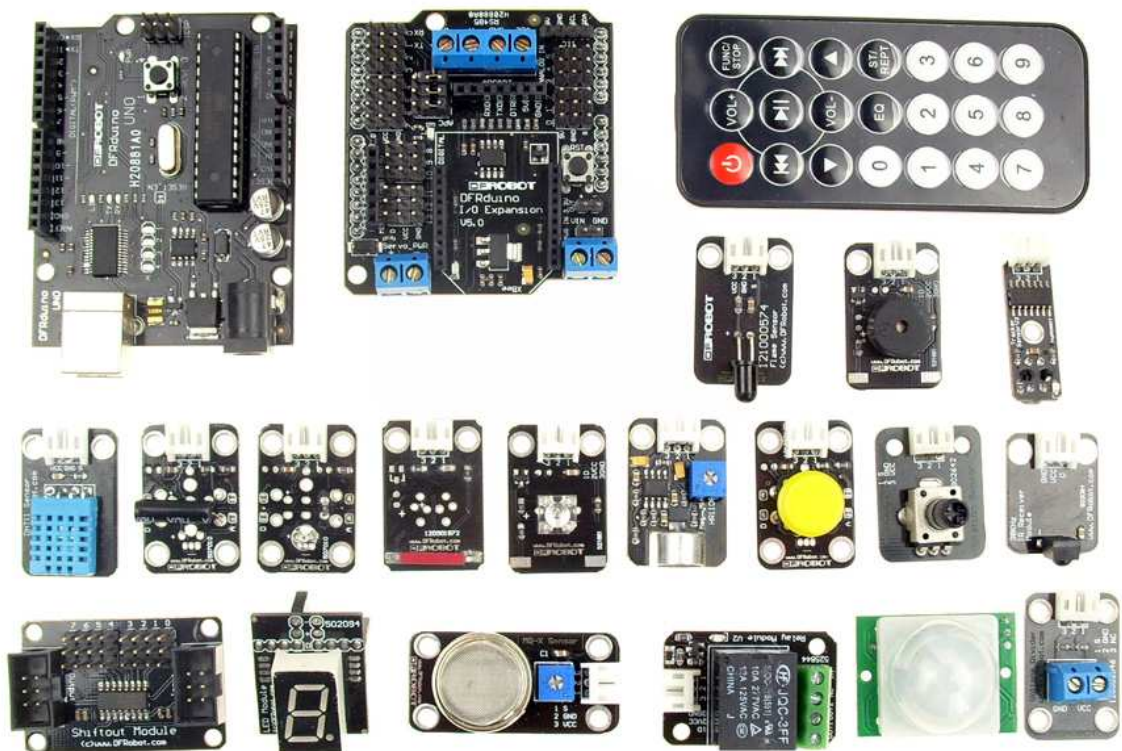




## Arduino 中级套件使用教程



DFRobot Version 1.1

## 商标声明

DFRobot 与图形为北京龙凡汇众机器人科技有限公司注册商标。

本公司基于对产品完善与升级，随时可能更改本资料或其中提到的产品内容恕不另行通知。未经本公司书面同意或授权，不得擅自盗用、复制、出版权使用手册中局部或全部内容。

## 免责声明

使用者在使用本产品做任何应用时（如实验、竞赛、二次开发）须自行承担风险。公司对因使用本产品造成的直接、间接或附带伤害（包括人身安全、利润、信誉损失等）不负任何责任。未满 14 周岁儿童须在成人陪同下方可使用本产品。

## 注意事项

- A. 注意！在没有认真阅读本说明之前，请勿给模块加电！错误接线将导致模块永久性损坏或烧毁微控制器。
- B. 注意！请认真查看引脚功能说明，正确接线！请勿将电源反接，否则将导致模块永久性损坏。
- C. 注意！请勿使用超出额定电压的电源！保证电源的稳定，如果出现高压脉冲，可能会导致微控制器永久性损坏。
- D. 注意！本产品无防水防潮功能，请在干燥环境下保存或使用！不可将重物堆积在上面。

## 勘误说明

为了准确的传达产品的使用信息，我们花费了大量时间与精力，然而难免出现疏漏，在此欢迎使用者为我们指正错误，如发现错误请发邮件至 [service@dfrobot.com](mailto:service@dfrobot.com)。为了使手册更加完善，我们会持续更新完善手册中的内容，并将手册实时的更新在 [wiki](#) 上，请大家继续关注我们！

公司网址：[www.dfrobot.com.cn](http://www.dfrobot.com.cn)

公司博客：<http://blog.dfrobot.com.cn/>

公司论坛：<http://www.dfrobot.com.cn/community/>

公司 WIKI：<http://wiki.dfrobot.com.cn>

公司地址：

北京市海淀区知春路 6 号锦秋家园七号楼 2204 室（北京龙凡汇众机器人科技有限公司）

上海市浦东新区郭守敬路 498 号 9 号楼 501 室（上海智位机器人有限公司）

成都市温江区海峡科技园蓉台大道南段 99 号 D 栋 4 楼（成都极趣科技有限公司）

## 概述

在这个科技高速发展的现代社会，我们的身边出现了越来越多的机械智能的玩意，激发了众多热爱制作的人们的兴趣，对于熟练使用 Arduino 的朋友都知道，Arduino 是一块基于开放原始代码的 Simple i/o 平台，并且具有开发语言与开发环境都很简单、易理解的特点。让您可以快速使用 Arduino 做出有趣的东西，使用起来非常方便，但是，对于刚接触 Arduino 的人们来说，如何使用？初学者应该配置哪些基础的模块配件来学习？因此，本公司推出这款 Arduino 互动中级套件，该套件是基于 DFRduino UNO 控制板的，配置了扩展板以及常用的数字、模拟传感器等。对于初学者，拥有这款套件让您可以快速使用 Arduino 语言与 Flash 或 Processing... 等软件互动。对于熟悉 arduino 使用的朋友们来说，您可以发挥您的想象，用套件里的各个模块 DIY 出更有趣的互动作品。

DFRduino UNO 与 Arduino 完全兼容。Arduino 的 IO 使用的排母，做互动作品需要面包板和针线搭配才能进行，此款套件特别配备了传感器扩展板，使用传感器直接接到扩展板上相应的口就行了，免去了使用面包线连接电路的繁琐。

本文简单介绍了 Arduino 的语法、Arduino 的 USB 驱动安装，并分别介绍了 10 多种传感器的使用方法，最后举 3 个综合应用的例子抛砖引玉，希望大家通过本文能学到有用的知识，更希望能启发您制作出有创意的作品。

## 目录

概述.....	2
目录.....	3
产品清单.....	4
Arduino 语言介绍.....	5
结构.....	7
功能.....	7
Arduino IDE 软件使用介绍.....	10
安装 USB 驱动.....	10
传感器实验篇.....	14
传感器信号线的使用.....	14
数字温湿度传感器 DHT11 的温湿度检测.....	16
环境光线传感器模块实验.....	18
数字人体红外热释电运动传感器模块的实验.....	19
红外发送接收实验.....	20
MIC 声音传感器模块实验.....	24
气体传感器模块的实验.....	25
10A 大电流继电器模块实验.....	26
数字按钮模块实验.....	27
数字蜂鸣器模块实验.....	28
模拟旋转角度传感器实验.....	29
模拟寻线传感器实验.....	30
电压检测模块实验.....	31
火焰传感器实验.....	32
Shiftout Module 模块实验.....	34
数字食人鱼红色 LED 发光模块实验.....	37
数字贴片磁感应传感器实验.....	38
震动传感器模块实验.....	39
综合实验篇.....	40
基于 MIC 声音传感器模块和食人鱼 LED 模块的声光互动实验.....	40
红外遥控器控制 Shiftout LED Kiti 显示数字.....	41
电位器角度传感器控制蜂鸣器音调与 LED 灯亮度实验.....	46

## 产品清单

货号	产品名称	数量
DFR0176	DFRduino UNO 控制板	1 个
DFR0088	xbee 传感器扩展板 V5	1 个
DFR0067	数字温湿度传感器 DHT11	1 个
DFR0026	环境光传感器模块	1 个
SEN0018	数字人体红外热释电运动传感器模块	1 个
DFR0107	红外遥控器	1 个
	红外接收模块	1 个
DFR0034	MIC 声音传感器模块	1 个
DFR0049	气体传感器模块	1 个
DFR0017	10A 大电流继电器模块	1 个
DFR0029	数字按钮模块	1 个
DFR0032	数字蜂鸣器模块	1 个
DFR0058	模拟电位器角度传感器	1 个
SEN0017	Mini 寻线传感器	1 个
DFR0051	电压检测模块	1 个
DFR0076	火焰传感器	1 个
DFR0027	数字震动传感器	1 个
DFR0072	Shiftout Module	1 个
	数码管模块	1 个
DFR0031R	数字食人鱼红色 LED 发光模块	1 个
DFR0033	数字贴片磁感应传感器	1 个
FIT0141	6 节 5 号电池盒	1 个
FIT0056	USB 线	1 根
FIT0012	杜邦线	5 根

## Arduino 语言介绍

Arduino 语言是建立在 C/C++ 基础上的，其实也就是基础的 C 语言，Arduino 语言只不过把 AVR 单片机（微控制器）相关的一些寄存器参数设置等都函数化了，不用我们去了解他的底层，让不太了解 AVR 单片机（微控制器）的朋友也能轻松上手。

那么这里就简单的注释一下 Arduino 语言。

/\*\*\*\*\*\*基础 C 语言\*\*\*\*\*\*/

关键字：

- if
- if...else
- for
- switch case
- while
- do... while
- break
- continue
- return
- goto

语法符号：

- :
- { }
- //
- /\* \*/

运算符：

- =
- +
- -
- \*
- /
- %
- ==
- !=
- <
- >
- <=

- >=
- &&
- ||
- !
- ++
- --
- +=
- -=
- \*=
- /=

数据类型:

- boolean 布尔类型
- char 字符类型
- byte 字节类型
- int 整数类型
- unsigned int 无符号整型
- long 长整型
- unsigned long 无符号长整型
- float 实数类型
- double
- string
- array
- void

数据类型转换:

- char()
- byte()
- int()
- long()
- float()

常量:

- HIGH | LOW 表示数字 IO 口的电平, HIGH 表示高电平 (1), LOW 表示低电平 (0)。
- INPUT | OUTPUT 表示数字 IO 口的方向, INPUT 表示输入 (高阻态), OUTPUT 表示输出 (AVR 能提供 5V 电压 40mA 电流)。
- true | false true 表示真 (1), false 表示假 (0)。

