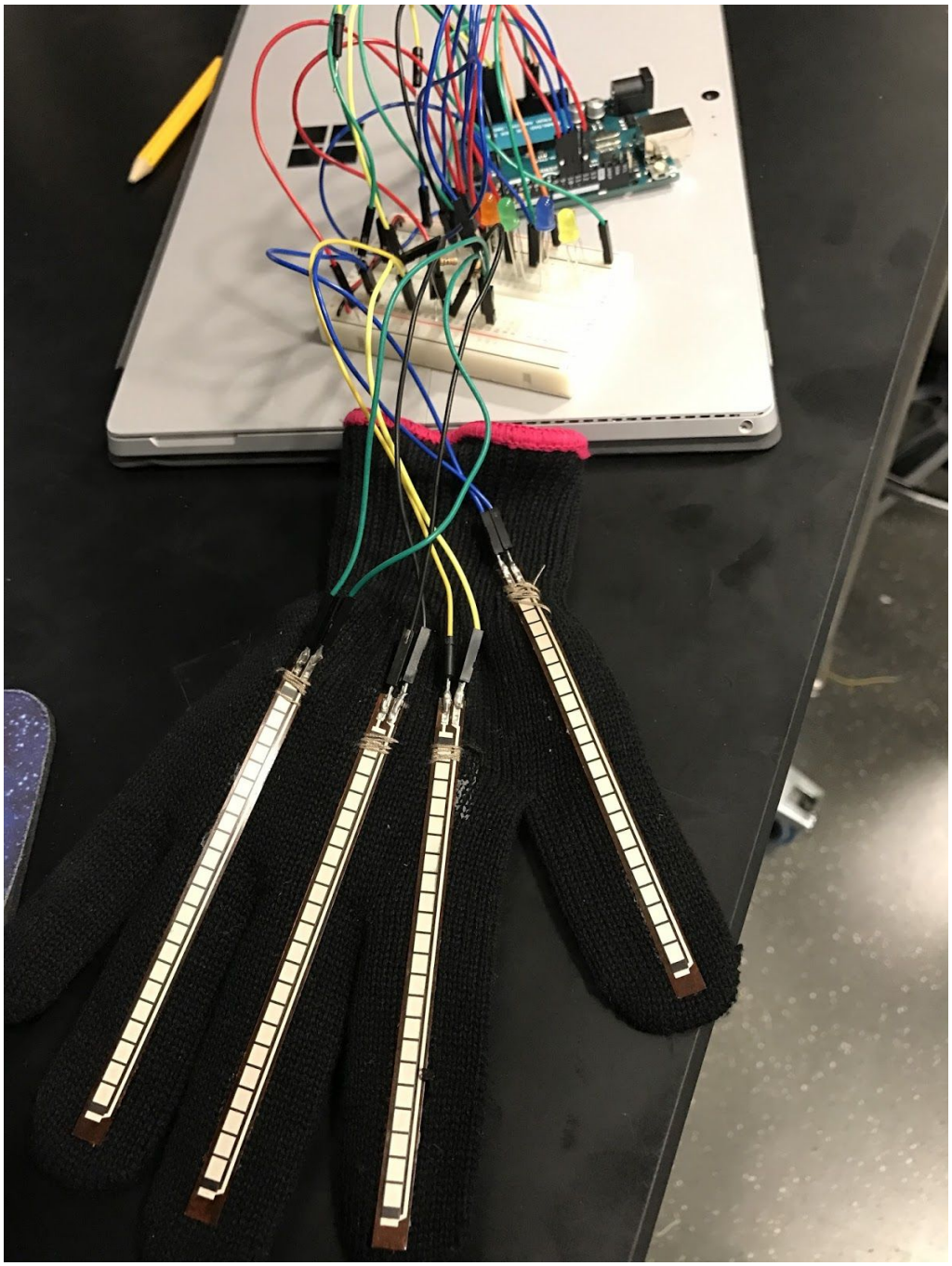
Smart Glove

ECE 110 Honors Final Report



Andre He(songtao2), Jingwei Zhuo(jingwei2), Luting Chen(lchen108)

12.12.2016

INTRODUCTION

Statement of Purpose

We would like to introduce a different type of motion pattern lock, which would be a glove lock that could identify certain gestures as the correct password to unlock a system. Since memorizing gestures and hand motions is much easier than memorizing numbers, this type of lock would be much easier to use than the typical numbers and letters lock.

