

AiLink WM 系列温湿度计应用手册

版本：V2.0

更新日期：2026 年 1 月 6 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	撰写者	测试者	审核者	发布日期	修改说明
V1.0	LYX	/	lx1	2022/3/30	1. 初版
V1.1	LYX	/	lx1	2022/4/8	2. 更新温湿度校准指令格式 3. 更新闹钟指令格式 4. 更新 A7 指令里面的 type 5. 闹钟/夜灯等指令增加读取指令
V1.2	LYX	/	lx1	2022/4/11	2. 修改例行/报警上发数据的格式 (type 改为 0xF2) 3. 修改 wifi 返回状态指令, 增加 wifi 返回失败的原因
V1.3	LYX	/	lx1	2022/4/25	1. 修改硬件参考电路
V1.5	LYX	/	lx1	2023/1/11	1、
V1.7	YYT	/	LXL	2024/12/17	1、增加测试用例
V1.8	FMG	/	lyx	2025/12/18	1. 增加获取天气的功能 2. 按照 A6 指令汇总修改休眠指令 3. 修改 A6 指令汇总修改 F2 指令 4. 增加 MCU demo 下载链接
V1.9	FMG	/	lyx	2025/12/24	1. 增加获取天气指令可获取的天气元素
V2.0	Yyt	Ljl	Lyx	2025/1/6	1. 增加获取具体天气指令

目录

修改记录	2
目录	3
1 概述	5
2 说明	5
3 模块版本	5
4 硬件参考设计	6
4.1 串口 UART	6
4.2 硬件连接	6
5 蓝牙接口（默认）	7
5.1 蓝牙名称: AiLink_xxxx	7
6 流程及软件协议	7
6.1 产品定义	8
6.1.1 产品形态	8
6.1.2 说明:	8
6.2 产品工作流程	10
6.3 工作流程图	11
6.4 基础交互指令	12
6.4.1 APP 获取设备支持的功能	12
6.4.2 APP 获取设备状态	13
6.4.3 APP 设置变化阈值(灵敏度)	14
6.4.4 APP 设置、读取温湿度校准	15
6.4.5 APP 获取设备报警功能	16
6.4.6 APP 设置设备报警功能	17
6.4.7 APP 设置设备 wifi 例行上报时间	17
6.4.8 APP 设置/读取设备蜂鸣器功能	18
6.4.9 APP 设置/读取设备闹钟功能	19
6.4.10 APP 设置/读取设备整点报时功能	20
6.4.11 APP 设置/读取设备小夜灯	21
6.4.12 APP 设置/读取背光亮度	22
6.4.13 APP 设置/读取单位	22
6.4.14 例行/报警上发数据格式	23
6.4.15 其他功能指令	24
7 模块通用指令集	26
7.1 APP 同步时间到 MCU (Type: 0x37、0x38)	26
7.2 设置模块立即进入休眠 Type = 0x19	27
7.3 设置模块立即唤醒 Type = 0x1A	28
7.4 设置模块自动修眠时间 Type = 0x17	29
7.5 设置、读取 CID、VID、PID (Type: 0x1D、0x1E)	30

7.6 设置模块重启 (Type: 0x21)	31
7.7 设置恢复出厂设置 (Type: 0x22)	32
7.8 设置、获取 WM 模块状态(Type = 0x26,0xAB)	32
7.9 获取模块时间 Type = 0x1C	34
7.10 设置模块时间 Type = 0x1B	35
8 场景指令:获取本地天气	36
8.1 获取固定天气数据指令格式	36
8.1.1 数据指令	36
8.1.2 举例	37
8.1.3 天气附件	37
8.2 获取指定天气数据指令格式	38
8.2.1 数据指令	38
8.2.2 举例	39
8.2.3 天气指标对应关系表	39
9 MCU 开发指导	40
10 测试指导	41
10.1 模块测试	41
11 生产测试指导	42
12 联系我们	42

1 概述

- 1.1 本文档适用于深圳市易连物联网 WM 系列 BLE+WIFI 模块 接入 ailink APP。
- 1.2 本文档适用于温湿度计的 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、云平台帮助客户的温湿度计快速实现智能化，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 我们提供的蓝牙模块具有功耗低、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 WM 模块使用 APP+BLE 的配网方式,该方式配网速度快,配网稳定.
- 2.5 使用 WM 模块,通过 wifi 进行数据的传输,不需要靠近设备,也能通过手机远程查看设备数据.
- 2.6 传统温湿度计加上 WM 模块后,即可实现产品智能化升级,通过 APP 可以实现多种功能:温湿度的校准,温湿度报警值设置,远程监控温湿度报警,设置闹钟,设置定时报警,设置背光亮度,远程查看历史数据等等.

