

AiLink 模块透传应用说明

版本：V3.1.21

更新日期：2020 年 07 月 09 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，易连物联网将不承担责任。

修改记录

文档版本	作者	发布日期	修改说明
3.1.15	梁永新	2020/3/25	1、增加 8 电极体脂秤协议
3.1.16	Lyx	2020/3/27	1、修改八电极体脂秤里的重量协议
3.1.17	Lyx	2020/5/18	1、增加婴儿体脂两用秤协议
3.1.18	Lyx	2020/5/22	1、增加血糖仪协议
3.1.19	LYX	2020/6/24	1、八电极体脂秤里的阻抗长度，从 2 个 byte 改为 4 个 byte
3.1.20	LYX	2020/7/1	1、添加颈椎按摩仪协议
3.1.21	LYX	2020/7/9	1、增加八电极体脂秤指令：MCU 发送测试完成，APP 需回复。

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）.....	- 9 -
4.1 蓝牙名称：AiLink_xxxx.....	- 9 -
4.2 UUID 说明.....	- 9 -
4.3 蓝牙连接服务列表 1： FFE0 举例.....	- 9 -
4.4 广播数据.....	- 9 -
4.4.1 第一类广播设置.....	- 10 -
4.4.2 第二类广播设置.....	- 11 -
5 BM 模块与 MCU 交互协议.....	- 12 -
5.1 设置（获取）指令.....	- 12 -
5.1.1 设置、获取广播名称（Type: 01、02）.....	- 14 -
5.1.2 设置、读取第二类自定义广播数据（Type: 03、04）.....	- 16 -
5.1.3 设置、获取广播间隔时间（Type: 05、06）.....	- 17 -
5.1.4 设置、获取连接间隔参数（Type: 07、08）.....	- 18 -
5.1.5 设置、获取发射功率（Type: 09、0A）.....	- 20 -
5.1.6 设置、读取串口波特率（Type: 0B、0C）.....	- 23 -
5.1.7 读取 MAC 地址（Type: 0D）.....	- 24 -
5.1.8 读取 BM 模块软、硬件版本号（Type: 0E）.....	- 25 -
5.1.9 设置、读取 MCU 软硬件版本号（Type: 0F、10）.....	- 26 -
5.1.10 设置、读取模块的主从模式（Type: 15、16）.....	- 27 -
5.1.11 设置、读取模块自动休眠时间（Type: 17、18）.....	- 28 -
5.1.12 设置模块进入睡眠（Type: 19）.....	- 30 -
5.1.13 设置模块唤醒（Type: 1A）.....	- 31 -
5.1.14 设置、读取系统当前时间（Type: 1B、1C）.....	- 33 -
5.1.15 设置、读取 CID、VID、PID（Type: 1D、1E）.....	- 35 -
5.1.16 设置模块重启（Type: 21）.....	- 36 -
5.1.17 设置恢复出厂设置（Type: 22）.....	- 37 -
5.1.18 设置、获取 BM 模块状态（Type: 25、26）.....	- 38 -
5.1.19 MCU 上报 MCU 电池状态（Type: 27、28）.....	- 39 -
5.1.20 APP 查询 MCU 拥有的单位（Type: 2C）.....	- 41 -
5.1.21 设置使能绑定、取消使能绑定（Type: 32）.....	- 43 -

5.1.22 MCU 设置蓝牙锁开锁类型 (Type: 33、34)	- 43 -
5.1.23 MCU 上传设备基本信息 (Type: 35、36)	- 45 -
5.1.24 APP 同步时间到 MCU (Type: 37、38)	- 47 -
5.1.25 BM 返回扫描到的数据 (Type: 30)	- 49 -
5.1.26 设置 BM 连接从设备 (Type: 38)	- 49 -
5.2 协议透传指令	- 50 -
5.3 数据透传	- 50 -
6 协议透传产品介绍	- 51 -
6.1 八电极体脂秤	- 52 -
6.1.1 交互流程	- 52 -
6.1.2 MCU 上报称重状态	- 53 -
6.1.3 MCU 上报阻抗数据	- 53 -
6.1.4 MCU 上报心率数据	- 55 -
6.1.5 MCU 发送温度数据	- 56 -
6.1.6 MCU 发送测量完成	- 57 -
6.1.7 APP 回复测量完成	- 57 -
6.1.8 APP 下发操作指令	- 58 -
6.1.9 设备上发错误码	- 59 -

