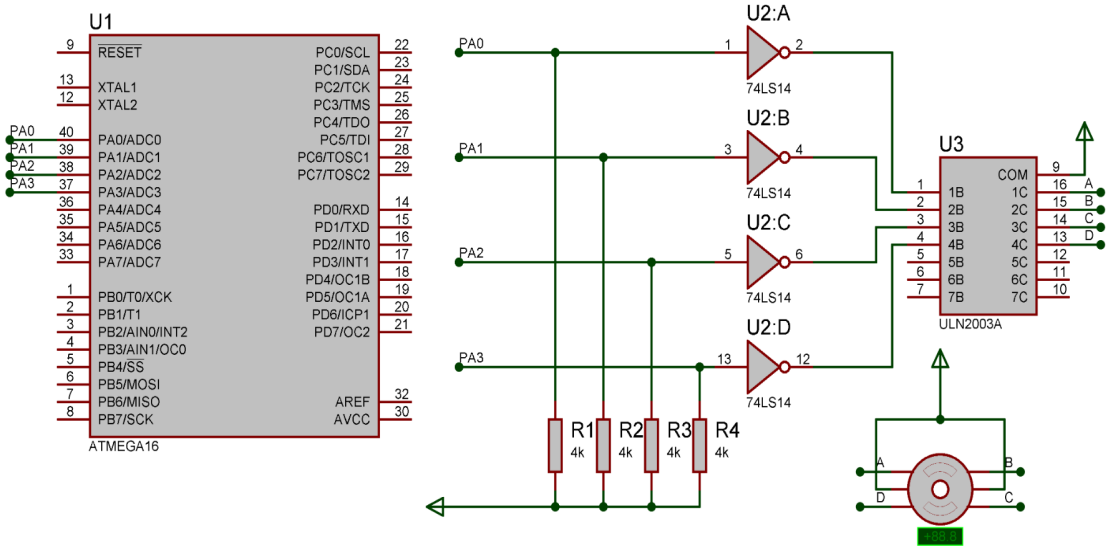
实验 9: 步进电机试验

1. 试验描述

本实验的目的是掌握步进机的控制操作，ATmega16 芯片 PA 口的低四位接步进电机接口，按相应的节拍赋予 PA 口的低四位相应的电平，实现步进转动，并通过延时（即通断频率）控制步进机的转速。