3 模块版本

本文档支持的固件版本：
WM10H1S2.0.0

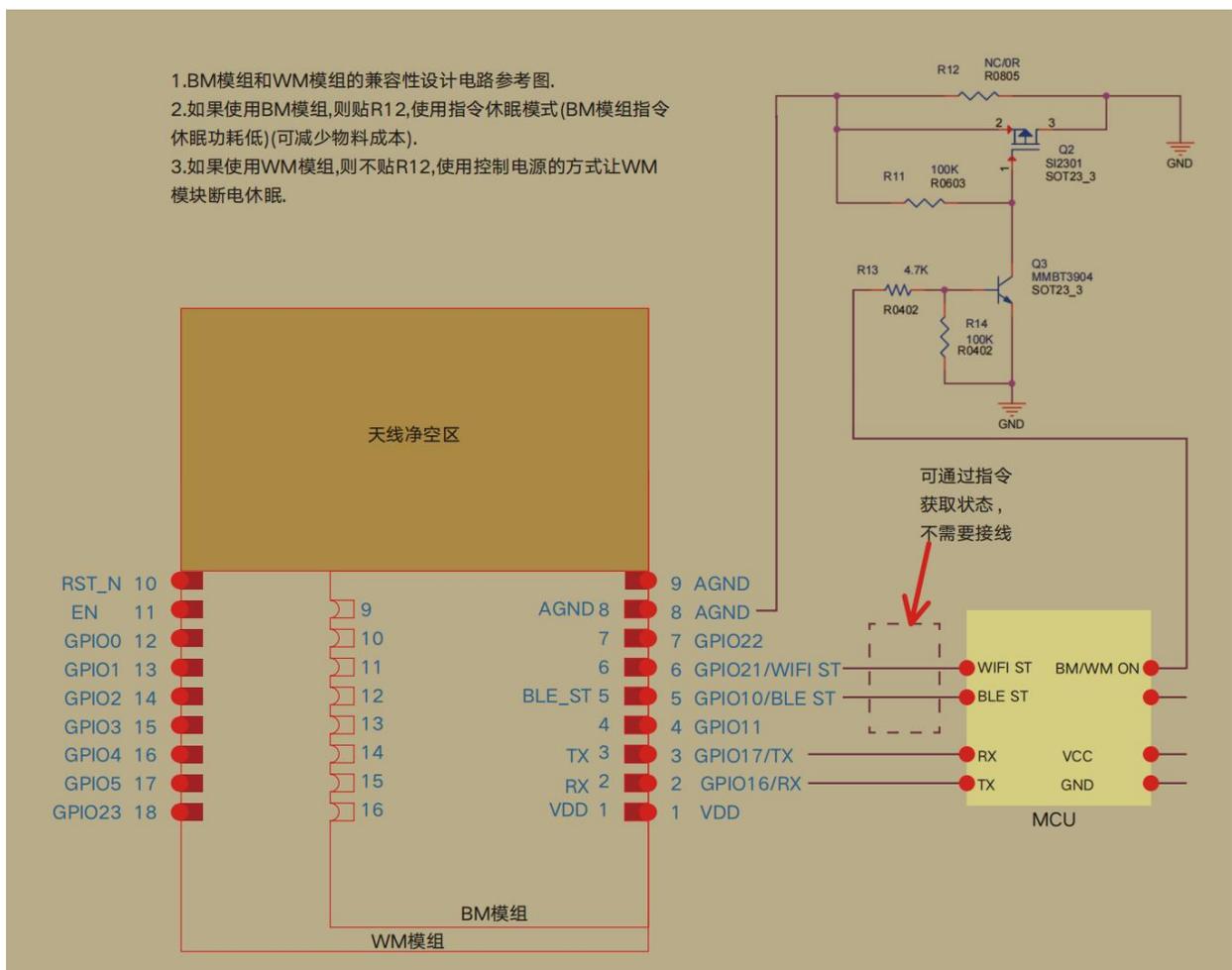
4 硬件参考设计

4.1 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。

4.2 硬件连接

4.2.1 参考电路：



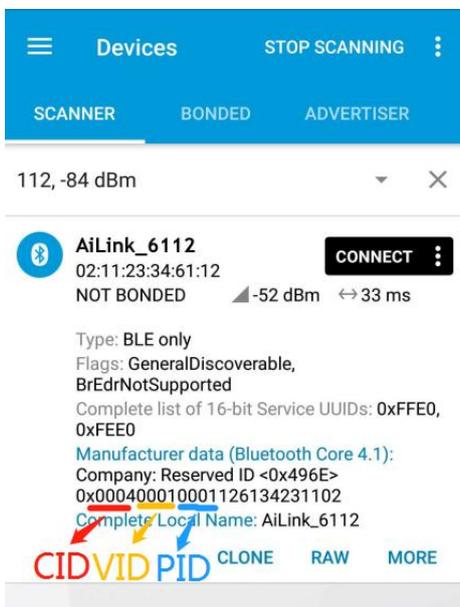
4.2.2 说明

4.2.2.1 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.

