University of Illinois at Chicago
CS 362: Computer Design, Spring 2018
Final Project Write Up

Fingerprint Garage Door System

Jahnvi Patel, Patrick O'Connell

ABSTRACT

In this paper, we introduce a fully functional Fingerprint Garage Door System with a garage model which is intended to test and demonstrate critical elements of the working system.

The Fingerprint Scanner (FPS GT-521F32) receives fingerprint data via an optical sensing area and stores data to identify an authorized user in the memory. On subsequent operation, once a fingerprint dataset is received, an attempt is made to match the incoming fingerprint with the previously stored set of authorized fingerprints. Thus, establishing the identity of the user according to highly reliable biometric principles.

The FPS GT-521F32 responds to the received fingerprint and activates the opening or closing of the garage door based on the current state of the garage if and only if a registered fingerprint is recognized. This provides a high level of security assurance and protection while enabling a seamless user experience.

INTRODUCTION

This project utilizes a Fingerprint Scanning Unit (FPS GT-521F32) to scan and identify fingerprints as well as opening and closing the garage door with verified fingerprint IDs. This process is controlled by Kuman UNO R3 Development Board(s) and programmed with open source Arduino Software (ARDUINO 1.8.5).

The main function of this module is to sense a fingerprint from the Fingerprint Scanner and send the received information to the UNO R3 to decode between authorized or unauthorized fingerprints. If an authorized fingerprint is encountered: the LCD prints a message, the garage door opens, and the garage state variables are toggled. If an unauthorized fingerprint is encountered, the LCD prints an error message for 2.5 seconds and the unit resumes normal functioning. Upon receiving a valid fingerprint ID, the UNO R3 sends a signal to another UNO R3 board, which then handles the opening of the garage.

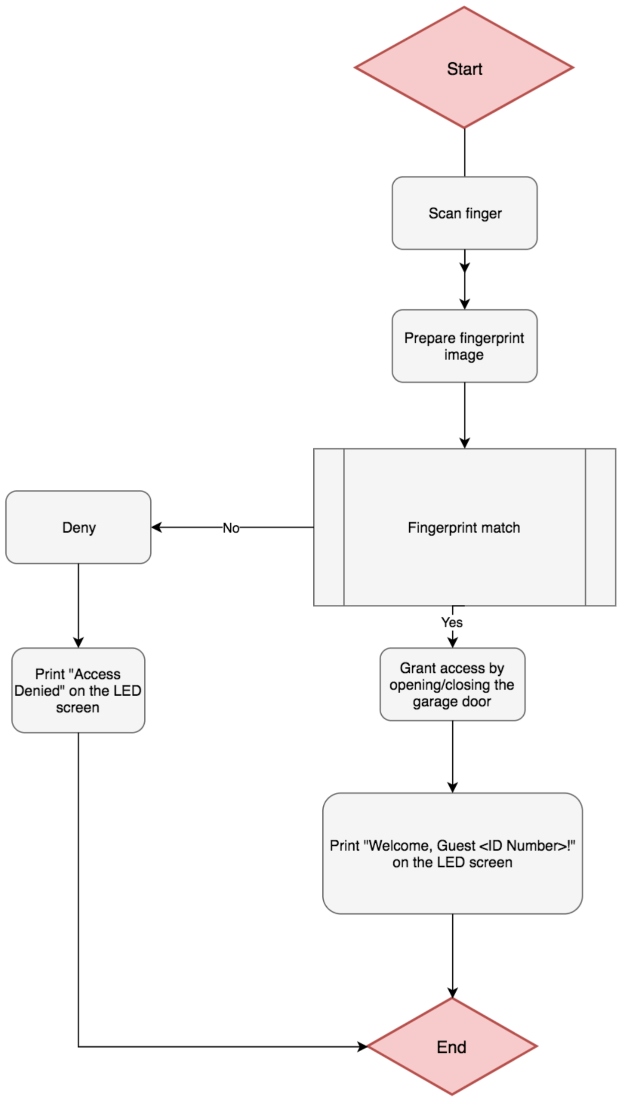


Figure 1 *The block overview of Fingerprint Garage Door System*

The system provides an automatically operable garage door system which positively blocks any unauthorized opening or closing of a garage door. It works in combination with a standard automatic electric control circuit that drives the garage panel upwards or downwards when the stepper motors are actuated in response to the receipt of an electrical signal generated when an authorized fingerprint is recognized by the FPS GT-521F32 module.

Electronics Needed:

Item	Qty
Kuman UNO R3 Development Board	2
Fingerprint Scanner - TTL (GT-521F32)	1
Standard LCD Screen – 16 x 2 Display	1
Key Switch w/ Key Caps	3
B10K Potentiometer	1
ULN2003 Stepper Motor Driver Board	2

Tools Needed:

Item	Qty
Soldering iron/solder	1
USB Cable	1
JST SH Jumper 4 Wire Assembly – 8”	1
Resistors	6
Glue gun w/ Glue sticks	1
Wooden Dowels	4
Hinges	16
Hot Wheels Wheel	8
830-Point Solderless Breadboard	1
¼ inch Smooth Particle Board	6
Sheet Metal	1
Nails	>20
Screws	>50
10” Wooden Roller	1
Female-to-male DuPont Line 1 x 10 Pin	2
Jumper Cables	25
Drill	1
32 sq. ft 3/16 inch beadboard	1
Strip Shingles	5
Fishing Line	12” to 14”

FINGERPRINT SCANNER:

Overview

For this system we have utilized the FPS GT-521F32 which is a small embedded module which consists of an optical sensor mounted on a small circuit board. The fingerprint scanner has the ability to enroll and identify a fingerprint as well as capability of 360° recognition. The optical sensor scans fingerprints while the micro-controller and software provide the module functionality to automatically processes the scanned fingerprint.

