Fingerprint Scanner Garage Door System

Jahnvi Patel, Patrick O'Connell

Abstract:

In this paper, we introduce a Fingerprint Scanner Garage Door System with a model of a garage 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 the data identifying an authorized user. On subsequent operation, once a fingerprint data set is received and an attempt is made to match the incoming data with the previously stored local database 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 data and activates the open/close 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 and open or close 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, send the received information to the UNO R3 to decode it from valid or invalid. If valid ID is encountered, print a message, open the garage door and time out. If invalid, print an error message and time out. 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.

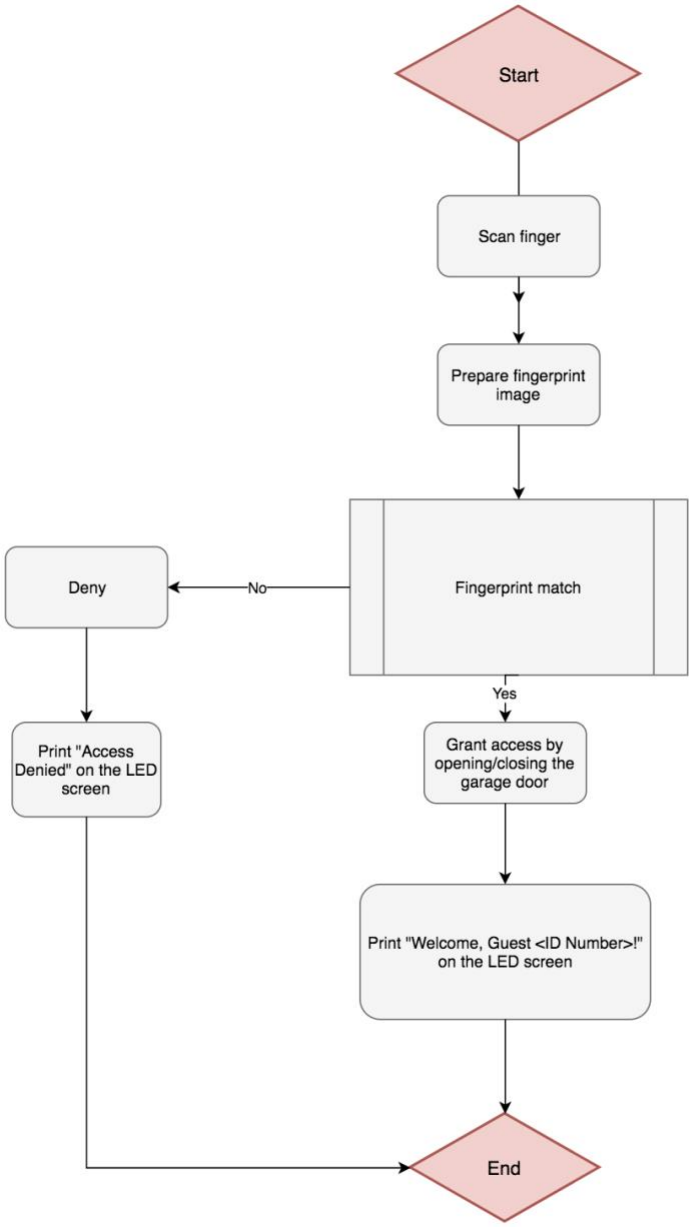

Figure 1 *The block overview of Fingerprint Scanner Garage Door System*

This system provides an automatically operable garage door security system which positively blocks the unauthorized opening or closing of a garage door; that travels upward and downward guided by a supporting roller powered two 5V Stepper Motors.

The system works in combination with a standard automatic electric control circuit that drives the door upward or downward when the Stepper Motor is actuated in response to 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
Smooth Particle Board	5
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

Fingerprint Scanner:

Overview

For this system we have utilized the FPS GT-521F32 which is a small embedded module that 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 a fingerprint and the microcontroller and software provides the modules functionality which automatically processes the scanned fingerprint.

Even though, the interface of the FPS is very basic consisting of only four pins – power, ground, serial transmit and serial receive, FPS GT-521F32 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 GT-521F32 have 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, see *Table 1*.

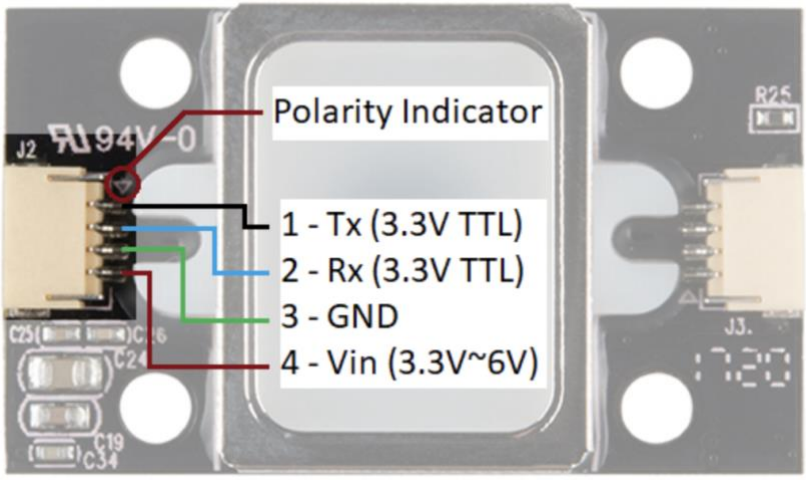

Figure 3 *Polarity Indicator and ICPCK Pins*

To enroll a fingerprint to the module, user would need to enroll their finger three times for each ID before the scanner can save it as a template. The white LED will light up to begin reading your fingerprint:

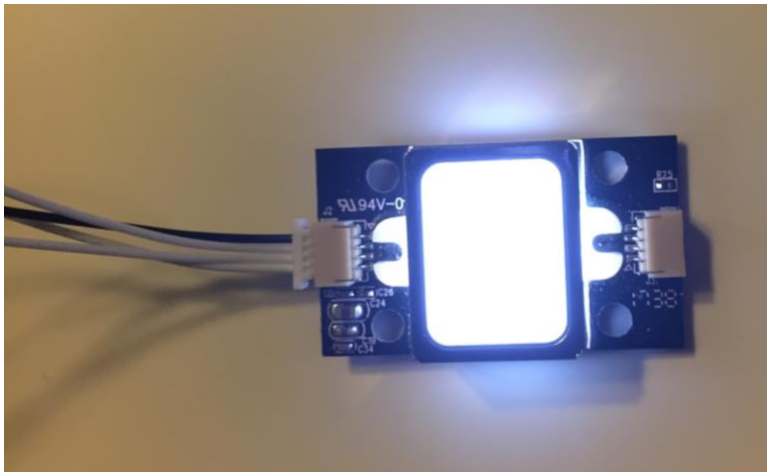

Figure 4 *Activated FPS GT-521F32*

To enroll a fingerprint:

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

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

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

5. A notification will be provided on the LED screen stating, "Fingerprint Stored!" when you have enrolled your fingerprint successfully.

If the scanner is unable to recognize a unique fingerprint or there is a any error in enrollment, the FPS GT-521F32 will stop the enrollment and process and print error message "Failed to store. Try again." to the screen. For best result, 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, you will want to test to see if the fingerprint can be identified. To test and verify, place the enrolled fingerprint on the optical sensing area of the fingerprint scanner. The white LED will have lit up and the LED screen will print "Welcome."

When the scanner fails to properly enroll the fingerprint:

Once a finger has been placed on the scanner, it will check the fingerprint through the local database to see if it can recognize fingerprint against the saved fingerprint templates. If successful, it will respond by sending a signal to circuit which will open/close the garage and the LED screen will print the following messages:

Looking at the output, "Access Denied" usually means that the fingerprint does not match any of the template IDs in the local database or when the scanner is not able to clearly read the fingerprint. If the finger has been enrolled, you would need to make sure that you place the fingerprint on the scanner just like when you scanned the finger.

```
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 Scanner Garage Door Opener, we have started 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 prototype 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 and/or a counter-clockwise motion when a valid fingerprint is encountered. The garage door panel have been fashioned out of four sheet metals cut in 12" x 2" rectangles. Four 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 plan to wrap a thick wire around the wooden roller and overhead of the garage door panel, which will outwardly lift the panel upwards and downwoards.

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

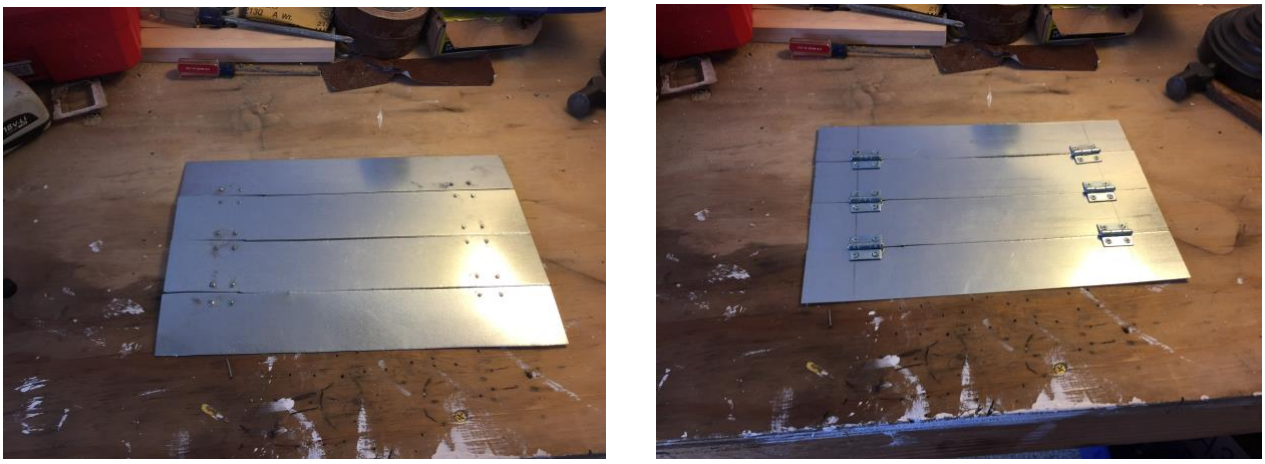

Figure 9 *Sheet Metal for the Garage Panel*

Figure 10 *Dowels and Hinges attached to the Panel*

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 LED screen, six resistors, ~25 jumper wires, two ULN2003 Stepper Motor Board and Stepper Motors

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

The functionality of the two key switches and the two Stepper Motors connected to UNO R3 #2 and the breadboard, 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. Moreover, the third Key Switch is crucial to FPS-GT521F32 module; this key switch resets the LED screen and allows us to enroll a new fingerprint when pressed for 3 seconds.

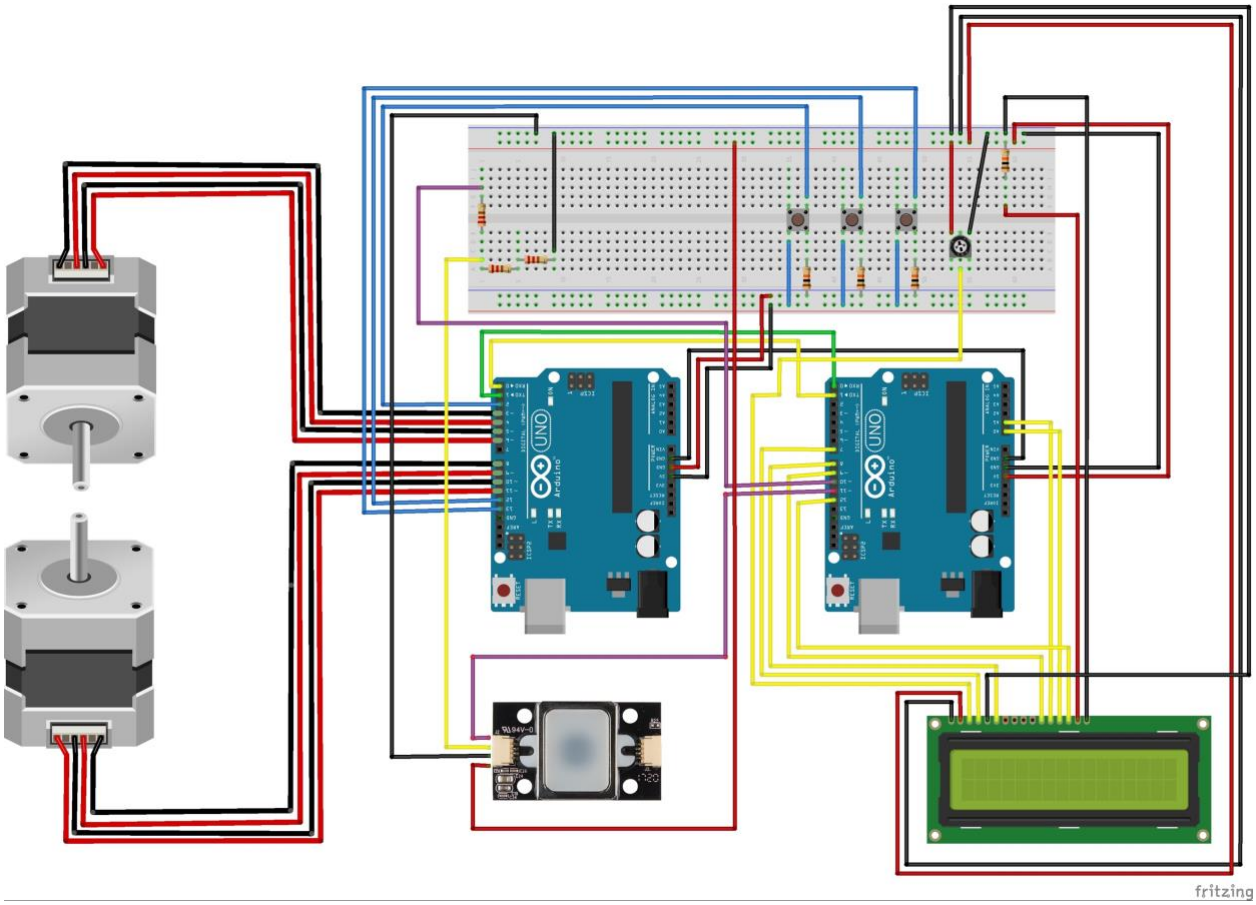

Figure 11 *Schematic of the Circuit*

Upcoming for the Project:

Even though we have a working code base that provides the basic implementation to the garage prototype, we still have other essential tasks to integrate. For example, a reliable power source is needed to power the LED screen, Stepper Motors and other crucial aspects of the circuit. The garage door panel needs to be put in place to ensure the receding and advancing phase of the project are in a working order. A compartment needs to be built for the development boards and electrical wires to be conceal in. The LED screen and fingerprint sensor need to be accessible to the user in the front of the model for ease of access. Also, we plan to paint the exterior of the prototype for the purpose of obtaining a smooth attractive finish. We will complete these proposed developments and expansions for the final report.

Stay tuned for more updates!

References:

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