

T1L 蓝牙充电宝应用手册

版本：V1.1

更新日期：2025 年 12 月 24 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	撰写者	测试者	审核者	发布日期	修改说明
V1.0	Lz	/	Yyt	2025/12/15	初版
V1.1	Lz	/	Yyt	2025/12/24	1、增加获取单节电池电压数据指令；重新说明 CID，VID，PID 的作用 2、修改说明。

目录

修改记录	- 2 -
目录	- 3 -
1 概述	- 4 -
2 说明	- 4 -
3 蓝牙芯片版本	- 4 -
4 硬件参考设计	- 5 -
4.1 串口 UART	- 5 -
4.2 硬件参考设计	- 5 -
5 蓝牙接口（默认）	- 5 -
5.1 蓝牙名称：Ailink_xxxx	- 5 -
5.2 UUID 说明	- 6 -
5.3 蓝牙连接服务列表 1： FFE0 举例	- 6 -
5.4 广播数据	- 6 -
5.4.1 第一类广播设置	- 6 -
6 流程及软件协议	- 8 -
6.1 基础交互流程	- 8 -
6.2 充电宝指令集	- 8 -
6.2.1 OP 列表	- 9 -
6.2.2 TLV 列表	- 10 -
7 蓝牙芯片通用指令集	- 15 -
7.1 获取蓝牙芯片状态（Type： 25、 26）	- 15 -
7.2 设置、读取 CID、 VID、 PID（Type： 1D、 1E）	- 16 -
7.3 APP 同步时间到 MCU（Type： 37、 38）	- 17 -
7.4 读取蓝牙芯片版本号（Type： 46）	- 18 -
7.5 设置芯片唤醒（Type： 1A）	- 19 -
7.6 设置芯片进入睡眠（Type： 19）	- 19 -
7.7 设置、读取芯片自动休眠时间（Type： 17、 18）	- 20 -
8 举例说明	- 23 -
9 生产测试指导	- 23 -
10 联系我们	- 23 -

1 概述

- 1.1 本文档适用于还有 T1L 开发充电宝的 MCU 端开发工程师使用。
- 1.2 本文档讲详细介绍硬件对接、固件对接。
- 1.3 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蓝牙芯片、app、云平台帮助客户的充电宝快速实现智能化，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 **充电宝 CID 固定默认为 0x008B, VID 固定默认为 0x0003, PID 默认为 0x0001**，使用此 CID PID VID 接入本公司充电宝小程序时，页面为本公司公版，如需个性化差异化定制，需向本公司申请 PID，请联系客户经理：邹曼娜 19065037169。
- 2.3 特别注意事项
 - 2.3.1 **T1L 芯片进入休眠模式，不支持开启低频广播。**
- 2.4 易连电宝管家小程序二维码：



3 蓝牙芯片版本

本文档支持的固件版本：
T1L_V3.0.0 等

深圳市易连物联网有限公司

电话：(86) 0755-81773367 邮箱：hw@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

