**Arduino Code :**

#include <Adafruit\_Fingerprint.h>

#include <SoftwareSerial.h>

#include<LiquidCrystal.h>

SoftwareSerial mySerial(2, 3);

LiquidCrystal lcd(A0,A1,A2,A3,A4,A5);

#define en\_sw 4

#define ver\_sw 5

#define m1 6

#define m2 7

#define buz 11

int u1=0,u2=0,u3=0;

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

bool ch\_status=false,flag\_check=true;

uint8\_t id = 0;

int i=0,id\_val,a=0,b=0,flag1 = 0,flag2=0,flag3=0;

int getFingerprintIDez();

/////////////////////////

void setup() {

pinMode(en\_sw, INPUT\_PULLUP);

pinMode(ver\_sw,INPUT\_PULLUP);

pinMode(m1,OUTPUT);

pinMode(m2,OUTPUT);

pinMode(buz,OUTPUT);

digitalWrite(buz,0);

digitalWrite(m1,0);

digitalWrite(m2,0);

lcd.begin(16,2);

lcd.setCursor(4,0);

lcd.print("WELCOME");

lcd.setCursor(0,1);

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

delay(1000);

lcd.clear();

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

delay(100);

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

finger.begin(57600);

if (finger.verifyPassword()) {

lcd.setCursor(0,0);

lcd.print("FOUND FP SENSOR!");

delay(1000);

} else {

lcd.setCursor(0,0);

lcd.print("FP SENSOR ");

lcd.setCursor(0,1);

lcd.print(" NOT FOUND!");

delay(1000);

while (1) { delay(1); } }

lcd.setCursor(0,0);

lcd.print("BIOMETRIC IGNTION");

lcd.setCursor(0,1);

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

delay(1000);

lcd\_msg(); }

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

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

if(ch\_status==false){

if(id >= 3){

lcd.setCursor(0,0);

lcd.print("Exceed Enroll ");

lcd.setCursor(0,1);

lcd.print("Limit ");lcd.print(id);

lcd\_msg(); }

if(id >= 0 && id < 3){

id++;

id\_val++;

lcd.setCursor(0,0);

lcd.print(F("Enrolling Id: ")); lcd.setCursor(0,1); lcd.print(id); lcd.print(F(" "));

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

//Serial.print(id);

while(p==-1) {

p=getFingerprintEnroll(); }

ch\_status=true;

p=-1; }}

else{

ch\_status=false;

p=-1; }}

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

lcd.setCursor(0,0);

lcd.print("Scan Your Finger");

delay(2000);

finger.getTemplateCount();

int ID;

ID=getFingerprintIDez();//returns Finger Id

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

if(ID==1){

Serial.println(F("Match Found With ID 1"));

u1++;

if(u1%2!=0){

Serial.println("Match Found With ID 1");

lcd.setCursor(0,0);

lcd.print("MATCH FOUND USR1");

lcd.setCursor(0,1);

lcd.print("IGNITION ON ");

digitalWrite(m1,1);

digitalWrite(m2,0);

flag1 = 1; }

if(u1%2==0){

Serial.println("Match Found With ID 1");

lcd.setCursor(0,0);

lcd.print("MATCH FOUND USR1");

lcd.setCursor(0,1);

lcd.print("IGNITION OFF ");

digitalWrite(m1,0);

digitalWrite(m2,0);lcd\_msg();

flag1 = 0; } }

else if(ID==2){

u2++;

if(u2%2!=0){

Serial.println("Match Found With ID 2");

lcd.setCursor(0,0);

lcd.print("MATCH FOUND USR2");

lcd.setCursor(0,1);

lcd.print("IGNITION ON ");

digitalWrite(m1,1);

digitalWrite(m2,0);

flag1 = 1;}

if(u2%2==0){

Serial.println("Match Found With ID 2");

lcd.setCursor(0,0);

lcd.print("MATCH FOUND USR2");

lcd.setCursor(0,1);

lcd.print("IGNITION OFF ");

digitalWrite(m1,0);

digitalWrite(m2,0);lcd\_msg();

flag1 = 0; } }

else if(ID==3){

u3++;

if(u3%2!=0){

Serial.println("Match Found With ID 3");

lcd.setCursor(0,0);

lcd.print("MATCH FOUND USR3");

lcd.setCursor(0,1);

lcd.print("IGNITION ON ");

digitalWrite(m1,1);

digitalWrite(m2,0);

flag1 = 1; }

if(u3%2==0){

Serial.println("Match Found With ID 3");

lcd.setCursor(0,0);

lcd.print("MATCH FOUND USR3");

lcd.setCursor(0,1);

lcd.print("IGNITION OFF ");

digitalWrite(m1,0);

digitalWrite(m2,0);lcd\_msg();

flag1 = 0; } }

else{

Serial.println("Match Not Found");

beep(3);

lcd.setCursor(0,1);

lcd.print("MATCH NOT FOUND ");delay(1000);

lcd.setCursor(0,1);

lcd.print("PLS. TRY AGAIN "); delay(1000);

lcd\_msg();

