

# 邵阳职业技术学院

## 毕 业 设 计

产品设计	工艺设计	方案设计
		√

设计题目：基于单片机的智能家居防盗系统的设计

学生姓名：郭琛

学 号：201810300875

系 部：电梯工程学院

专 业：机电一体化技术

班 级：机电 1182

指导老师：向浩

二 〇 二 一 年 六 月 一 日

# 目 录

一、引言.....	2
(一) 设计的必要性 .....	2
(二) 发展现状.....	2
二、系统总体设计方案.....	4
(一) 系统基本方案.....	4
(二) 方案选择和论证.....	5
三、硬件设计.....	7
(一) CPU 电路 .....	7
(二) 温度测量电路.....	9
(三) 液晶屏显示电路.....	10
(四) 按键电路.....	11
(五) 电源电路.....	12
(六) 无线告警模块.....	12
四、 软件设计.....	13
(一) Keil 简介.....	13
(二) 主程序设计.....	13
(三) 初始化子程序设计.....	14
六、系统测试.....	17
(一) 测试使用的仪器.....	17
(二) 硬件电路调试.....	17
参考文献.....	19
致谢 .....	20
附录 1: 实物演示图 .....	21
附录 2: 程序清单 .....	22

# 基于单片机的智能家居防盗系统的设计

## [摘要]

本设计采用一块 AT89C51 单片机作为控制核心,将温度传感器获取的温度传输给单片机,通过单片机控制液晶显示屏将当前的温度值显示出来,并利用按键设置布防和撤消防盗报警。当设置布防时,热释电传感器将检测入侵者是否闯入。如果闯入室内,热释电传感器将产生信号,无线发送模块将信号发送出去,接收模块接受到告警信号后解码送给单片机进行处理,单片机控制报警灯亮同时蜂鸣器开始报警,从而达到防盗的作用。

[关键词] 单片机 无线收发 温度采集 智能家居

# 一、引言

## (一) 设计的必要性

当代家庭防范盗贼的主要方式是以安装防盗安全门、防盗锁为主,但是这些防盗设备主要是通过增加相应入室的难度来达到防盗目的。这种传统的安防系统在实际使用时暴露出很多缺点,且防盗效果往往不尽如人意。这种防盗方式不仅影响房屋的美观,而且在发生火灾等事故时,住户难以逃生,威胁人们的生命安全。因此,人们需要有新型的防盗系统——智能家居防盗报警系统。

随着现代科技的快速发展,原来那些简单化、局部化的报警系统已向智能化、集成化的方向发展。当前市场上常规防盗报警系统主要的通信形式为家庭电话、以太网、集群系统等等,但是这些通信方式都存在各自的缺点。

(1) 盗贼在入室前比较容易切断固定电话线或者恶意占线,使其在关键时刻失灵。

(2) 以太网和固定电话一样都面临着线路被剪切的问题,且以太网不易被推广使用。

(3) 集群系统功耗相当大,其网络架设和管理维护费用很高,其正常使用需要购买固定的频点

随着新器件和新技术的出现,家居安防系统将会更加智能化,选择智能家居防盗报警系统,也将为人们提供更加安全可靠的保障。智能家居安防系统将为我们的家庭构筑一道坚固的安全防线,有了它我们的生活将会更加美好。

## (二) 发展现状

我国的智能家居相对于国外起步较晚,尚未形成一定的国家标准。目前,主要采用国外的一些技术产品,但也有一些企业推出了自己的产品,主要有:

(1)e 家庭(海尔),该系列产品以海尔电脑作为控制中心,各种网络家电作为终端设备,海尔移动电话作为移动数字控制中心。海尔在技术上同微软合作,利用微软的 Windows Me 技术和海尔的网络家电,使“e 家庭”已具雏形,已推出了网络洗衣机、网络冰箱、网络空调、网络微波炉等一系列网络家电。

(2) e-home 数字家园(清华同方),该智能家居控制系统是专门针对中国

家庭设计的，遵循国际技术标准，采用嵌入式软、硬件技术，提供网络、网络节点及末端设备。产品以功能模块开发为主，基于国外成熟的智能家居标准之上。其智能家居控制系统主要有以下三个部分：

A 系列：遵循 EIB 协议的家庭控制产品，适用于中高档住宅区。

B 系列：遵循 X-10 协议的家庭控制产品，适用于中档住宅区。

易家三代：配电箱集中安装式家庭控制产品。

国内各大软、硬件机构正在积极的研制、开发更为符合市场的智能化家居设备，以解决当前智能化产品实用性差、使用复杂及产品价格昂贵等缺点，而技术创新性也逐步向国际先进水平靠拢，这样的未来值得期待。

## 二、系统总体设计方案

### (一) 系统基本方案

根据题目设计，系统可以划分为单片机控制器模块，温度测量模块，液晶显示模块，电源模块电路，入侵信号发送模块，无线接收及告警模块电路。模块框图如图 2-1 所示。为了实现各模块的功能，分别做了几种不同的设计方案并进行了论证。

