

# 低功耗蓝牙透传模组 x95

## 使用说明书

Rev06 2021-01-12

本说明书适用于以下产品

### 透传模组

LSD4BT-E95ALSP001

LSD4BT-S95ALSP001

LSD4BTC-E95BLSP001

LSD4BTC-S95BLSP001

浙江利尔达物联网技术有限公司始终以为客户提供最及时、最全面的服务为宗旨，如需任何帮助,请随时联系我司各地分部或浙江总部，联系方式如下：

sss@lierda.com

**前言** 浙江利尔达物联网技术有限公司提供该文档内容用以支持其客户的产品设计。客户须按照文档中提供的规范，参数来设计其产品。由于客户操作不当而造成的人身伤害或财产损失，本公司不承担任何责任。在未声明前，利尔达公司有权对该文档进行更新。

**版权申明** 本文档版权属于利尔达公司，任何人未经我公司允许复制转载该文档将承担法律责任。

版权所有 © 利尔达科技集团，保留一切权利。

**Copyright © Lierda Science & Technology Group Co.,Ltd**

## 文件修订历史

固件版本	文件修订日期	作者	变更描述
<b>Rev01</b>	191102	WHJ	新建版本
	200102	WHJ	增添配对功能说明
	200320	WHJ	增加 S95 型号，删除容易产生误解项
	200422	COKIN	增加默认参数，修改 S95 脚号
<b>Rev04</b>	200520	COKIN	增加 Buffer 参数，修改文档模板
	200529	COKIN	修改引脚描述，修改广播类型设置描述，修改 0x41 指令长度
	200629	LJ	增加 UUID 修改指令
	200710	LJ	增加修改 MAC 地址指令，增加远程配置许可指令
	200723	LJ	修正部分查询指令返回格式
	200804	LJ	删除回连相关指令
<b>Rev05</b>	200825	COKIN	修正 S95 的串口引脚
	200901	COKIN	增加基础功耗描述，修改引脚描述问题
	200927	LJ	增加从机主动申请修改 MTU 指令
<b>Rev06</b>	210112	LJ	新增远程通道指令中指令格式描述
	210113	LJ	整合 x95A 和 x95B 型号说明书

## 目录

1	功能特点.....	5
2	硬件布局及接口说明.....	6
2.1	E95 模组.....	6
2.1.1	引脚布局.....	6
2.1.2	功能性引脚列表.....	7
2.2	S95 模组.....	8
2.2.1	引脚布局.....	8
2.2.2	功能性引脚列表.....	9
3	硬件设计规则.....	10
3.1	电气接线规则.....	10
3.2	Layout 规则.....	11
4	串口控制.....	12
4.1	串口帧结构.....	12
4.2	串口指令详解.....	13
5	参数.....	19
5.1	默认参数.....	19
5.2	功耗.....	19
5.2.1	不同引脚状态下的功耗.....	19
5.2.2	广播功耗.....	19
5.2.3	连接功耗.....	20
6	智能终端操作接口.....	21
6.1	服务及通道列表.....	21
6.2	下发通道.....	21
6.3	上传通道.....	21
6.4	远程指令通道.....	21
7	特殊指令操作.....	22
7.1	广播数据格式.....	22
8	透传.....	23
8.1	透传的时间规则与分包规则.....	23
8.2	透传操作示例.....	23
9	敬告用户.....	26

# 1 功能特点

本产品是基于 SIG 发布的 BLE 标准设计的低功耗蓝牙透传模组。模组的控制方式灵活，不仅可以通过串口和功能性引脚进行控制，而且支持移动端远程控制。使用该模组，用户无需关注复杂的蓝牙协议，就可在短期内开发出标准的 BLE 产品。

表 1- 1 模组功能简表

项目	说明
串口参数可配置	支持修改串口波特率（最高 921600），可设置串口输出延时。
连接参数可配置	可对连接间隔做修改，其余自动填充，也可自定义全部连接参数，可自动更新，也可手动触发。
广播参数可配置	支持修改广播间隔，广播数据，广播类型。
设备名称自定义	支持自定义不大于 20 字节的设备名称。
支持远程参数配置	支持手机远程配置参数，同时也可屏蔽此功能，防止恶意配置。
四线操作模式	支持仅通过连接 VCC-GND-RX-TX 四根线实现绝大部分功能。
传输速率	合适连接间隔条件下最高速率可达到 29KB/s 以上（极少数机型可能例外）。
设备信息接口	支持查询设备信息。
设备状态接口	支持查询设备状态（空闲、广播、连接），也可通过指令修改设备状态。
IO 控制	支持 IO 控制 CPU 唤醒和休眠、IO 控制蓝牙开关，IO 选择 UART 模式（透传或指令）。
IO 指示	支持 IO 指示串口数据输出，IO 指示连接状态，IO 指示透传可用状态。
BLE 5.0	支持 BLE 5.0 2MBPS PHY
模式切换	通过 IO 进行透传/配置模式切换。

