MECH 452 – Mechatronics Engineering

Laboratory #1 – Positioning with a Servomotor and the Stack

This laboratory provides instruction on the position control of a Futaba FP S148 servomotor (**Fig. 1**) with a rotary potentiometer (pot) as input. The lab provides a good illustration of a system that has an input from an analog device (pot) and has an output to a digital device (servomotor). **Fig. 2** shows the components found inside the servomotor: DC motor with a gear train, internal pot to measure the position and an ASIC control circuit to translate a digital command signal from the Arduino into an analog control signal for the DC motor. The setup for the lab is illustrated in **Fig. 3**. The concept of pulse control is shown in **Fig. 4**, where the width of the pulse from the Arduino provides the setpoint command to the servomotor. Source of motor: https://www.robotshop.com/ca/en/hitec-hs-422-servo-motor.html

Fig. 1. Servomotor

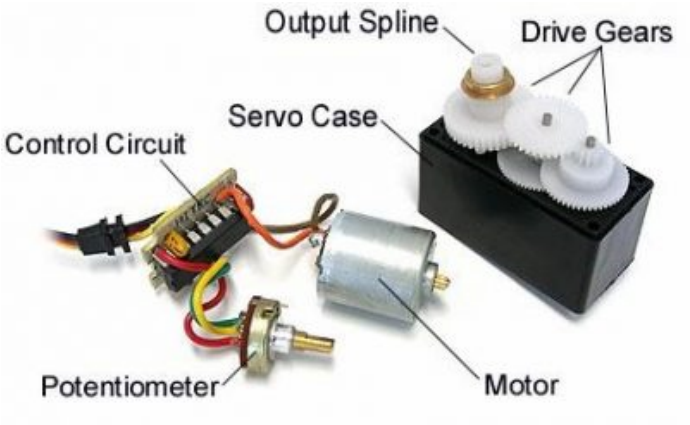

Fig. 2. Servomotor cutaway.

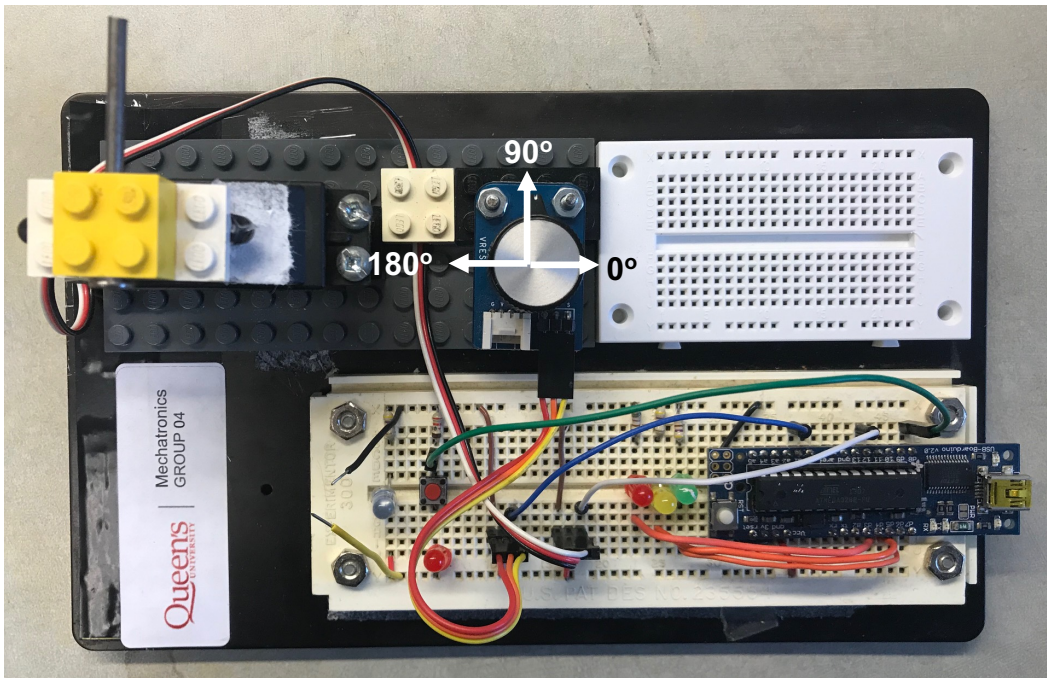

Fig. 3. Stack as setup for **Lab #1**, with both motor arrow and pot knob pointing "up" for 90°, pointing to the left for 180° and to the right for 0°, when looking down on the board.

