#include <Adafruit\_Fingerprint.h>  
#include <SoftwareSerial.h>  
  
SoftwareSerial mySerial(2, 3);  
  
Adafruit\_Fingerprint finger = Adafruit\_Fingerprint(&mySerial);  
  
void setup()  
{  
 Serial.begin(9600);  
 while (!Serial); // For Yun/Leo/Micro/Zero/...  
 delay(100);  
 Serial.println("fingertest");  
 pinMode(12, OUTPUT);  
 pinMode(11, OUTPUT);  
  
  
  
  
  
 // 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);  
 }  
 }  
  
 finger.getTemplateCount();  
 Serial.print("Sensor contains "); Serial.print(finger.templateCount); Serial.println(" templates");  
 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.  
 digitalWrite(12, LOW);  
 digitalWrite(11, LOW);  
}  
  
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;  
 }  
 {digitalWrite(11, HIGH);  
 delay(3000);  
 digitalWrite(11, LOW);  
 Serial.print("Not Found");   
 Serial.print("Error");   
 return finger.fingerID;  
 }  
  
 // 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!  
  
 {  
 digitalWrite(12, HIGH);  
 delay(3000);  
 digitalWrite(12, LOW);  
 Serial.print("Found ID #"); Serial.print(finger.fingerID);  
 Serial.print(" with confidence of "); Serial.println(finger.confidence);  
   
   
   
 } }