# 第一章 Arduino简介

## 1.1 Arduino套件简介

### 1 Arduino介绍

Arduino是一块简单易学的电子创意设计平台，它让你可以快速做出有趣的东西。Arduino可以配合一些电子元件使用，例如LED灯、蜂鸣器、按键、光敏电阻等等。Arduino配套软件基于“开放”原则，可以让您免费下载使用，开发出更多令人惊奇的互动作品。

### 2 设备参数及组成

官方说明：

采用Atmel微处理控制器。Arduino大小尺寸：宽70mm X高54mm。

**数字输入/输出端共0~13，注意0和1（RX TX）用于和电脑通信，不要占用。**3，5，6，9，10，11口可输出8bit (0-255)PWM信号

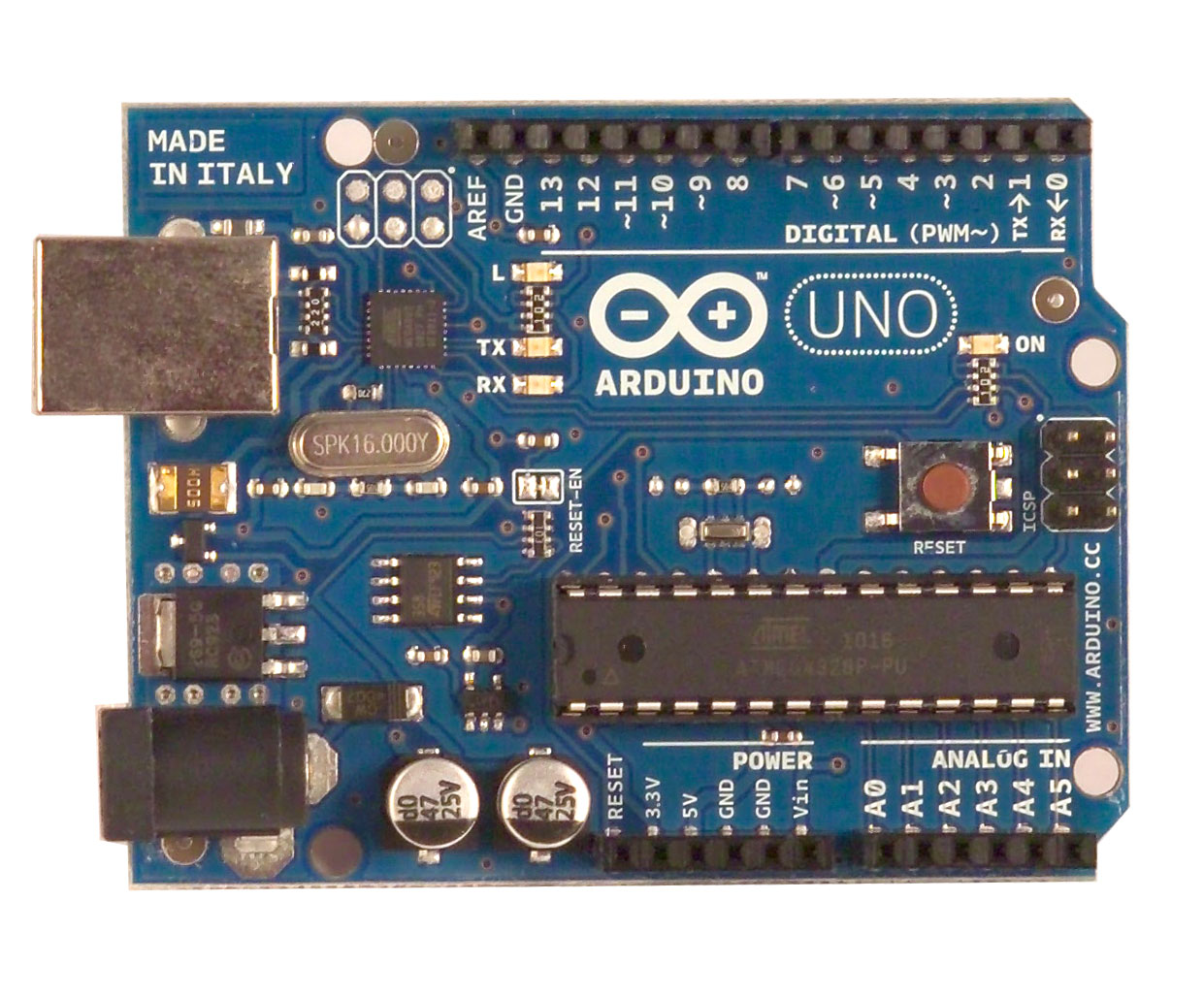
**模拟输入端共0~5。可以检测0-5V信号，10bit量化（0-1023）**

