

東南大學

毕业设计(论文)报告

题目 海水淡化设备相关协议解析及数据转发软件设计

院(系) 自动化学院

专业 自动化

学号 08014233

学生姓名 刘页恺

指导教师 黄永明

东南大学毕业（设计）论文独创性声明

本人声明所呈交的毕业（设计）论文是我个人在导师指导下进行的研究工作及取得的科研成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得东南大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

论文作者签名：_____ 日期：_____年____月____日

东南大学毕业（设计）论文使用授权声明

东南大学有权保留本人所送交毕业（设计）论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括刊登）论文的全部或部分内容。论文的公布（包括刊登）授权东南大学教务处办理。

论文作者签名：_____ 导师签名：_____
日期：_____年____月____日 日期：_____年____月____日

摘 要

近年来随着全球人口的飞速增长以及人类活动对淡水资源的不断破坏，淡水的短缺已成为影响人类可持续发展的重要因素，世界上有近 20 亿人口还处于缺乏淡水资源的状态。在我国，淡水资源短缺的形势也不容乐观，我国人均淡水拥有量还不及世界平均水平的百分之三十。而地球上的海水资源十分丰富，利用海水淡化系统将大量的海水转变为可以供人饮用的淡水无疑可以大大缓解淡水资源匮乏的问题。

本研究所设计的“海水淡化系统相关设备数据协议解析及转发软件”，正是用于海水淡化系统设备的数据监控和转发工作，可以帮助工作人员方便地在工控机界面或者后台服务器观察海水淡化系统各设备的工作参数运行状况，从而维护海水淡化系统的正常运行。为此本研究拟设计出一款用于工控机（IPC）的相关数据解析、展示及转发软件，该软件根据实际情况和需求对海水淡化系统相关设备（如风力发电机组，电池管理系统，交直流能量调控装置，反渗透海水淡化装置，微网系统电控柜硬件点表等）与工控机之间相互传输的信息照各相关协议进行解析，得到各个设备的各项工作状态参数，将解析得到的各项有效数据通过软件的图形用户界面（GUI）显示出来，同时软件还会按实际需求转发这些数据至后台服务器的数据库，以便于远程监控海水淡化系统各设备的工作状态。

本设计软件采用 Python3.6 编程分析用户所选择的 PCAPNG Wireshark 数据包文件，根据实际协议解析得到各项有效数据；并利用 QML 语言结合 Qt Quick 模块创建图形化用户界面；再通过 PyQt5 将 QML 设计的界面和 Python 应用程序结合起来，从而使得基于 QML 设计的图形用户界面能够动态及时地显示在 Python 脚本中进行的协议解析过程所得到的各项数据；同时软件程序会通过 pymysql 模块连接到已启动的 MySQL5.7.9 后台服务器 data_repost 中已建立的 equipment 数据库，并且按实际需求将解析得到的数据插入该后台服务器数据库的相应表中。该软件使用便捷，可以高效地对海水淡化系统各设备的数据进行协议解析得到有效数据并将这些数据动态地展示在图形用户界面中，还可以准确地完成数据的后台转发工作，为海水淡化系统各设备的监控和维护工作提供了有力的支持，提高了相关人员的工作效率，帮助整个海水淡化系统持续工作在正常状态。

关键词：海水淡化，数据协议解析，Python3+PyQt5+ QML，数据转发，MySQL 数据库

Abstract

In recent years, with the rapid growth of the global population and the continuous destruction of freshwater resources by human activities, the shortage of fresh water has become an important factor affecting the sustainable development of humankind. There are nearly 2 billion people in the world still lacking freshwater resources. In China, the issue of shortage of freshwater resources is not optimistic. China's per capita freshwater possession is still less than 30 percent of the world average. While the sea is rich in water resources on the earth, the use of seawater desalination systems to convert large quantities of seawater into freshwater for human consumption will undoubtedly greatly alleviate the shortage of freshwater resources.

The "Data and Protocol Analysis and Forwarding Software for Seawater Desalination System Equipment" designed in this study can be used to monitor and relay equipment data for seawater desalination systems. It can help operators to easily observe the operating parameters of equipment of the seawater desalination system at the IPC graphical user interface or backend server database to maintain the normal operation of the seawater desalination system. For this purpose, this study intends to design application software that can analyze, display and forward related data on the IPC. This software analyzes the network packet data transmitted between the equipment (such as wind turbines, battery management systems, AC and DC energy control devices, desalination equipment) and the IPC (industrial computer) according to the relevant protocols based on the actual situation and needs of the seawater desalination system. And the operating parameters extracted from the useful data analyzed from network packets are dynamically displayed through the software's GUI (graphical user interface). In the meantime, these data can be forwarded to the backend server database by the software according to actual needs. Therefore, it would be possible to monitor the operating status of seawater desalination equipment remotely.

The application uses Python 3.6 programming to analyze the PCAPNG Wireshark data packet, which can be selected by the user in the software interface, according to the network protocol analysis to obtain various valid parameters and takes advantage of the QML language combined with the Qt Quick module to design the graphical user interface, which is integrated with the data analysis program through PyQt5 to show the extracted data in GUI dynamically. At the same time, the application is connected to the already established system equipment database in the backend server of MySQL 5.7.9 through the pymysql module for inserting the valid parameters from the analyzed data into the corresponding tables of the MySQL server database according to actual needs. The software is easy to use and can efficiently analyze the data of each device of seawater desalination system to obtain valid parameters and display them in the graphical user interface in real-time. It can also accurately complete the work of relaying analyzed data to the backend server database. This software program provides strong support for the work of monitoring and maintaining the equipment of the seawater desalination system, which can effectively improve the working efficiency of operators and help the whole desalination system continue to work normally.

KEY WORDS: seawater desalination, data and protocol analysis, Python3+PyQt5+QML, data forwarding, MySQL database

目 录

摘 要	II
Abstract	III
第一章 绪论	1
1.1 研究背景	1
1.2 研究现状	2
1.3 研究内容	3
1.4 研究意义	2
第二章 总体设计	5
2.1 需求分析	5
2.1.1 总体功能需求	5
2.1.2 开发环境需求	6
2.2 设计工具及相关平台	6
2.2.1 Python3.6	6
2.2.2 JetBrains PyCharm 2018.1.3	7
2.2.3 PyQt5 工具集模块	7
2.2.4 Qt Quick2.5	8
2.2.5 MySQL5.7	8
2.2.6 MySQL 管理工具(Navicat for MySQL)	8
2.3 总体设计方案	8
2.4 总体设计方案流程图	9
2.5 软件总体架构设计	11
2.5.1 主程序文件	11
2.5.2 界面设计文件	11
2.5.3 辅助文件	12
2.5.4 MySQL 数据库表文件	12
2.5.5 网络数据封包文件	12
第三章 数据协议解析设计	13
3.1 协议解析设计思路	13
3.2 设计方案总结	20
3.3 协议解析核心代码设计	20
第四章 用户界面和数据展示软件设计	23
4.1 用户界面设计	23
4.1.1 Python3+PyQt5+ QML 设计界面	23
4.1.2 设计文件选择对话框	23
4.1.3 标签式界面设计	24
4.1.4 界面设计效果	25
4.2 数据展示软件设计	30
第五章 数据库和数据转发软件设计	31
5.1 数据库设计	31
5.1.1 设计方案	31

5.1.2	具体设计	31
5.2	数据转发软件设计	38
5.2.1	设计方案	38
5.2.2	具体设计	38
第六章	软件测试	43
6.1	协议解析和界面展示功能测试	43
6.2	数据转发功能测试	48
第七章	总结与展望	50
致 谢	51
参考文献	52

第一章 绪论

1.1 研究背景

近年来随着全球人口的飞速增长以及人类活动对淡水资源的不断破坏，淡水的短缺已成为影响人类可持续发展的重要因素，世界上有近 20 亿人口还处于缺乏淡水资源的状态。在中国，淡水资源短缺的情况并不乐观。而地球上的咸水资源十分丰富，全球近 96.5% 的水资源是咸水资源，如海水，数据显示，世界上有超过 42 亿人生活在距离海洋不到 80 公里的海域。因此，自 20 世纪下半叶以来，海水淡化一直被认为是连续提供淡水资源最有效的方法。与其他两种常见的淡水方法相比，在地下取水量和远程调水量方面，海水淡化消耗的能量最低，原水资源最为丰富，因此被认为是获取淡水的最可行和最经济的方式。而利用海水淡化系统将大量的海水转变为可以供人饮用的淡水无疑将大大缓解淡水资源匮乏的问题。错误!未找到引用源。

随着淡水资源的短缺，传统化石能源的减少和成本的增加，以可再生能源为驱动的海水淡化技术受到越来越多的关注。特别是在水电供应不足或小规模偏远地区。在社区中，可再生能源作为小型海水淡化厂的环保清洁能源的潜力近年来受到越来越多的关注。错误!未找到引用源。此外，传统化石燃料的燃烧将导致温室气体和有害物质的排放。因此，可再生能源的海水淡化也被认为是解决气候变化问题的潜在可行办法。目前，可再生能源的海水淡化主要包括太阳能，风能，地热能 and 海洋能。除能源外，利用核能作为海水淡化的可持续能源也被认为具有潜力以安全，经济和可持续的方式供应大量淡水。

海水淡化是解决能源短缺造成水资源短缺的重要手段之一。近年来，世界上许多国家在海水淡化，特别是海水淡化技术和风力发电技术方面取得了比较成熟的研究成果。世界范围内已经开展了一些风力海水淡化示范项目。

可再生能源风能海水淡化系统已逐渐引起人们的重视，特别是风能反渗透海水淡化技术已成为研究热点。反渗透海水淡化技术消耗电力，难以在远离大陆的能源不足的岛屿或地区普及。利用风能实现反渗透海水淡化是最成熟和工业化的可再生海水淡化方法。根据丹麦一家从事风电技术咨询的公司 btm 的统计，截至 2011 年，全球风电累计装机容量达到 199.5 吉瓦，中国装机容量达到 44.73 吉瓦。超过美国，居世界首位。中国沿海地区风能资源丰富，特别是东南沿海及其附近岛屿。利用中国沿海丰富的风能资源开发风能反渗透海水淡化系统具有很大的市场潜力^[1]。

风能脱盐技术主要用于将风能动能转化为电能，通过风机驱动动力设备的脱盐（脱盐淡化）。主要有两种实现方式：一种是风力涡轮机加海水淡化机器，两者均接入电网系统；二是将所产生的电能或转子叶片旋转产生的轴向功直接连接到脱盐系统上。不管采用哪种方式，脱盐系统的性能必将受风力发电机的影响，而自然风能不是持续均匀提供的，因此，发电量不均匀（由风能的不连续性和不稳定性引起），因此，整个系统还需要备用电源系统，如电池和储能双向转换器。

然而，风能脱盐系统使用各种各样的设备，并且连接复杂。只有当所有设备正常工

作时，整个海水淡化系统才能正常运行，生产出合格的淡水。因此，需要同时获取海水淡化系统中所有工作设备的全部相关运行状态参数。但是，系统中的设备类型和工作状态变量的数据格式都不同，所以它们传输到工控机的数据帧格式和使用的数据协议也有很大的不同。在这种情况下，需要为工控机设计一款协议解析软件，使得工控机能够准确地读取各设备传输的有效数据，并动态地显示在软件的人机交互的图形界面上，从而帮助操作人员和计算机掌控各设备的工作情况，为整个海水淡化系统的正常运行创造条件。

1.2 研究意义

淡水资源短缺已成为人类面临的普遍问题。仅中国每年因缺水造成的直接经济损失就达数千亿元。为解决淡水资源问题，各国在大力发展海水(苦咸水)淡化技术的同时，采取了多种节水和调水措施。目前，海水(苦咸水)淡化已成为解决淡水资源短缺的重要途径，特别是在中东和一些岛屿，海水淡化在当地经济社会发展中占有重要地位。但是海水淡化是一个能源密集型产业，而在淡水资源短缺的地方，特别是岛屿和偏远地区，电网基础也相对薄弱。同时，这些地区风能资源普遍丰富，考虑将风能直接用于海水淡化(苦咸水)，一方面可以解决淡水资源短缺，促进经济发展，另一方面可以利避免大规模并网风电对电网的影响^[2]。

所以在全球能源危机和环境恶化的影响下，基于风能开发的海水淡化技术在工程应用中得到迅速发展。风能等可再生能源直接应用于海水淡化已成为新能源利用技术的重要研究方向^[2]。根据风能利用方式的不同，风能反渗透海水淡化主要有两种方式：一是利用风能产生的机械能直接驱动反渗透装置水泵淡化海水，二是先将风能转化为电能，然后将电能用于驱动海水的脱盐过程。本课题设计的协议解析软件所用于的就是第二种风能海水淡化系统：风能通过两个风力发电机组转化为电能，存储在海水淡化系统的电池组内，电池组再利用储能双向变流器将直流电源转化为反渗透海水淡化装置所需要的交流电能，驱动海水淡化工作的进行。

而风能海水淡化系统中使用的设备种类繁多，联系复杂，只有各设备都工作在正常情况下，整个海水淡化系统才能正常运转，生产合格的淡水，所以需要同时得到海水淡化系统内各工作设备的所有相关运行状态参数，但是系统内各个设备的类型，工作状态变量的数据格式都不一样，因此它们向工控机传输的数据以及使用的数据协议大相径庭，在这种情况下，需要为工控机设计一款协议解析软件，使得工控机能够准确地读取各设备传输的有效数据，并动态地显示在软件的人机交互的图形界面上，从而帮助操作人员和计算机掌控各设备的工作情况，为整个风力海水淡化系统的正常运行创造条件。

本研究中所设计的“海水淡化系统相关设备数据协议解析及转发软件”，正是用于基于风能的海水淡化系统设备的数据监控和转发工作。该软件使用便捷，可以高效地对海水淡化系统各设备的数据进行协议解析得到有效数据并将这些数据动态地展示在图形用户界面中，还可以准确地完成数据的后台转发工作，因此能够帮助工作人员方便地在工

控机界面或者后台服务器观察海水淡化系统各设备的工作参数运行状况，从而为海水淡化系统各设备的监控和维护工作提供了有力的支持，提高了相关人员的工作效率，帮助整个海水淡化系统持续工作在正常状态。

1.3 研究现状

中国海水淡化技术的研究起步较早，也是世界上掌握先进风海水淡化技术的少数几个国家之一。主要研究机构是国家海洋局杭州水处理技术开发中心和国家海洋局天津海水淡化与综合利用研究所。此外，各大高校和研究机构也对海水淡化进行了研究。目前，随着多级水处理技术公司的出现，我国初步形成了全面的海水淡化市场。与此同时，外国海水淡化公司的进入也推动了中国海水淡化市场的发展。海水淡化市场蓬勃发展，但海水淡化的高能耗一直是制约海水淡化发展的根本原因^[3]。

在全球能源危机和环境恶化的影响下，风能开发和海水淡化技术在工程应用中得到迅速发展。将风能等可再生能源直接用于海水淡化已成为新能源利用技术的重要研究方向。目前，根据风能的不同使用方式，风能反渗透海水淡化系统主要可分为两种：第一，利用风能产生的机械能直接驱动反渗透装置水泵对海水进行淡化。第二是将风能转化为电能，然后将电力供应的海水淡化。本课题设计的协议解析软件所要研究的就是第二种风能海水淡化系统，风能通过两个风力发电机组转化为电能，存储在海水淡化系统的电池组内，电池组再利用储能双向变流器将直流电源转化为反渗透海水淡化装置所需要的交流电能，驱动海水淡化工作的进行。

现在，网络协议解析领域的研究中，做的很好的也是被广泛使用的一个协议解析器是 Wireshark，它不但能从网络中抓取数据封包，还具有很强的协议解析功能，其 tshark 软件可以直接打开网络数据包，读取封包中各条数据的信息并按相应的通讯协议传输帧格式将它们按功能字段分割成一片片数据展示，极大地简化了编程人员的协议解析工作。在研究中，海水淡化设备协议解析软件的核心部分——网络封包分析和数据协议解析模块的设计就是基于 Wireshark 强大的网络数据分析工具 tshark 实现的。

用于互联网嗅探的 Sniffer 在网络分析方面的研究也十分出色，作为便携式网络应用故障分析专家，无论在有线还是无线局域网中，Sniffer 都具有强大的数据包分析和捕获能力，随着各种有效方法的出现，基于这款软件的网络协议分析应用研究现在也已经很成熟了。

1.4 研究内容

本课题所研究的海水淡化系统各设备协议解析与数据转发软件是为了维护风力海水淡化系统的正常工作而设计的。为了保证该海水淡化系统工作在理想状态，需要设计出一款用于工控机（IPC）的相关数据解析、展示及转发软件，该软件根据实际情况和需

求对海水淡化系统相关设备（如风力发电机组，电池管理系统，交直流能量调控装置，海水淡化设备，反渗透海水淡化装置，微网系统电控柜硬件点表等）与工控机之间相互传输的信息照各相关协议进行解析，得到各个设备的各项工作参数状态，将解析得到的各项有效通过软件的图形用户界面（GUI）显示出来，同时软件还会按实际需求转发这些数据至后台服务器的数据库，以便于远程监控海水淡化系统各设备的工作状态。

本研究设计软件使用 Python3.6 编程分析用户所选择的 PCAPNG Wireshark 数据包文件,根据实际协议解析得到各项有效数据；并利用 QML 语言结合 Qt Quick 模块创建图形化用户界面；再通过 PyQt5 将 QML 设计的界面和 Python 应用程序结合起来，从而使得基于 QML 设计的图形用户界面能够动态及时地显示在 Python 脚本中进行的协议解析过程所得到的各项数据；同时软件程序会通过 pymysql 模块连接到已启动的 MySQL5.7.9 后台服务器 data_repost 中已建立的 equipment 数据库，并且按实际需求将解析得到的数据插入该后台服务器数据库的相应表中^[4]。

第二章 总体设计

2.1 需求分析

2.1.1 总体功能需求

应用本软件的风能海水淡化系统包含各种各样的设备，并且连接复杂。只有当所有设备正常工作时，整个海水淡化系统才能正常运行，生产出合格的淡水。因此，需要同时获取海水淡化系统中所有工作设备的所有相关运行状态参数。但是，系统中各设备类型和工作状态变量的数据格式都不相同，所以它们传输到工控机的数据帧格式和使用的传输协议也有很大的不同。在这种情况下，需要为工控机设计一款协议解析软件，使得工控机能够准确地读取各设备向工控机传输的有效数据，并动态地显示在软件的人机交互的图形界面上，从而帮助操作人员和计算机掌控各设备的工作情况，为整个海水淡化系统的正常运作创造条件。

为保证该海水淡化系统稳定运行，需要设计出一款用于工控机（IPC）的相关数据解析、展示及转发软件：该软件根据实际情况和需求对海水淡化系统内各相关设备（如风力发电机组，电池管理系统，交直流能量调控装置，反渗透海水淡化装置，微网系统电控柜硬件点表等）与工控机之间相互传输的信息按各相关协议进行解析，得到各个设备的各项工作状态参数，将解析得到的各项有效数据通过软件的图形用户界面（GUI）显示出来，同时软件还会按实际需求转发这些数据至后台服务器的数据库，以便于远程监控海水淡化系统各设备的工作状态。

