OPL1000

ULTRA-LOW POWER 2.4GHZ WI-FI + BLUETOOTH SMART SOC

AT Instruction Set and Examples

OPULINKS

REVISION HISTORY

Date	Version	Contents Updated
2018-04-16	0.1	• Initial Release
2018-05-18	0.2	• Add wifi example • Add AT+CWAUTOCONN
2018-05-25	0.3	• Fix some mistakes
2018-05-31	0.4	• Add tcp server example • Fix some mistakes
2018-06-20	0.5	• Fix some mistakes
2018-06-27	0.6	• Fixed some wrong description in several AT commands such • as AT+BLECFGMTU etc.
2018-06-28	0.7	• Fix some description mistakes
2018-07-17	0.8	• Revise CIPSTATUS description • Add AT+MACADDRDEF
2018-07-19	0.9	• Add AT+RFHP • Modify BLE public address AT+BLEADDR
2018-07-27	0.10	• Add more description for AT+BLEADDR and AT+CIPSTAMAC
2018-07-27	0.11	• Add AT+READFLASH, AT+WRITEFLASH and AT+ERASEFLASH
2018-08-02	0.12	• Add more description for AT+CWMODE and AT+CWLAP
2018-08-17	0.13	• Add AT+DHCPARPCHK
2018-08-27	0.14	• Revise AT+CWMODE description
2018-08-27	0.15	• Update AT+RFHP description
2018-09-04	0.16	• Update valid range for AT+WRITEFLASH and AT+ERASEFLASH
2018-09-06	0.17	• Update AT+SWITCHDBG description
2018-09-13	0.18	• Add AT+MACDATARATE • Set up Wi-Fi Tx data rate
2018-09-13	0.19	• Add AT+DTM
2018-09-18	0.20	• Add AT+WIFIMACCFG
2018-09-18	0.21	• Update AT+GSLP description

REVISION HISTORY

Date	Version	Contents Updated
2018-09-18	0.22	• Update AT+CWAUTOCONN description
2018-09-18	0.23	• Update AT+ BLEADVPARAM description
2018-09-19	0.24	• Add AT+CWFASTCONN
2018-09-20	0.25	• Add AT+RFTM
2018-09-20	0.26	• Add AT+BLEPHYREAD • Add AT+BLEPHYSET
2018-09-20	0.27	• Update AT+CIPSTAMAC description
2018-09-20	0.28	• Add RF-related AT commands
2018-10-05	0.29	• Update AT+CWJAP description
2018-11-21	0.30	• Add AT+RFTM (BeaconOnlyMode)
2020-08-04	0.31	• Remove AT+CIPSAT
2020-12-01	0.32	• Update AT+RFHP power setting

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

1. INTRODUCTION

1.1. Scope of Document Application

This file is a consolidated description of functions of OPL1000 AT Command Set, and of Application Methods.

The Command set mainly comprises of Basic AT Commands, WIFI Function AT Commands, TCP/IP Function AT Commands, and BLE Function AT Commands, etc. OPL1000 AT Commands, by default, use serial-port UART1 for transmission, with baud rate of 115200bps, and format of "8N1".

1.2. Command Descriptions

AT Commands can be categorized into 4 types:

Type	Command Format	Description
Test Command(s)	AT+<x>=?	This command is used for inquiry of the parameters and value range of installation commands.
Inquiry Command(s)	AT+<x>?	This command is used for restoring to the current value of parameters
Installation Command(s)	AT+<x>=<…>	This command is used for install user-defined parameter values.
Execution Command(s)	AT+<x>	This command is used for variable parameter functions controlled by internal procedure of modules.

Note：

- Not every AT command contains instructions of all 4 types described above.
- The value contained inside [ ] is of default value, which can be left blank, or may not have displayed value.
- AT commands do not distinguish between upper and lower cases.
- AT commands end with "ENTER" \r\n. Please note to install serial-port tool as "New Line Mode".

2. BASIC AT COMMANDS

2.1. List of Basic AT Commands

Command(s)	Description
AT	Test AT Module
AT+RST	Re-Activate Module
AT+GMR	Check Version Message
AT+GSLP	Enter Deep-Sleep Mode
ATE	Terminate Echo Function
AT+RESTORE	Restore to Factory Setting
AT+UART_CUR	Set up current UART Temporary Setting
AT+UART_DEF	Set up UART Setting, and Save to Flash
AT+SLEEP	Install Sleep Mode
AT+SYSRAM	Inquiry of Currently Remaining RAM Size
AT+MACADDRDEF	Install Dervice Origin Mac Address Stored to Flash
AT+RFHP	Install RF Power
AT+READFLASH	Read Flash
AT+WRITEFLASH	Write Flash
AT+ERASEFLASH	Erase Flash
AT+DHCPARPCHK	Install DHCP ARP Check Procedure
AT+MACDATARATE	Install Wi-Fi Tx Data Rate
AT+WIFIMACCFG	Install Wi-Fi Related Parameters

2.2. Description of Basic AT Commands

2.2.1. AT—Testing of AT Module

Command Execution	**AT**
Feedback	OK
Parameter Description(s)	-

2.2.2. AT+RST—Reset Module

Command Execution	**AT+RST**
Feedback	OK
Parameter Description(s)	-
Note	After having executed this command, the system will reboot, by default

2.2.3. AT+GMR—Version Inquiry Message

Command Execution	**AT+GMR**
Feedback	<AT version info> <SDK version info> <compile time> OK
Parameter Description(s)	• <AT version info> • <SDK version info> • <compile time>

2.2.4. AT+GSLP—Entry into Deep-Sleep Mode

Command Execution	**AT+GSLP=<Duration>, <I/O>**
Feedback	<Duration> OK
Parameter Description(s)	<Duration>： Install OPL1000 sleep duration in unit of millisecond. OPL1000 will be automatically woken after the sleep duration lapses. <I/O> GPIO number to Wakeup

2.2.5. ATE—On-Off Echo Function

Command Execution	**ATE**
Feedback	OK
Parameter Description(s)	• ATE0：Switch-off Echo • ATE1：Switch-on Echo

2.2.6. AT+RESTORE—Restore to Factory Setting

Command Execution	**AT+RESTORE**
Feedback	OK
Note	By restoring to factory setting, it will erase all parameters stored in Flash, as all parameters will be of default value.。 Restore factory setting will result in system reboot.

2.2.7. AT+UART_CUR—Install UART Temporary Setting

Command(s)	Inqury of Command(s)： AT+UART_CUR?	Designate Command(s)： AT+UART_CUR=<baudrate>,<databits>,<stopbits>,<parity>,<flow control>
Feedback	+UART_CUR:<baudrate>,<databits>,<stopbits>,<parity>,<flow control> OK The inquiry returns UART actual parameter values, and due to clock division, it is normal to have some deviance in value between the actual and default parameter values of UART.	OK
Parameter Description(s)	• <baudrate>：UART baudrate • <databits> ‣ 5：5 bit databit ‣ 6：6 bit databit ‣ 7：7 bit databit ‣ 8：8 bit databit • <stopbits>： ‣ 1：1 bit stopbit ‣ 2：1.5 bit stopbit ‣ 3：2 bit stopbit • <parity>： ‣ 0：None ‣ 1：Odd ‣ 2：Even • <flow control>： ‣ 0：Disable Flow Control ‣ 1：Reserved	

Command(s)	Inqury of Command(s)： AT+UART_CUR?	Designate Command(s) ： AT+UART_CUR=<baudrate>,<databits>,<stopbits>,<parity>,<flow control>
	‣ 2：Reserved ‣ 3：Enable both RTS & CTS	
Note	• This device does not save to Flash • Flow control requires hardware support •Range of Baudrate supported： 80 ~ 1000000	
Examples	AT+UART_CUR=115200,8,1,0,3	

2.2.8. AT+UART_DEF—Install UART Setting, and Save to Flash

Command(s)	Inqury of Command(s)： AT+UART_DEF?	Designate Command(s) ： AT+UART_DEF=<baudrate>,<databits>,<stopbits>,<parity>,<flow control>
Feedback	+UART_DEF:<baudrate>,<databits>,<stopbits>,<parity>,<flow control> OK	OK
Parameter Description(s)	• <baudrate>： UART baudrate • <databits>： ‣ 5： 5 bit databit ‣ 6： 6 bit databit ‣ 7： 7 bit databit ‣ 8： 8 bit databit • <stopbits>： ‣ 1： 1 bit stopbit ‣ 2： 1.5 bit stopbit ‣ 3： 2 bit stopbit	

Command(s)	**Inqury of Command(s)：** **AT+UART_DEF?**	**Designate Command(s) ：** **AT+UART_DEF=<baudrate>,<databits>,<stopbits>,<parity>,<flow control>**
	• <parity>： ‣ 0： None ‣ 1： Odd ‣ 2： Even • <flow control>： ‣ 0：Disable Flow Control ‣ 1：Reserved ‣ 2：Reserved ‣ 3：Enable both RTS & CTS	
Note	• This device will save to Flash, in which the saved data remains even after restarting power-on •Flow control requires hardware support •Range of Baudrate supported： 80 ~ 1000000	
Examples	AT+UART_DEF=115200,8,1,0,3	

2.2.9. AT+SLEEP—Install Sleep Mode

Command(s)	**Designate Command(s) ：** **<sleep mode> = 0** **AT+SLEEP=<sleep mode>** **<sleep mode> = 1,3** **AT+SLEEP=<sleep mode>, <I/O>** **<sleep mode> = 2** **AT+SLEEP=<sleep mode>, <Duration>, <I/O>**
Feedback	OK
Parameter Description(s)	<sleep mode>： ‣ 0：Terminate Smart-sleep mode

Command(s)	**Designate Command(s) :** **<sleep mode> = 0** **AT+SLEEP=<sleep mode>** **<sleep mode> = 1,3** **AT+SLEEP=<sleep mode>, <I/O>** **<sleep mode> = 2** **AT+SLEEP=<sleep mode>, <Duration>, <I/O>**
	‣ 1： Activate Smart-sleep mode <I/O> GPIO number to Wakeup ‣ 2： Enter into Timer-sleep mode <Duration> : Designate the sleep duration of OPL1000 with unit of millisecond. OPL1000 will automatically wake up from sleep mode. <I/O> GPIO number to Wakeup ‣ 3： Enter into Deep-sleep mode < I/O> GPIO number to Wakeup
Examples	AT+SLEEP=0 AT+SLEEP=1,20 // Wake-up by GPIO 20 AT+SLEEP=2,1000, 20 // Wake-up by GPIO 20 AT+SLEEP=3,20 // Wake-up by GPIO 20

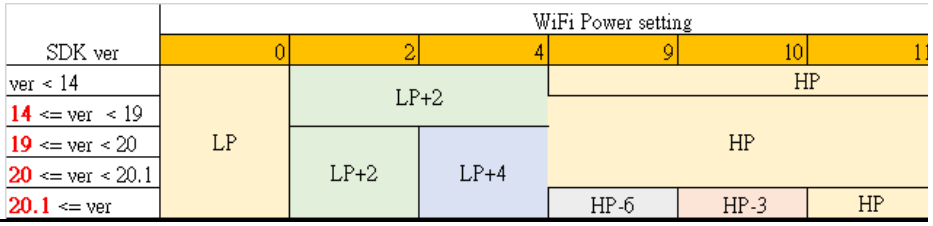
2.2.10. AT+SYSRAM—Inquiry of Currently Remaining RAM Size

