**FINGERPIRNT BASED ATM MONEY MANAGEMENT USING IoT**

**FINGERPRINT:**

\* FINGERPINT SENSOR = 2,3

\*ENROLL BUTTON = 4

\* IDENT BUTTON =5

\* RELAY FOR MOTOR = 7

\*/

#include <Adafruit\_Fingerprint.h>

#include<LiquidCrystal.h>

#include <SoftwareSerial.h>

SoftwareSerial mySerial (2,3);

SoftwareSerial SIM900 (11,12);

LiquidCrystal lcd (4,5,A2,A3,A4,A5);

///////////////////// finger print //////////////////////

#define en\_sw A0

#define ver\_sw A1

#define bz 5

#define rel 7

int f1=1,f2=1,f3=1;

Adafruit\_Fingerprint finger = Adafruit\_Fingerprint(&mySerial);

uint8\_t id=0;

//int i=0;

//int flagg = 1

short countt = 0;

short randd = 0;

int getFingerprintIDez();

void sendsms (String rr, String qq);

void setup()

{

Serial.begin(9600);

SIM900.begin(9600);

delay(100);

mySerial.listen();

delay(500);

pinMode(en\_sw,INPUT\_PULLUP);

pinMode(ver\_sw,INPUT\_PULLUP);

pinMode(bz,OUTPUT);

PinMode(rel,OUTPUT);

lcd.begin(16,2);

lcd.clear();

lcd.setCursor(4,0);

lcd.print("WELCOME");

lcd.setCursor(0,1);

lcd.print(" .............. ");

Serial.println(F("WELCOME"));

while (!Serial); // For Yun/Leo/Micro/Zero/...

delay(100);

Serial.println(F("\n\nAdafruit Fingerprint sensor enrollment & Verify"));

finger.begin(9600);

if (finger.verifyPassword()) {

Serial.println(F("Found fingerprint sensor!"));

} else {

Serial.println(F("Did not find fingerprint sensor :("));

while (1) { delay(1); }

}

lcd.clear();

lcd.setCursor(0,0);

lcd.print("FINGERPRINT BSD");

lcd.setCursor(0,1);

lcd.print("DOOR LOCKING SYSTEM");

Serial.println(F("FINGERPRINT BSD DOOR LOCKING SYSTEM"));

delay(2000);

}

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

{

//int en =digitalRead(en\_sw);

//int ve =digitalRead(ver\_sw);

int en =analogRead(en\_sw);

int ve =analogRead(ver\_sw);

Serial.print("enr=");Serial.println(en);

Serial.print("ver=");Serial.println(ve);

delay(1000);

lcd.clear();

//ENROLLING ------------------------

// if(digitalRead(en\_sw)==0)

if(en<20) {

id++;

Serial.print(F("Enrolling ID #"));

Serial.println(id);

lcd.setCursor(0,0);

lcd.print("Enrolling.......");

delay(500);

while (! getFingerprintEnroll() );

}

//IDENTIFICATION ------------------------//

if(digitalRead(ver\_sw)==0)

if(ve<20){

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Ver.Finger To Acc");

delay(2000)

finger.getTemplateCount();

Serial.print("Sensor contains "); Serial.print(finger.templateCount); Serial.println(" templates");

Serial.println("Waiting for valid finger...");

int ID;

ID=getFingerprintIDez();//returns Finger Id

delay(500); //don't ned to run this at full speed.

//person1---------------------

if (ID == 1)

{

Serial.println(F("Match Found With ID "));

countt = 0;

lcd.clear();

lcd.setCursor(0,0);

lcd.print("FINGER MATCHED ");

lcd.setCursor(0,1);

lcd.print("DOOR OPENED ");

digitalWrite(rel,HIGH);

delay(2000);

}

else if (ID == 2)

{

Serial.println(F("Match Found With ID "));

countt = 0;

lcd.clear();

lcd.setCursor(0,0);

lcd.print("FINGER MATCHED ");

lcd.setCursor(0,1);

lcd.print("DOOR CLOSED ");

digitalWrite(rel,LOW);

delay(2000);

}

else{

Serial.println("Match Not Found");

countt++ ;

lcd.setCursor(0,0);

lcd.print("FINGER NT MATCHED");

lcd.setCursor(0,1);

lcd.print("COUNT: "+String(countt));

buz();

Serial.println("countt: "+String(countt));

}

}//verification

} // Loop End

void ShowSerialData ()

{

if (SIM900.available() != 0)

{

while(SIM900.available() != 0)

{

Serial.write(SIM900.read());

}

}

}

//////////////////////// INIT GSM/SMS SENDING START //////////////////////////////

void buz()

{

digitalWrite(bz,1);

delay(200);

digitalWrite(bz,0);

delay(10);

digitalWrite(bz,1);

delay(200);

digitalWrite(bz,0);

}

////////////// Enroll ///////////////////

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\_PACKETRECIEVEERR:

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\_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;

}

Serial.println("Remove finger");

lcd.setCursor(0,0);

lcd.print("Remove Finger ");

buz();

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");

// lcd.clear();

lcd.setCursor(0,0);

lcd.print("Put Finger Again");

delay(2000);

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\_PACKETRECIEVEERR:

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\_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!

Serial.print("Creating model for #"); Serial.println(id);

p = finger.createModel();

if (p == FINGERPRINT\_OK) {

Serial.println("Prints matched!");

} else if (p == FINGERPRINT\_PACKETRECIEVEERR) {

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!");

// lcd.clear();

lcd.setCursor(0,0);

lcd.print("Enroll Success ");

} else if (p == FINGERPRINT\_PACKETRECIEVEERR) {

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;

}

}

////////////// Enroll End ///////////////////

////////////// Verify ///////////////////

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\_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) {

Serial.println("Found a print match!");

} 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);

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;

}

////////////// Verify End ////////////////////

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_//