Remote Modulated Security (RMS)

Lam Tran, Troy Cosino, Brian Aguirre

IEEE Quarterly Project ++ Team 104

Table of Content

Introduction

According to the latest *FBI Uniform Crime Reporting Statistics - Property Crime Rates*, there are roughly 2.5 million burglaries a year, and 66% of those being home break ins. The report also states that the police solve only 13% of reported burglaries due to lack of witness or physical evidence for most of those cases. With the recent rise in popularity of IoT in our society, our solution is to design a door with an integrated fingerprint scanner to create a more reliable security system as well as wireless controller that would be convenient for the user to let incoming guests enter the home. In addition to security, we also wanted to add a feature that enabled preferences for the ambiance of the home. In other words, when a house member enters the home the lighting of the room will automatically change depending on the user who entered the house as well as a customized greeting displayed through an LCD display.

For our product, we will be using an Arduino Mega 2560, Bluetooth 4.0 module, 16x2 LCD Arduino Display Screen, RGB LED, and an optical Fingerprint Reader Sensor Module. However, for the prototyping of our design we decided to develop CADs for the door and the casing through Solid Works. In addition to the wiring and resistors one may sufficiently develop our product through the mentioned hardware.

Software

I. Software:

A. Arduino IDE
B. Blynk for iOS
C. SolidWork
D. LulzBot

II. Libraries:

A. Adafruit-Fingerprint-Sensor-Library-master
B. Blynk 0.6.1
C. Servo
D. LiquidCrystal_I2C
E. SoftwareSerial

III. Code:

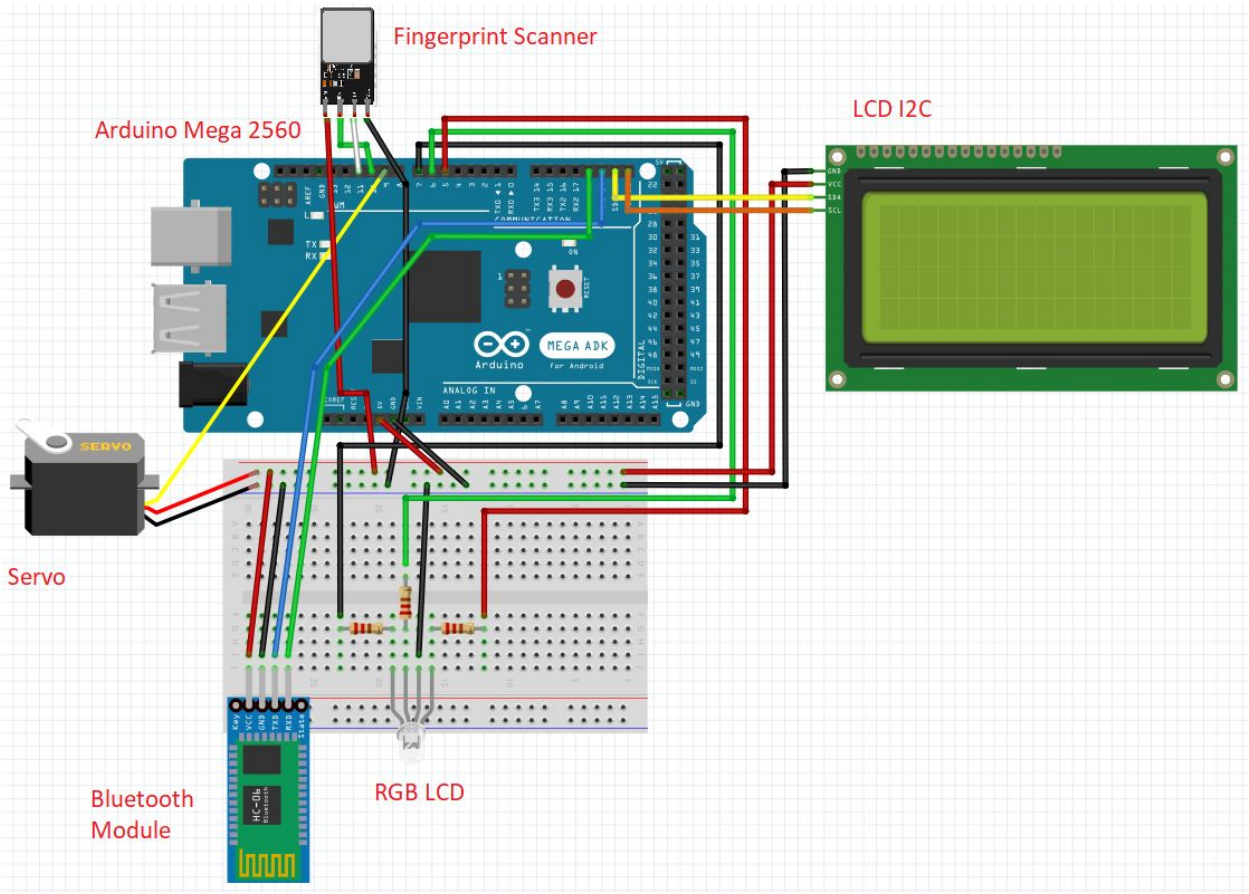
A. https://github.com/b2aguirr/RMS (servo_blynk.ino)