输入电压：

（1）USB供电。左侧为USB接口和电源接口。我们在使用时，将USB线分别接在Arduino和电脑USB口上。供电电压5V。

（2） 外部7V~12V直流电压输入。

输出电压：5V直流电压输出和 3.3V直流电压输出

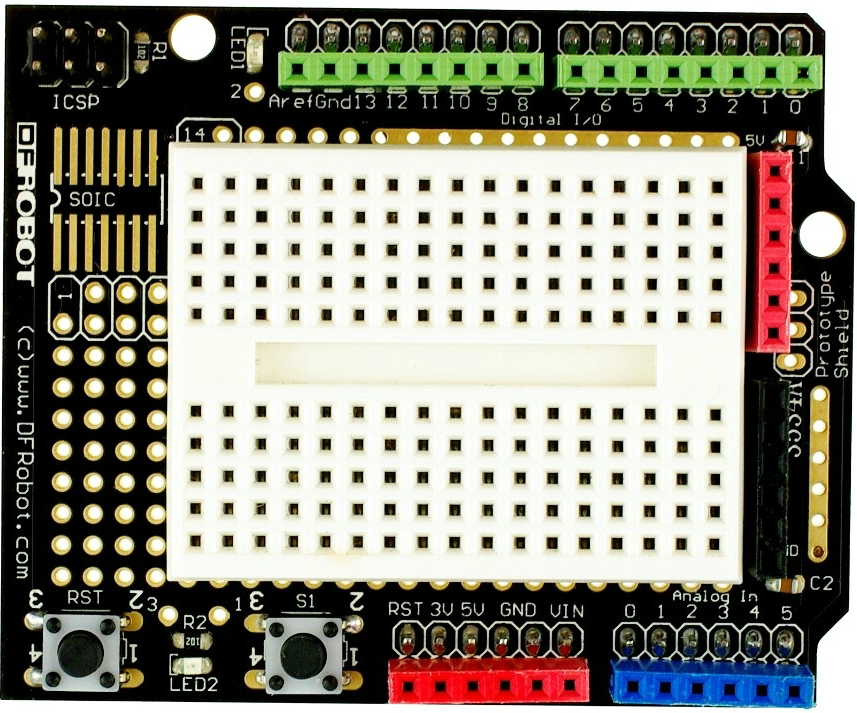


0-13数字输入/输出接口

A0-A5模拟输入接口

USB接口

电源接口



0-13数字输入/输出接口

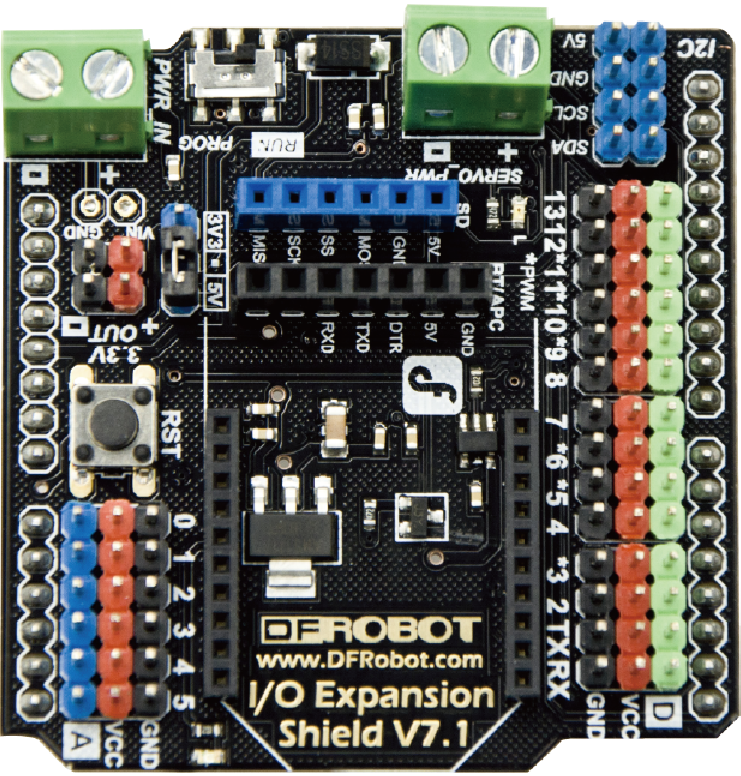
红正（5V，VCC），黑负（GND，0V，地）

A为模拟输入口，一般接传感器

D为数字输入输出口，一般接被控对象，也可接开关型传感器

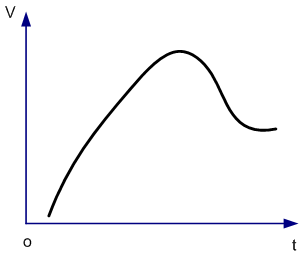
A0-A5模拟输入接口

0-13数字输入/输出接口



小知识

“数字”信号只有2种状态：高电平（电源正极电压，这里为5V）低电平（电源负极电压，这里为0V）。而“模拟”信号是“连续”的，在Arduino上可以是0-5V中任何数值。



模拟信号

数字信号

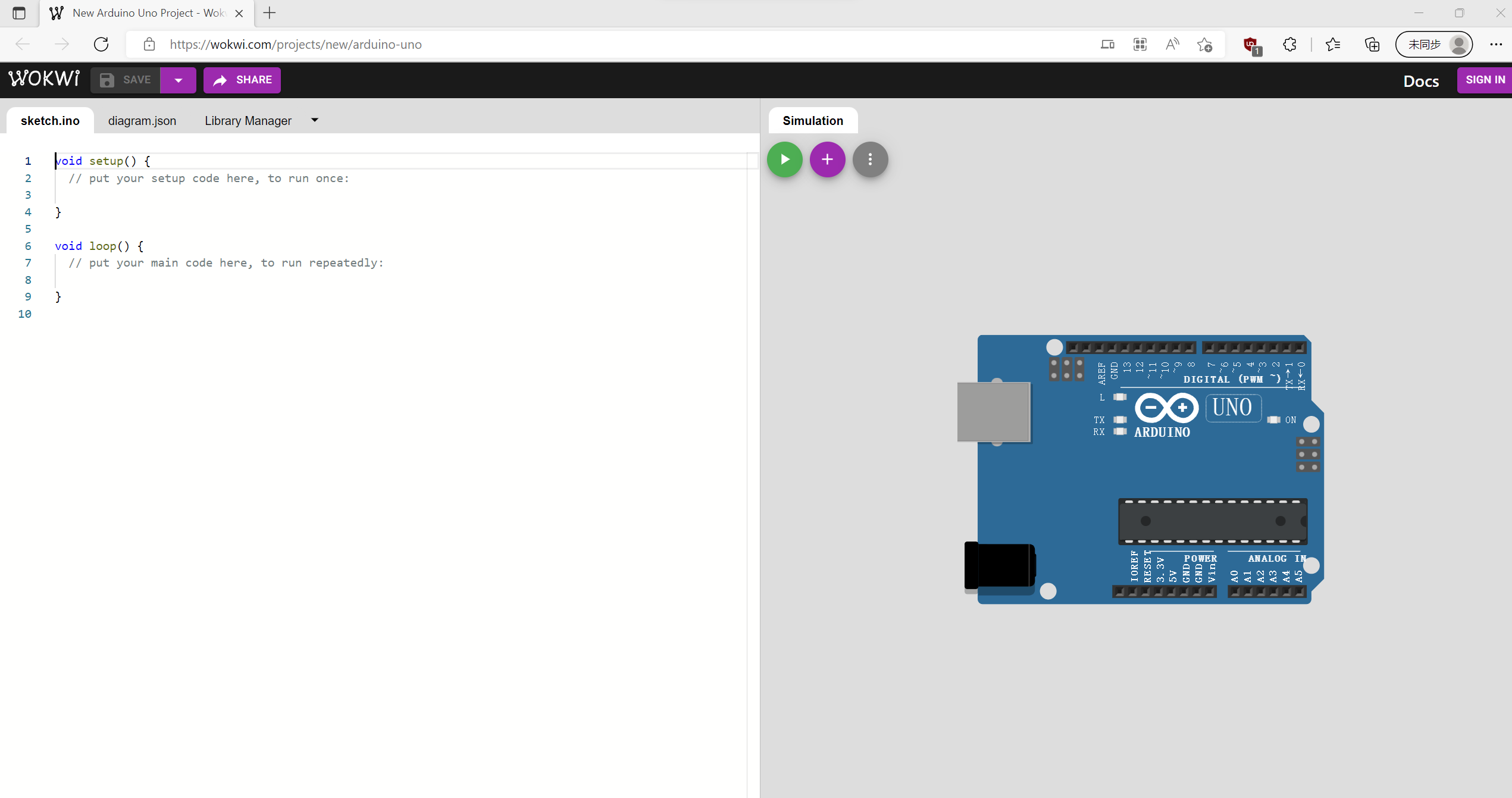
在Arduino上，Digital（数字）端口常用于“输出”。它只能输出数字信号，即0V或5V。因此我们可以把它看作是一个特殊的电源，能够根据我们的需要输出0V或5V电压。

