**实验报告**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 学 号 | 21030031009 | | 姓 名 | 惠欣宇 | 专业班级 | | 计算机4班 | |
| 课程名称 | **物联网系统设计与开发** | | | | 学期 | | **2024年春季学期** | |
| 任课教师 | 郭忠文 | | 完成日期 | 2024.3.21 | | 实验课时间 | | 10:25~12:15 |
| 实验名称 | | GM10 采集器数据解析 | | | | | | |
| **一、实验目的及要求**  目的：根据 GM10 传感设备，理解 GM10 传感设备的通信协议，通过传感设备的通信协议解析出返回的数据，掌握字符型通信协议解析过程，为后续的传感设备数据采集软件编程打下基础。  要求：预习 GM10 传感设备的通信协议，通过实验，熟练掌握根据 GM10 传感设备的通信协议解析数据的方法。  **二、实验内容及步骤**  内容：根据提供的 GM10 传感设备通信协议，对文件中的 GM10 返回数据进行解析和显示。  步骤：  （1）学习 GM10 传感设备的通信协议。  （2）打开仿真器桌面上的“琏雾实验系统\实验 2.1”文件夹，利用 Python 语言编写程序，读取 gm10data.txt 文件中的数据，根据 GM10 通信协议对文件中的数据进行解析和显示。GM10 解析数据显示示例如图 1所示：    **图1** GM10 解析数据格式示例  **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 代码实现思路 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***  代码主要分为两个部分：**数据解析**和**数据显示**。  **数据解析详细步骤**：  ①读取文件内容：   * 使用with open(file\_path, 'r') as file:确保文件在读取后能够正确关闭。 * 通过file.readlines()将文件的所有行读取到一个列表lines中，其中每个元素都是文件中的一行文本。   ②初始化变量：   * 创建一个空字典data用于存储每个通道的温度数据。   ③逐行解析数据：   * 对于每一行数据，根据其开头的字符来判断其类型，并进行相应的处理： * 如果以"DATE"开头，表示这是一行日期数据。使用split()方法分割字符串，并提取日期部分。 * 如果以"TIME"开头，表示这是一行时间数据。同样使用split()方法提取时间部分。 * 如果以"N"开头，表示这是一行包含传感器数据的标准数据。进一步处理这行数据： * 使用split()方法将行内容分割成多个部分。 * 提取通道编号，指数符号，尾数和指数。 * 根据指数计算实际的温度值。首先找到尾数和指数之间的"E"字符，然后根据指数的值计算出科学计数法表示的温度值。如果指数为负数，则温度值为尾数除以10的指数次幂；如果为正数，则为尾数乘以10的指数次幂。 * 将计算出的温度值四舍五入到一位小数，得到最终的温度值。 * 将当前通道编号和对应温度值存储到data字典中。   ④返回解析结果：   * 函数parse\_gm10\_data最终返回日期、时间和解析后的温度数据字典。   根据上述**数据解析**的思路及步骤来进行代码实现，代码如下图2所示。    **图2** parse\_gm10\_data函数具体实现  **数据显示详细步骤：**  1. 对话框部分  初始化一个Tkinter窗口，并为这个对话框添加了一张背景图片。Tkinter窗口中包含两个按钮（选择文件和另存为），分别用于打开文件选择对话框和保存数据，窗口下方还有两个文本框，左侧为文本框用于显示源数据，右侧文本框用于显示解析后的数据，同时这两个文本框分别配有一个滚动条。代码如下图3、4、5、6、7所示。    **图3** Tkinter窗口初始化    **图4** 添加Tkinter窗口背景    **图5** Tkinter窗口按钮部分    **图6** 源数据文本框和滚动条部分    **图7** 解析数据文本框和滚动条部分  2. 数据显示逻辑部分  ①选择文件：   * 点击“选择文件”按钮时，调用display\_gm10\_data,使用filedialog.askopenfilename()打开文件选择对话框，只允许选择一个文本文件。 * 一旦文件被选中，使用parse\_gm10\_data函数解析文件路径中的数据。   **display\_gm10\_data函数**代码如下图8所示。    **图8** display\_gm10\_data函数具体实现  ②更新显示：   * 解析数据后，首先配置源数据文本框source\_data\_text和解析数据文本框result\_text为可编辑状态，不然无法写入。 * 清空文本框的当前内容，使用delete('1.0', tk.END)，也就是清除上一次解析的数据。 * 插入新的解析结果，包括日期时间信息和每个通道的温度数据。 * 最后，将文本框状态重新设置为DISABLED，即不可编辑状态。   ③保存数据：   * 点击“另存为”按钮时，调用save\_data函数，使用filedialog.asksaveasfilename()打开保存文件对话框，可以选择保存位置和文件名。 * 选择后，将解析数据文本框中的内容（即解析后的数据）写入到指定的文件中。   **save\_data函数**代码如下图9所示。    **图9** save\_data函数具体实现  **这里需要注意一点**：“另存为”按钮起初需要设置为不可点击状态，因为最开始没有数据解析内容可以存储，只有在选择文件并解析之后，“另存为”按钮才进入可点击状态。  在实验的最后我使用Python的pyinstaller工具将本次gm10数据解析器项目打包成一个.exe可执行文件，同时使用另存为功能将所给的测试数据的解析结果输出到一个.txt文件中。    **图10** gm10数据解析器未选择数据源的运行效果    **图11** gm10数据解析器解析测试结果  如下图12所示，图12中的第1个文件为实验所给的与源数据文件。第2个图标为GM10数据解析器项目打包成的.exe可执行文件，双击即可展示图10的效果。第3个图标为经解析后并将解析结果另存的.txt文件，我们可以看看这个文件的内容，如图13。    **图12** 项目打包及数据文件    **图13** 使用另存功能存储的解析结果文件  **三、心得总结**  通过本次GM10采集器数据解析实验，我不仅加深了对物联网系统中数据采集和解析的理解，而且提升了我的编程能力和问题解决技巧。在实验过程中，我学习了GM10传感设备的通信协议，并成功实现了一个数据解析器，用于读取和解析传感器数据。  在进行数据解析的过程中，我们首先要对通信协议要有深入的理解。GM10传感设备的通信协议规定了数据的格式和解析规则，只有准确把握这些规则，才能正确地从原始数据中提取有用信息。对于任何物联网设备，理解其通信协议是进行有效数据交互的前提。  在编写数据解析代码时，我首先面临的是如何高效地读取和处理大量数据。我采用了逐行读取文件的方式，并在读取每一行时立即进行解析，这样做的好处是可以处理大型文件而不会消耗过多内存。这种方法也使得代码更加模块化，易于调试和维护。  在解析具体数据时，我遇到了处理科学计数法表示的温度值的问题。结合科学计数法的表示规则以及Python中的字符串处理方法，最终从字符串中提取有效数字，并将其转换为实际的温度值。提高了我的编程技巧，尤其是在处理复杂数值方面。  最后，通过本次实验，我更加深刻地认识到了调试的重要性。在编程过程中，我使用了Python的调试工具来逐步执行代码，观察变量的变化，快速定位并解决了多个问题。  通过完成本次GM10数据解析实验，同时提高了我的在数据处理方面的编程能力，也锻炼了我的逻辑思维和问题解决能力。通过这次实验，我更有兴趣面对物联网系统中的数据解析任务，并且为未来物联网课程的学习打下一定的基础。  **遇到的问题及解决方法的总结：**  1.文件读取与处理  在读取文件时，使用with open语句来自动管理文件资源，避免内存泄漏。如果遇到文件不存在或路径错误的问题，可以通过异常处理来提示用户，并允许重新输入文件路径。解决代码如下图14。    **图14** 解决文件读取与处理问题的代码块  2.数据格式解析  对于数据格式的解析，首先需要准确理解GM10传感设备的通信协议。在代码实现中，通过字符串操作（主要是split()方法）来分离数据字段。解决代码如下图15。    **图15** 解决数据格式解析问题的代码块  3.科学计数法温度值计算  对于科学计数法表示的温度值，需要写一个专门的转换公式来处理指数和尾数。这个公式应该能够识别指数的正负，计算出实际的温度值，并进行四舍五入处理。解决代码如下图16。    **图16** 解决科学计数法温度值计算问题的代码块 | | | | | | | | |