Hardware

- Bluetooth 4.0 BLE Slave UART Serial Module
  - Price: $7.99
  - Vendor: DSD Tech - Amazon
- Optical Fingerprint Reader Sensor Module
  - Price: $20.99
  - Vendor: DIYmall - Amazon
- 16x2 1602 LCD Arduino Display Screen Blue + IIC I2C Module Interface Adapter
  - Price: $9.99
  - Vendor: JANSANE - Amazon
- Servo, RGB, Arduino Mega 2560, Jumper Wires, 330Ω, Breadboard
  - Price: Free
  - Obtained from ECE 5 class
- Loctite fun-tak mounting putty
  - Price: ~$0.75 used
  - Vendor: Loctite - Amazon
- 3D printed door, doorframe, and base
  - Price: ~$14
  - Printed at SMEs Envision

Instructions

I. Circuit Diagram and Parts



Circuit Diagram (Created using *Fritzing*)

Purpose of Each Component

1. HM-HC-08 Bluetooth module:
    a. We chose this module for wireless communication rather a wifi module because the wifi at the presentation room might not be reliable.
2. RGB LED:
    a. Displays users prefered color.
3. Servo:
    a. Locking mechanism. Rotates between lock and unlock position.
4. Fingerprint Scanner:
    a. Biometric security.
5. LCD I2C:
    a. Displays state of door lock as well as customizable greetings per user.

II. Code

```
/**
 *  Unlocks door, changes lighting, and prints message for entering
 *  based on user through bluetooth app buttons.
**/
BLYNK_WRITE(V3)
{
  servo.write(param.asInt());

if(param.asInt() == 45){
  setColor(0,0,255);
      lcd.clear();
      lcd.setCursor(3,0);
      lcd.print("Heeeeere's");
      lcd.setCursor(6,1);
      lcd.print("LAM");
      delay(1000);
}
else if(param.asInt() == 46){
  setColor(0,255,0);
      lcd.clear();
      lcd.setCursor(3,0);
      lcd.print("Greetings");
      lcd.setCursor(6,1);
      lcd.print("Troy");
      delay(1000);
}
else if(param.asInt() == 47){
  setColor(128,0,128);
      lcd.clear();
      lcd.setCursor(2,0);
      lcd.print("Welcome Home");
      lcd.setCursor(5,1);
      lcd.print("Brian");
      delay(1000);
}

if(param.asInt() == 0){
  setColor(255,0,0);
      lcd.clear();
      lcd.setCursor(6,0);
      lcd.print("Door");
      lcd.setCursor(5,1);
      lcd.print("LOCKED");
      delay(1000);
```

BLYNK_WRITE(V3):

The buttons of our bluetooth app changes param.asInt() to the numbers from 45-47 where each number is a preset setting that sets the angular position of the servo while changing the lighting, prompts a special message, and unlocks the door for the house member. If the button is unchecked, it will turn zero and lock the door. In other words, param.asInt() is a switch which unlocks and locks the door.

```
void setup()
{
  // Debug console
  Serial.begin(9600);
  Serial1.begin(9600);
  finger.begin(57600);
  Blynk.begin(Serial1, auth);
  servo.attach(9);
  pinMode(redPin,OUTPUT);
  pinMode(greenPin,OUTPUT);
  pinMode(bluePin,OUTPUT);
  finger.verifyPassword();
}
```

