

OPL1000

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

AT Instruction Set and Examples



OPULINKS

<http://www.opulinks.com/>

Copyright © 2017-2020, Opulinks. All Rights Reserved.

OPL1000-AT-instruction-set-example | Version 0.42

Date	Version	Contents Updated
2018-04-16	0.1	<ul style="list-style-type: none"> Initial Release
2018-05-18	0.2	<ul style="list-style-type: none"> Add wifi example Add AT+CWAUTOCONN
2018-05-25	0.3	<ul style="list-style-type: none"> Fix some mistakes
2018-05-31	0.4	<ul style="list-style-type: none"> Add tcp server example Fix some mistakes
2018-06-20	0.5	<ul style="list-style-type: none"> Fix some mistakes
2018-06-27	0.6	<ul style="list-style-type: none"> Fixed some wrong description in several AT commands such as AT+BLECFGMTU etc.
2018-06-28	0.7	<ul style="list-style-type: none"> Fix some description mistakes
2018-07-17	0.8	<ul style="list-style-type: none"> Revise CIPSTATUS description Add AT+MACADDRDEF
2018-07-19	0.9	<ul style="list-style-type: none"> Add AT+RFHP Modify AT+BLEADDR—设置 BLE public 设备地址
2018-07-27	0.10	<ul style="list-style-type: none"> Add more description for AT+BLEADDR and AT+CIPSTAMAC
2018-07-27	0.11	<ul style="list-style-type: none"> Add AT+READFLASH, AT+WRITEFLASH and AT+ERASEFLASH
2018-08-02	0.12	<ul style="list-style-type: none"> Add more description for AT+CWMODE and AT+CWLAP
2018-08-17	0.13	<ul style="list-style-type: none"> Add AT+DHCPARPCHK
2018-08-27	0.14	<ul style="list-style-type: none"> Revise AT+CWMODE description
2018-08-27	0.15	<ul style="list-style-type: none"> Update AT+RFHP description
2018-09-04	0.16	<ul style="list-style-type: none"> Update valid range for AT+WRITEFLASH and AT+ERASEFLASH
2018-09-06	0.17	<ul style="list-style-type: none"> Update AT+SWITCHDBG description
2018-09-13	0.18	<ul style="list-style-type: none"> Add AT+MACDATARATE 设置 Wi-Fi Tx data rate
2018-09-13	0.19	<ul style="list-style-type: none"> Add AT+DTM
2018-09-18	0.20	<ul style="list-style-type: none"> Add AT+WIFIMACCFG
2018-09-18	0.21	<ul style="list-style-type: none"> Update AT+GSLP description

Date	Version	Contents Updated
2018-09-18	0.22	<ul style="list-style-type: none"> Update AT+CWAUTOCONN description
2018-09-18	0.23	<ul style="list-style-type: none"> Update AT+ BLEADVPARAM description
2018-09-19	0.24	<ul style="list-style-type: none"> Add AT+CWFASTCONN
2018-09-20	0.25	<ul style="list-style-type: none"> Add AT+RFTM
2018-09-20	0.26	<ul style="list-style-type: none"> Add AT+BLEPHYREAD Add AT+BLEPHYSET
2018-09-20	0.27	<ul style="list-style-type: none"> Update AT+CIPSTAMAC description
2018-09-20	0.28	<ul style="list-style-type: none"> Add RF-related AT commands
2018-10-05	0.29	<ul style="list-style-type: none"> Update AT+CWJAP description
2018-11-21	0.30	<ul style="list-style-type: none"> Add AT+RFTM (BeaconOnlyMode)
2018-12-24	0.31	<ul style="list-style-type: none"> OPL1000 acts as TCP server to send data
2019-03-12	0.32	<ul style="list-style-type: none"> Remove SSL support
2019-03-13	0.33	<ul style="list-style-type: none"> Support Transparent mode Update AT+CIPSTART Add AT+CIPMODE Update AT+CIPSEND Add AT transparent mode example
2019-03-19	0.34	<ul style="list-style-type: none"> Add AT+SAVETRANSLINK command
2019-05-17	0.35	<ul style="list-style-type: none"> Add new parameter for AT+CWLAPOPT
2019-10-02	0.36	<ul style="list-style-type: none"> Add WiFi HP boost 2db for AT+RFHP
2019-10-17	0.37	<ul style="list-style-type: none"> Add SMPS_RF tuning for AT+RFTM
2019-11-04	0.38	<ul style="list-style-type: none"> Add description for Fast connect
2019-11-05	0.39	<ul style="list-style-type: none"> Add WiFi LP boost 2db and 3db option for AT+RFHP
2020-01-02	0.40	<ul style="list-style-type: none"> Add commands for AUXADC
2020-03-03	0.41	<ul style="list-style-type: none"> Update AUXADC calibration
2021-12-01	0.42	<ul style="list-style-type: none"> Update supported UART Baud Rate

TABLE OF CONTENTS

1. 介绍	6
1.1. 文档应用范围	6
1.2. 指令说明	6
2. 基础 AT 指令	7
2.1. 基础 AT 指令一览表	7
2.2. 基础 AT 指令描述	8
2.2.1. AT—测试 AT 模块	8
2.2.2. AT+RST—重启模块	8
2.2.3. AT+GMR—查询版本信息	8
2.2.4. AT+GSLP—进入 Deep-sleep 模式	9
2.2.5. ATE—开关回显功能	9
2.2.6. AT+RESTORE—恢复出厂设置	9
2.2.7. AT+UART_CUR—设置 UART 临时配置	9
2.2.8. AT+UART_DEF—设置 UART 配置，保存到 Flash	11
2.2.9. AT+SLEEP—设置 sleep 模式	12
2.2.10. AT+SYSRAM—查询当前剩余 RAM 大小	13
2.2.11. AT+MACADDRDEF—设置设备来源 Mac Address，保存到 flash	14
2.2.12. AT+RFHP—设置 RF Power	15
2.2.13. AT+READFLASH—读取 Flash	15
2.2.14. AT+WRITEFLASH—写入 Flash	16
2.2.15. AT+ERASEFLASH—抹除 Flash	17
2.2.16. AT+DHCPARPCHK—设置 DHCP ARP 检查机制	18
2.2.17. AT+SWITCHDBG—切换至 Debug UART	18
2.2.18. AT+MACDATARATE—设置 Wi-Fi Tx data rate	20
2.2.19. AT+WIFIMACCFG—设置 Wi-Fi 相关参数	21
2.2.20. AT+RFTM—设置 Test Mode	21
2.2.21. AT+MODE—设定 WiFi 模式	22
2.2.22. AT+GO—设定 WiFi 封包与速度	22
2.2.23. AT+CHANNEL—设定 WiFi 通道	23
2.2.24. AT+RESET_CNTS—清除 WiFi Rx 统计量	24
2.2.25. AT+COUNTERS—读取 WiFi Rx 统计量	24
2.2.26. AT+TX—启动/关闭 WiFi Tx 测试	25
2.2.27. AT+RX—启动/关闭 WiFi Rx 测试	25
2.2.28. AT+DTM—设定与开始 BLE Tx/Rx 测试	25

2.2.29.	AT+AUXADC—讀取 AUXADC 原始資料	27
2.2.30.	AT+ADCCALVBAT—校準 VBAT	28
2.2.31.	AT+ADCCALGPIO—校準 GPIO	29
2.2.32.	AT+ADCDEF—恢復校準預設值	29
2.2.33.	AT+ADCERASE—清除 flash 校準設定	30
2.2.34.	AT+ADCSTORE—儲存校準設定到 flash	30
2.2.35.	AT+ADCRELOAD—重新讀取 flash 的校準值	30
2.2.36.	AT+ADCVBAT—查詢校準後 VBAT 電壓	31
2.2.37.	AT+ADCGPIO—查詢校準後 GPIO 電壓	31
3.	WIFI 功能 AT 指令	33
3.1.	WIFI 功能 AT 指令一覽表	33
3.2.	WIFI 功能 AT 指令描述	34
3.2.1.	AT+CWMODE—設置 Wi-Fi 模式	34
3.2.2.	AT+CWJAP—連接 AP	35
3.2.3.	AT+CWLAPOPT—設置 CWLAP 指令的屬性	36
3.2.4.	AT+CWLAP—掃描當前可用的 AP	37
3.2.5.	AT+CWQAP—斷開與 AP 的連接	37
3.2.6.	AT+CWAUTOCONN—上電是否自動連接 AP	38
3.2.7.	AT+CWFASTCONN—快速連接 AP	38
3.2.8.	AT+CIPSTAMAC—設置 OPL1000 Station 接口的 MAC 地址	39
3.2.9.	AT+CWHOSTNAME—設置 Station 的主機名稱	40
4.	TCP/IP 功能 AT 指令	41
4.1.	TCP/IP 功能 AT 指令一覽表	41
4.2.	TCP/IP 功能 AT 指令描述	41
4.2.1.	AT+CIPSTATUS—查詢網絡連接信息	41
4.2.2.	AT+CIPDOMAIN—域名解析功能	42
4.2.3.	AT+CIPSTART—建立 TCP 連接或 UDP 傳輸	43
4.2.4.	AT+CIPSEND—發送數據	45
4.2.5.	AT+CIPSENDEX—發送數據	46
4.2.6.	AT+CIPCLOSE—關閉 TCP/UDP 傳輸	46
4.2.7.	AT+CIFSR—查詢本地 IP 地址	47
4.2.8.	AT+CIPMUX—設置多連接	47
4.2.9.	AT+CIPSERVER—建立 TCP 服務器	48
4.2.10.	AT+CIPMODE—設置透傳模式	49
4.2.11.	AT+SAVETRANSLINK—保存透傳至 FLASH	49
4.2.12.	AT+CIPSTO—設置 TCP 服務器超時時間	50
4.2.13.	AT+CIPDINFO—接收網絡數據時是否提示對端 IP 和端口	51

4.2.14.	+IPD—接收网络数据	51
4.2.15.	AT+PING—Ping 功能	52
5.	BLE 相关 AT 指令	53
5.1.	BLE 指令一览表	53
5.2.	BLE 指令描述	54
5.2.1.	AT+BLEINIT—BLE 初始化	54
5.2.2.	AT+BLEADDR—设置 BLE 设备地址	55
5.2.3.	AT+BLENAME—设置 BLE 设备名称	55
5.2.4.	AT+BLESCANRSPDATA—设置 BLE 扫描响应	56
5.2.5.	AT+BLEADVPARAM—设置广播参数	56
5.2.6.	AT+BLEADVDATA—设置 BLE 广播数据	58
5.2.7.	AT+BLEADVSTART—开始 BLE 广播	58
5.2.8.	AT+BLEADVSTOP—结束 BLE 广播	59
5.2.9.	AT+BLECONNPARAM—更新 BLE 连接参数	59
5.2.10.	AT+BLEDISCONN—断开 BLE 连接	60
5.2.11.	AT+BLEDATALEN—设置 BLE 数据包长度	61
5.2.12.	AT+BLECFGMTU—设置 GATT MTU 的长度	61
5.2.13.	AT+BLEGATTSSRVCRE—GATTS 创建服务	62
5.2.14.	AT+BLEGATTSSRVSTART—GATTS 开启服务	62
5.2.15.	AT+BLEGATTSSRVSTOP—GATTS 停止服务	62
5.2.16.	AT+BLEGATTSSRV—GATTS 发现服务	63
5.2.17.	AT+BLEGATTSSCHAR—GATTS 发现服务特征	64
5.2.18.	AT+BLEGATTSENTFY—GATTS 通知服务特征值	64
5.2.19.	AT+BLEGATTSSIND—GATTS 指示服务特征值	65
5.2.20.	AT+BLEGATTSSSETATTR—GATTS 设置服务特征值	66
5.2.21.	AT+BLEGATTCPRIMSRV—GATTC 发现基本服务	67
5.2.22.	AT+BLEGATTCINCLSRV—GATTC 发现包含服务	67
5.2.23.	AT+BLEGATTCCHAR—GATTC 发现服务特征	68
5.2.24.	AT+BLEGATTCRD—GATTC 读取服务特征值	69
5.2.25.	AT+BLEGATTCWR—GATTC 写服务特征值	70
5.2.26.	AT+DTM—BLE 测试模式	72
5.2.27.	AT+BLEPHYREAD—读取目前連線的 PHY	73
5.2.28.	AT+BLEPHYSET—设置目前連線的 PHY	73
5.3.	BLE AT CMD Error Code	74
6.	AT 指令使用示例	76
6.1.	单连接 TCP 客户端	76
6.1.1.	建立连接	76

6.1.2. OPL1000 作为 tcp client 发送数据 77

6.1.3. OPL1000 作为 tcp client 接收数据 78

6.2. 单连接 TCP 服务器 79

6.2.1. 建立连接 79

6.2.2. OPL1000 作为 tcp server 发送数据 81

6.2.3. OPL1000 作为 tcp server 接收数据 82

6.3. UDP 传输 83

6.3.1. 固定远端的 UDP 通信 84

6.3.2. 远端可变的 UDP 通信 85

6.4. 多连接 TCP 服务器 86

6.4.1. 建立连接 86

6.4.2. OPL1000 作为 tcp server 发送数据 89

6.4.3. OPL1000 作为 tcp server 接收数据 91

6.5. Wi-Fi 透传模式 92

6.5.1. TCP 客户端单连接透传模式 92

6.5.2. UDP 透传模式 95

6.6. AUXADC 校准 98

6.6.1. 適用限制 98

6.6.2. 預設校准 98

6.6.3. 輸入電壓校准 98

LIST OF FIGURES

FIGURE 1: 网络调试助手建立 TCP 服务器.....77

FIGURE 2: TCP SERVER 接收数据78

FIGURE 3: 发送数据框78

FIGURE 4 : OPL1000 接收数据.....78

FIGURE 5: 客户端连接到 TCP 服务器.....80

FIGURE 6: 客户端接收数据81

FIGURE 7: TCP CLIENT 端发送数据82

FIGURE 8: 接收数据.....82

FIGURE 9: 多个客户端连接服务器的网络连接图87

FIGURE 10: 网络调试助手设置参数界面.....88

FIGURE 11: 连接界面按钮点击连接88

FIGURE 12: 连接成功.....89

FIGURE 13: CLIENTA 数据接收.....90

FIGURE 14: CLIENTB 数据接收.....90

FIGURE 15: 发送数据.....91

FIGURE 16: OPL1000 接收数据.....91

1. 介绍

1.1. 文档应用范围

本文档描述 OPL1000 AT 指令集功能以及使用方法。

指令集主要分为：基础 AT 指令、WIFI 功能 AT 指令、TCP/IP 功能 AT 指令、BLE 功能 AT 指令等。

OPL1000 AT 指令默认使用串口 UART1 传输，默认波特率为 115200bps，格式为 8N1。

1.2. 指令说明

AT 指令可以细分为四种类型：

类型	指令格式	描述
测试指令	AT+<x>=?	该命令用于查询设置指令的参数以及取值范围。
查询指令	AT+<x>?	该命令用于返回参数的当前值。
设置指令	AT+<x>=<...>	该命令用于设置用户自定义的参数值。
执行指令	AT+<x>	该命令用于执行受模块内部程序控制的变参数的功能。

