

成都大学第十届电子创意设计竞赛

技术报告

项目名称： 基于五一单片机的智能浇花系统

负 责 人： 锁浩杰

所在学院： 电子信息与电气工程学院

专 业： 自动化

班 级： 21级三班

联系电话：

**电子信息与电气工程学院“第二课”科创工作室**

**二0二二年 4 月**

**参赛人员信息**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 题目  名称 | **基于五一单片机的智能浇花系统** | | | |
|  | 姓名 | 所在班级 | 学号 | 项目分工 |
| 负责人 | 锁浩杰 | 2021级自动化三班 | 202110302301 | 项目统筹、代码编写及原理图设计 |
| 成员 | 锁浩杰 | 2021级自动化三班 | 202110302301 | 项目统筹、代码编写及原理图设计 |
|  |  |  | 记录实验过程及数据，报告及ppt制作 |
|  |  |  | PCB设计、电路板焊接及硬件调试 |

**摘 要：**该作品主要适用于家庭花卉绿植养护，通过检测空气温度、土壤湿度并使用LCD602显示屏显示出来，来达到实时检测花卉生长环境状况的能力。土壤湿度检测使用电容式土壤湿度检测器，使用其模拟信号输出端，并使用XPT2046实现模拟信号与数字信号的转化，实现高精度适度检测；在土壤湿度低于预设定温度时，会启用自动浇水模块，实现水分的补充以此达到维持土壤湿度的作用；同时利用DS18B20检测空气温度，达到预设定温度上限后，利用蜂鸣器进行报警；

**关键词**：51单片机；温湿度检测；智能浇花；传感器；系统设计；

**目 录**

[第一章 绪 论 5](#_Toc66795379)

[1.1 目的与意义 5](#_Toc66795380)

[1.2 本文研究的主要内容 5](#_Toc66795381)

[第二章 方案选择 8](#_Toc66795382)

[2.1 简介 8](#_Toc66795383)

[2.2设计总体方案及其论证 8](#_Toc66795384)

[2.3器件选定 8](#_Toc66795385)

[第三章 硬件设计 12](#_Toc66795386)

[3.1主控制电路 12](#_Toc66795387)

[3.2主要模块的电路 12](#_Toc66795388)

[第四章 系统调试 15](#_Toc66795389)

[4.1 软硬件调试 15](#_Toc66795390)

[4.2 系统功能 17](#_Toc66795391)

[第五章 总 结 18](#_Toc66795392)

[附一、实物图 1](#_Toc66795393)8

[附二、程序](#_Toc66795395) 19

第一章 绪 论

### 目的与意义

随着我国经济的不断增长，人民生活过的越来越富足，许多人把目光从过日子转变成了过好日子。于是，越来越多的人开始进行健康生活，或是培养自己的兴趣爱好，也或是希望家里种点绿植，净化空气，舒畅心情，盆景植物逐渐走入大众的家庭。与此同时，社会由于进入高速发展，很多人也会变得繁忙，经常粗心忘记照顾所养殖的盆栽，导致自己蒙受经济上的损失，也没有达到自己想要的目的。于是，一款能够实现自动浇花和温湿度检测的小装置就变得尤为重要，这样的装置不仅能降低培养成本，同时也能给人腾出更多的时间，给人现代科技的福利。同时，这款智能浇花系统有多种花型选择，给与一些没有花卉绿植养护经验的人便利，更好的照顾自己的盆景植物。

### 本文研究的主要内容

**1.2.1 DS18B20温度传感器的理解及使用**

DS18B20 是美信公司的一款温度传感器，单片机可以通过 1-Wire 协议与 DS18B20 进行通信，最终将温度读出。DS18B20是一种单总线数字温度传感器，测试温度范围-55℃-125℃，具有体积小，硬件开销低，抗干扰能力强，精度高的特点。同时，其单总线通信协议也使得信息的传输会更加简单，同时可挂载多个设备，实现更多温度信息的读取，为后续拓展也提供了方便。

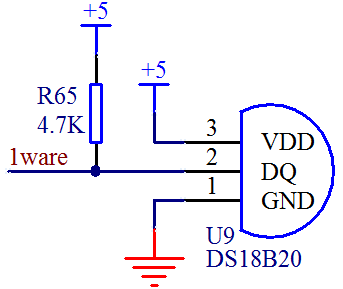


图 1 DS18B20电路原理图

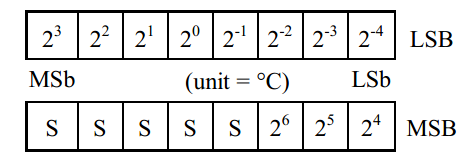


图 2 DS18B20补码储存

**1.2.2 湿度检测**

湿度检测选择使用电容式湿度传感器，该产品是一个简易的水分传感器可用于检测土壤的水分,表面镀镍而不易生锈，延长使用寿命，感应面积宽提高导电性能。模块双输出模式，数字量输出简单，模拟量输出更精确，在这里选择模拟输出，提高湿度检测精度。通过湿度对于电容的改变量而输出模拟信号，利用XPT2046芯片实现模拟信号与数字信号的转变。在这里需要完成输出的数字信号与电压之间关系的转化，以及SPI通信协议的学习。



