

# WizFi360

## AT Instruction Set

Version 1.1.1.2



<http://www.wiznet.io/>

# Contents

1	Document Revision History.....	3
2	AT Command Overview .....	4
2.1	AT Command Format .....	4
2.2	AT command returns a list of values .....	5
2.3	List of Messages.....	6
2.4	Enter normal transmission mode .....	7
3	AT Command Description .....	8
3.1	System Control Commands.....	11
3.1.1	AT: Tests AT Startup.....	11
3.1.2	AT+RST: Restarts the module .....	11
3.1.3	AT+GMR: Checks Version Information.....	11
3.1.4	AT+GSLP: Enters Deep-sleep Mode .....	11
3.1.5	ATE: AT Commands Echoing.....	12
3.1.6	AT+RESTORE: Restores the Factory Default settings.....	12
3.1.7	AT+UART_CUR: Current UART Configuration; Not saved to Flash .....	12
3.1.8	AT+UART_DEF: Default UART Configuration; Saved in the Flash .....	13
3.1.9	AT+SLEEP: Configures the Sleep Modes .....	14
3.1.10	AT+SYSIOSETCFG: Configures IO Working Mode .....	14
3.1.11	AT+SYSIOGETCFG: Checks IO Working Mode.....	15
3.1.12	AT+SYSGPIODIR: Configures the Direction of GPIO .....	16
3.1.13	AT+SYSGPIOWRITE: Configures the GPIO Output Level.....	16
3.1.14	AT+SYSGPIOREAD: Reads the GPIO Input Level.....	17
3.2	WiFi command.....	18
3.2.1	AT+CWMODE_CUR: Sets the Current WiFi mode; Not Saved in the Flash .....	18
3.2.2	AT+CWMODE_DEF: Set the operation mode, Save to Flash.....	18
3.2.3	AT+CWJAP_CUR: Connects to an AP; Configuration Not Saved in the Flash .....	19
3.2.4	AT+CWJAP_DEF: Connects to an AP; Configuration Saved in the Flash.....	19
3.2.5	AT+CWLAPOPT: Sets the Configuration for the Command AT+CWLAP .....	20
3.2.6	AT+CWLAP: Lists Available APs .....	21
3.2.7	AT+CWQAP: Disconnects from the AP.....	22
3.2.8	AT+CWSAP_CUR: Configures the WizFi360 SoftAP; Configuration Not Saved in the Flash .....	22
3.2.9	AT+CWSAP_DEF: Configures the WizFi360 SoftAP; Configuration Saved in the Flash .....	23
3.2.10	AT+CWLIF: IP of Stations which are connected to WizFi360 SoftAP.....	24
3.2.11	AT+CWDHCP_CUR: Enables/Disables DHCP; Configuration Not Saved in the Flash.....	24
3.2.12	AT+CWDHCP_DEF: Enables/Disables DHCP; Configuration Saved in the Flash .....	25
3.2.13	AT+CWDHCPS_CUR: Sets the IP Address Allocated by WizFi360 SoftAP DHCP; Configuration Not Saved in Flash .....	26
3.2.14	AT+CWDHCPS_DEF: Sets the IP Address Allocated by WizFi360 SoftAP DHCP; Configuration Saved in Flash.....	26
3.2.15	AT+CWAUTOCONN: Auto-Connects to the AP or Not .....	27
3.2.16	AT+CIPSTAMAC_CUR: Sets the MAC Address of the WizFi360 Station; Configuration Not Saved in the Flash .....	28
3.2.17	AT+CIPSTAMAC_DEF: Sets the MAC Address of the WizFi360 Station; Configuration Saved in the Flash.....	28
3.2.18	AT+CIPAPMAC_CUR: Sets the MAC Address of the WizFi360 SoftAP; Configuration Not Saved in the Flash.....	29
3.2.19	AT+CIPAPMAC_DEF: Sets the MAC Address of the WizFi360 SoftAP; Configuration Saved in the Flash .....	29
3.2.20	AT+CIPSTA_CUR: Sets the Current IP Address of the WizFi360 Station; Configuration Not Saved in the Flash .....	30
3.2.21	AT+CIPSTA_DEF: Set the static IP of WizFi360 Station, Saved to Flash .....	31
3.2.22	AT+CIPAP_CUR: Sets the IP Address of the WizFi360 SoftAP; Configuration Not Saved in the Flash .....	

.....	31
3.2.23 AT+CIPAP_DEF: Sets the IP Address of the WizFi360 SoftAP; Configuration Saved in the Flash .....	32
3.2.24 AT+CWSTARTSMART: Start SmartConfig .....	33
3.2.25 AT+CWSTOPSMART: Stop Smart Config .....	35
3.2.26 AT+WPS: Enables the WPS Function .....	35
3.2.27 AT+CWHOSTNAME: Configures the Name of WizFi360 Station .....	35
3.2.28 AT+CWCOUNTRY_CUR: Set WiFi Country Code of WizFi360; Configuration Not Saved in the Flash .....	36
3.2.29 AT+CWCOUNTRY_DEF: Set WiFi Country Code of WizFi360; Configuration Saved in the Flash .....	37
3.2.30 AT+WIZ_NETCONFIG: WebServer for setting SSID/PWD .....	38
3.3 TCP / IP command .....	39
3.3.1 AT+CIPSTATUS: Gets the Connection Status .....	39
3.3.2 AT+CIPDOMAIN: DNS Function .....	39
3.3.3 AT+CIPSTART: Establishes TCP Connection, UDP Transmission or SSL Connection .....	40
3.3.4 AT+CIPSSLSIZE: Sets the Size of SSL Buffer .....	42
3.3.5 AT+CIPSSLCONF: Sets Configuration of WizFi360 SSL Client .....	42
3.3.6 AT+CASEND: Sets the SSL certificate .....	43
3.3.7 AT+CIPSEND: Send data .....	43
3.3.8 AT+CIPSENDEX: Sends data .....	45
3.3.9 AT+CIPSENDERBUF: Writes Data into the TCP-Send-Buffer .....	45
3.3.10 AT+CIPBUFRESET: Resets the Segment ID Count .....	46
3.3.11 AT+CIPBUFSTATUS: Checks the Status of TCP-Send-Buffer .....	47
3.3.12 AT+CIPCHECKSEQ: Checks If a Specific Segment Was Successfully Sent .....	47
3.3.13 AT+CIPCLOSE: Closes the TCP/UDP/SSL Connection .....	48
3.3.14 AT+CIFSR: Gets the Local IP Address .....	48
3.3.15 AT+CIPMUX: Enable or Disable Multiple Connections .....	49
3.3.16 AT+CIPSERVER: Deletes/Creates TCP Server .....	50
3.3.17 AT+CIPSERVERMAXCONN: Set the Maximum Connection Number Allowed by Server .....	50
3.3.18 AT+CIPMODE: Sets transmission mode .....	51
3.3.19 AT+SAVETRANSLINK: Saves the Transparent Transmission Link in Flash; .....	51
3.3.20 AT+CIPSTO: Sets the TCP Server Timeout .....	52
3.3.21 AT+CIUPDATE: Updates the Software Through WiFi .....	52
3.3.22 AT+PING: Ping Packets .....	53
3.3.23 AT+CIPDINFO: Shows the Remote IP and Port with +IPD .....	54
3.3.24 +IPD: Receive Network Data .....	54
3.3.25 AT+CIPSNTPCFG: Sets the Configuration of SNTP .....	54
3.3.26 AT+CIPSNTPTIME: Checks the SNTP Time .....	55
3.3.27 AT+CIPDNS_CUR: Sets User-defined DNS Servers; Configuration Not Saved in the Flash .....	55
3.3.28 AT+CIPDNS_DEF: Sets User-defined DNS Servers; Configuration Saved in the Flash .....	56
3.3.29 AT+MQTTSET: Sets the Configuration of MQTT connection. ....	57
3.3.30 AT+MQTTTOPIC: Sets the Topic of Publish and Subscribe .....	57
3.3.31 AT+MQTTCON: Connects to a Broker .....	58
3.3.32 AT+MQTTPUB: Publish a message .....	58
3.3.33 AT+MQTTDIS: Disconnects from a Broker .....	59
3.3.34 AT+AZSET: Sets the Configuration of Azure IoT Hub connection. ....	59
3.3.35 AT+AZCON: Connects to a Azure IoT Hub. ....	59
3.3.36 AT+AWSPKSEND: Sets the Private Key .....	60
3.3.37 AT+CLICASEND: Sets the Client certificate .....	61
3.3.38 AT+AWSCON: Connects to AWS IoT Core. ....	62
4 Command History .....	63

# 1 Document Revision History

Version	Date	Descriptions
Ver. 1.0.0	1AUG2019	Initial Release
Ver. 1.0.1	14AUG2019	Add AT+CWCOUNTRY_CUR, AT+CWCOUNTRY_DEF, AT+SYSIOSETCFG, AT+SYSIOGETCFG, AT+SYSGPIODIR, AT+SYSGPIOWRITE, AT+SYSGPIOREAD, Pin List, AT+CIPSERVERMAXCONN, AT+CWSTARTSMART, AT+CWSTOPSMART, NOTE of UART_CUR(for PA1), Modify return value of AT+CIFSR, description of AT+SAVETRANSLINK, option of AT+CWLAP and AT+CWLAPOPT(adding wps parameter), AT Command Overview, description of AT+RESTORE, description of CIPAPMAC(not change the value)
Ver. 1.0.2	20AUG2019	Renewal Layout Add AT+WPS, AT+SLEEP, AT+GSLP
Ver. 1.0.3	23AUG2019	Add MQTT commands Add description of AT+CIPSTART SSL
Ver. 1.0.4	16SEP2019	Modify typing error
Ver.1.0.5	14OCT2019	Modify CWMODE Factory Default
Ver. 1.0.6	16OCT2019	Add Azure commands, AT+SSLCONF command and AT+CASEND command. Modify typing error Add Command History
Ver 1.0.7	09MAR2020	Add AT+WIZ_NETCONFIG Modify AT+CWSTARTSMART Modify minor typos and error
Ver 1.0.7.2	12MAR2020	Modify AT+WIZ_NETCONFIG Modify AT+CWSTARTSMART Modify minor typos and error
Ver 1.1.1.1	7JAN2021	Modify AT+MQTTCON
Ver 1.1.1.2	22APR2021	Add AT+AWSCON, AT+AWSPKSEND, AT+CLICASEND

## 2 AT Command Overview

### 2.1 AT Command Format

AT command is of the following type. Not all AT commands support all four variations.

Command Type	Command Format	Functional Description
Test Command	AT	Query to see if the module is in normal transmission mode
Set Command	AT+<command>=<para0>,<para1>,...,<paraN>	Set the value of a particular parameter. The number of parameters is different for each command.
Query Command	AT+<command>?	Query the current setting of a particular parameter value
Execute Command	AT+<command>	Performs a specific function

Note:

1. AT command must be capitalized, start with AT and end with CR LF(=0x0A 0x0d).
2. <command> means each command. For example, <command> can be RST or CWMODE\_CUR.
3. AT command can have several parameters, separated by a comma.
4. Optional parameters are indicated in square brackets [ ]. It may be either not required or not appear, and set to the default value if it is not set.
5. String values have to set in double quotation mark.

## 2.2 AT command returns a list of values

Return values for AT Command are as follows.

Return Type	Return value	Description
<b>Error Messages</b>	ERROR	AT command input error or execution error
	ALREADY CONNECTED	The TCP, UDP or SSL connection is already established.
	SEND FAIL	The network data transmission is failed.
<b>Success Message</b>	OK	Set command is executed correctly.
	+<Command>: <para1>,<para2>, ...,<paran> OK	Query or Execute command is executed correctly and return the parameter value.
	SEND OK	The network data transmission is success.
	<description> OK	Query or Execute command is executed correctly and return the specific value. <description> means the returned values, and the returned values are different for each command.

## 2.3 List of Messages

In addition to the return value for command, the following message is returned.

Tips	Explanation
ready	The AT firmware is ready.
WiFi CONNECTED	WizFi360 Station connected to the AP
WiFi GOT IP	WizFi360 Station got IP address from the AP
WiFi DISCONNECTED	WizFi360 Station disconnected from the AP
busy s...	It means busy sending. WizFi360 is sending for previous input, cannot response to the new input.
busy p...	It means busy processing. WizFi360 is processing for previous input, cannot response to the new input.
<Link ID>, CONNECT	A network connection of which <Link ID>
<Link ID>, CLOSED	A network close of which <Link ID>
+IPD	Received network data.
+STA_CONNECTED: <mac>	A Station connects to the WizFi360 SoftAP
+DIST_STA_IP: <mac>, <ip addr>	WizFi360 SoftAP distributes an IP address to the Station connected.
+STA_DISCONNECTED: <smac>	A Station disconnects to the WizFi360 SoftAP.

## 2.4 Enter normal transmission mode

There are normal transmission mode and transparent mode in WizFi360.

In case WizFi360 is Normal Command mode, WizFi360 executes AT command. Confirm Normal Command mode by inputting AT\r\n and returning \r\nOK\r\n.

In case transparent mode, WizFi360 doesn't execute AT command. Only transmit and receive data with peer. If input is "+++", switch to normal command mode.

*Note:*

- 1. In case that TCP connection is established and WizFi360 is transparent mode, don't switch to AT command to keep TCP connection.*
- 2. "+++" input rule: three "+" must be continuously transmitted by serial. After at least 1s, WizFi360 can be respond for AT Command.*
- 3. Factory default mode of WizFi360 is Normal command mode*



## 3 AT Command Description

### AT Command list

Type	Name	Features
<b>System control commands</b>	AT	Tests AT Startup
	AT+RST	Restarts the module
	AT+GMR	Checks Version Information
	AT+GSLP	Enters Deep-sleep Mode
	ATE	AT Commands Echoing
	AT+RESTORE	Restores the Factory Default settings
	AT+UART_CUR	Current UART Configuration; Not saved to Flash
	AT+UART_DEF	Default UART Configuration; Saved in the Flash
	AT+SLEEP	Configures the Sleep Modes
	AT+SYSIOSETCFG	Configures IO Working Mode
	AT+SYSIOGETCFG	Checks IO Working Mode
	AT+SYSGPIODIR	Configures the Direction of a GPIO
	AT+SYSGPIOWRITE	Configures the GPIO Output Level
	AT+SYSGPIOREAD	Reads the GPIO Input Level
<b>WiFi command</b>	AT+CWMODE_CUR	Sets the Current WiFi mode; Not Saved in the Flash
	AT+CWMODE_DEF	Set the operation mode, Save to Flash
	AT+CWJAP_CUR	Connects to an AP; Configuration Not Saved in the Flash
	AT+CWJAP_DEF	Connects to an AP; Configuration Saved in the Flash
	AT+CWLAPOPT	Sets the Configuration for the Command AT+CWLAP
	AT+CWLAP	Lists Available APs
	AT+CWQAP	Disconnects from the AP
	AT+CWSAP_CUR	Configures the WizFi360 SoftAP; Configuration Not Saved in the Flash
	AT+CWSAP_DEF	Configures the WizFi360 SoftAP; Configuration Saved in the Flash
	AT+CWLIF	IP of Stations to Which the WizFi360 SoftAP is Connected
	AT+CWDHCP_CUR	Enables/Disables DHCP; Configuration Not Saved in the Flash
	AT+CWDHCP_DEF	Enables/Disables DHCP; Configuration Saved in the Flash
	AT+CWDHCPS_CUR	Sets the IP Address Allocated by WizFi360 SoftAP DHCP; Configuration Not Saved in Flash

