

# week 07



## Output 3: Servo Motors

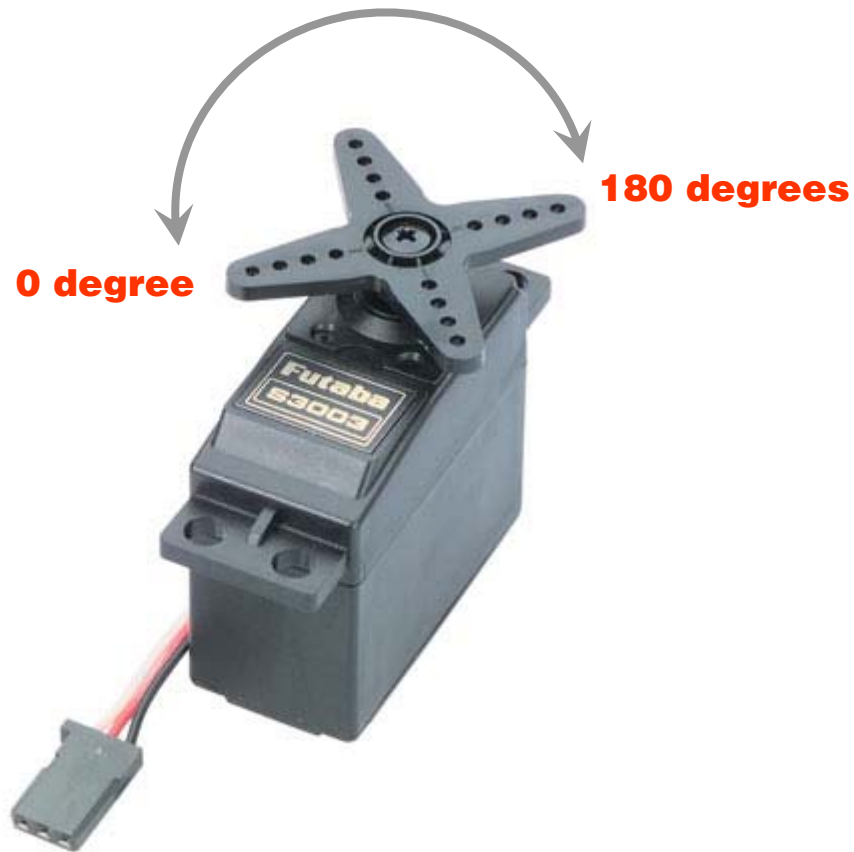
---

Making motions with servo motors

# Servo Motor



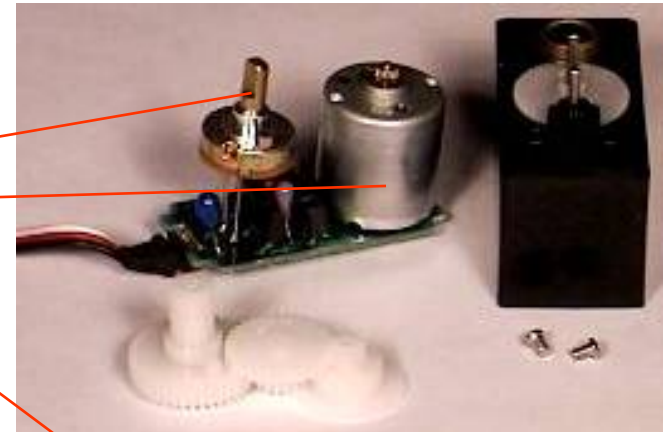
