

# User Manual

## DA16200 Mass Production

### UM-WI-011

#### **Abstract**

*This User Manual is intended to help customers setup the hardware development environment, install the required software, and download and run an example application on the DA16200 Development Kit development platform.*

---

## Contents

<b>Abstract</b> .....	<b>1</b>
<b>Contents</b> .....	<b>2</b>
<b>Figures</b> .....	<b>2</b>
<b>Tables</b> .....	<b>3</b>
<b>1 Terms and Definitions</b> .....	<b>3</b>
<b>2 References</b> .....	<b>3</b>
<b>3 Overview</b> .....	<b>4</b>
3.1 Production Procedure .....	4
<b>4 Firmware Download</b> .....	<b>5</b>
4.1 Download via UART .....	5
4.2 Download via SPI.....	5
4.3 NVRAM Initialization .....	5
4.4 S/W Version Verification .....	6
4.5 Download the Manufacturer and General Images .....	6
<b>5 XTAL Calibration</b> .....	<b>8</b>
5.1 Frequency Calibration .....	8
<b>6 RF Test</b> .....	<b>10</b>
6.1 Test Parameter .....	10
6.2 Test Channel.....	10
6.3 Test Channel.....	10
6.4 Test Command.....	11
<b>7 Writing MAC Address</b> .....	<b>13</b>
7.1 AT GUI Tool .....	13
7.2 Console Command // Check .....	14
<b>8 Wi-Fi Function Test</b> .....	<b>15</b>
<b>9 Standby Current Measurement</b> .....	<b>15</b>
<b>10 Factory Reset</b> .....	<b>16</b>
10.1 Console Command .....	16
10.2 AT Command .....	16
<b>11 Change Boot Index</b> .....	<b>16</b>
11.1 Console Command .....	16
<b>Revision History</b> .....	<b>18</b>

## Figures

Figure 1: Version Verification with Use of AT Command .....	6
Figure 2: Calibration Procedure .....	8
Figure 3: Certification Mode Screen in AT GUI tool .....	10
Figure 4: OTP Mode Screen in AT GUI Tool.....	13
Figure 5: Network Mode Screen in AT GUI tool .....	15

## Tables

Table 1: 2MB Serial Flash Memory Map .....	5
Table 2: 4MB Serial Flash Memory Map .....	5
Table 3: AT Command to Check Version .....	6
Table 4: XTAL Offset OTP Address .....	9
Table 5: RF Test Parameters .....	10
Table 6: AT Command for Writing/Reading MAC Address .....	13
Table 7: AT Command for Factory Reset.....	16

## 1 Terms and Definitions

GUI	Graphical User Interface
UART	Universal Asynchronous Receiver/Transmitter
SPI	Serial Peripheral Interface
RF	Radio Frequency
OTP	One Time Password

## 2 References

- [1] DA16200, Datasheet, Dialog Semiconductor
- [2] UM-B-109 DA16200 EVK User Manual, Dialog Semiconductor
- [3] UM-B-111 DA16200 AT command User Manual, Dialog Semiconductor
- [4] UM-B-112 DA16200 AT GUI Tool, Dialog Semiconductor
- [5] UM-B-128 DA16200 SPI SFlash Downloader User Manual, Dialog Semiconductor
- [6] UM-B-127 DA16200 UART SFlash Downloader User Manual, Dialog Semiconductor

---

## DA16200 Mass Production

---

### 3 Overview

This document explains the production procedure for DA16200 and what options and limitations the customer needs to consider in the production process. Each procedure or the order introduced here can be omitted or changed according to customer's production environment.

The following serial interfaces are used in the production process:

- UART0: Debug console and firmware download ([Note 1](#))
- UART1: AT command
- SPI: Firmware download with high speed

DA16200 provides firmware download tools that use UART0 or SPI, and the AT GUI tool that uses AT commands for RF test, Wi-Fi function test, OTP writing and so forth. If necessary, AT-Command can be implemented in UART0 as well, and part of RF TEST and OTP write can be implemented (Contact the technical support team more information on this). It is possible to use these tools in the production process or look at the sample application source codes to make a customer's production application.

