

GigaDevice Semiconductor Inc.

GD32VW553 AT 指令用户指南

应用笔记

AN151

1.3 版本

(2025 年 3 月)

目录

目录.....	2
表索引.....	5
1. AT 指令格式.....	7
1.1. 指令类型.....	7
1.2. 指令格式.....	7
1.3. 响应格式.....	7
2. AT 指令一览表.....	8
3. AT 基础指令集.....	10
3.1. AT	10
3.2. ATQ.....	10
3.3. AT+HELP	10
3.4. AT+RST	11
3.5. AT+GMR	11
3.6. AT+TASK.....	11
3.7. AT+HEAP.....	12
3.8. AT+SYSRAM.....	12
3.9. AT+UART.....	12
3.10. AT+TRANSINTVL	13
4. AT WIFI 指令集.....	14
4.1. AT+CWMODE_CUR.....	14
4.2. AT+CWJAP_CUR.....	14
4.3. AT+CWLAP.....	15
4.4. AT+CWSTATUS	15
4.5. AT+CWQAP	16
4.6. AT+CWSAP_CUR	16
4.7. AT+CWLIF	17
4.8. AT+CWAUTOCONN.....	17
5. AT TCPIP 指令集	18
5.1. AT+PING.....	18

5.2.	AT+CIPSTA.....	18
5.3.	AT+CIPSTART	19
5.4.	AT+CIPSEND	20
5.5.	AT+CIPSERVER	21
5.6.	AT+CIPCLOSE.....	22
5.7.	AT+CIPSTATUS	22
5.8.	AT+CIFSR	22
5.9.	AT+CIPMODE	23
6.	AT BLE 指令集.....	24
6.1.	AT+BLEENABLE	24
6.2.	AT+BLEDISABLE	24
6.3.	AT+BLENAME	24
6.4.	AT+BLEADVSTART.....	25
6.5.	AT+BLEADVSTOP	25
6.6.	AT+BLEADVDATA.....	26
6.7.	AT+BLEADVDATAEX	26
6.8.	AT+BLESCANRSPDATA	27
6.9.	AT+BLEPASSTH	27
6.10.	AT+BLEPASSTHAUTO	27
6.11.	AT+BLEPASSTHCLI.....	28
6.12.	AT+BLESCANPARAM	28
6.13.	AT+BLESCAN	29
6.14.	AT+BLESYNC.....	29
6.15.	AT+BLESYNCSTOP	30
6.16.	AT+BLECONN	30
6.17.	AT+BLESCONNPARAM	31
6.18.	AT+BLEDISCONN	31
6.19.	AT+BLEM TU.....	32
6.20.	AT+BLEPHY	32
6.21.	AT+BLEDATALEN.....	33
6.22.	AT+BLEADDR	34

6.23.	AT+BLESETAUTH	34
6.24.	AT+BLEPAIR	35
6.25.	AT+BLEENCRYPT	36
6.26.	AT+BLEPASSKEY	36
6.27.	AT+BLECOMPARE.....	37
6.28.	AT+BLELISTENCDEV	37
6.29.	AT+BLECLEARENCDEV	38
6.30.	AT+BLEGATTSSVC	38
6.31.	AT+BLEGATTSSCHAR.....	39
6.32.	AT+BLEGATTSDISC	39
6.33.	AT+BLEGATTSLISTALL	40
6.34.	AT+BLEGATTSENTF	40
6.35.	AT+BLEGATTSSIND.....	41
6.36.	AT+BLEGATTSSSETATTRVAL.....	41
6.37.	AT+BLEGATTCDISCSVC	42
6.38.	AT+BLEGATTCDISCCHAR	43
6.39.	AT+BLEGATTCDISCDESC.....	43
6.40.	AT+BLEGATTSCRD	44
6.41.	AT+BLEGATTTCWR.....	44
7.	版本历史	46

表索引

表 1-1. 指令类型	7
表 1-2. 指令格式	7
表 1-3. 响应格式	7
表 2-1. AT 指令	8
表 3-1. 进入 AT 指令模式	10
表 3-2. 离开 AT 指令模式	10
表 3-3. 查询所有 AT 指令	10
表 3-4. 模块复位指令	11
表 3-5. 查询版本信息	11
表 3-6. 查询当前操作系统所有任务	11
表 3-7. 查询当前操作系统空余 HEAP	12
表 3-8. 查询当前空余 SRAM 空间	12
表 3-9. 查询或设置串口参数	12
表 3-10. 设置或查询透传模式下的数据发送间隔	13
表 4-1. 查询或设置 WiFi 当前工作模式	14
表 4-2. 查询已连接 AP 信息或连接 AP	14
表 4-3. 扫描并列周围 AP 的信息	15
表 4-4. 查询 WiFi 状态, STA 或者 SoftAP 或者 MONITOR	15
表 4-5. 断开 AP	16
表 4-6. 启动 SoftAP	16
表 4-7. 查看连接上 SoftAP 的客户端	17
表 4-8. 设置开机是否自动连接 AP	17
表 5-1. Ping 功能	18
表 5-2. 查询或设置本地 STA 的 IP 地址	18
表 5-3. 建立 TCP 连接或 UDP 传输	19
表 5-4. 发送数据	20
表 5-5. 启动 TCP 服务器	21
表 5-6. 关闭 TCP 连接或 UDP 传输	22
表 5-7. 查询网络连接信息	22
表 5-8. 查询本地 IP 地址信息	22
表 5-9. 设置或查询传输模式	23
表 6-1. 使能 ble 模块	24
表 6-2. 失能 ble 模块	24
表 6-3. 设置名称	24
表 6-4. 开启蓝牙广播	25
表 6-5. 停止蓝牙广播	25
表 6-6. 设置广播内容	26
表 6-7. 设置广播内容	26
表 6-8. 设置扫描回复内容	27
表 6-9. 开启透传模式	27

表 6-10. 自动开启透传模式.....	27
表 6-11. 设置扫描参数.....	28
表 6-12. 开启扫描.....	29
表 6-13. BLE 开始/取消同步.....	29
表 6-14. BLE 停止同步.....	30
表 6-15. BLE 建立连接.....	30
表 6-16. 设置/查询连接参数.....	31
表 6-17. BLE 断开连接.....	31
表 6-18. 更新/查询 mtu.....	32
表 6-19. 更新/查询 phy.....	32
表 6-20. Data length extension.....	33
表 6-21. 查询/设置 ble bd address.....	34
表 6-22. 配置 AUTHENTICATION.....	34
表 6-23. 发起配对.....	35
表 6-24. 启动加密.....	36
表 6-25. 输入 passkey.....	36
表 6-26. 输入 compare 结果.....	37
表 6-27. 列出 bond device 列表.....	37
表 6-28. 移除 bond 设备.....	38
表 6-29. 列出本地注册的 service.....	38
表 6-30. 列出对应 service 的 characteristic.....	39
表 6-31. 列出对应 characteristic 的 descriptor.....	39
表 6-32. 列出本地所有 service 中信息.....	40
表 6-33. 发送 notification.....	40
表 6-34. 发送 indication.....	41
表 6-35. 设置 characteristic 的值.....	41
表 6-36. 发现 service.....	42
表 6-37. 发现 characteristic.....	43
表 6-38. 发现 descriptor.....	43
表 6-39. Read attribute value.....	44
表 6-40. Write attribute value.....	44
表 7-1. 版本历史.....	46

1. AT 指令格式

1.1. 指令类型

表 1-1. 指令类型

类型	格式	描述
帮助指令	AT+<x>=?	查看指令参数及取值范围
查询指令	AT+<x>?	查询指定目标的当前参数值
执行指令	AT+<x> 或 AT+<x>=<...>	运行命令 设置指定目标参数值

1.2. 指令格式

表 1-2. 指令格式

字段	说明
AT	指令前缀
<CMD>	指令字符串
[]	可选部分
<>	强制部分，针对特定命令，有些参数是强制要输入的
[p1],[p2],[p3],...	参数，参数支持字符串和数字两种，IP 地址采用字符串“x.x.x.x”格式输入 字符串：必须用双引号括起来 数字：支持十进制和十六进制

Note: AT [+<CMD>] [=] [p1],[p2],[p3],...