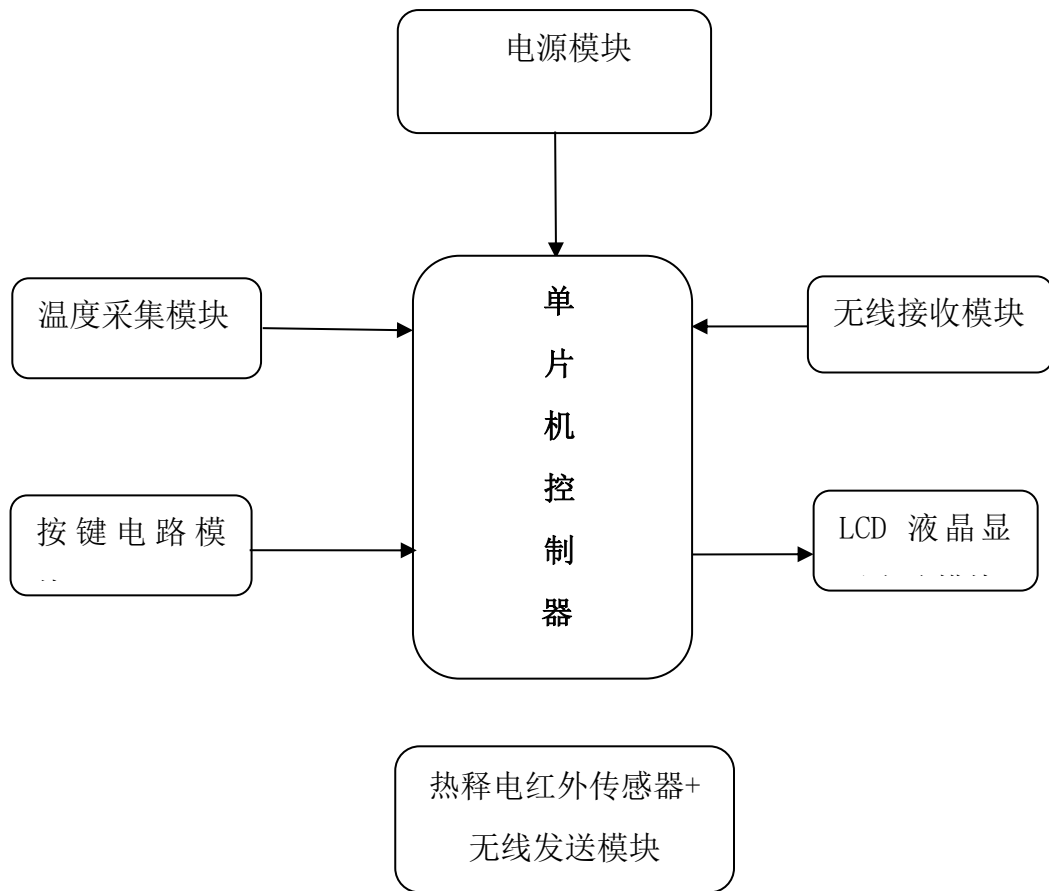


图 2-1 智能家庭控制系统框图

功能说明：

温度实时监控系统主要是将温度传感器采集的温度显示在 LCD 液晶显示屏上。主机上有两个按键，功能分别是布防、清除布防及清除告警。当主机设置了布防功能后，此时热释电传感器监测是否有入侵者闯入，如果有入侵者闯入，则将信号经无线发射模块传输给主机，无线接收模块接收到信号后，识别具体是哪个位置有入侵者闯入，显示在主机上，同时通过蜂鸣器发出告警声并点亮警告灯作为指示。当需要关闭告警时，按清除按键即可。

## （二）方案选择和论证

### 1. 控制器模块

方案一：采用 FPGA（现场可编程门阵列）作为系统的控制器。它将所有的器件集成在一块芯片上，缩小了它的体积，增强了它的稳定性，而且能够利用 EDA 软件进行仿真和调试，以至于能够方便的实施功能的扩展。另外，FPGA 能够实现系统的各种复杂的逻辑功能，规模大，密度高，FPGA 使用的输入方式为并行的，这样就很大的提升了系统的处理速度，合适用来作为大规模的实时系统的控制核心部分。

方案二：采用宏晶公司的 AT89C51 单片机作为系统的 MCU 控制器。它具有算术运算能力强，编程方式自由灵活，算法实现容易，且技术成熟，体积小，具有功耗低，成本低廉等有点，使其在各个领域应用广泛。

综上所述，选择方案二，采用单片机构成系统控制部分。

### 2. 温度检测模块

温度检测模块用于测量温度。系统需要利用测温传感器检测出实时温度，使控制模块做出正确的反应。对于测温传感器的选择有以下几种方案。

方案一：采用基础模拟温度传感器。集成温度传感器 LM35 灵敏度为 10mV 每摄氏度，常温下测量精度控制在  $\pm 5^{\circ}\text{C}$  以内，电流消耗最大也仅有 70UA，自身发热量对精度测量也会造成一定影响，所以其精度高。并且传感器输出的模拟信号还要交给 AD 转换器处理，使得外围电路复杂。

方案二：采用单总线可编程温度传感器测温度。DS18B20 数字可编程温度传感器可测温范围  $-55^{\circ}\text{C} \sim +125^{\circ}\text{C}$ ，在  $-10^{\circ}\text{C} \sim +85^{\circ}\text{C}$  时精度为  $\pm 0.5^{\circ}\text{C}$ 。能够进行编程的分辨率的范围在 9~12 位，相应的能够分辨的温度对应为  $0.5^{\circ}\text{C}$ 、 $0.25^{\circ}\text{C}$ 、 $0.125^{\circ}\text{C}$  以及  $0.0625^{\circ}\text{C}$ ，能够进行高精度的测温。测量结果将会以数字温度信号的形式直接输出，以“一线总线”串行的形式输送到 CPU，同时还可以传输 CRC 校验码，抗干扰以及纠错能力极其的强。所以其外围电路简单，精度很高，且编程也较容易，优势明显。

