



# XR806 BLE 控制台命令 使用指南

版本号：1.0

发布时间：2021-02-05

# 版本历史

| 版本  | 日期         | 责任人      | 版本描述  |
|-----|------------|----------|-------|
| 1.0 | 2021-02-05 | AWA 1451 | 创建文档。 |



# 目录

- 版本历史..... i
- 目录..... ii
- 图片目录..... vi
- 表格目录..... vii
- 1 前言..... 1
  - 1.1 文档简介..... 1
  - 1.2 目标读者..... 1
  - 1.3 适用范围..... 1
  - 1.4 文档约定..... 1
    - 1.4.1 标志说明..... 1
- 2 概述..... 2
  - 2.1 背景说明..... 2
  - 2.2 规格特性..... 2
  - 2.3 文件位置..... 2
- 3 应用说明..... 3
  - 3.1 应用简述..... 3
  - 3.2 配置说明..... 3
  - 3.3 格式说明..... 3
  - 3.4 命令说明..... 4
    - 3.4.1 通用配置命令..... 4
      - 3.4.1.1 init（BLE 模块初始化）..... 4
      - 3.4.1.2 deinit（BLE 模块反初始化）..... 4
    - 3.4.2 设备管理命令..... 5
      - 3.4.2.1 name（显示/命名 BLE 设备名称）..... 5
      - 3.4.2.2 id-create（创建 BLE 地址）..... 5
      - 3.4.2.3 id-reset（重置 BLE 地址）..... 5
      - 3.4.2.4 id-delete（删除 BLE 地址）..... 6
      - 3.4.2.5 id-show（显示 BLE 地址信息）..... 6
      - 3.4.2.6 id-select <id>（选择 BLE 地址）..... 6
    - 3.4.3 扫描命令..... 7
      - 3.4.3.1 scan（BLE 扫描）..... 7

|   |    |
|---|----|
| 3.4.4 广播命令.....                                     | 7  |
| 3.4.4.1 advertise (BLE 广播) .....                    | 7  |
| 3.4.4.2 directed-adv (BLE 定向广播) .....               | 8  |
| 3.4.5 链路管理命令.....                                   | 9  |
| 3.4.5.1 connect (BLE 连接) .....                      | 9  |
| 3.4.5.2 auto-conn (BLE 自动连接) .....                  | 9  |
| 3.4.5.3 disconnect (BLE 断开连接) .....                 | 10 |
| 3.4.5.4 select (选择默认连接链路) .....                     | 10 |
| 3.4.5.5 info (显示连接链路信息) .....                       | 11 |
| 3.4.5.6 conn-update (连接链路信息更新) .....                | 11 |
| 3.4.5.7 data-len-update (数据长度信息更新) .....            | 11 |
| 3.4.5.8 phy-update (PHY 配置更新) .....                 | 12 |
| 3.4.5.9 channel-map (更新跳频信道表) .....                 | 12 |
| 3.4.6 HCI 命令.....                                   | 12 |
| 3.4.6.1 hci-cmd (组装 HCI 命令发送) .....                 | 12 |
| 3.4.7 安全相关命令.....                                   | 13 |
| 3.4.7.1 oob (设置 OOB 信息) .....                       | 13 |
| 3.4.7.2 clear (清除链路配对信息) .....                      | 13 |
| 3.4.7.3 security (设置安全等级并启动进行配对绑定) .....            | 13 |
| 3.4.7.4 bondable (设置绑定使能) .....                     | 14 |
| 3.4.7.5 bonds (获取 BLE 绑定信息) .....                   | 14 |
| 3.4.7.6 connections (查询当前已连接链路) .....               | 15 |
| 3.4.7.7 auth (BLE 配对绑定方式配置) .....                   | 15 |
| 3.4.7.8 auth-cancel (取消正在进行的 BLE 配对绑定流程) .....      | 15 |
| 3.4.7.9 auth-passkey (设置配对 passkey) .....           | 16 |
| 3.4.7.10 auth-passkey-confirm (配对 passkey 确认) ..... | 16 |
| 3.4.7.11 auth-pairing-confirm (配对确认) .....          | 16 |
| 3.4.7.12 auth-oob-tk (设置 OOB 方式下的临时密钥) .....        | 16 |
| 3.4.7.13 oob-remote (设置 OOB 参数) .....               | 17 |
| 3.4.7.14 oob-clear (清除 OOB 参数) .....                | 17 |
| 3.4.7.15 fixed-passkey (修改本地 passkey) .....         | 17 |
| 3.4.7.16 wl-add (添加设备到白名单) .....                    | 18 |
| 3.4.7.17 wl-rem (从白名单移除设备) .....                    | 18 |

|   |    |
|---|----|
| 3.4.7.18 wl-clear (清空白名单)                             | 19 |
| 3.4.7.19 wl-connect (自动连接白名单的设备)                      | 19 |
| 3.4.8 GATT 命令   | 19 |
| 3.4.8.1 discover (查询服务)                               | 19 |
| 3.4.8.2 discover-characteristic (查询特性)                | 20 |
| 3.4.8.3 discover-descriptor (查询描述符)                   | 20 |
| 3.4.8.4 discover-include (查询包含服务)                     | 20 |
| 3.4.8.5 discover-primary (查询首要服务)                     | 21 |
| 3.4.8.6 discover-secondary (查询次要服务)                   | 21 |
| 3.4.8.7 exchange-mtu (MTU 包交互长度设置)                    | 21 |
| 3.4.8.8 att_mtu (获取 MTU 大小)                           | 22 |
| 3.4.8.9 read (读取特定 handle 对应的属性值)                     | 22 |
| 3.4.8.10 read-uuid (读取特定 UUID 的属性值)                   | 22 |
| 3.4.8.11 read-multiple (读取多个 handle 的属性值)             | 22 |
| 3.4.8.12 signed-write (签名写)                           | 23 |
| 3.4.8.13 subscribe (订阅服务)                             | 23 |
| 3.4.8.14 unsubscribe (取消订阅)                           | 23 |
| 3.4.8.15 write (有响应写命令)                               | 24 |
| 3.4.8.16 write-without-response (无响应写命令)              | 24 |
| 3.4.8.17 write-without-response-cb (使用特定写完成回调的无响应写命令) | 25 |
| 3.4.8.18 get (遍历范围内 ATTR)                             | 25 |
| 3.4.8.19 set (设置某个 ATTR)                              | 25 |
| 3.4.8.20 show-db (显示所有 ATTR)                          | 26 |
| 3.4.8.21 metrics (注册 metrics 服务)                      | 26 |
| 3.4.8.22 register (注册测试服务)                            | 26 |
| 3.4.8.23 unregister (关闭测试服务)                          | 26 |
| 3.4.8.24 notify (notify 通知)                           | 27 |
| 3.4.8.25 indicate (indicate 通知)                       | 27 |
| 4 示例说明  | 28 |
| 4.1 GATT 读写示例 (Slave Role)                            | 28 |
| 4.1.1 示例简介  | 28 |
| 4.1.1.1 获取方式  | 28 |
| 4.1.1.2 准备工作  | 28 |

|                                   |    |
|-----------------------------------|----|
| 4.1.1.3 操作步骤.....                 | 28 |
| 4.1.2 效果展示.....                   | 28 |
| 4.2 配对绑定示例 (Slave Role) .....     | 33 |
| 4.2.1 示例简介.....                   | 33 |
| 4.2.1.1 获取方式.....                 | 33 |
| 4.2.1.2 准备工作.....                 | 33 |
| 4.2.1.3 操作步骤.....                 | 33 |
| 4.2.2 效果展示.....                   | 33 |
| 4.3 GATT 读写示例 (Master Role) ..... | 34 |
| 4.3.1 示例简介.....                   | 34 |
| 4.3.1.1 获取方式.....                 | 34 |
| 4.3.1.2 准备工作.....                 | 34 |
| 4.3.1.3 操作步骤.....                 | 35 |
| 4.3.2 效果展示.....                   | 35 |
| 4.4 配对绑定示例 (Master Role) .....    | 37 |
| 4.4.1 示例简介.....                   | 37 |
| 4.4.1.1 获取方式.....                 | 37 |
| 4.4.1.2 准备工作.....                 | 37 |
| 4.4.1.3 操作步骤.....                 | 37 |
| 4.4.2 效果展示.....                   | 37 |

# 图片目录

|       |                       |    |
|-------|-----------------------|----|
| 图 4-1 | nRF Connect 扫描.....   | 30 |
| 图 4-2 | nRF Connect 连接界面..... | 31 |
| 图 4-3 | BLE GATT Service..... | 32 |
| 图 4-4 | BLE GATT 读写过程.....    | 32 |
| 图 4-5 | BLE 配对绑定过程.....       | 34 |



# 表格目录

|       |                  |   |
|-------|------------------|---|
| 表 2-1 | BLE 命令的文件位置..... | 2 |
| 表 3-1 | BLE 命令配置列表.....  | 3 |
| 表 3-2 | BLE 命令语法定义.....  | 3 |
| 表 3-3 | BLE 命令格式说明.....  | 3 |
| 表 3-4 | BLE 命令简要说明.....  | 4 |



