Arduino Lab 5

ME 2900

Instructor: Dr. Sandra Metzler

Date of Submission: 4/24/17

Emily Curtiss

I. Introduction

As an Ohio State student, I spend most of my time at home working on assignments and being preoccupied with my studies. To ensure that I am able to complete my tasks, I need to manage my time as efficiently as possible and not waste time doing things that can be dealt with in a more effective manner. One such duty is answering the door when there is a visitor. To save time, a smart lock was developed to determine whether I'll be required to come to the door. This lock should know when there is and isn't a visitor, should address them, determine their reason for visiting, and either send them away or allow them entry into my apartment.

Further details of this project are discussed in the following report.

II. Design Process

By utilizing the design process, I was able to determine the best possible solution to my problem. Initially, my main concern was how to determine whether someone was at the door or not. Several sensors came to mind, such as an Ultrasonic Rangefinder, an IR Distance Sensor, or a Force-Sensitive Resistor/Pressure Sensor. I originally thought the rangefinder or distance sensor would be best, but then I began to wonder if simple things such as an animal passing by or something as small as a leaf blowing past would trigger it. My next thought was to use the Pressure Sensor, and calibrate it so only the weight of a human could trigger it.

Once I determined the solution to triggering the system, my next concern was how I would convey a message to the visitor. The OLED Graphical Display and LCD Display were the two options I narrowed it down to. Since I'd only need to display words, I decided to use the LCD to maintain simplicity. A similar issue was encountered with how to have the visitor respond to the system. I originally thought that using a microphone would be the most efficient. However, it is often windy in Columbus, and I recalled that it is difficult to use a cellphone while it is windy, since the wind disrupts your voice as you speak into the phone. I didn't want this issue with my smart lock, so I decided a physical sensor would be best. A touch screen seemed like the best solution, since options could be displayed behind the screen and it would be simple for a visitor to use.

My final issue was how to convey to the visitor that they could either enter or that they should leave. I determined that four main types of visitor would be at my door. Family, a salesperson, a delivery person, or a Girl Scout selling cookies are the most likely people to be visiting my residence, and each type requires its own response. Family is permitted entry, a salesperson should leave, a delivery person should leave the package by the door, and a Girl Scout should wait for me to answer the door. To accomplish this, an RBG LED was used to display either red, green, or yellow. The red indicated that the visitor should leave, the green indicated that entry was permitted or that I was coming to answer the door, and the yellow was an indicator to leave the item by the door. In addition to this, a servo motor would rotate to either unlock or

lock the door accordingly, and the LCD would display an appropriate message. A sketch of the setup is shown in Figure 1 below.

Figure 1: Prototype Sketch

My design process allowed me to determine what solutions were available for my use, and which of those solutions would be the best to continue with in my project. It was helpful to think of multiple different ways I could complete the project, and it gave me a lot of options to create the best possible final product.

III. Discussion

The smart lock system uses a pressure sensor, LCD, touch screen, RGB LED, and servo motor to determine whether or not a visitor should be permitted entry, relinquishing my responsibility of answering the door.

The system starts out by determining if there is someone at the door. A pressure sensor placed underneath a door mat is constantly taking a reading, and only activates the rest of the system if the weight registered is higher than a predefined tolerance, the weight of a person. Once the system is activated, the LCD lights up with a greeting and asks the guest their reason for visiting. The person will then be presented with four options on the touch screen: family, salesperson, delivery, or Girl Scout with cookies. The visitor will tap the corresponding quarter of the screen, which will then activate the rest of the smart lock.

Family members are permitted immediate entry, so the LED lights up green, the servo motor unlocks the door by moving an arm out of the way, and the LCD lights up with a message welcoming them inside. A salesperson is not permitted entry and will likely waste my time, so the LED lights up red, the servo motor moves the arm in front of the door so it can't be opened, and the LCD displays a message asking them to leave. A delivery person is treated with more patience, but should still not enter the apartment. The LED turns yellow, the servo motor locks the door, but the LCD politely asks them to leave the package by the door and thanks them for their time. A Girl Scout is always welcome, but I don't want to waste their time so I'll come to the door for them. The LED changes to green, the servo unlocks the door, and the LCD politely asks them to wait while I come to the door. No matter how busy I am, I'll always have time for thin mints.