1 概述

本文档适用于 BM 系列模块（BM02/08/09/16/20/21/22.....）。

使用 UART 透传，MCU 可以通过 BM 模块与 APP 进行相互数据透传。也支持模块参数设置满足不同需求，也可以通过[协议透传命令](#)快速适配综合超级应用 APP：AiLink，快速实现[血压计](#)、[额温枪](#)、[体温计](#)、[婴儿秤](#)、[身高仪](#)等智能化。



请扫描此二维码下载 AiLink APP。

如使用 AiLink 时，需[严格按照协议透传产品介绍](#)里面的流程进行操作。

下文中表明的 MCU 为与 BM 模块连接交互的芯片，BLE 则为 BM 模块。

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。

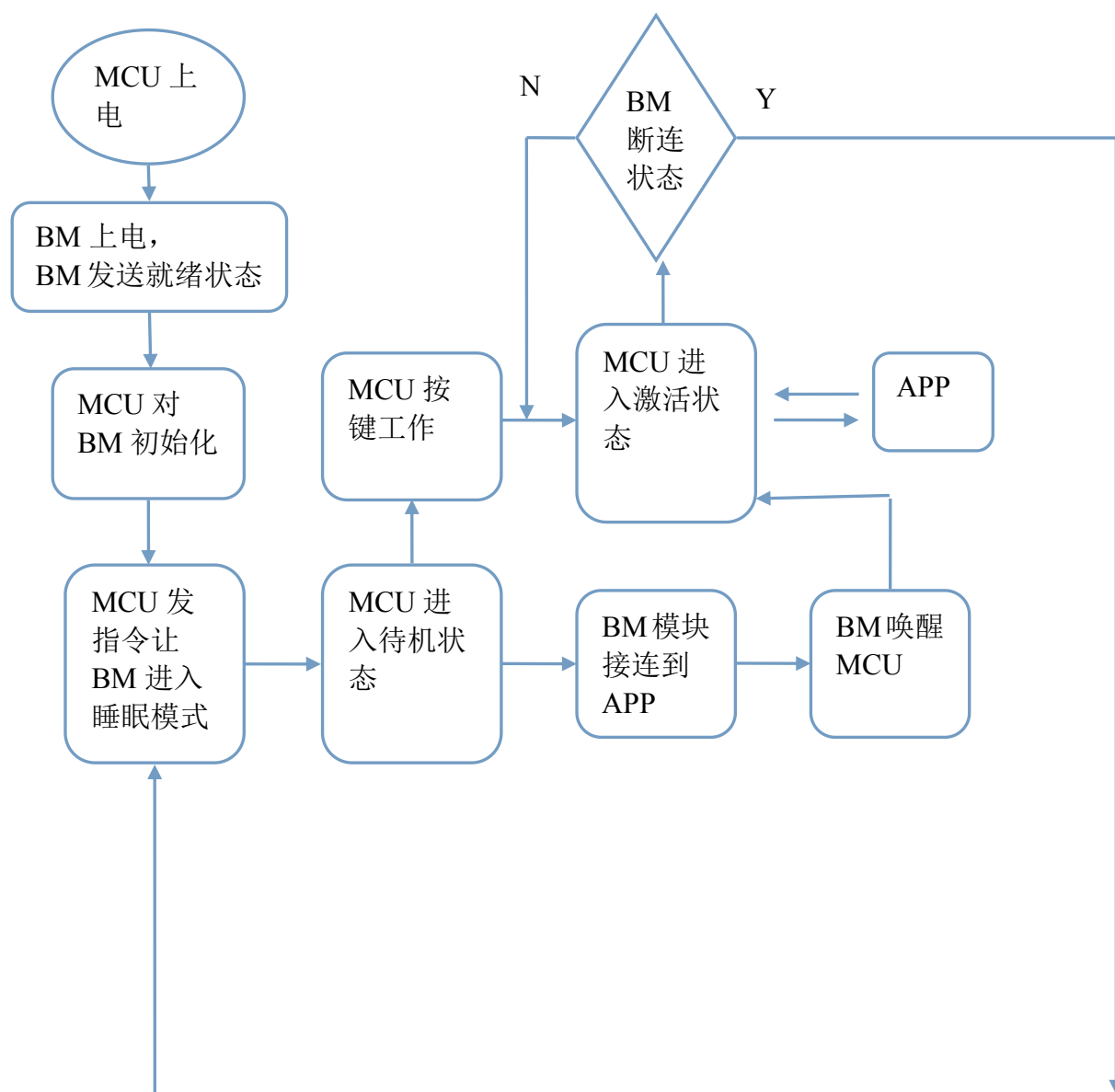
- 此模式下，BM 模块需要长供电，不会断电关机（串口已关闭，BM 处于低功耗模式）。
- 工作流程：