	AT+CWDHCPS_DEF	Sets the IP Address Allocated by WizFi360 SoftAP DHCP; Configuration Saved in Flash
	AT+CWAUTOCONN	Auto-Connects to the AP or Not
	AT+CIPSTAMAC_CUR	Sets the MAC Address of the WizFi360 Station; Configuration Not Saved in the Flash
	AT+CIPSTAMAC_DEF	Sets the MAC Address of the WizFi360 Station; Configuration Saved in the Flash
	AT+CIPAPMAC_CUR	Sets the MAC Address of the WizFi360 SoftAP; Configuration Not Saved in the Flash
	AT+CIPAPMAC_DEF	Sets the MAC Address of the WizFi360 SoftAP; Configuration Saved in the Flash
	AT+CIPSTA_CUR	Sets the Current IP Address of the WizFi360 Station; Configuration Not Saved in the Flash
	AT+CIPSTA_DEF	Set the static IP of WizFi360 Station, Saved to Flash
	AT+CIPAP_CUR	Sets the IP Address of the WizFi360 SoftAP; Configuration Not Saved in the Flash
	AT+CIPAP_DEF	Sets the IP Address of the WizFi360 SoftAP; Configuration Saved in the Flash
	AT+CWSTARTSMART	Start SmartConfig
	AT+CWSTOPSMART	Stop Smart Config
	AT+WPS	Enables the WPS Function
	AT+CWHOOSTNAME	Configures the Name of WizFi360 Station
	AT+CWCOUNTRY_CUR	Set WiFi Country Code of WizFi360; Configuration Not Saved in the Flash
	AT+CWCOUNTRY_DEF	Set WiFi Country Code of WizFi360; Configuration Saved in the Flash
	AT+WIZ_NETCONFIG	WebServer for setting SSID/PWD (Default: 192.168.36.1)
TCP / IP command	AT+CIPSTATUS	Gets the Connection Status
	AT+CIPDOMAIN	DNS Function
	AT+CIPSTART	Establishes TCP Connection, UDP Transmission or SSL Connection
	AT+CIPSSLSIZE	Sets the Size of SSL Buffer
	AT+SSLCONF	Sets Configuration of WiFi360 SSL Client
	AT+CASEND	Sets the SSL certificate
	AT+CIPSEND	Send data
	AT+CIPSENDEX	Sends data when length of data is <length>, or when \0 appears in the data
	AT+CIPSENDERBUF	Writes Data into the TCP-Send-Buffer
	AT+CIPBUFRESET	Resets the Segment ID Count
	AT+CIPBUFSTATUS	Checks the Status of TCP-Send-Buffer
	AT+CIPCHECKSEQ	Checks If a Specific Segment Was Successfully Sent

AT+CIPCLOSE	Closes the TCP/UDP/SSL Connection
AT+CIFSR	Gets the Local IP Address
AT+CIPMUX	Enable or Disable Multiple Connections
AT+CIPSERVER	Deletes/Creates TCP Server
AT+CIPSERVERMAXCONN	Set the Maximum Connection Number Allowed by Server
AT+CIPMODE	Sets transmission mode
AT+SAVETRANSLINK	Saves the Transparent Transmission Link in Flash;
AT+CIPSTO	Sets the TCP Server Timeout
AT+CIUPDATE	Update the Firmware
AT+PING	Ping Packets
AT+CIPDINFO	Shows the Remote IP and Port with +IPD
+IPD	Receive Network Data
AT+CIPSNTPCFG	Sets the Configuration of SNTP
AT+CIPSNTPIME	Checks the SNTP Time
AT+CIPDNS_CUR	Sets User-defined DNS Servers; Configuration Not Saved in the Flash
AT+CIPDNS_DEF	Sets User-defined DNS Servers; Configuration Saved in the Flash
AT+MQTTSET	Sets the Configuration of MQTT connection
AT+MQTTTOPIC	Sets the Topic of Publish and Subscribe
AT+MQTTCON	Connects to a Broker
AT+MQTTPUB	Publish a message
AT+MQTTDIS	Disconnects from a Broker
AT+AZSET	Sets the Configuration of Azure IoT Hub connection
AT+AZCON	Connects to a Azure IoT Hub
AT+AWSPKSEND	Sets Private Key
AT+CLICASEND	Sets Client Certificate
AT+AWSCON	Connects to AWS IoT Core

## 3.1 System Control Commands

### 3.1.1 AT: Tests AT Startup

	Execute command
<b>Commands</b>	AT
<b>Response</b>	OK

### 3.1.2 AT+RST: Restarts the module

	Execute command
<b>Commands</b>	AT+RST
<b>Response</b>	OK

### 3.1.3 AT+GMR: Checks Version Information

	Execute Command
<b>Commands</b>	AT+GMR
<b>Response</b>	<AT version info> <SDK version info> <compile time>  OK
<b>Parameter</b>	<AT version info>: information about the AT version. <SDK version info>: information about the SDK version. <compile time>: the duration of time for compiling the BIN.
<b>Example</b>	AT+GMR  AT version:1.0.1.0(Jun 6 2019 17:49:31) SDK version:3.0.0(a0ffff9f) compile time:Jun 6 2019 17:49:31  OK

### 3.1.4 AT+GSLP: Enters Deep-sleep Mode

	Set Command
<b>Commands</b>	AT+GSLP=<time>
<b>Function</b>	WizFi360 will wake up after Deep-sleep for as many milliseconds (ms) as <time> indicates.

<b>Response</b>	OK
<b>Parameter</b>	<time>: the duration of WizFi360's sleep within the range of 1000~65535 ms.
<b>Example</b>	AT+GSLP=3000
	OK

### 3.1.5 ATE: AT Commands Echoing

	Execute command
<b>Commands</b>	ATE
<b>Response</b>	OK
<b>Parameter</b>	ATE0: Switches echo off ATE1: Switches echo on.
<b>Note</b>	This command is used to trigger command echo. It means that entered commands can be echoed back to the sender after ATE command. Two Parameter are possible. The command returns OK in normal cases and ERROR when a parameter other than 0 or 1 was specified.

### 3.1.6 AT+RESTORE: Restores the Factory Default settings

	Execute command
<b>Commands</b>	AT+RESTORE[=<type>]
<b>Response</b>	OK
<b>Parameter</b>	[<type>]: <ul style="list-style-type: none"> <li>• 0: Restore only Station mac address factory setting (default)</li> <li>• 1: Restore all factory setting</li> </ul>
<b>Note</b>	This command will restore the Station mac address or all data in flash to the factory default settings. The module will restart after this command.

### 3.1.7 AT+UART\_CUR: Current UART Configuration; Not saved to Flash

	Query command	Set Command
<b>Commands</b>	AT+UART_CUR?	AT+UART_CUR=<baudrate>,<databits>,<stop bits>,<parity>,<flow control>
<b>Response</b>	+UART_CUR:<baudrate>,<databits>,<stop bits>,<parity>,<flow control> OK	OK
<b>Default Value</b>	115200,8,1,0,0	

<b>Parameter</b>	<p>&lt;baudrate&gt;: UART baud rate 2000000, 1500000, 1000000, 921600, 460800, 230400, 115200 (factory default), 57600, 38400, 19200, 14400, 9600, 4800, 2400, 1800, 1200, 600</p> <p>&lt;databits&gt;: data bits</p> <ul style="list-style-type: none"> <li>• 5: 5-bit data</li> <li>• 6: 6-bit data</li> <li>• 7: 7-bit data</li> <li>• 8: 8-bit data (factory default)</li> </ul> <p>&lt;stopbits&gt;: stop bits</p> <ul style="list-style-type: none"> <li>• 1: 1-bit stop bit (factory default)</li> <li>• 2: 2-bit stop bit</li> </ul> <p>&lt;parity&gt;: parity bit</p> <ul style="list-style-type: none"> <li>• 0: None (factory default)</li> <li>• 1: Odd</li> <li>• 2: Even</li> </ul> <p>&lt;flow control&gt;: flow control</p> <ul style="list-style-type: none"> <li>• 0: disable RTS/CTS flow control (factory default)</li> <li>• 1: enable RTS/CTS flow control</li> </ul>	
<b>Note</b>	<ul style="list-style-type: none"> <li>• The configuration changes will NOT be saved in the flash.</li> <li>• This command is linked with the PA1 pin of WizFi360. When the PA1 pin (refer to the WizFi360 datasheet) pulled down for more than 3 seconds, the Parameter of the command are restored to the default values.</li> <li>• If the WizFi360 hardware flow control function is enabled, the user device should connect to the flow control pin of WizFi360. For details, please refer to the WizFi360 datasheet.</li> </ul>	
<b>Example</b>	AT+UART_CUR?	AT+UART_CUR=115200,8,1,0,0
	+UART_CUR:115200,8,1,0,0 OK	OK

### 3.1.8 AT+UART\_DEF: Default UART Configuration; Saved in the Flash

	Query command	Set Command
<b>Commands</b>	AT+UART_DEF?	AT+UART_DEF=<baudrate>,<databits>,<stop bits>,<parity>,<flow control>
<b>Response</b>	+UART_DEF:<baudrate>,<databits>,<stop bits>,<parity>,<flow control> OK	OK
<b>Default Value</b>	115200,8,1,0,0	
<b>Parameter</b>	<p>&lt;baudrate&gt;: UART baud rate 2000000, 1500000, 1000000, 921600, 460800, 230400, 115200 (factory default), 57600, 38400, 19200, 14400, 9600, 4800, 2400, 1800, 1200, 600</p> <p>&lt;databits&gt;: data bits</p> <ul style="list-style-type: none"> <li>• 5: 5-bit data</li> <li>• 6: 6-bit data</li> <li>• 7: 7-bit data</li> <li>• 8: 8-bit data (factory default)</li> </ul> <p>&lt;stopbits&gt;: stop bits</p>	

	<ul style="list-style-type: none"> <li>• 1: 1-bit stop bit (factory default)</li> <li>• 2: 2-bit stop bit</li> </ul> <p>&lt;parity&gt;: parity bit</p> <ul style="list-style-type: none"> <li>• 0: None (factory default)</li> <li>• 1: Odd</li> <li>• 2: Even</li> </ul> <p>&lt;flow control&gt;: flow control</p> <ul style="list-style-type: none"> <li>• 0: flow control is not enabled (factory default)</li> <li>• 1: enable RTS/CTS flow control</li> </ul>	
<b>Note</b>	<ul style="list-style-type: none"> <li>• The configuration changes will be saved in the system parameter area in the flash.</li> <li>• If the WizFi360 hardware flow control function is enabled, the user device should connect to the flow control pin of WizFi360. For details, please refer to the WizFi360 datasheet.</li> </ul>	
<b>Example</b>	AT+UART_DEF?	AT+UART_DEF=115200,8,1,0,0
	+UART_DEF:115200,8,1,0,0	OK
	OK	

### 3.1.9 AT+SLEEP: Configures the Sleep Modes

	Query Command	Set Command
Commands	AT+SLEEP?	AT+SLEEP=<sleep mode>
Response	+SLEEP:<sleep mode>  OK	OK
Parameter	<sleep mode>: <ul style="list-style-type: none"><li>• 0: disables sleep mode</li><li>• 1: Light-sleep mode</li><li>• 2: Modem-sleep mode (factory default)</li></ul>	
Example	AT+SLEEP?	AT+SLEEP=1
	+SLEEP:2  OK	OK
Note	<ul style="list-style-type: none"><li>• This command can only be used in Station mode.</li></ul>	

### 3.1.10 AT+SYSIOSETCFG: Configures IO Working Mode

	Set command
<b>Commands</b>	AT+SYSIOSETCFG=<pin>,<mode>,<pull-up>
<b>Response</b>	OK
<b>Parameter</b>	<pin>: IO pin number

	<mode>: IO mode		
	Pin	Mode0	Mode1
	3	RESERVED	GPIOPA_0
	6	RESERVED	GPIOPB_6
	7	UART1_CTS	GPIOPB_9
	9	RESERVED	GPIOPB_15
	10	RESERVED	GPIOPB_18
	11	RESERVED	GPIOPB_13
	12	RESERVED	GPIOPB_14
	13	RESERVED	GPIOPB_17
	14	RESERVED	GPIOPB_16
	16	UART1_RTS	GPIOPB_10
	19	RESERVED	GPIOPB_7
20	RESERVED	GPIOPB_8	
	<pull-up>:		
	• 0: Disable pull-up		
	• 1: Enable pull-up		
Example	AT+SYSIOSETCFG=12,1,0		
	OK		

### 3.1.11 AT+SYSIOGETCFG: Checks IO Working Mode

	Set command
Commands	AT+SYSIOGETCFG=<pin>
Response	+SYSIOGETCFG:<pin>,<mode>,<pull-up> OK
Parameter	<p>&lt;pin&gt;: IO pin number</p> <p>&lt;mode&gt;:</p> <ul style="list-style-type: none"> <li>• 0: default mode</li> <li>• 1: GPIO mode</li> </ul> <p>&lt;pull-up&gt;:</p> <ul style="list-style-type: none"> <li>• 0: Disable pull-up</li> <li>• 1: Enable pull-up</li> </ul>
Example	AT+SYSIOGETCFG=12
	+SYSIOGETCFG:12,1,0 OK



### 3.1.12 AT+SYSGPIODIR: Configures the Direction of GPIO

	Set command
<b>Commands</b>	AT+SYSGPIODIR=<pin>,<dir>
<b>Response</b>	OK
	NOT GPIO MODE! ERROR
<b>Note</b>	If IO pin mode is not GPIO mode, the command will return "NOT GPIO MODE!"
<b>Parameter</b>	<pin>: IO pin number <dir>: <ul style="list-style-type: none"> <li>• 0: Set the pin to input mode (If GPIO Direction is input, set to pull-up automatically)</li> <li>• 1: Set the pin to output mode</li> </ul>
<b>Example</b>	AT+SYSIOSETCFG=12,1,1
	OK
	AT+SYSGPIODIR=12,0
	OK

### 3.1.13 AT+SYSGPIOWRITE: Configures the GPIO Output Level

	Set command
<b>Commands</b>	AT+SYSGPIOWRITE=<pin>,<level>
<b>Response</b>	OK
	NOT GPIO MODE! ERROR
<b>Note</b>	If IO pin mode is not output mode, the command will return "NOT OUTPUT MODE!"
<b>Parameter</b>	<pin>: IO pin number <level>: <ul style="list-style-type: none"> <li>• 0: Set the pin to low level</li> <li>• 1: Set the pin to high level</li> </ul>
<b>Example</b>	AT+SYSIOSETCFG=12,1,1
	OK
	AT+SYSGPIOWRITE=12,0
	OK

### 3.1.14 AT+SYSGPIOREAD: Reads the GPIO Input Level

	Set command
<b>Commands</b>	AT+SYSGPIOREAD=<pin>
<b>Response</b>	+SYSGPIOREAD:<pin>,<dir>,<level>
	OK
	NOT GPIO MODE! ERROR
<b>Note</b>	If IO pin mode is not GPIO mode, the command will return "NOT GPIO MODE!"
<b>Parameter</b>	<pin>: IO pin number <dir>: <ul style="list-style-type: none"> <li>• 0: input mode</li> <li>• 1: output mode</li> </ul> <level>: <ul style="list-style-type: none"> <li>• 0: low level</li> <li>• 1: high level</li> </ul>
<b>Example</b>	AT+SYSIOSETCFG=12,1,1
	OK
	AT+SYSGPIODIR=12,0
	OK
	AT+SYSGPIOREAD=12
	+SYSGPIOREAD:12,0,1
	OK

## 3.2 WiFi command

### 3.2.1 AT+CWMODE\_CUR: Sets the Current WiFi mode; Not Saved in the Flash

	Query command	Set Command
Commands	AT+CWMODE_CUR?	AT+CWMODE_CUR=<mode>
Response	+CWMODE_CUR:<mode>  OK	OK
Parameter	<mode>: <ul style="list-style-type: none"><li>• 1: Station mode</li><li>• 2: SoftAP mode (factory default)</li><li>• 3: Station + SoftAP mode</li></ul>	
Example	AT+CWMODE_CUR?	AT+CWMODE_CUR=1
	AT+CWMODE_CUR:1  OK	OK
Note	The configuration changes will NOT be saved in the flash.	

