

学生学号	0121618380615	实验课成绩	
------	---------------	-------	--

武汉理工大学 学 生 实 验 报 告 书

实验课程名称	单片机原理及接口技术
开 课 学 院	物流工程学院
指导教师姓名	袁 兵
学 生 姓 名	付清晨
学生专业班级	机设 1606

2018 -- 2019 学年 第 1 学期

实验课程名称：单片机原理及接口技术

实验项目名称	步进电机控制实验			实验成绩	
实 验 者	付清晨	专业班级	机设 1606	组 别	
同 组 者	无			实验日期	2018 年 12 月 7 日

1. 实验要求

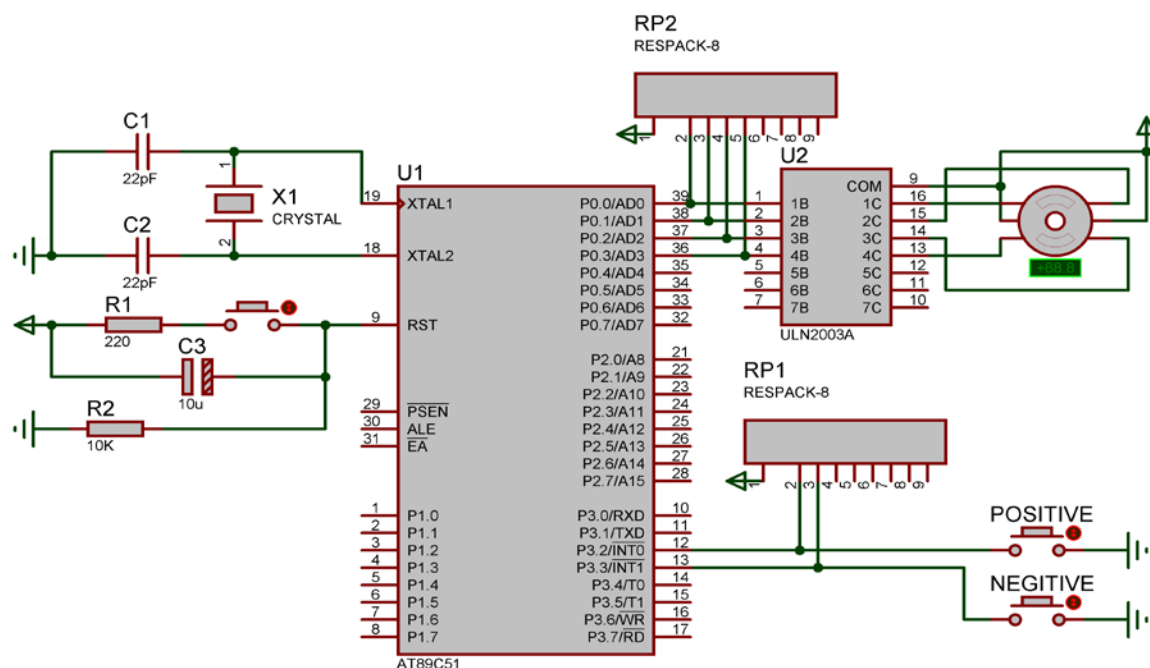
利用单片机实现对步进电机的控制，编写程序，用 4 路 I/O 口实现环形脉冲的分配，控制步进电机按固定方向连续转动。同时，要求按下“Positive（正转）”按键时，控制步进电机正转；按下“Negative（反转）”按键时，控制步进电机反转；松开按键时，电机停止转动。通过“正转”和“反转”两个按键，来控制电机的正转与反转。

2. 实验原理

对步进电机的驱动，是通过对步进电机每组线圈中的电流的顺序切换来使电机作步进式旋转。切换是通过单片机输出脉冲信号来实现的。所以调节脉冲信号的频率就可以改变步进电机的转速，改变各相脉冲的先后顺序，就可改变电机的转向。步进电机的转速应由慢到快。