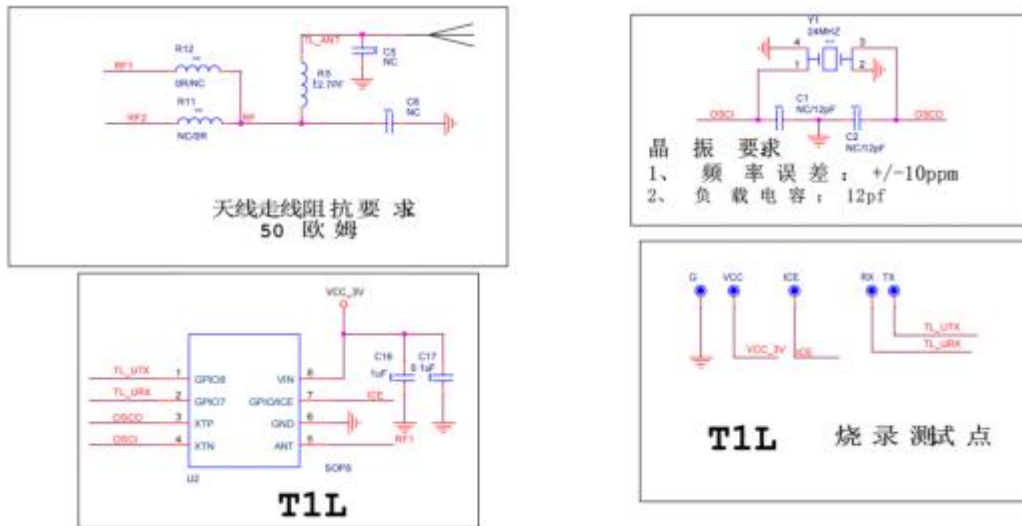
4 硬件参考设计

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 硬件参考设计

4.2.1 硬件参考设计图如下：



4.2.2 设计建议：

1. 画板建议使用 3.0VLDO（负载能力>300mA）。
2. 与 MCU 通信口：RX/TX。
3. 晶振和天线共用，R12 0R/R11 NC。
4. MCU 使用断电模式。
5. 设计原理图、 π 型天线封装可直接下载使用：

http://doc.elinkthings.com/web/#/40?page_id=358

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

深圳市易连物联网有限公司

电话：(86) 0755-81773367 邮箱：hw@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

5.2 UUID 说明

芯片有两个服务 UUID，一个是芯片固定的服务 UUID，为 FFE0，一个是用户可以自定义的服务 UUID，默认为 FFE0。

易联物联网的 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。

5.3 蓝牙连接服务列表 1： FFE0 举例

5.3.1 服务 UUID:

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

5.3.2 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

5.3.3 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

5.3.4 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

5.4 广播数据

第一类广播: AiLink 设置（默认）。使用我司 APP 接入的设置，需根据我司要求的格式进行设置。

5.4.1 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](#)）：

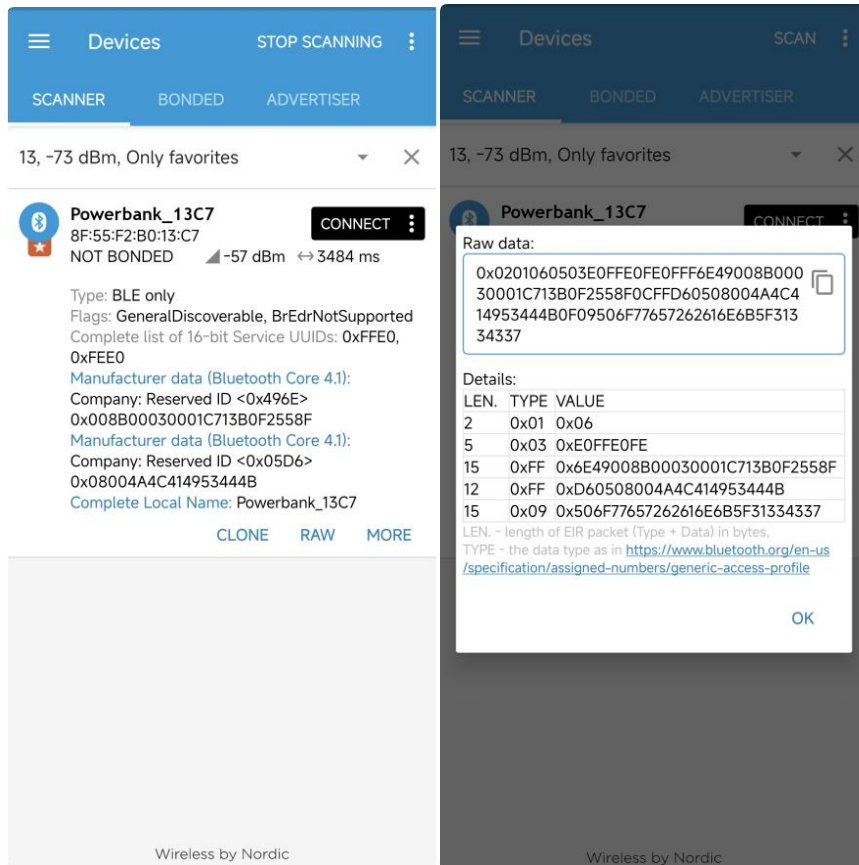
- 1、Company ID 。固定: 496e（In, Inet 缩写，此处预留后续申请 SIG company 号）
- 2、CID: 产品类型（2bytes）（例如血压计为 0x0001，额温枪为 0x0002，详查看[协议透传产品介绍](#)）
- 3、VID: 厂商 ID （2byte） （由我司分配）
- 4、PID: 产品 ID （2byte） （由厂商分配）

