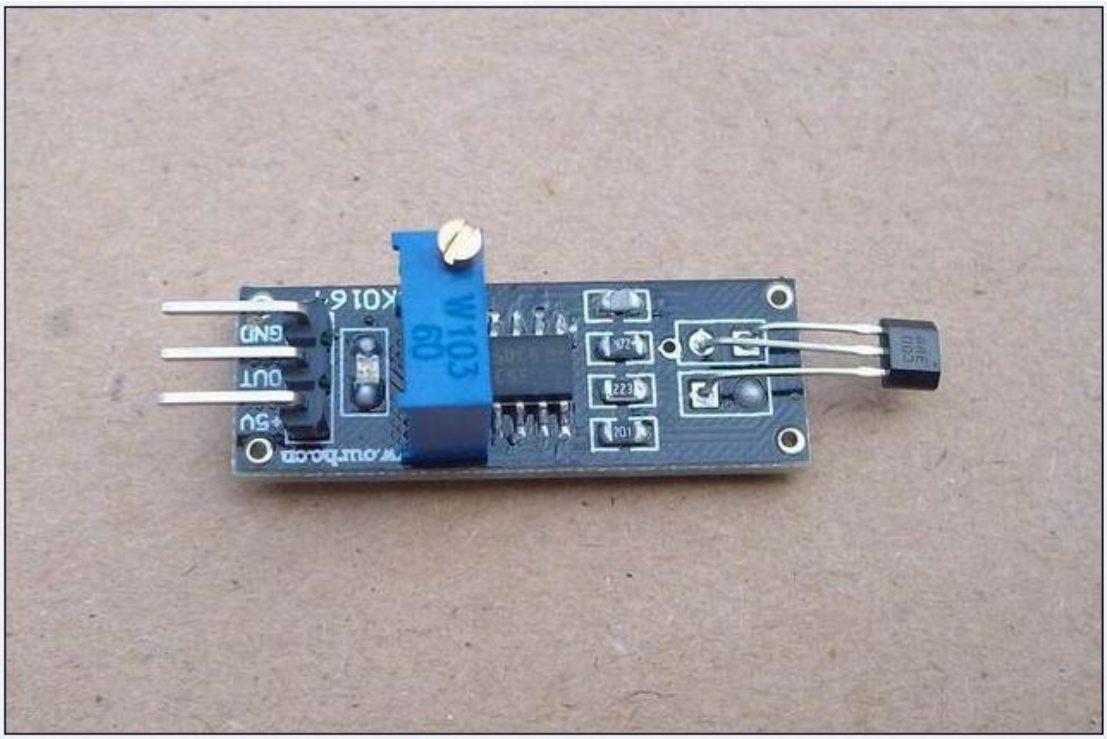
霍尔传感器使用说明书



简要说明：

一、长尺寸：32mm X 宽 11mm X 高 20mm

二、主要芯片：LM393、3144 霍尔传感器

三、工作电压：直流 5 伏

四、特点：

1、具有信号输出指示。

2、单路信号输出。

3、输出有效信号为低电平。

4、灵敏度可调（精调）。

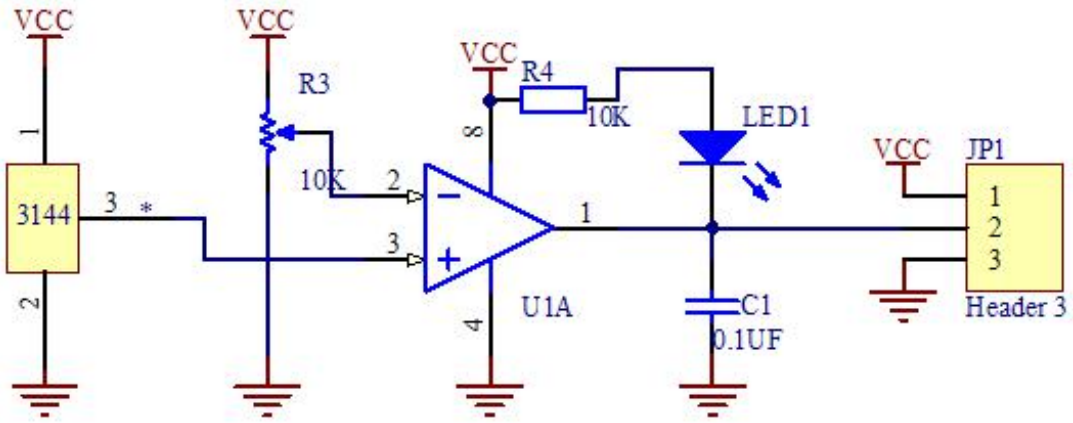
5、有磁场切割就有信号输出

6、电路板输出开关量！（可直接接单片机）

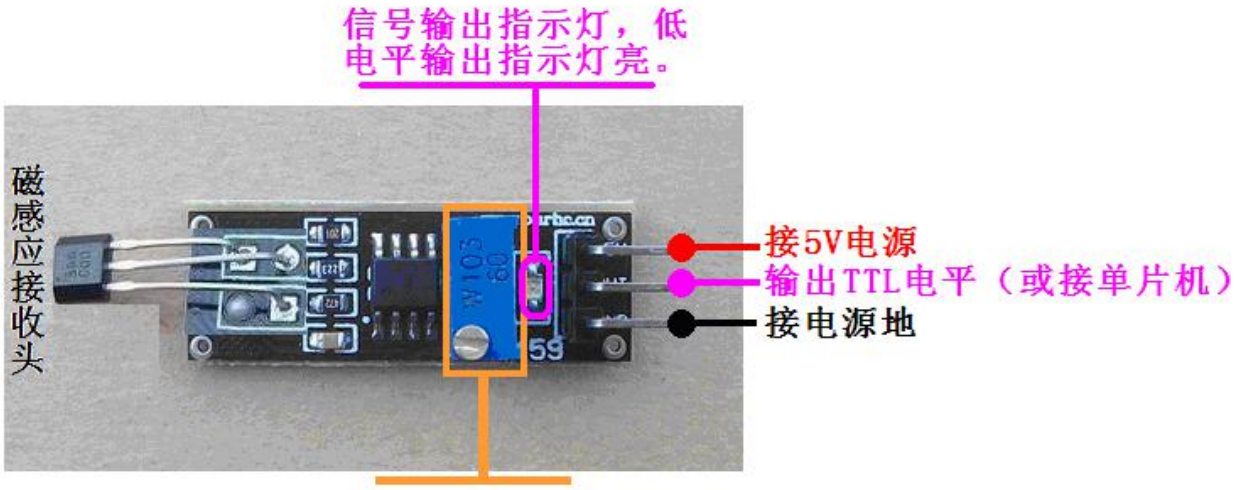
7、可用于电机测速/位置检测等场合

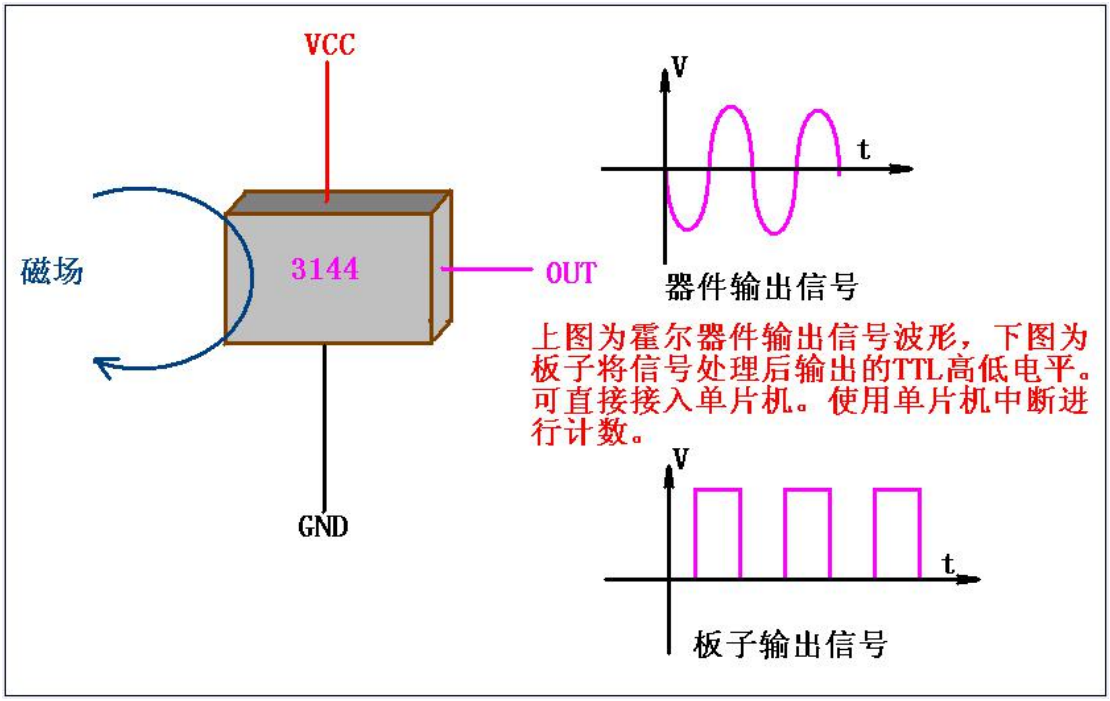
适用场合：单片机学习、电子竞赛、产品开发、毕业设计。。。

【原理图】



【图片展示】



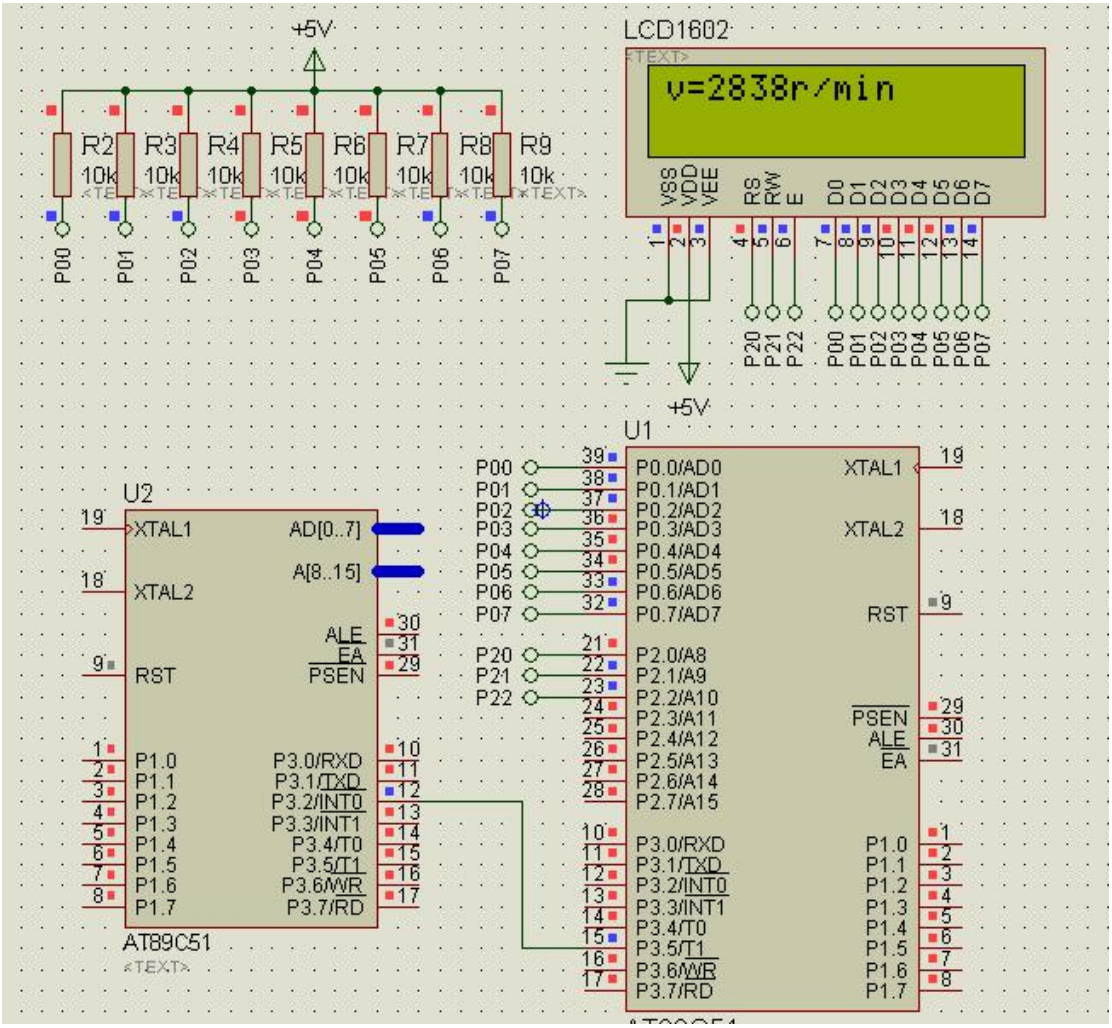


【与单片机连接测试程序】

```
/*********************************************************************
                              汇诚科技
