

HH-D03 AT 命令

# 使用指南



# 前言

## 概述

本文介绍 HH-D03 的 AT 指令式及场景，为用户提供相应的指令格式和参数示例解释。

## 产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
BS21	V100

## 读者对象






本文档主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

## 符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
----	----

符号	说明
 危险	表示如不避免则将会导致死亡或严重伤害的具有高等级风险的危害。
 警告	表示如不避免则可能导致死亡或严重伤害的具有中等级风险的危害。
 注意	表示如不避免则可能导致轻微或中度伤害的具有低等级风险的危害。
 须知	用于传递设备或环境安全警示信息。如不避免则可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 “须知”不涉及人身伤害。
 说明	对正文中重点信息的补充说明。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

# 目 录

前言.....	i
<b>1 指令说明 .....</b>	<b>1</b>
1.1 命令简介.....	1
1.2 指令类型.....	1
1.3 注意事项.....	2
<b>2 通用 AT 指令 .....</b>	<b>4</b>
2.1 通用 AT 指令一览表 .....	4
2.2 通用 AT 指令描述 .....	4
2.2.1 AT 测试 AT 功能 .....	4
2.2.2 AT+HELP 查看当前可用 AT 命令 .....	5
<b>3 BT 业务 AT 指令 .....</b>	<b>6</b>
3.1 BT 业务 AT 指令一览表 .....	6
3.1.1 BLE .....	6
3.1.1.1 gap 模块 AT 命令 .....	6
3.1.1.2 gatts 模块 AT 命令 .....	7
3.1.1.3 gattc 模块 AT 命令 .....	8
3.1.2 SLE .....	9
3.2 BT 业务 AT 指令描述 .....	11
3.2.1 BLE .....	11
3.2.1.1 gap 模块 AT 命令 .....	11
3.2.1.1.1 AT+BLEENABLE 使能 ble 协议栈 .....	11
3.2.1.1.2 AT+BLEDISABLE 关闭 ble 协议栈 .....	11
3.2.1.1.3 AT+BLESETADDR 设置本地设备地址 .....	11
3.2.1.1.4 AT+BLEGETADDR 获取本地设备地址 .....	12

3.2.1.1.5 AT+BLESETNAME 设置本地设备名称 .....	12
3.2.1.1.6 AT+BLEGETNAME 获取本地设备名称.....	12
3.2.1.1.7 AT+BLESETAPPEARANCE 设置本地设备外观 .....	13
3.2.1.1.8 AT+BLESETADVDATA 设置BLE 广播数据.....	13
3.2.1.1.9 AT+BLESETADVPAR 设置广播数据参数 .....	14
3.2.1.1.10 AT+BLESTARTADV 开始发送BLE 广播.....	14
3.2.1.1.11 AT+BLESTOPADV 停止发送 BLE 广播.....	15
3.2.1.1.12 AT+BLESETSCANPAR 设置 BLE 扫描参数 .....	15
3.2.1.1.13 AT+BLESTARTSCAN 启动 BLE 扫描 .....	15
3.2.1.1.14 AT+BLESTOPSCAN 停止 BLE 扫描 .....	16
3.2.1.1.15 AT+BLEPAIR 与对端设备发起配对.....	16
3.2.1.1.16 AT+BLEGETPAIREDNUM 获取 BLE 设备配对设备数量.....	17
3.2.1.1.17 AT+BLEGETPAIREDDEV 获取 BLE 设备配对设备 .....	17
3.2.1.1.18 AT+BLEGETPAIREDSTA 获取BLE 设备配对状态 .....	17
3.2.1.1.19 AT+BLEUNPAIR 取消配对.....	18
3.2.1.1.20 AT+BLEUNPAIR 取消所有配对.....	18
3.2.1.1.21 AT+BLECONNPARUPD 更新连接参数.....	18
3.2.1.1.22 AT+BLECONN 与 BLE 设备连接.....	19
3.2.1.1.23 AT+BLEDISCONN 与 BLE 设备断开连接.....	19
3.2.1.1.24 AT+BLEGAPREGCBK 注册 BLE 回调函数.....	20
3.2.1.2 gatts 模块 AT 命令.....	20
3.2.1.2.1 AT+GATTSREGSRV 创建一个GATT server.....	20
3.2.1.2.2 AT+GATTSUNREG 删除 GATT server, 释放资源.....	20
3.2.1.2.3 AT+GATTSADDSERV 添加一个GATT 服务 .....	21
3.2.1.2.4 AT+GATTSSYNCADDSERV 添加一个GATT 服务 (同步) .....	21
3.2.1.2.5 AT+GATTSADDCHAR 为 GATT 服务添加一个特征.....	21
3.2.1.2.6 AT+GATTSSYNCADDCHAR 为 GATT 服务添加一个特征 (同步) .....	22
3.2.1.2.7 AT+GATTSADDDESCR 为最新的特征添加一个描述符 .....	22
3.2.1.2.8 AT+GATTSSYNCADDDESCR 为最新的特征添加一个描述符 (同步) .....	23
3.2.1.2.9 AT+GATTSSTARTSERV 启动指定的GATT 服务.....	23
3.2.1.2.10 AT+GATTSDELALLSERV 删除指定 server 上的所有服务.....	24

3.2.1.2.11 AT+GATTSSENDRSP 发送响应 .....	24
3.2.1.2.12 AT+GATTSSNDNTFY 发送通知或指示 .....	25
3.2.1.2.13 AT+GATTSSNDNTFYBYUUID 根据 uuid 发送通知或指示 .....	25
3.2.1.2.14 AT+GATTSSREGCBK 注册 GATT 服务端回调函数 .....	26
3.2.1.2.15 AT+GATTSSSETMTU 在连接之前设置 server rx mtu .....	26
3.2.1.3 gattc 模块 AT 命令 .....	26
3.2.1.3.1 AT+GATTCREG 创建一个 GATT client .....	26
3.2.1.3.2 AT+GATTCUNREG 删除 GATT client, 释放资源 .....	27
3.2.1.3.3 AT+GATTCFNDSEV 发现服务 .....	27
3.2.1.3.4 AT+GATTCFNDCHAR 发现特征 .....	27
3.2.1.3.5 AT+GATTCFNDDESCR 发现描述符 .....	28
3.2.1.3.6 AT+GATTCREADBYHDL 读取 by hdl .....	28
3.2.1.3.7 AT+GATTCREADBYUUID 读取 by uuid .....	29
3.2.1.3.8 AT+GATTCWRITEREQ 写 by hdl req .....	29
3.2.1.3.9 AT+GATTCWRITECMD 写 by hdl cmd .....	30
3.2.1.3.10 AT+GATTCEXCHMTU 交换 MTU 请求 .....	30
3.2.1.3.11 AT+GATTCREGCBK 注册 GATT 客户端回调函数 .....	30
3.2.2 SLE .....	31
3.2.2.1 SLE Device Discovery 模块指令描述 .....	31
3.2.2.1.1 SLE 使能 .....	31
3.2.2.1.2 SLE 去使能 .....	31
3.2.2.1.3 设置 SLE 广播参数 .....	31
3.2.2.1.4 设置 SLE 广播数据 .....	32
3.2.2.1.5 起 SLE 广播 .....	33
3.2.2.1.6 停 SLE 广播 .....	33
3.2.2.1.7 设置扫描参数 .....	33
3.2.2.1.8 使能扫描 .....	34
3.2.2.1.9 关闭扫描 .....	34
3.2.2.1.10 设置本端名称 .....	35
3.2.2.1.11 获取本端名称 .....	35
3.2.2.1.12 设置本端地址 .....	35
3.2.2.1.13 获取本端地址 .....	36

3.2.2.2 SLE Connection manager 模块指令描述.....	36
3.2.2.2.1 建立 SLE 连接.....	36
3.2.2.2.2 星闪逻辑链路更新参数.....	37
3.2.2.2.3 星闪读取远端 rssi.....	37
3.2.2.2.4 断开 SLE 连接.....	38
3.2.2.2.5 设置 SLE PHY.....	38
3.2.2.2.6 设置 SLE 默认连接参数.....	39
3.2.2.2.7 SLE 断开所有连接.....	40
3.2.2.3 SLE security managemen 模块指令描述.....	40
3.2.2.3.1 进行加密配对.....	40
3.2.2.3.2 移除加密配对.....	40
3.2.2.3.3 获取配对设备数目.....	41
3.2.2.3.4 获取配对设备.....	41
3.2.2.3.5 获取设备配对状态.....	41
3.2.2.3.6 获取绑定设备.....	42
3.2.2.4 ssap server 模块指令描述.....	42
3.2.2.4.1 注册服务端.....	42
3.2.2.4.2 去注册服务端.....	43
3.2.2.4.3 添加服务.....	43
3.2.2.4.4 添加服务同步.....	43
3.2.2.4.5 添加属性.....	44
3.2.2.4.6 添加属性同步.....	45
3.2.2.4.7 添加属性描述符.....	45
3.2.2.4.8 添加属性描述符同步.....	46
3.2.2.4.9 服务端向客户端发送通知.....	46
3.2.2.4.10 服务端向客户端通过 uuid 发送通知.....	47
3.2.2.4.11 服务端发送响应.....	48
3.2.2.4.12 服务端注册回调.....	49
3.2.2.4.13 start service.....	50
3.2.2.5 ssap client 模块指令描述.....	50
3.2.2.5.1 注册 SSAPC 回调函数.....	50
3.2.2.5.2 发现 service.....	50

3.2.2.5.3 客户端向服务端写入数据.....	51
3.2.2.5.4 客户端向服务端发送写请求.....	52
3.2.2.5.5 客户端发起信息交换.....	52
3.2.2.5.6 客户端通过 uuid 发送读请求.....	53
3.2.2.5.7 客户端读取服务端属性数据.....	53