图 3 土壤湿度传感器

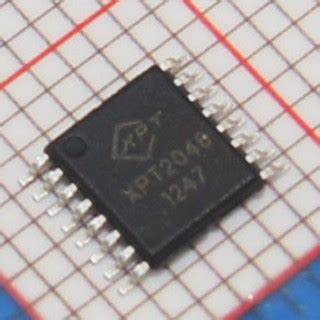


图 4 XPT2046芯片

**1.2.3 太阳能自动充电**

该太阳能板具有高效率、长寿命、高转化等优点，输出电压稳定，耐腐蚀，成本较低。这一方面主要是利用电路知识，设计好电路以达到太阳能自动充电实现绿色能源的使用。



图 5 太阳能充电板

**1.2.4 信息输出**

在采集到需要采集的温度湿度后，需要使用输出设备进行显示，使数据能够被直观的感受到，于是选择较为简单基础的LCD1602，既满足了数据输出的需求，同时也保证了成本在一个可接受的范围内。

第二章 方案选择

本章详细介绍了本次设计的主要任务，以及对于所要实现的内容加以分析，同时详细的介绍元器件的选择，以及各种的特性。

### 2.1 简介

一个以五一单片机为核心的智能浇花系统，需要实现的功能有：

* 能够精确实现对温度湿度的检测和显示。。
* 当环境温度高于（或低于）设定值时，蜂鸣器报警；环境湿度低于设定值时，实现自动浇花。
* 太阳能自动充放电。

### 2.2设计总体方案及其论证

本设计要实现的功能是实现实时显示当前的温度与湿度

（1）温度监控：对温度进行实时检测并显示在显示屏上；

（2）湿度检测：对土壤湿度进行实时检测并显示在显示屏上；

（3）显示：将测量数据实时显示在LCD1602显示屏上；

（4）太阳能板充电：实现太阳能板白天供电且给电池充电，晚上（或者阴雨天气）由电池供电。

依据功能设定，本系统主要分为以下四个模块： （1）温度、湿度采集模块；

（2）数据处理模块；

（3）信息显示模块；

（4）太阳能充电模块。

其中温度采集模块使用的是DS18B02温度传感器，它使用单总线方式，接口简单，而且无需另外校准。分辨率为12bit，完全能够满足植物日常环境温度的检测要求。湿度模块采用电容式湿度传感器，AD/DA数模转换模块（XPT2046）。

数据处理模块使用的是STC89C52单片机来处理数据和控制输出设备。

显示模块是把所有单片机处理的模块显示在LCD1602上。

单片机作为主控制器，主要负责处理ds18b20,XPT2046传来的数据和控制输出设备。

### 2.3器件选定

将单片机用作测控系统时，总要有被测信号输入通道，由计算机拾取必要的输入信息。对于测量系统而言，其核心任务是怎么样获得准确的被测信号；而对测控系统来说，不可缺少的环节是对条件的监测和对被控对象状态的测试，传感器是实现测量与控制的第一环节，是测控系统的关键部分，一切准确的测量和控制都将在传感器对于原始信号的准确可靠的转换和捕捉，工业生产过程的自动化测量和控制，基本主要依赖各种传感器来控制和检测生产过程中的各种量，使系统和设备在最佳状态正常运行，从而保证生产的高质量和高效率

1. 电源部分（水泵）

电源部分（水泵）采用电路设计以达到实现白天太阳能板充电供电，晚上电池供电的效果采用二极管和三极管巧妙地实现这一构想。

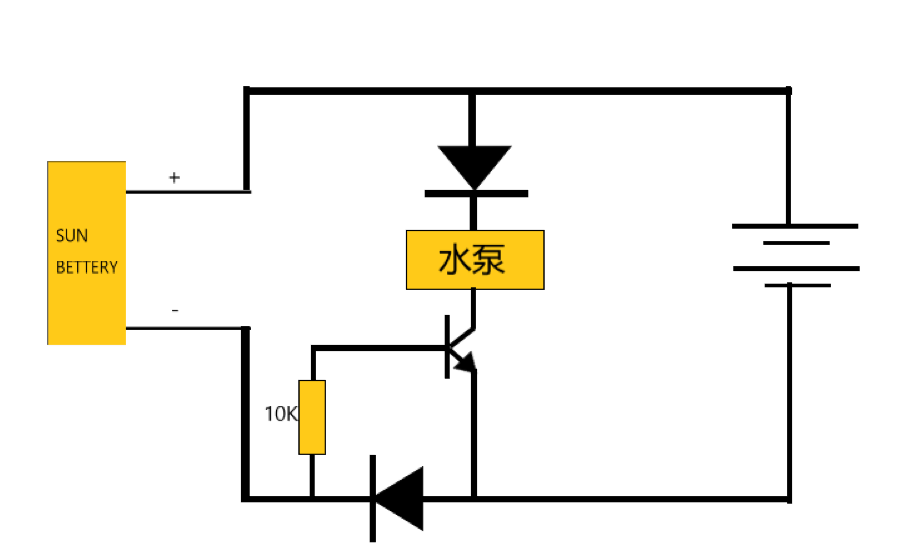


图 6 太阳能板自动充放电电路图

**2.3.1测量部分**

* DS18B20

DS18B20是常用的数字温度传感器，具有体积小，硬件开销低，抗干扰能力强，精度高的特点。DS18B20数字温度传感器接线方便，封装成后可应用于多种场合，如管道式，螺纹式，磁铁吸附式，不锈钢封装式，型号多种多样，有LTM8877，LTM8874等等。