# Servo Motor

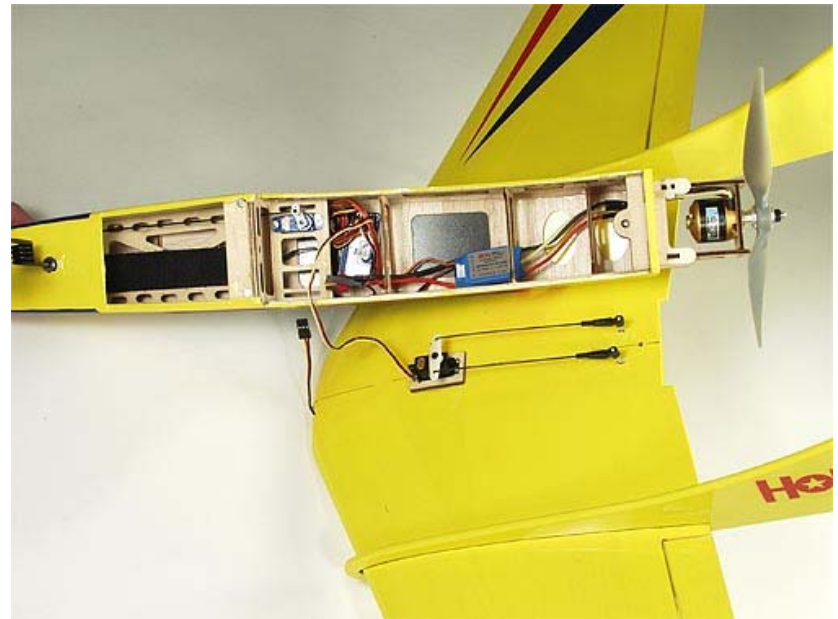
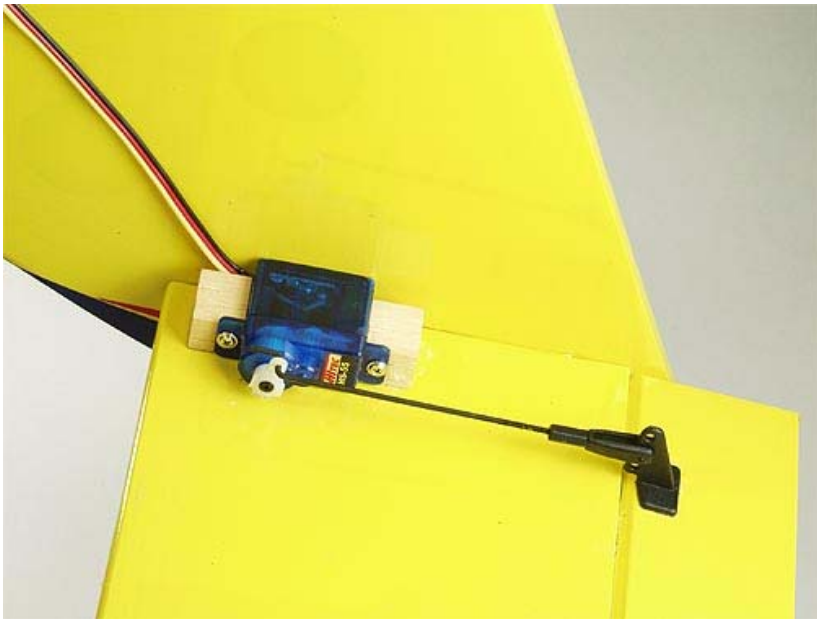
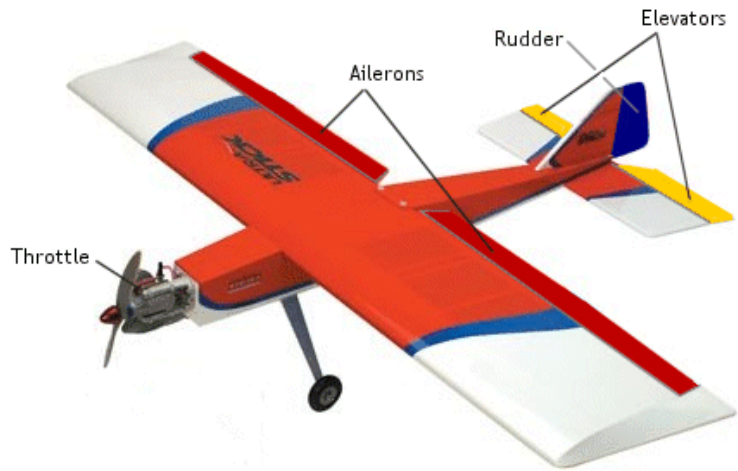


