

AiLink 两轮电动车应用手册

版本：V1.1

更新日期：2024 年 12 月 23 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	审核	发布日期	修改说明
V1.0	Lxl		2024/10/21	1、补充：两轮车不同产品类型的CID描述
V1.1	Lxl	Cfx	2024/12/23	基于目前 morego app 补充对应指令

目录

修改记录.....	- 2 -
目录.....	- 3 -
1 概述.....	- 5 -
2 说明.....	- 5 -
3 蓝牙接口（默认）.....	- 6 -
3.1 蓝牙名称：AiLink_xxxx.....	- 6 -
3.2 UUID 说明.....	- 6 -
3.3 蓝牙连接服务列表 1： FFE0 举例.....	- 6 -
3.4 广播数据.....	- 6 -
3.4.1 第一类广播设置.....	- 7 -
4 通用设置获取指令.....	- 8 -
6.1 读取 BLE 地址（Type: 0D）.....	- 8 -
6.2 读取模块版本号（Type: 0E）.....	- 9 -
6.3 设置模块 CIDVIDPID（Type: 1D）.....	- 10 -
6.4 读取模块 CIDVIDPID（Type: 1E）.....	- 11 -
6.5 复位模块 Type: 21）.....	- 11 -
6.6 恢复出厂设置（Type: 22）.....	- 12 -
6.7 获取模块状态（Type: 26）.....	- 12 -
6.8 APP 同步时间给 MCU（Type: 37）.....	- 13 -
6.9 MCU 请求同步时间（Type: 38）.....	- 14 -
6.10 设置模块自动修眠时间 Type = 0x17.....	- 14 -
6.11 读取模块自动修眠时间 Type = 0x18.....	- 15 -
6.12 设置模块立即进入休眠 Type = 0x19.....	- 16 -
6.13 唤醒模块 Type = 0x1a.....	- 17 -
5 协议透传指令.....	- 18 -
● 产品类型(CID)描述.....	- 18 -
5.1 查询接收端类型及支持的功能（Type: 00）.....	- 19 -
5.2 心跳数据交互（Type: 01）.....	- 21 -
5.3 设置控制器状态（Type: 02）.....	- 23 -
5.4 清除 MCU 模块参数（Type: 03）.....	- 24 -
5.5 设置模式档位速度（Type: 04）.....	- 25 -
5.6 设置灯光（Type: 05）.....	- 25 -
5.7 设置解锁车密码（Type: 06）.....	- 26 -
5.8 锁车及解锁（Type: 07）.....	- 27 -
5.9 设置查询仪表背光状态（Type: 09）.....	- 28 -
5.10 查询总里程（Type: 0D）.....	- 29 -
5.11 查询电池信息（Type: 0E）.....	- 30 -
5.12 查询电池厂商代码及编号（预留 Type: 0F）.....	- 31 -
5.13 查询控制器厂商代码及编号（Type: 10）.....	- 31 -
5.14 查询仪表固件版本（Type: 11）.....	- 32 -

5.15	天气信息 (Type: 13)	- 33 -
5.16	导航数据 (Type: 14)	- 34 -
5.17	当前时区 (Type: 15)	- 35 -
5.18	设置查询车辆开锁密码 (Type: 1A)	- 35 -
5.19	自动关机/锁车 (Type: 22)	- 36 -
5.20	车辆密码开关 (Type: 26)	- 37 -
6	数据透传	- 37 -
7	OTA	- 38 -
5.1	仪表 OTA	- 38 -
5.1.1	发送 OTA 开始指令(Type:0xA0)	- 38 -
5.1.2	发送 OTA 数据(Type:0xA1)	- 39 -
5.1.3	发送 OTA 完成指令(Type:0xA2)	- 40 -
5.2	控制器 OTA	- 40 -
5.2.1	发送 OTA 开始指令(Type:0xA3)	- 40 -
5.2.2	发送 OTA 数据(Type:0xA4)	- 41 -
5.2.3	发送 OTA 完成指令(Type:0xA5)	- 42 -
8	联系我们	- 43 -

1 概述

本文档适用于搭载易连物联网的 AiLink 系列模块（BM16/18/28...）。

本文档适用于两轮车仪表等设备搭载 BM 模块，接入 MoreGo 应用。



请扫描此二维码下载 APP。

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

2 说明

2.1 A7 指令为和电动车数据相关的协议透传指令，分设置和查询指令，具体协议参考相关章节描述

2.2 A7 分为设置类指令和查询类指令，设置类指令，BM 模块接收后，回复结果，和电动车状态相关的数据在心跳包中体现；查询类指令在 BM 模块接收后，返回指令对应的数据。

3 蓝牙接口（默认）

3.1 蓝牙名称：AiLink_XXXX

注：XXXX 为 Mac 地址后 4 个字符

3.2 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。

3.3 蓝牙连接服务列表 1： FFE0 举例

3.3.1 服务 UUID:

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

3.3.2 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

3.3.3 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

3.3.4 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

3.4 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。

3.4.1 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）：

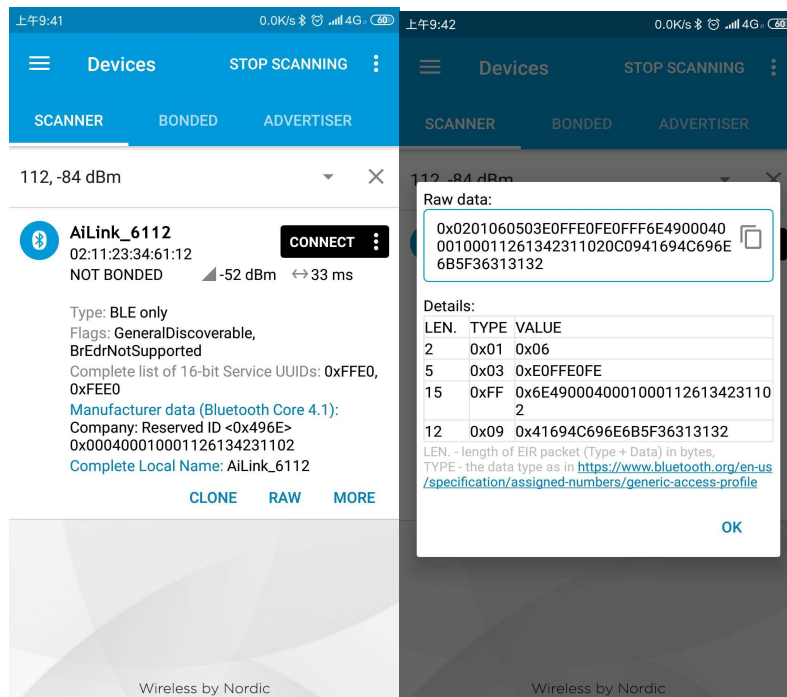
- 1、Company ID 。固定：496e（In, Inet 缩写，此处预留后续申请 SIG company 号）
- 2、CID：产品类型（2bytes）
- 3、VID：厂商 ID （2byte）
- 4、PID：产品 ID （2byte）
- 5、Mac 地址（MAC 是固定的，默认小端序）

例如广播出来的自定义数据为：

6e4900010001000112613423110200

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 通用设置获取指令

通用设置获取指令格式规范（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

6.1 读取 BLE 地址（Type: 0D）

BM 接收：

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

BM 响应：

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

BM 返回：A6 07 0D 66 55 44 33 22 11 79 6A

6.2 读取模块版本号 (Type: 0E)

BM 接收:

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

BM 响应:

Byte	Value	Description	
0	0xA6	包头	
1	0x0A	Payload 长度	
2	0x0E	Type: 回复模块软硬件版本号	
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 (1~11)	校验和	
13	0x6A	包尾	

➤ 举例: 如软硬件版本号为 WM06H1S1.0P0_20190507

解析: WM05 为产品型号, 对应实际数据为 0x57 0x4D 0x06

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

BM 返回: A6 0A 0E 57 4D 06 01 0A 00 13 05 07 EC 6A

6.3 设置模块 CIDVIDPID (Type: 1D)

- CID 为产品类型 ID, 请按照协议透传产品类型设置 (必须设)
- VID 为设备厂家 ID, 请联系我司分配 (必须设)
- 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 (必须设)
- 以上三个值默认为 0, 不代表任何产品 (调试阶段先设置 CID)
- ailnk CID VID PID 获取介绍: http://doc.elinkthings.com/web/#/40?page_id=144

BM 接收:

Byte	Value	Description	
0	0xA6	包头	
1		Payload 长度	
2	0x1D	Type: 设置 ID	Payload
3	0x07	ID 标志位	
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 响应:

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: 0x00: 成功 0x01: 失败 0x02: 不支持	
4	Sum (1~3)	校验和	
5	0x6A	包尾	

- 举例: 设置模块 CID 为滑板车

MCU 发送: A6 08 1D 07 00 33 00 00 00 00 5F 6A

6.4 读取模块 CIDVIDPID (Type: 1E)

BM 接收:

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

BM 响应:

Byte	Value	Description	
0	0xA6	包头	
1	Len	Payload 长度	
2	0x1E	Type: 返回 ID 值	Payload
3	0x07	ID 标志位	
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	

6.5 复位模块 Type: 21)

BM 接收:

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	校验和	
5	0x6A	包尾	

BM 响应:

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: 0x00: 成功 0x01: 失败 0x02: 不支持	
4	Sum (1~3)	校验和	
5	0x6A	包尾	

6.6 恢复出厂设置 (Type: 22)

BM 接收:

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	0x25	校验和	
5	0x6A	包尾	

BM 响应:

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: 0x00: 成功 (成功后, 100ms 后恢复出厂设置) 0x01: 失败 0x02: 不支持	
4	Sum (1~3)	校验和	
5	0x6A	包尾	

6.7 获取模块状态 (Type: 26)

BM 接收:

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

0	0xA6	包头	
1	0x01	Payload 长度	
2	0x26	Type: 获取状态	Payload
3	0x27	校验和	
4	0x6A	包尾	

BM 响应:

Byte	Value	Description	
0	0xA6	包头	
1	0x03	Payload 长度	
2	0x26	Type: 返回模块状态	Payload
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	

当蓝牙状态改变时，蓝牙模块都会通过获取模块状态的响应包格式主动进行上报状态变化。

6.8 APP 同步时间给 MCU (Type: 37)

APP 下发:

Byte	Default	Description		
0	0xA6	包头		
1		Payload 长度 (最大 15byte)		
2	0x37	Type: 设置 MCU 时间	Payload	
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 (1~9)	校验和		
11	0x6A	包尾		

MCU 响应:

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x37	Type: 回复设置时间结果	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	

在 APP 和 BM 模块通过 BLE 连接后, APP 通过此指令下发到 BM 模块, BM 模块会将此指令透传给 MCU, MCU 端需要时间日期的, 可以通过此指令更新 MCU 端的时间日期。

6.9 MCU 请求同步时间 (Type: 38)

MCU 发送:

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x38	Type: 请求同步时间	Payload
3	0x01	Value: 0x01	
4	0x3b	校验和	
5	0x6A	包尾	

在 BM 模块和 APP 有连接时, 会将此指令透传给 APP, APP 端会通过 0x37 指令同步时间到 MCU。

6.10 设置模块自动休眠时间 Type = 0x17

模块接收:

Byte	Value	Description		
0	0xA6	包头		
1	0x09	Payload 长度		
2	0x17	Type: 设置自动睡眠时间	Payload	
3		自动睡眠标志位: 0: 不开启自动休眠 1: 开启自动休眠		
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		范围:

6		自动睡眠时间的次低字节	5 ~ 0xffffffff/100 (建议设为: 60s)
7		自动睡眠时间的最低字节	
8		休眠模式: 0x00: BLE 关闭, 串口关闭 (掉电模式)。 0x01: BLE 正常工作, 串口关闭 (正常睡眠)。	
9	0xFF	保留位	
10	0xFF	保留位	
11	Sum (1~10)	校验和	
12	0x6A	包尾	

模块响应:

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: 0x00: 成功 0x01: 失败 0x02: 不支持	
4	Sum (1~3)	校验和	
5	0x6A	包尾	

➤ 举例: 设置 20S 自动进入掉电模式

发送: A6 09 17 01 00 00 00 14 00 FF FF 33 6A

6.11 读取模块自动修眠时间 Type = 0x18

模块接收:

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	校验和	
4	0x6A	包尾	

模块响应:

Byte	Value	Description
0	0xA6	包头
1	Len	Payload 长度

2	0x18	Type: 返回自动睡眠时间		Payload
3		自动睡眠标志位: 0: 不开启自动休眠 1: 开启自动休眠		
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		休眠模式: 0x00: BLE 关闭, WIFI 关闭, 串口关闭 (掉电模式)。 0x01: BLE 正常工作, WIFI 正常工作, 串口关闭 (正常睡眠)。		
9		保留位		
10		保留位		
11	Sum (1~10)	校验和		
12	0x6A	包尾		

6.12 设置模块立即进入休眠 Type = 0x19

模块接收:

Byte	Value	Description	
0	0xA6	包头	
1	Len	Payload 长度	
2	0x19	Type: 设置进入睡眠	Payload
3	0x01	Value: 0x01	
4		休眠模式: 0x00: BLE 关闭, WIFI 关闭, 串口关闭 (掉电模式)。 0x01: BLE 正常工作, WIFI 正常工作, 串口关闭 (正常睡眠)。	
5		保留位	
6		保留位	
7	Sum (1~6)	校验和	
8	0x6A	包尾	

模块响应:

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： 0x00：成功（成功后 100ms 后进入睡眠） 0x01：失败 0x02：不支持	
4	Sum (1~3)	校验和	
5	0x6A	包尾	

➤ 举例：设置进入掉电模式

发送：A6 05 19 01 00 00 00 1f 6A

6.13 唤醒模块 Type = 0x1a

模块接收：

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: 0x01	
4	0x1D	校验和	
5	0x6A	包尾	

模块响应：

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： 0x00：成功 0x01：失败 0x02：不支持	
4	Sum (1~3)	校验和	
5	0x6A	包尾	

如果模块处于睡眠状态，第一次发指令是没有响应的，此指令只是唤醒模块。

5 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。

● 产品类型(CID)描述

可根据不同的产品类型配置不同的 CID。

产品名称	产品类型 (CID)	默认产品图片
电动滑板车(蓝牙)	0x0033	
电动滑板车(4G)	0x0046	
电动自行车(蓝牙)	0x004A	
电动自行车(4G)	0x004B	

电动车(蓝牙)	0x004C	
电动车(4G)	0x004D	
电动平衡车(蓝牙)	0x004E	
电动平衡车(4G)	0x004F	

5.1 查询接收端类型及支持的功能 (Type: 00)

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	1	Payload 长度	
4	0x00	Type	Payload
5	SUM (1~4)	(1~4)校验和	
6	0x7A	包尾	

MCU 回复:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	n	Payload 长度	
4	0x00	Type	
5		Bit7	0: 仪表桥接; 1: BLE 透传 仪表桥接, 表示 BM 模块作为核心板, 搭配客户

			的底板，作为仪表功能； BLE 透传，表示 BM 模块只提供 BLE 功能，作为透传模块，和客户的仪表通过 uart 连接，做透传功能；	Payload
		Bit6	0, 预留位	
		Bit5	0, 预留位	
		Bit4	0, 预留位	
		Bit3	0, 预留位	
		Bit2	0, 预留位	
		Bit1	0, 预留位	
		Bit0	0, 预留位	
6		Bit7	定速巡航功能, 0: 不支持, 1: 支持	
		Bit6	定速巡航开关, 0: 不支持, 1: 支持	
		Bit5	零启动、非零启动切换, 0: 不支持, 1: 支持	
		Bit4	助力模式, 0: 不支持, 1: 支持	
		Bit3	0 档位, 0: 不支持, 1: 支持	
		Bit2	控制器厂商代码编号, 0: 不支持, 1: 支持	
		Bit1	电池信息, 0: 不支持, 1: 支持	
		Bit0	电池厂商代码编号, 0: 不支持, 1: 支持	
7		Bit7	0, 预留位	
		Bit6	0, 预留位	
		Bit5	0, 预留位	
		Bit4	0, 预留位	
		Bit3	0, 预留位	
		Bit2	0, 预留位	
		Bit1	0, 预留位	
		Bit0	前车灯开关, 0: 不支持, 1: 支持	
8		Bit7	自动背光调节, 0: 不支持, 1: 支持	
		Bit6-Bit0	背光调节支持等级, 比如 100 表示从 1-100 可调	
9		Bit7	手动背光调节, 0: 不支持, 1: 支持	
		Bit6-Bit0	背光调节支持等级, 比如 3 表示从 1-3 可调	
10		Bit7	APP 密码校验, 0: 不支持, 1: 支持(APP 默认是支持)绑定和 OTA 的时候是否需要密码校验	
		Bit6	档位切换, 0: 不支持, 1: 支持(APP 默认是支持)	
		Bit5	锁车开关, 0: 不支持, 1: 支持(APP 默认是支持)	
		Bit4	0, 预留位	
		Bit3	0, 预留位	
		Bit2	0, 预留位	
		Bit1	0, 预留位	
		Bit0	0, 预留位	
11		Bit7	启动设置(0 启动, 非 0 启动), 0: 不支持, 1: 支持 (APP 默认是支持)	

