Arduino Servo Motor and Sensor Lab

Experimenting with the HC-SR04 Sensor and a Servo Motor

Leonardo Fusser, 1946995

ABSTRACT

In this lab, we experiment with a sonar sensor and a servo motor with the Arduino UNO.

Experiment Performed on **3 October 2019**
Report Submitted on **14 October 2019**

Department of Computer Engineering Technology
The Profession of Computer Engineering Technology
Andreea Iftimie & Manijeh Khataie

VANIER
CÉGEP/COLLEGE
Learning today Leading tomorrow

Table of Contents

INTRODUCTION

Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices. Its easy-to-use software attracts many individuals (experienced or not experienced with programming) to create fun and exciting projects. In this lab, we will be experimenting with a DC Servo motor and the HC-SR04 Sonar Sensor with the Arduino to create a few tasks.

OBJECTIVES

- Learn some more advanced features with the Arduino.
- Learn about the HC-SR04 sensor.
- Learn about the DC Servo Motor.
- Implement these fundamentals with the upcoming Halloween Project.

MATERIAL USED

- (1x) Arduino UNO.
- (1x) Electronics Breadboard.
- (1x) DC Servo Motor.
- (1x) HC-SR04 Sensor.
- (1x) Wire stripper.
- Some basic electrical wire.
- (2x) Yellow LEDs.
- (1x) 470Ω Resistors.
- (1x) Computer for programming Arduino.

PROCEDURE

[From Lab 3]

- Before getting started, watch the video "You can learn Arduino in 15 minutes".
  https://www.youtube.com/watch?v=nL34zDTPkcs
- Next, an intro and installation to the Arduino IDE software could be found here:
  https://www.arduino.cc/en/Guide/ArduinoUno
- From the link above, click on "Use your Arduino/Genuino UNO on the Arduino Desktop IDE" and follow the instructions to install it onto your computer if it isn't already installed.
- Watch the tutorials by Paul McWhorter to begin programing the Arduino…
  (Lesson 1)
  https://www.youtube.com/watch?v=d8_xXNcGYgo
  (Lesson 2)
  https://www.youtube.com/watch?v=uHUSsSlZa24
  (Lesson 3)
  https://www.youtube.com/watch?v=O4JACbIQX_w

(Lesson 4)
https://www.youtube.com/watch?v=ysHY5JUkpUQ

- Based on the lessons from above, wire an LED and connect it with the resistor to the Arduino and define it as your "output" in the Arduino IDE.
- Connect the Arduino to your computer via USB and launch the Arduino application.
- After understanding some basic Arduino skills, try the "Blink" example that is built into the Arduino IDE (Go to File>Examples>Basic>Blink) to load the code.

[From Lab 4]

(Part 1)

- Read the tutorial on the HC-SR04 sensor:
  https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/
- After reading the article, build the circuit that is used in the article with the sensor.
- After building the circuit, take the code from the article and paste it in the Arduino IDE.
- Upload the code to the Arduino.
- Open the "Serial Port Monitor" and view what is being displayed.
- Explain the role these two lines of code:

```
cm = (duration/2) / 29.1;
inches = (duration/2) / 74;
```

(Part 2)

- Add an LED to the circuit, connect it with the resistor to the Arduino and define it as your "output" in the Arduino IDE.
- After the above is completed, create a condition in the Arduino IDE (if, ifelse statement) where if a piece of paper is closer than 10cm, the LED will turn on for two seconds.

(Part 3)

- Read the tutorial on the MG90S Metal Gear Micro Servo Motor:
  https://www.arduino.cc/en/Tutorial/Sweep
- After reading the article, build the circuit that is used in the article with the motor.
- After building the circuit, load the "Sweep" example that is built into the Arduino IDE (Go to File>Examples>Servo>Sweep) to load the code.
- Upload the code to the Arduino.
- Explain your observations.
- Explain the code.

(Part 4)

- Read the tutorial on the "Knob" example:
  https://www.arduino.cc/en/Tutorial/Knob
- After reading the article, build the circuit that is used in the article.

- Connect the Servo motor the same way as you did before in Part 3.
- After building the circuit, load the "Knob" example that is built into the Arduino IDE (Go to File>Examples>Servo>knob) to load the code.
- Modify the line "val = map(val, 0, 1023, 0, 180)" to "val = map(val, 0, 1023, 15, 170)" in order to reduce tension on the motor.
- After the modification to the code is done and the circuit is built, upload the code to the Arduino.
- Try adjusting the position of the Servo Motor with the potentiometer.
- Explain your observations.
- Explain the code.

RESULTS AND DISCUSSION

Results from Part 3 and 4 of the labs.

The "Blink" example:

(observation)

- Depending on how the delay was set in the Arduino IDE, the LED would stay off for that specified amount of time, then turn on again.

(explanation)

- In the "void setup", the LED is defined as the output.
- In the "void loop", the code that is written below is running continuously.
- "digitalWrite(LED_BUILTIN, HIGH);" specifies that "HIGH" will turn the LED on.
- "delay(1000);" specifies that the LED will remain on for 1 second.
- "digitalWrite(LED_BUILTIN, LOW);" specifies that "LOW" will keep the LED off.
- "delay(1000);" specifies that the LED will remain off for 1 second.
- The loop thus never ends.
- (Refer to Reference #1)

The "HC-SR04" experiment:

(observation)

- With the serial monitor console open, you could see the distance displayed between the object and the sensor. The closer the object was to the sensor, the closer the reading was on the console window. After creating the "if statement", the LED would light up when the sensor detected that the object was closer than 10cm.

