



硕士专业学位论文



论文题目 智能型光伏电站的远程数据采集与
监控系统的设计与实现

研究生姓名 朱 锐

指导教师姓名 王 林

专业名称 软件工程

研究方向 嵌入式系统

论文提交日期 2013 年 10 月

智能型光伏电站的远程数据采集与监控系统的设计与实现

中文摘要

本文介绍了太阳能光伏发电的原理与光伏电站的组成,并对目前智能型光伏电站的远程数据采集与监控系统的设计与实现做了分析研究,对后端的软件系统做了实现与测试。

一个完整的光伏电站是由太阳能电池板(光伏组件)、逆变器、数据采集器(网关)、监控系统软件构成。太阳能电池板负责将光能转化为电能,逆变器负责将直流电逆变为交流电,数据采集器可以同时采集多台逆变器的数据,然后通过以太网将数据发送至监控系统。

监控系统软件是整个光伏系统中重要的组成部分,由数据接收端和监控 Web 界面组成。接收端完成接收来自全球各地网关的数据、数据解析和存储。监控 Web 将数据库中的数据以数值、图表、报表等形式进行展示。监控系统可以及时发现电站中发生的故障、预警等信息,并通过邮件、短信的方式告知用户或电站管理者。使太阳能电站真正成为无人职守的,少维护的,智能化的电站系统。

本软件采用微软的 Visual Studio 2012 和 .NET Framework 4.5 作为开发平台,数据库服务器采用 MySQL 数据库。监控 Web 系统采用 B/S 模式,即浏览器与服务器交互的结构,采用 MVC 三层架构,将系统的模型层、视图层与控制层分离,以达到解耦目的,便于系统的开发与维护。

系统主要包括实时监控、布局界面、报表界面、报警界面几个模块,用户可随时随地通过 Web 浏览器登录自己的电站了解自己电站的发电状况。

关键词: 智能光伏电站; 数据采集; 监控系统; B/S

作者: 朱 锐

指导老师: 王 林

The Design and Implementation of Remote Data Acquisition and Monitoring System of Intelligent Photovoltaic Power Station

Abstract

This thesis introduces the principle of solar photovoltaic power generation and the constitute of photovoltaic power station, and analysis the Design and implementation of remote data acquisition and monitoring system of intelligent photovoltaic power station, do the developing and testing of the software system.

A complete photovoltaic power station includes solar panels(PV module), inverter, data collector(gateway) and the monitoring system. Solar panels convert the light energy to electrical energy, inverter is for DC to AC, data collector can collect multiple inverter data at the same time, then send the data to the monitoring system via Ethernet.

The monitoring system is an important part of the whole photovoltaic system, including two parts of the data receiver and the web user interface. The data receiver receives the data from gateway all over the world, do the data analysing and storing. The web user interface display the data in the database in a form of chart, report and so on. Monitoring system can detect the equipment failure, early warning information of the power station, and through emails, messages to inform the user or the plant manager, which makes solar power plant a really intelligent power plant system with nobody on duty and less maintenance.

The software uses Microsoft Visual Studio 2012 and .NET Framework 4.5 as the development platform, the database server using MySQL. The monitoring system is designed by B/S mode—— browser and server interaction structure, using MVC three-tier model layer, the system's model layer, view layer and control layer separation, in order to achieve decoupling purposes, to facilitate the development and maintenance of the system.

The system mainly includes the real-time monitoring, the layout interface, report interface, alarm interface, the user can probably understand the status of their own plant whenever and wherever through logining their own plant.

Key Words: Intelligent photovoltaic power station; Data acquisition; Monitoring system; B/S

Written by Zhu Rui

Supervised by Wang Lin

目 录

第一章 绪论	1
1.1 课题背景	1
1.2 国内外光伏发电技术现状	1
1.3 远程数据采集与监控技术	2
1.4 本文的研究背景和目的	3
1.5 本文的主要内容	3
第二章 太阳能光伏发电技术	5
2.1 太阳能光伏发电原理	5
2.1.1 光电效应概述	5
2.1.2 太阳能电池片	5
2.1.3 太阳能电池组件	6
2.2 智能型光伏电站系统组成	7
第三章 系统设计	12
3.1 系统需求分析	12
3.2 系统设计目标和原则	12
3.3 软件功能设计	13
3.3.1 数据接收程序	13
3.3.2 监控 Web 界面	13
3.4 数据库设计	16
3.5 技术方案	19
3.5.1 上位机软件开发及生产环境	19
3.5.2 项目模型	19
3.5.3 监控通讯协议	21
第四章 系统主要功能与实现	28
4.1 接收端的功能与实现	28
4.2 Web 界面展示及功能	36

4.2.1 登录界面	36
4.2.2 注册界面	37
4.2.3 主界面	37
4.2.4 布局界面	38
4.2.5 报表界面	39
4.2.6 历史故障界面	40
第五章 系统的测试	41
5.1 系统测试计划	41
5.2 系统测试环境	41
5.3 系统的测试方法与工具	41
5.4 测试结果与总结	41
5.4.1 压力测试	41
5.4.2 页面加载时间测试	46
第六章 总结和展望	48
6.1 总结	48
6.2 展望	49
参考文献.....	50
攻读学位期间公开发表的论文.....	53
致 谢	54

第一章 绪论

1.1 课题背景

21 世纪，人类将面临着实现经济和社会可持续发展的重大挑战。在有限资源和保护环境的双重制约下，能源问题将更加突出，主要体现在能源短缺、环境污染和温室效应等方面。因此，人类在解决能源问题，实现可持续发展时，只能依靠科技进步，大规模地开发利用可再生洁净能源^[1]。

太阳能具有储量大、普遍存在、利用经济、清洁环保等优点，进入 21 世纪以来，全球面临严峻的能源形势和生态环境压力，在有限资源和保护环境的双重制约下能源问题将更加突出，为此各国政府相继制定支持太阳能等新能源产业发展的政策法规，光伏产业进入了快速发展期。太阳能具有安全可靠、无噪声、无污染、可再生、制约少、故障率低、维护简便等优点，广泛应用于工业、农业、军事、交通、通讯等各个领域。据统计，最近六年（2008-2013）世界太阳能光伏产业年平均增长率都在 45% 以上，随着太阳能光伏发电技术的不断成熟与普及，太阳能光伏发电正逐步由特殊应用转向民用、由辅助能源向基础能源过渡，而光伏并网发电系统是光伏系统的必然趋势。光伏发电系统一般由太阳能电池组件、逆变器以及控制器这几部分组成。为了实时监控光伏并网电站设备的运行状态，并保证每部分期间均能正常运行，就需要建立智能型光伏电站的远程数据采集与监控系统，对相应参数进行采集、保存、展示和分析，实现对光伏电站的有效监控，对光伏电站的安全可靠运行，具有十分重要的现实意义^[2]。

1.2 国内外光伏发电技术现状

西方国家的光伏发电产业比较发达，其中对美国、德国以及亚洲的日本的光伏发电产业加以主要介绍。

美国是最早实现太阳能转换为电能的国家。1930 年，朗格首次提出用“光伏效应”制造“太阳能电池”，使太阳能变成电能。1954 年，恰宾和皮尔松在美国贝尔实验室首次制成了实用的单晶太阳能电池。在 2008 年新增装机容量已经列全球第 2 位，2011

年 2 月 4 日, 美国能源部发起“Sunshot”计划, 拟在 2020 年前将太阳能光伏系统总成本降低 75%, 达到每千瓦时 6 美分, 随后美国能源部宣布了一系列有条件贷款担保太阳能光伏项目, 整个 2011 年美国太阳能光伏发电装机容量为 185.5 万千瓦, 为 2010 年的两倍以上。美国光伏产业的迅猛发展主要得益于与光伏发电相关的组件价格大幅下降以及联邦政府的大力补贴。

德国可以说是世界上对光伏产业支持力度最大的国家。其太阳能光伏电池的用量接近整个世界的一半, 其光伏产业近几年来发展迅猛。德国 2009 年的太阳能总装机容量为 4 吉瓦, 2010 年则迅速增长到 10 吉瓦, 2011 年更是创下世界纪录, 年装机容量达到 7.5 吉瓦。但由于其光伏产业发展的过于迅速, 导致了光伏产业产能过剩, 一系列老牌光伏产业申请破产, 包括 Solon、Gecko Group 和 systaic 等著名厂商, 其中 Solon 是德国光伏企业中最早上市的公司, 为德国的明星企业。

日本是最早推广光伏发电产业的国家之一, 2004 年以前日本一直雄踞光伏产业的霸主地位, 但自从 2005 年开始, 日本取消了“阳光屋顶计划”, 装机容量呈下降趋势, 2009 年后, 日本重新开始太阳能屋顶补贴计划, 光伏产业呈现发展新面貌, 但由于 2005-2008 年之间政府的消极政策, 导致日本光伏技术比之欧洲国家相对落后, 其太阳能电池价格偏贵^[3]。

中国光伏产业发展相对滞后, 国内光伏市场狭小, 2006 年以来中国光伏产品的出口比例一直在 95% 以上, 而国内太阳能总装机容量仅为全球总容量的 2%。中国政府今年来连续颁布了一系列政策支持光伏产业, 在“十二五”期间, 国家已明确了从两方面推动光伏发电应用: 一是通过国际通行的招标方式, 启动建设一批规模较大的并网光伏电站; 二是继续大力推进和实施“金太阳示范工程”, 通过多种政策补贴的组合带动应用项目发展^{[4-6][10]}。

1.3 远程数据采集与监控技术

目前, 光伏并网电站采用的监控技术比较落后。光伏并网电站普遍采用以单片机为主的控制单元、以 RS-485 总线为主的通信网络以及采用价格高且通用性较差的组态软件为上位机检测系统。因此, 在完成光伏并网电站建设的同时, 如何提高电站监控系统的性能及智能水平, 具有非常重要的研究价值和意义。

光伏智能监控系统在国内外的主要发展体现在光伏逆变器厂家自身开发的监控系统及光伏电站监控开发商开发的系统监控系统上，国内外逆变器厂家基本都提供逆变器监控系统界面供客户使用，光伏电站监控开发商可以提供从逆变器到汇流箱、高压设备的监控服务，但这两者都有一个共同的缺点：监控界面 UI 界面不够人性化，无法真正让客户体验人性化的服务；没有智能判断、预警、对电站进行自动管理等智能功能。

随着分布式光伏发电的发展和运用，未来光伏监控系统更多的会面向终端家庭用户，提供人性化、美观简洁的界面及强大稳定功能的监控系统成为发展趋势，同时，性能优异的智能监控系统将大大推动光伏发电的应用^[7-9]。

1.4 本文的研究背景和目的

本课题是针对国内外光伏并网发电站，结合光伏并网逆变器、数据采集网关等硬件的基础上，经过对各厂家逆变器通讯接口的研究，自主定义了一套可根据设备类型适用于多种逆变器及智能设备的、基于 TCP/IP 的通讯协议，以及应用此协议的数据采集和监控系统。传统的监控系统安装必须配套一台 PC 机，本系统中采用的是无须 PC 机的远程网络监控系统，通过逆变器配套的数据采集网关，只要插上网线连通 Internet，就可以将逆变器中采集到的数据定时的发送给远程的数据接收服务器，实现了基于 Internet 的远程多点的数据采集。同时，研究开发一套适用于多浏览器的 Web 在线监控平台，将所采集到的数据以数值、曲线图、报表等方式展现，并结合运行经验，研究建立一套知识库，为各种警告信息、缺陷提供参考处理方案，可通过邮件的方式及时的通知电站运营人员。

1.5 本文的主要内容

本文共分为六章，其主要结构如下。

第一章绪论。主要介绍了本课题的背景、国内外光伏发电技术现状、数据采集和监控系统概念及目前的技术水平和发展趋势。

第二章太阳能发电技术。主要介绍了太阳能电池、太阳能组件的发电原理和制造工艺。智能型光伏电站的系统组成及工作原理。

第三章系统设计。主要介绍了监控系统的架构,以及软件设计上用到的主要技术、项目模型、通讯协议。

第四章系统主要功能实现。主要介绍了接收端的处理流程和逻辑设计以及部分代码实现。网页端的用户界面设计和功能介绍。

第五章系统的测试。针对网站和数据库服务器的压力测试以及多浏览器兼容性和页面加载速度的测试。

第六章总结和展望。总结此系统的功能优点及不足,后续需改进之处。

第二章 太阳能光伏发电技术

2.1 太阳能光伏发电原理

2.1.1 光电效应概述

光伏发电是利用半导体界面的光生伏特效应而将光能直接转变为电能的一种技术。这种技术的关键元件是太阳能电池。太阳能电池经过串联后进行封装保护可形成大面积的太阳电池组件，再配合上功率控制器等部件就形成了光伏发电系统装置^[11]。

2.1.2 太阳能电池片

制作太阳能电池主要是以半导体材料为基础，其工作原理是利用光电材料吸收光能后发生光电子转换反应。太阳能电池片又分为：晶体硅电池(又分为单晶硅和多晶硅)、非晶硅薄膜电池、化学太阳能电池、多元化合物薄膜太阳能电池等。晶体硅太阳能电池主要基材为硅(又称为硅片)，它对硅原料的纯度要求特别高，纯度要达到99.99999%以上，电池片的生产过程必须是在无尘车间生产的^[12]。

太阳能电池的主要工作原理是：把3个价电子硼掺杂到纯净的硅晶格后，少出1个自由电子，所以要从硅原子价键中获取1个电子来填充，同时硅原子产生1个空穴，而变成带负电。所以把3价硼称为受主型杂质，也叫做P杂质。用于太阳能电池的半导体材料是一种介于导体和绝缘体之间的特殊物质，和任何物质的原子一样，半导体的原子也是由带正电的原子核和带负电的电子组成，半导体硅原子的外层有4个电子，按固定轨道围绕原子核转动。当受到外来能量的作用时，这些电子就会脱离轨道而成为自由电子，并在原来的位置上留下一个“空穴”，在纯净的硅晶体中，自由电子和空穴的数目是相等的。如果在硅晶体中掺入硼、镓等元素，由于这些元素能够俘获电子，它就成了空穴型半导体，通常用符号P表示；如果掺入能够释放电子的磷、砷等元素，它就成了电子型半导体，以符号N代表。若把这两种半导体结合，交界面便形成一个P-N结。太阳能电池的P-N结就像一堵墙，阻碍着电子和空穴的移动。当太阳能电池受到阳光照射时，电子接受光能，向N型区移动，使N型区带负电，同时空穴向P型区移动，使P型区带正电。这样，在P-N结两端便产生了电动势，也就是通常所说的电压。这种现象就是上面所说的“光生伏打效应”。能产生光伏效应

的材料有许多种，如：单晶硅，多晶硅，非晶硅，砷化镓，硒铟铜等。它们的发电原理基本相同，现以晶体硅为例描述光发电过程。当光线照射太阳电池表面时，一部分光子被硅材料吸收；光子的能量传递给了硅原子，使电子发生了越迁，成为自由电子在 P-N 结两侧集聚形成了电位差，当外部接通电路时，在该电压的作用下，将会有电流流过外部电路产生一定的输出功率。这个过程的实质是：光子能量转换成电能的过程^[13]。

