**实验室管理系统：香橙派显示大屏工作日志**

本次显示大屏工作主要分为两个部分：基于pyqt5的界面设计与逻辑加载（采用界面与逻辑分离的方法），以及香橙派硬件移植显示。

**目录**

[第1章 第一阶段：简单界面设计与逻辑初步 3](#_Toc179998468)

[1.1 基于pyqt5的界面设计与逻辑加载 3](#_Toc179998469)

[1.2 香橙派硬件移植 10](#_Toc179998470)

[第2章 第二阶段：基础信号的接收、显示和上传 12](#_Toc179998471)

[2.1 界面进一步优化设计与逻辑加载 12](#_Toc179998472)

[2.2 主代码（逻辑代码MainWindows.py）剖析 13](#_Toc179998473)

[2.2.1 接收下位机数据处理代码 14](#_Toc179998474)

[2.2.2 显示屏实时更新显示代码 15](#_Toc179998475)

[2.2.3 发送云端服务器代码 16](#_Toc179998476)

[2.2.4 代码其余细节说明 18](#_Toc179998477)

[2.3 香橙派移植显示 19](#_Toc179998478)

[第3章 第三阶段：界面改变与功率曲线添加 21](#_Toc179998479)

[3.1 基础功能实现 21](#_Toc179998480)

[3.2 曲线优化与数据库保存 25](#_Toc179998481)

[3.2.1 曲线优化 25](#_Toc179998482)

[3.2.2 数据库连接与保存 27](#_Toc179998483)

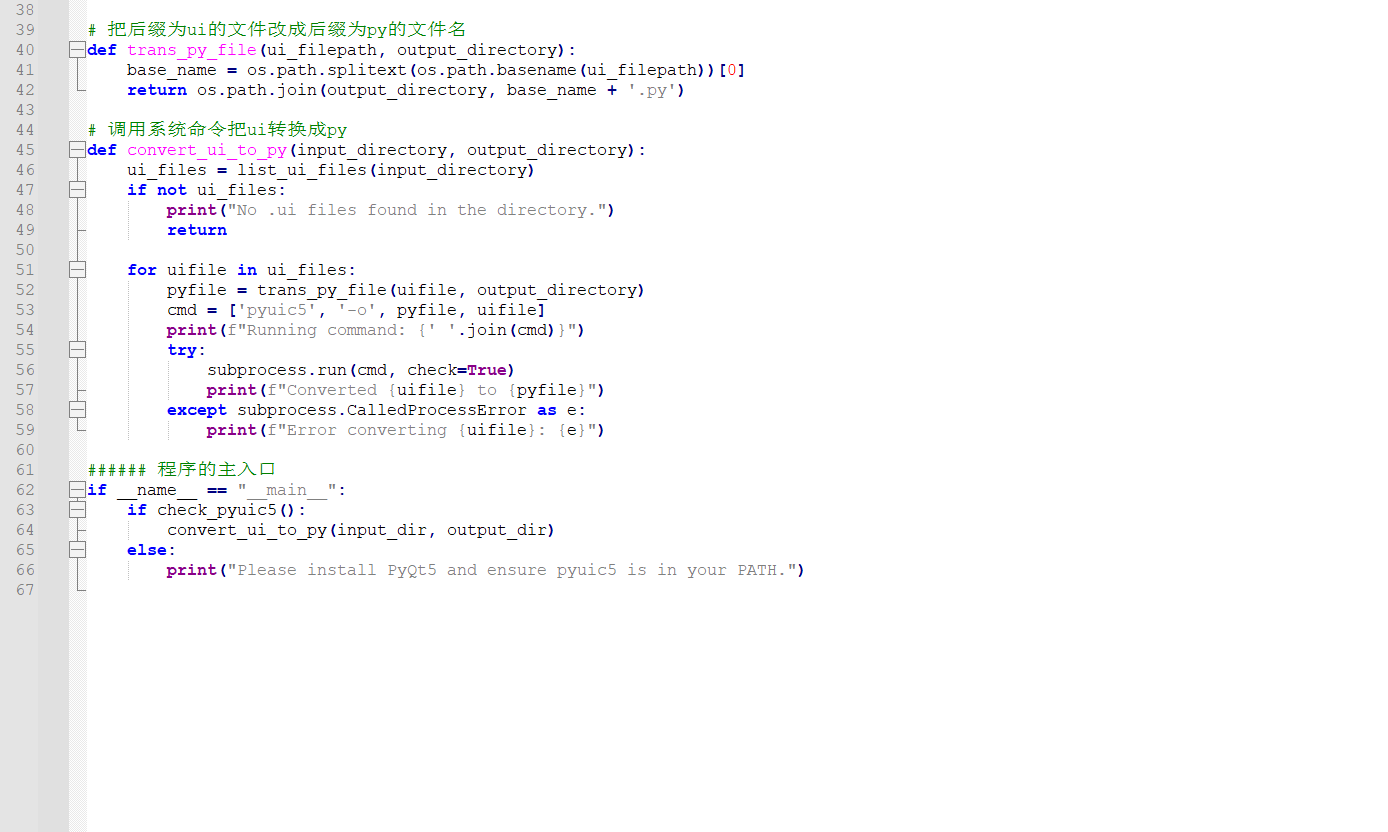
1. 第一阶段：简单界面设计与逻辑初步
   1. 基于pyqt5的界面设计与逻辑加载

界面设计不依赖于硬件，可以先在自己的电脑上选取一个ide，用pyqt5(python自带的一个专用于界面设计的包)和qtdesigner设计工具来进行第一部分的显示大屏设计。**要点补充：**通过命令行安装 pyqt5 和 pyqt5-tools，其中 pyqt5-tools 包含了 Qt Designer。安装完成后，使用 pip show pyqt5-tools 命令可以找到 Qt Designer 的安装路径。通常，Qt Designer 的可执行文件是 designer.exe，位于 Qt\bin 文件夹下。本次开发全程采用vs code来进行,如果采用的是pycharm开发，可以在安装了 Qt Designer 工具，并且配置了 PyCharm，你可以直接在 PyCharm 中通过 工具-External Tools-Qt Designer 路径打开它。这样，你可以直接在设计界面中创建和编辑你的应用程序界面。Vs code、designer的图标如下图所示。



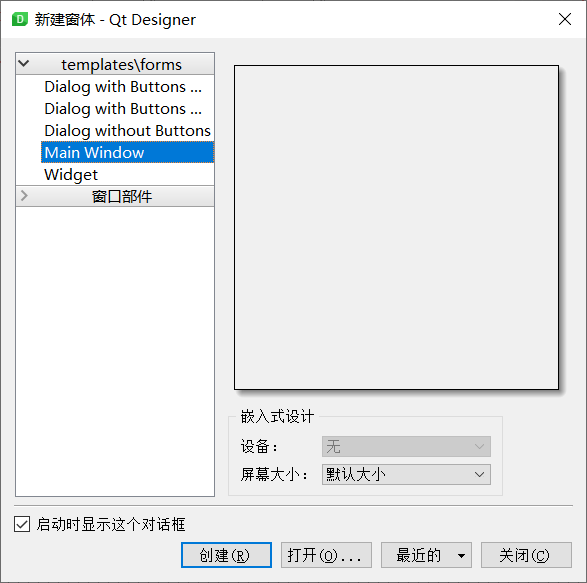
在上述准备工作完成后，开始进入界面设计，首先创建一个文件夹（为了便于系统管理图标、代码、资料），我命名为OrangePi。在里面建立了images文件夹（存储显示大屏会用到的图标、背景等一系列图片元素）、UI（存储由designer设计来的界面代码文件.ui文件）、工作日志文件（记录开发过程）、资料文件（存储香橙派：OrangePi\_5\_RK3588S\_用户手册\_v2.0、开发电路板的原理图等资料），最后在上述文件夹的同级目录下建立逻辑代码的文件：MainWindows.py。**要点补充：**1.在UI文件夹里面存放了一个写好了的convert\_ui.py文件（这个文件可以实现所有ui文件转换为py文件，转换为的py文件是一个类文件）；2.转换后的py文件也存放在了UI文件夹里面，便于统一管理。convert\_ui.py代码如下图所示。



