重 庆 交 通 大 学

期末大作业

课程名称 大数据开发语言

学 院 2020年级 计算机类 专业 2004 班

学 生 姓 名 李腾 学 号 632005010111

任 课 教 师 蒋震

开 课 时 间 2021 至 2022 学年第 1 学期

# 系统概述

# socket程序设计简介

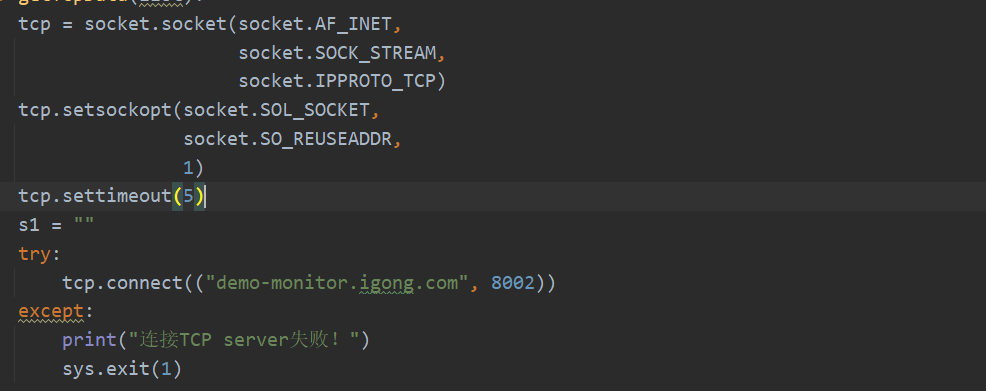
Socket又称"套接字"，应用程序通常通过"套接字"向网络发出请求或者应答网络请求，使主机间或者一台计算机上的进程间可以通讯。

要使用Socket，需要先导入Socket模块，并使用socket()函数来创建套接字，在socket()函数中可以设置是TCP/UDP。

由于该作业中只需要设计客户端Socket，去连接远程服务器来获取数据，故下面描述两种TCP/UDP客户端设计的简介。

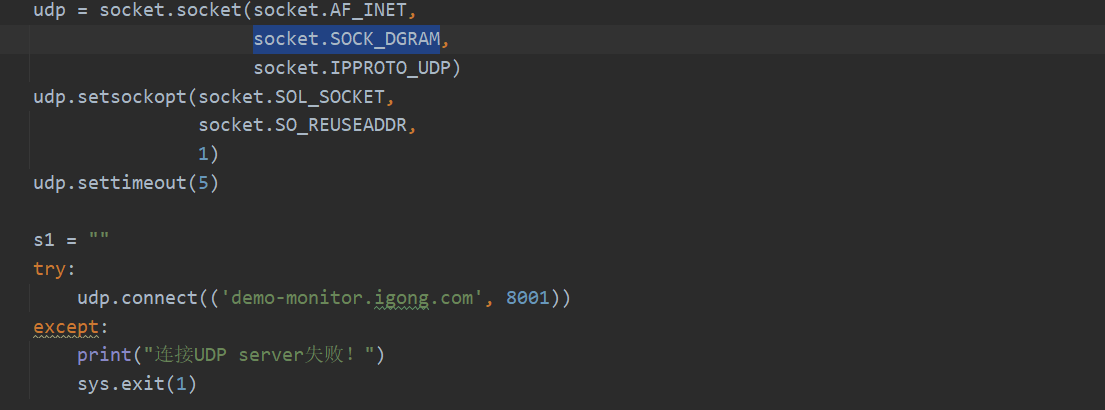
* TCP

首先给socket()函数创建tcp-socket对象，其中连接TCP的参数为socket.SOCK\_STREAM，然后设置给定套接字选项的值，并且还可以设置连接超时时间，然后连接远程服务器的ip和port，最后不使用了之后还要进行关闭。



* UDP

Udp-socket对象和tcp的创建方式类似，只不过在参数中socket.SOCK\_STREAM更换为socket.SOCK\_DGRAM。其余操作和TCP类似。



# ModBus协议介绍

Modbus是一种单主站的主/从通信模式。Modbus网络上只能有一个主站存在，主站在 Modbus网络上没有地址，从站的地址范围为 0 - 247，其中 0 为广播地址，从站的实际地址范围为 1 - 247。 Modbus通信标准协议可以通过各种传输方式传播，如 RS232C、RS485、光纤、无线电等。

Modbus具有两种串行传输模式，ASCII 和 RTU。它们定义了数据如何打包、解码的不同方式。支持 Modbus 协议的设备一般都支持 RTU 格式。通信双方必须同时支持上述模式中的一种。

其中modbus协议的发送采集指令格式如下

如下表所示

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 地址  1byte | 指令码  1byte | 起始寄存器  2byte | 寄存器个数  2byte | CRC16  2byte |
| 01～FF | 03 | 高位在前 | 高位在前 |  |

而接收数据的格式如下表所示

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 地址  1byte | 指令码  1byte | 数据区长度  2byte | 数据区  n byte | CRC16  2byte |
| 01～FF | 03 | 高位在前 | 高位在前 |  |

在这里需要注意，发送的字符都必须是以十六进制的数字进行表示！

下面以UDP传输中的温湿度传感器为例说明modbus协议。

发送数据：由于发送采集指令需要地址、指令码、起始寄存器与CRC16校验码，且都需要用十六进制进行表示，故需要将温湿度传感器的地址，在本题目中为1，需要将数字1进行打包，再与指令码03与温湿度传感器的寄存器的地址相结合并且最后与生成的CRC16校验码合并并且表示为十六进制，即为modbus的发送数据格式。

