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 EVKs 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 Save Module
EVK	Evaluation Kit
IDE	Integrated Development Environment
JDK	Java Development Kit
RTOS	Real Time Operating System
RTOS	Real Time Operating System
SDK	Software Development Kit
SFDP	Serial Flash Discoverable Parameter
SSID	Service Set Identifier
TIM	Traffic Indicator Module
UART	Universal Asynchronous Receiver/Transmitter
UC	Unicast Packet

2 References

[1] DA16200, Datasheet, Dialog Semiconductor

[2] UM-WI-046, DA16200, FreeRTOS SDK Programmer Guide, User Manual, Dialog Semiconductor

[3] UM-WI-xxx, DA16200, Example Application Guide, User Manual, Dialog Semiconductor

[4] UM-B-114, DA14531, Devkit Pro Hardware, User Manual, Dialog Semiconductor

[5] UM-WI-012, DA16200 SPI SFlash Downloader, User Manual, Dialog Semiconductor

[6] UM-WI-003, DA16200 DA16600 AT-Command, Dialog Semiconductor

[7] UM-WI-038, DA16200 Getting Started with AT-Command for AWS-IoT, User Manual, Dialog Semiconductor

[8] UM-WI-039, DA16200 DA16600 Multi Downloader tool, Dialog Semiconductor

[9] UM-WI-040, DA16200 EVK Pro User Manual, Dialog Semiconductor

[10] UM-WI-041, DA16600 EVK Pro User Manual, Dialog Semiconductor

[11] UM-WI-042, DA16200 Provisioning the Mobile App for Android/iOS, User Manual, Dialog Semiconductor

[12] UM-WI-044, DA16600 Provisioning the Mobile App for Android/iOS, User Manual, Dialog Semiconductor

[13] DA16200 DEVKT Electric Schematic

[14] DA16600 DEVKT Electric Schematic

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 a step by step guide to get started using the EVKs & SDKs by providing details on:

- The DA16200 and DA16600 Evaluation Kit hardware (see Section 4)
- How to connect to the EVKs (see Section 4.4)
- How to download firmware to the DA16200/600 (see Section 4.5)
- How to provision and test the Wi-Fi connection (see Section 4.6)
- How to install the development tools (see Section 5)
- How to install the SDK (see Section 5.6)
- How to build an application (see Section 5.7)
- How to debug an application (see Section 5.9)
- How to perform various test procedures to demonstrate the capabilities of the DA16200/600 (see Section 6)

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

If the Wi-Fi solution being developed is based on a host MCU that uses the DA16200/600 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 AT-Command User Manual Ref. [6]
- UM-WI-038 DA16200 Getting Started with AT-Command for AWS-IoT Ref. [7]

If the Wi-Fi solution being developed requires special functions or it is a standalone solution using only the DA16200/600, 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]
- UM-WI-xxx DA16200 Example Application Guide Ref. [3]

4 Evaluation Kits

4.1 Introduction

The DA16200 & DA16600 EVKs are designed to allow a customer to get started testing and developing Wi-Fi applications very quickly.

The EVK contains:

- The specific board with either DA16200 or DA16600MOD installed.
- USB Cable

4.2 DA16200 Module EVK Ver 9.0_210205

Figure 1 shows the hardware configuration of the DA16200 Module Evaluation Kit (EVK).

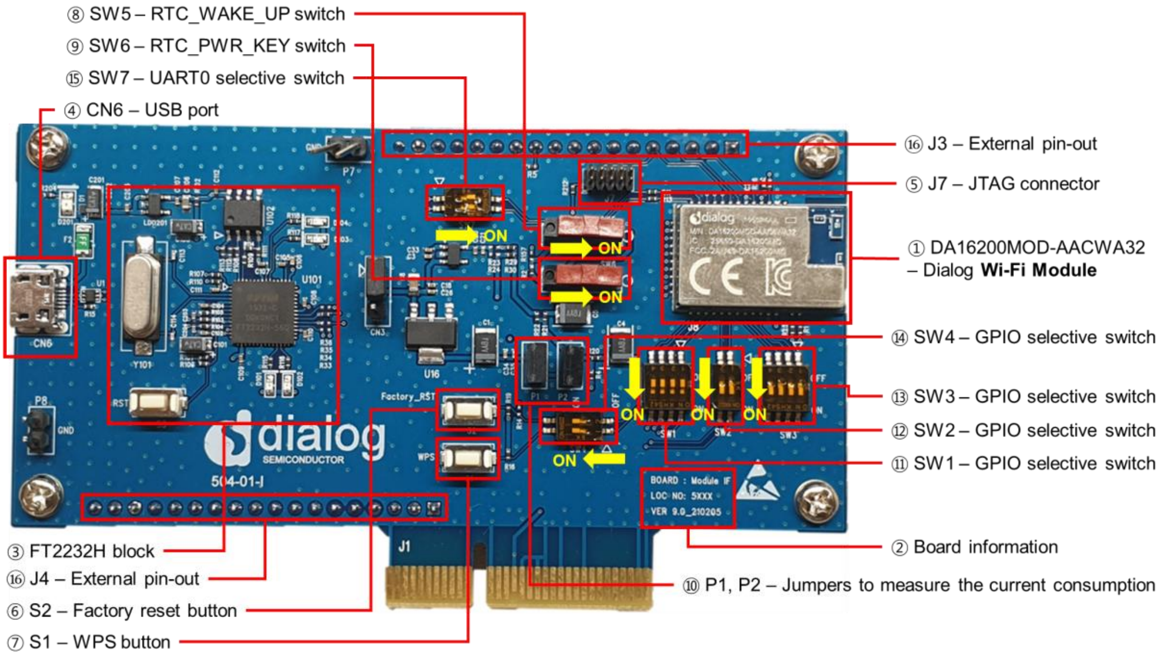

⑧ SW5 – RTC_WAKE_UP switch
⑨ SW6 – RTC_PWR_KEY switch
⑮ SW7 – UART0 selective switch
④ CN6 – USB port
⑯ J3 – External pin-out
⑤ J7 – JTAG connector
① DA16200MOD-AACWA32 – Dialog Wi-Fi Module
⑭ SW4 – GPIO selective switch
⑬ SW3 – GPIO selective switch
⑫ SW2 – GPIO selective switch
⑪ SW1 – GPIO selective 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

Figure 1: DA16200 Module EVK Hardware Configuration

DA16200 has the following components:

Table 1: Components on DA16200 EVK

No	Name	Description
1	**DA16200MOD-AACWA32**	Dialogs **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.

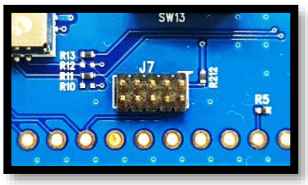
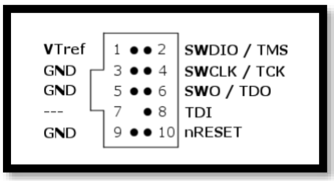
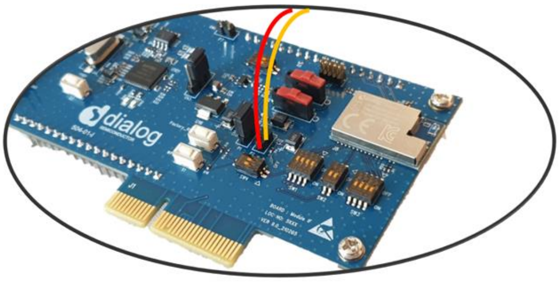
No	Name	Description
5	**J7 – JTAG connector**	Connector for the IAR I-jet JTAG Debugger   VTref GND GND --- GND 1 2 3 4 5 6 7 8 9 10 SWDIO / TMS SWCLK / TCK SWO / TDO TDI nRESET Note: Pin 7 of the I-Jet debugger cable is keyed with a white plug so pin 7 must be removed from the EVB.
6	**S2 – Factory 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.
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.
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 2 below.
12	**SW2 – GPIO selection switch**	GPIO selection switch, default ON, see Table 2 below.
13	**SW3 – GPIO selection switch**	GPIO selection switch, default ON, see Table 2 below.
14	**SW4 – GPIO selection switch**	GPIO selection switch, default ON, see Table 2 below.
15	**SW7 – UART0 selection switch**	UART0 selection switch, default ON, see Table 2 below.
16	**J3, J4 – External Connector**	Provides access to all pins of the DA16200MOD for testing and debugging.

Table 2: GPIO, 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 is used for measuring current consumption, the P1 jumper must be removed.

For more details on the DA16200 EVK see the DA16200 DEVKT Electric Schematic Ref. [13].

For more detailed information on using the power meter kit, see UM-WI-040 DA16200 EVK Pro User Manual Ref. [9].

4.3 DA16600 Module EVK Ver 4.0_201116

Figure 2 shows the hardware configuration of the DA16600 Module Evaluation Kit (EVK).

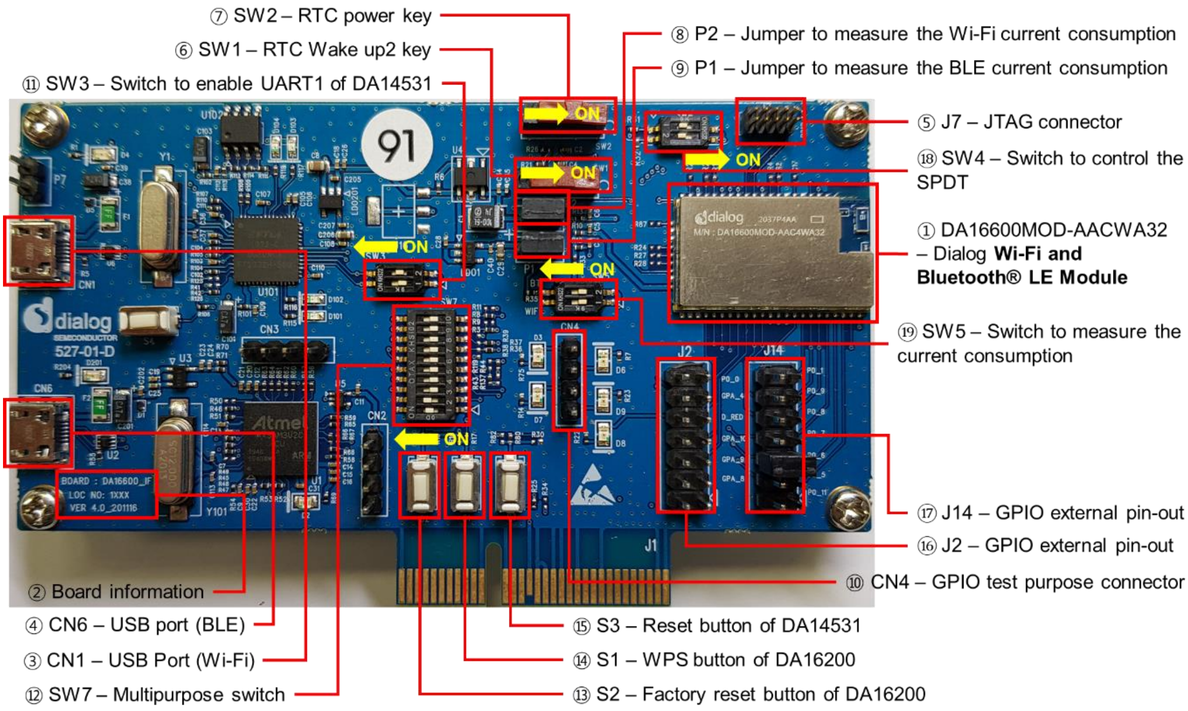
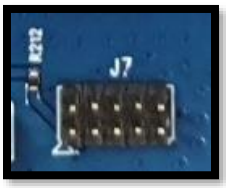
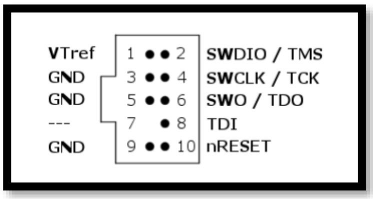
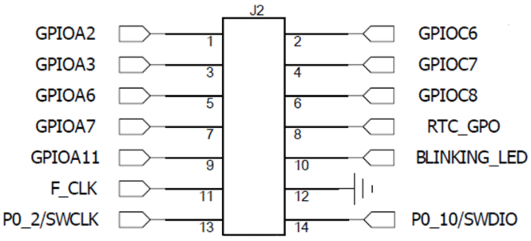
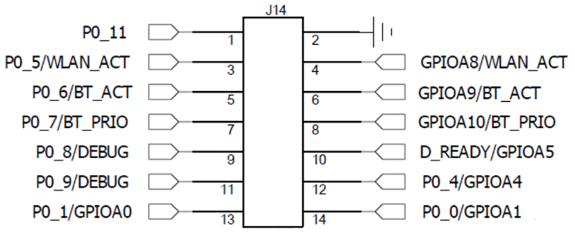

⑦ SW2 – RTC power key
⑥ SW1 – RTC Wake up2 key
⑪ SW3 – Switch to enable UART1 of DA14531
⑧ P2 – Jumper to measure the Wi-Fi current consumption
⑨ P1 – Jumper to measure the BLE current consumption
⑤ J7 – JTAG connector
⑱ SW4 – Switch to control the SPDT
① DA16600MOD-AACWA32 – Dialog Wi-Fi and Bluetooth® 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
91

Figure 2: DA16600 Module EVK Hardware Configuration

DA16600 EVB has the following components:

Table 3: Components on DA16600 EVB

ID	Name	Description
1	**DA16600MOD-AACWA32**	Dialogs **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 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.
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.
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 1 GPIOA3 3 GPIOA6 5 GPIOA7 7 GPIOA11 9 F_CLK 11 P0_2/SWCLK 13 2 GPIOC6 4 GPIOC7 6 GPIOC8 8 RTC_GPO 10 BLINKING_LED 12 14 P0_10/SWDIO

ID	Name	Description
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 EVK includes several switches, see description for each switch below.

Table 4: 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
	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 EVK see the DA16600 DEVKT Electric Schematic Ref. [14].

For more detailed information on using the power meter kit, see UM-WI-041 DA16600 EVK Pro User Manual Ref. [10].

4.4 Connecting to the EVKs

4.4.1 Configuring the DA16200/600 Serial Debug Interface

The DA16200/600 provides a command/debug interface on uart0 for performing configuration and diagnostic functions.

When the EVK is connected to the USB port (**CN6** on the DA16200 EVK or **CN1** on the DA16600 EVK), two virtual COM ports are created.

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

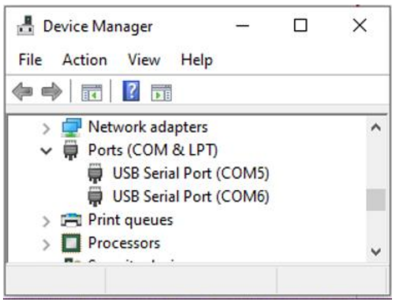


Figure 3: Windows - COM Ports

NOTE
On Windows, If the ftdi serial driver installation does not happen automatically, the drivers can be downloaded from the following URL and installed 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 is for the DA16200/600 debug interface.

The higher numbered COM port is for the DA16200/600 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 that supports **Y-Modem** is required to download the firmware to the DA16200.

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 EVK (**CN6** on DA16200 EVK or **CN1** on DA16600 EVK) 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 EVK (**SW6** on the DA16200 EVK or **SW2** on the DA16600 EVK) 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 now 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 the DA14531 Serial Debug Interface (DA16600 EVK Only)

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 EVK contains the DA14531 BLE device which is used for provisioning of the DA16600 Wi-Fi interface using a mobile application.