In order to accomplish this, the Arduino code cycles through a series of if and else statements while constantly taking readings from the pressure sensor and the touch screen. As previously stated, once the pressure sensor is triggered, the rest of the system activates and the code begins to cycle through all the possible inputs collected by the sensors. This system is self-sufficient and requires no input from me to operate. The smart lock runs while the pressure sensor is triggered, but is immediately powered down once the visitor leaves. This makes it efficient and accomplishes the goal of allowing me adequate time to complete my work.

IV. Conclusion

The smart lock operated better than expected. A video of it operating can be found at https://www.youtube.com/watch?v=tV9dbON7_h8&feature=youtu.be. I am particularly proud of how the touch screen reads and utilizes data. While the idea is simple in concept, getting the touch screen to work took a lot of effort. I had to determine the x and y coordinates that constrained each quadrant of the screen, and then use that knowledge to activate a loop. The coding for this was particularly difficult, since reading the data taken from the touch screen and setting variables for the x and y values required precise location within the overall code. It took

a lot of trial and error to get it functioning properly, but once it was, it was very satisfying to see the whole system work.

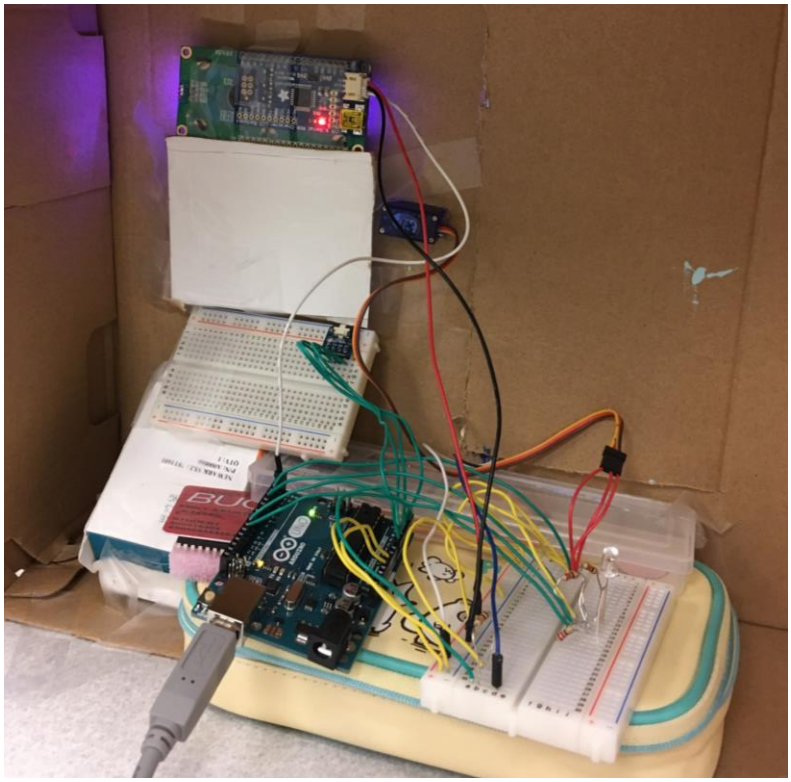
The main difficulty I faced in this project was placing everything in its proper location within the code. I knew the pressure sensor needed to be read first, then the LCD would display a greeting after. Following that, the touch screen would need to read a value and then the code would cycle through several different options. Though this main placement was easy to determine, I realized quickly that certain variables had to be global while others could be local. In addition to this, setting variables from a sensor reading and then using that variable later in the code was the issue. These problems stemmed from my novice experience with programming, but it was resolved through proper research and trial and error.

Given more time for the project, I would have liked to make the system more interactive and robust. Say, for instance, the delivery person needed my signature. I would've liked to have given sub options to allow for a more precise determination of why the person is visiting. Along with this, I would've liked to have added a notification system that alerted me when I was needed at the door. This could've been accomplished with a speaker or light inside the apartment, or even by sending a notification to my phone.

