

轮 趣 科 技

乐动激光雷达传感器 STP-23L 使用手册

推荐关注我们的公众号获取更新资料



版本说明:

版本	日期	内容说明
V1.0	2022/05/12	第一次发布

网址: www.wheeltec.net

序言

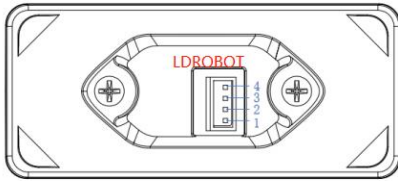
本手册主要介绍乐动激光雷达传感器 STP-23L 模块的连线，单片机、ROS 例程以及 Win10 上位机的使用方法。

目录

序言	2
1. STP-23L 模块	4
1.1 STP-23L 模块与 F1 单片机	4
1.2 STP-23L 模块的 ROS 驱动	6
1.3 STP-23L 模块的 Win10 上位机使用	11

1. STP-23L 模块

STP-23L 主要由激光测距核心和机械外壳组成，其使用 SMT 4PIN 1mm 连接器与外部系统连接，实现供电和数据接收如图 1-1-1：



序号	信号名	类型	描述	最小值	典型值	最大值
1	Tx	输出	UART TX	0V	3.3V	3.5V
2	RX	输入	UART RX	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

图 1-1-1 STP-23L 模块

TX 端口为传感器数据输出口，会相应发送数据

RX 端口不支持用户端指令下发到模组，仅用于生产使用

GND 端口为电源负极，P5V 端口为电源正极（电压典型值为 5V），根据上述端口描述则可对应连接 STP-23L 模块与外部系统：

1. 1STP-23L 模块与 F1 单片机

① F1 单片机烧录程序

资料中提供的 32 例程如图 1-1-2，其针对特定 F1 单片机如图 1-1-3（具体线路可参照相关原理图），可对例程解压并使用 Keil 打开查看例程源码，默认例程源码会在显示屏显示距离信息，具体烧录程序步骤可参照如图 1-1-4 相关教程：

STM32F1 例程源码

名称	修改日期	类型	大小
乐动激光雷达传感器STP-23例程.zip	2022/6/20 14:50	360压缩 ZIP 文件	6,948 KB

图 1-1-2 STP-23L 例程源码

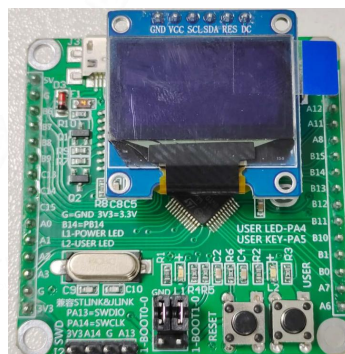


图 1-1-3 F1 单片机

机器人附送资料 > 1. Mini ROS 小车附送资料_2021.12.07 > 2. ROS从底层源码到进阶应用系列教程 > 1. STM32底层源码与ROS通讯视频教程

名称	修改日期	类型	大小
1. USB一键下载	2021/12/10 9:40	文件夹	
2. STLink下载程序	2021/12/10 9:42	文件夹	
3. STM32修改参数并生效+启动串口1发...	2021/12/10 9:40	文件夹	
4. 主板原理图	2021/12/10 9:41	文件夹	
5. 源码讲解：硬件初始化与车型选择	2021/12/10 9:41	文件夹	
6. 源码讲解：FreeRTOS任务与中断任务...	2021/12/10 9:41	文件夹	
7. 源码讲解：运动控制与PID	2021/12/10 9:42	文件夹	
8. 源码讲解：APP控制	2021/12/10 10:40	文件夹	
9. 源码讲解：航模与PS2手柄控制	2021/12/10 9:42	文件夹	
10. 源码讲解：串口与CAN控制	2021/12/10 9:40	文件夹	
11. 源码讲解：MPU9250初始化与陀螺仪...	2021/12/10 9:40	文件夹	
12. 源码讲解：人机交互	2021/12/10 9:40	文件夹	
13. 源码讲解：系统架构与总结	2021/12/10 9:40	文件夹	