实现功能:此版配套测试程序
使用芯片：AT89S52
晶振：11.0592MHZ
波特率：9600
编译环境：Keil
作者：zhangxinchunleo
网站：www.ourhc.cn
淘宝店：汇诚科技  http://shop36330473.taobao.com
【声明】此程序仅用于学习与参考，引用请注明版权和作者信息！
*********************************************************************/
/*********************************************************************
说明：1、 当测量浓度大于设定浓度时，单片机 IO 口输出低电平
*********************************************************************/
#include<reg52.h>                 //库文件
#define uchar unsigned char//宏定义无符号字符型
#define uint unsigned int    //宏定义无符号整型

/*********************************************************************
                              I/O 定义
*********************************************************************/
sbit LED=P1^0;   //定义单片机 P1 口的第 1 位 （即 P1.0）为指示端
```

```
sbit DOUT=P2^0;        //定义单片机 P2 口的第 1 位 （即 P2.0）为传感器的输入端
/*********************************************************************
                                延时函数
*********************************************************************/
void delay()//延时程序
{
uchar m,n,s;
for(m=20;m>0;m--)
for(n=20;n>0;n--)
for(s=248;s>0;s--);
}
/*********************************************************************
                                主函数
*********************************************************************/
void main()
{
      while(1)     //无限循环
      {
       LED=1;        //熄灭 P1.0 口灯
       if(DOUT==0)//当浓度高于设定值时 ， 执行条件函数
       {
      delay();//延时抗干扰
            if(DOUT==0)//确定  浓度高于设定值时 ， 执行条件函数
           {
             LED=0;        //点亮 P1.0 口灯
            }
       }
      }
}
/*********************************************************************
                                结束
*********************************************************************/
```



