**西安电子科技大学**

**微机原理课设 课程实验报告**

**实验名称 步进电机开环控制系统设计**

计算机科学与技术 学院 班

成 绩

姓名 学号

同作者

实验日期 年 月 日

实验地点 实验批次

|  |
| --- |
| 指导教师评语：  指导教师：  年 月 日 |
| **实验报告内容基本要求及参考格式**  一、实验目的  二、实验所用仪器（或实验环境）  三、实验基本原理及步骤（或方案设计及理论计算）  四、实验数据记录（或仿真及软件设计）  五、实验结果分析及回答问题（或测试环境及测试结果） |

**一、实验目的**

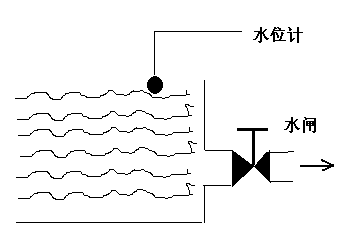
1.掌握微机系统总线与各芯片管脚连接方法，提高接口扩展硬件电路的连接能力。

2.加深对 A/D 和并行接口芯片的工作方式和编程方法的理解。

3.搞懂步进电机的工作原理及控制方式，掌握开环控制系统的设计思路和实现方法。

**二、实验内容**

手动调节电位器旋钮（0V-5V），通过ADC0809模拟输入水库水位 0米-50 米,CPU收到水位信号后， 根据水位高度控制步进电机（水闸）进行调节。



水库水位模拟控制示意图

**三、系统功能与设计要求**

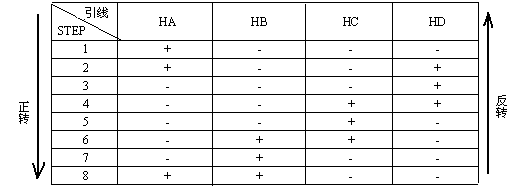
1.基本功能要求

基本功能要求手动调节电位器旋钮，步进电机根据水位实时调节水闸。设水闸全部打开需要逆时针旋转10圈（10 x 360）度。随着上游进入水库的水流量变化，水库水位不断变化（手动调节电位器旋钮），每到一定高度，步进电机顺时针（关）或逆时针（开）旋转一定的角度调节水闸开启程度，从而控制水库水位在 10-50米之间。调节精度控制在±5%。调节规律如下：

水位与水闸控制开启控制表

|  |  |  |
| --- | --- | --- |
| 水位高度 H（米） | 水闸开启程度（%） | 指示灯 |
| 0≤H<10 | 0%（0 圈） | 关闸蓄水，水位超低报警（黄LED闪烁） |
| 10≤H<20 | 20%（2 圈） | 正常调节（绿LED亮） |
| 20≤H<30 | 40%（4 圈） | 正常调节（绿LED亮） |
| 30≤H<40 | 60%（6 圈） | 正常调节（绿LED亮） |
| 40≤H<50 | 80%（8 圈） | 正常调节（绿LED亮） |
| H≥50 | 100%（10 圈） | 开闸泄洪，水位超高报警（红LED闪烁） |

步进电机采用四相八步控制，开关顺序如图：



步进电机控制图

注：当实验结束要立即关闭电源，否则一直停留在某一相上会使电机发热。

2.发挥部分

（1）增加速度调节功能。水位在10-40米期间，步进电机中速转动，水位低于10（水位过低）或高于40米（水位过高）时，步进电机高速转动。

（2）增加实时水位显示。用数码管DLED高两位显示当前水位（00-50 米）。

（3）增加水闸开启程度显示。用数码管DLED低两位实时显示水闸开启程度（00-10圈）

**四、实验环境**

接口实验箱（8086芯片）、星研电子软件平台

**五、方案设计**

1.主要设计思路

(1) 手动调节电位器旋钮(0-5V)，模拟水位变化（0-50米）

(2) 电位器输出到ADC0809进行模数转换（0-255），通过编程将模拟量实时采集并转变为对应的数字量（0-50），CPU 通过查询数字量可计算出实际的水位。

(3) 如果水位过低则快速关闭水闸。

(4) 如果水位过高则快速打开水闸。

(5) 如果在正常范围内，根据调节规律查出相应的水闸最终开启程度，并与上次开启程度比较，计算出需要调节步进电机旋转的圈数、方向，然后控制步进电机进行相应转动，调节水闸，从而达到控制水位的目的。