5、Mac 地址（MAC 是固定的，大小端序可设置，默认小端序）
例如广播出来的自定义数据为：

6e49008B00030001C713B0F2558F

6e49：为 In，**008B** 是 CID，表示产品类型，**0003** 是 VID，表示厂商 ID，**0001** 是 PID，表示产品 ID。
C713B0F2558F 是 Mac 地址，因为是小端序，所以 Mac 地址是：8F : 55 : F2 : B0 : 13 : C7

蓝牙工具显示如下图：



6 流程及软件协议

6.1 基础交互流程

1. 设备上电并唤醒芯片。
2. [蓝牙芯片返回状态信息](#)。
3. [MCU 设置 CID VID PID](#)。
4. APP 连接，蓝牙芯片会返回连接状态。
5. 连接 APP 后 MCU 主动发送充电宝状态信息。
6. 设备休眠关机。

6.2 充电宝指令集

A7 指令传输格式（MCU 每条指令的间隔大于 100ms）：

数据格式：

1. 内容说明：

OP:操作指令, 详情看列表

TLV:

T:type, 数据类型(1 字节)

L:length, 数据长度, 即 V 的长度(1 字节)

V:value, 数据内容(N 字节)

一条指令里可以拼接多个 TLV.

Byte	Value	Description
0	0xA7	包头
1	0x00	产品类型 (CID) 高字节
2	0x8B	产品类型 (CID) 低字节
3		Payload 长度(payload 部分的字节数量)
4		OP
5		T
6		L
7		V
....		...
n+1	SUM (1~n)	(1~n) 校验和(累加和, 取低八位)

n+2	0x7A	包尾
-----	------	----

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

6.2.1 OP 列表

op 值	功能描述	备注
0x01	App 设置功能	
0x02	App 查询信息	
0x03	MUC 主动返回状态信息	<p>MUC 实时读取充电宝状态，主动返回当前状态数据给 APP（根据需求返回充电宝状态数据）</p> <p>例：充电宝正在充电或者放电时，设备返回</p> <ol style="list-style-type: none"> 1、电池（NTC 获取）温度 2、主板（芯片内置温度传感器）温度 3、电量 4、电流 5、电压 6、功率 7、充电时间 8、充电量 9、充电 1%电量的时间（充电时才会返回此信息） 10、充满电还需时间（充电时才会返回此信息） <p>设备工作时，充电宝无状态时，设备返回</p> <ol style="list-style-type: none"> 1、电池（NTC 获取）温度 2、主板（芯片内置温度传感器）温度 3、电量 <p>设备边充边放时，设备返回</p> <ol style="list-style-type: none"> 1、电池（NTC 获取）温度 2、主板（芯片内置温度传感器）温度 3、电量 4、电流（总） 5、电压（总） 6、功率（总） 7、放电时间 8、充电时间

