

# User Manual

## DA16200 DA16600 Host Interface and AT Command

UM-WI-003

### Abstract

*This document describes how to use Host interfaces and AT commands in DA16200 and DA16600.*

---

## DA16200 DA16600 Host Interface and AT Command

### Contents

<b>Abstract</b>	<b>1</b>
<b>Contents</b>	<b>2</b>
<b>Figures</b>	<b>3</b>
<b>Tables</b>	<b>4</b>
<b>1 Terms and Definitions</b>	<b>6</b>
<b>2 References</b>	<b>7</b>
<b>3 Host Interface</b>	<b>8</b>
3.1 UART Host Interface	8
3.1.1 PIN MUX Configuration	8
3.2 SPI Host Interface	8
3.2.1 PIN MUX Configuration	8
3.2.2 SPI Protocol	9
3.2.3 AT Command – Sequences and Structures	13
3.2.4 ESC Command – Sequences and Structures	14
3.2.5 Header Format	16
3.2.6 SPI Definition and Structures	17
3.3 SDIO Host Interface	18
3.3.1 PIN MUX Configuration	18
3.3.2 SDIO Protocol	18
3.3.3 AT Command – Sequences and Structures	22
3.3.4 ESC Command – Sequences and Structures	23
3.3.5 SDIO Definition and Structures for Implementation	24
<b>4 AT Commands</b>	<b>26</b>
4.1 How to Include AT Command Feature in SDK	26
4.1.1 How to Run AT-CMD on UART	26
4.1.2 How to Run AT-CMD on SPI	26
4.1.3 How to Run AT-CMD on SDIO	27
4.2 AT Command Format	28
4.2.1 Basic Command Format	28
4.2.2 Extended Command Format	28
4.2.3 Response Format	29
<b>5 Function AT Commands</b>	<b>30</b>
5.1 Basic Function Commands	30
5.2 Network Function Commands	44
5.3 Wi-Fi Function Commands	53
5.4 Wi-Fi Function Commands for WPA3	76
5.5 Wi-Fi Function Commands for WPA Enterprise	79
5.5.1 WPA-Enterprise Connection Example	81
5.6 Advanced Function Commands	81
5.6.1 MQTT Commands	81
5.6.2 HTTP-Client Commands	110

## DA16200 DA16600 Host Interface and AT Command

5.6.3	HTTP-Server Commands .....	116
5.6.4	WebSocket-Client Commands .....	117
5.6.5	OTA Commands .....	118
5.6.6	Zeroconf Commands .....	129
5.6.7	Provision Commands .....	133
5.6.8	Bluetooth® LE Commands .....	134
5.6.9	Transfer Function Commands .....	135
5.6.10	RF Test Function Commands .....	151
5.6.11	System and Peripheral Function Commands .....	157
<b>6</b>	<b>Examples Using AT Commands .....</b>	<b>170</b>
6.1	Data Transfer Test .....	170
6.1.1	TCP Server Socket Test .....	170
6.1.2	TCP Client Socket Test .....	171
6.1.3	UDP Socket Test .....	172
	<b>Appendix A License Information .....</b>	<b>174</b>
	<b>Appendix B HTTP API Return Values .....</b>	<b>175</b>
B.1	Return Value as Defined by NetX Duo HTTP .....	175
B.2	Return Value as Defined by LWIP HTTP .....	176
	<b>Appendix C User UART Configuration .....</b>	<b>177</b>
C.1	How to Run AT-CMD on UART2 .....	177
C.2	User UART Configuration .....	177
C.3	Use Case .....	178
	<b>Appendix D DA16200/DA16600 Cipher Suites .....</b>	<b>179</b>
	<b>Appendix E Reason Code For Wi-Fi Connection Failure or Disconnection .....</b>	<b>181</b>
	<b>Appendix F Fast-reconnect on SLEEP Mode 2 .....</b>	<b>183</b>
F.1	Technical overview .....	183
	<b>Appendix G Bluetooth® LE Coexistence Feature .....</b>	<b>184</b>
	<b>Appendix H Wi-Fi Passive-Scan .....</b>	<b>185</b>
H.1	Passive-Scan with specified Channel and Scan-time Limit .....	185
H.2	Passive-Scan Result .....	185
H.3	Passive-Scan Stop .....	185
H.4	Passive Scan Sequence .....	185
	<b>Appendix I Detailed Error Codes for AT Command .....</b>	<b>187</b>
	<b>Appendix J AT Command Development Environment Configuration .....</b>	<b>204</b>
J.1	How to Connect the DA16200/DA16600 Board .....	204
J.2	Configure the Serial Port for UART .....	205
J.3	Configuration for MCU Wake-Up (Optional) .....	206
	<b>Revision History .....</b>	<b>208</b>

## Figures

Figure 1: Basic Format .....	9
------------------------------	---

## DA16200 DA16600 Host Interface and AT Command

Figure 2: Write Sequence .....	10
Figure 3: Structure for Write Operation .....	10
Figure 4: Read Sequence .....	11
Figure 5: Structure for Read Operation .....	12
Figure 6: AT Command Sequence .....	13
Figure 7: Structure of AT CMD .....	13
Figure 8: ESC Command Sequence .....	15
Figure 9: Structure of ESC CMD .....	15
Figure 10: SPI Signals for Write Request .....	16
Figure 11: SPI Signals for Read Response .....	17
Figure 12: Basic Format .....	19
Figure 13: Write Sequence .....	19
Figure 14: Structure .....	20
Figure 15: Read Sequence .....	21
Figure 16: Structure .....	21
Figure 17: AT Command Sequence .....	22
Figure 18: Structure .....	22
Figure 19: ESC Command Sequence .....	23
Figure 20: Structure .....	24
Figure 21: Example Sequence to Initiate AT-CMD through SDIO Interface .....	27
Figure 22: Example Sequence to Initiate MQTT Protocol with DPM .....	98
Figure 23: Procedure to Send MQTT Messages .....	99
Figure 24: Procedure to Process MQTT Messages .....	100
Figure 25: Broker Console - CleanSession=1 Connection .....	102
Figure 26: Broker Console - CleanSession=0 Connection .....	102
Figure 27: IO Ninja – TCP Client Socket Setting .....	171
Figure 28: IO Ninja – TCP Server Socket Setting .....	172
Figure 29: IO Ninja – UDP Socket Setting .....	173
Figure 30: Passive Scan Report .....	186
Figure 31: AT Command Development Environment .....	204
Figure 32: Check COM Ports on Device Manager .....	205
Figure 33: Initial Setup to Get Started with AT Command .....	206
Figure 34: GPIO Wake-Up .....	207

## Tables

Table 1: PIN MUX Configuration - UART .....	8
Table 2: PIN MUX Configuration - SPI .....	9
Table 3: SPI Address List .....	9
Table 4: SPI CMD Format .....	10
Table 5: SPI Definition .....	17
Table 6: SPI Response Structure .....	17
Table 7: SPI Request Structure .....	17
Table 8: PIN MUX Configuration - SDIO .....	18
Table 9: Address List .....	19
Table 10: SDIO Definition .....	24
Table 11: SDIO Response Structure .....	25
Table 12: SDIO Request Structure .....	25
Table 13: Basic Function Command List .....	30
Table 14: Initiation Response List .....	44
Table 15: Network Function Command List .....	44
Table 16: Certificate Command .....	52
Table 17: Wi-Fi Function Command List .....	53
Table 18: Wi-Fi Function Response List .....	75
Table 19: List of WPA3-Relevant Wi-Fi Function Commands .....	76
Table 20: WPA-Enterprise Wi-Fi Function Commands .....	79

## DA16200 DA16600 Host Interface and AT Command

Table 21: WPA-Enterprise Network Function Command .....	80
Table 22: MQTT Configuration Command List .....	82
Table 23: MQTT Operation Command List.....	90
Table 24: MQTT Optional Configuration Commands .....	92
Table 25: MQTT Response List .....	94
Table 26: CleanSession and QoS Matrix in Message Rx.....	104
Table 27: CleanSession and QoS Matrix in Message Tx .....	104
Table 28: HTTP-Client Command List .....	110
Table 29: HTTP-Client Response List .....	114
Table 30: HTTP-Server Command List.....	116
Table 31: WebSocket-Client Command List.....	117
Table 32: WebSocket-Client Response List .....	118
Table 33: OTA Command List.....	119
Table 34: OTA Response List.....	126
Table 35: OTA Response Code List .....	127
Table 36: Zeroconf Command List.....	129
Table 37: Provision Command List .....	133
Table 38: atcmd_provision_stat Enumeration.....	133
Table 39: Bluetooth® LE Command List.....	134
Table 40: Socket Command List.....	135
Table 41: Socket Connection Response List.....	139
Table 42: Data Transmission Command .....	140
Table 43: Data Reception Responses .....	143
Table 44: Secure Socket Command List .....	146
Table 45: RF Test Command List .....	151
Table 46: RF Test Examples.....	156
Table 47: SPI Command List .....	157
Table 48: OTP Command List.....	158
Table 49: OTP Memory Address for Writing MAC Address.....	160
Table 50: Size of Memory by XTAL Offset.....	160
Table 51: XTAL Command List.....	160
Table 52: Flash Dump Command List .....	161
Table 53: GPIO Command List.....	161
Table 54: LED Command List.....	164
Table 55: PWM Command List .....	165
Table 56: ADC Command List .....	166
Table 57: I2C Command List .....	167
Table 58: Sleep Command List.....	168
Table 59: CALWL Command List .....	169
Table 60: AT Command Error Codes .....	187

## DA16200 DA16600 Host Interface and AT Command

### 1 Terms and Definitions

AP	Access Point
ASCII	American Standard Code for Information Interchange
AT	Attention
CA	Certificate Authority
CCMP	Counter Mode Cipher Block Chaining Message Authentication Code Protocol
CID	Client ID
CMD	Command
COM	Communication Port
CRC	Cyclic Redundancy Check
CW	Continuous Wave
DHCP	Dynamic Host Configuration Protocol
DPM	Dynamic Power Management
EAP	Extensible Authentication Protocol
ESS	Extended Service Set
FW	Firmware
GPIO	General Purpose Input Output
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
JSON	JavaScript Object Notation
LMAC	Low MAC
MAC	Medium Access Control
MCU	Micro Controller Unit
MQTT	Message Queuing Telemetry Transport
NVRAM	Non-Volatile RAM
OTA	Over the Air
OTP	One-Time Programmable
OWE	Opportunistic Wireless Encryption
PBC	Push Button Connection
PC	Personal Computer
PER	Packet Error Rate
PSK	Pre-Shared Key
QoS	Quality of Service
RTC	Real-Time Clock
RTOS	Real-Time Operating System
RTS	Request to Send
RX	Receive
SAE	Simultaneous Authentication of Equals
SDK	Software Development Kit
SDIO	Secure Digital Input Output
SNTP	Simple Network Time Protocol
SPI	Serial Peripheral Interface
SSID	Service Set Identifier

---

**DA16200 DA16600 Host Interface and AT Command**

STA	Station
TCP	Transport Control Protocol
TIM	Traffic Indication Map
TKIP	Temporal Key Integrity Protocol
TLS	Transport Layer Security
TX	Transmit
UART	Universal Asynchronous Receiver-Transmitter
UDP	User Datagram Protocol
USB	Universal Serial Bus
URL	Universal Resource Locator
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network
WMM	Wi-Fi Multimedia
WPA	Wi-Fi Protected Access version 1
WPA2	Wi-Fi Protected Access version 2
WPS	Wi-Fi Protected Setup
XTAL	Crystal

## 2 References

- [1] UM-WI-056, DA16200 DA16600, FreeRTOS Getting Started Guide, User Manual, Renesas Electronics
- [2] UM-WI-046, DA16200 DA16600, FreeRTOS SDK Programmer Guide, User Manual, Renesas Electronics
- [3] DA16200, Datasheet, Renesas Electronics
- [4] UM-WI-004, DA16200, AT GUI Tool, User Manual, Renesas Electronics

## DA16200 DA16600 Host Interface and AT Command

### 3 Host Interface

This application note describes how an external processor system (refer to as External Host) communicates with a DA16200 via SPI and SDIO physical interface protocols. This document also includes the AT Command Protocol to be used with the External Host.

#### 3.1 UART Host Interface

##### 3.1.1 PIN MUX Configuration

DA16200 can use two interfaces, UART1 and UART2, and DA16600 can use only UART2. UART1 can be assigned to GPIOA[1:0], GPIOA[3:2], GPIOA[5:4], or GPIOA[7:6], and UART2 can be assigned to GPIOA[11:10] or GPIOC[7:6]. UART1 can use the hardware flow control via GPIOA[5:4] PIN, but there is no available PIN for UART2 hardware flow control.

For example:

- Assign GPIOA[5:4] as UART1 interface  
`_da16x_io_pinmux(PIN_CMUX, CMUX_UART1d);` // For GPIOA\_4, 5(UART1)
- Assign GPIOC[7:6] as UART2 interface  
`_da16x_io_pinmux(PIN_UMUX, UMUX_UART2GPIO);` // For GPIOC\_6, 7(UART2) ,  
`GPIOC_8(GPIO)`

**Table 1: PIN MUX Configuration - UART**

UART Interface	GPIO	Signal Name
UART1	GPIOA[0] (Note 1)	TXD
	GPIOA[1]	RXD
	GPIOA[2]	TXD
	GPIOA[3]	RXD
	GPIOA[4]	TXD
	GPIOA[5]	RXD
	GPIOA[6]	TXD
	GPIOA[7]	RXD
UART1 H/W flow control	GPIOA[4]	RTS
	GPIOA[5]	CTS
UART2	GPIOA[10]	TXD
	GPIOA[11]	RXD
	GPIOC[6]	TXD
	GPIOC[7]	RXD

**Note 1** Detailed information about PIN multiplexing can be found in DA16200 Datasheet. See Ref. [3].

#### 3.2 SPI Host Interface

##### 3.2.1 PIN MUX Configuration

The SPI slave interface can be assigned to GPIOA[1:0], GPIOA[3:2], GPIOA[7:6], GPIOA[9:8] or GPIOA[11:10] in DA16200.

For example) Assign GPIOA[3:2] or GPIOA[9:8] as SPI slave interface.



## DA16200 DA16600 Host Interface and AT Command

- `_da16x_io_pinmux(PIN_BMUX, BMUX_SPIs);` // For GPIOA 2,3
- `_da16x_io_pinmux(PIN_EMUX, EMUX_SPIs);` // For GPIOA 8,9

**Table 2: PIN MUX Configuration - SPI**

GPIO	Signal Name
GPIOA[0]	MISO
GPIOA[1]	MOSI
GPIOA[2]	CS
GPIOA[3]	CLK
GPIOA[6]	CS
GPIOA[7]	CLK
GPIOA[8]	MISO
GPIOA[9]	MOSI
GPIOA[10]	MISO
GPIOA[11]	MOSI

### 3.2.2 SPI Protocol

#### 3.2.2.1 Message Format

The format of the messages sent/received to/from the external processor is the DA16200 protocol format over SPI physical interface. Message format and parameters included in DA16200 are outlined in [Figure 1](#).



**Figure 1: Basic Format**

#### I. Address

The address list used by External Host is outlined.

**Table 3: SPI Address List**

Address Type	Address
General Command (Write Request)	0x50080254
AT Command	0x50080260
Response Command	0x50080258
Buffer Address	Received from slave in response message

#### II. CMD

The format of CMD fields is outlined in [Table 4](#).

## DA16200 DA16600 Host Interface and AT Command

**Table 4: SPI CMD Format**

Bit	Field	Description
7	Auto_Inc	1: Internal Address auto-increment 0: Address Fixed (Not used)
6	Read/Write	1: Read 0: Write
5:2		Not Used
1:0	CHIP_ID[1:0]	00: CHIP #0 (Default)

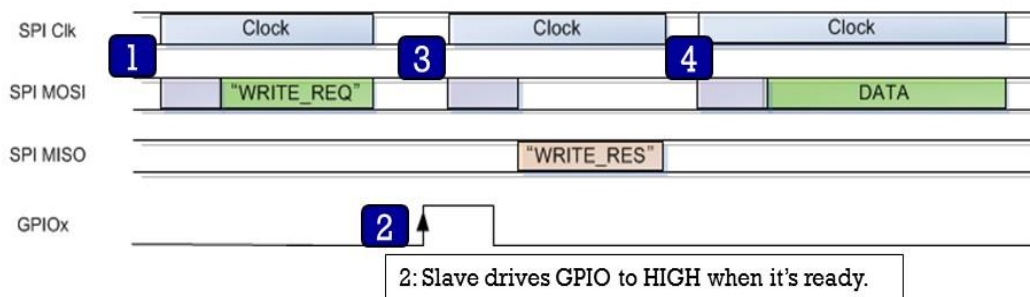
### III. Length

Payload length of the data field.

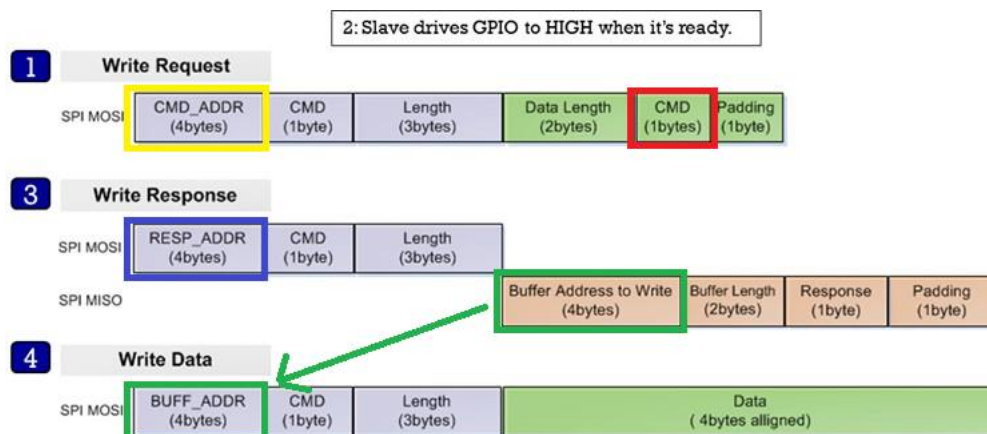
#### 3.2.2.2 Write Sequence

Host to Slave write operations are performed in three SPI transactions as shown in [Figure 2](#).

**Write Sequence (Host to Slave)**



**Figure 2: Write Sequence**



**Figure 3: Structure for Write Operation**

1. The Host sends a `WRITE_REQ` command (0x80, red rectangle in [Figure 3](#)) to the General Command address (0x50080254) (yellow rectangle in [Figure 3](#)).
2. The Host should wait for GPIO interrupt line is High from slave.
3. The Host reads the Write Response message by Response Command address (0x50080258, blue rectangle in [Figure 3](#)) and parse it using `struct _st_host_response` (see [Table 5](#)).

## DA16200 DA16600 Host Interface and AT Command

- The Host sends data to address (BUFF\_ADDR) which is received from the Slave in the Write Response message (green rectangle in [Figure 3](#)).

### NOTE

Buffer Length field contains the length of Data field, and it should be a multiple of 4. Padding field contains number of padded bytes in the Data field due to 4-byte aligned. For example, if the length of the actual data is 11 bytes, the Buffer Length will be 12 (multiples of 4) and Padding field will be 1.

$$\text{Buffer Length (12)} = \text{Actual data length (11)} + \text{number of padded bytes (1)}$$

Host can get the length of actual data using Buffer Length and Padding fields.  
Note that the usage of Padding field is applied to the SDK v3.2.8.0 or later.

An interval of several hundred microseconds is required between the “3” and “4” stages. If the interval between the two stages is too short, there is a possibility that two Interrupt Events are recognized as one. The interval depends on the type of application or CPU load. Roughly, when the CPU clock is 120 MHz, an interval of around 300  $\mu$ s is required.

### Example

When the host wants to write 8- byte data (0x8877665544332211) to DA16200:

- Host sends: (0x50-0x08-0x02-0x54)-(0x80)-(0x00-0x00-0x04)-(0x08-0x00-0x80-0x00)
- Host waits until GPIO interrupt line is high from DA16200.
- Host sends (0x50-0x08-0x02-0x58)-(0xC0)-(0x00-0x00-0x08), then reads responses from DA16200.  
Assume the buffer address from Slave is 0x12345678 for easy description.  
Then the read data should be 0x78-0x56-0x34-0x12-0x08-0x00-0x81-0x00.
- Host sends  
(0x12-0x34-0x56-0x78)-(0x80)-(0x00-0x00-0x08)-(0x11-0x22-0x33-0x44-0x55-0x66-0x77-0x88)  
Note that the payload data is transmitted MSB first and little-big endian system (see [Figure 10](#)).

### 3.2.2.3 Read Sequence and Structure

[Figure 4](#) shows a Slave device transmitting data to the Host when payload is available. This sequence is performed in two SPI transactions.

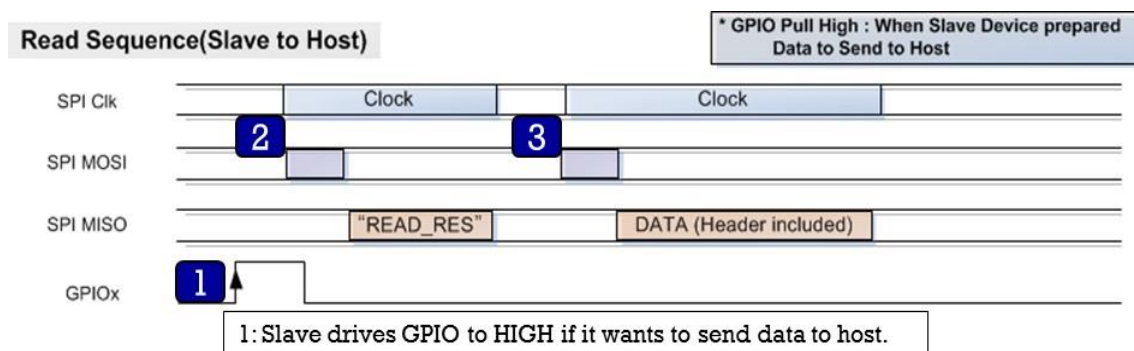
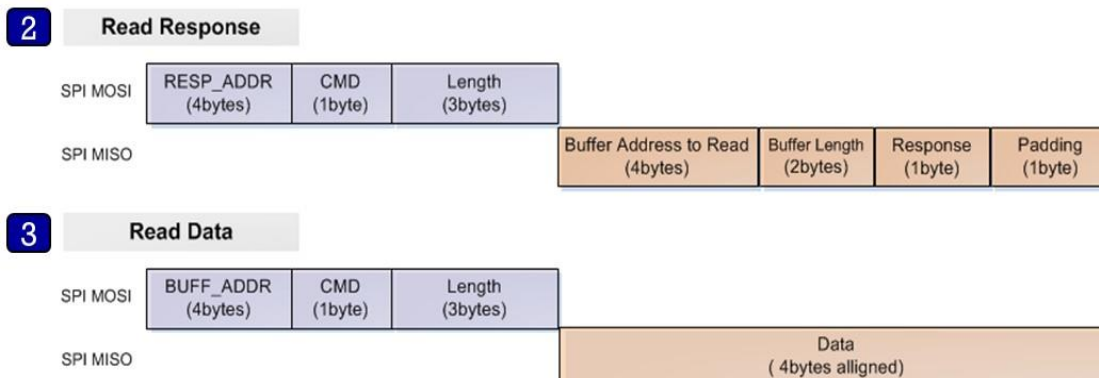


Figure 4: Read Sequence

## DA16200 DA16600 Host Interface and AT Command



**Figure 5: Structure for Read Operation**

1. The slave toggles the interrupt line high to inform the Host when data is available.
2. The Host reads the response message from Response Command address (0x50080258, blue rectangle in Figure 5) and parses it using `struct _st_host_response` (see Table 6).
3. The Host reads data from address (BUFF\_ADDR) which is received from Slave in the response message (green rectangle in Figure 5).

### NOTE

Buffer Length field contains the length of Data field, and it should be a multiple of 4. Padding field contains number of padded bytes in the Data field due to 4-byte aligned. For example, if the length of the actual data is 11 bytes, the Buffer Length will be 12 (multiples of 4) and Padding field will be 1.

$$\text{Buffer Length (12)} = \text{Actual data length (11)} + \text{number of padded bytes (1)}$$

Host can get the length of actual data using Buffer Length and Padding fields.

Note that the usage of Padding field is applied to the SDK v3.2.8.0 or later.

There is a 200 ms timeout between reading the response after the interrupt occurs and reading the data after reading the response. If host requires more than 200 ms between each interval, change the timeout value accordingly.

An interval of several hundred microseconds is required between the “2” and “3” stages. If the interval between the two stages is too short, there is a possibility that two Interrupt Events are recognized as one. The interval differs depending on the type of application or CPU load. Roughly, when the CPU clock is 120 MHz, an interval of around 300 μs is required.

### Example

1. When the host becomes high on GPIO interrupt line from DA16200, the host sends: (0x50-0x08-0x02-0x58)-(0xC0)-(0x00-0x00-0x08), then read response from DA16200. Assume the buffer address from Slave is 0x12345678 for easy description and the data length to be sent from DA16200 is 8 bytes.
  2. The read data should be 0x78-0x56-0x34-0x12-0x08-0x00-0x83-0x00.
  3. Host sends: (0x12-0x34-0x56-0x78)-(0xC0)-(0x00-0x00-0x08), then read data from DA16200.
- Note that the read data is transmitted MSB first and little-big endian system (see Figure 11).

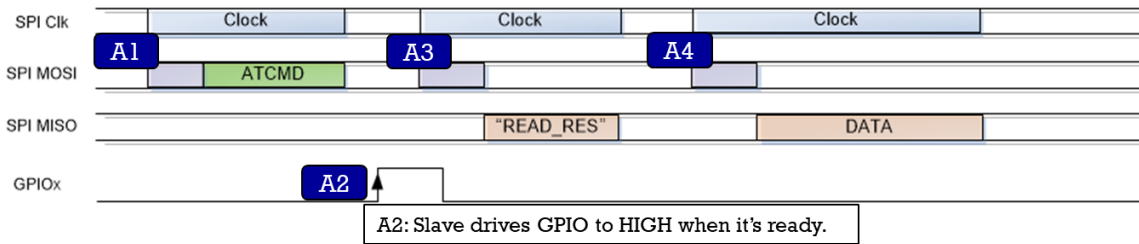
## DA16200 DA16600 Host Interface and AT Command

### 3.2.3 AT Command – Sequences and Structures

AT commands are instructions used to control a modem. AT is the abbreviation of Attention. Every command line starts with *AT* or *at*. Start *AT* is the prefix that informs the modem about the start of a command line. It is not part of the AT command name.

Figure 6 shows how to use the AT command via SPI on DA16200. This is because AT command uses a predetermined address and the maximum size of data is defined.

#### Command Sequence (Host to Slave)



#### Read Sequence (Slave to Host)

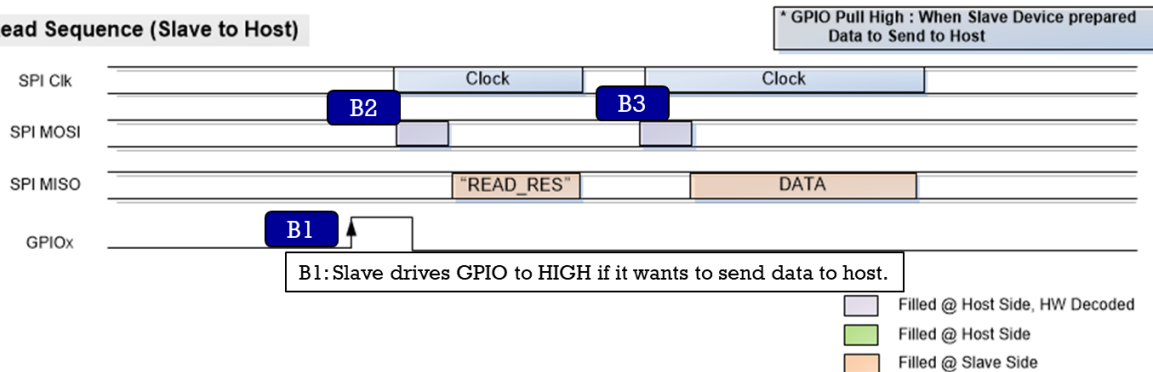


Figure 6: AT Command Sequence

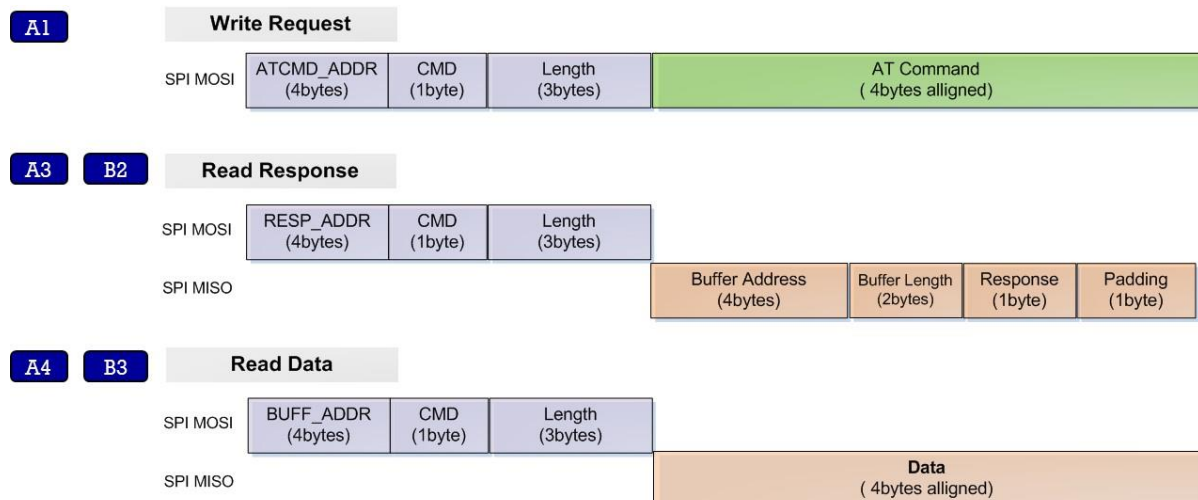


Figure 7: Structure of AT CMD

**A1:** The Host sends an AT command to AT command address.

**A2:** The Host waits for GPIO interrupt line to go high.

## DA16200 DA16600 Host Interface and AT Command

**A3:** The Host reads the response message from address and parses it using `struct _st_host_response`.

**A4:** The Host reads OK/Error, or data from address (BUF\_ADDR), depending on the command type

### Example

- ① To write `AT+VER` command to the DA16200, the host sends:  
(0x50-0x08-0x02-0x60)-(0x80)-(0x00-0x00-0x08)-('A'-'T'-'+'-'V'-'E'-'R'-0x00-0x00)
- ② The read sequence after writing is the same as the example of B1~B3 below.  
Note that the payload data is transmitted MSB first and little-big endian system (see [Figure 10](#)).

**B1:** The Slave toggles high the interrupt line to inform Host when data is available.

**B2:** The Host reads the response message from Response Command address, and parses it using `struct _st_host_response`.

**B3:** The Host reads data from address (BUF\_ADDR) parsed from the response message.

There is a 200 ms timeout between reading the response after the interrupt occurs and reading the data after reading the response. If host requires more than 200 ms between each interval, change the timeout value accordingly.

An interval of several hundred microseconds is required between the “A3” and “A4” stages, “B2” and “B3” stages. If the interval between the two stages is too short, there is a possibility that two Interrupt Events are recognized as one. The interval differs depending on the type of application or CPU load. Roughly, when the CPU clock is 120 MHz, an interval of around 300  $\mu$ s is required.

### Example

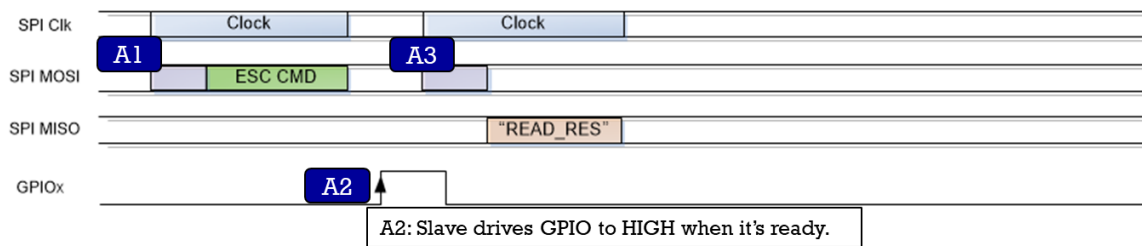
- ① When the host becomes *high* on GPIO interrupt line from DA16200, the host sends:  
(0x50-0x08-0x02-0x58)-(0xC0)-(0x00-0x00-0x08), then read response from DA16200.  
Assume the buffer address from Slave is 0x12345678 for easy description and the data length to be sent from DA16200 is 8 bytes.
- ② The read data should be 0x78-0x56-0x34-0x12-0x08-0x00-0x83-0x00.
- ③ Host sends: (0x12-0x34-0x56-0x78)-(0xC0)-(0x00-0x00-0x08), then read data from DA16200.
- ④ Note that the read data is transmitted MSB first and little-big endian system (see [Figure 11](#)).

### 3.2.4 ESC Command – Sequences and Structures

[Figure 8](#) shows how to use the ESC Command via SPI on DA16200. This is because ESC command uses a predetermined address and the maximum size of data is defined.

## DA16200 DA16600 Host Interface and AT Command

### Command Sequence (Host to Slave)



### Read Sequence (Slave to Host)

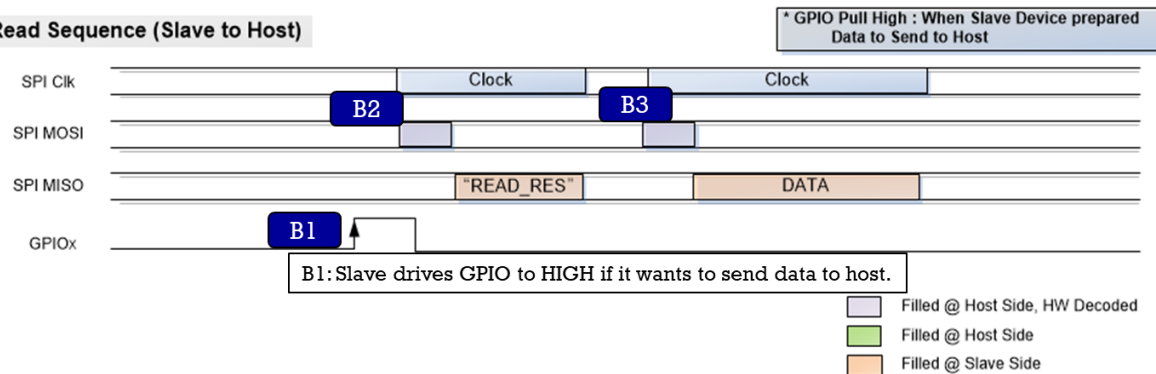


Figure 8: ESC Command Sequence

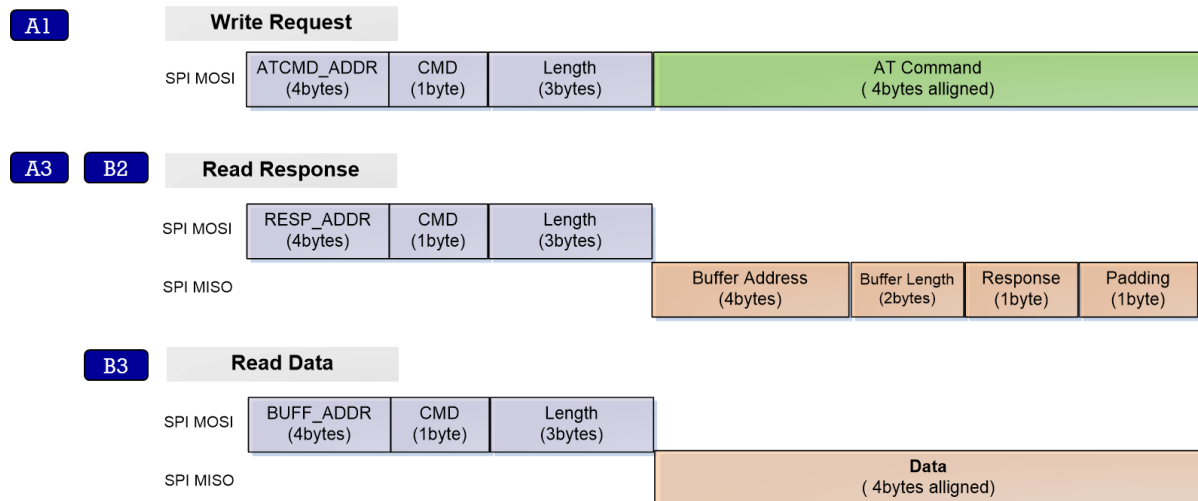


Figure 9: Structure of ESC CMD

**A1:** The Host sends an ESC command to AT command address.

**A2:** The Host waits for GPIO interrupt line to go high.

**A3:** The Host reads the response message from address and parses it using `struct _st_host_response`. The result for `esc` command is sent to the host as the `response` field of `struct _st_host_response`. The `response` field is a 1-byte decimal value. A value of 0x20 is a result of `OK`. All other values are `ERROR`. And in this case, the value of the `buf_address` field is read as 0xffffffff, and the value of the `host_length` field is read as 0x0. Therefore, the subsequent Read Sequence is not required.



## DA16200 DA16600 Host Interface and AT Command

### Example

- ⌚ To write <ESC>S010,192.168.0.18,43310,abcde12345 command to the DA16200, the host sends:  
(0x50-0x08-0x02-0x60)-(0x80)-(0x00-0x00-0x24)-(<ESC>-'S'-'0'-'1'-'0'-'-'-'1'-'9'-'2'-'-'-'1'-'6'-'8'-'-'-'0'-'-'-'1'-'8'-'-'-'4'-'3'-'3'-'1'-'0'-'-'-'a'-'b'-'c'-'d'-'e'-'1'-'2'-'3'-'4'-'5'-0x00)
- ⌚ The read sequence after writing is the same as the example of B1~B2 below.  
Note that the payload data is transmitted MSB first and little-big endian system (see [Figure 10](#)).

**B1:** The Slave toggles high the interrupt line to inform Host when data is available.

**B2:** The Host reads the response message from Response Command address, and parses it using `struct _st_host_response`.

**B3:** The Host reads data from address (BUF\_ADDR) parsed from the response message.

There is a 200 ms timeout between reading the response after the interrupt occurs and reading the data after reading the response. If host requires more than 200 ms between each interval, change the timeout value accordingly.

An interval of several hundred microseconds is required between the “B2” and “B3” stages. If the interval between the two stages is too short, there is a possibility that two Interrupt Events are recognized as one. The interval differs depending on the type of application or CPU load. Roughly, when the CPU clock is 120 MHz, an interval of around 300 μs is required.

### Example

- ⌚ When the host becomes *high* on GPIO interrupt line from DA16200, the host sends:  
(0x50-0x08-0x02-0x58)-(0xC0)-(0x00-0x00-0x08), then read response from DA16200.  
Assume the buffer address from Slave is 0x12345678 for easy description and the data length to be sent from DA16200 is 8 bytes.
- ⌚ The read data should be 0x78-0x56-0x34-0x12-0x08-0x00-0x83-0x00.
- ⌚ Host sends: (0x12-0x34-0x56-0x78)-(0xC0)-(0x00-0x00-0x08), then read data from DA16200.  
Note that the read data is transmitted MSB first and little-big endian system (see [Figure 11](#)).

## 3.2.5 Header Format

### 3.2.5.1 Write Request (Host to Slave)

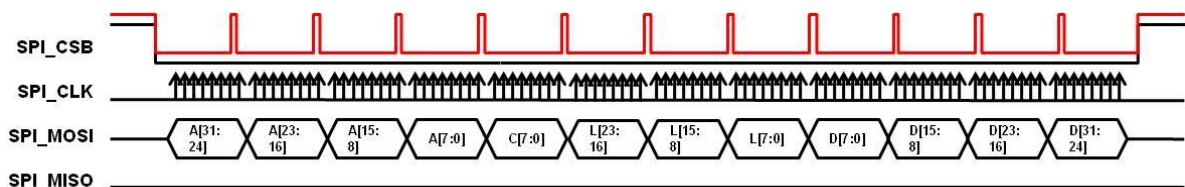


Figure 10: SPI Signals for Write Request



## DA16200 DA16600 Host Interface and AT Command

### 3.2.5.2 Read Response (Slave to Host)

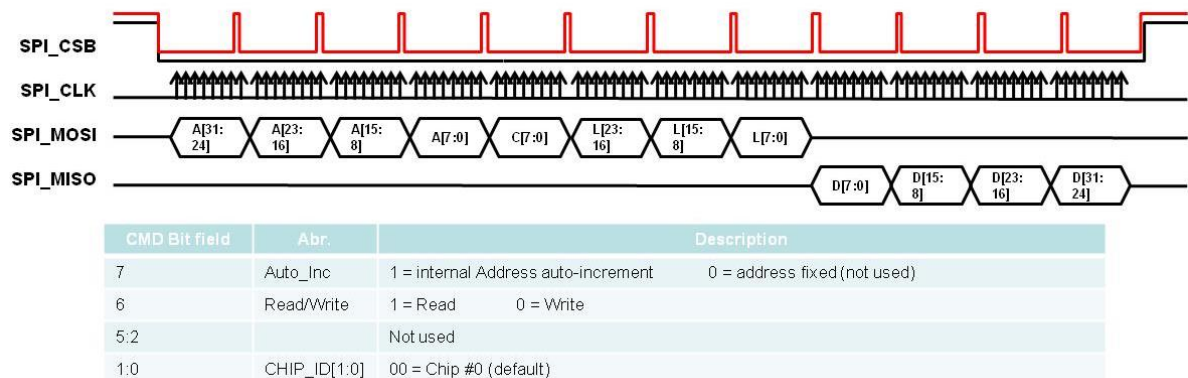


Figure 11: SPI Signals for Read Response

### 3.2.6 SPI Definition and Structures

Table 5: SPI Definition

```
#define HOST_MEM_WRITE_REQ      (0x80)
#define HOST_MEM_WRITE_RES      (0x81)
#define HOST_MEM_READ_REQ      (0x82)
#define HOST_MEM_READ_RES      (0x83)
#define FC9K_GEN_CMD_ADDR      (0x50080254) // Address to Write Command
#define FC9K_RESP_ADDR         (0x50080258) // Address to Read Response
#define FC9K_ATCMD_ADDR        (0x50080260) // Address to Send AT Command
```

Table 6: SPI Response Structure

```
typedef struct _st_host_response
{
    u32 buf_address;
    u16 host_length;
    u8  resp;
    u8  dummy;
} st_host_response;
```

Table 7: SPI Request Structure

```
typedef struct _st_host_request
{
    u16 host_write_length;
    u8  host_cmd;
```

## DA16200 DA16600 Host Interface and AT Command

```
u8 dummy;
} st_host_request;
```

### 3.3 SDIO Host Interface

#### 3.3.1 PIN MUX Configuration

SDIO slave is assigned to GPIOA[9:4] in DA16200. For interruption, the D1 port of SDIO can be used, but in some cases, GPIO can also be used.

However, there may be a pin mux initialization code in Renesas SDK that may look as follows:

```
• _dal6x_io_pinmux(PIN_CMUX, CMUX_GPIO); // For GPIOA 4,5
• _dal6x_io_pinmux(PIN_DMUX, DMUX_GPIO); // For GPIOA 6,7
• _dal6x_io_pinmux(PIN_EMUX, EMUX_GPIO); // For GPIOA 8,9
```

This means GPIOA[9:4] should be used as GPIOs, not SDIO slave. Therefore, the following code should be changed for SDIO slave at GPIOA[9:4]:

```
• _dal6x_io_pinmux(PIN_CMUX, CMUX_SDs); // For GPIOA 4,5
• _dal6x_io_pinmux(PIN_DMUX, DMUX_SDs); // For GPIOA 6,7
• _dal6x_io_pinmux(PIN_EMUX, EMUX_SDs); // For GPIOA 8,9
```

If SDIO D1 port is not used as Interrupt and GPIO is used as Interrupt, the following PAD Mux Setting is additionally required.

```
• _dal6x_io_pinmux(PIN_FMUX, FMUX_GPIO); // For GPIOA 10,11
```

**Table 8: PIN MUX Configuration - SDIO**

GPIO	Signal Name
GPIOA[4]	CMD
GPIOA[5]	CLK
GPIOA[6]	D3
GPIOA[7]	D2
GPIOA[8]	D1
GPIOA[9]	D0

#### 3.3.2 SDIO Protocol

##### 3.3.2.1 Message Format

The format of the messages sent/received to/from the external processor is the DA16200 protocol format over SDIO physical interface. The format and the parameters included are outlined in [Figure 12](#). For more information about SDIO header configuration, see SDIO Specification. When using SDIO protocol of DA16200, data should be aligned in units of 4-byte length.

## DA16200 DA16600 Host Interface and AT Command

### Basic Format

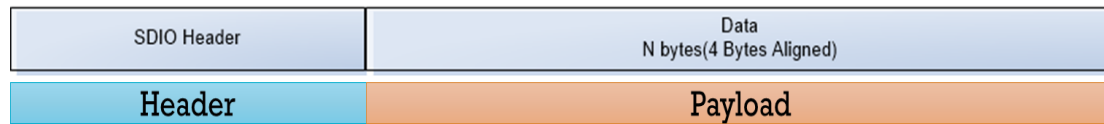


Figure 12: Basic Format

### 1. Address (included in the header)

The address list used by External Host is outlined in [Table 9](#).

Table 9: Address List

Address Type	Address
General Command (Write Request)	0x50080254
AT Command	Received from slave in initial stage
Response Command	0x50080258
Buffer Address	Received from slave in response message

### 2. Length (included in the header)

Payload length to follow.

### 3. Interrupt

According to SDIO Specification, Slave can cause Interrupt to Host by using D1 port. However, if the host cannot receive the interrupt using the D1 port, it must use the GPIO to generate the interrupt. The interrupt line used in the sequence diagram of this document means that the D1 port or GPIO is used.

#### 3.3.2.2 Write Sequence

Host to Slave write operations are performed in a three SDIO transactions.

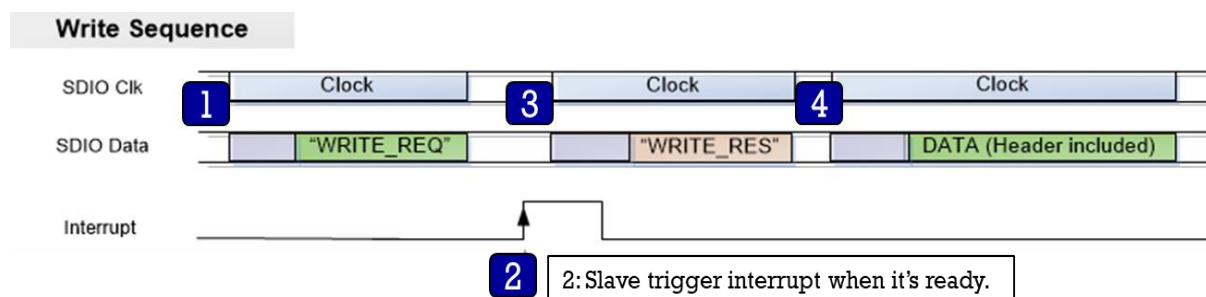
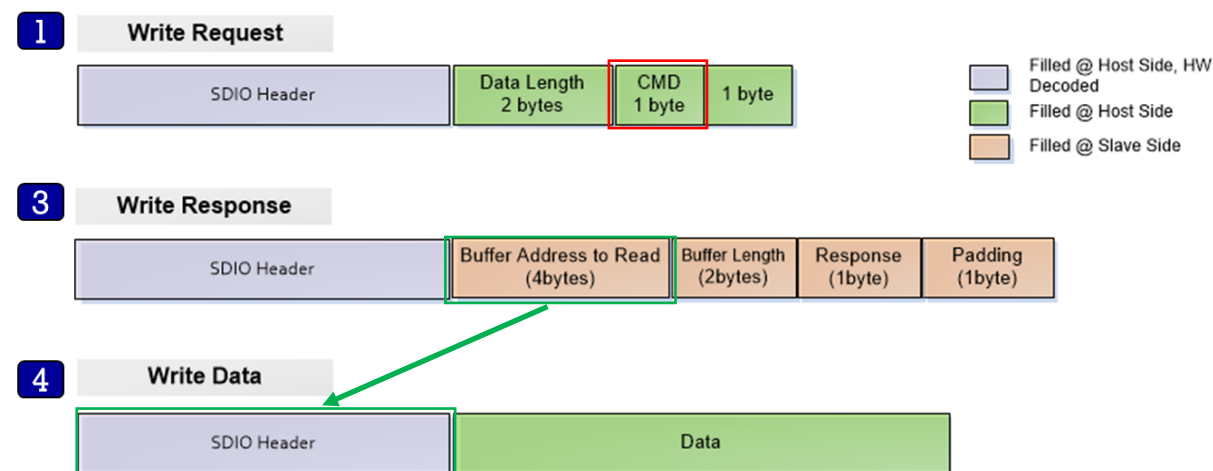


Figure 13: Write Sequence

## DA16200 DA16600 Host Interface and AT Command



**Figure 14: Structure**

1. The Host sends a `WRITE_REQ` command (0x80, red rectangle in Figure 3) to the General Command address (0x50080254).
2. The Host should wait for Interrupt from slave.
3. The Host reads the Write Response message by Response Command address (0x50080258) and parse it using `struct _st_host_response` (see Table 5).
4. The Host sends data to address (`BUFF_ADDR`) which is received from the Slave in the Write Response message (green rectangle in Figure 14).