(explanation)

- "cm = (duration/2) / 29.1;"…
- "inches = (duration/2) / 74;"…
- These two lines of code above take the time and convert into a distance. The sensor reads a "HIGH" who's the duration is the time (in microseconds) from the sending of the ping to the reception of its echo off an object.
- (Refer to Reference #2)

The "Servo Motor" (part 3) experiment:

(observation)

- When the board is powered on, the servo motor moves its position to 180 degrees, then stops for a short amount of time, then goes back to 0 degrees, then stops for a short amount of time then restarts its loop to continue on forever.

(explanation)

- Before the "void setup", a variable is created to store the position of the servo motor (in degrees).
- In the "void setup", the servo is attached to pin 9 to the servo object.
- In the "void loop", the code that is written below is running continuously.
- The first part of the code in the "void loop" consists of taking the position of the servo motor and rotating the motor from the current position to 180 degrees.
- In the second part of the code in the "void loop" is the same as before but instead taking the position of the servo motor and rotating the motor from the current position to 0 degrees.
- (Refer to Reference #3)

The "Servo Motor" (part 4) experiment:

(observation)

- When the board is powered on, the servo motor moves only changes position when the potentiometer dial was moved. When the potentiometer was turned all the way to the left, the motor changed position to 0 degrees. If the potentiometer was turned all the way to the right, the motor changed position to 180 degrees.

(explanation)

- Before the "void setup", a variable is created to assign the analog pin to connect the potentiometer. Also, another variable is created to read the value from the analog pin.
- In the "void setup", the servo is attached to pin 9 to the servo object.
- In the "void loop", the code that is written below is running continuously.
- "val = analogRead(potpin);" reads the value of the potentiometer.
- "val = map(val, 0, 1023, 0, 180);" scale it to use it with the servo motor.
- "myservo.write(val);" sets the servo positioning according to the scaled value.
- "delay(15);" waits for the servo to get there.
- (Refer to Reference #4)

CONCLUSION

- To conclude with this lab, the Arduino platform is not only useful but very powerful too. Even I, someone with little programming experience, is attracted to this platform. Using the Arduino, we programmed a Servo Motor and a Sonar Sensor to behave in certain ways. Ultimately, this experienced will help us develop the code for our Halloween Project.

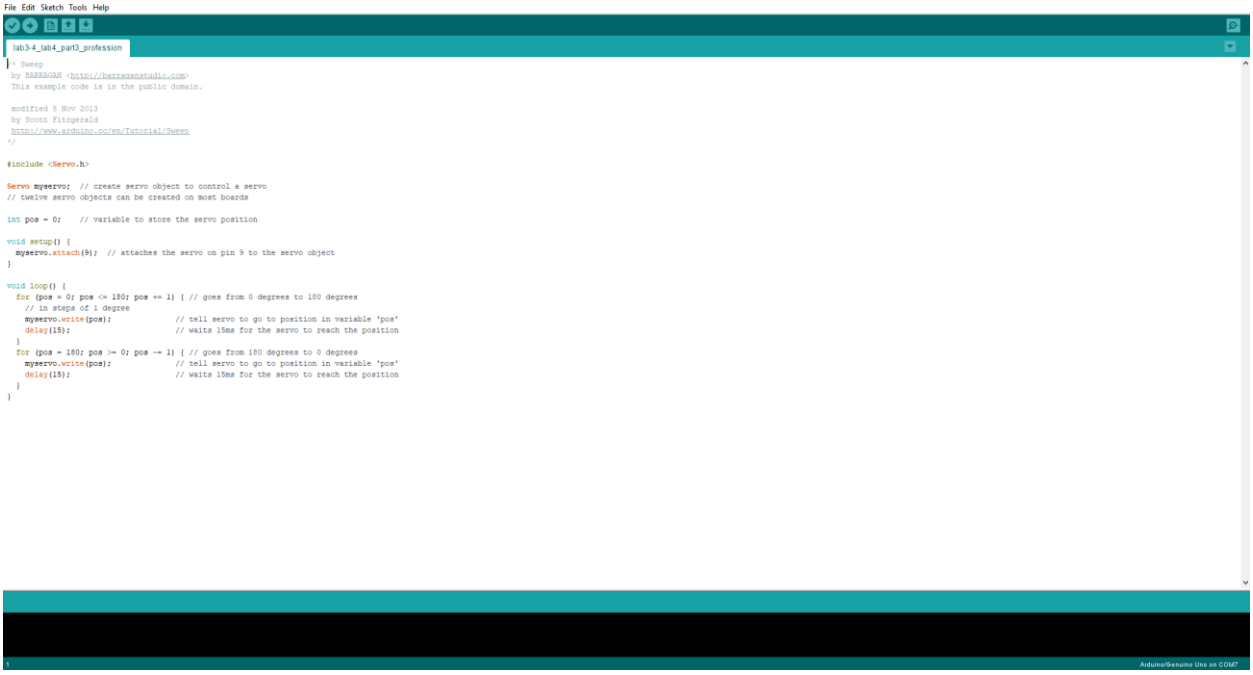
REFERENCES

[1] Reference 1

[2] Reference 2

```
cm = (duration/2) / 29.1;
inches = (duration/2) / 74;
```

[3] Reference 3



[4] Reference 4