5 蓝牙接口（默认）

5.1 蓝牙名称：AiLink_xxxx

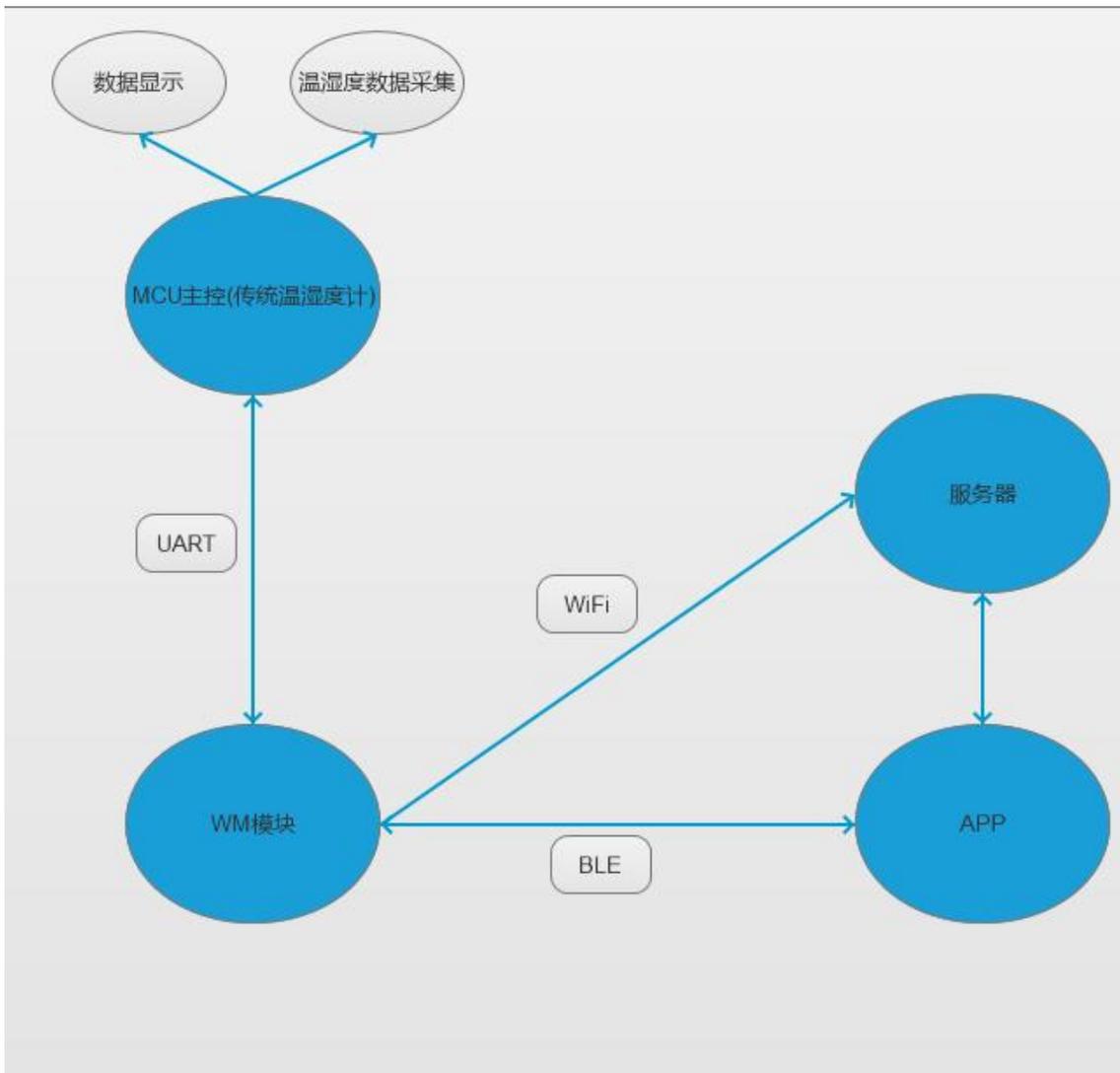
注：xxxx 为 Mac 地址后 4 个字符



6 流程及软件协议

6.1 产品定义

6.1.1 产品形态



6.1.2 说明:

1. 本产品包含三部分:
(1) MCU 主控 , WM 传输模块 , APP

- MCU 主控:
 - MCU 主控主要负责温湿度数据的采集,数据的显示,以及通过 UART 连接把数据传输到 WM 模块.
 - WM 传输模块:
 - WM 传输模块主要作为 APP 和 MCU 的沟通桥梁,同时把接收到的温湿度数据传输到 APP 和服务服务器上.
 - APP:
 - APP 通过与 WM 模块连接,与主机绑定,获取温湿度信息,设置功能等.
2. 数据保存规则建议
 - (1) 主控不需要实时保存数据,这样子会导致数据量巨大
 - (2) 可设置温湿度数据保存的灵敏度,当当前值与上一次值超过灵敏度时,才保存当前数据.
 - (3) 保存数据时,可以设置超过一定的时间才保存一次(例如 60s),低于该时间,就算符合条件 2,也不做保存.
 3. 唤醒 WM 模块传输到服务器的规则:
 - (1) 例行上发数据到服务器
 - ① 主控不能实时上发数据到服务器,否则功耗会很大.
 - ② 主控需设置例行上报的定时器(时间可通过 APP 修改),例行上报时间达到后,再把未上报的历史数据统一上报.
 - (2) 报警数据上发
 - ① 当温湿度值超过预设报警值时,MCU 需要立即触发例行上报功能.
 - ② 一段例行时间内只触发一次报警上发.(例如,例行上发时间是 10min,在第 6min 触发报警上发后,WM 会把 0-6min 的所有数据发给服务器,同时下一次上发时间变为第 16min,且 6-16min 内不再触发报警和例行上发).
 4. 设备与 APP 的绑定与连接
 - (1) 正常情况下,设备需要把 WM 模块断电
 - (2) 当设备需要和 APP 绑定和连接时,需要通过按键来触发给 WM 模块上电.
 5. Wifi 图标的建议
 - (1) Wm 模块不是处于长供电状态,因此 WM 模块的 wifi 会经常需要重新连接.
 - (2) 设备上需要显示 WM 断开连接,需要连续 2/3 个例行上报周期内,都连接不上网络,图标才显示无法连接网络.
 - (3) 当进入的是 120s 的 wifible 打开状态时,则需实时显示 wifi 状态(根据 wifi 状态指令)
 6. Unix 时间
 - (1) WM 模块获取到当前 Unix 时间时,会同步给设备,设备需要自行计算 Unix 时间(1s1s 往下加)
 - (2) 当设备给 WM 模块发数据上发到服务器时,需要带上 Unix 时间,否则 APP 无法计算产生温湿度值的时间.

6.2 产品工作流程

1. 设备首次上电
2. 给模块供电,并设置超时时间为 120s,120s 后可断电以节省功耗
3. WM 模块上电成功后,会主动返回状态.MCU 主控收到状态后,证明可以进行 UART 读写操作.
4. MCU 设置产品的 CID,VID,PID (必须设,否则 APP 无法按找到设备,CID=0x0036,VIDPID 在易连物联网后台配置).
ailnk CID VID PID 获取介绍: http://doc.elinkthings.com/web/#/40?page_id=144
5. APP 可连接设备进行网络及参数设置.
6. 120s 超时时间达到,把 WM 模块断电,节省功耗.
7. 当设备例行上报时间到达,给模块供电,并设置超时时间为 20s,20s 后数据已发送完成,可给模块断电节省功耗
8. 当 WM 模块返回连接服务器成功后,设备可把数据发给 WM 模块.
9. 当 APP 有修改参数时,WM 模块会从服务器获取,同时返回给主控.
10. 当数据发送完毕,给 WM 模块断电.或者超时 20s,给模块断电.
11. 当用户通过按键长按 5s 时,设备进入 WIFI/BLE 可连接设置模式

6.4 基础交互指令

- MCU 和模块通信时,每一帧的数据的间隔要不低于 100ms,默认设置 100ms.
- 当通过 BLE 连接 APP 时,可直接进行参数设置.当 BLE 不连接 APP 时,可通过 wifi 从服务器获取新参数进行设置(当参数有更新时才下发).

