DA16200/DA16600 Mass Production

This document explains how to set up the production tests with the DA16200 and DA16600.

Contents

Figures

Tables

1. Terms and Definitions

ADC	Analog to Digital Converter
CW	Continuous Waveform
EVB	Evaluation Board
GUI	Graphical User Interface
OTP	One Time Password
PER	Packet Error Rate
PLL	Phase-Locked Loop
PLT	Production Line Tool
RF	Radio Frequency
SNDR	Signal to Noise and Distortion Ratio
SPDT	Single Pole Double Throw
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter

2. References

[1] DA16200, Datasheet, Renesas Electronics.

[2] UM-WI-056, DA16200 DA16600 FreeRTOS Getting Started Guide, User Manual, Renesas Electronics.

[3] UM-WI-004, DA16200 AT GUI Tool User Manual, User Manual, Renesas Electronics.

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

[5] UM-WI-039, DA16200 DA16600 Multi Downloader, User Manual, Renesas Electronics.

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

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

[8] UM-B-041, SmartBond Production Line Tool, User Manual, Renesas Electronics.

[9] UM-B-083, SmartSnippets™ Toolbox, User Manual, Renesas Electronics.

[10] UM-B-119, DA1453x/DA1458x Software Platform Reference Manual, User Manual, Renesas Electronics.

Note 1 References are for the latest published version, unless otherwise indicated.

3. Overview

This document explains how to set up production tests for the DA16200 and DA16600 as well as the options and limitations that should be considered. Each procedure or the order introduced in this document can be omitted or changed according to the production environment. Figure 1 shows the recommended flow of the production test.

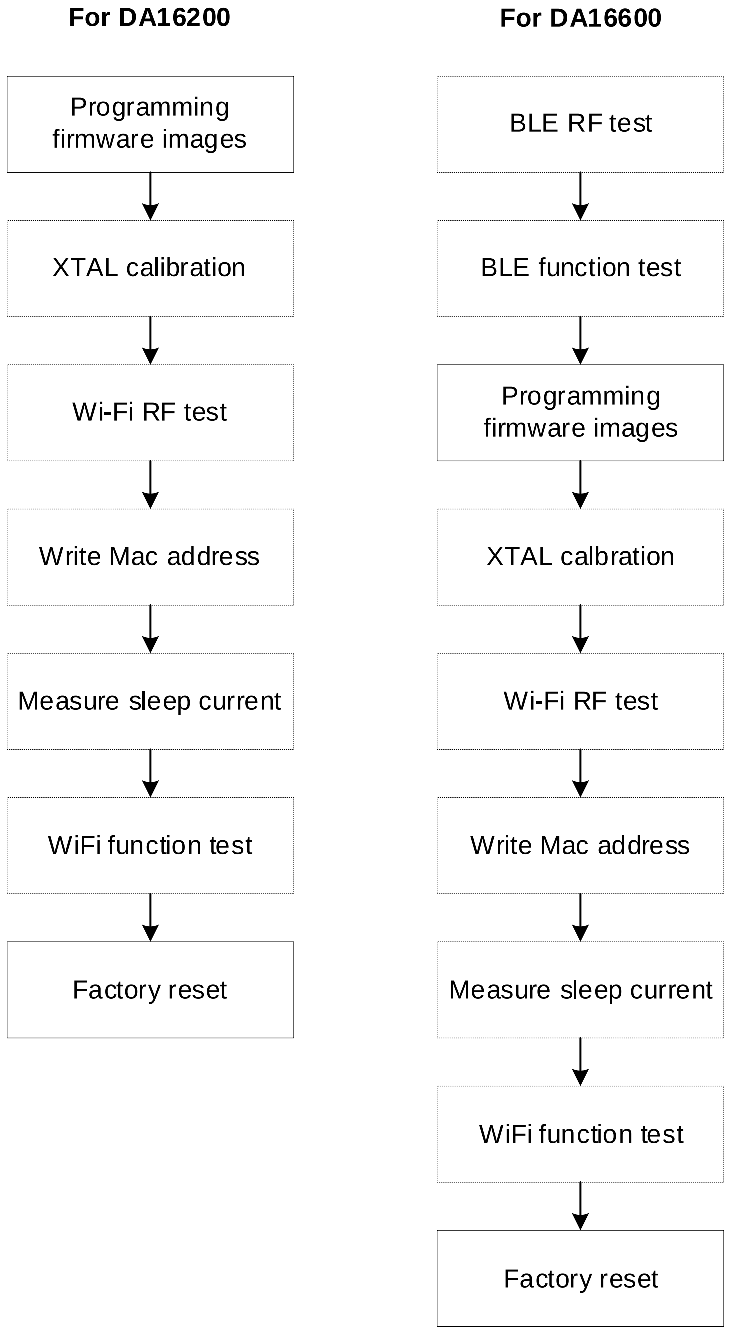

Figure 1. Production test flow for DA16200/DA16600

NOTE
The Flash should be empty when performing Bluetooth® LE test as Bluetooth® LE for the DA16600 needs be tested without firmware image.

4. Interface for Console, AT Command, and AT GUI

The DA16200 and DA16600 have an interface that can use AT command and UART0 interface for debugging. It can also use UART1 or UART2 to run the AT GUI and test Wi-Fi. See Ref. [7] on how to generate firmware image with AT command.

Table 1. Available interfaces for AT commands and Debug

Device	Debug console	AT command	AT GUI
DA16200	UART0	UART1/UART2/SPI/SDIO	UART1/UART2
DA16600	UART0	UART2/SPI/SDIO	UART2

5. Program Firmware Images

The first step is to program firmware images to serial flash memory over UART0 or SPI. For the address of each firmware image on the flash and how to program firmware images, see Ref. [2]. The programming firmware image can be done through SPI using GUI tool in Windows. For details, see Ref. [4].

After programming firmware images, NVRAM initialization can be done using the following console command.

```
[/DA16200] # nvram
[/DA16200/NVRAM] # nvedit erase sflash
[/DA16200/NVRAM] # nvedit clear
[/DA16200/NVRAM] # nvcfg update sflash
update, sflash completed
[/DA16200/NVRAM] # nvedit load sflash
nvedit, load complted
```

And the version of the programmed firmware image can be confirmed using the following console or AT command.

```
[/DA16200] # ver
        *******************************************************
        *              DA16200 SDK Information
        * ---------------------------------------------------
        *
        * - CPU Type          : Cortex-M4 (120 MHz)
        * - OS Type           : FreeRTOS 10.4.3
        * - Serial Flash      : 4 MB
        * - SDK Version       : V3.2.8.0 GEN-ATCMD
        * - F/W Version       : FRTOS-GEN01-01-f017bfdf51-006558
        * - F/W Build Time    : Aug 10 2023 13:44:35
        * - Boot Index        : 0
        *
        *******************************************************

[/DA16200] #                                                                       Firmware for DA16200 FreeRTOS
```

NOTE
You can check the version information by using AT command as shown in Table 2.

Table 2. AT command to check firmware version

Command	Parameters	Description
AT+VER	(none)	Get version information. Response: +VER:<ThredX RTOS version>,< ThredX SLIB version> Response: +VER:<FreeRTOS RTOS version>

```
ate                                           To see the AT command
Echo on
OK
AT+VER
+VER:FRTOS-GEN01-01-56c232799-004158          FreeRTOS verison
```

An efficient way for performing production test using AT commands and programming firmware image for final product at the same time is to program both images to different regions. The DA16200 provides two regions of RTOS image. The running image can be changed using boot index which is stored in flash memory and can be changed by console command or AT command. The default value of the boot index is 0 for the first region. The following steps are how to program both images and proceed production test.

1. Program BOOT firmware image (bootloader).
2. Program RTOS firmware image for AT commands to the first region and firmware image for final product to the second region.
3. Initialize the NVRAM.
4. Perform the production test.
5. After step 4 is complete, reset using the factory reset command ("Factory") for the final product. Optionally, it might need to change the boot index to 0 or 1.

See Ref. [7] on how to create firmware images for AT commands.

6. Wi-Fi XTAL Calibration

The calibration TX power and temperature frequency for the DA16200 are completed during the mass production. This section explains XTAL frequency calibration using AT GUI tool. The purpose of XTAL calibration is to match the desired tone for proper RF transmission and reception as shown in Figure 2.

The register range is 0x00 ~ 0x7F, and there is about 2 kHz deviation per register code.

- Example: if Measured Tone is #1,
  Offset is Measured – Desired (2411994000-2412000000) = -6000, and Offset value is < 0.
  You must decrease the Register value to the Desired Tone.
- Example: if Measured Tone is #2,
  Offset is Measured – Desired (2412006000-2412000000) = +6000, and Offset value is > 0.
  You must increase the Register value to the Desired Tone.

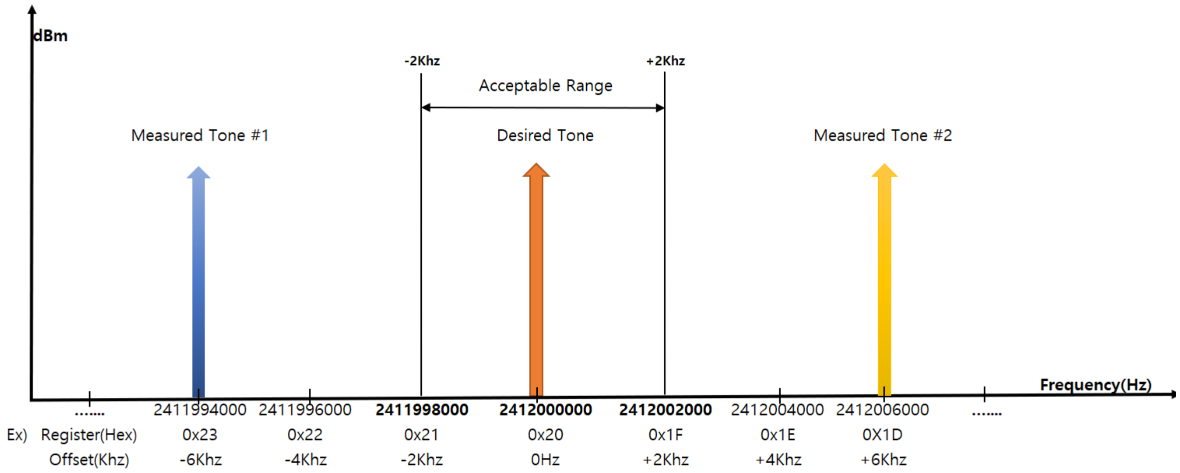

Figure 2. Offset in frequency domain

The procedure for XTAL calibration is:

1. Program firmware image for the AT command.
2. Run the TX test mode and Write default XTAL register with the following AT commands.
   a. AT+TMRFNOINIT=1 // Configure RF test mode
   b. AT+RESTART // Restart the DA16200 to enter RF test mode
   c. AT+RFTESTSTART // RF test start
   d. AT+RFCWTEST=2412,0,0 // Continuous waveform (CW) setting on channel 1
   e. AT+XTALWR=3F // 0x3F is the central register from 0x00 to 0x7F
3. Measure and calculate frequency offset.
   a. Offset = Measured Frequency (by Spectrum Analyzer) – Desire Tone Frequency (2412000000 Hz). For example,
      i. If Measured Frequency is 2412040000 Hz,
         Offset is 2412040000 Hz– 2412000000 Hz = +40000 Hz.
      ii. If Measured Frequency is 2411960000 Hz,
         Offset is 2411960000 Hz– 2412000000 Hz = -40000 Hz.
   b. **The Desire Tone can be 2413000000 Hz depending on the SDK version.**
4. Calculate the starting XTAL register value.
   a. To reduce the calibration time in mass production, it might be required to control the starting value.
   b. Starting XTAL register = Default XTAL register (0x3F) + offset/4000.
      (Frequency offset is expressed as a Decimal value, 40000/4000 = 10 (Dec) = 0x0A (Hex).)
      i. If Offset is +40000 Hz > 0,

Starting XTAL register is 0x3F + 0x0A = 0x49.

Write the value of 0x49 to the register.

AT+XTALWR = 49

c. Starting XTAL register = Default XTAL register - offset/4000.

i. If Offset is -40000 Hz < 0,

Starting XTAL register is 0x3F - 0x0A = 0x35.

Write the value of 0x35 to the register.

AT+XTALWR = 35

5. If the offset is less than +2 kHz and greater than -2 kHz, the value is within the margin. If not, perform step 5 to change the XTAL register value.