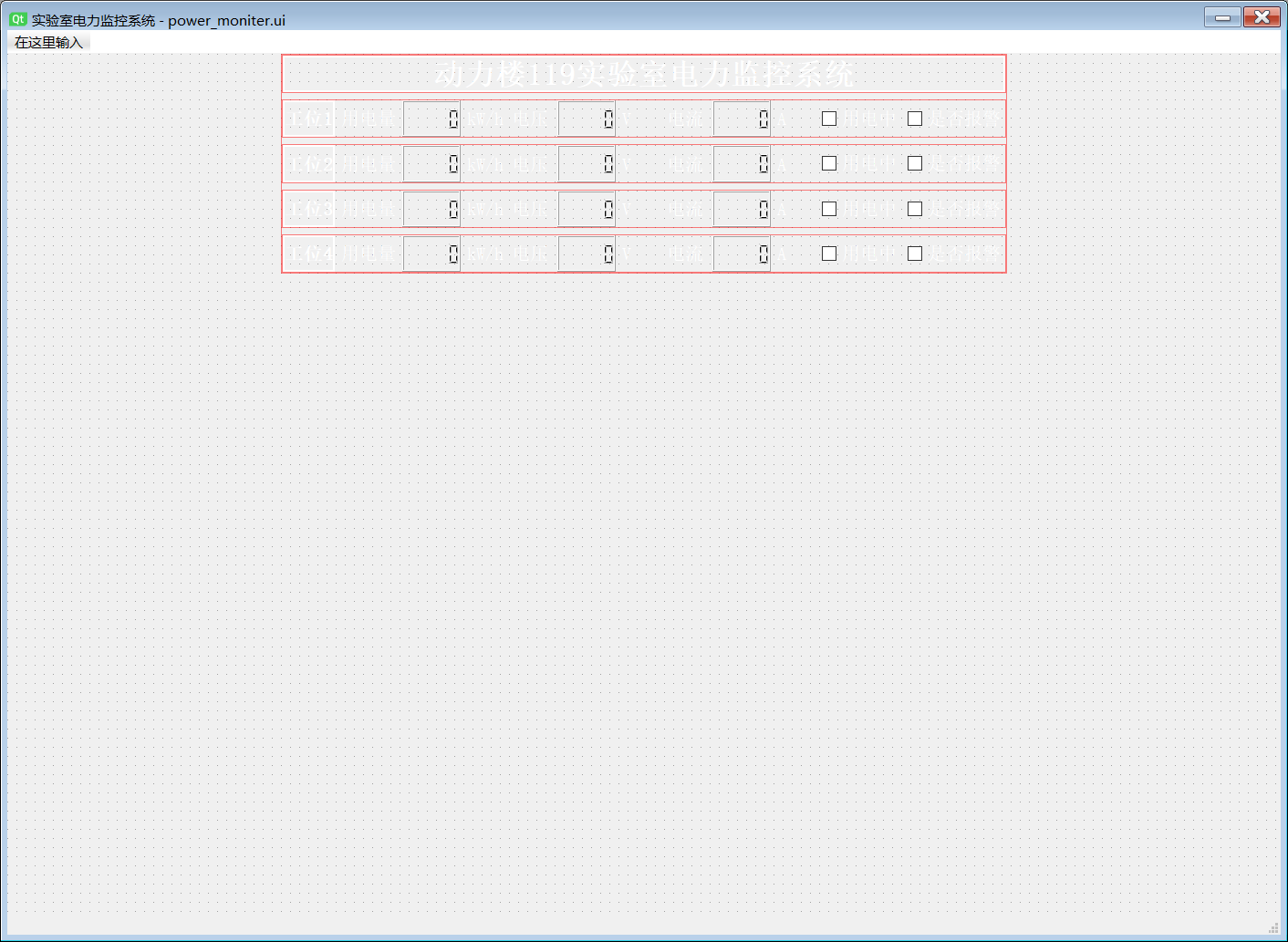


其次，打开designer开始进行显示大屏的界面设计。（需要自行上网查阅相关资料学习控件：常用的有Layouts、spacers、Buttons、Input Widgets和Display Widgets这些大类。）

具体工作如下：建立一个新的主窗口main window。如下图所示。设备、屏幕大小等其他的默认即可。



在创建好主窗口后，进行相应的界面设计。界面如下图所示：



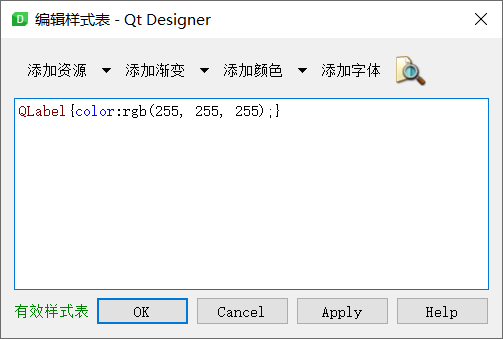
上述所示为工位电力监测部分，之后可在其他位置进行其他的界面设计。文字说明使用Label控件，数字显示使用LCD Number控件，用电中及是否报警使用Check box控件。文字大小、位置、粗细等属性在右侧的属性面板进行设置。

如下图所示。

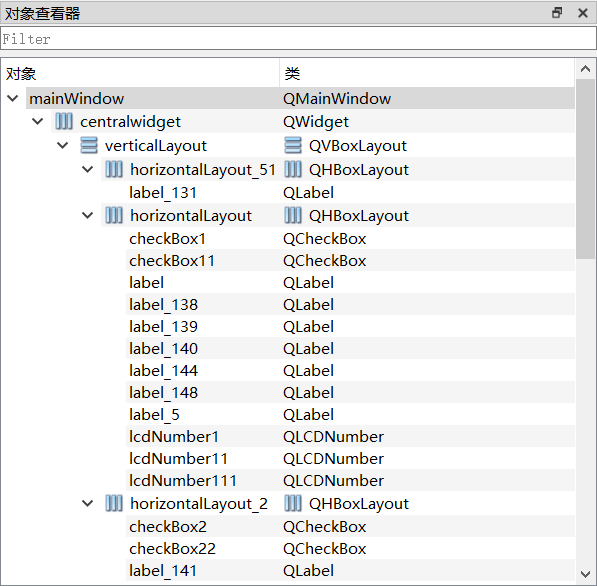


**要点补充：**控件显示的颜色需要通过样式表的方式来进行设置。右键相关的控件，选择改变样式表。样式表内容如下：通过增加一些伪代码来进行快速的设置。

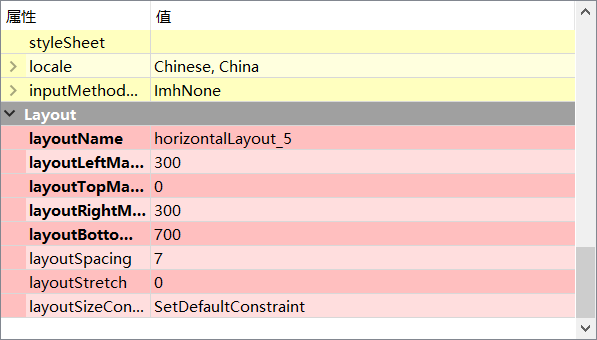
**特别注意：代码里面的Qlabel要随着控件的不同修改对应的名称。**



最后，我们需要通过布局管理器来进行界面控件元素之间的约束关系和在主界面中的相对的位置关系。（此处很容易发现，如果我们全通过python代码的形式来设置，那开发一个项目最后会很很多重复的代码，所以通过图形化设计加快开发速度）。



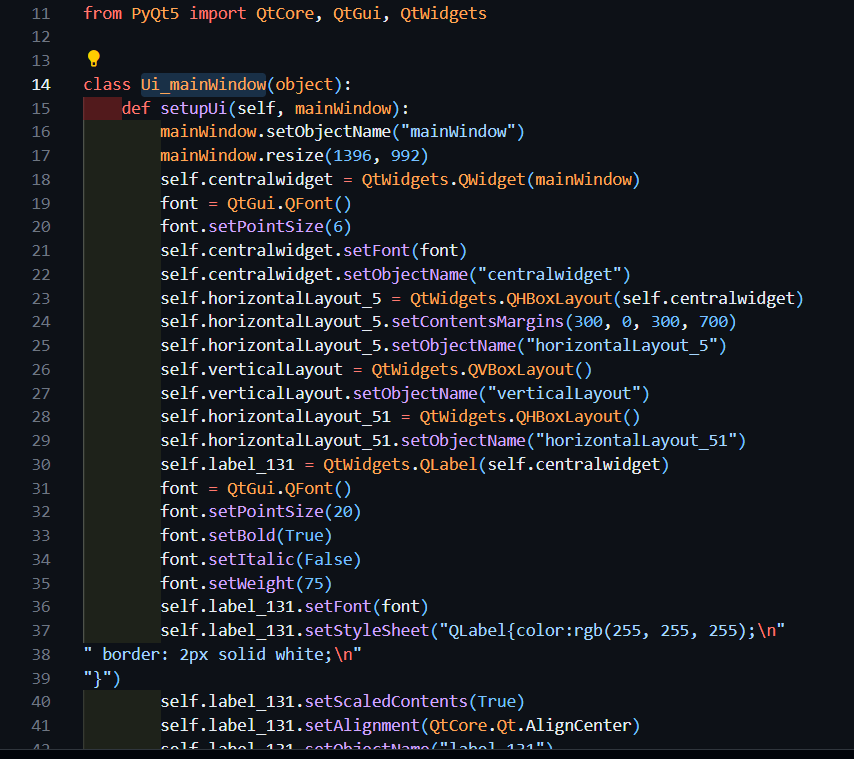
如上图所示，这是对象查看器，我们创建的所有对象都在里面，包括其对应的布局关系。我设置了是五个水平布局，这五个之间再来一个垂直布局关系。最后的这个整体来一个水平布局。**这里需要特别注意：**如果需要控件在显示屏的相对位置不随着屏幕的缩放而发生改变，我们需要使用布局管理器！！！最后还需要在对象查看器的mainWindow右键选择布局-水平布局。这时候界面会发生畸变，不要慌选择centealwidget，在下方的属性编辑器Layout模块设置四个页边距约束即可。如下图所示：