接收数据：接收数据与发送数据的格式几乎相同，不过需要进行解包，获取数据区寄存器的地址。

# 系统界面设计

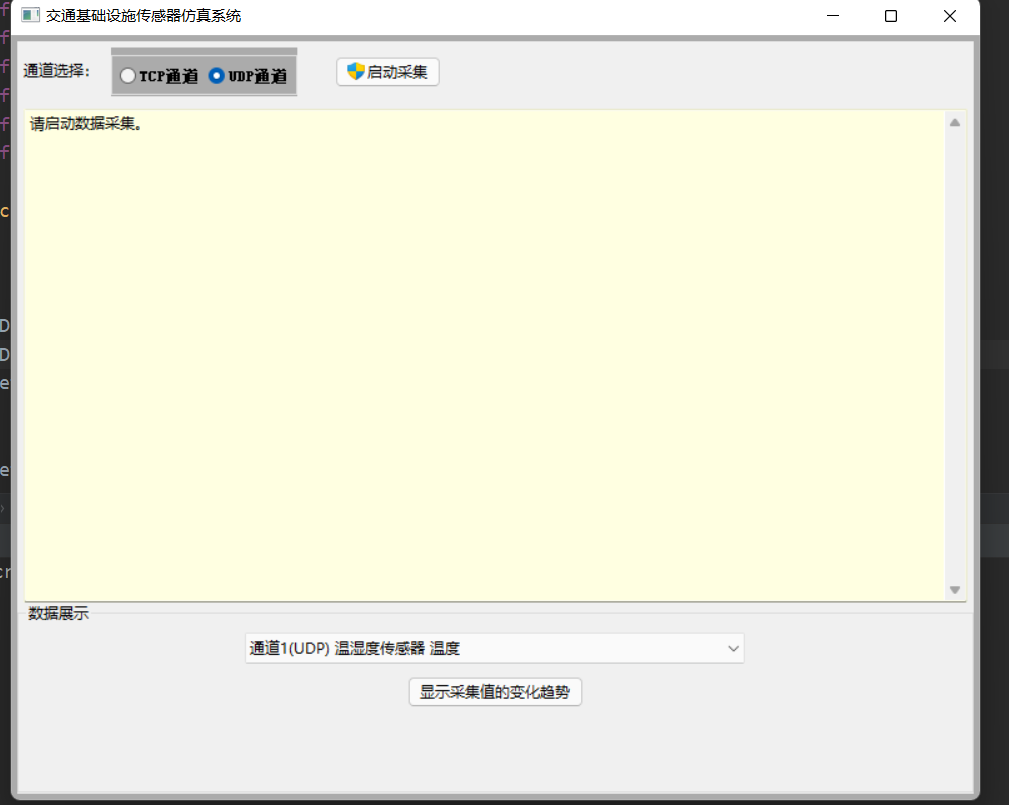
首先具有TCP和UDP通道的选择，并且在下面有显示数据区域。在通道的选择具有启动采集按钮，当进行点击启动采集后按钮会变成停止采集，即停止采集数据。并且在数据显示区域的下方具有数据展示选择，通过下拉框选择进行展示不同的采集的数据，即就是将采集的数据以图形化的形式展现出来，当未采集数据时而选择数据可视化不会显示任何结果，只会提示文字信息，且只有在关闭采集信息后才能图形化显示采集的数据。

# 系统实现思路

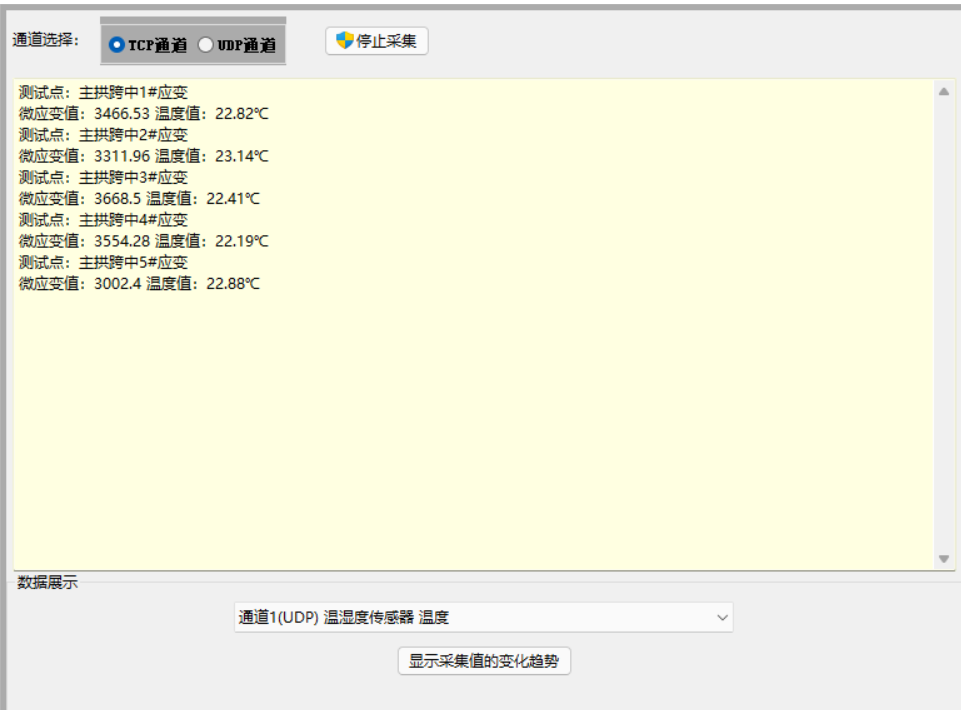
首先通过TCP或者UDP通道进行采集不同的数据，其中的数据采集需要通过modbus协议进行发送与接收，然后点击启动采集后通过定时器会每隔１秒钟进行采集并且展示在界面上，然后点击停止采集即可进行停止采集。当停止采集后通过matplotlib可以选择可视化进行展示。