NOTE
Main Clock (40 MHz) and Recommend XTAL (+/- 20 ppm) To make 2.4 GHz band, Phase-locked loop (PLL) uses 40 MHz main clock * 60. Additionally, **must trim below 1 ppm to take temperature change into account** because XTAL has a deviation of +/- 20 ppm depending on temperature change. Therefore, Variation: 40000000 * 1/1000000 = 40. From an RF perspective, the value increases to 40 * 60 = 2400, which is 2400 Hz. As a result, it becomes -2.4 kHz ~ Center frequency ~ +2.4 kHz (400 Hz as a margin).

6. Check the offset sign.

When calculated, the offset value has a plus (+) or minus (-) sign value.

If offset is plus (offset > 0), increase XTAL register. It can be increased < Max 0x7F (Dec: 127).

a. Example:
   i. AT+XTALWR = 50 (First time)
   ii. AT+XTALWR = 51 (Second time)
   iii. AT+XTALWR = 52 (Third time)

If offset is minus (offset < 0), decrease XTAL register. It can be decreased < Min 0x00 (Dec: 0).

b. Example:
   i. AT+XTALWR = 35 (First time)
   ii. AT+XTALWR = 34 (Second time)
   iii. AT+XTALWR = 33 (Third time)

7. Write the final value of XTAL register with AT commands to OTP memory.
   a. AT+UOTPWRASC = 0428,1, <XTAL register value>
   b. Example: AT+UOTPWRASC = 0428,1,33

The DA16200 has two slots to store the XTAL offset in the OTP memory (see Table 3). To use AT commands to write value at OTP address, address x 4 should be taken because address is 4-byte aligned address. For more details, see Ref. [7] for details of OTP commands.

Table 3. OTP address for XTAL offset

Slot	OTP address	Address for AT command	Size (Byte)
XTAL Offset #0	0x10A	428	2
XTAL Offset #1	0x10B	42c	2

Figure 3 shows the procedure of XTAL calibration. Given the sample variability, Renesas does not recommend using the same values for all samples.

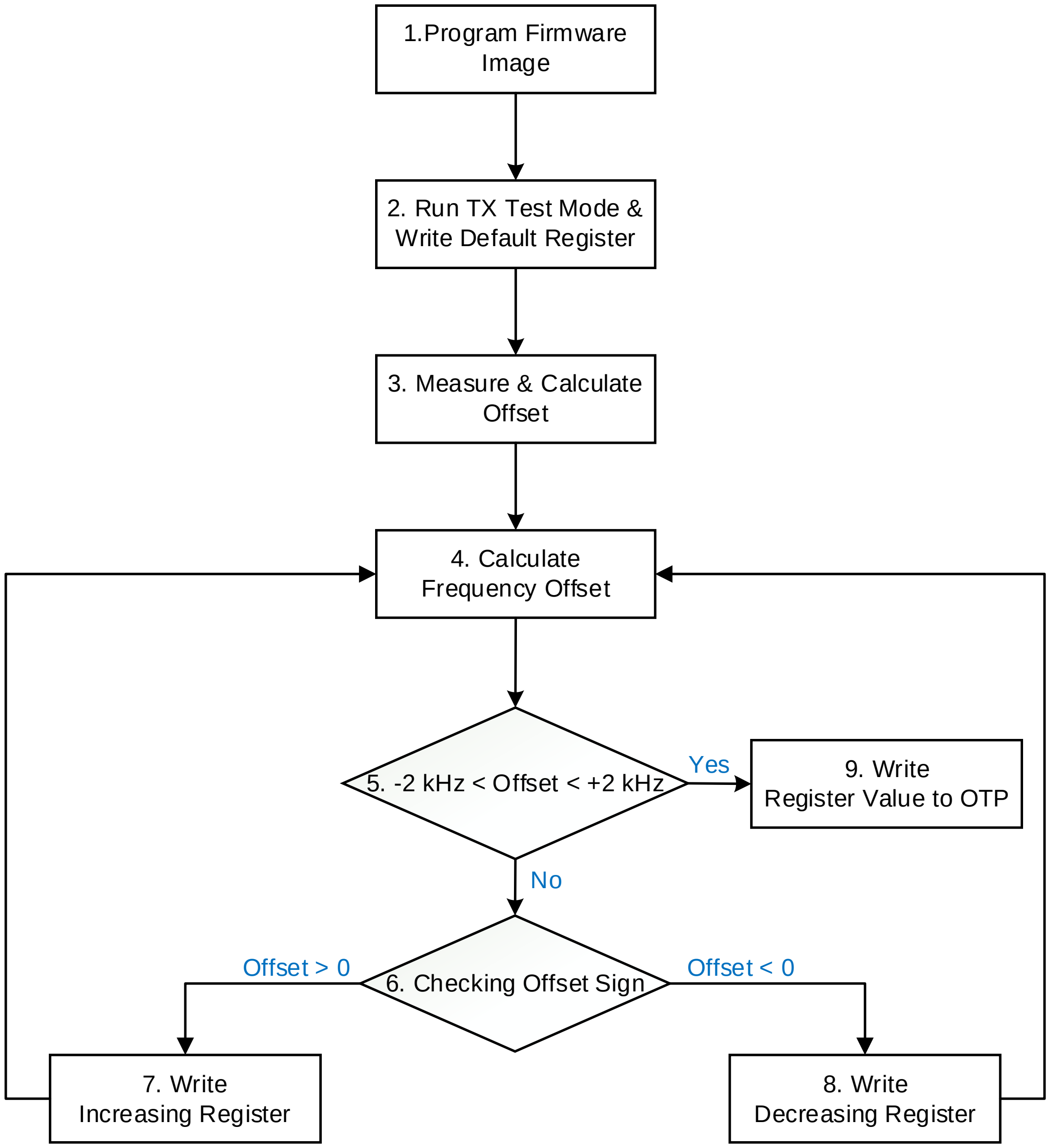

Figure 3. XTAL calibration procedure

7. Wi-Fi RF Test

The TX/RX performance of the DA16200 can be tested in the Certification mode of the AT GUI tool. See Figure 4. For more information about AT commands, see Ref. [3] for **RF Test Function Commands**.

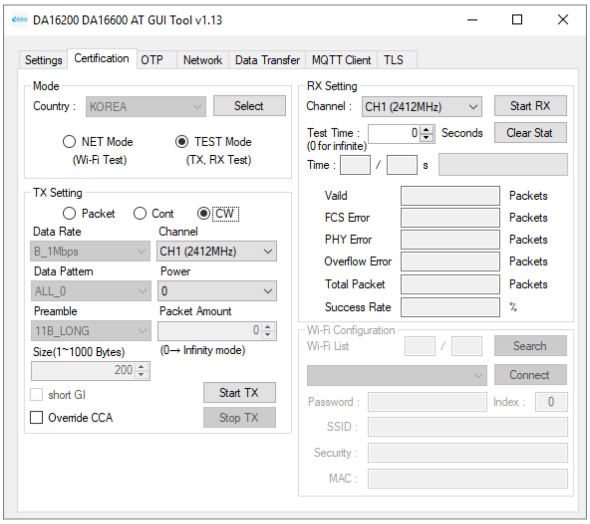

Figure 4. Certification mode in AT GUI tool

7.1 Test Parameter

Basic RF test parameters are listed in Table 4.

7.2 Test Channel

The DA16200 supports up to channel 13, but Renesas highly recommends checking the performance at CH1 (2412 MHz), CH7 (2442 MHz), and CH13 (2472 MHz). To confirm the best performance of the product, Renesas recommends checking the test parameter of the Receiver and Transmitter mentioned in Table 4.

Table 4. RF test parameters

Test parameter	802.11 B	802.11 G	802.11 N (HT20)
TX	EVM	EVM	EVM
	Frequency tolerance	Frequency tolerance	Frequency tolerance
	Output power	Output power	Output power
	Data rate	Data rate	Data rate
	Symbol clock tolerance	Symbol clock tolerance	Symbol clock tolerance
	TX carrier leakage	TX carrier leakage	TX carrier leakage
	Spectrum emission mask	Spectrum emission mask	Spectrum emission mask
RX	Sensitivity	Sensitivity	Sensitivity

7.3 Test Command

Before RX/TX test, it is required to run AT commands listed in Table 5 to initialize RF.

Table 5. RF AT commands

Command	Parameters	Description
AT+TMRFNOINIT=1	<0,1>	Set boot mode. 0: Normal boot 1: RF test mode boot
AT+RESTART	(none)	If set the boot mode as RF test mode (AT+TMRFNOINIT=1), restart the DA16200 (AT+RESTART).
AT+RFTESTSTART	(none)	Start RF test mode.

7.4 TX Test

7.4.1 11B Mode

Table 6. 11B_1 Mbps

Command	Description
AT+RFTX=2412,0,0,200,b1,0	11B 1 Mbps/Channel 1
AT+RFTXSTOP	Stop TX
AT+RFTX=2442,0,0,200,b1,0	11B 1 Mbps/Channel 7
AT+RFTXSTOP	Stop TX
AT+RFTX=2472,0,0,200,b1,0	11B 1 Mbps/Channel 13
AT+RFTXSTOP	Stop TX

7.4.2 11G Mode

Table 7. 11G_54 Mbps

Command	Description
AT+RFTX=2412,0,0,1000,g54,0	11G 54 Mbps/Channel 1
AT+RFTXSTOP	Stop TX
AT+RFTX=2442,0,0,1000,g54,0	11G 54 Mbps/Channel 7
AT+RFTXSTOP	Stop TX
AT+RFTX=2472,0,0,1000,g54,0	11G 54 Mbps/Channel 13
AT+RFTXSTOP	Stop TX

7.4.3 11N Mode

Table 8. 11N_MCS7

Command	Description
AT+RFTX=2412,0,0,1000,n65,0	11N MCS7/Channel 1
AT+RFTXSTOP	Stop TX
AT+RFTX=2442,0,0,1000,n65,0	11N MCS7/Channel 7
AT+RFTXSTOP	Stop TX
AT+RFTX=2472,0,0,1000,n65,0	11N MCS7/Channel 13
AT+RFTXSTOP	Stop TX

7.5 RX Test

For accurate RX measurement, measure lossless (Shield room or Anechoic chamber) conditions without external signal influence.

- Channel: Support CH1 ~ CH13
- Test Time: Maximum 3600s (Duration is 1 second fixed)
- RX Packet Rate: FCS + PHY + Overflow packet/Total packet = Error rate
  The error rate should not exceed 10%.

Any packet error rate (PER) greater than 10% at any RX sensitivity level should be considered a failure.

7.6 RF Test Specification

At the module production stage, it can be determined that there is no performance problem with Wi-Fi RF based on the RF specifications in Table 9.

Table 9. Specifications

Items	Channel	Frequency	Standard	Factor	Low limit	Measure target	High limit	Unit
Freq. calibration	1	2412	11N MCS7	Tx power	-3 >	0	< +3	ppm

Items	Channel	Frequency	Standard	Factor	Low limit	Measure target	High limit	Unit
TX	1,7,13	2412 2442 2472	11B 1 Mbps	EVM	-80 >	-24.4	<-12	dB
				EVM	0.01 >	6	< 25	%
				Target power	16 >	19	< 20	dBm
				SEM	1 >	3	< 20	dB
			11G 54 Mbps	EVM	-80 >	-24.4	< -26	dB
				EVM	0.01 >	-30	< 5	%
				Target power	12.5 >	14.5	< 16.5	dBm
				SEM	1 >	3	< 20	dB
			11N MCS7	EVM	-80 >	-30	< -28	dB
				EVM	0.01 >	6	< 4	%
				Target power	12.5 >	14.5	< 16.5	dBm
				SEM	1 >	3	< 20	dB
RX	1,7,13	2412	11B 1 Mbps	PER	0 >	@ -90 dBm	< 10	%
		2442	11G 54 Mbps	PER	0 >	@ -67 dBm	< 10	%
		2472	11N MCS7	PER	0 >	@ -64 dBm	< 10	%

8. Write MAC Address

The MAC addresses written in the OTP memory are used for the WLAN0 interface (Station) MAC address and the next number is automatically designated as the WLAN1 (Soft AP) MAC address. For example, if `AA:BB:11:22:33:44` is written in the OTP memory, then WLAN0 has `AA:BB:11:22:33:44` and WLAN1 has `AA:BB:11:22:33:45`.

As each DA16200 chip consumes two MAC addresses, when writing a MAC address to a DA16200 chip, the last byte of the MAC address should be bigger **by 2** than the previous DA16200 chip in the production line.

For example, `AA:BB:11:22:33:44`, `AA:BB:11:22:33:46`, `AA:BB:11:22:33:48`. The last digit of the WLAN0 MAC address should be an even number.

The MAC address is already pre-programmed in the DA16200 chipset production stage. There are two ways to write the MAC address, using AT GUI tool and Console command.

