

360°激光雷达STL27L

开发手册

目录

1.	产品描述	3
2.	通讯接口	4
3.	通讯协议	5
3.1.	数据包格式	5
3.2.	测量数据解析	8
3.3.	参考示例	9
4.	坐标系	10
5.	开发套件使用说明	11
5.1.	评估工具的使用方法	11
5.1.1.	硬件线材连接及说明	11
5.1.2.	windows 下驱动程序安装	12
5.1.3.	LdsPointCloudViewer 软件使用	14
5.2.	Linux 下基于 ROS 的使用操作	15
5.2.1.	ROS 环境介绍与安装	15
5.2.2.	获取功能包源码	15
5.3.	Linux 下基于 ROS2 的使用操作	16
5.3.1.	ROS2 环境介绍与安装	16
5.3.2.	获取功能包源码	16
5.4.	Linux 下 SDK 使用说明	17
5.4.1.	获取 SDK 源码	17
6.	修订记录	18

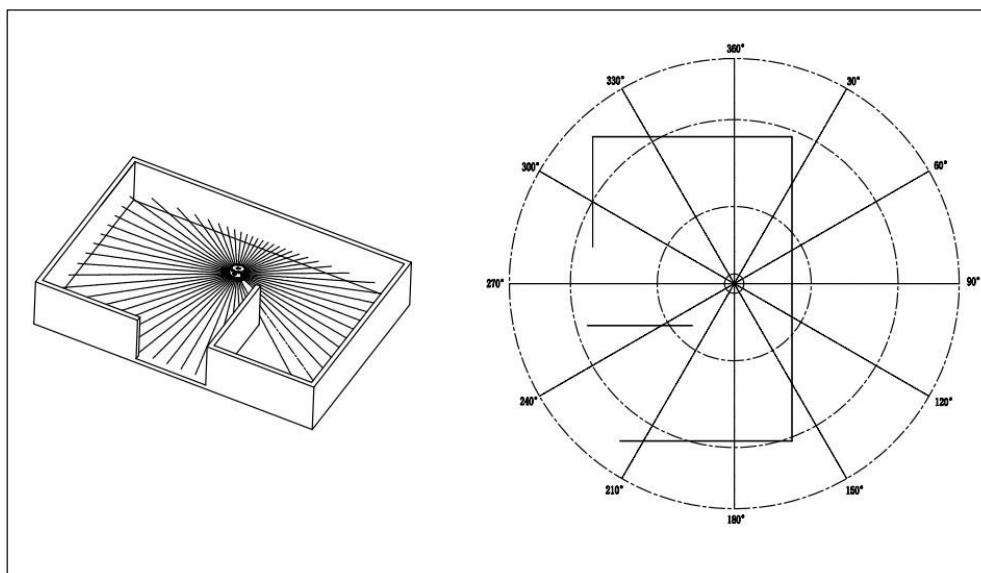
1. 产品描述

STL-27L 主要由激光测距核心，无线传电单元，无线通讯单元，角度测量单元、电机驱动单元和机械外壳组成。

STL-27L 测距核心采用DToF 技术，可进行每秒21600 次的测距。每次测距时，STL-27L 朝前发射出红外激光，激光遇到目标物体后被反射到单光子接收单元。由此，我们获取到了激光的发出时间和单光子接收单元收到激光的时间，两者的时间差即光的飞行时间，飞行时间再结合光速即可解算出距离。

获取到距离数据后，STL-27L 会融合角度测量单元测量到的角度值组成点云数据，然后通过无线通讯将点云数据发送到外部接口。STL-27L 支持内部控速，上电后 3 秒内转速可稳定至 $10 \pm 0.1\text{Hz}$ 。同时提供 PWM 外部输入接口，支持外部控速，外部控制单元获取到转速后，通过 PID 算法闭环控制，输入PWM 信号使 STL-27L 达到指定的转速。

STL-27L 点云数据形成的环境扫描图示，如下所示：



2. 通讯接口

STL-27L 使用 ZH1.5T-4P 1.5mm 连接器与外部系统连接，实现供电和数据接收，具体接口定义和参数要求见下图/表：



序号	信号名	类型	描述	最小值	典型值	最大值
1	Tx	输出	雷达数据输出	0V	3.3V	3.5V
2	PWM	输入	电机控制信号	0V	-	3.3V
3	GND	供电	电源负极	-	0V	-
4	P5V	供电	电源正极	4.5V	5V	5.5V