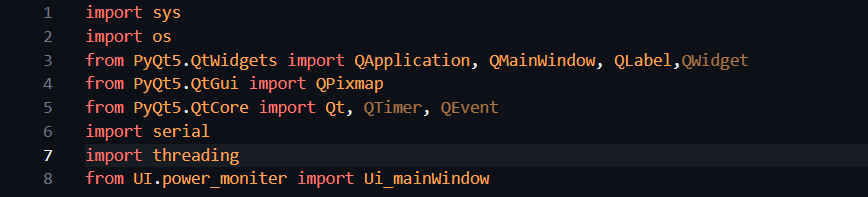
到这里，四个工位的电力系统监控界面设计大致步骤及关键细节陈述完毕。

我们保存文件就是相应的.ui文件。

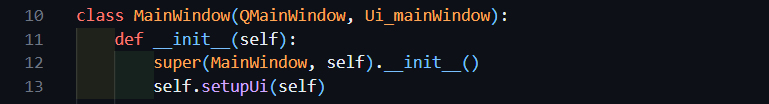
打开vs code 编辑器，选择运行convert\_ui.py文件，就会生成对应的.py文件。如下图所示只是一个界面的类封装文件。



上述只选取开头部分的代码截图。最后在MainWindows.py中建立逻辑代码与界面之间的联系，实现完整的界面显示功能。



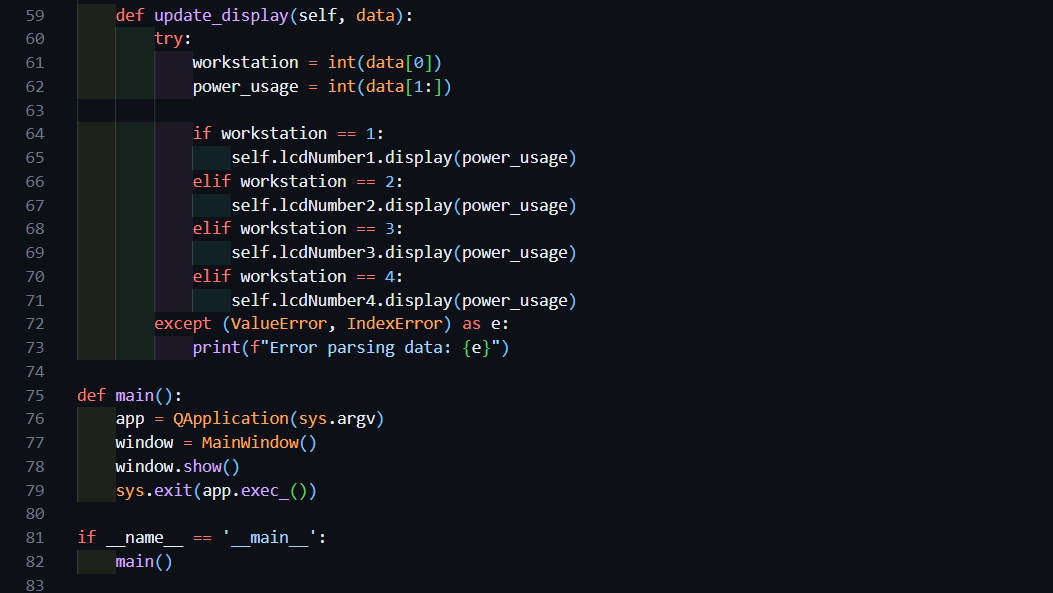
上述所示为导包，需要**特别注意**：第8行把ui文件转换为的py文件的那个类导入进来。



上述所示为界面代码的初始化，**关键：**第13行代码，缺少了会代码逻辑无法联系。



上述所示为界面背景图、事件接受等代码只选一部分展示。

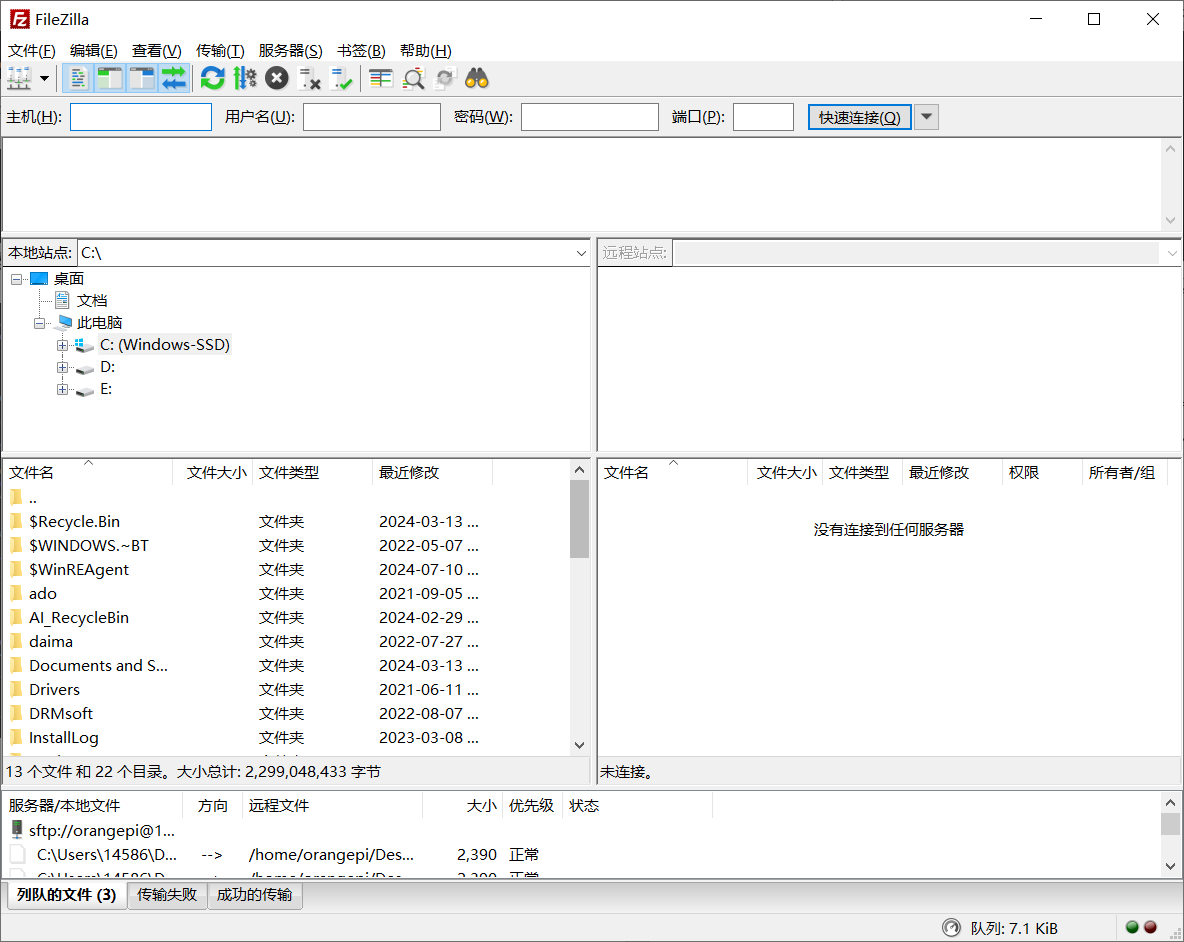


上述所示为接收数据的逻辑代码部分以及测试接口。（暂定，还需要进一步根据以后的修改）

* 1. 香橙派硬件移植

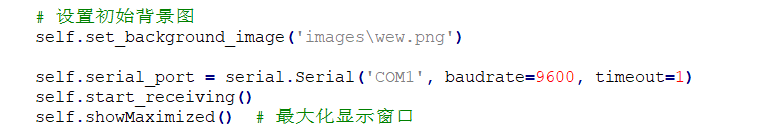
首先，掌握基本的linus系统操作指令。其次需要记录一些**关键信息：**1.快捷键ctrl+alt+t快速打开终端，输入su root，密码：orangepi（小写）。2.所有的动作指令：pip、python等需要写成pip3、python3。例如运行python文件，python3 xxx.py。3.检查香橙派的键盘、鼠标等外设是否插好，wifi模块是即插即用的。打开电脑的个人热点，让香橙派连接上去。4.个人电脑PC端需要下载一个filezilla软件（基于sftp文件传输协议的）,输入主机号（电脑个人热点192巴拉巴拉的）、用户名密码都是orangepi、端口默认22。

