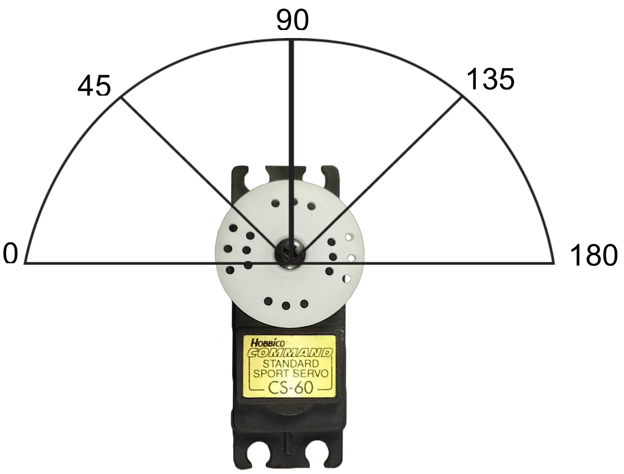
Servos

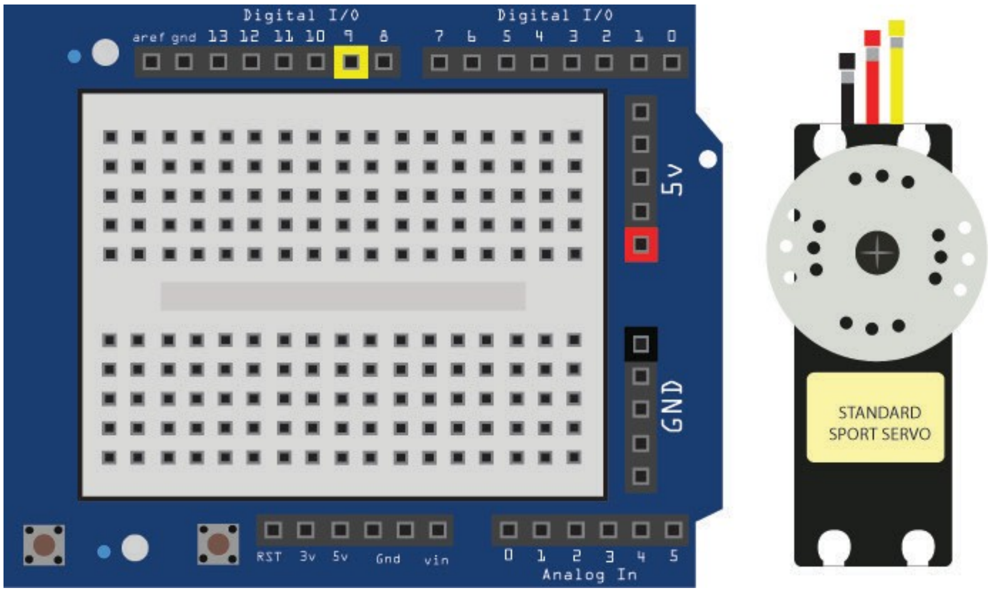
1

Up until now only DC motors have been used in each exercise. DC motors are designed to spin around for as long as they are powered on. They are not limited by amount of rotation, which makes them great for using with continuous mechanisms.

Another type of motor is a servo, which have the advantage of being able to move to a particular position by command. This servo can move between 0 and 180 degrees.



Follow these steps to wire your servo to the Arduino



Wire the black wire from the servo to GND, the red wire into 5v and the yellow wire into Digital pin 9.

Servos

2

Open a new file by going to **File** and **New**, then go to **File** again and **Save As**, typing whatever you like for the name.

Update your code to match what is shown on the right.

```
#include <Servo.h>
Servo servo;
void setup()
{
  servo.attach(9);
}

void loop() {}
```

The first thing you need is `#include <Servo.h>`, this is a library made for servos that will allow you to easily drive any servo. Next add `Servo servo;` to setup a servo object.