On the DA16600 EVK, 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, **SW3** and **SW7** must be set as follows:

Switch	Pin	Description	Relevant pin		Setting
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
SW7	5	Enables UART debug of DA14531 (UART2 RX)	DA14531	P0_8/DEBUG	**ON**
	6	Enables UART debug of DA14531 (UART2 TX)	DA14531	P0_9/DEBUG	**ON**

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:

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
```

4.5 Updating the Firmware

When first receiving an EVK, the firmware must be updated to the latest version.

Prebuilt versions of the firmware for DA16200 and DA16600 can be downloaded from the Dialog website: https://www.dialog-semiconductor.com/products/wi-fi

The firmware images can be found under the Resources page for the specific device in a file typically named "FreeRTOS SDK Image".

Alternately, the DA16200/600 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 EVK which contain either the Adesto AT25SL321 serial flash or 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

Use the appropriate firmware image based on the flash version.

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

Table 6: Serial Flash Memory Map

Item	4 MB		Notes
	Address	Size	
2nd Bootloader	0x0000_0000	136 kB	FBOOT firmware image
Boot Index	0x0002_2000	4 kB	Selects RTOS #0 or RTOS #1 as he active firmware
RTOS #0	0x0002_3000	1788 kB	FRTOS firmware image
RTOS #1	0x001E_2000	1792 kB	Alternate FRTOS 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 DA16200/600 using the **Y-Modem** file transfer protocol.

4.5.1 Firmware Update Using Commands (Windows/Linux)

The following steps will load the firmware images onto the DA16200/600:

1. Connect the EVK to a PC using a Micro USB cable and configure the debug interface and serial terminal as described in Section 4.4.1.
2. Power ON the DA16200/600 EVK.
3. At the `[/DA16xxx]` prompt, type `reset` to go to the Mask ROM prompt `[MROM]`:

```
[/DA16200] # reset                                      Switch to MROM command mode.
        ************************************************
        *      FCI   FC9K MaskROM BootLoader
        *  Cortex-M4 (XTAL 40000 KHz, SYS 120000 KHz)
        *  Console Baud Rate : 0 (00000000)
        *  HW Version Num.   : fc905010
        *  Build Option      : RomALL
        *  RoSDK Date & Time : Mar 13 2019 13:05:45
        *  Build Date & Time : Mar 13 2019 13:11:24
        *          http://www.fci.co.kr
        ************************************************
[MROM]
```

4. Load each firmware image:

   The FBOOT, FRTOS and optionally the DA14531 firmware images are downloaded individually. Use the following process to download each firmware image:

   a. At the [MROM] prompt, use one of the following `loady` commands to load the required firmware images:

Table 7: FW Image loady Commands

FW Image	Command	Notes
FBOOT	loady boot or loady 0	Second bootloader firmware. Only needs to be downloaded for a new EVK or if the SFDP changes.
FRTOS	loady 23000	Main firmware.
DA14531 firmware	loady 3ad000 1000 bin	For the DA16600MOD only. Only needs to be downloaded for a new EVK or if there is new DA14531 firmware available.

For example:

```
[MROM] loady boot                                    Start the Y-Modem download protocol.
Load Addr: 000f6684
To cancel a session, press Ctrl+X
Load Y-Modem (Load Offset:f6684)
CCC
```

At this point, the DA16200 is waiting for the download to start.

   b. In the terminal emulation program, start the Y-Modem file transfer:

   – For Windows **Tera Term**:Open the **File** tab and select **Transfer**->**YMODEM**->**Send**.

   Navigate to where the firmware image is stored, choose the required firmware image and start the download.

   – For Linux **minicom:**

   Press **{Ctrl+A}+S** and select **ymodem** from the menu.

   Navigate to where the firmware image is stored, choose the required firmware image and start the download.

NOTE
If the transfer does not start within a few seconds, cancel and start again.

   c. After the transfer completes, the total size is displayed:

```
xyzModem - CRC mode, 5(SOH)/91(STX)/0(CAN) PKTs, 8 retries, err:Timed out
```

```
## Total Size      = 0x000168a0 = 92320 Bytes
[MROM]
```

NOTE
Ignore the “ `err:..`” messages.

d. Repeat steps a. through c. to download the FRTOS firmware image and then the DA14531 firmware image (if using DA16600MOD).

5. Run the firmware:

   After all firmware images have been downloaded, run the `boot` command at the `[MROM]` prompt to start the firmware:

```
[MROM] boot                                       Use the "boot" command to start the firmware.

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 : 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] #
```

6. Reset to factory defaults:

   If it is the first time using the EVK or to return the EVK to a clean uninitialized state, run the factory reset command.

   At the `[/DA16200]` prompt, type `factory` to do a factory reset:

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

Start Factory-Reset ...

Rebooting....

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 : 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] #
```

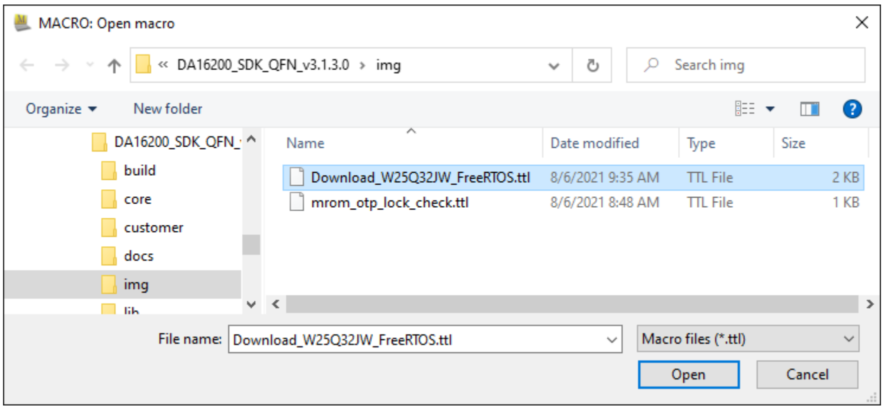
Once the firmware is updated successfully, the device can be configured by doing the Wi-Fi provisioning process. See Section 4.6 for details on how provision the Wi-Fi communication interface.

4.5.2 Firmware Update Using the Terra Term Script (Windows Only)

For Windows users, the firmware images can be downloaded automatically using a terra term script as follows:

1. Once terra term is running and connected to the DA16200/600, open the **Control** tab and select the **Macro** menu item.

   When the **MACRO: Open Macro** file selection window opens, navigate to the directory where the firmware images are stored and select the **.ttl** file (see Figure 4).



MACRO: Open macro

« DA16200_SDK_QFN_v3.1.3.0 › img

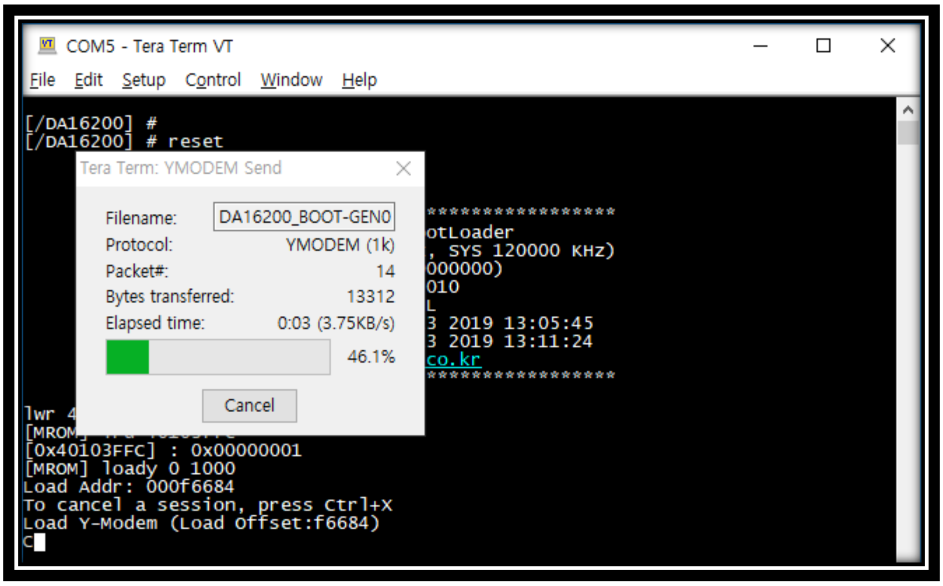
Name	Date modified	Type	Size
Download_W25Q32JW_FreeRTOS.ttl	8/6/2021 9:35 AM	TTL File	2 KB
mrom_otp_lock_check.ttl	8/6/2021 8:48 AM	TTL File	1 KB

File name: Download_W25Q32JW_FreeRTOS.ttl

Macro files (*.ttl)

Figure 4. Open the Macro File

2. After opening the macro file, the DA16200/600 will reset, and the BOOT firmware image is downloaded (see Figure 5).



COM5 - Tera Term VT

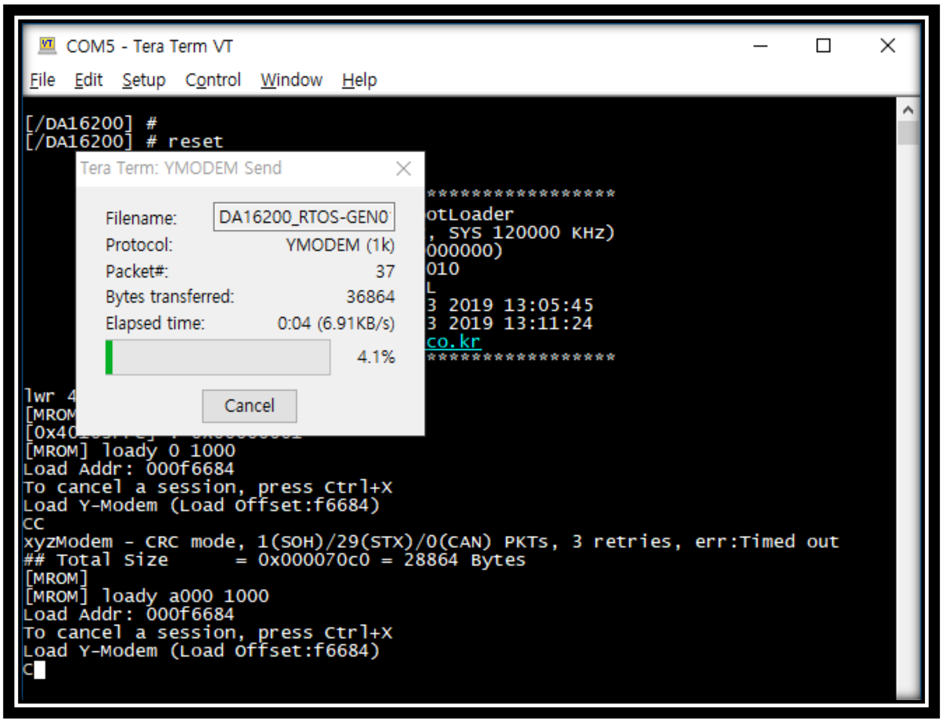
```
[/DA16200] #
[/DA16200] # reset
                      ******************
                      otLoader
                      , SYS 120000 KHz)
                      000000)
                      010
                      L
                      3 2019 13:05:45
                      3 2019 13:11:24
                      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
```

Tera Term: YMODEM Send

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

Figure 5. Download Bootloader

3. Download RTOS image (see Figure 6).



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

```
[/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
```

Figure 6. Download RTOS Image

4. DA16200 will reboot automatically after all images are downloaded.

4.5.3 Multi-Download Tool

A products manufacturing process requires that the firmware be download to many devices. To make this process more efficient, the Multi-Download Tool can be used to download the firmware to multiple devices at the same time. See the UM-WI-039 DA16200 DA16600 Multi Downloader Tool User Manual Ref. [8] for details.

4.5.4 Changing the Boot Index

Two versions of the main firmware can be stored in 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 “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     : V3.1.3.0 GEN
        * - F/W Version     : FRTOS-GEN01-01-15129-000000
        * - F/W Build Time  : Sep 29 2021 00:24:50
        * - Boot Index      : 1
        *
        ******************************************************
```

After the reboot, verify the “Boot Index” and “F/W Version” matches the one that was selected.

4.6 Provisioning Wi-Fi

4.6.1 Setup for Station Mode

The Wi-Fi functions of the DA16200/600 can easily be configured by running the **Easy Setup** Wi-Fi configuration wizard.

To configure the DA16200/600 to operate in Station 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 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 saved 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     : V3.1.3.0 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 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 DA16200/600 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 UM-WI-042_DA16200_Provisioning_the_Mobile_App Ref. [11] for details on how to use the mobile application.

4.6.3 Setup for Wi-Fi Provisioning Using Bluetooth® LE (DA16600)

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.

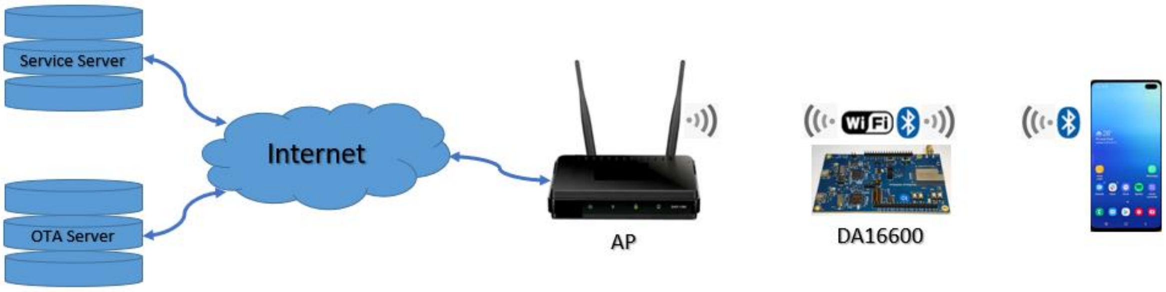


Figure 7: Diagram of Provisioning via Bluetooth® LE

Refer to UM-WI-044 Provisioning Mobile App User Manual [12] for details on how to use the mobile application to provision the Wi-Fi connection on the DA16600.

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 Eclipse IDE on either a Windows 10 or Linux based development system.

To start developing applications for the DA16200 the following steps must be performed:

- Install the ARM cross compiler
- Install and configure the Eclipse IDE
- Import the DA16200 SDK into the Eclipse 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 or Ubuntu 20.04.1 LTS
- GNU Arm GCC 10.2.1
- Eclipse 2020-09 or later
- Serial Terminal with Y-Modem support (Windows Tera Term / Linux minicom)
- J-Link Debug Probe

5.3 Installing the GNU Arm GCC Cross Compiler

The simplest method for installing the GNU Arm GCC 10.2.1 cross compiler is to use the xPack Project Manager (xpm) and install them from an xPack package.

If the xPack tools are not already installed, follow the instructions on the xPack website to install either the “Windows” or “GNU/Linux” version depending on your development system.

https://xpack.github.io/install/

Once the xPack tools are installed, open a Windows command line or a Linux terminal and install the GNU Arm GCC 10.2.1 cross compiler xPack.

Use the following command to install GNU Arm GCC 10.2.1 on either Windows 10 or Linux:

```
xpm install --global @xpack-dev-tools/arm-none-eabi-gcc@10.2.1-1.1.2
```

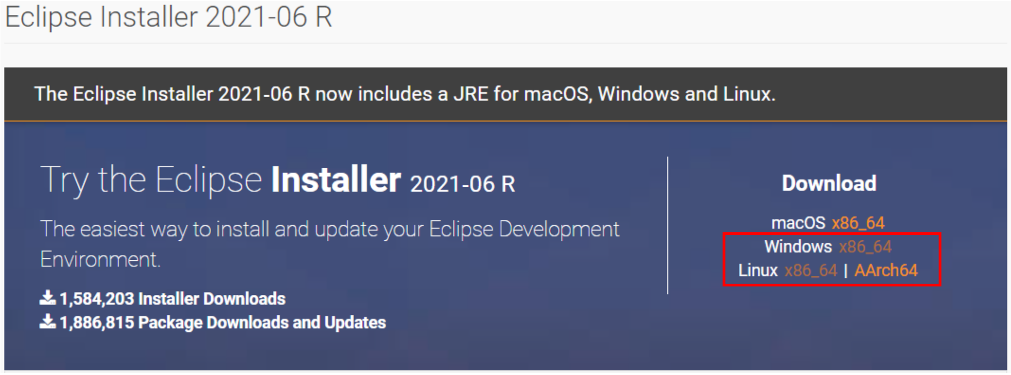
Windows 10 does not include any build tools by default, that is, “make”, therefore the Windows Build Tools xPack will also need to be installed.

Use the following command to install the Windows Build Tools on Windows 10:

```
xpm install --global @xpack-dev-tools/windows-build-tools
```

5.4 Installing the Eclipse IDE

To install the Eclipse IDE, download and run the Eclipse installer for either Windows or Linux from the Eclipse Installer website: https://www.eclipse.org/downloads/packages/installer



Eclipse Installer 2021-06 R

The Eclipse Installer 2021-06 R now includes a JRE for macOS, Windows and Linux.

Try the Eclipse **Installer** 2021-06 R

The easiest way to install and update your Eclipse Development Environment.

1,584,203 Installer Downloads

1,886,815 Package Downloads and Updates

Figure 8: Eclipse Installer Download

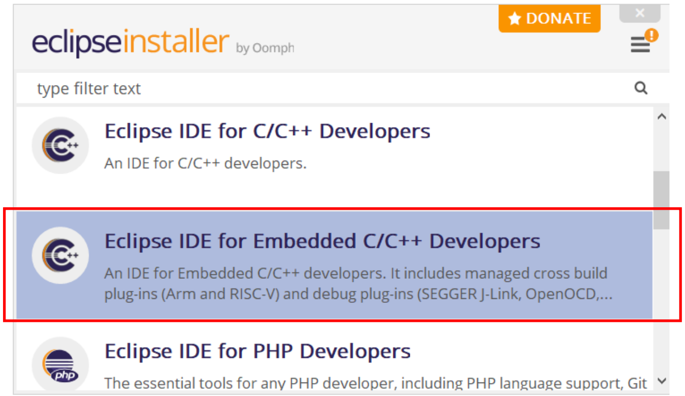
NOTE
Currently only Windows and Linux have been verified to work with the DA16xxx SDKs.

The Windows version can be installed by running the downloaded installer.

The Linux version can be installed using the following commands:

```
$
$ tar -xvzf eclipse-inst-jre-linux64.tar.gz
$ cd eclipse-installer
$ ./eclipse-inst
$
```

After running the installer, select the “Eclipse IDE for Embedded C/C++ Developers” package and then select “Install”.



Eclipse IDE for C/C++ Developers

An IDE for C/C++ developers.

Eclipse IDE for Embedded C/C++ Developers

An IDE for Embedded C/C++ developers. It includes managed cross build plug-ins (Arm and RISC-V) and debug plug-ins (SEGGER J-Link, OpenOCD,...

Eclipse IDE for PHP Developers

The essential tools for any PHP developer, including PHP language support, Git

Figure 9: Eclipse Installer Run

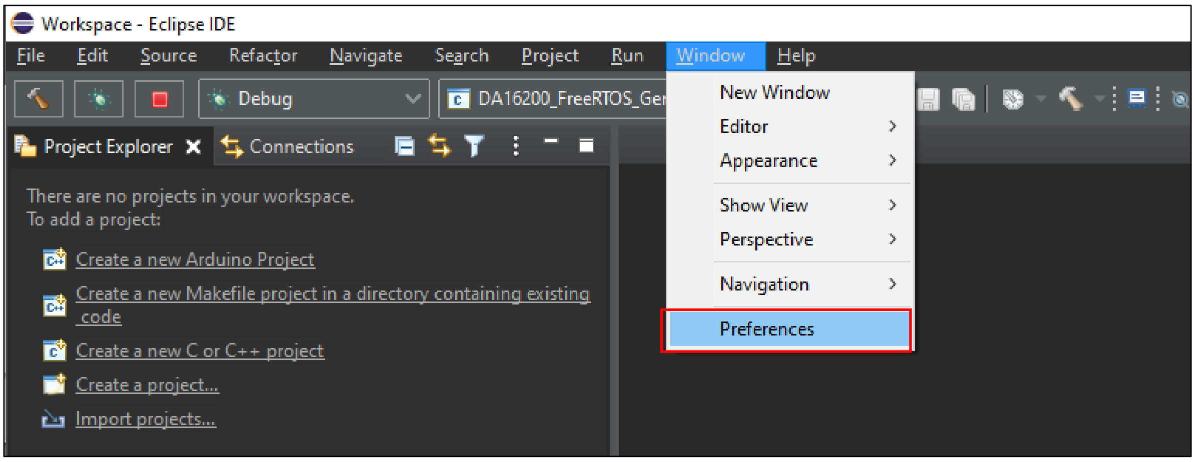
NOTE
The Eclipse Installer will automatically install the required JRE and embedded development extensions required by the DA16200/DA16600 FreeRTOS SDK.

After the installation is complete, run Eclipse and open a workspace.

5.5 Configuring the Eclipse IDE

For Eclipse to be able to build the DA16200 FreeRTOS SDK, the path to the compiler must be configured.

To setup the compiler path, open the **Window->Preferences** dialog box.



There are no projects in your workspace.
To add a project:

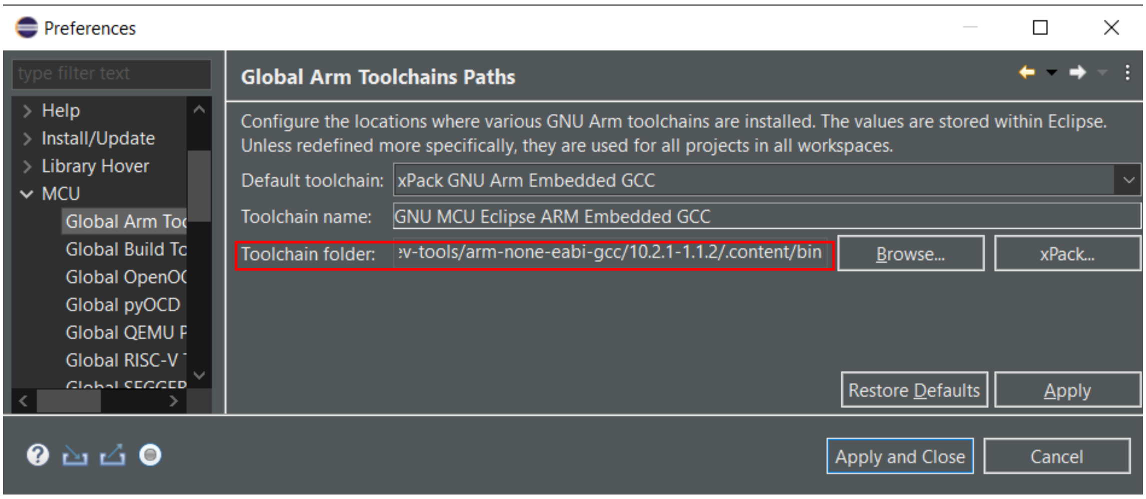
- Create a new Arduino Project
- Create a new Makefile project in a directory containing existing code
- Create a new C or C++ project
- Create a project...
- Import projects...

Figure 10: Eclipse Open Preferences Dialog Box

In the preferences dialog, select **MCU->Global Arm Toolchain Paths** and verify the Toolchain folder points to the where the GNU Arm GCC 10.2.1 toolchain is installed.

The GNU Arm GCC 10.2.1 xPack is installed in a path similar to the following:

```
C:/Users/{user}/AppData/Roaming/xPacks/@xpack-dev-tools/arm-none-eabi-gcc/{gcc version}/.content/bin
```



Global Arm Toolchains Paths

Configure the locations where various GNU Arm toolchains are installed. The values are stored within Eclipse. Unless redefined more specifically, they are used for all projects in all workspaces.

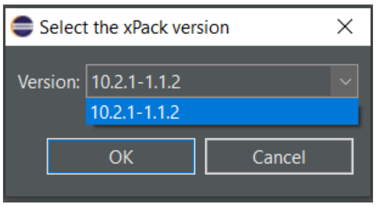
Default toolchain: xPack GNU Arm Embedded GCC

Toolchain name: GNU MCU Eclipse ARM Embedded GCC

Toolchain folder: :v-tools/arm-none-eabi-gcc/10.2.1-1.1.2/.content/bin

Figure 11: [MCU] Global Arm Toolchains Paths

If the toolchain folder is empty or incorrect, press the xPack button and select the correct Version.



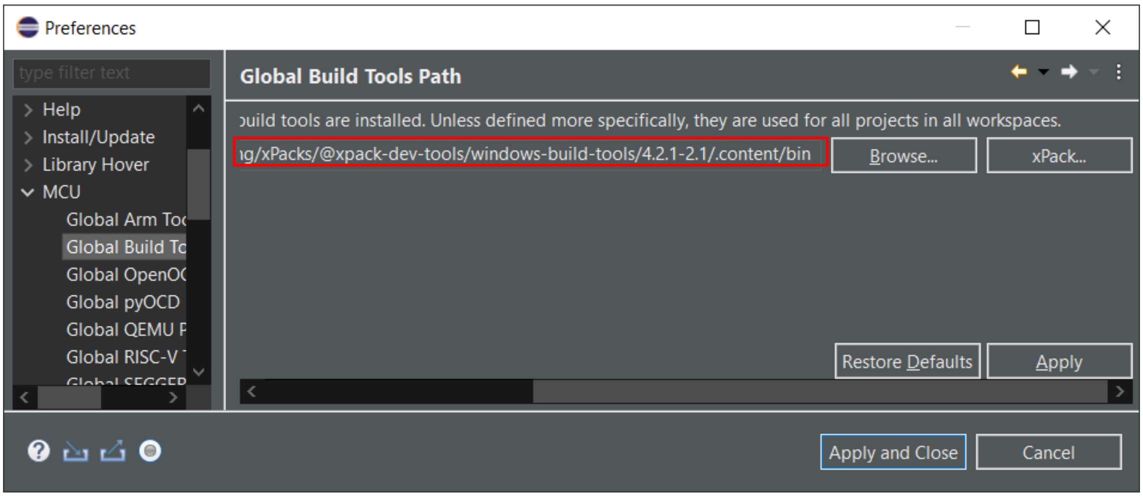
Version: 10.2.1-1.1.2

Figure 12: xPack Selection Dialog Box

For the Windows 10 installation, also check the **MCU->Global Build Tools path** and verify it points to where the Windows Build Tools xPack has been installed.

The Windows Build Tools xPack is installed in a path similar to the following:

```
C:/Users/{user}/AppData/Roaming/xPacks/@xpack-dev-tools/windows-build-tools/{Window Build Tools
version}/.content/bin
```



ɔuild tools are installed. Unless defined more specifically, they are used for all projects in all workspaces.

ıg/xPacks/@xpack-dev-tools/windows-build-tools/4.2.1-2.1/.content/bin

Figure 13: Windows 10 [MCU] Global Build Tools Path

5.6 Importing the DA16200 FreeRTOS SDK into Eclipse

Download the DA16200 FreeRTOS SDK from the Dialog website: https://www.dialog-semiconductor.com/products/wi-fi/da16200#tab-field_tab_content_resources

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_V3.2.0.0
```

