

# 霍尔测速

测速是工农业生产中经常遇到的问题，学会使用单片机技术设计测速仪表具有很重要的意义。要测速，首先要解决是采样的问题。在使用模拟技术制作测速表时，常用测速发电机的方法，即将测速发电机的转轴与待测轴相连，测速发电机的电压高低反映了转速的高低。使用单片机进行测速，可以使用简单的脉冲计数法。只要转轴每旋转一周，产生一个或固定的多个脉冲，并将脉冲送入单片机中进行计数，即可获得转速的信息。

下面以常见的玩具电机作为测速对象，用 CS3020 设计信号获取电路，通过电压比较器实现计数脉冲的输出，既可在单片机实验箱进行转速测量，也可直接将输出接到频率计或脉冲计数器，得到单位时间内的脉冲数，进行换算即可得电机转速。这样可少用硬件，不需编程，但仅是对霍尔传感器测速应用的验证。

## 1 脉冲信号的获得

霍尔传感器是对磁敏感的传感元件，常用于开关信号采集的有 CS3020、CS3040 等，这种传感器是一个 3 端器件，外形与三极管相似，只要接上电源、地，即可工作，输出通常是集电极开路（OC）门输出，工作电压范围宽，使用非常方便。如图 1 所示是 CS3020 的外形图，将有字面对准自己，三根引脚从左向右分别是 Vcc，地，输出。



图 1 CS3020 外形图

使用霍尔传感器获得脉冲信号，其机械结构也可以做得较为简单，只要在转轴的圆周上粘上一粒磁钢，让霍尔开关靠近磁钢，就有信号输出，转轴旋转时，就会不断地产生脉冲信号输出。如果在圆周上粘上多粒磁钢，可以实现旋转一周，获得多个脉冲输出。在粘磁钢时要注意，霍尔传感器对磁场方向敏感，粘之前可以先手动接近一下传感器，如果没有信号输出，可以换一个方向再试。这种传感器不怕灰尘、油污，在工业现场应用广泛。

## 2 硬件电路设计

测速的方法决定了测速信号的硬件连接，测速实际上就是测频，因此，频率测量的一些原则同样适用于测速。

通常可以用计数法、测脉宽法和等精度法来进行测试。所谓计数法，就是给定一个闸门时间，在闸门时间内计数输入的脉冲个数；测脉宽法是利用待测信号的脉宽来控制计数门，对一个高精度的高频计数信号进行计数。由于闸门与被测信号不能同步，因此，这两种方法都存在 $\pm 1$ 误差的问题，第一种方法适用于信号频率高时使用，第二种方法则在信号频率低时使用。等精度法则对高、低频信号都有很好的适应性。

图2是测速电路的信号获取部分，在电源输入端并联电容 $C_2$ 用来滤去电源尖啸，使霍尔元件稳定工作。HG表示霍尔元件，采用CS3020，在霍尔元件输出端（引脚3）与地并联电容 $C_3$ 滤去波形尖峰，再接一个上拉电阻 $R_1$ ，然后将其接入LM324的引脚3。用LM324构成一个电压比较器，将霍尔元件输出电压与电位器 $R_{P1}$ 比较得出高低电平信号给单片机读取。 $C_4$ 用于波形整形，以保证获得良好数字信号。LED便于观察，当比较器输出高电平时不亮，低电平时亮。微型电机M可采用型，通过电位器 $R_{P1}$ 分压，实现提高或降低电机转速的目的。 $C_1$ 电容使电机的速度不会产生突变，因为电容能存储电荷。

电压比较器的功能：比较两个电压的大小(用输出电压的高或低电平，表示两个输入电压的大小关系)：

当“+”输入端电压高于“-”输入端时，电压比较器输出为高电平；

当“+”输入端电压低于“-”输入端时，电压比较器输出为低电平；

比较器还有整形的作用，利用这一特点可使单片机获得良好稳定的输出信号，不至于丢失信号，能提高测速的精确性和稳定性。

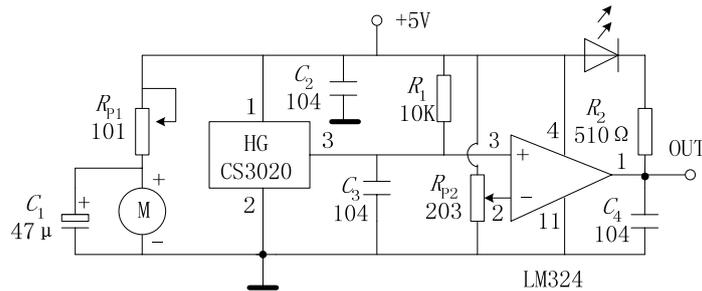


图.2 测速电路原理图

### 3 测速程序

测量转速，使用霍尔传感器，被测轴安装有1只磁钢，即转轴每转一周，产生1个脉冲，要求将转速值（转/分）显示在数码管上。

用C语言编制的程序如下：

//硬件：老版STC实验版

//P3-5口接转速脉冲

```
#include <STC12C5410AD.H> // 单片机内部专用寄存器定义
#define uchar unsigned char
#define uint unsigned int //数据类型的宏定义
uchar code LK[10]={0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xF8, 0x80, 0x90, } ;//数码管
0~9的字型码
```

```

uchar LK1[4]={0xfe, 0xfd, 0xfb, 0xf7}; //位选码
uint data z, counter;
//定义无符号整型全局变量 lk
//=====
void init(void) //定义名为 init 的初始化子函数
{ //init 子函数开始, 分别赋值
  TMOD=0X51; //GATE C/T M1 M0 GATE C/T M1 M0 计数器 T1 定时器 T0
            // 0 1 0 1 0 0 0 1
  TH1=0; //计数器初始值
  TL1=0;
  TH0--(50000/256); //定时器 t0 定时 50ms
  TL0--(50000%256);

  EA=1; // IE=0X00; //EA - ET1 ES ET1 EX1 ET0 EX0
  ET0=1; // 1 0 0 0 0 0 1 0
  TR1=1;
  TR0=1;
  TF0=1;
}
//=====
void delay(uint k) //延时程序
{
  uint data i, j;
  for(i=0; i<k; i++)
  {
    for(; j<121; j++) {};
  }
}
//=====
void display(void) //数码管显示
{
  P1=LK[z/1000]; P2=LK1[0]; delay(10);
  P1=LK[(z/100)%10]; P2=LK1[1]; delay(10);
  P1=LK[(z%100)/10]; P2=LK1[2]; delay(10);
  P1=LK[z%10]; P2=LK1[3]; delay(10);
}
//=====
void main(void) //主程序开始
{
  uint temp1, temp2;
  init(); //调用 init 初始化子函数

  for(;;)
  {

```

```

        temp1=TL1;temp2=TH1;
        counter=(temp2<<8)+temp1;    //读出计数器值并转化为十进制
        //z=counter;
        display();
    }                                //无限循环语句结束
}                                    //主程序结束
//=====
// uint chushi=60;
void timer0(void) interrupt 1 using 1
{

    TH0=-(50000/256);    //定时器 t0 定时 50ms
    TL0=-(50000%256);

    // chushi--;
    // if(chushi<=0) {
        z=counter /0.5 ;    //读出速度
    //}
    TH0=0;                //每 50MS 清一次定时器
    TL1=0;
}

```