**上海大学**

**光伏发电优化设计研究与平台开发**

**软件组结题报告**

****

指导老师：朱文浩，汪飞

报告人：宋建鑫，杨永华，方朝增，王舒，周王翼腾

[2016年1月3日]

目录

[**1** 项目背景 3](#_Toc439618896)

[**2** 项目实施方案 3](#_Toc439618897)

[**2.1** 项目综述 3](#_Toc439618898)

[2.1.1 NASA气象数据抓取、入库以及利用 3](#_Toc439618899)

[2.1.2 上位机平台开发 3](#_Toc439618900)

[**2.2** 项目准备 3](#_Toc439618901)

[2.2.1 Git应用 3](#_Toc439618902)

[2.2.2 Gitcafe应用 5](#_Toc439618903)

[2.2.3 开发环境 6](#_Toc439618904)

[2.2.4 数据库服务 6](#_Toc439618905)

[**2.3** 气象数据获取 7](#_Toc439618906)

[2.3.1 需求分析 7](#_Toc439618907)

[2.3.2 爬虫技术 7](#_Toc439618908)

[2.3.3 数据解析 11](#_Toc439618909)

[2.3.4 底层数据库设计 11](#_Toc439618910)

[2.3.5 数据入库 12](#_Toc439618911)

[2.3.6 数据应用 14](#_Toc439618912)

[**2.4** 上位机平台控制 17](#_Toc439618913)

[2.4.1 需求分析 17](#_Toc439618914)

[2.4.2 串口通信 17](#_Toc439618915)

[2.4.3 平台控制 18](#_Toc439618916)

[2.4.4 数据获取 19](#_Toc439618917)

[2.4.5 数据处理 20](#_Toc439618918)

[2.4.6 数据导出 20](#_Toc439618919)

[**3** 项目重点与难点 21](#_Toc439618920)

[**3.1** 团队编程的合作与分工 21](#_Toc439618921)

[**3.2** 应用爬虫的问题与解决 21](#_Toc439618922)

[**3.3** 数据仓库的设计与维护 21](#_Toc439618923)

[**3.4** 串口通信的异步与互斥 24](#_Toc439618924)

[**4** 总结 25](#_Toc439618925)

# 项目背景

太阳光电系统，也称为光生伏特，简称光伏，是指利用光伏半导体材料的光生伏打效应而将太阳能转化为直流电能的设施。光伏设施的核心是太阳能电池板。由于近年来各国都在积极推动可再生能源的应用，光伏产业的发展十分迅速。

截至2010年，太阳能光伏在全世界上百个国家投入使用。虽然其发电容量仍只占人类用电总量的很小一部分，不过，从2004年开始，接入电网的光伏发电量以年均60%的速度增长。到2009年，总发电容量已经达到21GW，是当前发展速度最快的能源。据估计，没有联入电网的光伏系统，目前的容量也约有3至4GW。

光伏系统可以大规模安装在地表上成为光伏电站，也可以置于建筑物的房顶或外墙上，形成光伏建筑一体化。自太阳能电池问世以来，使用材料、技术上的不断进步，以及制造产业的发展成熟，都驱使光伏系统的价格变得更加便宜。不仅如此，许多国家投入大量研发经费推进光伏的转换效率，给与制造企业财政补贴。更重要的，上网电价补贴政策以及可再生能源比例标准等政策极大地促进了光伏在各国的广泛应用。

而作为联合大作业项目——光伏电站优化设计研究与平台开发团队一支重要组成，由计算机工程与科学学院学生组成的光伏蓝队，需调研国内外光伏电站设计平台的软件水平现状，建立一套光伏电站优化设计平台，集成光伏电站优化设计方法和基于气象环境数据的优化分析结果，在此过程中也承担着软件开发、数据挖掘相关的任务，为机电工程与自动化学院的同学们的研究提供保障。

# 项目实施方案

## 项目综述

为了给光伏电站优化设计研究与平台开发研究所需的数据支持和软件支持，蓝队完成了如下工作：

### NASA气象数据抓取、入库以及利用

使用爬虫技术获取NASA网站所提供的全球气象数据，并将其存贮在我们自己的数据库中，在利用后期开发的工具予以提取、导出、利用等。

### 上位机平台开发

利用串口通信技术，在硬件平台的基础上，通过特定的协议与控制硬件并获取其数据，具体为闭环控制电机运转，选择光伏组件接入，获取IV曲线仪测得的数据，获取气象数据等，并将数据入库，再导出为Excel表格予以分析。

## 项目准备

### Git应用

1. 定义

Git是一款免费、开源的分布式版本控制系统，用于敏捷高效地处理任何或小或大的项目，简单点说它就是一个我们用来合作管理代码的工具。

1. 功能
2. 从服务器上克隆完整的Git仓库（包括代码和版本信息）到单机上。
3. 在自己的机器上根据不同的开发目的，创建分支，修改代码。
4. 在单机上自己创建的分支上提交代码。
5. 在单机上合并分支。
6. 把服务器上最新版的代码fetch下来，然后跟自己的主分支合并。
7. 生成补丁（patch），把补丁发送给主开发者。
8. 看主开发者的反馈，如果主开发者发现两个一般开发者之间有冲突（他们之间可以合作解决的冲突），就会要求他们先解决冲突，然后再由其中一个人提交。如果主开发者可以自己解决，或者没有冲突，就通过。
9. 优点

适合分布式开发，强调个体；公共服务器压力和数据量都不会太大；速度快、灵活任意两个开发者之间可以很容易的解决冲突；离线工作。

1. 示例



图 2‑1

图 2‑1是一个Git的代码提交记录功能，图中我们可以看到这个记录从左到右由四部分组成，分别是提交分之、提交说明、提交者以及提交时间，我们可以从这个记录中很容易地找到某一个人的提交记录，这样可以在之后的问题分析处理上面有很高的效率。

### Gitcafe应用

1. 定义

GitCafe是一个基于代码托管服务打造的技术协作与分享平台，程序开发爱好者们可以通过使用代码版本控制系统Git来将他们所写的开源或商业项目的代码托管在GitCafe上，与其他程序员针对这些项目在线协作开发。