Analog In用于“输入”，也就是检测模拟信号。这里我们可以把它看作是一个量程为5V的电压表。

## 1.2 准备工作

打开 <https://wokwi.com/projects/new/arduino-uno>

该网站包含一个虚拟Arduino开发板和常用零件，硬件条件不足时可用来学习。注意：国外网站打开较慢，请耐心等待。



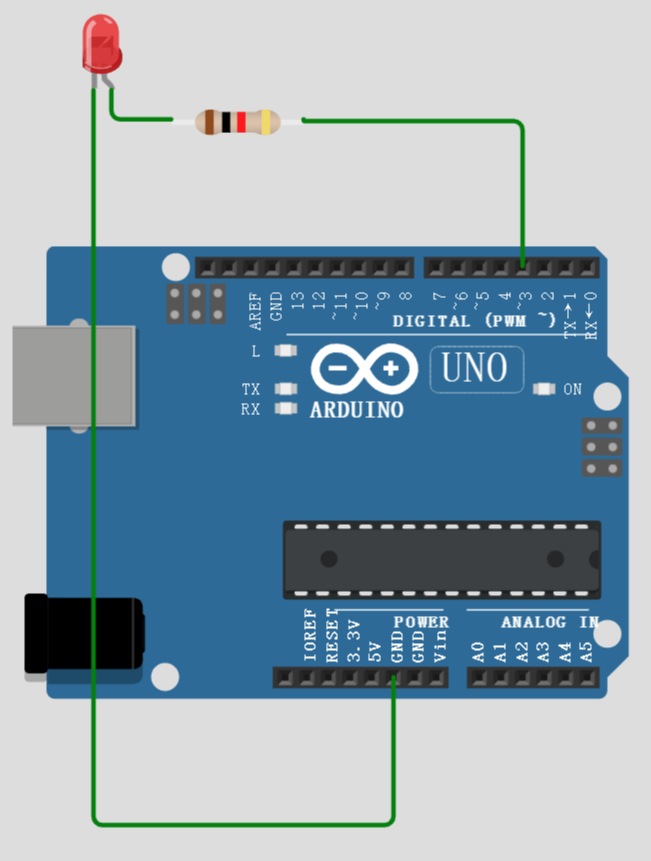
# 第二章 Arduino数字输出

本章主要学习如何用Arduino控制一个“被控对象”的开关，即输出数字量。

## 2.1 多彩led灯实验

### 1 第一个程序：闪烁led灯

#### 元件清单及电路连接



操作过程视频：



或者： https://m.shsbnu.net/mod/resource/view.php?id=9623

#### 程序代码（将以下文字复制到网页中）

int ledPin=3; //设定控制LED的数字IO脚

void setup()

{

pinMode(ledPin,OUTPUT);//设定数字IO口的模式，OUTPUT 为输出

}

void loop()

{

digitalWrite(ledPin,HIGH); //设定PIN3脚为HIGH = 5V左右

delay(1000); //设定延时时间，1000 = 1秒

digitalWrite(ledPin,LOW); //设定PIN8脚为LOW = 0V

delay(1000); //设定延时时间，1000 = 1秒

}

#### 实验结果

LED灯亮1秒，灭1秒不断闪烁。

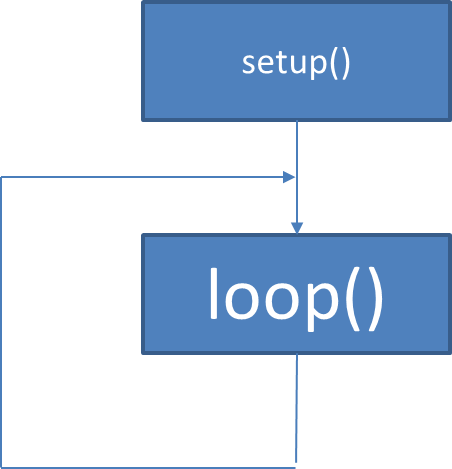
#### 工作原理分析

（1）程序部分

Arduino上的C++程序包含setup()初始化和loop()循环执行2段程序。先执行setup()的内容，完毕后反复执行loop()。**每个程序有且只有1个setup()和1个loop()函数。**

**setup()是用来“初始化”变量，管脚模式，调用库函数等等，此函数只运行一次。**本程序在setup()中用数字IO口输入输出模式定义函数pinMode（pin，mode），将数字的第3引脚设置为输出模式。

**loop()包含反复执行的内容，编写一个周期即可循环执行。**本程序在loop()中先用 数字IO口输出电平定义函数digitalWrite(pin, value)，将数字3口定义为高电平5V；接着调用延时函数delay(ms)（单位ms）延时1000ms； 再用数字IO口输出电平定义函数digitalWrite(pin, value)，将数字3口定义为低电平；接着再调用延时函数delay(ms) 单位ms）延时1000ms因为loop()函数是一个循环函数，所以这个过程会不断的循环，使3口电压形成一个5V方波。



常用语句示例：（注意大小写）

Delay(2000); 延时2000毫秒

pinMode(3,OUTPUT); 数字3口设为输出，常写在setup()里

digitalWrite(3,HIGH); 3口输出5V电压

digitalWrite(7,LOW); 7口输出0V电压

注意：用digitalWrite需在初始化中用pinMode设为输出

（2）电路部分

Led灯有两种连线方法：当led灯的阳极通过限流电阻与板子上的数字I/O口相连，数字口输出高电平时，led导通，发光二极管发出亮光；数字口输出低电平时，led截止，发光二极管熄灭。如下图：

VCC=5V（电源正极），GND=0V（电源负极）



当led灯的阴极与板子上的数字I/O口相连时，数字口输出高电平，led截止，发光二极管熄灭；数字口输出低电平，led灯导通，发光二极管点亮。



在这个电路中，数字3引脚为高电平5V时led灯熄灭，然后延时1s，数字3引脚为低电平0V led灯点亮，再延时1s。这样使led灯亮1s、灭1s，在视觉上就形成闪烁状态。如果想让led快速闪烁，可以将延时时间设置的小一些，如果想让led慢一点闪烁，可以将延时时间设置的大一些。

**注意：我们一般不使用Digital 0，1口，否则会无法写入程序**。

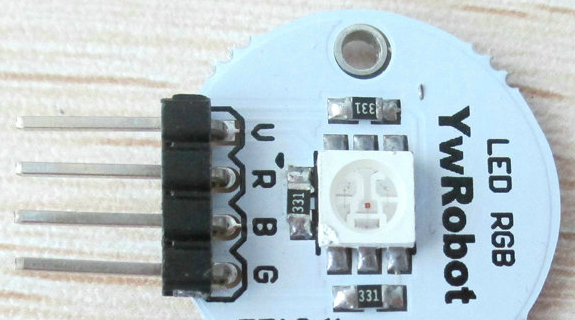
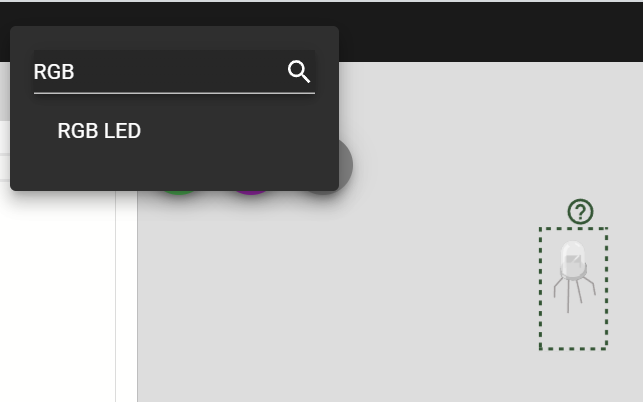
直插式LED的工作电压和颜色、亮度相关，约1.8-3.3V，工作电流5-20mA。和5V Arduino配套使用时需要增加限流电阻，常用220欧。

### 2 自主设计：交通灯控制系统

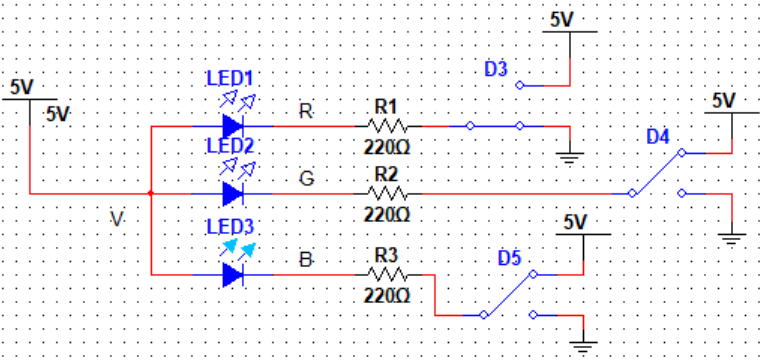
循环亮灯：绿灯亮持续8秒，然后绿灯灭，黄灯亮持续2秒，然后黄灯灭红灯亮持续8秒，再回到开始（绿灯亮）。