操作界面如下图所示：



左边是个人电脑的文件，右边是香橙派主机上面的文件系统。在香橙派主机界面创建一个类似的工程文件夹及相应的子目录，把左边的相应代码资料直接拖进来即可。

最后**特别注意**：移植的话代码相应的地方需要修改：



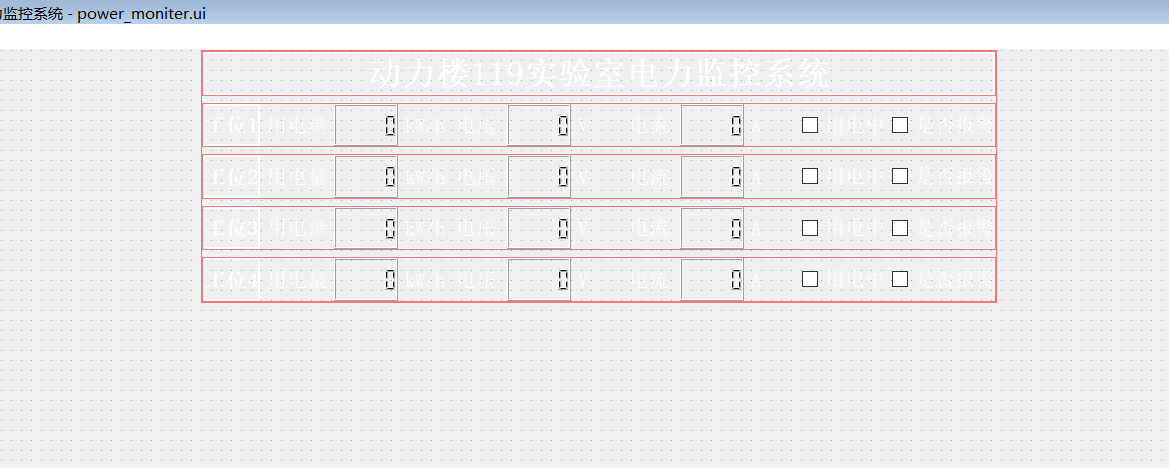
端口号需要改成香橙派对应的，需要查看原理图以及香橙派资料文档。例如打开pdf文档搜索orangepiEnv.txt，会显示相应的打开串口设备的代码/dev这种。

**上述是显示大屏工作的基本流程，还不涉及具体的数据接收，逻辑代码编写，界面显示与云端上传。**

1. 第二阶段：基础信号的接收、显示和上传

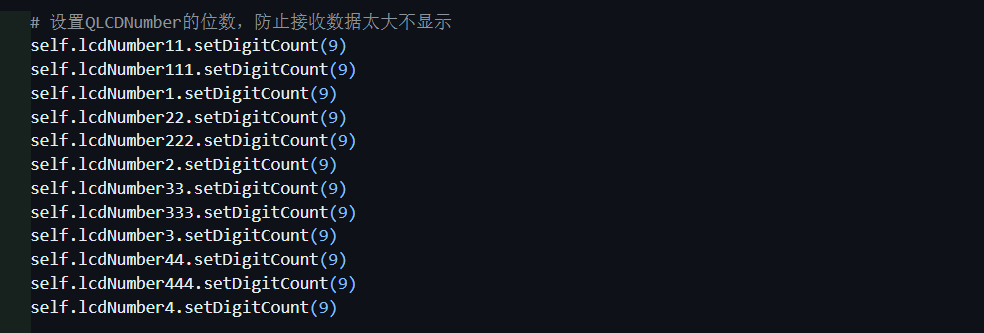
本次工作日志的记录还是按照两部分慢慢展开：基于pyqt5的界面设计与逻辑加载（采用界面与逻辑分离的方法），以及香橙派硬件移植显示。这次会记录一些在接收实际数据信号的过程中常见的一些bug，可供后来者快速查阅攻略。

* 1. 界面进一步优化设计与逻辑加载



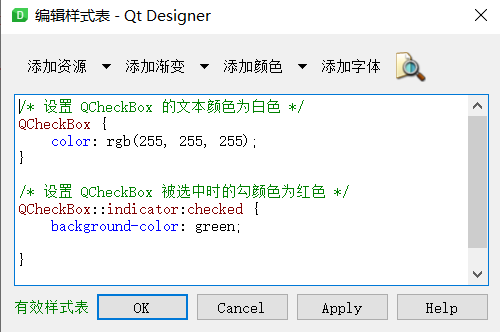
基本的UI界面不变，主要是里面的伪代码编写，以此来呈现更好的显示效果。

**1.LCD显示的位数变化。**需要在MainWindows主界面代码里面增加如下代码：

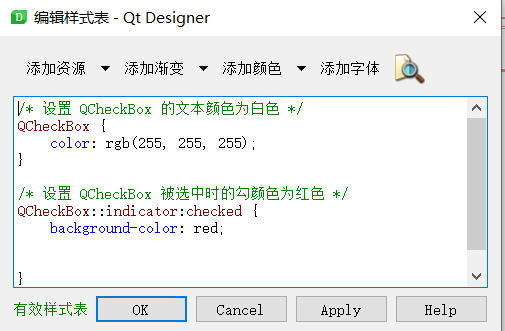


在一开始接收下位机信号来的时候，你会发现代码都没有报错，print调试也没问题，但主界面的LCD就是不显示数字，**这时候要注意是下位机发过来的电压电流用电量的值太大了。**发送过来的是16进制的值，转换为10进制后我发现有9位了，lcd默认显示的位数小于它！所以设置更高的位数防止出错。此外下位机需要更改下这三个的值，小一点合理一点！

2.**复选框checkBox的伪代码编写。**主要分为两个一个是用电量的，如下图所示。



上面这个是用电量的，我设置了选中是绿色，符合逻辑与美观（注释红色有误）。另外一个是是否报警的复选框，如下图所示。



这个我设置选中是红色，符合报警时候的提醒与警示。

**3.特别注意每个控件的名称编写，主代码是要连接对应的控件名称的**

* 1. 主代码（逻辑代码MainWindows.py）剖析

代码里面主要包括：处理下位机数据包的逻辑代码、显示大屏实时更新显示逻辑代码，以及发送云端服务器的发送代码三大块）。**讲解主要分为四部分，上述三大块+主代码中其他的细节问题。主要说明下核心代码，每个部分其余代码不细说。**

* + 1. 接收下位机数据处理代码



上述代码主要是打开一个线程用于专门的RS485串口数据接收，之后做lora还会再开一个线程。其次接收数据采用了一个buffer缓存区。Buffer = b””这个代码是创建的字节缓存区，一开始是空的。**这里要补充一个知识点特别注意不然逻辑代码会错乱：一个16进制数是一个字符，两个16进制数才是一个字节，这里会涉及之后的选位问题。**

此外大致讲解下这段代码：创建缓存区，每17个字节为一个单位进行后续的逻辑分析。因为发过来的数据包有很多数据，每个数据量是17个字节组成。如下图所示。

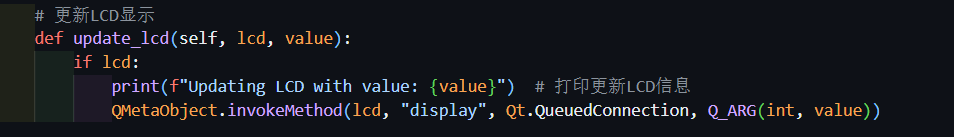


这里需要看懂，不然后续无法展开。**后续的捕捉逻辑是：每17个字节数据量为一个单位，去捕捉数据标识位和数据。由于数据包是一直循环发送的，而且速度很快115200，所以不能采用简单的从头数位数，拆分为单位来做更好自适应捕捉。**

