

AiLink 滑板车应用手册

版本：V0.3

更新日期：2022 年 2 月 11 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
0.1	公丕强		2021/12/30	1、初版
0.2	公丕强		2022/1/15	1、添加长供电工作模式 2、添加新指令
0.3	公丕强		2022/2/11	1、修改最大车辆支持最高速度为 5 个 bit 2、修改设置背光指令

目录

修改记录.....	- 2 -
目录.....	- 3 -
1 概述.....	- 5 -
2 说明.....	- 5 -
3 工作模式.....	- 6 -
3.1 模式一：断电模式。.....	- 6 -
3.2 模式二：长供电模式。.....	- 6 -
4 蓝牙接口（默认）.....	- 7 -
4.1 蓝牙名称：AiLink_xxxx.....	- 7 -
5 硬件通信接口.....	- 7 -
6 通用设置获取指令.....	- 7 -
6.1 读取 BLE 地址（Type: 0D）.....	- 8 -
6.2 读取模块版本号（Type: 0E）.....	- 8 -
6.3 设置模块 CIDVIDPID（Type: 1D）.....	- 9 -
6.4 读取模块 CIDVIDPID（Type: 1E）.....	- 10 -
6.5 复位模块 Type: 21）.....	- 11 -
6.6 恢复出厂设置（Type: 22）.....	- 11 -
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）.....	- 13 -
6.10 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.....	- 16 -
7 协议透传指令.....	- 18 -
7.0 查询接收端类型及支持的功能（Type: 00）.....	- 18 -
7.1 心跳数据交互（Type: 01）.....	- 20 -
7.2 设置控制器状态（Type: 02）.....	- 22 -
7.3 清除仪表参数（Type: 03）.....	- 23 -
7.4 设置模式档位速度（Type: 04）.....	- 24 -
7.5 设置灯光（Type: 05）.....	- 24 -
7.6 设置解锁车密码（Type: 06）.....	- 25 -
7.7 解锁及开关机（Type: 07）.....	- 26 -
7.8 售后密码查询（Type: 08）.....	- 27 -
7.9 设置、查询背光亮度（Type: 09）.....	- 28 -
7.10 设置自动关机时间（Type: 0A）.....	- 29 -
7.11 查询自动关机时间（Type: 0B）.....	- 30 -
7.12 查询单次行驶里程时间（Type: 0C）.....	- 31 -
7.13 查询总里程（Type: 0D）.....	- 32 -
7.14 查询电池信息（Type: 0E）.....	- 32 -

7.15	查询电池厂商代码及编号 (Type: 0F)	- 33 -
7.16	查询控制器厂商代码及编号 (Type: 10)	- 34 -
7.17	查询仪表固件版本 (Type: 11)	- 35 -
7.18	设置充电时间 (Type: 12)	- 36 -
7.19	读取充电时间 (Type: 13)	- 37 -
7.20	设置充电容量 (Type: 14)	- 37 -
7.21	读取充电容量 (Type: 15)	- 38 -
7.22	设置车辆报警、自动锁 (Type: 16)	- 38 -
7.23	读取车辆报警、自动锁 (Type: 17)	- 39 -
7.24	寻车 (Type: 18)	- 40 -
7.25	导航数据 (Type: 19)	- 40 -
7.26	天气推送 (预留 Type: 2A)	- 41 -
7.27	消息推送 (预留 Type: 2B)	- 41 -
8	数据透传	- 41 -
9	OTA	- 41 -
9.1	仪表 OTA	- 42 -
9.1.1	发送 OTA 开始指令	- 42 -
9.1.2	发送 OTA 数据	- 43 -
9.1.3	发送 OTA 完成指令	- 43 -
9.2	控制器 OTA	- 44 -
9.2.1	发送 OTA 开始指令	- 44 -
9.2.2	发送 OTA 数据	- 45 -
9.2.3	发送 OTA 完成指令	- 46 -
10	使用/测试指导	- 47 -
10.1	测试工具	- 47 -
10.2	测试步骤	- 47 -
11	自定义说明	- 47 -
12	联系我们	- 47 -
13	附录	- 47 -