- 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 FFE0。

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](#)）：

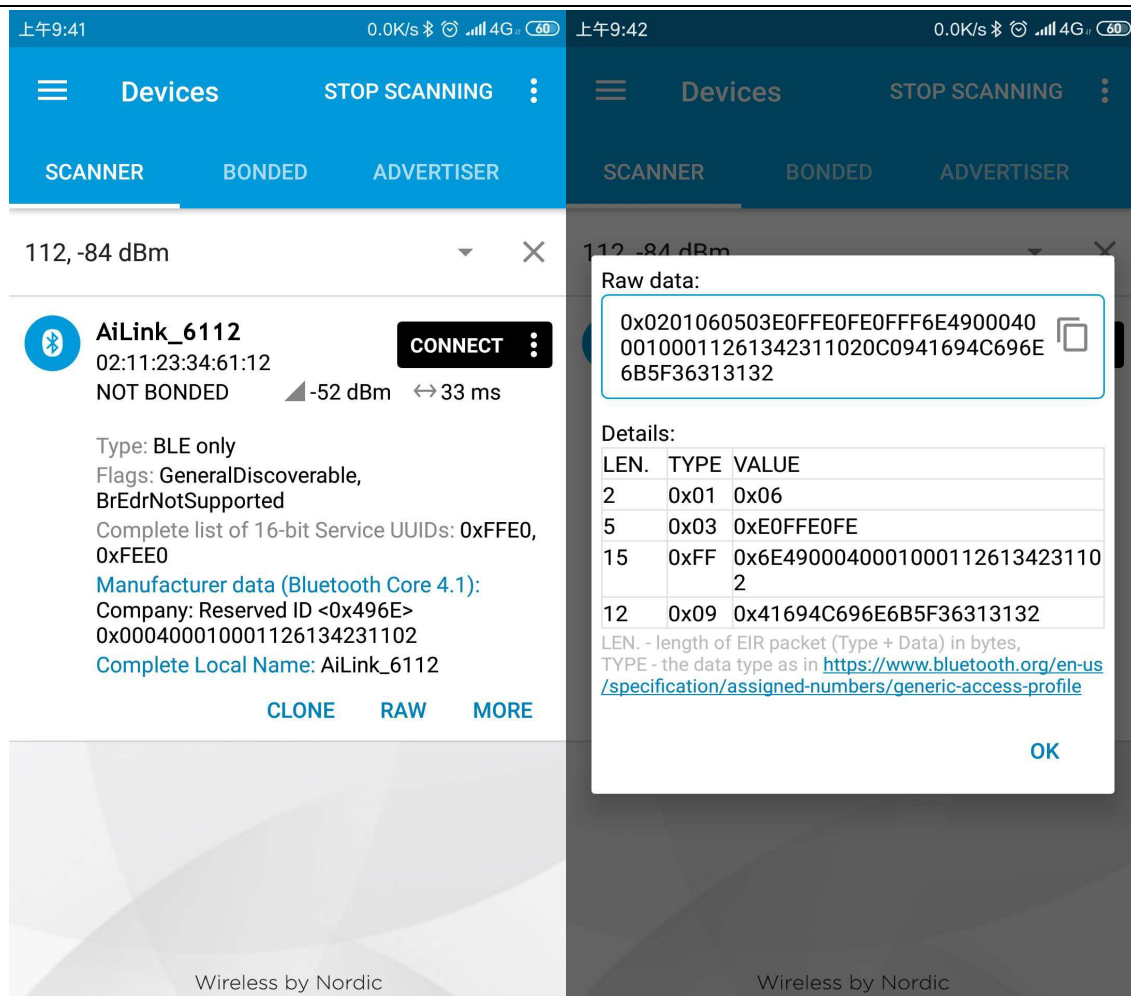
- 1、Company ID 。固定：496e（In, Inet 缩写，此处预留后续申请 SIG company 号）
- 2、CID：产品类型（2bytes）（例如血压计为 0x0001，额温枪为 0x0002，详查看[协议透传产品介绍](#)）
- 3、VID：厂商 ID （2byte） （由我司分配）
- 4、PID：产品 ID （2byte） （由厂商分配）
- 5、Mac 地址（MAC 是固定的，[大小端序可设置](#)，默认小端序）