提示：

1 器材选用 RGB全彩LED。用模拟器时可以省略电阻（连接真实电路时必须有！）

实物 软件中的名称、外形



电路原理（蓝框内的部分）及应用电路图

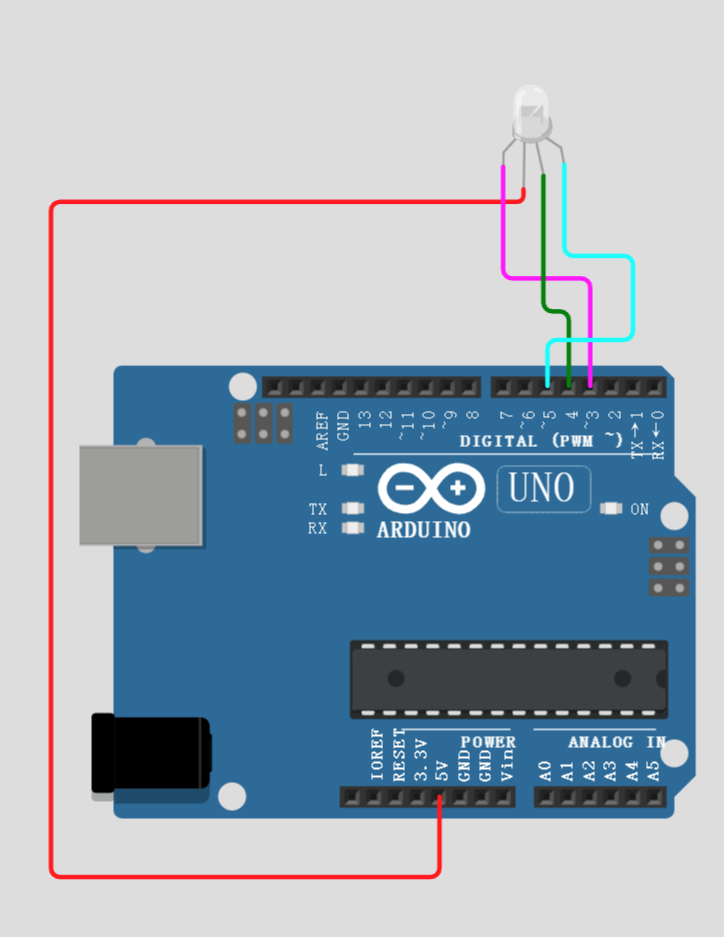
从原理图可知，**三个LED阳极是接在一起的（V脚），接电源正（5V）。**R，G，B脚的电压决定了对应LED是否点亮**。把RGB连接到Digital口，0V低电平时点亮。**

程序注意大小写。R,G,B分别接3个数字口（如3，4，5）。

程序框架不能动，有且只有1个setup()和1个loop()。

setup()里用3个pinMode（X,OUTPUT）初始化端口。

用digitalWrite控制亮灭，delay控制时间。3个digitalWrite配合使用可以控制RGB三路LED。红绿同时亮可以合成黄色。



参考操作视频

或者：https://m.shsbnu.net/mod/resource/view.php?id=9624

思考:

1. 如何编写: 变灯前闪烁2次。
2. 参照 “蜂鸣器”中的内容，在变灯时发声提示，发声时间1秒。

**请回答** [**https://ks.wjx.top/vj/YDHNQgD.aspx**](https://ks.wjx.top/vj/YDHNQgD.aspx) **中的问题，作为本节课出勤依据。**

## 2.2 其他可用“数字输出”控制的元件

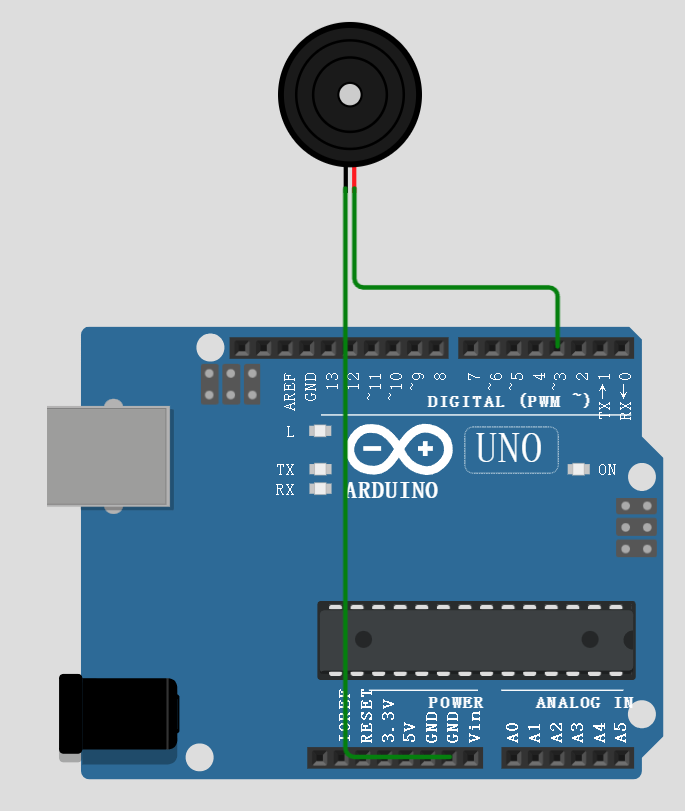
### 2.2.1 蜂鸣器

蜂鸣器是一种一体化结构的电子讯响器，采用直流电压供电，广泛应用于计算机、打印机、复印机、报警器、电子玩具、汽车电子设备、电话机、定时器等电子产品中作发声器件。

#### 连线

蜂鸣器发声原理是电流通过电磁线圈，使电磁线圈产生磁场来驱动振动膜发声的，因此需要一定的电流才能驱动它，本实验用的蜂鸣器内部带有驱动电路，所以可以直接使用。当与蜂鸣器连接的引脚为高电平时，内部驱动电路导通，蜂鸣器发出声音；当与蜂鸣器连接的引脚为低电平，内部驱动电路截止，蜂鸣器不发出声音。

**将蜂鸣器的正极（长腿）连接到数字口，蜂鸣器的负极（短腿）连接到GND插口中**。如下图，当数字口设为高电平时，蜂鸣器应该发声。





小知识



蜂鸣器按其驱动方式的不同，可分为：有源蜂鸣器（内含驱动线路）和无源蜂鸣器（外部驱动） 教你区分有源蜂鸣器和无源蜂鸣器,现在市场上出售的一种小型蜂鸣器因其体积小(直径只有11mm)、重量轻、价格低、结构牢靠，而广泛地应用在各种需要发声的电器设备、电子制作和单片机等电路中。有源蜂鸣器和无源蜂鸣器的外观如图a、b所示。a)有源 b) 无源。



从图a、b外观上看，两种蜂鸣器好像一样，但仔细看，两者的高度略有区别，有源蜂鸣器a，高度为9mm，而无源蜂鸣器b的高度为8mm。如将两种蜂鸣器的引脚郡朝上放置时，可以看出有绿色电路板的一种是无源蜂鸣器，没有电路板而用黑胶封闭的一种是有源蜂鸣器。进一步判断有源蜂鸣器和无源蜂鸣器，还可以用万用表电阻档Rxl档测试:用黑表笔接蜂鸣器 "+"引脚，红表笔在另一引脚上来回碰触，如果触发出咔、咔声的且电阻只有8Ω(或16Ω)的是无源蜂鸣器;如果能发出持续声音的，且电阻在几百欧以上的，是有源蜂鸣器。 有源蜂鸣器直接接上额定电源(新的蜂鸣器在标签上都有注明)就可连续发声;而无源蜂鸣器则和电磁扬声器一样，需要接在音频输出电路中才能发声。