There is a 200 ms timeout between reading the response after the interrupt occurs and reading the data after reading the response. If host requires more than 200 ms between each interval, change the timeout value accordingly.

An interval of several hundred microseconds is required between the “3” and “4” stages. If the interval between the two stages is too short, there is a possibility that two Interrupt Events are recognized as one. The interval differs depending on the type of application or CPU load. Roughly, when the CPU clock is 120 MHz, an interval of around 300  $\mu$ s is required.

### Example

When the host wants to write 8-byte data (0x88776655,0x44332211) to DA16200:

⌚ Host sends:

(SDIO Write-0x50080254, 4 bytes) - (0x08-0x00-0x80-0x00)

⌚ Host waits for interrupt triggering from DA16200.

⌚ Host sends:

(SDIO Read-0x50080258, 8 bytes) then read response from DA16200.

Assume the buffer address from Slave is 0x12345678 for easy description.

Then the read data should be 0x78-0x56-0x34-0x12-0x08-0x00-0x81-0x00.

⌚ Host sends:

(SDIO Write-0x12345678, 8 bytes)-( 0x55-0x66-0x77-0x88-0x11-0x22-0x33-0x44)

### 3.3.2.3 Read Sequence and Structure

Figure 15 shows a Slave device transmitting data to the Host when payload is available. This sequence is performed in a two SDIO transaction.

## DA16200 DA16600 Host Interface and AT Command

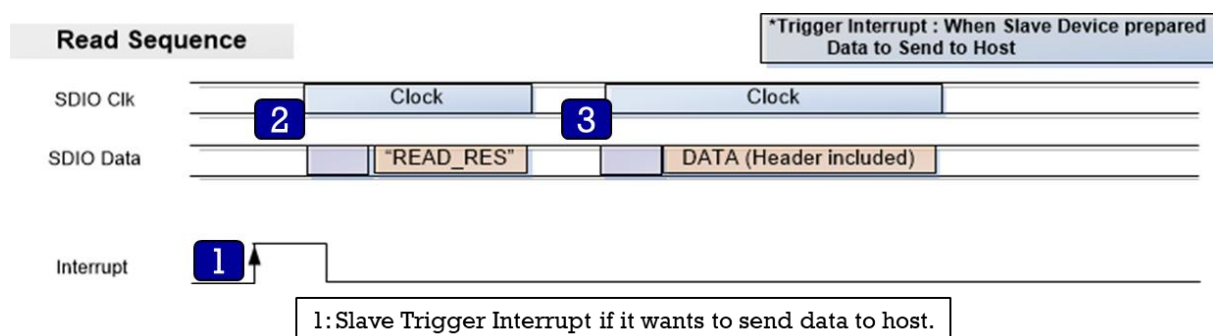


Figure 15: Read Sequence

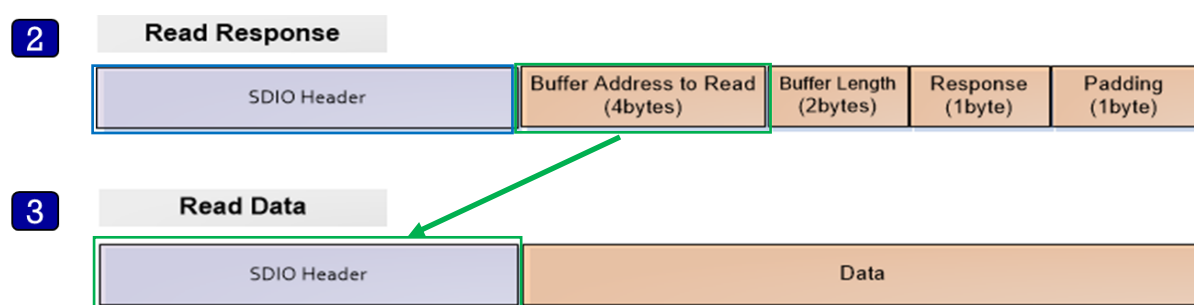


Figure 16: Structure

1. The Slave will trigger interrupt to inform the Host when data is available.
2. The Host reads the response message from Response Command address (0x50080258, blue rectangle in Figure 16), and parse it using `struct _st_host_response` (see Table 11).
3. The Host reads data from address (BUFF\_ADDR) which is received from Slave in the response message (green rectangle in Figure 16).

There is a 200 ms timeout between reading the response after the interrupt occurs and reading the data after reading the response. If host requires more than 200 ms between each interval, change the timeout value accordingly.

An interval of several hundred microseconds is required between the “2” and “3” stages. If the interval between the two stages is too short, there is a possibility that two Interrupt Events are recognized as one. The interval differs depending on the type of application or CPU load. Roughly, when the CPU clock is 120 MHz, an interval of around 300  $\mu$ s is required.

### Example

- ⌚ When the host received interrupt from DA16200,
- ⌚ Host sends:  
(SDIO Read-0x50080258, 8 bytes) then read response from DA16200.

Assume the buffer address from Slave is 0x12345678 for easy description and the data length to be sent from DA16200 is 8 bytes. The read data should be 0x78-0x56-0x34-0x12-0x08-0x00-0x83-0x00.

- ⌚ Host sends:  
(SDIO Read-0x12345678, 8 bytes) then read data from DA16200.

## DA16200 DA16600 Host Interface and AT Command

### 3.3.3 AT Command – Sequences and Structures

AT commands are instructions used to control a modem. AT is the abbreviation of Attention. Every command line starts with AT or at. Note that the starting *AT* is the prefix that informs the modem about the start of a command line. It is not part of the AT command name.

To use AT commands, read the address of AT (ESC) Command Buffer in the initial stage. Therefore, read the value of address 0x50080264 after SDIO is initialized, and write the command to that address when sending AT command afterwards.

Figure 17 illustrates how to use the AT command through SDIO in DA16200. This is because AT command uses a predetermined address and the maximum size of data is defined.

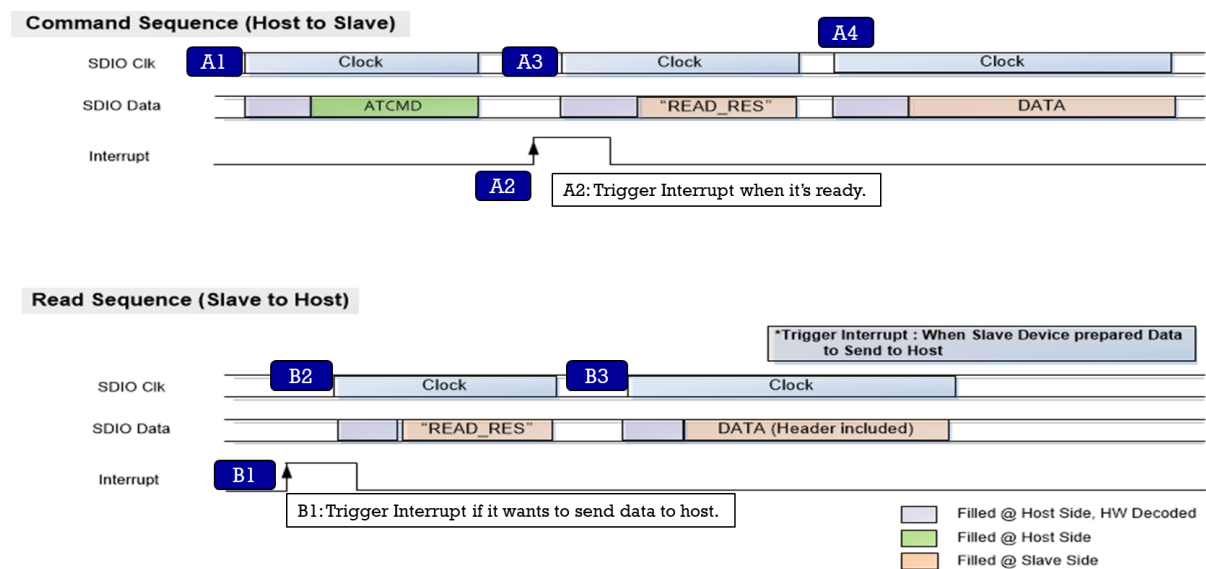


Figure 17: AT Command Sequence

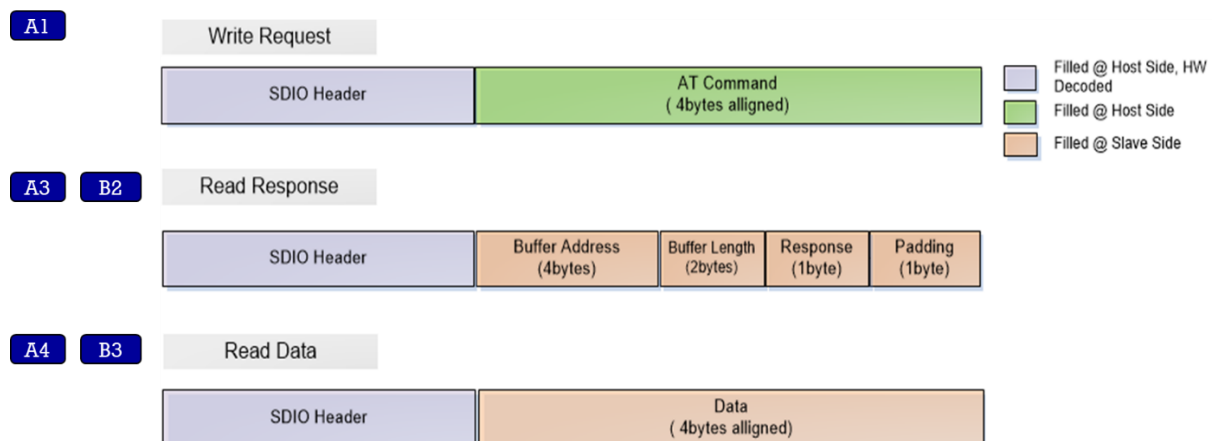


Figure 18: Structure

Descriptions of Figure 17 and Figure 18 are as follows.

A1: The Host sends an AT or ESC command to AT Command address.

A2: The Host waits for interrupt trigger.

## DA16200 DA16600 Host Interface and AT Command

A3: The Host reads the response message from address and parses it using `struct _st_host_response`.

A4: The Host reads OK, Error or data from address (BUF\_ADDR), depending on the type of command.

B1: The Slave will toggle high the interrupt line to inform Host when data is available.

B2: The Host reads the response message from Response Command address and parses it using `struct _st_host_response`.

B3: The Host reads data from address (BUF\_ADDR) parsed from the response message.

There is a 200 ms timeout between reading the response after the interrupt occurs and reading the data after reading the response. If host requires more than 200 ms between each interval, change the timeout value accordingly.

An interval of several hundred microseconds is required between the “A3” and “A4” stages, “B2” and “B3” stages. If the interval between the two stages is too short, there is a possibility that two Interrupt Events are recognized as one. The interval differs depending on the type of application or CPU load. Roughly, when the CPU clock is 120 MHz, an interval of around 300  $\mu$ s is required.

### 3.3.4 ESC Command – Sequences and Structures

To use ESC commands, read the address of AT (ESC) Command Buffer in the initial stage. Therefore, read the value of address 0x50080264 after SDIO is initialized, and write the command to that address when sending AT command afterwards.

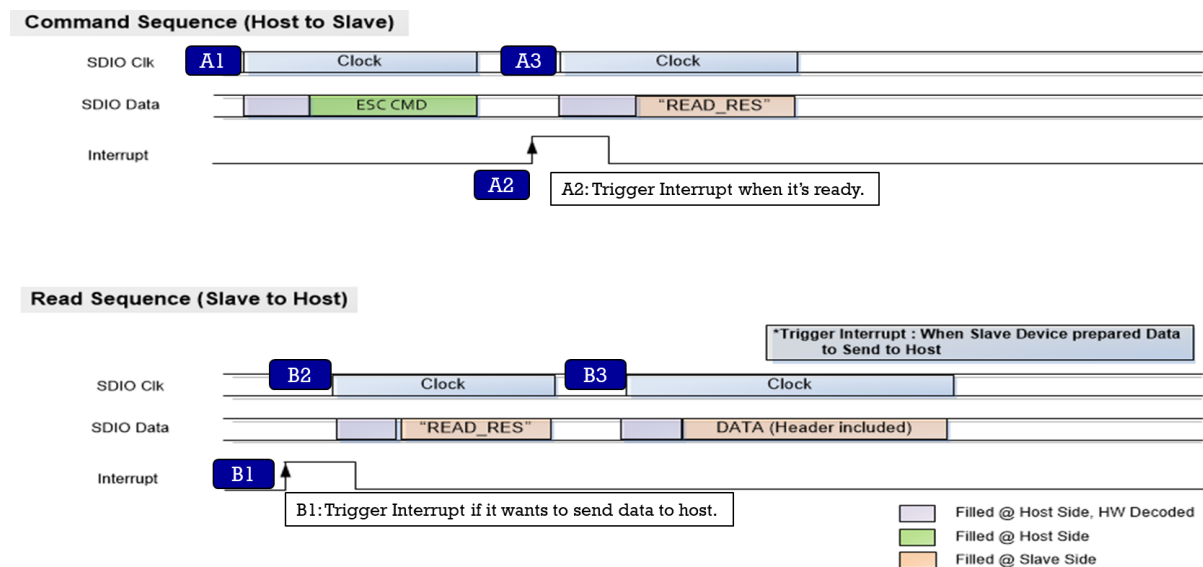


Figure 19: ESC Command Sequence

## DA16200 DA16600 Host Interface and AT Command

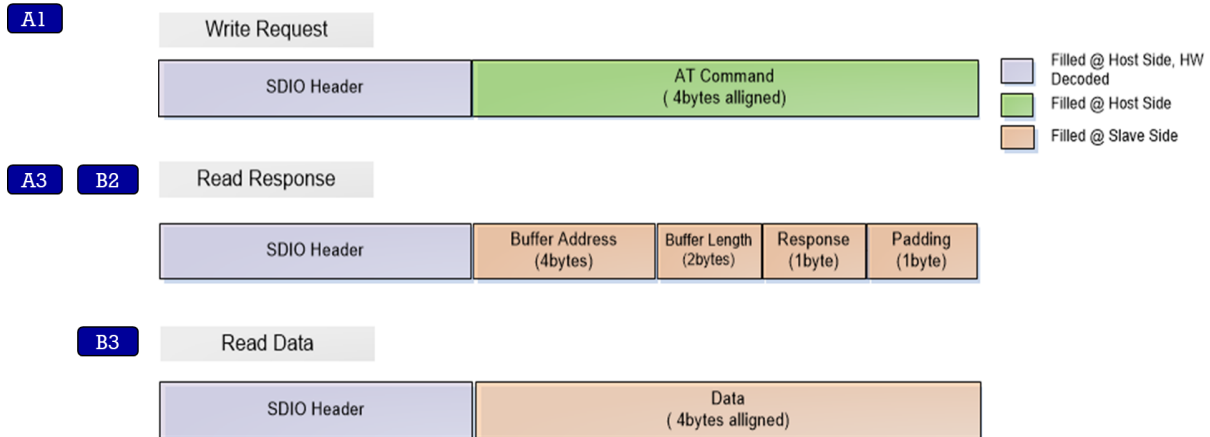


Figure 20: Structure

Description of Figure 19 and Figure 20 are as follows.

A1: The Host sends an ESC command to AT (ESC) Command address.

A2: The Host waits for interrupt trigger.

A3: The Host reads the response message from address and parses it using `struct _st_host_response`. The result for ESC command is sent to the host in the response field of `struct _st_host_response`. The response field is a 1-byte decimal value. A value of 0x20 is a result of OK. All other values are an ERROR. And in this case, the value of the `buf_address` field is read as 0xffffffff, and the value of the `host_length` field is read as 0x0. Therefore, the subsequent Read Sequence is not required.

B1: The Slave will toggle high the interrupt line to inform Host when data is available.

B2: The Host reads the response message from Response Command address and parses it using `struct _st_host_response`.

B3: The Host reads data from address (BUF\_ADDR) parsed from the response message.

There is a 200 ms timeout between reading the response after the interrupt occurs and reading the data after reading the response. If host requires more than 200 ms between each interval, change the timeout value accordingly.

An interval of several hundred microseconds is required between the “B2” and “B3” stages. If the interval between the two stages is too short, there is a possibility that two Interrupt Events are recognized as one. The interval differs depending on the type of application or CPU load. Roughly, when the CPU clock is 120 MHz, an interval of around 300  $\mu$ s is required.

### 3.3.5 SDIO Definition and Structures for Implementation

Table 10: SDIO Definition

#define	HOST_MEM_WRITE_REQ	(0x80)	
#define	HOST_MEM_WRITE_RES	(0x81)	
#define	HOST_MEM_READ_REQ	(0x82)	
#define	HOST_MEM_READ_RES	(0x83)	
#define	FC9K_GEN_CMD_ADDR	(0x50080254)	// Address to Write Command



## DA16200 DA16600 Host Interface and AT Command

#define	FC9K_RESP_ADDR	(0x50080258)	// Address to Read Response
---------	----------------	--------------	-----------------------------

**Table 11: SDIO Response Structure**

```
typedef struct _st_host_response
{
    u32 buf_address;
    u16 host_length;
    u8  resp;
    u8  dummy;
} st_host_response;
```

**Table 12: SDIO Request Structure**

```
typedef struct _st_host_request
{
    u16 host_write_length;
    u8  host_cmd;
    u8  dummy;
} st_host_request;
```

## DA16200 DA16600 Host Interface and AT Command

### 4 AT Commands

#### 4.1 How to Include AT Command Feature in SDK

##### 4.1.1 How to Run AT-CMD on UART

This section describes how to include AT command feature in SDK.

In SDK, open the file

~/SDK/apps/da16x00/get\_started/include/user\_main/config\_generic\_sdk.h using the editor tool and search the string `#undef __SUPPORT_ATCMD__`.

To enable AT Command feature in SDK, change `#undef` to `#define` and save the file. Then, rebuild the SDK package and new generated image will be work as AT-CMD module.

```
//-----
// AT-CMD Features
//-----

//
// Enable/Disable AT-CMD module
//
// When enable this feature,
// more detail features are support below sub-features.
// User can check all AT-CMDs in ~/core/system/src/at_cmd/atcmd.c
//
#undef __SUPPORT_ATCMD__
```

For AT Command module, default interface type is UART1 as shown below. If a user wants to use UART2, change `#undef __ATCMD_IF_UART2__` to `#define __ATCMD_IF_UART2__`.

```
#if defined ( __SUPPORT_ATCMD__ )

//
// Default interface of DA16200 EVK is UART1.
// User can change host-interface kind to change as below four-types one.
//
#define __ATCMD_IF_UART1__ // AT-CMD over UART1
#undef __ATCMD_IF_UART2__ // AT-CMD over UART2
#undef __ATCMD_IF_SPI__ // AT-CMD over SPI
#undef __ATCMD_IF_SDIO__ // AT-CMD over SDIO
```

##### 4.1.2 How to Run AT-CMD on SPI

AT-CMD is configured to use the UART1 interface by default and can be configured to use the SPI interface. To enable the AT-CMD over SPI interface, modify `config_generic_sdk.h` as shown in bold below.

## DA16200 DA16600 Host Interface and AT Command

```

...
// AT-CMD service
#define __SUPPORT_ATCMD__
...
#if defined ( __SUPPORT_ATCMD__ )
    #undef __ATCMD_IF_UART1__ // AT-CMD over UART1
    #undef __ATCMD_IF_UART2__ // AT-CMD over UART2
    ...
    #undef __USER_UART_CONFIG__ // Support Customer's UART configuration
    #define __ATCMD_IF_SPI__ // AT-CMD over SPI
    #undef __ATCMD_IF_SDIO__ // AT-CMD over SDIO
#endif /* __SUPPORT_ATCMD__ */

```

To configure and use the AT-CMD over SPI interface, see Section 3.2.

### 4.1.3 How to Run AT-CMD on SDIO

AT-CMD can also be configured to use the SDIO interface. To enable the AT-CMD over SDIO interface, modify config\_generic\_sdk.h as shown in bold below.

```

...
// AT-CMD service
#define __SUPPORT_ATCMD__
...
#if defined ( __SUPPORT_ATCMD__ )
    #undef __ATCMD_IF_UART1__ // AT-CMD over UART1
    #undef __ATCMD_IF_UART2__ // AT-CMD over UART2
    ...
    #undef __USER_UART_CONFIG__ // Support Customer's UART configuration
    #undef __ATCMD_IF_SPI__ // AT-CMD over SPI
    #define __ATCMD_IF_SDIO__ // AT-CMD over SDIO
#endif /* __SUPPORT_ATCMD__ */

```

To configure the AT-CMD over SDIO interface, see Section 3.3.

#### 4.1.3.1 Example Sequence for SDIO interface

An example of the sequence used to initiate a command through the SDIO interface is shown in Figure 21.

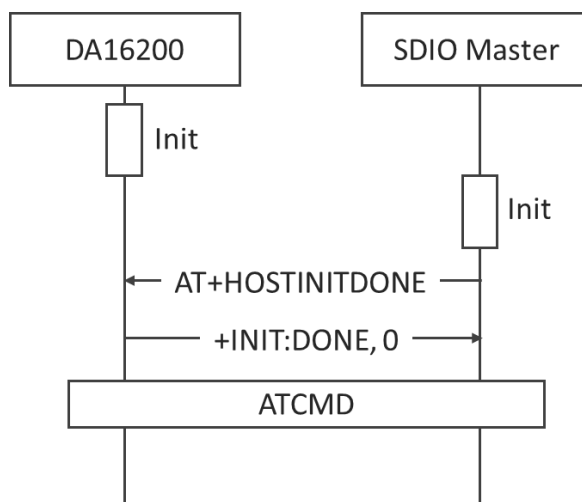


Figure 21: Example Sequence to Initiate AT-CMD through SDIO Interface

## DA16200 DA16600 Host Interface and AT Command

When using the SDIO interface, both the DA16200 and the SDIO master devices must be initialized before initiating an AT-CMD. The SDIO master device must send AT+HOSTINITDONE immediately after initialization is completed.

### NOTE

For details on how to use the SDK package, see Ref. [2].

## 4.2 AT Command Format

### 4.2.1 Basic Command Format

#### 4.2.1.1 Write CMD

Basic command execution.

ATXX

For example: ATZ

OK

#### 4.2.1.2 Read CMD

Get the parameter values of the command.

ATXX=?

For example: ATQ=?

Display result on

OK

### 4.2.2 Extended Command Format

#### 4.2.2.1 Write CMD

Extended command execution.

AT+XXX=<param1>,<param2>,<param3>,<param4>...<paramN>

For example: AT+NWIP=0,172.16.0.100,255.255.255.0,172.16.0.1

OK

If the SSID contains a comma or single quote, the SSID must be enclosed in single quotes.

For example:

```
SSID = MY,SSID'CS
sec = 4
idx = 2
Password = N12345678
```

Is encoded as:

```
AT+WFJAP='MY,SSID'CS',4,2,N12345678'
OK
```

### NOTE

The use of a single quote followed by a comma in a parameter is prohibited.  
For example: AT+WFJAP='MY,SSID',CS',4,2,N12345678 is invalid.

## DA16200 DA16600 Host Interface and AT Command

### 4.2.2.2 Read CMD

Get the parameter values of the command.

AT+XXX=?

For example: AT+NWIP=?

+ANIP:172.16.0.17,255.255.255.0,172.16.0.1

OK

#### NOTE

Not all commands support the AT+XXX=? query function such as AT+RESTART and ATF. Check the command table for the valid operation of each command.

### 4.2.3 Response Format

#### 4.2.3.1 Start-up Response

The AT command response when DA16200/DA16600 is reset.

<CR><LF>+INIT:DONE,<mode><CR><LF>

The AT command response when DA16200/DA16600 wakes up from DPM sleep.

#### 4.2.3.2 Basic Response

Basic response gives the command result and is accompanied by a carriage return and a line feed.

<CR><LF>+INIT:WAKEUP,<type><CR><LF>

#### 4.2.3.3 Normal Response

<CR><LF>OK<CR><LF>

#### 4.2.3.4 Error Response

<CR><LF>ERROR:<error code><CR><LF>

#### 4.2.3.5 Extended Response

Extended response gives the command setting values and is followed by a basic response

<CR><LF>+XXX:[value1],[value2],...

<CR><LF>OK<CR><LF>

#### NOTE

When an MCU (AT-CMD Host) waits for a response of a command (for those commands that give extended response as well) to take the next action, it should wait for both *normal* response (**OK** or **ERROR**) and *extended* response (also known as **Operation Result**).

- Error response codes: See [Appendix I](#).

#### NOTE

There are major changes in Error response code in SDK v3.2.5.0 and later versions. The examples in this document have updated based on the changes.

## DA16200 DA16600 Host Interface and AT Command

### 5 Function AT Commands

#### 5.1 Basic Function Commands

Table 13: Basic Function Command List

Command	Parameters	Description
?	(none)	Show AT command usage
	<p>Example</p> <pre> ? AT Commands: ? - No example for ? HELP=&lt;command&gt; - Print help message AT - Attention command AT+ - List available commands ATZ - AT command initialize ATF - Restore to Factory mode (NVRAM clean) ATE - Command echo ATQ - Result Codes On/Off AT+RESTART - System Restart  --- Middle omission ---  AT+TRSAVE - Save current status of all session  === User AT-CMD =====  OK </pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> </ul>	
help	<cmd_name>	AT command name to query the use of commands
	(none)	Same as the “?” command

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre> HELP AT Commands:   ?     - No example for ? HELP=&lt;command&gt;   - Print help message AT   - Attention command AT+   - List available commands ATZ   - AT command initialize ATF   - Restore to Factory mode (NVRAM clean) ATE   - Command echo ATQ   - Result Codes On/Off AT+RESTART   - System Restart ... --- Middle omission ---  OK  HELP=ATE ATE   - Command echo OK </pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> </ul>	
AT+	(none)	Show AT command list

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre> AT+ AT AT+ ATZ ATF ATE ATQ AT+RESTART  --- Middle omission ---  AT+TRSAVE OK </pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> </ul>	
ATZ	(none)	Initialize AT command
	<p>Example</p> <pre> ATZ  Display result on Echo off OK </pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> </ul>	
ATF	(none)	DA16200/DA16600 factory reset Response: "+INIT:DONE,0"
	<p>Example</p> <pre> ATF  +INIT:DONE,0 </pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>All NVRAM parameters that include Wi-Fi profile (Soft AP or STA) settings are removed, DUT restarts, and "+INIT:DONE,0" will be received</li> </ul>	
ATE	(none)	ECHO on/off
	?	Show Echo status - on/off



## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre> ATE  Echo on OK  ATE  Echo off OK  ATE=?  Echo on OK </pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> </ul>	
ATQ	(none)	Turn on/off whether to display result code
	?	Show the current status of result code being displayed or not
	<p>Example</p> <pre> ATQ Display results off  ATQ=? Display result on OK </pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> </ul>	
ATB	<baudrate> [[,<databits>] [,<parity>] [,<stopbits>]	Set UART parameters (the main purpose is to change baud rate)  <baudrate>: 9600/19200/38400/57600/115200/230400/460800/921600 <databits>: [optional], 5/6/7/8 (Default) <parity>: [optional], n (None, Default)/e (Even)/o (Odd) <stopbits>: [optional], 1 (Default)/2
	?	Show the current baud rate
	<p>Example</p> <pre> ATB=230400 OK </pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>If __USER_UART_CONFIG__ is enabled in SDK (See <a href="#">Appendix C</a>), this command will be disabled</li> </ul>	

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
AT+RESTART	(none)	System restart
	<p>Example</p> <pre>AT+RESTART OK  +INIT:DONE,0</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> </ul>	
AT+RESET	(none)	System reset Go to the Boot mode ([MROM] prompt)
	<p>Example</p> <pre>AT+RESET OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>Once the system goes into MROM mode, AT command is not available. Therefore, MCU needs to force POR booting or enter 'boot' command via UART0 console</li> </ul>	
AT+CHIPNAME	(none)	Get chip name, DA16200 or DA16600
	<p>Example</p> <pre>AT+CHIPNAME +CHIPNAME:DA16200 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> </ul>	
AT+VER	(none)	Get version information Response: +VER:<main version>
	<p>Example</p> <pre>AT+VER +VER:FRTOS-GEN01-01-xxxxxxxxxx-xxxxxx OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> </ul>	
AT+SDKVER	(none)	<p>Get the SDK version information</p> <p>Response:</p> <pre>+SDKVER:&lt;major&gt;.&lt;minor&gt;.&lt;revision&gt;.&lt;eng_number&gt; &lt;major&gt;: SDK major number &lt;minor&gt;: SDK minor number &lt;revision&gt;: SDK Revision number &lt;eng_number&gt;: SDK engineering number</pre>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	Example AT+SDKVER +SDKVER:3.2.8.0 OK  Note <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> </ul>	
AT+TIME	<date>,<time>	Set the current time <date>: yyyy-mm-dd <time>: hh:mm:ss Response: OK or ERROR
	?	Get the current time Response: +TIME:<yyyy-mm-dd> <hh:mm:ss>
	Example AT+TIME=2021-07-15,16:14:30 OK  AT+TIME=? +TIME:2021-07-15,16:14:32 OK  Note <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> </ul>	
AT+RLT	(none)	Get system running time Response: +RLT:<days>,<hh:mm:ss>
	Example AT+RLT +RLT:0,01:06.18 OK  Note <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> </ul>	
AT+TZONE	<sec>	GMT time zone setting (-43200 ~ 43200) <sec>: Time zone setting parameter Response: OK or ERROR
	?	Get GMT time zone parameter Response: +TZONE:<sec>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre>AT+TZONE=? +TZONE:0 OK  AT+TZONE=32400 OK  AT+TZONE=? +TZONE:32400 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>The &lt;sec&gt; parameter must be a multiple of 60 seconds. If the value for &lt;sec&gt; is not a multiple of 60 seconds, then the remainder will be discarded</li> </ul>	
AT+DEFAP	(none)	<p>All profiles in NVRAM are removed and set up in Soft AP mode with the default configuration. To initialize the Soft AP interface, the system will reboot automatically.</p> <p>Response: OK or ERROR (reboot)</p>
	<p>Example</p> <pre>AT+DEFAP OK  +INIT:DONE,1</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>Default configuration: <ul style="list-style-type: none"> <li>SSID: DA16200/DA16600_XXXXXX (for example, 9FFCF3: the last three hexadecimal values of the board's MAC address)</li> <li>Authentication: WPA2/CCMP</li> <li>IP address: 10.0.0.1</li> <li>Netmask: 255.255.255.0</li> <li>Gateway: 10.0.0.1</li> <li>PSK: 12345678</li> <li>DHCP server started <ul style="list-style-type: none"> <li>DHCP range: 10.0.0.2 ~ 10.0.0.11</li> <li>DHCP DNS: 8.8.8.8</li> </ul> </li> </ul> </li> <li>To query the configuration status, AT+WFSAP and/or AT+NWDHR can be used</li> </ul>	
AT+BIDX	<idx>	<p>Set Boot index</p> <p>&lt;idx&gt;: Boot index (0 or 1)</p> <p>Response: OK or ERROR</p>
	?	<p>Get the current Boot index</p> <p>Response: +BIDX:&lt;0 1&gt;</p>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre>AT+BIDX=? +BIDX:0 OK  AT+BIDX=1 OK  AT+BIDX=? +BIDX:1 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>System restart is required for changes to take effect. "AT+RESTART" command can be used to restart the system</li> </ul>	
AT+DPM	<dpm> [,<nvm_only>]	Set DPM on/off. System restart is required for DPM mode (On/Off) to take effect. <dpm>: 0 (Off), 1 (On) <nvm_only>: 1 (write dpm mode to nvram only, and not reboot), 0 or not specified (change dpm mode and reboot) Response: OK or ERROR
	?	Get the current DPM setting Response: +DPM:<0 1>
	<p>Prerequisite</p> <p>Station mode</p> <p>Example</p> <pre>AT+DPM=? +DPM:0 OK  AT+DPM=1 ; DPM enabled and system reboots automatically OK  +INIT:DONE,0  AT+DPM=1,1 ; DPM enabled without system reboots OK  AT+DPM=? +DPM:1 OK</pre> <p>Note</p>	

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
		<ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>DPM configuration is stored in NVRAM</li> <li>DA16200/DA16600 is restarted if the “&lt;nvrn_only&gt;” parameter is zero or not specified and AT command response is <b>OK</b> <ul style="list-style-type: none"> <li>+INIT:DONE,0 message is sent when DA16200/DA16600 boots up</li> <li>If the usage of the AT command is not valid, then DA16200/DA16600 sends an <b>ERROR</b> message without restarting</li> </ul> </li> <li>If the “nvrn_only” parameter is “1”, then restart the system manually using “AT+RESTART”</li> <li>When DA16200/DA16600 reboots, DA16200/DA16600 tries to connect to the AP if the Wi-Fi connection information is available in the NVRAM <ul style="list-style-type: none"> <li>+WFJAP:0,&lt;reason&gt; or +WFJAP:1,'&lt;SSID&gt;,&lt;IP Address&gt;' as result of Wi-Fi connection</li> <li>If Wi-Fi connection fails during boot-up due to some unexpected condition (for example, AP is offline, temporary communication issue with AP, and wrong password is stored), +WFJAP:x may not be sent immediately and takes some time, in which case, wait until +WFJAP:x is received. When a timeout occurs, depending on the application use case, either cancel the connection trial (AT+WFQAP) or retry the connection with the right info (AT+WFJAP/AT+WFJAPA)</li> </ul> </li> <li>If MQTT is configured, DA16200/DA16600 tries to connect to the MQTT broker after a Wi-Fi connection is established. The Operation Result – +NWMQCL:0 or +NWMQCL:1 – is sent over UART1 as a result</li> <li>DA16200/DA16600 operates DPM if it is set to 1 (<b>TRUE</b>) <ul style="list-style-type: none"> <li>If Wi-Fi connection is NOT established in DPM mode, DA16200/DA16600 enters an abnormal DPM operation</li> <li><b>Abnormal DPM operation:</b> While DA16200/DA16600 operates in DPM sleep, DA16200/DA16600 executes an Abnormal DPM operation if DA16200/DA16600 is in a “disconnected” state with the specified AP for some reason Abnormal DPM works as follows: DA16200/DA16600 enters sleep with a predefined RTC timer (This is called Abnormal DPM RTC Timer) configured if the AP connection fails. If DA16200/DA16600 is woken up by the Abnormal DPM RTC timer, DA16200/DA16600 tries to connect to the specified AP within a predefined period and sleeps again for a predefined time. The DA16200 library provides default predefined values for Abnormal DPM, but users can modify the relevant parameters based on their application use case scenarios</li> <li>If the Wi-Fi connection is established but MQTT connection is NOT established (if MQTT is enabled), DA16200/DA16600 tries to connect to the MQTT broker several times and enters DPM Sleep based on MQTT's abnormal DPM operation</li> </ul> </li> </ul>
AT+DPMKA	<period>	Set DPM keepalive period <period>: Keepalive period (millisecond, 0 ~ 600000) Response: OK or ERROR
	?	Get DPM keepalive period
	(none)	Response: +DPMKA=<millisecond>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre>AT+DPMKA +DPMKA:30000 OK  AT+DPMKA=5000 OK  AT+DPMKA=? +DPMKA:5000 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• The configuration is stored in NVRAM</li> <li>• System restart is required for changes to take effect</li> </ul>	
AT+DPMTIMWU	<count>	Set DPM TIM wake-up count <count>: TIM wake-up count (1 ~ 6000) Response: OK or ERROR
	?	Get DPM TIM wake-up count
	(none)	Response: +DPMTIMWU=<count>
	<p>Example</p> <pre>AT+DPMTIMWU +DPMTIMWU:10 OK  AT+DPMTIMWU=20 OK  AT+DPMTIMWU=? +DPMTIMWU:20 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• The configuration is stored in NVRAM</li> <li>• System restart is required for changes to take effect</li> </ul>	
AT+DPMUSERWU	<time>	Set DPM user wake-up time <time>: User wake-up period (millisecond, 0 ~ 86400000) Response: OK or ERROR
	?	Get DPM user wake-up time
	(none)	Response: + DPMUSERWU =<millisecond>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre>AT+DPMUSERWU +DPMUSERWU:0 OK  AT+DPMUSERWU=300 OK  AT+DPMUSERWU=? +DPMUSERWU:300 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• The configuration is stored in NVRAM</li> <li>• System restart is required for changes to take effect</li> </ul>	
AT+CLRDPM_SLP_EXT	(none)	Set the user application not to enter DPM sleep Response: OK or ERROR
	<p>Prerequisite</p> <p>DPM enabled</p> <p>Example</p> <pre>AT+CLRDPM_SLP_EXT OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• A host should execute this command within 200 ms after waking up the DA16200/DA16600 through the external wake-up pin, otherwise, DA16200/DA16600 will go into DPM sleep</li> </ul>	
AT+SETDPM_SLP_EXT	(none)	Set the user application ready to enter DPM sleep Response: OK or ERROR
	<p>Prerequisite</p> <p>DPM enabled</p> <p>Example</p> <pre>AT+SETDPM_SLP_EXT OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• If DA16200/DA16600 is woken up by an external wake-up signal and the "AT+CLRDPM_SLP_EXT" command is executed, this command should be issued once every job is done. If this command is not run after the job is done, DA16200/DA16600 will not enter DPM sleep</li> </ul>	



## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
AT+SETSLEEP2EXT	<period>,<use_retention_memory>	Enter Sleep 2 mode for the period specified. <period>: wake-up timeout, in millisecond. Max. period: 2097151000 (about 24 days) <use_retention_memory>: 1 (retain), 0 (not retain) Response: OK or ERROR
	<p>Example</p> <pre>AT+SETSLEEP2EXT=10000,0 OK +INIT:DONE,0</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>DA16200/DA16600 can be woken up by RTC_WAKE_UP while in sleep by AT+SETSLEEP2EXT</li> <li>DA16200/DA16600 sends "+INIT:DONE,0" when it wakes up</li> <li>A value of 0 for the &lt;period&gt; parameter sets the system to wakeup only when an RTC_WAKE_UP event occurs</li> <li>This command should be run in Non-DPM mode only, therefore, if you want to run this command in DPM mode, disable DPM first (AT+DPM=0,1), and run this command. When this command is run in DPM mode enabled, it returns ERROR (-316)</li> <li>The use of 1 as &lt;use_retention_memory&gt; is obsolete. If you want to use 1 as &lt;use_retention_memory&gt;, use AT+SETSLEEP3EXT command instead</li> </ul>	
AT+SETSLEEP3EXT	<period>	Enter Sleep 3 mode for the period specified <period>: wake-up timeout, in millisecond. Max. period: 2097151000 (about 24 days)
	<p>Example</p> <pre>AT+SETSLEEP3EXT=10000 OK +INIT:DONE,0</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>Retention memory is ON during Sleep3 mode</li> <li>DA16200/DA16600 can be woken up by RTC_WAKE_UP while in sleep by AT+SETSLEEP3EXT</li> <li>DA16200/DA16600 sends "+INIT:DONE,0" or "+INIT:WAKEUP,..." when it wakes up</li> <li>A value of 0 for the &lt;period&gt; parameter sets the system to wakeup only when an RTC_WAKE_UP event occurs</li> <li>This command can be used in DPM mode or Non-DPM mode</li> </ul>	
AT+SETSLEEP1EXT	Deprecated	
	<retain_dpm_memory>	Enter DPM Sleep 2 mode <retain_dpm_memory>: 1 (retain), 0 (not retain) Response: OK or ERROR

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre>AT+SETSLEEP1EXT=1 OK +INIT:DONE,0</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• This command is the same as AT+SETSLEEP2EXT with the period set to 0</li> <li>• It recommends to use the AT+SETSLEEP2EXT command instead of this one</li> <li>• Enabled by default in the SDK</li> <li>• DA16200/DA16600 can only be woken up by RTC_WAKE_UP or GPIO which has been assigned as a wake-up source</li> <li>• DA16200/DA16600 sends "+INIT:DONE:0" once it wakes up</li> </ul>	
AT+GETFASTCONN	(none)	Get the Wi-Fi Fast-reconnection mode status value Response: +GETFASTCONN:<0 1>
	<p>Example</p> <pre>AT+GETFASTCONN +GETFASTCONN:0 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK v3.2.3.0 or later</li> <li>• See <a href="#">Appendix G</a> for "Wi-Fi Fast-reconnect" for the DA16200/DA16600</li> </ul>	
AT+SETFASTCONN	<flag>	Enable/Disable the Wi-Fi Fast-reconnection mode <mode>: 0 (Disable), 1 (Enable) Response: OK or ERROR
	<p>Example</p> <pre>AT+SETFASTCONN=1 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK v3.2.3.0 or later</li> <li>• See <a href="#">Appendix G</a> for "Wi-Fi Fast-reconnect" for the DA16200/DA16600</li> </ul>	
AT+MCUWUDONE	(none)	Notify that the MCU wakes up completely. When this command is received, DA16200/DA16600 starts to send messages to the MCU (that is, MCU should send this command immediately after executing "External wakeup") Response: OK or ERROR

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre>AT+MCUWUDONE OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• When DA16200/DA16600 receives the command, it starts to send messages to the MCU</li> <li>• MCU should send this command immediately when it receives a notification like "+INIT:WAKEUP,UC"</li> <li>• If the "__DPM_TEST_WITHOUT_MCU__" is defined, then MCU does not need to send this command which means it is assumed that MCU is always ready to read a message(response) from DA16200/DA16600</li> </ul>	
AT+HOSTINITDONE	(none)	<p>Notify the DA16200 that the MCU has completed initialization (For SDIO interface, the MCU must send this command immediately after initialization.). The DA16200 returns its initialization status as a response. See <a href="#">Table 14</a>.</p> <p>Response: +INIT:DONE,&lt;mode&gt; or +INIT:WAKEUP,&lt;type&gt;)</p>
	<p>Example</p> <pre>AT+HOSTINITDONE +INIT:DONE,0</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> </ul>	
AT+DPMABNWFCNT	<count>	<p>Set Wi-Fi Connection Retry counts until System enters DPM Abnormal Sleep</p> <p>&lt;count&gt;: 0 (This feature not used. DPM Abnormal sleep scheme is followed), 1 to 6 (Wi-Fi Connection Retry count)</p> <p>Response: OK or ERROR</p>
	?	<p>Get the current DPM Abnormal Wi-Fi Connection Retry counts set</p> <p>Response: +DPMABNWFCNT:&lt;count&gt;</p>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <p>; If Wi-Fi connection trials are not successful two times in a row, the system goes to DPM Abnormal sleep</p> <p>AT+DPMABNWFCNT=2 OK</p> <p>AT+DPMABNWFCNT=? +DPMABNWFCNT:2 OK</p> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be disabled by default in the SDK</li> <li>• If <code>__WF_CONN_RETRY_CNT_ABN_DPM__</code> is enabled in the SDK (<code>config_generic_sdk.h</code>), this command will be enabled</li> <li>• The configuration is stored in NVRAM</li> <li>• If the cause of the Wi-Fi connection failure is "Wrong password" input, and if the application wants to cancel the auto-reconnect trial right away, <code>__WIFI_CONN_RETRY_STOP_AT_WK_CONN_FAIL__</code> should be defined in <code>sys_common_features.h</code></li> </ul>	

**Table 14: Initiation Response List**

Response	Parameters	Description
+INIT	DONE,<mode>	DA16200/DA16600 booting is complete: <mode>:0 (STA), 1 (Soft AP) For example: +INIT:DONE,0
	WAKEUP,<type>	DA16200/DA16600 wake-up is complete from DPM SLEEP state <type> wake-up type <ul style="list-style-type: none"> <li>• UC: Unicast packet received</li> <li>• NOBCN: No beacon from the connected AP</li> <li>• DEAUTH: Disconnected from the connected AP</li> <li>• EXT: External wakeup</li> <li>• RTC: By a timer registered</li> </ul> For example: +INIT:WAKEUP,UC

## 5.2 Network Function Commands

**Table 15: Network Function Command List**

Command	Parameters	Description
AT+NWIP	<iface>,<ip_addr>,<netmask>,<gw>	Set the IP address <iface>: WLAN interface. 0 (WLAN0, STA), 1 (WLAN1, Soft AP) <ip_addr>: IP Address <netmask>: Subnet mask <gw>: Gateway Response: OK or ERROR
	?	Get the IP address of the current WLAN interface

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	(none)	Response: +NWIP: <iface>,<ip_addr>,<netmask>,<gw>
	Example AT+NWIP=0,192.168.0.100,255.255.255.0,192.168.0.1 OK  AT+NWIP +NWIP:0,192.168.0.100,255.255.255.0,192.168.0.1 OK  At+NWIP=? +NWIP:0,192.168.0.100,255.255.255.0,192.168.0.1 OK	
	Note <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• In Soft AP mode, after changing IP address, DHCP server pool range should also be updated based on the class of the changed IP address. Use AT+NWDHR to re-define DHCP server pool range after running AT+NWIP</li> <li>• In Soft AP mode, if the IP configuration is changed while the DHCP server is running, then the DHCP server must be restarted using the AT+RESTART or AT+NWDHS=0 &gt; AT+NWDHS=1 command</li> </ul>	
AT+NWDNS	<dns_ip>	Set the DNS server IP address of STA interface <dns_ip>: DNS server IP address Response: OK or ERROR
	?	Get the DNS server IP address of STA interface
	(none)	Response: +NWDNS:<dns_ip>
	Example AT+NWDNS=8.8.8.8 OK  AT+NWDNS +NWDNS:8.8.8.8 OK  Note <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• If AT+NWDNS=? is run under DHCP mode, it returns the DNS IP address from DHCP provision data regardless of any DNS IP address set with AT+NWDNS=&lt;dns_ip&gt;. ERROR:-7 ("No result" or "Not configured") can be returned if there is no DHCP provision data existing</li> <li>• If AT+NWDNS=? is run under Static IP mode, it returns the DNS IP address from AT+NWDNS=&lt;dns_ip&gt; that run previously or default one</li> <li>• If AT+NWDNS=&lt;dns_ip&gt; is run under DHCP mode, and the changes to take effect in Static IP mode, it requires a system restart</li> </ul>	
AT+NWDNS2	<dns_ip>	Set the 2 <sup>nd</sup> DNS server IP address of STA interface <dns_ip>: DNS server IP address Response: OK or ERROR
	?	Get the 2 <sup>nd</sup> DNS server IP address of STA interface

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	(none)	Response: +NWDNS2:<dns_ip>
	Example AT+NWDNS2=8.8.8.8 OK  AT+NWDNS2 +NWDNS2:8.8.8.8 OK  Note <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> </ul>	
AT+NWHOSt	<name>	Get the host IP address by name <name>: Domain name Response: +NWHOSt:<ip>
	Example at+nwhost=www. Renesas Electronics-semiconductor.com +NWHOSt:54.192.175.64 OK  Note <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> </ul>	
AT+NWPING	<iface>,<dst_ip>,<count>	Ping test <iface>: WLAN interface. 0 (WLAN0), 1 (WLAN1) <dst_ip>: Target IP address <count>: The number of ICMP message transmissions Response: +NWPING:<sent_count>,<recv_count>,<avg_time>,<min_time>,<max_time>
	Example AT+NWPING=0,192.168.0.1,4 +NWPING:4,4,0,0,0 OK  Note <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> </ul>	
AT+NWDHC	<dhcpc>	Start/Stop the DHCP client <dhcpc>: 0 (stop), 1 (start) Response: OK or ERROR
	?	Get the DHCP client status
	(none)	Response: +NWDHC:<dhcpc>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	Prerequisite DA16200/DA16600 should be connected to AP.  Example AT+NWDHC=1 OK  AT+NWDHC +NWDHC:1 OK  Note <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> </ul>	
AT+NWDHCHN	<hostname>	Store the DHCP client host-name <hostname> DHCP client host-name Response: OK or ERROR
	?	Get the DHCP client host-name which is stored by user
	(none)	Response: +NWDHCHN=<hostname>
	Example at+nwdhchn=TEST_DHCP*HOSTNAME ERROR:-615 at+nwdhchn=TEST-DHCP-HOSTNAME OK  Note <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>The hostname can contain only uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and minus sign (-)</li> </ul>	
AT+NWDHCHN DEL	(none)	Delete DHCP client host-name which was stored by user
	Example at+nwdhchn=TEST_DHCP*HOSTNAME ERROR:-615 at+nwdhchn=TEST-DHCP-HOSTNAME OK  Note <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> </ul>	
AT+NWDHR	<start_ip>,<end_ip>	Set an IP address range of the DHCP server <start_ip>: Starting IP address assigned by the DHCP server <end_ip>: Ending IP address assigned by the DHCP server Response: OK or ERROR
	?	Get an IP address range of the DHCP server
	(none)	Response: +NWDHR:<start_ip>,<end_ip>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	Prerequisite Soft AP mode  Example AT+NWDHR=10.0.0.2,10.0.0.11 OK  AT+NWDHR +NWDHR:10.0.0.2,10.0.0.11 OK  Note <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• DHCP server restart (AT+RESTART or AT+NWDHS=0 &gt; AT+NWDHS=0) is required for changes to take effect</li> </ul>	
AT+NWDHLT	<lease_time>	Set an IP lease time (in seconds) of the DHCP server <lease_time>: IP lease time (from 60 to 86400 seconds) Response: OK or ERROR
	?	Get an IP lease time of the DHCP server
	(none)	Response: +NWDHLT:<lease_time>
	Prerequisite Soft AP mode  Example AT+NWDHLT=1800 OK  AT+NWDHLT +NWDHLT:1800 OK  Note <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• DHCP server restart (AT+RESTART or AT+NWDHS=0 &gt; AT+NWDHS=0) is required for changes to take effect</li> </ul>	
AT+NWDHS	<dhcpcd>	Start/Stop DHCP server <dhcpcd>: 0 (stop), 1 (start) Response: OK or ERROR
	<dhcpcd>, <start_ip>,<end_ip>, <lease_time>	Start the DHCP server with options <dhcpcd>: 1 (start) <start_ip>: Starting IP address for the DHCP client <end_ip>: Ending IP address for the DHCP client <lease_time>: IP lease time (optional, in second, default is 1800) Response: OK or ERROR
	?	Get the DHCP client status
	(none)	Response: +NWDHS:<dhcpcd>



## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	Prerequisite Soft AP mode  Example AT+NWDHS=1 OK  AT+NWDHS=1,10.0.0.2,10.0.0.10,1800 OK  AT+NWDHS +NWDHS:1,10.0.0.2,10.0.0.10,1800 OK  Note <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> </ul>	
AT+NWDHIP	(none)	Show the information of the DHCP Client(s) connected Response: OK or ERROR When the response is OK, the following response comes first before OK. +NWDHIP:<mac_addr_1>,<ip_addr_1>;<mac_addr_2>,<ip_addr_2>;.....
	Example // One DHCP client is in the connected state. AT+NWDHIP +NWDHIP:80:35:c1:79:c1:da,10.0.0.2 OK  // Two DHCP clients are in the connected state. AT+NWDHIP +NWDHIP:80:35:c1:79:c1:da,10.0.0.2;b4:f1:da:b4:27:11,10.0.0.3 OK  // No DHCP client exists. AT+NWDHIP ERROR:-622 // Clients are not connected  Note <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>Use this command when DHCP server is running</li> </ul>	
AT+NWSNS AT+NWSNS1 AT+NWSNS2	<server_ip>	Set the SNTP server IP address/domain name <server_ip>: SNTP server IP address/domain name Response: OK or ERROR
	?	Get the SNTP server IP address
	(none)	Response: +NWSNS:<sntp>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre>AT+NWSNS=8.8.8.8 OK  AT+NWSNS +NWSNS:8.8.8.8 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• Up to three SNTP servers can be specified by users; an SNTP server is contacted in round robin manner if DA16200/DA16600 fails to synchronize the system time with a server</li> <li>• If not specified, default SNTP server will be used</li> </ul>	
AT+NWSNUP	<period>	Set the SNTP client update period (in seconds) <period>: SNTP client update period (from 60 to 129600 seconds) Response: OK or ERROR
	?	Get the SNTP client update period
	(none)	Response: +NWSNUP:<period>
	<p>Example</p> <pre>AT+NWSNUP=86400 OK  AT+NWSNUP +NWSNUP:86400 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> </ul>	
AT+NWSNTP AT+NWSNTP1 AT+NWSNTP2	<sntp>	Start/Stop the SNTP Client <sntp>: 0 (stop), 1 (start) Response: OK or ERROR
	<sntp>, <server_ip>, <period>	Start the SNTP client with options <sntp>: 1 (start) <server_ip>: SNTP server IP address (or domain) <period>: SNTP client update period (optional, second, default is 86400) Response: OK or ERROR
	?	Get the SNTP status
	(none)	Response: +NWSNTP:<sntp>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre>AT+NWSNTP=0 OK  AT+NWSNTP=1,pool.ntp.org,86400 OK  AT+NWSNTP +NWSNTP:1,pool.ntp.org,86400 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>If &lt;sntp&gt; is 1, SNTP client is started immediately and tries to do time sync with the server specified. &lt;sntp&gt;=1 also enables "auto start" of the SNTP client if DA16200/DA16600 reboots. &lt;sntp&gt;=0 removes "auto start" flag from NVRAM, so DA16200/DA16600 does not try to sync time when rebooted</li> </ul>	
AT+NWCCRT	(none)	<p>Check if certificates exist</p> <p>There are three sets of certificates:</p> <ul style="list-style-type: none"> <li>Set #1: for MQTT Root CA (bit 2)/Cert (bit 1)/Key (bit 0)/DH param (bit 9)</li> <li>Set #2: for HTTPS client for OTA Root CA (bit 5)/Cert (bit 4)/Key (bit 3)/DH param (bit 10)</li> <li>Set #3: for WPA Enterprise Root CA (bit 8)/Cert (bit 7)/Key (bit 6)/DH param (bit 11)</li> </ul> <p>For example: if DA16200/DA16600 has the Root CA and Cert in Set #1, the return value is 6.</p> <p>Response: +VER:&lt;cert&gt;</p>
	<p>Example</p> <pre>AT+NWCCRT +NWCCRT:6           ; MQTT OK  AT+NWCCRT +NWCCRT:56          ; HTTPS OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> </ul>	
AT+NWDCRT	(none)	<p>Delete all TLS certificates including private key</p> <p>Response: OK or ERROR</p>
	<p>Example</p> <pre>AT+NWDCRT OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> </ul>	

## DA16200 DA16600 Host Interface and AT Command

**Table 16: Certificate Command**

Escape Sequence	Parameters	Description
<ESC>C	<cert_id>,<content><ETX>	<p>Store certificate or private key.</p> <p>&lt;ESC&gt;C: To enter certificate input mode, type in &lt;ESC&gt;(0x1B) and C keys together</p> <p>&lt;cert_id&gt;: Certificate ID</p> <p>There are three sets of certificates:</p> <ul style="list-style-type: none"> <li>Set #1: for MQTT 0 (Root CA)/1 (Client Certificate)/2 (Private Key)</li> <li>Set #2: for HTTPS client for OTA 3 (Root CA)/4 (Client Certificate)/5 (Private Key)</li> <li>Set #3: for WPA Enterprise 6 (Root CA)/7 (Client Certificate)/8 (Private Key)</li> </ul> <p>&lt;content&gt;: Certificate data. Copy and paste cert ascii text. Max length is 2048</p> <p>&lt;ETX&gt;: Indication of the end of content (Ctrl+C, 0x03)</p> <p>Response: OK or ERROR</p> <p>For example:</p> <p>&lt;ESC&gt;C1,----- BEGIN CERTIFICATE ----- MIlodknvfano923nf/ ...&lt;ETX&gt;</p>
	<p>Example</p> <pre>&lt;ESC&gt;C0,Root CA&lt;ETX&gt; OK &lt;ESC&gt;C1,Client CA&lt;ETX&gt; OK &lt;ESC&gt;C2,Provate Key&lt;ETX&gt; OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> </ul>	
<ESC>Cert	<module>, <certificate type>, <mode>[, <format>, <length>, <content>]	<p>Store or delete a certificate/CA/private key/DH params</p> <p>&lt;ESC&gt;CERT: To enter certificate input mode</p> <p>&lt;module&gt;: Module ID. 0 - MQTT, 1 - HTTPs client for OTA, 2 - WPA Enterprise</p> <p>&lt;certificate type&gt;: Certificate type, 0 - CA certificate, 1 - Certificate, 2 - Private key, 3 - DH params</p> <p>&lt;mode&gt;: Input mode. 0 - Store, 1 - Deletion</p> <p>&lt;format&gt;: Certificate format, 0 - DER, 1 - PEM if mode is 0 (Store)</p> <p>&lt;length&gt;: Length of certificate if mode is 0 (Store)</p> <p>&lt;content&gt;: Certificate data if mode is 0 (Store)</p> <p>Response: OK or ERROR</p> <p>For example:</p> <p>&lt;ESC&gt;CERT,0,1146,----- BEGIN CERTIFICATE ----- MIIDFDCCAf...</p>

## DA16200 DA16600 Host Interface and AT Command

Escape Sequence	Parameters	Description
	<p>Example</p> <pre>&lt;ESC&gt;CERT,0,0,0,1,980,-----BEGIN CERTIFICATE-----... OK &lt;ESC&gt;CERT,0,1,0,1,990,-----BEGIN CERTIFICATE-----... OK &lt;ESC&gt;CERT,0,2,0,1,31 AT+WFPBC  0,-----BEGIN EC PRIVATE KEY-----... OK &lt;ESC&gt;CERT,0,3,0,1,432,-----BEGIN DH PARAMETERS-----... OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> </ul>	

### 5.3 Wi-Fi Function Commands

Table 17: Wi-Fi Function Command List

Command	Parameters	Description
AT+WFMODE	<mode>	Set the Wi-Fi mode <mode>: 0 (STA), 1 (Soft AP) Response: OK or ERROR
	?	Get the current Wi-Fi mode
	(none)	Response: +WFMODE:<mode>
	<p>Example</p> <pre>AT+WFMODE=0          ; Set Station mode OK  AT+WFMODE=1          ; Set Soft AP mode OK  AT+WFMODE             ; Get current Wi-Fi mode +WFMODE:1 OK  AT+WFMODE=?          ; Get current Wi-Fi mode +WFMODE:1 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>Wi-Fi mode is stored in NVRAM</li> <li>System restart is required for changes to take effect</li> </ul>	
AT+WFMAC	<mac>	Write a user MAC address in the NVRAM Response: OK or ERROR

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	?	Get the current MAC address of the activated WLAN interface
	(none)	Response: +WFMAC:<mac>
	<p>Example</p> <pre>AT+WFMAC=EC:9F:0D:9F:FA:64 OK  AT+WFMAC=? +WFMAC:EC:9F:0D:9F:FA:64 OK  AT+WFMAC +WFMAC:EC:9F:0D:9F:FA:64 OK  AT+WFMAC=? ; In Soft AP mode +WFMAC:EC:9F:0D:9F:FA:65 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• The last digit should be an even number to be a valid MAC address</li> <li>• Can be enabled by default in the SDK</li> <li>• A user MAC address is stored in NVRAM and a system restart is required for changes to take effect</li> <li>• DA16200/DA16600 provides three types of the MAC address and the priority is in the following order: Spoofing MAC address, User MAC address, OTP MAC address</li> <li>• When reading the MAC address in Soft AP mode, it becomes the MAC address that was written + 1</li> </ul>	
AT+WFSPF	<mac>	Write the spoofing MAC address in the NVRAM Response: OK or ERROR
	<p>Example</p> <pre>AT+WFSPF=EC:9F:0D:90:00:48 OK  AT+WFSPF=? +WFSPF:EC:9F:0D:90:00:48 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• Either odd or even number last digit of MAC address is accepted. Use this command only in STA mode</li> <li>• A spoofing MAC address is stored in NVRAM and a system restart is required for changes to take effect</li> <li>• DA16200/DA16600 provides three types of the MAC address and the priority is in the following order: Spoofing MAC address, User MAC address, OTP MAC address</li> <li>• The AT+WFMAC=? command can be used to read back the spoofing MAC address as this command does not support query</li> </ul>	

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
AT+WFOTP	<mac>	Write the MAC address in the OTP memory Response: OK or ERROR The MAC address written in the OTP is used as WLAN0 MAC address and MAC address + 1 will be used as WLAN1 MAC address
	<p>Example</p> <pre>AT+WFOTP=EC:9F:0D:90:00:48 OK  AT+WFMAC=? +WFOTP:EC:9F:0D:90:00:48 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• The last digit should be an even number to be a valid MAC address</li> <li>• Can be enabled by default in the SDK</li> <li>• An OTP MAC address is stored in OTP and system restart is required for changes to take effect</li> <li>• An old MAC address in the OTP will be invalidated if it exists</li> <li>• There are four MAC address slots available in OTP. It is possible to write the OTP MAC address four times in total at the production</li> <li>• DA16200/DA16600 provides three types of the MAC address and the priority is in the following order: Spoofing MAC address, User MAC address, OTP MAC address</li> <li>• The AT+WFMAC=? command can be used to read back the OTP MAC address as this command does not support query</li> <li>• When reading the MAC address in Soft AP mode, it becomes the MAC address that was written + 1</li> </ul>	
AT+WFSTAT	(none)	Get Wi-Fi configuration Response: +WFSTAT:<Wi-Fi interface><var>...

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre>AT+WFSTAT  +WFSTAT:sta0 mac_address= ec:9f:0d:9f:fa:64 wpa_state=DISCONNECTED disconnect_reason=0  OK  AT+WFSTAT  +WFSTAT:softap1 mac_address=ec:9f:0d:9f:fa:65 wpa_state=DISCONNECTED disconnect_reason=0  OK  AT+WFSTAT  +WFSTAT:sta0 mac_address=ec:9f:0d:9f:fa:64 bssid=70:5d:cc:32:15:32 ssid=MY_AP_SSID id=0 mode=STATION key_mgmt=WPA2-PSK pairwise_cipher=CCMP group_cipher=CCMP channel=3 wpa_state=COMPLETED  OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• A response can be different depending on the current DA16200/DA16600 status or mode</li> </ul>	
AT+WFPBC	(none)	<p>Run the WPS PBC method</p> <p>Response: OK or ERROR</p>



## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre>AT+WFPBC</pre> <p>OK</p> <pre>+WFJAP:1,'MY_APS_SSID',192.168.0.3</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• A WPS button can be pressed after issuing the command</li> <li>• A router should support WPS and PBC</li> <li>• The existing connection, if any, will be lost when this command is run</li> </ul>	
AT+WFPIN	<pin>	Run the WPS PIN method
	(none)	<pin>: PIN (eight digits) (none): Generate a random PIN Response: +WFPIN:<pin> OK or ERROR
	?	Get the current PIN Response: +WFPIN:<pin>
	<p>Example</p> <pre>AT+WFPIN=13557799</pre> <pre>+WFPIN:13557799</pre> <p>OK</p> <pre>AT+WFPIN</pre> <pre>+WFPIN:36269112</pre> <p>OK</p> <p style="text-align: right;">; Generate random number.</p> <pre>AT+WFPIN=?</pre> <pre>+WFPIN:36269112</pre> <p>OK</p> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• An AP should support WPS PIN</li> </ul>	
AT+WFCWPS	(none)	Cancel WPS (both PBC and PIN) Response: OK or ERROR
	<p>Prerequisite</p> <p>WPS should be in progress.</p> <p>Example</p> <pre>AT+WFCWPS</pre> <p>OK</p> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• Return error if WPS is not in progress</li> </ul>	

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
AT+WFCC	<code>	Set a country code <code>: Country code (defined by ISO 3166-1 alpha-2 standard) such as KR, US, JP, and CH Response: OK or ERROR
	?	Get the current country code
	(none)	Response: AT+WFCC=<code>
	<p>Example</p> <pre>AT+WFCC=KR OK  AT+WFCC +WFCC:KR OK  AT+WFCC=? +WFCC:KR OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• A country code is stored in the NVRAM</li> <li>• A country code consists of two characters</li> <li>• If a country is invalid, DA16200/DA16600 returns an error code that is -113</li> <li>• System restart is required for changes to take effect</li> <li>• If this command is run in Soft AP mode with a new country code and the operating channel range of the new country does not cover the operating channel currently set, the operating channel is automatically switched to channel 1</li> </ul>	
AT+WFRSSI	(none)	Get the current RSSI value Response: +RSSI: -34
	<p>Prerequisite</p> <p>DA16200/DA16600 should be connected to AP.</p> <p>Example</p> <pre>AT+WFRSSI +RSSI:-25 OK  .... (if there is no connection to an AP) AT+WFRSSI +RSSI:NOT_CONN  ERROR:-400</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• DA16200/DA16600 will respond "+RSSI:NOT_CONN" with error (-400) if the connection is not established</li> </ul>	

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
AT+WFSCAN	(none)	Scan APs Response: +WFSCAN:<bssid><t><frequency><t><signal strength><t><flag><t><ssid><LF>...
	Prerequisite The country code should be set via AT+WFCC.  Example AT+WFSCAN  +WFSCAN:70:5d:cc:32:15:32 2422 -30 [WPA2-PSK-CCMP][WPS][ESS] IPTIME_N604BLACK_ERIC b4:a9:4f:62:39:46 2422 -32 [WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS] SK_WiFiGIGA3943  OK  Note <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>An SSID can be missed in case of hidden AP</li> </ul>	
AT+WFPSCAN	<channel time limit>, <ch>.. <ch>	Get the passive scan result for the given parameters <channel time limit>: Channel scan time limit (should be more than 30000 microsecond) <ch>: Carrier frequency (from 0 to 14) Response: BSSID Wi-Fi_Channel RSSI SSID Security Type
	Prerequisite The country code should be set via AT+WFCC. Station mode  Example AT+WFPSCAN=120000,1,3,5  70:5d:cc:8b:49:8e 2412 -47 Gen_Port_*.5_AP [WPA2-PSK-CCMP][WPS][ESS] 72:5d:cc:c0:9a:c4 2412 -47 IPTIME_A3004NS-M_Bell [WPS][ESS] ... +PSCAN:TIMEOUT  Note <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK v3.2.3.0 or later</li> <li>Multiple parameters can be typed in &lt;ch&gt; as example ('0' means all channel)</li> </ul>	
AT+WFPCDT MIN	<bssid>, <min_threshold>	Set the passive scan minimum RSSI threshold condition <bssid>: BSSID <min_threshold>: minimum threshold (from -10 to -100) Response: OK or ERROR
	?	Get the current condition
	(none)	

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	Prerequisite Station mode  Example AT+WFPCDTMIN=72:5d:cc:d0:82:bc,-80 OK  AT+WFPCDTMIN +WFPCDTMIN: 72:5d:cc:d0:82:bc,-80 OK  Note <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK v3.2.3.0 or later</li> </ul>	
AT+WFPCDTMAX	<bssid>, <max_threshold>	Set the passive scan maximum RSSI threshold condition <bssid>: BSSID <max_threshold>: maximum threshold (from -10 to -100) Response: OK or ERROR
	?	Get the current condition
	(none)	
	Prerequisite Station mode  Example AT+WFPCDTMAX=72:5d:cc:c0:82:bc,-20 OK  AT+WFPCDTMAX +WFPCDTMAX: 72:5d:cc:c0:82:bc,-20 OK  Note <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK v3.2.3.0 or later</li> </ul>	
AT+WFJAP	(none)	Stop passive scan Response: OK or ERROR
	Prerequisite Station mode.  Example AT+WFJAP OK  Note <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK v3.2.3.0 or later</li> </ul>	
AT+WFJAP	<ssid>,<sec>[,<hidden>] (sec=0 5)	Connect to an AP <ssid>: AP SSID

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<code>&lt;ssid&gt;,&lt;sec&gt;,&lt;idx&gt;,&lt;key&gt;[,&lt;hidden&gt;]</code> (sec=1)	<p><code>&lt;sec&gt;</code>: Security protocol. 0 (OPEN), 1 (WEP), 2 (WPA), 3 (WPA2), 4 (WPA+WPA2), 5 (WPA3 OWE), 6 (WPA3 SAE), 7 (WPA2 RSN &amp; WPA3 SAE)</p> <p><code>&lt;idx&gt;</code>: Key index for WEP. 0~3</p> <p><code>&lt;enc&gt;</code>: Encryption. 0 (TKIP), 1 (AES), 2 (TKIP+AES)</p> <p><code>&lt;key&gt;</code>: Passphrase. 8 ~ 63 characters are allowed</p> <p><code>&lt;hidden&gt;</code>: 1 (&lt;ssid&gt; is hidden), 0 or [not specified] (&lt;ssid&gt; is NOT hidden)</p> <p>Response: OK or ERROR</p> <p>Operation Results:            +WFJAP:&lt;OPS_RESULT&gt;[,&lt;SSID&gt;,&lt;IP_ADDRESS&gt;]            +WFJAP:&lt;OPS_RESULT&gt;,&lt;REASON&gt;,&lt;REASON_CODE&gt;            &lt;OPS_RESULT&gt; : 1 (SUCCESS), 0 (FAILED)            If &lt;OPS_RESULT&gt; : 1            &lt;SSID&gt;: The SSID will be surrounded by single quotation mark            &lt;IP_ADDRESS&gt;: Assigned IP address and format is xxx.xxx.xxx.xxx            If &lt;OPS_RESULT&gt; : 0            &lt;REASON&gt; : well-known reason in text            &lt;REASON_CODE&gt; : if &lt;REASON&gt; is OTHER, this shows the reason code            For details about &lt;REASON&gt; or &lt;REASON_CODE&gt;, see <a href="#">Table 18</a>.</p>
	?	Get the AP provisioning information
	(none)	<p>Operation Results:            If provisioning data available:            +WFJAP:&lt;SSID&gt;,&lt;sec&gt;,&lt;enc&gt;,&lt;Passphrase&gt;            If provisioning data is not available:            ERROR:-410 (No SSID is found)</p>
<p>Example</p> <pre> AT+WFJAP=MY_AP_SSID,0                ; Open security OK +WFJAP:1,'MY_AP_SSID',192.168.43.32  AT+WFJAP=MY_AP_SSID,0,1              ; Open security + hidden SSID OK +WFJAP:1,'MY_AP_SSID',192.168.43.32  AT+WFJAP=MY_AP_SSID,1,0,12345        ; WEP security OK +WFJAP:1,'MY_AP_SSID',192.168.0.7  AT+WFJAP=MY_AP_SSID,1,0,12345,1      ; WEP + hidden AP OK +WFJAP:1,'MY_AP_SSID',192.168.0.7  AT+WFJAP=MY_AP_SSID,4,2,N12345678   ; WPA2 security           </pre>		

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	OK +WFJAP:1,'MY_AP_SSID',192.168.0.7  AT+WFJAP=MY_AP_SSID,4,2,N12345678,1 ; WPA2 + hidden AP OK +WFJAP:1,'MY_AP_SSID',192.168.0.7  AT+WFJAP=? +WFJAP:'MY_AP_SSID',4,2,'N12345678' OK  Note <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• The host should wait for both command response <b>OK</b> or <b>ERROR</b> and <b>Operation Result</b> ; wait for OK, and +WFJAP:1,'&lt;SSID&gt;',&lt;IP Address&gt; for successful connection</li> <li>• Depending on the network condition, it may take more time to get an Operation Result due to internal connection re-trials</li> <li>• No system reboot happens after running this command</li> <li>• The AP configuration parameters (AP Profile) are stored in NVRAM</li> <li>• WPA3 Personal is enabled by default in the SDK v3.2.5.0 or later, see the <a href="#">example</a></li> <li>• If &lt;sec&gt; is set to 6 (WPA3 SAE) or 7 (WPA2 RSN &amp; WPA3 SAE), 1 (AES) is only valid as &lt;enc&gt; because WPA3 SAE allows only CCMP</li> </ul>	
AT+WFJAPA	<ssid>[,<key>][,<hidden>]	Connect to an AP If <key> exists, security protocol is WPA+WPA2 and encryption is TKIP+AES. if <key> is omitted, security protocol is OPEN. <hidden>: 1 (<ssid> is hidden), 0 or [not specified] (<ssid> is NOT hidden) if <hidden> is omitted, <ssid> is not hidden. <ssid>: AP SSID <key>: Passphrase. 8 ~ 63 characters are allowed Response: OK or ERROR  Operation Results: +WFJAP:<OPS_RESULT>[, '<SSID>', '<IP_ADDRESS>'] +WFJAP:<OPS_RESULT>,<REASON>,<REASON_CODE> <OPS_RESULT>: 1 (SUCCESS), 0 (FAILED) If <OPS_RESULT>: 1 <SSID>: The SSID will be surrounded by single quotation mark <IP_ADDRESS>: Assigned IP address and format is xxx.xxx.xxx.xxx If <OPS_RESULT>: 0 <REASON>: Well-known reason in text <REASON_CODE>: If < REASON > is OTHER, this shows the reason code For details about <REASON> or <REASON_CODE>, see <a href="#">Table 18</a> .
	?	Get the AP provisioning information (SSID and Passphrase only)

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	(none)	<p>Operation Results:</p> <p>If Wi-Fi connection is success :</p> <p style="padding-left: 40px;">+WFJAPA:'&lt;SSID&gt;','&lt;Passphrase&gt;'</p> <p>If Wi-Fi connection is failed :</p> <p style="padding-left: 40px;">ERROR:-425 (No SSID found)</p>
	<p>Example</p> <pre> AT+WFJAPA=MY_AP_SSID                ; Open security OK +WFJAP:1,'MY_AP_SSID',192.168.43.32  AT+WFJAPA=MY_AP_SSID,1                ; Open security + hidden SSID OK +WFJAP:1,'MY_AP_SSID',192.168.43.32  AT+WFJAPA=MY_AP_SSID,N12345678        ; WPA2 security OK +WFJAP:1,'MY_AP_SSID',192.168.43.32  AT+WFJAPA=MY_AP_SSID,N12345678,1      ; WPA2 security + hidden AP OK +WFJAP:1,'MY_AP_SSID',192.168.43.32  AT+WFJAPA=? +WFJAPA:'MY_AP_SSID','N12345678' OK </pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• The host should wait for both command response <b>OK</b> or <b>ERROR</b> and <b>Operation Result</b> ; wait for OK, and +WFJAP:1,'&lt;SSID&gt;','&lt;IP Address&gt;' for successful connection</li> <li>• Depending on the network condition, it may take more time to get an Operation Result due to internal connection re-trials</li> <li>• No system reboot is required after running this command</li> <li>• The AP configuration parameters (AP Profile) are stored in NVRAM</li> </ul>	
AT+WFCAP	(none)	<p>Connect to an AP with the current WLAN0 interface configuration</p> <p>Response: OK or ERROR</p>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Prerequisite</p> <p>AP profile parameters should exist in NVRAM.</p> <p>Example</p> <pre>AT+WFCAP OK +WFJAP:1,'MY_AP_SSID',192.168.0.7  AT+WFCAP ERROR:-503      ; Connect to AP fail. (for example, No AP profile found)  AT+WFCAP ERROR:-460      ; Already connected</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• An AP profile can be stored in NVRAM by issuing the “AT+WFJAPA” or “AT+WFJAP” command</li> <li>• If there is no AP profile found, DA16200/DA16600 returns an error (-503)</li> <li>• If DA16200/DA16600 is already in connection with an AP, it returns an error (-460)</li> </ul>	
AT+WFQAP	(none)	<p>Disconnect from the currently associated AP</p> <p>Response: OK or ERROR</p>
	<p>Prerequisite</p> <p>DA16200/DA16600 should be connected to AP.</p> <p>Example</p> <pre>AT+ WFQAP OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• No error returns if it has already been disconnected from an AP</li> </ul>	
AT+WFSTA	(none)	<p>Check Wi-Fi connection</p> <p>Response: +WFSTA:&lt;status&gt;</p> <p>&lt;status&gt; 1 (Connected), 0 (disconnected)</p>



## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	Prerequisite Station mode.  Example <pre>AT+WFSTA +WFSTA:0 OK  AT+WFSTA +WFSTA:1 OK</pre> Note <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>If DA16200/DA16600 runs the command in Soft AP mode, it returns an error (-100)</li> </ul>	
AT+WFROAP	<roam>	Operate the STA roaming <roam>: 1 (run), 0 (stop) Response: OK or ERROR
	?	Get the roaming status
	(none)	Response: +WFROAP:<roam>
	Prerequisite Station mode  Example <pre>AT+WFROAP=1 OK  AT+WFROAP=0 OK  AT+WFROAP=? +WFROAP:1 OK</pre> Note <ul style="list-style-type: none"> <li>Can be enabled by default in SDK</li> <li>This command enables "simple" roaming. The roaming configuration consists of one parameter called the roaming threshold (set to AT+WFROTH, -65 by). Assume that the DA16200/DA16600 is connected to an AP, and there are other APs that have the same SSID and security settings around the DA16200/DA16600. As the DA16200/DA16600 is not fixed, if the RSSI value of the currently connected AP is lower than the specified threshold, it will try to connect to an AP with a higher RSSI (same SSID and security). If the condition is met, the DA16200/DA16600 will silently switch to the new AP without a disconnection event.</li> <li>The auto roaming start flag is stored in NVRAM when &lt;roam&gt; is set to 1, and roaming operation is enabled if the flag setting is not changed regardless of system reboot. If &lt;roam&gt; is 0, the roaming flag is removed from NVRAM and roaming is disabled.</li> <li>"Simple" roaming is not supported in DPM mode. So, setting DPM mode with command "AT+DPM=1" disables "The auto roaming start flag".</li> </ul>	

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
AT+WFROTH	<rss>	Set the STA roaming threshold <rss>: Roaming threshold value (from 0 to -95 dBm) Response: OK or ERROR
	?	Get the STA roaming threshold
	(none)	Response: +WFROTH:<rss>
	Prerequisite Station mode  Example AT+WFROTH=-55 OK AT+WFROTH=?  +WFROTH:-55 OK  Note <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• This command writes roaming threshold in NVRAM</li> <li>• When AT+WFROAP=1 is run, the roaming is enabled with the new threshold</li> </ul>	
AT+WFDIS	<disabled>	Set the Wi-Fi STA profile unused. If set to 1, DA16200/DA16600 will not start to connect to the configured AP when rebooting <disabled >: 1 (Unused), 0 (Used) Response: OK or ERROR
	?	Get the status of the Wi-Fi profile
	(none)	Response: +WFDIS:<disabled>
	Example AT+WFDIS=1 OK  AT+WFDIS=? +WFDIS:1 OK  Note <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• The “unused” flag is stored in the NVRAM</li> <li>• The flag affects DA16200/DA16600 during boot-up procedure. System restart is required for changes to take effect</li> </ul>	
AT+WFSAP	<ssid>,<sec>,<ch>,<code> (sec=0 5)	Set up Soft AP interface <ssid>: AP SSID. Max 32 characters are allowed

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<code>&lt;ssid&gt;,&lt;sec&gt;, &lt;enc&gt;,&lt;key&gt;, &lt;ch&gt;,&lt;code&gt; (sec=2 3 4 6 7)</code>	<p><code>&lt;sec&gt;</code>: Security protocol. 0 (OPEN), 2 (WPA), 3 (WPA2), 4 (WPA+WPA2) ), 5 (WPA3 OWE), 6 (WPA3 SAE), 7 (WPA2 RSN &amp; WPA3 SAE)</p> <p><code>&lt;enc&gt;</code>: Encryption. 0 (TKIP), 1 (AES), 2 (TKIP+AES)</p> <p><code>&lt;key&gt;</code>: Passphrase. 8 ~ 63 characters are allowed</p> <p><code>&lt;ch&gt;</code>: Operating channel (optional). Default is 1 or uses the current channel if Soft AP is operating</p> <p><code>&lt;code&gt;</code>: Country code (optional). If exists, <code>&lt;ch&gt;</code> is essential</p> <p>Response: OK or ERROR</p>
	?	Get the Soft AP interface configuration
	(none)	<p>Response:</p> <p>+WFSAP:'&lt;ssid&gt;,&lt;auth&gt;,&lt;enc&gt;,&lt;key&gt;,&lt;ch&gt;,&lt;code&gt;</p> <p>Operation Result:</p> <p>+WFSAP:&lt;ssid&gt; is printed on success</p>
<p>Example</p> <pre>AT+WFSAP=DA16200_MY_SSID,0,1,KR ; Open-Mode +WFSAP:DA16200_MY_SSID OK  AT+WFSAP=? +WFSAP:'DA16200_MY_SSID',0,1,KR OK  AT+WFSAP=DA16200_MY_SSID,3,1,12345678,1,KR ; WPA2-AES +WFSAP:DA16200_MY_SSID OK  AT+WFSAP=? +WFSAP:'DA16200_MY_SSID',3,1,'12345678',1,KR OK  AT+WFSAP='DA16200_MY_SSID',3,2,'12345678',1,KR ; WPA2-AES +WFSAP:DA16200_MY_SSID OK  AT+WFSAP=? +WFSAP:'DA16200_MY_SSID',3,1,'12345678',1,KR OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• The Soft AP configuration parameters are stored in NVRAM</li> <li>• If the command is issued in station mode, a reboot is required to start as Soft AP mode. (If the command is issued in Soft AP mode, then no system restart is required)</li> <li>• The ',' (comma) is included in the SSID string and enclose the SSID with a single quotation mark</li> <li>• WPA3 Personal is enabled by default in the SDK v3.2.5.0 or later. See the <a href="#">example</a></li> </ul>		

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<ul style="list-style-type: none"><li>If &lt;sec&gt; is set to 6 (WPA3 SAE) or 7 (WPA2 RSN &amp; WPA3 SAE), 1 (AES) is only valid as &lt;enc&gt; because WPA3 SAE allows only CCMP</li></ul>	
AT+WFOAP	(none)	Operate Soft AP interface Response: OK or ERROR
	Prerequisite A Soft AP profile should be stored in NVRAM.  Example AT+WFOAP OK  Note <ul style="list-style-type: none"><li>Can be enabled by default in the SDK</li><li>Run this command in Soft AP mode</li><li>If there is no profile in NVRAM, it returns an error (-522)</li><li>DA16200/DA16600 returns an error (-522) if it already operates as Soft AP</li></ul>	
AT+WFTAP	(none)	Stop the Soft AP interface. Response: OK or ERROR
	Prerequisite Soft AP mode  Example AT+WFTAP OK  Note <ul style="list-style-type: none"><li>Can be enabled by default in the SDK</li><li>This command is valid while DA16200/DA16600 is running in Soft AP mode</li></ul>	
Additional note for AT+WFSAP, AT+WFOAP, AT+WFTAP:  Example: In STA mode, in factory reset state ... AT+WFSAP=DA16200_OPEN,0 // set up Soft AP AT+RESTART // reboot to start in the configured Soft AP mode ... DUT starts as Soft AP .... AT+WFTAP // stop Soft AP if required AT+WFOAP // start Soft AP if required		
AT+WFRAP	(none)	Restart the Soft AP interface Response: OK or ERROR

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	Prerequisite A profile for Soft AP should be stored in NVRAM.  Example AT+WFRAP  OK  Note <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>This command is valid in Soft AP mode</li> <li>If it runs in station mode, DA16200/DA16600 returns an error (-100)</li> </ul>	
AT+WFLCST	(none)	Get connected station information Response: +WFLCST:<mac><LF><flags><LF><var>...
	Example AT+WFLCST  +WFLCST:a6:f2:7c:d4:53:1c flags=[AUTH][ASSOC][AUTHORIZED][SHORT_PREAMBLE][WMM][MAYBE_WPS][HT] aid=1 capability=0x421 listen_interval=10 wifi_mode=802.11n timeout_next=NULLFUNC POLL rx_packets=290 tx_packets=4 rx_bytes=29625 tx_bytes=10658 inact_cnt=0 connected_time=20 sta_count=1  OK  AT+WFLCST +WFLCST:NOT_FOUND OK  Note <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>If there is no station connected, then DA16200/DA16600 returns "+WFLCST:NOT_FOUND"</li> </ul>	
AT+WFAPWM	<mode>	Set IEEE 802.11 Wi-Fi mode of Soft AP interface <mode>: 0 (B/G/N), 1 (G/N), 2 (B/G), 3 (N), 4 (G), 5 (B) Response: OK or ERROR
	?	Get IEEE 802.11 Wi-Fi mode of Soft AP interface

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	(none)	Response: +WFAPWM:<mode>
	<p>Example</p> <pre>AT+WFAPWM=0 OK  AT+WFAPWM=1 OK  AT+WFAPWM=? +WFAPWM:1 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• The configuration is stored in NVRAM</li> <li>• System restart is required for changes to take effect</li> </ul>	
AT+WFAPCH	<ch>	Set the operating channel number for the Soft AP interface <ch>: Operating channel (from 0 to 13, 0 is auto) Response: OK or ERROR
	?	Get the operating channel number for the Soft AP interface
	(none)	Response: +WFAPCH:<ch>
	<p>Example</p> <pre>AT+WFAPCH=5 OK  AT+WFAPCH=? +WFAPCH:5 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• The configuration is stored in NVRAM</li> <li>• System restart is required for changes to take effect</li> </ul>	
AT+WFAPBI	<interval>	Set the AP beacon interval <interval>: Beacon interval (ms) Response: OK or ERROR
	?	Get the AP beacon interval
	(none)	Response: +WFAPBI:<interval>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre>AT+WFAPBI=200 OK  AT+WFAPBI=? +WFAPBI:200 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• The configuration is stored in NVRAM</li> <li>• System restart is required for changes to take effect</li> </ul>	
AT+WFAPUI	<timeout>	Set station disconnection timeout in Soft AP mode <timeout>: Disconnection timeout (sec) (from 30 to 86400, step = 10) If 0, the default value (300 seconds) is used Response: OK or ERROR
	?	Get station disconnection timeout in Soft AP mode
	(none)	Response: +WFAPUI:<timeout>
	<p>Example</p> <pre>AT+WFAPUI=60 OK  AT+WFAPUI=? +WFAPUI:60 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• Within the specified time, if an STA does not send any frame, Soft AP sends a NULL frame after the timeout is expired to check STA's inactivity. If no ACK is received from the STA, Soft AP removes the STA</li> <li>• The configuration is stored in NVRAM</li> <li>• System restart is required for changes to take effect</li> </ul>	
AT+WFAPRT	<threshold>	Set the AP RTS threshold (octets) <threshold>: RTS threshold (from 1 to 2347) Response: OK or ERROR
	?	Get the AP RTS threshold
	(none)	Response: +WFAPRT:<threshold>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre>AT+WFAPRT=2100 OK  AT+WFAPRT=? +WFAPRT:2100 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• If a frame that is bigger than the RTS threshold specified is to be sent, RTS/CTS frames are sent first to avoid collision in the air. By default, the RTS threshold is 2347</li> <li>• The configuration is stored in NVRAM</li> <li>• System restart is required for changes to take effect</li> </ul>	
AT+WFAPDE	<mac>	<p>Send de-authentication frame to the connected station</p> <p>&lt;mac&gt;: MAC address of the connected station</p> <p>Response: OK or ERROR</p>
	<p>Prerequisite</p> <p>DA16200/DA16600 should be connected to AP.</p> <p>Example</p> <pre>AT+WFAPDE=E6:0D:E5:A5:5D:B3 +WFDST:e6:0d:e5:a5:5d:b3 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• Use this command in Soft AP mode</li> <li>• Check the MAC address of an STA that needs to send de-authentication frame by using the command "AT+WFLCST"</li> <li>• If the operation is not successful (for example, a wrong MAC address is specified), the operation result (+WFDST:&lt;mac_addr&gt;) does not come</li> </ul>	
AT+WFAPDI	<mac>	<p>Send disassociation frame to the connected station</p> <p>&lt;mac&gt;: MAC address of the connected station</p> <p>Response: OK or ERROR</p>



## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	Prerequisite DA16200/DA16600 should be connected to AP.  Example AT+WFAPDI=E6:0D:E5:A5:5D:B3 +WFDST:e6:0d:e5:a5:5d:b3 OK  Note <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• Use this command in Soft AP mode</li> <li>• Check the MAC address of an STA that needs to send disassociation frame by using the command "AT+WFLCST"</li> <li>• If the operation is not successful (for example, a wrong MAC address is specified), no operation result (+WFDST:&lt;mac_addr&gt;) are sent</li> </ul>	
AT+WFWM	<wmm>	Set WMM on/off <wmm>: 0 (off), 1 (on) Response: OK or ERROR
	?	Get the WMM status
	(none)	Response: +WFWM:<wmm>
	Prerequisite Soft AP mode  Example AT+WFWM=1 OK  AT+WFWM=? +WFWM:1 OK  Note <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• WMM is enabled by default. If WMM is enabled, Beacon/Probe Rsp/Assoc frames will have WMM information. WMM enables QoS on the AC category</li> <li>• The configuration is stored in NVRAM</li> </ul>	
AT+WFWMPS	<wmmmps>	Set WMM-PS (WMM Power Save) on/off <wmmmps>: 0 (off), 1 (on) Response: OK or ERROR
	?	Get the WMM-PS status Response: +WFWMPS:<wmmmps>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Prerequisite</p> <p>Soft AP mode</p> <p>Example</p> <p>AT+WFWM=0</p> <p>OK</p> <p>AT+WFWM=?</p> <p>+WFWM:0</p> <p>OK</p> <p>Note</p> <ul style="list-style-type: none"><li>• Can be enabled by default in the SDK</li><li>• By default, WMM-PS is disabled. If WMM-PS is enabled, Beacon/Probe Rsp/Assoc Rsp frames sent from Soft AP will have a U-APSD flag set. For WMM and WMM-PS to properly work, the STA should also have WMM and WMM-PS certified</li><li>• The configuration is stored in NVRAM</li></ul>	

## DA16200 DA16600 Host Interface and AT Command

**Table 18: Wi-Fi Function Response List**

Response	Parameters	Description
+WFJAP	<result>,<ssid>,<ip>  <result>, <well-known-reason>  [,<reason_code>]	<p>The result of AP connection in STA mode (The result of AT+WFJAP or AT+WFJAPA or AT+WFCAP).</p> <p>&lt;result&gt;: 0 (failed), 1 (succeeded)</p> <p>For &lt;result&gt;: 1</p> <p>&lt;ssid&gt;: SSID of the AP when succeeded</p> <p>&lt;ip&gt;: IP address of the station when succeeded</p> <p>For &lt;result&gt;: 0</p> <p>&lt;well-known-reason&gt;: connection trial failure reason in text format, TIMEOUT / WRONGPWD / ACCESSLIMIT / OTHER</p> <p>TIMEOUT: connection attempt failed after continuous connection attempts</p> <p>WRONGPWD: WPA 4-Way Handshake failed, Pre-shared key (password) may be incorrect</p> <p>ACCESSLIMIT: disconnected because the authorized access number limit has been reached</p> <p>OTHER: other reasons</p> <p>&lt;reason_code&gt;: if &lt;well-known-reason&gt; is OTHER, this field shows which reason caused the connection trial failure</p> <p>See <a href="#">Appendix E</a>.</p> <p>For example:</p> <p>+WFJAP:0,TIMEOUT</p> <p>+WFJAP:1,'ap_test',192.168.0.10 // The Wi-Fi connection is established, and the assigned IP address is 192.168.0.10.</p>
+WFDAP	<reserved>, <well-known-reason>  [,<reason_code>]	<p>Disconnected from the AP</p> <p>&lt;reserved&gt;: 0</p> <p>&lt;well-known-reason&gt;: disconnection reason in text format, AUTH_NOT_VALID / DEAUTH / INACTIVITY / APBUSY / OTHER</p> <p>AUTH_NOT_VALID: Previous authentication no longer valid</p> <p>DEAUTH: De-authenticated as STA is leaving</p> <p>INACTIVITY: Disassociated due to inactivity</p> <p>APBUSY: Disassociated because AP is unable to handle all currently associated STAs</p> <p>&lt;reason_code&gt; : If &lt;well-known-reason&gt; is OTHER, this field shows which reason caused the disconnection</p> <p>See <a href="#">Appendix E</a>.</p> <p>For example:</p> <p>+WFDAP:0,INACTIVITY</p> <p>+WFDAP:0,DEAUTH</p> <p>+WFDAP:0,OTHER,8 ⬢</p> <p>WLAN_REASON_CLASS2_FRAME_FROM_NONAUTH_STA (8)</p>
+WFCST	<mac>	<p>A Wi-Fi station connected in Soft AP mode.</p> <p>&lt;mac&gt;: MAC address of the connected station</p>
+WFDST	<mac>	<p>A Wi-Fi station disconnected in Soft AP mode.</p> <p>&lt;mac&gt;: MAC address of the disconnected station</p>

## DA16200 DA16600 Host Interface and AT Command

### 5.4 Wi-Fi Function Commands for WPA3

Customers and users can configure DA16200/DA16600 as WPA3 STA or WPA3 Soft AP. WPA3 Personal is enabled by default in the SDK v3.2.5.0 or later. For older SDKs, WPA3 is not enabled by default, contact Renesas Electronics. Syntax of all the Wi-Fi function commands is the same as described in [Table 19](#) apart from the following commands where it needs to specify WPA3 specific parameters.

**Table 19: List of WPA3-Relevant Wi-Fi Function Commands**

Command	Parameters	Description
AT+WFJAP	See <a href="#">AT+WFJAP</a>	-
	Example	
	AT+WFJAP=MY_AP_SSID,5 ; WPA3 OWE OK +WFJAP:1,'MY_WPA3_AP_SSID',192.168.0.7	
	AT+WFJAP=MY_AP_SSID,5,1 ; WPA3 OWE + hidden SSID OK +WFJAP:1,'MY_AP_SSID',192.168.43.32	
	AT+WFJAP=MY_AP_SSID,6,2,N12345678 ; WPA3 SAE security OK +WFJAP:1,'MY_WPA3_AP_SSID',192.168.0.7	
	AT+WFJAP=MY_AP_SSID,6,1,12345678,1 ; WPA3 SAE + hidden AP OK +WFJAP:1,'MY_AP_SSID',192.168.0.7	
	AT+WFJAP=MY_AP_SSID,7,2,N12345678 ; WPA2(RSN)+WPA3 SAE security OK +WFJAP:1,'MY_WPA3_AP_SSID',192.168.0.7	
	AT+WFJAP=? +WFJAP:'MY_AP_SSID',6,1,'N12345678' OK	

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
AT+WFJAPA3	<wpa3_flag>,<ssid> [,<key>][,<hidden>]	<p>Connect to an AP which includes WPA3 type.            &lt;wpa3_flag&gt;: WPA2/WPA3 AP-type            If 1, WPA3-AP.            If 0, WPA/WPA2-AP.</p> <p>If &lt;key&gt; exists, security protocol is WPA+WPA2 and encryption is TKIP+AES.            If &lt;key&gt; is omitted, security protocol is OPEN.            &lt;hidden&gt;: 1 (&lt;ssid&gt; is hidden), 0 or [not specified] (&lt;ssid&gt; is NOT hidden)            If &lt;hidden&gt; is omitted, &lt;ssid&gt; is not hidden.            &lt;ssid&gt;: AP SSID            &lt;key&gt;: Passphrase. 8 ~ 63 characters are allowed            Response: OK or ERROR</p> <p>Operation Results:            +WFJAP:&lt;OPS_RESULT&gt;['&lt;SSID&gt;','&lt;IP_ADDRESS&gt;']            +WFJAP:&lt;OPS_RESULT&gt;,&lt;REASON&gt;,&lt;REASON_CODE&gt;            &lt;OPS_RESULT&gt; : 1 (SUCCESS), 0 (FAILED)            If &lt;OPS_RESULT&gt; : 1            &lt;SSID&gt;: The SSID will be surrounded by single quotation mark            &lt;IP_ADDRESS&gt;: Assigned IP address and format is xxx.xxx.xxx.xxx            If &lt;OPS_RESULT&gt; : 0            &lt;REASON&gt; : well-known reason in text            &lt;REASON_CODE&gt; : if &lt; REASON &gt; is OTHER, this shows the reason code.            For an explanation of &lt;REASON&gt; or &lt;REASON_CODE&gt;, see <a href="#">Table 18</a>.</p>
	?	Get the AP profile information (SSID and Passphrase only)
	(none)	<p>Operation Results:            If Wi-Fi connection is success :                +WFJAP:'&lt;SSID&gt;','&lt;Passphrase&gt;'            If Wi-Fi connection is failed:                ERROR:-425 (No SSID found)</p>
<p>Example</p> <pre> AT+WFJAPA3=0,WPA2_AP_SSID                ; Open-Mode OK +WFJAP:1,'WPA2_AP_SSID',192.168.0.2  AT+WFJAPA3=0,WPA2_AP_SSID,1                ; Open-Mode + hidden SSID OK +WFJAP:1,'WPA2_AP_SSID',192.168.0.2  AT+WFJAPA3=0,WPA2_AP_SSID,N12345678      ; WPA2 security OK +WFJAP:1,'WPA2_AP_SSID',192.168.0.2 </pre>		

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>AT+WFJAPA3=0,WPA2_AP_SSID,N12345678,1 ; WPA2 security + hidden AP OK +WFJAP:1,'WPA2_AP_SSID',192.168.0.2</p> <p>AT+WFJAPA3=1,WPA3_AP_SSID ; WPA3-OWE OK +WFJAP:1,'WPA3_AP_SSID',192.168.0.2</p> <p>AT+WFJAPA3=1,WPA3_AP_SSID,N12345678 ; WPA3-SAE security OK +WFJAP:1,'WPA3_AP_SSID',192.168.0.2</p> <p>AT+WFJAPA3=1,WPA3_AP_SSID,N12345678,1 ; WPA3-SAE + hidden AP OK +WFJAP:1,'WPA3_AP_SSID',192.168.0.2</p> <p>AT+WFJAPA3=? +WFJAPA3:'MY_AP_SSID','N12345678' OK</p> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• The host should wait for both command response <b>OK</b> or <b>ERROR</b> and <b>Operation Result</b> ; wait for OK, and +WFJAPA:1,&lt;SSID&gt;,&lt;IP Address&gt; for successful connection</li> <li>• Depending on the network condition, it may take more time to get an Operation Result due to internal connection re-trials</li> <li>• No system reboot happens after running this command</li> <li>• The AP configuration parameters (AP Profile) are stored in NVRAM</li> <li>• WPA3 Personal is enabled by default in the SDK v3.2.5.0 or later</li> </ul>	
AT+WFSAP	See <a href="#">AT+WFSAP</a>	-
	<p>Example</p> <p>AT+WFSAP=DA16200_MY_SSID,5,1,KR ; WPA3 OWE +WFSAP:DA16200_MY_SSID OK</p> <p>AT+WFSAP=DA16200_MY_SSID,7,1,12345678,1,KR ; WPA2 RSN &amp; WPA3 SAE, AES +WFSAP:DA16200_MY_SSID OK</p> <p>AT+WFSAP=? +WFSAP:'DA16200_MY_SSID',7,1,'12345678',1,KR OK</p>	

## DA16200 DA16600 Host Interface and AT Command

### 5.5 Wi-Fi Function Commands for WPA Enterprise

AT-CMD of DA16200 provides Wi-Fi commands that can be used as STA in WPA-Enterprise environment. To connect to the WPA-Enterprise AP, the DA16200 need to have profile information for the WPA-Enterprise AP and user account information.

