

AiLink WIFIBLE 系列环境监测仪应用手册

版本：V1.6

更新日期：2023 年 04 月 13 日

深圳市易连物联网有限公司版权所有

本产品的应用说明书如有变更，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	审核人	发布日期	修改说明	审核	批准
V1.0	LYX	Lxl, hjb	2022/8/8	1. 初版		
V1.1	LYX	lxl	2022/9/6	1. 增加使能 mqtt 连接 ailink 指令, MCU 在交互 2. 修改工作流程: 连接 wifi 成功后, 需要使能 mqtt 连接		
V1.2	LYX		2022/11/7	1. 增加 CO 的功能		
V1.3	Hgb		2022/11/22	1. CO 统一 TYPE 为 0x15		
V1.4	LYX		2022/11/29	增加多个 TYPE 参数		
V1.5	GPQ		2023/02/23	添加及修改报警开关功能; 添加是否支持温度单位切换功能声明		
V1.6	LXL		2023/4/13	修正指令描述		

目录

修改记录	2
目录	3
1 概述	5
2 说明	5
3 模块版本	6
4 模块选型	6
5 硬件参考设计	6
5.1 串口 UART	6
5.2 硬件连接	6
6 蓝牙接口（默认）	7
6.1 蓝牙名称：AiLink_XXXX	7
7 流程及软件协议	8
7.1 产品定义	9
7.1.1 产品形态	9
7.1.2 说明:	9
7.2 产品工作流程	10
7.3 工作流程图	10
7.4 基础交互指令	11
7.4.1 通信 TLV 的格式声明	11
7.4.2 APP 获取设备功能列表	26
7.4.3 APP 获取设备状态	27
7.4.4 APP 设置/获取参数	28
7.4.5 MCU 上发保存数据	30
7.4.6 APP 与模块心跳	31
8 模块通用指令集	32
8.1 APP 同步时间到 MCU（CMD: 0x37、0x38）	32
8.2 设置模块立即进入休眠 CMD = 0x19	33
8.3 设置模块自动休眠时间 CMD = 0x17	34
8.4 设置、读取 CID、VID、PID（CMD: 0x1D、0x1E）	35
8.5 设置模块重启（CMD: 0x21）	37
8.6 设置恢复出厂设置（CMD: 0x22）	37
8.7 获取 WM 模块状态(CMD = 0x26,0xAB)	38
8.8 请求 Unix 时间 CMD = 0x44	39
8.9 设置 Unix 时间 CMD = 0x45	40
8.10 查询/使能 mqtt 连接状态 CMD = 0xC5	40
9 举例说明	42
10 测试指导	42
10.1 连接测试	42
10.2 功能测试	43

11 生产测试指导 44
12 联系我们 44

1 概述

- 1.1 本文档适用于深圳市易连物联网 WIFIBLE 模块(WM 模块)接入 ailink APP。
- 1.2 本文档适用于环境监测设备的 MCU 端开发工程师使用。
- 1.3 本文档讲详细介绍硬件对接、固件对接。
- 1.4 文档会保持更新，以[官网链接](#)为最新版本。

2 说明

- 2.1 我们提供标准化的连接模块、app、云平台帮助客户的环境监测设备快速实现智能化，并提供 sdk、云平台配置、增值服务和技术支持帮忙客户差异化、个性化。
- 2.2 我们提供的模块具有功耗低、认证齐全、APP 功能强大体验好等特点。扫描下面二维码下载 APP。



- 2.3 支持 MCU 配置模块（VID、PID）实现 APP 连接产品时型号自定义、图标自定义等个性化设计。
- 2.4 传统环境监测设备加上 WIFIBLE 模块后,即可实现产品智能化升级:
 - 2.4.1 传统款的空气质量检测只能提供当前测量数据，无法进行数据对比分析，无法判断当前环境状态，智能款可以通过手机 APP 或者大数据平台，对数据进行判断分析；
 - 2.4.2 标准判断：当 APP 连接上设备后读取当前设备数据值，根据预置的数据判断标准或者是通过网络查询，准备的判断出当前检测值的状态， APP 中会告诉我们现在的空气状况及建议，同时我们在 APP 中也能了解到室外的空气质量状况，来进行室内外的对比；
 - 2.4.3 数据分析：APP 将设备所有检测的数据，汇总成历史数据，用户可以导出图表，对数据进行分析，可以清晰的了解数据的变化过程；

3 模块版本

本文档支持的固件版本：
WM07H1S4.0.0xxx

4 模块选型

- 针对于环境监测设备产品的功能,主要的对比点及差异.

参数 \ 型号	WM05	WM07	WM09	WM16
WiFi 协议	11b	11b	11b/g/n	11b/g/n
主频	176MHz	176MHz	196MHz	196MHz
flash	2MByte	1MByte	2MByte	2MByte
最大发射功率	10db	10db	21db	21db
功耗(深度睡眠)	22uA	20uA	2.7uA	2.6uA
功耗(闲时工作)	7.2mA	12.3mA	35mA	34.7mA
功耗(峰值)	200mA	225mA	310mA	310mA
BLE 版本	5.0	5.0	5.0	5.0

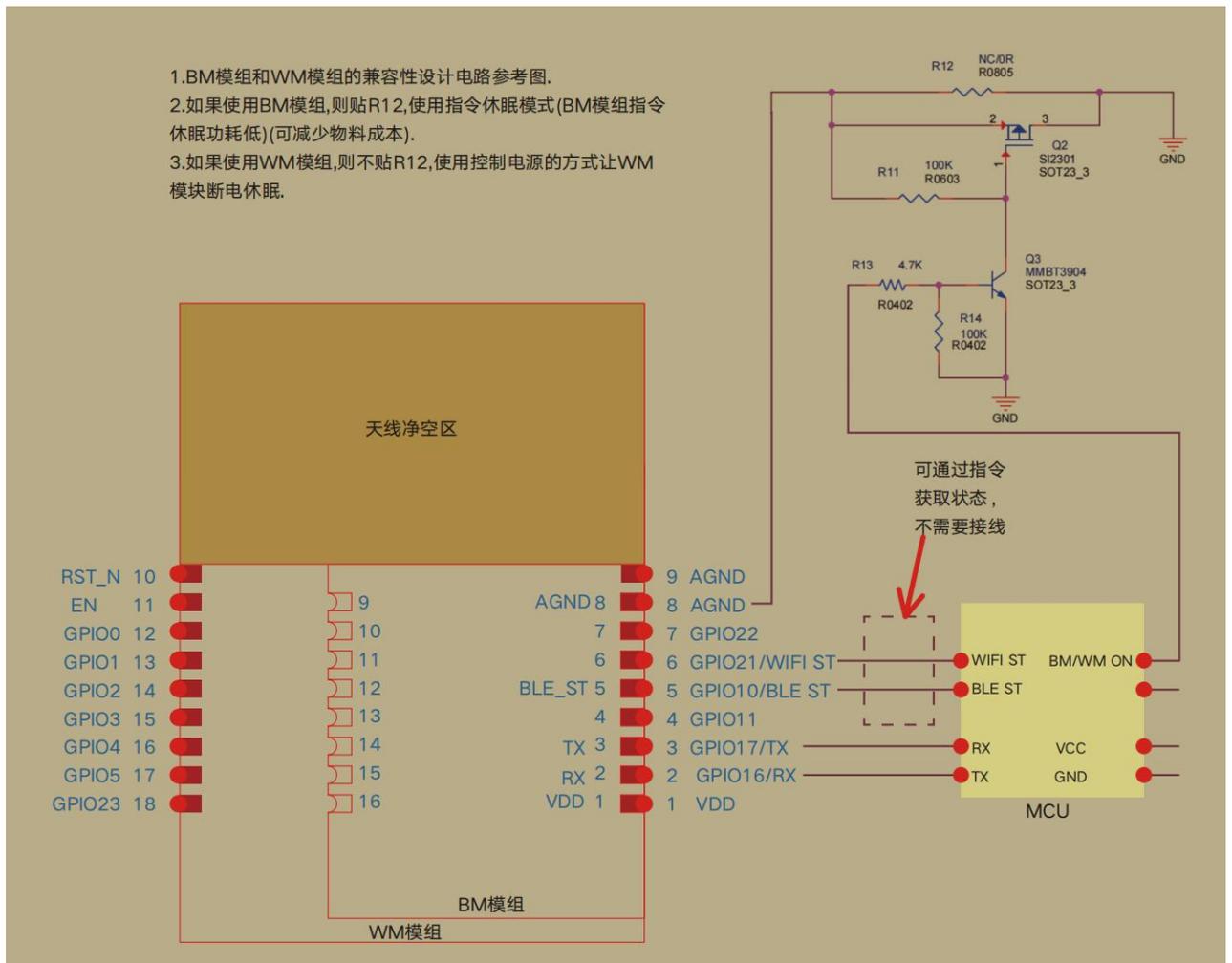
5 硬件参考设计

5.1 串口 UART

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

5.2 硬件连接

5.2.1 参考电路:



