

# Physical Computing: Lab 3

By: Daniel Speiser, Maria Shen, Mark Assad

October 21, 2015

# Contents

<b>1</b>	<b>Summary</b>	<b>3</b>
<b>2</b>	<b>System-Level Block Diagram</b>	<b>4</b>
<b>3</b>	<b>Software Description</b>	<b>5</b>
3.1	Summary . . . . .	5
3.2	Servo Motor . . . . .	5
3.3	Sonar Range Finder . . . . .	5
3.4	Source Code . . . . .	5
<b>4</b>	<b>References</b>	<b>7</b>

# 1 Summary

The objective of this lab was to build a radar system using a servo and sonar depth tracker. The servo takes in start, end, and increment angles from the serial port, and the depth tracker reads distances along each rotation.

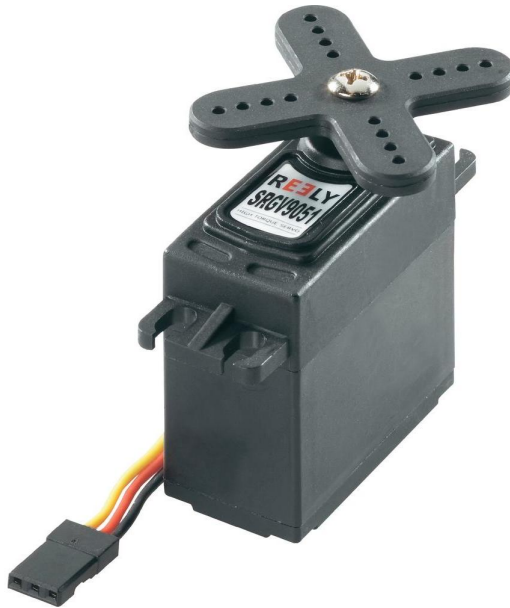


Figure 1: Servo Motor



Figure 2: Sonar Range Finder

## 2 System-Level Block Diagram

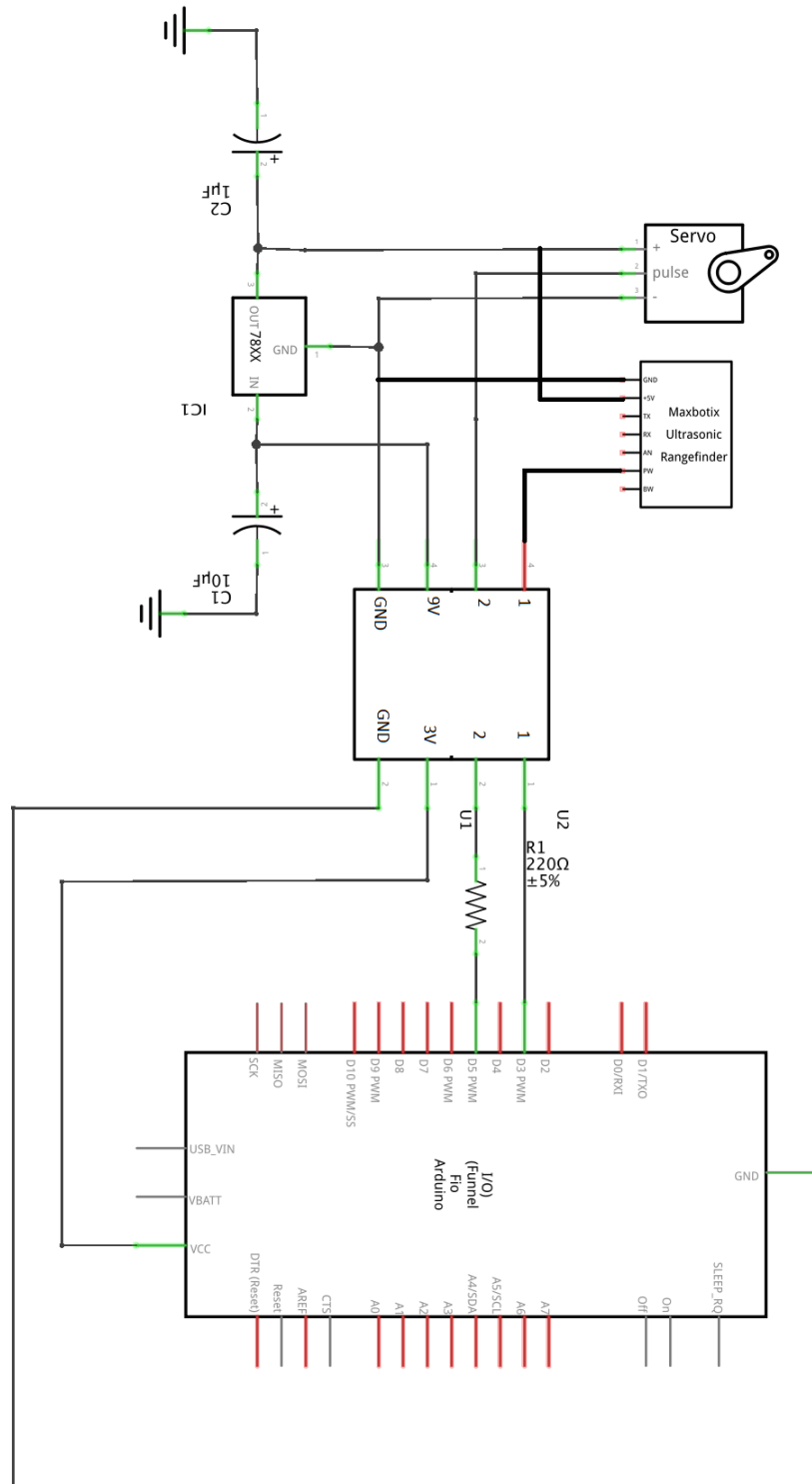


Figure 3: System Level Diagram

## 3 Software Description

### 3.1 Summary

Our software is designed to respond to user input from the serial port in order to instruct the radar system to begin. It does so by rotating the servo along the given set of axis (ranging from 0 - 180 degrees), moving in specified increments, taking depth readings between each position change. The radar continues to sweep along these axis until a user inputs a different set of axis and increment. Once this new input is given, the servo and range finder complete its current sweep, and then begins with the new input.

### 3.2 Servo Motor

For this lab, we are using the built-in servo library from arduino to take input from the user. The servo can rotate from 0 to 180 degrees. Our servo signal is attached to digital pin 5 on the arduino and receives 5V from the power supply (after being converted from 9V through the voltage regulator). The user specifies a starting angle, ending angle, and the increments that the servo should rotate. At every increment, the servo pauses and the arduino reads the input from the sonar. The servo will continue to move in the specified manner until new input is given.

### 3.3 Sonar Range Finder

This sonar has a range of 30cm to 5m. For this lab, our arduino reads the sonar measurements in millimeters using pulse width modulation. This pulse is read by arduino's `PulseIn()` function, which waits for the input to go to HIGH to start timing, then waits for LOW to stop timing. The timing of the pulse corresponds to 1uS per mm. The output of the sonar measurements is displayed in mm. Our sonar is supplied by a 5v power supply.

### 3.4 Source Code

---

```
/*
 * Lab 3
 * By: Daniel Speiser, Maria Shen, Mark Assad
 */
// library imports
#include <Servo.h>    // import servo library
// global variable declarations and initializations
Servo servo;         // create servo object to control a servo
int pwPin = 3;       // declare pw pin 3 for sonar depth tracker
int servoPin = 5;    // declare pin 5 as servo pin
int pos = 0;         // keep track of servo position
int startPos = 0;    // initial start position
int endPos = 0;      // end position
int increment = 0;   // angle degree in which to increment servo
bool isInput = false; // flag for detecting if input has been given