1 概述

本文档适用于搭载易连物联网的 AiLink 系列模块（BM02/08/09/16/18/20/21/22...、GM02）。

本文档适用于滑板车仪表等设备搭载 BM 模块、4G 模块，接入 AiLink 应用。



请扫描此二维码下载 AiLink APP。

下文中表明的 APP 为 AiLink 应用，BLE 则为 BM 模块，4G 则为 4G 模块，MCU 为与 BM 模块或 4G 模块连接交互的芯片。

2 说明

- 2.1 A6 指令为通用设置获取指令，支持 MCU 及 AiLink 应用和 AiLink 模块交互，对 AiLink 模块进行相关的参数设置或读取。
- 2.2 A7 指令为和滑板车数据相关的协议透传指令，BM 模块将此指令透传给 MCU 或 AiLink 应用，具体协议参考相关章节描述。
- 2.3 A7 分为设置类指令和查询类指令，以及心跳包数据：
 - 设置类指令，MCU 端接收后，需回复设置结果；
 - 查询类指令，MCU 端接收后，返回指令要查询的对应数据，由 AiLink 模块透传给 AiLink 应用；
 - 和滑板车状态相关的数据在心跳包中体现。
- 2.4 AiLink 模块需要 CID、VID、PID 配置参数以接入 AiLink 应用，其中 CID 为固定 0x0033，VID 和 PID 参数需要客户联系我司进行分配。

3 工作模式

3.1 模式一：断电模式。

- 在此模式下，AiLink 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。
工作流程：

- 1、AiLink 模块上电。
- 2、AiLink 模块上电就绪后，AiLink 模块会给 MCU 返回返回模块状态。
- 3、MCU 设置 CVPID。
- 4、MCU 设置模块其他内容。
- 5、MCU 接收及发送数据。
- 6、MCU、AiLink 模块断电关机。

3.2 模式二：长供电模式。

- 在此模式下，AiLink 模块长供电电，这种模式有利于省电。
工作流程：

- 1、AiLink 模块上电。
- 2、AiLink 模块上电就绪后，AiLink 模块会给 MCU 返回返回模块状态。
- 3、MCU 设置 CVPID。
- 4、MCU 设置模块其他内容。
- 5、MCU 接收及发送数据。
- 6、MCU、AiLink 模块休眠。

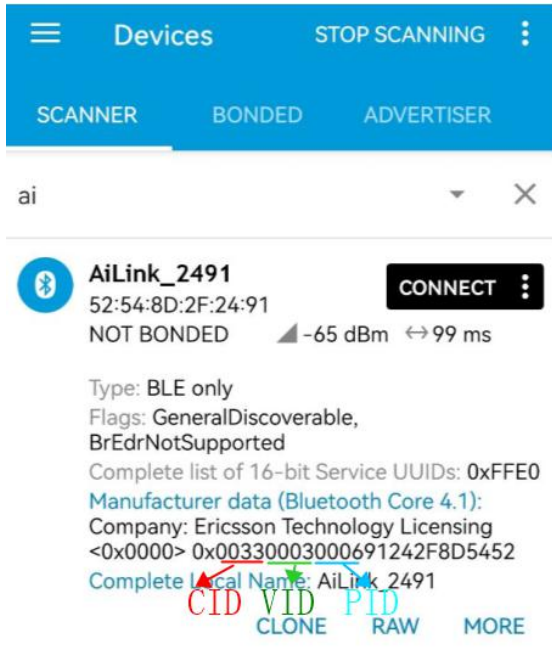
备注：

一些功能需要在模块长供电模式以及仪表 MCU 端长供电模式下才能支持，并且 MCU 端支持在休眠模式时可以通过 UART 唤醒，比如充电管理功能、自动解锁、车辆报警、寻车功能、APP 对车辆开关机等等，需要根据具体的硬件设计，确定可以支持的功能，并正确配置 7.0 章节相关的支持功能。

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



5 硬件通信接口

串口 UART：波特率 9600 ， 1 位开始位， 8 位数据位， 无停止位， 无奇偶校验位。

6 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	Payload
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	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 (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)

BM 接收：

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

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
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
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
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 BM 状态改变时, BM 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
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
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

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 置模块自动修眠时间 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	
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	Payload 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		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	
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
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
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
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
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。

