

编号: _____



桂林电子科技大学
GUILIN UNIVERSITY OF ELECTRONIC TECHNOLOGY

毕业设计说明书

题 目: 光伏发电系统
监控装置设计

学 院: 机电工程学院

专 业: 电气工程及其自动化

学生姓名: 李东琪

学 号: 1200120314

指导教师: 罗 奕

职 称: 高 工

题目类型: ☐理论研究 ☐实验研究 ☐工程设计 ☒工程技术研究 ☐软件开发

2016 年 6 月 3 日

摘 要

近年来，在全球能源危机不断加剧的情况下，世界上很多国家开始投入对新能源的探索和发展，我国也紧随其步伐，制定了相关的发展及推广使用计划，明确了发展安全清洁高效的现代能源技术，推动新能源生产和消费革命。在太阳能光伏发电的应用领域，保证太阳能发电系统稳定可靠的运行是基础，也是提高太阳能转换率必需具备的条件，所以工作状态监控是太阳能光伏发电系统的重要组成部分。

本文通过对目前太阳能发电技术的发展背景以及现状的分析，结合相关理论知识，对离网型光伏发电系统的组成结构进行了设计和分析，并结合智能电网的大主题，设计了一套基于 STC89C52 单片机的智能化太阳能监控装置，其主要功能包括了对光伏方阵工作温度、工作电压、工作电流以及环境光照强度的参数进行采集，并通过 RS-232 总线传输至 PC 端，供工作人员实时分析及控制，对于其中的温度、光强采集制作了硬件演示设备，实现了对参数的采集和显示，同时通过上位机与下位机之间的串口通信实现了人机互动。课题还针对太阳能的最大功率跟踪、提高太阳能转换效率的解决办法、温度控制充电电压等典型问题提出了可行性策略。

关键词：太阳能；光伏；单片机；监控，

Abstract

In recent years, with the increasing acute global energy crisis, many countries begin to explore and develop the new energy investment in the world. China also follows the pace and make a lot of plans that are concerning develop and promote the new energy .Our country defines the development of safe, clean and efficient modern energy technologies, what is more, China should promote the new energy production and consumption of the revolution. In the field of solar energy's application, to ensure the stability and reliability of the solar power system is the basis and it is also a condition of improving the conversion rate of solar energy. So the monitoring device is an important part of the solar photovoltaic power system.

Through the analysis of the current status of solar power generation technology and the development background of solar power generation technology also combined with theory knowledge, this paper is carried out the whole analysis and design of off-grid photovoltaic system's composition structure, combined with the theme of the smart grid.I have designing a set of intelligent solar monitoring device based on STC89c52 single-chip microcomputer, the main features include the work of photovoltaic array 's temperature, voltage, current and ambient light intensity parameters were collected and send it to the PC terminal through the RS-232 bus, the worker can timely to analysis and control for it.This project makes the temperature and light intensity collection hardware demonstration equipment, the demonstration equipment have parameter acquisition and display's functions and through serial port to communication between the PC and worker, realize the human-computer interaction. The topic also make the idears of solar energy maximum power tracking, the solution of the solar energy conversion efficiency and the temperature control of the charging voltage and so on.

Key words: solar energy; photovoltaic; single-chip microcomputer; monitoring,

目 录

摘 要.....	I
ABSTRACT.....	II
1 绪论.....	1
1.1 太阳能发电技术.....	1
1.1.1 发展背景.....	1
1.1.2 技术原理及应用.....	1
1.2 太阳能发电监控装置.....	2
1.3 本文主要研究内容.....	3
2 光伏发电系统组成结构设计.....	4
2.1 光伏发电系统结构及原理.....	4
2.2 电池单元.....	5
2.2.1 太阳能电池的分类.....	5
2.2.2 太阳能电池的工作原理.....	6
2.2.3 影响转换率的因素.....	6
2.3 控制器单元.....	6
2.4 逆变器单元.....	7
2.4.1 逆变器的分类.....	7
2.4.2 离网型光伏逆变器.....	7
2.4.3 并网型光伏逆变器.....	8
2.4.4 光伏逆变器性能参数.....	8
2.5 蓄电池储能单元.....	9
2.5.1 光伏发电储能装置分析.....	9
2.5.2 温度控制充电电压策略.....	10
3 光伏发电监控系统设计.....	11
3.1 光伏发电系统监控系统分析.....	11
3.2 测温装置设计.....	11
3.2.1 方案论证.....	11
3.2.2 原理图设计.....	12
3.3 电源电路设计.....	16
3.3.1 设计方案规划.....	16
3.3.2 变压设计.....	16
3.3.3 整流设计.....	17
3.3.4 滤波设计.....	18
3.3.5 稳压设计.....	19
3.4 直流电压电流检测电路设计.....	19
3.4.1 设计方案规划.....	19
3.4.2 模数转换模块设计.....	20

3.4.3 直流电流检测设计	20
3.4.4 直流电压检测设计	22
3.5 最大功率点跟踪电路设计	22
3.5.1 设计意义	22
3.5.2 设计方法分析	22
3.5.3 跟踪电路设计	23
3.6 太阳光照强度测量电路设计	25
3.6.1 方案论证	25
3.6.2 设计原理图	25
4 硬件制作及调试	26
4.1 硬件制作	26
4.2 硬件调试	26
5 软件程序设计	28
5.1 上位机设计可行性分析	28
5.2 初始化设计	28
5.2.1 数据初始化程序分析	28
5.2.2 串口初始化程序分析	28
5.2.3 显示窗口的初始化	29
5.2.4 建立数据日志设计	30
5.3 串口设置	30
5.4 温度、光强处理方案	31
5.5 控制下位机设计	34
5.6 界面设计	34
5.6.1 员工登录界面设计	34
5.6.2 采集界面设计	35
5.7 下位机程序设计	36
5.7.1 主程序设计	36
5.7.2 接收中断程序设计	37
6 结束语	40
致 谢	41
参考文献	42
附录一 硬件演示设备原理图	43
附录二 硬件演示设备 PCB 图	44
附录三 太阳能发电系统结构示意图	45
附录四 硬件演示设备源程序	46
附录五 元件清单	70

1 绪论

1.1 太阳能发电技术

1.1.1 发展背景

随着人类在科技领域的发展以及社会的进步，对新能源的要求逐日增加，如何选择新型能源，已经成为了人类所亟待解决的问题。在当前所使用的能源中，火电占了大部分比例，但伴随而来的是环境的污染以及将来可能面临资源枯竭的危险，而水电和核电也由于种种局限性，不能成为主要使用能源。而太阳能则是通过太阳的光照使光方阵进行光电变换来产生电能。这种方式与其他发电方式不一样的是，它具有取之不尽用之不竭、不产生任何污染物的特点，在地域上不受任何资源分布限制；在使用方面，可根据需求设立发电点；电能质量比较高；转换时间时间短。总而言之，太阳能具有极大优点，是未来能源结构调整和应对气候日益恶劣的最优能源。

目前，世界上很多经济发达的国家都对本国的光伏产业提出了发展规划。近邻日本，在 2004 年 6 月份有日本能源发展组织提出了《面向 2030 光伏路线图》即 PV2030 计划，这一规划的具体目标是：到达 2030 年后，安装的太阳能电池组件达到约 100GW，要使光伏技术成为日本最为关键的技术之一；在 2004 年 3 月份，欧盟的联合研究中心发布了《欧洲光伏研发路线图》，这一报道指出，预计到 2030 年，欧盟等国的通过太阳能发电产生的电能将占总电量的 10%以上，到 2040 年为止，这一指标将上升到 20%，到 21 世纪末，要突破 60%；在 2007 年的 6 月，美国的相关机构也提出了关于发展太阳能的《国家太阳能项目路线图》，针对美国国内几种重要太阳能电池，从价格、可靠度、工作效率、发电成本等方面分别提出了规划。

我国也于 2007 年由国家发改委提出了《可再生能源中长期发展规划》这一规划提出了未来光伏发电产业的中长期目标：预计到 2015 年累计安装量达到 250MW，到 2020 年安装要能够达到 1600MW，但实际安装情况已经远远超过预想计划，到 2015 年止，我国太阳能安装量已达 19GW。在 2012 年，国家工信部公布了《太阳能光伏产业“十二五”发展规划》这一报告，报告提出了对各大光伏企业的要求，要求到 2015 年止，多晶硅领先企业要达到 5 万吨级，骨干企业达到万吨级；太阳能电池企业安装量达到 5GW 级，整体成本下降约 35%。虽然我国光伏产业起步较晚，但基于政府的政策扶植和其他发展国家市场的带动，使得我国光伏生产能力也迅速增长，可以预期，21 世纪末期，太阳能将成为各国使用的主要能源。

1.1.2 技术原理及应用

太阳能光伏发电技术主要是利用太阳光照在半导体的 PN 结上，发生光生伏

打效应，从而产生了电压差。当前，典型的光伏发电方式一共有两种，一种是光能转换为电能，这一过程是直接转换的，利用的就是光生伏打效应，太阳能电池相当于一个半导体光电二极管，当光电二极管接受到照射时，光电二极管便将太阳能转换为电能，产生电流，很多种这样的电池通过串联或并联的方式连接在一起，就形成了太阳能发电方阵。

另外一种发电方式则是通过热能来发电，其原理是先将光能转化为热能，然后再由热能驱动蒸汽机产生蒸汽驱动汽轮机发电，这一过程与火力发电类似，但是效率不高且成本较高。

近年来，光伏系统的应用大体可分为离网系统与并网系统。离网系统就是我们所说的独立光伏系统，这一系统是完全依赖太阳能进行供电的，也是太阳能领域最主要的一种系统，在 2000 年之前，离网系统的安装量远远超过并网系统，当今在很多较为偏远的地方离网系统依旧发挥着不可取代的作用。

最简单的离网光伏系统可以看成光伏方阵与负载的直接连接，这中间并没有蓄电池等储能设备，所以这种直连系统只有在有光照的地方才能工作，框图如 1-1 所示。

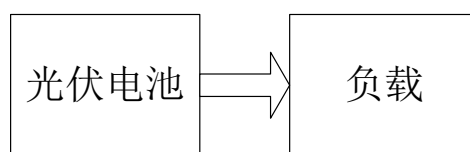
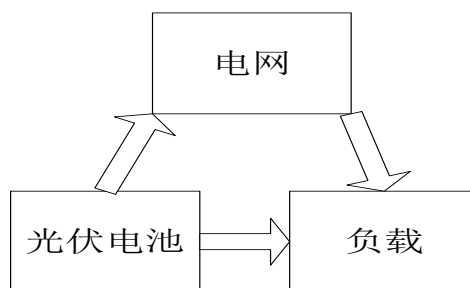


图 1-1 光伏离网发电

并网光伏系统最是将用户光伏系统与电网相连，在太阳光强度很大的时候，系统直接为负载提供电能，并且将多余的电能通过线缆传入公网，当无光照或者光照不足时则由电网向用户提供电能，这类光伏系统一般占地较小，可节约土地资源，同时也避免了远距离供电导致的电路损耗，其系统结构框图如图 1-2 所示。与离网式光伏发电系统相比，并网式系统是光伏系统的应用正式进入城市的能源体系的象征。



1-2 光伏并网发电

1.2 太阳能发电监控装置

太阳能发电系统中，转换率低一直是太阳能发展道路上的一块障碍，因此在

对太阳能应用的研究中，对提高转换率的研究一直占有主要地位。目前提高转换率的方式有以下三种：一是通过研究太阳能电池的工艺、材料来提高转换率；二是根据太阳的方位来调节太阳能板，使之能保持与照射光垂直，来获取最大的照射度；三则是通过采集、分析、控制的方式，使光伏系统始终处于能源最大利用率状态，这就需要开发良好的监控装置以达到对系统运行的实时控制。

目前，监控系统的实现方式主要有有线传输和无线传输，其中无线传输主要是通过无线电台或者无线通信网络来传输，这种方式抗干扰性较差；有线传输则是通过工业总线来实施现场数据传输及控制，这种方式抗干扰能力较无线通信强，传输速度快捷，设备简单，但是受到距离限制，较远地区则无法使用。本文采用 RS-232 总线来进行单片机模块与 PC 机间的数据通信。

1.3 本文主要研究内容

本文介绍了光伏发电系统的发展背景，对各个组成部分进行了分析，设计了一套独立光伏发电系统结构，对其中的采集控制系统进行了具体的研究设计，工作包括对工作电压、电流、光照强度、温度的采集以及相关原理图、PCB 图等。报告详细说明了制作温度、光强采集电路的过程及调试方法。对追踪最大功率输出问题和温度控制太阳能电池充放电问题提出了相关设计和策略。

2 光伏发电系统组成结构设计

2.1 光伏发电系统结构及原理

光伏发电系统主要由五部分组成：太阳能发电方阵、蓄电池、控制器、逆变器、交流直流负载。其中，太阳能电池组件和蓄电池起电源的作用，在光照强度大时，太阳能电池为双负载供电，并将多余电力转入蓄电池，而当光照强度低时，蓄电池转入电网为双负载供电；控制器和逆变器除了各自所具备的功能外都具有控制保护系统的作用，控制器主要是为了实现对系统的监控和控制，包括对工作电压电流、实时的光照强度等进行采集。逆变器的作用是将直流变交流，同时也起到对电路的保护作用；负载部分为系统的终端，包括直流负载和交流负载。

在独立光伏发电系统中，其工作方式一般有如下三种：

- （1）单电源双负载运行模式
- （2）双电源补给式供电模式
- （3）蓄电池充电模式

第一种模式中，有两种工作环境，即光照强度大时和无光照环境。当光照强度大时，太阳能电池为双负载供电同时将多余电量充入蓄电池；而在无光照情况下，光伏方阵不工作，只有蓄电池工作，为双负载提供电力。

第二种模式是在光照强度不强时所选用的一种工作方式，此时太阳能电池产生能量低，不能够满足负载，所以需要和储能装置一起工作为负载供电。

第三种模式是在光照不足且蓄电池没电时所启用的一种工作方式，此时需要关闭负载，先为蓄电池充电，待到蓄电池充满后在启用第二种工作模式。系统总结构如图 2-1 所示。

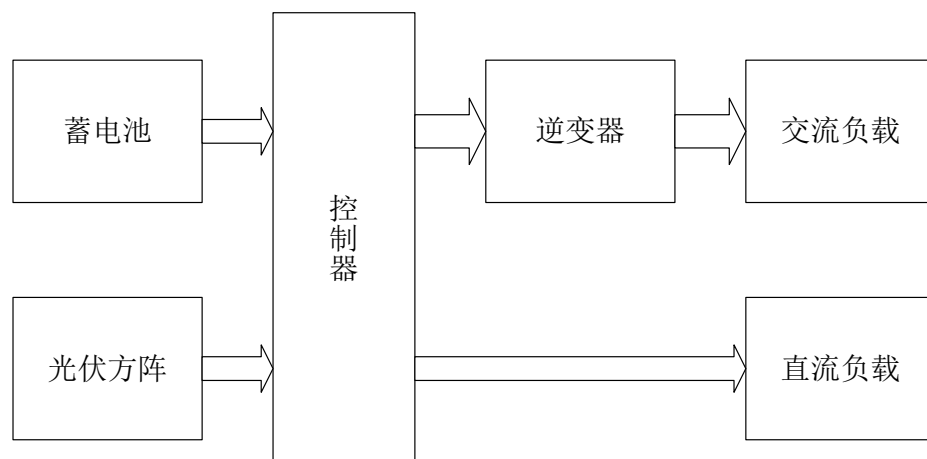


图 2-1 独立光伏系统总结构图

2.2 电池单元

2.2.1 太阳能电池的分类

太阳能电池的作用是将太阳能转化为电能，目前，它已经在人们生活的各个领域广泛使用，所以，从不同的角度出发，可以分为较多类别。

1. 按制作材料分类

从制作材料的角度出发，可以把太阳能电池分为由硅材料制作以及由化合物制作的太阳能电池，其中，硅太阳能电池是我们较为常用的，由于原料在大自然中含量很高，且具有重量轻、高硬度的特点，同时对环境几乎没有任何影响，所以较化合物太阳能电池具有广泛的使用价值。

单晶硅太阳能电池是发展较为早的一种太阳能电池，技术相比其他电池成熟，性能也较为稳定，但造价比较高，年产量已经被低成本的多晶硅太阳能电池超越，但依旧占有很高的市场比重。

双晶硅太阳能电池是与单晶硅电池制作工艺不同的一种低成本的电池，它制作时先将高纯硅熔成块型硅锭，再使用切割机将其切片，制成电池，这种电池转化率较低，一般在 17%-19%之间，但由于其低成本的优点已使双晶硅电池成为市场中所占比重较高的电池。

非晶硅太阳能电池相比单晶硅和双晶硅电池，它具有很薄的厚度，切割起来容易，制作流程简单，大大节省了硅材料的供应，降低了制作成本，但它的能源转化率较低，仅有约 6%-10%，同时由于非晶硅的材料硬度和熔点都不高，所以在特定环境使用会受到限制。目前这种太阳能电池还未大面积推广使用。