**Note 1** Option: Some RF test functions of AT commands can be implemented

#### 3.1 Production Procedure

Production is done in the following sequence:

- Firmware download
- XTAL calibration
- RF test
- Writing MAC address
- Wi-Fi function test
- Standby current measurement
- Factory reset
- Store certificate

## DA16200 Mass Production

### 4 Firmware Download

The first procedure is to write the initial images in an empty serial flash IC. DA16200 supports UART and SPI to download firmware and provides a GUI tool and sample source codes.

The download of firmware is done with an MROM prompt (boot mode).

See [Table 1](#) and [Table 2](#) for the download addresses of each firmware image.

**Table 1: 2MB Serial Flash Memory Map**

Address	Item	Size
0x0000_0000	2nd Bootloader	36 kB
0x0000_9000	Boot Index	4 kB
0x0000_A000	RTOS #0	924 kB
0x000F_1000	SLIB #0 (RamLib + TIM)	52 kB
0x000F_E000	RTOS #1	924 kB
0x001E_5000	SLIB #1 (RamLib + TIM)	52 kB

**Table 2: 4MB Serial Flash Memory Map**

Address	Item	Size
0x0000_0000	2nd Bootloader	36 kB
0x0000_9000	Boot Index	4 kB
0x0000_A000	RTOS #0	1536 kB
0x0018_A000	SLIB #0 (RamLib + TIM)	64 kB
0x0020_0000	RTOS #1	1536 kB
0x0038_0000	SLIB #1 (RamLib + TIM)	64 kB

#### 4.1 Download via UART

There are three images (`BOOT`, `SLIB` and `RTOS`) to be downloaded via UART0 with the `y modem` protocol. See section **Firmware Update** in the DA16200 EVK User Manual [\[2\]](#) about firmware download commands.

Dialog Semiconductor provides a GUI to use a download tool. For details, see UM-B-127 DA16200 UART SFlash Downloader User Manual [\[6\]](#).

#### 4.2 Download via SPI

SPI can reduce the firmware download time because the SPI interface has a higher speed. For details, see the UM-B-128 DA16200 SPI SFlash Downloader User Manual [\[5\]](#).

#### 4.3 NVRAM Initialization

Normally, the flash memory is empty before downloading firmware, so the `NVRAM` region should be initialized. You can select one of following methods to initialize the `NVRAM` region.

The console command in DA16200 prompt and the `NVRAM` region can be initialized with the following commands after the first firmware download.

```
[/DA16200] # nvram
```

DA16200 Mass Production

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

4.4 S/W Version Verification

Run the following console command or AT command to find out what the written firmware’s versions are.

- Console command `ver`

```
[/DA16200] # ver
*****
*
*          DA16200 SDK Information
* -----
*
*
* - CPU Type: Cortex-M4 (80 MHz)
* - OS Type: ThreadX 5.7
* - Serial Flash: 16 Mbits (2 MBytes)
* - SDK Type: Generic v1.0.0
* - F/W Version: RTOS-GEN01-01-7140-000000
*
*          : SLIB-GEN01-01-7089-000000
*
* - F/W Build Time: Jul  5 2019 17:35:59
* - Boot Index: 0
*
*
*****
[/DA16200] #
```

- AT command

Table 3: AT Command to Check Version

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

```
AT+VER
+VER: RTOS-GEN01-01-7140-000000,SLIB-GEN01-01-7089-000000
OK
```

Figure 1: Version Verification with Use of AT Command

4.5 Download the Manufacturer and General Images

A manufacturer's image set may be needed for calibration and an RF test during the production process. There is an effective procedure to download the General Image set and the Test Image set in one step to reduce production time.