其制造工艺如图 2.1 所示。

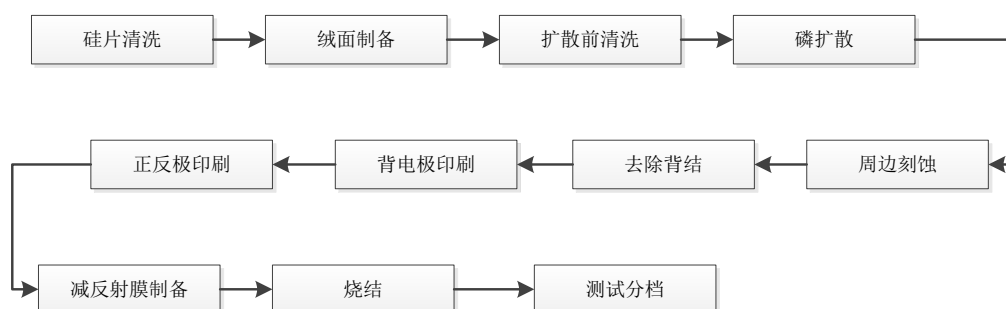


图 2.1 太阳能电池片制造工艺

2.1.3 太阳能电池组件

太阳能电池不能作为电源使用，必须将若干的太阳能电池经过串、并联连接，层压后封装为太阳能电池组件。太阳能电池组件由进口（或国产）单晶（或多晶）硅太阳能电池片串并联，用钢化玻璃、EVA 及 TPT 热压密封而成，周边加装铝合金边框，具有抗风、抗冰雹能力强、安装方便等特性。广泛应用于太阳能照明、灯具、户用供电、公路交通、建筑及光伏电站等领域。

太阳能电池板组件构成

1、钢化玻璃

低铁钢化玻璃（又称白玻璃），厚度 3.2 毫米，在太阳电池光谱响应的波长范围内（320-1100 纳米）透光率达 90% 以上，对于大于 1200 纳米的红外光有较高的反射率。此玻璃同时耐紫外光线的辐照，透光率不下降。钢化性能符合国标 GB9963-88 或者封装后的组件抗冲击性能达到国标 GB9535-88 地面用硅太阳能电池环境试验方法中规定的性能指标。

2、EVA

EVA 是一种热融胶粘剂，厚度在 0.4 毫米-0.6 毫米之间，表面平整，厚度均匀，

内含交联剂。常温下无黏性且具抗黏性，经过一定调价热压便发生熔融粘接与交联固化，并变的完全透明。固化后的 EVA 能承受大气变化且具有弹性，它将电池片“上盖下垫”，将其包封，并和上层保护材料-玻璃，下层保护材料背板（TPT，BBF 等），利用真空层压技术合为一体。另一方面，它和玻璃粘和后能提高玻璃的透光率，起着增透的作用，并对太阳能电池板的输出有增益作用。

3、太阳能电池片

太阳能电池片是光电转换的最小单元，尺寸一般为 125×125 或 156×156 英寸。太阳能电池片的工作电压约为 0.5 伏，一般不能单独作为电源使用。将太阳能电池片进行串并联封装后，就成为太阳能电池板，其功率一般为几瓦到几十瓦，一百瓦到两百瓦以上，可以单独作为电源使用。

4、背板

背板就是电池板背面的保护材料，一般有 TPT、BBF、DNP 等等。这些保护材料具有良好的抗环境侵蚀能力，绝缘能力并且可以和 EVA 良好粘接。太阳电池的背面覆盖物-氟塑料膜为白色，对阳光起反射作用，因此对电池板的效率略有提高，并因其具有较高的红外发射率，还可以降低电池板的工作温度，也有利于电池板的效率。当然，氟塑料膜首先具有太阳电池封装所要求的耐老化、耐腐蚀、不透气等基本要求。

5、接线盒

接线盒一般由 ABS 制成，并加有防老化和抗紫外辐射剂，能确保电池版纳在室外使用 25 年以上不出现老化破裂现象。接线柱由外镀镍层的高导电铜制成，可以确保电气导通及电气连接的可靠。接线盒用硅胶粘接在背板表面。

6、铝合金边框

边框采用硬制铝合金制成，表面氧化层厚度大于 10 微米，可以保证在室外环境长达 25 年以上的使用，不会被腐蚀，牢固耐用^[13]。

2.2 智能型光伏电站系统组成

太阳能光伏发电系统主要包括：太阳能电池组件(阵列)、控制器、蓄电池、逆变器、用户即照明负载等组成。其中，太阳能电池组件和蓄电池为电源系统，控制器和逆变器为控制保护系统，负载为系统终端。如图 2.2 所示。太阳能能量通过太阳电池

组件转化为直流电力,再通过并网型逆变器将直流电能转化为与电网同频率、同相位的正弦波电流,一部分给当地负荷供电,剩余电力馈入电网。可见光伏并网逆变器为太阳能发电系统中的关键设备^[22]。

太阳能电池(称电池单元)与蓄电池(称电能储存单元)组成系统的电源单元,因此蓄电池性能直接影响着系统工作特性。

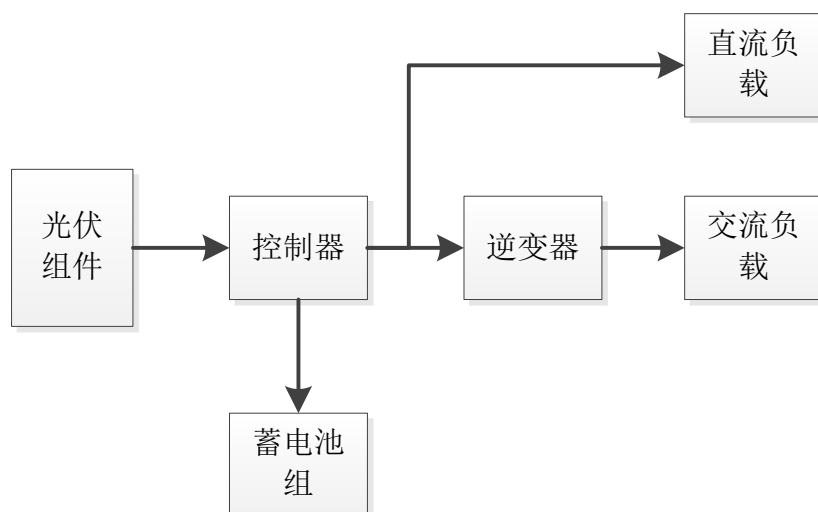


图 2.2 太阳能光伏发电系统

1、电池单元

太阳能电池发电的主要原理是半导体 P-N 的光电效应,如果在硅原子 4 个电子中掺入有 5 个电子的原子如磷原子,就成为带负电的 N 型半导体;若在纯硅中掺入有 3 个电子的原子,形成带正电的 P 型半导体。当 P 型和 N 型结合在一起时,接触面就会形成电势差,成为太阳能电池。当太阳光照射到 P-N 结后,带正电荷的空穴由 N 极区往 P 极区移动,电子由 P 极区向 N 极区移动,所谓光生伏打效应,简言之,就是当物体受到光照时,物体内的电荷发生变化而产生电动势和电流的一种效应。当太阳光照射半导体 P-N 时,就会在 P-N 结的空穴和电子发生移动,形成电流,两边出现电压,叫光生电压。这种能把光能转换成电能的能量转换器,就是太阳能电池。

由于技术和材料原因,单一电池的发电量是十分有限的,实用中的太阳能电池是单一电池经串、并联组成的电池系统,称为电池组件(阵列)。单一电池是一只硅晶体二极管,根据半导体材料的电子学特性,当太阳光照射到由 P 型和 N 型两种不同导电类型的同质半导体材料构成的 P-N 结上时,在一定的条件下,太阳能辐射被半导体材料吸收,在导带和价带中产生非平衡载流子即电子和空穴。同于 P-N 结势垒区存在

着较强的内建静电场，因而能在光照下形成电流密度，短路电流和开路电压。若在内建电场的两侧面引出电极并接上负载，理论上讲由 P-N 结、连接电路和负载形成的回路，就有“光生电流”流过，太阳能电池组件就实现了对负载的功率输出。理论研究表明，太阳能电池组件的峰值功率，由当地的太阳平均辐射强度与末端的用电负荷(需电量)决定。

2、电能储存单元

蓄电池组的作用是存储太阳电池阵列发出的电能，并可以随时向负载供电。在光伏发电系统中，蓄电池处于浮充放电状态。白天太阳电池方阵在给负载供电的同时，还给蓄电池充电。晚上负载用电全部由蓄电池供给。因此，要求蓄电池的自放电要小，而且充电效率要高，同时还要考虑价格和使用是否方便等因素。常用的蓄电池有铅酸蓄电池，和硅胶蓄电池。要求较高的场合，也用价格比较昂贵的镍镉蓄电池、镍氢电池或锂离子电池。目前，蓄电池技术是十分成熟的，但其容量要受到末端需电量，日照时间(发电时间)的影响。因此蓄电池瓦时容量和安时容量由预定的连续无日照时间决定，同时，蓄电池的特性也影响着系统的工作效率和特性。

3、控制器

控制器是对太阳能光伏系统，进行控制和管理的设备，主要由电子元器件、仪表、继电器、开关等组成，它的主要功能是使太阳能发电系统始终处于发电的最大功率点附近，以获得最高效率，这种技术叫做最大功率点跟踪技术 MPPT (Maxim Power Point Track)。而充电控制通常采用脉冲宽度调制技术即 PWM 控制方式，使整个系统始终运行于最大功率点附近区域。在不同类型的光伏发电系统中，充电控制器不尽相同，其功能多少及复杂程度差别很大，在太阳发电系统中，充电控制器的基本作用是，为蓄电池提供最佳的充电电流和电压。快速、平稳、高效的为蓄电池充电，并在充电过程中减少损耗、尽量延长蓄电池的使用寿命。同时保护蓄电池，避免过充电和过放电现象的发生。如果用户使用直流负载，通过充电控制器还能为负载提供稳定的直流电。

对于大中型光伏系统，控制器则还需要具有下列功能：

- a.信号检测，如输入电压、充电电压、输出电流、蓄电池温升等；
- b.设备保护，如系统中逆变电路出现故障，或负载短路就需要保护光伏系统不受损害；

c.故障诊断定位, 控制器可以自动检测故障类型, 指示故障位置;

d.运行状态指示, 一般通过指示灯、显示器等方式, 指示光伏系统的运行状态。

放电控制主要是指当电池缺电、系统故障, 如电池开路或接反时切断开关。

目前日立公司研制出了既能跟踪调控点, 又能跟踪太阳移动参数的“向日葵”式控制器, 将固定电池组件的效率提高了 50%左右。

4、DC-AC 逆变器

逆变器按激励方式, 可分为自激式振荡逆变和他激式振荡逆变。主要功能是将蓄电池的直流电逆变成交流电。通过全桥电路, 一般采用 SPWM 处理器经过调制、滤波、升压等, 得到与照明负载频率, 额定电压等匹配的正弦交流电供系统终端用户使用。由于太阳电池和蓄电池发出的都是直流电, 当负载是交流类型时, 逆变器就不可缺少。

按照用途, 逆变器分为离网型和并网型。并网型逆变器是并网型光伏发电系统的关键技术。并网逆变器与离网型逆变器不同之处在于, 它不仅要将直流电转换为交流电, 并且还可对转换的交流电的频率、电压、电流、相位、有功或无功、同步、电能品质(电压波动、高次谐波)等进行控制。并网型逆变器一般要包括 3 个部分: 逆变部分、控制部分和保护部分。逆变部分就是采用大功率晶体管, 将直流高速切割, 转换为交流电; 控制部分由电子回路构成, 起到控制逆变部分的作用; 保护部分也由电子回路构成, 起到保安作用。按照输出波形, 逆变器又可分为方波型和正弦波型两种; 前者电路简单, 造价低, 但谐波分量大, 只能用于要求不高的系统。正弦波逆变器造价相对较高, 但可以适用于各种负载。

对逆变器的基本要求是:

a.输出稳定, 包括交流电的电压和频率都要达到一定的稳定精度;

b.输出可调, 输出交流电的电压和频率, 应该在一定的范围内可以调节;

c.具有一定的过载能力;

d.输出电压波形谐波成分尽量少;

e.具有短路、过载、过热、过电压、欠电压等保护和报警功能;

f.启动平稳, 启动电流小, 运行稳定可靠;

g.换流损失小;

h.具有快速的动态响应。

可见，控制器和逆变器是太阳能光伏发电系统中关键设备，它决定了太阳能发电系统的品质^[23-27]。

5、微型逆变器

太阳能光伏微型逆变器是一种转换直流从单一太阳能电池组件至交流电的装置。它的优点是可以单独对每块太阳能电池组件做最大功率点跟踪，防止在遮挡情况下导致的单块组件对整串组件的影响，使整体的输出功率最大化，将发电效果最优化。与光伏数据采集网关组合，

还可用于监视各个模块的状态，检测出出现故障的模块，实现组件级监控。

6、光伏数据采集网关

通过 RS485、PLC、无线等方式与逆变器进行通信，收集逆变器的发电信息，并通过 GPRS、以太网、WIFI、3G 等方式传送到数据库的设备。

光伏数据采集器根据光伏电站大小分为小型光伏数据采集器和大型光伏数据采集器；光伏数据采集器根据监控方式分为普通光伏数据采集器和网络数据采集器；光伏网络数据采集器又分为 GPRS 光伏数据采集器、以太网数据采集器等；光伏数据采集器根据适用性又可以分为专用型光伏数据采集器和兼容性光伏数据采集器。

随着光伏网络监控宣传的力度不断加大和使用过的企业越来越多，网络监控将成为今后光伏电站售后管理的必备产品^[29-30]。

第三章 系统设计

一个完整的太阳能光伏发电系统，是由太阳能电池组件，逆变器，数据采集器和监控系统软件构成，监控系统软件又由固化在数据采集器中的下位机程序和上位机程序组成，共同完成对整个光伏发电系统的运行参数的监测与实时控制，真正实现无人职守的，少维护的，智能化的电站系统。

在本软件研制开发中，我们把软件工程的基本原理应用到实际工作中，作为项目组织者和规划者，遵照软件工程的设计原则，先后对本项目的实施进行了需求分析，通过需求分析制定了相应的开发计划、框架设计和测试计划，整个项目开发纳入软件工程质量控制过程，保证了软件系统开发的顺利完成。

3.1 系统需求分析

根据公司的业务要求，系统的总体目标是：为电站用户、安装商、系统管理员提供实时的数据监控服务及详细的历史报表，既能为决策层和管理层快速、准确、高质量地提供各种优化的太阳能监控系统信息和决策所需的支持信息，也能通过短信、邮件的报警方式及时的发现该系统中存在的问题，为产品改进提供重要的数据支持。