按构造方式的不同，可分为：电磁式蜂鸣器和压电式蜂鸣器；

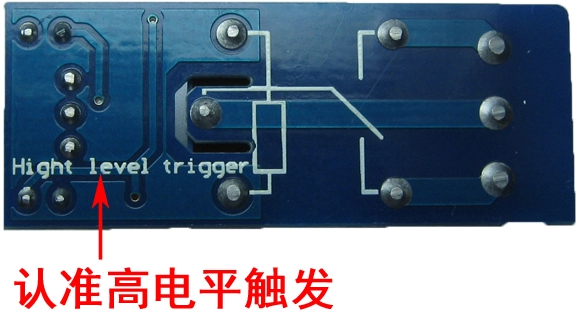
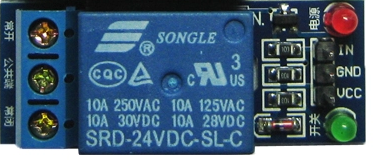
压电式蜂鸣器 压电式蜂鸣器主要由多谐振荡器、压电蜂鸣片、阻抗匹配器及共鸣箱、外壳等组成。有的压电式蜂鸣器外壳上还装有发光二极管。

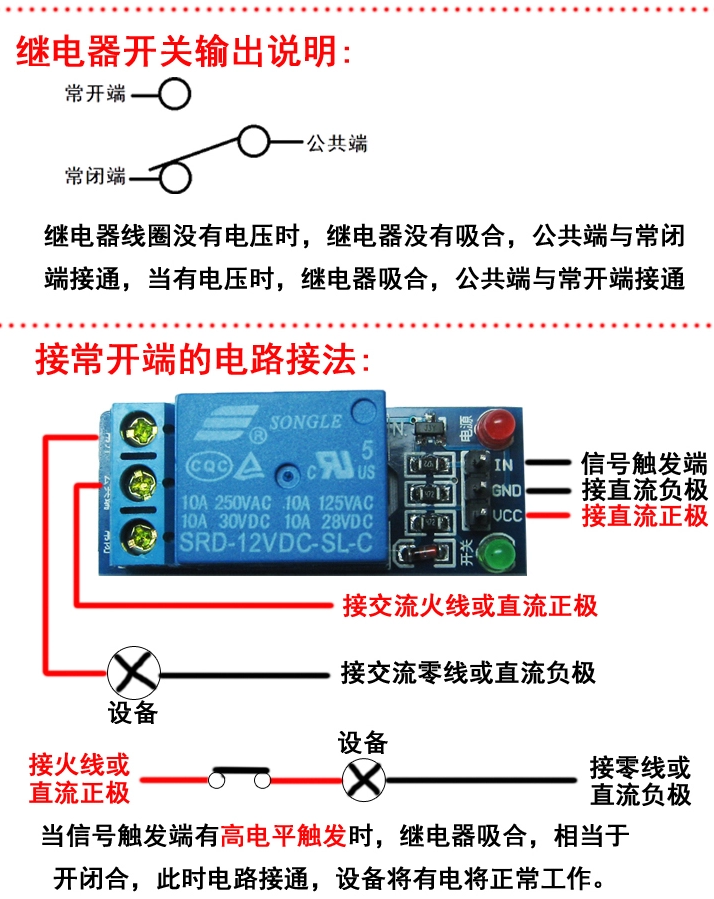
多谐振荡器由晶体管或集成电路极成。当接通电源后（1.5~15V直流工作电压）,多谐振荡器起振,输出1.5~2.5kHZ的音频信号，阻抗匹配器推动压电蜂鸣片发声压电蜂鸣片由锆钛酸铅或铌镁酸铅压电陶瓷材料制成。在陶瓷片的两面镀上银电极，经极化和老化处理后，再与黄铜片或不锈钢片粘在一起。

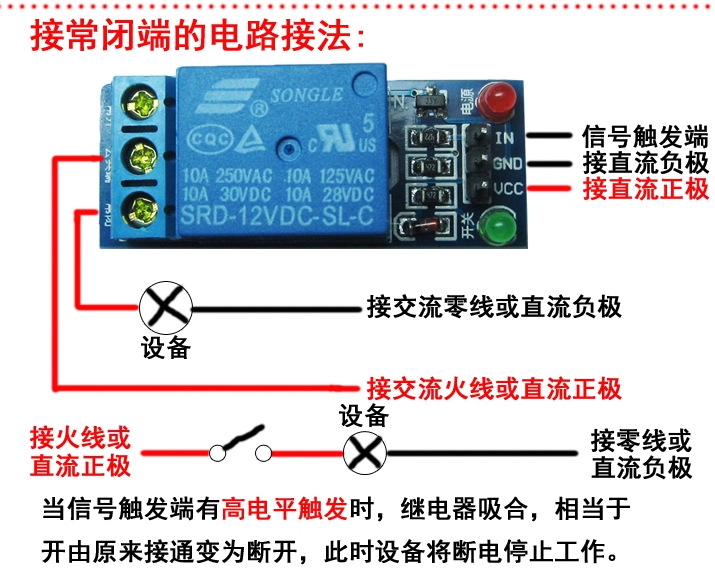
电磁式蜂鸣器由振荡器、电磁线圈、磁铁、振动膜片及外壳等组成。接通电源后，振荡器产生的音频信号电流通过电磁线圈，使电磁线圈产生磁场。振动膜片在电磁线圈和磁铁的相互作用下，周期性地振动发声。

### 2.2.2继电器

继电器可用来间接控制大功率的电器，我们可将它看作是一个由arduino控制的开关。连线时，将VCC，GND分别和arduino上的5V，GND相连；IN和arduino的Digital输出端口相连，就可以参照上一节的办法控制继电器的开关了。将大功率负载接在接线柱上，就实现了arduino对此负载的控制。







## 2.3 数码管实验

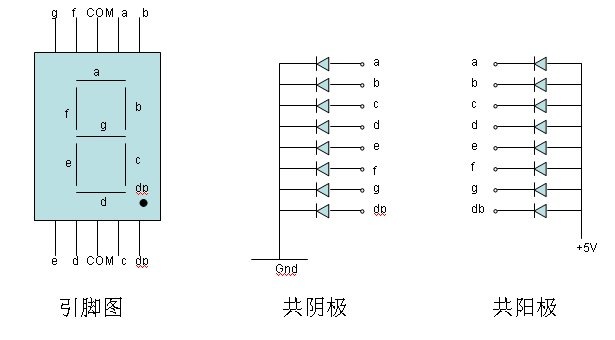
数码管是一种半导体发光器件，其基本单元是发光二极管.数码管按段数分为七段数码管和八段数码管，八段数码管比七段数码管多一个发光二极管单元（多一个小数点显示）；按能显示多少个“8”可分为1位、2位、4位等等数码管；





按发光二极管单元连接方式分为共阳极数码管和共阴极数码管。

共阳数码管是指将所有发光二极管的阳极接到一起形成公共阳极(COM)的数码管。共阳数码管在应用时应将公共极COM接到+5V，**当某一字段发光二极管的阴极为低电平时，相应字段就点亮。**当某一字段的阴极为高电平时，相应字段就不亮。