综上所述，选用方案二，利用单总线可编程温度传感器测量温度。

### 3. 显示模块

方案一：使用传统的数码管显示。其驱动每段数码管大约 10 毫安，所以能耗

很大,静态显示占用的端口多,动态显示又容易出现闪烁感。

方案二：采用 LCD 液晶显示屏来显示温度。由于液晶显示屏具有轻薄短小，低耗电量，显示稳定不闪烁等优点。画面效果好，显示的信息多,分辨率高，抗干扰能力强等特点。

综上所述，选择方案二，采用液晶显示屏显示温度。



### 三、硬件设计

系统整体电路如图 3-1 所示。由于板上芯片需求 5V，所以采用三端稳压芯片 LM7805，输入电源电压范围为 8V—12V，经过分压电阻 R1 分压，防止在稳压芯片上承受过高压降导致发热量过大，输入的电源电压经过线性稳压芯片滤波后输出恒定的 5V 电压，为各芯片提供电源，D1 发光二极管为电源指示灯，当接入电源后，开关打开，D1 发光二极管点亮，表示开始供电。

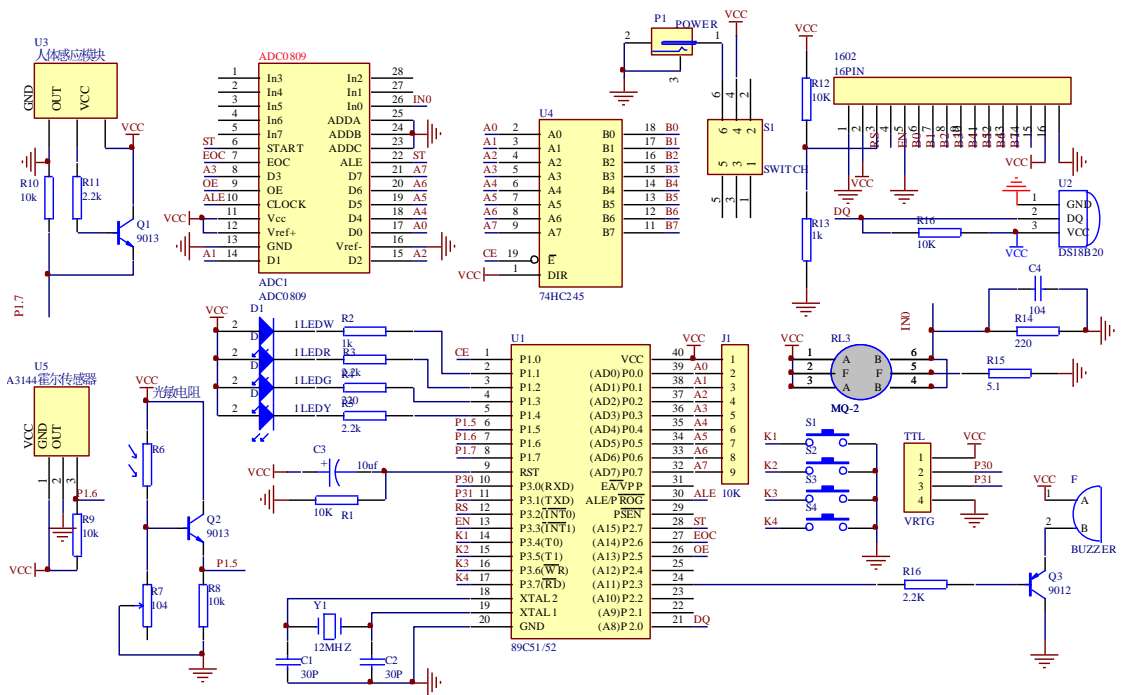


图 3-1 整体电路图

#### (一) CPU 电路

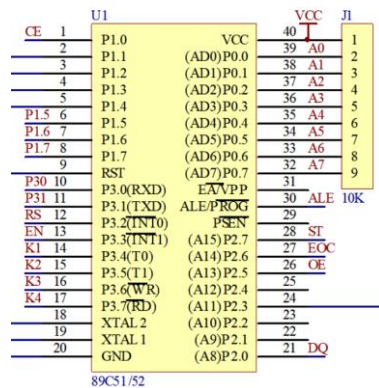


图 3-2 单片机基本电路图

AT89C51单片机是一种性能高、功耗低的微控制器，电路如图3.2所示。该单

片机具有2K 在系统可编程Flash 存储器，芯片上的存储区允许程序存储器在系统可编程。在单芯片上，拥有8 位CPU 和在系统可编程Flash，使得AT89C51可以给许多嵌入式控制提供较高灵活性和较有效的解决方案。AT89C51 的时钟可以降低到0Hz，支持2种软件可选择的节电模式。在低功耗模式下CPU不工作，允许内部数据存储器、定时器/计数器、中断串口、继续工作。在掉电的情况下，内部数据寄存器内容被保存，振荡器被停止，单片机所有的工作会被停止，直到下一个中断的到来或硬件复位为止。

P1 端口内部具有上拉电阻，有 8bit 双向输入输出端口，P1 输出缓冲器能够驱动的 TTL 逻辑电平的个数为 4 个。当闪存编程和验证，双向输入输出端口地址来接收低 8 个字节。其端口引脚还具有第二功能，功能如表 3-1 所示。

表 3-1 P1 端口引脚第二功能

引脚号	第二功能
P1.0	T2（定时器/计数器 T2 的外部计数输入），时钟输出
P1.1	T2EX（定时器/计数器 T2 的捕捉/重载触发信号和方向控制）
P1.5	MOSI（在系统编程用）
P1.6	MISO（在系统编程用）
P1.7	SCK（在系统编程用）