Even though, the basic interface of the FPS GT-521F32 consists of only four pins: power, ground, serial transmit and serial receive; it can hold up to 200 fingerprints.

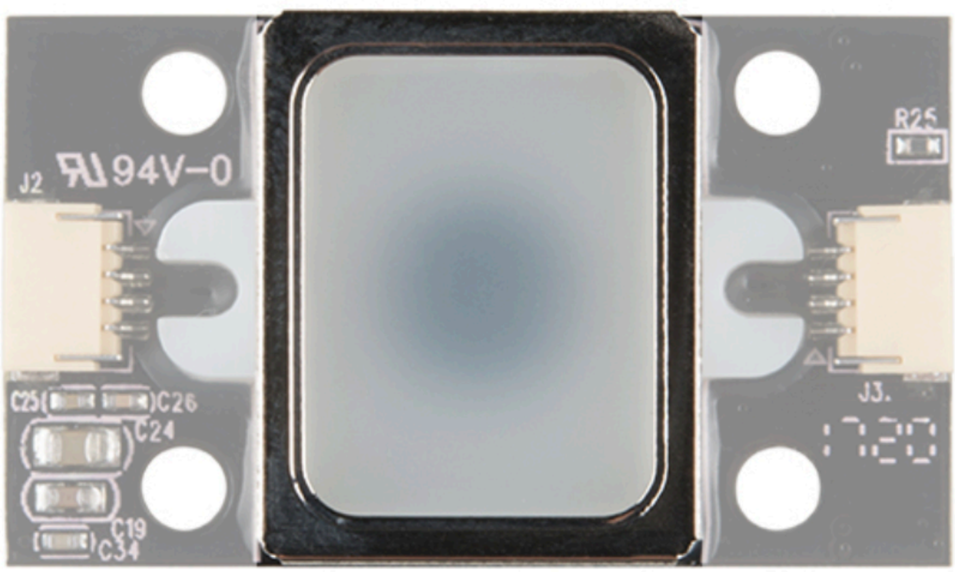


Figure 2 *GT-521F32's Optical Sensing Area*

Touch State	ICPCK Pin Status
Finger Initially Touching the Frame	LOW => HIGH
No Finger Touching	LOW => LOW
Finger Touching the Frame	HIGH => HIGH
Removing a Finger From the Frame	HIGH => LOW

Table 1 *Touch State and ICPCK Pin Status*

Enrolling

The FPS GT-521F32 has the ability to sense if a finger is placed on the optical sensing area. Upon contact with the metal frame around the optical sensing area, the ICPCK will output 3.3V (HIGH). Otherwise, the ICPCK will be 0V (LOW). For more information on the ICPCK pin status upon change in the touch state, see *Table 1*.

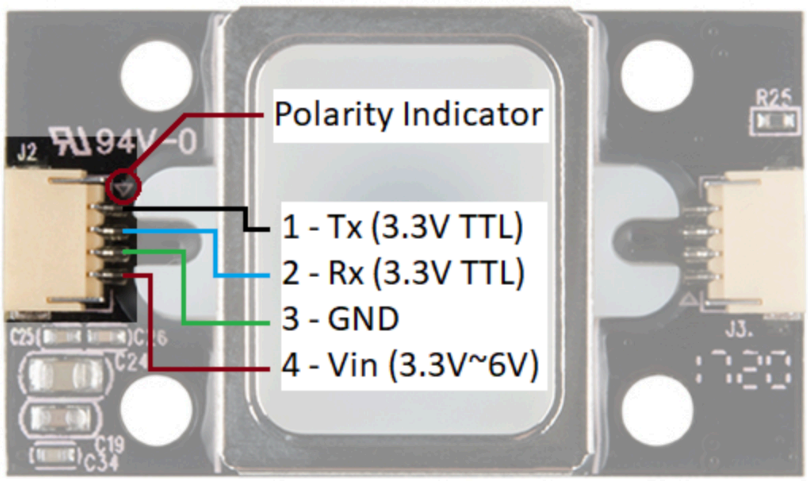


Figure 3 *Hardware of the Fingerprint Scanner (GT-521F32)*

To enroll a fingerprint to the database, the user must enroll their fingerprint three times for each authorizable ID before the Scanner can save it as a template. The white LED on the Scanner indicates the sensor is ready to scan a fingerprint:

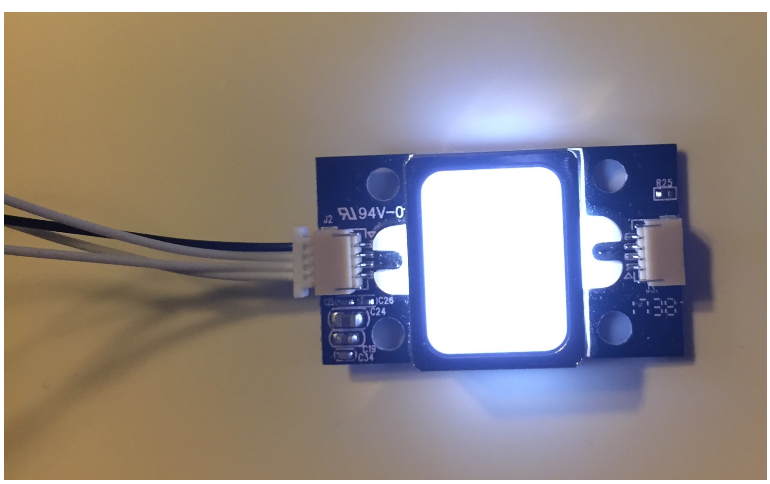
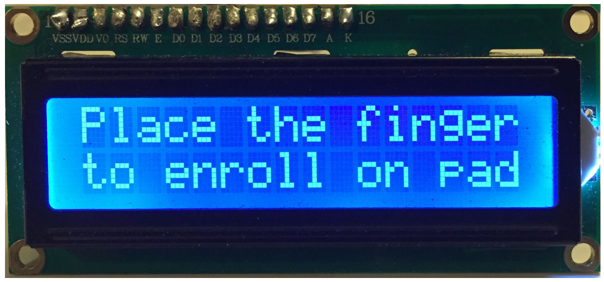