---

**DA16200 Mass Production**

DA16200 provides two image regions (see [Table 1](#) and [Table 2](#) for each region: #0 and #1) for an OTA update. It is possible to use these regions for each image set and change the index of the boot image set. The default value of the boot index points to #0, so it will work as a manufacturer's image on first boot.

1. Go to MROM prompt and download the secondary bootloader.
2. Download the manufacturer's image to the RTOS # 0 and SLIB # 0 region.
3. Download the general image to the RTOS # 1 and SLIB # 1 regions.
4. Boot - The default boot index is #0.
5. Initialize the NVRAM.
6. Execute the test process.
7. Execute the remainder of the procedure.

## 5 XTAL Calibration

DA16200 has several OTP slots for TX power, temperature and XTAL frequency calibration. The AT GUI tool provides menus for each calibration. This chapter explains only about XTAL frequency calibration. TX power and temperature frequency calibration has been done during the ATE test.

### 5.1 Frequency Calibration

This chapter describes how to do the frequency calibration.

Figure 22 gives an example flowchart of the procedure.

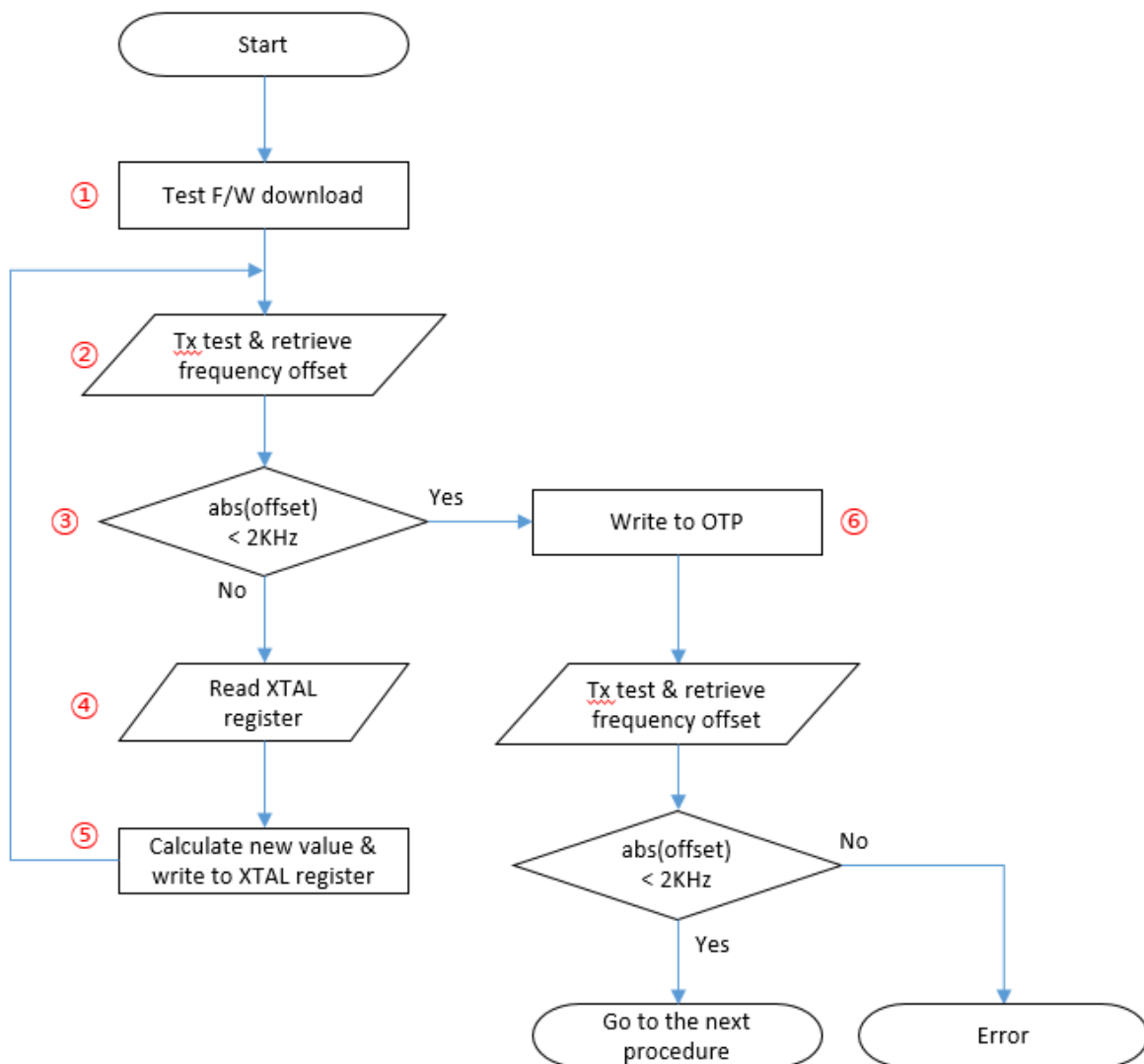


Figure 2: Calibration Procedure

1. Download the test firmware. This firmware can be the final image.
2. On the DA16200, run the TX test mode and read the frequency offset with the use of measuring equipment.
3. If the offset is below 2 kHz, then the value is within the margin. If not, then do the steps 4 and 5 to change the XTAL register value.
4. Read the XTAL register value with command `AT+XTALRD`. The example shows the AT command to read the current value of the XTAL register. The result is 0x29 in this example.