(6) LED 指示灯可以通过使用基本并行 I/O 接口（74LS273）控制显示，DLED 数码管可以采用系统提供的相应模块，控制可以通过 8255 可编程并行接口，如：PA口的底四位（PA0-PA3）控制步进电机， PB口和 PC口的低四位（PC0-PC3）控制数码管的段、位。

2.实验器件接线图见附件

**六、心得体会以及课后建议**

在本次课设中，我主要负责程序调试修改。在实验过程中，印象最深的部分是学习怎么用C语言操作步进电机的工作。通过查询实验手册和附带的例程，了解了步进电机的控制原理。经过了这三天的实验，我对单片机上的C语言有了更深刻的认知，学到了如何利用C语言提供的库函数对8086系统进行编程操控，如何输出数据到指定器件，如何选中目标器件的地址空间。本次实验课设收获颇丰。课设的各项安排都比较合理。至于课后建议，有一些实验仪器可能有问题，需要维修

**七、实验源代码**

#define u8 unsigned char

#define u16 unsigned int

#define IO8255\_PC 0x272 // 8255 C口

#define IO8255\_Con 0x273 // 8255 控制字

#define addr0809 0x250 // AD0809

#define IO273 0x230 // 74LS273

#define NO\_BUFFER 0x10 // 消隐

/\* 写I/O \*/

extern void outportb(unsigned int, char);

/\* 读IO \*/

extern char inportb(unsigned int);

// 键盘、数码管管理器初始化

extern void cInitKeyDisplay();

// pBuffer指向的8字节缓冲区内容显示于F5区数码管上

extern void cDisplay8(u8 \*pBuffer);

u8 buffer[8]; // 显示数码管缓冲区

u8 SpeedNo; // 速度

u8 StepDelay; // 转动一步后的延时常数

u16 StepCount; // 转动步数

u8 StepControl[8] = {0x33, 0x22, 0x66, 0x44, 0xcc, 0x88, 0x99, 0x11}; // 传给步进电机的值

u8 waterHeight = 0xff; // 水位 [0,255] AD转换8位精度

u8 waterHeight1 = 0xff; // waterHeight的副本

u16 adResult = 0; // 电位器读取的值

u16 lastResult = 0xff; // 电位器上次读取的值

u8 Steptemp = 0; // 步进电机的历史状态（上次旋转之后位于哪步）

u8 height = 0; // 水闸开启程度为height%

u8 acircle = 40; // 转1圈要步进电机需要走的总步数

u8 adtemp;

u8 count = 0;

u8 i = 0;

/\*刷新数码管\*/

void delayBuffer()

{

u16 count = 10; // TODO

while (count--)

{

cDisplay8(buffer);

}

}

/\*延时\*/

void delay(u16 ms)

{

u16 i;

while (ms--)

{

for (u16 i = 0; i < 100; i++)

;

}

}

/\*读AD0809数据\*/

u8 ad0809()

{

outportb(addr0809, 0x0);

u8 i = 100;

while (i--)

{

;

} // 延时,等待AD转换完成

return inportb(addr0809);

}

/\* 接收到电位器旋转变化，更新数码管 \*/

void displayData(u16 adResult)

{ // TODO

buffer[7] = adResult / 51; // 255/51 (16进制的1 = 1/51V) ，十位

adResult = (adResult % 51) \* 10;

buffer[6] = adResult / 51; // 个位

buffer[5] = NO\_BUFFER;

buffer[3] = NO\_BUFFER;

buffer[2] = NO\_BUFFER;

// 0 1 圈数

}

/\* 初始化8255 \*/

void init8255()

{

outportb(IO8255\_Con, 0x80); // 1 000 0 00 0

outportb(IO8255\_PC, 0xff); // 1 1111111

}

// 当水位为[10,50)时，绿灯亮；当水位为[0,10)时黄灯闪烁；当水位为50时，红灯闪烁

void testLED()

{

if (waterHeight1 >= 50)

{

outportw(IO273, 0xdf); // 红

delay(20);

outportw(IO273, 0xff); // 灭

}

else if (waterHeight1 >= 10)

{

outportw(IO273, 0x7f); // 绿

}

else

{

outportw(IO273, 0xbf); // 黄

delay(20);

outportw(IO273, 0xff);

}

}

/\* 实现步进电机旋转 circleN是转的圈数\*/

void stepCircle(u8 circleN)

{

u8 rate = 1; // 延迟 0是快速 1是中速

u8 htemp = 0;

if (circleN >= 8)

{

rate = 0; // 需要转10圈，即水位到了40以上，需要快速转动

}

if (circleN == 0)