7 协议透传指令

根据已定好的协议，做 MCU 和 APP 之间数据的透传。

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。

下述指令中，为表明是哪一方发送及接收的，为可以是 APP 端发送到 MCU 端，也可以是 MCU 端发送到 APP 端。

7.0 查询接收端类型及支持的功能（Type: 00）

APP 发送：

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	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，按实际长度填写
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	自动背光, 0: 不支持, 1: 支持	
	Bit5	手动背光, 0: 不支持, 1: 支持	
	Bit4	充电时间管理, 0: 不支持, 1: 支持 此功能在仪表端支持控制充电时间时, 可以通过 APP 配置充电时间段, 在配置时间段内才进行充电	
	Bit3	充电容量管理, 0: 不支持, 1: 支持 此功能在仪表端支持控制充电容量时, 可以通过 APP 配置充电容量	
	Bit2	导航数据, 0: 不支持, 1: 支持 此功能为 APP 端推送导航数据到仪表, 仪表根据导航数据做相应显示	
	Bit1	天气推送, 0: 不支持, 1: 支持 此功能为 APP 端推送天气数据到仪表, 仪表根据天气数据做相应显示	
	Bit0	消息推送, 0: 不支持, 1: 支持 此功能为 APP 端推送消息数据到仪表, 仪表根据消息数据做相应显示	
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	APP 开关机, 0: 不支持, 1: 支持 此功能为 APP 端发送开关机指令, 控制仪表开关机	
	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)	
	Bit6	0: (预留)	
	Bit5	右转灯开关, 0: 不支持, 1: 支持	
	Bit4	左转灯开关, 0: 不支持, 1: 支持	
	Bit3	氛围灯开关, 0: 不支持, 1: 支持	
	Bit2	日行灯开关, 0: 不支持, 1: 支持	
	Bit1	刹车灯开关, 0: 不支持, 1: 支持	
	Bit0	前车灯开关, 0: 不支持, 1: 支持	
8	Bit7	0: (预留)	
	Bit6	寻车功能, 0: 不支持, 1: 支持	
	Bit5	自动锁车, 0: 不支持, 1: 支持	

			此功能为手机 APP 和仪表有蓝牙连接时，车辆静置时，当手机远离车辆时，自动对车辆上锁
		Bit4	自动解锁，0：不支持，1：支持 此功能为车辆为锁车且静置状态时，当手机靠近车辆时，自动对车辆解锁
		Bit3	0：(预留)
		Bit2	0：(预留)
		Bit1	车辆移位报警，0：不支持，1：支持
		Bit0	车辆碰撞报警，0：不支持，1：支持
9	SUM	校验和	
10	0x7A	包尾	

备注：MCU 端接收到此指令后，按仪表支持的功能，配置相应的数据返回给 APP 端，APP 端可以根据相应的数据，调整 APP 相应的功能。

APP 端需要按 Payload 数据各个字节定义进行解析，注意 Payload 长度不同协议版本可能不同，方便后续新功能添加时，继续添加新的 Payload 数据，以做到版本的向前兼容。

7.1 心跳数据交互 (Type: 01)

MCU 接收:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	1	Payload 长度	
4	0x01	Type	Payload
5	SUM	校验和	
6	0x7A	包尾	

MCU 发送:

Byte	Default	Description		
0	0xA7	包头		
1~2		产品类型 CID		
3		Payload 长度，按当前版本协议实际长度填写		
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：开机状态，1：关机状态
		Bit6		0：(预留)
		Bit5	巡航状态， 0：未巡航，1：巡航中	

		Bit4	当前单位, 0: 公里制, 1: 英里制
		Bit3	0: (预留)
		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	车辆支持的最高速度, 单位 KM/H(对于可以解除限速的车辆, 解除恢复限速会有不同的最大速度, 放心跳包以便 app 对应显示)
		Bit6	
		Bit5	
		Bit4	
		Bit3	当前档位 (0-7)
		Bit2	
		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	转把有效状态, 0: 无效, 1: 有效
		Bit2	电子刹车状态, 0: 未刹车, 1: 刹车中
		Bit1	机械刹车状态, 0: 未刹车, 1: 刹车中
		Bit0	电机运行状态, 0: 电机未运行; 1: 电机运行
11		Bit7	预留
		Bit6	预留
		Bit5	右转灯开关, 0: 关闭, 1: 开启(预留)
		Bit4	左转灯开关, 0: 关闭, 1: 开启(预留)
		Bit3	氛围灯开关, 0: 关闭, 1: 开启(预留)
		Bit2	日行灯开关, 0: 关闭, 1: 开启(预留)
		Bit1	刹车灯开关, 0: 关闭, 1: 开启(预留)
		Bit0	前车灯开关, 0: 关闭, 1: 开启
12		Bit7	预留
		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 3: 控制器升级中
18	SUM	校验和
19	0x7A	包尾