# 1 前言

## 1.1 文档简介

本文档介绍了 XR806 SDK 的 BLE 命令的用法及使用示例，以帮助用户使用 BLE 命令快速上手 BLE 功能开发。

## 1.2 目标读者

XR806 BLE 功能测试、开发相关人员。

## 1.3 适用范围

此文档适用于 XR806 SDK，支持 XR806 系列芯片产品。

## 1.4 文档约定

### 1.4.1 标志说明

本文档采用各种醒目的标志来表示在操作过程中应该特别注意的地方，这些标志的含义如下：

| 标识  | 说明  |
|---|---|
|  <b>警告</b> | 该标志后的说明应给予格外关注，如果不遵守，可能会导致人员受伤或死亡。        |
|  <b>注意</b> | 提醒操作中应注意的事项。不当的操作可能会损坏器件，影响可靠性、降低性能等。     |
|  <b>说明</b> | 为准确理解文中指令、正确实施操作而提供的补充或强调信息。              |
|  <b>窍门</b> | 一些容易忽视的小功能、技巧。了解这些功能或技巧能帮助解决特定问题或者节省操作时间。 |

## 2 概述

### 2.1 背景说明

为了便于 BLE 功能的测试、开发，XR806 SDK 集成了 Zephyr 项目的 Bluetooth 命令。SDK 中的 BLE 命令主要封装在 cmd\_ble.c、cmd\_gatt.c 文件中，覆盖了 BLE 常用的功能场景。

### 2.2 规格特性

XR806 SDK 的 Bluetooth 命令支持扫描、广播、连接、配对绑定等常用场景的命令功能，具体命令的使用请参考“命令说明”小节。

### 2.3 文件位置

以 SDK 包为根目录，BLE 命令涉及到的主要文件位置如下。

表 2-1 BLE 命令的文件位置

| 组件名    | 文件分类 | 文件位置                               |
|--------|------|------------------------------------|
| BLE 命令 | 源码文件 | ./project/common/cmd/cmd_ble.c     |
|        | 源码文件 | ./project/common/cmd/cmd_gatt.c    |
|        | 示例工程 | ./project/demo/bluetooth/ble_demo/ |

## 3 应用说明

### 3.1 应用简述

BLE 命令已经集成到 XR806 SDK，下面将对 BLE 命令的使用进行介绍。

### 3.2 配置说明

使用 BLE 命令功能需要先将工程配置中的 BLE 功能项打开。

表 3-1 BLE 命令配置列表

| 配置项      | 配置说明  |
|----------|---|
| BLE 功能使能 | <p>设置说明：<br/>此项配置用于在 SDK 中启用 BLE 功能。</p> <p>设置方式：<br/>在控制台执行命令“make menuconfig”，将以下功能项打开：<br/>[*] BLE Controller<br/>[*] Ble Host</p> <p>建议参考 ble_demo 的工程配置 defconfig。</p> |

### 3.3 格式说明

BLE 命令的语约定如下表。

表 3-2 BLE 命令语约定

| 符号 | 说明    |
|----|-------|
| =  | 参数引导符 |
| ,  | 参数分隔符 |
| <> | 强制使用  |
| [] | 选择使用  |

在上述语约定下，BLE 命令有以下格式定义。

表 3-3 BLE 命令格式说明

| 分类   | 格式                                 | 说明                               |
|------|------------------------------------|----------------------------------|
| 命令格式 | ble <CMD> [[p1],[p2],[p3[=]],...]  | ble/gatt：命令前缀，可改变                |
|      | gatt <CMD> [[p1],[p2],[p3[=]],...] | <CMD>：命令字符<br>[p1]/[p2]/[p3]：参数。 |

| 分类   | 格式  | 说明  |
|------|---|---|
| 响应格式 | <ACK> <status-code> <respond-description> | <ACK>：命令响应前缀，固定。<br><status-code>：状态码，为 200 时表示执行成功，为 4xx 时表示执行出错。<br><respond-description>：响应消息。 |

### 3.4 命令说明

BLE 命令提供了扫描、广播、链路管理等命令，命令的简要说明如下表所示。

表 3-4 BLE 命令简要说明

| 命令     | 简要说明   |
|--------|--|
| 通用配置命令 | 主要包含了 init、deinit 两个命令，使用 BLE 命令必须先执行 init 命令进行 BLE 初始化。 |
| 设备管理命令 | 主要对 BLE 的设备名称、地址等进行配置。                                   |
| 扫描命令   | 提供 BLE 扫描功能，包括主动和被动扫描。                                   |
| 广播命令   | 提供 BLE 广播功能。   |
| 链路管理命令 | 提供了链路连接、更新等功能。   |
| HCI 命令 | 提供组装 HCI 命令功能。   |
| 安全相关命令 | 提供配对绑定等功能。   |

#### 3.4.1 通用配置命令

##### 3.4.1.1 init (BLE模块初始化)

| 信息项 | 说明   |
|-----|--|
| 原型  | ble init                                     |
| 功能  | BLE 模块初始化                                    |
| 参数  | N/A  |
| 响应  | <ACK> 200 OK                                 |
| 示例  | N/A  |
| 备注  | 若系统启动后没有进行 BLE 初始化，执行其他 BLE 命令前都需要先执行该初始化命令。 |

##### 3.4.1.2 deinit (BLE模块反初始化)

| 信息项 | 说明         |
|-----|------------|
| 原型  | ble deinit |
| 功能  | BLE 模块反初始化 |

| 信息项 | 说明                           |
|-----|------------------------------|
| 参数  | N/A                          |
| 响应  | <ACK> 200 OK                 |
| 示例  | N/A                          |
| 备注  | 该命令只在使能 CONFIG_BT_DEINIT 时有效 |

### 3.4.2 设备管理命令

#### 3.4.2.1 name (显示/命名BLE设备名称)

| 信息项 | 说明  |
|-----|---|
| 原型  | ble name [name]                               |
| 功能  | 显示/命名 BLE 名称                                  |
| 参数  | [name]: 要设置的 BLE 设备名称                         |
| 响应  | <ACK> 200 OK                                  |
| 示例  | 显示 BLE 名称: ble name                           |
| 备注  | 有[name]参数时为设置 BLE 的设备名称, 无参数时为显示当前 BLE 的设备名称。 |

#### 3.4.2.2 id-create (创建BLE地址)

| 信息项 | 说明  |
|-----|---|
| 原型  | ble id-create [addr: XX:XX:XX:XX:XX:XX]   |
| 功能  | 创建 BLE 地址   |
| 参数  | [addr]: 需要创建的 BLE 私有地址, 格式是 XX:XX:XX:XX:XX:XX   |
| 响应  | <ACK> 200 OK  |
| 示例  | ble id-create   |
| 备注  | 视系统设置允许创建多少个 BLE 私有地址, 当前最多只允许 1 个私有地址存在, 地址默认为 random 类型, 当前基本不使用。可在 ble init 之前调用, 但不会存储到 flash 之中。 |

#### 3.4.2.3 id-reset (重置BLE地址)

| 信息项 | 说明  |
|-----|---|
| 原型  | ble id-reset <id> [addr: XX:XX:XX:XX:XX:XX]   |
| 功能  | 重置 BLE 地址   |
| 参数  | <id>: 第几个 BLE 私有地址, 当前默认只有 1 个, 10 进制数<br>[addr]: 需要重置的 BLE 私有地址, 格式是 XX:XX:XX:XX:XX:XX |
| 响应  | <ACK> 200 OK  |

| 信息项 | 说明   |
|-----|--|
| 示例  | N/A  |
| 备注  | 视系统设置允许创建多少个 BLE 私有地址，当前最多只允许 1 个私有地址存在，地址默认为 random 类型，当前基本不使用。 |

#### 3.4.2.4 id-delete (删除BLE地址)

| 信息项 | 说明                                   |
|-----|--------------------------------------|
| 原型  | ble id-delete <id>                   |
| 功能  | 删除 BLE 地址                            |
| 参数  | <id>: 第几个 BLE 私有地址，当前默认只有 1 个，10 进制数 |
| 响应  | <ACK> 200 OK                         |
| 示例  | 删除 BLE 第 2 个私有地址：ble id-delete 1     |
| 备注  | 默认只有 1 个 BLE 私有地址，所以无法删除，当前无法正常使用。   |

#### 3.4.2.5 id-show (显示BLE地址信息)

| 信息项 | 说明                     |
|-----|------------------------|
| 原型  | ble id-show            |
| 功能  | 显示 BLE 地址信息            |
| 参数  | N/A                    |
| 响应  | <ACK> 200 OK           |
| 示例  | 显示当前私有地址信息：ble id-show |
| 备注  | N/A                    |

#### 3.4.2.6 id-select <id> (选择BLE地址)

| 信息项 | 说明                                   |
|-----|--------------------------------------|
| 原型  | ble id-select <id>                   |
| 功能  | 选择 BLE 地址                            |
| 参数  | <id>: 第几个 BLE 私有地址，当前默认只有 1 个，10 进制数 |
| 响应  | <ACK> 200 OK                         |
| 示例  | 当前选择使用第 2 个 BLE 地址：ble id-select 1   |
| 备注  | N/A                                  |

### 3.4.3 扫描命令

#### 3.4.3.1 scan (BLE扫描)

| 信息项 | 说明  |
|-----|---|
| 原型  | ble scan <value: on, passive, off> [dup: dups, nodups] [wl] [int=0x<hex>] [win=0x<hex>]   |
| 功能  | BLE 扫描  |
| 参数  | <p>&lt;value&gt;: 扫描的状态设置, 可取值:<br/>                     on: 主动扫描<br/>                     passive: 被动扫描<br/>                     off: 停止扫描</p> <p>[dup]: host 是否收重复包, 可取值:<br/>                     dups: host 收重复包<br/>                     nodups: host 不收重复包</p> <p>[wl]: 启用白名单</p> <p>[int=0x&lt;hex&gt;]: 扫描间隔。默认值为 0x60, 对应时间为 0x60*625us=60ms。<br/>                     [win=0x&lt;hex&gt;]: 扫描窗口。默认值为 0x30, 对应时间为 0x30*625us=30ms。</p> |
| 响应  | <ACK> 200 OK  |
| 示例  | BLE 开启扫描: ble scan on<br>BLE 停止扫描: ble scan off   |
| 备注  | 该命令只在使能 CONFIG_BT_OBSERVER 时有效; int 和 win 输入为 16 进制数。   |

