

# AiLink-血氧仪 应用手册

版本：V1.1

更新日期：2020 年 12 月 10 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
V1.0	lxl	2020/8/26	1. amlink 增加血氧仪协议
V1.1	lyx	2020/12/10	1. 更改 MCU 上报信息协议

# 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 广播数据.....	- 8 -
4.2.1 第一类广播设置.....	- 8 -
5 BM 模块与 MCU 交互协议.....	- 10 -
5.1 设置（获取）指令.....	- 10 -
5.1.1 设置、获取广播名称（Type: 01、02）.....	- 11 -
5.1.2 设置、获取广播间隔时间（Type: 05、06）.....	- 13 -
5.1.3 读取 MAC 地址（Type: 0D）.....	- 14 -
5.1.4 读取 BM 模块软、硬件版本号（Type: 0E）.....	- 15 -
5.1.5 设置、读取 MCU 软硬件版本号（Type: 0F、10）.....	- 16 -
5.1.6 设置、读取模块自动休眠时间（Type: 17、18）.....	- 17 -
5.1.7 设置模块进入睡眠（Type: 19）.....	- 19 -
5.1.8 设置模块唤醒（Type: 1A）.....	- 20 -
5.1.9 设置、读取系统当前时间（Type: 1B、1C）.....	- 21 -
5.1.10 设置、读取 CID、VID、PID（Type: 1D、1E）.....	- 23 -
5.1.11 设置模块重启（Type: 21）.....	- 24 -
5.1.12 设置恢复出厂设置（Type: 22）.....	- 25 -
5.1.13 设置、获取 BM 模块状态（Type: 25、26）.....	- 26 -
5.1.14 MCU 上报 MCU 电池状态（Type: 27、28）.....	- 27 -
5.1.15 APP 查询 MCU 拥有的单位（Type: 2C）.....	- 29 -
5.1.16 MCU 上传设备基本信息（Type: 35、36）.....	- 31 -
5.1.17 APP 同步时间到 MCU（Type: 37、38）.....	- 32 -
6 协议透传产品介绍.....	- 34 -
6.1 血氧仪.....	- 34 -
6.1.1 交互流程.....	- 34 -
6.1.2 交互指令.....	- 34 -
7 使用/测试指导.....	- 38 -
7.1 测试工具.....	- 38 -
7.2 测试步骤.....	- 38 -
8 自定义说明.....	- 38 -