注意：

- 不是每条 AT 指令都具备上述 4 种类型的命令。
- [] 括号内为缺省值，可以不填写或者可能不显示。
- AT 指令不区分大小写。
- AT 指令以回车换行符结尾 \r\n。请注意设置串口工具为“新行模式”。

2. 基础 AT 指令

2.1. 基础 AT 指令一览表

指令	说明
AT	测试 AT 模块
AT+RST	重启模块
AT+GMR	查看版本信息
AT+GSLP	进入 Deep-Sleep 模式
ATE	开关回显功能
AT+RESTORE	恢复出厂设置
AT+UART_CUR	设置 UART 当前临时配置
AT+UART_DEF	设置 UART 配置，保存到 flash
AT+SLEEP	设置 Sleep 模式
AT+SYSRAM	查询当前剩余 RAM 大小
AT+MACADDRDEF	设置设备来源 Mac Address，保存到 flash
AT+RFHP	设置 RF Power
AT+READFLASH	读取 flash
AT+WRITEFLASH	写入 flash
AT+ERASEFLASH	抹除 flash
AT+DHCPARPCHK	设置 DHCP ARP 检查机制
AT+MACDATARATE	设置 Wi-Fi Tx data rate
AT+WIFIMACCFG	设置 Wi-Fi Tx 相关参数

2.2. 基础 AT 指令描述

2.2.1. AT—测试 AT 模块

执行指令	AT
响应	OK
参数说明	-

2.2.2. AT+RST—重启模块

执行指令	AT+RST
响应	OK
参数说明	-
注意	执行此指令后，系统会强制重启。

2.2.3. AT+GMR—查询版本信息

执行指令	AT+GMR
响应	<AT version info> <SDK version info> <compile time> OK
参数说明	• <AT version info>：AT 版本信息 • <SDK version info>：SDK 版本信息 • <compile time>：编译生成时间

2.2.4. AT+GSLP—进入 Deep-sleep 模式

设置指令	AT+GSLP=<Duration>, <I/O>
响应	<Duration> OK
参数说明	<Duration> : 设置 OPL1000 的睡眠时长，单位：毫秒。OPL1000 会在休眠设定时长后自动唤醒。 <I/O> GPIO number to Wakeup

2.2.5. ATE—开关回显功能

执行指令	ATE
响应	OK
参数说明	• ATE0 : 关闭回显 • ATE1 : 开启回显

2.2.6. AT+RESTORE—恢复出厂设置

执行指令	AT+RESTORE
响应	OK
注意	恢复出厂设置，将擦除所有保存到 Flash 的参数，恢复为默认参数。 恢复出厂设置会导致机器重启。

2.2.7. AT+UART_CUR—设置 UART 临时配置

指令	查询指令： AT+UART_CUR?	设置指令： AT+UART_CUR=<baudrate>,<databits> >,<stopbits>,<parity>,<flow control>
响应	+UART_CUR:<baudrate>,<databits>,<stopbits>,<parity>,<flow control> OK	

指令	查询指令： AT+UART_CUR?	设置指令： AT+UART_CUR=<baudrate>,<databits>,<stopbits>,<parity>,<flow control>
	OK 查询返回的是 UART 实际参数值，由于时钟分频的原因，UART 实际参数值与设置值有一定误差，是正常现象。	
参数说明	<ul style="list-style-type: none">• <baudrate>：UART 波特率<ul style="list-style-type: none">▸ 5：5 bit 数据位▸ 6：6 bit 数据位▸ 7：7 bit 数据位▸ 8：8 bit 数据位• <stopbits>：停止位<ul style="list-style-type: none">▸ 1：1 bit 停止位▸ 2：1.5 bit 停止位▸ 3：2 bit 停止位• <parity>：校验位<ul style="list-style-type: none">▸ 0：None▸ 1：Odd▸ 2：Even• <flow control>：流控<ul style="list-style-type: none">▸ 0：不使能流控▸ 1：保留▸ 2：保留▸ 3：同时使能 RTS 和 CTS	
注意	<ul style="list-style-type: none">• 本设置不保存到 flash。• 使用流控需要硬件支持。• 波特率支持范围：9600, 14400, 19200, 38400, 57600, 115200, 230400 and 460800	
示例	AT+UART_CUR=115200,8,1,0,3	

2.2.8. AT+UART_DEF—设置 UART 配置 · 保存到 Flash

指令	查询指令： AT+UART_DEF?	设置指令： AT+UART_DEF=<baudrate>,<databits>,<stopbits>,<parity>,<flow control>
响应	+UART_DEF:<baudrate>,<data bits>,<stopbits>,<parity>,<flow control> OK	
参数说明	<ul style="list-style-type: none">• <baudrate>：UART 波特率• <databits>：数据位<ul style="list-style-type: none">▸ 5：5 bit 数据位▸ 6：6 bit 数据位▸ 7：7 bit 数据位▸ 8：8 bit 数据位• <stopbits>：停止位<ul style="list-style-type: none">▸ 1：1 bit 停止位▸ 2：1.5 bit 停止位▸ 3：2 bit 停止位• <parity>：校验位<ul style="list-style-type: none">▸ 0：None▸ 1：Odd▸ 2：Even• <flow control>：流控<ul style="list-style-type: none">▸ 0：不使能流控▸ 1：保留▸ 2：保留▸ 3：同时使能 RTS 和 CTS	
注意	<ul style="list-style-type: none">• 本设置将保存在到 flash · 重新上电后仍生效。• 使用流控需要硬件支持。• 波特率支持范围：9600, 14400, 19200, 38400, 57600, 115200, 230400 and 460800	

指令	查询指令： AT+UART_DEF?	设置指令： AT+UART_DEF=<baudrate>,<databits>,<stopbits>,<parity>,<flow control>
示例	AT+UART_DEF=115200,8,1,0,3	

2.2.9. AT+SLEEP—设置 sleep 模式

指令	设置指令： <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>
响应	OK
参数说明	<sleep mode> : ▸ 0：禁用 Smart-sleep 模式 ▸ 1：启用 Smart-sleep 模式 <I/O> GPIO number to Wakeup ▸ 2：进入 Timer-sleep 模式 <Duration>：设置 OPL1000 的睡眠时长，单位：毫秒。OPL1000 会在休眠设定时长后自动唤醒。 <I/O> GPIO number to Wakeup ▸ 3：进入 Deep-sleep 模式 <I/O> GPIO number to Wakeup
示例	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—查询当前剩余 RAM 大小

查询指令	AT+SYSRAM?
响应	+SYSRAM:<remaining RAM size> OK
参数说明	<remaining RAM size> : 当前剩余 RAM 大小 · 单位 : 字节
示例	AT+SYSRAM? +SYSRAM:148408 OK

2.2.11. AT+MACADDRDEF—设置设备来源 Mac Address · 保存到 flash

指令	查询指令： AT+MACADDRDEF?	设置指令： AT+MACADDRDEF=<iface>,<type>
响应	+MACADDRDEF:<iface_0>,<type> OK <iface_1>,<type> OK	
参数说明	<ul style="list-style-type: none">• <iface>：设备介面<ul style="list-style-type: none">▸ 0：Wifi Station▸ 1：BLE• <type>：MAC address 来源设置<ul style="list-style-type: none">▸ 0：From OTP▸ 1：From Flash	
注意	<ul style="list-style-type: none">• 本设置将保存在到 flash · 重新上电后仍生效。• 默认配置来源 MAC address 为 OTP。• 設置成功後，需重新上電使 MAC address 配置生效	
示例	AT+MACADDRDEF=0,1	

2.2.12. AT+RFHP—设置 RF Power

指令	查询指令： AT+RFHP?	设置指令： AT+RFHP=<rf_power>
响应	+RFHP:<rf_power> OK	OK
参数说明	<ul style="list-style-type: none"> • <rf_power> : RF 功率 <ul style="list-style-type: none"> ▸ 0 (0x00): Wi-Fi LPA and BLE LPA ▸ 15 (0x0F): Wi-Fi LPA and BLE HPA ▸ 32 (0x20): Wi-Fi LPA boost 2db and BLE LPA ▸ 176 (0xB0): Wi-Fi HPA and BLE LPA ▸ 208 (0xD0): Wi-Fi HPA boost 2db and BLE LPA ▸ 224 (0xE0) : Wi-Fi HPA boost 3db and BLE LPA ▸ 255 (0xFF): Wi-Fi HPA and BLE HPA 	
注意	<ul style="list-style-type: none"> • 本设置将保存在到 flash，重新上电后仍生效。 • 默认配置 176 : Wi-Fi HPA and BLE LPA。 • 設置成功後，需重新上電使配置生效 	
示例	AT+RFHP=176	

2.2.13. AT+READFLASH—读取 Flash

指令	查询指令： N/A	设置指令： AT+READFLASH=<address>,<number_of_bytes> >
响应	N/A	成功： <byte_0>,<byte_1>,<byte_2>,...,<byte_n> OK 失败：ERROR
参数说明	<ul style="list-style-type: none"> • <address> : Flash 位址 (十六進制)。 ▸ 允許讀取範圍為 0x00000000 ~ 0x000FFFFFF • <number_of_bytes> : 欲讀取的 byte 數量 (十進制)。 	

指令	查詢指令： N/A	設置指令： AT+READFLASH=<address>,<number_of_bytes> >
注意	• 不可超出允許範圍。	
示例	AT+READFLASH=0x00088000,8 FF,FF,FF,FF,FF,FF,FF,FF OK	

2.2.14. AT+WRITEFLASH—写入 Flash

指令	查詢指令： N/A	設置指令： AT+WRITEFLASH=<address>,<number_of_bytes>,<byte_0>,<byte_1>,...,<byte_n>
响应	N/A	成功：OK 失敗：ERROR
参数说明	<ul style="list-style-type: none"> • <address>：Flash 位址（十六進制）。 • 允許寫入範圍為 0x00000000 ~ 0x000FFFFF • <number_of_bytes>：欲寫入的 byte 數量（十進制）。 • <byte_0>~<byte_n>：写入值（十六進制）。 	
注意	<ul style="list-style-type: none"> • 不可超出允許範圍。 • 寫入前建議請先執行 AT+READFLASH，確認欲寫入的 Flash 範圍均處於可寫入狀態（全部值均為 0xFF）；若不是，請先執行 AT+ERASEFLASH 抹除包含欲寫入範圍的 sector。 • 由於 AT command 最多能輸入 255 個字元，依示例格式，一次最多可分別指定寫入 76 bytes。 • 若 <byte_0>~<byte_n> 少於 <number_of_bytes> 所指定的數量，會持續寫入 <byte_n> 直到寫滿 <number_of_bytes> 為止，可藉此方式將指定範圍全部寫為同一個值。 	
示例	1. AT+WRITEFLASH=0x00088000,4,F0,F1,F2,F3 OK AT+READFLASH=0x00088000,8 F0,F1,F2,F3,FF,FF,FF,FF OK	

指令	查询指令： N/A	设置指令： AT+WRITEFLASH=<address>,<number_of_bytes>,<byte_0>,<byte_1>,...,<byte_n>
----	--------------	---

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—抹除 Flash

指令	查询指令： N/A	设置指令： AT+ERASEFLASH=<start_address_of_sector>,<number_of_sectors>
响应	N/A	成功：OK 失败：ERROR

参数说明

- <start_address_of_sector>：sector 起始位址（十六进制）。
 - 允许值为 0x00000000, 0x00001000, 0x00002000, ... , 0x0000FE000, 或 0x000FF000。
- <number_of_sectors>：欲抹除的 sector 数量（十进制）。
 - 指定从 sector 起始位址开始，要一次抹除多少 sectors。
 - 单一 sector 大小为 4096 bytes。

注意

- 以 sector 为单位抹除 Flash，一次最少抹除 4096 bytes。

示例

1.
 抹除 0x00089000 ~ 0x00089FFF：4096 bytes
 AT+ERASEFLASH=0x00089000,1
 OK

2.
 抹除 0x00088000 ~ 0x00089FFF：8192 bytes
 AT+ERASEFLASH=0x00088000,2

指令	查询指令： N/A	设置指令： AT+ERASEFLASH= <start_address_of_sector>,<number_of_sectors>
	OK	

2.2.16. AT+DHCPARPCHK—设置 DHCP ARP 检查机制

指令	查询指令： AT+DHCPARPCHK?	设置指令： AT+DHCPARPCHK= <chk_mode>
响应	+DHCPARPCHK:<chk_mode> OK	OK
参数说明	• < chk_mode > : 0 : Disable 1 : Enable	
注意	• 本设置将保存在到 flash，重新上电后仍生效。 • 默认配置为 Enable。 • 设置完后无需重新上电，重新断线连线即生效。	
示例	AT+DHCPARPCHK=0 OK	

2.2.17. AT+SWITCHDBG—切换至 Debug UART

指令	查询指令： N/A	设置指令： AT+SWITCHDBG
响应	N/A	Switch: Dbg UART
参数说明	-	
注意	• 设置后会将 AT UART 与 Debug UART 的 IO pin 互换 • 切换 IO 时，UART 会收到几个无意义的字	
示例	>AT+SWITCHDBG C! Switch: Dbg UART	

2.2.18. AT+MACDATARATE—设置 Wi-Fi Tx data rate

指令	查询指令： AT+MACDATARATE?	设置指令： AT+MACDATARATE=<data_rate_id>
响应	+MACDATARATE:<data_rate_id> OK > OK	
参数说明	-<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	
注意	•本设置将保存在到 flash，重新上电后仍生效。 •默认配置为 Auto Rate Adaptation。 •设置完后即生效。	
示例	>AT+MACDATARATE=4 OK	

2.2.19. AT+WIFIMACCFG—设置 Wi-Fi 相关参数

指令	查询指令： AT+WIFIMACCFG?	设置指令： AT+WIFIMACCFG=<cfg_id>,<value>
响应	+WIFIMACCFG:<cfg_id>,<value> OK > OK	
参数说明	<p>-<cfg_id>：</p> <p>0: skip DTIM · 意思是跳过多少个 DTIM 包，范围为 0–255，设置 0 表示不跳过 DTIM。 设置 5 表示跳过 5 个 DTIM 包。</p> <p>-<value>：设置的值</p>	
注意	<ul style="list-style-type: none">• 本设置将保存在到 flash · 重新上电后仍生效。• 设置完后即生效。• 若设置过多 skip DTIM 可能会导致吞吐量下降。	
示例	>AT+WIFIMACCFG=0,5 OK	

2.2.20. AT+RFTM—设置 Test Mode

指令	查询指令： N/A	设置指令： AT+RFTM=<mode>,<param...>
响应	N/A	OK
参数说明	<p>• <mode>：RF 测试模式</p> <ul style="list-style-type: none">▸ 1： RF TX Single-Tone<ul style="list-style-type: none">▸ AT+RFTM=<mode>,<Frequency>▸ 2： MAC RX Beacon only mode<ul style="list-style-type: none">▸ AT+RFTM=2,<OnCtrl><OnCtrl> Setting:<ul style="list-style-type: none">▸ 0： Normal mode (Default)▸ 1： MAC only receive Beacon, no Tx▸ 4： Setup SMPS_RF to 1.4V	