3. 系统电路的设计

3.1 电路原理图



3.2 电子元器件型号和数量

名称	数量	位号
9C04021A2200FLH	1	R1
9C08052A1002JLH	1	R2
02013A220JAT2A	2	C1,C2
AT89C51	1	U1
CRYSTAL	1	X1
B45196H4106K309	1	C3
BUTTON	3	-
MOTOR-STEPPER	1	-
ULN2003A	1	U2
RESPACK-8	2	RP1,RP2

3.3 电路工作原理的重点说明

3.3.1 时钟电路与复位电路

电容 C1,C2 与晶振片 X1 并联接在 XTAL1, XTAL2 构成时钟电路

电容 C3 与按钮及电阻 R1 并联接在 RST 上, 构成复位电路

3.3.2 按钮控制模块

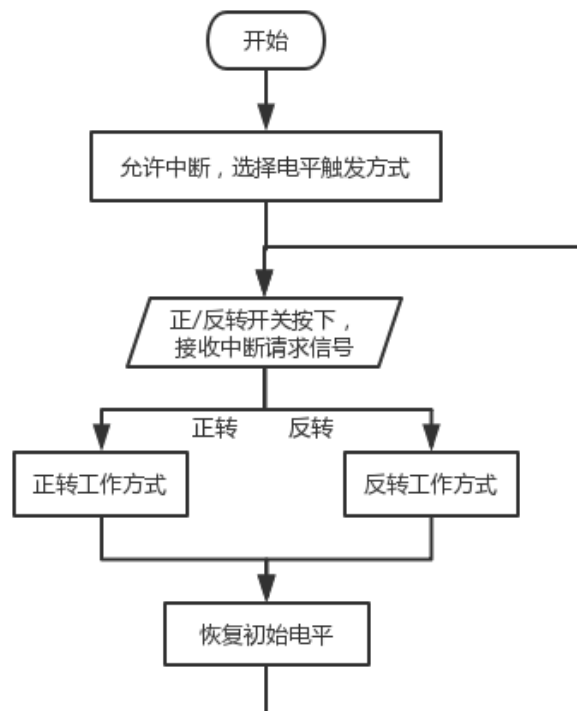
按钮 POSITIVE,NEGITIVE 分别接在 INT0,INT1 上, 通过 P3.3,P3.4 读取中断请求信号

3.3.3 步进电机控制模块

步进电机的 4 个接口分别通过驱动设备 ULN2003A 接在 P0 口, 通过 P0 控制步进电机的前进后退

4. 系统控制程序的设计

4.1 控制程序的工作流程图



4.2 控制程序的源代码

```

/*****
* @Author: fuqingchen@whut.edu.cn
* @Description: 步进电机控制实验
*****/

#include <reg51.h>
unsigned char turn[] = { 0x01,0x03,0x02,0x06,0x04,0x0C,0x08,0x09 };
unsigned int i = 0;

/*****
* @Description: 延时函数
* @Input: t:延长的时间
*****/

void delay(unsigned int t) {
    unsigned char i;
    while (t--) {
        for (i = 0; i<120; i++);
    }
}

```

```

/*****
* @Description: 使步进电机正转
*****/

void positive() interrupt 0 {
    //TODO:使步进电机正转
    i++;
    if (i>=8)
    {
        i = 0;
    }
    P0 = turn[i];
    delay(100);
    P0 = 0xff;
}

/*****
* @Description: 使步进电机反转
*****/

void negative() interrupt 2 {
    //TODO:使步进电机反转
    i--;
    if (i <= 0)
    {
        i = 7;
    }
    P0 = turn[i];
    delay(100);
    P0 = 0xff;
}

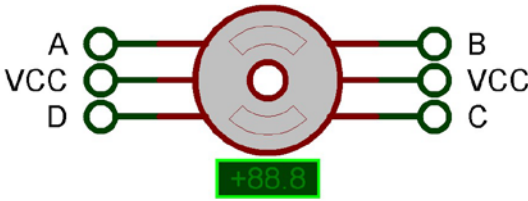
void main() {
    EA = 1; //允许总中断
    EX0 = 1; EX1 = 1; //允许外部中断0,1
    IT0 = 0; IT1 = 0; //选择电平触发方式
    IP = 0; //低优先级
    while (1) {
        delay(100);
    }
}