注：1.对于 BM 模块，心跳数据为滑板车常用状态数据，MCU 端定时 100ms 左右发送一次心跳数据给 APP 端，不需要 BM 模块发送；

2.对于 4G 模块，考虑到流量限制，在 MCU 端收到请求心跳数据后再发送心跳数据到 4G 模块；

3.关于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，车轮转动累计单次里程；MCU 端收到‘清除仪表参数’相关指令，清除单次里程及时间时，MCU 端重新开始计算单次里程及时间，并在心跳中发送。

4.APP 端需要按 Payload 数据各个字节定义进行解析，注意 Payload 长度不同协议版本可能不同，方便后续新功能添加时，继续添加新的 Payload 数据，以做到版本的向前兼容。

7.2 设置控制器状态 (Type: 02)

APP 发送:

Byte	Default	Description		
0	0xA7	包头		
1~2		产品类型 CID		
3	0x03	Payload 长度		
4	0x02	Type		
5		Bit7	Payload	
		Bit6		
		Bit5		
		Bit4		公英里制单位切换, 0: 公里制, 1: 英里制
		Bit3		(预留)
		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		(预留)
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	校验和
8	0x7A	包尾

MCU 回复：
原数据回复

7.3 清除仪表参数 (Type: 03)

APP 发送:

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
		Bit6	预留
		Bit5	预留
		Bit4	预留
		Bit3	预留
		Bit2	清除除里程和时间外的其它控制器及仪表参数 0: 无操作, 1: 清除
		Bit1	清除总里程, 0: 无操作, 1: 清除总里程
		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)
		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	校验和	
8	0x7A	包尾	

MCU 回复：
原数据回复

7.4 设置模式档位速度 (Type: 04)

APP 发送:

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

BM 回复:

原数据回复

7.5 设置灯光 (Type: 05)

APP 发送:

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

		Bit6	预留	Payload
		Bit5	右转灯开关, 0: 关闭, 1: 开启(预留)	
		Bit4	左转灯开关, 0: 关闭, 1: 开启(预留)	
		Bit3	氛围灯开关, 0: 关闭, 1: 开启(预留)	
		Bit2	日行灯开关, 0: 关闭, 1: 开启(预留)	
		Bit1	刹车灯开关, 0: 关闭, 1: 开启(预留)	
		Bit0	前车灯开关, 0: 关闭, 1: 开启	
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	校验和		
8	0x7A	包尾		

MCU 回复:
原数据回复

7.6 设置解锁车密码 (Type: 06)

APP 发送:

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	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	Payload
5		结果值: 0: 成功 1: 失败 2: 不支持	
6	SUM	校验和	
7	0x7A	包尾	

注: 仪表出厂锁车密码需设置为 000000, 其中 0 为字符, 对应协议数据中的 byte 为 0x30

7.7 解锁及开关机 (Type: 07)

APP 发送:

Byte	Default	Description		
0	0xA7	包头		
1~2		产品类型 CID		
3	0x08	Payload 长度		
4	0x07	Type	Payload	
5		Bit7		
		Bit6		
		Bit5		
		Bit4		
		Bit3		0: 解锁;
		Bit2		1: 上锁;
		Bit1		2: 验证密码;
		Bit0		3: 开机; 4: 关机;
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	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	Payload	
5		Bit7		0: 密码正确, 1: 密码错误 (锁车时, 不校验密码, 此 bit 位无效)
		Bit6		
		Bit5		
		Bit4		
5		Bit3		0: 解锁;
		Bit2		1: 上锁;
		Bit1		2: 验证密码
		Bit0		
6	SUM	校验和		
7	0x7A	包尾		

7.8 售后密码查询 (Type: 08)

APP 发送:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	1	Payload 长度	
4	0x08	Type	Payload
5	SUM	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	0x07	Payload 长度

4	0x08	Type	Payload
5		售后密码第 6 位	
6		售后密码第 5 位	
7		售后密码第 4 位	
8		售后密码第 3 位	
9		售后密码第 2 位	
10		售后密码第 1 位	
11	SUM	校验和	
12	0x7A	包尾	

注：在用户忘记设置的锁车密码时，可以通过 APP 端的售后密码查询功能，查询设置的锁车密码，APP 通过该指令发送给 MCU，MCU 需要返回售后密码，售后密码为锁车密码进行加密后的数据，加密算法请联系我司获取。