/\*\*\*\*\*/

以上为基础 c 语言的关键字和符号, 有 c 语言基础的都应该了解其含义, 这里也不作过多的解释。

/\*\*\*\*\*Arduino 语言\*\*\*\*\*/

## 结构

- void setup() 初始化变量，管脚模式，调用库函数等
- void loop() 连续执行函数内的语句

## 功能

### 数字 I/O

- pinMode(pin, mode) 数字 IO 口输入输出模式定义函数，pin 表示为 0~13，mode 表示为 INPUT 或 OUTPUT。
- digitalWrite(pin, value) 数字 IO 口输出电平定义函数，pin 表示为 0~13，value 表示为 HIGH 或 LOW。比如定义 HIGH 可以驱动 LED。
- digitalRead(pin) 数字 IO 口读输入电平函数，pin 表示为 0~13，value 表示为 HIGH 或 LOW。比如可以读数字传感器。

### 模拟 I/O

- analogRead(pin) 模拟 IO 口读函数，pin 表示为 0~5 (Arduino Diecimila 为 0~5, Arduino nano 为 0~7)。比如可以读模拟传感器 (10 位 AD, 0~5V 表示为 0~1023)。
- analogWrite(pin, value) - **PWM** 数字 IO 口 PWM 输出函数，Arduino 数字 IO 口标注了 PWM 的 IO 口可使用该函数，pin 表示 3, 5, 6, 9, 10, 11, value 表示为 0~255。比如可用于电机 PWM 调速或音乐播放。

### 扩展 I/O

- shiftOut(dataPin, clockPin, bitOrder, value) SPI 外部 IO 扩展函数，通常使用带 SPI 接口的 74HC595 做 8 个 IO 扩展，dataPin 为数据口，clockPin 为时钟口，bitOrder 为数据传输方向 (**MSBFIRST** 高位在前，**LSBFIRST** 低位在前)，value 表示所要传送的数据 (0~255)，另外还需要一个 IO 口做 74HC595 的使能控制。
- unsigned long pulseIn(pin, value) 脉冲长度记录函数，返回时间参数 (us)，pin 表示为 0~13，value 为 HIGH 或 LOW。比如 value 为 HIGH，那么当 pin 输入为高电平时，开始计时，当 pin 输入为低电平时，停止计时，然后返回该时间。

### 时间函数

- unsigned long millis() 返回时间函数 (单位 ms)，该函数是指，当程序运行就开始计时并返回记录的参数，该参数溢出大概需要 50 天时间。
- delay(ms) 延时函数 (单位 ms)。
- delayMicroseconds(us) 延时函数 (单位 us)。

### 数学函数

- min(x, y) 求最小值
- max(x, y) 求最大值



- abs(x) 计算绝对值
- constrain(x, a, b) 约束函数，下限 a，上限 b，x 必须在 ab 之间才能返回。
- map(value, fromLow, fromHigh, toLow, toHigh) 约束函数，value 必须在 fromLow 与 toLow 之间和 fromHigh 与 toHigh 之间。
- pow(base, exponent) 开方函数，base 的 exponent 次方。
- sq(x) 平方
- sqrt(x) 开根号

### 三角函数

- sin(rad)
- cos(rad)
- tan(rad)

### 随机数函数

- randomSeed(seed) 随机数端口定义函数，seed 表示读模拟口 analogRead(pin) 函数。
- long random(max) 随机数函数，返回数据大于等于 0，小于 max。
- long random(min, max) 随机数函数，返回数据大于等于 min，小于 max。

### 外部中断函数

- attachInterrupt(interrupt, , mode) 外部中断只能用到数字 IO 口 2 和 3，interrupt 表示中断口初始 0 或 1，表示一个功能函数，mode: **LOW** 低电平中断，**CHANGE** 有变化就中断，**RISING** 上升沿中断，**FALLING** 下降沿中断。
- detachInterrupt(interrupt) 中断开关，interrupt=1 开，interrupt=0 关。

### 中断使能函数

- interrupts() 使能中断
- noInterrupts() 禁止中断

### 串口收发函数

- Serial.begin(speed) 串口定义波特率函数，speed 表示波特率，如 9600，19200 等。
- int Serial.available() 判断缓冲器状态。
- int Serial.read() 读串口并返回收到参数。
- Serial.flush() 清空缓冲器。
- Serial.print(data) 串口输出数据。
- Serial.println(data) 串口输出数据并带回车符。

/\*\*\*\*\*\*Arduino 语言库文件\*\*\*\*\*\*/

官方库文件 下载地址: <http://arduino.cc/en/Reference/Libraries>

- [EEPROM](#) - EEPROM 读写程序库
- [Ethernet](#) - 以太网控制器程序库
- [LiquidCrystal](#) - LCD 控制程序库
- [Servo](#) - 舵机控制程序库
- [SoftwareSerial](#) - 任何数字 IO 口模拟串口程序库
- [Stepper](#) - 步进电机控制程序库
- [Wire](#) - TWI/I2C 总线程序库
- [Matrix](#) - LED 矩阵控制程序库
- [Sprite](#) - LED 矩阵图象处理控制程序库

/\*\*\*\*\*/

## Arduino IDE 软件使用介绍

Arduino 的编译软件我们会提供最新的软件包，用户也可以直接到 Arduino 官网下载，官网软件下载地址 <http://arduino.cc/en/Main/Software>，这里推荐下载 0023 版本，软件包解压即可直接使用。

