引用本文需添加作者信息：

# 项目简介

本项目使用 LMS511 2D单线激光雷达和转台，

结合python通信、qt界面与vtk图形化，

实现激光雷达旋转扫描读取三维点云数据并显示。

# 目录

按住ctrl以快速移动

**快速使用**

**激光雷达简介**

**点云数据简介**

**设备简介**

**程序开发流程**

**软件环境**

**通信流程**

# 快速使用

1. 单次\连续读取:

搭建软件环境，见**软件环境**

连接网线, 修改雷达ip, 见**通信流程**的前10行, **sopas登录**的修改ip地址

使用pycharm创建工程, 见**软件环境**，添加源文件，运行

先connect. 然后read once 或者 read always

1. 旋转扫描读取三维点云数据并显示:

搭建硬件,见 **设备组合,**

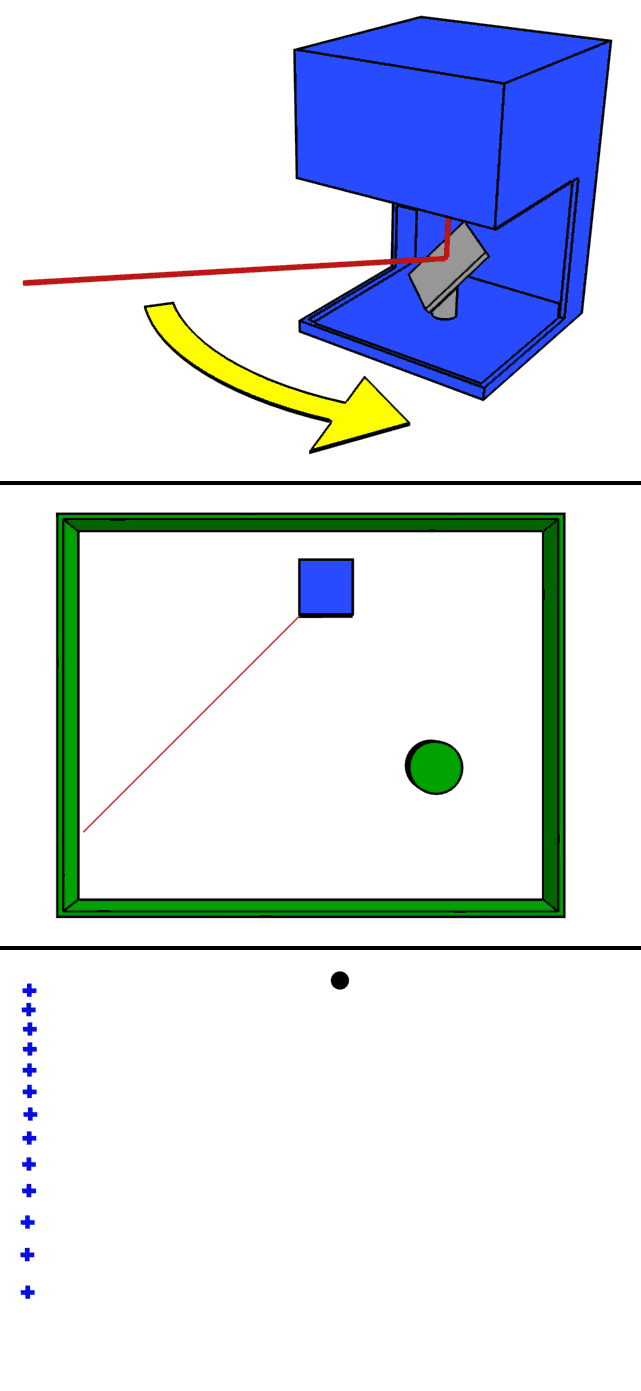
运行程序, connect

遥控设置转台周期为20s , 向左旋转, 按下旋转

点击 round scan

# 激光雷达简介

激光雷达通过从物体表面接收反射光线来获取信息。激光雷达属于主动光电探测仪器，工作时，激光器对准感兴趣的目标，发射激光脉冲，激光通过介质，击中物体，然后反射再次通过介质，最后被激光探测器接收。激光探测器或阵列根据返回的激光得到信息。



