**中国海洋大学计算机科学与技术学院**

实验报告

**姓名：惠欣宇 年级：2021级 专业：计算机科学与技术**

**科目：物联网系统设计与开发 题目：电参表仿真设备交互编程**

**实验时间：2024年 4月 28日 实验教师：郭忠文**

**目录**

[一、实验目的及要求 2](#_Toc165832103)

[1. 实验目的 2](#_Toc165832104)

[2. 实验要求 2](#_Toc165832105)

[二、实验内容 2](#_Toc165832106)

[三、通信协议 2](#_Toc165832107)

[四、实验步骤 3](#_Toc165832108)

[五、报告要求 5](#_Toc165832109)

[1. 简要叙述实验结果 5](#_Toc165832110)

[2. 把采集程序源代码和存储数据的文本文件附在实验报告中 6](#_Toc165832111)

[3. 总结编写数据采集软件及软件调试的关键步骤 11](#_Toc165832112)

[4. 写出实验过程中遇到的问题和解决方法 11](#_Toc165832113)

[六、实验总结 12](#_Toc165832114)

一、实验目的及要求

1. 实验目的

通过编写电参表传感数据采集软件，培养利用串行通信接口获取传感数据的编程能力，为物联网系统综合数据采集软件的设计与开发打下基础。

2. 实验要求

根据实验内容，预先编写数据采集软件，实验过程中认真调试、思考和记录，掌握利用软件自动获取串行接口传感设备数据的方法，撰写实验报告。

二、实验内容

编写数据采集程序，通过 USB 转串口连接线获取 WT310 仿真传感设备的电压、电流、功率和耗电量传感数据，把解析后的数据写入桌面的文本文件中，文件以姓名加学号命名，每次采集的数据在文件中占据一行，每行的开头为采集数据的时间格式为“YYYY-MM-DD,hh:mm:ss”，如：2017-03-26，15:28:32。不同数据 项以“;”号分隔，共采集 10 次。

三、通信协议

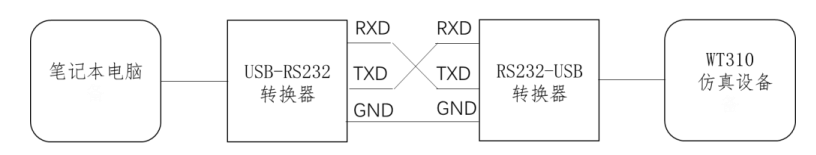
WT310 数据获取命令为**NUMeric:NORMal:VALue?**，命令后面以回车符+换行符（十六进制 0D 0A）为命令结束符。WT310 接收到命令后会返回测试数据，如：

**103.79E+00,1.0143E+00,105.27E+00,50.001E+00,655.27E+00,0.27E+00**

各参数以“,”作为分隔符，分别对应电压、电流、功率、频率、耗电量、功率因数，其数值以科学计数法方式表示（E 表示指数，E 前面的数值乘以 10 的 n 次幂，n为 E 后面三位字符串代表的数值，如：105.27E+00 表示 105.27\*10+00，仍然为 105.27）。

四、实验步骤

（1）WT310 仿真设备硬件连接如图 5.3.1 所示，根据该图将笔记本电脑通过 USB-232 转换器与 WT310 仿真设备进行硬件连接，USB-RS232 转换器外观结构如图 5.3.2 所示。



**图5.3.1** WT310 仿真设备硬件连接

图示

描述已自动生成

**图 5.3.2** USB-RS232 转换器外观结构

（2）打开仿真器桌面上的“琏雾实验系统\实验 5.3”文件夹，运行“wt310”快捷方式，仿真器屏幕显示 WT310 仿真设备，用鼠标点击仿真设备的“power”按钮(电源开关)，仿真设备开始工作，如图 5.3.3 所示：

电脑萤幕

中度可信度描述已自动生成

**图 5.3.3** 仿真 WT310 运行界面

点击 INTERFACE 键，进行通信参数设置，如图 5.3.4 所示，默认设置为数据格式：For0，波特率：9600，结束符：Cr+LF。数据格式有四种，For0(数据位 8；停止位 1；校验位 NONE)；For1(数据位 7；停止位 1；校验位 ODD)；For2(数据位 7；停止位 1；校验位 EVEN)；For3(数据位 7；停止位 2；校验位 NONE)。波特率可以选择 1200、2400、4800、9600、19200、38400 六种。命令结束符有三种，Cr+LF（回车加换行）；Cr（回车）；LF（换行）。