### 3.4.4 广播命令

#### 3.4.4.1 advertise (BLE广播)

| 信息项 | 说明  |
|-----|---|
| 原型  | ble advertise <type: off, on, scan, nconn> [mode: discov, non_discov] [whitelist: wl, wl-scan, wl-conn] [identity] [no-name] [1shot/one-time] [nconn] [min=0x<hex>] [max=0x<hex>]   |
| 功能  | BLE 广播  |
| 参数  | <p>&lt;type&gt;: 广播状态选择<br/>                     off: 关闭广播<br/>                     on: 打开广播<br/>                     scan: 发送有名称, 不可连接广播包<br/>                     nconn: 发送无名称, 不可连接广播包</p> <p>[mode]: 广播模式配置<br/>                     discov: 可被扫描发现</p> |

| 信息项 | 说明   |
|-----|--|
|     | <p>non_discov: 不可被扫描发现</p> <p>[whitelist]: 白名单配置</p> <p>wl: 使用白名单过滤可请求 scan rsp 和可连接的设备</p> <p>wl-scan: 使用白名单过滤可请求 scan rsp 的设备</p> <p>wl-conn: 使用白名单过滤可连接的设备</p> <p>[identity]: 使用私有身份地址作为设备地址</p> <p>[no-name]: 广播包或 scan rsp 中不包含有 name 字段</p> <p>[1shot/one-time]: 置起后 Peripheral 被连接上后不发广播包</p> <p>[nconn]: 不可被连接</p> <p>[min=0x&lt;hex&gt;]: 广播事件最小间隔。默认值为 0xa0, 对应时间为 0xa0*625us=100ms。</p> <p>[max=0x&lt;hex&gt;]: 广播事件最大间隔。默认值为 0xf0, 对应时间为 0xf0*625us=150ms。</p> |
| 响应  | <ACK> 200 OK   |
| 示例  | <p>打开广播: ble advertise on</p> <p>关闭广播: ble advertise off</p>   |
| 备注  | 该命令需要使能 CONFIG_BT_OBSERVER 时有效; min 和 max 输入为 16 进制数。  |

### 3.4.4.2 directed-adv (BLE定向广播)

| 信息项 | 说明   |
|-----|--|
| 原型  | ble directed-adv <address: XX:XX:XX:XX:XX:XX> <type: (public random)> [mode: low] [identity] [dir-rpa]   |
| 功能  | BLE 定向广播   |
| 参数  | <p>&lt;address&gt;: 定向广播地址, 格式是 XX:XX:XX:XX:XX:XX</p> <p>&lt;type&gt;: 广播地址类型</p> <p>public: 公共地址</p> <p>random: 随机地址</p> <p>[mode]: 广播模式</p> <p>low: 正常定向广播</p> <p>[identity]: 使用私有身份地址作为设备地址</p> <p>[dir-rpa]: 使能目标设备为可解析私有地址的定向广播</p> |
| 响应  | <ACK> 200 OK   |

| 信息项 | 说明  |
|-----|---|
| 示例  | ble directed-adv 78:17:FB:8E:CE:19 random low dir-rpa |
| 备注  | 该命令需要使能 CONFIG_BT_PERIPHERAL 时有效                      |

### 3.4.5 链路管理命令

#### 3.4.5.1 connect (BLE连接)

| 信息项 | 说明  |
|-----|---|
| 原型  | ble connect <address: XX:XX:XX:XX:XX:XX> <type: (public random)> [lat=0x<hex>] [to=0x<hex>] [min=0x<hex>] [max=0x<hex>]   |
| 功能  | BLE 连接  |
| 参数  | <p>&lt;address&gt;: 需要连接设备的地址, 格式是 XX:XX:XX:XX:XX:XX</p> <p>&lt;type&gt;: 设备地址类型<br/>                     public: 公共地址<br/>                     random: 随机地址</p> <p>[lat=0x&lt;hex&gt;]: 从机时延(默认值为 0, 即每个 master 的连接事件 slave 都要接收)</p> <p>[to=0x&lt;hex&gt;]: 连接超时(默认值为 400, 对应时间为 400*10ms=4s)</p> <p>[min=0x&lt;hex&gt;]: 连接间隔最小值(默认值为 0x18, 对应时间为 0x18*1.25ms=30ms)</p> <p>[max=0x&lt;hex&gt;]: 连接间隔最大值(默认值为 0x28, 对应时间为 0x28*1.25ms=50ms)</p> |
| 响应  | <ACK> 200 OK  |
| 示例  | ble connect 5B:81:D0:0C:A3:DB random  |
| 备注  | 该命令需要使能 CONFIG_BT_CENTRAL 时有效;<br>连接间隔设置一个最小值和最大值, 最终由 Controller 自行决定使用什么值。  |

#### 3.4.5.2 auto-conn (BLE自动连接)

| 信息项 | 说明  |
|-----|---|
| 原型  | ble auto-conn <address: XX:XX:XX:XX:XX:XX> <type: (public random)> <value:on,off>   |
| 功能  | BLE 自动连接  |
| 参数  | <p>&lt;address&gt;: 需要连接设备的地址, 格式是 XX:XX:XX:XX:XX:XX</p> <p>&lt;type&gt;: 设备地址类型<br/>                     public: 公共地址<br/>                     random: 随机地址</p> <p>&lt;value&gt;: 状态配置<br/>                     on: 自动连接开启</p> |

| 信息项 | 说明  |
|-----|---|
|     | off：自动连接关闭  |
| 响应  | <ACK> 200 OK  |
| 示例  | ble auto-conn 64:99:4E:F9:AB:59 random on   |
| 备注  | 与白名单功能相斥，无法同时打开使用（即需要关闭宏 CONFIG_BT_WHITELIST 该命令才有效）；与普通连接区别在于即使扫描不到设备，也不会停止连接，只能通过命令关闭，而普通连接命令在一定时间内连接不上设备会显示失败。 |

### 3.4.5.3 disconnect (BLE断开连接)

| 信息项 | 说明   |
|-----|--|
| 原型  | ble disconnect [<address: XX:XX:XX:XX:XX:XX> <type: (public random)>]                    |
| 功能  | 断开 BLE 连接  |
| 参数  | <address>：需要断开设备的地址，格式是 XX:XX:XX:XX:XX:XX<br><type>：设备地址类型<br>public：公共地址<br>random：随机地址 |
| 响应  | <ACK> 200 OK   |
| 示例  | ble disconnect   |
| 备注  | 该命令若带参数则表示断开指定的连接，否则断开当前的连接  |

### 3.4.5.4 select (选择默认连接链路)

| 信息项 | 说明   |
|-----|--|
| 原型  | ble select <address: XX:XX:XX:XX:XX:XX> <type: (public random)>                        |
| 功能  | 选择默认连接链路   |
| 参数  | <address>：连接设备的地址，格式是 XX:XX:XX:XX:XX:XX<br><type>：设备地址类型<br>public：公共地址<br>random：随机地址 |
| 响应  | <ACK> 200 OK   |
| 示例  | ble select 64:99:4E:F9:AB:59 random  |
| 备注  | 系统支持多链路下使用，选择当前哪条链路为默认连接链路   |

### 3.4.5.5 info (显示连接链路信息)

| 信息项 | 说明  |
|-----|---|
| 原型  | ble info <address: XX:XX:XX:XX:XX:XX> <type: (public random)>                                   |
| 功能  | 显示连接链路信息  |
| 参数  | <address>: 需要查询连接设备的地址, 格式是 XX:XX:XX:XX:XX:XX<br><type>: 设备地址类型<br>public: 公共地址<br>random: 随机地址 |
| 响应  | <ACK> 200 OK  |
| 示例  | ble info  |
| 备注  | 默认显示当前链路信息, 主要打印连接双方地址及在空气中的地址, 以及 interval、latency 和 timeout 参数, 需在与设备连接后使用。                   |

### 3.4.5.6 conn-update (连接链路信息更新)

| 信息项 | 说明   |
|-----|--|
| 原型  | ble conn-update <min> <max> <latency> <timeout>                        |
| 功能  | 连接链路信息更新   |
| 参数  | <min>: 连接间隔最小值<br><max>: 连接间隔最大值<br><latency>: 从机时延<br><timeout>: 连接超时 |
| 响应  | <ACK> 200 OK   |
| 示例  | 更新链路参数: ble conn-update 48 80 1 500                                    |
| 备注  | 四个参数都是 16 进制格式输入。  |

### 3.4.5.7 data-len-update (数据长度信息更新)

| 信息项 | 说明  |
|-----|---|
| 原型  | ble data-len-update <tx_max_len> [tx_max_time]        |
| 功能  | 数据长度信息更新  |
| 参数  | <tx_max_len>: 最大数据荷载传输长度<br>[tx_max_time]: 最大数据荷载传输时间 |
| 响应  | <ACK> 200 OK  |
| 示例  | N/A   |
| 备注  | 两个参数一起影响数据的传输速度。                                      |

### 3.4.5.8 phy-update (PHY配置更新)

| 信息项 | 说明   |
|-----|--|
| 原型  | ble phy-update <tx_phy> [rx_phy] [s2] [s8]   |
| 功能  | PHY 配置更新   |
| 参数  | <p>&lt;tx_phy&gt;: 发送的参考 PHYs 掩码</p> <p>0: 没有选择 PHY, 即不更新 PHY</p> <p>1: 1 Mbps PHY</p> <p>2: 2 Mbps PHY</p> <p>4: CODED PHY</p> <p>[rx_phy]: 目前与 tx_phy 配置一致</p> <p>[s2]: S=2, tx_phy 配置成 4 时有效</p> <p>[s8]: S=8, tx_phy 配置成 4 时有效</p> |
| 响应  | <ACK> 200 OK   |
| 示例  | ble phy-update 1   |
| 备注  | 该命令需要使能 CONFIG_BT_USER_PHY_UPDATE 时有效  |