### 3.2.2 AT+CWMODE\_DEF: Set the operation mode, Save to Flash

	Query command	Set Command
Commands	AT+CWMODE_DEF?	AT+CWMODE_DEF=<mode>
Response	+CWMODE_DEF:<mode>  OK	OK
Parameter	<mode>: <ul style="list-style-type: none"><li>• 1: Station mode</li><li>• 2: SoftAP mode (factory default)</li><li>• 3: Station + SoftAP mode</li></ul>	
Example	AT+CWMODE_DEF?	AT+CWMODE_DEF=1
	+CWMODE_DEF:1  OK	OK
Note	The configuration changes will be saved in the system parameter area in the flash.	

### 3.2.3 AT+CWJAP\_CUR: Connects to an AP; Configuration Not Saved in the Flash

	Query command	Set Command
<b>Commands</b>	AT+CWJAP_CUR?	AT+CWJAP_CUR=<ssid>,<pwd>[,<bssid>]
<b>Function</b>	To query the AP to which the WizFi360 Station is already connected.	To set the AP to which the WizFi360 Station needs to be connected.
<b>Response</b>	+CWJAP_CUR:<ssid>,<bssid>,<channel>,<rssi>  OK	OK
		+CWJAP_CUR:<error code>  FAIL
<b>Parameter</b>	<ssid>: string parameter, the SSID of the target AP, MAX: 32 bytes. <pwd>: string parameter, the password of the target AP, MAX: 64-byte ASCII. [<bssid>]: string parameter, the MAC address of the target AP, used when multiple APs have the same SSID. <channel>: channel number <rssi>: signal strength. <error code>: (for reference only) <ul style="list-style-type: none"> <li>• 1: connection timeout.</li> <li>• 2: wrong password.</li> <li>• 3: cannot find the target AP.</li> <li>• 4: connection failed.</li> </ul>	
<b>Example</b>	AT+CWJAP_CUR?	AT+CWJAP_CUR="ab\\,c","12345678\\", "00:08:DC:11:12:13" (SSID: ab\,c Password: 12345678\)
	+CWJAP_CUR="WIZNETSZ","00:08:dc:9c:ef:b6",12,-75  OK	OK
<b>Note</b>	<ul style="list-style-type: none"> <li>• The configuration changes will NOT be saved in the flash.</li> <li>• This command is only available in Station mode and SoftAP+Station mode.</li> <li>• If the SSID or password contains any special characters such as " , \, you need an escape character.</li> </ul>	

### 3.2.4 AT+CWJAP\_DEF: Connects to an AP; Configuration Saved in the Flash

	Query command	Set Command
<b>Commands</b>	AT+CWJAP_DEF?	AT+CWJAP_DEF=<ssid>,<pwd>[,<bssid>]
<b>Function</b>	To query the AP to which the WizFi360 Station is already connected.	To set the AP to which the WizFi360 Station needs to be connected.
<b>Response</b>	+CWJAP_DEF:<ssid>,<bssid>,<channel>,<rssi>	OK

	OK	+CWJAP_DEF:<error code>  FAIL
Parameter	<p>&lt;ssid&gt;: string parameter, the SSID of the target AP, MAX: 32 bytes.</p> <p>&lt;pwd&gt;: string parameter, the password of the target AP, MAX: 64-byte ASCII.</p> <p>[&lt;bssid&gt;]: string parameter, the MAC address of the target AP, used when multiple APs have the same SSID.</p> <p>&lt;channel&gt;: channel number</p> <p>&lt;rssi&gt;: signal strength.</p> <p>&lt;error code&gt;: (for reference only)</p> <ul style="list-style-type: none"><li>• 1: connection timeout.</li><li>• 2: wrong password.</li><li>• 3: cannot find the target AP.</li><li>• 4: connection failed.</li></ul>	
Example	AT+CWJAP_DEF?	AT+CWJAP_DEF="ab\\,c","12345678\\", "00:08:DC:11:12:13" (SSID: ab,c Password: 12345678\)
	+CWJAP_DEF="WIZNETSZ","00:08:dc:9c:ef:b6",12,-75	OK
	OK	
Note	<ul style="list-style-type: none"><li>• The configuration changes will be saved in the user parameter area in the flash.</li><li>• This command is only available in Station mode and SoftAP+Station mode.</li><li>• If the SSID or password contains special characters such as " , \ , you need an escape character.</li></ul>	

### 3.2.5 AT+CWLAPOPT: Sets the Configuration for the Command AT+CWLAP

	Set command																						
Commands	AT+CWLAPOPT=<sort_enable>,<mask>																						
Response	OK																						
Parameter	<p>&lt;sort_enable&gt;: determines whether the result of command AT+CWLAP will be listed according to RSSI:</p> <ul style="list-style-type: none"><li>• 0: the result is not ordered according to RSSI (factory default)</li><li>• 1: the result is ordered according to RSSI.</li></ul> <p>&lt;mask&gt;: determines the Parameter shown in the result of AT+CWLAP; 0 means not showing the parameter corresponding to the bit, and 1 means showing it.</p> <table><tr><td>Bit10</td><td>Bit9</td><td>Bit8</td><td>Bit7</td><td>Bit6</td><td>Bit5</td><td>Bit4</td><td>Bit3</td><td>Bit2</td><td>Bit1</td><td>Bit0</td></tr><tr><td>WPS</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>CH</td><td>MAC</td><td>RSSI</td><td>SSID</td><td>ECN</td></tr></table>	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	WPS	-	-	-	-	-	CH	MAC	RSSI	SSID	ECN
Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0													
WPS	-	-	-	-	-	CH	MAC	RSSI	SSID	ECN													
Example	AT+CWLAPOPT=1,1055 (The first parameter is 1, meaning that the result of the command AT+CWLAP will be ordered according to RSSI;																						

	The second parameter is 1055, which is 10000011111 in binary; meaning that the corresponding bits of <mask> are all set to 1. All Parameter will be shown in the result of AT+CWLAP.)
	OK
<b>Note</b>	<ul style="list-style-type: none"> <li>The configuration changes will NOT be saved in the flash.</li> </ul>

### 3.2.6 AT+CWLAP: Lists Available APs

	Query command	Set Command
<b>Commands</b>	AT+CWLAP	AT+CWLAP=<ssid>[,<mac>][,<channel>]
<b>Function</b>	To list all available APs..	To query the APs with specific SSID and MAC on a specific channel.
<b>Response</b>	+CWLAP:([<ecn>,<ssid>,<rssi>,<mac>,<channel>,<wps>]) OK	
<b>Parameter</b>	<ecn>: encryption method. • 0: OPEN • 1: WEP • 2: WPA_PSK • 3: WPA2_PSK • 4: WPA_WPA2_PSK <ssid>: string parameter, SSID of the AP. <rssi>: signal strength. [<mac>]: string parameter, MAC address of the AP. [<channel>]: channel number <wps>: • 0: WPS is disabled • 1: WPS is enabled	
<b>Example1</b>	AT+CWLAP	AT+CWLAP="WIZNETSZ"
	+CWLAP: (4,"WIZnet",-57,"00:08:dc:6a:46:2e",1,1) +CWLAP: (3,"WIZNETSZ",-75,"00:08:dc:9c:ef:b6",12,1) OK	+CWLAP:(3,"WIZNETSZ",-75,"00:08:dc:9c:ef:b6",12,1) OK
<b>Example2</b>	-	AT+CWLAP="WIZNETSZ","",12
		+CWLAP:(3,"WIZNETSZ",-75,"00:08:dc:9c:ef:b6",12,1)
		OK
<b>Note</b>	<ul style="list-style-type: none"> <li>This command is only available in Station mode and SoftAP+Station mode.</li> <li>The parameters displayed of Response change according to the setting of CWLAPOPT command.</li> <li>If you want to set the optional parameter &lt;channel&gt; only, write it as in Example2.</li> </ul>	

### 3.2.7 AT+CWQAP: Disconnects from the AP

	Execute Command
<b>Commands</b>	AT+CWQAP
<b>Response</b>	OK
<b>Example</b>	AT+CWMODE_DEF=1
	OK
	AT+CWJAP_DEF="WIZNETSZ","12345678"
	OK
	AT+CWQAP
	OK

### 3.2.8 AT+CWSAP\_CUR: Configures the WizFi360 SoftAP; Configuration Not Saved in the Flash

	Query command	Set Command
<b>Commands</b>	AT+CWSAP_CUR?	AT+CWSAP_CUR=<ssid>,<pwd>,<chl>,<ecn>[,<max conn>][,<ssid hidden>]
<b>Function</b>	To obtain the configuration Parameter of the WizFi360 SoftAP.	To configure the WizFi360 SoftAP
<b>Response</b>	+CWSAP_CUR:<ssid>,<pwd>,<chl>,<ecn>,<max conn>,<ssid hidden> OK	OK
<b>Default Value</b>	WizFi360_36036","",1,0,4,0	
<b>Parameter</b>	<ssid>: string parameter, SSID of AP. Length from 1 to 32 bytes of visible character. <pwd>: string parameter, length of password: 8 ~ 64 bytes ASCII. <chl>: channel ID. With range of [0,13] <ecn>: encryption method: • 0: OPEN • 2: WPA_PSK • 3: WPA2_PSK [<max conn>]: maximum number of Stations to which WizFi360 SoftAP can be connected; within the range of [1, 4]. [<ssid hidden>]: • 0: SSID is broadcasted. (factory default) • 1: SSID is not broadcasted.	
<b>Example</b>	AT+CWSAP_CUR?	AT+CWMODE_CUR=2
	+CWSAP_CUR="WizFi360","12345678",5,3,4,0	OK

	OK	AT+CWSAP_CUR="WizFi360","12345678",5,3,4,0
		OK
<b>Note</b>	<ul style="list-style-type: none"> <li>• The configuration changes will NOT be saved in the flash.</li> <li>• This command is only available in SoftAP mode and SoftAP+Station mode.</li> <li>• To use &lt;ssid hidden&gt;, &lt;max conn&gt; must be set first.</li> </ul>	

### 3.2.9 AT+CWSAP\_DEF: Configures the WizFi360 SoftAP; Configuration Saved in the Flash

	Query command	Set Command
<b>Commands</b>	AT+CWSAP_DEF?	AT+CWSAP_DEF=<ssid>,<pwd>,<chl>,<ecn>[,<max conn>][,<ssid hidden>]
<b>Function</b>	To obtain the configuration Parameter of the WizFi360 SoftAP.	To configure the WizFi360 SoftAP
<b>Response</b>	+CWSAP_DEF:<ssid>,<pwd>,<chl>,<ecn>,<max conn>,<ssid hidden>  OK	OK
<b>Default Value</b>	WizFi360_36036","",1,0,4,0	
<b>Parameter</b>	<p>&lt;ssid&gt;: string parameter, SSID of AP. Length from 1 to 32 bytes of visible character.          &lt;pwd&gt;: string parameter, length of password: 8 ~ 64 bytes ASCII.          &lt;chl&gt;: channel ID. With range of [0,13]          &lt;ecn&gt;: encryption method:</p> <ul style="list-style-type: none"> <li>• 0: OPEN</li> <li>• 2: WPA_PSK</li> <li>• 3: WPA2_PSK</li> </ul> <p>[&lt;max conn&gt;]: maximum number of Stations to which WizFi360 SoftAP can be connected; within the range of [1, 4].          [&lt;ssid hidden&gt;]:</p> <ul style="list-style-type: none"> <li>• 0: SSID is broadcasted. (factory default)</li> <li>• 1: SSID is not broadcasted.</li> </ul>	
<b>Example</b>	AT+CWSAP_DEF?	AT+CWMODE_DEF=2
	+CWSAP_DEF="WizFi360","12345678",5,3,4,0	OK
	OK	AT+CWSAP_DEF="WizFi360","12345678",5,3,4,0
		OK
<b>Note</b>	<ul style="list-style-type: none"> <li>• The configuration changes will be saved in the flash system parameter area.</li> <li>• This command is only available in SoftAP mode and SoftAP+Station mode.</li> <li>• To use &lt;ssid hidden&gt;, &lt;max conn&gt; must be set first.</li> </ul>	



### 3.2.10 AT+CWLIF: IP of Stations which are connected to WizFi360 SoftAP

	Execute command
<b>Commands</b>	AT+CWLIF
<b>Response</b>	<ip>,<mac> OK
<b>Parameter</b>	<ip>: string parameter, IP address of Stations to which WizFi360 SoftAP is connected. <mac>: string parameter, MAC address of Stations to which WizFi360 SoftAP is connected.
<b>Example</b>	AT+CWMODE_DEF=2
	OK
	AT+CWSAP_DEF="WizFi360","12345678",1,2
	OK
	AT+CWLIF
	"192.168.4.2","18:cf:5e:c5:ce:76"
<b>Note</b>	<ul style="list-style-type: none"> <li>• This command cannot get a static IP.</li> <li>• It only works when both DHCPs of the WizFi360 SoftAP, and of the Station to which WizFi360 is connected, are enabled.</li> </ul>

### 3.2.11 AT+CWDHCP\_CUR: Enables/Disables DHCP; Configuration Not Saved in the Flash

	Query command	Set Command
<b>Commands</b>	AT+CWDHCP_CUR?	AT+CWDHCP_CUR=<mode>,<en>
<b>Function</b>	To check the DHCP status	To enable/disable DHCP.
<b>Response</b>	+CWDHCP_CUR:<para> OK	OK
<b>Parameter</b>	<para> <ul style="list-style-type: none"> <li>• 0: SoftAP DHCP and Station DHCP are disabled.</li> <li>• 1: SoftAP DHCP is enabled and Station DHCP is disabled.</li> <li>• 2: SoftAP DHCP is disabled and Station DHCP is enabled.</li> <li>• 3: SoftAP DHCP and Station DHCP are enabled. (factory default)</li> </ul>	<mode> <ul style="list-style-type: none"> <li>• 0: Sets WizFi360 SoftAP</li> <li>• 1: Sets WizFi360 Station</li> <li>• 2: Sets both SoftAP and Station</li> </ul> <en> <ul style="list-style-type: none"> <li>• 0: Disables DHCP</li> <li>• 1: Enables DHCP</li> </ul>

Example	AT+CWDHCP_CUR?	AT+CWDHCP_CUR=1,1
	+CWDHCP_CUR:1	OK
	OK	
Note	<ul style="list-style-type: none"><li>• The configuration changes will NOT be saved in the flash.</li><li>• This Set Command interacts with static-IP-related AT commands (AT+CIPSTA-related and AT+CIPAP-related commands):<ul style="list-style-type: none"><li>• If DHCP is enabled, static IP will be disabled.</li><li>• If static IP is enabled, DHCP will be disabled.</li></ul></li><li>• Whether it is DHCP or static IP that is enabled depends on the last configuration.</li></ul>	