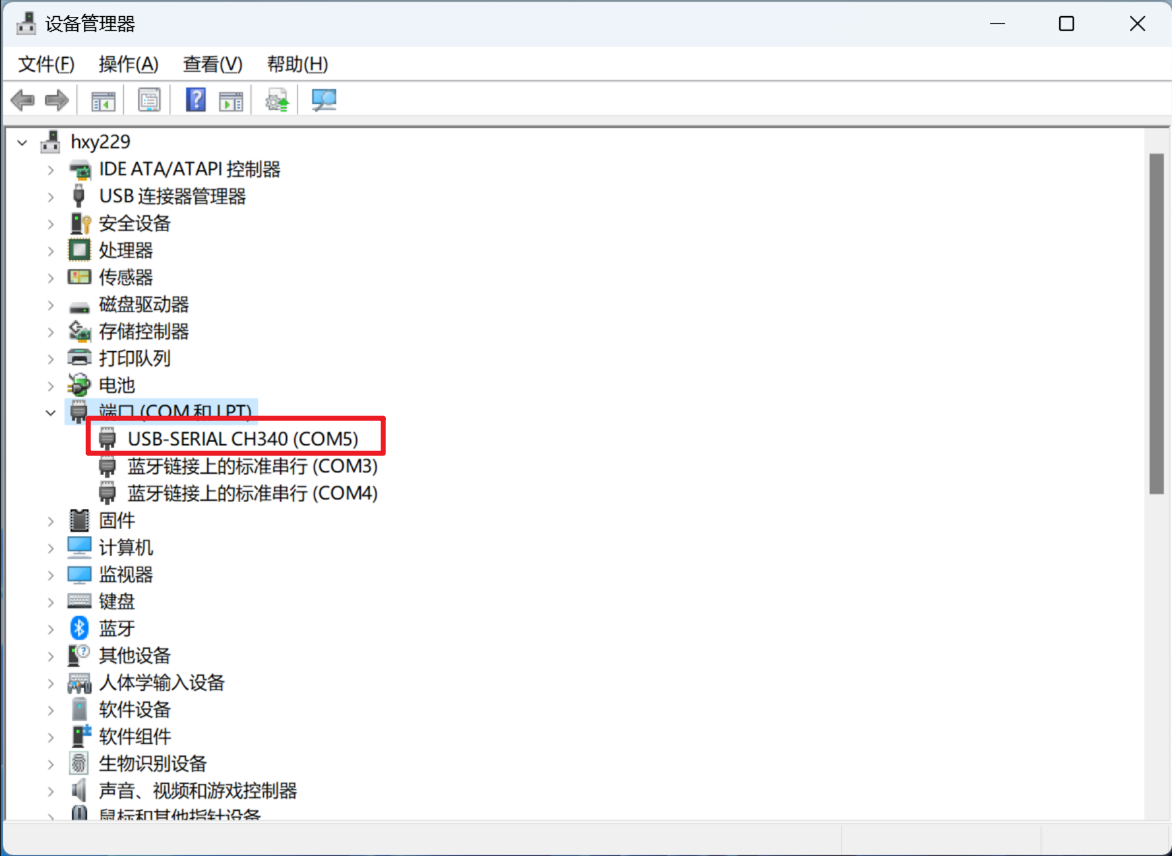
图片包含 日历

描述已自动生成

**图 5.3.4** 仿真 WT310 通信参数设置界面

（3）打开计算机桌面上的“琏雾实验系统\工具”文件夹中的串口助手软件进行通信调试，确保硬件通信线路正常。

这里我们还是按照串口助手的连接过程进行，将笔记本电脑通过 USB-232 转换器与 WT310 仿真设备进行硬件连接后，打开笔记本电脑设备管理器查看相应连接端口，如下图5.3.5所示，这里是COM5。