# 1 指令说明

## 1.1 命令简介

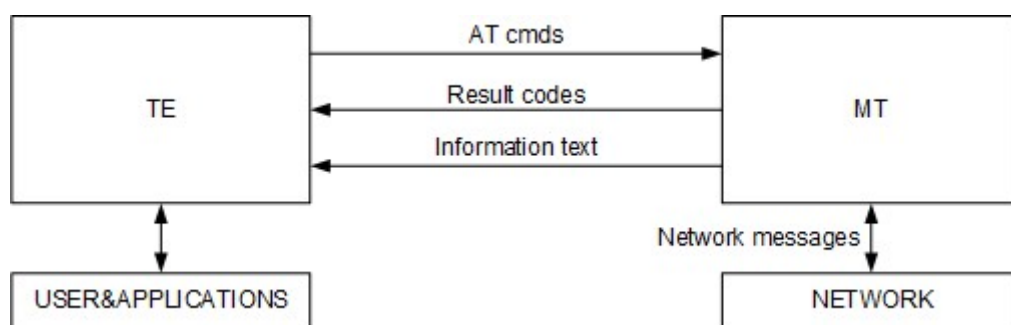
## 1.2 指令类型

## 1.3 注意事项

## 1.1 命令简介

AT 命令用于 TE（例如：PC 等用户终端）和 MT（例如：移动台等移动终端）之间控制信息的交互，如图 1-1 所示。

图1-1 AT 命令示意图



## 1.2 指令类型

AT 指令类型如表 1-1 所示。

表1-1 AT 指令类型说明

类型	格式	用途
测试指令	AT+<cmd>=?	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+<cmd>?	该命令用于返回参数的当前值。
设置指令	AT+<cmd>=<parameter>,...	设置参数值或执行。
执行指令	AT+<cmd>	用于执行本指令的功能。

1.3 注意事项

- 不是每一条指令都具备上述 4 种类型的命令。
- 文档中有但当前软件版本不支持的 AT 指令会返回 ERROR:TBD。
- 双引号表示字符串数据 “string” ， 例如: AT+SCANSSID= “string”。
- 串口通信默认：波特率为 115200、8 个数据位、1 个停止位、无校验。
- []内为可选值，参数可选。
- 命令中的参数以 “,” 作为分隔符，除双引号括起来的字符串参数外，不支持参数本身带 “,” 。
- AT 指令中的参数不能有多余的空格。
- AT 指令必须大写，且必须以回车换行符作为结尾（CR LF），部分串口工具在用户敲击键盘回车键时只有回车符（CR）没有换行符（LF），导致 AT 指令无法识别，如需使用串口工具手动输入 AT 指令，需在串口工具中将回车键设置为回车符（CR）+换行符（LF）。以 IPOP V4.1 和 SecureCRT8.1 为例，说明如图 1-2、图 1-3 所示。

图1-2 IPOP V4.1 CR+LF 设置示例

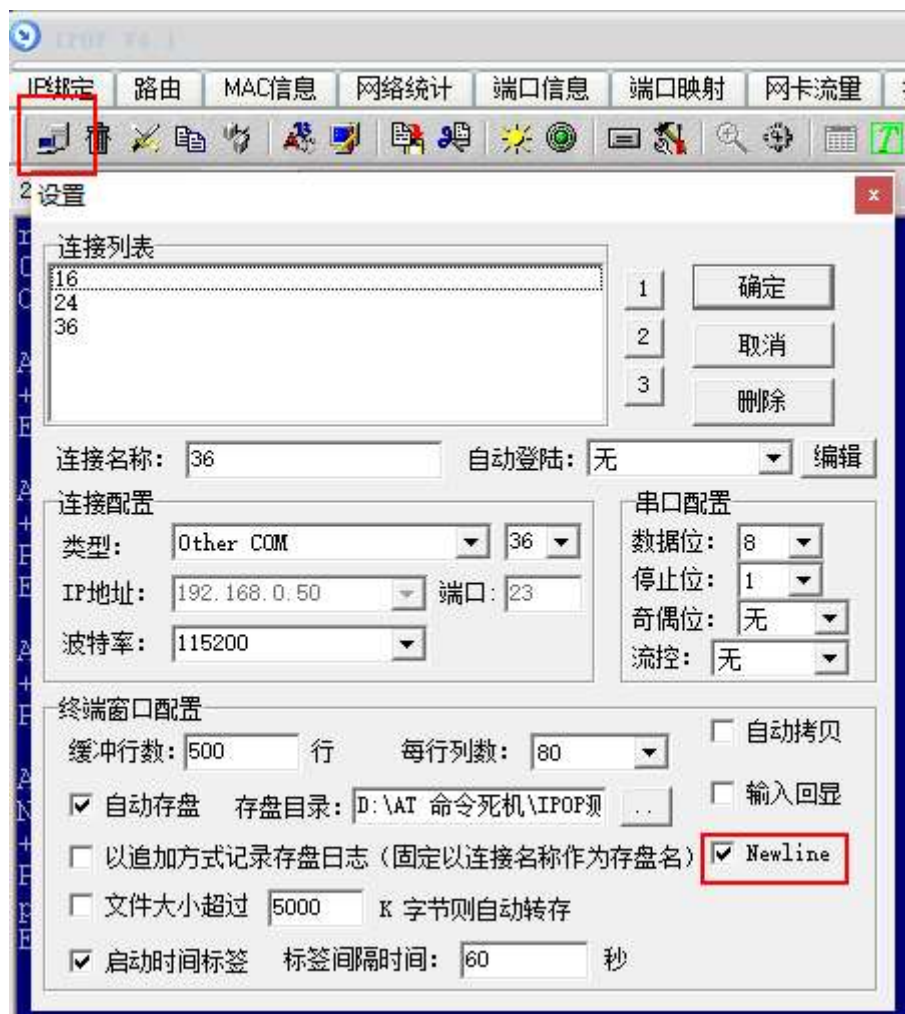
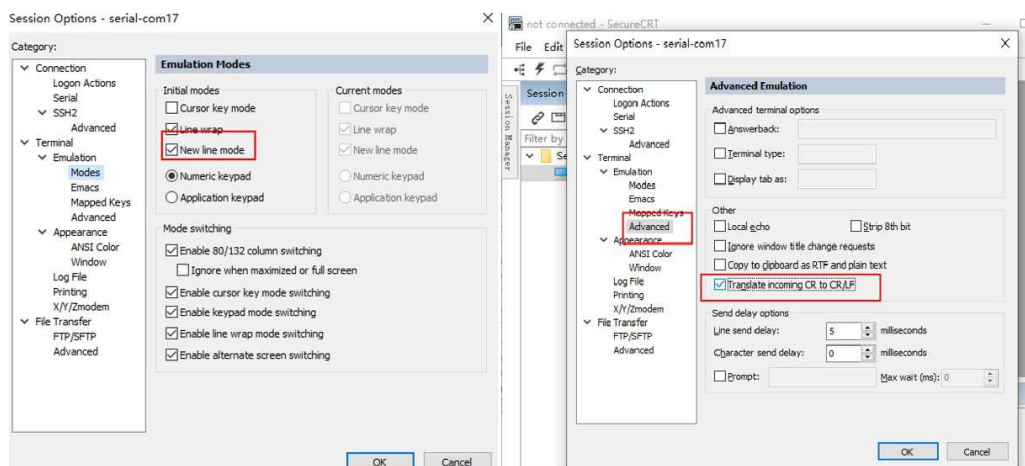


图1-3 SecureCRT8.1 CR+LF 设置示例



# 2 通用 AT 指令

## 2.1 通用 AT 指令一览表

## 2.2 通用 AT 指令描述

## 2.1 通用AT 指令一览表

表2-1 通用 AT 指令说明

指令	描述
AT	测试 AT 功能。
AT+TESTSUITE	AT 调试切换为 TESTSUITE 调试。

## 2.2 通用AT 指令描述

### 2.2.1 AT 测试 AT 功能

表2-2 AT 测试说明

格式	AT
响应	OK
参数说明	-

示例	AT
注意	-

## 2.2.2 AT+HELP 查看当前可用 AT 命令

表2-3 查看可用 AT 命令说明

格式	AT+HELP
响应	+HELP: 显示当前支持的 AT 命令 OK
参数说明	-
示例	AT+HELP
注意	-

# 3 BT 业务 AT 指令

## 3.1 BT 业务 AT 指令一览表

## 3.2 BT 业务 AT 指令描述

### 3.1 BT 业务 AT 指令一览表

#### 3.1.1 BLE

##### 3.1.1.1 gap 模块 AT 命令

指令	描述
AT+BLEENABLE	使能 BLE 协议栈。
AT+BLEDISABLE	关闭 BLE 协议栈。
AT+BLESETADDR=<参数>	设置本地设备地址。
AT+BLEGETADDR	获取本地设备地址。
AT+BLESETNAME=<参数>	设置本地设备名称。
AT+BLEGETNAME	获取本地设备名称。
AT+BLESETAPPEARANCE=<参数>	设置本地设备外观。
AT+BLESETADVDATA=<参数>	设置 BLE 广播数据。
AT+BLESETADVPAR=<参数>	设置 BLE 广播参数。
AT+BLESTARTADV=<参数>	开始发送 BLE 广播。

指令	描述
AT+BLESTOPADV=<参数>	停止发送 BLE 广播。
AT+BLESETSCANPAR=<参数>	设置 BLE 扫描参数。
AT+BLESTARTSCAN	启动 BLE 扫描。
AT+BLESTOPSCAN	停止 BLE 扫描。
AT+BLEPAIR=<参数>	与对端设备发起配对。
AT+BLEGETPAIREDNUM	获取 BLE 设备配对设备数量。
AT+BLEGETPAIREDDEV	获取 BLE 设备配对设备。
AT+BLEGETPAIREDSTA=<参数>	获取 BLE 设备配对状态。
AT+BLEUNPAIR=<参数>	取消配对。
AT+BLEUNPAIRALL	取消所有配对。
AT+BLECONNPARUPD=<参数>	连接参数更新。
AT+BLECONN=<参数>	与 BLE 设备连接。
AT+BLEDISCONN=<参数>	断开 BLE 设备连接。
AT+BLEGAPREGCBK	注册 gap 回调函数。

### 3.1.1.2 gatts 模块 AT 命令

指令	描述
AT+GATTSREGSRV=<参数>	创建一个 GATT server。
AT+GATTSUNREG=<参数>	删除 GATT server，释放资源。
AT+GATTSADDSERV=<参数>	添加一个 GATT 服务。
AT+GATTSSYNCADDSERV=<参数>	添加一个 GATT 服务（同步）。
AT+GATTSADDCHAR=<参数>	为 GATT 服务添加一个特征。
AT+GATTSSYNCADDCHAR=<参数>	为 GATT 服务添加一个特征（同步）。