Features and Benefits

Most locks for today require users either to keep a physical key with them or to memorize a serial of numbers/letters or a certain shape pattern. On the one hand, surveys have shown that in every five people, there are two whose passwords have been compromised in the past year. On the other hand, there are also researches showing that through subsequent practice using muscle memory, people will be able to reproduce movement patterns effortlessly and "subconsciously". Since the risk of using both physical keys and electronic passwords are fairly high, it would be reasonable to introduce a new kind of lock that only requires a much simpler and more basic memorization for users ---- muscle memory for basic single hand movements.

DESIGN

System Overview

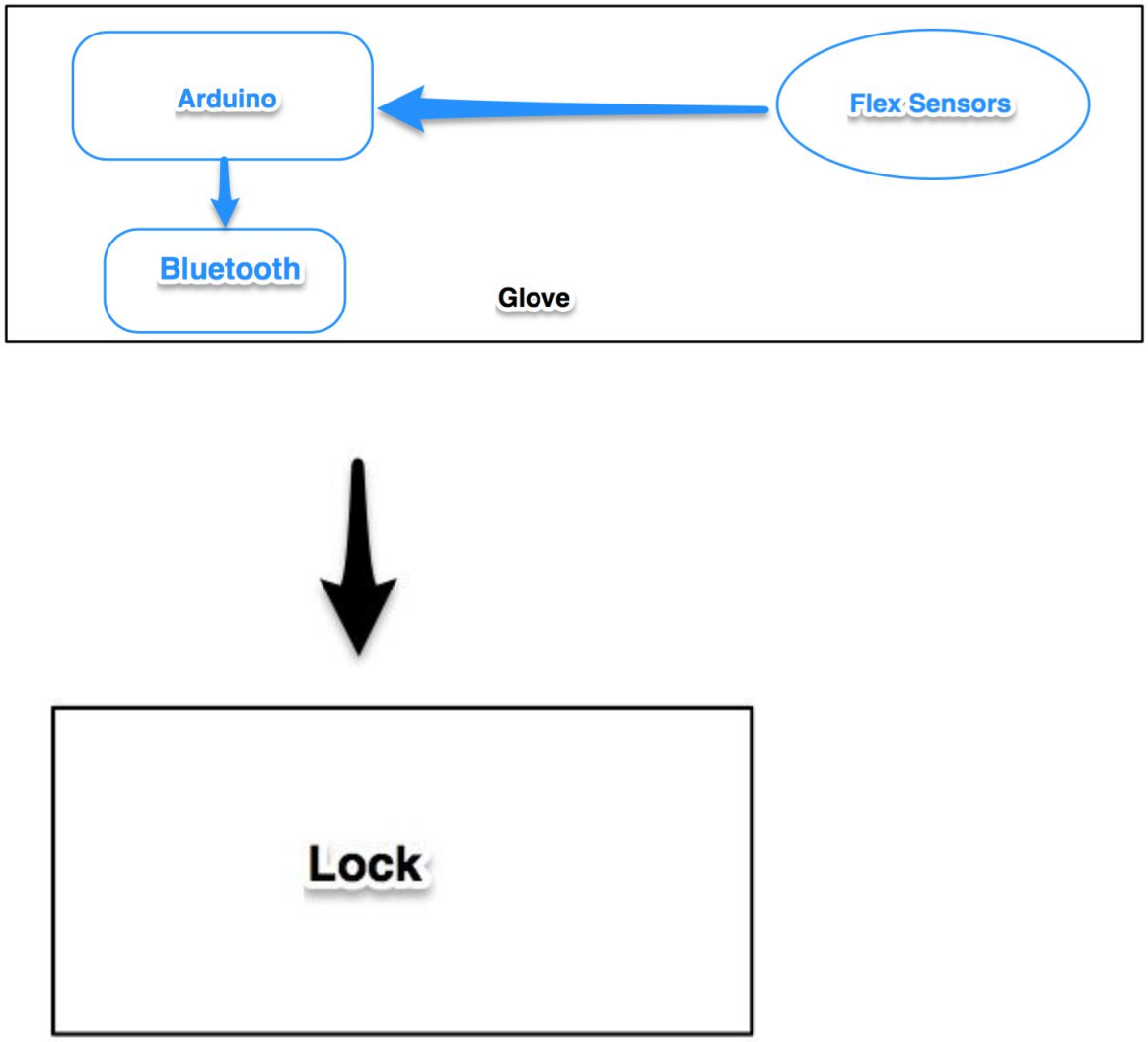
Our project contains two main component: the glove part and the microcontroller part.