5.2.2 说明

5.2.2.1 产品智能化升级, 不仅可以通过接入 BLE 模块, 也可以通过 wifi 模块. 我司提供了这两种方案的技术支持. 因此, 在产品的设计阶段, 建议 PCB 可以进行兼容式设计. 以后 BLE 或者 WiFi 方案相互替换时, 则不需再重新设计 PCB 版.

6 蓝牙接口 (默认)

6.1 蓝牙名称: AiLink_xxxx

注: xxxx 为 Mac 地址后 4 个字符

☰ Devices STOP SCANNING ⋮

SCANNER BONDED ADVERTISER

112, -84 dBm ▾ ✕

 **AiLink_6112** **CONNECT** ⋮
02:11:23:34:61:12
NOT BONDED ▲ -52 dBm ↔ 33 ms

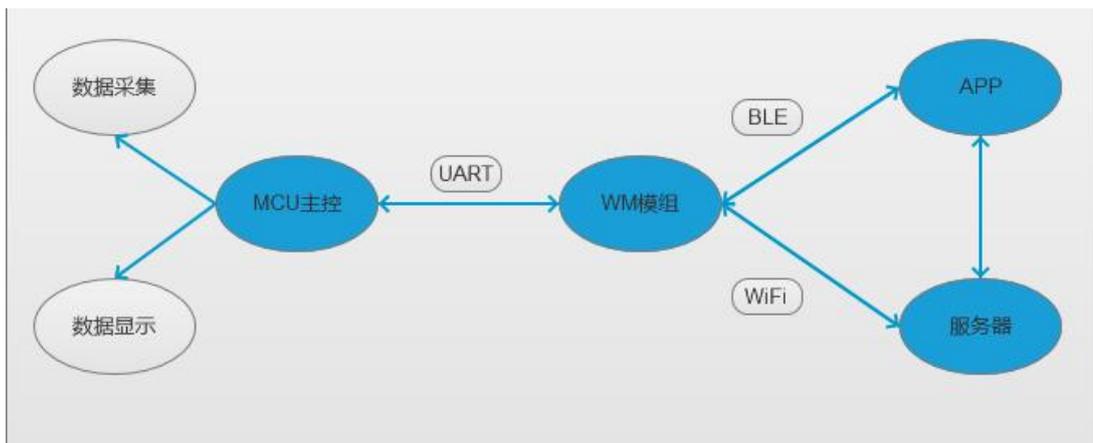
Type: BLE only
Flags: GeneralDiscoverable,
BrEdrNotSupported
Complete list of 16-bit Service UUIDs: 0xFFE0,
0xFEE0
Manufacturer data (Bluetooth Core 4.1):
Company: Reserved ID <0x496E>
0x000400010001126134231102
Complete Local Name: AiLink_6112

CID **VID** **PID** CLONE RAW MORE

7 流程及软件协议

7.1 产品定义

7.1.1 产品形态



7.1.2 说明:

1. 本产品包含三部分:
 - (1) MCU 主控 , WM 传输模块 , APP/服务器
 - MCU 主控:
MCU 主控主要负责环境数据的采集,数据的显示,以及通过 UART 连接把数据传输到 WM 模块.
 - WM 模块:
WM 模块主要作为 APP/服务器和 MCU 的沟通桥梁,把接收到的环境数据传输到 APP/服务器上.
 - APP/服务器:
APP 通过与 WM 模块连接,与主机绑定,获取环境数据信息,设置功能等.
2. WM 模块数据传输原则:
 - (1) 在 APP 通过蓝牙连接设备时,数据直接通过蓝牙传输数据.
 - (2) APP 没有连接到设备时,设备数据通过 WiFi 传输到 APP 上.

7.2 产品工作流程

1. 设备上电,给 WM 模块通电
2. WM 模块上电成功后,会主动返回状态(CMD=0x26 指令).MCU 主控收到状态后,证明可以进行 UART 读写操作.
3. MCU 通过 UART 设置产品的 CID,VID,PID (必须设,否则 AiLink APP 无法找到设备). AiLink CID VID PID 获取介绍: http://doc.elinkthings.com/web/#/40?page_id=144
4. 打开 APP,搜索设备,绑定设备.
5. APP 连接上设备后,会发起绑定指令,请求用户按按键确认,设备绑定后,设备上发的数据即可在 APP 上显示.
6. 打开配网界面,对设备进行配网.
7. 模块返回配网成功指令(CMD = 0x26 指令)后, MCU 需要发送使能 MQTT 连接指令(CMD = 0xC5 指令)。
8. WM 模块没有通过蓝牙连接 APP 时,WM 模块会通过 WiFi 通过 mqtt 协议传输数据到 APP 上.
9. MCU 传输数据到 APP 上.
10. APP 进行各项的参数设置.
11. WM 状态指令查看模块是否连接上路由器,mqtt 状态指令查看模块是否连接到服务器.
12. 当设备关机时,断开设备供电.

7.3 工作流程图

7.4 基础交互指令

- MCU 和模块通信时, 由于数据是需要通过 WiFi 的 mqtt 协议传输到服务器的, 因此 MCU 可能通过模块返回结果判断是否已上发到服务器
- 当数据状态出现变化时, 例如数据变化, 或者认为操作, 或者报警等, 才需要更新给模块.
- 数据上发时, 需要分帧传输.

Byte	Value	Description
0	0xA7	包头
1	0x00	产品类型 (CID) 高字节
2	0x48	产品类型 (CID) 低字节
3		Payload 长度(payload 部分的字节数量)
4		Payload
5~n		
n+1	SUM (1~n)	(1~n) 校验和(累加和, 取低八位)
n+2	0x7A	包尾

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

7.4.1 通信 TLV 的格式声明.

针对于设备有该项功能时, 设备端在声明功能时 , 才需声明 , 若设备端无该项功能, 则不需理会.

列表:

Type	Length(后面 Value 的 byte 数)	Value(小端序)
		如无特殊说明 , 定义数据超过 1 个 byte 的数据, 都使用小端序格式 例如: 数值为十进制数 1259 , 对应的十六进制为 0x04EB , 小端序格式的, 则先发 0xEB , 再发 0x04
0x80: 协议版本		
0x01: 甲醛		
0x02: 温度		
0x03: 湿度		
0x04: PM2.5		
0x05: PM1.0		
0x06: PM10		
0x07: VOC		
0x08: CO2		

0x09:AQI		
0x0A:报警功能		
0x0B:音量		
0x0C:报警时长		
0x0D:报警铃声		
0x0E:设备故障		
0x0F:设备自检		
0x10:TVOC		
0x11:单位切换		
0x12:电池状态		
0x13:设备绑定		
0x14:心跳包(用以模块和APP通信)		
0x15:CO(一氧化碳)		
0x16:闹钟功能		
0x17:恢复出厂设置		
0x18:参数校准		
0x19:时间格式		
0x1A:设备亮度		
0x1B:按键音效		
0x1C:报警音效		
0x1D:图标显示		
0x1E:监控显示数据		
0x1F:数据显示模式		