1.3. 响应格式

表 1-3. 响应格式

输出类型	说明
[+<CMD>:<MSG>]	输出结果或错误提示
<RSP>	OK: 代表成功 ERROR: 代表失败

注意：响应格式里面的汉字仅仅是对命令响应的解释，实际上不会显示。

2. AT 指令一览表

表 2-1. AT 指令

指令	描述
AT	进入 AT 指令模式
ATQ	离开 AT 指令模式
AT+HELP	查询所有 AT 指令
AT+RST	模块复位
AT+GMR	查询版本信息
AT+TASK	查询当前操作系统所有任务
AT+HEAP	查询当前操作系统空余 HEAP
AT+SYSRAM	查询当前空余 SRAM 空间
AT+UART	设置 LOG UART 参数或读取当前参数
AT+TRANSINTVL	查询或设置透传模式下的数据发送间隔
AT+CWMODE_CUR	查询或设置 WiFi 当前工作模式：SoftAP 或 STA
AT+CWJAP_CUR	连接 AP
AT+CWLAP	扫描并显示 AP 列表
AT+CWSTATUS	查询 WiFi 当前工作模式和状态
AT+CWQAP	断开与 AP 的连接
AT+CWSAP_CUR	启动 SoftAP 模式
AT+CWLIF	查询所有连接到 SoftAP 的 STA 信息
AT+CWAUTOCONN	设置上电时是否自动连接 AP
AT+PING	Ping 功能
AT+CIPSTA	查询或设置本地 STA 的 IP 地址
AT+CIPSTART	建立 TCP 连接或 UDP 传输
AT+CIPSEND	发送数据
AT+CIPSERVER	启动 TCP 服务器
AT+CIPCLOSE	关闭 TCP 连接或 UDP 传输
AT+CIPSTATUS	查询网络连接信息
AT+CIFSR	查询本地 IP 地址信息
AT+CIPMODE	查询或设置传输模式
AT+BLEENABLE	使能 ble 模块
AT+BLEDISABLE	失能 ble 模块
AT+BLENAM	设置名称
AT+BLEADVSTART	开启蓝牙广播
AT+BLEADVSTOP	停止蓝牙广播
AT+BLEADVDATA	设置广播内容
AT+BLEADVDATA EX	设置广播内容
AT+BLESCANRSPDATA	设置扫描回复内容
AT+BLEPASSTH	开启透传模式
AT+BLEPASSTHAUTO	自动开启透传模式

指令	描述
AT+BLES SCANPARAM	设置扫描参数
AT+BLES SCAN	开启扫描
AT+BLES SYNC	BLE 开始/取消同步
AT+BLES SYNCSTOP	BLE 停止同步
AT+BLES CONN	BLE 建立连接
AT+BLES CONNPARAM	设置/查询连接参数
AT+BLES DISCONN	BLE 断开连接
AT+BLES MTU	更新/查询 mtu
AT+BLES PHY	更新/查询 phy
AT+BLES DATALEN	Data length extension
AT+BLES ADDR	查询/设置 ble bd address
AT+BLES SETAUTH	配置 AUTHENTICATION
AT+BLES PAIR	发起配对
AT+BLES ENCRYPT	启动加密
AT+BLES PASSKEY	输入 passkey
AT+BLES COMPARE	输入 compare 结果
AT+BLES LISTENCDEV	列出 bond device 列表
AT+BLES CLEARCDEV	移除 bond 设备
AT+BLES GATTSVC	列出本地注册的 service
AT+BLES GATTCHAR	列出对应 service 的 characteristic
AT+BLES GATTDESC	列出对应 characteristic 的 descriptor
AT+BLES GATTSLISTALL	列出本地所有 service 中信息
AT+BLES GATTNTF	发送 notification
AT+BLES GATTIND	发送 indication
AT+BLES GATTSETATTRVAL	设置 characteristic 的值
AT+BLES GATTCDISCV C	发现 service
AT+BLES GATTCDISCCHAR	发现 characteristic
AT+BLES GATTCDISCDESC	发现 descriptor
AT+BLES GATTCRD	Read attribute value
AT+BLES GATTCWR	Write attribute value

3. AT 基础指令集

3.1. AT

表 3-1. 进入 AT 指令模式

指令	参数	响应
执行指令 AT		执行结果
示例： AT 正确响应： OK		

3.2. ATQ

表 3-2. 离开 AT 指令模式

指令	参数	响应
执行指令 ATQ		执行结果
示例： ATQ 正确响应： OK		

3.3. AT+HELP

表 3-3. 查询所有 AT 指令

指令	参数	响应
执行指令 AT+HELP		显示全部 AT 命令列表
示例： AT+HELP 正确响应： AT COMMAND LIST: ===== ATQ AT+HELP OK		

3.4. AT+RST

表 3-4. 模块复位指令

指令	参数	响应
执行指令 AT+RST		重启消息
示例: AT+RST 正确响应: OK ALW: MBL: First print. ALW: MBL: Boot from Image 0. ALW: MBL: Validate Image 0 OK. ALW: MBL: Jump to Main Image (0x0800a000). READY		

3.5. AT+GMR

表 3-5. 查询版本信息

指令	参数	响应(类似格式信息)
执行指令 AT+GMR		相关版本信息
示例: AT+GMR 正确响应: ===== SDK revision: v1.0.0 SDK git revision: 0.1.0-487-gb2937736-b2937736b33393b3 SDK build date: 2023/07/03 15:23:20 OK		

3.6. AT+TASK

表 3-6. 查询当前操作系统所有任务

指令	参数	响应(类似格式信息)
执行指令 AT+TASK		当前 task 信息列表
示例: AT+TASK 正确响应: ATCMD X 20 402 2 0x2001a780		

指令	参数	响应(类似格式信息)
...		
RX	B 18 416 6 0x200203c8	
OK		

3.7. AT+HEAP

表 3-7. 查询当前操作系统空余 HEAP

指令	参数	响应(类似格式信息)
执行指令 AT+HEAP		heap 使用情况
示例: AT+HEAP 正确响应: ===== Total free heap size = 113784 Total min free heap size = 109480 OK		

3.8. AT+SYSRAM

表 3-8. 查询当前空余 SRAM 空间

指令	参数	响应(类似格式信息)
执行指令 AT+SYSRAM		剩余 SRAM 空间
示例: AT+SYSRAM 正确响应: ===== Free SRAM size = 108472 OK		

3.9. AT+UART

表 3-9. 查询或设置串口参数

指令	参数	响应
帮助指令 AT+UART=?		+UART=<baudrate>,<databits>,<stopbits>,<parity>,<flow control>
查询指令 AT+UART?		当前串口参数
执行指令	<baudrate>: UART 波特率	执行结果

指令	参数	响应
AT+UART=<baudrate>,<data bits>,<stopbits>,<parity>,<flow control>	<data bits>: 数据位 8: 8 bit <stopbits>: 停止位 1: 1 bit 2: 1.5 bit 3: 2 bit <parity>: 校验位 0: None 1: Odd 2: Even <flow control>: 流控 0: 不使能流控 1: 使能 RTS 2: 使能 CTS 3: 同时使能 RTS 和 CTS	
示例: AT+UART=115200,8,1,0,0 正确响应: OK		