Setup():

Assigns the pins for wiring/connections, fingerprint scanner and initializes channels for app/bluetooth connectivity.

```
// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK)  return -1;


  p = finger.image2Tz();
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK)  return -1;

  // found a match!
  Serial.print("Found ID #"); Serial.print(finger.fingerID);
  Serial.print(" with confidence of "); Serial.println(finger.confidence);
  return finger.fingerID;
}
```

getFingerprintIDez():

Checks the fingerprint scanner's internal memory which returns fingerID associated with registered fingerprint else returns -1.

```
/** Sets color of LED **/
void setColor(int red, int green, int blue){
  analogWrite(redPin,red);
  analogWrite(greenPin,green);
  analogWrite(bluePin,blue);
}
```

setColor():

Changes the RGB LED values. Ex: (255,0,0) is red).

```
/** Changes the orientation of Servo CCW **/
void ServoknobCCW(){
  servo.write(45);
  delay(5);
}

/** Changes the orientation of Servo CW **/
void ServoknobCW(){
  servo.write(0);
  delay(5);
}
```

ServoKnobCWW() & ServoknowCW():

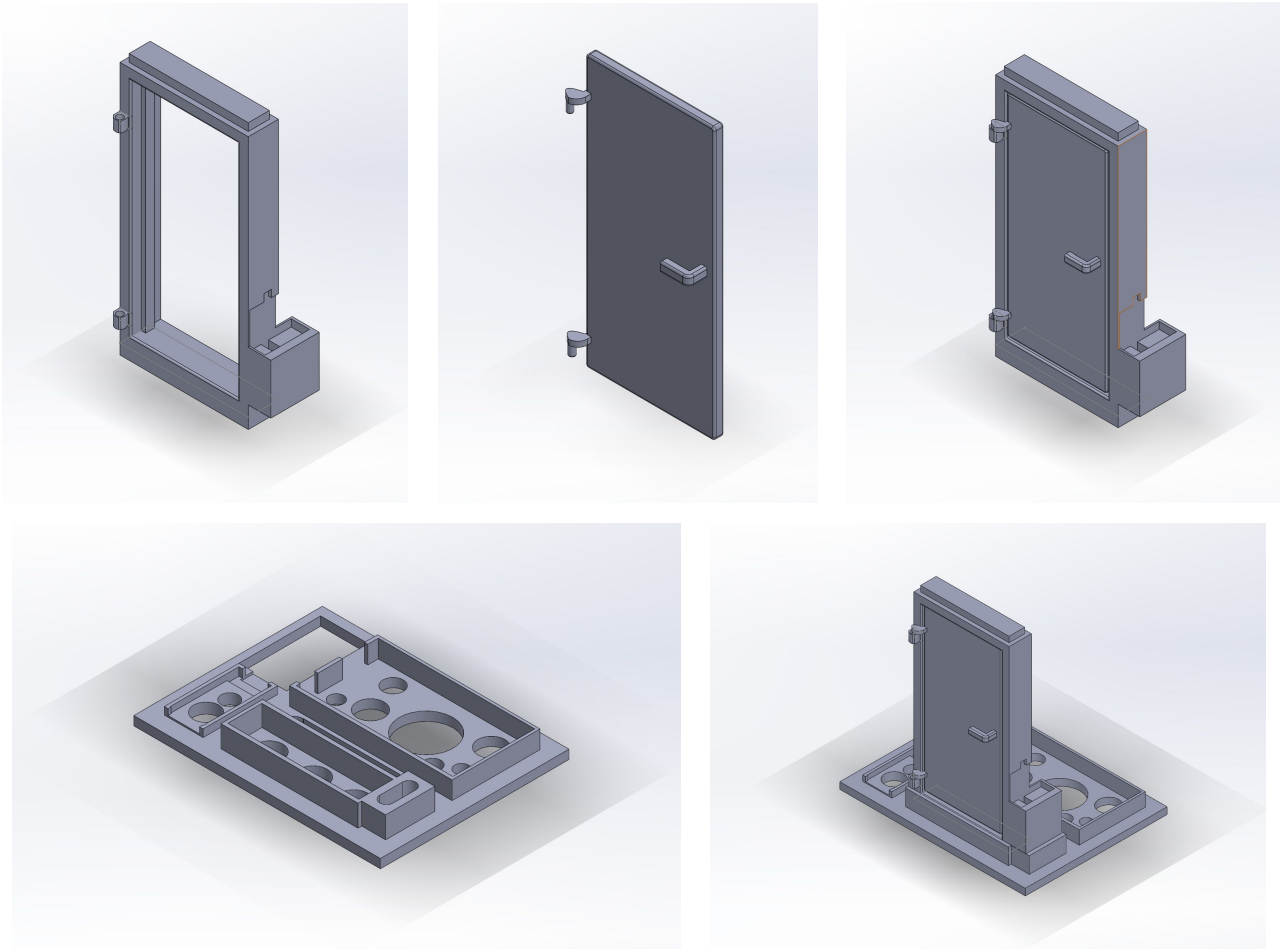
Locking system which changes the orientation of servo for unlocking and locking the door.

```
/**
 *  Continuously checks fingerprint scanner for household members. If the user
 *  is a valid home member then the door will unlock, change lighting, and print
 *  a message. The door will lock after 5 seconds.
**/
void loop()
{
  Blynk.run();

    lcd.init();
    lcd.backlight();
    lcd.setCursor(6,0);
    lcd.print("Door");

    int member = getFingerprintIDez(); // Reading fingerprint scanner
    delay(50);

    // Checks if user is a member
    if(finger.fingerID == member){
        if(member == 1 || member == 2 || member == 3){
            setColor(0,0,255);
            ServoknobCCW();
            lcd.clear();
            lcd.setCursor(2,0);
            lcd.print("Heeeeere's");
            lcd.setCursor(4,1);
            lcd.print("LAM");
            delay(5000);
        }

        ServoknobCW();
        lcd.clear();
        lcd.setCursor(6,0);
        lcd.print("Door");
        lcd.setCursor(5,1);
        lcd.print("LOCKED");
        setColor(255,0,0);
    }
}
```