Appendix A: Final Project Photos

Figure 2: Polished Final Prototype

Figure 3: Final Prototype Circuitry



Appendix B: Circuit Schematics

Figure 4: Circuit Schematic

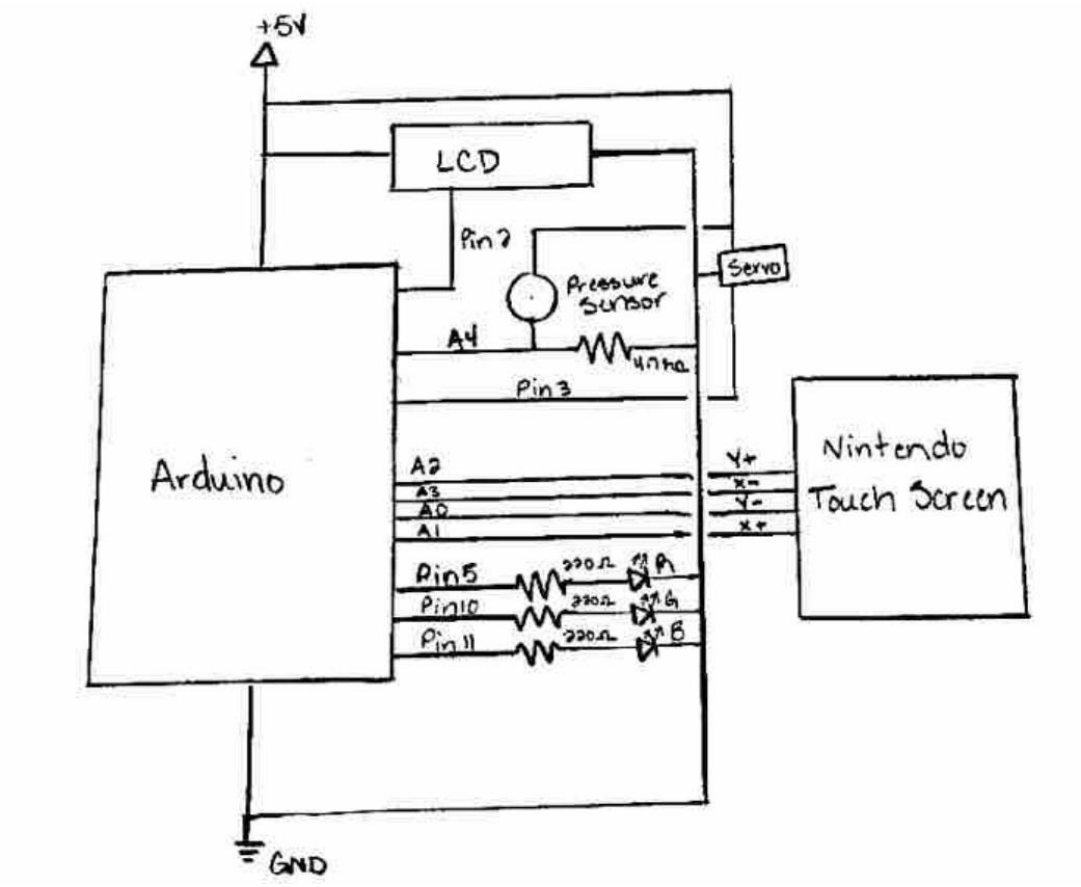
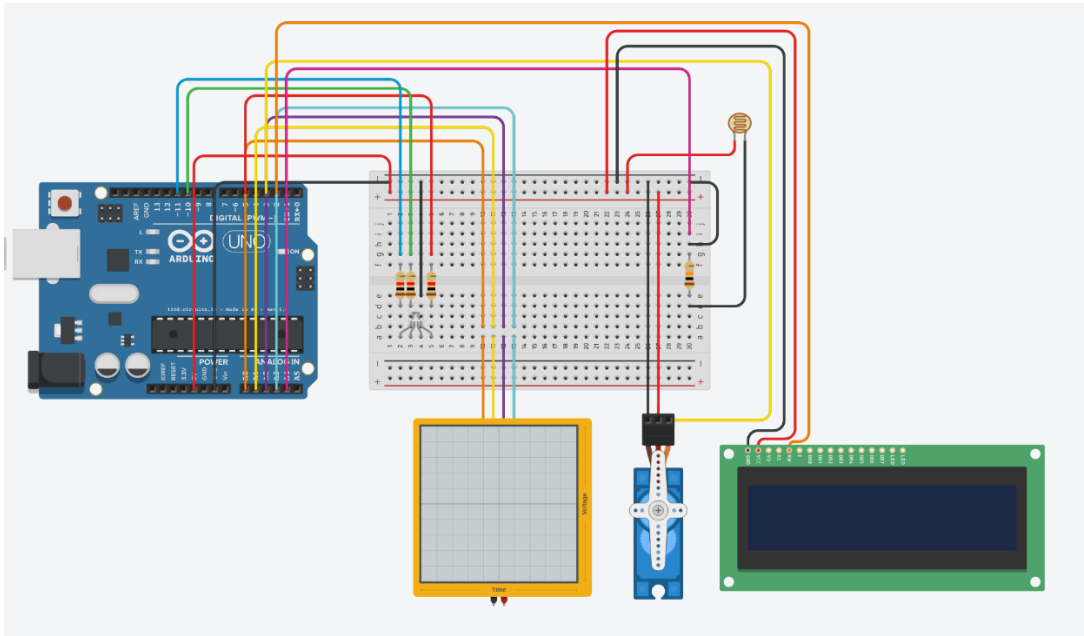