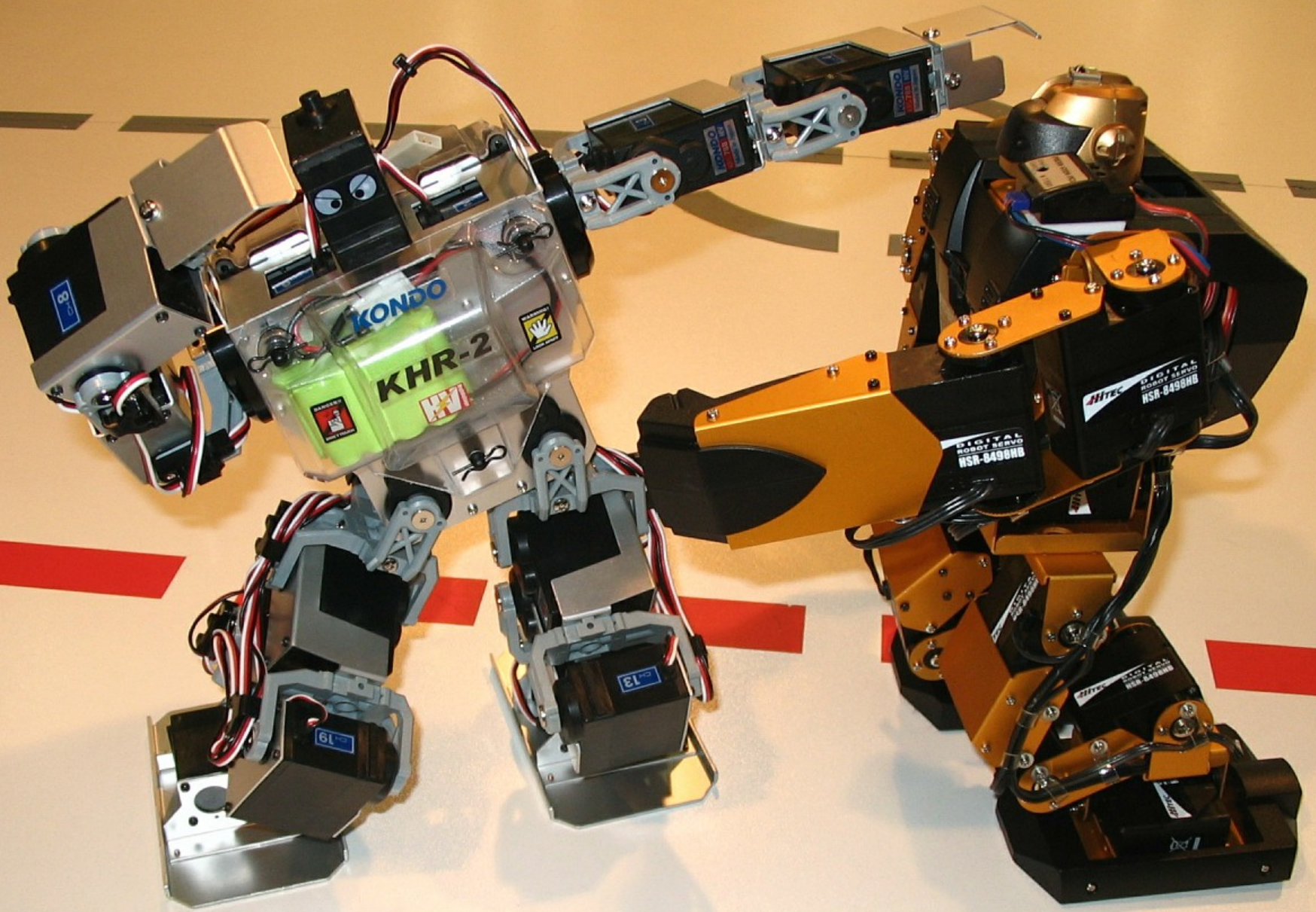
# Servo Motor

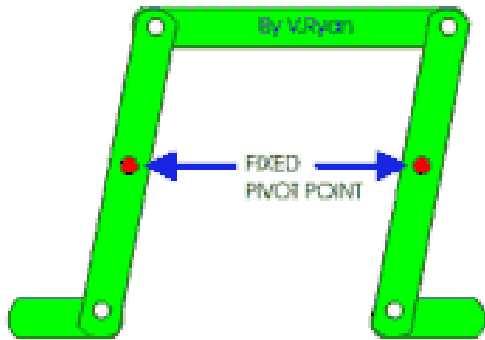


- 1. Gears
- 2. Potentiometer
- 3. Motor
- 4. Electronics









**Parallel Motion Linkage**



**Mechanism