

User Manual

DA16200 AT Command

UM-WI-003

Abstract

This document describes the use of the AT commands to configure the DA16200.

DA16200 AT Command

Contents

Abstract	1
Contents	2
Figures	3
Tables	3
1 Terms and Definitions	5
2 References	6
3 Introduction	7
4 AT Command Development Environment Configuration	7
4.1 How to Connect the DA16200 Board	7
4.2 Configure the Serial Port for UART	8
4.3 Configuration for MCU Wakeup (Optional)	10
5 AT Command Format	11
5.1 Basic Command Format	11
5.2 Extended Command Format	11
5.3 Response Format	11
6 Basic Function Commands	13
7 Network Function Commands	17
8 Wi-Fi Function Commands	20
8.1 Wi-Fi Function Commands for WPA3	26
9 Advanced Function Commands	29
9.1 MQTT Commands	29
9.1.1 MQTT Client Connection Example	32
9.1.2 MQTT TLS Connection Example	32
9.1.3 MQTT Example with DPM	33
9.1.4 MQTT Example: Changing Subscription Topic when Running	35
9.1.5 MQTT Example: Reading Subscription Topic when Running	36
9.2 HTTP-Client Commands	36
9.2.1 HTTP-Client Connection Example	37
9.3 HTTP-Server Commands	37
9.3.1 HTTP/HTTPS-Server Start Example	37
9.4 OTA Commands	37
9.4.1 OTA Download Example	40
9.4.2 OTA Download Progress Example	40
9.4.3 OTA Renew Example	40
9.4.4 MCU FW Transport Example	40
9.4.5 SFLASH User Area address setting Example	41
9.4.6 SFLASH READ/COPY/ERASE Example	41
10 Transfer Function Commands	42
10.1 Socket Commands	42
10.2 Data Transfer Commands	43
10.3 Data Transfer with DPM	44
10.3.1 TCP Server	44

DA16200 AT Command

10.3.2	TCP Client	45
10.3.3	UDP Session	45
11	RF Test Function Commands	46
12	System and Peripheral Function Commands.....	48
12.1	SPI Commands	48
12.2	OTP Commands	48
12.3	XTAL Commands.....	50
12.4	Flash Dump Commands	50
12.5	GPIO Commands.....	51
12.6	Sleep Commands.....	52
13	Examples.....	52
13.1	Data Transfer Test	52
13.1.1	TCP Server Socket Test.....	53
13.1.2	TCP Client Socket Test	54
13.1.3	UDP Socket Test	55
Appendix A License Information		56
Appendix B HTTP API Return Values.....		57
Appendix C User UART Configuration.....		59
C.1	How to Run AT-CMD on UART2	59
C.2	User UART Configuration	59
C.3	Use Case	60
Revision History		61

Figures

Figure 1:	AT Command Development Environment	7
Figure 2:	Check COM Ports on Device Manager.....	8
Figure 3:	Serial Port Configuration	9
Figure 4:	GPIO Wakeup	10
Figure 5:	Example Sequence to Initiate MQTT Protocol with DPM	33
Figure 6:	Procedure to Send MQTT Messages.....	34
Figure 7:	Procedure to Process MQTT Messages.....	35
Figure 8:	IO Ninja – TCP Client Socket Setting.....	53
Figure 9:	IO Ninja – TCP Server Socket Setting	54
Figure 10:	IO Ninja – UDP Socket Setting	55

Tables

Table 1:	Error Response Codes.....	12
Table 2:	Basic Function Command List.....	13
Table 3:	Initiation Response List	16
Table 4:	Network Function Command List	17
Table 5:	Certificate Command.....	19
Table 6:	Wi-Fi Function Command List.....	20
Table 7:	Wi-Fi Function Response List	26
Table 8:	List of WPA3-relevant Wi-Fi Function Commands.....	26
Table 9:	MQTT Command List	29
Table 10:	MQTT Response List	31
Table 11:	HTTP-Client Command List	36

DA16200 AT Command

Table 12: HTTP-Client Response List.....	36
Table 13: HTTP-Server Command List.....	37
Table 14: OTA Command List.....	37
Table 15: OTA Response List.....	38
Table 16: OTA Response Code List.....	39
Table 17: Socket Command List.....	42
Table 18: Socket Connection Response List.....	43
Table 19: Data Transmission Command.....	43
Table 20: Data Reception Responses.....	44
Table 21: RF Test Command List.....	46
Table 22: SPI Command List.....	48
Table 23: OTP Command List.....	48
Table 24: OTP Memory Address for Writing MAC Address.....	49
Table 25: Size of Memory by XTAL Offset.....	49
Table 26: XTAL Command List.....	50
Table 27: Flash Dump Command List.....	50
Table 28: GPIO Command List.....	51
Table 29: Sleep Command List.....	52

 DA16200 AT Command

1 Terms and Definitions

AT	Attention
PC	Personal Computer
SDK	Software Development Kit
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
WMM	Wi-Fi Multimedia
HTTP	Hypertext Transfer Protocol
FW	Firmware
URL	Universal Resource Locator
COM	Communication Port
MCU	Micro Controller Unit
CMD	Command
DPM	Dynamic Power Management
NVRAM	Non-Volatile RAM
RTC	Real Time Clock
MQTT	Message Queuing Telemetry Transport
DHCP	Dynamic Host Configuration Protocol
TIM	Traffic Indicator Map
WLAN	Wireless Local Area Network
STA	Station
ICMP	Internet Control Message Protocol
OTA	Over the Air
SNTP	Simple Network Time Protocol
IP	Internet Protocol
CA	Certificate Authority
MAC	Medium Access Control
PBC	Push Button Connection
WPS	Wi-Fi Protected Setup
AP	Access Point
SSID	Service Set Identifier
WPA	Wi-Fi Protected Access
IEEE	Institute of Electrical and Electronics Engineers
RTS	Request to Send
WEP	Wired Equivalent Privacy
QoS	Quality of Service
CRC	Cyclic Redundancy Check
SLIB	System Library
RTOS	Real Time Operating System
TCP	Transport Control Protocol
UDP	User Datagram Protocol
CID	Client ID
LMAC	Low MAC
RX	Receive
TX	Transmit
CW	Continuous Wave
PER	Packet Error Rate

DA16200 AT Command

SPI	Serial Peripheral Interface
OTP	One Time Programmable memory
ASCII	American Standard Code for Information Interchange
XTAL	Crystal
GPIO	General Purpose Input Output
TLS	Transport Layer Security

2 References

- [1] DA16200 EVK User Manual, Dialog Semiconductor.

DA16200 AT Command

3 Introduction

The DA16200 AT Command is a command to control smart modem products of Hayes Microcomputer and other compatible modems. AT Command is applied to almost all standardized modems that are available at present. The original title is Hayes Command and is also called AT Command since the command begins mostly with AT. AT is an abbreviation of "attention", which means to take note of or fix one's sight upon something. An example of an AT Command is `ATZ`. With this command the modem becomes initialized and returns to a state with no command input.

Hayes Command has a quite simple command structure of AT+Command. This is convenient as you can just put any desired command right after AT, even more than one command in a row.

Most of the Hayes Modem Commands have a command right before AT except "+++", which is an independently used command. You may put a series of commands together whether they contain a capital letter, a small letter, or a blank.

4 AT Command Development Environment Configuration

4.1 How to Connect the DA16200 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.

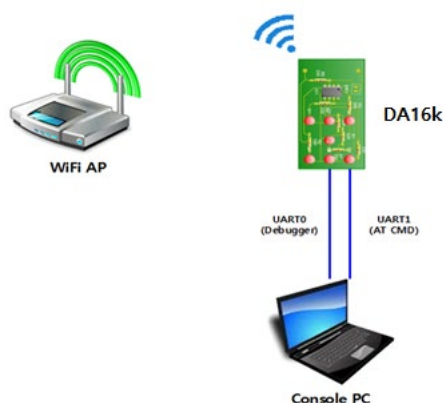


Figure 1: AT Command Development Environment

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.

There are two virtual COM ports created by the Windows driver. The first COM port (lower number, COM35 in [Figure 2](#)) provides a UART interface for debugging or firmware download between the PC

DA16200 AT Command

and the DA161200. The second (higher number, COM70 in [Figure 2](#)) is used for AT COMMAND as shown in [Figure 2](#).

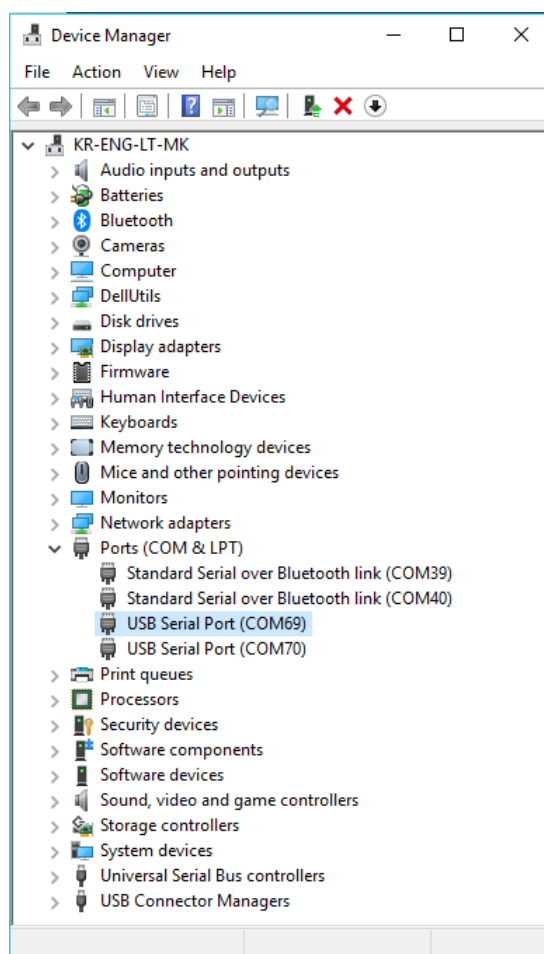


Figure 2: Check COM Ports on Device Manager

4.2 Configure the Serial Port for UART

On a Windows Host, the utility **Tera Term** is used to connect to the DA16200 EVK [\[1\]](#).

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

1. Download Tera Term from <https://tssh2.osdn.jp/>.
2. Run the teraterm-x.yy.exe.
3. Follow the installation wizard.

To make sure that the communication between the DA16200 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 EVK to the PC.
2. Make sure that the PC discovered the two serial ports in Windows Device Manager as shown in [Figure 2](#). 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** dialog:
 - a. Select **Serial**.
 - b. Select the COM Port to use.
 - c. Click **OK**.

DA16200 AT Command

5. Select **Setup > Serial Port** and configure the UART port with the parameters as shown in [Figure 3](#). Select the higher COM port number as discovered in step 2.

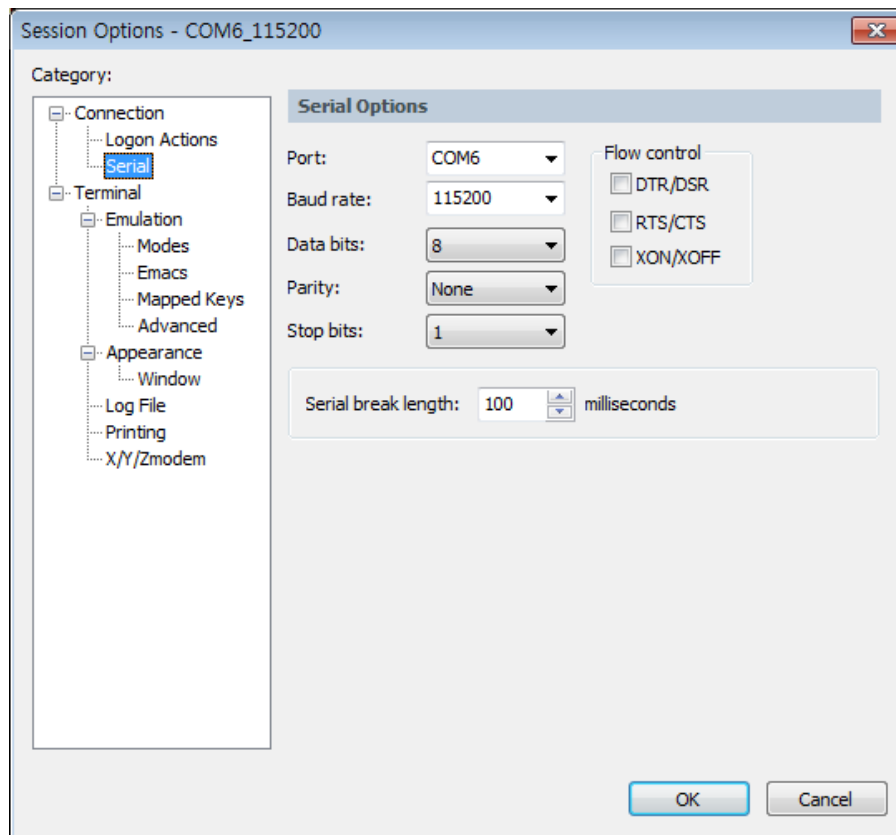


Figure 3: Serial Port Configuration

4.3 Configuration for MCU Wakeup (Optional)

Depending on the application, an MCU may want to be in the SLEEP state while the DA16200 is in the Sleep mode and wake up when the DA16200 wakes up from DPM Sleep.

