Fingerprint Reader Scanner/Sensor (Optical) Model: AS608

A QuickStart Guide

BITSTOC ELECTRONICS
www.bitstoc.com

PRODUCT OVERVIEW

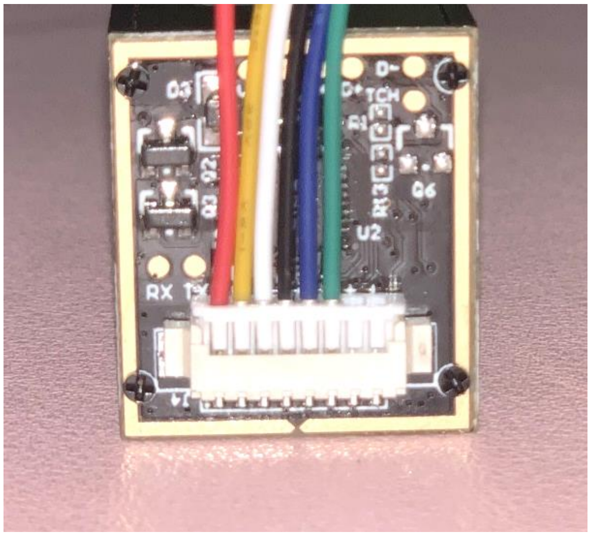
The AS608 module is used to scan fingerprints and send the data to a microcontroller via serial communication. The module can store up to 127 individual fingerprints.

Optical fingerprint sensors take low-resolution snapshots of the tip of a finger and create arrays of identifiers that are then used to uniquely identify a given fingerprint. The AS608 is capable of storing up to 128 individual fingerprints. This tutorial will introduce the AS608 Arduino-compatible fingerprint sensor and how to validate and reject fingerprints based on the enrolled fingerprint information that will be given to the sensor. The fingerprint algorithm is handled by the AS608 and Arduino, so this tutorial will focus on implementation and putting the pieces together to make a working fingerprint sensor with Arduino.

Specifications:

- Fingerprint storage memory: 127 fingerprints
- Resolution: 500dpi
- Supply current: <60mA
- Supply voltage: 3.3V
- Fingerprint image entry time: <1.0 seconds
- Peak current: <60mA

Pin Configuration:



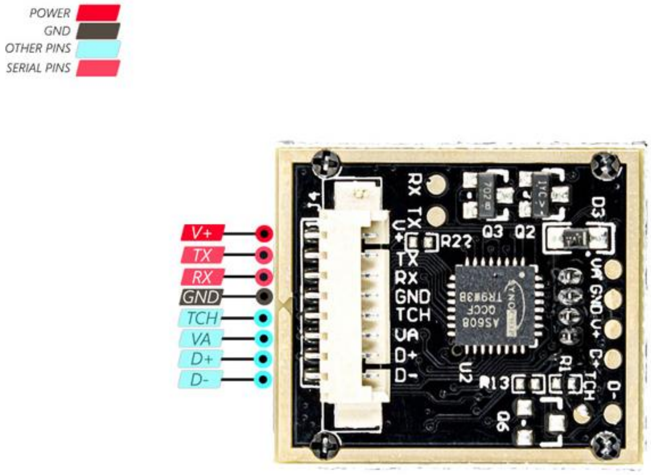


*Image from electropeak.com

SAMPLE PROJECT: Fingerprint Enrollment and Verification

Overview:

In this QuickStart Guide, one sample project will be given using the Fingerprint Reader Sensor to be interfaced on an Arduino microcontroller. As one of the applications of this device, we will be making a "Fingerprint Enrollment" used in security and attendance applications. This guide will follow two procedures: first is to enroll fingerprint and second is to verify fingerprint.