Inquiry of Command(s)	**AT+SYSRAM?**
Feedback	+SYSRAM:<remaining RAM size> OK
Parameter Description(s)	<remaining RAM size>： Unit: Byte
Examples	AT+SYSRAM? +SYSRAM:148408 OK

2.2.11. AT+MACADDRDEF—Install Dervice Origin Mac Address Stored to Flash

Command(s)	Inqury of Command(s)： AT+MACADDRDEF?	Designate Command(s) ： AT+MACADDRDEF=<iface>,<type>
Feedback	+MACADDRDEF:<iface_0>,<type>,<iface_1>,<type> OK	OK
Parameter Description(s)	• <iface> ： Device Interface ‣ 0 : Wifi Station ‣ 1 : BLE • <type> ：MAC address origin of set-up ‣ 0 ： From OTP ‣ 1 ： From Flash	
Note	•This device will save to Flash, in which the saved data remains even after restarting power-on. • Default origin of installed set-up for MAC address as OTP •Once successfully installed, it needs to be powered on in restart in order for MAC address set-up to come into effect.	
Examples	AT+MACADDRDEF=0,1	

2.2.12. AT+RFHP—RF Power Set up

Command(s)	Inqury of Command(s) : AT+RFHP?	Designate Command(s) : AT+RFHP=<rf_power>
Feedback	+RFHP:<rf_power> OK	OK
Parameter Description(s)	• <rf_power>：RF Power ‣ [3:0]: BLE setting 0: BLE LPA Other: BLE HPA ‣ [7:4]: WiFi setting 	
Note	•This device will save to Flash, in which the saved data remains even after restarting power-on •Default set-up as "240": Wi-Fi HPA and BLE LPA •Once successfully installed, it needs to be powered on in restart for set-up to come into effect.	
Examples	AT+RFHP=208 (0xD0) BLE setting: 0 (LPA) WiFi setting: 0xD (HP+3) AT+RFHP=64 (0x40) BLE setting: 0 (LPA) WiFi setting: 0x4 (LP+4)	

2.2.13. AT+READFLASH—Read FLASH

Command(s)	**Inquiry of Command(s)：** **N/A**	**Designate Command(s) ：** **AT+READFLASH=<address>,<number_of_bytes>**
Feedback	N/A	Success： <byte_0>,<byte_1>,<byte_2>,...,<byte_n> OK Fail：ERROR
Parameter Description(s)	• <address>：Flash address (Hexadecimal) ‧Permissible reading range 0x00000000 ~ 0x000FFFFF • <number_of_bytes>：Number of bytes to be read (Decimal)	
Note	•Not to exceed permissible range	
Examples	AT+READFLASH=0x00088000,8 FF,FF,FF,FF,FF,FF,FF,FF OK	

2.2.14. AT+WRITEFLASH—Write Flash

Command(s)	**Inqury of Command(s)：** **N/A**	**Designate Command(s) ：** **AT+WRITEFLASH=<address>,<number_of_bytes>,<byte_0>,<byte_1>,...,<byte_n>**
Feedback	N/A	Success：OK Fail：ERROR
Parameter Description(s)	• <address>：Flash Address (Hexadecimal) ‧Permissible reading range 0x00000000 ~ 0x000FFFFF • <number_of_bytes>：Number of bytes to be read (Decimal) • <byte_0>~<byte_n>：Write in value(Hexadecimal)	
Note	•Not to exceed permissible range	

Command(s)	**Inqury of Command(s) :** **N/A**	**Designate Command(s) :** **AT+WRITEFLASH=<address>,<number_of_bytes>,<byte_0>,<byte_1>,...,<byte_n>**
	•Before write-in, we recommend to execute AT+READFLASH so as to confirm the storage space of Flash reserved for write-in able to accommodate write-in (all values being "0xFF"); If not, please execute AT+ERASEFLASH first to clear all sectors to accommodate write-in. •As AT command only allows maximum 255 characters, so according to the format shown in the examples, there can only be 76 bytes allowed for any single write-in. •If the number designated by <byte_0>~<byte_n> is less than that by <number_of_bytes>, write-in of <byte_n> can be continuously executed until it reaches the cap number represented by <number_of_bytes>, which would ensure all designated sectors are of the same value.	
Examples	1. AT+WRITEFLASH=0x00088000,4,F0,F1,F2,F3 OK AT+READFLASH=0x00088000,8 F0,F1,F2,F3,FF,FF,FF,FF OK 2. AT+WRITEFLASH=0x00089000,8,A0 OK AT+READFLASH=0x00089000,16 A0,A0,A0,A0,A0,A0,A0,A0,FF,FF,FF,FF,FF,FF,FF,FF OK	

2.2.15. AT+ERASEFLASH—Erase Flash

Command(s)	Inqury of Command(s)： N/A	Designate Command(s)： AT+ERASEFLASH=<start_address_of_sector>,<number_of_sectors>
Feedback	N/A	Succeed：OK Fail：ERROR
Parameter Description(s)	• <start_address_of_sector>：sector start address。(Hexadecimal) ‣ Permissible value to be 0x00000000, 0x00001000, 0x00002000, … , 0x0000FE000, or 0x000FF000 • <number_of_sectors>：Number of sectors to be erased (Decimal) ‣ Starting with the addresses of the designated sector to be erased ‣ Size of a single sector is 4096 bytes.	
Note	• Erasure of Flash is done by the unit of sector, with each erasure takes out at least 4096 bytes.	
Examples	1. Erase 0x00089000 ~ 0x00089FFF：4096 bytes AT+ERASEFLASH=0x00089000,1 OK 2. Erase 0x00088000 ~ 0x00089FFF：8192 bytes AT+ERASEFLASH=0x00088000,2 OK	

2.2.16. AT+DHCPARPCHK—Set up DHCP ARP Check Procedure

Command(s)	Inqury of Command(s)： AT+DHCPARPCHK?	Designate Command(s)： AT+DHCPARPCHK=<chk_mode>
Feedback	+DHCPARPCHK:<chk_mode>	OK

Command(s)	Inqury of Command(s) : AT+DHCPARPCHK?	Designate Command(s) : AT+DHCPARPCHK=<chk_mode>
	OK	
Parameter Description(s)	• < chk_mode > : 0 : Disable 1 : Enable	
Note	•This device will save to Flash, in which the saved data remains even after restarting power-on. •Default set-up as "Enable" •Once successfully installed, it does not need to be powered on in restart, as long as the connection is terminated before re-connected again for set-up to come into effect.	
Examples	AT+DHCPARPCHK=0 OK	

2.2.17. AT+SWITCHDBG—Switch to Debug

Command(s)	Inqury of Command(s) : N/A	Designate Command(s) : AT+SWITCHDBG
Feedback	N/A	Switch: Dbg UART
Parameter Description(s)	-	
Note	•After installation is completed, switch IO pin of "AT UART"and "Debug UART"., •When switching IO, UART will receive some random and meaningless text	
Examples	>AT+SWITCHDBG C! Switch: Dbg UART	

2.2.18. AT+MACDATARATE—Set up Wi-Fi Tx Data Rate

Command(s)	Inqury of Command(s) : AT+MACDATARATE?	Designate Command(s) : AT+MACDATARATE=<data_rate_id>
Feedback	+MACDATARATE:<data_rate_id> OK	OK
Parameter Description(s)	-<data_rate_id> : 0 : Auto Rate Adaptation 1 : Fix Mac Tx data rate in 1 Mbps 2 : Fix Mac Tx data rate in 2 Mbps 3 : Fix Mac Tx data rate in 5.5 Mbps 4 : Fix Mac Tx data rate in 11 Mbps	
Note	•This device will save to Flash, in which the saved data remains even after restarting power-on. •Defaullt set-up being "Auto Rate Adaptation". •Once installation is completed, it would come into effect immediately.	
Examples	>AT+MACDATARATE=4 OK	

2.2.19. AT+WIFIMACCFG—Set up Wi-Fi Related Parameters

Command(s)	Inqury of Command(s) : AT+WIFIMACCFG?	Designate Command(s) : AT+WIFIMACCFG=<cfg_id>,<value>
Feedback	+WIFIMACCFG:<cfg_id>,<value> OK	OK
Parameter Description(s)	-<cfg_id> :	

Command(s)	Inqury of Command(s) : AT+WIFIMACCFG?	Designate Command(s) : AT+WIFIMACCFG=<cfg_id>,<value>
	0: skip DTIM, i.e. the number of DTIM packets to be skiped, ranging from 0 to 255, as the value setting of ZERO means "Not skipping DTIM", and value-setting of FIVE means "Skipping 5 DTIM packets". -<value> : Set-Value	
Note	•This device will save to Flash, in which the saved data remains even after restarting power-on. •Once installation is completed, the set-up comes into effect immediately. •If the set-value of "DTIM Skipped" is too high, it may lead to drop in throughput volume	
Examples	>AT+WIFIMACCFG=0,5 OK	

2.2.20. AT+RFTM—Set up Test Mode

Command(s)	Inqury of Command(s) : N/A	Designate Command(s) : AT+RFTM=<mode>, <param...>
Feedback	N/A	OK
Parameter Description(s)	• <mode> : RF Testing Mode ‣ 1 : RF TX Single-Tone ‣AT+RFTM=<mode>, <Frequency> ‣ 2 : MAC RX Beacon only mode ‣AT+RFTM=2, <OnCtrl> <OnCtrl> Setting: ‣ 0 : Normal mode (Default) ‣ 1 : MAC only receive Beacon, no Tx	

Command(s)	Inqury of Command(s) : N/A	Designate Command(s) : AT+RFTM=<mode>,<param...>
Note	•	
Examples	AT+RFTM=1, 2442	

2.2.21. AT+MODE—Set Up WiFi Mode

Command(s)	Inqury of Command(s) : N/A	Designate Command(s) : AT+MODE=<mode>
Feedback	N/A	OK
Parameter Description(s)	• <mode> : For WiFi testing mode, please use parameter "3".	
Note	•	
Examples	AT+MODE=3 Mode is RF OK	

2.2.22. AT+GO—Set Up WiFi Packet and Speed

Command(s)	Inqury of Command(s) : N/A	Designate Command(s) : AT+GO=<preamble>,<data_length>,<interval>,<data_rate>,<packet_count>
Feedback	N/A	OK
Parameter Description(s)	• <preamble>: 1: long	

	Others for short • <data_length>: n bytes • <interval>: n us (packet interval) • <data_rate>: 1, 2, 5.5, 11 Mbps • <packet_count>: 0 for infinite Others for given number
Note	•
Examples	AT+GO=1,30,40,1,0 Preamble type: LONG Data length: 30 bytes Interval: 40 us Data rate: 1 Mbps Tx Counts: 0 OK

