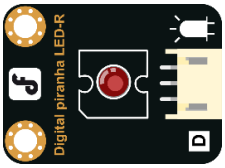
项目四 呼吸灯

在前面几章中，我们知道了如何控制 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。

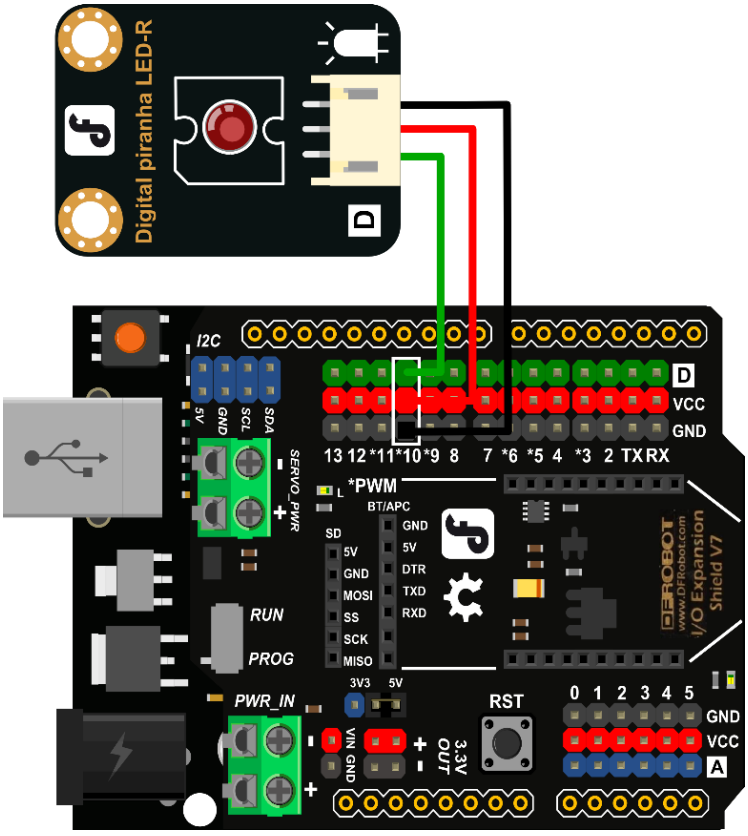
所需元件

- 1×　数字食人鱼红色 LED 发光模块



硬件连接

数字食人鱼红色 LED 发光模块 → 数字 10



输入代码

样例代码 5-1：

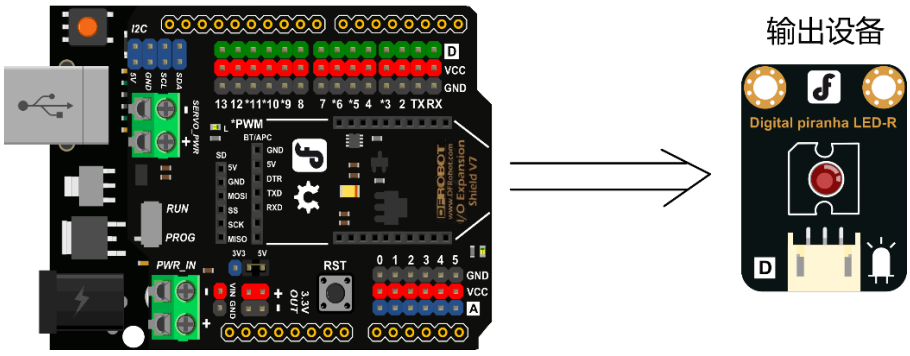
```
//项目五 – 呼吸灯
int ledPin = 10;

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

void loop(){
  for (int value = 0 ; value < 255; value=value-1){
     analogWrite(ledPin, value);
     delay(5);
  }
  for (int value = 255; value >0; value=value-1){
     analogWrite(ledPin, value);
     delay(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。

硬件分析（模拟输出）

和项目一（点亮一盏灯）类似的装置，同样没有输入设备，只有一个输出设备，但又有所不同。项目一 LED 是作为数字输出，而这里我们是作为模拟输出。代码部分会说明。

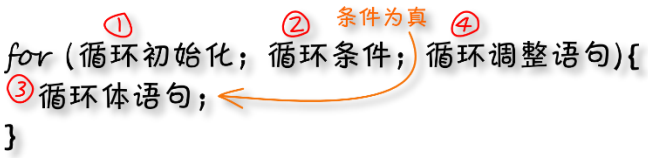
控制设备



代码分析

当我们需要重复执行某句话时，我们可以使用 for 语句。

for 语句格式如下：



for 循环顺序如下：

第一轮：1 → 2 → 3 → 4

第二轮：2 → 3 → 4

…

直到 2 不成立，for 循环结束。

知道了这么个顺序之后，回到代码中：

```
for (int value = 0; value < 255; value=value+1){
        ...
}
for (int value = 255; value >0; value=value-1){
        ...
}
```

这两个 for 语句实现了 value 的值不断由 0 增加到 255，随之在从 255 减到 0，在增加到 255……,无限循环下去。

再看下 for 里面，涉及一个新函数 analogWrite()。

我们知道数字口只有 0 和 1 两个状态，那如何发送一个模拟值到一个数字引脚呢？就要用到该函数。观察一下 Arduino 板，查看数字引脚，你会发现其中 6 个引脚旁标有“~”，这些引脚不同于其他引脚，它们可以输出 PWM 信号。

函数格式如下：

```
analogWrite(pin,value)
```

analogWrite()函数用于给 PWM 口写入一个 0~255 的模拟值。所以，value 是在 0~255 之间的值。特别注意的是，analogWrite()函数只能写入具有 PWM 功能的数字引脚，也就是 3，5，6，9，10，11 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。

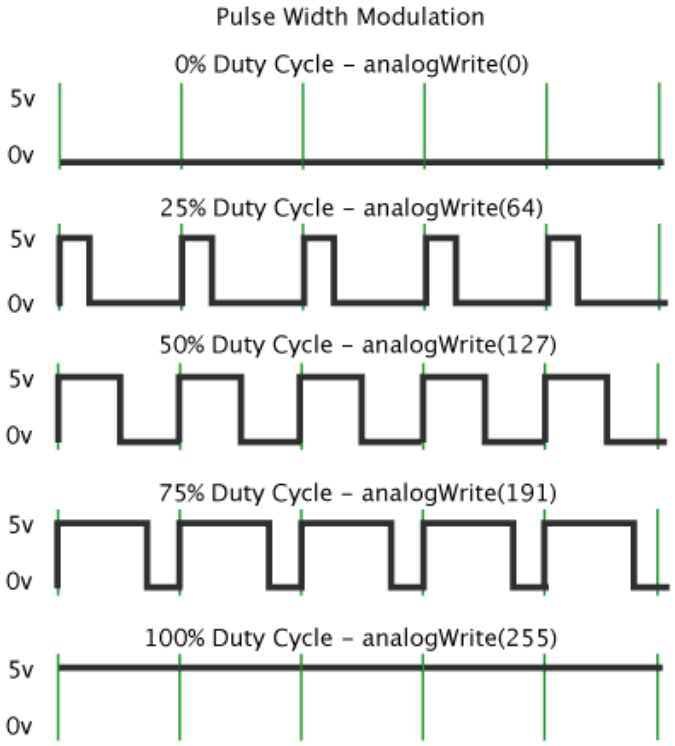


图 4-2 PWM 示意图

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。

这一节介绍结束了!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 很酷炫!