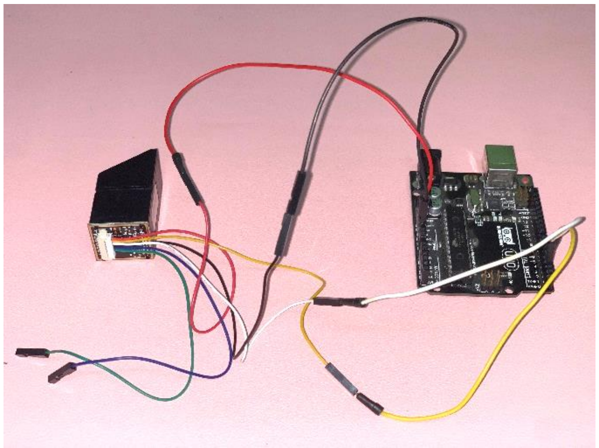
Materials Needed:

The following materials will be needed to complete these QuickStart guide. Links are provided where these materials are available and affordable to purchase.

 Arduino UNO R3 with USB Cable https://www.bitstoc.com/product/1/	 Connecting Wires – Male-Male https://www.bitstoc.com/product/133/	 Fingerprint Reader Scanner/Sensor (optical) Model: AS608 https://www.bitstoc.com/product/1067/

Circuit/Wiring:

The following circuit will be observed in interfacing the Fingerprint Reader Sensor to Arduino Uno.



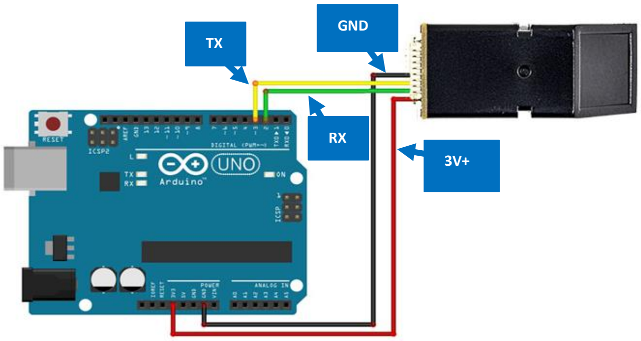
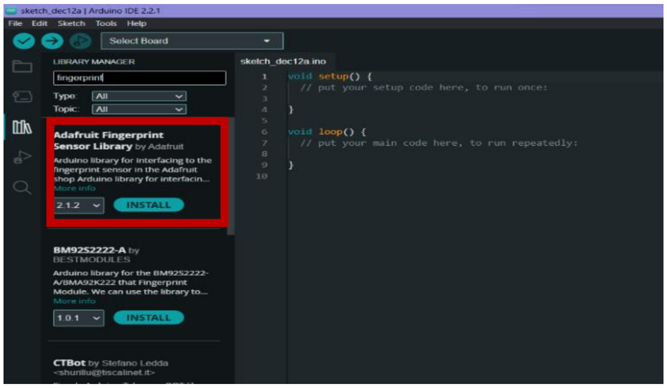


*Image from electropeak.com

Required Library:

To start coding, an Adafruit Fingerprint Sensor library should be installed. Click install on the boxed library from the figure below. When successfully installed, proceed to the 1st procedure which is to enroll fingerprint.



Adafruit Fingerprint Sensor Library by Adafruit
Arduino library for interfacing to the fingerprint sensor in the Adafruit shop Arduino library for interfacin...
More info

BM92S2222-A by BESTMODULES
Arduino library for the BM92S2222-A/BMA92K222 that Fingerprint Module. We can use the library to...
More info