In `void setup()` add `servo.attach(9);` this tells the servo it will be controlled by digital pin 9.

In the `void loop()` function the calls to `servo.write()` will tell the servo what position you want it to turn to. **Not** how many degrees to turn.

The program starts by telling the servo to turn to the 0 degree position followed by the 90 degree position and finally the 180 degree position. During this cycle the servo will be turning in the clockwise direction.

When the loop starts over, sending the servo to position 0, it will spin counter clockwise until the servo reaches 0.

Note: A delay is used in between servo commands to give the motor time to move between positions. A minimum delay of 1000 is sufficient.

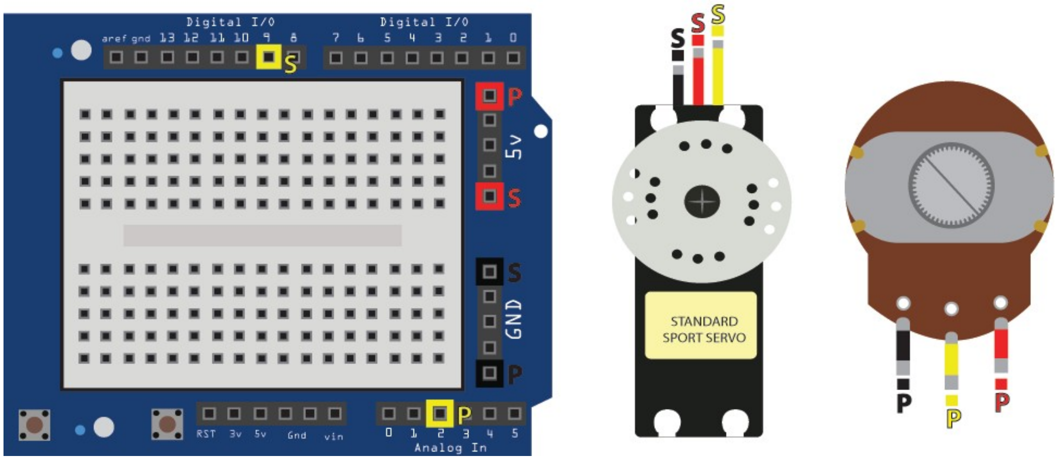
```
#include <Servo.h>
Servo servo;
void setup()
{
  servo.attach(9);
}

void loop()
{
  servo.write(0);
  delay(4000);
  servo.write(90);
  delay(4000);
  servo.write(180);
  delay(4000);

}
```

Press the verify button ✓ to compile your code. Plug the USB cable into the Arduino, then press the upload button → . When the IDE is finished uploading the program unplug the USB cable from the Arduino and continue.

While the program is running the servo motor should move between 3 positions.

3

One way to control a servo physically is with a potentiometer.
(colloquially called a "knob")



Wiring the potentiometer (pot) :
Take the black wire and plug into GND
Take the yellow wire and plug into Analog pin 2
Take the red wire and plug into 5v.

Wiring the servo :
Take the black wire and plug into GND
Take the yellow wire and plug into digital pin 9
Take the red wire and plug into 5v.

Update the previous program to match what is shown on the right.

The Artbotics library will be added so we can use the sensor object.

We will call the Sensor object pot for potentiometer and make sure to keep the Servo object from the previous exercise.

```
#include <Artbotics.h>
#include <Servo.h>
Servo servo;
Sensor pot(A2);
void setup()
{
   servo.attach(9);
}
void loop()
{
  servo.write(map(pot.getReading(), 0, 1023, 0, 180));
}
```

What is being passed to `servo.write` is the reading from the pot. The pot will give a value between 0 and 1023, map will return a degree between 0 and 180.

Press the verify button ☑ to compile your code. Plug the USB cable into the Arduino, then press the upload button ➔. When the IDE is finished uploading the program unplug the USB cable from the Arduino and continue.