# 系统界面

首次进去界面



启动采集后界面



# 主要事件设计

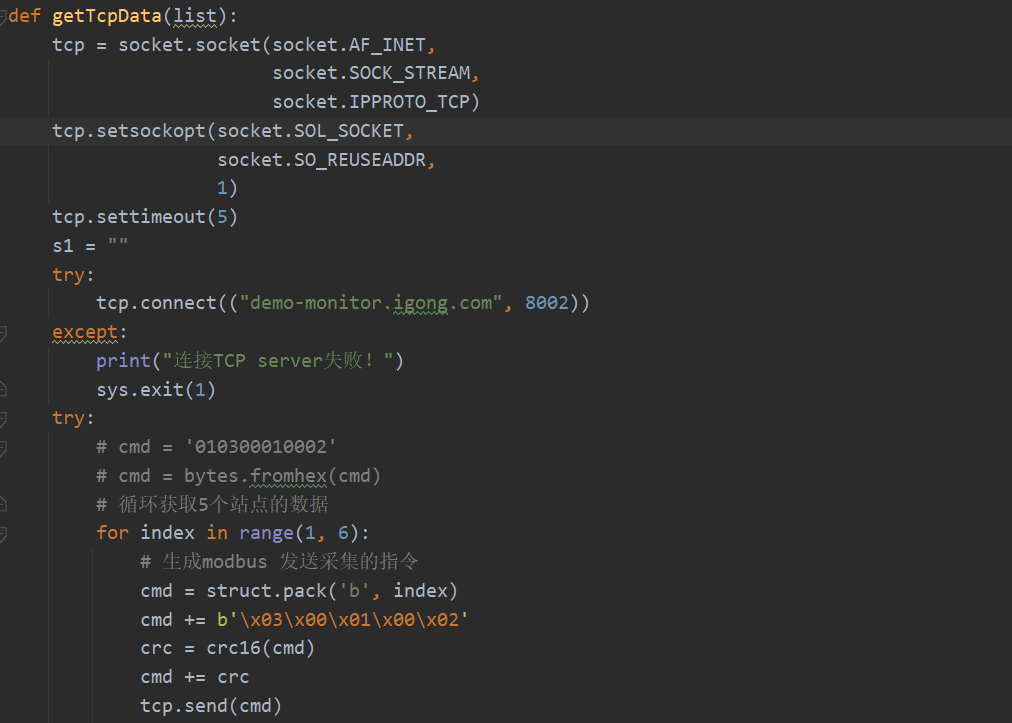
定时器事件的设计：当未点击采集数据按钮时，会在界面上显示出请启动数据采集。当选择TCP通道或者UDP通道后再点击数据采集按钮后会判断是否采集到数据，若未采集到数据则不会在界面上显示任何内容，采集到后每隔1秒钟进行显示。

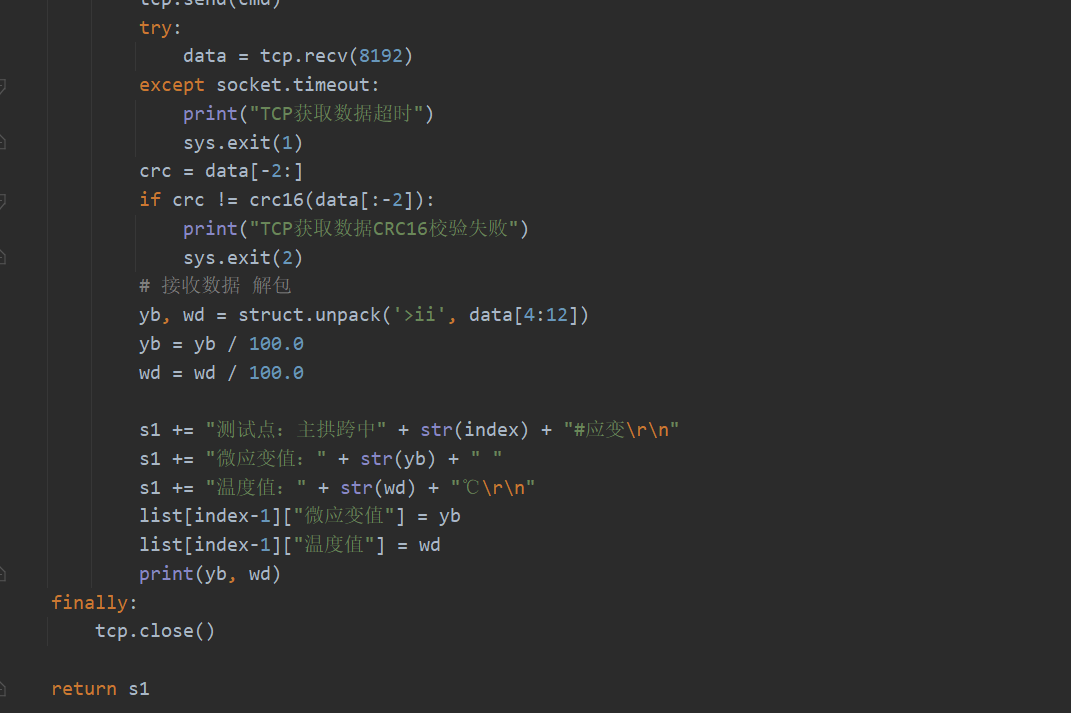
显示图形化事件的设计：点击按钮后首先判断是否关闭采集数据按钮了，只有当关闭采集数据按钮后，才可进行图形化显示。通过选择不同的传感器和相应的测点名称进行不同的展示。