3.10. AT+TRANSINTVL

表 3-10. 设置或查询透传模式下的数据发送间隔

指令	参数	响应
帮助指令 AT+TRANSINTVL=?		+TRANSINTVL=<interval>
查询指令 AT+TRANSINTVL?		当前数据发送间隔 +TRANSINTVL:<interval>
执行指令 AT+TRANSINTVL =<interval>	<interval>: 数据发送间隔, 毫秒, 默认值 20 毫秒	执行结果
示例: AT+TRANSINTVL=800 正确响应: OK		

4. AT WIFI 指令集

4.1. AT+CWMODE_CUR

表 4-1. 查询或设置 WiFi 当前工作模式

指令	参数	响应
帮助指令 AT+CWMODE_CUR=?		+CWMODE_CUR: <mode:0-2>
查询指令 AT+CWMODE_CUR?		当前工作模式 +CWMODE_CUR: <mode>
执行指令 AT+CWMODE_CUR=<mode>	<mode>: 0: MONITOR 模式 1: STA 模式 2: Soft AP 模式	执行结果
示例: AT+CWMODE_CUR=2 正确响应: OK		

4.2. AT+CWJAP_CUR

表 4-2. 查询已连接 AP 信息或连接 AP

指令	参数	响应
帮助指令 AT+CWJAP_CUR=?		+CWJAP_CUR=<ssid>,<pw d>
查询指令 AT+CWJAP_CUR?		+CWJAP_CUR: <ssid>,<mac>,<channel>,<rsi>
执行指令 AT+CWJAP_CUR=<ssid>,<pw d>	<ssid>: 字符串参数 <pw d>: 字符串参数	执行结果
示例 1: AT+CWJAP_CUR="totolink","12345678" 正确响应 1: WIFI CONNECTED OK 示例 2: AT+CWJAP_CUR="tplink","" 正确响应 2: WIFI CONNECTED		

指令	参数	响应
OK		

4.3. AT+CWLAP

表 4-3. 扫描并列出现周围 AP 的信息

指令	参数	响应
帮助指令 AT+CWLAP=?		+CWLAP: [ssid]
执行指令 AT+ CWLAP[=<ssid>]	<ssid>: 字符串参数	扫描结果 +CWLAP: <ssid>,<rssi>,<mac>,<channel>,<enctype>
<p>示例 1:</p> <pre>AT+CWLAP 正确响应 1: +CWLAP: iQOO Neo5, -44, d6:4f:86:cb:c8:d0, 1, WPA2 CCMP; +CWLAP: GD-guest, -43, 08:3a:38:cc:2f:d1, 1, OPEN ; +CWLAP: OpenWrt, -33, c4:70:ab:d9:bd:11, 1, OPEN ; +CWLAP: GD-internet, -44, 08:3a:38:cc:2f:d0, 1, OPEN ; +CWLAP: Redmi K40, -56, ba:fa:07:50:63:f6, 1, WPA2 CCMP; +CWLAP: D-Link_DIR-822, -30, 1c:5f:2b:fd:be:60, 1, WPA2 CCMP; +CWLAP: iPhone 24 Pro Max Ultr, -48, fa:da:47:72:f0:b3, 2, WPA2 CCMP; +CWLAP: TP-LINK_8659, -20, 68:77:24:bd:86:59, 4, WPA2/WPA3 CCMP; OK</pre> <p>示例 2:</p> <pre>AT+CWLAP= "xiaomi_4a" 正确响应 2: +CWLAP: xiaomi_4a, -55, 88:c3:97:0d:c3:70, 1, OPEN OK</pre>		

Note: 如果带参数 ssid, 就只显示相应 AP 信息。

4.4. AT+CWSTATUS

表 4-4. 查询 WiFi 状态, STA 或者 SoftAP 或者 MONITOR

指令	参数	响应
执行指令 AT+CWSTATUS		+CWSTATUS: STA, connected, <ssid>,<channel>,<mac>, 或

指令	参数	响应
		+CWSTATUS: MONITOR, <channel>, <mac> 或 +CWSTATUS: STA, disconnected 或 +CWSTATUS: SoftAP, <ssid>, <password>, <channel>
示例: AT+CWSTATUS 正确响应: +CWSTATUS: STA, connected, xiaomi_4a, 1, 76:ba:ed:20:22:a2 OK		

4.5. AT+CWQAP

表 4-5. 断开 AP

指令	参数	响应
执行指令 AT+CWQAP		断开连接消息
示例: AT+CWQAP 正确响应: OK		

4.6. AT+CWSAP_CUR

表 4-6. 启动 SoftAP

指令	参数	响应
帮助指令 AT+CWSAP_CUR=?		+CWSAP_CUR: <ssid>,<password>,<channel>:1-13>,<hidden>:0-1>
执行指令 AT+CWSAP_CUR=<ssid>,<password>,<channel>,<hidden>	<ssid>: 字符串参数 <password>: 字符串参数 <channel>: 1, 13 <hidden>: 0: SSID Broadcast 1: Hidden SSID	执行结果
示例: AT+CWSAP_CUR="test_ap","12345678",6,0 正确响应: OK		

4.7. AT+CWLIF

表 4-7. 查看连接上 SoftAP 的客户端

指令	参数	响应
执行指令 AT+CWLIF		+CWLIF: [0] <mac1> +CWLIF: [1] <mac2>
示例: AT+CWLIF 正确响应: +CWLIF: [0] e0:2b:e9:8a:46:ac OK		

4.8. AT+CWAUTOCONN

表 4-8. 设置开机是否自动连接 AP

指令	参数	响应
帮助指令 AT+CWAUTOCONN=?		+CWAUTOCONN:(0-1)
查询指令 AT+CWAUTOCONN?		+CWAUTOCONN: <enable>
执行指令 AT+CWAUTOCONN=<enable>	<enable>: 0~1 0: disable auto connect 1: enable auto connect	执行结果
示例: AT+CWAUTOCONN=1 正确响应: OK		
补充说明: +CWAUTOCONN 设置为 1 后, 连接 AP 成功就会将 AP 信息保存到 FLASH 中, 重启后就会自动根据 FLASH 中存储的 AP 信息连接 AP。		

5. AT TCP/IP 指令集

5.1. AT+PING

表 5-1. Ping 功能

指令	参数	响应
帮助指令 AT+PING=?		+PING: <ip or domain name>
执行指令 AT+PING=<ip or domain>	<ip>: 字符串, 可以是 IP 地址或 域名	+<delay_time> +<delay_time>
<p>示例 1: AT+PING="192.168.0.1" 正确响应 1: +80 +47 +49 +55 +53 OK</p> <p>示例 2 注: 使用网址时, 必须要接入互联网, 否则会失败。 AT+PING="w w w.baidu.com" 正确响应 2: +149 +47 +51 +47 +112 OK</p>		

5.2. AT+CIPSTA

表 5-2. 查询或设置本地 STA 的 IP 地址

指令	参数	响应
帮助指令 AT+CIPSTA=?		+CIPSTA: <ip>,<netmask>,<gw >
查询指令 AT+CIPSTA?		+CIPSTA:<ip> +CIPSTA:<netmas k> +CIPSTA:<gw >
执行指令 AT+CIPSTA=<ip>,<netmas	<ip>: 字符串参数 <netmask>: 字符串参数	执行结果

指令	参数	响应
k>,<gw >	<gw >: 字符串参数	
示例 1: AT+CIPSTA? 正确响应 1: +CIPSTA: 192.168.185.1 +CIPSTA: 255.255.255.0 +CIPSTA: 192.168.185.43 OK 示例 2: AT+CIPSTA="192.168.185.45","255.255.255.0","192.168.185.1" 正确响应 2: OK		