Inside a Toolbox**

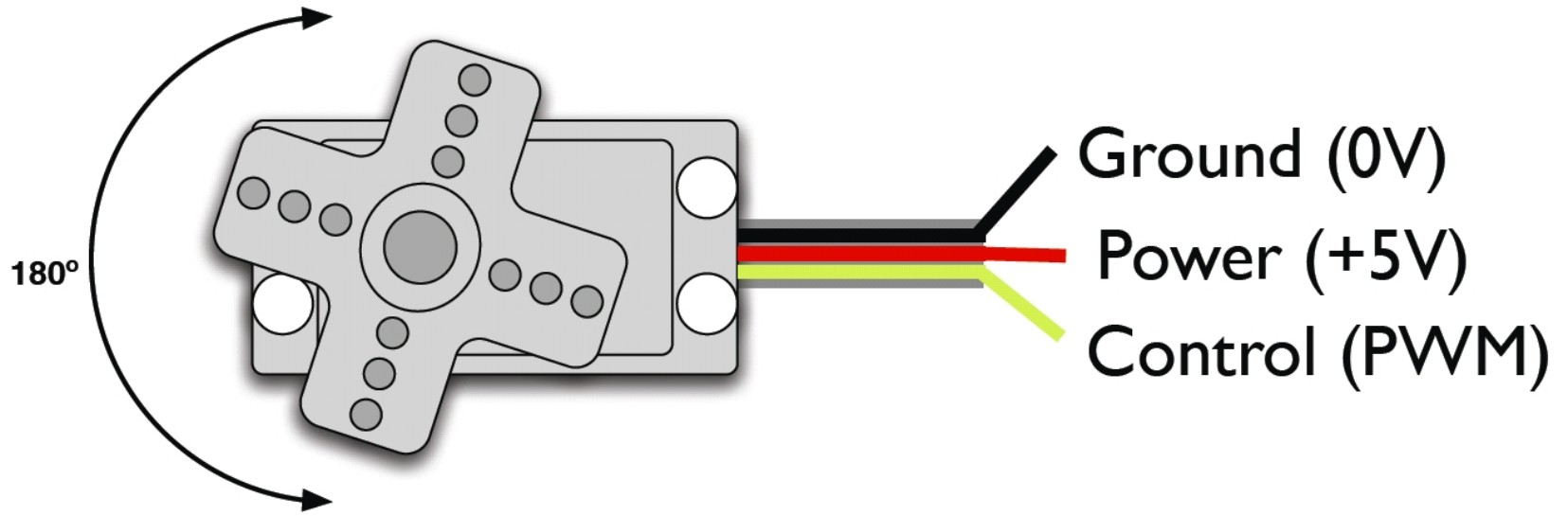


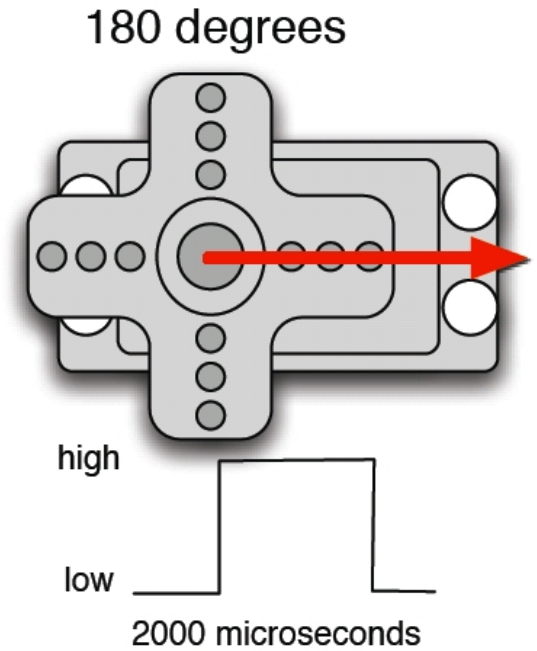
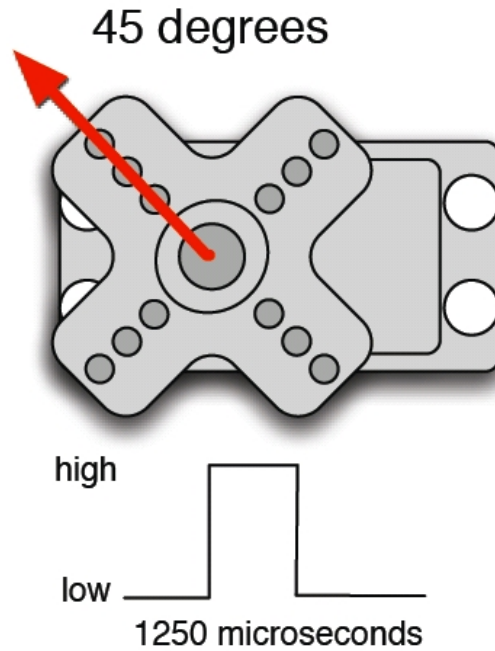
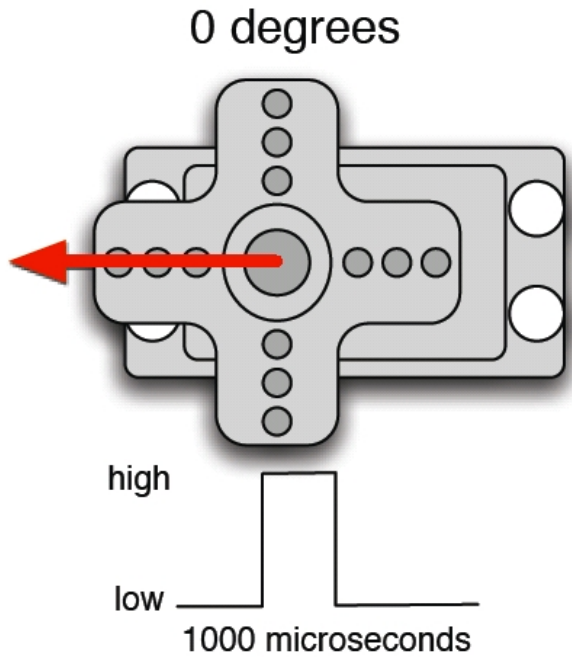
**4~6g**