To use the MCU wakeup feature, connect pin GPIO_11 of the DA16200 to an MCU PIN that is used for wakeup. Then, when the DA16200 wakes up, GPIO_11 is put in the Output mode and set to High (Active High). The wakeup PIN of MCU should be configured to detect the rising edge of GPIO_11 for wakeup.

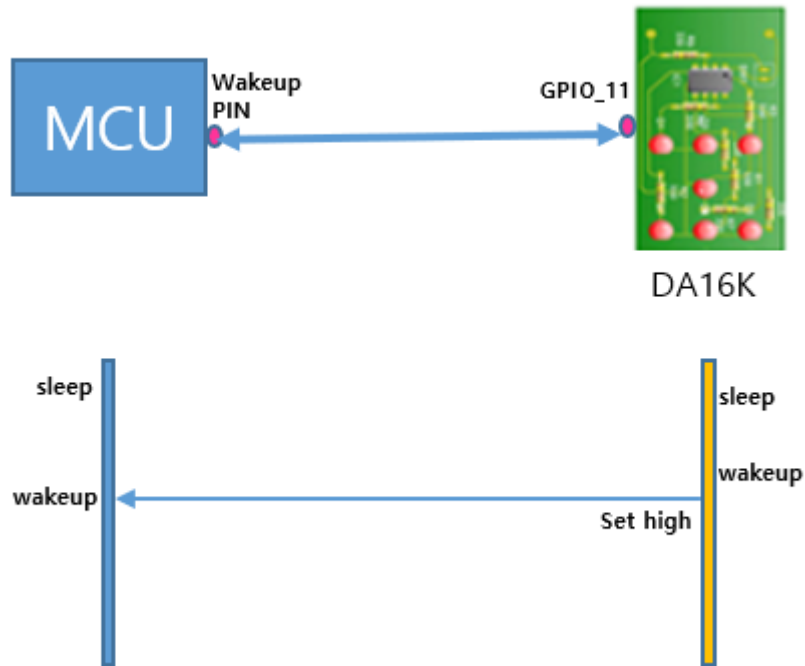


Figure 4: GPIO Wakeup

DA16200 AT Command

5 AT Command Format

5.1 Basic Command Format

- Write CMD

Basic command execution.

ATXX

For example: ATZ

OK

- Read CMD

Get the parameter values of the command.

ATXX=?

For example: ATQ=?

Display result on

OK

5.2 Extended Command Format

- 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 comma(s) or single-quote(s) is(are) used in the parameter, then a single-quote is required before and after.

For example: AT+WEJAP='ssid,comma'single-quote',3,1,'password'

OK

NOTE
The use of the phrase ', (single-quote and comma next to each other) in the parameter is prohibited.

- 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

5.3 Response Format

- Startup response

The AT command response when DA16200 is reset.

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

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

DA16200 AT Command

- 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>

- Normal response

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

- Error response

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

- Extended response

Extended response gives the command setting values and is followed by 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/ERROR**) and *extended* response (also known as **Operation Result**).

- Error response codes

Table 1: Error Response Codes

Code	Description
-1	Unknown command
-2	Insufficient parameter
-3	Too many parameters
-4	Wrong parameter value
-5	Unsupported function
-6	Not connected to an AP
-7	No result
-8	Response buffer overflow
-9	Function is not configured
-10	Command timeout
-11	NVRAM write failure
-12	Retention memory write failure
-99	Unknown error

DA16200 AT Command

6 Basic Function Commands

Table 2: Basic Function Command List

Command	Parameters	Description
?	(none)	Same with the "Help" command.
help	(none)	Show AT command usage.
AT+	(none)	AT command list.
ATZ	(none)	Initialize AT command.
ATF	(none)	DA16200 factory reset. All Wi-Fi profile (Soft-AP or STA) settings are removed and DUT restarts. Response: "+INIT:DONE,0" No Wi-Fi access since all NVRAM parameters are removed.
ATE	(none)	ECHO on/off.
ATQ	(none)	Set displaying result on/off.
ATB	<baudrate> [[,<databits>] [,<parity>] [,<stopbits>] [,<flow control>]]	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 <flow control>: [optional], 0 (Default)/1 For example: ATB=230400 ATB will not be available if __USER_UART_CONFIG__ is enabled in SDK. See Appendix C .
AT+RESTART	(none)	System restart.
AT+RESET	(none)	System reset. Go to the Boot mode ([MROM] prompt).
AT+VER	(none)	Get version info. For SDK V2.x.x.x Response: +VER:<ramlib version>,<ptim version>,<main version> For SDK V3.x.x.x Response: +VER:<main version>
AT+TIME	<date>,<time>	Set the current time. <date>: yyyy-mm-dd <time>: hh:mm:ss Response: OK or ERROR For example: AT+TIME=2016-12-14,11:40:00
	?	Get the current time. Response: +TIME:<yyyy-mm-dd> <hh:mm:ss>
AT+RLT	(none)	Get system running time. Response: +RLT:<days>,<hh:mm:ss>
AT+TZONE	<sec>	GMT Time zone setting (-43200 ~ 43200). <sec>: Time zone setting parameter. Response: OK or ERROR For example: AT+TZONE=32400 (+09:00; KR)

DA16200 AT Command

Command	Parameters	Description
	?	Get GMT Time zone parameter. Response: +TZONE:<sec>
AT+DEFAP	(none)	Factory reset and setup Soft-AP mode with the default configuration. To initialize the Soft-AP interface, the system will reboot automatically. <ul style="list-style-type: none"> • SSID: DA16200_xxxxxx (using a part of the MAC address) • Authentication: WPA2/CCMP • IP address: 10.0.0.1 • DHCP server: Enabled • ... Response: OK or ERROR (reboot)
AT+BIDX	<idx>	Set Boot index. <idx>: Boot index (0 or 1). Response: OK or ERROR For example: AT+BIDX=1
	?	Get the current Boot index. Response: +BIDX:<0 1>
AT+DPM	<dpm>	Set DPM on/off. To change the use of DPM, the system will reboot automatically. <dpm>: 0 (Off), 1 (On). Response: OK or ERROR For example: AT+DPM=1
	?	Get the current DPM setting. Response: +DPM:<0 1>

DA16200 AT Command

Command	Parameters	Description
		<p>Note the following expected operation of DA16200 by running this command:</p> <ul style="list-style-type: none"> ● DA16200 is restarted if AT command format is OK <ul style="list-style-type: none"> ○ +INIT:DONE,0 message is sent when DA16200 boots up ○ If usage of the AT command is not valid, then DA16200 sends ERROR message without restarting ● DA16200 tries to connect to the AP if Wi-Fi connection information is stored in NVRAM <ul style="list-style-type: none"> ○ +WFJAP:0 or +WFJAP:1,'<SSID>',<IP Address> as result of Wi-Fi connection ○ If Wi-Fi connection information such as SSID and key is NOT stored correctly in the DA16200 NVRAM, then +WFJAP:x response is NOT sent ● If MQTT is configured, DA16200 tries to connect to the MQTT broker after Wi-Fi connection is established. The Operation Result – +NWMQCL:0 or +NWMQCL:1 – is sent over UART1 as a result ● DA16200 operates DPM if it is set to 1 (TRUE) <ul style="list-style-type: none"> ○ If Wi-Fi connection is NOT established, then DA16200 enter an abnormal DPM operation ○ ** Abnormal DPM: While DA16200 operates in DPM sleep, DA16200 executes an Abnormal DPM mode if DA16200 is disconnected from the home AP for some reason. Abnormal DPM works like this: if DA16200 is woken up by the Abnormal DPM RTC timer, DA16200 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 ○ If the Wi-Fi connection is established and MQTT connection is NOT established (if MQTT is enabled), DA16200 tries to connect to the MQTT broker several times and enters DPM mode (sleep) based on MQTT's abnormal DPM operation
AT+DPMKA	<period>	<p>Set DPM keepalive period. <period>: Keepalive period (millisecond, 0 ~ 600000). Response: OK or ERROR For example: AT+DPMKA=30000</p>
AT+DPMTIMWU	<count>	<p>Set DPM TIM wakeup count. <count>: TIM wakeup count (1 ~ 65535). Response: OK or ERROR For example: AT+DPMTIMWU=10</p>
AT+DPMUSERWU	<time>	<p>Set DPM user wakeup time. <time>: User wakeup period (second, 0 ~ 86400). Response: OK or ERROR For example: AT+DPMUSERWU=3600</p>
AT+CLRDPM_SLP_EXT	(none)	<p>Set the user application not to enter DPM sleep in case of external wakeup. The AT command host system should execute this command within 200 ms after waking up the DA16200 with the external wakeup pin, otherwise, the DA16200 will go into DPM sleep. Response: OK or ERROR</p>

DA16200 AT Command

Command	Parameters	Description
AT+ SETDPMSLPTEXT	(none)	Set the user application ready to enter DPM sleep in case of external wakeup. After an external wakeup pin wakes up the DA16200, the AT command host system should execute this command after the jobs are finished. Response: OK or ERROR
AT+ SETDPMSLP2EXT	(none)	Set the user application ready to enter DPM Sleep 2 mode in case of external wakeup. After an external wakeup pin wakes up DA16200, the AT command host system should execute this command after the jobs are finished. <period>: wakeup timeout, in second(s) <retain_dpm_memory>: 1 (retain), 0 (not retain) Response: OK or ERROR For example: AT+ SETDPMSLP2EXT=5,1 // wakeup in 5 seconds after DPM Sleep 2. During DPM Sleep2, DPM memory is retained.
AT+MCUWUDONE	(none)	Notify that the MCU wakes up completely. After this command is received, DA16200 starts to send messages to the MCU (that is, MCU should send this command immediately after executing "External wakeup".) Response: OK or ERROR

Table 3: Initiation Response List

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

DA16200 AT Command

7 Network Function Commands

Table 4: Network Function Command List

Command	Parameters	Description
AT+NWIP	<iface>,<ip_addr>, <netmask>,<gw>	Set 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 For example: AT+NWIP=0,192.168.0.100,255.255.255.0,192.168.0.1
	?	Get IP address of the current WLAN interface.
	(none)	Response: +NWIP: <iface>,<ip_addr>,<netmask>,<gw>
AT+NWDNS	<dns_ip>	Set DNS server IP address of STA interface. <dns_ip>: DNS server IP address. Response: OK or ERROR For example: AT+NWDNS=8.8.8.8
	?	Get DNS server IP address of STA interface.
	(none)	Response: +NWDNS:<dns_ip>
AT+NWHOOST	<name>	Get the host IP address by name. <name>: Domain name. Response: +NWHOOST:<ip> For example: AT+NWHOOST=https://www.dialog-semiconductor.com/
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>,<rcv_count>, <avg_time>,<min_time>,<max_time> For example: AT+NWPING=0,192.168.0.1,4
AT+NWDHC	<dhcpc>	Start/Stop DHCP client. <dhcpc>: 0 (stop), 1 (start). Response: OK or ERROR For example: AT+NWDHC=1
	?	Get DHCP client status.
	(none)	Response: +NWDHC:<dhcpc>
AT+NWDHR	<start_ip>,<end_ip>	Set IP address range of DHCP server. <start_ip>: Starting IP address assigned by DHCP server. <end_ip>: Ending IP address assigned by DHCP server. Response: OK or ERROR For example: AT+NWDHR=10.0.0.2,10.0.0.11
	?	Get IP address range of DHCP server.
	(none)	Response: +NWDHR:<start_ip>,<end_ip>