5.3. AT+CIPSTART

表 5-3. 建立 TCP 连接或 UDP 传输

指令	参数	响应
帮助指令 AT+CIPSTART=?		+CIPSTART=<type:TCP or UDP>,<remote ip>,<remote port>,[udp local port],[tcp keep alive:0-1]
执行指令 AT+CIPSTART=<type>,<remote ip>,<remote port>,[udp local port],[tcp keep alive]	<type>: "TCP" or "UDP", 字符串参数 <remote ip>: Server IP, 字符串参数 <remote port>: Server Port, 整型 [udp local port] 绑定本机的端口值, 整型 [tcp keep alive]: 0 or 1, 整型	执行结果 OK 或者 ERROR
示例 1: AT+CIPSTART="TCP","192.168.0.2",2001,1 正确响应 1: 0,OK 示例 2: AT+CIPSTART="UDP", "192.168.0.2",5001,0 正确响应 2: 1,OK 示例 3: 固定 UDP 通信对端		

AT+CIPSTART="UDP", "192.168.0.2",5001,8888

正确响应 3:

3,OK

注：该项测试需要在测试机上运行 `sokit` 工具。

5.4. AT+CIPSEND

表 5-4. 发送数据

指令	参数	响应
帮助指令 AT+CIPSEND=?		Usage: Normal Mode Usage: +CIPSEND=<fd:0-4>,<len>,[<remote ip>,<remote port>] PassThrough Mode Usage: +CIPSEND
进入普通传输模式， 执行指令 AT+CIPSEND=<fd>,<len>,[<remote ip>,<remote port>]	<fd>: 0~4，网络连接 ID 号，整型 <len>: <= 2048，发送长度，整型 [remote ip]: 远端 IP，字符串参数 [remote port]: 远端端口，整型	><input from keyboard> SEND OK
进入 WiFi 透传模式，执行 指令 AT+CIPSEND		OK > <input from keyboard>

示例 1:

AT+CIPSEND=0,10

正确响应 1:

>SEND OK

OK

示例 2:

AT+CIPSEND=1,20,"192.168.0.2",5001

正确响应 2:

>SEND OK

OK

示例3: GD32VW553作为TCP 客户端，建立单连接，实现UART Wi-Fi透传连接到路由器

AT+CWJAP="test_ap","1234567890"

查询GD32VW553设备IP地址，以192.168.1.27为例

<p>AT+CIPSTA?</p> <p>PC与GD32VW553设备连接到同一个路由器，并运行网络调试工具，创建一个TCP服务器。例如IP地址为192.168.1.2，端口号为5678。GD32VW553连接该TCP服务器</p> <p>AT+CIPSTART="TCP","192.168.1.2",5678,0</p> <p>进入透传接收模式</p> <p>AT+CIPMODE=1</p> <p>进入透传发送模式，并发送数据</p> <p>AT+CIPSEND</p> <p>OK</p> <p>></p> <p>停止发送数据，在透传发送数据过程中，若识别到单独的一包数据+++，则系统会退出透传发送，此时请至少等待1秒，再发下一条命令。</p> <p>+++</p> <p>退出UART WiFi透传接收模式</p> <p>AT+CIPMODE=0</p> <p>关闭TCP连接</p> <p>AT+CIPCLOSE</p> <p>注：</p> <p>进入 WiFi 透传模式，GD32VW553 设备每次最大接收 8192 字节，最大发送 2920 字节。如果收到的数据长度大于等于 2920 字节，数据会被分为 2920 字节一组的块进行发送，否则会等待 20 毫秒（您可以通过 AT+TRANSINTVL 命令配置此间隔）或等待收到的数据大于等于 2920 字节再发送数据。当输入单独一包+++时，退出透传模式下的数据发送模式，请至少间隔 1 秒再发送下一条 AT 命令。</p> <p>AT+CIPSEND 命令必须在开启透传模式以及单连接下使用。若为 WiFi-UDP 透传，AT+CIPSTART 命令的<udp local port>必须指定。</p> <p>该项测试需要在测试机上运行 sokit 或其他网络测试工具。</p> <p>透传模式仅支持 TCP 单连接和 UDP 固定通信对端的情况。</p>

5.5. AT+CIPSERVER

表 5-5. 启动 TCP 服务器

指令	参数	响应
帮助指令 AT+CIPSERVER=?		+CIPSERVER:<mode:0-1>,[port]
执行指令 AT+CIPSERVER=<mode>, [port]	<mode>: 0: 关闭服务器 1: 建立服务器 [port]: 可选参数，整型	执行结果
示例:		

AT+CIPSERVER=1,3001

正确响应:

OK

5.6. AT+CIPCLOSE

表 5-6. 关闭 TCP 连接或 UDP 传输

指令	参数	响应
帮助指令 AT+CIPCLOSE=?		+CIPCLOSE: <fd>
执行指令 AT+CIPCLOSE=<fd>	<fd>: 0-7, 网络连接 ID 号, 整型	close <fd>
示例: AT+CIPCLOSE=0 正确响应 close 0 OK		

5.7. AT+CIPSTATUS

表 5-7. 查询网络连接信息

指令	参数	响应
执行指令 AT+CIPSTATUS		STATUS: 5
示例: AT+CIPSTATUS 正确响应: STATUS: 2 OK		
补充说明: STATUS 2: STA 已和 AP 建立连接并且获得 IP 地址 3: STA 已建立 TCP 连接或 UDP 传输客户端 4: DHCP 处理中 5: 其他状态		

5.8. AT+CIFSR

表 5-8. 查询本地 IP 地址信息

指令	参数	响应
执行指令 AT+CIFSR		+CIFSR:APIP,<ip> +CIFSR:APMAC,<mac> Or

指令	参数	响应
		+CIFSR: STA IP,<ip> +CIFSR: STA MAC,<mac>
示例: AT+CIFSR 正确响应: +CIFSR: STA IP,192.168.2.3 +CIFSR: STA MAC,76:ba:ed:20:22:a2 OK		

5.9. AT+CIPMODE

表 5-9. 设置或查询传输模式

指令	参数	响应
帮助指令 AT+CIPMODE=?		+CIPMODE=<mode:0-1>
查询指令 AT+CIPMODE?		当前传输模式 +CIPMODE: <mode>
执行指令 AT+CIPMODE =<mode>	<mode>: 传输模式 0: 正常传输模式 1: WiFi 透传接收模式	执行结果
示例: AT+CIPMODE=1 正确响应: OK 注: WiFi 透传接收模式, 仅支持 TCP 单连接、UDP 固定通信对端的情况。 WiFi 透传接收模式, 每次接收的数据最大长度是 2920 字节。		

6. AT BLE 指令集

6.1. AT+BLEENABLE

表 6-1. 使能 ble 模块

指令	参数	响应
执行指令 AT+BLEENABLE		执行结果
示例 1: AT+BLEENABLE 正确响应: OK		

6.2. AT+BLEDISABLE

表 6-2. 失能 ble 模块

指令	参数	响应
执行指令 AT+BLEDISABLE		执行结果
示例 1: AT+BLEDISABLE 正确响应: OK		

6.3. AT+BLENAME

表 6-3. 设置名称

指令	参数	响应
帮助指令 AT+BLENAME=?		+BLENAME=<name>
查询指令 AT+BLENAME?		+BLENAME: <name>
执行指令 AT+BLENAME=<name>	<name>: 设备名称	执行结果
示例 1: AT+BLENAME? 正确响应 1: +BLENAME:GD-BLE-01:23:45:67:89:ab OK		

指令	参数	响应
示例 2: AT+BLENAM=est 正确响应 2: OK 备注: 1. 设置后广播中 name 也会同步改变。		

6.4. AT+BLEADVSTART

表 6-4. 开启蓝牙广播

指令	参数	响应
帮助指令 AT+BLEADVSTART=?		+BLEADVSTART=<type>,[intv],[ch_map],[prop],[pri_phy],[sec_phy],[wl_enable],[own_addr_type],[disc_mode],[addr_type],[addr]
执行指令 AT+BLEADVSTART=<type>,[intv],[ch_map],[prop],[pri_phy],[sec_phy],[wl_enable],[own_addr_type],[disc_mode],[addr_type],[addr]	<type>: 广播类型 [intv]: 广播间隔 [ch_map]: 信道选择 [property]: 属性配置 [pri_phy]: primary 信道 phy [sec_phy]: secondary 信道 phy [wl_enable]: white list 是否使能 [own_addr_type]: 本地地址类型 [disc_mode]: 发现模式 [addr_type]: 对端地址类型 [addr]: 对端地址	执行结果
示例 1: AT+BLEADVSTART=0 正确响应 1: OK		

