**《微机控制技术》课程设计报告**

|  |  |
| --- | --- |
| 单片机液位控制系统 | |
| 学生姓名 | 陈伟钿 |
| 学 号 | 201211921404 |
| 手机号码 | 13690077489 |
| 所在学院 | 理学院 |
| 所在班级 | 电科1124班 |
| 任课教师 | 王淑青 |
| 提交时间 | 2015年 12 月 11日 |

## 一、设计要求

输出控制系统的框图，硬件设计图，软件流程图，说明采用的相关算法。

## 二、制定方案

#### 1．实验方案

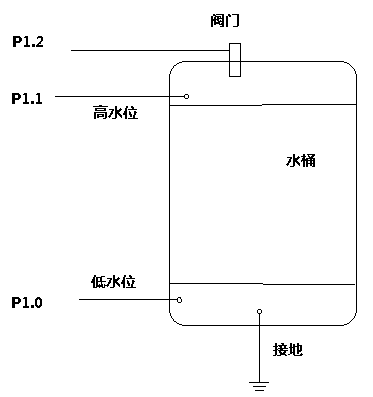
控制液位的系统，主要的功能便是对水箱或者是水桶储水容器中的储水量进行判断，当达到一定的水量的时候，控制阀门停止送水。当水量过低的时候，控制其阀门防水。所以首先要有一部分检测液位的模块，改模块主要是检测液位，并向单片机发送液位信号。然后就是单片机部分，单片机通过判断输入信号，对信号进行判断和处理，然后通过控制阀门进水与出水。如图1所示，因为一般的水都含有一定杂质，所以具有导电性。利用这一特性，如果水位到达低水位，而不到高水位，那么P1.0低电平，P1.1高电平,如果水位到达高水位，那么P1.1与P1.0都会输出低电平，如果水位低于低水位，那么P1.0与P1.1都会输出高电平。根据这个原理，我们可以通过检测P1.0与P1.1的电平高低，从而实现对阀门的控制，从而达到控制水位的目的。

图1

#### 2.系统总体框架图

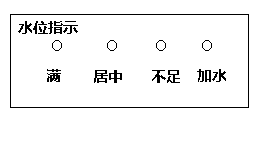
由图3系统框架图可以看出，该系统主要模块由水位检测模块，阀门模块，指示模块组成。水位检测模块主要检测水位的高低，然后单片机通过读取对应端口(P1.1,P1.0)的状态，然后判断对应状态对应的水位，控制阀门(P1.2)的开关。并在指示模块上面对水位状态，工作状态进行显示。指示模块由4个led组成，第一个指示灯表示当前水位满了，第二个指示灯表示水位居中也就是当前的水位介于不足与满之间，第三个指示灯表示水位不足。就是当前水位低于最低水位。最后一个是加水的显示，主要是当前正在加水的时候，会显示加水。水位指示器如图2所示。