6.2.2 TLV 列表

如果 MUC 不支持 TLV 中 Type 对应的功能，则 MUC 返回 Length=1，Value=0xFF

Type	Length	Value
		uint8 无符号一个字节 ,int8 有符号一个字节 (如果没有特别说) 超过一个字节的数使用 小端序发送
0x01 充电宝功能列表 (只查)	N	根据下面充电宝功能列表决定字节数 uint8: 第一个字节 uint8: 第二个字节
0x02 充电宝充电状态 (只查)	1	uint8: 0: 充电宝正在充电 1: 充电宝没有在充电
0x03 充电宝放电状态和电流电压值/充电宝放电的端口 (只查)	N	uint8 : 充电宝正在放电的端口数量 0~255 0 表示充电宝没有端口在放电 { 端口 1: uint8 : 端口的类型 (0: Type-C 口 1: Type-A 口 ...) uint8 : 端口的编号 0-255 uint16: 放电时间 单位 s 分度 1s uint8: 放电量 单位% 分度 1% 如果设备不支持获得单个端口的放电量则 uint8=0xFF uint24: 电流值 单位 mA 分度 0.01mA uint24: 电压值 单位 mV 分度 0.01mV 如果设备不支持获得单个端口的电流电压值 则这 uint24=0xFF FF FF。 } { 端口 2 (如果有多个端口数量处于放电状态) uint8 : 端口类型 uint8 : 端口编号 uint16: 放电时间 单位 s 分度 1s uint8: 放电量 单位% 分度 1% 如果设备不支持获得单个端口的放电量则 uint8=0xFF uint24: 电流值 单位 mA 分度 0.01mA uint24: 电压值 单位 mV 分度 0.01mV 如果设备不支持获得单个端口的电流电压值 则这 uint24=0xFF FF FF。 } { 端口 3 同上格式 }
0x04 当前电量 (只查)	1	uint8 : 电量的大小 单位% 0~100
0x05 充电宝充电总电流 (只查)	3	uint24: 单位 mA 分度 0.1mA

0x06 充电宝放电总电流（只查）	3	uint24: 单位 mA 分度 0.1mA
0x07 充电宝充电总电压（只查）	3	uint24: 单位 mV 分度 0.1mV
0x08 充电宝放电总电压（只查）	3	uint24: 单位 mV 分度 0.1mV
0x09 VBAT/电池 电压（只查）	3	uint24: 单位 mV 分度 0.1mV
0x0A 电池（NTC 获取）温度（只查）	2	int16: 单位 °C 分度 0.1 °C
0x0B 主板（芯片内置温度传感器）温度（只查）	2	int16: 单位 °C 分度 0.1 °C
0x0C 当前电池容量（只查）	3	uint24: 单位 mAh 分度 1mAh
0x0D 充电宝充电功率（只查）	3	uint24: 单位 W 分度 0.01W
0x0E 充电宝放电功率（只查）	3	uint24: 单位 W 分度 0.01W
0x0F PD 快充功率（只查）	3	uint24: 单位 W 分度 0.01W
0x10 标配充电器功率（只查）	3	uint24: 单位 W 分度 0.01W
0x11 无线充电功率（只查）	3	uint24: 单位 W 分度 0.01W
0x12 充电宝充电次数（只查）	2	uint16: 次数的大小
0x13 出厂时间（查/设）	3	uint8 : yy 年份 2000+yy=实际年份 uint8 : 月份 uint8 : 日
0x14 激活时间（查/设）	6	uint8 : yy 年份 2000+yy=实际年份 uint8 : 月份 uint8 : 日 uint8 : 时 uint8 : 分 uint8 : 秒
0x15 关闭打开全部端口快充功能（设）	1	uint8 : 0-打开 1-关闭（app 设置 MUC 不写） uint8 : 0-成功 1-失败（设备回应 app 不写）
0x16 获得充电宝充电时间（查）	2	uint16: 单位 s 分度 1s
0x17 获得充电宝的端口的放电时间（查）	N	{ 端口 1: uint8 : 端口的类型（0: Type-C 口 1: Type-A 口 ...） uint8 : 端口的编号 0-255 uint16: 单位 s 分度 1s } { 端口 2: uint8 : 端口的类型（0: Type-C 口 1: Type-A 口 ...） uint8 : 端口的编号 0-255 uint16: 单位 s 分度 1s } { 端口 3: 格式同上 }