封装后的DS18B20可用于电缆沟测温，高炉水循环测温，锅炉测温，机房测温，农业大棚测温，洁净室测温，弹药库测温等各种非极限温度场合。耐磨耐碰，体积小，使用方便，封装形式多样，适用于各种狭小空间设备数字测温和控制领域。



图 7 DS18B20实物图

* AD/DA转换模块：

XPT2046是一种典型的逐次逼近型模数转换器（SARADC），包含了采样/保持、模数转换、串口数据输出等功能。同时芯片集成有一个2.5V的内部参考电压源、温度检测电路，工作时使用外部时钟。XPT2046可以单电源供电，电源电压范围为2.7V～5.5V。参考电压值直接决定ADC的输入范围，参考电压可以使用内部参考电压，也可以从外部直接输入1V～VCC范围内的参考电压（要求外部参考电压源输出阻抗低）。X、Y、Z、VBAT、Temp和AUX模拟信号经过片内的控制寄存器选择后进入ADC，ADC可以配置为单端或差分模式。选择VBAT、Temp和AUX时可以配置为单端模式；作为触摸屏应用时，可以配置为差分模式，这可有效消除由于驱动开关的寄生电阻及外部的干扰带来的测量误差，提高转换准确度。

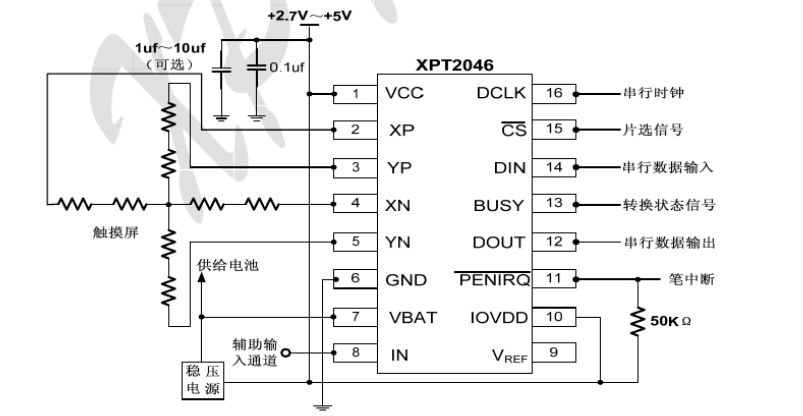


图 8 XPT2046电路图

**2.3.2 输出部分**

* LCD1602显示屏

LCD1602是一种工业[字符](http://www.hqpcb.com/" \t "http://www.elecfans.com/xianshi/_blank)型液晶，能够同时显示16x02即32个字符。LCD1602[液晶显示](http://www.elecfans.com/baike/waijiepeijian/lcd/201803206502)原理 LCD1602液晶显示的原理是利用液晶的物理特性，通过电压对其显示区域进行控制，有电就有显示，这样即可以显示出[图形](http://www.hqpcb.com/)。1602液晶也叫1602字符型液晶，它是一种专门用来显示字母、数字、符号等的点阵型液晶模块。