7.9 设置、查询背光亮度（Type: 09）

发送：

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	0x02	Payload 长度	
4	0x09	Type	
5		0x00: 查询 0x01: 设置	
6		Bit7	背光状态：0，手动调光状态，1，自动调光状态 背光亮度百分比 (0-100)
		Bit6	
		Bit5	
		Bit4	
		Bit3	
		Bit2	
		Bit1	
		Bit0	
7	SUM	校验和	
8	0x7A	包尾	

回复：

Byte5 为 0x00 时，为查询指令，由 APP 端发送，MCU 端需回复当前背光状态及亮度，回复数据按上述数据结构回复，Byte5 为 0x00；

Byte5 为 0x01 时，为设置指令，接收端根据 Byte6 数据做相应处理，原数据回复；

7.10 设置自动关机时间 (Type: 0A)

发送:

Byte	Default	Description		
0	0xA7	包头		
1~2		产品类型 CID		
3	0x03	Payload 长度		
4	0x0A	Type	Payload	
5		Bit7		开锁状态下, 自动关机时间高 8 位
		Bit6		
		Bit5		
		Bit4		
		Bit3		
		Bit2		
		Bit1		
		Bit0		
6		Bit7		开锁状态下, 自动关机时间低 8 位
		Bit6		
		Bit5		
		Bit4		
		Bit3		
		Bit2		
		Bit1		
		Bit0		
7		Bit7	锁车状态下, 自动关机时间高 8 位	
		Bit6		
		Bit5		
		Bit4		
		Bit3		
		Bit2		
		Bit1		
		Bit0		
8		Bit7	锁车状态下, 自动关机时间低 8 位	
		Bit6		
		Bit5		
		Bit4		
		Bit3		
		Bit2		
		Bit1		
		Bit0		
9	SUM	校验和		

10	0x7A	包尾
----	------	----

回复:

原数据回复

注：在仪表端有实现车辆静置无操作超时一定时间后自动关机功能时，可以通过该指令由用户修改超时关机时间，仪表未实现时，MCU 端不处理；

7.11 查询自动关机时间 (Type: 0B)

APP 发送:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	1	Payload 长度	
4	0x0B	Type	Payload
5	SUM	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	0x03	Payload 长度		
4	0x0B	Type	Payload	
5		Bit7		
		Bit6		
		Bit5		
		Bit4		
		Bit3		开锁状态下，自动关机时间高 7 位
		Bit2		
		Bit1		
		Bit0		
6		Bit7		
		Bit6		
		Bit5		
		Bit4		
		Bit3		开锁状态下，自动关机时间低 8 位
		Bit2		
		Bit1		

		Bit0		
7		Bit7	锁车状态下，自动关机时间高 7 位	
		Bit6		
		Bit5		
		Bit4		
		Bit3		
		Bit2		
		Bit1		
		Bit0		
8		Bit7	锁车状态下，自动关机时间低 8 位	
		Bit6		
		Bit5		
		Bit4		
		Bit3		
		Bit2		
		Bit1		
		Bit0		
9	SUM	校验和		
10	0x7A	包尾		

注：在仪表端有实现车辆静置无操作超时一定时间后自动关机功能时，可以通过该指令由用户修改超时关机时间，仪表未实现时，MCU 端不处理；

7.12 查询单次行驶里程时间（Type: 0C）

APP 发送:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	1	Payload 长度	
4	0x0C	Type	Payload
5	SUM	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	0x0C	Type	Payload
5		单次行驶里程高字节(单位:0.1km)	

6		单次行驶里程低字节	
7		单次行驶时间高字节(单位:秒)	
8		单次行驶时间低字节	
9	SUM	校验和	
10	0x7A	包尾	

7.13 查询总里程 (Type: 0D)

APP 发送:

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	校验和	
10	0x7A	包尾	

7.14 查询电池信息 (Type: 0E)

APP 发送:

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	0x0B	Payload 长度	
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		Bit0: 0, 实时电流; 1: 电池限流 其它位备用	
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	SUM	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 FF

7.15 查询电池厂商代码及编号 (Type: 0F)

APP 发送:

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	Payload
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	0x0F	Type	
5~(n+4)		厂商代码编号 (数字或字母的 ASCII 码) 示例: BATT_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	