6.5. AT+BLEADVSTOP

表 6-5. 停止蓝牙广播

指令	参数	响应
帮助指令 AT+BLEADVSTOP=?		+BLEADVSTOP=<adv_idx>
执行指令 AT+BLEADVSTOP=<	< adv_idx > : 广播索引	执行结果

指令	参数	响应
adv_idx >		
示例 1: AT+BLEADVSTOP=0 正确响应 1: OK		

6.6. AT+BLEADVDATA

表 6-6. 设置广播内容

指令	参数	响应
帮助指令 AT+BLEADVDATA=?		+BLEADVDATA=<data>
执行指令 AT+BLEADVDATA=<data>	<data> : 广播内容, 为 hex 字符串, 例如 AT+BLEADV DATA="020106020941"代表将广播数据设置为" 0x02 0x01 0x06 0x02 0x09 0x41"	执行结果
示例 1: AT+BLEADVDATA="020106020941" 正确响应 1: OK		

6.7. AT+BLEADVDATAEX

表 6-7. 设置广播内容

指令	参数	响应
帮助指令 AT+BLEADVDATA EX=?		+BLEADVDATA EX =<dev_name>,<uuid>,<manufacturer_data>,<include_power>
执行指令 AT+BLEADVDATA EX =<dev_name>,<uuid>,<manufacturer_data>,<include_power>	<dev_name>: 设备名称 <uuid>: service uuid <manufacturer_data>: 厂家数据 <include_power>: 是否包含 power	执行结果
示例 1: AT+BLEADVDATA EX="test","a002","2b0c112233",1 正确响应 1: OK		

6.8. AT+BLESCANRSPDATA

表 6-8. 设置扫描回复内容

指令	参数	响应
帮助指令 AT+BLESCANRSPDATA=?		+BLESCANRSPDATA=<data>
执行指令 AT+BLESCANRSPDATA=<data>	<data> : 广播内容, 为 hex 字符串, 例如 AT+BLESCANRSPDATA = "020941"代表将广播数据设置为"0x02 0x09 0x41"	执行结果
示例 1: AT+BLESCANRSPDATA="020941" 正确响应 1: OK		

6.9. AT+BLEPASSTH

表 6-9. 开启透传模式

指令	参数	响应
执行指令 AT+BLEPASSTH		执行结果
示例 1: 先开启广播 AT+BLEADVSTART=0, 对端与之建立连线 输入 AT+BLEPASSTH 开启透传模式 “+++”退出 无响应		

6.10. AT+BLEPASSTHAUTO

表 6-10. 自动开启透传模式

指令	参数	响应
帮助指令 AT+BLEPASSTHAUTO=?		+BLEPASSTHAUTO=<enable>
执行指令 AT+BLEPASSTHAUTO=<enable>	<enable> : 是否开启自动进入透	执行结果

指令	参数	响应
enable>	透传模式	
注:主从都通过同一命令自动进入透传模式 示例 1: 输入 AT+BLEPASSTHAUTO=1 先开启广播 AT+BLEADVSTART=0, 或者 AT+BLECONN=0, AB:89:67:45:23:01 (对端地址), 建立连接后自动开启透传模式 “+++”退出 正确响应 1: OK		

6.11. AT+BLEPASSTHCLI

表 6-11. 开启透传模式

指令	参数	响应
执行指令 AT+BLEPASSTHCLI		执行结果
示例 1: AT+BLECONN=0, AB:89:67:45:23:01 (对端地址), 对端与之建立连线 输入 AT+BLEPASSTHCLI 开启透传模式 “+++”退出 无响应		

6.12. AT+BLESCANPARAM

表 6-12. 设置扫描参数

指令	参数	响应
帮助指令 AT+BLESCANPARAM=?		+BLESCANPARAM=<type>,<own_addr_type>,<dup_filt_pol>,<scan_intv_1m>,<scan_win_1m>
查询指令 AT+BLESCANPARAM?		+BLESCANPARAM:<type>,<own_addr_type>,<dup_filt_pol>,<scan_intv_1m>,<scan_win_1m>
执行指令 AT+BLESCANPARAM:<type>,<own_addr_type>,<dup_filt_pol>,<scan_intv_1m>,<scan_win_1m>	<type> : 扫描类型 <own_addr_type> : 本地地址类型 <dup_filt_pol> : 重复包过滤政策 <scan_intv_1m> : 1M 扫描间隔 <scan_win_1m> : 1M 扫描窗口大	执行结果

指令	参数	响应
	小	
示例 1: AT+BLESKANPARAM? 正确响应 1: +BLESKANPARAM:0,0,1,160,32 OK 示例 2: AT+BLESKANPARAM=0,0,1,160,48 正确响应 2: OK		

6.13. AT+BLESKAN

表 6-13. 开启扫描

指令	参数	响应
帮助指令 AT+BLESKAN=?		+BLESKAN=<enable>
执行指令 AT+BLESKAN=<enable>	<enable> : 是否开启扫描	执行结果
示例 1: AT+BLESKAN=1 正确响应 1: OK		

6.14. AT+BLESYNC

表 6-14. BLE 开始/取消同步

指令	参数	响应
帮助指令 AT+BLESYNC=?		+BLESYNC=<enable>,<addr_type>,<addr>
执行指令 AT+BLESYNC=<enable>,<addr_type>,<addr>	<enable> : 是否开启 sync <addr_type> : 本地地址类型 <addr> : 对端地址	执行结果
示例 1: 开始同步 AT+BLESYNC=1,0,AB:89:67:45:23:01		

指令	参数	响应
正确响应 1: OK 示例 2: 取消同步 AT+BLESYNC=0 正确响应 2: OK		

6.15. AT+BLESYNCSTOP

表 6-15. BLE 停止同步

指令	参数	响应
执行指令 AT+BLESYNCSTOP		执行结果
示例 1: AT+BLESYNCSTOP 正确响应 1: OK		

6.16. AT+BLECONN

表 6-16. BLE 建立连接

指令	参数	响应
帮助指令 AT+BLECONN=?		+BLECONN=<addr_type>,<addr>
执行指令 AT+BLECONN=<type>,<addr>	<addr_type> : 本地地址类型 <addr> : 本地地址	执行结果
示例 1: 对端先开启广播 AT+BLECONN=0, AB:89:67:45:23:01 (对端地址) 正确响应 1: OK		

6.17. AT+BLECONNPARAM

表 6-17. 设置/查询连接参数

指令	参数	响应
帮助指令 AT+BLECONNPARAM=?		+BLECONNPARAM=<conn_idx>,<interval>,<latency>,<supv_to>
查询指令 AT+BLECONNPARAM?		+BLECONNPARAM:<conn_idx>,<interval>,<latency>,<supv_to>
执行指令 AT+BLECONNPARAM=<conn_idx>,<interval>,<latency>,<supv_to>	<conn_idx> : 连接索引 < interval > : 连接间隔 < latency > : 允许 slave 不交互的 interval 数 < supv_to > : 超时时间	执行结果
<p>示例 1: 查询连接参数</p> <p>先建立连接 AT+BLECONN=0,<addr> 或者先开启广播 AT+BLEADVSTART=0, 对端建立连线 输入 AT+BLECONNPARAM?</p> <p>正确响应 1:</p> <p>+BLECONNPARAM:0,40,0,500</p> <p>OK</p> <p>示例 2: 修改连接参数</p> <p>先建立连接 AT+BLECONN=0,<addr>或者先开启广播 AT+BLEADVSTART=0, 对端建立连线 输入 AT+BLECONNPARAM=0,50,20,500</p> <p>正确响应 2:</p> <p>OK</p>		