2. 系统框图

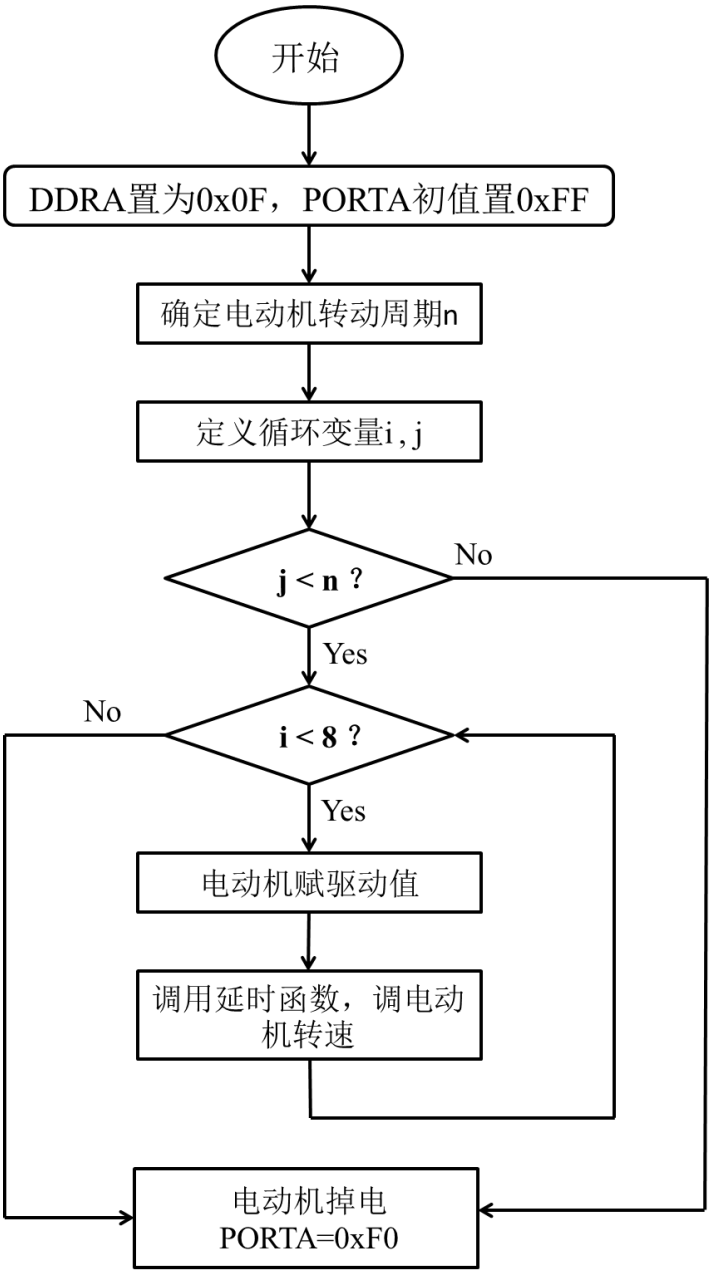
➢ 硬件电路



➢ 元件清单

单片机 ATmega16	电阻 RES
步进电机 MOTOR-STEPPER	反相器 74LS14
步进电机驱动芯片 ULN2003A	

➢ 软件流程



➢ 步进电机原理

步进电机是将电脉冲信号转变为角位移或线位移的控制元件。在非超载的情况下，电动机的转速、停止的位置只取决于脉冲信号的频率和脉冲数,而不受负载变化的影响。步进电机必须由双环形脉冲信号、 功率驱动电路等组成控制系统方可使用。

步进电机的主要特性：

1）步进电机必须加驱动才可以运转，驱动型号必须为脉冲信号,没有脉冲的

时候，步进电机静止，如果加入适当的脉冲信号，就会以一定的角度(称为步距角)转动。转动的速度和脉冲的频率成正比。

2）本试验步进电机的步距角为 7.5 度,一圈 360 度，需要 48 个脉冲完成。

3）步进电机有瞬间启动和急速停止的优越特性。

4）改变脉冲的顺序，可以改变转动的方向。

步进电机应用驱动电路：

小型步进电机对电压和电流的要求不是很高，可采用简单的驱动电路，如图 9-1 所示。在实际应用中驱动路数般不止一路。由于分立电路体积大，所以很多场合采用现成的集成电路作为多路驱动。常用的小型步进电机驱动电路可以使用 ULN2003A 或 ULN2803。

ULN2003A 是高电压、大电流达林顿晶体管阵列系列产品，具有电流增益高(灌电流可达 500mA)、工作电压高(可承受 50V 的电压)、温度范围宽、带负载能力强等特点，适用于各类要求是高速、大功率驱动的系统。ULN2003A 的输出端将会允许通过 200mA 的 IC 电流，饱和压降 V_{CE} 约为 1V 左右，耐压 V_{CEO} 约为 36V。由于其输出电流大，故可以直接驱动继电器或固体继电器(SSR)等外接控制器 P 件，也可直接驱动低压灯泡。ULN2003A 由 7 组达林顿体管阵列、 相应的电阻网络和钳位二极管网络构成，具有同时驱动 7 组负载的能力，为单片双极型大功率高速集成电路，其内部结构如图 9-2 所示。

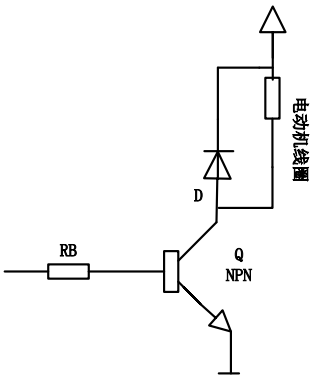


图 9-1　简单驱动电路

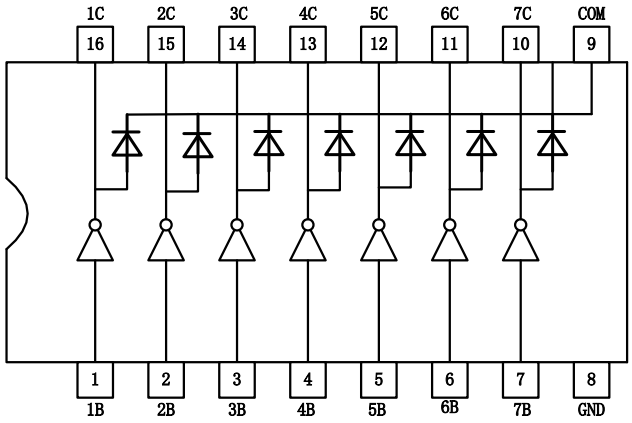


图 9-2　ULN2003A 的内部结构图

ULN2003A 的每一对达林顿都串联一个 2.7k 的基极电阻，在 5V 的工作电压下它能与 TTL 和 CMOS 电路直接相连，可以直接处理原先需要标准逻辑缓冲器来处理的数据。

ULN2003A 可以并联使用，在相应的 OC 输出引脚上串联几个欧姆均流的电阻后再并联使用，可以防止阵列电流不平衡。由于 ULN2003A 的输出结构是集电极开路的，所以要在输出端接一个上拉电阻，在输入为低电平时输出才是高电平。在用它驱动负载时，电流是由电源通过负载灌入 ULN2003A 的。