### 安装 USB 驱动

首先连接下载程序用的下载线。一段接 UNO 控制板，另一端与电脑 USB 接口连接。插好后，控制板的电源指示灯会被点亮，电脑会出现以下对话框：



- 选择从列表或指定位置安装，点击下一步，出现如下图：



- 然后点下一步，将出现如下图所示，电脑开始搜索 UNO 的驱动：

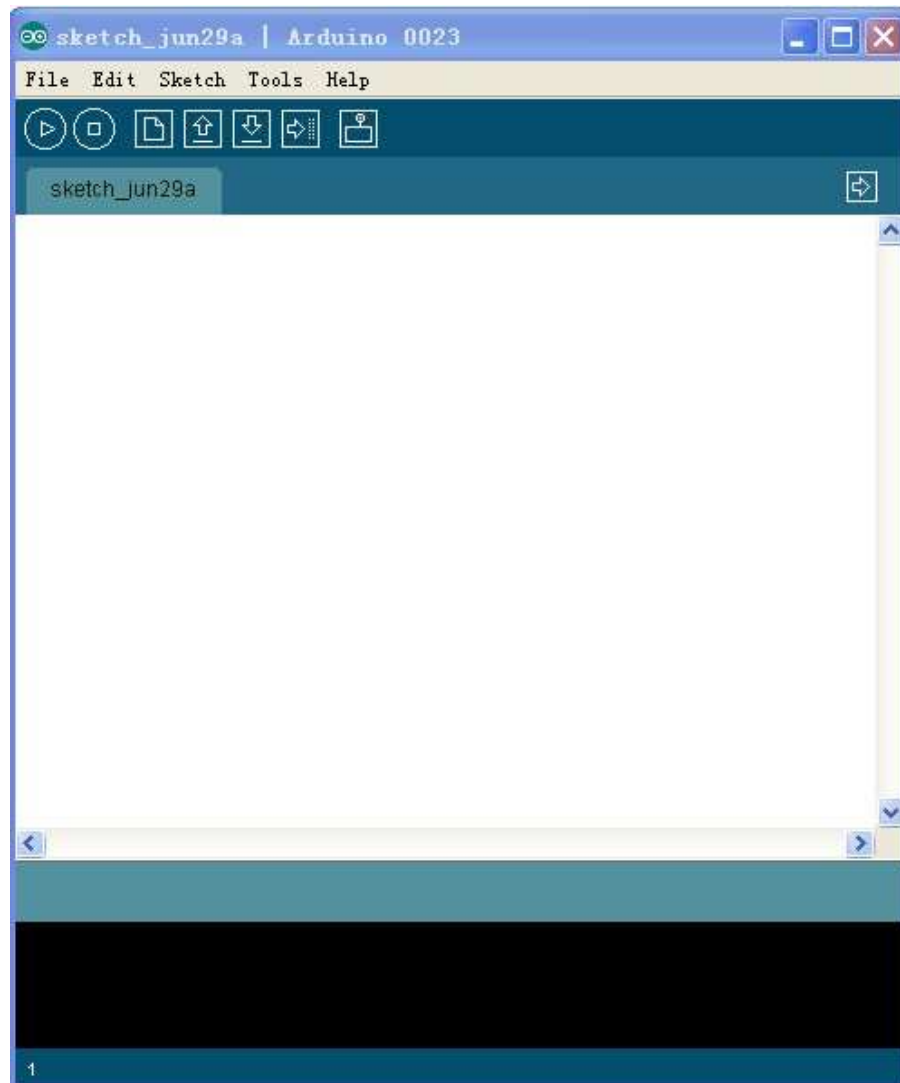


- 片刻之后，电脑自动安装驱动。如下图安装完成：



• 点击完成，这样驱动就安装好了，下次再将数据线插到电脑就不会出现安装驱动对话框了，插上数据线就可以下载程序了。

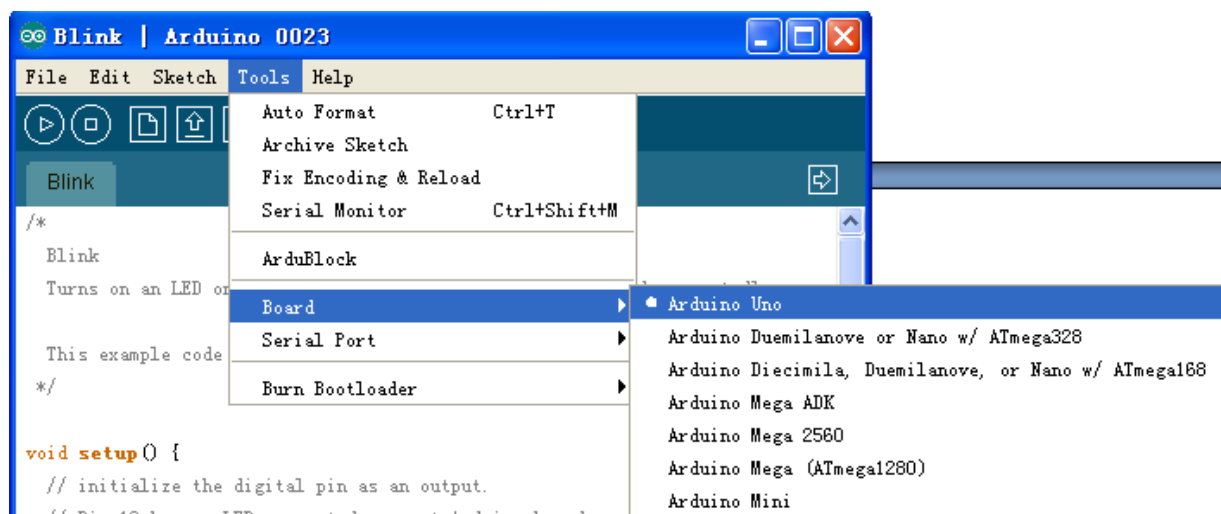
接下来我们就可以开始练手了，打开 Arduino0018。



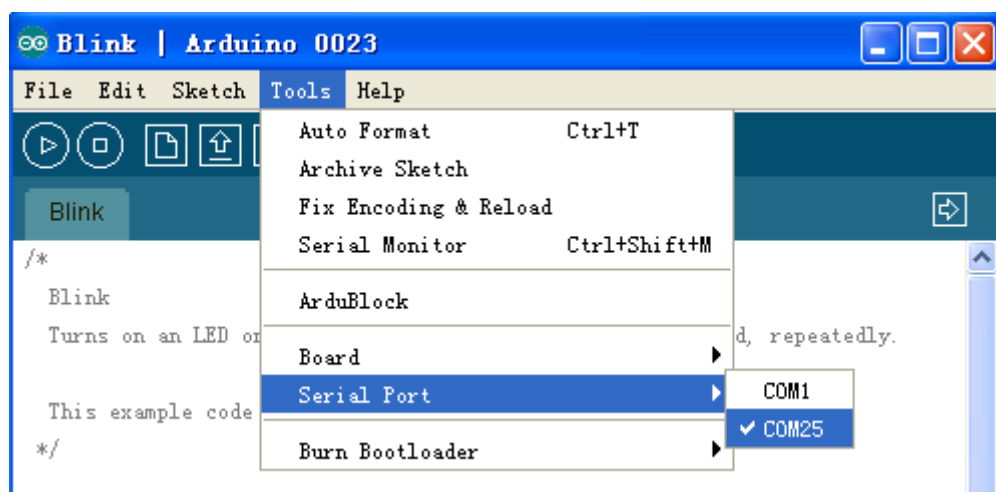
Arduino 0023 开发编译环境很简洁，各个功能键功能描述如下：



接下来点 Tools->Board 选择开发板型号，



这里我们使用的是 Arduino Duemilanove 328，然后点 Tools->Serial Port 选择串口，即 USB 映射的串口地址，



前面的工作做好后，我们就可以进入实验阶段了。

## 传感器实验篇

### 传感器信号线的使用

Arduino 电子积木传感器分为数字传感器和模拟传感器两类，数字传感器就接到数字口，模拟传感器就接到模拟口，两者绝对不能互换。

我们的数字传感器的线序是根据舵机的线序定义的（即信号、电源、地），由于数字传感器的电源在中间引脚上，所以

就不怕被插接反而烧坏。而模拟传感器的线序是根据夏普 GP2D12红外线的线序定义的（即信号、地、电源），由于电源不在中间，所以在使用时就需要注意线的方向。我们为2种传感器分别加工了数据线，在使用过程中，需要区分传感器的种类和传感器连接线的颜色。

数字传感器连接线（黑红绿）



模拟传感器连接线（红黑蓝）



数字与模拟传感器端口有 1, 2, 3 的数字标记，定义分别如下：

数字传感器：

- 1 脚-----信号数输出 (D)----对应数字传感器连接线绿色
- 2 脚-----电源正 (VCC) ----对应数字传感器连接线红色
- 3 脚-----电源负 (GND) ----对应数字传感器连接线黑色



模拟传感器：

1 脚-----信号数输出(S)----对应模拟传感器连接线蓝色

2 脚-----电源负(GND)----对应模拟传感器连接线黑色

3 脚-----电源正(VCC)----对应模拟传感器连接线红色

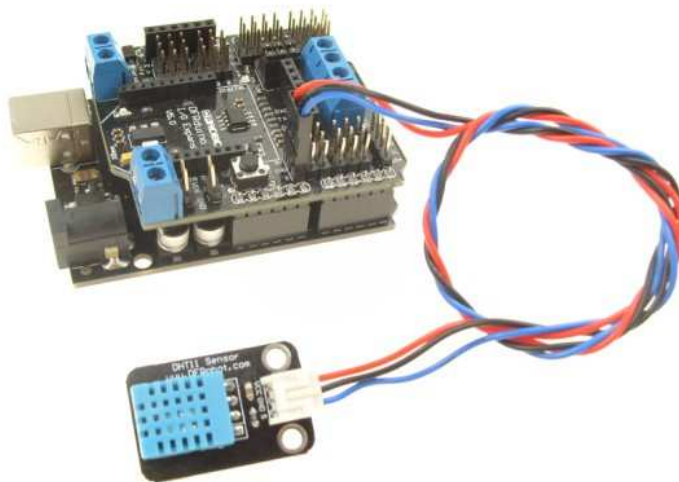
### 数字温湿度传感器 DHT11 的温湿度检测

DHT11 数字温湿度传感器是一款含有已校准数字信号输出的温湿度复合传感器。它应用专用的数字模块采集技术和温湿度传感技术，确保产品具有极高的可靠性与卓越的长期稳定性。传感器包括一个电阻式感湿元件和一个 NTC 测温元件，并与一个高性能 8 位单片机相连接。因此该产品具有品质卓越、超快响应、抗干扰能力强、性价比极高等优点。每个 DHT11 传感器都在极为精确的湿度校验室中进行校准。校准系数以程序的形式储存在 OTP 内存中，传感器内部在检测信号的处理过程中要调用这些校准系数。单线制串行接口，使系统集成变得简易快捷。超小的体积、极低的功耗，信号传输距离可达 20 米以上，使其成为各类应用甚至最为苛刻的应用场合的最佳选则。

DHT11 数字温湿度传感器模块为 3 针 PH2.0 封装。连接方便。值得注意的是，该模块的接口设计为模拟口接口，也就是将模拟口作为数字口用，可以不占用 Arduino 的原始数字口。

首先，把 DHT11 模块接到 UNO 控制板的模拟口 0。

接线图如下：



**Arduino 实验代码：**

```
#define DHT11_PIN 0      // ADC0 接到模拟口0
```

```
byte read_dht11_dat()
```

```
{
```

```
    byte i = 0;
```

```
byte result=0;
for(i=0; i< 8; i++){
    while(!(PINC & _BV(DHT11_PIN))); // wait for 50us
    delayMicroseconds(30);

    if(PINC & _BV(DHT11_PIN))
        result |= (1<<(7-i));
    while((PINC & _BV(DHT11_PIN))); // wait '1' finish
}
return result;
}

void setup()
{
    DDRC |= _BV(DHT11_PIN);
    PORTC |= _BV(DHT11_PIN);
    Serial.begin(19200);
    Serial.println("Ready");
}

void loop()
{
    byte dht11_dat[5];
    byte dht11_in;
    byte i;
    // start condition
    // 1. pull-down i/o pin from 18ms
    PORTC &= ~_BV(DHT11_PIN);
    delay(18);
    PORTC |= _BV(DHT11_PIN);
    delayMicroseconds(40);

    DDRC &= ~_BV(DHT11_PIN);
    delayMicroseconds(40);

    dht11_in = PINC & _BV(DHT11_PIN);

    if(dht11_in){
        Serial.println("dht11 start condition 1 not met");
        return;
    }
    delayMicroseconds(80);
```

```
dht11_in = PINC & _BV(DHT11_PIN);

if(!dht11_in){
    Serial.println("dht11 start condition 2 not met");
    return;
}
delayMicroseconds(80);
// now ready for data reception
for (i=0; i<5; i++)
    dht11_dat[i] = read_dht11_dat();
DDRC |= _BV(DHT11_PIN);
PORTC |= _BV(DHT11_PIN);

byte dht11_check_sum = dht11_dat[0]+dht11_dat[1]+dht11_dat[2]+dht11_dat[3];
// check check_sum
if(dht11_dat[4]!= dht11_check_sum)
{
    Serial.println("DHT11 checksum error");
}
Serial.print("Current humidity = ");
Serial.print(dht11_dat[0], DEC);
Serial.print(".");
Serial.print(dht11_dat[1], DEC);
Serial.print("% ");
Serial.print("temperature = ");
Serial.print(dht11_dat[2], DEC);
Serial.print(".");
Serial.print(dht11_dat[3], DEC);
Serial.println("C ");

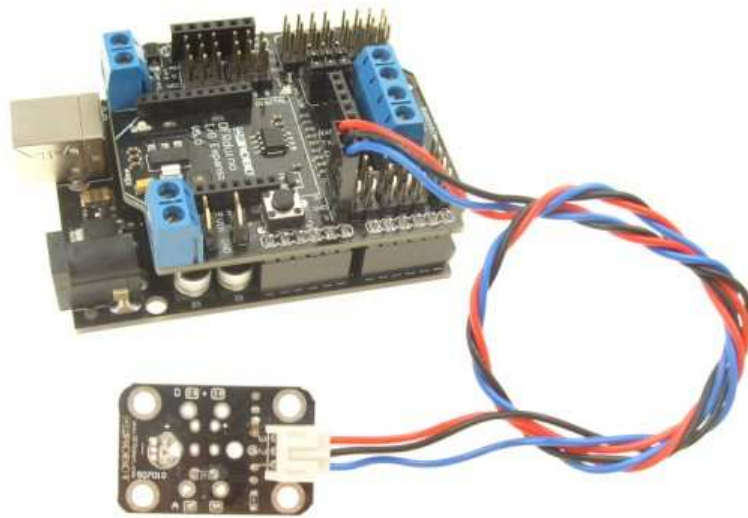
delay(2000);
}
```

编译代码后下载到 Arduino 中，打开串口助手即可看见实际测量的温度与湿度。

## 环境光线传感器模块实验

基于环保型光敏二极管的光线传感器，可以对环境光的强弱进行检测。

接线图如下：



**Arduino 实验代码：**

```
void setup()
{
  Serial.begin(9600); // 9600 bps
}
void loop()
{
  int val;
  val=analogRead(0);
  Serial.println(val,DEC);
  delay(100);
}
```

根据光线强弱的变化，读取到的值也会随其变化。打开串口助手可以看到检测结果。

### 数字人体红外热释电运动传感器模块的实验

热释电红外传感器是一种能检测人或动物身体发射的红外线而输出电信号的传感器。在这里我们把它作为机器人的一种传感器来应用，除了在我们熟知的楼道自动开关、防盗报警上得到应用外，在更多的领域应用前景看好。如果你有更妙的想法，比如：在房间无人时会自动停机的空调机、饮水机，电视机能判断无人观看或观众已经睡觉后自动关机的机构，开启监视器或自动门铃上的应用，结合摄影机或数码照相机自动记录动物或人的活动等等。

接线图如下：



**Arduino 实验代码：**

```
int ledPin = 13;           // LED connected to digital pin 13
int sensor = 2;
int val = 0;
void setup()
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
  pinMode(sensor, INPUT);  // sets the digital pin as input
}
void loop()
{
  val = digitalRead(sensor); // read the input pin
  if(HIGH == val)
    digitalWrite(ledPin, HIGH); // sets the LED to the high
  else
    digitalWrite(ledPin, LOW);  // sets the LED to the low
}
```