<https://en.wikipedia.org/wiki/Lidar>

常见激光雷达测距的方式有两种：三角式测距和飞行时间（TOF）测距。

三角测距中，激光发射器与接收器CCD（电荷耦合器件）摆放在不同的位置，不同距离的物体所返回的激光根据三角法则落在CCD不同的位置上，根据三角公式计算就能得出距离。

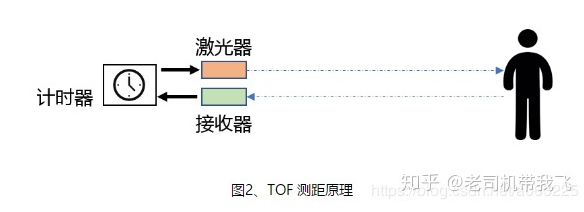
此种方法测量距离进，远距离分辨率不高，采样速率不高。



<https://zhuanlan.zhihu.com/p/163109172>

飞行时间测距中，激光发射器与接收器可以放在相近的位置上。激光器配有计时设备，激光发射时开始计时，接收时计时结束，通过计算时间差得到距离。

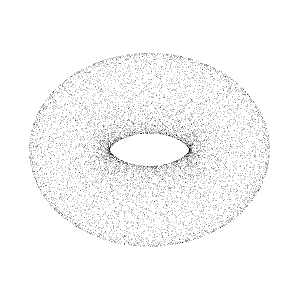
此种方式的困难在于时间精确度，在近距离测量对设备精度要求极高。TOF测距主要应用在远距离测距中，同时还有较高的扫描频率。