Figure 5: Circuit Diagram



Appendix C: Pseudocode

I. Prototype Pseudocode

```
Set RGB pins to PWM pins
Set Pressure Sensor pin to analog pin
Set Touch Screen pins to analog pins
Set servo motor and LCD pins
Read the data taken from the pressure sensor
If the reading is greater than the weight of a human
    Use the LCD to display a greeting
    Read touch screen data
    If the x and y readings are in the family quadrant
        Turn LED green
        Unlock the door by adjusting the servo motor 90 degrees right
        Display a message on the LCD welcoming the visitor inside
    If the x and y readings are in the salesperson quadrant
        Turn the LED red
        Lock the door by adjusting the servo motor 90 degrees left
        Display a message on the LCD requesting they leave
    If the x and y readings are in the delivery quadrant
        Turn the LED yellow
        Lock the door by adjusting the servo motor 90 degrees left
        Display a message asking that the delivery is left by the door
    If the x and y readings are in the Girl Scout quadrant
        Turn the LED green
        Unlock the door by adjusting the servo motor 90 degrees right
        Display a message asking them to wait while I come to the door
If the reading is less than the weight of a person
    Do not activate the rest of the system
Repeat the previous sequence indefinitely, constantly reading the pressure sensor
```

Appendix D: Arduino Sketch

I. Prototype Arduino Sketch

```
//include libraries
#include "stdint.h"
#include "Servo.h"
#include "TouchScreen.h"
#include "Arduino.h"

//touch screen setup
#define YP A2  // must be an analog pin, use "An" notation!
#define XM A3  // must be an analog pin, use "An" notation!
#define YM A0  // can be a digital pin
#define XP A1  // can be a digital pin
#define MINPRESSURE 10
#define MAXPRESSURE 1000
TouchScreen ts = TouchScreen(XP, YP, XM, YM, 300);


//LCD setup
#if defined(ARDUINO_ARCH_SAMD) || defined(__SAM3X8E__)
  #define lcd Serial1
#else
  #include <SoftwareSerial.h>
  SoftwareSerial lcd = SoftwareSerial(0,2);
#endif

//Pressure sensor setup
int fsrAnalogPin = A4; // FSR is connected to analog 0
int fsrReading;      // the analog reading from the FSR resistor divider

//RGB LED setup
const int RLED = 5;
const int GLED = 10;
const int BLED = 11;

//servo setup
int servoPin = 3;
Servo servo;

void setup(void) {
  Serial.begin(9600);   //use to determine values for sensors

  //set RGB LED as outputs
  pinMode(RLED, OUTPUT);
  pinMode(GLED, OUTPUT);
  pinMode(BLED, OUTPUT);

  pinMode(servoPin,OUTPUT);
  //setup servo
  servo.attach(servoPin);

  //setup LCD
  lcd.begin(9600);
  lcd.write(0xFE);
  lcd.write(0xD1);
  lcd.write(16);
```

```
  lcd.write(2);
  delay(10);

  //set contrast to 200