该软件在协议解析方面的功能需求有：

- 1) 读取网络数据包中的数据
- 2) 按协议解析来自风力机组 1 的数据，得到该设备的各项工作状态参数
- 3) 按协议解析来自风力机组 2 的数据，得到该设备的各项工作状态参数
- 4) 按协议解析来自电池管理系统，得到该设备的各项工作状态参数
- 5) 按协议解析来自交直流能量调控装置的数据，得到该设备的各项工作状态参数
- 6) 按协议解析来自反渗透海水淡化装置的数据，得到该设备的各项工作状态参数
- 7) 按协议解析来自故障报警设备 1 的数据，得到该设备的各项工作状态参数
- 8) 按协议解析来自故障报警设备 2 的数据，得到该设备的各项工作状态参数
- 9) 按协议解析来自故障报警设备 3 的数据，得到该设备的各项工作状态参数

该软件在界面展示方面的功能需求有：

- 1) 生成文件选择对话框，供用户选择需要解析的网络数据包
- 2) 在图形用户界面动态地展示解析自风力机组 1 的数据
- 3) 在图形用户界面动态地展示解析自风力机组 2 的数据
- 4) 在图形用户界面动态地展示解析自电池管理系统的数据
- 5) 在图形用户界面动态地展示解析自交直流能量调控装置的数据
- 6) 在图形用户界面动态地展示解析自反渗透海水淡化装置的数据

- 7) 在图形用户界面动态地展示解析自故障报警设备 1 的数据
 - 8) 在图形用户界面动态地展示解析自故障报警设备 2 的数据
 - 9) 在图形用户界面动态地展示解析自故障报警设备 3 的数据
- 该软件在协议数据转发方面的功能需求有：

- 1) 新建 MySQL 服务器连接，在服务器中建立数据库
- 2) 在数据库中设计表 windturb1，用于转存解析自风电机组 1 的数据
- 3) 在数据库中设计表 adam1，用于转存解析自故障报警设备 1 的数据
- 4) 在数据库中设计表 adam2，用于转存解析自故障报警设备 2 的数据
- 5) 在数据库中设计表 adam3，用于转存解析自故障报警设备 3 的数据
- 6) 将解析自风电机组 1 的数据转发至数 MySQL 数据库中的表 windturb1
- 7) 将解析自故障报警设备 1 的数据转发至数 MySQL 数据库中的表 adam1
- 8) 将解析自故障报警设备 2 的数据转发至数 MySQL 数据库中的表 adam2
- 9) 将解析自故障报警设备 3 的数据转发至数 MySQL 数据库中的表 adam3

2.1.2 开发环境需求

硬件环境：PC 端

操作系统：Windows XP 及以上

软件环境：Python3.6 + Qt Quick2.5 + MySQL5.7

开发工具：Pycharm + Qt Creator4.5.1 + Navicat for MySQL

2.2 设计工具及相关平台

2.2.1 Python3.6

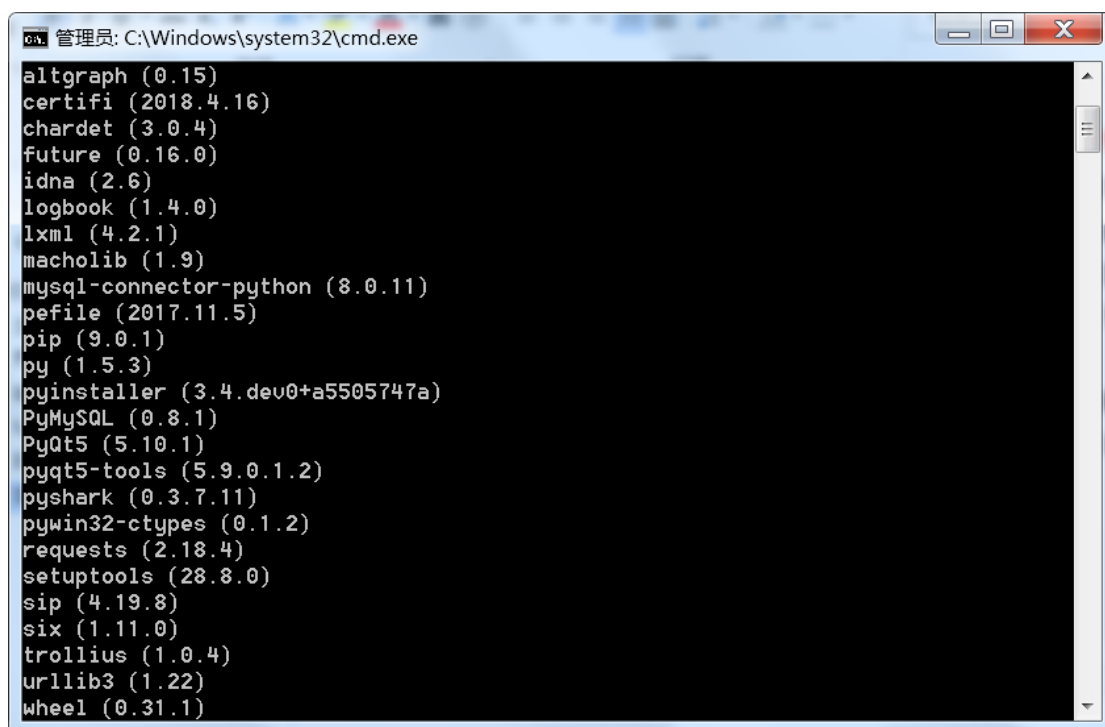
Python 是一种解释型编程语言，在各种平台下都具有良好的兼容性。本研究中所设计的“数据协议解析及展示转发软件”，就是用 Python3.6 编写的，其协议解析、数据展示和转发后台服务器的功能也都是基于 Python 语言实现的，且该软件可运行于安装了 Python 的 Windows 平台。在本研究中之所以选择设计基于 Windows 平台的 Python 软件编程，是因为考虑到以下两点：

(1) 本设计中所研究的海水淡化系统所位于的工业现场实际配置的工控机（IPC）使用的是 Windows7 操作系统

(2) Python 具有丰富且强大的标准库，基于 Python 语言可以使得实现软件各项功能的编程过程变得更为简单和高效^[5]。比如本研究中需要对从工控机上抓取的 .PCAPNG 网络数据包进行协议解析以得到工控机与海水淡化系统各个设备间传送的各项有效数据，而 Python 所提供的 PyShark 模块可以直接打开已经抓取的 PCAPNG 网络数据包并提供各种用于分析网络封包的函数接口，还可以调用 tshark 内置的各种数据包解码器，因此该 Python 模块库在用于分析 PCAPNG 时功能强大，十分好用，本设计中就是通过调用这个 PyShark 模块来帮助解析网络数据包的^[6]。除了 PyShark 之外，本软件设计中还使

用到了很多别的功能强大的 Python 模块，比如 tkinter 模块库，它是 Python3.x 自带的图形界面模块，可以调出文件选择对话框并直接返回用户所选择文件的路径。再比如 pymysql 模块，它可以帮助 python 程序连接到已建立的 mysql 服务器数据库，并对该数据库的各项数据内容进行插值，替换或者更新操作。大部分模块都可以在 py 文件头部用“import xxxx”的方式导入。在本设计中还用用到了一些第三方模块，由于 Python 没有自带这些模块，它们有些还需要安装后才能引用在软件程序中，不过安装方式也很简单，可以通过在命名窗口输入“pip install xxxx”进行安装然后使用这些模块来简化软件编写或者实现本软件需要的一些复杂功能。

需要在 Python 中安装的模块库：



```
管理员: C:\Windows\system32\cmd.exe
altgraph (0.15)
certifi (2018.4.16)
chardet (3.0.4)
future (0.16.0)
idna (2.6)
logbook (1.4.0)
lxml (4.2.1)
macholib (1.9)
mysql-connector-python (8.0.11)
pefile (2017.11.5)
pip (9.0.1)
py (1.5.3)
pyinstaller (3.4.dev0+a5505747a)
PyMySQL (0.8.1)
PyQt5 (5.10.1)
pyqt5-tools (5.9.0.1.2)
pyshark (0.3.7.11)
pywin32-ctypes (0.1.2)
requests (2.18.4)
setuptools (28.8.0)
sip (4.19.8)
six (1.11.0)
trollius (1.0.4)
urllib3 (1.22)
wheel (0.31.1)
```

图 2-1 需要在 Python 中安装的模块库

2.2.2 JetBrains PyCharm 2018.1.3

这是一款较为流行的使用方便的 IDE 软件，可以辅助 Python 编程，提高编程时的效率，因为 Python 编程对缩进的格式要求比 C 语言更加严谨，缩进格式差一点可能导致软件运行时不明所以地出错，所以在软件设计时，我使用了这个 Python IDE 辅助编程，更容易对缩进格式进行调整，让编程更加简单、高效。

2.2.3 PyQt5 工具集模块

PyQt5 内含丰富的与图形用户界面相关的 Python 模块，作为一个强大的工具集，它可以将 Qt 中用 QML 设计的界面和 Python 应用程序结合起来，从而可以在 Python 脚本中对相应的基于 QML 设计的图形用户界面进行操作，如更新、修改界面中某个数值。

在本软件设计中，就是通过它将 QML 设计的界面和 Python 应用程序结合起来，从

而使得基于 QML 设计的图形用户界面也能够动态及时地显示在 Python 脚本中进行的协议解析过程所得到的各项数据。

2.2.4 Qt Quick2.5

Qt quick2.5 是一个 QML 工具库,在 Qt quick 中,使用 QML 语言来制作用户图形界面^[7],在本软件设计中,所用来的开发人机交互界面的 qml 文件都引用了 Qt Quick2.5 工具包,引用方式如下: `import Qt Quick 2.x`,引入后就可以在 qml 界面设计中引用工具包所提供的各种功能和控件了^[8]。

2.2.5 MySQL5.7

MySQL 是现在最流行的关系型数据库管理系统,在本软件设计中,使用 MySQL 服务器中的数据库存储解析得到的各设备的各项数据^[9],可以通过在 Python 应用程序中引入 pymysql 模块的 `pymysql.Connect()`函数帮助 Python 脚本连接到已建立的 MySQL 服务器的数据库,并对该数据库的各项数据内容进行插值,替换或者更新操作,从而实现解析数据的后台转发工作^[10]。

2.2.6 MySQL 管理工具(Navicat for MySQL)

Navicat for MySQL 是一款辅助开发 MySQL 数据库的管理工具,其中的 .bat 文件可以用来启动和关闭 MySQL 服务器。在本设计中利用 Navicat for MySQL 方便地对 MySQL 数据库表进行操作:先使用 .bat 文件启动 MySQL 服务器,再用 Navicat for MySQL 新建到 MySQL 服务器的连接(或者连接到已建立的 MySQL 服务器),然后就可以在服务器中创建数据库,或者在已建立的数据库内完成新建表,设计表,清空表或者删除表的一系列操作。

2.3 总体设计方案

在该研究开展之前,海水淡化系统所在工业现场实际配置的工控机与各个设备之间已经通过工控机中其他软件程序以 Modbus 协议建立端到端的连接了,所以无法通过直接访问的方式读取工控机与各个设备之间的传输信息,又需要在不改变其他原软件程序的基础上取得这些传输信息并对他们按相关协议进行解析从而获得有效数据。在这种情况下,必须通过其他方式且尽可能实时地取得这些传输信息,因此在研究中利用 Wireshark 网络抓包的方式获取这些通信数据^[11],具体操作方式是在工控机上利用网络封包分析软件 Wireshark 捕获机器网卡的网络数据包^[12],这样 Wireshark 所截取的网络封包文件.PCAPNG 就是本研究中所设计的数据协议解析软件需要分析的对象^[12],设计软件按照工控机与各设备之间的实际协议对该网络数据包进行协议解析^[13],从中获取工控机与各设备间传输的各项有效数据并设计人机交互图形界面展示这些数据,同时按需求转发解析得到的数据至后台服务器的数据库。

因此总体设计方案设计如下:

(1) 在工控机上部署好开发环境，安装各项开发工具，搭建软件设计框架

(2) 在工控机上使用网络封包分析工具，撷取工控机网卡正在收发的网络封包，获取一段时间内工控机与海水淡化系统各设备间的通信数据。

(3) 按实际相关协议设计数据协议解析软件，对网络数据包 PCAPNG 内的各项相关通信数据进行解析以得到海水淡化系统各相关设备：1、2 号风电机组，储能双向变流器 (PCS)，电池管理系统 (BMS)，海水淡化设备以及故障报警设备 (ADAM1,ADAM2,ADAM3)向工控机(IPC)传输的各项有效数据。

(4) 完成基本的人机交互图形界面设计，用于展示相关数据和提供更改等功能，比如用户可以通过文件选择对话框选中相应的网络数据包 PCAPNG 文件进行数据解析、展示和转发。

(5) 开发程序将解析后得到的各项有效数据按实际需求转发至后台 MySQL 服务器数据库。

2.4 总体设计方案流程图

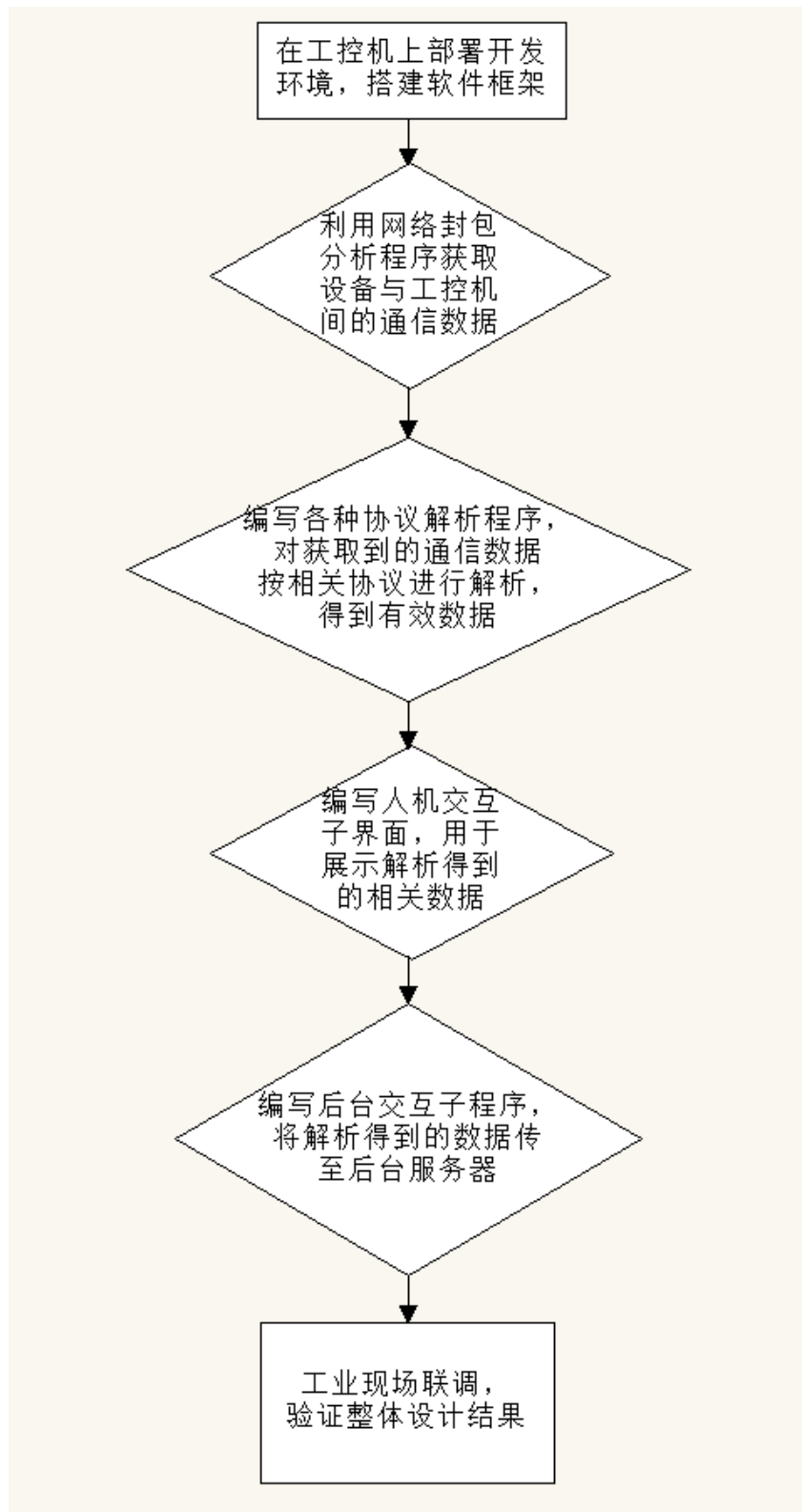


图 2-2 总体设计方案流程图

2.5 软件总体架构设计

一款功能完善的软件，要有一个好的软件架构设计，在软件架构设计中既要考虑到系统的整体功能分别用哪些部分实现，又得考虑到系统各个部分之间如何相互作用，为实现前述的总体设计方案，本设计中采用 Python 编程与 QML 界面还有 MySQL 数据库相结合的软件架构：

1)利用 Python3.6 编程分析用户所选择的 PCAPNG Wireshark 数据包文件,根据实际协议解析得到各项有效数据；并利用 QML 语言结合 Qt Quick 模块创建图形化用户界面；

2)再通过 PyQt5 将 QML 设计的界面和 Python 应用程序结合起来，从而使得基于 QML 设计的图形用户界面能够动态及时地显示在 Python 脚本中进行的协议解析过程所得到的各项数据；

3)同时软件程序会通过 pymysql 模块连接到已启动的 MySQL5.7.9 后台服务器中已建立的数据库，并且按实际需求将解析得到的数据插入至该后台服务器数据库的相应表中。

因此，软件整体架构可按功能划分为如下几个设计部分：

2.5.1 主程序文件

主程序文件是用 Python 语言编写的脚本文件，作为整体软件的可执行文件，是软件架构中的基础部分，也是核心部分。

主程序文件包括 main.py, QmlView.py，两者都是可执行文件，QmlView.py 是真正的主程序文件，软件的协议解析、界面展示、数据转发等全部功能都编写在 QmlView.py 主程序文件中，通过主程序文件或者借由主程序文件调用系统中其他设计文件实现。而 main.py 则是简化了的主程序文件，通过调用在 QmlView.py 中声明和定义的各种函数，执行后可以完成和 QmlView.py 一样的功能，只不过代码比起 QmlView.py 精简了很多。

2.5.2 界面设计文件

界面设计文件是用 QML 语言编写的 qml 文件，作为界面展示文件，可以由主程序文件调用，并且界面中各项内容的值也可以由 QmlView.py 进行改写。这些 qml 文件都引用了 Qt Quick2.5 工具包，引入后就可以在 qml 界面设计中引用工具包所提供的各种功能和控件了。

界面设计文件包括 ADAM1.qml, ADAM2.qml, ADAM3.qml, BMS.qml, EMS.qml, DesalinPlant.qml, WINDTURB2.qml, WINDTURB2.qml, PageView.qml 和 ListViewIndicator.qml。前面几个 qml 文件分别是故障报警点表设备 1，故障报警点表设备 2，故障报警点表设备 3，电池组，交直流能量逆变器（储能双向变流器），反渗透海水淡化装置，风电机组 1，风电机组 2 的界面设计文件，最后两个 qml 文件是为了在人机用户界面中实现通过按住鼠标左键左右滑动来切换显示界面的功能而编写的组件。通

