

## 2<sup>nd</sup> Y.B.S.C-IT PROJECT

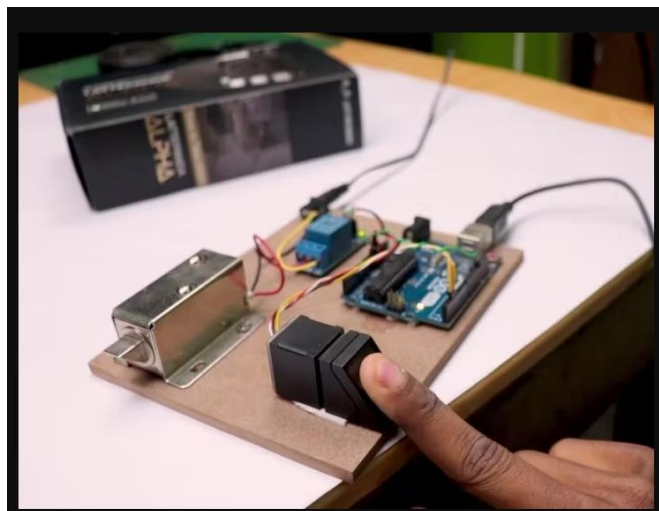
**Name:- Mudassir Raza**

**Project Title:- Make a Fingerprint door Lock by using Arduino uno.**

---

### ❖ INTRODUCTION:-

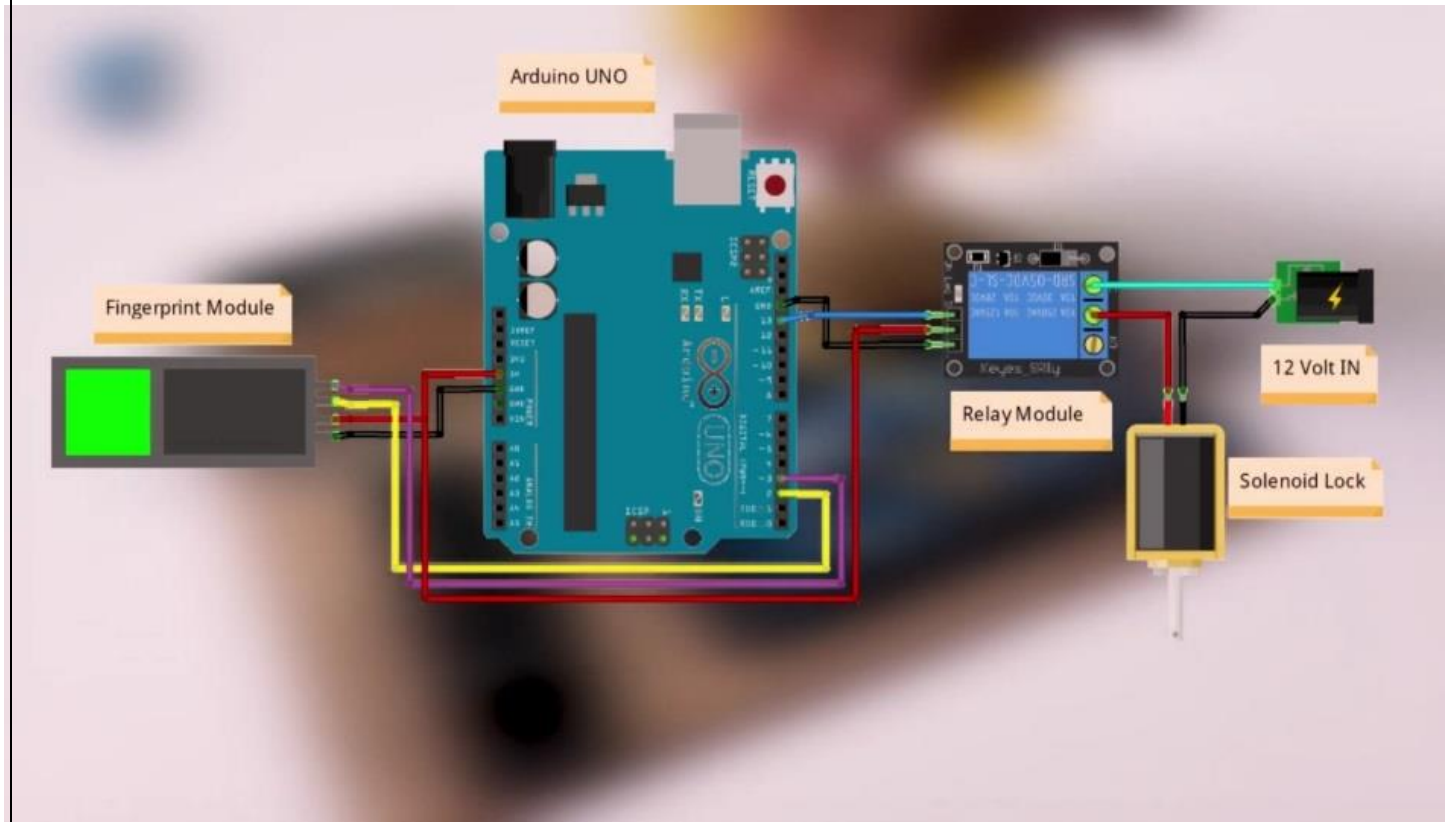
By using embedded system we are making a fingerprint Sensor door lock. This door lock will only open the door when the user scans the right fingerprint that is recorded on the system, but the door will remained close upon entering the wrong fingerprint. Fingerprint reader scanning is the most mature and tested type of biometric technology. It provides a good security system. Fingerprint reader scanning is the most mature and tested type of biometric Technology. A much broader application of fingerprint is for personal authentication, for instance to access a computer, a network, an ATM machine, a car or a home. Fingerprint reader scanning is the most mature and tested type of biometric technology. Biometric security guarantees a positive method of user identification with something that cannot be lost, replicated or stolen.it is a system that allows authorized people access as a result of recognizing their fingerprints, as they are a great method of identification for humans.