DA16200 AT Command

Command	Parameters	Description
AT+NWDHLT	<lease_time>	Set IP lease time (in seconds) of DHCP server. <lease_time>: IP lease time. Response: OK or ERROR For example: AT+NWDHLT=1800
	?	Get IP lease time of DHCP server.
	(none)	Response: +NWDHLT:<lease_time>
AT+NWDHDNS	<dns_ip>	Set DNS server IP address of DHCP server. <dns_ip>: DNS server IP address. Response: OK or ERROR For example: AT+NWDHDNS=8.8.8.8
	?	Get DNS server IP address of DHCP server.
	(none)	Response: +NWDHDNS:<dns_ip>
AT+NWDHS	<dhcpd>	Start/Stop DHCP server. <dhcpd>: 0 (stop), 1 (start). Response: OK or ERROR For example: AT+NWDHS=1
	<dhcpd>, <start_ip>,<end_ip>, <lease_time>, <dns_ip>	Start DHCP server with options. <dhcpd>: 1 (start). <start_ip>: Starting IP address for DHCP client. <end_ip>: Ending IP address for DHCP client. <lease_time>: IP lease time (optional, in second, default is 1800). <dns_ip>: DNS server IP address (optional). Response: OK or ERROR For example: AT+NWDHS=1,10.0.0.2,10.0.0.10,1800,168.126.63.1
	?	Get DHCP client status.
	(none)	Response: +NWDHS:<dhcpd>
AT+NWSNS AT+NWSNS1 AT+NWSNS2	<server_ip>	Set SNTP server IP address/domain name. <server_ip>: SNTP server IP address/domain name. Response: OK or ERROR For example: AT+NWSNS=8.8.8.8
	?	Get SNTP server IP address.
	(none)	Response: +NWSNS:<sntp>
AT+NWSNUP	<period>	Set SNTP client update period (in seconds). <period>: SNTP client update period. Response: OK or ERROR For example: AT+NWSNUP=86400
	?	Get SNTP client update period.
	(none)	Response: +NWSNUP:<period>
AT+NWSNTP AT+NWSNTP1 AT+NWSNTP2	<sntp>	Start/Stop SNTP. <sntp>: 0 (stop), 1 (start). Response: OK or ERROR For example: AT+NWSNTP=0

DA16200 AT Command

Command	Parameters	Description
	<sntp>, <server_ip>, <period>	Start SNTP 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 For example: AT+NWSNTP=1,pool.ntp.org,86400
	?	Get SNTP status.
	(none)	Response: +NWSNTP:<sntp>
AT+NWTLSV	<ver>	Set TLS version. <ver>: TLS Version. 0 (TLSv 1.0), 1 (TLSv1.1), 2 (TLSv1.2). Response: OK or ERROR For example: AT+NWDTLS=0
	?	Get the current TLS version.
	(none)	Response: +NWTLSV=<ver>
AT+NWCERT	(none)	Check if certificates exist. There are two sets of certificates: <ul style="list-style-type: none"> Set #1: for MQTT Root CA (bit 2)/Cert (bit 1)/Key (bit 0) Set #2: for HTTPS client for OTA Root CA (bit 5)/Cert (bit 4)/Key (bit 3) For example: if DA16200 has the Root CA and Cert in Set #1, return value is 6. Response: +VER:<cert>
AT+NWCERT	(none)	Delete all TLS certificates including private key. Response: OK or ERROR

Table 5: Certificate Command

Escape Sequence	Parameters	Description
<ESC>C	<cert_id>,<content><ETX>	Store certificate or private key. Max length is 2048. <cert_id>: Certificate ID. There are two sets of certificates: <ul style="list-style-type: none"> Set #1: for MQTT 0 (Root CA)/1 (Client Certificate)/2 (Private Key) Set #2: for HTTPS client for OTA 3 (Root CA)/4 (Client Certificate)/5 (Private Key) <content>: Certificate data. <ETX>: Indication of the end of content (Ctrl+C, 0x03). Response: OK or ERROR For example: <ESC>C1,----- BEGIN CERTIFICATE ----- Mllodknvfano923nf/ ...<ETX>

DA16200 AT Command

8 Wi-Fi Function Commands

Table 6: Wi-Fi Function Command List

Command	Parameters	Description
AT+WFMODE	<mode>	Set Wi-Fi mode. <mode>: 0 (STA), 1 (Soft-AP). Response: OK or ERROR For example: AT+WFMODE=1
	?	Get the current Wi-Fi mode.
	(none)	Response: +WFMODE:<mode>
AT+WFMAC	<mac>	Write user MAC address in the NVRAM. The last digit should be an even number. Response: OK or ERROR For example: AT+WFMAC=EC:9F:0D:90:00:48
	?	Get the current MAC address of the activated WLAN interface.
	(none)	DA16200 provides three types of MAC address (OTP MAC address, user MAC address, and spoofing MAC address). The priority is in the following order: Spoofing, User, OTP. Response: +WFMAC:<mac>
AT+WFSPF	<mac>	Write spoofing MAC address in the NVRAM. The last digit should be an even number. Response: OK or ERROR For example: AT+WFSPF=EC:9F:0D:90:00:48
AT+WFOTP	<mac>	Write MAC address in the OTP memory. An old MAC address in the OTP will be invalidated if it exists. (There are four MAC address slots available in OTP. So, there is the possibility to write only four MAC addresses in total at production.) Response: OK or ERROR For example: AT+WFOTP=EC:9F:0D:90:00:48 The last hex of <mac> should be an even number. The MAC address written in the OTP is used as WLAN0 MAC address and then WLAN's MAC+1 will be used as WLAN1 MAC address.
AT+WFSTAT	(none)	Get Wi-Fi configuration. Response: +WFSTAT:<Wi-Fi interface><var>...
AT+WFPBC	(none)	Run WPS PBC method. Response: OK or ERROR
AT+WFPIN	<pin>	Run WPS PIN method.
	(none)	<pin>: PIN (eight digits). (none): Generate a random PIN. Response: +WFPIN:<pin> OK or ERROR For example: AT+WFPIN AT+WFPIN=42170518
	?	Get the current PIN. Response: +WFPIN:<pin>

DA16200 AT Command

Command	Parameters	Description
AT+WFCWPS	(none)	Cancel WPS (both PBC and PIN). Response: OK or ERROR
AT+WFCC	<code>	Set country code. <code>: Country code (defined by ISO 3166-1 alpha-2 standard). Such as KR, US, JP, CH, and so on. Response: OK or ERROR For example: AT+WFCC=KR
	?	Get the current country code.
	(none)	Response: AT+WFCC=<code>
AT+WFSCAN	(none)	Scan APs. Response: +WFSCAN:<bssid><frequency><signal strength><flag><ssid><LF>...
AT+WFJAP	<ssid>,<sec> (sec=0)	Connect to an AP. <ssid>: AP SSID.
	<ssid>,<sec>,<idx>,<key> (sec=1)	<sec>: Security protocol. 0 (OPEN), 1 (WEP), 2 (WPA), 3 (WPA2), 4 (WPA+WPA2). <idx>: Key index for WEP. 0~3 <enc>: Encryption. 0 (TKIP), 1 (AES), 2 (TKIP+AES).
	<ssid>,<sec>,<enc>,<key> (sec=2 3 4)	<key>: Passphrase. Response: OK or ERROR For example: AT+WFJAP=DA16200_OPEN,0 AT+WFJAP=DA16200_WEP,1,0,12345 AT+WFJAP=DA16200_WPA2,3,0,1234qwer AT+WFJAP='SSID,COMMA',3,1,'pass'phrase'
	?	Get the current AP information.
	(none)	Operation Results: +WFJAP:0 If Wi-Fi connection is NOT established, the Operation Result is: +WFJAP:1,<SSID>,<IP_ADDRESS> If Wi-Fi connection is established successfully: <SSID>: The SSID will be surrounded by single quotation mark <IP_ADDRESS>: Assigned IP address and format is xxx.xxx.xxx.xxx +WFJAP:0 If Wi-Fi connection is NOT established. Depending on the network condition, it may take more time to get an Operation Result due to internal connection retrials. The host should wait for both command responses OK or ERROR and Operation Result . This means that the host should wait for +WFJAP:1,<SSID>,<IP Address> and OK. No system reboot happens after running this command.

DA16200 AT Command

Command	Parameters	Description
AT+WFJAPA	<ssid>,<key>	<p>Connect to an AP.</p> <p>If <key> exists, then security protocol is WPA+WPA2 and encryption is TKIP+AES.</p> <p>if <key> is omitted, then security protocol is OPEN.</p> <p><ssid>: AP SSID.</p> <p><key>: Passphrase.</p> <p>Response: OK or ERROR</p> <p>For example:</p> <p>AT+WFJAPA=DA16200_OPEN</p> <p>AT+WFJAPA=DA16200_WPA,12345678</p> <p>AT+WFJAPA='SSID,COMMA','pass'phrase'</p>
	?	Get the current AP information. (SSID and Passphrase only.)
	(none)	<p>Operation Result:</p> <p>+WFJAP:1,'<SSID>', '<IP_ADDRESS>'</p> <p>If Wi-Fi connection is established successfully:</p> <p><SSID>: The SSID is surrounded by single quotation mark</p> <p><IP_ADDRESS>: The assigned IP address and format is xxx.xxx.xxx.xxx</p> <p>+WFJAP:0</p> <p>If Wi-Fi connection is NOT established, then depending on the network condition, it may take more time to get an Operation Result due to internal connection retries.</p> <p>The host should wait for both command response OK or ERROR and for Operation Result.</p> <p>This means that the host should wait for +WFJAP:1,'<SSID>', '<IP Address>' and OK.</p> <p>No system reboot happens after running this command.</p>
AT+WFCAP	(none)	<p>Connect to an AP with the current WLAN0 interface configuration.</p> <p>Response: OK or ERROR</p>
AT+WFQAP	(none)	<p>Disconnect from the currently associated AP.</p> <p>Response: OK or ERROR</p>
AT+WFSTA	(none)	<p>Check Wi-Fi connection.</p> <p>Response: +WFSTA:<status></p> <p><status> 1 (Connected), 0 (disconnected)</p>
AT+WFROAP	<roam>	<p>Operate STA roaming.</p> <p><roam>: 1 (run), 0 (stop).</p> <p>Response: OK or ERROR</p> <p>For example: AT+WFROAP=1</p>
	?	Get the roaming status.
	(none)	Response: +WFROAP:<roam>
AT+WFROTH	<rssi>	<p>Set STA roaming threshold.</p> <p><rssi>: Roaming threshold value (0 ~ -95 dBm).</p> <p>Response: OK or ERROR</p> <p>For example: AT+WFROTH=-80</p>
	?	Get the STA roaming threshold.
	(none)	Response: +WFROTH:<rssi>

DA16200 AT Command

Command	Parameters	Description
AT+WFDIS	<disabled>	Set the Wi-Fi profile unused. If set to 1, DA16200 will not start to connect to the configured AP when rebooting. <disabled >: 1 (Unused), 0 (Used). Response: OK or ERROR For example: AT+WFDIS=1
	?	Get the status of Wi-Fi profile.
	(none)	Response: +WFDIS:<disabled>
AT+WFENTAP	<ssid>,<wpa>,<enc>, <phase1>,<phase2>	Set AP information for WPA Enterprise (EAP). <ssid>: AP SSID. <wpa>: Authentication. 0 (WPA), 1 (WPA2), 2 (WPA+WPA2). <enc>: Encryption. 0 (TKIP), 1 (AES), 2 (TKIP+AES). <phase1>: EAP Phase 1 Type. 0 (PEAP), 1 (TTLS), 2 (FAST), 3 (TLS). <phase2>: EAP Phase 2 Type (optional). 0 (MSCHAPV2), 1 (GTC). Response: OK or ERROR For example: AT+WFENTAP=DA16200_ENT_AP,1,1,0,0 AT+WFENTAP=DA16200_ENT_AP,2,2,3
	?	Get the current AP information for WPA Enterprise (EAP).
	(none)	Response: +WFENTAP=<ssid>,<wpa>,<enc>,<phase1>,<phase2> +WFENTAP=<ssid>,<wpa>,<enc>,3 (EAP-TLS)
AT+WFENTLI	<id>,<pw>	Set Enterprise login information. <id>: ID. <pw>: Password (optional). Response: OK or ERROR For example: AT+WFENTLI=peap,12345678 AT+WFENTLI=tls
	?	Get the current Enterprise login information.
	(none)	Response: +WFENTLI=<id>,<pw> +WFENTLI=<id> (EAP-TLS)
AT+WFSAP	<ssid>,<sec>, <ch>,<code> (sec=0)	Set up Soft-AP interface. <ssid>: AP SSID.