6.18. AT+BLEDISCONN

表 6-18. BLE 断开连接

指令	参数	响应
帮助指令 AT+BLEDISCONN=?		+BLEDISCONN=<conn_idx>
执行指令 AT+BLEDISCONN=<conn_idx>	<conn_idx> : 连接索引	执行结果
<p>示例 1:</p> <p>先建立连接 AT+BLECONN=0,<addr>或者先开启广播 AT+BLEADVSTART=0, 对端建立连线</p>		

指令	参数	响应
输入 AT+BLEDISCONN=0 正确响应 1: OK		

6.19. AT+BLEMTU

表 6-19. 更新/查询 mtu

指令	参数	响应
帮助指令 AT+BLEMTU=?		+BLEMTU=<conn_idx>,<pref_mtu>
查询指令 AT+BLEMTU?		+BLEMTU:<conn_idx>,<mtu_size>
执行指令 AT+BLEMTU=<conn_idx>,<pref_mtu>	<conn_idx>: 连接索引 <pref_mtu>:期望 mtu	执行结果
示例 1: 查询 mtu 先建立连接 AT+BLECONN=0,<addr>或者先开启广播 AT+BLEADVSTART=0, 对端建立连线 输入 AT+BLEMTU? 正确响应 1: +BLEMTU:0,23 OK 示例 2: 更新 MTU 先建立连接 AT+BLECONN=0,<addr> 输入 AT+BLEMTU=0,1000 正确响应 2: OK		

6.20. AT+BLEPHY

表 6-20. 更新/查询 phy

指令	参数	响应
帮助指令 AT+BLEPHY=?		+BLEPHY =<conn_idx>,<tx_phy>,<rx_phy>,<phy_opt>

指令	参数	响应
查询指令 AT+BLEPHY?		+BLEPHY: <conn_idx>, <tx_phy>, <rx_phy>
执行指令 AT+BLEPHY=<conn_idx>,<tx_phy>,<rx_phy>,<phy_opt>	<conn_idx>: 连接索引 <tx_phy>: tx phy <rx_phy>: rx phy <phy_opt>: coded phy option	执行结果
<p>示例 1: 查询 phy</p> <p>先建立连接 AT+BLECONN=0,<addr>或者先开启广播 AT+BLEADVSTART=0, 对端建立连线 输入 AT+BLEPHY?</p> <p>正确响应 1: +BLEPHY:0,0,0 OK</p> <p>示例 2: 更新 phy</p> <p>先建立连接 AT+BLECONN=0,<addr>或者先开启广播 AT+BLEADVSTART=0, 对端建立连线 输入 AT+BLEPHY=0,1,1,0</p> <p>正确响应 2: OK</p>		

6.21. AT+BLEDATALEN

表 6-21. Data length extension

指令	参数	响应
帮助指令 AT+BLEDATALEN=?		+BLEDATALEN=<conn_idx>,<tx_oct>
执行指令 AT+BLEDATALEN=<conn_idx>,<tx_oct>	<conn_idx> : 连接索引 <tx_oct> : 期望的 payload 长度	执行结果
<p>示例 1:</p> <p>先建立连接 AT+BLECONN=0,<addr>或者先开启广播 AT+BLEADVSTART=0, 对端建立连线 输入 AT+BLEDATALEN=0,200</p> <p>正确响应 1: OK</p>		

6.22. AT+BLEADDR

表 6-22. 查询/设置 ble bd address

指令	参数	响应
帮助指令 AT+BLEADDR=?		+BLEADDR=<bd_addr>
查询指令 AT+BLEADDR?		+BLEADDR:<bd_addr>
执行指令 AT+BLEADDR=<bd_addr>	< bd_addr > : ble bd address	执行结果
<p>示例 1: 查询 ble bd address</p> <p>AT+BLEADDR?</p> <p>正确响应 1:</p> <p>+BLEBDADDR:77:66:55:44:33:22</p> <p>OK</p> <p>示例 2: 设置 ble bd address</p> <p>AT+BLEADDR=22:33:44:55:66:77</p> <p>正确响应 2:</p> <p>OK</p>		

6.23. AT+BLESETAUTH

表 6-23. 配置 AUTHENTICATION

指令	参数	响应
帮助指令 AT+BLESETAUTH=?		+BLESETAUTH=<bond>, <mitm>, <sc>, <iocap>, <oob>, <key_size>

指令	参数	响应
执行指令 AT+BLESETAUTH=<bond> ,<mitm>,<sc>,<iocap>,<oob> ,<key_size>	<bond> : bonding flag 0x00: no bonding 0x01: bonding <mitm>: mitm flag 0x00: mitm protection not required 0x01: mitm protection required <sc>: secure connections flag 0x00: secure connections pairing is not supported 0x01: secure connections pairing is supported <iocap>: io capability to set 0x00: display only 0x01: display yes no 0x02: keyboard only 0x03: no input no output 0x04: keyboard display <oob>: oob flag for authentication [key size]: encryption key size requirement, default is 16 if not set	执行结果
示例 1: 配置 AUTHENTICATION AT+BLESETAUTH=1,0,0,3,0,16 正确响应 1: OK		

6.24. AT+BLEPAIR

表 6-24. 发起配对

指令	参数	响应
帮助指令 AT+BLEPAIR=?		+BLEPAIR=<conidx>
执行指令 AT+BLEPAIR=<conidx>	<conidx>: 连接索引	执行结果
示例 1: 发起配对 先建立连接 AT+BLECONN=0,<addr>或者先开启广播 AT+BLEADVSTART=0, 对端建立连线 输入 AT+BLEPAIR=0 正确响应 1:		

指令	参数	响应
OK		

6.25. AT+BLEENCRYPT

表 6-25. 启动加密

指令	参数	响应
帮助指令 AT+BLEENCRYPT=?		+BLEENCRYPT=<conidx >
执行指令 AT+BLEENCRYPT=<conidx >	< conidx > : 连接索引	执行结果
<p>示例 1: 启动加密 (需要之前配对过的设备)</p> <p>先建立连接 AT+BLECONN=0,<addr>或者先开启广播 AT+BLEADVSTART=0, 对端建立连线</p> <p>输入 AT+BLEENCRYPT=0</p> <p>正确响应 1:</p> <p>OK</p>		

6.26. AT+BLEPASSKEY

表 6-26. 输入 passkey

指令	参数	响应
帮助指令 AT+BLEPASSKEY=?		+BLEPASSKEY=<conidx>,<passkey>
执行指令 AT+BLEPASSKEY=<conidx>,<passkey>	< conidx > : 连接索引 <passkey>: passkey	执行结果
<p>示例 1: 输入 passkey</p> <p>输入 AT+BLESETAUTH=1,1,0,2,0,16</p> <p>先建立连接 AT+BLECONN=0,<addr>或者先开启广播 AT+BLEADVSTART=0, 对端建立连线</p> <p>输入 AT+BLEPASSKEY=0,123456 (输入显示的值)</p> <p>正确响应 1:</p> <p>OK</p>		

指令	参数	响应

6.27. AT+BLECOMPARE

表 6-27. 输入 compare 结果

指令	参数	响应
帮助指令 AT+BLECOMPARE=?		+BLECOMPARE=<conidx>,<value>
执行指令 AT+BLECOMPARE=<conidx>,<value>	< conidx > : 连接索引 <value>: 比较结果	执行结果
示例 1: 输入 compare 结果 输入 AT+BLESETAUTH=1,1,1,4,0,16 先建立连接 AT+BLECONN=0,<addr>或者先开启广播 AT+BLEADVSTART=0, 对端建立连线 AT+BLECOMPARE=0,1 正确响应 1: OK		