The smart-glove attaches a PCB board connecting multiple flex resistors between the upper layers

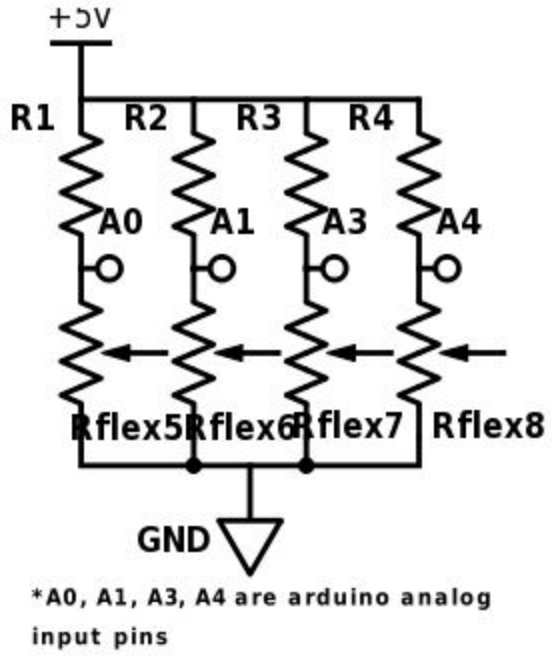
of the glove. When the wearer curls each finger, the corresponding flex resistor changes its resistance and so does the current. So far, we would like to allow four movements of fingers to transmit five different signals, which are automatically sent from glove to Bluetooth module of microcontroller.

Microcontroller opens the lock only if the gestures (converted to signals) are in the right order. Logic gates and functions will be implemented to build the circuit with an Arduino Uno.

Block Diagram and Flowcharts



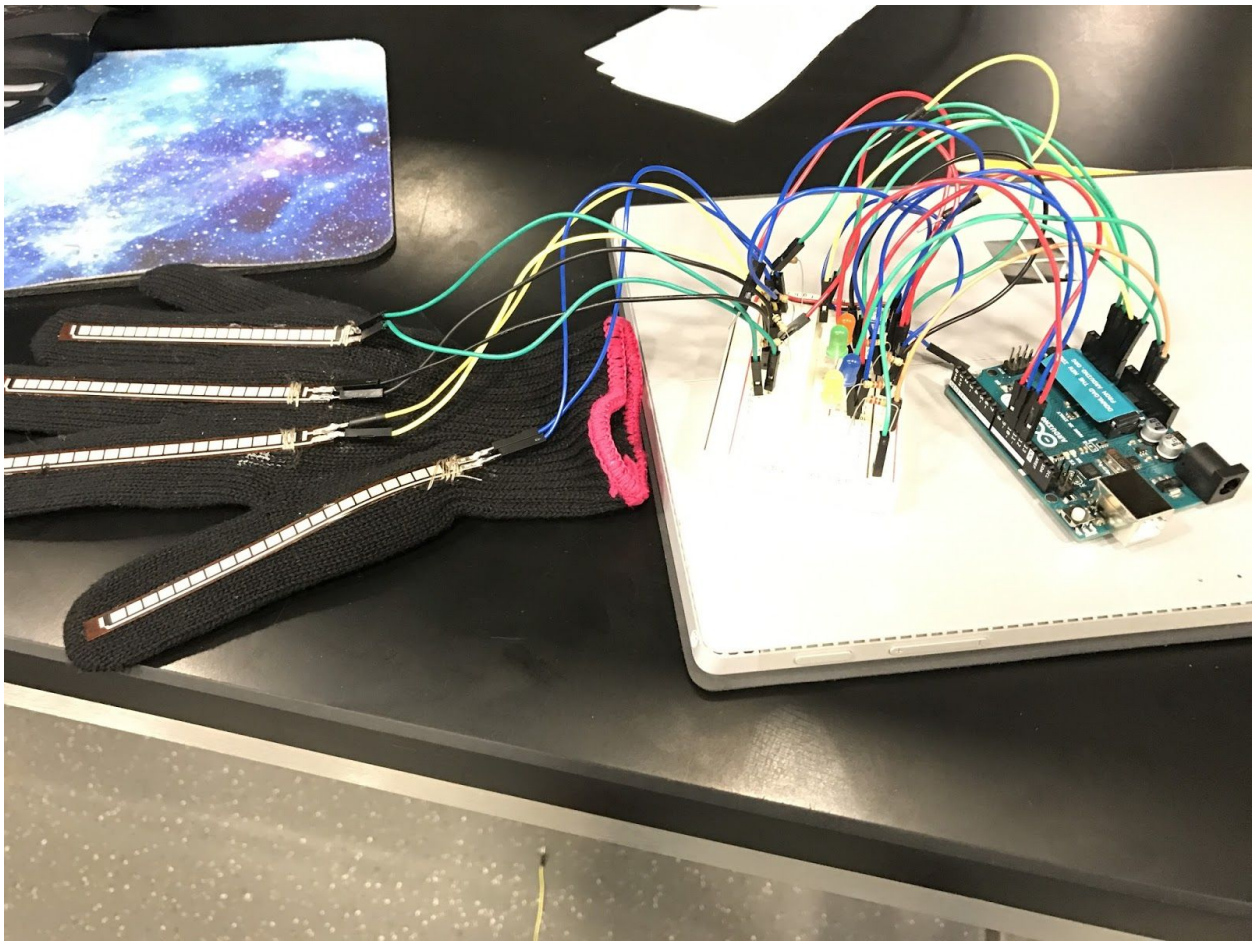
The four flex sensors on the glove (each on a finger) sense the hand motion and can generate an output according to the finger is curled or not. Each flex sensor is connected in series with a resistor so when it's curled, the voltage across this flex sensor will increase. This signal is sent to the Arduino (analog input pins). If the voltage across the flex sensor goes beyond a threshold voltage values, the arduino will read it as a "true" value for that flex sensor.The arduino board collects signals from each finger and reads the pattern. If the pattern corresponds to the one we set, it will produce an output control signal to unlock the device. One of the flex sensor is used as reset control, which is whenever it's curled, the arduino will be reset and start to read the signals from the beginning.



