

(SKU:RB-03T005)红外接收模块

来自ALSROBOT Wiki

目录

- 1 产品概述
- 2 规格参数
- 3 引脚定义
 - 3.1 连接示意图
- 4 使用方法
 - 4.1 使用硬件
 - 4.2 接线方法
 - 4.3 例子程序
 - 4.3.1 发送部分程序
 - 4.3.2 接收部分程序
 - 4.4 程序效果
- 5 视频演示
- 6 产品相关推荐
 - 6.1 产品购买地址
 - 6.2 周边产品推荐
 - 6.3 相关问题解答
 - 6.4 相关学习资料



产品概述

IR Receiver Module是一款Arduino兼容的38KHz红外线接收模块，可接收标准38KHz调制的遥控器信号，通过对Arduino进行编程，即可实现对遥控器信号的解码操作。可使用Arduino制作一款带学习功能的万能遥控器。

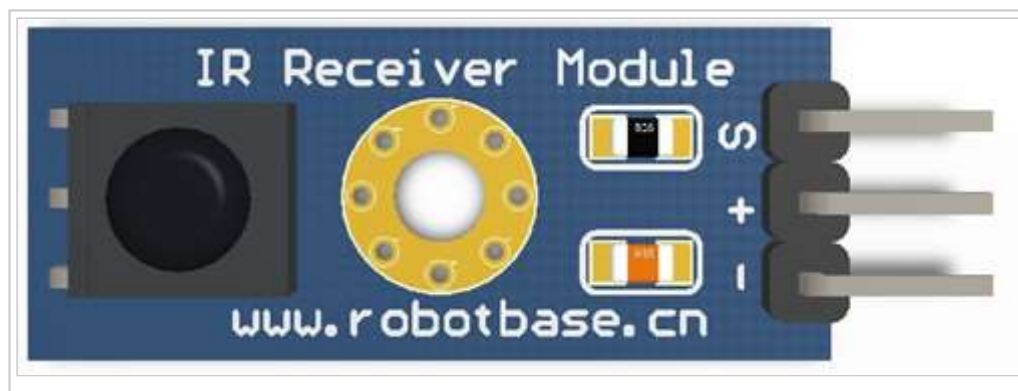
规格参数

- 供电电压: 5V
- 工作环境: -25~+85℃
- 储存温度: -30~+100 °C
- 接收频率: 38KHz
- 模块尺寸: 13.7mm×27.8mm
- 模块重量: 2g

