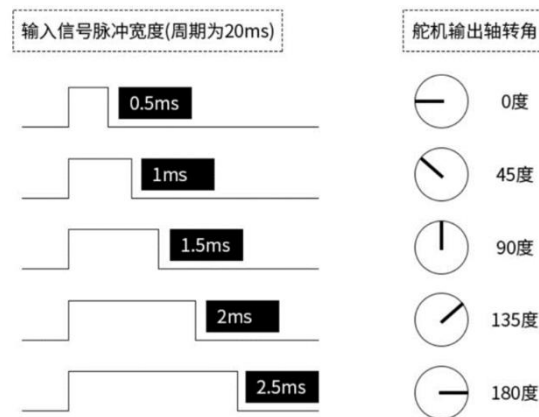
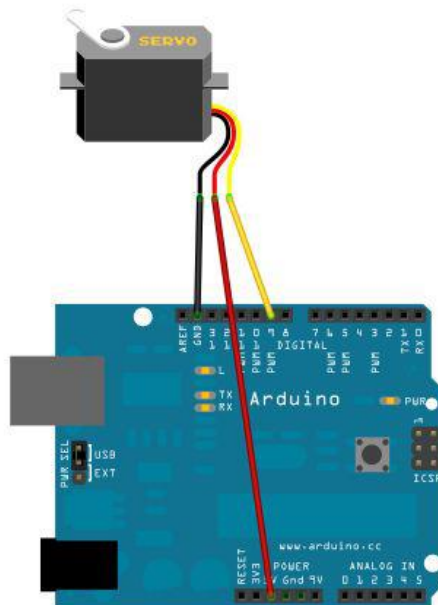


舵机的控制一般需要一个 20ms 左右的时基脉冲，该脉冲的高电平部分一般为 0.5ms~2.5ms 范围内的角度控制脉冲部分。以 180 度舵机为例那么对应的控制关系是这样的：



用 Arduino 控制舵机的方法有两种，一种是通过 Arduino 的普通数字传感器接口产生占空比不同的方波，模拟产生 PWM 信号进行舵机定位，第二种是直接利用 Arduino 自带的 Servo 函数进行舵机的控制，这种控制方法的优点在于程序编写，缺点是只能控制 2 路舵机，因为 Arduino 自带函数只能利用数字 9、10 接口。Arduino 的驱动能力有限，所以当需要控制 1 个以上的舵机时需要外接电源。

如下图所示将舵机接数字 9 接口上。编写一个程序让舵机转动到用户输入数字所对应的角度的位置，并将角度打印显示到屏幕上。



参考源程序如下：

```
//UART send 1~9==>20~180 degree
int servopin=9;//设置舵机驱动脚到数字口 9
int myangle;//定义角度变量
int pulsewidth;//定义脉宽变量
int val;
```

```

void servopulse(int servopin,int myangle)/*定义一个脉冲函数，用来模拟方式产生 PWM 值
*/
{
pulsewidth=(myangle*11)+500;//将角度转化为 500-2480 的脉宽值
digitalWrite(servopin,HIGH);//将舵机接口电平置高
delayMicroseconds(pulsewidth);//延时脉宽值的微秒数
digitalWrite(servopin,LOW);//将舵机接口电平置低
delay(20-pulsewidth/1000); //延时周期内剩余时间
}
void setup()
{
pinMode(servopin,OUTPUT);//设定舵机接口为输出接口
Serial.begin(9600);//设置波特率为 9600
Serial.println("servo=o_serai_simple ready" );
}
void loop()//将 0 到 9 的数转化为 0 到 180 角度，并让 LED 闪烁相应数的次数
{
val=Serial.read();//读取串口收到的数据
if(val>'0'&&val<='9') //判断收到数据值是否符合范围
{
val=val-'0'; //将 ASCII 码转换成数值，例'9'-'0'=0x39-0x30=9
val=val*(180/9);//将数字转化为角度，例 9*（180/9）=180
Serial.print("moving servo to ");
Serial.print(val,DEC);
Serial.println();
for(int i=0;i<=50;i++) //给予舵机足够的时间让它转到指定角度
{
servopulse(servopin,val);//引用脉冲函数
}
}
}
}

```