图 1-1-4 32 程序烧录

② STP-23L 模块与 F1 单片机连线

将 STP-23L 模块与 F1 单片机对应端口连接如图 1-1-5：

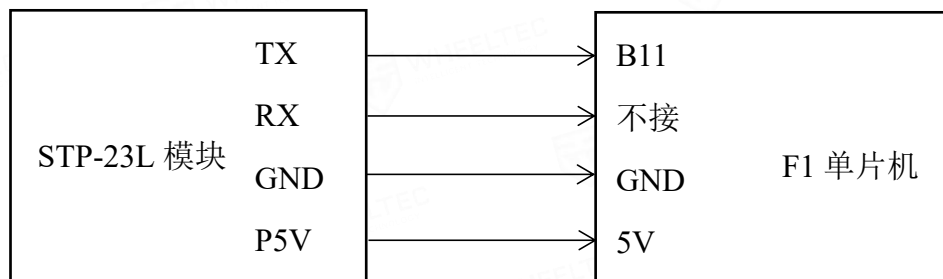


图 1-1-5 STP-23L 模块与 F1 单片机连线

完成连接后可在显示屏观察到相应物体与传感器的距离，在 串口助手端也能接收到距离信息，效果如图 1-1-6：

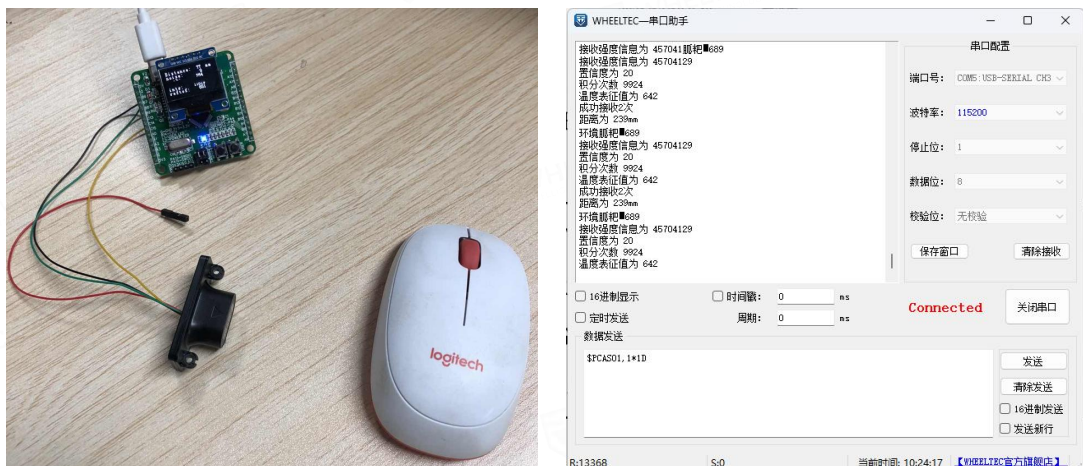


图 1-1-6 例程演示效果

1. 2STP-23L 模块的 ROS 驱动

① STP-23L 模块与 TTL 转 USB 模块连线

TTL 转 USB 模块如图 1-2-1，经 USB to Type-C 线可读取 STP-23L 模块数据后可结合 ROS 进行处理输出。

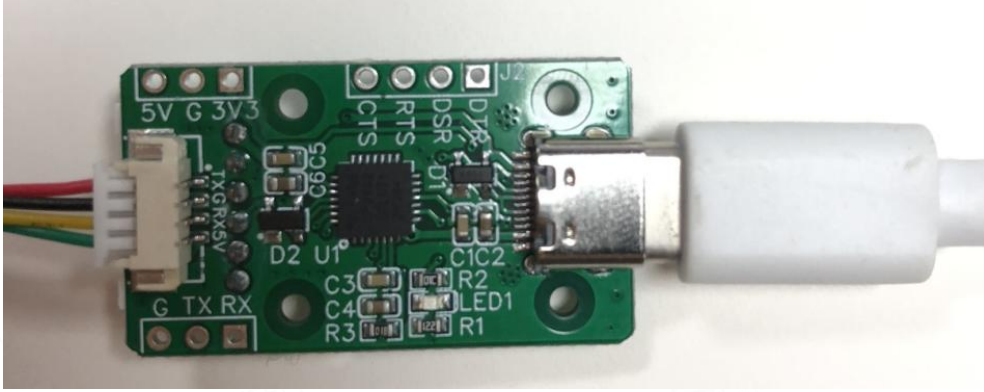


图 1-2-1 TTL 转 USB 模块

结合上一讲的 STP-23L 模块端口描述，STP-23L 模块与 TTL 转 USB 模块连线情况应如图 1-2-2：

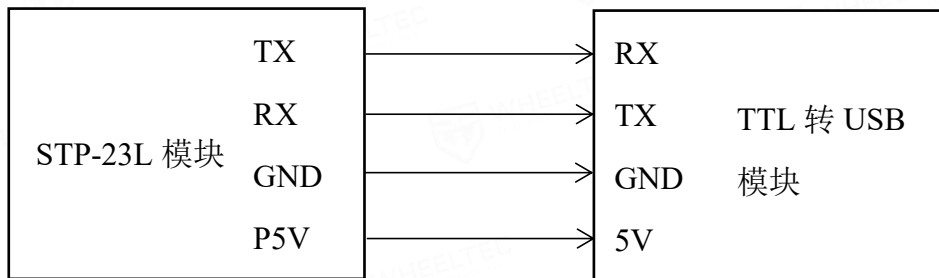


图 1-2-2 STP-23L 模块与 TTL 转 USB 模块连线

连线仅需将提供的线对应插入接口即可，效果应如图 1-2-3 所示：

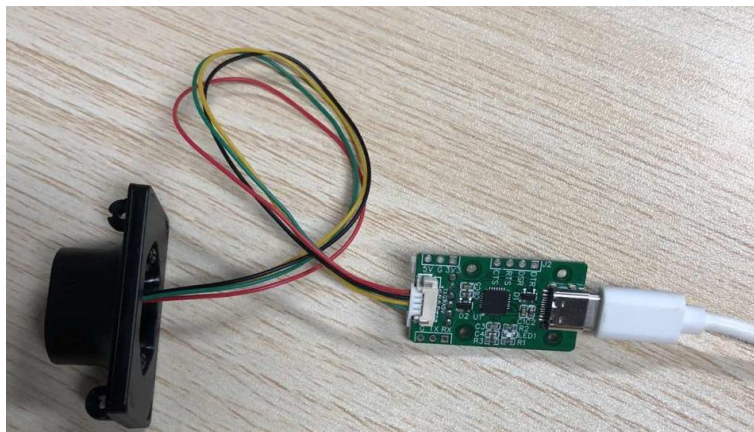
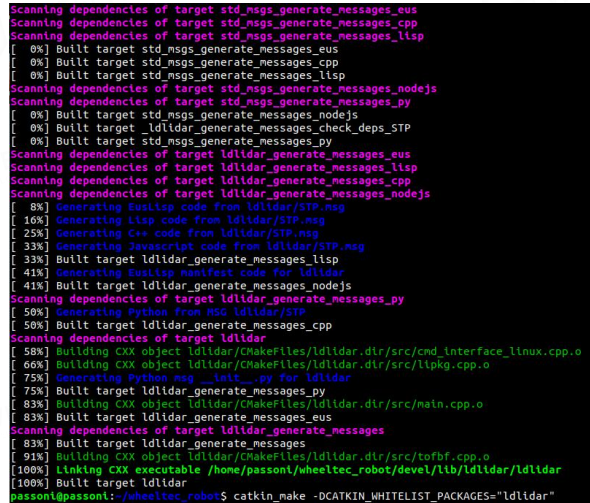


图 1-2-3 连线实物图

② STP-23L 模块 ROS1 功能包的使用

把用户资料 1.Linux、ROS/STP23L_ROS1_SDK 源码中的 stp23l_ws 里的两个功能包(ldlidar_driver 和 ros_app)拷贝到 ROS1 的工作空间中，编译功能包。

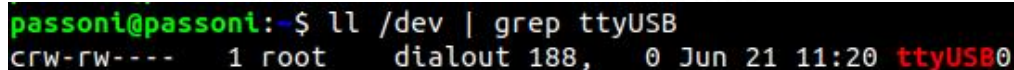
```
catkin_make -DCATKIN_WHITELIST_PACKAGES="ldlidar"
source devel/setup.bash
```



```
Scanning dependencies of target std_msgs_generate_messages_eus
Scanning dependencies of target std_msgs_generate_messages_cpp
Scanning dependencies of target std_msgs_generate_messages_lisp
[ 0%] Built target std_msgs_generate_messages_eus
[ 0%] Built target std_msgs_generate_messages_cpp
[ 0%] Built target std_msgs_generate_messages_lisp
Scanning dependencies of target std_msgs_generate_messages_nodejs
Scanning dependencies of target std_msgs_generate_messages_py
[ 0%] Built target std_msgs_generate_messages_nodejs
[ 0%] Built target std_msgs_generate_messages_py
Scanning dependencies of target ldlidar_generate_messages_check_deps_STP