DA16200 AT Command

Command	Parameters	Description
	<ssid>, <sec>, <enc>, <key>, <ch>, <code> (sec=2 3 4)	<p><sec>: Security protocol. 0 (OPEN), 2 (WPA), 3 (WPA2), 4 (WPA+WPA2).</p> <p><enc>: Encryption. 0 (TKIP), 1 (AES), 2 (TKIP+AES).</p> <p><key>: Passphrase.</p> <p><ch>: Operating channel (optional). Default is 1 or uses the current channel if Soft-AP is operating.</p> <p><code>: Country code (optional). If exists, <ch> is essential.</p> <p>Response: OK or ERROR</p> <p>For example: AT+WFSAP=DA16200_OPEN,0 AT+WFSAP=DA16200_WPA2,3,1,12345678,1,KR AT+WFSAP='DA16200,COMMA',3,1,'12345678',1,KR</p>
	?	Get the Soft-AP interface configuration.
	(none)	<p>Response: +WFSAP:'<ssid>',<auth>,<enc>,<key>,<ch>,<code></p> <p>Operation Result: +WFSAP:<ssid> is printed on success</p>
AT+WFOAP	(none)	Operate Soft-AP interface. Response: OK or ERROR
AT+WFTAP	(none)	Stop Soft-AP interface. Response: OK or ERROR
<p>Note that for AT+WFSAP/AT+WFOAP/AT+WFTAP:</p> <p>OK: If DA16200 operates as Soft-AP and there is no error during stopping Soft-AP</p> <p>ERROR: If DA16200 does not operate as Soft-AP or there is an error during stopping Soft-AP</p> <p>For example:</p> <pre>AT+WFSAP=DA16200_OPEN,0 // set up Soft-AP AT+RESTART // restart to reflect Soft-AP configuration ... DUT starts as Soft-AP AT+WFTAP // stop Soft-AP if you want AT+WFOAP // start Soft-AP if you want</pre> <p>AT+WFTAP/ AT+WFOAP → ERROR:-99: If you want to stop but the Soft-AP interface is not running, use this command while DUT is in Soft-AP mode.</p>		
AT+WFRAP	(none)	Restart Soft-AP interface. Response: OK or ERROR
AT+WFLCST	(none)	Get connected station information. Response: +WFLCST:<mac><LF><flags><LF><var>...
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 For example: AT+WFAPWM=1
	?	Get IEEE 802.11 Wi-Fi mode of Soft-AP interface.
	(none)	Response: +WFAPWM:<mode>

DA16200 AT Command

Command	Parameters	Description
AT+WFAPCH	<ch>	Set the operating channel number for the Soft-AP interface. <ch>: Operating channel (0 ~ 14, 0 is auto). Response: OK or ERROR For example: AT+WFAPCH=6
	?	Get the operating channel number for the Soft-AP interface. Response: +WFAPCH:<ch>
	(none)	
AT+WFAPBI	<interval>	Set AP beacon interval. <interval>: Beacon interval (ms). Response: OK or ERROR For example: AT+WFAPBI=100
	?	Get AP beacon interval. Response: +WFAPBI:<interval>
	(none)	
AT+WFAPUI	<timeout>	Set station disconnection timeout in Soft-AP mode. <timeout>: Disconnection timeout (sec). Response: OK or ERROR For example: AT+WFAPUI=300
	?	Get station disconnection timeout in Soft-AP mode. Response: +WFAPUI:<timeout>
	(none)	
AT+WFAPRT	<threshold>	Set AP RTS threshold (octets). <threshold>: RTS threshold (1 ~ 2347). Response: OK or ERROR For example: AT+WFAPRT=2347
	?	Get AP RTS threshold. Response: +WFAPRT:<threshold>
	(none)	
AT+WFAPDE	<mac>	Send de-authentication frame to the connected station. <mac>: MAC address of the connected station. Response: OK or ERROR For example: AT+WFAPDE=f8:a9:d0:49:b1:ba
AT+WFAPDI	<mac>	Send disassociation frame to the connected station. <mac>: MAC address of the connected station. Response: OK or ERROR For example: AT+WFAPDI=f8:a9:d0:49:b1:ba
AT+WFWMM	<wmm>	Set WMM on/off. <wmm>: 0 (off), 1 (on). Response: OK or ERROR For example: AT+WFWMM=1
	?	Get WMM status. Response: +WFWMM:<wmm>
	(none)	
AT+WFWMP	<wmps>	Set WMM-PS (WMM Power Save) on/off. <wmps>: 0 (off), 1 (on). Response: OK or ERROR For example: AT+WFWMP=0
	?	Get WMM-PS status. Response: +WFWMP:<wmps>

DA16200 AT Command

Table 7: Wi-Fi Function Response List

Response	Parameters	Description
+WFJAP	<result>,<ssid>,<ip>	The result of AP connection in STA mode. (The result of AT+WFJAP or AT+WFJAPA or AT+WFCAP.) <result>: 0 (failed), 1 (succeeded). <ssid>: SSID of the AP when succeeded. <ip>: IP address of the station when succeeded. For example: +WFJAP:0 The Wi-Fi connection is not established. +WFJAP:1,'ap_test',192.168.0.10 The Wi-Fi connection is established, and the assigned IP address is 192.168.0.10.
+WFDAP	<reserved>	Disconnected from the AP. <reserved>: 0 For example: +WFDAP:0
+WFCST	<mac>	A Wi-Fi station connected in Soft-AP mode. <mac>: MAC address of the connected station.
+WFDST	<mac>	A Wi-Fi station disconnected in Soft-AP mode. <mac>: MAC address of the disconnected station.

8.1 Wi-Fi Function Commands for WPA3

You can configure DA16200 as WPA3 Station or WPA3 Soft-AP with a special DA16200 SDK where WPA3 feature is enabled. By default, WPA3 is not enabled in DA16200 Generic SDK. Please ask Dialog if you need this specific SDK.

Syntax of all the Wi-Fi function commands are the same as described in Table 6 apart from the following commands where you need to specify WPA3 specific parameters.

Table 8: List of WPA3-relevant Wi-Fi Function Commands

Command	Parameters	Description
AT+WFJAP	<ssid>,<sec> (sec=0 5)	Connect to an AP. <ssid>: AP SSID.
	<ssid>,<sec>,<idx>,<key> (sec=1)	<sec>: Security protocol. 0 (OPEN), 1 (WEP), 2 (WPA), 3 (WPA2), 4 (WPA+WPA2), 5 (WPA3_OWE), 6 (WPA3_SAE), 7 (WPA2+WPA3_SAE) <idx>: Key index for WEP. 0~3
	<ssid>,<sec>,<enc>,<key> (sec=2 3 4 6 7)	<enc>: Encryption. 0 (TKIP), 1 (AES), 2 (TKIP+AES). <key>: Passphrase. Response: OK or ERROR For example: AT+WFJAP=DA16200_OPEN,0 AT+WFJAP=DA16200_WEP,1,0,12345 AT+WFJAP=DA16200_WPA2,3,0,1234qwer AT+WFJAP='SSID,COMMA',3,1,'pass'phrase'
	?	WPA3 examples: AT+WFJAP=DA16200_WPA3_OWE,5 AT+WFJAP=DA16200_WPA2_SAE,6,1,12345678 AT+WFJAP=DA16200_WPA23,7,1,12345678
		Get the current AP information.

DA16200 AT Command

Command	Parameters	Description
	(none)	<p>Operation Results: +WFJAP:0</p> <p>If Wi-Fi connection is NOT established, the Operation Result is: +WFJAP:1,'<SSID>','<IP_ADDRESS>'</p> <p>If Wi-Fi connection is established successfully: <SSID>: The SSID will be surrounded by single quotation mark <IP_ADDRESS>: Assigned IP address and format is xxx.xxx.xxx.xxx</p> <p>+WFJAP:0</p> <p>If Wi-Fi connection is NOT established. Depending on the network condition, it may take more time to get an Operation Result due to internal connection retrials. The host should wait for both command responses OK or ERROR and Operation Result. This means that the host should wait for +WFJAP:1,'<SSID>','<IP Address>' and OK. No system reboot happens after running this command.</p>
AT+WFJAPA	<is_wpa3>,<ssid>,<key>	<p>Connect to an AP. For connection to a WPA3 (OWE, SAE, or WPA2+WPA3) AP, give <is_wpa3> 1, otherwise, give 0. If <key> exists, then security protocol is WPA+WPA2 and encryption is TKIP+AES. if <key> is omitted, then security protocol is OPEN. <ssid>: AP SSID. <key>: Passphrase. Response: OK or ERROR For example: AT+WFJAPA=0,DA16200_OPEN AT+WFJAPA=0,DA16200_WPA,12345678 AT+WFJAPA=0,'SSID,COMMA','pass'phrase' Examples to connect to a WPA3 APs: AT+WFJAPA=1,DA16200_OWE AT+WFJAPA=1,DA16200_WPA3_SAE,12345678 AT+WFJAPA=1,DA16200_WPA23,12345678</p>
	?	Get the current AP information. (SSID and Passphrase only.)
	(none)	

DA16200 AT Command

Command	Parameters	Description
	(none)	<p>Operation Result: +WFJAP:1,'<SSID>','<IP_ADDRESS>' If Wi-Fi connection is established successfully: <SSID>: The SSID is surrounded by single quotation mark. <IP_ADDRESS>: The assigned IP address and format is xxx.xxx.xxx.xxx +WFJAP:0 If Wi-Fi connection is NOT established, then depending on the network condition, it may take more time to get an Operation Result due to internal connection retrials. The host should wait for both command response OK or ERROR and for Operation Result. This means that the host should wait for +WFJAP:1,'<SSID>','<IP Address>' and OK. No system reboot happens after running this command.</p>
AT+WFSAP	<ssid>,<sec>,<ch>,<code> (sec=0 5)	<p>Set up Soft-AP interface. <ssid>: AP SSID. <sec>: Security protocol. 0 (OPEN), 2 (WPA), 3 (WPA2), 4 (WPA+WPA2), WPA3_OWE (5), WPA3_SAE (6), WPA2+WPA3 (7) <enc>: Encryption. 0 (TKIP), 1 (AES), 2 (TKIP+AES) <key>: Passphrase. <ch>: Operating channel (optional). Default is 1 or uses the current channel if Soft-AP is operating <code>: Country code (optional). If exists, <ch> is essential. Response: OK or ERROR For example: AT+WFSAP=DA16200_OPEN,0 AT+WFSAP=DA16200_WPA2,3,1,12345678,1,KR AT+WFSAP='DA16200,COMMA',3,1,'12345678',1,KR WPA3 Soft-AP Setup examples: AT+WFSAP=DA16200_OWE,5,1,KR AT+WFSAP=DA16200_WPA3_SAE,6,1,12345678,1,KR AT+WFSAP=DA16200_WPA23,7,1,12345678,1,KR</p>
	<ssid>,<sec>,<enc>,<key>,<ch>,<code> (sec=2 3 4 6 7)	
	?	<p>Get the Soft-AP interface configuration. Response: +WFSAP:'<ssid>','<auth>','<enc>','<key>','<ch>','<code>'</p>
	(none)	<p>Operation Result: +WFSAP:<ssid> is printed on success</p>

DA16200 AT Command

9 Advanced Function Commands

9.1 MQTT Commands

Table 9: MQTT Command List

Command	Parameters	Description
AT+NWMQCL	<mqtt_client>	Enable/disable the MQTT client. <mqtt_client>: 0 (disable), 1 (enable) Response: OK or ERROR For example: AT+NWMQCL=1
	?	Get the MQTT client status.
	(none)	Response: +NWMQCL:<mqtt_client>
AT+NWMQMSG	<msg>,<topic>	Publish an MQTT message. <msg>: Message to be published (max length: 256 bytes). <topic>: MQTT topic (optional). Response: OK or ERROR For example: AT+NWMQMSG=Hello World AT+NWMQMSG='{ "name1": "val1", "name2": "val2" }'
AT+NWMQBR	<ip>,<port>	Set the IP address and the port number of MQTT Broker. <ip>: Broker's IP address. <port>: Broker's port number. Response: OK or ERROR For example: AT+NWMQBR=192.168.0.15,1883
	?	Get the IP address and the port number of MQTT Broker.
	(none)	Response: +NWMQBR:<ip>,<port>
AT+NWMQQOS	<qos>	Set the MQTT QoS level. <qos>: 0 (at most once), 1 (at least once), 2 (exactly once) Response: OK or ERROR For example: AT+NWMQQOS=1
	?	Get the MQTT QoS level.
	(none)	Response: +NWMQQOS:<qos>
AT+NWMQTLS	<tls>	Enable/disable MQTT TLS function. <tls>: 1 (enable), 0 (disable) Response: OK or ERROR For example: AT+NWMQTLS=1
	?	Get MQTT TLS status.
	(none)	Response: +NWMQTLS:<tls>
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). Response: OK or ERROR For example: AT+NWMQTS=2,da16k,dialog
	?	Get the MQTT subscriber topic(s).
	(none)	Response: +NWMQTS:<num>,<topic#1>,<topic#2>,...