```
AT+XTALRD
```



---



---

**DA16200 Mass Production**

0x29
------

5. Calculate a new value for the XTAL register and write the XTAL register. The example code shows how to calculate a new XTAL value.

<pre> if (abs(offset) &lt; 2000) {     // Go to write OTP } else {     newXtalValue = currentXtalValue - (offset / 4000);     if (offset % 4000 &gt; 0) newXtalValue--;     else if (offset % 4000 &lt; 0) newXtalValue++;     // Go to write the XTAL register } </pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

6. Use the AT command `AT+XTALWR=<newValue>` to write a new value for the XTAL register.

<pre> AT+XTALWR=&lt;newValue&gt; OK </pre>
--------------------------------------------

7. Write the final value with command `AT+UOTPWRASC=0428,1,<newValue>` to OTP and go to the next step to confirm if the result is good.

<pre> AT+UOTPWRASC=0428,1,&lt;newValue&gt; OK </pre>
------------------------------------------------------

8. DA16200 has two slots to store the XTAL offset in the OTP memory. See [Table 44](#). To use AT command to write value at OTP address, address x 4 should be taken because address is 4-byte aligned address. For more details, see section **OTP Commands** in DA16200 AT Command User Manual [\[3\]](#)

**Table 4: XTAL Offset OTP Address**

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

## 6 RF Test

You can test TX/RX performance of DA16200 in the Certification Mode menu of the AT GUI tool. See [Figure 33](#). For more information about AT commands, see section **RF Test Function Commands** in the UM-B-112 DA16200 AT GUI Tool User Manual [\[4\]](#).