STL-27L 具有可无级调速的电机驱动器，支持内部控速和外部控速。在 PWM 引脚接地时，默认为内部调速，默认转速为 10Hz。外部控速需要在 PWM 引脚接入方波信号，可通过 PWM 信号占空比控制电机的启、停和转速。触发外部控速的条件：a、输入 PWM 频率 20-50K，推荐 30K；b、占空比在(45%, 55%)区间内(不包含 45%和 55%)，且最少 100ms 持续输入时间。触发外部控速后就一直处于外部控速状态，除非断电重启才会恢复内部控速；同时可以通过调节 PWM 占空比进行转速控制。由于每个产品电机的个体差异，占空比设置为典型值时实际转速可能会有差异，如要精确控制电机转速，需根据接收数据中的转速信息进行闭环控制。**注：不使用外部控速时，必须将 PWM 引脚接地。**

STL-27L 的数据通讯采用标准异步串口(UART)单向发送，其传输参数如下表所示：

波特率	数据长度	停止位	奇偶校验位	流控制
921600bit/s	8 Bits	1	无	无

3. 通讯协议

3.1. 数据包格式

STL-27L 采用单向通讯，稳定工作后，即开始发送测量数据包，不需要发送任何指令。测量数据包格式如下图所示。

起始符	VerLen	雷达转速		起始角度		数据	结束角度		时间戳		CRC 校验
54H	1 Byte	LSB	MSB	LSB	MSB	LSB	MSB	LSB	MSB	1 Byte

- **起始符**: 长度 1 Byte，值固定为 0x54，表示数据包的开始；
- **VerLen**: 长度 1 Byte，高三位表示帧类型，目前固定为 1，低五位表示一个包的测量点数，目前固定为 12，故该字节值固定为 0x2C；
- **雷达转速**: 长度 2 Byte，单位为度每秒；
- **起始角度**: 长度 2 Byte，单位为 0.01 度；
- **数据**: 一个测量数据长度为 3 个字节，详细解析请见下一小节；
- **结束角度**: 长度 2 Byte，单位为 0.01 度；
- **时间戳**: 长度 2 Byte，单位为 ms，最大为 30000，到达 30000 会重新计数；
- **CRC 校验**: 前面所有数据的校验和；

数据结构参考如下：

```
#define POINT_PER_PACK 12

#define HEADER 0x54

typedef struct __attribute__((packed))
{
    uint16_t distance;

    uint8_t intensity;
} LidarPointStructDef;

typedef struct __attribute__((packed)) {
```

```

uint8_t    header;

uint8_t    ver_len;

uint16_t   speed;

uint16_t   start_angle;

LidarPointStructDef point[POINT_PER_PACK];

uint16_t   end_angle;

uint16_t   timestamp;

uint8_t    crc8;

}LiDARFrameTypeDef;

```

CRC 校验计算方式如下：

```

static const uint8_t CrcTable[256]
={ 0x00, 0x4d, 0x9a, 0xd7, 0x79, 0x34,
0xe3,
0xae, 0xf2, 0xbf, 0x68, 0x25, 0x8b, 0xc6, 0x11, 0x5c, 0xa9, 0xe4, 0x33,
0x7e, 0xd0, 0x9d, 0x4a, 0x07, 0x5b, 0x16, 0xc1, 0x8c, 0x22, 0x6f, 0xb8,
0xf5, 0x1f, 0x52, 0x85, 0xc8, 0x66, 0x2b, 0xfc, 0xb1, 0xed, 0xa0, 0x77,
0x3a, 0x94, 0xd9, 0x0e, 0x43, 0xb6, 0xfb, 0x2c, 0x61, 0xcf, 0x82, 0x55,
0x18, 0x44, 0x09, 0xde, 0x93, 0x3d, 0x70, 0xa7, 0xea, 0x3e, 0x73, 0xa4,
0xe9, 0x47, 0x0a, 0xdd, 0x90, 0xcc, 0x81, 0x56, 0x1b, 0xb5, 0xf8, 0x2f,
0x62, 0x97, 0xda, 0x0d, 0x40, 0xee, 0xa3, 0x74, 0x39, 0x65, 0x28, 0xff,
0xb2, 0x1c, 0x51, 0x86, 0xcb, 0x21, 0x6c, 0xbb, 0xf6, 0x58, 0x15, 0xc2,
0x8f, 0xd3, 0x9e, 0x49, 0x04, 0xaa, 0xe7, 0x30, 0x7d, 0x88, 0xc5, 0x12,
0x5f, 0xf1, 0xbc, 0x6b, 0x26, 0x7a, 0x37, 0xe0, 0xad, 0x03, 0x4e, 0x99,
0xd4, 0x7c, 0x31, 0xe6, 0xab, 0x05, 0x48, 0x9f, 0xd2, 0x8e, 0xc3, 0x14,
0x59, 0xf7, 0xba, 0x6d, 0x20, 0xd5, 0x98, 0x4f, 0x02, 0xac, 0xe1, 0x36,
0x7b, 0x27, 0x6a, 0xbd, 0xf0, 0x5e, 0x13, 0xc4, 0x89, 0x63, 0x2e, 0xf9,
0xb4, 0x1a, 0x57, 0x80, 0xcd, 0x91, 0xdc, 0x0b, 0x46, 0xe8, 0xa5, 0x72,