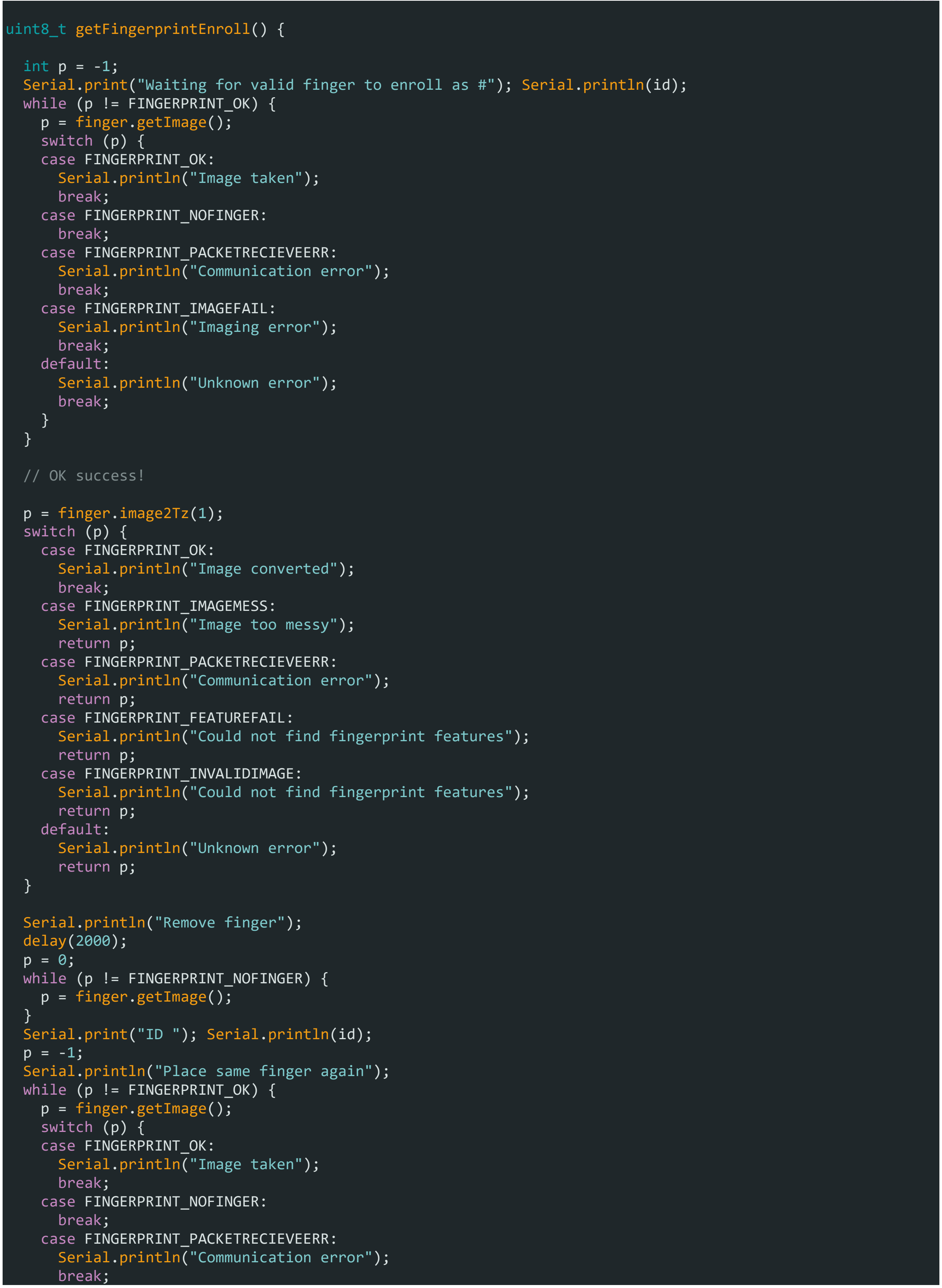
CTBot by Stefano Ledda <shurillu@tiscalinet.it>

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

1st Procedure [Enroll Fingerprint]:

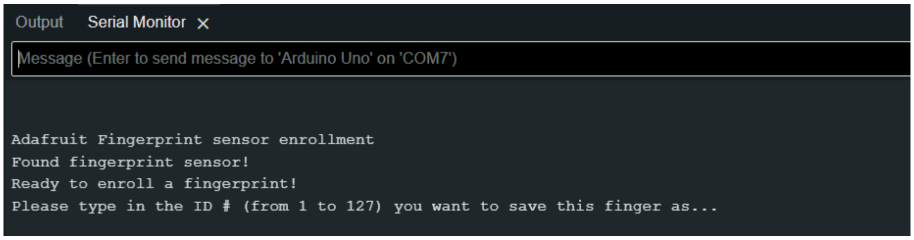
To start with the fingerprint enrollment, copy the following code to Arduino IDE. Verify and upload.

```
#include <Adafruit_Fingerprint.h>

SoftwareSerial mySerial(2, 3); // TX/RX
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\nAdafruit Fingerprint 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()                     // run over and over again
{
  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() );
}
```