Byte	Value	Description
0	0xA7	包头
1	0x00	产品类型 (CID) 高字节
2		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。

6.4.1 APP 获取设备支持的功能

- 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	0x0036	产品类型: 温湿度计	
3		Payload 长度	
4	0x80	Type: 获取设备支持的功能	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	0x0036	产品类型: 温湿度计	
3		Payload 长度	
4	0x81	Type: 设备返回功能	Payload
5		如无特殊说明: 1:支持. 0:不支持	

		Bit0: 温湿度校准 Bit1: 蜂鸣器报警 Bit2: 闹钟功能 Bit3: 整点报时 Bit4:小夜灯 Bit5:背光亮度 Bit6: 0=接收 APP 下发指令, 1=不接收 APP 下发指令 Bit7: 0=支持温度单位切换, 1=不支持温度单位切换	
6		Bit0: 0=支持设备端温度报警, 1=不支持设备端温度报警 Bit1: 0=支持设备端湿度报警, 1=不支持设备端湿度报警 Bit2~7: 预留, 为 0	
7	0x00	预留	
8	0x00	预留	
9-17	0x00	预留	
18	SUM	校验和	
19	0x7A	包尾	

6.4.2 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.

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0036	产品类型: 温湿度计	
3		Payload 长度	
4	0x01	Type: 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	0x0036	产品类型: 温湿度计	
3		Payload 长度	
4	0x31	Type: 设备返回状态	Payload
5		电量 0% - 100%	
6-9		当前 unix 时间,单位:S (小端序,unix 时间为 0 时区时间戳)	

10-11		温度 (2byte, 小端序, 精度 0.1) Bit15: 正负号 (0: 正数, 1: 负数) Bit14: 单位(0:°C, 1:°F) Bit13-Bit0: 数据大小 例如 -10.0°C,则(byte10=0x64,byte11=0x80)
12-13		湿度 (2byte, 小端序, 单位%, 精度 0.1%)
14		状态: 0x00:APP 读取时上报
16	SUM	校验和
17	0x7A	包尾

6.4.3 APP 设置变化阈值(灵敏度)

- 当 MCU 检测的温湿度差值超过该阈值(灵敏度)时,需要通过指令”MCU 主控回应设备状态”返回设备状态.

默认阈值(灵敏度)为 温度: $\pm 0.3^{\circ}\text{C}$, 湿度: $\pm 2\%$

数据格式:

Byte	Default	Description
0	0xA7	包头
1~2	0x0036	产品类型: 温湿度计
3		Payload 长度
4	0x03	Type: 设置设备阈值(灵敏度)
5		温度阈值(灵敏度)(注:单位是°C) 1-100 (0.1-10.0°C)
6		湿度阈值(灵敏度) 1-100 (0.1-10.0%)
7	0x00	保留
8	SUM	校验和
9	0x7A	包尾

设备回应:

设备会定时返回这条状态指令

Byte	Default	Description
0	0xA7	包头
1~2	0x0036	产品类型: 温湿度计
3		Payload 长度
4	0x04	Type: 设备返回阈值(灵敏度)
5		温度阈值(灵敏度)(注:单位是°C) 1-100 (0.1-10.0°C)
6		湿度阈值(灵敏度)

		1-100 (0.1-10.0%)	
7	0x00	保留	
8	SUM	校验和	
9	0x7A	包尾	

6.4.4 APP 设置、读取温湿度校准

APP 可以设置温湿度的校准值，即通过 sensor 读数据出来后，再通过该值进行偏置。设置该值后，新的温湿度数据都会进行偏置。（默认值为 0）

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0036	产品类型：温湿度计	
3		Payload 长度	
4	0x0A	Type: 设置温湿度偏置	Payload
5		0x01: 设置(APP 下发数据时,两个单位的值都要算好,当设备处于哪个单位时,就使用哪个偏置的值,避免了单位转化导致精度丢失) 0x02: 读取	
6		温度偏置 (单位:°C .范围-10.0 到 +10.0°C , 默认值 0) Bit7:0: 正偏; 1: 负偏 Bit6-bit0: 数值, 一位小数点, 范围 (0-100: 0.0 - 10.0)	
7		温度偏置 (单位:°F .范围-10.0 到 +10.0 , 默认值 0) Bit7:0: 正偏; 1: 负偏 Bit6-bit0: 数值, 一位小数点, 范围 (0-100: 0.0 - 10.0)	
8		湿度偏置 (范围-10.0 到 +10.0% , 默认值为 0) Bit7:0: 正偏; 1: 负偏 Bit6-bit0: 数值, 一位小数点, 范围 (0-100: 0.0 - 10.0)	
9	0x00	保留位置	
10	SUM	校验和	
11	0x7A	包尾	

设备回应:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0036	产品类型：温湿度计	
3		Payload 长度	
4	0x0B	Type: 设置温湿度偏置	Payload

5	0x00		
6		温度偏置 (单位:℃ .范围-10.0 到 +10.0 , 默认值 0) Bit7:0: 正偏; 1: 负偏 Bit6-bit0: 数值, 一位小数点, 范围 (0-100: 0.0 - 10.0)	
7		温度偏置 (单位:F .范围-10.0 到 +10.0 , 默认值 0) Bit7:0: 正偏; 1: 负偏 Bit6-bit0: 数值, 一位小数点, 范围 (0-100: 0.0 - 10.0)	
8		湿度偏置 (范围-10.0 到 +10.0% , 默认值为 0) Bit7:0: 正偏; 1: 负偏 Bit6-bit0: 数值, 一位小数点, 范围 (0-100: 0.0 - 10.0)	
9		保留位置 0x00	
10	SUM	校验和	
11	0x7A	包尾	

6.4.5 APP 获取设备报警功能

数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0036	产品类型: 温湿度计	
3		Payload 长度	
4	0x12	Type: 获取设备报警温湿度	Payload
5	0x00	保留	
6	SUM	校验和	
7	0x7A	包尾	

设备回复数据格式:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0036	产品类型: 温湿度计	
3		Payload 长度	
4	0x12	Type: 获取设备报警温湿度	Payload
5		温度报警打开 0x00:未打开预警功能 0x01:打开预警功能	
6-7		高温报警值,大端序,单位:0.1℃	
8-9		低温报警值,大端序,单位:0.1℃	
10		湿度报警打开 0x00:未打开预警功能 0x01:打开预警功能	

11-12		高湿度报警值	
13-14		低湿度报警值	
15	SUM	校验和	
16	0x7A	包尾	

6.4.6 APP 设置设备报警功能

- APP 下发设置后,设备会以查询的返回指令返回.即上面的指令.