8.1 AT GUI Tool

In AT GUI tool, write MAC addresses in OTP mode. The DA16200 provides four slots to store MAC addresses in the OTP memory. When a new MAC address is written, the previous slot is automatically invalidated. See Ref. [3].

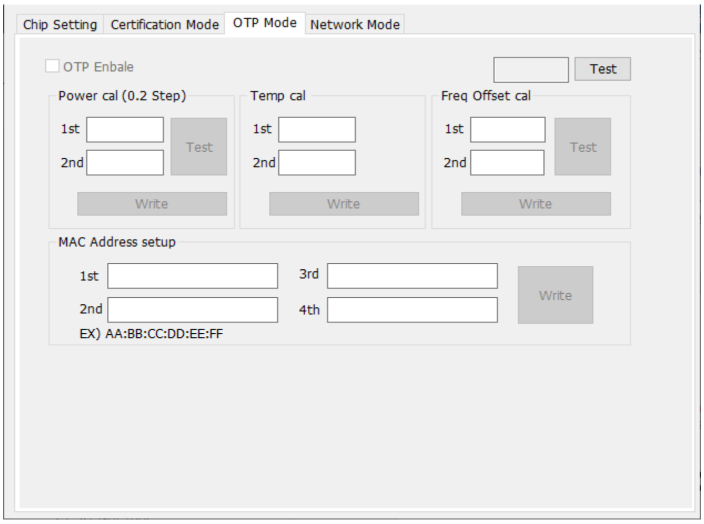

Figure 5. OTP mode in AT GUI tool

NOTE
Power calibration and temperature calibration are completed for the chipset by Renesas Electronics.

Table 10. AT commands for writing/reading MAC address

Command	Parameters	Description
`AT+WFOTP`	<mac>	Write MAC address in the OTP memory. When a new MAC address is written, the previous MAC address is invalidated in the OTP. There are four mac address slots available in the OTP. Therefore, only a maximum of four MAC addresses are written in total in the production stage. Response: `OK` or `ERROR` Example: `AT+WFOTP=EC:9F:0D:90:00:48` The last hex of <mac> should be an even number. The MAC address written in the OTP is used as WLAN0 MAC address and then WLAN's MAC+1 is used as WLAN1 MAC address.
`AT+WFMAC`	(none)	Get the current MAC address of the activated WLAN interface. The DA16200 provides three types of MAC addresses (OTP MAC address, user MAC address, and spoofing MAC address). The priority is OTP < User < Spoofing. Response: `+WFMAC:<mac>`

8.2 Console Command

Use command `setotpmac` to write the new MAC address to an empty slot. This command invalidates the previous slot and validates the new slot.

```
[/DA16200] # setotpmac AA:BB:11:22:33:44
```

Use command `getwlanmac` to check what the new MAC address is.

```
[/DA16200] # getwlanmac

MAC TYPE: OTP MAC

WLAN0 - AA:BB:11:22:33:44

WLAN1 - AA:BB:11:22:33:45
```

9. Wi-Fi Function Test

To test the basic Wi-Fi functions (station and Soft AP), use the Network tab in the AT GUI tool. See Ref. [3] for more information on AT GUI tool, and see Ref. [7] for related commands under the **Network Function Commands** section.

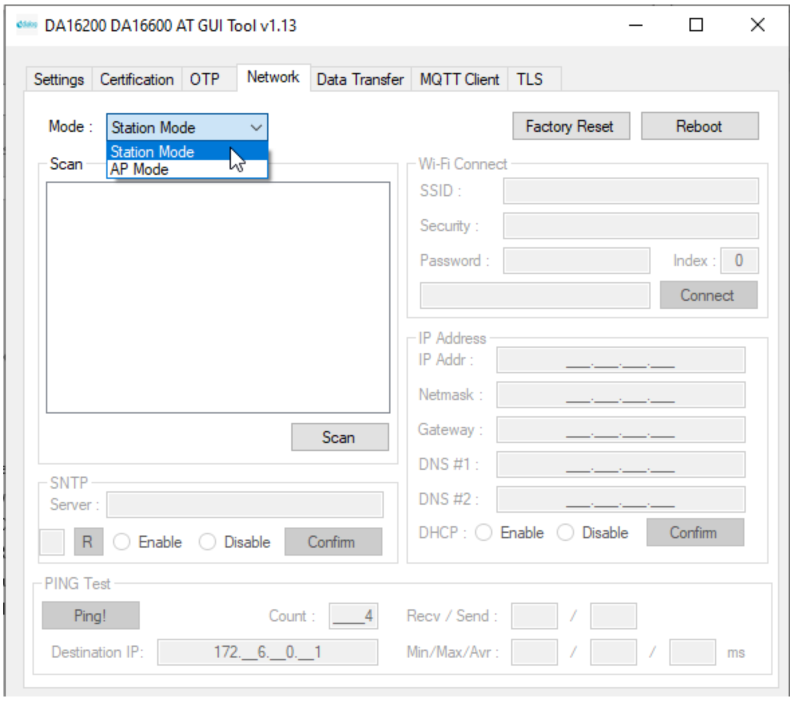

Figure 6. Network in AT GUI tool

10. Sleep Current Measurement

To detect any current leakage in the DA16200, measure the sleep current consumption. The following example code makes the DA16200 enter Sleep mode.

```
[/DA16200] # sys.hal
[/DA16200/SYS] # sleep [mode] [time]
Mode: sleep mode
2: Sleep mode 2.
3: Sleep mode 3.
Time: DA16200 wakes up after this time passes (second)
```

See Ref. [2] for measuring current consumption under the **Current Measurement** section.

11. Factory Reset

The DA16200 must be initialized because multiple profiles may be written in the NVRAM during the production process. To erase all user NVRAM items, use the command code example in Section 11.1 or AT commands in Table 11.

11.1 Console Command

```
[/DA16200] # factory
FACTORY RESET [N/y/?]y


Start Factory-Reset ...


Rebooting ....
```

11.2 AT Command

Table 11. AT command for factory reset

Command	Parameters	Description
ATF	(none)	DA16200 factory reset

12. Change Boot Index

The boot index can be changed by using console command `boot_idx`. Also, the SDK version and boot index can be confirmed during boot as shown in following sections.

12.1 DA16200 Console Command

```
[/DA16200] # boot_idx 1                                                  Moving the boot index from 0 to 1
[/DA16200] # reboot
******************************************************
        *             DA16200 SDK Information
        * ---------------------------------------------------
        *
        * - CPU Type        : Cortex-M4 (120 MHz)
        * - OS Type         : FreeRTOS 10.4.3
        * - Serial Flash    : 4 MB
        * - SDK Version     : V3.2.8.0 GEN
        * - F/W Version     : FRTOS-GEN01-01-f017bfdf51-006558
        * - F/W Build Time  : Aug 10 2023 13:44:35
        * - Boot Index      : 1                                          Must check the boot index
        *
        ******************************************************
```

12.2 DA16600 Console Command

For the DA16600, Bluetooth® LE firmware image is programmed from the DA16200 to DA14531 during boot. Check whether the programming was successful through the following logs from the DA16200.

```
        ******************************************************
        *             DA16600 SDK Information
        * ---------------------------------------------------
        *
        * - CPU Type        : Cortex-M4 (120 MHz)
        * - OS Type         : FreeRTOS 10.4.3
        * - Serial Flash    : 4 MB
        * - SDK Version     : V3.2.8.0 GEN-ATCMD
        * - F/W Version     : FRTOS-GEN01-01-f017bfdf51-006558
        * - F/W Build Time  : Aug 10 2023 14:09:33
        * - Boot Index      : 1                              Must check the boot index
        *
        ******************************************************

gpio wakeup enable 00000402
[combo][iot_sensor]
   is_provisioned = 0
   is_sensor_started = 0
[combo] dpm_boot_type = 0

>>> UART1 : Clock=80000000, BaudRate=115200
>>> UART1 : DMA Enabled ...
[combo] BLE_BOOT_MODE_0
[combo] BLE FW VER to transfer ....
   >>> v_6.0.14.1114.3 (id=1) at bank_1
RTC switched to XTAL
combo] BLE FW transfer done              Make sure to check this message "BLE FW Transfer done"

System Mode : Station Only (0)
>>> Start DA16X Supplicant ...
>>> DA16x Supp Ver2.7 - 2022_03
>>> MAC address (sta0) : d4:3d:39:11:5e:c6
>>> sta0 interface add OK
>>> Start STA mode...
by default, rf_meas_btcoex(1, 0, 0)

>>> UART2 : Clock=80000000, BaudRate=115200
>>> UART2 : DMA Enabled ...
<<< GAPM_DEVICE_READY_IND
IoT dev_name="DA16600-5EC6", len=12
[combo] Advertising...                   Make sure to check this message "Advertising..."
```

13. Bluetooth® LE Test

The DA16600 is a module which internally includes DA16200 (Wi-Fi) and DA14531 (Bluetooth). To test the XTAL and RF of the DA14531, make sure to check the four configurations in advance. Using the SmartSnippets Tool and the Production Line Tool Reference CLI are described in this section.

13.1 Getting Started

This section describes how to perform production tests for Bluetooth® LE device on Evaluation Board (EVB) for helping users to understand the test procedure in the final product.

13.1.1 Erase Flash Content

Before Bluetooth® LE test, if the flash is not empty, run the following command to erase the flash content of the DA16600 module and go to `[MROM]` prompt by typing `reset` on the prompt.

The flash should be empty and the DA16200 does not download Bluetooth® LE image binary to DA14531 or it does not boot as no images in the flash.

```
[/DA16600] # reset

Reset BLE ...

        ***********************************************

        *      FCI    FC9K MaskROM BootLoader

        *  Cortex-M4 (XTAL 40000 kHz, SYS 120000 kHz)

        *  Console Baud Rate : 0 (00000000)

        *  HW Version Num.    : fc905010

        *  Build Option       : RomALL

        *  RoSDK Date & Time : Mar 13 2019 13:05:45

        *  Build Date & Time : Mar 13 2019 13:11:24

        *            http://www.fci.co.kr

        ***********************************************

[MROM] sflash erase 0 400000                                    Command to Erase flash content
```

13.1.2 1-Wire UART

In general, 1-wire UART is used to test DA14531. Figure 7 is the DIP SWITCH applied to DA16600 EVB. Renesas recommends using P0_5 for 1-wire UART.

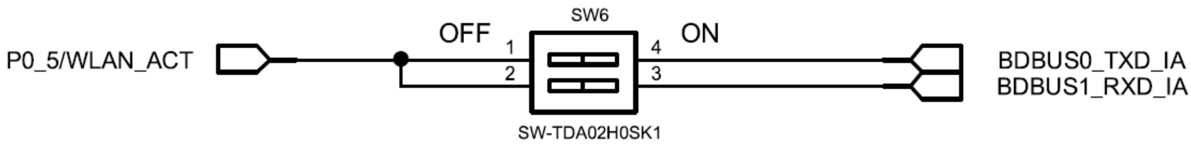

Figure 7. 1-wire UART setup on DA16600 EVB

13.1.3 Hardware Reset

Hardware reset is required to load DA14531 binary. P0_0 pin of DA14531 and GPIOA1 pin of DA16200 are internally connected as a generic transport layer (GTL) line. For the reset, this pin must be pinned out.

Figure 8 shows a circuit configured for DA14531 reset in DA16600 EVB. Make sure to release the Switch (S3) quickly after pressing the button. Hardware reset must be performed by following the SmartSnippets Tool or the command to press reset in the Production Line Tool command window.

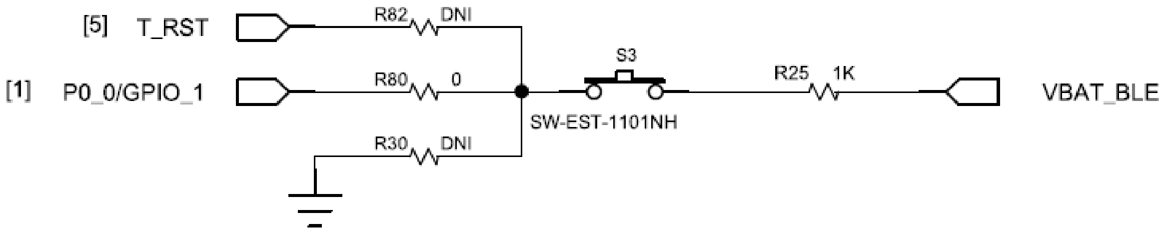

Figure 8. DA14531 hardware reset on DA16600 EVB

13.1.4 RF Path Selection in SPDT

The DA16600 has an RF switch, Single Pole Double Throw (SPDT), for coexistence of the DA14531 and DA16200. This RF switch can be switched by the BT_ACT(P0_6) signal of the DA14531. To test DA14531 Bluetooth, it must be maintained in High state as shown in Table 12 and Figure 9 because it is basically set to the path of the DA16200.