【与单片机连接测速参考程序】

```
/*********************************************************************
                                汇诚科技
实现功能: 电机转速表设计
使用芯片：AT89S52
晶振：11.0592MHZ
波特率：9600
编译环境：Keil
作者：zhangxinchunleo
网站：www.ourhc.cn
淘宝店：汇诚科技  http://shop36330473.taobao.com
【声明】此程序仅用于学习与参考，引用请注明版权和作者信息！
#include<reg52.h>       //包含单片机寄存器的头文件
#include<intrins.h>    //包含_nop_()函数定义的头文件
sbit RS=P2^0;       //寄存器选择位，将 RS 位定义为 P2.0 引脚
sbit RW=P2^1;       //读写选择位，将 RW 位定义为 P2.1 引脚
```

```
sbit E=P2^2;        //使能信号位，将 E 位定义为 P2.2 引脚
sbit BF=P0^7;       //忙碌标志位，，将 BF 位定义为 P0.7 引脚
unsigned char code digit[ ]={"0123456789"}; //定义字符数组显示数字
unsigned int v;     //储存电机转速
unsigned char count;   //储存定时器 T0 中断次数
bit flag;           //计满 1 秒钟标志位
/*************************************************
函数功能：延时 1ms
(3j+2)*i=(3×33+2)×10=1010(微秒)，可以认为是 1 毫秒
*************************************************/
void delay1ms()
{
    unsigned char i,j;
        for(i=0;i<10;i++)
          for(j=0;j<33;j++)
            ;
 }
 /*************************************************
函数功能：延时若干毫秒
入口参数：n
*************************************************/
 void delay(unsigned char n)
 {
    unsigned char i;
      for(i=0;i<n;i++)
          delay1ms();
 }

/*************************************************
函数功能：判断液晶模块的忙碌状态
返回值：result。result=1，忙碌;result=0，不忙
*************************************************/
 unsigned char BusyTest(void)
   {
     bit result;
      RS=0;          //根据规定，RS 为低电平，RW 为高电平时，可以读状态
     RW=1;
     E=1;            //E=1，才允许读写
     _nop_();    //空操作
     _nop_();
     _nop_();
     _nop_();    //空操作四个机器周期，给硬件反应时间
     result=BF;   //将忙碌标志电平赋给 result
    E=0;             //将 E 恢复低电平
```

```
    return result;
  }
/*************************************************
函数功能：将模式设置指令或显示地址写入液晶模块
入口参数：dictate
*************************************************/
void WriteInstruction (unsigned char dictate)
{
    while(BusyTest()==1); //如果忙就等待
      RS=0;                          //根据规定，RS 和 R/W 同时为低电平时，可以写入指令
      RW=0;
      E=0;                           //E 置低电平(根据表 8-6，写指令时，E 为高脉冲，
                                     //  就是让 E 从 0 到 1 发生正跳变，所以应先置"0"
      _nop_();
      _nop_();                      //空操作两个机器周期，给硬件反应时间
      P0=dictate;                    //将数据送入 P0 口，即写入指令或地址
      _nop_();
      _nop_();
      _nop_();
      _nop_();                       //空操作四个机器周期，给硬件反应时间
      E=1;                           //E 置高电平
      _nop_();
      _nop_();
      _nop_();
      _nop_();                       //空操作四个机器周期，给硬件反应时间
      E=0;                           //当 E 由高电平跳变成低电平时，液晶模块开始执行命令
 }
/*************************************************
函数功能：指定字符显示的实际地址
入口参数：x
*************************************************/
 void WriteAddress(unsigned char x)
 {
     WriteInstruction(x|0x80); //显示位置的确定方法规定为"80H+地址码 x"
 }
/*************************************************
函数功能：将数据(字符的标准 ASCII 码)写入液晶模块
入口参数：y(为字符常量)
*************************************************/
 void WriteData(unsigned char y)
 {
    while(BusyTest()==1);
      RS=1;                 //RS 为高电平，RW 为低电平时，可以写入数据
      RW=0;
```

```
        E=0;                    //E 置低电平(根据表 8-6，写指令时，E 为高脉冲，
                                //  就是让 E 从 0 到 1 发生正跳变，所以应先置"0"
        P0=y;                   //将数据送入 P0 口，即将数据写入液晶模块
        _nop_();
        _nop_();
        _nop_();
      _nop_();          //空操作四个机器周期，给硬件反应时间
        E=1;              //E 置高电平
        _nop_();
        _nop_();
        _nop_();
      _nop_();          //空操作四个机器周期，给硬件反应时间
      E=0;                //当 E 由高电平跳变成低电平时，液晶模块开始执行命令
 }
/*************************************************
函数功能：对 LCD 的显示模式进行初始化设置
*************************************************/
void LcdInitiate(void)
{
   delay(15);                   //延时 15ms，首次写指令时应给 LCD 一段较长的反应时间