DA16200 AT Command

Command	Parameters	Description
AT+NWMQTP	<topic>	Set the topic(s) of the MQTT publisher. <topic>: MQTT publisher topic. Response: OK or ERROR For example: AT+NWMQTP=da16k
	?	Get the MQTT publisher topic.
	(none)	Response: +NWMQTP:<topic>
AT+NWMQPING	<period>	Set MQTT ping period. <period>: Ping period (second). Response: OK or ERROR For example: AT+NWMQPING=600
	?	Get the current MQTT ping period.
	(none)	Response: +NWMQPING:<period>
AT+NWMQCID	<client_id>	Set MQTT Client ID. <client_id>: Client ID. Response: OK or ERROR For example: AT+NWMQCID=MQTT_00F3
	?	Get the current MQTT Client ID
	(none)	Response: +NWMQCID:<client_id>
AT+NWMQLI	<name>,<pw>	MQTT login information. <name>: ID. <pw>: Password. Response: OK or ERROR For example: AT+NWMQLI=da16k_user,12345678
	?	Get the MQTT login information.
	(none)	Response: +NWMQLI:<name>,<pw>
AT+NWMQAUTO	<auto>	Enable/Disable auto-start of MQTT Client at reboot. <auto>: 1 (Enable), 0 (Disable) For example: AT+NWMQAUTO=1
	?	Get the MQTT Client's auto start configuration status
	(none)	Response: +NWMQAUTO:<auto>
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 For example: AT+NWMQWILL=da16k_will,bye,0
	?	Get the MQTT Will message.
	(none)	Response: +NWMQWILL:<topic>,<msg>,<qos>
AT+NWMQDEL	(none)	Reset MQTT configurations. Response: OK or ERROR

DA16200 AT Command

Command	Parameters	Description
AT+NWMQTT	<ip>,<port>, <sub_topic>, <pub_topic>, <qos>,<tls>, <username>, <password>	<p>Run MQTT Client with options. After entering this command, system will reboot automatically. At reboot, DA16200 tries to connect to the MQTT broker after the Wi-Fi connection is successfully established.</p> <p><ip>: Broker's IP address, <port>: Broker's port number. <sub_topic>: MQTT subscriber topic. <pub_topic>: MQTT publisher topic. <qos>: MQTT QoS level. <tls>: Enable/disable MQTT TLS. 1 (enable), 0 (disable). <username>: Login ID (optional). <password>: Login password (optional).</p> <p>Response: OK or ERROR</p> <p>For example: AT+NWMQTT=192.168.0.15,1883,da16k_sub,da16k_pub,2,1,mqtt_id,12345678</p> <p>For the Operation Result: see +NWMQCL response.</p>

Table 10: MQTT Response List

Response	Parameters	Description
+NWMQCL	<result>	<p>The result of MQTT client connection. <result>: 0 (disconnected), 1 (connected)</p> <p>For example: +NWMQCL:1</p> <p>If MQTT connection to the MQTT broker is successfully established.</p> <p>+NWMQCL:0</p> <p>If MQTT connection is NOT successfully established.</p> <p>Until you get an Operation Result, it may take more time if the DA16200 connection retrial happens depending on your test network condition.</p> <p>Expected result after AT+NWMQTT is run or if any MQTT configuration command is run and then the system is restarted.</p> <ol style="list-style-type: none"> DA16200 restarts if the AT command format is OK. <ol style="list-style-type: none"> +INIT:DONE,0 message is sent as DA16200 boots up. If usage of the AT command is not valid, DA16200 sends ERROR message without restarting. DA16200 tries to connect to the AP after the reboot. <ol style="list-style-type: none"> +WFJAP:0 or +WFJAP:1,'<SSID>','<IP Address>' as result of the Wi-Fi connection. If the Wi-Fi connection information such as SSID or key is NOT stored correctly in the DA16200 NVRAM, +WFJAP:x response is NOT sent and the MQTT connection is NOT attempted as well. Because the MQTT connection needs successful Wi-Fi connection first. DA16200 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.

DA16200 AT Command

Response	Parameters	Description
+NWMQMSG	<msg>,<topic>,<length>	Received MQTT message. <msg>: Message data. <topic>: Received topic. <length>: Message length.

9.1.1 MQTT Client Connection Example

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

```
AT+NWMQBR=172.16.0.1,1884
AT+NWMQTS=1,da16k_sub
AT+NWMQTP=da16k_pub
AT+NWMQCL=1
```

If the connection is successful, the following text is shown:

```
+NWMQCL:1
```

If DA16K receives a PUBLISH from a broker, the following text 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
```

9.1.2 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
```

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. (<ESC>C in Section 7)

After all settings are made, start the client:

```
AT+NWMQCL=1
```


9.1.3 MQTT Example with DPM

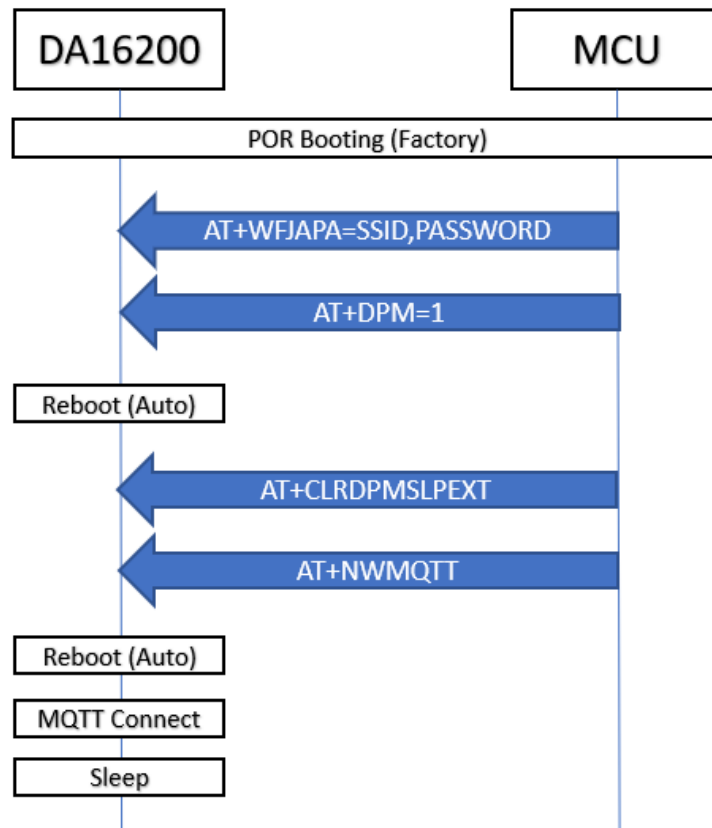


Figure 5: Example Sequence to Initiate MQTT Protocol with DPM

Figure 5 is an example sequence to initiate the MQTT protocol with DPM in the DA16200.

In the normal BOOT state, connect to an AP (AT+WFJAPA) and change the DA16200 run mode to DPM mode (AT+DPM=1). To configure the MQTT connection information, enter command AT+CLRDPM_SLP_EXT and type the following as an example:

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

DA16200 AT Command

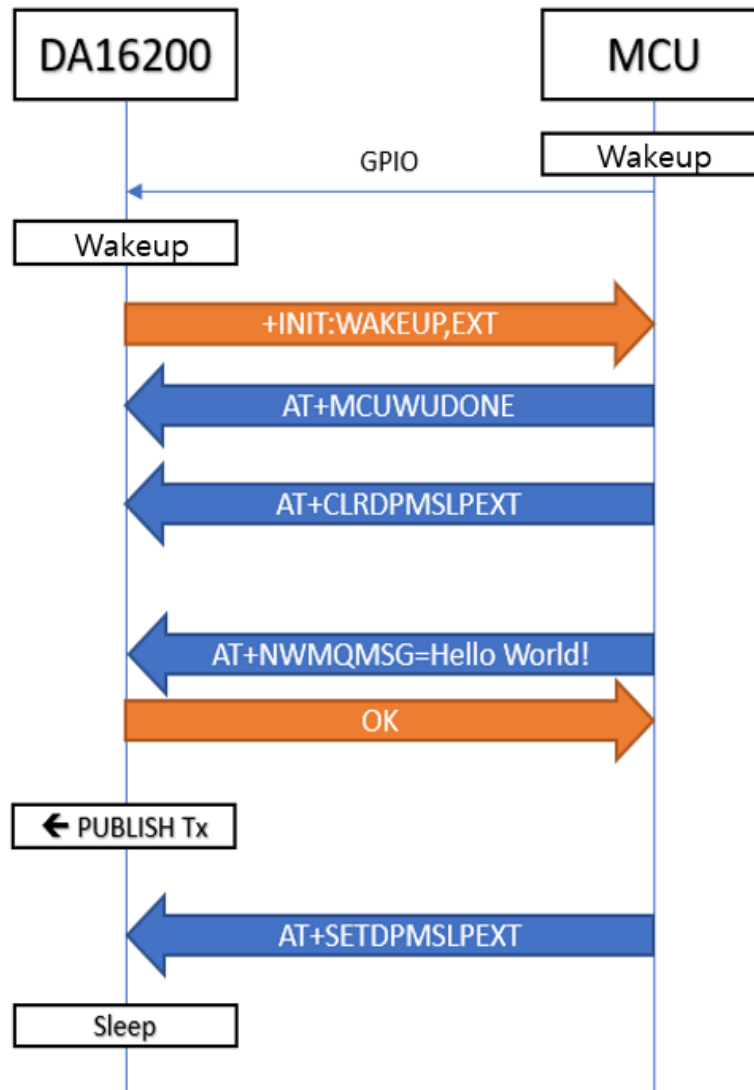


Figure 6: Procedure to Send MQTT Messages

Figure 6 shows the procedure to send an MQTT message in Sleep mode.

When MCU wakes up the DA16200, 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 enters DPM Sleep mode, MCU should send command AT+CLRDPM_SLP_EXT before an MQTT PUBLISH is sent. To make the DA16200 enter DPM Sleep mode again, send a PUBLISH with command AT+NWMQMSG, and then enter command AT+SETDPM_SLP_EXT.

DA16200 AT Command

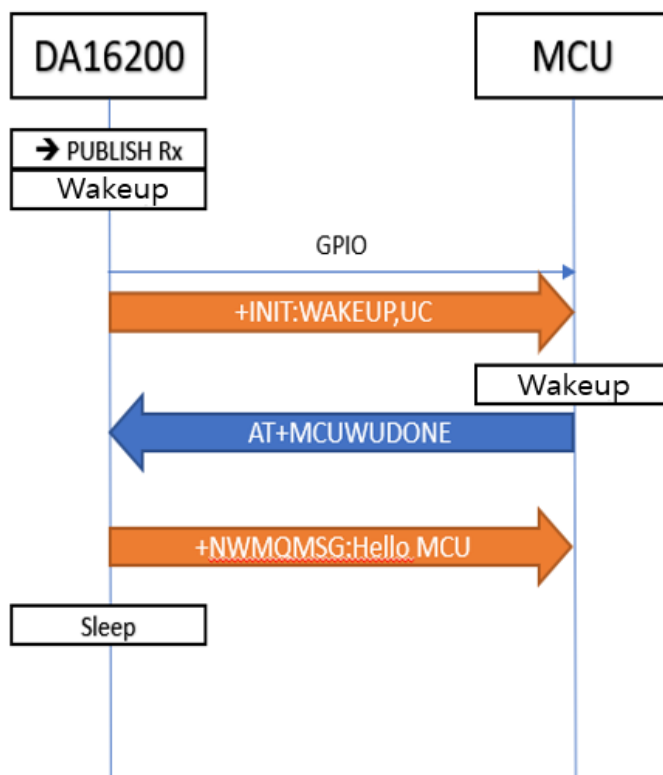


Figure 7: Procedure to Process MQTT Messages

Figure 7 shows the procedure to process an MQTT message received while in Sleep mode.