功能声明		
Type	Length(后面Value的byte数)	Value(小端序)
0x80:协议版本	1	Feature(1byte): 0x01 该功能必须声明
0x01:甲醛	5	Feature(1byte): [Bit0-Bit1:小数点位数(数值要根据小数点来计算.例如,最小值为1.00,即小数点为2,Min=100)] Min:甲醛的最小值(2bytes)[..] Max:甲醛的最大值(2bytes)[..]
0x02:温度	5	Feature(1byte): [Bit0-Bit1:小数点位数

		Bit2: 最小值的正负号(0=正值,1=负值)] Min:温度的最小值(2bytes)[单位°C..] Max:温度的最大值(2bytes)[单位°C..]
0x03:湿度	5	Feature(1byte): [Bit0-Bit1:小数点位数] Min:最小值(2bytes)[..] Max:最大值(2bytes)[..]
0x04:PM2.5	5	Feature(1byte): [Bit0-Bit1:小数点位数] Min:最小值(2bytes)[..] Max:最大值(2bytes)[..]
0x05:PM1.0	5	Feature(1byte): [Bit0-Bit1:小数点位数] Min:最小值(2bytes)[..] Max:最大值(2bytes)[..]
0x06:PM10	5	Feature(1byte): [Bit0-Bit1:小数点位数] Min:最小值(2bytes)[..] Max:最大值(2bytes)[..]
0x07:VOC	5	Feature(1byte): [Bit0-Bit1:小数点位数] Min:最小值(2bytes)[..] Max:最大值(2bytes)[..]
0x08:CO2	5	Feature(1byte): [Bit0-Bit1:小数点位数] Min:最小值(2bytes)[..] Max:最大值(2bytes)[..]
0x09:AQI	5	Feature(1byte): [Bit0-Bit1:小数点位数] Min:最小值(2bytes)[..] Max:最大值(2bytes)[..]
0x0A:报警功能	1	0x01:表示支持报警功能,且各报警功能可单独设置开关; 0x02:表示支持报警功能,但只支持所有传感器数据

		报警功能同时打开和关闭;																											
0x0B:音量	1	Max:音量最大值(1bytes)																											
0x0C:报警时长	2	Max:报警时长最大值(2bytes,单位:s)																											
0x0D:报警铃声	1	Max:铃声的数量(1byte)																											
0x0E:设备故障	1	0x01:常量(表示支持上发设备故障功能)																											
0x0F:设备自检	1	0x01:常量(表示支持上发设备故障功能)																											
0x10:TVOC	5	Feature(1byte): [Bit0-Bit1:小数点位数] Min:最小值(2bytes)[..] Max:最大值(2bytes)[..]																											
0x11:单位切换	2	Feature(1byte): [Bit0: 0=不支持温度单位单位, 1=支持温度单位切换] 预留(1byte)																											
0x15:CO(一氧化碳)	5	Feature(1byte): [Bit0-Bit1:小数点位数] Min:最小值(2bytes)[..] Max:最大值(2bytes)[..]																											
0x16:闹钟功能声明	3	Feature1 (3byte): <table border="1" style="margin-left: 20px;"> <tr> <td rowspan="5">Byte0</td> <td>Bit7</td> <td>0=闹钟不设置不显示 1=直接显示所有闹钟 针对于有固定闹钟图标的设备,不开闹钟也会有相应的符号,可以选择 1</td> </tr> <tr> <td>Bit6</td> <td>0: 不支持添加及删除闹钟功能; 1: 支持添加删除闹钟功能;</td> </tr> <tr> <td>Bit5</td> <td>保留,为 0</td> </tr> <tr> <td>Bit4</td> <td>保留,为 0</td> </tr> <tr> <td>Bit3-Bit0</td> <td>闹钟数量,最多 15 个</td> </tr> <tr> <td rowspan="6">Byte1</td> <td>Bit7</td> <td>模式 0:一次性,当天有效 0=不支持该项功能 1 =支持该项功能</td> </tr> <tr> <td>Bit6</td> <td>模式 1:每天都响</td> </tr> <tr> <td>Bit5</td> <td>模式 2:周一至周五</td> </tr> <tr> <td>Bit4</td> <td>模式 3:周一至周六</td> </tr> <tr> <td>Bit3</td> <td>模式 4:自定义,从周一到周日任意搭配</td> </tr> <tr> <td>Bit2</td> <td>保留,为 0</td> </tr> <tr> <td></td> <td>Bit1</td> <td>保留,为 0</td> </tr> </table>	Byte0	Bit7	0=闹钟不设置不显示 1=直接显示所有闹钟 针对于有固定闹钟图标的设备,不开闹钟也会有相应的符号,可以选择 1	Bit6	0: 不支持添加及删除闹钟功能; 1: 支持添加删除闹钟功能;	Bit5	保留,为 0	Bit4	保留,为 0	Bit3-Bit0	闹钟数量,最多 15 个	Byte1	Bit7	模式 0:一次性,当天有效 0=不支持该项功能 1 =支持该项功能	Bit6	模式 1:每天都响	Bit5	模式 2:周一至周五	Bit4	模式 3:周一至周六	Bit3	模式 4:自定义,从周一到周日任意搭配	Bit2	保留,为 0		Bit1	保留,为 0
Byte0	Bit7	0=闹钟不设置不显示 1=直接显示所有闹钟 针对于有固定闹钟图标的设备,不开闹钟也会有相应的符号,可以选择 1																											
	Bit6	0: 不支持添加及删除闹钟功能; 1: 支持添加删除闹钟功能;																											
	Bit5	保留,为 0																											
	Bit4	保留,为 0																											
	Bit3-Bit0	闹钟数量,最多 15 个																											
Byte1	Bit7	模式 0:一次性,当天有效 0=不支持该项功能 1 =支持该项功能																											
	Bit6	模式 1:每天都响																											
	Bit5	模式 2:周一至周五																											
	Bit4	模式 3:周一至周六																											
	Bit3	模式 4:自定义,从周一到周日任意搭配																											
	Bit2	保留,为 0																											
	Bit1	保留,为 0																											

		<table border="1"> <tr> <td>Bit0</td> <td>保留,为 0</td> </tr> <tr> <td>Byte2</td> <td>保留,为 0</td> </tr> </table>	Bit0	保留,为 0	Byte2	保留,为 0					
Bit0	保留,为 0										
Byte2	保留,为 0										
0x17:恢复出厂设置声明	1	Data: 0x00:不支持恢复出厂设置指令 0x01:支持恢复出厂设置指令									
0x18:参数校准声明	N	TYPE 列表 举例: 如果甲醛和 PM2.5 支持参数校准 Type 列表有两个,即有两个 byte,内容是 0x01,0x04 说明: 1.APP 不做最小/最大值限制,APP 只下发加 1(最小精度) 或者减 1,由设备端返回当前实际校准值.									
0x19:时间格式声明	1	Feature1 (1byte): 时间格式切换功能 <table border="1"> <tr> <td rowspan="3">Byte0</td> <td>Bit7- Bit2</td> <td>保留,为 0</td> </tr> <tr> <td>Bit1</td> <td>支持 12h 格式</td> </tr> <tr> <td>Bit0</td> <td>支持 24h 格式,默认</td> </tr> </table>	Byte0	Bit7- Bit2	保留,为 0	Bit1	支持 12h 格式	Bit0	支持 24h 格式,默认		
Byte0	Bit7- Bit2	保留,为 0									
	Bit1	支持 12h 格式									
	Bit0	支持 24h 格式,默认									
0x1A:设备亮度声明	1	Feature1 (1byte): <table border="1"> <tr> <td rowspan="4">Byte0</td> <td>Bit7</td> <td>自动调节: 0=不支持 1=支持</td> </tr> <tr> <td>Bit6</td> <td>手动调节: 0=不支持 1=支持</td> </tr> <tr> <td>Bit5</td> <td>模式: 0:百分比模式:0-100% 1:档位模式</td> </tr> <tr> <td>Bit4- Bit0</td> <td>档位数量:(1-15) 档位模式下,档位的数量,例如 5,则亮度共 5 个档次. 若是百分比模式,则该值为 0</td> </tr> </table>	Byte0	Bit7	自动调节: 0=不支持 1=支持	Bit6	手动调节: 0=不支持 1=支持	Bit5	模式: 0:百分比模式:0-100% 1:档位模式	Bit4- Bit0	档位数量:(1-15) 档位模式下,档位的数量,例如 5,则亮度共 5 个档次. 若是百分比模式,则该值为 0
Byte0	Bit7	自动调节: 0=不支持 1=支持									
	Bit6	手动调节: 0=不支持 1=支持									
	Bit5	模式: 0:百分比模式:0-100% 1:档位模式									
	Bit4- Bit0	档位数量:(1-15) 档位模式下,档位的数量,例如 5,则亮度共 5 个档次. 若是百分比模式,则该值为 0									
0x1B:按键音效声明	1	Feature1 (1byte):									