APP 下发设置:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0036	产品类型: 温湿度计	
3		Payload 长度	
4	0x13	Type: 设置设备报警温湿度	Payload
5		温度报警打开 0x00:未打开预警功能 0x01:打开预警功能	
6-7		高温度报警值,大端序,单位:0.1℃	
8-9		低温度报警值,大端序,单位:0.1℃	
10		湿度报警打开 0x00:未打开预警功能 0x01:打开预警功能	
11-12		高湿度报警值	
13-14		低湿度报警值	
15	SUM	校验和	
16	0x7A	包尾	

6.4.7 APP 设置设备 wifi 例行上报时间

- APP 下发例行上报时间,设备每隔该时间,需要把该时间段内产生所有 MCU 里历史记录,传输给 WM 模块,通过 WM 模块发到服务器.
- 间隔值默认 600s .

APP 下发设置:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0036	产品类型: 温湿度计	
3		Payload 长度	
4	0x14	Type: 设置设备例行上报时间	Payload
5-6		时间,单位:s,大端序	
7	SUM	校验和	

8	0x7A	包尾
---	------	----

设备返回设置:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0036	产品类型: 温湿度计	
3		Payload 长度	
4	0x15	Type: 设置返回例行上报时间	Payload
5-6		时间,单位:s,大端序 默认 600s	
7	SUM	校验和	
8	0x7A	包尾	

6.4.8 APP 设置/读取设备蜂鸣器功能

当设备有蜂鸣器时才需要该设置

APP 发送:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0036	产品类型: 温湿度计	
3		Payload 长度	
4	0x20	Type: 设置蜂鸣器	Payload
5		0x00:读取 0x01:设置	
6		0x00:打开(默认) 0x01:关闭	
7	SUM	校验和	
8	0x7A	包尾	

设备返回:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0036	产品类型: 温湿度计	
3		Payload 长度	
4	0x21	Type: 返回蜂鸣器状态	Payload
5		0x00:打开(默认) 0x01:关闭	
6	SUM	校验和	
7	0x7A	包尾	

6.4.9 APP 设置/读取设备闹钟功能

APP 发送:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0036	产品类型: 温湿度计	
3		Payload 长度	
4	0x22	Type: 设置闹钟	Payload
5		0x00:增加闹钟 0x01:删除闹钟(不需要处理时间值) 0x02:停止该闹钟(停止对应的编号闹钟,不需要处理时间值) 0x02:获取设备已存在的闹钟(编号模式时间都设为 0)	
6		编号: 1-5(对应 5 的个闹钟编号)	
7		模式: 0x00: 只触发 1 次 0x01: 每天都触发 1 次	
8-10		时间: 3 个 bytes 时(byte8) 分(byte9) 秒(byte10)	
11		状态: 0x00:已停止/已关闭 0x01:已激活	
12-17		预留位: 0x00	
18	SUM	校验和	
19	0x7A	包尾	

设备返回:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0036	产品类型: 温湿度计	
3		Payload 长度	
4	0x23	Type: 返回闹钟	Payload
5		0x00:增加闹钟 0x01:删除闹钟(不需要处理时间值) 0x02:停止该闹钟(停止对应的编号闹钟,不需要处理时间值) 0x02:获取设备已存在的闹钟(编号模式时间都设为 0)(陆续返回设备里面的所有闹钟,若设备没有闹钟,则返回 编号/模式/时间)	

		都为 0)	
6		编号: 1-5	
7		模式: 0x00: 只触发 1 次 0x01: 每天都触发 1 次	
8-10		时间: 3 个 byte 时 分 秒	
11		状态: 0x00:已停止/已关闭 0x01:已激活	
12-17		预留位: 0x00	
18	SUM	校验和	
19	0x7A	包尾	

6.4.10 APP 设置/读取设备整点报时功能

APP 发送:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0036	产品类型: 温湿度计	
3		Payload 长度	
4	0x24	Type: 设置整点报时	Payload
5		0x00:读取 0x01:设置	
6		0x00:关闭整点报时(默认) 0x01:打开整点报时	
7-17		预留位: 0x00	
18	SUM	校验和	
19	0x7A	包尾	

设备返回:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0036	产品类型: 温湿度计	
3		Payload 长度	
4	0x25	Type: 返回整点报时设置	Payload
5	0x00	返回状态	

6		0x00:关闭(默认) 0x01:打开	
7-17		预留位: 0x00	
18	SUM	校验和	
19	0x7A	包尾	

6.4.11 APP 设置/读取设备小夜灯

APP 发送:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0036	产品类型: 温湿度计	
3		Payload 长度	
4	0x26	Type: 设置小夜灯	Payload
5		0x00:读取 0x01:设置	
6		0x00:关闭小夜灯(默认) 0x01:打开小夜灯	
7-17		预留位: 0x00	
18	SUM	校验和	
19	0x7A	包尾	

设备返回:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0036	产品类型: 温湿度计	
3		Payload 长度	
4	0x27	Type: 返回小夜灯	Payload
5	0x00	返回状态	
6		0x00:关闭小夜灯(默认) 0x01:打开小夜灯	
7-17		预留位: 0x00	
18	SUM	校验和	
19	0x7A	包尾	

6.4.12 APP 设置/读取背光亮度

APP 发送:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0036	产品类型: 温湿度计	
3		Payload 长度	
4	0x28	Type: 设置背光亮度	Payload
5		0x00:读取 0x01:设置	
6		0x00:关闭(默认) 0x01:打开	
7		亮度值 0-100	
8-17		预留位: 0x00	
18	SUM	校验和	
19	0x7A	包尾	

设备返回:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0036	产品类型: 温湿度计	
3		Payload 长度	
4	0x29	Type: 返回背光亮度	Payload
5	0x00	返回状态	
6		0x00:关闭(默认) 0x01:打开	
7		亮度值 0-100	
8-17		预留位: 0x00	
18	SUM	校验和	
19	0x7A	包尾	

6.4.13 APP 设置/读取单位

APP 发送:

Byte	Default	Description
0	0xA7	包头
1~2	0x0036	产品类型: 温湿度计

3		Payload 长度	Payload
4	0x2A	Type: 设置单位	
5		0x00:获取 0x01:设置	
6		0x00:°C 0x01:°F	
7-17		预留位: 0x00	
18	SUM	校验和	
19	0x7A	包尾	

设备返回:

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0036	产品类型: 温湿度计	
3		Payload 长度	
4	0x2B	Type: 返回单位	Payload
5		0x00:返回单位	
6		0x00:°C 0x01:°F	
7-17		预留位: 0x00	
18	SUM	校验和	
19	0x7A	包尾	

6.4.14 例行/报警上发数据格式

- 当设备例行上报时间达到,或者报警触发时,通过该指令把未上发的数据统一上发.
- 数据量较大时,可分组上发.(等上一组上发完毕后再上发下一组)

Byte	Default	Description	
0	0xA7	包头	
1~2	0x0036	产品类型: 温湿度计	
3		Payload 长度	
4	0xF2	Type: 上发历史数据 (透传标记位)	Payload
5	0x01	T=0x01(历史)	
6		L=(长度 bytes,历史数据的长度,例如历史数据有 2 组,则 byte 的长度为 byte7-byte30,共 24bytes, 则 L=24)	
7-10		历史数据总数量 (4bytes, 小端序)	
11-14		已发数量 (4bytes, 小端序) APP 需要根据总数量和已发数量判断是否有数据, 数据是否发	

		送完成。	
15-18		Uinx 时间 1 (4bytes , 小端序)	
19-20		温度 1 (2byte, 小端序, 单位℃, 精度 0.1℃) Bit15: 正负号 (0: 正数, 1: 负数) Bit14:0=摄氏度 ,1=华氏度 Bit13-Bit0: 数据大小 例如(-10.1℃,则 byte19=0x65, byte20=0x80)	
21-22		湿度 1 (2byte, 小端序, 单位%, 精度 0.1%)	
23-26		Uinx 时间 2 (4bytes , 小端序)	
27-28		温度 2 (2byte, 小端序, 单位℃, 精度 0.1℃)	
29-30		湿度 2 (2byte, 小端序, 单位%, 精度 0.1%)	
	 (每次传数据, 最长是 126 组, 但不是每次都是 126 组)	
		Uinx 时间 126	
		温度 126	
		温度 126	
	0x02	T=0x02(电量)	
	0x01	L=1	
N-2		电量值 1-100%	
N-1	SUM	校验和	
N	0x7A	包尾	

WM 回复设置结果:

Byte	Default	Description	
0	0xA7	包头	
1~2		产品类型 CID:	
3	0x02	Payload 长度	
4	0xF2	Type: 回复上发结果	Payload
5		0x00 : 固定值	
6		0x01 : 固定值	
7		结果值: 0: 成功 1: 失败 2: 不支持 4: 正在上传数据	
8	Sum	(1~5) 校验和	
9	0x7A	包尾	

6.4.15 其他功能指令

- 由于产品形态多,功能类型大,若需要 APP 支持的指令,可通过后续讨论增加

7 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

7.1 APP 同步时间到 MCU (Type: 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	Type: APP 同步时间
3~9		时间: 7 个 byte 年 (当前年份-2000) 月 日 时 分 秒 星期 (1~7 1=周一 ~ 7=周日)
10	SUM (1~n)	(1~n)校验和
11	0x6A	包尾

● MCU 返回同步时间结果

Byte	Value	Description
0	0xA6	包头
1	0x02	Payload 长度
2	0x37	Type: MCU 返回时间同步结果
3		结果值: 0: 成功

		1: 失败 2: 不支持	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

● MCU 请求时间

设备有时间功能，且在与 APP 连接状态时，可以请求时间更新，APP 收到该请求，会下发时间同步。

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x38	Type: MCU 请求 APP 下发时间	Payload
3		Value 0x01	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

7.2 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

7.3 设置模块立即唤醒 Type = 0x1A

模块接收:

Byte	Value	Description	
0	0xA6	包头	
1	Len	Payload 长度	
2	0x1A	Type: 设置模块唤醒	Payload
3	0x01	Value: 0x01	
4	Sum (1~3)	校验和	
5	0x6A	包尾	

模块响应:

Byte	Value	Description	
0	0xA6	包头	
1	Len	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	

7.4 设置模块自动休眠时间 Type = 0x17

模块接收:

Byte	Value	Description		
0	0xA6	包头		
1	0x08	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 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	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

7.5 设置、读取 CID、VID、PID (Type: 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

设置 ID:

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

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: 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	

7.6 设置模块重启 (Type: 0x21)

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	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	

7.7 设置恢复出厂设置 (Type: 0x22)

设置恢复出厂设置:

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x22	Type: 设置恢复出厂设置	Payload
3	0x01	Value: 0x01	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

BM 回复设置结果:

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x22	Type: 回复设置模块重启结果	Payload
3		结果值: 0: 成功 (成功后, 100ms 后恢复出厂设置) 1: 失败 2: 不支持	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

7.8 设置、获取 WM 模块状态 (Type = 0x26, 0xAB)

获取模块状态:

Byte	Value	Description	
0	0xA6	包头	
1		Payload 长度	
2	0x26	Type: 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	Type: 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	Type: 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	

7.9 获取模块时间 Type = 0x1C

设备需要模块时间来计算产生温湿度的时间. 模块时间连接上 app 后, 需要 app 主动发送时间 (Type=0x1B) 给模块才有效

模块接收:

Byte	Value	Description	
0	0xA6	包头	
1	0x01	Payload 长度	
2	0x1C	Type: 请求 Unix 时间	Payload
4	0x1D	(1~2) 校验和	
5	0x6A	包尾	

模块响应:

Byte	Value	Description	
0	0xA6	包头	
1	0x08	Payload 长度	
2	0x1C	Type: 返回系统当前时间	Payload
3		系统时间有效位 0: 系统时间无效 1: 系统时间有效	
4		年: 年份=年+2000	
5		月 (1~12)	
6		日 (1~31)	
7		时 (0~23)	
8		分 (0~59)	
9		秒 (0~59)	
10	Sum (1~6)	校验和	
11	0x6A	包尾	