图 2‑2

1. 说明

我们团队主要利用Git将我们的代码托管到GitCafe这个在线平台，同时我们也可以在gitcafe上面派发任务，通过创建工单的形式，将我们的任务具体到每一个人的身上，这样做不仅提高了效率，也是的团队管理运营更加规范。（图 2‑2是GitCafe上面的我们团队的平台，图 2‑3是我们在一次例会之后创建的工单，将任务具体分配到每一个人身上）



图 2‑3

### 开发环境

由于本次项目采用的编程语言是C#所以我们结合实际编程需求，采用的是微软的Visual Studio 2013在Windows 7操作系统下来进行项目的开发，Visual Studio 2013是美国微软公司的开发工具包系列产品。VS是一个基本完整的开发工具集，它包括了整个软件生命周期中所需要的大部分工具，如UML工具、代码管控工具、集成开发环境(IDE)等等。所写的目标代码适用于微软支持的所有平台，包括Microsoft Windows、Windows Mobile、Windows CE、.NET Framework、.NET Compact Framework和Microsoft Silverlight 及Windows Phone。Visual Studio是目前最流行的Windows平台应用程序的集成开发环境。

### 数据库服务

根据项目前期的数据获取量等情况，我们小组这次用到的数据库是微软的SQL Server 2008和Access数据库。

SQL Server 2008 在Microsoft的数据平台上发布，可以组织管理任何数据。可以将结构化、半结构化和非结构化文档的数据直接存储到数据库中。可以对数据进行查询、搜索、同步、报告和分析之类的操作。数据可以存储在各种设备上，从数据中心最大的服务器一直到桌面计算机和移动设备，它都可以控制数据而不用管数据存储在哪里。SQL Server 2008 允许使用 Microsoft .NET 和Visual Studio开发的自定义应用程序中使用数据，在面向服务的架构（SOA）和通过 Microsoft BizTalk Server 进行的业务流程中使用数据。信息工作人员可以通过日常使用的工具直接访问数据。

Microsoft Office Access是由微软发布的关系数据库管理系统。它结合了 MicrosoftJet Database Engine 和 图形用户界面两项特点，是 Microsoft Office 的系统程序之一。

其中SQL Server 2008是用来为点击地图取点获取当地气象数据服务的，它主要用来作为服务器端的数据库，作为地图取点之后气象数据的及时获取（如图 2‑4）。而Access数据库主要是用来将数据解析器解析的数据存储起来，为以后的研究做准备，同时也可以作为更新数据来更新地图上面的气象数据。



图 2‑4

## 气象数据获取

### 需求分析

为了对光伏电站发电面板的角度、倾角的算法进行优化，需大量的天气数据，为了获取这些数据，找到了NASA的天气数据源，对数据的获取需经过一下几个步骤：

1. 使用网络爬虫软件获取NASA天气数据源的网页文件；
2. 使用网络页面解析软件将获取到的网页文件进行解析，提取出需要的天气数据；
3. 建立数据库，建立格式表格，将数据按格式整理后存入数据库。

### 爬虫技术

网络爬虫（又被称为网页蜘蛛，网络机器人，在FOAF社区中间，更经常的称为网页追逐者），是一种按照一定的规则，自动地抓取万维网信息的程序或者脚本。

随着网络的迅速发展，万维网成为大量信息的载体，如何有效地提取并利用这些信息成为一个巨大的挑战。搜索引擎(Search Engine)，例如传统的通用搜索引擎AltaVista，Yahoo!和Google等，作为一个辅助人们检索信息的工具成为用户访问万维网的入口和指南。但是，这些通用性搜索引擎也存在着一定的局限性，如：

1. 不同领域、不同背景的用户往往具有不同的检索目的和需求，通用搜索引擎所返回的结果包含大量用户不关心的网页。
2. 通用搜索引擎的目标是尽可能大的网络覆盖率，有限的搜索引擎服务器资源与无限的网络数据资源之间的矛盾将进一步加深。
3. 万维网数据形式的丰富和网络技术的不断发展，图片、数据库、音频、视频多媒体等不同数据大量出现，通用搜索引擎往往对这些信息含量密集且具有一定结构的数据无能为力，不能很好地发现和获取。
4. 通用搜索引擎大多提供基于关键字的检索，难以支持根据语义信息提出的查询。

为了解决上述问题，定向抓取相关网页资源的聚焦爬虫应运而生。聚焦爬虫是一个自动下载网页的程序，它根据既定的抓取目标，有选择的访问万维网上的网页与相关的链接，获取所需要的信息。与通用爬虫(general?purpose web crawler)不同，聚焦爬虫并不追求大的覆盖，而将目标定为抓取与某一特定主题内容相关的网页，为面向主题的用户查询准备数据资源。

#### 聚焦爬虫工作原理以及关键技术概述

网络爬虫是一个自动提取网页的程序，它为搜索引擎从万维网上下载网页，是搜索引擎的重要组成。传统爬虫从一个或若干初始网页的URL开始，获得初始网页上的URL，在抓取网页的过程中，不断从当前页面上抽取新的URL放入队列,直到满足系统的一定停止条件。聚焦爬虫的工作流程较为复杂，需要根据一定的网页分析算法过滤与主题无关的链接，保留有用的链接并将其放入等待抓取的URL队列。然后，它将根据一定的搜索策略从队列中选择下一步要抓取的网页URL，并重复上述过程，直到达到系统的某一条件时停止。另外，所有被爬虫抓取的网页将会被系统存贮，进行一定的分析、过滤，并建立索引，以便之后的查询和检索；对于聚焦爬虫来说，这一过程所得到的分析结果还可能对以后的抓取过程给出反馈和指导。