### 3.2.12 AT+CWDHCP\_DEF: Enables/Disables DHCP; Configuration Saved in the Flash

	Query command	Set Command
Commands	AT+CWDHCP_DEF?	AT+CWDHCP_DEF=<mode>,<en>
Function	To check the DHCP status	To enable/disable DHCP.
Response	+CWDHCP_DEF:<para>  OK	OK
Parameter	<para> <ul style="list-style-type: none"><li>• 0: SoftAP DHCP and Station DHCP are disabled.</li><li>• 1: SoftAP DHCP is enabled and Station DHCP is disabled.</li><li>• 2: SoftAP DHCP is disabled and Station DHCP is enabled.</li><li>• 3: SoftAP DHCP and Station DHCP are enabled.</li></ul> Factory default: In v. 1.0.x.x: 0 In v. 1.1.x.x: 3	<mode> <ul style="list-style-type: none"><li>• 0: Sets WizFi360 SoftAP</li><li>• 1: Sets WizFi360 Station</li><li>• 2: Sets both SoftAP and Station</li></ul> <en> <ul style="list-style-type: none"><li>• 0: Disables DHCP</li><li>• 1: Enables DHCP</li></ul>
Example	AT+CWDHCP_DEF?	AT+CWDHCP_DEF=1,1
	+CWDHCP_DEF:1  OK	OK
Note	<ul style="list-style-type: none"><li>• The configuration changes will be saved in the system parameter area in the flash.</li><li>• This Set Command interacts with static-IP-related AT commands (AT+CIPSTA-related and AT+CIPAP-related commands):</li><li>• If DHCP is enabled, static IP will be disabled.</li><li>• If static IP is enabled, DHCP will be disabled.</li><li>• Whether it is DHCP or static IP that is enabled depends on the last configuration.</li></ul>	

### 3.2.13 AT+CWDHCP\_CUR: Sets the IP Address Allocated by WizFi360 SoftAP DHCP; Configuration Not Saved in Flash

	Query command	Set Command
<b>Commands</b>	AT+CWDHCP_CUR?	AT+CWDHCP_CUR=<enable>[,<lease time>,<start IP>,<end IP>]
<b>Function</b>	To obtain the IP address range of the WizFi360 SoftAP DHCP Server.	Sets the IP address range of the WizFi360 SoftAP DHCP server
<b>Response</b>	+CWDHCP_CUR: <lease time>,<start IP>,<end IP> OK	OK
<b>Default Value</b>	120,"192.168.36.2","192.168.36.101"	
<b>Parameter</b>	<enable>: • 0: Disable the settings and use the default IP range. • 1: Enable setting the IP range, and the Parameter below have to be set. [<lease time>]: lease time; unit: minute; range [1, 2880]. [<start IP>]: string parameter, start IP of the IP range that can be obtained from WizFi360 SoftAP DHCP server. [<end IP>]: string parameter, end IP of the IP range that can be obtained from WizFi360 SoftAP DHCP server.	
<b>Example1</b>	AT+CWDHCP_CUR?	AT+CWDHCP_CUR=0,1
	+CWDHCP_CUR:120,"192.168.0.2","192.168.0.101"	OK
	OK	AT+CWDHCP_CUR=1,120,"192.168.0.100","192.168.0.200"
		OK
<b>Example2</b>	-	AT+CWDHCP_CUR=0
		OK
<b>Note</b>	<ul style="list-style-type: none"> <li>The configuration changes will NOT be saved in the flash.</li> <li>This AT command is enabled when WizFi360 runs as SoftAP, and when DHCP is enabled.</li> <li>The IP address should be in the same network segment as the IP address of SoftAP.</li> <li>To use &lt;lease time&gt;, &lt;start IP&gt; and &lt;end IP&gt; must be set. Likewise, to use &lt;start IP&gt; and &lt;end IP&gt;, &lt;lease time&gt; must be set.</li> </ul>	

### 3.2.14 AT+CWDHCP\_DEF: Sets the IP Address Allocated by WizFi360 SoftAP DHCP; Configuration Saved in Flash

	Query command	Set Command
<b>Commands</b>	AT+CWDHCP_DEF?	AT+CWDHCP_DEF=<enable>[,<lease time>,<start IP>,<end IP>]

<b>Function</b>	To obtain the IP address range of the WizFi360 SoftAP.	Sets the IP address range of the WizFi360 SoftAP DHCP server
<b>Response</b>	+CWDHCPS_DEF: <lease time>,<start IP>,<end IP>  OK	OK
<b>Default Value</b>	120,"192.168.36.2","192.168.36.101"	
<b>Parameter</b>	<enable>: • 0: Disable the settings and use the default IP range. • 1: Enable setting the IP range, and the Parameter below have to be set. [<lease time>]: lease time; unit: minute; range [1, 2880]. [<start IP>]: string parameter, start IP of the IP range that can be obtained from WizFi360 SoftAP DHCP server. [<end IP>]: string parameter, end IP of the IP range that can be obtained from WizFi360 SoftAP DHCP server.  Factory Default: In v. 1.0.x.x: [<start IP>] and [<end IP>] are 0.0.0.0 In v. 1.1.x.x: [<start IP>] is 192.168.36.2 and [<end IP>] is 192.168.36.101	
<b>Example1</b>	AT+CWDHCPS_DEF?	AT+CWDHCP_DEF=0,1
	+CWDHCPS_DEF:120,"192.168.0.2","192.168.0.101"	OK
	OK	AT+CWDHCPS_DEF=1,120,"192.168.0.100","192.168.0.200"
		OK
<b>Example2</b>	-	AT+CWDHCPS_DEF=0
		OK
<b>Note</b>	<ul style="list-style-type: none"> <li>• The configuration changes will be saved in the flash system parameter area.</li> <li>• This AT command is enabled when WizFi360 runs as SoftAP, and when DHCP is enabled.</li> <li>• The IP address should be in the same network segment as the IP address of SoftAP.</li> <li>• To use &lt;lease time&gt;, &lt;start IP&gt; and &lt;end IP&gt; must be set. Likewise, to use &lt;start IP&gt; and &lt;end IP&gt;, &lt;lease time&gt; must be set.</li> </ul>	

### 3.2.15 AT+CWAUTOCONN: Auto-Connects to the AP or Not

	Query Command	Execute Command
<b>Commands</b>	AT+CWAUTOCONN?	AT+CWAUTOCONN=<enable>
<b>Function</b>	Obtain setting for auto-connection	Set auto-connection
<b>Response</b>	+CWAUTOCONN:1  OK	OK

<b>Parameter</b>	<enable>: • 0: does NOT auto-connect to AP on power-up. • 1: connects to AP automatically on power-up. (factory default)	
<b>Example</b>	AT+CWAUTOCONN?	AT+CWJAP_DEF="WIZNETSZ","12345678"
	+CWAUTOCONN:1	OK
	OK	
		AT+CWAUTOCONN=1
<b>Note</b>		OK
	• The configuration changes will be saved in the user parameter area in the flash. • This command is only available in Station mode and SoftAP+Station mode.	

### 3.2.16 AT+CIPSTAMAC\_CUR: Sets the MAC Address of the WizFi360 Station; Configuration Not Saved in the Flash

	Query command	Set Command
Commands	AT+CIPSTAMAC_CUR?	AT+CIPSTAMAC_CUR=<mac>
Function	Obtain the MAC address of the WizFi360 Station.	Set the MAC address of the WizFi360 Station.
Response	+CIPSTAMAC_CUR:<mac>  OK	OK
Parameter	<mac>: string parameter, MAC address of the WizFi360 Station.	
Example	AT+CIPSTAMAC_CUR?	AT+CIPSTAMAC_CUR="00:08:DC:11:12:13"
	+CIPSTAMAC_CUR:"00:08:dc:11:12:13"	OK
	OK	
Note	<ul style="list-style-type: none"><li>• The configuration changes will NOT be saved in the flash.</li><li>• The MAC address of WizFi360 SoftAP is different from that of the WizFi360 Station.</li><li>• e.g. If the MAC address in Station mode is "00:08:DC:11:12:13"; the MAC address in SoftAP mode is "02:08:DC:11:12:13".</li><li>• Bit 0 of the WizFi360 MAC address CANNOT be 1. For example, a MAC address can be "00:..." but not "01:..."</li></ul>	

### 3.2.17 AT+CIPSTAMAC\_DEF: Sets the MAC Address of the WizFi360 Station; Configuration Saved in the Flash

	Query command	Set Command
<b>Commands</b>	AT+CIPSTAMAC_DEF?	AT+CIPSTAMAC_DEF=<mac>
<b>Function</b>	Obtain the MAC address of the WizFi360 Station.	Set the MAC address of the WizFi360 Station.

Response	+CIPSTAMAC_DEF:<mac>  OK	OK
Parameter	<mac>: string parameter, MAC address of the WizFi360 Station.	
Example	AT+CIPSTAMAC_DEF?	AT+CIPSTAMAC_DEF="00:08:DC:11:12:13"
	+CIPSTAMAC_DEF:"00:08:dc:11:12:13"	OK
	OK	
Note	<ul style="list-style-type: none"><li>• The configuration changes will be saved in the user parameter area in the flash.</li><li>• The MAC address of WizFi360 SoftAP is different from that of the WizFi360 Station.</li><li>• e.g. If the MAC address in Station mode is "00:08:DC:11:12:13"; the MAC address in SoftAP mode is "02:08:DC:11:12:13".</li><li>• Bit 0 of the WizFi360 MAC address CANNOT be 1. For example, a MAC address can be "00:..." but not "01:..."</li></ul>	

### 3.2.18 AT+CIPAPMAC\_CUR: Sets the MAC Address of the WizFi360 SoftAP; Configuration Not Saved in the Flash

	Query command	Set Command
Commands	AT+CIPAPMAC_CUR?	AT+CIPAPMAC_CUR=<mac>
Function	Obtain the MAC address of the WizFi360 SoftAP.	Set the MAC address of the WizFi360 SoftAP.
Response	+CIPAPMAC_CUR:<mac>  OK	OK
Parameter	<mac>: string parameter, MAC address of the WizFi360 SoftAP.	
Example	AT+CIPAPMAC_CUR?	AT+CIPAPMAC_CUR="02:08:DC:11:12:13"
	+CIPAPMAC_CUR:"02:08:dc:11:12:13"	OK
	OK	
Note	<ul style="list-style-type: none"><li>• The configuration changes are not allowed.</li><li>• The MAC address of WizFi360 SoftAP depends on MAC address of the WizFi360 Station at boot time.</li><li>• e.g. If the MAC address in Station mode is "00:08:DC:11:12:13"; the MAC address in SoftAP mode is "02:08:DC:11:12:13".</li></ul>	

### 3.2.19 AT+CIPAPMAC\_DEF: Sets the MAC Address of the WizFi360 SoftAP; Configuration Saved in the Flash

	Query command	Set Command
<b>Commands</b>	AT+CIPAPMAC_DEF?	AT+CIPAPMAC_DEF=<mac>

Function	Obtain the MAC address of the WizFi360 SoftAP.	Set the MAC address of the WizFi360 SoftAP.
Response	+CIPAPMAC_DEF:<mac>  OK	OK
Parameter	<mac>: string parameter, MAC address of the WizFi360 SoftAP.	
Example	AT+CIPAPMAC_DEF?	AT+CIPAPMAC_DEF="02:08:DC:11:12:13"
	+CIPAPMAC_DEF:"02:08:dc:11:12:13"  OK	OK
Note	<ul style="list-style-type: none"><li>• The configuration changes are not allowed.</li><li>• The MAC address of WizFi360 SoftAP depends on MAC address of the WizFi360 Station at boot time.</li><li>• e.g. If the MAC address in Station mode is "00:08:DC:11:12:13"; the MAC address in SoftAP mode is "02:08:DC:11:12:13".</li></ul>	

### 3.2.20 AT+CIPSTA\_CUR: Sets the Current IP Address of the WizFi360 Station; Configuration Not Saved in the Flash

	Query command	Set Command
Commands	AT+CIPSTA_CUR?	AT+CIPSTA_CUR=<ip>[,<gateway>][,<netmask>]
Function	To obtain the current IP address of the WizFi360 Station	To set the current IP address of the WizFi360 Station
Response	+CIPSTA_CUR:ip:<ip> +CIPSTA_CUR:gateway:<gateway> +CIPSTA_CUR:netmask:<netmask>  OK	OK
Parameter	<ip>: string parameter, the IP address of the WizFi360 Station [<gateway>]: string parameter, the gateway of the WizFi360 Station [<netmask>]: string parameter, the netmask of the WizFi360 Station	
Example	AT+CIPSTA_CUR?	AT+CIPSTA_CUR="192.168.1.88","192.168.1.1","255.255.255.0"
	+CIPSTA_CUR:ip:"192.168.1.88" +CIPSTA_CUR:gateway:"192.168.1.1" +CIPSTA_CUR:netmask:"255.255.255.0"  OK	OK
Note	<ul style="list-style-type: none"><li>• The configuration changes will NOT be saved in the flash.</li><li>• The Set Command interacts with DHCP-related AT commands (AT+CWDHCP-related commands):</li><li>• If static IP is enabled, DHCP will be disabled.</li><li>• If DHCP is enabled, static IP will be disabled.</li><li>• Whether it is DHCP or static IP that is enabled depends on the last configuration.</li><li>• To use &lt;gateway&gt;, &lt;netmask&gt; must be set first.</li></ul>	

### 3.2.21 AT+CIPSTA\_DEF: Set the static IP of WizFi360 Station, Saved to Flash

	Query command	Set Command
<b>Commands</b>	AT+CIPSTA_DEF?	AT+CIPSTA_DEF=<ip>[,<gateway>][,<netmask>]
<b>Function</b>	To obtain the default IP address of the WizFi360 Station	To set the default IP address of the WizFi360 Station
<b>Response</b>	+CIPSTA_DEF:ip:<ip> +CIPSTA_DEF:gateway:<gateway> +CIPSTA_DEF:netmask:<netmask>  OK	OK
<b>Parameter</b>	<ip>: string parameter, the IP address of the WizFi360 Station [<gateway>]: string parameter, the gateway of the WizFi360 Station [<netmask>]: string parameter, the netmask of the WizFi360 Station  Factory default values: In v. 1.0.x.x: <ip>: 0.0.0.0 [<gateway>]: 0.0.0.0 [<netmask>]: 0.0.0.0  In v. 1.1.x.x: <ip>: 192.168.1.1 [<gateway>]: 192.168.1.1 [<netmask>]: 255.255.255.0	
<b>Example</b>	AT+CIPSTA_DEF?	AT+CIPSTA_DEF="192.168.1.88","192.168.1.1","255.255.255.0"
	+CIPSTA_CUR:ip:"192.168.1.88" +CIPSTA_CUR:gateway:"192.168.1.1" +CIPSTA_CUR:netmask:"255.255.255.0"  OK	OK
<b>Note</b>	<ul style="list-style-type: none"> <li>• The configuration changes will be saved in the user parameter area in the flash.</li> <li>• The Set Command interacts with DHCP-related AT commands (AT+CWDHCP-related commands):</li> <li>• If static IP is enabled, DHCP will be disabled.</li> <li>• If DHCP is enabled, static IP will be disabled.</li> <li>• Whether it is DHCP or static IP that is enabled depends on the last configuration.</li> <li>• To use &lt;gateway&gt;, &lt;netmask&gt; must be set first.</li> </ul>	

### 3.2.22 AT+CIPAP\_CUR: Sets the IP Address of the WizFi360 SoftAP; Configuration Not Saved in the Flash

	Query command	Set Command
<b>Commands</b>	AT+CIPAP_CUR?	AT+CIPAP_CUR=<ip>[,<gateway>][,<netmask>]