```

```

0xef, 0x41, 0x0c, 0xdb, 0x96, 0x42, 0x0f, 0xd8, 0x95, 0x3b, 0x76, 0xa1,
0xec, 0xb0, 0xfd, 0x2a, 0x67, 0xc9, 0x84, 0x53, 0x1e, 0xeb, 0xa6, 0x71,
0x3c, 0x92, 0xdf, 0x08, 0x45, 0x19, 0x54, 0x83, 0xce, 0x60, 0x2d, 0xfa,
0xb7, 0x5d, 0x10, 0xc7, 0x8a, 0x24, 0x69, 0xbe, 0xf3, 0xaf, 0xe2, 0x35,
0x78, 0xd6, 0x9b, 0x4c, 0x01, 0xf4, 0xb9, 0x6e, 0x23, 0x8d, 0xc0, 0x17,
0x5a, 0x06, 0x4b, 0x9c, 0xd1, 0x7f, 0x32, 0xe5, 0xa8
};

uint8_t CalcCRC8(uint8_t *p, uint8_t
len){ uint8_t crc = 0;
uint16_t i;
for (i = 0; i < len; i++){
    crc = CrcTable[(crc ^ *p++) & 0xff];
}
return crc;
}

```

3.2. 测量数据解析

每个测量数据点由 2 个字节长度的距离值和 1 个字节长度的置信度值组成，如下图所示。

起始符	VerLen	雷达转速		起始角度		数据	结束角度		时间戳		CRC 校验
54H	2CH	LSB	MSB	LSB	MSB	LSB	MSB	LSB	MSB	1Byte



测量点 1		测量点 2		...	测量点 n			
距离值		信号强度			距离值		信号强度	
LSB	MSB	1 Byte		LSB	MSB	1 Byte		...

距离值的单位为 mm。信号强度值反映的是光反射强度，强度越高，信号强度值越大；强度越低，信号强度值越小。

每个点的角度值是通过起始角度和结束角度线性插值得来，其角度计算方法如下：

$$step = (end_angle - start_angle) / (len - 1);$$

$$angle = start_angle + step * i;$$

其中 len 为一个数据包的测量点数，i 的取值范围为 [0, len)。

3.3. 参考示例

假设我们收到了如下所示的一段数据

54 2C 68 08 AB 7E E0 00 E4 DC 00 E2 D9 00 E5 D5 00 E3 D3 00 E4 D0 00 E9 CD
00 E4 CA 00 E2 C7 00 E9 C5 00 E5 C2 00 E5 C0 00 E5 BE 82 3A 1A 50

我们对其解析如下：

起始符	VerLen	雷达转速		起始角度		数据	结束角度		时间戳		CRC 校验
54H	2CH	68H	08H	ABH	7EH	BEH	82H	3AH	1AH	50H