Table 12. RF path for DA16600

P0_6 (BT ACT)	RF path
Low	DA16200 Wi-Fi
High	DA14531 Bluetooth

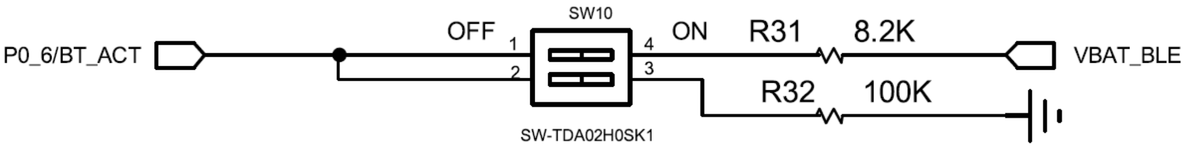

Figure 9. RF switch setup on DA16600 EVB

13.2 XTAL Calibration

This section describes how to use SmartSnippets tool and Production Line tool.

13.2.1 SmartSnippets Tool

SmartSnippets™ Toolbox is provided with the Development Kit of Renesas's Bluetooth® LE chipset. The toolbox is a tool that helps Bluetooth® LE smart application developers test without expensive and bulky equipment. Renesase recommends using this tool for Bluetooth® LE RF test. See Ref. [9] for installation and operation in detail. To get the tool and user manuals, download it from Renesas website (https://www.renesas.com/us/en/products/wireless-connectivity/wi-fi/low-power-wi-fi).

Using the unmodulated TX signal to check how far the XTAL is from the center frequency. Make sure to check the appropriate TRIM value by adjusting this value in the RF Master. The values shown are decimal values.

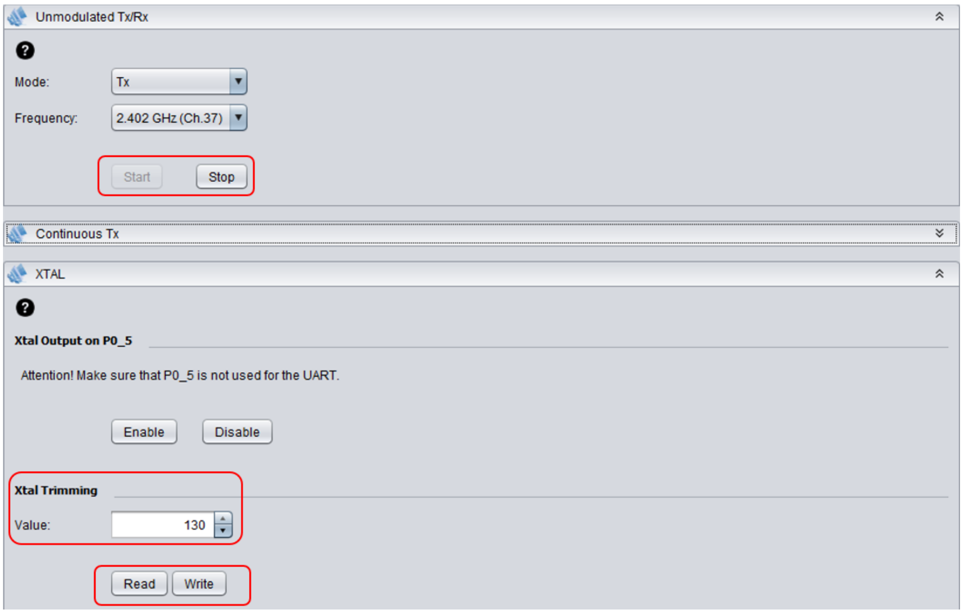

Figure 10. XTAL trimming with TX

13.2.1.1 OTP Write

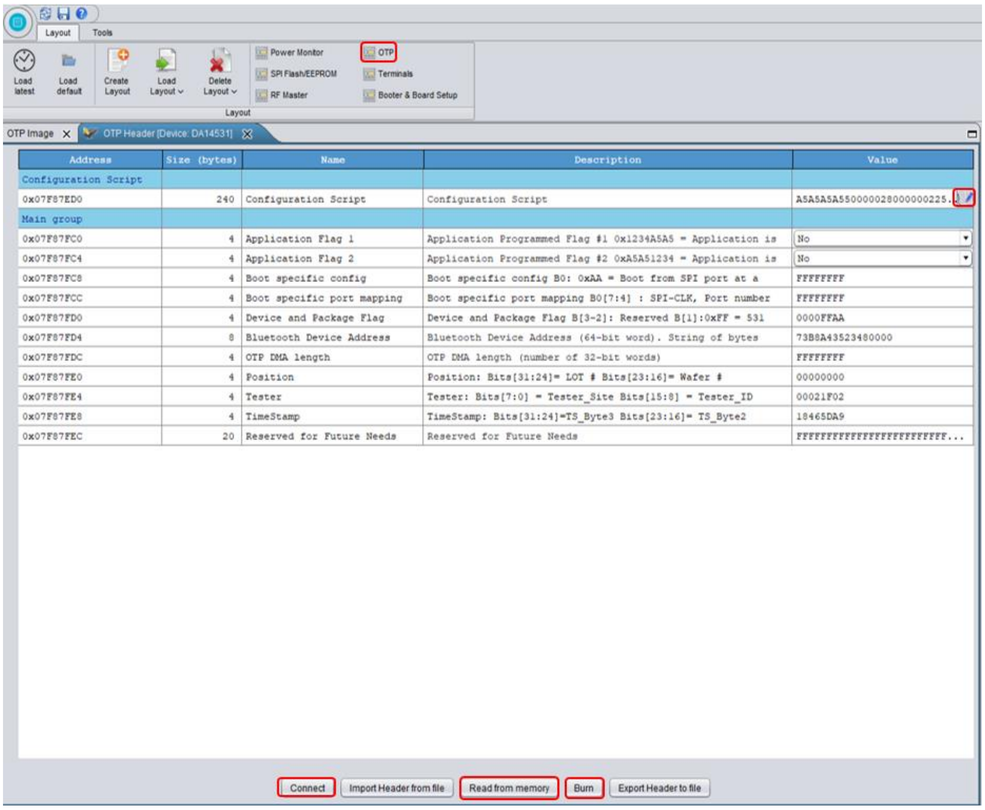

Figure 11. Writing XTAL trimming to OTP

- Connect: Connection to OTP header
- Read from Memory: Reading XTAL trimming value after OTP writing
- Burn: Write XTAL trimming value to OTP

To write the TRIM value to OTP, find the register and enter the desired value. The displayed value is expressed as a Hex value.

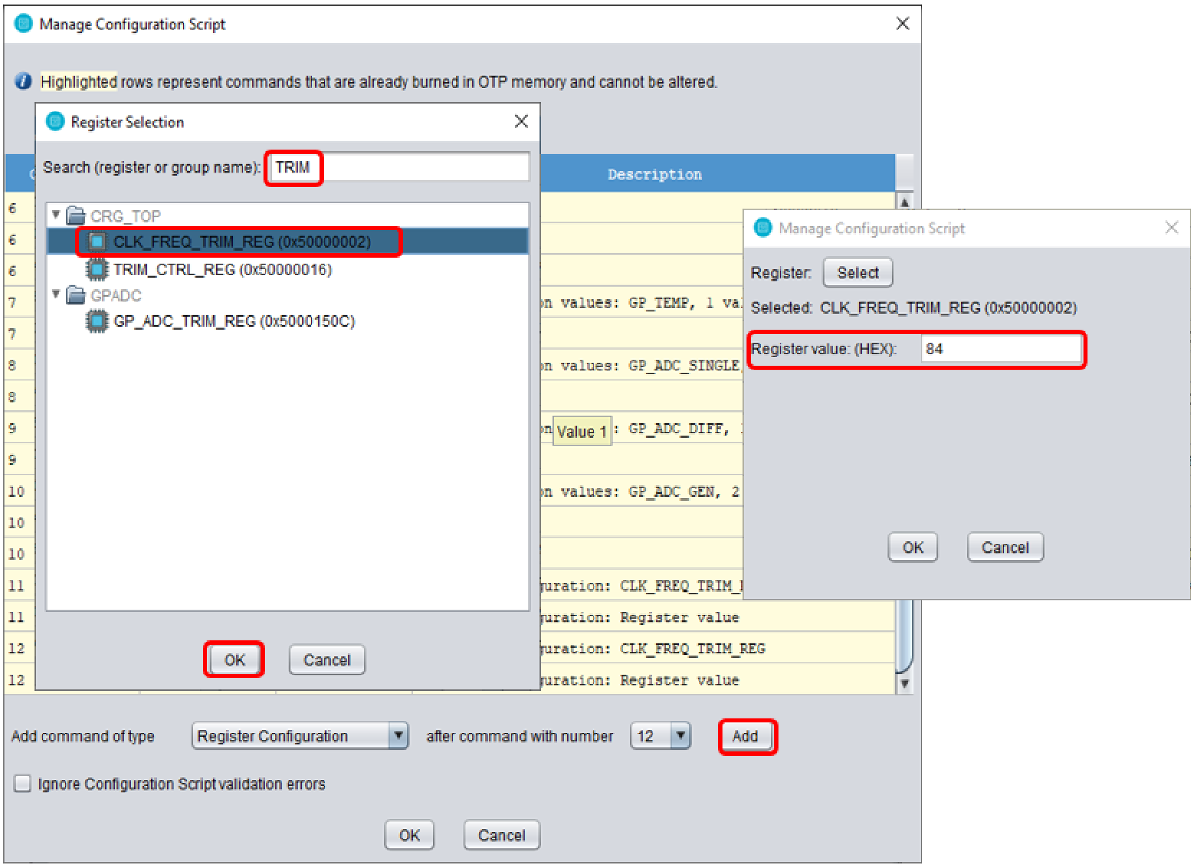

Figure 12. OTP register for XTAL trimming

Write the value to the OTP with the **Burn** button and check the written XTAL trim value is written to the OTP with the **Read from memory** button.

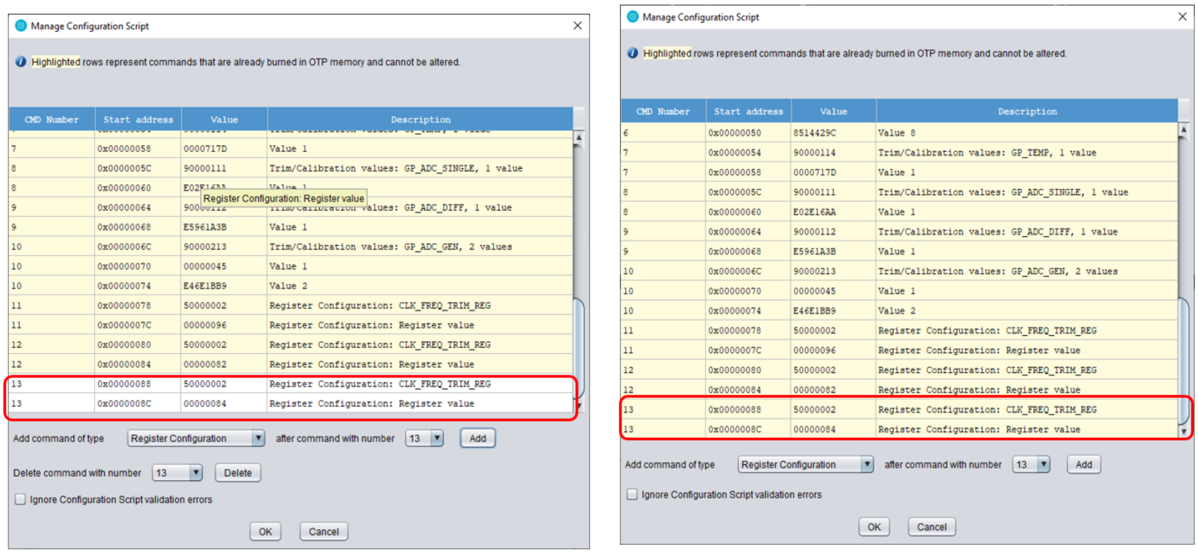

Figure 13. OTP register check for XTAL trimming

13.2.2 Production Line Tool

XTAL trimming can be performed through Prod_test. This method also requires running SmartSnippets tool. Basically, to test it with production line tool, you must know the locations of the SmartSnippets.exe, DTM test binary, and Prod_test.exe in advance. See Ref. [10] for operation with Production Line Tool.