指令	描述
AT+GATTSADDDESCR=<参数>	为最新的特征添加一个描述符。
AT+GATTSSYNCADDDESCR=<参数>	为最新的特征添加一个描述符（同步）。
AT+GATTSSTARTSERV=<参数>	启动指定的 GATT 服务。
AT+GATTSDELALLSERV=<参数>	删除指定 server 上的所有服务。
AT+GATTSSENDRSP=<参数>	发送响应。
AT+GATTSSNDNTFY=<参数>	发送通知或指示。
AT+GATTSSNDNTFYBYUUID=<参数>	根据 uuid 发送通知或指示。
AT+GATTSREGCBK	注册 GATT 服务端回调函数。
AT+GATTSSETMTU=<参数>	在连接之前设置 server rx mtu。

### 3.1.1.3 gattc 模块 AT 命令

指令	描述
AT+GATTCREG=<参数>	创建一个 GATT client。
AT+GATTCUNREG=<参数>	删除 GATT client，释放资源。
AT+GATTCFNDSERV=<参数>	发现所有服务(可 by uuid)。
AT+GATTCFNDCHAR=<参数>	发现所有特征。
AT+GATTCFNDDESCR=<参数>	发现所有描述符。
AT+GATTCREADBYHDL=<参数>	读取 by hdl。
AT+GATTCREADBYUUID=<参数>	读取 by_uuid
AT+GATTCWRITEREQ=<参数>	写 by hdl req。
AT+GATTCWRITECMD=<参数>	写 by hdl cmd。
AT+GATTCEXCHMTU=<参数>	交换 mtu 请求。
AT+GATTCREGCBK	注册 GATT 客户端回调函数。



### 3.1.2 SLE

指令	描述
AT+SLEENABLE	SLE 使能。
AT+SLESETADVPAR	设置 SLE 广播参数。
AT+SLESETADVDATA	设置指令。
AT+SLESTARTADV	起 SLE 广播。
AT+SLESTOPADV	停 SLE 广播。
AT+SLESTARTSCAN	启动扫描。
AT+SLESTOPSCAN	关闭扫描。
AT+SLESETNAME	设置本端名称。
AT+SLEGETNAME	获取本端名称。
AT+SLESETADDR	设置本端地址。
AT+SLEGETADDR	获取本端地址。
AT+SLECONN	建立 SLE 连接。
AT+SLEDISCONN	断开 SLE 连接。
AT+SLESETPHY	设置 SLE PHY。
AT+SLESETDEFAULTCONNP	设置 SLE 默认连接参数。
AT+SLEPAIR	进行加密配对。
AT+SLEUNPAIR	移除加密配对。
AT+SLEGETPAIREDNUM	获取配对设备数目。
AT+SLEGETPAIRDEV	获取配对设备。
AT+SLEGETPAIRSTA	获取配对状态。
AT+SLEGETBONDDEV	获取绑定设备状态。
AT+SLECONNPARUPD	星闪逻辑链路更新参数。

指令	描述
AT+SLEREADPEERRSSI	读取对端 rssi。
AT+SSAPSADDSRV	注册服务端。
AT+SSAPSDELALLSRV	删除服务端。
AT+SSAPSADDSERV	添加服务。
AT+SSAPSSYNCADDSERV	添加服务同步。
AT+SSAPSADDPROPERTY	添加属性。
AT+SSAPSSYNCADDPROPERTY	添加属性同步。
AT+SSAPSADDDESCR	添加属性描述符。
AT+SSAPSSYNCADDDESCR	添加属性描述符同步。
AT+SSAPSSTARTSERV	start service。
AT+SSAPSSNDNTFY	服务端向客户端发送通知。
AT+SSAPSNTFYBYUUID	服务端向客户端通过 uuid 发送通知。
AT+SSAPSSNDRESP	服务端向客户端发送响应。
AT+SSAPSREGCBK	服务端注册回调函数。
AT+SSAPCREGCBK	注册 SSAPC 回调函数。
AT+SSAPCFNDSTRU	发现 service。
AT+SSAPCWRITECMD	客户端向服务端写入数据。
AT+SSAPCWITEREQ	客户端向服务端发送写请求。
AT+SSAPCEXCHINFO	客户端发起信息交换。
AT+SSAPCREADBYUUID	客户端通过 uuid 发送读请求。
AT+SSAPCREADREQ	客户端读取服务端属性数据。
AT+SLESETSCANPAR	设置扫描参数。
AT+SLEDISCONNALL	SLE 断开所有连接。

## 3.2 BT 业务 AT 指令描述

### 3.2.1 BLE

#### 3.2.1.1 gap 模块 AT 命令

##### 3.2.1.1.1 AT+BLEENABLE 使能 ble 协议栈

格式	AT+BLEENABLE
响应	打开 BLE 开关 成功: OK
参数说明	-
示例	AT+BLEENABLE
注意	-

##### 3.2.1.1.2 AT+BLEDISABLE 关闭 ble 协议栈

格式	AT+BLEDISABLE
响应	关闭 BLE 开关 成功: OK
参数说明	-
示例	AT+BLEDISABLE
注意	-

##### 3.2.1.1.3 AT+BLESETADDR 设置本地设备地址

格式	AT+BLESETADDR=<addr_type,addr>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;addr_type&gt;: 蓝牙设备类型</li> </ul>

	<ul style="list-style-type: none"> <li>• &lt;addr&gt;: 蓝牙设备地址</li> </ul>
示例	AT+BLESETADDR=0,0x112233445566
注意	设备类型取值范围为{0 (公共设备地址), 1 (随机设备地址), 2 (公共本端地址), 3 (随机静态本端地址)}, 设备地址为长度为 14 的字符串

#### 3.2.1.1.4 AT+BLEGETADDR 获取本地设备地址

格式	AT+BLEGETADDR
响应	<ul style="list-style-type: none"> <li>• 成功: 本地设备地址</li> <li>• 失败: ERROR</li> </ul>
参数说明	-
示例	AT+BLEGETADDR
注意	-

#### 3.2.1.1.5 AT+BLESETNAME 设置本地设备名称

格式	AT+BLESETNAME=<len,name>
响应	<ul style="list-style-type: none"> <li>• 成功: OK</li> <li>• 失败: ERROR</li> </ul>
参数说明	<len>: 本地设备名称长度; <name>: 本地设备名称
示例	AT+BLESETNAME=10,atcmdtest
注意	名称长度取值范围为[0,255], 设备名称长度为 len-1 的字符串, 名称最后默认存在 ' /0'

#### 3.2.1.1.6 AT+BLEGETNAME 获取本地设备名称

格式	AT+BLEGETNAME
----	---------------

响应	<ul style="list-style-type: none"> <li>成功：本地设备名称</li> <li>失败：ERROR</li> </ul>
参数说明	-
示例	AT+BLEGETNAME
注意	-

### 3.2.1.1.7 AT+BLESETAPPEARANCE 设置本地设备外观

格式	AT+BLESETAPPEARANCE=<appearance>
响应	<ul style="list-style-type: none"> <li>成功：OK</li> <li>失败：ERROR</li> </ul>
参数说明	<appearance>：本地设备外观；
示例	AT+BLESETAPPEARANCE=961
注意	参数值应为规定值，示例中 961 为键盘的外观值，具体请参考表 3-1

### 3.2.1.1.8 AT+BLESETADVDATA 设置 BLE 广播数据

格式	AT+BLESETADVDATA=<adv_length,adv_data,scan_rsp_length,scan_rsp_data,adv_id>
响应	<ul style="list-style-type: none"> <li>成功：OK</li> <li>失败：ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;adv_length&gt;：广播数据长度</li> <li>&lt;adv_data&gt;：广播数据</li> <li>&lt;scan_rsp_length&gt;：扫描返回数据长度</li> <li>&lt;scan_rsp_data&gt;：扫描返回数据</li> <li>&lt;adv_id&gt;：广播 id</li> </ul>
示例	AT+BLESETADVDATA=6,0x112233445566,0,0,1
注意	广播数据长度单位为字节，所以广播数据应为长度两倍的字符串，扫描返回数据同理，广播 ID 取值范围为[1,255]

### 3.2.1.1.9 AT+BLESETADVPAR 设置广播数据参数

格式	AT+BLESETADVPAR=<min_interval,max_interval,adv_type,own_addr,peer_addr_type,peer_addr,channel_map,adv_filter_policy,tx_power,duration,adv_id>
响应	<ul style="list-style-type: none"> <li>成功：OK</li> <li>失败：ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;min_interval&gt;：最小扫描间隔；取值范围[0x20, 0x4000],Time=N×0.625ms</li> <li>&lt;max_interval&gt;：最大扫描间隔；取值范围[0x20, 0x4000],Time=N×0.625ms</li> <li>&lt;adv_type&gt;：广播类型</li> <li>&lt;own_addr&gt;：本端地址</li> <li>&lt;peer_addr_type&gt;：对端地址类型</li> <li>&lt;peer_addr&gt;：对端地址</li> <li>&lt;channel_map&gt;：信道；取值范围为[0x01, 0x07]</li> <li>&lt;adv_filter_policy&gt;：过滤策略</li> <li>&lt;tx_power&gt;：扫描功率</li> <li>&lt;duration&gt;：扫描周期；仅取值为 0</li> <li>&lt;adv_id&gt;：广播 ID；取值范围[1, 255]</li> </ul>
示例	AT+BLESETADVPAR=48,48,0,0x112233445577,0,0x112233445566,7,0,1,0,1
注意	-

### 3.2.1.1.10 AT+BLESTARTADV 开始发送 BLE 广播

格式	AT+BLESTARTADV=<adv_id>
响应	<ul style="list-style-type: none"> <li>成功：OK</li> <li>失败：ERROR</li> </ul>
参数说明	<adv_id>：广播 id
示例	AT+BLESTARTADV=1

注意	-
----	---

### 3.2.1.1.11 AT+BLESTOPADV 停止发送 BLE 广播

格式	AT+BLESTOPADV=<adv_id>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<adv_id>: 广播 id
示例	AT+BLESTOPADV=1
注意	-