[ 0%] Built target ldlidar_generate_messages_check_deps_STP
Scanning dependencies of target ldlidar_generate_messages_eus
Scanning dependencies of target ldlidar_generate_messages_lisp
Scanning dependencies of target ldlidar_generate_messages_cpp
Scanning dependencies of target ldlidar_generate_messages_nodejs
[ 8%] Generating C++ code from ldlidar/STP.msg
[ 16%] Generating C++ code from ldlidar/STP.msg
[ 25%] Generating C++ code from ldlidar/STP.msg
[ 33%] Generating Javascript code from ldlidar/STP.msg
[ 33%] Built target ldlidar_generate_messages_lisp
[ 41%] Generating C++ code from ldlidar/STP.msg
[ 41%] Built target ldlidar_generate_messages_nodejs
Scanning dependencies of target ldlidar_generate_messages_py
[ 50%] Generating Python from ROS ldlidar/STP
[ 50%] Built target ldlidar_generate_messages_py
Scanning dependencies of target ldlidar
[ 58%] Building CXX object ldlidar/CMakeFiles/ldlidar.dir/src/cnd_interface_linux.cpp.o
[ 66%] Building CXX object ldlidar/CMakeFiles/ldlidar.dir/src/lipkg.cpp.o
[ 75%] Generating Python msg _init_.py for ldlidar
[ 75%] Built target ldlidar_generate_messages_py
[ 83%] Building CXX object ldlidar/CMakeFiles/ldlidar.dir/src/main.cpp.o
[ 83%] Built target ldlidar_generate_messages_eus
Scanning dependencies of target ldlidar_generate_messages
[ 83%] Built target ldlidar_generate_messages
[ 91%] Building CXX object ldlidar/CMakeFiles/ldlidar.dir/src/tofbf.cpp.o
[ 100%] Linking CXX executable /home/passont/wheeltec_robot/devel/lib/ldlidar/ldlidar
[100%] Built target ldlidar
passont@passont:~/wheeltec_robot$ catkin_make -DCATKIN_WHITELIST_PACKAGES="ldlidar"
```