## 2 硬件布局及接口说明

### 2.1 E95 模组

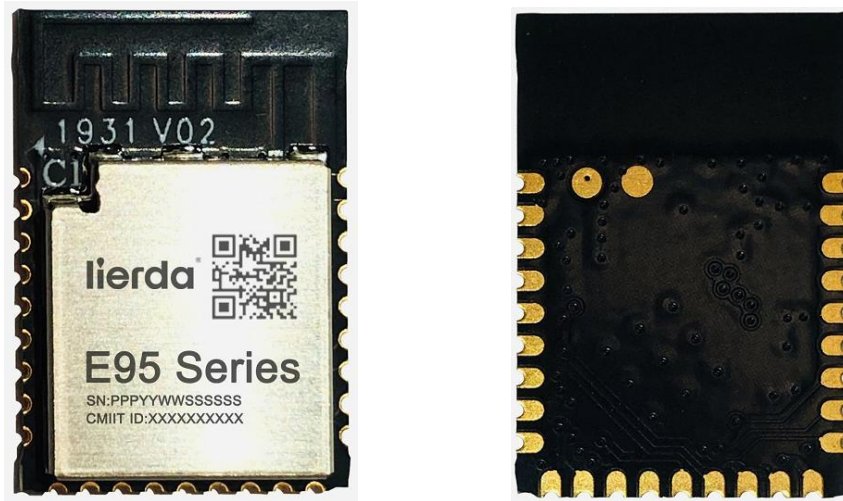


图 2- 1 E95 模组外观

#### 2.1.1 引脚布局

模组引脚布局如图 1 所示。

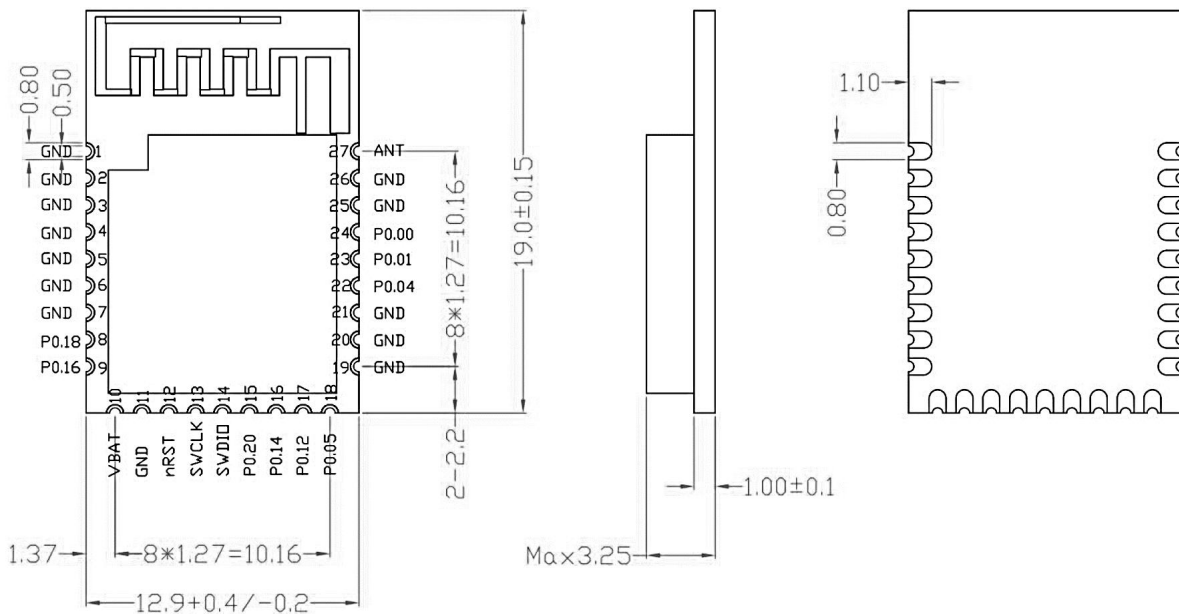


图 2- 2 E95 模组引脚布局

## 2.1.2 功能性引脚列表

序号	引脚	功能	方向	备注说明
1	GND	NC	/	不连接
2	GND	GND	/	接地
3	GND	GND	/	接地
4	GND	GND	/	接地
5	GND	GND	/	接地
6	GND	GND	/	接地
7	GND	GND	/	接地
8	P0.18	BUSY_IND	output	<b>数据拥塞指示引脚</b> (透传模式下才有意义, 指令模式下无意义) 高电平: 指示 MCU 可以继续向模组写入不超过 200 字节数据 低电平: 指示 MCU 不应该继续向模组写入数据 (有丢包风险)
9	P0.16	CONN_IND	output	<b>连接状态指示引脚</b> 高电平: 指示模组处于未连接状态 低电平: 指示模组处于连接状态
10	VCC	VCC	/	额定 3.3V
11	GND	GND	/	接地
12	RESET	RESET	input	<b>模组复位引脚</b> 高电平: 模组正常运行 低电平: 模组保持复位
13	SWCLK	SWCLK	/	/
14	SWD	SWD	/	/
15	P0.20	DOUT_IND	output	<b>串口数据指示引脚</b> 高电平: 指示模组此时没有数据要发送给 MCU 低电平: 指示模组即将或正在有数据发送给 MCU
16	P0.14	PWR_IND 复用 TEST_CTL	output	<b>休眠指示引脚</b> 高电平: 模组处于允许休眠状态 (不能接收串口数据) 低电平: 模组处于保持唤醒状态 (可以接收串口数据) <b>测试控制引脚 (上电有效, 后可作为普通引脚使用)</b> 置高: 用户模式 置低: 测试模式, 用户禁用 注意: 该引脚在上电 1s 内可悬空或接高电平, 切不可接低电平。
17	P0.12	TX	output	串口数据输出引脚
18	P0.05	RX	input	串口数据输入引脚
19	GND	GND	/	接地
20	GND	GND	/	接地
21	GND	GND	/	接地
22	P0.04	MODE_CTL	input	<b>模式控制引脚 (不可悬空)</b> 置高: 进入并保持透传模式 置低: 进入并保持指令模式
23	P0.01	PWR_CTL	input	<b>休眠控制引脚 (不可悬空)</b>

				置低：唤醒模组，并且不允许模组进入休眠状态 置高：允许模组进入休眠状态
24	P0.00	BT_CTL	input	<b>蓝牙控制引脚（不可悬空）</b> 置高：关闭蓝牙功能，停止广播或断开连接 置低：开始蓝牙广播，使它可以被主机（手机）发现
25	GND	GND	/	接地
26	GND	GND	/	接地
27	ANT	ANT	/	/