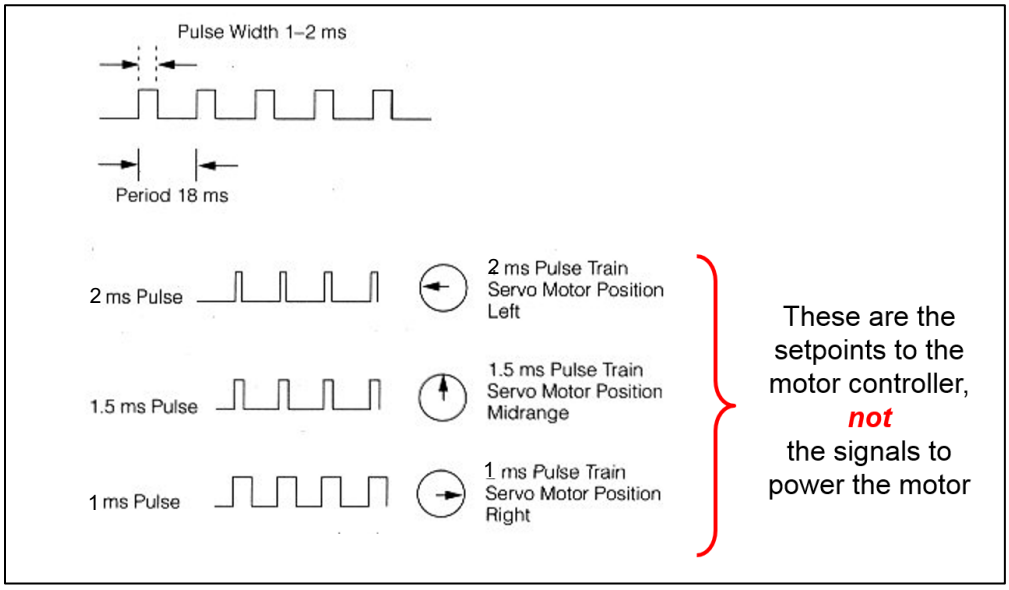

Fig. 4. Servomotor control (https://www.servocity.com/how-does-a-servo-work)

Procedure – Part One (familiarization, potentiometer checked but not used)

1) As shown in **Fig. 3**, place the servomotor on the Stack and connect its black/red/white ribbon wire to the 3 pin connector on the protoboard located nearest to the Arduino. Align black to the ground rail on protoboard, red to the 5 *v* rail and white (signal) connected with a jumper to **Pin d12** (digital)**.** The orientation of your Lego-based "arrow" should be as in **Fig. 3**.

2) Place the pot on the Stack and connect its yellow/orange/red ribbon wire to the remaining 3 pin connector on the protoboard. Align yellow (G on pot) to the ground rail, orange to the 5 *v* rail (V on pot) and red (signal or S on pot) connected with a jumper to **Pin A3** (analog). To confirm that the pot is working properly, use your DMM to measure the pot signal directly. Note that full rotation is 270º (not 180º) and the signal should go from 0 V to 5 V over that range

3) From the course ***onQ*** website, download the program ***M452Lab1ServoMotor***. Open the program in the Arduino editor. As stated in the header, the program is based upon code that was found in Evan's Arduino Programming Notebook (no longer available, but need to acknowledge the source: *http://playground.arduino.cc/uploads/Main/arduino_notebook_v1-1.pdf*).