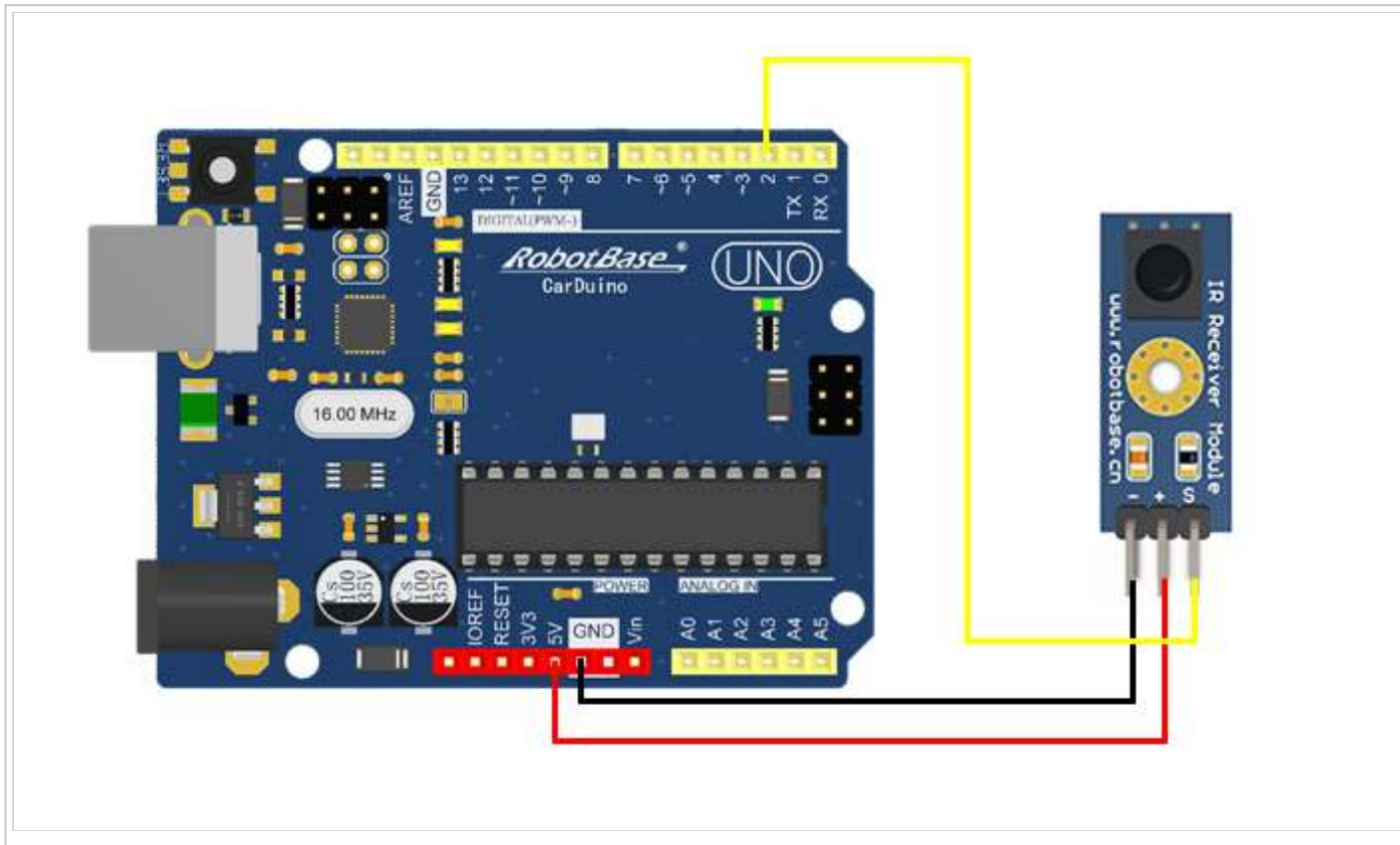
引脚定义

红外接收传感器引脚的定义是

- S: 输出信号
- +: 电源(VCC)
- -: 地(GND)