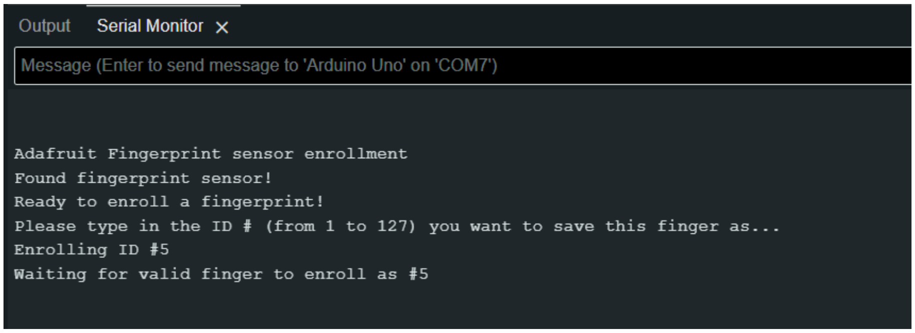
```
uint8_t getFingerprintEnroll() {

  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:
      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:
      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;
  }
}
```

Open the serial monitor and the following lines should show:



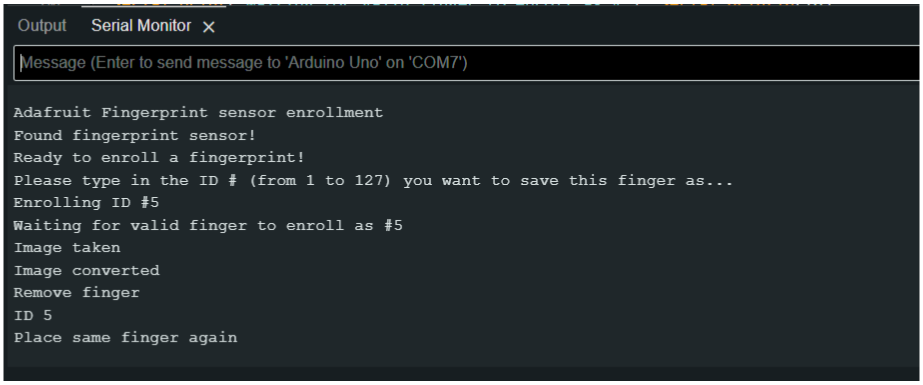
```
Adafruit Fingerprint sensor enrollment
Found fingerprint sensor!
Ready to enroll a fingerprint!
Please type in the ID # (from 1 to 127) you want to save this finger as...
```

In the next step, you have to choose your ID number between 1 to 127. In this example, if we choose 5 and hit “Enter”, the next monitor should show the following:



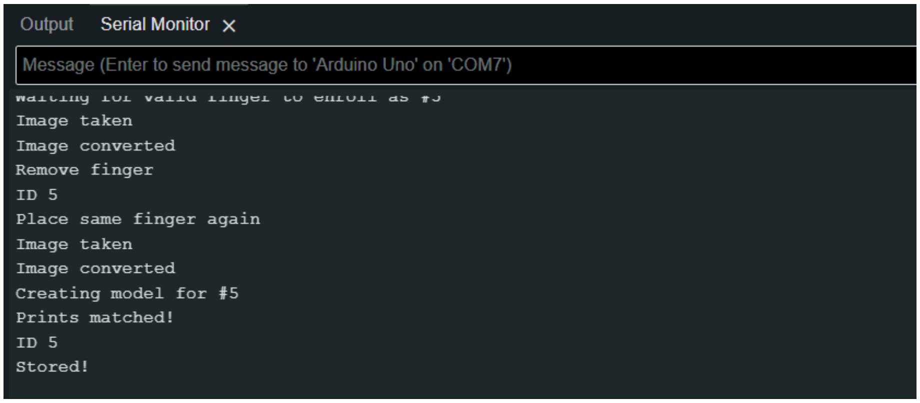
```
Adafruit Fingerprint sensor enrollment
Found fingerprint sensor!
Ready to enroll a fingerprint!
Please type in the ID # (from 1 to 127) you want to save this finger as...
Enrolling ID #5
Waiting for valid finger to enroll as #5
```

