

# AiLink 蓝牙锁应用手册

版本：V1.0

更新日期：2022 年 8 月 19 日

深圳市易连物联网有限公司版权所有

本产品的规格书如有变更，恕不另行通知。

深圳市易连物联网有限公司保留在不另行通知的情况下，对其中所包含的规格书和材料进行更改的权利，同时由于信任所引用的材料所造成的损害（包括结果性损害），包括但不限于印刷上的错误和其他与此出版物相关的错误，易连物联网公司将不承担责任。

## 修改记录

文档版本	作者	发布日期	修改说明
V1.0	lx1	2022/8/19	1. 把 ailink-蓝牙锁部分单独出来，删除无关指令。

# 目录

修改记录	- 2 -
目录	- 3 -
1 概述	- 5 -
2 说明	- 5 -
3 工作模式	- 5 -
3.1 模式一：断电模式。	- 6 -
3.2 模式二：长供电休眠模式。	- 6 -
4 蓝牙接口（默认）	- 8 -
4.1 蓝牙名称：AiLink_xxxx	- 8 -
4.2 UUID 说明	- 8 -
4.3 蓝牙连接服务列表 1：FFE0 举例	- 8 -
4.4 广播数据	- 8 -
4.4.1 第一类广播设置	- 9 -
4.4.2 第二类广播设置	- 10 -
5 BM 模块与 MCU 交互协议	- 11 -
5.1 设置（获取）指令	- 11 -
5.1.1 设置、获取广播名称（Type: 01、02）	- 12 -
5.1.2 设置、获取广播间隔时间（Type: 05、06）	- 14 -
5.1.3 读取 MAC 地址（Type: 0D）	- 15 -
5.1.4 读取 BM 模块软、硬件版本号（Type: 0E）	- 16 -
5.1.5 设置、读取 MCU 软硬件版本号（Type: 0F、10）	- 17 -
5.1.6 设置、读取模块自动休眠时间（Type: 17、18）	- 18 -
5.1.7 设置模块进入睡眠（Type: 19）	- 20 -
5.1.8 设置模块唤醒（Type: 1A）	- 21 -
5.1.9 设置、读取系统当前时间（Type: 1B、1C）	- 22 -
5.1.10 设置、读取 CID、VID、PID（Type: 1D、1E）	- 24 -
5.1.11 设置模块重启（Type: 21）	- 25 -
5.1.12 设置恢复出厂设置（Type: 22）	- 26 -
5.1.13 设置、获取 BM 模块状态（Type: 25、26）	- 27 -
5.1.14 MCU 上报 MCU 电池状态（Type: 27、28）	- 28 -
5.1.15 设置使能绑定、取消使能绑定（Type: 32）	- 30 -
5.1.16 MCU 设置蓝牙锁开锁类型（Type: 33、34）	- 30 -
5.1.17 MCU 上传设备基本信息（Type: 35、36）	- 32 -
5.1.18 APP 同步时间到 MCU（Type: 37、38）	- 34 -
5.1.19 BM 模块自动唤醒设置、自动进入睡眠返回（Type: 3A、3B）	- 35 -
5.2 协议透传指令	- 36 -
5.3 数据透传	- 36 -
6 协议透传产品介绍	- 36 -

---

6.1 蓝牙门锁 .....	- 37 -
6.1.1 工作流程 .....	- 37 -
6.1.2 获取蓝牙门锁的开锁类型 .....	- 39 -
6.1.3 APP 绑定蓝牙锁流程 .....	- 39 -
6.1.4 APP 添加用户流程 .....	- 39 -
6.1.5 APP 分享用户使用 .....	- 40 -
6.1.6 一次性密码生成与使用 .....	- 40 -
6.1.7 响铃功能 .....	- 40 -
6.1.8 开关锁和异常操作的记录的 .....	- 40 -
6.1.9 用户 ID 编号说明 .....	- 40 -
6.1.10 数据指令 .....	- 41 -
6.2 使用/测试指导 .....	- 65 -
6.3 测试工具 .....	- 65 -
6.4 测试步骤 .....	- 65 -
7 联系我们 .....	- 65 -

## 1 概述

- 1.1 本文档适用于深圳市易连物联网 BM 系列 BLE 模块 接入 ailink APP。
- 1.2 本文档适用于蓝牙锁的 MCU 端开发工程师使用。
- 1.3 本文档讲详细介绍硬件对接、固件对接。
- 1.4 文档会保持更新，以[官网链接](#)为最新版本。
- 1.5 我们提供的蓝牙模块具有功耗低、认证齐全、APP 功能强大体验好等特点。扫描下面二维码  
下载 APP。



- 1.6 支持 MCU 配置模块（VID、PID）实现 APP 连接产品时型号自定义、图标自定义等个性化设计。

## 2 说明

- 2.1 BLE（Slave）与 APP（Master）交互的每包数据默认最大为 20byte，当 MCU 端一次性发送超过 20byte 时，BLE 会将数据进行分包发送给 APP，需 50byte 则分为 20+20+10，分 3 次发送给 APP。
- 2.2 BM 模块上电需要时间进行配置，当配置完成，进入就绪时，BM 模块会主动给 MCU 返回一个 BM 模块状态信息。详情请查看“[设置、获取 BM 模块状态](#)”。

## 3 工作模式

- BM 模块支持两种工作模式，断电模式和不断电休眠模式，用户可以根据自身需求合理选择工作模式。  
用户可以在设计 PCB 的时候，预留两种方式的电路。详情请查看硬件规格书规格书。

## 3.1 模式一：断电模式。

- 在此模式下，BM 模块完全断电，需要供电才能正常工作，这种模式有利于省电。
- 在此模式下，MCU 可以根据 BM 模块的连接状态选择合适的时间断电关机，例如，在非蓝牙连接状态时，MCU 工作完 10s 后断电关机，在蓝牙连接状态时，工作完 30s 后断电关机。获取 BM 模块的连接状态，可以根据蓝牙状态脚（BT-CS）进行判断，也可以通过串口读取模块状态。这种做法有利于用户能够顺利传输数据到 APP 上，而不会出现反复关机断连问题。

工作流程：

- 1、BM 模块上电。
- 2、BM 模块上电就绪后，BM 会给 MCU 返回 [BM 模块状态](#)。
- 3、MCU [设置 CID](#)。
- 4、MCU 设置模块其他内容。
- 5、MCU 发送数据。
- 6、MCU、BM 断电关机。

## 3.2 模式二：长供电休眠模式。

- 此模式下，BM 模块需要长供电，不会断电关机（串口已关闭，BM 处于低功耗模式）。

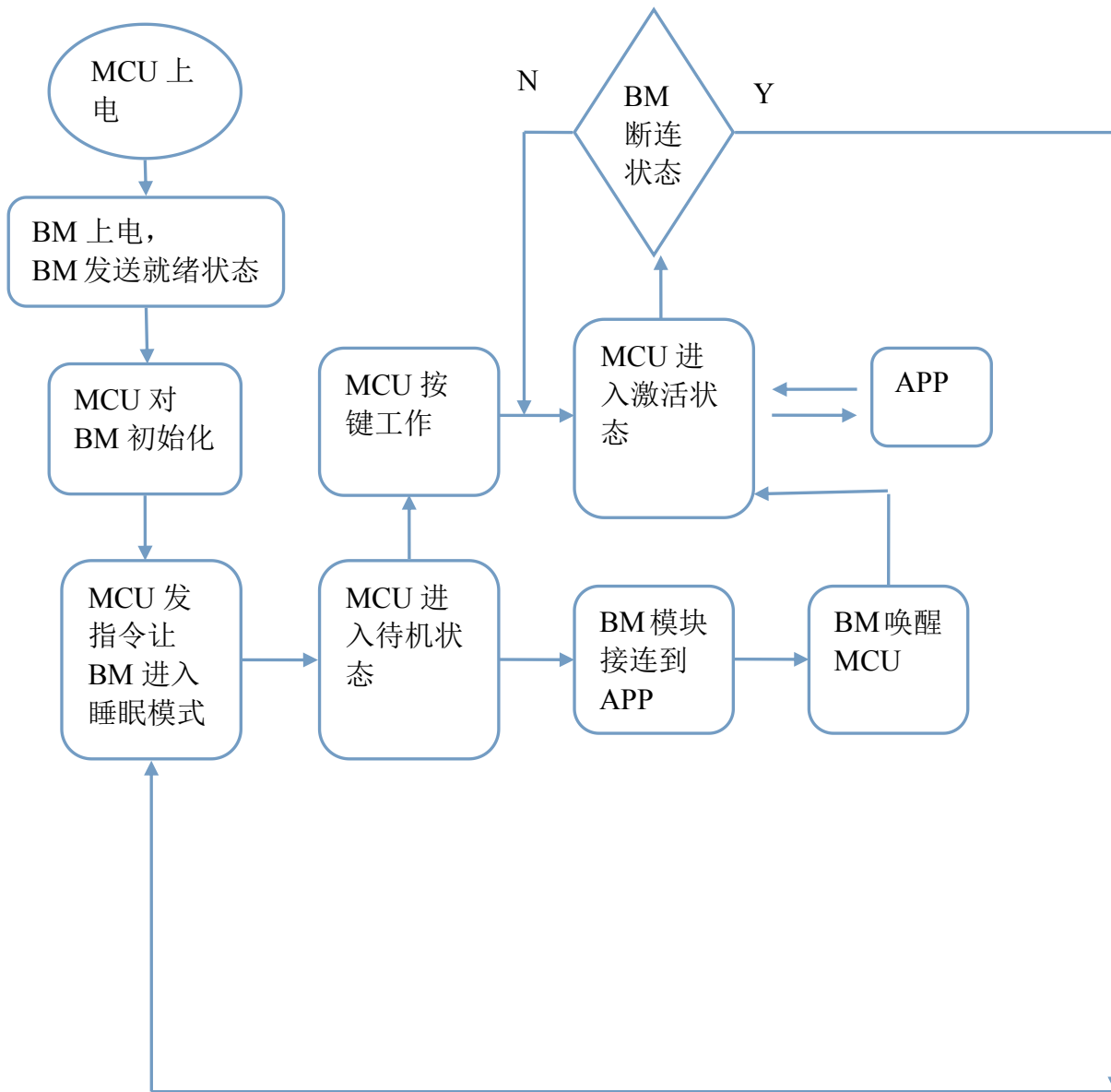
➤ 工作流程：

- 1、BM 模块上电。
- 2、BM 模块上电就绪后，BM 会给 MCU 返回 [BM 模块状态](#)。
- 3、MCU [设置 CID](#)。
- 4、MCU 设置模块其他内容。
- 5、MCU 发送数据。
- 6、MCU 发送[睡眠指令](#)（可以选择是否开启低频广播），使 BM 模块进入低功耗模式。