3.2 系统设计目标和原则

智能型光伏电站远程数据采集与监控系统主要分为三层：微型逆变器、光伏数据采集器、数据中心。系统总体结构框架如图 3.1 所示。

系统的第一层为微型逆变器，安装在每一块光伏组件上，负责将直流逆变为交流，并将所发的电送至电网；第二层为光伏数据采集器，负责收集微型逆变器的数据并进行存储，再通过 Internet 将数据发送到远程的数据中心；第三层为数据中心，由 App 服务器、Web 服务器和数据库服务器组成，主要负责数据接收、存储和提供前端 Web 网页显示。第一层与第二层之间采用电力载波进行通信，第二层与第三层之间采用 Internet 网络进行通信。

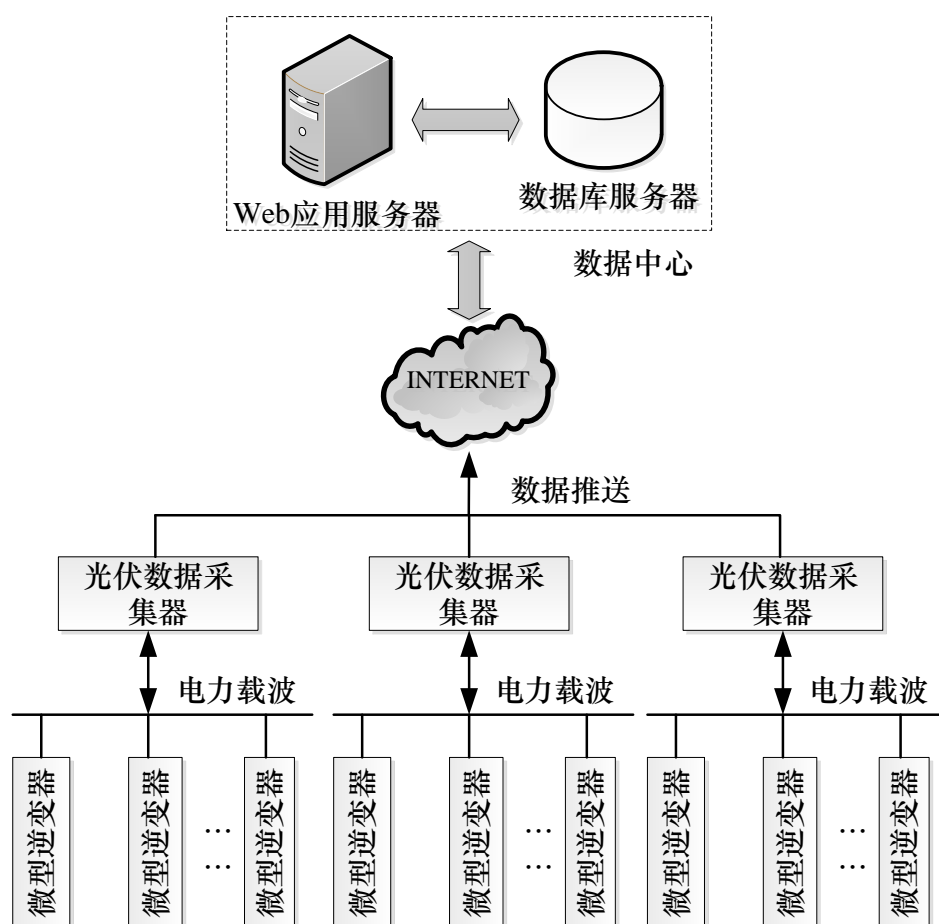


图 3.1 系统总体结构框架图

3.3 软件功能设计

3.3.1 数据接收程序

数据接收程序部署在应用服务器上，其功能分为以下几点：

- 1、数据接收：通过 Internet 接收分布在全球各地的光伏数据采集器的数据信号。
- 2、数据解析：按照约定的通讯规约将其采集到的数据信号进行解析。
- 3、数据存储：将解析后的字段按照一定的数据格式存入到数据库中。
- 4、日志记录：记录每一次的通讯的时间、原始数据信息、异常信息，方便日后的数据分析和数据追溯。

3.3.2 监控 Web 界面

B/S 模式介绍

B/S 模式即为浏览器和服务器（Browser/Server）模式。随着互联网技术的广泛使用，网络技术已被广泛用于在互联网上，但 Web 应用程序的初期呈现给观众的一些

文字信息都是静态的 HTML 页面。在这种结构中，网络技术，包括三个层次：浏览器，Web 服务器和数据库服务器。用户只需要使用浏览器软件代替各种客户端软件，所有的操作都通过浏览器完成。这种结构为核心的 Web 服务器，这是负责接受 HTTP 请求，根据 request 条件到数据库服务器获取数据，然后翻译成 HTML 及各种页面描述语言，返回 response 到发出请求的浏览器即呈现页面结果给用户。B/S 的结构图如图 3.2 所示^[14-15]。

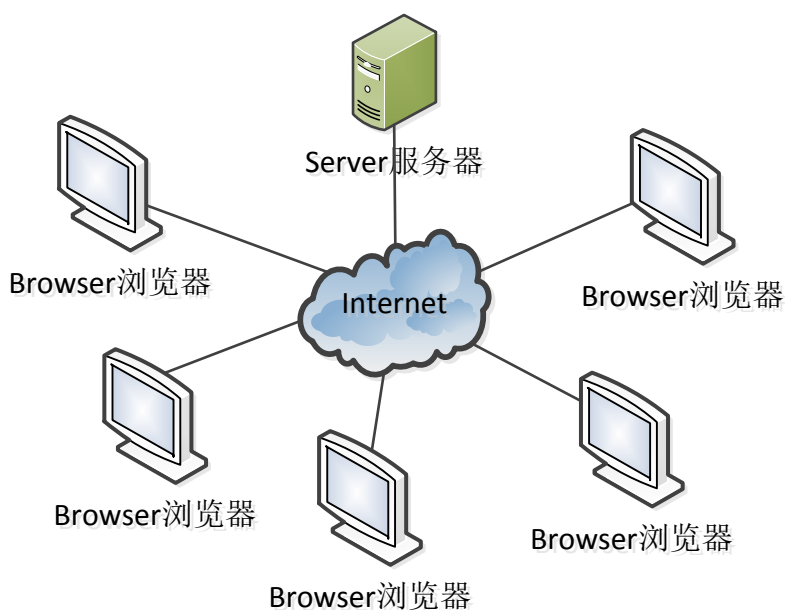


图 3.2 B/S 模式结构图

MVC 三层结构

MVC 模型-视图-控制器是一种软件体系结构，目前认为是一种在软件工程中使用的架构模式。该架构的核心思想是：将程序分成相对独立，而又能协同工作的三个部分，将用户的应用程序逻辑与用户界面的输入和展示隔离，通过使用 MVC 架构，允许独立开发每个模块，可以降低模块之间的耦合，单独测试和维护每一个分离的关注点，提高应用程序的可扩展性。在不同方面的应用（输入逻辑，业务逻辑和 UI 逻辑）场合下分开使用 MVC 结构的结果^[20]。

1、视图(View)

多个视图可以对应一个模块。根据 MVC 设计模式，多个视图对应一个模型，可以减少代码副本和代码维护，一旦模型发生变化时，也易于维护。

2、模型(Model)

模型层是应用程序的主体，它用于表示业务数据和业务逻辑，形成一个软件应用程序，用于存储，可选独立，数据的收集。一个单一的前端类，它可以沟通，任何用户界面（例如：控制台，图形用户界面，或 Web 应用程序）。

3、控制器(Controller)

控制器根据视图层传输过来的用户输入，结合业务逻辑代码调用不同的模型和视图，返回用户真正想要的信息。它提供一些措施，以改变模型的状态。控制器接收请求 request 并根据业务逻辑时间情况选择不同的 Service 服务来处理这些不同的请求，Service 层再调用数据层，从数据库中读写数据，再返回结果给 Service，Service 结合逻辑判断决定选用哪个视图显示模型处理后得到的数据，再作为 response 传给用户 [32]。

虽然 MVC 有不同的实现方式，但其主要控制流程一般如下：以某种方式的用户界面（例如，通过按鼠标按钮）进行用户交互。从用户界面，往往通过注册的处理程序或回调控制器处理输入事件，并转换成相应的用户操作的事件和对模型的理解。控制器通知用户操作模式，可能导致改变模型的状态。（例如，控制器更新用户的购物车）。视图查询模型来生成相应的用户界面（例如视图列出了购物车的内容）。从模型的观点得到了自己的数据。在某些实现中，控制器会发出一般性的指令呈现自己的观点。在其他国家，视图会自动通知状态变化（观察员），需要一个屏幕更新的模式。用户界面，等待进一步的用户交互，这将重新启动控制流程周期 [21][34-35][37]。

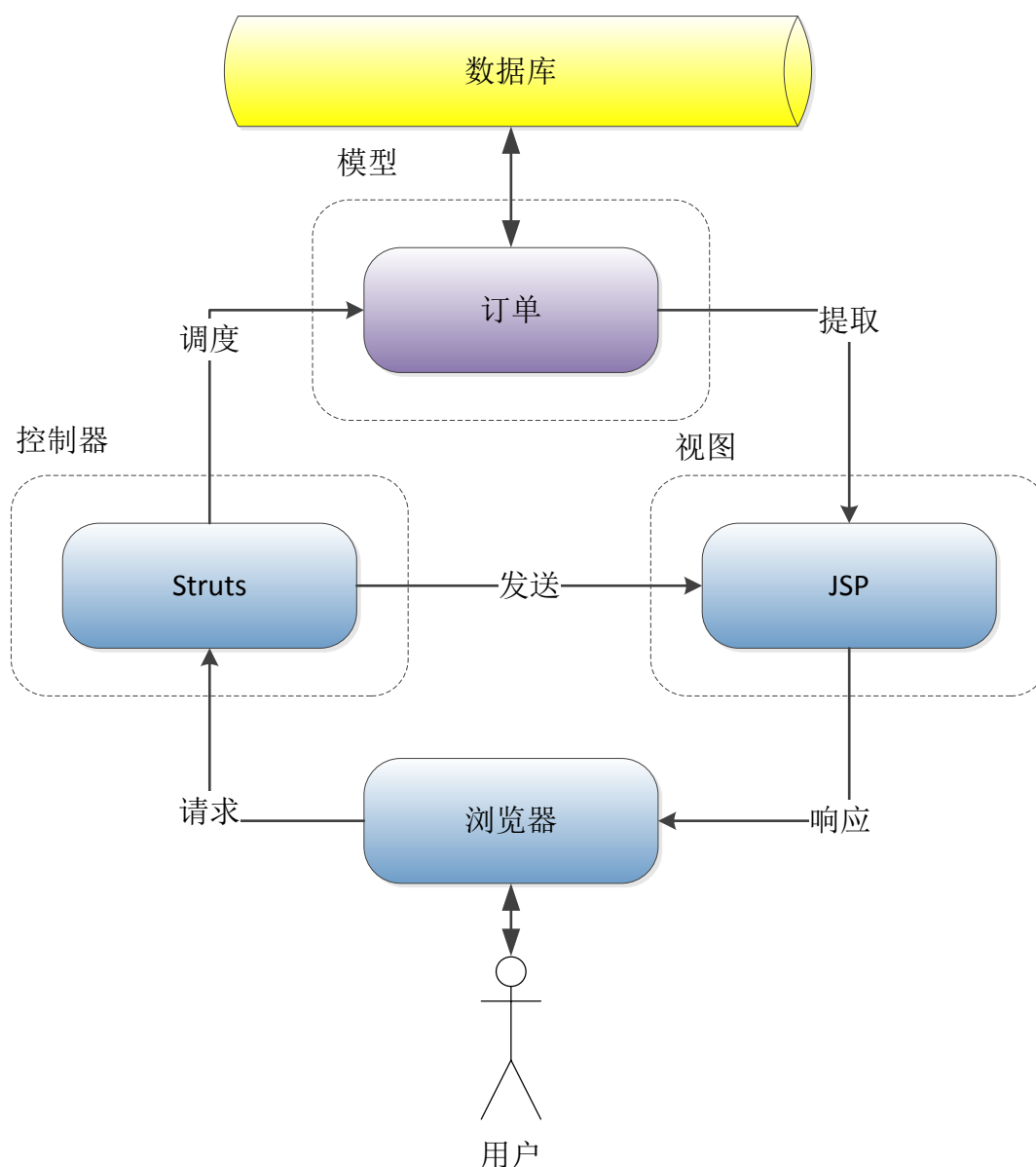


图 3.3 MVC 主要控制流程

3.4 数据库设计

MySQL 介绍

MySQL 是一个关系型数据库管理系统，由瑞典的 MySQLAB 公司开发，目前属于 Oracle 公司。MySQL 是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL 的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策，它分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其

是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库^[42-43]。

在本系统中，主要涉及到的数据库信息有：

用户信息表（用户 ID、用户名、密码、邮箱、姓名）。

表 3.1 用户信息表

列名	数据类型	描述	是否主键
ID	Int	用户 ID	是
UserID	Varchar(20)	用户名	否
Password	Varchar(20)	密码	否
Email	Varchar(20)	邮箱	否
UserName	Varchar(20)	姓名	否

电站信息表（电站 ID、电站名称、电站照片、国家、省\州、市、街道、时区、经度、纬度、电站容量、安装公司、联系方式）。

表 3.2 电站信息表

列名	数据类型	描述	是否主键
PlantID	Int	电站 ID	是
PlantName	Varchar(20)	电站名称	否
PlantPhoto	Bitmap	电站照片	否
Country	Varchar(20)	国家	否
Province	Varchar(20)	省\州	否
City	Varchar(20)	市	否
Street	Varchar(50)	街道	否
Timezone	Varchar(20)	时区	否
Longitude	Float(8)	经度	否
Latitude	Float(8)	纬度	否
PlantSize	Int	电站容量	否
Installer	Varchar(20)	安装公司	否
contactNumber	Int	联系方式	否

网关信息表（网关 ID、网关序列号）。

表 3.3 网关信息表

列名	数据类型	描述	是否主键
GatewayID	Int	网关 ID	是
GatewaySN	Varchar(20)	网关序列号	否

微逆信息表（微逆 ID、微逆序列号）。

表 3.4 微逆信息表

列名	数据类型	描述	是否主键
InverterID	Int	微逆 ID	是
InverterSN	Varchar(20)	微逆序列号	否

网关组表（ID、网关组 ID、网关 ID）。

表 3.5 网关组表

列名	数据类型	描述	是否主键
ID	Int	序号	是
GroupID	Int	网关组 ID	否
GatewayID	Int	网关 ID	否

电站网关组关系表（ID、电站 ID、网关组 ID）。

表 3.6 电站网关组关系表

列名	数据类型	描述	是否主键
ID	Int	序号	是
PlantID	Int	电站 ID	否
GroupID	Int	网关组 ID	否

微逆网关关系表（ID、微逆 ID、网关 ID）。

表 3.7 微逆网关关系表

列名	数据类型	描述	是否主键
ID	Int	序号	是
InverterID	Int	微逆 ID	否
GatewayID	Int	网关 ID	否

实时发电数据表（ID、日期时间、微逆 ID、电流、电压、功率、状态）。

表 3.8 实时发电数据表

列名	数据类型	描述	是否主键
ID	Int	序号	是
TimeStamp	Datetime	日期时间	否
InverterID	Int	微逆 ID	否
Voltage	Float(8)	电流	否
Current	Float(8)	电压	否
Power	Float(8)	功率	否
Status	Varchar(10)	状态	否