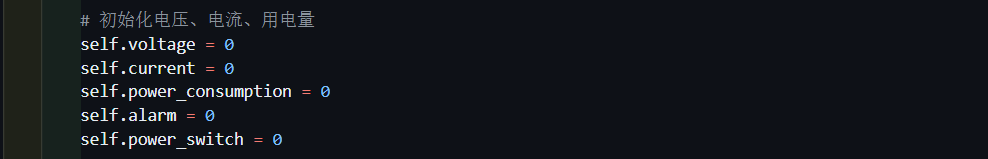


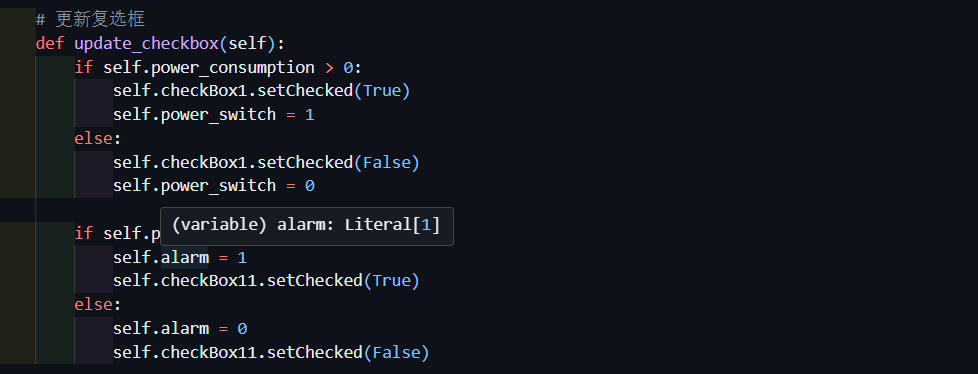
上图所示是处理数据包的逻辑代码，主要分为标志符、值的捕捉，通过if语句来进行赋值，给之前建立的全局变量新的值。

* + 1. 显示屏实时更新显示代码



Lcd更新显示的代码，便于实时更新显示，而不是捕捉固定的值不变。





上面分别是先定义全局变量来设置默认值，主要是便于后续一起更新后存入json文件发给服务器。只能这样才可以一起实时更新显示服务器，如果不定义的话，每次服务器只会接收到新的一个特殊亮，其他的参数是默认设置的就会。

* + 1. 发送云端服务器代码

与云端服务器连接主要采用mqtt协议的代码。需要自己去了解学习下mqtt协议这块，我主要说明下核心知识点。

mqtt 协议，基于订阅发布。http 协议基于请求要公网 ip 服务器。mqtt 协议基于物联网更适合硬件之类的信息传输，传感器。

mqtt 协议：

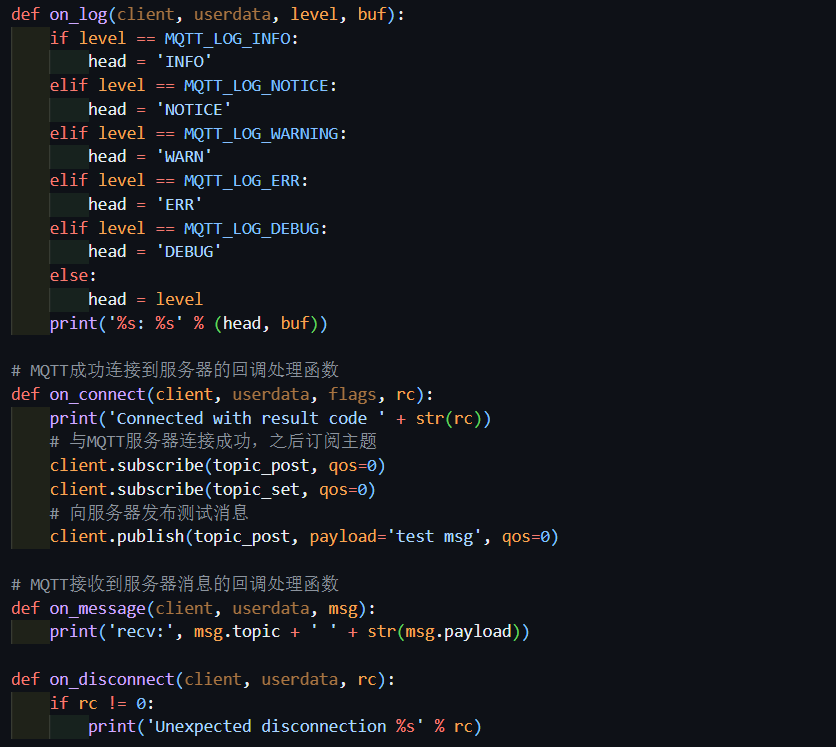
客户端（我的电脑主机）

代理（服务器）

所有的客户端连接到代理，代理处理消息接收，也可以分发消息到客户端。



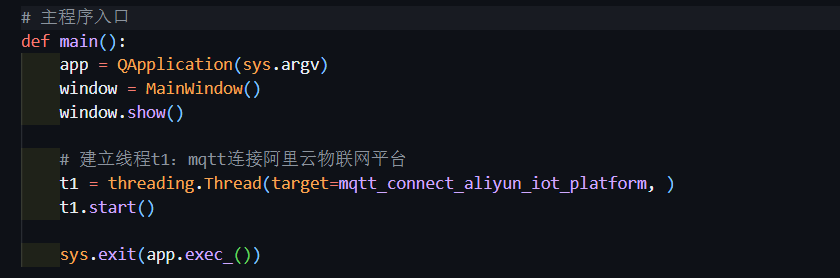
上述所示是服务器基本的一些配置参数。



上述所示是设置的一些与服务器连接的回调函数，用于测试便于。



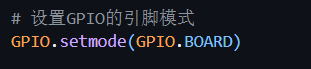
这个代码是与服务器连接通信的核心代码。

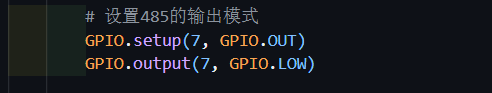


**特别注意，需要在main函数这里创建一个与服务器通信的线程！！！**

* + 1. 代码其余细节说明



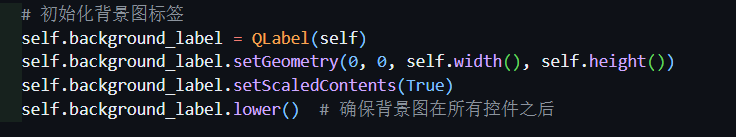


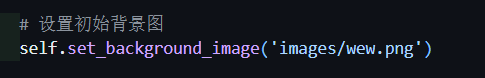


**rs485 有一个控制端口，低电平接受，高电平发送。如果没设置会出现代码都没错，但为什么没信号！**

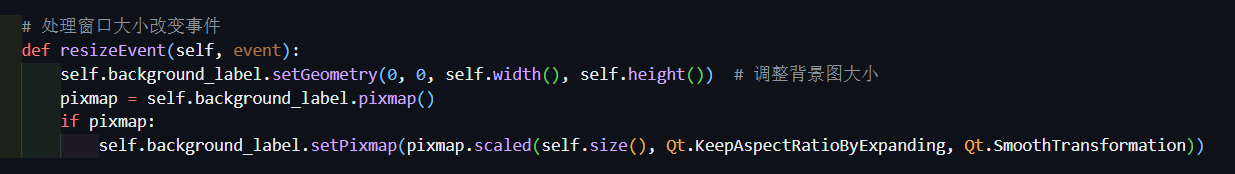
**import OPi.GPIO as GPIO 导入香橙派端口控制的包**

**GPIO.setmode(GPIO.BOARD)设置序号控制引脚**







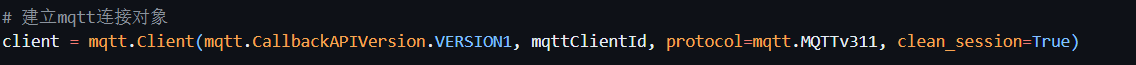


**上述所示是背景图自适应窗口大小的核心代码，因为有一个bug。如果用样式表设置背景图，它不会随着窗口缩放自适应大小，需要用Qlabel的方式来做**

* 1. 香橙派移植显示

本次香橙派移植显示没什么技术难度，因为我的开发是软件独立做好后直接把代码导入香橙派运行显示。主要说明一下本次移植过程中出现的一些bug。Bug的调试还是很费时间。特别强调：需要会linus基本的操作命令，对linus系统熟悉，之前一些常用的linus指令操作注意事项都说明过了，这次就主要记录下小bug。