<b>Function</b>	To obtain the current IP address of the WizFi360 SoftAP.	To set the current IP address of the WizFi360 SoftAP.
<b>Response</b>	+CIPAP_CUR:ip:<ip> +CIPAP_CUR:gateway:<gateway> +CIPAP_CUR:netmask:<netmask>  OK	OK
<b>Parameter</b>	<ip>: string parameter, the IP address of the WizFi360 SoftAP. [<gateway>]: string parameter, the gateway of the WizFi360 SoftAP. [<netmask>]: string parameter, the netmask of the WizFi360 SoftAP.	
<b>Example</b>	AT+CIPAP_CUR?	AT+CIPAP_CUR="192.168.0.1","192.168.0.1","255.255.255.0"
	+CIPAP_CUR:ip:"192.168.0.1" +CIPAP_CUR:gateway:"192.168.0.1" +CIPAP_CUR:netmask:"255.255.255.0"	OK
	OK	AT+CIPAP_CUR="192.168.0.1"
		OK
<b>Note</b>	<ul style="list-style-type: none"> <li>• The configuration changes will NOT be saved in the flash.</li> <li>• Currently, WizFi360 only supports class C IP addresses.</li> <li>• The Set Command interacts with DHCP-related AT commands (AT+CWDHCP-related commands):</li> <li>• If static IP is enabled, DHCP will be disabled.</li> <li>• If DHCP is enabled, static IP will be disabled.</li> <li>• Whether it is DHCP or static IP that is enabled depends on the last configuration.</li> <li>• To use &lt;gateway&gt;, &lt;netmask&gt; must be set first.</li> </ul>	

### 3.2.23 AT+CIPAP\_DEF: Sets the IP Address of the WizFi360 SoftAP; Configuration Saved in the Flash

	Query command	Set Command
<b>Commands</b>	AT+CIPAP_DEF?	AT+CIPAP_DEF=<ip>[,<gateway>][,<netmask>]
<b>Function</b>	To obtain the default IP address of the WizFi360 SoftAP.	To set the default IP address of the WizFi360 SoftAP.
<b>Response</b>	+CIPAP_DEF:ip:<ip> +CIPAP_DEF:gateway:<gateway> +CIPAP_DEF:netmask:<netmask>  OK	OK
<b>Parameter</b>	<ip>: string parameter, the IP address of the WizFi360 SoftAP. [<gateway>]: string parameter, the gateway of the WizFi360 SoftAP. [<netmask>]: string parameter, the netmask of the WizFi360 SoftAP.  Factory default values: In v. 1.0.x.x: <ip>: 0.0.0.0	

	[<gateway>]:0.0.0.0 [<netmask>]:0.0.0.0  In v. 1.1.x.x: <ip>: 192.168.36.1 [<gateway>]:192.168.36.1 [<netmask>]:255.255.255.0	
<b>Example</b>	AT+CIPAP_DEF?	AT+CIPAP_DEF="192.168.0.1","192.168.0.1","255.255.255.0"
	+CIPAP_DEF:ip:"192.168.0.1"	OK
	+CIPAP_DEF:gateway:"192.168.0.1"	AT+CIPAP_DEF="192.168.0.1"
	+CIPAP_DEF:netmask:"255.255.255.0"	OK
<b>Note</b>	<ul style="list-style-type: none"> <li>• The configuration changes will be saved in the user parameter area in the flash.</li> <li>• Currently, WizFi360 only supports class C IP addresses.</li> <li>• The Set Command interacts with DHCP-related AT commands (AT+CWDHCP-related commands):</li> <li>• If static IP is enabled, DHCP will be disabled.</li> <li>• If DHCP is enabled, static IP will be disabled.</li> <li>• Whether it is DHCP or static IP that is enabled depends on the last configuration.</li> <li>• To use &lt;gateway&gt;, &lt;netmask&gt; must be set first.</li> </ul>	

### 3.2.24 AT+CWSTARTSMART: Start SmartConfig<sup>1</sup>

	Execute command	Set command
<b>Commands</b>	AT+CWSTARTSMART	In v. 1.0.x.x <sup>2</sup> : AT+CWSTARTSMART=<type>, <try count>  In v. 1.1.x.x <sup>3</sup> : AT+CWSTARTSMART=<try count>
<b>Function</b>	To start SmartConfig of ESP-TOUCH + AirKiss.	To start SmartConfig of a designated type.
<b>Response</b>	OK	
<b>Parameter</b>	-	<type>: Start the Smart Config to a configured type <ul style="list-style-type: none"> <li>• 1: ESP-TOUCH</li> <li>• 2: AirKiss</li> <li>• 3: ESP-TOUCH + AirKiss</li> </ul>

<sup>1</sup> In some APs, it may fail to connect Airkiss. We recommend using Airkiss with WebServer..

<sup>2</sup> From Version 1.0.7.2

<sup>3</sup> Supports only Airkiss

		<try count> set number of trials to setup via SmartConfig.
<b>Messages</b>	After smartconfig start and connect to the AP, it will return as below	
	smartconfig type:<type> smart get WiFi info ssid:<ssid> password:<password> smartconfig connected WiFi	
<b>Parameter</b>	<type>: AIRKISS or ESPTOUCH <ssid>: AP's ssid <password>: AP's password	
<b>Example</b>	AT+CWMODE_DEF=1	
	OK	
	AT+CWSTARTSMART	
	OK	
	(After smartconfig start and connect to the AP, it will return as below)	
	Smartconfig type:AIRKISS smart get WiFi info ssid:Wiznet_AP password:1234567890 smartconfig connected WiFi	
<b>Note</b>	<ul style="list-style-type: none"> <li>• SmartConfig is only available in the Station mode. (AT+CWMODE_CUR=1)</li> <li>• The message "smart get wifi info" means that SmartConfig has successfully acquired the AP information. WizFi360 will try to connect to the target AP.</li> <li>• The message "smartconfig connected wifi" is printed if the connection is successful. Use command AT+CWSTOPSMART to stop SmartConfig before running other commands. Please make sure that you do not execute other commands during SmartConfig.</li> <li>• The LED(PB_07) is blinking fast during running Oneshot via Airkiss.</li> </ul> <p>The LED(PB_07) is blinking slowly during running Oneshot via WebServer. • SmartConfig operation process is as following.</p> <ol style="list-style-type: none"> <li>1. Set the WizFi360 to Station mode and start smartconfig.</li> <li>2. Connect to the AP on smartphone. Number of trials is set by &lt;try count&gt;. If not set, unlimited trials.</li> <li>3. Open the ESP-TOUCH APP or AirKiss on WeChat APP.</li> <li>4. Set the AP's ssid and password on the APP and check connection to the AP on WizFi360.</li> <li>5. After fail to connect via ESP-TOUCH/Airkiss, WebServer(Default 192.168.36.1) will be started<sup>4</sup></li> </ol>	

<sup>4</sup> If <try count> is not set, number of trials will be unlimited

### 3.2.25 AT+CWSTOPSMART: Stop Smart Config

	Execute command
<b>Commands</b>	AT+CWSTOPSMART
<b>Response</b>	OK
<b>Example</b>	AT+CWSTOPSMART
	OK
<b>Note</b>	<ul style="list-style-type: none"> <li>No matter what of whether SmartConfig succeeds or not, please always call AT+CWSTOPSMART before any other AT commands.</li> </ul>

### 3.2.26 AT+WPS: Enables the WPS Function

	Execute command
<b>Commands</b>	AT+WPS=<enable>
<b>Response</b>	wps started OK
<b>Parameter</b>	<enable>: <ul style="list-style-type: none"> <li>1: enables WPS/WiFi Protected Setup</li> <li>0: disables WPS</li> </ul>
<b>Example</b>	AT+WPS=1
	wps started
	OK
<b>Note</b>	<ul style="list-style-type: none"> <li>WPS must be used in Station mode and SoftAP+Station mode.</li> <li>WPS does not support WEP/Wired-Equivalent Privacy encryption</li> </ul>

### 3.2.27 AT+CWHOSTNAME: Configures the Name of WizFi360 Station

	Query Command	Set Command
<b>Commands</b>	AT+CWHOSTNAME?	AT+CWHOSTNAME=<hostname>
<b>Function</b>	Checks the host name of WizFi360 Station	Sets the host name of WizFi360 Station
<b>Response</b>	+CWHOSTNAME:<host name>	OK
	OK	
	<i>(Station mode disabled)</i> +CWHOSTNAME:<NULL>	

	OK	
Parameter	<hostname>: string parameter, the host name of the WizFi360 Station, the maximum length is 32 bytes.	
Example	AT+CWHOSTNAME?	AT+CWHOSTNAME="WizFi360_FF6179"
	+CWHOSTNAME:WizFi360_FF6179	OK
	OK	
Note	• The configuration changes will NOT be saved in the flash.	

### 3.2.28 AT+CWCOUNTRY\_CUR: Set WiFi Country Code of WizFi360; Configuration Not Saved in the Flash

	Query Command	Set Command
<b>Commands</b>	AT+CWCOUNTRY_CUR?	AT+CWCOUNTRY_CUR=<policy>,<country_code>,<channel_option>
<b>Function</b>	Check the actual value of WiFi country code, which may be changed to the same as the AP it connected to.	Set the current WiFi Country code of WizFi360
<b>Response</b>	+CWCOUNTRY_CUR:<policy>,<country_code>,<channel_option> OK	OK
<b>Parameter</b>	<p>&lt;policy&gt;:</p> <ul style="list-style-type: none"> <li>0: will change the county code to be the same as the AP that WizFi360 is connected to</li> <li>1: the country code will not change, always be the one set by command</li> </ul> <p>&lt;country_code&gt;: string parameter, country code, the length can be 3 characters at most; but the third one is a special character which will not be shown after AT+CWCOUNTRY_CUR? command.</p> <p>&lt;channel_option&gt;:</p> <ul style="list-style-type: none"> <li>0: select the channel to 1~11</li> <li>1: select the channel to 1~13</li> <li>2: select the channel to 10~11</li> <li>3: select the channel to 10~13</li> <li>4: select the channel to 14</li> <li>5: select the channel to 1~14</li> <li>6: select the channel to 3~9</li> <li>7: select the channel to 5~13</li> </ul>	

Example	AT+CWOUNTRY_CUR?	AT+CWOUNTRY_CUR=1,"KR",1
	+CWOUNTRY_CUR=1,"KR",1	OK
	OK	
Note	• The configuration changes will NOT be saved in the flash.	

### 3.2.29 AT+CWOUNTRY\_DEF: Set WiFi Country Code of WizFi360; Configuration Saved in the Flash

	Query Command	Set Command
<b>Commands</b>	AT+CWOUNTRY_DEF?	AT+CWOUNTRY_DEF=<policy>,<country_code>,<channel_option>
<b>Function</b>	Check the default WiFi country code which is stored in the flash.	Set the default WiFi Country code of WizFi360, and save in the flash.
<b>Response</b>	+CWOUNTRY_DEF:<policy>,<country_code>,<channel_option> OK	OK
<b>Parameter</b>	<p>&lt;policy&gt;:</p> <ul style="list-style-type: none"> <li>0: will change the county code to be the same as the AP that WizFi360 is connected to</li> <li>1: the country code will not change, always be the one set by command</li> </ul> <p>&lt;country_code&gt;: string parameter, country code, the length can be 3 characters at most; but the third one is a special character which will not be shown after AT+CWOUNTRY_DEF? command.</p> <p>&lt;channel_option&gt;:</p> <ul style="list-style-type: none"> <li>0: select the channel to 1~11</li> <li>1: select the channel to 1~13</li> <li>2: select the channel to 10~11</li> <li>3: select the channel to 10~13</li> <li>4: select the channel to 14</li> <li>5: select the channel to 1~14</li> <li>6: select the channel to 3~9</li> <li>7: select the channel to 5~13</li> </ul>	
<b>Example</b>	AT+CWOUNTRY_DEF?	AT+CWOUNTRY_DEF=1,"KR",1
	+CWOUNTRY_DEF=1,"KR",1	OK
	OK	

**Note**

- The configuration changes will be saved in the flash user parameter area.

### 3.2.30 AT+WIZ\_NETCONFIG: WebServer for setting SSID/PWD

	Query Command
<b>Commands</b>	AT+WIZ_NETCONFIG
<b>Function</b>	Start WebServer for setting SSID/PWD.
<b>Response</b>	OK  smartconfig type:Web Smart get wifi info ssid:<AP SSID> password:<password> smartconfig connected wifi
<b>Parameter</b>	N/A
<b>Example</b>	AT+WIZ_NETCONFIG  OK  smartconfig type:Web Smart get wifi info ssid:<AP SSID> password:<password> smartconfig connected wifi
<b>Note</b>	<ul style="list-style-type: none"> <li>• Module should be in SoftAP or Station+SoftAP mode in order to perform this command</li> <li>• The LED(PB_07) is blinking slowly during running Oneshot via WebServer.</li> <li>• Server will wait for connection unless AT+CWSTOPSMART is executed</li> <li>• WebServer setting process is as following:               <ol style="list-style-type: none"> <li>1. Set WizFi360 to SoftAP or Station+SoftAP mode. Run WebServer</li> <li>2. Connect via smartphone or PC to AP created by WizFi360 (AP will be unsecured. If MAC is 00:08:DC:11:22:33, SSID will be "WizFi360_112233" )</li> <li>3. Access to AP via WebBrowser. If no prior settings were made, AP IP is 192.168.36.1</li> <li>4. Enter SSID and password</li> <li>5. WizFi360 switches to Station mode and connects to router using entered credentials</li> </ol> </li> </ul>

## 3.3 TCP / IP command

### 3.3.1 AT+CIPSTATUS: Gets the Connection Status

	Set command
<b>Commands</b>	AT+CIPSTATUS
<b>Response</b>	STATUS:<stat> +CIPSTATUS:<link ID>,<type>,<remote IP>,<remote port>,<local port>,<tcp type>  OK
<b>Parameter</b>	<stat>: status of the WizFi360 Station interface <ul style="list-style-type: none"> <li>• 2: The WizFi360 Station is connected to an AP and its IP is obtained</li> <li>• 3: The WizFi360 Station has created a TCP or UDP transmission</li> <li>• 4: The TCP or UDP transmission of WizFi360 Station is disconnected</li> <li>• 5: The WizFi360 Station does NOT connect to an AP</li> </ul> <link ID>: ID of network connection (0~4), used for multiple connections (AT+CIPMUX=1) <type>: string parameter, TCP or UDP <remote IP>: string parameter indicating the remote IP address <remote port>: the remote port number <local port>: WizFi360 local port number <tcp type> <ul style="list-style-type: none"> <li>• 0: WizFi360 runs as a client</li> <li>• 1: WizFi360 runs as a server</li> </ul>
<b>Example1</b>	AT+CIPSTATUS
	STATUS:2  OK
<b>Example2</b>	AT+CIPSTATUS
	STATUS:3 +CIPSTATUS:1,"TCP","192.168.4.2",5000,6000,1  OK

### 3.3.2 AT+CIPDOMAIN: DNS Function

	Execute Command
<b>Commands</b>	AT+CIPDOMAIN=<domain name>



Response	+CIPDOMAIN:<ip address>
	OK
	DNS Fail
	ERROR
Parameter	<domain name>: string parameter, the domain name, length should be less than 64 bytes <ip address>: string parameter, IP address corresponding to the domain name
Example	AT+CIPDOMAIN="www.wiznet.io"
	+CIPDOMAIN:"183.111.174.49"
	OK

### 3.3.3 AT+CIPSTART: Establishes TCP Connection, UDP Transmission or SSL Connection

#### Establish TCP Connection

	Single TCP connection (AT+CIPMUX=0)	Multiple TCP connections (AT+CIPMUX=1)
Commands	AT+CIPSTART=<type>,<remote IP>,<remote port>[,<TCP keep alive>]	AT+CIPSTART=<link ID>,<type>,<remote IP>,<remote port>[,<TCP keep alive>]
Response	OK	
	ERROR	
	ALREADY CONNECTED // If the TCP transmission is already established	
Parameter	<link ID>: ID of network connection (0~4), used for multiple connections (AT+CIPMUX=1) <type>: string parameter indicating the connection type: TCP, UDP, SSL; This is TCP in this case <remote IP>: string parameter indicating the remote IP address <remote port>: the remote port number, range [1, 65535] [<TCP keep alive>]: detection time interval when TCP is kept alive; this function is disabled by default <ul style="list-style-type: none"> <li>• 0: disable TCP keep-alive</li> <li>• 1 ~ 7200: detection time interval; unit: second (s)</li> </ul>	
Example	AT+CIPSTART="TCP","192.168.1.99",5000	AT+CIPSTART=1,"TCP","www.iwiznet.cn",5000,10
	OK	OK

#### Establish UDP Connection

	Single UDP connection (AT+CIPMUX=0)	Multiple UDP connections (AT+CIPMUX=1)
--	-------------------------------------	--

<b>Commands</b>	AT+CIPSTART=<type>,<remote IP>,<remote port>[,<UDP local port>][,<UDP mode>]	AT+CIPSTART=<link ID>,<type>,<remote IP>,<remote port>[,<UDP local port>][,<UDP mode>]
<b>Response</b>	OK	
	ERROR	
	ALREADY CONNECTED // If the UDP connection is already established	
<b>Parameter</b>	<link ID>: ID of network connection (0~4), used for multiple connections (AT+CIPMUX=1) <type>: string parameter indicating the connection type: TCP, UDP, SSL; This is UDP in this case <remote IP>: string parameter indicating the remote IP address <remote port>: the remote port number within the range of 0~65535. If set to 0, it is assigned random value. [<UDP local port>]: UDP port number within the range of 1~65535. of WizFi360 [<UDP mode>]: In the UDP transparent transmission, the value of this parameter must be 0 <ul style="list-style-type: none"> <li>• 0: the destination peer entity of UDP will not change (default)</li> <li>• 1: the destination peer entity of UDP can change once</li> <li>• 2: the destination peer entity of UDP is allowed to change</li> </ul>	
<b>Example</b>	AT+CIPSTART="UDP","192.168.1.99",5000	AT+CIPSTART=1,"UDP","www.iwiznet.cn",5000,6000,2
	OK	OK
<b>Note</b>	To use <UDP mode>, <UDP local port> must be set first.	