**Table 20: WPA-Enterprise Wi-Fi Function Commands**

Command	Parameters	Description
AT+WFENTAP	<ssid>,<auth>,<enc>,<phase1>,<phase2>[,<hidden>] (phase1=0 1 2 4)	Create Enterprise profile to NVRAM. <ssid>: Enterprise AP SSID <auth>: Authentication mode for WAP-Enterprise. 8 (WPA-EAP), 9 (WPA2-EAP), 10 (WPA/WPA2-EAP). <enc>: Encryption Type. 0 (TKIP), 1 (AES), 2 (TKIP+AES) <phase1>: Phase #1 EAP type. 0 (Mixed), 1 (PEAP0), 2 (PEAP1), 3 (FAST), 4 (TTLS), 5 (TLS) <pahse2>: Phase #2 EAP type. 0 (Mixed), 1 (MSCHAPV2), 2 (GTC) <hidden>: 1 (<ssid> is hidden), 0 or [not specified] (<ssid> is NOT hidden)  Response: OK or ERROR Operation Results: +WFENTAP:<SSID>
	<ssid>,<auth>,<enc>,<phase1>[,<hidden>] (phase1=3 5)	
	?	Get the WPA-Enterprise configuration.
	(none)	Response: +WFENTAP:<ssid>,<auth>,<enc>,<phase1>,<pahse2>
	<p>Example</p> <pre> AT+WFENTAP=MY_AP_SSID,10,2,0      ; Phase#1 Mixed mode OK +WFENTAP:'MY_AP_SSID'  AT+WFENTAP=MY_AP_SSID,10,2,0,0    ; Phase#1, #2 Mixed mode OK +WFENTAP:'MY_AP_SSID'  AT+WFENTAP=MY_AP_SSID,10,2,0,0,1  ; Phase#1, #2 Mixed mode OK                                + hidden AP                                 +WFENTAP:'MY_AP_SSID' </pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK v3.2.3.0 or later</li> <li>WPA-Enterprise profile is stored in NVRAM</li> <li>"EAP-FAST" type for &lt;phase_1&gt; is not supported in SDK 3.2.3.0 and earlier SDK</li> <li>If &lt;phase1&gt; is set to 3 (FAST) or 5 (TLS), &lt;phase2&gt; is not allowed</li> <li>Need user account to connect to WPA-Enterprise AP. See "AT+WFENTLI" command</li> <li>System restart is required for changes to take effect</li> </ul>	

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
AT+WFENTLI	<id>[,<pw>]	Set User-ID/Password for WPA-Enterprise user account  <id>: Login-ID for WPA-Enterprise user account <pw>: Login-Password for WPA-Enterprise user account  Response: OK or ERROR
	?	Get current saved User-ID / Password for WPA-Enterprise user account  Response: OK or ERROR Operation Result: +WFENTLI:<id>,<pwd>
	(none)	
	<div>Example</div> <pre>AT+WFENTLI='USER_ACCOUNT_ID','USER_ACCOUNT_PWD'</pre> <p>OK</p> <pre>AT+WFENTLI</pre> <pre>+WFENTLI='USER_ACCOUNT_ID','USER_ACCOUNT_PWD'</pre> <pre>AT+WFENTLI=?</pre> <pre>+WFENTLI='USER_ACCOUNT_ID','USER_ACCOUNT_PWD'</pre> <div>Note</div> <ul style="list-style-type: none"><li>● Enabled by default in the SDK</li><li>● User account ID and PASSWORD are stored in NVRAM</li><li>● System restart is required for changes to take effect</li></ul>	

**Table 21: WPA-Enterprise Network Function Command**

Command	Parameters	Description
AT+NWTLSV	<ver>	Set the minimum accepted TLS protocol version when Phase#1 EAP type is TTLS or TLS of WPA Enterprise. The maximum accepted TLS protocol version is TLSv1.2.  For example, If TLSv1.0 is setup, the version of TLS session can be TLSv1.0, TLSv1.1 or TLSv1.2.  <ver>: Enterprise AP SSID. 0 <TLSv1.0>, 1 <TLSv1.1>, 2 <TLSv1.2>  Response: OK or ERROR
	?	Get current saved TLS version
	(none)	Response: +NWTLSV:<tls_version>
	Example  AT+NWTLSV=NEW_TLS_VER OK  AT+NWTLSV +NWTLSV:2 OK	



## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	AT+NWTLSV=? +NWTLSV:2 OK  Note <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK v3.2.3.0 or later</li> <li>• TLS Version number is stored as internal value</li> <li>• Default minimum accepted TLS protocol version is TLSv1.2</li> <li>• System restart is required for changes to take effect</li> </ul>	

### 5.5.1 WPA-Enterprise Connection Example

Create WPA-Enterprise profile and restart the DA16200/DA16600 to start Wi-Fi connection. For all cases, WPA-Enterprise user account information is needed.

Case #1, "Mixed" mode for EAP-type.  
 In this case, Encryption-type is configured as "Mixed" mode.  
 EAP-type and Encryption type are selected automatically.  
 AT+WFENTAP='WPA-Ent-AP-SSID',10,2,0  
 AT+WFENTLI='WPA-Ent\_User\_ID','WPA-Ent\_PWD'  
 AT+RESTART

Case #2, "Mixed" mode for EAP-type and Encryption type.  
 EAP-type and Encryption type are selected automatically.  
 AT+WFENTAP='WPA-Ent-AP-SSID',10,2,0,0  
 AT+WFENTLI='WPA-Ent\_User\_ID','WPA-Ent\_PWD'  
 AT+RESTART

Case #3, in case of PEAP0 and MSCHAPV2 for WPA-Enterprise.  
 AT+WFENTAP='WPA-Ent-AP-SSID',10,2,1,1  
 AT+WFENTLI='WPA-Ent\_User\_ID','WPA-Ent\_PWD'  
 AT+RESTART

Case #4, "Mixed" mode for EAP-type and set with TLS v1.0.  
 In this case, Encryption-type is configured as "Mixed" mode.  
 EAP-type and Encryption type are selected automatically.  
 AT+NWTLSV=1  
 AT+WFENTAP='WPA-Ent-AP-SSID',10,2,0  
 AT+WFENTLI='WPA-Ent\_User\_ID','WPA-Ent\_PWD'  
 AT+RESTART

## 5.6 Advanced Function Commands

### 5.6.1 MQTT Commands

The commands in [Table 22](#) are for configuring MQTT Client parameters. Restart the MQTT client for the configuration to take effect after running the commands.

## DA16200 DA16600 Host Interface and AT Command

### NOTE

In DPM mode, stop the MQTT Client (AT+NWMQCL=0) first before running the configuration commands. If any configuration commands are running in DPM mode without stopping MQTT Client, it returns ERROR:-635.

**Table 22: MQTT Configuration Command List**

Command	Parameters	Description
AT+NWMQBR	<host_name>,<port>	Set the host name (or IP address) and the port number of the MQTT Broker. <host_name>: Broker's domain name, or IP address <port>: Broker's port number Response: OK or ERROR
	?	Get the host name or IP address and the port number of the MQTT Broker. Response: +NWMQBR:<host_name>,<port>
	(none)	
	Prerequisite MQTT client should be disabled (+NWMQCL:0).	
	Example AT+NWMQBR=192.168.0.65,1884 OK  AT+NWMQBR=? +NWMQBR:192.168.0.65,1884 OK  Note <ul style="list-style-type: none"><li>Can be enabled by default in the SDK</li><li>The broker host name (or IP address) and port configured are stored in the NVRAM</li><li>MQTT restart is required for the new configuration to take effect</li></ul>	
AT+NWMQQOS	<qos>	Set the MQTT QoS level <qos>: 0 (at most once), 1 (at least once), 2 (exactly once) Response: OK or ERROR
	?	Get the MQTT QoS level
	(none)	Response: +NWMQQOS:<qos>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Prerequisite</p> <p>MQTT client should be disabled (+NWMQCL:0).</p> <p>Example</p> <pre>AT+NWMQQOS=1 OK  AT+NWMQQOS +NWMQQOS:1 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>MQTT restart is required for the new configuration to take effect</li> </ul>	
AT+NWMQTLS	<tls>	Enable/disable the MQTT TLS function <tls>: 1 (enable), 0 (disable) Response: OK or ERROR
	?	Get MQTT TLS status
	(none)	Response: +NWMQTLS:<tls>
	<p>Prerequisite</p> <p>Certificate should be stored. See <a href="#">Table 16</a>.</p> <p>MQTT client should be disabled (+NWMQCL:0).</p> <p>Example</p> <pre>AT+NWMQTLS=1 OK  AT+NWMQTLS +NWMQQOS:1 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>MQTT restart is required for the new configuration to take effect</li> </ul>	
AT+NWMQCS	<clean_session>	Set clean session mode <clean_session>: 1(session cleared), 0(session retained) Response: OK or ERROR
	?	Get clean session status
	(none)	Response: +NWMQCS:<clean_session>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	Prerequisite MQTT client should be disabled (+NWMQCL:0).  Example AT+NWMQCS=1 OK  AT+NWMQCS=? +NWMQCS:1 OK  Note <ul style="list-style-type: none"> <li>Can be enabled by default in SDK v3.2.3.0</li> <li>To disable/enable this feature, use <code>__MQTT_CLEAN_SESSION_MODE_SUPPORT__</code> in <code>config_generic_sdk.h</code></li> <li>If <code>__MQTT_CLEAN_SESSION_MODE_SUPPORT__</code> is not defined, mqtt client always connects to a mqtt broker in CleanSession=1 mode</li> <li>MQTT re-connection is required for this new configuration to take effect</li> <li>See MQTT Example: Using CleanSession=0</li> </ul>	
AT+NWMQTS	<num>,<topic#1>,<topic#2>,...	Set the topic(s) of the MQTT subscriber <num>: Number of topics <topic#n>: MQTT subscriber topic(s). Max topic length = 64 Response: OK or ERROR
	?	Get the MQTT subscriber topic(s)
	(none)	Response: +NWMQTS:<num>,<topic#1>,<topic#2>,...
	Prerequisite MQTT client should be disabled (+NWMQCL:0).  Example AT+NWMQTS=? ERROR:-654  AT+NWMQTS=1,da16k_sub OK  AT+NWMQTS=? +NWMQTS:1,"da16k_sub" OK  Note <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>Return "ERROR:-654" when there is no subscriber topic set</li> <li>After this command is run, the previously configured subscriber topic(s) is(are) cleared and set to the new one(s)</li> <li>MQTT restart is required for the new configuration to take effect</li> </ul>	
AT+NWMQATS	<topic>	Add the specified topic to MQTT configuration

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Prerequisite</p> <p>MQTT client should be disabled (+NWMQCL:0).</p> <p>Example</p> <pre>AT+NWMQATS=ABCD OK  AT+NWMQTS=? +NWMQTS:1,"ABCD" OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>Query command (AT+NWMQATS=?) not supported. If AT+NWMQATS=? is run, "?" is added as a topic</li> </ul>	
AT+NWMQDTS	<topic>	Delete the specified topic from MQTT configuration
	<p>Prerequisite</p> <p>MQTT client should be disabled (+NWMQCL:0).</p> <p>Example</p> <pre>AT+NWMQTS=? +NWMQTS:2,"ABCD","EFGH" OK  AT+NWMQDTS=ABCD OK  AT+NWMQTS=? +NWMQTS:1,"EFGH" OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> </ul>	
AT+NWMQTP	<topic>	Set the topic(s) of the MQTT publisher <topic>: MQTT publisher topic Response: OK or ERROR
	?	Get the MQTT publisher topic
	(none)	Response: +NWMQTP:<topic>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Prerequisite</p> <p>MQTT client should be disabled (+NWMQCL:0).</p> <p>Example</p> <pre>AT+NWMQTP=? ERROR:-662  AT+NWMQTP=da16k_pub OK  AT+NWMQTP=? +NWMQTP:da16k_pub OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• "ERROR:-662" means "No Publish topic exists"</li> <li>• MQTT restart is required for the new configuration to take effect</li> </ul>	
AT+NWMQV311	<use_v311>	Use MQTT protocol v3.1.1. Default is v3.1 <use_v311>: 1 (v3.1.1) / 0 (v3.1)
	?	Shows the MQTT protocol version currently set
	<p>Prerequisite</p> <p>MQTT client should be disabled (+NWMQCL:0).</p> <p>Example</p> <pre>AT+NWMQV311=? +NWMQV311:0 OK  AT+NWMQV311=1 OK  AT+NWMQV311=? +NWMQV311:1 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• MQTT restart is required for the new configuration to take effect</li> </ul>	
AT+NWMQPING	<period>	Set MQTT ping period <period>: Ping period (second) Response: OK or ERROR
	?	Get the current MQTT ping period
	(none)	Response: +NWMQPING:<period>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Prerequisite</p> <p>MQTT client should be disabled (+NWMQCL:0).</p> <p>Example</p> <pre>AT+NWMQPING=? +NWMQPING:600 OK  AT+NWMQPING=300 OK  AT+NWMQPING +NWMQPING:300 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>MQTT restart is required for the new configuration to take effect</li> </ul>	
AT+NWMQCID	<client_id>	Set the MQTT Client ID <client_id>: Client ID Response: OK or ERROR
	?	Get the current MQTT Client ID
	(none)	Response: +NWMQCID:<client_id>
	<p>Prerequisite</p> <p>MQTT client should be disabled (+NWMQCL:0).</p> <p>Example</p> <pre>AT+NWMQCID=? +NWMQCID:da16x_CCA4 //generate a default cid if there is no cid stored in NVRAM  AT+NWMQCID=client-1 OK  AT+NWMQCID +NWMQCID:client-1 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>MQTT restart is required for the new configuration to take effect</li> </ul>	
AT+NWMQLI	<name>,<pw>	MQTT login information <name>: ID <pw>: Password Response: OK or ERROR
	?	Get the MQTT login information

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	(none)	Response: +NWMQLI:<name>,<pw>
	Prerequisite MQTT client should be disabled (+NWMQCL:0).	
	Example AT+NWMQLI=? ERROR:-673  AT+NWMQLI=da16k_user,12345678 OK  AT+NWMQLI +NWMQLI:da16k_user,12345678 OK	
	Note <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• "ERROR:-673" means "No user name exists"</li> <li>• MQTT restart is required for the new configuration to take effect</li> </ul>	
AT+NWMQAUTO	<auto>	Enable/Disable auto-start of MQTT Client at reboot <auto>: 1 (Enable), 0 (Disable)
	?	Get the MQTT Client's auto start configuration status
	(none)	Response: +NWMQAUTO:<auto>
	Prerequisite MQTT client should be disabled (+NWMQCL:0).  Example AT+NWMQAUTO=? +NWMQAUTO:0 OK  AT+NWMQAUTO=1 OK  AT+NWMQAUTO +NWMQAUTO:1 OK  Note <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• Default is 0 (disable)</li> <li>• MQTT restart is required for the new configuration to take effect</li> </ul>	



## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
AT+NWMQWILL	<topic>,<msg>,<qos>	Set MQTT Will message <topic>: Will topic <msg>: Will message <qos>: Will QoS. 0 (at most once), 1 (at least once), 2 (exactly once) Response: OK or ERROR
	?	Get the MQTT Will message
	(none)	Response: +NWMQWILL:<topic>,<msg>,<qos>
	Prerequisite MQTT client should be disabled (+NWMQCL:0).  Example AT+NWMQWILL=? ERROR:-664 Or ERROR:-665  AT+NWMQWILL=da16k_will,bye,0 OK  AT+NWMQWILL +NWMQWILL:da16k_will,bye,0 OK  Note <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• "ERROR:-664 or -665" means topic or message is missing</li> <li>• MQTT restart is required for the new configuration to take effect</li> </ul>	
AT+NWMQDEL	(none)	Reset the MQTT configurations Response: OK or ERROR
	Prerequisite MQTT client should be disabled (+NWMQCL:0).  Example AT+NWMQDEL OK  Note <ul style="list-style-type: none"> <li>• This command will reset all MQTT configurations</li> <li>• If the MQTT client is running, run this command after the MQTT client is disabled by AT+NWMQCL=0</li> </ul>	

## DA16200 DA16600 Host Interface and AT Command

**Table 23: MQTT Operation Command List**

Command	Parameters	Description
AT+NWMQCL	<mqtt_client>	Enable/disable the MQTT client <mqtt_client>: 0 (disable), 1 (enable) Response: OK or ERROR
	?	Get the MQTT client status
	(none)	Response: +NWMQCL:<mqtt_client>
	Prerequisite DA16200/DA16600 should be connected to AP.  Example <pre>AT+NWMQCL=1 OK  AT+NWMQCL +NWMQCL:1 OK</pre> Note <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK</li> <li>• If the system restarts, then the MQTT client is not started automatically as this command is just to start/stop the MQTT client</li> <li>• Setting MQTT configuration parameters such as MQTT broker IP, port number, and subscriber topic, are required to be done before issuing this command</li> </ul>	
AT+NWMQMSG	<msg>,<topic>	Publish an MQTT message <msg>: Message to be published <topic>: MQTT topic (optional) Response: OK or ERROR Operation Results: Send Success: +NWMQMSGSEND:1 Send Failure: +NWMQMSGSEND:0,<err_code>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Prerequisite</p> <p>MQTT client should be enabled (+NWMQCL:1).</p> <p>Example</p> <pre>AT+NWMQMSG=Hello world !!! OK +NWMQMSGSEND:1  AT+NWMQMSG='{ "car": "red", "type": "bus" }' OK +NWMQMSGSEND:1  AT+NWMQMSG=Hello OK +NWMQMSGSEND:0,-6 // Send failed due to mqtt is not in connected state</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK v3.2.3.0 or later</li> <li>• If a single quotation is used in a message, surrounded by double quotation marks</li> <li>• In the default AT command image, the maximum total combined string length allowed for &lt;msg&gt; + &lt;topic&gt; should be less than or equal to 2066. So, if it needs to send a max length &lt;msg&gt; (2048 bytes long) with an explicit &lt;topic&gt; specified, the &lt;topic&gt; length should be 18 characters long or less. If it needs to send a message with the maximum length topic allowed (which is 64), send 2002 bytes &lt;msg&gt; in maximum</li> <li>• About Operation Results (+NWMQMSGSEND:1 or +NWMQMSGSEND:0,&lt;err_code&gt;):</li> <li>• Depending on network condition, a message publishing transaction (Qos 0, 1, 2) may take some time if network condition is not good</li> <li>• A new async response +NWMQMSGSEND:1 or +NWMQMSGSEND:0 that comes after "OK" indicates either completion of a publish transaction or failure</li> <li>• In CleanSession=1 mode, if mqtt is disconnected, the host can immediately get +NWMQMSGSEND:0, but in CleanSession=0 mode, +NWMQMSGSEND:0 is not sent but instead the transaction resumes when mqtt client re-connects. See MQTT Example: Using CleanSession=0</li> </ul>	
AT+NWMQUTS	<topic>	Unsubscribe from the specified topic
	<p>Prerequisite</p> <p>MQTT client should be enabled (+NWMQCL:1).</p> <p>Example</p> <pre>AT+NWMQUTS=ABCD OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK (available from SDK v3.2.3.0 or later)</li> <li>• This command should be run while the MQTT client is in a connected state with Broker</li> </ul>	

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
AT+NWMQTT	<code>&lt;host_name&gt;,&lt;port&gt;, &lt;sub_topic&gt;, &lt;pub_topic&gt;, &lt;qos&gt;,&lt;tls&gt;, &lt;username&gt;, &lt;password&gt;</code>	<p>Run the MQTT Client with options. After entering this command, system will reboot automatically. At reboot, DA16200/DA16600 tries to connect to the MQTT broker after the Wi-Fi connection is successfully established.</p> <p><code>&lt;host_name&gt;</code>: Broker's domain name or IP address,  <code>&lt;port&gt;</code>: Broker's port number  <code>&lt;sub_topic&gt;</code>: MQTT subscriber topic  <code>&lt;pub_topic&gt;</code>: MQTT publisher topic  <code>&lt;qos&gt;</code>: MQTT QoS level  <code>&lt;tls&gt;</code>: Enable/disable MQTT TLS. 1 (enable), 0 (disable)  <code>&lt;username&gt;</code>: Login ID (optional)  <code>&lt;password&gt;</code>: Login password (optional)  Response: OK or ERROR</p>
	<p><b>Prerequisite</b>  MQTT client should be disabled (+NWMQCL:0).</p> <p><b>Example</b>  AT+NWMQTT=192.168.0.65,1884,da16k_sub,da16k_pub,0,0  ; Below are logs after DA16200 reboot  +INIT:DONE,0    +WFJAP:1,'test_ap_ssid',192.168.0.88    +NWMQCL:1</p> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK</li> <li>After the system reboot, operation result is sent, see "+NWMQCL" response</li> </ul>	

The table below shows optional MQTT configuration commands for the MQTT brokers that require TLS ALPN, SNI, or Cipher Suite information from MQTT Client at the connection stage. These commands are enabled by default in SDK v3.2.3.0 or later.

**Table 24: MQTT Optional Configuration Commands**

Command	Parameters	Description
AT+NWMQALPN	<code>&lt;num&gt;, &lt;alpn#1&gt;, &lt;alpn#2&gt;, &lt;alpn#3&gt;</code>	<p>Set the TLS ALPN protocol name for MQTT</p> <p><code>&lt;num&gt;</code>: Number of ALPNs. Maximum number of ALPN is three  <code>&lt;alpn#n&gt;</code>: TLS ALPN protocol name. Maximum length of each ALPN protocol name is 24  Response: OK or ERROR</p>
	?	<p>Get the TLS ALPN(s) that have been set</p> <p>Response:  +NWMQALPN:&lt;num&gt;,&lt;alpn#1&gt;,&lt;alpn#2&gt;,&lt;alpn#3&gt;</p>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Prerequisite</p> <p>MQTT client should be disabled (+NWMQCL:0).</p> <p>Example</p> <p>AT+NWMQALPN=? ERROR:-644</p> <p>AT+NWMQALPN=2,alpn-protrol-name-an,alpn-protocol-name-ax OK</p> <p>AT+NWMQALPN +NWMQALPN:2,"alpn-protrol-name-an","alpn-protocol-name-ax" OK</p> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__MQTT_TLS_OPTIONAL_CONFIG__</code> is enabled in the SDK, this command will be enabled</li> <li>"ERROR:-644" means "No ALPN is set"</li> </ul>	
AT+NWMQSNI	<sni>	Set TLS SNI for MQTT <sni>: Server Name Indication. Maximum length of SNI is 64 Response: OK or ERROR
	?	Get the TLS SNI that has been set Response: +NWMQSNI:<sni>
	<p>Prerequisite</p> <p>MQTT client should be disabled (+NWMQCL:0).</p> <p>Example</p> <p>AT+NWMQSNI=? ERROR:-648</p> <p>AT+NWMQSNI=a38a9rhiu3roqb-ats.myserver.com OK</p> <p>AT+NWMQSNI +NWMQSNI: a38a9rhiu3roqb-ats.myserver.com OK</p> <p>Note</p> <ul style="list-style-type: none"> <li>Can be enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__MQTT_TLS_OPTIONAL_CONFIG__</code> is enabled in the SDK, this command will be enabled</li> <li>"ERROR:-648" means "No SNI is set"</li> </ul>	
AT+NWMQCSUIT	<cipher suite 1>, <cipher suite 2>, ...	Set TLS Cipher suites <cipher suite>: A hex decimal value of cipher suite. See <a href="#">Appendix D</a> . Maximum number of cipher suites is 17 Response: OK or ERROR

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	?	Get TLS cipher suites that have been set Response: +NWMQSNL:<number of cipher suites>,<cipher suite 1>,<cipher suite 2>,...
	Prerequisite MQTT client should be disabled (+NWMQCL:0).  Example AT+NWMQCSUIT=? ERROR:-650 or ERROR:-651  AT+NWMQCSUIT=c024,c023,c00a,c009,c00d,c032 OK  AT+NWMQCSUIT +NWMQCSUIT:6,c024,c023,c00a,c009,c00d,c032 OK  Note <ul style="list-style-type: none"> <li>• Can be enabled by default in the SDK v3.2.3.0 or later</li> <li>• If <code>__MQTT_TLS_OPTIONAL_CONFIG__</code> is enabled in the SDK, this command will be enabled</li> <li>• "ERROR:-650 or -651" means "No CSUIT info is set"</li> <li>• The hex pre-fix "0x" should be removed when one is typed</li> <li>• DA16200/DA16600 does not support all the cipher suites due to memory limitation. Contact Renesas Electronics for using other cipher suites that are not specified in <a href="#">Appendix D</a></li> </ul>	

**Table 25: MQTT Response List**

Response	Parameters	Description
+NWMQCL	<result> [,TOO_LONG_MSG_RX]	The result of the MQTT client connection <result>: 0 (disconnected), 1 (connected) For example: +NWMQCL:1 If MQTT connection to the MQTT broker is successfully established +NWMQCL:0 If the MQTT connection is NOT successfully established +NWMQCL:0,TOO_LONG_MSG_RX Can be sent when the MQTT is disconnected by unsupported message length only if the current mqtt connection is with clean_session=0 and qos greater than or equal to 1.
	Example ; When MQTT connection to the MQTT broker is successfully established +NWMQCL:1  ; When MQTT broker is down,	

## DA16200 DA16600 Host Interface and AT Command

Response	Parameters	Description
	+NWMQCL:0	<p>Note</p> <ul style="list-style-type: none"> <li>DA16200/DA16600 Debug Console log (UART0) <ul style="list-style-type: none"> <li>When MQTT connection to the MQTT broker is successfully established, DA16200/DA16600 sends "+NWMQCL:1" message to MCU (or AT command console). At this moment, the following log appears: <pre>&gt;&gt;&gt; MQTT Client connection OK</pre> </li> <li>When the MQTT broker is down, DA16200/DA16600 sends "+NWMQCL:0" message to MCU (or AT command console) after retrying connection. In console log, the following logs appears: <pre>Failed to receive pkt. (0x38) Failed to read pkt(0x7880) MQTT Client disconnected ... (state=6) [SUB] REQ mqtt_restart (count=1) Connecting FAIL (0x38) Unable to connect (The connection was refused.) ... [SUB] REQ mqtt_restart (count=5) Connecting FAIL (0x38) Unable to connect (The connection was refused.) [SUB] MAX Retry (Retry Cnt=6).</pre> </li> </ul> </li> <li>To get this <b>Operation Result</b>, it may take more time if the DA16200/DA16600 connection retrial happens depending on the test network condition</li> <li>This message is also sent after AT+NWMQTT is run or if any MQTT configuration command is run and then the system is restarted <ul style="list-style-type: none"> <li>DA16200/DA16600 restarts if the AT command format is <b>OK</b> <ul style="list-style-type: none"> <li>+INIT:DONE,0 message is sent as DA16200/DA16600 boots up</li> <li>If usage of the AT command is not valid, DA16200/DA16600 sends an <b>ERROR</b> message without restarting</li> </ul> </li> <li>DA16200/DA16600 tries to connect to the AP after the reboot <ul style="list-style-type: none"> <li>+WFJAP:0,&lt;reason&gt; or +WFJAP:1,'&lt;SSID&gt;','&lt;IP Address&gt;' as result of the Wi-Fi connection</li> <li>If the Wi-Fi connection information such as SSID or key is NOT stored correctly in the DA16200/DA16600 NVRAM, +WFJAP:x response is NOT sent and the MQTT connection is NOT attempted as well. Because the MQTT connection needs a successful Wi-Fi connection first</li> </ul> </li> <li>DA16200/DA16600 tries to connect to the MQTT broker after the Wi-Fi connection is established. The MQTT broker information is stored in NVRAM. Connection result – +NWMQCL:0 or +NWMQCL:1 – is sent over UART1 as a result</li> </ul> </li> <li>+NWMQCL:0 is sent when the Wi-Fi Link goes down due to the following conditions <ul style="list-style-type: none"> <li>Sometimes, ATCMD host can get an unsolicited +NWMQCL:0 message when DPM Wakeup under poor signal condition with the AP connected. An example is when Wi-Fi connection with AP becomes unstable – such as when DPM Keepalive fails, Beacon loss detected. In these cases, as STA, DA16200 / DA16600 tries to get "Wi-Fi link" down and up to reconnect with AP (while doing this, Wi-Fi is disconnected and then re-connected). MQTT client, when this kind of situation is detected, will try to re-connect to Broker after forcing disconnection. When MQTT disconnection happens, +NWMQCL:0 is sent to ATCMD host. On receipt of this unsolicited message, ATCMD host should wait for +NWMQCL:1</li> </ul> </li> <li>+NWMQCL:0,TOO_LONG_MSG_RX <ul style="list-style-type: none"> <li>If the current mqtt_client connection with Broker is configured with <b>clean_session=0</b> and <b>qos</b> is greater than or equal to 1,</li> </ul> </li> </ul>

## DA16200 DA16600 Host Interface and AT Command

Response	Parameters	Description
	<p>+NWMQCL:0,TOO_LONG_MSG_RX can be sent to ATCMD host (exceptional case). This message indicates that mqtt_client is disconnected by receiving a message that exceeds the message length limit (2048) of da16x mqtt client. What ATCMD host should do, in this situation, is that configuring clean_session to 1 and connect to Broker to delete the message, and then disconnect from Broker and reconnect with clean_session=0 again to start over (a kind of Recovery). According to MQTT Spec, Broker keeps sending a message that has not been ACKed by the client. In this case, the long message (valid for Broker, but not valid for da16x mqtt client) can repeatedly be sent to da16x mqtt client unless connection with clean_session=1 is made from da16x mqtt client</p> <ul style="list-style-type: none"> <li>○ This response message is enabled by default in SDK v3.2.3.0</li> <li>○ Example recovery flow on receipt of +NWMQCL:0,TOO_LONG_MSG_RX</li> </ul> <pre> ... // mqtt client is disconnected by receiving a message with // unsupported length +NWMQCL:0,TOO_LONG_MSG_RX AT+NWMQCS=1 // set clean_session=1 OK // connect to Broker (to clear the invalid long message) AT+NWMQCL=1 OKtable  +NWMQCL:1 AT+NWMQCL=0 // disconnect from Broker +NWMQCL:0  OK AT+NWMQCS=0 // set clean_session=0 OK AT+NWMQCL=1 // connect to Broker with clean_session=0 OK  +NWMQCL:1 </pre>	
+NWMQMSG	<msg>,<topic>,<length>	<p>Received the MQTT message</p> <p>&lt;msg&gt;: Message data</p> <p>&lt;topic&gt;: Received topic</p> <p>&lt;length&gt;: Message length</p>
	<p>Example</p> <p>; When DA16200/DA16600 receives a message from the MQTT publisher, the following message will be sent from DA16200/DA16600 to AT command console:</p> <pre>+NWMQMSG&gt;Hello world!!!!,da16k_sub,15</pre> <p>Note</p> <ul style="list-style-type: none"> <li>● MQTT client is in a connected state with the broker (+NWMQCL:1)</li> </ul>	



## DA16200 DA16600 Host Interface and AT Command

### 5.6.1.1 MQTT Client Connection Example

Configure the parameters and start the MQTT Client (After Wi-Fi Connection):

```
AT+NWMQBR=172.16.0.1,1884
AT+NWMQTS=1,da16k_sub
AT+NWMQTP=da16k_pub
AT+NWMQAUTO=1 (Optional, if DPM mode is used, setting this parameter is needed)
AT+NWMQCL=1
```

If the connection is successful, the following is shown:

```
+NWMQCL:1
```

If DA16K receives a PUBLISH from a broker, the following is shown:

```
+NWMQMSG>Hello World,da16k,11
```

DA16K can send a PUBLISH to a broker. Type the following command:

```
AT+NWMQMSG='Hello I'm DA16K'
```

### 5.6.1.2 MQTT TLS Connection Example

```
Configure the MQTT parameters:AT+NWMQBR=172.16.0.1,8883
AT+NWMQTS=1,da16k_sub
AT+NWMQTP=da16k_pub
AT+NWMQTLS=1
AT+NWMQAUTO=1 (Optional, if DPM mode is used, setting this parameter is needed)
```

To check the validity of a certificate, the DA16K should set the exact current time:

```
AT+TIME=yyyy-mm-dd,hh:mm:ss
```

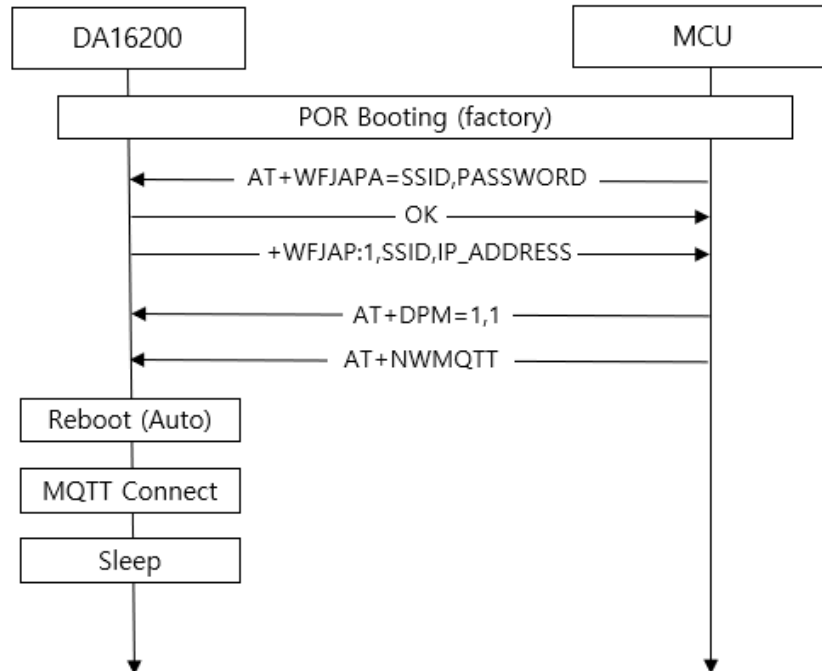
And store the certificate and private key if needed. (See <ESC>C in [Table 16](#))

After all settings are made, start the client:

```
AT+NWMQCL=1
```

## DA16200 DA16600 Host Interface and AT Command

### 5.6.1.3 MQTT Example with DPM



**Figure 22: Example Sequence to Initiate MQTT Protocol with DPM**

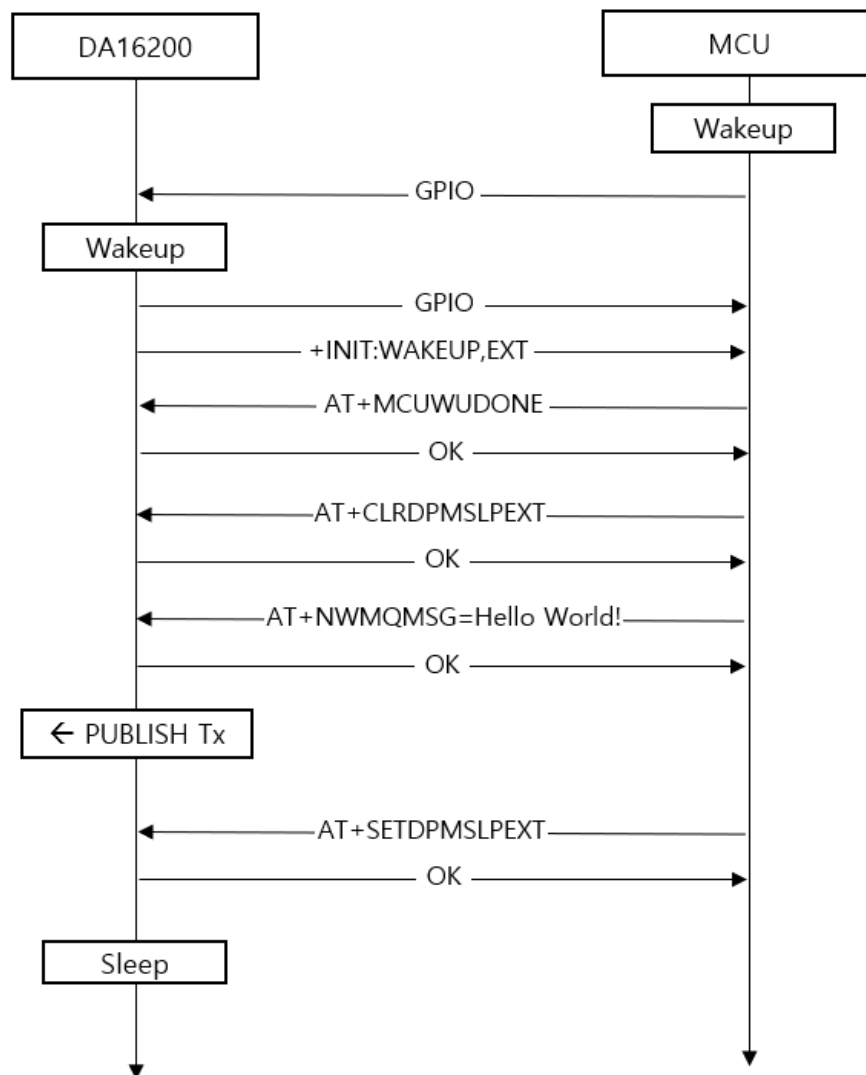
Figure 22 is an example sequence to initiate the MQTT protocol with DPM in the DA16200/DA16600.

In the normal BOOT state, connect to an AP (`AT+WFJAPA`) and change its run mode to DPM mode (`AT+DPM=1,1` ↓ optional parameter '1' means writing DPM mode to NVRAM and does not reboot. To make DPM mode take effect, a reboot is required).

To configure the MQTT connection information, enter command `AT+CLRDPMSEXT` and type the following as an example:

```
AT+NWMQTT=test.mosquitto.org,1883,sub_topic,pub_topic,0,0
```

## DA16200 DA16600 Host Interface and AT Command

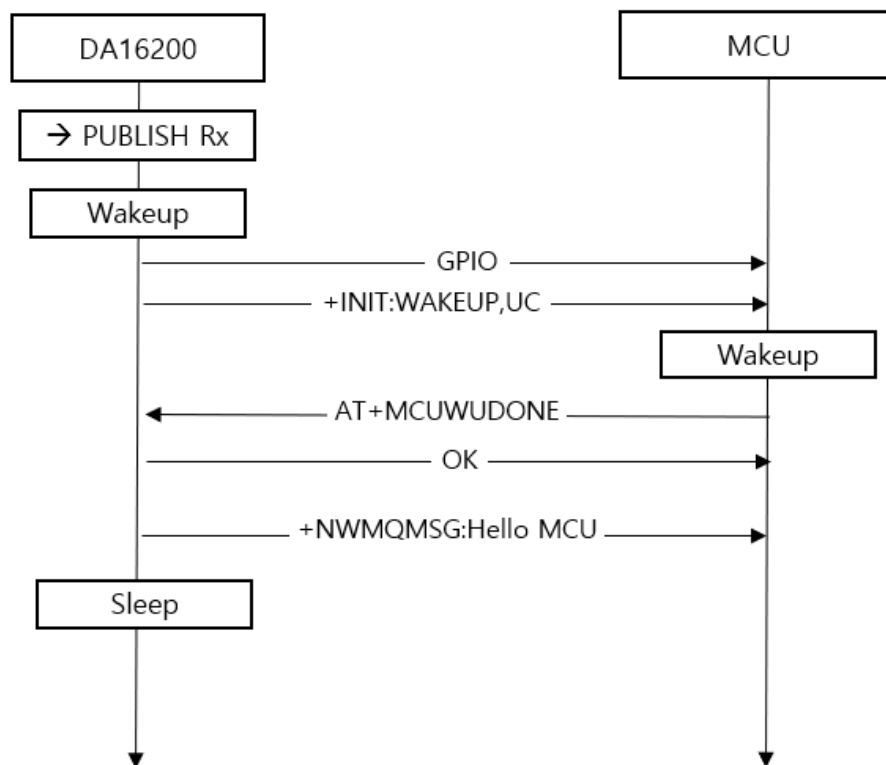


**Figure 23: Procedure to Send MQTT Messages**

Figure 23 shows the procedure to send an MQTT message in Sleep mode.

When MCU wakes up the DA16200/DA16600, the response +INIT:WAKEUP,EXT is sent. The MCU sends the command AT+MCUWUDONE to inform that MCU is ready to operate. To prevent that the DA16200/DA16600 enters DPM Sleep mode, MCU should send command AT+CLRDPM\_SLP\_EXT before an MQTT PUBLISH is sent. To make the DA16200/DA16600 enter DPM Sleep mode again, send a PUBLISH with command AT+NWMQMSG, and then enter command AT+SETDPM\_SLP\_EXT.

## DA16200 DA16600 Host Interface and AT Command



**Figure 24: Procedure to Process MQTT Messages**

Figure 24 shows how to process a received MQTT message while in Sleep mode.

When the DA16200/DA16600 wakes up by a PUBLISH message from an MQTT broker, the response +INIT:WAKEUP,UC is sent. The MCU sends the AT+MCUWUDONE to inform that it is ready to operate. Next, the DA16200/DA16600 sends the received PUBLISH to the MCU and enters DPM Sleep mode again.

### 5.6.1.4 MQTT Example: Changing Subscription Topic when Running

Assume that the Wi-Fi/MQTT connection is configured properly and DPM is set to 1 (TRUE). Below is the recommended sequence. Note that the double quotation marks are used.

## DA16200 DA16600 Host Interface and AT Command

```

A. Trigger RTC_WAKE_UP Event (by MCU)
B. Wait for "+INIT:WAKEUP,EXT" Response. Send AT+MCUWUDONE, and wait for "OK"
C. Run "AT+CLRDPMSPLEXT" command
D. Wait for "OK" response
E. loop running "AT+NWMQCL=?"
    E.1 if responses are "+NWMQCL:0" and "OK"
    E.2     then, goto E. to run "AT+NWMQCL=?" command
    E.3 else if responses are "+NWMQCL:1" and "OK"
    E.4     then, goto next, F.
    E.5 else if response is "ERROR:x"
    E.6     then, Run "AT+SETDPMSPLEXT"
    E.7         Wait for "OK" response
    E.8         return
F. Run "AT+NWMQCL=0"
G. Wait for "+NWMQCL:0" and "OK" response
H. Run "AT+NWMQTS=<New MQTT Subscription Topic>"
I. Wait for "OK" response
J. Run "AT+RESTART"
K. Wait for "+INIT:DONE,0" response
L. Wait for "+WFIJAP:1,'<SSID>','<IP ADDRESS>"
M. Wait for "+NWMQCL:1" response

```

### 5.6.1.5 MQTT Example: Reading Subscription Topic when Running

Assume that the Wi-Fi/MQTT connection is configured properly and DPM is set to 1 (TRUE). The reading of the MQTT publishing topic would be similar. Below is the recommended sequence. Note that the double quotation marks are used.

```

1. Trigger RTC_WAKE_UP Event
Wait for "+INIT:WAKEUP,EXT" Response. Send AT+MCUWUDONE, and wait for "OK"
2. Run "AT+CLRDPMSPLEXT" command
3. Wait for "OK" response
4. Run "AT+NWMQTS=?"
5. Wait for "+NWMQTS:<MQTT Subscription Topic>" and "OK" response

    Note that there are possibilities to receive the ERROR response if the format of the command
    has some errors.

6. Run "AT+SETDPMSPLEXT"
7. Wait for "OK" response

```

Assume that the Wi-Fi/MQTT connection is configured properly and DPM is set to 1 (TRUE). The reading of the MQTT publishing topic would be similar. Below is the recommended sequence. Note that the double quotation marks are used.

```

● Trigger RTC_WAKE_UP Event
1. Wait for "+INIT:WAKEUP,EXT" Response. Send AT+MCUWUDONE, and wait for "OK"
2. Run "AT+CLRDPMSPLEXT" command
3. Wait for "OK" response
4. Run "AT+NWMQTS=?"
5. Wait for "+NWMQTS:<MQTT Subscription Topic>" and "OK" response
6. Run "AT+SETDPMSPLEXT"

```

## DA16200 DA16600 Host Interface and AT Command

### 5.6.1.6 MQTT Example: Using CleanSession=0

#### 1) CleanSession=0 Mode

When an MQTT Client (Mqttd onward) establishes connection with an MQTT Broker (Broker onward), there are two types of session: CleanSession=1 and CleanSession=0.

**CleanSession=1:** default session type. when Broker gets a connect request from an Mqttd that tries to connect with an option "CleanSession=1" (which is default config on DA16x), Broker treats the connection as a "new" session. If there is any existing session associated with the same client\_id found, Broker clears that previous session and creates a new one with the client\_id.

**CleanSession=0:** when Broker gets a connect request from an Mqttd that tries to connect with an option "CleanSession=0", Broker tries to find a session (session data) with the same client\_id first. If it finds one, it keeps using that session for the new Mqttd.

While Mqttd is in operation with Broker, there may be times when the TCP connection gets unstable and disconnected (for example, mqtt ping failed) which may cause some messages that had been published to Broker at that specific disconnected time may not be delivered to a subscriber. If new messages (with QoS > 0) are published to Broker and for sessions that have been configured in "CleanSession=0", Broker retains and re-send them when the Mqttd is re-connected. Mqttd (if CleanSession=0 is enabled) also should retain the state of the unfinished / unacked messages until reconnection.

```
1647307743: New connection from 192.168.0.2 on port 8883.
1647307743: New client connected from 192.168.0.2 as da16x_D9CC (c1, k60).
1647307743: Sending CONNACK to da16x_D9CC (0, 0)
1647307743: Received SUBSCRIBE from da16x_D9CC
1647307743: SUB_TOPIC (QoS 2)
1647307743: da16x_D9CC 2 SUB_TOPIC
1647307743: Sending SUBACK to da16x_D9CC
```

Figure 25: Broker Console - CleanSession=1 Connection

```
1647307894: Client da16x_D9CC disconnected.
1647307898: New connection from 192.168.0.2 on port 8883.
1647307898: New client connected from 192.168.0.2 as da16x_D9CC (c0, k60).
1647307898: Sending CONNACK to da16x_D9CC (0, 0)
1647307898: Received SUBSCRIBE from da16x_D9CC
1647307898: SUB_TOPIC (QoS 2)
1647307898: da16x_D9CC 2 SUB_TOPIC
1647307898: Sending SUBACK to da16x_D9CC
```

Figure 26: Broker Console - CleanSession=0 Connection

Even with CleanSession=0 connection, Broker does not maintain session data if MQTT is disconnected in the following cases.

- If a new message is published with QoS 0 after MQTT is disconnected
- If MQTT connection QoS is 0

DA16x supports CleanSession=0 mode in the following way.

## DA16200 DA16600 Host Interface and AT Command

- "CleanSession=0 feature" is enabled by default in SDK v3.2.3.0 or later
- If customer application decides that **QoS 1 or QoS 2 and CleanSession=0** is used in their application, the message (payload) size (both Tx and Rx) should be pre-decided (because there is limitation in dpm user pool size). By default, 100 bytes are defined  

```
#define MQTT_MSG_TBL_PRESVD_MAX_PAYLOAD_LEN 100
```
- Depending on application's expected maximum payload size while operation, other value can be defined
- DPM User Pool has limited amount ( about 8K in total ) in the system
- Check available "free" DPM User Pool size first (by using the console command "dpm user\_pool"), and then calculate the max payload length and message number for the application if needed
- The default configuration (payload\_len: 100, max\_count: 10) allocates about 1.9KB of dpm user pool (Check mq\_msg\_tbl\_presvd\_t for detail)
- Search the following compiler options in config\_generic\_sdk.h  

```
//max payload length of a preserved message
#define MQTT_MSG_TBL_PRESVD_MAX_PAYLOAD_LEN          100
// max number of preserved messages
#define MQTT_MSG_TBL_PRESVD_MAX_MSG_CNT              10
```
- Supported command to set CleanSession mode: AT+NWMQCS=<1|0>

### 7. CleanSession and QoS Matrix Table for PUBLISH Rx

Subscriber			Unacked Message Delivery (After MQTT Reconnection)	QoS (Effective Actual)	Publisher Message's QoS
Case	Clean Session	QoS			
1	1	0	X	0	0
2	1	1	X	0	0
3	1	2	X	0	0
4	1	0	X	0	1
5	1	1	X	1	1
6	1	2	X	1	1
7	1	0	X	0	2
8	1	1	X	1	2
9	1	2	X	2	2
10	0	0	X	0	0
11	0	1	X	0	0
12	0	2	X	0	0
13	0	0	X	0	1
14	0	1	O	1	1
15	0	2	O	1	1
16	0	0	X	0	2
17	0	1	O	1	2

## DA16200 DA16600 Host Interface and AT Command

18	0	2	O	2	2
----	---	---	---	---	---

**Table 26: CleanSession and QoS Matrix in Message Rx**

With CleanSession=1, no unacked message delivery happens when MQTT reconnect happens (marked as x)

With CleanSession=0, only case 14, 15, 17, and 18 makes message redelivery happen for messages that had been delivered to Broker while the MQTTc was offline (marked as O).

### 8. CleanSession and QoS Matrix Table for PUBLISH Tx

Expectation 1 Application assumes that it failed to send a message and waits until mqtt gets re-connected.

Behavior 1 Application sends messages again.

Expectation 2 Application assumes that it failed to send a message but will resume sending the message when mqtt re-connected.

Behavior 2 Application simply waits as mqtt send the message automatically.

Publisher			Expectation if MQTT gets disconnected (while QoS 1/2 message is not fully acked or QoS 0 Send is being sent)	Behavior expected when MQTT Client re-connected
Case	Clean Session	QoS		
1	1	0	Expectation 1	Behavior 1
2	1	1	Expectation 1	Behavior 1
3	1	2	Expectation 1	Behavior 1
4	0	0	Expectation 1	Behavior 1
5	0	1	Expectation 2	Behavior 2
6	0	2	Expectation 2	Behavior 2

**Table 27: CleanSession and QoS Matrix in Message Tx**

When publishing a message from DA16x, application's expectation and action/behavior may be different if CleanSession=0 and QoS 1 or 2 are used in some specific cases.

In normal network condition, there is no difference in message send behavior between CleanSession=0 and CleanSession=1.