微晶硅太阳能电池的转化效率较非晶硅电池高，且不具有光致衰退效应，但目前还不具有大规模生产的条件。

化合物太阳能电池主要利用的是半导体材料，常用的化合物光伏电池有砷化镓电池、碲化镉电池、铜铟锡电池等。

2. 按用途分类

从使用用途方面考虑，可分为两类，一类是地面太阳能电池，这种光伏电池一般用于地面，使用较为广泛，应用于生活的很多方面，其性价比一般较高。另一类是空间太阳能电池。这种太阳能电池主要用于航天器、太空站、太空机器人等，这些太阳能电池往往要具备重量轻、抗高温、抗冲击、不具有光致衰弱等特点，常用的有单晶硅、砷化镓光伏电池。

2.2.2 太阳能电池的工作原理

太阳能电池的作用是将自然界中的太阳能转化为供我们使用的电能，这一过程是通过半导体材料来实现的。当太阳照射半导体材料的表面时，如果太阳光所具有的光子能量大于半导体材料的禁带宽度，那么此时半导体中的电子就会脱离原子核，在 PN 结中形成很多电子-空穴，在内电场的作用下，电子空穴分开形成电位差，这一原理就是光生伏打效应。

2.2.3 影响转换率的因素

影响太阳能转化效率的原因主要有如下 8 种：

（1）温度：太阳能电池中，转换效率受温度的影响很大，在硅电池中，温度每上升一度，转换率大约下降 0.4%，所以保持工作温度在一定范围里有助于稳定系统的太阳能转换效率。

（2）禁带宽度：禁带宽度越大，需要的光子能量越高，反之则需要的光子能量越少，转换率越高。

（3）光照强度：在特定环境中，光照强度越大，单位太阳能板表面将产生更多电子-空穴，使得产生的压强差更大，因此转换率也随之增加。

（4）金属栅线：在太阳能板表面，排列着很多条不透光的栅线，减少这些栅线所占用的面积，可以很好地增大转换率。。

（5）表面光反射：硅片表面存在着光反射，降低这部分反射光将使太阳能电池接受更多的照射，因此可以增加有效的涂层，是电池表面的反射率降低，以增大转换率。

2.3 控制器单元

光伏发电系统中，控制器的主要作用是对太阳能系统进行监控和控制，通过采集系统在工作时的各种参数，了解工作状态，并发出控制指令，同时控制器也兼备了设备保护的功能，其中包括了负载短路保护、反向放电保护、雷电保护等。对于不同的太阳能系统，所需要的功能不同，控制系统也大不相同，大体可分为以下五种：

- （1） 并联型控制器
- （2） 串联型控制器
- （3） 制型控制器
- （4） 智能型控制器
- （5） 最大功率跟踪型控制器

并联型控制器是利用电子元件把太阳能方阵所输出的能量分流到内部并联的电阻或功率器上，然后以热的形式消耗掉，这种控制器由于要耗热，所以一般只用于小功率系统；串联型控制器是利用机械继电器来控制充放电过程，用于较高功率的系统；脉宽

调制型控制器是指以 PWM 脉冲方式来控制光伏方阵的运行；智能型控制器主要是由带 CPU 的单片机组成的，通过对光伏系统的实时工作状态进行采集和分析，并可以配合单片机的串口通信对系统进行远程控制；最大功率跟踪控制器是一种通过判断光伏方阵的工作点是否为最大输出功率点，来对光伏方阵工作点进行改变的一种控制器，这一控制器的优点是可以让光伏矩阵始终以最大功率输出，充分利用了其输出能量，增加了光伏发电的转换率。

本文设计的是一套智能型控制器，通过使用可编程的单片机对系统工作参数进行采集、显示，通过程序对系统进行接通、断开等控制，也可以通过单片机的串口通信进行人机互动。

2.4 逆变器单元

2.4.1 逆变器的分类

在光伏发电系统中，逆变器所起到的作用是将太阳能电池产生的直流电转换为交流电供交流负载所使用，这一过程是整流电路的逆过程。

逆变器的种类不同，其分类也较繁多，大体可分为如下 5 种：

（1）判断逆变器的输入输出是否隔离，如果隔离则为隔离型逆变器；如果没有隔离，则为非隔离型逆变器。

（2）根据输出的波形来判断，可分为：方波逆变器、正弦波逆变器以及阶梯波逆变器。

（3）按照其输出频率不同，可以分为三种，即工频逆变器、高频逆变器、中频逆变器。

（4）按照输出是否与公网相连有并网型逆变器和离网型逆变器之分。

（5）根据结构不同有全桥式、半桥式、推挽式逆变器三种。

在以上分类中，我们主要是按照输出是否接电网来进行分类的，即通常分为离网逆变器和并网逆变器，两者的硬件结构区别不大，基本都是由逆变桥、控制电路和滤波电路构成，与离网逆变器相比，并网型有较高的输出要求，它的输出与公网相连，必须满足电网的相关规定，以防对公网造成影响及谐波污染等问题。它的功能是将电能转换为适合电网使用的一种电能，所以也被称为功率调节单元。

2.4.2 离网型光伏逆变器

离网型光伏逆变器是指光伏系统与公网断开，独立运行，其主要应用于较为偏远的地区，如边防哨岗、海岛等远离公网的特殊地区。离网光伏系统通常是由光伏方阵、控制器、逆变器和储能装置等组成的，当有太阳光照时，光伏系统直接将光能转化为电能为负载供电，同时为储能装置充电；而在无光照时，则通过储能装置放电为负载供电。

在整个离网系统中，逆变器是核心，常用的离网逆变器主要有三种拓扑结构：

（1）工频隔离单级逆变器：由于常用的负载需要使用 110V 或者 220V 的交流电，而光伏方阵输出的为直流电，且电压很低，所以就需要使用工频隔离单级逆变器将得到的低压交流电转化为高压交流电。

（2）高频隔离两级逆变器：这种逆变器是先将低压直流进行高频逆变，然后再通过高压逆变转化为高压高频交流电，然后进行整流得到高压直流电，最后通过工频逆变便可得到高压工频交流电供负载使用。这种逆变器具有体积小的优点但是结构较为复杂。

（3）无隔离两级逆变器：这种逆变器是直接将低压直流电通过升压环节升为高压直流电，逆变后再进行滤波得到负载所需电压。

2.4.3 并网型光伏逆变器

并网光伏逆变器是将光伏方阵产生的电能转化为交流电并传输到公共电网上，这种逆变器要保证输出的电流与电网的电压频率以及相位完全一样。其分类可分为四类，按照拓扑结构来分有全桥逆变拓扑、半桥逆变拓扑、多电平逆变拓扑、推挽逆变拓扑、正激逆变拓扑、反激逆变拓扑；按照输出相数可分为三相和单向逆变器，按照输出功率等级有大功率并网逆变器（大于 50KW）、中等功率并网逆变器（大于 1KW, 小于 50KW）、小功率并网逆变器（小于 1KW）；按照隔离方式可分为隔离式并网逆变器、非隔离式并网逆变器。

纵观光伏并网型逆变器的发展历程，其发展趋势越来越高频化、小型化以及智能化。相信在未来最具有研究及应用前景的当属于小功率型逆变器，因为这种逆变器同时其具了小型、高效、集成、灵活等特点，据报道，采用小功率逆变器技术可以再现有的技术上，将光伏系统发电效率提高百分之五到百分之二十五左右。

2.4.4 光伏逆变器性能参数

应用于光伏发电系统中的逆变器一般有一下几个参数要求：

（1）一般单向额定输出电压为 220 伏，三相额定输出电压为 380 伏，稳定状态时电压波动不能超过 3%-5%，发生负载突变时，电压波动不能超过 8%-10%。

（2）正常工作时，输出的三相电压不平衡度不能超过 8%；波形失真度不能超过 5%。

（3）正常工作时，输出额定频率偏差不应该超过 1%，国标规定输出频率在 49 到 51 之间。

（4）额定输出效率一般应在 70%以上（额定输出效率=输出功率/输入功率）

在逆变器的性能中，其转换率为最为主要的性能参数之一，转换率为输出端交流功率与输出直流功率的比值：

$$\eta = \frac{P_{AC}}{P_{DC}} = \frac{P_{DC} - P_L}{P_{DC}} \quad (2-1)$$

式中 P_{AC} ——为输出交流电的功率；
 P_{DC} ——为输出直流电的功率；
 P_L ——为转化过程中的功率损耗。

2.5 蓄电池储能单元

2.5.1 光伏发电储能装置分析

太阳能电池的工作主要依赖于光照，当光照强时，电池将光能转化为电能供负载运行，而当光照强度低时，想要负载能够稳定运行，就需要用蓄电池提供电能。在并网和离网系统中，一般都配置有蓄电池，其中，离网系统必须配有储能设备，因为当夜晚无光照时，太阳能电池不能产生电力，负载就无法正常工作；而并网系统中，由于可以使用来自电网的电力，则无需担心无光照时负载无法正常运行，如果负载中有重要负载，无法中断，则也需要配备储能装置。

目前普遍使用的蓄电池包括铅酸蓄电池和镍镉蓄电池。其中铅酸蓄电池的主要成分是海绵状铅体、稀硫酸、二氧化铅，这种电池价格低廉、原料丰富，是目前使用最为广泛的一种蓄电池。相比铅酸蓄电池，镍镉蓄电池的密封性较好，使用的寿命较长，所存储的电量高，但由于具有“记忆效应”，并且其原料金属镉有毒，会对环境造成很大的损伤，同时造价也较铅酸电池高，所以这种电池正在面临着逐步被淘汰。

表一 蓄电池的技术参数表

技术参数	等级
电压	3 伏、6 伏、12 伏、24 伏、36 伏等
容量	4 安时、6 安时、12 安时、24 安时等

这里设充电时间为 t ，电池容量为 C ，电池工作电压为 U ，充电电压如式 2-1 所示：

$$t = \frac{C}{U} \quad (2-2)$$

当光伏系统为蓄电池充电时，光伏方阵的输出电压要超过蓄电池工作电压的 20%-30%。为了防止发生蓄电池的反向放电，最好在充电系统中加装一只整流二极管。若能假设充放电自动保护系统更好，即可以保护过充或者过放，又可以延长电池使用寿命，增加经济性。

在为独立光伏系统选择适配储能电池时，要经过多方面的考量，如电压电流特性、成本、质量、安全性。维护难度、大小尺寸等，在此基础上再考虑经济性，其设计步骤可以简单分为三步：

（1）首先对负载所需要的输入电压和输出电压进行计算，确定电池组件数；然后查询安装场所的太阳照射数据，记录没有太阳照射时间。

（2）设定放电深度。

（3）计算蓄电池容量，公式为：

$$C = \frac{\text{一天所消耗的电能} \times \text{无日照天数}}{\text{放电深度} \times \text{放电终止电压} \times \text{衰减率}} \quad (2-3)$$

2.5.2 温度控制充电电压策略

光伏发电系统中，蓄电池作为后备能源可以解决无光照时系统电能供应限制，但电池容量和使用寿命一直是亟待解决的问题，目前在对蓄电池的充电过程中，存在着随着充电时间的增加，温度会越来越高，导致使用寿命的降低，同时效率也会越来越低，据了解，大部分蓄电池的高效充电温度一般都集中在 10-40 摄氏度之间，小于或者大于这个温度，都将降低充电效率，所以本课题针对此问题，提出了两种可行性的解决策略。

（1）智能充电控制：这种方法是在充电电路中加入 boost 调压电路，控制器在对蓄电池的温度采集后，根据充电最大效率曲线，通过改变调压电路的占空比，使充电电压处于最佳充电状态。

（2）温限法：设置蓄电池的最高充电温度，高于温限时，停止充电，降于温限后，继续充电，这个方法较智能充电控制法简单，但极易受到环境温度因素影响，误差较大。

3 光伏发电监控系统设计

3.1 光伏发电系统监控系统分析

光伏发电的监控系统想要完成其职能，就要对三部分进行设计。第一部分用于对太阳能板的现场工作温度、电压电流、光照强度进行采集，然后将采集到的数据传输到第二部分微机控制单元（MCU），通过 LCD 模块将数据能在现场显示出来，并通过 MCU 的串口通信将数据上传至第三部分 PC 机进行显示、分析以及储存。

本设计 MCU 选用的是 Atmel 公司制造的 AT89s52 芯片，串口通信选用 MAX232，上位机使用 VB 编写。系统结构如 3-1 所示。

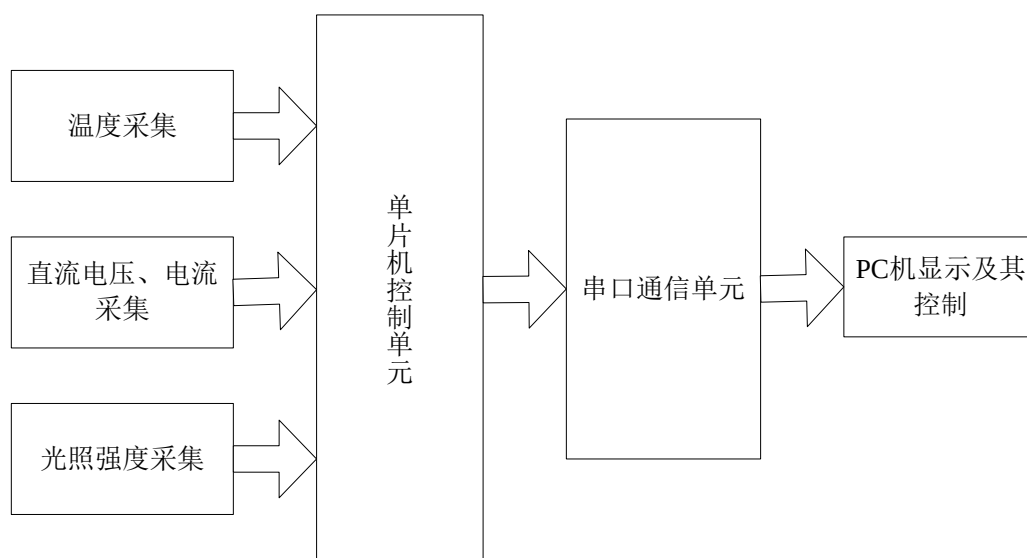


图 3-1 监控系统数据采集框图

3.2 测温装置设计

3.2.1 方案论证

太阳能电池中，温度很大程度上会对太阳能系统造成影响，随着温度的上升，太阳能电板的开路电压降低，短路电流升高，所以保持工作温度在一定范围里有助于系统工作状态和转换效率的稳定。通过查阅相关资料，提出了以下两种可行的方案，方案一：可采用热敏电阻对温度进行测量，从测温范围考虑，常用热敏电阻可满足太阳能电池工作温度-20 摄氏度到+50 摄氏度的测量，但从测量精度以及可靠性来考虑，其精度和可靠性均较差，且元件非线性严重。在整个温度测量系统中，常用的热敏电阻，很多输出的都是都是模拟信号，必须要经过模数转换后才能送到控制系统，这样会使结构复杂化。第二个方案，则是考虑使用温度传感器，在单片机电路的设计中，使用传感器可以使结构变得直接化，所以在这里我们可以采用一只数字温度传感器，这种传感器输出为数字

信号，单片机可以很容易读取被测量的温度值。通过对上述两种方案的对比可以容易看出，方案二设计为优选，其结构简单，软件设计也不复杂，所以本次设计我们选用方案二。

设计选用 ds18b20，它抗干扰性能强，体积小、精度高、造价也相对低。在传输数据方面，只要一个引脚便可和微机控制器进行双向通讯，其输出为数字信号，可直接供单片机接收；它的测温范围为-5-125 度，可满足光伏发电系统的工作环境测温，固有误差为 1 摄氏度；电源与 89s52 单片机一样都可用 5v 供电，接线较为方便。其引脚如图 3-2 所示。在实际的连接中，一般是将 ds18b20 的信号线，即 2 口与单片机的一位 I/O 口相连，电路连接如图 3-3 所示，本设计中将信号线与 at89s52 的 P3.6 口相连，并接 10k 欧的上拉电阻，VCC 接 5V 外接电源，GND 端接地。