指令	查询指令： N/A	设置指令： AT+RFTM=<mode>, <param...>
	▶AT+RFTM=4, <OnCtrl> <OnCtrl> Setting: ▶ 0 : Normal mode (Default) ▶ 1 : Setup SMPS_RF to 1.4V	
注意	•	
示例	AT+RFTM=1, 2442	

2.2.21. AT+MODE—设定 WiFi 模式

指令	查询指令： N/A	设置指令： AT+MODE=<mode>
响应	N/A	OK
参数说明	• <mode> : WiFi 测试模式，请固定使用 3。	
注意	•	
示例	AT+MODE=3 Mode is RF OK	

2.2.22. AT+GO—设定 WiFi 封包與速度

指令	查询指令： N/A	设置指令： AT+GO=<preamble>,<data_length>,<interval>,<data_rate>,<packet_count>
响应	N/A	OK
参数说明	• <preamble>: 1: long Others for short	

指令	查询指令： N/A	设置指令： AT+GO=<preamble>,<data_length>,<interval>,<data_rate>,<packet_count>
	<ul style="list-style-type: none"> • <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 	
注意	•	
示例	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—设定 WiFi 通道

指令	查询指令： N/A	设置指令： AT+CHANNEL=<channel>
响应	N/A	OK
参数说明	• <channel> : WiFi 通道 · 范围为 1~14 ·	
注意	•	
示例	AT+CHANNEL=7	

指令	查询指令 :	设置指令 :
	N/A	AT+CHANNEL= <channel>
	99, 7	
	OK	

2.2.24. AT+RESET_CNTS—清除 WiFi Rx 统计量

指令	查询指令 :	设置指令 :
	N/A	AT+RESET_CNTS
响应	N/A	OK
参数说明	N/A	
注意	•	
示例	AT+RESET_CNTS OK	

2.2.25. AT+COUNTERS—读取 WiFi Rx 统计量

指令	查询指令 :	设置指令 :
	AT+COUNTERS?	N/A
响应	OK	N/A
参数说明	N/A	
注意	•	
示例	AT+COUNTERS? ok: 70558, err: 3836, rssi: -38 OK	

2.2.26. AT+TX—启动/关闭 WiFi Tx 测试

指令	查询指令： N/A	设置指令： AT+TX=<mode>
响应	N/A	OK
参数说明	• <mode> : 1: 启动 WiFi Tx 测试 0: 关闭 WiFi Tx 测试	
注意	•	
示例	AT+TX=1 OK	

2.2.27. AT+RX—启动/关闭 WiFi Rx 测试

指令	查询指令： N/A	设置指令： AT+RX=<mode>
响应	N/A	OK
参数说明	• <mode> : 1: 启动 WiFi Rx 测试 0: 关闭 WiFi Rx 测试	
注意	•	
示例	AT+RX=1 OK	

2.2.28. AT+DTM—设定与开始 BLE Tx/Rx 测试

指令	查询指令： N/A	设置指令： AT+DTM=<mode>,<channel>,<data_length>,<packet_type>
响应	N/A	OK

指令	查询指令： N/A	设置指令： AT+DTM=<mode>,<channel>,<data_length>,<packet_type>
参数说明	<ul style="list-style-type: none">• <mode> :<ul style="list-style-type: none">tx: BLE Tx 测试rx: BLE Rx 测试end: 结束 BLE Tx/Rx 测试• <channel> :<ul style="list-style-type: none">0 ~ 39• <data_length> :<ul style="list-style-type: none">n bytes• <packet_type> :<ul style="list-style-type: none">0: PRBS91: Pattern 111100002: Pattern 101010103: PRBS154: Pattern 111111115: Pattern 00000000	
注意	<ul style="list-style-type: none">•	
示例	<ul style="list-style-type: none">• 开始 BLE Tx 测试<ul style="list-style-type: none">AT+DTM=tx,20,30,2Start DTM TxFrequency: 20, length: 30, type: 2OK• 结束 BLE Tx 测试<ul style="list-style-type: none">AT+DTM=endRX CNT: 0CRC OK: 0CRC FAIL: 0Packet count: 0OK	

指令	查询指令： N/A	设置指令： AT+DTM=<mode>,<channel>,<data_length>,<packet_type>
<ul style="list-style-type: none"> 开始 BLE Rx 测试 AT+DTM=rx,20 Start DTM Rx frequency: 20 OK 结束 BLE Rx 测试 AT+DTM=end RX CNT: 28613 CRC OK: 28613 CRC FAIL: 0 Packet count: 28613 OK 		

2.2.29. AT+AUXADC—讀取 AUXADC 原始資料

指令	查询指令： N/A	设置指令： AT+AUXADC=<src>, <gpio_idx>
响应	N/A	成功：OK 失败：ERROR
参数说明	<ul style="list-style-type: none"> <src>： <ul style="list-style-type: none"> 0: GPIO 1: VBAT 2: LDO_VCO 3: LDO_RF 4: TEMP_SEN 5: HPBG 	

指令	查询指令： N/A	设置指令： AT+AUXADC=<src>, <gpio_idx>
	6: LPBG 7: PMU_SF 8:VSS <gpio_idx>: 0 ~ 15	
注意	• 當<src>不為 GPIO 時, <gpio_idx>可不填	
示例	at+auxadc=0,10 Ref points (mV, Data) = (30, 0x75) and (1100, 0x1B3) Auxadc(gpio = 10) value = 0x0128 (633 mV) OK at+auxadc=1 Ref points (mV, Data) = (30, 0x75) and (1100, 0x1B3) Auxadc(tSrc = VBAT) value = 0x039B (2744 mV) OK	

2.2.30. AT+ADCCALVBAT—校準 VBAT

指令	查询指令： N/A	设置指令： AT+ADCVCALBAT=<mvlot>, <cal_pts_idx>
响应	N/A	成功：OK 失敗：ERROR
参数说明	• < mvlot > : mini-volt of current VBAT <cal_pts_idx>: 0 / 1	
注意	•	
示例	at+adccalvbat=3000,1	

指令	查询指令： N/A OK	设置指令： AT+ADCVCALBAT=<mvlot>,<cal_pts_idx>
----	--------------------	--

2.2.31. AT+ADCCALGPIO—校準 GPIO

指令	查询指令： N/A	设置指令： AT+ADCCALGPIO=<gpio_idx>,<mvlot>,<cal_pts_idx>
响应	N/A	成功：OK 失败：ERROR
参数说明	•<gpio_idx>: 0 ~ 15 < mvlot > : mini-volt of current VBAT <cal_pts_idx>: 0 / 1	
注意	• 需要確認該 GPIO 已經設置為輸入, 且接好電壓	
示例	at+adccalgpio=10,3000,1 OK	

2.2.32. AT+ADCDEF—恢復校準預設值

指令	查询指令： N/A	设置指令： AT+ADCDEF
响应	N/A	成功：OK 失败：ERROR
参数说明	N/A	
注意	•	
示例	at+adcdef OK	

2.2.33. AT+ADCERASE—清除 flash 校準設定

指令	查詢指令： N/A	設置指令： AT+ADCERASE
響應	N/A	成功：OK 失敗：ERROR
參數說明	•N/A	
注意	•	
示例	AT+ADCERASE OK	

2.2.34. AT+ADCSTORE—儲存校準設定到 flash

指令	查詢指令： N/A	設置指令： AT+ADCSTORE
響應	N/A	成功：OK 失敗：ERROR
參數說明	•N/A	
注意	•	
示例	AT+ADCSTORE OK	

2.2.35. AT+ADCRELOAD—重新讀取 flash 的校準值

指令	查詢指令： N/A	設置指令： AT+ADCRELOAD
響應	N/A	成功：OK 失敗：ERROR
參數說明	• N/A	

指令	查询指令： N/A	设置指令： AT+ADCRELOAD
注意	•	
示例	AT+ADCRELOAD OK	

2.2.36. AT+ADCVBAT—查詢校準後 VBAT 電壓

指令	查询指令： N/A	设置指令： AT+ADCVBAT
响应	N/A	成功：OK 失败：ERROR
参数说明	• N/A	
注意	•	
示例	AT+ADCVBAT Got Vbat = 2.751000 OK	

2.2.37. AT+ADCGPIO—查詢校準後 GPIO 電壓

指令	查询指令： N/A	设置指令： AT+ADCGPIO=<gpio_idx>
响应	N/A	成功：OK 失败：ERROR
参数说明	•<gpio_idx>: 0 ~ 15	
注意	•	
示例	AT+ADCGPIO=10 Got GPIO = 3.003000	

指令	查询指令：	设置指令：
	N/A	AT+ADCGPIO= <gpio_idx>
	OK	

3. WIFI 功能 AT 指令

3.1. WIFI 功能 AT 指令一览表

指令	说明
AT+CWMODE	设置 WIFI 模式
AT+CWJAP	连接 AP
AT+CWLAPOPT	设置 CWLAP 指令的属性
AT+CWLAP	扫描当前可用的 AP
AT+CWQAP	断开与 AP 连接
AT+CWDHCP	设置 DHCP
AT+CWAUTOCONN	上电是否自动连接 AP
AT+CWFASTCONN	是否快速连接 AP
AT+CIPSTAMAC	设置 STA 接口的 MAC 地址
AT+CIPSAT	设置 STA 的 IP 地址
AT+CWHOSTNAME	设置 STA 的主机地址

3.2.2. AT+CWJAP—连接 AP

指令	查询指令： AT+CWJAP? 功能：查询 OPL1000 Station 已连接的 AP 信息。	设置指令： AT+CWJAP=<ssid>,<pwd>[,<bssid>] 功能：设置 OPL1000 Station 需连接的 AP。
响应	+CWJAP:<ssid>,<bssid>,<channel>,<rssi> OK	OK 或者 +CWJAP:<error code> ERROR
参数说明	<ul style="list-style-type: none">• <ssid>：字符串参数，AP 的 SSID• <bssid>：AP 的 MAC 地址• <channel>：信道号• <rssi>：信号强度	<ul style="list-style-type: none">• <ssid>：目标 AP 的 SSID• <pwd>：密码最长 64 字节 ASCII• [<bssid>]：目标 AP 的 MAC 地址，一般用于有多个 SSID 相同的 AP 的情况• <error code>：（仅供参考，并不可靠）<ul style="list-style-type: none">▸ 1：连接超时▸ 2：密码错误▸ 3：找不到目标 AP▸ 4：连接失败▸ 其他值：未知错误
提示信息	// 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	
注意	<ul style="list-style-type: none">• 参数设置需要开启 Station 模式。• 若 SSID 或者 password 中含有特殊符号时，例如，" 者，即无效指令。	
示例	AT+CWJAP="abc","0123456789"	

3.2.3. AT+CWLAPOPT—设置 CWLAP 指令的属性

指令	查询指令： AT+CWLAPOPT? 功能：查询当前 CWLAPOPT 设置属性	设置指令： AT+CWLAPOPT=<sort_enable>,<mask>,<times>,<probe_counters> 功能：设置 CWLAP 指令的属性
响应	+CWLAPOPT:<sort_enable>,<mask>,<times>,<probe_counters> OK	OK 或者 ERROR
参数说明	<ul style="list-style-type: none">• <sort_enable>：指令 AT+CWLAP 的扫描结果是否按照信号强度 RSSI 值排序：<ul style="list-style-type: none">▸ 0：不排序▸ 1：根据 RSSI 排序• <mask>：对应 bit 若为 1，则指令 AT+CWLAP 的扫描结果显示相关属性，对应 bit 若为 0，则不显示。具体如下：<ul style="list-style-type: none">▸ bit 0：设置 AT+CWLAP 的扫描结果是否显示 <ecn>▸ bit 1：设置 AT+CWLAP 的扫描结果是否显示 <ssid>▸ bit 2：设置 AT+CWLAP 的扫描结果是否显示 <rssi>▸ bit 3：设置 AT+CWLAP 的扫描结果是否显示 <mac>▸ bit 4：设置 AT+CWLAP 的扫描结果是否显示 <channel>• <times>：停留在 channel 上聆听 AP 讯息包的倍数。每一倍数是停留 150ms。<ul style="list-style-type: none">▸ 预设为 1 倍，最大设置为 10 倍。即停留在每一 channel 上的时间为 150ms – 1500ms。• <probe_counters>：发出 probe request 包次数在每一 channel 中。<ul style="list-style-type: none">▸ 预设为 1 次，最大设置为 5 次。发送间隔为(150* <times>)/<probe_counters>，单位为 ms。	
示例	<p>AT+CWLAPOPT=1,31,1,5</p> <p>第一个参数为 1，表示后续如果使用 AT+CWLAP 指令，扫描结果将按照信号强度 RSSI 值排序；</p> <p>第二个参数为 31，即 0x1F，表示 <mask> 的相关 bit 全部置为 1，后续如果使用 AT+CWLAP 指令，扫描结果将显示所有参数。</p> <p>第三个参数为 1，即默认停留倍数。150ms 乘以 14 个 channel，表示扫描一次需花费 2100ms。</p> <p>第四个参数为 5，即在每个通道中发出 5 个探测请求包。间隔为 (150 * 1) / 5 = 30ms。</p>	

3.2.4. AT+CWLAP—扫描当前可用的 AP

执行指令	AT+CWLAP 功能：扫描或列出当前可用的 AP。
响应	+CWLAP:<ecn>,<ssid>,<rssi>,<mac>,<channel> OK
参数说明	<ul style="list-style-type: none"> • <ecn>：加密方式 <ul style="list-style-type: none"> ▸ 0：OPEN ▸ 1：WEP ▸ 2：WPA_PSK ▸ 3：WPA2_PSK ▸ 4：WPA_WPA2_PSK ▸ 5：WPA2_Enterprise（目前 AT 不支持连接这种加密 AP） • <ssid>：字符串参数 · AP 的 SSID • <rssi>：信号强度 • [<mac>]（选填参数）：字符串参数 · AP 的 MAC 地址 • [<channel>]（选填参数）：信道号
示例	AT+CWLAP="WiFi","ca:d7:19:d8:a6:44",6 或者查找指定 SSID 的 AP： AT+CWLAP="WiFi" 若查到超过一台指定 AP ‘WiFi’，则会把所有 SSID 中带有 ‘WiFi’ 的 AP 都找出来，方便查找