4) Upload the program to the Arduino and run the program. Note the following sequence of operations:
   a) Green LED flashes ON and OFF once program is loaded
   b) User presses Button to start the program cycle
   c) Yellow LED goes ON, servomotor should cycle from 0º to 180º, and then back to 0º
   d) Red LED goes ON, there is a pause, then program returns to **Step 4a)** with Green LED flashing

   You should observe that the servomotor doesn't line up with the actual 0º and 180º limits as shown in **Fig. 3**. In fact, it probably doesn't even come close. The reason for the discrepancy is that the program assumes that pulse setpoints given in **Fig. 4** are correct. They are not, they are only rough approximations. Each servomotor has to be calibrated. Look at the program listing and in particular the function ***servoPulse*** and the command:

$$\textbf{\textit{pulseWidth}} = (\textbf{\textit{myAngle}}*a) + b = (\textbf{\textit{myAngle}}*5) + 1000; \qquad (1)$$

   which means that when ***myAngle*** = 0 then ***pulseWidth*** = 1000 *μs* and when ***myAngle*** = 180 then ***pulseWidth*** = 1900 *μs*. This lines up with the logic shown in **Fig. 4**, which has the servomotor fully right (0º) for a ≈1 *ms* pulse and the servomotor fully left (180º) for a ≈2 *ms* pulse.

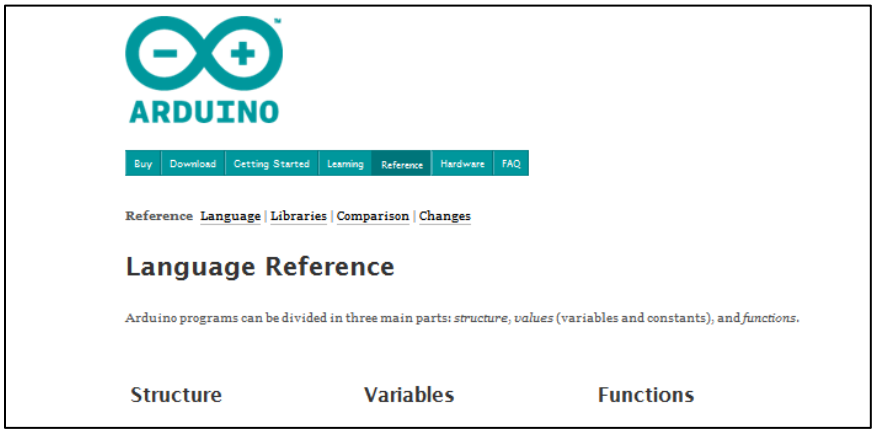

Fig. 5. Homepage for Arduino's on-line language reference guide.

Procedure – Part Two (tuning of the servomotor)

1) Make a copy of the program ***M452Lab1ServoMotor*** and name it ***M452Lab1ServoTuned***. Update the text in the header of the program as you did in **IntroStack**.
2) Close the ***M452Lab1ServoMotor*** sketch and open ***M452Lab1ServoTuned***. Calibrate the program such that the motor sweeps between 0º and 180º, to within ±10º of the positions shown in **Fig. 3.** This will require you to adjust the coefficients *a* and *b* in Equation (1). Recommended procedure is: i) reduce the intercept *b* to get 0º and then ii) increase the slope *a* to get 180º.

Q1. As ***pulseWidth*** and ***myAngle*** are integers, do the coefficients in **Step 2)** have to be treated as integers? Why or why not?

Q2. What software change could you make, that would speed up (or slow down) the sweep speed of the motor? You may wish to check the web for possible answers to this question. Make a change to test your answer.