图2

系统的总体框架图如下：

**指示模块**

**水位检测模块**

**时钟电路**

**STC89C52**

**阀门模块**

**复位电路**

图3

## 三、电路原理图

### a)原理图概述

这是液位控制系统的完整原理图。

可以看到，该系统严格按照前面的要求进行设计的。

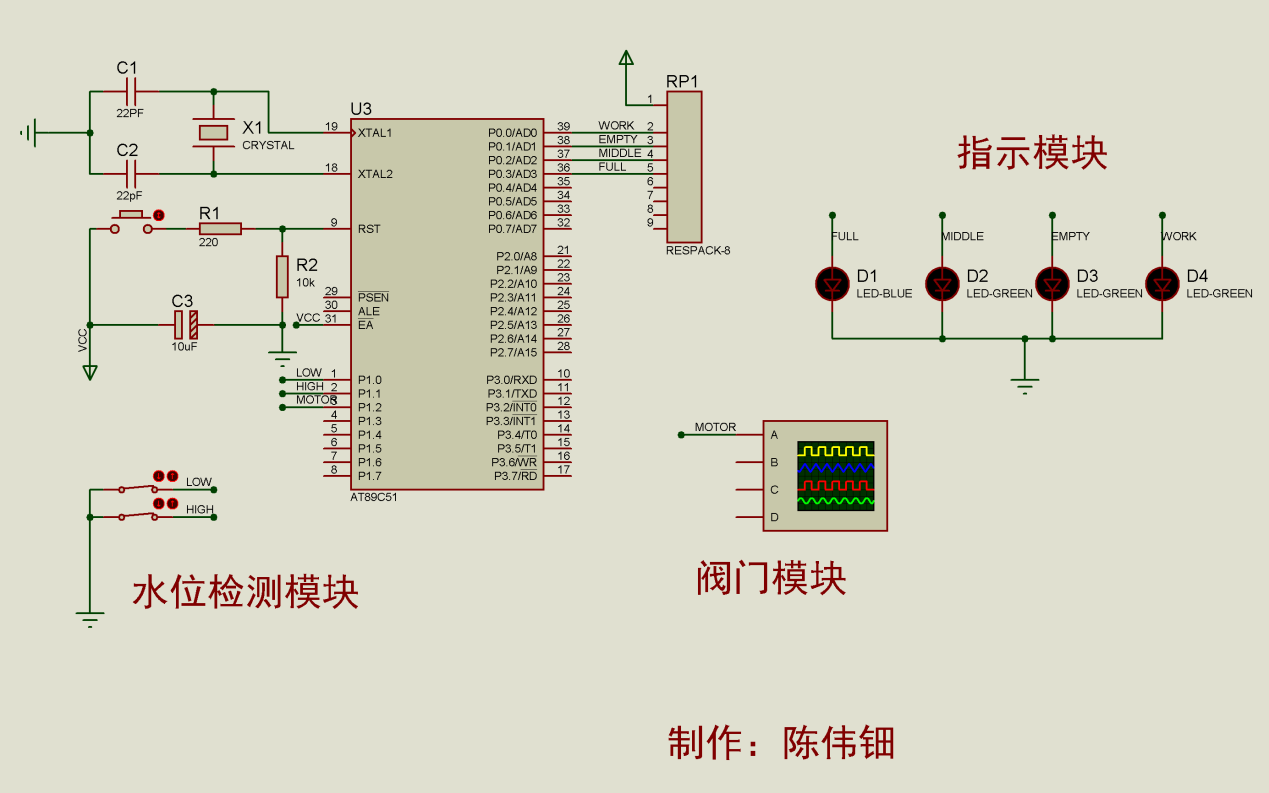


图4

### b)模块功能分析

#### 1. 复位模块

复位模块，满足单片机上电的时候，通过Vcc(+5V)电源给电容C充电加给RST引脚一个短暂的高电平信号，此信号随着Vcc对电容的充电过程而逐渐回落，即RST的高电平持续时间取决与电容C充电时间，因此为了保证系统能可靠复位，RST引脚上的高电平必须大于复位所要求的高电平事件。除了上电复位之外，还需要人工复位功能。按键复位通过RST端经过2个电阻对电源Vcc接通分压产生的高电平来实现。一般来说，单片机的复位速度比外围I/O接口电路要快一些。所以为了保证系统可靠复位，在单片机应用程序的初始化程序中要安排一段复位延迟时间，以保证单片机与外围I/O接口电路都能够可靠的复位。复位电图如图5所示。

图5

#### 2．时钟模块

C51单片机各种外围部件的运行都是以时钟控制信号为基准，有条不紊，一拍一拍的工作，因此，时钟电路的质量也直接影响单片机系统的稳定性，常用的时钟电路有两种方式，一种是内部时钟，一种是外部时钟。我们这个系统使用的是外部时钟。

外部时钟是以现有的外部振荡器产生时钟脉冲信号，常用与多片单片机的工作，其电路图如图6所示。

图6

#### 3.水位检测模块

水位检测模块主要是通过2个开关来模拟，如右图所示。开关的状态分别对应水位的状态，如果水位漫过最高带你，相当与LOW，与HiGH2个接口都接地，所以我们这里则表示为2个开关闭合，当水位达到居中的状态的时候，相当于HIGH开路，LOW短路，LOW为低电平，HIGH为高电平。当水位低于低水位，那么此时LOW，与HIGH都是出于开路的状态。所以此时LOW，与HIGH均为高电平。当然，这个电路也存在一个不符合逻辑的状态，那就是LOW被悬空，HIGH被短路，此时这种状态必定是系统异常所导致的，此时，我将单片机的指示器都显示位高电平，即4个指示灯都亮，表示异常。