## 2.2 S95 模组

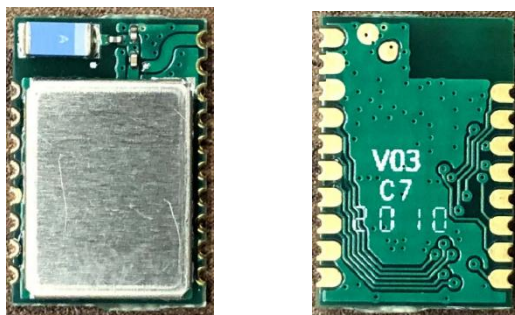


图 2- 3 S95 模组实物图

### 2.2.1 引脚布局

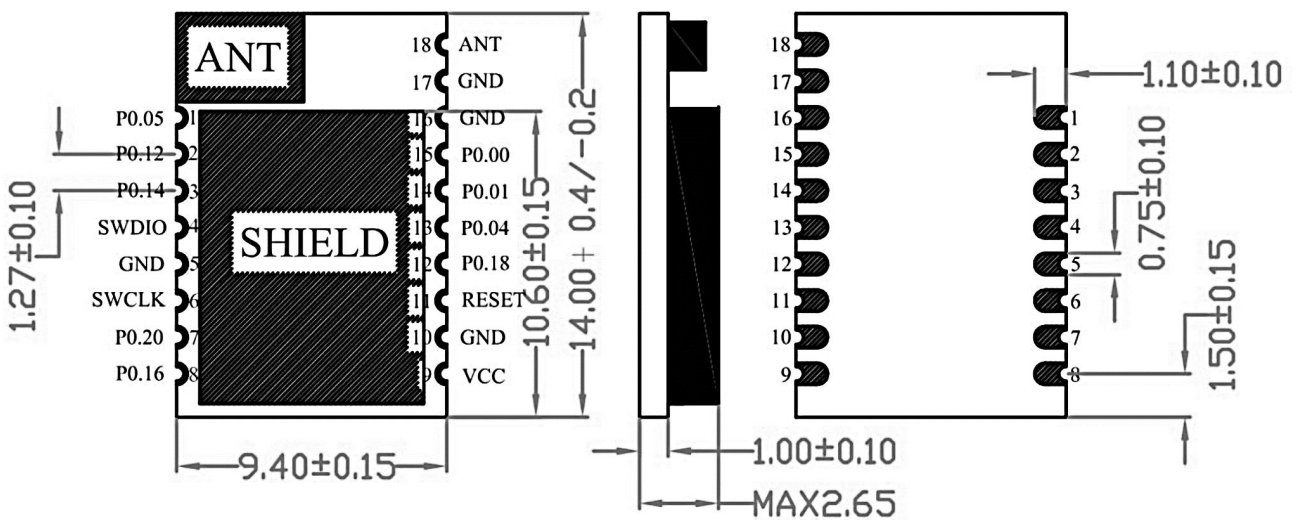


图 2- 4 S95 模组引脚布局



## 2.2.2 功能性引脚列表

序号	引脚	功能	方向	备注说明
1	P0.05	RX	input	串口数据输入引脚
2	P0.12	TX	output	串口数据输出引脚
3	P0.14	PWR_IND 复用 TEST_CTL	output	<b>休眠指示引脚</b> 高电平：模组处于允许休眠状态（不能接收串口数据） 低电平：模组处于保持唤醒状态（可以接收串口数据） <b>测试控制引脚（上电有效，后可作为普通引脚使用）</b> 置高：用户模式 置低：测试模式，用户禁用 注意：该引脚在上电 1s 内可悬空或接高电平，切不可接低电平。
4	SWDIO	SWDIO	/	/
5	GND	GND	/	接地
6	SWCLK	SWCLK	/	/
7	P0.20	DOUT_IND	output	<b>串口数据指示引脚</b> 高电平：指示模组此时没有数据要发送给 MCU 低电平：指示模组即将或正在有数据发送给 MCU
8	P0.16	CONN_IND	output	<b>连接状态指示引脚</b> 高电平：指示模组处于未连接状态 低电平：指示模组处于连接状态
9	VCC	VCC	/	额定 3.3V
10	GND	GND	/	接地
11	RESET	RESET	input	<b>模组复位引脚</b> 高电平：模组正常运行 低电平：模组保持复位
12	P0.18	PWR_CTL	input	<b>休眠控制引脚（不可悬空）</b> 置低：唤醒模组，并且不允许模组进入休眠状态 置高：允许模组进入休眠状态
13	P0.04	MODE_CTL	input	<b>模式控制引脚（不可悬空）</b> 置高：进入并保持透传模式 置低：进入并保持指令模式
14	P0.01	BUSY_IND	output	<b>数据拥塞指示引脚</b> （透传模式下才有意义，指令模式下无意义） 高电平：指示 MCU 可以继续向模组写入不超过 200 字节数据 低电平：指示 MCU 不应该继续向模组写入数据（有丢包风险）
15	P0.00	BT_CTL	input	<b>蓝牙控制引脚（不可悬空）</b> 置高：关闭蓝牙功能，停止广播或断开连接 置低：开始蓝牙广播，使它可以被主机（手机）发现
16	GND	GND	/	接地
17	GND	GND	/	接地
18	ANT	ANT	/	/

### 3 硬件设计规则

#### 3.1 电气接线规则

上一章节描述了各功能性引脚的含义，在这些引脚中，凡是描述为“不可悬空”的引脚都应该给予一个确定的电平，RX 引脚不使用时应该上拉至模组电源（注意 RX 电平和上拉电平要一致，否则会造成电流泄露）。在串口数据需要高速发送的应用场景中，建议将 BUSY\_IND 引脚与主控 MCU 的引脚相连，以监控数据发送状态，便于 MCU 处理高速数据，提高处理效率。