**37.2g (1.3oz)**

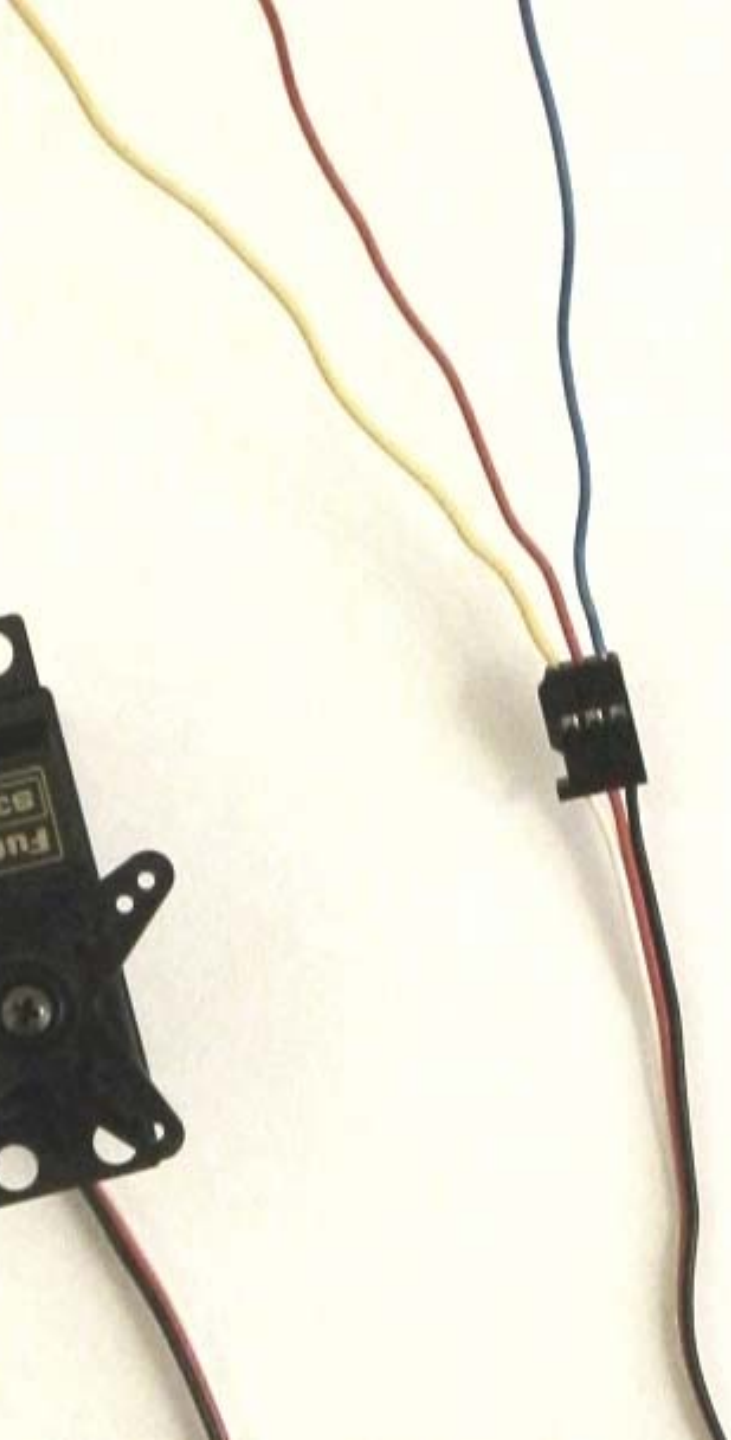


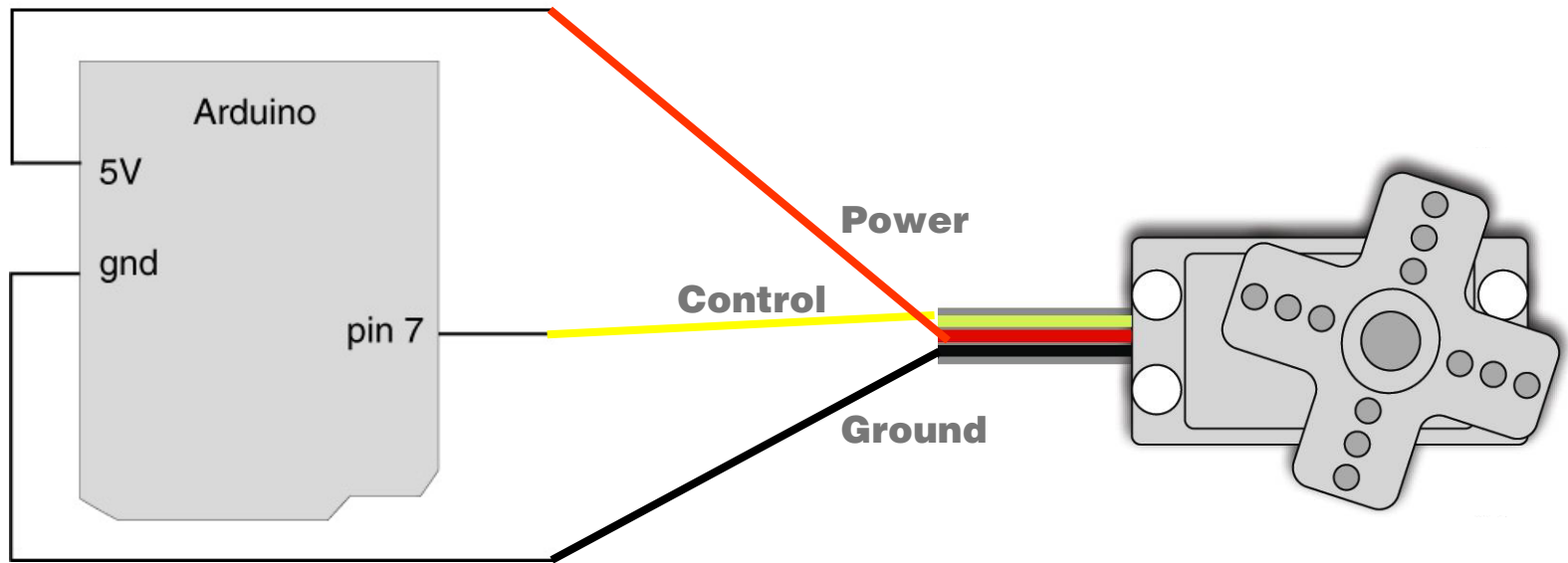


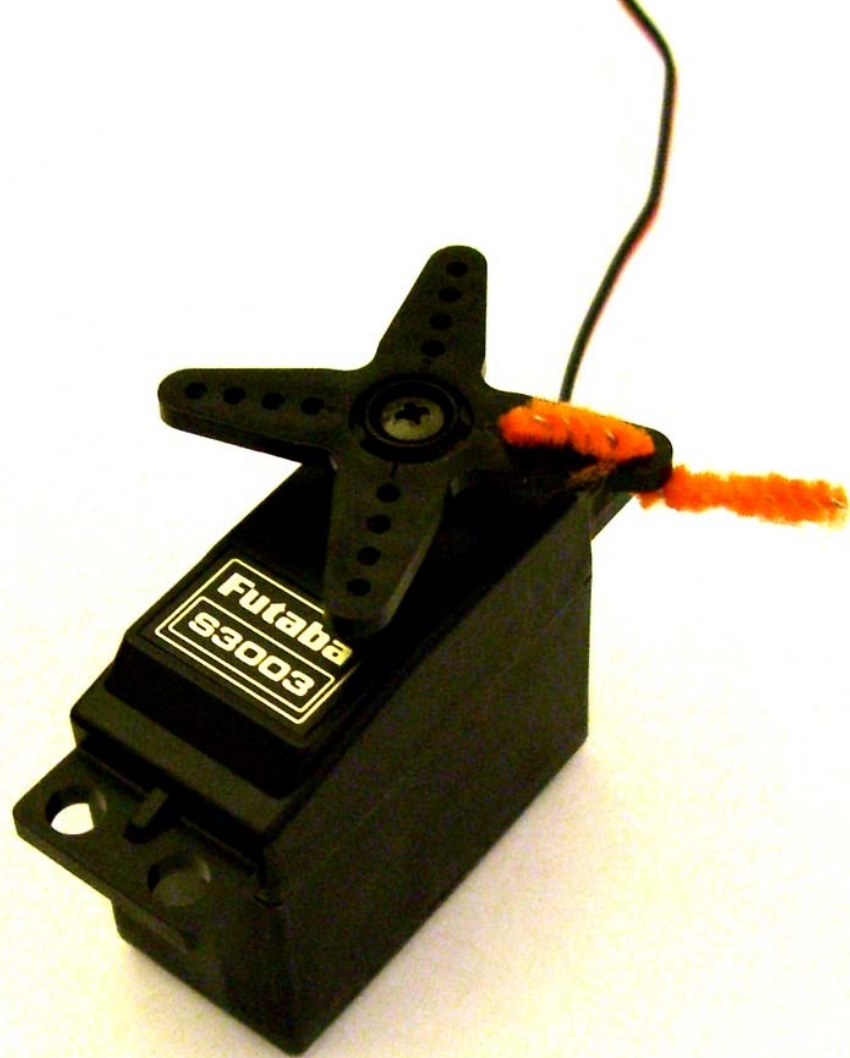


## **In Class Exercise**

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

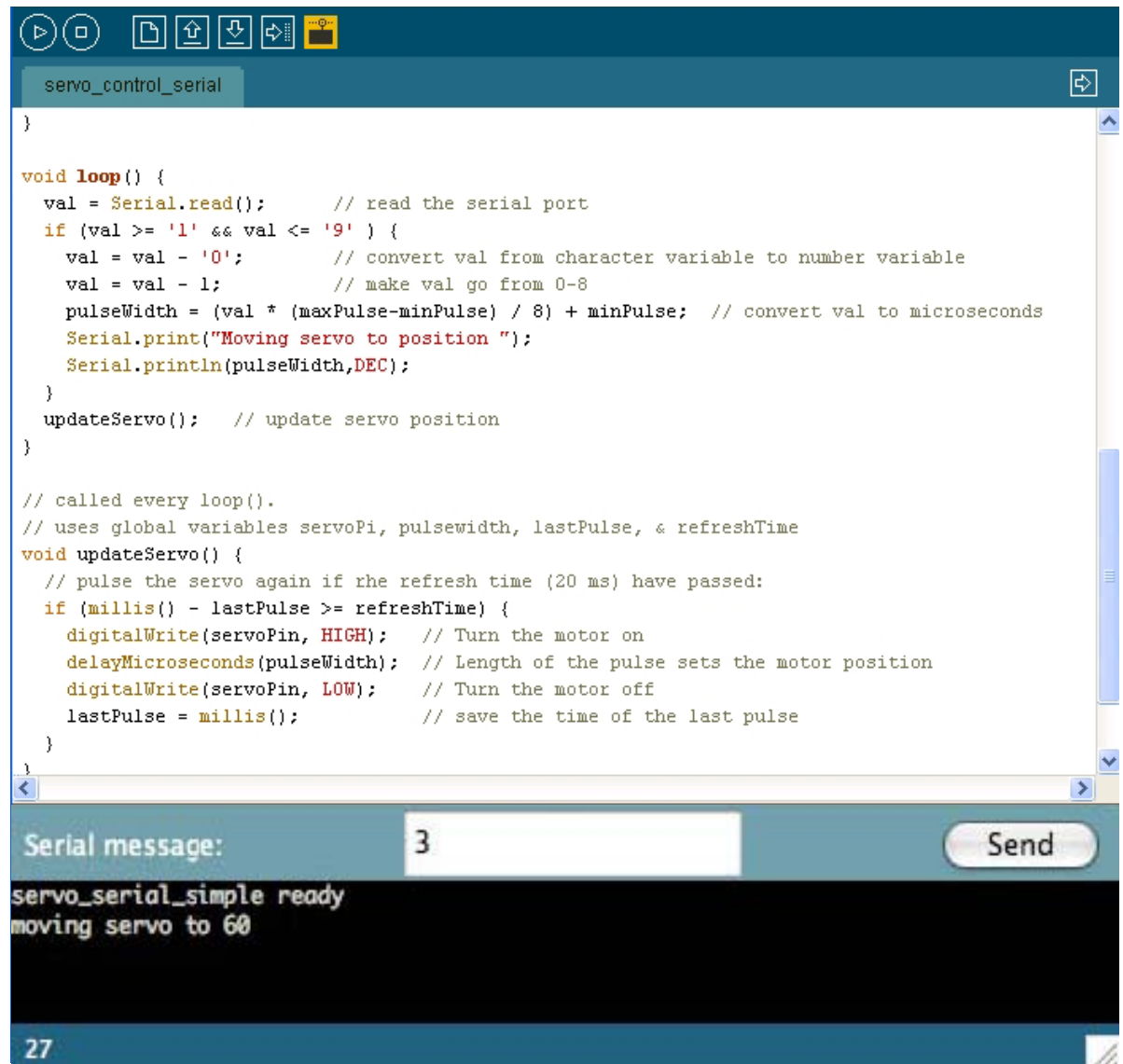