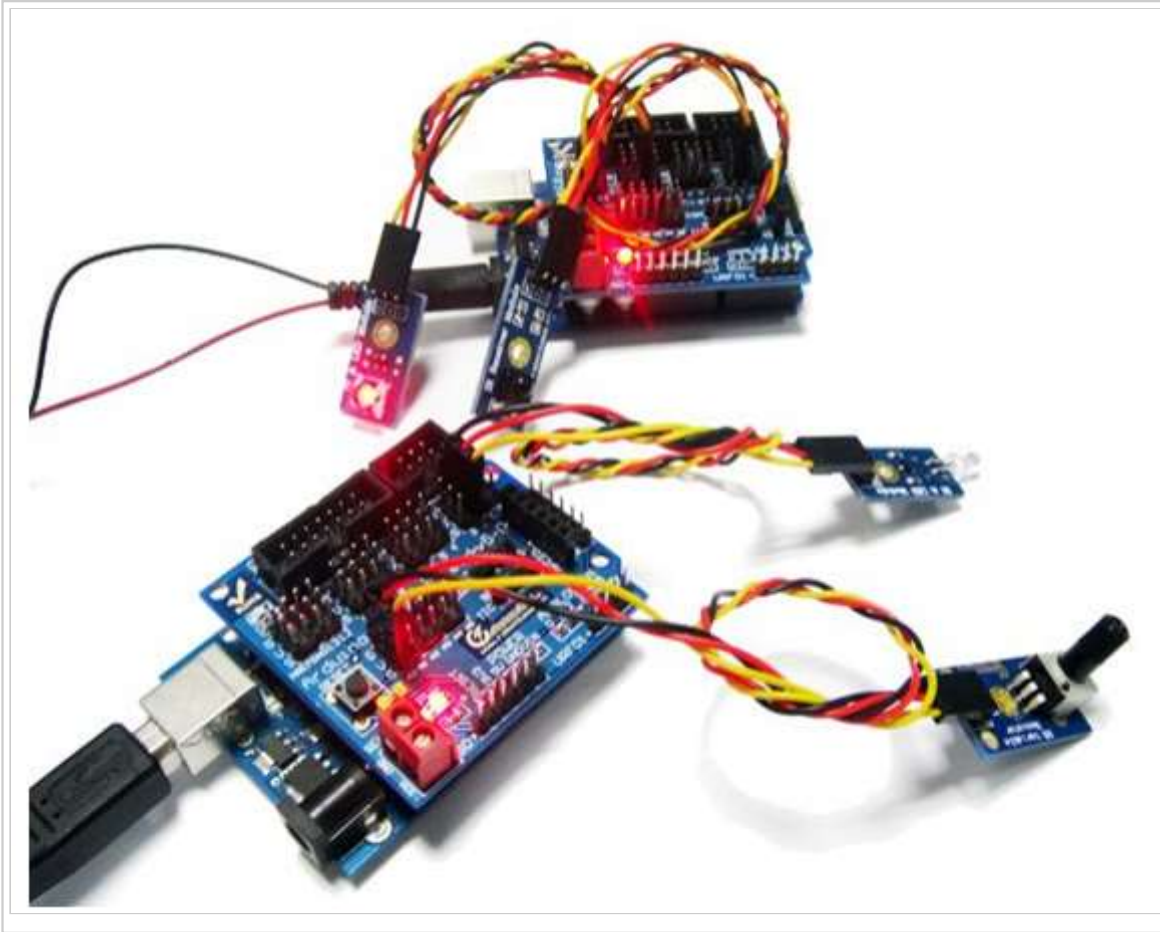
连接示意图



使用方法

使用硬件

- Carduino UNO R3 控制器 (<http://www.alsrobot.cn/goods-546.html>) * 2
- Arduino 传感器扩展板V5.0 (<http://www.alsrobot.cn/goods-147.html>) * 2
- 红外发射模块 (<http://www.alsrobot.cn/goods-155.html>) * 1个
- 红外接收模块 (<http://www.alsrobot.cn/goods-157.html>) * 1个
- 电位计模块 (<http://www.alsrobot.cn/goods-167.html>) * 1个
- LED 发光模块 (<http://www.alsrobot.cn/goods-133.html>) * 1个
- USB 数据通信线 (<http://www.alsrobot.cn/goods-90.html>) * 2个



接线方法

产品	UNO引脚
红外发射模块	数字接口2
红外接收模块	数字接口2
旋转角度电位计	模拟接口0
LED模块	数字接口9

将发送部分代码编译后下载到作为发送部分的Arduino里，将接收部分代码编译后下载到作为接收部分的Arduino里，就可以在串口助手窗口上显示接收到的当前值（注：串口助手波特率调到115200）。Arduino实验代码如下。

例子程序

程序安装前，需要先进行库文件的安装，点此下载 (<http://pan.baidu.com/s/1sjtaCqT>) IRremote 库文件

发送部分程序

```
#define ADD 0x00
int IR_S = 2;    // 定义数字口2 为发射模块接口
int a;
void setup()
{
    pinMode(IR_S, OUTPUT); //定义IR_S为输出模式
    Serial.begin(115200); //定义频率为115200
}
void loop()
{
    uint8_t dat,temp;
    {
        a=analogRead(0); // 读取模拟口0 的值
        temp =a/4;
        Serial.println(temp,DEC ); // 将读取的数值打印到串口上
        IR_Send38KHZ(280,1); //发送9ms 的起始码
        IR_Send38KHZ(140,0); //发送4.5ms 的结果码

        IR_Sendcode(ADD); //用户识别码
        dat=~ADD;
        IR_Sendcode(dat); //用户识别码反码
        IR_Sendcode(temp); // 操作码
        dat=~temp;
        IR_Sendcode(dat); //操作码反码
        IR_Send38KHZ(21,1); // 发送结束码
    }
    delay(200);
}
void IR_Send38KHZ(int x,int y) //产生38KHZ红外脉冲
{
    for(int i=0;i<x;i++) //15=386US
    {
        if(y==1)
        {
```

```
        digitalWrite(IR_S, 1);
        delayMicroseconds(9);
        digitalWrite(IR_S, 0);
        delayMicroseconds(9);
    }
    else
    {
        digitalWrite(IR_S, 0);
        delayMicroseconds(20);
    }
}
}

void IR_Sendcode(uint8_t x)
{
    for(int i=0;i<8;i++)
    {
        if((x&0x01)==0x01)
        {
            IR_Send38KHZ(23, 1);
            IR_Send38KHZ(64, 0);
        }
        else
        {
            IR_Send38KHZ(23, 1);
            IR_Send38KHZ(21, 0);
        }
        x=x>>1;
    }
}
```

此代码的功能是从模拟口0 读取电位计的值，并通过红外发射头将读取的数值发送出去。

接收部分程序

```
#define IR_LED 2    //IR 接收数字口2
#define MAX 128
#define MICRO_STEP 10
#define IDLE_PULSE 4000
unsigned long pulses[MAX];
unsigned char IRCOM[7];
unsigned long z;
int w;
byte f=B00000000; // 定义f 为位
int n;
int ledpin=9; // 定义数字口9 为LED 模块接口
void setup()
{