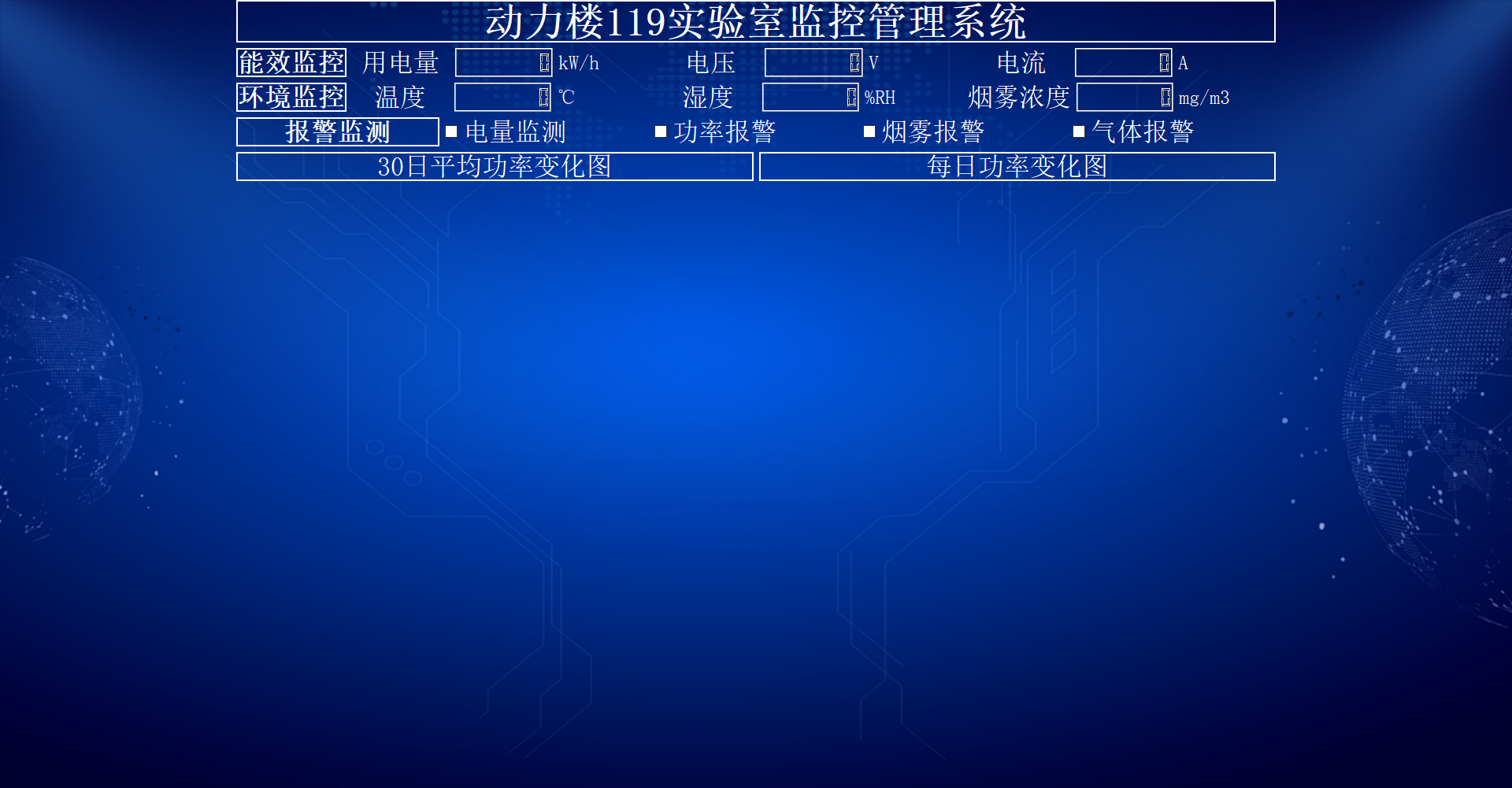
1. 可以不直接su root先进入根节点，可以每次运行代码**加上sudo**，再次强调pip3,python3要指定版本。
2. 因为本次RS485出现了通过序号直接控制gpio口的引脚，所以需要在主机里面查看哪一个是控制引脚，终端输入**gpio readall**。对照我工程文件Orangepi的资料文件中的香橙派原理图查看,序号7.
3. 特别注意，因为要和服务器连接，会要下载一个包在香橙派主机paho-mqtt。这里容易出现两个bug，**第一个**是下载包也用pip3了但就是显示没安装，此时需要加上sudo给它权限去下载。sudo pip3 install paho-mqtt；**还有一个bug**可能会出现，就是显示paho这个包API2.0要求在创建客户端传入一个版本参数2.0，但你会发现，假如后又说不能加入很矛盾。此时在在代码里面找到客户端创建代码修改如下：



要在形参的最前面加上：mqtt.CallbackAPIVersion.VERSION1

1. 补充注意背景图，**背景图不能太小**，几kb是不行的加载不了主机会。用qlabel方法，不要用样式表做。
2. 补充说明进制：16 进制数例如：0x458388收到数据的时候没有 0x，其次字符上是 6 个字符，字节是 3 字节两个 16 进制数是一个字节！
3. **代码不要通过 qq 或者微信聊天框直接复制粘贴发送**，有时候运行不了会显示非 utf-8 编码。以文件的形式发过来。
4. 在自己的笔记本修改了界面后，把代码传到香橙派要注意：如果 ui 界面有更改了，首先在 qt designer 中要保存，然后运行 convert 转换，把新的 ui 和 py 转移过去，且注意特别注意：**要开个新的终端运行代码，不然界面不会有变化 bug！！！**
5. 第三阶段：界面改变与功率曲线添加
   1. 基础功能实现

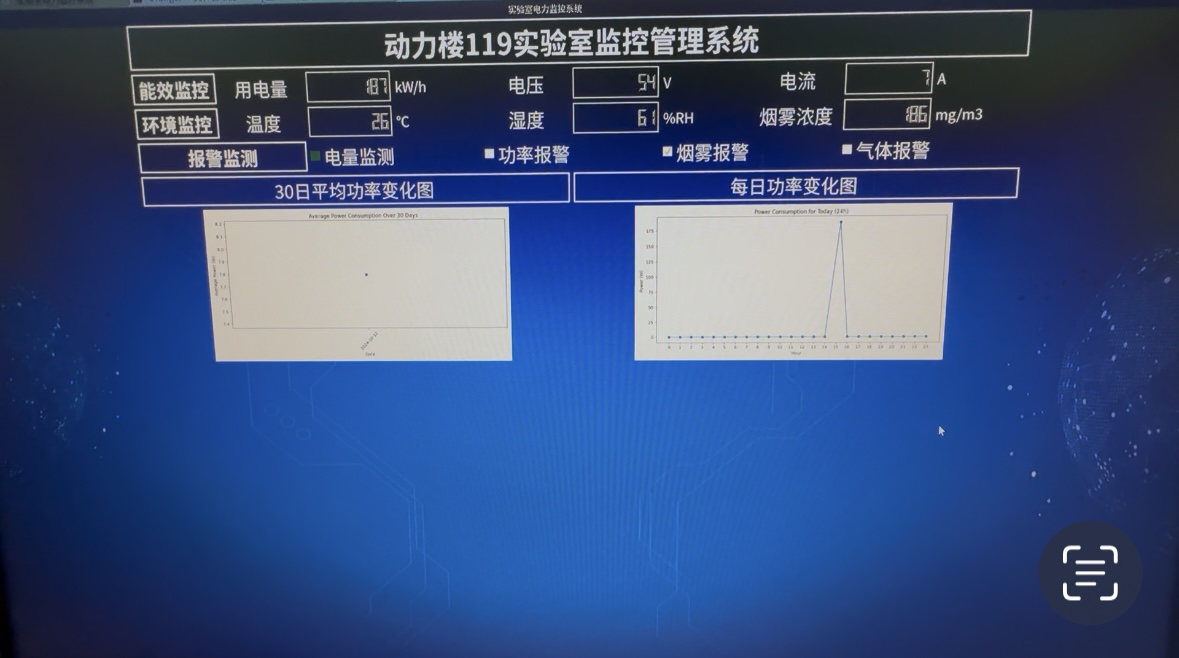
显示大屏界面优化调整与功能添加，在之前的界面基础上改变，涉及的技术没变化。 现有界面如下图所示：