P2 口：P2 口是一个具有内部上拉电阻的 8 位双向 I/O 口，P2 输出缓冲器能驱动 4 个 TTL 逻辑电平。对 P2 端口写“1”时，内部上拉电阻把端口拉高，此时可以作为输入口使用。

P3 口：P3 端口是一个 8 bit 的具有内部上拉电阻的双向输入输出端口，P2 输出缓冲器可以驱动 4 个 TTL 逻辑电平。

表 3-2 P3 端口引脚第二功能

引脚号	第二功能
P3.0	RXD（串行输入）
P3.1	TXD（串行输出）
P3.2	INT0(外部中断 0)
P3.3	INT0(外部中断 0)
P3.4	T0(定时器 0 外部输入)

P3.5	T1(定时器1外部输入)
P3.6	WR(外部数据存储器写选通)
P3.7	RD(外部数据存储器写选通)

## (二) 温度测量电路

数字温度传感器 DS18B20 电路如图 3-3 所示。

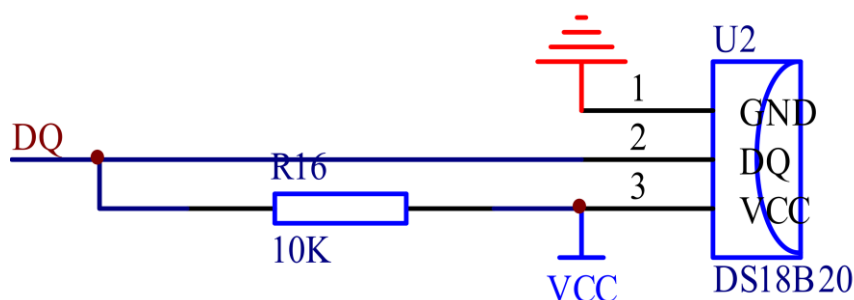


图 3-3 温度传感器 DS18B20 电路

### 1. DS18B20 的主要特性：

1) 可适应 3.0 伏到 5.5V 宽范围内的电压，在寄生电源供电方式下可由数据线供电；

2) 独特的单线接口双向通讯方式，DS18B20 仅需要一条接口线便能和微处理器连接；

3) 许多个该温度传感器可以并联在一个的三线上实现组网多点测温，多点组网功能；

4) 该温度传感器在实际使用的时候可以借助其他的外围元件，使用传感元件和转换电路，使它们集成在形状像一只三极管的集成电路内；

### 2. DS18B20 内部结构：

主要包括 64 位光刻 ROM、温度传感器、非挥发的温度报警触发器高位寄存器和低位寄存器、配置寄存器四部分组成。DS18B20 的外形及内部结构如图 3-4、图 3-5 所示。图 3-4 中引脚分别为：(1)DQ：数字信号输入/输出端；(2)GND：电源地；(3)VDD：供电电源输入端。

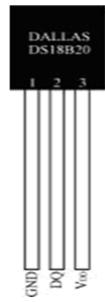


图 3-4 DS18B20 的外形

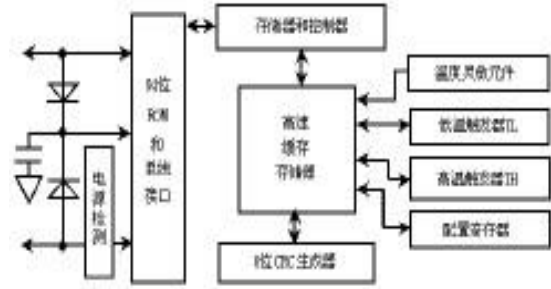


图 3-5 DS18B20 内部结构图

### 3. DS18B20 工作原理：

DS18B20 的读写时序和测温原理都与旧款的 DS1820 相同，只是得到的温度值的位数因分辨率不同，且温度转换时的延时时间由 2 秒钟减为 750 毫秒。如图 3.7 所示，低温度系数晶振的振荡频率不容易受温度的影响，可以用来产生固定频率的脉冲信号再传送到计数器 1。高温系数晶振会受温度的影响，它所产生的信号将会作为计数器的脉冲输入。计数器 1 和温度寄存器被预放在一个负的 55℃ 所相对应的基数值。计数器 1 对低温度系数晶振产生的脉冲信号进行减法计数，当计数器 1 的预置值减到 0 时，温度寄存器的值将被加 1，计数器 1 的预置将重新被装入，计数器 1 重新开始对低温度系数的晶体振荡器产生的脉冲信号进行统计，采用此循环直到计数器 2 计数到 0 时，终止累加温度寄存器值，这个时候温度寄存器中的数值就可以用作所测温度。

### (三) 液晶屏显示电路

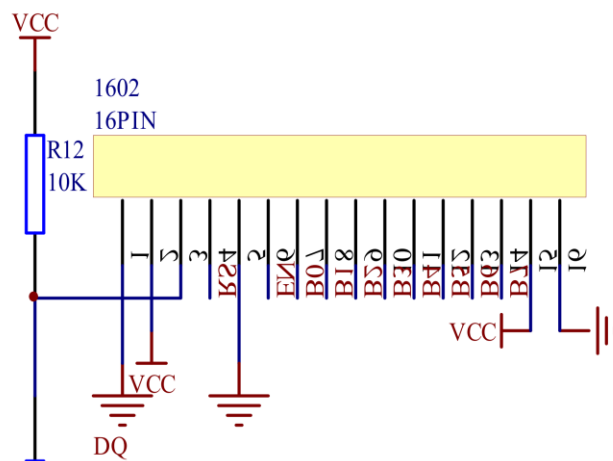


图 3-6 液晶屏显示电路

在本系统中采用液晶屏来显示当前的温度，如图3-6所示。图3-6中采用 FM1602液晶屏来显示，电路中电阻R9、R10是液晶显示屏的对比度调节电阻，调

节该电阻可以得到清晰的显示。为了节约功率，当无按键达到5分钟时候将液晶显示屏的背光灯关闭，采用单片机控制三极管Q1，达到关断液晶显示屏的背光电源。FM1602C实物如图3.7所示，各引脚情况如表3.6所示：