* 水泵

我们采用直流立式水泵，工作电压为2.5-6V，因防止其工作电流太大且为了更安全的操控水泵工作，需要继电器模块

。****

图 9 立式直流水泵（25-6V）

* 蜂鸣器

蜂鸣器因为没有过多的工作内容，只有温度警报功能，所以选择5V有源蜂鸣器，操作简单，且不需要外部震荡源，容易上手。

****

图 10 5V有源蜂鸣器

第三章 硬件设计

根据以上硬件选择，所以最终选择以**清翔51开发板为**核心进行项目的开发，外接湿度检测模块以达到上述所有功能的正常实现。

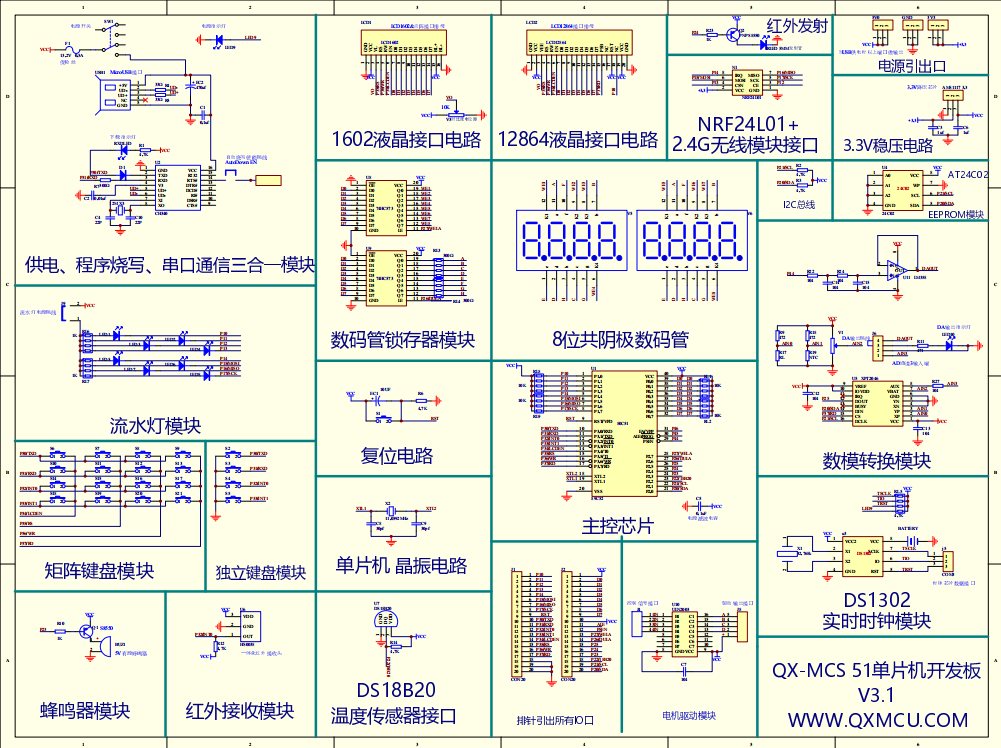


图 11 清翔51开发板电路原理图

### 3.1主控芯片

主控芯片是该系统的核心，是实现上述所有功能的大脑，本次使用的是STC89C52RC芯片。

STC89C52中有一个用于构成内部振荡器的高增益反相放大器，引脚XTAL1和XTAL2分别是放大器的输入端和输出端。这个放大器与作为反馈元件的片外石英晶体或者陶瓷谐振器构成自激振荡器，他们与电容C1，C2接在放大器的反馈电路中构成并联震荡电路，虽然电容没有一个严格的要求，但是电容的大小会轻微影响振荡频率的高低、温度稳定性以及振荡器工作的稳定性（即下述晶振电路）。

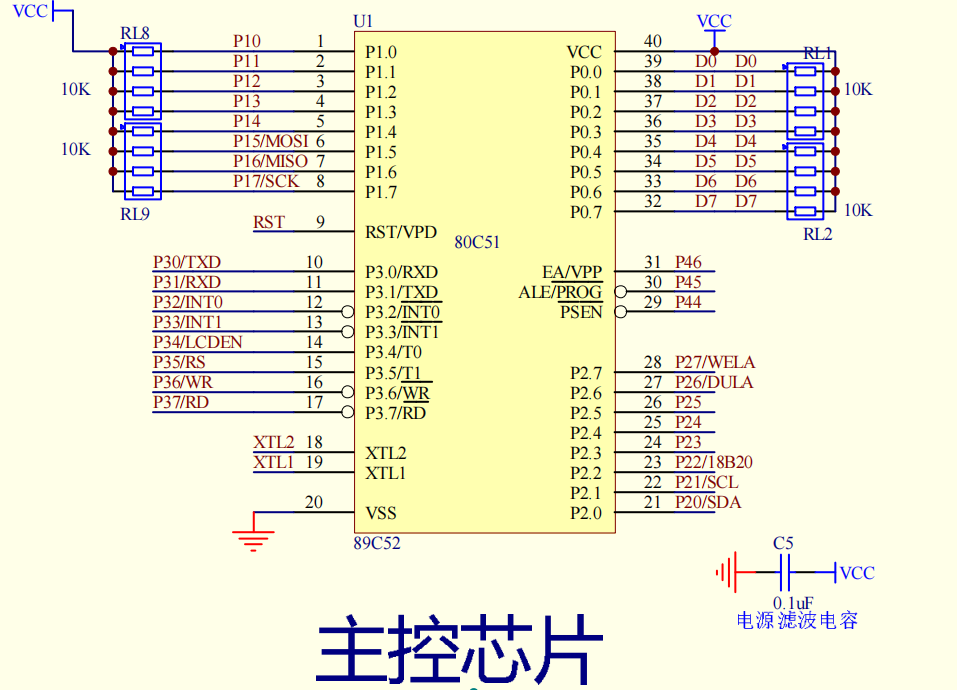


图 12 主控芯片电路

### 3.2主要模块的电路

**3.2.1晶振电路**

在单片机系统里晶振的作用非常大，他结合单片机内部的电路，产生单片机所必须的时钟频率，单片机的一切指令的执行都是建立在这个基础上的，晶振的提供的时钟频率越高，那单片机的运行速度也就越快。

(不同频率的晶振有着不同的作用, 例如：32.768K晶振通常用于时间显示，16MHZ、26MHZ等用于传输信号的)

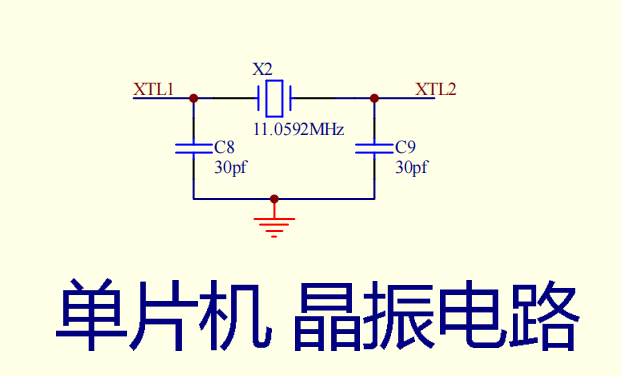


图 13 晶振电路

**3.2.2复位电路**

复位电路是一种用来使电路恢复到起始状态的电路设备，它的操作原理与计算器有着异曲同工之妙，只是启动原理和手段有所不同。复位电路，就是利用它把电路恢复到起始状态。就像计算器的清零按钮的作用一样，以便回到原始状态，重新进行计算。