- Path of SmartSnippets tool
- Path of DTM test binary

- Precompiled binaries are provided for both the production test firmware and tool in the following SDK paths.
    - sdk_root_directory\binaries\da14531\prod_test\prod_test_531_UART_P05.bin
- Path of Prod_test.exe
    - Renesas provides the HCI test command and program files in the SDK. It is usually located in the following path.
        - sdk_root_directory\binaries\host\windows\prod_test_cmds\prodtest.exe

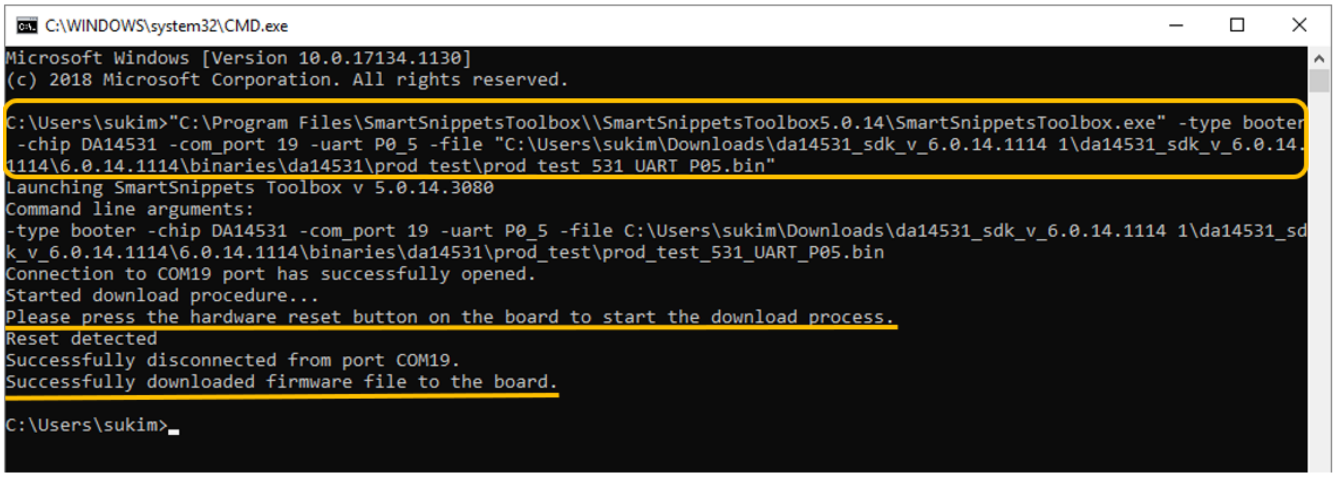

Figure 14. Launch SmartSnippets with command.exe

Example:

This is an example of a command to run SmartSnippets 5.0.14 and download Prod_test_531_UART_P0_5.bin using 1-wire COM21. For 1-wire UART, see Figure 7.

"C:\Program Files\SmartSnippetsToolbox**SmartSnippetsToolbox5.0.14**\SmartSnippetsToolbox.exe" -type booter -chip DA14531 **-com_port 21** -uart P0_5 -file "C:\Us\da14531_sdk_v_6.0.14.1114 1\da14531_sdk_v_6.0.14.1114\6.0.14.1114\binaries\da14531\prod_test**prod_test_531_UART_P05.bin**"

XTAL trimming and RF tests are available by running SmartSnippets and downloading the DTM binary to the device. While downloading the DTM binary, you must press the hardware reset button (see Figure 8). After downloading the DTM binary, open the path of Prod_test.exe in the command window and use RF CLI as shown in Figure 15.

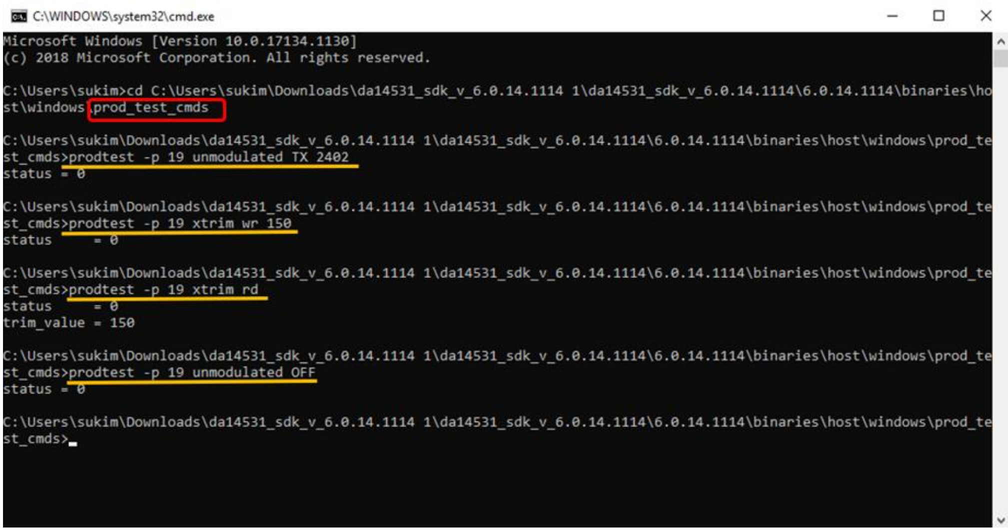

Figure 15. Write and test XTAL trim value with prodtest.exe

To find the XTAL trim value, measuring equipment such as a spectrum analyzer is required and find the offset value using the unmodulated signal. Complete the following steps to check and write the value.

1. Start unmodulated TX.

2. Write XTAL trim value to the register.
3. Read XTAL trim value from the register.
4. Stop unmodulated TX.

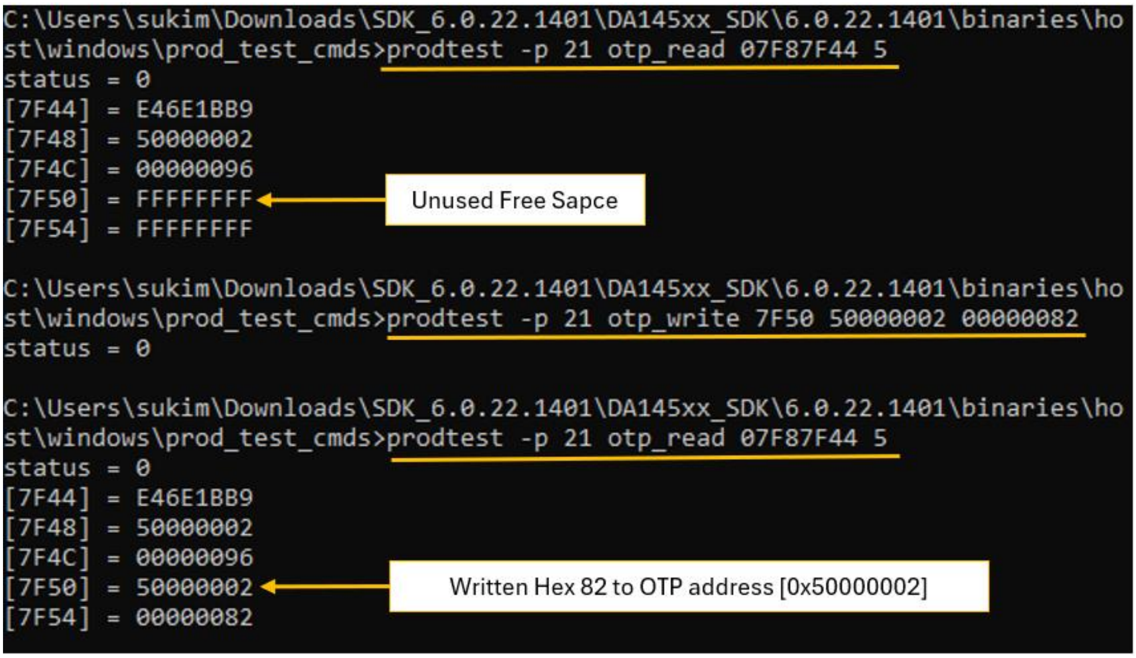

Figure 16. Write XTAL trim value to OTP address

5. Find empty OTP register through the OTP read command (Generally available free addresses start from 7F4C).
   a. Find the end of the configuration script by using otp_read and looking for FFFFFFFF.
6. Write the trim value in the empty space of the OTP.
   a. In this case, the first free address is 7F50. Write the address of the CLK_FREQ_TRIM_REG (50000002) followed by the hexadecimal trim value (for example, decimal 130).

13.3 RF Test

13.3.1 SmartSnippets Tool

1. For continuous transmission, select **Continuous TX** (duty cycle close to 100%). Set the correct frequency and click **Start**. The DUT appears responding successfully to the command in the **Log** tab (response in green text). Click **Stop** to end the command.

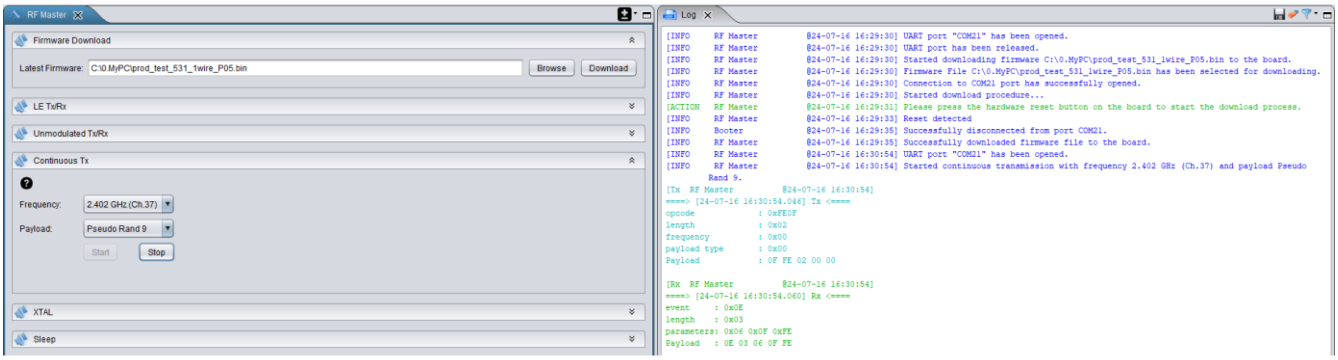

Figure 17. Continuous TX

2. For continuous reception, go to **LE TXRX** and select **Mode Receiver**. Set the correct frequency and click **Start**. The DUT appears responding successfully to the command in the **Log** tab (response in green text). Click **Stop** to end the command.

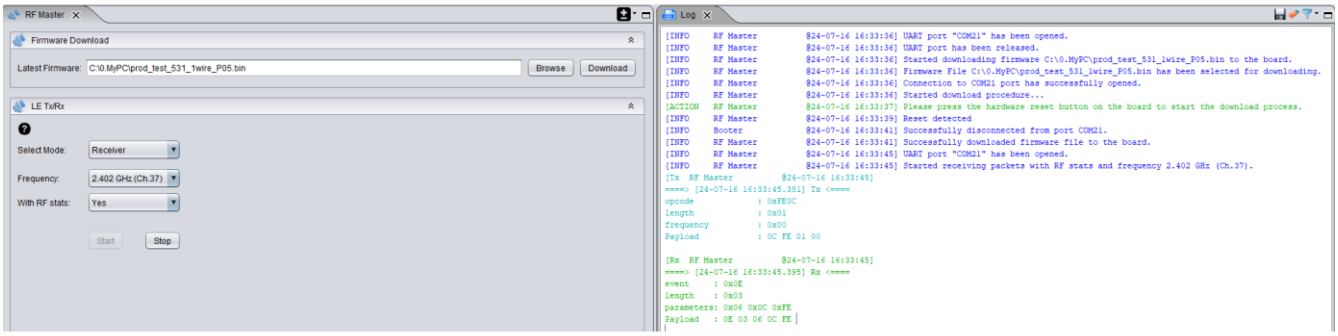

Figure 18. Continuous reception using LE RC

13.3.2 Production Line Tool

By using the production line tool, it is possible to test, calibrate, and load firmware for 16 different devices under test (DUTs) in parallel. There is a Production tool that can perform a production test setup for the DA14531. This tool can be used to test all functions of Bluetooth® LE including XTAL trimming and to write the board address into OTP.

For more information about using PLT, see Ref. [8]. RF testing can be done with the commands specified in Table 13 and see Ref. [10].