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（以下指令不透传）。
 - 设置、读取 模块广播名称
 - 设置、读取第二类自定义广播数据
 - 设置、读取 广播间隔时间
 - 设置、读取 连接间隔
 - 设置、读取串口波特率
 - 读取 MAC 地址
 - 读取 BM 模块硬件\软件版本号
 - 设置、读取 MCU 设备版本号
 - 设置、读取 模块主从模式
 - 设置、读取 模块无连接模式自动休眠时间。
 - 设置 模块进入睡眠模式
 - 设置 唤醒模块
 - 设置、读取 当前系统时间。
 - 设置、读取 CID、VID 、PID
 - 设置 模块重启
 - 设置 模块恢复出厂设置。
 - 设置、获取模块状态
 - 设置 MCU 的电池状态
 - 设置 读取 MCU 所支持的单位
 - 设置、取消使能绑定
 - MCU 端设置蓝牙锁的开锁类型
 - MCU 上传设备基本信息
 - APP 同步时间到 MCU
-
- master 模式下设置、获取扫描的白名单
 - master 模式下设置、获取扫描识别的蓝牙名称
 - master 模式下设置、获取扫描状态（待完成）
 - master 模式下 BM 返回扫描到的广播厂商自定义数据

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。

检验和是指 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
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、读取第二类自定义广播数据（Type: 03、04）

➤ 当用户想使用自定义广播数据时，不接入我司 AiLink APP 时，可使用此设置，不设置则默认为我司设置（第一类广播设置）。

设置广播自定义数据：

Byte	Value	Description	
0	0xA6	包头	
1	Len	Payload 长度（最大 16byte）	
2	0x03	Type: 设置为第二类广播自定义数据	Payload
3 ~ n		设置内容 最长为 15byte	
n + 1	Sum	(1~n)校验和	
N+2	0x6A	包尾	

BM 返回设置结果：

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x03	Type: 回复设置为第二类广播自定义数据结果	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	

➤ 举例：

BM 的 Mac 地址为 11:22:33:44:55:66 .

用户若要想设置广播自定义数据为 “0x01 0x02 0x03 0x04 0x05” + “MAC 地址”。

步骤：

1、用户先读取 BM 的 MAC 地址

发送[读取 MAC 地址](#)的指令，获取 MAC 地址：11 : 22 : 33 : 44 : 55 : 66

2、MCU 再发送设置广播自定义数据。

A6 0C 03 01 02 03 04 05 11 22 33 44 55 66 83 6A

获取第二类广播自定义数据：

Byte	Value	Description	
0	0xA6	包头	
1	0x01	Payload 长度	
2	0x04	Type: 获取第二类广播自定义数据	Payload
3	0x05	(1~2)校验和	
4	0x6A	包尾	

返回第二类广播自定义数据：

Byte	Value	Description	
0	0xA6	包头	
1	Len	Payload 长度（最大 16byte）	
2	0x04	Type: 返回第二类广播自定义数据	Payload
3 ~ n		返回的内容 最长为 15byte	
n + 1	Sum	(1~n)校验和	
N+2	0x6A	包尾	

5.1.3 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：

深圳市易连物联网有限公司

电话：(86) 0755-81773367 邮箱：hw@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	0x02	Payload 长度
2	0x05	Type: 回复设置蓝牙广播间隔结果
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
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)
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