本次设计采用的是上电复位，当RST引脚上出现了两个周期以上的高电平就会触发内部复位，这里的EA端与复位电路无关，由于数据都放在了内部存储器，所以连接EA只是直接拉高引脚。

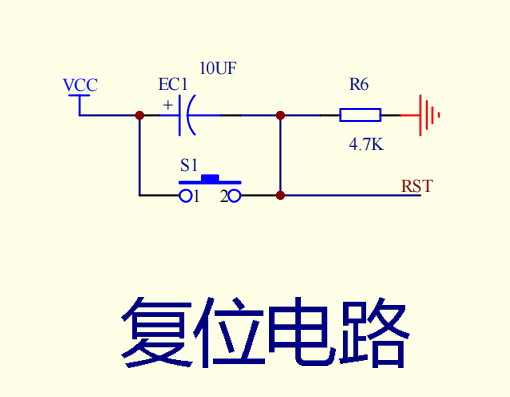


图 14 复位电路

**3.2.3 DS18B20电路**

DS18B20是常用的数字温度传感器，其输出的是数字信号，具有体积小，硬件开销低，抗干扰能力强，精度高的特点。

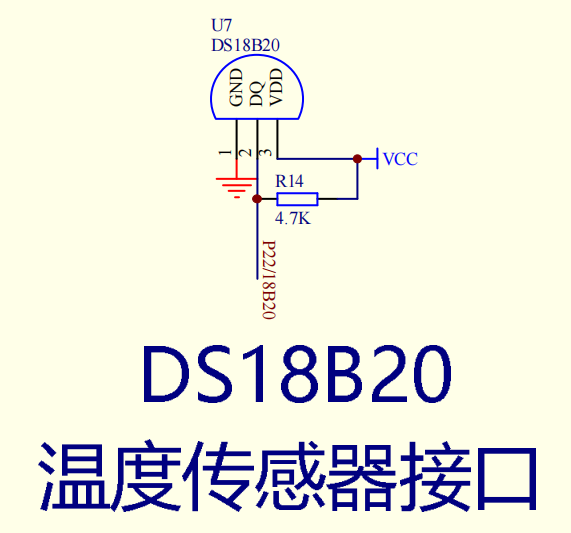


图 16 DS18B20温度传感器电路图

**3.2.4 LCD1602电路**

1602LCD是指显示的内容为16X2，即可以显示两行，每行16个字符液晶模块（显示字符和数字）。市面上字符液晶大多数是基于HD44780液晶芯片的，控制原理是完全相同的，因此基于HD44780写的控制程序可以很方便地应用于市面上大部分的字符型液晶。

同时使用103可调电阻，实现对显示的调节，方便数据读取。

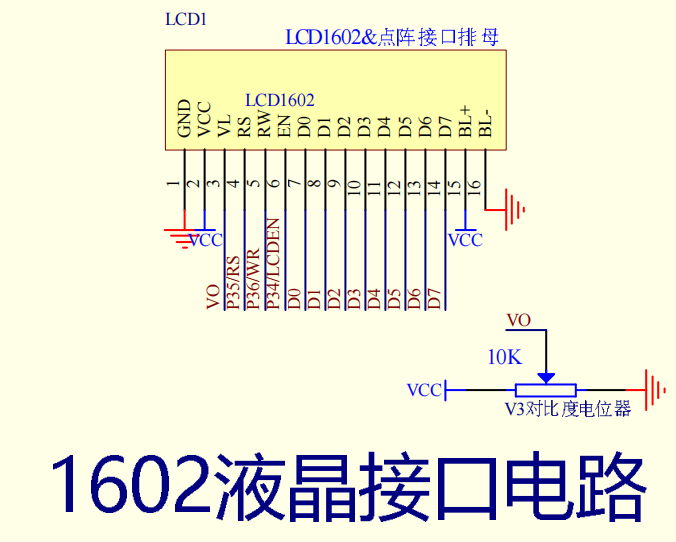


图 15 LCD1602电路图

**图19 电源转换模块**

**3.2.5 AD/DA数模转换电路**

此开发板采用XPT2046芯片进行数模转换，使用SPI通信协议。SPI，是一种高速的，全双工，同步的通信总线，并且在芯片的管脚上只占用四根线，节约了芯片的管脚，同时为PCB的布局上节省空间，提供方便。不过，没有指定的流控制，没有应答机制确认是否接收到数据，所以跟IIC总线协议比较在数据可靠性上有一定的缺陷。