过在其他 qml 界面设计文件中调用这两个组件就可以实现按住鼠标左键左右滑动来切换显示界面这样的功能。

2.5.3 MySQL 数据库表文件

MySQL 数据库存放在 MySQL5.7 目录下的 data 文件夹内，在 data 文件夹下新建一个文件夹，就会成为一个 MySQL 数据库。本软件使用 MySQL 后台服务器的数据库中的表来转存解析得到的数据，为此需要先将解析数据转发至 MySQL 数据库相应的表中，这个转发过程编写在主程序文件中。

本设计中包括如下的 MySQL 数据库表文件：adam1.frm, adam1.ibd, adam2.frm, adam2.ibd, adam3.frm, windturb1.ibd, ibdata1，它们分别用来转存故障报警点表设备 1，故障报警点表设备 2，故障报警点表设备 3 和风电机组 1 的解析数据。

2.5.4 辅助文件

本软件设计中的辅助文件是指放在软件目录下 config 文件夹里的两个文件：ip.py 和 settings.py。在它们中定义了一些全局变量和程序运行的初始参数，这样可以在主程序文件中通过引入这两个辅助文件使用这些全局变量以实现某些具体功能。

ip.py 中通过定义相应的变量记录了海水淡化系统各个设备的 ip 地址。setting.py 文件除了定义了一些主程序会用到的全局变量外还定义了几个程序运行时会参考的初始参数。

2.5.5 网络数据封包文件

Wireshark 所截取的网络封包文件.PCAPNG 就是本研究中所设计的数据协议解析软件需要分析的对象，其中包含了工控机在某一段时间内网卡上所收发的全部网络数据，所以包括了海水淡化系统各个设备与工控机传输的各项有效数据，比如各设备的工作状态参数值，运行状态位的值等等。

本设计中解析的网络数据封包文件是 DATA.pcapng。本软件需要对该网络数据包进行协议解析以获取这些有效数据的准确数值并动态显示在图形用户界面上。

第三章 数据协议解析设计

3.1 协议解析设计思路

在本设计中,协议解析的对象是 Wireshark 抓取的网络数据封包文件 DATA.pcapng^[14]。在设计数据协议解析程序之前,有必要研究清楚 PCAPNG 文件中存储数据的格式。用 Wireshark 软件打开该数据包文件,可以看到如下的界面:

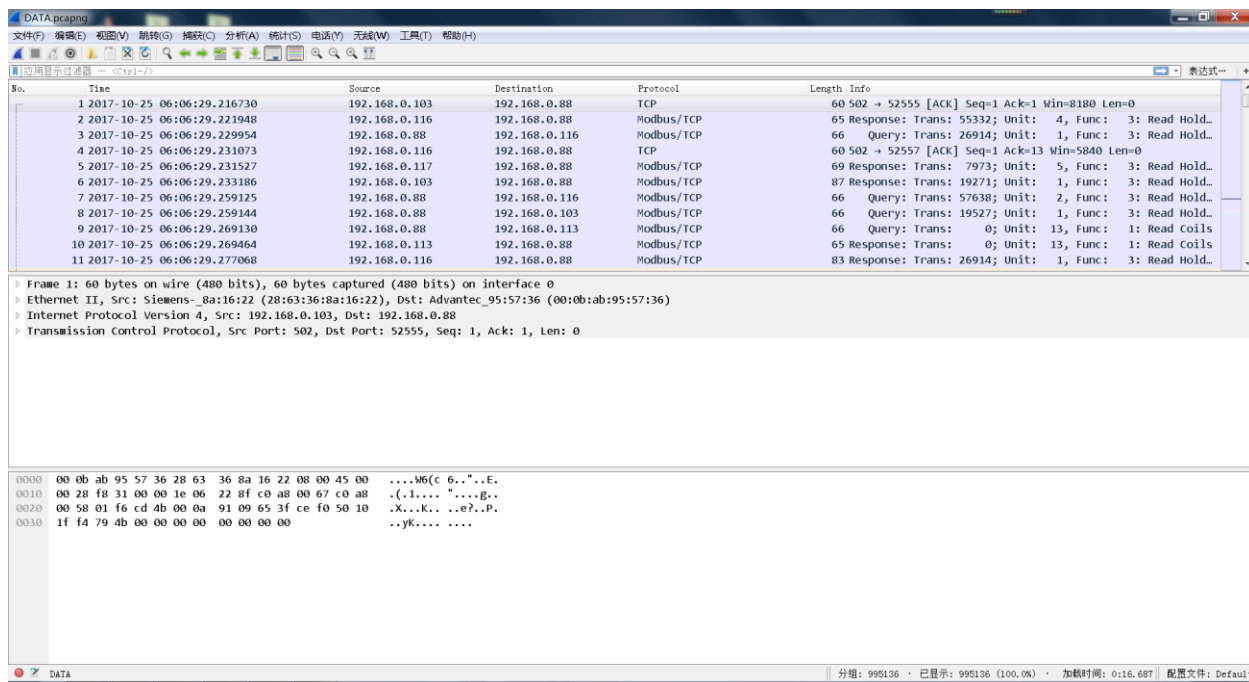


图 3-1 Wireshark 打开 DATA.pcapng 文件效果示意图

可以发现 Wireshark 为 PCAPNG 数据包中的每一条数据都由左到右分别标注出了编号 (No.) 抓取时间 (Time), 数据源 IP 地址 (Source), 数据目的 IP 地址 (Destination), 数据传输使用的通讯协议 (Protocol)。这些都是对一些对数据协议解析很重要的信息。点击其中编号为 6 的那一条数据, 发现其源 IP 地址是 192.168.0.103, 目的 IP 地址是 192.168.0.88^[15]。从本课题提供的各设备通讯协议文件中知道, 192.168.0.103 是设备风电机组 1 的 IP 地址, 而 192.168.0.88 正是用来抓取网络数据包的工控机自身的 IP 地址。所以可以判断出这条数据是由风电机组 1 发送至工控机的数据, 其中可能就携带着需要我们进行解析的有效数据。

继续研究, 可以发现最下面的窗口显示的是这条数据携带的以 16 进制形式编码的信息, 再点击这些 16 进制编码中的任意一个, 中间的窗口就会显示这个编码所对应的详细信息, 即从中间窗口可以查看某一条数据所携带的全部编码分别对应的具体信息。如下图所示:

```

▶ Frame 6: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface 0
▶ Ethernet II, Src: Siemens_8a:16:22 (28:63:36:8a:16:22), Dst: Advantec_95:57:36 (00:0b:ab:95:57:36)
▶ Internet Protocol Version 4, Src: 192.168.0.103, Dst: 192.168.0.88
▶ Transmission Control Protocol, Src Port: 502, Dst Port: 52555, Seq: 1, Ack: 1, Len: 33
▶ Modbus/TCP
▶ Modbus

```

图 3-2 Wireshark 中间窗口显示效果示意图

从上图窗口中的 Modbus/TCP 字段可以看出这条数据是以 Modbus/TCP 协议传输的。又从本课题所提供的各设备通讯协议文件中了解到各个设备与 IPC 工控机间传输的有效信息都是在 Modbus/TCP 协议下传输的^[16]。因此其中确实就携带着就是我们需要解析的数据。

又通过学习 Modbus/TCP 数据报文结构^[17]，了解到数据携带的有效信息存储在对应于中间窗口标有 Modbus 的字段内。展开上图窗口中的 Modbus 字段：

```

▶ Transmission Control Protocol, Src Port: 502, Dst Port: 52555, Seq: 1, Ack: 1, Len: 33
▶ Modbus/TCP
▲ Modbus
  .000 0011 = Function Code: Read Holding Registers (3)
  Byte Count: 24
  ▶ Register 0 (UINT16): 2
  ▶ Register 1 (UINT16): 2
  ▶ Register 2 (UINT16): 2
  ▶ Register 3 (UINT16): 2
  ▶ Register 4 (UINT16): 333
  ▶ Register 5 (UINT16): 396
  ▶ Register 6 (UINT16): 3510

```

图 3-3 Modbus 字段示意图(a)

```

  ▶ Register 4 (UINT16): 333
  ▶ Register 5 (UINT16): 396
  ▶ Register 6 (UINT16): 3510
  ▲ Register 7 (UINT16): 2714
    Register Number: 7
    Register Value (UINT16): 2714
  ▶ Register 8 (UINT16): 0
  ▶ Register 9 (UINT16): 0
  ▶ Register 10 (UINT16): 1
  ▲ Register 11 (UINT16): 5
    Register Number: 11
    Register Value (UINT16): 5

```

图 3-3 Modbus 字段示意图(b)

展开后可以知道 Modbus 字段内的第一个字节表示这条数据的功能码 Function Code，如果其数值为 1 代表这条数据携带的有效信息读自线圈（一个线圈可读的数据是

一位，即位操作)；如果其数值为 3 代表这条数据携带的有效信息读自保持寄存器（一个寄存器可读的数据是一个字，即字操作，且一个字有 16 位）。

Modbus 字段内的第二个字节表示这条数据所读取的有效信息的字节数 Byte Count，这里显示的是 24，代表这条数据读取的有效信息字节数是 24，因为一个字（寄存器）占两个字节，所以这条数据包含了从风电机组 1 设备的 12 个寄存器读取的 12 个字的有效信息。且 Modbus 字段内第二个字节之后的内容都是所读取的有效信息，即位于风电机组 1 设备上的 12 个寄存器所携带的设备信息。

至此，协议解析软件的设计思路就逐渐清晰了：设计基于 Python 的软件程序，从 PCAPNG 网络数据封包中提取出这一条条数据，从中筛选出符合要求的需要解析的数据条，再从这些需要解析的数据条中获取其 Modbus 字段内的有效信息，这些信息就代表着某一台设备的某些状态参数的具体数值，再将这些解析得到的数值用人机交互界面展示出来即可。

可是这样自然产生了一个问题：怎么确定解析出来的具体数值对应的是哪个设备的哪个状态参数呢？这就需要结合本课题所提供的各设备通讯协议文件来判断了。还是以上面分析的那条数据为例，因为其 IP 对应的设备是风电机组 1，所以打开风电机组 1 的通讯协议文件，查看其中的风电机组 1 的通讯信息表，可以发现风电机组 1 有三张通讯信息表，依次分别为：

表 3-1 风电机组 1 通讯信息表(a)

序号	地址	字节	数据类型	变量名称	说 明	单位	备注
1	40001	2	UInt	机组控制方式	1=本地方式；2=远程方式	无	
2	40002	2	UInt	机组运行模式	1=手动模式；2=自动模式；3=维护模式；4=停机模式	无	
3	40003	2	UInt	机组运行状态	1=启动;2=运行;3=暂停;4=停机;5=急停;6=空转		
4	40004	2	UInt	报警信息	1=正常;2=警告;3=故障		
5	40005	2	Int	1s 风速	实际值 x100;	m/s	
6	40006	2	Int	5min 平均风速	实际值 x100;	m/s	
7	40007	2	Int	10min 平均风向	实际值 x10	°	
8	40008	2	Int	风轮/发电机转速	实际值 x100;	rpm	
9	40009	2	Int	叶片角度	实际值 x10	°	
10	40010	2	UInt	报警信息 1	实际值，详见附录一。		
11	40011	2	UINT	通讯检测	PLC->PC，循环发送，周期为 1s，其中 0 为 500ms，1 为 500ms		
12	40012	2		未定义	备用		

表 3-1 风电机组 1 通讯信息表(b)

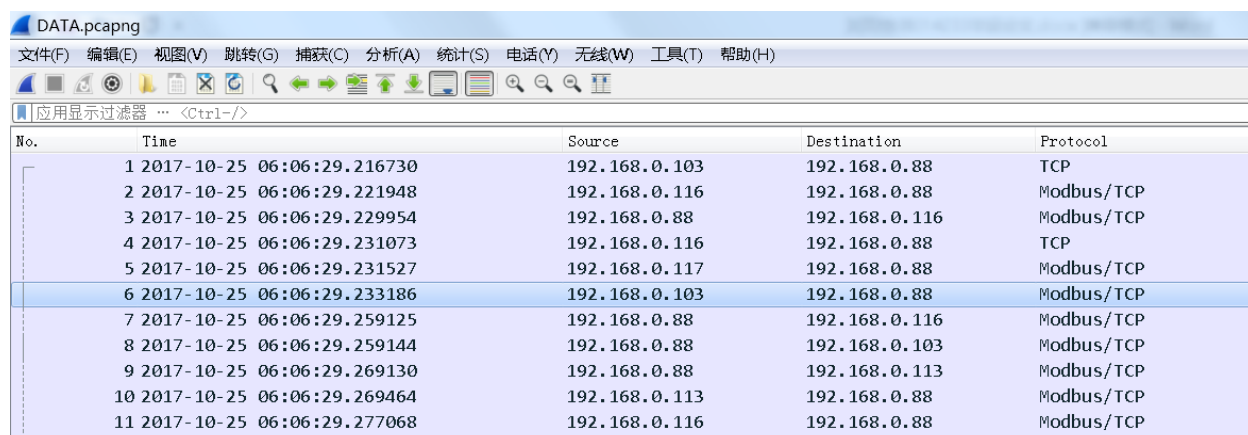
序号	地址	字节	数据类型	变量名称	说 明	单位	备注
1	40065	2	Int	电网电压 U	实际值×10	Vac	
2	40066	2	Int	电网电压 V	实际值×10	Vac	
3	40067	2	Int	电网电压 W	实际值×10	Vac	
4	40068	2	Int	输出电流	实际值×100	Aac	
5	40069	2	Int	输出电流	实际值×100	Aac	
6	40070	2	Int	输出电流	实际值×100	Aac	
7	40071	2	Int	输出功率	实际值×10	kW	
8	40072	2	Int	输出无功	实际值×10	kVar	
9	40073	2	Int	功率因素	实际值×10000		
10	40074	2	UInt	总有功发电量(高字)	实测值×100	kWh	
11	40075	2	UInt	总有功发电量(低字)	实测值×100	kWh	

表 3-1 风电机组 1 通讯信息表(c)

序号	地址	字节	数据类型	变量名称	说 明	单位	备注
1	40257	2	UInt	远程急停	实际值, 0=无操作; 1=急停; 2=停止急停		[0, 1]
2	40258	2	UInt	机组运行模式选择	1=手动模式; 2=自动模式(启动); 3=维护模式; 4=停机		[1, 2, 3, 4]
3	40259	2	UInt	复位按钮	0=无操作; 1=复位操作;		[0, 1]
4	40260	2	UInt	有功给定	100=100kW		
5	40261	2	UInt	通讯检测	PC->PLC,循环检测,周期为 1s, 其中 0 为 500ms, 1 为 500ms; 连续 30s 无变化则认为通讯故障, 停机。		

从以上的三张通讯表可以判断出,风电机组 1 是通过前后共三次的 Modbus/TCP 通讯来传输其全部状态参数给工控机的,由于风电机组 1 通讯信息表(c)中只有 5 个字(寄存器)的有效信息传输,而上面分析的那条数据包含的有效信息的字节数 Byte Count 为 24,即读取了 12 个字(寄存器)的有效信息^[18]。因此,那条数据所对应的设备工作状态参数一定不是风电机组 1 通讯信息表(c)中所标的那 5 个状态变量,同理也不是风电机组 1 通讯信息表(b)中所标的那 11 个状态变量,由此可以判断编号为 6 的那条数据携带的有效信息来自风电机组 1 通讯信息表(a)^[19]。

为了验证这个判断,可以回到 PCAPNG 数据包,查看源 IP 为风电机组 1 的其他各条数据与编号为 6 的数据的位置关系。重新定位到 PCAPNG 里的第 6 条数据,如下图所示:



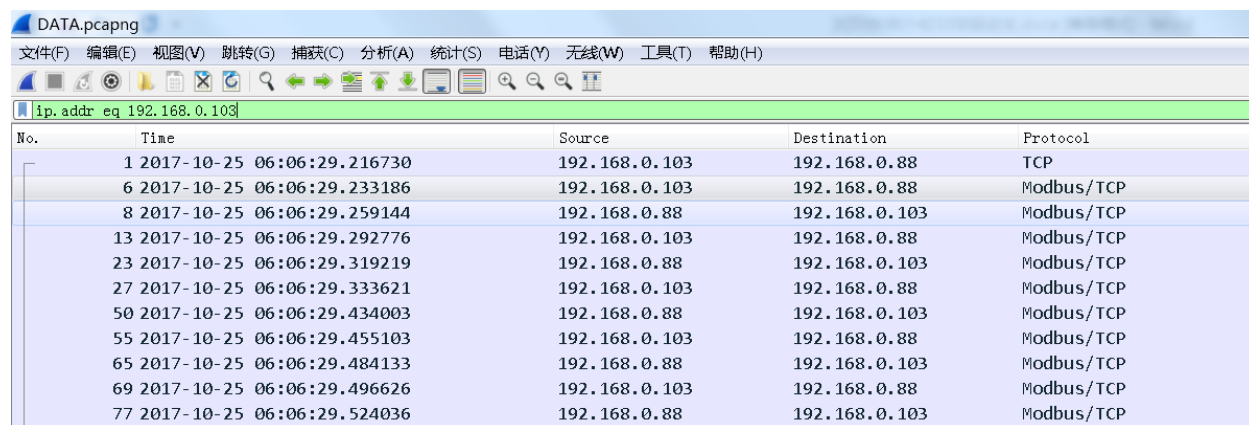
No.	Time	Source	Destination	Protocol
1	2017-10-25 06:06:29.216730	192.168.0.103	192.168.0.88	TCP
2	2017-10-25 06:06:29.221948	192.168.0.116	192.168.0.88	Modbus/TCP
3	2017-10-25 06:06:29.229954	192.168.0.88	192.168.0.116	Modbus/TCP
4	2017-10-25 06:06:29.231073	192.168.0.116	192.168.0.88	TCP
5	2017-10-25 06:06:29.231527	192.168.0.117	192.168.0.88	Modbus/TCP
6	2017-10-25 06:06:29.233186	192.168.0.103	192.168.0.88	Modbus/TCP
7	2017-10-25 06:06:29.259125	192.168.0.88	192.168.0.116	Modbus/TCP
8	2017-10-25 06:06:29.259144	192.168.0.88	192.168.0.103	Modbus/TCP
9	2017-10-25 06:06:29.269130	192.168.0.88	192.168.0.113	Modbus/TCP
10	2017-10-25 06:06:29.269464	192.168.0.113	192.168.0.88	Modbus/TCP
11	2017-10-25 06:06:29.277068	192.168.0.116	192.168.0.88	Modbus/TCP

图 3-4 需要分析的 PCAPNG 数据包中的数据条示意图

其编号 NO.为 6，代表它是数据包抓取的第 6 条数据，在上图的“应用显示过滤器”输入框内输入：

`ip.addr eq 192.168.0.103`

得到如下图的窗口：



No.	Time	Source	Destination	Protocol
6	2017-10-25 06:06:29.233186	192.168.0.103	192.168.0.88	Modbus/TCP
8	2017-10-25 06:06:29.259144	192.168.0.88	192.168.0.103	Modbus/TCP
13	2017-10-25 06:06:29.292776	192.168.0.103	192.168.0.88	Modbus/TCP
23	2017-10-25 06:06:29.319219	192.168.0.88	192.168.0.103	Modbus/TCP
27	2017-10-25 06:06:29.333621	192.168.0.103	192.168.0.88	Modbus/TCP
50	2017-10-25 06:06:29.434003	192.168.0.88	192.168.0.103	Modbus/TCP
55	2017-10-25 06:06:29.455103	192.168.0.103	192.168.0.88	Modbus/TCP
65	2017-10-25 06:06:29.484133	192.168.0.88	192.168.0.103	Modbus/TCP
69	2017-10-25 06:06:29.496626	192.168.0.103	192.168.0.88	Modbus/TCP
77	2017-10-25 06:06:29.524036	192.168.0.88	192.168.0.103	Modbus/TCP