Procedure – Part Three (modify program such that servomotor tracks potentiometer)

1) Make a copy of the program ***M452Lab1ServoTuned*** and name it ***Group#Lab1ServoMotor***. Update the text in the header of the program as you did in **IntroStack**.
2) Close the sketch for ***M452Lab1ServoTuned*** and open ***Group#Lab1ServoMotor***. Modify the program to use the servomotor code given as **Fig. 6**. Note the following commands:

#include <Servo.h>	loads servo library (routine that rotates servo)
Servo myservo;	defines ***myservo*** as the call to the library
potVal = analogRead(potPin);	reads the pot input, from 0 to 1023 (or 5 V)
myAngle = map(potVal,0,1023,180,0);	converts the pot input to 0º to 180º
myservo.write (myAngle);	output command to ***myservo***

You can check the syntax of these commands by looking at the Arduino on-line reference. Click on **Help** in the top toolbar and select **Reference**. **Fig. 5** should appear. For example, select **map** from the menu of commands to get the syntax of this command. You must also:

- Define any new variables or constants (e.g. ***potPin*** and ***potVal***)
- Remove all commands from **Part Two** that are no longer used (i.e. the function ***servoPulse***)

3) If implemented correctly, the position of the servomotor should continuously track the position of the potentiometer. The required sequence of operations for ***Group#Lab1ServoMotor*** is:
   a) Green LED flashes ON and OFF once program is loaded
   b) User presses Button to start the program cycle

```
Before void setup
    #include <Servo.h>    //servo library call
    Servo myservo;        // create myservo object for library Servo

In void setup
    myservo.attach(servoPin);

In void loop
   potVal = analogRead(potPin);            // analog range assumes 5 v = 1023
   myAngle = map(potVal,0,1023,180,0);
   myservo.write(myAngle);
   delay(15);                              // give servo time to get there
```

Fig. 6. Servomotor code for **Part Three** (from https://www.arduino.cc/en/tutorial/knob with library background at https://www.arduino.cc/en/Reference/Servo).

c) Yellow LED goes ON, servo continuously tracks position of the pot, as you rotate the pot

d) Program runs **Step 3c)** continuously until the Reset button is pressed.

4) Similar to **Part Two**, you need to calibrate the program to ensure that the pot and motor point in the same direction at 0°, 90° and at 180°. Specifically, work with the command:

$$myAngle = map\ (potVal,c,d,180,0) = map\ (potVal,0,1023,180,0) \quad (2)$$

To do this, use a ***Serial.println(potVal)*** command to print the value of ***potVal*** when the pot is at 0° and when it is at 180°, to get initial values for coefficients *c* and *d*. Use these values in the map command and observe whether or not the pot and the motor line up at 0°, 90° and at 180°. You may need to recalibrate the values of *c* and *d* to get within ±10°. This may not be possible for 0°, but should be possible for 90° and 180°.

Q3. If you do have a problem with 0° in **Step 4),** what could be the reason? If you don't have a problem, why do think others may?

Q4. The supplied map command uses reverse mapping (i.e. 0 maps to 180 and 1023 maps to 0). What happens if change your program and you use as the mapping:

$$myAngle = map\ (potVal,0,1023,0,180) \quad (3)$$

What hardware change could you make to correct for this "bug", that is use Equation (3), but have the pot and the motor go in the same direction? Make the hardware change to test your answer. **Hint:** Review the basic operating principal of a potentiometer, as illustrated in **Fig. 7**.

Lab Group Evaluation:

When you are ready to be evaluated, you will be asked to:

a) Demonstrate your program from **Part Two** (to check your alignment)**,** report your values of the coefficients *a* and *b*, and discuss your preliminary answers to ***Q1*** and ***Q2***.

b) Demonstrate your program from **Part Three** (to check your alignment), report your values of the coefficients *c* and *d*, and discuss your preliminary answers to ***Q3*** and ***Q4***.

Do not disassemble the circuit or the setup with the servomotor. You will need it for the next lab.