若开启了 BM 模块进入了低功耗模式并且开启了低频广播，当 BM 模块连上 APP 后，BM 模块会发送[模块状态](#)信息到 MCU，同时蓝牙状态脚会拉低，用以唤醒 MCU（MCU 可以用串口唤醒或者蓝牙状态脚唤醒）。

7、MCU 主动唤醒 BM 模块。当 BM 模块处于休眠状态时，若 MCU 需要发送数据到 BM 模块，需要先发一条[唤醒指令](#)到 BM 模块。注：BM 模块刚唤醒时，是无法正常接收数据的，所以 BM 收到第一组的唤醒指令时，BM 是不会回复 MCU 状态的。MCU 可以发两次唤醒指令。

- 工作流程参考如下：



## 4 蓝牙接口（默认）

### 4.1 蓝牙名称：AiLink\_xxxx

注：xxxx 为 Mac 地址后 4 个字符

### 4.2 UUID 说明

BM 模块有两个服务 UUID，一个是模块固定的服务 UUID，为 FFE0，一个是用户可以自定义的服务 UUID，默认为 FEE0。

易联物联网的 AiLink APP 交互使用的服务 UUID 为 FFE0。

同时，两个 UUID 都可以作为普通的数据交互 UUID。

### 4.3 蓝牙连接服务列表 1： FFE0 举例

#### 4.3.1 服务 UUID：

0000**FFE0**-0000-1000-8000-00805F9B34FB

#### 4.3.2 特征值 UUID1：

0000**FFE1**-0000-1000-8000-00805F9B34FB

属性： read,write,write no response

功能： APP 下发的数据会通过此 UUID 透传给 MCU

#### 4.3.3 特征值 UUID2：

0000**FFE2**-0000-1000-8000-00805F9B34FB

属性： read,notify

功能： MCU 发给 BLE 的数据由此 UUID 透传给 APP

#### 4.3.4 特征值 UUID3：

0000**FFE3**-0000-1000-8000-00805F9B34FB

属性： read,write,write no response,notify

功能： APP 与 BLE 进行[设置类指令](#)的 UUID，有 write 和 notify

### 4.4 广播数据

说明：广播数据有两类，用户只能选择其中的一类进行设置。

**第一类广播：AiLink 设置（默认）。**使用我司 AiLink APP 接入的设置，需根据我司要求的格式进行设置。



**第二类广播：用户自定义设置。**不使用我司 AiLink APP 接入的设置，用户可以根据自身需求进行设置。若不设置则默认为我司设置。

## 4.4.1 第一类广播设置

AiLink 设置广播数据内容包含（详情设置请查看[设置读取 CID、PID、VID](#)）：

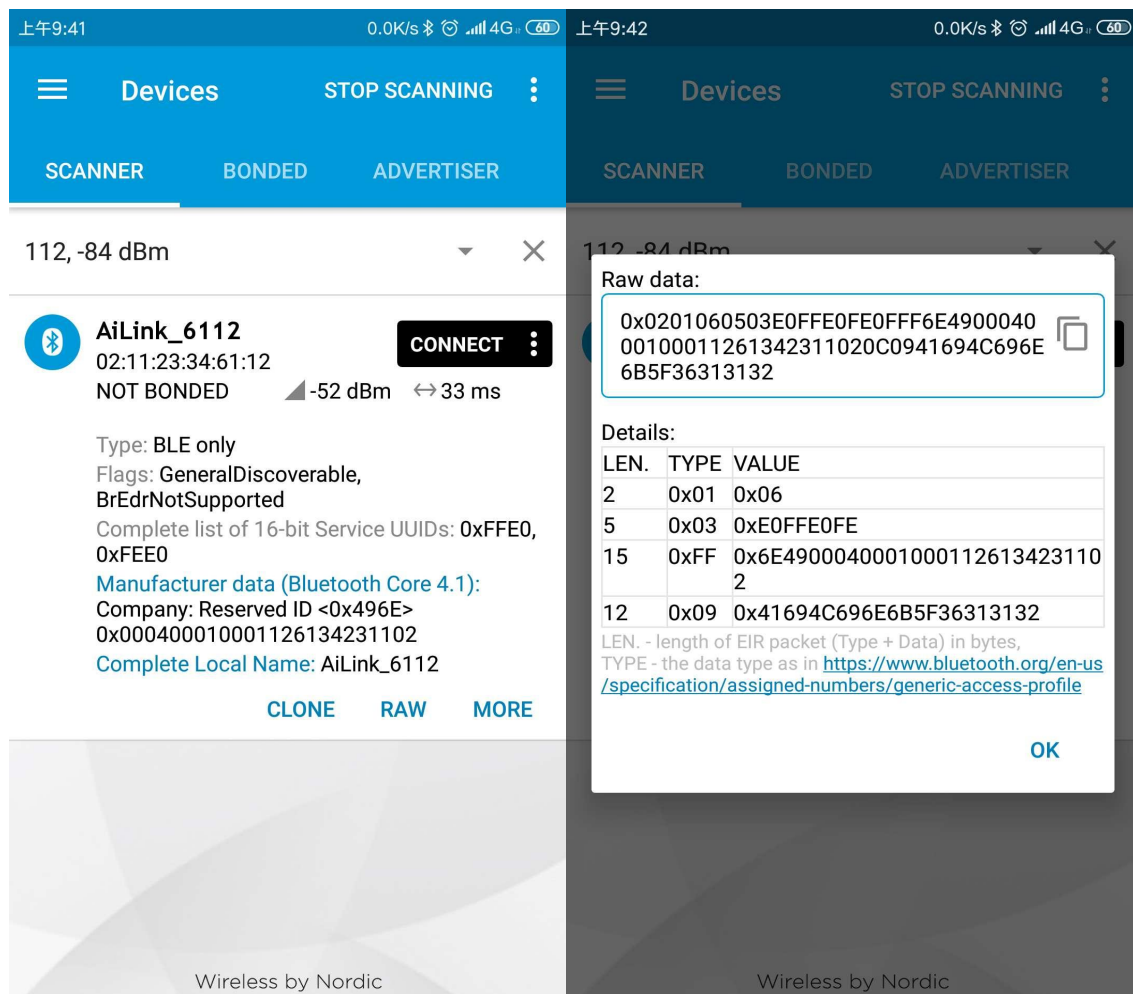
- 1、Company ID 。固定：496e（In, Inet 缩写，此处预留后续申请 SIG company 号）
- 2、CID：产品类型（2bytes）（例如血压计为 0x0001，额温枪为 0x0002，详查看[协议透传产品介绍](#)）
- 3、VID：厂商 ID（2byte）（由我司分配）
- 4、PID：产品 ID（2byte）（由厂商分配）
- 5、Mac 地址（MAC 是固定的，[大小端序可设置](#)，默认小端序）

例如广播出来的自定义数据为：

6e49000100010001126134231102

6e49：为 In，0001 是 CID，表示产品类型，0001 是 VID，表示厂商 ID，0001 是 PID，表示产品 ID。126134231102 是 Mac 地址，因为是小端序，所以 Mac 地址是：02 : 11 : 23 : 34 : 61 : 12

蓝牙工具显示如下图：



## 4.4.2 第二类广播设置

若使用此类广播自定义数据，则第一类的广播自定义数据不启动（CID、VID、PID 等设置不启用）。详情设置请查看[第二类广播自定义设置](#)。

## 5 BM 模块与 MCU 交互协议

### 5.1 设置（获取）指令

设置类指令格式规范（不透传）：

Byte	Value	Description
0	0xA6	包头
1		Payload 长度（最大 16byte）
2 ~n		Payload
n+1	SUM（1~n）	(1~n)校验和
n+2	0x6A	包尾（注：n+2 不能超过 20）

包头和包尾是固定的，分别为 0xA6，和 0x6A。

校验和是指 byte1 + byte2 + ...+byte n 的和，取低位 1 byte。

设置指令里，数据的 Byte 数不能超过 20

### 5.1.1 设置、获取广播名称 (Type: 01、02)

#### 设置蓝牙名称:

- 设置蓝牙名称可以设置为固定字符作为蓝牙名称，例如设置为 swan，所有的模块都会显示为 swan。同时也可以设置为固定蓝牙名称+ “\_” + Mac 地址的方式，这样子有利于每个模块的名称都有差异。详细见如下指令说明：

Byte	Value	Description	
0	0xA6	包头	
1	Len	Payload 长度 (最大 16byte)	
2	0x01	Type: 设置蓝牙名称	Payload
3 ~ n	Name	名称 (需要对应 ASCII 表)	
n+1	Num	MAC 字符个数: 名称后面跟随的 MAC 字符的个数 0: 代表没有, 则是固定蓝牙名称。 1: 代表后面带有 mac 地址的 1 个字符, 例如: Swan_x。 2: 代表后面带有 mac 地址的 2 个字符, 例如: Swan_xx。 默认 Num=4; Num 最大为 12 注: Name 长度+ “_” +Num 最大为 15	
n + 2	Sum	(1~n)校验和	
N+3	0x6A	包尾	

举例： 蓝牙的 MAC 地址为 12 : 34 : 56 : 78 : 9A : BC。

- 如果设置蓝牙名称为 swan, 且不带 MAC 地址时, 那么发送 A6 06 01 73 77 61 6E 00 C0 6A , 则蓝牙名称为 swan
- 如果设置蓝牙名称为 swan, 且带 MAC 地址 2 个字符, 那么发送 A6 06 01 73 77 61 6E 02 C2 6A , 则蓝牙名称为 swan\_BC
- 如果设置蓝牙名称为 swan, 且带 MAC 地址 4 个字符, 那么发送 A6 06 01 73 77 61 6E 04 C4 6A , 则蓝牙名称为 swan\_9ABC
- 整个蓝牙名称长度最长为 15 个字符。

#### BM 回复设置结果:

Byte	Value	Description
------	-------	-------------

0	0xA6	包头	
1	0x02	Payload 长度	
2	0x01	Type: 回复设置蓝牙名称结果	Payload
3		结果值: 0: 成功 (立即生效) 1: 失败 2: 不支持	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