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

Run Eclipse 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
~$
```

To setup the SDK in Eclipse, several projects need to be imported. The first one is the top-level project contained in the `<sdk_root_directory>`. This will provide a view into the full SDK source code and will set up the launch and debug configurations required by the SDK.

Import the top-level project into the Eclipse workspace as follows:

1. Under the **File** menu select **Import...** to open the Import dialog box and then select **General**->**Existing project into workspace** and press **Next**.

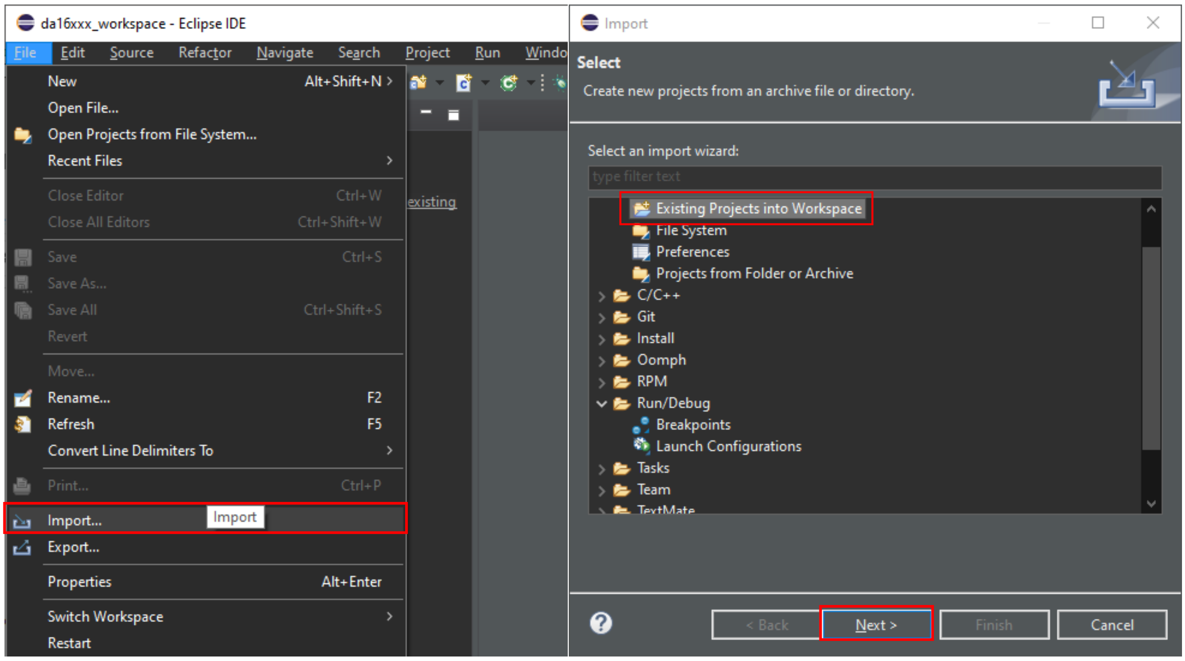
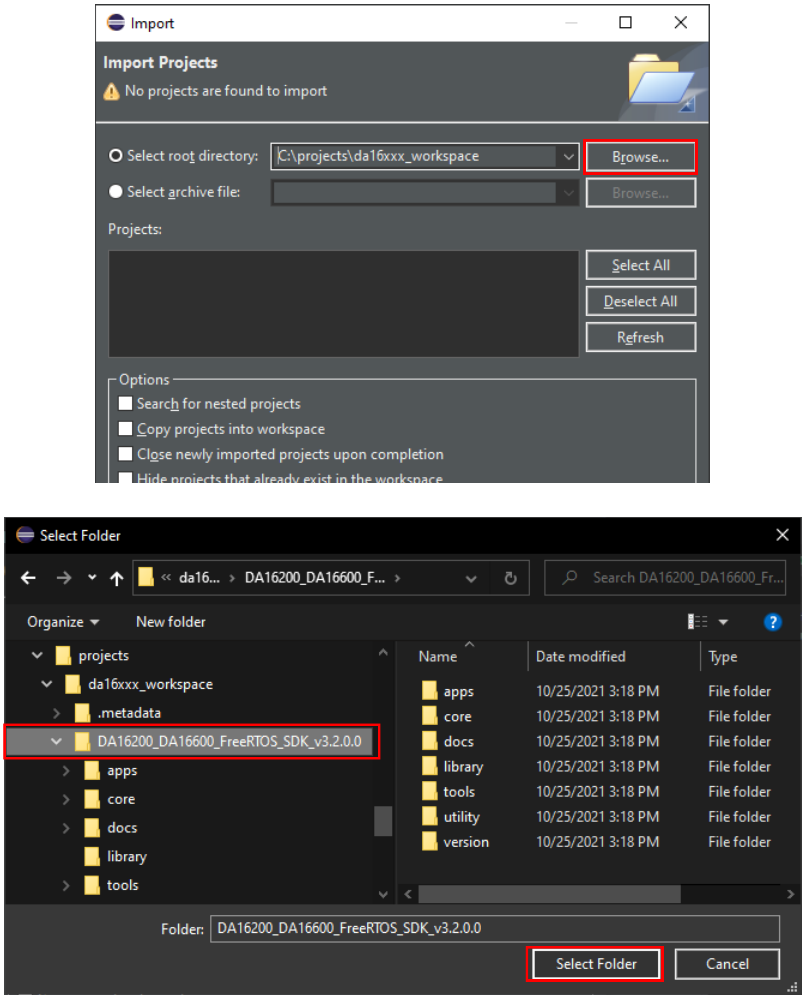


Figure 14: Import SDK to the Eclipse IDE

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 root directory** option and press the **Browse** button. Use the file manager to navigate to the `<sdk_root_directory>/` directory and then press **Select Folder**.



Import Projects

No projects are found to import

○ Select root directory: C:\projects\da16xxx_workspace [Browse...]

● Select archive file: [Browse...]

Projects:

Options

□ Search for nested projects
□ Copy projects into workspace
□ Close newly imported projects upon completion
□ Hide projects that already exist in the workspace

Select Folder

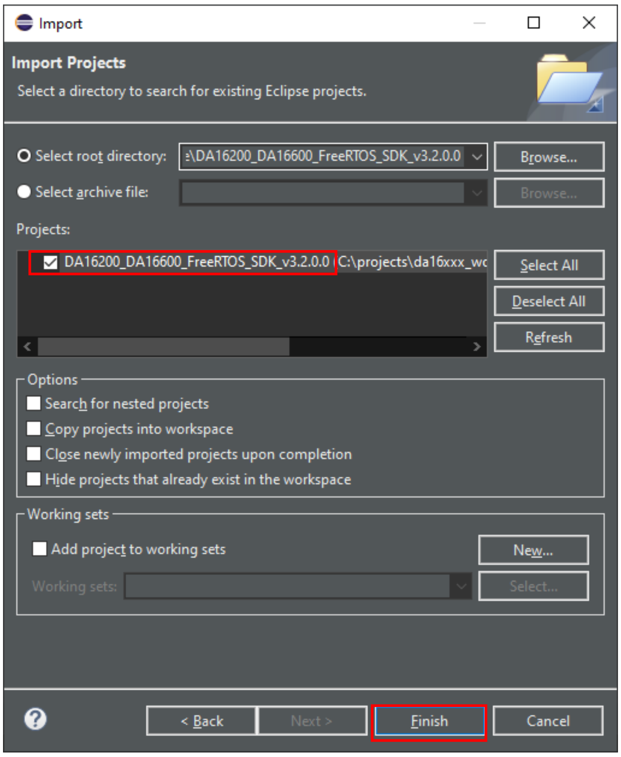
« da16... › DA16200_DA16600_F... ›

Name	Date modified	Type
apps	10/25/2021 3:18 PM	File folder
core	10/25/2021 3:18 PM	File folder
docs	10/25/2021 3:18 PM	File folder
library	10/25/2021 3:18 PM	File folder
tools	10/25/2021 3:18 PM	File folder
utility	10/25/2021 3:18 PM	File folder
version	10/25/2021 3:18 PM	File folder

Folder: DA16200_DA16600_FreeRTOS_SDK_v3.2.0.0

Figure 15: Select Folder to import the SDK into the Eclipse IDE

3. One project should appear in the **Projects** list which matches the name of the `<sdk_root_directory>`. Select the project and then press **Finish**.



Import
Import Projects
Select a directory to search for existing Eclipse projects.
○ Select root directory: :\DA16200_DA16600_FreeRTOS_SDK_v3.2.0.0
Browse...
● Select archive file:
Browse...
Projects:
☑ DA16200_DA16600_FreeRTOS_SDK_v3.2.0.0 (C:\projects\da16xxx_wc
Select All
Deselect All
Refresh
Options
□ Search for nested projects
□ Copy projects into workspace
□ Close newly imported projects upon completion
□ Hide projects that already exist in the workspace
Working sets
□ Add project to working sets
New...
Working sets:
Select...
< Back
Next >
Finish
Cancel

Figure 16: Import SDK to the Eclipse IDE

4. The project will appear in the Eclipse Project Explorer.

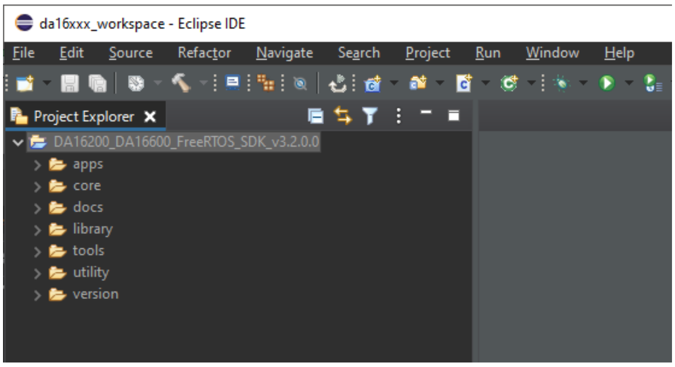
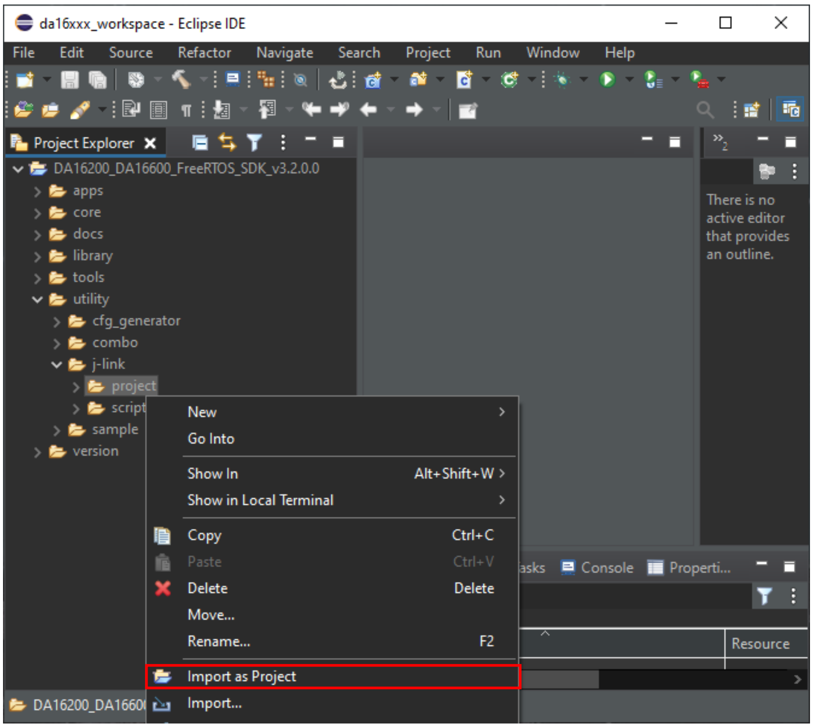


Figure 17: Eclipse Project Explorer

5. The second project to import is the **scripts_project** which provides an absolute path for certain scripts to use.
   This can be imported using the Eclipse Project Explorer by navigating to the `<sdk_top_directory>/utility/j-link/project/` and then right click on that directory and select **Import as Project**.



There is no active editor that provides an outline.

Figure 18: Import scripts_project.

The basic setup for the SDK is now complete.

The next step is to import a target project and build that project.

5.7 Import and Build a Project

Various projects are contained within the SDK directory structure. These projects are all stored under 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 which contain a project folder for either the DA16200 or the DA16600:

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

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

Imported any of these projects using the Eclipse Project Explorer by navigating to the specific project directory and then right click on project directory and select **Import as Project**.

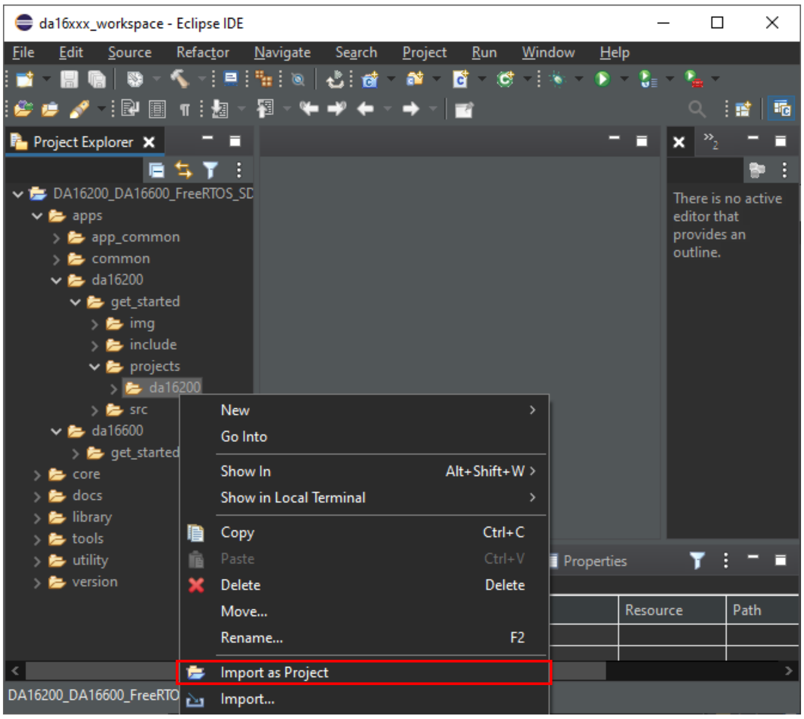


Figure 19: Import a Project.

Once the target project is imported, it will appear in the Eclipse Project Explorer.

Build the project by Right-clicking on the imported project and select **Build Project**.

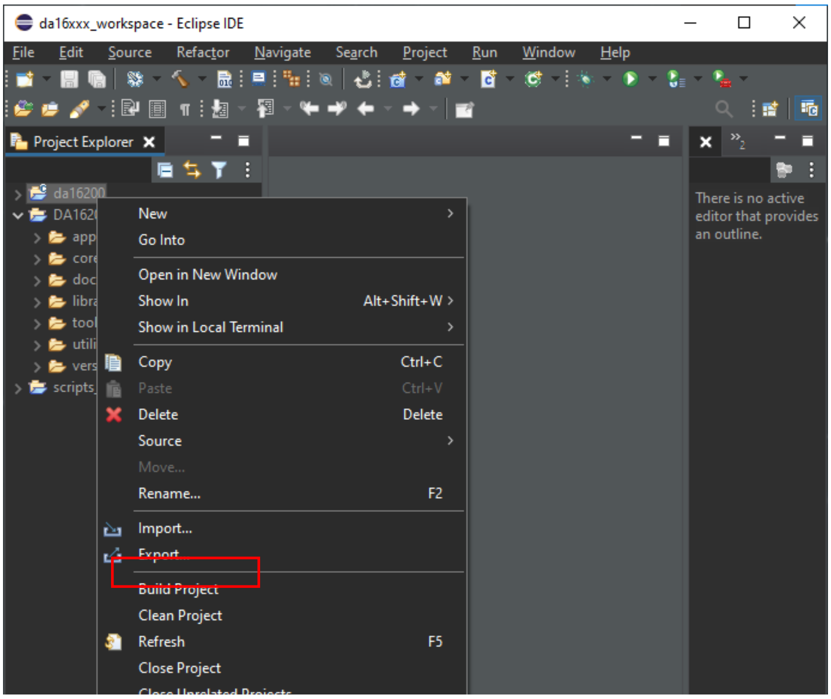
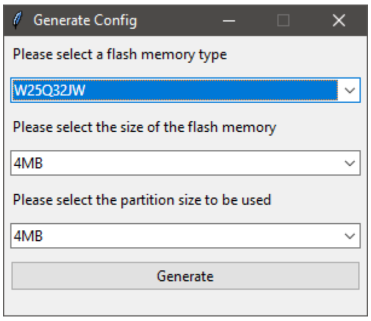


Figure 20: Build a Project

The first time the build process is performed, the Generate Configuration pop-up window is displayed automatically so that the flash memory type can be selected. Select the flash type and press 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

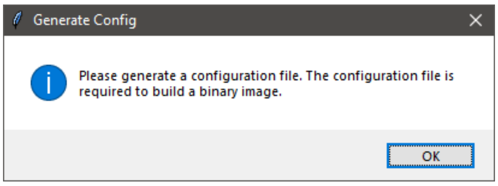
4MB

Please select the partition size to be used

4MB

Figure 21: Generate Configuration Window

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

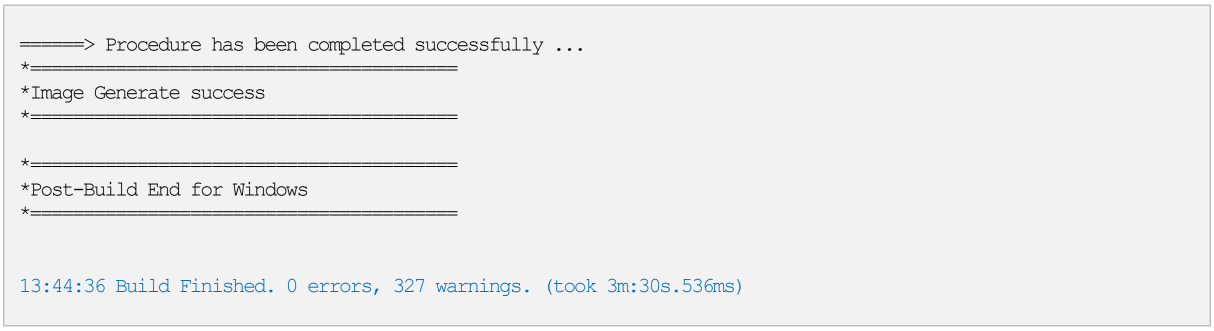


Please generate a configuration file. The configuration file is required to build a binary image.

Figure 22: Generate Configuration Warning Pop-Up

A flash configuration file is required to build the firmware so this step cannot be skipped.

After the build completes the following output is in the Eclipse Console window:



```
=====> Procedure has been completed successfully ...
*========================================
*Image Generate success
*========================================

*========================================
*Post-Build End for Windows
*========================================


13:44:36 Build Finished. 0 errors, 327 warnings. (took 3m:30s.536ms)
```

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-GEN01-01-14128-834dea5b8_W25Q32JW.img
        DA16200_FRTOS-GEN01-01-15129-866facc56.img
```

can be found in the directory show in Figure 23.

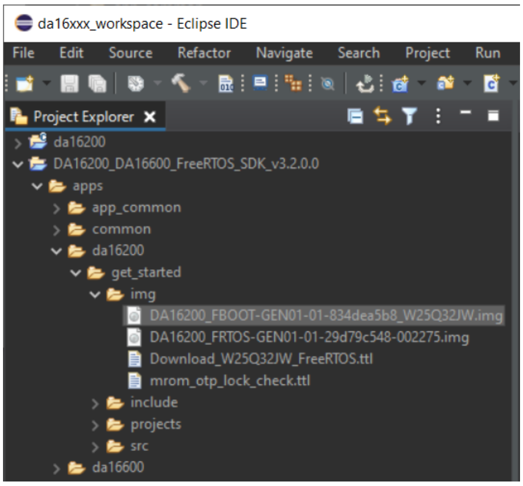

da16xxx_workspace - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run
Project Explorer
da16200
DA16200_DA16600_FreeRTOS_SDK_v3.2.0.0
apps
app_common
common
da16200
get_started
img
DA16200_FBOOT-GEN01-01-834dea5b8_W25Q32JW.img
DA16200_FRTOS-GEN01-01-29d79c548-002275.img
Download_W25Q32JW_FreeRTOS.ttl
mrom_otp_lock_check.ttl
include
projects
src
da16600

Figure 23: 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 must be the first thing loaded into flash for a new device.
- Since the bootloader image contains SFDP (flash specific) information, the bootloader must be loaded into flash before other images are loaded.
- When the SDK is updated, always load the bootloader image first.

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

5.7.1 Changing the Flash Configuration

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

Generate Configuration can be run by going to the **Run->Externel Tools** menu and selecting either **Generate Configuration For Window** or **Generate Configuration For Linux**.

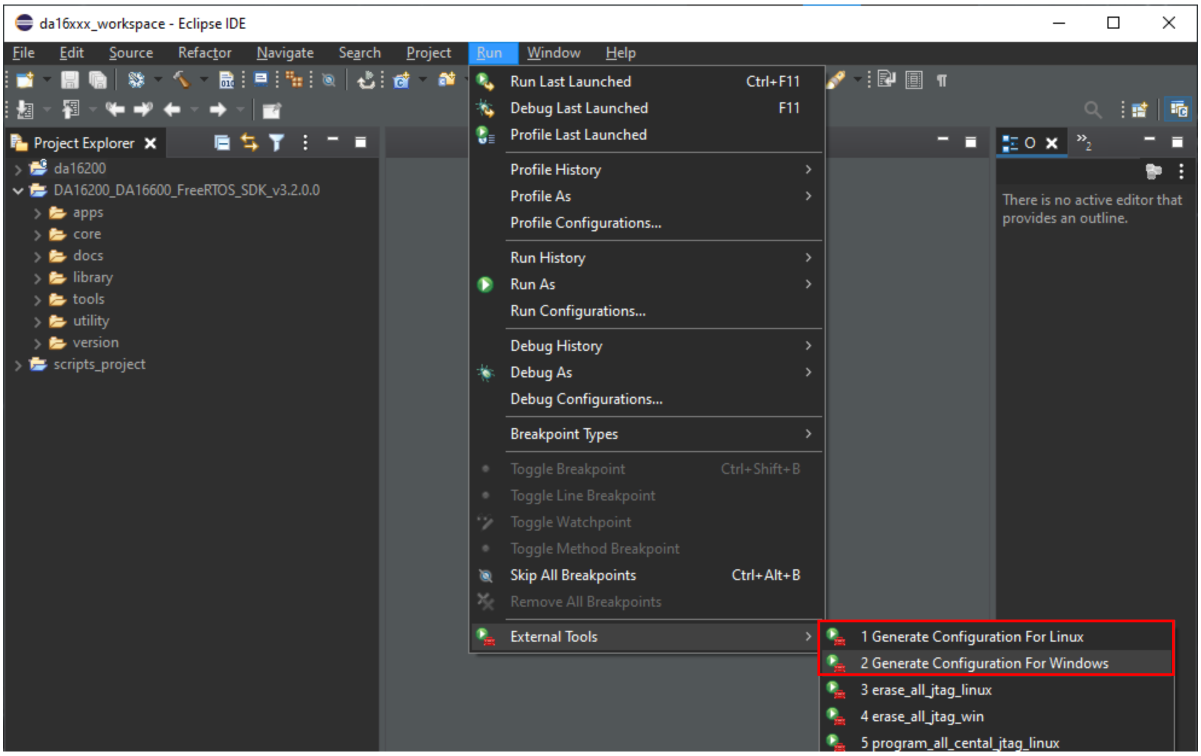

da16xxx_workspace - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
da16200
DA16200_DA16600_FreeRTOS_SDK_v3.2.0.0
apps
core
docs
library
tools
utility
version
scripts_project
Run Last Launched Ctrl+F11
Debug Last Launched F11
Profile Last Launched
Profile History
Profile As
Profile Configurations...
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 Ctrl+Alt+B
Remove All Breakpoints
External Tools
1 Generate Configuration For Linux
2 Generate Configuration For Windows
3 erase_all_jtag_linux
4 erase_all_jtag_win
5 program_all_cental_jtag_linux
There is no active editor that provides an outline.