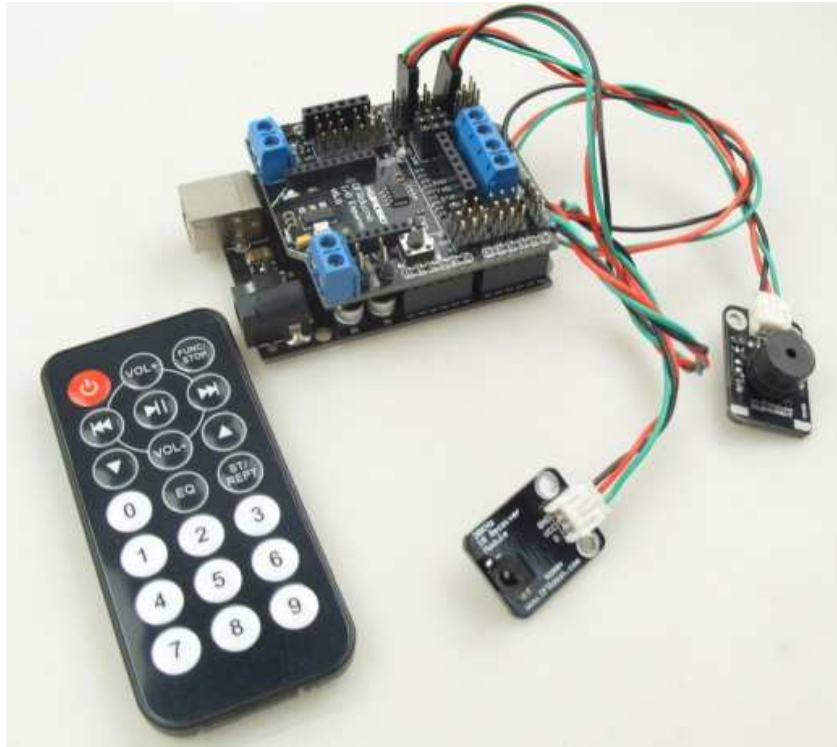
当人或动物进入传感器探测范围，传感器输出高电平，使 Arduino 上的 L 灯亮，当人或动物消失在探测范围内，传感器输出将延时 30 秒后恢复低电平，L 灯也跟着灭掉。传感器上的电位器可以调节触发延时时间。

## 红外发送接收实验

红外遥控器发出的信号是一连串的二进制脉冲码。为了使其在无线传输过程中免受其他红外信号的干扰,通常都是先将其调制在特定的载波频率上,然后再经红外发射二极管发射出去,而红外线接收装置则要滤除其他杂波,只接收该特定频

率的信号并将其还原成二进制脉冲码,也就是解调。内置接收管将红外发射管发射出来的光信号转换为微弱的电信号,此信号经由 IC 内部放大器进行放大,然后通过自动增益控制、带通滤波、解调变、波形整形后还原为遥控器发射出的原始编码,经由接收头的信号输出脚输入到电器上的编码识别电路。

电路图如下:



红外接收模块接到数字口2,蜂鸣器接到数字口6。用遥控器发送红外信号,按 VOL+蜂鸣器响,STOP 蜂鸣器停止。

**Arduino 实验代码:**

```
#define BUZZER 6//蜂鸣器
#define IR_IN 2 //红外接收

int Pulse_Width=0;//存储脉宽
int ir_code=0x00;//命令值

void timer1_init(void)//定时器初始化函数
{
    TCCR1A = 0X00;
    TCCR1B = 0X05;//给定时器时钟源
    TCCR1C = 0X00;
    TCNT1 = 0X00;
    TIMSK1 = 0X00; //禁止定时器溢出中断
}

void remote_deal(void)//执行译码结果函数
{
    switch(ir_code)
    {

```

```
case 0xff00://停止
    digitalWrite(BUZZER,LOW);//蜂鸣器不响
    break;
case 0xfe01://VOL+
    digitalWrite(BUZZER,HIGH);//蜂鸣器响
    break;
}
}
char logic_value()//判断逻辑值“0”和“1”子函数
{
    while(!(digitalRead(IR_IN))); //低等待
    Pulse_Width=TCNT1;
    TCNT1=0;
    if(Pulse_Width>=7&&Pulse_Width<=10)//低电平560us
    {
        while(digitalRead(IR_IN)); //是高就等待
        Pulse_Width=TCNT1;
        TCNT1=0;
        if(Pulse_Width>=7&&Pulse_Width<=10)//接着高电平560us
            return 0;
        else if(Pulse_Width>=25&&Pulse_Width<=27) //接着高电平1.7ms
            return 1;
    }
    return -1;
}
void pulse_deal()//接收地址码和命令码脉冲函数
{
    int i;
    //执行8个0
    for(i=0; i<8; i++)
    {
        if(logic_value() != 0) //不是0
            return;
    }
    //执行6个1
    for(i=0; i<6; i++)
    {
        if(logic_value() != 1) //不是1
            return;
    }
    //执行1个0
    if(logic_value() != 0) //不是0
        return;
```

```
//执行1个1
if(logic_value()!= 1) //不是1
    return;

//解析遥控器编码中的 command 指令
ir_code=0x00;//清零
for(i=0; i<16;i++ )
{
    if(logic_value() == 1)
    {
        ir_code |= (1<<i);
    }
}

void remote_decode(void)//译码函数
{
    TCNT1=0X00;
    while(digitalRead(IR_IN))//是高就等待
    {
        if(TCNT1>=1563) //当高电平持续时间超过100ms，表明此时没有按键按下
        {
            ir_code = 0xff00;
            return;
        }
    }

    //如果高电平持续时间不超过100ms
    TCNT1=0X00;

    while(!(digitalRead(IR_IN))); //低等待
    Pulse_Width=TCNT1;
    TCNT1=0;
    if(Pulse_Width>=140&&Pulse_Width<=141)//9ms
    {

        while(digitalRead(IR_IN));//是高就等待
        Pulse_Width=TCNT1;
        TCNT1=0;
        if(Pulse_Width>=68&&Pulse_Width<=72)//4.5ms
        {
            pulse_deal();
            return;
        }
    }
}
```



```
    }
    else if(Pulse_Width>=34&&Pulse_Width<=36)//2.25ms
    {
        while(!(digitalRead(IR_IN)));//低等待
        Pulse_Width=TCNT1;
        TCNT1=0;
        if(Pulse_Width>=7&&Pulse_Width<=10)//560us
        {
            return;
        }
    }
}

void setup()
{
    unsigned char i;
    pinMode(BUZZER,OUTPUT);//设置与蜂鸣器连接的引脚为输出模式
    pinMode(IR_IN,INPUT);//设置红外接收引脚为输入
}

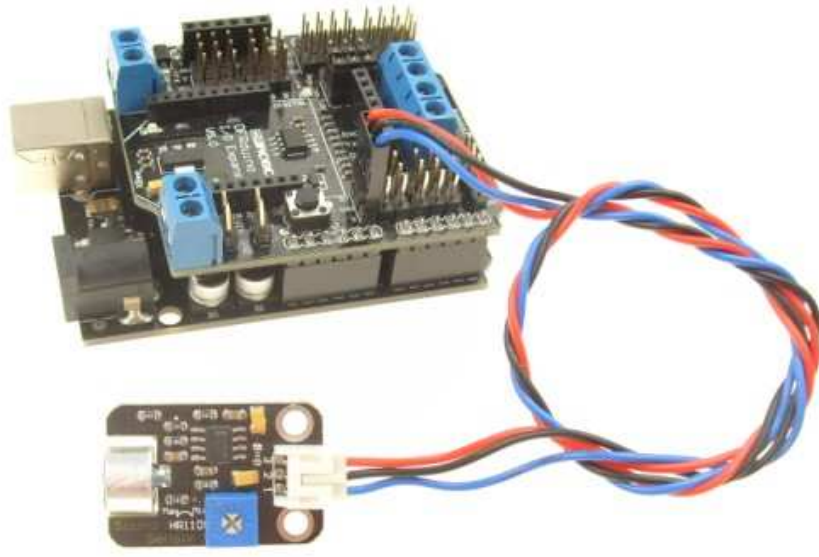
void loop()
{
    timer1_init();//定时器初始化
    while(1)
    {
        remote_decode(); //译码
        remote_deal();   //执行译码结果
    }
}
```

用红外遥控器按 VOL+按键，蜂鸣器响，按红色电源按钮，蜂鸣器停止发声。

## MIC 声音传感器模块实验

MIC 声音传感器是用来对周围环境中的声音强度进行检测，可以用来实现根据声音大小进行互动的效果。MIC（麦克风）是将声音信号转换为电信号的能量转换器件，换后的电信号还需要放大器放大才能使用，后面便是一个300dB 的放大器，放大后的模拟信号大小和声音强度成正比（AD 量化数值0-500左右）。MIC 声音传感器上有个电位器可以调整输出信号的幅度。根据板子上的指示，Min 端是调小，Max 端是调大。

接线图如下：



#### Arduino 测试代码：

```
void setup()
{
    Serial.begin(9600); // 打开串口，设置波特率为9600 bps
}
void loop()
{
    int val;
    val=analogRead(0); // MIC 声音传感器接到模拟口0
    Serial.println(val,DEC); // 从串口发送字符串并换行
    delay(100);
}
```

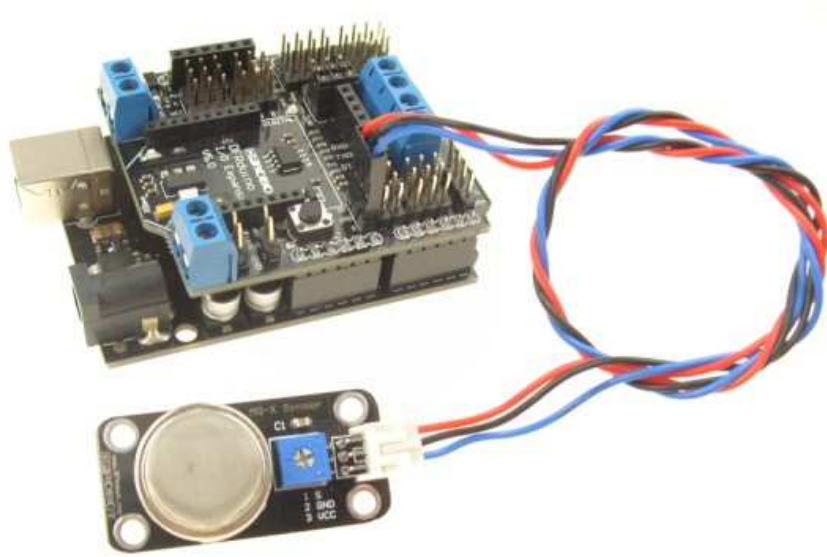
MIC 声音模块直接接到模拟口0，代码下载完毕后，打开串口助手可以看到检测到的数字值会跟随声音的强弱变化。