```

4.3 控制程序的重点说明

4.3.1 步进电机的控制

查询相关资料可得，Proteus 中六线制步进电机接线如图所示：



为了使步进电机运行平稳，此处采用单、双八拍方式即正转：

A→AB→B→BC→C→CD→D→DA

反转：

AD→D→DC→C→CB→B→BA→A

通过上述原理可以得到步进电机控制代码

通电	1B	2B	3B	4B	代码
A	1	0	0	0	0x01
AB	1	1	0	0	0x03
B	0	1	0	0	0x02
BC	0	1	1	0	0x06
C	0	0	1	0	0x04
CD	0	0	1	1	0x0C
D	0	0	0	1	0x08
DA	1	0	0	1	0x09

表中 1B，2B，3B，4B 对应电路图中 U2 的 1B，2B，3B，4B

在程序中通过这个数组实现步进电机的运动。

4.3.2 中断的配置/初始化

```
EA = 1;    // 允许总中断
EX0 = 1; EX1 = 1; // 允许外部中断源 0,1
IT0 = 0; IT1 = 0; // 选择电平触发方式
IP = 0;    // 低优先级，这是书上写出的，我也跟着写了这句
```

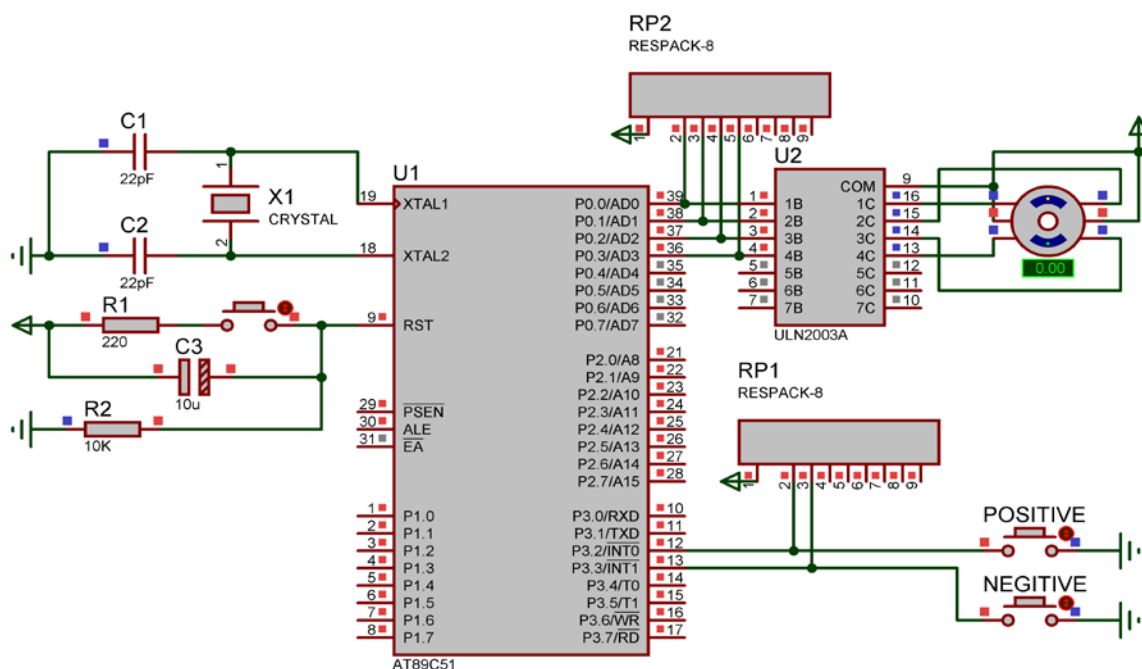
5. 系统的调试和结果

5.1 系统调试的方法

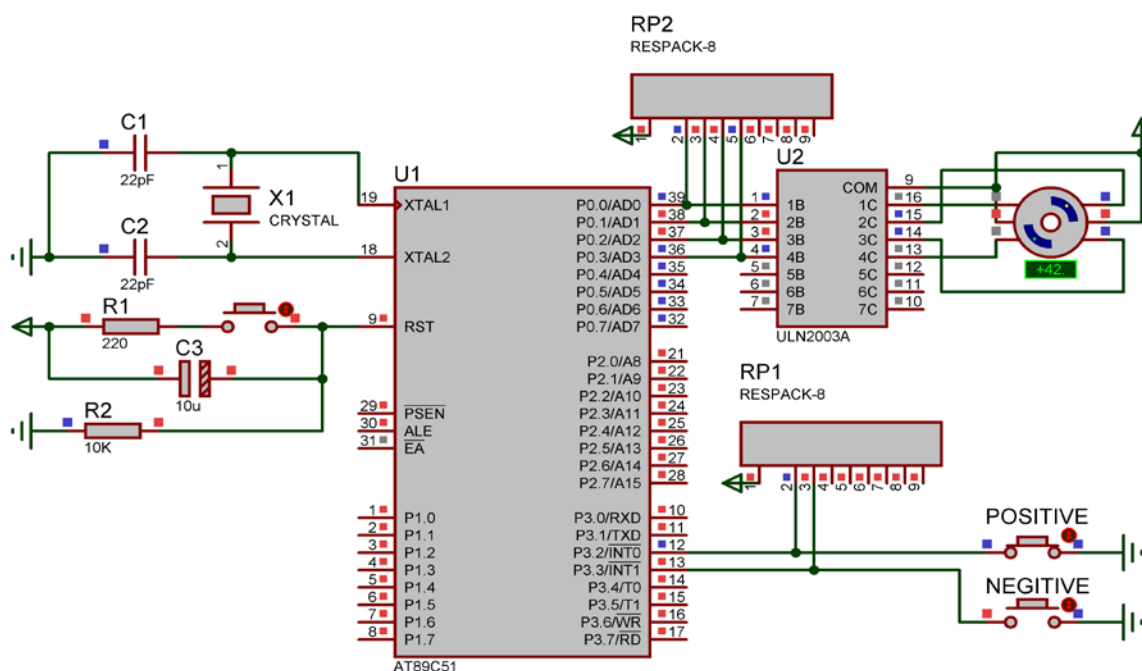
1. 在 Keil uVision 中将写好的控制程序编译生成 hex 文件
2. 在 Proteus 画好电路图后，将软件中的模拟单片机与生成的 hex 文件链接
3. 在 Proteus 中进行仿真，观察现象。
4. 若不满足要求，重新修改控制程序并生成 hex 文件

