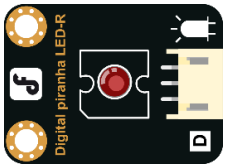
项目六 灯光调节器

所谓灯光调节器，就是可以自由控制灯的亮度，我们这里通过一个模拟角度传感器来 LED 灯的亮度。随着旋转角度的变化，LED 亮度也发生相应改变。角度越大，LED 灯也就越亮，相反，角度越小，LED 灯也就越暗。这里只是用了小小的 LED 来做演示效果，如果想运用到我们的生活之中的话，也是同样的原理。那就先做个小型的灯光调节器吧!

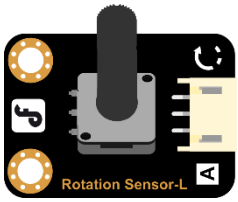
模拟角度传感器还能用到很多地方,比如我们后面会接触的舵机,可以通过这个传感器来控制转动角度,又或者以后有机会接触直流电机的小伙伴,可以尝试下用角度传感器来控制转速等等,用处很多!

所需元件

- 1× 数字食人鱼红色 LED 发光模块



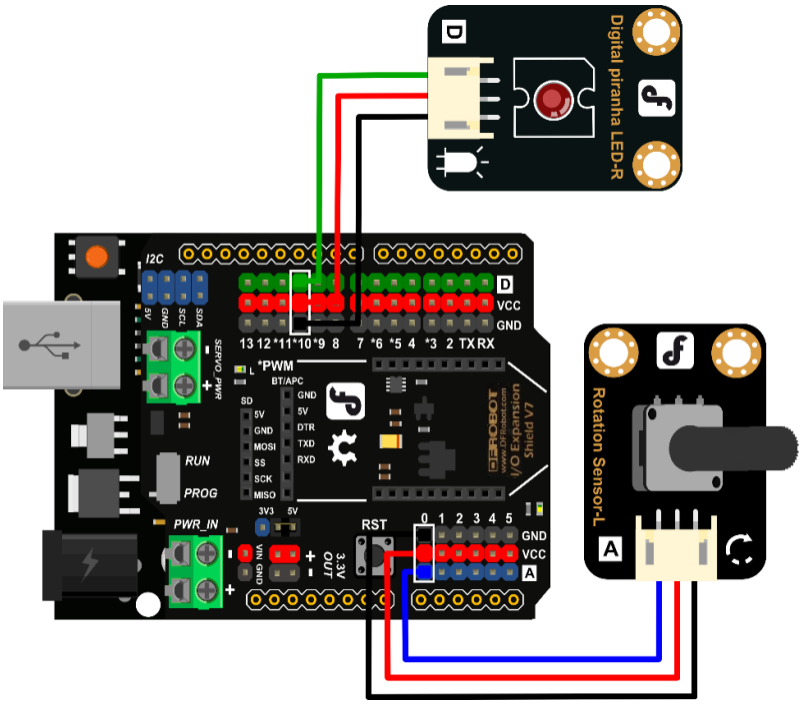
- 1× 模拟角度传感器



硬件连接

模拟角度传感器 → 模拟 0

数字食人鱼红色 LED 发光模块 → 数字 10



输入代码

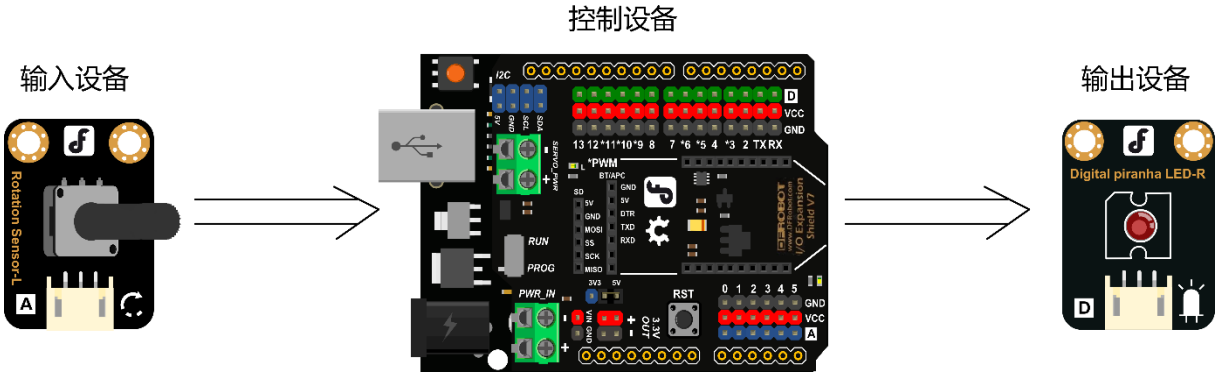
样例代码 6-1：

```
//项目六 —— 灯光调节器
int potPin = 0;                    //电位器连接到模拟 0
int ledPin = 10;                   //LED 连接到数字 10

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

void loop() {
    int sensorValue = analogRead(potPin);    //读取模拟口 0 的值
    //通过 map()把 0~1023 的值转换为 0~255
    int outputValue = map(sensorValue, 0, 1023, 0, 255);
    analogWrite(ledPin, outputValue);   //给 LED 写入对应值
    delay(2);
}
```

缓慢旋转电位器，仔细观察 LED 的亮度是否发生变化。

硬件分析（模拟输入—模拟输出）

在呼吸灯一节，我们已经学会了如何用数字引脚的 PWM 口来做模拟输出。这一节将加入互动元素，通过模拟输入来控制模拟输出。



代码回顾

这里主要讲下 map 函数。

函数格式如下：

```
map(value, fromLow, fromHigh, toLow, toHigh)
```

map 函数的作用是将一个数从一个范围映射到另外一个范围。也就是说，会将 fromLow 到 fromHigh 之间的值映射到 toLow 在 toHigh 之间的值。

map 函数参数含义：

value：需要映射的值

fromLow：当前范围值的下限

fromHigh：当前范围值的上限

toLow：目标范围值的下限

toHigh：目标范围值的上限

map 的神奇之处还在于，两个范围中的“下限”可以比“上限”更大或者更小，因此 map()函数可以用来翻转数值的范围，可以这么写：

```
y = map(x, 1, 50, 50, 1);
```

这个函数同样可以处理负数，请看下面这个例子：

```
y = map(x, 1, 50, 50, -100);
```

回到代码中，

```
int outputValue = map(sensorValue, 0, 1023, 0, 255);
```

我们是想将模拟口读到的 0~1023 的值，转换为 PWM 口的 0~255。