其他不使用的引脚应该保持悬空状态，如下图所示。

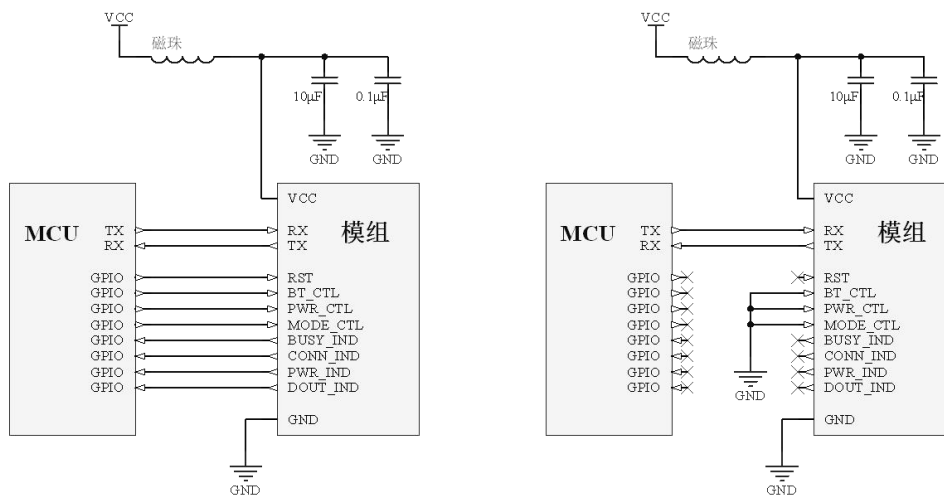


图 3- 1 全功能电气连接 (左) 和最简功能电气连接 (右)

#### 最简功能设计原则：

- 如果用户不需要快速的数据传输， BUSY\_IND 可不接；
- 如果用户不需要通过引脚获取蓝牙的连接状态， CONN\_IND 可不接，如果用户不需要通过引脚关闭蓝牙的广播功能， BT\_CTL 可直接接地，使模组保持广播状态，此时也可通过指令控制模组的蓝牙功能；
- 如果用户不需要通过引脚获取模组的功耗模式， PWR\_IND 可不接，如果用户不关心模组的功耗， PWR\_CTL 可直接接地，使模组保持唤醒状态；
- 如果用户的 MCU 不需要休眠或者在休眠状态下也能接收 UART 数据， DOUT\_IND 可不接；
- 如果用户不需要通过引脚控制模组的指令/透传模式切换， MODE\_CTL 可直接接地，使模组保持指令模式。

### 3.2 Layout 规则

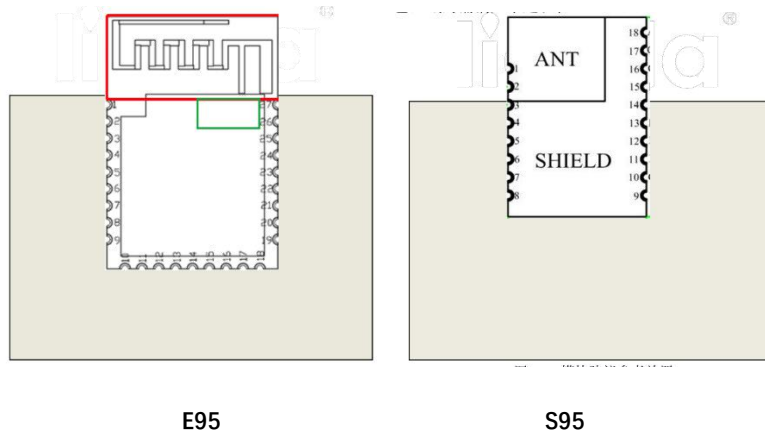


图 3- 2 Layout 规则示意图

Layout 规则如图 所示，总体要旨是靠近天线部分去除敷铜，在天线所在的空间范围内保持无明显的电磁阻碍部件，如金属外壳、电池、手持部位等，尽量远离电路中强电，高频、高噪声的部分。

## 4 串口控制

在指令模式下，MCU 给蓝牙模块发送一条指令，模块收到指令后解析并执行指令，并将指令执行的结果通过串口返回给 MCU，整个过程如下图所示。



图 4- 1 MCU 操作蓝牙模块流程图

### 4.1 串口帧结构

MCU 通过串口将指令发给蓝牙模块，为了便于模块识别，指令必须符合一定的格式；同时，蓝牙模块执行完指令以后也会按照约定的格式将数据返回给 MCU，输入帧指令格式如下表所示，响应帧指令格式如下表所示。

表 4- 1 输入帧指令格式 (hex)

起始符	指令 ID	指令长度	指令参数
01 FC	xx	xx	xx……xx

表 4- 2 响应帧指令格式 (hex)

起始符	指令 ID	指令长度	执行状态	应答数据
04 FC	xx	xx	xx	xx……xx

- ◆ **起始符**：所有的指令帧必须以“01 FC”开头，所有响应帧都以“04 FC”开头。
- ◆ **指令 ID**：不同功能指令的 ID 具有唯一性，下文有关于所有指令的枚举与详解。
- ◆ **指令长度**：输入帧长度即输入帧指令参数的字节数，响应帧长度即执行状态和应答数据的长度之和。
- ◆ **指令参数**：指令帧的指令参数是指令执行的参数。
- ◆ **执行状态**：执行状态是应答帧包含的内容，1 字节，紧跟在应答帧长度之后，00 表示命令成功接收，其他非零表示接收/处理失败。
- ◆ **应答数据**：应答数据是应答帧所携带的数据，具体见下文关于串口指令帧的详解。
- ◆ **数据单元**：指令中每个不低于 2 字节的独立数据单元，都应以低字节序传输，例如：1000 (0x03E8) 的传输顺序是 E8 03。