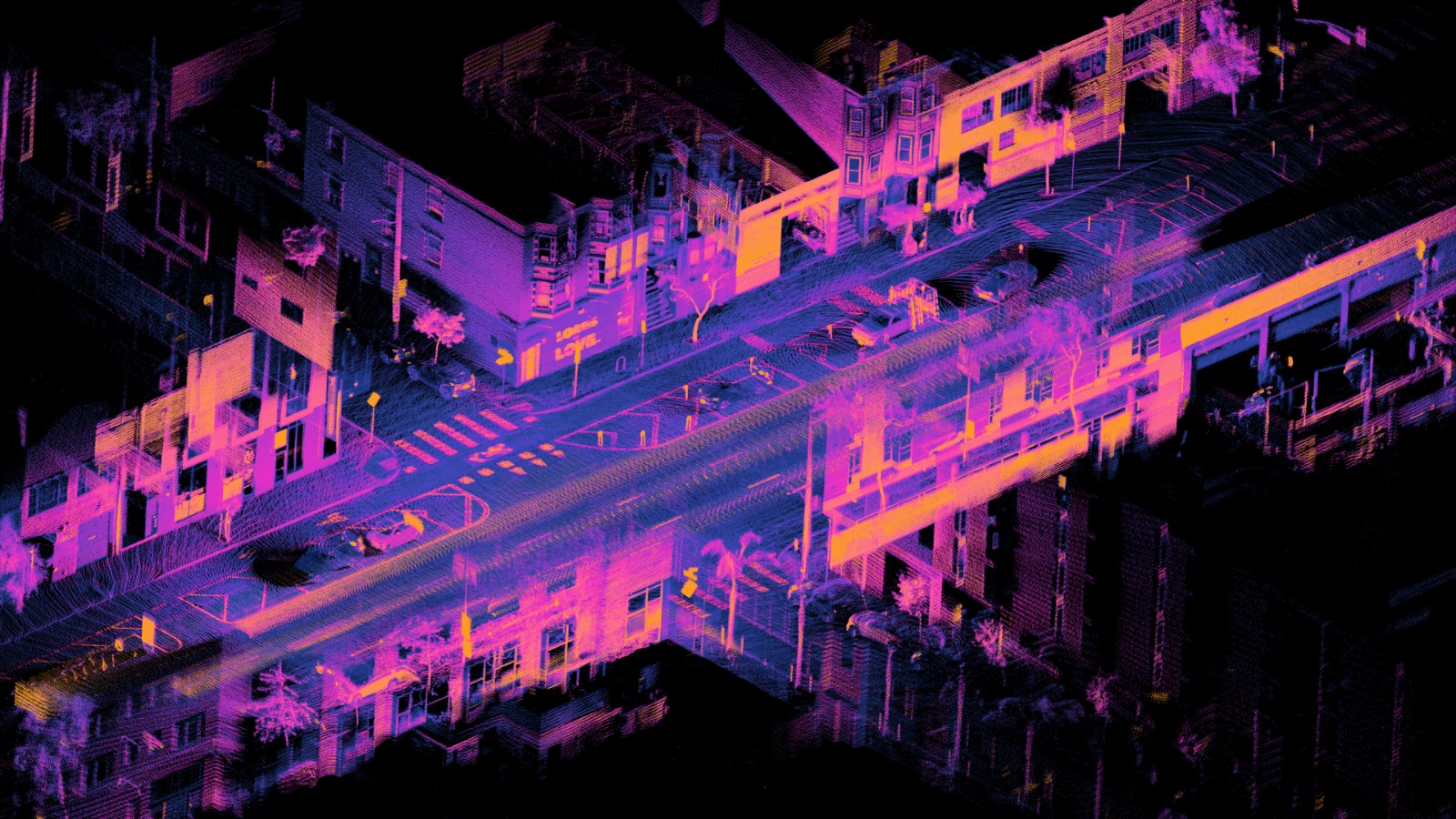
# 点云数据简介

激光雷达所能提供的数据是点云。激光雷达通过测量距离和角度得到兴趣点的坐标。

点云是空间中的一组数据点。这些点代表一个三维形状或物体。每个点都有它的X, Y和Z坐标集合。



<https://en.wikipedia.org/wiki/Point_cloud>

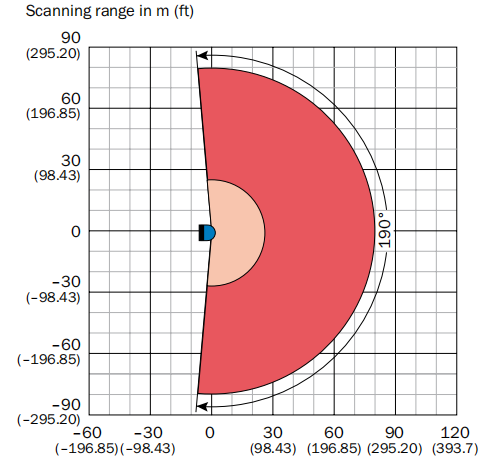


# 设备简介

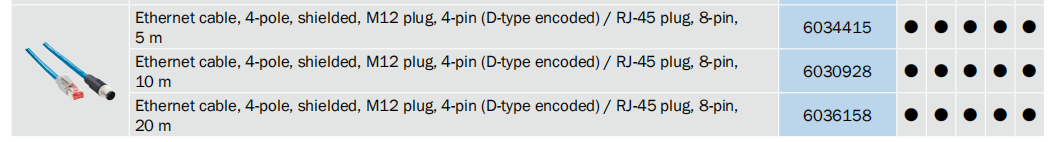
## 激光雷达

LMS511是一种2D激光雷达，扫描角度大于半周，测量距离可达80m，扫描频率可选为25hz-100hz。

LMS511 下简称雷达。



LMS511支持以太网通信，使用有线连接LMS511和PC：



## 转台：

淘宝常见的转台。

有合适的承重，平面大小，转速。

配有一个红外遥控器，可以调节旋转开始暂停，旋转方向，旋转速度。

## 

## 设备组合：

将雷达向左倾倒停放在转台平面上。此时雷达扫描垂直方向半周角度的距离。

转台旋转，雷达每运动一个角度，扫描一次，如此反复，扫描一周360°数据，为3D点云数据。



左倾倒是因为程序坐标这样设计的

# 程序开发流程：

# 软件环境

教程：使用conda创建环境，组合python3.6 vtk pyqt5 numpuy pclpy

* Conda简介

Conda 是一个开源的**软件包管理系统和环境管理系统**，用于安装多个版本的软件包及其依赖关系，并在它们之间轻松切换。 Conda 是为 Python 程序创建的，适用于 Linux，OS X 和Windows，也可以打包和分发其他软件。

* Download MiniConda：

<https://docs.conda.io/en/latest/miniconda.html>

* 创建环境

使用conda终端：开始菜单找到 anaconda prompt 运行，输入以下命令

conda使用指南<https://zhuanlan.zhihu.com/p/44398592>

添加清华大学镜像下载源：

conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/

conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/

conda config --set show\_channel\_urls yes