2.2.23. AT+CHANNEL—Set Up WiFi Channel

Command(s)	**Inqury of Command(s)：** **N/A**	**Designate Command(s) ：** **AT+CHANNEL=<channel>**
Feedback	N/A	OK
Parameter Description(s)	• <channel>：WiFi Channel, with ranging between 1 to 14	
Note	•	
Examples	AT+CHANNEL=7 99, 7 OK	

2.2.24. AT+RESET_CNTS—Clear WiFi Rx Data Volume Count

Command(s)	Inqury of Command(s) : N/A	Designate Command(s) : AT+RESET_CNTS
Feedback	N/A	OK
Parameter Description(s)	N/A	
Note	•	
Examples	AT+RESET_CNTS OK	

2.2.25. AT+COUNTERS—Read WiFi Rx Data Volume Count

Command(s)	Inqury of Command(s) : AT+COUNTERS?	Designate Command(s) : N/A
Feedback	OK	N/A
Parameter Description(s)	N/A	
Note	•	
Examples	AT+COUNTERS? ok: 70558, err: 3836, rssi: -38 OK	

2.2.26. AT+TX—Activate/Terminate Off WiFi Tx Testing

Command(s)	Inqury of Command(s) : N/A	Designate Command(s) : AT+TX=<mode>
Feedback	N/A	OK
Parameter Description(s)	• <mode> :	

Command(s)	Inqury of Command(s)： N/A	Designate Command(s) ： AT+TX=<mode>
	1: Activate WiFi Tx Testing 0: Terminate WiFi Tx Testing	
Note	•	
Examples	AT+TX=1 OK	

2.2.27. AT+RX—Activate/Terminate WiFi Rx Testing

Command(s)	Inqury of Command(s)： N/A	Designate Command(s) ： AT+RX=<mode>
Feedback	N/A	OK
Parameter Description(s)	• <mode> ： 1: Activate WiFi Rx Testing 0: Terminate WiFi Rx Testing	
Note	•	
Examples	AT+RX=1 OK	

2.2.28. AT+DTM—Set-Up and Start BLE Tx/Rx Testing

Command(s)	Inqury of Command(s)： N/A	Designate Command(s) ： AT+DTM=<mode>,<channel>,<data_length>,<packet_type>
Feedback	N/A	OK
Parameter Description(s)	• <mode> ： tx: BLE Tx Testing rx: BLE Rx Testing end:Terminate BLE Tx/Rx Testing	

	• <channel> : 0 ~ 39 • <data_length> : n bytes • <packet_type> : 0: PRBS9 1: Pattern 11110000 2: Pattern 10101010 3: PRBS15 4: Pattern 11111111 5: Pattern 00000000
Note	•
Examples	•Start BLE Tx Testing AT+DTM=tx,20,30,2 Start DTM Tx Frequency: 20, length: 30, type: 2 OK •Terminate BLE Tx Testing AT+DTM=end RX CNT: 0 CRC OK: 0 CRC FAIL: 0 Packet count: 0 OK •Start BLE Rx Testing AT+DTM=rx,20 Start DTM Rx frequency: 20 OK •Terminate BLE Rx Testing AT+DTM=end

RX CNT: 28613
CRC OK: 28613
CRC FAIL: 0
Packet count: 28613
OK

3. WIFI FUNCTION AT COMMANDS

3.1. List of WIFI Function AT Commands

Command(s)	Descripption
AT+CWMODE	Set up Wi-Fi Mode
AT+CWJAP	Connect AP
AT+CWLAPOPT	Set up CWLAP Command Attributes
AT+CWLAP	Scan Currently Available AP
AT+CWQAP	Terminate Connection with AP
AT+CWDHCP	Set up DHCP
AT+CWAUTOCONN	Whether Power-On Would Result in Automatic Connection to AP
AT+CWFASTCONN	Whether Quickly Connect to AP
AT+CIPSTAMAC	Set up MAC Address of STA Port
AT+CWHOSTNAME	Set up Station Address of STA

3.2. Description of WIFI Function AT Commands

3.2.1. AT+CWMODE—Set up Wi-Fi Mode

Command(s)	AT+CWMODE=?	Inqury of Command(s)：AT+CWMODE?。 Function: Inquiry of OPL1000's current Wi-Fi Mode	Designate Command(s)：AT+CWMODE=<mode>。 Function: Set up OPL1000's current Wi-Fi Mode
Feedback	If Wi-Fi has not been initialized, inquiry before Return +CWMODE: 0 OK If Wi-Fi has been initialized, inquiry before Return +CWMODE: 1 OK	+CWMODE:<mode> OK	OK
Parameter Description(s)	<mode>： ‣ 0： No Wi-Fi Mode ‣ 1： Station Mode		
Note	•Before using WIFI and TCPIP related AT CMD, please first use AT+CWMODE to set up Station Mode •After Wi-Fi is initialized, if mode needs to be changed, then after having restarted AT+RST, initialization may be re-installed. •This command current only support station mode		
Examples	AT+CWMODE=1		

3.2.2. AT+CWJAP—Connect AP

Command(s)	Inqury of Command(s) : AT+CWJAP? Inquiry of the AP Message of OPL1000 Station has been connected to	Designate Command(s) : AT+CWJAP=<ssid>,<pwd>[,<bssid>] Function: Install the AP OPL1000 Station needs to be connected to.
Feedback	+CWJAP:<ssid>,<bssid>,<channel>,<rssi> OK	OK OR +CWJAP:<error code> ERROR
Parameter Description(s)	• <ssid> : String Parameter, SSID of AP • <bssid> : MAC Address of AP • <channel> : Channel Number • <rssi> : Signal Strength	• <ssid> : SSID of Target AP • <pwd> : The password length must be in 64 bytes with ASCII code • [<bssid>] : MAC Address of Target AP, normally used in the case of having multiple SSID's with the same AP • <error code> : (Only for reference) ‣ 1 : Overtime to making connection ‣ 2 : Error in password ‣ 3 : Uable to find target AP ‣ 4 : Error in connection ‣ Other Value: Unknown errors

Prompted Message(s)	// If OPL1000 station connects to an AP, it will prompt messages:
	WIFI CONNECTED
	WIFI GOT IP
	// If the WiFi connection ends, it will prompt messages:
	WIFI DISCONNECT	

Note	•Parameter installation needs to activate Station Mode • If SSID or password contains special characters such as (,") , it is invalid command.	
Examples	AT+CWJAP="abc","0123456789"	

3.2.3. AT+CWLAPOPT—Set up CWLAP Command Attributes

Designate Command(s)	**AT+CWLAPOPT=<sort_enable>,<mask>**
Feedback	OK
Parameter Description(s)	• <sort_enable>：Whether the scanned result of Command AT+CWLAP is listed according to the signal strength of RSSI values: ‣ 0： Not in the order ‣ 1：According to the priority order of RSSI • <mask>：If the corresponding bit is to be "1", then the scanned result of Command AT+CWLAP displays the related attributes, and if the corresponding bit is to be "0", then there will be no display, with details as follows: ‣ bit 0：Whether the scanned result of installation of AT+CWALAP display <ecn> ‣ bit 1：Whether the scanned result of installation of AT+CWALAP display <ssid> ‣ bit 2：Whether the scanned result of installation of AT+CWALAP display <rssi> ‣ bit 3：Whether the scanned result of installation of AT+CWALAP display <mac> ‣ bit 4：Whether the scanned result of installation of AT+CWALAP display <channel>
Examples	AT+CWLAPOPT=1,31 When the first parameter is "1", it means that subsequently Command AT+CWLAP is used, the scanned result will be displayed in the order of RSSI values of signal strength; When the second parameter is "31", i.e. "0x1F", it indicates that all the related bit of <mask> are all set as "1", and if subsequently Command AT+CWLAP is to be used, the scanned result will display all parameters.

3.2.4. AT+CWLAP—Scan Currently Available AP

Command(s)	**AT+CWLAP** **Function: List the currenty available AP's**
Feedback	+CWLAP:<ecn>,<ssid>,<rssi>,<mac>,<channel> OK
Parameter Description(s)	• <ecn>： Encryption Methods ‣ 0： OPEN ‣ 1： WEP ‣ 2： WPA_PSK ‣ 3： WPA2_PSK ‣ 4： WPA_WPA2_PSK ‣ 5： WPA2_Enterprise（Currently, AT does not support connection to this type of encrypted AP) • <ssid>： AP SSID String Parameter, SSID of AP • <rssi>： Signal Strength • [<mac>](Parameter Selection): String Parameter, MAC Address of AP • [<channel>](Parameter Selection): Channel Number
Examples	AT+CWLAP="WiFi","ca:d7:19:d8:a6:44",6 Or search for AP of designated SSID: AT+CWLAP="WiFi" If there are more than one designated AP "WiFi", all AP's containing "WiFi" in all SSID's will be sorted for east access and reference.

3.2.5. AT+CWQAP—Terminate Connection with AP

Command Execution	**AT+CWQAP**
Feedback	OK
Parameter Description(s)	-

3.2.6. AT+CWAUTOCONN—Whether Power-On Would Result in Automatic Connection to AP

Command(s)	**Inqury of Command(s) :** **AT+CWAUTOCONN?**	**Designate Command(s) :** **AT+CWAUTOCONN=<enable>,<ap_num>** **Function: Set up AUTOCONN**
Feedback	+CWAUTOCONN:<enable>,<ap_num> OK	OK OR +CWAUTOCONN:<error code> ERROR
Parameter Description	<enable>: auto connect <ab_num>: Display the maximum number of stored AP's	• <enable> : ‣ 0 : Power-on without automatic connection to AP ‣ 1 : Power-on with automatic connection to AP • <ap_num> : ‣ Set up the maximum number of stired auto connect AP, with the range of 1 to 3. •<error code> : ‣ 1 : Invalid Parameter ‣ Other Value: Other Errors
Note	• This device is stored in Flash	

Command(s)	**Inqury of Command(s) :** **AT+CWAUTOCONN?**	**Designate Command(s) :** **AT+CWAUTOCONN=<enable>,<ap_num>** **Function: Set up AUTOCONN**
	• By default, power-on results in automatic connection to AP's, maximum of 3 AP's. • After power-on, the Command AT+CWMODE=1 needs to be issued first, before the setup comes into effect.	
Examples	AT+CWAUTOCONN=1,3	

3.2.7. AT+CWFASTCONN—Quick Connection to AP

Command(s)	**Inqury of Command(s) :** **AT+CWFASTCONN?**	**Designate Command(s) :** **AT+CWFASTCONN=<list_id>,<enable>** 。 **Function: Install FASTCONN**
Feedback	+CWFASTCONN:<list_id>,<enable> > OK	OK OR +CWFASTCONN:<error code> ERROR
Parameter Description(s)	<list_id>: Index in the auto connect list <enable>:Whether to activate fast connect	• <list_id>:index Indexed from "0" • <enable> : ‣ 0 : Power-on without fast connection to AP ‣ 1 : Power-on with fast connection to AP • <error code> : ‣ 1 : Invalid parameters ‣ 2: This index does not have AP messages. ‣ Other value: Other Errors
Note	• This device is stored in Flash.	