In some abnormal cases where QoS 1/2's ACK message (PUBACK, PUBREC, PUBREL, or PUBCOMP) get lost due to some bad network conditions (which can cause Mqttc re-connection), CleanSession=0 can recover the previous message state and resume the communication with Broker.

However, if CleanSession=1 is used, when Mqttc is disconnected, it can safely re-transmit the message when Mqttc is reconnected. Depending on use cases of applications / host applications, either approach (CleanSession=0 or CleanSession=1 ) can be utilized.



## DA16200 DA16600 Host Interface and AT Command

### 2) How to Connect with CleanSession=0

```
AT+NWMQOS=2
OK
AT+NWMQCS=0
OK
AT+NWMQCL=1
OK
+NWMQCL:1
```

To activate “**CleanSession=0 support mode**” in DA16x, QoS should be 1 or 2 and CleanSession option should be set to 0. If either option (CleanSession and QoS) is not set as above, CleanSession=0 support mode is disabled.

### 3) How to Restart CleanSession=0 test

If it needs to re-test (fresh new test) with CleanSession=0 mode, depending on the previous session type, it may need Broker to clear the previous session.

The reason is that since an Mqttc connects with CleanSession=0, Broker does not delete the session data until the Mqttc re-connects with CleanSession=1.

Case 1: Previous session is CleanSession=1 and need to restart a new CleanSession=0 test

```
AT+NWMQCL=0
OK
AT+NWMQCS=0
OK
AT+NWMQCL=1
OK
+NWMQCL:1
```

Case 2: Previous session is CleanSession=0 and need to re-test another CleanSession=0 test run.

```
AT+NWMQCL=0
+NWMQCL:0

OK
AT+NWMQCS=1
OK
AT+NWMQCL=1
OK

+NWMQCL:1
AT+NWMQCL=0
+NWMQCL:0

OK
AT+NWMQCS=0
OK
AT+NWMQCL=1
OK

+NWMQCL:1
```

## DA16200 DA16600 Host Interface and AT Command

### 4) PUBLISH Rx Test Steps

Test steps are as follows under non-DPM and DPM mode.

#### Non-DPM mode:

- DA16x: connect to Broker
- Publisher: send one or two messages
- DA16x: check if the messages are received
- DA16x: disconnect from Broker
- Publisher: send one or two messages (let say msg\_A)
- DA16x: reconnect to Broker
- DA16x: check if msg\_A (sent while DA16x is offline) is received

#### DPM mode:

- DA16x: connect to Broker. Enter DPM Sleep
- Publisher: send one or two messages
- DA16x: check if the messages are received
- DA16x: turn off AP. Do not turn on AP, but wait for the mqtt keep alive period (to make sure Broker recognizes the Mqttc disconnection)
- Publisher: send one or two messages (let say msg\_A)
- DA16x: turn on AP. Wait until DA16x is connected to AP
- DA16x: reconnected to AP and check if msg\_A (sent while DA16x is offline) is received

#### NOTE

The mosquitto broker (Broker), mosquitto publisher (Publisher), and DA16x (Subscriber) are used for the test. Message length from publisher should be less than or equal to 100. If longer messages are sent, they may not be restored properly when mqtt is reconnected.

### 5) PUBLISH Rx Test Steps - Example 1 (Non-DPM)

Below are the test steps for case 15 (non-DPM mode).

[DA16x] connect Mqttc with CleanSession=0 and QoS 2

```
AT+NWMQOOS=2
OK
AT+NWMQCS=0
OK
AT+NWMQCL=1
OK
+NWMQCL:1
```

[Other Publisher] publish messages

```
C:\mosquitto>mosquitto_pub -h 192.168.0.230 -p 8883 --cafile cas.pem --cert
wifiuser.pem --key wifiuser.key --tls-version tlsv1 --insecure -q 2 -t SUB_TOPIC -
m "Hello_q2"
```

## DA16200 DA16600 Host Interface and AT Command

```
C:\mosquitto>mosquitto_pub -h 192.168.0.230 -p 8883 --cafile cas.pem --cert
wifuser.pem --key wifuser.key --tls-version tlsv1 --insecure -q 2 -t SUB_TOPIC -
m "Hello_q2_2"
```

[DA16x] check the messages are successfully received

```
+NWMQMSG:Hello_q2,SUB_TOPIC,8
+NWMQMSG:Hello_q2_2,SUB_TOPIC,10
```

[DA16x] disconnect from Broker

```
AT+NWMQCL=0
+NWMQCL:0
OK
```

[Other Publisher] publish two messages (while DA16x is in disconnected state)

```
C:\mosquitto>mosquitto_pub -h 192.168.0.230 -p 8883 --cafile cas.pem --cert
wifuser.pem --key wifuser.key --tls-version tlsv1 --insecure -q 2 -t SUB_TOPIC -
m "Hello_q2_3"

C:\mosquitto>mosquitto_pub -h 192.168.0.230 -p 8883 --cafile cas.pem --cert
wifuser.pem --key wifuser.key --tls-version tlsv1 --insecure -q 2 -t SUB_TOPIC -
m "Hello_q2_4"
```

[DA16x] reconnect to Broker and check if the two messages that had been published while DA16x is in disconnected state are received successfully.

```
AT+NWMQCL=1
OK

+NWMQCL:1

+NWMQMSG:Hello_q2_3,SUB_TOPIC,10
+NWMQMSG:Hello_q2_4,SUB_TOPIC,10
```

### 6) PUBLISH Rx Test Steps - Example 2 (DPM)

Below are the test steps for case 18 (DPM mode). Note that mosquitto broker and mosquitto publisher are used for test.

[DA16x] Connect with CleanSession=0 and QoS 2

```
AT+NWMQOOS=2
OK
AT+NWMQCS=0
OK
AT+RESTART
OK

+INIT:DONE,0

+WFJAP:1,'SYN_TEST_AP',192.168.1.195
```

## DA16200 DA16600 Host Interface and AT Command

```
+ATPROV=STATUS 0  
  
+NWMQCL:1
```

[Other Publisher] Publish messages

```
C:\mosquitto>mosquitto_pub -h 192.168.0.230 -p 8883 --cafile cas.pem --cert  
wifiuser.pem --key wifiuser.key --tls-version tlsv1 --insecure -q 2 -t SUB_TOPIC -  
m "Hello_q2_1"  
  
C:\mosquitto>mosquitto_pub -h 192.168.0.230 -p 8883 --cafile cas.pem --cert  
wifiuser.pem --key wifiuser.key --tls-version tlsv1 --insecure -q 2 -t SUB_TOPIC -  
m "Hello_q2_2"
```

[DA16x] check the messages are successfully received

**Note:** At wakeup time, ATCMD host should send AT+CLRDPMPLPEXT to get +NWMQMSG after which AT+SETDPMPLPEXT should be sent by ATCMD host to let DA16x enter DPM Sleep.

```
+INIT:WAKEUP,UC  
  
+ATPROV=STATUS 0  
  
+NWMQMSG:Hello_q2_1,SUB_TOPIC,10  
  
+INIT:WAKEUP,UC  
  
+ATPROV=STATUS 0  
  
+NWMQMSG:Hello_q2_2,SUB_TOPIC,10
```

[DA16x] Turn OFF AP

```
+INIT:WAKEUP,NOBCN  
  
+ATPROV=STATUS 0  
  
+WFDAP:0,INACTIVITY  
...
```

[Broker] make sure Mqttc is disconnected

```
...  
1647405247: Socket error on client da16x_D9CC, disconnecting.  
...
```

[Other Publisher] publish two messages (while DA16x is in disconnected state)

```
C:\mosquitto>mosquitto_pub -h 192.168.0.230 -p 8883 --cafile cas.pem --cert  
wifiuser.pem --key wifiuser.key --tls-version tlsv1 --insecure -q 2 -t SUB_TOPIC -  
m "Hello_q2_3"  
  
C:\mosquitto>mosquitto_pub -h 192.168.0.230 -p 8883 --cafile cas.pem --cert  
wifiuser.pem --key wifiuser.key --tls-version tlsv1 --insecure -q 2 -t SUB_TOPIC -  
m "Hello_q2_4"
```

## DA16200 DA16600 Host Interface and AT Command

[DA16x] Turn ON AP

[DA16x] Wait until AP is connected and see whether “hello\_qos\_3” and “hello\_qos\_4” are received

```
+INIT:DONE,0  
  
+WFJAP:1, 'SYN_TEST_AP',192.168.1.195  
  
+ATPROV=STATUS 0  
  
+NWMQCL:1  
  
+NWMQMSG:Hello_q2_3,SUB_TOPIC,10  
  
+NWMQMSG:Hello_q2_4,SUB_TOPIC,10
```

### 7) PUBLISH Tx Test Steps

Test steps are as follows:

- DA16x: connect to Broker
- DA16x: send a messages
- DA16x: check if the message send is successful

#### NOTE

Message length from DA16x should be less than or equal to 100 bytes for case 5 and 6 configuration. Sending longer messages returns failure. For cases other than case 5 or 6, message length limit is 2048 bytes

### 8) PUBLISH Tx Test Steps – Example

Below are the test steps for case 6 (non-DPM mode).

```
AT+NWMQOOS=2  
OK  
AT+NWMQCS=0  
OK  
AT+NWMQCL=1  
OK  
  
+NWMQCL:1  
AT+NWMQMSG=hello_q2  
OK  
  
+NWMQMSGSD:1
```

## DA16200 DA16600 Host Interface and AT Command

### 5.6.2 HTTP-Client Commands

Table 28: HTTP-Client Command List

Command	Parameters	Description
AT+NWHTC	<url>,<method>(<body>)	Start the HTTP client with options <url>: HTTP server address <method>: GET, POST or PUT <body>: Body for POST and PUT methods, omitted for other methods. For JSON type, enclose the message with the single quotation - <'body'>
	Prerequisite DA16200/DA16600 should be connected to AP.  Example 1 AT+NWHTC=http://httpbin.org/get,get OK  Example 2 AT+NWHTC=http://httpbin.org/post,post,HTTP-Client POST method sample test! OK For JSON type, AT+NWHTC=http://httpbin.org/post,post,{ username: "aaa", password: "1234"} OK	
	Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If __SUPPORT_HTTP_CLIENT_FOR_ATCMD__ is enabled in the SDK, this command will be enabled</li> </ul>	
	<url>,message,<'header+body'>	Users can directly input header and body as plain text. Line feeds and carriage returns are inserted as \r\n. <url>: HTTP server address message: Use the message as a fixed option (not the http method) <'header+body'>: Enter a plain text string in the form of 'header+body'

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Prerequisite</p> <p>DA16200/DA16600 should be connected to AP.</p> <p>Example 1 : GET method request (header)</p> <pre>AT+NWHTC=http://httpbin.org/get,message,'GET /get HTTP/1.1\r\nHost: httpbin.org\r\nConnection: Close\r\n\r\n'</pre> <p>OK</p> <p>Example 2 : POST method request (header+body)</p> <pre>AT+NWHTC=http://httpbin.org/post,message,'POST /postHTTP/1.1\r\nHost: httpbin.org\r\nAccept: */*\r\nContent-Length: 10\r\nConnection: Close\r\n\r\nHelloWorld\r\n'</pre> <p>OK</p> <p>For JSON type,</p> <pre>AT+NWHTC=http://httpbin.org/post,message,'POST /postHTTP/1.1\r\nHost: httpbin.org\r\nContent-Type: application/json\r\nContent-Length: 40\r\nConnection: Close\r\n\r\n{username: "aaa", password: "1234"}\r\n'</pre> <p>OK</p> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__SUPPORT_HTTP_CLIENT_FOR_ATCMD__</code> is enabled in the SDK, this command will be enabled</li> </ul>	
AT+NWHATCH	<url>,<method> >(<msg>)	<p>AT+NWHATCH with H appended after AT+NWHTC.</p> <p>All parameters and functions are exactly the same as AT+NWHTC.</p> <p>The difference is that data size is inserted in front of the received data and transmitted.</p> <p>Start the HTTP client with options.</p> <p>&lt;url&gt;: HTTP server address</p> <p>&lt;method&gt;: GET, POST or PUT</p> <p>&lt;msg&gt;: Request message for POST and PUT methods</p>

## DA16200 DA16600 Host Interface and AT Command

	<p>Prerequisite</p> <p>DA16200/DA16600 should be connected to AP.</p> <p>Example 1</p> <p>AT+NWHTCH=https://httpbin.org/get,get OK</p> <p>+NWHTCDATA:225,HTTP/1.1 200 OK Date: Fri, 02 Dec 2022 01:17:30 GMT Content-Type: application/json Content-Length: 297 Connection: close Server: gunicorn/19.9.0 Access-Control-Allow-Origin: * Access-Control-Allow-Credentials: true</p> <p>+NWHTCDATA:297,{ "args": {}, "headers": { "Accept": "*/*", "Host": "httpbin.org", "User-Agent": "lwIP/2.1.2 (http://savannah.nongnu.org/projects/lwip)", "X-Amzn-Trace-Id": "Root=1-6389522a-4ceffbc701b0e20f348c5ecc" }, "origin": "124.50.108.25", "url": "https://httpbin.org/get" }</p> <p>+NWHTCSTATUS:0</p> <p>Example 2</p> <p>AT+NWHTCH=https://httpbin.org/post,post,HTTP-Client POST method sample test! OK</p> <p>+NWHTCDATA:225,HTTP/1.1 200 OK Date: Fri, 02 Dec 2022 01:25:38 GMT Content-Type: application/json Content-Length: 426 Connection: close Server: gunicorn/19.9.0 Access-Control-Allow-Origin: * Access-Control-Allow-Credentials: true</p> <p>+NWHTCDATA:426,{  "args": {}, "data": "HTTP-Client POST method sample test!"</p>
--	---



## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<pre> , "files": {}, "form": {}, "headers": { "Accept": "**/*", "Content-Length": "36", "Host": "httpbin.org", "User-Agent": "lwIP/2.1.2 (http://savannah.nongnu.org/projects/lwip)", "X-Amzn-Trace-Id": "Root=1-63895412-4f92fb296482283c68e2155f" }, "json": null, "origin": "124.50.108.25", "url": "https://httpbin.org/post" }  +NWHTCSTATUS:0 </pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.5.0 or later</li> <li>If <code>__SUPPORT_HTTP_CLIENT_FOR_ATCMD__</code> is enabled in the SDK, this command will be enabled</li> </ul>	
AT+NWHTCSNI	<sni>	Set the server name indication <sni>: Server name
	<p>Example</p> <pre>AT+NWHTCSNI=httpbin.org OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK</li> <li>It must be set up before connecting to the server</li> </ul>	
AT+NWHTCALPN	<alpn_number>,<alpn1>,<alpn2>,<alpn3>	Set the application layer protocol negotiation <alpn_number>: Number of alps <alpn1>: First alpn <alpn2>: Second alpn <alpn3>: Third alpn
	<p>Example</p> <pre>AT+NWHTCALPN=1,http/1.1 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK</li> <li>It must be set up before connecting to the server</li> </ul>	

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
AT+NWHTCSNIDEL	(none)	Delete the saved SNI
	Example AT+NWHTCSNIDEL OK  Note <ul style="list-style-type: none"><li>Enabled by default in the SDK</li><li>It must be set up before connecting to the server</li></ul>	
AT+NWHTCALPNDEL	(none)	Delete all saved ALPNs
	Example AT+NWHTCALPNDEL OK  Note <ul style="list-style-type: none"><li>Enabled by default in the SDK</li><li>It must be set up before connecting to the server</li></ul>	
AT+NWHTCTLSAUTH	<tls_auth_mod e>	Set the certificate verification mode.  #define MBEDTLS_SSL_VERIFY_NONE 0 #define MBEDTLS_SSL_VERIFY_OPTIONAL 1 #define MBEDTLS_SSL_VERIFY_REQUIRED 2
	Example AT+NWHTCTLSAUTH=1 OK  Note <ul style="list-style-type: none"><li>Enabled by default in the SDK</li><li>It must be set up before connecting to the server</li></ul>	

**Table 29: HTTP-Client Response List**

Response	Parameters	Description
+NWHTCSTATUS	<status>	<p>Return status along with the received payload according to the requested method.</p> <p>&lt;status&gt;: 0x00 is success See <a href="#">Appendix B</a>. For example: +NWHTCSTATUS:0x00</p>
+NWHTCDATA	<data size,>	<p>Insert size information of data received only from AT+NWHTC command.</p> <p>An integer data size and a comma are inserted, and the actual received data is after that. For example, +NWHTCDATA:426,</p>

## DA16200 DA16600 Host Interface and AT Command

### 5.6.2.1 HTTP-Client Connection Example

GET method request:

```
AT+NWHTC=https://httpbin.org/get,get
OK
HTTP/1.1 200 OK
Date: Tue, 07 Dec 2021 01:19:49 GMT
Content-Type: application/json
Content-Length: 457
Connection: keep-alive
Server: gunicorn/19.9.0
Access-Control-Allow-Origin: *
Access-Control-Allow-Credentials: true

{
  "args": {},
  "headers": {
    "Accept": "*/*",
    "Accept-Encoding": "identity",
    "Accept-Language": "ko-KR,Ko;q=0.9,en-US;q=0.8,en;q=0.7",
    "Host": "httpbin.org",
    "User-Agent": "Mozilla/5.0 (windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.186 Safari/537.36",
    "X-Amzn-Trace-Id": "Root=1-61aeb6b5-67d7324c112a7f1631adcc72"
  },
  "origin": "124.50.108.25",
  "url": "https://httpbin.org/get"
}

+NWHTCSTATUS:0x00
```

POST method request:

```
AT+NWHTC=https://httpbin.org/post,post,HTTP-Client POST method sample test!
OK
HTTP/1.1 200 OK
Date: Tue, 07 Dec 2021 01:25:05 GMT
Content-Type: application/json
Content-Length: 586
Connection: keep-alive
Server: gunicorn/19.9.0
Access-Control-Allow-Origin: *
Access-Control-Allow-Credentials: true

{
  "args": {},
  "data": "HTTP-Client POST method sample test!",
  "files": {},
  "form": {},
  "headers": {
    "Accept": "*/*",
    "Accept-Encoding": "identity",
    "Accept-Language": "ko-KR,Ko;q=0.9,en-US;q=0.8,en;q=0.7",
    "Content-Length": "36",
    "Host": "httpbin.org",
    "User-Agent": "Mozilla/5.0 (windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.186 Safari/537.36",
  }
}
```

## DA16200 DA16600 Host Interface and AT Command

```

    "X-Amzn-Trace-Id": "Root=1-61aeb7f1-341bbb8c3f3d6bc7484370e2"
  },
  "json": null,
  "origin": "124.50.108.25",
  "url": "https://httpbin.org/post"
}

+NWHTCSTATUS:0x00

```

### PUT method request:

```

AT+NWHTC=https://httpbin.org/put,put,HTTP-Client PUT method sample test!
OK
HTTP/1.1 200 OK
Date: Tue, 07 Dec 2021 02:04:19 GMT
Content-Type: application/json
Content-Length: 584
Connection: keep-alive
Server: gunicorn/19.9.0
Access-Control-Allow-Origin: *
Access-Control-Allow-Credentials: true

{
  "args": {},
  "data": "HTTP-Client PUT method sample test!",
  "files": {},
  "form": {},
  "headers": {
    "Accept": "*/*",
    "Accept-Encoding": "identity",
    "Accept-Language": "ko-KR,Ko;q=0.9,en-US;q=0.8,en;q=0.7",
    "Content-Length": "35",
    "Host": "httpbin.org",
    "User-Agent": "Mozilla/5.0 (windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.186 Safari/537.36",
    "X-Amzn-Trace-Id": "Root=1-61aec123-4c3c5d390c6b31992bb803be"
  },
  "json": null,
  "origin": "124.50.108.25",
  "url": "https://httpbin.org/put"
}

+NWHTCSTATUS:0x00

```

## 5.6.3 HTTP-Server Commands

Table 30: HTTP-Server Command List

Command	Parameters	Description
AT+NWHTS	<flag>	Start or stop the HTTP server depending on the option. <start>: 1 (start), 0 (stop) Response: OK or ERROR

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	Prerequisite DA16200/DA16600 should be connected to AP.  Example AT+NWHTS=1 OK  Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK</li> </ul>	
AT+NWHTSS	<flag>	Start or stop the HTTPS server depending on the option. <start>: 1 (start), 0 (stop) Response: OK or ERROR
	Prerequisite DA16200/DA16600 should be connected to AP.  Example AT+NWHTSS=1 OK  Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__SUPPORT_HTTP_SERVER_FOR_ATCMD__</code> is enabled in the SDK, this command will be enabled</li> </ul>	

### 5.6.3.1 HTTP/HTTPS-Server Start Example

HTTP start:

```
AT+NWHTS=1
```

HTTPS start:

```
AT+NWHTSS=1
```

### 5.6.4 WebSocket-Client Commands

**Table 31: WebSocket-Client Command List**

Command	Parameters	Description
AT+NWWSC	<operation>,<uri> > (<msg>)	Start the WebSocket client with options. <operation> connect, send, or disconnect <uri>: WebSocket server address <msg>: Request message
	Prerequisite DA16200/DA16600 should be connected to AP.  Example 1 AT+NWWSC=connect,ws://192.168.86.182:8080 +NWWSC:1	

## DA16200 DA16600 Host Interface and AT Command

	<p>Example 2</p> <pre>AT+NWWSC=send,Send Message Test OK</pre> <p>Example 3</p> <pre>AT+NWWSC=disconnect +NWWSC:0</pre> <p>Note</p> <ul style="list-style-type: none"> <li>If <code>__SUPPORT_WEBSOCKET_CLIENT__</code> is enabled in SDK, this command will be enabled</li> </ul>
--	--

**Table 32: WebSocket-Client Response List**

Response	Parameters	Description
+NWWSC	<code>&lt;status&gt;(&lt;opcode&gt;,&lt;received msg length&gt;,&lt;received msg&gt;)</code>	<p>Return status along with the received payload.</p> <p><code>&lt;status&gt;</code>  0 is disconnected.  1 is connected.</p> <p><code>&lt;opcode&gt;</code>  Continuation Frame : 0x00  Text Frame : 0x01  Binary Frame : 0x02  Close Frame : 0x08  Ping Frame : 0x09  Pong Frame : 0x0a</p> <p>Example 1: +NWWSC:1  Example 2: +NWWSC:0  Example 3: +NWWSC:1,1,12,Test Message</p>

### 5.6.5 OTA Commands

NOTE
When DPM mode enabled and OTA update is in progress (firmware download is in progress), it does not enter DPM sleep because SFLASH write operation occurs. After downloading the firmware, the DA16200 resumes to enter DPM sleep mode.

---

**DA16200 DA16600 Host Interface and AT Command****Table 33: OTA Command List**

Command	Parameters	Description
AT+NWOTADWSTART	<fw_type>,<uri> [,<fw_name>]	Start downloading firmware from an OTA server <fw_type>: Set the type of FW to be downloaded <uri>: Server URL where a FW exists <fw_name>: Optional. The maximum input size of fw_type is 7 bytes. MCU_FW will be stored by default if not specified. (Only for MCU FW) Response: +NWOTADWSTART:0x00

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
		<p>Prerequisite</p> <p>DA16200/DA16600 should be connected to AP.</p> <p>Example</p> <p>; RTOS download  AT+NWOTADWSTART=rtos,https://server/DA16200_RTOS-GEN01-01-1111-000000.img  OK  +NWOTADWSTART:0x00</p> <p>; BLE FW download (DA16600 only)  AT+NWOTADWSTART=ble_fw,https://server/ble_firmware.img  OK  +NWOTADWSTART:0x00</p> <p>; MCU FW download  AT+NWOTADWSTART=mcu_fw,https://server/mcu_firmware.img  OK  +NWOTADWSTART:0x00</p> <p>; MCU FW download (Enter the name of MCU FW within 8 characters. Default is "MCU_FW")  AT+NWOTADWSTART=mcu_fw,https://server/mcu_firmware.img,ver01  OK  +NWOTADWSTART:0x00</p> <p>; BLE FW download  AT+NWOTADWSTART=ble_fw,https://server/pxr_sr_coex_ext_531_6_0_14_1114_2_ota.img  OK  +NWOTADWSTART:0x00</p> <p>; Cert Key download:  AT+NWOTADWSTART=cert_key,https://server/ca.pem  OK  +NWOTADWSTART:0x00</p> <p>Note</p> <ul style="list-style-type: none"> <li>• Enabled by default in the SDK</li> <li>• The parameter, fw_type should be lowercase</li> <li>• BLE FW download is available on SDK V3.2.3.0 or later</li> </ul>
AT+NWOTARENEW	(none)	Reboot with updated FW



## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	Prerequisite Download RTOS or BLE images.  Example AT+NWOTARENEW +NWOTARENEW:0x00  Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK</li> <li>Will reboot automatically after renewing is completed</li> <li>BLE image is supported in SDK V3.2.3.0 or later, either RTOS or BLE, or both of them can be supported</li> </ul>	
AT+NWOTADWPROG	<fw_type>  Example ; RTOS download progress: AT+NWOTADWPROG=rtos +NWOTADWPROG:100 OK  ; MCU FW download progress: AT+NWOTADWPROG=mcu_fw +NWOTADWPROG:100 OK  ; BLE FW download progress (DA16600 only): AT+NWOTADWPROG=ble_fw +NWOTADWPROG:100 OK  ; Cert Key download progress: AT+NWOTADWPROG=cert_key +NWOTADWPROG:100 OK  Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK</li> </ul>	FW download progress. <fw_type>: Set a firmware type among rtos, ble_fw, mcu_fw, or cert_key Response: +NWOTADWPROG:100
AT+NWOTADWSTOP	(none)  Example AT+NWOTADWSTOP OK	Stop while downloading FW
AT+NWOTAFWNAME	(none)	Read a name in the header of the MCU firmware (Only for MCU FW)

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre>AT+NWOTAFWNAME +NWOTAFWNAME:MCU OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__OTA_UPDATE_MCU_FW__</code> is enabled in the SDK, this command will be enabled</li> </ul>	
AT+NWOTAFWSIZE	(none)	Read a size in the header of the MCU firmware (Only for MCU FW)
	<p>Example</p> <pre>AT+NWOTAFWSIZE +NWOTAFWSIZE:4128 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__OTA_UPDATE_MCU_FW__</code> is enabled in the SDK, this command will be enabled</li> </ul>	
AT+NWOTAFWCRC	(none)	Read a CRC in the header of the MCU firmware (Only for MCU FW)
	<p>Example</p> <pre>AT+NWOTAFWCRC +NWOTAFWCRC:5aa8b6c4 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__OTA_UPDATE_MCU_FW__</code> is enabled in the SDK, this command will be enabled</li> </ul>	
AT+NWOTAREADFW	<read_addr>, <read_size>	<p>Read the MCU firmware as much as the read_size from the read_addr and transmit it (Only for MCU FW)</p> <p>&lt;read_addr&gt;: Hexadecimal without "0x" prefix &lt;read_size&gt;: Decimal</p> <p>MCU_FW default address</p> <ul style="list-style-type: none"> <li>DA16200 : 0x003A_D000</li> <li>DA16600 : 0x003C_2000</li> </ul>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre>AT+NWOTAREADFW=3ad000,128 DA16FMCUyyy123456789012345612345678901234561234567890123 456123456789012345612345678901234561234567890123456 +NWOTAREADFW:COMPLETE OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__OTA_UPDATE_MCU_FW__</code> is enabled in the SDK, this command will be enabled</li> </ul>	
AT+NWOTATRANSFW	(none)	Transmit the downloaded MCU firmware to the MCU. Transmission will be failed if no header (16 bytes) information exist (Only for MCU FW)
	<p>Example</p> <pre>AT+NWOTATRANSFW MCU DA16FMCUyyy123456789012345612345678901234561234567890123 4561234567890123456123456789012345612345678901234567 89012345612345678901234 ..... +NWOTATRANSFW:COMPLETE OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__OTA_UPDATE_MCU_FW__</code> is enabled in the SDK, this command will be enabled</li> </ul>	
AT+NWOTAERASEFW	(none)	Erase the MCU firmware stored in a serial flash of DA16200/DA16600. (Only for MCU FW)
	<p>Example</p> <pre>AT+NWOTAERASEFW +NWOTAERASEFW:COMPLETE OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__OTA_UPDATE_MCU_FW__</code> is enabled in the SDK, this command will be enabled</li> </ul>	
AT+NWOTASETADDR	<sflash_addr>	<p>Designate an address where data can be downloaded within the range of User Area and TLS Certificate Key in the SFLASH area.</p> <p>MCU_FW / CERT_KEY default address</p> <ul style="list-style-type: none"> <li>DA16200 : 0x003A_D000</li> <li>DA16600 : 0x003C_2000</li> </ul> <p>CERT_KEY should be copied to the operating area if downloaded to the user area, which is the default address.</p> <p>CERT_KEY area address</p> <ul style="list-style-type: none"> <li>DA16200 / DA16600 : 0x003A_3000</li> </ul>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre>AT+NWOTASETADDR=3ad000 +NWOTASETADDR:0x00 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK</li> </ul>	
AT+NWOTAGETADDR	<fw_type>	<p>Return the value set with NWOTASETADDR. MCU_FW / CERT_KEY default address</p> <ul style="list-style-type: none"> <li>DA16200 : 0x003A_D000</li> <li>DA16600 : 0x003C_2000</li> </ul>
	<p>Example</p> <pre>AT+NWOTAGETADDR=mcu_fw +NWOTAGETADDR:3ad000 OK</pre> <pre>AT+NWOTAGETADDR=cert_key +NWOTAGETADDR:3ad000 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK</li> </ul>	
AT+NWOTAREADFLASH	<sflash_addr>,<size>	<p>Read as much as size from <i>sflash_addr</i>. MCU_FW default address</p> <ul style="list-style-type: none"> <li>DA16200 : 0x003A_D000</li> <li>DA16600 : 0x003C_2000</li> </ul>
	<p>Example</p> <pre>AT+NWOTAREADFLASH=3ad000,128 MCU_FW ?"ZDA16FMCUÿÿÿÿ123456789012345612345678901234561234567890 123456123456789012345612345678901234561234567890123456 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK</li> </ul>	
AT+NWOTAERASEFLASH	<sflash_addr>,<size>	<p>Delete as much as size from <i>sflash_addr</i>. MCU_FW default address</p> <ul style="list-style-type: none"> <li>DA16200 : 0x003A_D000</li> <li>DA16600 : 0x003C_2000</li> </ul>
	<p>Example</p> <pre>AT+NWOTAERASEFLASH=3ad000,1000 +NWOTAERASEFLASH:COMPLETE OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK</li> <li>It will be erased in 4 kB increments</li> </ul>	

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
AT+NWOTACOPYFLASH	<dest_sflash_addr>,<src_sflash_addr>,<size>	Copy as much as size from src_sflash_addr to dest_sflash_addr. MCU_FW default address - DA16200 : 0x003A_D000 - DA16600 : 0x003C_2000
	<p>Example</p> <pre>AT+NWOTACOPYFLASH=3c2000,3ad000,1000 +NWOTACOPYFLASH:COMPLETE OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK</li> <li>It will be copied in 4 kB increments</li> </ul>	
AT+NWOTATLSAUTH	<tls_auth_mode>	Set the certificate verification mode. #define MBEDTLS_SSL_VERIFY_NONE 0 #define MBEDTLS_SSL_VERIFY_OPTIONAL 1 #define MBEDTLS_SSL_VERIFY_REQUIRED 2
	<p>Example</p> <pre>AT+NWOTATLSAUTH=1 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.5.0 or later</li> </ul>	
AT+ NWOTABYMCU	rtos,<full_size>	Transmit the RTOS stored in the MCU to the DA16200/DA16600 when there is no network access. If the AT+NWOTABYMCU= rtos,<full_size> command is OK, RTOS is transmitted as much as the partial size with tx_size=<partial_size>,<binary data>. Every transmission sends "OK" as a response unless there is an error. If an error occurs or transmission is complete, it responds with +NWOTABYMCU:0x00 including status. Note that only RTOS can be downloaded.
	<p>Example</p> <pre>AT+NWOTABYMCU=rtos,1335408 OK tx_size=4096,4643394BDA4F27840000000000000000... OK ... tx_size=112,000000000000000000000000000000001000000... +NWOTABYMCU:0x00</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.5.0 or later</li> <li>If __OTA_UPDATE_MCU_FW__ is enabled in the SDK, this command will be enabled</li> </ul>	

## DA16200 DA16600 Host Interface and AT Command

### NOTE

When DPM mode is enabled and OTA update is in progress (firmware download is in progress), it does not enter DPM sleep due to SFLASH write operation. After downloading the firmware, the DA16200 resumes to enter DPM sleep mode.

**Table 34: OTA Response List**

Response	Parameters	Description
+NWOTADWSTART	<status>	Return the status of FW download. <status>: 0x00 is success. See <a href="#">Table 35</a> for other status value. For example: +NWOTADWSTART:0x00
+NWOTARENEW	<status>	Return the status for FW RENEW. <status>: 0x00 is success. See OTA Response Code List for others For example: +NWOTARENEW:0x00
+NWOTADWPROG	<progress>	Return the percentage value (%) of the FW download progress. <progress>: Print download progress in percent For example: +NWOTADWPROG:100
+NWOTADWSTOP	<status>	Return the status of FW download stop. <status>: 0x00 is success. See <a href="#">Table 35</a> for other status values For example: +NWOTADWSTOP:0x00
+NWOTATRANSFW	COMPLETE or FAIL	Return result of MCU FW transmission. (Only for MCU FW) For example: +NWOTATRANSFW:COMPLETE
+NWOTAFWNAME	<name>	String entered by a user. (Default is MCU_FW) Returns "(NULL)" if there is no MCU FW. (Only for MCU FW)
+NWOTAFWSIZE	<size>	Downloaded MCU FW size. It returns 0 if there is no MCU FW. (Only for MCU FW)
+NWOTAFWCRC	<crc>	Downloaded MCU FW CRC. It returns 0 if there is no MCU FW. (Only for MCU FW)
+NWOTAREADFW	COMPLETE or FAIL	Success: COMPLETE Failure: FAIL (Only for MCU FW)
+NWOTAERASEFW	COMPLETE or FAIL	Success: COMPLETE Failure: FAIL (Only for MCU FW)
+NWOTASETADDR	<status>	<status>: 0x00 is success See <a href="#">Table 35</a> for other status values.
+NWOTAGETADDR	<sflash_addr>	Return the value of sflash_addr.
(AT+NWOTAREADFLASH)	(Binary)	Return binary data as much as entered SFLASH address and size.

## DA16200 DA16600 Host Interface and AT Command

Response	Parameters	Description
+NWOTAERASEFLASH	COMPLETE or FAIL	Success: COMPLETE Failure: FAIL
+NWOTACOPYFLASH	COMPLETE or FAIL	Success: COMPLETE Failure: FAIL
+NWOTABYMCU	<status>	<status>: 0x00 is success See <a href="#">Table 35</a> for other status values.

**Table 35: OTA Response Code List**

Return Value	Description
0x00	Return success.
0x01	Return fail.
0x02	SFLASH address is wrong.
0x03	FW type is unknown.
0x04	Server URL is unknown.
0x05	FW size is too big.
0x06	CRC is not correct.
0x07	FW version is unknown.
0x08	FW version is incompatible.
0x09	FW not found on the server.
0x0A	Failed to connect to the server.
0x0B	All new FWs have not been downloaded.
0x0C	Failed to allocate memory.
0xA1	BLE FW version is unknown.

### 5.6.5.1 OTA Download Example

RTOS download:

```
AT+NWOTADWSTART=rtos,https://server/DA16200_RTOS-GEN01-01-1111-000000.img
```

BLE FW download: (DA16600 only)

```
AT+NWOTADWSTART=ble_fw,https://server/ble_firmware.img
```

MCU FW download:

```
AT+NWOTADWSTART=mcu_fw,https://server/mcu_firmware.img
AT+NWOTADWSTART=mcu_fw,https://server/mcu_firmware.img,ver01
```

Cert Key download:

```
AT+NWOTADWSTART=cert_key,https://server/ca.pem
```

## DA16200 DA16600 Host Interface and AT Command

### 5.6.5.2 OTA Download Progress Example

RTOS download progress:

```
AT+NWOTADWPROG=rtos
```

BLE FW download progress: (DA16600 only)

```
AT+NWOTADWPROG=ble_fw
```

MCU FW download progress:

```
AT+NWOTADWPROG=mcu_fw
```

Cert Key download progress:

```
AT+NWOTADWPROG=cert_key
```

### 5.6.5.3 OTA Renew Example

Renew Firmware (reboot with updated FW):

```
AT+NWOTARENEW
```

### 5.6.5.4 MCU FW Transport Example

MCU FW transmission:

```
AT+NWOTATRANSFW
```

Get MCU FW name:

```
AT+NWOTAFWNAME
```

Get MCU FW size:

```
AT+NWOTAFWSIZE
```

Get MCU FW CRC:

```
AT+NWOTAFWCRC
```

Read MCU FW as much as specified size:

```
AT+NWOTAREADFW=3ad000,128
```

Delete MCU FW stored in the DA16200/DA16600 SFLASH:

```
AT+NWOTAERASEFW
```

### 5.6.5.5 SFLASH User Area Address Setting Example

SET ADDR:

```
AT+NWOTASETADDR=3ad000
```

GET ADDR:

```
AT+NWOTAGETADDR=mcu_fw
```

```
AT+NWOTAGETADDR=cert_key
```



## DA16200 DA16600 Host Interface and AT Command

### 5.6.5.6 SFLASH READ/COPY/ERASE Example

SFLASH Read:

```
AT+NWOTAREADFLASH=3ad000,128
```

SFLASH Copy:

```
AT+NWOTACOPYFLASH=3ad000,0x3c2000,128
```

SFLASH Erase:

```
AT+NWOTAERASEFLASH=0x1f2000,128
```

### 5.6.5.7 TLS Certificate verification mode Setting Example

SET MBEDTLS\_SSL\_VERIFY\_NONE:

```
AT+NWOTATLSAUTH=0
```

SET MBEDTLS\_SSL\_VERIFY\_OPTIONAL:

```
AT+NWOTATLSAUTH=1
```

SET MBEDTLS\_SSL\_VERIFY\_REQUIRED:

```
AT+NWOTATLSAUTH=2
```

### 5.6.5.8 RTOS by MCU download Example

Initialization:

```
AT+NWOTABYMCU=rtos,1335408
```

Data transmission:

```
tx_size=4096,4643394BDA4F27840000000000000000...
```

## 5.6.6 Zeroconf Commands

Table 36: Zeroconf Command List

Command	Parameters	Description
AT+NWMDNSSTART	<Mode>	Start the mDNS module. The mDNS module is communicated through multicast. DA16200/DA16600 could frequently be changed from/to DPM sleep and wake-up states. It may consume more power. <Mode>: The mode in which the WLAN interface is running, 0 (Station) or 1 (Soft AP).

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	Prerequisite DA16200/DA16600 should be connected to AP. The host name of mDNS module should be set up.  Example AT+NWMDNSSTART=1 OK  Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If __SUPPORT_ZERO_CONFIG__ is enabled in the SDK, this command will be enabled.</li> </ul>	
	?	Get the string representing the status of mDNS module, "started" or "not started".
	Example AT+NWMDNSSTART=? +NWMDNSSTART:started	
AT+NWMDNSHNREG	<Host name>	Register the host name in the mDNS module. mDNS supports one configured host name only, to change or set a new mDNS host name. mDNS service must be stopped and started again.  <Host name>: The name of the host to be registered.
	Example AT+NWMDNSSHNREG=da16x OK  Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If __SUPPORT_ZERO_CONFIG__ is enabled in the SDK, this command will be enabled.</li> </ul>	
AT+NWMDNSSRVREG	<Instance name>,<Protocol>, <Port>[,<Text record>]	Register a service in the mDNS module. <Instance name>: The instance name of service to be registered. <Protocol>: The protocol and the type of the service to be registered. <Port>: The port number of the service to be registered. <Text record>: The text record of the service that must be registered and mentioned in "Key=Value" format. Multiple pairs of text records should be separated using a ",".

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Prerequisite</p> <p>DA16200/DA16600 should be connected to AP. The host name of mDNS module should be set up.</p> <p>Example</p> <pre>AT+NWMDNSSRVREG=_WEBAPP,_http,_tcp,80,LIGHT=OFF,FAN=ON OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__SUPPORT_ZERO_CONFIG__</code> is enabled in the SDK, this command will be enabled.</li> </ul>	
AT+NWMDNSSUPDATETXT	<Text record>	<p>Update the text record of a service in the mDNS module.</p> <p>&lt;Text record&gt;: The text record of the service to be updated.</p>
	<p>Example</p> <pre>AT+NWMDNSSUPDATETXT=LIGHT=OFF,FAN=ON OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__SUPPORT_ZERO_CONFIG__</code> is enabled in the SDK, this command will be enabled.</li> </ul>	
AT+NWMDNSSRVDEREG	(None)	<p>Unregister a service in the mDNS module.</p> <p>Response: OK or ERROR</p> <p>For example: AT+NWMDNSSRVDEREG</p>
	<p>Prerequisite</p> <p>DA16200/DA16600 should be connected to AP. The mDNS module should be running. The service in the mDNS module should be registered.</p> <p>Example</p> <pre>AT+NWMDNSSRVDEREG OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__SUPPORT_ZERO_CONFIG__</code> is enabled in the SDK, this command will be enabled.</li> </ul>	
AT+NWMDNSSTOP	(None)	Stop the mDNS module.

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	Prerequisite DA16200/DA16600 should be connected to an AP. The mDNS module should be running.  Example AT+NWMDNSSTOP OK  Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If __SUPPORT_ZERO_CONFIG__ is enabled in the SDK, this command will be enabled.</li> </ul>	
	?	Get the string representing the status of the mDNS module, "stopped" or "running".
	Example AT+NWMDNSSTOP +NWMDNSSTOP:stopped	

### 5.6.6.1 Zeroconf Example

Configure the parameters and start the mDNS module (After Wi-Fi Connection):

```
AT+NWMDNSHNREG=da16x
```

```
AT+NWMDNSSTART=0
```

If the mDNS module is started successfully, the following response is shown:

```
OK
```

Register a service in the mDNS module:

```
AT+NWMDNSSRVREG=_WEBAPP,_http._tcp,80,LIGHT=OFF,FAN=ON
```

If the service is registered successfully, the following response is shown:

```
OK
```

Registered service and host name can be discovered by other mDNS services. In this example, Bonjour service ([https://support.apple.com/kb/DL999?viewlocale=en\\_US&locale=zh\\_TW](https://support.apple.com/kb/DL999?viewlocale=en_US&locale=zh_TW)) on Windows is used to discover them.

To discover DA16200/DA16600's mDNS, the "-G" option can be used like the following:

```
C:> dns-sd -G v4 da16x.local

Timestamp      A/R  Flags if  Hostname                Address
TTL
18:04:04.474   Add    2 24  da16x.local.           192.168.0.4
120
```

To discover the service, the "-L" option can be used like the following:

## DA16200 DA16600 Host Interface and AT Command

```
C:>dns-sd -L _WEBAPP._http._tcp local
Lookup _WEBAPP._http._tcp.local
18:04:29.453 _WEBAPP._http._tcp.local. can be reached at da16x.local.:80
(interface 24)

LIGHT=OFF FAN=ON
```

### 5.6.7 Provision Commands

The provision commands are used for starting to provision procedure and getting provisioning status. To use these commands, `__PROVISION_ATCMD__` feature should be enabled in SDK.

**Table 37: Provision Command List**

Command	Parameters	Description
AT+PROVSTART	(none)	Removed all profile data in NVRAM and configure a new station or Soft AP profile is set Response: ERROR
	<p>Example</p> <pre>AT+PROVSTART +INIT:DONE,1</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__PROVISION_ATCMD__</code> is enabled in the SDK, this command will be enabled</li> <li>In case of DA16200/DA16600, <code>__SUPPORT_FACTORY_RST_APMODE__</code> should be enabled. For DA16600, <code>__SUPPORT_FACTORY_RST_STAMODE__</code> should be used</li> <li>After restarting, the system will be ready for the provision procedure</li> </ul>	
AT+PROVSTAT	(none)	Get status of provisioning Response: OK or ERROR
	<p>Example</p> <pre>AT+PROVSTAT +ATPROV=STATUS 1 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__PROVISION_ATCMD__</code> is enabled in the SDK, this command will be enabled</li> <li>The list of provision status can be found in the thread <code>_atcmd.h</code>. Check the <code>atcmd_provision_stat</code> enumeration</li> </ul>	

**Table 38: atcmd\_provision\_stat Enumeration**

Value	Name	Description
0	ATCMD_PROVISION_IDLE	Not run or finish
1	ATCMD_PROVISION_START	Start

## DA16200 DA16600 Host Interface and AT Command

Value	Name	Description
101	ATCMD_PROVISION_SELECTED_AP_SUCCESS	Receive AP information
102	ATCMD_PROVISION_SELECTED_AP_FAIL	
103	ATCMD_PROVISION_WRONG_PW	AP connection fail by the wrong PW
104	ATCMD_PROVISION_NETWORK_INFO	Get Network info. from Mobile App
105	ATCMD_PROVISION_AP_FAIL	AP connection fail
106	ATCMD_PROVISION_DNS_FAIL_SERVER_FAIL	Network connection check
107	ATCMD_PROVISION_DNS_FAIL_SERVER_OK	
108	ATCMD_PROVISION_NO_URL_PING_FAIL	
109	ATCMD_PROVISION_NO_URL_PING_OK	
110	ATCMD_PROVISION_DNS_OK_PING_FAIL_N_SERVER_OK	
113	ATCMD_PROVISION_DNS_OK_PING_N_SERVER_FAIL	
111	ATCMD_PROVISION_DNS_OK_PING_OK	Reboot after provisioning
112	ATCMD_PROVISION_REBOOT_ACK	

### 5.6.8 Bluetooth® LE Commands

The Bluetooth® LE commands are available when the DA16600 is running the Bluetooth® LE enabled version of the firmware. Select and build the DA16600 project from the SDK to use these commands.