### 3.2.1.1.12 AT+BLESETSCANPAR 设置 BLE 扫描参数

格式	AT+BLESETSCANPAR=<scan_interval,scan_window,scan_type,scan_phy,scan_rsp_policy>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<p>&lt;scan_interval&gt;: 扫描间隔; 取值范围[0x04, 0x4000], Time=N×0.625ms</p> <p>&lt;scan_window&gt;: 扫描窗口; 取值范围[0x04, 0x4000], Time=N×0.625ms</p> <p>&lt;scan_type&gt;: 扫描类型; {0 (被动扫描), 1 (主动扫描)}</p> <p>&lt;scan_phy&gt;: 扫描 phy</p> <p>&lt;scan_rsp_policy&gt;: 扫描过滤策略</p>
示例	AT+BLESETSCANPAR=0x48,0x48,0,1,0
注意	-

### 3.2.1.1.13 AT+BLESTARTSCAN 启动 BLE 扫描

格式	AT+BLESTARTSCAN
----	-----------------

响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	-
示例	AT+BLESTARTSCAN
注意	-

#### 3.2.1.1.14 AT+BLESTOPSCAN 停止 BLE 扫描

格式	AT+BLESTOPSCAN
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	-
示例	AT+BLESTOPSCAN
注意	-

#### 3.2.1.1.15 AT+BLEPAIR 与对端设备发起配对

格式	AT+BLEPAIR=<addr_type,addr>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;addr_type&gt;: 蓝牙设备类型</li> <li>&lt;addr&gt;: 蓝牙设备地址</li> </ul>
示例	AT+BLEPAIR=0,0x112233445566
注意	设备类型取值范围为{0 (公共设备地址) ,1 (随机设备地址) ,2 (公共本端地址) ,3 (随机静态本端地址) },设备地址为长度为 14 的字符串



### 3.2.1.1.16 AT+BLEGETPAIREDNUM 获取 BLE 设备配对设备数量

格式	AT+BLEGETPAIREDNUM
响应	<ul style="list-style-type: none"> <li>成功：配对设备数量</li> <li>失败：ERROR</li> </ul>
参数说明	-
示例	AT+BLEGETPAIREDNUM
注意	-

### 3.2.1.1.17 AT+BLEGETPAIREDDEV 获取 BLE 设备配对设备

格式	AT+BLEGETPAIREDDEV
响应	<ul style="list-style-type: none"> <li>成功：配对设备地址</li> <li>失败：ERROR</li> </ul>
参数说明	-
示例	AT+BLEGETPAIREDDEV
注意	-

### 3.2.1.1.18 AT+BLEGETPAIREDSTA 获取 BLE 设备配对状态

格式	AT+BLEGETPAIREDSTA=<addr_type,addr>
响应	<ul style="list-style-type: none"> <li>成功：BLE 设备配对状态</li> <li>失败：ERROR</li> </ul>
参数说明	<addr_type>：蓝牙设备类型 <addr>：蓝牙设备地址
示例	AT+BLEGETPAIREDSTA=0,0x112233445566
注意	设备类型取值范围为{0（公共设备地址）,1（随机设备地址）,2（公共本端地址）,3（随机静态本端地址）},设备地址为长度为 14 的字符串

### 3.2.1.1.19 AT+BLEUNPAIR 取消配对

格式	AT+BLEUNPAIR=<addr_type,addr>
响应	<ul style="list-style-type: none"> <li>成功：断连</li> <li>失败：ERROR</li> </ul>
参数说明	<addr_type>：蓝牙设备类型 <addr>：蓝牙设备地址
示例	AT+BLEUNPAIR=0,0x112233445566
注意	设备类型取值范围为{0（公共设备地址）,1（随机设备地址）,2（公共本端地址）,3（随机静态本端地址）},设备地址为长度为 14 的字符串

### 3.2.1.1.20 AT+BLEUNPAIR 取消所有配对

格式	AT+BLEUNPAIR
响应	<ul style="list-style-type: none"> <li>成功：断连</li> <li>失败：ERROR</li> </ul>
参数说明	-
示例	AT+BLEUNPAIR
注意	-

### 3.2.1.1.21 AT+BLECONNPARDUPD 更新连接参数

格式	AT+BLECONNPARDUPD=<conn_handle,interval_min,interval_max,slave_latency,timeout_multiplier>
响应	<ul style="list-style-type: none"> <li>成功：OK</li> <li>失败：ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;conn_handle&gt;：连接句柄</li> <li>&lt;interval_min&gt;：链路调度最小间隔，[0x06, 0x0C80], Time=N×1.25ms</li> </ul>

	<ul style="list-style-type: none"> <li>• &lt;interval_max&gt;: 链路调度最大间隔, [0x06, 0x0C80], Time=N×1.25ms</li> <li>• &lt;slave_latency&gt;: 延迟周期, 单位 slot (该值表示在设置值的周期内可以不回复, 为 0 时则表示每包都需回复)</li> <li>• &lt;timeout_multiplier&gt;: 超时断连间隔</li> </ul>
示例	AT+BLECONNPARUPD=0, 0x48,0x48,0,500
注意	-

### 3.2.1.1.22 AT+BLECONN 与 BLE 设备连接

格式	AT+BLECONN=<addr_type,addr>
响应	<ul style="list-style-type: none"> <li>• 成功: OK</li> <li>• 失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>• &lt;addr_type&gt;: 蓝牙设备类型</li> <li>• &lt;addr&gt;: 蓝牙设备地址</li> </ul>
示例	AT+BLECONN=0,0x112233445566
注意	设备类型取值范围为{0 (公共设备地址), 1 (随机设备地址), 2 (公共本端地址), 3 (随机静态本端地址)}, 设备地址为长度为 14 的字符串

### 3.2.1.1.23 AT+BLEDISCONN 与 BLE 设备断开连接

格式	AT+BLEDISCONN=<addr_type,addr>
响应	<ul style="list-style-type: none"> <li>• 成功: OK</li> <li>• 失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>• &lt;addr_type&gt;: 蓝牙设备类型</li> <li>• &lt;addr&gt;: 蓝牙设备地址</li> </ul>
示例	AT+BLEDISCONN=0,0x112233445566
注意	设备类型取值范围为{0 (公共设备地址), 1 (随机设备地址), 2 (公共本

	端地址),3 (随机静态本端地址)},设备地址为长度为 14 的字符串
--	-------------------------------------

### 3.2.1.1.24 AT+BLEGAPREGCBK 注册 BLE 回调函数

格式	AT+BLEGAPREGCBK
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	-
示例	AT+BLEGAPREGCBK
注意	-

## 3.2.1.2 gatts 模块 AT 命令

### 3.2.1.2.1 AT+GATTSREGRV 创建一个 GATT server

格式	AT+GATTSREGRV=<uuid>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<uuid>: 应用 uuid
示例	AT+GATTSREGRV=0x1122
注意	-

### 3.2.1.2.2 AT+GATTSUNREG 删除 GATT server, 释放资源

格式	AT+GATTSUNREG=<uuid>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<uuid>: 应用 uuid
示例	AT+GATTSUNREG=0x1122

注意	-
----	---

### 3.2.1.2.3 AT+GATTSADDSEV 添加一个 GATT 服务

格式	AT+GATTSADDSEV=<server_id,svc_uuid,is_primary_flag>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;server_id&gt;: 服务端 id</li> <li>&lt;svc_uuid&gt;: 服务 uuid</li> <li>&lt;is_primary_flag&gt;: 是否是首要服务</li> </ul>
示例	AT+GATTSADDSEV=1,0x1812,1
注意	-

### 3.2.1.2.4 AT+GATTSSYNCAADDSEV 添加一个 GATT 服务（同步）

格式	AT+GATTSSYNCAADDSEV=<server_id,svc_uuid,is_primary_flag>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;server_id&gt;: 服务端 id</li> <li>&lt;svc_uuid&gt;: 服务 uuid</li> <li>&lt;is_primary_flag&gt;: 是否是首要服务</li> </ul>
示例	AT+GATTSSYNCAADDSEV=1,0x1812,1
注意	-

### 3.2.1.2.5 AT+GATTSADDCHAR 为 GATT 服务添加一个特征

格式	AT+GATTSADDCHAR=<server_id,service_handle,chara_uuid,permissions,properties,value_len,value>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> </ul>

	<ul style="list-style-type: none"> <li>失败：ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;server_id&gt;：服务端 id</li> <li>&lt;service_handle&gt;：服务句柄</li> <li>&lt;chara_uuid&gt;：特征 uuid</li> <li>&lt;permissions&gt;：权限</li> <li>&lt;propertise&gt;：特性</li> <li>&lt;value_len&gt;：值长</li> <li>&lt;value&gt;：值</li> </ul>
示例	AT+GATTSADDCHAR=1,1,0x2a4a,0x01,0x02,4,01010003
注意	值是长度为值长度两倍的字符串

### 3.2.1.2.6 AT+GATTSSYNCADDCHAR 为 GATT 服务添加一个特征（同步）

格式	AT+GATTSSYNCADDCHAR=<server_id,service_handle,chara_uuid,permissions,properties,value_len,value>
响应	<ul style="list-style-type: none"> <li>成功：OK</li> <li>失败：ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;server_id&gt;：服务端 id</li> <li>&lt;service_handle&gt;：服务句柄</li> <li>&lt;chara_uuid&gt;：特征 uuid</li> <li>&lt;permissions&gt;：权限</li> <li>&lt;propertise&gt;：特性</li> <li>&lt;value_len&gt;：值长度</li> <li>&lt;value&gt;：值</li> </ul>
示例	AT+GATTSSYNCADDCHAR=1,1,0x2a4a,0x01,0x02,4,01010003
注意	值是长度为值长度两倍的字符串

### 3.2.1.2.7 AT+GATTSADDDDESCR 为最新的特征添加一个描述符

格式	AT+GATTSADDDDESCR=<server_id,service_handle,chara_uuid,permis
----	---

	sions,value_len,value>
响应	<ul style="list-style-type: none"> <li>成功：OK</li> <li>失败：ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;server_id&gt;：服务端 id</li> <li>&lt;service_handle&gt;：服务句柄</li> <li>&lt;chara_uuid&gt;：特征 uuid</li> <li>&lt;permissions&gt;：权限</li> <li>&lt;value_len&gt;：值长度</li> <li>&lt;value&gt;：值</li> </ul>
示例	AT+GATTSSADDDESCR=1,1,0x2902,0x03,2,0100
注意	值是长度为值长度两倍的字符串