float pulse, mm;     // calculate and store pulse and mm of sonar
// helper methods
```

```

/*
 * The parseInput method reads and parses serial input for
 * servo rotation in the form of: int start, end, increment
 */
void parseInput() {
    isInput = true;           // set flag to true
    startPos = Serial.parseInt(); // parse first int as start pos
    endPos = Serial.parseInt(); // parse second int as end pos
    increment = Serial.parseInt(); // parse third int as angle increment
}

/*
 * The readSensor method pulses the pw pin to high
 * in order to allow the sonar depth tracker to
 * read in and record an object's distance.
 */
void readSensor() {
    pulse = pulseIn(pwPin, HIGH); // set pw pin to high, record pulse
    mm = pulse;                   // mm is equal to pulse width
    delay(100);                   // delay 100ms for servo to reach position
}

/*
 * The printReadings method prints the recorded depth
 * readings to the console. TODO: Implement with GUI
 */
void printReadings() {
    Serial.print(mm);
    Serial.print(",");
    Serial.println(pos);
}

/*
 * the rotate method rotates the servo in the in the given
 * angle range (0-180 degrees), angle increment, and direction
 */
void rotate(int startPos, int endPos, int increment, boolean dir) {
    if (dir) // rotate if direction is forward
        // rotate servo from start position to end position, moving by given increment
        for (pos = startPos; pos <= endPos; pos += increment)
            readSonarAndRotateServo(pos);
    else // rotate backwards
        // rotate servo from end position to start position, moving by given increment
        for (pos = endPos; pos >= startPos; pos -= increment)
            readSonarAndRotateServo(pos);
}

/*
 * reads the sonar and rotates the servo together to accrue
 * distance readings along with the servo's rotation
 */
void readSonarAndRotateServo(int pos) {
    readSensor();           // call read sensor
    printReadings();        // display sensor readings
    servo.write(pos);       // move servo to position pos
    delay(100);             // delay 100ms for servo to reach position
}

// initial setup

```

```

void setup() {
  Serial.begin(9600);    // opens serial port, sets data rate to 9600 bps
  pinMode(pwPin, INPUT); // set pw pin for sonar depth tracker as input
  servo.attach(servoPin); // attach servo to the servo pin declared above
}
// main loop
void loop() {
  if (Serial.available()) // if input is available
    parseInput();         // call parse helper
  else if (isInput == true) { // if there is input, start rotating servo
    // rotate servo from start position to end position, moving by given increment
    rotate(startPos, endPos, increment, true); // call helper to rotate forward
    rotate(startPos, endPos, increment, false); // call helper to rotate backward
  }
}

```

---

## 4 References

- Serial Parsing: <https://www.arduino.cc/en/Reference/ParseInt>
- Servo Rotation: <https://www.arduino.cc/en/Tutorial/Sweep>
- Sonar Reading: <http://playground.arduino.cc/Main/MaxSonar>