# 代码设计

# 主要代码

通过TCP获取数据





通过UDP获取数据





图形化展示



# 指令及数据解析

UDP进行modbus协议发送与接收



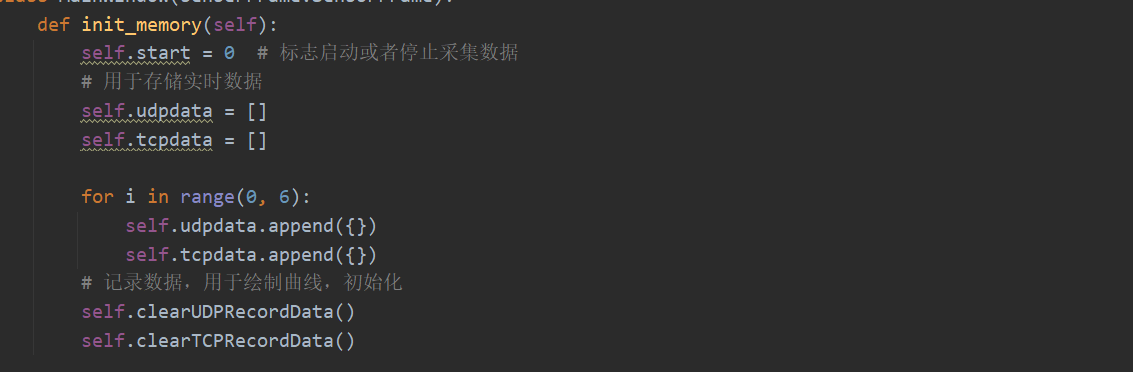


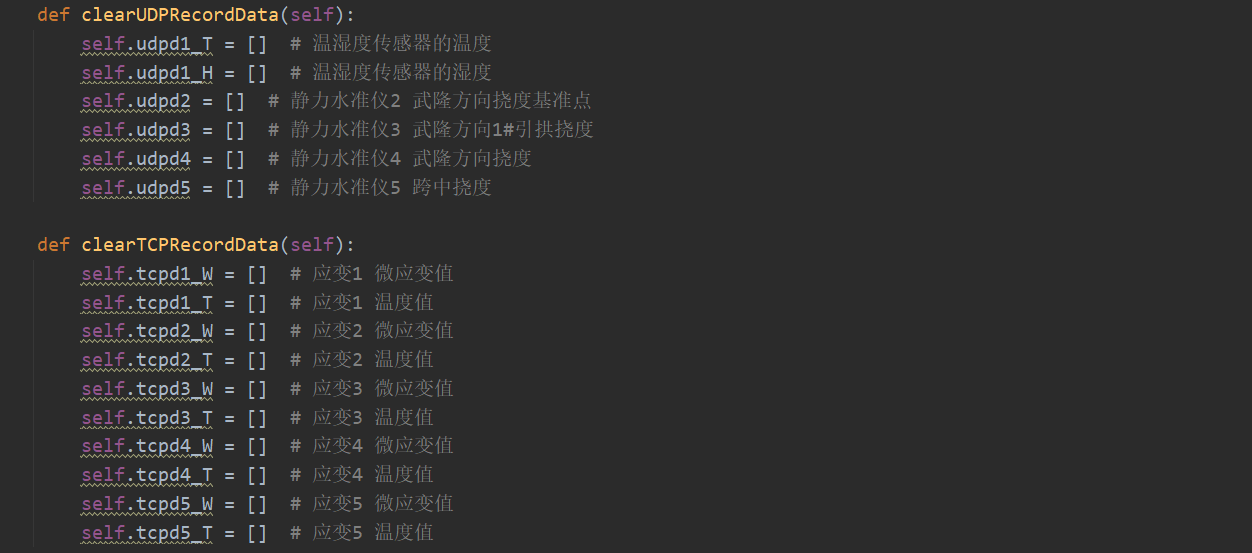