## 4.2 串口指令详解

ID	指令功能	说明	掉电保存																						
01	串口 参数设置	格式：01 FC 01 03 Baudrate ActiveMode Delay <b>Baudrate</b> : 1 字节无符号整数，表示串口波特率，取值如下： <table border="1" style="margin: 10px 0;"> <thead> <tr> <th>值</th> <th>00</th> <th>01</th> <th>02</th> <th>03</th> <th>04</th> <th>05</th> <th>06</th> <th>07</th> <th>08</th> <th>09</th> </tr> </thead> <tbody> <tr> <td>kbps</td> <td>2.4</td> <td>4.8</td> <td>9.6</td> <td>19.2</td> <td>38.4</td> <td>57.6</td> <td>115.2</td> <td>230.4</td> <td>460.8</td> <td>921.6</td> </tr> </tbody> </table> <b>ActiveMode</b> : 1 字节无符号整数，未使用。 <b>Delay</b> : 1 字节无符号整数，指定功能性引脚 DOUT_IND 拉低到串口输出数据之间的延时，单位 100 微秒，范围[0-100]。 注意：波特率修改不会立即生效，但最慢 60ms 后生效。 返回：04 FC 01 01 status	值	00	01	02	03	04	05	06	07	08	09	kbps	2.4	4.8	9.6	19.2	38.4	57.6	115.2	230.4	460.8	921.6	是
值	00	01	02	03	04	05	06	07	08	09															
kbps	2.4	4.8	9.6	19.2	38.4	57.6	115.2	230.4	460.8	921.6															
02	串口 参数查询	格式：01 FC 02 00 返回：04 FC 02 len status [Baudrate ActiveMode Delay]	/																						
03	连接 参数设置	格式 1：01 FC 03 02/03 CIMax [Delay] 格式 2：01 FC 03 08/09 CIMin CIMax Latency Timeout [Delay] <b>CIMin</b> : 2 字节无符号整数，连接间隔下限，单位毫秒，范围[8-4000]； <b>CIMax</b> : 2 字节无符号整数，连接间隔上限，单位毫秒，范围[8-4000]； <b>Latency</b> : 2 字节无符号整数，从机潜伏次数，范围[0-499]； <b>Timeout</b> : 2 字节无符号整数，连接丢失监测超时时间，单位毫秒，范围[100-32000]；如果超出范围，将归一到边界。 <b>Delay</b> : 1 字节无符号整数，自动更新延时，指定连接成功到自动更新连接参数的时间，单位 100 毫秒，范围[0-100]；特别地，该参数为 0 表示不自动更新连接参数，该参数省略时，默认为 0。 注意：这里配置的连接参数与实际连接参数之间存在微小误差（小于 2ms），不影响正常使用，可忽略。 其他说明： <ol style="list-style-type: none"> <li>1. 以格式 1 进行设置时，CIMin 默认 8ms, Latency 默认 0, Timeout 默认等于 8* CIMax, 并且强制归一到[3000-6000]ms 范围内。</li> <li>2. 所有参数必须满足如下两个条件：</li> </ol>	是																						

		$CIMin \leq CIMax - 1.25$ $Timeout > 2 \times (1 + Latency) \times CIMax$ <p>CIMin 和 CIMax 会向下对齐 1.25ms 整数倍, Timeout 向下对齐为 10ms 的整数倍。</p> <ol style="list-style-type: none"> <li>3. 特别注意, 通过格式 1 进行设置时, 由于 Timeout 进行了归一化处理, CIMax 的范围应该是[8-3000)。强制归一到[8-3000]ms 范围内</li> <li>4. 一旦收到该指令, 并且指令无误, 模组都会立即尝试进行连接参数更新, 如果 Delay 不为 0, 则每次连接建立时, 都会执行延时一段时间后自动更新连接参数的动作。</li> <li>5. 更新最终产生 CI、Latency、Timeout, 分别是连接间隔、从机潜伏和超时时间, <math>CIMin \leq CI \leq CIMax</math> (对于绝大多数智能终端, <math>CI \approx CIMax</math>), CI 越小, 通信速率和实时性越好, 功耗则偏高, 鉴于大多数的时间并不处于连接状态, 因此仍然建议不要将 CI 设置得太长, Timeout 越长, 丢包的容限越大, 连接丢失几率越小。</li> </ol> <p>返回: 04 FC 03 01 status</p>	
04	连接 参数查询	格式: 01 FC 04 00 返回: 04 FC 04 len status [CIMin CIMax Latency Timeout Delay]	/
05	配对密码设置	格式: 01 FC 05 04 PassCode PassCode: 4 字节无符号整数, 配对密码, 取值范围[0-999999] 10 进制, 十六进制设置时注意不要超出范围, 设为 0 表示无需配对加密。 该指令只允许串口使用。 返回: 04 FC 05 01 status	
06	配对密码查询	格式: 01 FC 06 00 返回: 04 FC 06 len status [PassCode] 注意: 该指令只允许串口使用。	
07	设备名称设置	格式: 01 FC 07 len Name <b>len:</b> 1 字节无符号整数, 设备名称字节数, 范围[1-20]; <b>Name:</b> 设备名称, 通常为可见的字符串。 返回: 04 FC 07 01 status	是
08	设备名称查询	格式: 01 FC 08 00 返回: 04 FC 08 len status [Name]	/
09	广播间隔设置	格式: 01 FC 09 02 ADVInterval <b>ADVInterval:</b> 2 字节无符号整数, 指定广播间隔, 单位毫秒, 范围[20-10240] 返回: 04 FC 09 01 status	是

0A	广播间隔查询	格式: 01 FC 0A 00 返回: 04 FC 0A len status [ADVInterval]	/
0B	发送功率设置	格式: 01 FC 0B 01 TxPwr TxPwr: 1 字节无符号整数, 指定发射功率, 取值范围[0-8], 取值码如下: 00: -40dBm      05: -4Bm 01: -20dBm      06: 0dBm 02: -16dBm      07: 3dBm 03: -12dBm      08: 4dBm 04: -8dBm 返回: 04 FC 0B 01 status	是
0C	发送功率查询	格式: 01 FC 0C 00 返回: 04 FC 0C len status [TxPwr]	/
0D	广播数据设置	格式: 01 FC 0D len SaveFlag ADVData len: 1 字节无符号整数, SaveFlag 和 ADVData 字节数之和, 取值范围[4-29]; SaveFlag: 1 字节无符号整数, 保存标志, 00 表示掉电不保存, 01 表示掉电保存; ADVData: 3-28 字节无符号整数, 广播数据, 数据格式详见下文 注意: 频繁修改广播数据时, SaveFlag 应该使用掉电不保存 返回: 04 FC 0D 01 status	可选
0E	广播数据查询	格式: 01 FC 0E 00 返回: 04 FC 0E len status [ADVData]	
0F	远程配置许可设置	格式: 01 FC 0F 01 Permission Permission: 1 字节无符号整数, 远程配置许可标识, 00-允许远程配置, 01-不允许远程配置。 注意: 该指令值允许串口使用。如果不希望参数被智能终端修改, 应该禁止远程配置。 返回: 04 FC 0F 01 status	
10	远程配置许可查询	格式: 01 FC 10 00 返回: 04 FC 10 len status [Permission]	
11	设置 UUID	格式: 01 FC 11 len item UUID len: item 和 UUID 的字节数之和, 可取 3 或者 17, 分别对应 16Bit 和 128Bit 的 UUID; Item: 1 字节无符号整数, 指定需要设置的 UUID 通道 (或服务), 取值如下: 00 - 透传服务 UUID	