## In Class Exercise

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



```
servo_control_serial

}

void loop() {
  val = Serial.read();    // read the serial port
  if (val >= '1' && val <= '9' ) {
    val = val - '0';      // convert val from character variable to number variable
    val = val - 1;        // make val go from 0-8
    pulseWidth = (val * (maxPulse-minPulse) / 8) + 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 servoPi, 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
  }
}
}

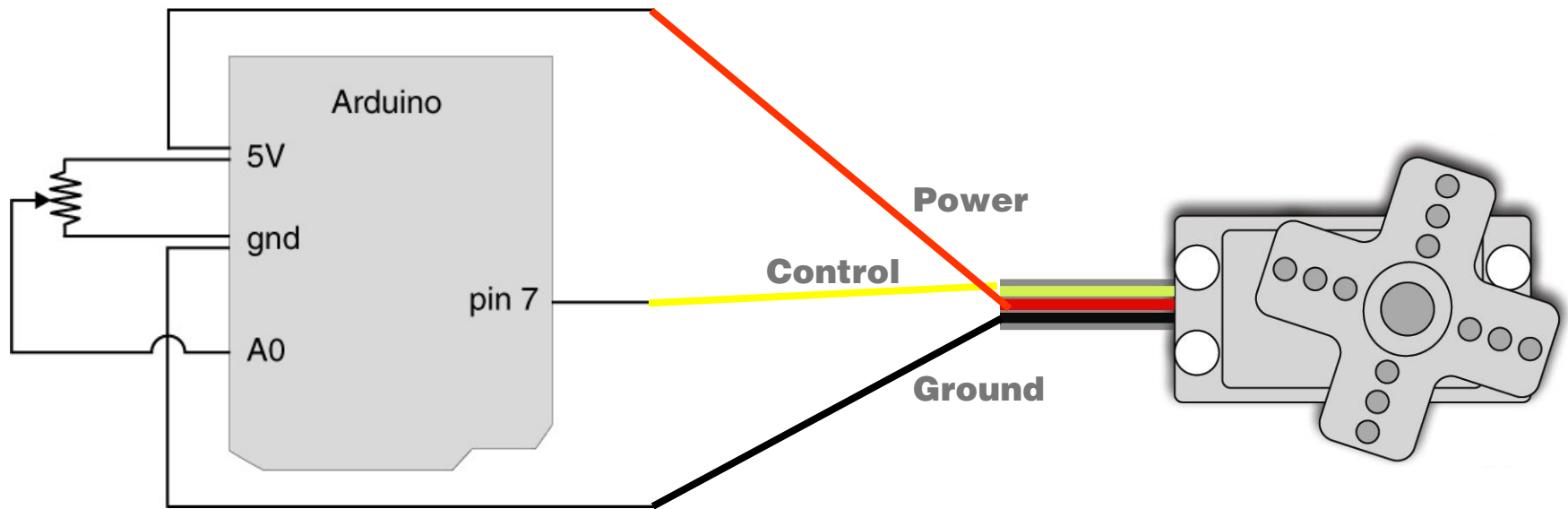
Serial message: 3 Send
servo_serial_simple ready
moving servo to 60

27
```