天发电量表（ID、日期、微逆 ID、日发电量）。

表 3.9 天发电量表

列名	数据类型	描述	是否主键
ID	Int	序号	是
Date	Date	日期	否
InverterID	Int	微逆 ID	否
Todayenergy	Float(8)	日发电量	否

故障信息表（故障 ID、发生时间、电站 ID、故障信息）。

表 3.10 故障信息表

列名	数据类型	描述	是否主键
FaultID	Int	故障 ID	是
FaultTime	Datetime	发生时间	否
PlantID	Int	电站 ID	否
Comment	Varchar(200)	故障信息	否

3.5 技术方案

Web 系统基于 ASP.NET MVC 来实现。采用 Code First 技术来实现 ORM（对象关系映射），避免使用复杂的 SQL 语句直接查询数据库，使代码易于维护。在此基础上抽象出 Service 层来封装常用业务逻辑，使用依赖注入来维护这些 Service。

3.5.1 上位机软件开发及生产环境

开发平台：

应用程序开发平台：Visual Studio 2012/.NET Framework 4.5

数据库开发平台：MySQL

主要 Web 框架：jQuery、ASP.NET MVC 3.0 和 Code First

其它框架/工具：Log4net 和 Ninject

生产平台：

Web 服务器平台：Windows Server 2008 R2/IIS 7.0

数据库平台：MySQL

3.5.2 项目模型

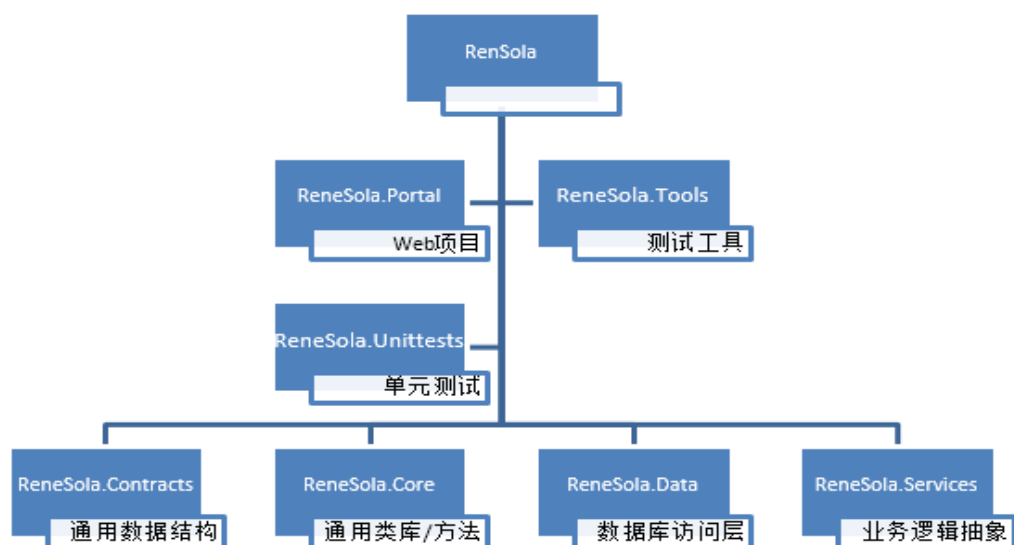


图 3.4 项目模型

ReneSola.Contract [Library]

定义常用数据结构,包括数据库表结构实体(定义在 ReneSola.Contract.DatabaseEntities 命名空间内)。

ReneSola.Core [Library]

定义了项目中用到的扩展方法,以及部分常用方法(如记录日志的帮助类)。

ReneSola.Data [Library]

实现了数据库访问层。使用 Code Frist,定义了 ReneSolaContext 和 DataCon text 分别实现用户数据库和产生数据库的 ORM。实现了产生数据的分割,在网关/电站扩展后可通过部署多个 DataContext 实现快速水平扩展。

ReneSola.Services [Library]

定义和实现了常用数据查询的操作接口,封装了查询操作,可在应用程序中直接调用。

ReneSola.Portal [Release]

项目的 Web 实现。采用 ASP.NET MVC 框架,分作 Controller/Model/View 三层,分别实现控制,数据模型,页面呈现三层。其中核心代码在 Controller 层。详细情况解析如下。

AccountController.cs: 采用 Membership 实现用户的登录注册等功能;

AlertController.cs: 实现 Alerts 增删改查等相关功能;

BaseController.cs: 所有 Controller 的基类,抽取定义公用函数;

CoreController.cs: 定义不属于特定业务逻辑范畴的方法,如天气获取、验证码生成等功能;

DeviceController.cs: 主要页面,实现了普通用户登录后看到的 Overview/Graphs/Benefits/Layout/Settings 等页面的业务逻辑;

DeviceController.Query.cs: 主要为 JsonResult 页面,提供数据供 Ajax 访问用以处理呈现;

HomeController.cs: 用户首页、预留。目前直接跳转到 Device 页面;

ReportController.cs: 报表页面,实现 Reports 的业务逻辑;

ViewSwitcherController.cs: 预留, 实现手机端与电脑端的版本切换功能。

ReneSola.Zeus [Release]

项目的接收端实现。采用 TCP 监听方式, 按照约定协议采集、分析、处理并储存数据, 在将来版本中将按照约定协议实现对网关的反控。默认监听端口为 3884。

ReneSola.Tools [Tools]

在开发调试中用到的小工具, 如 TCP 调试工具, 数据恢复工具, 仅供开发人员使用, 不属于发布内容。

ReneSola.Unittests [Test]

单元测试项目, 实现部分重要流程和方法的单元测试, 也包含部分 Console 调试工具, 模拟生成数据。

3.5.3 监控通讯协议

格式约定

1、通讯双方以客户端-服务器模式建立 TCP 连接, 以客户端驱动方式发起连接请求, 操作完成或超时后断开连接。

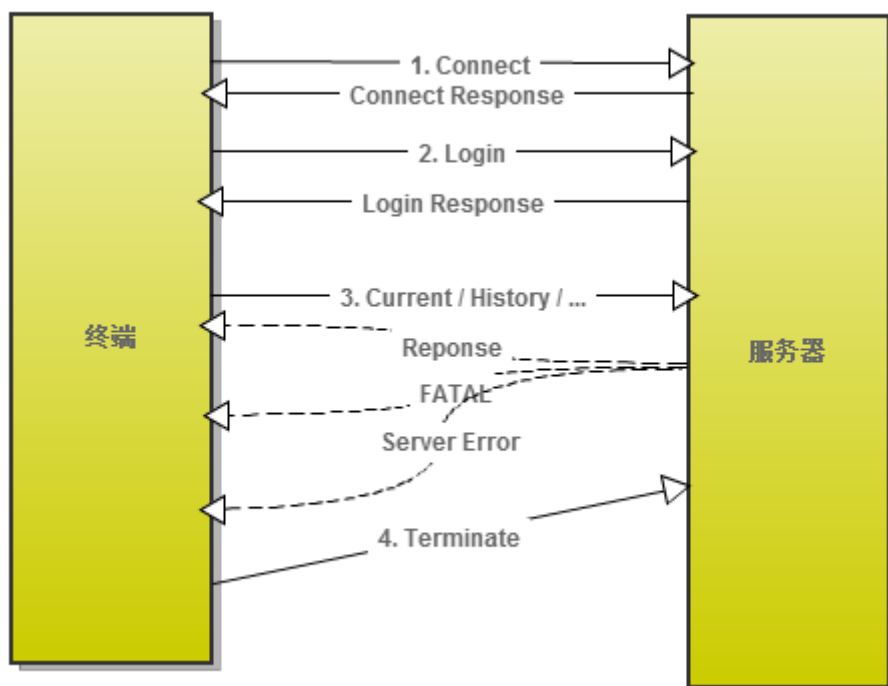


图 3.5 终端与服务器之间的通讯流程

2、客户端 LOGIN 获得的响应中如果 HASTO 字段为是,客户端应发送 GETTODO 命令,服务端收到该命令后发送操作指令。

3、消息包由消息头和消息体构成。所有字节为高位在前、低位在后。

4、一个常见的通讯流程如下:

消息头格式

表 3.1 消息头格式

名称	字节数	定义	备注
Length	4	消息长度	含消息头
Command	4	命令或响应类型	见下表定义
Sequence	4	消息流水号	从 0 开始以 1 递增,循环使用。 请求和对应的响应消息流水号相同
Device Type	1	设备类型	微型逆变器暂定为 1
Version	4	版本号	与发送程序的版本号对应,格式为年(以 2012 年为基数,如 2013 年为 01)+月+日+子版本号 (以此版本号作为远程更新的判定条件)

表 3.2 Command 定义

名称	值	描述	备注
CONNECT	0x00000001	请求连接	
CONNECT_RESP	0x80000001	请求连接响应	随机字符串
LOGIN	0x00000002	登录请求	字符串加密结果
LOGIN_RESP	0x80000002	登录请求响应	验证结果
TERMINATE	0x00000003	终止连接	不响应,直接中断
INITIALIZE_RESP	0x80000004	绑定服务器响应	
CURRENT	0x00000005	实时数据通知	
CURRENT_RESP	0x80000005	实时数据通知响应	
HISTORY	0x00000006	历史数据通知	
HISTORY_RESP	0x80000006	历史数据通知响应	
STATUS	0x00000007	设备状态通知	开关机
STATUS_RESP	0x80000007	设备状态通知响应	
TIME	0x00000008	服务器时间请求	
TIME_RESP	0x80000008	时间请求响应	
FATAL	0x80000010	请求包格式错误	
SERVER_ERROR	0x80000011	服务器错误	
GETTODO	0x00000012	获得操作列表	

0x00000001 – 0x00000099 预设为客户端发起的命令代码;

0x10000001 – 0x00000099 预设为客户端响应的命令代码。

表 3.3 服务器操作列表

RESTART	0x70000001	重启	重启网关
RESTART_RESP	0x10000001	重启请求响应	
UPDATE	0x70000002	程序更新	
UPDATE_RESP	0x10000002	程序更新响应	
READMIPARA	0x70000003	读取微逆参数	
READMIPARA_RESP	0x10000003	读取微逆参数响应	
WRITEMIPARA	0x70000004	修改微逆参数	
WRITEMIPARA_RESP	0x10000004	修改微逆参数响应	

0x70000001 – 0x70000099 预设为服务器端发起的代码；

0x80000001 – 0x80000099 预设为服务器端响应的代码。

消息体格式

1.CONNECT

名称	字节	说明	备注
SN	14		

2.CONNECT_RESP

名称	字节	说明	备注
RANDOM	16	随机字符串	

3.LOGIN

名称	字节	说明	备注
CIPHER	32	AES 加密结果	

4.LOGIN_RESP

名称	字节	说明	备注
STATUS	1	成功/失败	0/1
HASTODO	1	否/有	0/1

5.TERMINATE: 空消息

6.INITIALIZE_RESP

名称	字节	说明	备注
URI	不固定	服务器地址	

7. CURRENT_RESP

名称	字节	说明	备注
Status	1	接收成功/重传	0/1
TIME_START	5	时间	可空。不为空时为未上传历史数据的起始时间
TIME_END	5	时间	可空。不为空时为未上传历史数据的结束时间

8.HISTORY_RESP

名称	字节	说明	备注
Status	1	接收成功/重传	0/1

9.CURRENT

名称	字节	说明		备注	
TIME	5	时间		以 2000 年为基点的年月日时分, 如 1310312355	
COUNT	1	微逆数量		0-255	
STATUS	1	运行/故障/通讯异常		0/1/2	
ALLPOW	4	总输出功率		0.1W	
ALLTODAY	4	总日发电量		0.001kWh	
ALLTOTAL	8	总发电量		0.1kWh	
MI-1 第一台逆变器数据		MISN	4	微逆 SN 号	
		MIDCVOLT	2	直流电压	0.1V
		MIDCCUR	2	直流电流	0.1A
		MIACVOLT	2	交流电压	0.1V
		MIACCUR	2	交流电流	0.1A
		MIPOW	2	输出功率	0.1W
		MISTATUS	4	状态(故障码)	
		MITODAY	2	今日发电量	0.001kWh
		MITOTAL	4	总发电量	0.1kWh
		MIFREQ	2	频率	
		MITEMP	2	温度	
		Unuesd1	2		
		Unuesd2	2		
		Unuesd3	2		
MI-2 第二台逆变器数据					
...					
MI-N 第 N 台逆变器数据					

10.HISTORY

名称	字节	说明	备注
COUNT	2	历史数据条数	
DATA-1	23+34*N(微逆数)	第一条历史数据	数据结构参见实时数据结构体
DATA-2	23+34*N	第二条	
...		...	

11.STATUS

名称	字节	说明	备注
Status	1	开/关	0/1

12.STATUS_RESP

名称	字节	说明	备注
Status	1	接收成功/重传	0/1

13.TIME: 空消息

14. READMIPARA_RESP

名称		字节	说明	备注
COUNT		1	数量	
第一台微逆	MISN	4	微逆 SN 号	
	SW_VERSION	4	微逆软件版本号	只读
	VAC_MAX	2		0.1V
	VAC_MAX_2	2		0.1V
	VAC_MAX_3	2		0.1V
	TIME_VAC_MAX_2	2		0.1S
	TIME_VAC_MAX_3	2		0.1S
	VAC_MIN	2		0.1V
	VAC_MIN_2	2		0.1V
	VAC_MIN_3	2		0.1V
	TIME_VIN_MAX_2	2		0.1S
	TIME_VIN_MAX_3	2		0.1S
	FREQ_MAX	2		0.01Hz
	FREQ_MAX_2	2		0.01Hz
	FREQ1_MAX_3	2		0.01Hz
	TIME_FREQ_MAX_2	2		0.1S
	TIME_FREQ_MAX_3	2		0.1S
	FREQ_MIN	2		0.01Hz
	FREQ_MIN_2	2		0.01Hz
	FREQ_MIN_3	2		0.01Hz
	TIME_FREQ_MIN_2	2		0.1S
	TIME_FREQ_MIN_3	2		0.1S
	MAX_POWER_OUTPUT	2		0.1w
	MIPACKETREV_DAY	4	PLC 收包数	MIPACKETREV_DAY
	MIPACKETSEND_DAY	4	PLC 发包数	MIPACKETSEND_DAY
	RECONNECTION_TIME	2	重连时间	
	Unused2	2		
	Unused3	2		
	Unused4	2		
	Unused5	2		
	Unused6	2		
	Unused7	2		
	Unused8	2		
第 N 台微逆				