将数字I/O和数码管的字段引脚相连，公共极COM如果是共阳极的就接到+5V，如果是共阴极的就接到GND。公共极上需要接限流电阻，否则电流过大会烧毁发光二极管的。

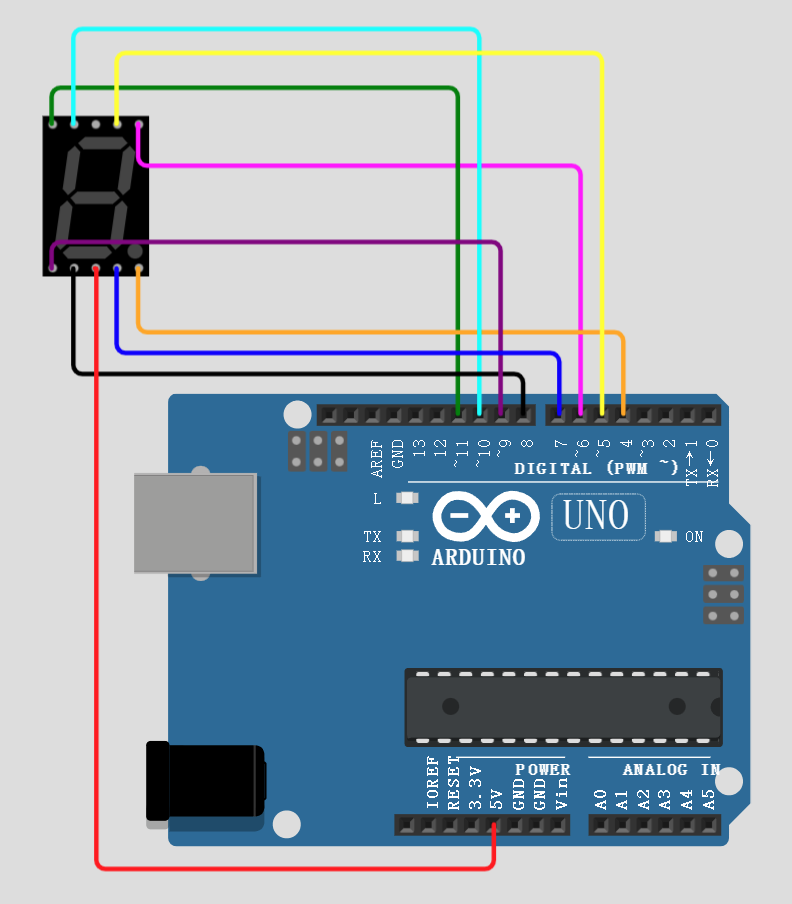
#### 元件清单

数码管：1个 220Ω的电阻：1个 多彩面包板实验跳线：若干

#### 电路连接

COM端接5V（共阳），注意需要接220欧限流电阻（在网页模拟器中可以忽略）。

各管脚的名称参见上面的图。和Arduino管脚号的连接关系参见程序代码中红色字体部分。



#### 程序代码

int a=5; //设定控制LED的数字IO脚

int b=6; //设定控制LED的数字IO脚

int c=7; //设定控制LED的数字IO脚

int d=8; //设定控制LED的数字IO脚

int e=9; //设定控制LED的数字IO脚

int f=10; //设定控制LED的数字IO脚

int g=11; //设定控制LED的数字IO脚

int dp=4; //设定控制LED的数字IO脚

void setup()

{

pinMode(a,OUTPUT);

pinMode(b,OUTPUT);

pinMode(c,OUTPUT);

pinMode(d,OUTPUT);

pinMode(e,OUTPUT);

pinMode(f,OUTPUT);

pinMode(g,OUTPUT);

pinMode(dp,OUTPUT);

}

void loop()

{

digital\_1();

delay(1000); //设定延时时间

digital\_2();

delay(1000); //设定延时时间

}

void digital\_1()

{

digitalWrite(a,HIGH);

digitalWrite(b,LOW);

digitalWrite(c,LOW);

digitalWrite(d,HIGH);

digitalWrite(e,HIGH);

digitalWrite(f,HIGH);

digitalWrite(g,HIGH);

}

//显示数字2

void digital\_2()

{

digitalWrite(a,LOW);

digitalWrite(b,LOW);

digitalWrite(c,HIGH);

digitalWrite(d,LOW);

digitalWrite(e,LOW);

digitalWrite(f,HIGH);

digitalWrite(g,LOW);

}

//显示数字3

void digital\_3()

{

digitalWrite(a,LOW);

digitalWrite(b,LOW);

digitalWrite(c,LOW);

digitalWrite(d,LOW);

digitalWrite(e,HIGH);

digitalWrite(f,HIGH);

digitalWrite(g,LOW);

}

//显示数字4

void digital\_4()

{

digitalWrite(a,HIGH);

digitalWrite(b,LOW);

digitalWrite(c,LOW);

digitalWrite(d,HIGH);

digitalWrite(e,HIGH);

digitalWrite(f,LOW);

digitalWrite(g,LOW);

}

//显示数字5

void digital\_5()

{

digitalWrite(a,LOW);

digitalWrite(b,HIGH);

digitalWrite(c,LOW);

digitalWrite(d,LOW);

digitalWrite(e,HIGH);

digitalWrite(f,LOW);

digitalWrite(g,LOW);

}

//显示数字6

void digital\_6()

{

digitalWrite(a,LOW);

digitalWrite(b,HIGH);

digitalWrite(c,LOW);

digitalWrite(d,LOW);

digitalWrite(e,LOW);

digitalWrite(f,LOW);

digitalWrite(g,LOW);

}

//显示数字7

void digital\_7()

{

digitalWrite(a,LOW);

digitalWrite(b,LOW);

digitalWrite(c,LOW);

digitalWrite(d,HIGH);

digitalWrite(e,HIGH);

digitalWrite(f,HIGH);

digitalWrite(g,HIGH);

}

//显示数字8

void digital\_8()

{

digitalWrite(a,LOW);

digitalWrite(b,LOW);

digitalWrite(c,LOW);

digitalWrite(d,LOW);

digitalWrite(e,LOW);

digitalWrite(f,LOW);

digitalWrite(g,LOW);

}

//显示数字9

void digital\_9()

{

digitalWrite(a,LOW);

digitalWrite(b,LOW);

digitalWrite(c,LOW);

digitalWrite(d,LOW);

digitalWrite(e,HIGH);

digitalWrite(f,LOW);

digitalWrite(g,LOW);

}

//显示数字a

void digital\_a()

{

digitalWrite(a,LOW);

digitalWrite(b,LOW);

digitalWrite(c,LOW);

digitalWrite(d,HIGH);

digitalWrite(e,LOW);

digitalWrite(f,LOW);

digitalWrite(g,LOW);

}

//显示数字b

void digital\_b()

{

digitalWrite(a,HIGH);

digitalWrite(b,HIGH);

digitalWrite(c,LOW);

digitalWrite(d,LOW);

digitalWrite(e,LOW);

digitalWrite(f,LOW);

digitalWrite(g,LOW);

}

//显示数字c

void digital\_c()

{

digitalWrite(a,LOW);

digitalWrite(b,HIGH);

digitalWrite(c,HIGH);

digitalWrite(d,LOW);

digitalWrite(e,LOW);

digitalWrite(f,LOW);

digitalWrite(g,HIGH);

}

//显示数字d

void digital\_d()

{

digitalWrite(a,HIGH);

digitalWrite(b,LOW);

digitalWrite(c,LOW);

digitalWrite(d,LOW);

digitalWrite(e,LOW);

digitalWrite(f,HIGH);

digitalWrite(g,LOW);

}

//显示数字e

void digital\_e()

{

digitalWrite(a,LOW);

digitalWrite(b,HIGH);

digitalWrite(c,HIGH);

digitalWrite(d,LOW);

digitalWrite(e,LOW);

digitalWrite(f,LOW);

digitalWrite(g,LOW);

}

//显示数字f

void digital\_f()

