Output 3: Servo Motors

Making motions with servo motors
Servo Motor
Servo Motor

0 degree

180 degrees
Servo Motor

1. Gears
2. Potentiometer
3. Motor
4. Electronics
Thursday Week 7: Servo Motors

Theory and Practice of Tangible User Interfaces

http://www.hobby-lobby.com/vdiamond.htm
Parallel Motion Linkage

Mechanism Inside a Toolbox

http://www.technologystudent.com/cams/link1.htm
Thursday Week 7: Servo Motors

Theory and Practice of Tangible User Interfaces

4~6g

37.2g (1.3oz)
Thursday Week 7: Servo Motors

Theory and Practice of Tangible User Interfaces

0 degrees

45 degrees

180 degrees

high

low

1000 microseconds

1250 microseconds

2000 microseconds
In Class Exercise

1. Connect the servo to Arduino
2. Control the servo via serial communication
3. Control the servo with a pot
4. Make a crawler!
In Class Exercise

1. Connect the servo to Arduino
2. Control the servo via serial communication
3. Control the servo with a pot
4. Make a crawler!
```cpp
#include <Servo.h>

Servo myServo;

void setup() {
    Serial.begin(9600);
    myServo.attach(9);
}

void loop() {
    char c = Serial.read(); // read the serial port
    if (c >= '1' && c <= '5') {
        int val = c - '0'; // convert val from character variable to number variable
        int pulseWidth = (val * (maxPulse - minPulse) / 6) + minPulse; // convert val to microseconds
        Serial.print("Moving servo to position ");
        Serial.println(pulseWidth, DEC);
        updateServo(); // update servo position
    }
}

// called every loop().
// uses global variables servoPin, pulsewidth, lastPulse, & refreshTime
void updateServo() {
    // pulse the servo again if the refresh time (20 ms) have passed:
    if (millis() - lastPulse >= refreshTime) {
        digitalWrite(servoPin, HIGH); // Turn the motor on
        delayMicroseconds(pulseWidth); // Length of the pulse sets the motor position
        digitalWrite(servoPin, LOW); // Turn the motor off
        lastPulse = millis(); // save the time of the last pulse
    }
}
```
In Class Exercise

1. Connect the servo to Arduino
2. Control the servo via serial communication
3. Control the servo with a pot
4. Make a crawler!
In Class Exercise

1. Connect the servo to Arduino
2. Control the servo via serial communication
3. Control the servo with a pot
4. Make a crawler!
Thursday Week 7: Servo Motors

Theory and Practice of Tangible User Interfaces
Homework

Post descriptions and photo(s) of your crawler on the course website.

Once you get your crawler to move forward, perhaps you would want to generate movement from your program and use your potentiometer to control the speed of the movement. You may also team up with a friend and use two servos instead of one.
Thanks!