15.TIME_RESP

名称	字节	说明	备注
Time	5	服务器时间	

16. WRITEMIPARA

名称	字节	说明	备注
COUNT	1	数量	
第一台微逆	MISN	4	微逆 SN 号
	SW_VERSION	4	微逆软件版本号
	VAC_MAX	2	0.1V
	VAC_MAX_2	2	0.1V
	VAC_MAX_3	2	0.1V
	TIME_VAC_MAX_2	2	0.1S
	TIME_VAC_MAX_3	2	0.1S
	VAC_MIN	2	0.1V
	VAC_MIN_2	2	0.1V
	VAC_MIN_3	2	0.1V
	TIME_VIN_MAX_2	2	0.1S
	TIME_VIN_MAX_3	2	0.1S
	FREQ_MAX	2	0.01Hz
	FREQ_MAX_2	2	0.01Hz
	FREQ1_MAX_3	2	0.01Hz
	TIME_FREQ_MAX_2	2	0.1S
	TIME_FREQ_MAX_3	2	0.1S
	FREQ_MIN	2	0.01Hz
	FREQ_MIN_2	2	0.01Hz
	FREQ_MIN_3	2	0.01Hz
	TIME_FREQ_MIN_2	2	0.1S
	TIME_FREQ_MIN_3	2	0.1S
	MAX_POWER_OUTPUT	2	0.1w
	MIPACKETREV_DAY	4	PLC 收包数
	MIPACKETSEND_DAY	4	PLC 发包数
	RECONNECTION_TIME	2	重连时间
	Unused2	2	
	Unused3	2	
	Unused4	2	
	Unused5	2	
	Unused6	2	
	Unused7	2	
	Unused8	2	
第 N 台微逆			

17.FATAL: 空消息

18.SERVER_ERROR: 空消息

19.GETTODO: 空消息

20.RESTART: 空消息

21.RESTART_RESP

名称	字节	说明	备注
STATUS	1	是/否接收成功	0/1

22.UPDATE

名称	字节	说明	备注
URI	不固定	更新包地址	

23.UPDATE_RESP

名称	字节	说明	备注
Status	1	是/否接收成功	0/1

24. READMIPARA: 空消息

25. WRITEMIPARA_RESP

名称	字节	说明	备注
Status	1	写入成功/失败	0/1

第四章 系统主要功能与实现

4.1 接收端的功能与实现

网关登录

一旦有网关向接收端发送登录请求，接收端会首先返回一串 16 位的随机验证码至网关，网关通过约定的加密算法进行加密后返回密钥字符串，接收端完成解码后核对是否正确，如信息无误，返回成功。

加密方法代码如下：

```
public static class AESCryptography
{
    [SuppressMessage("Microsoft.Usage", "CA2202:Do not dispose objects multiple times")]
    public static byte[] Encrypt(string plaintext, byte[] key, byte[] iv)
    {
        // Check arguments.
        if (plaintext == null || plaintext.Length <= 0)
        {
            throw new ArgumentNullException("plaintext");
        }
        if (key == null || key.Length <= 0)
        {
            throw new ArgumentNullException("key");
        }
        if (iv == null || iv.Length <= 0)
        {
            throw new ArgumentNullException("key");
        }

        // Create an RijndaelManaged object
        // with the specified key and IV.
        using (var rijAlg = new RijndaelManaged())
        {
            rijAlg.Key = key;
            rijAlg.IV = iv;

            // Create a decryptor to perform the stream transform.
            ICryptoTransform encryptor = rijAlg.CreateEncryptor(rijAlg.Key, rijAlg.IV);
```

```

        // Create the streams used for encryption.
        using (var msEncrypt = new MemoryStream())
        {
            using (var csEncrypt = new CryptoStream(msEncrypt, encryptor,
CryptoStreamMode.Write))
            {
                using (var swEncrypt = new StreamWriter(csEncrypt))
                {
                    //Write all data to the stream.
                    swEncrypt.Write(plaintext);
                }
                return msEncrypt.ToArray();
            }
        }
    }
}

```

解密方法代码如下：

```

public static string Decrypt(byte[] cipherText, byte[] key, byte[] iv)
{
    // Check arguments.
    if (cipherText == null || cipherText.Length <= 0)
        throw new ArgumentNullException("cipherText");
    if (key == null || key.Length <= 0)
        throw new ArgumentNullException("key");
    if (iv == null || iv.Length <= 0)
        throw new ArgumentNullException("key");

    // Create an RijndaelManaged object
    // with the specified key and IV.
    using (var rijAlg = new RijndaelManaged())
    {
        rijAlg.Key = key;
        rijAlg.IV = iv;

        // Create a decryptor to perform the stream transform.
        ICryptoTransform decryptor = rijAlg.CreateDecryptor(rijAlg.Key, rijAlg.IV);

        // Create the streams used for decryption.
        using (var msDecrypt = new MemoryStream(cipherText))
        {
            using (var csDecrypt = new CryptoStream(msDecrypt, decryptor,
CryptoStreamMode.Read))

```

```

        {
            using (var srDecrypt = new StreamReader(csDecrypt))
            {
                // Read the decrypted bytes from the decrypting stream
                // and place them in a string.
                return srDecrypt.ReadToEnd();
            }
        }
    }
}

```

数据解析

网关登录成功后，会向接收端发送一串实时数据，包括直流电压、直流电流、功率、今日发电量、总计发电量几个字段，接收端收到数据后，按照约定的协议格式进行解析。

数据解析部分代码如下：

```

public class MicroInverterData
{
    public static string OkStatusString = InverterDisplayModel.OkStatusString;

    public string SerailNumber { get; set; }
    /// <summary>
    /// 0.1 V
    /// </summary>
    public ushort DCVolt { get; set; }
    /// <summary>
    /// 0.1 A
    /// </summary>
    public ushort DCCurrent { get; set; }
    /// <summary>
    /// 0.1 W
    /// </summary>
    public ushort Power { get; set; }
    public string Status { get; set; }
    /// <summary>
    /// 0.001 kWh
    /// </summary>
    public ushort TodayKilowattHour { get; set; }
    /// <summary>
    /// 0.1 kWh

```

```

/// </summary>
public uint TotalKilowattHour { get; set; }
public byte[] GetBytes()
{
    var bytes = new byte[ByteLength];
    // bug, wrong bytes
    var sn = SerailNumber;
    if (sn.StartsWith("0x"))
    {
        sn = sn.Substring(2);
    }
    HexToByte(sn).CopyTo(bytes, 0);
    ZeusBitConverter.GetBytes(DCVolt).CopyTo(bytes, 4);
    ZeusBitConverter.GetBytes(DCCurrent).CopyTo(bytes, 6);
    ZeusBitConverter.GetBytes(Power).CopyTo(bytes, 8);
    HexToByte(Status).CopyTo(bytes, 10);
    ZeusBitConverter.GetBytes(TodayKilowattHour).CopyTo(bytes, 14);
    ZeusBitConverter.GetBytes(TotalKilowattHour).CopyTo(bytes, 16);
    return bytes;
}

private static byte[] HexToByte(string hexString)
{
    var returnBytes = new byte[hexString.Length / 2];
    for (var i = 0; i < returnBytes.Length; i++)
        returnBytes[i] = Convert.ToByte(hexString.Substring(i * 2, 2), 16);
    return returnBytes;
}

public readonly static int ByteLength = 20;
}
}

```

数据存储

将实时数据解析后，会对解析后的数据进行判定和处理，其流程如图 4.1 所示。

在收到某台网关的数据时，如果这台网关中的微逆 ID 在数据库中不存在，首先会创建此网关和其微逆的对应关系，将信息新增至 inverter 和 grid 表中。

代码部分如下：

```

protected override void Seed(MapContext context)
{
    // Create indexes
    context.Database.CreateIndex("Inverter", "GridId");
}

```

```
context.Database.CreateIndex("Grid", "SerialNumber");
context.Database.CreateIndex("User", "Name");

// Seed
var configs = new List<Config>
{
    new Config
    {
        CurrentPreferredUserDatabaseId = 0
    }
};

configs.ForEach(m => context.Configurations.Add(m));
context.SaveChanges();

base.Seed(context);
}
```

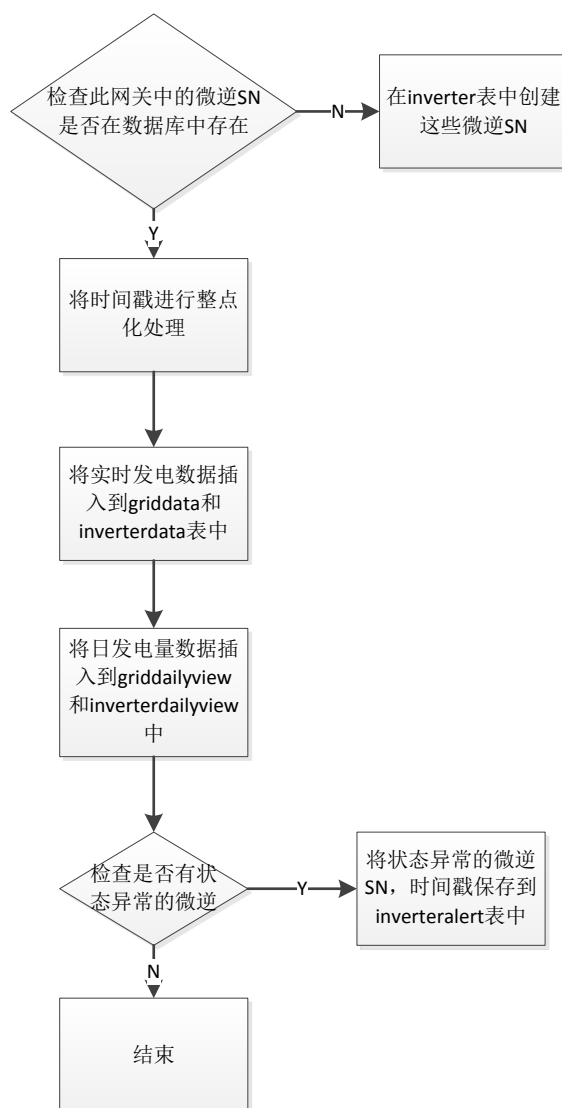



图 4.1 数据存储流程图

如果存在，则找到此 `gridSerialNumber` 对应的 `gridid` 将实时发电数据保存至 `griddata` 表和 `inverterdata` 表。由于数据是 5 分钟上传一次，为了保证数据的统一性，这里会使用 `Timestamp.ToFiveMinutes` 方法将 `timestamp` 时间戳的值转化为 0 分或 5 分的整点数据。之后，将网关的日发电量和每台微逆的日发电量分别保存至 `griddailyview` 表和 `inverterdailyview` 表，目的是在 Web 中 `graph` 模块的曲线绘制。最后，如果某台微逆有异常状态，会将这台微逆的 `SN` 以及发生故障的时间保存至 `inverteralert` 表。

其代码部分如下：

```

public CurrentResponseBody Store(IReneSolaService service, string gridSerialNumber)
{
    var response = new CurrentResponseBody

```

```

        {
            Status = CurrentRequestReceivedStatus.Success,
            EndTime = null,
            StartTime = null
        };

using (var mapContext = DataStore.CreateMappingDataContext())
{

    var grid = service.GridService.FindBySerialNumber(gridSerialNumber, true);
    if (grid == null)
    {
        LogUtility.LogDebug(string.Format("grid with serialnumber {0} not exists, creating",
gridSerialNumber));
        service.GridService.Create(gridSerialNumber, true);
        LogUtility.LogDebug(string.Format("grid with serialnumber {0} created",
gridSerialNumber));

        grid = service.GridService.FindBySerialNumber(gridSerialNumber, true);
    }

    foreach (var microInverterData in MicroInverters)
    {
        var inverter =
            mapContext.Inverters.SingleOrDefault(l => l.SerialNumber ==
microInverterData.SerailNumber);
        if (inverter != null)
            inverter.Enabled = true;
    }

    grid.Actived = true;
    var nearestTimeslot = Timestamp.ToFiveMinutes();

    lock (Locker)
    {
        var duplicateDataInDatabase =
service.GridDataService.GetGridDataByGridAndDate(grid.Id, nearestTimeslot);
        // drop the last data if there is already one in database
        if (duplicateDataInDatabase != null)
        {
            LogUtility.LogDebug(string.Format("duplicated data for grid {0} at {1}", grid.Id,
Timestamp.ToString(CultureInfo.InvariantCulture)));
            return response;
        }
    }
}

```

```

    }
}

var gridData = new GridData
{
    Volt = GridVolt,
    GridId = grid.Id,
    Status = Status,
    Timestamp = nearestTimeslot,
    Year = nearestTimeslot.Year,
    Month = nearestTimeslot.Month,
    Day = nearestTimeslot.Day,
    Date = nearestTimeslot.Date
};

List<InverterData> inverterData;
List<InverterAlert> alertData;
ProcessInverterData(service, mapContext, grid, gridData, out inverterData, out alertData);

using (var dataContext = DataStore.CreateDataContextByGrid(grid.Id))
{
    var latestGridDataId = service.GridDataService.Latest(grid.Id, Timestamp);
    var latestGridData = dataContext.GridDataSet.SingleOrDefault(l => l.Id ==
latestGridDataId);

    if (latestGridData != null)
    {
        CheckMissedData(latestGridData, response, dataContext, grid);
    }

    // set gridData
    gridData.Power = inverterData.Sum(l => l.Power);
    gridData.EnergyGenerated = inverterData.Sum(l => l.GenerationToday);
    dataContext.GridDataSet.Add(gridData);

    // add inverterData
    foreach (var data in inverterData)
    {
        dataContext.InverterDataSet.Add(data);
        ProcessInverterDailyView(dataContext, data, gridData);
    }

    ProcessGridDailyView(dataContext, grid, gridData);

    // update/add inverter daily view

```

```
dataContext.SaveChanges();

// do the alert thing after all is saved in database
foreach (var inverterAlert in alertData)
{
    inverterAlert.GridDataId = gridData.Id;
    service.AlertService.AddAlert(grid.DatabaseId, inverterAlert);
}

var gridLifetimeEnergy = 0;
foreach (var data in inverterData)
{
    service.InverterDataService.SetTotalEnergyOfInverter(grid.Id, data.InverterId,
data.GenerationTotal);

    gridLifetimeEnergy += data.GenerationTotal;
}

service.GridDataService.SetTotalEnergyOfGrid(grid.Id, gridLifetimeEnergy);
}

mapContext.SaveChanges();
return response;
}
```

4.2 Web 界面展示及功能

4.2.1 登录界面

供终端用户登录使用。另集成了地图功能，可显示电站在全球的数量分布情况。登录界面如图 4.2 所示。



图 4.2 登录界面

4.2.2 注册界面

注册界面如图 4.3 所示。

Replus
RenovaSola

Customer Register

* E-mail Please input a valid Email address, used for login and password retrieving

* Name First Name Surname

* Password 6-16 characters, case sensitive

* Re-type Password 6-16 characters, case sensitive

* Name of Your Plant Maximum 20 Letters

Upload Image of Your Plant 未选择文件

Street

* City

* State / Province Afghanistan

* Country Afghanistan

Zip Code

* Mark your plant

* Timezone (UTC) Casablanca

* Temperature Unit Fahrenheit

* Power of Plant (KW) 0

* Currency Unit USD 0 / 1 kWh

Module Supplier Renesola

* Gateway S/N The 14 digit number on a bar code sticker which at the back of the Gateway