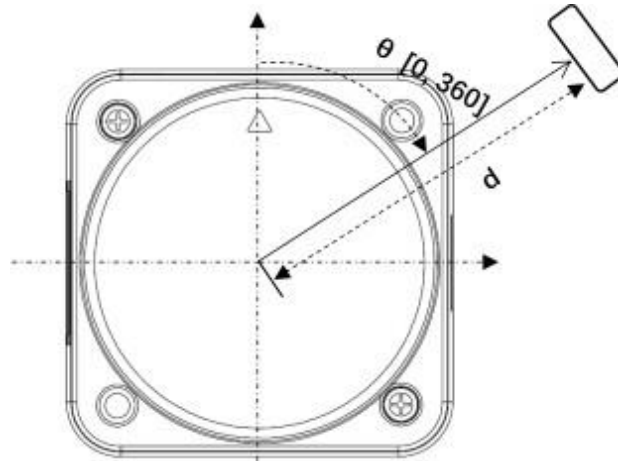


测量点 1		测量点 2				...	测量点 12		
距离值	信号强度	距离值		信号强度			距离值	信号强度	
E0H	00H	E4H	DCH	00H	E2H	...	B0H	00H	EAH

字段信息	解析
雷达转速	0868H = 2152 度/s;
起始角度	7EABH = 32427, 即 324.27 度
结束角度	82BEH = 33470, 即 334.7 度
测量点 1-距离值	00E0H = 224mm
测量点 1-信号强度	E4H = 228
测量点 2-距离值	00DCH = 200mm
测量点 2-信号强度	00E2H = 226
...	...
测量点 12-距离值	00B0H = 176mm
测量点 12-信号强度	EAH = 234

4. 坐标系

STL-27L 使用左手坐标系，旋转中心为坐标原点，传感器的正前方定义为零度方向，旋转角度沿着顺时针方向增大，具体如下图所示。



5. 开发套件使用说明

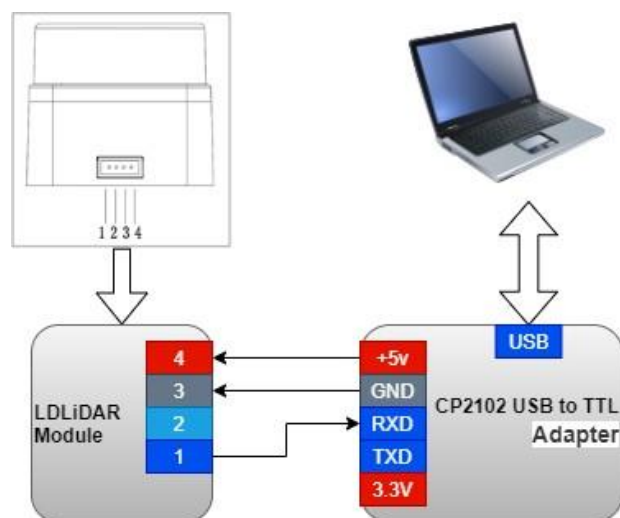
5.1. 评估工具的使用方法

5.1.1. 硬件线材连接及说明

1) 产品、线材、USB 转接板，如下图所示：



2) 连接示意，如下图所示：



5.1.2. windows 下驱动程序安装

在 windows 下对本公司产品进行评估时，需要安装 USB 转接板的串口驱动程序，原因是本公司提供的开发套件中 USB 转接板采用了 CP2102 USB 转串口信号芯片，其驱动程序可以从 Silicon Labs 的官方网站上进行下载：

<https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers>

解压 CP210x_Universal_Windows_Driver 驱动程序包后，执行驱动程序安装包目录下的 exe 文件，根据 Windows 系统版本，选择 X86(32 位)或者 X64(64 位)。