举例：设置成功

返回 A6 02 01 00 03 6A

设置失败

返回 A6 02 01 01 04 6A

获取蓝牙名称：

Byte	Value	Description	
0	0xA6	包头	
1	0x01	Payload 长度	
2	0x02	Type: 获取蓝牙名称	Payload
3	0x03	(1~2)校验和	
4	0x6A	包尾	

BM 返回蓝牙名称：

Byte	Value	Description	
0	0xA6	包头	
1	Len	Payload 长度 (最大 16 byte)	
2	0x02	Type: 回复蓝牙名称	Payload
3 ~ n	Name	蓝牙名称 (最长 15 byte)	
n + 1	Sum	(1~n)校验和	
n + 2	0x6A	包尾	

- 举例：蓝牙名称为 swan\_BC
- 发送查询指令：A6 01 02 03 6A
- BM 返回名称：A6 08 02 73 77 61 6E 5F 42 43 A7 6A

## 5.1.2 设置、获取广播间隔时间（Type: 05、06）

### 设置广播间隔：

Byte	Value	Description	
0	0xA6	包头	
1	0x03	Payload 长度	
2	0x05	Type: 设置广播间隔（单位：ms 范围：20-2000；默认 200）	Payload
3		广播间隔时间的高字节	
4		广播间隔时间的低字节	
5	Sum	(1~4)校验和	
6	0x6A	包尾	

- 举例：设置广播间隔为：1000ms  
 发送：A6 03 05 03 E8 F3 6A

### BM 回复设置结果：

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x05	Type: 回复设置蓝牙广播间隔结果	Payload
3		结果值： 0: 成功 1: 失败 2: 不支持	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

### 获取广播间隔时间：

Byte	Value	Description	
0	0xA6	包头	
1	0x01	Payload 长度	
2	0x06	Type: 获取蓝牙广播间隔	Payload
3	0x07	(1~2)校验和	
4	0x6A	包尾	

**BM 返回广播间隔：**

Byte	Value	Description	
0	0xA6	包头	
1	0x03	Payload 长度	
2	0x06	Type: 回复广播间隔时间 (单位: ms)	Payload
3		广播间隔时间的高字节	
4		广播间隔时间的低字节	
5	Sum	(1~4)校验和	
6	0x6A	包尾	

➤ 举例：广播间隔为 1000ms

返回 A6 03 06 03 E8 F4 6A

### 5.1.3 读取 MAC 地址 (Type: 0D)

**读取 Mac 地址值：**

Byte	Value	Description	
0	0xA6	包头	
1	0x01	Payload 长度	
2	0x0D	Type: 读取 MAC 地址值	Payload
3	0x0E	(1~2)校验和	
4	0x6A	包尾	

**BM 返回 MAC 地址值：**

Byte	Value	Description	
0	0xA6	包头	
1	0x07	Payload 长度	
2	0x0D	Type: 回复 Mac 地址值	Payload
3~8		Mac 地址值 (6byte、小端序)	
9	Sum	(1~8)校验和	
10	0x6A	包尾	

举例：MAC 地址为 11 : 22 : 33 : 44 : 55 : 66

返回：A6 07 0D 66 55 44 33 22 11 79 6A

## 5.1.4 读取 BM 模块软、硬件版本号 (Type: 0E)

读取 BM 模块软硬件版本号:

Byte	Value	Description	
0	0xA6	包头	
1	0x01	Payload 长度	
2	0x0E	Type: 读取 BM 模块软硬件版本号	Payload
3	0x0F	(1~2)校验和	
4	0x6A	包尾	

BM 返回软硬件版本号:

Byte	Value	Description	
0	0xA6	包头	
1		Payload 长度	
2	0x0E	Type: 回复 BM 模块软硬件版本号	Payload
3		产品型号。byte3 、byte4 为 ASCII 字符, byte5 为数字。	
4			
5			
6		硬件版本号 H	
7		软件版本号 S	
8		定制版本号 P	
9		年 实际年份=年+2000 例如: 2019 年 年=2019-2000=19	
10		月 1~12	
11		日 1~31	
12	Sum	校验和	
13	0x6A	包尾	

➤ 举例: 如软硬件版本号为 BM16H1S1.0P0\_20190507

解析: BM16 为产品型号, 对应实际数据为 0x42 0x4D 0x10

H1 为硬件版本号 1, 对应实际数据为 0x01

S1.0 为软件版本号 1.0, 对应实际数据为: 0x0A (带 1 位小数点)

P0 为定制版本号, 对应实际数据为 0

年: 2019-2000=19, 对应实际数据 0x13

则返回: A6 0A 0E 42 4D 10 01 0A 00 13 05 07 E1 6A



## 5.1.5 设置、读取 MCU 软硬件版本号 (Type: 0F、10)

### MCU 设置 MCU 软硬件版本号:

- 建议: 客户更新 MCU 软硬件版本号给 BM 模块, 以便于 APP 区分统计。
- 数据格式与 BM 的软硬件版本号格式保持一致。

Byte	Value	Description	
0	0xA6	包头	
1		Payload 长度	
2	0x0F	Type: MCU 设置 MCU 软硬件版本号	Payload
3		MCU 类型: 由厂家自己定义, 可以不定义	
4		硬件版本号	
5		软件版本号	
6		年 实际年份=年+2000 例如: 2019 年 年=2019-2000=19	
7		月 1~12	
8		日 1~31	
9	Sum	校验和	
10	0x6A	包尾	

### BM 回复设置结果:

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x0F	Type: 回复设置 MCU 软硬件版本号结果	Payload
3		结果值: 0: 成功 1: 失败 2: 不支持	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

### 获取 MCU 软硬件版本号:

Byte	Value	Description	
0	0xA6	包头	
1	0x01	Payload 长度	
2	0x10	Type: 获取 MCU 软硬件版本号	Payload
3	0x11	(1~2)校验和	
4	0x6A	包尾	

### BM 返回 MCU 软硬件版本号:

深圳市易连物联网有限公司

电话: (86) 0755-81773367 FAE 邮箱: hw@elinkthings.com 销售邮箱: sale@elinkthings.com

地址: 深圳市宝安区西乡街道银田工业区侨鸿盛文化创意园写字楼 A 栋五层 502 室 邮编: 518000

Byte	Value	Description
0	0xA6	包头
1		Payload 长度
2	0x10	Type: 返回 MCU 软硬件版本号
3		MCU 类型: 由厂家自己定义
4		硬件版本号
5		软件版本号
6		年 实际年份=年+2000 例如: 2019 年 年=2019-2000=19
7		月 1~12
8		日 1~31
9	Sum	校验和
10	0x6A	包尾

### 5.1.6 设置、读取模块自动休眠时间 (Type: 17、18)

可以设置模块无数据自动进入休眠模式（低功耗模式，此时设备的 Tx 要保持为高），但是要注意的是，当模块进入（休眠模式时，MCU 在发数据前，需要提前发一组数据数据过来唤醒模块，模块才能正常开始接受数据。详可查看[工作模式说明](#)）

设置自动睡眠时间：

Byte	Value	Description
0	0xA6	包头
1		Payload 长度
2	0x17	Type: 设置无连接自动休眠时间
3		自动睡眠标志位： 0: 不开启自动休眠 1: 开启自动休眠，模块没有连接自动进入低功耗模式
4		自动睡眠时间的最高字节
5		自动睡眠时间的次高字节
6		自动睡眠时间的次低字节
7		自动睡眠时间的最低字节
8		睡眠后是否立刻断开连接，是否开启低频广播： 0: 断开连接，停止广播。 1: 保持连接，开启广播。 2: 断开连接，开启广播。 3: 保持连接，停止广播。
9		低频广播间隔时间的高字节
10		低频广播间隔时间的低字节

11	Sum	(1~10)校验和
12	0x6A	包尾

### BM 回复设置结果:

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x17	Type: 回复设置自动睡眠时间结果	Payload
3		结果值: 0: 成功 1: 失败 2: 不支持	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

### 获取自动睡眠时间:

Byte	Value	Description	
0	0xA6	包头	
1	0x01	Payload 长度	
2	0x18	Type: 获取自动睡眠时间值	Payload
3	0x19	(1~2)校验和	
4	0x6A	包尾	

### BM 返回自动睡眠时间值:

Byte	Value	Description		
0	0xA6	包头		
1	Len	Payload 长度		
2	0x18	Type: 返回无连接时自动睡眠时间	Payload	
3		自动睡眠标志位: 0: 不开启自动休眠 1: 开启自动休眠, 模块没有连接		
4		自动睡眠时间的最高字节		单位: s
5		自动睡眠时间的次高字节		
6		自动睡眠时间的次低字节		
7		自动睡眠时间的最低字节		
8		睡眠后是否开启低频广播:		

		0: 不开启 1: 开启	
9		低频广播间隔的高字节	单位 : ms ; 范围 20~2000
10		低频广播间隔的低字节	
11	Sum	(1~9)校验和	
12	0x6A	包尾	

### 5.1.7 设置模块进入睡眠 (Type: 19)

- 当 BM 模块进入休眠后,支持串口唤醒(MCU 可以发任意数据唤醒模块,或者发送唤醒指令),支持蓝牙连接唤醒 (需要开启睡眠后带广播功能,详情看下面设置进入睡眠指令格式)。

设置睡眠唤醒:

Byte	Value	Description		
0	0xA6	包头		
1	Len	Payload 长度		
2	0x19	Type: 设置进入睡眠	Payload	
3		Value: 0x01		
4		睡眠后是否断开连接, 是否开启低频广播: 0: 断开连接, 关闭广播。 1: 保持连接, 开启广播。 2: 断开连接, 开启广播。 3: 保持连接, 关闭广播。		
5		低频广播间隔时间的高字节		单位 : ms ; 范围 20~2000 ( 建议 1000ms)
6		低频广播间隔时间的低字节		
7	Sum	(1~6)校验和		
8	0x6A	包尾		

BM 回复设置结果:

Byte	Value	Description	
0	0xA6	包头	
1	Len	Payload 长度	
2	0x19	Type: 回复设置进入睡眠的结果	Payload
3		结果值: 0: 成功 (成功后 100ms 后进入睡眠) 1: 失败 2: 不支持	
4	Sum	(1~3)校验和	