#### 气体传感器模块的实验

该传感器是基于 QM-NG1 探头的气体传感器，QM-NG1 是采用目前国际上工艺最成熟，生产规模最大的 SnO<sub>2</sub> 材料作为敏感基体制作的广谱性气体传感器。该产品的最大特点是对各种可燃性气体(如氢气、液化石油气、一氧化碳、烷烃类等气体)以及酒精、乙醚、汽油、烟雾等有毒气体具有高度的敏感性。

用途：用于排风扇、儿童玩具和广泛污染场所上的检验、提醒、报警功能！

接线图如下：



**Arduino 实验代码如下：**

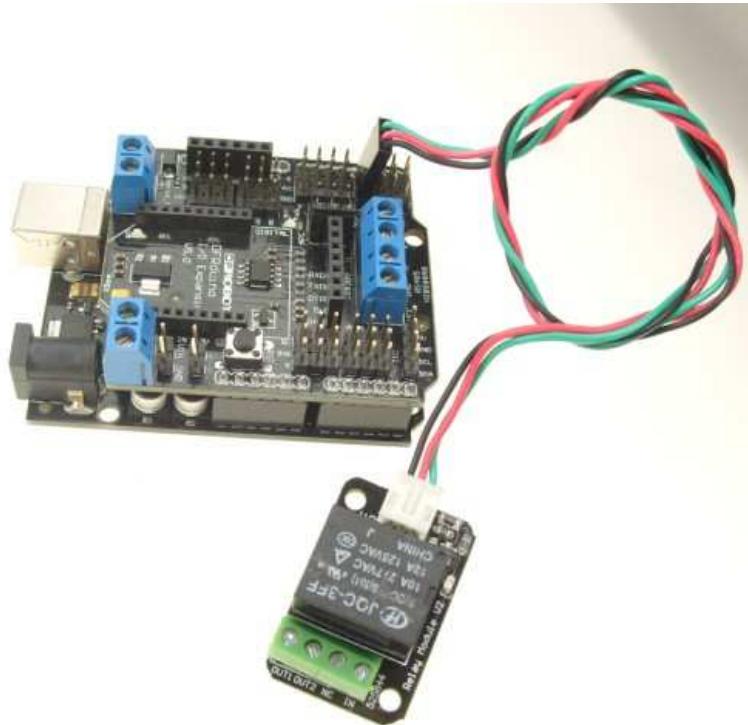
```
void setup()
{
  Serial.begin(9600); // 9600 bps
}
void loop()
{
  int val;
  val=analogRead(0);
  Serial.println(val,DEC);//
  delay(100);
}
```

代码功能是，通过模拟口0采集气体传感器的信号，然后通过串口输出到电脑上，我们可以使用串口助手看到结果。根据气体传感器内部构造，加上电后需要等待1分钟预热后才能进行测量，预热后能感受到探头有明显的温度。

## 10A 大电流继电器模块实验

在使用 Arduino 做互动项目时，很多大电流或高电压的设备通常无法直接用 Arduino 的数字 IO 口进行控制（如电磁阀、电灯、电机等），此时可以考虑用继电器的方案解决。Arduino Relay Shield 采用优质大电流继电器，提供常开与常闭接线柱，最高可以接277V/10A 的交流设备或28V/10A 的直流设备，因此能够用来控制电灯、电机等设备。

接线图如下图所示：



模块引脚定义：1脚--控制端，2脚--电源（VCC），3脚--地（GND）

输出端口：IN-输入，OUT1-输出（常开），OUT2-输出（常闭），NC-空脚

将模块连接到 UNO 控制板的任一数字口，直接由数字口给出高低电平信号来开启关闭电磁阀。

#### Arduino 测试代码：

```
int Relay = 3;

void setup()

{
    pinMode(Relay, OUTPUT); //定义 PIN3端口为输出
}

void loop()
{
    digitalWrite(Relay, HIGH); //继电器接通
    delay(2000);
    digitalWrite(Relay, LOW); //继电器断开
    delay(2000);
}
```

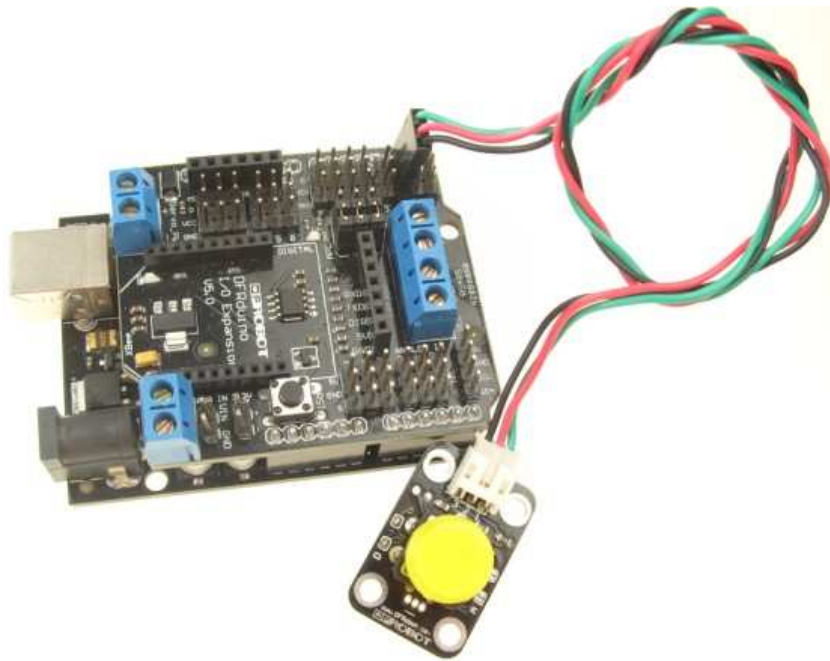
由代码可以看出，该实验中，继电器的信号端是接到数字口3的，功能是使继电器2s 循环的接通与断开，使用的时候可以明显听到继电器接通与断开时发出的声音。

#### 数字按钮模块实验

按钮我们都比较熟知了，是通过按压来实现通断的，可以用作触发事件一类的互动器材。这个实验中，我们就通过按

钮的通断来控制 UNO 板上 LED 灯得亮灭。

接线图如下：



**Arduino 实验代码：**

```
int ledPin = 13;           // choose the pin for the LED
int inputPin = 2;          // choose the input pin (for a pushbutton) 按键接到数字2
int val = 0;               // variable for reading the pin status

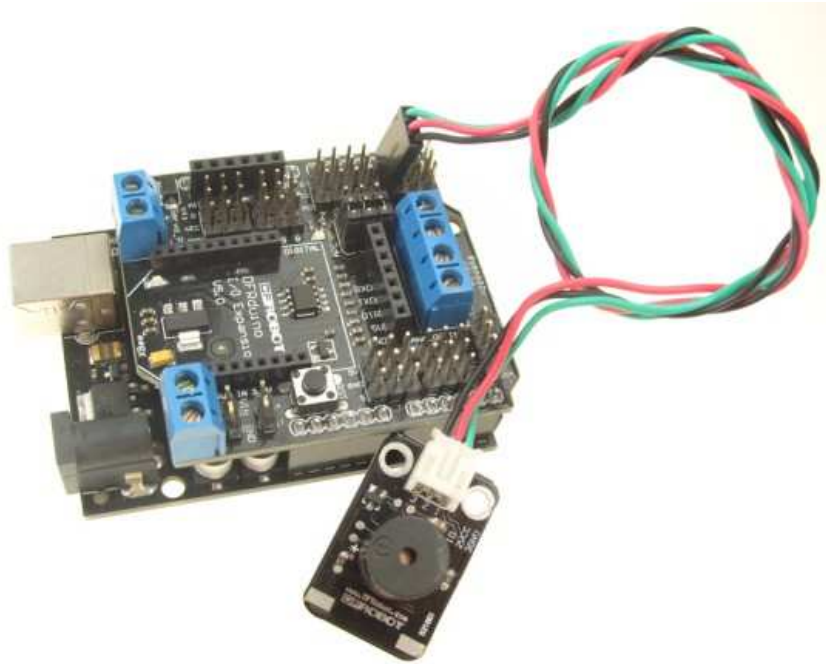
void setup() {
  pinMode(ledPin, OUTPUT); // declare LED as output
  pinMode(inputPin, INPUT); // declare pushbutton as input
}

void loop(){
  val = digitalRead(inputPin); // read input value
  if (val == HIGH) {           // check if the input is HIGH
    digitalWrite(ledPin, LOW); // turn LED OFF
  } else {
    digitalWrite(ledPin, HIGH); // turn LED ON
  }
}
```

### 数字蜂鸣器模块实验

蜂鸣器模块可以用来做声音报警或者播放 MIDI 音乐。

接线图如下：



#### Arduino 测试代码：

```
int ledPin = 3;    //蜂鸣器模块接于数字口3
```

```
void setup()
```

```
{  
  pinMode(ledPin, OUTPUT);  
}
```

```
void loop()
```

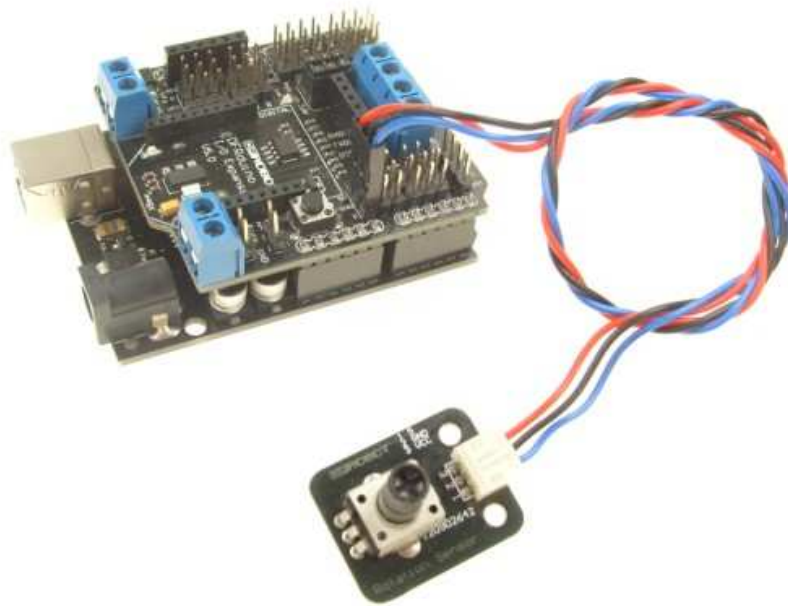
```
{  
  digitalWrite(ledPin, HIGH);  
  delay(1000);  
  digitalWrite(ledPin, LOW);  
  delay(1000);  
}
```

