舵机是一种电机,它使用一个反馈系统来控制电机的位置。可以很好掌握电机角度。大多数舵机是可以最大旋转 180°的。也有一些能转更大角度,甚至 360°。舵机比较多的用于对角度有要求的场合,比如摄像头,智能小车前置探测器,需要在某个范围内进行监测的移动平台。又或者把舵机放到玩具,让玩具动起来。还可以用多个舵机,做个小型机器人,舵机就可以作为机器人的关节部分。所以,舵机的用处很多。

Ardruino 也提供了<Servo.h>库,让我们使用舵机变得更方便了。

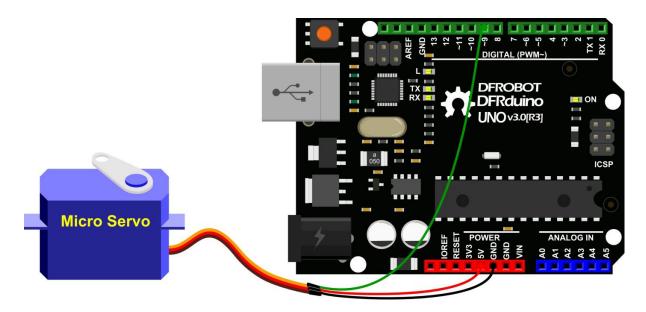
先从简单入手,套件这个 9G 小舵机是  $180^{\circ}$ 的,我们就让它在  $0\sim180^{\circ}$ 之间来回转动。

#### 所需材料

- 1× DFduino UNO R3
- 1× Micro Servo 9g

### STEP 1: 硬件连接

这个项目的连线很简单,只需按图所示连接舵机三根线就可以了,连的时候注意线序,舵机引出三根线。一根是红色,连到+5V上。一根棕色(有些是黑的),连到 GND。还有一根是黄色或者橘色,连到数字引脚 9。



### STEP 2: 输入代码

```
1. #include <Servo.h> // 声明调用 Servo.h 库
2. Servo myservo; // 创建一个舵机对象
3. int pos = 0; // 变量 pos 用来存储舵机位置
4. void setup() {
5. myservo.attach(9); // 将引脚9上的舵机与声明的舵机对象连接起来
6. }
7.
8. void loop() {
9. for(pos = 0; pos < 180; pos += 1){ // 舵机从 0°转到 180°,
  每次增加1°
10. myservo.write(pos); // 给舵机写入角度
11. delay(15); // 延时 15ms 让舵机转到指定位
 置.
12.
13. for (pos = 180; pos>=1; pos-=1) { // 舵机从 180°转回到 0°,
  每次减小1°
14. myservo.write(pos); // 写角度到舵机
15. delay(15); // 延时 15ms 让舵机转到指定位置
16.
17. }
```

下载代码,下载成功后我们可以看到舵机0~180°来回转动。

### STEP 3: 代码回顾

代码的开始先调用 < Servo.h > 库

1. #include <Servo.h>

这个库已经在 Arduino IDE 中了,可以打开 Arduino-1.0.5/ libraries/ Servo/ Servo.h, 这就是 Servo 库所在位置。

我们怎么理解库呢?和我们前面讲到的函数意义是差不多的。函数通常按一个个功能来划分的,就像一个个小的储物柜,函数名好比储物柜标签名。我们使用的时候,直接看标签就好了,方便我们使用。那库是什么呢?库则是把多个函数封装打包起来,好比大的储物柜,里面含有一个个小的储物柜。不知道这样说,你是不是能理解库和函数的关系?

同样,大储物柜也需要一个标签,这标签的学术名叫做"对象"。所以这里叫创建一个对象。 就是我们接下来的这句语句:

- - 1. myservo.attach(9);

这里就开始调用 Servo 库中的函数了,和我们以前函数调用有点区别。这里,我们需要先指明这是哪个库中的函数。所以,先指出对象名,再指出函数名。每次要用到储物柜的东西就要先指明这个标签。这样程序才知道要去哪里找东西。

库函数调用格式如下:

#### 对象名.函数名();

不要忘了中间的"."!myservo 是我们前面设的标签(对象),然后调用的函数是:

1. attach (pin);

attach(pin)函数有一个传递参数——pin,任意一个数字引脚(不建议使用数字 0,1)。我们这里选择数字引脚 9。

进入主函数,有两个 for 循环,第一段是从 0 开始,循环到 180,每次增加 1 度。第二个 for 循环则是从 180 开始,每次减小 1 度,一直减到 0。在回到上面那个循环中…… for 循环中又调用了一个 Servo 库中的函数 write(pos),我们可以不用管函数内部复杂的程序,只要先会使用就可以了。

### 1. myservo.write(pos);

和上面那个函数调用一样,先要指明是哪个库。该函数的传递参数就是角度,单位为°。如果还想了解 Servo 库中还有哪些好用的函数的话,可以参看下面的网址,里面会有相关介绍的。

Servo 库: http://arduino.cc/en/reference/servo