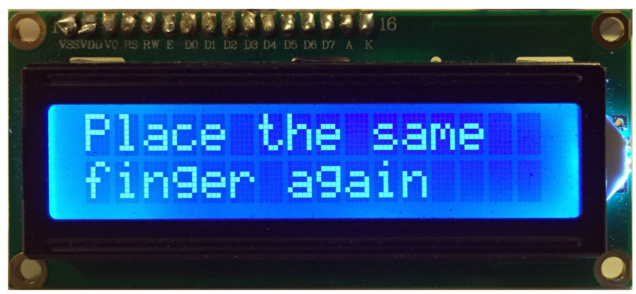
Figure 4 *Activated FPS GT-521F32*

To enroll a fingerprint:

1. Select a fingerprint that has no fingerprint template stored in the database.
2. Press and hold the enroll button for 3 seconds. The LCD screen will output "Place the finger to enroll on pad"

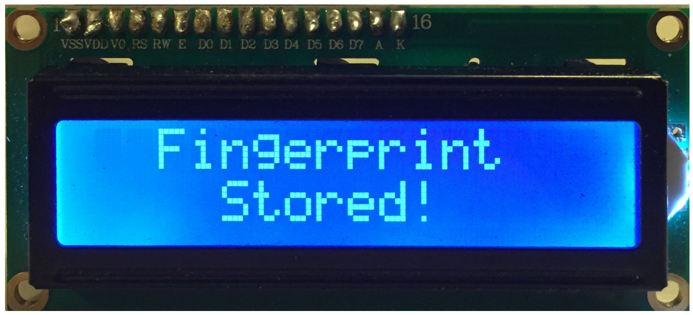


3. Place and remove your finger when LCD screen displays "Remove finger…" until the LCD screen reads "Place the same finger again"



4. Again, place and remove your finger when LCD screen displays "Remove finger…" until the LCD screen reads "Last time…place finger"

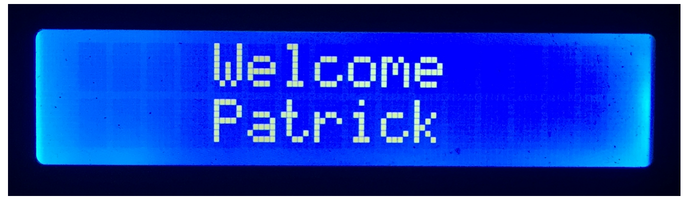
5. A notification will be provided on the LCD screen stating, “Fingerprint Stored!” when you have enrolled a fingerprint successfully.



If the Scanner is unable to recognize a unique fingerprint or if an error occurs during the enrollment process, the FPS GT-521F32 will terminate the enrollment process and print “Failed to store. Try again.” on the LCD screen. For best results, make sure that there is sufficient contact with the Scanner and that the finger is placed in the same position during enrollment.

Identifying

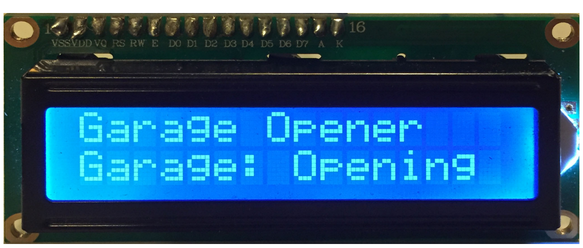
After enrolling a fingerprint to the database, you will want to test to confirm that the stored fingerprint can be identified. To test and verify, place the enrolled fingerprint on the optical sensing area of the Fingerprint Scanner. Once the white LED backlight is lit up, the sensor on the Scanner is ready to scan a fingerprint for authorization. Upon success the LCD screen will output a welcome message which can be customized.

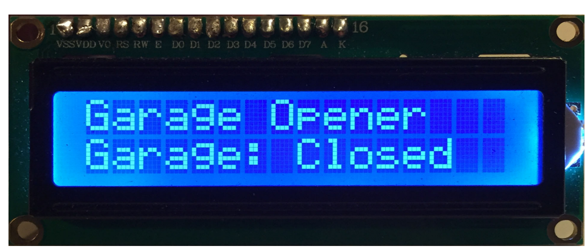


When the Scanner encounters a fingerprint that is not authorized, it will print:

Access denied usually means that the fingerprint does not match any of the template IDs in the database or when the scanner is not able to clearly read the fingerprint.