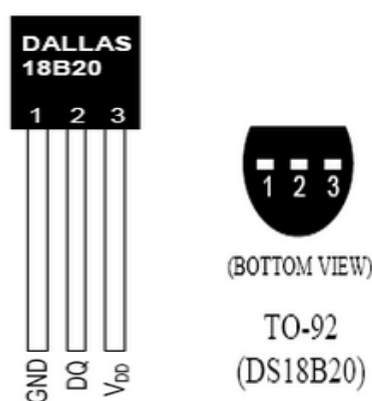


图 3-2 ds18b20 引脚图

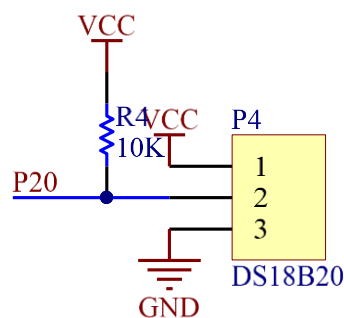


图 3-3 ds18b20 连接图

3.2.2 原理图设计

(1) 单片机最小系统设计

单片机最小系统由以下几部分构成，设计原理图如图 3-4 所示，第一部分就是微机控制器，即单片机本身，它相当于一块微机处理器，用来烧写程序并实现程序功能；第二部分是复位电路，可对单片机进行复位；第三部分是外部晶振电路，提供单片机所需要的时钟源。

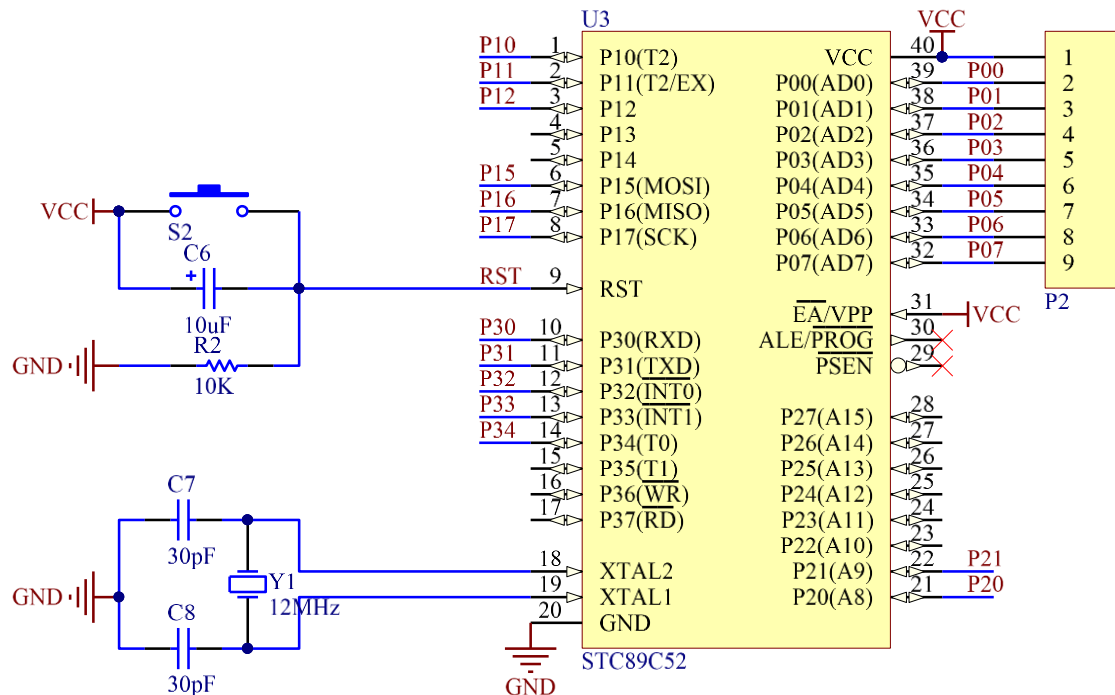


图 3-4 单片机最小系统

(2) 显示模块

在显示模块的选择中，我们提供了两个方案，方案一：用数码管显示数据；方案二：用 LCD 液晶显示屏进行显示，其对比如表三所示，明显的发现不管是在显示、能耗还是接线方面，LCD 液晶都优于数码管显示，同时考虑到设计要求对太阳能工作温度以及光照强度进行显示，其显示字符较多，单一的数码管显示并不能满足这一条件，所以在本设计中我们选用 LCD1602 液晶屏进行数据显示。

表二 显示模块方案对比表

方案对比	显示	能耗	亮度	接线
LCD 液晶屏	多样	低	低	方便
数码管显示	单一	较高	较高	复杂

LCD1602 属于工业字符型液晶，具备了低压以及低能耗的特点，2-3V 的电压就能使这种液晶屏工作，能够显示数字、字母以及文字等，其优点远远多于传统的数码管 LED。在小型仪表和低能耗产品中有广泛的应用。

表三 lcd1602 引脚介绍

1 号引脚	接电源地
2 号引脚	接+5V 电源
3 号引脚	对屏幕的对比度度调节端，一般外接一个 10 千欧的电位器进行调节
4 号引脚	寄存器选择端
5 号引脚	信号读写端
6 号引脚	使能端
7 号-14 号引脚	LCD1602 数据端
15 号-16 号引脚	背光灯电源脚

这种液晶常用于小仪表的显示模块中，因为它具有小体积、低功耗的特点，其电路如图 3-5 所示。

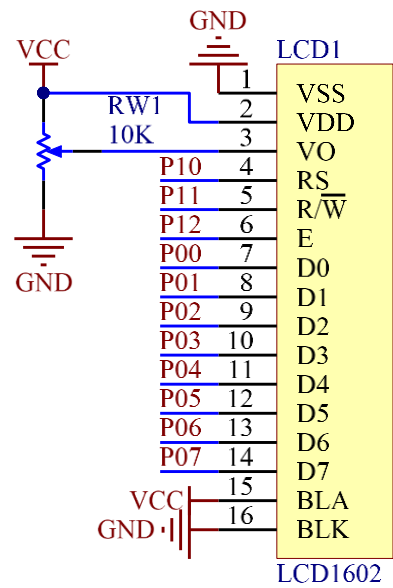


图 3-5 LCD1602 电路图

(3) 串口通信模块设计

AT89s52 的串行口是一个全双工通信接口，可以同时进行数据的发送与接收。在与上位机进行通信时，涉及到电平转换问题，这里选择 max232，具有低能耗、体积小等特点。其原理图如图 3-6 所示。

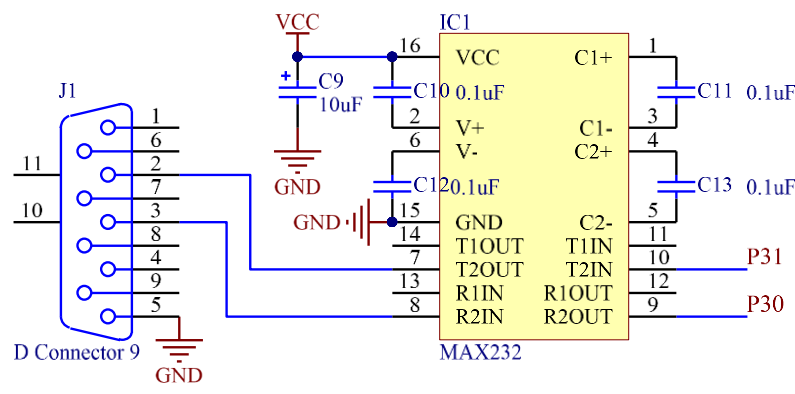


图 3-6 max232 通信模块

（4）蜂鸣器报警设计

目前常见的蜂鸣器可以按两种方式来分类，其一是按照驱动方式分类如表五所示：

表四 蜂鸣器驱动方式对比	
分类	区别
有源蜂鸣器	内部含有驱动，所以上电后可以直接工作，由于含有一个振荡电路，所以造价较高
有源蜂鸣器	内部没有驱动，需要外界提供震荡电路，造价较低

第二种是按构造方式来分类的，如表六所示：

表五 蜂鸣器构造对比	
分类	组成
电磁式蜂鸣器	振荡器、电磁线圈、振动膜片、磁铁及外壳
压电式蜂鸣器	多谐振荡器、压电蜂鸣片、阻抗匹配器、共鸣箱、外壳

在本设计中，出于电路简洁、减少程序复杂度以及经济性等角度的考虑，设计选用市面上常见的有源电磁式蜂鸣器。其接线原理图如图 3-7 所示。

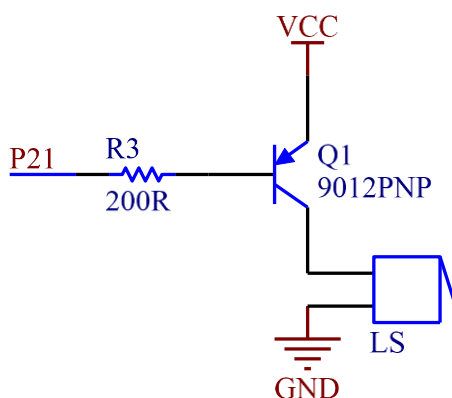


图 3-7 蜂鸣器模块

3.3 电源电路设计

3.3.1 设计方案规划

电源电路的稳定性对于整个电路的稳定起着至关重要的作用，常用的电源有直流电源和交流电源，最为简单的直流供电方法就是使用蓄电池，但这种方法所用的成本较高、充能周期性过短，而使用交流电，则很大程度上避免了直流电的缺点，通过简单的整流电路就可以供直流负载使用，经济、方便又可靠。

电源部分设计框图如 3-8 所示，考虑到本课题中硬件电路的电源要求统一是 5V 直流电，所以设计电源模块的目的是要把 220V 的工频交流电转换为电子电路能够使用的 5 伏直流电。

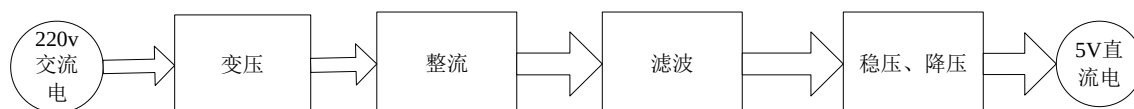


图 3-8 电源设计框图

首先把 220 V 工频交流电通过降压环节变成低压交流电，为了达到去交流的目的再通入整流电路，得到带脉动直流电，最后通过滤波去交流脉动，考虑到电压的不稳定波动，最好再增加一个稳压环节。

3.3.2 变压设计

本设计中由于所有设备供电都是基于 5V 直流电，所有在变压环节，我们选用以 12V 电压输出的小功率变压器。

3.3.3 整流设计

整流电路主要利用的是半导体二极管的单向导电性，其分类如下表七所示：

表六 整流电路分类表

分类	组成	原理	结构图
半波整流	一般只有一个二极管	交流电正向通过，反向截止，形成只有正向的脉动直流电	如图 3-9 (a) 所示
全波整流	用到两个二极管	利用变压器抽头使交流电正负半周皆为负载提供同向电流	如图 3-9 (b) 所示
全波桥式整流流	四个二极管组成	利用四个二极管的导通截止使交流电的正负半周为负载提供同向电流	如图 3-9 (c) 所示

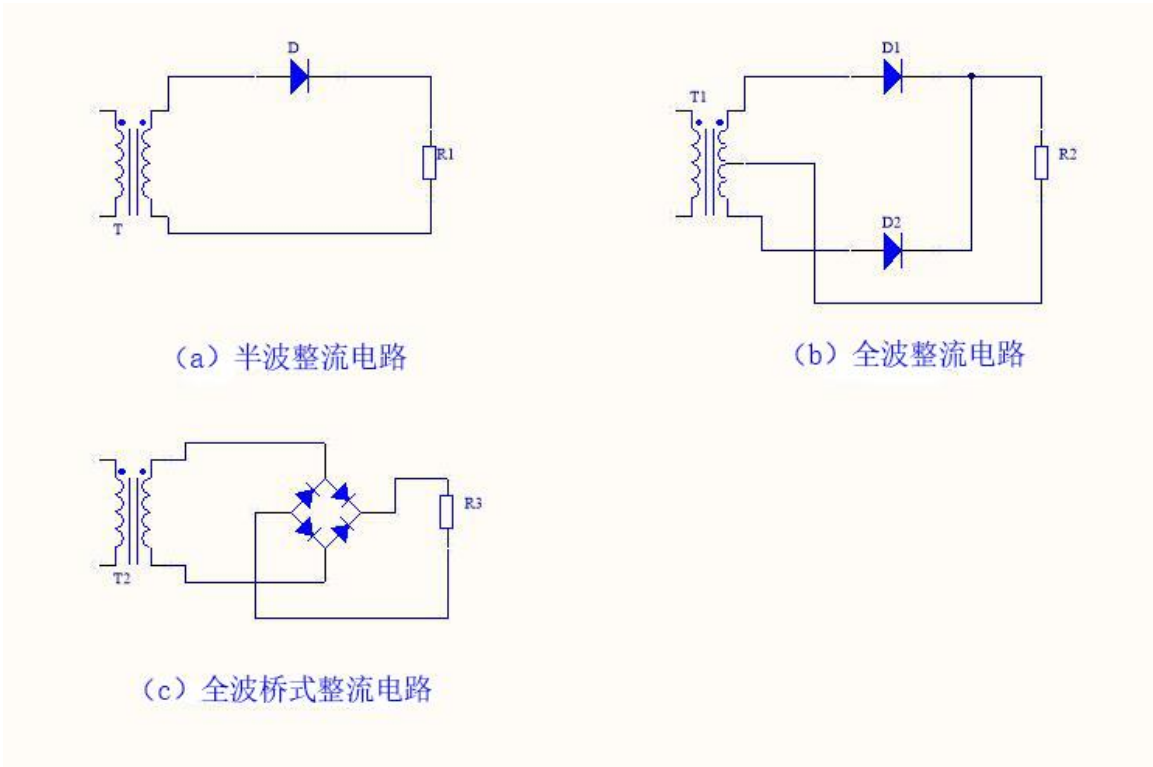


图 3-9 整流电路

在本设计中，我们选用全波桥式整流，这种电路使用了四支整流元件，输出电压脉动与全波一致，半波整流电路虽然电路简单，但是整流后电压脉动较大，所以需要大的滤波电容，对电源电压器的利用率较低，一般只用于对电源要求不是很高的电路。

3.3.4 滤波设计

常用的滤波电路一般有电容滤波、电感滤波、LC 滤波和 RC 滤波电路四种。

（1）电容滤波：结构图如图 3-10 所示，这种滤波是直接将电容器与负载并联，当正半周时电容被充电，负半周时电容放电，这样就可以使负载得到平滑的直流电。

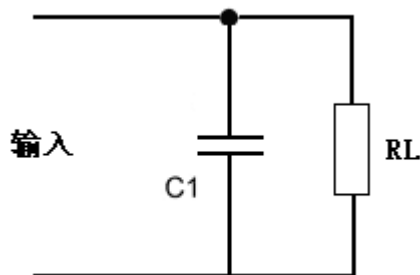


图 3-10 电容滤波

（2）电感滤波：将负载与电感串联起来，当电压增大时，流过电感的电流增大，电感将储存部分能量，当电压减小时，电感电流降低，电感释放之前储存的能量，这样便使负载得到较为平滑的直流电。

（3）LC 滤波：这一滤波电路是将电感与负载串联，电容与负载并联，通过电感和电容在升压与降压之间起到充能与释放作用，这样便使得电压输出曲线处于平滑。滤波电路的基本形式如图 3-11 所示：

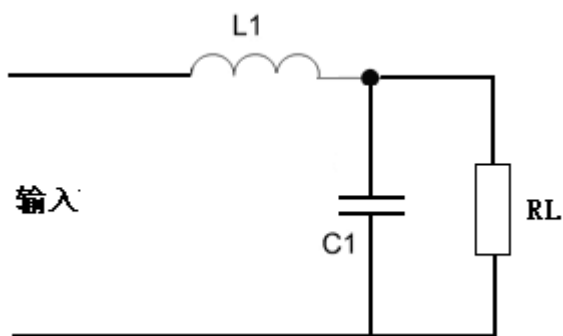


图 3-11 LC 滤波电路

（4）RC 滤波：这种滤波电路一般分为 L 型和 π 型，L 型是将电阻与负载串联，再并一个电容，而 π 型则是在串联的电阻两端各并了一个电容，其结构图如图 3-12 所示，与输入端相连的电容 C1 是消除大部分电压脉动，再通过电阻 R 的降压来减少残余脉动，最后由 C2 再旁路掉。

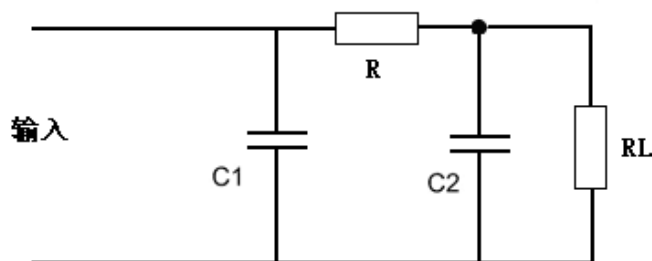


图 3-12 RC 滤波电路

3.3.5 稳压设计

由于交流电网的波动和负载电流的变化都会使整流电路的输出电压发生波动，所以对于电路要求较高的则有必要使用稳压电路。本设计中，为了电路的方便简洁，我们选用集成化稳压电路，由于整流电路输出的电压为 12V，所以这里我们使用 LM7812 三端稳压管来稳定整流出来电压，再通过一个 LM7805 将稳定后的电压降为我们电路所使用的 5V 电压。其设计图如 3-13 所示，到此为止，在经过了变压、滤波、整流、稳压等处理后，便达到了课题硬件电路的用电要求。