* Installer

* Installer email

Contact Person

Contact Number

* Type In Verification Code Click me, Verify Code display

* Required Fields

[About](#) [Contact](#) [Investors](#) [Legal](#) [Help](#)

Copyright © 2012 Renesola All rights reserved

图 4.3 注册界面

注册信息包括：用户名、姓名、密码、电站名称、电站图片、国家、州、城市、街道、邮政编码、电站位置、时区、温度单位、电站容量、钱币换算、组件类型、网关 sn 号、安装商、安装商邮箱、联系人、联系人电话，其中打星号的为必填项。完成注册后该网关 sn 即与电站用户名建立对应关系。

4.2.3 主界面

主界面由 3 部分组成：overview、graphs、benefits。如图 4.4 所示。

Overview 主要是显示当前电站每台微逆的实时发电数据，如功率、当日发电量、

过去 7 天发电量、过去 4 周发电量、总累计发电量。

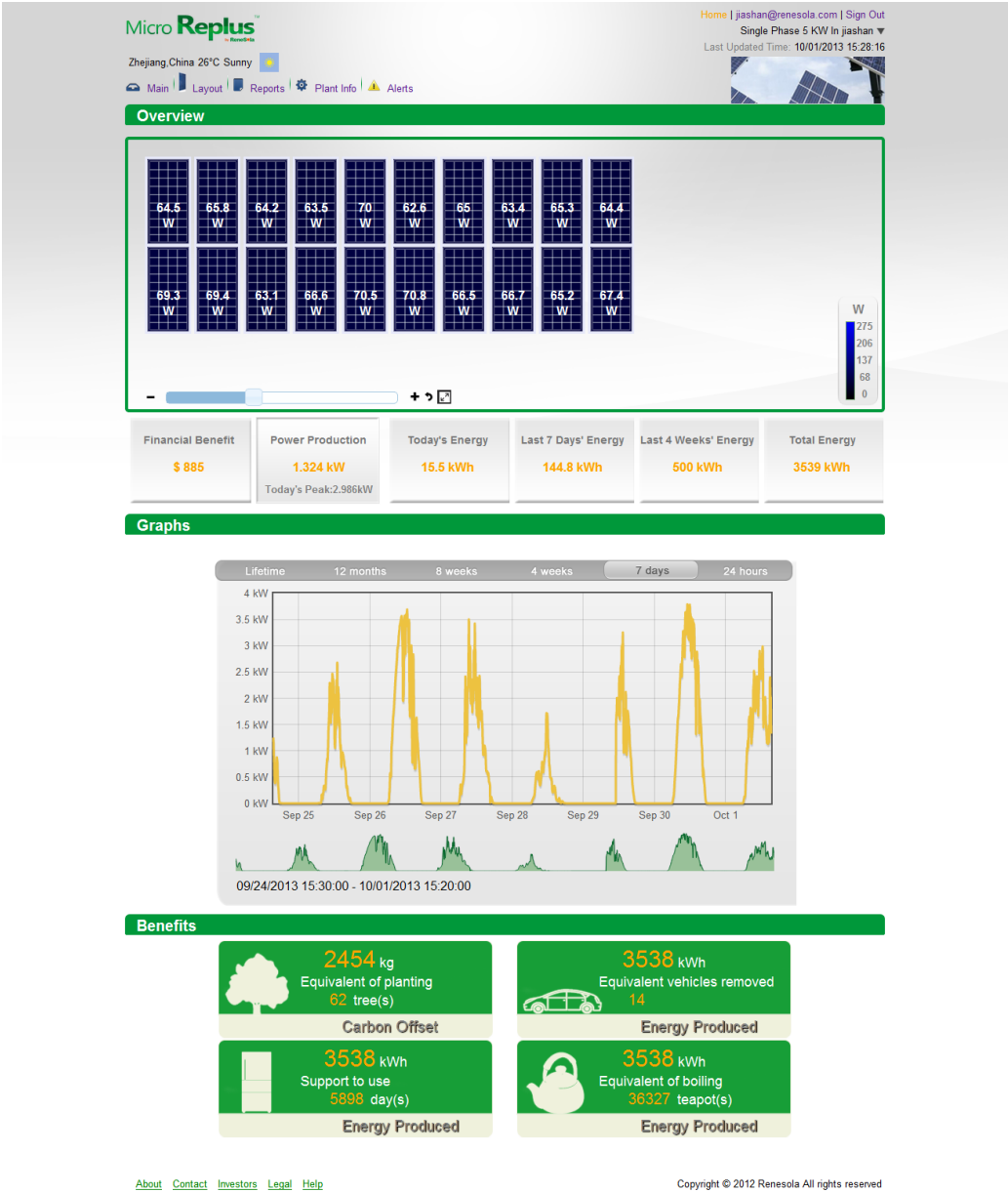


图 4.4 主界面

Graphs 主要显示该电站的功率曲线图与发电量曲线图。

Benefits 主要显示一些节能减排的数据信息。

4.2.4 布局界面

用户可在此界面上更改布局，保存成功后与 overview 中显示布局一致。

布局界面如图 4.5 所示。

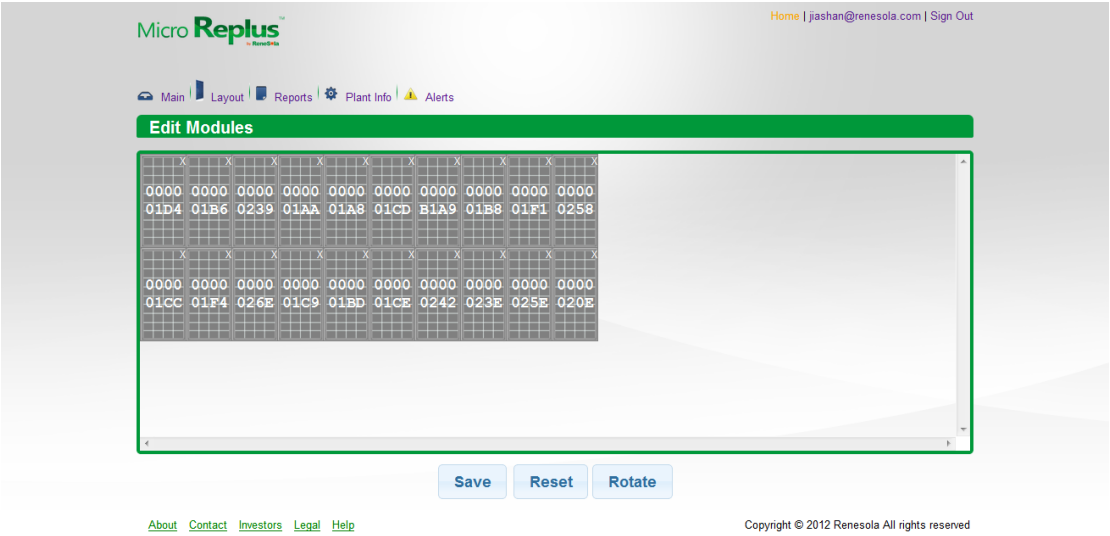


图 4.5 布局界面

4.2.5 报表界面

可查询单个网关或整个电站的报表信息,支持 csv 文件导出。报表共分为 4 种:

- 1、日报表: 可查询单独微逆或网关的每 5 分钟数据, 查询时段为每日的 4:00-21:00。
- 2、周报表: 同日报表, 查询时间段为 7 天。
- 3、月报表: 查询所选月份每日的发电量。
- 4、年报表: 查询所选年份每月的发电量。

报表界面如图 4.6 所示。

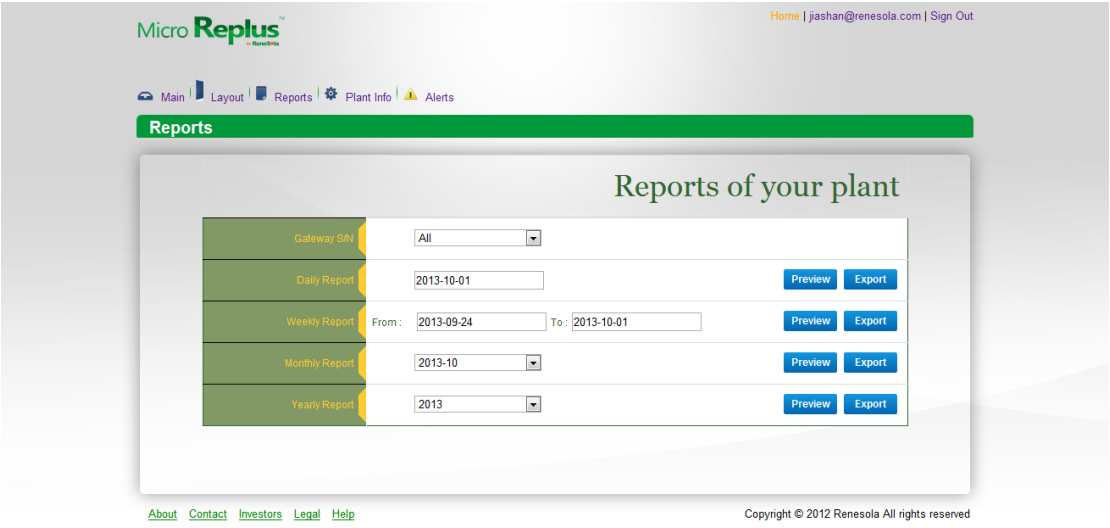
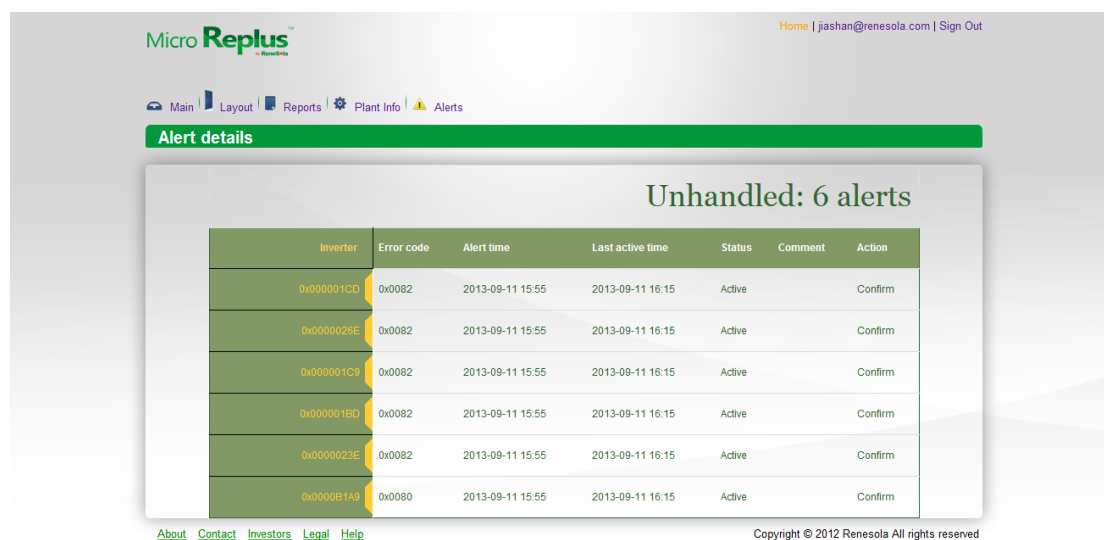


图 4.6 报表界面

4.2.6 历史故障界面

根据网关发过来的每台微逆的 error code 进行判定,只要 error code 不为 0(正常),就作为 alert 进行记录。如时间连续,记录的时候会合并为一条。

历史故障界面如图 4.7 所示。



Unhandled: 6 alerts						
Inverter	Error code	Alert time	Last active time	Status	Comment	Action
0x000001CD	0x0082	2013-09-11 15:55	2013-09-11 16:15	Active		Confirm
0x0000026E	0x0082	2013-09-11 15:55	2013-09-11 16:15	Active		Confirm
0x000001C9	0x0082	2013-09-11 15:55	2013-09-11 16:15	Active		Confirm
0x000001BD	0x0082	2013-09-11 15:55	2013-09-11 16:15	Active		Confirm
0x0000023E	0x0082	2013-09-11 15:55	2013-09-11 16:15	Active		Confirm
0x000001A9	0x0080	2013-09-11 15:55	2013-09-11 16:15	Active		Confirm

图 4.7 历史故障界面

第五章 系统的测试

5.1 系统测试计划

测试是项目管理系统顺利实施的一项重要工作,通过测试可确保软件系统的安全性、与可靠性。在智能型光伏电站的远程数据采集与监控系统中,由于用户遍布全球,所以此系统必须具备良好的稳定性与高并发性。此系统测试主要分为:网站服务器和数据库服务器的压力测试和浏览器的页面加载时间测试^[31]。

5.2 系统测试环境

本系统配备了以下测试环境:

网站服务器	CPU	Inter(R) Core™ i5-3470 CPU @ 3.20GHz
	内存	4GB
	操作系统	Windows 8 Enterprise
数据库服务器	CPU	AMD Athlon™ II X2 220 Processor 2.80 GHz
	内存	4GB
	操作系统	Windows 7 Enterprise

5.3 系统的测试方法与工具

测试脚本	Visual Studio 2012 Ultimate, C#
测试数据收集	Performance Monitor
测试报告	Visual Studio 2012, Excel, Word
监控工具	Visual Studio 2012, PerfMon, Event Viewer
其他	HTTPWatch, Fiddler

5.4 测试结果与总结

5.4.1 压力测试

1、测试设置。注:运行时间为1小时。

用户数	50
延迟时间 (Think Time)	0
延迟配置 (Think Profile)	关

2、性能验收标准: 以下数值需要在网站实际运行中评估。

反应时间	?
每月请求数	?
每天请求数	?
每秒请求数（RPS）	?
RPS 峰值	?