3.2.5. AT+CWQAP—断开与 AP 的连接

执行指令	AT+CWQAP
响应	OK
参数说明	-

3.2.6. AT+CWAUTOCONN—上电是否自动连接 AP

指令	查询指令：	设置指令：
	AT+CWAUTOCONN?	AT+CWAUTOCONN=<enable>,<ap_num> 功能：设置 AUTOCONN。
响应	+CWAUTOCONN:<enable>,<ap_num> OK	OK 或者 +CWAUTOCONN:<error code> ERROR
参数	<enable>:是否启动 auto connect	• <enable>：
说明	<ap_num>: 显示最大储存 AP 数量	▸ 0：上电不自动连接 AP ▸ 1：上电自动连接 AP
		• <ap_num>：
		▸ 设置最大储存 auto connect AP 的数量，范围为 1 – 3。
		• <error code>：
		▸ 1: 无效参数 ▸ 其他值: 其他错误
注意		
• 本设置保存在 flash。		
• 默认上电自动连接 AP，且最大数量为 3。		
• 上电后，需先下 AT+CWMODE=1 指令后才会生效。		
示例 AT+CWAUTOCONN=1,3		

3.2.7. AT+CWFASTCONN—快速连接 AP

指令	查询指令：	设置指令：
	AT+CWFASTCONN?	AT+CWFASTCONN=<list_id>,<enable> 功能：设置 FASTCONN。 AT+CWFASTCONN= id0, en0, id1, en1, id2, en2
响应	+CWFASTCONN:<list_id>,<enable>	OK

指令	查询指令： AT+CWFASTCONN?	设置指令： AT+CWFASTCONN=<list_id>,<enable> 功能：设置 FASTCONN。 AT+CWFASTCONN= id0, en0, id1, en1, id2, en2
	OK	或者 +CWFASTCONN:<error code> ERROR
参数说明	<list_id>:存在于 auto connect list 的 index <enable>:是否启动 fast connect	<ul style="list-style-type: none">• <list_id>:index 從 0 開始• <enable>：<ul style="list-style-type: none">▸ 0：上电不快速连接 AP▸ 1：上电自动快速连接 AP• <error code>：<ul style="list-style-type: none">▸ 1: 无效参数▸ 2: 该 index 没有 ap 的信息▸ 其他值：其他错误
注意	<ul style="list-style-type: none">• 本设置保存在 flash。• 默认上电不快速连接 AP。• 需先下 AT+CWMODE=1 指令后才会生效。• 此功能需要先存在 auto connect 信息才能使用	
示例	AT+CWFASTCONN=0,1, 1, 1, 2, 1	

3.2.8. AT+CIPSTAMAC—设置 OPL1000 Station 接口的 MAC 地址

指令	查询指令： AT+CIPSTAMAC? 功能：查询 OPL1000 Station 的 MAC 地址。	设置指令： AT+CIPSTAMAC=<mac> 功能：设置 OPL1000 Station 的 MAC 地址。
响应	+CIPSTAMAC:<mac> OK	OK
参数说明	<mac>：字符串参数，OPL1000 Station 的 MAC 地址	

指令	查询指令： AT+CIPSTAMAC? 功能：查询 OPL1000 Station 的 MAC 地址。	设置指令： AT+CIPSTAMAC=<mac> 功能：设置 OPL1000 Station 的 MAC 地址。
注意	<ul style="list-style-type: none"> • 本设置保存到 flash。可以用 AT+MACADDRDEF=0,1 命令设置 MAC 源为 Flash • MAC 地址第一个字节的 bit 0 不能为 1，例如 MAC 地址可以为 "1a:..." 但不能为 "15:..."。 • FF:FF:FF:FF:FF:FF 和 00:00:00:00:00:00 为非法 MAC，无法进行设置。 • 为使修改的 MAC address 有效，需要使用 AT+RST 进行复位。 	
示例	AT+CIPSTAMAC="18:fe:35:98:d3:7b"	

3.2.9. AT+CWHOSTNAME—设置 Station 的主机名称

指令	查询指令： AT+CWHOSTNAME? 功能：查询 OPL1000 Station 的主机名称。	设置指令： AT+CWHOSTNAME=<hostname> 功能：设置 OPL1000 Station 的主机名称。
响应	+CWHOSTNAME:<host name> OK 如果未使能 OPL1000 Station 模式，则返回 +CWHOSTNAME:<null> OK	如果成功，返回 OK 如果未使能 OPL1000 station 模式，则提示 ERROR
参数说明	<hostname>：主机名称，最长支持 32 字节	
注意	<ul style="list-style-type: none"> • 本设置不保存到 Flash，重启后将恢复默认值。 • OPL1000 Station 默认的主机名称为 "opulink"。 	
示例	AT+CWMODE=1 AT+CWHOSTNAME="my_test"	

4. TCP/IP 功能 AT 指令

4.1. TCP/IP 功能 AT 指令一览表

指令	说明
AT+CIPSTATUS	查询网络连接信息
AT+CIPDOMAIN	域名解析功能
AT+CIPSTART	建立 TCP 连接或 UDP 传输
AT+CIPSEND	发送数据
AT+CIPSENDER	发送数据
AT+CIPCLOSE	关闭 TCP/UDP 传输
AT+CIFSR	查询本地 IP 地址
AT+CIPMUX	设置多连接
AT+CIPSERVER	建立 TCP 服务器
AT+CIPMODE	设置透传模式
AT+SAVETRANSLINK	保存透传至 FLASH
AT+CIPSTO	设置 TCP 服务器超时时间
AT+CIPDINFO	接收网络数据是否提示对端 IP 和端口号
+IPD	接收网络数据
AT+PING	PING 功能

4.2. TCP/IP 功能 AT 指令描述

4.2.1. AT+CIPSTATUS—查询网络连接信息

执行指令	AT+CIPSTATUS
响应	STATUS:<stat>

执行指令 AT+CIPSTATUS

+ CIPSTATUS:<link ID>,<type>,<remote IP>,<remote port>,<local port>,<tetype>

- 参数说明**
- <stat> : OPL1000 Station 接口的状态
 - 2 : OPL1000 Station 已连接 AP · 获得 IP 地址
 - 3 : OPL1000 Station 已建立 TCP 或 UDP 传输
 - 4 : OPL1000 Station 断开网络连接
 - 5 : OPL1000 Station 未连接 AP
 - <link ID> : 网络连接 ID (0 ~ 4) · 用于多连接的情况
 - <type> : 字符串参数 · "TCP" 或者 "UDP"
 - <remote IP> : 字符串 · 远端 IP 地址
 - <remote port> : 远端端口值
 - <local port> : OPL1000 本地端口值
 - <tetype> :
 - 0 : OPL1000 作为客户端
 - 1 : OPL1000 作为服务器
-

4.2.2. AT+CIPDOMAIN—域名解析功能**执行指令 AT+CIPDOMAIN=<domain name>**

响应 +CIPDOMAIN:<IP address>

 OK

 或者

 ERROR

参数说明 <domain name> : 待解析的域名

示例 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—建立 TCP 连接或 UDP 传输

■ 建立 TCP 连接

设置指令	TCP 单连接 (AT+CIPMUX=0) 时 :	TCP 多连接 (AT+CIPMUX=1) 时 :
	AT+CIPSTART=<type>,<remote IP>,<remote port>[,<TCP keep alive>]	AT+CIPSTART=<link ID>,<type>,<remote IP>,<remote port>[,<TCP keep alive>]
响应	OK	
参数说明	<ul style="list-style-type: none">• <link ID> : 网络连接 ID (0 ~ 4) , 用于多连接的情况• <type> : 字符串参数 , 连接类型 , "TCP"或"UDP"• <remote IP> : 字符串参数 , 远端 IP 地址• <remote port> : 远端端口号• [<TCP keep alive>] : TCP keep-alive 侦测时间 , 默认关闭此功能 , 建议自行设置开启此功能<ul style="list-style-type: none">▸ 0 : 关闭 TCP keep-alive 功能▸ 1 ~ 7200 : 侦测时间 , 单位为 1s	
提示信息	// 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	
注意	建议创建 TCP 连接时 , 开启 keep-alive 功能。	
示例	AT+CIPSTART="TCP","192.168.101.110",1000	

■ 建立 UDP 传输

设置指令	单连接模式 (AT+CIPMUX=0) 时 :	多连接模式 (AT+CIPMUX=1) 时 :
	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>]
响应	OK	
参数说明	<ul style="list-style-type: none">• <link ID> : 网络连接 ID (0 ~ 4) , 用于多连接的情况	

设置指令	单连接模式 (AT+CIPMUX=0) 时 :	多连接模式 (AT+CIPMUX=1) 时 :
	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>]
<hr/>		
<ul style="list-style-type: none">• <type> : 字符串参数 , 连接类型 , "TCP"或"UDP"• <remote IP> : 字符串参数 , 远端 IP 地址• <remote port> : 远端端口号• [<UDP local port>] : UDP 本地端口• [<UDP mode>] : UDP 传输的属性 , 若为透传模式 , 则必须为 0<ul style="list-style-type: none">▸ 0 : 收到数据后 , 不更改远端目标 , 默认值为 0▸ 1 : 收到数据后 , 改变一次远端目标▸ 2 : 收到数据后 , 改变远端目标		
注意 :		
使用 <UDP mode> 必须先填写 <UDP local port> 。		
<hr/>		
提示信息	// 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	
<hr/>		
示例	AT+CIPSTART="UDP","192.168.101.110",1000,1002,2	
<hr/>		

4.2.4. AT+CIPSEND—发送数据

设置指令	<p>1. 单连接时：(+CIPMUX=0) AT+CIPSEND=<length></p> <p>2. 多连接时：(+CIPMUX=1) AT+CIPSEND=<link ID>,<length></p> <p>3. 如果是 UDP 传输，可以设置远端 IP 和端口： AT+CIPSEND=[<link ID>,<length>,<remote IP>,<remote port>]</p> <p>功能：在普通传输模式时，设置发送数据的长度。</p>	<p>执行指令： AT+CIPSEND</p> <p>功能：在透传模式时，开始传输数据。</p>
响应	<p>发送指定长度的数据。</p> <p>收到此命令后先换行返回 >，然后开始接收串口数据，当数据长度满 length 时发送数据，回到普通指令模式，等待下一条 AT 指令。如果未建立连接或连接被断开，返回：</p> <p>ERROR</p> <p>如果数据发送成功，返回：</p> <p>SEND OK</p> <p>如果数据发送失败，返回：</p> <p>SENDFAIL</p>	<p>收到此命令后先打印 OK 并换行返回>符号。</p> <p>透传模式发送数据，每包最大 2048 bytes 或者每 20 ms 间隔区分。</p> <p>当输入单一包 +++ 时，返回一般 AT 指令。</p> <p>当退出透传时，请至少间隔 1 秒后再发送 AT 指令。</p> <p>若为 UDP 透传，指令 AT+CIPSTART 参数 <UDP mode> 必须为 0。</p>
参数说明	<p>• <link ID>：网络连接 ID 号 (0 ~ 4)，用于多连接的情况</p> <p>• <length>：数字参数，表明发送数据的长度，最大长度为 2048</p> <p>• [<remote IP>]：UDP 传输可以设置对端 IP</p> <p>• [<remote port>]：UDP 传输可以设置对端端口</p>	
示例	<p>- 詳細請參考第 6 章 AT 指令使用示例。</p>	

4.2.5. AT+CIPSENDEX—发送数据

指令	设置指令： 1. 单连接时：(+CIPMUX=0) AT+CIPSENDEX=<length> 2. 多连接时：(+CIPMUX=1) AT+CIPSENDEX=<link ID>,<length> 3. 如果是 UDP 传输，可以设置远端 IP 和端口： AT+CIPSENDEX=[<link ID>,<length>,<remote IP>,<remote port>] 指令功能：在普通传输模式时，设置发送数据的长度。
响应	发送指定长度的数据。 收到此命令后先换行返回>，然后开始接收串口数据，当数据长度满 length 或者遇到字符 \0 时，发送数据。 如果未建立连接或连接被断开，返回： ERROR 如果数据发送成功，返回： SEND OK 如果数据发送失败，返回： SENDFAIL
参数说明	<ul style="list-style-type: none">• <link ID>：网络连接 ID 号 (0 ~ 4)，用于多连接的情况• <length>：数字参数，表明发送数据的长度，最大长度为 2048• 当接收数据长度满 length 或者遇到字符 \0 时，发送数据，回到普通指令模式，等待下一条 AT 指令。• 用户如需发送 \0，请转义为 \\0。

4.2.6. AT+CIPCLOSE—关闭 TCP/UDP 传输

指令	设置指令（用于多连接的情况）： AT+CIPCLOSE=<link ID> 功能：关闭 TCP/UDP 传输。	执行指令（用于单连接的情况）： AT+CIPCLOSE
响应	OK	
参数说明	<link ID>：需要关闭的连接 ID 号。当 ID 为 5 时，关闭所有连接。	

指令	设置指令（用于多连接的情况）： AT+CIPCLOSE=<link ID> 功能：关闭 TCP/UDP 传输。	执行指令（用于单连接的情况）： AT+CIPCLOSE
提示信息	// When connection ends, it will prompt message as below [<link ID>,] CLOSED	

4.2.7. AT+CIFSR—查询本地 IP 地址

执行指令	AT+CIFSR	
响应	+CIFSR:STAIP,<Station IP address> +CIFSR:STAMAC,<Station MACaddress> OK	
参数说明	<IP address> : OPL1000 Station 的 IP 地址 <MAC address> : OPL1000 Station 的 MAC 地址	
注意	OPL1000 Station IP 需连上 AP 后，才可以查询。	

4.2.8. AT+CIPMUX—设置多连接

指令	查询指令： AT+CIPMUX?	设置指令： AT+CIPMUX=<mode> 功能：设置连接类型。
响应	+CIPMUX:<mode> OK	OK
参数说明	<mode> : ‣ 0：单连接模式 ‣ 1：多连接模式	
注意	• 默认为单连接； • 只有非透传模式 (AT+CIPMODE=0)，才能设置为多连接； • 必须在没有连接建立的情况下，设置连接模式；	

指令	查询指令：	设置指令：
	AT+CIPMUX?	AT+CIPMUX=<mode>
		功能：设置连接类型。
	• 如果建立了 TCP 服务器，想切换为单连接，必须关闭服务器 (AT+CIPSERVER=0)，服务器仅支持多连接。	
示例	AT+CIPMUX=1	

4.2.9. AT+CIPSERVER—建立 TCP 服务器

指令	查询指令：	设置指令：
	AT+CIPSERVER?	AT+CIPSERVER=<mode>[,<port>]
		功能：设置服务器。
响应	+CIPSERVER:<mode>,<port> OK	OK
参数说明	<mode>： ‣ 0：关闭服务器 ‣ 1：建立服务器 [<port>]：选填参数。端口号，默认为 333。	
提示信息	// 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	
注意	• 多连接情况下 (AT+CIPMUX=1)，才能开启服务器。 • 创建服务器后，自动建立服务器监听。 • 当有客户端接入，会自动占用一个连接 ID。	
示例	• 建立 TCP 服务器 AT+CIPMUX=1 AT+CIPSERVER=1,80	