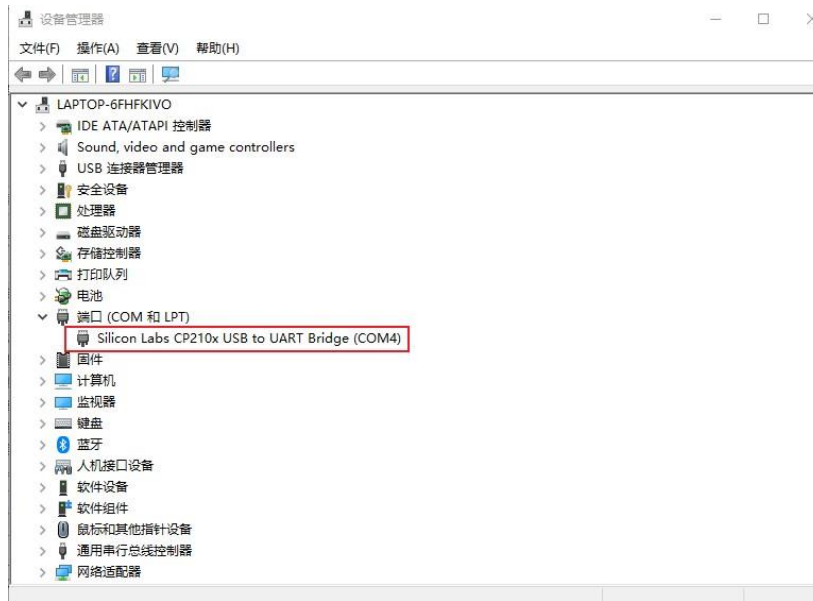
名称	修改日期	类型	大小
arm	2018/12/8 0:06	文件夹	
arm64	2018/12/8 0:06	文件夹	
x64	2018/12/8 0:06	文件夹	
x86	2018/12/8 0:06	文件夹	
CP210x_Universal_Windows_Driver_ReleaseNotes...	2018/12/7 23:53	TXT 文件	20 KB
CP210xVCPInstaller_x64.exe	2018/5/8 6:05	应用程序	1,026 KB
CP210xVCPInstaller_x86.exe	2018/5/8 6:05	应用程序	903 KB
dpinst.xml	2018/5/8 5:46	XML 文件	12 KB
silabser.cat	2018/12/4 2:17	安全目录	13 KB
silabser.inf	2018/12/4 2:17	安装信息	10 KB
SLAB_License_Agreement_VCP_Windows.txt	2016/4/27 22:26	TXT 文件	9 KB

双击 exe 文件，按提示进行安装操作。



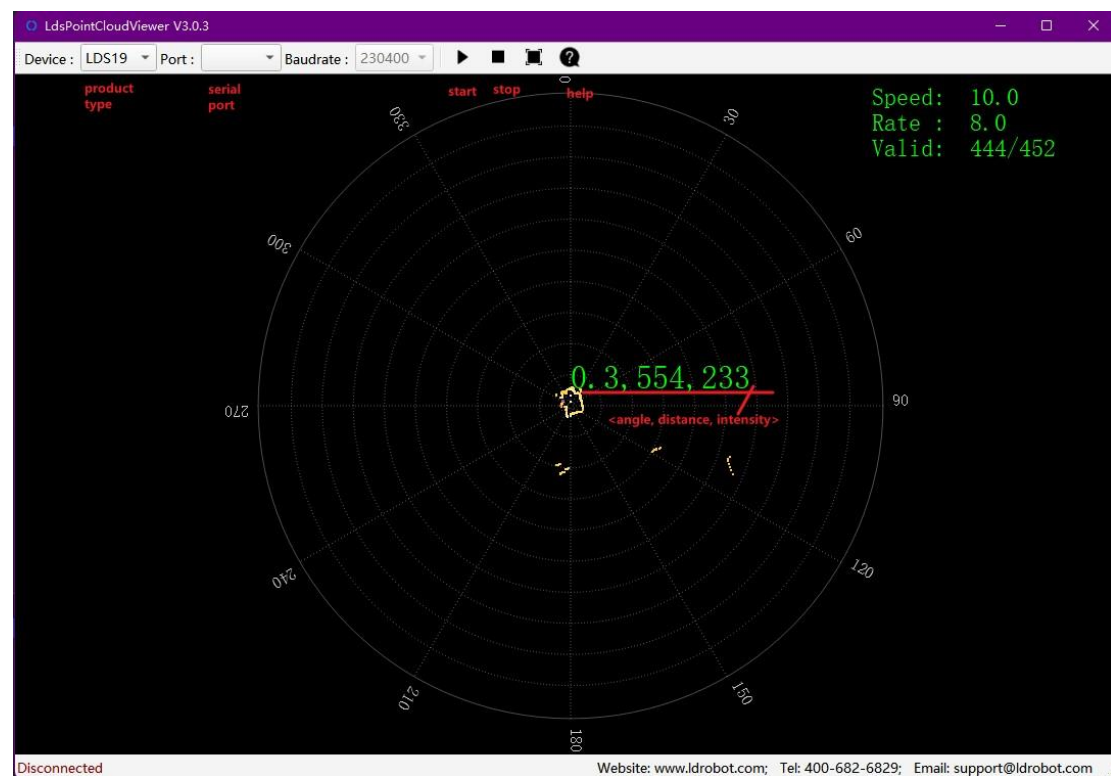
安装完成后，将开发套件中 USB 转接板与电脑相连，可以右键点击【我的电脑】，选择【属性】，在打开的【系统】界面下，选择左边菜单中的【设备管理器】进入到设备管理器，展开【端口】，可看到识别到的 CP2102 USB 适配器所