5.1.4 设置、获取连接间隔参数 (Type: 07、08)

设置连接间隔参数:

Byte	Value	Description
0	0xA6	包头
1		Payload 长度
2	0x07	Type: 设置连接间隔参数 (连接间隔时间, 从设备延时, 连接超时时间)
3		连接间隔时间的高字节
4		连接间隔时间的低字节
5		从设备延时 (Latency) (取值: 0-4; 默认值 0)
6		连接超时的高字节
7		连接超时的低字节
8	Sum	(1~7)校验和
9	0x6A	包尾

BM 回复设置结果:

Byte	Value	Description
0	0xA6	包头
1	0x02	Payload 长度
2	0x07	Type: 回复设置蓝牙连接间隔结果
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

获取连接间隔时间:

Byte	Value	Description
0	0xA6	包头

1	0x01	Payload 长度	
2	0x08	Type: 获取蓝牙连接间隔	Payload
3	0x09	(1~2)校验和	
4	0x6A	包尾	

BM 返回连接间隔:

Byte	Value	Description	
0	0xA6	包头	
1		Payload 长度	
2	0x08	Type: 设置连接间隔参数（连接间隔时间，从设备延时，连接超时时间）	Payload
3		连接间隔时间的高字节	
4		连接间隔时间的低字节	
5		从设备延时（Latency）（取值：0-4；默认值 0）	
6		连接超时的高字节	
7		连接超时的低字节	
8	Sum	(1~7)校验和	
9	0x6A	包尾	

5.1.5 设置、获取发射功率（Type: 09、0A）

设置蓝牙发射功率:

Byte	Value	Description
0	0xA6	包头

1	0x02	Payload 长度（最大 17byte）	Payload
2	0x09	Type: 设置蓝牙发射功率	
3	dbm	发射功率值 0: -5dbm 1: -4dbm 2: -3dbm 3: -2dbm 4: -1dbm 5: -0dbm 6: 1dbm 7: 2dbm 8: 3dbm 9: 4dbm 10: 5dbm	
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	0x09	Type: 回复设置蓝牙发射功率结果	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	

获取蓝牙发射功率:

Byte	Value	Description	
0	0xA6	包头	
1	0x01	Payload 长度	
2	0x0A	Type: 获取蓝牙发射功率	Payload
3	0x0B	(1~2)校验和	

4	0x6A	包尾
---	------	----

BM 返回发射功率值:

Byte	Value	Description
0	0xA6	包头
1	0x02	Payload 长度
2	0x0A	Type: 回复蓝牙发射功率
3	dbm	发射功率值 0: -5dbm 1: -4dbm 2: -3dbm 3: -2dbm 4: -1dbm 5: -0dbm 6: 1dbm 7: 2dbm 8: 3dbm 9: 4dbm 10: 5dbm
4	Sum	(1~3)校验和
5	0x6A	包尾

Payload

5.1.6 设置、读取串口波特率（Type: 0B、0C）

设置串口波特率：

Byte	Value	Description
0	0xA6	包头
1	0x02	Payload 长度
2	0x0B	Type: 设置串口波特率
3		Value: 0: 9600 （默认） 1: 19200 2: 38400 3: 57600 4: 115200 5: 921600
4	Sum	(1~3)校验和
5	0x6A	包尾

➤ 举例：设置串口波特率：9600

发送： A6 02 0B 00 0D 6A

BM 回复设置结果：

Byte	Value	Description
0	0xA6	包头
1	0x02	Payload 长度
2	0x0B	Type: 回复设置串口波特率结果
3		结果值: 0: 成功。（以旧的波特率回复后再切换新的波特率） 1: 失败 2: 不支持
4	Sum	(1~3)校验和
5	0x6A	包尾

获取串口波特率：

Byte	Value	Description
0	0xA6	包头
1	0x01	Payload 长度
2	0x0C	Type: 获取串口波特率
3	0x0D	(1~2)校验和
4	0x6A	包尾

BM 返回串口波特率值:

Byte	Value	Description
0	0xA6	包头
1	0x02	Payload 长度
2	0x0C	Type: 回复串口波特率设置值
3		串口波特率设置值 0: 9600 1: 19200 2: 38400 3: 57600 4: 115200 5: 921600
4	Sum	(1~3)校验和
5	0x6A	包尾