用该模块实现对湿度数据采集的数字量的输出。

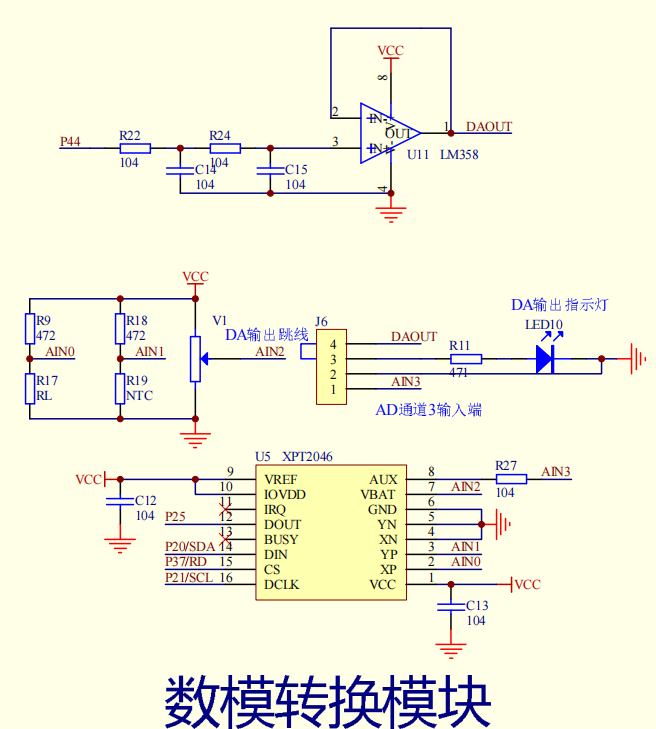


图 17 数模转换模块电路图

第四章 系统调试

软件设计是本次设计中不可缺少的环节，是本次设计能够完成的最重要的环节之一。

该作品主要实现的功能是通过AD/DA模数转化模块读取温度、湿度、PH的值，通过单片机处理数据并发送到LCD2004上显示出来。并且判断值的范围是否偏离于正常范围来实现自动洒水施肥。

首先将各个模块单独调试通，然后逐渐加入后面的程序。

### 4.1 软硬件调试

**4.1.1 软件调试**

（需要有软件流程图、流程图的简要阐述等）

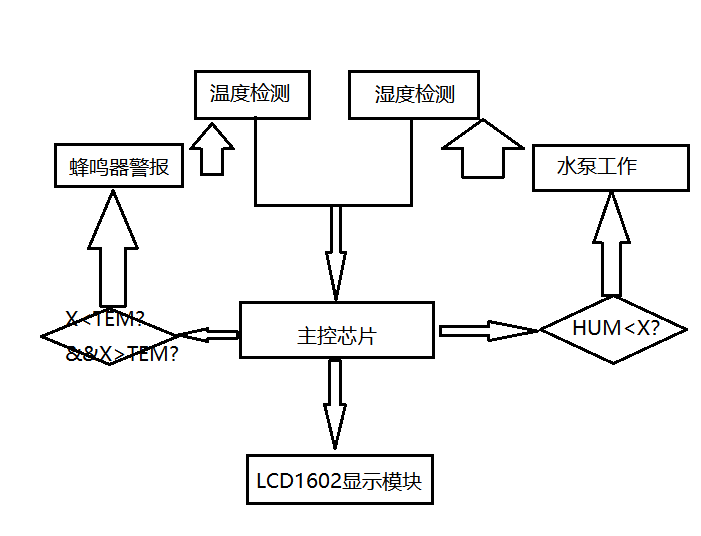


图 18 系统工作逻辑图

如图所示，本系统先对环境中温度，土壤湿度进行检测，然后上传给主控芯片，后将信息传输给LCD1602，实现对数据的可视化输出。如果湿度低于其参考值，则会启动水泵进行工作，直到达到湿度的正常范围；温度超出其参考范围则会进行蜂鸣器警报，以达到警示的作用。