## ❖ HOW FINGERPRINT SENSOR MODULE WORK :-

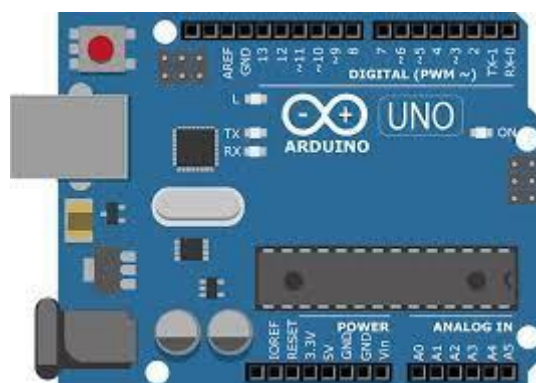
The way this optical fingerprint sensor works is that it captures a photo of our finger ridges, and then it uses certain algorithm to match it with stored data and displays result of the same.

## ❖ FUNCTION DIAGRAM :-

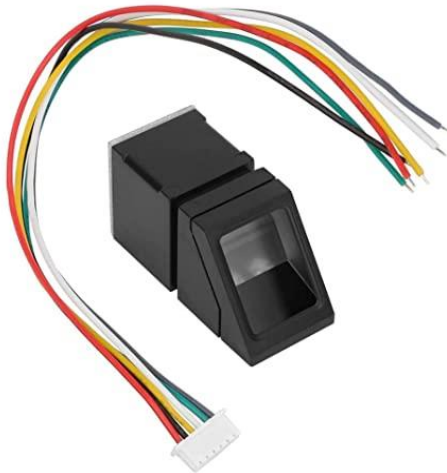


## ❖ COMPONENTS ARE USE :-

1.Arduino UNO:



2.Fingerprint Sensor R307:



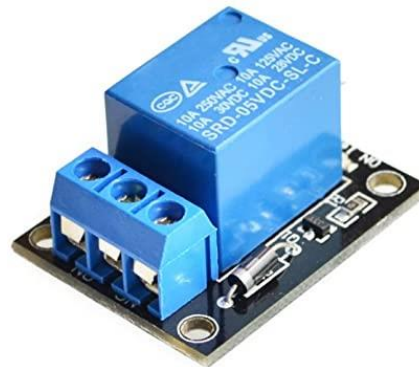
3.Jumper Wires:



4.Solenoid Lock :



5.Relay:



6.12V Adapter



7. Battery :



## //CODE OF ENROLL

```
#include <Adafruit_Fingerprint.h>
#if defined(__AVR__) || defined(ESP8266)
// For UNO and others without hardware
serial, we must use software serial...
// pin #5 is IN from sensor (WHITE wire)
// pin #6 is OUT from arduino (YELLOW
wire)
// Set up the serial port to use
softwareserial..
SoftwareSerial mySerial(6, 5);
#else
// On Leonardo/M0/etc, others with hardware
serial, use hardware serial!
// #0 is green wire, #1 is white
#define mySerial Serial1
#endif
Adafruit_Fingerprint finger =
Adafruit_Fingerprint(&mySerial);
uint8_t id;
void setup() {
Serial.begin(9600);
while (!Serial); // For
Yun/Leo/Micro/Zero/...
delay(100);
Serial.println("\n\nAdafruit Fingerprint
sensor enrollment");

// set the data rate for the sensor serial
port
finger.begin(57600);
```

```

if (finger.verifyPassword()) {
Serial.println("Found fingerprint
sensor!");
}
else {
Serial.println("Did not find fingerprint
sensor :(");
while (1) { delay(1); }
}
Serial.println(F("Reading sensor
parameters"));
finger.getParameters();
Serial.print(F("Status: 0x"));
Serial.println(finger.status_reg, HEX);
Serial.print(F("Sys ID: 0x"));
Serial.println(finger.system_id, HEX);
Serial.print(F("Capacity: "));
Serial.println(finger.capacity);
Serial.print(F("Security level: "));
Serial.println(finger.security_level);
Serial.print(F("Device address: "));
Serial.println(finger.device_addr, HEX);
Serial.print(F("Packet len: "));
Serial.println(finger.packet_len);
Serial.print(F("Baud rate: "));
Serial.println(finger.baud_rate);
}

uint8_t readnumber(void) {
uint8_t num = 0;

while (num == 0) {
while (! Serial.available());
num = Serial.parseInt();
}
}

```

```

    }
    return num;
}

void loop() {                                     // run
over and over again
    Serial.println("Ready to enroll a
fingerprint!");
    Serial.println("Please type in the ID #
(from 1 to 127) you want          to save
this finger as...");
    id = readnumber();
    if (id == 0) { // ID #0 not allowed, try
again!
        return;
    }
    Serial.print("Enrolling ID #");
    Serial.println(id);

while (!  getFingerprintEnroll() );
}
uint8_t getFingerprintEnroll() {
    int p = -1;
    Serial.print("Waiting for valid finger to
enroll as #");          Serial.println(id);
    while (p != FINGERPRINT_OK) {
        p = finger.getImage();
        switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Image taken");
            break;
        case FINGERPRINT_NOFINGER:
            Serial.println(".");
            break;

```

```

case FINGERPRINT_PACKETRECEIVEERR:
Serial.println("Communication error");
break;
case FINGERPRINT_IMAGEFAIL:
Serial.println("Imaging error");
break;
default:
Serial.println("Unknown error");
break;
}}
// OK success!
p = finger.image2Tz(1);
switch (p) {
case FINGERPRINT_OK:
Serial.println("Image converted");
break;
case FINGERPRINT_IMAGEMESS:
Serial.println("Image too messy");
return p;
case FINGERPRINT_PACKETRECEIVEERR:
Serial.println("Communication error");
return p;
case FINGERPRINT_FEATUREFAIL:
Serial.println("Could not find fingerprint
features");
return p;
case FINGERPRINT_INVALIDIMAGE:
Serial.println("Could not find fingerprint
features");
return p;
default:
Serial.println("Unknown error");
return p;

```

```

}
Serial.println("Remove finger");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
p = finger.getImage();
}
Serial.print("ID "); Serial.println(id);
p = -1;
Serial.println("Place same finger again");
while (p != FINGERPRINT_OK) {
p = finger.getImage();
switch (p) {
case FINGERPRINT_OK:
Serial.println("Image taken");
break;
case FINGERPRINT_NOFINGER:
Serial.print(".");
break;
case FINGERPRINT_PACKETRECEIVEERR:
Serial.println("Communication error");
break;
case FINGERPRINT_IMAGEFAIL:
Serial.println("Imaging error");
break;
default:
Serial.println("Unknown error");
break;
}}
// OK success!
p = finger.image2Tz(2);
switch (p) {
case FINGERPRINT_OK:

```



```

Serial.println("Image converted");
break;
case FINGERPRINT_IMAGEMESS:
Serial.println("Image too messy");
return p;
case FINGERPRINT_PACKETRECEIVEERR:
Serial.println("Communication error");
return p;
case FINGERPRINT_FEATUREFAIL:
Serial.println("Could not find fingerprint
features");
return p;
case FINGERPRINT_INVALIDIMAGE:
Serial.println("Could not find fingerprint
features");
return p;
default:
Serial.println("Unknown error");
return p;
}
// OK converted!
Serial.print("Creating model for #");
Serial.println(id);
p = finger.createModel();
if (p == FINGERPRINT_OK) {
Serial.println("Prints matched!");
} else if (p ==
FINGERPRINT_PACKETRECEIVEERR) {
Serial.println("Communication error");
return p;
} else if (p ==
FINGERPRINT_ENROLLMISMATCH) {