Command(s)	Inqury of Command(s)： AT+CWFASTCONN?	Designate Command(s) ： AT+CWFASTCONN=<list_id>,<enable> 。 Function: Install FASTCONN
	•By default, Power-on without fast connection to AP • Need to issue Command AT+CWMODE=1 before the set-up starts to come into effect. • This function needs to be stored in “auto connect” message before it becomes functional	
Examples	AT+CWFASTCONN=0,1	

3.2.8. AT+CIPSTAMAC—Set up MAC Address of OPL1000 Station Port

Command(s)	Inqury of Command(s)： AT+CIPSTAMAC? Function: Inquiry of MAC Address of OPL1000 Station	Designate Command(s) ： AT+CIPSTAMAC=<mac> Function: Install MAC Address of OPL1000 Station
Feedback	+CIPSTAMAC:<mac> OK	OK
Parameter Description(s)	<mac>：String Parameter, MAC Address of OPL1000 Station	
Note	• This device is stored in Flash. Command AT+MACADDRDEF=0,1 can be issued to install MAC source as Flash • The bit 0 of the first bite of MAC Address cannot be “1”, e.g. MAC Address can be "1a:..." but cannot be "15:...". • “FF:FF:FF:FF:FF:FF” and “00:00:00:00:00:00” are illegal MAC, and hence unable to proceed with installation. • In order for revised MAC address to be effective, AT+RS needs to be used to be reset.	
Examples	AT+CIPSTAMAC="18:fe:35:98:d3:7b"	

3.2.9. AT+CWHOSTNAME—Set up the name of Station Host

Command(s)	Inqury of Command(s)： AT+CWHOSTNAME? Function: Inquiry of station name of OPL1000 Station	Designate Command(s) ： AT+CWHOSTNAME=<hostname> Function: Install the station name of OPL1000 Station
Feedback	+CWHOSTNAME:<host name> OK If OPL1000 Station Model is disabled, then RETURN +CWHOSTNAME:<null> OK	If succeed, return:OK ERROR If OPL1000 Station Model is disabled, then prompt “ERROR”
Parameter Description(s)	<hostname>：Station Name, supporting 32 bytes maximum	
Note	• This device does not save to Flash, and will be restored to default vaule after reboot. • The default Station ame of OPL1000 Station is “opulink”.	
Examples	AT+CWMODE=1 AT+CWHOSTNAME="my_test"	

4. TCP/IP FUNCTION AT COMMAND

4.1. List of TCP/IP Function AT Commands

Command(s)	Description
AT+CIPSTATUS	Inquiry of Message of Network Connection
AT+CIPDOMAIN	Domain Name Analysis Function
AT+CIPSTART	Establish TCP Connection or UDP Transmission
AT+CIPSEND	Send Data
AT+CIPSENDEX	Send Data
AT+CIPCLOSE	Terminate TCP/UDP Transmission
AT+CIFSR	Inquiry of Local IP Address
AT+CIPMUX	Install Multiple Connections
AT+CIPSERVER	Establish TCP Server
AT+CIPSTO	Set up TCP Server Over-Time
AT+CIPDINFO	Whether Prompt for Opposite-End IP and Port During Reception of network data
+IPD	Reception of Network Data
AT+PING	Ping Function

4.2. Description of TCP/IP Function AT Commands

4.2.1. AT+CIPSTATUS—Inquiry of Message of Network Connection

Command Execution	AT+CIPSTATUS
Feedback	STATUS:<stat> +CIPSTATUS:<link ID>,<type>,<remote IP>,<remote port>,<local port>,<tetype>

Command Execution	AT+CIPSTATUS
Parameter Description(s)	• <stat>： OPL1000 Station Port Status ‣ 2：OPL1000 Station has been connected to AP, and obtained IP address ‣ 3： OPL1000 Station has established TCP or UDP transmission ‣ 4： OPL1000 Station terminates network connection ‣ 5： OPL1000 Station is not connected to AP • <link ID>：Network connection ID (0-4), used for multiple-connection scenarios • <type>：String parameter, "TCP" or "UDP". • <remote IP>： String, Remote IP address • <remote port>： Remote Port Value • <local port>： OPL1000 Local Port Value • <tetype>： ‣ 0： OPL1000 as Client End ‣ 1： OPL1000 As Server

4.2.2. AT+CIPDOMAIN—Domain Name Analysis Function

Command Execution	AT+CIPDOMAIN=<domain name>
Feedback	+CIPDOMAIN:<IP address> OK OR ERROR
Parameter Description(s)	<domain name>：Domain name to be analyzed
Examples	AT+CWMODE=1 // set Station mode AT+CWJAP="SSID","password" // access to the internet AT+CIPDOMAIN="www.baidu.com" // DNS function

4.2.3. AT+CIPSTART—Establish TCP Connection or UDP Transmission

■ Establish TCP Connection

Designate Command(s)	**TCP single-connect (AT+CIPMUX=0) : AT+CIPSTART=<type>,<remote IP>,<remote port>[,<TCP keep alive>]**	**TCP multi-connect (AT+CIPMUX=1) : AT+CIPSTART=<link ID>,<type>,<remote IP>,<remote port>[,<TCP keep alive>]**
Feedback	OK	
Parameter Description(s)	• <link ID> : Network Connection ID (0~4), used for multiple connection scenarios. • <type> : String parameter, connection type,"TCP", "UDP", or "SSL". • <remote IP> : String, Remote IP address • <remote port> : Remote Port Value • [<TCP keep alive>] : During TCP keep-alive detection, this function is switched off, by default, and we recommend that you set up this function as activated. ‣ 0 : Terminate TCP keep-alive function. ‣ 1 ~ 7200 : Detection time, based on the unit of 1 second.	

Prompted Message(s)	// If the TCP connection is established, it will prompt message as below
	[<link ID>,] CONNECT
	// If the TCP connection ends, it will prompt message as below
	[<link ID>,] CLOSED	

Note	We recommend that when establishing TCP connection, "Keep-Alive"function should be activated.	
Examples	AT+CIPSTART="TCP","192.168.101.110",1000	

■ Establish UDP Transmission

Designate Command(s)	When in Single-Connection Mode AT+CIPSTART=<type>,<remote IP>,<remote port>[,(<UDP local port>),(<UDP mode>)]	Multiple-Connection Mode AT+CIPSTART=<link ID>,<type>,<remote IP>,<remote port>[,<UDP local port>,<UDP mode>]
Feedback	OK	
Parameter Description(s)	• <link ID>：Network Connection ID (0~4), used for multiple connection scenarios. • <type>：String parameter, connection type,"TCP", "UDP", or "SSL". • <remote IP>： String, Remote IP address • <remote port>： Remote Port Value • [<UDP local port>]： UDO Local Port • [<UDP mode>]： If the attribute of UDP transmission is "Penetrating", the value should be "0". ‣ 0：Having received data, without changing remote target, the default value is "0". ‣ 1：Having received data, change remote target once. ‣ 2：Having received data, change remote target. Note： When using <UDP mode>, <UDP local port> must be filled in first.	

Prompted Message(s)	// If the UDP transmission is established, it will prompt message as below
	[<link ID>,] CONNECT
	// If the UDP transmission ends, it will prompt message as below
	[<link ID>,] CLOSED	

Examples	AT+CIPSTART="UDP","192.168.101.110",1000,1002,2	

4.2.4. AT+CIPSEND—Send Data

Designate Command(s)	1. **When making single connection： (+CIPMUX=0)** **AT+CIPSEND=<length>** 2. **When making multiple connection: (+CIPMUX=1)** **AT+CIPSEND=<link ID>,<length>** 3. **If UDP transmission is adopted, remote IP and port and can be installed:** **AT+CIPSEND=[<link ID>,]<length>[,<remote IP>,<remote port>]** **Function: When in normal transmission mode, set length of transmitted data.**
Feedback	Send data of designated length After having received this command, proceed to RETURN >, and then start to receive serial-port data, and when data length XXXXXXX, back to normal command mode, awaiting for the next AT command. If connection is not established, or is terminated, RETURN ERROR If data dispatch has succeeded, then RETURN: SEND OK If data dispatch has failed, then RETURN: SENDFAIL
Parameter Description(s)	• <link ID>：Network connection ID (0-4), used for multiple-connection scenarios • <length>： Digital Parameter indicating data transmission speed, with 2048 maximum. • [<remote IP>]： UDP transmission can install opposite-end IP • [<remote port>]： UDP transmission can set ip opposite port
Examples	-

4.2.5. AT+CIPSENDEX—Send Data

Command(s)	**Designate Command(s) ：** 1. **When making single connection: (+CIPMUX=0)** **AT+CIPSENDEX=<length>** 2. **When making multiple connection ： (+CIPMUX=1)** **AT+CIPSENDEX=<link ID>,<length>** 3. **If UDP transmission is adopted, remote IP and port and can be installed:** **AT+CIPSENDEX=[<link ID>,]<length>[,<remote IP>,<remote port>]** **Function: When in normal transmission mode, set the length of transmitted data**
Feedback	Send data of designated speed After having received this command, proceed to RETURN >, and then start to receive serial-port data, and when data speed XXXXXXX, or encountering character\0, send data. If connection is not established, or is terminated, proceed to RETURN: ERROR If data dispatch has succeeded, then RETURN: SEND OK If data dispatch has failed, then RETURN: SENDFAIL
Parameter Description(s)	• <link ID> ：Network connection ID (0-4), used for multiple-connection scenarios • <length> ：Digital Parameter indicating data transmission length, with 2048 as maximum length • When received data length is out of range or met char "\0", sending data will back to normal command mode, awaiting for the next AT command. If connection is not established, or is terminated, RETURN • If user needs to send "\0", please convert to "\\0".

4.2.6. AT+CIPCLOSE—Terminate TCP/UDP Transmission

Command(s)	Designate Command(s) (used for multiple connection scenarios) AT+CIPCLOSE=<link ID> Function: Terminate TCP/UDP Transmission	Command Execution (used for single connection scenarios) AT+CIPCLOSE
Feedback	OK	
Parameter Description(s)	<link ID>：Need to terminate the connected ID number. When ID is "5", terminate all connections.	
Prompted Message(s)	// When connection ends, it will prompt message as below [<link ID>,] CLOSED	

4.2.7. AT+CIFSR—Inquiry of Local IP Address

Command Execution	AT+CIFSR
Feedback	+CIFSR:STAIP,<Station IP address> +CIFSR:STAMAC,<Station MACaddress> OK
Parameter Description(s)	<IP address>： OPL1000 Station IP Address <MAC address>： OPL1000 Station MAC Address
Note	OP1000 Station IP needs to be connected to AP, before it can be subject to inquiry.