Once a finger has been placed on the Scanner, it will check through the fingerprint database to check if a match can be found against the saved fingerprint templates. Once the match is found, it will respond by sending a signal to circuit which will open or close the garage panel (based on the current status of the panel) and the LCD screen will print the following messages:





```
void Register_Finger() {
    int finger_ID_number = 0;
    bool ID_number_in_use = true;
    while (ID_number_in_use == true) {
        ID_number_in_use = fps.CheckEnrolled(finger_ID_number);
        if (ID_number_in_use == true) finger_ID_number++;
    }
    fps.EnrollStart(finger_ID_number);
    lcd_print("Place the finger", "to enroll on pad");
    while (fps.IsPressFinger() == false) delay(100);
    bool finger_capture_boolean = fps.CaptureFinger(true);
    int store_success_int = 0;
    if (finger_capture_boolean != false) {
        lcd_print("Remove finger...", "                ");
        fps.Enroll1();
        while (fps.IsPressFinger() == true) delay(100);
        lcd_print("Place the same  ", "finger again    ");
        while (fps.IsPressFinger() == false) delay(100);
        finger_capture_boolean = fps.CaptureFinger(true);
        if (finger_capture_boolean != false)
        {
            lcd_print("Remove finger...", "                ");
            fps.Enroll2();
            while (fps.IsPressFinger() == true) delay(100);
            lcd_print("Last time...    ", "place finger    ");
            while (fps.IsPressFinger() == false) delay(100);
            finger_capture_boolean = fps.CaptureFinger(true);
            if (finger_capture_boolean != false) {
                lcd_print("Remove finger...", "                ");
                store_success_int = fps.Enroll3();
                if (store_success_int == 0) {lcd_print("  Fingerprint   ", "     Stored!    ");}
                else {lcd_print("Failed to store.", "   Try again.   ");}
            }
            else {lcd_print("Failed to store.", "   Try again.   ");}
        }
        else {lcd_print("Failed to store.", "   Try again.   ");}
    }
    else {
        lcd_print("Failed to store.", "   Try again.   ");
    }
    delay(3000);
}
```

Figure 5 *Code Snippet of the Enroll Function*

GARAGE MODEL:

To test and demonstrate the critical elements of the Fingerprint Garage Door Opener, we have constructed a prototype of a garage. The skleteon of the prototype has been designed by using smooth particle boards cut 12" x 16" x 10" to form the four walls of the garage.

Figure 6 *Skeleton of Garage Model*

A sloped ramp has been constructed on the adajacent sides of the base structure to facilitate the sliding motion of the garage door panel.

Figure 7 *Ramp for the Garage Door*

The model incorporates two stepper motors and two Stepper Motor Driver Boards hinged between the sides of a wooden roller; this will allow the motors to recede and advance the garage door on the ramp by rotating in a clockwise or a counter-clockwise motion when a valid fingerprint is encountered. The garage door panels have been fashioned out of sheet metal, cut into four 12" x 2" rectangles. Five wooden dowels with *Hot Wheels* wheels attached on both ends have fasten to the sheet metal by screws and hinges. This will assist the garage panel slide smoothly on the ramp. We have wrapped 20 lb. test fishing wire around the wooden roller and overhead of the garage door panel, which outwardly lifts the panel upwards and downwoards.