### 3.2.1.2.8 AT+GATTSSYNCAADDDESCR 为最新的特征添加一个描述符（同步）

格式	AT+GATTSSYNCAADDDESCR=<server_id,service_handle,chara_uuid,permissions,value_len,value>
响应	<ul style="list-style-type: none"> <li>成功：OK</li> <li>失败：ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;server_id&gt;：服务端 id</li> <li>&lt;service_handle&gt;：服务句柄</li> <li>&lt;chara_uuid&gt;：特征 uuid</li> <li>&lt;permissions&gt;：权限</li> <li>&lt;value_len&gt;：值长度</li> <li>&lt;value&gt;：值</li> </ul>
示例	AT+GATTSSYNCAADDDESCR=1,1,0x2902,0x03,2,0100
注意	值是长度为值长度两倍的字符串

### 3.2.1.2.9 AT+GATTSSSTARTSERV 启动指定的 GATT 服务

格式	AT+GATTSSSTARTSERV=<server_id,service_handle>
----	---

响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;server_id&gt;: 服务端 id</li> <li>&lt;service_handle&gt;: 服务句柄</li> </ul>
示例	AT+GATTSSSTARTSERV=1,1
注意	-

### 3.2.1.2.10 AT+GATTSDDELALLSERV 删除指定 server 上的所有服务

格式	AT+GATTSDDELALLSERV=<server_id>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<server_id>: 服务端 id
示例	AT+GATTSDDELALLSERV=1
注意	-

### 3.2.1.2.11 AT+GATTSSSENDRESP 发送响应

格式	AT+GATTSSSENDRESP=<server_id,conn_handle,request_id,status,offset,value_len,value>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;server_id&gt;: 服务端 id</li> <li>&lt;conn_handle&gt;: 连接句柄</li> <li>&lt;request_id&gt;: 请求 id</li> <li>&lt;status&gt;: 请求结果</li> <li>&lt;offset&gt;: 偏移</li> <li>&lt;value_len&gt;: 值长度</li> <li>&lt;value&gt;: 值</li> </ul>



示例	AT+GATTSSENDRSP=1,0,req_id,0,0,2,0x4562
注意	值是长度为值长度两倍的字符串

### 3.2.1.2.12 AT+GATTSSNDNTFY 发送通知或指示

格式	AT+GATTSSNDNTFY=<server_id,conn_handle,attr_handle,value_len,value>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;server_id&gt;: 服务端 id</li> <li>&lt;conn_handle&gt;: 连接句柄</li> <li>&lt;attr_handle&gt;: 偏移</li> <li>&lt;value_len&gt;: 值长度</li> <li>&lt;value&gt;: 值</li> </ul>
示例	AT+GATTSSNDNTFY=1,0,9,7,0x00000000000014
注意	值是长度为值长度两倍的字符串

### 3.2.1.2.13 AT+GATTSSNDNTFYBYUUID 根据 uuid 发送通知或指示

格式	AT+GATTSSNDNTFYBYUUID=<server_id,conn_handle,chara_uuid,start_handle,end_handle,value_len,value>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;server_id&gt;: 服务端 id</li> <li>&lt;conn_handle&gt;: 连接句柄</li> <li>&lt;chara_uuid&gt;: 特征 uuid</li> <li>&lt;start_handle&gt;: 起始句柄</li> <li>&lt;end_handle&gt;: 结束句柄</li> <li>&lt;value_len&gt;: 值长度</li> <li>&lt;value&gt;: 值</li> </ul>

示例	AT+GATTSSNDNTFYBYUUID=1,0,0x2a4d,1,9,7,0x0000000000000014
注意	值是长度为值长度两倍的字符串

### 3.2.1.2.14 AT+GATTSREGCBK 注册 GATT 服务端回调函数

格式	AT+GATTSREGCBK
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	-
示例	AT+GATTSREGCBK
注意	-

### 3.2.1.2.15 AT+GATTSSETMTU 在连接之前设置 server rx mtu

格式	AT+GATTSSETMTU=<server_id,mtu_size>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;server_id&gt;: 服务端 id</li> <li>&lt;mtu_size&gt;: mtu 大小</li> </ul>
示例	AT+GATTSSETMTU=1,23
注意	-

## 3.2.1.3 gattc 模块 AT 命令

### 3.2.1.3.1 AT+GATTCREG 创建一个 GATT client

格式	AT+GATTCREG=<uuid>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>

参数说明	<uuid>: 应用 uuid
示例	AT+GATTCREG=0x1212
注意	-

### 3.2.1.3.2 AT+GATTCUNREG 删除 GATT client, 释放资源

格式	AT+GATTCUNREG=<client_id>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<client_id>: 客户端 id
示例	AT+GATTCUNREG=1
注意	-

### 3.2.1.3.3 AT+GATTCFNDSEV 发现服务

格式	AT+GATTCFNDSEV=<client_id,conn_id,uuid>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<client_id>: 客户端 id; <conn_id>: 连接 id <uuid>: 应用 uuid
示例	AT+GATTCFNDSEV=1,1,0x1212
注意	-

### 3.2.1.3.4 AT+GATTCFNDCHAR 发现特征

格式	AT+GATTCFNDCHAR=<client_id,conn_id,server_handle,uuid>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> </ul>

	<ul style="list-style-type: none"> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;client_id&gt;: 客户端 id</li> <li>&lt;conn_id&gt;: 连接 id</li> <li>&lt;server_handle&gt;: 服务句柄</li> <li>&lt;uuid&gt;: 应用 uuid</li> </ul>
示例	AT+GATTCFNDCHAR=1,1,0,0x1212
注意	-

### 3.2.1.3.5 AT+GATTCFNDDDESCR 发现描述符

格式	AT+GATTCFNDDDESCR=<client_id,conn_id,handle>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;client_id&gt;: 客户端 id</li> <li>&lt;conn_id&gt;: 连接 id</li> <li>&lt;handle&gt;: 服务句柄</li> </ul>
示例	AT+GATTCFNDDDESCR=1,1,0
注意	-

### 3.2.1.3.6 AT+GATTREADBYHDL 读取 by hdl

格式	AT+GATTREADBYHDL=<client_id,conn_id,handle>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;client_id&gt;: 客户端 id</li> <li>&lt;conn_id&gt;: 连接 id</li> <li>&lt;handle&gt;: 服务句柄</li> </ul>
示例	AT+GATTREADBYHDL=1,1,0

注意	-
----	---

### 3.2.1.3.7 AT+GATTCREADBYUUID 读取 by\_uuid

格式	AT+GATTCREADBYUUID=<client_id,conn_id,start_hdl,end_hdl,uuid>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;client_id&gt;: 客户端 id</li> <li>&lt;conn_id&gt;: 连接 id</li> <li>&lt;start_hdl&gt;: 起始句柄</li> <li>&lt;end_hdl&gt;: 结束句柄</li> <li>&lt;uuid&gt;: 想要读的 uuid</li> </ul>
示例	AT+GATTCREADBYUUID=1,0,13,13,2a4d
注意	-

### 3.2.1.3.8 AT+GATTCWRITEREQ 写 by hdl req

格式	AT+GATTCWRITEREQ=<client_id,conn_id,handle,data_len,data>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;client_id&gt;: 客户端 id</li> <li>&lt;conn_id&gt;: 连接 id</li> <li>&lt;handle&gt;: 句柄</li> <li>&lt;data_len&gt;: 数据长度</li> <li>&lt;data&gt;: 数据</li> </ul>
示例	AT+GATTCWRITEREQ=1,0,13,1,0x11
注意	-

### 3.2.1.3.9 AT+GATTCWRITECMD 写 by hdl cmd

格式	AT+GATTCWRITECMD=<client_id,conn_id,handle,data_len,data>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;client_id&gt;: 客户端 id</li> <li>&lt;conn_id&gt;: 连接 id</li> <li>&lt;handle&gt;: 句柄</li> <li>&lt;data_len&gt;: 数据长度</li> <li>&lt;data&gt;: 数据</li> </ul>
示例	AT+GATTCWRITECMD=1,0,13,1,0x11
注意	-

### 3.2.1.3.10 AT+GATTCEXCHMTU 交换 MTU 请求

格式	AT+GATTCEXCHMTU=<server_id,conn_id,mtu_size>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;server_id&gt;: 服务端 id</li> <li>&lt;conn_id&gt;: 连接 id</li> <li>&lt;mtu_size&gt;: client rx mtu 大小</li> </ul>
示例	AT+GATTCEXCHMTU=1,0,100
注意	-

### 3.2.1.3.11 AT+GATTCREGCBK 注册 GATT 客户端回调函数

格式	AT+GATTCREGCBK
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>

参数说明	-
示例	AT+GATTCREGCBK
注意	-

## 3.2.2 SLE

### 3.2.2.1 SLE Device Discovery 模块指令描述

#### 3.2.2.1.1 SLE 使能

设置指令	AT+SLEENABLE
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	-
示例	AT+SLEENABLE
注意	-

#### 3.2.2.1.2 SLE 去使能

设置指令	AT+SLEDISABLE
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	-
示例	AT+SLEDISABLE
注意	-

#### 3.2.2.1.3 设置 SLE 广播参数

设置指令	AT+SLESETADVPAR=<announce_handle>,<announce_mode>,<announce_interval_min>,<announce_interval_max>,<own_addr_type>,<own_addr_addr>,<peer_addr_type>,<peer_addr_addr>
------	---

响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;announce_handle&gt;: 设备公开句柄, 取值范围[0, 0x10]</li> <li>&lt;announce_mode&gt;: 设备公开类型 /*可连接可扫描*/</li> <li>&lt;announce_interval_min&gt;: 最小设备公开周期, 0x000020~0xffffffff, 单位 125μs</li> <li>&lt;announce_interval_max&gt;: 最大设备公开周期, 0x000020~0xffffffff, 单位 125μs</li> <li>&lt;own_addr_type&gt;: SLE 本端地址类型 0: 公有地址, 6 : 随机地址</li> <li>&lt;own_addr_addr&gt;: SLE 本端设备地址</li> <li>&lt;peer_addr_type&gt;: SLE 对端设备地址类型 0: 公有地址, 6 : 随机地址</li> <li>&lt;peer_addr_addr&gt;: SLE 对端设备地址</li> </ul>
示例	AT+SLESETADVPAR=1,3,200,200,0,000000000000,0,000000000000
注意	此命令需在 SLE 使能 AT+SLEENABLE 后下发.