4.2.8. AT+CIPMUX—Set up Multiple Connections

Command(s)	**Inqury of Command(s)：** **AT+CIPMUX?**	**Designate Command(s)：** **AT+CIPMUX=<mode>** **Function: Install Connection Type**
Feedback	+CIPMUX:<mode> OK	OK
Parameter Description(s)	<mode>： ‣ 0： Single Connection Mode ‣ 1： Multiple Connection Mode	
Note	• Single Connection as default • Only through non-penetrating mode (AT+CIPMODE=0), multiple connection can then be installed. • Connection mode can be installed only when no connection has been established. • If TCP server has been established, while wishing to switch to single connection, server must be shut down (AT+CIPSERVER=0), as the server only supports multiple connections.	
Examples	AT+CIPMUX=1	

4.2.9. AT+CIPSERVER—Establish TCP Server

Command(s)	**Inqury of Command(s)：** **AT+CIPSERVER?**	**Designate Command(s)：** **AT+CIPSERVER=<mode>[,<port>]** **Function: Install Server**
Feedback	+CIPSERVER:<mode>,<port> OK	OK
Parameter Description(s)	<mode>： ‣ 0： Shut down server ‣ 1： Establish server [<port>]：Fill in parameters. Default serial-port number is “333”.	

Command(s)	Inqury of Command(s) : AT+CIPSERVER?	Designate Command(s) : AT+CIPSERVER=<mode>[,<port>] Function: Install Server

Prompted Message(s)	// If the connection is established, it will prompt message as below
	[<link ID>,] CONNECT
	// If the connection ends, it will prompt message as below
	[<link ID>,] CLOSED	

Note	• Server can only be activated under multiple connection scenario (AT+CIPUX=1) •After establishing server, server surveilence is automatically established. • When client end enters the connection, it would take up a connecting ID.	
Examples	• Established TCP Server AT+CIPMUX=1 AT+CIPSERVER=1,80	

4.2.10. AT+CIPSTO—Set up TCP Server Over-Time

Command(s)	Inqury of Command(s) : AT+CIPSTO? Function: Inquiry of TCP Server Over-Time	Designate Command(s) : AT+CIPSTO=<time> Function: Install T Install TCP Server Over-Time
Feedback	+CIPSTO:<time> OK	OK
Parameter Description(s)	<time>：TCP server over-time, with values ranging between 0 ~ 7200s.	
Note	•With OPL1000 as TCP server,, it will terminate the TCP client-end connection that is not communicating exceeding over-time period. •If installing AT+CIPSTO=0, over-time will never come into effect, which we do NOT recommend.	
Examples	AT+CIPMUX=1 AT+CIPSERVER=1,1001 AT+CIPSTO=10	

4.2.11. AT+CIPDINFO—Whether Prompt for Opposite-End IP and Port During Reception of network data

Command	AT+CIPDINFO=<mode>
Feedback	OK
Parameter Descripption(s)	<mode>： ‣ 0： Not displaying opposite-end IP and port ‣ 1： Displaying opposite-end IP and port
Examples	AT+CIPDINFO=1

4.2.12. +IPD—Reception of Network Data

Command(s)	When making single connection (+CIPMUX=0)+IPD,<len>[,<remote IP>,<remote port>]:<data>	When making multiple connections, (+CIPMUX=1)+IPD,<link ID>,<len>[,<remote IP>,<remote port>]:<data>
Parameter Description(s)	This command is effective under normal command mode, as when OPL1000 receives network data dispatch +IPD and data to serial-port • [<remote IP>]：Network communication with opposite-end IP, enabled by Command AT+CIPDINFO=1 • [<remote port>]：Network communication with opposite-end port, enabled by Command AT+CIPDINFO=1 • <link ID>： ID number that receives network connection • <len>： Data length • <data>： Received data	

4.2.13. AT+PING—Ping Function

Designate Command(s)	**AT+PING=<IP>** **Function: Ping Function**
Feedback	+PING:<time> OK OR +PING:TIMEOUT ERROR
Parameter Description(s)	• <IP>： String parameter, IP address • <time>：Ping feedback time
Examples	AT+PING="192.168.1.1" AT+PING="*www.baidu.com*"

5. BLE RELATED AT COMMANDS

5.1. List of BLE Commands

Command(s)	Description(s)
AT+BLEINIT	BLE Initiation
AT+BLEADDR	Designate BLE Device Address
AT+BLENAME	Desginate BLE Device Name
AT+BLESCANRSPDATA	Install BLE Scan Response
AT+BLEADVPARAM	Install Broadcast Parameters
AT+BLEADVDATA	Install BLE Broadcast Data
AT+BLEADVSTART	Initiate BLE Broadcast
AT+BLEADVSTOP	Terminate BLE Broadcast
AT+BLECONNPARAM	Renew BLE Connection Parameters
AT+BLEDISCONN	Terminate BLE Connection
AT+BLEDATALEN	Set up BLE Data Packet Length
AT+BLECFGMTU	Set up GATT MTU Length
AT+BLEGATTSSRVCRE	GATTS Service Establishment
AT+BLEGATTSSRVSTART	GATTS Service Activation
AT+BLEGATTSSRVSTOP	GATTS Service Termination
AT+BLEGATTSSRV	GATTS Inquiry Service
AT+BLEGATTSCHAR	GATTS Inquiry Service Attributes
AT+BLEGATTSNTFY	GATTS Service Attribute Value Notification
AT+BLEGATTSIND	GATTS Service Attribute Value Indication
AT+BLEGATTSSETATTR	GATTS Service Attribute Value Installation
AT+BLEGATTCPRIMSRV	GATTC Basic Service Discovery
AT+BLEGATTCINCLSRV	GATTC Included Service Discovery

Command(s)	Description(s)
AT+BLEGATTCINCLSRV	GATTC Included Service Discovery
AT+BLEGATTCCHAR	GATTC Inquiry Service Attributes
AT+BLEGATTCRD	GATTC Service Attribute Value Reading
AT+BLEGATTCWR	GATTC Service Attribute Value Writing
AT+BLEPHYREAD	Reading of Currently Connected PHY
AT+BLEGATTCWR	Installation of Currently Connected PHY

5.2. Description of BLE Commands

5.2.1. AT+BLEINIT—BLE Initiation

Command(s)	**Inqury of Command(s)：** **AT+BLEINIT?** **Function: Inquiry of whether BLE has been intialized**	**Designate Command(s)：** **AT+BLEINIT=<init>** **Function: Installation of the role of BLE initialization.**
Feedback	If BLE has not been initialized, then inquiry RETURNS +BLEINIT:0 OK If BLE has been initialized, then inquiry RETURNS +BLEINIT:<role> OK	OK
Parameter Description(s)	<init>： ‣ 1： client role ‣ 2： server + client role	
Note	•Before using BLE related AT commands, this installation command must be adopted to initialize BLE mode.	
Examples	AT+BLEINIT=1	

5.2.2. AT+BLEADDR—Designate BLE Device Address

Command(s)	Inqury of Command(s)： AT+BLEADDR? Function: Inquiry of the public address of BLE device	Designate Command(s)： AT+BLEADDR=<addr_type>,<random_addr> Function: Install address of BLE device, as currently it only supports installation of random address.
Feedback	+BLEADDR:<BLE_public_addr> OK	OK
Parameter Description(s)	<addr_type>： ‣ 0： public address ‣ 1： random address	
Note	• Currently, the public address can be installed or inquired, and only the installation of random address can be supported. • Before installing BLE public address, Command AT+BLEINIT=1 is needed to complete BLE initiation. •In order for the modified BLE public address to be effective, Command AT+RST is needed to execute reset. • Random address must have the highest 2 bit to be both of “2”, abd for details, please refer to BLE spec.	
Examples	AT+BLEADDR=1,"08:7f:24:87:1c:f7"	

5.2.3. AT+BLENAME—Desginate BLE Device Name

Command(s)	Inqury of Command(s)： AT+BLENAME? FunctionL Inquiry of BLE device name	Designate Command(s)： AT+BLENAME=<device_name> Function: Installl BLE device name.
Feedback	+BLENAME:<device_name> OK	OK

Command(s)	Inqury of Command(s) : AT+BLENAME? FunctionL Inquiry of BLE device name	Designate Command(s) : AT+BLENAME=<device_name> Function: Installl BLE device name.
Parameter Description(s)	<device_name> : BLE device name	
Note	• Default device name is "BLE_AT". • The device name designated by this Command can only be received by the opposite-end device after BLE connection is established, as it actually installs the values of device name characteristic for GAP service, so for details, please refer to BLE core v4.2 vol.3 part C 12.1. • If you need to obtain the device name when scanning for broadcasting packets, then it can only be installed through Command AT+BLEADVDATA.	
Examples	AT+BLENAME="opl_demo"	

5.2.4. AT+BLESCANRSPDATA—Install BLE Scan Response

Command(s)	Designate Command(s) : AT+BLESCANRSPDATA=<scan_rsp_data> Function: Install BLE Scan Feedback.
Feedback	OK
Parameter Description(s)	Scan feedback. Parameters are actually comprises of HEX bytes. For example, installation scan feedback should be "0x11 0x22 0x33 0x44 0x55", as Command can be installed as "AT+BLESCANRSPDATA="1122334455".
Note	The maximum byte supported by scan feedback is 31 bytes.
Examples	AT+BLEINIT=2 // Initialized as server AT+BLESCANRSPDATA="1122334455"

5.2.5. AT+BLEADVPARAM—Set up Broadcast Parameters

Command(s)	**Inqury of Command(s)：** **AT+BLEADVPARAM?** **Function: Inquiry of BLE Broadcast Parameter(s)**	**Designate Command(s)：** **AT+BLEADVPARAM=<adv_int_min>,<adv_int_max>,** **<adv_type>,<own_addr_type>,<channel_map>** **[,<adv_filter_policy>,<peer_addr_type>,<peer_addr>]** **Function: Designate Broadcast Parameter(s)**
Feedback	+BLEADVPARAM:<adv_int_min>,<adv_int_max>,<adv_type>,<own_addr_type>,<channel_map>,<adv_filter_policy>,<peer_addr_type>,<peer_addr> OK	OK
Parameter Description(s)	Minimum Broadcast Interval of values ranging from 0x0020 ~ 0x4000, with 20ms as minimum, and 10240ms as maximum. Maximum Broadcast Interval of values ranging from 0x0020 ~ 0x4000, with 20ms as minimum, and 10240ms as maximum. <adv_type>：Broadcast Types ‣ 0：ADV_TYPE_IND ‣ 1: ADV_TYPE_DIRECT_IND_HIGH ‣ 2：ADV_TYPE_SCAN_IND ‣ 3：ADV_TYPE_NONCONN_IND <own_addr_type>： BLE Address Type ‣ 0：BLE_ADDR_TYPE_PUBLIC ‣ 1：BLE_ADDR_TYPE_RANDOM <channel_map>：Broadcast Channel ‣ 1：ADV_CHNL_37 ‣ 2：ADV_CHNL_38	