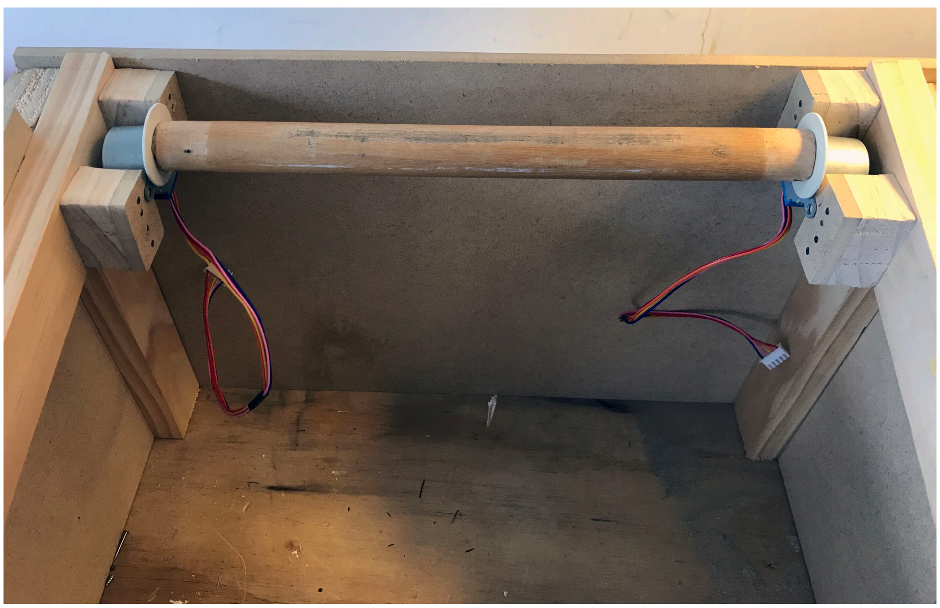


Figure 8 *Addition of a Wooden Roller to Mediate Garage Door Movement*

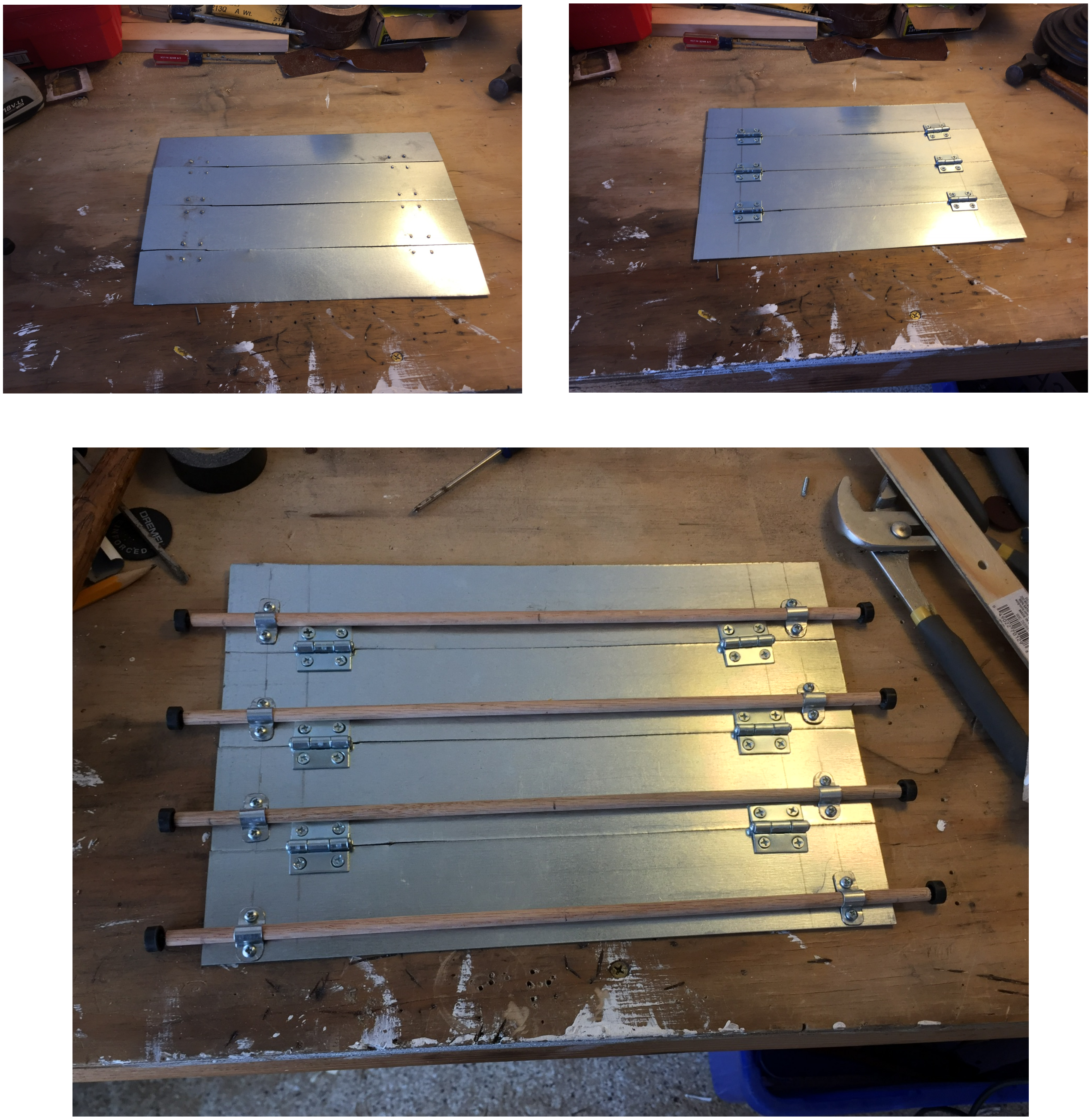


***Figure 9** Sheet Metal for the Garage Panel*

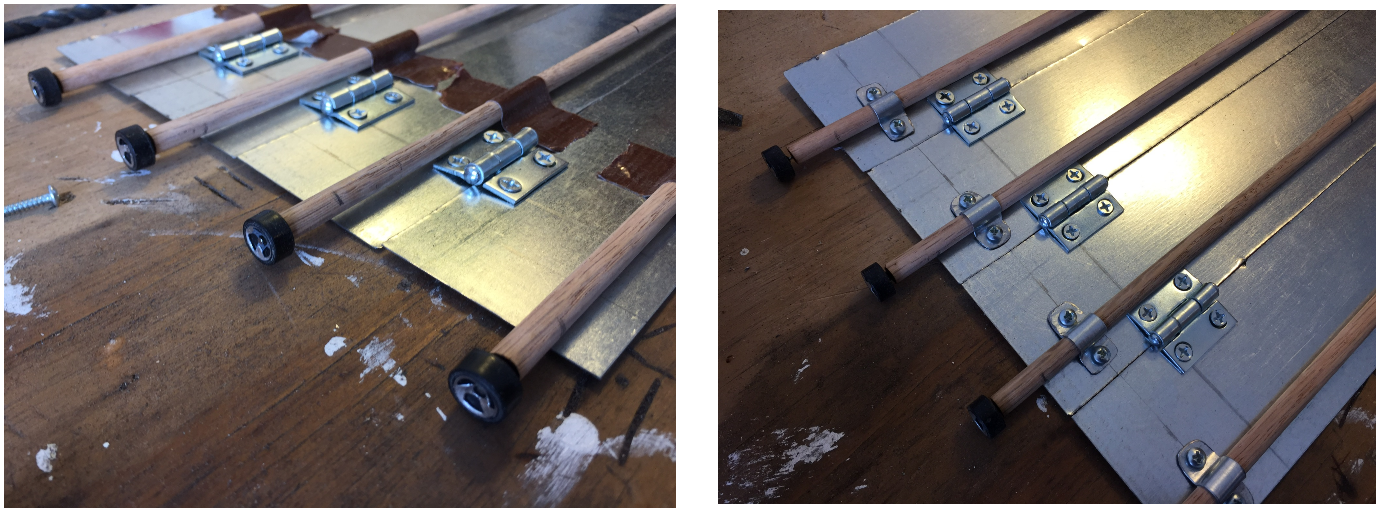


Figure 10 *Dowels and Hinges attached to the Panel*

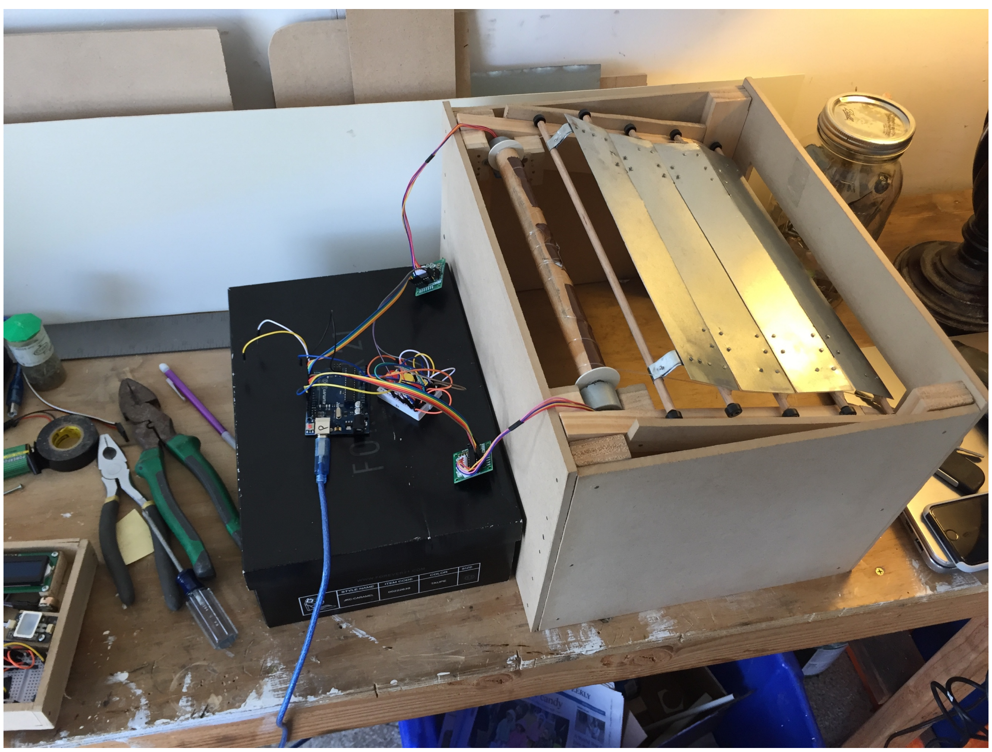


Figure 11 *Attachment of the Panel to the Motors*

In order to attach the panel to a wooden roller, we used 20" of fishing line rolled aroud the roller to recede and advance the panel on the ramp as the stepper motors step clockwise and counter-clockwise. After the attachment of the garage panel to the stepper motors was complete, we modified the GarageMotorRunner.ino to work in conjuction with the garage model.

The GarageMotorRunner.ino runs two stepper motors simultaneously stepping in the opposite directions with respect to the wooden roller of the garage model. Placed on opposite sides of the wooden roller, the opposing rotations of the motors results in the closing and opening of the garage panel. When an open or close sequence is initiated by the GarageFingerprint.ino module, each motor rotates exactly 6100 steps to open or close the garage panel; this amounts to about 4 full rotations of the wooden roller. Also, it is important to note that the simultaneous movements of the respective motors are an illusion. A motor steps exactly once, then the other motor steps exactly once and this movement switches back and forth until each motor has completed exactly 6100 steps in total.

Additionally, the GarageMotorRunner.ino also handles input from the three admin pushbuttons. The first and second buttons handle the manual opening and closing of the garage panel. The third button handles the fingerprint enrollment process of a new fingerprint, when held for more than 3 seconds and less than 10 seconds. Finally, this button is also responsible for clearing all fingerprints from the database when held for more than 10 seconds.