相对于通用网络爬虫，聚焦爬虫还需要解决三个主要问题：

1. 对抓取目标的描述或定义；
2. 对网页或数据的分析与过滤；
3. 对URL的搜索策略。

抓取目标的描述和定义是决定网页分析算法与URL搜索策略如何制订的基础。而网页分析算法和候选URL排序算法是决定搜索引擎所提供的服务形式和爬虫网页抓取行为的关键所在。这两个部分的算法又是紧密相关的。

#### 抓取目标描述

现有聚焦爬虫对抓取目标的描述可分为基于目标网页特征、基于目标数据模式和基于领域概念3种。

基于目标网页特征的爬虫所抓取、存储并索引的对象一般为网站或网页。根据种子样本获取方式可分为：

1. 预先给定的初始抓取种子样本；
2. 预先给定的网页分类目录和与分类目录对应的种子样本，如Yahoo!分类结构等；
3. 通过用户行为确定的抓取目标样例，分为：
4. 用户浏览过程中显示标注的抓取样本；
5. 通过用户日志挖掘得到访问模式及相关样本。

其中，网页特征可以是网页的内容特征，也可以是网页的链接结构特征，等等。

基于目标数据模式的爬虫针对的是网页上的数据，所抓取的数据一般要符合一定的模式，或者可以转化或映射为目标数据模式。

另一种描述方式是建立目标领域的本体或词典，用于从语义角度分析不同特征在某一主题中的重要程度。

网页的抓取策略可以分为深度优先、广度优先和最佳优先三种。深度优先在很多情况下会导致爬虫的陷入(trapped)问题，目前常见的是广度优先和最佳优先方法。

广度优先搜索策略是指在抓取过程中，在完成当前层次的搜索后，才进行下一层次的搜索。该算法的设计和实现相对简单。在目前为覆盖尽可能多的网页，一般使用广度优先搜索方法。也有很多研究将广度优先搜索策略应用于聚焦爬虫中。其基本思想是认为与初始URL在一定链接距离内的网页具有主题相关性的概率很大。另外一种方法是将广度优先搜索与网页过滤技术结合使用，先用广度优先策略抓取网页，再将其中无关的网页过滤掉。这些方法的缺点在于，随着抓取网页的增多，大量的无关网页将被下载并过滤，算法的效率将变低。

最佳优先搜索策略按照一定的网页分析算法，预测候选URL与目标网页的相似度，或与主题的相关性，并选取评价最好的一个或几个URL进行抓取。它只访问经过网页分析算法预测为“有用”的网页。存在的一个问题是，在爬虫抓取路径上的很多相关网页可能被忽略，因为最佳优先策略是一种局部最优搜索算法。因此需要将最佳优先结合具体的应用进行改进，以跳出局部最优点。将在第4节中结合网页分析算法作具体的讨论。研究表明，这样的闭环调整可以将无关网页数量降低30%~90%。

深度优先搜索策略从起始网页开始，选择一个URL进入，分析这个网页中的URL，选择一个再进入。如此一个链接一个链接地抓取下去，直到处理完一条路线之后再处理下一条路线。深度优先策略设计较为简单。然而门户网站提供的链接往往最具价值，PageRank也很高，但每深入一层，网页价值和PageRank都会相应地有所下降。这暗示了重要网页通常距离种子较近，而过度深入抓取到的网页却价值很低。同时，这种策略抓取深度直接影响着抓取命中率以及抓取效率，对抓取深度是该种策略的关键。相对于其他两种策略而言。此种策略很少被使用。

网页分析算法可以归纳为基于网络拓扑、基于网页内容和基于用户访问行为三种类型。

拓扑分析算法：基于网页之间的链接，通过已知的网页或数据，来对与其有直接或间接链接关系的对象（可以是网页或网站等）作出评价的算法。又分为网页粒度、网站粒度和网页块粒度这三种。

1. 网页(Webpage)粒度的分析算法

PageRank和HITS算法是最常见的链接分析算法，两者都是通过对网页间链接度的递归和规范化计算，得到每个网页的重要度评价。PageRank算法虽然考虑了用户访问行为的随机性和Sink网页的存在，但忽略了绝大多数用户访问时带有目的性，即网页和链接与查询主题的相关性。针对这个问题，HITS算法提出了两个关键的概念：权威型网页（authority）和中心型网页（hub）。

基于链接的抓取的问题是相关页面主题团之间的隧道现象，即很多在抓取路径上偏离主题的网页也指向目标网页，局部评价策略中断了在当前路径上的抓取行为。文献[21]提出了一种基于反向链接（BackLink）的分层式上下文模型（Context Model），用于描述指向目标网页一定物理跳数半径内的网页拓扑图的中心Layer0为目标网页，将网页依据指向目标网页的物理跳数进行层次划分，从外层网页指向内层网页的链接称为反向链接。

1. 网站粒度的分析算法

网站粒度的资源发现和管理策略也比网页粒度的更简单有效。网站粒度的爬虫抓取的关键之处在于站点的划分和站点等级(SiteRank)的计算。SiteRank的计算方法与PageRank类似，但是需要对网站之间的链接作一定程度抽象，并在一定的模型下计算链接的权重。

网站划分情况分为按域名划分和按IP地址划分两种。文献[18]讨论了在分布式情况下，通过对同一个域名下不同主机、服务器的IP地址进行站点划分，构造站点图，利用类似PageRank的方法评价SiteRank。同时，根据不同文件在各个站点上的分布情况，构造文档图，结合SiteRank分布式计算得到DocRank。文献[18]证明，利用分布式的SiteRank计算，不仅大大降低了单机站点的算法代价，而且克服了单独站点对整个网络覆盖率有限的缺点。附带的一个优点是，常见PageRank 造假难以对SiteRank进行欺骗。

1. 网页块粒度的分析算法

在一个页面中，往往含有多个指向其他页面的链接，这些链接中只有一部分是指向主题相关网页的，或根据网页的链接锚文本表明其具有较高重要性。但是，在PageRank和HITS算法中，没有对这些链接作区分，因此常常给网页分析带来广告等噪声链接的干扰。在网页块级别(Block?level)进行链接分析的算法的基本思想是通过VIPS网页分割算法将网页分为不同的网页块(page block)，然后对这些网页块建立page?to?block和block?to?page的链接矩阵，?分别记为Z和X。于是，在page?to?page图上的网页块级别的PageRank为?W?p=X×Z；?在block?to?block图上的BlockRank为?W?b=Z×X。已经有人实现了块级别的PageRank和HITS算法，并通过实验证明，效率和准确率都比传统的对应算法要好。