## Establish SSL Connection

	Single SSL connection (AT+CIPMUX=0)	Multiple SSL connections (AT+CIPMUX=1)
<b>Commands</b>	AT+CIPSTART=<type>,<remote IP>,<remote port>[,<TCP keep alive>]	AT+CIPSTART=<link ID>,<type>,<remote IP>,<remote port>[,<TCP keep alive>]
<b>Response</b>	OK	
	ERROR	
	ALREADY CONNECTED // If the TCP connection is already established	
<b>Parameter</b>	<link ID>: ID of network connection (0~4), used for multiple connections (AT+CIPMUX=1) <type>: string parameter indicating the connection type: TCP, UDP, SSL; This is SSL in this case <remote IP>: string parameter indicating the remote IP address <remote port>: the remote port number [<TCP keep alive>]: detection time interval when TCP is kept alive; this function is disabled by default <ul style="list-style-type: none"> <li>• 0: disable TCP keep-alive</li> <li>• 1 ~ 7200: detection time interval; unit: second (s)</li> </ul>	

<b>Example</b>	AT+CIPSTART="SSL","www.wiznet.io",443	AT+CIPSTART=1,"SSL","www.wiznet.io",443
	OK	OK
<b>Note</b>	<ul style="list-style-type: none"> <li>• WizFi360 can only set one SSL connection at most.</li> <li>• SSL connection does not support UART-WiFi passthrough mode (transparent transmission).</li> <li>• SSL connection needs a large amount of memory; otherwise, it may cause system reboot. The command AT+CIPSSLSIZE=&lt;size&gt; can be used to enlarge the SSL buffer size.</li> <li>•</li> </ul>	

### 3.3.4 AT+CIPSSLSIZE: Sets the Size of SSL Buffer

	Query command	Set command
<b>Commands</b>	AT+CIPSSLSIZE?	AT+CIPSSLSIZE=<size>
<b>Response</b>	+CIPSSLSIZE:<size>	OK
<b>Parameter</b>	<size>: the size of the SSL buffer; range of value: [2048, 4096]	
<b>Example</b>	AT+CIPSSLSIZE?	AT+CIPSSLSIZE=4096
	+CIPSSLSIZE:2048	OK

### 3.3.5 AT+CIPSSLCONF: Sets Configuration of WiFi360 SSL Client

	Query command	Set command
<b>Commands</b>	AT+CIPSSLCONF?	AT+ CIPSSLCONF =<SSL mode>
<b>Function</b>	Get Configuration of the WizFi360 SSL certificate option.	Set Configuration of the WizFi360 SSL certificate option.
<b>Response</b>	+CIPSSLCONF:<SSL mode> OK	OK
<b>Parameter</b>	<SSL mode>: <ul style="list-style-type: none"> <li>• 0: WizFi360 not checks certificate of SSL Server.</li> <li>• 1: WizFi360 checks certificate of SSL Server. But WizFi360 connects to the server even if verification failed.</li> <li>• 2: WizFi360 checks certificate of SSL Server. If verification failed, WizFi360 doesn't connect to the server.</li> </ul>	
<b>Example</b>	AT+ CIPSSLCONF?	AT+ CIPSSLCONF =2
	+ CIPSSLCONF:2 OK	OK
<b>Note</b>	<ul style="list-style-type: none"> <li>• This configuration will be saved in the flash user parameter area.</li> <li>• If certificates need to be enabled, please call this command before SSL connection is established.</li> </ul>	

- Certificate is generated via AT+CASEND command.

### 3.3.6 AT+CASEND: Sets the SSL certificate

	Query command	Set command
<b>Commands</b>	AT+CASEND?	AT+CASEND=<mode>
<b>Function</b>	Get the SSL certificate.	Set the SSL certificate.
<b>Response</b>	-----BEGIN CERTIFICATE----- ... ... -----END CERTIFICATE-----	>
	ERROR	OK
<b>Parameter</b>		<mode>: <ul style="list-style-type: none"> <li>• 0: Delete a certificate</li> <li>• 1: Generate a certificate</li> </ul>
<b>Example1</b>	AT+CASEND?	AT+CASEND=1
	-----BEGIN CERTIFICATE----- ... ... -----END CERTIFICATE-----	>
		-----BEGIN CERTIFICATE----- ... ... -----END CERTIFICATE-----
		OK
<b>Example2</b>	AT+CASEND?	AT+CASEND=0
	ERROR	OK
<b>Note</b>	• Wrap return > after the set command. WizFi360 begins receiving serial data. User should input the data of certificate including lines "-----BEGIN CERTIFICATE-----" and -----END CERTIFICATE----- .	

### 3.3.7 AT+CIPSEND: Send data

#### UART-WiFi passthrough (transparent transmission) mode

	Execute command
<b>Commands</b>	AT+CIPSEND
<b>Function</b>	To start sending data in transparent transmission mode. Wrap return > after this command. Enter transparent transmission, with a 20-ms interval between each packet, and a maximum of 2048 bytes per packet.
<b>Response</b>	OK

	>
Example	AT+CIPSEND
	OK >
Note	<ul style="list-style-type: none"> <li>When a single packet containing +++ is received, WizFi360 returns to normal command mode. Please wait for at least one second before sending the next AT command.</li> <li>This command can only be used in transparent transmission mode which requires single connection.</li> <li>For UDP transparent transmission, the value of &lt;UDP mode&gt; has to be 0 in AT+CIPSTART.</li> </ul>

## Normal transmission mode

	Send in single connection (AT+CIPMUX=0)	Send in multiple connections (AT+CIPMUX=1)
Commands	AT+CIPSEND=<length>[,<remote IP>,<remote port>]	AT+CIPSEND=<link ID>,<length> [,<remote IP>,<remote port>]
Function	Send data of designated length in normal transmission mode. Wrap return > after the Set Command. WizFi360 begins receiving serial data. When data length defined by <length> is met, the transmission of data starts.	
Response	OK >	
	<i>If the connection cannot be established or gets disrupted during data transmission, the system returns:</i> ERROR	
	<i>If data is transmitted successfully, the system returns:</i> SEND OK	
	<i>If it failed, the system returns:</i> SEND FAIL	
Parameter	<link ID>: ID of network connection (0~4), used for multiple connections (AT+CIPMUX=1) <length>: data length, MAX: 2048 bytes. [<remote IP>]: string parameter, remote IP can be set in UDP transmission. [<remote port>]: remote port can be set in UDP transmission.	
Example	AT+CIPSEND=1220	AT+CIPSEND=0,1220,"192.168.0.10",50000
	OK >	OK >
Note	<ul style="list-style-type: none"> <li>In UDP transmission, WizFi360 sends data to destination set in &lt;remote IP&gt; and &lt;remote port&gt;. If you do not set the optional parameter, WizFi360 sends data to the destination set in AT + CIPSTART command.</li> <li>To use &lt;remote IP&gt;, &lt;remote port&gt; must be set. Likewise, to use &lt;remote port&gt;, &lt;remote IP&gt; must be set.</li> </ul>	

### 3.3.8 AT+CIPSENDEX: Sends data

	Send in single connection (AT+CIPMUX=0)	Send in multiple connections (AT+CIPMUX=1)
<b>Commands</b>	AT+CIPSENDEX=<length>[,<remote IP>,<remote port>]	AT+CIPSENDEX=<link ID>,<length> [,<remote IP>,<remote port>]
<b>Function</b>	Send data of designated length in normal transmission mode. Wrap return > after the Set Command. WizFi360 begins receiving serial data. When the requirement of data length, determined by <length>, is met, or when \0 appears in the data, the transmission starts.	
<b>Response</b>	OK >	
	<i>If the connection cannot be established or gets disrupted during data transmission, the system returns:</i> ERROR	
	<i>If data is transmitted successfully, the system returns:</i> SEND OK	
	<i>If it failed, the system returns:</i> SEND FAIL	
<b>Parameter</b>	<link ID>: ID of network connection (0~4), used for multiple connections (AT+CIPMUX=1) <length>: data length, MAX: 2048 bytes. [<remote IP>]: string parameter, remote IP can be set in UDP transmission. [<remote port>]: remote port can be set in UDP transmission.	
<b>Example</b>	AT+CIPSENDEX=1220	AT+CIPSENDEX=0,1220,"192.168.0.10",50000
	OK >	OK >
<b>Note</b>	<ul style="list-style-type: none"> <li>When the requirement of data length, determined by &lt;length&gt;, is met, or when \0 appears, the transmission of data starts. Go back to the normal command mode and wait for the next AT command.</li> <li>In UDP transmission, WizFi360 sends data to destination set in &lt;remote IP&gt; and &lt;remote port&gt;. If you do not set the optional parameter, WizFi360 sends data to the destination set in AT + CIPSTART command.</li> <li>To use &lt;remote IP&gt;, &lt;remote port&gt; must be set. Likewise, to use &lt;remote port&gt;, &lt;remote IP&gt; must be set.</li> </ul>	

### 3.3.9 AT+CIPSENDERBUF: Writes Data into the TCP-Send-Buffer

	single connection (AT+CIPMUX=0)	multiple connections (AT+CIPMUX=1)
<b>Commands</b>	AT+CIPSENDERBUF=<length>	AT+CIPSENDERBUF=<link ID>,<length>
<b>Function</b>	Wrap return > begins receiving serial data; when the length of data defined by the parameter <length> is met, the data is sent.	

Response	<current segment ID>,<segment ID of which sent successfully>	
	OK >	
	<i>If the data length over the value of &lt;length&gt;, the data will be discarded, the system returns:</i> busy	
	<i>If the connection cannot be established, or if it is not a TCP connection, or if the buffer is full, or some other error occurs, the system returns:</i> ERROR	
	<i>If data is transmitted successfully, the system returns:</i> <segment ID>,SEND OK	<i>If data is transmitted successfully, the system returns:</i> <link ID>,<segment ID>,SEND OK
<i>If it failed, the system returns:</i> SEND FAIL		
Parameter	<link ID>: ID of network connection (0~4), used for multiple connections (AT+CIPMUX=1) <segment ID>: uint32; the ID assigned to each data packet, starting from 1; the ID number increases by 1 every time a data packet is written into the buffer. <length>: data length, MAX: 2048 bytes.	
Example	AT+CIPSENDERBUF=1024	AT+CIPSENDERBUF=0,1024
	0	0,0
	OK >	OK >
Note	<ul style="list-style-type: none"> <li>• This command only writes data into the TCP-send-buffer, so it can be called continually, and the user need not wait for SEND OK; if a TCP segment is sent successfully, it will return &lt;segment ID&gt;,SEND OK.</li> <li>• Before data length reaches the value defined by &lt;length&gt;, input +++ can switch back from data mode to command mode, and discard the data received before.</li> <li>• This command can NOT be used for SSL connections.</li> </ul>	

### 3.3.10 AT+CIPBUFRESET: Resets the Segment ID Count

	single connection (AT+CIPMUX=0)	multiple connections (AT+CIPMUX=1)
Commands	AT+CIPBUFRESET	AT+CIPBUFRESET=<link ID>
Response	OK	
	<i>If the connection is not established or there is still TCP data waiting to be sent, the response will be:</i> ERROR	
Parameter	<link ID>: ID of network connection (0~4), used for multiple connections (AT+CIPMUX=1)	
Example	AT+CIPBUFRESET	AT+CIPBUFRESET=1
	OK	OK

**Note**

- This command can only be used after AT+CIPSENDERBUF command.

### 3.3.11 AT+CIPBUFSTATUS: Checks the Status of TCP-Send-Buffer

	single connection (AT+CIPMUX=0)	multiple connections (AT+CIPMUX=1)
<b>Commands</b>	AT+CIPBUFSTATUS	AT+CIPBUFSTATUS=<link ID>
<b>Response</b>	<next segment ID>,<segment ID sent>,<segment ID successfully sent>,<remain buffer size>,<queue number>  OK	
<b>Parameter</b>	<link ID>: ID of network connection (0~4), used for multiple connections (AT+CIPMUX=1) <next segment ID>: the next segment ID obtained by AT+CIPSENDERBUF <segment ID sent>: the ID of the TCP segment last sent <segment ID successfully sent>: the ID of the last successfully sent TCP segment <remain buffer size>: the remaining size of the TCP-send-buffer <queue number>: available TCP queue number; it's not reliable and should be used as a reference only.	
<b>Example</b>	AT+CIPBUFRESET	AT+CIPBUFRESET=1
	20,15,10,200,7	20,15,10,200,7
	OK	OK
<b>Example</b>	<ul style="list-style-type: none"> <li>• 20: means that the latest segment ID is 19; so when calling AT+CIPSENDERBUF the next time, the segment ID returned is 20</li> <li>• 15: means that the TCP segment with the ID 15 is the last segment sent, but the segment may not be successfully sent</li> <li>• 10: means that the TCP segment with the ID 10 was sent successfully</li> <li>• 200: means that the remaining size of the TCP-send-buffer is 200 bytes</li> <li>• 7: the available TCP queue number; it is not reliable and should be used as a reference only; when the queue number is 0, no TCP data can be sent.</li> </ul>	
<b>Note</b>	<ul style="list-style-type: none"> <li>• This command can not be used for SSL connection.</li> <li>• Only when &lt;next segment ID&gt; - &lt;segment ID sent&gt; = 1, can AT+CIPBUFRESET be called to reset the counting.</li> <li>• TCP buffer size is 21,900 bytes.</li> </ul>	

### 3.3.12 AT+CIPCHECKSEQ: Checks If a Specific Segment Was Successfully Sent

	single connection (AT+CIPMUX=0)	multiple connections (AT+CIPMUX=1)
<b>Commands</b>	AT+CIPCHECKSEQ=<segment ID>	AT+CIPCHECKSEQ=<link ID>,<segment ID>
<b>Response</b>	<segment ID>,<status>  OK	<link ID>,<segment ID>,<status>  OK



<b>Parameter</b>	<link ID>: ID of network connection (0~4), used for multiple connections (AT+CIPMUX=1) <segment ID>: the segment ID obtained by calling AT+CIPSENDBUF <status> • FALSE: the segment-sending failed • TRUE: the segment was sent successfully	
<b>Example</b>	AT+CIPCHECKSEQ=20	AT+CIPCHECKSEQ=1,20
	20,TRUE	1,20,TRUE
	OK	OK
<b>Note</b>	• This command can only be used after AT+CIPSENDBUF command.	