**Table 39: Bluetooth® LE Command List**

Command	Parameters	Description
AT+BLENAM	<device name>	Change the device name of the DA16600/Bluetooth® LE device. Response: OK or ERROR
	?	Get the device name of the DA16600/Bluetooth® LE device.
	(none)	Response: +BLENAM:<device name>
	Example <pre>AT+BLENAM=DA16600-BLE OK  AT+BLENAM +BLENAM:DA16600-BLE OK AT+BLENAM=? +BLENAM:DA16600-BLE</pre> Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>Enabled getting the device name by default in the SDK v3.2.5.0 or later</li> </ul>	
AT+ADVSTOP	(none)	Stop advertising of DA16600 / Bluetooth® LE device. Response: OK or ERROR

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	Example AT+ADVSTOP OK  Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> </ul>	
AT+ADVSTART	(none)	Start advertising of DA16600 / Bluetooth® LE device. Response: OK or ERROR
	Example AT+ADVSTART OK  Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> </ul>	

### 5.6.9 Transfer Function Commands

#### 5.6.9.1 Socket Commands

Table 40: Socket Command List

Command	Parameters	Description
AT+TRTS	<local_port>[,<max allowed connection>]	Open a TCP server socket. <local_port>: Local port number of the socket. <max allowed connection>: It is optional. Set max allowed TCP session. Response: OK (with '+TRCTS:***' See <a href="#">Table 41</a> or ERROR. Async message: CID(+TRTS:<Assigned CID>)

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Prerequisite</p> <p>DA16200/DA16600 should be connected to AP.</p> <p>Example</p> <pre>; Open first TCP server AT+TRTS=10194 +TRTS:0 OK  ; Open second TCP server AT+TRTS=10195 +TRTS:3 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Enabled by default in the SDK</li> <li>• CID number 0 (TCP server), 1 (TCP client), 2 (UDP) are pre-assigned number. New CID started from 3 is assigned on the order of opening TCP session</li> <li>• The &lt;max allowed connection&gt; parameter is supported by SDK v3.2.5.0 or later</li> <li>• Multiple TCP server and async message with assigned CID are supported by SDK v3.2.5.0 or later. Total eight sessions can be created for transfer function. But it depends on DA16200/DA16600 SDK configuration. Basically, DA16200/DA16600 can be created eight TCP sockets</li> </ul>	
AT+TRTC	<p>&lt;server_ip&gt;, &lt;server_port&gt;[, &lt;local_port&gt;]</p>	<p>Open a TCP client socket and connect to a TCP server.</p> <p>&lt;server_ip&gt;: IP address of TCP server to be accessed</p> <p>&lt;server_port&gt;: Port number of TCP server</p> <p>&lt;local_port&gt;: Local port number of the socket (optional, 0: auto)</p> <p>Response: OK or ERROR</p> <p>Async message: CID(+TRTC:&lt;Assigned CID&gt;)</p>
	<p>Prerequisite</p> <p>DA16200/DA16600 should be connected to AP.</p> <p>Example</p> <pre>AT+TRTC=192.168.0.18,1025,1024 +TRTC:1 OK  AT+TRTC=192.168.0.18,1025,1025 +TRTC:3 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Enabled by default in the SDK</li> <li>• CID number 0 (TCP server), 1 (TCP client), 2 (UDP) are pre-assigned number. New CID started from 3 is assigned on the order of opening TCP session.</li> <li>• Multiple TCP client and async message with assigned CID are able to be supported by SDK v3.2.5.0 or later. Total eight sessions can be created for transfer function. But it depends on DA16200/DA16600 SDK configuration. Basically, DA16200/DA16600 can be created eight TCP sockets</li> </ul>	



## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
AT+TRUSE	<local_port>	Open a UDP socket. <local_port>: Local port number of the socket Response: OK or ERROR Async message: CID(+TRUSE:<Assigned CID>)
	<p>Example</p> <pre>AT+TRTALL (optional, run this first if 'ERROR' is responded) AT+TRUSE=10195 +TRUSE:2 OK  AT+TRUSE=10196 +TRUSE:3 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Enabled by default in the SDK</li> <li>• CID number 0 (TCP server), 1 (TCP client), 2 (UDP) are pre-assigned number. New CID started from 3 is assigned on the order of opening TCP session</li> <li>• Multiple UDP socket and async message with assigned CID are able to be supported by SDK v3.2.5.0 or later. Total eight sessions can be created for transfer function. But it depends on DA16200/DA16600 SDK configuration. Basically, DA16200/DA16600 can be created eight UDP sockets</li> </ul>	
AT+TRUR	<remote_ip>, <remote_port>	Set remote IP and port of the UDP socket. <remote_ip>: Remote IP address <remote_port>: Remote port number Response: OK or ERROR Async message: CID(+TRUR:2)
	<p>Example</p> <pre>AT+TRUR=192.168.0.18,1027 +TRUR:2 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Enabled by default in the SDK</li> <li>• It is only for the CID 2</li> </ul>	
AT+TRPRT	<cid>	Get session information by CID <cid>: Assigned CID Response: <cid>,[TCP UDP],<remote_ip>,<remote_port>,<local_port>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Prerequisite</p> <p>A UDP socket should be opened (AT+TRUSE).</p> <p>Example</p> <pre>AT+TRPRT=2 +TRPRT:2,UDP,192.168.0.18,10194,10195 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK</li> <li>CID number 0 (TCP server), 1 (TCP client), and 2 (UDP) are pre-assigned number. New CID started from 3 is assigned on the order of opening TCP session</li> </ul>	
AT+TRPALL	(none)	<p>Get all session information</p> <p>Response: &lt;cid&gt;,[TCP UDP],&lt;remote_ip&gt;,&lt;remote_port&gt;,&lt;local_port&gt;&lt;LF&gt;...</p>
	<p>Prerequisite</p> <p>The target system should be connected to any UDP or TCP server.</p> <p>Example</p> <pre>AT+TRPALL +TRPALL:2,UDP,192.168.0.18,10194,10195 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK</li> </ul>	
AT+TRTRM	<p>&lt;cid&gt;</p> <p>[,&lt;remote_ip&gt;,&lt;remote_port&gt;]</p>	<p>Close (terminate) a session by CID. If CID is 0 (TCP server), remote_ip, and remote_port are input, the session with the specific remote will be closed.</p> <p>&lt;cid&gt;: Assigned CID</p> <p>&lt;remote_ip&gt;: Remote IP address connected to TCP server. It is only allowed in TCP server</p> <p>&lt;remote_port&gt;: Remote port number connected to TCP server. It is only allowed in TCP server</p> <p>Response: OK or ERROR</p>
	<p>Prerequisite</p> <p>The target system should be connected to any UDP or TCP server.</p> <p>Example</p> <pre>AT+TRTRM=2 OK  AT+TRTRM=0,192.168.0.18,10194 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK</li> <li>The remote_ip and remote_port parameters are only supported in SDK v3.2.3.0 or later</li> </ul>	

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
AT+TRTALL	(none)	Close (terminate) all sessions Response: OK or ERROR
	Example <pre>AT+TRTALL OK</pre> Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK</li> </ul>	
AT+TRSAVE	(none)	Save status of all sessions to NVRAM Response: OK or ERROR
	Example <pre>AT+TRSAVE OK</pre> Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK</li> </ul>	
AT+TCPDATAMODE	<mode>	Set the mode of the received TCP data <mode> 0: text mode (default) 1: hex string mode In text mode, data is returned as an ascii string: "0123ABCD" In hex mode, the data is returned as a hex encoded text string "3031323341424344" Response: OK or ERROR
	Example <pre>AT+TCPDATAMODE=1 OK</pre> Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.2.1 or later</li> </ul>	

**Table 41: Socket Connection Response List**

Response	Parameters	Description
+TRCTS	<cid>, <remote_ip>, <remote_port>	When sending the AT command (AT+TRTS=40000), receive this response if there is no error <cid>: Assigned CID for TCP server <remote_ip>: TCP client IP address <remote_port>: TCP client port number
	Example <pre>+TRCTS:0,192.168.0.18,41014</pre>	
+TRXTS	<cid>, <remote_ip>, <remote_port>	A remote TCP client is disconnected from the TCP server that was opened by AT+TRTS <cid>: Assigned CID for TCP server <remote_ip>: TCP client IP address <remote_port>: TCP client port number

## DA16200 DA16600 Host Interface and AT Command

Response	Parameters	Description
	Example ; When a remote peer is disconnected +TRXTS:0,192.168.0.18,41014	
+TRXTC	<cid>, <remote_ip>, <remote_port>	The TCP client socket that was opened by AT+TRTC is disconnected <cid>: Assigned CID for TCP client <remote_ip>: TCP server IP address <remote_port>: TCP server port number
	Example ; When the TCP client socket is disconnected +TRXTC:1,192.168.0.18,1025	

### 1) Data Transfer Commands

Table 42: Data Transmission Command

Escape Sequence	Parameters	Description
<ESC>S	<cid><length>, <remote_ip >, <remote_port >,<mode>,<data>	<p>Transmit data through a socket with the CID specified.</p> <p>&lt;ESC&gt;S: To enter data input mode, type in &lt;ESC&gt; (0x1B) and S keys together.</p> <p>&lt;cid&gt;: Assigned CID</p> <p>&lt;length&gt;: Data length. Data length can be 0 in only text mode. If this is 0, data is read until "\r" or "\n" is met. In raw mode, data is read until the length.</p> <p>&lt;remote_ip&gt;: Remote IP address</p> <p>&lt;remote_port&gt;: Remote port number</p> <ul style="list-style-type: none"> <li>For TCP Server, &lt;remote_ip&gt; and &lt;remote_port&gt; of a TCP Client should be given. Maximum four TCP Clients can be connected to the TCP Server</li> <li>For TCP Client, 0, 0 is given (as the destination is the server)</li> <li>For UDP: if 0,0 is given, the data is sent to the destination that AT+TRUR has specified. if non-0 &lt;remote_ip&gt; and &lt;remote_port&gt; are given, UDP temporarily sends to the destination &lt;remote_ip&gt; and &lt;remote_port&gt; specifies</li> </ul> <p>&lt;mode&gt;: Mode to transmit data in raw or text mode. It is optional. If there is no option, data will be transmitted in text mode. This option is allowed only for UART communication.</p> <ul style="list-style-type: none"> <li>r: The raw mode is active. In raw mode, Data is read until data length. The data length is specified in &lt;length&gt; parameter</li> <li>t: The text mode is active. In text mode, the data can be affected if it has unprintable control codes like backspace(0x08)</li> </ul> <p>Response: OK or ERROR</p>

## DA16200 DA16600 Host Interface and AT Command

Escape Sequence	Parameters	Description
	<p>Prerequisite</p> <p>The target system should be connected to any UDP or TCP server/client.</p> <p>Example1 – To send data to TCP client            &lt;ESC&gt;S010,192.168.0.18,43110,abcde12345            OK</p> <p>Example2 – To send data to TCP server            &lt;ESC&gt;S110,192.168.0.18,1025,abcde12345            OK</p> <p>Example3 – To send data to TCP server with '0, 0' as the destination/server            &lt;ESC&gt;S110,0,0,abcde12345            OK</p> <p>Example4 – To send data to UDP receiver            &lt;ESC&gt;S210,192.168.0.18,1024,abcde12345            OK</p> <p>Note</p> <ul style="list-style-type: none"> <li>• Enabled by default in the SDK</li> <li>• The maximum length of data depends on TX_PAYLOAD_MAX_SIZE definition. It is defined in atcmd.h for SDK v3.x. TX_PAYLOAD_MAX_SIZE includes all parameters of AT command. Therefore, the maximum length of 'length' parameter depends on length of other parameters</li> <li>• TX_PAYLOAD_MAX_SIZE is defined 4,096 bytes in the SDK v3.2.3.0 or later</li> <li>• For ATCMD over SPI or SDIO:            The result for this command is sent to the host as the "response" field of "struct _st_host_response". The "response" field is a 1-byte decimal value. A value of 0x20 is a result of "OK". All other values are an "ERROR"</li> <li>• Recommended to use &lt;ESC&gt;H command if UART baud rate is over 230400 bps. There might be data loss during DA16200/DA16600 parses this AT-CMD</li> </ul>	
<ESC>M	<cid>,<length>, <remote_ip >, <remote_port >,<mode>,<data>	<p>Transmit data through a socket with the CID specified.</p> <p>&lt;ESC&gt;M: To enter data input mode, type in &lt;ESC&gt;(0x1B) and M key together</p> <p>&lt;cid&gt;: Assigned CID</p> <p>&lt;length&gt;: Data length. Data length can be 0 in only text mode. If this is 0, data is read until "\r" or "\n" is met. In raw mode, data is read until the length</p> <p>&lt;remote_ip&gt;: Remote IP address</p> <p>&lt;remote_port&gt;: Remote port number</p> <ul style="list-style-type: none"> <li>• For TCP Server, &lt;remote_ip&gt; and &lt;remote_port&gt; of a TCP Client should be given</li> <li>• For TCP Client, 0, 0 is given (as the destination is the server)</li> <li>• For UDP: if 0,0 is given, the data is sent to the destination that AT+TRUR has specified. if non-0 &lt;remote_ip&gt; and &lt;remote_port&gt; are given, UDP temporarily sends to the destination &lt;remote_ip&gt; and &lt;remote_port&gt; specifies</li> </ul> <p>Response: OK or ERROR</p>

## DA16200 DA16600 Host Interface and AT Command

Escape Sequence	Parameters	Description
	<p>Prerequisite</p> <p>The target system should be connected to any UDP or TCP server/client.</p> <p>Example1 – To send data to TCP client &lt;ESC&gt;M0,10,192.168.0.18,43110,abcde12345 OK</p> <p>Example2 – To send data to TCP server &lt;ESC&gt;M1,10,192.168.0.18,1025,abcde12345 OK</p> <p>Example3 – To send data to TCP server with '0, 0' as the destination/server &lt;ESC&gt;M1,10,0,0,abcde12345 OK</p> <p>Example4 – To send data to UDP receiver &lt;ESC&gt;M2,10,192.168.0.18,1024,abcde12345 OK</p> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK</li> <li>The maximum length of data depends on TX_PAYLOAD_MAX_SIZE definition. It is defined in atcmd.h for SDK v3.x, TX_PAYLOAD_MAX_SIZE includes all parameters of AT command. Therefore, the maximum length of 'length' parameter depends on length of other parameters</li> <li>TX_PAYLOAD_MAX_SIZE is defined 4,096 bytes in the SDK v3.2.3.0 or later</li> <li>For ATCMD over SPI or SDIO: The result for this command is sent to the host as the "response" field of "struct _st_host_response". The "response" field is a 1-byte decimal value. A value of 0x20 is a result of "OK". All other values are an "ERROR"</li> <li>Recommended to use &lt;ESC&gt;H command if UART baud rate is over 230400 bps. There might be data loss during DA16200/DA16600 parses this AT-CMD</li> </ul>	
<ESC>H	<p>&lt;cid&gt;,&lt;length&gt;, &lt;remote_ip &gt;, &lt;remote_port &gt;</p>	<p>Transmit data through a socket with the CID specified. Host must send data after getting response OK.</p> <p>&lt;ESC&gt;H: To enter data input mode, type in &lt;ESC&gt;(0x1B) and H key together</p> <p>&lt;cid&gt;: Assigned CID</p> <p>&lt;length&gt;: Data length. Data is read until the length after getting OK response</p> <p>&lt;remote_ip&gt;: Remote IP address</p> <p>&lt;remote_port&gt;: Remote port number</p> <ul style="list-style-type: none"> <li>For TCP Server, &lt;remote_ip&gt; and &lt;remote_port&gt; of a TCP Client should be given</li> <li>For TCP Client, 0, 0 is given (as the destination is the server)</li> <li>For UDP: if 0,0 is given, the data is sent to the destination that AT+TRUR has specified. if non-0 &lt;remote_ip&gt; and &lt;remote_port&gt; are given, UDP temporarily sends to the destination &lt;remote_ip&gt; and &lt;remote_port&gt; specifies</li> </ul> <p>Response: OK or ERROR</p>

## DA16200 DA16600 Host Interface and AT Command

Escape Sequence	Parameters	Description
	<p>Prerequisite</p> <p>The target system should be connected to any UDP or TCP server/client.</p> <p>Example1 – To send data to TCP client</p> <pre>&lt;ESC&gt;H0,10,192.168.0.18,43110 OK abcde12345 OK</pre> <p>Example2 – To send data to TCP server</p> <pre>&lt;ESC&gt;H1,10,192.168.0.18,1025 OK abcde12345 OK</pre> <p>Example3 – To send data to TCP server with '0, 0' as the destination/server</p> <pre>&lt;ESC&gt;H1,10,0,0 OK abcde12345 OK</pre> <p>Example4 – To send data to UDP receiver</p> <pre>&lt;ESC&gt;H2,10,192.168.0.18,1024 OK abcde12345 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Enabled by default in the SDK</li> <li>• The maximum length of data depends on TX_PAYLOAD_MAX_SIZE definition. It is defined in atcmd.h for SDK v3.x, TX_PAYLOAD_MAX_SIZE includes all parameters of AT command. Therefore, the maximum length of 'length' parameter depends on length of other parameters</li> <li>• TX_PAYLOAD_MAX_SIZE is defined 4,096 bytes in the SDK v3.2.3.0 or later</li> <li>• Not supported over SPI or SDIO</li> </ul>	

**Table 43: Data Reception Responses**

Response	Parameters	Description
+TRDTS	<pre>&lt;cid&gt;, &lt;src_ip&gt;,&lt;src_port&gt;, &lt;length&gt;,&lt;data&gt;</pre>	<p>Receive data through TCP server socket.</p> <p>&lt;cid&gt;: Assigned CID</p> <p>&lt;src_ip&gt;: Source IP address</p> <p>&lt;src_port&gt;: Source port number</p> <p>&lt;length&gt;: Data length</p> <p>&lt;data&gt;: Received data</p>
	<p>Example</p> <p>; When data is sent from a TCP client</p> <pre>+TRDTC:1,192.168.0.18,1025,4,test</pre>	

## DA16200 DA16600 Host Interface and AT Command

Response	Parameters	Description
+TRDTC	<cid>, <src_ip>,<src_port>, <length>,<data>	Receive data through TCP client socket. <cid>: Assigned CID <src_ip>: Source IP address <src_port>: Source port number <length>: Data length <data>: Received data
	Example ; When TCP client receives data, +TRDTC:1,192.168.0.18,1025,4,test	
+TRDUS	<cid>, <src_ip>,<src_port>, <length>,<data>	Receive data through UDP socket. <cid>: Assigned CID <src_ip>: Source IP address <src_port>: Source port number <length>: Data length <data>: Received data
	Example ; When UDP session receives data, +TRDUS:2,192.168.0.18,10194,4,test	

## 2) Data Transfer with DPM

### I. TCP Server

After a connection to an AP is made in the normal BOOT state, open a TCP server socket, and save the config to NVRAM.

```
AT+TRTS=32000
AT+TRSAVE
```

The TCP server socket that has been opened should be closed before switching to DPM mode.

```
AT+TRTRM=0
```

Change the DA16200/DA16600 state to DPM mode (AT+DPM=1). When the DA16200/DA16600 starts the session on DPM mode successfully, the following is shown:

```
+INIT:DONE,0
+WFJAP:1,'WI-FI_AP',192.168.5.19
+TRPALL:0,TCP,0.0.0.0,0,32000
```

When a TCP client connects to DA16200/DA16600, the following is shown:

```
+INIT:WAKEUP,UC
```

To receive +TRCTS message, send AT+MCUWUDONE immediately after “+INIT:WAKEUP,UC”

```
+TRCTS:0,192.168.0.1,42000
```

When the DA16200/DA16600 receives a message from a client, the following is shown:

```
+INIT:WAKEUP,UC
+TRDTS:0,192.168.0.1,42000,10,1234567890
```

To send a TCP message, send AT+MCUWUDONE immediately after “external wake-up” is triggered (+INIT:WAKEUP,EXT). To prevent that DA16200/DA16600 enters DPM Sleep mode, MCU should



## DA16200 DA16600 Host Interface and AT Command

send AT+CLRDPMSPLEXT before a message is sent. The DA16200/DA16600 can send data to a TCP client with the command "<ESC>S". Finally, to enter DPM sleep mode, send "AT+SETDPMSPLEXT".

```
+INIT:WAKEUP,EXT // external wake-up
AT+MCUWUDONE
AT+CLRDPMSPLEXT
...
<ESC>S...
...
AT+SETDPMSPLEXT
```

When a TCP client disconnects from DA16200/DA16600, the following is shown:

```
+INIT:WAKEUP,UC
```

To receive +RTXTS message, send AT+MCUWUDONE immediately after "+INIT:WAKEUP,UC"

```
+RTXTS:0,192.168.0.1,42000
```

### II. TCP Client

After a connection is made to an AP in the normal BOOT state, connect the TCP client of the DA16200/DA16600 to a TCP server and save the config to NVRAM. (To save TCP client config information, the DA16200/DA16600 should connect to the server successfully beforehand.)

```
AT+TRTC=192.168.5.1,34000
AT+TRSAVE
```

Before switching to DPM mode, disconnect the TCP Client:

```
AT+TRTRM=1
```

Change the DA16200/DA16600 state to DPM mode (AT+DPM=1). When the DA16200/DA16600 starts the session on DPM mode successfully, the following is shown:

```
+INIT:DONE,0
+WFJAP:1,'WI-FI_AP',192.168.5.19
+TRPALL:1,TCP,192.168.5.1,34000,30000
```

The procedure to exchange TCP data is the same as in Section I. When the DA16200/DA16600 receives a message from the server, the following is shown:

```
+INIT:WAKEUP,UC
+TRDTC:1,192.168.5.1,34000,10,1234567890
```

## DA16200 DA16600 Host Interface and AT Command

### III. UDP Session

After a connection is made to an AP in the normal BOOT state, open a UDP socket and save the config to NVRAM:

```
AT+TRUSE=48000
AT+TRSAVE
```

Before switching to DPM mode, disconnect TCP Client:

```
AT+TRTRM=2
```

Change the DA16200/DA16600 state to DPM mode. When the DA16200/DA16600 starts the session in DPM mode successfully, the following is shown:

```
+INIT:DONE,0
+WFJAP:1,'WI-FI_AP',192.168.5.19
+TRPALL:2,UDP,0.0.0.0,0,48000
```

The procedure to exchange UDP data is the same as in Section I. When the DA16200/DA16600 receives a message from the server, the following is shown:

```
+INIT:WAKEUP,UC
+TRDUS:2,192.168.5.23,35000,10,1234567890
```

#### 5.6.9.2 Secure Socket Commands

Table 44: Secure Socket Command List

Command	Parameters	Description
AT+TRSSLINIT	<Role>	Initialize the SSL module. DA16200/DA16600 allows to create a module of TLS client <Role>: The role of SSL, 1 – Client.
	<p>Example</p> <pre>AT+TRSSLINIT=1 +TRSSLINIT:0</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__SUPPORT_ATCMD_TLS__</code> is enabled in the SDK, this command will be enabled</li> </ul>	

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
AT+TRSSLCFG	<CID>,<Configuration ID>,<Configuration value>	<p>Configure SSL connection.</p> <p>&lt;CID&gt;: The CID obtained after issuing the AT+TRSSLINIT command</p> <p>&lt;Configuration ID&gt;: The configuration ID available in the below list of configurations:</p> <ul style="list-style-type: none"> <li>0 - Invalid configuration parameter</li> <li>2 - To set SSL CA Certificate</li> <li>3 - To set SSL Certificate</li> <li>6 - To set the SNI</li> <li>9 - To enable/disable server validation</li> <li>10 - To set the Incoming buffer length</li> <li>11 - To set the Outgoing buffer length</li> </ul> <p>&lt;Configuration value&gt;: Value to the configuration provided in configuration ID</p> <p>CONF_ID:CONF_VAL</p> <ul style="list-style-type: none"> <li>0 - Invalid</li> <li>2 - SSL CA Certificate Name</li> <li>3 - SSL Certificate Name</li> <li>6 - To Set the SNI (supported only for TLS client)</li> <li>9 - To enable/disable server validation <ul style="list-style-type: none"> <li>0: Disables server validation (Default)</li> <li>1: Enables server validation</li> </ul> </li> <li>10 - To set the Incoming buffer length</li> <li>11 - To set the outgoing buffer length</li> </ul>
<p>Prerequisite</p> <p>CID should be obtained (AT+TRSSLINT).</p> <p>SSL CA certificate and SSL certificate should be set up.</p> <p>Example</p> <pre>AT+TRSSLCFG=0,2,CA_CERT OK AT+TRSSLCFG=0,3,CERT OK AT+TRSSLCFG=0,6,da16x OK AT+TRSSLCFG=0,9,0 OK AT+TRSSLCFG=0,10,6144 OK AT+TRSSLCFG=0,11,6144 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If __SUPPORT_ATCMD_TLS__ is enabled in the SDK, this command will be enabled</li> </ul>		

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
AT+TRSSLCO	<CID>,<Server IP Address>,<Server Port number>	<p>Connect to an SSL server.</p> <p>&lt;CID&gt;: The CID obtained after issuing the AT+TRSSLINIT command</p> <p>&lt;Server IP Address&gt;: The IP Address of the server to connect. Only supported IPv4 address</p> <p>&lt;Server Port&gt;: The port number of the SSL server to connect</p>
	<p>Prerequisite</p> <p>CID should be obtained (AT+TRSSLINT).</p> <p>Example</p> <pre>AT+TRSSLCO=0,192.168.0.11,30000 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__SUPPORT_ATCMD_TLS__</code> is enabled in the SDK, this command will be enabled</li> </ul>	
AT+TRSSLWR	<CID>,[<Server IP Address>,<Server Port number>,<mode>],<Data length>,<Data>	<p>Send the data to the SSL server that is already established.</p> <p>&lt;CID&gt;: The CID obtained after issuing the AT+TRSSLINIT command</p> <p>&lt;Server IP Address&gt;: The IP Address of the SSL server is already established. If no input, the IP address would internally be used to the SSL server IP address</p> <p>&lt;Server Port number&gt;: The port number of the SSL server is already established. If no input, the Port number is internally used to the SSL server port number</p> <p>&lt;mode&gt;: Transmit data in raw or text mode. It is optional. If there is no option, data will be transmitted in text mode</p> <ul style="list-style-type: none"> <li>r: The raw mode is active. In raw mode, Data is read until data length. The data length is specified in &lt;length&gt; parameter</li> <li>t: The text mode is active. In text mode, the data can be affected if it has unprintable control codes like backspace(0x08)</li> </ul> <p>&lt;Data length&gt;: The length of data to send</p> <p>&lt;Data&gt;: The data to send. The input can be closed by &lt;Ctrl&gt;+C or reaching data length</p>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Prerequisite CID should be obtained (AT+TRSSLINT).</p> <p>Example AT+TRSSLWR=0,10,0123456789&lt;Ctrl&gt; + C OK</p> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__SUPPORT_ATCMD_TLS__</code> is enabled in the SDK, this command will be enabled</li> <li>The maximum length of data depends on <code>TX_PAYLOAD_MAX_SIZE</code> definition. It is defined in <code>atcmd.h</code> for SDK v3.x. <code>TX_PAYLOAD_MAX_SIZE</code> includes all parameters of AT command. Therefore, the maximum length of 'length' parameter depends on length of other parameters</li> </ul>	
AT+TRSSLCL	<CID>	<p>Close the SSL connection</p> <p>&lt;CID&gt;: The CID obtained after issuing AT+TRSSLINIT command</p>
	<p>Prerequisite CID should be obtained (AT+TRSSLINT).</p> <p>Example AT+TRSSLCL=0 OK</p> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__SUPPORT_ATCMD_TLS__</code> is enabled in the SDK, this command will be enabled</li> </ul>	
AT+TRSSLCERTLIST	<Certificate Type>	<p>Show a list of certificates or a list of CA data available in sflash memory</p> <p>&lt;Certificate Type&gt;: The value of the certificate. 0 – CA Certificates, 1 – Client/Server Certificates</p>
	<p>Example AT+TRSSLCERTLIST=0 +TRSSLCERTLIST=0,CA_CERT</p> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__SUPPORT_ATCMD_TLS__</code> is enabled in the SDK, this command will be enabled</li> </ul>	

## DA16200 DA16600 Host Interface and AT Command

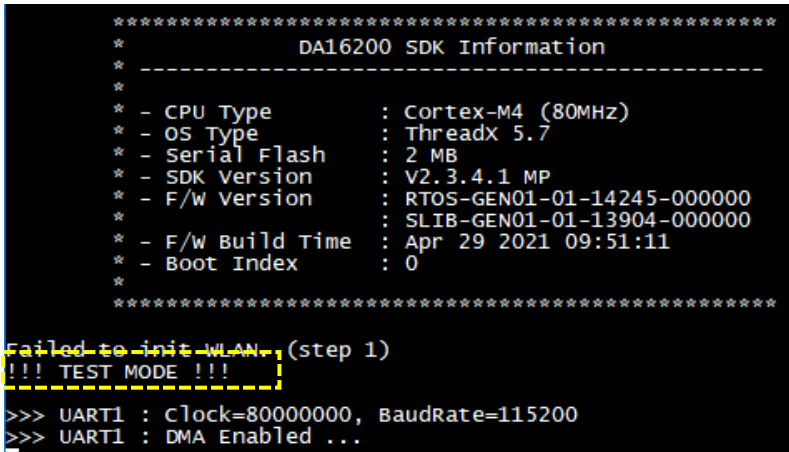
Command	Parameters	Description
AT+TRSSLCERTSTORE	<Certificate Type>, <Sequence>, <Format>, <Name>, [<Data length>, ]<Data>	<p>Store a certificate and CA list data in sflash memory.</p> <p>&lt;Certificate Type&gt;: The value of the certificate. 0 – CA Certificates, 1 – Client/Server Certificates</p> <p>&lt;Sequence&gt;: If the value of certificate type is 0 (CA), the number of certificates in the sequence is 1-5. If the certificate type is 1 (Client/Server certificate), then several certificates in a sequence is 1-SSL cert or 2-SSL key</p> <p>&lt;Format&gt;: The value of the CA/Certificate/Key0 – DER, 1 – PEM</p> <p>&lt;Name&gt;: The name of the certificate. While loading certificate and key file separately, the same name should be used in both commands</p> <p>&lt;Data length&gt;: The length of certificate data. If certificate is DER format, data length parameter is mandatory</p> <p>&lt;Data&gt;: The certificate data to be store</p>
	<p>Example</p> <pre>AT+TRSSLCERTSTORE=0,1,1,CA_CERT, -----BEGIN CERTIFICATE----- ... -----END CERTIFICATE-----&lt;Ctrl&gt;+C OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__SUPPORT_ATCMD_TLS__</code> is enabled in the SDK, this command will be enabled</li> </ul>	
AT+TRSSLCERTDELETE	<Certificate Type>,<Name>	<p>Delete a certificate or CA list data in sflash memory</p> <p>&lt;Certificate Type&gt;: The type of the certificate. 0 – CA Certificates, 1 – Client/Server Certificates</p> <p>&lt;Name&gt;: The name of the certificate</p>
	<p>Example</p> <pre>AT+TRSSLCERTDELETE=0,CA_CERT OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__SUPPORT_ATCMD_TLS__</code> is enabled in the SDK, this command will be enabled</li> </ul>	
AT+TRSSLSAVE	(none)	Store the current SSL module's configuration in NVRAM
	<p>Example</p> <pre>AT+TRSSLSAVE OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__SUPPORT_ATCMD_TLS__</code> is enabled in the SDK, this command will be enabled</li> </ul>	
	(none)	Delete the stored SSL module's configuration in NVRAM

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
AT+TRSSLDELETE	<p>Example</p> <pre>AT+TRSSLDELETE OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__SUPPORT_ATCMD_TLS__</code> is enabled in the SDK, this command will be enabled</li> </ul>	

### 5.6.10 RF Test Function Commands

Table 45: RF Test Command List

Command	Parameters	Description
AT+TMRFNOINIT	<p>&lt;flag&gt;</p> <p>Example</p> <pre>AT+TMRFNOINIT=1 OK  AT+RESTART OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>To test RF performance, set the boot mode as RF test mode (AT+TMRFNOINIT=1) and restart the DA16200/DA16600 (AT+RESTART)</li> <li>After DA16200/DA16600 is restarted, “!!! TEST MODE !!!” log is displayed</li> </ul>  <p>The screenshot shows the DA16200 SDK Information and a log entry. The information includes CPU Type (Cortex-M4 80MHz), OS Type (ThreadX 5.7), Serial Flash (2 MB), SDK Version (V2.3.4.1 MP), F/w Version (RTOS-GEN01-01-14245-000000), F/w Build Time (Apr 29 2021 09:51:11), and Boot Index (0). The log entry shows 'Failed to init WLAN. (step 1)' and '!!! TEST MODE !!!'.</p>	<p>Set boot mode</p> <p>&lt;flag&gt;: 0 (normal boot), 1 (RF test mode boot)</p> <p>Response: OK or ERROR</p>
AT+TMLMACINIT	(none)	<p>Initialize LMAC (for test mode)</p> <p>Response: OK or ERROR</p>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	Prerequisite Boot as RF test mode (AT+TMRFNOINIT=1).  Example AT+TMLMACINIT OK  Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> </ul>	
AT+RFTESTSTART	(none)	Start RF test mode
	Prerequisite Boot as RF test mode (AT+TMRFNOINIT=1).  Example AT+RFTESTSTART OK  Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> </ul>	
AT+RFTX	<Ch>, <BW>, <numFrames>, <frameLen>, <txRate>, <txPower>, <destAddr>, <bssid>, <htEnable>, <GI>, <greenField>, <preambleType>, <qosEnable>, <ackPolicy>, <scrambler>, <aifsnVal>, <ant>	Start RF TX test. <Ch>: Carrier frequency (2412 ~ 2484 MHz) <BW>: [0]: Fixed. Carrier bandwidth. 20 MHz fixed <numFrames>: Number of frames to transmit <frameLen>: Length of frame (bytes) <txRate>: Data rate b1: 11b DSSS 1 Mbps b2: 11b DSSS 2 Mbps b5_5: 11b DSSS 5.5 Mbps b11: 11b DSSS 11 Mbps g6: 11g 6 Mbps g9: 11g 9 Mbps g12: 11g 12 Mbps g18: 11g 18 Mbps g24: 11g 24 Mbps g36: 11g 36 Mbps g48: 11g 48 Mbps g54: 11g 54 Mbps n6_5: 11n 6.5 Mbps (7.2 Mbps @Short GI) n13: 11n 13 Mbps (14.4 Mbps @Short GI) n19_5: 11n 19.5 Mbps (21.7 Mbps @Short GI) n26: 11n 26 Mbps (28.9 Mbps @Short GI) n39: 11n 39 Mbps (43.3 Mbps @Short GI) n52: 11n 52 Mbps (57.8 Mbps @Short GI) n58_5: 11n 58.5 Mbps (65 Mbps @Short GI) n65: 11n 65 Mbps (72.2 Mbps @Short GI) <txPower>: TX power (0 ~ 15), 0.8 dB step <destAddr>: MAC address to send packet



## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
		<p>&lt;bssid&gt;: BSSID</p> <p>&lt;htEnable&gt;: N/A</p> <p>&lt;GI&gt;: [short long]. Guard interval. 11n mode only</p> <p>&lt;greenField&gt;: [on off]. Set greenfield mode on/off</p> <p>&lt;preambleType&gt;: [short long]. Preamble type @DSSS mode</p> <p>&lt;qosEnable&gt;: [on off]. MAC header QoS control</p> <p>&lt;ackPolicy&gt;: [NO NORM BA CBA]</p> <p>&lt;scrambler&gt;: N/A</p> <p>&lt;aifsnVal&gt;: [0 ~ 15]. Indicate the AIFS in units of slots after SIFS that HW should wait for before starting backoff, for access category</p> <p>&lt;ant&gt;: [0]. Fixed</p> <p>Response: OK or ERROR</p>
	<p>Prerequisite</p> <p>Start RF test mode (AT+RFTESTSTART).</p> <p>Example</p> <p>; Tx test with 11N MCS7, 2412MHz and power grade as '0' (max power)</p> <p>AT+RFTX 2412,0,0,1000,n65,0</p> <p>OK</p> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> </ul>	
AT+RFTXSTOP	(none)	Stop RF TX test
	<p>Prerequisite</p> <p>Start RF TX test (AT+RFTX).</p> <p>Example</p> <p>AT+RFTX 2412,0,0,1000,n65,0</p> <p>OK</p> <p>AT+RFTXSTOP</p> <p>OK</p> <p>AT+RFTX 2442,0,0,1000,n65,0</p> <p>OK</p> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>AT+RFTXSTOP is required before testing other items</li> </ul>	
AT+RFCWTEST	<p>&lt;Ch&gt;, &lt;BW&gt;, &lt;txPower&gt;, &lt;ant&gt;, &lt;CWCycle&gt;</p>	<p>Start CW test</p> <p>&lt;Ch&gt;: Carrier frequency (2412 ~ 2484 MHz)</p> <p>&lt;BW&gt;: [0]: Fixed. Carrier bandwidth. 20 MHz fixed</p> <p>&lt;txPower&gt;: TX power (0 ~ 15), 0.8 dB step</p> <p>&lt;ant&gt;: [0]. Fixed</p> <p>&lt;CWCycle&gt;: 1 MHz fixed</p> <p>Response: OK or ERROR</p>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	Prerequisite Start RF test mode (AT+RFTESTSTART).  Example AT+RFCWTEST 2442,0,2 OK CW Tx test with 2442 MHz and power grade as 2  Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> </ul>	
AT+RFCWSTOP	(none)	Stop CW test Response: OK or ERROR
	Prerequisite Start RF CW test (AT+RFCWTEST).  Example AT+RFCWTEST 2442,0,2 OK  AT+RFCWSTOP OK  AT+RFCWTEST 2472,0,2 OK  Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>AT+RFCWSTOP is required before testing other items</li> </ul>	
AT+RFCONTSTART	<txRate>, <txPower>, <Ch>	Start RF continuous TX test <txRate>: Data rate. See the AT+RFTX command <txPower>: TX power (0 ~ 15), 0.8 dB step <Ch>: Carrier frequency (2412 ~ 2484 MHz) Response: OK or ERROR
	Prerequisite Start RF test mode (AT+RFTESTSTART).  Example ; Continuous Tx test with 11G 54 MHz, 2472 MHz and power grade as 2 AT+RFCONTSTART g54,2,2472 OK  Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> </ul>	
AT+RFCONTSTOP	(none)	Stop RF continuous TX test Response: OK or ERROR

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	Prerequisite Start RF continuous TX test (AT+RFCONTSTART).  Example AT+RFCONTSTART g54,2,2412 OK  AT+RFCONTSTOP OK  AT+RFCONTSTART g54,2,2472 OK  Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> </ul>	
AT+RFCHANNEL	<Ch>	Change RF channel for PER test <Ch>: Carrier frequency (2412 ~ 2484 MHz) Response: OK or ERROR
	Prerequisite Start RF test mode (AT+RFTESTSTART).  Example AT+RFCHANNEL 2412 OK  Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> </ul>	
AT+RFPERRESET	(none)	Reset PER count Response: OK or ERROR
	Prerequisite Start RF test mode (AT+RFTESTSTART).  Example AT+RFPERRESET OK  Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> </ul>	
AT+RFPER	(none)	Display PER state Indicate a number of Valid packets, FCS Errors packets, PHY Errors packets, and Overflow Errors Response: OK or ERROR

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	Prerequisite Start RF test mode (AT+RFTESTSTART).  Example AT+RFPER 20 0 0 0 OK  Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> </ul>	
AT+RFTESTSTOP	(none)	Stop RF test mode.
	Example AT+RFTESTSTOP OK  Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> </ul>	

**Table 46: RF Test Examples**

Test Step	Command	Description
ECHO on	ATE	ECHO ON
Set boot mode	AT+TMRFNOINIT=1	Set boot mode as RF test mode
Restart the DA16200/DA16600	AT+RESTART	Reboot as RF test mode
ECHO on	ATE	ECHO ON
Start RF test mode	AT+RFTESTSTART	Start RF test mode.
Tx Test @11B 1Mbps	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
Tx Test @11G 54Mbps	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
Tx Test @11N MCS7	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

## DA16200 DA16600 Host Interface and AT Command

Test Step	Command	Description
	AT+RFTXSTOP	Stop Tx
	AT+RFTX 2472,0,0,1000,n65,0	11N MCS7/Channel 13
	AT+RFTXSTOP	Stop Tx
Rx Test	AT+RFCHANNEL 2412	Change RF channel to 1
	AT+RFPERRESET	Reset PER count
	AT+RFPER	Display PER state
	AT+RFCHANNEL 2442	Change RF channel to 7
	AT+RFPERRESET	Reset PER count
	AT+RFPER	Display PER state
	AT+RFCHANNEL 2472	Change RF channel to 13
	AT+RFPERRESET	Reset PER count
	AT+RFPER	Display PER state
Stop RF test mode	AT+RFTESTSTOP	Stop RF test mode

### NOTE #1

Renesas Electronics provides the AT-GUI tool to test RF performance easily. The tool and manual are available on the Renesas website (<https://www.renesas.com/us/en/products/wireless-connectivity/wi-fi/low-power-wi-fi>). See Ref. [4].

### NOTE #2

The 2.4 GHz band is divided into 14 channels at 5 MHz intervals centered at 2.412 GHz, starting with channel 1. The last channel (CH 14) has additional restrictions or cannot be used for use in all regulatory areas.

- TX power setting value range: 0x0 ~ 0xB
- Setting value for unsupported channel: 0xF

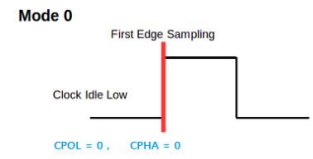
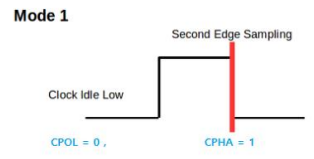
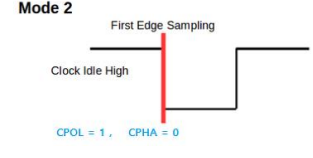
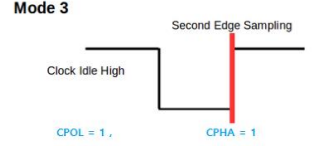
## 5.6.11 System and Peripheral Function Commands

### 5.6.11.1 SPI Commands

Table 47: SPI Command List

Command	Parameters	Description
AT+SPICONF	<clockpol>, <clockpha>	Configure SPI <clockpol>: Clock polarity [0 1] <clockpha>: Clock phase [0 1]

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre>AT+SPICONF=1,1 OK  AT+SPICONF=0,1 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__SUPPORT_PERI_CMD__</code> is enabled in the SDK, this command will be enabled</li> <li>The <code>&lt;clockpol&gt;</code> sets the polarity of the clock signal during the idle state. The idle state is defined as the period when CS is high and transitioning to low at the start of the transmission and when CS is low and transitioning to high at the end of the transmission. The <code>&lt;clockpha&gt;</code> selects the clock phase. Depending on the <code>&lt;clockpha&gt;</code>, the rising or falling clock edge is used to sample the data. The default value of DA16200/DA16600 are <code>clockpol,0</code> and <code>clockpha 0</code></li> </ul> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p><b>Mode 0</b></p>  <p>First Edge Sampling</p> <p>Clock Idle Low</p> <p>CPOL = 0, CPHA = 0</p> </div> <div style="text-align: center;"> <p><b>Mode 1</b></p>  <p>Second Edge Sampling</p> <p>Clock Idle Low</p> <p>CPOL = 0, CPHA = 1</p> </div> </div> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p><b>Mode 2</b></p>  <p>First Edge Sampling</p> <p>Clock Idle High</p> <p>CPOL = 1, CPHA = 0</p> </div> <div style="text-align: center;"> <p><b>Mode 3</b></p>  <p>Second Edge Sampling</p> <p>Clock Idle High</p> <p>CPOL = 1, CPHA = 1</p> </div> </div>	

### 5.6.11.2 OTP Commands

Table 48: OTP Command List

Command	Parameters	Description
AT+UOTPRDASC	<addr>,<cnt>	<p>Read OTP data</p> <p>&lt;addr&gt;: OTP address to read 4-byte aligned</p> <p>&lt;cnt&gt;: Bytes to read</p> <p>Response: OK or Error</p> <p>A string of four-bit HEXA value represented by the ASCII code</p>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre> ; Reading 4 bytes at offset h180 (h180 * 4 = h600) ; If data "12345678" is written to 0x600, can read the values AT+UOTPRDASC=600,4 12345678 OK </pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Enabled by default in the SDK v3.2.3.0 or later</li> <li>• If __SUPPORT_PERI_CMD__ is enabled in the SDK, this command will be enabled</li> <li>• Physical OTP offset range of DA16200/DA16600 is h0~h1FF; at each offset, 4 bytes are stored or read</li> <li>• For accessing OTP using this command, 4-byte aligned address should be given. For example: h0, h4, h8 ...</li> </ul>	
AT+UOTPWRASC	<addr>,<cnt>,<value>	<p>Write OTP data</p> <p>&lt;addr&gt;: OTP address to write 4-byte aligned</p> <p>&lt;cnt&gt;: Bytes to write</p> <p>&lt;value&gt;: A string of four-bit HEXA value represented by the ASCII code</p> <p>Response: OK or Error</p> <p><b>Important</b></p> <p>For MAC address read or write, AT+WFOTP (write) and AT+WFMAC (read) must be used. Do not use AT+UOTPRDASC or AT+UOTPWRASC for this purpose.</p> <p>OTP offset from 0x00 ~ 0x2b should not be written as this section is for "secure" boot.</p>
	<p>Example</p> <pre> ; Writing h12345678 to OTP Address 0x600: ; To write "12345678" data into the 0x600, AT+UOTPWRASC=600,4,12345678 OK  ; To read written Data via UOTPRDASC AT+UOTPRDASC=600,4 12345678 OK </pre> <p>Note</p> <ul style="list-style-type: none"> <li>• Enabled by default in the SDK v3.2.3.0 or later</li> <li>• If __SUPPORT_PERI_CMD__ is enabled in the SDK, this command will be enabled</li> <li>• Physical OTP offset range of DA16200/DA16600 is h0~h1FF; at each offset, 4 bytes are stored or read</li> <li>• For accessing OTP using this command, 4-byte aligned address should be given. For example: h0, h4, h8 ...</li> </ul>	

DA16200/DA16600 provides four slots to store MAC addresses and 8 bytes are allocated for each slot.

## DA16200 DA16600 Host Interface and AT Command

**Table 49: OTP Memory Address for Writing MAC Address**

Slot	OTP Address	Description	Size (Byte)
MAC Address #0	0x100	MAC Address Low	4
	0x101	MAC Address High	4
MAC Address #1	0x102	MAC Address Low	4
	0x103	MAC Address High	4
MAC Address #2	0x104	MAC Address Low	4
	0x105	MAC Address High	4
MAC Address #3	0x106	MAC Address Low	4
	0x107	MAC Address High	4

DA16200/DA16600 provides two slots to store XTAL offset in the OTP memory. Slot #0 is the primary slot while Slot#1 is for back-up, which is used when overriding Slot #0.

**Table 50: Size of Memory by XTAL Offset**

Slot	OTP Address	Description	Size (Byte)
XTAL Offset #0	0x10A	XTAL Offset #0 value	1
XTAL Offset #1	0x10B	XTAL Offset #1 value	1

### 5.6.11.3 XTAL Commands

These commands are used for XTAL calibration and the usage is described in DA16200/DA16600 Mass Production Guide.