Command(s)	Inqury of Command(s)： AT+BLEADVPARAM? Function: Inquiry of BLE Broadcast Parameter(s)	Designate Command(s)： AT+BLEADVPARAM=<adv_int_min>,<adv_int_max>, <adv_type>,<own_addr_type>,<channel_map> [,<adv_filter_policy>,<peer_addr_type>,<peer_addr>] Function: Designate Broadcast Parameter(s)
	‣ 4：ADV_CHNL_39 ‣ 7：ADV_CHNL_ALL [<adv_filter_policy>](Fill-in Parameters): Filter Principle ‣ 0：ADV_FILTER_ALLOW_SCAN_ANY_CON_ANY ‣ 1：ADV_FILTER_ALLOW_SCAN_WLST_CON_ANY ‣ 2：ADV_FILTER_ALLOW_SCAN_ANY_CON_WLST ‣ 3：ADV_FILTER_ALLOW_SCAN_WLST_CON_WLST [<peer_addr_type>] ((Fill-in Parameters): BLE Address Types of the Opposite End ‣ 0：PUBLIC ‣ 1：RANDOM [<peer_addr>](Fill-in Parameters): BLE Address of the Opposite End	
Note	<adv_filter_policy>,<peer_addr_type>,<peer_addr> Three parameters demands default or not at the same time.	
Examples	AT+BLEINIT=2 // Initialized as server AT+BLEADVPARAM=50,50,0,0,4,0,0,"12:34:45:78:66:88"	

5.2.6. AT+BLEADVDATA—Set up BLE Broadcast Data

Command(s)	Designate Command(s)： AT+BLEADVDATA=<adv_data> Function: Install BLE Broadcast Data
Feedback	OK

Command(s)	**Designate Command(s) ：** **AT+BLEADVDATA=<adv_data>** **Function: Install BLE Broadcast Data**
Parameter Description(s)	Broadcast Data Packets. Parameters are actually comprised of HEX bytes. For example, installation scan feedback should be "0x11 0x22 0x33 0x44 0x55", as Command can be installed as "AT+BLEADVDATA ="1122334455".
Note	The maximum length of broadcast packet is 31 bytes.
Examples	AT+BLEINIT=2 // Initialized as server. AT+BLEADVDATA="1122334455"

5.2.7. AT+BLEADVSTART—Initiate BLE Broadcast

Command(s)	**Execute Command：** **AT+BLEADVSTART** **Function: Initiate BLE Broadcast**
Feedback	OK
Parameter Description(s)	N/A
Note	• If Command Broadcast Parameter （AT+BLEADVPARAM=<adv_parameter>）is not installed, default broadcast parameters can be adopted. • If Command Broadcast Parameter（AT+BLEADVPARAM=<adv_data>）is not installed, all packets of "0" are sent.
Examples	AT+BLEINIT=2 // Initialized as server AT+BLEADVSTART

5.2.8. AT+BLEADVSTOP—Terminate BLE Broadcast

Command(s)	**Execute Command：** **AT+BLEADVSTOP** **Function: Terminate Broadcast**
Feedback	OK
Parameter Description(s)	N/A
Note	When broadcast has been intiated, unless BLE connection is successfully establishing, otherwise BLE broadcast will be automatically terminated without having to adopt this Command.
Examples	AT+BLEINIT=2 // Initialized as server AT+BLEADVSTART AT+BLEADVSTOP

5.2.9. AT+BLECONNPARAM—Renew BLE Connection Parameters

Command(s)	**Inqury of Command(s)：** **AT+BLECONNPARAM?** **Function: Inquiry of BLE Connection Parameter(s).**	**Designate Command(s) ：** **AT+BLECONNPARAM=<conn_index>,<min_interval>,<max_interval>,<latency>,<timeout>** **Function: Renew BLE Connection Parameter(s)**
Feedback	+BLECONNPARAM:<conn_index>,<cur_interval>,<latency>,<timeout> OK	OK //Command has been received, and more connection parameters will be updated. +BLECONNPARAM：<conn_index>,0 If update fails, it will be promoted. +BLECONNPARAM：<conn_index>,-1
Parameter Description(s)	<conn_index>： BLE Connection Number currently only supports single connection with index of 0.	

Command(s)	**Inqury of Command(s)：** **AT+BLECONNPARAM?** **Function: Inquiry of BLE Connection Parameter(s).**	**Designate Command(s)：** **AT+BLECONNPARAM=<conn_index>,<min_interval>,<max_interval>,<latency>,<timeout>** **Function: Renew BLE Connection Parameter(s)**
	<min_interval>：The minimum connection interval, with values ranging between 0x0006 ~ 0x0C80. <max_interval>：The maximum connection interval, with values ranging between 0x0006 ~ 0x0C80. <cur_interval>：Current Connecting Interval <latency>：Time delay, with values ranging between 0x0000 ~ 0x01F3. <timeout>：Over-Time, with values ranging between 0x000A ~ 0x0C80.	
Note	This command demands to establish connection first, and only supports BLE client update connection parameters.	
Examples	AT+BLEINIT=1 // Initialized as client AT+BLECONN=0,"24:0a:c4:09:34:23" // Estabblish BLE connection AT+BLECONNPARAM=0,12,14,1,500 // Update BLE connection parameters.	

5.2.10. AT+BLEDISCONN—BLE Connection

Command(s)	**Designate Command(s)：** **AT+BLEDISCONN=<conn_index>** **Function: Terminate BLE Connection**
Feedback	+BLEDISCONN:<conn_index>,<remote_address> OK
Parameter Description(s)	<conn_index>： BLE connection number currently only supports single connection of index of "0". <remote_address>：Opposite-end BLE device address
Examples	AT+BLEINIT=1 // Initialized as client AT+BLECONN=0,"24:0a:c4:09:34:23" // Establish BLE connection

Command(s)	Designate Command(s) ： AT+BLEDISCONN=<conn_index> Function: Terminate BLE Connection
	AT+BLEDISCONN=0 // Terminate BLE connection

5.2.11. AT+BLEDATALEN—Install BLE Data Packet Length

Command(s)	Designate Command(s) ： AT+BLEDATALEN=<conn_index>,<pkt_data_len> Function: Install BLE data packet length
Feedback	OK
Parameter Description(s)	<conn_index>：BLE connection number currently only supports single connection of index of "0". <pkt_data_len>：Data Packet Length, with values randing from 0x001b ~ 0x00fb.
Note	BLE connection needs to be established first, before install packet length.
Examples	AT+BLEINIT=1 // Initialized as client AT+BLECONN=0,"24:0a:c4:09:34:23" AT+BLEDATALEN=0,30

5.2.12. AT+BLECFGMTU—Set up GATT MTU Length

Command(s)	Designate Command(s) ： AT+BLECFGMTU? Function: Inquiry of GATT MTU Length	Designate Command(s) ： AT+BLECFGMTU=<conn_index>,<mtu_size> Function: Install GATT MTU Length
Feedback	+BLECFGMTU:<conn_index>,<mtu_size> OK	OK // Command has been received
Parameter Description(s)	<conn_index>： BLE Connection Number currently only supports single connection with index of 0.	

Command(s)	Designate Command(s) ： AT+BLECFGMTU? Function: Inquiry of GATT MTU Length	Designate Command(s) ： AT+BLECFGMTU=<conn_index>,<mtu_size> Function: Install GATT MTU Length
	<mtu_size> ： The length of the largest transmission unit of BLE	
Note	• The ultimate actul length of MTU needs to be conferred, as installation command returns OK only means there is attempt to confer on MTU, therefore, the installed length may not come into effect, so we recommend that after installation, Inquiry Command AT+BLECFGMTU can be used to inquire on the actual MTU length.	
Examples	AT+BLEINIT=1 // Initialized as client AT+BLECONN=0,"24:12:5f:9d:91:98"// Establish BLE connection AT+BLECFGMTU=0,300	

5.2.13. AT+BLEGATTSSRVCRE—GATTS Service Establishment

Command(s)	Execute Command ： AT+BLEGATTSSRVCRE Function: GATTS Service Establishment
Feedback	OK
Parameter Description(s)	N/A
Note	• After initialization is completed, OPL1000, as a server, should establish service. Once BLE connection is established, service may not be created.
Examples	AT+BLEINIT=2 // Initialized as server AT+BLEGATTSSRVCRE

5.2.14. AT+BLEGATTSSRVSTART—GATTS Service Activation

Command(s)	Execute Command： AT+BLEGATTSSRVSTART Function: GATTS activates all services	Designate Command(s)： AT+BLEGATTSSRVSTART=<srv_index> Function: GATTS activates a designated service.
Feedback	OK	
Parameter Description(s)	N/A	Service serial number, starting in ascending order from "1".
Examples	AT+BLEINIT=2 // Initialized as server AT+BLEGATTSSRVCRE AT+BLEGATTSSRVSTART	

5.2.15. AT+BLEGATTSSRVSTOP—GATTS Service Termination

Command(s)	Execute Command： AT+BLEGATTSSRVSTOP Function: GATTS terminates all services	Designate Command(s)： AT+BLEGATTSSRVSTOP=<srv_index> Function: GATTS terminates a designated service.
Feedback	OK	
Parameter Description(s)	N/A	<srv_index>：Service serial number, starting in ascending order from "1".
Examples	AT+BLEINIT=2 // Initialized as server AT+BLEGATTSSRVCRE AT+BLEGATTSSRVSTART AT+BLEGATTSSRVSTOP	

5.2.16. AT+BLEGATTSSRV—GATTS Discovery Service

Command(s)	**Inqury of Command(s)：** **AT+BLEGATTSSRV?** **Function: GATTS Discovery Service**
Feedback	+BLEGATTSSRV:<srv_index>,<start>,<srv_uuid>,<srv_type> OK
Parameter Description(s)	<srv_index>：Service serial number, starting in ascending order from "1". <start>： ‣ 0：Service not yet activated ‣ 1：Service already activated <srv_uuid>：Service UUID <srv_type>：Service Types ‣ 0：Secondary Service ‣ 1：Primary Service
Examples	AT+BLEINIT=2 // Initialized server AT+BLEGATTSSRVCRE AT+BLEGATTSSRV?

5.2.17. AT+BLEGATTSCHAR—GATTS Discovery Service Attributes

Command(s)	**Inqury of Command(s)：** **AT+BLEGATTSCHAR?** **Function: GATTS Discovery Service Attributes**

Feedback	//Regarding service attribute messages, they area displayed as follows:
	+BLEGATTSCHAR:"char",<srv_index>,<char_index>,<char_uuid>,<char_prop>
	//Regarding descriptor messages, they area displayed as follows:
	+BLEGATTSCHAR:"desc",<srv_index>,<char_index>,<desc_index>
	OK

Command(s)	**Inqury of Command(s)：** **AT+BLEGATTSCHAR?** **Function: GATTS Discovery Service Attributes**
Parameter Description(s)	<srv_index>：Service serial number, starting in ascending order from "1". <char_index>：Service attributes serial number, starting in ascending order from "1". <char_uuid>：Service attributes UUID <char_prop>：Property of Service attributes <desc_index>：Attributes Descriptor Serial Number
Examples	AT+BLEINIT=2 // Initialized as server AT+BLEGATTSSRVCRE AT+BLEGATTSSRVSTART AT+BLEGATTSCHAR?