➤ 举例: 若串口波特率为 9200

返回: A6 02 0C 00 0E 6A

5.1.7 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
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
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

5.1.8 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
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
3		Payload
4		
5		
6		
7		
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

5.1.9 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
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
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
3	0x11	(1~2)校验和
4	0x6A	包尾

BM 返回 MCU 软硬件版本号：

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

5.1.10 设置、读取模块的主从模式 (Type: 15、16)

设置主从模式：

Byte	Value	Description
0	0xA6	包头
1	0x02	Payload 长度
2	0x15	Type: 设置主从模式
3		Value: 0: 从机模式 (默认) 1: 主机模式
4		(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	0x15	Type: 回复设置主从模式结果
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
---	------	----

获取主从模式设置：

Byte	Value	Description
0	0xA6	包头
1	0x01	Payload 长度
2	0x16	Type: 获取主从模式
3	0x17	(1~2)校验和
4	0x6A	包尾

BM 返回主从模式设置：

Byte	Value	Description
0	0xA6	包头
1	0x02	Payload 长度
2	0x16	Type: 回复主从模式
3		设置值： 0: 从机模式 1: 主机模式
4	Sum	(1~3)校验和
5	0x6A	包尾

5.1.11 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: 设置无连接自动睡眠时间
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	单位 ： s 范围： 5 ~ 0xffffffff/100 （建议设为： 60s）
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	单位： ms ； 范围： 20~2000；（建议设为 1000）；
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	
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	

5.1.12 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	
6		低频广播间隔时间的低字节	
		单位 : ms ; 范围 20~2000 (建议 1000ms)	

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
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](#)”。

5.1.13 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
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		

5.1.14 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
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
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
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	Payload
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	

5.1.15 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
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
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:

深圳市易连物联网有限公司

电话: (86) 0755-81773367 邮箱: hw@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	0x01	Payload 长度
2	0x1E	Type: 获取 ID 设置值
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
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

5.1.16 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
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	

5.1.17 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	

5.1.18 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	

5.1.19 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
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
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

5.1.20 APP 查询 MCU 拥有的单位 (Type: 2C)

- APP 端界面的单位显示是根据 MCU 端所拥有的单位来做处理的，所以当 APP 连接到 MCU 时，会发送读取指令来获取 MCU 端所拥有的单位，所以 MCU 端收到该指令时，则务必返回相应的单位数据（不返回则使用系统默认值）。
- **MCU 端需要开机后直接主动上传单位到 APP。**

APP 查询 MCU 端单位指令：
(BM 模块直接将此指令传给 MCU)

Byte	Value	Description	
0	0xA6	包头	
1		Payload 长度	
2	0x2C	Type: APP 读取 MCU 端单位	Payload
3		Value 0x01	
4		校验和	
5	0x6A	包尾	

MCU 端返回所拥有的单位指令：
单位类型

类型编号	类型	支持类型 (Bit15~Bit0) Bit=0 不支持 Bit=1 支持
01	重量	Bit0: kg Bit1: 斤 Bit2: lb: oz Bit3: oz Bit4: st: lb Bit5: g Bit6: lb (纯 lb 显示) Bit7-bit15 保留
02	长度	Bit0: cm Bit1: inch Bit2: ft-in Bit3-bit15 保留
03	温度	Bit0 : C Bit1 : F Bit2-bit15 保留
04	血压	Bit0: mmhg Bit1: kPa

		Bit2-bit15 保留
05	轮胎胎压压力	Bit0: Kpa Bit1: Psi Bit2: Bar Bit3-bit15 保留
06	血糖仪	Bit0: mmol/L Bit1: mg/dL

(BM 模块通过 A6 指令协议传给 APP)

数据格式* (每组数据长度不能多于 20 个 byte)