**4.1.2 硬件调试**

（需要有硬件部分的调试步骤与结果）

### 4.2 系统功能

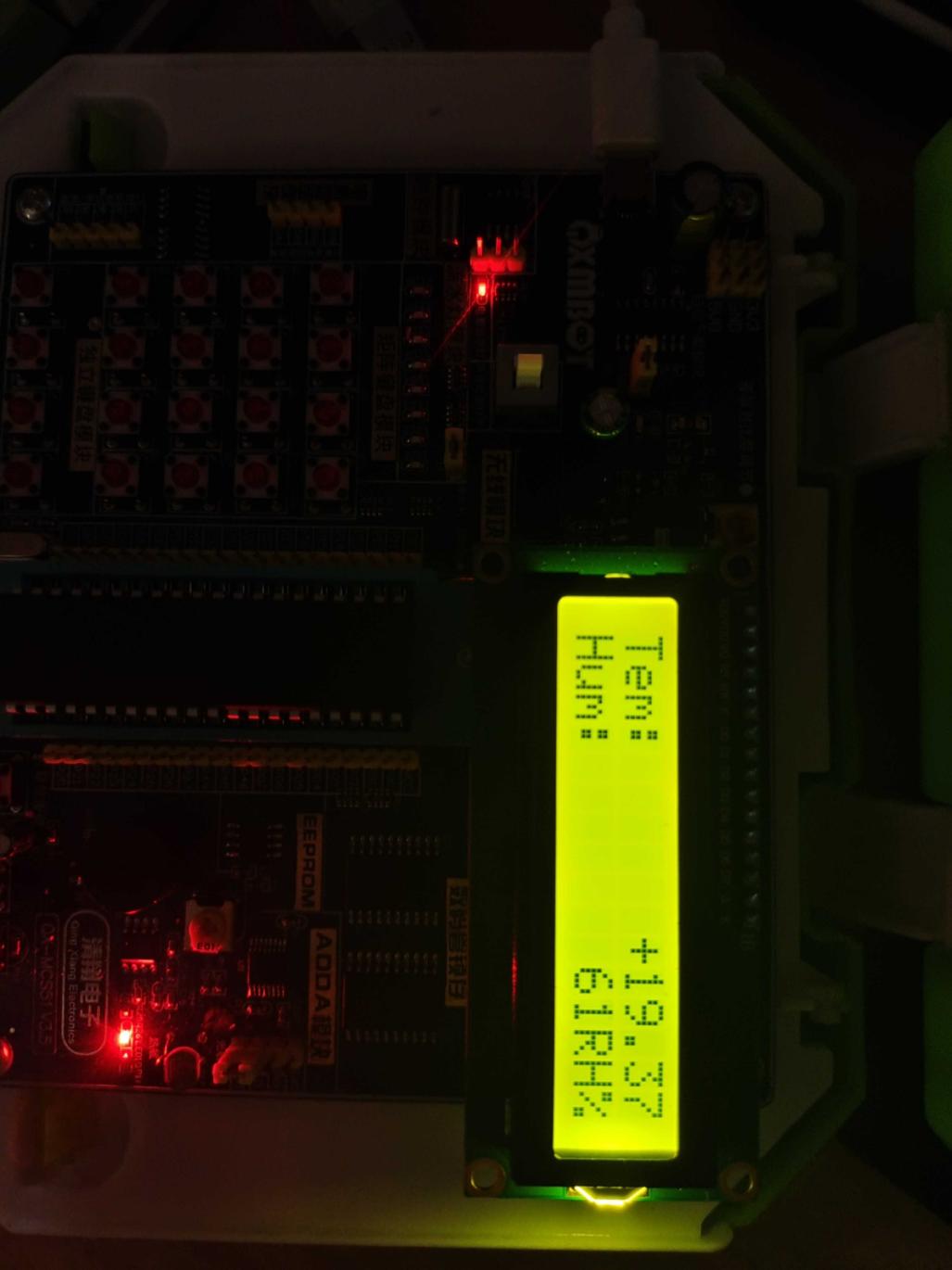
（对系统的整体功能作总结）

通过DS18B20获取温度，ADC数模转换器获取湿度；利用单片机来读取所测数据并进行处理和控制输出设备，根据是否偏离正常值从而启动相应设备。

第五章 总 结

本次项目的实现，使我对于51单片机有了更为深刻的理解，同时多次运用到了文献检索功能，在文献检索的过程中，不仅检索到了我需要找到的知识点，更在过程中了解到了许多新的知识，拓宽了我的视野，也丰富了我的知识面。在过程中，我对于单片机的知识了解的更加深刻，对于电路设计也有了更加深刻的理解，同时，对于一些通信协议，例如：I2C、SPI等通信协议有了一个更深刻的认识，对于以后的项目工作，也有了极大的提升。

附一、实物图



附二、程序

* 主函数部分

#include <REGX52.H>

#include <AD.H>

#include "LCD1602.h"

#include "DS18B20.h"

#include <timer.h>

float T;

unsigned int x,H;

void main()

{

unsigned int i=0;

P2\_7=0;

P2\_6=0;

Timer0Init();

DS18B20\_ConvertT();

Delay(100);

LCD\_Init();

LCD\_ShowString(1,1,"Tem:");

LCD\_ShowString(2,1,"Hum:");

LCD\_ShowString(2,14,"RH%");

while(1)

{

if(i >= 100)

{

i = 0;

H = ReadAD(0xe4);

H = H \* 1.2207 ;

H =(5000-H)/3100.0\*100;

LCD\_ShowNum(2,11,H,3);

}

Delay\_Ms(5);

i++;

if(H<=20)

P1\_1=1;

else

P1\_1=0;

}

}

void temp() interrupt 1

{

static unsigned int T0Count;

TL0 = 0x18;

TH0 = 0xFC;

T0Count++;

if(T0Count>=500)

{

DS18B20\_ConvertT();

T=DS18B20\_ReadT();

if(T<0)

{

LCD\_ShowChar(1,11,'-');

T=-T;

}

else

{

LCD\_ShowChar(1,11,'+');

}

LCD\_ShowNum(1,12,T,2);

LCD\_ShowChar(1,14,'.');

LCD\_ShowNum(1,15,(unsigned long)(T\*10000)%10000,4);

if(T>27)

{

P2\_3=0;

}

else

P2\_3=1;

T0Count=0;

}

}

* AD部分

#include <reg52.h>

#include <intrins.h>

#define MAIN\_Fosc 11059200UL

sbit CS = P3^7;

sbit DCLK = P2^1;

sbit DIN = P2^0;

sbit DOUT = P2^5;

typedef unsigned char INT8U;

typedef unsigned char uchar;

typedef unsigned char u8;

typedef unsigned int INT16U;

typedef unsigned int uint;

typedef unsigned int u16;

uint voltage;

void Delay(unsigned int xms)

{

unsigned char i, j;

while(xms--)

{

i = 2;

j = 239;

do

{

while (--j);

} while (--i);

}

}

void Delay\_Ms(INT16U ms)

{

INT16U i;

do{

i = MAIN\_Fosc / 96000;

while(--i);

}while(--ms);

}

void SPI\_Write(uchar DAT)

{

uchar i;

for(i=0; i<8; i++)

{

DCLK = 0;

if(DAT & 0x80)

DIN = 1;

else

DIN = 0;

DCLK = 1;

DAT <<= 1;

}

}

uint SPI\_Read()

{

uchar i;

uint DAT;

for(i=0; i<12; i++)