#### 3.2.2.1.4 设置 SLE 广播数据

设置指令	AT+SLESETADVDATA=<adv_handle>,<announce_data_len>,<seek_rsp_data_len>,<announce_data>,<seek_rsp_data>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;adv_handle&gt;: 广播 handle, 取值范围[0, 0x10]</li> <li>&lt;announce_data_len&gt;: 设备公开数据长度</li> <li>&lt;seek_rsp_data_len&gt;: 扫描响应数据长度</li> <li>&lt;announce_data&gt;: 设备公开数据 (hex 类型字符串, 最大长度</li> </ul>



	521 个字符) <ul style="list-style-type: none"> <li>• &lt;seek_rsp_data&gt;: 扫描响应数据 (hex 类型字符串, 最大长度 521 个字符)</li> </ul>
示例	AT+SLESETADVDATA=1,10,4,aabbccddeeff11223344,11224455
注意	此命令需在 SLE 使能 AT+SLEENABLE 后下发

### 3.2.2.1.5 起 SLE 广播

设置指令	AT+SLESTARTADV=<adv_enable>
响应	<ul style="list-style-type: none"> <li>• 成功: OK</li> <li>• 失败: ERROR</li> </ul>
参数说明	<adv_handle>: adv handle, 取值范围[0, 0x10]
示例	AT+SLESTARTADV=1
注意	此命令需在 SLE 使能 AT+SLEENABLE 后下发

### 3.2.2.1.6 停 SLE 广播

设置指令	AT+SLESTOPADV=<adv_handle>
响应	<ul style="list-style-type: none"> <li>• 成功: OK</li> <li>• 失败: ERROR</li> </ul>
参数说明	<adv_handle>: adv handle
示例	AT+SLESTOPADV=1
注意	此命令需在 SLE 起广播 AT+SLESTARTADV=1 后下发

### 3.2.2.1.7 设置扫描参数

设置指令	AT+SLESETSCANPAR=<scan_type>,<scan_interval>,<scan_window>
响应	<ul style="list-style-type: none"> <li>• 成功: OK</li> </ul>

	<ul style="list-style-type: none"> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;scan_type&gt;: 扫描类型 0: 被动扫描 1: 主动扫描</li> <li>&lt;scan_interval&gt;: 扫描间隔, 取值范围[0x14, 0xFFFF], 单位 125 <math>\mu</math>s</li> <li>&lt;scan_window&gt;: 扫描窗口, 取值范围[0x14, 0xFFFF], 单位 125 <math>\mu</math>s</li> </ul>
示例	AT+SLESETSCANPAR=0,0x48,0x48
注意	此命令需在 SLE 起扫描 AT+SLESTARTSCAN 前下发

### 3.2.2.1.8 使能扫描

设置指令	AT+SLESTARTSCAN
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	-
示例	AT+SLESTARTSCAN
注意	-

### 3.2.2.1.9 关闭扫描

设置指令	AT+SLESTOPSCAN
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	-
示例	AT+SLESTOPSCAN
注意	-

### 3.2.2.1.10 设置本端名称

设置指令	AT+SLESETNAME
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<p>&lt;len&gt;: name 长度</p> <p>&lt;name&gt;: 名字</p>
示例	AT+SLESETNAME=7,SDKTEST
注意	-

### 3.2.2.1.11 获取本端名称

设置指令	AT+SLEGETNAME
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	-
示例	AT+SLEGETNAME
注意	-

### 3.2.2.1.12 设置本端地址

设置指令	AT+SLESETADDR
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;addr_type&gt;: 地址类型</li> <li>&lt;addr&gt;: 地址</li> </ul>
示例	AT+SLESETADDR=0,0x0000000000001

注意	-
----	---

### 3.2.2.1.13 获取本端地址

设置指令	AT+SLEGETADDR
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	-
示例	AT+SLEGETADDR
注意	-

## 3.2.2.2 SLE Connection manager 模块指令描述

### 3.2.2.2.1 建立 SLE 连接

当前只支持 0、6 两种，通常常用的类型是 0。地址类型会校验，如果不匹配会返回错误。

设置指令	AT+SLECONN=<sle_addr_type>,<sle_addr>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul> <p>连接成功后会打印[connected]字样以及对端设备地址与 handle 值</p>
参数说明	<ul style="list-style-type: none"> <li>&lt;sle_addr_type&gt;: SLE 设备地址类型 <ul style="list-style-type: none"> <li>0: 公有地址</li> <li>6 : 随机地址</li> </ul> </li> <li>&lt;sle_addr&gt;: SLE 设备地址</li> </ul>
示例	AT+SLECONN=0,0x000000000000
注意	-

### 3.2.2.2.2 星闪逻辑链路更新参数

设置指令	AT+SLECONNPARUPD=<conn_id>,<interval_min>,<interval_max>,<max_latency>,<supervision_timeout>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;conn_id&gt;: 连接 ID</li> <li>&lt;interval_min&gt;: 链路调度最小间隔, 取值范围[0x0002, 0x32000], 单位 125μs</li> <li>&lt;interval_max&gt;: 链路调度最大间隔, 取值范围[0x0002, 0x32000], 单位 125μs</li> <li>&lt;max_latency&gt;: 延迟周期, 单位 slot (该值表示在设置值的周期内可以不回复, 为 0 时则表示每包都需回复)</li> <li>&lt;supervision_timeout&gt;: 超时时间, 单位 10ms</li> </ul>
示例	AT+SLECONNPARUPD=0,20,20,0,500
注意	-

### 3.2.2.2.3 星闪读取远端 rssi

该命令的作用: 读取当前 SLE ACB 链路的 RSSI 信号强度, RSSI (Received Signal Strength Indicator) 是接收信号的强度指示。具体含义

设置指令	AT+SLEREADPEERRSSI=<conn_id>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<conn_id>: 连接 ID
示例	AT+SLEREADPEERRSSI=0
注意	-

### 3.2.2.2.4 断开 SLE 连接

设置指令	AT+SLEDISCONN=<sle_addr_type>,<sle_addr>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul> <p>连接成功后会打印[disconnected]字样以及对端设备地址与 handle 值</p>
参数说明	<ul style="list-style-type: none"> <li>&lt;sle_addr_type&gt;: SLE 设备地址类型 0: 公有地址 6: 随机地址</li> <li>&lt;sle_addr&gt;: SLE 设备地址</li> </ul>
示例	AT+SLEDISCONN=0,000000000000
注意	-

### 3.2.2.2.5 设置 SLE PHY

默认 1M、4M 暂时不支持

该命令作用: 某些场景下需要高的传输速率, 此时就通过设置 tx\_phy、rx\_phy 参数为 2M。如果两端都支持 2M, 才能设置成功。

设置指令	AT+SLESETPHY=<conn_id>,<tx_phy>,<rx_phy>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;conn_id&gt;: 连接 id</li> <li>&lt;tx_phy&gt;: tx phy 值 0: 1M PHY 1: 2M PHY 2: 4M PHY</li> <li>&lt;rx_phy&gt;: tx phy 值 0: 1M PHY 1: 2M PHY</li> </ul>

	2: 4M PHY
示例	AT+SLESETPHY=0,1,1
注意	-

### 3.2.2.2.6 设置 SLE 默认连接参数

设置指令	AT+ SLESETDEFAULTCONNP =<enable_filter_policy>,<initiate_phys>,<gt_negotiate>,<scan_interval>,<scan_window>,<max_interval>,<min_interval>,<timeout>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;enable_filter_policy&gt;: 是否打开链路过滤</li> <li>&lt;initiate_phys&gt;: 链路扫描带宽</li> <li>&lt;gt_negotiate&gt;: 链路建立时是否进行 G 和 T 交互</li> <li>&lt;scan_interval&gt;: 扫描对端设备的 interval 最小值: 0x14 最大值: 无上限</li> <li>&lt;scan_window&gt;: 扫描对端设备的 windows 最小值: 0x14 最大值: 传入的&lt;scan_interval&gt;</li> <li>&lt;max_interval&gt;: 链路最大调度 interval 最大值: 32000</li> <li>&lt;min_interval&gt;: 链路最小调度 interval 最小值: 10</li> <li>&lt;timeout&gt;: 链路超时时间(N×10ms) 最小值: 10 最大值: 3200</li> </ul>
示例	AT+SLESETDEFAULTCONNP=0,1,0x1,0x20,0x20,0x64,0x64,0x1FC
注意	-

### 3.2.2.2.7 SLE 断开所有连接

设置指令	AT+SLEDISCONNALL
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	
示例	AT+SLEDISCONNALL
注意	-

### 3.2.2.3 SLE security managemen 模块指令描述

#### 3.2.2.3.1 进行加密配对

设置指令	AT+SLEPAIR=<sle_addr_type>,<sle_addr>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;sle_addr_type&gt;: SLE 设备地址类型 0: 公有地址, 6: 随机地址</li> <li>&lt;sle_addr&gt;: SLE 设备地址</li> </ul>
示例	AT+SLEPAIR=0,000000000000
注意	需在 sle 建立连接以后, 和对端启动加密配对

#### 3.2.2.3.2 移除加密配对

设置指令	AT+SLEUNPAIR=<sle_addr_type>,<sle_addr>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;sle_addr_type&gt;: SLE 设备地址类型</li> </ul>



	0: 公有地址, 6: 随机地址 • <sle_addr>: SLE 设备地址
示例	AT+SLEUNPAIR=0,000000000000
注意	-

### 3.2.2.3.3 获取配对设备数目

设置指令	AT+SLEGETPAIREDNUM
响应	• 成功: OK • 失败: ERROR
参数说明	-
示例	AT+SLEGETPAIREDNUM
注意	-

### 3.2.2.3.4 获取配对设备

设置指令	AT+SLEGETPAIRDEV
响应	• 成功: OK • 失败: ERROR
参数说明	-
示例	AT+SLEGETPAIRDEV
注意	--

### 3.2.2.3.5 获取设备配对状态

设置指令	AT+SLEGETPAIRSTA=<sle_addr_type>,<sle_addr>
响应	• 成功: OK

	<ul style="list-style-type: none"> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;sle_addr_type&gt;: SLE 设备地址类型 0: 公有地址, 6: 随机地址</li> <li>&lt;sle_addr&gt;: SLE 设备地址</li> </ul>
示例	AT+SLEUNPAIR=0,000000000000
注意	-

### 3.2.2.3.6 获取绑定设备

设置指令	AT+SLEGETBONDDEV
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	-
示例	AT+SLEGETBONDDEV
注意	-