		Bit6	恢复出厂设置,0: 不支持, 1: 支持(APP 默认是支持)
		Bit5	0, 预留位
		Bit4	0, 预留位
		Bit3	自动关机/自动锁车, 0: 不支持, 1: 支持(APP 默认不支持)
		Bit2	0, 预留位
		Bit1	0, 预留位
		Bit0	0, 预留位
12		Bit7	0, 预留位
		Bit6	0, 预留位
		Bit5	0, 预留位
		Bit4	0, 预留位
		Bit3	0, 预留位
		Bit2	0, 预留位
		Bit1	查询设置车辆开锁密码,0: 不支持, 1: 支持(APP 默认不是支持)
		Bit0	单位切换,0: 不支持, 1: 支持(APP 默认是支持)
3+n+1	SUM	校验和	
3+n+2	0x7A	包尾	

备注: 在 APP 端连接 MCU 模块后, 可以发送此指令获取支持的功能, 以根据 MCU 模块支持的功能, 进行相应的菜单显示或者在设备绑定时发送一次此指令获取对应设备支持的功能; 协议解析不要固定长度, 根据 payload 长度解析相关字节数据, 后续如有新内容, 可能会继续增加 payload 数据长度, 已经支持的字节数据不会更改以做兼容。

5.2 心跳数据交互 (Type: 01)

数据格式:

Byte	Default	Description		
0	0xA7	包头		
1~2		产品类型 CID		
3		Payload 长度		
4	0x01	Type	Payload	
5		Bit7: 充电状态, 0: 未充电; 1: 充电中 Bit6-Bit0: 电池电量百分比(0-100)		
6		车辆实时速度高 7 位 (0.1 当前单位)		
7		车辆实时速度低 8 位 (0.1 当前单位)		
8		Bit7		0, 预留位
		Bit6		0, 预留位
		Bit5		巡航状态, 0: 未巡航, 1: 巡航中

		Bit4	公英里制单位, 0: 公里制, 1: 英里制
		Bit3	0, 预留位
		Bit2	巡航功能开关状态, 0:巡航关闭, 1: 巡航开启
		Bit1	启动方式, 0:零启动, 1: 非零启动
		Bit0	锁车状态, 0:解锁, 1: 锁车
9		Bit7	0, 预留位
		Bit6	0, 预留位
		Bit5	0, 预留位
		Bit4	0, 预留位
		Bit3	挡位起始, 0:档位从 0 开始,1:当前从 1 开始
		Bit2	当前档位(0-7)
		Bit1	
		Bit0	
10		Bit7	支持的档位个数
		Bit6	示例: 3 表示有三个档位
		Bit5	
		Bit4	
		Bit3	0, 预留位
		Bit2	0, 预留位
		Bit1	0, 预留位
		Bit0	0, 预留位
11		Bit7	0, 预留位
		Bit6	0, 预留位
		Bit5	0, 预留位
		Bit4	0, 预留位
		Bit3	0, 预留位
		Bit2	0, 预留位
		Bit1	0, 预留位
		Bit0	前车灯开关, 0: 关闭, 1: 开启
12		Bit7	0, 预留位
		Bit6	电池故障, 0: 无故障, 1: 有故障
		Bit5	控制器故障, 0: 无故障, 1: 有故障
		Bit4	电机相线或者 MOS 管短路故障, 0: 无故障, 1: 有故障
		Bit3	电机霍尔故障, 0: 无故障, 1: 有故障
		Bit2	刹车故障, 0: 无故障, 1: 有故障
		Bit1	转把故障, 0: 无故障, 1: 有故障
		Bit0	通讯故障, 0: 无故障, 1: 有故障
13		单次行驶里程高字节 (单位 0.1 当前单位)	
14		单次行驶里程低字节 (单位 0.1 当前单位)	
15		单次行驶时间高字节 (单位:秒)	
16		单次行驶时间低字节 (单位:秒)	

17		升级状态 0: 未升级 1: 仪表升级中 2: BM 模块升级中(模块作为透传时, 会有仪表和 BM 模块区分) 3: 控制器升级中
18	SUM (1~17)	(1~17)校验和
19	0x7A	包尾

注: 单次行驶里程及时间, 仪表每次开机都重置为 0, 从仪表开机后开始计单次行驶时间, 车轮转动累计里程; APP 设计中, 做单次行驶里程信息统计, 使用仪表的单次行驶里程即时间数据时, 在开启一次新的单次里程时, 使用 ‘清除 BM 模块参数’ 相关指令, 清除单次行驶里程及时间。