		<table border="1"> <tr> <td>Byte0</td> <td>Bit7</td> <td>按键音效开/关切换支持 0=不支持 1=支持</td> </tr> <tr> <td></td> <td>Bit6- Bit0</td> <td>保留,为 0</td> </tr> </table>	Byte0	Bit7	按键音效开/关切换支持 0=不支持 1=支持		Bit6- Bit0	保留,为 0									
Byte0	Bit7	按键音效开/关切换支持 0=不支持 1=支持															
	Bit6- Bit0	保留,为 0															
0x1C:报警音效声明	1	Feature1 (1byte): <table border="1"> <tr> <td>Byte0</td> <td>Bit7</td> <td>报警音效开/关切换支持 0=不支持 1=支持</td> </tr> <tr> <td></td> <td>Bit6- Bit0</td> <td>保留,为 0</td> </tr> </table>	Byte0	Bit7	报警音效开/关切换支持 0=不支持 1=支持		Bit6- Bit0	保留,为 0									
Byte0	Bit7	报警音效开/关切换支持 0=不支持 1=支持															
	Bit6- Bit0	保留,为 0															
0x1D:图标显示声明	1	Feature1 (1byte): <table border="1"> <tr> <td>Byte0</td> <td>Bit7</td> <td>图标显示开/关切换支持 0=不支持 1=支持</td> </tr> <tr> <td></td> <td>Bit6- Bit0</td> <td>保留,为 0</td> </tr> </table>	Byte0	Bit7	图标显示开/关切换支持 0=不支持 1=支持		Bit6- Bit0	保留,为 0									
Byte0	Bit7	图标显示开/关切换支持 0=不支持 1=支持															
	Bit6- Bit0	保留,为 0															
0x1E:监控显示数据声明	1	Feature1 (1byte): <table border="1"> <tr> <td>Byte0</td> <td>Bit7</td> <td>监控显示开/关切换支持 0=不支持 1=支持</td> </tr> <tr> <td></td> <td>Bit6- Bit0</td> <td>保留,为 0</td> </tr> </table>	Byte0	Bit7	监控显示开/关切换支持 0=不支持 1=支持		Bit6- Bit0	保留,为 0									
Byte0	Bit7	监控显示开/关切换支持 0=不支持 1=支持															
	Bit6- Bit0	保留,为 0															
0x1F:数据显示模式声明	5	Feature1 (5byte): <table border="1"> <tr> <td>Byte0</td> <td></td> <td>VID 低位</td> </tr> <tr> <td>Byte1</td> <td></td> <td>VID 高位</td> </tr> <tr> <td>Byte2</td> <td></td> <td>PID 低位</td> </tr> <tr> <td>Byte3</td> <td></td> <td>PID 低位</td> </tr> <tr> <td>Byte4</td> <td></td> <td>模式数量</td> </tr> </table> <p>说明: 正常情况,APP 只会显示模式 1/2/3...切换 , 若想具体显示细节内容,可联系我司定制.</p>	Byte0		VID 低位	Byte1		VID 高位	Byte2		PID 低位	Byte3		PID 低位	Byte4		模式数量
Byte0		VID 低位															
Byte1		VID 高位															
Byte2		PID 低位															
Byte3		PID 低位															
Byte4		模式数量															

--	--	--

设备返回状态信息		
Type	Length(后面 Value 的 byte 数)	Value(小端序)
0x01:甲醛	2	当前值(2byte),小数点位数与前面的申明保持一致
0x02:温度	3	Feature(1byte): [Bit0-Bit1:小数点位数 Bit2: 当前值的正负号(0=正值,1=负值) Bit3: 当前单位 0=°C. 1=°F.] Data:当前温度值(2bytes)
0x03:湿度	2	Data:当前值(2byte),小数点位数与前面的申明保持一致
0x04:PM2.5	2	Data:当前值(2byte),小数点位数与前面的申明保持一致
0x05:PM1.0	2	Data:当前值(2byte),小数点位数与前面的申明保持一致
0x06:PM10	2	Data:当前值(2byte),小数点位数与前面的申明保持一致
0x07:VOC	2	Data:当前值(2byte),小数点位数与前面的申明保持一致
0x08:CO2	2	Data:当前值(2byte),小数点位数与前面的申明保持一致
0x09:AQI	2	Data:当前值(2byte),小数点位数与前面的申明保持一致
0x0A:报警状态	5	Feature(1byte): [Bit0:报警状态:0=报警功能未打开. 1=报警功能已打开.] Data:: (4byte -uint32) =0:没有报警. =1:正在进行的报警 Bit0: 甲醛过高 Bit1: 温度过低 Bit2: 温度过高 Bit3: 湿度过低 Bit4: 湿度过高

		Bit5: PM2.5 报警 Bit6: PM1.5 报警 Bit7: PM10 报警 Bit8: VOC 报警 Bit9: CO2 报警 Bit10: AQI 报警 Bit11: TVOC 报警																	
0x0B:音量状态	2	Feature(1byte): [Bit0:音量状态:0=音量已关闭. 1=音量已打开.] Data::音量值(1byte)																	
0x0C:报警时长	2	Data::报警时长值(2byte)																	
0x0D:报警铃声	1	Data::报警铃声值(1byte)																	
0x0E:设备故障	4	Err(4byte) =0:没有异常. =1:出现异常 Bit0:																	
0x0F:设备自检	1	Data:(1byte) 0x00:正常 0x01:正在自检																	
0x10:TVOC	2	当前值(2byte),小数点位数与前面的申明保持一致																	
0x12:电池状态	2	Feature(1byte): [Bit0:充电状态:0=未充电. 1=在充电. Bit1:电压状态: 0=电压正常. 1=低电] Data::电量百分比(1byte)																	
0x15:CO	2	Data:当前值(2byte),小数点位数与前面的申明保持一致																	
0x16:闹钟状态	N:根据闹钟的数量,该值不一样	闹钟 1: <table border="1" style="margin-left: 20px;"> <tr> <td rowspan="3">Byte0</td> <td>Bit7</td> <td>状态: 0=关闭, 1=启动</td> </tr> <tr> <td>Bit6- Bit4</td> <td>保留,为 0</td> </tr> <tr> <td>Bit3- Bit0</td> <td>闹钟编号,1-15</td> </tr> <tr> <td rowspan="2">Byte1</td> <td>Bit7- Bit4</td> <td>保留,为 0</td> </tr> <tr> <td>Bit3- Bit0</td> <td>模式:0-15 例如:周一至周五, 对应上面声明模式的模式 2,则该值为 2</td> </tr> <tr> <td rowspan="2">Byte2</td> <td>Bit7- Bit5</td> <td>保留,为 0</td> </tr> <tr> <td>Bit4-</td> <td>时:0-23</td> </tr> </table>	Byte0	Bit7	状态: 0=关闭, 1=启动	Bit6- Bit4	保留,为 0	Bit3- Bit0	闹钟编号,1-15	Byte1	Bit7- Bit4	保留,为 0	Bit3- Bit0	模式:0-15 例如:周一至周五, 对应上面声明模式的模式 2,则该值为 2	Byte2	Bit7- Bit5	保留,为 0	Bit4-	时:0-23
Byte0	Bit7	状态: 0=关闭, 1=启动																	
	Bit6- Bit4	保留,为 0																	
	Bit3- Bit0	闹钟编号,1-15																	
Byte1	Bit7- Bit4	保留,为 0																	
	Bit3- Bit0	模式:0-15 例如:周一至周五, 对应上面声明模式的模式 2,则该值为 2																	
Byte2	Bit7- Bit5	保留,为 0																	
	Bit4-	时:0-23																	