Design Details

For each block in your system block diagram, provide the design details. Include any relevant design considerations, block diagrams, calculations, pictures, circuit schematic, flow charts, truth tables, etc. that will aid your description.

Originally we considered to use all five of the fingers as input. However, when we were testing out our device, we found that the it's hard to curl the little finger independently. It always moves when the ring finger is curled. So we decided to only use the first four fingers and leave out the little finger.



I.hardware

1) Flex Resistors

2) Wifi module

3) LEDs

4) Glove (pre-made/row material)

II.Software

1) AT command lines

2) NetAssist

RESULTS

1. In this section provide the characterization of your sensors and actuators. Also provide the results of any test you have conducted with your system that verifies its functionality.

Flex sensors respond to the angle bent by fingers by varying its resistance. We used 5 Flex sensors to register signals presented by a gesture.

Wifi module reads data from Arduino Uno and sends it to the computer. We set up a TCP server on our computer which shows all connections. Wifi module is the TCP client which reports data to the computer.

PROBLEMS AND CHALLENGES

Originally our design was to utilize the wifi module to send feedback. However, we encountered some problems with the wifi modules when we're trying to enter the correct IP address of the server. It continues to report wrong IP address. As said on official tutorial, if you entered the right AT commands but such a problem still exists, ESP8266 may have some issues and customers can report to the company service.

FUTURE PLANS

Use a PCB board instead of Arduino Uno. Improve on code to reset the lock.

Test result summary

Since we did not finish the Wi-Fi module coding part on time, we just test our Arduino code.

We had problem when we tried to use array<> method with our gloves, while we want to keep "multiple gesture attestation." So we just adopted decimal computation and run the "success" code when n = 15 (we manipulate this value). We made sure that only a special sequence of gesture by curling fingers would trigger the "success" code.

We set 1 seconds interval for Arduino to process following gesture. If user did not continue interactions, Arduino would repeatedly read the same code (which usually triggered "locked" code since combination was set randomly).

We also designed the "locked" system, which would automatically locked the Arduino with three unsuccessful trials. If the glove is placed in 20 seconds without interaction, the Arduino

would lock itself as well.

When we test our glove, we realize that the locker system functioned well. Curling fingers would run the program as we planned. However, we only successfully ran “unlocked” code once.

In summary, we realize that our glove system was not completed. The trigger time for 1 second is not very reasonable, for each user may prefer different interacting pattern with different intervals. Besides, the resistance of different flex resistors are not uniform, which may result in pattern mismatch and erroneously trigger “locked” code.