4.2.10. AT+CIPMODE—设置透传模式

指令	查询指令： AT+CIPMODE?	设置指令： AT+CIPMODE=<mode> 功能：设置透传模式。
响应	+CIPMODE:<mode> OK	OK
参数说明	<mode>： ‣ 0：一般传输模式 ‣ 1：透传模式，仅支持 TCP 单连接及 UDP 固定远程目标的配置。	
注意	• 本设置不保存到 Flash • 透传模式时，如果是 TCP 连接断开，将会不停地尝试重连直到输入+++退出透传模式以及停止重连。	
示例	AT+CIPMODE=1	

4.2.11. AT+SAVETRANSLINK—保存透传至 FLASH

保存 TCP 透传 (单连接) 至 FLASH

设置指令	AT+SAVETRANSLINK=<mode>,<remote IP or domain name>,<remote port>[,<type>,<TCP keep alive>]
响应	OK
参数说明	• <mode>： ‣ 0：一般 AT 模式，即不会开机后进入透传模式。 ‣ 1：保存至 FLASH 并开机后进入透传模式。 • <remote IP>：远程 IP 或域名。 • <remote port>：远程埠 • [<type>]：TCP 或 UDP，缺省预设为 TCP。(选填参数) • [<TCP keep alive>]：TCP keep alive 侦测，预设为关闭。(选填参数) ‣ 0：关闭 TCP keep alive 侦测功能。 ‣ 1 ~ 7200：侦测时间，单位为秒。
注意	• 本设置将透传设置储存至 FLASH，下次上电自动建立联机并进入透传。

设置指令 AT+SAVETRANSLINK=<mode>,<remote IP or domain name>,<remote port>[,<type>,<TCP keep alive>]

- 只要参数符合规范，设置就会储存至 FLASH。

示例 AT+SAVETRANSLINK=0
AT+SAVETRANSLINK=1,"192.168.1.100",80,"TCP"

保存 UDP 透传 至 FLASH

设置指令 AT+SAVETRANSLINK=<mode>,<remote IP or domain name>,<remote port>,<type>[,<UDP local port>]

响应 OK

参数说明 • <mode> :

- 0：一般 AT 模式，即不会开机后进入透传模式。
- 1：保存至 FLASH 并开机后进入透传模式。
- <remote IP>：远程 IP 或域名。
- <remote port>：远程埠
- <type>：UDP，缺省预设 TCP。(选填参数)
- [<UDP local port>]：UDP 传输时，使用的本地端口。(选填参数)

注意 • 本设置将透传设置储存至 FLASH，下次上电自动建立联机并进入透传。
• 只要参数符合规范，设置就会储存至 FLASH。

示例 AT+SAVETRANSLINK=0
AT+SAVETRANSLINK=1,"192.168.1.100",23456,"UDP",1000

4.2.12. AT+CIPSTO—设置 TCP 服务器超时时间

指令	查询指令： AT+CIPSTO? 功能：查询 TCP 服务器超时时间。	设置指令： AT+CIPSTO=<time> 功能：设置 TCP 服务器超时时间。
-----------	--	--

响应	+CIPSTO:<time> OK	OK
-----------	----------------------	----

参数说明 <time>：TCP 服务器超时时间，取值范围 0 ~ 7200s。

指令	查询指令： AT+CIPSTO? 功能：查询 TCP 服务器超时时间。	设置指令： AT+CIPSTO=<time> 功能：设置 TCP 服务器超时时间。
注意	• OPL1000 作为 TCP 服务器，会断开一直不通信直至超时的 TCP 客户端连接。 • 如果设置 AT+CIPSTO=0，则永远不会超时，不建议这样设置。	
示例	AT+CIPMUX=1 AT+CIPSERVER=1,1001 AT+CIPSTO=10	

4.2.13. AT+CIPDINFO—接收网络数据时是否提示对端 IP 和端口

设置指令	AT+CIPDINFO=<mode>
响应	OK
参数	<mode>：
说明	▸ 0：不显示对端 IP 和端口 ▸ 1：显示对端 IP 和端口
示例	AT+CIPDINFO=1

4.2.14. +IPD—接收网络数据

指令	单连接时： (+CIPMUX=0)+IPD,<len>[,<remote IP>,<remote port>]:<data>	多连接时： (+CIPMUX=1)+IPD,<link ID>,<len>[,<remote IP>,<remote port>]:<data>
参数说明	此指令在普通指令模式下有效，OPL1000 接收到网络数据时向串口发送 +IPD 和数据。 • [<remote IP>]：网络通信对端 IP，由指令 AT+CIPDINFO=1 使能显示 • [<remote port>]：网络通信对端端口，由指令 AT+CIPDINFO=1 使能 • <link ID>：收到网络连接的 ID 号 • <len>：数据长度 • <data>：收到的数据	

4.2.15. AT+PING—Ping 功能

设置指令	AT+PING=<IP> 功能：ping 功能。
响应	+PING:<time> OK 或 +PING:TIMEOUT ERROR
参数说明	• <IP>：字符串参数，IP 地址 • <time>：ping 响应时间
示例	AT+PING="192.168.1.1" AT+PING=" www.baidu.com "

5. BLE 相关 AT 指令

5.1. BLE 指令一览表

指令	说明
AT+BLEINIT	BLE 初始化
AT+BLEADDR	设置 BLE 设备地址
AT+BLENNAME	设置 BLE 设备名称
AT+BLESCANRSPDATA	设置 BLE 扫描回应
AT+BLEADVPARAM	设置 BLE 广播参数
AT+BLEADVDATA	设置 BLE 广播数据
AT+BLEADVSTART	开始 BLE 广播
AT+BLEADVSTOP	结束 BLE 广播
AT+BLECONNPARAM	更新 BLE 连接参数
AT+BLEDISCONN	断开 BLE 连接
AT+BLEDATALEN	设置 BLE 数据包长度
AT+BLECFGMTU	设置 BLE MTU 长度
AT+BLEGATTSSRVCRE	GATTs 创建服务
AT+BLEGATTSSRVSTART	GATTs 开启服务
AT+BLEGATTSSRVSTOP	GATTs 关闭服务
AT+BLEGATTSSRV	GATTs 查询服务
AT+BLEGATTSSCHAR	GATTs 查询服务特征
AT+BLEGATTSENTFY	GATTs 通知服务特征值
AT+BLEGATTSSIND	GATTs 指示服务特征值
AT+BLEGATTSSSETATTR	GATTs 设置服务特征值
AT+BLEGATTCPRIMSRV	GATTC 发现基本服务
AT+BLEGATTCINCLSRV	GATTC 发现包含服务

5.2. BLE 指令描述

5.2.1. AT+BLEINIT—BLE 初始化



OpuLinks
旺凌科技

5.2.2. AT+BLEADDR—设置 BLE 设备地址

指令	查询指令：	设置指令：
	AT+BLEADDR?	AT+BLEADDR=<addr_type>,<random_addr>
	功能：查询 BLE 设备的 public address。	功能：设置 BLE 设备的地址。目前仅支持设置 random address。
响应	+BLEADDR:<BLE_public_addr> OK	OK
参数	<addr_type>：	
说明	▸ 0：public address ▸ 1：random address	
注意	<ul style="list-style-type: none">• 目前可设置/查询 public address，对 random address 仅支持设置。• 在设置 BLE public address 之前需要用 AT+BLEINIT=1 指令对 BLE 完成初始化操作• 为使修改的 BLE public address 有效，需要使用 AT+RST 进行复位。• random address 要求最高两个 bit 必须全 1，详细可参考 BLE spec。	
示例	AT+BLEADDR=1,"08:7f:24:87:1c:f7"	

5.2.3. AT+BLENAME—设置 BLE 设备名称

指令	查询指令：	设置指令：
	AT+BLENAME?	AT+BLENAME=<device_name>
	功能：查询 BLE 设备名称。	功能：设置 BLE 设备名称。
响应	+BLENAME:<device_name> OK	OK
参数	<device_name>：BLE 设备名称	
说明		
注意	<ul style="list-style-type: none">• 默认设备名称为“BLE_AT”。• 本指令设置的设备名称，需要在建立 BLE 连接之后，对端设备才能获取到，它其实设置的是 GAP service 中 device name characteristic 的值，详情请见 BLE core v4.2 vol.3 part C 12.1。• 如果是需要在扫描广播包时得到的设备名称，则需要通过 AT+BLEADVDATA 设置。	
示例	AT+BLENAME="opl_demo"	

5.2.4. AT+BLESCANRSPDATA—设置 BLE 扫描响应

指令	设置指令： AT+BLESCANRSPDATA=<scan_rsp_data> 功能：设置 BLE 扫描响应。
响应	OK
参数说明	<scan_rsp_data>：扫描响应。参数实际为 HEX 字串。例如，设置扫描响应为 0x11 0x22 0x33 0x44 0x55，则设置指令为：AT+BLESCANRSPDATA="1122334455"
注意	扫描响应支持的最大长度为 31 字节。
示例	AT+BLEINIT=2 // 初始化为 server AT+BLESCANRSPDATA="1122334455"

5.2.5. AT+BLEADVPARAM—设置广播参数

指令	查询指令： AT+BLEADVPARAM? 功能：查询广播参数。	设置指令： AT+BLEADVPARAM=<adv_int_min>,<adv_int_max>,<adv_type>,<own_addr_type>,<channel_map>,<adv_filter_policy>,<peer_addr_type>,<peer_addr> 功能：设置广播参数。
响应	+BLEADVPARAM:<adv_int_min>,<adv_int_max>,<adv_type>,<own_addr_type>,<channel_map>,<adv_filter_policy>,<peer_addr_type>,<peer_addr> OK	OK
参数说明	<adv_int_min>：最小广播间隔，取值范围：0x0020 ~ 0x4000，最小 20ms，最大 10240ms <adv_int_max>：最大广播间隔，取值范围：0x0020 ~ 0x4000，最小 20ms，最大 10240ms	

指令	<p>查询指令：</p> <p>AT+BLEADVPARAM?</p> <p>功能：查询广播参数。</p>	<p>设置指令：</p> <p>AT+BLEADVPARAM=<adv_int_min>,<adv_int_max>,<adv_type>,<own_addr_type>,<channel_map>[,<adv_filter_policy>,<peer_addr_type>,<peer_addr>]</p> <p>功能：设置广播参数。</p>
----	---	--

<adv_type>：广播类型

- 0：ADV_TYPE_IND
- 1: ADV_TYPE_DIRECT_IND_HIGH
- 2：ADV_TYPE_SCAN_IND
- 3：ADV_TYPE_NONCONN_IND

<own_addr_type>：BLE 地址类型

- 0：BLE_ADDR_TYPE_PUBLIC
- 1：BLE_ADDR_TYPE_RANDOM

<channel_map>：广播信道

- 1：ADV_CHNL_37
- 2：ADV_CHNL_38
- 4：ADV_CHNL_39
- 7：ADV_CHNL_ALL

[<adv_filter_policy>]（选填参数）：过滤器规则

- 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>]（选填参数）：对方 BLE 地址类型

- 0：PUBLIC
- 1：RANDOM

[<peer_addr>]（选填参数）：对方 BLE 地址

指令	查询指令： AT+BLEADVPARAM? 功能：查询广播参数。	设置指令： AT+BLEADVPARAM=<adv_int_min>,<adv_int_max>,<adv_type>,<own_addr_type>,<channel_map>[,<adv_filter_policy>,<peer_addr_type>,<peer_addr>] 功能：设置广播参数。
注意	<adv_filter_policy>,<peer_addr_type>,<peer_addr>三个参数要求同时缺省，或者同时设置。	
示例	AT+BLEINIT=2 // 初始化为 server AT+BLEADVPARAM=50,50,0,0,4,0,0,"12:34:45:78:66:88"	

5.2.6. AT+BLEADVDATA—设置 BLE 广播数据

指令	设置指令： AT+BLEADVDATA=<adv_data> 功能：设置 BLE 广播数据。
响应	OK
参数说明	<adv_data>：广播数据包。参数实际为 HEX 字串。例如，设置广播数据为 0x11 0x22 0x33 0x44 0x55，则设置指令为：AT+BLEADVDATA="1122334455"
注意	广播包最大长度为 31 字节。
示例	AT+BLEINIT=2 // 初始化为 server AT+BLEADVDATA="1122334455"

5.2.7. AT+BLEADVSTART—开始 BLE 广播

指令	执行指令： AT+BLEADVSTART 功能：开始 BLE 广播。
响应	OK
参数说明	无

指令	执行指令： AT+BLEADVSTART 功能：开始 BLE 广播。
注意	• 若未设置广播参数 (AT+BLEADVPARAM=<adv_parameter>) ，则使用默认广播参数； • 若未设置广播数据 (AT+BLEADVDATA=<adv_data>) ，则发送全 0 数据包。
示例	AT+BLEINIT=2 // 初始化为 server AT+BLEADVSTART

5.2.8. AT+BLEADVSTOP—结束 BLE 广播

指令	执行指令： AT+BLEADVSTOP 功能：结束 BLE 广播。
响应	OK
参数说明	无
注意	若开始广播后，成功建立 BLE 连接，则会自动结束 BLE 广播，无需调用本指令。
示例	AT+BLEINIT=2 // 初始化为 server AT+BLEADVSTART AT+BLEADVSTOP

5.2.9. AT+BLECONNPARAM—更新 BLE 连接参数

指令	查询指令： AT+BLECONNPARAM? 功能：查询 BLE 连接参数。	设置指令： AT+BLECONNPARAM= <conn_index>,<min_interval>,<max_interval>,<latency>,<timeout> 功能：更新 BLE 连接参数。
响应	+BLECONNPARAM:<conn_index>,<cur_interval>,<latency>,<timeout> OK	OK //指令已接收，将尝试更新连接参数 +BLECONNPARAM: <conn_index>,0 如果更新失败，将提示

指令	查询指令： AT+BLECONNPARAM? 功能：查询 BLE 连接参数。	设置指令： AT+BLECONNPARAM=<conn_index>,<min_interval>,<max_interval>,<latency>,<timeout> 功能：更新 BLE 连接参数。
		+BLECONNPARAM: <conn_index>,-1
参数说明 <conn_index>：BLE 连接号，当前只支持 index 为 0 的单连接 <min_interval>：最小连接间隔，取值范围：0x0006 ~ 0x0C80 <max_interval>：最大连接间隔，取值范围：0x0006 ~ 0x0C80 <cur_interval>：当前连接间隔 <latency>：时延，取值范围：0x0000 ~ 0x01F3 <timeout>：超时，取值范围：0x000A ~ 0x0C80		
注意	本指令要求先建立连接，并且仅支持 BLE client 更新连接参数。	
示例	AT+BLEINIT=1 // 初始化为 client	
	AT+BLECONN=0,"24:0a:c4:09:34:23" // 建立 BLE 连接	
	AT+BLECONNPARAM=0,12,14,1,500 // 更新 BLE 连接参数	

5.2.10. AT+BLEDISCONN—断开 BLE 连接

指令	设置指令： AT+BLEDISCONN=<conn_index> 功能：断开 BLE 连接。
响应	+BLEDISCONN:<conn_index>,<remote_address> OK
参数说明	<conn_index>：BLE 连接号，当前只支持 index 为 0 的单连接 <remote_address>：对方 BLE 设备地址
示例	AT+BLEINIT=1 // 初始化为 client AT+BLECONN=0,"24:0a:c4:09:34:23" // 建立 BLE 连接 AT+BLEDISCONN=0 // 断开 BLE 连接