		<table border="1"> <tr> <td></td> <td>Bit0</td> <td></td> </tr> <tr> <td rowspan="2">Byte3</td> <td>Bit7- Bit6</td> <td>保留,为 0</td> </tr> <tr> <td>Bit5- Bit0</td> <td>分:0-59</td> </tr> <tr> <td rowspan="9">Byte4</td> <td>Bit7</td> <td>周日 0=该天闹钟不响 1=该天闹钟响</td> </tr> <tr> <td>Bit6</td> <td>周六</td> </tr> <tr> <td>Bit5</td> <td>周五</td> </tr> <tr> <td>Bit4</td> <td>周四</td> </tr> <tr> <td>Bit3</td> <td>周三</td> </tr> <tr> <td>Bit2</td> <td>周二</td> </tr> <tr> <td>Bit1</td> <td>周一</td> </tr> <tr> <td>Bit0</td> <td>保留,为 0</td> </tr> </table> <p>备注:模式优先级高,即先看模式,再判断时间.</p> <p>闹钟 2: 闹钟 2 格式和闹钟 1 一致,不做重复描述</p>		Bit0		Byte3	Bit7- Bit6	保留,为 0	Bit5- Bit0	分:0-59	Byte4	Bit7	周日 0=该天闹钟不响 1=该天闹钟响	Bit6	周六	Bit5	周五	Bit4	周四	Bit3	周三	Bit2	周二	Bit1	周一	Bit0	保留,为 0
	Bit0																										
Byte3	Bit7- Bit6	保留,为 0																									
	Bit5- Bit0	分:0-59																									
Byte4	Bit7	周日 0=该天闹钟不响 1=该天闹钟响																									
	Bit6	周六																									
	Bit5	周五																									
	Bit4	周四																									
	Bit3	周三																									
	Bit2	周二																									
	Bit1	周一																									
	Bit0	保留,为 0																									
	0x17:恢复出厂设置	1	<p>Data:</p> <p>0x00:正常工作状态</p> <p>0x01:正在恢复出厂设置</p> <p>0x02:恢复出厂设置成功</p>																								
0x18:参数校准状态	N	<p>参数 1:</p> <table border="1"> <tr> <td>Byte0</td> <td></td> <td>参数 type 例如,甲醛的是 0x01</td> </tr> <tr> <td rowspan="2">Byte1</td> <td>Bit7</td> <td>0=正值 1=负值</td> </tr> <tr> <td>Bit6- Bit0</td> <td>校准值的高位</td> </tr> <tr> <td>Byte2</td> <td>Bit7- Bit0</td> <td>校准值的低位</td> </tr> </table> <p>参数 2: 参数 2 格式和参数 1 一致,不做重复描述</p> <p>说明:当 APP 下发设置校准值时,设备端需要把校准值,和校准后的参数的值,一起上发.</p> <p>举例:如果甲醛当前值为 0.505(校准后的值),校准值为 -0.007. 当 APP 下发加 1 时, 则设备需要返回甲醛值当前值 0.506 ,返回校准值-0.006.</p>	Byte0		参数 type 例如,甲醛的是 0x01	Byte1	Bit7	0=正值 1=负值	Bit6- Bit0	校准值的高位	Byte2	Bit7- Bit0	校准值的低位														
Byte0		参数 type 例如,甲醛的是 0x01																									
Byte1	Bit7	0=正值 1=负值																									
	Bit6- Bit0	校准值的高位																									
Byte2	Bit7- Bit0	校准值的低位																									

		参数 1 的 byte0= 0x01 ,byte1=0x80 ,byte2=0x06						
0x19:时间格式状态	1	Feature1 (1byte): 0x01:当前为 24H 制 0x02:当前为 12H 制						
0x1A:设备亮度状态	1	Feature1 (1byte): <table border="1" style="margin-left: 20px;"> <tr> <td>Byte0</td> <td>Bit7</td> <td>自动调节 0=关闭 1=打开</td> </tr> <tr> <td></td> <td>Bit6- Bit0</td> <td>亮度值 0-100% /档位</td> </tr> </table>	Byte0	Bit7	自动调节 0=关闭 1=打开		Bit6- Bit0	亮度值 0-100% /档位
Byte0	Bit7	自动调节 0=关闭 1=打开						
	Bit6- Bit0	亮度值 0-100% /档位						
0x1B:按键音效状态	1	Feature1 (1byte): <table border="1" style="margin-left: 20px;"> <tr> <td>Byte0</td> <td>Bit7</td> <td>按键音效 0=关闭 1=打开</td> </tr> <tr> <td></td> <td>Bit6- Bit0</td> <td>保留,为 0</td> </tr> </table>	Byte0	Bit7	按键音效 0=关闭 1=打开		Bit6- Bit0	保留,为 0
Byte0	Bit7	按键音效 0=关闭 1=打开						
	Bit6- Bit0	保留,为 0						
0x1C:报警音效状态	1	Feature1 (1byte): <table border="1" style="margin-left: 20px;"> <tr> <td>Byte0</td> <td>Bit7</td> <td>报警音效 0=关闭 1=打开</td> </tr> <tr> <td></td> <td>Bit6- Bit0</td> <td>保留,为 0</td> </tr> </table>	Byte0	Bit7	报警音效 0=关闭 1=打开		Bit6- Bit0	保留,为 0
Byte0	Bit7	报警音效 0=关闭 1=打开						
	Bit6- Bit0	保留,为 0						
0x1D:图标显示状态	1	Feature1 (1byte): <table border="1" style="margin-left: 20px;"> <tr> <td>Byte0</td> <td>Bit7</td> <td>图标显示 0=关闭 1=打开</td> </tr> <tr> <td></td> <td>Bit6- Bit0</td> <td>保留,为 0</td> </tr> </table>	Byte0	Bit7	图标显示 0=关闭 1=打开		Bit6- Bit0	保留,为 0
Byte0	Bit7	图标显示 0=关闭 1=打开						
	Bit6- Bit0	保留,为 0						
0x1E:监控显示数据状态	1	Feature1 (1byte): <table border="1" style="margin-left: 20px;"> <tr> <td>Byte0</td> <td>Bit7</td> <td>监控显示 0=关闭 1=打开</td> </tr> <tr> <td></td> <td>Bit6- Bit0</td> <td>保留,为 0</td> </tr> </table>	Byte0	Bit7	监控显示 0=关闭 1=打开		Bit6- Bit0	保留,为 0
Byte0	Bit7	监控显示 0=关闭 1=打开						
	Bit6- Bit0	保留,为 0						

			Bit0
0x1F:数据显示模式	1	Feature1 (1byte):	Byte0 当前模式

APP 参数设置		
Type	Length(后面 Value 的 byte 数)	Value(小端序)
0x01:甲醛	3	Feature(1byte): [Bit0:0=关闭报警 . 1=打开报警] Data(2byte):报警值: (小数点位数与前面的申明保持一致)
0x02:温度	5	Feature(1byte): [Bit0-Bit1:小数点位数 Bit2: 下限值的正负号(0=正值,1=负值) Bit3: 上限值的正负号(0=正值,1=负值) Bit4: 当前单位 0=°C. 1=°F. Bit5: 0=关闭报警 . 1=打开报警] Min:报警下限值(2bytes)[..] Max:报警上限值(2bytes)[..]
0x03:湿度	5	Min:报警下限值(2bytes)[..] Max:报警上限值(2bytes)[..] Feature(1byte): [Bit0:0=关闭报警 . 1=打开报警]
0x04:PM2.5	3	Feature(1byte): [Bit0:0=关闭报警 . 1=打开报警] Data(2byte):报警值: (小数点位数与前面的申明保持一致)
0x05:PM1.0	3	Feature(1byte): [Bit0:0=关闭报警 . 1=打开报警] Data(2byte):报警值: (小数点位数与前面的申明保持一致)
0x06:PM10	3	Feature(1byte): [Bit0:0=关闭报警 . 1=打开报警] Data(2byte):报警值:

		(小数点位数与前面的申明保持一致)
0x07:VOC	3	Feature(1byte): [Bit0:0=关闭报警 . 1=打开报警] Data(2byte):报警值: (小数点位数与前面的申明保持一致)
0x08:CO2	3	Feature(1byte): [Bit0:0=关闭报警 . 1=打开报警] Data(2byte):报警值: (小数点位数与前面的申明保持一致)
0x09:AQI	3	Feature(1byte): [Bit0:0=关闭报警 . 1=打开报警] Data(2byte):报警值: (小数点位数与前面的申明保持一致)
0x0A	1	0x00: 所有报警关闭 0x01: 所有报警打开 对只支持一个总开关的情况使用, 支持控制各传感器数据报警的情况, 在各传感器自己的 type 类型 TLV 数据中设置报警开关,
0x0B:音量状态	2	Feature(1byte): [Bit0:音量状态:0=音量已关闭. 1=音量已打开.] Level:音量值(1byte)
0x0C:报警时长	2	Time:报警时长值(2byte)
0x0D:报警铃声	1	Enum: 报警铃声值(1byte)
0x0F:设备自检	1	State:(1byte) 0x00:正常 0x01:正在自检
0x10:TVOC	3	Feature(1byte): [Bit0:0=关闭报警 . 1=打开报警] Data(2byte):报警值: (小数点位数与前面的申明保持一致)
0x11:单位切换	1	温度单位切换(1byte) 0x00:°C 0x01:°F
0x13:设备绑定	1	Data(1byte):(APP 首次连接设备绑定时,会通过指令请求用户确认) 0x01:APP 发起,请求用户按键通过绑定 0x02:MCU 返回,MCU 等待用户按键 0x03:MCU 返回,用户已按按键 0x04:MCU 返回,用户超时(30s)没按按键 0x05:APP 发起,APP 取消绑定.
0x15:CO	3	Feature(1byte): [Bit0:0=关闭报警 . 1=打开报警] Data(2byte):报警值:

		(小数点位数与前面的申明保持一致)																																									
0x16:闹钟设置	N:	闹钟 1: <table border="1" style="margin-left: 20px;"> <tr> <td rowspan="4">Byte0</td> <td>Bit7</td> <td>状态: 0=关闭, 1=启动</td> </tr> <tr> <td>Bit6</td> <td>0=关闭/启动, 1=删除</td> </tr> <tr> <td>Bit5- Bit4</td> <td>保留,为 0</td> </tr> <tr> <td>Bit3- Bit0</td> <td>闹钟编号,1-15</td> </tr> <tr> <td rowspan="2">Byte1</td> <td>Bit7- Bit4</td> <td>保留,为 0</td> </tr> <tr> <td>Bit3- Bit0</td> <td>模式:0-15 例如:周一至周五, 对应上面声明模式的模式 2,则该值为 2</td> </tr> <tr> <td rowspan="2">Byte2</td> <td>Bit7- Bit5</td> <td>保留,为 0</td> </tr> <tr> <td>Bit4- Bit0</td> <td>时:0-23</td> </tr> <tr> <td rowspan="2">Byte3</td> <td>Bit7- Bit6</td> <td>保留,为 0</td> </tr> <tr> <td>Bit5- Bit0</td> <td>分:0-59</td> </tr> <tr> <td rowspan="8">Byte4</td> <td>Bit7</td> <td>周日 0=改天闹钟不响 1=改天闹钟响</td> </tr> <tr> <td>Bit6</td> <td>周六</td> </tr> <tr> <td>Bit5</td> <td>周五</td> </tr> <tr> <td>Bit4</td> <td>周四</td> </tr> <tr> <td>Bit3</td> <td>周三</td> </tr> <tr> <td>Bit2</td> <td>周二</td> </tr> <tr> <td>Bit1</td> <td>周一</td> </tr> <tr> <td>Bit0</td> <td>保留,为 0</td> </tr> </table> <p>备注:模式优先级高.即先看模式,再判断时间. APP 设置闹钟时,一般都是一个一个设置的.</p>	Byte0	Bit7	状态: 0=关闭, 1=启动	Bit6	0=关闭/启动, 1=删除	Bit5- Bit4	保留,为 0	Bit3- Bit0	闹钟编号,1-15	Byte1	Bit7- Bit4	保留,为 0	Bit3- Bit0	模式:0-15 例如:周一至周五, 对应上面声明模式的模式 2,则该值为 2	Byte2	Bit7- Bit5	保留,为 0	Bit4- Bit0	时:0-23	Byte3	Bit7- Bit6	保留,为 0	Bit5- Bit0	分:0-59	Byte4	Bit7	周日 0=改天闹钟不响 1=改天闹钟响	Bit6	周六	Bit5	周五	Bit4	周四	Bit3	周三	Bit2	周二	Bit1	周一	Bit0	保留,为 0
Byte0	Bit7	状态: 0=关闭, 1=启动																																									
	Bit6	0=关闭/启动, 1=删除																																									
	Bit5- Bit4	保留,为 0																																									
	Bit3- Bit0	闹钟编号,1-15																																									
Byte1	Bit7- Bit4	保留,为 0																																									
	Bit3- Bit0	模式:0-15 例如:周一至周五, 对应上面声明模式的模式 2,则该值为 2																																									
Byte2	Bit7- Bit5	保留,为 0																																									
	Bit4- Bit0	时:0-23																																									
Byte3	Bit7- Bit6	保留,为 0																																									
	Bit5- Bit0	分:0-59																																									
Byte4	Bit7	周日 0=改天闹钟不响 1=改天闹钟响																																									
	Bit6	周六																																									
	Bit5	周五																																									
	Bit4	周四																																									
	Bit3	周三																																									
	Bit2	周二																																									
	Bit1	周一																																									
	Bit0	保留,为 0																																									
0x17:恢复出厂设置	1	Data: 0x01:APP 设置开始恢复出厂设置																																									
0x18:参数校准设置		参数 1: <table border="1" style="margin-left: 20px;"> <tr> <td>Byte0</td> <td></td> <td>参数 type 例如,甲醛的是 0x01</td> </tr> <tr> <td>Byte1</td> <td>Bit7</td> <td>增加/减少 0=加 1</td> </tr> </table>	Byte0		参数 type 例如,甲醛的是 0x01	Byte1	Bit7	增加/减少 0=加 1																																			
Byte0		参数 type 例如,甲醛的是 0x01																																									
Byte1	Bit7	增加/减少 0=加 1																																									

		<table border="1"> <tr> <td></td> <td></td> <td>1=减 1</td> </tr> <tr> <td>Bit6</td> <td></td> <td>恢复默认值 0=加/减 1 1=恢复默认值(校准值为 0)</td> </tr> <tr> <td>Bit5- Bit0</td> <td></td> <td>保留</td> </tr> </table> <p>参数 2: 参数 2 格式和参数 1 一致,不做重复描述</p> <p>说明:当 APP 下发设置校准值时,设备端需要把校准值,和校准后的参数的值,一起上发.</p>			1=减 1	Bit6		恢复默认值 0=加/减 1 1=恢复默认值(校准值为 0)	Bit5- Bit0		保留
		1=减 1									
Bit6		恢复默认值 0=加/减 1 1=恢复默认值(校准值为 0)									
Bit5- Bit0		保留									
0x19:时间格式设置	1	Feature1 (1byte): 时间格式切换功能 0x01:切为 24H 制 0x02:切为 12H 制									
0x1A:设备亮度设置	1	Feature1 (1byte): <table border="1"> <tr> <td>Byte0</td> <td>Bit7</td> <td>自动调节 0=关闭 1=打开</td> </tr> <tr> <td></td> <td>Bit6- Bit0</td> <td>亮度值 0-100%/档位</td> </tr> </table>	Byte0	Bit7	自动调节 0=关闭 1=打开		Bit6- Bit0	亮度值 0-100%/档位			
Byte0	Bit7	自动调节 0=关闭 1=打开									
	Bit6- Bit0	亮度值 0-100%/档位									
0x1B:按键音效设置	1	Feature1 (1byte): <table border="1"> <tr> <td>Byte0</td> <td>Bit7</td> <td>按键音效 0=关闭 1=打开</td> </tr> <tr> <td></td> <td>Bit6- Bit0</td> <td>保留,为 0</td> </tr> </table>	Byte0	Bit7	按键音效 0=关闭 1=打开		Bit6- Bit0	保留,为 0			
Byte0	Bit7	按键音效 0=关闭 1=打开									
	Bit6- Bit0	保留,为 0									
0x1C:报警音效设置	1	Feature1 (1byte): <table border="1"> <tr> <td>Byte0</td> <td>Bit7</td> <td>报警音效 0=关闭 1=打开</td> </tr> <tr> <td></td> <td>Bit6- Bit0</td> <td>保留,为 0</td> </tr> </table>	Byte0	Bit7	报警音效 0=关闭 1=打开		Bit6- Bit0	保留,为 0			
Byte0	Bit7	报警音效 0=关闭 1=打开									
	Bit6- Bit0	保留,为 0									
0x1D:图标显示设置	1	Feature1 (1byte):									