## In Class Exercise

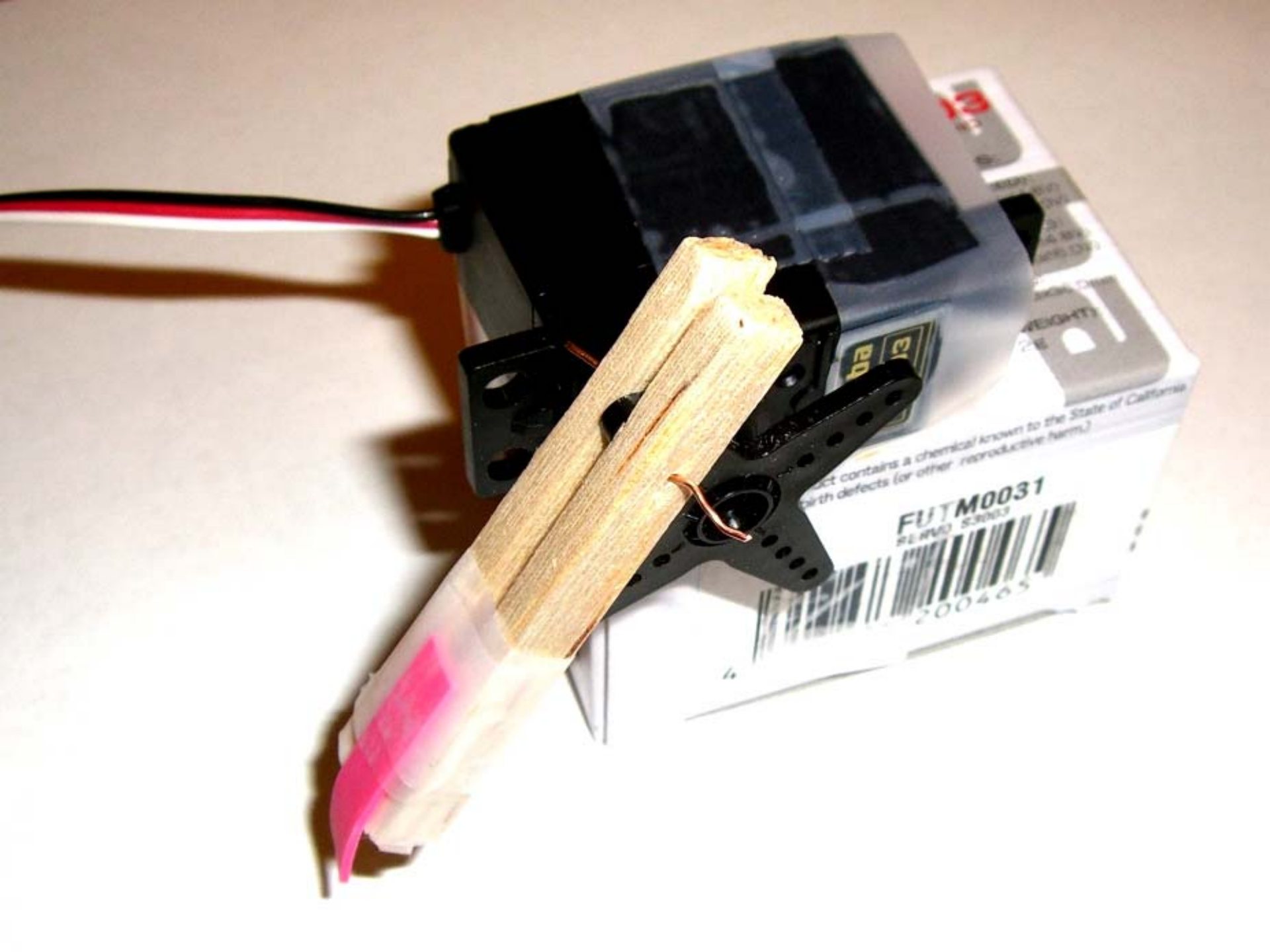
- 1. Connect the servo to Arduino**
- 2. Control the servo via serial communication**
- 3. Control the servo with a pot**
- 4. Make a crawler!**



## **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!**





Product contains a chemical known to the State of California  
to cause birth defects (or other reproductive harm.)

**FUTM0031**  
REWORK S-3003



4



# 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!