### 3.4.5.9 channel-map (更新跳频信道表)

| 信息项 | 说明   |
|-----|--|
| 原型  | ble channel-map <channel-map: XXXXXXXXXX> (36-0) |
| 功能  | 更新跳频信道表  |
| 参数  | <channel-map>: 新的跳频信道表, 每个 bit 代表一个信道            |
| 响应  | <ACK> 200 OK                                     |
| 示例  | 更新跳频信道表: ble channel-map FFFFFFFF                |
| 备注  | 每个 bit 代表一个信道, 设置为 0 则表示不使用该信道, 为 1 表示可跳至该信道     |

## 3.4.6 HCI 命令

### 3.4.6.1 hci-cmd (组装HCI命令发送)

| 信息项 | 说明  |
|-----|---|
| 原型  | ble hci-cmd <ogf> <ocf> [data]                    |
| 功能  | 组装 HCI 命令发送                                       |
| 参数  | <ogf>: OpCode 的 Group Fileds (参数使用介绍详见蓝牙核心协议, 下同) |

| 信息项 | 说明   |
|-----|--|
|     | <ocf>: OpCode 的 Command Fileds<br>[data]: 具体的命令数据, 以 16 进制输入 |
| 响应  | <ACK> 200 OK   |
| 示例  | N/A  |
| 备注  | 参数 ogf 和 ocf 输入为十六进制数字                                       |

### 3.4.7 安全相关命令

#### 3.4.7.1 oob (设置OOB信息)

| 信息项 | 说明           |
|-----|--------------|
| 原型  | ble oob      |
| 功能  | 设置 OOB 信息    |
| 参数  | N/A          |
| 响应  | <ACK> 200 OK |
| 示例  | N/A          |
| 备注  | N/A          |

#### 3.4.7.2 clear (清除链路配对信息)

| 信息项 | 说明  |
|-----|---|
| 原型  | ble clear <remote: addr, all>   |
| 功能  | 清除链路配对信息  |
| 参数  | <remote><br>addr: 特定设备信息<br>all: 所有已配对设备                                    |
| 响应  | <ACK> 200 OK  |
| 示例  | 清除某个设备配对信息: ble clear 30:74:96:99:1A:76 public<br>清除全部设备配对信息: ble clear all |
| 备注  | N/A   |

#### 3.4.7.3 security (设置安全等级并启动进行配对绑定)

| 信息项 | 说明  |
|-----|---|
| 原型  | ble security <security level BR/EDR: 0 - 3, LE: 1 - 4> [force-pair] |
| 功能  | 设置安全等级并启动进行配对绑定   |

| 信息项 | 说明  |
|-----|---|
| 参数  | <security level BR/EDR: 0 - 3, LE: 1 - 4>: 目前只支持 LE<br>1: 无加密, 无认证<br>2: 加密不认证<br>3: 加密且认证<br>4: 128bit key 的认证安全连接<br><br>[force-pair]: 强制重新进行配对过程 |
| 响应  | <ACK> 200 OK  |
| 示例  | 设置加密等级: ble security 2  |
| 备注  | 默认加密等级为 1, 如需要设置加密等级为 1, 则需要添加 force-pair 参数进行强制重新配对绑定, 等级 2~4 不需该参数, 但也可携带。  |

#### 3.4.7.4 bondable (设置绑定使能)

| 信息项 | 说明  |
|-----|---|
| 原型  | ble bondable <bondable: on, off>            |
| 功能  | 设置能否绑定                                      |
| 参数  | <bondable>: 本设备是否允许绑定<br>on: 允许<br>off: 不允许 |
| 响应  | <ACK> 200 OK                                |
| 示例  | N/A   |
| 备注  | N/A   |

#### 3.4.7.5 bonds (获取BLE绑定信息)

| 信息项 | 说明           |
|-----|--------------|
| 原型  | ble bonds    |
| 功能  | 获取 BLE 绑定信息  |
| 参数  | N/A          |
| 响应  | <ACK> 200 OK |
| 示例  | N/A          |
| 备注  | N/A          |

**3.4.7.6 connections (查询当前已连接链路)**

| 信息项 | 说明              |
|-----|-----------------|
| 原型  | ble connections |
| 功能  | 查询当前已连接链路       |
| 参数  | N/A             |
| 响应  | <ACK> 200 OK    |
| 示例  | N/A             |
| 备注  | N/A             |

**3.4.7.7 auth (BLE配对绑定方式配置)**

| 信息项 | 说明   |
|-----|--|
| 原型  | ble auth <method: all, input, display, yesno, confirm, oob, none>  |
| 功能  | BLE 配对绑定方式配置   |
| 参数  | <method><br>all: 支持所有认证方式<br>input: 输入认证<br>display: 显示认证<br>yesno: 显示 yes/no<br>confirm: 确认认证<br>oob: 带外方式<br>none: 不使用额外认证 |
| 响应  | <ACK> 200 OK   |
| 示例  | 设置 BLE 配对方式为 confirm: ble auth confirm   |
| 备注  | 一个上电周期内只允许设置一次   |

**3.4.7.8 auth-cancel (取消正在进行的BLE配对绑定流程)**

| 信息项 | 说明                            |
|-----|-------------------------------|
| 原型  | ble auth-cancel               |
| 功能  | 取消正在进行的 BLE 配对绑定流程            |
| 参数  | N/A                           |
| 响应  | <ACK> 200 OK                  |
| 示例  | 清除本设备 BLE 配对: ble auth-cancel |
| 备注  | N/A                           |

**3.4.7.9 auth-passkey (设置配对passkey)**

| 信息项 | 说明                                    |
|-----|---------------------------------------|
| 原型  | ble auth-passkey <passkey>            |
| 功能  | 设置配对 passkey                          |
| 参数  | <passkey>: 配对密钥, 为 6 位数字              |
| 响应  | <ACK> 200 OK                          |
| 示例  | 设置配对 passkey: ble auth-passkey 123456 |
| 备注  | N/A                                   |

**3.4.7.10 auth-passkey-confirm (配对passkey确认)**

| 信息项 | 说明                       |
|-----|--------------------------|
| 原型  | ble auth-passkey-confirm |
| 功能  | 配对 passkey 确认            |
| 参数  | N/A                      |
| 响应  | <ACK> 200 OK             |
| 示例  | N/A                      |
| 备注  | N/A                      |

**3.4.7.11 auth-pairing-confirm (配对确认)**

| 信息项 | 说明                       |
|-----|--------------------------|
| 原型  | ble auth-pairing-confirm |
| 功能  | 配对确认                     |
| 参数  | N/A                      |
| 响应  | <ACK> 200 OK             |
| 示例  | N/A                      |
| 备注  | N/A                      |

**3.4.7.12 auth-oob-tk (设置OOB方式下的临时密钥)**

| 信息项 | 说明                   |
|-----|----------------------|
| 原型  | ble auth-oob-tk <tk> |
| 功能  | 设置 OOB 方式下的临时密钥      |
| 参数  | <tk>: 临时密钥, 16 字节大小  |

| 信息项 | 说明           |
|-----|--------------|
| 响应  | <ACK> 200 OK |
| 示例  | N/A          |
| 备注  | N/A          |

### 3.4.7.13 oob-remote (设置OOB参数)

| 信息项 | 说明   |
|-----|--|
| 原型  | ble oob-remote <address: XX:XX:XX:XX:XX:XX> <type: (public random)> <oob rand> <oob confirm>   |
| 功能  | 设置 OOB 参数  |
| 参数  | <address>: 本机地址<br><type>: 地址类型<br>public: 公共地址<br>random: 随机地址<br><br><oob rand>: OOB 随机数 16 进制, 16*8=128bit<br><oob confirm>: OOB confirm 数据, 16 进制, 16*8=128bit |
| 响应  | <ACK> 200 OK   |
| 示例  | 设置 OOB 参数: ble oob-remot 00:00:3D:06:00:00 public 1234 1234  |
| 备注  | 设置特定设备所有用的 OOB 配对信息参数, 当前未使用   |

### 3.4.7.14 oob-clear (清除OOB参数)

| 信息项 | 说明                       |
|-----|--------------------------|
| 原型  | ble oob-clear            |
| 功能  | 清除 OOB 参数                |
| 参数  | N/A                      |
| 响应  | <ACK> 200 OK             |
| 示例  | 清除 OOB 参数: ble oob-clear |
| 备注  | 清除已设置的 OOB 配对绑定信息        |

### 3.4.7.15 fixed-passkey (修改本地passkey)

| 信息项 | 说明                          |
|-----|-----------------------------|
| 原型  | ble fixed-passkey [passkey] |

| 信息项 | 说明   |
|-----|--|
| 功能  | 修改本地 passkey   |
| 参数  | [passkey]: 认证密钥, 为 6 位数字                             |
| 响应  | <ACK> 200 OK   |
| 示例  | 修改本设备 passkey: ble fixed-passkey 1234                |
| 备注  | 该命令需要使能 CONFIG_BT_FIXED_PASKEY 有效, 会替代随机产生的 passkey。 |