Figure 24: Run Generate Configuration Window

The Generate Configuration pop-up window is displayed as shown in Figure 21. Select the flash type, then click on generate to create the appropriate flash configuration file required to build the firmware.

5.8 Downloading the Firmware Images (Firmware Update)

There are two different methods for downloading the firmware images.

1. Using the debug serial port of the DA16200/600.

   Connect the EVK and configure the debug interface as described in Section 4.4

   Follow the process described in Section 4.5 to download the firmware to the EVK.

2. Using Eclipse and the SEGGER J-Link debug probe.

   See Section 5.10 for more details.

Verify the firmware is working by running the tests defined in Section 6.

5.9 Debugging with J-Link Debug Probe

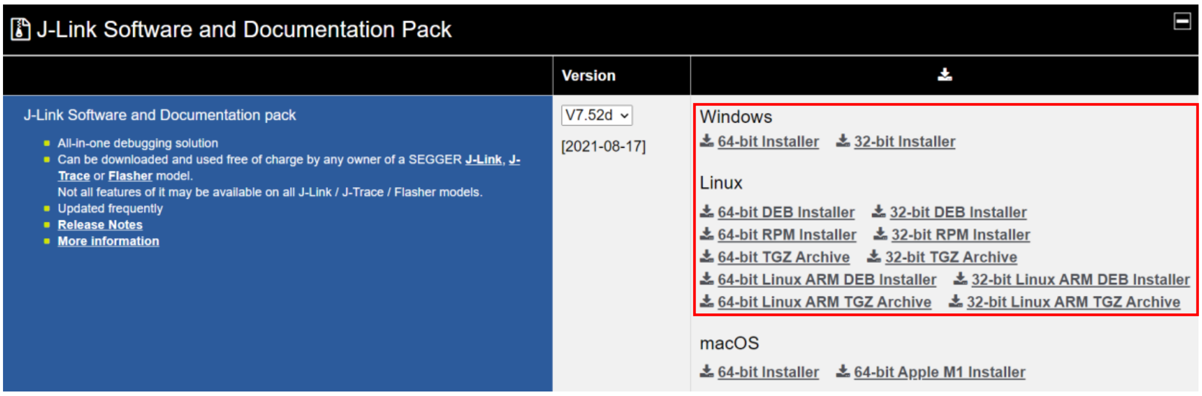
5.9.1 Installing J-Link

To debug the DA16200/600, 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

	Version	
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**	V7.52d [2021-08-17]	Windows 64-bit Installer 32-bit Installer Linux 64-bit DEB Installer 32-bit DEB Installer 64-bit RPM Installer 32-bit RPM Installer 64-bit TGZ Archive 32-bit TGZ Archive 64-bit Linux ARM DEB Installer 32-bit Linux ARM DEB Installer 64-bit Linux ARM TGZ Archive 32-bit Linux ARM TGZ Archive macOS 64-bit Installer 64-bit Apple M1 Installer

Figure 25: 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_V752d_x86_64.deb
$
```

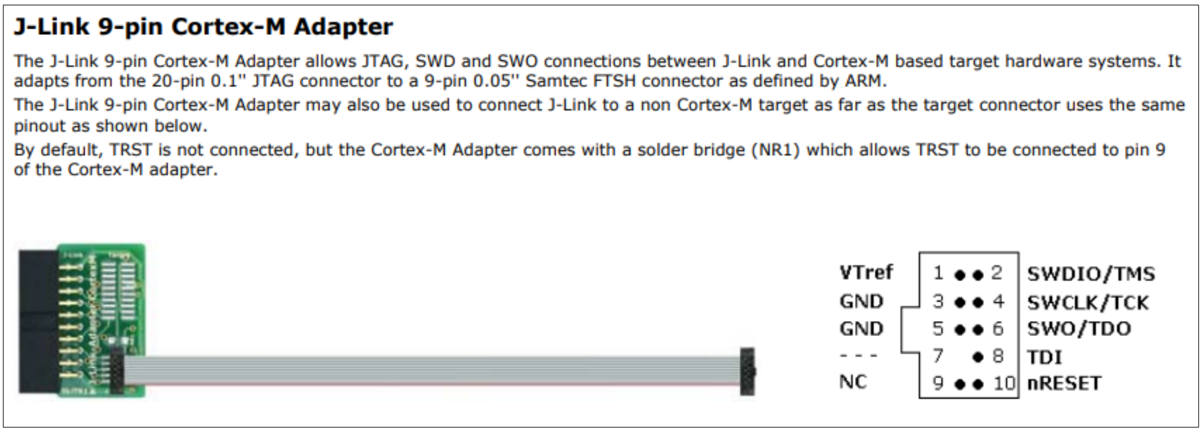
5.9.2 Connect J-Link

The following section describes how to connect the J-Link debug probe to the DA16200 Module.

There are two ways to connect J-Link Debugger.

Method 1: Connect to the JTAG connector

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 DA16200/600 EVK 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 26. J-Link 9-pin Cortex-M Adapter

Method 2: Connect to the DA16200 EVKs External Connector

If a J-Link 9-pin Cortex-M Adapter is not available, jumper wires can be used to connect a J-Link 20 pin connector to the DA16200 EVKs external connector (J3, J4) as shown below.

Table 8: DA16200 Module External Connector Mapping for J-LINK 20Pin

DA16200 (J3, J4)	J-LINK
VBAT (J4_3)	5 V Supply (19)
VDD_DIO1 (J4_4)	VTref (1)
GND (J3_18)	GND (8)
JTAG_TMS (J3_4)	SWDIO (7)
JTAG_TCLK (J3_5)	SWCLK (9)
UART0_TXD_MA (J3_9)	J-Link Tx (5)
UART0_RXD_MA (J3_10)	J-Link Rx (17)

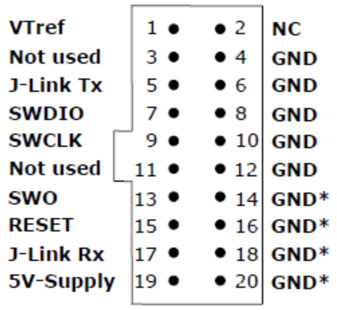


Figure 27: J-Link 20 Pin

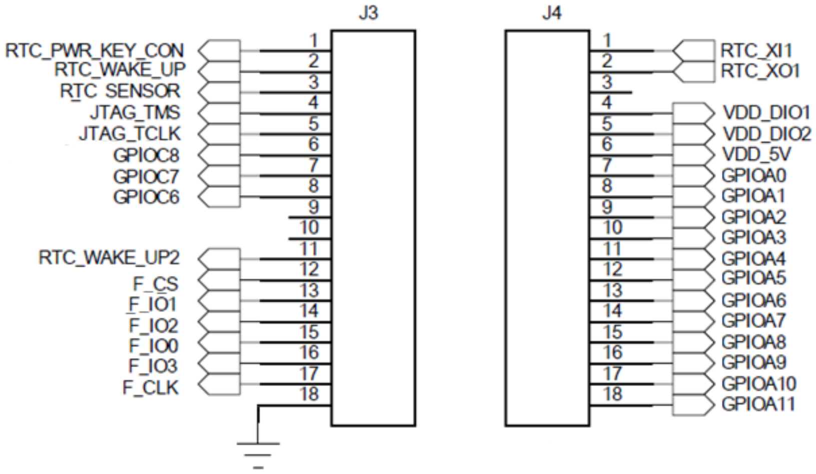


Figure 28: DA16200 Module External Connector

5.9.3 Run Debug Mode

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

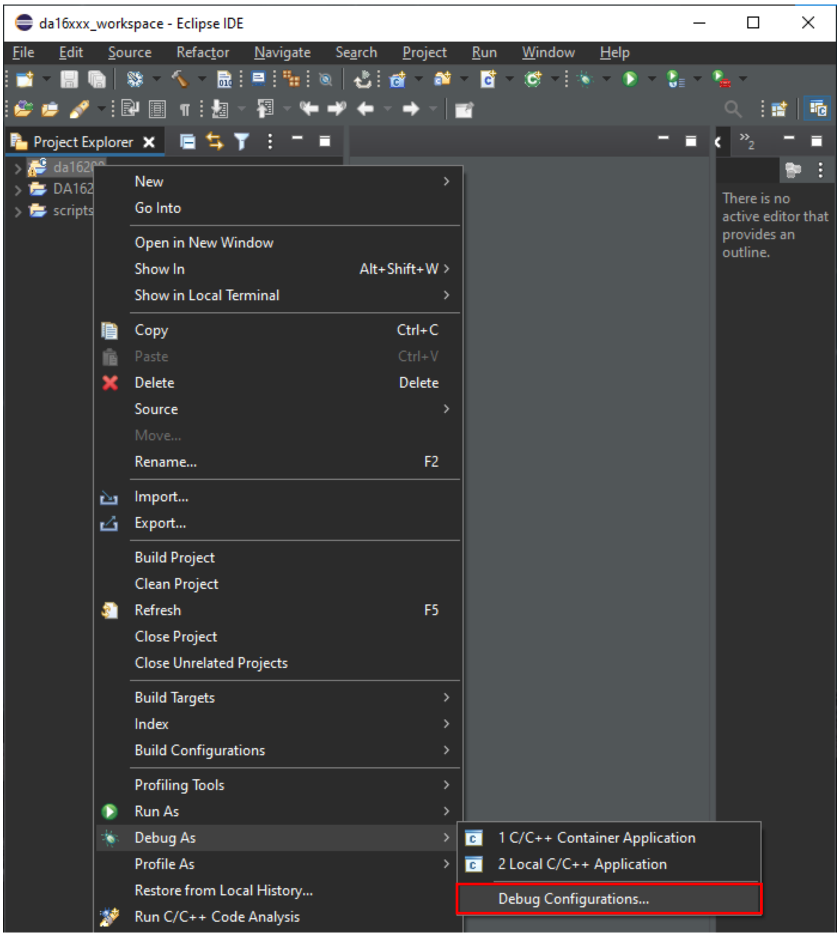
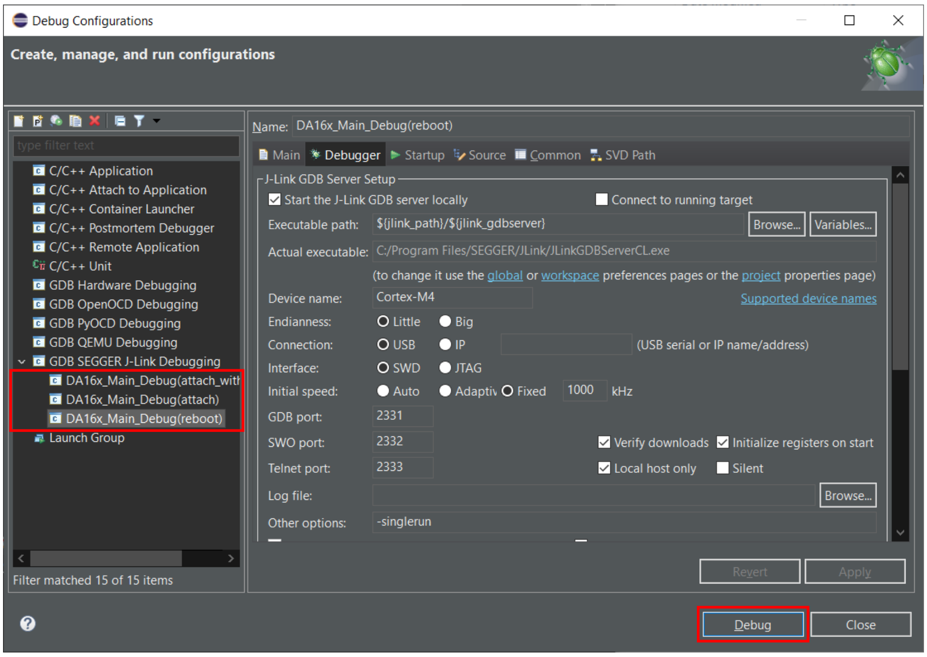


Figure 29. Run Debug Mode

Open the **GDB SEGGER J-Link Debugging** entry in the list and select one of the three debugging modes (reboot, attach, attach with RTOS info) and then select **Debug**.



Debug Configurations

Create, manage, and run configurations

- C/C++ Application
- C/C++ Attach to Application
- C/C++ Container Launcher
- C/C++ Postmortem Debugger
- C/C++ Remote Application
- C/C++ Unit
- GDB Hardware Debugging
- GDB OpenOCD Debugging
- GDB PyOCD Debugging
- GDB QEMU Debugging
- GDB SEGGER J-Link Debugging
  - DA16x_Main_Debug(attach_wit
  - DA16x_Main_Debug(attach)
  - DA16x_Main_Debug(reboot)
- Launch Group

Filter matched 15 of 15 items

Name: DA16x_Main_Debug(reboot)

Main | Debugger | Startup | Source | Common | SVD Path

J-Link GDB Server Setup

☑ Start the J-Link GDB server locally ☐ Connect to running target

Executable path: ${jlink_path}/${jlink_gdbserver} Browse... Variables...

Actual executable: C:/Program Files/SEGGER/JLink/JLinkGDBServerCL.exe

(to change it use the global or workspace preferences pages or the project properties page)

Device name: Cortex-M4 Supported device names

Endianness: ○ Little ● Big

Connection: ○ USB ● IP (USB serial or IP name/address)

Interface: ○ SWD ● JTAG

Initial speed: ● Auto ● Adaptiv ○ Fixed 1000 kHz

GDB port: 2331

SWO port: 2332 ☑ Verify downloads ☑ Initialize registers on start

Telnet port: 2333 ☑ Local host only ☐ Silent

Log file: Browse...

Other options: -singlerun

Figure 30. Select Debug Mode

There are three debug methods supported by J-Link: "reboot", "attach" and "attach with RTOS info" which are defined as follows:

- **DA16x_Main_Debug (Reboot) / Linux_DA16x_Main_Debug(reboot)**
  In this mode, the “debugger” is executed after rebooting the image stored in SFLASH.
  In this mode, “watchdog” is turned off and “wdt_kicking thread” is not executed
- **DA16x_Main_Debug (Attach) / Linux_DA16x_Main_Debug(attach)**
  This mode executes the “debugger” in attach mode without rebooting the image currently stored in SFLASH

NOTE

Before using “Attach” mode, first turn off the “watchdog” using the “sys.wdog off” command as follows:
`[/DA16200] # sys.wdog off`
`WATCHDOG off`

- **DA16x_Main_Debug (attach_with_rtos_info) / Linux_DA16x_Main_Debug(attach_with_rtos_info)**
  Same as attach mode but displays “thread” information when in the debugger suspend state

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

5.10 Programming DA16200 with SEGGER J-Link in Eclipse

This chapter describes how to setup and program the DA16200/600 firmware using the SEGGER J-link debug probe.

5.10.1 Requirements

1. J-Link Lite or higher.

   See: https://www.segger.com/products/debug-probes/j-link/models/model-overview/
2. J-Link software V6.98 or later.

   See: https://www.segger.com/downloads/jlink/
3. Eclipse 2021-06 (4.20.0) or later (see Section: 5.4).

5.10.2 Preparation

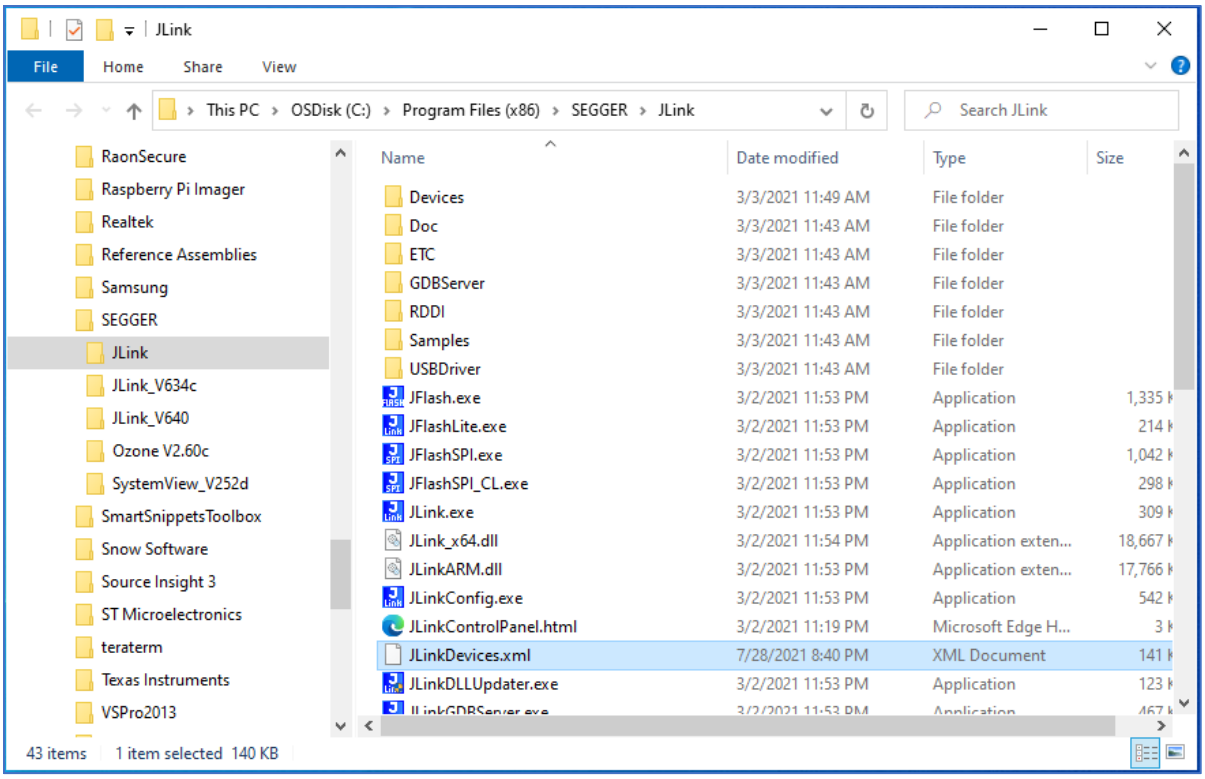
The J-Link setup procedure required to work with J-Flash is described in chapter 2 of the J-Link / J-Trace User Guide (UM08001).

5.10.3 Setup

5.10.3.1 J-Link Flash Loader Installation

In order to use the flash loader for DA16200, it should be installed with the following steps.

1. Browse to the installation directory of the J-Link software package (ex. `C:\Program Files (x86)\SEGGER\JLink`) and locate the `JLinkDevices.xml` file.



Name	Date modified	Type	Size
Devices	3/3/2021 11:49 AM	File folder	
Doc	3/3/2021 11:43 AM	File folder	
ETC	3/3/2021 11:43 AM	File folder	
GDBServer	3/3/2021 11:43 AM	File folder	
RDDI	3/3/2021 11:43 AM	File folder	
Samples	3/3/2021 11:43 AM	File folder	
USBDriver	3/3/2021 11:43 AM	File folder	
JFlash.exe	3/2/2021 11:53 PM	Application	1,335 K
JFlashLite.exe	3/2/2021 11:53 PM	Application	214 K
JFlashSPI.exe	3/2/2021 11:53 PM	Application	1,042 K
JFlashSPI_CL.exe	3/2/2021 11:53 PM	Application	298 K
JLink.exe	3/2/2021 11:53 PM	Application	309 K
JLink_x64.dll	3/2/2021 11:54 PM	Application exten...	18,667 K
JLinkARM.dll	3/2/2021 11:53 PM	Application exten...	17,766 K
JLinkConfig.exe	3/2/2021 11:53 PM	Application	542 K
JLinkControlPanel.html	3/2/2021 11:19 PM	Microsoft Edge H...	3 K
JLinkDevices.xml	7/28/2021 8:40 PM	XML Document	141 K
JLinkDLLUpdater.exe	3/2/2021 11:53 PM	Application	123 K
JLinkGDBServer.exe	3/2/2021 11:53 PM	Application	467 K

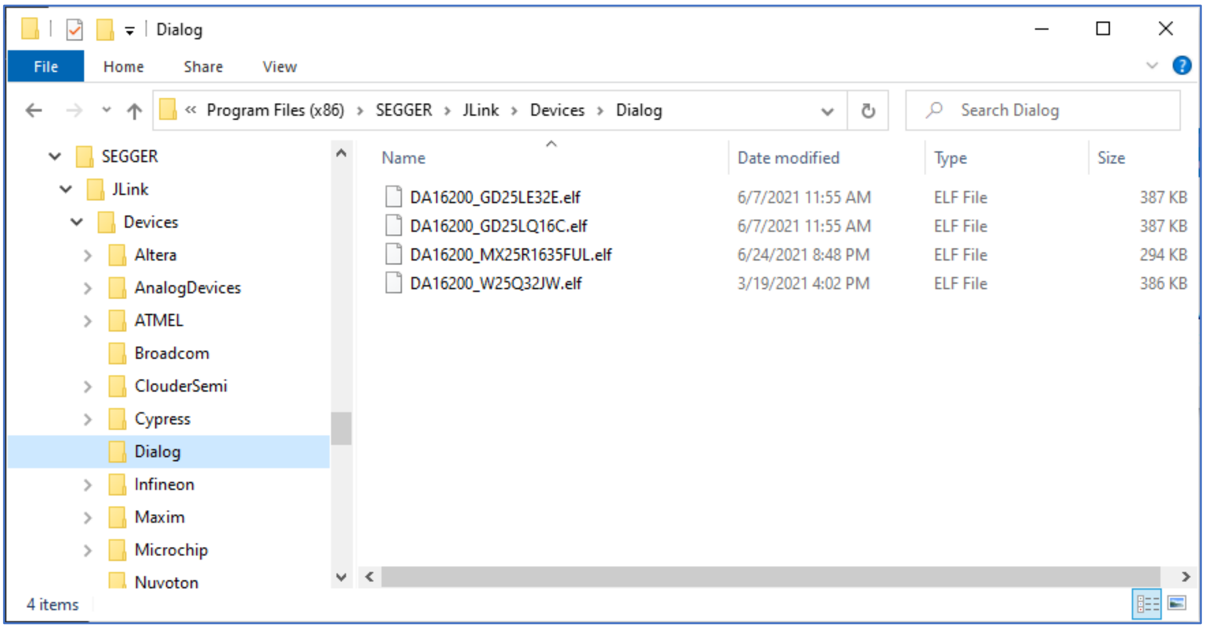
Figure 31: The installation of the J-Link Software Package

2. Open `JLinkDevices.xml` in a text editor and add a device entry for the DA16200 to the **<DataBase>** section as follows:

   `C:\Program Files (x86)\SEGGER\JLink_V640\JLinkDevices.xml`