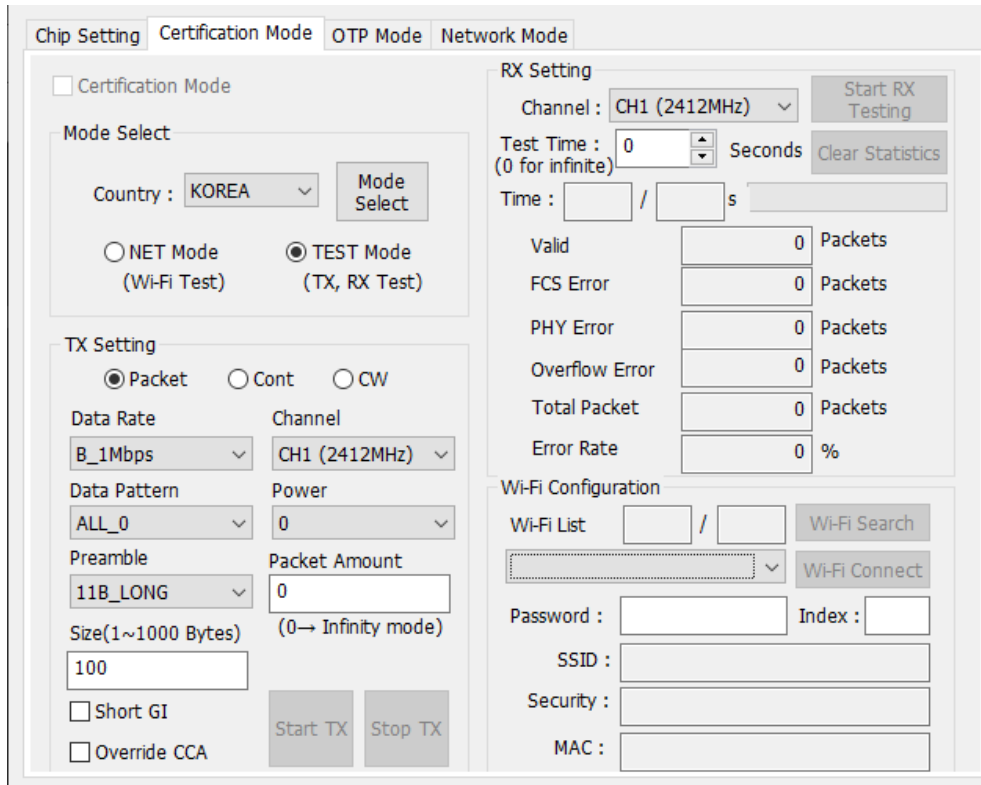


Figure 3: Certification Mode Screen in AT GUI tool

### 6.1 Test Parameter

Basic RF test parameters are listed in [Table 5](#).

### 6.2 Test Channel

DA16200 supports up to CH14, but it is highly recommended to check what the performance is at CH1 (2,412 MHz), CH7 (2,442 MHz) and CH13 (2,472 MHz).

### 6.3 Test Channel

To confirm the best performance of the product, the recommendation is to check the test parameter of the Receiver and Transmitter mentioned in [Table 55](#).

Table 5: 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

---



---

**DA16200 Mass Production**

Test Parameter	802.11 B	802.11 G	802.11 N (HT20)
	Tx Carrier Leakage	Tx Carrier Leakage	Tx Carrier Leakage
	Spectrum Emission Mask	Spectrum Emission Mask	Spectrum Emission Mask
Rx	Sensitivity	Sensitivity	Sensitivity

## 6.4 Test Command

RFTESTSTART	RF Test Mode Start	atcmd_rftest	Lmacinit()cmd_lmac_ops_start()	Enable mac hardware
RFTESTSTOP	RF Test Mode Stop	atcmd_rftest	cmd_lmac_ops_stop()	Disable mac hardware (need reset)
RFTX	RF TX packet send	atcmd_rftx	cmd_lmac_rftx()	<p>Send Tx packet &lt;Ch&gt;,&lt;BW&gt;,&lt;numFrames&gt;,&lt;frameLen&gt;,&lt;txRate&gt;,&lt;txPower&gt;,&lt;dstAddr&gt;,&lt;bssid&gt;,&lt;htEnable&gt;,&lt;GI&gt;,&lt;greenField&gt;,&lt;preambleType&gt;,&lt;qosEnable&gt;,&lt;ackPolicy&gt;,&lt;scrambler&gt;,&lt;aifsnVal&gt;,&lt;ant&gt;</p> <p>Ch : frequency, BW : always 0, not used it numFrames : number of frames frameLen : frame length txRate: rate of TX packet ( see below ) txPower: power of TX packet ( 0 is max 15 is min ) dstAddr, change destination MAC address , "11:22:33:44:55:66" Bssid : change BSSID MAC address, "11:22:33:44:55:66" htEnable : always 0 GI : short GI or long GI, "short" or "long" greenField : GreenField , "on" or "off" preambleType : Preamble Type, "long" or "short" qosEnable: QOS enable, "on" or "off" ackPolicy : Ack Polick , "NO", "NORM", "BA" &amp; "CBA" Scrambler : always 0 aifsnVal : chage EDCA aifsn value (0 to 15) Ant : always 0</p>
RFTXSTOP	RF TX Stop	atcmd_rftx	txp_on_off=0	Send TX packet operation off
RFCHANNEL	RF Channel	atcmd_rf_channel	int freq set_rf_channel(freq);	Change rf frequency