3、测试场景：测试用例分布%

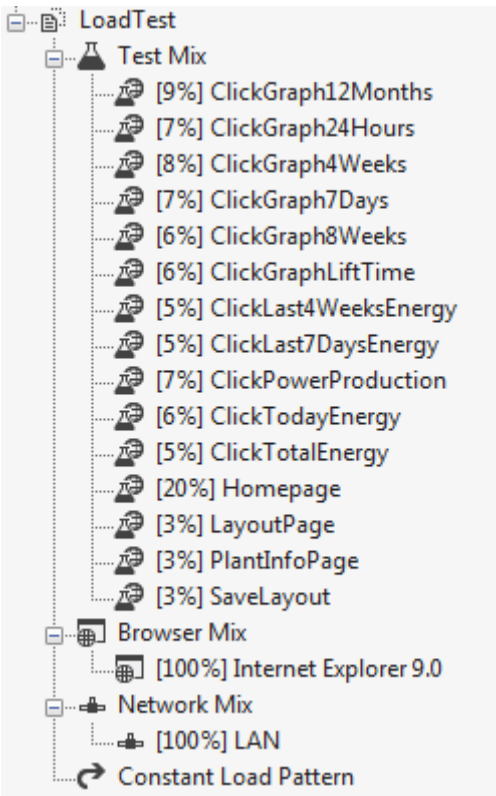


图 5.1 测试用例分布

4、页面反应时间（Page Response Time）。

Counter	Instance	Category	Compu...	Color	Range	Min	Max	Avg.
Key Indicators								
<input checked="" type="checkbox"/> User Load	_Total	LoadTest...	LILIAN-...		100	50	50	50
<input checked="" type="checkbox"/> Pages/Sec	_Total	LoadTest...	LILIAN-...		1,000	41.6	209	170
<input checked="" type="checkbox"/> Avg. Page Time	_Total	LoadTest...	LILIAN-...		1,000	0.16	1.45	0.26
<input checked="" type="checkbox"/> Errors/Sec	_Total	LoadTest...	LILIAN-...		10	0	0	0
<input checked="" type="checkbox"/> Threshold Violations/...	_Total	LoadTest...	LILIAN-...		10	0	1.60	0.74
Page Response Time								
<input checked="" type="checkbox"/> Avg. Page Time	3{GET}	LoadTest...	LILIAN-...		6	0.30	3.18	0.46
<input checked="" type="checkbox"/> Avg. Page Time	GetPowerGraphForLife{PO...	LoadTest...	LILIAN-...		6	0.074	0.61	0.16
<input checked="" type="checkbox"/> Avg. Page Time	3{GET}	LoadTest...	LILIAN-...		6	1.17	2.96	1.49
<input checked="" type="checkbox"/> Avg. Page Time	GetCurrencyData{POST}	LoadTest...	LILIAN-...		6	0.14	1.91	0.22
<input checked="" type="checkbox"/> Avg. Page Time	3{GET}	LoadTest...	LILIAN-...		6	0.29	5.39	0.52
<input checked="" type="checkbox"/> Avg. Page Time	GetPowerGraphForLastEig...	LoadTest...	LILIAN-...		6	0.079	0.80	0.16
<input checked="" type="checkbox"/> Avg. Page Time	GetEnergyForLastFourWee...	LoadTest...	LILIAN-...		6	0.098	1.62	0.19
<input checked="" type="checkbox"/> Avg. Page Time	GetEnergyForLife{POST}	LoadTest...	LILIAN-...		6	0.12	1.36	0.20
<input checked="" type="checkbox"/> Avg. Page Time	GetPowerGraphForLastSev...	LoadTest...	LILIAN-...		6	0.065	0.64	0.16
<input checked="" type="checkbox"/> Avg. Page Time	GetPowerGraphForTwenty...	LoadTest...	LILIAN-...		6	0.090	1.31	0.16
<input checked="" type="checkbox"/> Avg. Page Time	GetEnergyForToday{POST}	LoadTest...	LILIAN-...		6	0.11	2.07	0.19
<input checked="" type="checkbox"/> Avg. Page Time	GetPowerGraphForToday{...	LoadTest...	LILIAN-...		6	0.078	1.64	0.16
<input checked="" type="checkbox"/> Avg. Page Time	SaveLayout{POST}	LoadTest...	LILIAN-...		6	0.11	1.16	0.18
<input checked="" type="checkbox"/> Avg. Page Time	GetEnergyForLastEightDay...	LoadTest...	LILIAN-...		6	0.090	0.51	0.19
<input checked="" type="checkbox"/> Avg. Page Time	GetPowerGraphForSevenD...	LoadTest...	LILIAN-...		6	0.077	1.12	0.16
<input checked="" type="checkbox"/> Avg. Page Time	GetPowerGraphForLastFou...	LoadTest...	LILIAN-...		6	0.062	0.93	0.16
<input checked="" type="checkbox"/> Avg. Page Time	GetPowerGraphForLastTw...	LoadTest...	LILIAN-...		6	0.082	0.74	0.16

图 5.2 页面反应时间

5、测试报告：CPU、Memory 和 RPS。

a.网站服务器

性能计数器	最小	最大	平均
% Processor Time	19.698%	54.373%	41.496%
Available MBytes	1,778.000	1,913.000	1,849.000
Requests/Sec	85.762	193.595	169.546

CPU 使用率

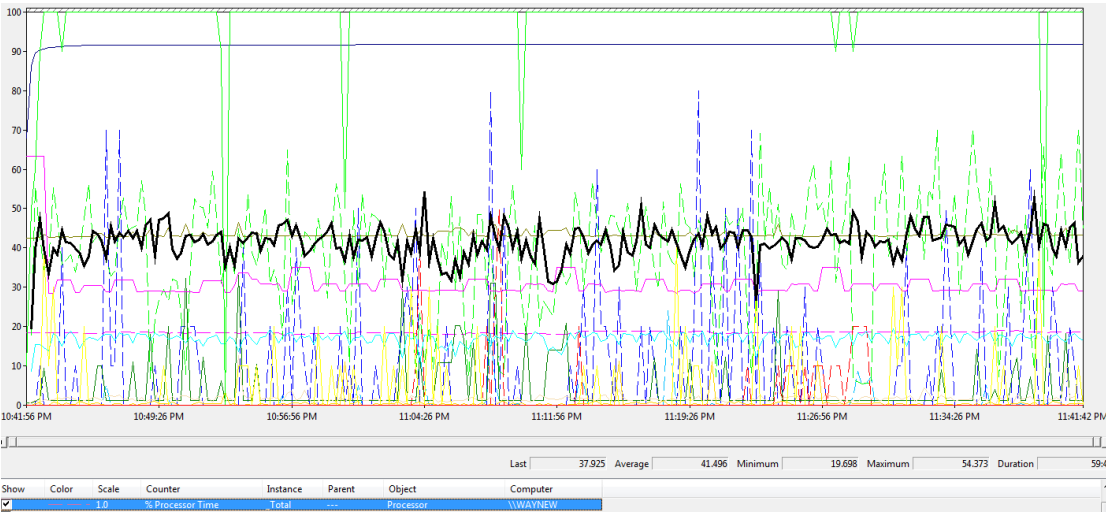


图 5.3 网站服务器 CPU 使用率

内存使用

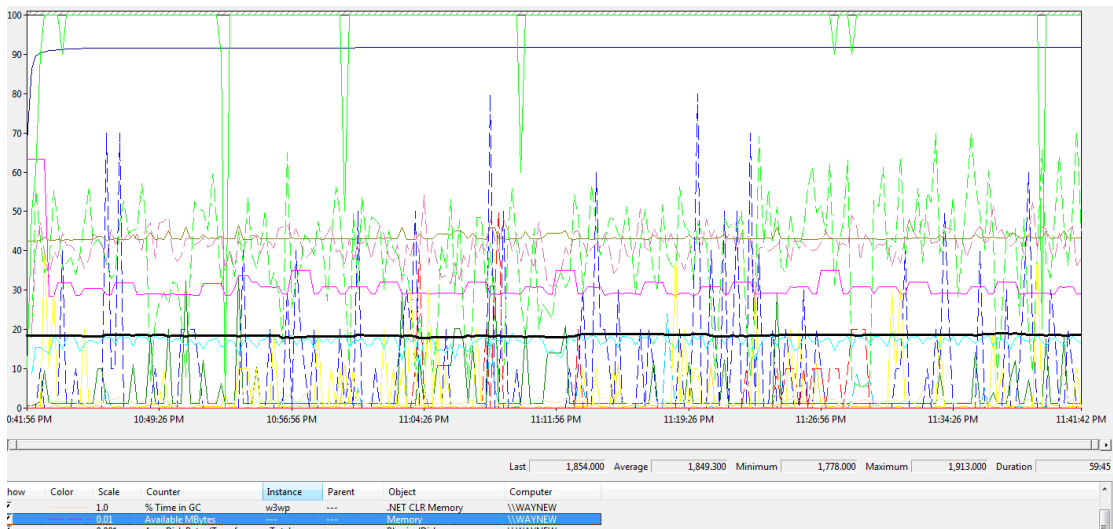


图 5.4 网站服务器内存使用

每秒请求数 Request/Sec

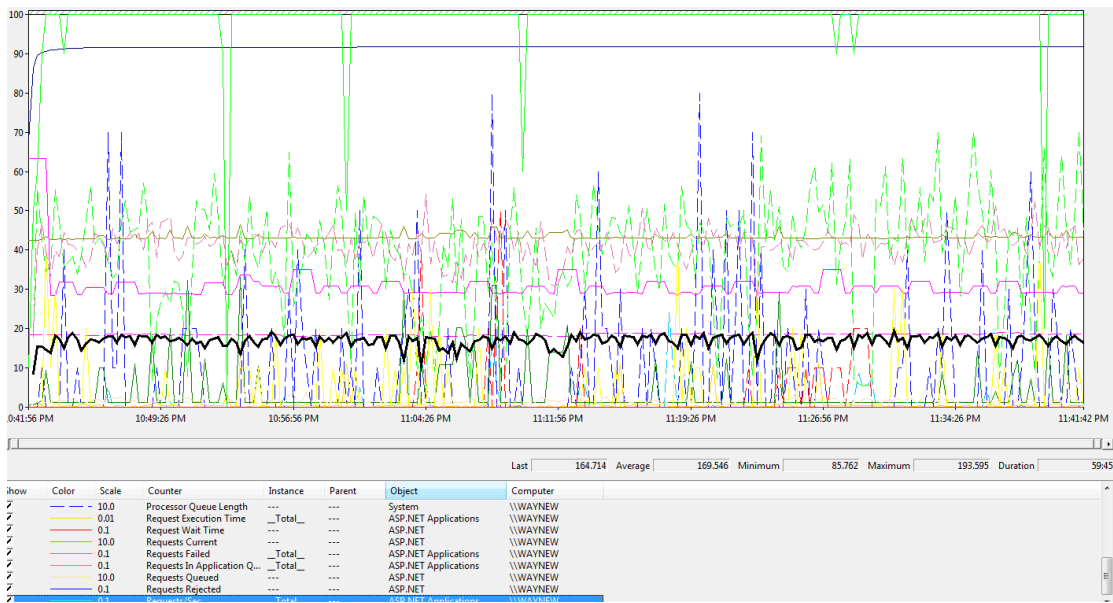


图 5.5 网站服务器每秒请求

b.数据库服务器

性能计数器	最小	最大	平均
% Processor Time	27.337%	77.055%	59.861%
Available MBytes	1,908.000	2,040.000	1,981.390

CPU 使用率

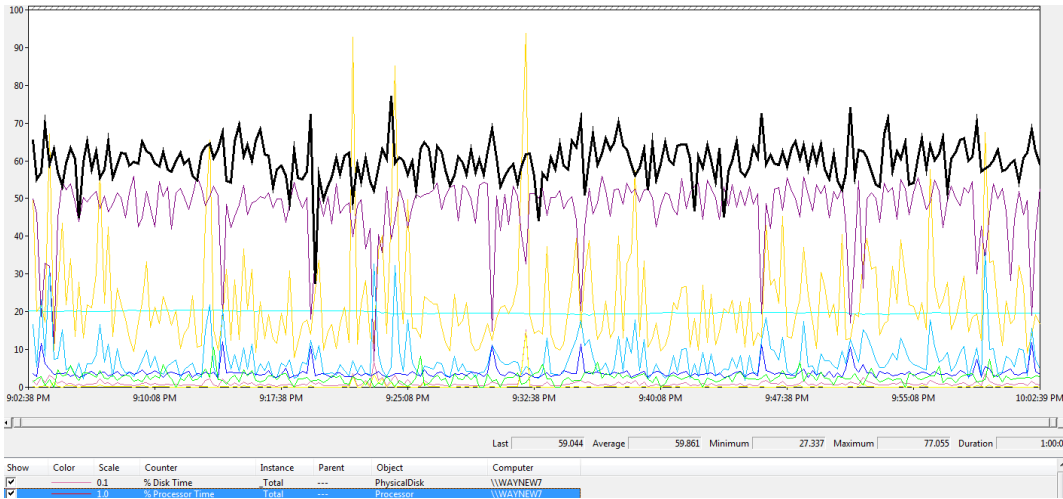


图 5.6 数据库服务器 CPU 使用率

内存使用

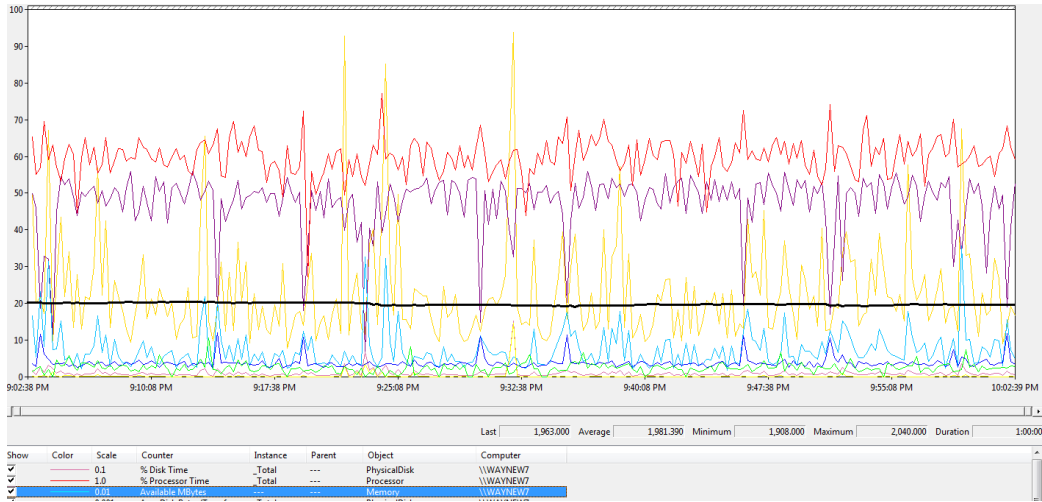


图 5.7 数据库服务器内存使用

6、性能指标和报告：性能计数器。

a.数据库服务器

序号	类别	性能计数器	最大	最小	平均
1	CPU	% Processor Time	77.055%	27.337%	59.861%
2	Memory	% Available Mbytes	2,040.000	1,908.000	1,981.390
3	Disk I/O	% Disk Time	151.987	2.447	9.929
4	Disk I/O	Avg. Disk Queue Length	1.520	0.024	0.099
5	Disk I/O	Avg. Disk Read Queue Length	1.466	0	0.025
6	Disk I/O	Avg. Disk Write Queue Length	0.356	0.017	0.074
7	Disk I/O	Avg. Disk sec/Read	0.106	0	0.025
8	Disk I/O	Avg. Disk sec/Transfer	0.009	0.001	0.002
9	Disk I/O	Disk Writes/sec	121.207	23.786	38.402

b.网站服务器

序号	类别	性能计数器	最大	最小	平均
1	CPU	% Processor Time	54.373%	19.698%	41.496%
2	CPU	System\Processor Queue Length	8	0	0.875
3	Memory	% Available Mbytes	1,778.000	1,913.000	1,849,000
4	Memory	Memory\Pages/sec	2,398.403	0	60.900
5	Memory	Server\Pool Nonpaged Failures	0	0	0
6	Disk	PhysicalDisk\Avg. Disk sec/Transfer	70,478.769	5,409.858	39,237.281
7	Disk	PhysicalDisk\Current Disk Queue Length	5	0	0.158
8	Process	Process(w3wp)\Private Bytes	464,117,760	424,800,256	433,153,553
9	.NET CLR Exceptions	# of Excepts Thrown	271,690,000	76	135,189.954
10	.NET CLR Memory	% Time in GC	12.532	0	1.714
11	.NET CLR Memory	Gen 2 Collections	499	2	241.546
12	.NET CLR Memory	Large Object Heap size	33,146,536	565,200	3,754,460
13	ASP.NET Apps	Cache Total Entries	633	284	311.683
14	ASP.NET Apps	Cache Hit Ratio	91.726	67.966	91.541
15	ASP.NET Apps	Errors Total/sec	0	0	0
16	ASP.NET Apps	Requests Failed	0	0	0
17	ASP.NET Apps	Requests In Application Queue	0	0	0
18	ASP.NET Apps	Requests/Sec	193.595	85.762	169.546
19	ASP.NET Apps	Request Execution Time	1,132	0	49.704
20	ASP.NET	Request Wait Time	0	0	0
21	ASP.NET	Requests Current	10	0	9.790
22	ASP.NET	Requests Queued	4	0	0.304
23	ASP.NET	Requests Rejected	0	0	0