{

digitalWrite(a,LOW);

digitalWrite(b,HIGH);

digitalWrite(c,HIGH);

digitalWrite(d,HIGH);

digitalWrite(e,LOW);

digitalWrite(f,LOW);

digitalWrite(g,LOW);

}

//清屏

void clear()

{

digitalWrite(a,HIGH);

digitalWrite(b,HIGH);

digitalWrite(c,HIGH);

digitalWrite(d,HIGH);

digitalWrite(e,HIGH);

digitalWrite(f,HIGH);

digitalWrite(g,HIGH);

digitalWrite(dp,HIGH);

}

void setpoint(bool isOn)

{

if(isOn)

digitalWrite(dp,LOW);

else

digitalWrite(dp,HIGH);

}

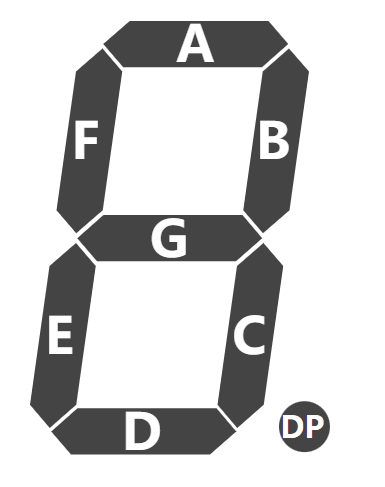
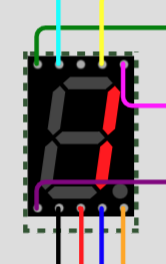
#### 程序功能

运行后我们可以看到，数码管循环显示数字1，2。掌握本程序后，大家可以发挥自己的想象，做出各种数码管实验。

#### 实验原理

数码管共有七段显示数字的段，还有一个显示小数点的段。当让数码管显示数字时，只要将相应的段点亮。

实例将每个数字的显示功能写成一个子程序，如Digital\_1() 把b,c段设为低电平0V（点亮），则显示1.

在主程序中调用相应子程序，可以方便的控制数码管。

#### 练习

示例中缺少显示0的子程序，请自行编写，并在主程序中调用，实现循环显示0，1，2.（提示：0和8很接近，可以在digital\_8()的基础上复制、修改。不必从头做）

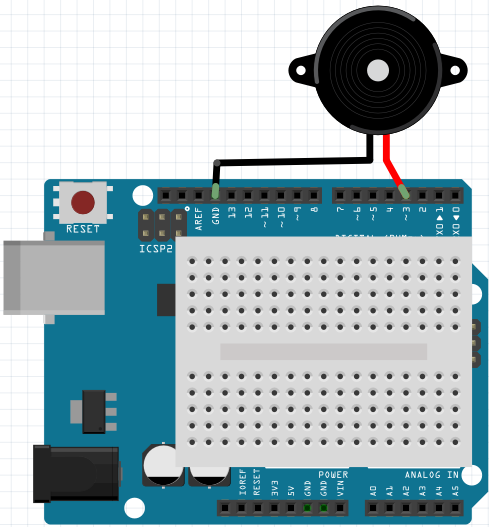
结合前面的交通灯项目，制作包含倒计时的交通灯。

**回答<https://ks.wjx.top/vj/wiMVz0m.aspx>中的问题，作为本节课出勤记录。**

# 第三章 Arduino模拟输出

## 3.1 简单音乐制作

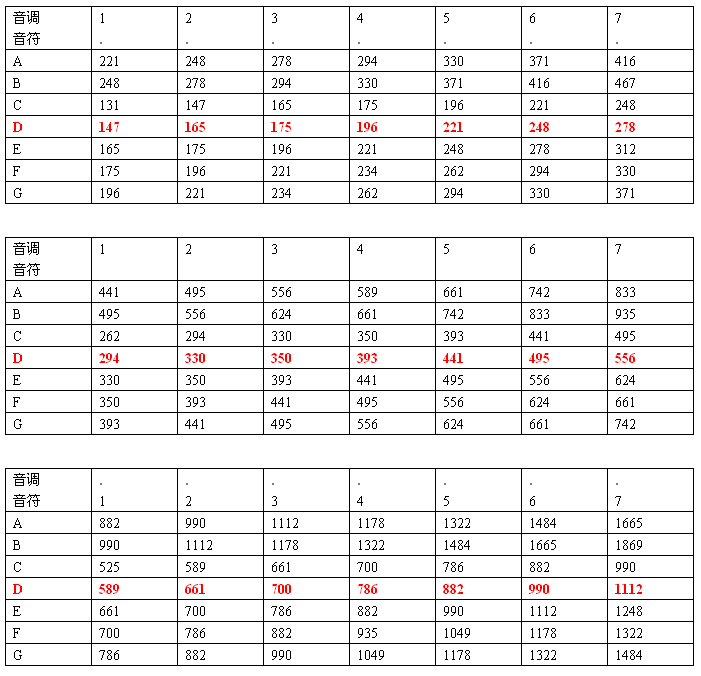
#### 元件清单及电路连接



#### 工作原理

首先讲下简单的乐理知识，知道音乐是怎么演奏出来的自然就可以通过代码来进行编排了。

1.演奏单音符  一首乐曲有若干音符组成，一个音符对应一个频率。这里有个表供大家参考.按表中的频率，在程序中使用tone()函数，可以播放声音。

 2.音符的演奏时间  音符节奏分为一拍、半拍、1/4拍、1/8拍，我们规定一拍音符的时间为1；半拍为0.5；1/4拍为0.25；1/8拍为0.125……，所以我们可以为每个音符赋予这样的拍子播放出来，音乐就成了。用delay()函数可以控制时间长度。

我们看看如何将简谱翻译成对应频率和拍子。以葫芦娃为例：

 先看下左上角1=D这里，用的是D调，好那我们就看《音符频率表》中的D行（红色部分），

第一个音符是1，时间是一拍=1，第二个音符6（没有点），时间也是一拍=1，第三个音符5，因为有下划线所以是半拍=0.5，……以此类推。

第四0这里要注意下，这里是没有声音，拍子=1拍。

第五，再接着看到第一句歌词葫芦娃 这个娃的音是3—，这表示是两拍，后面每加一个“-”，表示拍子+1（1+1），本例中最多是加到4.

第六，第二句歌词葫芦娃这个娃，3•带个点，点的意思是去3的拍子的一半，即3•的拍子是1+0.5

第七，大家可能会问那弧线怎么表示，这在音乐中属于连音，操作上更复杂了，本例没有做连音的处理。

所以规律就是时间上单个音符没有下划线，就是一拍（1），有下划线是半拍（0.5），两个下划线是四分之一拍（0.25），有“—”=前面音符的拍子+1；频率上就是按照音符是否带点，点在上还是在下到表中查找就可以了。

至此原理清楚，随便拿个简谱来我们都可以翻译成代码了。

#### 程序代码

//每种声音对应的频率

#define D0 -1

#define D1 294

#define D2 330

#define D3 370

#define D4 393

#define D5 441

#define D6 495

#define D7 556

#define DL1 147

#define DL2 165

#define DL3 190

#define DL4 196

#define DL5 221

#define DL6 248

#define DL7 278

#define DH1 589

#define DH2 661

#define DH3 700

#define DH4 786

#define DH5 882

#define DH6 990

#define DH7 1120

//乐谱：音阶

int tune[] =

{

DH1,D6,D5,D6,D0,

DH1,D6,D5,DH1,D6,D0,D6,

D6,D6,D5,D6,D0,D6,

DH1,D6,D5,DH1,D6,D0,

D1,D1,D3,

D1,D1,D3,D0,

D6,D6,D6,D5,D6,

D5,D1,D3,D0,

DH1,D6,D6,D5,D6,

D5,D1,D2,D0,

D7,D7,D5,D3,

D5,

DH1,D0,D6,D6,D5,D5,D6,D6,

D0,D5,D1,D3,D0,

DH1,D0,D6,D6,D5,D5,D6,D6,

D0,D5,D1,D2,D0,

D3,D3,D1,DL6,

D1,

D3,D5,D6,D6,

D3,D5,D6,D6,

DH1,D0,D7,D5,

D6,

};