5.2.18. AT+BLEGATTSNTFY—GATTS Notification Service Attribute Values

Command(s)	**Designate Command(s)：** **AT+BLEGATTSNTFY=<conn_index>,<srv_index>,<char_index>,<length>** **Function: GATTS Notification Service Attribute Values**
Feedback	After having received this command, proceed to RETURN >, and then start to receive serial-port data, and when data length reaches <length>, then execute notification operation. If notification operates successfully, then prompt OK.
Parameter Description(s)	<conn_index>：BLE Connection currently only supports single connection of index of "0". <srv_index>：Service serial numbers can be inquired with Command AT+BLEGATTSCHAR?. <char_index>：Service attributes can be inquired with Command AT+BLEGATTSCHAR?. <length>：Data Length
Examples	Below are the simple examples of "notify"

Command(s)	**Designate Command(s) ：** **AT+BLEGATTSNTFY=<conn_index>,<srv_index>,<char_index>,<length>** **Function: GATTS Notification Service Attribute Values**

	AT+BLEINIT=2 // Initialized as server
	AT+BLEGATTSSRVCRE
	AT+BLEGATTSSRVSTART
	AT+BLEADVSTART// Initiate broadcast, until connection with client, while installed to receive “notify”.
	AT+BLEGATTSCHAR?//Inquiry of allowed “notify” attributes
	//For example, use data of No. 6 attribute notification of No. 3 service with data length of 4
	AT+BLEGATTSNTFY=0,3,6,4
	// After prompting “>” symbol, enter 4-byte data, such as “1234”.

5.2.19. AT+BLEGATTSIND—GATTS Indication Service Attribute Values

Command(s)	**Designate Command(s) ：** **AT+BLEGATTSIND=<conn_index>,<srv_index>,<char_index>,<length>** **Function: GATTS Inidication Service Attribute Values**
Feedback	After having received this command, proceed to RETURN >, and then start to receive serial-port data, and when data length reaches <length>, then execute notification operation. If notification operates successfully, then prompt OK.
Parameter Description(s)	<conn_index> ： BLE Connection currently only supports single connection of index of “0” <srv_index> ：Service attributes can be inquired with Command AT+BLEGATTSCHAR?. <char_index> ：Service attributes can be inquired with Command AT+BLEGATTSCHAR?. <length> ：Data Length

Command(s)	**Designate Command(s) ：** **AT+BLEGATTSIND=<conn_index>,<srv_index>,<char_index>,<length>** **Function: GATTS Inidication Service Attribute Values**

Examples	Below are the simple examples of "indicate"
	AT+BLEINIT=2 // Initialized as server
	AT+BLEGATTSSRVCRE
	AT+BLEGATTSSRVSTART
	AT+BLEADVSTART// Once broadcast is initiated, and wait for client to connect, and once client end has been connected, it would be installed as reception indication.
	AT+BLEGATTSCHAR?// Inquiry of allowed "indicate" attributes
	//For example, use data of No. 7 attribute notification of No. 3 service with data length of 4
	
	AT+BLEGATTSIND=0,3,7,4
	// After prompting ">" symbol, enter 4-byte data, such as "1234".

5.2.20. AT+BLEGATTSSETATTR—GATTS Installation Service Attribute Values

Command(s)	**Designate Command(s) ：** **AT+BLEGATTSSETATTR=<srv_index>,<char_index>[,<desc_index>],<length>** **Function: GATTS Installation Service Attribute (Descriptor) Values**
Feedback	After having received this command, proceed to RETURN >, and then start to receive serial-port data, and when data length reaches <length>, then execute notification operation. If notification operates successfully, then prompt OK.
Parameter Description(s)	<srv_index>：Service discovery result serial numbers can be inquired with Command AT+BLEGATTSCHAR?. <char_index>：Service attributes can be inquired with Command AT+BLEGATTSCHAR?.

Command(s)	**Designate Command(s) ：** **AT+BLEGATTSSETATTR=<srv_index>,<char_index>[,<desc_index>],<length>** **Function: GATTS Installation Service Attribute (Descriptor) Values**
	[<desc_index>](Fill-in Parameter): Attribute Descriptor Serial Number. If filled in, the value of descriptor will be installed. If not filled in, default value will be installed. <length> ：Data Length
Note	<length> cannot exceed the maximum length supported by this attribute (descriptor). For example, this service attribute value is "0x30 0x31", with maximum length of "2", and if the installed <length> being "3" which exceeds the maximum length, ERROR would be prompted.

Examples	AT+BLEINIT=2 // Initialized as server
	AT+BLEGATTSSRVCRE
	AT+BLEGATTSSRVSTART
	AT+BLEGATTSCHAR?
	//For example, use data of No. 1 attribute notification of No. 1 service with data length of 4
	AT+BLEGATTSSETATTR=1,1,,4
	// After prompting ">" symbol, enter 4-byte data, such as "1234".

5.2.21. AT+BLEGATTCPRIMSRV—GATTC Basic Discovery Service

Command(s)	**Designate Command(s) ：** **AT+BLEGATTCPRIMSRV=<conn_index>** **Function: GATTC Discovery Basic Service**
Feedback	+BLEGATTCPRIMSRV:<conn_index>,<srv_index>,<srv_uuid>,<srv_type> OK
Parameter Description(s)	<conn_index> ： BLE Connection currently only supports single connection of index of "0".

Command(s)	**Designate Command(s) ：** **AT+BLEGATTCPRIMSRV=<conn_index>** **Function: GATTC Discovery Basic Service**
	<srv_index>：Service discovery result serial number, starting in ascending order from "1". <srv_uuid>：Service UUID <srv_type>：Service Types ‣ 0：Secondary Service ‣ 1：Primary Service
Note	To adopt this Command, BLE connection needs to be established first.
Examples	AT+BLEINIT=1 // Initialized as client AT+BLECONN=0,"24:12:5f:9d:91:98"// Establish BLE connection AT+BLEGATTCPRIMSRV=0

5.2.22. AT+BLEGATTCINCLSRV—GATTC Discovery Inclusion Service

Command(s)	**Designate Command(s) ：** **AT+BLEGATTCINCLSRV=<conn_index>,<srv_index>** **Function: GATTC Discovery Inclusion Service**
Feedback	+BLEGATTCINCLSRV:<conn_index>,<srv_index>,<srv_uuid>,<srv_type>,<included_srv_uuid>,<included_srv_type> OK
Parameter Description(s)	<conn_index>：BLE Connection currently only supports single connection of index of "0". <srv_index>： Service discovery result serial numbers can be inquired with Command AT+BLEGATTCPRIMSRV=<conn_index>. <srv_uuid>：Service UUID <srv_type>：Service Types ‣ 0：Secondary Service ‣ 1：Primary Service <included_srv_uuid>：UUID Including service

Command(s)	**Designate Command(s) :** **AT+BLEGATTCINCLSRV=<conn_index>,<srv_index>** **Function: GATTC Discovery Inclusion Service**
	<included_srv_type> : Type Including service ‣ 0 : Secondary Service ‣ 1 : Primary Service
Note	To adopt this Command, BLE connection needs to be established first.
Examples	AT+BLEINIT=1 // Initialized as client AT+BLECONN=0,"24:12:5f:9d:91:98"// Establish BLE Connection AT+BLEGATTCPRIMSRV=0 AT+BLEGATTCINCLSRV=0,1//According to the inquiry result of the previous command, designate index for inquiry.

5.2.23. AT+BLEGATTCCHAR—GATTC Discovery Service Attributes

Command(s)	**Designate Command(s) :** **AT+BLEGATTCCHAR=<conn_index>,<srv_index>** **Function: GATTC Discovery Service Attributes**

Feedback	//Regarding service attribute message, and they are displayed as below:
	+BLEGATTCCHAR:"char",<conn_index>,<srv_index>,<char_index>,<char_uuid>,<char_prop>
	//Regarding descriptor messages, and they are displayed as below:
	+BLEGATTCCHAR:"desc",<conn_index>,<srv_index>,<char_index>,<desc_index>,<desc_uuid>
	OK

Parameter Description(s)	<conn_index> : BLE Connection currently only supports single connection of index of "0". <srv_index> : Service discover result serial can be inquired with command AT +BLEGATTCPRIMSRV=<conn_index> <char_index> : Service attributes serial number, starting in ascending order from "1".

Command(s)	**Designate Command(s) ：** **AT+BLEGATTCCHAR=<conn_index>,<srv_index>** **Function: GATTC Discovery Service Attributes**
	<char_uuid>：Service Attribute UUID <char_prop>：Property of Service Attribute Attributes <desc_index>：Attribute Descriptor Serial Number <desc_uuid>：Attribute Descriptor UUID
Note	To adopt this Command, BLE connection needs to be established first.
Examples	AT+BLEINIT=1 //Initialized as client AT+BLECONN=0,"24:12:5f:9d:91:98"// Establish BLE connection AT+BLEGATTCPRIMSRV=0 AT+BLEGATTCCHAR=0,1//According to the inquiry result of the previous command, designate index for inquiry.

5.2.24. AT+BLEGATTCRD—GATTC Read Service Attribute Values

Command(s)	**Designate Command(s) ：** **AT+BLEGATTCRD=<conn_index>,<srv_index>,<char_index>[,<desc_index>]** **Function: GATTC Read Service Attribute (Descriptor) Values**
Feedback	+BLEGATTCRD:<conn_index>,<len>,<value> OK
Parameter Description(s)	<conn_index>：BLE Connection currently only supports single connection of index of "0" <srv_index>：Service discovery result serial numbers can be inquired with Command AT+BLEGATTCPRIMSRV=<conn_index>. <char_index>：Service Attribute serial numbers can be inquired with Command AT+BLEGATTCCHAR=<conn_index>,<srv_index>.