### 3.3.13 AT+CIPCLOSE: Closes the TCP/UDP/SSL Connection

	single connection (AT+CIPMUX=0)	multiple connections (AT+CIPMUX=1)
<b>Commands</b>	AT+CIPCLOSE	AT+CIPCLOSE=<link ID>
<b>Function</b>	To close the TCP/UDP Connection	
<b>Response</b>	OK	
<b>Parameter</b>	<link ID>: ID of network connection (0~4), used for multiple connections (AT+CIPMUX=1) When ID is 5, all connections will be closed. (In server mode, the ID 5 has no effect.)	
<b>Example</b>	AT+CIPCLOSE	AT+CIPCLOSE=1
	OK	OK

### 3.3.14 AT+CIFSR: Gets the Local IP Address

	Execute Command
<b>Commands</b>	AT+CIFSR
<b>Response</b>	If WizFi360 is Station mode(AT+CWMODE=1), the system returns: +CIFSR:STAIP,<Station IP address> +CIFSR:STAMAC,<Station MAC address> OK
	If WizFi360 is SoftAP mode(AT+CWMODE=2), the system returns: +CIFSR:APIP,<SoftAP IP address> +CIFSR:APMAC,<SoftAP MAC address> OK

	<p>If WizFi360 is Station+SoftAP mode(AT+CWMODE=3), the system returns:</p> <p>+CIFSR:APIP,&lt;SoftAP IP address&gt;</p> <p>+CIFSR:APMAC,&lt;SoftAP MAC address&gt;</p> <p>+CIFSR:STAIP,&lt;Station IP address&gt;</p> <p>+CIFSR:STAMAC,&lt;Station MAC address&gt;</p> <p>OK</p>
<b>Parameter</b>	<p>&lt;SoftAP IP address&gt;: string parameter, IP address of the WizFi360 SoftAP</p> <p>&lt;Station IP address&gt;: string parameter, IP address of the WizFi360 Station</p> <p>&lt;SoftAP MAC address&gt;: string parameter, MAC address of the WizFi360 SoftAP</p> <p>&lt;Station MAC address&gt;: string parameter, MAC address of the WizFi360 Station</p>
<b>Example</b>	<p>AT+CIFSR</p> <p>+CIFSR:APIP,"192.168.4.1"</p> <p>+CIFSR:APMAC,"02:08:dc:11:12:13"</p> <p>+CIFSR:STAIP,"192.168.1.88"</p> <p>+CIFSR:STAMAC,"00:08:dc:11:12:13"</p> <p>OK</p>

### 3.3.15 AT+CIPMUX: Enable or Disable Multiple Connections

	Query Command	Set Command
<b>Commands</b>	AT+CIPMUX?	AT+CIPMUX=<mode>
<b>Function</b>	To obtain information about connection type	To set the connection type
<b>Response</b>	<p>+CIPMUX:&lt;mode&gt;</p> <p>OK</p>	OK
<b>Parameter</b>	<p>&lt;mode&gt;</p> <ul style="list-style-type: none"> <li>• 0: single connection (factory default)</li> <li>• 1: multiple connections</li> </ul>	
<b>Example</b>	AT+CIPMUX?	AT+CIPMUX=1
	<p>+CIPMUX:1</p> <p>OK</p>	OK
<b>Note</b>	<ul style="list-style-type: none"> <li>• Multiple connections can only be set when transparent transmission is disabled (AT+CIPMODE=0).</li> <li>• This mode can only be changed after all connections are disconnected.</li> <li>• If the TCP server is running, it must be deleted (AT+CIPSERVER=0) before the single connection mode is activated.</li> </ul>	

### 3.3.16 AT+CIPSERVER: Deletes/Creates TCP Server

	Set command
<b>Commands</b>	AT+CIPSERVER=<mode>[,<port>]
<b>Response</b>	OK
<b>Parameter</b>	<mode> <ul style="list-style-type: none"> <li>• 0: deletes server</li> <li>• 1: creates server</li> </ul> <port>: port number within the range of 1 ~ 65535; 333 by default
<b>Note</b>	<ul style="list-style-type: none"> <li>• A TCP server can only be created when multiple connections are activated (AT+CIPMUX=1).</li> <li>• A server monitor will automatically be created when the TCP server is created.</li> <li>• When a client is connected to the server, it will take up one connection and be assigned an ID.</li> </ul>
<b>Example</b>	AT+CIPMUX=1
	OK
	AT+CIPSERVER=1,5000
	OK

### 3.3.17 AT+CIPSERVERMAXCONN: Set the Maximum Connection Number Allowed by Server

	Query Command	Set Command
<b>Commands</b>	AT+CIPSERVERMAXCONN?	AT+CIPSERVERMAXCONN=<num>
<b>Function</b>	To obtain the maximum number of clients allowed to connect to the TCP or SSL server.	To set the maximum number of clients allowed to connect to the TCP or SSL server.
<b>Response</b>	+CIPSERVERMAXCONN:<num> OK	OK
<b>Parameter</b>	<num>: the maximum number of clients allowed to connect to the TCP or SSL server within the range of 1~4. Default value is 4.	
<b>Example</b>	AT+CIPSERVERMAXCONN?	AT+CIPSERVERMAXCONN=2
	+CIPSERVERMAXCONN:2	OK
	OK	AT+CIPSERVER=1,5000
		OK
<b>Note</b>	<ul style="list-style-type: none"> <li>• To set this configuration, you should call the command AT+CIPSERVERMAXCONN=&lt;num&gt; before creating a server.</li> </ul>	

### 3.3.18 AT+CIPMODE: Sets transmission mode

	Query Command	Set Command
<b>Commands</b>	AT+CIPMODE?	AT+CIPMODE=<mode>
<b>Function</b>	To check the transmission mode.	To set the transmission mode
<b>Response</b>	+CIPMODE:<mode> OK	OK
<b>Parameter</b>	<mode> <ul style="list-style-type: none"> <li>• 0: normal transmission mode. (factory default)</li> <li>• 1: UART-WiFi passthrough mode (transparent transmission), which can only be enabled in TCP single connection mode or in UDP mode when the remote IP and port do not change.</li> </ul>	
<b>Example</b>	AT+CIPMODE?	AT+CIPMODE=1
	+CIPMODE:1	OK
	OK	
<b>Note</b>	<ul style="list-style-type: none"> <li>• The configuration changes will NOT be saved in flash.</li> <li>• During the UART-WiFi passthrough transmission, if the TCP connection breaks, WizFi360 will keep trying to reconnect until +++ is input to exit the transmission.</li> <li>• During the normal transmission and the TCP connection breaks, WizFi360 will give a prompt and will not attempt to reconnect.</li> </ul>	

### 3.3.19 AT+SAVETRANSLINK: Saves the Transparent Transmission Link in Flash;

#### Save TCP Single Connection in Flash

	Set command
<b>Commands</b>	AT+SAVETRANSLINK=<mode>,<remote IP or domain name>,<remote port>[,<type>,<TCP keep alive>]
<b>Function</b>	Enter UART-WiFi passthrough mode and try to TCP connection on power-up.
<b>Response</b>	OK
<b>Parameter</b>	<mode> <ul style="list-style-type: none"> <li>• 0: WizFi360 will NOT enter UART-WiFi passthrough mode on power-up (factory_default)</li> <li>• 1: WizFi360 will enter UART-WiFi passthrough mode on power-up</li> </ul> <remote IP>: string parameter, remote IP or domain name <remote port>: remote port [<type>]: string parameter, TCP or UDP, TCP by default [<TCP keep alive>]: TCP is kept alive. This function is disabled by default <ul style="list-style-type: none"> <li>• 0: disables the TCP keep-alive function</li> <li>• 1 ~ 7200: keep-alive detection time interval; unit: second (s)</li> </ul>

Example	AT+SAVETRANSLINK=1,"192.168.2.2",5000,"TCP",5
	OK

### Save UDP Single Connection in Flash

	Set command
Commands	AT+SAVETRANSLINK=<mode>,<remote IP>,<remote port>,<type>[,<UDP Local port>]
Function	Enter UART-WiFi passthrough mode and try to UDP connection on power-up.
Response	OK
Parameter	<mode> <ul style="list-style-type: none"> <li>• 0: normal mode; WizFi360 will NOT enter UART-WiFi passthrough mode on power-up</li> <li>• 1: WizFi360 enters UART-WiFi passthrough mode on power-up</li> </ul> <remote IP>: string parameter, remote IP or domain name <remote port>: remote port <type>: string parameter, TCP or UDP, TCP by default; UDP in this case [<UDP local port>]: local port when UDP transparent transmission is enabled on power-up
Example	AT+SAVETRANSLINK=1,"192.168.2.2",5000,"UDP",6000
	OK

### 3.3.20 AT+CIPSTO: Sets the TCP Server Timeout

	Query Command	Set Command
Commands	AT+CIPSTO?	AT+CIPSTO=<time>
Function	To check the TCP server timeout.	To set the TCP server timeout.
Response	+CIPSTO:<time> OK	OK
Parameter	<time>: TCP server timeout within the range of 0 ~ 7200s.	
Example	AT+CIPSTO?	AT+CIPSTO=180
	+CIPSTO:10	OK
	OK	

### 3.3.21 AT+CIUPDATE: Updates the Software Through WiFi<sup>5</sup>

Execute Command
-----------------

<sup>5</sup> In v. 1.0.x.x: [http://wizwiki.net/download/WizFi360/WizFi360\\_SDK.img](http://wizwiki.net/download/WizFi360/WizFi360_SDK.img)

<b>Commands</b>	AT+CIUPDATE[=<url>]	AT+CIUPDATE="http://<ip>:8080/<filename>"
<b>Function</b>	Updates firmware by connecting to url	Updates firmware using the Upgrade Tool
<b>Response</b>	+CIUPDATE: <n> OK	
<b>Parameter</b>	[<url>]: string parameter, Firmware file path location.	<ip>: Local IP address <filename>: firmware file name
	<n>: <ul style="list-style-type: none"> <li>• 1: find the server</li> <li>• 2: connect to server</li> <li>• 3: get the software version</li> <li>• 4: start updating</li> </ul>	
<b>Example</b>	AT+CIUPDATE	AT+CIUPDATE="http://192.168.0.2:8080/WizFi360_SDK.img"
	+CIUPDATE:<1>	+CIUPDATE:<1>
	+CIUPDATE:<2>	+CIUPDATE:<2>
	+CIUPDATE:<3>	+CIUPDATE:<3>
	+CIUPDATE:<4>	+CIUPDATE:<4>
	OK	OK
<b>Note</b>	<ul style="list-style-type: none"> <li>• In case that updates firmware using the Upgrade Tool, please refer to Firmware Update Guide</li> </ul>	

### 3.3.22 AT+PING: Ping Packets

	Execute Command
<b>Commands</b>	AT+PING=<IP address>
<b>Response</b>	+<Time>
	OK
	+timeout ERROR
<b>Parameter</b>	<IP>: string parameter, host IP or domain name <time>: the response time of ping (in ms)
<b>Example</b>	AT+PING="www.google.com"

In v. 1.1.x.x: [http://wizwiki.net/download/WizFi360/O11/WizFi360\\_SDK.img](http://wizwiki.net/download/WizFi360/O11/WizFi360_SDK.img)

	+52
	OK

### 3.3.23 AT+CIPDINFO: Shows the Remote IP and Port with +IPD

	Query Command	Set Command
<b>Commands</b>	AT+CIPDINFO?	AT+CIPDINFO=<mode>
<b>Response</b>	+CIPDINFO:TRUE or FALSE	OK
<b>Parameter</b>	<mode> • 0: does not show the remote IP and port with +IPD. • 1: shows the remote IP and port with +IPD.	
<b>Example</b>	AT+CIPDINFO?	AT+CIPDINFO
	+CIPDINFO:TRUE	OK
	OK	

### 3.3.24 +IPD: Receive Network Data

	single connection (AT+CIPMUX=0)	multiple connections (AT+CIPMUX=1)
<b>Commands</b>	+IPD,<len>[,<remote IP>,<remote port>]:<data>	+IPD,<link ID>,<len>[,<remote IP>,<remote port>]:<data>
<b>Parameter</b>	<link ID>: ID of network connection (0~4) <len>: data length. [<remote IP>]: string parameter, remote IP, enabled by command AT+CIPDINFO=1. [<remote port>]: remote port, enabled by command AT+CIPDINFO=1. <data>: data received.	
<b>Example</b>	+IPD,5:12345	+IPD,1,5,"192.168.0.10",50000:12345
<b>Note</b>	• The command is valid in normal command mode. When the module receives network data, it will send the data through the serial port using the +IPD command.	

### 3.3.25 AT+CIPSNTPCFG: Sets the Configuration of SNTP

	Query Command	Set Command
<b>Commands</b>	AT+CIPSNTPCFG?	AT+CIPSNTPCFG=<enable>[,<timezone>][,<SNTP server0>][,<SNTP server1>][,<SNTP server2>]
<b>Function</b>	To check the SNTP Server.	To set the SNTP Server.

<b>Response</b>	+CIPSNTPCFG:<enable>[,<timezone>,<SNTP server1>,<SNTP server2>,<SNTP server3>] OK	OK
<b>Parameter</b>	<enable> <ul style="list-style-type: none"> <li>• 0: SNTP is disabled (factory default)</li> <li>• 1: SNTP is enabled</li> </ul> [<timezone>]: time zone; range: [-11,13]; if SNTP is enabled, the <timezone> has to be set [<SNTP server0>]: string parameter indicating the first SNTP server [<SNTP server1>]: string parameter indicating the second SNTP server [<SNTP server2>]: string parameter indicating the third SNTP server	
<b>Example</b>	AT+CIPSNTPCFG?	AT+CIPSNTPCFG=1,8,"cn.ntp.org.cn","ntp.sjtu.edu.cn","us.pool.ntp.org"
	+CIPSNTPCFG:1,8,"cn.ntp.org.cn" OK	OK
<b>Note</b>	• If the <SNTP server> Parameter are not set, servers "cn.ntp.org.cn","ntp.sjtu.edu.cn", and "us.pool.ntp.org" will be used by default.	