loop():

Code for the whole integrated system which continuously checks the fingerprint scanner and gets updated from bluetooth app to open the door. If the fingerprint scanner identifies a household member then the door unlocks, changes the lighting of the LED and immediately locks after 5 seconds. The system reacts the same if the bluetooth app is pressed however you must lock the system by pressing the same button one more time.

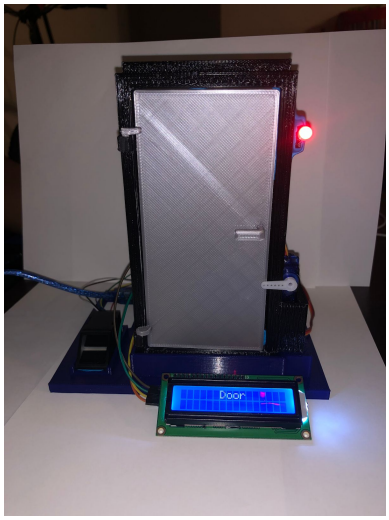
III. 3D Printing



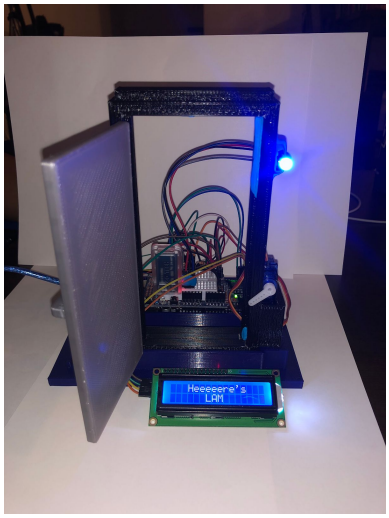
SolidWorks Renditions of Parts

To create a functioning door, we measured our apartment door then scaled it down with a ratio of 13:1. The hinge was improvised as the real hinge was made of metal and much too complex. A small raised platform was created to house the servo. A base was implemented with housing for the doorframe, the fingerprint scanner, the Arduino Mega 2560, and single 9-volt battery. Multiple circles were extrude cut into the base to reduce printing time. Each component was printed on the Lulzbot 3D printer using PLA material and standard printing profile at 10% infill and skirt plate adhesion. Due to time constraint and resources, we created the components with high tolerance to account for printing errors. During assembly, small amounts of mounting putty were utilized at each joint to stabilize the pieces as well as the fingerprint scanner and Arduino.