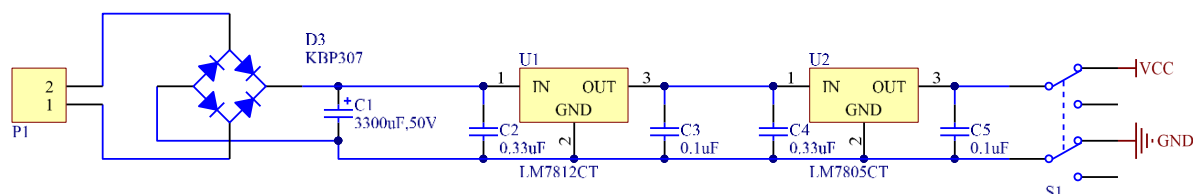


图 3-13 电源原理图

3.4 直流电压电流检测电路设计

3.4.1 设计方案规划

本设计主要由三大模块组成：模数转换模块、MCU 模块、LCD 显示模块。作为核心的 MCU 模块采用 AT89C52 单片机，模数转换模块采用 PCF8591 芯片；显示模块采用 LCD1602 液晶来显示直流电压及电流值。本模块设计框图如图 3-14 所示。

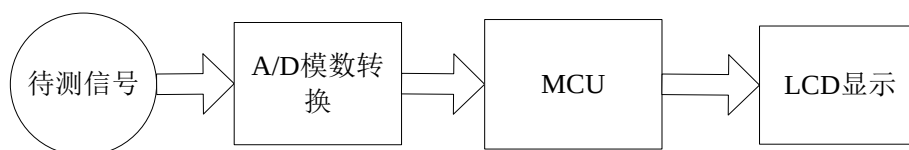


图 3-14 电压电流测量框图

3.4.2 模数转换模块设计

模数转换芯片选用常见的 PCF8591，PCF8591 是一个具有 I²C 总线结构的多通道 8 位逐次逼近型 A/D 转换器和一个内置 8 位单通道 D/A 转换器，它具有 4 个模拟输入、1 个模拟输出和 1 个串行 I²C 总线接口，并且具有单独供电、低能耗的优点。

综合上述几种 A/D 转换芯片的特点，前两种芯片的性能和精度都不如第四种芯片。在本设计中，我们的目标是设计精度 1%以内的高精度电压电流测量，因此在此，我们选择 PCF8591 芯片。

表七 PCF8691 芯片引脚说明

引脚号	名称	说明
1-4	AIN0-AIN4	模拟信号输入端。
5-7	A0-A2	引脚地址端。
8、16	V _{SS} /V _{DD}	电源端。（2.5-6V）
9	SDA	总线的数据线
10	SCL	总线的时钟线
11	OSC	输入/输出时钟端
12	EXT	内、外部时钟选择线
13	AGND	模拟信号地
15	AOUT	D/A 转换输出端
14	V _{ref}	基准电源端

3.4.3 直流电流检测设计

（1）设计规划

电流是电学中最为基础的物理量，可直接反映出负载以及电路的工作状态，对于不同的场合，其电流的检测要求也不同，本设计要求设计对 0-100A 直流电流进行检测，可从检测的响应速度、功耗、体积、发热温度等方面考虑。对于直流电流的测量，目前常用的测量方法一般有两种：直接测量和非直接测量。直接测量方式一般是通过串联电阻来进行而非直接测量则是通过测量电流流经时所产生的磁场大小来进行测量的一种方法。

（2）电阻检测法

电阻检测法是一种直接测量的方法，一般是通过测量串联于电路的电阻电压，然后通过欧姆定律来对电路的电流进行计算，这种方法所测量的电流不太大，而且最终输出

的是模拟信号且较为微弱，所以要在输出口再接一个信号放大器，然后通过模数转换电路，将模拟信号转换为数字信号。这种方法比较简单且成本较低，但由于电阻的特性受环境的影响较大，会存在温度漂移的现象，而且这种方法一般只是用于电流较小的电路中，如果电流过大，其发热状况以及电路损耗会比较大，所以这个方法不适用于本设计的要求来测量 0-100A 电流。

（3）霍尔电流传感器检测法

霍尔电流传感器主要利用的是利用霍尔效应，在进行测量时，只需将传感器外接正负直流电源，把被测电流导线从传感器的测量孔中穿过，这样便完成了隔离测量，整个电路较为简洁，但是其输出为模拟信号，这就要求在后续电路中加入模数转换电路。基于霍尔传感器所具备的精度高、线性好、频带宽、响应快、过载能力强等优点，所以本设计中我们就选用这种方法来对电流进行测量。

（4）霍尔电流传感器的选择

在对霍尔电流传感器的选择中，我们主要是从发热温度、寿命、过载能力、测量响应速度、功耗、体积、价格等来进行考虑，这里我们选择 HCS-HY-50 系列的闭环磁平衡电流传感器，其实物图如图 3-15 所示，这种传感器具有高精度、低能耗、体积小、强过载能力以及价格较低等特点，测量范围可达 150A，所以可满足设计的要求，原理图如 3-16。



图 3-15 HCS-HY-50

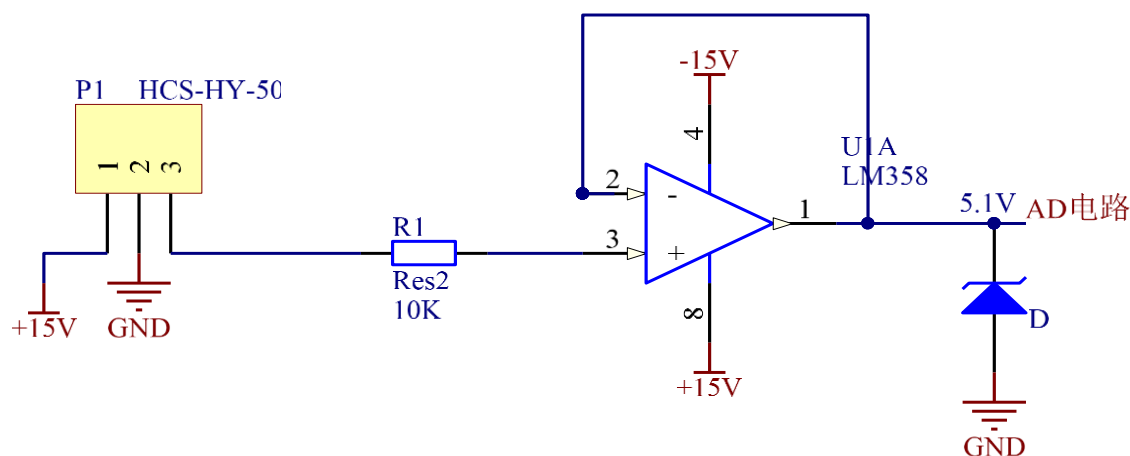


图 3-16 电流检测原理图

3.4.4 直流电压检测设计

（1）设计规划

对于 0-48V 直流电压的测量，本设计选用分压式测量，采用一个前置分压电路加一个电压跟随电路。由于太阳能方阵输出电压随环境不断变化，所以在这里设计一个电压跟随电路相当于加了一个缓冲器，因为其输入电阻相当于无穷大，输入阻抗相当于零，对电路起到很好的缓冲作用。

（2）原理图

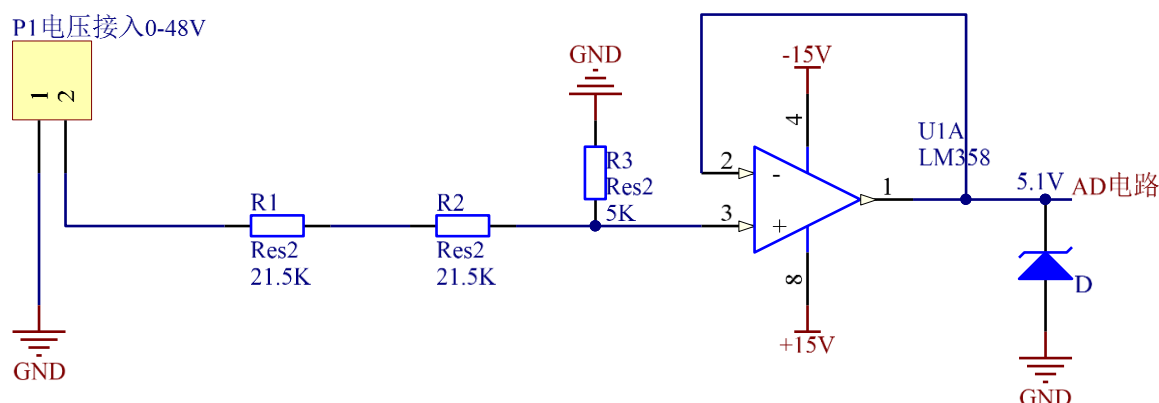


图 3-17 电压检测原理图

3.5 最大功率点跟踪电路设计

3.5.1 设计意义

太阳能电池在工作时受到环境的影响很大，环境中温度或者太阳辐射度的改变，都将会导致系统的输出特性发生变化，这就使得太阳能系统输出值处于一种时刻不定的变化中，而最大功率跟踪技术，则是根据这种环境的变化，通过控制输出电压或者电流使输出功率始终位于最大输出功率点上，保证了系统在环境变化时依旧能够有效的输出。

3.5.2 设计方法分析

目前，实现太阳能最大功率点跟踪的方法有很多种，但大多都是依据电压或依据功率来调节的。依据输出电压是指在系统工作时，通过测量当前环境下的输出特性曲线而获取最大功率点，再通过调节工作电压，使之尽可能接近最大输出功率点电压，通过不断的计算对比，可以实现对最大功率点的电压跟踪。但由于计算耗费的时间较多，面临环境的迅速改变时，这种方法不能迅速的对最大功率点进行跟踪。

依据功率是指依据蓄电池的充电功率，如果能够保证蓄电池最大功率输入，则表明此时太阳能电池的输出功率也将为最大。

在常见的最大功率点跟踪方法中，应用较为广泛的是恒定电压法、爬山法和电导增量法。所谓恒定电压法，是指使太阳能电池的输出电压等于现场环境的最大输出功率点电压，这样就保证了在该温度下系统处于最大功率输出。这种方法控制简单但误差较大，控制精度不高，没有考虑温度对系统的影响，由于环境温度时刻变动，最大功率点的输出电压也将发生变化，对于那些日温差变化较大的地区，这一方法显然不能有效的追踪最大功率点，如果仍然想要使用该方法，则可通过手工调节不同时段或者不同季度所对应的最大功率点电压。

爬山法又叫扰动法，是指借助对光伏发电系统的工作点电压进行扰动，判断扰动所导致的输出功率变化，但这种方法只适用于温度变化缓慢的情况，当温度持续变化且较大时，会出现误判现象，导致判别最大功率点错误。

爬山法不能判断出最大功率点电压与工作电压之间的关系，它只是利用对测量前后两次功率的对比而对最大功率点进行模糊的定位，电导增量法则弥补了这一缺陷，避免了爬山法的误判现象。这一方法是通过对比太阳能电池的导抗与导抗变化量来实现对最大功率点的跟踪，当系统处于最大功率工作点时，此时功率对电压的导数为 0，如果这一导数小于 0 时，需要减小工作电压；当这一导数大于 0 时，需要增大工作电压，经过多次调节，最后稳定在导数为 0 的最大功率点上。这一方法可以在环境变化大时迅速的追踪到最大功率点，但是电导增量法对仪器精度要求较高，经济性也不如爬山法。

3.5.3 跟踪电路设计

本设计是在负载与太阳能电池之间加入一个 DC/DC 调节器，通过改变调节器中导通管的导通率来调节电路，使电路时刻处于最大功率输出点，这样便实现了太阳能发电系统的最大功率点跟踪。设计控制框图如图 3-4-1 所示，光伏方阵通过 DC/DC 调节器为负载供能，控制器对光伏方阵的工作电压电流进行采集，将采集到的数据与最大功率点数据进行对比，如果没到到达最大输出功率状态，则控制器通过调节控制信号的占空比来对方阵的工作点进行调节。

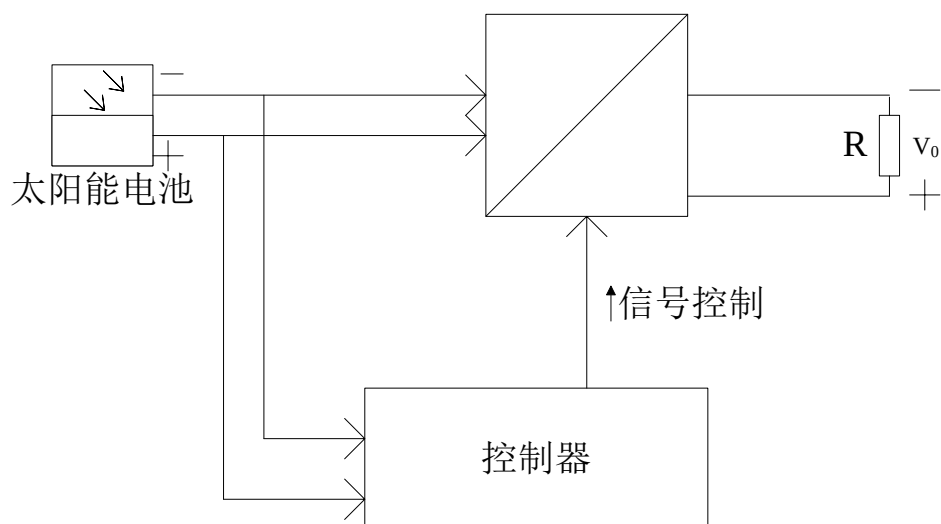


图 3-18 最大功率设计框图

这里我们采用 Boost 升压式电路，电路主要是由负载 R 、二极管 D 、开关管 T 、电感 L 、电容 C 组成，工作原理为：当 T 导通时， D 发生反偏，太阳能电池为电感 L 充电，当 T 关断后， D 导通，与负载并联的电容 C 进行充能，电感与太阳能电池同时为负载供电，因此加在负载两端的电压较没有叠加前的输出电压大，所以此电路起升压作用。由式 3-1 可知，Boost 升压电路的压变只与占空比有关，所以通过调节电路的占空比 D 便可找到与最大输出功率点相对于的 V_i 。其原理图如图 3-19 所示，

Boost 升压电路的输入输出关系为：

$$V_0 = \frac{V_L}{1-D} \quad (3-1)$$

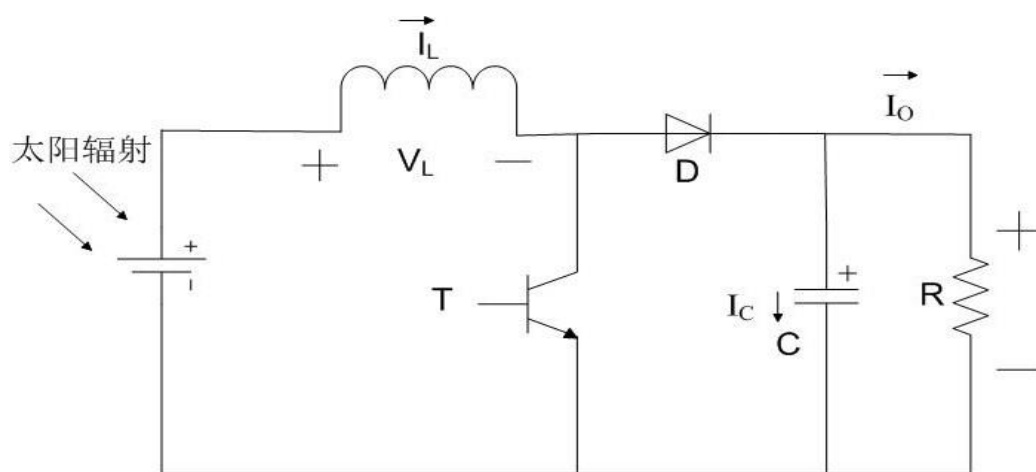


图 3-19 调节器原理图

3.6 太阳光照强度测量电路设计

3.6.1 方案论证

光照强度增大,太阳能电池的开路电压和短路电流也将增大,因此结合采集到的光照强度数据,对最大输出功率点进行分析,将有助于提高太阳能转换效率。在对光照强度的采集方面,很多用的光电池以及光电三极管,但因为要应用此类元件务必要使用A/D转换、放大电路等必要模块,这样便增加了系统的复杂度以及成本。本设计则是选用光强传感器BH1750,这一传感器可以弥补传统测量系统的不足,具有低成本高灵敏度的特点。