DA16200 Mass Production

RFPER	RF Display PER	atcmd_rf_per	get_per(0, &pass, &fcs, &phy, &overflow); PRINTF_ATCMD D("\r\n%d %d %d %d\r\n", pass, fcs, phy, overflow);	Print rcvd, errored, phy_error_tot, rx_ovfl_tot  get_per(0, &rcvd, &errored, &phy_error_tot, &rx_ovfl_tot); Rcvd: RX_FRAME_COUNT Errored : FCS ERROR COUNT Phy_error_tot : RX PHY ERROR COUNT Rx_ovfl_tot : RX FIFO OVERFLOW COUNT
RFPERRESET	RF Reset PER	atcmd_rf_per	PRINTF_ATCMD D("\r\nRESET PER COUNT !\r\n"); get_per(1, NULL, NULL, NULL, NULL);	Reset PER count

txRate	means
b1	11b 1Mbps
b2	11b 2Mbps
b5.5	11b 5.5Mbps
b11	11b 11Mbps
g6	11g 6Mbps
g9	11g 9Mbps
g12	11g 12Mbps
g18	11g 18Mbps
g24	11g 24Mbps
g36	11g 36Mbps
g48	11g 48Mbps
g54	11g 54Mbps
n6.5	11n 6.5Mbps
n13	11n 13Mbps
n19.5	11n 19.5Mbps
n26	11n 26Mbps
n39	11n 39Mbps
n52	11n 52Mbps
n58.5	11n 58.5Mbps
n65	11n 65Mbps

DA16200 Mass Production

## 7 Writing MAC Address

The MAC address written in the OTP memory is 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 you write a mac address to a DA16200 chip, the last byte of the mac address should be bigger 'by 2' than that of 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 and so on. The last digit of the WLAN0 MAC address should be an even number.

### 7.1 AT GUI Tool

You can write MAC addresses in OTP Mode menu in the AT GUI tool. DA16200 provides 4 slots to store MAC addresses in the OTP memory. When a new MAC address is written, the previous slot should be invalidated. See DA16200 AT GUI Tool User Manual [4].