Byte	Value	Description
0	0xA6	包头
1		Payload 长度
2	0x2C	Type: MCU 回复单位
3		单位类型: 例如: 重量类型 =01, 长度=02, 温度=03
4		该单位支持类型高位: Bit15~Bit0
5		该单位支持类型低位: 每一个 Bit 代表一个单位 Bit=0: 不支持 Bit=1: 支持 例如: 重量支持 kg 和 oz 则 byte4=0x00, byte5=0x09
6		单位类型:
7		单位支持类型高位 Bit15~Bit0
8		单位支持类型低位 每一个 Bit 代表一个单位 Bit=0: 不支持 Bit=1: 支持
9		单位类型:
10		单位支持类型高位 Bit15~Bit0
11		单位支持类型低位 每一个 Bit 代表一个单位 Bit=0: 不支持 Bit=1: 支持
12		校验和
13	0x6A	包尾

➤ 举例:

APP 读取 MCU 单位, 发送指令: A6 02 2C 01 2F 6A

- 若 MCU 只拥有重量单位 kg 和斤, 则返回: A6 04 2C 01 00 03 34 6A
- 若 MCU 只拥有重量单位 kg 和长度单位 inch, 则返回: A6 07 2C 01 00 01 02 00 02 39 6A
- 若 MCU 只拥有胎压单位 Kpa、Psi、Bar 和温度单位 °C、°F 和重量单位 kg 和 长度单位 cm, 则返回: A6 0D 2C 05 00 07 03 00 03 01 00 01 02 00 01 50 6A
- 若是 MCU 支持的类型太多, 一组数据传不完, 则可以分开多组来传, 数据格式不变。

5.1.21 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
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
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

5.1.22 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
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	

5.1.23 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	

5.1.24 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

5.1.25 BM 返回扫描到的数据 (Type: 30)

Byte	Value	Description
0	0xA6	包头
1		Payload 长度
2	0x30	Type: 返回扫描到的数据
3~8		该设备的 mac 地址, 小端序
9		RSSI 值 (如 50 是 -50dbm 的意思)
10~n		厂商自定义数据 (当不需返回该数据时无该数据)
n+1		校验和
n+2	0x6A	包尾

此指令比较特殊, 返回的总数据长度可以超过 20byte

举例:

A6 19 30 BB FF B9 EC B4 01 32 AC 00 C6 5A 5A 01 00 7B 26 0B 0B BB FF B9 EC B4 01 81 6A

mac 地址: BB FF B9 EC B4 01 (小端序)

信号强度: -50dbm

厂商自定义数据为: AC 00 C6 5A 5A 01 00 7B 26 0B 0B BB FF B9 EC B4 01

5.1.26 设置 BM 连接从设备 (Type: 38)

设置白名单

Byte	Value	Description
0	0xA6	包头
1		Payload 长度
2	0x38	Type: 设置 BM 连接
3~8		MAC 地址 (6 个字节, 小端序)
9	Sum	(1~8)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	0x38	Type: 回复设置结果
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

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）
1	血压计	0001
2	额温枪	0002
3	体温计	0003
4	婴儿秤	0004
5	身高仪	0005
6	光学验光仪控制器	000C
7	胎压转接板	000D
8	体重/体脂秤	000E
9	蓝牙门锁	000B
10	箱包锁	000F
11	八电极体脂秤	0013

6.1 八电极体脂秤

6.1.1 交互流程

以下流程步骤，缺一不可。

1. BM 模块上电。
2. [BM 模块返回状态信息](#)。
3. [MCU 设置 CID](#)。
4. [MCU 上传自身支持的单位](#)。
5. MCU 设置蓝牙其他参数，（可根据自身需求设置）。
6. [APP 查询 MCU 所拥有的单位](#)。
7. 设备启动，上传数据。
8. 设备测量完毕，上报测量完毕指令。
9. APP 设置单位。
10. MCU 休眠。

6.1.2 MCU 上报称重状态

数据格式：

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0013	产品类型（CID）	
3		Payload 长度	
4		类型：称重 01	Payload
5		Type：测量状态 01：实时体重 02：稳定体重	
6		重量数据高位	
7		重量数据次高位	
8		重量数据低位	
9		数据标志： Bit7~4：重量数据精度（该数据标识的是次级单位里的数据格式） 0000：0 位小数 0001：1 位小数 0010：2 位小数 0011：3 位小数 Bit3~0：当前单位：（标识整一组数据的单位类型） 0000：kg 0001：斤 0100：st.lb 0110：lb	
10		保留为：00	
11	SUM	校验和（byte1~byte12）	
12	0x7A	包尾	