---

9 联系我们.....	- 38 -
10 附录.....	- 38 -

# 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）。

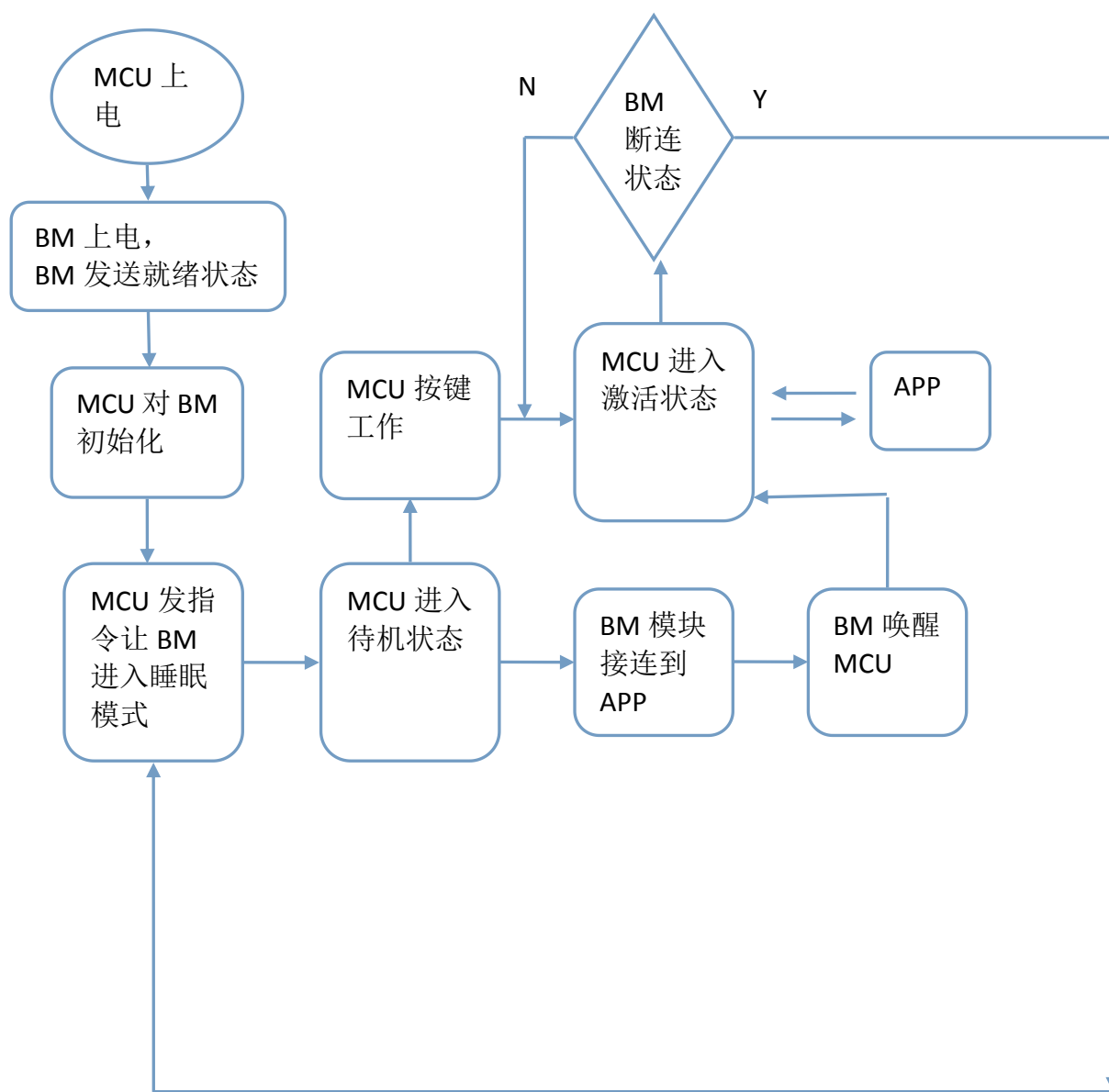
➤ 工作流程：

- 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 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。

#### 4.2.1 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）

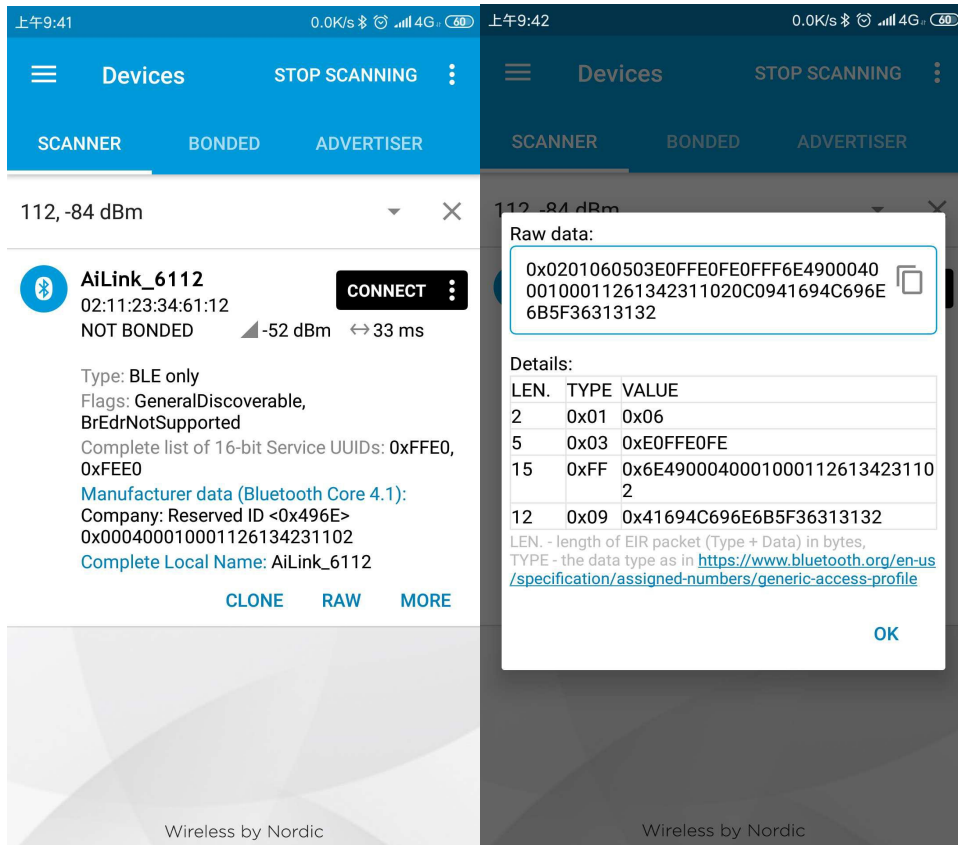
例如广播出来的自定义数据为：

**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：





## 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

更多指令请查看 [ailink 通用指令设置](http://www.elinkthings.com/cn/help-detail-168.html)：<http://www.elinkthings.com/cn/help-detail-168.html>

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
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
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
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:**

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
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

			为 1000) ;
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
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
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
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:**