**图 5.3.5** 笔记本电脑上的端口信息

下面我们打开笔记本电脑上的串口助手进行测试，串口号选择COM5，其余参数与上述仿真 WT310 通信参数一致，然后打开串口。在命令发送区发送命令**NUMeric:NORMal:VALue?**，WT310会给我们返回一组数据**2.1970E+02,4.3300E-01,4.8200E+01,4.2170E+03,5.0000E+01,5.0000E-01**，这说明我们的硬件通信线路是正常的。测试如下图5.3.6所示。

图形用户界面, 文本, 应用程序, Word

描述已自动生成

**图 5.3.6** 测试确保硬件通信线路正常

（4）对根据实验内容编写的数据采集程序进行调试。【数据采集程序放在报告要求的第二部分解释】

五、报告要求

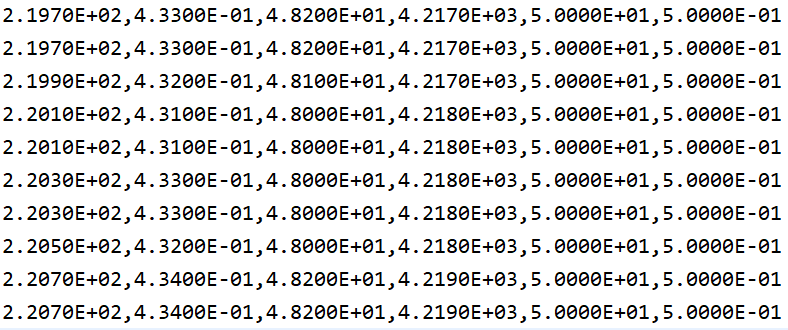
1. 简要叙述实验结果

测试阶段：

使用串口助手对串口通信功能进行测试，在上述图5.3.6中即是测试成功的串口助手返回结果，这个过程就是：使用笔记本电脑串口助手向WT310仿真设备发送命令**NUMeric:NORMal:VALue?**，WT310会给我们返回一组数据**2.1970E+02,4.3300E-01,4.8200E+01,4.2170E+03,5.0000E+01,5.0000E-01**，这些数据表示的是WT310 仿真传感设备的电压、电流、功率和耗电量传感数据，我们把这样一个过程称为一次数据采集。我们的实验就是需要复现这一过程，写代码完成十次的数据采集。

实验阶段：

实现代码并配置好串口相应参数后向WT310仿真设备循环发送10次命令**NUMeric:NORMal:VALue?，**在笔记本上会获得10组返回数据如下图5.3.7：



**图 5.3.7** WT310仿真设备返回数据结果

然后将这些数据按照列分别解析为电压、电流、功率、耗电量、频率和功率因数，其中每一行解析开头需要插入解析时间，解析结果如下图5.3.8：



**图 5.3.8** 数据解析结果

2. 把采集程序源代码和存储数据的文本文件附在实验报告中

这里不仅给出**完整代码**，同时在每一个模块之后给出相应的模块**功能解释**。

首先导入三个包：serial、datetime和tkinter。serial包用于在Python中实现串口通信，提供了一种在计算机和外部设备之间进行串行通信的方式。可以打开串口、设置串口参数、发送和接收数据等。datetime包用于处理日期和时间。本实验需要使用这个包获取当前日期时间与格式化日期时间。tkinter包用于创建GUI的工具包，提供了创建窗口、按钮、标签、文本框等GUI组件的方法，以及管理事件和布局的机制。

1. import serial  # 串口模块
2. import datetime
3. import tkinter as tk

下面的代码创建了一个GUI主窗口，并设置了窗口的标题为"数据采集"。然后分别创建了两个框架raw\_frame和parsed\_frame，用于容纳原始数据和解析后的数据。这两个框架都是放置在主窗口root上的。在raw\_frame框架中，创建了一个文本框raw\_data\_textbox，用于显示原始数据。在主窗口root上直接创建一个文本框data\_textbox，用于显示解析后的数据。在每个框架中都创建了一个标签，用于显示相应数据的描述。raw\_label显示"WT310返回数据："，parsed\_label显示"解析后的数据："。