5	0x6A	包尾
---	------	----

- MCU 和 APP 都可以设置 BM 模块进入睡眠, BM 模块在回复 MCU/APP 时, 同时向 APP/MCU 发送 BM 当前状态 “[BM 返回模块状态](#)”。

### 5.1.8 设置模块唤醒 (Type: 1A)

设置模块唤醒:

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x1A	Type: 设置模块唤醒	Payload
3	0x01	Value: 1: 唤醒模块	
4	0x1D	(1~3)校验和	
5	0x6A	包尾	

BM 回复设置结果:

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x1A	Type: 回复设置模块唤醒结果	Payload
3		结果值: 0: 成功 1: 失败 2: 不支持	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

### 5.1.9 设置、读取系统当前时间（Type: 1B、1C）

该时间为蓝牙模块的系统时间。

设置系统当前时间：

Byte	Value	Description	
0	0xA6	包头	
1	0x07	Payload 长度	
2	0x1B	Type: 设置系统当前时间	Payload
3		0x00 : 除能, 关闭时间功能 (默认) 0x01 : 使能, 开启时间功能	
4		年: 年份=年+2000	
5		月 (1~12)	
6		日 (1~31)	
7		时 (0~23)	
8		分 (0~59)	
9		秒 (0~59)	
10	Sum	校验和	
11	0x6A	包尾	

BM 回复设置结果：

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x1B	Type: 回复设置系统时间结果	Payload
3		结果值: 0: 成功 1: 失败 2: 不支持	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

获取系统当前时间：

Byte	Value	Description	
0	0xA6	包头	
1	0x01	Payload 长度	
2	0x1C	Type: 获取系统当前时间	Payload
3		(1~2)校验和	
4	0x6A	包尾	

**BM 返回系统当前时间:**

Byte	Value	Description
0	0xA6	包头
1	0x08	Payload 长度
2	0x1C	Type: 返回系统当前时间
3		系统时间有效位 0: 系统时间无效 1: 系统时间有效
4		年: 年份=年+2000
5		月 (1~12)
6		日 (1~31)
7		时 (0~23)
8		分 (0~59)
9		秒 (0~59)
10	Sum	(1~9)校验和
11	包尾	0x6A

Payload

### 5.1.10 设置、读取 CID、VID、PID (Type: 1D、1E)

- CID 为产品类型 ID，请按照协议透传产品类型设置（必须设。详情查看[协议透传指令产品介绍](#)）
- VID 为设备厂家 ID，请联系我司分配（选设）
- PID 为产品型号 ID，厂商自己分配，建议根据产品型号分配唯一值（选设）
- 以上三个值默认为 0，不代表任何产品（调试阶段先设置 CID）

#### 设置 ID:

Byte	Value	Description	
0	0xA6	包头	
1		Payload 长度	
2	0x1D	Type: 设置 ID	Payload
3		设置 ID 标志位 Bit0: 0 : 不设置 CID (CID 值清 0)。 1: 设置 CID Bit1: 0 : 不设置 VID (VID 值清 0)。 1: 设置 VID Bit2: 0 : 不设置 PID (PID 值清 0)。 1: 设置 PID	
4		CID: 产品类型 ID 的高字节	
5		CID: 产品类型 ID 的低字节	
6		VID: 厂商 ID 的高字节	
7		VID: 厂商 ID 的低字节	
8		PID: 产品 ID 的高字节	
9		PID: 产品 ID 的低字节	
10	Sum	(1~9)校验和	
11	0x6A	包尾	

#### BM 回复设置结果:

Byte	Value	Description	
0	0xA6	包头	
1	Len	Payload 长度	
2	0x1D	Type: 回复设置 ID 结果	Payload
3		结果值: 0: 成功 1: 失败 2: 不支持	
4	Sum	(1~3)校验和	
5	0x6A	包尾	



### 获取 ID:

Byte	Value	Description	
0	0xA6	包头	
1	0x01	Payload 长度	
2	0x1E	Type: 获取 ID 设置值	Payload
3	0x1F	(1~2)校验和	
4	0x6A	包尾	

### BM 返回 ID 值:

Byte	Value	Description	
0	0xA6	包头	
1	Len	Payload 长度	
2	0x1E	Type: 返回 ID	Payload
3		设置 ID 标志位 Bit0 : 0 : 不设置 CID。 1: 设置 CID Bit1 : 0 : 不设置 VID。 1: 设置 VID Bit2: 0 : 不设置 PID。 1: 设置 PID	
4		CID: 产品类型 ID 的高字节	
5		CID: 产品类型 ID 的低字节	
6		VID: 厂商 ID 的高字节	
7		VID: 厂商 ID 的低字节	
8		PID: 产品 ID 的高字节	
9		PID: 产品 ID 的低字节	
10	Sum	(1~9)校验和	
11	0x6A	包尾	

## 5.1.11 设置模块重启 (Type: 21)

### 设置重启模块:

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x21	Type: 设置模块重启	Payload
3	0x01	Value: 0x01	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

**BM 回复设置结果:**

Byte	Value	Description	
0	0xA6	包头	
1	Len	Payload 长度	
2	0x21	Type: 回复设置模块重启结果	Payload
3		结果值: 0: 成功 (成功后, 100ms 后模块重启) 1: 失败 2: 不支持	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

### 5.1.12 设置恢复出厂设置 (Type: 22)

**设置恢复出厂设置:**

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x22	Type: 设置恢复出厂设置	Payload
3	0x01	Value: 0x01	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

**BM 回复设置结果:**

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x22	Type: 回复设置模块重启结果	Payload
3		结果值: 0: 成功 (成功后, 100ms 后恢复出厂设置) 1: 失败 2: 不支持	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

### 5.1.13 设置、获取 BM 模块状态 (Type: 25、26)

#### 设置蓝牙连接状态

Byte	Value	Description
0	0xA6	包头
1		Payload 长度
2	0x25	Type: 设置蓝牙连接状态
3		主动断开连接标志位 1: 立刻断开连接 0: 不断开连接
4	Sum	(1~3)校验和
5	0x6A	包尾

#### BM 回复设置结果:

Byte	Value	Description
0	0xA6	包头
1	0x02	Payload 长度
2	0x25	Type: 回复设置蓝牙连接状态结果
3		结果值: 0: 成功 1: 失败 2: 不支持
4	Sum	(1~3)校验和
5	0x6A	包尾

#### 获取模块状态

Byte	Value	Description
0	0xA6	包头
1		Payload 长度
2	0x26	Type: 获取状态
3	Sum	(1~2)校验和
4	0x6A	包尾

#### BM 返回模块状态:

Byte	Value	Description
0	0xA6	包头
1		Payload 长度
2	0x26	Type: 返回模块状态
3		连接状态:

		0: 无连接 1: 已连接	
4		工作状态: 0: 唤醒 1: 进入休眠 2: 模块准备就绪	
5	Sum	(1~4)校验和	
6	0x6A	包尾	

### 5.1.14 MCU 上报 MCU 电池状态 (Type: 27、28)

#### 上报 MCU 电池状态

Byte	Value	Description	
0	0xA6	包头	
1		Payload 长度	
2	0x27	Type: 设置 MCU 电池状态	Payload
3		电池充电状态: 0: 没有充电 (默认) 1: 充电中 2: 充满电 3: 充电异常	
4		电池电量百分比 (0—100%)	
5	Sum	(1~4)校验和	
6	0x6A	包尾	

#### BM 回复 MCU 上报结果

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x27	Type: 回复 MCU 设置电池结果	Payload
3		结果值: 0: 成功 (成功后会把电池电量上传到 APP) 1: 失败 2: 不支持	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

### 查询 MCU 电池状态

Byte	Value	Description	
0	0xA6	包头	
1		Payload 长度	
2	0x28	Type: 获取 MCU 电池状态	Payload
3	Sum	(1~2)校验和	
4	0x6A	包尾	

### 返回 MCU 电池状态

Byte	Value	Description	
0	0xA6	包头	
1		Payload 长度	
2	0x28	Type: 返回 MCU 电池状态	Payload
3		电池充电状态： 0: 没有充电（默认） 1: 充电中 2: 充满电 3: 充电异常	
4		电池电量百分比（0—100%） MCU 没有数据上传时，默认为 0xFFFF	
5	Sum	(1~4)校验和	
6	0x6A	包尾	

### 5.1.15 设置使能绑定、取消使能绑定 (Type: 32)

此条指令主要用以 AILink APP 连接协议透传产品，例如蓝牙锁，MCU 开发端可以根据协议透传产品里的工作流程查看此指令的使用。（流程里无用到则说明该产品类型不需调到此指令）

#### MCU 设置指令

Byte	Value	Description	
0	0xA6	包头	
1		Payload 长度（最大 16byte）	
2	0x32	Type: MCU 设置 BM 使能、取消绑定	Payload
3		Value: 0: 取消绑定 1: 使能绑定	
4	Sum	(1~n)校验和	
5	0x6A	包尾	

#### BM 回复设置结果:

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x32	Type: BM 回复设置绑定结果	Payload
3		结果值: 0: 成功 1: 失败 2: 不支持	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

### 5.1.16 MCU 设置蓝牙锁开锁类型 (Type: 33、34)

此条指令主要用以 AILink APP 的蓝牙锁设备，MCU 端需要主动设置蓝牙锁的开锁类型。

#### MCU 设置指令

Byte	Value	Description	
0	0xA6	包头	
1		Payload 长度（最大 16byte）	
2	0x33	Type: MCU 上传 MCU 支持的蓝牙锁开锁类型	Payload
3		MCU 使用的绑定类型： MCU 可以根据自身的需求选择不同的绑定方式与 APP 进行绑定。 1: 绑定码绑定方式（需要用户输入 APP 界面上的绑	

		定码) 2: 两次按功能键确认方式 (需要用户按两次功能键) 3: 一次按功能键确认方式 (需要用户按一次功能键)	
4~5		设备支持的开锁类型: (MCU 需要声明自身所支持的开锁类型, 以方便 APP 界面的同步)  Bit =1 支持, Bit=0 不支持 Byte4 ~ byte5 Byte4_Bit 0: 按键密码 (界面) Byte4_Bit 1: 指纹密码 Byte4_Bit 2: IC 卡 Byte4_Bit 3: 蓝牙遥控器 .....	
6	Sum	(1~n)校验和	
7	0x6A	包尾	

#### BM 回复设置结果:

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x33	Type: BM 回复接收蓝牙锁支持类型结果	Payload
3		结果值: 0: 成功 1: 失败 2: 不支持	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