创建环境： conda create --name py36 python=3.6

激活环境： conda activate py36 每行标志会从（base）变成（py36）

安装以下包 conda install numpy

pip install pclpy

pip install pyqt5

conda install vtk=8.1

* **在pycharm 下使用conda环境**

<https://zhuanlan.zhihu.com/p/39542494>

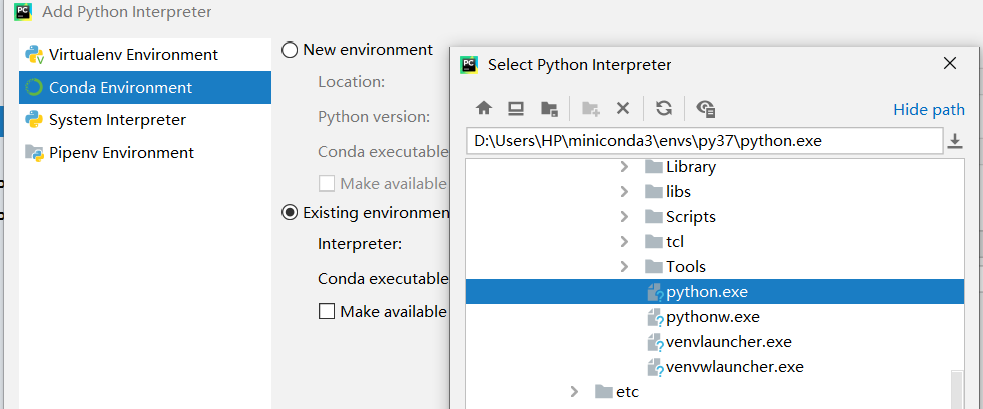
<https://www.jianshu.com/p/f5f07ac800d8>

设置-工程设置-工程解释器：

    添加python 解释器：

        从conda环境，已有环境（py37），添加解释器：注：程序最终使用py36

            \*\*conda\envs\env\_name\python.exe



Q：什么是conda环境？

anaconda是python的一种？安装anaconda即拥有python。

conda是python的功能组件？管理版本，创建环境

某某\*\*环境包括，安装的特定版本的python（即含解释器），安装的包（pyqt5）

anaconda主目录下存在python.exe

子目录envs下包含conda创建的环境文件夹，每个环境都有对应python.exe

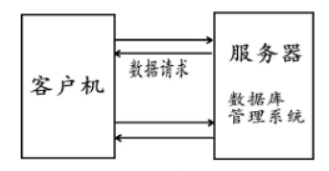
# 通信流程：

### 连接，socket流程中的connect

雷达接通电源启动，等待绿灯亮起，即可通讯读取。

用 以太网线 连接雷达和PC。

在socket通信中，PC为Client，LMS511为Server。

##在这里画一幅client server的示意图

为雷达配置ip。用sopas登录和修改设备的ip，见sopas登录。

在PC端，用socket通信，connect函数连接雷达。

**（example in python ::）**

import socket

s = socket. socket ( socket. AF\_INET, socket. SOCK\_STREAM )

ip = "169.254.104.114"

port = 2111

addr = (ip, port)

s. connect(addr)