7.10 设置模块时间 Type = 0x1B

➤ 说明:

- 该时间为模块的系统时间。仅支持 APP 设置

模块接收:

Byte	Value	Description	
0	0xA6	包头	
1	0x08	Payload 长度	
2	0x1B	Type: 设置系统当前时间	Payload
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 (1~6)	校验和	
11	0x6A	包尾	

模块响应:

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x1B	Type: 回复设置 Unix 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	

8 场景指令:获取本地天气

- 功能:
- 通过指令获取本地的实时天气/未来天气

注意: 为了确保服务器的稳定性, 同时避免产生无效申请, 同一外网 IP 不能短时间反复请求 (上限 1 分钟 10 次), 超出限制后该 IP 需要等待 5 分钟后才能再次请求到天气。

8.1 获取固定天气数据指令格式

8.1.1 数据指令

CMD	Response						
获取所有天气指令: eAT+weatherMore=<type> type 值请查看附件	+weatherMore:OK (指令接收成功) or +weatherMore:NG,<reason> (失败) <table border="1" style="margin-top: 10px;"> <thead> <tr> <th>reason</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>指令无效</td> </tr> <tr> <td>2</td> <td>状态重复</td> </tr> </tbody> </table>	reason	描述	0	指令无效	2	状态重复
reason	描述						
0	指令无效						
2	状态重复						

CMD	Response										
获取当前状态: eAT+weatherMore?	+weatherMore=<state>,[param1] <table border="1" style="margin-top: 10px;"> <thead> <tr> <th>state</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>未进行获取</td> </tr> <tr> <td>1</td> <td>正在获取</td> </tr> <tr> <td>2</td> <td>获取成功,返回 param1 天气内容</td> </tr> <tr> <td>3</td> <td>获取失败</td> </tr> </tbody> </table> <p>param1:获取到的天气参数,获取成功时该值有效.具体内容请查看附件</p>	state	描述	0	未进行获取	1	正在获取	2	获取成功,返回 param1 天气内容	3	获取失败
state	描述										
0	未进行获取										
1	正在获取										
2	获取成功,返回 param1 天气内容										
3	获取失败										

8.1.2 举例

举例 1. 获取实况天气

eAT+weatherMore=0

+weatherMore:OK

+weatherMore=2,20250510154553,306,27,73;

举例 2. 获取未来 3 天预报天气

eAT+weatherMore=3

+weatherMore:OK

+weatherMore=2,20250510154700,20250510,307,307,19,29,87;20250510154700,20250511,101,151,19,25,63;20250510154700,20250512,101,151,21,29,75;

8.1.3 天气附件

附件为“eAT+weatherMore=<type>”指令能够获取到的固定天气类型。

type	返回数据解析
实况天气：0	示例数据：20251224152840,101,25,58,24,135,3,19,0.0,997,30,10,15;
	第 1 个元素：当地时间，格式：yyyyMMddHHmmss
	第 2 个元素：天气状况代码，当前天气状况和图标的代码，表示的天气状况描述列表请查看表格后的天气状况和图标连接；
	第 3 个元素：实况温度，默认单位：摄氏度；
	第 4 个元素：实况相对湿度，百分比数值
预报天气：1/3/7/10/15	多天的天气之间使用英文分号分隔
	示例数据： 2,20250510154700,20250510,307,307,19,29,87;20250510154700,20250511,101,151,19,25,63;20250510154700,20250512,101,151,21,29,75;
	第 1 个元素：当地时间，格式：yyyyMMddHHmmss；
	第 2 个元素：预报日期，格式：yyyyMMdd
	第 3 个元素：白天天气状况代码，当天天气状况和图标的代码，表示的天气状况描述列表请查看表格后的天气状况和图标连接
	第 4 个元素：夜晚天气状况代码，当天天气状况和图标的代码，表示的天气状况描述列表请查看表格后的天气状况和图标连接
	第 5 个元素：预报当天最低温度，单位：摄氏度
第 6 个元素：预报当天最高温度，单位：摄氏度	

第 7 个元素：相对湿度，百分比数值

- 天气状况和图标连接：<https://dev.qweather.com/docs/resource/icons/#icon-code>

8.2 获取指定天气数据指令格式

8.2.1 数据指令

CMD	Response										
获取指定天气指令： eAT+weatherMoreC={<"weatherType":"string">,<"infoArray":"string">,<"responseDataFormat":int>}	+weatherMoreC=stat,{val} <table border="1" style="margin-top: 10px;"> <thead> <tr> <th>stat</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>未进行获取</td> </tr> <tr> <td>1</td> <td>正在获取</td> </tr> <tr> <td>2</td> <td>获取成功,返回 val 天气内容</td> </tr> <tr> <td>3</td> <td>获取失败</td> </tr> </tbody> </table> <p>{val}:获取到的天气参数,获取成功时该值有效.具体内容请查看关系表</p>	stat	描述	0	未进行获取	1	正在获取	2	获取成功,返回 val 天气内容	3	获取失败
stat	描述										
0	未进行获取										
1	正在获取										
2	获取成功,返回 val 天气内容										
3	获取失败										

参数说明	weatherType:天气类型,可取值请查看下文“ 天气指标对应关系表 ”
	infoArray:需要的天气指标;可取值请查看下文“ 天气指标对应关系表 ”
	responseDataFormat:返回数据格式;1:有序 value 格式

CMD	Response										
获取当前状态： eAT+weatherMoreC?	+weatherMoreC:OK (指令接收成功) +weatherMoreC=stat,{val} <table border="1" style="margin-top: 10px;"> <thead> <tr> <th>stat</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>未进行获取</td> </tr> <tr> <td>1</td> <td>正在获取</td> </tr> <tr> <td>2</td> <td>获取成功,返回 val 天气内容</td> </tr> <tr> <td>3</td> <td>获取失败</td> </tr> </tbody> </table> <p>{val}:获取到的天气参数,获取成功时该值有效.具体内容请查看关系表</p>	stat	描述	0	未进行获取	1	正在获取	2	获取成功,返回 val 天气内容	3	获取失败
stat	描述										
0	未进行获取										
1	正在获取										
2	获取成功,返回 val 天气内容										
3	获取失败										

