

以步进电机为例探讨《单片机应用技术》理实一体课教学过程

李春满¹ 巴图²

1. 内蒙古机电职业技术学院 内蒙古 呼和浩特 010070

2. 内蒙古航天红岗机械有限公司 内蒙古 呼和浩特 010072

摘要: 随着科技进步,嵌入式技术得到广泛的发展,单片机在日常生活及生产中,得到了大规模的应用,在一些简单的控制系统中,是主要的控制元件。单片机相关的课程成为了高职院校电气自动化等专业必备的课程设置。随着高职教育的发展,对单片机相关课程的教学提出了较高的要求,必须要有一定课时和高质量的实操课程来配合理论课的教学,才能使学生更好的掌握单片机的开发,提高动手实践的能力,培养创新精神。单片机的学习重点是熟悉硬件,灵活编程,这些知识都比较抽象,学生在学习过程中会有畏难情绪,没有动力。结合多年的教学实践,以单片机教学中比较难理解的步进电机项目教学为例,讲解单片机的应用及编程方法,探讨《单片机应用技术》理实一体课的教学过程。

关键词: 单片机; 步进电机; 理实一体

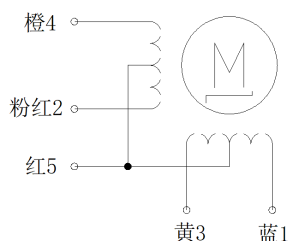
1 任务分析

步进电动机因控制灵活、性能好、运行可靠、误差不会长期累积、适于数字控制等多种优点而广泛应用于各行业的数控加工设备、自动生产线、自动控制仪表、计算机及办公室自动化设备甚至家用电器中^[1]。通过单片机控制步进电机的正反转、加减速,加深对步进电机原理的了解。熟练掌握单片机控制步进电机的基本方法。为后续毕业综合实训做好铺垫。

2 步进电机工作原理

本课程中用到的步进电机产品型号为: 28BYJ-48。接线示意图如下:

接线指示图



由接线图可以判定此步进电机为五线四相步进电机,其旋转一周的工作状态采用单双八拍模式,具体通电

课题编号: NJDRW2210

课题名称: 提质培优背景下,“四维度”创新创业课程体系的探索与实践—以电力系统自动化技术专业群为例

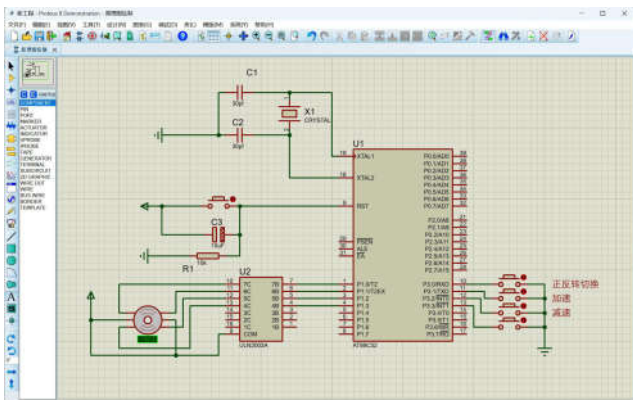
顺序如表:

导线颜色	第1步	第2步	第3步	第4步	第5步	第6步	第7步	第8步
红5	5V	5V	5V	5V	5V	5V	5V	5V
橙4	GND	GND						GND
黄3		GND	GND	GND				
粉红2				GND	GND	GND		
蓝1						GND	GND	GND

红线为公共端,电压大小为+5V,当步进电机其余四根引出线(四相绕组)按照一定的时序与公共端接通时,使所产生的脉冲序列按照一定的时序送到绕组线圈上,这样步进电机就可以逐步旋转了^[2]。单双八拍工作方式的通电顺序依次为:橙→橙黄→黄→黄粉→粉→粉蓝→蓝→蓝橙→橙,如果要让步进电机反转,只需按相反的顺序通电即可(即通电顺序为:橙←橙黄←黄←黄粉←粉←粉蓝←蓝←蓝橙←橙)。