```
<DataBase>
  <!--                    -->
  <!-- Dialog Semiconductor -->
  <!--                      -->
  <Device>
   <ChipInfo Vendor="Dialog Semiconductor" Name="DA16200_W25Q32JW" Core="JLINK_CORE_CORTEX_M4"
WorkRAMAddr="0x80000" WorkRAMSize="0x00040000" />
   <FlashBankInfo Name="QSPI Flash" BaseAddr="0x100000" MaxSize="0x400000"
Loader="Devices/Dialog/DA16200_W25Q32JW.elf" LoaderType="FLASH_ALGO_TYPE_OPEN" />
</Device>

  <!--                   -->
  <!-- Altera -->
  <!--                   -->
  <Device>
    <ChipInfo Vendor="Altera" Name="CYCLONE V"  Core="JLINK_CORE_CORTEX_A9"
JLinkScriptFile="Devices/Altera/Cyclone_V/Altera_Cyclone_V.JLinkScript"/>
  </Device>
  <!--                   -->
  <!-- AmbiqMicro -->
  <!--                   -->
  <Device>
    <ChipInfo Vendor="AmbiqMicro" Name="AMA3B1KK-KBR" Core="JLINK_CORE_CORTEX_M4"
JLinkScriptFile="Devices/AmbiqMicro/AmbiqMicro_Apollo3.pex" />
  </Device>
  .
  .
  .
```

3. Copy the flash loader files from the `<sdk_root_directory>/utility/j-link/scripts/flashloader/Devices/Dialog` directory (which are referenced in the `JLinkDevices.xml` entry) into the directory where the `JLinkDevices.xml` is located.

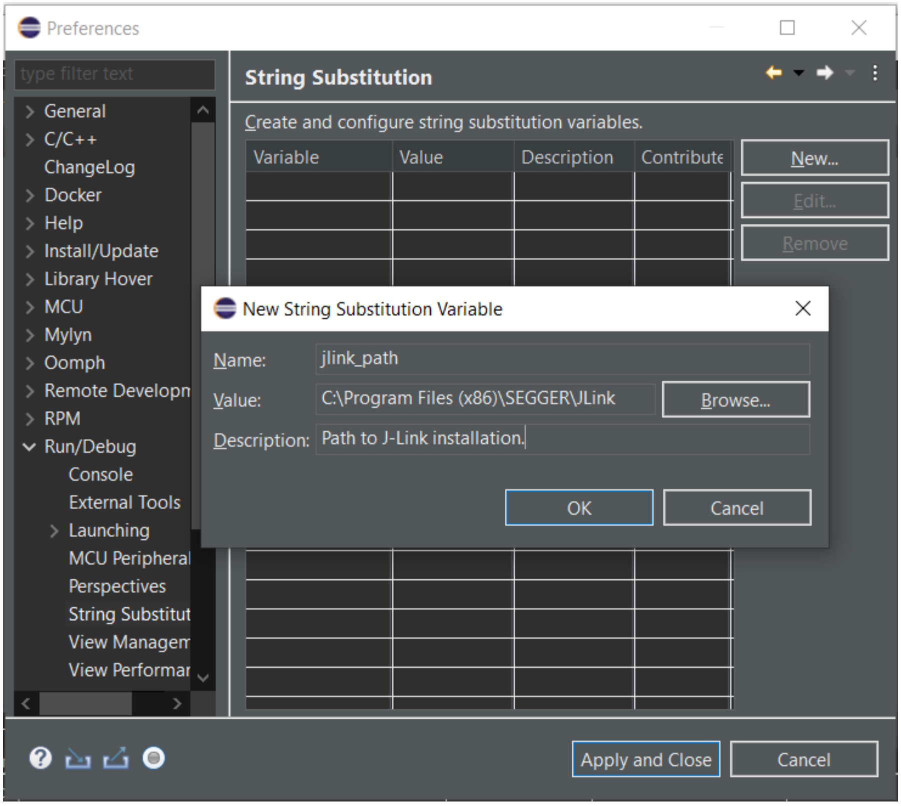


Name	Date modified	Type	Size
DA16200_GD25LE32E.elf	6/7/2021 11:55 AM	ELF File	387 KB
DA16200_GD25LQ16C.elf	6/7/2021 11:55 AM	ELF File	387 KB
DA16200_MX25R1635FUL.elf	6/24/2021 8:48 PM	ELF File	294 KB
DA16200_W25Q32JW.elf	3/19/2021 4:02 PM	ELF File	386 KB

Figure 32: The Flash Loader Files

5.10.3.2 Setup the Path to the J-Link Installation in Eclipse

1. In Eclipse under the **Window** menu, select **Preferences.**
2. In the Preferences dialog box, find the **Run/Debug** -> **String Substitution** settings.
3. Select 'New...' to create a new string substitution.
4. Enter `jlink_path` as the variable name and set the value to the path where the J-Link software is installed (ex. `C:\Program Files (x86)\SEGGER\JLink`).
5. Click **OK** and the **Apply and Close** to save the setting.



Create and configure string substitution variables.

New String Substitution Variable

Name: jlink_path

Value: C:\Program Files (x86)\SEGGER\JLink [Browse...]

Description: Path to J-Link installation.

Figure 33: The Path of J-Link Installation in Eclipse

5.10.4 Programming

To program binaries into the target device, run the **program_all_jtag_win** script for Windows or the **program_all_jtag_linux** script for Linux in the External Tools.

1. Select the project to be programmed in the Project Explorer.
2. Under the **Run** menu, select **External Tools**.
3. Select '**program_all_jtag_win**' or '**program_all_jtag_linux**'.

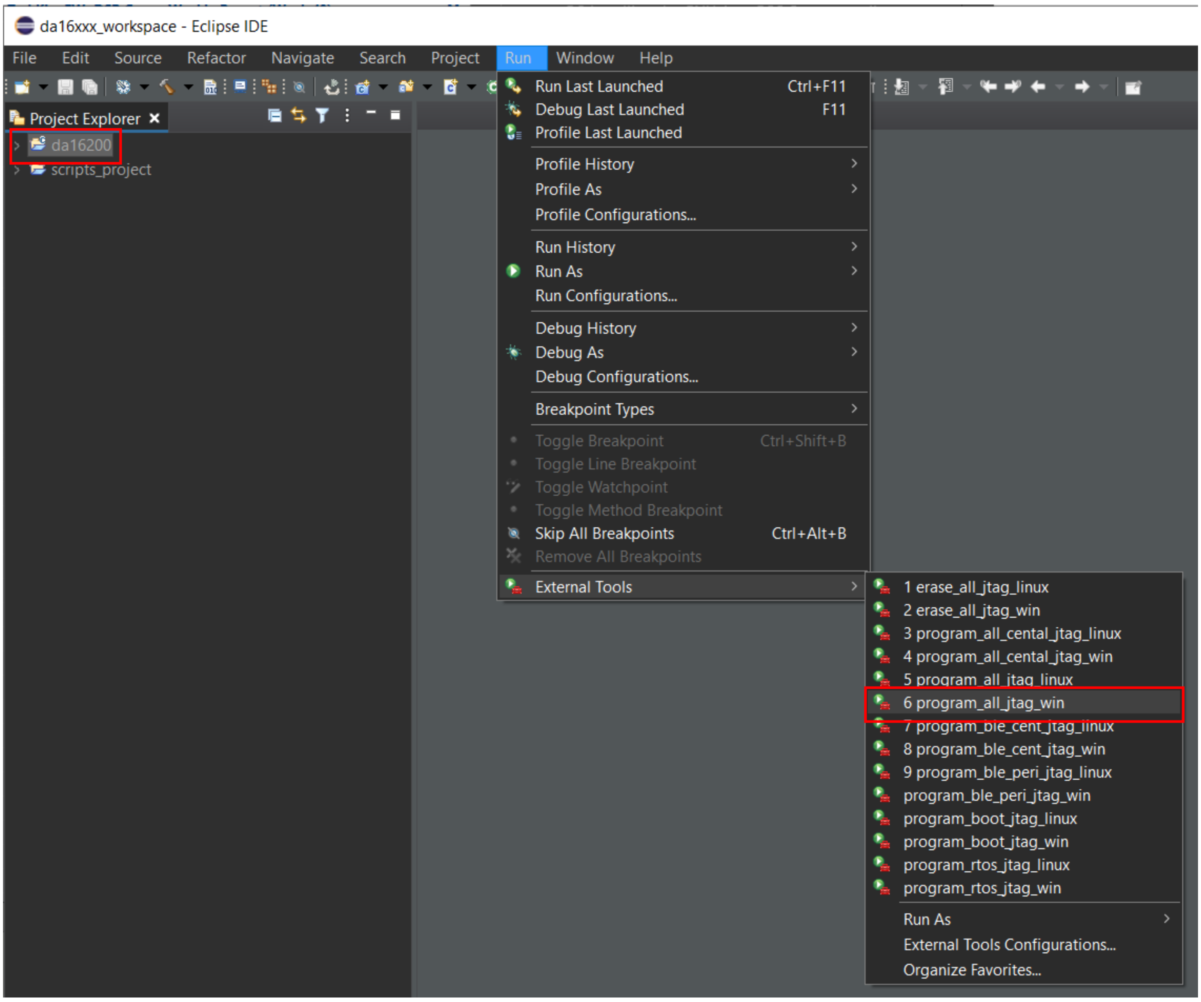


Figure 34: Run program_all_jtag_win Script

The following scripts are included in the SDK:

- erase_all_jtag_win, erase_all_jtag_linux : Erase all contents of the flash memory
- program_all_jtag_win, program_all_jtag_linux: Program all images into the flash memory
- program_boot_jtag_win, program_boot_jtag_linux: Program BOOT image into the flash memory
- program_rtos_jtag_win, program_rtos_jtag_linux: Program RTOS image into the flash memory
- program_all_central_jtag_win, program_all_central_jtag_linux: Program all images for BLE central role into the flash memory. (For DA16600)
- program_ble_peri_jtag_win, program_ble_peri_jtag_linux: Program BLE image for a peripheral role into the flash memory. (For DA16600)
- program_ble_cent_jtag_win, program_ble_cent_jtag_linux: Program BLE image for a central role into the flash memory. (For DA16600)

6 Test Procedures

6.1 Introduction

The following sections describe several tests that will verify the proper operation of the DA16200/600 and demonstrates its various features.

The tests include:

- Ping Test – Verifys the connection between the DA16200/600 and another device connected to the same AP
- Throughput Test – Demonstrates the Wi-Fi performance of the DA16200/600
- DPM Test – Demonstrates the various power modes of the DA16200/600
- Current Test – Demonstrates the amount of power used when the DA16200/600 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 DA16200/600 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 EVK and a laptop. The two stations must be connected to the same sub-network AP as shown in Figure 35.

The DA16200 must be connected to the AP via WIFI, and the laptop must be connected to the AP with an Ethernet cable.

After the DA16200/600 is configured it will go directly into DPM Sleep mode (sleep 3) and will only wake up from sleep 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 DA16200/600. This shows that when a DA16200/600 is in DPM Sleep mode, it can successfully wake up and receive the unicast packets in real-time.

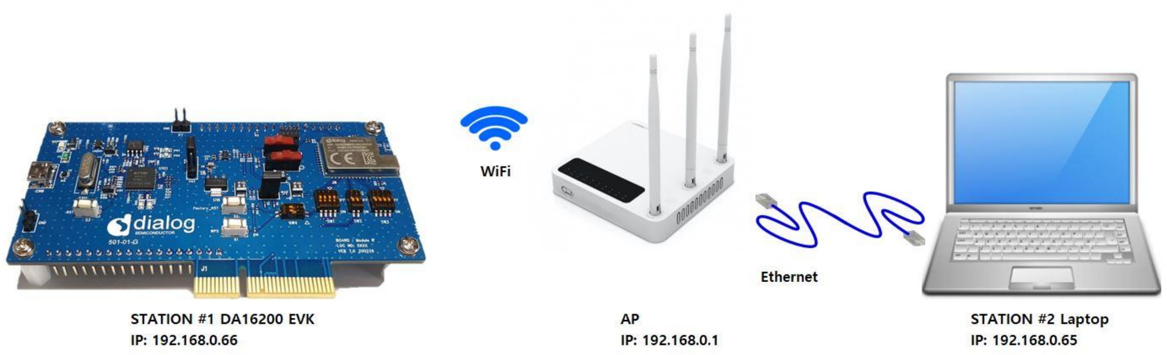


Figure 35: Ping Test Environment

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 DA16200/600 to operate in Station Mode as described in Section 4.6.1.

   Once the DA16200/600 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 DA16200/600 is `192.168.0.66`.

4. Verify the setup.

   Using the ip address of the DA16200/600 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 the 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 DA16200/600 is in Power-Down mode.

2. Run the ping command on the Laptop.

Using the ip address of the DA16200/600 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 DA16200/600 will wake up and print a message on the serial terminal and then return to sleep 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 an ARP Record

If the DA16200/600 is in DPM sleep 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 DA16200/600.

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.

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 DA16200/600 which is operating in DPM sleep 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 you set the ARP cache to static with the command `arp -s` on higher versions of Windows, you may get an error like `Failed to add ARP entry, Access is denied`. It is recommended to use 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

DA16200 has a command `iperf` for measuring packet transfer performance. This is known as the throughput test.

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.

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

6.3.1 Test Setup

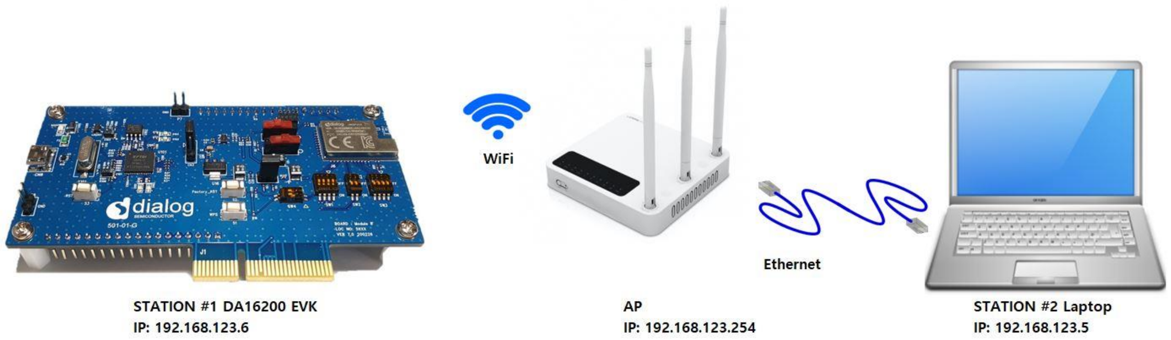


Figure 36: Iperf Test Environment

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

Do the following steps to setup Iperf tool:

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 Iperf Test with DA16200/600 as a Client

To set-up the Iperf test with the DA16200/600 operating as a client and the laptop operating as a server, do the following steps:

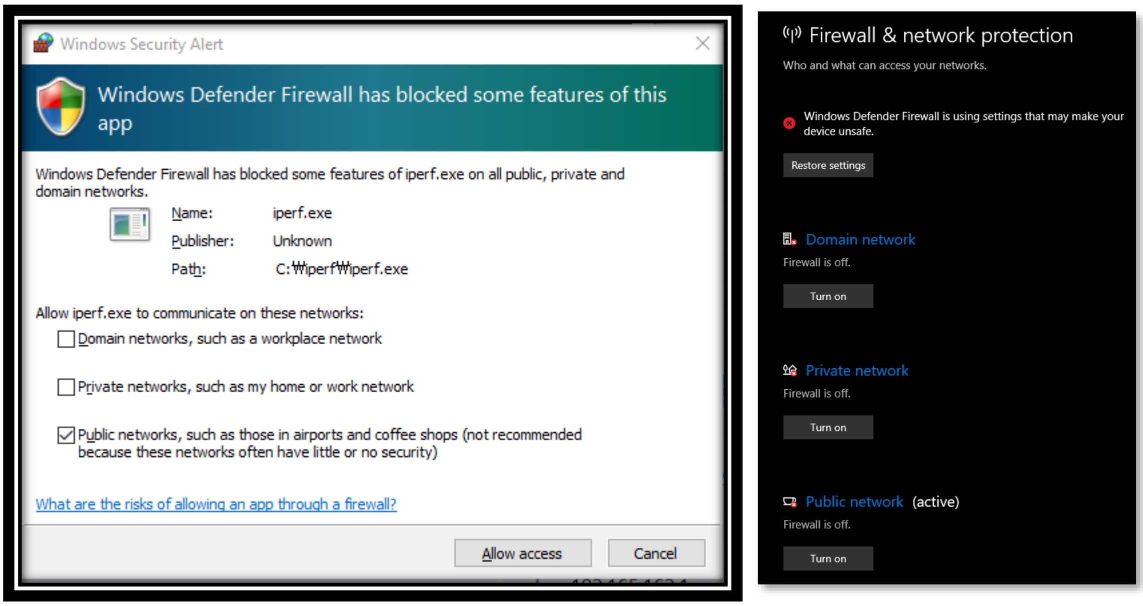
1. Connect the laptop which is the server to the AP.
2. In the CMD window, use the command `ipconfig /all` to find the IP address:

```
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 (see Figure 37).



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 37: 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 DA16200/600 to start the client test app.

5. In the DA16200/600 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 DA16200/600:

```
[/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 Iperf Test with DA16200/600 as a Server

To set-up the Iperf test with the DA16200/600 operating as a server and the laptop operating as a client, do the following steps:

1. Power on the DA16200/600 and check the assigned IP address in the DA16200/600 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 DA16200/600 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 DA16200/600 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 DA16200/600 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

DPM (Dynamic Power Management) 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 and 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 DA16200/600 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 (in the case 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 DA16200/600 will enter DPM sleep 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 DA16200/600 is in DPM sleep 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 DA16200/600 is in DPM sleep mode, user input is not accepted by the debug terminal. This is because the UART interfaces are powered down during DPM Sleep.

To exit this state and start over with `setup`, do the following:

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 do the following:
   a. Paste the `dpm hold` string in the terminal window and immediately press 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 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 DA16200/600 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] #
```

6.5 Current Measurement

For more detailed information on the Sleep modes, see the “Low Power Operation Mode” section in the DA16200 Datasheet Ref. [1].

To measure the current waveform, connect the EVK's current measurement point (P2) with the measurement instrument (KEYSIGHT 14585A).

6.5.1 Test Setup

Figure 38 shows a typical test setup environment.

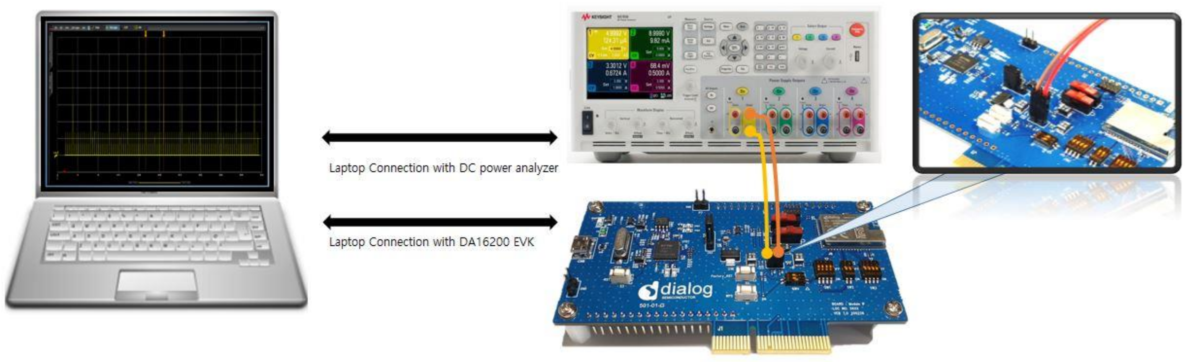

Laptop Connection with DC power analyzer
Laptop Connection with DA16200 EVK
dialog

Figure 38: Current Test Environment

6.5.2 Sleep 1

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

6.5.3 Sleep 2

To measure the Sleep 2 current use either the CLI command or the AT command as described in the following section.

6.5.3.1 CLI Command

1. To measure the Sleep 2 current, power on the EVK by setting **RTC_PWR_KEY** to the **ON** position (**SW6** on the DA16200 EVK or **SW2** on the DA16600 EVK).
2. Type the `factory` command to set the DA16200/600 to its default settings.
3. After the EVK 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 sleep for a set amount of time (10 seconds), and then reboot and wake up

6.5.3.2 AT Command

‘sys.hal’ mode is not supported in AT-CMD image and sleep 2 current measurement is possible over AT command. Do the following steps and refer to “UM-WI-003_DA16200_AT-Command_User_Manual” for more details on how to use the AT command on DA16200.

1. Connect UART1 of DA16200 for AT-CMD. Refer to the relevant section “AT Command Development Environment Configuration” in the document in UM-WI-003.
2. Run the following AT commands.
   - `ATE     //ECHO on/off`
   - `AT+SETSLEEP2EXT=<period>,<retain_dpm_memory>`
   - For instance, `AT+SETSLEEP2EXT=10,0`
   - `It will sleep for a set amount of time (10 seconds), and then reboot and wake up. During the sleep mode, DPM memory is not retained and this is basic sleep 2 operation.`
   - `When the DPM memory is retained, the sleep current is same with sleep 3 mode. The fast reconnection functionality is possible in this mode.`
3. Read the explanation of “AT+SETSLEEP2EXT” in the following table. Also see to the relevant section “Basic Function Commands” in the document UM-WI-003 for details.

Command	Parameters	Description
AT+SETSLEEP2EXT	<period>,<retain_dpm_memory>	Enter DPM Sleep 2 mode for the period specified. <period>: wakeup timeout, in second(s) <retain_dpm_memory>: 1 (retain), 0 (not retain) Response: OK or ERROR
	Pre-requisites None. *Example* AT+SETSLEEP2EXT=10,1 OK +INIT:DONE,0 *Note* DA16200 can be woken up by RTC_WAKE_UP while it’s in sleep. DA16200 send the “+INIT:DONE:0” response once it wakes up.	

NOTE

‘AT+SETSLEEP2EXT’ is not possible in the old version of SDK. Try with ‘AT+SETDPMSLP2EXT’ in the old version of SDK.

- For instance, `AT+SETSLEEP2EXT=10,0`

6.5.4 Sleep 3

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

For example: the current waveform in Figure 39 shows settings DTIM 10 (about 1sec @ AP DTIM=1) and Keep Alive 30s.

Sleep 3 current means current between RX or between RX and TX.

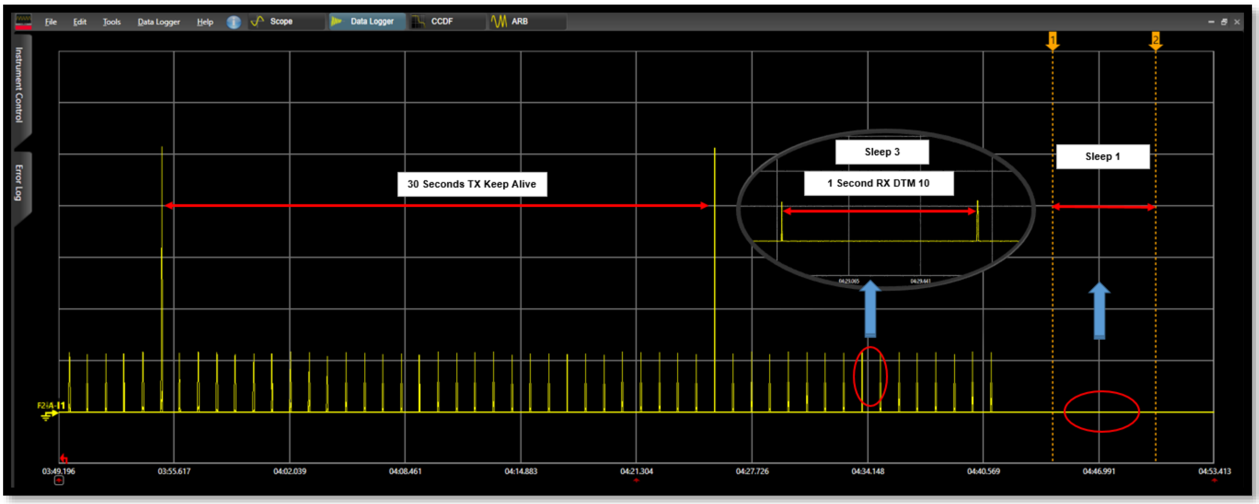


Figure 39: Current Measurement with DPM

Appendix A Configure DA16600 EVK to Use AT-CMD via UART

DA16600MOD supports AT command via the DA16200's UART. DA16200 supports 3 UARTs and UART2 should be used for AT-CMD. UART0 is used for debug and UART1 is used for communication between DA16200 and DA14531 on DA16600MOD.

UART2 is not connected to FT2232H on EVK so additional manual modifications are required to use AT-CMD via UART. See below steps.

A.1 Turn OFF Switch

UART1 is connected to FT2232H on the board, so switch needs to be turned off first.

To disconnect UART1 from FT2232H, SW3 and 5~8 of SW should be turned off. See Figure 40.