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
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
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
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 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 <b>0x01</b>	
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

Payload

➤ 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。

### 5.1.16 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
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

### 5.1.17 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	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	

● **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	Payload
3		Value 0x01	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

## 6 协议透传产品介绍

### 6.1 血氧仪

#### 6.1.1 交互流程

1. 设备开机。
2. BM 模块返回状态信息到 MCU。
3. MCU 设置 CID、VID、PID（必须设，具体值联系我司）。
4. APP 连接，BM 模块会返回连接状态。
5. MCU 上传设备状态。
6. MCU 发送测量结果数据。
7. MCU 发送设置数据。
8. APP 发送设置报警数据。
9. APP 保存数据。
10. 设备关机。

#### 6.1.2 交互指令

##### 6.1.2.1 APP 获取设备状态

- 为了保证 APP 界面与设备上的界面同步，所以 APP 接连设备后，需查询设备的状态。MCU 接收到 APP 查询状态的指令后，MCU 需要回复设备当前状态。

Byte	Default	Description
0	0xA7	包头
1~2	0x0021	产品类型： 0x0021
3		Payload 长度
4	0x01	Type: APP 获取设备状态
5		0x01
6	SUM (1~5)	(1~5)校验和
7	0x7A	包尾

### 6.1.2.2 MCU 上报信息

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0021	产品类型: 0x0021	
3		Payload 长度	
4	0x03	Type: MCU 上报测量结果	Payload
5		测量标识符 0x00 : 开始测试 0x01 : 正在测试 0xFF : 测试结束	
6		血氧 (SpO2 , 单位 %) 0-100 % 0: 显示-- 若该值无效, 则为 0xFF	
7		脉率/心率 (pulse rate , 单位 bpm) 0-250 0: 显示-- 若该值无效, 则为 0xFF	
8		PI (血流灌注指数, 无单位, 1位小数) 0-254 (对应 0.0-25.4) 0: 显示-- 若该值无效, 则为 0xFF	
9		电量 (power , 单位%) 0-100 若该值无效, 则为 0xFF	
10 - 11		呼吸频率 Respiratory Rate (RR/min , 大端序, 一位小数点) 0.0-100.0 0.0: 显示-- 若该值无效, 则为 0xFFFF	
12-13		脉率曲线值 (Plethysmogram 体积描记图, 大端序) 0-0xFFFE 若该值无效, 则为 0xFFFF	
14-15		脉率曲线波谷 (棒图, 大端序) 0-0xFFFF 若该值无效, 则为 0xFFFF	
16		佩戴状态 0: 未佩戴 1: 已佩戴	

17		保留位 0	
18	SUM (1~n)	(1~n)校验和	
19	0x7A	包尾	

注：如果设备不支持哪项参数，该参数填写 0xFF。

### 6.1.2.3 APP 设置报警值

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0021	产品类型： 0x0021	
3		Payload 长度	
4	0x04	Type: APP 下发报警值	Payload
5		血氧饱和度 (SpO2) 报警上限值： 0-100 % 若该值无效，则为 0xFF	
6		血氧饱和度 (SpO2) 报警下限值 0-100 % 若该值无效，则为 0xFF	
7		脉率 (PR) 报警上限值 0-255 若该值无效，则为 0xFF	
8		脉率 (PR) 报警下限值 0-255 若该值无效，则为 0xFF	
9		血流灌注指数 (PI) 报警上限值 (无单位，1 位小数) 0-100 若该值无效，则为 0xFF	
10		血流灌注指数 (PI) 报警下限值 (无单位，1 位小数) 0-100 若该值无效，则为 0xFF	
11	SUM (1~n)	(1~n)校验和	
12	0x7A	包尾	

注：如果设备不支持哪项参数，该参数填写 0xFF。

MCU 回复设置结果:

Byte	Default	Description
0	0xA7	包头
1~2	0x0021	产品类型: 0x0021
3		Payload 长度
4	0x05	Type: MCU 返回设置结果
5		0x00: 成功 0x01: 失败
6		(1~5)校验和
7	0x7A	包尾

#### 6.1.2.4 设备上报错误码

Byte	Default	Description
0	0xA7	包头
1~2	0x0021	产品类型: 0x0021
3		Payload 长度
4	0xFF	Type: 错误码
5		0: 血氧饱和率不稳定 1: 脉率不稳定 2: 测量出错 3: 设备低电
6	SUM (1~n)	(1~n)校验和
7	0x7A	包尾

## 7 使用/测试指导

### 7.1 测试工具

### 7.2 测试步骤

## 8 自定义说明

## 9 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

Email: [hw@elinkthings.com](mailto:hw@elinkthings.com)

Web: [www.elinkthings.COM](http://www.elinkthings.COM)

## 10 附录