TCP进行modbus协议发送与接收

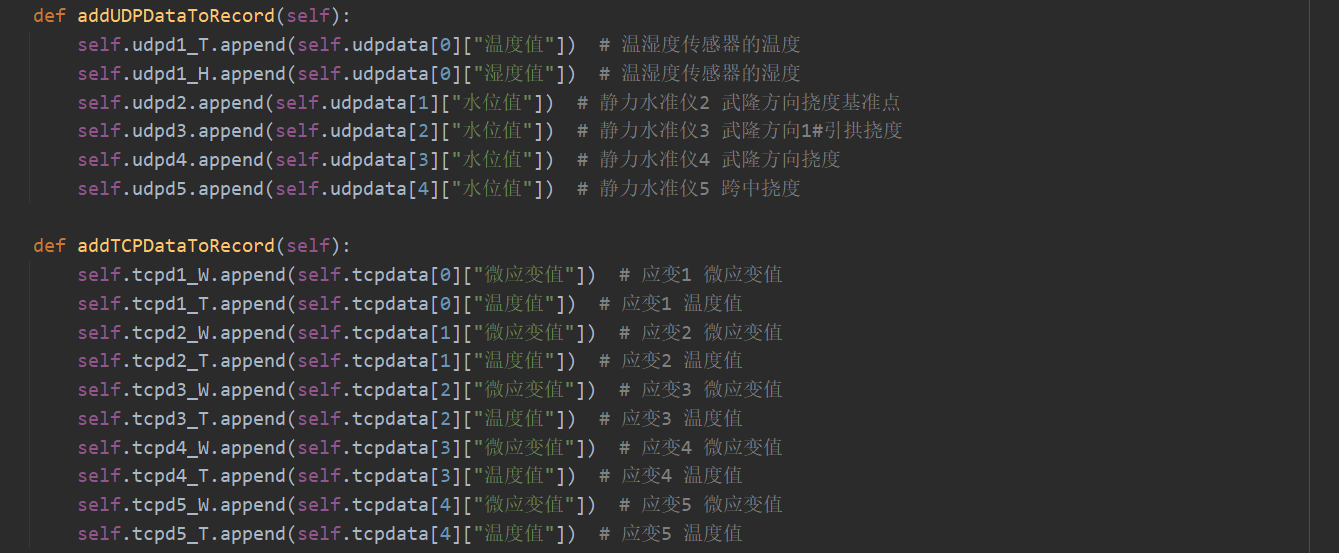


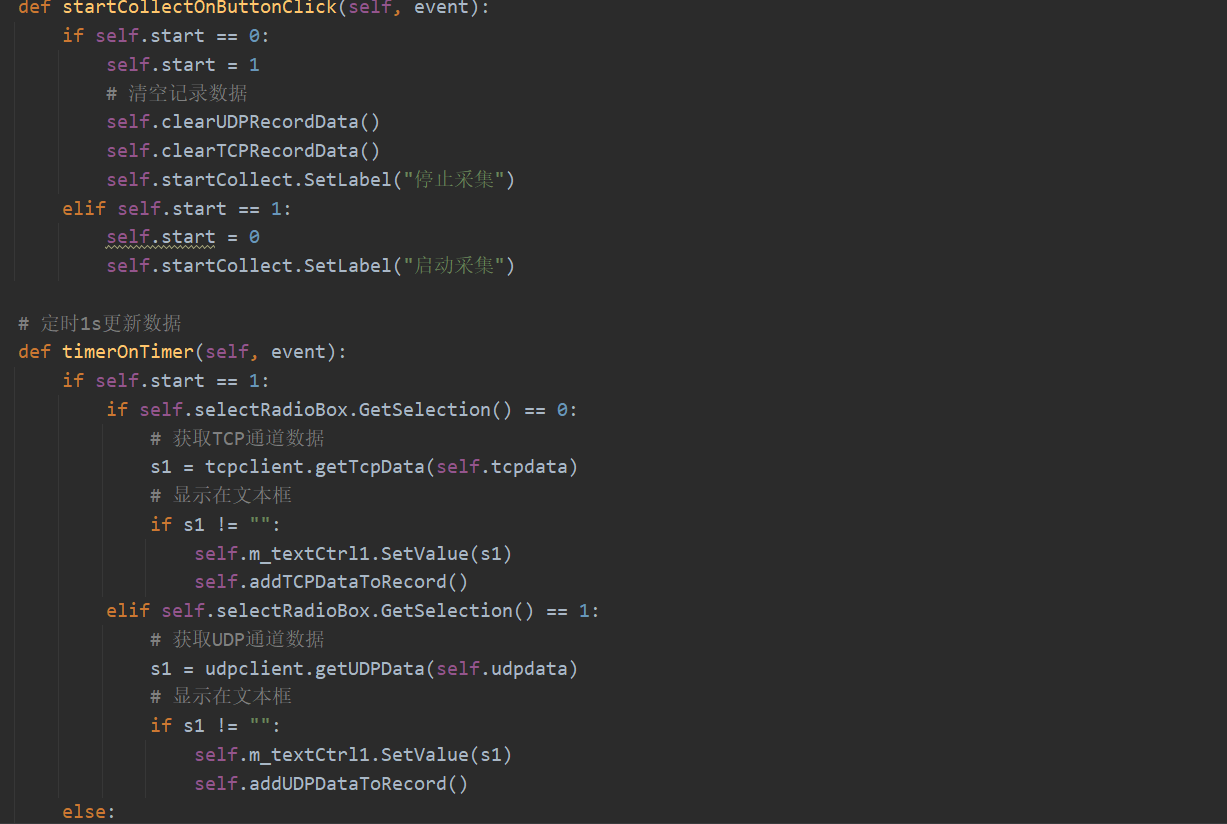


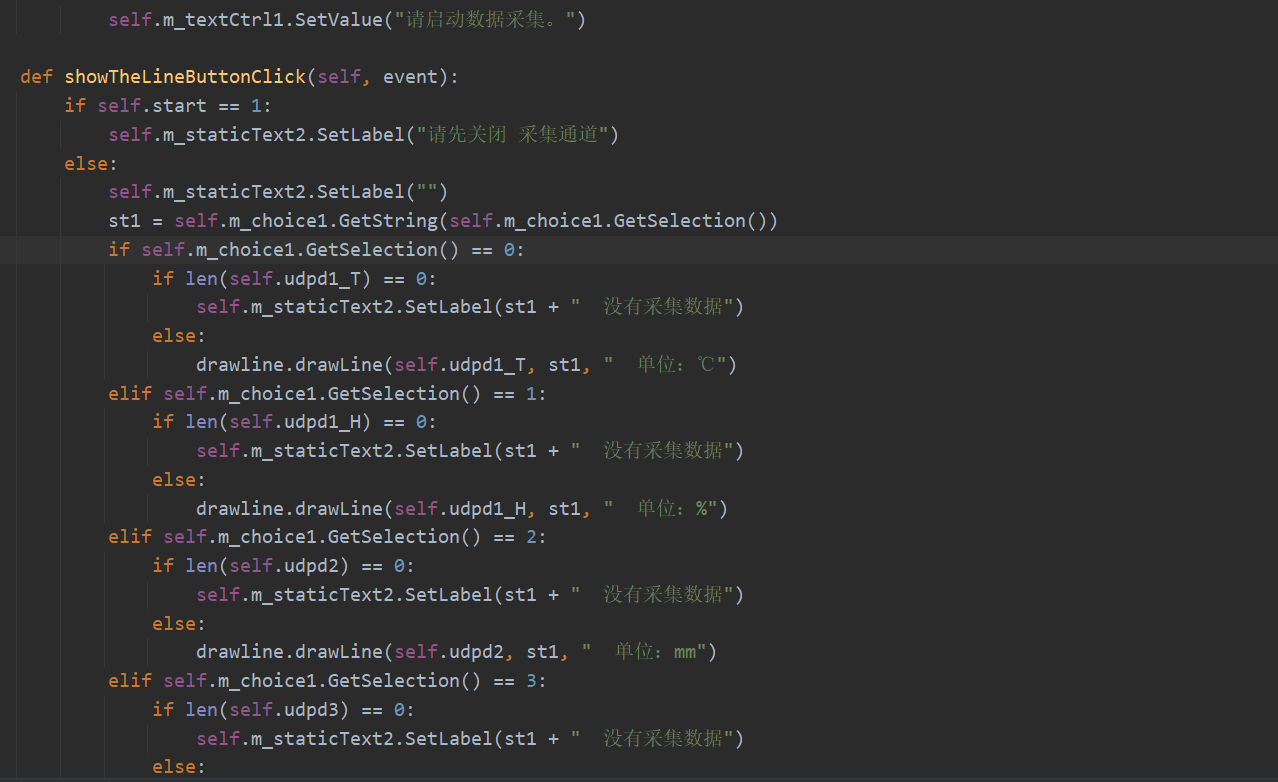