**Table 51: XTAL Command List**

Command	Parameters	Description
AT+XTALWR	<value>	Write XTAL Offset to DA16200/DA16600 system register. <value>: Seven-bits to write [h'1 ~ h'7f] Response: OK or Error
	<p>Example</p> <pre>AT+XTALWR=7f OK  AT+XTALWR=80 ERROR</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If __SUPPORT_PERI_CMD__ is enabled in the SDK, this command will be enabled</li> </ul>	
AT+XTALRD	(none)	Read XTAL Offset from DA16200/DA16600 System Response: <CR><LF><A string of seven-bit HEXA value represented by the ASCII Code><CR><LF>OK<CR><LF> or Error



## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre>AT+XTALRD 0x7f OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__SUPPORT_PERI_CMD__</code> is enabled in the SDK, this command will be enabled</li> </ul>	

### 5.6.11.4 Flash Dump Commands

Table 52: Flash Dump Command List

Command	Parameters	Description
AT+FLASHDUMP	<p>&lt;address&gt;, &lt;length&gt;</p> <p>Example</p> <pre>; The following example reads 32 kB from 0x00, (1024*32 = 32768) AT+FLASHDUMP=0,32768</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__SUPPORT_PERI_CMD__</code> is enabled in the SDK, this command will be enabled</li> </ul>	<p>Dump serial flash data</p> <p>&lt;address&gt;: Start address [h'0 ~ h'3ffff]</p> <p>&lt;length&gt;: Data length [d']</p> <p>Response:</p> <pre>&lt;CR&gt;&lt;LF&gt; &lt;dump data&gt; &lt;CR&gt;&lt;LF&gt;OK&lt;CR&gt;&lt;LF&gt; or Error</pre>

### 5.6.11.5 GPIO Commands

Table 53: GPIO Command List

Command	Parameters	Description
AT+GPIOSTART	<p>&lt;port&gt;, &lt;pin &gt;, &lt;direction&gt;</p>	<p>Configure the GPIO pin mux and the direction of a GPIO</p> <p>&lt;port&gt;: GPIO port number</p> <ul style="list-style-type: none"> <li>0: GPIOA</li> <li>2: GPIOC</li> </ul> <p>&lt;pin&gt;: GPIO pin number. This is a hexadecimal value and indicates a GPIO bitmap</p> <ul style="list-style-type: none"> <li>GPIOA: GPIOA0 ~ GPIOA11</li> <li>GPIOC: GPIOC6 ~ GPIOC8</li> </ul> <p>&lt;direction&gt;: GPIO pin direction</p> <ul style="list-style-type: none"> <li>0: Set the pin as an input</li> <li>1: Set the pin as an output</li> </ul> <p>Response: OK or Error</p>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <p>; To configure GPIOA [3:0] output with using UART interface: ; GPIO (0, 1, 2, 3) is set to binary 1 (0000 0000 0000 1111). AT+GPIOSTART=0,f,1 OK</p> <p>; To configure GPIOA [3:0] output with using SPI interface: : Avoid reassigning default SPI-pin. ; GPIO (4, 5, 6, 7) is set to binary 1 (0000 0000 1111 0000). AT+GPIOSTART=0,f,0,1 OK</p> <p>; To configure GPIOC [8:6] input: ; GPIO (6, 7, 8) is set to binary 1 (0000 0001 1100 0000). AT+GPIOSTART=2,1c0,0 OK</p> <p>Note</p> <ul style="list-style-type: none"> <li>• Enabled by default in the SDK v3.2.3.0 or later</li> <li>• If <code>__SUPPORT_PERI_CMD__</code> is enabled in the SDK, this command will be enabled</li> </ul>	
AT+GPIORD	<port>, <pin>	<p>Read the GPIO input level</p> <p>&lt;port&gt;: GPIO port number</p> <ul style="list-style-type: none"> <li>• 0: GPIOA</li> <li>• 2: GPIOC</li> </ul> <p>&lt;pin&gt;: GPIO pin number. This is a hexadecimal value and indicates a GPIO bitmap</p> <p>Response: &lt;Read value&gt;: [h'0 ~ h'1fff] OK or Error</p>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre>; Configure GPIOC[8:6] as output and set to high. AT+GPIOSTART=2,1c0,1 OK  AT+GPIOWR=2,1c0,1 OK  ; Read back the status of the pins: AT+GPIORD=2,1c0 0x01c0 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__SUPPORT_PERI_CMD__</code> is enabled in the SDK, this command will be enabled</li> <li>The read value indicates GPIO bitmap. If a value is 0x1c0, it means GPIO #6, #7, #8 high</li> </ul>	
AT+GPIOWR	<p>&lt;port&gt;, &lt;pin&gt;, &lt;level&gt;</p>	<p>Configures the output level of GPIO pins</p> <p>&lt;port&gt;: GPIO port number</p> <ul style="list-style-type: none"> <li>0: GPIOA</li> <li>2: GPIOC</li> </ul> <p>&lt;pin&gt;: GPIO pin number. This is a hexadecimal value and indicates a GPIO bitmap</p> <p>&lt;level&gt;: GPIO output level</p> <ul style="list-style-type: none"> <li>0: Low</li> <li>1: High</li> </ul> <p>Response: OK or Error</p>
	<p>Prerequisite</p> <p>Change the direction of GPIO to output (AT+GPIOSTART).</p> <p>Example</p> <pre>; Configure GPIOC[8:6] as output and set to high. AT+GPIOSTART=2,1c0,1 OK  AT+GPIOWR=2,1c0,1 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__SUPPORT_PERI_CMD__</code> is enabled in the SDK, this command will be enabled</li> </ul>	
AT+SAVE_PININFO	(none)	<p>Save pin mux information</p> <p>Response: OK or Error</p>

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre>AT+SAVE_PININFO OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__SUPPORT_PERI_CMD__</code> is enabled in the SDK, this command will be enabled</li> <li>It is to save a current PIN mux configured</li> </ul>	
AT+RESTORE_PININFO	(none)	Restore pin mux information Response: OK or Error
	<p>Example</p> <pre>AT+RESTORE_PININFO OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If <code>__SUPPORT_PERI_CMD__</code> is enabled in the SDK, this command will be enabled</li> <li>It is to restore the PIN multiplexing status saved through the AT+SAVE_PININFO command</li> </ul>	

### 5.6.11.6 LED Commands

Table 54: LED Command List

Command	Parameters	Description
AT+LEDINIT	<none>	Configure GPIOC_6 (LED1), GPIOC_7 (LED2), and GPIOC_8 (LED3) pins to GPIO output Response: OK or Error
	<p>Example</p> <pre>AT+LEDINIT +OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.2.1 or later</li> </ul>	
AT+LEDCTRL	<port >, <status>	Set LED1/2/3 (GPIOC_6/7/8) pin to output High or Low <port>: GPIO port number 1: GPIOC_6 2: GPIOC_7 3: GPIOC_8 <status>: LED status off: LED off on: LED on Response: OK or Error

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
	<p>Example</p> <pre>AT+LEDCTRL=1,off OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.2.1 or later</li> </ul>	

### 5.6.11.7 PWM Commands

Table 55: PWM Command List

Command	Parameters	Description
AT+PWMINIT	<none>	Configure GPIOA_10 pin to PWM output Response: OK or Error
	<p>Example</p> <pre>AT+PWMINIT +OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.2.1 or later</li> <li>If <code>__ATCMD_IF_UART1__</code> is enabled in the SDK, this command will be enabled</li> </ul>	
AT+PWMSTART	<channel >, <period>, <duty> <mode cycle>	Start PWM output (GPIOA_10) with given period and duty <channel>: PWM channel, fixed as 0 <period>: period of one clock (microsecond) <duty>: duty as percentage <mode cycle>: fixed as 0 Response: OK or Error
	<p>Example</p> <pre>AT+PWMSTART=0,40,50,0 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.2.1 or later</li> <li>If <code>__ATCMD_IF_UART1__</code> is enabled in the SDK, this command will be enabled</li> </ul>	
AT+PWMSTOP	<none>	Stop PWM output Response: OK or Error
	<p>Example</p> <pre>AT+PWMSTOP OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.2.1 or later</li> <li>If <code>__ATCMD_IF_UART1__</code> is enabled in the SDK, this command will be enabled</li> </ul>	

## DA16200 DA16600 Host Interface and AT Command

### 5.6.11.8 ADC Commands

**Table 56: ADC Command List**

Command	Parameters	Description
AT+ADCINIT	<none>	Configure GPIOA_0, GPIOA_1, GPIOA_2, GPIOA_3 to analog input pins for ADC GPIOA_0: ADC channel 0 GPIOA_1: ADC channel 1 GPIOA_2: ADC channel 2 GPIOA_3: ADC channel 3 Response: OK or Error
	<p>Example</p> <pre>AT+ADCINIT OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.2.1 or later</li> <li>If __ATCMD_IF_UART1__ is enabled in the SDK, this command will be enabled</li> </ul>	
AT+ADCCHEN	<channel >, < resolution >	Enable given ADC channel <channel>: ADC channel number [0 1 2 3] <resolution>: ADC resolution. fixed as 12-bit Response: OK or Error
	<p>Example</p> <pre>AT+ADCCHEN=0,12 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.2.1 or later</li> <li>If __ATCMD_IF_UART1__ is enabled in the SDK, this command will be enabled</li> </ul>	
AT+ADCSTART	<divider>	Start ADC function <divider>: divider ADC sampling rate For example, when divider is 1, 1MHz / (divider (1) + 1) = 500khz Response: OK or Error
	<p>Example</p> <pre>AT+ADCSTART=1 OK</pre> <p>Note</p> <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.2.1 or later</li> <li>If __ATCMD_IF_UART1__ is enabled in the SDK, this command will be enabled</li> </ul>	

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
AT+ADCREAD	<channel>, <sample count>	Read ADC value <channel>: ADC channel number [0 1 2 3] <sample count>: count of sample to read Response: <Read value>: [sample count]  Response: OK or Error
	Example <pre>AT+ADCREAD=0,16 [ 279 275 269 271 270 268 268 274 274 277 276 269 271 276 264 274 ] OK</pre> Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.2.1 or later</li> <li>If __ATCMD_IF_UART1__ is enabled in the SDK, this command will be enabled</li> </ul>	
AT+ADCSTOP	(none)	Stop ADC function Response: OK or Error
	Example <pre>AT+ADCSTOP OK</pre> Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.2.1 or later</li> <li>If __ATCMD_IF_UART1__ is enabled in the SDK, this command will be enabled</li> </ul>	

### 5.6.11.9 I2C Commands

Table 57: I2C Command List

Command	Parameters	Description
AT+I2CINIT	<none>	Configure GPIOA_8 (I2C_SDA), GPIOA_9 (I2C_SCL) pins to I2C pins Response: OK or Error
	Example <pre>AT+I2CINIT OK</pre> Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.2.1 or later</li> </ul>	

## DA16200 DA16600 Host Interface and AT Command

Command	Parameters	Description
AT+I2CREAD	<slave address>, <register>, <length>	Read values from registers of I2C device <slave address>: 8bit slave address of I2C device (hex) <register>,: register value to read (hex) <length>: data length to read (decimal) Response: <Read values> (hex)  Response: OK or Error
	Example AT+I2CREAD=d0,10,1 66 OK  Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.2.1 or later</li> </ul>	
AT+ I2CWRITE	<slave address>, <register>, <length>, <values>	Write values to I2C register of I2C device <slave address>: 8-bit slave address of I2C device (hex) <register>: register value to write (hex) <length>: data length to write (decimal) <values>: data to write (hex) Response: OK or Error
	Example AT+I2CWRITE=d0,10,3,670292 OK  Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.2.1 or later</li> </ul>	

### 5.6.11.10 Sleep Commands

Table 58: Sleep Command List

Command	Parameters	Description
AT+SLEEPMS	<period>	Make DA16200/DA16600 go to Sleep mode 3 and wake up after <period> milliseconds <period>: Wake-up time in milliseconds. Max period: 2097151000 (about 24 days) Response: OK or Error
	Example AT+SLEEPMS=5000 +INIT:DONE,0  Note <ul style="list-style-type: none"> <li>Enabled by default in the SDK v3.2.3.0 or later</li> <li>If __SUPPORT_PERI_CMD__ is enabled in the SDK, this command will be enabled</li> </ul>	



## DA16200 DA16600 Host Interface and AT Command

### 5.6.11.11 CALWL Commands

Table 59: CALWL Command List

Command	Parameters	Description
AT+CALWR	<code>&lt;gmode_tx_rf_proc&gt;</code> , <code>&lt;txpga_gmode_cal&gt;</code>	<p>Change RF TX GAIN Calibration register for test</p> <p><code>&lt;gmode_tx_rf_proc&gt;</code></p> <ul style="list-style-type: none"> <li>7 bits hexadecimal without "0x" prefix</li> <li>Offset = bit[5:0] x 0.8 dB</li> <li>MSB[6] bits 0 then TX gain is decreased</li> <li>MSB[6] bits 1 then TX gain is increased</li> </ul> <p><code>&lt;txpga_gmode_cal&gt;</code></p> <ul style="list-style-type: none"> <li>0 : 0dB offset</li> <li>1 : -0.2dB offset</li> <li>2 : -0.4dB offset</li> <li>3 : -0.6dB offset</li> </ul>
<p>Example</p> <p>1. TX measured +14 dBm and change to +13 dBm</p> <pre> AT+CALWR=1,1 OK gmode_tx_rf_proc = 1, -0.8 dB txpga_gmode_cal = 1, -0.2 dB  Changed TX Gain: 14dBm - 0.8 dB - 0.2 dB = 13.0 dBm </pre> <p>2. TX measured +13 dBm and change to +13.6 dBm</p> <pre> AT+CALWR=41,3 OK gmode_tx_rf_proc = 41, +0.8 dB txpga_gmode_cal = 1, -0.2 dB  Changed TX Gain: 13dBm + 0.8 dB - 0.2 dB = 13.6 dBm </pre> <p>Note</p> <ul style="list-style-type: none"> <li>This setting is not saved to system and changed when reboot system.</li> <li>If <code>__SUPPORT_PERI_CMD__</code> is enabled in the SDK, this command will be enabled</li> </ul>		

## DA16200 DA16600 Host Interface and AT Command

## 6 Examples Using AT Commands


### 6.1 Data Transfer Test

This section describes how to test the transfer function commands with a data terminal emulator. Some of the terminal applications to use for this purpose are:

- IO Ninja: <http://ioninja.com/>
  - HEXA data and file transmitting function
- Socket Test: <http://sockettest.sourceforge.net/>
  - Text data transmittable only
- Script Communicator: <http://sourceforge.net/projects/scriptcommunicator>
  - Socket communication, UART Rx/Tx data color-distinguished output function and HEXA data transmission

The following sections describe test procedures for socket communication between the DA16200/DA16600 and a PC with IO Ninja. Run DA16200/DA16600 AT commands on a serial terminal application on the local PC. The terminal must be connected to the UART1 interface of the DA16200/DA16600.

#### 6.1.1 TCP Server Socket Test

1. DA16200/DA16600 AT command:
  - a. `AT+TRTS=1234` ← Open a TCP server socket of which the port is 1234.
2. PC:
  - a. Select **TCP Connection Socket** (#1, [Figure 27](#)).
  - b. Enter the IP address and the port number of DA16200/DA16600 (#2, [Figure 27](#)).
  - c. Click  to connect the socket (#3, [Figure 27](#)).
3. DA16200/DA16600 AT command:
  - a. `+TRCTS:0,192.168.0.5,3713` ← A TCP client socket connected, and IP address is 192.168.0.5 and port is 3713.
4. PC:
  - a. Send data (#4, [Figure 27](#)).
5. DA16200/DA16600 AT command:
  - a. `+TRDTS:0,192.168.0.5,3713,10,DIA_AT_TCP` ← Received 10 bytes of data DIA\_AT\_TCP.

## DA16200 DA16600 Host Interface and AT Command

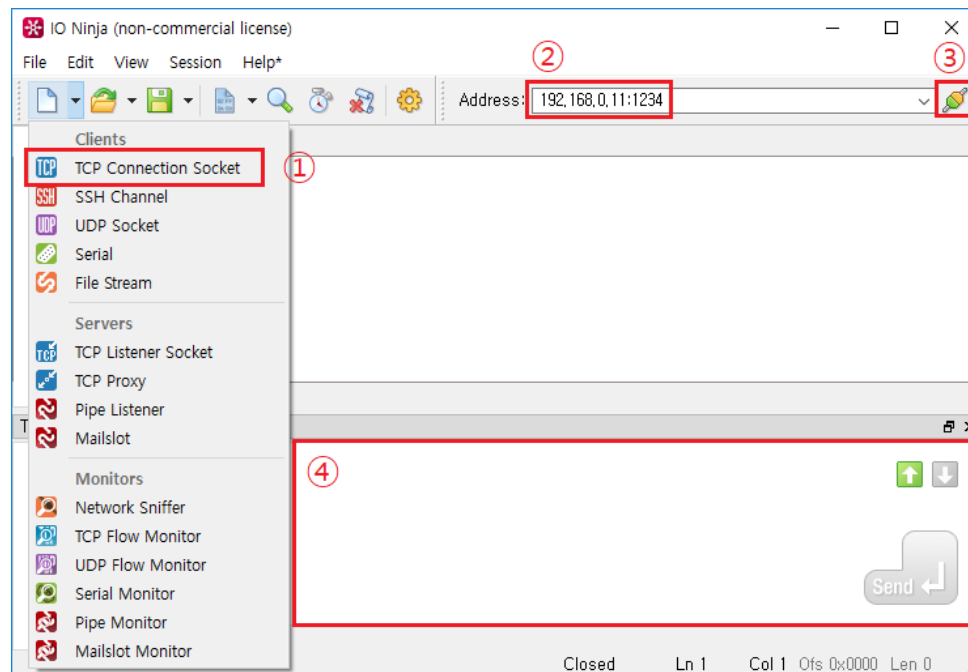



Figure 27: IO Ninja – TCP Client Socket Setting

### 6.1.2 TCP Client Socket Test

1. PC:
  - a. Select **TCP Listener Socket** (#1, Figure 28).
  - b. Enter the port number to be used (#2, Figure 28).
  - c. Click  to start to Listen (#3, Figure 28).
2. DA16200/DA16600 AT command:
  - a. `AT+TRTC=192.168.0.5,1234,2300` ← Open a TCP client socket and set the server IP (192.168.0.5), port (1234), and the local port (2300).
  - b. `<ESC>S18,0,0,12345678` ← Send 8 bytes of data 12345678.
3. PC:
  - a. Received data.
  - b. Send data (#4, Figure 28).
4. DA16200/DA16600 AT command:
  - a. `+TRDTC:1,192.168.0.5,1234,10,DIA_AT_TCP` ← Received 10 bytes of data DIA\_AT\_TCP.

## DA16200 DA16600 Host Interface and AT Command

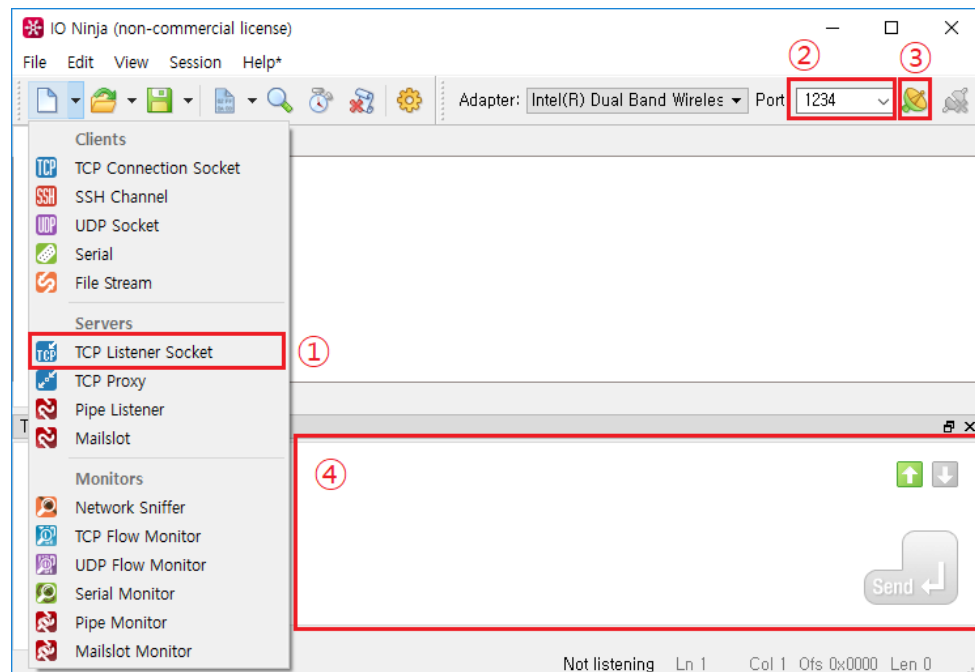


Figure 28: IO Ninja – TCP Server Socket Setting

### 6.1.3 UDP Socket Test

1. PC:
  - v Select **UDP Socket** (#1, Figure 29).
  - a. Enter the port number to be used and click to open the socket (#2, Figure 29).
  - b. Enter the IP address and port of the counterpart's UDP socket, click and get ready for data transmission (#3, Figure 29).
  - c. Enter data and click **Send** to transmit (#4, Figure 29).
2. DA16200/DA16600 AT command:
  - a. `AT+TRUSE=4567` ← Open a UDP socket and set the local port (4567).
  - b. `AT+TRUR=192.168.0.5,1234` ← Set the remote IP (192.168.0.5) and port (1234).
  - c. `<ESC>S210,0,0,1234567890` ← Send 10 bytes of data 1234567890.
  - d. `+TRDTC:0,10,DIA_AT_UDP` ← Received 10 bytes of data DIA\_AT\_UDP.

## DA16200 DA16600 Host Interface and AT Command

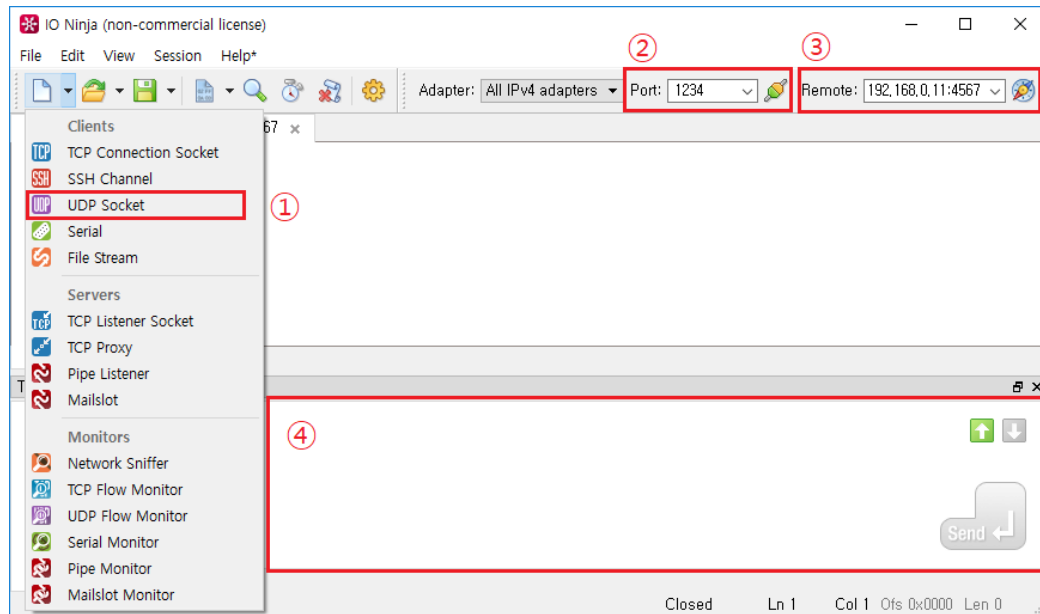


Figure 29: IO Ninja – UDP Socket Setting

## DA16200 DA16600 Host Interface and AT Command

### Appendix A License Information

Mosquitto 1.4.14 License

Eclipse Distribution License 1.0

Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the follow disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the Eclipse Foundation, Inc.

Nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-----  
UMAC GPL License

Linux kernel 3.9.0 rc3 version (backport 4.2.6-1)

## DA16200 DA16600 Host Interface and AT Command

### Appendix B HTTP API Return Values

#### B.1 Return Value as Defined by NetX Duo HTTP

Define	Value	Define	Value
NX_SUCCESS	0x00	NX_RESERVED_CODE1	0x25
NX_NO_PACKET	0x01	NX_SOCKET_UNBOUND	0x26
NX_UNDERFLOW	0x02	NX_NOT_CREATED	0x27
NX_OVERFLOW	0x03	NX_SOCKETS_BOUND	0x28
NX_NO_MAPPING	0x04	NX_NO_RESPONSE	0x29
NX_DELETED	0x05	NX_POOL_DELETED	0x30
NX_POOL_ERROR	0x06	NX_ALREADY_RELEASED	0x31
NX_PTR_ERROR	0x07	NX_RESERVED_CODE2	0x32
NX_WAIT_ERROR	0x08	NX_MAX_LISTEN	0x33
NX_SIZE_ERROR	0x09	NX_DUPLICATE_LISTEN	0x34
NX_OPTION_ERROR	0x0A	NX_NOT_CLOSED	0x35
NX_DELETE_ERROR	0x10	NX_NOT_LISTEN_STATE	0x36
NX_CALLER_ERROR	0x11	NX_IN_PROGRESS	0x37
NX_INVALID_PACKET	0x12	NX_NOT_CONNECTED	0x38
NX_INVALID_SOCKET	0x13	NX_WINDOW_OVERFLOW	0x39
NX_NOT_ENABLED	0x14	NX_ALREADY_SUSPENDED	0x40
NX_ALREADY_ENABLED	0x15	NX_DISCONNECT_FAILED	0x41
NX_ENTRY_NOT_FOUND	0x16	NX_STILL_BOUND	0x42
NX_NO_MORE_ENTRIES	0x17	NX_NOT_SUCCESSFUL	0x43
NX_ARP_TIMER_ERROR	0x18	NX_UNHANDLED_COMMAND	0x44
NX_RESERVED_CODE0	0x19	NX_NO_FREE_PORTS	0x45
NX_WAIT_ABORTED	0x1A	NX_INVALID_PORT	0x46
NX_IP_INTERNAL_ERROR	0x20	NX_INVALID_RELISTEN	0x47
NX_IP_ADDRESS_ERROR	0x21	NX_CONNECTION_PENDING	0x48
NX_ALREADY_BOUND	0x22	NX_TX_QUEUE_DEPTH	0x49
NX_PORT_UNAVAILABLE	0x23	NX_NOT_IMPLEMENTED	0x4A
NX_NOT_BOUND	0x24	NX_NOT_SUPPORTED	0x4B
NX_INVALID_INTERFACE	0x4C	NX_DUPLICATED_ENTRY	0x52
NX_INVALID_PARAMETERS	0x4D	NX_PACKET_OFFSET_ERROR	0x53
NX_NOT_FOUND	0x4E	NX_OPTION_HEADER_ERROR	0x54
NX_CANNOT_START	0x4F	NX_CONTINUE	0x55
NX_NO_INTERFACE_ADDRESS	0x50	NX_PARAMETER_ERROR	0xFF
NX_INVALID_MTU_DATA	0x51		

## DA16200 DA16600 Host Interface and AT Command

### B.2 Return Value as Defined by LWIP HTTP

Define	Value	Define	Value
ERR_OK	0	ERR_ISCONN	-10
ERR_MEM	-1	ERR_CONN	-11
ERR_BUF	-2	ERR_IF	-12
ERR_TIMEOUT	-3	ERR_ABRT	-13
ERR_RTE	-4	ERR_RST	-14
ERR_INPROGRESS	-5	ERR_CLSD	-15
ERR_VAL	-6	ERR_ARG	-16
ERR_WOULDBLOCK	-7	ERR_UNKNOWN	-17
ERR_USE	-8	ERR_NOT_FOUND	-18
ERR_ALREADY	-9		



## DA16200 DA16600 Host Interface and AT Command

### Appendix C User UART Configuration

#### C.1 How to Run AT-CMD on UART2

AT-CMD is configured to use the UART1 interface by default and can be configured to use the UART2 interface. To configure AT-CMD to use the UART2 interface, modify config\_generic\_sdk.h as shown in bold below:

```
// AT-CMD service
#define __SUPPORT_ATCMD__
...
#if defined ( __SUPPORT_ATCMD__ )
    #undef __ATCMD_IF_UART1__          // AT-CMD over UART1
    #define __ATCMD_IF_UART2__          // AT-CMD over UART2
    ...
    #undef __USER_UART_CONFIG__ // Support Customer's UART configuration
    #undef __ATCMD_IF_SPI__      // AT-CMD over SPI
    #undef __ATCMD_IF_SDIO__     // AT-CMD over SDIO
    #endif /* __SUPPORT_ATCMD__ */
...
```

#### C.2 User UART Configuration

There is a feature called User UART Configuration that is enabled by `__USER_UART_CONFIG__`.

When the SDK is built with `__USER_UART_CONFIG__` defined, the UART settings for the AT-CMD interface can be configured. In this case, ATB will not be available.

For example, to run AT-CMD on UART2 with a static baud rate of 230400, the SDK should be configured as shown in bold below.

```
// config_generic_sdk.h
...
// AT-CMD service
#define __SUPPORT_ATCMD__
...
#if defined ( __SUPPORT_ATCMD__ )
    #undef __ATCMD_IF_UART1__ // AT-CMD over UART1
    #define __ATCMD_IF_UART2__ // AT-CMD over UART2
    ...
    #define __USER_UART_CONFIG__ // Support Customer's UART configuration
    ...
    #endif /* __SUPPORT_ATCMD__ */
...

// user_interface.c
...
#if defined ( __USER_UART_CONFIG__ )
/*
 * Customer configuration for AT-CMD UART
 */
uart_info_t ATCMD_UART_config_info =
{
    UART_BAUDRATE_230400,      /* baud */
    UART_DATABITS_8,           /* bits */
    UART_PARITY_NONE,          /* parity */
    UART_STOPBITS_1,           /* stopbit */
}
```

## DA16200 DA16600 Host Interface and AT Command

```

        UART_FLOWCTL_OFF          /* flow control */
};
#endif // __USER_UART_CONFIG__
...

```

With changes above, when DA16200/DA16600 boots, AT-CMD is initialized in baud rate 230400 by default and cannot change at run time.

### C.3 Use Case

// \_\_USER\_UART\_CONFIG\_\_ **disabled**

- Baud rate (and other parameters) configurable by NVRAM
- ATB available, UART Setting can change at run-time without SDK rebuild
- Example Use case
  - MCU: Run on UART in baud rate 115200
  - MCU: Run ATF
  - DA16200/DA16600: AT-CMD is initialized in 115200
  - MCU: ATB=230400
  - MCU: Now it should change its UART baud rate to 230400 to communicate with DA16200/DA16600

// \_\_USER\_UART\_CONFIG\_\_ **enabled**

- AT-CMD UART's baud rate (and other parameters) is configurable statically
- ATB NOT available
- Example Use Case
  - DA16200/DA16600: DA16200/DA16600 boots and AT-CMD is initialized in 230400 by default now.
  - MCU: Start on UART in baud rate 230400
  - MCU: AT-CMD operation ...

## DA16200 DA16600 Host Interface and AT Command

### Appendix D DA16200/DA16600 Cipher Suites

No.	Cipher Suite Supported by DA16200/DA16600	Hex Code
1	TLS_RSA_WITH_AES_128_CBC_SHA	2F
2	TLS_RSA_WITH_AES_256_CBC_SHA	35
3	TLS_RSA_WITH_AES_128_CBC_SHA256	3C
4	TLS_RSA_WITH_AES_256_CBC_SHA256	3D
5	TLS_RSA_WITH_AES_128_GCM_SHA256	9C
6	TLS_RSA_WITH_AES_256_GCM_SHA384	9D
7	TLS_RSA_WITH_AES_128_CCM	C09C
8	TLS_RSA_WITH_AES_256_CCM	C09D
9	TLS_RSA_WITH_AES_128_CCM_8	C0A0
10	TLS_RSA_WITH_AES_256_CCM_8	C0A1
11	TLS_RSA_WITH_DES_CBC_SHA	9
12	TLS_DHE_RSA_WITH_AES_128_CBC_SHA	33
13	TLS_DHE_RSA_WITH_AES_256_CBC_SHA	39
14	TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	67
15	TLS_DHE_RSA_WITH_AES_256_CBC_SHA256	6B
16	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	9E
17	TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	9F
18	TLS_DHE_RSA_WITH_AES_128_CCM	C09E
19	TLS_DHE_RSA_WITH_AES_256_CCM	C09F
20	TLS_DHE_RSA_WITH_AES_128_CCM_8	C0A2
21	TLS_DHE_RSA_WITH_AES_256_CCM_8	C0A3
22	TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA	16
23	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	C011
24	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	C014
25	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	C027
26	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	C028
27	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	C02F
28	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	C030
29	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	C012
30	TLS_ECDH_RSA_WITH_AES_128_CBC_SHA	C00E
31	TLS_ECDH_RSA_WITH_AES_256_CBC_SHA	C00F
32	TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256	C029
33	TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384	C02A
34	TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256	C031
35	TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384	C032
36	TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA	C00D
37	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	C009

## DA16200 DA16600 Host Interface and AT Command

No.	Cipher Suite Supported by DA16200/DA16600	Hex Code
38	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	C00A
39	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	C023
40	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	C024
41	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	C02B
42	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	C02C
43	TLS_ECDHE_ECDSA_WITH_AES_128_CCM	C0AC
44	TLS_ECDHE_ECDSA_WITH_AES_256_CCM	C0AD
45	TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8	C0AE
46	TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8	C0AF
47	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	C008
48	TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA	C004
49	TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA	C005
50	TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256	C025
51	TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384	C026
52	TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256	C02D
53	TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384	C02E
54	TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA	C003

## DA16200 DA16600 Host Interface and AT Command

### Appendix E Reason Code For Wi-Fi Connection Failure or Disconnection

SDK v3.x.x.x: core\wifistack\supplicant\src\common\ieee802\_11\_defs.h.

1. /* Reason codes (IEEE Std 802.11-2016, 9.4.1.7, Table 9-45) */	
2. #define WLAN_REASON_UNSPECIFIED	1
3. #define WLAN_REASON_PREV_AUTH_NOT_VALID	2
4. #define WLAN_REASON_DEAUTH_LEAVING	3
5. #define WLAN_REASON_DISASSOC_DUE_TO_INACTIVITY	4
6. #define WLAN_REASON_DISASSOC_AP_BUSY	5
7. #define WLAN_REASON_CLASS2_FRAME_FROM_NONAUTH_STA	6
8. #define WLAN_REASON_CLASS3_FRAME_FROM_NONASSOC_STA	7
9. #define WLAN_REASON_DISASSOC_STA_HAS_LEFT	8
10. #define WLAN_REASON_STA_REQ_ASSOC_WITHOUT_AUTH	9
11. /* IEEE 802.11h */	
12. #define WLAN_REASON_PWR_CAPABILITY_NOT_VALID	10
13. #define WLAN_REASON_SUPPORTED_CHANNEL_NOT_VALID	11
14. #define WLAN_REASON_BSS_TRANSITION_DISASSOC	12
15. /* IEEE 802.11i */	
16. #define WLAN_REASON_INVALID_IE	13
17. #define WLAN_REASON_MICHAEL_MIC_FAILURE	14
18. #define WLAN_REASON_4WAY_HANDSHAKE_TIMEOUT	15
19. #define WLAN_REASON_GROUP_KEY_UPDATE_TIMEOUT	16
20. #define WLAN_REASON_IE_IN_4WAY_DIFFERS	17
21. #define WLAN_REASON_GROUP_CIPHER_NOT_VALID	18
22. #define WLAN_REASON_PAIRWISE_CIPHER_NOT_VALID	19
23. #define WLAN_REASON_AKMP_NOT_VALID	20
24. #define WLAN_REASON_UNSUPPORTED_RSN_IE_VERSION	21
25. #define WLAN_REASON_INVALID_RSN_IE_CAPAB	22
26. #define WLAN_REASON_IEEE_802_1X_AUTH_FAILED	23
27. #define WLAN_REASON_CIPHER_SUITE_REJECTED	24
28. #define WLAN_REASON_TDLS_TEARDOWN_UNREACHABLE	25
29. #define WLAN_REASON_TDLS_TEARDOWN_UNSPECIFIED	26
30. #define WLAN_REASON_SSP_REQUESTED_DISASSOC	27
31. #define WLAN_REASON_NO_SSP_ROAMING_AGREEMENT	28
32. #define WLAN_REASON_BAD_CIPHER_OR_AKM	29
33. #define WLAN_REASON_NOT_AUTHORIZED_THIS_LOCATION	30
34. #define WLAN_REASON_SERVICE_CHANGE_PRECLUDES_TS	31
35. #define WLAN_REASON_UNSPECIFIED_QOS_REASON	32
36. #define WLAN_REASON_NOT_ENOUGH_BANDWIDTH	33
37. #define WLAN_REASON_TDLS_TEARDOWN_UNSPECIFIED	26
38. /* IEEE 802.11e */	
39. #define WLAN_REASON_DISASSOC_LOW_ACK	34

## DA16200 DA16600 Host Interface and AT Command

40. #define WLAN_REASON_EXCEEDED_TXOP	35
41. #define WLAN_REASON_STA_LEAVING	36
42. #define WLAN_REASON_END_TS_BA_DLS	37
43. #define WLAN_REASON_UNKNOWN_TS_BA	38
44. #define WLAN_REASON_TIMEOUT	39
45. #define WLAN_REASON_PEERKEY_MISMATCH	45
46. #define WLAN_REASON_AUTHORIZED_ACCESS_LIMIT_REACHED	46
47. #define WLAN_REASON_EXTERNAL_SERVICE_REQUIREMENTS	47
48. #define WLAN_REASON_INVALID_FT_ACTION_FRAME_COUNT	48
49. #define WLAN_REASON_INVALID_PMKID	49
50. #define WLAN_REASON_INVALID_MDE	50
51. #define WLAN_REASON_INVALID_FTE	51
52. #define WLAN_REASON_MESH_PEERING_CANCELLED	52
53. #define WLAN_REASON_MESH_MAX_PEERS	53
54. #define WLAN_REASON_MESH_CONFIG_POLICY_VIOLATION	54
55. #define WLAN_REASON_MESH_CLOSE_RCVD	55
56. #define WLAN_REASON_MESH_MAX_RETRIES	56
57. #define WLAN_REASON_MESH_CONFIRM_TIMEOUT	57
58. #define WLAN_REASON_MESH_INVALID_GTK	58
59. #define WLAN_REASON_MESH_INCONSISTENT_PARAMS	59
60. #define WLAN_REASON_MESH_INVALID_SECURITY_CAP	60
61. #define WLAN_REASON_MESH_PATH_ERROR_NO_PROXY_INFO	61
62. #define WLAN_REASON_MESH_PATH_ERROR_NO_FORWARDING_INFO	62
63. #define WLAN_REASON_MESH_PATH_ERROR_DEST_UNREACHABLE	63
64. #define WLAN_REASON_MAC_ADDRESS_ALREADY_EXISTS_IN_MBSS	64
65. #define WLAN_REASON_MESH_CHANNEL_SWITCH_REGULATORY_REQ	65
66. #define WLAN_REASON_MESH_CHANNEL_SWITCH_UNSPECIFIED	66

---

## DA16200 DA16600 Host Interface and AT Command

### Appendix F Fast-reconnect on SLEEP Mode 2

When Wi-Fi STA tries to connect to an AP after a power-on by power-switch or RTC timer wakeup on SLEEP mode 2, it requires some time to make a Wi-Fi connection.

To reduce this Wi-Fi connection time between Wi-Fi AP and STA, the DA16200/DA16600 provides the Fast-connection function. In AT-CMD of the DA16200/DA16600, this function is enabled by default.

#### F.1 Technical overview

- **Direct Probe Request for Wi-Fi SCAN**

During the Wi-Fi SETUP processing after factory status, associated channel number will be saved in NVRAM automatically after success Wi-Fi connect.

After saving the connected channel number to NVRAM, when reconnecting to Wi-Fi is attempted, the total Wi-Fi SCAN time to connect is reduced because only the registered channel number is scanned without performing full-channel scan for Wi-Fi connection.

- **Network Address without DHCP Client Procedure**

The DA16200/DA16600 obtains an IP address by DHCP Client procedure when the first Wi-Fi connection is completed after Factory-reset. The DHCP Client operation may take a lot of time in some cases.

To reduce DHCP Client procedure time, the DA16200/DA16600 saves the IP address, subnet mask, gateway address, and DNS address information in NVRAM after a successful DHCP client procedure, and changes to STATIC IP mode internally. STATIC IP mode removes DHCP processing time and improves connection speed.

---

**DA16200 DA16600 Host Interface and AT Command****Appendix G Bluetooth® LE Coexistence Feature**

The Bluetooth® LE Coexistence feature is defined as follows:

- DA16200 AT command image: Bluetooth® LE Coexistence feature is disabled
- DA16600 AT command image: Bluetooth® LE Coexistence feature is enabled (3-Pin interface)

If the Bluetooth® LE Coexistence feature or the 1-Pin interface are required, the SDK must be rebuilt. For more details see Ref. [\[2\]](#).



## DA16200 DA16600 Host Interface and AT Command

### Appendix H Wi-Fi Passive-Scan

A client can use two scanning methods: active and passive. During an active scan, the client radio sends a probe request and receives a probe response from an AP. With a passive scan, the client radio listens for beacons periodically sent by the AP on each channel. A passive scan generally takes more time, since the client must listen and wait for a beacon, than actively probing to find an AP. Another limitation with a passive scan is that if the client does not wait long enough on a channel, then the client may miss an AP beacon.

The DA16200/DA16600 supports active scan and passive scan. It accepts both probe responses and beacons. The DA16200 passive scan consists of frequency and time remaining on the channel, and the result is delivered to the host firmware within 10 ms.

#### H.1 Passive-Scan with specified Channel and Scan-time Limit

The Wi-Fi component should be able to perform a passive Wi-Fi scan that only scans a given list of channels in given time limit. The DA16200 provides passive scan command with ATCMD.

- Related command is AT+WFPSCAN

#### H.2 Passive-Scan Result

During a passive scan, the Wi-Fi component should be able to report each beacon signal to the host firmware within 10 ms received because of the passive scan. The format which is reported to host firmware is as follows:

- BSSID SSID RSSI Security Type Wi-Fi Channel

#### H.3 Passive-Scan Stop

The Wi-Fi component should be able to stop an ongoing passive Wi-Fi scan and any power use associated with the scan within 100 ms of receiving a Wi-Fi beacon signal that meets the following criteria. Beacon BSSID matches given pattern AND either of the following:

- Beacon RSSI is greater than the minimum threshold, OR
- Beacon RSSI is less than the maximum threshold

When it stops scanning by condition, it prints out "+PSCAN:CONDITIONMET".

Related commands are "AT+WFPCTMIN", "AT+WFPCTMAX" and "AT+WFPSTOP".

#### H.4 Passive Scan Sequence

This section describes basic procedure for passive scan between the DA16200/DA16600 and a PC. Run the DA16200 AT commands on a serial terminal application on the local PC. The terminal must be connected to the UART1 interface of the DA16200.

1. Enable ATCMD feature in "config\_generic\_sdk.h":  
**#define \_\_SUPPORT\_ATCMD\_\_**
2. Set passive scan condition using ATCMD:  
**ATCMD: AT+WFPCTMIN=72:5d:cc:d0:82:bc,-80**
3. Start passive scan using ATMD:  
**ATCMD: AT+WFPSCAN=120000,1,3,5**
4. Stop passive scan using ATMD:  
**ATCMD: AT+WFPSTOP**
5. Check passive scan report; see [Figure 30](#).

## DA16200 DA16600 Host Interface and AT Command

```

c8:5b:a0:05:23:43      2412    -48    360_V5P      [WPA-PSK-CCMP][WPA2-PSK-CCMP][ESS]
3c:a3:15:05:de:72      2412    -33    Z10-2509N    [WPA2-PSK-CCMP][WPS][ESS]
08:bd:43:a8:54:16      2417    -40    M_N300_OPEN  [ESS]
04:5e:a4:85:6e:86      2412    -31    NETIS_MEX01  [WPA2-PSK+SAE-CCMP][WPS][ESS]
68:77:24:4e:29:72      2412    -37    TPLINK_TL-XDR3010 [WPA-PSK-CCMP][WPA2-PSK-CCMP][ESS]
72:77:24:4e:29:72      2412    -37    [WPA-PSK-CCMP][WPA2-PSK-CCMP][ESS]
00:be:d5:e3:3a:22      2412    -51    H3C_N12      [WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]
62:58:6d:bd:33:24      2412    -59    HUAWEI_WS5200_new_V4 [WPA2-PSK-CCMP][WPS][ESS]
70:5d:cc:8b:49:8e      2412    -38    Gen_Port_*.5_AP [WPA2-PSK-CCMP][WPS][ESS]

```

Figure 30: Passive Scan Report

## DA16200 DA16600 Host Interface and AT Command

### Appendix I Detailed Error Codes for AT Command

Table 60: AT Command Error Codes

Category	Value	Error Code	Description
Common	0	AT_CMD_ERR_CMD_OK	OK, no error
	-1	AT_CMD_ERR_UNKNOWN_CMD	Unknown command
	-2	AT_CMD_ERR_INSUFFICIENT_ARGS	Insufficient parameter
	-3	AT_CMD_ERR_TOO_MANY_ARGS	Too many parameters
	-4	AT_CMD_ERR_WRONG_ARGUMENTS	Wrong parameter value
	-5	AT_CMD_ERR_NOT_SUPPORTED	Unsupported function
	-6	AT_CMD_ERR_NOT_CONNECTED	Not connected to an AP
	-7	AT_CMD_ERR_NO_RESULT	No result
	-8	AT_CMD_ERR_TOO_LONG_RESULT	Response buffer overflow
	-9	AT_CMD_ERR_INSUFFICIENT_CONFIG	Function is not configured
	-10	AT_CMD_ERR_TIMEOUT	Command timeout
	-11	AT_CMD_ERR_NVR_WRITE	NVRAM write failure
	-12	AT_CMD_ERR_RTM_WRITE	Retention memory write failure
	-13	AT_CMD_ERR_SYS_BUSY	System busy
	-14	AT_CMD_ERR_MEM_ALLOC	Memory allocation failure
	-20	AT_CMD_ERR_DATA_TX	Data Tx failure
	-22	AT_CMD_ERR_IP_ADDRESS	IP address get failure
	-100	AT_CMD_ERR_COMMON_SYS_MODE	Wrong system running mode
	-110	AT_CMD_ERR_COMMON_ARG_TYPE	Wrong argument type
	-111	AT_CMD_ERR_COMMON_ARG_RANGE	Argument int-value range error
	-112	AT_CMD_ERR_COMMON_ARG_LEN	Argument value length error
FLASH	-113	AT_CMD_ERR_COMMON_WRONG_CC	Wrong country-code
	-114	AT_CMD_ERR_COMMON_WRONG_MAC_ADDR	Wrong MAC address
	-170	AT_CMD_ERR_SFLASH_READ	SFLASH driver read failure
	-171	AT_CMD_ERR_SFLASH_WRITE	SFLASH drive write failure
	-172	AT_CMD_ERR_SFLASH_ERASE	SFLASH driver erase failure
	-173	AT_CMD_ERR_SFLASH_ACCESS	SFLASH driver access failure

## DA16200 DA16600 Host Interface and AT Command

Category	Value	Error Code	Description
NVRAM	-180	AT_CMD_ERR_NVRAM_READ	NVRAM driver read failure
	-181	AT_CMD_ERR_NVRAM_WRITE	NVRAM driver write failure
	-182	AT_CMD_ERR_NVRAM_ERASE	NVRAM driver erase failure
	-183	AT_CMD_ERR_NVRAM_DIGIT	-
	-184	AT_CMD_ERR_NVRAM_SAME_MAC	-
	-185	AT_CMD_ERR_NVRAM_CANCELLED	-
	-186	AT_CMD_ERR_NVRAM_INVALID	-
	-187	AT_CMD_ERR_NVRAM_UNKNOWN	-
	-188	AT_CMD_ERR_NVRAM_NOT_SAVED_VALUE	NVRAM name does not exist
Basic	-200	AT_CMD_ERR_BASIC_ARG_NULL_PTR	Not used in AT-CMD module
	-201	AT_CMD_ERR_BASIC_ARG_DATE	Argument "Date" format failure
	-202	AT_CMD_ERR_BASIC_ARG_TIME	Argument "Time" format failure
	-203	AT_CMD_ERR_BASIC_ARG_TIME_ETC	Argument Time value failure
UART	-220	AT_CMD_ERR_UART_INTERFACE	Not defined UART type
	-221	AT_CMD_ERR_UART_BAUDRATE	Argument "BaudRate" failure
	-222	AT_CMD_ERR_UART_DATABITS	Argument "DataBits" failure
	-223	AT_CMD_ERR_UART_PARITY	Argument "Parity" failure
	-224	AT_CMD_ERR_UART_STOPBIT	Argument "StopBits" failure
	-225	AT_CMD_ERR_UART_FLOWCTRL	Argument "FlowCtrl" failure
	-226	AT_CMD_ERR_UART_BAUDRATE_NV_WR	NVRAM Write failure - Baudrate
	-227	AT_CMD_ERR_UART_DATABITS_NV_WR	NVRAM Write failure - DataBits
	-228	AT_CMD_ERR_UART_PARITY_NV_WR	NVRAM Write failure - Parity
	-229	AT_CMD_ERR_UART_STOPBIT_NV_WR	NVRAM Write failure - StopBit
	-230	AT_CMD_ERR_UART_FLOWCTRL_NV_WR	NVRAM Write failure - FlowCtrl
DPM	-300	AT_CMD_ERR_DPM_MODE_DISABLED	DPM operation is not enabled
	-301	AT_CMD_ERR_DPM_SLEEP_STARTED	DPM sleep function is already running

## DA16200 DA16600 Host Interface and AT Command

Category	Value	Error Code	Description
	-302	AT_CMD_ERR_DPM_FAST_CON N_EN	Fast-connection function is enabled
	-303	AT_CMD_ERR_DPM_USER_RTM _ALLOC	Failed to allocate memory in user area of RTM
	-304	AT_CMD_ERR_DPM_USER_RTM _DUP	Same task name already exists
	-305	AT_CMD_ERR_DPM_MODE_AR G	Wrong argument type: DPM flag
	-306	AT_CMD_ERR_DPM_NVRAM_FL AG_ARG	Wrong argument type: NVSRM flag
	-309	AT_CMD_ERR_DPM_SLP2_PERI OD_TYPE	Wrong argument type: Period
	-310	AT_CMD_ERR_DPM_SLP2_PERI OD_RANGE	Wrong argument value range: Period
	-311	AT_CMD_ERR_DPM_SLP2_RTM _FLAG_ARG	Wrong argument type: RTM flag
	-312	AT_CMD_ERR_DPM_SLP1_RTM _FLAG_RANGE	Wrong argument value range: RTM flag
	-313	AT_CMD_ERR_DPM_SLP1_RTM _FLAG_ARG	Wrong argument type: RTM flag
	-314	AT_CMD_ERR_DPM_SLP1_RTM _FLAG_RANGE	Wrong argument value range: RTM flag
	-315	AT_CMD_ERR_DPM_ABN_ARG	Wrong argument type: DPMABN
	-316	AT_CMD_ERR_DPM_SLP2_DPM _MODE_ENABLED	Wrong system mode: DPM mode
	-317	AT_CMD_ERR_DPM_SLP3_PERI OD_TYPE	Wrong argument t type: Period
	-318	AT_CMD_ERR_DPM_SLP3_PERI OD_RANGE	Wrong argument value range: Period
Wi-Fi	-400	AT_CMD_ERR_WIFI_NOT_CONN ECTED	Not connected to AP
	-401	AT_CMD_ERR_WIFI_RUN_MODE _TYPE	Wrong argument type
	-402	AT_CMD_ERR_WIFI_RUN_MODE _RANGE	Wrong argument value range
	-403	AT_CMD_ERR_WIFI_MAC_ADDR	Wrong string type for MAC address
	-404	AT_CMD_ERR_WIFI_WPS_PIN_N UM	Wrong PIN number for WPS connection
	-406	AT_CMD_ERR_WIFI_SCAN_UN SUPPORTED	SCAN command not supported
	-407	AT_CMD_ERR_WIFI_PSCAN_FR EQ_RANGE	Wrong argument value range: Frequency
	-408	AT_CMD_ERR_WIFI_PSCAN_CM AX_RANGE	Wrong argument value: Max RSSI threshold
	-409	AT_CMD_ERR_WIFI_PSCAN_CM IN_RANGE	Wrong argument value: Min RSSI threshold

## DA16200 DA16600 Host Interface and AT Command

Category	Value	Error Code	Description
	-410	AT_CMD_ERR_WIFI_JAP_SSID_NO_VALUE	SSID information not found in NVRAM
	-411	AT_CMD_ERR_WIFI_JAP_SSID_LEN	Too long SSID string (Max length: 32 bytes)
	-412	AT_CMD_ERR_WIFI_JAP_SECU_ARG_TYPE	Wrong argument type: Auth
	-413	AT_CMD_ERR_WIFI_JAP_SECU_ARG_RANGE	Wrong argument value range: Auth
	-414	AT_CMD_ERR_WIFI_JAP_OPEN_TOO_MANY_ARG	Too many arguments for OPEN-mode
	-415	AT_CMD_ERR_WIFI_JAP_OPEN_HIDDEN_TYPE	Wrong argument type (OPEN): Hidden flag
	-416	AT_CMD_ERR_WIFI_JAP_OPEN_HIDDEN_RANGE	Wrong argument value (OPEN): Hidden flag
	-417	AT_CMD_ERR_WIFI_JAP_SECU_HIDDEN_TYPE	Wrong argument type (Security): Hidden flag
	-418	AT_CMD_ERR_WIFI_JAP_SECU_HIDDEN_RANGE	Wrong argument value (Security): Hidden flag
	-419	AT_CMD_ERR_WIFI_JAP_WEP_IDX_TYPE	Wrong argument type: WEP Index
	-420	AT_CMD_ERR_WIFI_JAP_WEP_IDX_RANGE	Wrong argument value range: WEP Index
	-421	AT_CMD_ERR_WIFI_JAP_WEP_KEY_LEN	Wrong argument: WEP key length
	-422	AT_CMD_ERR_WIFI_JAP_WPA_MODE_TYPE	Wrong argument type: Encrypt
	-423	AT_CMD_ERR_WIFI_JAP_WPA_MODE_RANGE	Wrong argument value range: Encrypt
	-424	AT_CMD_ERR_WIFI_JAP_WPA_KEY_LEN	Wrong argument: WPA PSK length
	-425	AT_CMD_ERR_WIFI_JAP_SSID_NO_VALUE	SSID information not found in NVRAM
	-426	AT_CMD_ERR_WIFI_JAP_SSID_LEN	Too long SSID string (Max length: 32 bytes)
	-427	AT_CMD_ERR_WIFI_JAP_WPA_PSK_LEN	Wrong argument: WPA PSK length
	-428	AT_CMD_ERR_WIFI_JAP_WEP_NOT_SUPPORT	Not supported security mode: WEP-mode
	-429	AT_CMD_ERR_WIFI_JAP_HIDDEN_TYPE	Wrong argument type: Hidden flag
	-430	AT_CMD_ERR_WIFI_JAP_HIDDEN_RANGE	Wrong argument value range: Hidden flag
	-431	AT_CMD_ERR_WIFI_JAP_WPA3_MODE_TYPE	Wrong argument type: WPA3 flag
	-432	AT_CMD_ERR_WIFI_JAP_WPA3_MODE_RANGE	Wrong argument value range: WPA3 flag