3 单片机控制步进电机电路设计

本课程主要学习单片机的程序编写及步进电机的基本知识,因此本项目的电路设计采用仿真软件进行。51单片机仿真软件主要推荐Proteus。本实验项目通过ULN2003驱动模块控制28BYJ-48步进电机^[3],实现步进电机正反转的控制、速度的调节。通过按键实现信号输入,当按下KEY1键可调节电机旋转方向,当按下KEY2键,电机加速,当按下KEY3键,电机加速,电机减速。通过仿真软件Proteus,搭建的电路图如下:



4 单片机控制步进电机程序设计 [4][5][6]

基于keilC51的C语言步进电机控制程序设计如下：

```
#include "reg52.h"
typedef unsigned int u16; //对系统默认数据类型进行重定义
typedef unsigned char u8;
//定义ULN2003控制步进电机管脚
sbit IN1_A=P1^0;
sbit IN2_B=P1^1;
sbit IN3_C=P1^2;
sbit IN4_D=P1^3;
//定义独立按键控制脚
sbit KEY1=P3^1;
sbit KEY2=P3^0;
sbit KEY3=P3^2;
sbit KEY4=P3^3;
//使用宏定义独立按键按下的键值
#define KEY1_PRESS 1
#define KEY2_PRESS 2
#define KEY3_PRESS 3
#define KEY4_PRESS 4
#define KEY_UNPRESS 0
//定义步进电机速度，值越小，速度越快
//最小不能小于1
#define STEPMOTOR_MAXSPEED 1
#define STEPMOTOR_MINSPEED 5

void delay_10us(u16 ten_us)
{
    while(ten_us--);
}

/*****
```

```
*****/
* 函数名: delay_ms
* 函数功能: ms延时函数, ms = 1时, 大约延时1ms
* 输入: ten_us
* 输出: 无
*****/
void delay_ms(u16 ms)
{
    u16 i,j;
    for(i=ms;i>0;i--)
        for(j=110;j>0;j--);
}

/*****
*****/
* 函数名: step_motor_28BYJ48_send_pulse
* 函数功能: 输出一个数据给ULN2003从而实现向步进电机发送一个脉冲
* 输入: step: 指定步进序号, 可选值0~7
        dir: 方向选择,1: 顺时针,0: 逆时针
* 输出: 无
*****/
void step_motor_28BYJ48_send_pulse(u8 step,u8 dir)
{
    u8 temp=step;

    if(dir==0) //如果为逆时针旋转
        temp=7-step;//调换节拍信号
    switch(temp)//8个节拍控制: A->AB->B->BC->C->CD->D->DA
    {
        case 0: IN1_A=1;IN2_B=0;IN3_C=0;IN4_D=0;break;
        case 1: IN1_A=1;IN2_B=1;IN3_C=0;IN4_D=0;break;
        case 2: IN1_A=0;IN2_B=1;IN3_C=0;IN4_D=0;break;
        case 3: IN1_A=0;IN2_B=1;IN3_C=1;IN4_D=0;break;
        case 4: IN1_A=0;IN2_B=0;IN3_C=1;IN4_D=0;break;
        case 5: IN1_A=1;IN2_B=0;IN3_C=1;IN4_D=0;break;
        case 6: IN1_A=1;IN2_B=0;IN3_C=0;IN4_D=1;break;
        case 7: IN1_A=0;IN2_B=0;IN3_C=0;IN4_D=1;break;
    }
}

/*****
```


```

        case 5: IN1_A=0;IN2_B=0;IN3_C=1;IN4_ 速度旋转
D=1;break;                                u8 step=0;
        case 6: IN1_A=0;IN2_B=0;IN3_C=0;IN4_
D=1;break;                                while(1)
        case 7: IN1_A=1;IN2_B=0;IN3_C=0;IN4_
D=1;break;                                {
        default: IN1_A=0;IN2_B=0;IN3_
C=0;IN4_D=0;break;//停止相序                key=key_scan(0);
    }                                        if(key==KEY1_PRESS)//换向
}                                           {
u8 key_scan(u8 mode)                        dir=!dir;
{                                           }
    static u8 key=1;                        else if(key==KEY2_PRESS)//加速
                                           {
        if(mode)key=1;//连续扫描按键        if(speed>STEPMOTOR_
        if(key==1&&(KEY1==0||KEY2==0||KEY3==0||K  MAXSPEED)
        EY4==0))//任意按键按下                speed-=1;
        {                                     }
            delay_10us(1000);//消抖        else if(key==KEY3_PRESS)//减速
            key=0;                            {
            if(KEY1==0)                        if(speed<STEPMOTOR_
                return KEY1_PRESS;                MINSPEED)
            else if(KEY2==0)                    speed+=1;
                return KEY2_PRESS;                }
            else if(KEY3==0)                    step_motor_28BYJ48_send_
                return KEY3_PRESS;                pulse(step++,dir);
            else if(KEY4==0)                    if(step==8)step=0;
                return KEY4_PRESS;                delay_ms(speed);
        }                                       }
    }                                           }
    else if(KEY1==1&&KEY2==1&&KEY3==1&&K  }
    EY4==1) //无按键按下
    {
        key=1;
    }
    return KEY_UNPRESS;
}

void main()
{
    u8 key=0;
    u8 dir=0;//默认逆时针方向
    u8 speed=STEPMOTOR_MAXSPEED;//默认最大