Command(s)	**Designate Command(s) ：** **AT+BLEGATTCRD=<conn_index>,<srv_index>,<char_index>[,<desc_index>]** **Function: GATTC Read Service Attribute (Descriptor) Values**
	[<desc_index>](Fill-in Parameter): Attribute Descriptor Serial Number. If not filled in, default value will be read. If filled in, the value of descriptor will be read. <len>：Data Length <value>： HEX String ‣ If the reading of the value of service attributes is done through Command AT+BLEGATTCRD=<conn_index>,<srv_index>,<char_index>, e.g. Command READ RETURN "+BLEGATTCRD:0,1,30" means that attribute value length is of 1 byte, with content of HEX string “0x30”. ‣ If the reading of the value of service attributes is done through Command AT+BLEGATTCRD=<conn_index>,<srv_index>,<char_index>,<desc_index>, e.g. Command READ RETURN "+BLEGATTCRD:0,4,30313233" means that attribute value length is of 4 bytes, with content of HEX string “0x30 0x31 0x32 0x33”.
Note	• To adopt this Command, BLE connection needs to be established first. • If this service attribute does not support read operation, then command will prompt “ERROR”.
Examples	AT+BLEINIT=1 // Initialized as client Initialized as client AT+BLECONN=0,"24:12:5f:9d:91:98"// Establish BLE Connection AT+BLEGATTCPRIMSRV=0 AT+BLEGATTCCHAR=0,3//According to the inquiry result of the previous command, designate index for inquiry. AT+BLEGATTCRD=0,3,2,1// For example, read the No. 1 descriptor message of the No.2 attribute of the No. 3 service

5.2.25. AT+BLEGATTCWR—GATTC Write Service Attribute Values

Command(s)	**Designate Command(s) ：** **AT+BLEGATTCWR=<conn_index>,<srv_index>,<char_index>[,<desc_index>],<length>** **Function: GATTC Read Service Attribute (Descriptor) Values**
Feedback	After having received this command, proceed to RETURN >, and then start to receive serial-port data, and when data length reaches <length>, then execute read operation. If read operates successfully, then prompt OK.
Parameter Description(s)	<conn_index> ： BLE Connection currently only supports single connection of index of "0" <srv_index> ：Service discovery result serial numbers can be inquired with Command AT+BLEGATTCPRIMSRV=<conn_index>. <char_index> ：Service attributes can be inquired with Command AT+BLEGATTCCHAR=<conn_index>,<srv_index>. [<desc_index>](Fill-in Parameter): Attribute Descriptor Serial Number. If not filled in, default value will be installed. If filled in, the value of descriptor will be installed. <length> ：Data Length
Note	• To adopt this Command, BLE connection needs to be established first. • If this service attribute does not support write operation, then command will prompt "ERROR".
Examples	AT+BLEINIT=1 // Initialized as client AT+BLECONN=0,"24:12:5f:9d:91:98"// Establish BLE Connection AT+BLEGATTCPRIMSRV=0 AT+BLEGATTCCHAR=0,3// According to the inquiry result of the previous command, designate index for inquiry. // For example, use data of No. 4 attribute notification of No. 3 service to write in the data length of 6

Command(s)	**Designate Command(s) ：** **AT+BLEGATTCWR=<conn_index>,<srv_index>,<char_index>[,<desc_index>],<length>** **Function: GATTC Read Service Attribute (Descriptor) Values**
	AT+BLEGATTCWR=0,3,4,,6 // After prompting ">" symbol, enter data of "123456" through serial port

5.2.26. AT+DTM—BLE Testing Mode

Command(s)	**Designate Command(s) ：** **AT+DTM=TX,<channel>,<data length>,<packet type>** **Function: Test BLE TX** **AT+DTM=RX,<channel>** **Function: Test BLE TX** **AT+DTM=END** **Function: Terminate BLE Testing Mode**
Feedback	Execute AT+DTM=END to en BLE Testing Mode, before the result can be displayed. RX CNT: AAA CRC OK: BBB CRC FAIL: CCC RSSI: DDD AAA：Total number of packets received BBB：The correct number of packets from CRC in the meantime. CCC：The incorrect number of packets from CRC in the meantime. DDD：RSSI Value
Parameter Description(s)	<channel>：0~39 <data length>：Packet Length <packet type>：Packet Type ● 0: PRBS9 ● 1: Pattern 11110000

Command(s)	**Designate Command(s)：** **AT+DTM=TX,<channel>,<data length>,<packet type>** **Function: Test BLE TX** **AT+DTM=RX,<channel>** **Function: Test BLE TX** **AT+DTM=END** **Function: Terminate BLE Testing Mode**
	● 2: Pattern 10101010 ● 3: PRBS15 ● 4: Pattern 11111111 ● 5: Pattern 00000000

Examples	// TX-end Device
	AT+DTM=TX, 20, 27, 2
	AT+DTM=END
	//RXXX-end Device
	AT+DTM=RX, 20
	AT+DTM=END

5.2.27. AT+BLEPHYREAD—Read of Currently Connected PHY

Command(s)	**Designate Command(s) ：** **AT+BLEPHYREAD=< conn_index >** **Function: Read of currently connected PHY setting**
Feedback	OK
Parameter Description(s)	<conn_index>： BLE Connection currently only supports single connection of index of "0"
Note	To adopt this Command, BLE connection needs to be established first.
Examples	AT+BLEINIT=2 // Initialized as server AT+BLEADVSTART // Initiate broadcast, and waiting for connection AT+BLEPHYREAD=0 +BLEPHYREAD:0,0,1,1 // <conn_index>,<status>,<phy_tx>,<phy_rx>

Command(s)	Designate Command(s) ： AT+BLEPHYREAD=< conn_index > Function: Read of currently connected PHY setting
Feedback Parameters	Status: 0: 0: Success – 1: Error (no phy_tx, phy_rx) Phy_tx: 1: 1M 2: 2M Phy_rx: 1: 1M 2: 2M

5.2.28. AT+BLEPHYSET—Set up of Currently Connected PHY

Command(s)	Designate Command(s) ： AT+BLEPHYSET=< conn_index >< phy_tx >< phy_rx> Function: Install the currently connected PHY setting
Feedback	OK
Parameter Description(s)	<conn_index> ： BLE Connection currently only supports single connection of index of "0" < phy_tx >: 1: 1M 2: 2M 3: 1M+2M < phy_rx >: 1: 1M 2: 2M 3: 1M+2M
Note	To adopt this Command, BLE connection needs to be established first.
Examples	AT+BLEINIT=2 // Initialized as server AT+BLEADVSTART //Initiate broadcast, and waiting for connection AT+BLEPHYSET=0,2,2 +BLEPHYSET:0,0,2,2 // <conn_index>,<status>,<phy_tx>,<phy_rx>
Feedback Parameters	Status: 0: Success -1: Failure : No phy_tx, phy_rx) Phy_tx: 1: 1M 2: 2M Phy_rx: 1: 1M 2: 2M

5.3. BLE AT CMD Error Code

Code	Description
1	BLE is not initialized

Code	Description
2	The memory is not enough
3	No such command
4	Invalid parameter
5	Invalid state
6	Command is in progress
7	Fail
8	Already (in the wanted state)
9	Wrong role
10	Busy
11	No random address
12	No peer address
13	The number of connections is out of max (only one connection is supported)
14	Service does not start
15	Invalid characteristic property
16	No GATT service
17	No GATT include service
18	No GATT characteristic
19	No GATT characteristic descriptor
20	No read permission
21	No write permission
22	GATT read fail
23	GATT write fail
24	Invalid characteristic value length

6. EXAMPLES OF AT COMMAND APPLICATIONS

This section will introduce several types of examples of common AT command application

6. 1. Single-Connection to TCP Client

1 · Set up Wi-Fi Mode

AT+CWMODE=1 // Station mode

Feedback

OK

2 · Scan AP

AT+CWLAP

Feedback

+CWLAP:2,Opulinks,-22,30:fc:68:90:a8:a1,1

+CWLAP:3,Opulinks-S,-24,00:d0:41:df:1e:25,6

OK

3 · Connection to Router

AT+CWJAP="SSID","password"

Feedback

OK

WIFI CONNECTED

WIFI GOT IP

4 · Inquiry of IP message of device

AT+CIFSR

Feedback

+CIFSR:STAIP, "169.254.119.102"

+CIFSR:STAMAC,"22:33:44:55:66:76"

OK

5 · Set up PC to be connected with OPL000 via the same router, and use network testing assistant on PC to establish UDP transmission:

OPL00 serves as the client end to be connected to TCP server

AT+CIPSTART="TCP","192.2168.1.101",8080 // protocol、server IP & port

Feedback

CONNECT

OK

6 · Send Data

AT+CIPSEND=4

OK

>ABCD

Recv 4 bytes

SEND OK

7 · Data Reception

+IPD,n:xxxxx // received n bytes, data=xxxxx

6. 2. UDP Transmission

1. Set up Wi-Fi Mode
   AT+CWMODE=1 // Station mode
   Feedback
   OK

2. Scan AP
   AT+CWLAP
   Feedback
   +CWLAP:2,Opulinks,-22,30:fc:68:90:a8:a1,1
   +CWLAP:3,Opulinks-S,-24,00:d0:41:df:1e:25,6

   OK
3. Connection to Router
   AT+CWJAP="SSID","password"
   Feedback
   OK

   WIFI CONNECTED
   WIFI GOT IP
4. Inquiry of IP message of device
   AT+CIFSR
   Feedback
   +CIFSR:STAIP, "169.254.119.102"
   +CIFSR:STAMAC,"22:33:44:55:66:76"
   OK

5. Set up PC to be connected with OPL000 via the same router, and use network testing assistant on PC to establish UDP transmission:

Below are the two examples of UDP communication.

6.2.1. Fixed Remote UDP Communication

The fixed remote UDP communication is determined by the last parameter of "0" of AT+CIPSTART command, and a connection number is distributed to the said fixed remote connection, as the message to the remote end will not be altered during the connected period

1 · Enable Multiple Connection
AT+CIPMUX=1
Feedback
OK

2 · Establish UDP Transmission, e.g. Set distribution Connection ID as "4".
AT+CIPSTART=4,"UDP","192.168.1.101",8080,1112,0
Feedback
4,CONNECT
OK

3 · Send Data

AT+CIPSEND=4,5

OK

> ABCDE

Recv 5 bytes

SEND OK

4 · Data Reception

+IPD,n:xxxxx // received n bytes, data=xxxxx

5 · Terminate UDP Transmission

4,CLOSED

OK

6.2.2. Variable Remote UDP Communication

1. Establish UDP Transmission, the last parameter set as "2".

   AT+CIPSTART="UDP","192.168.1.101",8080,1112,2

   Feedback

   CONNECT

   OK

2. Send Data

   AT+CIPSEND=5

   OK

   > ABCDE

   Recv 5 bytes

   SEND OK

3. Data Reception

   +IPD,n:xxxxx // received n bytes, data=xxxxx

4. Terminate UDP Transmission

   0,CLOSED

   OK

6.3. Multiple-Connection TCP Server

OPL000 only supports establishing one TCP server with multiple connections.

As OPL000 can only serve as Station, therefore connection to router needs to be established before setting up the server.

1. Set up Wi-Fi Mode:
   AT+CWMODE=1
   Feedback
   OK

2. Scan AP
   AT+CWLAP
   Feedback
   +CWLAP:2,Opulinks,-22,30:fc:68:90:a8:a1,1
   +CWLAP:3,Opulinks-S,-24,00:d0:41:df:1e:25,6
   OK

3. Connection to Router
   AT+CWJAP="SSID","PASSWD"
   Feedback
   OK

   WIFI CONNECTED
   WIFI GOT IP
4. Check IP Message:

   AT+CIFSR
   Feedback
   +CIFSR:STAIP,"192.168.1.103"
   +CIFSR:STAMAC,"22:33:44:55:66:76"

   OK

5. Enable Multiple-Connection

AT+CIPMUX=1
Feedback
OK

6. TCP SERVER：
   Set up TCP SERVER:
   AT+CIPSERVER=1,8080
   Feedback
   OK
   0,CONNECT // Display after establishing TCP Client Connection on PC

7. Data Reception
   +IPD,0,7,192.168.1.104,54789:abcdefg
   +IPD,0,7,192.168.1.104,54789:abcdefg0,CLOSED

CONTACT

sales@Opulinks.com