3. 程序代码

- ICCAVR 程序

```
#include <iom16v.h>
#define uchar unsigned char
#define uint unsigned int
//头文件
//数据类型说明 //数据类型说明
const uchar FFW[8]={0xf9,0xf1,0xf3,0xf2,0xf6,0xf4,0xfc,0xf8};
const uchar REV[8]={0xf6,0xf2,0xf3,0xf1,0xf9,0xf8,0xfc,0xf4};

void delayms(uint n)    //定义延时子函数
{
     uint i,K=0 ;
     for (K=0 ;K<n ;K++)
     for (i=0 ;i<250 ;i++) ;
}

void motor_ffw(uint n)  //步进电机正转驱动子函数
{
     uchar i;        //循环变量 i,j
     uint j;
     for ( j=0; j<n; j++)  //根据 n 值进行 n 个周期循环
     {
         for (i=0; i<8; i++)  // 输出 8 次脉冲信号
         {
             PORTA = FFW[i]; //取数据
             delayms(25); //调转速
         }
    }
    PORTA=0xf0; //循环完成后使电机掉电
}
void motor_rev(uint n)  //步进电机反转驱动子函数
{
     uchar i;        //循环变量 i,j
     uint j;
```

```
     for ( j=0; j<n; j++)  //根据 n 值进行 n 个周期循环
     {
          for (i=0; i<8; i++) //输出 8 次脉冲信号
          {
               PORTA = REV[i]; //取数据
               delayms(25);//调转速
          }
     }
     PORTA=0xf0; //循环完成后使电机掉电
     }

//该步进电机一个脉冲周期转 30°
int main(void)
{
    DDRA=0x0F;    //置 PA 口为输出
    PORTA=0xFF; //置 PA 口初值
    while(1)
    {
        motor_ffw(12); //调用电机正传函数 进行 12 个周期的正转
        delayms(1000); //延时
        motor_rev(6);  //调用电机反传函数 进行 6 个周期的反转
        delayms(1000); //延时
    }
}
```

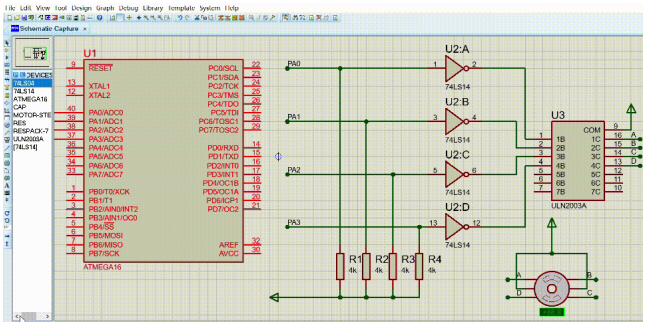
- CVAVR 程序

```
#include <mega16.h>
#define uchar unsigned char
#define uint unsigned int
//头文件
//数据类型说明 //数据类型说明
const uchar FFW[8]={0xf9,0xf1,0xf3,0xf2,0xf6,0xf4,0xfc,0xf8};
const uchar REV[8]={0xf6,0xf2,0xf3,0xf1,0xf9,0xf8,0xfc,0xf4};

void delayms(uint n)   //定义延时子函数
{
    uint i,K=0 ;
    for (K=0 ;K<n ;K++)
    for (i=0 ;i<250 ;i++) ;
}

void motor_ffw(uint n)     //步进电机正转驱动子函数
{
     uchar i;         //循环变量 i,j
```

```
     uint j;
     for ( j=0; j<n; j++)  //根据 n 值进行 n 个周期循环
     {
           for (i=0; i<8; i++)  //输出 8 次脉冲信号
         {
               PORTA = FFW[i]; //取数据
               delayms(25); //调转速
         }
    }
    PORTA=0xf0; //循环完成后使电机掉电
}
void motor_rev(uint n)  //步进电机反转驱动子函数
{
     uchar i;          //循环变量 i,j
     uint j;
     for ( j=0; j<n; j++)  //根据 n 值进行 n 个周期循环
     {
           for (i=0; i<8; i++) //输出 8 次脉冲信号
         {
             PORTA = REV[i]; //取数据
             delayms(25);//调转速
         }
     }
     PORTA=0xf0; //循环完成后使电机掉电
     }

//该步进电机一个脉冲周期转 30°
void main(void)
{
    DDRA=0x0F;       //置 PA 口为输出
    PORTA=0xFF; //置 PA 口初值
    while(1)
    {
        motor_ffw(12); //调用电机正传函数 进行 12 个周期的正转
        delayms(1000); //延时
        motor_rev(6);  //调用电机反传函数 进行 6 个周期的反转
        delayms(1000); //延时
    }
}
```

4. 仿真结果

步进电机，先正转 360 度，再反转 180 度，再正转 360 度，再反转 180 度•••。



（双击图片，演示仿真结果）