5.3 设置控制器状态 (Type: 02)

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	0x03	Payload 长度	
4	0x02	Type	
5		Bit7	0, 预留位
		Bit6	0, 预留位
		Bit5	0, 预留位
		Bit4	公英里制单位切换, 0: 公里制, 1: 英里制
		Bit3	0, 预留位
		Bit2	定速巡航功能, 0: 关闭巡航, 1: 开启巡航
		Bit1	启动方式, 0: 零启动, 1: 非零启动
		Bit0	0, 预留位
6	Byte5 各位有效标志, 和 Byte5 的每一位对应	Bit7	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)
		Bit6	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)
		Bit5	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)
		Bit4	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)
		Bit3	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)
		Bit2	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)
		Bit1	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)
		Bit0	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)
7	SUM (1~6)	(1~6)校验和	
8	0x7A	包尾	

MCU 回复：
原数据回复

5.4 清除 MCU 模块参数 (Type: 03)

数据格式:

Byte	Default	Description		
0	0xA7	包头		
1~2		产品类型 CID		
3	0x03	Payload 长度		
4	0x03	Type		
5		Bit7	0: 无操作, 1: 恢复出厂设置	Payload
		Bit6	0, 预留位	
		Bit5	0, 预留位	
		Bit4	0, 预留位	
		Bit3	0, 预留位	
		Bit2	0, 预留位	
		Bit1	0, 预留位	
		Bit0	清除单次里程时间, 0: 无操作, 1: 清除	
6	Byte5 各位有效标志, 和 Byte5 的每一位对应	Bit7	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)	Payload
		Bit6	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)	
		Bit5	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)	
		Bit4	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)	
		Bit3	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)	
		Bit2	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)	
		Bit1	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)	
		Bit0	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)	
7	SUM (1~6)	(1~6)校验和		
8	0x7A	包尾		

MCU 回复：
原数据回复

5.5 设置模式档位速度 (Type: 04)

数据格式:

Byte	Default	Description		
0	0xA7	包头		
1~2		产品类型 CID		
3	0x03	Payload 长度		
4	0x04	Type		
5		Bit7	0, 预留位	Payload
		Bit6	0, 预留位	
		Bit5	0, 预留位	
		Bit4	0, 预留位	
		Bit3	设置档位有效标志 0: 无操作, 1: 有效	
		Bit2	档位(1-7)	
		Bit1		
		Bit0		
6		Bit7	有效标志 0: 无操作, 1: 有效	
		Bit6	0, 预留位	
		Bit5	0, 预留位	
		Bit4	0, 预留位	
		Bit3	0, 预留位	
		Bit2	0, 预留位	
		Bit1	0, 预留位	
		Bit0	0, 预留位	
7	SUM (1~6)	(1~6)校验和		
8	0x7A	包尾		

MCU 回复:

原数据回复

5.6 设置灯光 (Type: 05)

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	0x03	Payload 长度	
4	0x05	Type	
5		Bit7	0, 预留位

		Bit6	0, 预留位	Payload
		Bit5	0, 预留位	
		Bit4	0, 预留位	
		Bit3	0, 预留位	
		Bit2	0, 预留位	
		Bit1	0, 预留位	
		Bit0	前车灯开关, 0: 关闭, 1: 开启	
6	Byte6 各位有效标志, 和 Byte5 的每一位对应	Bit7	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)	Payload
		Bit6	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)	
		Bit5	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)	
		Bit4	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)	
		Bit3	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)	
		Bit2	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)	
		Bit1	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)	
		Bit0	屏蔽位 (0: 屏蔽, 1: 有效不屏蔽)	
7	SUM (1~6)	(1~6)校验和		
8	0x7A	包尾		

MCU 回复:
原数据回复

5.7 设置解锁车密码 (Type: 06)

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	0x0D	Payload 长度	
4	0x06	Type	Payload
5		原密码第 6 位	
6		原密码第 5 位	
7		原密码第 4 位	
8		原密码第 3 位	
9		原密码第 2 位	
10		原密码第 1 位	
11		新密码第 6 位	
12		新密码第 5 位	
13		新密码第 4 位	
14		新密码第 3 位	
15		新密码第 2 位	
16		新密码第 1 位	

17	SUM (1~16)	(1~16)校验和
18	0x7A	包尾

MCU 回复:

Byte	Default	Description
0	0xA7	包头
1~2		产品类型 CID
3	2	Payload 长度
4	0x06	Type
5		结果值: 0: 成功 1: 失败 2: 不支持
6	SUM (1~5)	(1~5)校验和
7	0x7A	包尾

5.8 锁车及解锁 (Type: 07)

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	0x08	Payload 长度	
4	0x07	Type	
5		Bit7	0, 预留位
		Bit6	0, 预留位
		Bit5	0, 预留位
		Bit4	0, 预留位
		Bit3	0: 解锁;
		Bit2	1: 上锁;
		Bit1	2: 验证密码
		Bit0	3: 查询仪表开锁验证密码功能状态 4: 打开仪表开锁验证密码功能(需要携带密码) 5: 关闭仪表开锁验证密码功能(需要携带密码)
6		密码第 6 位	
7		密码第 5 位	
8		密码第 4 位	
9		密码第 3 位	
10		密码第 2 位	

11		密码第 1 位	
12	SUM (1~11)	(1~11)校验和	
13	0x7A	包尾	

MCU 回复:

Byte	Default	Description		
0	0xA7	包头		
1~2		产品类型 CID		
3	2	Payload 长度		
4	0x07	Type		
5		Bit7	Payload	
		0: 密码正确, 1: 密码错误 (锁车时, 不校验密码, 此 bit 位无效)		
		Bit6		0, 预留位
		Bit5		0, 预留位
		Bit4		0, 预留位
		Bit3		0: 解锁;
		Bit2		1: 上锁;
Bit1	2: 验证密码			
Bit0	4: 仪表开锁验证密码功能开 5: 仪表开锁验证密码功能关			
6	SUM (1~5)	(1~5)校验和		
7	0x7A	包尾		

备注: APP 端发送 查询仪表开锁验证密码功能状态 3 后, MCU 会按当前仪表端端状态返回 4 或 5:

APP 端发送 4 或 5 打开关闭仪表开锁验证密码功能时, 需携带密码, 仪表端会验证密码, 并返回当前功能状态及密码验证结果;

5.9 设置查询仪表背光状态 (Type: 09)

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	0x02	Payload 长度	
4	0x09	Type	
5		0: 查询背光状态, 1: 设置背光状态	
6		Bit7	0: 自动背光, 1: 手动背光
		Bit6	
		Bit5	

		Bit4	背光等级	Payload
		Bit3		
		Bit2		
		Bit1		
		Bit0		
7	SUM (1~5)	(1~5)校验和		
8	0x7A	包尾		

MCU 回复:

查询状态时, byte6 对应仪表当前设置数据回复; 设置状态时, 原数据回复

5.10 查询总里程 (Type: 0D)

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	1	Payload 长度	
4	0x0D	Type	Payload
5	SUM (1~4)	(1~4)校验和	
6	0x7A	包尾	

MCU 回复:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	0x05	Payload 长度	
4	0x0D	Type	Payload
5		行驶总里程(32 位无符号整数, 高字节在前, 单位:0.1km)	
6			
7			
8			
9	SUM (1~8)	(1~8)校验和	
10	0x7A	包尾	

5.11 查询电池信息 (Type: 0E)

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	1	Payload 长度	
4	0x0E	Type	Payload
5	SUM (1~4)	(1~4)校验和	
6	0x7A	包尾	

MCU 回复:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	0x0D	Payload 长度(设备不支持额定电压的时候长度会变化)	
4	0x0E	Type	Payload
5		电池温度(最高位表示正负, 0 位正, 1 为负, 单位℃)	
6		0, 预留位	
7		电流高字节(16 位有符合数, 单位 0.1A, 正数位放电电流, 负数为充电电流)	
8		电流低字节(16 位有符合数, 单位 0.1A, 正数位放电电流, 负数为充电电流)	
9		电池电压高字节(单位 0.1V)	
10		电池电压低字节(单位 0.1V)	
11		电池总容量高字节(单位 mAh)	
12		电池总容量低字节(单位 mAh)	
13		电池充放电次数高字节	
14		电池充放电次数低高字节	
15		电池额定电压高字节(单位 V)	
16		电池额定电压低字节(单位 V)	
17	SUM (1~16)	(1~16)校验和	
16	0x7A	包尾	

如果只支持部分电池信息, 不支持的部分回复:0xFF
 额定电压可能会不发送,也是代表不支持

5.12 查询电池厂商代码及编号（预留 Type: 0F）

数据格式:

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

MCU 回复:

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

5.13 查询控制器厂商代码及编号（Type: 10）

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	1	Payload 长度	
4	0x10	Type	Payload
5	SUM (1~4)	(1~5)校验和	
6	0x7A	包尾	

MCU 回复:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	9	Payload 长度	
4	0x10	Type	Payload

5		控制器厂商代码（数字或字母的 ASCII 码）	
6			
7			
8			
9		硬件版本整数字节	
10		硬件版本小数字节（1 表示 0.01）	
11		软件版本整数字节	
12		软件版本小数字节（1 表示 0.01）	
13	SUM（1~12）	(1~12)校验和	
14	0x7A	包尾	

此指令需要控制器支持，MCU 模块收到此指令，会按控制器协议转发，如果控制器支持，在 MCU 模块收到返回的版本信息，按照上述协议格式发给应用。如果控制器不支持，无返回数据

5.14 查询仪表固件版本（Type: 11）

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	1	Payload 长度	
4	0x11	Type	Payload
5	SUM（1~4）	(1~4)校验和	
6	0x7A	包尾	

MCU 回复:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	n+5	Payload 长度（n 为仪表项目代码名称长度,至少 2 字节长度）	
4	0x11	Type	Payload
5~(n+4)		项目代码（数字或字母的 ASCII 码） 示例:CM02_B05	
n+5		硬件版本整数字节	
n+6		硬件版本小数字节（1 表示 0.01）	
n+7		软件版本整数自己	

n+8		软件版本小数字节（1 表示 0.01）	
n+9	SUM（1~n+8）	(1~n+8)校验和	
n+10	0x7A	包尾	

示例：payload 中 0x11 之后的数据为 ‘C’ ‘M’ ‘0’ ‘2’ ‘_’ ‘B’ ‘0’ ‘5’ 0x01 0x00 0x0A 0x06
 解析出来 app 部分显示的版本号为 CM02_B05H1S1.0.6

由于使用 BM12 模块作为核心板，客户定制底板，导致不同客户由于外围差异会有不同的固件，不同固件会在这里有体现。比如 CM02_E03 中，E03 表示底板不同，对应的有不同的 E02 底板等。通过获取此版本，可以区分不同的项目及客户等。

