#include <Adafruit\_Fingerprint.h>  
#include <SoftwareSerial.h> // SoftwareSerial(rxPin, txPin, inverse\_logic)  
SoftwareSerial mySerial(2, 3); //you can change them if it is not working on 2 or 3  
  
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\nFingerprint 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); }  
 }  
}  
  
uint8\_t readnumber(void) {  
 uint8\_t num = 0;  
   
 while (num == 0) {  
 while (! Serial.available());  
 num = Serial.parseInt();  
 }  
 return num;  
}  
  
void loop() // program wil repeat this part (loop here)  
{  
 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\_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");  
 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\_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!");  
 } 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;  
 }   
}