网页分析算法：基于网页内容的分析算法指的是利用网页内容（文本、数据等资源）特征进行的网页评价。网页的内容从原来的以超文本为主，发展到后来动态页面（或称为Hidden Web）数据为主，后者的数据量约为直接可见页面数据（PIW，Publicly Indexable Web）的400~500倍。另一方面，多媒体数据、Web Service等各种网络资源形式也日益丰富。因此，基于网页内容的分析算法也从原来的较为单纯的文本检索方法，发展为涵盖网页数据抽取、机器学习、数据挖掘、语义理解等多种方法的综合应用。本节根据网页数据形式的不同，将基于网页内容的分析算法，归纳以下三类：第一种针对以文本和超链接为主的无结构或结构很简单的网页；第二种针对从结构化的数据源（如RDBMS）动态生成的页面，其数据不能直接批量访问；第三种针对的数据界于第一和第二类数据之间，具有较好的结构，显示遵循一定模式或风格，且可以直接访问。

基于文本的网页分析算法

* 1. 纯文本分类与聚类算法

很大程度上借用了文本检索的技术。文本分析算法可以快速有效的对网页进行分类和聚类，但是由于忽略了网页间和网页内部的结构信息，很少单独使用。

* 1. 超文本分类和聚类算法

根据网页链接网页的相关类型对网页进行分类，依靠相关联的网页推测该网页的类型。

### 数据解析

网页文件的解析所使用的工具是正则表达式。

正则表达式，又称正规表示法、常规表示法（英语：Regular Expression，在代码中常简写为regex、regexp或RE），计算机科学的一个概念。正则表达式使用单个字符串来描述、匹配一系列符合某个句法规则的字符串。在很多文本编辑器里，正则表达式通常被用来检索、替换那些符合某个模式的文本。

正则表达式是对字符串操作的一种逻辑公式，就是用事先定义好的一些特定字符、及这些特定字符的组合，组成一个“规则字符串”，这个“规则字符串”用来表达对字符串的一种过滤逻辑。

给定一个正则表达式和另一个字符串，我们可以达到如下的目的：

1. 给定的字符串是否符合正则表达式的过滤逻辑（称作“匹配”）；
2. 可以通过正则表达式，从字符串中获取我们想要的特定部分。

正则表达式的特点是：

1. 灵活性、逻辑性和功能性非常的强；
2. 可以迅速地用极简单的方式达到字符串的复杂控制。
3. 对于刚接触的人来说，比较晦涩难懂。

由于正则表达式主要应用对象是文本，因此它在各种文本编辑器场合都有应用，小到著名编辑器EditPlus，大到Microsoft Word、Visual Studio等大型编辑器，都可以使用正则表达式来处理文本内容。

### 底层数据库设计

数据库设计包含3张表：

1） 区块名与区块编号，如图 2‑5

2） 单元注解表，如图 2‑6

3） 单元数据表，如图 2‑7

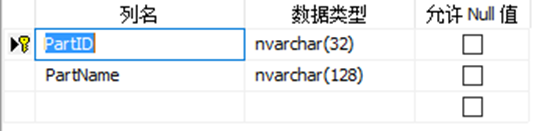


图 2‑5



图 2‑6

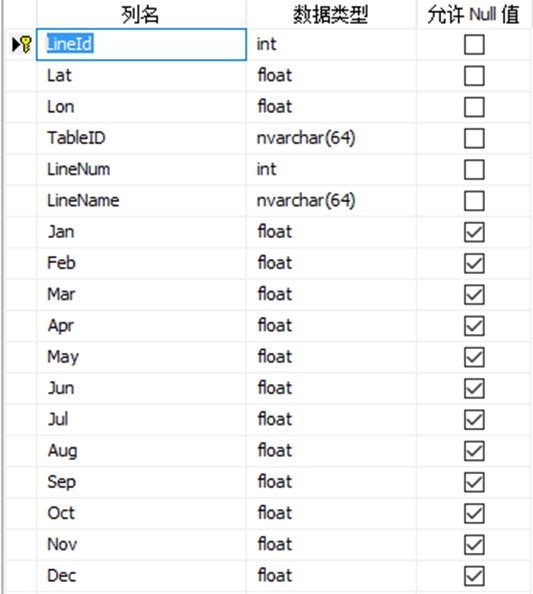


图 2‑7

### 数据入库

在C#System.Data类的基础上设计了一个数据操作核心类SHUPV.Database.Core类，这个类包含了连接数据库、断开数据库、插入数据、更新数据、查找数据、删除数据这五个数据库相关的方法。部分代码如下：

|  |
| --- |
| /// <summary>  /// 查询函数  /// </summary>  /// <param name="tableName">表名字</param>  /// <param name="queryTerms">查询条件where键值对</param>  /// <returns>查询返回的数据</returns>  public DataTable SelectData(string tableName, Dictionary<string,string> queryTerms)  {  string queryString = "select \* from " + tableName;  if (queryTerms != null && queryTerms.Count != 0)  {  queryString += " where";  bool isFirst = true;  foreach(KeyValuePair<string,string> kvp in queryTerms)  {  if(!isFirst)  queryString += " and";  isFirst =false;  queryString += " [" + kvp.Key + "] = \'" + kvp.Value + "\'";  }  }  SqlDataAdapter sda = new SqlDataAdapter(queryString, \_sqlCon);  DataTable result = new DataTable(tableName);  sda.Fill(result);  return result;//返回查询数据  } |