```

```

pinMode(IR_LED, INPUT);
Serial.begin(115200);
pinMode(ledpin, OUTPUT); //定义ledpin 为输出模式
}

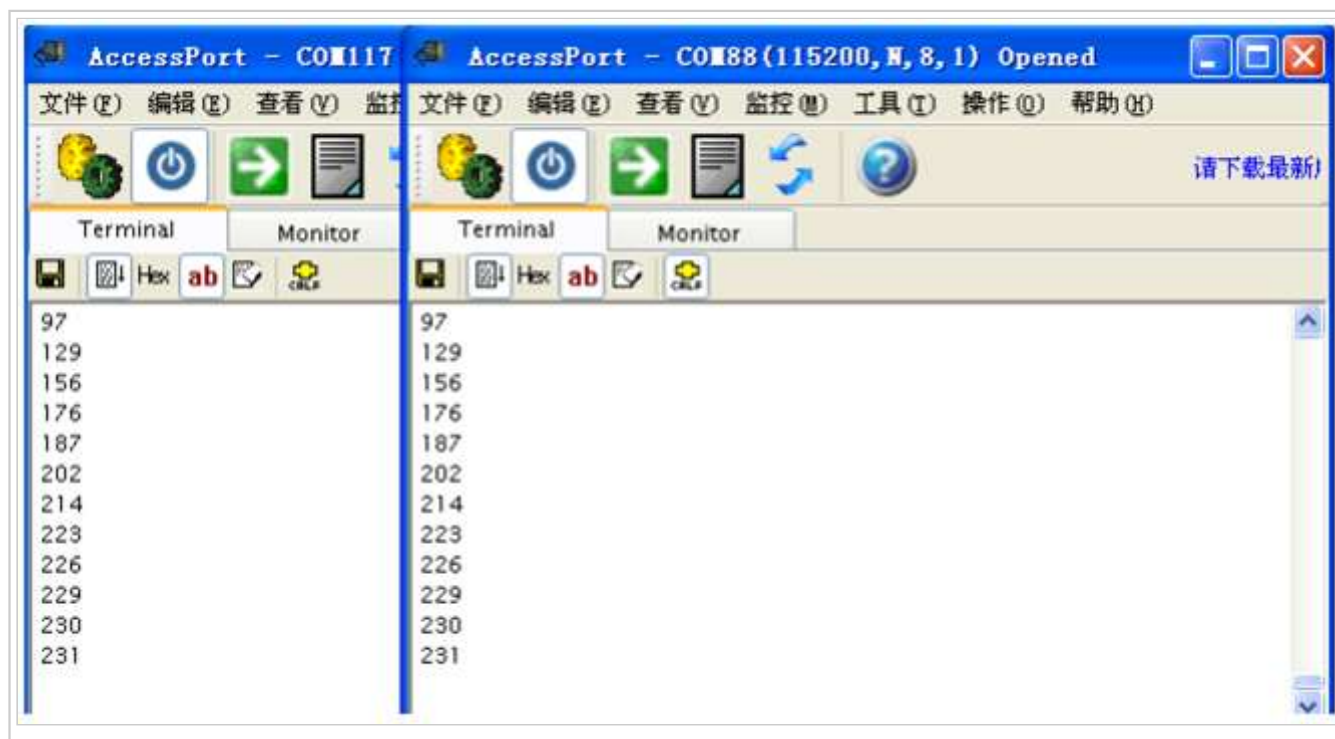
void loop()
{
  if( digitalRead(IR_LED) == LOW)
  {
    // 开始接收数据
    int count = 0;
    int exit = 0;
    while(!exit)
    {
      while( digitalRead(IR_LED) == LOW )
        delayMicroseconds(MICRO_STEP);
      unsigned long start = micros();
      int max_high = 0;
      while( digitalRead(IR_LED) == HIGH )
      {
        delayMicroseconds(MICRO_STEP);
        max_high += MICRO_STEP;
        if( max_high > IDLE_PULSE )
        {
          exit = 1;
          break;
        }
      }
      unsigned long duration = micros() - start;
      pulses[count++] = duration;
    }
    for(int i=3; i<4; i++)
    {
      for(int j=0; j<8; j++)
      {
        if(pulses[ i*8+j+1] < IDLE_PULSE)
        {
          IRCOM[i]=IRCOM [i] >> 1;
          if((pulses[i*8+j+1])>1000)
            {IRCOM[i] = IRCOM[i] | 0x80;}
        }
        z= pulses[i*8+j+1]; //将接收到的脉冲数据转换成十进制
        if(z<800)
          w=100000000; //如果Z 小于800  w=100000000
        else
          w=000000000; //如果Z 大于800  w=000000000
        f=f>>1; // 将f 右移1 位
        f=f+w;
      }
    }
    n=int(f);
  }
}

```

```
Serial.print(n); // 将接收到的数据打印到串口上
analogWrite(ledpin,n); //将接收到的数据写入ledpin 接口，控制LED 亮度
}
```

程序效果

此代码的功能是通过红外接收头接收发送部分发送出的电位计的值，将读到的值来控制LED的亮度，并通过串口助手显示当前接收到的值。如下图所示左侧串口助手（串口号为：117）显示的是红外发送部分当前发送出的电位计的值，右侧串口助手（串口号为：88）显示的是红外接收部分接收到的数值。这样就实现了通过旋转角度电位计模块无线控制接收端的LED亮度。



视频演示