**（C++：：）**

#include<WinSock2.h>

#pragma comment (lib ,"ws2\_32.lib")

#include<WS2tcpip.h>

const std::string myserver = "169.254.104.114";

const short myport = 2111;

client = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

SOCKADDR\_IN seraddr;

ZeroMemory(&seraddr, sizeof(seraddr));

seraddr.sin\_family = AF\_INET;

inet\_pton(AF\_INET, myserver.c\_str(), &seraddr.sin\_addr.s\_addr);

seraddr.sin\_port = htons(myport);

int szsa = sizeof seraddr;

int iresult = connect(client, (sockaddr\*)&seraddr, szsa);

### LMS511的通信协议，Telegram报文介绍：

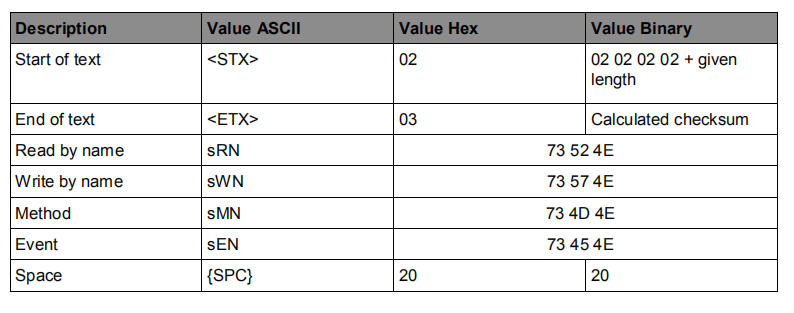
ASCII电报是一种通信的格式，使用ASCII的原因是它更好理解。

1.电报的框架在开始时是STX (\x02)，在每个电报的末尾是ETX (\x03)。

2.命令用ASCII码写的 (sRN)，后面是该文档中定义的参数。

3.参数可以以十六进制或十进制格式传输，但以十进制格式传输，它们需要一个符号（例如：扫描频率25Hz：09C4h/2500d）

一些常见命令的解释：



本文显示命令示例

读取一次： sRN LMDscandata

发送命令示例：

（python）

send\_data = b'\x02sRN LMDscandata\x03' \\要注意python显示的字符串是unicode

s.send(send\_data)

（C++）

string cmd = "\x02sMN SetAccessMode 03 F4724744\x03"; \\string是类

send(client, cmd.c\_str(), (int)cmd.length(), 0)

### 参数化扫描工作流程

1.登录: sMN SetAccessMode

2.设置扫描频率和角度分辨率: sMN mLMPsetscancfg

3. Configure scan data content: sWN LMDscandatacfg

4. Configure scan data output: sWN LMPoutputRange

5. 永久存储设置: sMN mEEwriteall

6. 登出: sMN Run

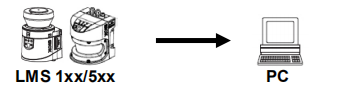
7. 扫描: sRN LMDscandata / sEN LMDscandata