5.2.11. AT+BLEDATALEN—设置 BLE 数据包长度

指令	设置指令： AT+BLEDATALEN=<conn_index>,<pkt_data_len> 功能：设置 BLE 数据包长度。
响应	OK
参数说明	<conn_index>：BLE 连接号，当前只支持 index 为 0 的单连接 <pkt_data_len>：数据包长度，取值范围：0x001b ~ 0x00fb
注意	需要先建立 BLE 连接，才能设置 packet length。
示例	AT+BLEINIT=1 // 初始化为 client AT+BLECONN=0,"24:0a:c4:09:34:23" AT+BLEDATALEN=0,30

5.2.12. AT+BLECFGMTU—设置 GATT MTU 的长度

指令	设置指令： AT+BLECFGMTU? 功能：查询 GATT (Generic Attribute Profile) MTU 的长度。	设置指令： AT+BLECFGMTU=<conn_index>,<mtu_size> 功能：设置 GATT MTU 的长度。
响应	+BLECFGMTU:<conn_index>,<mtu_size> OK // 指令已接收, OK	
参数	<conn_index>：BLE 连接号，当前只支持 index 为 0 的单连接	
说明	<mtu_size>：BLE 最大传输单元的长度	
注意	• 最终实际的 MTU 长度需经过协商，设置指令返回 OK 仅表示尝试协商 MTU，因此，设置长度不一定生效，建议设置后，使用查询指令 AT+BLECFGMTU? 查询实际的 MTU 长度。	
示例	AT+BLEINIT=1 // 初始化为 client AT+BLECONN=0,"24:12:5f:9d:91:98"// 建立 BLE 连接 AT+BLECFGMTU=0,300	

5.2.13. AT+BLEGATTSSRVCRE—GATTS 创建服务

指令	执行指令： AT+BLEGATTSSRVCRE 功能：GATTS 创建服务。
响应	OK
参数说明	无
注意	• OPL1000 作为 server 应该在初始化完成后，及时创建服务。BLE 连接建立后，无法创建服务。
示例	AT+BLEINIT=2 // 初始化为 server AT+BLEGATTSSRVCRE

5.2.14. AT+BLEGATTSSRVSTART—GATTS 开启服务

指令	执行指令： AT+BLEGATTSSRVSTART 功能：GATTS 开启全部服务。	设置指令： AT+BLEGATTSSRVSTART=<srv_index> 功能：GATTS 开启某指定服务。
响应	OK	
参数说明	无	<srv_index>：服务序号，从 1 起始递增。
示例	AT+BLEINIT=2 // 初始化为 server AT+BLEGATTSSRVCRE AT+BLEGATTSSRVSTART	

5.2.15. AT+BLEGATTSSRVSTOP—GATTS 停止服务

指令	执行指令： AT+BLEGATTSSRVSTOP 功能：GATTS 停止全部服务。	设置指令： AT+BLEGATTSSRVSTOP=<srv_index> 功能：GATTS 停止某指定服务。
响应	OK	
参数说明	无	<srv_index>：服务序号，从 1 起始递增。
示例	AT+BLEINIT=2 // 初始化为 server	

指令	执行指令：	设置指令：
	AT+BLEGATTSSRVSTOP	AT+BLEGATTSSRVSTOP=<srv_index>
	功能：GATTS 停止全部服务。	功能：GATTS 停止某指定服务。
	AT+BLEGATTSSRVCRE	
	AT+BLEGATTSSRVSTART	
	AT+BLEGATTSSRVSTOP	

5.2.16. AT+BLEGATTSSRV—GATTS 发现服务

指令	查询指令：
	AT+BLEGATTSSRV?
	功能：GATTS 发现服务。
响应	+BLEGATTSSRV:<srv_index>,<start>,<srv_uuid>,<srv_type>
	OK
参数说明	<srv_index>：服务序号，从 1 起始递增
	<start>：
	▸ 0：服务未开始
	▸ 1：服务已开始
	<srv_uuid>：服务的 UUID
	<srv_type>：服务的类型
	▸ 0：次要服务
	▸ 1：首要服务
示例	AT+BLEINIT=2 // 初始化为 server
	AT+BLEGATTSSRVCRE
	AT+BLEGATTSSRV?

5.2.17. AT+BLEGATTCHAR—GATTS 发现服务特征

指令	查询指令： AT+BLEGATTCHAR? 功能：GATTS 发现服务特征。
响应	//对于服务特征信息，显示如下： +BLEGATTCHAR:"char",<srv_index>,<char_index>,<char_uuid>,<char_prop> //对于描述符信息，显示如下： +BLEGATTCHAR:"desc",<srv_index>,<char_index>,<desc_index> OK
参数说明	<srv_index>：服务序号，从 1 起始递增 <char_index>：服务特征的序号，从 1 起始递增 <char_uuid>：服务特征的 UUID <char_prop>：服务特征的属性 <desc_index>：特征描述符序号
示例	AT+BLEINIT=2 // 初始化为 server AT+BLEGATTSSRVCRE AT+BLEGATTSSRVSTART AT+BLEGATTCHAR?

5.2.18. AT+BLEGATTSTNTFY—GATTS 通知服务特征值

指令	设置指令： AT+BLEGATTSTNTFY=<conn_index>,<srv_index>,<char_index>,<length> 功能：GATTS 通知服务特征值。
响应	收到此命令后先换行返回 >，然后开始接收串口数据，当数据长度满 <length> 时，执行通知操作。若通知操作成功，则提示 OK
参数说明	<conn_index>：BLE 连接号，当前只支持 index 为 0 的单连接 <srv_index>：服务序号，由指令 AT+BLEGATTCHAR? 查询可得 <char_index>：服务特征的序号，由指令 AT+BLEGATTCHAR? 查询可得 <length>：数据长度

指令 设置指令：
AT+BLEGATTSENTFY=<conn_index>,<srv_index>,<char_index>,<length>
功能：GATTS 通知服务特征值。

示例 以下为 notify 的简单示例，

```
AT+BLEINIT=2 // 初始化为 server
AT+BLEGATTSSRVCRE
AT+BLEGATTSSRVSTART
AT+BLEADVSTART// 开始广播，等待 client 连接，并配置接收 notify
AT+BLEGATTSSCHAR?// 查询允许 notify 的特征
//例如，使用 3 号服务的 6 号特征通知长度为 4 的数据
AT+BLEGATTSENTFY=0,3,6,4
// 提示 > 符号后，输入 4 字节数据即可，例如 "1234"
```

5.2.19. AT+BLEGATTSIND—GATTS 指示服务特征值

指令 设置指令：
AT+BLEGATTSIND=<conn_index>,<srv_index>,<char_index>,<length>
功能：GATTS 指示服务特征值。

响应 收到此命令后先换行返回 >，然后开始接收串口数据，当数据长度满 <length> 时，执行指示操作。若指示操作成功，则提示 OK

参数说明 <conn_index>：BLE 连接号，当前只支持 index 为 0 的单连接
 <srv_index>：服务序号，由指令 AT+BLEGATTSSCHAR? 查询可得
 <char_index>：服务特征的序号，由指令 AT+BLEGATTSSCHAR? 查询可得
 <length>：数据长度

示例 以下为 indicate 的简单示例，

```
AT+BLEINIT=2 // 初始化为 server
AT+BLEGATTSSRVCRE
AT+BLEGATTSSRVSTART
AT+BLEADVSTART// 开始广播，等待 client 连接，client 端连接后，应该设置为接收 indication
AT+BLEGATTSSCHAR?// 查询允许 indicate 的特征
```

指令	设置指令： AT+BLEGATTSIND=<conn_index>,<srv_index>,<char_index>,<length> 功能：GATTS 指示服务特征值。
	//例如，使用 3 号服务的 7 号特征指示长度为 4 的数据 AT+BLEGATTSIND=0,3,7,4 // 提示 > 符号后，输入 4 字节数据即可，例如 "1234"

5.2.20. AT+BLEGATTSSETATTR—GATTS 设置服务特征值

指令	设置指令： AT+BLEGATTSSETATTR=<srv_index>,<char_index>[,<desc_index>],<length> 功能：GATTS 设置服务特征（描述符）值。
响应	收到此命令后先换行返回 >，然后开始接收串口数据，当数据长度满 <length> 时，执行设置操作。若设置操作成功，则提示 OK
参数说明	<srv_index>：服务发现结果序号，由 AT+BLEGATTSCHAR? 查询结果中获得 <char_index>：服务特征的序号，由 AT+BLEGATTSCHAR? 查询结果中获得 [<desc_index>]（选填参数）：特征描述符序号。若填写，则设置描述符的值；若未填写，则设置特征值。 <length>：数据长度
注意	<length> 不能超过该特征（描述符）支持的最大长度。例如，该服务特征值为 "0x30 0x31"，最大长度为 2，如果设置 <lengh> 为 3 超过最大长度，则会报错。
示例	AT+BLEINIT=2 // 初始化为 server AT+BLEGATTSSRVCRE AT+BLEGATTSSRVSTART AT+BLEGATTSCHAR? //例如，向 1 号服务的 1 号特征写入长度为 4 的数据 AT+BLEGATTSSETATTR=1,1,,4 // 提示 > 符号后，输入 4 字节数据即可，例如 "1234"

5.2.21. AT+BLEGATTCPRIMSRV—GATTC 发现基本服务

指令	设置指令： AT+BLEGATTCPRIMSRV=<conn_index> 功能：GATTC 发现基本服务。
响应	+BLEGATTCPRIMSRV:<conn_index>,<srv_index>,<srv_uuid>,<srv_type> OK
参数说明	<conn_index>：BLE 连接号，当前只支持 index 为 0 的单连接 <srv_index>：服务发现结果序号，从 1 起始递增 <srv_uuid>：服务的 UUID <srv_type>：服务的类型 ▸ 0：次要服务 ▸ 1：首要服务
注意	使用本指令，需要先建立 BLE 连接。
示例	AT+BLEINIT=1 // 初始化为 client AT+BLECONN=0,"24:12:5f:9d:91:98"// 建立 BLE 连接 AT+BLEGATTCPRIMSRV=0

5.2.22. AT+BLEGATTCINCLSRV—GATTC 发现包含服务

指令	设置指令： AT+BLEGATTCINCLSRV= <conn_index>,<srv_index> 功能：GATTC 发现包含服务。
响应	+BLEGATTCINCLSRV:<conn_index>,<srv_index>,<srv_uuid>,<srv_type>,<included_srv_uuid>,<included_srv_type> OK
参数说明	<conn_index>：BLE 连接号，当前只支持 index 为 0 的单连接 <srv_index>：服务发现结果序号，由 AT+BLEGATTCPRIMSRV=<conn_index> 查询结果中获得 <srv_uuid>：服务的 UUID <srv_type>：服务的类型

指令	设置指令： AT+BLEGATTCINCLSRV=<conn_index>,<srv_index> 功能：GATTC 发现包含服务。
	<ul style="list-style-type: none">▸ 0：次要服务▸ 1：首要服务 <p><included_srv_uuid>：包含服务的 UUID</p> <p><included_srv_type>：包含服务的类型</p> <ul style="list-style-type: none">▸ 0：次要服务▸ 1：首要服务
注意	使用本指令，需要先建立 BLE 连接。
示例	AT+BLEINIT=1 // 初始化为 client AT+BLECONN=0,"24:12:5f:9d:91:98"// 建立 BLE 连接 AT+BLEGATTCPRIMSRV=0 AT+BLEGATTCINCLSRV=0,1//根据前一条指令的查询结果，指定 index 查询

5.2.23. AT+BLEGATTCCHAR—GATTC 发现服务特征

指令	设置指令： AT+BLEGATTCCHAR=<conn_index>,<srv_index> 功能：GATTC 发现服务特征。
响应	//对于服务特征信息，显示如下： +BLEGATTCCHAR:"char",<conn_index>,<srv_index>,<char_index>,<char_uuid>,<char_prop> //对于描述符信息，显示如下： +BLEGATTCCHAR:"desc",<conn_index> , <srv_index>,<char_index>,<desc_index>,<desc_uuid> OK
参数说明	<conn_index>：BLE 连接号，当前只支持 index 为 0 的单连接 <srv_index>：服务发现结果序号，由 AT+BLEGATTCPRIMSRV=<conn_index> 查询结果中获得 <char_index>：服务特征的序号，从 1 起始递增 <char_uuid>：服务特征的 UUID

指令	设置指令： AT+BLEGATTCCHAR=<conn_index>,<srv_index> 功能：GATTC 发现服务特征。
	<char_prop>：服务特征的属性 <desc_index>：特征描述符序号 <desc_uuid>：特征描述符的 UUID
注意	使用本指令，需要先建立 BLE 连接。
示例	AT+BLEINIT=1 // 初始化为 client AT+BLECONN=0,"24:12:5f:9d:91:98"// 建立 BLE 连接 AT+BLEGATTCPRIMSRV=0 AT+BLEGATTCCHAR=0,1//根据前一条指令的查询结果，指定 index 查询

5.2.24. AT+BLEGATTCRD—GATTC 读取服务特征值

指令	设置指令： AT+BLEGATTCRD=<conn_index>,<srv_index>,<char_index>[,<desc_index>] 功能：GATTC 读取服务特征（描述符）值。
响应	+BLEGATTCRD:<conn_index>,<len>,<value> OK
参数说明	<conn_index>：BLE 连接号，当前只支持 index 为 0 的单连接 <srv_index>：服务发现结果序号，由 AT+BLEGATTCPRIMSRV=<conn_index> 查询结果中获得 <char_index>：服务特征的序号，由 AT+BLEGATTCCHAR=<conn_index>,<srv_index> 查询结果中获得 [<desc_index>]（选填参数）：特征描述符序号。若不设置，读取特征值；若设置，读取描述符的值。 <len>：数据长度 <value>：HEX 字符串 ▸ 若由指令 AT+BLEGATTCRD=<conn_index>,<srv_index>,<char_index> 读取服务特征的值，例如指令 读取返回"+BLEGATTCRD:0,1,30"表示特征值长度为 1 个字节，内容为 HEX 字符串 "0x30"。

指令	<p>设置指令：</p> <p>AT+BLEGATTCRD=<conn_index>,<srv_index>,<char_index>[,<desc_index>]</p> <p>功能：GATTC 读取服务特征（描述符）值。</p>
	<p>• 若由指令</p> <p>AT+BLEGATTCRD=<conn_index>,<srv_index>,<char_index>,<desc_index> 读取服务特征描</p> <p>述符的值，例如指令读取返回“+BLEGATTCRD:0,4,30313233”表示特征描述符的值长度为 4 个字节，内容为 HEX 字符串“0x30 0x31 0x32 0x33”。</p>
注意	<p>• 使用本指令，需要先建立 BLE 连接。</p> <p>• 如果该服务特征属性不支持读操作，则指令会报错。</p>
示例	<p>AT+BLEINIT=1 // 初始化为 client</p> <p>AT+BLECONN=0,"24:12:5f:9d:91:98"// 建立 BLE 连接</p> <p>AT+BLEGATTCPRIMSRV=0</p> <p>AT+BLEGATTCCHAR=0,3//根据前一条指令的查询结果，指定 index 查询</p> <p>AT+BLEGATTCRD=0,3,2,1//例如，读取第 3 号服务的第 2 号特征的第 1 号描述符信息</p>