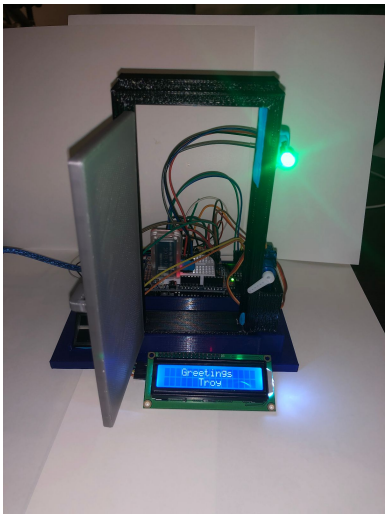
IV. RMS IN ACTION



Locked

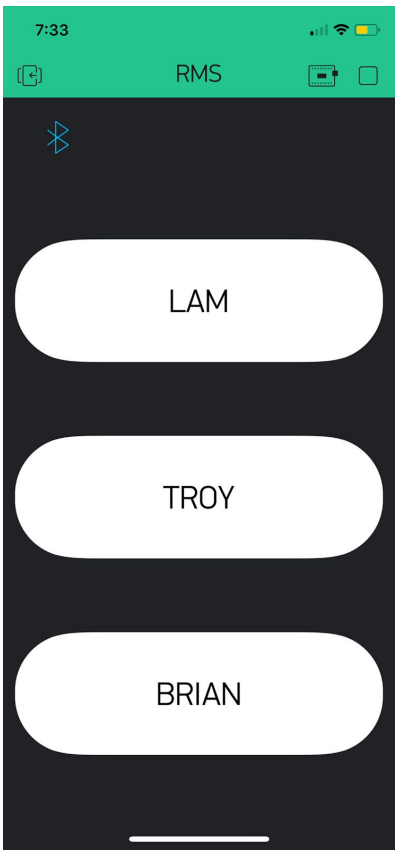


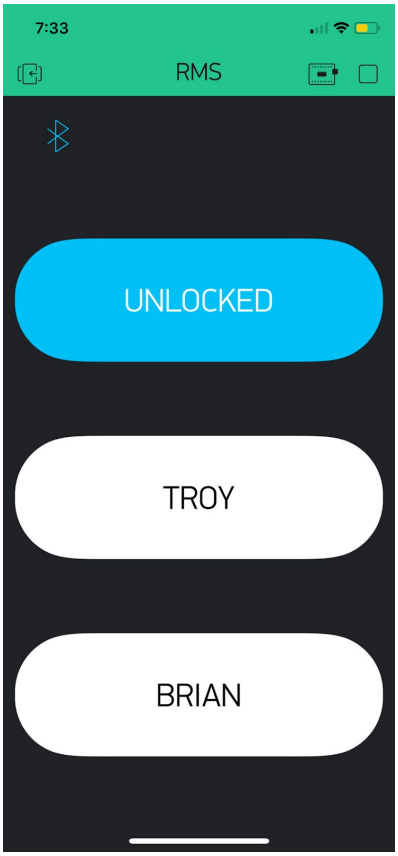
User: Lam

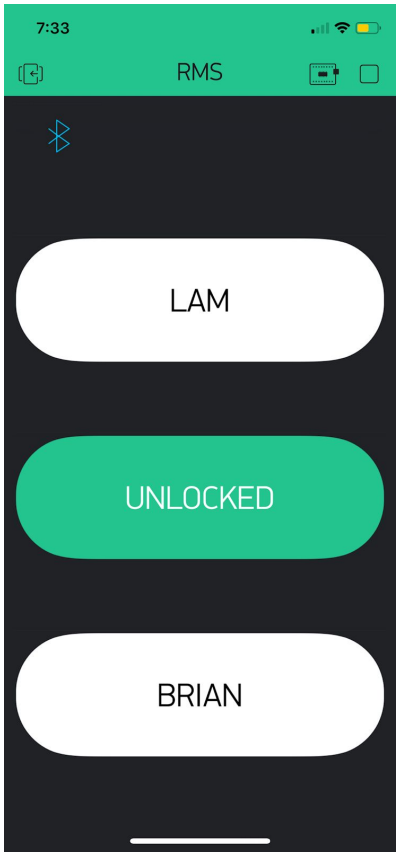


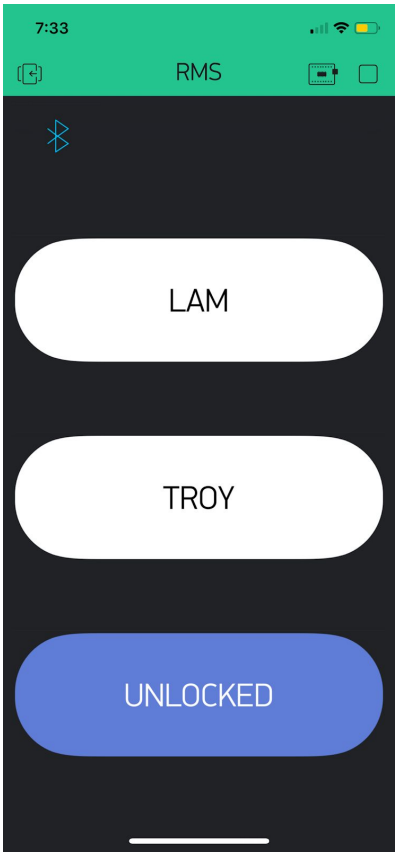
User: Troy

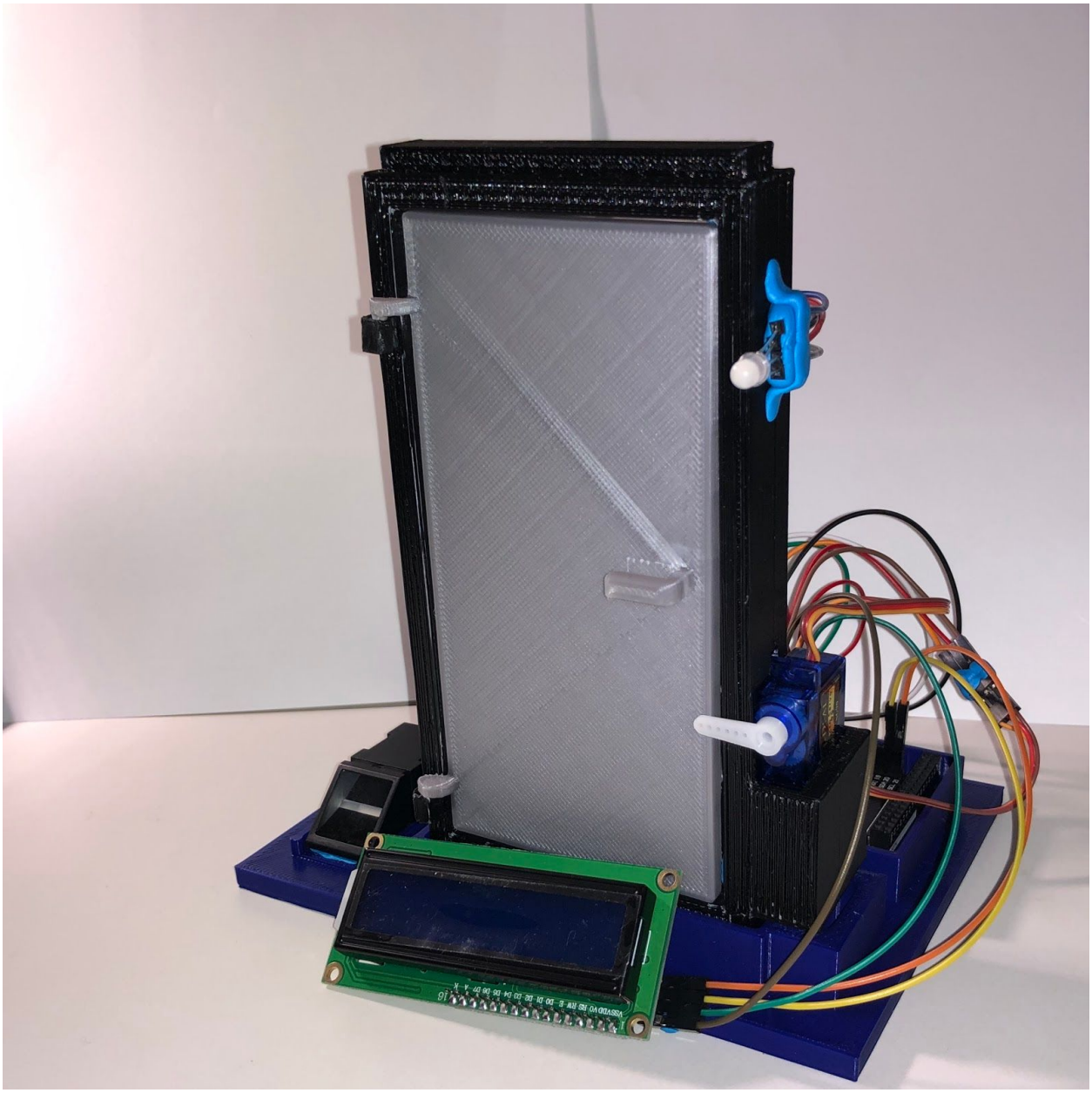
User: Brian











Assembled Product

Conclusion

Possible Improvements

Our product is a base model, however, if we were given more time we would focus on creating temperature, lighting, and music (optional) preferences when entering the home. Getting the door to open wirelessly would also be a more desirable option rather than using bluetooth. Also, we would implement an alarm system for multiple failed attempts in trying to read fingerprints. A small camera could be added to quickly record each fingerprint attempt. Our last idea was also to create a website to implement the same concept but through your home computer or laptop while also keeping records on who entered the house along with timestamps.