//乐谱：拍数。例如：四分音符为1拍，二分为2，八分为0.5.

float duration[]=

{

1,1,0.5,0.5,1,

0.5,0.5,0.5,0.5,1,0.5,0.5,

0.5,1,0.5,1,0.5,0.5,

0.5,0.5,0.5,0.5,1,1,

1,1,1+1,

0.5,1,1+0.5,1,

1,1,0.5,0.5,1,

0.5,1,1+0.5,1,

0.5,0.5,0.5,0.5,1+1,

0.5,1,1+0.5,1,

1+1,0.5,0.5,1,

1+1+1+1,

0.5,0.5,0.5+0.25,0.25,0.5+0.25,0.25,0.5+0.25,0.25,

0.5,1,0.5,1,1,

0.5,0.5,0.5+0.25,0.25,0.5+0.25,0.25,0.5+0.25,0.25,

0.5,1,0.5,1,1,

1+1,0.5,0.5,1,

1+1+1+1,

0.5,1,0.5,1+1,

0.5,1,0.5,1+1,

1+1,0.5,0.5,1,

1+1+1+1

};

int tonePin=3;//蜂鸣器的端口

void setup()

{

pinMode(tonePin,OUTPUT);//设置蜂鸣器的pin为输出模式

}

void loop()

{

int length = sizeof(tune)/sizeof(tune[0]);//固定格式，查出tone序列里有多少个音符

for(int x=0;x<length;x++)//循环音符的次数

{

tone(tonePin,tune[x]);//此函数依次播放tune序列里的数组，即每个音符

delay(400\*duration[x]);//每个音符持续的时间，即节拍duration，400是调整时间//越大，曲子速度越慢，越小曲子速度越快，自己掌握吧

noTone(tonePin);//停止当前音符，进入下一音符

}

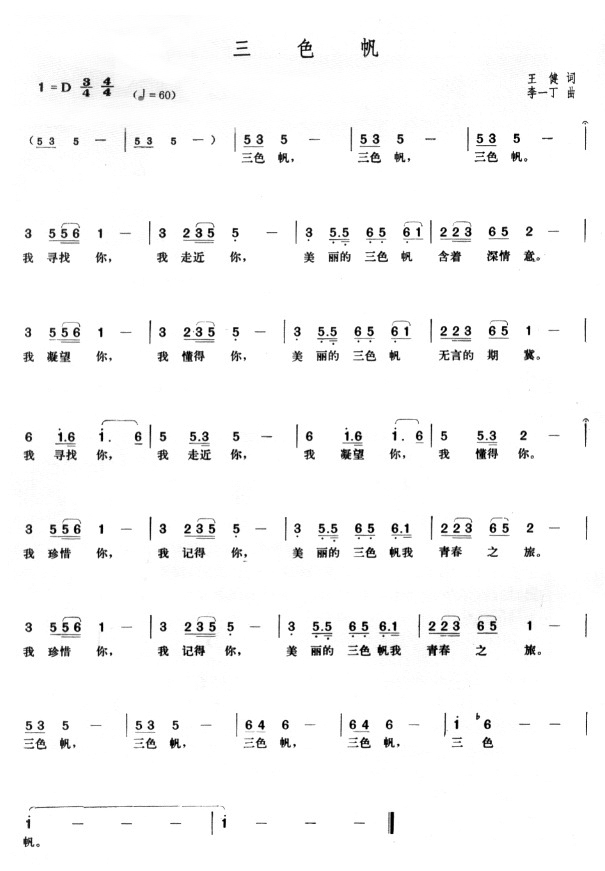
delay(2000);//等待2秒后重新播放

}

思考

1.前面示例程序中有个音不准，请改正。

2.编写三色帆校歌乐曲，简谱如下：



参考乐谱：

int tune[] =

{

D5,D3,D5,D5,D3,D5,D5,D3,D5,

D3,D5,D5,D6,D1,D3,D2,D3,D5,DL5,DL3,DL5,DL5,DL6,DL5,DL6,D1,D2,D2,D3,D6,D5,D2,

D3,D5,D5,D6,D1,D3,D2,D3,D5,DL5,DL3,DL5,DL5,DL6,DL5,DL6,D1,D2,D2,D3,D6,D5,D1,

D6,DH1,D6,DH1,D6,D5,D5,D3,D5,D6,DH1,D6,DH1,D6,D5,D5,D3,D2,

D3,D5,D5,D6,D1,D3,D2,D3,D5,DL5,DL3,DL5,DL5,DL6,DL5,DL6,D1,D2,D2,D3,D6,D5,D2,

D3,D5,D5,D6,D1,D3,D2,D3,D5,DL5,DL3,DL5,DL5,DL6,DL5,DL6,D1,D2,D2,D3,D6,D5,D1,

D5,D3,D5,D5,D3,D5,D6,D4,D6,D6,D4,D6,DH1,D5,DH1

};

float duration[]=

{

0.5,0.5,2,0.5,0.5,2,0.5,0.5,2,

1,0.5,0.25,0.25,2,1,0.5,0.25,0.25,2,1,0.5,0.25,0.5,0.5,0.5,0.5,0.5,0.25,0.25,0.5,0.5,2,

1,0.5,0.25,0.25,2,1,0.5,0.25,0.25,2,1,0.5,0.25,0.5,0.5,0.5,0.5,0.5,0.25,0.25,0.5,0.5,2,

1,0.75,0.25,1,0.5,1,0.75,0.25,2,1,0.75,0.25,1,0.5,1,0.75,0.25,2,

1,0.5,0.25,0.25,2,1,0.5,0.25,0.25,2,1,0.75,0.25,0.5,0.5,0.75,0.25,0.5,0.25,0.25,0.5,0.5,2,

1,0.5,0.25,0.25,2,1,0.5,0.25,0.25,2,1,0.75,0.25,0.5,0.5,0.75,0.25,0.5,0.25,0.25,0.5,0.5,2,

0.5,0.5,2,0.5,0.5,2,0.5,0.5,2,0.5,0.5,2,1,3,8,

};

**同时必须修改播放速度。**

2.在上述基础上，每播放一个音符闪一次灯。

参考答案（LED接digital 6口）

void loop()

{

int length = sizeof(tune)/sizeof(tune[0]);//固定格式，查出tone序列里有多少个音符

for(int x=0;x<length;x++)//循环音符的次数

{

tone(tonePin,tune[x]);//此函数依次播放tune序列里的数组，即每个音符

digitalWrite(6,LOW);

delay(400\*duration[x]);//每个音符持续的时间，即节拍duration，400调整时间的

//越大，曲子速度越慢，越小曲子速度越快，自己掌握吧

digitalWrite(6,HIGH);

delay(400\*duration[x]);

noTone(tonePin);//停止当前音符，进入下一音符

}

delay(2000);//等待2秒后重新播放

}

教学建议

建议2-3课时完成第二章。1-2课时完成第三章。

2.1包含演示视频，可帮助学生熟悉环境，建议教师至少演示1次。

交通灯和数码管任务后面有问卷星链接，可作为平时成绩计算依据。