实验结果可以听到蜂鸣器模块间隙发出声音。

#### 模拟旋转角度传感器实验

基于电位器的旋转角度传感器，旋转角度从0到300度，使用8位 AD 可以将电压细分为255份，使用10位 AD 可以将电压细分为1024份，与 Arduino 传感器扩展板结合使用，可以精确地实现角度微小变化,可以非常容易地实现与旋转位置相关的互动效果或制作 MIDI 乐器。

连接图如下：



**Arduino 实验代码：**

```
void setup()
{
    Serial.begin(9600); // 9600 bps
}
void loop()
{
    int val;
    val=analogRead(0);
    Serial.println(val ,DEC);
    delay(100);
}
```

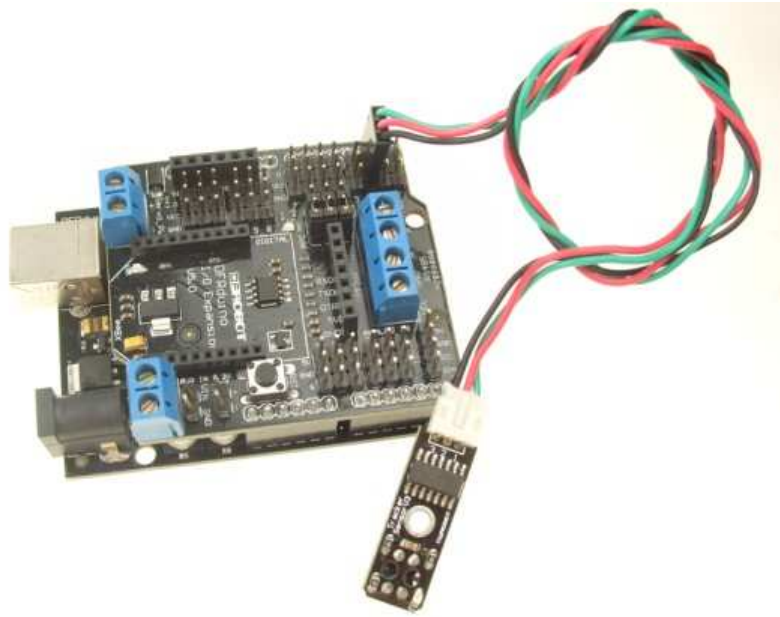
将传感器接到模拟口0，程序加载完毕后，打开串口助手观察，转动模块上的旋钮，窗口显示的值也会随之变化。

### 模拟寻线传感器实验

Mini 寻线传感器可以帮助进行白线或者黑线的跟踪，可以检测白底中的黑线，也可以检测黑底中的白线。Mini 寻线传感器红外探头输出经过门电路整形，以保证提供稳定的 TTL 输出信号，使寻线更准确更稳定。可任意选择多路组合，安装随方便是寻线机器人的必备传感器。



连接图如下：



#### Arduino 实验代码：

```
int ledPin = 13;           // choose the pin for the LED
int inputPin = 2;          //接寻线传感器
int val = 0;               // variable for reading the pin status

void setup() {
  pinMode(ledPin, OUTPUT); // declare LED as output
  pinMode(inputPin, INPUT); // declare pushbutton as input
}

void loop(){
  val = digitalRead(inputPin); // read input value
  if (val == HIGH) {           // check if the input is HIGH
    digitalWrite(ledPin, LOW); // turn LED OFF
  } else {
    digitalWrite(ledPin, HIGH); // turn LED ON
  }
}
```

黑线低电平有效，灯熄灭，白线高电平有效，灯点亮。

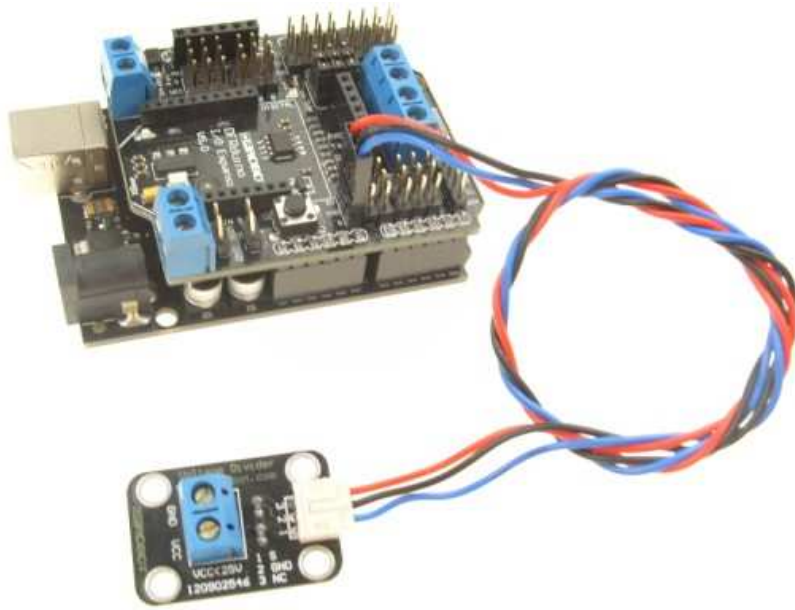
#### 电压检测模块实验

电压检测模块主要用来检测电源电压的大小。在机器人的应用中，智能机器人能知道自己的能量不多后，会主动去充电，那么就需要有电源电压检测功能。

我们设计的这款电压检测模块是一个简易的传感器，使用电阻分压的原理实现的，电压检测模块能使输入的电压缩小5倍。由于 Arduino 模拟输入电压最大为5V，故电压检测模块的输入电压不能大于 $5V \times 5 = 25V$ 。Arduino 是10位 AD，所以模拟分辨率最小为 $5V \div 1023 = 0.0049V$ ，那么电压检测模块输入最小电压为 $0.0049V \times 5 = 0.0245V$ 。



接线图如下：



```
void setup()
{
    Serial.begin(9600); // 9600 bps
}

void loop()
{
    int val;
    val=analogRead(0);
    int v = map(val, 0, 1023, 0, 25); // 0-25V
    Serial.println(v, DEC); //
    delay(100);
}
```

将被测电源是正接到模块 VCC 端，地接到 GND 端，实验结果直接通过串口助手来观测，可以看到输入端输入的电压，通过模块检测读取，在串口助手串口中显示出实际电压值。

### 火焰传感器实验

火焰传感器可以用来探测火源或其它波长在760纳米~1100纳米范围内的光源。在灭火机器人比赛中，火焰探头起着非常重要的作用，它可以用作机器人的眼睛来寻找火源或足球。利用它可以制作灭火机器人、足球机器人等。探测角度达60度，对火焰光谱特别灵敏，2个 M3安装孔，可以稳定模块不会旋转。

接线图如下：



#### Arduino 实验代码：

```
void setup()
{
  Serial.begin(9600); // 9600 bps
}
void loop()
{
  int val;
  val=analogRead(0);
  Serial.println(val,DEC);
  delay(100);
}
```

供电电压为5V 时，火焰传感器实测数据，以一根蜡烛为火源，室内打开节能灯实测：

无火源时，对着节能灯：0.3V

有火源时，

20cm: 4.8V

30cm: 4.6V

40cm: 3.9V

50cm: 2.9V

60cm: 2.5V

70cm: 2V

80cm: 1.5V

90cm: 1.2V

100cm: 1V

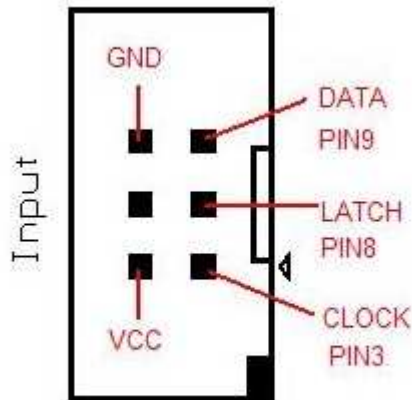
值得注意的是，火焰传感器的工作温度为-25摄氏度~85摄氏度，在使用过程中应注意火焰探头离火焰的距离不能太近，

以免造成损坏。

## Shiftout Module 模块实验

使用 SPI 通讯可将 3 个数字口扩展为 8 个数字口。当 Arduino 数字口不够用时，使用这个模块扩展是个理想的选择。并且可同时以级联多个模块，也适用于 51、AVR、PIC 等单片机

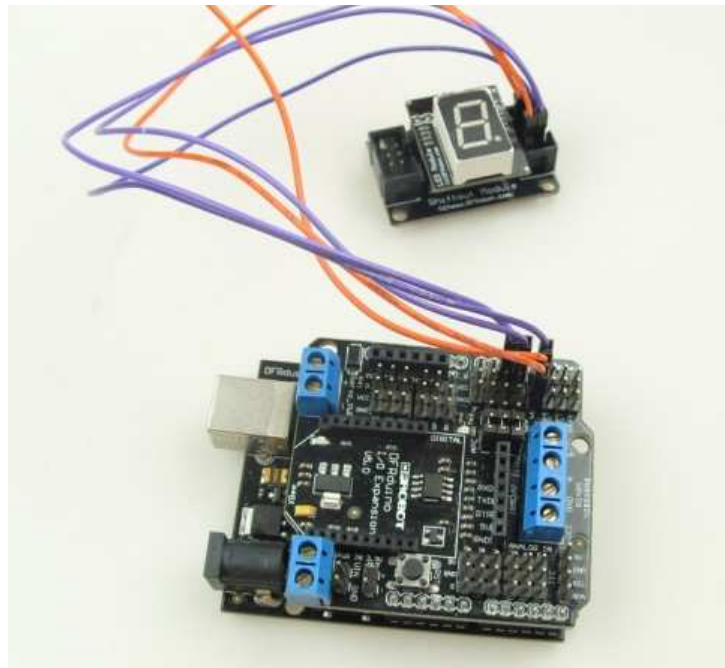
模块引脚图如下：



图中的引脚定义是配合本公司 Arduino Interface shield V1.1 多功能接口扩展板而固定的接线引脚，本套件没有配备这款扩展板，所以用户可以直接用杜邦线进行连线，端口连线也不受扩展板限制，可以将信号端口接到未被占用的任意数字端口上，而不是图中固定的接线方式。下面举2个例子说明，程序中的端口设置应与接线相对应。