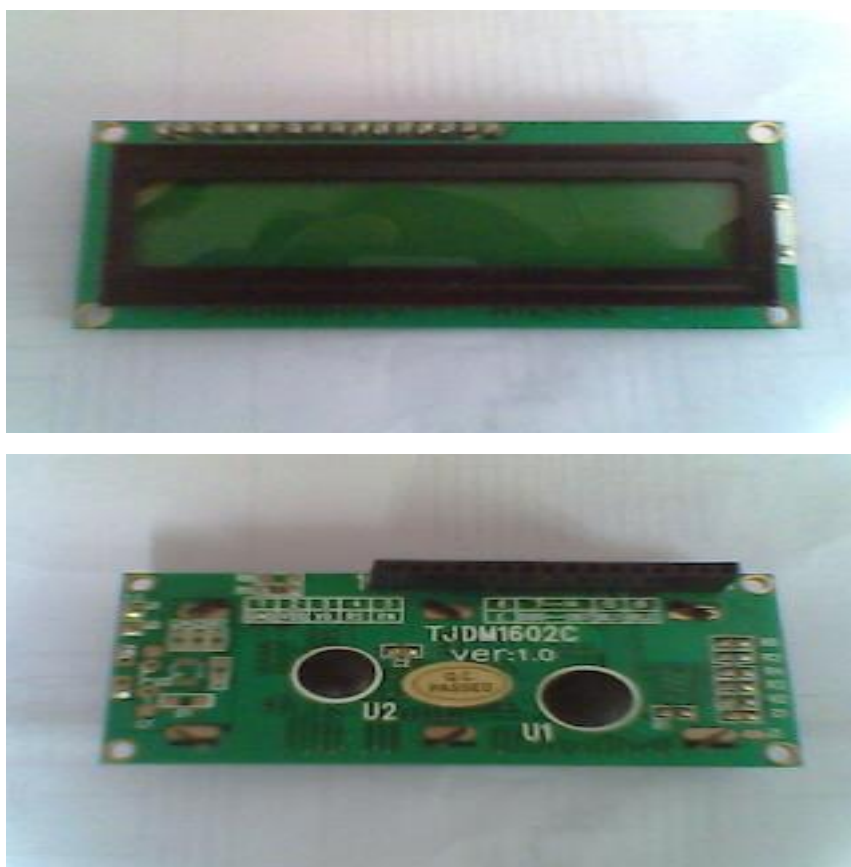


图 3-7 FM1602C 实物

#### (四) 按键电路

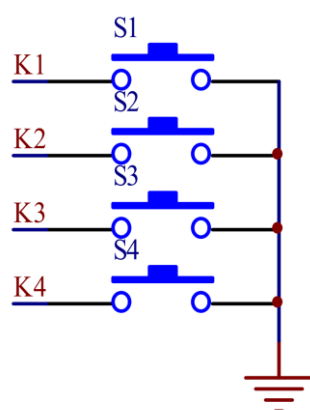


图 3-8 按键电路

按键电路如图 3-8 所示，单片机的 P1 的低 4bit 用来输入数据，当 P1.0~P1.1 没有键按下时，他们将通过电阻接电源将电平拉高，使 P1.0~P1.2 维持高电平，

当 P1.0~P1.2 有键按下时，他们将通过按键将 P1.0~P1.2 接通地，使 P1.0~P1.2 为低电平，使 P1 口能有一个确定的状态，用来输入 1 或者 0。

### （五）电源电路

电源电路+5 电源电路，其原理图如图 3-9 所示。+5 电源主要用来给主电路供电。

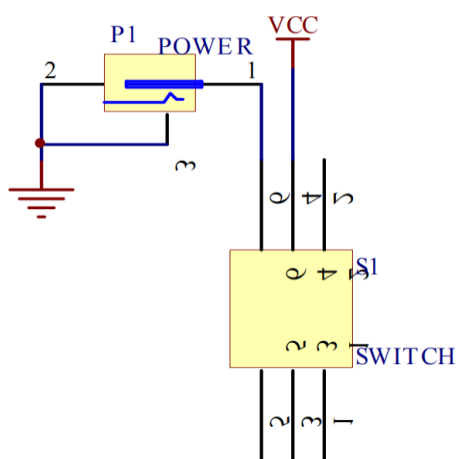


图 3-9 电路原理图

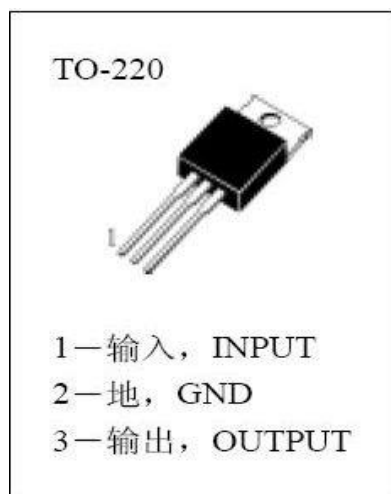


图 3-10 LM7805 外形图及引脚排列

LM7805 的电路原理图如图 3.10 所示，外接电源从 LM7805 的 1 脚输入，电容 C10, C11 用来低频滤波，小电容用来进行高频滤波，分别滤掉输入与输出电路中的高低频干扰，使 LM7805 的 3 脚能稳定输出+5V。