7.16 查询控制器厂商代码及编号 (Type: 10)

APP 发送:

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	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	

7.17 查询仪表固件版本 (Type: 11)

APP 发送:

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		软件版本整数字节	
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	

7.18 设置充电时间 (Type: 12)

发送:

Byte	Default	Description		
0	0xA7	包头		
1~2		产品类型 CID		
3	6	Payload 长度		
4	0x12	Type	Payload	
5	0x00	Bit7		1: 设置周日充电 0: 不设置周日充电
		Bit6		1: 设置周六充电 0: 不设置周六充电
		Bit5		1: 设置周五充电 0: 不设置周五充电
		Bit4		1: 设置周四充电 0: 不设置周四充电
		Bit3		1: 设置周三充电 0: 不设置周三充电
		Bit2		1: 设置周二充电 0: 不设置周二充电
		Bit1		1: 设置周一充电 0: 不设置周一充电
		Bit0	1: 设置单次充电时间, 有效时, Bit1-Bit7 无效 0: 设置重复充电时间, 根据 Bit1-Bit7 位解析	
6	0x00	充电开始时间小时(24 小时制)		
7	0x00	充电开始时间分钟		
8	0x00	充电结束时间小时(24 小时制)		
9	0x00	充电结束时间分钟		
10	SUM	校验和		
11	0x7A	包尾		

回复:

原数据回复

备注:

Byte5-Byte9 都为 0x00, 即关闭充电时间功能;

当设置了充电时间后, 充电器有连接时, 在到达设置的开始时间后, 开始充电; 到达设置的结束时间后, 停止充电;

设置的充电时间需要 MCU 端可以保存到 flash，即 APP 设置过充电时间后，如果未重新设置充电时间，需要以保存的数据做充电时间的管理；

如果在仪表端支持设置充电时间，比如通过按键或屏幕操作，在设置完成后，需要按此指令格式数据发送给 APP 端，以使 APP 端同步更新；

7.19 读取充电时间（Type: 13）

APP 发送:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	1	Payload 长度	
4	0x13	Type	Payload
5	SUM	校验和	
6	0x7A	包尾	

MCU 回复:

按上述“设置充电时间”指令数据格式回复

7.20 设置充电容量（Type: 14）

APP 发送:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	2	Payload 长度	
4	0x14	Type	Payload
5	0x64	电池电量到达设置百分比后，停止充电	
6	SUM	校验和	
7	0x7A	包尾	

MCU 回复:

原数据回复

备注:

设置的充电容量，需要 MCU 端可以保存到 flash，即 APP 设置过充电容量后，如果未重新设置充电容量，需要以保存的数据做充电容量的管理；

如果在仪表端支持设置充电容量，比如通过按键或屏幕操作，在设置完成后，需要按此指令格式数据发送给 APP 端，以使 APP 端同步更新；

7.21 读取充电容量 (Type: 15)

APP 发送:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	1	Payload 长度	
4	0x15	Type	Payload
5	SUM	校验和	
6	0x7A	包尾	

MCU 回复:

按上述“设置充电容量”指令数据格式回复

7.22 设置车辆报警、自动锁 (Type: 16)

发送:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	3	Payload 长度	
4	0x16	Type	Payload
5	0x00	Bit7	0: (预留)
		Bit6	寻车功能 0: 关闭 1: 打开
		Bit5	自动锁车 0: 关闭 1: 打开
		Bit4	自动解锁 0: 关闭 1: 打开
		Bit3	0: (预留)
		Bit2	0: (预留)
		Bit1	移位报警开关 0: 关闭 1: 打开
		Bit0	碰撞报警开关 0: 关闭

			1: 打开	
6	0x00(对应 Byte5 的屏蔽位)	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	校验和		
8	0x7A	包尾		

回复:

原数据回复

备注: 在仪表端支持设置车辆报警开关, 比如通过按键或屏幕操作, 在设置完成后, 需要按此指令格式数据发送给 APP 端, 以使 APP 端同步更新;

7.23 读取车辆报警、自动锁 (Type: 17)

APP 发送:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	1	Payload 长度	
4	0x17	Type	Payload
5	SUM	校验和	
6	0x7A	包尾	

MCU 回复:

按上述“设置车辆报警、自动锁”指令数据格式回复

7.24 寻车 (Type: 18)