In order to conceal the circuit without interrupting the connection with the stepper motors we constructed a compartment above the model for the circuit to reside securely (see Figure 12).

Figure 12 *Construction of the Ceiling Compartment*

Another smooth particle board rectangle was cut out to make a ‘ceiling’ for the garage model. The main purpose of this addition is to compartmentalize the circuit mechanisms to reside in. The added board had to be built with a little opening above the floor structure of the gargae to allow flexible space for the panel to move up and down (see Figure 13). This opening also allows the Stepper Motor Boards wires to communicate with stepper motors effectively.

Figure 13 *Close Up of Gap between Ceiling and Framework*

The ceiling of the structure houses the circuit components of the system including the breadboard, UNO R3(s), manual buttons and two stepper motors driver boards as displayed in Figure 14.

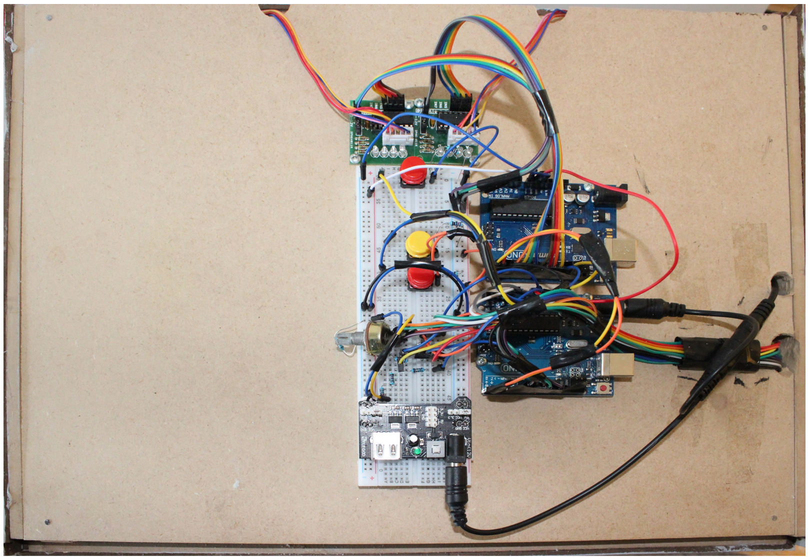


Figure 14 *Contents of the Circuit Compartment*

Figure 15 *Left/Right Enclosure for the Roof*

Next, two triangles as well as two rectangles cutouts were carved out of smooth particle board to enclose the ceiling section of the module. This 'roof' structure will live on top of the celing structure surrounding the 'circuit' comparment.

Figure 16 *Structure of the Garage Model*

The completed structue of the garage model frame is depicted in Figure 16. After the initial construction, the code and circuit were put through rigorous testing to evalutate their compatibility with the new framework and it's components. The main purpose of the testing phase is to verify and validate the software component of the system meets the structural component of the system practically.