### 直接扫描工作流程

扫描: sRN LMDscandata

### 读取，LMS511的通信协议，socket中send 和 recv

先发送send 

后接收recv

### 单次扫描流程

发送 sRN LMDscandata

ascii : <STX>sRN(SPC}LMDscandata<ETX>

接收 sRA LMDscandata+数据

数据格式见Telegram

接收数据样式：

\x02sRA LMDscandata 0 1 119883A 0 0 2796 2801 F87F5C00 F87FA401 0 0 3F 0 0 1388 168 0 1 DIST1 40000000 00000000 0 1388 F1 363 364 35B 35A 35A 342 33F 33E 341 344 341 353 35A 35F 35D 37A 38E 38E 39A 3A4 3A6 3AE 3B8 3BF 3C3 3C2 3C5 3CE 3D6 3DA 3E0 3E3 3E0 3E5 3E9 3E6 3EB 3EF 3F4 410 4E5 4FB 501 505 508 50B 511 518 51E 4E0 4D5 4E2 51B 4E4 4D0 4F9 4DE 4BC 4B2 4B4 49D 48E 484 481 484 487 491 495 49E 4A9 4AD 4B1 4BB 4C2 4CD 4D3 4DB 4E9 4ED 4F7 4FD 508 50E 51D 525 531 53C 544 551 55C 567 570 57D 58D 596 5A5 5B1 5C0 5D0 5E0 5EE 5FD 60E 622 62F 63F 64D 664 676 689 6A0 6B6 6C9 6DD 6F8 70C 725 73B 756 76E 789 7A8 7C4 7E0 804 823 844 865 88D 8AF 8D2 8FE 926 956 982 9B0 9E3 A17 A50 A84 ABD B02 B41 B85 BD3 C1D C6F CC2 D1A D76 DE3 E5D EDC F5E FF2 1095 1146 11FD 12D3 13C8 14BB 15CB 16EC 185A 1941 194F 1948 1945 194C 1958 1957 1962 1966 0 0 7B96 7B98 7BA9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0\x03

**通信示例代码:** （python）

send\_data = b'\x02sRN LMDscandata\x03' \\要注意python显示的字符串是unicode

self.s.send(send\_data)

recv\_data = self.s.recv(bufsiz) \\ bufsize是8192

data = recv\_data.decode() \\转化成python unicode字符串便于分析

数据处理函数, 将接收的数据转化成点坐标

**def** data\_process(data): \\函数原型

data = data.split(**' '**) \\按空格分割成list

if not data[0] == "\x02sRA": \\检测开头

return []

if data[-1].endswith('\x03'): \\检测结尾,有时候结尾会和最后一组数据混叠

data[-1] = data[-1][:-1]

# 回波系数，乘数. 表示距离应该乘以的倍数

if data[21] == '40000000':  
 factors = 2  
elif data[21] == '3F800000':  
 factors = 1  
else:  
 return []

起始角度

startangle = int(data[23], 16) / 10000

角度分辨率

anglestep = int(data[24], 16) / 10000

数据总量

datanum = int(data[25], 16)

数据类型：points是一个list, 包含元素为point. point是一个list, 包含元素为x,y,z坐标值.

例如：[ [ 1.0001, 2.0001, 3.0], [ ]]

points=[]

for i in range(datanum):

point=[]

point.append(int(data[26 + i], 16) \* factors \* math.cos((startangle + i \* anglestep) / 180 \* math.pi) / 1000)

point.append(int(data[26 + i], 16) \* factors \* math.sin((startangle + i \* anglestep) / 180 \* math.pi) / 1000)

point.append(0)

points.append(point)