### 1. 驱动数码管的实验

接线图如下：



**Arduino 实验代码:**

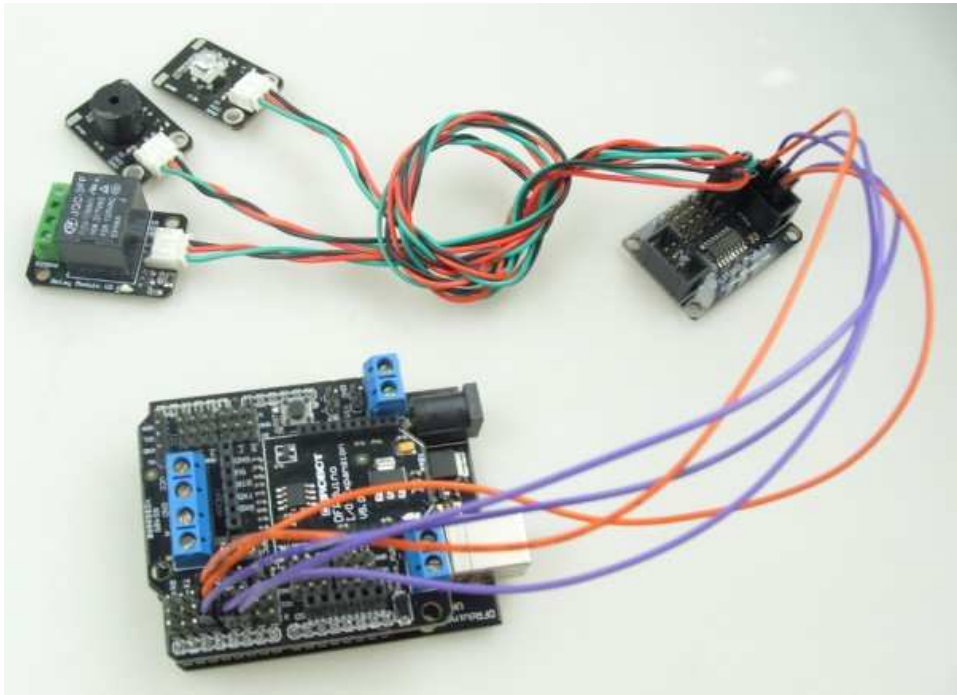
```
//Pin connected to latch pin (ST_CP) of 74HC595
const int latchPin = 4; //数字口4
//Pin connected to clock pin (SH_CP) of 74HC595
const int clockPin = 5; //数字口5
////Pin connected to Data in (DS) of 74HC595
const int dataPin = 3; //数字口3
byte Tab[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90,0xff};
void setup() {
    // 4 pins to output because they are addressed in the main loop
    pinMode(latchPin, OUTPUT);
    pinMode(dataPin, OUTPUT);
    pinMode(clockPin, OUTPUT);
    Serial.begin(9600);
    Serial.println("reset");
}
void loop() {
    if (Serial.available() > 0) {
        // ASCII '0' through '9' characters are
        // represented by the values 48 through 57.
        // so if the user types a number from 0 through 9 in ASCII,
        // you can subtract 48 to get the actual value:
        int bitToSet = Serial.read() - 48;
        // write to the shift register with the correct bit set high:
        digitalWrite(latchPin, LOW);
        // shift the bits out:
        shiftOut(dataPin, clockPin, MSBFIRST, Tab[bitToSet]);
        // turn on the output so the LEDs can light up:
        digitalWrite(latchPin, HIGH);
    }
}
```

代码下载成功后，打开串口助手，在发送区发送0-9任意数字，数码管上会显示相应数字。

**2.同时控制蜂鸣器、继电器模块和食人鱼 LED 模块的实验**

Shiftout Module 的 SPI 功能，仅用了 3 个数字口扩展出了相当于 8 个数字口的功能（仅作为输出端）。这里我们控制三个模块为例。

接线图如下：



三个模块分别接在 Shiftout Module 扩展出来的 0,1,2 三个口上。

#### Arduino 实验代码：

```
//Pin connected to latch pin (ST_CP) of 74HC595
const int latchPin = 4; //数字口 4
//Pin connected to clock pin (SH_CP) of 74HC595
const int clockPin = 5; //数字口 5
////Pin connected to Data in (DS) of 74HC595
const int dataPin = 3; //数字口 3
byte Tab[]={0x01,0x02,0x04};
void setup() {
  //t pins to output because they are addressed in the main loop
  pinMode(latchPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
}
void loop() {
  // write to the shift register with the correct bit set high:
  for(int i=0;i<3;i++)
  {
    digitalWrite(latchPin, LOW);
    // shift the bits out:
    shiftOut(dataPin, clockPin, MSBFIRST, Tab[i]);
    // turn on the output so the LEDs can light up:
    digitalWrite(latchPin, HIGH);
```

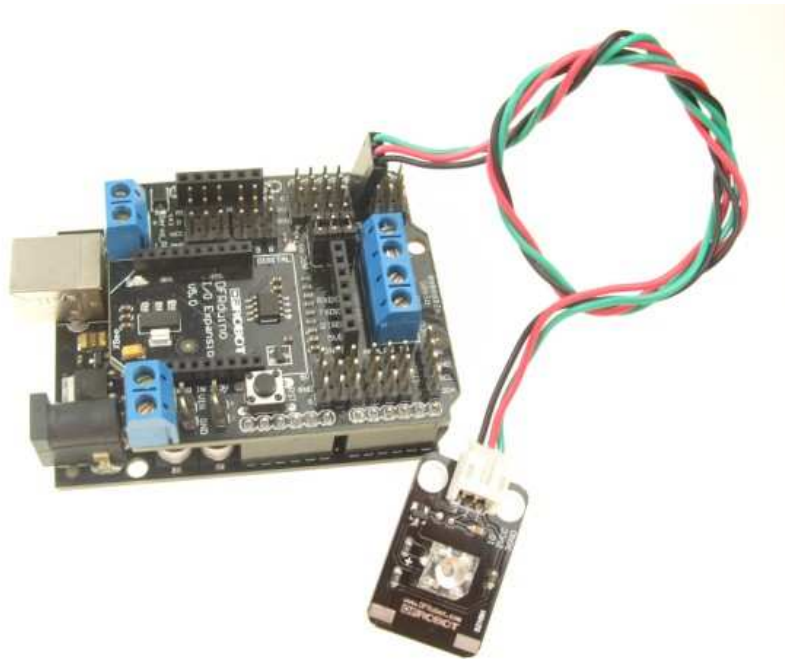
```
    delay(1000);  
  }  
}
```

LED 灯，继电器模块和蜂鸣器各自依次响应 1s，响应时间可以修改，程序中可以看出，这里我是把数字口 3,4,5 作为 SPI 接口的。

### 数字食人鱼红色 LED 发光模块实验

LED 模块主要用来照明和作为指示，可以用来做一些声光互动。食人鱼 LED 使用三极管驱动，亮度更亮。

接线图如下：



#### Arduino 实验代码：

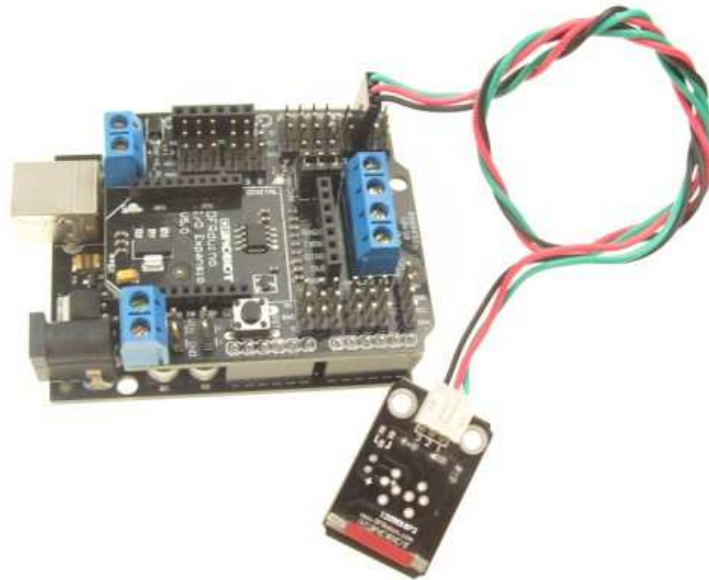
```
int ledPin = 3;    // LED connected to digital pin 3  
void setup()  
{  
  pinMode(ledPin, OUTPUT);  
}  
void loop()  
{  
  digitalWrite(ledPin, HIGH);  
  delay(1000);  
  digitalWrite(ledPin, LOW);  
  delay(1000);  
}
```

实验结果可看到 LED 不停的闪烁。

## 数字贴片磁感应传感器实验

基于感磁材料的磁感应传感器，可以用来对磁性材料（磁铁）的探测，探测范围可达3cm 左右(探测范围和磁性强弱有关)，可以制作与磁性材料（磁铁）相关的互动作品。

接线图如下：



**Arduino 实现代码：**

```
int ledPin = 13;           // choose the pin for the LED
int inputPin = 2;          // choose the input pin 磁感应接到数字2
int val = 0;               // variable for reading the pin status

void setup() {
  pinMode(ledPin, OUTPUT); // declare LED as output
  pinMode(inputPin, INPUT); // declare pushbutton as input
}

void loop(){
  val = digitalRead(inputPin); // read input value
  if (val == HIGH) {           // check if the input is HIGH
    digitalWrite(ledPin, LOW); // turn LED OFF
  } else {
    digitalWrite(ledPin, HIGH); // turn LED ON
  }
}
```

实验结果，磁铁靠近 L 灯亮，磁铁远离 L 灯灭。



## 震动传感器模块实验

这是一款基于弹簧摆动原理设计的震动传感器，震动势能传到弹簧末端引起共振，左右摆动碰触到金属外壁形成通路。由于导通时间和弹簧的刚性有关，所以该震动传感器的输出带宽很窄，在使用 Arduino 读取时，建议使用外部中断功能。

接线图如下：



### Arduino 外部中断代码：

//接口 D2： 振动，水银，干簧管 数字输入传感器。

```
#define SensorLED      13
#define SensorINPUT    2
unsigned char state = 0;
void setup()
{
    pinMode(SensorLED, OUTPUT);
    pinMode(SensorINPUT, INPUT);
    attachInterrupt(0, blink, FALLING); //D2口为外部中断0，当有下降沿触发的时候调用 blink 函数
}
void loop()
{
    if(state!=0)
    {
        state = 0;
        digitalWrite(SensorLED,HIGH);
        delay(500);
    }
    else
```



```
digitalWrite(SensorLED,LOW);  
}  
  
void blink()//数字输入传感器下降沿触发中断服务函数  
{  
    state++;  
}
```

最后结果可观察数字口13脚上的 L 指示灯，震动后可以看见 L 灯亮500ms。

## 综合实验篇

### 基于 MIC 声音传感器模块和食人鱼 LED 模块的声光互动实验

#### 实验原理：

采用一个 MIC 声音传感器，采集音箱的音频信号。然后转换为 PWM 输出控制 1 个 LED 模块，食人鱼 LED 模块的亮度就表示左右声道音频信号的幅度大小（如果有两个 MIC 声音模块和食人鱼 LED 就可以分别采集左右声道音响的信号）。