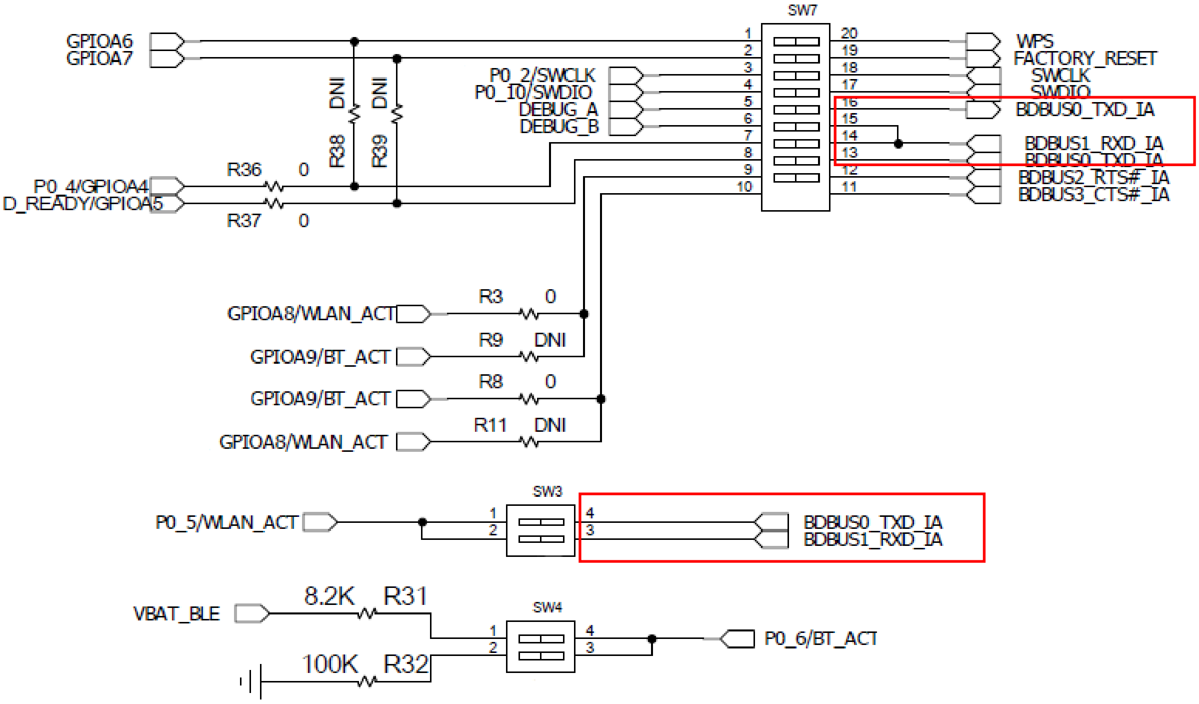
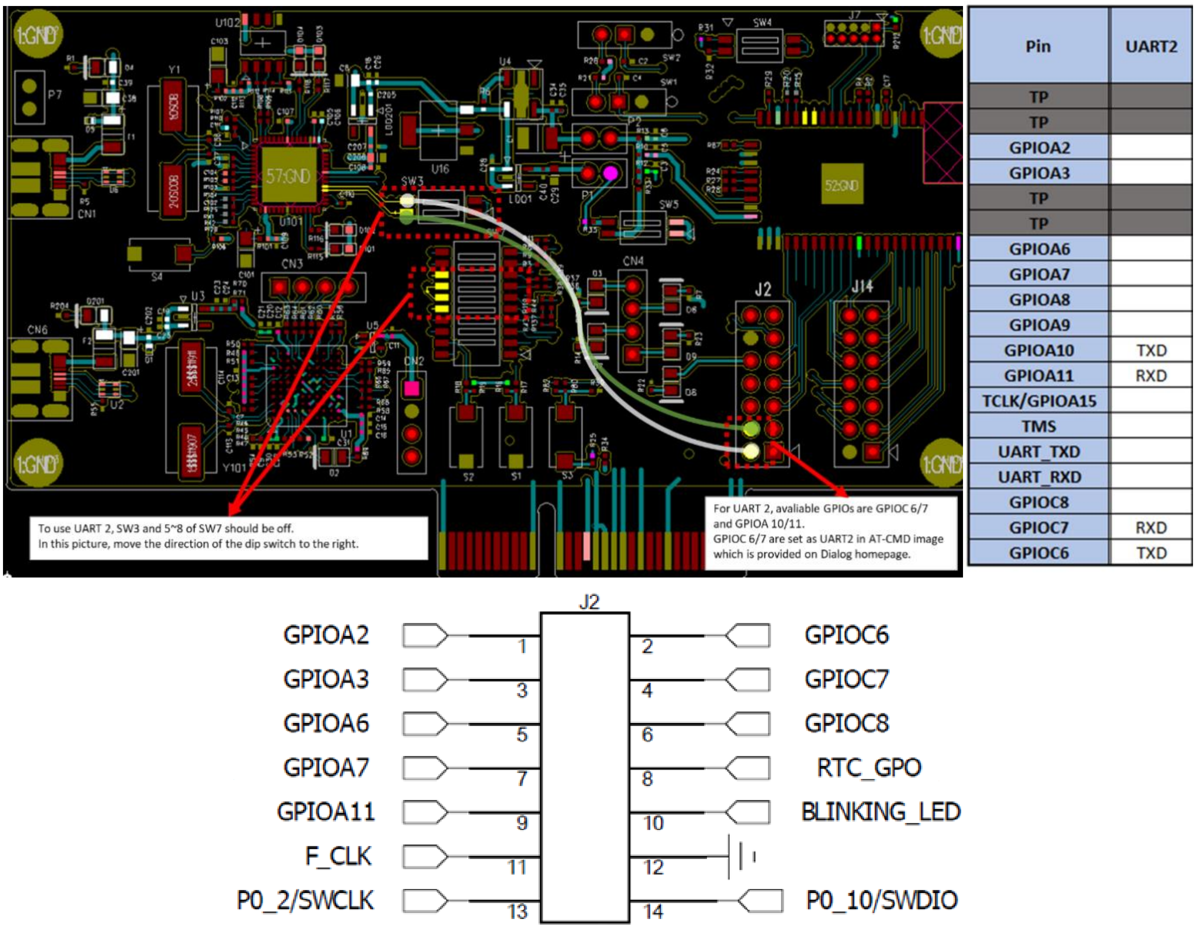


Figure 40: SW3 and SW7 UART1 Connection

A.2 Connect UART2 to FT2232H

UART2 is available with GPIOA10/11 or GPIOC6/7. See pin mux in Figure 41. GPIOC6/7 are set as UART2 in AT-CMD image, so UART2 can be connected to FT2232H to use AT-CMD. The image is available on the Dialog website. (https://www.dialog-semiconductor.com/products/wi-fi/da16600-modules#tab-field_tab_content_resources)

GPIOC6/7 are 2/4 pins of J2 connector, connect the wires from SW3 to J2. See Figure 41.



To use UART 2, SW3 and 5~8 of SW7 should be off.
In this picture, move the direction of the dip switch to the right.

For UART 2, avaliable GPIOs are GPIOC 6/7 and GPIOA 10/11.
GPIOC 6/7 are set as UART2 in AT-CMD image which is provided on Dialog homepage.

Pin	UART2
TP	
TP	
GPIOA2	
GPIOA3	
TP	
TP	
GPIOA6	
GPIOA7	
GPIOA8	
GPIOA9	
GPIOA10	TXD
GPIOA11	RXD
TCLK/GPIOA15	
TMS	
UART_TXD	
UART_RXD	
GPIOC8	
GPIOC7	RXD
GPIOC6	TXD

Figure 41: Connect UART2 to FT2232H

See Figure 42 for the picture of AT-CMD connection via UART on EVK.

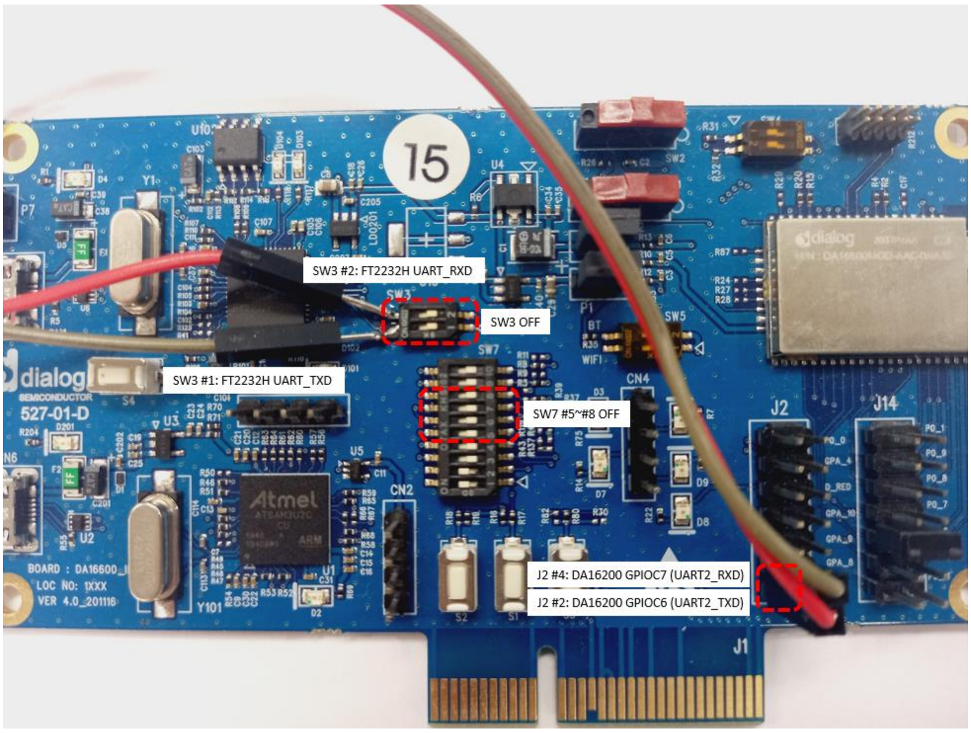


Figure 42: Setting for AT-CMD over UART2

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 follow:

- 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 commands 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 9: 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
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 enable or disable ● 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 10: Network Commands

Command	Parameter	Description

`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

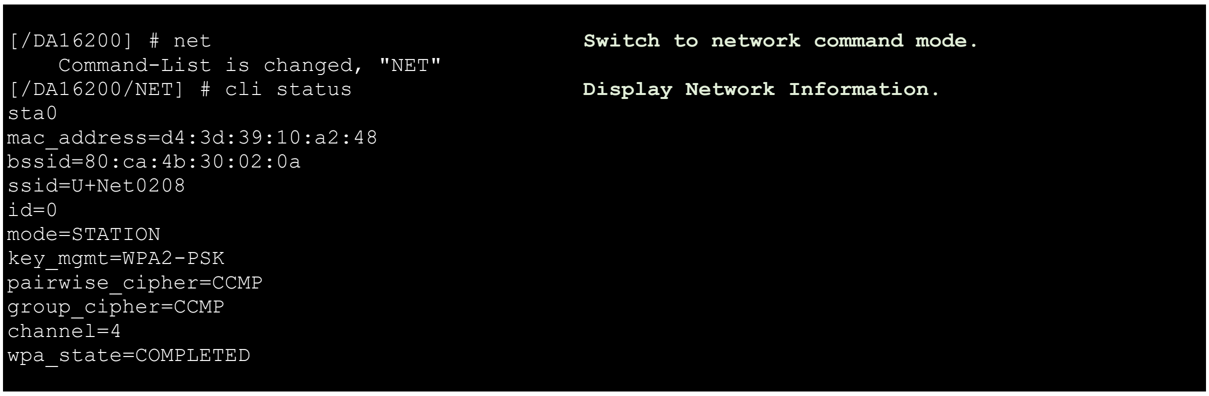
`ping`	`-I [interface wlan0|wlan1] [domain|ip] -n [count] -l [size] -w [timeout] -i [interval]`	Ping test to the target address with a certain option ● `[interface wlan0|wlan1]`: ○ Network interface. With no designated interface, an interface for a subnet band of the same destination IP address is designated ● `[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`
`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`

Command	Parameter	Description
dhcpd	[interface] [options]	DHCP server setting ● 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
iperf	-I [interface] [-s|-c host] [options]	Setup Iperf client/server
cli	[options]	Refer to 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
act	[on | off]	Start or stop DPM Auto Configuration

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 CLI (Command Line Interface) 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 11: 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` (option)
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) For example: `[/DA16200/NET] # cli remove_network 1` ○ Delete a profile for AP Mode

CLI	Parameter	Description
set_net work	<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_net work	<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` ○ KR

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

B.2.4 STA Commands

Table 12: 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 (option) 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 13: 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` 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 (1~14) 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`

Command	Parameter	Description

`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` ○ 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

Command	Parameter	Description
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
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

Command	Parameter	Description
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 14: Advanced CLI Commands

Command	Parameters	Description
wps_pbc	(none)	Run WPS PBC (Push Button Configuration)
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 15: 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 16: 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 17: 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 18: 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 19: 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 20: Additional Commands in MROM

Command	Parameters	Description
bmcfg	[op] [bootmode]	Boot mode config Not used

Command	Parameters	Description
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 DA16200/600 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 DA16200/600 SDK currently supports the following SFLASH types: ● **4-MB SFLASH** : Adesto - AT25SL321 : GigaDevice - GD25LE32E : ISSI - IS25LP032D, IS25LQ032B : MXIC - MX25L3233F, MX25L25635F, MX25R3235F, MX25U3232F, MX25U3235F : Winbond - W25Q32JV, W25Q32JW To use a different type of SFLASH with the DA16200/600, contact Dialog Semiconductor to confirm compatibility.

C.2 SFLASH Memory Map

The DA16200/600 FreeRTOS SDK only supports a 4-MB SFLASH memory map.

Table 21: 4-MB SFLASH Memory Map

Address	Description	Size
0x0000_0000	2nd Bootloader	136 KB
0x0002_2000	Boot Index	4KB
0x0002_3000	RTOS #0	1792KB
0x001E_2000	RTOS #1	1792KB
0x003A_2000	Debug/RMA Certificate	4KB
0x003A_3000	TLS Certificate Key #0	16KB
0x003A_7000	TLS Certificate Key #1	16KB
0x003A_B000	NVRAM	4KB
0x003A_C000	NVRAM Backup	4KB

NOTE
Technical support is required to use a 2-MB SFLASH memory map in the FreeRTOS SDK.

The DA16200/600 SDK contains a script to automatically select the SFLASH type. See section 5.7.1 for details on how the SFLASH is selected in Eclipse.

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 43.

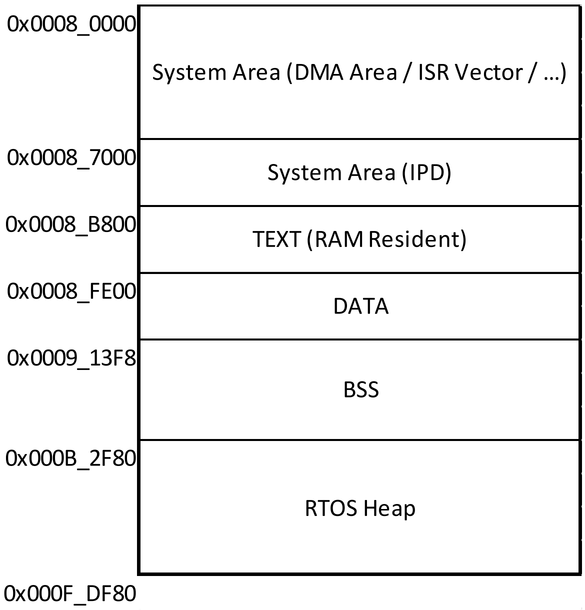


Figure 43: System Memory Map

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 downloading the firmware which could cause the DA16200/600 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 DA16200/600 requires the SFDP for the SFLASH to operate properly. The SFDP parameters are included in the FBOOT firmware image and can be recovered by downloading 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 firmware images (both the FBOOT and the FRTOS images) must be downloaded again. See section 4.5. Recovery from Unresponsive Boot

If the DA16200/600 does not boot and the serial command interface is unresponsive then access can be regained using the following steps:

1. Run the terminal program and connect to the DA16200/600 as normal.
2. Power off the DA16200/600 using the RTC_PWR_KEY (move it to the OFF position).
3. In the terminal program repeatedly press ‘ESC’ key and at the same time power on the DA16200/600 using the RTC_PWR_KEY (move it to the ON position).
4. The DA16200/600 will detect the ‘ESC’ key and boot into [MROM] mode.
5. After entering [MROM] mode, follow the instructions in Section 4.5.1 to manually download both the FBOOT and FRTOS firmware images.

NOTE
Do not reset the board between the FBOOT and FRTOS downloads or the download will fail.

Appendix E Country Codes

Table 22: 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
1.1	29-Nov-2021	Title was changed.
1.0	27-Oct-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.

Disclaimer

Unless otherwise agreed in writing, the Dialog Semiconductor products (and any associated software) referred to in this document are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of a Dialog Semiconductor product (or associated software) can reasonably be expected to result in personal injury, death or severe property or environmental damage. Dialog Semiconductor and its suppliers accept no liability for inclusion and/or use of Dialog Semiconductor products (and any associated software) in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Information in this document is believed to be accurate and reliable. However, Dialog Semiconductor does not give any representations or warranties, express or implied, as to the accuracy or completeness of such information. Dialog Semiconductor furthermore takes no responsibility whatsoever for the content in this document if provided by any information source outside of Dialog Semiconductor.

Dialog Semiconductor reserves the right to change without notice the information published in this document, including, without limitation, the specification and the design of the related semiconductor products, software and applications. Notwithstanding the foregoing, for any automotive grade version of the device, Dialog Semiconductor reserves the right to change the information published in this document, including, without limitation, the specification and the design of the related semiconductor products, software and applications, in accordance with its standard automotive change notification process.

Applications, software, and semiconductor products described in this document are for illustrative purposes only. Dialog Semiconductor makes no representation or warranty that such applications, software and semiconductor products will be suitable for the specified use without further testing or modification. Unless otherwise agreed in writing, such testing or modification is the sole responsibility of the customer and Dialog Semiconductor excludes all liability in this respect.

Nothing in this document may be construed as a license for customer to use the Dialog Semiconductor products, software and applications referred to in this document. Such license must be separately sought by customer with Dialog Semiconductor.

All use of Dialog Semiconductor products, software and applications referred to in this document is subject to Dialog Semiconductor's Standard Terms and Conditions of Sale, available on the company website (www.dialog-semiconductor.com) unless otherwise stated.

Dialog, Dialog Semiconductor and the Dialog logo are trademarks of Dialog Semiconductor Plc or its subsidiaries. All other product or service names and marks are the property of their respective owners.

RoHS Compliance

Dialog Semiconductor's 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.

Contact Dialog Semiconductor

General Enquiry:	Local Offices:
Enquiry Form	https://www.dialog-semiconductor.com/contact/sales-offices