1. root = tk.Tk()  # 创建GUI主窗口
2. root.title("数据采集")  # 设置主窗口标题为"数据采集"
3. # 创建框架来显示原始数据和解析后的数据
4. raw\_frame = tk.Frame(root)  # 创建框架用于容纳原始数据
5. raw\_frame.pack(pady=5)
6. parsed\_frame = tk.Frame(root)  # 创建框架用于容纳解析后的数据
7. parsed\_frame.pack(pady=5)
8. # 原始数据文本框
9. raw\_label = tk.Label(raw\_frame, text="WT310返回数据：")
10. raw\_label.grid(row=0, column=0)
11. raw\_data\_textbox = tk.Text(raw\_frame, height=11, width=113)  # 创建文本框，用于显示原始数据
12. raw\_data\_textbox.grid(row=1, column=0)
13. # 解析后的数据文本框
14. parsed\_label = tk.Label(parsed\_frame, text="解析后的数据：")
15. parsed\_label.grid(row=0, column=0)
16. data\_textbox = tk.Text(root, height=11, width=113)  # 创建文本框，用于显示解析后的数据
17. data\_textbox.pack()

接下来配置串口通信参数。经过设备管理器端口信息检查发现与WT310仿真设备进行连接的端口是COM5，于是我们打开COM5串口，并设置波特率为9600，超时时间为1秒。这里设置超时时间为1秒是为了防止读取数据的时候发生无限等待。然后根据报告的要求定义一个超参数COLLECTIONS\_TIMES，值为10，表示数据采集10次。

1. # 串口通信设置
2. ser = serial.Serial('COM5', 9600, timeout=1)  # 打开串口'COM5'，波特率设置为9600，超时时间设置为1秒
3. # 定义采集次数的超参数，报告中要求采集10次
4. COLLECTIONS\_TIMES = 10

下面是函数**update\_data()**，他从串口中获取数据并更新GUI。函数首先检查采集次数是否达到预设上限COLLECTIONS\_TIMES，如果未达到上限则继续执行数据采集操作。接着，它调用get\_data() 函数从串口获取原始数据，并将原始数据显示在GUI界面上的原始数据文本框中。然后调用parse\_data() 函数对原始数据进行解析，然后调用save\_data() 函数将解析后的数据与原始数据进行保存，并将解析后的数据显示在GUI界面上的解析数据文本框中。再通过data\_textbox.see(tk.END)确保解析后的数据文本框滚动到末尾，以显示最新的解析数据。最后，通过递归调用自身，以1秒的时间间隔进行连续的数据采集，同时更新计数器来对采集次数进行累加。当采集次数达到预设上限时，关闭串口通信，结束数据采集过程。

1. def update\_data(count):
2. if count <= COLLECTIONS\_TIMES:  # 当计数小于等于10时执行以下操作
3. data = get\_data()  # 获取数据
4. raw\_data\_textbox.insert(tk.END, data + '\n')  # 将原始数据插入原始数据文本框中
5. parsed\_data = parse\_data(data)  # 解析数据
6. save\_data(parsed\_data, data)  # 保存原始数据与解析数据
7. data\_textbox.insert(tk.END, parsed\_data + '\n')  # 将解析后的数据插入解析后的数据文本框中
8. data\_textbox.see(tk.END)  # 滚动解析后的数据文本框到末尾
9. root.after(1000, update\_data, count + 1)  # 1秒后递归调用update\_data函数，计数加1
10. else:
11. ser.close()  # 当达到10次后关闭串口

在函数update\_data() 中调用了函数**get\_data()**，那么下面来解释get\_data()函数的功能，它用于从串口获取数据。首先，它通过ser.write(b'NUMeric:NORMal:VALue?\r\n')向串口发送命令NUMeric:NORMal:VALue?，以请求数据。然后通过ser.readline() 从串口读取返回的数据，并使用decode('utf-8')方法将其解码为UTF-8格式的字符串，再通过strip() 方法去除字符串两侧的空白字符。最后将获取到的数据返回给update\_data()。

1. # 从串口获取数据的函数
2. def get\_data():
3. ser.write(b'NUMeric:NORMal:VALue?\r\n')  # 发送命令获取数据
4. response = ser.readline().decode('utf-8').strip()  # 从串口读取数据并解码为UTF-8格式的字符串，去除两侧空白字符
5. return response  # 返回获取的数据