0x18 获得充电宝总充电量（查）	1	uint8: 单位% 分度 1% 当充电宝边充边放时,此时的充放电量为净电量,不是实际充放电电量
0x19 获得充电宝总放电量（查）	1	uint8: 单位% 分度 1%
0x1A 出厂时设置的电池容量（查/设）	3	uint24: 单位 mAH 分度 1mAH
0x1B 获得设备端口数量（查）	1	uint8: 单位 个 分度 1 个
0x1C 设置获取设备 SN 号(查/设)	N	N SN 号的 ASCLL 值
0x1D 获取充电宝充电 1%电量所需时间（查）	2	uint16: 单位 s 分度 1s
0x1E 获取充满电还需时间（查）	2	uint16: 单位 s 分度 1s
0x1F 获取每节电池的电压（查）	N	uint8 : 电池的节数 1-8 最多 8 节电池 { 电池 1: uint8: 电池编号 1 uint24: 单位 mV 分度 0.1mV } { 电池 2: uint8: 电池编号 2 uint24: 单位 mV 分度 0.1mV } {电池 3: 格式同上}

- 为了保证 APP 的功能和设备端的功能同步,APP 连接设备时,会主动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.

充电宝功能列表(0 不支持,1 支持)

Value	信息
Byte 1 Bit0	快充指示
Byte 1 Bit1	充电宝充电和放电状态指示
Byte 1 Bit2	端口状态指示
Byte 1 Bit3	输入输出总 VOUT 电压（输出给设备的电压）
Byte 1 Bit4	输入输出总 IBUS 电流
Byte 1 Bit5	VBAT 电压（电池电压）
Byte 1 Bit6	充电宝温度
Byte 1 Bit7	芯片温度
Byte 2 Bit0	当前电池容量
Byte 2 Bit1	电量
Byte 2 Bit2	充电宝充电和放电的功率
Byte 2 Bit3	电池端充放电功率

Byte 2 Bit4	保持设备工作
Byte 2 Bit5	关闭打开快充功能
Byte 2 Bit6	Pd 快充功率
Byte 2 Bit7	无线充电功率
Byte 3 Bit0	每个端口的单独输出电流
Byte 3 Bit1	每个端口的单独输出电压
Byte 3 Bit2	每个端口的单独充电电量
Byte 3 Bit3	每个端口的单独放电电量
Byte 3 Bit4	获取单节电池的电压

说明：支持功能列表持续更新，功能字节数随功能数量变化

APP 设置指令：T1L 收到后透传给 MUC

Byte	Default	Description
0	0xA7	包头
1~2	0x008B	产品类型：充电宝
3		Payload 长度
4	0x01	CMD: 0x01 设置指令
	TYPE	Payload
	Length	
	Value	
	TYPE	
	Length	
	Value	

	SUM	校验和
	0x7A	包尾

MUC 返回设置指令结果：T1L 透传给 APP

Byte	Default	Description
0	0xA7	包头
1~2	0x008B	产品类型：
3		Payload 长度
4	0x01	CMD: 0x01 设置指令
	TYPE	Payload
	Length	
	Value	
	TYPE	
	Length	
	Value	

	SUM	校验和
	0x7A	包尾

APP 查询指令: T1L 收到后透传给 MUC

Byte	Default	Description
0	0xA7	包头
1~2	0x008B	产品类型:
3		Payload 长度
4	0x02	CMD: 0x02 查询指令
	TYPE1	Payload
	TYPE2	
	...	
	SUM	
	0x7A	校验和
		包尾

MCU 返回查询信息:T1L 收到后透传给 APP

Byte	Default	Description
0	0xA7	包头
1~2	0x008B	产品类型:
3		Payload 长度
4	0x02	CMD: 0x02 查询指令
	TYPE	TLV 个数和 APP 设置的 TYPE 个数保持一致
	Length	
	Value	
	TYPE	
	Length	
	Value	