## 3.2.2.4 ssap server 模块指令描述

### 3.2.2.4.1 注册服务端

设置指令	AT+SSAPSADDSRV=<uuid>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	-
示例	AT+SSAPSADDSRV=0x1234
注意	-

### 3.2.2.4.2 去注册服务端

设置指令	AT+SSAPSDELALLSRV
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	-
示例	AT+SSAPSDELALLSRV
注意	-

### 3.2.2.4.3 添加服务

通常情况下注册为主要服务。

设置指令	AT+SSAPSADDSERV=<uuid>,<is_primary>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	-
示例	AT+SSAPSADDSERV=0x2222,1
注意	-

### 3.2.2.4.4 添加服务同步

设置指令	AT+SSAPSSYNCADDSERV=<uuid>,<is_primary>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	-
示例	AT+SSAPSSYNCADDSERV=0x2222,1
注意	-

属性参数说明:

```

typedef enum {
    SSAP_PERMISSION_READ                = 0x01,      /*可读*/
    SSAP_PERMISSION_WRITE               = 0x02,      /*可写*/
    SSAP_PERMISSION_ENCRYPTION_NEED     = 0x04,      /*需要加密*/
    SSAP_PERMISSION_AUTHENTICATION_NEED = 0x08,      /*需要认证*/
    SSAP_PERMISSION_AUTHORIZATION_NEED  = 0x10,      /*需要授权*/
} ssap_permission_t;

typedef enum {
    SSAP_OPERATE_INDICATION_BIT_READ = 0x01,      /*数据值可被读取*/
    SSAP_OPERATE_INDICATION_BIT_WRITE_NO_RSP = 0x02, /*数据值可被写入，写入后无反馈*/
    SSAP_OPERATE_INDICATION_BIT_WRITE = 0x04,     /*数据值可被写入，写入后产生反馈给客户端*/
    SSAP_OPERATE_INDICATION_BIT_NOTIFY = 0x08,    /*数据值通过通知方式传递给客户端*/
    SSAP_OPERATE_INDICATION_BIT_INDICATE = 0x10,   /*数据值通过指示方式传递给客户端*/
    SSAP_OPERATE_INDICATION_BIT_BROADCAST = 0x20, /*数据值可携带在广播中*/
    SSAP_OPERATE_INDICATION_BIT_DESCRIPTOR_WRITE = 0x100, /*数据值说明描述符可被写入*/
    SSAP_OPERATE_INDICATION_MAX,                  /*操作指示最大值*/
} ssap_operate_indication_t;
    
```

### 3.2.2.4.5 添加属性

设置指令	AT+SSAPSADDPROPERTY=<service_handle>,<uuid>,<permissions>,<operate_indication>,<value_len>,<value>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;service_handle&gt;: 服务的 handle</li> <li>&lt;uuid&gt;: SSAP 特征 UUID</li> <li>&lt;permissions&gt;: 特征权限</li> <li>&lt;operate_indication&gt;: 操作指示</li> </ul>

	<ul style="list-style-type: none"> <li>• &lt;value_len&gt;: 响应的数据长度</li> <li>• &lt;value&gt;: 响应的数据</li> </ul>
示例	AT+SSAPSADDPROPERTY=1,0x2323,5,5,2,0x1234
注意	-

### 3.2.2.4.6 添加属性同步

设置指令	AT+SSAPSSYNCADDPROPERTY=<service_handle>,<uuid>,<permissions>,<operate_indication>,<value_len>,<value>
响应	<ul style="list-style-type: none"> <li>• 成功: OK</li> <li>• 失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>• &lt;service_handle&gt;: 服务的 handle</li> <li>• &lt;uuid&gt;: SSAP 特征 UUID</li> <li>• &lt;permissions&gt;: 特征权限</li> <li>• &lt;operate_indication&gt;: 操作指示</li> <li>• &lt;value_len&gt;: 响应的数据长度</li> <li>• &lt;value&gt;: 响应的数据</li> </ul>
示例	AT+SSAPSSYNCADDPROPERTY=1,0x2323,5,5,2,0x1234
注意	-

### 3.2.2.4.7 添加属性描述符

设置指令	AT+SSAPSADDDESCRIPTOR=<service_handle>,<property_handle>,<uuid>,<permissions>,<operate_indication>,<type>,<value_len>,<value>
响应	<ul style="list-style-type: none"> <li>• 成功: OK</li> <li>• 失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>• &lt;service_handle&gt;: 服务 handle</li> <li>• &lt;property_handle&gt;: 属性 handle</li> <li>• &lt;uuid&gt;: SSAP 描述符 UUID</li> </ul>

	<ul style="list-style-type: none"> <li>• &lt;permissions&gt;: 特征权限</li> <li>• &lt;operate_indication&gt;: 操作指示</li> <li>• &lt;type&gt;: 描述符类型</li> <li>• &lt;value_len&gt;: 数据长度</li> <li>• &lt;value&gt;: 数据</li> </ul>
示例	AT+SSAPSADDDESCR=1,2,0x3333,5,5,2,2,0x0200
注意	-

### 3.2.2.4.8 添加属性描述符同步

设置指令	AT+SSAPSSYNCADDDESCR=<service_handle>,<property_handle>,<uuid>,<permissions>,<operate_indication>,<type>,<value_len>,<value>
响应	<ul style="list-style-type: none"> <li>• 成功: OK</li> <li>• 失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>• &lt;service_handle&gt;: 服务 handle</li> <li>• &lt;property_handle&gt;: 属性 handle</li> <li>• &lt;uuid&gt;: SSAP 描述符 UUID</li> <li>• &lt;permissions&gt;: 特征权限</li> <li>• &lt;operate_indication&gt;: 操作指示</li> <li>• &lt;type&gt;: 描述符类型</li> <li>• &lt;value_len&gt;: 数据长度</li> <li>• &lt;value&gt;: 数据。</li> </ul>
示例	AT+SSAPSSYNCADDDESCR=1,2,0x3333,5,5,2,2,0x0200
注意	-

### 3.2.2.4.9 服务端向客户端发送通知

设置指令	AT+SSAPSSNDNTFY=<conn_id>,<handle>,<type>,<value_len>,<value>
响应	<ul style="list-style-type: none"> <li>• 成功: OK</li> </ul>

	<ul style="list-style-type: none"> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;conn_id&gt;: 服务 handle</li> <li>&lt;handle&gt;: 属性 handle</li> <li>&lt;type&gt;: SSAP 特征类型 <ul style="list-style-type: none"> <li>0: 特征值</li> <li>1: 属性说明描述符</li> <li>2: 客户端配置描述符</li> <li>3: 服务端配置描述符</li> <li>4: 格式描述符</li> <li>5: 服务管理保留描述符, 0x05–0x1F</li> <li>0xFF: 厂商自定义描述符</li> </ul> </li> <li>&lt;value_len&gt;: 数据长度</li> <li>&lt;value&gt;: 数据</li> </ul>
示例	AT+SSAPSSNDNTFY=0,2,0,2,0x0200
注意	-

#### 3.2.2.4.10 服务端向客户端通过 uuid 发送通知

设置指令	AT+SSAPSNTFYBYUUID=<conn_id>,<uuid>,<start_hdl>,<end_hdl>,<type>,<value_len>,<value>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;conn_id&gt;: 服务 handle</li> <li>&lt;uuid&gt;: 属性 uuid</li> <li>&lt;start_hdl&gt;: 开始句柄</li> <li>&lt;end_hdl&gt;: 结束句柄</li> <li>&lt;type&gt;: SSAP 特征类型 <ul style="list-style-type: none"> <li>0: 特征值</li> <li>1: 属性说明描述符</li> <li>2: 客户端配置描述符</li> </ul> </li> </ul>

	3：服务端配置描述符 4：格式描述符 5：服务管理保留描述符，0x05–0x1F 0xFF：厂商自定义描述符 <ul style="list-style-type: none"> <li>• &lt;value_len&gt;：数据长度</li> <li>• &lt;value&gt;：数据</li> </ul>
示例	AT+SSAPSNTFYBYUUID=0,0x1234,0,0xFFFF,0,2,0x0200
注意	-

### 3.2.2.4.11 服务端发送响应

发送响应原因说明：

```

typedef enum {
    ERRCODE_SSAP_INVALID_PDU = ERRCODE_SLE_SSAP_BASE + 0x01,          /*
    服务端接收的 PDU 无效*/
    ERRCODE_SSAP_PDU_NOT_SUPPORT = ERRCODE_SLE_SSAP_BASE + 0x02,
    /*服务端不支持处理接收的PDU*/
    ERRCODE_SSAP_UNKNOW = ERRCODE_SLE_SSAP_BASE + 0x03,
    /*服务端执行请求时发生未知错误*/
    ERRCODE_SSAP_INVALID_HANDLE = ERRCODE_SLE_SSAP_BASE + 0x04,
    /*请求中的句柄无效*/
    ERRCODE_SSAP_INSUFFICIENT_RESOURCES = ERRCODE_SLE_SSAP_BASE + 0x05,
    /*服务端没有足够资源完成请求*/
    ERRCODE_SSAP_PROHIBIT_READING = ERRCODE_SLE_SSAP_BASE + 0x06,
    /*服务端禁止客户端读取值*/
    ERRCODE_SSAP_PROHIBIT_WRITE = ERRCODE_SLE_SSAP_BASE + 0x07,
    /*服务端禁止客户端写入值*/
    ERRCODE_SSAP_CLIENT_NOT_AUTHENTICATED = ERRCODE_SLE_SSAP_BASE + 0x08,
    /*客户端未经过认证*/
    ERRCODE_SSAP_CLIENT_NOT_AUTHORIZATION = ERRCODE_SLE_SSAP_BASE + 0x09,
    /*客户端未被授权*/
    ERRCODE_SSAP_BEARER_NOT_ENCRYPTED = ERRCODE_SLE_SSAP_BASE + 0x0A,
    /*传输 PDU 的承载未加密*/