在函数update\_data() 中调用了函数**parse\_data()**，下面解释一下**parse\_data()**函数的功能，它用于解析从串口获取的原始数据。首先使用try语句块包裹解析过程，这样就可以随时捕获解析过程中可能发生的异常。然后从串口获取的原始数据并按逗号分割，同时将分割后的字符串列表中的每个元素转换为浮点数。这样就得到了电压、电流、功率、耗电量、频率和功率因数等数据。然后使用datetime模块获取当前的日期时间，并使用strftime('%Y-%m-%d, %H:%M:%S')方法将其格式化为年-月-日, 时:分:秒的字符串格式。最后将解析后的各项数据与对应的单位以及当前时间进行格式化，并组合成一个字符串，表示解析后的数据。最后，将解析后的数据返回给update\_data()。如果在解析数据时发生错误（例如无法将字符串转换为浮点数），则会捕获ValueError异常，并返回一个表示解析错误的字符串信息"解析数据时出错"。

1. # 解析数据的函数
2. def parse\_data(data):
3. try:  # 解析数据
4. voltage, current, power, consumption, frequency, power\_factor = map(float, data.split(','))  # 将数据按逗号分割并转换为浮点数
5. current\_time = datetime.datetime.now().strftime('%Y-%m-%d, %H:%M:%S') # 获当前时间
6. parsed\_data = (
7. f"{current\_time}: 电压: {voltage} V; 电流: {current} A; 功率: {power} W; 耗电量: {consumption} Wh; 频率: {frequency} Hz; 功率因数: {power\_factor}"
8. )  # 格式化解析后的数据
9. return parsed\_data  # 返回解析后的数据
10. except ValueError:  # 如果解析错误
11. return "解析数据时出错."  # 返回错误信息

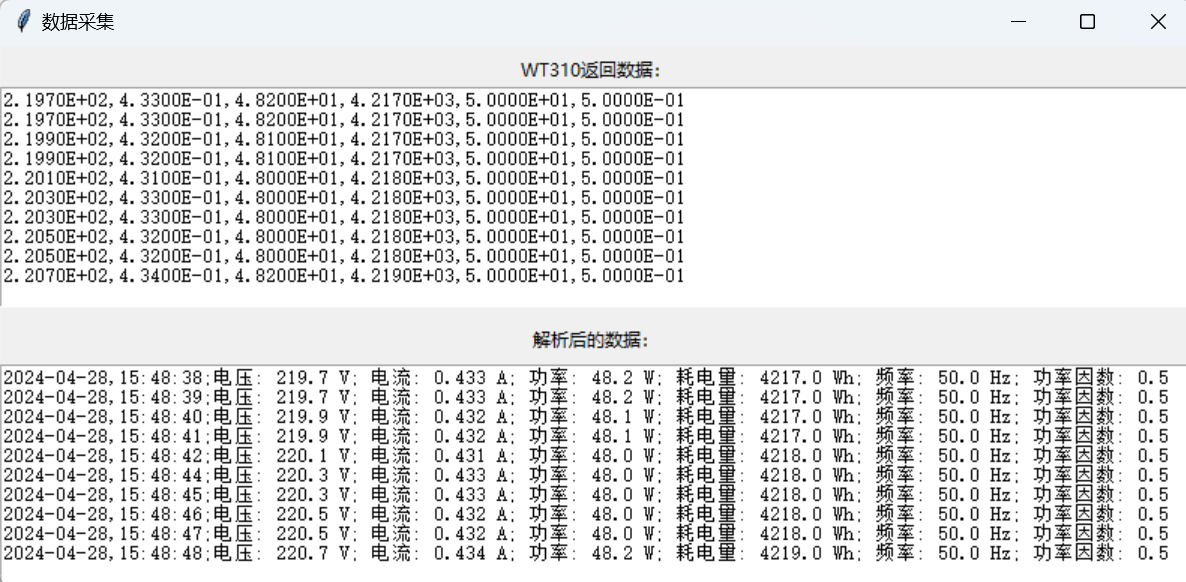
在函数update\_data() 中调用了函数**save\_data(parsed\_data, raw\_data)**，下面解释一下save\_data(parsed\_data, raw\_data) 函数的功能，他主要是将原始数据和解析后的数据保存到文件中。首先使用try语句块包裹保存数据的操作，这样就可以随时捕获解析过程中可能发生的异常。然后打开名为raw\_data.txt的文件，将原始数据raw\_data写入文件中。然后，它打开名为21030031009.txt的文件，将解析后的数据parsed\_data写入文件中。最后，如果在保存数据到文件时发生错误（例如文件操作权限问题），则会捕获ValueError异常，并将表示错误信息的字符串"解析数据时出错"写入名为21030031009.txt的文件中。