对应的串口号，即驱动程序安装成功，下图为 COM4。



5.1.3. LdsPointCloudViewer 软件使用

本公司提供了本产品实时扫描的点云可视化软件 LdsPointCloudViewer，开发者可使用该软件直观地观察本产品的扫描效果图。在使用该软件前要确保本产品的 USB 转接板的驱动程序已安装成功，并将本产品与 Windows 系统 PC 的 USB 口互连，然后双击 LdsPointCloudViewer.exe，选择对应的产品型号(如：STL27L)与端口号，点击启动点云刷新按钮，如下图所示。



上图中，Speed 表示激光雷达扫描频率，单位：Hz；Rate 表示激光雷达数据包解析速率；Valid 表示激光雷达测量一圈的有效点。

5.2. Linux 下基于 ROS 的使用操作

5.2.1. ROS 环境介绍与安装

ROS (Robot Operating System, 简称“ROS”) 是一个适用于机器人的开源的元操作系统和建立在 Linux 系统之上的中间件。它提供了操作系统应有的服务, 包括硬件抽象, 底层设备控制, 常用函数的实现, 进程间消息传递, 以及包管理。它也提供用于获取、编译、编写、和跨计算机运行代码所需的工具和库函数。ROS 各个版本的安装步骤, 请参考 ROS 官方网址: <http://wiki.ros.org/ROS/Installation>

本产品的 ROS 功能包, 支持的版本环境如下:

- ROS Kinetic(Ubuntu16.04);
- ROS Melodic(Ubuntu18.04);
- ROS Noetic(Ubuntu20.04).

5.2.2. 获取功能包源码

本产品 ROS 功能包源码, 需要联系本公司的 FAE 等营销人员来获取, 源码中的 README 文档有相关使用说明。若想通过 Github 或 Gitee 等托管平台利用私库进行相关开发对接, 同样请向本公司的 FAE 等营销人员反馈, 我们会努力满足你的开发需求。

5.3. Linux 下基于 ROS2 的使用操作

5.3.1. ROS2 环境介绍与安装

ROS (Robot Operating System, 简称“ROS”) 是一个适用于机器人的开源的元操作系统和建立在 Linux 系统之上的中间件。它提供了操作系统应有的服务, 包括硬件抽象, 底层设备控制, 常用函数的实现, 进程间消息传递, 以及包管理。它也提供用于获取、编译、编写、和跨计算机运行代码所需的工具和库函数。自 ROS 于 2007 年启动以来, 机器人和 ROS 社区发生了很大变化。ROS2 项目的目标是适应这些变化, 利用 ROS1 的优点并改进不足之处。ROS2 安装步骤, 请参考 ROS2 官方网址: <https://docs.ros.org/en/foxy/Installation.html>

本产品的 ROS2 功能包, 支持在 ROS2 Foxy(Ubuntu20.04)版本环境下使用。

5.3.2. 获取功能包源码

本产品 ROS2 功能包源码, 需要联系本公司的 FAE 等营销人员来获取, 源码中的 README 文档有相关使用说明。若想通过 Github 或 Gitee 等托管平台利用私库进行相关开发对接, 同样请向本公司的 FAE 等营销人员反馈, 我们会努力满足你的开发需求。

5.4. Linux 下 SDK 使用说明

5.4.1. 获取 SDK 源码

本产品 Linux SDK 源码，需要联系本公司的 FAE 等营销人员来获取，源码中的 README 文档有相关使用说明。若想通过 Github 或 Gitee 等托管平台利用私库进行相关开发对接，同样请向本公司的 FAE 等营销人员反馈，我们会努力满足你的开发需求。