图 1-2-4 编译 ldlidar

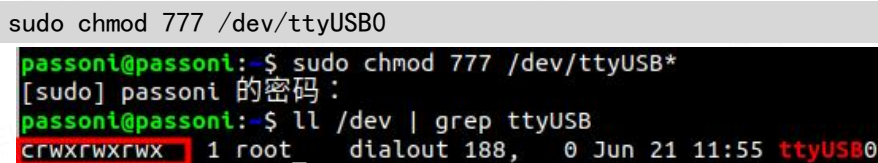
把 STP23L 模块接入到主板中，修改串口设备权限，(以/dev/ttyUSB0 为例)，实际使用时，根据模块在系统中的实际挂载情况来设置，可以使用`ll /dev | grep ttyUSB`命令查看：



```
passont@passont:~$ ll /dev | grep ttyUSB
crw-rw----  1 root  dialout 188,  0 Jun 21 11:20 ttyUSB0
```

图 1-2-5 查看设备号

输入修改权限指令，执行完成后可发现权限变更为 crwxrwxrwx：



```
passont@passont:~$ sudo chmod 777 /dev/ttyUSB0
[sudo] passont 的密码：
passont@passont:~$ ll /dev | grep ttyUSB
crwxrwxrwx  1 root  dialout 188,  0 Jun 21 11:55 ttyUSB0
```

图 1-2-6 修改设备权限

默认以/dev/ttyUSB 启动模块，如有变动可在以下文件中修改

```
/home/wheeltec/stp23l_ws/src/ros_app/src/ldlidar/launch/stp23l.launch
```

```

<launch>
<!-- ldlidar message publisher node -->
<node name="STP231L" pkg="ldlidar" type="stp231_rosnode" output="screen" >
  <param name="product_name" value="LDLIDAR_STP231L"/>
  <param name="topic_name" value="laser"/>
  <param name="port_name" value="/dev/ttyUSB0"/>
  <param name="port_baudrate" value="230400"/>
  <param name="frame_id" value="base_laser"/>
</node>
<!-- publisher tf transform, parents frame is base_link, child frame is
base_laser -->
<!-- args="x y z yaw pitch roll parents_frame_id child_frame_id period_in_ms"-->
<node name="base_to_laser_stp231" pkg="tf" type="static_transform_publisher"
args="0.0 0.0 0.18 0.0 0.0 0.0 base_link base_laser 100"/>
</launch>

```

图 1-2-7 stp231.launch 文件内容

启动模块

roslaunch ldlidar stp231.launch

启动成功后，终端会不断打印距离信息。

```

[INFO] [1683774633.971086287]: Wait recv lidar version information.
[INFO] [1683774633.975476893]: Get lidar version information is success.
[INFO] [1683774633.975582878]: lidar firmware version:v1.1.8
[INFO] [1683774633.97598943]: lidar hardware version:v0.0.23
[INFO] [1683774633.975994492]: lidar manufacture times:1287-231-7,3:42:12
[INFO] [1683774633.975936705]: lidar mcu id number:81311038373534d50120536
[INFO] [1683774633.976185539]: lidar sn code:#
?? ????????
[INFO] [1683774633.976337540]: lidar pitch angle:65535,65535,65535,65535
[INFO] [1683774633.976431044]: lidar blind area, near:30,far:7500
[INFO] [1683774633.976527493]: lidar frequency val:120
[INFO] [1683774633.976618488]: Start measure.
[LD5][INFO] [1683774633.993143815]:[ACK,id:0x2,result:1]
[INFO] [1683774634.080367950]: get lidar frame Data, timestamp:26930557
[INFO] [1683774634.081247573]: Measure data:
[INFO] [1683774634.081937530]: distance:226,noise:6,peak:807781,confidence:100,intg:8552,reftof:336
[INFO] [1683774634.082505983]: -----
[INFO] [1683774634.082984386]: Measure data:
[INFO] [1683774634.083442950]: distance:225,noise:5,peak:1754865,confidence:100,intg:7996,reftof:339
[INFO] [1683774634.083958066]: -----
[INFO] [1683774634.084266380]: Measure data:
[INFO] [1683774634.084372749]: distance:227,noise:7,peak:792785,confidence:100,intg:8367,reftof:336
[INFO] [1683774634.084545422]: -----
[INFO] [1683774634.084752976]: Measure data:
[INFO] [1683774634.084951282]: distance:227,noise:5,peak:767100,confidence:100,intg:8137,reftof:338
[INFO] [1683774634.085045590]: -----
[INFO] [1683774634.085396661]: Measure data:
[INFO] [1683774634.085755414]: distance:226,noise:5,peak:767100,confidence:100,intg:8137,reftof:338
[INFO] [1683774634.085756088]: -----
[INFO] [1683774634.085904601]: Measure data:
[INFO] [1683774634.086046426]: distance:225,noise:4,peak:763489,confidence:100,intg:8121,reftof:339
[INFO] [1683774634.08622419]: -----
[INFO] [1683774634.086469501]: Measure data:
[INFO] [1683774634.086810176]: distance:226,noise:7,peak:824645,confidence:100,intg:8746,reftof:336