1. # 保存数据到文件的函数
2. def save\_data(parsed\_data, raw\_data):
3. try:  # 保存数据
4. with open('raw\_data.txt', 'a') as file:
5. file.write(f"{raw\_data}\n")  # 写入原始数据
6. with open('21030031009.txt', 'a') as file:
7. file.write(f"{parsed\_data}\n")  # 写入解析后的数据
8. except ValueError:  # 如果保存错误
9. with open('21030031009.txt', 'a') as file:
10. file.write("解析数据时出错.\n")  # 写入错误信息

最后直接调用update\_data() 函数执行完整过程的模拟，这里需要传入参数1，因为在update\_data() 函数内部是通过传入参数的增长来实现对采集次数的计算，当采集次数达到预设参数10次后关闭串口。

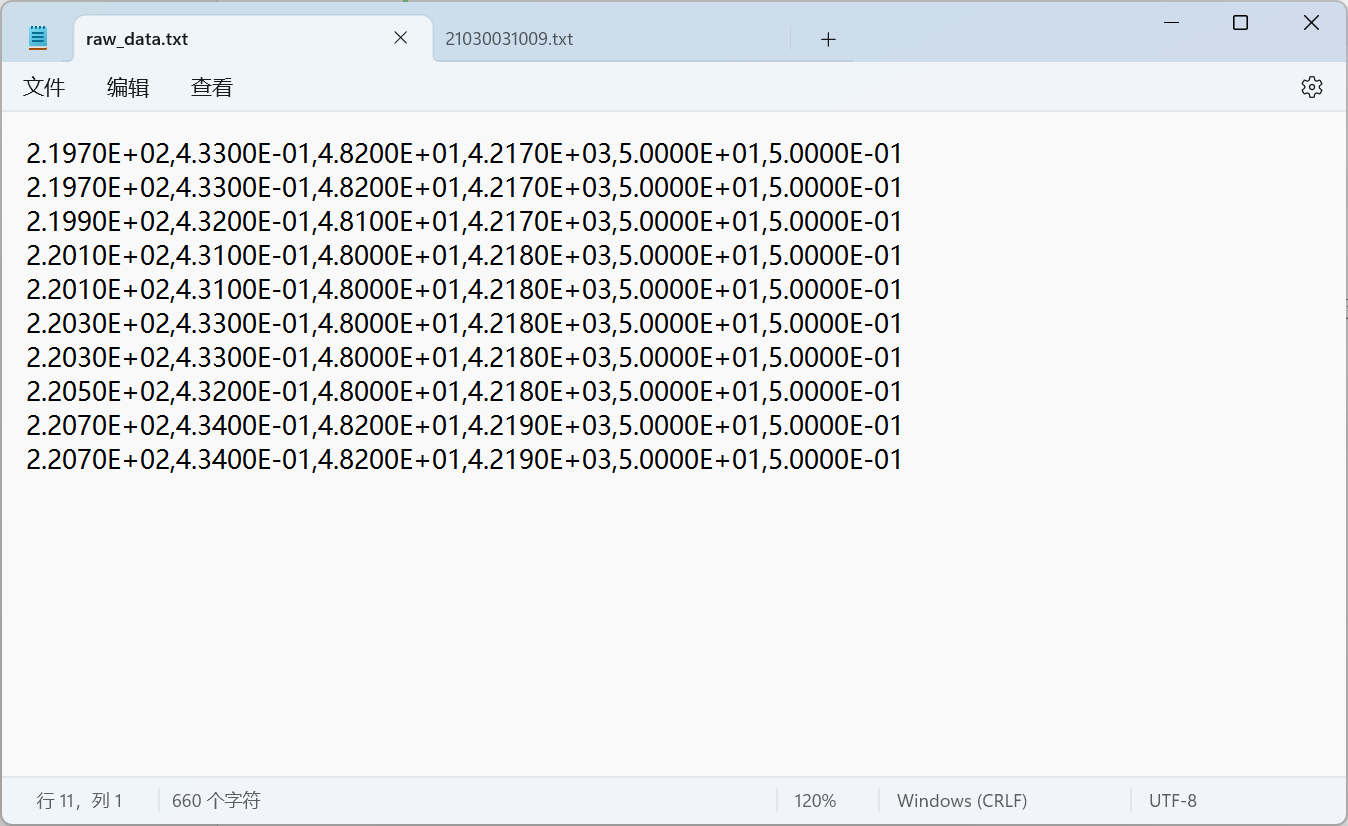
1. # 开始数据采集
2. update\_data(1)  # 从1开始计数
3. root.mainloop()