```

```

    Serial.println("Fingerprints did not
match");
    return p;
}
else {
    Serial.println("Unknown error");
    return p;
}
Serial.print("ID "); Serial.println(id);
p = finger.storeModel(id);
if (p == FINGERPRINT_OK) {
    Serial.println("Stored!");
} else if (p ==
FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    return p;
}
else if (p == FINGERPRINT_BADLOCATION) {
    Serial.println("Could not store in that
location");
    return p;
}
else if (p == FINGERPRINT_FLASHERR) {
    Serial.println("Error writing to flash");
    return p;
}
else {
    Serial.println("Unknown error");
    return p;
}
return true;
}

```

## Main Code:-

```
#include <SoftwareSerial.h>
#include <Adafruit_Fingerprint.h>
SoftwareSerial mySerial(6, 5);
Adafruit_Fingerprint finger =
Adafruit_Fingerprint(&mySerial);
void setup() {
Serial.begin(9600);
pinMode(8, OUTPUT);
digitalWrite(8, HIGH);

while (!Serial); // For
Yun/Leo/Micro/Zero/...
delay(10);
Serial.println("\n\nAdafruit finger detect
test");

// set the data rate for the sensor serial
port
finger.begin(57600);
delay(5);
if (finger.verifyPassword()) {
Serial.println("Found fingerprint
sensor!");
}
else {
Serial.println("Did not find fingerprint
sensor :(");
while (1) { delay(1); }
}
Serial.println(F("Reading sensor
parameters"));
```

```

    finger.getParameters();
    Serial.print(F("Status: 0x"));
    Serial.println(finger.status_reg, HEX);
    Serial.print(F("Sys ID: 0x"));
    Serial.println(finger.system_id, HEX);
    Serial.print(F("Capacity: "));
    Serial.println(finger.capacity);
    Serial.print(F("Security level: "));
    Serial.println(finger.security_level);
    Serial.print(F("Device address: "));
    Serial.println(finger.device_addr, HEX);
    Serial.print(F("Packet len: "));
    Serial.println(finger.packet_len);
    Serial.print(F("Baud rate: "));
    Serial.println(finger.baud_rate);

    finger.getTemplateCount();
    if (finger.templateCount == 0) {
    Serial.print("Sensor doesn't contain any
    fingerprint data. Please run the 'enroll'
    example.");
    }
    else {
    Serial.println("Waiting for valid
    finger...");
    Serial.print("Sensor contains ");
    Serial.print(finger.templateCount);
    Serial.println(" templates");
    }
    }

    void loop() {                                     // run
    over and over again

```

```

getFingerprintID();
delay(50); //don't need
to run this at full speed.
}
uint8_t getFingerprintID() {
uint8_t p = finger.getImage();
switch (p) {
case FINGERPRINT_OK:
Serial.println("Image taken");
break;
case FINGERPRINT_NOFINGER:
Serial.println("No finger detected");
return p;
case FINGERPRINT_PACKETRECEIVEERR:
Serial.println("Communication error");
return p;
case FINGERPRINT_IMAGEFAIL:
Serial.println("Imaging error");
return p;
default:
Serial.println("Unknown error");
return p;
}
// OK success!
p = finger.image2Tz();
switch (p) {
case FINGERPRINT_OK:
Serial.println("Image converted");
break;
case FINGERPRINT_IMAGEMESS:
Serial.println("Image too messy");
return p;
case FINGERPRINT_PACKETRECEIVEERR:

```

```

Serial.println("Communication error");
return p;
case FINGERPRINT_FEATUREFAIL:
Serial.println("Could not find fingerprint
features");
return p;
case FINGERPRINT_INVALIDIMAGE:
Serial.println("Could not find fingerprint
features");
return p;
default:
Serial.println("Unknown error");
return p;
}

// OK converted!
p = finger.fingerSearch();
if (p == FINGERPRINT_OK) {
Serial.println("Found a print match!");
delay(10);
digitalWrite(8, LOW);
delay(5000);
digitalWrite(8, HIGH);
} else if (p ==
FINGERPRINT_PACKETRECEIVEERR) {
Serial.println("Communication error");
return p;
} else if (p == FINGERPRINT_NOTFOUND) {
Serial.println("Did not find a match");
return p;
} else {
Serial.println("Unknown error");
return p;
}

```

```

}

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

// 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;
}

```

### ❖ CONCLUSION:-

For security propose fingerprint place an important role in human recognition from past years and biometric system only be present at the recent years. For the development of fingerprint standards, government and other industries had done

developments on fingerprint techniques. This development over the highly quality products and faster use of devices and improve the reliability on fingerprint recognition system. Where this technique is mainly used for the government legal methods and investigation propose and science community developments and these are mainly useful for the biometrics. Behind this development we have so many reasons because biometric is not cure all security identifications. In this paper we discussed the overview of fingerprint identification and techniques we are using in fingerprint for recognition and also we discussed the how it is use for the biometric system. For determination of fingerprint industry government and other industries will done led on coming generation for fingerprint identification.