#### APP 读取开锁类型指令

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x34	Type: 锁的 A6 指令	Payload
3		Value: 0x01: APP 读取蓝牙锁开锁类型	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

#### BM 回复给 APP 的开锁类型

Byte	Value	Description
0	0xA6	包头

1		Payload 长度（最大 16byte）	Payload
2	0x34	Type: 锁的 A6 指令	
3		Value: 0x01: BM 上传 MCU 支持的蓝牙锁开锁类型	
4		MCU 使用的绑定类型： MCU 可以根据自身的需求选择不同的绑定方式与 APP 进行绑定。 1: 绑定码绑定方式（需要用户输入 APP 界面上的绑定码） 2: 两次按功能键确认方式（需要用户按两次功能键） 3: 一次按功能键确认方式（需要用户按一次功能键）	
5~6		设备支持的开锁类型：（MCU 需要声明自身所支持的开锁类型，以方便 APP 界面的同步）  Bit =1 支持，Bit=0 不支持 Byte5~ byte6 Byte5_Bit 0: 按键密码（界面） Byte5_Bit 1: 指纹密码 Byte5_Bit 2: IC 卡 Byte5_Bit 3: 蓝牙遥控器 .....	
7	Sum	(1~n)校验和	
8	0x6A	包尾	

### 5.1.17 MCU 上传设备基本信息（Type: 35、36）

此条指令主要用 MCU 上传设备的一些基本的信息，用于 APP 端数据的同步，详情请查看对应的协议透传产品介绍。

#### MCU 设置设备基本信息指令

Byte	Value	Description	
0	0xA6	包头	
1	0x10	Payload 长度	
2	0x35	Type: MCU 上传设备的基本信息	Payload
3	0x01	数据有效标志位 0x01	
4~17		数据	
18	Sum	(1~n)校验和	
19	0x6A	包尾	



### BM 回复设置结果:

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x35	Type: BM 回复结果	Payload
3		结果值: 0: 成功 1: 失败 2: 不支持	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

### 读取设备基本信息指令

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x36	Type: 读取设备的基本信息指令	Payload
3		Value 0x01	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

### BM 返回设备基本信息指令

Byte	Value	Description	
0	0xA6	包头	
1	0x10	Payload 长度	
2	0x36	Type: MCU 上传设备的基本信息	Payload
3	0x01	数据有效标志位 0x01	
4~17		数据	
18	Sum	(1~n)校验和	
19	0x6A	包尾	

### 5.1.18 APP 同步时间到 MCU (Type: 37、38)

对于某些设备，具有时间功能的，此时，可利用此指令进行数据的同步。

#### ● APP 下发时间。

Byte	Default	Description
0	0xA6	包头
1		Payload 长度 (最大 15byte)
2	0x37	Type: APP 同步时间
3~9		时间: 7 个 byte 年 (当前年份-2000) 月 日 时 分 秒 星期 (1~7 1=周一 ~ 7=周日)
10	SUM (1~n)	(1~n)校验和
11	0x6A	包尾

#### ● MCU 返回同步时间结果

Byte	Value	Description
0	0xA6	包头
1	0x02	Payload 长度
2	0x37	Type: MCU 返回时间同步结果
3		结果值: 0: 成功 1: 失败 2: 不支持
4	Sum	(1~3)校验和
5	0x6A	包尾

#### ● MCU 请求时间

设备有时间功能，且在与 APP 连接状态时，可以请求时间更新，APP 收到该请求，会下发时间同步。

Byte	Value	Description
0	0xA6	包头
1	0x02	Payload 长度
2	0x38	Type: MCU 请求 APP 下发时间
3		Value 0x01

4	Sum	(1~3)校验和
5	0x6A	包尾

### 5.1.19 BM 模块自动唤醒设置、自动进入睡眠返回 (Type: 3A、3B)

当 BM 模块处于休眠状态时，BM 模块连接、断连、收发数据时的唤醒设置。当 BM 模块处于唤醒状态时，不会触发唤醒设置机制。

带 flash 的模块，该数据断电保存。

#### ● MCU 设置。

Byte	Default	Description	
0	0xA6	包头	
1		Payload 长度 (最大 15byte)	
2	0x3A	Type: BM 模块唤醒设置	Payload
3		连接唤醒: (BM 模块从断连状态切为连接状态时) 0x00: APP 连接时, 不唤醒 BM 模块和 MCU。 0x01: APP 连接时, 唤醒 BM 模块和 MCU。(默认)	
4		断连唤醒: (BM 模块从连接状态切为断连状态时) 0x00: APP 断连时, 不唤醒 BM 模块和 MCU。(默认) 0x01: APP 断连时, 唤醒 BM 模块和 MCU。	
5		收数据唤醒: (BM 收到 APP 数据, 同时需要把数据发到 MCU 时) 0x00: 收到 APP 数据时, 不唤醒 BM 和 MCU 0x01: 收到 APP 数据时, 唤醒 BM 和 MCU (默认)	
6		自动睡眠返回指令: 0x00: 自动睡眠后, 不返回睡眠指令。 0x01: 自动睡眠后, 返回睡眠指令。(默认)	
7	SUM (1~n)		
8	0x6A	包尾	

#### ● BM 返回设置结果

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x3A	Type: MCU 返回设置结果	Payload
3		结果值: 0: 成功 1: 失败	

		2: 不支持	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

## 5.2 协议透传指令

根据已定好的协议，做数据的传输。

传输格式：

Byte	Default	Description
0	0xA7	包头
1~2		产品类型 CID
3		Payload 长度（最大 15byte）
n		Payload
n+1	SUM (1~n)	(1~n)校验和
n+2	0x7A	包尾

协议透传指令的识别：

包头和包尾是固定的：0xA7，0x7A。

校验和是指 byte1 + ... + byte n 的值，取低 8 位。

## 5.3 数据透传

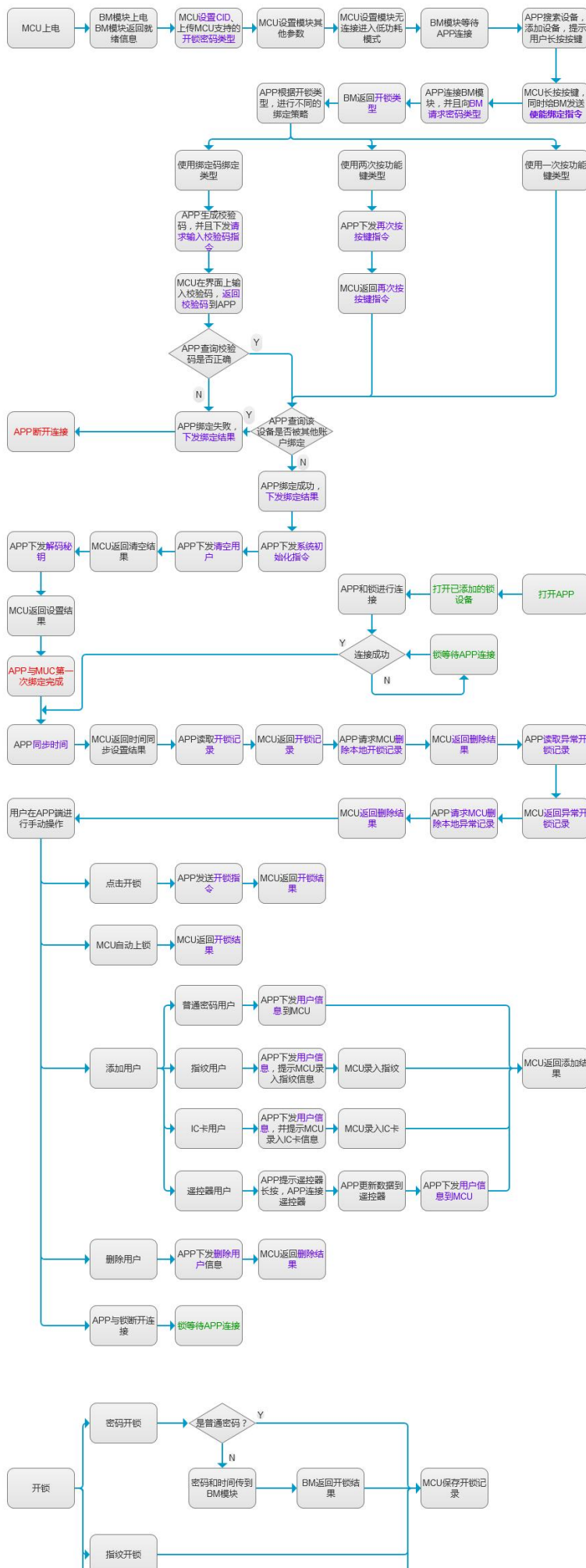
不符合设置指令与协议透传指令的数据一律采用数据透传，即收到什么数据就传什么数据。

# 6 协议透传产品介绍

产品编号	产品名称	产品类型（CID）
9	蓝牙门锁	000B

## 6.1 蓝牙门锁

### 6.1.1 工作流程



## 6.1.2 获取蓝牙门锁的开锁类型

- APP 需要获取蓝牙锁拥有的开锁类型，从而确定绑定方式以及功能使用。例如，无输入密码型的锁的绑定方式不一样。
- MCU 端需要主动将开锁类型上报到 BM 进行保存。数据格式 [点击查看蓝牙锁开锁类型](#)

## 6.1.3 APP 绑定蓝牙锁流程

- 蓝牙锁绑定码绑定方式
  - 1、打开 APP，搜索蓝牙锁设备
  - 2、APP 搜索到蓝牙锁后提示用户长按设备上的按键
  - 3、用户长按按键后，蓝牙锁与 APP 进行连接
  - 4、APP 生成绑定码，提示用户在蓝牙锁上输入
  - 5、用户输入校验码后，MCU 将校验码发送到 APP 上，APP 收到校验码，判断校验码是否正确。
  - 6、校验码正确后，APP 需要从服务器里此锁是否已被绑定。
  - 7、若此锁被绑定则绑定失败。未绑定则绑定成功。
- 蓝牙锁按两次功能键绑定方式
  - 1、打开 APP，搜索蓝牙锁设备
  - 2、APP 搜索到蓝牙锁后提示用户长按设备上的按键
  - 3、用户长按按键后，蓝牙锁与 APP 进行连接
  - 4、APP 与蓝牙锁连接成功后，提示用户再次按按键确认绑定
  - 5、APP 收到确认后，APP 需要从服务器里寻找此锁是否已被绑定。
  - 6、若此锁被绑定则绑定失败。未绑定则绑定成功。
- 蓝牙锁按一次功能键绑定方式
  - 1、打开 APP，搜索蓝牙锁设备
  - 2、APP 搜索到蓝牙锁后提示用户长按设备上的按键
  - 3、用户长按按键后，蓝牙锁与 APP 进行连接
  - 4、APP 连接后，APP 需要从服务器里寻找此锁是否已被绑定。
  - 5、若此锁被绑定则绑定失败。未绑定则绑定成功。