Biggest Challenges

One challenge was changing the whole project three weeks before presentation. We had to scrap the idea of facial recognition as the main component for our project due to miscommunication. One of our partners had the main component for the project (Google Coral) but he repeatedly missed meetings at the last minute. Software side of the project then led to no progress, however, the hardware side of the project finished and met the deadlines in an appropriate manner.

The other challenge was with the coding. We had a lot of trouble integrating the fingerprint scanner and the bluetooth system, which had conflicting libraries and wiring. Since it was our first time working with the bluetooth module, we did not realize that the TX and RX pin it used was conflicting with the fingerprint sensor's own pins. So we had to do a bit of research to integrate both of the codes.

The last issue that we faced was 3D printing. We had to reprint the door and door frame 3 times. The first print had scaling issues, the hinge did not line up properly. For the second print, we returned to Envision after the print was finished but it was misplaced so we had to reprint. So just to be safe, we made the tolerance for the connections and joints of the final print relatively high to account for any possibility of errors.

References

1. https://www.alarms.org/burglary-statistics/
2. https://blynk.io/en/getting-started
3. https://drive.google.com/file/d/1xpg27co8GutNphsmxyQD886YpeVYxViO/view
4. https://www.amazon.com/gp/product/B0762FF6MS/ref=ppx_yo_dt_b_asin_title_o01_s00?ie=UTF8&psc=1
5. https://www.amazon.com/Optical-Fingerprint-Reader-Arduino-Mega2560/dp/B077GKPWMN/ref=sr_1_1?keywords=diymall+optical+fingerprint&qid=1574037197&sr=8-1
6. https://www.amazon.com/JANSANE-Arduino-Display-Interface-Raspberry/dp/B07D83DY17/ref=sr_1_fkmr0_1?keywords=cd+16x2+with+an+i2c+backpack&qid=1573335699&sr=8-1-fkmr0
7. https://github.com/b2aguirr/RMS
8. https://fritzing.org/home/
9. https://www.arduino.cc/en/Tutorial/TwoPortReceive