3.6.2 设计原理图

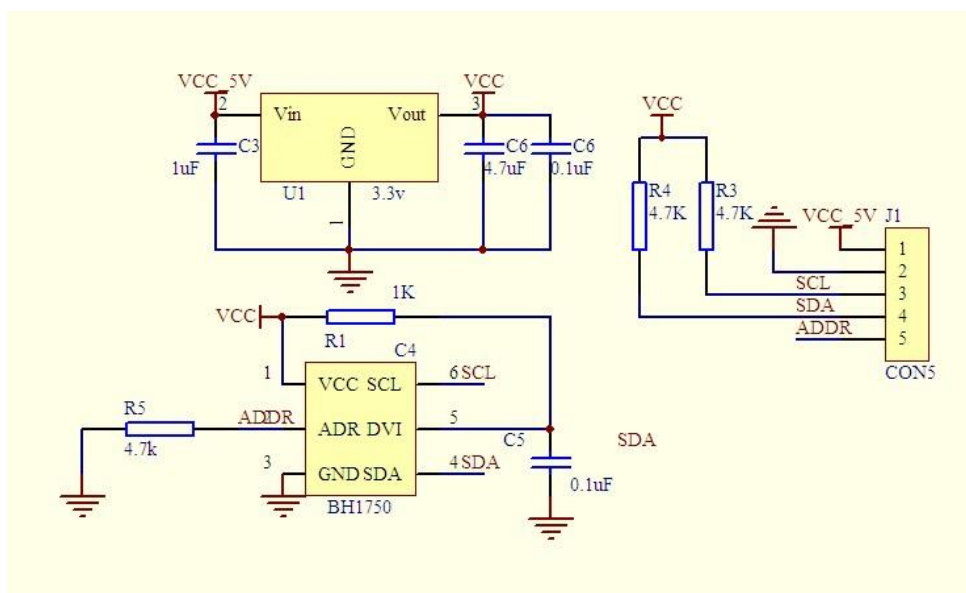
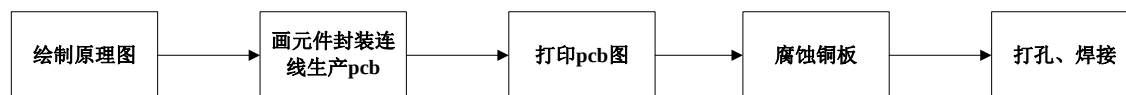


图 3-20 光强测量原理图

4 硬件制作及调试

4.1 硬件制作

硬件制作过程可以简单分为五步，其流程图如 4-1 所示：



4-1 硬件制作流程图

第一步：绘制原理图时要根据设计图严格连线，避免出现连错及少元件等情况。

第二步：对于库中已经含有的封装可以直接使用，库中没有的则需要自己手工绘制，绘制时需要对原件实物的引脚尺寸进行测量，以防出现打好孔后元件无法直插的问题。

第三步：在打印 PCB 时，需要专用的纸张打印，往铜板印图时要先对铜板进行去氧化层处理，这里一般使用砂纸打磨，采用高温印图时要把握好高温印制时间，时间不够，电路图则无法完整印制，时间过长或温度过高将会导致纸张的塑料层熔在铜板上，影响腐蚀。

第四步：对铜板进行腐蚀时，要合理搭配腐蚀液比例，才能起到最好的腐蚀效果，同时在腐蚀过程中，要不断的进行搅拌或者摇晃腐蚀容器，促进腐蚀液与铜板的充分接触。

第五步：打孔时选用粗细合适的钻针，避免过粗或者过细，过粗会破坏焊盘，过细则会影响元件的插入。

4.2 硬件调试

以下列举了在本设计的硬件演示装置制作调试过程中遇到的几个典型问题以及解决方法：

（1）温度传感器刚上电时初始采集温度为 85 摄氏度，蜂鸣器短暂性的误报警，通过查阅相关资料，发现这是 DS18B20 这个传感器的内部初始化温度，并非在采集时出现的程序错误，通过反复试验，采取了上电时加入延时的方式让主循环避开这一初始化过程，这样便解决了蜂鸣器发生误报警的情况。

（2）在调试过程中，通过对比采集到的数据和室内温度计的数据，我们发现传感器采集到的温度偏高，经过对电路板的研究，我们发现两个三端整流管，起降压整流的 LM7805 散热量较大，其周边温度较室温高 1-2 摄氏度，所以导致传感器采集到的温度

普遍偏高，对于这一问题，我们采取给 LM7805 加装散热片的方式，使热量快速散出，避免对温度采集的影响，同时把温度传感器偏向远离 LM7805 的方向，减少受热。

（3）在向上位机发送数据后，上位机温度框显示数字不全，分析原因，可能是在字符串中提取温度数据时错了位，重新检查改正后，显示正常。

（4）在设计上位机报警声响时，由于采用了重复播放&H8 语句，所以导致了报警响后无法结束的问题，修改方法是在温度恢复正常后添加一段 2 秒无声音频，并定义为 &H1，这样便可结束报警的循环播放。

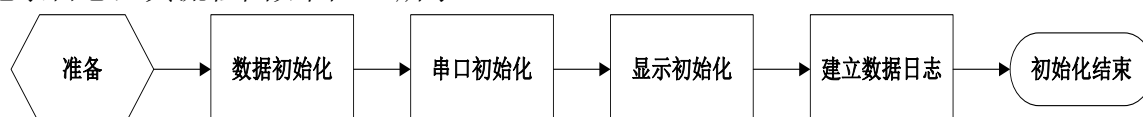
5 软件程序设计

5.1 上位机设计可行性分析

本设计上位机使用的软件是 VB6.0。通过上位机要实现对下位机上传的温度、光强采集信息实时显示功能，并且将温度的实时变化通过曲线表现出来，当温度超过设定的范围时，给出相应的警告，并记录到日志中，同时也可以通过上位机实现对下位机温度报警的上下限进行控制。对于接收下位机发来的数据以及要发出的控制数据这里主要通过 vb6.0 的 Mscomm 控件来实现。整个系统设计主要包括初始化，串口设置和温度、光强处理、控制处理、界面设计五个部分。

5.2 初始化设计

初始化设计：初始化过程包括了对串口的初始化，对显示窗口的初始化以及建立信息记录日志，其流程图如图 5-1 所示：



5-1 初始化流程图

5.2.1 数据初始化程序分析

首先要对变量进行定义：

```
Public wendu As Double, guangqiang As Double, last_wendu As Double, wendu_h As Double, wendu_l As Double, zuobiao_x As Double
```

```
Public ok_flag, over_flag, send_flag
```

定义温度值、光强值、上一次接收到的温度值、温度下限、温度上限。X 轴坐标为双精度浮点型变量，定义了三个标志位，接着对数据进行初始化：

```
wendu_h = 50          '定义初始温度上限为 50 摄氏度
```

```
wendu_l = 0          '定义初始温度下限为 0 摄氏度
```

```
location_x = -1       '定义画图时从 x=-1 开始画图
```

```
ok_flag = 0           '定义接受数据正确标志位为 0
```

```
Picture1.BackColor = &HFF&    这里将串口开关状态可视化，初始色为红色，表示串口断开
```

5.2.2 串口初始化程序分析

由于在通信时可能存在多样化的串口参数，所以本程序将串口设置通过列表框可以

进行选择，其程序如下：

```
Combo1.Text = "COM4"           '串口号选择列表框，初始显示为
com4

Combo1.AddItem "COM1"
Combo1.AddItem "COM2"
Combo1.AddItem "COM3"
Combo1.AddItem "COM4"

Combo2.Text = "4800"           '波特率列表框设置
Combo2.AddItem "1200"
Combo2.AddItem "4800"
Combo2.AddItem "9600"
Combo2.AddItem "19200"

Combo3.Text = "NONE"           '校验位列表框设置
Combo3.AddItem "NONE"         'n
Combo3.AddItem "奇校验"       'o
Combo3.AddItem "偶校验"       'e

Combo4.Text = "1"              '停止位列表框设置
Combo4.AddItem "1"
Combo4.AddItem "1.5"
Combo4.AddItem "2"

Combo5.Text = "8"              '数据位选择设置
Combo5.AddItem "8"
Combo5.AddItem "7"
Combo5.AddItem "6"
```

5.2.3 显示窗口的初始化

```
Private Sub Form_Activate()
    Picture2.DrawWidth = 3
    Picture2.Line (0, 50)-(200, 50), RGB(0, 0, 255)
    Picture2.DrawWidth = 2
    Picture2.Line (0, 40)-(200, 40), RGB(0, 0, 255)
```

```
Picture2.Line (0, 30)-(200, 30), RGB(0, 0, 255)
```

```
Picture2.Line (0, 20)-(200, 20), RGB(0, 0, 255)
```

```
Picture2.Line (0, 10)-(200, 10), RGB(0, 0, 255)
```

```
End Sub
```

这里对曲线显示框进行了初始化，分别在 y=10,20,30,40,50 绘制一条垂直于 y 轴的线，作为温度标记线。下列程序是对编程者个人信息的显示：

```
s = " 电气工程" & vbCrLf & " 李东琪" & vbCrLf & "1200120314"
```

```
Label16.Caption = s
```

对于上下限显示的初始化这里设计为触发事件，当从接收第一个字符开始，才对上下限框中的上下限进行显示，其程序如下：

```
MSComm1.RThreshold = 1
```

```
Text4.Text = Format(CStr(wendu_h), "#,#0.0") + "°C" '在 text4 框中显示  
上限
```

```
Text5.Text = Format(CStr(wendu_l), "#,#0.0") + "°C" '在 text5 的框中显  
示下限
```

5.2.4 建立数据日志设计

```
“st = Now
```

```
Open App.Path & "\" & Format(st, "yyyy-mm-dd") & ".log" For Append As #1”
```

其中 st=now 是将此时此刻的系统时间赋值给 st，通过 open app.path 指令进行数据日志的建立。

5.3 串口设置

串口设置只要是要实现对串口各个参数的选择，以及在通信中能够正常开关串口。串口设置程序如图：

```
Private Sub Command1_Click()
```

```
    If MSComm1.PortOpen = False Then
```

```
        Command1.Caption = "关闭串口"
```

```
        Picture1.BackColor = &HFF00&
```

```
        Select Case Combo1.Text '使用 select case 语句对设置串口号进行选择
```

```
            Case "COM1"
```

```
                MSComm1.CommPort = 1
```

```
            Case "COM2"
```

```
                MSComm1.CommPort = 2
```

```

Case "COM3"
    MSComm1.CommPort = 3
Case "COM4"
    MSComm1.CommPort = 4
Case Else
End Select

MSComm1_Settings = Combo2.Text '波特率
Select Case Combo3.Text '校验位
    Case "NONE"
        MSComm1_Settings = MSComm1_Settings & "," & "N"
    Case "奇校验"
        MSComm1_Settings = MSComm1_Settings & "," & "O"
    Case "偶校验"
        MSComm1_Settings = MSComm1_Settings & "," & "E"
    Case Else
End Select
MSComm1_Settings = MSComm1_Settings & "," & Combo5.Text '
数据位

MSComm1_Settings = MSComm1_Settings & "," & Combo4.Text '
停止位

MSComm1.Settings = MSComm1_Settings '设置串口
MSComm1.PortOpen = True
Else
    Command1.Caption = "打开串口"
    Picture1.BackColor = &HFF&
    MSComm1.PortOpen = False
End If
End Sub

```

5.4 温度、光强处理方案

当串口接收到下位机发来的数据后，要对数据进行处理，才能正常显示。首先将接收到的数据提取出来赋值给临时变量，然后对接收到的数据进行校验，标志位正确，则将温度与光强从字符串中提取出来，并且进行显示，其程序如下：

```
Private Sub MSComm1_OnComm()  
    rec = MSComm1.Input  
    If ok_flag = 1 And Right(rec, 1) = "?" Then          '可以记录数据且最后一个  
字节是分隔符  
        last_wendu = wendu                                '记录上次温度,  
供画图用  
        wendu = Val(Mid(rec, 7, 5))                      '去除分隔符  
        guangqiang = Val(Left(rec, 5))  
        Text1.Text = Format(CStr(wendu), "#,##00.0") + "°C"    '处理显示  
        Text2.Text = Format(CStr(guangqiang)) + "XL"
```

接下来要对接收到的实时温度与设置好的温度上下限进行对比,如果大于上限,或者小于下限,则在警报框中输出警报时间以及警报信息,并将报警信息输出到数据日志中进行保存。程序设计如下:

```
If wendu > wendu_l And wendu < wendu_h And over_flag <> 0 Then  
    st = Now  
    List1.AddItem Format(st, "hh:mm:ss") + " 恢复正常"  
    Print #1, Format(st, "hh:mm:ss") + " 恢复正常"输出至日志  
    over_flag = 0  
End If  
  
If wendu > wendu_h Then  
    Text4.BackColor = &HFF&  
    If over_flag <> 1 Then  
        st = Now  
        List1.AddItem Format(st, "hh:mm:ss") + " 超过上限"  
        Print #1, Format(st, "hh:mm:ss") + " 超过上限"  
        over_flag = 1  
    End If  
  
Else  
    Text4.BackColor = &H80000005  
End If  
  
If wendu < wendu_l Then  
    Text5.BackColor = &HFF0000  
    If over_flag <> 2 Then
```

```
st = Now
List1.AddItem Format(st, "hh:mm:ss") + " 低过下限"
Print #1, Format(st, "hh:mm:ss") + " 低过下限"
over_flag = 2
End If
Else
Text5.BackColor = &H80000005
End If
```

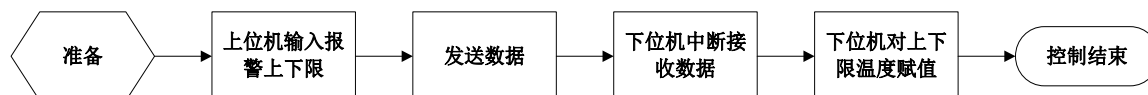
对温度、光强进行显示后，要求设计将温度的实时曲线显示出来，设计思路如下：先将收到的温度与温度上下限进行对比，如果温度为正常温度，则输出绿色曲线描点；如果温度为异常温度，则输出红色曲线，以便醒目的提示工作人员，其设计程序如下：

```
If wendu < wendu_l Or wendu > wendu_h Then
Picture2.DrawWidth = 2
Picture2.Line (location_x, 50 - last_wendu)-(location_x + 1, 50 - wendu),
RGB(255, 0, 0)
location_x = location_x + 1
Else

Picture2.DrawWidth = 2
Picture2.Line (location_x, 50 - last_wendu)-(location_x + 1, 50 - wendu),
RGB(0, 255, 0) '画图
location_x = location_x + 1
End If
If location_x > 200 Then
location_x = 0
Picture2.Picture = LoadPicture("")
Picture2.DrawWidth = 2
Picture2.Line (0, 50)-(200, 50), RGB(0, 0, 255)
Picture2.DrawWidth = 1
Picture2.Line (0, 40)-(200, 40), RGB(0, 0, 255)
Picture2.Line (0, 30)-(200, 30), RGB(0, 0, 255)
Picture2.Line (0, 20)-(200, 20), RGB(0, 0, 255)
Picture2.Line (0, 10)-(200, 10), RGB(0, 0, 255)
End If
```

5.5 控制下位机设计

本设计要实现对下位机的控制，主要体现在通过上位机输入报警温度的上下限，通过串口通信发送到下位机，对下位机的上下限温度重新赋值，是下位机报警温度更新，其程序设计流程图如图 5-2 所示：



5-2 下位机控制流程图

设计程序如下所示：

```

If send_flag = 1 Then
    wendu_h = Text6.Text
    wendu_l = Text7.Text
    MSComm1.Output = "C"           '以 0x43 做为引导作为标志位
If Val(Text6.Text) >= 0 Then
    MSComm1.Output = "+"
Else
    MSComm1.Output = "-"
End If
    MSComm1.Output = Format(CStr(Abs(Val(Text6.Text))), "#000")      '上限
If Val(Text7.Text) >= 0 Then
    MSComm1.Output = "+"
Else
    MSComm1.Output = "-"
End If
    MSComm1.Output = Format(CStr(Abs(Val(Text7.Text))), "#000")      '下限
End If                                     '对发送数据进行处理
End If
  
```

5.6 界面设计

5.6.1 员工登录界面设计

这一设计是出于对系统的安全性考虑所设置的一道安全屏障，可以有效地防止非工作人员进行的误操作，设计界面如图 5-3 所示，界面主要包括了软件名、账号输入框、密码输入框、确定及退出操作，背景主题为绿色能源。



图 5-3 系统登录界面

5.6.2 采集界面设计

采集界面包括了用于串口通信的串口设置、实时温度显示框、实时光强显示框、温度上下限显示框、温度上下限设置框、报警信息提示框、温度曲线绘制框以及个人信息的显示。设计如图 5-4 所示。

