Learning Objectives

- Be able to identify characteristics that distinguish a servo and a DC motor
- Be able to describe the difference a conventional servo and a continuous rotation servo
- Be able to use the Arduino Servo library to control servo position

References

Information on Arduino Servo library:
- http://www.arduino.cc/playground/Learning/SingleServoExample

Additional descriptions of servos:
- http://makeprojects.com/Wiki/Servos
- http://www.seattlerobotics.org/guide/servos.html
What is a servo?

A servo-motor is an actuator with a built-in feedback mechanism that responds to a control signal by moving to and holding a position, or by moving at a continuous speed.

DC Motors and Servos

DC Motor
- Motion is continuous
- Speed controlled by applied voltage

Servo
- Capable of holding a position
- Speed controlled by delay between position updates
- Hybrid of motor, gears and controller.

Conventional and Continuous Rotation

Two types of servos
- Continuous rotation
  can rotate all the way around in either direction
- Standard
  can only rotate 180 degrees

- Pulse tells servo which way to spin & how fast to spin
- Pulse tells servo which position to hold
Control signal is a pulse train

<table>
<thead>
<tr>
<th>20 ms constant pulse timing</th>
<th>Pulse frequency is fixed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ms = pulse width</td>
<td>Typical: 20 ms</td>
</tr>
<tr>
<td>1 ms</td>
<td>Pulse width determines position</td>
</tr>
<tr>
<td>Typical: 1ms to 2ms</td>
<td></td>
</tr>
</tbody>
</table>

Servo components
1. Small DC motor
2. Gearbox with small plastic gears to reduce the RPM and increase output torque
3. Special electronics to interpret a pulse signal and deliver power to the motor

Servo from the Sparkfun kit
The micro servo from the Sparkfun Inventor's kit is a conventional servo, i.e. the control signal results in moving the shaft to an angular position.
Arduino Servo library handles the details

- Must connect servos on pin 9 or pin 10
- From the Arduino web site:
  
  "...use of the library disables analogWrite() (PWM) functionality on pins 9 and 10, whether or not there is a Servo on those pins"


---

Arduino Servo library handles the details

- Three components of the Servo Library
  - Create the servo object
    
    ```
    Servo my_servo_object;
    ```

  - Attach the object
    ```
    my_servo_object.attach(servo_pin);
    ```

  - Send control signal
    ```
    my_servo_object.write(pos);
    ```

  Name of the object is like a variable name.

  `attach` and `write` are pre-defined methods that act on the servo object.

---

Modified version of the sweep function

```c
// File: sweep_variable_wait

// Modified version of Sweep by BARRAGAN (<http://barraganstudio.com>)
// Use variable dtwait to make the speed of sweep apparent

#include <Servo.h>   // Make code in Servo.h available to this sketch

Servo myservo;       // Create servo object called "myservo"
int servo_pin = 9;     // The servo must be attached to pin 9 or pin 10
int dtwait = 15;        // Duration of wait at the end of each step

void setup()
{
    myservo.attach(servo_pin);   // attaches the servo pin to myservo object
}

void loop()
{
    int pos = 0;          // variable to store the servo position
    int direct = 1;       // duration of wait at the end of each step
    for(pos = 0; pos < 180; pos += 1) {
        myservo.write(pos);               // Move to position in variable 'pos'
        delay(dtwait);                    // wait dtwait for the servo to reach the position
    }
    for(pos = 180; pos >= 1; pos -= 1) {
        myservo.write(pos);              // Move to position in variable 'pos'
        delay(dtwait);                   // wait dtwait for the servo to reach the position
    }
}
```
Experiment

- What happens when you adjust dtwait?
- Can adjust the sweep angle?
  - Make new variable to define end angle of the loop
- Open the Knob demo from the Arduino IDE
  - Connect a potentiometer to an analog input
  - Use the potentiometer to control the servo position