

## 项目四 呼吸灯

在前面几章中，我们知道了如何通过程序来控制 LED 亮灭。但 Arduino 还有个很强大的功能通过程序来控制 LED 的明亮度。Arduino UNO 数字引脚中有六个引脚标有“~”，这个符号就说明该口具有 PWM 功能。我们动手做一下，在做的过程中体会 PWM 的神奇力量！下面就介绍一个呼吸灯，所谓呼吸灯，就是让灯有一个由亮到暗，再到亮的逐渐变化的过程，感觉像是在均匀的呼吸。

### 所需元件

● 1× 5mm LED 灯



● 1× 220 欧电阻



### 硬件连接

这个项目的硬件连接与项目一是完全相同的。如有不明白，可以回看项目一。

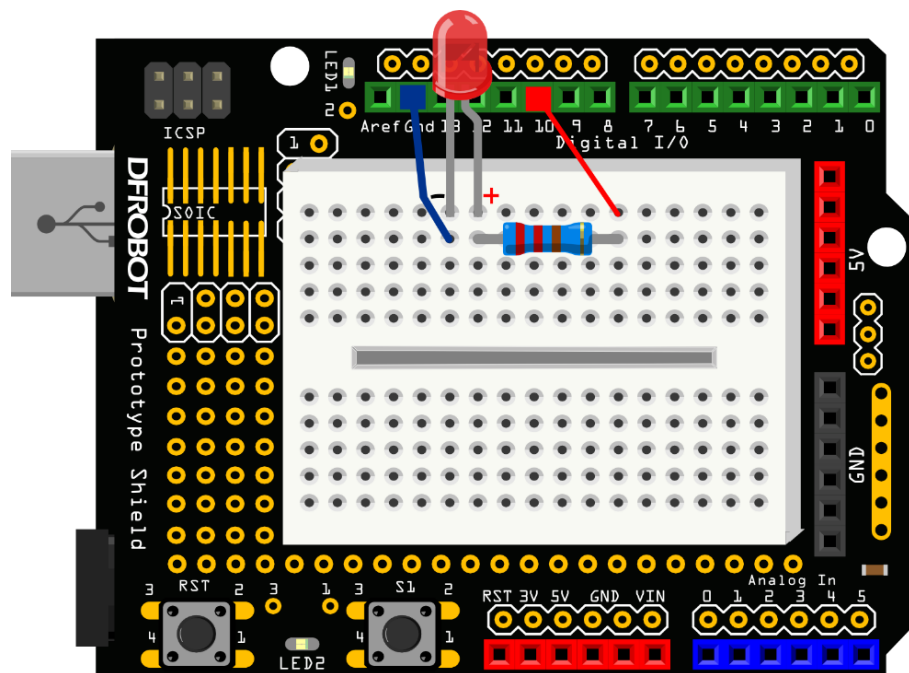


图 4-1 呼吸灯连线图

## 输入代码

样例代码 4-1:

```
//项目四 - 呼吸灯

int ledPin = 10;

void setup() {
    pinMode(ledPin, OUTPUT);
}

void loop() {
    fadeOn(1000, 5);
    fadeOff(1000, 5);
}

void fadeOn(unsigned int time,int increament){
    for (byte value = 0 ; value < 255; value+=increament){
        analogWrite(ledPin, value);
        delay(time/(255/5));
    }
}

void fadeOff(unsigned int time,int decreament){
    for (byte value = 255; value >0; value-=decreament){
        analogWrite(ledPin, value);
        delay(time/(255/5));
    }
}
```

代码下载完成后，我们可以看到 LED 会有个逐渐由亮到灭的一个缓慢过程，而不是直接的亮灭，如同呼吸一般，均匀变化。

## 代码回顾

大部分代码我们已经很熟悉了，比如初始化变量声明、引脚设置、for 循环、以及函数调用。

在主函数中，只有两个调用函数，先看其中一个就能明白了。

```
void fadeOn(unsigned int time,int increament){  
    for (byte value = 0 ; value < 255; value+=increament){  
        analogWrite(ledPin, value);  
        delay(time/(255/5));  
    }  
}
```

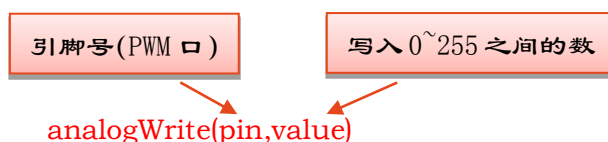
fadeOn()函数有两个传递参数，从参数名称中就可以简单看出，int time 指的是时间，int increament 指的是增量。函数中包含了一个 for 循环，循环条件是 value<255，变量的增量由 increament 决定。

for 语句中涉及了一个新函数

```
analogWrite(ledPin, value);
```

如何发送一个模拟值到一个数字引脚呢？就要用到该函数，使用这个函数是要具备特定条件的——该数字引脚需具有 PWM 功能。观察一下 Arduino 板，查看数字引脚，你会发现其中 6 个引脚（3、5、6、9、10、11）旁标有“~”，这些引脚不同于其他引脚，因为它们可以输出 PWM 信号。