```

图 1-2-8 终端打印内容

运行过程中会发布话题/laser，可以在终端输入命令查看话题信息：

rostopic echo /laser

```

wheeltec@wheeltec:~$ rostopic list
/laser
/rosout
/rosout_agg
/tf
wheeltec@wheeltec:~$ rostopic echo /laser
header:
  seq: 2921
  stamp:
    secs: 1683774722
    nsecs: 412759941
  frame_id: "base_laser"
radiation_type: 1
field_of_view: 0.10000000149
min_range: 0.0
max_range: 12.0
range: 0.224999999404
---
header:
  seq: 2922
  stamp:
    secs: 1683774722
    nsecs: 413767049
  frame_id: "base_laser"
radiation_type: 1
field_of_view: 0.10000000149
min_range: 0.0
max_range: 12.0
range: 0.224999999404
---
header:
  seq: 2923
  stamp:
    secs: 1683774722
    nsecs: 414027434
  frame_id: "base_laser"
radiation_type: 1
field_of_view: 0.10000000149
min_range: 0.0
max_range: 12.0
range: 0.2259999996066
---

```

图 1-2-9 rostopic echo /laser

③ STP-23L 模块 ROS2 功能包的使用

把用户资料 1.Linux、ROS/STP23L_ROS2_SDK 源码中的 stp23l_ws 里的两个功能包(ldlidar_driver 和 ros2_app)拷贝到 ROS2 的工作空间中，编译功能包。

```
colcon build
source install/local_setup.bash
```

```
wheeltec@wheeltec:~/lidar_ws$ colcon build
Starting >>> ldlidar
Finished <<< ldlidar [1.25s]

Summary: 1 package finished [2.44s]
wheeltec@wheeltec:~/lidar_ws$
```

图 1-2-10 编译功能包

把 STP23L 模块接入到主板中，修改串口设备权限，(以/dev/ttyUSB0 为例)，实际使用时，根据模块在系统中的实际挂载情况来设置，可以使用`ll /dev | grep ttyUSB`命令查看。:

```
sudo chmod 777 /dev/ttyUSB0
```

默认以/dev/ttyUSB 启动模块，如有变动可在以下文件中修改

/home/wheeltec/lidar_ws/src/ros2_app/src/ldlidar/launch/stp23l.launch.py

```
stp23l.launch.py
1#!/usr/bin/env python3
2from launch import LaunchDescription
3from launch_ros.actions import Node
4
5def generate_launch_description():
6    # LDROBOT LiDAR publisher node
7    ldlidar_node = Node(
8        package='ldlidar',
9        executable='stp23l_ros2node',
10       name='STP23L',
11       output='screen',
12       parameters=[
13           {'product_name': 'LDLiDAR_STP23L'},
14           {'topic_name': 'laser'},
15           {'port_name': '/dev/ttyUSB0'},
16           {'port_baudrate': 230400},
17           {'frame_id': 'base_laser'}
18       ]
19    )
20
21    # base_link to base_laser tf node
22    base_link_to_laser_tf_node = Node(
23        package='tf2_ros',
24        executable='static_transform_publisher',
25        name='base_link_to_base_laser_stp23l',
26        arguments=['0', '0', '0.18', '0', '0', '0', 'base_link', 'base_laser']
27    )
28
29    # Define LaunchDescription variable
30    ld = LaunchDescription()
31
32    ld.add_action(ldlidar_node)
33    ld.add_action(base_link_to_laser_tf_node)
34
35    return ld
36
```