Table 13. RF prod test command

Command	Description
prodtest -p 21 cont_pkt_tx 2442 37 0	Pseudo-Random bit sequence 9
prodtest -p 21 stoptest	Stop TX
prodtest -p 21 cont_pkt_tx 2442 37 1	Pattern of alternating bits 11110000
prodtest -p 21 stoptest	Stop TX
prodtest -p 21 cont_pkt_tx 2442 37 2	Pattern of alternating bits 10101010
prodtest -p 21 stoptest	Stop TX
prodtest -p 21 start_pkt_rx_stats 2442	
prodtest -p 21 stop_pkt_rx_stats	

13.4 RF Test Specification

In the module production stage, it can be determined that there is no performance problem with Bluetooth RF based on the RF specifications in Table 14.

Table 14. Bluetooth test specifications

Items	Channel	Frequency	Standard	Factor	Low limit	Measure target	High limit
TX	46	2442	1 Mbps @ 37 bytes	RMS Power	-4 >	0	< 12
				DELTA F1 AVG	225 >	250	< 275
				DELTA F2 AVG	185 >	250	< 275
				DELTA F2 MAX	185 >	230	< 275
				DELTA F2/F1 AVG	0.8 >	1	< 1.1
				Init Freq Error	-150 >	7	< 150
				Freq Drift	0 >	0	< 50
				|F0-FN| Max	0 >	4	< 50
				|F1-F0|	0 >	3	< 20
				|FN-F5|	0 >	2	< 20
				ACP_MAX_POWER_CH-5	-99 >	-57	< -30
				ACP_MAX_POWER_CH-4	-99 >	-52	< -30
				ACP_MAX_POWER_CH-3	-99 >	-54	< -30
				ACP_MAX_POWER_CH-2	-99 >	-45	< -20
				ACP_MAX_POWER_CH-1	-99 >	-20	< 99
				ACP_MAX_POWER_CH+0	-99 >	1	< 99
				ACP_MAX_POWER_CH+1	-99 >	-20	< 99
				ACP_MAX_POWER_CH+2	-99 >	-45	< -20
				ACP_MAX_POWER_CH+3	-99 >	-54	< -30
				ACP_MAX_POWER_CH+4	-99 >	-54	< -30
				ACP_MAX_POWER_CH+5	-99 >	-57	< -30
				PER	0 >	@ -70 dBm	< 10
				ACP_MAX_POWER_CH+1	-99 >	-20	< 99
				ACP_MAX_POWER_CH+2	-99 >	-45	< -20
				ACP_MAX_POWER_CH+3	-99 >	-54	< -30
				ACP_MAX_POWER_CH+4	-99 >	-54	< -30
				ACP_MAX_POWER_CH+5	-99 >	-57	< -30
RX	46	2442	1 Ms/s	PER	0 >	@ -70 dBm	< 20%

Appendix A Wi-Fi RF Test with Console

This section describes how to check the Wi-Fi TX and RX RF performance using a UART debug port. There are two UARTs available on the DA16200 which can be used to check RF performance.

UART0 is used for debugging and checking Wi-Fi functions by programming firmware images and using CLI commands.

A.1 RF Test Setup

Console commands are normally undefined because of memory limitations and must be enabled to use console commands for performing RF tests through the debug port. See the SDK application note for enabling the console command interface for debugging.

To test using the DA16200 EVB, firmware image for AT commands must be programmed. Prebuilt firmware images for AT commands can be downloaded from the Renesas website (https://www.renesas.com/us/en/products/wireless-connectivity/wi-fi/low-power-wi-fi).

```
******************************************************
  *             DA16200 SDK Information
  * ---------------------------------------------------
  *
  * - CPU Type         : Cortex-M4 (120 MHz)
  * - OS Type          : FreeRTOS 10.4.3
  * - Serial Flash     : 4 MB
  * - SDK Version      : V3.2.8.0 GEN-ATCMD                              Must check the image version
  * - F/W Version      : FRTOS-GEN01-01-f017bfdf51-006558
  * - F/W Build Time   : Aug 10 2023 13:44:35
  * - Boot Index       : 0
  *
  ******************************************************
```

For more details on using CLI commands, see Ref. [2]. List the available NVRAM commands by using the `?` or `Help` command. Enter test mode using the `setenv NOINITWLAN 1` command and verify that the message "`!!! TEST MODE !!!`" is displayed.

NOTE
setenv NOINITWLAN 0 can be used to change to normal mode.

```
[/DA16200] # nvram                                              Enter the NVRAM
    Command-List is changed, "NVRAM"
[/DA16200/NVRAM] # setenv NOINITWLAN 1                          setenv NOINITWLAN 1 : Test mode
[/DA16200/NVRAM] # reboot                               Needed reboot command

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



        ******************************************************
        *             DA16200 SDK Information
        * ---------------------------------------------------
        *
        * - CPU Type        : Cortex-M4 (120 MHz)
        * - OS Type         : FreeRTOS 10.4.3
        * - Serial Flash    : 4 MB
        * - SDK Version     : V3.2.8.0 GEN-ATCMD
        * - F/W Version     : FRTOS-GEN01-01-f017bfdf51-006558
        * - F/W Build Time  : Aug 10 2023 13:44:35
        * - Boot Index      : 0
        *
        ******************************************************

Fail to initialize WLAN. (step 1)
!!! TEST MODE !!!                                               setenv NOINITWLAN 1 : Test mode
>>> UART1 : Clock=80000000, BaudRate=115200
>>> UART1 : DMA Enabled ...
```

Before performing an RX test or TX test, the device must be initialized using the following commands:

```
[/DA16200] # lmac.tx.init                               Initializing to enable LMAC command
MAC init. OK
[/DA16200/LMAC.LMAC_TX] # start                                 start command
START
[/DA16200/LMAC.LMAC_TX] #
```

NOTE
When all tests are complete, the DA16200 must be returned to Wi-Fi operational mode by using the "Factory" command.

A.2 Commands for TX Test

A typical TX test environment is shown in Figure 19. TX power of the EVB can be measured using a signal analyzer. Accurate measurements can be achieved only by using Conducted RF testing. Measuring RF with an antenna might show a significant difference in performance depending on the surrounding environment and the distance from the device, therefore Renesas recommends to perform the measurements using an RF cable connected to the U.FL connector on the module.

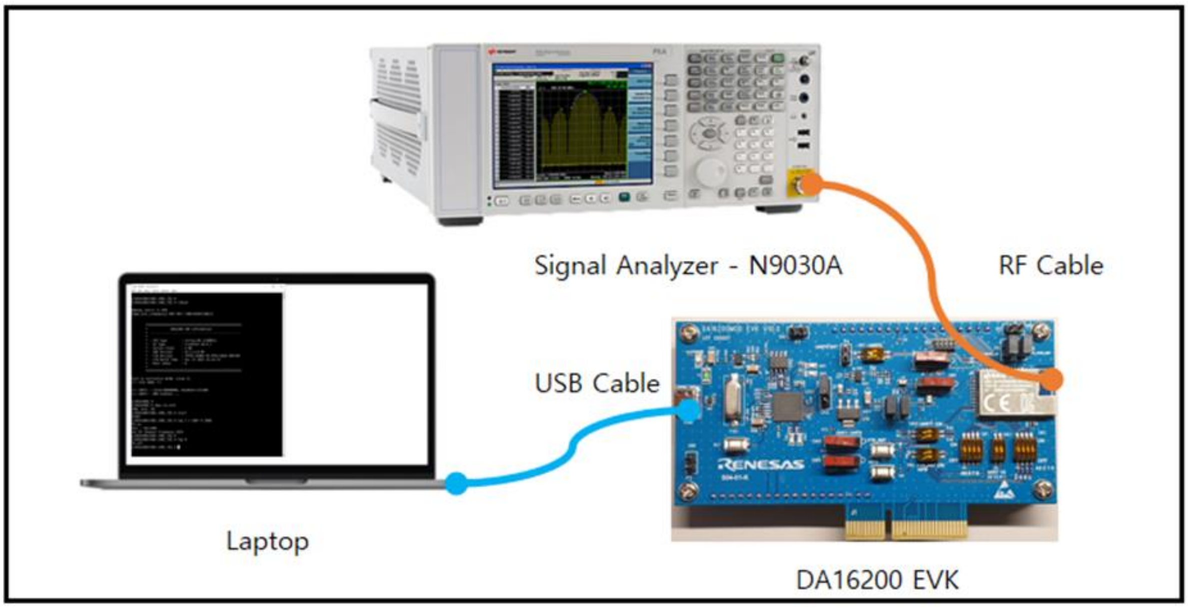

Figure 19. TX test setup

```
[/DA16200] # lmac.tx.init
LMAC init. OK
[/DA16200/LMAC.LMAC_TX] # start
START
[/DA16200/LMAC.LMAC_TX] #
[/DA16200/LMAC.LMAC_TX] # txp 1 1 1007 0 1194                    TX ON command
TX on.
desc = 0xc5d80
Set RF Channel Frequency 2412
[/DA16200/LMAC.LMAC_TX] # txp 0                                  TX OFF command
TX off.
[/DA16200/LMAC.LMAC_TX] #
```

The following sequence is for TX signal generation:

1. Type `lmac` and `tx` to enter LMAC > TX command mode.
2. Type `init` and `start`.
3. Type `txp 1 [channel] [modulation type] [power level] [data length]` to generate the TX signal.
   - `Example 1) txp 1 1 400 0 32`
     - CH 1, 11B 1 Mbps, Power level = 0 (max power)
   - `Example 2) txp 1 11 1007 4 1194`
     - CH 11, 11N MCS7, Power grade = 4

Depending on the TX power, it has approximately 0.8 dB steps.

At the end of the test, type `txp 0` which sets the TX signal to the off state. Otherwise, `txp 1` sets the TX signal to the on state. It is required to turn off the signal generation using `txp 0` before performing another test.

Here is an example of TX command for each Wi-Fi standards:

Modulation		Setting parameter	Duty cycle : 95%	For example		
			Data length	Duty cycle : 95%, power grade 3		
			Byte[Hex]	Channel 1	Channel 7	Channel 13
11b	1Mbps	400	32	txp 1 1 400 3 32	txp 1 7 400 3 32	txp 1 13 400 3 32
	2M	401	64	txp 1 1 401 3 64	txp 1 7 401 3 64	txp 1 13 401 3 64
	5.5M	402	12C	txp 1 1 402 3 12C	txp 1 7 402 3 12C	txp 1 13 402 3 12C
	11M	403	258	txp 1 1 403 3 258	txp 1 7 403 3 258	txp 1 13 403 3 258
11g	6M	404	1F4	txp 1 1 404 3 1F4	txp 1 7 404 3 1F4	txp 1 13 404 3 1F4
	9M	405	2BC	txp 1 1 405 3 2BC	txp 1 7 405 3 2BC	txp 1 13 405 3 2BC
	12M	406	3E8	txp 1 1 406 3 3E8	txp 1 7 406 3 3E8	txp 1 13 406 3 3E8
	18M	407	578	txp 1 1 407 3 578	txp 1 7 407 3 578	txp 1 13 407 3 578
	24M	408	708	txp 1 1 408 3 708	txp 1 7 408 3 708	txp 1 13 408 3 708
	36M	409	BB8	txp 1 1 409 3 BB8	txp 1 7 409 3 BB8	txp 1 13 409 3 BB8
	48M	40a	FA0	txp 1 1 40a 3 FA0	txp 1 7 40a 3 FA0	txp 1 13 40a 3 FA0
	54M	40b	FA0	txp 1 1 40b 3 FA0	txp 1 7 40b 3 FA0	txp 1 13 40b 3 FA0
11n	MCS0	1000	1F4	txp 1 1 1000 3 1F4	txp 1 7 1000 3 1F4	txp 1 13 1000 3 1F4
	MCS1	1001	3E8	txp 1 1 1001 3 3E8	txp 1 7 1001 3 3E8	txp 1 13 1001 3 3E8
	MCS2	1002	5DC	txp 1 1 1002 3 5DC	txp 1 7 1002 3 5DC	txp 1 13 1002 3 5DC
	MCS3	1003	7D0	txp 1 1 1003 3 7D0	txp 1 7 1003 3 7D0	txp 1 13 1003 3 7D0
	MCS4	1004	BB8	txp 1 1 1004 3 BB8	txp 1 7 1004 3 BB8	txp 1 13 1004 3 BB8
	MCS5	1005	FA0	txp 1 1 1005 3 FA0	txp 1 7 1005 3 FA0	txp 1 13 1005 3 FA0
	MCS6	1006	1194	txp 1 1 1006 3 1194	txp 1 7 1006 3 1194	txp 1 13 1006 3 1194
	MCS7	1007	1194	txp 1 1 1007 3 1194	txp 1 7 1007 3 1194	txp 1 13 1007 3 1194

Figure 20. Reference for TX test

A.3 Commands for RX Test

For RX and conducted RF test, measurements should be performed without external signal interference. When checking the weak electric field signal such as 11b 1 Mbps mode, accurate measurement results may not be obtained if the measurement is not performed under shield room conditions.