		01 - 透传接收通道 UUID 【智能终端->蓝牙模组】 02 - 透传发送通道 UUID 【蓝牙模组->智能终端】 UUID: 2 字节或者 16 字节 UUID。 注意: 1、该指令需重启模组后生效。 2、设置 128Bit 的 UUID 时, 透传通道的 UUID 除第 12 和 13 字节, 其他字节需要与服务 UUID 保持一致。 返回: 04 FC 11 02 status item	
12	查询 UUID	格式: 01 FC 12 01 item 返回: 04 FC 12 len status [item UUID]	
13	设备状态设置	格式: 01 FC 13 01 State <b>State:</b> 1 字节无符号整数, 要设置的设备状态, 取值如下: 00-关闭蓝牙 (断开连接 (如果处于连接状态), 关闭广播) 01-开启蓝牙 (开启广播, 连接状态无效) 02-切换到广播状态 (连接状态无效, 都切换到广播状态) 返回: 04 FC 13 01 status	
14	设备状态查询	格式: 01 FC 14 00 返回: 04 FC 14 len status [State] 说明: State 有三种, 分别是: 00-空闲状态, 没广播也没有连接 01-模组正在广播 02-模组处于连接状态	
15	设置 MTU	格式: 01 FC 15 01 MTU <b>MTU:</b> 默认为 23, 空中包每可发送数据的长度为 (MTU-3), 范围[23, 252]。 <b>一次连接仅可设置一次。设置多次会返回错误码 04。</b> 返回: 04 FC 15 01 status <b>status:</b> 00-设置成功 01-未连接 02-参数不合法 03-设置超时 04-其他原因导致的失败	
18	产品信息查询	格式: 01 FC 18 01 item <b>item:</b> 1 字节无符号整数, 信息项, 取值码如下:	



		00-查询产品型号 01-查询固件版本 02-查询源码版本 03-查询 MAC 地址 04-查询最近一次断开连接的原因 05-当前的 MTU-3 (可传输的有效数据字节数) 值 06-保留 返回: 04 FC 18 len status [item Info] <b>Info:</b> 与 item 对应的信息数据。 当 item 为 04 时, 具体 info 信息请查看本模组遵循的协议规范 《Core_v5.1:Vol 2:PartD:Error Codes》	
19	恢复和复位	格式: 01 FC 19 01 item <b>item:</b> 1 字节无符号整数, 恢复和复位选项, 取值如下: 00-复位重启设备 (同复位按键功能) 01-恢复出厂设置, 该指令只允许串口使用 02-保留 返回: 04 FC 19 01 status。	
20	设置广播类型	格式: 01 FC 20 02 SaveFlag AdvType <b>SaveFlag:</b> 1 字节无符号整数, 0-掉电不保存, 1-掉电保存; <b>AdvType:</b> 1 字节无符号整数, 广播类型, 取值如下: 0x00: CONNECTABLE_SCANNABLE_UNDIRECTED 0x01: CONNECTABLE_NONSCANNABLE_DIRECTED_HIGH_DUTY_CYCLE 0x02: CONNECTABLE_NONSCANNABLE_DIRECTED 0x03: NONCONNECTABLE_SCANNABLE_UNDIRECTED 0x04: NONCONNECTABLE_NONSCANNABLE_UNDIRECTED 返回: 04 FC 20 01 status	
21	查询广播类型	格式: 01 FC 21 00 返回: 04 FC 21 02 status AdvType	
22	设置蓝牙 MAC 地址	格式: 01 FC 22 06 MAC MAC: 6 字节无符号整数, MAC 地址。 <b>备注: 需重启设备生效。</b> 返回: 04 FC 22 01 status	
23	查询蓝牙	格式: 01 FC 23 00	

	MAC 地址	返回: 04 FC 23 len status [MAC] MAC: 设备当前的 mac 地址。	
41	发送数据	格式: 01 FC 41 len Data <b>len</b> : 1 字节无符号整数, 需要发送数据的字节数, 1- (MTU-3-4) 字节; <b>Data</b> : 需要发送的数据。 返回: 04 FC 41 01 status	
42	接收数据	MCU 向模组发送该指令无效, 当模组收到数据, 并且模组处于指令模式时, 模组主动向 MCU 发送该 ID 指令作为响应, 格式如下: 返回: 04 FC 42 len Data	
4F	OTA 升级	格式: 01 FC 4F 06 MAC MAC: 模组的 MAC 地址, LSB format。 返回: 04 FC 4F 01 status	

## 5 参数

### 5.1 默认参数

表 5- 1 模组默认参数列表

项目	默认参数
串口波特率	9600bps
广播参数	广播间隔 100ms 广播数据 02:01:06:03:02:60:FE 扫描响应 13:09:BLE Device-XXXXXX,其中 XXXXXX 是设备地址后 3 字节
连接参数	最小连接间隔-8ms 最大连接间隔-10ms 从机潜伏-0 连接丢失监测超时时间 3 秒
发射功率	0dBm
接收缓存	1 KB