To resolve those problems, we need to unify the resistance of each flex resistor. We may apply array<> code in future since it is more flexible and easy to debug. We may abolish the “interval” code, while command that no element in the combination can be repeated continuously.

REFERENCES

1. "Password Statistics: The Bad, the Worse and the Ugly (Infographic)", 2015.[Online] https://www.entrepreneur.com/article/246902 [26-Sep-2016].
2. Lee, D.T., & Schmidt, A.R. (2005). Motor Control and Learning: A Behavioural Emphasis. (4th ed). Windsor, ON: Human Kinetics
3. http://www.digikey.com/schemeit/project/

CODE APPENDIX

AT commands used:

AT+RST————————————-reset

AT+CWLAP ————————————-list Wifi options

AT+CWMODE=3 ————————————-allow to show SPA and AP

AT+CIPMUX=0 ————————————allow data transferring

AT+CWJAP=“ssid”, “passwd” ——-connect to Wifi

AT+CIPSTART=“type”, “IP”, port —start TCP transmission

AT+CIPSEND=——————————send TCP sockets

Arduino Code:

```
/ The trigger value
  for the flex resistor
  is 560/1024*5 = 2.73V
*/
const int initial = 560;
int n = 0;;
int g = 0;

void setup() {
  Serial.begin(9600);
  pinMode(9,OUTPUT);
  pinMode(10,OUTPUT);
  pinMode(11,OUTPUT);
  pinMode(12,OUTPUT);
  //blink for 3 times
  digitalWrite(10, HIGH);
  digitalWrite(11, HIGH);
  digitalWrite(12, HIGH);
  delay(200);
  digitalWrite(10, LOW);
  digitalWrite(11, LOW);
  digitalWrite(12, LOW);
```

```
  delay(200);
  digitalWrite(10, HIGH);
  digitalWrite(11, HIGH);
  digitalWrite(12, HIGH);
  delay(200);
  digitalWrite(10, LOW);
  digitalWrite(11, LOW);
  digitalWrite(12, LOW);
  delay(200);
  digitalWrite(10, HIGH);
  digitalWrite(11, HIGH);
  digitalWrite(12, HIGH);
  delay(200);
  digitalWrite(10, LOW);
  digitalWrite(11, LOW);
  digitalWrite(12, LOW);
  delay(200);
}

void loop() {
  int i = analogRead(A1); //signal 1
  int j = analogRead(A2); //signal 2
  int k = analogRead(A3); //signal 3
  int m = analogRead(A0); //signal Retryt
```

```
//  if (m > initial) {
//    n == 0;
//    g == 0;
//    digitalWrite(9,HIGH);
//    delay(500);
//    digitalWrite(9,LOW);
//    delay(500);
//    digitalWrite(9,HIGH);
//    delay(500);
//    digitalWrite(9,LOW);
//    delay(500);
//  }

  if (i<initial and j<initial) {
      g = g + 1;
      digitalWrite(9,HIGH);
      delay(50);
      digitalWrite(9 , LOW);
      delay(1050);
  }
```

```
if (i>initial and j<initial and k<initial) {
  n = n + 10;
}

if (i>initial and j>initial and k<initial) {
  n = n - 2;
}

if (i>initial and j>initial and k>initial) {
  n = n * 2 + 1;
  Serial.println(n);
}

if (i>initial and j<initial and k>initial) {
  n = n + 1;
}

if (i<initial and j>initial and k<initial) {
  n = n + 1;
}

if (i<initial and j<initial and k>initial) {
  n = n + 1;
  Serial.println(n);
```

```
}

if (i<initial and j>initial and k>initial) {
  n = n + 1;
}

//success code
if (n == 15) {
  Serial.println("Unlocked");
  digitalWrite(10,HIGH);
  digitalWrite(11,HIGH);
  digitalWrite(12,HIGH);
  delay(5000);
  digitalWrite(10,LOW);
  digitalWrite(11,LOW);
  digitalWrite(12,LOW);
  while(1);
}

if (n == 3 || n == 5 || n==7 || n==13 || n==23  || n==19 || n >= 50 || g >=20) {
  Serial.println("Error");
  digitalWrite(9,HIGH);
  delay(3000);
  digitalWrite(9,LOW);
```

```
    while(1);
  }
  //1s interval
   else {
      delay(1000);
      Serial.println("Please try again");
    }
}
```