Decoration

Figure 17 *Side View of Garage Model*

For the four walls of the garage, a beadboard cut in 1 ½ inches increments and hot glued to the structure to create a 'ripple' effect to mimic the siding shingles of a real-life garage structure.

Figure 18 *After Attachment of Beadboard Siding*

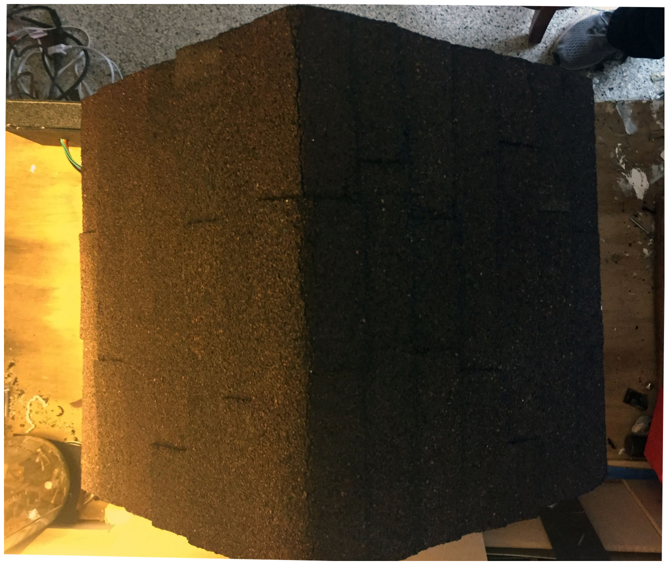


Figure 19 *Aerial View of Roof Shingles*

For the roof of the structure, strip shingles were cut in 6” x 4” and hot glued to the top of the roof. After the shingles and the beadboard were secured on structure we put painter’s tape on the roof shingles to protect from getting paint on them for the painting process.

Figure 20 *Painter's Tape on Roof Shingles*

Although the purpose is mainly aesthetic and decorative, adding a color to the structure draws attention to it and makes it more noticable. After final touches, the completed garage model is depicted in Figure 21.

Figure 21 *Garage Model*

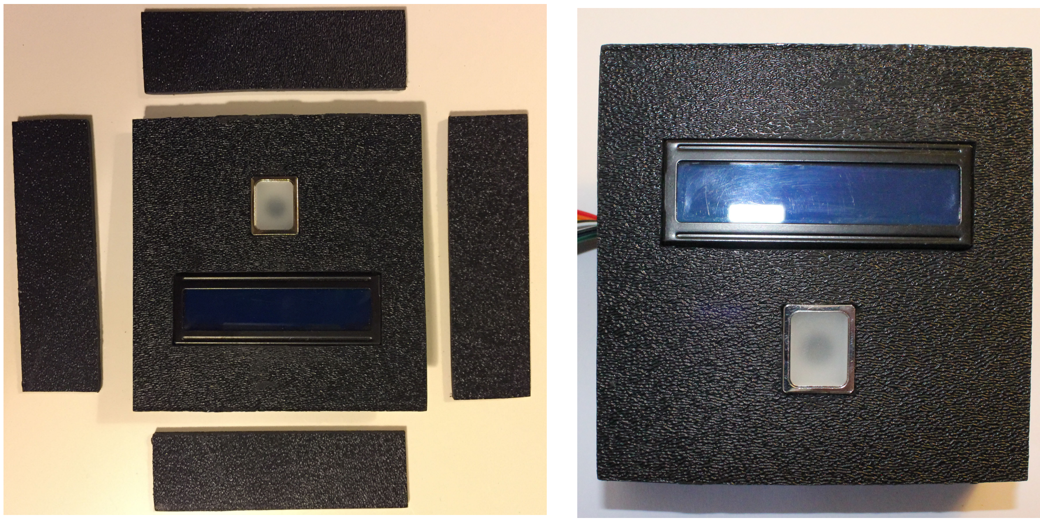


Figure 22 *Shell of the Compartment*

Next, to enclose the LCD screen and the Fingerprint Scanner another compartment has been created. The main purpose of this compartment is to be placed in the front of the garage and act as the user interface for conveniently interacting with the system. Also, this compartment stimulates the real setting of a garage, where users enter their credientials at a keypad in front of the garage.

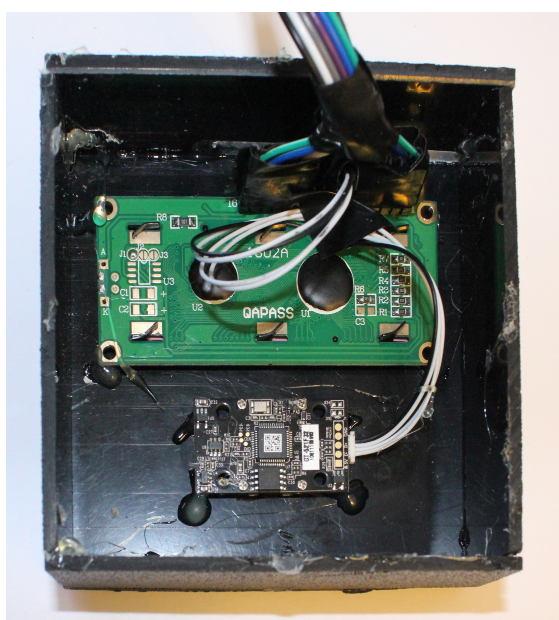


Figure 23 *Internal Components of the Compartment*

CONNECTION BETWEEN THE FINGERPRINT SCANNER AND GARAGE MODEL

To connect the FPS GT-521F32 to the garage prototype we have utilized two UNO R3 development boards, three key switches, one LCD screen, six resistors, ~25 jumper wires, two ULN2003 stepper motor Board and stepper motors.