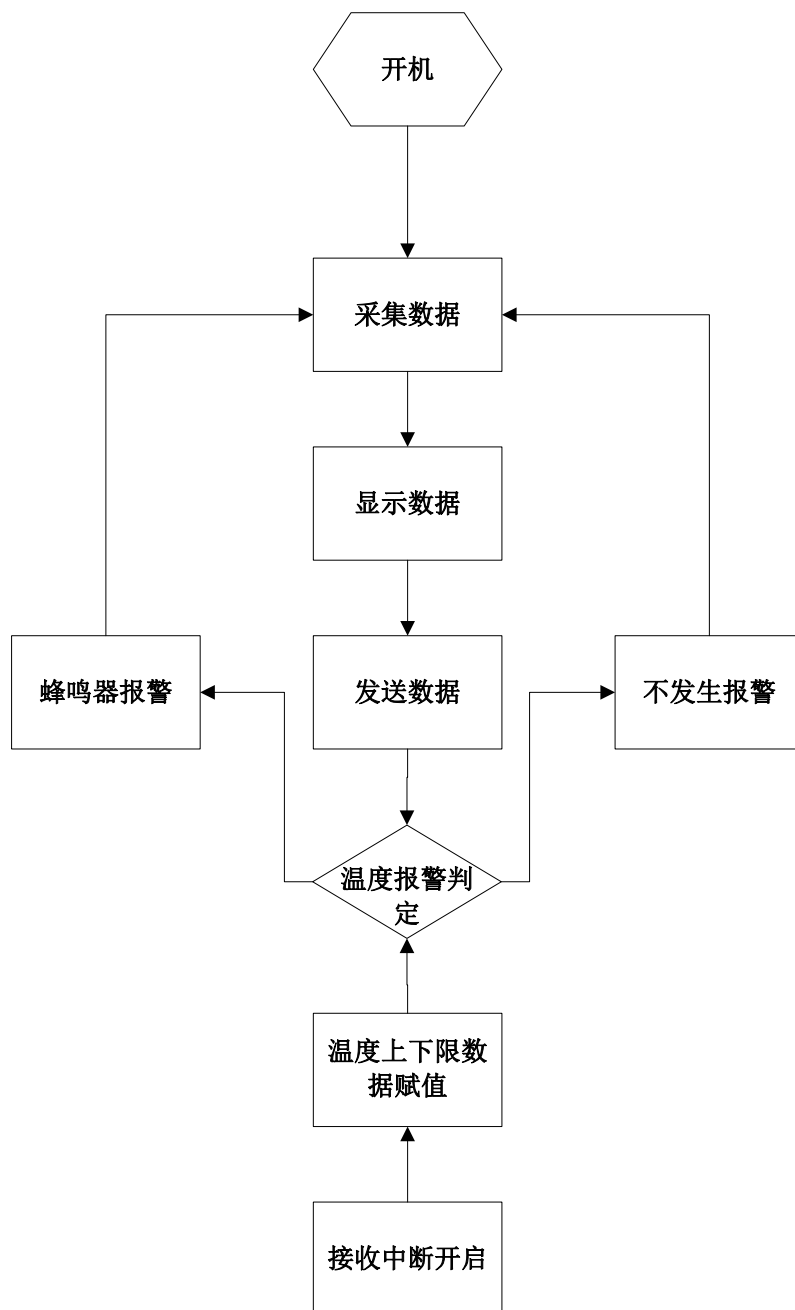


图 5-4 上位机采集显示界面

5.7 下位机程序设计

5.7.1 主程序设计

下位机的功能是对温度以及光照强度进行采集和显示，并且通过串口通信发送到上位机，设置接收中断来接受来自上位机的温度上下限控制数据，本设计分为显示、通信、光强采集、温度采集四部分进行设计。其设计主程序流程图如图 5-5 所示：



5-5 主程序设计流程图

主程序设计及分析如下所示：

```
void main(void)
{
```

```

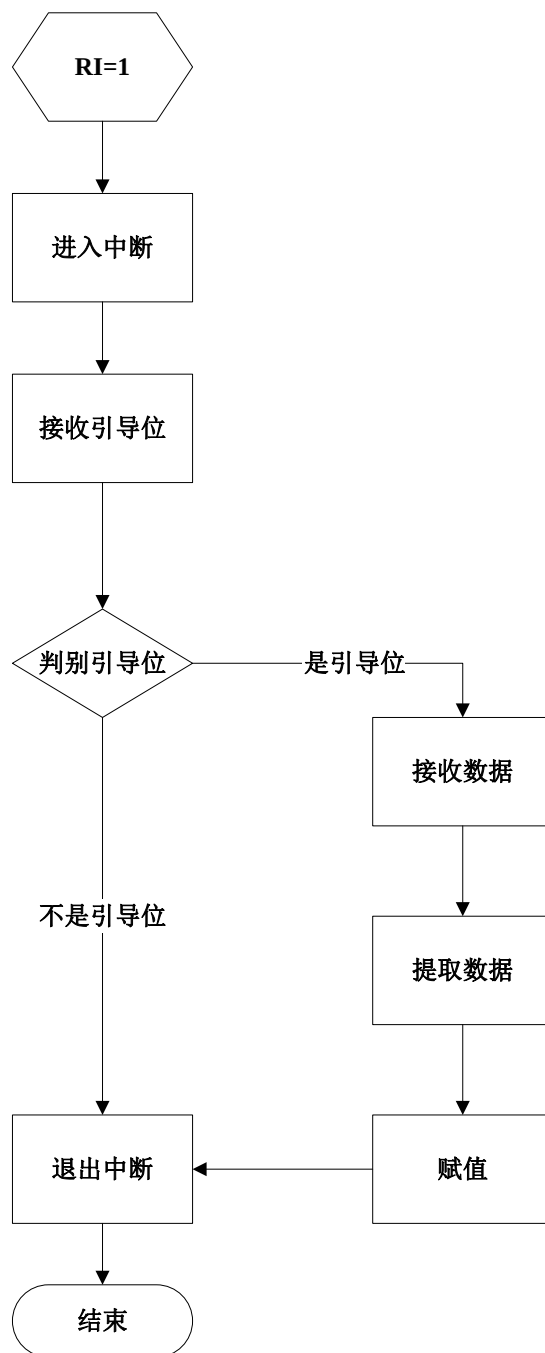
UsartConfiguration();对串口进行初始化
BEEP=1;对蜂鸣器进行初始化
Init_LCD1602();对 LCD1602 进行初始化
Init_BH1750(); 对 BH1750 模块进行初始化
Write_Str(0x00,"Light:      Lx "); 将固定界面进行显示-光强名称及其单位
Write_Str(0x40,"Wendu:      C  ");  将固定界面进行显示-温度及其单位
WenDu=ReadTemperature();在进入主循环前读取温度，并加延时是为了避开温
度传感器的上电温度 85 度，防止蜂鸣器短时误报警
Delay_ms(1000);

while(1)进入主循环
{
    Single_Write_BH1750(0x01);// power on
    Single_Write_BH1750(0x10);// H- resolution mode
    Delay_ms(180);
    AD=Multiple_read_BH1750();    读出 BH1750 数据，赋值给 AD 变量
    Display_1(AD);                光照强度的显示函数
    WenDu=ReadTemperature();      获取温度
    Display_2(WenDu);             显示温度
    if(WenDu<WenDu_l||WenDu_h<WenDu)蜂鸣器报警条件语句
        { BEEP=0;如果测得温度大于上限或者小于下限，则蜂鸣器报
警
        }
    else
        {BEEP=1;  }
}
}

```

5.7.2 接收中断程序设计

首先定义函数为串口中断函数，当上位机发来数据是，RI 自动置 1，此时进入接收中断，首先要将 RI 手工置回 0，因为系统不会自动置 0，需要软件置 0，接下来将接收到的第一位数据赋值给变量，对变量进行判别，是否为上位机发来的标志字符，如果时，则对后面的字符串进行接收，再将数据的每一位进行提取，分类出上下限温度并赋值给下位机原上下限温度，退出中断等待下一次接收数据。设计流程图如图 5-6 所示：



5-6 中断程序设计流程图

程序设计以及分析如下所示：

```
void Uart_Isr() interrupt 4
```

```
{
```

```
    if (RI)当 RI=1 时进入中断
```

```
    {
```

```
        RI = 0;
```

```
        Uart_date = SBUF; SBUF 为接收寄存器
```

```
if (Uart_date==0x43)引导位判别
{
    get_date_flag=1;接收数据标志置 1 开始接收
}
else if (get_date_flag)
{
    get_date[i]=Uart_date;将数据放入数组

    i++;

    if(i==8)        对数据进行提取及赋值，并判断数据正负
    {
        i=0;
        get_date_flag=0;
        WenDu_h=(get_date[1]-0x30)*100+(get_date[2]-0x30)*10
                +(get_date[3]-0x30);
        if(get_date[0]==0x2D)
        {
            WenDu_h=WenDu_h*(-1);
        }
        WenDu_l=(get_date[5]-0x30)*100+(get_date[6]-0x30)*10
                +(get_date[7]-0x30);
        if(get_date[4]==0x2D)
        {
            WenDu_l=WenDu_l*(-1);
        }
    }
}
}
```

（注：详细子程序见附录）

6 结束语

随着环境的日益恶化，污染的大大加重，人类对太阳能这种清洁能源的使用已成为了必然，同时随着技术的不断革新、研究不断的深入，太阳能的转换效率会大幅度增加，伴着使用成本的降低，相信那时的太阳能供电会成为城市供电的主能源，所以并网发电技术还有很大的提升空间，也是目前最有前景的一项能源发展技术。

本设计在传统的监控系统控制装置上进行了优化，根据近年来的智能电网概念，设计出了智能型监控装置，通过人工编程便可以实现对数据的采集、现场的异常预警以及上下位机间的人机互动，在实际应用中，可以降低劳动力成本，实现现场无人监管化。

太阳能的转换效率是太阳能发展的最为关键的指标，在很多大型的太阳能电站使用的都是可调角式太阳能板，随着太阳角的变化对方阵板进行调节，使之转换率处于最大，但是对于一些特殊场所无法进行调角的太阳能板，则可以设计最大功率跟踪电路，本课题设计的跟踪电路是根据控制器调节信号占空比来调节电路的最大功率输出点，在一定程度上提高了光伏发电转换效率。

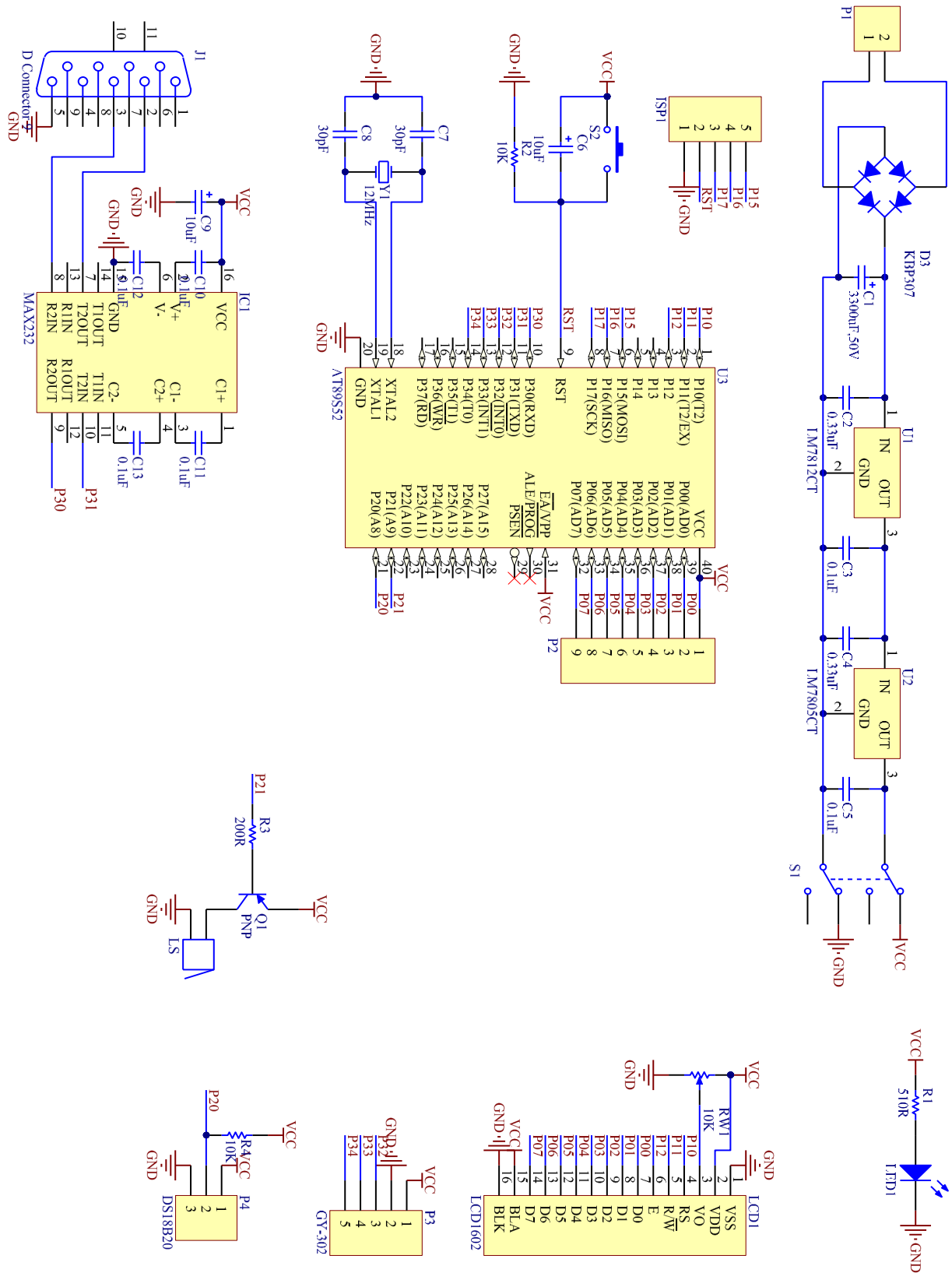
致 谢

"不积跬步无以至千里，不积小流无以成江海"，这次毕业论文能够最终顺利完成，归功于各位老师四年间的认真负责，使我较好的掌握了相关的专业知识，并在毕业论文中得以体现。也正是你们长期不懈的支持和帮助才使得我的毕业论文最终顺利完成，同时也感谢母校浓厚的学术氛围，舒适的学习条件，使得我的四年青春没有白白浪费。最后，我向电气专业全体老师们再次表示衷心感谢：谢谢你们四年的辛勤栽培！