} } }// Loop End

\\ Enroll //

uint8\_t getFingerprintEnroll() {

//Serial1.print(F("Waiting for valid finger to enroll as #")); Serial1.println(id);

Serial.print(F("Enrolling ")); Serial.print(id);

delay(1000);

while (p != FINGERPRINT\_OK) {

p = finger.getImage();

switch (p) {

case FINGERPRINT\_OK:

Serial.println(F("Image taken"));

break;

case FINGERPRINT\_NOFINGER:

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

break;

case FINGERPRINT\_PACKETRECIEVEERR:

Serial.println(F("Communication error"));

break;

case FINGERPRINT\_IMAGEFAIL:

Serial.println(F("Imaging error"));

break;

default:

Serial.println(F("Unknown error"));

break;} } // OK success!

Serial.print(F("Remove Finger"));

lcd.setCursor(0,0);

lcd.print(F("Remove Finger "));

delay(1000);

p = 0;

while (p != FINGERPRINT\_NOFINGER) {

p = finger.getImage(); }

Serial.print(F("ID ")); Serial.println(id);

p = -1;

Serial.println(F("Place same finger again"));

// Serial.print(F("Put Same Finger Again"));

// lcd.clear();

lcd.setCursor(0,0);

lcd.print(F("Put Finger Again"));

delay(3000);

while (p != FINGERPRINT\_OK) {

p = finger.getImage();

switch (p) {

case FINGERPRINT\_OK:

Serial.println(F("Image taken"));

break;

case FINGERPRINT\_NOFINGER:

Serial.print(F("."));

break;

case FINGERPRINT\_PACKETRECIEVEERR:

Serial.println(F("Communication error"));

break;

case FINGERPRINT\_IMAGEFAIL:

Serial.println(F("Imaging error"));

break;

default:

Serial.println(F("Unknown error"))

break;} }

// OK success! // OK converted!

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

p = finger.createModel();

if (p == FINGERPRINT\_OK) {

Serial.println(F("Prints matched!"));

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

Serial.println(F("Communication error"));

return p;

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

Serial.println(F("Fingerprints did not matches"));

delay(1000);

Serial.print(F("Enroll Fail"));

lcd.setCursor(0,0);

lcd.print(F("FAIL TO ENROLL "));

if(id>0) { id--; }

lcd\_msg();

return p;

} else {

Serial.println(F("Unknown error"));

return p; }

Serial.print(F("ID ")); Serial.println(id);

p = finger.storeModel(id);

if (p == FINGERPRINT\_OK) {

Serial.println(F("Stored!"));

Serial.print(F("Enroll Success"));

lcd.clear();

lcd.setCursor(0,0);

lcd.print(F("Enroll Success ")); lcd.setCursor(0,1);lcd.print(id);lcd.print(" ");delay(2000);

lcd.clear();

lcd.setCursor(0,0);

lcd.print(F("Enroll -- Verify"));

return p;

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

Serial.println(F("Communication error"));

return p;

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

Serial.println(F("Could not store in that location"));

return p;

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

Serial.println(F("Error writing to flash"));

return p;

} else {

Serial.println(F("Unknown error"));

return p;

}}

//// Enroll End //// Verify

uint8\_t getFingerprintID() {

uint8\_t p = finger.getImage();

switch (p) {

case FINGERPRINT\_OK:

Serial.println(F("Image taken"));

lcd.setCursor(0,1);

lcd.print(F("Image Taken"));

break;

case FINGERPRINT\_NOFINGER:

Serial.println(F("No finger detected"));

lcd.setCursor(0,1);

lcd.print(F("NO FINGER "));

return p;

case FINGERPRINT\_PACKETRECIEVEERR:

Serial.println(F("Communication error"));

return p;

case FINGERPRINT\_IMAGEFAIL:

Serial.println(F("Imaging error"));

return p;

default:

Serial.println(F("Unknown error"));

return p; } // OK success!

// 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(F("Found ID #")); Serial1.print(finger.fingerID);

Serial.print(F(" with confidence of ")); Serial.println(finger.confidence);

return finger.fingerID;

}

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