# 数据展示设计

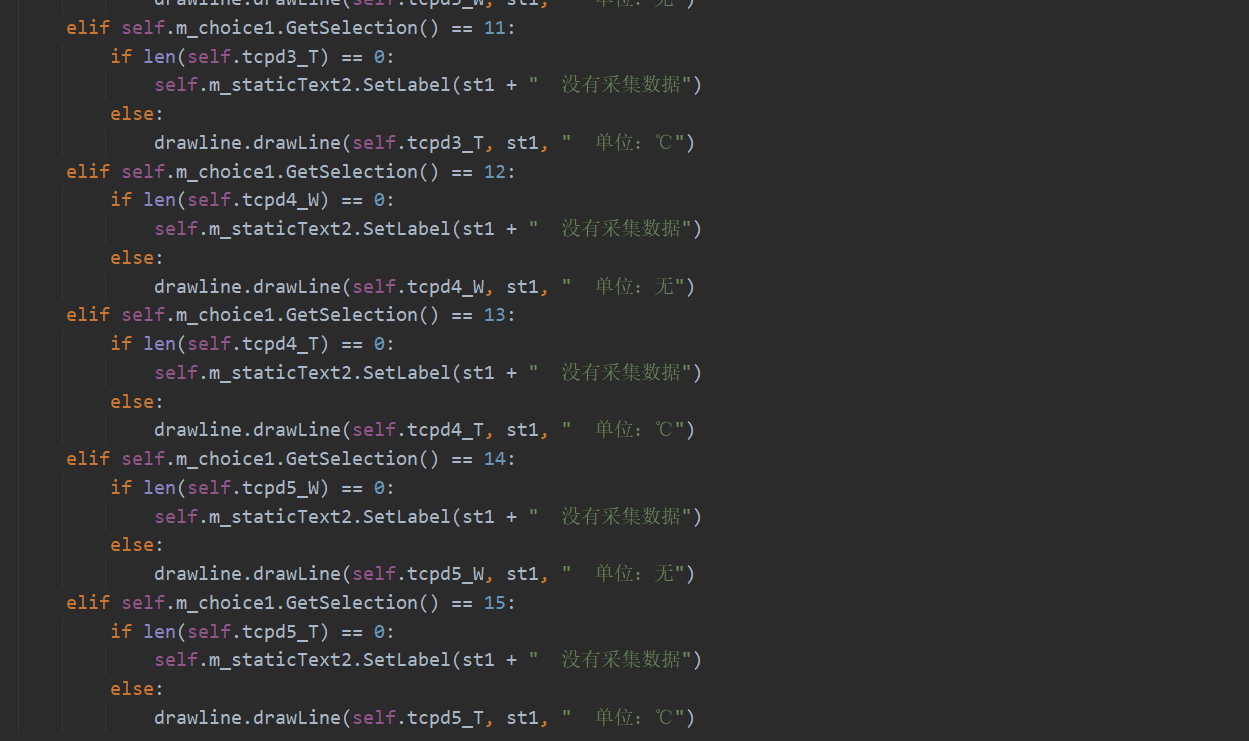












# 多线程设计

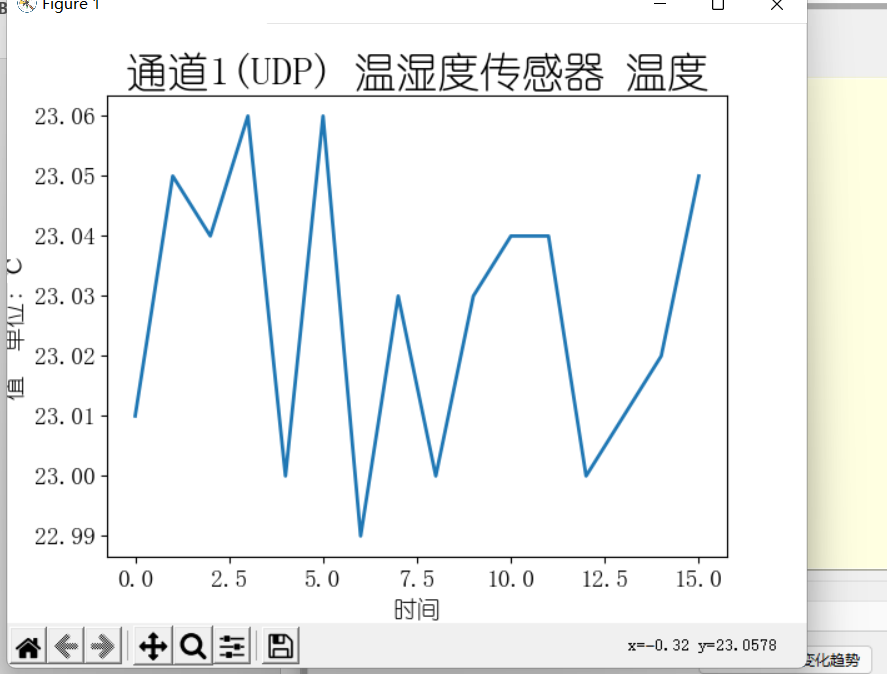
在本次设计中未用到多线程技术(感觉用不到！！！)