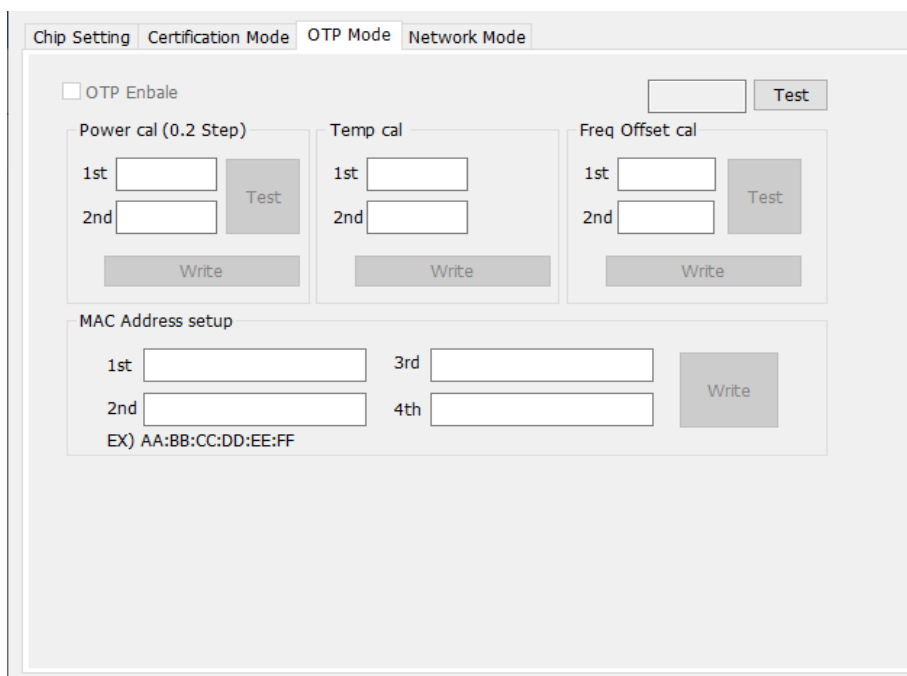


Figure 4: OTP Mode Screen in AT GUI Tool

Table 6: AT Command for Writing/Reading MAC Address

Command	Parameters	Description
AT+WFOTP	<mac>	Write MAC address in the OTP memory. An old MAC address in the OTP will be invalidated if one exists. There are four mac address slots available in OTP, so only a maximum of four MAC addresses are written in total at production. Response: OK or ERROR For 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 will be used as WLAN1 MAC address.
AT+WEMAC	(none)	Get the current MAC address of the activated WLAN interface.

---



---

**DA16200 Mass Production**

Command	Parameters	Description
		DA16200 provides three types of MAC addresses (OTP MAC address, user MAC address and spoofing MAC address). The priority is OTP < User < Spoofing. Response: +WEMAC:<mac>

## 7.2 Console Command // Check

1. Use command `setotpmac` as shown in the example code, 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
```

2. 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
```

## DA16200 Mass Production

### 8 Wi-Fi Function Test

To test the basic Wi-Fi function (station and soft-AP), use the Network Mode menu in the AT GUI tool. See DA16200 AT GUI Tool User Manual [4]. And for related commands, see section **Network Function Commands** in DA16200 AT Command User Manual [3].