除此之外，相应的有 VID、PID 的区别。

对于 AiLink 应用，可以通过这些区别，设计对应不同固件的 OTA 包获取逻辑。

Payload 长度根据不同的项目代码名称会有变化，硬件版本及软件版本固定占 4 个字节。

如不支持，无返回数据

5.15 天气信息 (Type: 13)

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3		Payload 长度	
4	0x13	Type	Payload
5	n	地址长度	
6~5+n		地址(省份, 城市 UTF8 编码)	
6+n~10+n		时间戳(大端序)	
11+n~12+n		有效期(分钟)(大端序)	
13+n		当前温度: bit7: 0-正,1-负 bit6~bit0: 温度值	
14+n		最高温度: bit7: 0-正,1-负 bit6~bit0: 温度值	
15+n		最低温度: bit7: 0-正,1-负 bit6~bit0: 温度值	
16+n		天气代码: 0x00: 未知 0x01: 晴天 0x02: 多云	

		0x03: 阴天 0x04: 小雨 0x05: 中雨 0x06: 大雨 0x07: 阵雨 0x08: 暴雨 0x09: 雷雨 0x0A: 雨夹雪 0x0B: 小雪 0x0C: 中雪 0x0D: 大雪 0x0E: 暴雪 0x0F: 雾 0x10: 沙尘暴 0x11: 雾霾	
17+n		风速(公里/小时)	
18+n		相对湿度(百分比数值)	
19+n		能见度(公里)	
20+n		紫外线强度	
21+n~22+n		雨量(大端序), 单位:mm	
23+n	SUM	校验和	
24+n	0x7A	包尾	

MCU 回复: 原数据回复

5.16 导航数据 (Type: 14)

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3		Payload 长度	
4	0x14	Type	Payload
5		0x02: 更新导航数据 0x01: 开启导航 0x00: 关闭(结束)导航(开启和关闭导航时, 其它 payload 数据全发 0x00)	
6		导航方向代码 0x00: 直行 0x01: 左转 0x02: 右转 0x03: 左侧掉头	

		0x04: 右侧掉头	
7		里程单位 0x00: 米(m) 0x01: 千米(km) 0x02: 码(yd) 0x03: 英里(mi)	
8~11		当前方向剩余里程(大端序)	
12~15		总剩余里程(大端序)	
16~17		预计花费时间(分钟)	
18~19		预留	
20	SUM	校验和	
21	0x7A	包尾	

备注：蓝牙连接断开，仪表会自动退出导航界面，如果 APP 端在导航界面，连接蓝牙后，需要重新发送开始导航指令。

APP 当前发送的当前方向剩余里程和总剩余里程单位默认是米，设备可以做超过 1km 后单位显示为 km 并进行换算

MCU 回复：原数据回复

5.17 当前时区 (Type: 15)

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3		Payload 长度	
4	0x15	Type	Payload
5		0x00: 东时区 0x01: 西时区	
6		时区: 0-12	
17~19		预留	
20	SUM	校验和	
21	0x7A	包尾	

备注：蓝牙连接上后，需要把本地时区同步给设备。

BM 回复：原数据回复

5.18 设置查询车辆开锁密码 (Type: 1A)

此密码设置给车的开机密码,与 APP 没什么关系

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	n	Payload 长度	
4	0x1A	Type	Payload
5		0: 查询, 1: 设置,	
6~n		0xFF 代表未设置,几位密码就几个 0xFF. 例如:3 为密码 123 = {0x01,0x02,0x03}	
n+1	SUM (1~n)	(1~n)校验和	
n+2	0x7A	包尾	

MCU 回复:

APP 查询的时候 Payload 长度为 0x02

查询时,设备返回旧密码不支持返回旧密码就返回 0xFE,未设置返回 0xFF,

如果安全考虑到安全问题,建议设备可以不返回旧密码,但是要返回 0xFE,支持几个密码就返回几个 0xFE

5.19 自动关机/锁车 (Type: 22)

APP 需要先查询,然后显示当前状态

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	n	Payload 长度	
4	0x22	Type	Payload
5		0x00: 查询 0x01: 设置	
6		自动关机/锁车: 0x00:关闭 0x01:开启	
7~8		时间,大端序:单位(秒) 例:1 分钟=60 秒=[0x00,0x3C]	
9	SUM (1~8)	校验和	
10	0x7A	包尾	

MCU 回复:

设备原数据回复,
 APP 查询的时候 Payload 长度为 0x02,设备回复查询并带上所有的数据

5.20 车辆密码开关 (Type: 26)

此开关主要发送给车辆,车辆用于判断是否启用和关闭密码
 当不支持车辆开锁密码时,车辆密码开关也不支持

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3		Payload 长度	
4	0x26	Type	Payload
5		0x00:查询 0x01:设置	
6		0x00:开启 0x01:关闭	
7	SUM (1~6)	校验和	
8	0x7A	包尾	

MCU 回复:

APP 查询发送 payload:0x26, 0x00

设备回复查询开启 payload:0x26, 0x00, 0x00

6 数据透传

非 AiLink 协议类指令数据,采用原数据透传。
 目前暂定控制器 OTA 的数据,采用数据透传方式。

7 OTA

根据 BM 模块作为接收端类型的角色不同，OTA 会有不同的区分。

BM 模块作为仪表桥接，OTA 会有对 BM 模块的 OTA 和控制器的 OTA；

BM 模块作为 BLE 透传，OTA 会有对 BM 模块的 OTA、客户仪表的 OTA、控制器的 OTA；

App 应用端，先查询接收端类型，确定需要支持的 OTA 类型；再查询相应的版本，确定是否要对相应的固件执行 OTA。

5.1 仪表 OTA

5.1.1 发送 OTA 开始指令(Type:0xA0)