	SUM	校验和
	0x7A	包尾

MCU 主动返回充电宝信息: T1L 收到后透传给 APP

Byte	Default	Description
0	0xA7	包头
1~2	0x008B	产品类型:
3		Payload 长度
4	0x03	CMD:
	TYPE	TLV 个数和 APP 设置的 TYPE 个数保持一致
	Length	
	Value	
	TYPE	
	Length	
	Value	

	SUM	校验和

	0x7A	包尾
--	------	----

APP 收到 MCU 主动返回的指令时，不会回复 MUC。

当设备连接上 APP 后，MUC 需要实时主动返回充电宝信息。

7 蓝牙芯片通用指令集

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 获取蓝牙芯片状态（Type：25、26）

获取蓝牙芯片状态

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

蓝牙芯片返回状态：

Byte	Value	Description
0	0xA6	包头
1		Payload 长度
2	0x26	Type: 返回蓝牙芯片状态
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: 蓝牙芯片准备就绪	
5	Sum	(1~4)校验和	
6	0x6A	包尾	

7.2 设置、读取 CID、VID、PID (Type: 1D、1E)

- CID、VID、PID 是 ailink 蓝牙芯片重要的一组数据，设备要接入 ailink，必须要设置。具体 PID 的值可到我司 ailink 管理后台申请，详情请查看：

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	0x00	CID: 产品类型 ID 的高字节	
5	0x8B	CID: 产品类型 ID 的低字节	
6	0x00	VID: 厂商 ID 的高字节	
7	0x03	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	

蓝牙芯片回复设置结果:

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	

蓝牙芯片返回 ID 值:

Byte	Value	Description	
0	0xA6	包头	
1	Len	Payload 长度	
2	0x1E	Type: 返回 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	

7.3 APP 同步时间到 MCU (Type: 37、38)

对于某些设备，具有时间功能的，此时，可利用此指令进行数据的同步。

● APP 下发时间。

Byte	Default	Description	
0	0xA6	包头	
1		Payload 长度 (最大 15byte)	
2	0x37	Type: APP 同步时间	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~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	Payload
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	

7.4 读取蓝牙芯片版本号 (Type: 46)

■ 读取蓝牙芯片软硬件版本号:

Byte	Value	Description	
0	0xA6	包头	
1	0x01	Payload 长度	
2	0x46	Type: 读取蓝牙芯片软硬件版本号	Payload
3	0x47	(1~2)校验和	
4	0x6A	包尾	

➤ 蓝牙芯片返回软硬件版本号:

Byte	Value	Description	
0	0xA6	包头	
1		Payload 长度	
2	0x46	Type: 回复蓝牙芯片软硬件版本号	Payload
3		针对于某些芯片 MTU 不支持高于 23bytes,一条指令发不完版本号,固可分包发送. Bit7-4:指令条数(当只有 1 条指令时,该值为 0,类推) Bit3-0:当前指令 index(当是第一条指令时,该值为 0,类	

		推)	
4-N		版本号(ASCII 字符) 例如:T1LH1S2.0.0_20250530	
N+1	Sum	校验和	
N+2	0x6A	包尾	

7.5 设置芯片唤醒 (Type: 1A)

设置芯片唤醒:

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x1A	Type: 设置芯片唤醒	Payload
3	0x01	Value: 1: 唤醒芯片	
4	0x1D	(1~3)校验和	
5	0x6A	包尾	

芯片回复设置结果:

Byte	Value	Description	
0	0xA6	包头	
1	0x02	Payload 长度	
2	0x1A	Type: 回复设置芯片唤醒结果	Payload
3		结果值: 0: 成功 1: 失败 2: 不支持	
4	Sum	(1~3)校验和	
5	0x6A	包尾	

7.6 设置芯片进入睡眠 (Type: 19)

➤ **T1L 芯片进入休眠模式，不支持开启低频广播。**

➤ 当芯片进入休眠后，支持串口唤醒（MCU 可以发任意数据唤醒芯片，或者发送唤醒指令）

设置睡眠唤醒:

Byte	Value	Description
0	0xA6	包头

1	Len	Payload 长度		Payload
2	0x19	Type: 设置进入睡眠		
3		Value: 0x01		
4		睡眠后是否断开连接，是否开启低频广播： 0: 断开连接，关闭广播。 1: 保持连接，开启广播。 2: 断开连接 ， 开启广播 。 3: 保持连接 ， 关闭广播 。		
5		低频广播间隔时间的高字节	单位 ： ms ； 范围 20~2000 （ 建 议 1000ms）	
6		低频广播间隔时间的低字节		
7	Sum	(1~6)校验和		
8	0x6A	包尾		

芯片回复设置结果:

Byte	Value	Description		
0	0xA6	包头		
1	Len	Payload 长度		
2	0x19	Type: 回复设置进入睡眠的结果	Payload	
3		结果值: 0: 成功（成功后 100ms 后进入睡眠） 1: 失败 2: 不支持		
4	Sum	(1~3)校验和		
5	0x6A	包尾		

- MCU 和 APP 都可以设置芯片进入睡眠，芯片在回复 MCU/APP 时，同时向 APP/MCU 发送芯片当前状态“[返回芯片状态](#)”。

7.7 设置、读取芯片自动休眠时间 (Type: 17、18)

- **T1L 芯片设置定时休眠模式，不支持开启低频广播。**

可以设置芯片无数据 (UART/APP) 自动进入休眠模式 (低功耗模式，此时设备的 Tx 要保持为高)，但是要注意的是，当芯片进入 (休眠模式时，MCU 在发数据前，需要提前发一组数据过来唤醒芯片，芯片才能正常开始接受数据。)

注意：该设置芯片会保存。

设置自动睡眠时间:

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

0	0xA6	包头		Payload
1		Payload 长度		
2	0x17	Type: 设置无连接自动睡眠时间		
3		自动睡眠标志位: 0: 不开启自动休眠 1: 开启自动休眠, 芯片没有连接自动进入低功耗模式		
4		自动睡眠时间的最高字节	单位 : s 范围: 5 ~ 0xffff/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		睡眠后是否立刻断开连接, 是否开启低频广播: 0: 断开连接, 停止广播。 1: 保持连接, 开启广播。 2: 断开连接 , 开启广播 。 3: 保持连接 , 停止广播 。		
9		低频广播间隔时间的高字节	单位: ms ; 范围: 20~2000; (建议设 为 1000);	
10		低频广播间隔时间的低字节		
11	Sum	(1~10)校验和		
12	0x6A	包尾		

芯片回复设置结果:

Byte	Value	Description
0	0xA6	包头
1	0x02	Payload 长度
2	0x17	Type: 回复设置自动睡眠时间结果
3		结果值: 0: 成功 1: 失败 2: 不支持
4	Sum	(1~3)校验和
5	0x6A	包尾

获取自动睡眠时间:

Byte	Value	Description
0	0xA6	包头
1	0x01	Payload 长度
2	0x18	Type: 获取自动睡眠时间值
3	0x19	(1~2)校验和
4	0x6A	包尾

芯片返回自动睡眠时间值：

Byte	Value	Description	
0	0xA6	包头	
1	Len	Payload 长度	
2	0x18	Type: 返回无连接时自动睡眠时间	Payload
3		自动睡眠标志位: 0: 不开启自动休眠 1: 开启自动休眠, 芯片没有连接	
4		自动睡眠时间的最高字节	
5		自动睡眠时间的次高字节	
6		自动睡眠时间的次低字节	
7		自动睡眠时间的最低字节	
8		睡眠后是否开启低频广播: 0: 不开启 1: 开启	
9		低频广播间隔的高字节	
10		低频广播间隔的低字节	单位 : ms ; 范围 20~2000
11	Sum	(1~9)校验和	
12	0x6A	包尾	

8 举例说明

9 生产测试指导

T1L 生产测试指导说明详细文档请点击连接下载：http://doc.elinkthings.com/web/#!/40?page_id=389

10 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