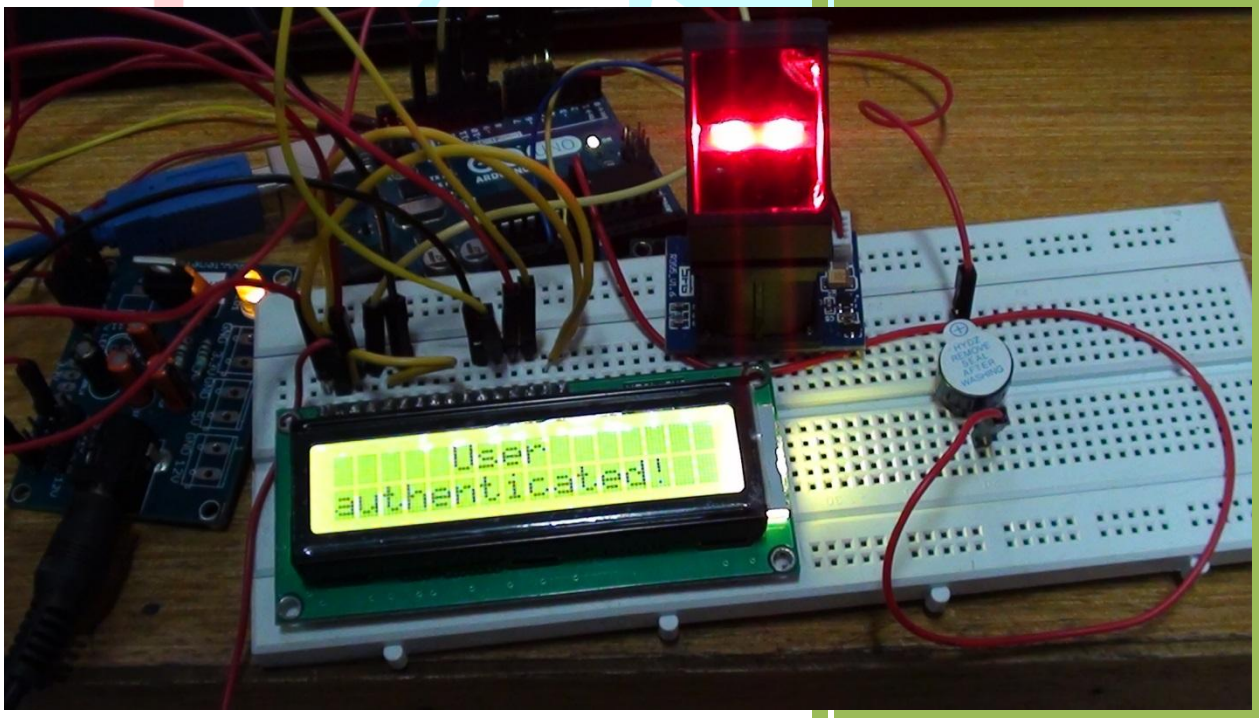
TENET TECHNETRONICS

2014

Security System Based On Biometric Finger Print Scanner with Arduino.