5.2.25. AT+BLEGATTCWR—GATTC 写服务特征值

指令	<p>设置指令：</p> <p>AT+BLEGATTCWR=<conn_index>,<srv_index>,<char_index>[,<desc_index>],<length></p> <p>功能：GATTC 写服务特征（描述符）值。</p>
响应	<p>收到此命令后先换行返回 >，然后开始接收串口数据，当数据长度满 <length> 时，执行写操作。若写操作成功，则提示 OK</p>
参数说明	<p><conn_index>：BLE 连接号，当前只支持 index 为 0 的单连接</p> <p><srv_index>：服务发现结果序号，由 AT+BLEGATTCPRIMSRV=<conn_index> 查询结果中获得</p> <p><char_index>：服务特征的序号，由</p> <p>AT+BLEGATTCCHAR=<conn_index>,<srv_index> 查询结果中获得</p> <p>[<desc_index>]（选填参数）：特征描述符序号。若不设置，则写特征值；若设置，写描述符的值。</p> <p><length>：数据长度</p>
注意	<p>• 使用本指令，需要先建立 BLE 连接。</p>

指令 设置指令：

AT+BLEGATTCWR=<conn_index>,<srv_index>,<char_index>[,<desc_index>],<length>

功能：GATTC 写服务特征（描述符）值。

• 如果该服务特征（描述符）属性不支持写操作，则指令会报错。

示例 AT+BLEINIT=1 // 初始化为 client

AT+BLECONN=0,"24:12:5f:9d:91:98"// 建立 BLE 连接

AT+BLEGATTCPRIMSRV=0

AT+BLEGATTCCHAR=0,3//根据前一条指令的查询结果，指定 index 查询

// 例如，向第 3 号服务的第 4 号特征，写入长度为 6 的数据

AT+BLEGATTCWR=0,3,4,,6

// 提示 > 后，通过串口输入数据 "123456" 即可

5.2.26. AT+DTM—BLE 测试模式

指令	<p>设置指令：</p> <p>AT+DTM=TX,<channel>,<data length>,<packet type></p> <p>功能：测试 BLE TX。</p> <p>AT+DTM=RX,<channel></p> <p>功能：测试 BLE TX。</p> <p>AT+DTM=END</p> <p>功能：结束 BLE 测试模式。</p>
响应	<p>执行 AT+DTM=END 结束 BLE 测试模式时才会显示结果</p> <p>RX CNT: AAA</p> <p>CRC OK: BBB</p> <p>CRC FAIL: CCC</p> <p>RSSI: DDD</p> <p>AAA：收到封包总数</p> <p>BBB：期间收到 CRC 正确封包数</p> <p>CCC：期间收到的 CRC 错误封包数</p> <p>DDD：RSSI 值</p>
参数说明	<p><channel>：频道</p> <ul style="list-style-type: none">● 0~39 <p><data length>：封包长度</p> <p><packet type>：封包内容格式</p> <ul style="list-style-type: none">● 0: PRBS9● 1: Pattern 11110000● 2: Pattern 10101010● 3: PRBS15● 4: Pattern 11111111● 5: Pattern 00000000
示例	<p>// TX 端装置</p> <p>AT+DTM=TX, 20, 27, 2</p> <p>AT+DTM=END</p> <p>// RX 端装置</p>

指令	设置指令： AT+DTM=TX,<channel>,<data length>,<packet type> 功能：测试 BLE TX。 AT+DTM=RX,<channel> 功能：测试 BLE TX。 AT+DTM=END 功能：结束 BLE 测试模式。
	AT+DTM=RX, 20 AT+DTM=END

5.2.27. AT+BLEPHYREAD—讀取目前連線的 PHY

指令	设置指令： AT+BLEPHYREAD=< conn_index > 功能：讀取目前連線的 PHY setting。
响应	OK
参数说明	<conn_index>：BLE 连接号，当前只支持 index 为 0 的单连接
注意	使用本指令，需要先建立 BLE 连接。
示例	AT+BLEINIT=2 // 初始化为 server AT+BLEADVSTART // 開始廣播, 等待連線 AT+BLEPHYREAD=0 +BLEPHYREAD:0,0,1,1 // <conn_index>,<status>,<phy_tx>,<phy_rx>
响应参数	Status: 0: 成功 -1: 失敗 (沒有 phy_tx, phy_rx) Phy_tx: 1: 1M 2: 2M Phy_rx: 1: 1M 2: 2M

5.2.28. AT+BLEPHYSET—設置目前連線的 PHY

指令	设置指令： AT+BLEPHYSET=< conn_index > < phy_tx > < phy_rx > 功能：設置目前連線的 PHY setting。
响应	OK
参数说明	<conn_index>：BLE 连接号，当前只支持 index 为 0 的单连接

指令	设置指令： AT+BLEPHYSET=< conn_index > < phy_tx > < phy_rx> 功能：設置目前連線的 PHY setting。
	< phy_tx >: 1: 1M 2: 2M 3: 1M+2M < phy_rx >: 1: 1M 2: 2M 3: 1M+2M
注意	使用本指令，需要先建立 BLE 连接。
示例	AT+BLEINIT=2 // 初始化为 server AT+BLEADVSTART // 開始廣播, 等待連線 AT+BLEPHYSET=0,2,2 +BLEPHYSET:0,0,2,2 // <conn_index>,<status>,<phy_tx>,<phy_rx>
响应参数	Status: 0: 成功 -1: 失敗 (沒有 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
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)

Code	Description
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. AT 指令使用示例

本章介绍几种常见的 AT 指令使用示例。

6.1. 单连接 TCP 客户端

6.1.1. 建立连接

- 1 · 设置 Wi-Fi 模式：

```
AT+CWMODE=1      // Station mode
```

响应：

OK

- 2 · 扫描 AP：

```
AT+CWLAP
```

响应：

```
+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 · 连接路由：

```
AT+CWJAP="SSID","password"
```

响应：

OK

WIFI CONNECTED

WIFI GOT IP

- 4 · 查询设备 IP 信息：

```
AT+CIFSR
```

响应：

```
+CIFSR:STAIP,"169.254.119.102"
```

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

OK

- 5 · 设置 PC 与 OPL000 连接同一个路由，在 PC 上使用网络调试助手，如图 Figure 1，创建一个 TCP 服务器：

Figure 1：网络调试助手建立 TCP 服务器



- 6 · OPL00 作为客户端连接到 TCP 服务器：

AT+CIPSTART="TCP","192.168.1.113",8181 // protocol、server IP & port

响应：

CONNECT

OK

6.1.2. OPL1000 作为 tcp client 发送数据

- 1 · 发送数据：

AT+CIPSEND=4

OK

>ABCD

Recv 4 bytes

SEND OK

2 · 在 tcp server 上收到的数据如图 Figure 2 所示 · 接收到数据 ABCD :

Figure 2: tcp server 接收数据



6.1.3. OPL1000 作为 tcp client 接收数据

1 · 在 tcp server 上发送数据 · 如图 Figure 3 发送框中所示 :

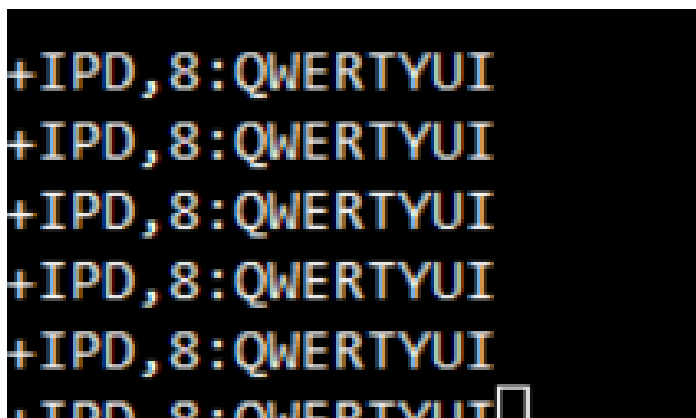
Figure 3: 发送数据框



2 · OPL1000 接收数据 · 如图 Figure 4 所示 :

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

Figure 4 : OPL1000 接收数据



6.2. 单连接 TCP 服务器

6.2.1. 建立连接

- 1 · 设置 Wi-Fi 模式：

```
AT+CWMODE=1          // Station mode
```

响应：

OK

- 2 · 扫描 AP：

```
AT+CWLAP
```

响应：

```
+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 · 连接路由：

```
AT+CWJAP="SSID","password"
```

响应：

OK

WIFI CONNECTED

WIFI GOT IP

- 4 · 查询设备 IP 信息：

AT+CIFSR

响应：

+CIFSR:STAIP,"169.254.119.102"

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

OK

- 5 · 设置 PC 与 OPL000 连接同一个路由，在 AT 串口上使用 AT 命令创建一个 TCP 服务器：

AT+CIPMUX=1

响应：

OK

AT+CIPSERVER=1,23456 //假设设置的端口号是 23456

响应：

OK

- 6 · 利用 NetAssisant 作为客户端连接到 TCP 服务器，如图 Figure 5：

Figure 5：客户端连接到 TCP 服务器



连接成功响应：0 · CONNECT

6.2.2. OPL1000 作为 tcp server 发送数据

- 1 · OPL1000 发送数据：
AT+CIPSEND=0,4 //建立服务器的时候需要多连接，多连接设备有编号，只有一个设备，所以编号为 0.
OK
>ABCD
Recv 4 bytes

SEND OK

- 2 · 客户端接收数据，接收数据如图 Figure 6 所示：

Figure 6：客户端接收数据



6.2.3. OPL1000 作为 tcp server 接收数据

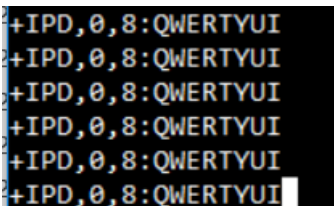
- 1 · 在 tcp client 端发送数据 · 如图 Figure 7 框中：

Figure 7: tcp client 端发送数据



- 2 · OPL1000 作为 tcp server 接收数据,如图 Figure 8 所示：
+IPD,n:xxxxx // received n bytes, data=xxxxx

Figure 8: 接收数据



6.3. UDP 传输

1. 设置 Wi-Fi 模式：

AT+CWMODE=1 // Station mode

响应：

OK

2. 扫描 AP：

AT+CWLAP

响应：

+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. 连接路由：

AT+CWJAP=" SSID" ," password"

响应：

OK

WIFI CONNECTED

WIFI GOT IP

4. 查询设备 IP 信息：

AT+CIFSR

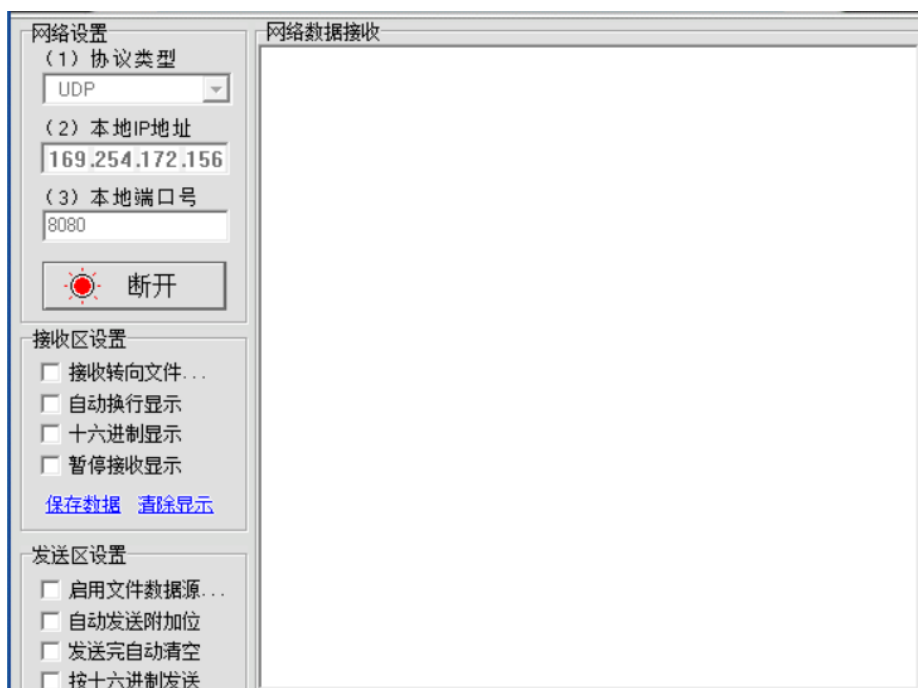
响应：

+CIFSR:STAIP,"169.254.119.102"

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

OK

5. 设置 PC 与 OPL000 连接同一个路由，在 PC 上使用网络调试助手，创建 UDP 传输：



下面介绍两种 UDP 通信的示例：

6.3.1. 固定远端的 UDP 通信

UDP 通信的远端固定，由 AT+CIPSTART 指令的最后参数 0 决定，分配一个连接号给这个固定连接，在通信过程中远端信息不会被改变。

1. 使能多连接：

AT+CIPMUX=1

响应：

OK

2. 创建 UDP 传输，例如，分配连接 ID 为 4。

AT+CIPSTART=4,"UDP","192.168.1.101",8080,1112,0

响应：

4,CONNECT

OK

3. 发送数据：

AT+CIPSEND=4,5

OK

> ABCDE

Recv 5 bytes

SEND OK

4. 接收数据：

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

5. 断开 UDP 传输：

4,CLOSED

OK

6.3.2. 远端可变的 UDP 通信

1. 创建 UDP 传输，最后参数为 2：

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

响应：

CONNECT

OK

2. 发送数据

AT+CIPSEND=5

OK

> ABCDE

Recv 5 bytes

SEND OK

3. 接收数据：

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

4. 断开 UDP 传输：

0,CLOSED

OK

6.4. 多连接 TCP 服务器

6.4.1. 建立连接

目前 OPL000 仅支持建立一个 TCP 服务器，且必须使能多连接。

因为 OPL000 只能作为 Station，所以需要连接路由后再建立服务器。

1. 设置 Wi-Fi 模式：

AT+CWMODE=1

响应：

OK

2. 扫描 AP：

AT+CWLAP

响应：

+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. 连接路由：

AT+CWJAP=" SSID" ," PASSWD"

响应：

OK

WIFI CONNECTED

WIFI GOT IP

4. 查看 IP 信息：

AT+CIFSR

响应：

+CIFSR:STAIP,"192.168.1.103"

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

OK

5. 使能多连接:

AT+CIPMUX=1

响应:

OK

6. 建立 TCP SERVER:

AT+CIPSERVER=1,8080

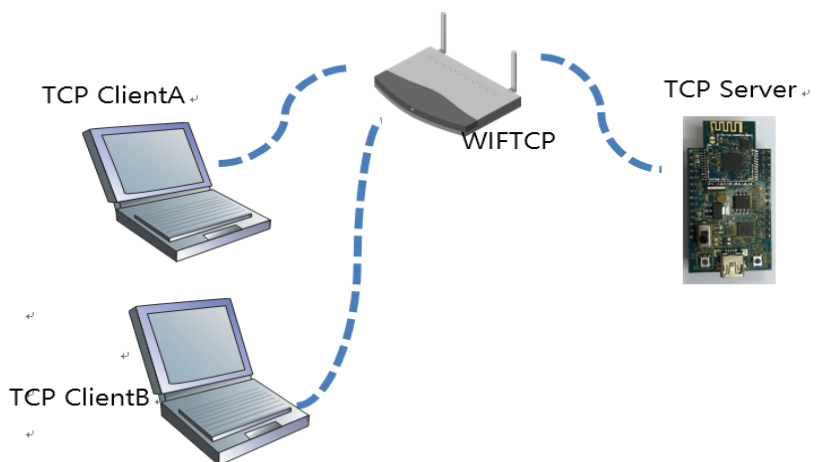
响应:

OK

0,CONNECT //在 PC 上建立 TCP Client 并连接后显示

7. 进行如下网络设置,本节用两台电脑为例,在两台 PC 同时运行网络调试助手, 连接到 tcp server。
多个客户端连接服务器网络连接图如图 Figure 9。

Figure 9: 多个客户端连接服务器的网络连接图



8. 利用 NetAssisant 作为客户端连接到 TCP 服务器,并且同时打开多个 Client , 在多个电脑上同时打开网络调试助手 , 即开启多个客户端,即上图中的 ClientA 和 ClientB 。

如下图錯誤! 找不到參照來源。所示打开 ClientA 和 ClientB 上的网络调试助手。

Figure 10: 网络调试助手设置参数界面



9. 建立 TCP 服务器 , 输入如下 AT 命令 :

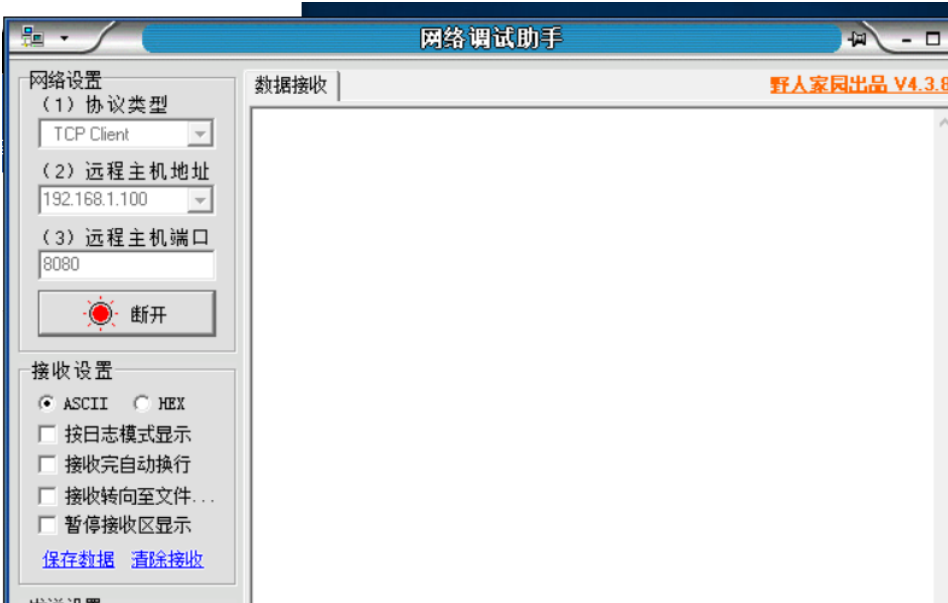
AT+CIPSERVER=1,8080

响应 :

OK

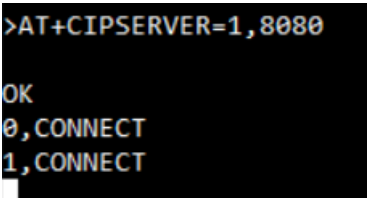
10. 在 ClientA 上点击黑色的“连接”按钮 , 即连接 , 此时显示的是红色的“断开”按钮 , 即建立连接 :

Figure 11: Client 端连接 Server



- 11. 点击上述 ClientA 中的按钮后，Tcp Server 端显示，即 server 和 ClientA 建立连接：
0，CONNECT //ClientA 的终端编号是 0
- 12. 同样点击上述 ClientB 中的按钮后，Tcp Server 端显示，即 server 和 ClientB 建立连接：
1，CONNECT //ClientB 的终端编号是 1
- 13. 客户端用编号区别显示,上述多设备都连接成功后，如图 Figure 12，显示 CONNECT:

Figure 12：连接成功



6.4.2. OPL1000 作为 tcp server 发送数据

- 1. 发送给 ClientA 设备 4 字节：
AT+CIPSEND=0,4 //多连接设备有编号，ClientA 的设备编号是 0，向 ClientA 发送 4 字节数据

OK

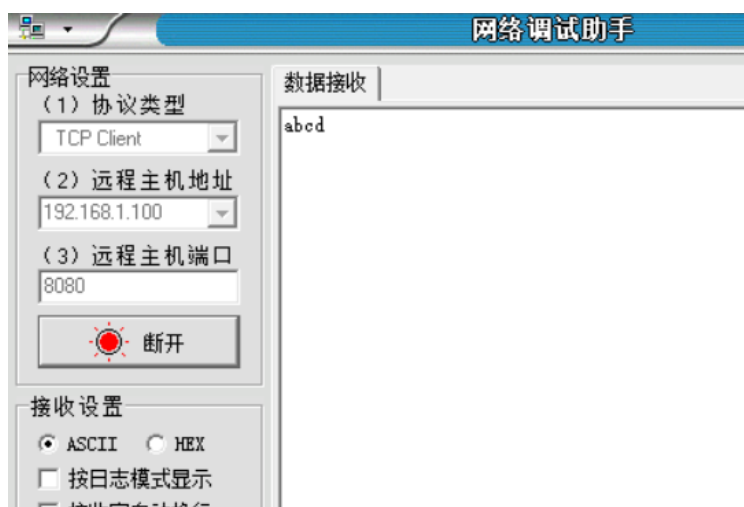
>ABCD

Recv 4 bytes

SEND OK

其中，ClientA 端收到的数据如图 Figure 13：

Figure 13: ClientA 数据接收



2. 发送给 ClientB 设备 6 字节：

AT+CIPSEND=1,6 //多连接设备有编号，ClientB 的设备编号是 1，向 ClientB 发送 6 字节数据

OK

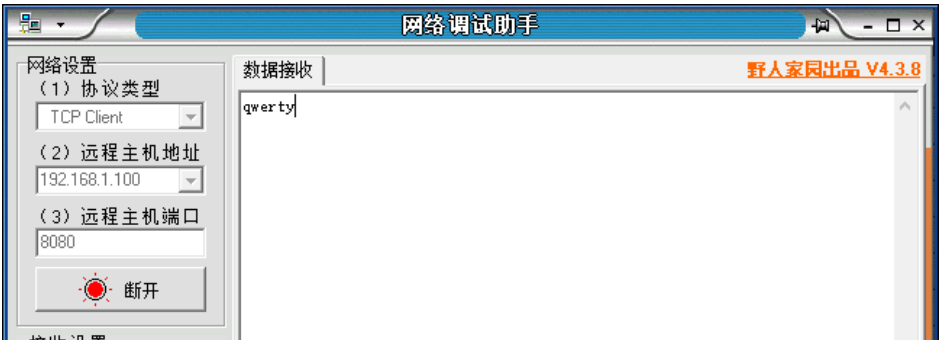
>qwerty

Recv 6 bytes

SEND OK

其中，ClientB 端收到的数据如图 Figure 14：

Figure 14: ClientB 数据接收



6.4.3. OPL1000 作为 tcp server 接收数据

- 1. 在 client 端发送数据，OPL1000 接收数据

+IPD,0,17 : 123456789qwertyui

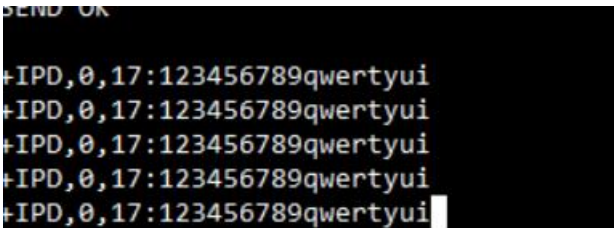
客户端发送框如下图 Figure 15 :

Figure 15: 发送数据



- 2. Opl1000 成功接收到的数据如下图 Figure 16，其中 IPD 后面的编号显示的是 0 设备，如果是 1 设备发送的数据，则显示编号 1：

Figure 16: opl1000 接收数据



6.5. Wi-Fi 透传模式

AT 指令提供 OPL1000 作为 TCP 客户端单连接或 UDP 固定远程目标传输时之透传功能。

6.5.1. TCP 客户端单连接透传模式

以下为 OPL1000 作为 TCP 客户端单连接透传的示例。

- 1. 设置为 Wi-Fi station 模式：

```
AT+CWMODE=1 // Wi-Fi station mode
```

响应：

```
OK
```

- 2. 扫描所有 AP

```
AT+CWLAP
```

响应：

```
列出所有扫描到的 AP。
```

- 3. 连接 AP

```
AT+CWJAP="SSID","PASSPHRASE"
```

响应：

```
OK
WIFI CONNECTED
WIFI GOT IP
```

- 4. 查询设备 IP 地址

```
AT+CIFSR
```

响应：

```
+CIFSR:STAIP,"192.168.1.133"
+CIFSR:STAMAC,"22:33:44:55:66:76"
OK
```

- 5. PC 端与设备连接同一个 AP，在 PC 端上建立 TCP 服务器。如：PC 的 IP 地址为 192.168.1.134 端口为 8080。
- 6. 设备作为 TCP 客户端并连接 TCP 服务器

```
AT+CIPSTART="TCP","192.168.1.134",8080
```

响应：

```
OK
```

- 7. 设置透传模式

```
AT+CIPMODE=1
```

响应：

```
OK
```

- 8. 发送数据

```
AT+CIPSEND
```

响应：

```
OK  
> //此时已进入透传模式，数据将从 UART 输入并透过透传传输至服务器。
```

- 9. 离开透传模式

```
+++
```

注意：

- 1. 若收到单独一包数据+++，则会离开透传发送。请至少间隔 1 秒钟输入下一道 AT 指令。
- 2. 若是从键盘输入+++指令，有可能时间太慢不会被认为是连续的+。建议使用复制贴上的方式离开透传发送。
- 3. 离开透传传送后，此时 TCP socket 仍然保持连接的。

10. 设置一般模式

```
AT+CIPMODE=0
```

响应：

```
OK
```

11. 断开 TCP 连接

```
AT+CIPCLOSE
```

响应：

```
0,CLOSED
```

```
OK
```

6.5.2. UDP 透传模式

以下为 OPL1000 作为 UDP 固定远程目标透传的示例。

1. 设置为 Wi-Fi station 模式：

```
AT+CWMODE=1 // Wi-Fi station mode
```

响应：

```
OK
```

2. 扫描所有 AP

```
AT+CWLAP
```

响应：

```
列出所有扫描到的 AP。
```

3. 连接 AP

```
AT+CWJAP="SSID","PASSPHRASE"
```

响应：

```
OK
WIFI CONNECTED
WIFI GOT IP
```

4. 查询设备 IP 地址

```
AT+CIFSR
```

响应：

```
+CIFSR:STAIP,"192.168.1.133"
+CIFSR:STAMAC,"22:33:44:55:66:76"
OK
```

5. PC 端与设备连接同一个 AP，PC 与设备建立 UDP 固定端口的联机。如：PC 的 IP 地址为 192.168.1.134 端口为 23456。

```
AT+CIPSTART="UDP","192.168.1.134",23456
```

响应：

OK

6. 设置透传模式

AT+CIPMODE=1

响应：

OK

7. 发送数据

AT+CIPSEND

响应：

OK

> //此时已进入透传模式，数据将从 UART 输入并透过透传传输至目标。

8. 离开透传模式

+++

注意：

1. 若收到单独一包数据+++，则会离开透传发送。请至少间隔 1 秒钟输入下一道 AT 指令。
2. 若是从键盘输入+++指令，有可能时间太慢不会被认为是连续的+。建议使用复制贴上的方式离开透传发送。
3. 离开透传发送后，此时 UDP socket 仍然保持建立的。

9. 设置一般模式

```
AT+CIPMODE=0
```

响应：

```
OK
```

10. 断开 UDP 连接

```
AT+CIPCLOSE
```

响应：

```
0,CLOSED
```

```
OK
```

6.6. AUXADC 校準

AUXADC 的數電壓與輸出值為線性, 可以利用兩個已知點推算轉換的直線方程式

可以由 `at+adcvbat` 與 `at+adcgpio` 指令來取得校準後的電壓推估值.

OPL_1000 提供[預設校準] 與額外 [輸入電壓校準] 兩種方式:

6.6.1. 適用限制

從 `vbat` 校準時須注意不可低於最低工作電壓

如果是從 `GPIO` 校準, 僅適用外接非電阻分壓類型的電路

6.6.2. 預設校準

調用校準初始化(`Hal_Aux_AdcCal_Init`)後,會根據以下的順序決定校準值:

- 1) `Flash` 是否有記錄過校準值
- 2) 是否有 `OTP` 校準值可以參考.
- 3) 內部電壓準位. 採用 `LDO_RF` (1100mV)與 `VSS`(0mV)

6.6.3. 輸入電壓校準

利用 `at+adccalvbat` 與 `at+adccalgpio` 進行額外輸入電壓的校準.

6.6.3.1. 輸入電壓建議值

首先, `GPIO` 要先配置成 `input`.

輸入的校準電壓值比 `LDO_RF`(1100mV)大者為佳,

如果條件允許, 建議 3000mV 會獲得比較準確的結果.

操作範例: 將 GPIO_10 修改成 input 後 -> 接好輸入電壓 3V -> 輸入指令" at+adccalgpio=10,3000,1", 即完成校準

如果條件許可多一組校正電壓輸入, 建議採用 500mv

操作範例: 將接好輸入電壓 0.5V -> 輸入指令" at+adccalgpio=10,500,0", 即完成校準

6.6.3.2. 儲存/重讀/恢復預設校正值

校準後的數值需要使用指令 at+adcstore 來儲存到 flash 上,

如果校準後不要採用, 可以利用 at+adcreload 來讀取 flash 的紀錄值,

或是利用 at+adcdef 重置成內部校準

CONTACT

sales@Opulinks.com