## 6.1.4 APP 添加用户流程

- 1、APP 连接蓝牙锁。
- 2、APP 将需要添加的信息发到蓝牙锁里。
- 3、蓝牙锁回复添加成功。

### 6.1.5 APP 分享用户使用

APP 可以通过分享给以其他用户使用，但此用户只有开锁权限。

### 6.1.6 一次性密码生成与使用

1、一次性密码生成

- APP 在和蓝牙锁在绑定成功后，APP 会随机生成一个密钥，发送到 BM 上。BM 会利用此密钥检测一次性密码的合法性。

2、当一次性被使用后，BM 端需要删除此密码。

3、当用户再界面上输入密码时，如果此密码非普通的密码，则可能是一次性密码，则 MCU 需要将此密码以及当前时间发送到 BM 模块上，让 BM 模块进行密码校验。

### 6.1.7 响铃功能

在用户按响铃功能按键时，当蓝牙没连接时，BM 通过改变广播格式，iBeacon 广播和 BLE 广播交替发，从而唤醒 APP 打开 APP 连接蓝牙锁且响铃。当已连接状态，直接响铃。

### 6.1.8 开关锁和异常操作的记录的

MCU 端需要保存正常开锁的数据，包括开锁的时间，类型（APP 开锁不用记录）等，同时也需要记录异常数据，例如输入错误密码等，以便于 APP 读取记录。

### 6.1.9 用户 ID 编号说明

用户 ID 编号有 2 个 byte（16bit）数据构成，高 4bit 表明 ID 的属性，低 12bit 表明 ID 的编号。

用户 ID 编号（16 bit）															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
属性				编号											
0: 锁端添加的管理员用户				主要用以用户编号的区分以及操作界面上的显示。 (编号=0 为 APP 专属编号)											
1: 锁端添加的普通用户															
2: APP 端添加的普通用户															



## 6.1.10 数据指令

编号	操作	内容	包头	数据类型	长度	命令	数据	校验和	包尾	
一、 使用 绑定 码的 绑定 方式	MCU 上传	MCU 长按按键	0xA6	数据格式查看 <a href="#">设置使能、取消使能绑定</a>						0x6A
	BM 回应	收到按键结果	0xA6	数据格式查看 <a href="#">设置使能、取消使能绑定</a>						
	APP 下发	提示用户输入 校验	0xA7	0x000B	0x02	0x02	0x01		0x7A	
	MCU 应答	回应校验码	0xA7	0x000B	0x07	0x02	6 个 byte 校验码		0x7A	
	APP 下发	绑定结果			0x02	0x03	0: 设备绑定码正确, 绑定成功 1: 设备绑定码错误, 绑定失败 2: 设备已经被其他用户 绑定, 绑定失败		7A	
使用 两次 按 功 能 的 绑 定 方 式	MCU 上传	MCU 长按按键	A6	数据格式查看 <a href="#">设置使能、取消使能绑定</a>						6A
	BM 回应	收到按键结果	A6	数据格式查看 <a href="#">设置使能、取消使能绑定</a>						6A
	APP 下发	提示用户再次 按按键	A7		0x02	0x02	0x02		7A	
	MCU 回应	再次按按键	A7		0x02	0x02	0x02		7A	
	APP 下发	绑定结果	A7		0x02	0x03	0: 绑定成功 1: 绑定失败 2: 设备已经被其他用户 绑定, 绑定失败			
使用 一 次 按 功 能 的 方 式	MCU 上传	MCU 长按按键	A6	数据格式查看 <a href="#">设置使能、取消使能绑定</a>						6A
	BM 回应	收到按键结果	A6	数据格式查看 <a href="#">设置使能、取消使能绑定</a>						6A
	APP 下发	绑定结果	A7		0x02	0x03	0: 绑定成功 1: 绑定失败 2: 设备已经被其他用户 绑定, 绑定失败		7A	
二	APP 下发	开、关锁	A7		0x05	0x04	用户类型 (1byte) : 1: APP 操作 2: BLE 遥控器操作 + 操作内容 (1byte) : 0: 开锁 1: 关锁  用户编号 ID (2byte)		7A	
	MCU 应答				0x03	0x04	操作内容: 0: 开锁 1: 关锁 +			

							操作结果： 0：成功 1：失败		
三	APP 下发	添加用户数据帧 1			0x09	0x05	数据帧 1 + 用户类型 + 用户编号 ID (2byte) + 用户密码 (3byte) + 密码有效期类型		
	APP 下发	添加用户数据帧 2			N	0x05	数据帧 2 + 密码有效时间		
	MCU 回应	MCU 回复添加结果				0x05	添加结果		
四	APP 下发	删除用户			0x04	0x07	用户类型： 0：普通密码用户 1：指纹用户 2：IC 卡用户 3：遥控器 + 用户编号 ID (2byte)		
	MCU 回应				0x02	0x07	删除结果 0：成功 1：失败		
五	APP 下发	清空用户			0x02	0x08	0x01		
	MCU 回应				0x02	0x08	清空结果 0：清空成功 1：清空失败		
六	APP 下发	系统初始化			0x02	0x09	0x01		
	MCU 回应				0x02	0x09	0：初始化成功 1：初始化失败		
七	APP 下发	读取开锁记录			0x02	0x0a	0x01		
	MCU 回应				0x0E	0x0a	记录总条数 (2byte) + 当前序号 (2byte) + 用户类型 +		

							用户编号 ID (2byte) + 开锁时间 (6byte)			
八	APP 下发	读取异常开锁记录			0x02	0x0b	0x01			
	MCU 回应				0x0B	0x0b	记录异常总条数 (2byte) + 当前序号 (2byte) + 异常操作用户类型 + 异常开锁时间 (6byte)			
九	APP 下发	同步时间			0x08	0x0c	时间 (7 个 byte) 年+月+日+时+分+秒+星期			
	MCU 回应					0x0c	同步结果 0: 成功 1: 失败			
十	APP 下发	同步用户			0x02	0x0d	0x01			
	MCU 回应	每组用户第一帧			0x09	0x0d	数据帧 1 + 用户类型 + 用户编号 ID (2byte) + 用户密码 (3byte) + 密码类型			
		每组用户第二帧			N	0x0d	数据帧 2 + 密码有效时间			
		全部用户同步完成时回应			0x02	0x0d	0xFF			
十一	APP 下发	获取 MCU 支持的密码类型	数据格式查看 <a href="#">蓝牙锁开锁类型</a>							
	MCU 回应		数据格式查看 <a href="#">蓝牙锁开锁类型</a>							
十二	MCU 上传	上传一次性密码				0x0f	密码 (3byte) + 时间 (6byte)			
十三	APP 下发	操作开锁、异常开锁记录				0x10	操作内容: 0x00: 开锁记录			

						0x01: 异常开锁记录 + 操作类型: 0x00: 删除 MCU 端历史记录 0x01: 取消读取记录		
	MCU 回应					0x10 操作内容: 0x00: 开锁记录 0x01: 异常开锁记录 + 操作类型: 0x00: 删除 MCU 端历史记录 0x01: 取消读取历史记录 + 结果: 0x00: 成功 0x01: 失败		
十四	APP 下发	查询锁当前开 关锁状态				0x11		
	MCU 回应					0x11		

### 6.1.10.1 MCU 上传门锁类型

MCU 在开机后，主动上传门锁的类型（只需上传一次）。

#### MCU 设置设备基本信息指令

Byte	Value	Description	
0	0xA6	包头	
1	0x10	Payload 长度	
2	0x35	Type: MCU 上传设备的基本信息	Payload
3	0x01	数据有效标志位 0x01	
4		类型: 1: 全自动 2: 半自动	
5~17		0	
18	Sum	(1~n)校验和	
19	0x6A	包尾	

## 6.1.10.2 APP 绑定锁

1、APP 进入绑定页面，提示用户长按按键进行绑定操作。  
 用户长按绑定按键，MCU 发送绑定信息到 BM 模块。

数据格式查看 [设置使能、取消使能绑定](#)

### BM 回复 MCU

数据格式查看 [设置使能、取消使能绑定](#)

### 使用绑定码的方式 指令集 (2、3、6)

2、APP 连接到 BM 模块后，会发送提示指令到 MCU，要求其输入屏幕上的绑定码。

#### ● APP 下发指令提示用户输入绑定码

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型： 0x000B	
3		Payload 长度（最大 15byte）	
4	0x02	Type: 提示用户发送绑定码	Payload
5		Value 0x01	
6	SUM (1~n)	(1~n)校验和	
7	0x7A	包尾	

1、MCU 将界面输入的绑定码返回到 APP。

#### ● MCU 输入 APP 上显示的绑定码，并且传输到 APP 上。

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型： 0x000B	
3		Payload 长度（最大 15byte）	
4	0x02	Type: 用户发送绑定码	Payload
5~10		6 位绑定码号 例如：若绑定号为 12345 则：byte5=0x01，byte6=0x02，byte7=0x03，byte8=0x04， byte9=0x05，byte10=0x06，	
11	SUM (1~n)	(1~n)校验和	
12	0x7A	包尾	

## 使用两次按功能键的 指令集(4、5、6)

2、APP 连接到 BM 模块后，会用户按下功能按键来确认绑定。

- APP 下发指令提示用户输入绑定码

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型: 0x000B	
3		Payload 长度 (最大 15byte)	
4	0x02	Tpye: 提示用户按功能按键	Payload
5		Value 0x02	
6	SUM (1~n)	(1~n)校验和	
7	0x7A	包尾	

3、MCU 将界面输入的绑定码返回到 APP。

- MCU 输入 APP 上显示的绑定码，并且传输到 APP 上。

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型: 0x000B	
3		Payload 长度 (最大 15byte)	
4	0x02	Tpye: 用户发送绑定码	Payload
5		Value: 0x02	
6	SUM (1~n)	(1~n)校验和	
7	0x7A	包尾	