{

DAT <<= 1;

DCLK = 1;

DCLK = 0;

if(DOUT)

DAT |= 0X01;

}

return(DAT);

}

uint ReadAD(uchar cmd)

{

uint DAT;

CS = 0;

SPI\_Write(cmd);

DCLK = 0;

\_nop\_();

\_nop\_();

\_nop\_();

\_nop\_();

\_nop\_();

DAT = SPI\_Read();

CS = 1;

return(DAT);

}

* LCD1602部分

#include <REGX52.H>

#define LCD\_DataPort P0

sbit LCD\_RS=P3^5;

sbit LCD\_RW=P3^6;

sbit LCD\_EN=P3^4;

void LCD\_Delay()

{

unsigned char i, j;

i = 2;

j = 239;

do

{

while (--j);

} while (--i);

}

void LCD\_WriteCommand(unsigned char Command)

{

LCD\_RS=0;

LCD\_RW=0;

LCD\_DataPort=Command;

LCD\_EN=1;

LCD\_Delay();

LCD\_EN=0;

LCD\_Delay();

}

void LCD\_WriteData(unsigned char Data)

{

LCD\_RS=1;

LCD\_RW=0;

LCD\_DataPort=Data;

LCD\_EN=1;

LCD\_Delay();

LCD\_EN=0;

LCD\_Delay();

}

void LCD\_SetCursor(unsigned char Line,unsigned char Column)

{

if(Line==1)

{

LCD\_WriteCommand(0x80|(Column-1));

}

else if(Line==2)

{

LCD\_WriteCommand(0x80|(Column-1+0x40));

}

}

void LCD\_Init()

{

LCD\_WriteCommand(0x38);

LCD\_WriteCommand(0x0c);

LCD\_WriteCommand(0x06);

LCD\_WriteCommand(0x01);

}

void LCD\_ShowChar(unsigned char Line,unsigned char Column,char Char)

{

LCD\_SetCursor(Line,Column);

LCD\_WriteData(Char);

}

void LCD\_ShowString(unsigned char Line,unsigned char Column,char \*String)

{

unsigned char i;

LCD\_SetCursor(Line,Column);

for(i=0;String[i]!='\0';i++)

{

LCD\_WriteData(String[i]);

}

}

int LCD\_Pow(int X,int Y)

{

unsigned char i;

int Result=1;

for(i=0;i<Y;i++)

{

Result\*=X;

}

return Result;

}

void LCD\_ShowNum(unsigned char Line,unsigned char Column,unsigned int Number,unsigned char Length)

{

unsigned char i;

LCD\_SetCursor(Line,Column);

for(i=Length;i>0;i--)

{

LCD\_WriteData(Number/LCD\_Pow(10,i-1)%10+'0');

}

}

* DS18B20部分

#include <REGX52.H>

#define DS18B20\_SKIP\_ROM 0xCC

#define DS18B20\_CONVERT\_T 0x44

#define DS18B20\_READ\_SCRATCHPAD 0xBE

sbit OneWire\_DQ=P2^2;

unsigned char OneWire\_Init(void)

{

unsigned char i;

unsigned char AckBit;

OneWire\_DQ=1;

OneWire\_DQ=0;

i = 247;while (--i);

OneWire\_DQ=1;

i = 32;while (--i);

AckBit=OneWire\_DQ;

i = 247;while (--i);

return AckBit;

}

void OneWire\_SendBit(unsigned char Bit)

{

unsigned char i;

OneWire\_DQ=0;

i = 4;while (--i);

OneWire\_DQ=Bit;

i = 24;while (--i);

OneWire\_DQ=1;

}

unsigned char OneWire\_ReceiveBit(void)

{

unsigned char i;

unsigned char Bit;

OneWire\_DQ=0;

i = 2;while (--i);

OneWire\_DQ=1;

i = 2;while (--i);

Bit=OneWire\_DQ;

i = 24;while (--i);

return Bit;

}

void OneWire\_SendByte(unsigned char Byte)

{

unsigned char i;

for(i=0;i<8;i++)

{

OneWire\_SendBit(Byte&(0x01<<i));

}

}

unsigned char OneWire\_ReceiveByte(void)

{

unsigned char i;

unsigned char Byte=0x00;

for(i=0;i<8;i++)

{

if(OneWire\_ReceiveBit()){Byte|=(0x01<<i);}

}

return Byte;

}

void DS18B20\_ConvertT(void)

{

OneWire\_Init();

OneWire\_SendByte(DS18B20\_SKIP\_ROM);

OneWire\_SendByte(DS18B20\_CONVERT\_T);

}

float DS18B20\_ReadT(void)

{

unsigned char TLSB,TMSB;

int Temp;

float T;

OneWire\_Init();

OneWire\_SendByte(DS18B20\_SKIP\_ROM);

OneWire\_SendByte(DS18B20\_READ\_SCRATCHPAD);

TLSB=OneWire\_ReceiveByte();

TMSB=OneWire\_ReceiveByte();

Temp=(TMSB<<8)|TLSB;

T=Temp/16.0;

return T;

}

* 定时器部分

void Timer0Init(void)

{

TMOD &= 0xF0;

TMOD |= 0x01;

TL0 = 0x66;

TH0 = 0xFC;

TF0 = 0;

TR0 = 1;

ET0=1;

EA=1;

PT0=0;

}