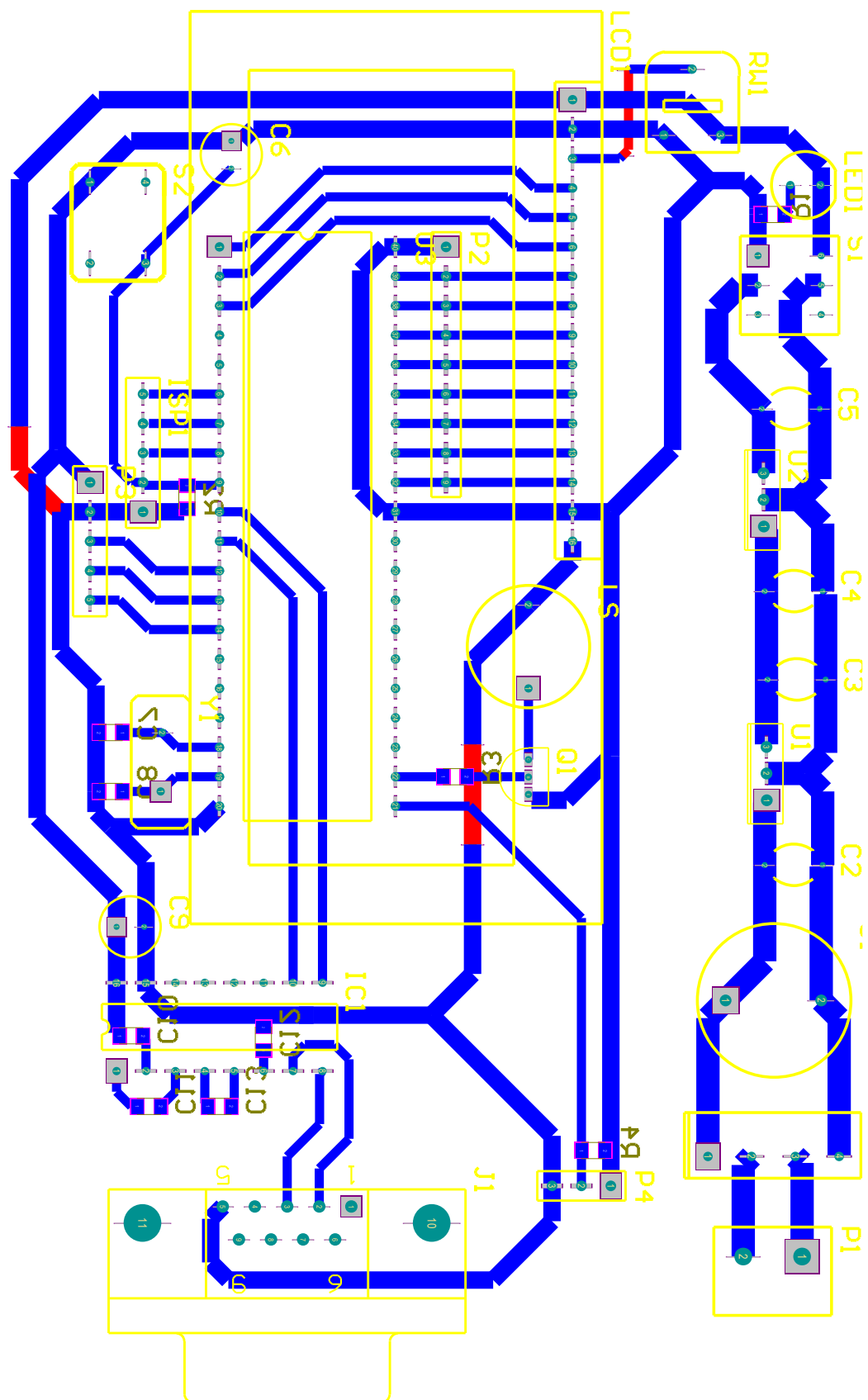
参考文献

- [1] 林楠. 独立光伏发电系统远程监控装置设计与实现[D].哈尔滨工程大学,2012.
- [2] 熊远生. 太阳能光伏发电系统的控制问题研究[D].浙江工业大学,2009.
- [3] 王传辉. 太阳能光伏发电系统的研究[D].哈尔滨工程大学,2008.
- [4] 王武军. 太阳能光伏发电系统最大功率点跟踪研究[D].陕西师范大学,2009.
- [5] 艾雨. 基于单片机的太阳能电池数据采集系统研究[D].武汉理工大学,2010.
- [6] 王成福,李锐,刘辉荣,王超,段录平. 光伏发电监控系统的设计与实现[J]. 电力系统通信,2011,06:53-57.
- [7] 车本佳,冯毅. 太阳能最大功率跟踪装置的设计[J]. 节能技术,2011,06:552-555+559.
- [8] 时剑. 太阳能最大功率点跟踪控制系统的设计与实现[D].江苏大学,2009.
- [9] 付文辉. 太阳能光伏发电监控系统的设计与实现[D].电子科技大学,2008.
- [10] 郭莉. 太阳能光伏发电单元综合监控系统的设计与实现[D].华北电力大学（河北）,2009.
- [11] 缪进荣. 太阳能光伏电站监控系统的设计与实现[D].北京交通大学,2012.
- [12] 王超. 独立运行光伏发电系统控制器的设计与实现[D].浙江大学,2004.
- [13] 李瑶. 太阳能光伏发电监控系统的设计与实现[D].东北大学,2011.
- [14] 李建军. 太阳能光伏发电监控系统的研究和设计[J]. 中国电业(技术版),2013,07:57-60.
- [15] 朱相丽. 世界光伏发电产业的现状及原材料的发展趋势[J]. 新材料产业,2008,11:34-39.
- [16] 肖丽仙,何永泰,彭跃红,刘晋豪,李雷. 温度对光伏电池转换特性影响的理论及实验研究[J]. 电测与仪表,2014,17:62-66.
- [17] S. Lyden, M.E. Haque. Maximum Power Point Tracking techniques for photovoltaic systems: A comprehensive review and comparative analysis[J]. 2015,52.

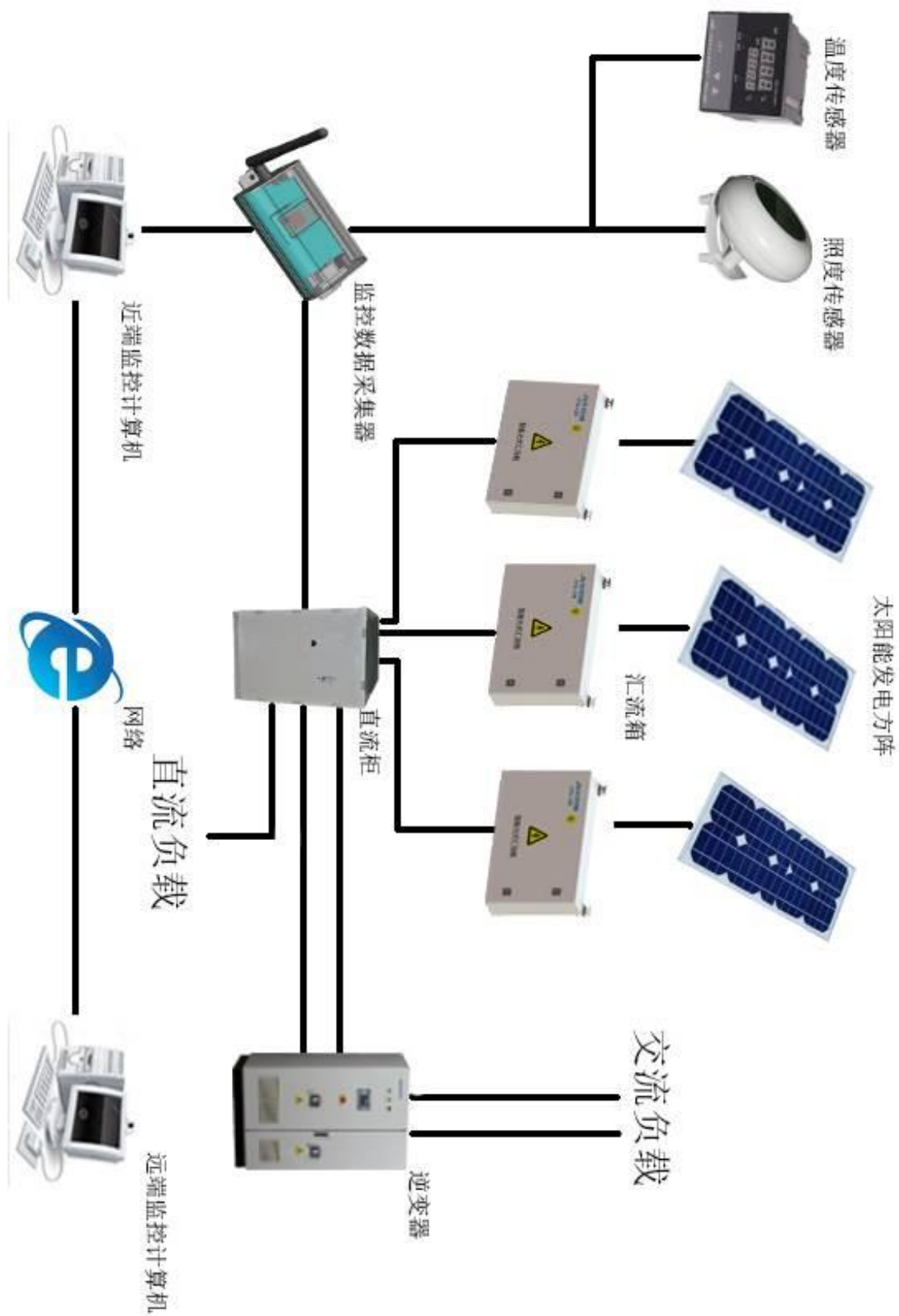
附录一 硬件演示设备原理图



附录二 硬件演示设备 PCB 图



附录三 太阳能发电系统结构示意图



附录四 硬件演示设备源程序

（1）上位机源程序：

```
Public wendu As Double, guangqiang As Double, last_wendu As Double, wendu_h As Double, wendu_l As Double, location_x As Double
Public ok_flag, over_flag, send_flag
Private Declare Function sndPlaySound Lib "winmm.dll" Alias "sndPlaySoundA" (ByVal lpszSoundName As String, ByVal uFlags As Long) As Long
Dim sound As String
Private Sub Command1_Click()
    If MSComm1.PortOpen = False Then
        Command1.Caption = "关闭串口"
        Picture1.BackColor = &HFF00&
        Select Case Combo1.Text
            Case "COM1"
                MSComm1.CommPort = 1
            Case "COM2"
                MSComm1.CommPort = 2
            Case "COM3"
                MSComm1.CommPort = 3
            Case "COM4"
                MSComm1.CommPort = 4
            Case Else

        End Select
        MSComm1_Settings = Combo2.Text
        Select Case Combo3.Text
            Case "NONE"
                MSComm1_Settings = MSComm1_Settings & "," & "N"
            Case "奇校验"
                MSComm1_Settings = MSComm1_Settings & "," & "O"
            Case "偶校验"
                MSComm1_Settings = MSComm1_Settings & "," & "E"
            Case Else
```

```
End Select
MSComm1_Settings = MSComm1_Settings & "," & Combo5.Text
MSComm1_Settings = MSComm1_Settings & "," & Combo4.Text
MSComm1.Settings = MSComm1_Settings
MSComm1.PortOpen = True
Else
    Command1.Caption = "打开串口"
    Picture1.BackColor = &HFF&
    MSComm1.PortOpen = False
End If
End Sub

Private Sub Command2_Click()
    send_flag = 1
End Sub

Private Sub Form_Activate()
    Picture2.DrawWidth = 2
    Picture2.Line (0, 50)-(200, 50), RGB(0, 0, 255)
    Picture2.DrawWidth = 1
    Picture2.Line (0, 40)-(200, 40), RGB(0, 0, 255)
    Picture2.Line (0, 30)-(200, 30), RGB(0, 0, 255)
    Picture2.Line (0, 20)-(200, 20), RGB(0, 0, 255)
    Picture2.Line (0, 10)-(200, 10), RGB(0, 0, 255)
End Sub

Private Sub Form_Load()
    wendu_h = 30
    wendu_l = 24
    location_x = -1
    ok_flag = 0
```

```
Picture1.BackColor = &HFF&
```

```
Combo1.Text = "COM4"
```

```
Combo1.AddItem "COM1"
```

```
Combo1.AddItem "COM2"
```

```
Combo1.AddItem "COM3"
```

```
Combo1.AddItem "COM4"
```

```
Combo2.Text = "4800"
```

```
Combo2.AddItem "1200"
```

```
Combo2.AddItem "2400"
```

```
Combo2.AddItem "4800"
```

```
Combo2.AddItem "9600"
```

```
Combo2.AddItem "19200"
```

```
Combo3.Text = "NONE"
```

```
Combo3.AddItem "NONE"
```

```
Combo3.AddItem "奇校验"
```

```
Combo3.AddItem "偶校验"
```

```
Combo4.Text = "1"
```

```
Combo4.AddItem "1"
```

```
Combo4.AddItem "1.5"
```

```
Combo4.AddItem "2"
```

```
Combo5.Text = "8"
```

```
Combo5.AddItem "8"
```

```
Combo5.AddItem "7"
```

```
Combo5.AddItem "6"
```

```
MSComm1.RThreshold = 1
```

```
Text4.Text = Format(CStr(wendu_h), "#,#0.0") + "°C"
```

```
Text5.Text = Format(CStr(wendu_l), "#,#0.0") + "°C"
```

```
Text6.Text = ""
```

```
Text7.Text = ""

s = " 电气工程" & vbCrLf & " 李东琪" & vbCrLf & "1200120314"
Label16.Caption = s
st = Now
Open App.Path & "\" & Format(st, "yyyy-mm-dd") & ".log" For Append As #1
End Sub

Private Sub MSComm1_OnComm()
    rec = MSComm1.Input
    If ok_flag = 1 And Right(rec, 1) = "?" Then
        last_wendu = wendu
        wendu = Val(Mid(rec, 7, 5))
        guangqiang = Val(Left(rec, 5))
        Text1.Text = Format(CStr(wendu), "#,##00.0") + "°C"
        Text2.Text = Format(CStr(guangqiang)) + "XL"
        If wendu > wendu_l And wendu < wendu_h And over_flag <> 0 Then
            st = Now
            List1.AddItem Format(st, "hh:mm:ss") + " 恢复正常"
            Print #1, Format(st, "hh:mm:ss") + " 恢复正常"
            sound = App.Path & "\无声.wav"
            sndPlaySound sound, &H1
            over_flag = 0
        End If
        If wendu > wendu_h Then
            Text4.BackColor = &HFF&
            If over_flag <> 1 Then
                st = Now
                List1.AddItem Format(st, "hh:mm:ss") + " 超过上限"
                Print #1, Format(st, "hh:mm:ss") + " 超过上限"
                sound = App.Path & "\报警.wav"
                sndPlaySound sound, &H1 Or &H8
                over_flag = 1
            End If
        End If
    End If
End Sub
```

```
Else
    Text4.BackColor = &H80000005

End If
If wendu < wendu_l Then
    Text5.BackColor = &HFF0000
    If over_flag <> 2 Then
        st = Now
        List1.AddItem Format(st, "hh:mm:ss") + " 低过下限"
        Print #1, Format(st, "hh:mm:ss") + " 低过下限"
    sound = App.Path & "\报警.wav"
        sndPlaySound sound, &H1 Or &H8
        over_flag = 2
    End If
Else
    Text5.BackColor = &H80000005
End If
If last_wendu < wendu_l Or last_wendu > wendu_h Then
    Picture2.DrawWidth = 2
    Picture2.Line (location_x, 50 - last_wendu)-(location_x + 1, 50 - wendu),
    RGB(255, 0, 0)
    location_x = location_x + 1
Else

    Picture2.DrawWidth = 2
    Picture2.Line (location_x, 50 - last_wendu)-(location_x + 1, 50 - wendu),
    RGB(0, 255, 0)    '画图
    location_x = location_x + 1
End If

If location_x > 200 Then

    location_x = 0
    Picture2.Picture = LoadPicture("")
    Picture2.DrawWidth = 2
```

```
Picture2.Line (0, 50)-(200, 50), RGB(0, 0, 255)
Picture2.DrawWidth = 1
Picture2.Line (0, 40)-(200, 40), RGB(0, 0, 255)
Picture2.Line (0, 30)-(200, 30), RGB(0, 0, 255)
Picture2.Line (0, 20)-(200, 20), RGB(0, 0, 255)
Picture2.Line (0, 10)-(200, 10), RGB(0, 0, 255)

End If
If (Val(Text4.Text) = Val(Text6.Text)) And (Val(Text5.Text) = Val(Text7.Text))
Then
    send_flag = 0
Else
    If send_flag = 1 Then
        wendu_h = Text6.Text

wendu_l = Text7.Text
Text4.Text = Format(CStr(wendu_h), "#, #0.0") + "°C"

Text5.Text = Format(CStr(wendu_l), "#, #0.0") + "°C"
MSComm1.Output = "C"
If Val(Text6.Text) >= 0 Then
    MSComm1.Output = "+"
Else
    MSComm1.Output = "-"
End If
MSComm1.Output = Format(CStr(Abs(Val(Text6.Text))), "#000")

'上限
If Val(Text7.Text) >= 0 Then
    MSComm1.Output = "+"
Else
    MSComm1.Output = "-"
End If
MSComm1.Output = Format(CStr(Abs(Val(Text7.Text))), "#000")

'下限

End If
```


End If

Else

ok_flag = 0

MSComm1.RThreshold = 1 '

End If

If Right(rec, 1) = "?" And ok_flag = 0 Then

ok_flag = 1

MSComm1.RThreshold = 10

End If

End Sub

登录界面源程序：

Private Sub Command1_Click()

If (Text2.Text <> "123456") Then

MsgBox ("密码错误！")

ElseIf Text1.Text <> "1200120314" Then

MsgBox ("用户不存在！")

ElseIf (Text1.Text = "1200120314" And Text2.Text = "123456") Then

Form1.Hide

Form2.Show

End If

End Sub

Private Sub Command2_Click()