{

rate = 0; // 需要转0圈，即水位到了10以下，需要快速转动

}

if (buffer[1] == 1 || circleN <= buffer[0]) // 逆时针

{

if (buffer[1] == 1)

{

circleN = 10 - circleN; // 从第10圈反转多少圈到达第circleN圈

}

else

{

circleN = buffer[0] - circleN;

}

while (circleN)

{

u8 control = 0;

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

{ // i表示1步

control = (Steptemp + i) % 8;

delay(rate \* 100); // 速度控制

buffer[4] = 2 - rate; // 显示的速度

outportb(IO8255\_PC, StepControl[control]); // 步机旋转

if (i % (acircle / 10) == 0)

{

// 转1圈水闸变化10%，一圈要旋转acircle次

height--;

buffer[1] = 0x00;

buffer[0] = height / 10;

if (buffer[0] == 0x0a)

{

buffer[1] = 0x01;

buffer[0] = 0x00;

}

delayBuffer(); // 刷新显示水闸开启程度

}

htemp = i;

testLED();

delay(40);

}

circleN--;

}

}

// 而circleN代表假设水闸开启程度为0时需要转的圈数,buffer[7]可以视作水闸已经转了多少圈,circleN减buffer[7]即代表还需要转多少圈

// else if (circleN <= buffer[0])

// { // 关闸，逆时针转

// circleN = buffer[0] - circleN;

// while (circleN)

// {

// u8 control = 0;

// for (i = acircle; i > 0; i--)

// {

// control = (Steptemp + i) % 8;

// delay(rate \* 100); // 速度控制

// buffer[4] = 2 - rate;

// outportb(IO8255\_PC, StepControl[control]); // 步机旋转

// if (i % (acircle / 10) == 0)

// {

// // 转1圈水闸变化10%，一圈要旋转acircle次

// height--;

// buffer[1] = 0x00;

// buffer[0] = height / 10;

// if (buffer[0] == 0x0a)

// {

// buffer[1] = 0x01;

// buffer[0] = 0x00;

// }

// delayBuffer(); // 刷新显示水闸开启程度

// }

// htemp = i;

// testLED();

// delay(40);

// }

// circleN--;

// }

// }

else

{ // 开闸，顺时针转；

circleN = circleN - buffer[0];

while (circleN)

{

u8 control = 0;

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

{

control = (Steptemp + i) % 8;

delay(rate \* 100);

buffer[4] = 2 - rate;

outportb(IO8255\_PC, StepControl[control]);

if (i % (acircle / 10) == 0)

{

height++;

buffer[1] = 0x00;

buffer[0] = height / 10;

if (buffer[0] == 0x0a)

{

buffer[1] = 0x01;

buffer[0] = 0x00;

}

delayBuffer();

}

htemp = i;

testLED();

delay(40);

}

circleN--;

}

}

Steptemp = htemp;

}

void main()

{

init8255();

cInitKeyDisplay(); // 键盘，数码管初始化

while (1)

{

count = 0;

adtemp = 0;

buffer[4] = 0; // 步进电机速度为0

// 采样时长为4s

while (1)

{

adResult = ad0809(); // 获取电压值

displayData(adResult); // 刷新数码管

if (adResult == adtemp)

{ // adResult不等于adtemp时，说明电压旋钮在旋转

count++;

}

else

{

adtemp = adResult;

}

if (count == 4)

{

break;

}

delayBuffer();

testLED();

delay(1000);

}

if (lastResult != adResult)

{ // 与历史版本比较，只有电压变化时才会重新设置水位

lastResult = adResult; // 记录新电压值的副本作为历史版本

waterHeight = buffer[7] \* 10 + buffer[6]; // 更新水位

if (waterHeight1 == waterHeight)

{

waterHeight = 0xff;

}

else

{

waterHeight1 = waterHeight; // 保存水位的副本作为历史版本

}

}

delayBuffer();

testLED();

if (waterHeight > 100)

{

waterHeight = 0xff;

}

else if (waterHeight >= 50)

{

stepCircle(10);

waterHeight = 0xff;

}

else if (waterHeight >= 40)

{

stepCircle(8);

waterHeight = 0xff;

}

else if (waterHeight >= 30)

{

stepCircle(6);

waterHeight = 0xff;

}

else if (waterHeight >= 20)

{

stepCircle(4);

waterHeight = 0xff;

}

else if (waterHeight >= 10)

{

stepCircle(2);

waterHeight = 0xff;

}

else

{

stepCircle(0);

waterHeight = 0xff;

}

}

}

附件：实验连线图