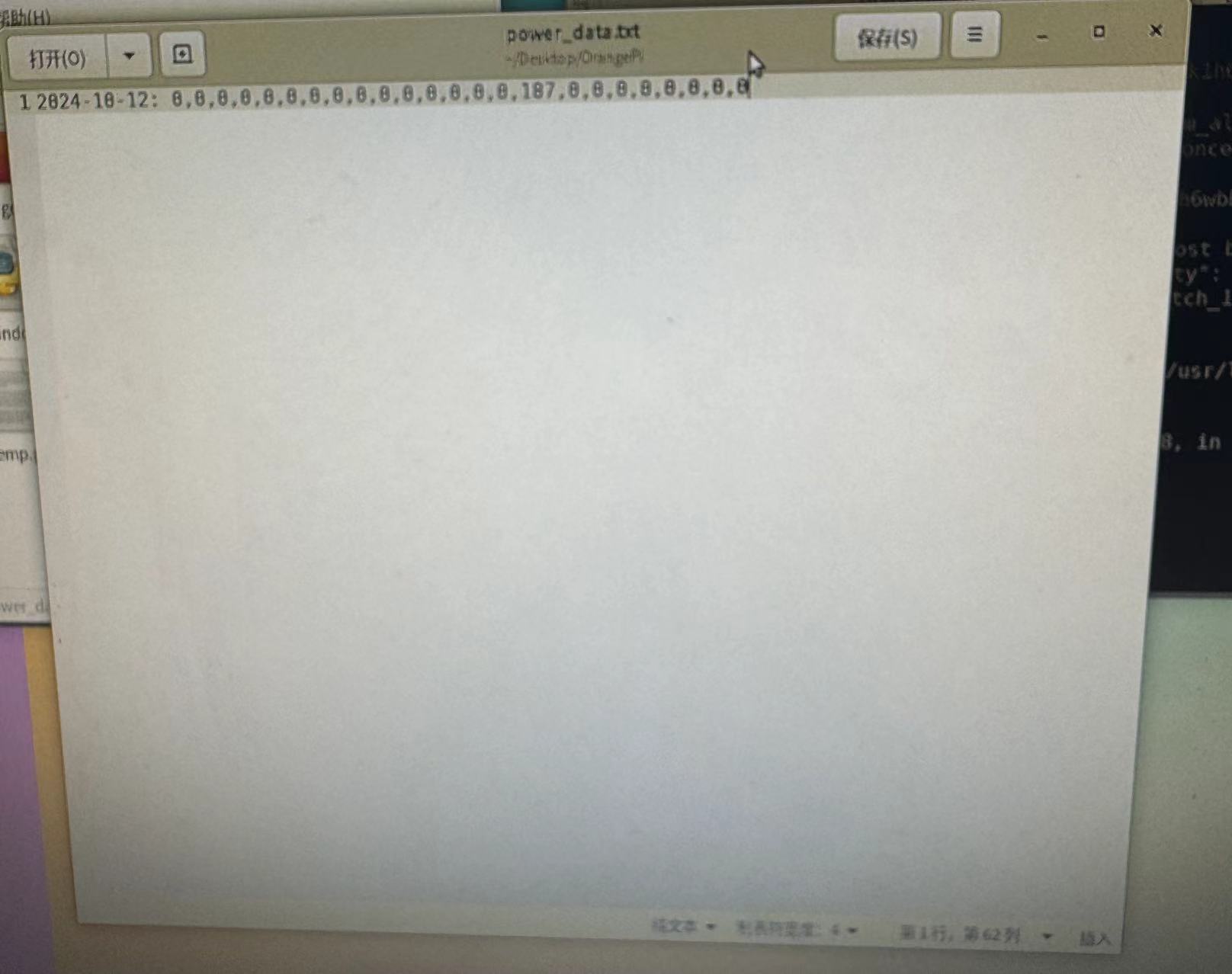
增加了新功能：温度、湿度、烟雾浓度、烟雾报警、气体报警等。

除此之外，还加入了可以显示30天平均功率和当天24小时功率曲线展示。采集到的功率数据由从机发送给主机，对每隔1h的功率数据进行保存，保存进txt日志文件中，这样每天采集24个数据。

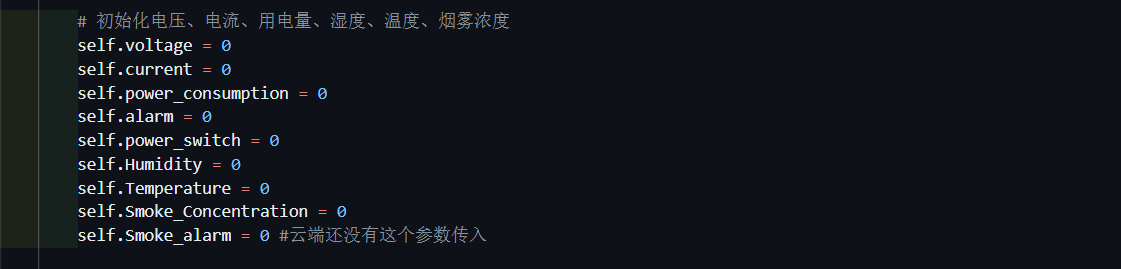
由24个数据进行当天24h功率曲线绘制。30天平均功率曲线的计算是：每天采集的24个功率求平均值，作为当天的平均功率。30天曲线绘制只会绘制近30天的数据，第31天来了，会删除第一天的数据，保存第31天的，以此类推。



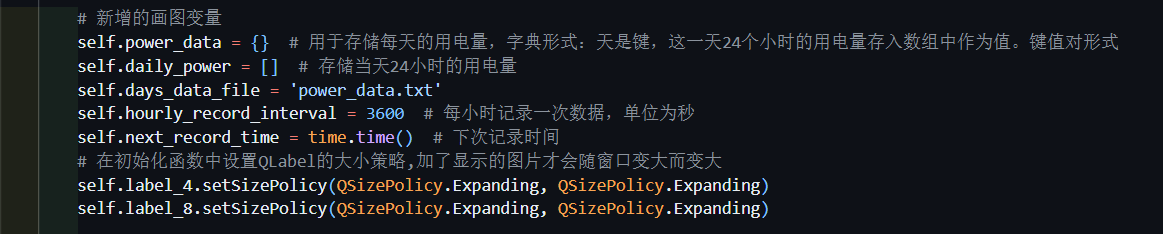
香橙派Orangepi工作空间目录下存放power\_data.txt文件，其中存储的数据如下所示：



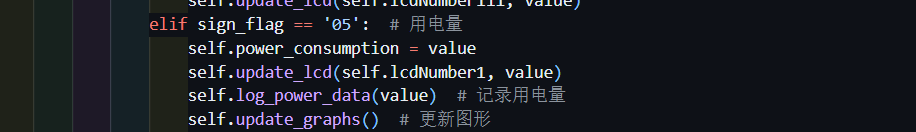
温度、湿度、烟雾浓度等新功能初始化：



画功率曲线图变量初始化：



保存功率数据，以及画图函数调用（自己封装的）



Txt保存数据以及自动存储近30天功能代码：



画当天功率曲线图代码：



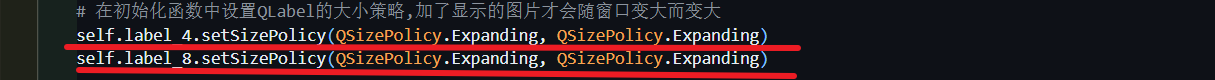
画近30天平均功率代码：



把画好的曲线图放入对应的qlabel界面：



关键代码：设置两个曲线图的qlabel大小特效，少了的话曲线图很小不随页面缩放，很容易出错。



* 1. 曲线优化与数据库信息保存
     1. 曲线优化

之前每日功率曲线的展示是把每天化为24份，每1h选取一个点记录数据保存并画图展示。现在进行了优化调整，功率曲线为实时接收传入的功率数据，实时画图展示。随着数据的过来，曲线图在界面上实时变化显示。