Report:

Please refer to ***onQ*** for the due date of the report. Submit your report as an Adobe (***.pdf***) file by uploading to ***onQ***. Use ***Group#Lab1Report*** as the filename. The report must include:

a) **Summary** section

b) **Program** section with flowchart for the program ***Group#Lab1ServoMotor***. Remember to provide a short explanation on what the program does. **Fig. 8** can be used as your flowchart.

c) **Results** section, which must include:

i) Calibrated coefficients for Equations (1) and (2), presented in the form of a table.

ii) Answers to the two questions in **Part Two**.

iii) Answers to the two questions in **Part Three**.

d) **Appendix** section with listing of the program ***Group#Lab1ServoMotor***. Remember to highlight the changes that you made to the original ***M452Lab1ServoMotor.***

Use the provided **Word** template and please review the marking rubric for **Lab #1** as posted to ***onQ***, under the **Assessments/Assignments/Lab #1** tab.

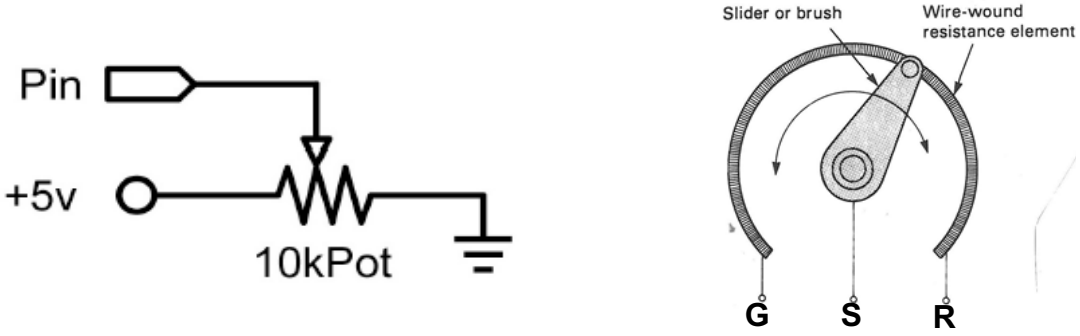

Fig. 7. Potentiometer circuits: a) linear and b) rotary, **G** = ground, **S** = signal and **R** = +5 *v* supply.

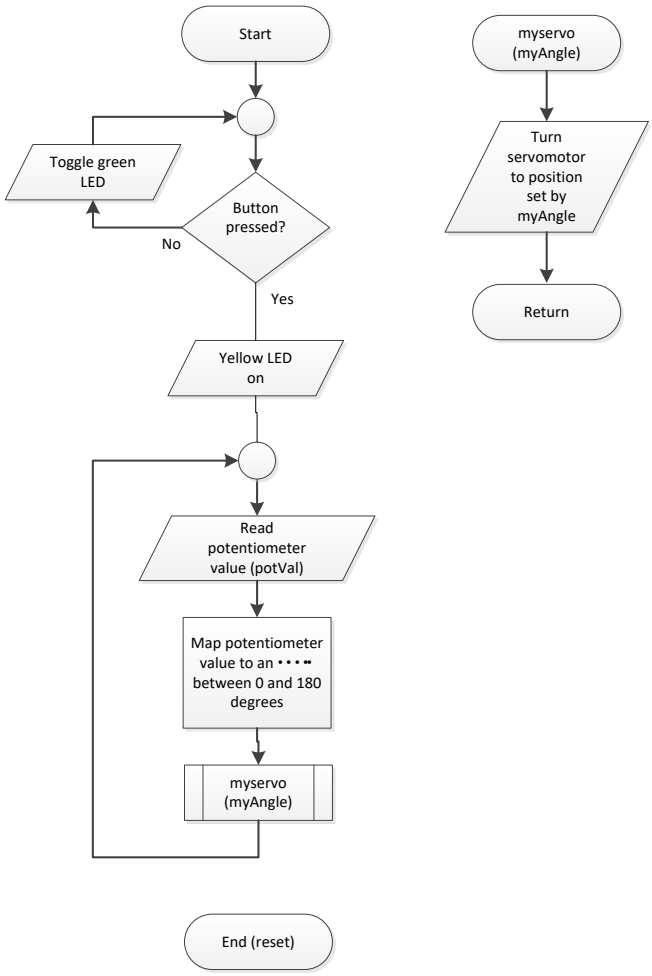

Fig. 8. Flowchart for ***Group#Lab1ServoMotor***