## 使用一次按功能键的 指令集(6)

4、APP 返回绑定码验证结果

- 绑定码正确，并且该设备还没被其他用户绑定，才能绑定成功。

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型: 0x000B	
3		Payload 长度 (最大 15byte)	
4	0x03	Tpye: APP 返回绑定码结果	Payload
5		0: 设备绑定码正确，绑定成功	

		1: 设备绑定码错误, 绑定失败 2: 设备已经被其他用户绑定, 绑定失败	
6	SUM (1~n)	(1~n)校验和	
7	0x7A	包尾	

### 6.1.10.3 APP、MCU 下发开/关锁

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型: 0x000B	
3		Payload 长度 (最大 15byte)	
4	0x04	Tpye: APP 下发开关锁	Payload
5		操作的用户: 1: APP 操作 (用户编号 ID 设为 0) 2: BLE 遥控器操作 3: 一次性密码开锁 (MCU 上传一次性密码后, BM 下发开锁)	
6~7		用户编号 ID: 2 个 byte 例如 ID=0x0012, 则 Byte6=0x00, Byte7=0x12	
8		0: 开锁 1: 关锁 0xff: 密码错误 (此标识主要用于一次性密码校验后异常返回)	
9	SUM (1~n)	(1~n)校验和	
10	0x7A	包尾	

- MCU 回复开关锁结果
- 此指令可以在 APP 查询时回复, 或者锁状态发生变化时, 主动上报。

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型: 0x000B	
3		Payload 长度 (最大 15byte)	
4	0x04	Tpye: MCU 回复开关锁	Payload
5		操作内容: 0: 开锁 1: 关锁 2: 开锁中 3: 关锁中	
6		操作结果:	

		0: 成功 1: 失败	
7	SUM (1~n)	(1~n)校验和	
8	0x7A	包尾	

### 6.1.10.4 APP 添加用户

- APP 申请 MCU 添加用户。  
由于指令长度问题，分两组数据发送  
第一组数据：

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型： 0x000B	
3		Payload 长度（最大 15byte）	
4	0x05	Type: APP 申请添加用户	Payload
5	0x01	数据组编号： 1	
6		添加类型： 0: 普通密码用户 1: 指纹用户 2: IC 卡用户 3: 遥控器 ..... 255:一次性密码	
7~8		用户 ID 编号（两个 byte） 如 ID=0x0123，则 byte7=0x01，byte8=0x23； 若编号已存在，则覆盖。 0xFFFF 为无编号，MCU 端自定义编号。	
9~11		用户密码（3 个 byte 组成的数据，使用低 6 位作为有效数字， <b>普通密码类型才会用到，其他用户类型使用 0xFFFFFFFF</b> ） 例如：数据 byte9=01,byte10=E6,byte11=A2 数据=0x01E6A2=124578 那么密码就是：124578  数据=0x9227B9=9578425 那么密码就是：578425  数据=0x8E=142	



		那么密码就是：000142	
12		密码有效期类型： 1：24小时有效 2：7天有效 3：永久有效 4：自定义时间段 5：保姆密码	
13	SUM (1~n)	(1~n)校验和	
14	0x7A	包尾	

第二组数据：

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型： 0x000B	
3		Payload 长度（最大 15byte）	
4	0x05	Tpyc: APP 确认添加用户	Payload
5	0x02	数据组编号： 2	
		<p>当设置有效期为 1~4 类型时（24 小时有效、7 天有效、永久有效、自定义时间段），共 10 个 byte</p> <p>5 个 byte 表示时间段起始</p> <p>年：年=2019-2000</p> <p>月：（1~12）</p> <p>日：（0~31）</p> <p>时：（0~23）</p> <p>分：（0~59）</p> <p>5 个 byte 表示时间段结束</p> <p>年：年=2019-2000</p> <p>月：（1~12）</p> <p>日：（0~31）</p> <p>时：（0~23）</p> <p>分：（0~59）</p> <p>当起始时间段为 00 00 00 00 00，结束时间段为 FF FF FF FF FF 时，表示时间段无限制。</p> <p>当设置有效期为类型 5 时（保姆密码）</p> <p>2 个 byte 时间段起始：</p> <p>Byte 6=时（0~23）</p> <p>byte 7=分（0~59）</p> <p>2 个 byte 为时间段结束</p>	

		<p>Byte 8=时 (0~23)</p> <p>Byte9 =分 (0~59)</p> <p>重复规则:</p> <p>Byte10</p> <p>=1 : 每天</p> <p>=2: 每周</p> <p>=3: 每月</p> <p>当 byte10=2 (每周) 时:</p> <p>Byte11:</p> <p>Bit0~Bit6 =周 1~周日</p> <p>Bit =0 : 当天不能开</p> <p>Bit=1 : 当天能开</p> <p>当 byte10=3 (每月) 时:</p> <p>Byte 11: bit0~bit6 =1 号 ~ 7 号</p> <p>Byte 12: bit0~bit6 =8 号~14 号</p> <p>Byte13: bit0~bit6 =15 号~21 号</p> <p>Byte 14: bit0~bit6 =22 号 ~28 号</p> <p>Byte 15: bit0~bit2=29 号~31 号</p> <p>Bit7 = 每月的最后一天</p> <p>Bit=0 : 当天不能开</p> <p>Bit=1 : 当天能开</p>	
	SUM (1~n)	(1~n)校验和	
	0x7A	包尾	

### ● MCU 回复申请添加结果

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型: 0x000B	
3		Payload 长度 (最大 15byte)	
4	0x05	Tpye: MCU 回复添加结果	Payload
5		回复结果: 0: 添加成功 1: 添加失败 (需要在状态里说明失败原因) 2: 添加中 (例如添加指纹用户或者 IC 卡用户等, 需要分步进行时可使用到)	

6		用户类型： 0: 普通密码用户 1: 指纹用户 2: IC 卡用户 3: 遥控器 ..... 255:一次性密码	
7~8		用户 ID 编号（2 个 byte） 如 ID=0x0123，则 byte7=0x01，byte8=0x23； <b>需根据 ID 编号说明格式来定义。</b>	
9		状态 state： <ul style="list-style-type: none"> <li>● 如果结果是添加成功时，该值为 state = 0</li> <li>● 如果结果是添加失败时，该值为失败原因： State = 1: 密码相似 2: 找不到指纹 3: 找不到 IC 卡 .... 0xFF :其他异常</li> <li>● 如果结果是添加中时，该值为对应的步骤，例如指纹的步骤，以方便 APP 同步做提示（如果需要提示用户换位置时，State=0xFF）。</li> </ul> State= 1~N	
10		指纹录入百分比：0%~100% 数值 0~100 （其他用户类型则填 0）	
11	SUM（1~n）	(1~n)校验和	
12	0x7A	包尾	

- APP 下发添加结果
- 当 APP 确认能够添加的话，就会返回这个确认指令。

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型： 0x000B	
3		Payload 长度（最大 15byte）	
4	0x05	Tpye: APP 确认添加用户	Payload

5	0x03	数据组编号： 3	
6		结果： 0：添加成功 1：添加失败	
7	SUM (1~n)	(1~n)校验和	
8	0x7A	包尾	

### 6.1.10.5 APP 取消添加用户

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型： 0x000B	
3		Payload 长度（最大 15byte）	
4	0x06	Tpye: APP 取消添加用户	Payload
5		操作内容： 1：取消添加用户	
7	SUM (1~n)	(1~n)校验和	
8	0x7A	包尾	

#### ● MCU 回复取消添加结果。

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型： 0x000B	
3		Payload 长度（最大 15byte）	
4	0x06	Tpye: MCU 回复取消添加结果	Payload
5		回复结果： 0：取消成功 1：取消失败	
6	SUM (1~n)	(1~n)校验和	
7	0x7A	包尾	

## 6.1.10.6 APP 下发删除用户

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型: 0x000B	
3		Payload 长度 (最大 15byte)	
4	0x07	Tpye: APP 删除用户	Payload
5		用户类型: 0: 普通密码用户 1: 指纹用户 2: IC 卡用户 3: 遥控器	
6~7		用户编号 (2 byte) 如 ID=0x0123, 则 byte6=0x01, byte7=0x23; (若用户编号为 0xFFFF, 则表示删除同一类型的用户。)	
8		提示 Flag 0: 设备需语音调试 (人为在 APP 界面上操作删除) 1: 设备不需语音调试 (异常需删除)	
9	SUM (1~n)	(1~n)校验和	
10	0x7A	包尾	

### ● MCU 返回删除结果

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型: 0x000B	
3		Payload 长度 (最大 15byte)	
4	0x07	Tpye: MCU 返回删除用户结果	Payload
5		返回结果: 0: 成功 1: 失败	
6	SUM (1~n)	(1~n)校验和	
7	0x7A	包尾	

### 6.1.10.7 APP 下发清空用户

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型: 0x000B	
3		Payload 长度 (最大 15byte)	
4	0x08	Tpye: APP 清空用户	Payload
5		Value 0x01	
6	SUM (1~n)	(1~n)校验和	
7	0x7A	包尾	

#### ● MCU 返回清空结果

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型: 0x000B	
3		Payload 长度 (最大 15byte)	
4	0x08	Tpye: MCU 返回清空用户结果	Payload
5		返回结果: 0: 成功 1: 失败	
6	SUM (1~n)	(1~n)校验和	
7	0x7A	包尾	

### 6.1.10.8 APP 下发系统初始化

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型: 0x000B	
3		Payload 长度 (最大 15byte)	
4	0x09	Tpye: APP 初始化系统	Payload
5		Value 0x01	
6	SUM (1~n)	(1~n)校验和	
7	0x7A	包尾	

● MCU 返回系统初始化结果

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型: 0x000B	
3		Payload 长度 (最大 15byte)	
4	0x09	Tpye: MCU 返回初始化结果	Payload
5		返回结果: 0: 成功 1: 失败	
6	SUM (1~n)	(1~n)校验和	
7	0x7A	包尾	

### 6.1.10.9 APP 读取开锁记录

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型: 0x000B	
3		Payload 长度 (最大 15byte)	
4	0x0A	Tpye: APP 读取开锁记录	Payload
5		Value 0x01	
6	SUM (1~n)	(1~n)校验和	
7	0x7A	包尾	