End

Unload Me

End Sub

（2）下位机源程序：

/******

包含的引用文件

*****/

#include <reg52.h>

#include <intrins.h>

```
#include "lcd1602.h"
#include "ds18b20.h"
#include "BH1750.h"
#define uchar unsigned char
#define uint unsigned int
#define PIO P0

/*****
引脚定义
*****/

sbit BEEP=P2^1;      // 蜂鸣器引脚定义，0-开
sbit SCL=P3^2;      //集成模块 IIC 时钟引脚定义
sbit SDA=P3^3;      //集成模块 IIC 数据引脚定义
sbit L_RS=P1^0;
sbit L_RW=P1^1;
sbit L_E=P1^2;
sbit L_BF=P0^7;
sbit DQ=P2^0;

/*****
变量定义
*****/

uchar code table[10]={"0123456789"};
uint AD=0;
float WenDu=0;
int WenDu_l=0;
int WenDu_h=30;
uchar Uart_date;
bit get_date_flag;
uchar get_date[6],i;

/*****
函数声明
*****/

void Delay_ms(uint x);      // 延时函数
void KeyScan(void);        // 按键检测函数
void Display_1(uint n);     // 显示光照强度
void Display_2(float n);    // 显示温度
```

```
void UsartConfiguration(); //串口初始化
void Uart_Isr();           //接收中断
void delay(unsigned int i); //延时
/*****
    主函数
    *****/
void main(void)
{
    UsartConfiguration();

    BEEP=1;
    Init_LCD1602(); // LCD1602 初始化
    Init_BH1750(); //初始化 BH1750
                  // 0123456789abcdef

    Write_Str(0x00,"Light:      Lx "); // LCD1602 界面显示
    Write_Str(0x40,"Wendu:      C  "); // LCD1602 界面显示
    Delay_ms(500);

    while(1)
    {

        Single_Write_BH1750(0x01); // power on
        Single_Write_BH1750(0x10); // H- resolution mode
        Delay_ms(180);
        AD=Multiple_read_BH1750(); // 读出 BH1750 数据
        Display_1(AD);

        WenDu=ReadTemperature(); // 获取温度
        Display_2(WenDu);         // 显示温度
        if(WenDu<WenDu_l||WenDu_h<WenDu)
            { BEEP=0;}
        else
            {BEEP=1; }
    }
}
```

```
    }
}

/*****

延时函数

*****/

void Delay_ms(uint x)
{

    uchar i;
    while(x--)
        for(i=0;i<115;i++)
            ;
}

/*****

光强显示及发送数据函数

*****/

void Display_1(uint n)
{
    uchar str[6]={0};          // 存储要显示的 9 个字符

    str[0]=table[n/10000%10];   // 取出万位
    SBUF = table[n/10000%10];   //将接收到的数据放入到发送寄存器
    while (!TI);               //等待发送数据完成
    TI = 0;

    str[1]=table[n/1000%10];    // 千位
    SBUF = table[n/1000%10];    //将接收到的数据放入到发送寄存器
    while (!TI);               //等待发送数据完成
    TI = 0;

    str[2]=table[n/100%10];     // 百位
    SBUF = table[n/100%10];     //将接收到的数据放入到发送寄存器
    while (!TI);               //等待发送数据完成
    TI = 0;

    str[3]=table[n/10%10];      // 十位
    SBUF = table[n/10%10];      //将接收到的数据放入到发送寄存器
```

```
        while (!TI);          //等待发送数据完成
        TI = 0;
        str[4]=table[n%10];      // 个位
        SBUF = table[n%10];      //将接收到的数据放入到发送寄存器
        while (!TI);          //等待发送数据完成
        TI = 0;
        SBUF = '/';             //将接收到的数据放入到发送寄存器
        while (!TI);          //等待发送数据完成
        TI = 0;
        str[5]='\0';             // 结束显示符号
        Write_Str(0x07,str);     // 在 LCD1206 上显示
    }

/*****
    温度显示及发送函数函数
    *****/

void Display_2(float n)
{
    uchar CNCHAR[6] = "摄氏度";
    unsigned char i;
    uchar str[5]={0};           // 存储要显示的 4 个字符
    uint t;
    t=n*10;
    str[0]=table[t/100%10];      // 取出十位
    SBUF = table[t/100%10];      //将接收到的数据放入到发送寄存器
    while (!TI);                //等待发送数据完成
    TI = 0;

    str[1]=table[t/10%10];       // 个位
    SBUF = table[t/10%10];       //将接收到的数据放入到发送寄存器
    while (!TI);                //等待发送数据完成
    TI = 0;

    str[2]='.';
    SBUF = '.';                  //将接收到的数据放入到发送寄存器
    while (!TI);                //等待发送数据完成
    TI = 0;

    str[3]=table[t%10];          // 个小数位
```

```

    SBUF = table[t%10];          //将接收到的数据放入到发送寄存器
    while (!TI);                //等待发送数据完成
    TI = 0;
    str[4]='\0';                 // 结束显示符号
    /* for(i=0; i<6; i++)
    {
        SBUF = CNCHAR[i];       //将接收到的数据放入到发送寄存器
        while (!TI);            //等待发送数据完成
        TI = 0;
    } */
    SBUF = '?';                  //将接收到的数据放入到发送寄存器
    while (!TI);                //等待发送数据完成
    TI = 0;
    Write_Str(0x48,str);         // 在 LCD1206 上显示

}

/*****
*****

串口初始化函数

*****/

void UsartConfiguration()
{
    SCON=0X50;                  //设置为工作方式 1
    TMOD=0x20;                  //设置计数器工作方式 2
    PCON=0X80;                  //波特率加倍
    TH1=0XF4;                   //计数器初始值设置，注意波特率是 4800 的
    TL1=0XF4;
    ES=1;                       //打开接收中断
    EA=1;                       //打开总中断
    TR1=1;                      //打开计数器
}

/*****
接收中断函数

```

*****/

```
void Uart_Isr() interrupt 4
{
    if (RI)
    {
        RI = 0;
        Uart_date = SBUF;
        //SendData(Uart_date);
        if (Uart_date==0x43)
        {
            get_date_flag=1;
        }
        else if (get_date_flag)
        {
            get_date[i]=Uart_date;

            i++;

            if(i==8)
            {
                i=0;
                get_date_flag=0;
                WenDu_h=(get_date[1]-0x30)*100+(get_date[2]-0x30)*10
                    +(get_date[3]-0x30);
                if(get_date[0]==0x2D)
                {
                    WenDu_h=WenDu_h*(-1);
                }
                WenDu_l=(get_date[5]-0x30)*100+(get_date[6]-0x30)*10
                    +(get_date[7]-0x30);
                if(get_date[4]==0x2D)
                {
                    WenDu_l=WenDu_l*(-1);
                }
            }
        }
    }
}
```

```
        }
    }
}

}

/*****
    初始化 lcd1602
    *****/
void Init_LCD1602(void)
{
    delayxms(1);
    L_E =0;
    delayxms(1);
    Write_Com(0x38);
    delayxms(1);
    Write_Com(0x0c);
    delayxms(1);
    Write_Com(0x06);
    delayxms(1);
    Write_Com(0x01);
    delayxms(1);
}

/*****
lcd1602 延时
    *****/

void delayxms(uint xms)
{
    uchar i;
    while(xms--)
        for(i=114;i>0;i--)
            ;
}
```



```
/******
```

```
    lcd1602 判忙
```

```
*****/
```

```
bit BusyTest(void)
```

```
{
```

```
    bit result;
```

```
    L_RS=0;           // RS=0, 操作指令（状态）
```

```
    L_RW=1;           // RW=1, 选择读
```

```
    L_E =1;           // E=1, 允许读取信息
```

```
    _nop_();
```

```
    _nop_();
```

```
    _nop_();
```

```
    _nop_();
```

```
    result=L_BF;       // 读出当前状态
```

```
    L_E =0;           // E=0
```

```
    return result; // 返回当前状态值
```

```
}
```

```
/******
```

```
lcd1602 写入指令
```

```
*****/
```

```
void Write_Com(uchar Com)
```

```
{
```

```
    while(BusyTest()!=1);
```

```
    L_RS=0;
```

```
    L_RW=0;
```

```
    L_E =0;
```

```
    _nop_();
```

```
    _nop_();
```

```
    PIO=Com;
```

```
    _nop_();
```

```
    _nop_();
```

```
    _nop_();
```

```
    _nop_();
```

```
    L_E =1;
```

```
_nop();
_nop();
_nop();
_nop();
L_E =0;
}

/*****

lcd1602 写入地址

*****/

void Write_Addr(uchar Addr)
{
    Write_Com(0x80+Addr);
}

/*****

lcd1602 写入数据

*****/

void Write_Data(uchar Data)
{
    while(BusyTest()==1) ;
    L_RS=1;
    L_RW=0;
    L_E =0;

    _nop();
    _nop();
    PIO=Data;
    _nop();
    _nop();
    _nop();
    _nop();
    L_E =1;
    _nop();
    _nop();
    _nop();
    _nop();
```

```
L_E =0;
}
/*****
lcd1602 写入字符串
*****/
void Write_Str(uchar addr,uchar *str)
{
    uchar i=0;
    Write_Addr(addr);
    while(str[i] != '\0')
    {
        Write_Data(str[i]);
        delayxms(1);
        i++;
    }
}
/*****
Ds18b20 初始化
*****/
bit Init_ds18b20(void)
{
    bit flag;                // 储存 ds18b20 是否存在的标志，flag=0，存在；
    flag=1，不存在
    uchar time;
    DQ = 1;
    for(time=0;time<2;time++) // 略微延时约 6 微秒
        ;
    DQ = 0;
    for(time=0;time<200;time++)
        ;
    DQ = 1;
    for(time=0;time<10;time++)
        ;
    flag=DQ;                // 检测是否存在脉冲（DQ=0 表示存在）
    for(time=0;time<200;time++)
```

```
        ;
    return (flag);                // 返回检测存在标志
}

/*****
Ds18b20 读取一个字节数据
*****/
uchar ReadOneChar(void)
{
    uchar i=0,time;
    uchar dat;
    for (i=0;i<8;i++)            // 读数据 dat
    {
        DQ =1;
        _nop_();
        DQ = 0;
        _nop_();
        DQ = 1;
        for(time=0;time<2;time++)
            ;

        dat>>=1;
        if(DQ==1)
            dat|=0x80;            // 如果读到的数据是 1，则将 1 存入 dat
        else
            dat|=0x00;            // 如果读到的数据是 0，则将 0 存入 dat

        for(time=0;time<8;time++)
            ;                    // 延时 3us,两个读时序之间必须有大于 1us
的恢复期
    }

    return(dat);                // 返回数据
}
```

```
/******
```

Ds18b20 写入数据

```
*****/
```

```
void WriteOneChar(uchar dat)
```

```
{
    uchar i=0,time;
    for (i=0; i<8; i++)          // 写数据 dat
    {
        DQ =1;
        _nop_();
        DQ=0;
        DQ=dat&0x01;

        for(time=0;time<10;time++)
            ;
        DQ=1;
        for(time=0;time<1;time++)
            ;
        dat>>=1;
    }
    for(time=0;time<4;time++)
        ;
}
```

```
/******
```

Ds18b20 读取温度准备函数

```
*****/
```

```
void ReadyReadTemp(void)
```

```
{
    Init_ds18b20();    // 将 DS18B20 初始化
    WriteOneChar(0xCC); // 跳过读序号列号的操作
    WriteOneChar(0x44); // 启动温度转换

    Init_ds18b20();    // 将 DS18B20 初始化
    WriteOneChar(0xCC); // 跳过读序号列号的操作
```

```
WriteOneChar(0xBE);    // 读取温度寄存器,前两个分别是温度的低位和高位
}
```

```
/******
```

```
    Ds18b20 读取温度函数
```

```
*****/
```

```
float ReadTemperature(void)
```

```
{
```

```
    uchar flag,TH,TL,t_TL,TN,TD;
```

```
    float result;
```

```
    if(Init_ds18b20()==1)
```

```
        return 0;          // 不存在 ds18b20
```

```
    flag=0;
```

```
    ReadyReadTemp();      // 读温度准备
```

```
    TL=ReadOneChar();     // 先读温度值低位    整数位(4)| 小数位(4)
```

```
    TH=ReadOneChar();     // 再读温度值高位    符号位(5)| 整数位(3)
```

```
    if((TH&0xf8)!=0x00)    // 高 5 位, 全 1 -> 负温度, 需要对 TH、TL 处理
```

```
{
```

```
    flag=1;              // 负温度标志
```

```
    TL=~TL;              // 取反加 1
```

```
    TH=~TH;
```

```
    t_TL=TL+1;
```

```
    TL=t_TL;             // 若向高位进 1
```

```
    if(t_TL>255)
```

```
        TH++;
```

```
}
```

```
    TN=TH*16+TL/16;      // 得温度的整数部分
```

```
    TD=(TL%16)*10/16;    // 得温度的小数部分, 1 位小数
```

```
    if(flag == 1)
```

```
        result = -(TN + TD*0.1);
```

```
    else
```

```
        result = TN + TD*0.1;
```

```
    return result;
```

```
}
```

```
/******  
    Bh1750 延时 5 微秒  
*****/  
  
static  
void Delay5us()  
{  
    _nop_();_nop_();_nop_();_nop_();  
    _nop_();_nop_();_nop_();_nop_();  
    _nop_();_nop_();_nop_();_nop_();  
    _nop_();_nop_();_nop_();_nop_();  
}  
/******  
    起始信号  
*****/  
  
void BH1750_Start()  
{  
    SDA = 1;                //拉高数据线  
    SCL = 1;                //拉高时钟线  
    Delay5us();             //延时  
    SDA = 0;                //产生下降沿  
    Delay5us();             //延时  
    SCL = 0;                //拉低时钟线  
}  
/******  
    停止信号  
*****/  
  
void BH1750_Stop()  
{  
    SDA = 0;                //拉低数据线  
    SCL = 1;                //拉高时钟线  
    Delay5us();             //延时  
    SDA = 1;                //产生上升沿  
    Delay5us();             //延时  
}  
/******
```

发送应答信号

入口参数:ack (0:ACK 1:NAK)

*****/

void BH1750_SendACK(bit ack)

```
{
    SDA = ack;           //写应答信号
    SCL = 1;             //拉高时钟线
    Delay5us();          //延时
    SCL = 0;             //拉低时钟线
    Delay5us();          //延时
}
```

*****/

接收应答信号

*****/

bit BH1750_RecvACK()

```
{
    SCL = 1;             //拉高时钟线
    Delay5us();          //延时
    CY = SDA;            //读应答信号
    SCL = 0;             //拉低时钟线
    Delay5us();          //延时

    return CY;
}
```

*****/

向 IIC 总线发送一个字节数据

*****/

void BH1750_SendByte(uchar dat)

```
{
    uchar i;
    for (i=0; i<8; i++)    //8 位计数器
    {
        dat <<= 1;        //移出数据的最高位
        SDA = CY;          //送数据口
        SCL = 1;          //拉高时钟线
    }
}
```



```
        Delay5us();                //延时
        SCL = 0;                    //拉低时钟线
        Delay5us();                //延时
    }
    BH1750_RecvACK();
}

/*****
    从 IIC 总线接收一个字节数据
    *****/

uchar BH1750_RecvByte()
{
    uchar i;
    uchar dat = 0;
    SDA = 1;                        //使能内部上拉,准备读取数据,
    for (i=0; i<8; i++)              //8 位计数器
    {
        dat <<= 1;
        SCL = 1;                    //拉高时钟线
        Delay5us();                 //延时
        dat |= SDA;                 //读数据
        SCL = 0;                    //拉低时钟线
        Delay5us();                 //延时
    }
    return dat;
}

/*****
    对 BH1750 写入寄存器控制命令
    *****/

void Single_Write_BH1750(uchar REG_Address)
{
    BH1750_Start();                //起始信号
    BH1750_SendByte(0x46);          //发送设备地址+写信号
    BH1750_SendByte(REG_Address);   //内部寄存器地址
    BH1750_Stop();                  //发送停止信号
}

/*****
```

初始化 BH1750

```
*****/

void Init_BH1750()
{
    Single_Write_BH1750(0x01);
}

/*****

    读出 BH1750 内部数据

*****/

uint Multiple_read_BH1750(void)
{
    uchar i,BUF[3];
    uint n;
    BH1750_Start();                //起始信号
    BH1750_SendByte(0x46+1);       //发送设备地址+读信号

    for (i=0; i<3; i++)            //连续读取 2 个地址数据，存储中 BUF
    {
        BUF[i] = BH1750_RecvByte(); //BUF[0]存储 0x32 地址中的数据
        if (i == 3)
            BH1750_SendACK(1);      //最后一个数据需要回 NOACK
        else
            BH1750_SendACK(0);      //回应 ACK
    }
    BH1750_Stop();                 //停止信号
    i=200;
    while(i--)
        Delay5us();
    n=BUF[0]*256+BUF[1];
    return n;
}
```

附录五 元件清单

编号	型号、规格	描述	数量
R1	510 Ω	贴片电阻	1
R2、R4	10k Ω	贴片电阻	2
R3	200 Ω	贴片电阻	1
RW1	10k Ω	电位器	1
LED1	LED	发光二极管	1
P1	KF2EDGK2P	接线端子	1
P2	4.7k Ω	排阻	1
P3	GY-302	光强采集集成模块	1
P4	DS18B20	温度传感器	1
LCD1	LCD1602	液晶显示器	1
ISP1	Header5	单排排针	1
J1	COM1	九针串口	1
C1	1000uf	电解电容	1
C2	0.33uf	瓷片电容	1
C3、C5	0.1uf	瓷片电容	2
C4	0.33uf	瓷片电容	1
C6、C9	10uf	电解电容	2
C7、C8	30pf	贴片电容	2
C10-C13	0.1uf	贴片电容	4
S1	双排六脚	自锁开关	1
S2	SW-PB	四脚按键	1
D3	KBP307	整流桥	1
U1	LM7812	三端稳压管	1
U2	LM7805	三端稳压管	1
U3	STC89C52	单片机	1
Q1	S9012	直插三极管	1
LS	TMB12A05	有源蜂鸣器	1
IC1	max232	电平转换器	1