In this lesson, you will learn how to control two servos connected to an Arduino using Python code running on a Raspberry Pi. The control connection between the RPi and Arduino is through the USB serial port. It can be challenging to develop a control system in this manner, but perhaps worth the effort. The code can be expanded to control more than two devices.

To set up the lesson, you will need to attach the yellow signal line of your "left" servo to pin 10 on the Arduino and the signal line of your "right" servo to pin 9. The power lines for the servos must be connected to a separate battery supply (about 5.2 volts, 4 rechargeable AA batteries). You will need to connect the ground of the RPi to the ground of the Arduino.

First, we will start with the Arduino code, which represents the largest part of the coding. The Arduino receives data in the form of ASCII character codes. It is necessary for you to understand this character encoding in order to understand the code below. You can find tables of ASCII code on the Internet, but I will provide some information here. The ASCII code for the character zero is the value 48. That is, the software interprets a value of 48 to be equivalent to the character of zero. Here are some ASCII code values and their character equivalents: 48 = 0, 49 = 1, 50 = 2, 51 = 3, 52 = 4, 53 = 5, 54 = 6, 55 = 7, 56 = 8, 57 = 9, 108 = l (a lower case L), 114 = r, and 10 = \n (the new line character). These are the ASCII codes you need to know for this lesson.

Rather than make this a long-winded lesson, I am going to stop my explanation here. I have added many comments (//) in the code. It is likely you will have other questions even after reading the comments. In that case, ask me.

Now open up the Arduino IDE on your Raspberry Pi and enter the code below. Then verify and save it. When it verifies, then upload it to your Arduino.

```
#include <Servo.h>              // include Servo library


Servo myservoRight;           // create servo object to control right motor
Servo myservoLeft;            // create servo object to control left motor

int value;                    // variable to temporarily store data sent by RPi
int varCode;                  // r = right motor l = left motor
int getRight = 0;             // 1  = yes, apply value to right motor
int getLeft = 0;              // 1 = yes, apply value to left motor
int loopLap = 0;              // loop control variable
```

```
void setup() {
  myservoRight.attach(9);            // attach right servo to pin 9 of Arduino
  myservoLeft.attach(10);            // attach left servo to pin 10 of Arduino
  Serial.begin(9600);                // start serial communications with RPi at 9600 bits per second
}

void loop() {
  while (Serial.available()) {      // while there is serial data available
   if (loopLap == 0){               // the first character in serial data packet will be code for data identity, the letter r for
                                    // right motor, letter l for left motor, this is read only once, first time through loop
     varCode = Serial.read();       // read first character and assign to variable varCode
   }
   if (varCode == 114 && loopLap == 0) {        // the letter r has the ASCII code value of 114
     getRight = 1;                              // sets loop to get value for right motor
     loopLap = 1;                            // now second and subsequent passes through loop will skip varCode assignment
   }

   else if (varCode == 108 && loopLap == 0) {            // the letter l has the ASCII code value of 108
     getLeft = 1;                                        // sets loop to get value for left motor
     loopLap = 1;                           // now second and subsequent passes through loop will skip varCode assignment
   }

   if (getRight == 1) {                           // here we collect the number value for pulse length of right motor
     char ch = Serial.read();                     // we already read first character and assigned it to varCode,
                                                  // the second and subsequent characters of the serial buffer are
                                                  // read here and accumulate in the variable named value
     if (ch >= '0' && ch <= '9') {           // if character is not between 0 and 9, it is not a number and we will just ignore it
         value = (value * 10) + (ch - '0');  // What we are doing is subtracting the ASCII value of zero (which is 48), from
                                             // the ASCII character value.  In essence we are converting the string number
                                             // character to an  integer. For example, if character was a zero, its  ASCII value
                                             // would be 48 and the expression (ch –'0') would evaluate to  the number zero.
    }
```

```
     else if (ch == 10) {      // the ASCII value for line return (\n} is 10. When this character is read, then we have reached the
                               // end of the data packet (we add the  \n in the Python program when sending the motor command).
         setServoRight(value);                    // here we apply the speed value to the motor
         Serial.write("right motor value is ");   // we print out speed value to use as check
         Serial.println(value);
         value = 0;                             // we must reset this to zero so that it can be used for next serial data
         getRight = 0;                          // reset
         loopLap = 0;                           // reset
       }
 }

   else if (getLeft == 1) {                       // here we collect the number value for pulse length of left motor
                                                  // same code below as for the right motor
     char ch = Serial.read();

     if (ch >= '0' && ch <= '9') {
         value = (value * 10) + (ch - '0');
       }

     else if (ch == 10) {
         setServoLeft(value);
         Serial.write("left motor value is ");
         Serial.println(value);
         value = 0;                               // reset for next serial communication
         getLeft = 0;                             // reset
         loopLap = 0;                             // reset
       }
    }
  }                                               // end while (Serial.available()) loop
}                                                 // end void loop ()
```

```
void setServoRight(int pulse) {                       // pulse = pulse width in microseconds
  myservoRight.writeMicroseconds(pulse);              // sets the servo position
  delay(15);                                          // waits for the servo to get there
}

void setServoLeft(int pulse) {                        // pulse = pulse width in microseconds
  myservoLeft.writeMicroseconds(pulse);               // sets the servo position
  delay(15); // waits for the servo to get there
}
```

Now you will enter the Python code that sends motor commands to the Arduino. The program below allows the user to input commands as the Python program runs. Of course this can be altered to another method of control, such as transferring commands from Adafruit IO. In order for the program to work properly, the user must enclose the motor command in quotes. For example, if you wanted the right servo to turn all the way counter clockwise, you would enter this: "r600" and then press Enter. For full clockwise it would be "r2400" and for the midpoint "r1500" . Save the program below and try it out. You should be able to control both servos.

```
import serial

#make sure you have the correct serial port specified, it can vary (ttyACM0 in example)
serialMsg = serial.Serial("/dev/ttyACM0", 9600)

while True:

    #user must input first a letter (r for right motor l for left
    #then a number between 600 and 2400
    #all must be in quotes or an error message will appear and program halts

    position = input('input position, for example "r600" ')
    serialMsg.write(position)
    serialMsg.write('\n')
```