APP 发送:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	2	Payload 长度	
4	0x18	Type	Payload
5	0x01	0x01: 寻车默认参数, 后续版本可能扩展	
6	SUM	校验和	
7	0x7A	包尾	

MCU 回复:

原数据回复

7.25 导航数据 (Type: 19)

APP 发送:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	9	Payload 长度	
4	0x19	Type	Payload
5		导航状态 0: 未开启 1: 开启	
6		导航方向代码 1, 直行 2, 向左转弯 3, 向右转弯 4, 左侧掉头 5, 右侧掉头	
7		当前方向剩余路程高 8 位, 单位 M	
8		当前方向剩余路程低 8 位, 单位 M	
9		总剩余路程 24-32 位, 单位 M	
10		总剩余路程 16-23 位, 单位 M	
11		总剩余路程 8-15 位, 单位 M	
12		总剩余路程 0-8 位, 单位 M	
13	SUM	校验和	

14	0x7A	包尾
----	------	----

MCU 回复:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID	
3	3	Payload 长度	
4	0x19	Type	Payload
5		导航状态 0: 未开启 1: 开启	
6		0x00: 接收成功 0x01: 接收失败	
7	SUM	校验和	
8	0x7A	包尾	

备注:

1. 在开始导航后, APP 端指令中, 导航状态位置 1, 可以不确认 MCU 端回复; MCU 端收到导航状态数据, 可以不回复 APP 端;
2. 在结束导航时, APP 需要发送导航状态位置 0 指令数据, 需要确认 MCU 端回复; MCU 端收到导航状态为 0 指令时, 处理相应逻辑, 并回复接收

7.26 天气推送 (预留 Type: 2A)

7.27 消息推送 (预留 Type: 2B)

8 数据透传

非 AiLink 协议类指令数据, 采用原数据透传。

9 OTA

APP 可以对仪表和控制器进行 OTA 升级, 同时需要仪表端 MCU 以及控制器支持 OTA 升级功能。如果要进行 OTA, 需要按协议透传指令中相应指令返回正确的相应版本; 并提供相应的 OTA 升级文件由我司做 OTA 服务器端的相应配置。

OTA 固件为单个二进制升级 bin 文件, 按下述协议透传到 MCU 端, 由 MCU 端按照协议进行相应的处理, 以完成 OTA 升级。

注：在 app 端发送下述 OTA 指令后，如果 MCU 解析数据无错误，按下述返回相应结果；如果 MCU 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。

9.1 仪表 OTA

9.1.1 发送 OTA 开始指令

APP 发送:

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	crc32 计算方法由我司提供	
13	高字节	期望发送的 OTA 数据包最大长度(2 字节)	
14	低字节		
15	SUM (1~14)	(1~14)校验和	
16	0x7A	包尾	

MCU 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 字节)	
7	低字节	不包括协议中其他字节，不超过 APP 发送的期望最大包长度	
8	SUM (1~7)	(1~7)校验和	
9	0x7A	包尾	

9.1.2 发送 OTA 数据

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
5	高字节	Payload
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

MCU 回复:

Byte	Default	Description
0	0xA7	包头
1~2		产品类型 CID
3	4	Payload 长度
4	0xA1	Type
5		Payload
6	高字节	
7		
8		
9	低字节	接收到的 OTA 数据包编号(4 字节)
10	SUM (1~9)	(1~9)校验和
11	0x7A	包尾

MCU 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。

9.1.3 发送 OTA 完成指令

APP 发送:

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	

MCU 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	

9.2 控制器 OTA

9.2.1 发送 OTA 开始指令

APP 发送:

Byte	Default	Description		
0	0xA7	包头		
1~2		产品类型 CID		
3	11	Payload 长度		
4	0xA3	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	crc32 计算方法由我司提供		
13	高字节	期望发送的 OTA 数据最大包长度(2 字节)		
14	低字节			
15	SUM (1~14)	(1~14)校验和		
16	0x7A	包尾		

MCU 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), 不包括协议中其他字节, 不超过 APP 发送的期望最大包长度	
7	低字节		
8	SUM (1~7)	(1~7)校验和	
9	0x7A	包尾	

9.2.2 发送 OTA 数据

APP 发送:

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	

MCU 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。

9.2.3 发送 OTA 完成指令

APP 发送:

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	

MCU 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	

10 使用/测试指导

10.1 测试工具

10.2 测试步骤

11 自定义说明

12 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

13 附录