"Simplifying

Author: Mr. Prajwal R

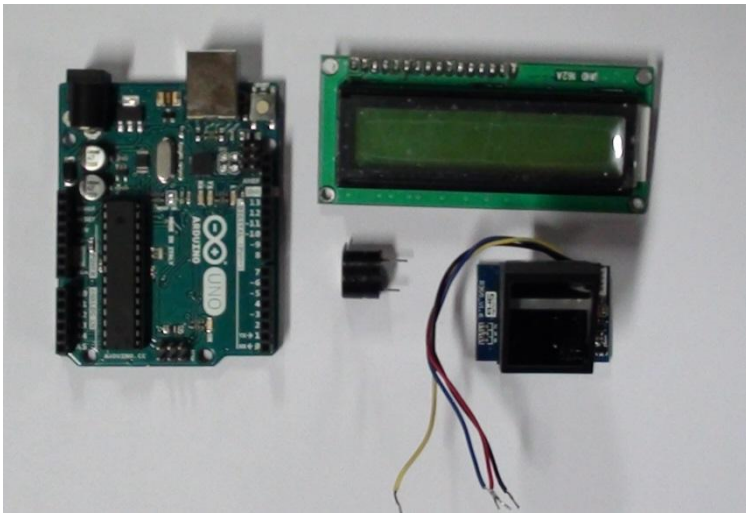
Introduction

Nothing is absolutely secure in security systems. Locks can be picked, safes can be broken into, and online passwords& PIN codes can be guessed sooner or hacked or once a password is known it can be used by anyone. How, then, can we protect the things that we value? One way is to use biometrics fingerprints, since every fingerprint is special, and different from any other in the world. Because there are countless combinations, and also cannot be duplicated without extensive preparation.so fingerprints have become an ideal means of identification.

Here is a quick demo on how to create a Security system based on finger print scanner with Arduino.

HardwareRequired:

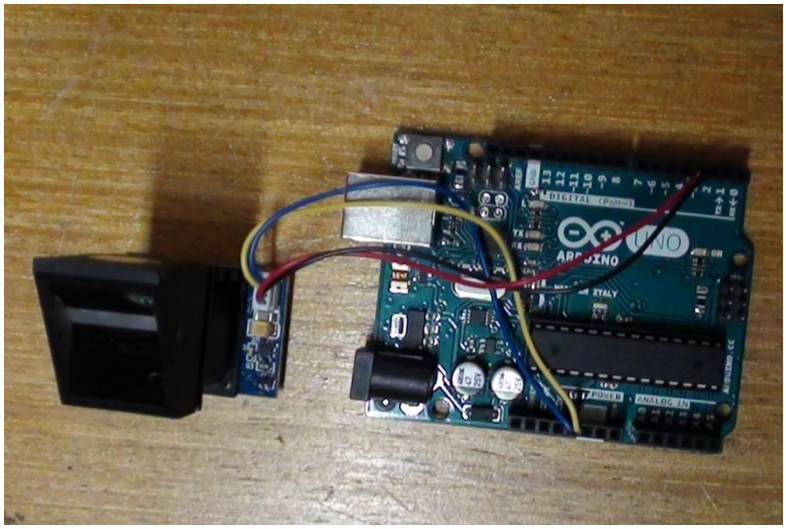
- Arduino Uno board.
- Arduino type B USB cable.
- Finger print Scanner (R305).
- 16X2 LCD Display.
- Buzzer.
- Power supply breakout board.



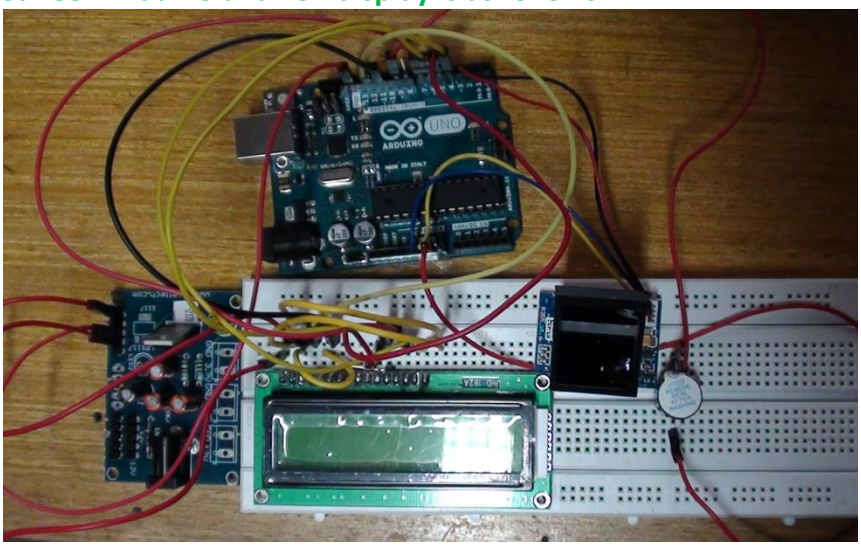
Step one: connection

a) Connection between Arduino and Finger print scanner is as follows.