## DA16200 DA16600 Host Interface and AT Command

Category	Value	Error Code	Description
	-433	AT_CMD_ERR_WIFI_JAPA_WPA3_HIDDEN_TYPE	Wrong argument type: Hidden flag
	-434	AT_CMD_ERR_WIFI_JAPA_WPA3_HIDDEN_RANGE	Wrong argument value range: Hidden flag
	-435	AT_CMD_ERR_WIFI_ROAP_ROAM_TYPE	Wrong argument type
	-436	AT_CMD_ERR_WIFI_ROAP_ROAM_RANGE	Wrong argument value range
	-437	AT_CMD_ERR_WIFI_ENTAP_SSID_NO_VALUE	SSID information not found in NVRAM
	-438	AT_CMD_ERR_WIFI_ENTAP_SSID_LEN	Too long SSID string (Max length: 32 bytes)
	-439	AT_CMD_ERR_WIFI_ENTAP_AUTH0_UNSupport	Unsupported security mode
	-440	AT_CMD_ERR_WIFI_ENTAP_ENCRYPT0_UNSupport	Unsupported encrypt mode
	-441	AT_CMD_ERR_WIFI_ENTAP_EAP_PHASE1	Unsupported EAP Phase #1 value
	-442	AT_CMD_ERR_WIFI_ENTAP_EAP_PHASE2	Wrong argument value range: EAP Phase #2
	-443	AT_CMD_ERR_WIFI_ENTAP_SECURITY_MODE	Wrong argument value range: Auth
	-444	AT_CMD_ERR_WIFI_ENTAP_ENCRYPT_MODE	Wrong argument value range: Encrypt
	-445	AT_CMD_ERR_WIFI_ENTAP_EAP_MODE	Wrong argument value range: EAP Phase #1
	-446	AT_CMD_ERR_WIFI_ENTAP_EAP_ID_NO_VALUE	Login ID information not found in NVRAM
	-447	AT_CMD_ERR_WIFI_ENTAP_EAP_ID_LEN	Too long ID string (Max length: 64 bytes)
	-448	AT_CMD_ERR_WIFI_ENTAP_EAP_PWD_LEN	Too long PWD string (Max length: 64 bytes)
	-449	AT_CMD_ERR_WIFI_SOFTAP_SSID_NO_VALUE	SSID for Soft AP not found in NVRAM
	-450	AT_CMD_ERR_WIFI_SOFTAP_SECURITY_MODE	Wrong argument value: Security
	-451	AT_CMD_ERR_WIFI_SOFTAP_ENCRYPT_MODE	Wrong argument value range: Encrypt
	-452	AT_CMD_ERR_WIFI_SOFTAP_CHANNEL_VALUE_TYPE	Wrong argument type: Channel
	-453	AT_CMD_ERR_WIFI_SOFTAP_CHANNEL_VALUE_RANGE	Wrong argument value range: Channel
	-454	AT_CMD_ERR_WIFI_SOFTAP_OPEN_TOO_MANY_ARG	Too many arguments for OPEN-mode
	-455	AT_CMD_ERR_WIFI_SOFTAP_CHANNEL_TX_PWR_VALUE	Wrong channel Tx-power value

## DA16200 DA16600 Host Interface and AT Command

Category	Value	Error Code	Description
	-456	AT_CMD_ERR_WIFI_SOFTAP_WEP_NOT_SUPPORT	Unsupported security mode on Soft AP
	-457	AT_CMD_ERR_WIFI_SOFTAP_ENC_MODE_TYPE	Wrong argument type: Encrypt
	-458	AT_CMD_ERR_WIFI_SOFTAP_ENC_MODE_RANGE	Wrong argument value range: Encrypt
	-459	AT_CMD_ERR_WIFI_SOFTAP_PASSKEY_LEN	Too short/long PSK length (Length: 8 ~ 63 bytes)
	-460	AT_CMD_ERR_WIFI_ALREADY_CONNECTED	Wi-Fi session already connected
	-461	AT_CMD_ERR_WIFI_CONCURRENT_NO_PROFILE	Concurrent-mode profile information not found in NVRAM
	-462	AT_CMD_ERR_WIFI_PSCAN_DURATION	Duration value out of range
	-463	AT_CMD_ERR_WIFI_JAPAWPA3_PSK_LEN	Too short/long PSK length (Length: 8 ~ 63 bytes )
	-464	AT_CMD_ERR_WIFI_SOFTAP_OWE_TOO_MANY_ARG	Too many arguments for OWE
	-465	AT_CMD_ERR_WIFI_SOFTAP_SSID_LEN	Too long SSID string (Max length: 32 bytes)
	-466	AT_CMD_ERR_WIFI_ENTAP_WPA_HIDDEN_TYPE	Wrong argument type: Hidden flag
	-467	AT_CMD_ERR_WIFI_ENTAP_WPA_HIDDEN_RANGE	Wrong argument value range: Hidden flag
CLI	-500	AT_CMD_ERR_WIFI_CLI_STATUS	Failed to run "cli status" command
	-501	AT_CMD_ERR_WIFI_CLI_SET_NETWORK	Failed to run "cli set_network 0"
	-502	AT_CMD_ERR_WIFI_CLI_SET_NETWORK_HIDDEN	Failed to run "cli set_network 0" w/hidden flag
	-503	AT_CMD_ERR_WIFI_CLI_SELECT_NETWORK	Failed to run "cli select_network 0"
	-504	AT_CMD_ERR_WIFI_CLI_SAVE_CONF	Failed to run "cli save_config"
	-505	AT_CMD_ERR_WIFI_CLI_SAVE_CONF_HIDDEN	Failed to run "cli save_config" w/hidden flag
	-506	AT_CMD_ERR_WIFI_CLI_DISCONNECT	Failed to run "cli disconnect"
	-507	AT_CMD_ERR_WIFI_CLI_DEAUTHENTICATE	Failed to run "cli deauthenticate"
	-508	AT_CMD_ERR_WIFI_CLI_DISASSOCIATE	Failed to run "cli disassociate"
	-510	AT_CMD_ERR_WIFI_CLI_WPS_PBC_ANY	Failed to run "cli wps_pbc any"
	-511	AT_CMD_ERR_WIFI_CLI_WPS_PIN_GET	Failed to run "cli wps_pin get"



## DA16200 DA16600 Host Interface and AT Command

Category	Value	Error Code	Description
	-512	AT_CMD_ERR_WIFI_CLI_WPS_PIN_ANY	Failed to run "cli wps_pin any"
	-513	AT_CMD_ERR_WIFI_CLI_WPS_PIN_NUM	Wrong argument: PIN value (Length: 8 bytes)
	-514	AT_CMD_ERR_WIFI_CLI_WPS_CANCEL	Failed to run "cli wps_cancel"
	-515	AT_CMD_ERR_WIFI_CLI_COUNTRY	Failed to run "cli country"
	-516	AT_CMD_ERR_WIFI_CLI_PSCAN_CH_TL	Failed to run "cli passive_scan chan_time_limit"
	-517	AT_CMD_ERR_WIFI_CLI_PSCAN_STOP	Failed to run "cli passive_scan_stop"
	-518	AT_CMD_ERR_WIFI_CLI_PSCAN_CMAX_GET	Failed to run "cli passive_scan_condition_max"
	-519	AT_CMD_ERR_WIFI_CLI_PSCAN_CMAX_SET	Failed to run "cli passive_scan_condition_max ..."
	-520	AT_CMD_ERR_WIFI_CLI_PSCAN_CMIN_GET	Failed to run "cli passive_scan_condition_min"
	-521	AT_CMD_ERR_WIFI_CLI_PSCAN_CMIN_SET	Failed to run "cli passive_scan_condition_min ..."
	-522	AT_CMD_ERR_WIFI_CLI_SOFTAP_START	Failed to run "cli ap start"
	-523	AT_CMD_ERR_WIFI_CLI_SOFTAP_STOP	Failed to run "cli ap stop"
	-524	AT_CMD_ERR_WIFI_CLI_SOFTAP_RESTART	Failed to run "cli ap restart"
Network Basic	-600	AT_CMD_ERR_NW_NET_IF_NOT_INITIALIZE	Network interface does not initialize
	-601	AT_CMD_ERR_NW_NET_IF_IS_DOWN	Network interface is DOWN
	-602	AT_CMD_ERR_NW_IP_IFACE_TYPE	Wrong argument type: interface
	-603	AT_CMD_ERR_NW_IP_IFACE_RANGE	Wrong argument value range: interface
	-604	AT_CMD_ERR_NW_IP_ADDR_CLASS	Invalid IP address class
	-605	AT_CMD_ERR_NW_IP_INVALID_ADDR	Invalid IP address type
	-606	AT_CMD_ERR_NW_IP_NETMASK	Invalid Netmask address type
	-607	AT_CMD_ERR_NW_IP_GATEWAY	Invalid Gateway address type
	-608	AT_CMD_ERR_NW_DNS_A_QUERY_FAIL	Failed to get IP address by DNS Query

## DA16200 DA16600 Host Interface and AT Command

Category	Value	Error Code	Description
	-609	AT_CMD_ERR_NW_PING_IFACE_ARG_TYPE	Wrong argument type: Interface
	-610	AT_CMD_ERR_NW_PING_IFACE_ARG_RANGE	Wrong argument value range: Interface
	-611	AT_CMD_ERR_NW_PING_DST_ADDR	Invalid destination IP address
	-612	AT_CMD_ERR_NW_PING_TX_COUNT	Wrong argument: Ping Tx count
DHCP Client	-613	AT_CMD_ERR_NW_DHCPC_START_FAIL	Failed to start DHCP client
	-614	AT_CMD_ERR_NW_DHCPC_HOSTNAME_LEN	Too long DHCP hostname (Max length: 32 bytes)
	-615	AT_CMD_ERR_NW_DHCPC_HOSTNAME_TYPE	Wrong format for DHCP hostname
DHCP Server	-616	AT_CMD_ERR_NW_DHCPS_START_ADDR_NOT_EXIST	IP pool start-address not found in NVRAM
	-617	AT_CMD_ERR_NW_DHCPS_END_ADDR_NOT_EXIST	IP pool end-address not found in NVRAM
	-618	AT_CMD_ERR_NW_DHCPS_WRONG_START_IP_CLASS	Invalid start IP address class
	-619	AT_CMD_ERR_NW_DHCPS_WRONG_END_IP_CLASS	Invalid end IP address class
	-620	AT_CMD_ERR_NW_DHCPS_IPADDR_RANGE_MISMATCH	Mismatch IP address class range
	-621	AT_CMD_ERR_NW_DHCPS_IPADDR_RANGE_OVERFLOW	Exceed IP address pool count (Max. 10)
	-622	AT_CMD_ERR_NW_DHCPS_NO_CONNECTED_CLIENT	No connected client information to DHCP server
	-623	AT_CMD_ERR_NW_DHCPS_RUN_FLAG_TYPE	Wrong argument type: dhcpd flag
	-624	AT_CMD_ERR_NW_DHCPS_RUN_FLAG_VAL	Wrong argument value range: dhcpd flag
	-625	AT_CMD_ERR_NW_DHCPS_LEASE_TIME_TYPE	Wrong argument type: dhcpd lease_time

## DA16200 DA16600 Host Interface and AT Command

Category	Value	Error Code	Description
	-626	AT_CMD_ERR_NW_DHCPD_LEASE_TIME_RANGE	Wrong argument value range: dhcpd lease_time
SNTP Client	-629	AT_CMD_ERR_NW_SNTP_NOT_SUPPORTED	SNTP client does not supported
	-630	AT_CMD_ERR_NW_SNTP_FLAG_TYPE	Wrong argument type: SNTP flag
	-631	AT_CMD_ERR_NW_SNTP_FLAG_VAL	Wrong argument value range: SNTP flag
	-632	AT_CMD_ERR_NW_SNTP_PERIOD_TYPE	Wrong argument type: SNTP period
	-633	AT_CMD_ERR_NW_SNTP_PERIOD_RANGE	Wrong argument value range: SNTP period
MQTT Client	-634	AT_CMD_ERR_NW_MQTT_NOT_CONNECTED	MQTT client is currently not connected
	-635	AT_CMD_ERR_NW_MQTT_NEED_TO_STOP	Need to disconnect the already connected MQTT session
	-636	AT_CMD_ERR_NW_MQTT_UNKNOWN_OP_ID	Input not supported
	-637	AT_CMD_ERR_NW_MQTT_CLIENT_TASK_START	MQTT client start failed by unknown reason
MQTT Broker	-638	AT_CMD_ERR_NW_MQTT_BROKER_NAME_NOT_FOUND	MQTT broker name not found
	-639	AT_CMD_ERR_NW_MQTT_BROKER_PORT_NUM_TYPE	Wrong argument type: MQTT Broker port
	-640	AT_CMD_ERR_NW_MQTT_BROKER_PORT_NUM_RANGE	Wrong argument value range: MQTT Broker port
	-641	AT_CMD_ERR_NW_MQTT_BROKER_NAME_LEN	MQTT Broker name string: max length (MQTT_BROKER_MAX_LEN) exceeded
MQTT TLS	-642	AT_CMD_ERR_NW_MQTT_TLS_TYPE	Wrong argument type: tls
	-643	AT_CMD_ERR_NW_MQTT_TLS_RANGE	Wrong argument value range: tls
	-644	AT_CMD_ERR_NW_MQTT_TLS_ALPN_NOT_EXIST	ALPN information not found in NVRAM

## DA16200 DA16600 Host Interface and AT Command

Category	Value	Error Code	Description
	-645	AT_CMD_ERR_NW_MQTT_TLS_ALP N_COUNT_TYPE	Wrong argument type: count
	-646	AT_CMD_ERR_NW_MQTT_TLS_ALP N_COUNT_RANGE	Wrong argument value range: count (1 ~ 3)
	-647	AT_CMD_ERR_NW_MQTT_TLS_ALP N_NAME_LEN	Too long ALPN name length (Max: 24 bytes)
	-648	AT_CMD_ERR_NW_MQTT_TLS_SNI NOT_EXIST	SNI information not found in NVRAM
	-649	AT_CMD_ERR_NW_MQTT_TLS_SNI LEN	Too long SNI string length (Max: 64 bytes)
	-650	AT_CMD_ERR_NW_MQTT_TLS_CSU ITE_NUM_NOT_EXIST	CipherSuite count value not found in NVRAM
	-651	AT_CMD_ERR_NW_MQTT_TLS_CSU ITE_NOT_EXIST	CipherSuite information not found in NVRAM
	-652	AT_CMD_ERR_NW_MQTT_TLS_CSU ITE_NUM_NVRAM_WR	Failed to write Cipher Suit count info to NVRAM
	-653	AT_CMD_ERR_NW_MQTT_TLS_CSU ITE_NVRAM_WR	Failed to write Cipher Suit info to NVRAM
MQTT Sub-Topic	-654	AT_CMD_ERR_NW_MQTT_SUBS_T OPIC_NOT_EXIST	SUB topic does not exist in NVRAM
	-655	AT_CMD_ERR_NW_MQTT_SUBS_T OPIC_NUM_TYPE	Wrong argument type: count
	-656	AT_CMD_ERR_NW_MQTT_SUBS_T OPIC_NUM_RANGE	Wrong argument value range: count (1~4)
	-657	AT_CMD_ERR_NW_MQTT_SUBS_T OPIC_LEN	Too long topic string length (Max: 64 bytes)
	-658	AT_CMD_ERR_NW_MQTT_SUBS_T OPIC_DUP	Duplicate Sub-topic string
	-659	AT_CMD_ERR_NW_MQTT_SUBS_T OPIC_NUM_NVRAM_WR	Failed to write topic count to NVRAM
	-660	AT_CMD_ERR_NW_MQTT_SUBS_T OPIC_NUM_OVERFLOW	Adding a topic exceeds the max topic count (4)
	-661	AT_CMD_ERR_NW_MQTT_SUBS_T OPIC_ALREADY_EXIST	Subscribe topic already exists

## DA16200 DA16600 Host Interface and AT Command

Category	Value	Error Code	Description
MQTT Pub-Topic	-662	AT_CMD_ERR_NW_MQTT_PUB_TO PIC_NOT_EXIST	PUB topic not found in NVRAM
	-663	AT_CMD_ERR_NW_MQTT_PUB_TO PIC_LEN	Too long topic string length (Max: 64 bytes)
MQTT WILL Message	-664	AT_CMD_ERR_NW_MQTT_WILL_TO PIC_NOT_EXIST	WILL topic does not exist in NVRAM
	-665	AT_CMD_ERR_NW_MQTT_WILL_M ESSAGE_NOT_EXIST	WILL message not existing in NVRAM
	-666	AT_CMD_ERR_NW_MQTT_WILL_TO PIC_LEN	Too long WILL topic length (Max: 64 bytes)
	-667	AT_CMD_ERR_NW_MQTT_WILL_M ESSAGE_LEN	Too long WILL message length (Max: 64 bytes)
	-668	AT_CMD_ERR_NW_MQTT_WILL_Q OS_TYPE	Wrong argument type: qos
	-669	AT_CMD_ERR_NW_MQTT_WILL_Q OS_RANGE	Wrong argument value range: qos
MQTT Common	-670	AT_CMD_ERR_NW_MQTT_PROTOC OL	Network protocol error occurred with Broker
	-671	AT_CMD_ERR_NW_MQTT_PING_PE RIOD_TYPE	Invalid Ping Period value is invalid
	-672	AT_CMD_ERR_NW_MQTT_PING_PE RIOD_RANGE	Wrong argument value range: (0 ~ 86400)
	-673	AT_CMD_ERR_NW_MQTT_USERNA ME_NOT_EXIST	User name does not exist in NVRAM
	-674	AT_CMD_ERR_NW_MQTT_USERNA ME_LEN	Too long username length (Max: 64 bytes)
	-675	AT_CMD_ERR_NW_MQTT_PASSW ORD_LEN	Too long password length (Max: 160 bytes)
	-676	AT_CMD_ERR_NW_MQTT_PUB_ME SSAGE_LEN	Too long message length (Max: 2048 bytes)
	-677	AT_CMD_ERR_NW_MQTT_PUB_TX_ IN_PROGRESS	Previous message Tx is still in progress
HTTP(s) Server	-680	AT_CMD_ERR_NW_HTTPS_TASK_CRE ATE_FAIL	Failed to create HTTP server task

## DA16200 DA16600 Host Interface and AT Command

Category	Value	Error Code	Description
	-681	AT_CMD_ERR_NW_HTSS_TASK_CREATE_FAIL	Failed to create HTTPs server task
HTTP Client	-682	AT_CMD_ERR_NW_HTC_TASK_CREATE_FAIL	Failed to create HTTP client task
	-683	AT_CMD_ERR_NW_HTC_ALPN_COUNT_TYPE	Wrong argument type: alpn_number
	-684	AT_CMD_ERR_NW_HTC_ALPN_COUNT_RANGE	Wrong argument value range: alpn_number
	-685	AT_CMD_ERR_NW_HTC_ALPN1_STRING_LENGTH	Too long ALPN #1 string length (Max 24 bytes)
	-686	AT_CMD_ERR_NW_HTC_ALPN2_STRING_LENGTH	Too long ALPN #2 string length (Max 24 bytes)
	-687	AT_CMD_ERR_NW_HTC_ALPN3_STRING_LENGTH	Too long ALPN #3 string length (Max 24 bytes)
	-688	AT_CMD_ERR_NW_HTC_SNI_LENGTH	Too long SNI string length (Max 64 bytes)
Web-Socket Client	-689	AT_CMD_ERR_NW_WSC_URL_STRING_LENGTH	Too short URL string length (Min 1 byte)
	-690	AT_CMD_ERR_NW_WSC_INVALID_URL	Invalid URL string
	-691	AT_CMD_ERR_NW_WSC_TASK_ALREADY_EXIST	WebSocket session is already exist
	-692	AT_CMD_ERR_NW_WSC_CB_FUNCTION_DOES_NOT_EXIST	Not registered user Websocket cb-function
	-693	AT_CMD_ERR_NW_WSC_INVALID_STATE	No connected session to disconnect
	-694	AT_CMD_ERR_NW_WSC_TASK_CREATE_FAIL	Failed to create WebSocket client task
	-695	AT_CMD_ERR_NW_WSC_CLOSE_FAIL	Failed to send "Session-Close" frame
	-696	AT_CMD_ERR_NW_WSC_SESSION_NOT_CONNECTED	No connected session to send message
	-697	AT_CMD_ERR_NW_WSC_UNKNOWN_CMD	Unknown WebSocket internal error

## DA16200 DA16600 Host Interface and AT Command

Category	Value	Error Code	Description
OTA	-700	AT_CMD_ERR_NW_OTA_WRONG_FW_TYPE	Wrong argument: fw_type
	-701	AT_CMD_ERR_NW_OTA_DOWN_OK_AND_WAIT_RENEW	Already downloaded
	-702	AT_CMD_ERR_NW_OTA_FLASH_READ_SIZE_TYPE	Wrong argument type: read_addr
	-703	AT_CMD_ERR_NW_OTA_FLASH_COPY_SIZE_TYPE	Wrong argument type: size
	-704	AT_CMD_ERR_NW_OTA_FLASH_ERASE_SIZE_TYPE	Wrong argument type: size
	-705	AT_CMD_ERR_NW_OTA_BY_MCU_INIT	Failed to initialize MCU configuration for OTA
	-706	AT_CMD_ERR_NW_OTA_SET_TLS_AUTH_MODE_NVRAM	Failed to save TLS certificate in NVRAM
	-707	AT_CMD_ERR_NW_OTA_SET_MCU_FW_NAME	Failed to set MCU_FW name. (Max length: 8 bytes)
Zero Config	-710	AT_CMD_ERR_NW_MDNS_WRONG_FLAG	Wrong argument: flag of MDNS
	-711	AT_CMD_ERR_NW_MDNS_WRONG_MODE	Wrong argument: mode of MDNS
	-712	AT_CMD_ERR_NW_MDNS_NOT_RUNNING	MDNS is not running
	-713	AT_CMD_ERR_NW_MDNS_ALREADY_RUN	MDNS is already running
	-714	AT_CMD_ERR_NW_MDNS_IN_PROGRESS	Progressing Probing and Announcing on MDNS
	-715	AT_CMD_ERR_NW_MDNS_UNKNOWN_FAULT	Unknown MDNS internal error
	-716	AT_CMD_ERR_NW_MDNS_START_RUN_MODE_VAL	Invalid interface
	-717	AT_CMD_ERR_NW_MDNS_SOCKET_FAIL	Failed to initialize socket
	-718	AT_CMD_ERR_NW_DNS_SD_NOT_RUNNING	DNS-SD is not running

## DA16200 DA16600 Host Interface and AT Command

Category	Value	Error Code	Description
	-719	AT_CMD_ERR_NW_DNS_SD_ALREADY_RUN	Already DNS-SD is running
	-720	AT_CMD_ERR_NW_DNS_SD_IN_PROCESS	Progressing Probing and Announcing of DNS-SD
	-721	AT_CMD_ERR_NW_DNS_SD_SERVICE_CREATE_FAIL	Failed to register service of DNS-SD
	-722	AT_CMD_ERR_NW_DNS_SD_SERVICE_PARAMS	Invalid parameters to register service
	-723	AT_CMD_ERR_NW_DNS_SD_SERVICE_NAME_NVRAM_WRITE	Failed to write service name to NVRAM
	-724	AT_CMD_ERR_NW_DNS_SD_SERVICE_PROTOCOL_NVRAM_WRITE	Failed to write service protocol to NVRAM
	-725	AT_CMD_ERR_NW_DNS_SD_SERVICE_PORT_NO_NVRAM_WRITE	Failed to write service port to NVRAM
	-726	AT_CMD_ERR_NW_DNS_SD_SERVICE_TXT_EXT_NVRAM_WRITE	Failed to write service TXT to NVRAM
Transport Function (TCP/UDP)	-730	AT_CMD_ERR_TCP_SERVER_LOCAL_PORT_TYPE	Wrong argument: local port of TCP server
	-731	AT_CMD_ERR_TCP_SERVER_MAX_PEER_TYPE	Wrong argument : max allowed peer
	-732	AT_CMD_ERR_TCP_SERVER_TASK_CREATE	Failed to start TCP server
	-733	AT_CMD_ERR_TCP_CLIENT_SERVER_PORT_TYPE	Wrong argument: TCP server port of TCP client
	-734	AT_CMD_ERR_TCP_CLIENT_LOCAL_PORT_TYPE	Wrong argument: local port of TCP client
	-736	AT_CMD_ERR_TCP_CLIENT_TASK_CREATE	Failed to start TCP client
	-737	AT_CMD_ERR_UDP_SESSION_LOCAL_PORT_TYPE	Wrong argument: local port of UDP session
	-738	AT_CMD_ERR_UDP_SESSION_LOCAL_PORT_RANGE	Invalid range of local port of UDP session
	-739	AT_CMD_ERR_UDP_SESSION_TASK_CREATE	Failed to start UDP session



## DA16200 DA16600 Host Interface and AT Command

Category	Value	Error Code	Description
	-740	AT_CMD_ERR_UDP_CID2_SESS_NOT_EXIST	UDP session, CID 2, does not exist
	-741	AT_CMD_ERR_UDP_CID2_ALREADY_EXIST	UDP session, CID 2, already exists
	-742	AT_CMD_ERR_UDP_CID2_SESS_INFO	Invalid UDP session, CID 2, information
	-743	AT_CMD_ERR_UDP_CID2_REMOTE_PORT_TYPE	Invalid remote port of UDP session, CID 2
	-744	AT_CMD_ERR_NO_CONNECTED_SESSION_EXIST	No session information
	-745	AT_CMD_ERR_NO_FOUND_REQ_CID_SESSION	No assigned CID to terminate session
	-746	AT_CMD_ERR_CONTEXT_CID_TYPE	Wrong argument type: cid
	-747	AT_CMD_ERR_CONTEXT_DELETE	Failed to terminate session
	-748	AT_CMD_ERR_CONTEXT_TYPE_IS_NOT_TCP_SVR	Wrong CID value: Not TCP server session
	-749	AT_CMD_ERR_CONTEXT_INVALID_SESS_TYPE	Invalid session type to save session information
	-750	AT_CMD_ERR_TRTRM_CID_TYPE	Wrong argument: CID to terminate session
	-751	AT_CMD_ERR_TRTRM_REMOTE_PORT_NUM_TYPE	Wrong argument type: remote_port
	-752	AT_CMD_ERR_TRTRM_TCP_SVR_REMOTE_SESS_DISCONNECT	Failed to disconnect TCP client from TCP server
	-753	AT_CMD_ERR_TCP_SERVER_TERMINATE	Failed to terminate TCP server
	-754	AT_CMD_ERR_TCP_CLIENT_TERMINATE	Failed to terminate TCP client
	-755	AT_CMD_ERR_UDP_SESSION_TERMINATE	Failed to terminate UDP session
	-756	AT_CMD_ERR_MULTI_SESSION_CID_TERMINATE	No assigned CID to terminate session
	-757	AT_CMD_ERR_NO_SESSION_INFO_TO_SAVE_NVRAM	No session information to save

## DA16200 DA16600 Host Interface and AT Command

Category	Value	Error Code	Description
SSL/TLS	-760	AT_CMD_ERR_SSL_ROLE_NOT_SUPPORT	Not supported role of TLS session
	-761	AT_CMD_ERR_SSL_CONF_CID_TYPE	Wrong argument: CID of TLS session
	-762	AT_CMD_ERR_SSL_CONTEXT_NOT_FOUND	No assigned CID of TLS session
	-763	AT_CMD_ERR_SSL_CONTEXT_ALREADY_EXIST	TLS session is already running to configure
	-764	AT_CMD_ERR_SSL_CONF_ID_NOT_SUPPORTED	Not supported configuration
	-765	AT_CMD_ERR_SSL_SAVE_CLR_ALL_NV	Failed to erase TLS session from NVRAM
	-766	AT_CMD_ERR_SSL_SAVE_FAIL_NV	Failed to save TLS session to NVRAM
	-767	AT_CMD_ERR_SSL_CONF_ID_TYPE	Wrong argument: configuration ID
	-768	AT_CMD_ERR_SSL_CONF_ID_RANGE	Invalid range of configuration ID
	-769	AT_CMD_ERR_SSL_CONF_CID_CA_CERT	CA certification does not exist for assigned CID
	-770	AT_CMD_ERR_SSL_CONF_CID_CERT	Certification does not exist for assigned CID
	-771	AT_CMD_ERR_SSL_CONF_CID_SNI	Failed to configure SNI of assigned CID
	-772	AT_CMD_ERR_SSL_CONF_CID_SVR_VALID_TYPE	Wrong argument: auth mode of assigned CID
	-773	AT_CMD_ERR_SSL_CONF_CID_SVR_VALID_RANGE	Invalid range of auth mode of assigned CID
	-774	AT_CMD_ERR_SSL_CONF_CID_RX_BUF_LEN	Wrong argument: Rx buffer length of CID
	-775	AT_CMD_ERR_SSL_CONF_CID_TX_BUF_LEN	Wrong argument: Tx buffer length of CID
	-776	AT_CMD_ERR_SSL_CONF_CID_TYPE	Wrong argument: CID to configure TLS session
	-777	AT_CMD_ERR_SSL_CONN_ALREADY_CONNECTED	Already TLS session is connected
	-778	AT_CMD_ERR_SSL_CONN_PORT_NUM_TYPE	Wrong argument: peer_port of TLS client

## DA16200 DA16600 Host Interface and AT Command

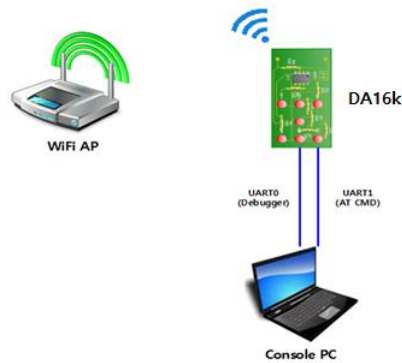
Category	Value	Error Code	Description
	-779	AT_CMD_ERR_SSL_CONN_UNKNO WN_HOSTNAME	Unknown hostname to connect TLS server
	-780	AT_CMD_ERR_SSL_CONN_CFG_SET UP_FAIL	Failed to setup TLS client
	-781	AT_CMD_ERR_SSL_CONN_TLS_CLIE NT_RUN_FAIL	Failed to connect TLS client
SSL Certificate	-782	AT_CMD_ERR_SSL_CERT_TYPE	Wrong argument: type
	-783	AT_CMD_ERR_SSL_CERT_RANGE	Invalid range of certificate type
	-784	AT_CMD_ERR_SSL_CERT_STO_SEQ_ TYPE	Wrong argument: sequence type
	-785	AT_CMD_ERR_SSL_CERT_STO_SEQ_ RANGE	Invalid range of sequence type
	-786	AT_CMD_ERR_SSL_CERT_STO_FOR MAT_TYPE	Wrong argument: format type
	-787	AT_CMD_ERR_SSL_CERT_STO_FOR MAT_RANGE	Invalid range of format type
	-788	AT_CMD_ERR_SSL_CERT_STO_ALRE ADY_EXIST	Already certificate is existed
	-789	AT_CMD_ERR_SSL_CERT_STO_NO_ SPACE	Not enough space to save certificate
	-790	AT_CMD_ERR_SSL_CERT_DEL_LIST_ NOT_FOUND	Not found certificate to delete
	-791	AT_CMD_ERR_SSL_CERT_MODULE	Invalid module
	-792	AT_CMD_ERR_SSL_CERT_FORMAT	Invalid format
	-793	AT_CMD_ERR_SSL_CERT_LENGTH	Invalid length
	-794	AT_CMD_ERR_SSL_CERT_FLASH_A DDR	Invalid address of sflash memory
	-795	AT_CMD_ERR_SSL_CERT_EMPTY_C ERT	No certificate
	-796	AT_CMD_ERR_SSL_CERT_INTERNAL	Internal error
	-999	AT_CMD_ERR_UNKNOWN	Undefined Error

## DA16200 DA16600 Host Interface and AT Command

### Appendix J AT Command Development Environment Configuration

#### J.1 How to Connect the DA16200/DA16600 Board

This section describes the installation procedure for the drivers, the configuration of the serial port, and all necessary steps to set up and check the connection with the PC.



**Figure 31: AT Command Development Environment**

On first connection to a host PC with Microsoft Windows as operating system, the system will detect several devices and will automatically install all necessary drivers. If the driver is not automatically installed, then get the driver from the following URL:  
[http://www.ftdichip.com/Drivers/CDM/CDM21224\\_Setup.zip](http://www.ftdichip.com/Drivers/CDM/CDM21224_Setup.zip).

## DA16200 DA16600 Host Interface and AT Command

There are two virtual COM ports created by the Windows driver. The first COM port (lower number, COM69 in Figure 32) provides a UART interface for debugging or firmware download between the PC and the DA161200. The second (higher number, COM70 in Figure 32) is used for AT command.

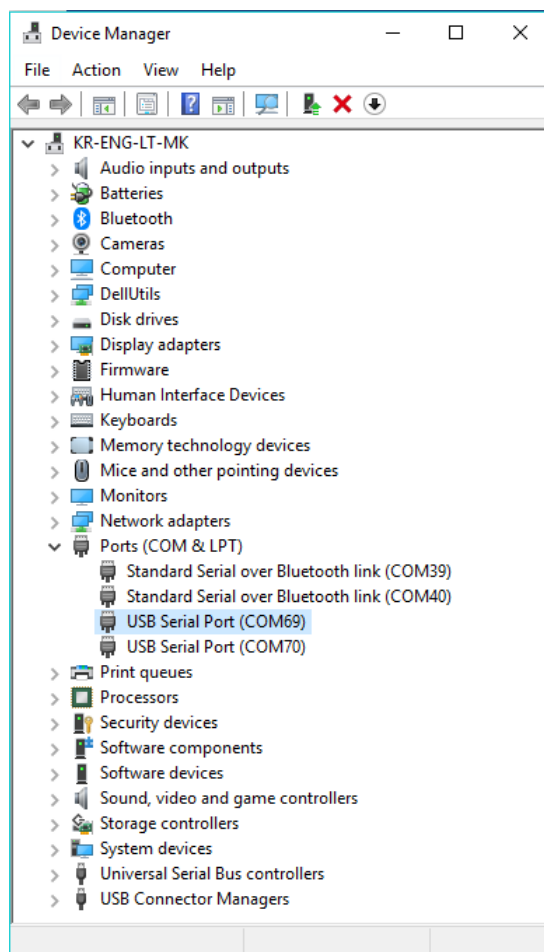


Figure 32: Check COM Ports on Device Manager

### J.2 Configure the Serial Port for UART

On a Windows Host, the utility **Tera Term** is used to connect. See Ref. [1].

**Tera Term** is a free terminal emulator (communication program) that supports multiple communication including serial port connections.

1. Download Tera Term from <https://ttssh2.osdn.jp/>.
2. Run the teraterm-x.yy.exe.
3. Follow the installation wizard.

To make sure that the communication between the DA16200/DA16600 EVK and the host PC is established correctly, check the UART connection between the two nodes. Do the following steps:

1. Use a USB cable to connect the DA16200/DA16600 EVK to the PC.
2. Make sure that the PC discovered the two serial ports in Windows Device Manager as shown in Figure 32. The higher COM port number is connected to UART1.
3. Open **Tera Term** from the Windows Start menu.
4. In the **Tera Term: New connection** Renesas Electronics:
  - a. Select **Serial**.

## DA16200 DA16600 Host Interface and AT Command

- b. Select the COM Port to use.
  - c. Click **OK**.
5. Select **Setup > Serial Port** and configure the UART port with the parameters as shown in [Figure 33](#). Select the higher COM port number as discovered in step 2.

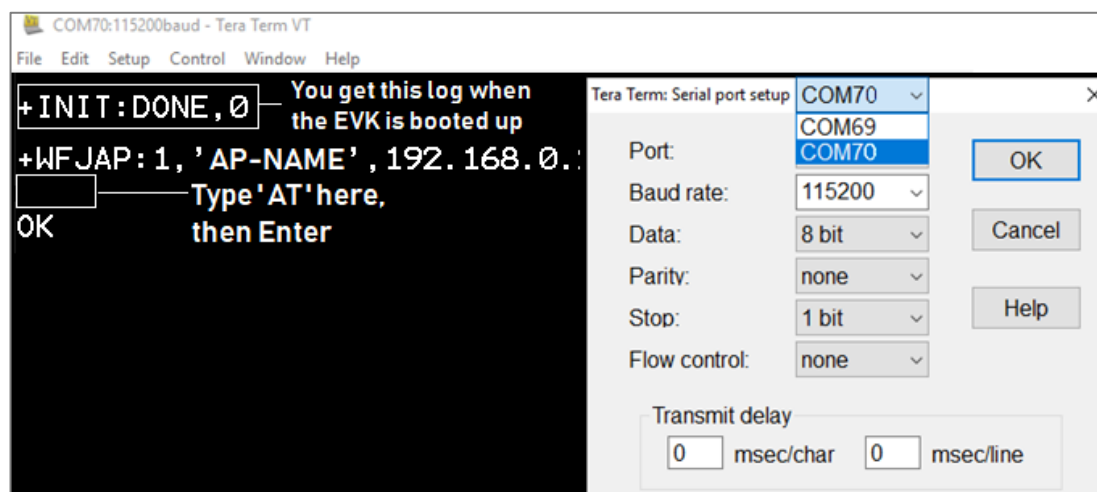


Figure 33: Initial Setup to Get Started with AT Command

### J.3 Configuration for MCU Wake-Up (Optional)

Depending on the application scenarios, both MCU and DA16200/DA16600 may want to be in the SLEEP state and MCU wants to be woken up (by DA16200/DA16600) when DA16200/DA16600 wakes up from DPM Sleep. This can be achieved with the MCU wake-up feature of the DA16200/DA16600.

To use the MCU wake-up feature, connect pin GPIO\_11 of the DA16200/DA16600 to the wake-up pin on the MCU. Then, when the DA16200/DA16600 wakes up, GPIO\_11 becomes an Output and is set to High (Active High) to trigger the wake-up of the MCU. The wake-up PIN of MCU should be configured to detect the rising edge of GPIO\_11 for wake-up.

## DA16200 DA16600 Host Interface and AT Command

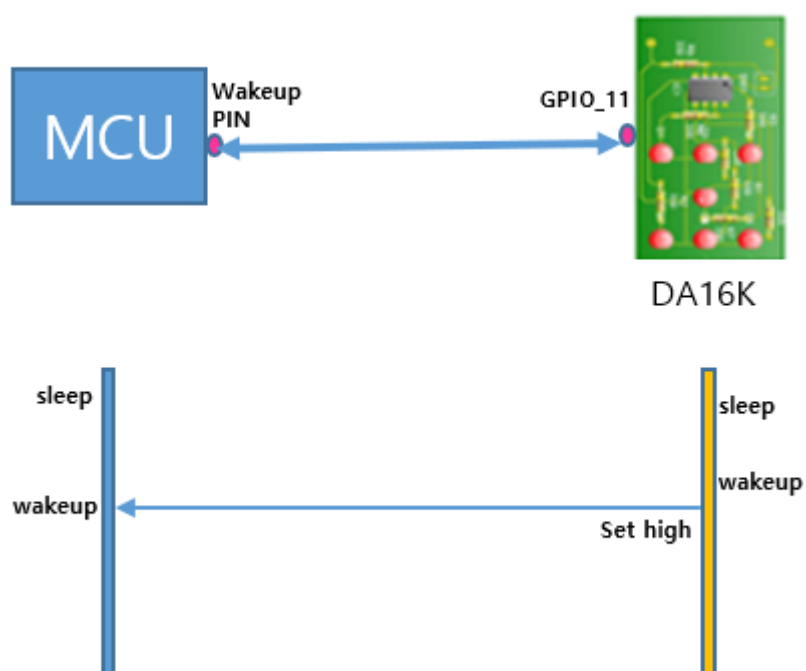


Figure 34: GPIO Wake-Up

## DA16200 DA16600 Host Interface and AT Command

### Revision History

Revision	Date	Description
3.2	Sep. 18, 2023	<ul style="list-style-type: none"> <li>Updated AT command error codes</li> </ul>
3.1	Aug. 18, 2023	<ul style="list-style-type: none"> <li>Updated AT+GPIOSTART and AT+LEDCTRL examples</li> <li>Added error code to Detailed Error Codes for AT Command</li> <li>AT+SETSLEEP2EXT, AT+DPMUSERWU: changed the time input unit from seconds to milliseconds</li> <li>AT+WFENTAP: updated hidden AP settings</li> <li>AT+TRSSLCERTSTORE: added data length parameter</li> <li>AT+TRSSLWR: fixed incorrect optional parameter</li> <li>AT+WFENTLI: fixed incorrect optional parameter</li> <li>AT+NWCCRT: added DH param for Set #1 and 2, and Set #3 for WPA Enterprise</li> <li>&lt;ESC&gt;C: added Set #3 for WPA Enterprise</li> <li>&lt;ESC&gt;CERT: added new AT command in Certificate Command table</li> <li>AT+SETSLEEP2EXT: description updated</li> <li>AT+SETSLEEP3EXT added</li> <li>Updated the usage of padding field in SPI protocol</li> </ul>
3.0	June 30, 2023	<ul style="list-style-type: none"> <li>Updated OTA commands and descriptions in Secure Socket Command List table</li> <li>AT+WFCC: additional note added</li> <li>AT+WFJAP, AT+WFSAP: additional note added on &lt;sec&gt; &lt;enc&gt; for WPA3</li> </ul>
2.17	Apr. 10, 2023	Corrected examples and descriptions of OTA commands
2.16	Feb. 10, 2023	Corrected the range of AT+SETSLEEP2EXT command
2.15	Jan. 27, 2023	Added ESC Command Sequence
2.14	Jan. 12, 2023	<p>Merged user guides and changed the titles.</p> <ul style="list-style-type: none"> <li>UM-WI-003, DA16200 DA16600 AT Command</li> <li>UM-WI-020, DA16200 SPI Host Interface</li> <li>UM-WI-053, DA16200 SDIO Host Interface</li> </ul>
2.13	Dec. 16, 2022	<ul style="list-style-type: none"> <li>AT+NWHTCH added</li> <li>AT+NWHTCTLSAUTH added</li> <li>AT+WFCC: note is added</li> <li>AT+NWMQMSG: note updated on max length</li> <li>AT+WFAPUI: description updated</li> <li>AT+NWIP: note added</li> <li>AT+WFAPCH: valid range changed</li> <li>AT+WFWMP: note updated</li> <li>AT+BLENAM: added to get the BLENAM.</li> <li>AT+WFJAPA3: added to connect to the WPA3-AP</li> <li>MQTT Commands re-arranged: split to MQTT configuration command and MQTT operation commands.</li> <li>Pre-requisite added for MQTT Configuration commands.</li> <li>Updated DA16200/DA16600 Cipher Suites</li> <li>New added Appendix J AT-CMDs error code</li> <li>Typo fixed in the example - AT+WFSPF, AT+WFOTP, &lt;ESC&gt;S</li> </ul>
2.12	Aug. 08, 2022	<ul style="list-style-type: none"> <li>Updated Appendix for running AT command through SDIO or SPI.</li> </ul>



## DA16200 DA16600 Host Interface and AT Command

Revision	Date	Description
2.11	June 14, 2022	<ul style="list-style-type: none"> <li>Added the section of Wi-Fi Function Commands for WPA Enterprise</li> <li>AT+SDKVER added</li> <li>Added the Command Format section</li> <li>Added AT+HOSTINITDONE</li> <li>Added Appendix E</li> <li>Added more info on AT+TZONE</li> <li>Info updated or typo fixed: ATQ, ATB, AT+NWDHIP, AT+WFRSSI</li> <li>Changed default status by some features are enabled</li> <li>Added AT+CHIPNAME, AT+CALWR</li> <li>AT+NWMQMSG : operation response added (+NWMQMSGSEND)</li> <li>Updated +WFJAP:0 and +WFDAP:0 (reason parameter is added)</li> <li>Updated AT+NWMQBR, AT+NWMQTT</li> <li>Updated MQTT optional configuration commands (enabled by default in SDK v3.2.3.0)</li> <li>Added AT+NWMQCS, and CleanSession=0 Guide.</li> <li>AT+WFJAP / AT+WFJAPA / AT+WFSAP : passphrase range added</li> <li>AT+TRTS example updated</li> <li>Added AT+NWMQUTS</li> <li>Updated Note : AT+NWIP, AT+NWDHR, AT+NWDHLT, AT+NWMQMSG, &lt;ESC&gt;S</li> <li>+NWMQCL updated</li> <li>Added Appendix H</li> <li>Added AT+NWWSC</li> <li>Added note for &lt;ESC&gt;S : for ATCMD on SPI</li> </ul>
2.10	Mar. 22, 2022	<ul style="list-style-type: none"> <li>Updated logo, disclaimer, and copyright.</li> <li>Added AT+TCPDATAMODE</li> <li>Added LED/PWM/ADC/I2C related AT commands</li> </ul>
2.9	Dec. 14, 2021	<ul style="list-style-type: none"> <li>AT+NWMQMSG, AT+NWMQTS: updated max length info on topic and added message</li> <li>Added AT+NWDNS, HELP, ATE, and ATQ information</li> </ul>
2.8	Dec. 08, 2021	<ul style="list-style-type: none"> <li>Updated OTP Size Reduced (8 KB-&gt;2 KB) in 5.6.11.2 section</li> <li>Updated Data Transfer Commands section</li> <li>Updated AT+TRSSLWR</li> <li>Added AT+NWMQATS, AT+NWMQDTS, AT+NWMQV311, and AT+NWDHIP</li> <li>Added AT+ NWHTCSNI, AT+ NWHTCALPN, AT+NWHTCSNIDEL, and AT+NWHTCALPNDEL</li> </ul>
2.7	Nov. 25, 2021	Changed the title
2.6	Oct. 28, 2021	Added AT+WFAPUI: <timeout> valid value range <ul style="list-style-type: none"> <li>Updated TCP Client Socket Test section</li> <li>Updated TCP Server section</li> <li>Updated: 5th parameter removed from AT+NWDHS</li> <li>Added: AT+NWDNS2</li> <li>AT+XTALRD, AT+FLASHDUMP: &lt;CR&gt;&lt;LF&gt; is appended to data</li> </ul>
2.5	Sep. 07, 2021	<ul style="list-style-type: none"> <li>Updated table format of ATCMD (Prerequisite, example, note added)</li> <li>Added Zeroconf Commands</li> <li>Added Secure Socket Commands</li> </ul>

## DA16200 DA16600 Host Interface and AT Command

Revision	Date	Description
2.4	June 17, 2021	<ul style="list-style-type: none"> <li>Updated MQTT commands (MQTT Client Connection Example and MQTT TLS Connection Example)</li> <li>AT+NWDHNS deleted (not needed as WAN Port is not available in Soft AP mode)</li> <li>Added AT+NWMQALPN, AT+NWMQSNI, AT+NWMQCSUIT, AT+SETDPMSLP1EXT, AT+DPMABNWFCCNT</li> <li>Updated AT+WFJAP, AT+WFJAPA : Optional parameter &lt;hidden&gt; added</li> </ul>
2.3	Apr. 01, 2021	<ul style="list-style-type: none"> <li>Added OTA update command</li> <li>Added support for SDK V3.x.x.x</li> </ul>
2.2	Mar. 15, 2021	<ul style="list-style-type: none"> <li>Added Appendix B HTTP API Return Values</li> <li>Added Appendix C</li> <li>Added AT+NWMQAUTO and ATB</li> </ul>
2.1	Jan. 13, 2021	Added Wi-Fi Function Commands for WPA3, and updated minor changes
2.0	Dec. 08, 2020	<ul style="list-style-type: none"> <li>Added additional description on the following commands: AT+WFSAP, AT+WFOAP, AT+WFTAP, ATF, AT+WFJAPA, AT+NWMQTT, +NWMQCL, AT+DPM</li> <li>Added new sections: <ul style="list-style-type: none"> <li>Added MQTT Example: Changing Subscription Topic while running</li> <li>Added MQTT Example: Reading Subscription Topic while running</li> </ul> </li> </ul>
1.9	Nov. 11, 2020	<ul style="list-style-type: none"> <li>AT+NWCERT, &lt;ESC&gt;C updated</li> <li>AT+NWSNS updated</li> <li>AT+NWHTS updated</li> <li>AT+NWHTSS updated</li> </ul>
1.8	Aug. 18, 2020	<ul style="list-style-type: none"> <li>Added SNTP commands in Network Function Commands</li> <li>Added HTTP-client command in HTTP-Client Commands</li> <li>Added MCU FW update command using OTA in OTA Commands</li> </ul>
1.7	June 30, 2020	<ul style="list-style-type: none"> <li>Added Configuration for MCU Wake-up</li> <li>Correct typos and wordings</li> </ul>
1.6	Apr. 29, 2020	<ul style="list-style-type: none"> <li>Added AT+WFDIS and AT+SETDPMSLP2EXT</li> <li>Updated MQTT commands to operate with one-port</li> <li>Updated to process the comma in the parameters</li> </ul>
1.5	Apr. 03, 2020	<ul style="list-style-type: none"> <li>Added AT+BIDX for changing boot index</li> <li>Added example code of MQTT commands</li> <li>Updated RF Test function commands</li> <li>Updated GPIO commands</li> </ul>
1.4	Oct. 21, 2019	<ul style="list-style-type: none"> <li>Updated Serial Port configuration steps.</li> <li>Removed draft status</li> </ul>
1.3	Oct. 15, 2019	<ul style="list-style-type: none"> <li>Error correction</li> <li>Added explanation to serial program</li> </ul>
1.2	Oct. 07, 2019	Editorial review and add code: UM-B-111
1.1	July 25, 2019	Added OTP Memory Address for writing MAC address
1.0	July 03, 2019	Preliminary DRAFT Release

## DA16200 DA16600 Host Interface and AT Command

### 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.

## DA16200 DA16600 Host Interface and AT Command

### 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.