将 MIC 声音模块接到模拟口 0，LED 接到任一 PWM 输出口，这里选择数字口 5 作 PWM 输出。

接线图如下：



**Arduino 实验代码:**

```
int LED = 5;
void setup()
{
    pinMode(LED, OUTPUT);
}
void loop()
{
    int val;
    val=analogRead(0);

    val = map(val, 30, 0, 0, 255);//声音采集幅度 0-30 转换为 PWM 输出 255-0
    analogWrite(LED,val);
    delay(50);
}
```

代码下载完毕后,用音箱播放您喜爱的歌曲,将 MIC 声音模块的麦克风对着音箱,就可以看到 LED 灯随着音乐的节奏一闪一闪的了。

**红外遥控器控制 Shiftout LED Kiti 显示数字**

实验原理:红外遥控器上有 0-9 十个数字键,利用红外遥控器发出红外信号,红外接收模块将对应的键值转换成数码管显示数字。

**Arduino 实验代码:**

```
#define IR_IN 5 //红外接收

int Pulse_Width=0;//存储脉宽
int ir_code=0x00;//命令值
int bitToSet ;
//Pin connected to latch pin (ST_CP) of 74HC595
int latchPin = 8;
//Pin connected to clock pin (SH_CP) of 74HC595
int clockPin = 3;
////Pin connected to Data in (DS) of 74HC595
int dataPin = 9;
byte Tab[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90,0xff};
void timer1_init(void)//定时器初始化函数
{
    TCCR1A = 0X00;
    TCCR1B = 0X05;//给定时器时钟源
    TCCR1C = 0X00;
    TCNT1 = 0X00;
    TIMSK1 = 0X00; //禁止定时器溢出中断
}
```

void remote\_deal(void)//执行译码结果函数

```
{
  switch(ir_code)
  {
    case 0xff00://停止
      // bitToSet=10;
      break;
    case 0xf30c://显示 0
      bitToSet=0;
      break;
    case 0xef10://显示 1
      bitToSet=1;
      break;
    case 0xee11://显示 2
      bitToSet=2;
      break;
    case 0xed12://显示 3
      bitToSet=3;
      break;
    case 0xeb14://显示 4
      bitToSet=4;
      break;
    case 0xea15://显示 5
      bitToSet=5;
      break;
    case 0xe916://显示 6
      bitToSet=6;
      break;
    case 0xe718://显示 7
      bitToSet=7;
      break;
    case 0xe619://显示 8
      bitToSet=8;
      break;
    case 0xe51a://显示 9
      bitToSet=9;
      break;
  }
}
```

char logic\_value()//判断逻辑值“0”和“1”子函数

```
{
  while(!(digitalRead(5))); //低等待
  Pulse_Width=TCNT1;
```

```
TCNT1=0;
if(Pulse_Width>=7&&Pulse_Width<=10)//低电平 560us
{
    while(digitalRead(5));//是高就等待
    Pulse_Width=TCNT1;
    TCNT1=0;
    if(Pulse_Width>=7&&Pulse_Width<=10)//接着高电平 560us
        return 0;
    else if(Pulse_Width>=25&&Pulse_Width<=27) //接着高电平 1.7ms
        return 1;
}
return -1;
}
void pulse_deal()//接收地址码和命令码脉冲函数
{
    int i;
    //执行 8 个 0
    for(i=0; i<8; i++)
    {
        if(logic_value() != 0) //不是 0
            return;
    }
    //执行 6 个 1
    for(i=0; i<6; i++)
    {
        if(logic_value() != 1) //不是 1
            return;
    }
    //执行 1 个 0
    if(logic_value() != 0) //不是 0
        return;
    //执行 1 个 1
    if(logic_value() != 1) //不是 1
        return;

    //解析遥控器编码中的 command 指令
    ir_code=0x00;//清零
    for(i=0; i<16;i++ )
    {
        if(logic_value() == 1)
        {
            ir_code |= (1<<i);
        }
    }
}
```

```
    }  
  }  
}  
void remote_decode(void)//译码函数  
{  
  TCNT1=0X00;  
  while(digitalRead(5))//是高就等待  
  {  
    if(TCNT1>=1563) //当高电平持续时间超过 100ms，表明此时没有按键按下  
    {  
      ir_code = 0xff00;  
      return;  
    }  
  }  
  
  //如果高电平持续时间不超过 100ms  
  TCNT1=0X00;  
  
  while(!(digitalRead(5))); //低等待  
  Pulse_Width=TCNT1;  
  TCNT1=0;  
  if(Pulse_Width>=140&&Pulse_Width<=141)//9ms  
  {  
  
    while(digitalRead(5));//是高就等待  
    Pulse_Width=TCNT1;  
    TCNT1=0;  
    if(Pulse_Width>=68&&Pulse_Width<=72)//4.5ms  
    {  
      pulse_deal();  
      return;  
    }  
    else if(Pulse_Width>=34&&Pulse_Width<=36)//2.25ms  
    {  
      while(!(digitalRead(5)));//低等待  
      Pulse_Width=TCNT1;  
      TCNT1=0;  
      if(Pulse_Width>=7&&Pulse_Width<=10)//560us  
      {  
        return;  
      }  
    }  
  }  
}
```

```
}  
void setup()  
{  
    unsigned char i;  
    pinMode(IR_IN, INPUT); //设置红外接收引脚为输入  
    pinMode(latchPin, OUTPUT);  
    pinMode(dataPin, OUTPUT);  
    pinMode(clockPin, OUTPUT);  
}  
void loop()  
{  
    timer1_init(); //定时器初始化  
    while(1)  
    {  
        remote_decode(); //译码  
        remote_deal(); //执行译码结果  
        digitalWrite(latchPin, LOW);  
        shiftOut(dataPin, clockPin, MSBFIRST, Tab[bitToSet]);  
        digitalWrite(latchPin, HIGH);  
    }  
}
```

从程序中可以看出，红外接收模块是接到数字口 5 的，Shiftout LED Kit 模块的接线参照 Shiftout LED Kit 实验处的模块接线图。这里依然是将数据端口 `dataPin` 接到数字口 9，时钟信号口 `clockPin` 接到数字口 3，使能口 `latchPin` 接到数字口 8。用户可以自行修改接线设置，修改后程序中要做相应的修改。

实验结果如下图：



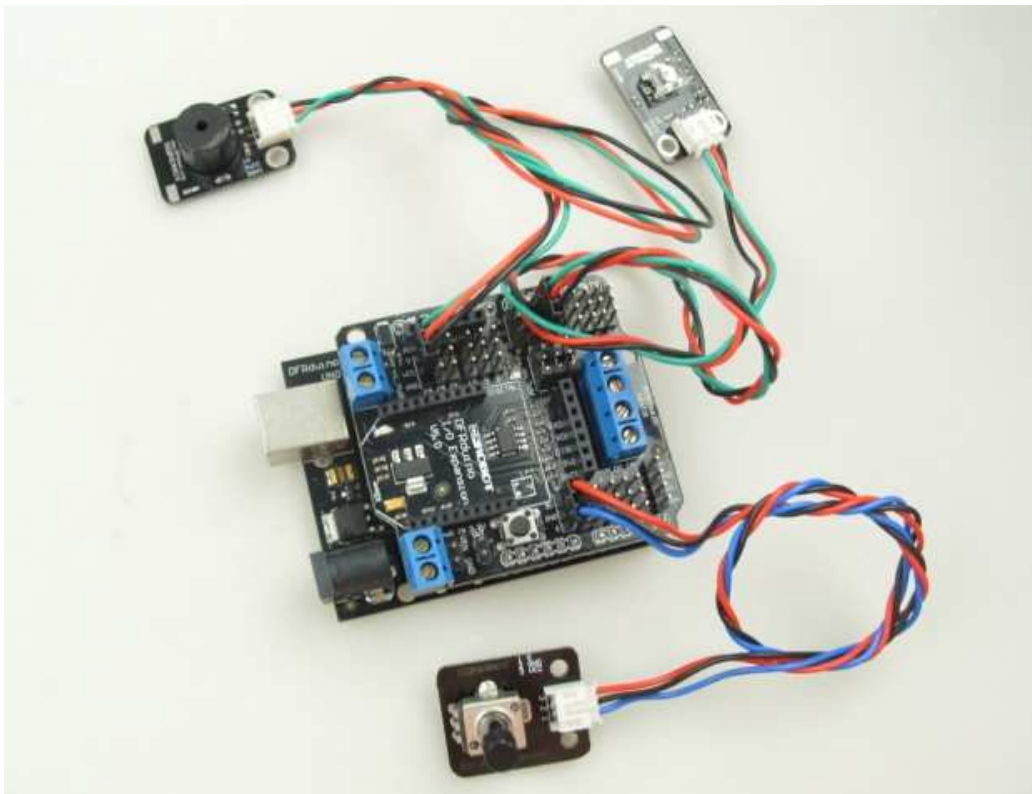
用红外发送遥控器对着红外接收模块按键，输入 0-9 任一数字，数码管上会显示相应键值并保持，直到下一次按键有效方才替换为下次按键键值。

## 电位器角度传感器控制蜂鸣器音调与 LED 灯亮度实验

### 实验原理：

电位器角度传感器的旋转角度从0到300度，使用10位 AD 可以将电压细分为1024份，可以精确地实现角度微小变化，利用旋转位置的值不同控制 LED 的 PWM 输出的值与蜂鸣器音调的变化。

接线图如下：



电位器角度传感器直接接在模拟口0，蜂鸣器接在数字口13，食人鱼 LED 灯接在具有 PWM 输出的端口，这里选择数字口5，用户也可以根据自己意愿修改端口，要注意的是，程序中用到的端口要与接线选择的端口一致。

### Arduino 实验代码：

```
#include <Tone.h>
Tone tone1;
//全局变量定义
#define led 5
int i=0;
int distance=0;
int data[4];
int val=0;
/*****/
void setup()
```

```
{
  tone1.begin(13); // 定义喇叭接线引脚
  // tone1.play(NOTE_A4);
  pinMode(led,OUTPUT);
}
/*****/
void loop()
{
  int val=analogRead(0);
  int temp=map(val,0,1023,0,255);
  analogWrite(led,temp);
  if(val>5&&val<1020)
  {
    tone1.play(val*5);
  }
  else
  {
    pinMode(13,INPUT);
  }
  delay(10);
}
```

程序下好后，试着旋转电位器角度模块上的旋钮，就可以看到，灯的亮度和蜂鸣器播放的声音的音调也在随之改变。

以上程序都进过验证。

版权归北京龙凡汇众机器人科技有限公司所有

版本号	日期	备注
1.0	2011-12-28	创建文档
1.1	2012-6-29	修改软件介绍部分