When the DA16200 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 sends the received PUBLISH to the MCU and enters DPM Sleep mode again.

9.1.4 MQTT Example: Changing Subscription Topic when Running

Let's 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.

- A. Trigger RTC WAKE_UP Event (by MCU)
- B. Wait for "+INIT:WAKEUP,EXT" Response
- C. Run "AT+CLRDPM_SLP_EXT" 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+SETDPM_SLP_EXT"
 - 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

DA16200 AT Command

- 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

9.1.5 MQTT Example: Reading Subscription Topic when Running

Let's 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.

- A. Trigger RTC_WAKE_UP Event
- B. Wait for "+INIT:WAKEUP,EXT" Response
- C. Run "AT+CLRDPM_SLP_EXT" command
- D. Wait for "OK" response
- H. Run "AT+NWMQTS=?"
- I. Wait for "+NWMQTS:<MQTT Subscription Topic>" and "OK" response

NOTE

You need to consider that you can get the ERROR response in case that format of the command may have some error.

- H. Run "AT+SETDPM_SLP_EXT"
- I. Wait for "OK" response

Let's 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.

- A. Trigger RTC_WAKE_UP Event
- B. Wait for "+INIT:WAKEUP,EXT" Response
- C. Run "AT+CLRDPM_SLP_EXT" command
- D. Wait for "OK" response
- H. Run "AT+NWMQTS=?"
- I. Wait for "+NWMQTS:<MQTT Subscription Topic>" and "OK" response
- H. Run "AT+SETDPM_SLP_EXT"

9.2 HTTP-Client Commands

Table 11: HTTP-Client Command List

Command	Parameters	Description
AT+NWHTC	<url>,<method>(<msg>)	Start HTTP client with options. <url>: HTTP server address. <method>: GET, POST or PUT. <msg>: Request message for POST and PUT methods.

Table 12: HTTP-Client Response List

Response	Parameters	Description
+ NWHTC	<status>	Returns status along with the received payload according to the requested method. <status>: 0x00 is success. See Appendix B . For example: +NWHTCSTATUS:0x00

DA16200 AT Command

9.2.1 HTTP-Client Connection Example

GET method request:

AT+NWHTC=https://httpbin.org/get,get

POST method request:

AT+NWHTC=https://httpbin.org/post,post,HTTP-Client POST method sample test!

PUT method request:

AT+NWHTC=https://httpbin.org/put,put,HTTP-Client PUT method sample test!

9.3 HTTP-Server Commands

Table 13: HTTP-Server Command List

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

9.3.1 HTTP/HTTPS-Server Start Example

HTTP start:

AT+NWHTS=1

HTTPS start:

AT+NWHSS=1

9.4 OTA Commands

Table 14: 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. It is available if <i>fw_type</i> is <i>mcu_fw</i> . Maximum input size is eight bytes. MCU_FW will be stored as default if there is no <i>fw_name</i> information. (Only for MCU FW) Response: +NWOTADWSTART:0x00 For example: AT+NWOTADWSTART=rtos,https://server/DA16200_RTOS-GEN01-01-1111-000000.img
AT+NWOTARENEW	(none)	Reboot with updated FW. For example: AT+NWOTARENEW
AT+NWOTADWPROG	<fw_type>	FW download progress. <fw_type>: slib/rtos. Response: +NWOTADWPROG:100 For example: AT+NWOTADWPROG=slib (in SDK V3.x.x.x, only rtos is available)

DA16200 AT Command

Command	Parameters	Description
AT+NWOTADWSTOP	(none)	Stop while downloading FW.
AT+NWOTAFWNAME	(none)	Read a name in the header of the MCU firmware. (Only for MCU FW)
AT+NWOTAFWSIZE	(none)	Read a size in the header of the MCU firmware. (Only for MCU FW)
AT+NWOTAFWCRC	(none)	Read a CRC in the header of the MCU firmware. (Only for MCU FW)
AT+NWOTAREADFW	<read_addr>, <read_size>	Read the MCU firmware as much as the <i>read_size</i> from the <i>read_addr</i> and transmit it. (Only for MCU FW) <read_addr>: Hexadecimal without "0x" prefix. <read_size>: Decimal.
AT+NWOTATRANSFW	(none)	Transmit an MCU firmware to MCU through UART1. Transmission will be failed if no header (16 bytes) information exist. (Only for MCU FW)
AT+NWOTAERASEFW	(none)	Erase the MCU firmware stored in serial flash of DA16200. (Only for MCU FW)
AT+NWOTASETADDR	<sflash_addr>	Data to be downloaded can be designated as an address within the range of <i>User Area</i> and <i>TLS Certificate Key</i> in the SFLASH area. The default value is 0x003A_D000.
AT+NWOTAGETADDR	<fw_type>	Returns the value set with NWOTASETADDR.
AT+NWOTAREADFLASH	<sflash_addr>,<size>	Read as much as size from <i>sflash_addr</i> .
AT+NWOTAERASEFLASH	<sflash_addr>,<size>	Delete as much as size from <i>sflash_addr</i> .
AT+NWOTACOPYFLASH	<dest_sflash_addr>,<src_sflash_addr>,<size>	Copy as much as size from <i>src_sflash_addr</i> to <i>dest_sflash_addr</i> .

Table 15: OTA Response List

Response	Parameters	Description
+NWOTADWSTART	<status>	Returns the status of FW download. <status>: 0x00 is success. See Table 16 for other status values. For example: +NWOTADWSTART:0x00
+NWOTARENEW	<status>	Returns the status for FW RENEW. <status>: 0x00 is success. See OTA Response Code List for others. For example: +NWOTARENEW:0x00
+NWOTADWPROG	<progress>	Returns the percentage value (%) of the FW download progress. <progress>: Print download progress in percent. For example: +NWOTADWPROG:100

DA16200 AT Command

Response	Parameters	Description
+NWOTADWSTOP	<status>	Returns the status of FW download stop. <status>: 0x00 is success. See Table 16 for other status values. For example: +NWOTADWSTOP:0x00
+NWOTATRANSFW	COMPLETE or FAIL	Returns 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 Table 16 for other status values.
+NWOTAGETADDR	<sflash_addr>	Returns the value of sflash_addr.
(AT+NWOTAREADFLASH)	(Binary)	Returns binary data as much as the entered SFLASH address and size.
+NWOTAERASEFLASH	COMPLETE or FAIL	Success: COMPLETE Failure: FAIL
+NWOTACOPYFLASH	COMPLETE or FAIL	Success: COMPLETE Failure: FAIL

Table 16: OTA Response Code List

Return Value	Description
0x00	Return success.
0x01	Return failed.
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.

DA16200 AT Command

Return Value	Description
0x0A	Failed to connect to server.
0x0B	All new FWs have not been downloaded. Be sure to download both slib and rtos in case of SDK V2.x.x.x.
0x0C	Failed to allocate memory.

9.4.1 OTA Download Example

SLIB download: (SDK V2.x.x.x only)

AT+NWOTADWSTART=slib,https://server/DA16200_SLIB-GEN01-01-1111-000000.img

RTOS download:

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

MCU FW download:

AT+NWOTADWSTART=other_fw,https://server/mcu_firmware.img

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

Cert Key download:

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

9.4.2 OTA Download Progress Example

SLIB download progress: (SDK V2.x.x.x only)

AT+NWOTADWPROG=slib

RTOS download progress:

AT+NWOTADWPROG=rtos

MCU FW download progress:

AT+NWOTADWPROG=mcu_fw

Cert Key download progress:

AT+NWOTADWPROG=cert_key

9.4.3 OTA Renew Example

Renew Firmware (reboot with updated FW):

AT+NWOTARENEW

9.4.4 MCU FW Transport Example

MCU FW transmission:

AT+NWOTATRANSEW

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:

DA16200 AT Command

AT+NWOTAREADFW=1f2000,128

Delete MCU FW stored in the DA16200 SFLASH:

AT+NWOTAERASEFW

9.4.5 SFLASH User Area address setting Example

SET ADDR:

AT+NWOTASETADDR=0x1f2000

GET ADDR:

AT+NWOTAGETADDR=mcu_fw

AT+NWOTAGETADDR=cert_key

9.4.6 SFLASH READ/COPY/ERASE Example

SFLASH Read:

AT+NWOTAREADFLASH=0x1f2000,128

SFLASH Copy:

AT+NWOTACOPYFLASH=0x1f2000,0x1f3000,128

SFLASH Erase:

AT+NWOTAERASEFLASH=0x1f2000,128

DA16200 AT Command

10 Transfer Function Commands

10.1 Socket Commands

Table 17: Socket Command List

Command	Parameters	Description
AT+TRTS	<local_port>	Open a TCP server socket. <local_port>: Local port number of the socket. Response: OK or ERROR For example: AT+TRTS=88
AT+TRTC	<server_ip>, <server_port>, <local_port>	Open a TCP client socket and connect to a TCP server. <server_ip>: IP address of TCP server to be accessed. <server_port>: Port number of TCP server. <local_port>: Local port number of the socket (optional, 0: auto). Response: OK or ERROR For example: AT+TRTC=192.168.20.1,88
AT+TRUSE	<local_port>	Open a UDP socket. <local_port>: Local port number of the socket. Response: OK or ERROR For example: AT+TRUSE=89
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 For example: AT+TRUR=192.168.20.1,90
AT+TRPRT	<cid>	Get session information by CID. <cid>: 0 (TCP server), 1 (TCP client), 2 (UDP) Response: <cid>,[TCP UDP],<remote_ip>,<remote_port>, <local_port> For example: AT+TRPRT=1
AT+TRPALL	(none)	Get all session information. Response: <cid>,[TCP UDP],<remote_ip>,<remote_port>, <local_port><LF>...
AT+TRTRM	<cid>	Close (terminate) a session by CID. <cid>: 0 (TCP server), 1 (TCP client), 2 (UDP) Response: OK or ERROR For example: AT+TRTRM=1
AT+TRTALL	(none)	Close (terminate) all sessions. Response: OK or ERROR
AT+TRSAVE	(none)	Save status of all sessions to NVRAM. Response: OK or ERROR

DA16200 AT Command

Table 18: Socket Connection Response List

Response	Parameters	Description
+TRCTS	<cid>, <remote_ip>, <remote_port>	A remote TCP client is connected to the TCP server that was opened by AT+TRTS. <cid>: 0 (TCP Server). <remote_ip>: TCP client IP address. <remote_port>: TCP client port number. For example: +TRCTS:0,192.168.0.5,3713
+TRXTS	<cid>, <remote_ip>, <remote_port>	A remote TCP client is disconnected from the TCP server that was opened by AT+TRTS. <cid>: 0 (TCP Server). <remote_ip>: TCP client IP address. <remote_port>: TCP client port number. For example: +TRXTS:0,192.168.0.5,3713
+TRXTC	<cid>, <remote_ip>, <remote_port>	The TCP client socket that was opened by AT+TRTC is disconnected. <cid>: 1 (TCP Client). <remote_ip>: TCP server IP address. <remote_port>: TCP server port number. For example: +TRXTC:1,192.168.0.5,3713