然后使用SHUPV.Database.Core类将所有数据提交至数据库，共一千六百万条数据，图 2‑8为截取的一部分数据。

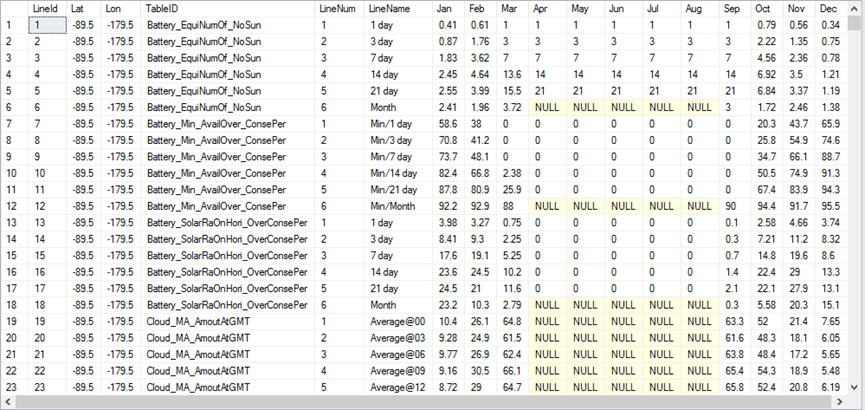


图 2‑8

### 数据应用

开发了一款数据应用：在地图上选点，显示用户想要的气象数据。

1. 打开软件，如图 2‑9：



图 2‑9

1. 在地图上选择上海大学本校区，自动获得其经纬坐标，如图 2‑10：



图 2‑10

1. 点击查询，连接远程数据库，如图 2‑11：

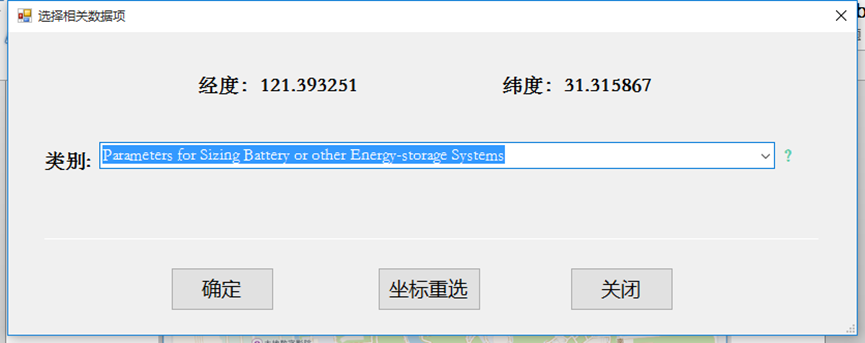


图 2‑11

1. 选择要查询的数据类别，如图 2‑12：

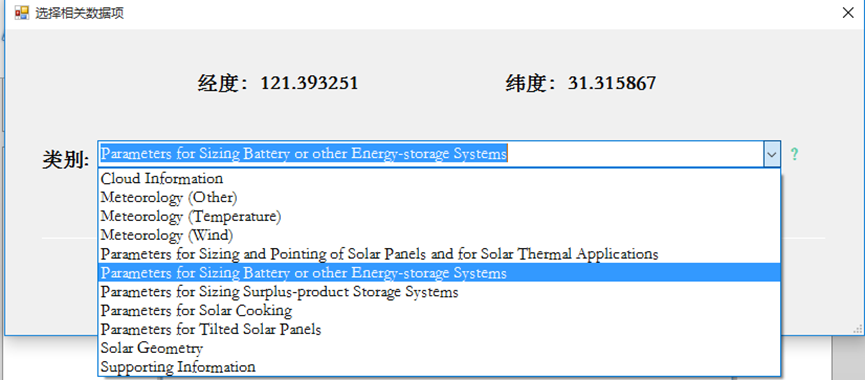


图 2‑12

1. 选择该类别下的数据项，如图 2‑13：

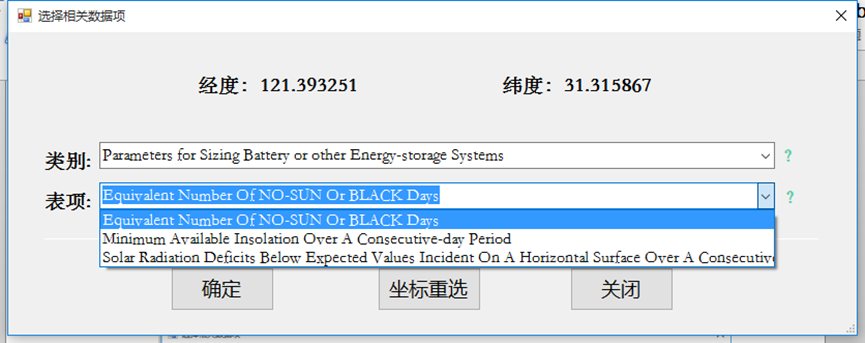


图 2‑13

1. 确定，获得该坐标点、该项的所有数据，如图 2‑14：



图 2‑14

## 上位机平台控制

### 需求分析

开发一个上位机软件平台与硬件进行通信，具有设备控制，数据采集，数据存储的功能。

1. 设备控制：要求可以通过软件平台向硬件发出指令，如设备开始运转，停止，旋转一定角度，切换组件，采集数据等。
2. 数据采集：要求软件平台能时刻接收硬件发送回来的数据，并进行解析。
3. 数据存储：软件平台将采集到的数据存储到数据库中，以备将来进行数据分析时使用。

### 串口通信

串口是计算机上一种非常通用的设备通信协议。大多数计算机（不包括笔记本电脑）包含两个基于RS-232的串口。串口同时也是仪器仪表设备通用的通信协议。

串口通信是指串口按位(bit)发送和接收字节，即将数据字节分成一位一位的形式在一条传输线上逐个地进行传输。特点是传输线少，长距离传送时的成本低，但数据的传输控制比并行通信复杂。

串口通信是异步的，端口能够在一根线上发送数据同时在另一根线上接收数据。串口通信最重要的参数是波特率、数据位、停止位和奇偶校验。对于两个进行通信的端口，这些参数必须匹配。