图 3-5 按 IP 地址过滤后的 PCAPNG 数据包数据示意图

可以发现按 IP 地址过滤后的数据只显示源和目的 IP 来自风电机组 1 的数据条。因为编号为 1 的数据条使用的不是 Modbus/TCP 传输协议，所以不含来自风电机组 1 的有效信息，而编号为 8 的数据条不是来自风电机组 1，因此也不含来自风电机组 1 的有效信息。这样逐个分析排除下来，筛选出编号为 No.13 和 27 的数据条，单击它们，查看其包含的有效信息的字节数 Byte Count，如下图所示：

DATA.pcapng

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计

ip.addr eq 192.168.0.103

No.	Time
1	2017-10-25 06:06:29.216730
6	2017-10-25 06:06:29.233186
8	2017-10-25 06:06:29.259144
13	2017-10-25 06:06:29.292776
23	2017-10-25 06:06:29.319219
27	2017-10-25 06:06:29.333621
50	2017-10-25 06:06:29.434003
55	2017-10-25 06:06:29.455103
65	2017-10-25 06:06:29.484133
69	2017-10-25 06:06:29.496626
77	2017-10-25 06:06:29.524036

- Modbus/TCP
 - Transaction Identifier: 19527
 - Protocol Identifier: 0
 - Length: 25
 - Unit Identifier: 1
- Modbus
 - .000 0011 = Function Code: Read Holding Register
 - [Request Frame: 8]
 - Byte Count: 22
 - Register 64 (UINT16): 4024
 - Register Number: 64
 - Register Value (UINT16): 4024

图 3-6 No.13 数据条 Modbus 字段示意图

DATA.pcapng		
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T)		
ip.addr eq 192.168.0.103		
No.	Time	Source
1	2017-10-25 06:06:29.216730	192.168.0.103
6	2017-10-25 06:06:29.233186	192.168.0.103
8	2017-10-25 06:06:29.259144	192.168.0.88
13	2017-10-25 06:06:29.292776	192.168.0.103
23	2017-10-25 06:06:29.319219	192.168.0.88
27	2017-10-25 06:06:29.333621	192.168.0.103
50	2017-10-25 06:06:29.434003	192.168.0.88
55	2017-10-25 06:06:29.455103	192.168.0.103
65	2017-10-25 06:06:29.484133	192.168.0.88
69	2017-10-25 06:06:29.496626	192.168.0.103
77	2017-10-25 06:06:29.524036	192.168.0.88

Modbus/TCP Transaction Identifier: 19783 Protocol Identifier: 0 Length: 13 Unit Identifier: 1
Modbus .000 0011 = Function Code: Read Holding Registers (3) [Request Frame: 23] Byte Count: 10 Register 256 (UINT16): 0 Register Number: 256 Register Value (UINT16): 0

图 3-7 No.13 数据条 Modbus 字段示意图

由上图 3-6 和图 3-7 可以发现编号为 13 和 27 的数据条分别包含了 11 和 5 个寄存器（字）的有效信息，再对照表 3-1，可以发现编号 6,13,27 所对应的三个数据条代表了风电机组 1 通过前后共三次的 Modbus/TCP 通讯完成了数据包中第一轮传输其全部状态参数给工控机的工作。

而且可以发现上面研究的那条编号为 6 的数据条是风电机组 1 完成的第一轮数据传输过程中的第一次数据传输，结合风电机组 1 的通讯协议文件中通讯表的次序也可以判断出编号为 6 的那条数据携带的有效信息来自风电机组 1 通讯信息表(a)。从而成功地进行验证。

由此确定该数据条内包含的 12 个字的有效信息（由第 1 个字到第 12 个字）分别依次对应风电机组 1 通讯信息表(a)中标出的那 12 个状态变量（机组控制方式状态字，机组运行模式状态字，机组运行状态状态字，报警信息，风速，5min 平均风速，10min 平均风向，叶片角度，报警信息 1 和通讯检测字，未定义量）的实时数据，注意有效信息的数值不一定就等于这些状态变量的数值，它们只是对应关系，有的相等，有的是整数倍的关系，关于这个具体可以看通讯信息表内的状态变量说明。

这样根据设备的实际通讯协议文件内的通讯信息表，就可以判断出相应的待解析数据内包含的有效信息分别对应着该设备的哪些状态变量了。如此，就可以将解析出的数据

展示在图形界面中对应的状态变量值所在的位置了。

本软件设计的 Python 协议解析程序通过导入 pyshark 模块调用 Wireshark 软件提供的命令行工具 tshark，使用其提供的 Filecapture 函数可以打开已抓取的网路数据封包对其进行协议解析，若使用 Livecapture 函数则可以直接从本机网卡进行嗅探获取网路数据包并对其进行协议解析。

3.2 设计方案总结

综上所述，可以总结出本软件中数据协议解析编程的设计方案：

- (1) 弹出文件选择对话框，返回用户选择的待解析的数据封包的文件路径
- (2) 用 Pyshark 的 Filecapture 函数打开待分析的数据包，依次读取包内各条数据的信息
- (3) 分析当前读取的数据条是否包含“Modbus”字段，如果包含前往 (4)，否则回到 (2)
- (4) 分析当前读取的数据条的原 IP 地址是否等于海水淡化系统各设备的 IP 地址，如果不等于任一设备的 IP 地址回到 (2)，否则前往 (5)
- (5) 根据(4)中分析的当前数据条的 IP 地址，选择执行对应的协议解析函数
- (6) 分析当前数据条 Modbus 字段内的 Function Code 功能码是否满足值为 1 或者 3，如果满足前往 (7)，否则回到 (2)
- (7) 分析当前数据条 Modbus 字段内的 Byte Count 字节数（或者 Bit Count 位数），将其与对应设备的通讯协议书内所有通讯信息表的状态变量数进行比较，从而判断该数据条内的有效信息对应于哪一个通讯信息表，继而将该数据条内的各有效信息分别传给对应通讯信息表内的各状态变量
- (8) 当前数据条的数据协议解析工作进行完毕，回到 (2)

3.3 协议解析核心代码设计

网络协议解析是利用软件程序分析网络数据包的传输协议及其数据格式，以了解网络数据包中携带的有效信息。拥有协议解析功能的软件往往需要解析多种不同的网络协议，因此协议解析程序需要具备区别解析在不同通讯协议下传输的数据的能力，这是协议解析程序设计的重点。而本软件在设计中，先详细掌握了海水淡化系统各设备的通讯协议后，再根据各设备实际的通讯协议，为每一个设备都单独设计了一个协议解析函数，以解析来自相关设备的数据。

3.1 中以风电机组 1 的数据协议解析为例分析了设计思路。这里再以其为例，按 3.2 总结的设计方案，设备风电机组 1 的数据协议解析函数流程图设计如下：

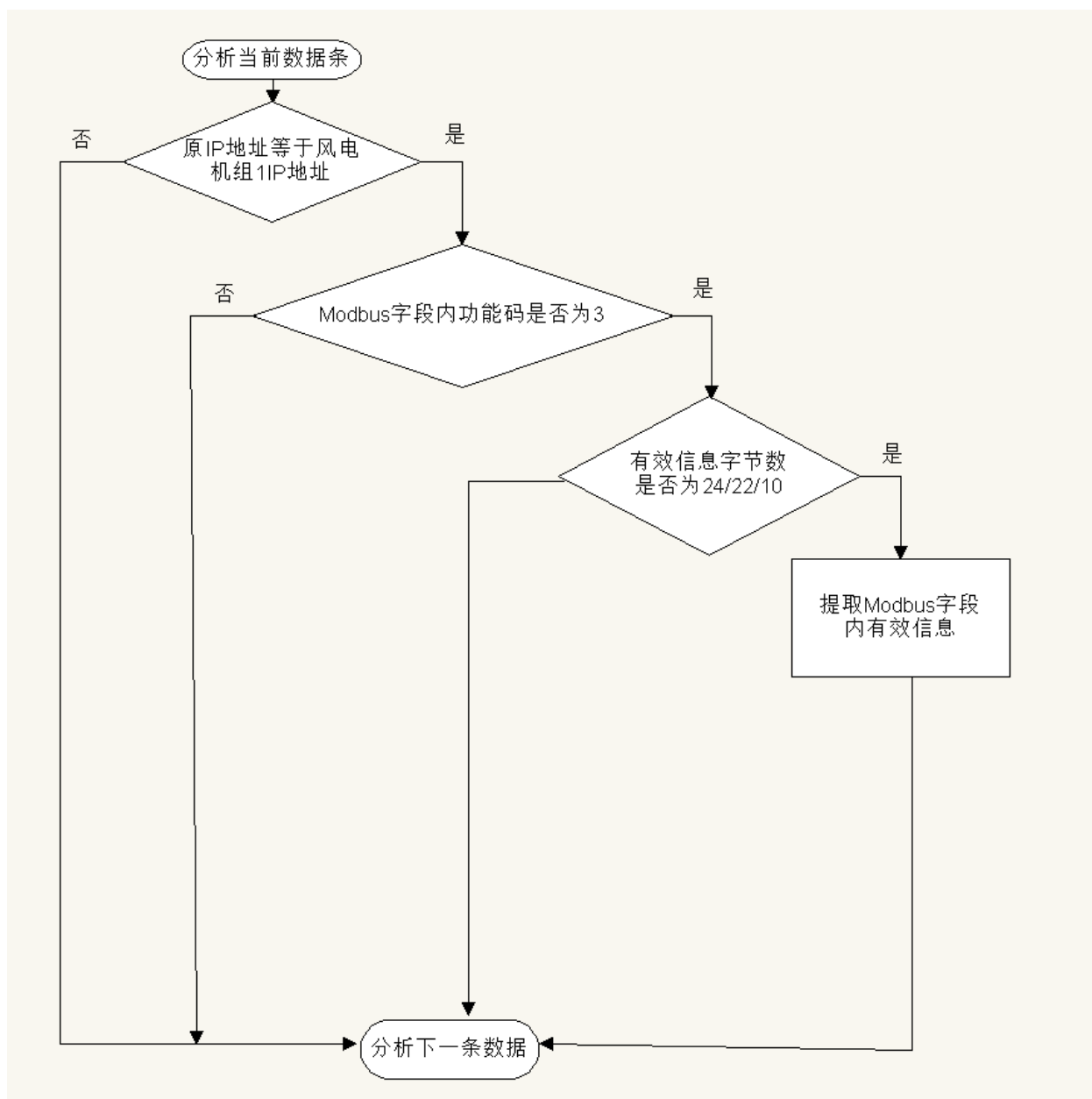


图 3-8 风电机组 1 数据协议解析函数流程图

核心代码设计如下：

```

for p in cap:
    if 'MODBUS' in p:
        if p['IP'].src == config.ip.WINDTURB1:           //判断 IP 地址是否满足
            self.packet_windturbine_1(p)
        elif p['IP'].src == ...
            .....
            continue
    else:
        return
  
```

```

def packet_windturbine_1(self, packet):
    packet_data_str = []
    if packet['MODBUS'].func_code == '3':           //判断功能码是否满足
        pass
    else:
        return
    if packet['MODBUS'].byte_cnt == '24':           //判断有效信息长度是否满足 24
        for field in packet['MODBUS']._get_all_fields_with_alternates():
            if isinstance(field, pyshark.packet.layer.LayerFieldsContainer):
                if field.main_field.get_default_value()[0] == 'R':
                    # print(field.main_field.get_default_value(), ' ', field.raw_value)
                    packet_data_str.append(field.raw_value)
            if isinstance(field, pyshark.packet.layer.LayerField):
                if field.get_default_value()[0] == 'R':
                    packet_data_str.append(field.raw_value)    //获取有效信息
        else:
            return

    elif packet['MODBUS'].byte_cnt == '22':           //判断有效信息长度是否满足 22
        for field in packet['MODBUS']._get_all_fields_with_alternates():
            //获取有效信息
            .....
        else:
            return

    elif packet['MODBUS'].byte_cnt == '10':           //判断有效信息长度是否满足 10
        //获取有效信息
        .....
    else:
        return

```

第四章 用户界面和数据展示软件设计

4.1 用户界面设计

4.1.1 Python3+PyQt5+ QML 设计界面

图形界面是软件程序的外在体现，设计一个交互友好、简洁清楚的界面也是软件设计的重要目标。而用 QML 编程语言设计界面具有指令简洁，功能强大的优点^[20]。因此本软件利用 qml 文件设计界面。这些 qml 文件都引用了 Qt Quick2.5 工具包，以便在 qml 界面设计中引用工具包所提供的各种功能和控件。

本设计中主体程序文件使用 Python 编程，图形界面使用 QML 文件显示，通过在 Python3 中调用 PyQt5 工具集，将在 Qt 中用 QML 设计的界面和 Python 应用程序很好地结合起来。

通过在 Python 主程序中引入 PyQt5.QtQuick 的 QQuickView 模块，可以在 Python 脚本执行过程中调用指定路径下的 QML 文件，并且显示基于这个 QML 文件设计的用户图形界面。设计核心代码如下：

```
class QmlView(QQuickView):
    def __init__(self):
        super().__init__()
        self.data_source = DataSource()
        self.data_source.sigUpdate.connect(self.update)
        self.data_source.start()

    @pyqtSlot(str, str)
    def update(self, prop, value):
        self.rootObject().setProperty(prop, value)

path = 'qml/main.qml'
view = QmlView()
view.setSource(QUrl(path))
view.show()
```

其中 path 是在 Python 程序中打开的作为主界面 QML 文件路径。view.show()用来执行打开 QML 界面的操作。

4.1.2 设计文件选择对话框

通过在图形用户界面中引入文件选择对话框，可以使得人机互动性更强，软件功能更加完善友好，可以通过调用 Python3.2 版本后系统自带的图形界面库 tkinter 实现这一功能。

在 Python 脚本中调用文件选择对话框的代码如下：

```
filename=filedialog.askopenfilename(filetypes=[("抓包数据文件", "pcapng")]))
```

其中，代码执行后会弹出一个文件选择对话框，`filetypes` 的参数设置决定了用户只能选择 PCAPNG 文件打开，选择文件打开后，对话框会自动关闭，同时将用户刚刚选择的文件完整路径赋给 `filename`。

代码执行后还会自动弹出一个标有“tk”的界面框，可以通过在调用文件选择对话框的代码之前添加如下两句命令让该界面框不再显示：

```
root = tk.Tk()
root.withdraw()
```

显示文件选择对话框的效果图如下：

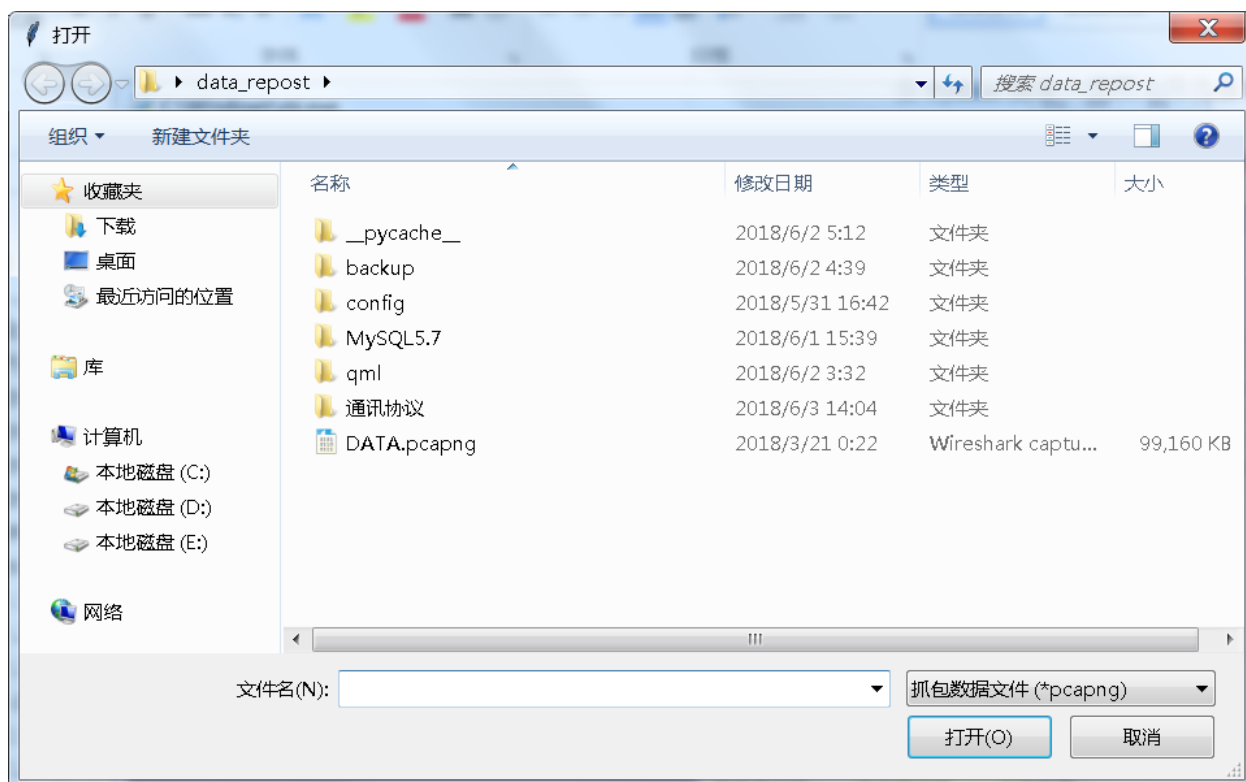


图 3-1 文件选择对话框效果图

4.1.3 标签式界面设计

可以在主界面设计 `qml` 文件中引入 `Tabview` 定义格式，生成标签式的图形用户界面，这样设计需要一个主界面 `main.qml`。在主界面中定义其他所有用户界面中会用到的变量，而且采用如下形式的代码设计：

```
Tabview
{
    property string windturb1_40005: '00 00'
    property string windturb1_40006: '0000'
    .....
    Tab{
        title: "风电机组 1"
        WINDTURB1{
```

```

    }

    }

    Tab{
        title: "风电机组 2"
        WINDTURB2{
        }
    }
}

```

其中 QML 语句 `property string windturb1_40005: '0000'` 为整个界面定义了一个字符型变量 `windturb1`，其初始化为‘0000’（子界面 `WINDTURB1`，`WINDTURB2` 均可使用这个变量）。

上面这段代码用 `Tab{}` 定义了两个标签式界面，其中的 `title` 就是标签名，而 `WINDTURB1{}` 表示与 `main.qml` 同目录下的 `WINDTURB1.qml` 界面设计文件所定义的界面将作为主界面 `main.qml` 的一个标签式子界面展示在主界面中，其上贴有标签“风电机组 1”。同理，贴有“风电机组 2”标签的子界面由与主界面设计文件 `main.qml` 同目录下的 `WINDTURB2.qml` 定义，并且这两个标签式子界面之间可以通过用鼠标单击标签的方式进行替换显示，切换效果如下图：



图 3-2 Tabview 标签设计效果图(a)



图 3-2 Tabview 标签设计效果图(b)