### 3.4.7.16 wl-add (添加设备到白名单)

| 信息项 | 说明  |
|-----|---|
| 原型  | ble wl-add <address: XX:XX:XX:XX:XX:XX> <type: (public random)> |
| 功能  | 添加设备到白名单  |
| 参数  | <address>: 本机地址<br><type>: 地址类型<br>public: 公共地址<br>random: 随机地址 |
| 响应  | <ACK> 200 OK  |
| 示例  | 添加设备到白名单: ble wl-add 00:00:3D:06:00:00 public                   |
| 备注  | 该命令需要使能 CONFIG_BT_WHITELIST 有效                                  |

### 3.4.7.17 wl-rem (从白名单移除设备)

| 信息项 | 说明  |
|-----|---|
| 原型  | ble wl-rem <address: XX:XX:XX:XX:XX:XX> <type: (public random)> |
| 功能  | 从白名单移除设备  |
| 参数  | <address>: 本机地址<br><type>: 地址类型<br>public: 公共地址<br>random: 随机地址 |
| 响应  | <ACK> 200 OK  |
| 示例  | 从白名单中删除某个设备: ble wl-rem 00:00:3D:06:00:00 public                |
| 备注  | 该命令需要使能 CONFIG_BT_WHITELIST 有效                                  |

### 3.4.7.18 wl-clear (清空白名单)

| 信息项 | 说明                             |
|-----|--------------------------------|
| 原型  | ble wl-clear                   |
| 功能  | 清空白名单                          |
| 参数  | N/A                            |
| 响应  | <ACK> 200 OK                   |
| 示例  | 清空白名单：ble wl-clear             |
| 备注  | 该命令需要使能 CONFIG_BT_WHITELIST 有效 |

### 3.4.7.19 wl-connect (自动连接白名单的设备)

| 信息项 | 说明  |
|-----|---|
| 原型  | ble wl-connect <on, off>                                      |
| 功能  | 自动连接白名单的设备  |
| 参数  | <on, off><br>on：启动自动连接白名单设备，每执行该命令自动连接白名单中的一个设备<br>off：停止自动连接 |
| 响应  | <ACK> 200 OK  |
| 示例  | N/A   |
| 备注  | 该命令需要使能 CONFIG_BT_WHITELIST 有效                                |

## 3.4.8 GATT 命令

### 3.4.8.1 discover (查询服务)

| 信息项 | 说明  |
|-----|---|
| 原型  | gatt discover [UUID] [start handle] [end handle]                                |
| 功能  | 查询服务  |
| 参数  | [UUID]：需要查询服务的 UUID；<br>[start handle]：查询起始 handle；<br>[end handle]：查询结束 handle |
| 响应  | <ACK> 200 OK  |
| 示例  | 查询所有服务：gatt discover  |
| 备注  | N/A   |

**3.4.8.2 discover-characteristic (查询特性)**

| 信息项 | 说明   |
|-----|--|
| 原型  | gatt discover-characteristic [UUID] [start handle] [end handle]                    |
| 功能  | 查询特性   |
| 参数  | [UUID]: 需要查询服务的 UUID;<br>[start handle]: 查询起始 handle;<br>[end handle]: 查询结束 handle |
| 响应  | <ACK> 200 OK   |
| 示例  | 查询设备名称特性: gatt discover-characteristic 2A00  |
| 备注  | N/A  |

**3.4.8.3 discover-descriptor (查询描述符)**

| 信息项 | 说明   |
|-----|--|
| 原型  | gatt discover-descriptor [UUID] [start handle] [end handle]                        |
| 功能  | 查询描述符  |
| 参数  | [UUID]: 需要查询服务的 UUID;<br>[start handle]: 查询起始 handle;<br>[end handle]: 查询结束 handle |
| 响应  | <ACK> 200 OK   |
| 示例  | 查询特定描述符: gatt discover-characteristic 2A00 0 1234                                  |
| 备注  | N/A  |

**3.4.8.4 discover-include (查询包含服务)**

| 信息项 | 说明   |
|-----|--|
| 原型  | gatt discover-include [UUID] [start handle] [end handle]                           |
| 功能  | 查询包含服务   |
| 参数  | [UUID]: 需要查询服务的 UUID;<br>[start handle]: 查询起始 handle;<br>[end handle]: 查询结束 handle |
| 响应  | <ACK> 200 OK   |
| 示例  | 查询所有包含服务: gatt discover-include  |
| 备注  | N/A  |

**3.4.8.5 discover-primary (查询首要服务)**

| 信息项 | 说明   |
|-----|--|
| 原型  | gatt discover-primary [UUID] [start handle] [end handle]                           |
| 功能  | 查询首要服务   |
| 参数  | [UUID]: 需要查询服务的 UUID;<br>[start handle]: 查询起始 handle;<br>[end handle]: 查询结束 handle |
| 响应  | <ACK> 200 OK   |
| 示例  | 查询所有首要服务: gatt discover-primary  |
| 备注  | N/A  |

**3.4.8.6 discover-secondary (查询次要服务)**

| 信息项 | 说明   |
|-----|--|
| 原型  | gatt discover-secondary [UUID] [start handle] [end handle]                         |
| 功能  | 查询次要服务   |
| 参数  | [UUID]: 需要查询服务的 UUID;<br>[start handle]: 查询起始 handle;<br>[end handle]: 查询结束 handle |
| 响应  | <ACK> 200 OK   |
| 示例  | 查询所有次要服务: gatt discover-secondary  |
| 备注  | N/A  |

**3.4.8.7 exchange-mtu (MTU包交互长度设置)**

| 信息项 | 说明                              |
|-----|---------------------------------|
| 原型  | gatt exchange-mtu               |
| 功能  | MTU 包交互长度设置                     |
| 参数  | N/A                             |
| 响应  | <ACK> 200 OK                    |
| 示例  | 设置 MTU 包交互长度: gatt exchange-mtu |
| 备注  | N/A                             |

### 3.4.8.8 att\_mtu (获取MTU大小)

| 信息项 | 说明                      |
|-----|-------------------------|
| 原型  | gatt att_mtu            |
| 功能  | 获取 MTU 大小               |
| 参数  | N/A                     |
| 响应  | <ACK> 200 OK            |
| 示例  | 获取 MTU 大小: gatt att_mtu |
| 备注  | N/A                     |

### 3.4.8.9 read (读取特定handle对应的属性值)

| 信息项 | 说明  |
|-----|---|
| 原型  | gatt read <handle> [offset]   |
| 功能  | 读取特定 handle 对应的属性值  |
| 参数  | <handle>: 需要读取的 handle 值, 16 进制数据;<br>[offset]: 相对第一个字节的偏移值, 16 进制数据; |
| 响应  | <ACK> 200 OK  |
| 示例  | 从 handle 对应属性值的第 8 字节开始读取: gatt read 2a00 8                           |
| 备注  | N/A   |

### 3.4.8.10 read-uuid (读取特定UUID的属性值)

| 信息项 | 说明   |
|-----|--|
| 原型  | gatt read-uuid <UUID> [start handle] [end handle]                                  |
| 功能  | 读取指定范围内特定 UUID 的属性值  |
| 参数  | [UUID]: 需要查询特性的 UUID;<br>[start handle]: 查询起始 handle;<br>[end handle]: 查询结束 handle |
| 响应  | <ACK> 200 OK   |
| 示例  | 从 handle 对应属性值的第 8 字节开始读取: gatt read 2a00 8  |
| 备注  | N/A  |

### 3.4.8.11 read-multiple (读取多个handle的属性值)

| 信息项 | 说明   |
|-----|--|
| 原型  | gatt read-multiple <handle 1> <handle 2> ... |

| 信息项 | 说明  |
|-----|---|
| 功能  | 读取多个 handle 的属性值                          |
| 参数  | handle: 需要查询特性的 handle, 16 进制数据。          |
| 响应  | <ACK> 200 OK                              |
| 示例  | 读取 2 个 handle 属性值: gatt read-multiple 1 2 |
| 备注  | N/A                                       |

#### 3.4.8.12 signed-write (签名写)

| 信息项 | 说明  |
|-----|---|
| 原型  | gatt signed-write <handle> <data> [length] [repeat]   |
| 功能  | 签名写   |
| 参数  | <handle>: 需要写的 handle 值, 16 进制数据<br><data>: 需要写入的数据, 16 进制数据, 当前只能输入 1 个 8bit 数据<br>[length]: 需要写入的数据长度, 上限为 512 字节, 会将 8bit 数据重复扩充到数据长度大小<br>[repeat]: 将写入数据重复发送次数 |
| 响应  | <ACK> 200 OK  |
| 示例  | 向为 0 的 handle 写入长度为 2 的数据 0x12: gatt signed-write 0 12 2  |
| 备注  | 签名写命令只能在配对绑定后使用   |

#### 3.4.8.13 subscribe (订阅服务)

| 信息项 | 说明  |
|-----|---|
| 原型  | gatt subscribe <CCC handle> <value handle> [ind]  |
| 功能  | 订阅服务  |
| 参数  | <CCC handle>: 16 进制数据, CCC 对应的 handle 值, UUID 为 2902;<br><value handle>: 16 进制数据, 一般为 CCC 对应的 handle 之前的一个 handle;<br>[ind]: 是否需要指示 |
| 响应  | <ACK> 200 OK  |
| 示例  | 以 notify 形式订阅某个 CCC: gatt subscribe 1a 19   |
| 备注  | GATT 服务中包含有 CCC 服务, 订阅才会生效  |

#### 3.4.8.14 unsubscribe (取消订阅)

| 信息项 | 说明               |
|-----|------------------|
| 原型  | gatt unsubscribe |

| 信息项 | 说明                    |
|-----|-----------------------|
| 功能  | 取消订阅                  |
| 参数  | N/A                   |
| 响应  | <ACK> 200 OK          |
| 示例  | 取消订阅：gatt unsubscribe |
| 备注  | 与订阅相对应，取消之前的订阅操作      |