```

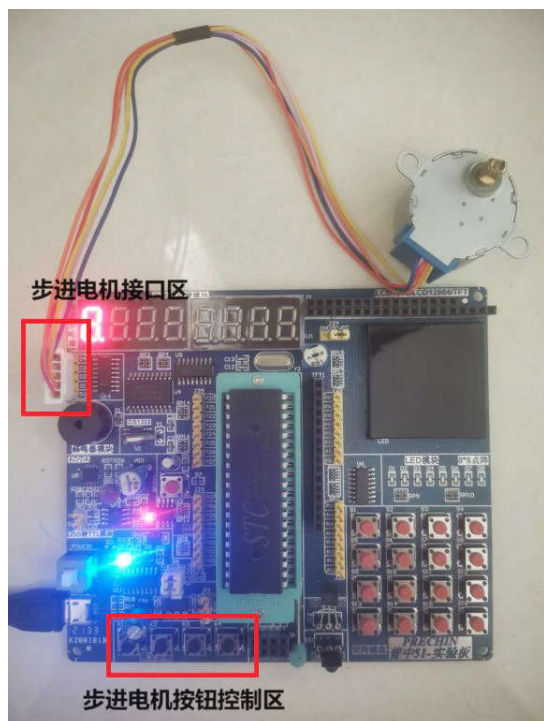
5 仿真运行结果

以proteus软件仿真为例：点击电路中的单片机最小应用系统，按鼠标右键，找到Edit Properties添加程序（必须是编译程序时生成的hex文件格式），点出右侧文件夹，如图 ，找到生成的hex文件并打开，即完成添加程序^[3]。然后在proteus界面左下角点击“运行”按钮，再点击K1、K2、K3按键，步进电机按照设定程序运行。

6 步进电机的控制运行

通过电路的搭建及程序的编写，仿真运行、利用单片机控制步进电机的关键步骤已经学习完成，仿真运行可以检查程序是否编写正确，如果仿真运行没有错误，就可

将编写好的程序通过软件写入实训课程中使用的实验板中，将步进电机接入开发板，如图所示，通过开发板上的按键来控制电机的运行，验证所学内容是否正确。



结论

通过采用keilC51对单片机控制步进电机系统进行硬件电路设计、软件编程、仿真实验、实物调试，系统学习单片机最小系统、单片机按钮应用、c语言编程思路、步进电机运行原理等。学习过程中，学生通过理论到实践的逐步过渡，将抽象知识凝练到具体的实物当中，通过按键让单片机按照控制要求动作，直观生动的学习理论知识。达到单片机实训课的教学效果。

参考文献

- [1]袁炜,张宝,吴饶,等.基于单片机的步进电机控制系统设计[J].电脑知识与技术,2020,16(18):214-216
- [2]李娟,马利祥.步进电机控制系统设计[J].科学技术创新,2021(8):191-192.
- [3]孙巧智.基于单片机的步进电机控制系统研究[J].电子世界,2020(11):86-87.
- [4]郭天祥.新概念51单片机C语言教程.北京:电子工业出版社,2009.324~331
- [5]张天凡.51单片机C语言开发详解.北京:电子工业出版社,2008.152~164
- [6]侯殿有.单片机C语言程序设计.北京:人民邮电出版社,2010.28~47