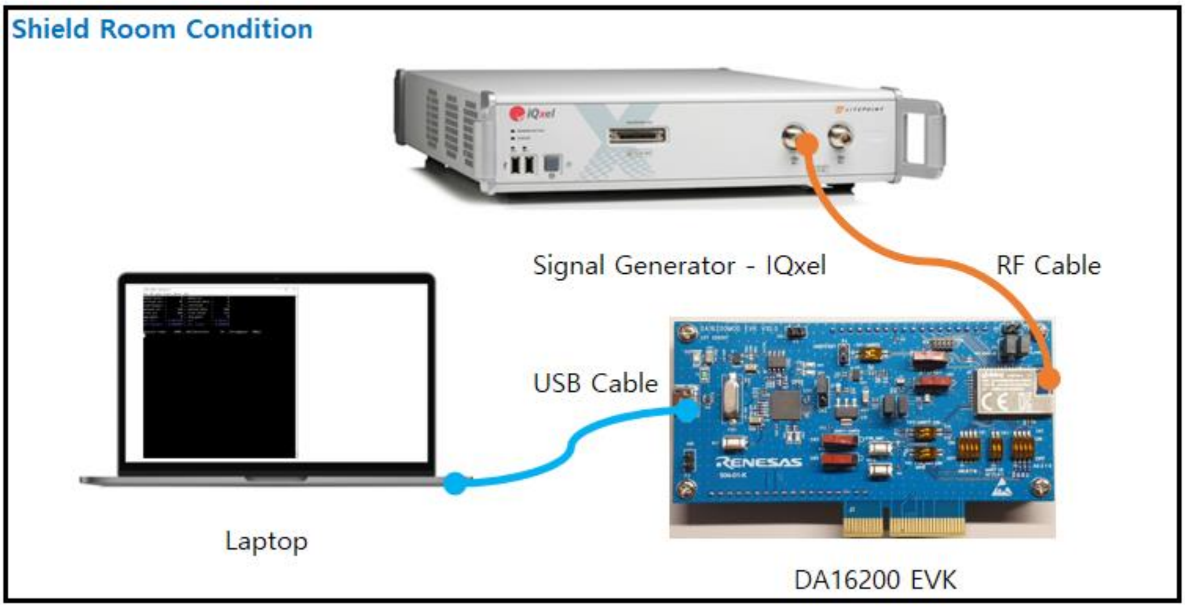

Figure 21. RX test setup

- The sequence of commands to initialize with MAC layer is the same as for the TX test.
- With `lmac.per` command, check the PER of the signals received by the DA16200.
- Type `PER` on the console to stop receiving packets and end the test. This is known as RX test OFF.
  - `Lmac.per`: RX test On
  - `per`: RX test Off (Must be typed during RX test packet reception)

```
[/DA16200] #
[/DA16200] # lmac.tx.init
LMAC already initiated
[/DA16200/LMAC.LMAC_TX] # start
START
[/DA16200/LMAC.LMAC_TX] # lmac.per                    RX Test ON command
```

The PER for the signal received from the signal generator is displayed in the output generated by the `lmac.per` command as marked with the green box in Figure 22. The error rate is displayed based on how many packets were received compared to the transmitted packets.

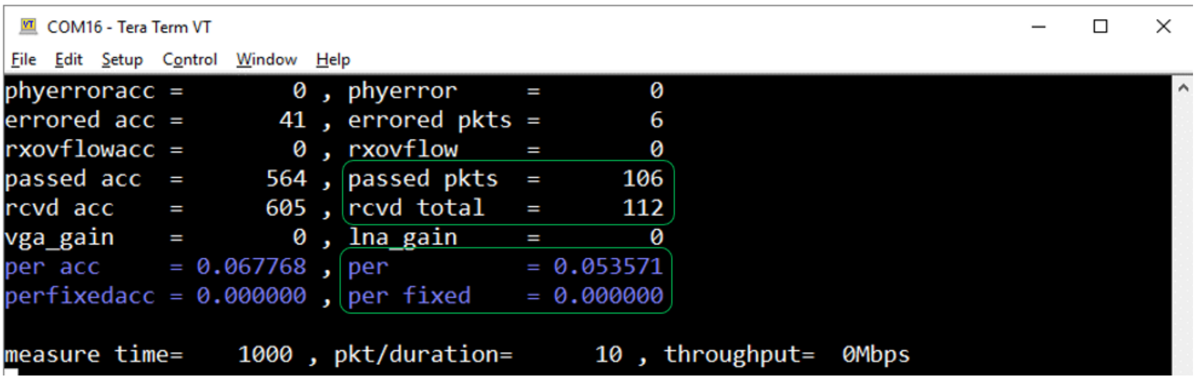

Figure 22. RX PER (Good)

The number of receivable packets per standard can be varied. An error rate of less than 10% compared to sending packets can be considered as a performance indicator. Figure 23 shows an error highlighted in red, but it is not an issue because the error rate is about 0.1%.

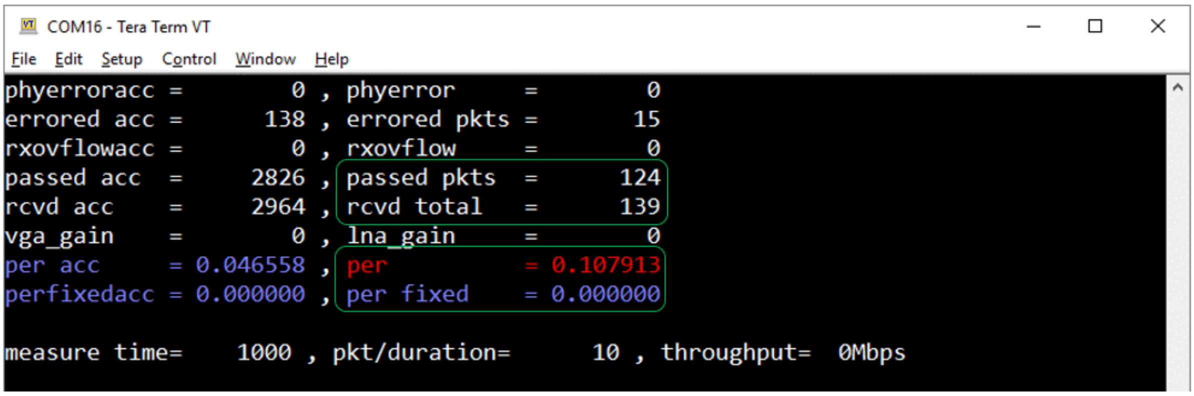

Figure 23. RX PER (Not good)

Appendix B ADC Calibration

This section describes how to calibrate Analog to Digital Converter (ADC) in the target device. The purpose of ADC calibration is to calculate ADC offset and gain and then to compensate ADC error. See Ref. [1]. The Signal to Noise and Distortion Ratio (SNDR) is 61.7 dB and ENOB is 9.96 bits. ADC error tolerance is within 2.04 bits, 4 values in 4096 range and 1.4 mV in 1.4 voltage range.

B.1 ADC Error Compensation

The ADC error is divided into static and dynamic errors. The static error is from DC noise and the dynamic error is from frequency characteristic noise. The static error can be compensated by calibrating the ADC offset and gain in the target device.

```
Compensated ADC value = (Measured ADC value – ADC offset) / ADC gain
```

The dynamic error can be compensated by averaging ADC sample values in software and adding the capacitance (1 uF or 2.2 uF) in ADC GPIO port in hardware design.

B.2 ADC Offset and Gain Calculation

ADC offset and gain are calculated with the following formulas.

```
ADC offset = (Measure_10 x Ideal_90 - Measure_90 x Ideal_10) / (Ideal_90 - Ideal_10)
ADC gain = (Measure_90 - Measure_10) / (Ideal_90 - Ideal_10)
```

NOTE
▪ `Ideal_10` is the ideal ADC value of 0.1 voltage input (0.1/1.4 x 4095).
▪ `Ideal_90` is the ideal ADC value of 1.3 V voltage (1.3/1.4 x 4095).
▪ `Measure_10` is an ADC value read by the DA16200 with 0.1 voltage input on target device.
▪ `Measure_90` is an ADC value read by the DA16200 with 1.3 voltage input on target device.

B.3 ADC Calibration

The procedure for ADC calibration:

1. Prepare target board.
2. Select ADC channel of GPIOA[0:3]. See Ref. [2].
3. Program prebuilt AT command image to the target board.
4. Measure an ADC value for `Measure_10` through AT commands.
   a. Input 0.1v voltage to GPIO for ADC using power supply.
   b. AT+ADCINIT // Configure GPIO to analog input pins for ADC
   c. AT+ADCCHEN=0,12 // Enable channel 0 and 12-bit resolution
   d. AT+ADCSTART=1 // Start ADC function
   e. AT+ADCREAD=0,16 // Read ADC value for channel 0 and 16 samples
   f. AT+ADCSTOP // Stop ADC function
5. Measure an ADC value for `Measure_90` through AT commands.
   a. Input 1.3v voltage to GPIO for ADC using power supply.
   b. Read an ADC value for `Measure_90` like step 4.
6. Write `Measure_10` and `Measure_90` to OTP user area (0x10C – 0x1FE) through AT commands. DA16200 has OTP user area where is from 0x10C to 0x1FE and a value can be written and read to the OTP user area using AT+UOTPWRASC and AT+UOTPRDASC commands. See the Ref. [7].

   Example:

   a. Write the `Measure_10` (0x102) and `Measure_90` (0xEC4)

      AT+UOTPWRASC=10C,4,0EC40102
      OK

b. Read `Measure_10` (0x102) and `Measure_90` (0xEC4) to make sure those values are written properly.

```
AT+UOTPRDASC=10C,4
0ec40102
OK
```

7. Compensate ADC values through ADC compensation sample. Figure 24 shows the procedure of ADC calibration.

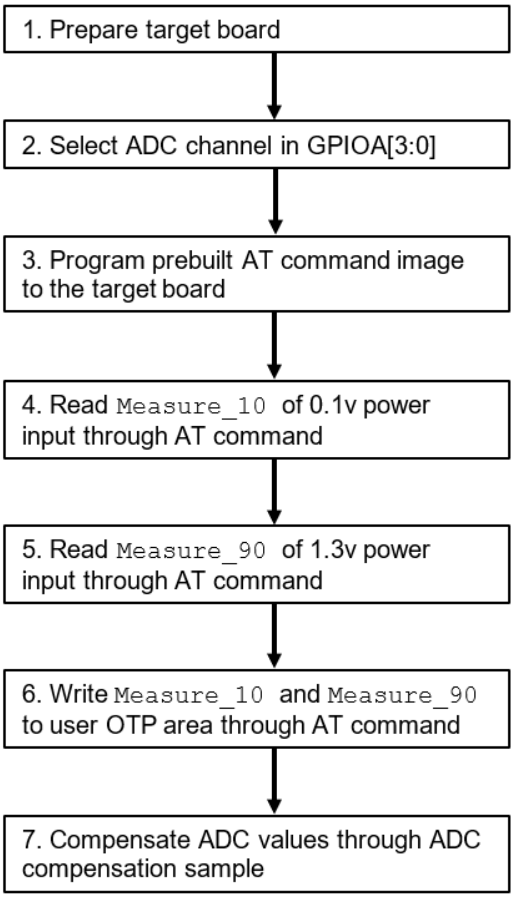

Figure 24. ADC calibration procedure

B.4 ADC Compensation Sample

This section shows a sample ADC compensation sample.

```
void adc_sample(void *param)
{
   int IDEAL_10 = 2925;  // Ideal 0.1V ADC value * 10
   int IDEAL_90 = 38025; // Ideal 1.3V ADC value * 10

   int MEASURE_10, MEASURE_90;
   int offset, gain_err;
   int sum_org_data, avr_org_data;
   int adc_com_data, sum_com_data, avr_com_data;

   unsigned char rdata[4];
   unsigned int addr, cnt, i;
   int status = 0;

   cnt = 4;     // Measure_10(2bytes) and Measure_90(2bytes)
   addr=0x10C; // OTP address that Measured ADC values is written

   // Read Measure_10 and Measure_90 from OTP
   otp_mem_create();
   for (i = 0; i <= (cnt - 1) / 4; i++) {
       status = otp_mem_read((addr/4) + (i), (UINT32 *)(&rdata[i * 4]));

        if (status != 0) {
           PRINTF("OTP ERROR =%X\n", status);
        }
    }
   otp_mem_close();

   MEASURE_10 = (int) (((rdata[2]&0xff)<<8)|(rdata[3]&0xff))*10;      // 0.1V measured ADC * 10
   MEASURE_90 = (int) (((rdata[0]&0xff)<<8)|(rdata[1]&0xff))*10;      // 1.3V measured ADC * 10

   PRINTF("MEASURE_10=%02x, MEASURE_90=%02x\r\n",MEASURE_10/10, MEASURE_90/10);

   // Calculate ADC offset
   offset = ((MEASURE_10 * IDEAL_90 - MEASURE_90 * IDEAL_10)/(IDEAL_90 - IDEAL_10))/10;
   PRINTF("ADC offset is %d\r\n",offset);

   // Calculate ADC gain
   gain_err = ((MEASURE_90 - MEASURE_10)*10000)/(IDEAL_90 - IDEAL_10);
   PRINTF("ADC gain*10000 is %d\r\n",gain_err);

   for (i = 0; i < DA16x_ADC_NUM_READ; i++) {
        // Compensate static error
        adc_com_data = (((int)GET_VALID_ADC_VALUE(data0[i]) - offset)*100000/gain_err + 5)/10;

        sum_org_data = sum_org_data + (int)GET_VALID_ADC_VALUE(data0[i]);
        sum_com_data = sum_com_data + adc_com_data;
   }

   // Compensate dynamic error
   avr_org_data = sum_org_data/DA16x_ADC_NUM_READ;
   avr_com_data = sum_com_data/DA16x_ADC_NUM_READ;

   // Exclude outside the ADC range (0-4095)
   if(avr_com_data < 0) avr_com_data = 0;
   if(avr_com_data > 4095) avr_com_data = 4095;

   PRINTF("\r\nMeasured ADC value %05d, Compensated ADC value %05d\r\n", avr_org_data, avr_com_data);
```

B.5 ADC Compensation Result

Table 15 shows ADC values and errors before and after ADC compensation.