### 3.4.8.15 write (有响应写命令)

| 信息项 | 说明  |
|-----|---|
| 原型  | gatt write <handle> <offset> <data> [string]  |
| 功能  | 有响应写命令  |
| 参数  | <handle>：需要写入数据对应的 handle，16 进制数据<br><offset>：与第一个属性数值的偏移量，16 进制数据<br><data>：需要写入的数据，16 进制数据，当前只能输入 1 个 8bit 数据<br>[string]：是否以字符串格式写 |
| 响应  | <ACK> 200 OK  |
| 示例  | 向值为 0 的 handle 内写入值为“ab”的字符串：gatt write 0 0 ab string   |
| 备注  | 需对端设备进行 response 响应   |

### 3.4.8.16 write-without-response (无响应写命令)

| 信息项 | 说明   |
|-----|--|
| 原型  | gatt write-without-response <handle> <data> [length] [repeat]  |
| 功能  | 无响应写命令   |
| 参数  | <handle>：需要写的 handle 值，16 进制数据<br><data>：需要写入的数据，16 进制数据，当前只能输入 1 个 8bit 数据<br>[length]：需要写入的数据长度，上限为 512 字节，会将 8bit 数据重复扩充到数据长度大小<br>[repeat]：将写入数据重复发送次数 |
| 响应  | <ACK> 200 OK   |
| 示例  | 向 handle 为 0 写入长度为 2 的数据 0x12：gatt write-without-response 0 12 2   |
| 备注  | 该写入不需对端设备响应，故无法判断写入有没有成功   |

**3.4.8.17 write-without-response-cb (使用特定写完成回调的无响应写命令)**

| 信息项 | 说明  |
|-----|---|
| 原型  | <code>gatt write-without-response-cb &lt;handle&gt; &lt;data&gt; [length] [repeat]</code>   |
| 功能  | 使用特定写完成回调的无响应写命令  |
| 参数  | <handle>: 需要写的 handle 值, 16 进制数据<br><data>: 需要写入的数据, 16 进制数据, 当前只能输入 1 个 8bit 数据<br>[length]: 需要写入的数据长度, 上限为 512 字节, 会将 8bit 数据重复扩充到数据长度大小<br>[repeat]: 将写入数据重复发送次数 |
| 响应  | <ACK> 200 OK  |
| 示例  | 向 handle0 内写入长度为 2 的数据 0x12: <code>gatt write-without-response-cb 0 12 2</code>   |
| 备注  | 与无响应写命令的区别在于写命令完成后使用指定的回调函数   |

**3.4.8.18 get (遍历范围内ATTR)**

| 信息项 | 说明  |
|-----|---|
| 原型  | <code>gatt get &lt;start handle&gt; [end handle]</code> |
| 功能  | 遍历范围内 ATTR  |
| 参数  | <start handle>: 起始 handle<br>[end handle]: 结束 handle    |
| 响应  | <ACK> 200 OK  |
| 示例  | 遍历 0~0x1000 范围内 ATTR: <code>gatt get 0 1000</code>      |
| 备注  | 输入为 16 进制数据   |

**3.4.8.19 set (设置某个ATTR)**

| 信息项 | 说明  |
|-----|---|
| 原型  | <code>gatt set &lt;handle&gt; [data]</code>             |
| 功能  | 设置某个 ATTR   |
| 参数  | <handle>: 需要设置的 handle<br>[data]: 需要写入的数据               |
| 响应  | <ACK> 200 OK  |
| 示例  | 设置 handle 为 10 的数据为 2a00: <code>gatt set 10 2a00</code> |
| 备注  | 直接修改当前某个 handle 特性的属性值                                  |

**3.4.8.20 show-db (显示所有ATTR)**

| 信息项 | 说明  |
|-----|---|
| 原型  | gatt show-db [uuid] [num_matches]                       |
| 功能  | 显示所有 ATTR   |
| 参数  | [uuid]: 需要查找的特定 UUID<br>[num_matches]: 需要查找的特定 UUID 的数量 |
| 响应  | <ACK> 200 OK  |
| 示例  | 查找遍历整个 GATT 数据: gatt show-db                            |
| 备注  | 不带参数为遍历整个 GATT 属性库, 带参数则只显示打印特定 UUID 及对应 handle         |

**3.4.8.21 metrics (注册metrics服务)**

| 信息项 | 说明   |
|-----|--|
| 原型  | gatt metrics [value: on, off]                      |
| 功能  | 开启/关闭 metrics 服务                                   |
| 参数  | [value]<br>on: 开启 metrics 服务<br>off: 关闭 metrics 服务 |
| 响应  | <ACK> 200 OK                                       |
| 示例  | 开启 metrics 服务: gatt metrics on                     |
| 备注  | 向 GATT 内注册特定的 metrics 服务                           |

**3.4.8.22 register (注册测试服务)**

| 信息项 | 说明                                      |
|-----|---|
| 原型  | gatt register                           |
| 功能  | 开启测试服务                                  |
| 参数  | N/A                                     |
| 响应  | <ACK> 200 OK                            |
| 示例  | 启用测试服务: gatt register                   |
| 备注  | 开启特定测试服务, 当前 CCC 服务在测试服务中, 需注册后才能进行订阅操作 |

**3.4.8.23 unregister (关闭测试服务)**

| 信息项 | 说明              |
|-----|-----------------|
| 原型  | gatt unregister |

| 信息项 | 说明                      |
|-----|-------------------------|
| 功能  | 关闭测试服务                  |
| 参数  | N/A                     |
| 响应  | <ACK> 200 OK            |
| 示例  | 关闭测试服务： gatt unregister |
| 备注  | N/A                     |

#### 3.4.8.24 notify (notify 通知)

| 信息项 | 说明                                  |
|-----|-------------------------------------|
| 原型  | gatt notify [data]                  |
| 功能  | notify 通知                           |
| 参数  | [data]: 需要通知的数据, 一个 8bit 的 16 进制数据  |
| 响应  | <ACK> 200 OK                        |
| 示例  | 主动通知对端设备数据修改为了 0x12: gatt notify 12 |
| 备注  | N/A                                 |

#### 3.4.8.25 indicate (indicate 通知)

| 信息项 | 说明                                    |
|-----|---------------------------------------|
| 原型  | gatt indicate [data]                  |
| 功能  | indicate 通知                           |
| 参数  | [data]: 需要通知的数据, 一个 8bit 的 16 进制数据    |
| 响应  | <ACK> 200 OK                          |
| 示例  | 被动通知对端设备数据修改为了 0x12: gatt indicate 12 |
| 备注  | N/A                                   |

## 4 示例说明

本章节提供了四个示例，分别以 Slave、Master 角色对 BLE 的 GATT 读写、配对绑定等常用场景进行演示说明。

### 4.1 GATT 读写示例 (Slave Role)

#### 4.1.1 示例简介

本示例通过演示 GATT 读写场景，简要介绍 GATT 命令的基本使用方法。

##### 4.1.1.1 获取方式

GATT 读写命令集成在 cmd\_gatt.c 文件中，通过 ble\_demo 工程可以完成 GATT 读写场景演示，ble\_demo 工程位于 XR806 SDK 的 /project/demo/bluetooth/ble\_demo 目录。

##### 4.1.1.2 准备工作

ble\_demo 工程的硬件准备有如下。

1. 评估板：运行示例工程代码。
2. 串口线：连接评估板的 Uart0 插针，用于 console 控制台的输入输出。
3. PC 机：用于镜像烧录和 console 控制的输入输出。
4. 手机/平板：用于运行 nRF Connect 软件，本文档的 nRF Connect 采用的版本是 V4.23.0。

##### 4.1.1.3 操作步骤

1. 编译 ble\_demo 工程，完成烧写后，复位即可，示例代码自动运行。
2. 手机/平板安装并运行 nRF Connect 软件。

#### 4.1.2 效果展示

1. 在评估板中运行 ble\_demo 工程后，在控制台中会打印出系统启动信息。

```

...
platform information =====
XR806 SDK v0.3.0 Feb 6 2021 20:11:07

heap space [0x20f200, 0x247c00], size 231936

cpu clock 160000000 Hz
HF clock 40000000 Hz

sdk option:
XIP : enable
INT LF OSC : enable
SIP flash : enable
    
```

```
=====
$>Welcome to Bluetooth Console Tools .....
$>
```

2. 进行 BLE 初始化，往串口输入 BLE 初始化命令：

```
$>ble init
```

成功初始化后，可以看到以下打印信息：

```
...
Device A not active,waking up!
ble rf_init done!
BLE INIT ALL DONE!
BT Coex. Init. OK.
== XRadio BLE HOST V2.3.0.0 ==
<ACK> 200 OK
$ [bt] [WRN] set_flow_control: Controller to host flow control not supported
[bt] [INF] bt_init: No ID address. App must call settings_load()
[cmd] Bluetooth initialized
[bt] [ERR] settings_call_set_handler: set-value failure. key: bt/hash error(-2)
[bt] [INF] bt_dev_show_info: Identity[0]: DA:1F:33:CA:3A:25 (random)
[bt] [INF] bt_dev_show_info: Identity[1]: FD:96:FD:48:C2:C7 (random)
[bt] [INF] bt_dev_show_info: HCI: version 5.0 (0x09) revision 0x0110, manufacturer 0x063d
[bt] [INF] bt_dev_show_info: LMP: version 5.0 (0x09) subver 0x0110
```

3. 开始进行 BLE 广播，往串口输入广播命令：

```
$>ble advertise on
```

成功广播后有如下提示：

```
*****
[RandomAddress 7D:2D:D1:24:23:FD ]
*****
[cmd] Advertising started
<ACK> 200 OK
```