LM7805 系列为三端正稳压电路，能提供多种固定的输出电压，具有过载保护电路内含过流。当有散热片时可以输出电流可达 1 安。虽然是固定稳压电路，但使用外接元件，可获得不同的电压和电流。

## 四、 软件设计

### (一) Keil 简介

Keil C51 是美国 Keil Software 公司出品的 51 系列兼容单片机 C 语言软件开发系统，与汇编相比，C 语言在功能上、结构性、可读性、可维护性上有明显的优势，因而易学易用。Keil 提供了包括 C 编译器、宏汇编、链接器、库管理和一个功能强大的仿真调试器等在内的完整开发方案，通过一个集成开发环境（ $\mu$  Vision）将这些部分组合在一起。

KeilSoftware 公司推出的 uVision3 是一款可用于多种 8051MCU 的集成开发环境 (IDE)，该 IDE 同时也是 PK51 及其它开发套件的一个重要组件。除增加了源代码、功能导航器、模板编辑以及改进的搜索功能外，uVision3 还提供了一个配置向导功能，加速了启动代码和配置文件的生成。此外其内置的仿真器可模拟目标 MCU，包括指令集、片上外围设备及外部信号等。uVision3 提供逻辑分析器，可监控基于 MCUI/O 引脚和外设状态变化下的程序变量。uVision3 提供对多种最新的 8051 类微处理器的支持，包括 AnalogDevices 的 ADuC83x 和 ADuC84x，以及 Infineon 的 XC866 等。

### (二) 主程序设计

系统的软件设计采用 C 语言，对单片机进行编程实现各项功能。程序是在 Window XP 环境下采用 Keil u Vision 3 软件进行编写的。

主程序结构如图 4-1 所示。

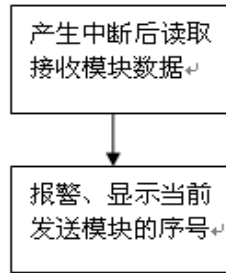
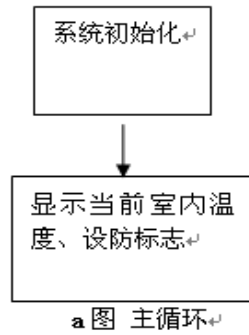


图 4-1 主程序结构图

### (三) 初始化子程序设计

初始化程序如下：

```

uchar Init_DS18B20()
{
    uchar status;
    DQ = 1;
    Delay(8);
    DQ = 0;
    Delay(90);
    DQ = 1;
    Delay(8);
    DQ = 1;
    return status;
}
  
```



## 五、主电路仿真

### (一) Proteus 仿真软件介绍

#### 1. Proteus 简介

Proteus 软件是出自英国 Labcenter electronics 公司的 EDA 工具软件，它集中了先进的原理布图、混合模式 SPICE 仿真，PCB 的设计和全自动布置出线路来完成对一整套设计的原理。Proteus 产品系列中也拥有了人类革命性的 VSM 技术，用户能够对微型控制器附带电子元件共同进行仿真。人们还能够同时采用其他外部设备，例如：LED/LCD、键盘、RS232 终端等来相互交换进行仿真。

#### 2. Proteus 软件的使用

运行 Proteus 程序后，进入软件的主界面。通过左侧工具栏中的 P(从库中选择元件命令)命令，在 Pick Devices 左侧窗口中选择所需元件的关键字，然后放置元件并调整方向和位置以及参数设置，最后进行连线，完成电路原理图后，首先载入 hex 文件，双击 AT89C51 单片机，会出现如下图所示的界面：

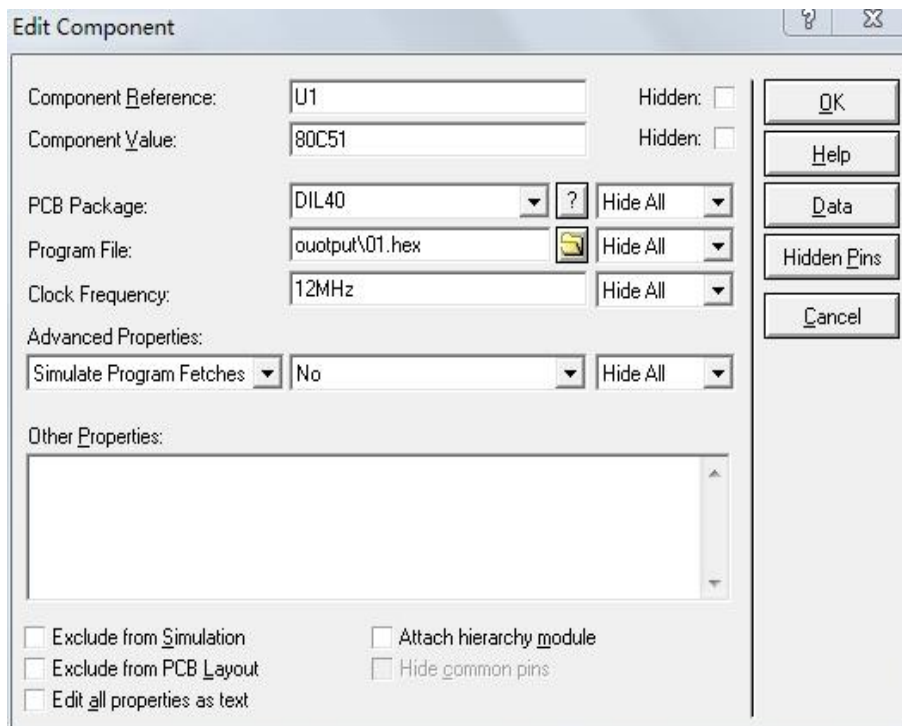


图 5.1 生成烧写文件

点击 载入所需要的 hex 文件。然后就可以进行模拟仿真。



## 六、系统测试

### (一) 测试使用的仪器

测试使用的仪器设备有以下两个：数字万用表和温度计



图 6-1 数字万用表



图 6-1 温度计

### (二) 硬件电路调试

#### 1. 液晶屏显示模块的调试

液晶屏显示模块由于接口比较简单，程序也不用太复杂，所以调试过程一直都很顺利，只是有时候会出现显示乱码现象，经检查是因为接口焊接出现虚焊现象造成的，排除之后显示正常。