return points

数据为points

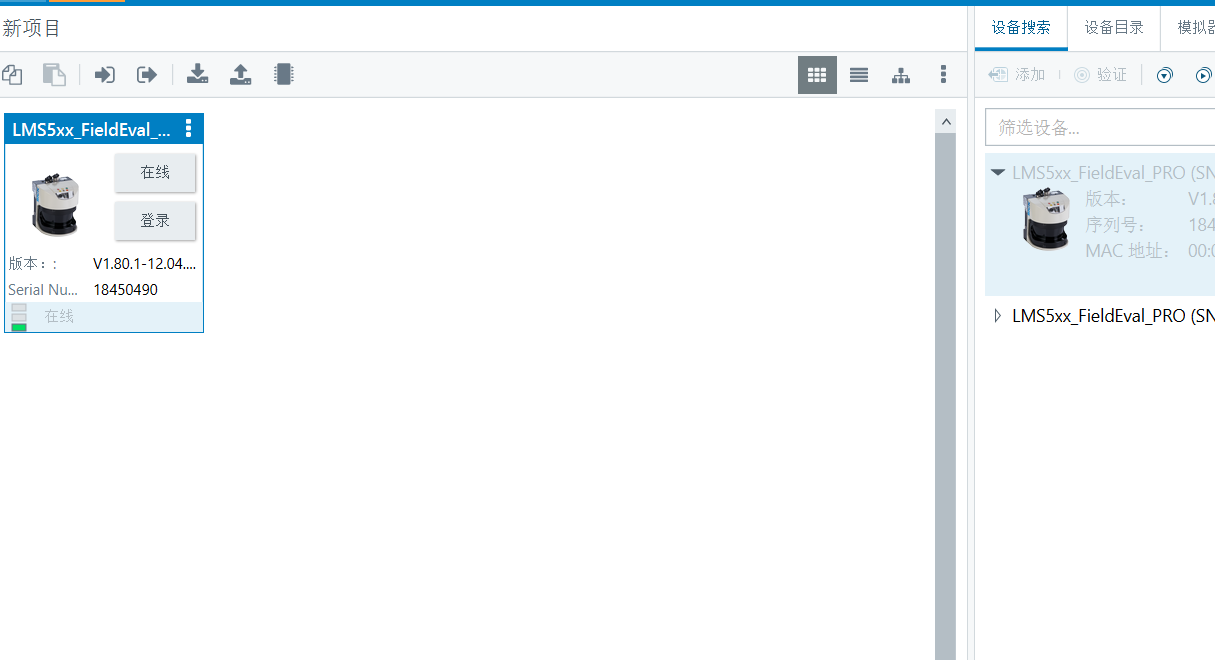
元素为point，是三元list

输出points

## sopas登录

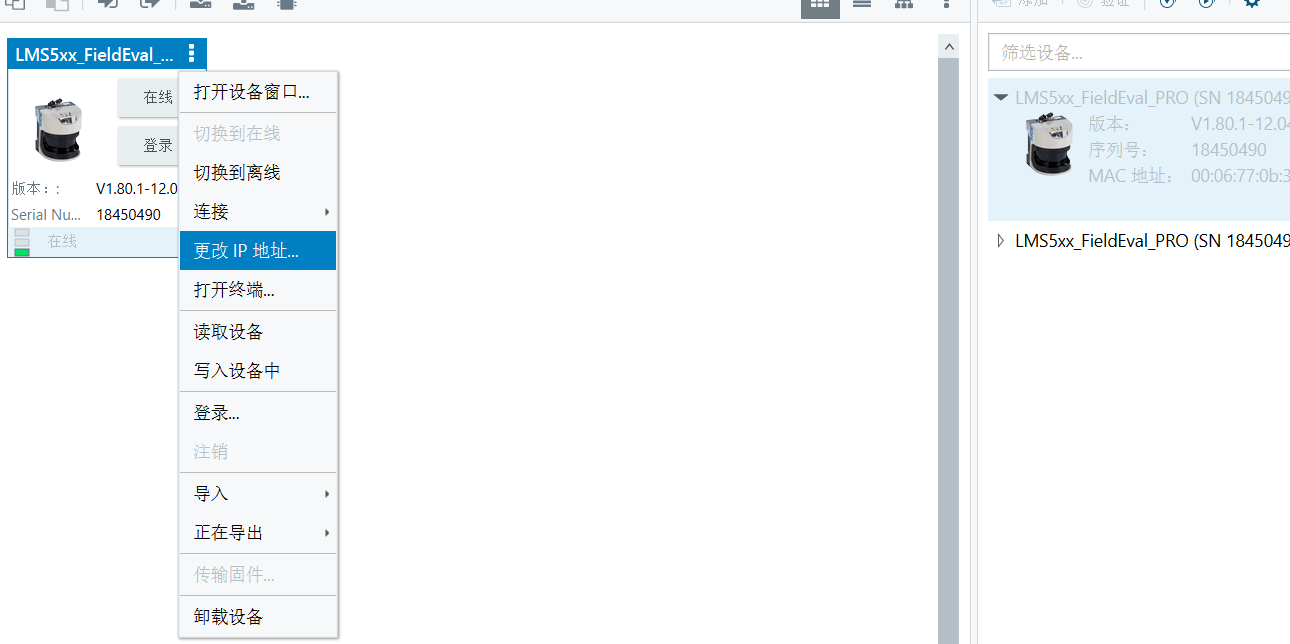
连接雷达, 打开sopas软件, 自动搜索, 搜索到设备.