   WriteInstruction(0x38);  //显示模式设置：16×2 显示，5×7 点阵，8 位数据接口
     delay(5);                      //延时 5ms  ，给硬件一点反应时间
   WriteInstruction(0x38);
     delay(5);
     WriteInstruction(0x38); //连续三次，确保初始化成功
     delay(5);
     WriteInstruction(0x0c);  //显示模式设置：显示开，无光标，光标不闪烁
     delay(5);
     WriteInstruction(0x06);  //显示模式设置：光标右移，字符不移
     delay(5);
     WriteInstruction(0x01);  //清屏幕指令，将以前的显示内容清除
     delay(5);

 }
/*****************************************************************************
 函数功能：显示速度提示符
*****************************************************************************/
void display_sym(void)
{
   WriteAddress(0x00);  //写显示地址,将在第 1 行第 1 列开始显示
     WriteData('v');          //将字符常量 v 写入 LCD
     WriteData('=');          //将字符常量=写入 LCD

 }
```

```
 /*****************************************************************************
 函数功能：显示速度数值
*****************************************************************************/
void display_val(unsigned int x)
{
 unsigned char i,j,k,l;        //j,k,l 分别储存温度的百位、十位和个位
      i=x/1000;               //取千位
      j=(x%1000)/100;     //取百位
   k=(x%100)/10;        //取十位
   l=x%10;                  //取个位
      WriteAddress(0x02);         //写显示地址,将在第 1 行第 3 列开始显示
      WriteData(digit[i]);       //将千位数字的字符常量写入 LCD
      WriteData(digit[j]);       //将百位数字的字符常量写入 LCD
      WriteData(digit[k]);       //将十位数字的字符常量写入 LCD
      WriteData(digit[l]);       //将个位数字的字符常量写入 LCD

 }
/*******************************************************
函数功能：显示速度单位“r/min”
*******************************************************/
void display_unit(void)
{
   WriteAddress(0x06);       //写显示地址,将在第 2 行第 7 列开始显示
      WriteData('r');       //将字符常量 r 写入 LCD
      WriteData('/');       //将字符常量/写入 LCD
      WriteData('m');       //将字符常量 m 写入 LCD
      WriteData('i');       //将字符常量 i 写入 LCD
      WriteData('n');       //将字符常量 n 写入 LCD
 }
/*******************************************************
函数功能：主函数
*******************************************************/
 void main(void)
 {
   LcdInitiate();                  //调用 LCD 初始化函数
   TMOD=0x51;                        //定时器 T1 工作于计数模式 1，定时器 T0 工作于计时模式 1；
      TH0=(65536-46083)/256;      //定时器 T0 的高 8 位设置初值，每 50ms 产生一次中断
      TL0=(65536-46083)%256;      //定时器 T0 的低 8 位设置初值，每 50ms 产生一次中断
      EA=1;                            //开总中断
      ET0=1;                           //定时器 T0 中断允许
      TR0=1;                           //启动定时器 T0
      count=0;                        //将 T0 中断次数初始化为 0
      display_sym();                 //显示速度提示符
  display_val(0000);             //显示器工作正常标志
```

```
   display_unit();                  //显示速度单位
        while(1)                         //无限循环
         {
         TR1=1;                  //定时器 T1 启动
               TH1=0;                    //定时器 T1 高 8 位赋初值 0
               TL1=0;                    //定时器 T1 低 8 位赋初值 0
               flag=0;                  //时间还未满 1 分钟
               while(flag==0)       //时间未满等待
                   ;
              v=(TH1*256+TL1)*60/16; //计算速度，每周产生 16 个脉冲
              display_val(v);              //显示速度
     }
}
/*********************************************************
函数功能：定时器 T0 的中断服务函数
*********************************************************/
void Time0(void ) interrupt 1 using 1 //定时器 T0 的中断编号为 1，使用第 1 组工作寄存器
   {
      count++;                 //T0 每中断 1 次，count 加 1
        if(count==20)         //若累计满 20 次，即计满 1 秒钟
         {
          flag=1;             //计满 1 秒钟标志位置 1
               count=0;            //清 0，重新统计中断次数
         }
      TH0=(65536-46083)/256;     //定时器 T0 高 8 位重新赋初值
        TL0=(65536-46083)%256;     //定时器 T0 低 8 位重新赋初值

   }
/*********************************************************************
                                    结束
*********************************************************************/
```