#### 2. 画硬件电路时的调试

在原理图画好后生成 PCB 的时候，载入 PCB 时报错，经反复检查，是在层次图中间忘画两张原理图的电源和地，排除故障后 PCB 制作成功完成。

#### 3. 温度传感器的调试

DS18B20 温度传感器其外围简单，对于时序要求严格，因此对晶振的频率要准确，否则在程序上进行调整。否则将会造成信号不稳定。

## 七、成果

经过这次毕业设计，我学会了如何使用单片机进行一些设计，本次毕业设计主要包含有系统设计、单元电路设计、软件设计、主电路系统仿真、系统测试。此外，本次设计还进行了大量的编写程序代码，在编写程序过程中，锻炼了自己的编写代码的能力，同时加深了对计算机语言理论的理解。

但是由于自己的能力有限，加上精力和时间上的有限，本次设计——基于单片机的智能家居防盗系统的设计，存在很多不足的地方，并不是那么完美，我相信，在以后时间精力充裕，自己的能力也比较高的条件下，功能设计会更加完美的。比如说，本次设计只进行了其中一块的设计，主要是防盗报警，以后会将远程控制这块也加上去，使得智能家居防盗系统更完整。

## 参考文献

- [1] 徐建军, 单片机应用及接口[M]. 北京:人民邮电出版社, 2013:8-36
- [2] 王建华, 计算机控制技术[M]. 北京:高等教育出版社, 2012:10-30
- [3]程望斌,刘硕卿,汤姗姗,刘碧篮,全思燕,王曾盛. 基于单片机的智能家居防火防盗系统设计[J]. 湖南理工学院学报(自然科学版), 2019, 32(01):39-42.
- [4]肖思竹,刘宝民. 基于 51 单片机的智能防盗系统设计[J]. 信息通信, 2018, {4} (08):52-53.
- [5]何志文,杨帆,刘帆,焦耀晗. 基于单片机的家用多点防盗系统的设计[J]. 机械管理开发, 2013, {4} (05):73-76.
- [6]向镍锌,邝乙桐,王雪. 基于单片机的智能家居监控系统设计[J]. 电脑知识与技术, 2020, 16(28):198-201.
- [7]马潇宇. 基于单片机的智能家居控制系统设计[J]. 电脑知识与技术, 2020, 16(06):239-240+242.
- [8]阳领. 基于单片机的智能家居加湿系统设计[J]. 科技风, 2018, {4} (27):155.

## 致谢

漫长而又倍感充实的毕业设计阶段即将结束，通过这几十天的学习，我觉得自己的专业知识和独立思考问题的能力有了很大的提高，对我走向社会从事专业工作有着深远的影响。

首先，我感触最深的就是：实践的重要性。这次设计中我做了许多重复性的工作，耽误了很多的时间，但是这些重复性的工作却增强了我的实践能力和动手能力，积累了设计经验。同时也得到一条经验，搞设计不能只在脑子里想它的结构，必须动手，即使你想的很完美，但是到实际的设计过程时，会遇到许多意想不到的问题。

设计的完成，给了我很大的信心：我完全有能力利用自己所学过的知识和技能完成我并不熟悉的任务。在设计过程我更深切的体会到：独立自主是关键，互协作更重要。在完成毕业设计的过程中，向浩老师教了我不公是严谨，认真的工作作风，还有许多是我在学校所不能学到的东西。在此我向向浩老师深表谢意。

附录 1：实物演示图



## 附录 2：程序清单

```
#include"HEAD.H"
main()
{
    unsigned char count=0,wiredata;
    unsigned char keyvalue;
    SP=0x50;
    LCD_SWITCH=0;
    InitSystem();
    curtemp=GetCurTempValue();
    delayms(50);
    curtemp=GetCurTempValue();
    alarmtemp=GetAlarmTemp();
    Display_Temp(curtemp, CurTempType);
    Display_Temp(alarmtemp, AlarmTempType);

    while(1)
    {
        keyvalue=GetKeyValue();
        KeyValueProcess(keyvalue);
        if(keyvalue==Nofind) DisFlag++;
        else
        {
            DisFlag=0;
            LCD_SWITCH=0;

        }
        delayms(10);
        count++;

        if(DisFlag>30000)
        {
            LCD_SWITCH=1;
            DisFlag=0;
        }

        if(count==200)
        {
            count=0;
            curtemp=GetCurTempValue();
            Display_Temp(curtemp, CurTempType);

            if(curtemp>alarmtemp)
            {
                wiredata=P2&0xc3;
                P2=wiredata|0x04;           //发送 D3=1
                WireEnable=0;             //设置 PT2262 发送使能
                TransFlag=1;
            }
        }
    }
}
```





```

    EA=0;
    ISPCR=0x82;
    IFMT=0x03;
    IFADRH=addr;//DPH
    IFADRL=0x00;//,DPL
    SCMD=0x5A;
    SCMD=0x0A5;
    EA=b_ea;
}

void WRITE_BYTE(uchar hi_addr,uchar lo_addr,uchar dat)
{
    bit b_ea;
    b_ea=EA;
    EA=0;
    ISPCR=0x82;
    IFMT=0x02;
    IFADRH=hi_addr;//,DPH
    IFADRL=lo_addr;//,DPL
    IFD=dat;
    SCMD=0x5A;
    SCMD=0x0A5;
    EA=b_ea;
}

uchar READ_BYTE(uchar hi_addr,uchar lo_addr)
{
    bit b_ea;
    b_ea=EA;
    EA=0;
    ISPCR=0x82;
    IFMT=0x01;
    IFADRH=hi_addr;//DPH
    IFADRL=lo_addr;//DPL
    SCMD=0x5A;
    SCMD=0x0A5;
    EA=b_ea;
    return (IFD);
}

/*****
    设置告警温度
*****/
void SetAlarmTemp(char dat)
{
    uchar hi_addr, lo_addr;
    hi_addr=BASSPROGRAM1>>8;lo_addr=0x00;
    ERASE_SECTOR(BASSPROGRAM1);
    WRITE_BYTE(hi_addr, lo_addr, dat);
}