上述代码实现了一个可视化界面来进行命令发送和数据解析，下面是相应可视化界面，主要展示解析过程的结果，如下图5.3.9所示。

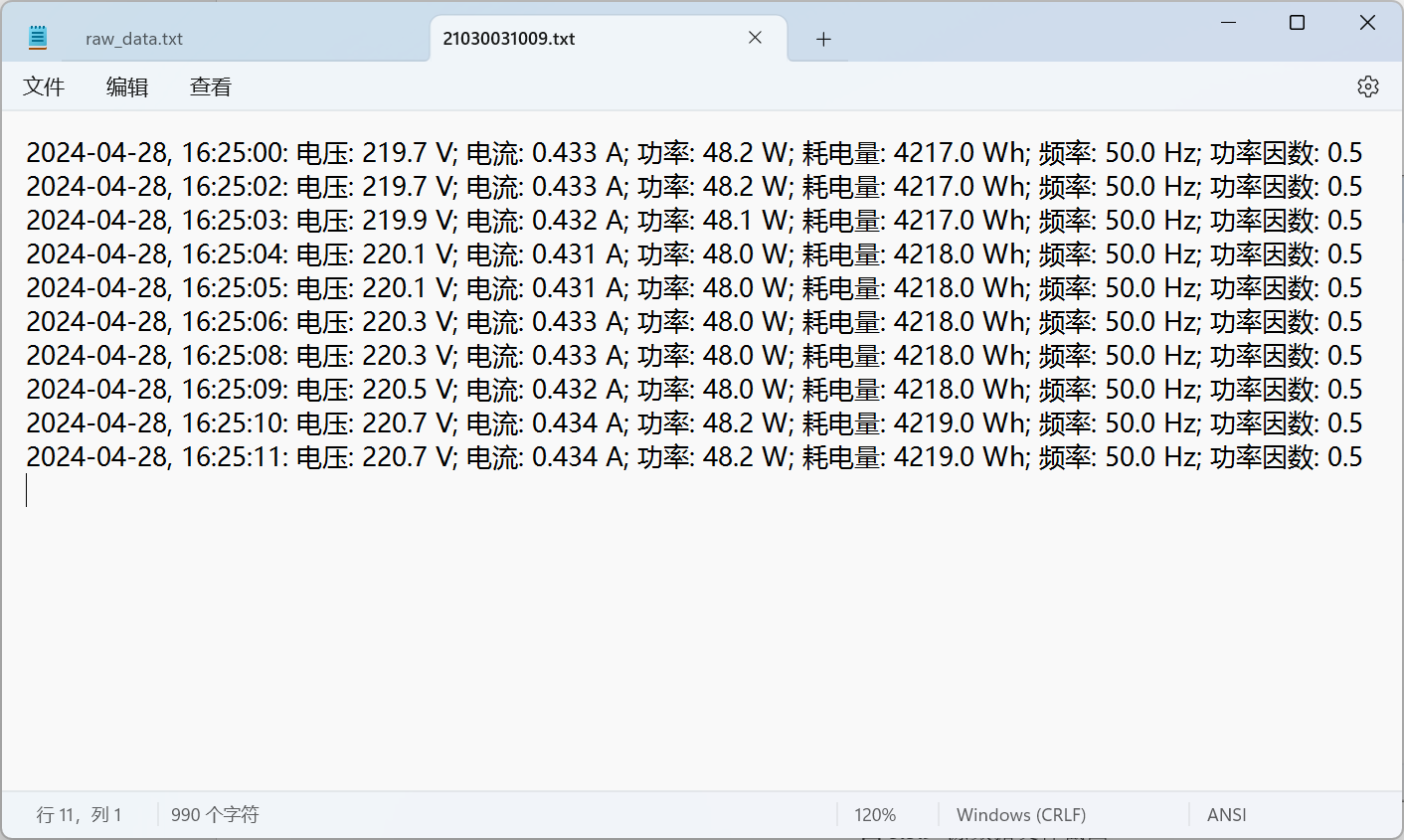


**图5.3.9** 源数据文件截图

下面给出存储数据的文本文件raw\_data.txt与21030031009.txt，分别表示源数据文件与解析后数据文件，如下图5.3.10与5.3.11所示。



**图5.3.10** 源数据文件截图



**图5.3.11** 解析数据文件截图

3. 总结编写数据采集软件及软件调试的关键步骤

* **关键步骤1——获取数据**

本次实验中第一个较为关键的步骤就是如何模仿串口助手进行命令的发送与数据的获取，经过上网查询Python有关串口操作的软件包我们发现serial包提供了相关串口操作的方法，在代码中相关的部分即为get\_data() 函数。get\_data() 函数从串口获取原始数据。原始数据是数据采集过程的开始，它包含从WT310仿真设备获取的实时数据。因此获取准确的原始数据对于后续的数据处理和分析至关重要。

* **关键步骤2——解析数据**

在进行数据的获取之后，就需要按照实验手册要求来对返回数据进行解析，在代码中相关部分即为parse\_data() 函数。parse\_data() 函数负责解析原始数据，将其转换为可理解和可处理的格式。通过解析原始数据，我们可以得到有用的信息，如电压、电流、功率、耗电量、频率和功率因数等。这些信息可以用于监控设备状态、分析性能指标等，对本次实验得到正确结果至关重要。

* **关键步骤3——保存数据**

在进行数据的解析之后，就需要按照实验手册要求来对解析数据进行保存，在代码中相关部分即为save\_data() 函数。save\_data() 函数将解析后的数据和原始数据保存到文件中。数据持久化存储对于后续的数据分析以及历史数据回溯有很大的意义。通过将数据保存到文件中，可以确保数据的完整性和可靠性，也完成了相应实验要求。