### 5.2 功耗

#### 5.2.1 不同引脚状态下的功耗

条件	平均功耗
BT_CTL = 1, BRT = 1	2uA ± 0.5 uA
BRT = 0	3.5 mA
BT_CTL = 0, BRT = 1	参考下表

#### 5.2.2 广播功耗

广播间隔 ms	广播字节数	休眠电流 uA	平均功耗 uA
100	7	2.35	74.16
200	7	2.35	38.25
500	7	2.35	16.71
1000	7	2.35	9.53

2000	7	2.35	5.94
5000	7	2.35	3.78
100	15	2.35	81.08
200	15	2.35	41.71
500	15	2.35	18.09
1000	15	2.35	10.22
2000	15	2.35	6.28
5000	15	2.35	3.92
100	25	2.35	94.51
200	25	2.35	48.43
500	25	2.35	20.78
1000	25	2.35	11.56
2000	25	2.35	6.95
5000	25	2.35	4.19

注意：此处为纯广播功耗，实际使用中会产生扫描包，功耗会比此处理论值要大

### 5.2.3 连接功耗

连接间隔 ms	休眠电流 $\mu\text{A}$	平均功耗 $\mu\text{A}$
20	2.35	201.07666
50	2.35	81.840664
100	2.35	42.095332
200	2.35	22.222666
500	2.35	10.2990664
1000	2.35	6.3245332

注意：以上数据为握手包功耗，不带实际用户数据。如带用户数据，功耗比理论值要大

## 6 智能终端操作接口

### 6.1 服务及通道列表

智能终端通过标准 GATT 协议与模组进行通信，下表所示的是模组所用的 GATT 服务及特征通道列表。

S/C	Describe	UUID	权限	说明
服务	/	0xFE60	/	利尔达低功耗蓝牙数据传输服务
特征值	Data M-S	0xFE61	WN/W	数据透传通道，移动端通过此通道将数据下发给 MCU
特征值	Data S-M	0xFE62	N	数据透传通道，蓝牙模组通过此通道将数据上传给移动端
特征值	Settings	0xFE63	WN/W/R/N	设置通道，移动端通过此通道修改和查询蓝牙模组的属性设置
特征值	RFU	0xFE64	WN/W/R/N	Reserve for Future Use

*M: Master, 表示主机*

*N: Notify*

*W: Write*

*S: Slave, 表示从机*

*R: Read*

*WN: Write no response, 不回复的写入*

### 6.2 下发通道

下发通道的 UUID 是 0xFE61，智能终端通过该通道以 Write Command 或者 Write Request 的方式将数据下发给模组，模组通过串口将数据发给主控 MCU，实现智能终端到主控 MCU 的数据透传。

### 6.3 上传通道

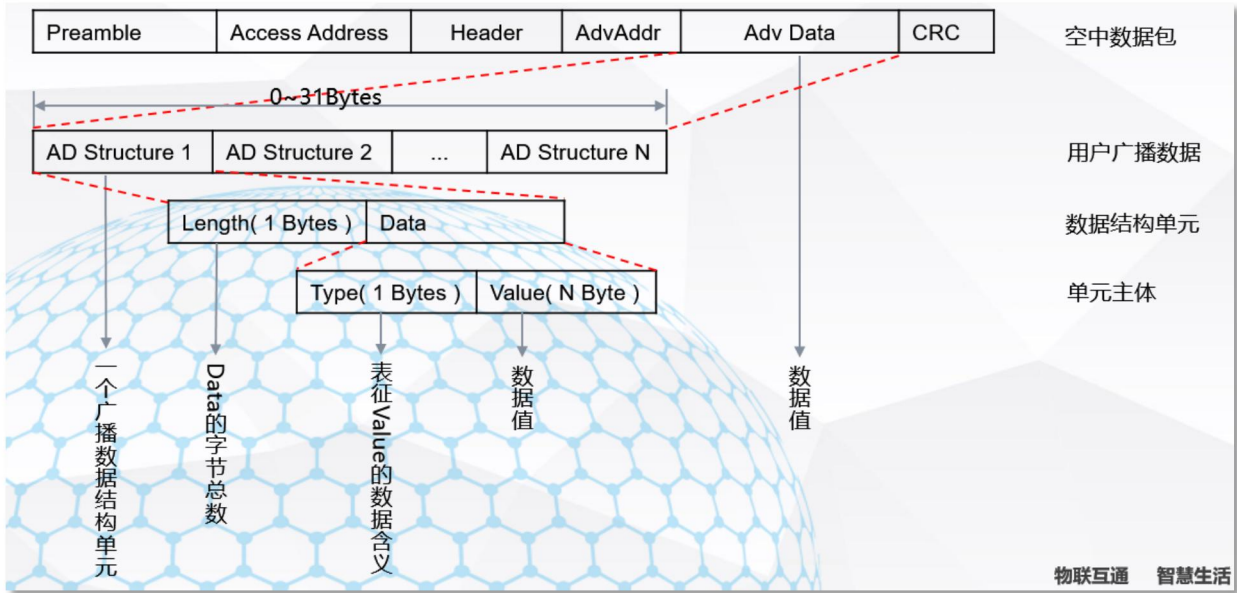
上传通道的 UUID 是 0xFE62，模组通过该通道以 Notify 的方式将来自主控 MCU 的数据上传给智能终端，实现主控 MCU 到智能终端的数据透传。

### 6.4 远程指令通道

远程指令通道的 UUID 是 0xFE63，该通道用于远程指令的收发，指令格式与串口一样。移动端通过该通道以 Write Command 的形式下发查询或设置指令，指令格式与串口指令相同，但不支持 ID 为 41、20、21 号指令。蓝牙模组收到指令后进行解析、执行，最后将执行的结果以 Notify 的形式发给智能终端。

# 7 特殊指令操作

## 7.1 广播数据格式



其中：

Type 是一个具有共用标准的取值，具体多少代表什么含义可到蓝牙联盟官网查询，常用的几个如下：  
 不完全 16Bit 服务 UUID (0x02)、不完全 128Bit 服务 UUID (0x06)、设备全名 (0x08)、设备简称 (0x09)、  
 厂商自定义数据 (0xFF)、...