红色部分就是 BLE 的广播地址。

4. 注册 GATT 服务，往串口输入命令：

```
$>gatt metrics on
```

成功注册后有如下提示：

```
[cmd] Registering GATT metrics test Service.
[cmd] GATT write cmd metrics on.
```

<ACK> 200 OK

5. 开始进行连接：先打开 nRF Connect 软件，进行周围设备扫描，可以找到评估板的设备信息，即下图红色部分，然后点击“CONNECT”发起连接。

图 4-1 nRF Connect 扫描



连接成功后可以看到 nRF Connect 跳转到以下界面了：

图 4-2 nRF Connect 连接界面

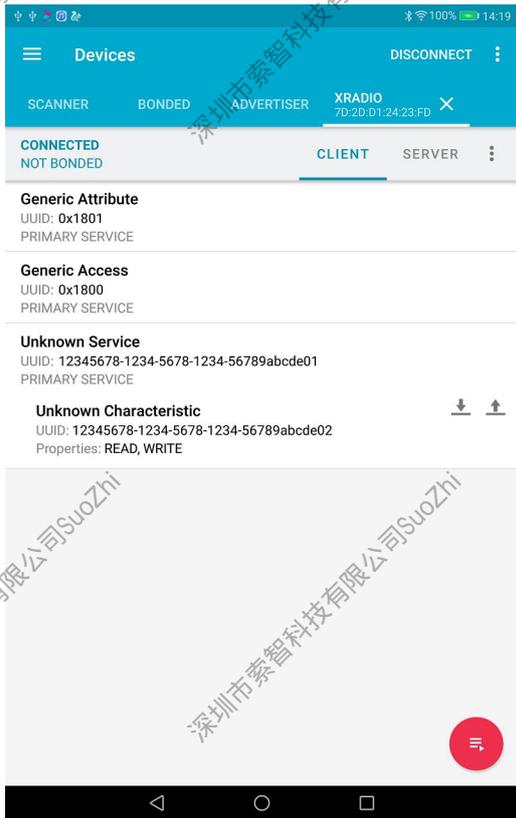


同时评估板出现以下连接成功的打印信息：

```
[cmd] Connected: 6E:8A:99:6F:0F:E7 (random)
[cmd] Remote LMP version 4.2 (0x08) subversion 0x0608 manufacturer 0x010f
[cmd] LE Features: 0x0000000000000019
[cmd] LE conn param updated: int 0x0006 lat 0 to 2000
[cmd] LE conn param updated: int 0x0028 lat 0 to 2000
```

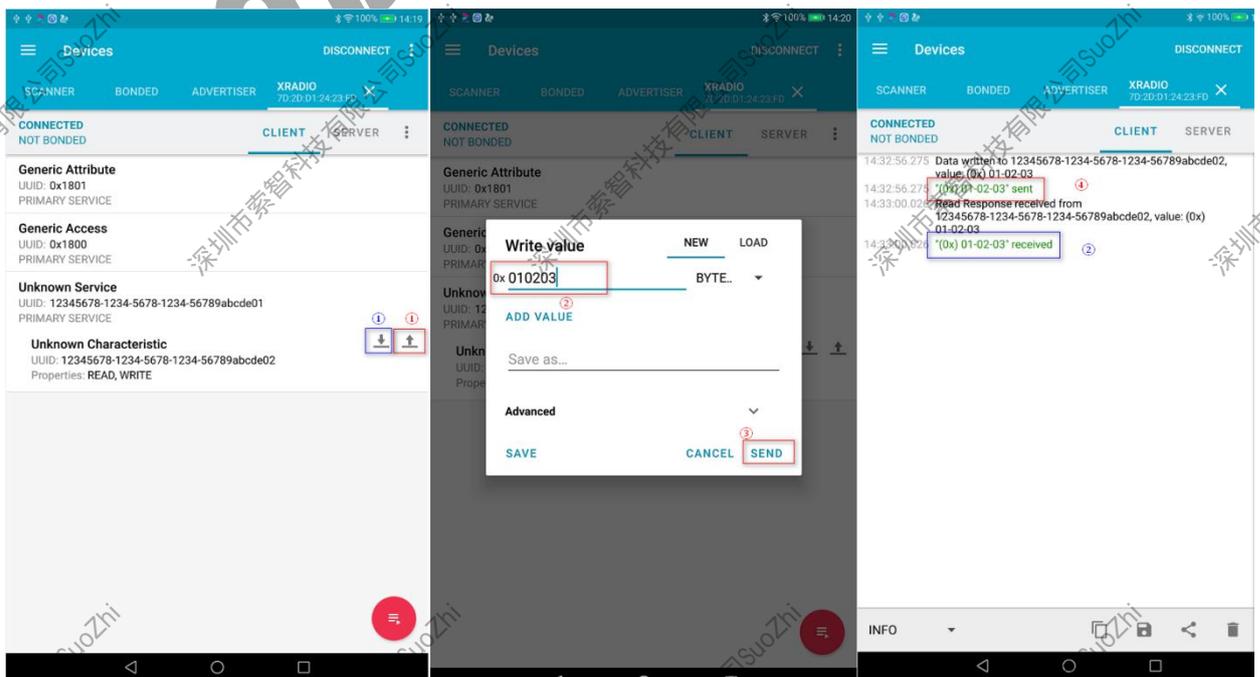
红色部分的 Service 即是步骤 4 所创建的 GATT 服务。点开“Unknown Service”，可以看到注册的服务包括一个可读可写的属性。

图 4-3 BLE GATT Service



6. 进行读写操作：如写入值“0x010203”，按照下图的红色序号“①->②->③”操作，然后在红色序号④可以看到数据正常写入。此时再进行读操作，点击蓝色的①，可以在蓝色序号②的位置看到读取的结果。

图 4-4 BLE GATT 读写过程



## 4.2 配对绑定示例 (Slave Role)

### 4.2.1 示例简介

本示例通过演示“输入配对密钥配对”的场景，简要介绍 BLE 配对绑定的基本使用方法。

#### 4.2.1.1 获取方式

配对绑定命令集成在 cmd\_ble.c 文件中，通过 ble\_demo 工程可以完成 BLE 配对场景演示，ble\_demo 工程位于 XR806 SDK 的/project/demo/bluetooth/ble\_demo 目录。

#### 4.2.1.2 准备工作

准备工作同 4.1.1.2。

#### 4.2.1.3 操作步骤

操作步骤同 4.1.1.3。

### 4.2.2 效果展示

1. 在评估板中运行 ble\_demo 工程。
2. 进行 BLE 初始化，往串口输入 BLE 初始化命令：

```
$>ble init
```

3. 使能和配置配对绑定方式：

```
$>ble bondable on
$>ble auth input
```

4. 开始进行 BLE 广播，往串口输入广播命令：

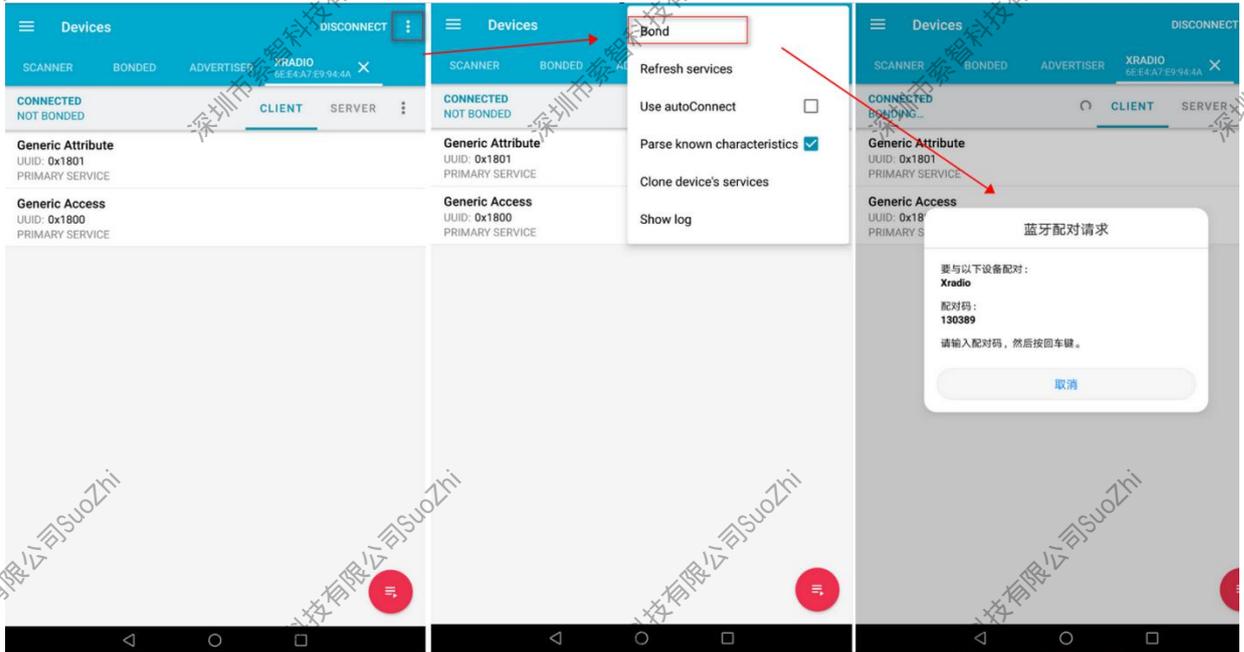
```
$>ble advertise on
```

成功广播后有如下提示：

```
*****
[RandomAddress 6E:E4:A7:E9:94:4A ]
*****
[cmd] Advertising started
<ACK> 200 OK
```