函数格式如下：



analogWrite()函数用于给 PWM 口写入一个 0~255 的模拟值。特别注意的是，analogWrite()函数只能写入具有 PWM 功能的数字引脚。

PWM 是一项通过数字方法来获得模拟量的技术。数字控制来形成一个方波，方波信号只有开关两种状态（也就是我们数字引脚的高低）。通过控制开与关所持续时间的比值就能

模拟到一个 0 到 5V 之间变化的电压。开（学术上称为高电平）所占用的时间就叫做脉冲宽度，所以 PWM 也叫做脉冲宽度调制。

通过下面五个方波来更形象的了解一下 PWM。

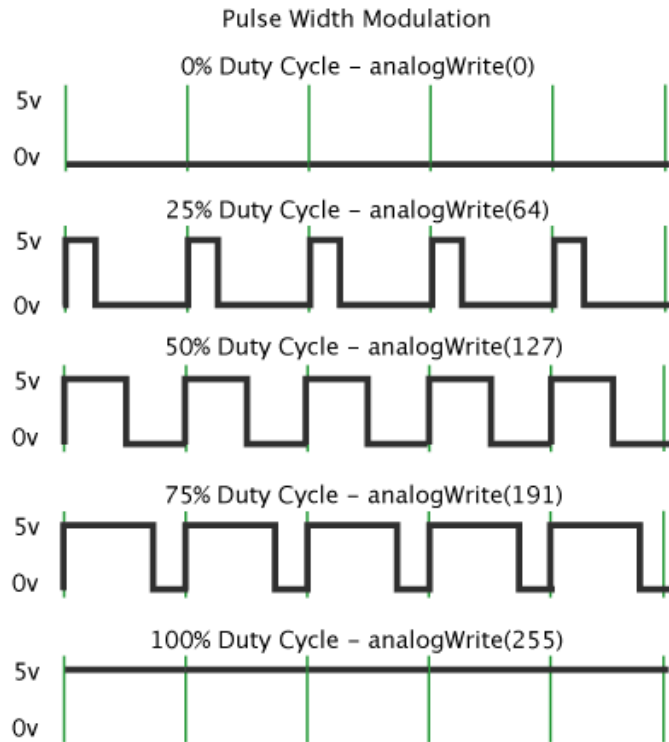


图 4-2 PWM 示意图

上图绿色竖线代表方波的一个周期。每个 `analogWrite(value)` 中写入的 `value` 都能对应一个百分比，这个百分比也称为占空比(Duty Cycle)，指的是一个周期内高电平持续时间比上低电平持续时间得到的百分比。图中，从上往下，第一个方波，占空比为 0%，对应的 `value` 为 0。LED 亮度最低，也就是灭的状态。高电平持续时间越长，也就越亮。所以，最后一个占空比为 100% 的对应 `value` 是 255，LED 最亮。50% 就是最亮的一半了，25% 则相对更暗。

PWM 比较多的用于调节 LED 灯的亮度。或者是电机的转动速度，电机带动的车轮速度也就能很容易控制了，在玩一些 Arduino 小车时，更能体现 PWM 的好处。

这一章介绍结束了！同样的硬件连接，通过软件的变化，可以呈现出完全不一样的效果，是不是觉得 Arduino 很神奇！

## 课后作业

1.用 LED 能否做个火焰的效果，通过 PWM 使 LED 产生随机的亮度变化，来模拟一个火焰闪烁的效果。找个用个浅色罩子盖住效果更佳，可以放在家中作为小夜灯。

主要材料：一个红色 LED、两个黄色 LED 以及 220 欧电阻。在这个实验中，有个函数会比较好用——`random()`。`random()`可是产生一定范围内的随机数。

提示：可以先设定 LED 灯亮度，在其值附近产生一个随机数，比如 `random(120)+135`，让其值稳定在 135 附近，产生这种小幅变化，就更具有火焰跳跃感。不妨尝试一下。

具体用法可以查看下面链接的编程参考手册，会详细介绍这个函数的用法。之后的讲解中，我们可能有对些函数不进行详细说明，你可以通过这种方法来学习某个新函数。

点击查看：[DFRobot 中文版 Arduino 编程参考手册](#)

点击查看：[Arduino 官方编程参考手册](#)

2.再尝试一个稍微有点难度的，通过两个按键，一个按键控制 LED 逐次变亮，另一个按键控制 LED 逐次变暗。

可以参考程序：<http://www.geek-workshop.com/thread-1054-1-1.html>