Arduino Uno board	Finger print scanner
5v	VCC
GND	GND
D2	TX
D3	RX



b) Connection between Arduino and LCD display is as follows.



16X2 LCD Display	Arduino Uno board
GND	GND
VCC	5V
VEE	GND
RS	D12
R/W	GND
EN	D11
DB0	-
DB1	-
DB2	-
DB3	-
DB4	5

DB5	4
DB6	7
DB7	6
LED+	5V
LED-	GND

c) Connection between Arduino with Buzzer.

Arduino Uno board	Buzzer
D10	plus
GND	minus

Step two:

Connect the Arduino board to the computer using Arduino USB cable. Launch the Arduino IDE and select the appropriate serial port and board.

Note:

1. Before uploading the code to the Arduino board make sure you have installed Adafruit_fingerprint library has been installed.

2. Initially the authorized persons need to be authenticated by enrolling his finger prints. This authenticated finger prints will be stored in the finger print scanner database, so that if a match is found, the user is allowed to access. This enrolling process can be done by first executing enroll example in the Adafruit_fingerprint library.

3. I have enrolled my right hand index finger in the database address location 5, and I programed in such a way that, only if address location 5 is matched, it allow you to access.

Program:

```
#include <Adafruit_Fingerprint.h>

#include <SoftwareSerial.h>

#include <LiquidCrystal.h>

intgetFingerprintIDez();



// pin #2 is IN from sensor (GREEN wire)
```

```
// pin #3 is OUT from arduino  (WHITE wire)

SoftwareSerialmySerial(2, 3);



LiquidCrystallcd(12, 11, 5, 4, 7, 6);

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);



void setup()

{

lcd.begin(16, 2);

 // Print a message to the LCD.

lcd.setCursor(0, 0);

lcd.print("Welcome to");

lcd.setCursor(0, 1);

lcd.print("security system!");

delay(1000);

pinMode(13,OUTPUT);

pinMode(10,OUTPUT);

Serial.begin(9600);

Serial.println("fingertest");



 // set the data rate for the sensor serial port
```

```
finger.begin(57600);


if (finger.verifyPassword()) {
lcd.clear();
lcd.print("sensor found");
delay(1000);
Serial.println("Found fingerprint sensor!");
 }
else {
Serial.println("Did not find fingerprint sensor :(");
while (1);
 }
lcd.clear();
lcd.print("keep finger");
Serial.println("Waiting for valid finger...");
}


void loop()                    // run over and over again
{
getFingerprintIDez();
delay(50);          //don't ned to run this at full speed.
```

```
}


uint8_tgetFingerprintID() {
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_PACKETRECIEVEERR:
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_PACKETRECIEVEERR:
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.fingerFastSearch();
if (p == FINGERPRINT_OK) {

 }
else if (p == FINGERPRINT_PACKETRECIEVEERR) {
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);
}


// returns -1 if failed, otherwise returns ID #
intgetFingerprintIDez() {
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) {
Serial.println("finger not matched");
```

```
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Unauthenticated");
lcd.setCursor(0, 1);
lcd.print("    User!");
return -1;
 }
if(finger.fingerID==5)
 {
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("    User");
lcd.setCursor(0, 1);
lcd.print("authenticated!");
digitalWrite(10,HIGH);
delay(500);
digitalWrite(10,LOW);
 }
Serial.print("Found ID #");
Serial.print(finger.fingerID);
```

```
Serial.print(" with confidence of ");

Serial.println(finger.confidence);

 // return finger.fingerID;

}
```

Step three:

Finally to see the sensor output, click the serial monitor icon in the Arduino IDE and also you can see in LCD display.