4.1.4 界面设计效果

本软件中的界面设计文件除了前面提到的主界面文件 `main.qml`，标签式子界面文件 `WINDTURB2.qml`，`WINDTURB2.qml` 外，还有 `ADAM1.qml`，`ADAM2.qml`，`ADAM3.qml`，`BMS.qml`，`EMS.qml`，`DesalinPlant.qml`，`PageView.qml` 和 `ListViewIndicator.qml`。

前面几个 `qml` 文分别是故障报警点表设备 1，故障报警点表设备 2，故障报警点表设备 3，电池组，交直流能量逆变器（储能双向变流器），反渗透海水淡化装置的界面设计文件，最后两个 `qml` 文件是为了在人机用户界面中实现通过按住鼠标左键左右滑动来切换显示界面的功能而编写的组件。通过在其他 `qml` 界面设计文件中使用这两个组件所

定义类 `Pageview{}`，就可以实现按住鼠标左键左右滑动来切换显示界面这样的功能。

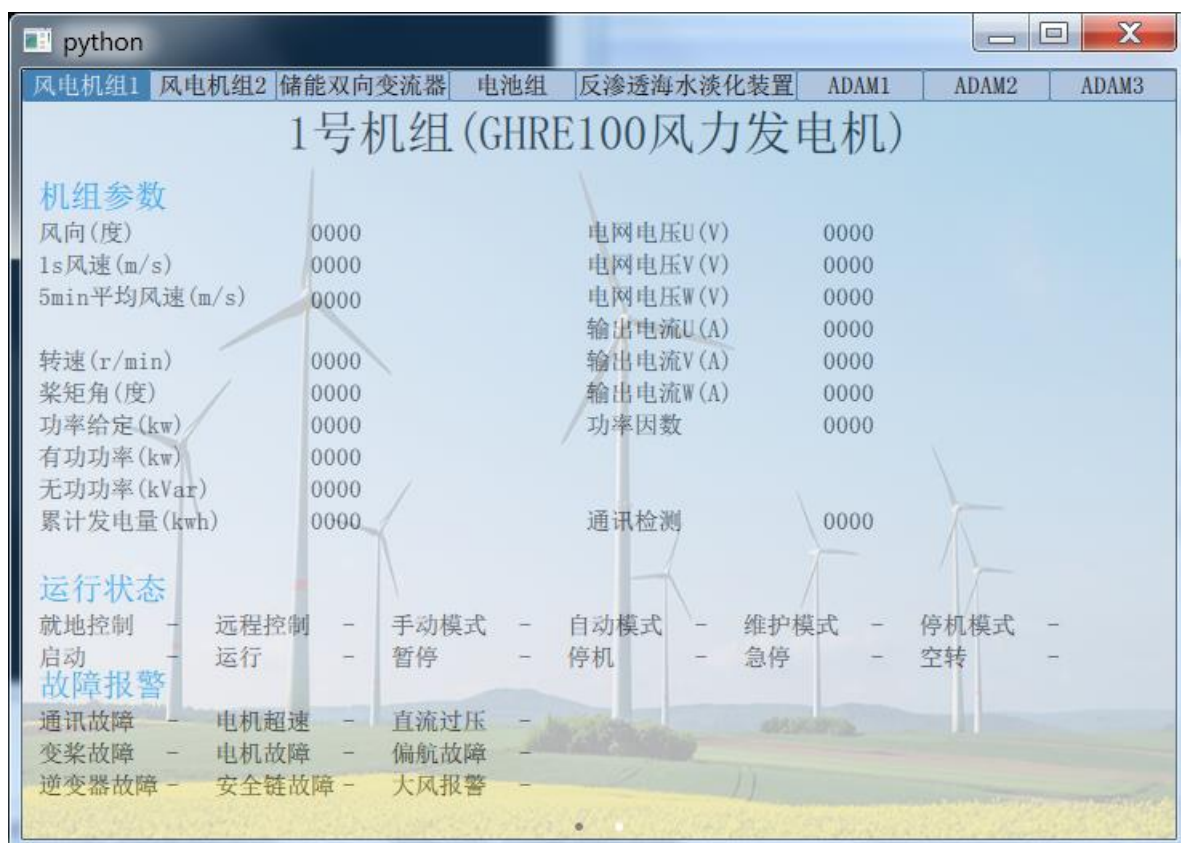


图 3-3 风电机组 1 界面设计效果图



图 3-4 风电机组 2 界面设计效果图



图 3-5 EMS 界面设计效果图

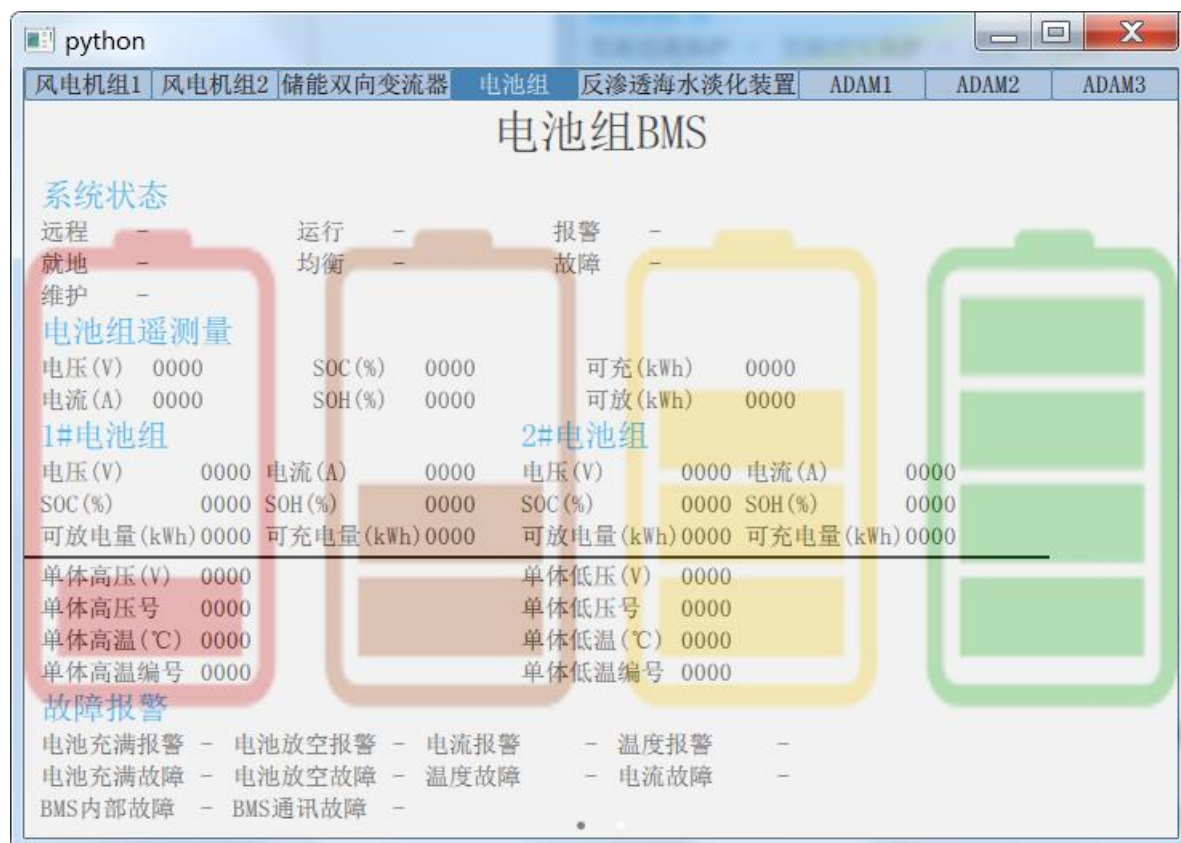


图 3-6 BMS 界面设计效果图



图 3-7 反渗透海水淡化装置界面设计效果图

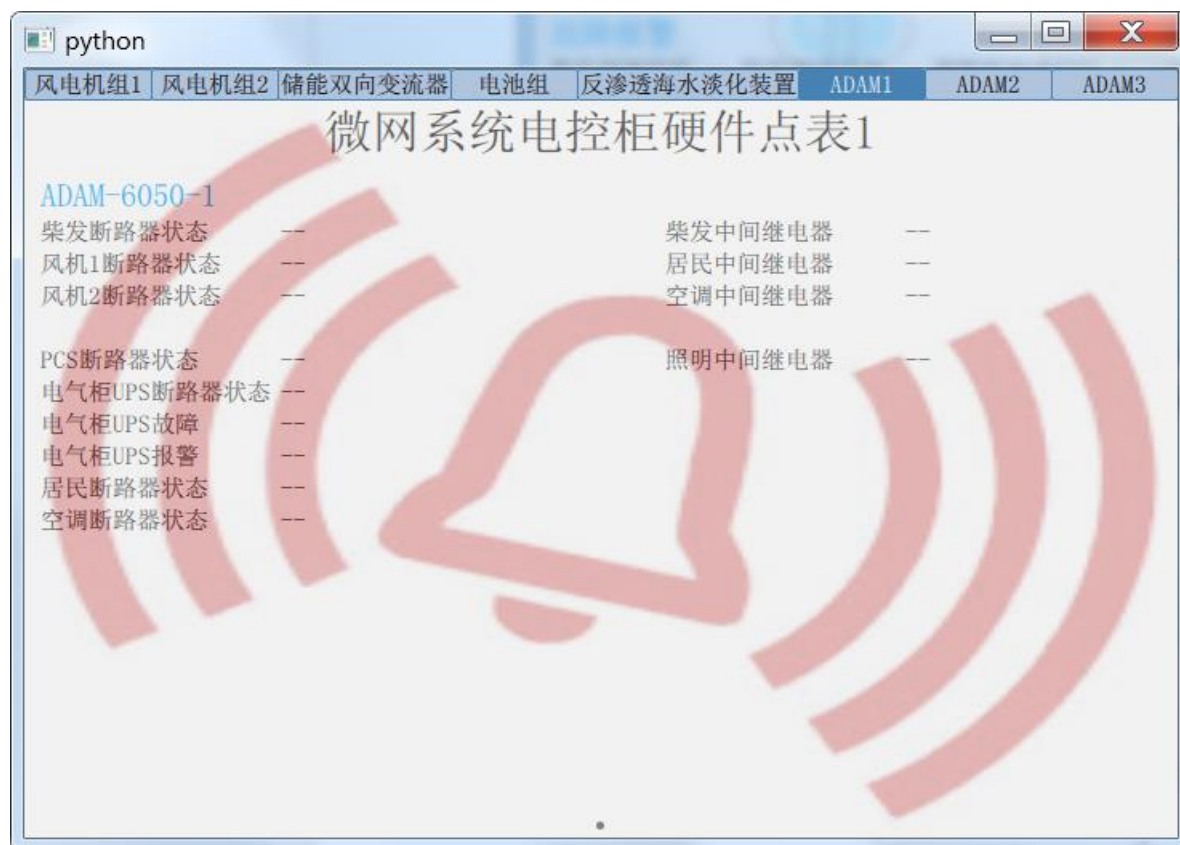


图 3-8 ADAM1 界面设计效果图



图 3-9 ADAM2 界面设计效果图



图 3-10ADAM3 界面设计效果图

4.2 数据展示软件设计

通过在 Python 主程序中引入 PyQt5.QtCore 的 QThread 模块，可以在 Python 脚本中对基于 qml 文件设计的图形用户界面中的变量进行值操作，如更新、修改界面中某个数值。这样使得基于 QML 设计的图形用户界面也能够及时动态地显示在 Python 脚本中进行的协议解析过程所得到的各项数据，从而实现利用图形界面动态展示解析得到的相关数据。简单的来说就是通过设计数据展示软件，使得 Python 主程序中指定的变量能实时地传值给 QML 界面中指定的变量，从而借助 QML 界面展示 Python 主程序中的变量值。

设计核心代码如下：

```
from PyQt5.QtCore import QThread

class DataSource(QThread):
    sigUpdate = pyqtSignal(str, str)

    def setProperty_wrap(self, prop, value):
        self.sigUpdate.emit(prop, value)
```

其中 setProperty_wrap(self, prop, value)函数用来执行 Python 主程序到 QML 文件界面的传值操作，prop 是在 QML 文件中定义了的接受传值的变量，value 是将被传过去的 Python 主程序中的变量值。

第五章 数据库和数据转发软件设计

5.1 数据库设计

5.1.1 设计方案

本软件要求将 Python 应用程序对网络数据封包进行协议解析后得到的各项有效数据转发至后台服务器的数据库。因此需要设计用来存储解析数据的数据库。在本设计中使用 MySQL 服务器的数据库来存储转发的数据^[21]，建立数据库时先使用.bat 文件启动 MySQL 服务器，再用 Navicat for MySQL 新建到 MySQL 服务器的连接（或者连接到已建立的 MySQL 服务器），然后就可以在服务器中创建数据库，或者在已建立的数据库内完成新建表，设计表，清空表或者删除表的一系列操作。

5.1.2 具体设计

先下载 MySQL 管理工具(Navicat for MySQL)

Navicat for MySQL 是一款辅助开发 MySQL 数据库的管理工具，其中的.bat 文件还可以用来启动和关闭 MySQL 服务器。在本设计中利用 Navicat for MySQL 方便地对 MySQL 数据库表进行操作。

首先使用.bat 文件启动 MySQL 服务器,打开 MySQL 管理工具.bat,输入 1,敲回车,选择启动服务器:

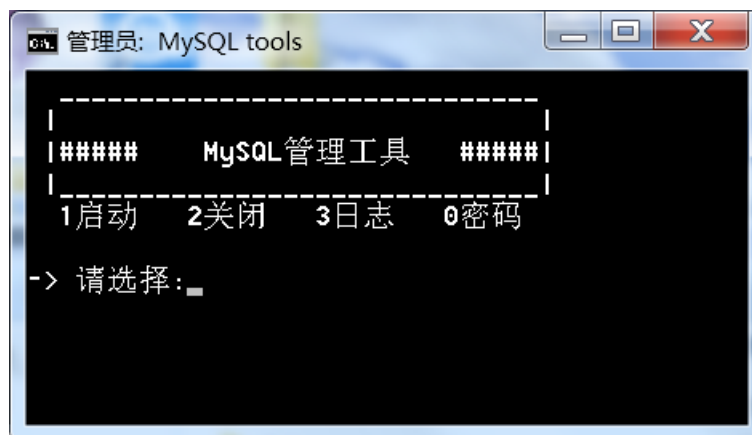


图 5-1 用.bat 管理工具启动 MySQL 服务器示意图(a)

如果服务器已经启动，会有如下提示，此时先关闭服务器再次启动即可：



图 5-1 用.bat 管理工具启动 MySQL 服务器示意图(b)

服务器启动成功后，打开 Navicat for MySQL 数据库辅助设计工具，选择左上角的连接：

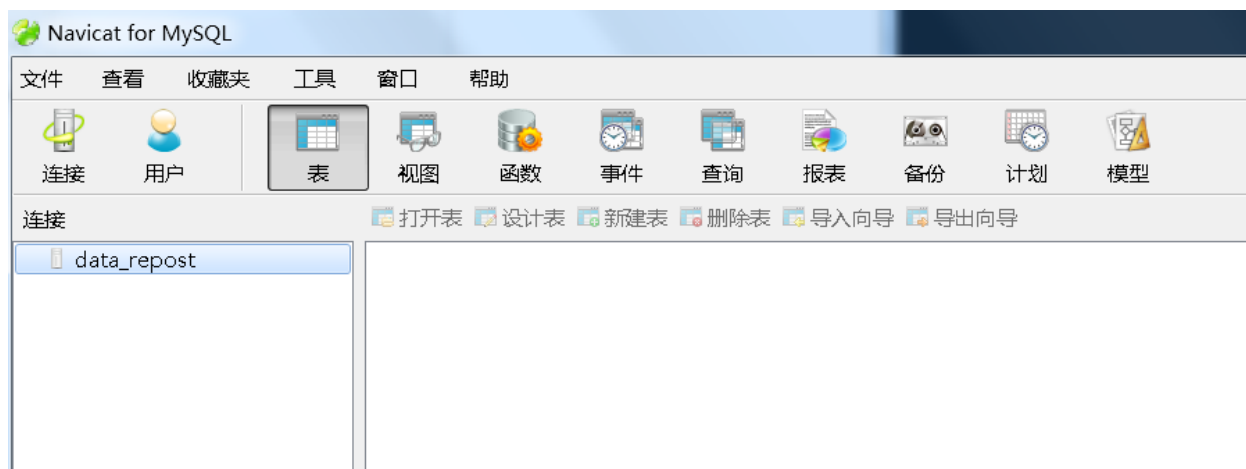


图 5-1 用 Navicat for MySQL 设计数据库示意图(a)

会出现如下界面，输入连接名，在本机建立数据库就输入 `localhost`，密码无需输入，端口和用户名保持默认即可，如下图所示：



图 5-1 用 Navicat for MySQL 设计数据库示意图(b)

然后点击连接测试按钮，如果刚刚没有用.bat 启动 MySQL 服务器，会像下图一样提示连接失败：

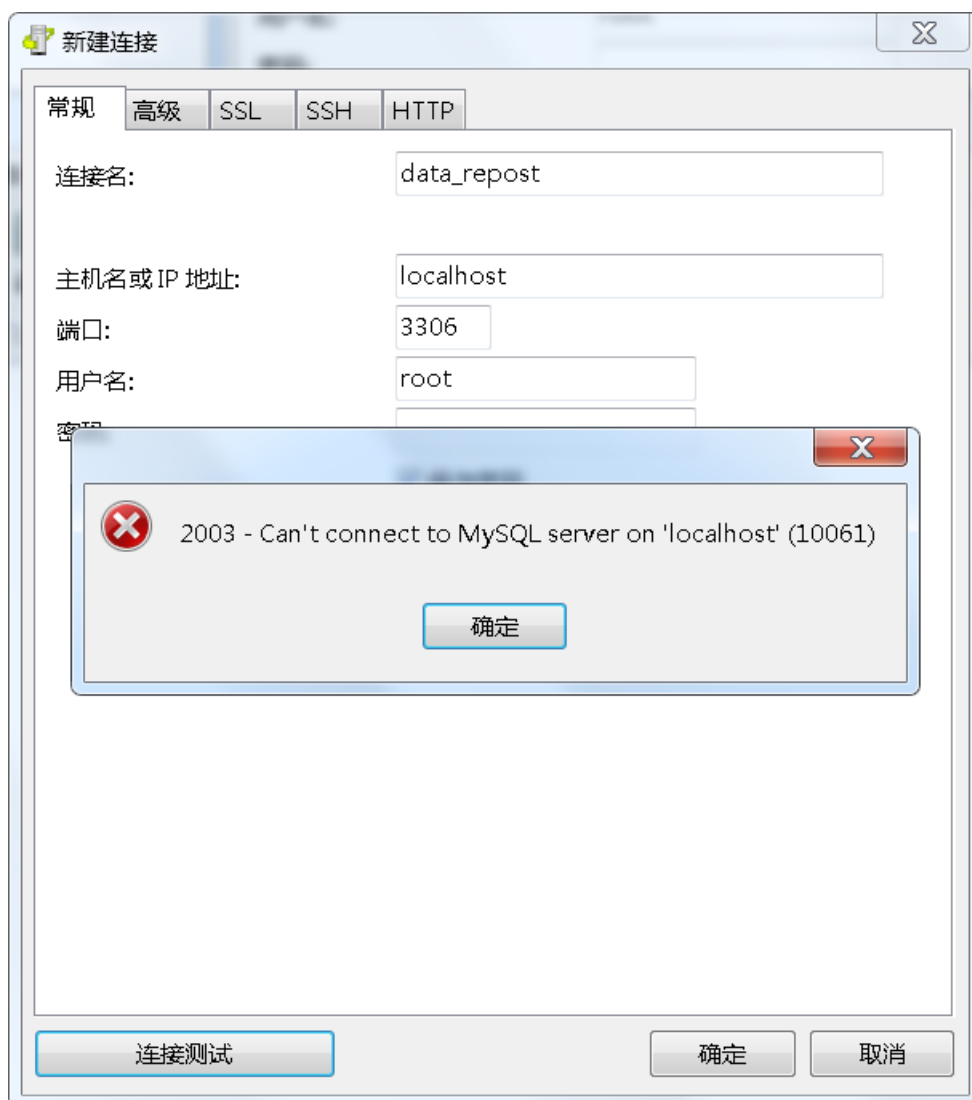


图 5-1 用 Navicat for MySQL 设计数据库示意图(c)