```



```

ERRCODE_SSAP_ENTRIES_NOT_FOUND = ERRCODE_SLE_SSAP_BASE + 0x0B,
/*服务端未找到对应条目*/
ERRCODE_SSAP_DATA_NOT_FOUND = ERRCODE_SLE_SSAP_BASE + 0x0C,
/*服务端未找到对应类型数据*/
ERRCODE_SSAP_INCORRECT_DATA_TYPE = ERRCODE_SLE_SSAP_BASE + 0x0D,
/*客户端发送写入数据类型不符的错误*/
ERRCODE_SSAP_INCORRECT_DATA_VALUE = ERRCODE_SLE_SSAP_BASE + 0x0E,
/*客户端发送写入值不符的错误*/
ERRCODE_SSAP_VALUE_OUT_OF_RANGE = ERRCODE_SLE_SSAP_BASE + 0x0F,
/*客户端写入的值超出范围*/
ERRCODE_SSAP_UPPERLAYER_APPLICATION_ERROR_MIN =
ERRCODE_SLE_SSAP_BASE + 0xAF,          /*预留给上层协议定义应用错误*/
ERRCODE_SSAP_UPPERLAYER_APPLICATION_ERROR_MAX =
ERRCODE_SLE_SSAP_BASE + 0xFF,          /*预留给上层协议定义应用错误*/
} errcode_sle_ssap_t;

```

设置指令	AT+SSAPSSNDRESP=<conn_id>,<request_id>,<status>,<value_len>,<value>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;conn_id&gt;: 服务 handle</li> <li>&lt;request_id&gt;: 请求 id</li> <li>&lt;status&gt;: 发送响应原因</li> <li>&lt;value_len&gt;: 数据长度</li> <li>&lt;value&gt;: 数据</li> </ul>
示例	AT+SSAPSSNDRESP=0,0,1,2,0x0200
注意	-

### 3.2.2.4.12 服务端注册回调

设置指令	AT+SSAPSREGCBK
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>

参数说明	-
示例	AT+SSAPSREGCBK
注意	-

### 3.2.2.4.13 start service

设置指令	AT+SSAPSSTARTSERV=<service_handle>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<service_handle>: 服务 handle
示例	AT+SSAPSSTARTSERV=1
注意	-

## 3.2.2.5 ssap client 模块指令描述

### 3.2.2.5.1 注册 SSAPC 回调函数

设置指令	AT+SSAPCREGCBK
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	-
示例	AT+SSAPCREGCBK
注意	-

### 3.2.2.5.2 发现 service

查找类型字段说明:

```
typedef enum {
    SSAP_FIND_TYPE_SERVICE_STRUCTURE = 0x00,    /*服务结构*/
    SSAP_FIND_TYPE_PRIMARY_SERVICE    = 0x01,    /*首要服务*/
}
```

```
SSAP_FIND_TYPE_REFERENCE_SERVICE = 0x02,    /*引用服务*/
SSAP_FIND_TYPE_PROPERTY           = 0x03,    /*属性*/
SSAP_FIND_TYPE_METHOD             = 0x04,    /*方法*/
SSAP_FIND_TYPE_EVENT              = 0x05,    /*事件*/
} ssap_find_type_t;
```

设置指令	AT+SSAPCFNDSTRU=<client_id>,<conn_id>,<type>,<uuid>,<start_hdl>,<end_hdl>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;client_id&gt;: 客户端 id</li> <li>&lt;conn_id&gt;: 连接 id</li> <li>&lt;type&gt;: 查找类型 <ul style="list-style-type: none"> <li>0: 服务结构</li> <li>1: 首要服务</li> <li>3: 属性</li> </ul> </li> </ul>
示例	AT+SSAPCFNDSTRU=0,0,1,0x1234,0,0xff
注意	-

### 3.2.2.5.3 客户端向服务端写入数据

设置指令	AT+SSAPCWRITECMD=<client_id>,<conn_id>,<handle>,<type>,<len>,<write_data>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;client_id&gt;: 客户端 id</li> <li>&lt;conn_id&gt;: 连接 id</li> <li>&lt;handle&gt;: 属性 handle</li> <li>&lt;type&gt;: 客户端类型, 取值: 0/1/3</li> <li>&lt;len&gt;: 写入数据长度</li> <li>&lt;write_data&gt;: 写入数据段</li> </ul>

示例	AT+SSAPCWRITECMD=0,0,2,0,2,0x8899
注意	-

#### 3.2.2.5.4 客户端向服务端发送写请求

设置指令	AT+SSAPCWITEREQ=<client_id>,<conn_id>,<handle>,<type>,<len>,<write_data>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;client_id&gt;: 客户端 id</li> <li>&lt;conn_id&gt;: 连接 id</li> <li>&lt;handle&gt;: 属性 handle</li> <li>&lt;type&gt;: 客户端类型, 取值: 0/1/3</li> <li>&lt;len&gt;: 写入数据长度</li> <li>&lt;write_data&gt;: 写入数据段</li> </ul>
示例	AT+SSAPCWITEREQ=0,0,2,0,2,0x8899
注意	-

#### 3.2.2.5.5 客户端发起信息交换

设置指令	AT+SSAPCEXCHINFO=<client_id>,<conn_id>,<mtu_size>,<version>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;client_id&gt;: 客户端 id</li> <li>&lt;conn_id&gt;: 连接 id</li> <li>&lt;mtu_size&gt;: ssap 通道 mtu 最小值: 251 最大值: 520</li> <li>&lt;version&gt;: 版本号</li> </ul>

	0: BLE 1: SLE
示例	AT+SSAPCEXCHINFO=0,0,251,1
注意	-

### 3.2.2.5.6 客户端通过 uuid 发送读请求

设置指令	AT+SSAPCREADBYUUID=<client_id>,<conn_id>,<uuid>,<type>,<start_hdl>,<end_hdl>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;client_id&gt;: 客户端 id</li> <li>&lt;conn_id&gt;: 连接 id</li> <li>&lt;uuid&gt;: uuid</li> <li>&lt;type&gt;: 客户端类型, 取值: 0/1/3</li> <li>&lt;start_hdl&gt;: 开始 handle</li> <li>&lt;end_hdl&gt;: 结束 handle</li> </ul>
示例	AT+SSAPCREADBYUUID=0,0,0x1234,0,0,0xFFFF
注意	-

### 3.2.2.5.7 客户端读取服务端属性数据

设置指令	AT+SSAPCREADREQ=<client_id>,<conn_id>,<handle>,<type>
响应	<ul style="list-style-type: none"> <li>成功: OK</li> <li>失败: ERROR</li> </ul>
参数说明	<ul style="list-style-type: none"> <li>&lt;client_id&gt;: 客户端 id (预留参数)</li> <li>&lt;conn_id&gt;: 连接 id</li> <li>&lt;handle&gt;: 属性 handle (连接成功后的回调里会打印)</li> <li>&lt;type&gt;: 客户端类型, 取值: 0/1/3</li> </ul>

示例	AT+SSAPCREADREQ=0,0,2,0
注意	读数据时的 handle 需与写入数据时的 handle 一致

表3-1 设备外观值一览表

设备外观	值	说明
BLE_APPEARANCE_UNKNOWN	0	None
BLE_APPEARANCE_GENERIC_PHONE	64	Generic category
BLE_APPEARANCE_GENERIC_COMPUTER	128	Generic category
BLE_APPEARANCE_GENERIC_WATCH	192	Generic category
BLE_APPEARANCE_WATCH_SPORTS_WATCH	193	Watch subtype
BLE_APPEARANCE_GENERIC_CLOCK	256	Generic category
BLE_APPEARANCE_GENERIC_DISPLAY	320	Generic category
BLE_APPEARANCE_GENERIC_REMOTE_CONTROL	384	Generic category
BLE_APPEARANCE_GENERIC_EYE_GLASSES	448	Generic category
BLE_APPEARANCE_GENERIC_TAG	512	Generic category
BLE_APPEARANCE_GENERIC_KEYRING	576	Generic category
BLE_APPEARANCE_GENERIC_MEDIA_PLAYER	640	Generic category
BLE_APPEARANCE_GENERIC_BARCODE_SCANNER	704	Generic category
BLE_APPEARANCE_GENERIC_THERMOMETER	768	Generic category
BLE_APPEARANCE_THERMOMETER_EAR	769	Thermometer subtype
BLE_APPEARANCE_GENERIC_HEART_RATE_SENSOR	832	Generic category
BLE_APPEARANCE_HEART_RATE_SENSOR	833	Heart Rate Sensor

设备外观	值	说明
ENSOR_HEART_RATE_BELT		subtype
BLE_APPEARANCE_GENERIC_BLOOD_PRESSURE	896	Generic category
BLE_APPEARANCE_BLOOD_PRESSURE_ARM	897	Blood Pressure subtype
BLE_APPEARANCE_BLOOD_PRESSURE_WRIST	898	Blood Pressure subtype
BLE_APPEARANCE_HUMAN_INTERFACE_DEVICE_HID	960	HID Generic
BLE_APPEARANCE_KEYBOARD	961	HID subtype
BLE_APPEARANCE_MOUSE	962	HID subtype
BLE_APPEARANCE_JOYSTICK	963	HID subtype
BLE_APPEARANCE_GAMEPAD	964	HID subtype
BLE_APPEARANCE_DIGITIZER_TABLET	965	HID subtype
BLE_APPEARANCE_CARD_READER	966	HID subtype
BLE_APPEARANCE_DIGITAL_PEN	967	HID subtype
BLE_APPEARANCE_BARCODE_SCANNER	968	HID subtype
BLE_APPEARANCE_GENERIC_GLUCOSE_METER	1024	Generic category
BLE_APPEARANCE_GENERIC_RUNNING_WALKING_SENSOR	1088	Generic category
BLE_APPEARANCE_RUNNING_WALKING_SENSOR_IN_SHOE	1089	Running Walking Sensor subtype
BLE_APPEARANCE_RUNNING_WALKING_SENSOR_ON_SHOE	1090	Running Walking Sensor subtype
BLE_APPEARANCE_RUNNING_WALKING_SENSOR_ON_HIP	1091	Running Walking Sensor subtype
BLE_APPEARANCE_GENERIC_CYCLING	1152	Generic category
BLE_APPEARANCE_CYCLING_CYCLING_COMPUTER	1153	Cycling subtype
BLE_APPEARANCE_CYCLING_SPEED_SENSOR	1154	Cycling subtype
BLE_APPEARANCE_CYCLING_CADENCE_SENSOR	1155	Cycling subtype

设备外观	值	说明
BLE_APPEARANCE_CYCLING_POWER_SENSOR	1156	Cycling subtype
BLE_APPEARANCE_CYCLING_SPEED_AND_CADENCE_SENSOR	1157	Cycling subtype
BLE_APPEARANCE_INVALID	65536	out of uint16_t