点击添加雷达, 选择端口号port 为2111的,

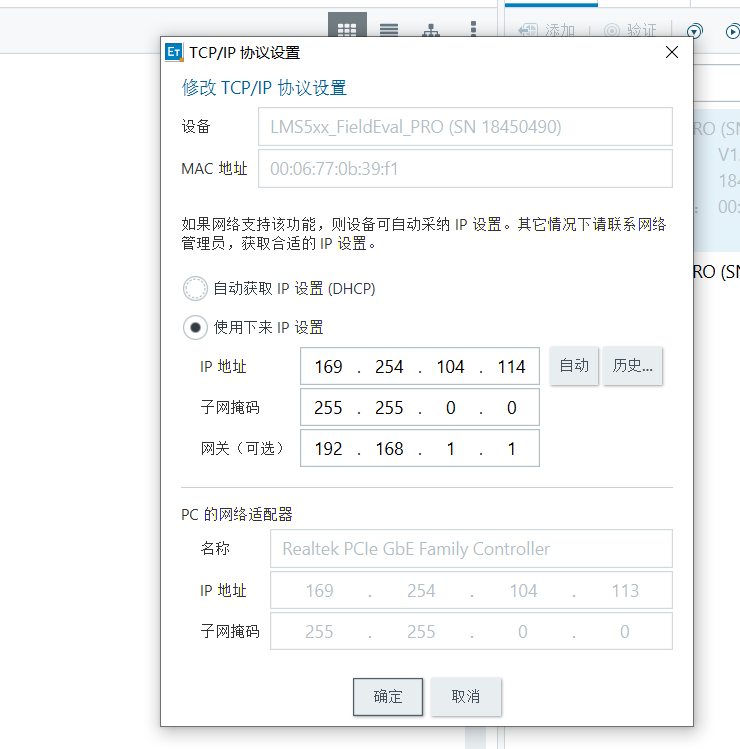


### 修改IP地址教程:

添加设备后, 点击左侧设备图标, 选择**更多—更改ip地址**



设置为 169.254.104.114 , 此为本程序指定ip地址

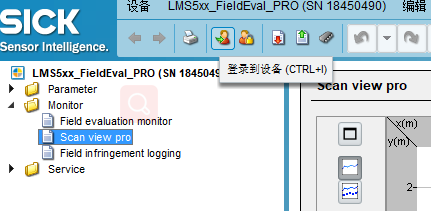


OK继续

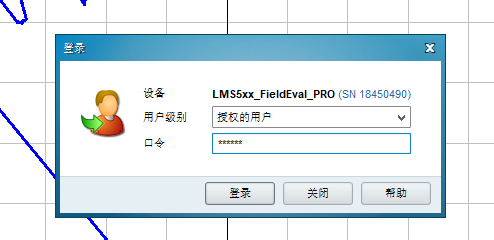
双击左侧设备图标, 打开此雷达通信面板



点击登录到设备



选择授权的用户, 密码为 client



打开此页面, 实时显示雷达测量的结果