连接测试成功后，点确定，就可以看见新建的连接出现在左边的空白栏处：

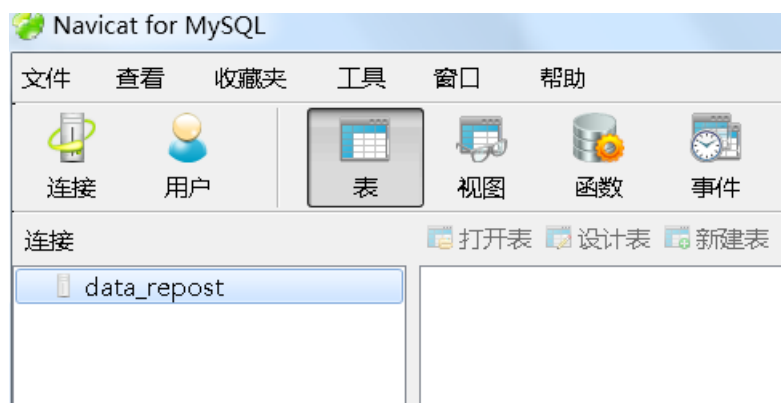


图 5-1 用 Navicat for MySQL 设计数据库示意图(d)

双击打开该连接如下：



图 5-1 用 Navicat for MySQL 设计数据库示意图(e)

`data_repost` 下显示的这些都是 MySQL 数据库，它们保存在 MySQL 目录下的 `data` 文件夹里，可以鼠标右键连接名选择新建数据库，也可以直接在 `data` 文件夹下新建一个文件夹，MySQL 服务器就会把它作为一个新的数据库。

双击其中的 `equipment` 数据库，可以在右边看到该数据库中已经建立好的表：



图 5-1 用 Navicat for MySQL 设计数据库示意图(f)

右键点击表名，可以完成新建表，设计表，清空表或者删除表等一系列操作，十分方便：

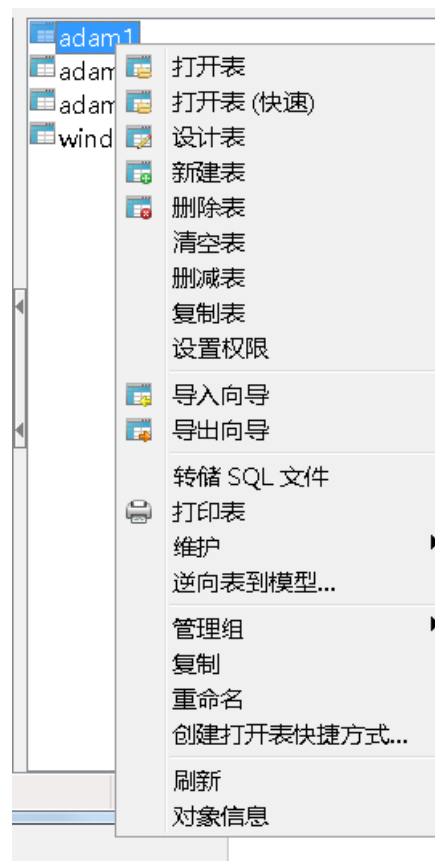


图 5-1 用 Navicat for MySQL 设计数据库示意图(g)

右键单击 adam2，选择设计表，就可以在 Navicat 对 MySQL 后台服务器 equipment 数据库中的表 adam2 进行设计，它被用来存储本软件解析得到的微网系统电控柜硬件表 ADAM-6020-2 设备的数据，我对该表的设计如下图所示，表中有一个主键 id，用来标识表中存储的是解析软件第几次发来的数据，要注意在同一个表中，设计主键 id 的数值不能重复，否则转发数据时软件会报错。该表中其他位置存储的都是 ADAM2 设备的各个工作状态参数：

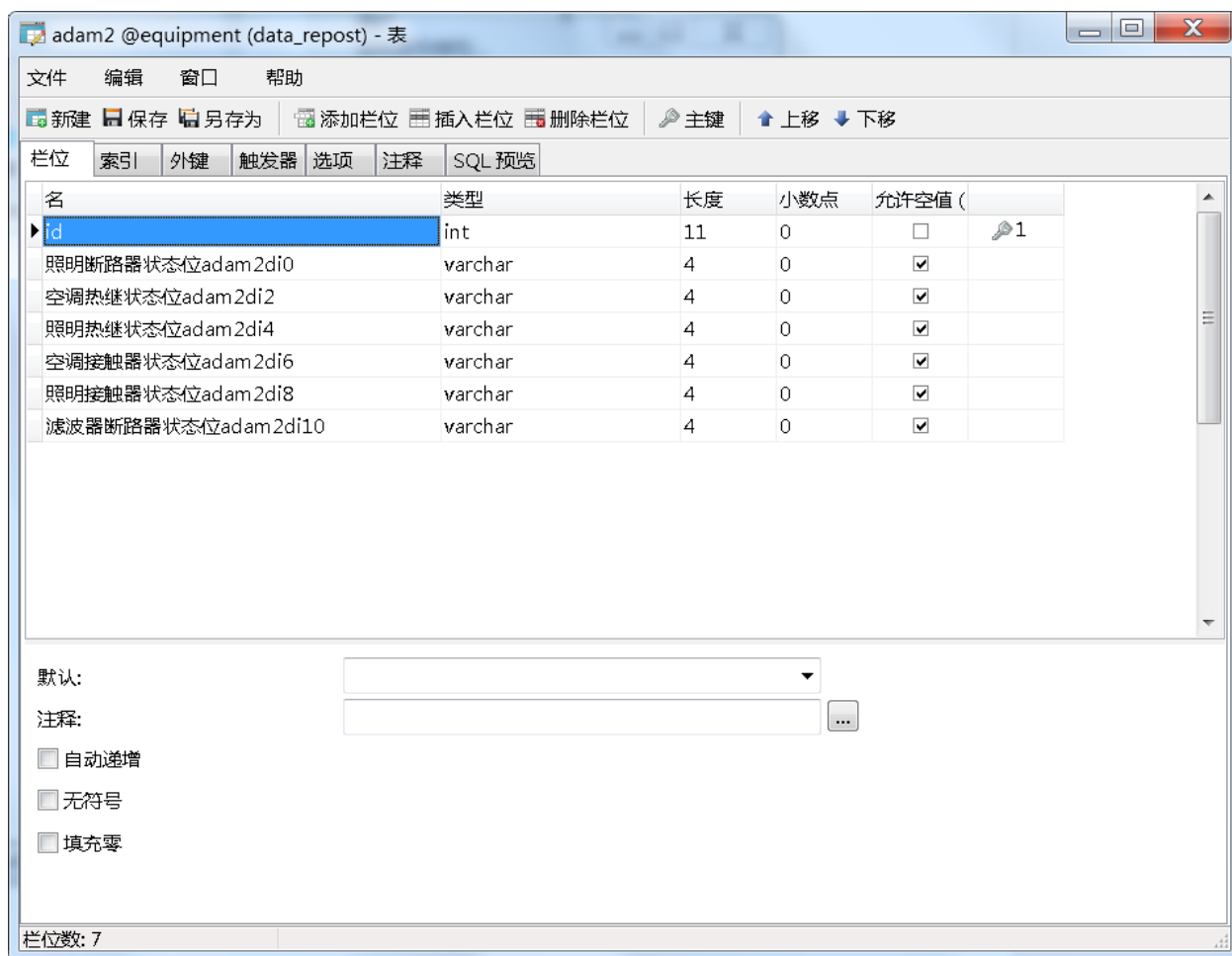


图 5-2 equipment 数据库内表 adam2 设计示意图

本软件设计了如下的 MySQL 数据库表文件：adam1.frm, adam1.ibd, adam2.frm, adam2.ibd, adam3.frm, windturb1.ibd, ibdata1，它们分别用来转存故障报警点表设备 1，故障报警点表设备 2，故障报警点表设备 3 和风电机组 1 的解析数据。如下是 equipment 数据库内表 windturb1 的设计示意图：

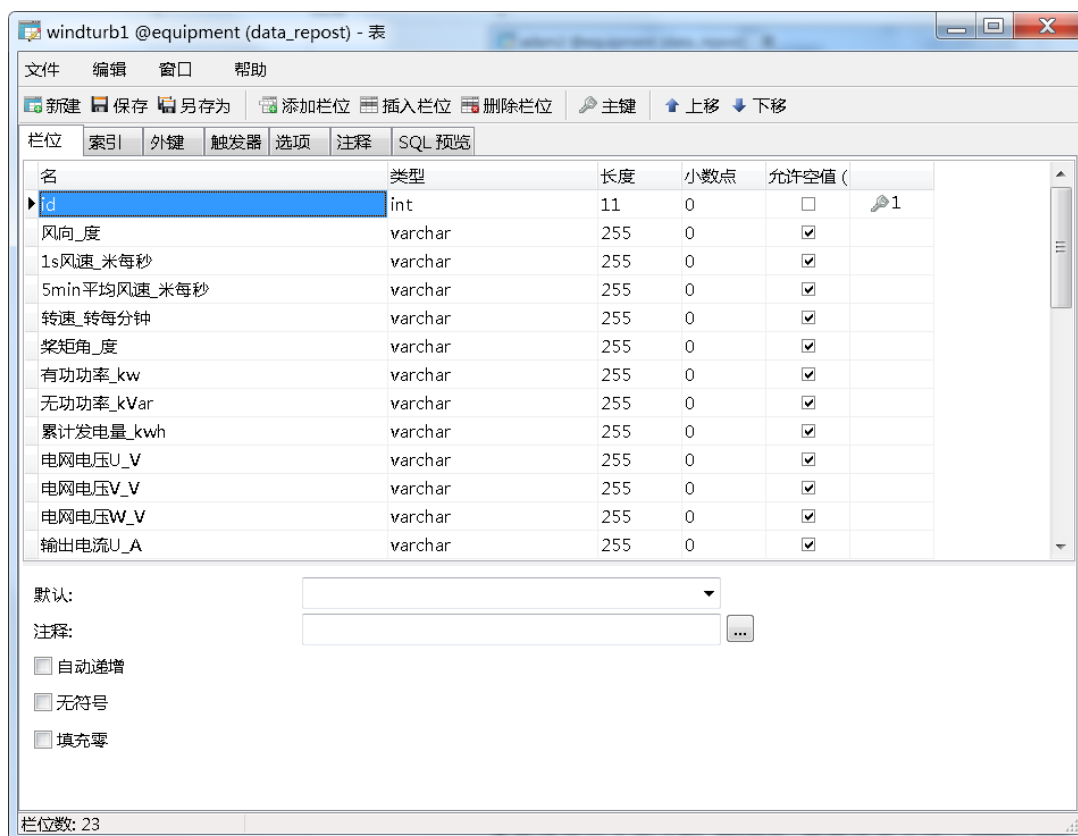


图 5-3 equipment 数据库内表 windturb1 设计示意图(a)

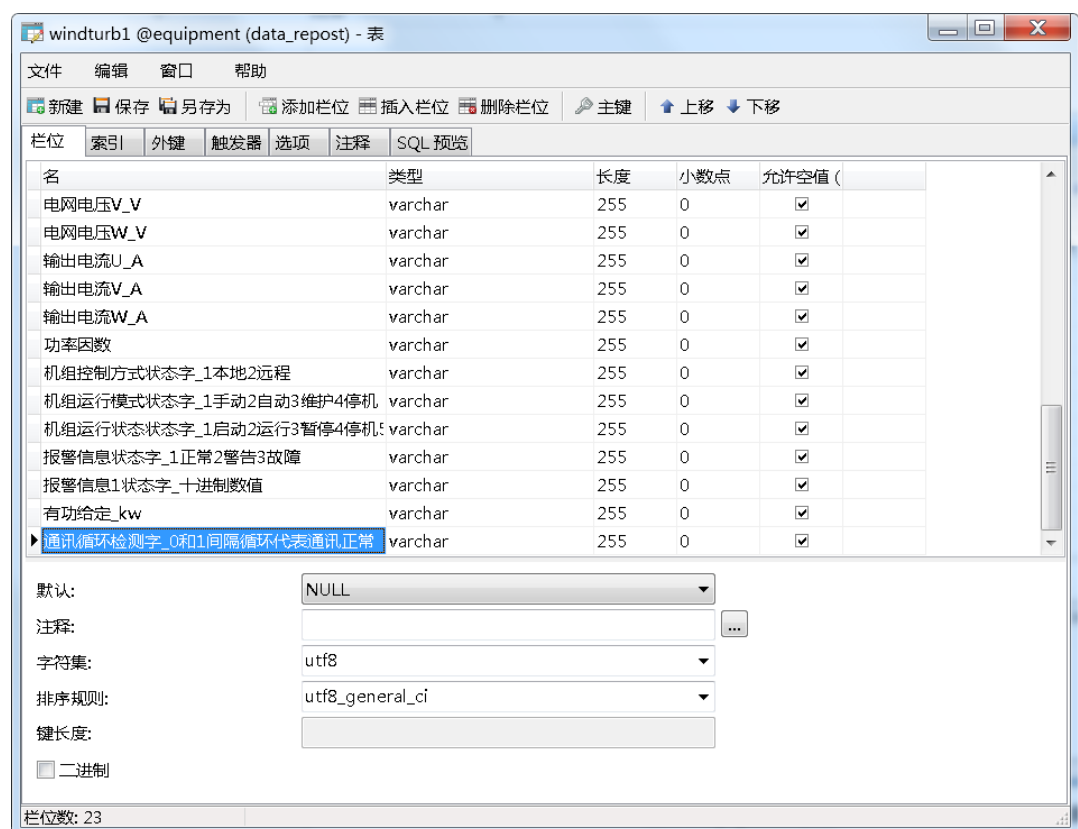


图 5-3 equipment 数据库内表 windturb1 设计示意图(b)

5.2 数据转发软件设计

5.2.1 设计方案

本软件使用 MySQL 后台服务器数据库中的表来转存解析得到相应设备的各个数据，为此需要在主程序文件中编写数据转发软件，在解析得到有效数据的同时将其转发至 MySQL 服务器的数据库表中。Pymysql 是 Python3.6 中用来对 MySQL 数据库进行操作的模块^[22]，因此可以在 Python 主程序文件中引入 pymysql 模块，使用 pymysql.Connect() 函数帮助 Python 脚本连接到已建立的 MySQL 服务器的数据库，之后就可以通过 Python 编程实现对该数据库中各表的数据内容进行插值，替换或者更新操作，从而实现解析数据的转发工作^[23]。

5.2.2 具体设计

在数据转发之前先要用 .bat 文件启动 Mysql 服务器，该过程已在 5.1.2 节中进行了详细描述。如果在 MySQL 服务器关闭的情况下数据是没法转发数据到数据库的（因为无法连接到数据库服务器）。

在 Python 主程序文件中引入 pymysql 模块，引入方式为：import pymysql

引入 pymysql 模块后，就可以在 Python 脚本中连接到 MySQL 数据库了，连接使用的代码为：

```
db = pymysql.Connect(host='localhost', port=3306, user='root', passwd='', db='equipment',
charset='utf8')
```

其中 host, port, user, password 都和 3.3.2 小节中新建连接时使用的值一样，db 是需要连接到的 MySQL 数据库名称，charset 是设定数据编码方式。

执行该语句，如果没有返回错误提示，就说明已经连接成功了。接下来就可以在 Python 程序中转发解析得到的数据要刚刚连接上的数据库了。

转发数据的代码为：

```
cursor = db.cursor()
sql = "REPLACE INTO ADAM2 (id,照明断路器状态位 adam2di0,空调热继状态位 adam2di2,照明热继状态位 adam2di4,\
空调接触器状态位 adam2di6,照明接触器状态位 adam2di8,滤波器断路器状态位 adam2di10) VALUES (%d, '%s', '%s',\
'%s', '%s', '%s', '%s', '%s')"
```

其中 ADAM2 是数据库 equipment 中准备接受转发数据的表，(id,照明断路器状态位 adam2di0,空调热继状态位 adam2di2,照明热继状态位 adam2di4,空调接触器状态位 adam2di6,照明接触器状态位 adam2di8,滤波器断路器状态位 adam2di10)都是表 adam2 中的设计的数据内容，(%d, '%s', '%s', '%s', '%s', '%s', '%s', '%s')和这些数据内容依次一一对应，表示数据是以何种变量格式转发。

REPLACE 表示如果两次转发的 id（主键）数值相同，那么前一次转发的数据将被完全删除，只有第二次转发的数值能保存在表内。

而如果换做 INSERT 就是普通的插值并且如果两次转发的 id（主键）数值一样，程

序会报错。

在设计数据转发软件时，我遇到了两个实际问题，下面简单介绍一下这两个问题并说明我的解决方式。

1、如何规定向数据库内某一张表中转发数据的次数，即 id 值达到上限后自动停止转发？

我最初的解决方式是在主程序文件 QmlView.py 中定义一个 global 变量 flag，每执行数据转发一次就将其加一，直到达到规定的次数就不再执行转发代码。这样做在 QmlView 主程序文件中的确可行，但是当我执行 main.py 主程序文件时软件就没有用了，因为 QmlView.py 中定义的全局变量 flag 在 main.py 中不再生效了。因此我又改进了解决方式，还是用标记变量 flag，但是这次 flag 不是定义在 QmlView.py 中，而是利用 config 文件夹内的 settings.py 作为配置文件，在其中定义全局变量类，然后在主程序文件 main.py 和 QmlView.py 中引用这个配置文件即可，引用方式为 import settings.py。这样主程序文件都可以使用在该配置文件中定义的全局变量类中的 flag 变量了。代码设计如下：

```
class global_flag

    flag =0

    def sv6(flagvalue):

        global_flag.flag = flagvalue

    def gv6():

        return global_flag.flag
```

2、对于一次性发完全部数据的设备，使用 REPLACE INTO 没有问题，但是如果是需要分三次才能转发完全部数据的设备（比如风电机组 1，因为其一轮数据需要前后接受三次才能解析完）使用 REPLACE INTO 就会发现数据库中的表 windturb1 中只存在最后一次转发的数据，为什么？

因为三次转发数据共用一个 id，转发到表内同一栏，第二次转发会删除第一次转发的全部数据，第三次转发也会删除之前转存在同一个 id 栏内的全部数据。所以用 REPLACE INTO 是行不通的。