5.4.2 页面加载时间测试

页面加载时间（Page Load Time）

注：PLT1-没有任何缓存情况下的页面加载时间。

PLT2-有缓存情况下的页面加载时间。

区域	浏览器	PLT1	PLT2
美国	IE10	5.575	4.248
	IE9	6.594	4.748
	IE8	11.971	10.100
	Firefox	6.501	4.273
	Chrome	6.290	4.080
	Safari	7.583	6.442
	Opera	10.295	9.297
欧洲	IE10	13.842	8.648
	IE9	9.272	5.803
	IE8	16.960	13.471
	Firefox	13.867	11.243
	Chrome	12.860	8.230
	Safari	18.197	16.982
	Opera	24.479	20.975

第六章 总结和展望

6.1 总结

本文研究的智能型光伏电站的远程数据采集与监控系统的设计与实现,为太阳能电站的智能化,提供了一套完整的解决方案,也为逆变器生产厂家的售后维护,提供了强有力的支持。该系统可以让电站管理人员进行统一的监测和管理,无需到现场逐台设备查看状况,更有利于进行数据汇总、生成曲线、数据分析,更加便于进行远程管理,大量节约逆变器售后的人力成本。

光伏电站一般建设在屋顶或者偏僻的地区,现场查看不是很方便,而且发电量直接和经济效益挂钩,因此远程数据采集与监控系统有利于电站管理者及时发现并解决问题,且可以通过手机、邮件等方式通知管理者,第一时间发现问题,并通过远程查看数据来判断问题的原因。

本文通过对光伏电站监控系统的研究,详细说明了该系统的架构,从需求、逻辑设计、流程处理上阐述了接收端程序的功能及实现方式,配合 Web 用户界面,实现了一套可远程操控、管理光伏电站的监控系统。最后通过对网站服务器和数据库服务器的压力测试,以及多浏览器的兼容性和页面加载时间测试,验证了此套系统在性能上的稳定性和可靠性。主要工作如下:

- 1、深入调研、了解了目前国内外主流光伏监控系统的用户需求,通讯方式,基于 TCP/IP 自定义了一套可适配于多种设备类型的网络通讯协议。

- 2、深入分析了基于 SOCKET 的网络编程技术,对多线程技术进行了研究,分析对比了几种不同的 I/O 模型,设计出具有高效率、高并发性、高稳定性的数据接收程序,并详细介绍了其工作流程和部分代码实现。

- 3、基于 ASP.NET MVC 来设计监控 web 用户界面,进行了主界面(实时数据)、布局功能、报表功能、报警功能几大功能模块设计,介绍其功能及截图展示。

- 4、搭建测试环境,使用 PerfMon, Event Viewer 等测试软件,分别针对网站服务器和数据库服务器进行了压力测试,在不同国家的服务器上对 web 的访问进行了页面加载时间的测试。

本系统已经在多个光伏微型逆变器并网电站实际使用，相较于传统的光伏电站监控系统，从实际使用的情况表明，该系统能够实时准确的传输数据，实时性和远程数据的传输能力都有很大的提高，达到了预期的效果。另外，本系统所设计的通讯协议具有很强的通用性和兼容性，根据不同设备类型定义，便可以兼容多种的智能设备数据采集和监控。

6.2 展望

由于时间、精力有限，也为了适应将来的需要，获取更好的用户体验，智能型光伏电站的远程数据采集与监控系统还存在一些有待完善的地方，主要为以下几个方面：

- 1、增加电站安装商、分区管理员等多角色系统，不同的角色对应不同的权限，可在配置界面做一些全局参数的配置。
- 2、增加远程修改参数功能，针对一些特定机型可能要在多个国家或地区进行使用，所以必须具备电网参数修改的功能，该功能在通讯协议中已经具备，但接收端和界面中尚未实现。
- 3、丰富日志管理系统，更方便的对日志进行查询、分析、删除。

参考文献

- [1] 耿亚新, 周新生. 太阳能光伏产业的理论及发展路劲[J]. 中国软科学. 2010,(4): 20-25
- [2] 杜勇慧. 太阳能光伏产业市场调查报告[R]. 北京: 中国电力企业联合会. 2010
- [3] Alfnov. 2011 海外太阳能光伏发电上网补贴政策现状一览[J]. 能源经济技术, 2011,(4)
- [4] 许继. 2011 太阳能光伏产业发展报告[R]. 北京: 许继集团智能电网研究中心. 2011
- [5] 工信部. 太阳能光伏产业“十二五”发展规划[R]. 北京: 工业和信息化部, 2011
- [6] 谷依露. 我国太阳能产业发展现状与发展路径研究[J]. 开发研究, 2011,(5):39-42
- [7] 朱永彬. 50kWp 屋顶太阳能光伏发电系统研究[D]. 天津: 天津大学, 2010
- [8] Zawodny, High Performance MySQL, New York City: Oreilly & Associates Inc., 2005:124-154
- [9] 陈鹏. 基于微电网的电力市场交易及其经济运行[D]. 北京: 北京交通大学. 2011
- [10] 胡阿芹, 孟祺. 我国太阳能光伏发电产业发展研究[J]. 湖北社会科学, 2011,(4)
- [11] 王志娟. 太阳能光伏技术[M]. 北京: 太阳能利用培训丛书, 2009, (1)
- [12] 赵玉文, 王斯成, 王文静等. 中国光伏产业发展研究报告[R]. 国家发改委/全球环境基金/世界银行中国可再生能源发展项目. 2008,(6)
- [13] 王长贵, 王斯成编著. 太阳能光伏发电实用技术[M]. 化学工业出版社, 2003
- [14] 吴功宜等编著. 计算机网络应用技术教程[M]. 清华大学出版社. 2002.
- [15] 胡伏湘等. 计算机网络技术教程[M]. 清华大学出版社. 2004.
- [16] 李征译. VisualC++.NET 开发手册[M]. 清华大学出版社. 2003.
- [17] 裘宗燕译. C++程序设计语言[M]. 机械工业出版社. 2002.
- [18] 毕潜, 李飞. 利用 ASP+ADO 访问站点数据库[J]. 计算机应用研究, Vol.14 (05), 2000. pp.51-53
- [19] 郭乐明. 基于 J2EE 和 AJAX 技术的实验室管理信息系统[J]. 长江大学学报(自然科学版)
- [20] 石磊, 陈朔鹰. 基于 MVC 设计模式的 Struts 框架的研究[J]. 微型电脑应用,

- Vol.24 (03),2005. pp. 47-49
- [21] 吴昊, 李志蜀, 杨正海. 基于 Struts 框架构建 GSM 电子运行维护系统[J]. 计算机应用与软件, Vol.24 (05), 2005. pp. 53-57
- [22] 张筱文等. 光伏电站监控系统的设计[J]. 电工电气, 2010.9:12-20.
- [23] 冯旭, 方晨, 刘昊. 基于 Web 的太阳能电站远程监控中心设计[J]. 现代电子技术, 2008,24:135
- [24] J. W. Amulf, O. Heinz. Wind and Photovoltaic in the European Union[J]. Renewable and Sustainable Energy Reviews. 2004.8(2): 157-182.
- [25] Y.Tsur and A.Zemel, Long-term Perspective on the Development of Solar Energy[J]. Solar Energy. 2000. 68(5):379-392.
- [26] Jean-Baptiste. Lesourd. Solar Photovoltaic Systems: the Economics of a Renewable Energy Resource[J]. Environmental Modeling & Software. 2001.16(2):147-156.
- [27] Chengmei Lin, Guisheng Jie, Zhenxing Wu. Design of Networked Monitoring System of PV Grid-Connected Power Plant [C], 2011 International Conference on Electronic&Mechanical Engineering and Information Technology. Haerbin.2011.
- [28] 孔明, 艾树峰, 杜辉. 基于 TCP/IP 的嵌入式 Webserver 网络测控系统设计[J]. 电讯技术. 2005.5:167-171.
- [29] 郭剑鸣. EAC 电能量数据采集自动化管理系统[J]. 广东输电与变电技术, Vol.24 (06), 2004.pp. 56-62
- [30] 蒋严庆. 电能量自动采集及管理系统的应⽤[J]. 华东电力, Vol.24(10), 2004. pp. 51-57
- [31] 赵雷. 基于 SOA 的自动化测试系统的设计及实现[D]. 北京邮电大学, 2007.
- [32] Ronald J. Norman. 面向对象系统分析与设计[M], 北京:清华大学出版社. 2000: 45-50
- [33] 金雪云. ASP. NET 简明教程[M], 北京:清华大学出版社, 2000: 80-110
- [34] 邓玉龙. MVC 设计模式在电子商务系统中的设计与应⽤[N]. 南京邮电学院学报, 2002,22(2): 80-83
- [35] 马争. MVC 设计模式在网管系统中的应⽤与研究[J]. 电子科技大学学报, 2005, 35(5):638-641

- [36] 廖若雪. ASP. NET 动态网站编程指南[M]. 北京:机械工业出版社, 2001: 95-113
- [37] 李朔. ASP. NET 环境下 MVC 模式的一种实现方法与应用[C]. 华东交通大学学报, 2008(6):22-25
- [38] 张海藩. 软件工程[M]. 北京:清华大学出版社, 1998: 220-223
- [39] 万建成, 卢雷编著. 软件体系结构的原理组成与应用[M]. 北京:科学出版社, 2002: 32-210
- [40] 王少锋. 面向对象技术 UML 教程[M]. 北京:清华大学出版社, 2004: 67-78
- [41] 冀振燕. UML 系统分析设计与应用案例[M]. 北京:人民邮电出版社, 2003: 3-9
- [42] 萨师煊, 王珊. 数据库系统概论[M]. 北京:高等教育出版社. 2000: 281-296
- [43] 王珊, 陈红. 数据库系统原理教程[M]. 北京:清华大学出版社. 2000: 257-286
- [44] Robert Powell, Richard Weeks. C# and the NET Framework[M], Australia: John Wiley & Sons Inc., 2001: 34-37
- [45] Karli Watson, Christian Nagel, Beginning Visual C#[M], Australia John Wiley & Sons Inc., 2006:178-195

攻读学位期间公开发表的论文

朱锐, 王林. 基于电力载波的光伏数据采集系统设计[J]. 中国科技信息, 2013, 472:115

致 谢

本文是在我的导师王林副教授的悉心指导下完成的。首先要感谢在本文从选题、分析、研究到编写整个过程中，我的导师给予我的帮助与指导。由于我的工作在外地，与导师不能经常见面，所以一直是保持着邮件沟通。即使如此，王老师在百忙之中对每一封邮件都认真、细心且及时的回复，让我能第一时间得到反馈，不但对其中的问题给予一一解答，而且给我许多建议与启发，使我受益良多。尤其是在论文的修改过程中，从论文的结构、格式、技术性中每一处细小的地方，都给了我很大的帮助，倾注了很多的心血。经过这段时间的学习，直到论文的顺利完成，使我的专业技术水平有了很大的提高，我要向王林老师表示感激和崇高的敬意。

与此同时，我也感谢和我一起攻读学位的同学，在我遇见困难和疑惑的时候，你们总是不厌其烦的给予我热心的帮助和支持，让我克服困难，完成论文的写作。

还要感谢项目组的老师们，感谢他们抽出时间对本论文提出宝贵的意见和建议，使本论文更加完善。

感谢工作单位的同事、领导们对我工作的支持与技术上的帮助。

最后，向对我的论文进行审阅、批评与指正的各位老师专家致以深深的谢意！

作者: [朱锐](#)
学位授予单位: [苏州大学](#)

参考文献(16条)

1. [谷依露](#) [我国太阳能产业发展现状与发展路径研究](#)[期刊论文]-[开发研究](#) 2011(05)
2. [朱永彬](#) [50kWp屋顶太阳能光伏发电系统研究](#)[学位论文]硕士 2010
3. [陈鹏](#) [基于微电网的电力市场交易及其经济运行](#)[学位论文]硕士 2011
4. [胡阿芹, 孟祺](#) [我国太阳能光伏发电产业发展研究](#)[期刊论文]-[湖北社会科学](#) 2011(04)
5. [毕潜, 李飞](#) [利用ASP+ADO访问站点数据库](#)[期刊论文]-[计算机应用研究](#) 2000(05)
6. [石磊, 陈朔鹰](#) [基于MVC设计模式的Struts框架的研究](#)[期刊论文]-[微型电脑应用](#) 2005(03)
7. [吴昊, 李志蜀, 杨正海](#) [基于Struts框架构建GSM电子运行维护系统](#)[期刊论文]-[计算机应用与软件](#) 2005(05)
8. [张筱文, 郑建勇](#) [光伏电站监控系统的设计](#)[期刊论文]-[电工电气](#) 2010(09)
9. [冯旭, 方晨, 刘昊](#) [基于Web的太阳能电站远程监控中心设计](#)[期刊论文]-[现代电子技术](#) 2008(24)
10. [孔明, 艾树峰, 杜辉](#) [基于TCP/IP的嵌入式WebServer网络测控系统设计](#)[期刊论文]-[电讯技术](#) 2005(05)
11. [郭剑鸣](#) [EAC电能数据采集自动化管理系统](#)[期刊论文]-[广东输电与变电技术](#) 2004(06)
12. [蒋严庆](#) [电能量自动采集及管理系统的應用](#)[期刊论文]-[华东电力](#) 2004(10)
13. [赵雷](#) [基于SOA的自动化测试系统的设计及实现](#)[学位论文]硕士 2007
14. [邓玉龙](#) [MVC设计模式在电子商务系统中的研究与应用](#)[期刊论文]-[南京邮电学院学报\(自然科学版\)](#) 2002(02)
15. [MVC设计模式在网管系统中的应用与研究](#)[期刊论文]-[电子科技大学学报](#) 2005(05)
16. [李朔, 李培松](#) [ASP.NET环境下MVC模式的一种实现方法与应用](#)[期刊论文]-[华东交通大学学报](#) 2008(03)

引用本文格式: [朱锐](#) [智能型光伏电站的远程数据采集与监控系统的设计与实现](#)[学位论文]硕士 2013