图7

#### 4.指示模块

指示模块仅仅由4个LED指示灯组成，在这里，我采用了共阴极的方式连接。因为使用到了单片机的P0口，所以要接上拉电阻。因为P0口是没有办法输出高电平的，只有高阻态，所以为了使其输出高电平，必须接上拉电阻 。如图8所示。为了表示效果更佳，我使用了不同颜色的LED灯。

#### 5.阀门模块

图8

阀门模块由于不好仿真，所以我使用示波器代替，通过示波器上脉冲的变化，表示控制端的电平的升高与降低。高电平表示阀门打开，低电平表示阀门关闭。

### c)PCB印刷电路图

图9

### 

图10

注 PCB中并没有引出P1.1与P1.0与P1.2引脚。

## 四、代码实现部分

a)软件流程图

开始

初始化

采集P1口状态

是

关闭阀门，指示

P1=0x00

否

是

异常指示

P1=0x01

否

是

居中指示

P1=0x02

否

是

打开阀门，指示

P=0x03

图11

b)源代码

|  |
| --- |
| main.c文件 |
| 源代码 |
| /\*-----------------------------------------------  名称：main.c  编写：陈伟钿  日期：2015.12.09  修改：无  内容：晶振频率 11.0592MHz  ------------------------------------------------\*/  #include <reg51.h>  void DelayMs(unsigned char t); //延时函数的声明  void DelayUs2x(unsigned char t);  sbit Motor=P1^2; //阀门控制端  unsigned char dat; //读取P1.0,P1.1数据  void main(){  //P1=1;//使用P1口作为IO口读取数据，必须先写1  Motor=0;  DelayMs(2);  while(1){  dat=P1&0x03; //读取P1.0,P1.1口  switch(dat){  case 0x00: //P1.0=0,P1.1=0，即水位满  Motor=0; //水位将满的时候关闭阀门  P0=0x08;  break;  case 0x01: //P1.0=1,P1.1=0，异常状态  P0=0xff; //所有指示灯亮，表示异常状态  break;  case 0x02: //P1.0=0,P1.1=1，水位适中  P0=0x04;  break;  case 0x03: //P1.0=1,P1.1=1，水位不足  P0=0x03;  Motor=1; //开启阀门  break;  }  }  }  /\*------------------------------------------------  mS延时函数，含有输入参数 unsigned char t，无返回值  unsigned char 是定义无符号字符变量，其值的范围是  0~255 这里使用晶振12M，精确延时请使用汇编  ------------------------------------------------\*/  void DelayMs(unsigned char t)  {    while(t--)  {  //大致延时1mS  DelayUs2x(245);  DelayUs2x(245);  }  }  void DelayUs2x(unsigned char t)  {  while(--t);  } |

## 五、用Proteus仿真

启动Proteus进行仿真，通过拨动开关的闭合，断开，模拟水位的不同状态。通过判断状态，对指示器进行指示。

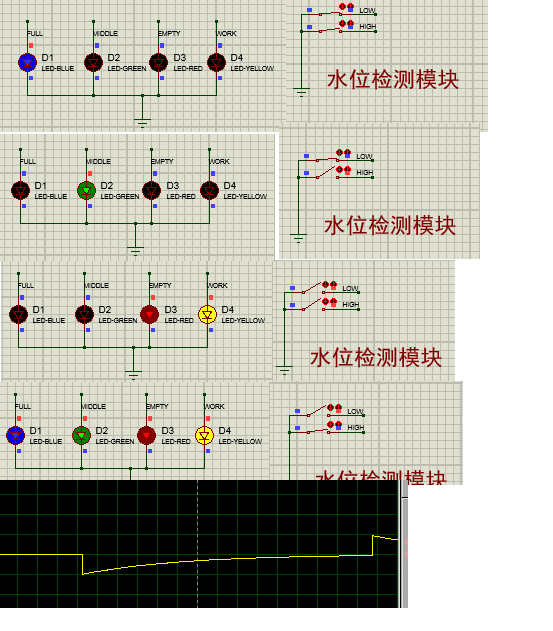


图12

## 六、心得和体会

通过这次微机控制技术的课程设计，我深刻的体验到了单片机的魅力，对于其实际价值有了更加深刻的理解，相对于课堂上的理论知识，这是一门需要注重实践的课程。对自我动手能力的提升有很大的帮助，同时在设计中遇到问题。发现自己掌握知识的不足。对以后的发展目标有了明确的方向。