_FRTOS-GEN01-01-c4ca8087e8-006537.img
Please enter file name or number in list, and click enter: 1
Entering download mode...
2023-07-31 16:50:48.824 Ready for download.
Download file 1: DA16200_FBOOT-GEN01-01-c7f4c6cc22_W25Q32JW.img : 100.00% : 8.08s
2023-07-31 16:50:58.033 Done successfully.
Will you continue to download? y or n: y
Please enter address (0 ~ 3FF000) and click enter: 23000
0. da14531_multi_part_proxr.img
1. DA16200_FBOOT-GEN01-01-c7f4c6cc22_W25Q32JW.img
2. DA16200_FRTOS-GEN01-01-c4ca8087e8-006537.img
Please enter file name or number in list, and click enter: 2
Entering download mode...
2023-07-31 16:54:14.624 Ready for download.
Download file 1: DA16200_FRTOS-GEN01-01-c4ca8087e8-006537.img : 100.00% : 72.76s
2023-07-31 16:55:28.515 Done successfully.
Will you continue to download? y or n: n
Entering console mode...
Ready for console mode. Input anything.

[MROM]  reboot
reboot
Wakeup source is 0x4
[dpm_init_retmemory] DPM INIT CONFIGURATION(1)


        ********************************************************
        *             DA16200 SDK Information
        * ---------------------------------------------------
        *
        * - CPU Type        : Cortex-M4 (120MHz)
        * - OS Type         : FreeRTOS 10.4.3
        * - Serial Flash    : 4 MB
        * - SDK Version     : V3.2.8.0 GEN
        * - F/W Version     : FRTOS-GEN01-01-c4ca8087e8-006537
        * - F/W Build Time  : Jul 31 2023 14:07:09
        * - Boot Index      : 0
        *
        ********************************************************



System Mode : Station Only (0)
>>> Start DA16X Supplicant ...
>>> DA16x Supp Ver2.7 - 2022_03
>>> MAC address (sta0) : d4:3d:39:11:34:fc
>>> sta0 interface add OK
>>> Start STA mode...
RTC switched to XTAL
```