图 1-2-11 stp23l.launch.py 文件内容

启动模块

```
ros2 launch ldlidar stp23l.launch.py
```

启动成功后，终端会不断打印距离信息。

```
wheeltec@wheeltec:~/lidar_ws$
wheeltec@wheeltec:~/lidar_ws$ ros2 launch ldldidar stp23l.launch.py
[INFO] [launch]: All log files can be found below /home/wheeltec/.ros/log/2023-05-11-10-52-01-769922-wheeltec-11122
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [stp23l_ros2node-1]: process started with pid [11126]
[INFO] [static_transform_publisher-2]: process started with pid [11128]
[stp23l_ros2node-1] [INFO] [1683773522.531613948] [STP23L]: [ldrobot] SDK Pack Version is v2.0.1
[stp23l_ros2node-1] [INFO] [1683773522.531912962] [STP23L]: [ldrobot] <product_name>: LDL IDAR_STP23L
[stp23l_ros2node-1] [INFO] [1683773522.531983015] [STP23L]: [ldrobot] <topic_name>: laser
[stp23l_ros2node-1] [INFO] [1683773522.532022912] [STP23L]: [ldrobot] <port_name>: /dev/ttyUSB0
[stp23l_ros2node-1] [INFO] [1683773522.532059996] [STP23L]: [ldrobot] <port_baudrate>: 230400
[stp23l_ros2node-1] [INFO] [1683773522.532096350] [STP23L]: [ldrobot] <frame_id>: base_laser
[stp23l_ros2node-1] [INFO] [1683773522.533306679] [STP23L]: open lidar serial device:/dev/ttyUSB0 is success.
[stp23l_ros2node-1] [INFO] [1683773522.544263386] [STP23L]: lidar device stop measure.
[stp23l_ros2node-1] [INFO] [1683773522.544467554] [STP23L]: get lidar version information.
[stp23l_ros2node-1] [INFO] [1683773522.544548076] [STP23L]: Wait recv lidar version information.
[stp23l_ros2node-1] [INFO] [1683773522.548282708] [STP23L]: Get lidar version information is success.
[stp23l_ros2node-1] [INFO] [1683773522.548392710] [STP23L]: lidar firmware version:v1.1.8
[stp23l_ros2node-1] [INFO] [1683773522.548446262] [STP23L]: lidar hardware version:v0.0.23
[stp23l_ros2node-1] [INFO] [1683773522.548479690] [STP23L]: lidar manufacture times:1287-231-7,3:42:12
[stp23l_ros2node-1] [INFO] [1683773522.548514743] [STP23L]: lidar mcu id number:813110383733534d50120536
[stp23l_ros2node-1] [INFO] [1683773522.548550003] [STP23L]: lidar sn code:#
[stp23l_ros2node-1] 0000000000
[stp23l_ros2node-1] [INFO] [1683773522.548587244] [STP23L]: lidar pitch angle:65535,65535,65535,65535
[stp23l_ros2node-1] [INFO] [1683773522.548626463] [STP23L]: lidar blind area, near:30,far:7500
[stp23l_ros2node-1] [INFO] [1683773522.548666734] [STP23L]: lidar frequency val:120
[stp23l_ros2node-1] [INFO] [1683773522.548708131] [STP23L]: Start measure.
[static_transform_publisher-2] [INFO] [1683773522.550100024] [base_link_to_base_laser_stp23l]: Spinning until killed publishing transform from 'base_link' to 'base_laser'
[stp23l_ros2node-1] [INFO] [1683773522.653024290] [STP23L]: get lidar frame Data, timestamp:29293637
[stp23l_ros2node-1] [INFO] [1683773522.653298200] [STP23L]: Measure data:
[stp23l_ros2node-1] [INFO] [1683773522.653417003] [STP23L]: distance:211,noise:7,peak:813576,confidence:100,intg:11298,reftof:340
[stp23l_ros2node-1] [INFO] [1683773522.653506796] [STP23L]: -----
[stp23l_ros2node-1] [INFO] [1683773522.653615079] [STP23L]: Measure data:
[stp23l_ros2node-1] [INFO] [1683773522.653728132] [STP23L]: distance:212,noise:2,peak:769487,confidence:100,intg:10720,reftof:341
[stp23l_ros2node-1] [INFO] [1683773522.653809821] [STP23L]: -----
[stp23l_ros2node-1] [INFO] [1683773522.653928085] [STP23L]: Measure data:
[stp23l_ros2node-1] [INFO] [1683773522.654029667] [STP23L]: distance:211,noise:7,peak:910887,confidence:100,intg:12659,reftof:338
[stp23l_ros2node-1] [INFO] [1683773522.654143211] [STP23L]: -----
[stp23l_ros2node-1] [INFO] [1683773522.654287483] [STP23L]: Measure data:
[stp23l_ros2node-1] [INFO] [1683773522.654398526] [STP23L]: distance:212,noise:8,peak:821035,confidence:100,intg:11415,reftof:339
```

图 1-2-12 终端打印内容

查看话题:

```
wheeltec@wheeltec:~$ ros2 topic list
/laser
/parameter_events
/rosout
/tf_static
wheeltec@wheeltec:~$ ros2 topic echo /laser
header:
  stamp:
    sec: 1683773559
    nanosec: 565643060
    frame_id: base_laser
  radiation_type: 1
  field_of_view: 0.10000000149011612
  min_range: 0.0
  max_range: 12.0
  range: 0.21299999952316284
---
header:
  stamp:
    sec: 1683773559
    nanosec: 569142950
    frame_id: base_laser
  radiation_type: 1
  field_of_view: 0.10000000149011612
  min_range: 0.0
  max_range: 12.0
  range: 0.21299999952316284
---
header:
  stamp:
    sec: 1683773559
    nanosec: 569677384
    frame_id: base_laser
  radiation_type: 1
  field_of_view: 0.10000000149011612
  min_range: 0.0
  max_range: 12.0
  range: 0.21199999749660492
---
```