5. 开始连接，对于配对绑定也可以不需要连接这一步骤。先打开 nRF Connect 软件，进行周围设备扫描，可以找到评估板的设备信息，然后点击“CONNECT”发起连接。
6. 连接成功后，在 nRF Connect 菜单中点击“Bond”发起配对，会弹出配对请求的提示框。

图 4-5 BLE 配对绑定过程



在提示框可以看到配对码是“130389”。

7. 往串口输入配对码：

```
$>ble auth-passkey 130389
```

配对成功后可以看到以下打印：

```
[cmd] Security changed: 58:BA:72:B0:4C:EB (random) level 4
[cmd] Identity resolved 58:BA:72:B0:4C:EB (random) -> 30:74:96:99:1A:76 (public)
[cmd] Bonded with 30:74:96:99:1A:76 (public)
[cmd] LE conn param updated: int 0x0028 lat 0 to 2000
```

### 4.3 GATT 读写示例 (Master Role)

#### 4.3.1 示例简介

本示例通过演示 GATT 读写场景，简要介绍 GATT 命令的基本使用方法。

##### 4.3.1.1 获取方式

GATT 读写命令集成在 cmd\_gatt.c 文件中，通过 ble\_demo 工程可以完成 GATT 读写场景演示，ble\_demo 工程位于 XR806 SDK 的/project/demo/bluetooth/ble\_demo 目录。

##### 4.3.1.2 准备工作

ble\_demo 工程的硬件准备有如下。

1. 评估板：2 块，运行示例工程代码。
2. 串口线：连接评估板的 Uart0 插针，用于 console 控制台的输入输出。
3. PC 机：用于镜像烧录和 console 控制的输入输出。

4.3.1.3 操作步骤

1. 编译 ble\_demo 工程，两块评估板完成烧写后，复位即可，示例代码自动运行。

4.3.2 效果展示

1. 两块评估板分别运行 ble\_demo 工程。两块评估板以下用 A/B 简称，A 作为 Master、B 作为 Slave。
2. 评估板 A/B 分别执行 BLE 初始化命令：

```
$>ble init
```

3. 评估板 B 执行 “ble advertise on” 命令开始广播：

```
*****
[RandomAddress 5F:6F:67:DA:79:61 ]
*****
[cmd] Advertising started
<ACK> 200 OK
```

4. 评估板 B 注册 GATT 服务，往串口输入命令

```
$>gatt metrics on
```

成功注册后有如下提示：

```
[cmd] Registering GATT metrics test Service.
[cmd] GATT write cmd metrics on.
<ACK> 200 OK
```

5. 评估板 A 开始扫描设备：

```
$>ble scan on
```

可以在串口中获取到扫描的结果：

```
...
[cmd] [DEVICE]: 5F:6F:67:DA:79:61 (random), AD evt type 4, RSSI -23 Xradio C:0 S:1 D:0 SR:1 E:0 Prim: LE 1M, Secn: No packets
[cmd] [DEVICE]: 71:CE:BD:2C:89:47 (random), AD evt type 0, RSSI -62 C:1 S:1 D:0 SR:0 E:0 Prim: LE 1M, Secn: No packets
[cmd] [DEVICE]: 71:CE:BD:2C:89:47 (random), AD evt type 4, RSSI -62 ble-32768-test-48-hour C:0 S:1 D:0 SR:1 E:0 Prim: LE 1M, Secn: No packets
[cmd] [DEVICE]: 14:08:16:04:46:4A (public), AD evt type 0, RSSI -88 C:1 S:1 D:0 SR:0 E:0 Prim: LE 1M, Secn: No packets
...
```

从结果信息可以找到评估板 B 广播的信息，即上面方框红色部分。

6. 评估板 A 开始连接 B：

```
$>ble connect 5F:6F:67:DA:79:61 random
```

连接成功后，A 有以下打印：

```
*****
[RandomAddress 56:25:4C:81:C7:1E ]
*****
[cmd] Connection pending
<ACK> 200 OK
$ [cmd] Connected: 5F:6F:67:DA:79:61 (random)
```

同时 B 也有成功连接的打印：

```
[cmd] Connected: 56:25:4C:81:C7:1E (random)
[cmd] Remote LMP version 5.0 (0x09) subversion 0x0110 manufacturer 0x063d
[cmd] LE Features: 0x00000000000179ff
```

7. 进行服务的查找，往 A 评估板输入命令：

```
$>gatt discover
```

串口打印出以下的查询结果：

```
$ [cmd] Descriptor 2800 found: handle 1
[cmd] Descriptor 2803 found: handle 2
[cmd] Descriptor 2a05 found: handle 3
[cmd] Descriptor 2902 found: handle 4
[cmd] Descriptor 2800 found: handle 5
[cmd] Descriptor 2803 found: handle 6
[cmd] Descriptor 2a00 found: handle 7
[cmd] Descriptor 2803 found: handle 8
[cmd] Descriptor 2a01 found: handle 9
[cmd] Descriptor 2803 found: handle a
[cmd] Descriptor 2aa6 found: handle b
[cmd] Descriptor 2800 found: handle c
[cmd] Descriptor 2803 found: handle d
[cmd] Descriptor 12345678-1234-5678-1234-56789abcde02 found: handle e
[cmd] Discover complete
```

可以看到，“12345678-1234-5678-1234-56789abcde02”是评估板 B 注册的属性，可读可写，其 handle 是 e。

8. 进行 GATT 写操作：评估板 A 往 handle 是 e 的属性写字符串“ab”：

```
$>gatt write e 0 ab string
```

成功写入后的响应是：

```
[cmd] Write pending
<ACK> 200 OK
$ [cmd] Write complete: err 0x00
```

9. 进行 GATT 读操作：评估板 A 往 handle 是 e 的属性读数据：

```
$>gatt read e
```

成功读取到打印是：

```
[cmd] Read pending
<ACK> 200 OK
$ [cmd] Read complete: err 0x00 length 2
[cmd] Read Value:
6162
```

其中 61/62 正是 a/b 的 ASCII 码。

## 4.4 配对绑定示例 (Master Role)

### 4.4.1 示例简介

本示例通过演示“输入配对密钥配对”的场景，简要介绍 BLE 配对绑定的基本使用方法。

#### 4.4.1.1 获取方式

配对绑定命令集成在 cmd\_ble.c 文件中，通过 ble\_demo 工程可以完成 BLE 配对场景演示，ble\_demo 工程位于 XR806 SDK 的/project/demo/bluetooth/ble\_demo 目录。

#### 4.4.1.2 准备工作

准备工作同 4.3.1.2。

#### 4.4.1.3 操作步骤

操作步骤同 4.3.1.3。

### 4.4.2 效果展示

1. 两块评估板分别运行 ble\_demo 工程。两块评估板以下用 A/B 简称，A 作为 Master、B 作为 Slave。
2. 评估板 A/B 分别执行 BLE 初始化命令：

```
$>ble init
```

3. 评估板 B 执行“ble advertise on”命令开始广播：

```
*****
[RandomAddress 4C:02:97:82:F7:F4 ]
*****
[cmd] Advertising started
<ACK> 200 OK
```

4. 评估板 B 使能和配置配对绑定方式：

```
$>ble bondable on
$>ble auth input
```

5. 评估板 A 配置认证的方式：

```
$>ble auth input
```

6. 评估板 A 连接上 B，输入命令 “ble connect 4C:02:97:82:F7:F4 random”：

```
*****
[RandomAddress 68:D6:26:BE:96:E0 ]
*****
[cmd] Connection pending
<ACK> 200 OK
$ [cmd] Connected: 4C:02:97:82:F7:F4 (random)
```

7. 评估板 A 使用 security 命令配置安全等级：

```
$> ble security 4
```

此时评估板 A 打印出以下信息：

```
$ [cmd] Remote pairing features: IO: 0x02, OOB: 0, AUTH: 0x0d, Key: 16, Init Kdist: 0x06, Resp Kdist: 0x06
[cmd] Enter passkey for 4C:02:97:82:F7:F4 (random)
```

评估板 B 打印出以下信息：

```
[cmd] Remote pairing features: IO: 0x02, OOB: 0, AUTH: 0x0d, Key: 16, Init Kdist: 0x07, Resp Kdist: 0x07
[cmd] Enter passkey for 68:D6:26:BE:96:E0 (random)
```

8. 分别往 A、B 输入配对密钥，如 123456：

```
$> ble auth-passkey 123456
```

然后可以看到评估板 A 打印出以下信息：

```
$ [cmd] Security changed: 4C:02:97:82:F7:F4 (random) level 4
[cmd] Identity resolved 4C:02:97:82:F7:F4 (random) -> DD:3D:03:F5:11:09 (random)
[cmd] Bonded with DD:3D:03:F5:11:09 (random)
```

评估板 B 打印出以下信息：

```
$ [cmd] Security changed: 68:D6:26:BE:96:E0 (random) level 4
[cmd] Identity resolved 68:D6:26:BE:96:E0 (random) -> DA:1F:33:CA:3A:25 (random)
[cmd] Bonded with DA:1F:33:CA:3A:25 (random)
```

至此，配对绑定成功了。

## 著作权声明

版权所有©2020 广州芯之联科技有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由广州芯之联科技有限公司（“芯之联”）拥有并保留一切权利。

本文档是芯之联的原创作品和版权财产，未经芯之联书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

## 商标声明



KRAD TECH、**芯之联**（不完全列举）均为广州芯之联科技有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

## 免责声明

您购买的产品、服务或特性应受您与广州芯之联科技有限公司（“芯之联”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，芯之联概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。芯之联尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，芯之联概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予芯之联的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。芯之联不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。芯之联不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。