6.28. AT+BLELISTENCDEV

表 6-28. 列出 bond device 列表

指令	参数	响应
查询指令 AT+BLELISTENCDEV?		+BLEADDR:<dev_idx><addr>
示例 1: 列出 bond device 列表 需要先有设备配对过 AT+BLELISTENCDEV? 正确响应 1: +BLELISTENCDEV=0,AB:89:67:45:23:01 +BLELISTENCDEV=1,D0:20:DD:EE:5C:3C OK		

6.29. AT+BLECLEARENCDEV

表 6-29. 移除 bond 设备

指令	参数	响应
帮助指令 AT+BLECLEARENCDEV =?		+BLECLEARENCDEV=<dev_idx>
执行指令 AT+BLECLEARENCDEV=< dev_idx>	< dev_idx > : 设备索引	执行结果
示例 1: 输入移除 bond 设备 需要先有设备配对过 AT+BLECLEARENCDEV=0 正确响应 1: OK		

6.30. AT+BLEGATTSSVC

表 6-30. 列出本地注册的 service

指令	参数	响应
查询指令 AT+BLEGATTSSVC?		+BLEGATTSSVC:<svc_id><uuid>
示例 1: 列出本地注册的 service AT+BLEGATTSSVC? 正确响应 1: +BLEGATTSSVC:0,00000000000000000000000000000000180A,1 +BLEGATTSSVC:1,000011110000000000123456789ABCDEF,1 +BLEGATTSSVC:2,00000000000000000000000000000000101,1 OK		

6.31. AT+BLEGATTSCHAR

表 6-31. 列出对应 service 的 characteristic

指令	参数	响应
帮助指令 AT+BLEGATTSCHAR=?		+BLEGATTSCHAR=<svc_idx>
执行指令 AT+BLEGATTSCHAR=<svc_idx>	< svc_idx > : service 索引	执行结果 +BLEGATTSCHAR:<uuid ><value_index>
示例 1: 列出本地注册的 characteristic AT+BLEGATTSCHAR=1 正确响应 1: +BLEGATTSCHAR:00002222000000000123456789ABCDEF,2 +BLEGATTSCHAR:00003333000000000123456789ABCDEF,4 +BLEGATTSCHAR:00004444000000000123456789ABCDEF,6 OK		

6.32. AT+BLEGATTSDESC

表 6-32. 列出对应 characteristic 的 descriptor

指令	参数	响应
帮助指令 AT+BLEGATTSDESC=?		+BLEGATTSDESC=<svc_idx>,<char_idx>
执行指令 AT+BLEGATTSDESC=<svc_idx>,<char_idx>	< svc_idx > : service 索引 <char_idx>: characteristic 索引	执行结果 +BLEGATTSDESC:<uuid ><desc_idx>
示例 1: 列出对应 characteristic 的 descriptor AT+BLEGATTSDESC=1,6 正确响应 1: +BLEGATTSDESC:0000000000000000000000000002902,7 OK		

6.33. AT+BLEGATTSLISTALL

表 6-33. 列出本地所有 service 中信息

指令	参数	响应
查询指令 AT+BLEGATTSLISTALL ?		+BLEGATTSSVC:<svc_id><uuid> +BLEGATTCHAR:<uuid ><value_index> +BLEGATTSDESC:<uuid ><desc_idx>
示例 1: 列出本地所有 service 中信息 AT+BLEGATTSLISTALL? 正确响应 1: +BLEGATTCHAR:000000000000000000000000000002A29,2 +BLEGATTCHAR:000000000000000000000000000002A24,4 +BLEGATTCHAR:000000000000000000000000000002A25,6 +BLEGATTCHAR:000000000000000000000000000002A27,8 +BLEGATTCHAR:000000000000000000000000000002A26,10 +BLEGATTCHAR:000000000000000000000000000002A28,12 +BLEGATTCHAR:000000000000000000000000000002A23,14 +BLEGATTCHAR:000000000000000000000000000002A2A,16 +BLEGATTCHAR:000000000000000000000000000002A50,18 +BLEGATTSSVC:1,00001111000000000123456789ABCDEF,1 +BLEGATTCHAR:00002222000000000123456789ABCDEF,2 +BLEGATTCHAR:00003333000000000123456789ABCDEF,4 +BLEGATTCHAR:00004444000000000123456789ABCDEF,6 +BLEGATTSDESC:000000000000000000000000000002902,7 +BLEGATTSSVC:2,00000000000000000000000000000101,1 +BLEGATTCHAR:00000000000000000000000000000102,2 +BLEGATTCHAR:00000000000000000000000000000103,4 +BLEGATTSDESC:000000000000000000000000000002902,5 OK		

6.34. AT+BLEGATTSNTF

表 6-34. 发送 notification

指令	参数	响应
帮助指令 AT+BLEGATTSNTF=?		+BLEGATTSNTF=<conn_idx>,<svc_id>,<char_idx>,<tx_len>
执行指令 AT+BLEGATTSNTF=<conn_idx>,<svc_id>,<char_idx>,<tx_len>	<conn_idx> : 连接索引 <svc_id>: service id <char_idx>: characteristic 索引 <tx_len>: 数据长度	执行结果
示例 1: 发送 notification		

指令	参数	响应
先开启广播 AT+BLEADVSTART=0，对端建立连线 输入 AT+BLEGATTSNTF=0,1,6,5 > 输入 AAAAA(对端会收到数据) 正确响应 1: OK		

6.35. AT+BLEGATTSIND

表 6-35. 发送 indication

指令	参数	响应
帮助指令 AT+BLEGATTSIND=?		+BLEGATTSIND=<conn_idx>,<svc_id>,<char_idx>,<tx_len>
执行指令 AT+BLEGATTSIND=<conn_idx>,<svc_id>,<char_idx>,<tx_len>	<conn_idx> : 连接索引 <svc_id>: service id <char_idx>: characteristic 索引 <tx_len>: 数据长度	执行结果
示例 1: 发送 indication 先开启广播 AT+BLEADVSTART=0，对端建立连线 输入 AT+BLEGATTSIND=0,1,6,5 > 输入 AAAAA(对端会收到数据) 正确响应 1: OK		

6.36. AT+BLEGATTSSETATTRVAL

表 6-36. 设置 characteristic 的值

指令	参数	响应
帮助指令 AT+BLEGATTSSETATTRVAL=?		+BLEGATTSSETATTRVAL=<conn_idx>,<svc_id>,<char_idx>,<tx_len>
执行指令 AT+BLEGATTSSETATTRVAL=<conn_idx>,<svc_id>,<char_idx>,<tx_len>	<conn_idx> : 连接索引 <svc_id>: service id <char_idx>: characteristic 索引 <tx_len>: 数据长度	执行结果

指令	参数	响应
示例 1: 设置 characteristic 的值 先开启广播 AT+BLEADVSTART=0, 对端建立连线 输入 AT+BLEGATTSSSETATTRVAL=0,1,4,5 > 输入 AAAAA(本地数据改变) 正确响应 1: OK		

6.37. AT+BLEGATTCDISCSVC

表 6-37. 发现 service

指令	参数	响应
帮助指令 AT+BLEGATTCDISCSVC C=?		+BLEGATTCDISCSVC=<conn_idx>,<start_hdl>,<end_hdl>
执行指令 AT+BLEGATTCDISCSVC= <conn_idx>,<start_hdl>,<e nd_hdl>	<conn_idx> : 连接索引 < start_hdl >: start attribute handle < end_hdl >: end attribute handle	执行结果 +BLEGATTCDISCSVC:<start_hdl >,<end_hdl >,<uuid>
示例 1: 发现 service 先建立连接 AT+BLECONN=0,<addr> 输入 AT+BLEGATTCDISCSVC=0,1,ffff 正确响应 1: +BLEGATTCDISCSVC:1,8,00001111000000000123456789ABCDEF +BLEGATTCDISCSVC:9,14,00000000000000000000000000000101 +BLEGATTCDISCSVC:16,25,000000000000000000000000000001801 +BLEGATTCDISCSVC:32,40,000000000000000000000000000001800 +BLEGATTCDISCSVC:43,61,00000000000000000000000000000180A OK		