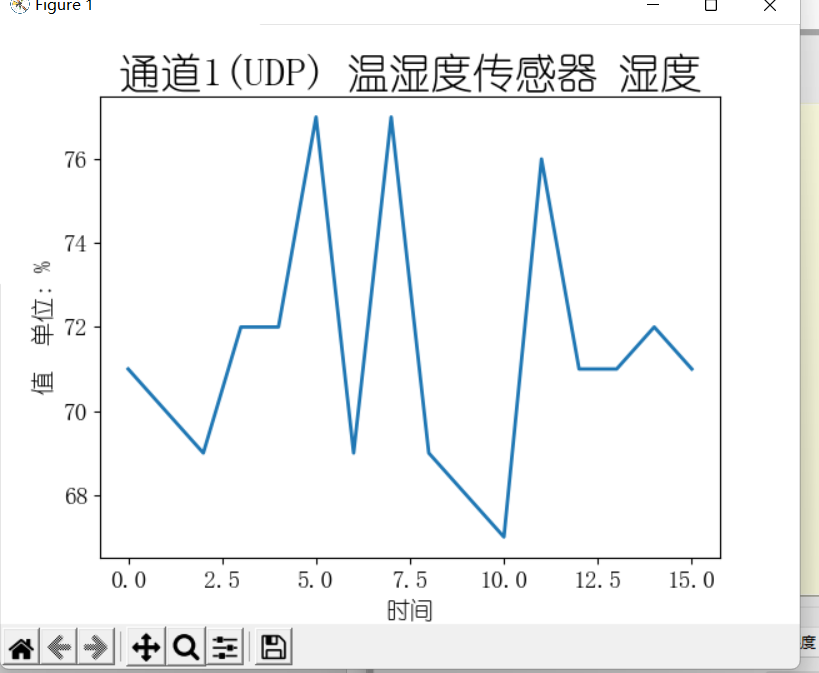
# 系统运行

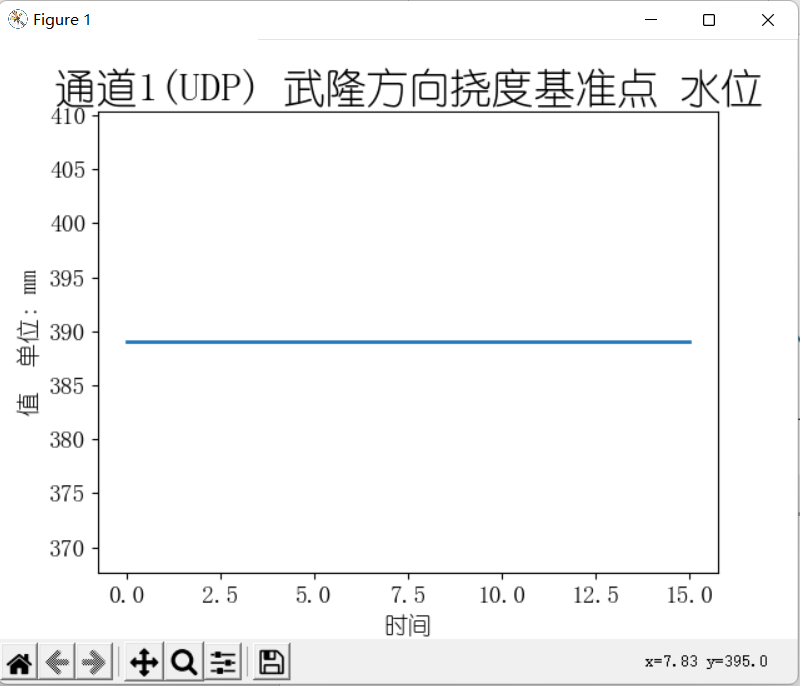
# 运行结果

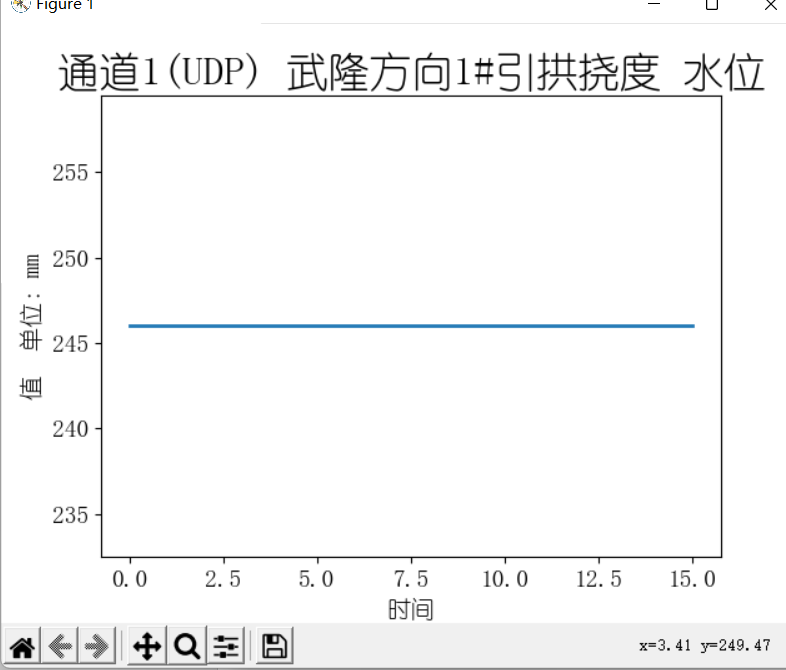


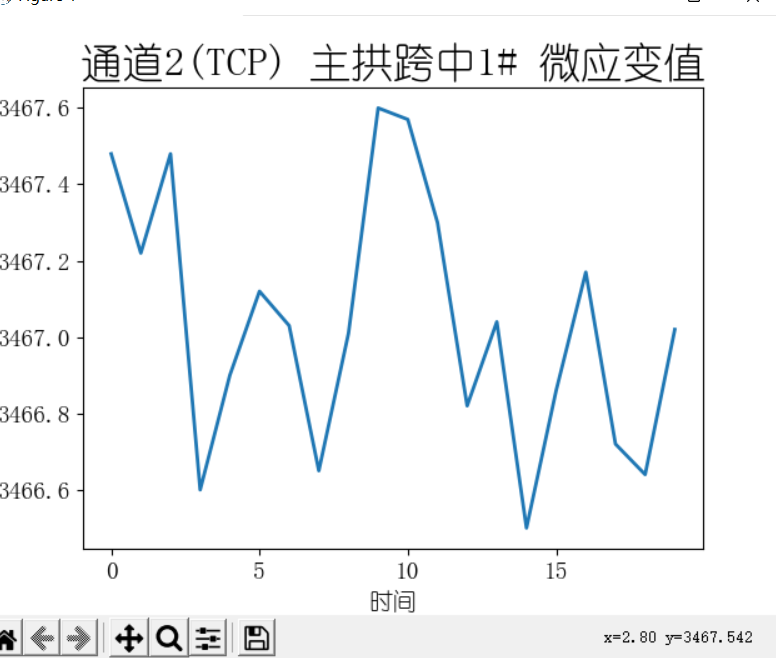


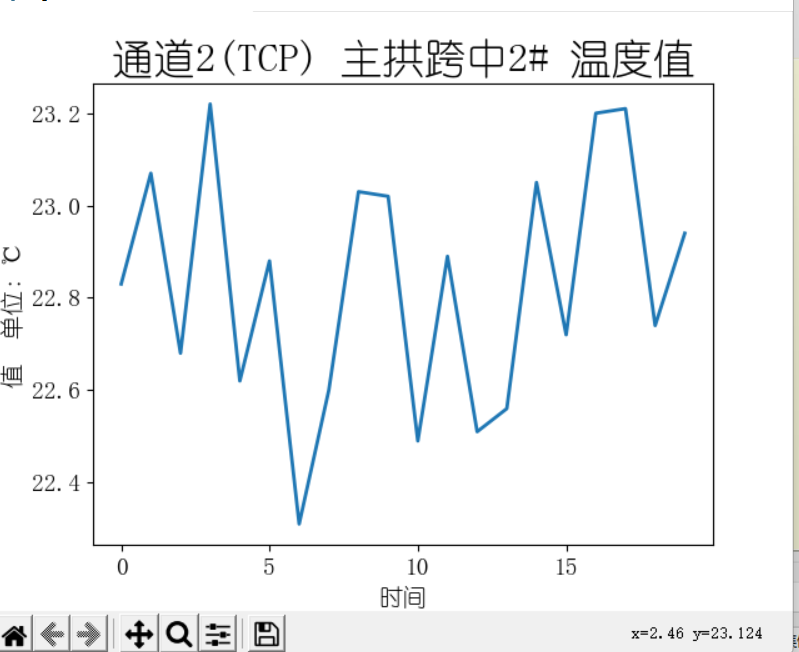




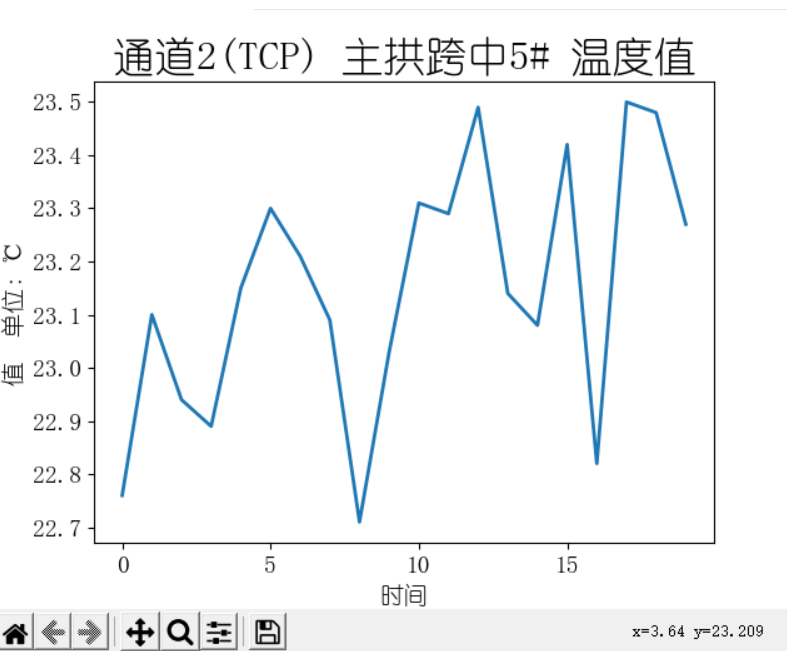