  lcd.write(0xFE);
  lcd.write(0x50);
  lcd.write(200);
  delay(10);

  //max brightness
  lcd.write(0xFE);
  lcd.write(0x99);
  lcd.write(255);
  delay(10);

  //turn off cursors
  lcd.write(0xFE);
  lcd.write(0x4B);
  lcd.write(0xFE);
  lcd.write(0x54);

  //create custom character
  lcd.write(0xFE);
  lcd.write(0x4E);
  lcd.write((uint8_t)0);
  lcd.write((uint8_t)0x00);
  lcd.write(0x0A);
  lcd.write(0x15);
  lcd.write(0x11);
  lcd.write(0x11);
  lcd.write(0x0A);
  lcd.write(0x04);
  lcd.write((uint8_t)0x00);
  delay(10);

  // clear screen
  lcd.write(0xFE);
  lcd.write(0x58);
  delay(10);

  // go 'home'
  lcd.write(0xFE);
  lcd.write(0x48);
  delay(1000);
}

void loop(void) {

  //read values Pressure Sensor
  fsrReading = analogRead(fsrAnalogPin);
  delay(500);

  //set pikachu weight
  int pikachuWeight=100;
```

```
  //if there is a visitor
  if (fsrReading > pikachuWeight){
    lcd.print("Hello!");
    delay(1000);
    lcd.write(0xFE);
    lcd.write(0x58);
    delay(10);
    TSPoint p = ts.getPoint();
    if (p.z > ts.pressureThreshhold) {
       Serial.print("X = "); Serial.print(p.x);
       Serial.print("\tY = "); Serial.print(p.y);
       Serial.print("\tPressure = "); Serial.println(p.z);
    }
    int xval = p.x;
    int yval = p.y;
    if (xval > 550 && xval <940 && yval > 180 && yval < 430)
      {
      salesPerson();
      delay(1000);
      }
    else if (xval > 550 && xval <940 && yval > 550 && yval < 830)
      {
      delivery();
      delay(1000);
      }
    else if (xval > 110 && xval <500 && yval > 550 && yval < 830)
      {
      relatives();
      delay(1000);
      }
    else if (xval > 110 && xval <500 && yval > 180 && yval < 430)
      {
      cookies();
      delay(1000);
      }
    }
  else if (fsrReading<pikachuWeight){
    noVisitors();
    delay(1000);
  }
}

void salesPerson(){
  digitalWrite(RLED,HIGH);
  digitalWrite(BLED,LOW);
  digitalWrite(GLED,LOW);
  lcd.println("Not available.");
  servo.write(0);
  delay(3000);
  lcd.write(0xFE);
  lcd.write(0x58);
}
```

```
void delivery(){
  analogWrite(RLED,200);
  analogWrite(BLED,0);
  analogWrite(GLED,90);
  lcd.println("Leave it.");
  servo.write(0);
  delay(3000);
  lcd.write(0xFE);
  lcd.write(0x58);
}

void relatives(){
  digitalWrite(RLED,LOW);
  digitalWrite(BLED,LOW);
  digitalWrite(GLED,HIGH);
  lcd.println("Come inside!");
  servo.write(180);
  delay(3000);
  lcd.write(0xFE);
  lcd.write(0x58);
}

void cookies(){
  digitalWrite(RLED,LOW);
  digitalWrite(BLED,LOW);
  digitalWrite(GLED,HIGH);
  lcd.println("Be right there!");
  servo.write(180);
  delay(3000);
  lcd.write(0xFE);
  lcd.write(0x58);
}

void noVisitors(){
  digitalWrite(RLED,LOW);
  digitalWrite(BLED,LOW);
  digitalWrite(GLED,LOW);
  servo.write(0);
}
```