6.38. AT+BLEGATTCDISCCHAR

表 6-38. 发现 characteristic

指令	参数	响应
帮助指令 AT+BLEGATTCDISCCH AR=?		+BLEGATTCDISCCHAR=<conn_ idx>,<start_hdl>,<end_hdl>
执行指令 AT+BLEGATTCDISCCH AR=<conn_idx>,<start_hdl >,<end_hdl>	<conn_idx> : 连接索引 < start_hdl >: start attribute handle < end_hdl >: end attribute handle	执行结果 +BLEGATTCDISCCHAR:<char_h dl >,<val_hdl >,<prop >,<uuid>
<p>示例 1: 发现 characteristic</p> <p>先建立连接 AT+BLECONN=0,<addr></p> <p>输入 AT+BLEGATTCDISCCHAR=0,1,ffff</p> <p>正确响应 1:</p> <pre>+BLEGATTCDISCCHAR:2,3,2,0000222200000000123456789ABCDEF +BLEGATTCDISCCHAR:4,5,12,0000333300000000123456789ABCDEF +BLEGATTCDISCCHAR:6,7,16,0000444400000000123456789ABCDEF +BLEGATTCDISCCHAR:10,11,12,0000000000000000000000000102 +BLEGATTCDISCCHAR:12,13,16,000000000000000000000000103 +BLEGATTCDISCCHAR:17,18,32,0000000000000000000000002A05 +BLEGATTCDISCCHAR:20,21,10,000000000000000000000002B29 +BLEGATTCDISCCHAR:22,23,2,0000000000000000000000002B2A +BLEGATTCDISCCHAR:24,25,2,0000000000000000000000002B3A +BLEGATTCDISCCHAR:33,34,10,000000000000000000000002A00 +BLEGATTCDISCCHAR:35,36,10,000000000000000000000002A01 +BLEGATTCDISCCHAR:37,38,2,000000000000000000000002A04 +BLEGATTCDISCCHAR:39,40,2,000000000000000000000002AA6 +BLEGATTCDISCCHAR:44,45,2,000000000000000000000002A29 +BLEGATTCDISCCHAR:46,47,2,000000000000000000000002A24 OK</pre>		

6.39. AT+BLEGATTCDISCDESC

表 6-39. 发现 descriptor

指令	参数	响应
帮助指令 AT+BLEGATTCDISCDE SC=?		+BLEGATTCDISCDESC=<conn_i dx>,<start_hdl>,<end_hdl>
执行指令 AT+BLEGATTCDISCDE SC=<conn_idx>,<start_hdl >,<end_hdl>	<conn_idx> : 连接索引 < start_hdl >: start attribute handle	执行结果 +BLEGATTCDISCDESC:<desc_h dl >,<uuid>

指令	参数	响应
	< end_hdl >: end attribute handle	
示例 1: 发现 descriptor 先建立连接 AT+BLECONN=0,<addr> 输入 AT+BLEGATTCDISCDESC=0,1,ffff 正确响应 1: +BLEGATTCDISCDESC: 8,0000000000000000000000000000002902 +BLEGATTCDISCDESC: 14,0000000000000000000000000000002902 +BLEGATTCDISCDESC: 19,0000000000000000000000000000002902 OK		

6.40. AT+BLEGATTCRD

表 6-40. Read attribute value

指令	参数	响应
帮助指令 AT+BLEGATTCRD=?		+BLEGATTCRD=<conn_idx>,<handle>,<max_len>
执行指令 AT+BLEGATTCRD=<conn_idx>,<handle>,<max_len>	<conn_idx> : 连接索引 < handle >: attribute handle < max_len >: max length	执行结果 +BLEGATTCRD:<conn_idx>,<length>,<data>
示例 1: Read attribute value 先建立连接 AT+BLECONN=0,<addr> 输入 AT+BLEGATTCRD=0,3,100 正确响应 1: +BLEGATTCRD:0,2,2222 OK		

6.41. AT+BLEGATTCWR

表 6-41. Write attribute value

指令	参数	响应
帮助指令 AT+BLEGATTCWR=?		+BLEGATTCWR=<conn_idx>,<handle>,<write_type>,<len>
执行指令 AT+BLEGATTCWR=<conn_idx>,<handle>,<write_type>,<len>	<conn_idx> : 连接索引 < handle >: attribute handle < write_type >: 写类型	执行结果

指令	参数	响应
	<len>: 写长度	
<p>示例 1: Write attribute value 先建立连接 AT+BLECONN=0,<addr> 输入 AT+BLEGATTCWR=0,5,0,5 > 输入 AAAAA(对端会收到数据) 正确响应 1: OK</p>		

7. 版本历史

表 7-1. 版本历史

版本号.	说明	日期
1.0	首次发布	2023 年 10 月 17 日
1.1	新增了 AT+TRANSINTVL 和 AT+CIPMODE 命令，并扩展了原有的 AT+CIPSEND 支持数据透传	2024 年 7 月 16 日
1.2	新增 BLE 相关 AT 命令	2024 年 10 月 8 日
1.3	新增 BLE 相关 AT 命令	2025 年 3 月 20 日

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the "Product"), is owned by the Company according to the laws of the People's Republic of China and other applicable laws. The Company reserves all rights under such laws and no Intellectual Property Rights are transferred (either wholly or partially) or licensed by the Company (either expressly or impliedly) herein. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

To the maximum extent permitted by applicable law, the Company makes no representations or warranties of any kind, express or implied, with regard to the merchantability and the fitness for a particular purpose of the Product, nor does the Company assume any liability arising out of the application or use of any Product. Any information provided in this document is provided only for reference purposes. It is the sole responsibility of the user of this document to determine whether the Product is suitable and fit for its applications and products planned, and properly design, program, and test the functionality and safety of its applications and products planned using the Product. The Product is designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only, and the Product is not designed or intended for use in (i) safety critical applications such as weapons systems, nuclear facilities, atomic energy controller, combustion controller, aeronautic or aerospace applications, traffic signal instruments, pollution control or hazardous substance management; (ii) life-support systems, other medical equipment or systems (including life support equipment and surgical implants); (iii) automotive applications or environments, including but not limited to applications for active and passive safety of automobiles (regardless of front market or aftermarket), for example, EPS, braking, ADAS (camera/fusion), EMS, TCU, BMS, BSG, TPMS, Airbag, Suspension, DMS, ICMS, Domain, ESC, DCDC, e-clutch, advanced-lighting, etc.. Automobile herein means a vehicle propelled by a self-contained motor, engine or the like, such as, without limitation, cars, trucks, motorcycles, electric cars, and other transportation devices; and/or (iv) other uses where the failure of the device or the Product can reasonably be expected to result in personal injury, death, or severe property or environmental damage (collectively "Unintended Uses"). Customers shall take any and all actions to ensure the Product meets the applicable laws and regulations. The Company is not liable for, in whole or in part, and customers shall hereby release the Company as well as its suppliers and/or distributors from, any claim, damage, or other liability arising from or related to all Unintended Uses of the Product. Customers shall indemnify and hold the Company, and its officers, employees, subsidiaries, affiliates as well as its suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Product.

Information in this document is provided solely in connection with the Product. The Company reserves the right to make changes, corrections, modifications or improvements to this document and the Product described herein at any time without notice. The Company shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. Information in this document supersedes and replaces information previously supplied in any prior versions of this document.