4. 写出实验过程中遇到的问题和解决方法

* **问题及解决方法1**

**问题：**在进行串口助手测试时测试总是不成功，检测到端口但是端口无法使用，导致不能得到仿真设备的数据返回。

**解决方法：**重新检查了端口的占用情况，最后发现是在之前串口助手实验时我自行创建了一对虚拟端口COM5与COM6，这里的COM5正好与实验中与WT310仿真设备相连的COM5冲突，也就是虚拟端口占用了COM5，导致本次实验的测试不能成功。最后在Configure Virtual Serial Port Driver中删除了先前创建的COM5与COM6端口，再次进行测试成功。

六、实验总结

通过本次电参表仿真设备交互编程实验，我成功地完成了电参表传感数据采集软件的编写，并掌握了利用串行通信接口获取传感数据的编程技能。实验过程中完成了硬件连接、软件调试、数据采集和解析等步骤。

实验中较为重要的三个步骤：获取数据、解析数据和保存数据也做了相应的分析。通过使用与学习Python的serial包，实现了与WT310仿真设备之间的有效的串口通信，并成功获取了传感数据。在进行数据解析时，通过编写的parse\_data()函数，将原始数据转换为实验手册中要求的格式，其中包括电压、电流、功率、耗电量、频率以及功率因数，也学会了如何正确处理科学计数法表示的数据，这对于数据分析和处理非常关键。最后使用save\_data()函数将解析后的数据和原始数据保存到了文件中，实现了数据的持久化存储。

在实验过程中，我遇到了串口助手测试不成功的问题，原因是之前创建的虚拟端口与实验中使用的COM5端口发生了冲突。通过删除虚拟端口并重新进行测试，也是成功解决了这个问题。

除了实验手册中的要求我还实现了解析过程的可视化，增添了解析过程的交互性。

通过本次实验，我不仅提升了编程能力，还加深了对物联网系统设计与开发的理解。我认识到了在实际应用中，确保硬件通信线路正常和正确配置串口参数的重要性。