项目中利用.net提供的串口通信的功能来使软件平台与硬件进行通信。

### 平台控制

项目中利用自己开发的软件平台来进行平台控制。



图 2‑15

图 2‑5所示即为软件平台的主界面，包括参数设置、信息、运行时间设置、运行状态实时显示、指令设置、功能设置和数据查询的功能。

其中参数设置功能主要是设置串口通信时的串口号、波特率、数据位、停止位和检验位，由于项目中与硬件通讯的波特率是固定的，所以波特率在软件内部设定，不能手动设置。

信息面板用来显示系统当前时间和平台运行的条件未满足情况。

运行时间设置是用来设置平台运行的起止时间。

运行状态面板显示当前测量的组件号、角度仪返回的组件角度信息以及是否在解析数据和解析过程中是否发生异常。

指令设置界面用来打开指令文件和设置发送周期。

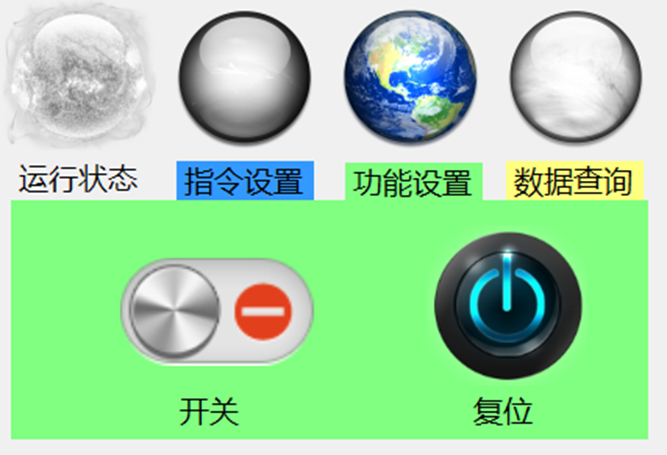


图 2‑16

图 2‑6是功能设置界面，开关是用来打开或关闭数据采集功能，复位按钮是将所有的组件恢复到初始状态。



图 2‑17

图 2‑7是数据查询界面，用来查询某一天所采集到的数据，以Excel表格形式显示。

### 数据获取

数据获取是获取各仪器通过485总线发回的数据。在C#中利用串口类的数据接收事件方法，当串口有数据返回时，程序调用相应的数据处理函数来处理接收到的数据。数据处理函数从接收到的数据中判断是哪个仪器发回的数据，再调相应的函数来处理特定的数据。

### 数据处理

数据处理部分包括将接受到的十六进制数据解析成我们需要的十进制数据和将解析好的数据存进数据库中。

从IV曲线仪返回的数据中需要解析出温度、开路电压、短路电流、最大功率点电压、最大功率点电流、最大功率等数据，从气象仪返回的数据中需要解析出太阳辐射、环境温度、湿度、风速、风向、组件温度等数据，最后将解析到的数据分别存储到数据库中的两张表中。



图 2‑18



图 2‑19

图 2‑8和图 2‑9所示为IV曲线仪和气象仪对应的数据库中的表的部分字段。

### 数据导出

数据导出是将数据库中的数据导出到Excel表格中以便查看，对应着软件平台中的数据查询功能。

在项目中利用C#提供的Excel程序接口来创建表格，将数据库中的数据按照一定的格式导出到创建好的表格中，当用户软件平台中点击数据查询按钮时，执行这一过程，打开Excel表格供用户查看数据。

# 项目重点与难点

## 团队编程的合作与分工

团队的分工主要通过开会讨论之后在Git上面进行分工，其中充分考虑组中每个成员的编程情况进行分工，以达到每个组员每一次都可以有充分的体验，当然，不可能每次的分组都会达到满意的效果，但是下一次的任务分配会避免出现之前出现的错误，比如有次任务分配不均，导致出现有的组员没有分配到任务的情况，那么在随后的任务分配中侧重分配给这位同学任务，这从侧面展示出了组员们在团队间合作相互协同彼此关注。

在涉及到主要任务上，根据组员之前知识储备，侧重让同学去完成自己熟练的任务以达到整个任务的高效性。其中组员周王翼腾因为熟悉正则表达式，所以在后期关于数据解析的匹配都让其用正则表达式进行完善；组员方朝增同学既对编程有着兴趣也对界面设计感兴趣，所以在让其完成自己编程任务的基础上还让其设计了数据解析器和地图气象取点的页面；组员王舒因为之前完成过从数据解析到存储数据库以及从数据库到服务器的数据传送，所以主要完成数据库的设计以及气象资料的获取；组员杨永华因为编程能力较强，除了完成自己本身的任务还协同完成数据库接口，数据解析以及其他重点程序设计；组长宋建兴作为整个团队的领导者，除了领导整个团队的走向，还兼顾者将团队资源进行整合展现出来，起着增强团队凝聚力的作用。

## 应用爬虫的问题与解决

在使用网络爬虫抓取网页的过程中，由于被抓取网站使用了反爬技术，当同一IP地址在一定时间内请求网页次数过多时便会对IP封禁，再数小时内禁止该IP请求网页。所以，为了防止IP被封禁的情况发生，就得降低网络爬虫对网站的请求频率，但是，由于所需抓取的网页的数量巨大，在降低爬虫请求频率的情况下，对网页的抓取进度缓慢，于是便需要采取更好的爬虫策略来提高抓取速度且避免被网站封禁IP的情况。

首先新加的一个策略是，让爬虫轮换使用多个代理IP，这样，在一段时间内的网页请求是由不同的IP地址发出，降低了单个IP地址对网站的请求频率，使得爬虫不会被网站识别为过多访问，避免单一IP地址过频访问而被封禁的请况。