6.1.3 MCU 上报阻抗数据

数据格式：

深圳市易连物联网有限公司

电话：(86) 0755-81773367 邮箱：hw@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	Default	Description
0	0xA7	包头
1~2	0x0013	产品类型 (CID)
3		Payload 长度
4		类型: 测试阻抗 02
5		Type: 阻抗测量 01: 测阻抗中 02: 测阻抗失败 03: 测阻抗成功, 带上阻抗数据, 并使用 APP 算法(APP 会根据 byte7 的算法标识进行计算) 04: 测阻抗结束。
6		通道: 0x00 双脚阻抗 0x01 双手阻抗 0x02 左手阻抗 0x03 右手阻抗 0x04 左脚阻抗 0x05 右脚阻抗 0x06 左全身阻抗 0x07 右全身阻抗 0x08 右手左脚阻抗 0x09 左手右脚阻抗 0x0A 躯干阻抗
7-10		阻抗数据(大端序, 精度 1 Ω)
11		体脂算法 ID, 每个客户都有自己的算法 ID (1~255)
12	0	保留位
13	SUM	校验和 (byte1~byte12)
14	0x7A	包尾

Payload

6.1.4 MCU 上报心率数据

数据格式：

Byte	Default	Description
0	0xA7	包头
1~2	0x0013	产品类型（CID）
3		Payload 长度
4		类型：测试心率 03
5		Type: 心率测量 01: 测心率中 02: 测心率成功，带上心率数据 03: 测心率失败
6		心率数据（精度 1bpm）
7	0x00	保留
8	SUM	校验和
9	0x7A	包尾

6.1.5 MCU 发送温度数据

数据格式：

Byte	Default	Description
0	0xA7	包头
1~2	0x0013	产品类型（CID）
3		Payload 长度
4		类型：温度数据 04
5		温度正负 0：正温度 1：负温度
6		温度数据高字节
7		温度数据低字节
8		数据标志： Bit7~4：温度数据精度 0000：0 位小数 0001：1 位小数 0010：2 位小数 0011：3 位小数 Bit3~0：当前单位：（标识整一组数据的单位类型） 0000：℃ 0001：°F
9	0	保留
10	SUM	校验和
11	0x7A	包尾

6.1.6 MCU 发送测量完成

数据格式：

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0013	产品类型（CID）	
3		Payload 长度	
4	0x0F	Type: 测量完成 0x0F	Payload
5	0	保留	
6	SUM	校验和	
7	0x7A	包尾	

6.1.7 APP 回复测量完成

APP 收到 MCU 的测量完成指令后，需回复该指令。

数据格式：

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0013	产品类型（CID）	
3		Payload 长度	
4	0x84	Type: APP 测量完成 0x84	Payload
5	0	保留	
6	SUM	校验和	
7	0x7A	包尾	

6.1.8 APP 下发操作指令

数据格式：

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0013	产品类型（CID）	
3		Payload 长度	
4	0x81	Type: app 下发操作指令 0x81	Payload
5		操作： 0x01：校准 0x02：温度单位切换 0x03：重量单位切换	
6		当操作是 0x02 时 0：摄氏度℃ 1：华氏度°F 当操作是 0x03 时 0：kg 1：斤 4：st:lb 6：lb	
7		保留 0x00	
8	SUM	校验和	
9	0x7A	包尾	

MCU 回复操作结果：

数据格式：

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0013	产品类型（CID）	
3		Payload 长度	
4	0x82	Type: MCU 回复操作结果 0x82	Payload
5		操作： 0x01：校准 0x02：温度单位切换	

		0x03: 重量单位切换	
6		操作结果: 0: 操作成功 1: 操作失败 2: 正在操作中	
7		保留 0x00	
8	SUM	校验和	
9	0x7A	包尾	

6.1.9 设备上发错误码

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0013	产品类型 (CID)	
3		Payload 长度	
4	0xFF	Tpye: 错误码	Payload
5		错误内容: 1: 超重 ...	
6	SUM	校验和	
7	0x7A	包尾	