修改的底层源码如下：

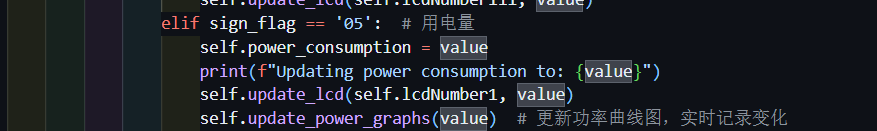


图1.用电量（功率）数据更新函数调用，实时显示的封装函数

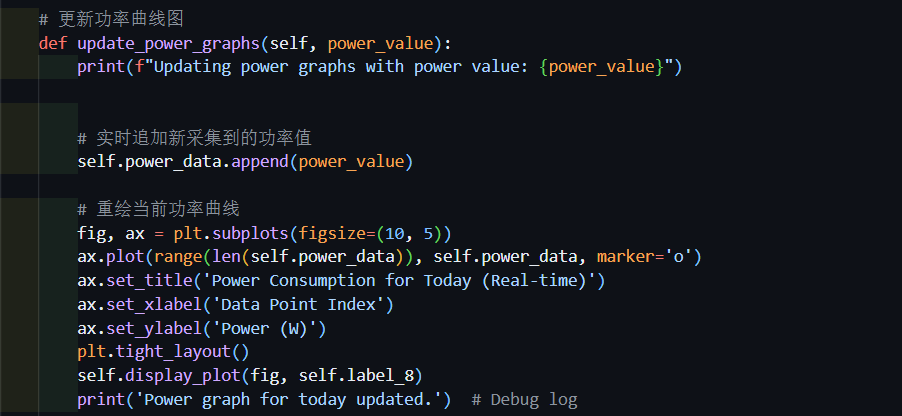


图2.每日功率曲线的实时更新函数

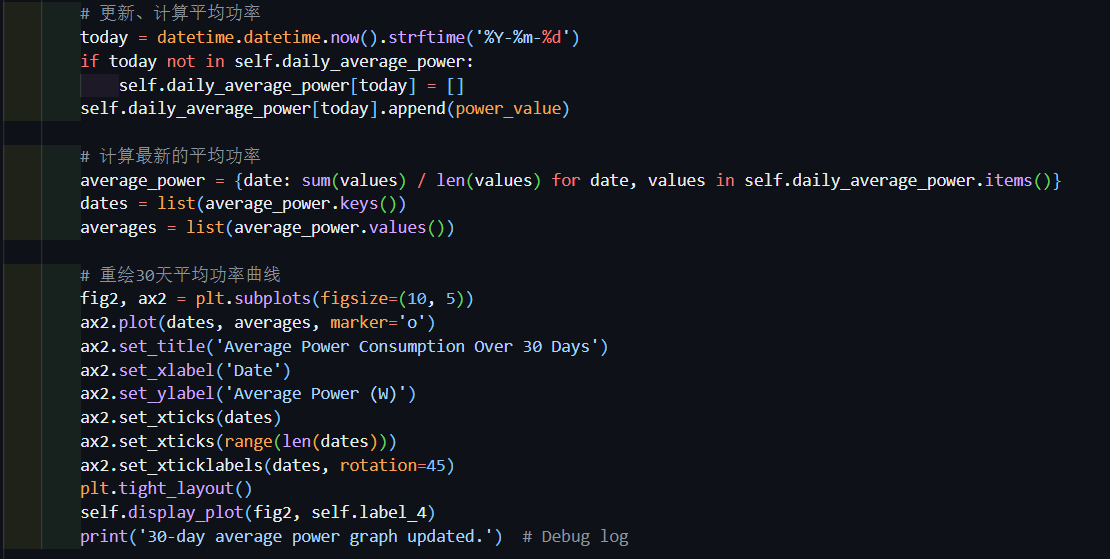
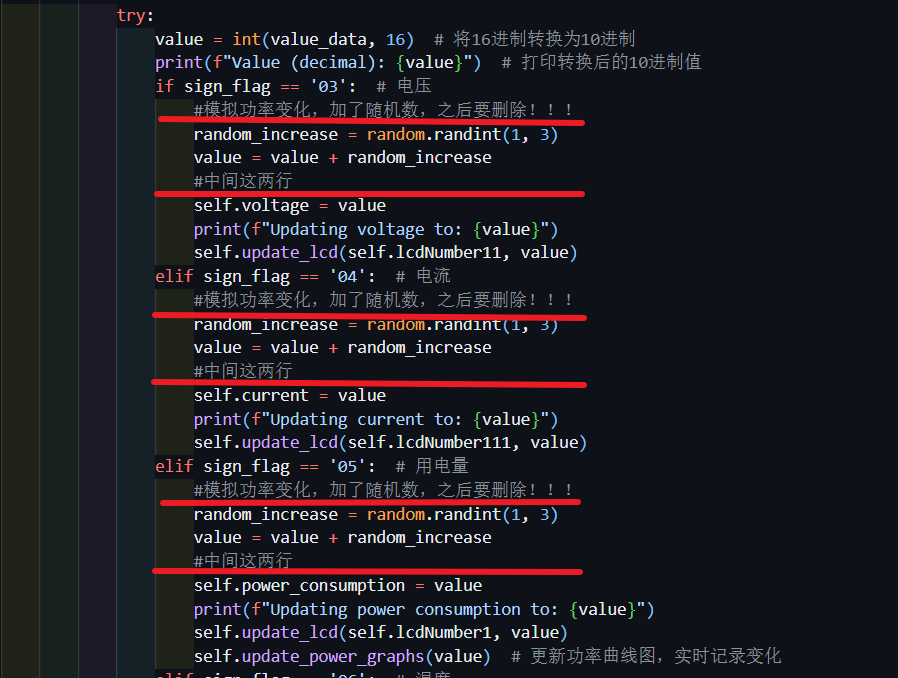


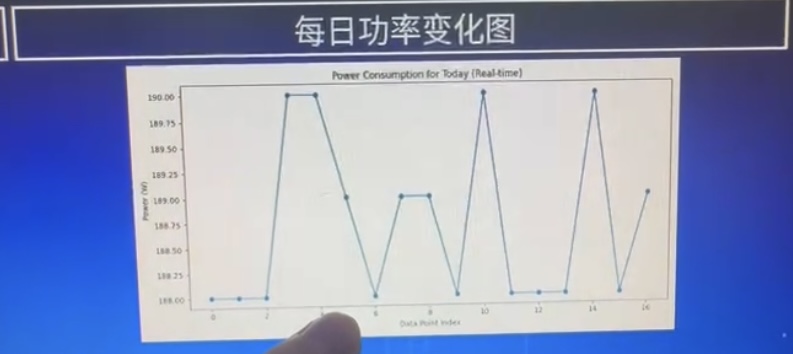
图3.30日平均功率曲线代码

上述是相关的代码，除了每日功率曲线的变化调整。30日曲线的底层代码也进行了相应的变化。随着当天功率数据的不断变化，当日平均功率也会波动。当这天过完到下一天，会自动计算前一天的平均功率，作为前一天的固定值。

由于从机发送的是固定的数据过来，曲线图是先增加随机数来模拟变化。代码如下图所示。主要是电压、电流、用电量的模拟！



曲线模拟如下：



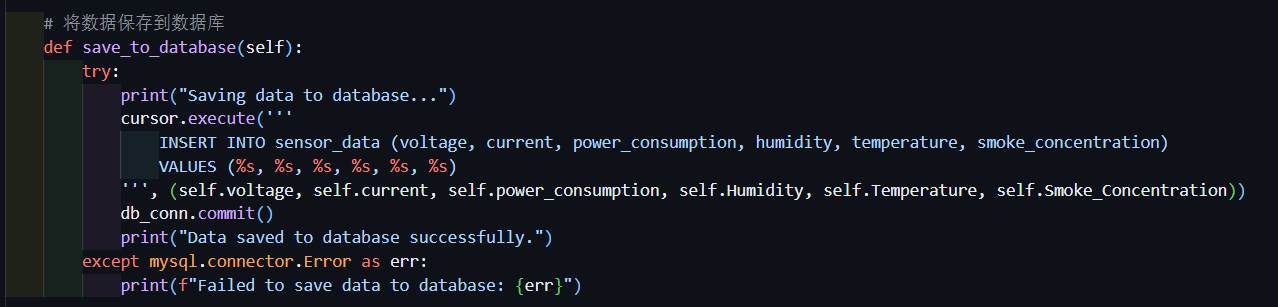
* + 1. 数据库连接与保存

从机传输过来的数据保存方式由日志文件转变为数据库保存。

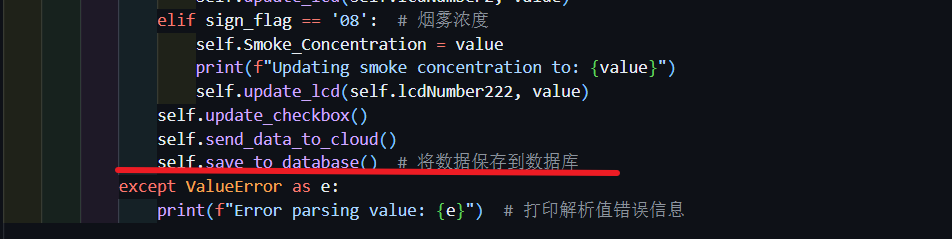
数据库连接代码如下：



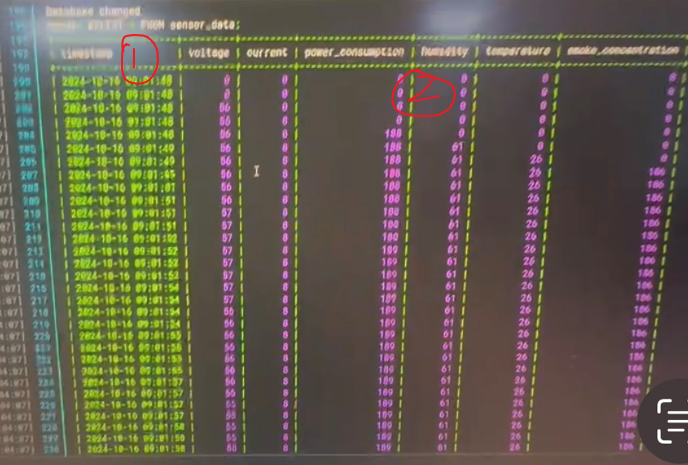
信息上传数据库函数：



上传函数的调用：



下面是数据库存储信息的展示（可以保存所有收到数据的字段）：



1. ①是时间戳：记录信息保存进来的时间
2. ②是字段信息，所有的数据