Table 15. ADC compensation result

Voltage (mV)	ADC value			ADC error	
	Ideal	Measured	Compensated	Before compensation	After compensation
100	292.5	258	293	-34.5	0.5
200	585	551	585	-34	0
300	877.5	844	877	-33.5	-0.5
400	1170	1139	1170	-31	0
500	1462.5	1432	1463	-30.5	0.5
600	1755	1723	1753	-32	-2
700	2047.5	2019	2048	-28.5	0.5
800	2340	2314	2341	-26	1
900	2632.5	2607	2633	-25.5	0.5
1000	2925	2896	2922	-29	-3
1100	3217.5	3192	3217	-25.5	-0.5
1200	3510	3489	3512	-21	2
1300	3802.5	3780	3803	-22.5	0.5

NOTE

- ADC error before compensation = Ideal ADC value – Measured ADC value
- ADC error after compensation = Ideal ADC value – Compensated ADC value

Appendix C Program Firmware Images with Segger J-Flash

The following describes how to install the DA16200/DA16600 flash loader for J-Link software and program images into the device using the Segger J-Flash.

C.1 Install Flash Loader to J-Link Software

The DA16200/DA16600 flash loader for J-Link software is in the directory in the SDK.

utility\j-link\scripts\flashloader\Devices\Dialog\

- Copy DA16200_4MB.elf into the location of the central folder of J-Link software, which depends on the host OS. The location of this folder can be found on the following page.
https://wiki.segger.com/J-Link_Device_Support_Kit#JLinkDevices_folder
Example of JLinkDevices folder structure:
- JLinkDevices\Renesas\DA16x00\
- Create Devices.xml and place it in the same folder as the flashloader.

```
<DataBase>
  <!--                       -->
  <!-- Renesas Electronics -->
  <!--                       -->
  <Device>
    <ChipInfo Vendor="Renesas Electronics" Name="DA16200" Core="JLINK_CORE_CORTEX_M4" WorkRAMAddr="0x83000"
WorkRAMSize="0x00020000" />
         <FlashBankInfo Name="QSPI Flash" BaseAddr="0x10000000" AlwaysPresent="1" >
                  <LoaderInfo Name="Default" MaxSize="0x400000" Loader="DA16200_4MB.elf"
LoaderType="FLASH_ALGO_TYPE_OPEN" />
         </FlashBankInfo>
  </Device>
  <Device>
    <ChipInfo Vendor="Renesas Electronics" Name="DA16600" Core="JLINK_CORE_CORTEX_M4" WorkRAMAddr="0x83000"
WorkRAMSize="0x00020000" />
         <FlashBankInfo Name="QSPI Flash" BaseAddr="0x10000000" AlwaysPresent="1" >
                  <LoaderInfo Name="Default" MaxSize="0x400000" Loader="DA16200_4MB.elf"
LoaderType="FLASH_ALGO_TYPE_OPEN" />
         </FlashBankInfo>
  </Device>
</DataBase>
```

C.2 Configuration for Initialization Steps in J-Flash Script

C.2.1 Disable Watchdog

```
Write 32bit - Address: 0x40008C00, Value: 0x1ACCE551
Write 32bit - Address: 0x40008008, Value: 0x00000000
Write 32bit - Address: 0x40008C00, Value: 0x00000000
```

C.2.2 Enable OTP Lock

```
Write 32bit - Address: 0x40120000, Value: 0x34000000
Write 32bit - Address: 0x40103FFC, Value: 0x04000000
Write 32bit - Address: 0x40120000, Value: 0x38000000
Verify 32bit - Address: 0x40103FFC, Value: 0x00000001
```

C.2.3 Set Boot Mode

```
Write 32bit - Address: 0x50000000, Value: 0x00000001
```

C.3 Address of Firmware Images to Program

- The bootloader image should be programmed to the address, **0x10000000**.
- The RTOS image should be programmed to the address, **0x10023000**.
- For the DA16600, the Bluetooth image should be programmed to the address, **0x103ad000**.

Appendix D GPIO Test

D.1 DA16200

To determine whether the device has any physical production problems, you can use AT commands to set GPIO as Input/Output and perform Open/Short tests. Table 16 is a GPIO pair for DA16200 reference. For tests, each paired GPIO must be physically connected.

Table 16. DA16200 Wi-Fi GPIO test

No	Hardware pairs		Description
PAIR 1	GPIOA 0	GPIOA 1	Set GPIOA 0, Read GPIOA 1
PAIR 2	GPIOA 2	GPIOA 3	Set GPIOA 2, Read GPIOA 1 & 3
PAIR 3	GPIOA 6	GPIOA 7	Set GPIOA 6, Read GPIOA 6
PAIR 4	GPIOA 8	GPIOA 9	Set GPIOA 8, Read GPIOA 7 & 9
PAIR 5	GPIOA 10	GPIOA 11	Set GPIOA 10, Read GPIOA 9 & 11
PAIR 6	GPIOC 7	GPIOC 8	Set GPIOC 7, Read GPIOC 6 & 8

```
;;; Pair 1 ------------------------------------------------------
;; Set GPIOA0 output,high, then read GPIOA1, set low then read
;; set GPIOA0 to output
sendln 'AT+GPIOSTART=0,1,1'
waitln 'OK' 'ERROR'
if result=1 then
    ret = 'OK'
else
    messagebox 'check AT+GPIOSTART=0,1,1' '#2,ERROR Detected'
         exit
endif
;; set GPIOA0 high
sendln 'AT+GPIOWR=0,1,1'
waitln 'OK' 'ERROR'
if result=1 then
    ret = 'OK'
else
    messagebox 'check AT+GPIOWR=0,1,1' '#3,ERROR Detected'
         exit
endif
;; Read GPIOA1
sendln 'AT+GPIORD=0,2'
waitln '0x0002' 'ERROR'
if result=1 then
    ret = 'OK'
else
    messagebox 'it should be 0x0002' '#4,ERROR Detected'
         exit
endif
;; set GPIOA0 low
sendln 'AT+GPIOWR=0,1,0'
waitln 'OK' 'ERROR'
if result=1 then
    ret = 'OK'
else
    messagebox 'check AT+GPIOWR=0,1,0' '#5,ERROR Detected'
         exit
endif
;; Read GPIOA1
sendln 'AT+GPIORD=0,2'
waitln '0x0000' 'ERROR'
if result=1 then
    ret = 'OK'
else
    messagebox 'it should be 0x0000 for AT+GPIORD=0,2' '#6,ERROR Detected'
         exit
```

```
endif
```

D.2 DA16600

Table 17 is a GPIO pair for DA16200 (Wi-Fi) reference in the DA16600 module. Table 18 is for GPIO pair for DA14531 (Bluetooth) reference in the DA16600 module. For test, each paired GPIO must be physically connected.

D.2.1 Wi-Fi GPIO

Table 17. DA16600 Wi-Fi GPIO test

No	Hardware pairs of Wi-Fi GPIO		Description
PAIR 1	GPIOA 2	GPIOA 3	Set GPIOA 2, Read GPIOA 1 & 3
PAIR 2	GPIOA 6	GPIOA 7	Set GPIOA 6, Read GPIOA 7
PAIR 3	GPIOA 8	GPIOA 9	Set GPIOA 8, Read GPIOA 7 & 9
PAIR 4	GPIOA 10	GPIOA 11	Set GPIOA 10, Read GPIOA 9 & 11
PAIR 5	GPIOC 7	GPIOC 8	Set GPIOC 7, Read GPIOC 6 & 8

D.2.2 Bluetooth® GPIO

Table 18. DA16600 Bluetooth GPIO test

No	Hardware pairs of Bluetooth GPIO		Description
PAIR 1	P0_2	P0_9	Set P0_9, Read P0_2
PAIR 2	P0_7	P0_10	Set P0_7, Read P0_10
PAIR 3	P0_8	P0_11	Set P0_11, Read P0_8

With the GPIOs in Table 18 physically connected, testing can be performed in the following order.

- Run the SmartSnippets tool to load the Bluetooth binary using Command Prompt. See Figure 14.
- Run Prodtest using prodtest.exe. See Figure 15.
- Enter all the commands one by one. See Table 19.
  - The following example shows how to test GPIO with a script file.

Table 19. DA16600 Bluetooth GPIO test command

Command	Description
prodtest -p 21 write_reg16 50003000 000	Set all zero
prodtest -p 21 read_reg16 50003000	Read P0_x port to check DA14531 pin status
prodtest -p 21 write_reg16 50003018 300	Set P0_9 register output
prodtest -p 21 write_reg16 50003000 200	Set P0_9 high
prodtest -p 21 read_reg16 50003000	Read P0_x and check P0_9 and P0_2 are high
prodtest -p 21 write_reg16 50003018 000	Set P0_9 input.
prodtest -p 21 write_reg16 50003014 300	Set P0_7 register output
prodtest -p 21 write_reg16 50003000 080	Set P0_7 high
prodtest -p 21 read_reg16 50003000	Read P0_x and check P0_7 and P0_10 are high
prodtest -p 21 write_reg16 50003014 000	Set P0_7 input
prodtest -p 21 write_reg16 5000301C 300	Set P0_11 register output
prodtest -p 21 write_reg16 50003000 800	Set P0_11 high
prodtest -p 21 read_reg16 50003000	Read P0_x and check P0_11 and P0_8 are high
prodtest -p 21 write_reg16 5000301C 000	Set P0_11 input

14. Revision History

Revision	Date	Description
2.1	Nov 29, 2024	▪ Updated Section 4 and Section 7.5 ▪ Added Sections 13.3 and 13.4 ▪ Added Appendix D GPIO Test ▪ Added Appendix C Programming with Segger J-Flash ▪ Removed Configure DA16600 EVB 4v0 to Use AT Command with UART2 ▪ Updated Section 13.2 Calibration ▪ Added Appendix B ADC XTAL Calibration
2.0	Sept 27, 2023	▪ Reorganized document according to sequence of production test ▪ Updated XTAL Calibration section
1.9	June 30, 2023	Updated the reference section
1.8	Jan 12, 2023	▪ Updated descriptions in Writing MAC Address section ▪ Removed supporting channel 14
1.7	Aug 16, 2022	▪ DA16600MOD Mass Production guide (UM-WI-054) merged into this document ▪ Removed Manufacture Image parts
1.6	Apr 13, 2022	▪ Updated RF Test with Console section ▪ Updated reference section
1.5	Mar 28, 2022	Updated logo, disclaimer, and copyright
1.4	Mar 17, 2021	Added Note 1
1.3	Nov 21, 2019	Finalized for publication
1.2	Nov 18, 2019	▪ Editorial review ▪ Added description for OTP write command
1.1	Nov 12, 2019	▪ Added Download the AT command and General Images ▪ Added Change Boot Index section ▪ Error correction on Table 4
1.0	July 31, 2019	First Release

Status Definitions

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

RoHS Compliance

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