10.2 Data Transfer Commands

Table 19: Data Transmission Command

Escape Sequence	Parameters	Description
<ESC>S	<cid><length>, <remote_ip >, <remote_port >,<data>	Transmit data through a socket with the CID specified. <ESC>: The escape character (0x1B) <cid>: 0 (TCP server), 1 (TCP client), 2 (UDP) <length>: Data length (If this is 0 then read the command until '\r' or '\n' is met. Max length is 2048 bytes, but data will be fragmented by MTU). <remote_ip>: Remote IP address. <remote_port>: Remote port number. <ul style="list-style-type: none"> For TCP Server, <remote_ip> and <remote_port> of a TCP Client should be given. Maximum four TCP Clients can be connected to the TCP Server For TCP Client, 0, 0 is given (as the destination is the server) For UDP: if you give 0,0, the data is sent to the destination that AT+TRUR has specified. if non-0 <remote_ip> and <remote_port> are given, UDP temporarily sends to the destination <remote_ip> and <remote_port> specifies Response: OK or ERROR For example: <ESC>S010,192.168.0.1,3713,abcde12345 <ESC>S110,0,0,abcde12345 <ESC>S210,0,0,abcde12345 <ESC>S210,10.0.0.1,2543,abcde12345

DA16200 AT Command

Table 20: Data Reception Responses

Response	Parameters	Description
+TRDTS	<cid>, <src_ip>,<src_port>, <length>,<data>	Receive data through TCP server socket. <cid>: 0 (TCP Server). <src_ip>: Source IP address. <src_port>: Source port number. <length>: Data length. <data>: Received data. For example: +TRDTS:0,172.16.7.1,54687,10,DIA_ACT_TS
+TRDTC	<cid>, <src_ip>,<src_port>, <length>,<data>	Receive data through TCP client socket. <cid>: 1 (TCP Client). <src_ip>: Source IP address. <src_port>: Source port number. <length>: Data length. <data>: Received data. For example: +TRDTC:1,192.168.20.1,88,10, DIA_ACT_TC
+TRDUS	<cid>, <src_ip>,<src_port>, <length>,<data>	Receive data through UDP server socket. <cid>: 2 (UDP Session). <src_ip>: Source IP address. <src_port>: Source port number. <length>: Data length. <data>: Received data. For example: +TRDUS:0,172.16.7.1.54687,10.DIA_ACT_US

10.3 Data Transfer with DPM

10.3.1 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
```

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

```
+INIT: WAKEUP, UC
+TRCTS: 0, 192.168.0.1, 42000
```

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

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

To send a TCP message, enter AT+MCUWUDONE immediately after “external wakeup” is triggered. To prevent that DA16200 enters DPM Sleep mode, MCU should send AT+CLRDPM_SLP_EXT before

DA16200 AT Command

a message is sent. The DA16200 sends data to a TCP client with command “<ESC>S”. Finally, to enter DPM sleep mode, enter “AT+SETDPMSTEXT”.

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

```
+INIT:WAKEUP,UC  
+TRXTS:0,192.168.0.1,42000
```

10.3.2 TCP Client

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

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

Change the DA16200 state to DPM mode (AT+DPM=1). When the DA16200 starts the session on DPM mode successfully, the following text 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 10.3.1. When the DA16200 receives a message from the server, the following text is shown:

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

10.3.3 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
```

Change the DA16200 state to DPM mode. When the DA16200 starts the session in DPM mode successfully, the following text 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 10.3.1. When the DA16200 receives a message from the server, the following text is shown:

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

DA16200 AT Command

11 RF Test Function Commands

Table 21: RF Test Command List

Command	Parameters	Description
AT+ TMRFNINIT	<flag>	Set boot mode. <flag>: 0 (normal boot), 1 (RF test mode boot) Response: OK or ERROR For example: AT+TMRFNINIT=1
AT+ TMLMACINIT	(none)	Initialize LMAC (for test mode). Response: OK or ERROR
AT+RFTESTSTART	(none)	Start RF test mode.
AT+RFTESTSTOP	(none)	Stop RF test mode.
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>: 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. <bssid>: BSSID. <htEnable>: N/A <GI>: [short long]. Guard interval. 11n mode only. <greenField>: [on off]. Set greenfield mode on/off. <preambleType>: [short long]. Preamble type @DSSS mode. <qosEnable>: [on off]. MAC header QoS control. <ackPolicy>: [NO NORM BA CBA] <scrambler>: N/A

DA16200 AT Command

Command	Parameters	Description
		<aifsnVal>: [0 ~ 15]. Indicate the AIFS in units of slots after SIFS that HW should wait for before starting backoff, for access category. <ant>: [0]. Fixed.
AT+RFTXSTOP	(none)	Stop RF TX test.
AT+RFCWTEST	<Ch>, <BW>, <txPower>, <ant>, <CWCycle>	Start CW test. <Ch>: Carrier frequency (2412 ~ 2484 MHz). <BW>: Carrier bandwidth. 20 MHz fixed. <txPower>: TX power (0 ~ 15), 0.8 dB step. <ant>: [0]. Fixed. <CWCycle>: [1 2 4 5 8 10] MHz.
AT+RFCWSTOP	(none)	Stop CW test.
AT+RFPER	(none)	Display PER state. Indicate number of Valid packets, FCS Errors packets, PHY Errors packets, Overflow Errors.
AT+RFPERRESET	(none)	Reset PER count.
AT+RFCONTSTART	<txRate>, <txPower>, <Ch>	Start RF continuous TX test. <txRate>: Data rate. Refer to AT+RFTX command. <txPower>: TX power (0 ~ 15), 0.8 dB step. <Ch>: Carrier frequency (2412 ~ 2484 MHz).
AT+RFCONTSTOP	(none)	Stop RF continuous TX test.
AT+RFCHANNEL	<Ch>	Change RF channel. <Ch>: Carrier frequency (2412 ~ 2484 MHz).

DA16200 AT Command

12 System and Peripheral Function Commands

12.1 SPI Commands

Table 22: SPI Command List

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

12.2 OTP Commands

Table 23: OTP Command List

Command	Parameters	Description
AT+UOTPRDASC	<addr>,<cnt>	<p>Read OTP data.</p> <p>DA16200's physical OTP offset range is h0~h7FF; at each offset, four bytes are stored or read.</p> <p>For accessing OTP using this command, four-byte aligned address should be given. For example: h0, h4, h8 ...</p> <p><addr>: OTP address to read four-byte aligned.</p> <p><cnt>: Bytes to read.</p> <p>Response: OK or Error</p> <p>A string of four-bit HEXA value represented by the ASCII code.</p> <p>For example:</p> <p>Reading four bytes at offset h100 → h100 * 4 = h400</p> <p>AT+UOTPRDASC=400,4</p> <p>aabbccdd</p> <p>OK</p>

DA16200 AT Command

Command	Parameters	Description
AT+UOTPWRASC	<addr>,<cnt>,<value>	<p>Write OTP data.</p> <p>DA16200's physical OTP offset range: h0~h7FF; at each offset, four bytes are stored or read.</p> <p>For accessing OTP using AT-COMMAND, four-byte aligned address should be given: For example: h0, h4, h8 ...</p> <p><addr>: OTP address to write four-byte aligned.</p> <p><cnt>: Bytes to write.</p> <p><value>: A string of four-bit HEXA value represented by the ASCII code</p> <p>Response: OK or Error</p> <p>For example:</p> <p>Writing h12345678 to OTP Address 0x400: AT+UOTPWRASC=400,4,12345678 OK AT+UOTPRDASC=400,4 12345678 OK</p> <p>Important:</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>

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

Table 24: 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 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 25: Size of Memory by XTAL Offset

Slot	OTP Address	Description	Size (Bytes)
XTAL Offset #0	0x10A	XTAL Offset #0 value	2
XTAL Offset #1	0x10B	XTAL Offset #1 value	2

DA16200 AT Command

12.3 XTAL Commands

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

Table 26: XTAL Command List

Command	Parameters	Description
AT+XTALWR	<value>	Write XTAL Offset to DA16200 system register. This value will be erased after system reset and the register is set with the OTP XTAL offset value. <value>: seven-bits to write [h'1 ~ h'7f]. Response: OK or Error For example: AT+XTALWR=7f OK AT+XTALWR=80 ERROR
AT+XTALRD	(none)	Read XTAL Offset from DA16200 System. Response: <A string of seven-bit HEXA value represented by the ASCII Code> OK or Error For example: AT+XTALRD 0x7f OK

12.4 Flash Dump Commands

Table 27: Flash Dump Command List

Command	Parameters	Description
AT+FLASHDUMP	<address>, <length>	Dump serial flash data. <address>: Start address [h'0 ~ h'3ffff]. <length>: Data length [d]. Response: <dump data> <CR><LF>0x[crc[7] crc[6] crc[5] crc[4] crc[3] crc[2] crc[1] crc[0]] <CR><LF><CR><LF>OK<CR><LF> or Error For example: the following example reads 32 KB from 0x0. AT+FLASHDUMP=0,32768

DA16200 AT Command

12.5 GPIO Commands

Table 28: GPIO Command List

Command	Parameters	Description
AT+GPIOSTART	<port>, <pin >, <direction>	Configures the GPIO pin mux and the direction of a GPIO. <port>: GPIO port number. * 0: GPIOA * 1: GPIOC <pin>: GPIO pin number. This is a hexadecimal value and indicates a GPIO bitmap. <direction>: GPIO pin direction. * 0: Sets the pin as an input * 1: Sets the pin as an output Response: OK or Error For example: To configure GPIOA[3:0] output: AT+GPIOSTART=0,f,1 To configure GPIOC[8:6] input: AT+GPIOSTART=2,1c0,0
AT+GPIORD	<port>, <pin>	Reads the GPIO input level. <port>: GPIO port number. * 0: GPIOA * 1: GPIOC <pin>: GPIO pin number. This is a hexadecimal value and indicates a GPIO bitmap. Response: <Read value>: [h'0 ~ h'1fff] OK or Error The read value indicates GPIO bitmap. If the value is 0x1c0, it means GPIO #6, #7, #8 high. For example: To configure GPIOC[8:6] input: AT+GPIOSTART=2,1c0,0 To read GPIOC[8:6] input level: AT+GPIORD=2,1c0
AT+GPIOWR	<port>, <pin>, <level>	Configures the output level of GPIO pins. <port>: GPIO port number. 0: GPIOA 1: GPIOC <pin>: GPIO pin number. This is a hexadecimal value and indicates a GPIO bitmap. <level>: GPIO output level. 0: Low 1: High Response: OK or Error. For example: To configure GPIOC[8:6] output: AT+GPIOSTART=2,1c0,1 To write GPIOC[8:6] high level: AT+GPIOWR=2,1c0,1

DA16200 AT Command

Command	Parameters	Description
AT+SAVE_PININFO	(none)	Save pin mux information. Response: OK or Error.
AT+RESTORE_PININFO	(none)	Restore pin mux information. Response: OK or Error.

12.6 Sleep Commands

Table 29: Sleep Command List

Command	Parameters	Description
AT+SLEEPMS	<period>	Make DA16200 go to Sleep mode 3 and wake up after <period> milliseconds. <period>: Wakeup time in milliseconds. Response: OK or Error For example: AT+SLEEPMS=5000

13 Examples

13.1 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 and a PC with IO Ninja. You can run DA16200 AT commands on a serial terminal application on your PC. The terminal must be connected to the UART1 interface of the DA16200.

DA16200 AT Command

13.1.1 TCP Server Socket Test

1. DA16200 AT command:
 - a. AT+TRIS=1234 ← Open a TCP server socket of which the port is 1234.
2. PC:
 - a. Select **TCP Connection Socket** (1, [Figure 8](#)).
 - b. Enter the IP address and the port number of DA16200 (2, [Figure 8](#)).
 - c. Click  to connect the socket (3, [Figure 8](#)).
3. DA16200 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 8](#)).
5. DA16200 AT command:
 - a. +TRDTS:0,192.168.0.5,3713,10,DIA_AT_TCP ← Received ten bytes of data “DIA_AT_TCP”.