8.2.2 举例

举例 1. 获取实况温度数据和实况降水量

```
eAT+weatherMoreC={"weatherType":"0","infoArray":"3,9","responseDataFormat":1}
```

```
+weatherMoreC=2,{
  "status": "1",
  "data": "22,0.0"
}
```

举例 2. 获取未来 3 天的最低温和最高温

```
eAT+weatherMoreC={"weatherType":"3","infoArray":"5,6","responseDataFormat":1}
```

```
+weatherMoreC=2,{
  "status": "1",
  "data": "9,20;10,24;9,19"
}
```

8.2.3 天气指标对应关系表

weatherType	infoArray
实况天气:0	1: 当地时间, 格式: yyyyMMddHHmmss
	2: 天气图标代码: 当前天气状况和图标的代码, 表示的天气状况描述列表请查看本文末的“Remark->天气状况和图标”, 如: 101
	3: 实况温度, 默认单位: 摄氏度; 如: -10.1;
	4: 实况相对湿度, 百分比数值
	5: 实况体感温度, 默认单位: 摄氏度
	6: 实况风向 360 角度, 风向可以根据风角度换算, 详情查看“Remark->风向风速和等级”
	7: 风力等级, 如: 1-3 或者 1
	8: 风速, 单位: 公里/小时
	9: 实况降水量, 默认单位: 毫米
	10: 实况大气压强, 默认单位: 百帕
	11: 实况能见度, 默认单位: 公里
	12: 实况云量, 百分比数值, 可能为空
	13: 实况露点温度, 默认单位: 摄氏度, 可能为空
	多天的天气之间使用英文分号分隔
	1: 预报地区的本地时间, 如: 20260106115553;
	2: 预报日期, 格式: yyyyMMdd, 如: 20260106
	3: 预报白天天气状况的图标代码, 表示的天气状况描述列表请查看本文末的“Remark->天气状况和图标”

预报天气： 1/3/7/10/15	4: 预报夜间天气状况的图标代码，表示的天气状况描述列表请查看本文末的“Remark->天气状况和图标”
	5: 预报当天最低温度，单位：摄氏度
	6: 预报当天最高温度，单位：摄氏度
	7: 预报当天相对湿度，百分比数值
	8: 日出时间
	9: 日落时间
	10: 月升时间
	11: 月落时间
	12: 预报白天风向 360 角度
	13: 预报白天风力等级,如：1-3
	14: 预报白天风速，公里/小时
	15: 预报夜间风向 360 角度
	16: 预报夜间风力等级,如：1-3
	17: 预报夜间风速，公里/小时
	18: 预报当天降水量，默认单位：毫米
	19: 预报当天大气压强，默认单位：百帕
	20: 预报能见度，默认单位：公里
	21: 预报云量，百分比数值，可能为空
	22: 预报当天紫外线强度指数

- 天气状况和图标连接: <https://dev.qweather.com/docs/resource/icons/#icon-code>
- 风向风速和等级: <https://dev.qweather.com/docs/resource/wind-info/>

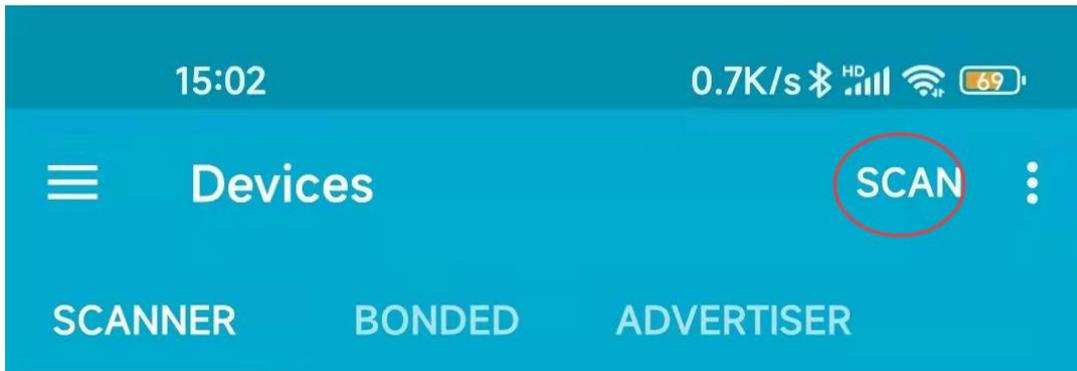
9 MCU 开发指导

1. 易连针对使用该文档进行 MCU 开发提供了开发 demo(参考代码)。下载链接：
http://doc.elinkthings.com/web/#/40?page_id=406

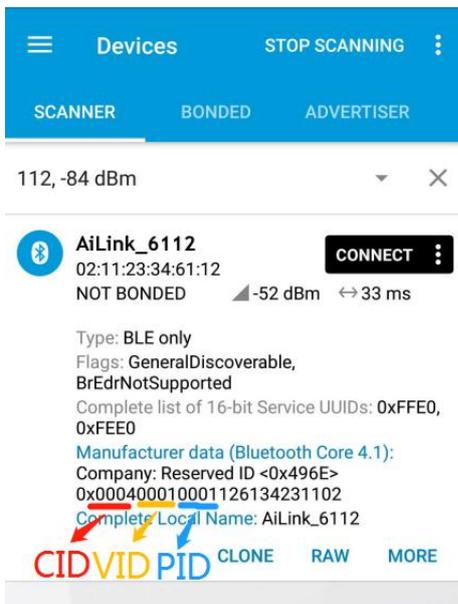
10 测试指导

10.1 模块测试

2. 模块上电
3. 下载手机通用测试 APP “nrf connect”
4. 安装 APP,打开 APP
5. 点击 scan



6. 找到名为 ailink 的模块 ,则该模块上电正常



11 生产测试指导

我们有生产使用的测试工具，能够高效、快速、批量辅助生产测试。批量时，联系我司。

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