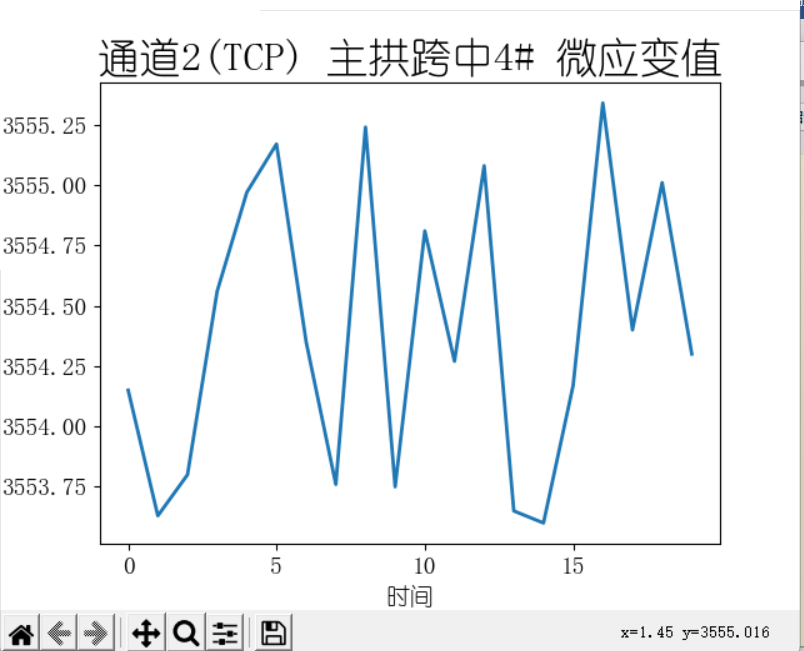






# 系统测试





总结：通过该项目设计，学习到了python中的网络编程socket与相关的modbus传输协议与其相关用法，并且学到了运用wxpython进行可视化界面的设计。我认为其中最重要的是对远程服务器数据的获取，学习到了在获取数据的过程中大部分数据是以二进制或者十六进制进行传输的并且需要相关操作进行打包与解包最终获取。并且额外设计的图形化界面是运用matplotlib进行可视化进行展示。但是，可能在项目设计中还有很多设计不好的地方，如界面美观度，代码的重复性太大等等，还需要不断进行深入学习python知识。