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	11	Payload 长度	
4	0xA0	Type	Payload
5	高字节	OTA 文件长度(4 字节)	
6			
7			
8	低字节		
9	高字节	OTA 文件 crc32 校验码(4 字节)	
10		在 BM 模块作为 BLE 透传功能时有效，	
11		在 BM 模块作为仪表功能时无效	
12	低字节		
13	高字节	期望发送的 OTA 数据包最大长度(2 字	
14	低字节	节)	
15	SUM (1~14)	(1~14)校验和	
16	0x7A	包尾	

BM 回复:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	4	Payload 长度	
4	0xA0	Type	Payload
5		0:可以升级；1:不支持升级	
6	高字节	可以接收的 OTA 数据包最大长度(2 字	

7	低字节	节), 不包括协议中其他字节	
8	SUM (1~7)	(1~7)校验和	
9	0x7A	包尾	

5.1.2 发送 OTA 数据(Type:0xA1)

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	n+5	Payload 长度 (n 为 OTA 数据的长度)	
4	0xA1	Type	Payload
5	高字节	OTA 数据包编号(4 字节)	
6			
7			
8	低字节		
9~(8+n)		OTA 数据(n 字节)	
9+n	SUM (1~8+n)	(1~8+n)校验和	
10+n	0x7A	包尾	

APP 端应根据 BM 返回的可以接收的 OTA 数据包最大长度, 发送相应长度的 OTA 数据包, 比如 BM 之前返回的 OTA 数据包最大长度为 10, 一帧数据的长度不要超过 $10 + 11 = 21$

BM 回复:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	4	Payload 长度	
4	0xA1	Type	Payload
5		0:发送下一包; 1:数据错误, 请求重发	
6	高字节	接收到的 OTA 数据包编号(4 字节)	
7			
8			
9	低字节		
10	SUM (1~9)	(1~9)校验和	
11	0x7A	包尾	

接收端接收到 OTA 包数据后, 做相应处理, 处理有异常或包不正确, 可以返回请求重发的结果值, 并携带对应的数据包编号; 发送端根据数据包编号重发对应 OTA 包数据。

5.1.3 发送 OTA 完成指令(Type:0xA2)

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	1	Payload 长度	
4	0xA2	Type	Payload
5	SUM (1~4)	(1~4)校验和	
6	0x7A	包尾	

BM 回复:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	2	Payload 长度	
4	0xA2	Type	Payload
5		0:OTA 成功; 1:OTA 失败	
6	SUM (1~5)	(1~5)校验和	
7	0x7A	包尾	

注: 在 app 端发送上述 OTA 指令后, 如果模块解析数据无错误, 按上述返回相应结果; 如果模块解析数据出错, 比如接收的一帧数据中有某些字节出错, 校验不过, 即解析数据出错, 返回 A7 相应指令失败, app 端需要重发数据。

5.2 控制器 OTA

5.2.1 发送 OTA 开始指令(Type:0xA3)

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	11	Payload 长度	
4	0xA3	Type	
5	高字节	OTA 文件长度(4 字节)	
6			
7			

8	低字节	OTA 文件 crc32 校验码(4 字节) 在 BM 模块作为 BLE 透传功能时有效, 在 BM 模块作为仪表功能时无效	Payload
9	高字节		
10			
11			
12	低字节	期望发送的 OTA 数据最大包长度(2 字节)	
13	高字节		
14	低字节		
15	SUM (1~14)	(1~14)校验和	
16	0x7A	包尾	

BM 回复:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	4	Payload 长度	
4	0xA3	Type	Payload
5		0:可以升级; 1:不支持升级	
6	高字节	可以接收的 OTA 数据最大包长度(2 字节)	
7	低字节		
8	SUM (1~7)	(1~7)校验和	
9	0x7A	包尾	

5.2.2 发送 OTA 数据(Type:0xA4)

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	n+5	Payload 长度 (n 为 OTA 数据的长度)	
4	0xA4	Type	Payload
5	高字节	OTA 数据包编号(4 字节)	
6			
7			
8	低字节		
9~(8+n)		OTA 数据(n 字节)	
9+n	SUM (1~8+n)	(1~8+n)校验和	
10+n	0x7A	包尾	

BM 回复:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	4	Payload 长度	
4	0xA4	Type	Payload
5		0:发送下一包; 1:数据错误, 请求重发	
6	高字节	接收到的 OTA 数据包编号(4 字节)	
7			
8			
9	低字节		
10	SUM (1~9)	(1~9)校验和	
11	0x7A	包尾	

接收端接收到 OTA 包数据后, 做相应处理, 处理有异常或包不正确, 可以返回请求重发的结果值, 并携带对应的数据包编号; 发送端根据数据包编号重发对应 OTA 包数据。

5.2.3 发送 OTA 完成指令(Type:0xA5)

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	1	Payload 长度	
4	0xA5	Type	Payload
5	SUM (1~4)	(1~4)校验和	
6	0x7A	包尾	

BM 回复:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	2	Payload 长度	
4	0xA5	Type	Payload
5		0:OTA 成功; 1:OTA 失败	
6	SUM (1~5)	(1~5)校验和	
7	0x7A	包尾	

注: 在 app 端发送上述 OTA 指令后, 如果模块解析数据无错误, 按上述返回相应结果; 如果模块解析数据出错, 比如接收的一帧数据中有某些字节出错, 校验不过, 即解析数据出错, 返回 A7 相应指令失败, app 端需要重发数据。

8 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

Web: www.elinkthings.com