5.2 系统正确运行的图片

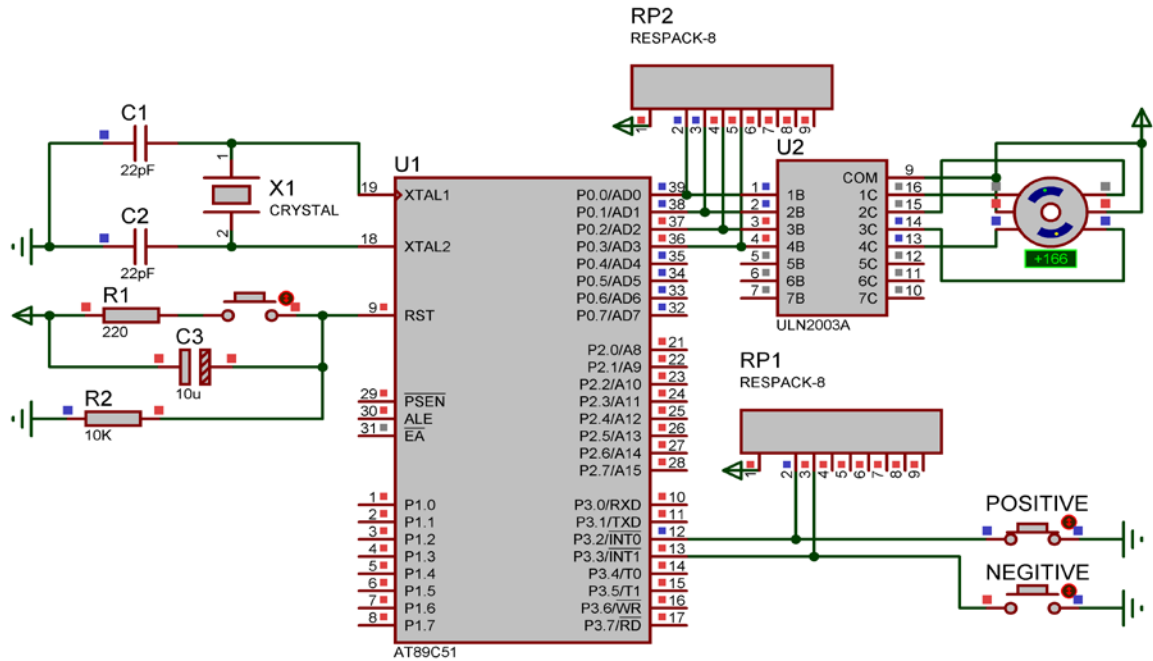
5.2.1 初始状态：



5.2.2 按下 POSITIVE 后：



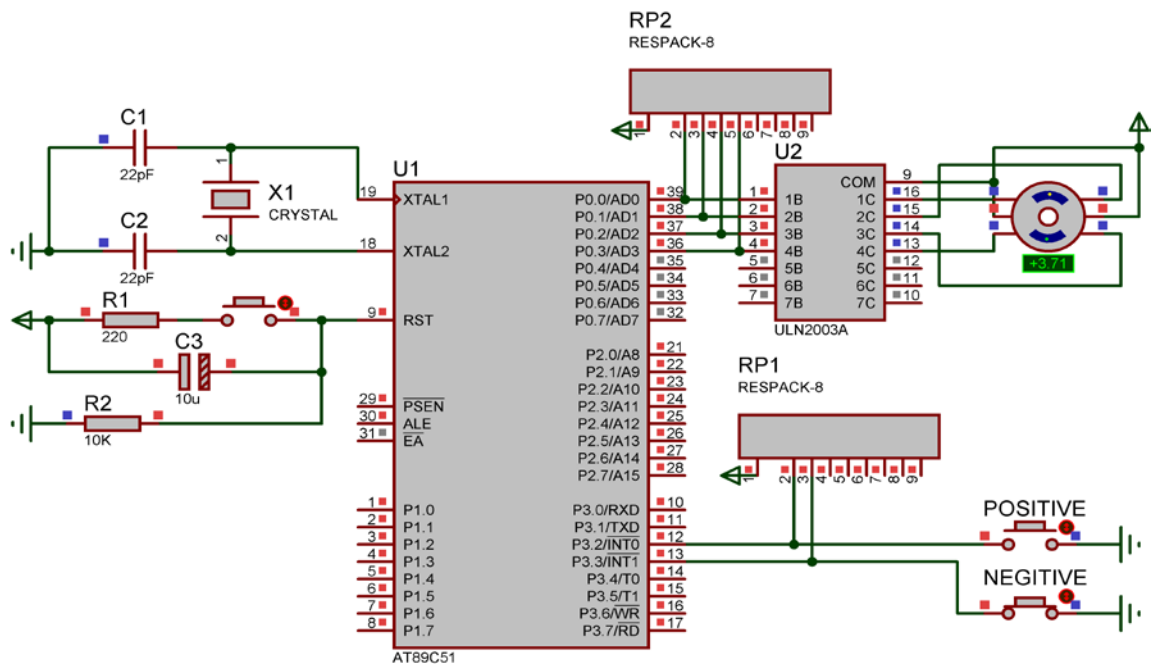
一小段时间内，步进电机顺时针旋转了 42° ($0-42^{\circ}$)