但是通过学习 pymysql 模块的函数表，我知道了除了 INSERT/REPLACE INTO 外，pymysql 模块还允许 Python 编程使用 UPDATE 更新数据库表某一 id 栏内的指定数据内容。

使用方式为：

```
UPDATE name
SET property1=setting1,property2=setting2,...
WHERE key=certain_value;
```

上面这段代码用来更新表名为 name 的数据库表, 将表中 property1 值更新为 setting1, 将表中 property2 值更新为 setting2, 并且只更新主键为 certian_value 的那些栏内的 property1 和 property2 值。

综上, 一个设备在同一个 id 下多次转发数据时软件流程图设计如下:

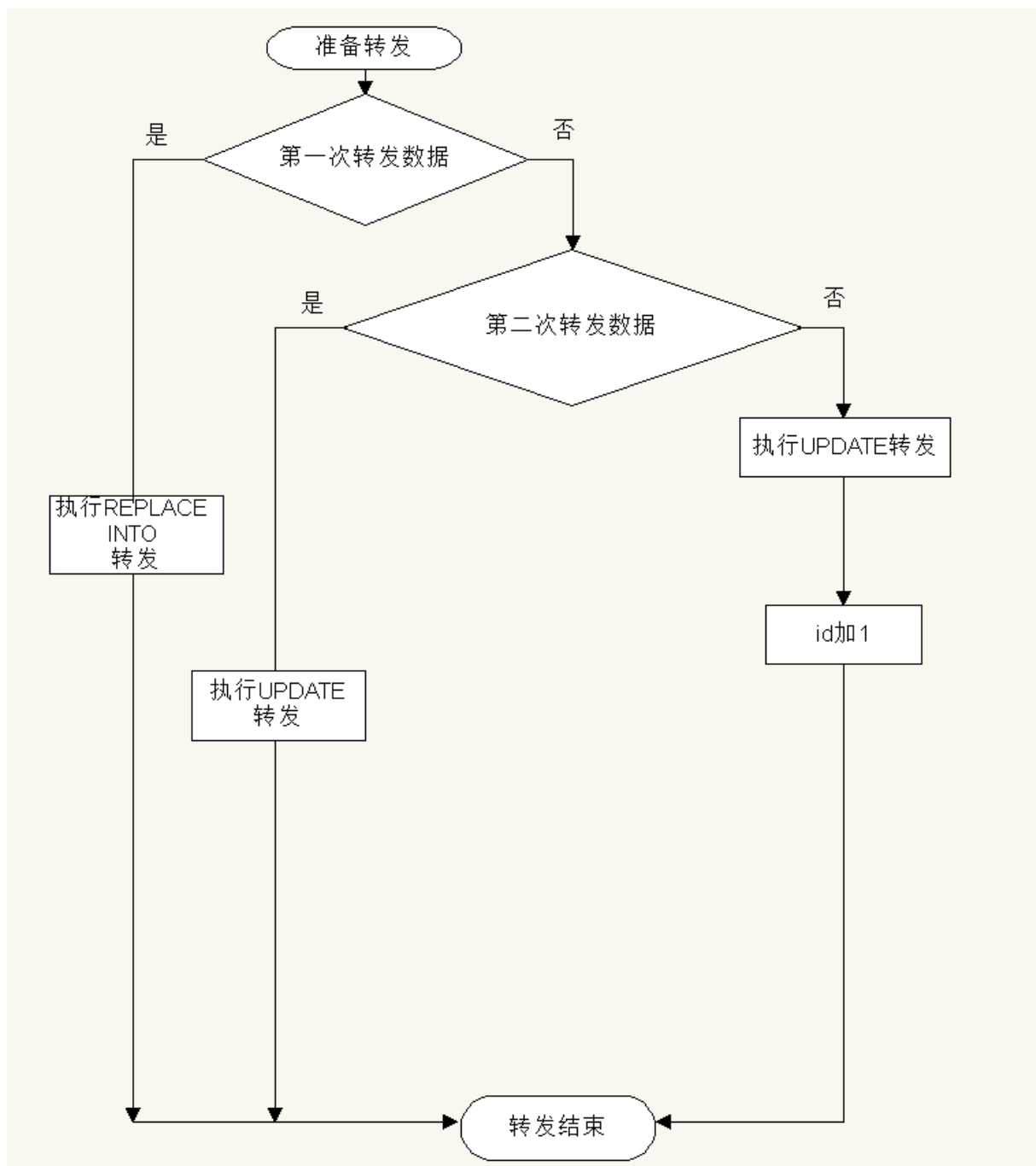


图 5-4 同一个 id 下需 3 次转发数据时软件流程图

因此对 windturb1 表（风电机组 1）的数据转发软件重新设计代码如下：

第一次数据转发：

```

if config.settings.gv6() <= config.settings.times:
    dbw11 = pymysql.Connect(host='localhost', port=3306, user='root', passwd='', db='equipment',
charset='utf8')
  
```

```

cursorw11 = dbw11.cursor()    #不能有\ ( ) 出现在 id 后的名字里!
sqlw11 = "REPLACE INTO windturb1 (id, 机组控制方式状态字_1 本地 2 远程, 机组运行模式状态字_1 手动 2 自
动 3 维护 4 停机, \
        机组运行状态状态字_1 启动 2 运行 3 暂停 4 停机 5 急停 6 空转, 报警信息状态字_1 正常 2
警告 3 故障, 风向_度, 1s 风速_米每秒, \
        5min 平均风速_米每秒, 转速_转每分钟, 桨矩角_度, 报警信息 1 状态字_十进制数值)\
        VALUES( % d, '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s')\"
data = ( config.settings.gv6(), wtb_1_control_method, wtb_1_running_mode, wtb_1_running_state,
wtb_1_message_warn0,
        wtb_1_wind_dir10min, wtb_1_wind_velo1s, wtb_1_wind_velo5min, wtb_1_gen_rot, wtb_1_vane_ang,
wtb_1_message_warn1)
cursorw11.execute(sqlw11%data)
dbw11.commit()    #注意不能都命名为 sql 要有区分比如 sql1, 2, 3
else:
    return

```

第二次数据转发:

```

if config.settings.gv6() <= config.settings.times:
    dbw13 = pymysql.Connect(host='localhost', port=3306, user='root', passwd='',
                             db='equipment', charset='utf8')

    cursorw13 = dbw13.cursor()

    sqlw13 = "UPDATE windturb1 SET 有功功率_kw=(%s),无功功率_kVar=(%s),\
            累计发电量_kwh=(%s), 电网电压 U_V=(%s),电网电压 V_V=(%s),电网电压 W_V=(%s),输出电流 U_A=(%s),
输出电流 V_A=(%s),输出电流 W_A=(%s),功率因数 =(%s)\
            WHERE id=(%s)"

    cursorw13.execute(sqlw13, (\
        wtb_1_actpow, wtb_1_reactpow, wtb_1_elec_gened, wtb_1_voltageU, wtb_1_voltageV, wtb_1_voltageW, \
        wtb_1_currentU, wtb_1_currentV, wtb_1_currentW, wtb_1_powfactor, config.settings.gv6()))

    dbw13.commit()

```

第三次数据转发:

```

if config.settings.gv6() <= config.settings.times:
    dbw12 = pymysql.Connect(host='localhost', port=3306, user='root', passwd='',
                             db='equipment', charset='utf8')

    cursorw12 = dbw12.cursor()

    sqlw12 = "UPDATE windturb1 SET 有功给定_kw= (%s),通讯循环检测字_0 和 1 间隔循环代表通讯正常=
(%s)WHERE id=(%s)"    # id=(%d) wrong!

    cursorw12.execute(sqlw12, (wtb_1_powset, wtb_1_communication_test, config.settings.gv6()))

    dbw12.commit()

    config.settings.sv6(config.settings.gv6() + 1)

```

else:

if config.settings.gv6() == config.settings.times+1:

 print()

 print('解析得到的风电机组 1 的数据已转发至 MySQL 后台服务器 equipment 数据库的表 windturb1
中!')

 config.settings.sv6(config.settings.gv6() + 1)

return

else:

return

第六章 软件测试

6.1 协议解析和界面展示功能测试

双击 main.py 运行软件程序，会弹出文件选择对话框，命令窗口提示我们选择需要解析的网络封包文件：

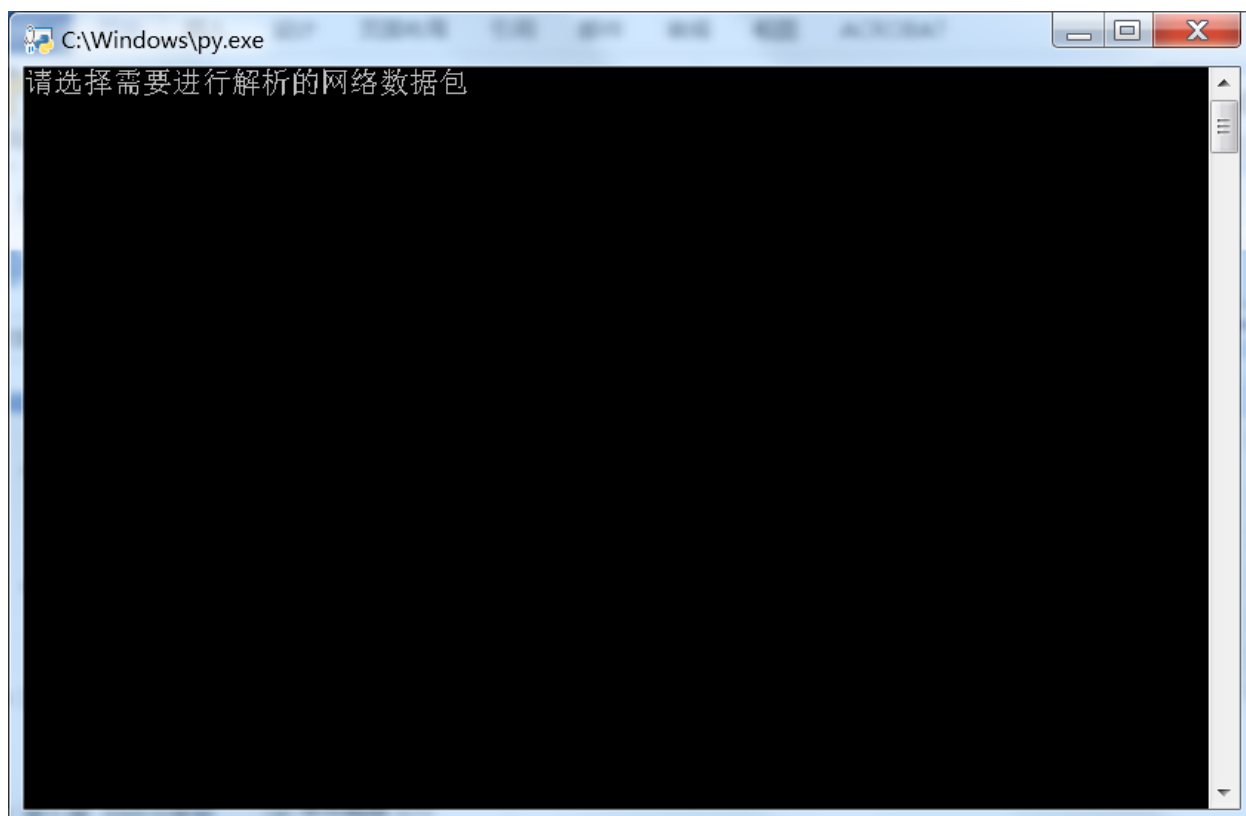


图 6-1 程序初始化

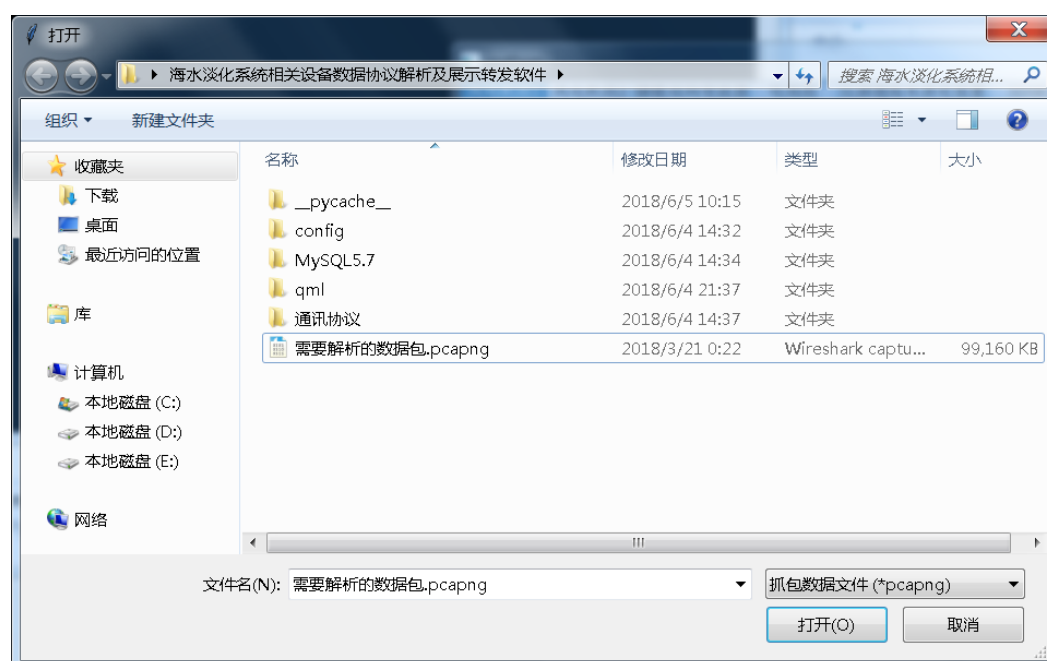


图 6-2 使用文件选择对话框

选择需要解析的数据包打开后，协议解析程序开始工作，并将解析出来的各项有效数据显示在对应设备的图形界面中：

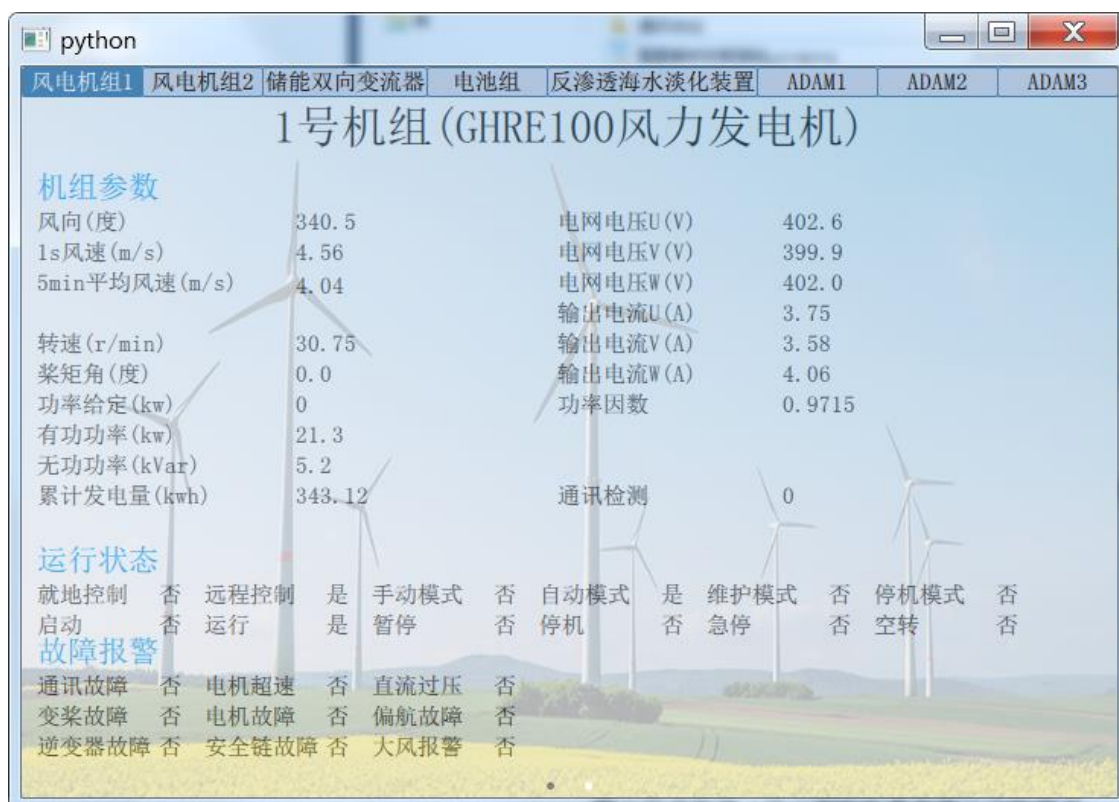


图 6-3 风电机组 1 协议解析和界面展示功能测试



图 6-4 风电机组 2 协议解析和界面展示功能测试



图 6-5 EMS 协议解析和界面展示功能测试

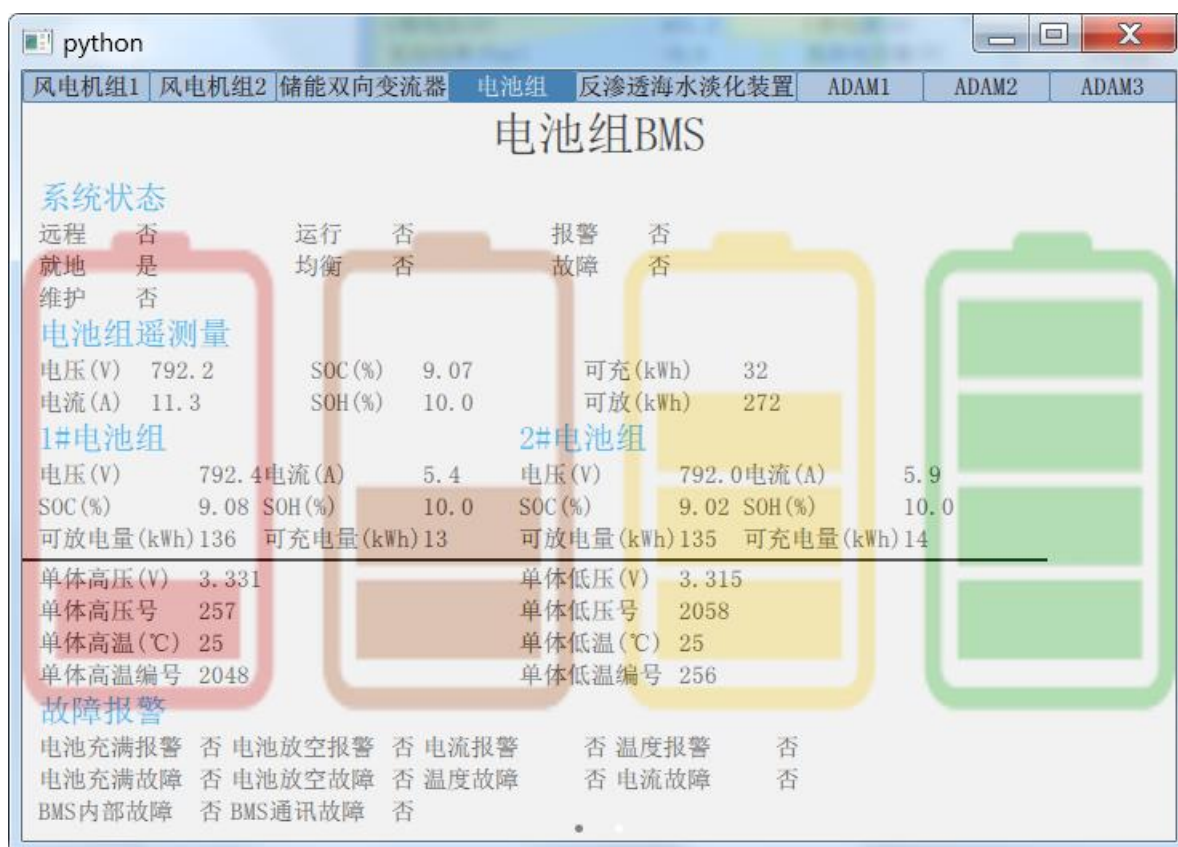


图 6-6 BMS 协议解析和界面展示功能测试



图 6-7 反渗透海水淡化装置协议解析和界面展示功能测试

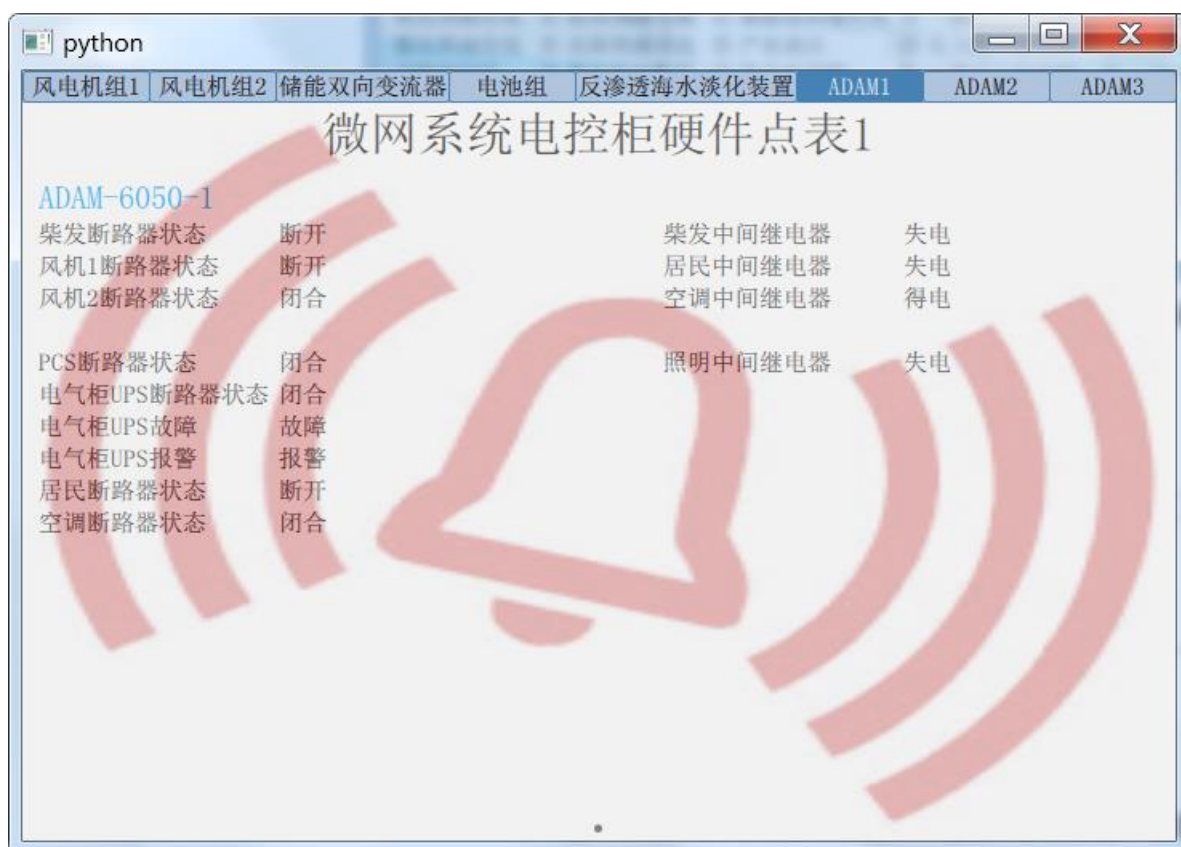


图 6-8 ADAM1 协议解析数据和界面展示解析数据示意图

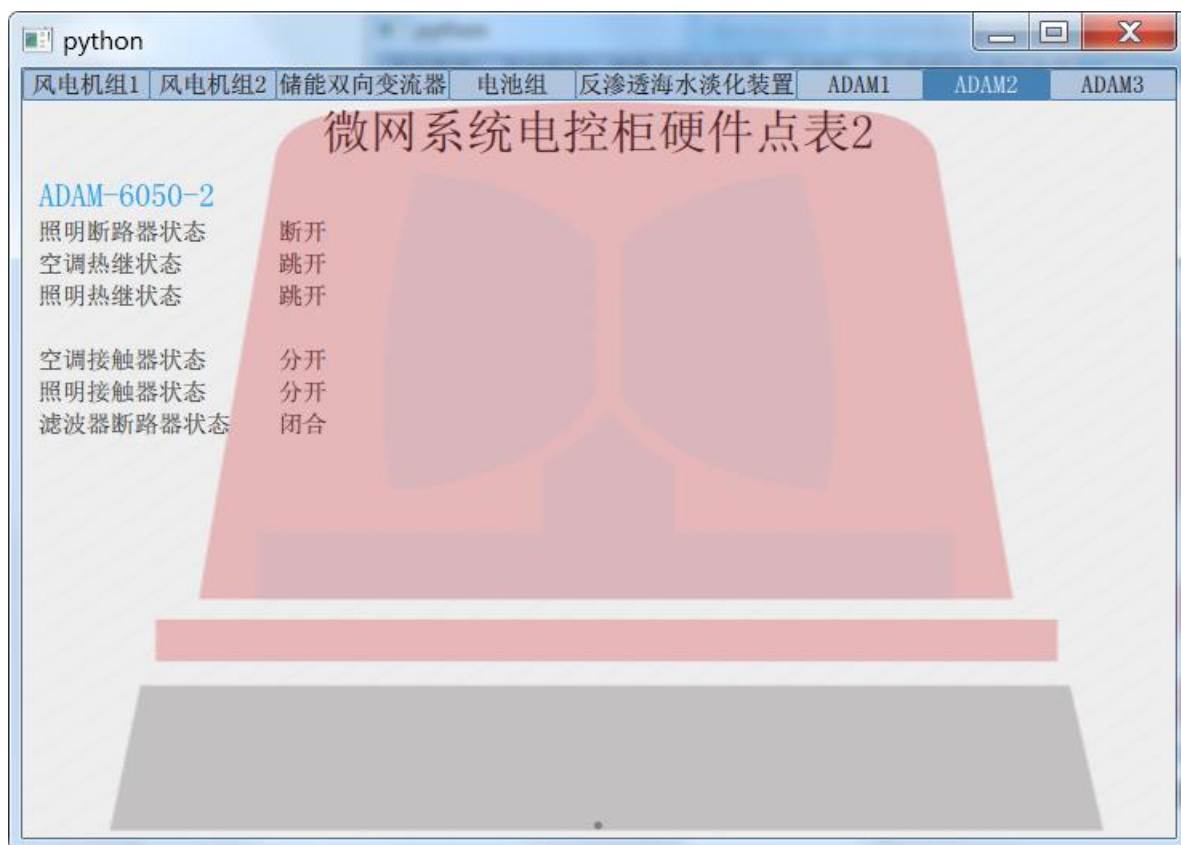


图 6-9 ADAM2 协议解析数据和界面展示解析数据示意图



图 6-10 ADAM3 协议解析数据和界面展示解析数据示意图

可以发现解析得到的风电机组 1 设备的各项工作状态参数成功地转存到了 MySQL 后台服务器中的 equipment 数据库中。

再分别查看 adam1, adam2 和 adam3 的表文件：

id	柴油发电机状态位	风机1断路器状态位	风机2断路器状态位	PCS断路器状态位	电气柜UPS断路器状态位	电气柜UPS故障状态位	电气柜UPS报警状态位	层民断路器状态位	空调断路器状态位	柴发中间继电器状态位	层民中间继电器状态位	空调中间继电器状态位
1	0	0	1	1	1	0	0	0	1	0	0	1
2	0	0	1	1	1	0	0	0	1	0	0	1
3	0	0	1	1	1	0	0	0	1	0	0	1
4	0	0	1	1	1	0	0	0	1	0	0	1
5	0	0	1	1	1	0	0	0	1	0	0	1
6	0	0	1	1	1	0	0	0	1	0	0	1
7	0	0	1	1	1	0	0	0	1	0	0	1
8	0	0	1	1	1	0	0	0	1	0	0	1
9	0	0	1	1	1	0	0	0	1	0	0	1
10	0	0	1	1	1	0	0	0	1	0	0	1

图 6-10 ADAM1 数据转发功能测试

id	照明断路器状态位	空调热继电器状态位	照明热继电器状态位	空调接触器状态位	照明接触器状态位	滤波器断路器状态位
1	0	0	0	0	0	1
2	0	0	0	0	0	1
3	0	0	0	0	0	1
4	0	0	0	0	0	1
5	0	0	0	0	0	1
6	0	0	0	0	0	1
7	0	0	0	0	0	1
8	0	0	0	0	0	1
9	0	0	0	0	0	1
10	0	0	0	0	0	1

图 6-11 ADAM2 数据转发功能测试

id	浪涌保护状态位	220V供电总空开状态位	220_24V开关电源状态位	24V空开状态位	散热_除湿空开状态位	控制柜UPS1报警状态位	控制柜UPS2报警状态位	海淡负载断路器状态位
1	1	1	1	1	1	0	0	1
2	1	1	1	1	1	0	0	1
3	1	1	1	1	1	0	0	1
4	1	1	1	1	1	0	0	1
5	1	1	1	1	1	0	0	1
6	1	1	1	1	1	0	0	1
7	1	1	1	1	1	0	0	1
8	1	1	1	1	1	0	0	1
9	1	1	1	1	1	0	0	1
10	1	1	1	1	1	0	0	1

图 6-12 ADAM3 数据转发功能测试

可以发现解析得到的 1-3 号故障报警设备的各项工作状态参数也都成功地转存到了 MySQL 后台服务器中的 equipment 数据库中。可以通过连接服务器在后台查看这些数据。数据转发功能测试效果良好。

第七章 总结与展望

通过几个月的设计与反复测试、修改，海水淡化设备相关协议解析及数据转发软件的设计已日趋完善，从最初的还不能完全理解项目要求，到如今的项目所要求的各项功能都已经实现，这几个月的毕业设计工作不仅对提高了我的软件编程能力，还使我对软件设计的理解更加深刻。我在完成软件设计的过程中犯了很多编程错误，碰到了很多困难，但是在毕设指导老师的悉心帮助下，在我止于至善的不懈追求中，都得到了修正和解决。

最终本设计完成了课题任务要求实现的全部工作：

- 1、利用网络封包分析程序获取 PLC 及其他设备与工控机间的通信数据
- 2、编写各种协议解析程序，对获取到的通信数据按相关协议进行解析，得到有效数据
- 3、编写人机交互子界面，用于展示解析得到的相关数据
- 4、编写后台交互子程序，将解析得到的数据传至后台服务器

这是我第一次编写多文件组织的软件程序：基于 Python 的协议解析程序要和 QML 界面设计文件还有 MySQL 数据库共同完成设计要求，这是因为 Python 语言很适合编写协议解析程序，但是界面设计功能没有利用 Qt 库的 QML 文件强大，而实现数据转发功能需要设计和使用 MySQL 数据库。

在设计过程中，我发现 Python 语言的确很适于编写多文件组织架构中的主程序文件。Python 主程序文件（可执行脚本）通过引入各种丰富而强大的库模块，可以像粘合剂一样将 QML 界面设计文件和 MySQL 数据库表文件与主体编程结合到一起，让它们作为一个整体为设计需求服务，使得各个文件、甚至各个不同语言编写的文件交织在一起，和谐地发挥着自己应尽的那一份作用，最终谱写出一曲功能宏大的交响曲。

在使用 Python 编程时，我还被它那简洁、工整的缩进格式吸引住了，和之前熟悉的 C++ 不一样，在 Python 脚本执行时还会对软件代码的缩进情况进行检查，如果用错了缩进，在 C++ 中只会导致界面看起来乱糟糟的，但是在 Python 中会直接导致软件不能正常运行。因此，我用 Python 语言编写出来的程序代码，从上到下都是整整齐齐，看起来赏心悦目，并且在熟悉了 Python 后，我今后使用其他语言编程时，也会注意保持编码格式的规范工整，因此可以说在这个项目的开发过程中，我的编程习惯也得到了良好的培养。

虽然课题任务所规定的内容和指标我已全部完成和实现，但是我希望未来能够继续完善这款软件，因为软件的主程序文件（可执行文件）是 py 文件，程序需要在安装了 Python 的软件环境下运行，我希望能够将软件整体打包成一个可以直接在 Windows 系统下运行的 exe 可执行文件。展望未来的学习和研究道路，这几个月的毕业设计过程为我增添了信心，它让我明白通过日复一日地努力学习，积累知识，不管多么复杂的问题都有可能得到解决。

致 谢

随着毕业设计进入尾声，大学四年生涯即将划上句号，想起往昔在教室勤奋学习的时光，觉得很快乐也很充实，在这四年期间我得到了很多老师和同学的帮助和支持，也受到了东南大学浓郁的学术氛围的影响。在论文即将完成之际，我想表达对大学的感谢，感谢东南大学为我提供了四年来这么丰富多彩的学习生活，使我有幸能受到这么多优秀的老师的谆谆教诲和悉心关怀，使我能接触到这么多志同道合、热爱学习和生活的同学和朋友。

我想要感谢所有教导过我，帮助过我的老师，我每次向你们请教问题，你们都能耐心认真地为我解答，是你们用精湛的专业知识和教书育人的高尚品德为我点亮学习的道路，让我热爱在大学学习的每一门功课，对学习生活充满了信心。

还要感谢我身边的同学和朋友们，在我心情低落的时候，你们总是能鼓励我或是帮助我，我们在四年的大学生活中朝夕相伴，共同进步和成长，一起走向更辉煌的明天。当然也不能忘了父母和长辈在这四年对我的照顾和关心，假期回到家中，他们无微不至地照顾我的生活，鼓励我锻炼身体，为我烹饪美味的菜肴，他们对我的爱如此深沉，时刻都能让我感到温暖。

在此刻，论文即将完成，回顾过去几个月毕业设计所带给我的充实和忙碌，我最想要感谢的是我的指导老师黄永明老师，从毕业设计选题到中期检查再到今天毕设即将完成，黄老师一直都给予了我细心的指导和有力的支持，为我提供了大量宝贵的意见和坚实的帮助，黄老师不但在毕设上对我的帮助很大，他在学术研究的道路上的勤勤恳恳和一丝不苟，他对待职业的无私奉献精神和对待生活的积极乐观，都永远是我学习的榜样和进步的动力。请允许我在此向尊敬的黄永明指导老师表达深深的谢意！

在毕设论文完成过程中，还要感谢陈峥学长的热心关怀，他以良好的学习态度和扎实的学科知识对我编程思想和方法上提供了帮助。

最后，感谢东南大学所有老师们四年来对我的教导和帮助，再次感谢黄永明老师给予我的精心指导和为我们毕业论文付出的巨大心血和精力，正是在黄老师的认真指导和大力帮助下我才能顺利地完成这篇毕业论文！

参考文献

- [1] 邵天宝,王生辉,吴水波,赵河立.风能反渗透海水淡化技术研究进展[J].中国给水排水,2014,30(06):25-27.
- [2] 冯宾春,赵卫全.风能海水(苦咸水)淡化现状[J].水利水电技术,2009,40(09):8-11.赵敬恩.基于风能利用的海水淡化技术途径研究[D].中国科学院研究生院(工程热物理研究所),2013.
- [3] 赵敬恩.基于风能利用的海水淡化技术途径研究[D].中国科学院研究生院(工程热物理研究所),2013.
- [4] 狄博,王晓丹.基于 Python 语言的面向对象程序设计课程教学[J].计算机工程与科学,2014,36(S1):122-125.
- [5] 刘畅. Python 代码风格对软件维护的影响研究[D].南京大学,2017.
- [6] 肖旻,陈行.基于 Python 语言编程特点及应用之探讨[J].电脑知识与技术,2014,10(34):8177-8178.
- [7] 郑纯军,贾宁.基于 Qt Quick 的跨平台移动应用开发关键技术研究[J].软件工程师,2015,18(04):33-35.
- [8] Xiaohua Zhang. RESEARCH ON CROSS PLATFORM DEVELOPMENT MODEL BASED ON QT QUICK [A]. 信息化与工程国际学会.Proceedings of 2016 6th International Conference on Information Engineering for Mechanics and Materials(ICIMM 2016)[C].信息化与工程国际学会:,2016:6.
- [9] 吴沧舟,兰逸正,张辉.基于 MySQL 数据库的优化[J].电子科技,2013,26(09):182-184.
- [10] 赵光亮,舒小松.Navicat for MySQL 平台中的 SQL 语言分析与应用[J].无线互联科技,2017(19):74-75.
- [11] Delel Rhouma, Lotfi Ben Romdhane. A new centrality measure for identifying influential nodes in social networks[P]. International Conference on Machine Vision, 2018.
- [12] 张海斌,周翔,张松,梁静.基于 Wireshark 和 Lua 的数据链协议解析插件开发[J].中国新通信,2017,19(15):58-59.
- [13] 白洁.用 Wireshark 抓包分析帧格式[J].电脑知识与技术,2011,7(28):6831-6832+6835.
- [14] 曹利,王丹丹.基于 Wireshark 的协议解析器开发研究和实现[J].计算机时代,2015(06):13-16.
- [15] 罗青林,徐克付,臧文羽,刘金刚.Wireshark 环境下的网络协议解析与验证方法[J].计算机工程与设计,2011,32(03):770-773.
- [16] Q. Bai, B. Jin, D. Wang, Y. Wang, X. Liu. Compact Modbus TCP/IP protocol for data acquisition systems based on limited hardware resources[J]. Journal of Instrumentation, 2018, 13(04).
- [17] Chenxi Wang, Mingzhe Liu, Aidong Xu, Jilong Zhang, Ni Jin. Design and Implementation of Embedded Network Master and Slave Communication Based on Modbus/TCP[P]. 2017 International Conference on Information Technology and Intelligent Manufacturing (ITIM 2017), 2017.
- [18] Yingjuan Zhao, Jingnan Ma, Shaojuan Li, Jia Li. Design of Modbus Wireless Communication System Based on Remote Data Transmission[P]. 2016 International Forum on Mechanical, Control and Automation (IFMCA 2016), 2017.
- [19] 左卫,程永新.Modbus 协议原理及安全性分析[J].通信技术,2013,46(12):66-69.
- [20] 冯源.QML 语言在显控界面开发中的应用[J].电脑编程技巧与维护,2018(02):62-64.
- [21] B. Douglas Blansit. The Basics of Relational Databases Using MySQL[J]. Journal of

Electronic Resources in Medical Libraries,2006,3(3).

[22]郑岚.Python 访问 MySQL 数据库[J].电脑编程技巧与维护,2010(06):59-61.

[23]张秀云.解决 MySQL 同步延迟[J].网络安全和信息化,2018(04):73-76.