```
>>> Hello World #1 ( Non network dependent application ) !!!
```

NOTE
Do not reset or power on/off the board between the FBOOT and FRTOS image programming. It causes the programming failure.

Appendix E Country Codes

Table 38: Country Codes

Code	Country	Code	Country	Code	Country	Code	Country
AD	Andorra	EC	Ecuador	LC	Saint Lucia	RE	Reunion
AE	United Arab Emirates	EE	Estonia	LI	Liechtenstein	RO	Romania
AF	Afghanistan	EG	Egypt	LK	Sri Lanka	RS	Serbia
AI	Anguilla	ES	Spain	LS	Lesotho	RU	Russia
AL	Albania	ET	Ethiopia	LT	Lithuania	RW	Rwanda
AM	Armenia	EU	Europe	LU	Luxembourg	SA	Saudi
AR	Argentina	FI	Finland	LV	Latvia	SE	Sweden
AS	Samoa	FM	Micronesia	MA	Morocco	SG	Singapore
AT	Austria	FR	France	MC	Monaco	SI	Slovenia
AU	Australia	GA	Gabon	MD	Moldova	SK	Slovakia
AW	Aruba	GB	United Kingdom	ME	Montenegro	SN	Senegal
AZ	Azerbaijan	GD	Grenada	MF	Saint Martin	SR	Suriname
BA	Bosnia	GE	Georgia	MH	Marshall Islands	SV	El Salvador
BB	Barbados	GF	French Guiana	MK	Macedonia	SY	Syria
BD	Bangladesh	GH	Ghana	MN	Mongolia	TC	Turks Caicos
BE	Belgium	GL	Greenland	MO	Macao	TD	Chad
BF	Burkina Faso	GP	Guadeloupe	MP	Northern Mariana Islands	TG	Togo
BG	Bulgaria	GR	Greece	MQ	Martinique	TH	Thailand
BH	Bahrain	GT	Guatemala	MR	Mauritania	TN	Tunisia
BL	Barthelemy	GU	Guam	MT	Malta	TR	Turkey
BM	Bermuda	GY	Guyana	MU	Mauritius	TT	Trinidad and Tobago
BN	Brunei	HK	Hong Kong	MV	Maldives	TW	Taiwan
BO	Bolivia	HN	Honduras	MW	Malawi	TZ	Tanzania
BR	Brazil	HR	Croatia	MX	Mexico	UA	Ukraine
BS	Bahamas	HT	Haiti	MY	Malaysia	UG	Uganda
BT	Bhutan	HU	Hungary	NG	Nigeria	UK	United Kingdom
BY	Belarus	ID	Indonesia	NI	Nicaragua	US	USA
BZ	Belize	IE	Ireland	NL	Netherlands	UY	Uruguay
CA	Canada	IL	Israel	NO	Norway	UZ	Uzbekistan
CF	Central Africa	IN	India	NP	Nepal	VA	Vatican City

Code	Country	Code	Country	Code	Country	Code	Country
CH	Switzerland	IR	Iran	NZ	New Zealand	VC	St. Vincent and Grenadines
CI	Ivory Coast	IS	Iceland	OM	Oman	VE	Venezuela
CL	Chile	IT	Italy	PA	Panama	VI	Virgin Islands, US
CN	China	JM	Jamaica	PE	Peru	VN	Vietnam
CO	Colombia	JO	Jordan	PF	Polynesia	VU	Vanuatu
CR	Costa Rica	JP	Japan	PG	Papua New Guinea	WF	Wallis and Futuna Islands
CU	Cuba	KE	Kenya	PH	Philippines	WS	Samoa
CX	Christmas Island	KH	Cambodia	PK	Pakistan	YE	Yemen
CY	Cyprus	KN	St. Kitts and Nevis	PL	Poland	YT	Mayotte
CZ	Czech	KP	N.Korea	PM	St. Pierre and Miquelon	ZA	S.Africa
DE	Germany	KR	S.Korea	PR	Puerto Rico	ZW	Zimbabwe
DK	Denmark	KW	Kuwait	PT	Portugal	ALL	ALL
DM	Dominica	KY	Cayman Islands	PW	Palau		
DO	Dominican Rep	KZ	Kazakhstan	PY	Paraguay		
DZ	Algeria	LB	Lebanon	QA	Qatar		

Revision History

Revision	Date	Description
1.8	Aug. 18, 2023	● Changed IDE to e^{2}studio in section 5. ● Updated how to program the firmware image in section 4.5.1. ● Updated recovering way in Appendix D. ● Updated how to measure current at each mode in section 6.5. ● Updated descriptions about DPM mode in section 6.4. ● Changed “DPM sleep” to “DPM low power.”
1.7	Jun. 30, 2023	● Fixed the name of section A.3. ● Updated the reference section.
1.6	Mar. 16, 2023	Changed Windows build tools version.
1.5	Jan.12, 2023	● Added installation using batch/shell script. ● Changed location of firmware image files in section 5.4. ● Added build configuration (Release/Debug) in section 5.5.1. ● Added how to move project location in section 5.5.3. ● Added Document (UM-WI-040), Current measurement to 6.5.2.
1.4	Oct. 25, 2022	Added description of project presentation option in section 5.7.
1.3	Jun. 14, 2022	● Update console commands in Appendix B.1.1 and B.1.2. ● Update SFLASH memory map in Appendix C.2. ● Update J-Link section 5.8.
1.2	Apr. 11, 2022	● Update logo, disclaimer, copyright. ● Added link to J-Link / J-Trace User Guide (UM08001) in section 5.6.
1.1	Nov. 29, 2021	Title was changed.
1.0	Oct. 27, 2021	First Release.

Status Definitions

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

RoHS Compliance

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

Important Notice and Disclaimer

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers skilled in the art designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only for development of an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising out of your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu

Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

Contact Information

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

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

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.