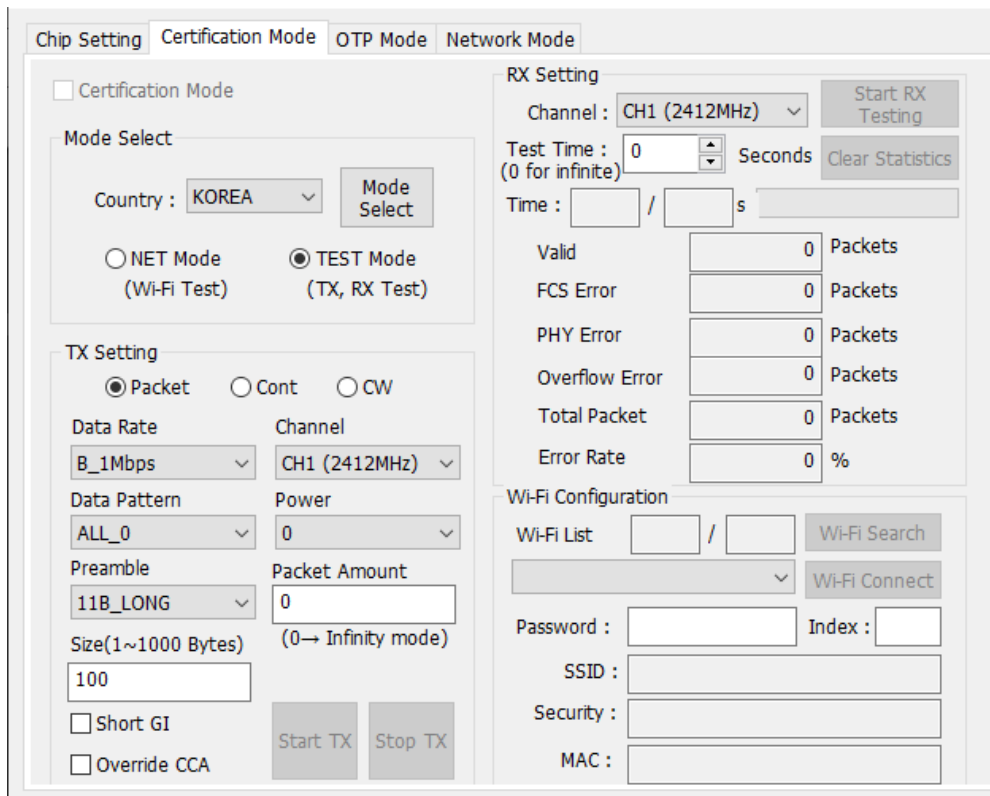


Figure 5: Network Mode Screen in AT GUI tool

### 9 Standby Current Measurement

You may need to measure the standby current consumption to detect any current leakage in the DA16200. The following example code makes DA16200 go to 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 section **Test Methods** in UM-B-109 DA16200 EVK User Manual [2] about measuring current consumption.

## 10 Factory Reset

Many profiles may be written in the NVRAM during the production process so DA16200 may need to be initialized to the factory status. The command code example in 10.1 or the use of the AT command in Table 77 erases all user NVRAM items.

### 10.1 Console Command

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

Start Factory-Reset ...

Rebooting....
```

### 10.2 AT Command

Table 7: AT Command for Factory Reset

Command	Parameters	Description
ATF	(none)	DA16200 factory reset.

## 11 Change Boot Index

To change from the manufacturer's image to the General Image, change the boot index and then reboot. See the example code in 11.1. After the reboot is completed, check if the version printed at boot and the boot index values have changed. See section 4.4.

### 11.1 Console Command

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

>>> Network Interface (wlan0): DOWN
[wpasupplicant_event_disassoc] CTRL-EVENT-DISCONNECTED bssid=aa:ab:ac:ad:ae:af
reason=3 locally_generated=1
[wpasupplicant_ev_disassoc_fin] Disconnect event - remove keys
RaLIB is relocated to RETMEM (20f815c0, 564, 12904718, 12904718)
P.TIM is relocated to RETMEM (20f835c0, 3)
dpm_init_retmemory::316 DPM INIT CONFIGURATION(1)

Wakeup source is 0x0

*****
*
*          DA16200 SDK Information
* -----
*
* - CPU Type: Cortex-M4 (80 MHz)
* - OS Type: ThreadX 5.7
* - Serial Flash: 16 Mbits (2 MBytes)
* - SDK Type: Generic v1.0.0
* - F/W Version: RTOS-GEN01-01-7140-000000
```



---

DA16200 Mass Production

```
*           : SLIB-GEN01-01-7089-000000
* - F/W Build Time: Jul  5 2019 17:35:59
* - Boot Index: 1
*
*****
```

## Revision History

Revision	Date	Description
1.5	28-Mar-2022	Update logo, disclaimer, copyright.
1.4	17-Mar-2021	Add Note 1 at page 4
1.3	21-Nov-2019	Finalized for publication
1.2	18-Nov-2019	Editorial review Add description for OTP write command at page 9
1.1	12-Nov-2019	Add 2.5 Download the Manufacture and General Images Add Section 9 Change Boot Index Error correction on Table 4
1.0	31-Jul-2019	First Release.

---

**DA16200 Mass Production****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**

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.

---

## DA16200 Mass Production

---

### Important Notice and Disclaimer

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

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

## Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu

Koto-ku, Tokyo 135-0061, Japan

[www.renesas.com](http://www.renesas.com)

### Contact Information

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

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

### Trademarks

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