		<table border="1"> <tr> <td>Byte0</td> <td>Bit7</td> <td>图标显示 0=关闭 1=打开</td> </tr> <tr> <td></td> <td>Bit6- Bit0</td> <td>保留,为 0</td> </tr> </table>	Byte0	Bit7	图标显示 0=关闭 1=打开		Bit6- Bit0	保留,为 0
Byte0	Bit7	图标显示 0=关闭 1=打开						
	Bit6- Bit0	保留,为 0						
0x1E:监控显示数据	1	Feature1 (1byte): <table border="1"> <tr> <td>Byte0</td> <td>Bit7</td> <td>监控显示 0=关闭 1=打开</td> </tr> <tr> <td></td> <td>Bit6- Bit0</td> <td>保留,为 0</td> </tr> </table>	Byte0	Bit7	监控显示 0=关闭 1=打开		Bit6- Bit0	保留,为 0
Byte0	Bit7	监控显示 0=关闭 1=打开						
	Bit6- Bit0	保留,为 0						
0x1F:数据显示模式设置	1	Feature1 (1byte): <table border="1"> <tr> <td>Byte0</td> <td></td> <td>模式</td> </tr> </table>	Byte0		模式			
Byte0		模式						

7.4.2 APP 获取设备功能列表

- APP 为了保证 APP 的功能和设备端的功能同步,APP 连接设备时,需要获取设备支持的功能.
- 若设备不支持功能,则相对应的功能和指令则不需支持及设计.

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0048	产品类型: 环境监测设备	
3		Payload 长度	
4	0x01	CMD: 获取设备支持的功能列表	Payload
5	0x00	保留	
6	SUM	校验和	
7	0x7A	包尾	

MCU 主控回应设备功能列表:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0048	产品类型: 环境监测设备	
3		Payload 长度	
4	0x02	CMD: 设备返回功能	Payload
		Group:总组数(一组指令只能发 200byte,当设备指令过长时,需要分包发送)	
	TYPE	具体的内容参考”功能说明”TLV ,只需填写支持的功能 TLV	
	Length		
	Value		
	TYPE	...	
	Length		
	Value		
	SUM	校验和	
	0x7A	包尾	

7.4.3 APP 获取设备状态

- APP 连接上设备后，需获取设备的状态,MCU 收到指令后需要返回.
- 当设备状态发生改变时,也需要通过该指令返回.
- APP 获取时,MCU 端需要全部支持的 TLV 返回. 当 MCU 主动上发时,只需要发改变了的 TLV 即可.
- 该指令的数据不会保存到服务器.

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0048	产品类型: 环境监测设备	
3		Payload 长度	
4	0x03	CMD: 获取设备状态	Payload
5	0x00	保留	
6	SUM	校验和	
7	0x7A	包尾	

MCU 主控回应设备状态:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0048	产品类型: 环境监测设备	
3		Payload 长度	
4	0x04	CMD: 设备返回功能	Payload
		Group:总组数(一组指令只能发 200byte,当设备指令过长时,需要分包发送)	
	TYPE	具体的内容参考”设备返回状态信息”TLV	
	Length		
	Value		
	TYPE		
	Length		
	Value		
	
	SUM	校验和	
	0x7A	包尾	

7.4.4 APP 设置/获取参数

- APP 在界面上进行功能获取/设置时,通过该指令设置
- MCU 收到获取指令时,需返回所有的参数,收到设置指令时,只需返回设置项的 TLV 内容即可

APP 设置:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0048	产品类型: 环境监测设备	
3		Payload 长度	
4	0x05	CMD: 设置/获取参数功能	Payload
		Op: 0x01:获取(当指令为获取时,指令里不需要包含 TLV,MCU 返回所有的参数即可) 0x02:设置	
		Group:总组数(一组指令只能发 200byte,当设备指令过长时,需要分包发送)	
	TYPE	具体的内容参考"APP 参数设置"TLV	
	Length		
	Value		
	TYPE		
	Length		
	Value		
	
	SUM	校验和	
	0x7A	包尾	

MCU 返回:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0048	产品类型: 环境监测设备	
3		Payload 长度	
4	0x06	CMD: MCU 返回参数功能	Payload
		Group:总组数(一组指令只能发 200byte,当设备指令过长时,需要分包发送)	
	TYPE	具体的内容参考"APP 参数设置"TLV	
	Length		
	Value		
	TYPE		
	Length		

	Value		
	
	SUM	校验和	
	0x7A	包尾	

7.4.5 MCU 上发保存数据

- 该指令的数据,会通过 HTTP 协议上发到服务器. 服务器会保存该数据当做设备的历史数据.
- 设备端定时 10 分钟发一组数据,若当前值与上一次上发值完全一致,则不需要上发数据.
- 当设备端数据出现报警状态时,则需立即上发数据,上发成功后,等 10 分钟后再上发数据即可(若出现其他报警,则可立即上发,不需再等 10 分钟).

APP 设置:

Byte	Default	Description
0	0xA7	包头
1~2	0x0048	产品类型: 环境监测设备
3		Payload 长度
4	0xF1	CMD: 上发保存数据
		Group:总组数(一组指令只能发 255byte ,当设备指令过长时,需要分包发送)
		Unix 时间(4bytes,小端序) 若设备端没有 Unix 时间,则该值为 0xFFFFFFFF
	TYPE	具体的内容参考”设备返回状态信息”TLV
	Length	
	Value	
	TYPE	
	Length	
	Value	

	SUM	校验和
	0x7A	包尾

MCU 返回:

Byte	Default	Description
0	0xA7	包头
1~2	0x0036	产品类型: 0x0036
3	0x02	Payload 长度
4	0xF1	Type: WM 回复数据上传结果
5		结果: 0: 上传成功 1: 上传失败
6	SUM (1~5)	(1~5)校验和
7	0x7A	包尾

7.4.6 APP 与模块心跳

- 该功能用于模块与 APP 通信,MCU 端不需处理.
- APP 打开的时候,mqtt 数据上发才有效,因此当 APP 打开后,需要定时 5s 发心跳包给模块.
- APP 可以根据心跳包判断设备是否在线.
- 当模块超时 4 次没收到心跳包,则停止 mqtt 数据上发.