图 1-2-13 /laser 话题数据

1.3 STP-23L 模块的 Win10 上位机使用

STP-23L 模块还可使用 Win10 上位机查看相应的点云距离强度等信息,使用前需按照 1.2 节内容将 TTL 转 USB 模块与 STP-23L 模块相连并将 USB 插入 Win10 电脑中:

① 驱动安装

首先需要确保 Win10 电脑已安装最新的 CP2102 驱动,在提供资料包内找到 CP2102usb 驱动如图 1-3-1:

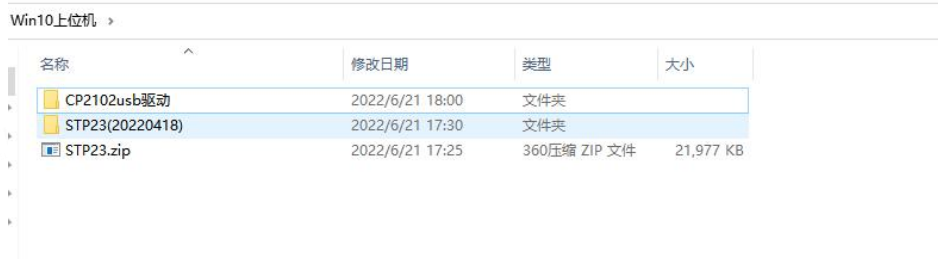


图 1-3-1 CP2102usb 驱动

双击打开后根据电脑操作系统位数相应打开 exe, 这里仅举例常用的 64 位操作系统, 打开 CP210xVCPIInstaller_x64.exe 后显示如图 1-3-2:



图 1-3-2 CP210xVCPIInstaller_x64.exe

点击下一步并接受协议安装完成后界面如图 1-3-3:



图 1-3-3 CP2102usb 驱动安装成功

② 识别 CP2102 的 COM 号

完成驱动安装后，右击此电脑——>管理可打开计算机管理，单击设备管理器找到端口并扫描检测硬件改动可看到电脑已识别 CP2102 且为 COM9:

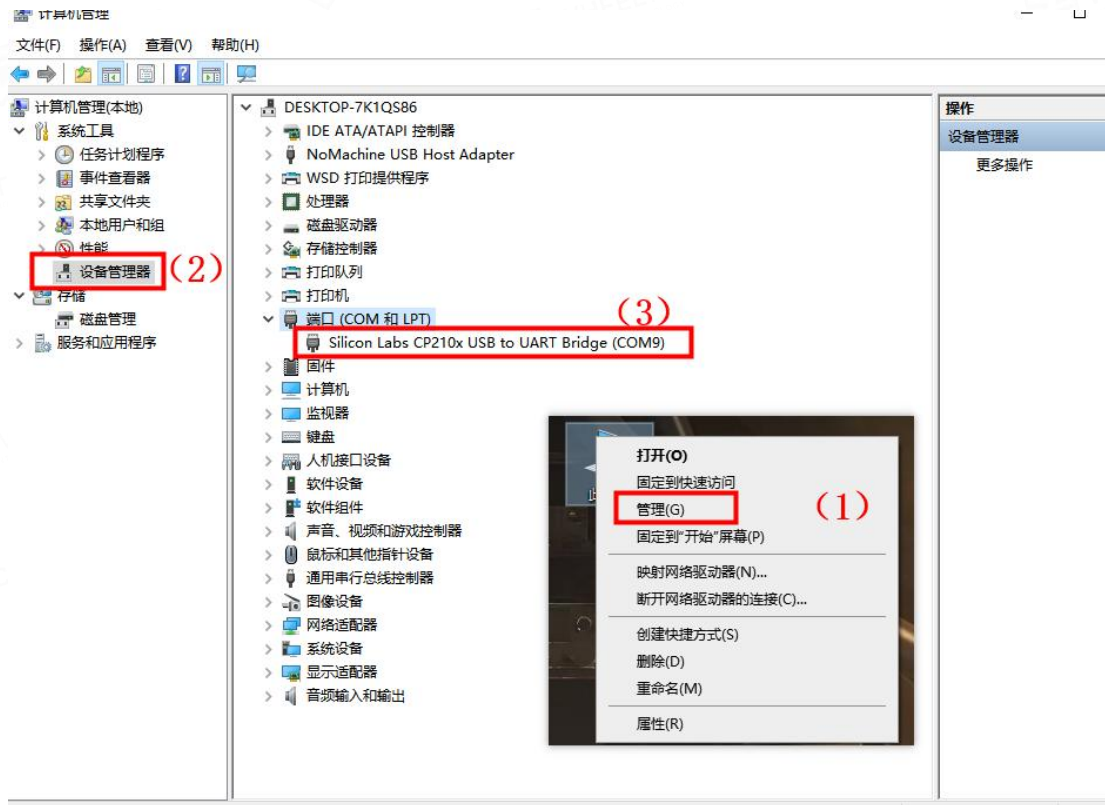


图 1-3-4 识别 CP2102 的 COM 号

③ 使用 STP-23LWin10 上位机

将资料包 Win10 上位机中的[STP-23L_点激光上位机]解压并打开相应文件夹如图 1-3-5:

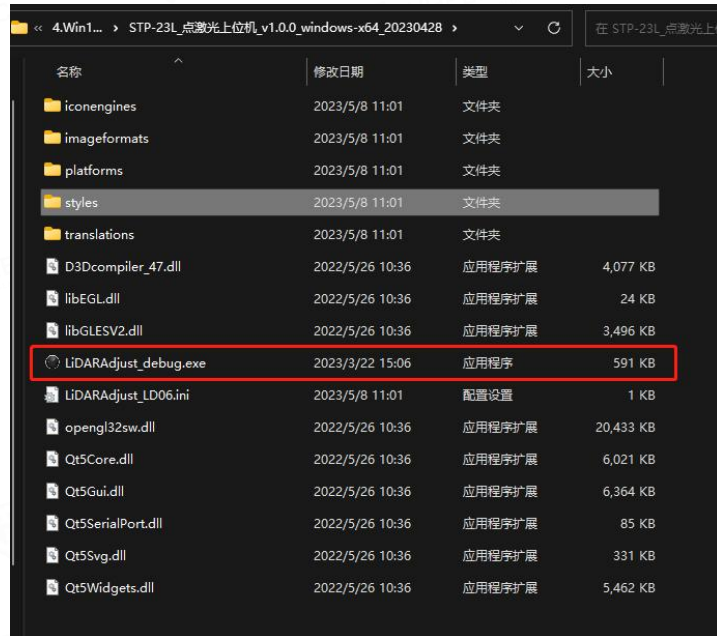


图 1-3-5 STP-23L_点激光上位机

找到 LiDARAdjust_debug.exe 双击打开后点击设置打开串口，在弹窗中选择 STP-23L 的 COM 号后，点击应用则成功连接上位机。

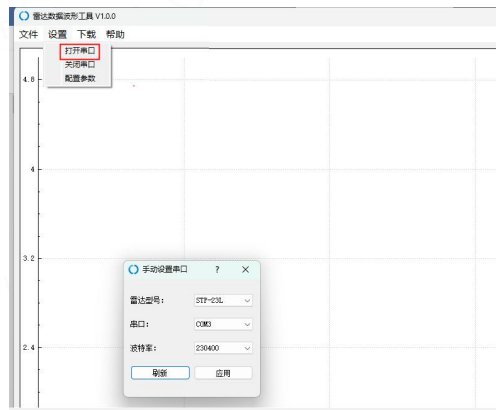


图 1-3-6 STP-23L_点激光上位机

可在上位机中观察到相应的测试距离及强度信息等如图 1-3-7，上位机默认显示波形图，x 轴为时间，y 轴为测距数值。在上位机中可以保存数据（点击文件选项即可操作），也可以在右侧读取模块的测距距离等系统信息。

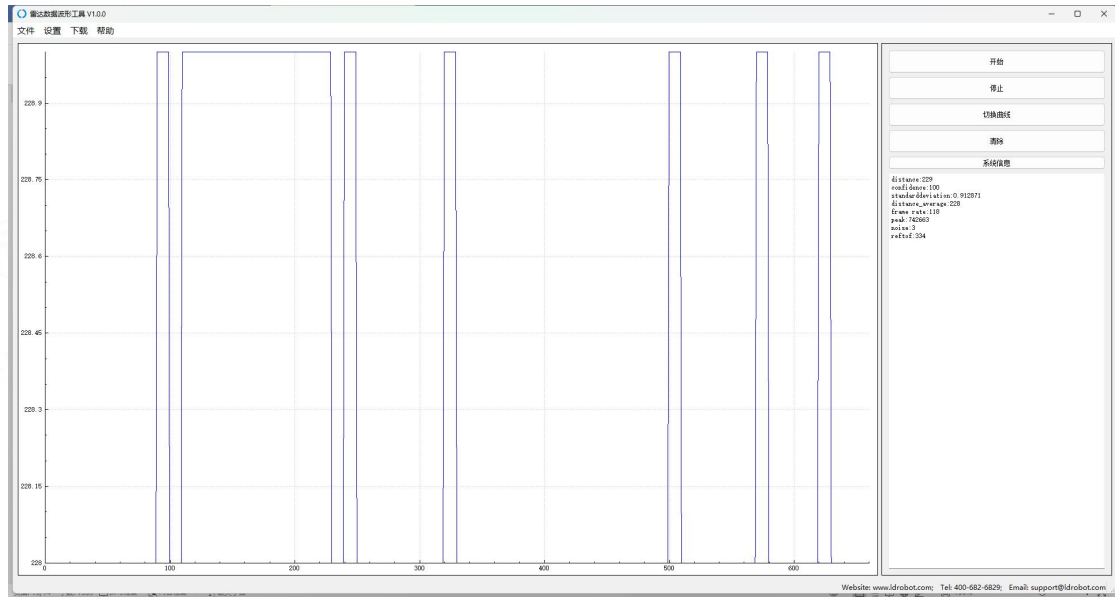


图 1-3-7 上位机视图