The LCD screen is connected to an UNO R3 #1 and the breadboard which enables the user to read the display and follow the instructions efficiently.

The functionality of the two pushbuttons and the two stepper motors, connected to UNO R3 #2, is to turn the stepper motors clockwise and counter-clockwise, allowing for a manual opening and closing process of the garage door panel if needed (for trimming purposes). Moreover, the third pushbutton is crucial to FPS-GT521F32 module in terms of resetting the fingerprint database and enrolling a new fingerprint. This pushbutton resets the LCD screen and allows us to enroll a new fingerprint when pressed for more than 3 seconds and less than 10 seconds and clears all the fingerprints in the database when pressed for longer than 10 seconds.

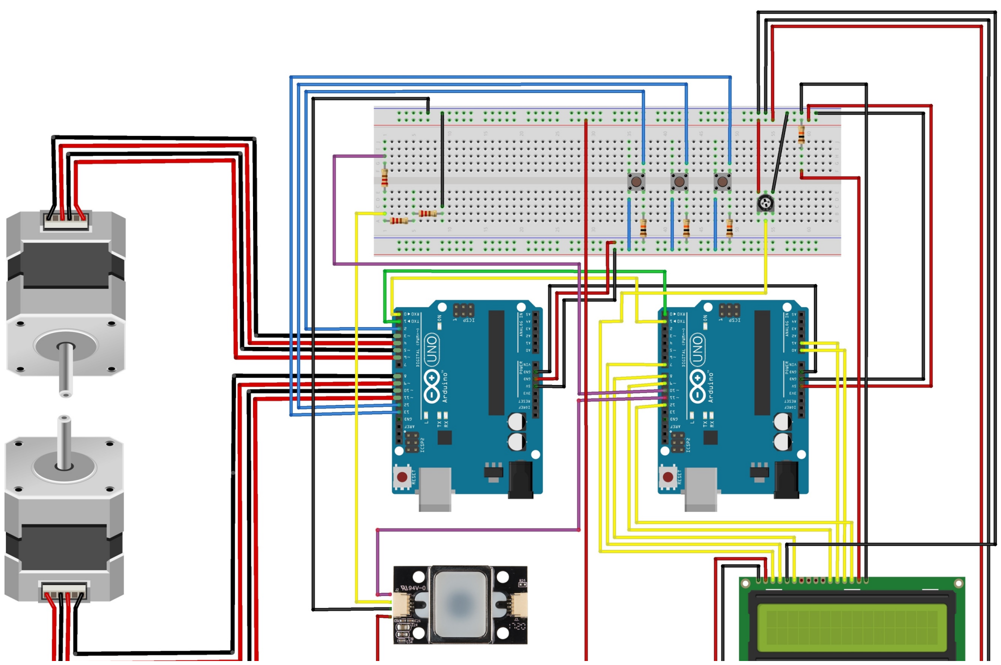


Figure 24 *Schematic of the Circuit*

DISCUSSION

This project contains highlights of what we learned this semester by incorporating pushbuttons, an LCD screen and micro-controller communication, while exploring new realms with the Fingerprint Scanner Module and stepper motors. With the knowledge of circuit design, we planned to create a system that can be applicable in real life.

Throughout the course of this project, we were able to closely design and build the system according to our own exact specifications. We were able to explore the possibilities of adding materials to our system in order to evaluate the capacity of our project. In effect, we were able to learn the importance of project management in terms of what is possible and not possible in terms of time, cost and complexity. We appreciated the freedom of having to design our own project as we encountered our own problems and explored different possibilities when resolving them. For example, we ran into multiple conflicts when coming up with a stable power source for our system.

Originally, we attempted to use a 9V wall plug-in charger to power both the garage motor runner module and the fingerprint module. The 9V charger power supply was split via a power splitter and then connected to each respective UNO R3. We noticed that the built-in linear voltage regulator on the UNO R3 responsible for running the two stepper motors seemed to be getting quite hot-to-the-touch. We surmised that this was due to the 9V/5V disparity from the wall charger the garage system. We then decided that we would purchase a 5V plugin power source to try to run both UNO R3s, however, this power supply was not enough power to run the stepper motors and lift the garage door. After further research, we learned that the built-in voltage regulators are built to run at temperatures of up to 125 degrees Celsius. This eased our concerned about frying the UNO R3, but just to be safe we decided to run the power for the stepper motors through the voltage regulator on a 3.3/5.0V power supply adapter to avoid the destruction of the built-in voltage regulator. In a perfect world, we would have bought another plugin power adapter that ran at a voltage somewhere between 5 and 9 volts and that still successfully performed the heavy lifting via the stepper motors, but we did not feel that it would be a necessary expenditure.

Through this project, we have come to learn a great detail about the FPS GT-521F32 and the stepper motors. And even though we still have a lot to learn about both of these mechanisms and all other UNO R3 components, this project has allowed us to perceive the scope of applications that can be developed using open source hardware.

REFERENCE:

Coro, Noah Del. "DIY Fingerprint Scanning Garage Door Opener." *Instrutables.com.* October 19, 2017. Web. April 4, 2018. https://goo.gl/6S6ESL

Demay, Vincent. "Building a garage door fingerprint lock with Arduino." *homautomation.org.* August 7, 2015. Web. April 4, 2018. https://goo.gl/xsPfjN

Chan, Bobby. "Fingerprint Scanner (GT-521Fxx) Hookup Guide." *learn.sparkfun.com.* Web. April 4, 2018. https://goo.gl/ZzYMUs

"10 Ways to Destroy An Arduino" *rugged-circuits.com.* Web. April 8, 2018. https://goo.gl/GgCYqz