● MCU 返回开锁记录

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型: 0x000B	
3		Payload 长度 (最大 15byte)	
4	0x0A	Tpye: MCU 返回开锁记录	Payload
5~6		记录总条数 (两个 byte) 如总条数=0x0123, 则 byte5=0x01, byte6=0x23;	
7~8		当前序号 (0 ~ 总条数-1) 两个 byte 如当前序号=0x0123, 则 byte7=0x01, byte8=0x23;	
9		用户类型: 0: 普通密码用户	

		1: 指纹用户 2: IC 卡用户 3: 遥控器 .... 255: 一次性密码（一次性密码是无用户编号 ID 的，所有的一次性密码定义的 ID 都为 0xFFFE）	
10~11		用户编号 ID（2 个 byte）	
12~17		开锁时间（6byte） 年（当前年份-2000） 月 日 时 分 秒	
18	SUM（1~n）	(1~n)校验和	
19	0x7A	包尾	

### 6.1.10.10 APP 读取异常开锁记录

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型： 0x000B	
3		Payload 长度（最大 15byte）	
4	0x0B	Tpye: APP 读取异常记录	Payload
5		Value 0x01	
6	SUM（1~n）	(1~n)校验和	
7	0x7A	包尾	

#### ● MCU 返回异常开锁记录

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型： 0x000B	
3		Payload 长度（最大 15byte）	
4	0x0B	Tpye: MCU 返回异常开锁记录	Payload
5~6		记录异常总条数（两个 byte） 如总条数=0x0123，则 byte5=0x01，byte6=0x23；	
7~8		当前序号（0 ~ 总条数-1）两个 byte	



		如序号=0x0123, 则 byte7=0x01, byte8=0x23;	
9		异常操作类型: 0: 普通密码 1: 指纹操作 2: IC 卡用户 3: 遥控器开锁 ....	
10~15		异常开锁时间 (6byte) 年 月 日 时 分 秒	
16	SUM (1~n)	(1~n)校验和	
17	0x7A	包尾	

### 6.1.10.11 APP 同步时间

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型: 0x000B	
3		Payload 长度 (最大 15byte)	
4	0x0C	Tpye: APP 同步时间	Payload
5~11		时间: 7 个 byte 年 (当前年份-2000) 月 日 时 分 秒 星期 (1~7 1=周一 ~ 7=周日)	
12	SUM (1~n)	(1~n)校验和	
13	0x7A	包尾	

#### ● MCU 返回同步时间结果

Byte	Default	Description
0	0xA7	包头
1~2	0x000B	产品类型:

		0x000B	
3		Payload 长度（最大 15byte）	
4	0x0C	Tpye: MCU 返回同步时间结果	Payload
5		Value 0: 成功 1: 失败	
6	SUM (1~n)	(1~n)校验和	
7	0x7A	包尾	

### 6.1.10.12 APP 同步 MCU 端用户

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型: 0x000B	
3		Payload 长度（最大 15byte）	
4	0x0D	Tpye: APP 同步 MCU 端用户	Payload
5		Value 0x01 : 同步用户 0x02 : 取消同步用户（MCU 需返回同步完成指令）	
6	SUM (1~n)	(1~n)校验和	
7	0x7A	包尾	

- **MCU 回复用户数据（若收到的是同步用户，则返回同步信息）**

每组用户分 2 帧数据回复，如果用户数据同步完成，需返回同步完成指令，若没用用户，则直接回复同步完成指令。

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型: 0x000B	
3		Payload 长度（最大 15byte）	
4	0x0D	Tpye: MCU 回复用户数据	Payload
5~6		总用户数量（大端序）	
7	0x01	数据组编号: 第一帧: 1	
8		用户类型: 0: 普通密码用户 1: 指纹用户 2: IC 卡用户	

		3: 遥控器	
9~10		用户 ID 编号 (两个 byte) 如 ID=0x0123, 则 byte7=0x01, byte8=0x23;	
11~13		用户密码 (3 个 byte 组成的数据, 使用低 6 位作为有效数字, <b>普通密码类型才会用到, 其他用户类型使用 0xFFFFFFFF</b> ) 例如: 数据 byte9=0x01, byte10=0xE6, byte11=0xA2 数据=0x01E6A2=124578 那么密码就是: 124578  数据=0x9227B9=9578425 那么密码就是: 578425  数据=0x8E=142 那么密码就是: 000142	
14		密码有效期类型: 1 : 24 小时有效 2 : 7 天有效 3 : 永久有效 4 : 自定义时间段 5 : 保姆密码	
15	SUM (1~n)	(1~n)校验和	
16	0x7A	包尾	

### 第二组数据:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型: 0x000B	
3		Payload 长度 (最大 15byte)	
4	0x0D	Type: MCU 回复用户数据	Payload
5~6		<b>总用户数量 (大端序)</b>	
7	0x02	数据组编号: 第二帧: 2	
		当设置有效期为 1~4 类型时 (24 小时有效、7 天有效、永久有效、自定义时间段), 共 10 个 byte 5 个 byte 表示时间段起始 年: 年=2019-2000 月: (1~12) 日: (0~31) 时: (0~23) 分: (0~59)	

		<p>5 个 byte 表示时间段结束            年: 年=2019-2015=4            月: (1~12)            日: (0~31)            时: (0~23)            分: (0~59)            当起始时间段为 00 00 00 00 00 , 结束时间段为 FF FF FF FF FF 时, 表示时间段无限制。</p> <p><b>当设置有效期为类型 5 时 (保姆密码)</b>            2 个 byte 时间段起始:            Byte 8=时 (0~23)            Byte 9 =分 (0~59)            2 个 byte 为时间段结束            Byte 10=时 (0~23)            Byte 11 =分 (0~59)</p> <p>重复规则:            Byte12            =1 : 每天            =2: 每周            =3: 每月</p> <p>当 byte12=1 (每天) 时:            byte13~byte17 为 0</p> <p>当 byte12=2 (每周) 时:            Byte13:            Bit0~Bit6 =周 1~周日            Bit =0 : 当天不能开            Bit=1 : 当天能开            (byte14~byte17 为 0)</p> <p>当 byte12=3 (每月) 时:            Byte 13: bit0~bit6 =1 号 ~ 7 号            Byte 14: bit0~bit6 =8 号~14 号            Byte15: bit0~bit6 =15 号~21 号            Byte 16: bit0~bit6 =22 号 ~28 号            Byte 17: bit0~bit2=29 号~31 号      Bit7 = 每月的最后一天</p> <p>Bit=0 : 当天不能开</p>	
--	--	---	--

		Bit=1 : 当天能开	
	SUM (1~n)	(1~n)校验和	
	0x7A	包尾	

### 同步完成指令

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型: 0x000B	
3		Payload 长度 (最大 15byte)	
4	0x0D	Tpye: MCU 回复用户数据	Payload
5		数据组编号: 0xFF (用户同步完成) 0xFE (取消同步)	
6	SUM (1~n)	(1~n)校验和	
7	0x7A	包尾	

### 6.1.10.13 APP 获取 MCU 支持的开锁密码类型

数据格式 [点击查看蓝牙锁开锁类型](#)

### 6.1.10.14 MCU 主动上传支持的开锁密码类型

数据格式 [点击查看蓝牙锁开锁类型](#)

### 6.1.10.15 MCU 上传一次性密码和当前时间

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型: 0x000B	
3		Payload 长度 (最大 15byte)	
4	0x0F	Tpye: MCU 上传密码	Payload
5~7		一次性密码 (3 个 byte 组成的数据, 使用低 6 位作为有效数字) 例如: byte5=0x01; byte6=0xE6; byte7=0xA6 则, 数据=0x01E6A2=124578	

		那么密码就是：124578  数据=0x9227B9=9578425 那么密码就是：578425  数据=0x8E=142 那么密码就是：000142	
8~13		时间：6 个 byte 年（当前年份-2000） 月 日 时 分 秒	
14	SUM (1~n)	(1~n)校验和	
15	0x7A	包尾	

### 6.1.10.16 APP 操作开锁\异常开锁历史记录

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型： 0x000B	
3		Payload 长度（最大 15byte）	
4	0x10	Tpye: APP 操作 开锁\异常开锁 记录	Payload
5		操作内容： 0x00: 开锁记录 0x01: 异常开锁记录	
6		操作类型： 0x00: 删除 MCU 端的记录 0x01: 取消读取 MCU 端的记录	
7	SUM (1~n)	(1~n)校验和	
8	0x7A	包尾	

#### ● MCU 回复操作开锁历史结果

Byte	Default	Description
0	0xA7	包头
1~2	0x000B	产品类型：

		0x000B	
3		Payload 长度（最大 15byte）	
4	0x10	Tpye: MCU 回复操作开锁记录	Payload
5		操作内容: 0x00: 开锁记录 0x01: 异常开锁记录	
6		操作类型: 0x00: 删除 MCU 端的历史记录 0x01: 取消读取 MCU 端的历史记录	
7		操作结果: 0x00: 成功 0x01: 失败	
8	SUM (1~n)	(1~n)校验和	
9	0x7A	包尾	

### 6.1.10.17 APP 查询锁的当前状态

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型: 0x000B	
3		Payload 长度（最大 15byte）	
4	0x11	Tpye: APP 查询当前锁的状态	Payload
5		Value : 0x01	
6	SUM (1~n)	(1~n)校验和	
7	0x7A	包尾	

- 当 MCU 收到此指令时，就返回当前锁的状态。

- **MCU 回复锁当前状态**

Byte	Default	Description	
0	0xA7	包头	
1~2	0x000B	产品类型: 0x000B	
3		Payload 长度（最大 15byte）	
4	0x11	Tpye: MCU 回复锁当前状态记录	Payload
5		State: 0: 已开锁状态 1: 正在开锁状态	

		2: 已上锁状态 3: 正在上锁状态	
6	SUM (1~n)	(1~n)校验和	
7	0x7A	包尾	



## 6.2 使用/测试指导

## 6.3 测试工具

## 6.4 测试步骤

# 7 联系我们

深圳市易连物联网有限公司

地址：深圳市宝安区西乡街道银田工业区侨鸿盛文化创意园写字楼 A 栋五层 502 室

Tel: + (86) 0755-81773367

市场部邮箱：[marketing@elinkthings.com](mailto:marketing@elinkthings.com)

FAE 邮箱：[hw@elinkthings.com](mailto:hw@elinkthings.com)

官网：[www.elinkthings.com](http://www.elinkthings.com)