### 3.3.26 AT+CIPSNTPTIME: Checks the SNTP Time

	Query Command
<b>Commands</b>	AT+CIPSNTPTIME?
<b>Response</b>	+CIPSNTPTIME:<time> OK
<b>Parameter</b>	<time>: SNTP time
<b>Example</b>	AT+CIPSNTPTIME?
	+CIPSNTPTIME:Wed Jul 24 11:38:25 2019 OK

### 3.3.27 AT+CIPDNS\_CUR: Sets User-defined DNS Servers; Configuration Not Saved in the Flash

	Query command	Set Command
<b>Commands</b>	AT+CIPDNS_CUR?	AT+CIPDNS_CUR=<enable>[,<DNS server0>][,<DNS server1>]
<b>Function</b>	Get the current DNS server	Set user-defined DNS servers
<b>Response</b>	[+CIPDNS_CUR:<DNS server0>] [+CIPDNS_CUR:<DNS server1>]	OK



	OK	
Parameter	-	<enable> <ul style="list-style-type: none"> <li>• 0: disable to use user-defined DNS servers</li> <li>• 1: enable to use user-defined DNS servers</li> </ul>
	<DNS server0>: string parameter indicating the first DNS server <DNS server1>: string parameter indicating the second DNS server	
Example	AT+CIPDNS_CUR?	AT+CIPDNS_CUR=1,"1.1.1.1","8.8.8.8"
	+CIPDNS_CUR: 1.1.1.1 +CIPDNS_CUR: 8.8.8.8	OK
	OK	
Note	<ul style="list-style-type: none"> <li>• The configuration changes will NOT be saved in the flash.</li> <li>• If &lt;enable&gt; is 1 and &lt;DNS server0&gt; and &lt;DNS server1&gt; are not fill, DNS server will be used "208.67.222.222".</li> <li>• DNS server(s) may be changed after AT+CWDHCP-commands.</li> <li>• &lt;DNS server0&gt; &amp; &lt;DNS server1&gt; must be different.</li> </ul>	

### 3.3.28 AT+CIPDNS\_DEF: Sets User-defined DNS Servers; Configuration Saved in the Flash

	Query command	Set Command
Commands	AT+CIPDNS_DEF?	AT+CIPDNS_DEF=<enable>[,<DNS server0>][,<DNS server1>]
Function	Get the user-defined DNS servers which saved in flash.	Set user-defined DNS servers
Response	[+CIPDNS_DEF:<DNS server0>] [+CIPDNS_DEF:<DNS server1>]  OK	OK
Parameter	-	<enable> <ul style="list-style-type: none"> <li>• 0: disable to use user-defined DNS servers</li> <li>• 1: enable to use user-defined DNS servers</li> </ul>
	<DNS server0>: string parameter indicating the first DNS server <DNS server1>: string parameter indicating the second DNS server  Factory default value In v.1.0.x.x: no value In v. 1.1.x.x: 208.67.220.220	
Example	AT+CIPDNS_DEF?	AT+CIPDNS_DEF=1,"1.1.1.1","8.8.8.8"
	+CIPDNS_DEF: 1.1.1.1	OK

	+CIPDNS_DEFF: 8.8.8.8	
	OK	
<b>Note</b>	<ul style="list-style-type: none"> <li>The configuration changes will be saved in the system parameter area in the flash.</li> <li>If &lt;enable&gt; is 1 and &lt;DNS server0&gt; and &lt;DNS server1&gt; are not fill, DNS server will be used "208.67.222.222".</li> <li>DNS server(s) may be changed after AT+CWDHCP-commands.</li> <li>&lt;DNS server0&gt; &amp; &lt;DNS server1&gt; must be different.</li> </ul>	

### 3.3.29 AT+MQTTSET: Sets the Configuration of MQTT connection.

	Query command	Set Command
<b>Commands</b>	AT+MQTTSET?	AT+MQTTSET=<User Name>,<Password>,<ClientID>,<AliveTime>
<b>Response</b>	<UserName>,<Password>,<ClientID>,<AliveTime> > OK	OK
<b>Parameter</b>	<UserName>: string parameter, User Name used in the broker authentication Max: 50byte <Password>: string parameter, Password used in the broker authentication. Max: 50byte <ClientID>: string parameter, Client ID connected to the broker. Max: 50byte <AliveTime>: keep-alive time setting with the broker within the range of 30s~300s.	
<b>Example</b>	AT+MQTTSET=?	AT+MQTTSET="wiznet","12345678","wiznet-01",60
	"wiznet","12345678","wiznet-01",60 OK	OK
<b>Note</b>	<ul style="list-style-type: none"> <li>This command should be set before connecting to a broker.</li> </ul>	

### 3.3.30 AT+MQTTTOPIC: Sets the Topic of Publish and Subscribe

	Query command	Set Command
<b>Commands</b>	AT+MQTTTOPIC?	AT+MQTTTOPIC=<publish topic>,<subscribe topic1>,<subscribe topic2>,<subscribe topic3>]
<b>Response</b>	<publish topic>,<subscribe topic> OK	OK
<b>Parameter</b>	<publish topic>: string parameter, The topic published on the WizFi360 <subscribe topic1>: string parameter, The topic subscribed by the WizFi360 [<subscribe topic2>]: string parameter, The topic subscribed by the WizFi360 [<subscribe topic3>]: string parameter, The topic subscribed by the WizFi360	
<b>Example1</b>	AT+MQTTTOPIC?	AT+MQTTTOPIC="pubTopic","subTopic"

	"pubTopic","subTopic"	OK
	OK	
<b>Example2</b>	AT+MQTTTOPIC?	AT+MQTTTOPIC="pubTopic","subTopic1","SubTopic2","SubTopic3"
	"pubTopic","subTopic1","SubTopic2","SubTopic3"	OK
	OK	
<b>Note</b>	<ul style="list-style-type: none"> <li>• This command should be set before connecting to a broker.</li> <li>• &lt;subscribe topic2&gt; and &lt;subscribe topic3&gt; are available after firmware v1.0.5.0.</li> </ul>	

### 3.3.31 AT+MQTTCON: Connects to a Broker

	Single Connection (AT+CIPMUX=0)	Multiple Connection (AT+CIPMUX=1)
<b>Commands</b>	AT+MQTTCON=<enable>,<broker IP>,<broker port>	AT+MQTTCON=<linkID>,<enable>,<broker IP>,<broker port>
<b>Response</b>	CONNECT	
	OK	
<b>Response</b>	CONNECT FAIL	
	ERROR	
<b>Parameter</b>	<link ID>: ID of network connection (0~4), used for multiple connections (AT+CIPMUX=1) <enable>: <ul style="list-style-type: none"> <li>• 0: Connect to a broker without authentication</li> <li>• 1: Connect to a broker with authentication</li> </ul> <broker IP>: string parameter indicating the broker IP address <broker port>: the broker port number	
<b>Message</b>	Whenever messages of subscribe topic is received, it will return as below	
	<subscribe topic> -> "subscribe data"	
<b>Example</b>	AT+MQTTCON=0,0,"192.168.1.20",1883	
	CONNECT	
	OK	

### 3.3.32 AT+MQTTPUB: Publish a message

	Set Command
<b>Commands</b>	AT+MQTTPUB=<message>

<b>Response</b>	OK
<b>Parameter</b>	<message>: string parameter, Publish the message to subscribed Client
<b>Example</b>	AT+MQTTPUB="publish data"
	OK
<b>Note</b>	<ul style="list-style-type: none"> <li>• This command can only be used after MQTT Connection is established.</li> <li>• Topic of published data is defined by AT+MQTTTOPIC, you should set a topic of publish before connecting to a broker.</li> </ul>

### 3.3.33 AT+MQTTDIS: Disconnects from a Broker

	Execute Command
<b>Commands</b>	AT+MQTTDIS
<b>Response</b>	CLOSE

### 3.3.34 AT+AZSET: Sets the Configuration of Azure IoT Hub connection.

	Set Command
<b>Commands</b>	AT+AZSET=<hub ID>,<device ID>,<key>
<b>Response</b>	OK
<b>Parameter</b>	<hub ID>: string parameter, ID of IoT Hub <device ID>: string parameter, ID of IoT Device <key>: string parameter, Key of IoT Device
<b>Example</b>	AT+AZSET="wiznet-iothub","wiznettest","-----"
	OK
<b>Note</b>	<ul style="list-style-type: none"> <li>• This command should be set before connecting to a Azure IoT Hub.</li> </ul>

### 3.3.35 AT+AZCON: Connects to a Azure IoT Hub.

	Set Command
<b>Commands</b>	AT+AZCON
<b>Response</b>	CONNECT
	OK
	CONNECT FAIL
	ERROR

<b>Message</b>	Whenever messages of subscribe topic is received, it will return as below. At that time, <Subscribe topic> should be set as "devices/<device ID>/message /devicebound/#" through AT + MQTTTOPIC command.
	devices/<device ID>/messages/devicebound/<property_bag> -> <subscribe data>
<b>Parameter</b>	<device ID>: IoT device ID <property bag>: url-encoded key/value pairs of message properties. <subscribe data>: subscribe data
<b>Example</b>	AT+AZSET="WIZnetTest","wizfi360","-----"
	OK
	AT+MQTTTOPIC="devices/wizfi360/messages/events/","devices/wizfi360/messages/devicebound/#"
	OK
	AT+AZCON
	CONNECT
	OK
	Whenever messages of subscribe topic is received, it will return as below
	devices/wizfi360/messages/devicebound/%24.mid=test1 &%24.to=%2Fdevices%2Fwizfi360%2Fmessages%2FdeviceBound&iotHub-ack=full -> Hello Wiznet!
<b>Note</b>	<ul style="list-style-type: none"> <li>• This command can only be used after AT+AZSET command and AT+MQTTTOPIC command.</li> <li>• Send the data to Azure Server using AT+MQTTPUB command after connection.</li> <li>• For details refer to <a href="https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-mqtt-support">https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-mqtt-support</a></li> </ul>

### 3.3.36 AT+AWSPKSEND: Set Private Key

	Query command	Set command
<b>Commands</b>	AT+AWSPKSEND?	AT+AWSPKSEND=<mode>
<b>Function</b>	Get the Private Key.	Set the Private Key.
<b>Response</b>	-----BEGIN RSA PRIVATE KEY----- ... ... -----END RSA PRIVATE KEY-----	>
	ERROR	OK
<b>Parameter</b>		<mode>: <ul style="list-style-type: none"> <li>• 0: Delete a key</li> <li>• 1: Generate a key</li> </ul>

<b>Example1</b>	AT+AWSPKSEND?	AT+AWSPKSEND=1
	-----BEGIN RSA PRIVATE KEY----- ... ... -----END RSA PRIVATE KEY-----	>
		-----BEGIN RSA PRIVATE KEY----- ... ... -----END RSA PRIVATE KEY-----
		OK
<b>Example2</b>	AT+AWSPKSEND?	AT+AWSPKSEND=0
	ERROR	OK
<b>Note</b>	<ul style="list-style-type: none"> <li>• Wrap return &gt; after the set command. WizFi360 begins receiving serial data. User should input the data of certificate including lines "-----BEGIN RSA PRIVATE KEY-----" and "-----END RSA PRIVATE KEY-----".</li> </ul>	

### 3.3.37 AT+CLICASEND: Set Client certificate

	Query command	Set command
<b>Commands</b>	AT+CLICASEND?	AT+CLICASEND=<mode>
<b>Function</b>	Get the Client Certificate.	Set the Client Certificate.
<b>Response</b>	-----BEGIN CERTIFICATE----- ... ... -----END CERTIFICATE-----	>
	ERROR	OK
<b>Parameter</b>		<mode>: <ul style="list-style-type: none"> <li>• 0: Delete a certificate</li> <li>• 1: Generate a certificate</li> </ul>
<b>Example1</b>	AT+CLICASEND?	AT+CLICASEND=1
	-----BEGIN CERTIFICATE----- ... ... -----END CERTIFICATE-----	>
		-----BEGIN CERTIFICATE----- ... ... -----END CERTIFICATE-----
		OK
<b>Example2</b>	AT+CLICASEND?	AT+CLICASEND=0
	ERROR	OK
<b>Note</b>	Wrap return > after the set command. WizFi360 begins receiving serial data. User should input the	

	data of certificate including lines "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----".
--	--

### 3.3.38 AT+AWSCON: Connect to AWS IoT Core.

	Set Command
<b>Commands</b>	AT+AWSCON
<b>Response</b>	CONNECT
	OK
	CONNECT FAIL
	ERROR
<b>Message</b>	Whenever messages of subscribe topic is received, it will return as below. At that time, <Subscribe topic> should be set as "\$aws/things/<thingName>/shadow/update/accepted" through AT + MQTTPUB command.
	\$aws/things/<thingName>/shadow/update/accepted -> <subscribe data>
<b>Parameter</b>	<thingName>: Thing Name <subscribe data>: subscribe data
<b>Example</b>	AT+MQTTSET="","",<ThingName>,60
	OK
	AT+MQTTTOPIC="\$aws/things/<thingName>/shadow/update","\$aws/things/<thingName>/shadow/update/accepted"
	OK
	AT+AWSCON
	CONNECT
	OK
	Whenever messages of subscribe topic is received, it will return as below
	\$aws/things/<thingName>/shadow/update/accepted -> Hello Wiznet!
<b>Note</b>	<ul style="list-style-type: none"> <li>• This command can only be used after AT+MQTTSET command and AT+MQTTTOPIC command.</li> <li>• Send the data to AWS IoT Core using AT+MQTTPUB command after connection.</li> <li>• This command cannot be used when AT+CIPMUX=1</li> <li>• To use this command SSL certificate option shall be set as AT+CIPSSLCONF=2</li> </ul>

## 4 Command History

Commands can be used after the firmware version.

Firmware Version	Command
<b>1.0.1.0</b>	AT, AT+RST, AT+GMR, AT+GSLP, ATE, AT+RESTORE, AT+UART_CUR, AT+UART_DEF, AT+SLEEP, AT+SYSIOSETCFG, AT+SYSIOGETCFG, AT+SYSGPIODIR, AT+SYSGPIOWRITE, AT+SYSGPIOREAD, AT+CWMODE_CUR, AT+CWMODE_DEF, AT+CWJAP_CUR, AT+CWJAP_DEF, AT+CWLAPOPT, AT+CWLAP, AT+CWQAP, AT+CWSAP_CUR, AT+CWSAP_DEF, AT+CWLIF, AT+CWDHCP_CUR, AT+CWDHCP_DEF, AT+CWDHCPS_CUR, AT+CWDHCPS_DEF, AT+CWAUTOCONN, AT+CIPSTAMAC_CUR, AT+CIPSTAMAC_DEF, AT+CIPAPMAC_CUR, AT+CIPAPMAC_DEF, AT+CIPSTA_CUR, AT+CIPSTA_DEF, AT+CIPAP_CUR, AT+CIPAP_DEF, AT+CWSTARTSMART, AT+CWSTOPSMART, AT+WPS, AT+CWHOSTNAME, AT+CWCOUNTRY_CUR, AT+CWCOUNTRY_DEF, AT+CIPSTATUS, AT+CIPDOMAIN, AT+CIPSTART, AT+CIPSSLSIZE, AT+CIPSEND, AT+CIPSENDEX, AT+CIPSENDERBUF, AT+CIPBUFRESET, AT+CIPBUFSTATUS, AT+CIPCHECKSEQ, AT+CIPCLOSE, AT+CIFSR, AT+CIPMUX, AT+CIPSERVER, AT+CIPSERVERMAXCONN, AT+CIPMODE, AT+SAVETRANSLINK, AT+CIPSTO, AT+CIUPDATE, AT+PING, AT+CIPDINFO, +IPD, AT+CIPSNTPCFG, AT+CIPSNTPTIME, AT+CIPDNS_CUR, AT+CIPDNS_DEF
<b>1.0.2.4</b>	AT+MQTTSET, AT+MQTTTOPIC, AT+MQTTCON, AT+MQTTPUB, AT+MQTTDIS
<b>1.0.4.1</b>	AT+AZSET, AT+AZCON
<b>1.0.5.1</b>	AT+CIPSSLCCONF, AT+CASEND
<b>1.0.7.2</b>	AT+WIZ_NETCONFIG
<b>1.1.1.2</b>	AT+AWSCON, AT+AWSPKSEND, AT+CLICASEND



## Copyright Notice

Copyright 2020 WIZnet Co., Ltd. All Rights Reserved.

Technical Support: <https://forum.wiznet.io/>

Wiki : <https://wizwiki.net>

Sales & Distribution: [sales@wiznet.io](mailto:sales@wiznet.io)

For more information, visit our website at <http://www.wiznet.io/>