APP 设置:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0048	产品类型: 环境监测设备	
3		Payload 长度	
4	0x07	CMD: 定时心跳包	Payload
		Uinx 时间(4byte,小端序)	
	SUM	校验和	
	0x7A	包尾	

MCU 返回:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0048	产品类型: 环境监测设备	
3		Payload 长度	
4	0x08	CMD: 模块返回心跳包	Payload
		Uinx 时间(4byte,小端序)	
	SUM	校验和	
	0x7A	包尾	

8 模块通用指令集

指令格式

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) 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

8.1 APP 同步时间到 MCU (CMD: 0x37、0x38)

➤ 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	CMD: 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	CMD: 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	CMD: MCU 请求 APP 下发时间	Payload
3		Value 0x01	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

8.2 设置模块立即进入休眠 CMD = 0x19

- 当模块进入睡眠后，模块的 Tx 输出高,Rx 作为输入唤醒口.
- MCU 的 Tx 输出高,Rx 作为输入口.
- 通过指令进入的休眠,模块都会有一定的电流,若想完全让模块停止工作.可通过电路断开模块的供电.(可参考硬件参考电路设计)

模块接收:

Byte	Value	Description	
0	0xA6	包头	
1	Len	Payload 长度	
2	0x19	CMD: 设置进入睡眠	Payload
3	0x01	Value: 0x01	
4		休眠模式: 0x02: 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)。 0x00: BLE 关闭, WIFI 关闭, 串口关闭, 系统内部 rc 时钟保持	
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	CMD: 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

8.3 设置模块自动休眠时间 CMD = 0x17

➤ 设置自动睡眠时间,当模块超时接收不到数据后,便可进入自动进入休眠模式.

模块接收:

Byte	Value	Description		
0	0xA6	包头		
1	0x08	Payload 长度		
2	0x17	CMD: 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 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, 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	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	CMD: 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

8.4 设置、读取 CID、VID、PID (CMD: 0x1D、0x1E)

- 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

MCU/APP 设置 ID:

Byte	Value	Description	
0	0xA6	包头	
1		Payload 长度	
2	0x1D	CMD: 设置 ID	Payload
3	0x07	设置 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 回复设置结果:

深圳市易连物联网有限公司

电话: (86) 0755-81773367 FAE 邮箱: hw@elinkthings.com 销售邮箱: marketing@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	Len	Payload 长度	
2	0x1D	CMD: 回复设置 ID 结果	Payload
3		结果值: 0: 成功 1: 失败 2: 不支持	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

MCU/APP 获取 ID:

Byte	Value	Description	
0	0xA6	包头	
1	0x01	Payload 长度	
2	0x1E	CMD: 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	CMD: 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	

8.5 设置模块重启 (CMD: 0x21)

设置重启模块:

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x21	CMD: 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	CMD: 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	

8.6 设置恢复出厂设置 (CMD: 0x22)

设置恢复出厂设置:

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x22	CMD: 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	CMD: 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	

8.7 获取 WM 模块状态(CMD = 0x26,0xAB)

- MCU/APP 可通过指令获取模块的状态
- 当模块状态变化时,会主动返回状态指令

获取模块状态:

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

WM 返回模块状态:

Byte	Value	Description	
0	0xA6	包头	
1	0x03	Payload 长度	
2	0x26	CMD: 返回模块状态	Payload
3		模块状态: bit0-bit3 表示 BLE 状态: 0: 无连接 1: 已连接 2: 配对完成 Bit4-bit7 表示 wifi 状态: 0: 未配置 AP; 1: 连接 AP 失败, 连接时密码错误、AP 信号不好、主动断开都会是这个状态; 2: 与服务器通讯失败; 3: 成功连接上 AP; 4: 正在连接 AP;	

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

WM 返回 WiFi 连接失败原因：

当 WM 连接 AP 失败时，WM 返回 WiFi 连接失败原因。其他状态不回应此条指令。

Byte	Value	Description	
0	0xA6	包头	
1	0x03	Payload 长度	
2	0xAB	CMD：返回 WiFi 连接失败原因	Payload
3		原因值： 0x00：未知原因 0x01：AP 信号差 0x02：密码错误 0x03：获取不到 IP	
4	0x00	保留位	
5	Sum (1~4)	校验和	
6	0x6A	包尾	

8.8 请求 Unix 时间 CMD = 0x44

➤ MCU 可指令获取 Unix 时间。

模块接收：

Byte	Value	Description	
0	0xA6	包头	
1	0x01	Payload 长度	
2	0x44	CMD：请求 Unix 时间	Payload
4	0x45	(1~2) 校验和	
5	0x6A	包尾	

模块响应：

按“设置 Unix 时间 CMD = 45”指令回复

8.9 设置 Unix 时间 CMD = 0x45

模块接收:

Byte	Value	Description
0	0xA6	包头
1	0x05	Payload 长度
2	0x45	CMD: 设置 Unix 时间
3~6		0x00000000: 未获取到准确值 (失败) 其他: Unix 时间, 4byte 小端序 (零时区)
7	Sum (1~6)	校验和
8	0x6A	包尾

模块响应:

Byte	Value	Description
0	0xA6	包头
1	0x02	Payload 长度
2	0x45	CMD: 回复设置 Unix 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

8.10 查询/使能 mqtt 连接状态 CMD = 0xC5

模块接收:

Byte	Value	Description
0	0xA6	包头
1	0x05	Payload 长度
2	0xC5	CMD: 查询 mqtt 连接状态
3		0x05
4		0x01
5		0x03:使能 MQTT 连接 AILINK 服务器 0x04:查询
6	Sum (1~5)	校验和
7	0x6A	包尾

模块响应:

Byte	Value	Description
0	0xA6	包头
1	0x05	Payload 长度
2	0xC5	CMD: 查询 mqtt 连接状态
3		0x05
4		0x01
5		0x00: 连接 0x01: 连接断开 0x02: 重新连接 在建立 MQTT 连接后, 如果模块和服务端出现断 连后, 模块会尝试重新连接服务端, 在连接状态 发生变化时, 模块端也会主动发送相应的状态给 MCU 端
6	Sum (1~5)	校验和
7	0x6A	包尾

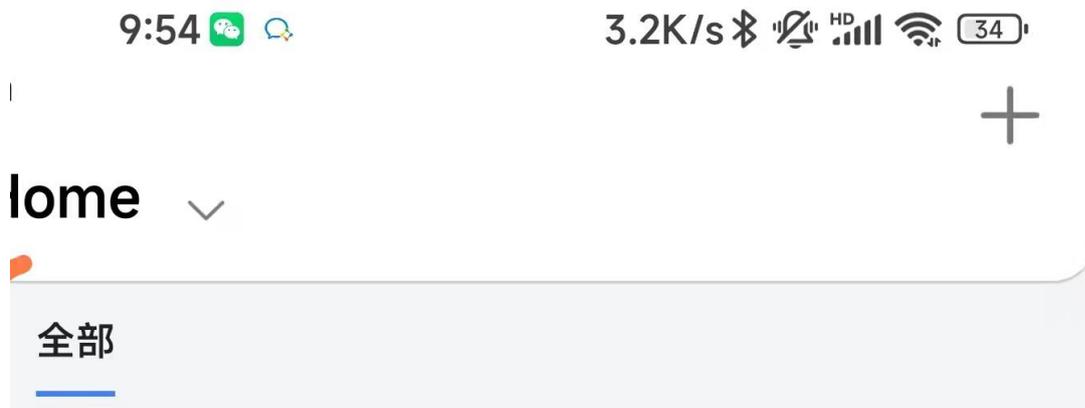
Payload

9 举例说明

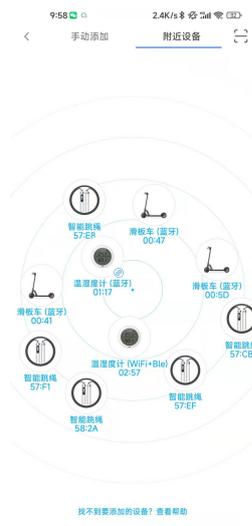
10 测试指导

10.1 连接测试

1. 给设备上电
2. 打开 Ailink APP
3. 点击右上角的”+”添加设备



4. 选择”手动添加”,找对应类型的设备添加.或者点击”附近设备”.



5. 选择设备进行绑定.

10.2 功能测试

具体的功能测试,请到[官网链接](#)下载产品的测试用例.测试完成后,通过对接窗口提交与我司审核.

11 生产测试指导

我们有生产使用的测试盒（BTS02），能够高效、快速、批量辅助生产测试。批量时，联系我司购买即可。



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

市场部邮箱：marketing@elinkthings.com

FAE 邮箱：hw@elinkthings.com

官网：www.elinkthings.com