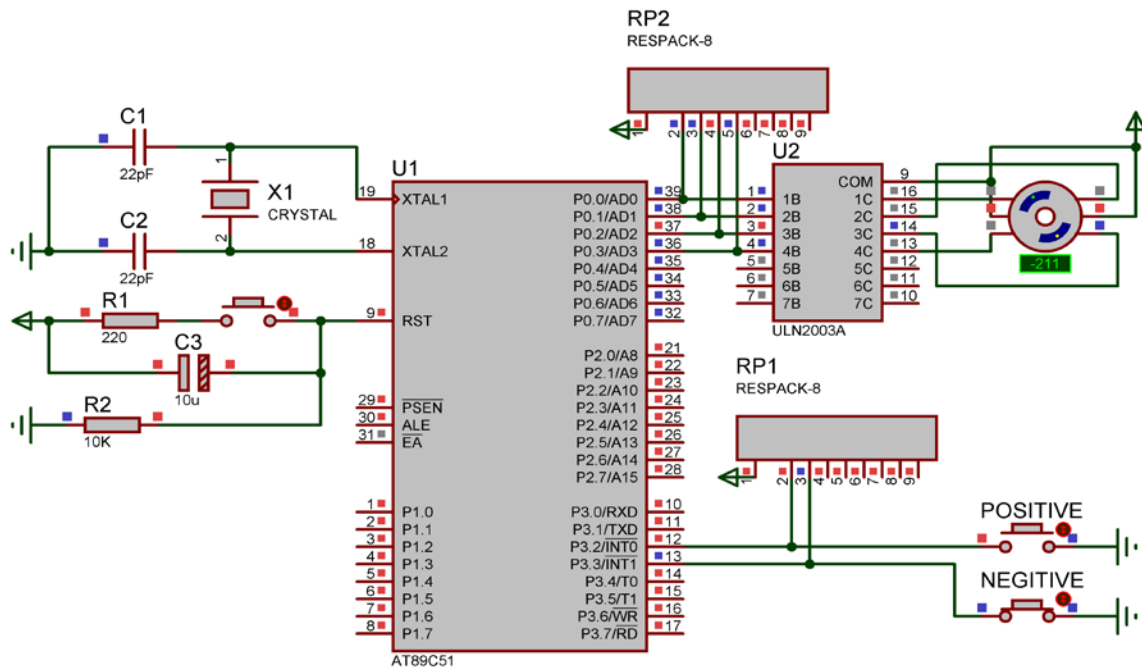
一小段时间内，步进电机顺时针旋转了 124° ($42^{\circ}-166^{\circ}$)

5.2.3 按下 NEGATIVE 后：

5.2.4



一小段时间内，步进电机逆时针旋转了 162° ($166^{\circ}-4^{\circ}$)



一小段时间内，步进电机逆时针旋转了 215° (-4° - 211°)

6. 实验过程中所遇到的问题，解决方法和建议

6.1 【程序】步进电机的控制

6.1.1 问题的描述

一开始每按一次开关，步进电机都会跳到初始位置，从初始位置开始旋转。

6.1.2 解决方法

将中断函数中控制相位的变量设置为全局变量，两个中断函数公用同一个变量

```
unsigned int i = 0;
void positive() interrupt 2 {
    .....
    i++;
    .....
    P0 = turn[i];
    .....
}
void negative() interrupt 2 {
    .....
    i--;
    .....
    P0 = turn[i];
    .....
}
```