在使用一段时间的多代理IP的策略后发现，代理IP并不稳定，一个代理IP在使用一段时间后便会出现很大的延迟而无法使用，于是开始使用分布式爬虫来解决问题。

分布式网络爬虫可以看做由多个集中式网络爬虫组合而成。分布式系统中的每个节点都可以看作一个集中式网络爬虫。多个网络爬虫共同运行，肯定会涉及到各个节点间相互通讯。在此采用的通讯方法是使用同一个数据库进行数据的通讯，在数据库中使用标记状态来分别每一个爬虫结点的任务。在此策略下，网络爬虫的抓取速度得到了保证，也不会出现被网站封禁IP 的情况。

## 数据仓库的设计与维护

在NASA数据网页上，有数据11类：

|  |
| --- |
| Parameters for Sizing Battery or other Energy-storage Systems |
| Cloud Information |
| Meteorology (Other) |
| Parameters for Sizing and Pointing of Solar Panels and for Solar Thermal Applications |
| Parameters for Solar Cooking |
| Solar Geometry |
| Supporting Information |
| Parameters for Sizing Surplus-product Storage Systems |
| Meteorology (Temperature) |
| Parameters for Tilted Solar Panels |
| Meteorology (Wind) |

而又细分为65个表项：

|  |
| --- |
| Equivalent Number Of NO-SUN Or BLACK Days |
| Minimum Available Insolation Over A Consecutive-day Period |
| Solar Radiation Deficits Below Expected Values Incident On A Horizontal Surface Over A Consecutive-day Period |
| Monthly Averaged Cloud Amount At Indicated GMT Times |
| Monthly Averaged Daylight Cloud Amount |
| Monthly Averaged Frequency Of Broken-cloud Skies At Indicated GMT Times |
| Monthly Averaged Frequency Of Clear Skies At Indicated GMT Times |
| Monthly Averaged Frequency Of Near-overcast Skies At Indicated GMT Times |
| Monthly Averaged Atmospheric Pressure |
| Monthly Averaged Precipitation |
| Monthly Averaged Relative Humidity |
| Monthly Averaged Specific Humidity At 10 m Above The Surface Of The Earth |
| Monthly Averaged Total Column Precipitable Water |
| Monthly Averaged Clear Sky Insolation Clearness Index |
| Monthly Averaged Clear Sky Insolation Normalized Clearness Index |
| Monthly Averaged Clear Sky Insolation Incident On A Horizontal Surface |
| Monthly Averaged Diffuse Radiation Incident On A Horizontal Surface |
| Monthly Averaged Direct Normal Radiation |
| Monthly Averaged Downward Longwave Radiative Flux |
| Monthly Averaged Insolation Clearness Index |
| Monthly Averaged Insolation Normalized Clearness Index |
| Monthly Averaged Insolation Incident On A Horizontal Surface |
| Monthly Averaged Insolation Incident On A Horizontal Surface At Indicated GMT Times |
| Minimum And Maximum Difference From Monthly Averaged Direct Normal Radiation |
| Minimum And Maximum Difference From Monthly Averaged Insolation |
| Monthly Averaged Clear Sky Days |
| Monthly Averaged Clear Sky Insolation Incident On A Horizontal Surface |
| Monthly Averaged Insolation Incident On A Horizontal Surface |
| Monthly Averaged Midday Insolation Incident On A Horizontal Surface |
| Monthly Averaged Cosine Solar Zenith Angle At Mid-Time Between Sunrise And Solar Noon |
| Monthly Averaged Daylight Average Of Hourly Cosine Solar Zenith Angles |
| Monthly Averaged Daylight Hours |
| Monthly Averaged Declination |
| Monthly Averaged Hourly Solar Azimuth Angles |
| Monthly Averaged Hourly Solar Angles Relative To The Horizon |
| Monthly Averaged Maximum Solar Angle Relative To The Horizon |
| Monthly Averaged Solar Noon |
| Monthly Averaged Sunset Hour Angle |
| Monthly Averaged Surface Albedo |
| Monthly Averaged Top-of-atmosphere Insolation |
| Available Surplus Insolation Over A Consecutive-day Period |
| Average Daily Temperature Range |
| Average Minimum, Maximum and Amplitude Of The Daily Mean Earth Temperature |
| Monthly Averaged Air Temperature At 10 m Above The Surface Of The Earth |
| Monthly Averaged Arctic Heating Degree Days Below 0 °C |
| Monthly Averaged Arctic Heating Degree Days Below 10 °C |
| Monthly Averaged Cooling Degree Days Above 18 °C |
| Dew/Frost Point Temperature At 10 m |
| Monthly Averaged Frost Days |
| Monthly Averaged Heating Degree Days Below 18 °C |
| Monthly Averaged Earth Skin Temperature |
| Monthly Averaged Radiation Incident On An Equator-Pointed Tilted Surface |
| Maximum Radiation Incident On An Equator-pointed Tilted Surface |
| Minimum Radiation Incident On An Equator-pointed Tilted Surface |
| Difference Between The Average Wind Speed At 10 m Above The Surface Of The Earth And The Average Wind speed At 50 m Above The Surface Of The Earth |
| Monthly Averaged Wind Direction At 50 m Above The Surface Of The Earth |
| Monthly Averaged Wind Direction At 50 m Above The Surface Of The Earth For Indicated GMT Times |
| Monthly Averaged Percent Of Time The Wind Speed At 50 m Above The Surface Of The Earth Is Within The Indicated Range |
| Monthly Averaged Wind Speed At 10 m Above The Surface Of The Earth For Terrain Similar To Airports |
| Monthly Averaged Wind Speed At 50 m Above The Surface Of The Earth |
| Monthly Averaged Wind Speed At 50 m Above The Surface Of The Earth For Indicated GMT Times |
| Monthly Averaged Wind Speed At 50, 100, 150 and 300 m Above The Surface Of The Earth |
| Monthly Averaged Wind Speed For Several Vegetation And Surface Types |
| Monthly Averaged Wind Speed Adjusted For Height And Vegetation Type |
| Minimum And Maximum Difference From Monthly Averaged Wind Speed At 50 m |