在设置自定义广播时广播数据必须符合上图所示的用户数据单元的格式，否则不能设置成功。

## 8 透传

透传是指以模组为数据中转器件，将数据原样传送给接收端的方式，透传不会添加任何的其他数据。

### 8.1 透传的时间规则与分包规则

当模组检测到 RX 信号线上空闲时间长于 5 毫秒的接收时间或者连续收到的字节数达到 249 字节时，就会判定 MCU 的单次数据已经传输完成，会立即开始处理，鉴于此，在考虑到与蓝牙空中速率的配合，正常无丢包的数据传输应符合以下规则：

- 单次传输数据量不大于 249 字节，如大于 249 字节应该分为两次发给模组。
- 如果希望分两次传输的数据，两次传输之间的空闲时间应至少 5 毫秒。
- 如果 MCU 性能无限制，建议使用 115200bps 及以上的波特率。
- 模组处于激活状态（PWR\_CTL=0 并且 PWR\_IND=0）是蓝牙模组能够收到串口数据的必要条件。
- 默认情况下，空中数据的单帧最大字节数是 249，模组单次从串口收到的数据会被分包发送到空中，每包发送的字节数就是单帧最大字节数，如果不到单帧最大字节数，则以实际为准。
- 通信的速率受到串口波特率和智能终端性能的影响。因此，调节连接间隔是控制通信速率的主要方法之一，具体请参照串口指令中设置连接参数环节。
- 协调好串口波特率和蓝牙通信速率之间的关系，建议串口字节速率要达到蓝牙通信速率的 2-4 倍，给模组 CPU 留足空闲时间处理其他事件。

### 8.2 透传操作示例

根据数据量的不同，可以选择不同的操作逻辑，让模组发挥出最佳的性能。在数据量比较小时，可以直接发送，如示例 1 透传发送示例所示。

```
/* 数据发送示例 */
lsd_ble_send_data(void *buffer, unsigned char len)
{
    if( CONN_IND != 0 )
    {
        return NotConnected;
    }
    if( len > 200 )//超出200，直接返回
    {
```

```

        return InvalidLength;
    }

    PWR_CTL = 0; //激活模组
    while(PWR_IND==1); //等待模组状态切换成功，在PWR_IND没接的情况下也可以用delay_us(600)代替
    MODE_CTL = 1; //切换到透传模式

    mcu_send_data_via_uart(buffer, len); //发送数据
    delay_ms(5); //等待最后一字节数据发完，并且保证在下次发送之前模组RX引脚上有一定的空闲时间，防止帧间串扰
    /*
    //可选择恢复默认状态，节省功耗，方便设置
    PWR_CTL = 1; //允许休眠
    MODE_CTL = 0; //切换到命令模式
    */
    return SUCCESS;
}

```

### 示例 1 透传发送示例

数据量比较大时，如[错误!未找到引用源。](#)所示。

```

/* 大量数据发送示例 */
const unsigned char bytes_per_packet = 200;
lsd_ble_send_mass_data(buffer, len)
{
    unsigned int index = 0;

    if( CONN_IND != 0 )
    {
        return NotConnected;
    }

    PWR_CTL = 0; //激活
    while(PWR_IND==1); //等待模组状态切换成功，在PWR_IND没接的情况下也可以用delay_us(600)代替
    MODE_CTL = 1; //切换到透传模式

    while( index < len )
    {
        if(BUSY_IND==1)
        {
            //阻塞式发送串口数据，就是最后一字节发送完再返回
            mcu_send_data_via_uart(&buffer[index], (len-index)>bytes_per_packet?bytes_per_packet:len-index );
            delay_ms(5); //等待最后一字节数据发完，并且保证在下次发送之前模组RX引脚上有一定的空闲时间，防止帧间串扰
            index += bytes_per_packet;
        }
    }
}

```



```

    }
}
/* 可选择恢复默认状态, 节省功耗, 方便设置
PWR_CTL = 1; //允许休眠
MODE_CTL = 0; //切换到命令模式 */
return SUCCESS;
}

```

### 示例 2 透传发送示例 (大数据量)

如果没有使用 BUSY\_IND 引脚, 则可以通过延时发送的方式来控制发送速率, 如示例 3 透传发送示例 (大数据量延时发送) 所示

```

/* 大量数据发送示例 */
const unsigned char bytes_per_packet = 200;
lsd_ble_send_mass_data(buffer, len)
{
    unsigned int index = 0;
    if( CONN_IND != 0 )
    {
        return NotConnected;
    }
    PWR_CTL = 0; //激活
    while(PWR_IND==1); //等待模组状态切换成功, 在PWR_IND没接的情况下也可以用delay_us(600)代替
    MODE_CTL = 1; //切换到透传模式
    while( index < len )
    {
        //阻塞式发送串口数据, 就是最后一字节发送完再返回
        mcu_send_data_via_uart(&buffer[index], (len-index)>bytes_per_packet?bytes_per_packet:len-index);
        delay_ms(5); //等待最后一字节数据发完, 并且保证在下次发送之前模组RX引脚上有一定的空闲时间, 防止帧间串扰
        delay_ms(T); //T是蓝牙发送这个串口帧所需要的时间, 一般情况下T=bytes_per_packet*(l+1)/60
        //能满足大多数的情况; 当然, T长一点更安全, l是连接间隔, 单位是毫秒
        index += bytes_per_packet;
    }
    /* 可选择恢复默认状态, 节省功耗, 方便设置
    PWR_CTL = 1; //允许休眠
    MODE_CTL = 0; //切换到命令模式 */
    return SUCCESS;
}

```

### 示例 3 透传发送示例 (大数据量延时发送)

## 9 敬告用户

欢迎您使用利尔达科技有限公司的产品，在使用我公司产品前，请先阅读此敬告；如果您已开始使用说明您已阅读并接受本敬告。

利尔达科技有限公司保留所配备全部资料的最终解释和修改权，如有更改恕不另行通知。

编制：利尔达科技集团股份有限公司 无线传感网

2020年3月