```

```

/*****
    读取告警温度
*****/
uchar GetAlarmTemp(void)
{
    uchar hi_addr, lo_addr, dat;
    hi_addr=BASSPROGRAM1>>8; lo_addr=0x00;
    dat=READ_BYTE(hi_addr, lo_addr);

    return dat;
}

#include"HEAD.H"

sbit DQ = P3^2;                //DS18B20 的数据口位 P3.2

/*****
延时 X 微秒 (STC12C5A60S2@12M)
不同的工作环境, 需要调整此函数
此延时函数是使用 1T 的指令周期进行计算, 与传统的 12T 的 MCU 不同
*****/
void DelayXus(BYTE n)
{
    while (n--)
    {
        _nop_();
        _nop_();
    }
}

/*****
复位 DS18B20, 并检测设备是否存在
*****/
void DS18B20_Reset()
{
    CY = 1;
    while (CY)
    {
        DQ = 0;                //送出低电平复位信号
        DelayXus(240);         //延时至少 480us
        DelayXus(240);
        DQ = 1;                //释放数据线
        DelayXus(60);          //等待 60us
        CY = DQ;               //检测存在脉冲
        DelayXus(240);         //等待设备释放数据线
        DelayXus(180);
    }
}

```

```

/*****
从 DS18B20 读 1 字节数据
*****/
BYTE DS18B20_ReadByte()
{
    BYTE i;
    BYTE dat = 0;

    for (i=0; i<8; i++)          //8 位计数器
    {
        dat >>= 1;
        DQ = 0;                  //开始时间片
        DelayXus(1);             //延时等待
        DQ = 1;                  //准备接收
        DelayXus(1);             //接收延时
        if (DQ) dat |= 0x80;      //读取数据
        DelayXus(60);            //等待时间片结束
    }

    return dat;
}

/*****
向 DS18B20 写 1 字节数据
*****/
void DS18B20_WriteByte(BYTE dat)
{
    char i;

    for (i=0; i<8; i++)          //8 位计数器
    {
        DQ = 0;                  //开始时间片
        DelayXus(1);             //延时等待
        dat >>= 1;               //送出数据
        DQ = CY;
        DelayXus(60);            //等待时间片结束
        DQ = 1;                  //恢复数据线
        DelayXus(1);             //恢复延时
    }
}

BYTE GetCurTempValue(void)
{
    BYTE TPH;                    //存放温度值的高字节
    BYTE TPL;                    //存放温度值的低字节
    BYTE tempvalue;
    DS18B20_Reset();             //设备复位
    DS18B20_WriteByte(0xCC);     //跳过 ROM 命令
    DS18B20_WriteByte(0x44);     //开始转换命令
}

```

```

while (!DQ);          //等待转换完成

DS18B20_Reset();    //设备复位
DS18B20_WriteByte(0xCC); //跳过 ROM 命令
DS18B20_WriteByte(0xBE); //读暂存存储器命令
TPL = DS18B20_ReadByte(); //读温度低字节
TPH = DS18B20_ReadByte(); //读温度高字节
tempvalue=(TPH*256+TPL)/16-8;
return tempvalue;
}
#include"HEAD.H"

#include"HEAD.H"
unsigned char GetKeyValue(void)
{
    unsigned char Keyvalue;
    Keyvalue=Nofind; //首先未发现按键
    if(KeyDec==0)
    {
        delays(260);
        if(KeyDec==0)
        {
            Keyvalue=Key_Dec;
        }
    }

    else if(KeyAdd==0)
    {
        delays(260);
        if(KeyAdd==0)
        {
            Keyvalue=Key_Add;
        }
    }

    else if(KeyConf==0)
    {
        delays(260);
        if(KeyConf==0)
        {
            Keyvalue=Key_Conf;
        }
    }
    return Keyvalue;
}

/*****
    按键值处理程序
*****/

```

```

void KeyValueProcess(unsigned char keyvalue)
{
    if(keyvalue==Nofind) return;

    switch(keyvalue)
    {
        case Key_Add:

            if(alarmtemp<99)    alarmtemp++;Diplay_Temp(alarmtemp, AlarmTempType);

            break;

        case Key_Dec:

            if(0<alarmtemp)    alarmtemp--;Diplay_Temp(alarmtemp, AlarmTempType);

            break;

        case Key_Conf:

            SetAlarmTemp(alarmtemp);

            break;

        default:break;

    }
}

```