为了将所有的数据能统一在一个数据表中，数据库在设计时，采用了三级关系型数据表，并采用纵表的设计理念来设计数据表。

横表就是普通的建表方式，如一个表结构为：主键、字段1、字段2、字段3……。如果变成纵表后，则表结构为：主键、字段代码、字段值。而字段代码则为字段1、字段2、字段3。 具体为电信行业的例子。以用户帐单表为例一般出账时用户有很多费用客户，其数据一般存储为：时间，客户ID,费用科目，费用。这种存储结构一般称为纵表，其特点是行数多，字段少。纵表在使用时由于行数多，统计用户数或对用户进行分档时还需要进行GROUP BY 操作，性能低，且操作不便，为提高性能，通常根据需要将纵表进行汇总，形成横表，比如：时间、客户ID,基本通话费、漫游通话费，国内长途费、国际长途费……。通常形成一个客户一行的表，这种表统计用户数或做分档统计时比较方便。另外，数据挖掘时用到的宽表一般也要求是横表结构。纵表对从数据库到内存的映射效率是有影响的，但细一点说也要一分为二：纵表的初始映射要慢一些；纵表的变更的映射可能要快一些，如果只是改变了单个字段时，毕竟横表字段比纵表要多很多。

综上，应该把不容易改动表结构的设计成横表，把容易经常改动不确定的表结构设计成纵表。数据库在设计时，综合考虑了灵活性和高效性，将类别、表明、数据名这一类不宜固定的数据采用灵活的纵表设计，但每条数据都有12个字段对应一年中的12个月，这一规则对于任何数据都是有效的，故采用横表的设计。

最后数据库里产生了一千六百多万条的记录，每次查询都将花费很久的时间，为了使得数据查询更高效，符合实际的使用需求，必须加以索引。

数据库索引，是数据库管理系统中一个排序的数据结构，以协助快速查询、更新数据库表中数据。索引键值的逻辑顺序与索引所服务的表中相应行的物理顺序相同的索引，被称为聚集索引，反之为非聚集索引，索引一般使用二叉树排序索引键值的，聚集索引的索引值是直接指向数据表对应元组的，而非聚集索引的索引值仍会指向下一个索引数据块，并不直接指向元组，因为还有一层索引进行重定向，所以非聚集索引可以拥有不同的键值排序而拥有多个不同的索引。而聚集索引因为与表的元组物理顺序一一对应，所以只有一种排序，即一个数据表只有一个聚集索引。

为了符合数据查询的特点，我们给数据的三个字段添加了索引，即时间、经度、纬度、数据名，成功提高了数据的查询速度，使得查任意一条数据记录，耗时都在2秒以下，符合后续开发的需求。

## 串口通信的异步与互斥

当程序接收到返回的数据时，会将数据缓存在一个数据队列中，当有几个进程同时处理这个队列时会造成数据的不一致问题，造成数据解析异常。解决方法是对数据队列进行互斥。

互斥是指某一资源同时只允许一个访问者对其进行访问，具有唯一性和排它性。但互斥无法限制访问者对资源的访问顺序，即访问是无序的。程序中的数据解析过程是互斥的，对于接收到的数据，同一时间只能有一个数据解析函数进行解析。

我们在程序中给该数据队列增加一个锁，只有拿到锁的进程才能对数据队列进行操作，操作完毕后就释放锁。其它进程在对数据进行访问时若发现锁已被某个进程持有就进入等待状态，等待锁被释放后重新对数据发出访问申请。这就保证了每次只有一个进程对数据进行操作，解决了数据的不一致问题。

还有一个问题是程序在执行过程中会一直等待仪器返回的数据，当有数据返回时才进行解析，如果程序一直阻塞在这里，其他功能就无法得到执行，程序就会卡死。解决方法就是异步。

异步指一个进程在执行某个请求的时候，该进程不需要一直等下去，而是继续执行下面的操作，不管其他进程的状态。当有消息返回时系统会通知进程进行处理，这样可以提高执行的效率。例如程序中的数据获取过程就是异步的，程序在没有数据返回时会处理其他事情，不会一直等待直到数据返回才往下进行。

# 总结

第十二届全国人民代表大会中，李克强总理提出制定“互联网+”行动计划，此次联合大作业项目——光伏电站优化设计研究与平台开发，将计算机技术与其他领域的技术相互交融，是对“互联网+”技术的一次初探，也是对新形势的中国发展格局下，合作共赢的一次大胆尝试。

此次联合大作业项目，横跨机电工程与自动化学院与计算机工程与科学学院，在两方人马合作过程中，在技术方面相互弥补，在精神层次相互勉励，将合作意识融入到工作的每个环节，将合作精神在传递到任务的方方面面，不仅良好的完成了计划的任务，初步达到了预期的目标，也使得参与其中的每个人都在各个方面都得到了长足的进步。

作为由计算机工程与科学学院学生组成的蓝队，在整个联合大作业团队的合作过程中，一直承担着与软件开发、数据挖掘相关的任务，先后完成了NASA气象数据的获取与入库，上位机平台的开发与测试，为机电工程与自动化学院的同学们提供了强大的数据支持和软件支持，同时也积极寻找新的气象数据源，撰写商业软件计划书等，为项目的完备性贡献出了一份力量，获得了老师和同学们的赞许。

然而，在参与联合大作业的过程中，我们也暴露了很多缺点和不足之处，由于缺乏对硬件的了解，以及与其他人的沟通不到位，使得我们在开发软件的过程中一直遇到这样或那样的问题。而作为一群爱写代码的“极客”，我们在思维活跃、想法独特的另一面，也缺乏机自的同学的严谨和认真，耐心与细心，犯了一些粗心造成的低级错误，值得我们好好反省。然而有不足才会有进步的空间，我们很高兴能从其他学院的同学的身上学习到不同领域的知识和优秀的品质精神。

此次联合大作业项目，是我们认识到了很多，学习到了很多，认识到了很多，它的圆满结题不仅是对我们的一次褒奖，也是对我们的一次勉励，在今后的学习、工作道路上，联合大作业各位老师、各位同学，所表现出的合作共赢、开拓创新，将是我们可贵的精神财富，和坚实的精神力量。