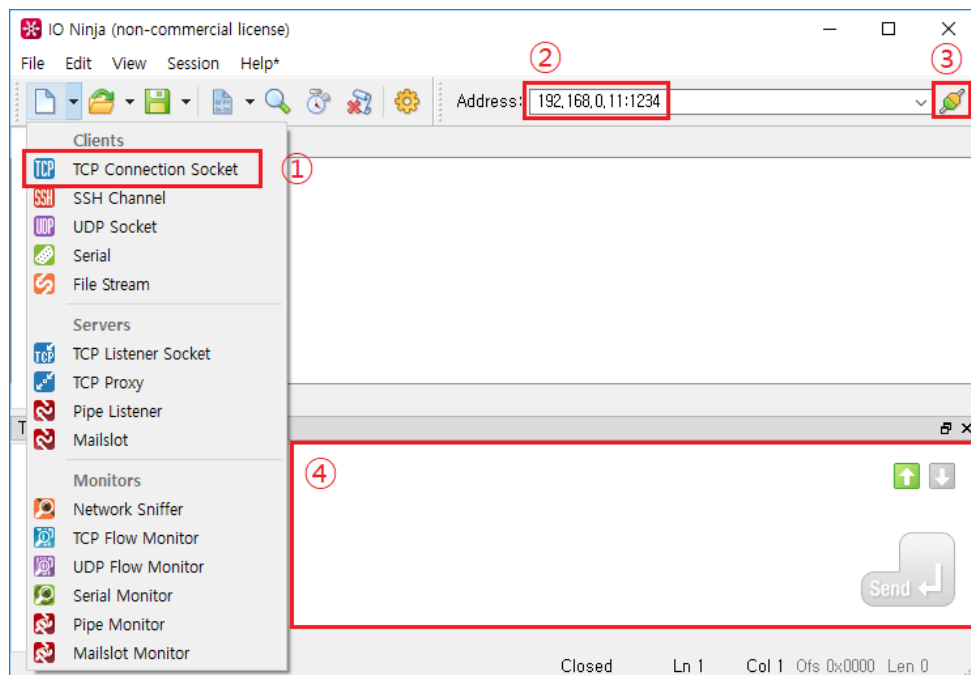



Figure 8: IO Ninja – TCP Client Socket Setting

DA16200 AT Command

13.1.2 TCP Client Socket Test

1. PC:
 - a. Select **TCP Listener Socket** (1, Figure 9).
 - b. Enter the port number to be used (2, Figure 9).
 - c. Click  to start to “Listen” (3, Figure 9).
2. DA16200 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 eight bytes of data “12345678”.
3. PC:
 - a. Received data.
 - b. Send data (4, Figure 9).
4. DA16200 AT command:
 - a. +TRDTC:1,10,DIA_AT_TCP ← Received ten bytes of data “DIA_AT_TCP”.

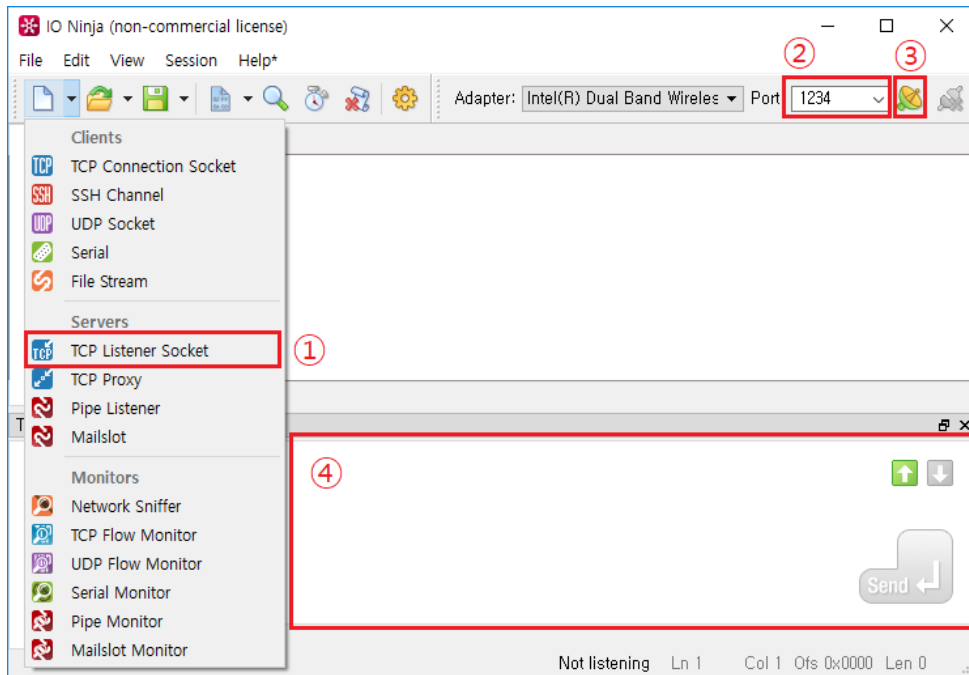




Figure 9: IO Ninja – TCP Server Socket Setting

DA16200 AT Command

13.1.3 UDP Socket Test

1. PC:
 - a. Select **UDP Socket** (1, [Figure 10](#)).
 - b. Enter the port number to be used and click  to open the socket (2, [Figure 10](#)).
 - c. Enter the IP address and port of the counterpart's UDP socket, click  and get ready for data transmission (3, [Figure 10](#)).
 - d. Enter data and click **Send** to transmit (4, [Figure 10](#)).
2. DA16200 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 ten bytes of data “1234567890”.
 - d. +TRDTC:0,10,DIA_AT_UDP ← Received ten bytes of data “DIA_AT_UDP”.

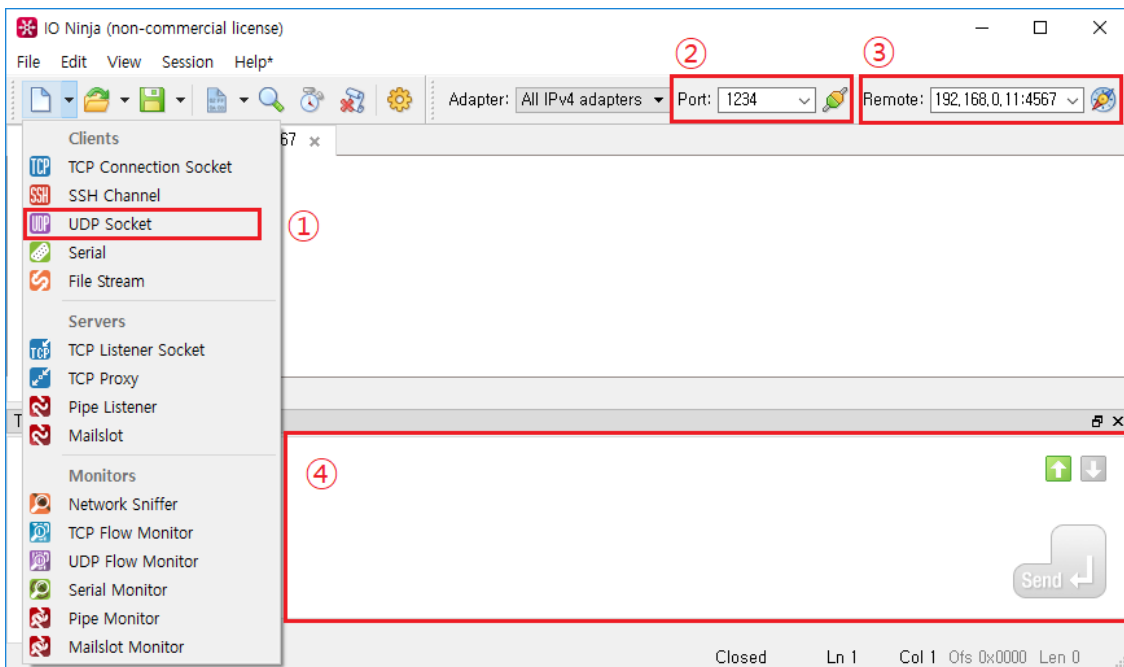


Figure 10: IO Ninja – UDP Socket Setting

Appendix A License Information

Mosquitto1.4.14 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 AT Command

Appendix B HTTP API Return Values

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

DA16200 AT Command

Define	Value	Define	Value
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		

Appendix C User UART Configuration

C.1 How to Run AT-CMD on UART2

AT-CMD is running on UART1 by default.

AT-CMD, depending on user hardware's configuration, can be running on UART2.

Build SDK with the following change (in **bold font**) in config_generic_sdk.h, then UART2 is used for AT-CMD.

```
...
// AT-CMD features
#define __SUPPORT_ATCMD__ // Support AT-CMD
#ifdef __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's a feature called "User UART Configuration" that is enabled by `__USER_UART_CONFIG__` (available in SDK V2.3.4.0 and later version).

If SDK is built with `__USER_UART_CONFIG__`, a user can configure AT-CMD's UART Setting programmatically. (ATB will not be available with `__USER_UART_CONFIG__` enabled).

For example, if a user wants to run AT-CMD on UART2 w/ static baud rate as 230400, SDK should be configured as below (see necessary changes highlighted in **bold font**) before build.

```
// config_generic_sdk.h
...
// AT-CMD features
#define __SUPPORT_ATCMD__ // Support AT-CMD
#ifdef __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_atcmd.h
...
#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 */
}
```

DA16200 AT Command

```
    UART_STOPBITS_1,          /* stopbit */
    UART_FLOWCTL_OFF         /* flow control */
};
#endif // __USER_UART_CONFIG__
...
```

With changes above, when DA16200 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: AT-CMD is initialized in 115200
 - MCU: ATB=230400
 - MCU: Now it should change its UART baud rate to 230400 to communicate with DA16200

// __USER_UART_CONFIG__ enabled

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

DA16200 AT Command

Revision History

Revision	Date	Description
2.3	01-Apr-2021	Added OTA Update command Added support for SDK V3.x.x.x
2.2	15-Mar-2021	Added Appendix B HTTP API Return Values Added Appendix C Added AT+NWMQAUTO and ATB
2.1	13-Jan-2021	New section added: Section 8.1 Wi-Fi Function Commands for WPA3 minor update (typo, or minor change done)
2.0	08-Dec-2020	Added additional description on the following commands AT+WFSAP, AT+WFOAP, AT+WFTAP, ATF, AT+WFJAPA, AT+NWMQTT, +NWMQCL, AT+DPM Added new sections: 7.1.4 MQTT Example: Changing Subscription Topic while running 7.1.5 MQTT Example: Reading Subscription Topic while running
1.9	11-Nov-2020	AT+NWCCRT, <ESC>C updated AT+NWSNS updated AT+NWHTS updated AT+NWHTSS updated
1.8	18-Aug-2020	Added SNTP command to Section 5 Network Function Commands Added Http-client command to Section 7.2 HTTP Commands Added MCU FW update command using OTA to Section 7.3 OTA Commands paragraph
1.7	30-June-2020	Added 2.4 Configuration for MCU Wake-up Correct typos and wordings
1.6	29-Apr-2020	Added AT+WFDIS and AT+SETDPMSLP2EXT Updated MQTT commands to operate with one-port Updated to process the comma in the parameters
1.5	03-Apr-2020	Added AT+BIDX for changing boot index in Chapter 4. Added example code of MQTT command in Section 7.1.1 ~ 7.1.3 Updated RF Test function commands in Chapter 9. Updated GPIO commands in Section 10.5
1.4	21-Oct-2019	Updated 2.2 Serial Port configuration steps. Removed draft status
1.3	15-Oct-2019	Error correction Added explanation to serial program at page 6
1.2	07-Oct-2019	Editorial review and add code: UM-B-111
1.1	25-Jul-2019	Added OTP Memory Address for writing MAC address in page 27
1.0	03-Jul-2019	Preliminary DRAFT Release

DA16200 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.

Disclaimer

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

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

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

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

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

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

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

© 2021 Dialog Semiconductor. All rights reserved.

RoHS Compliance

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

Contacting Dialog Semiconductor

United Kingdom (Headquarters)

Dialog Semiconductor (UK) LTD
Phone: +44 1793 757700

Germany

Dialog Semiconductor GmbH
Phone: +49 7021 805-0

The Netherlands

Dialog Semiconductor B.V.
Phone: +31 73 640 8822

Email:

enquiry@diasemi.com

North America

Dialog Semiconductor Inc.
Phone: +1 408 845 8500

Japan

Dialog Semiconductor K. K.
Phone: +81 3 5769 5100

Taiwan

Dialog Semiconductor Taiwan
Phone: +886 281 786 222

Web site:

www.dialog-semiconductor.com

Hong Kong

Dialog Semiconductor Hong Kong
Phone: +852 2607 4271

Korea

Dialog Semiconductor Korea
Phone: +82 2 3469 8200

China (Shenzhen)

Dialog Semiconductor China
Phone: +86 755 2981 3669

China (Shanghai)

Dialog Semiconductor China
Phone: +86 21 5424 9058