The next step is to place your finger on the sensor enroll its image. When placing the finger for about 3 seconds, the following should show from the Serial Monitor:



```
Adafruit Fingerprint sensor enrollment
Found fingerprint sensor!
Ready to enroll a fingerprint!
Please type in the ID # (from 1 to 127) you want to save this finger as...
Enrolling ID #5
Waiting for valid finger to enroll as #5
Image taken
Image converted
Remove finger
ID 5
Place same finger again
```

You will then be asked to place the same finger for the second time. After that, the word “Stored!” will appear from the monitor:



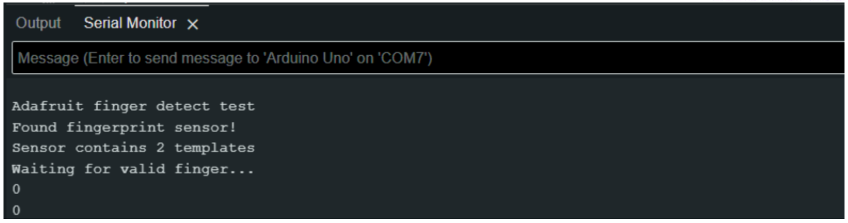
```
Waiting for valid finger to enroll as #5
Image taken
Image converted
Remove finger
ID 5
Place same finger again
Image taken
Image converted
Creating model for #5
Prints matched!
ID 5
Stored!
```

2^{nd} Procedure [Verify Fingerprint]:

Once the first procedure is completed, copy the code below and paste it to a separate sketch of Arduino IDE to start with the fingerprint verification process. Run the code and open Serial Monitor.

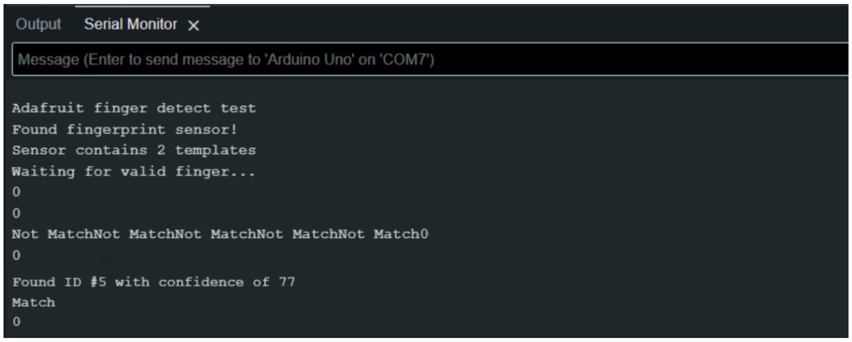
```
#include <Adafruit_Fingerprint.h>

volatile int finger_status = -1;
SoftwareSerial mySerial(2, 3); // TX/RX on fingerprint sensor
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

void setup()
{
  Serial.begin(9600);
  while (!Serial);  // For Yun/Leo/Micro/Zero/...
  delay(100);
  Serial.println("\n\nAdafruit finger detect test");

  // 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
{
  finger_status = getFingerprintIDez();
  if (finger_status!=-1 and finger_status!=-2){
    Serial.print("Match");
  } else{
    if (finger_status==-2){
      for (int ii=0;ii<5;ii++){
        Serial.print("Not Match");
      }
    }
  }
  delay(50);            //don't ned to run this at full speed.
}

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
  uint8_t p = finger.getImage();
  if (p!=2){
    Serial.println(p);
  }
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.image2Tz();
  if (p!=2){
    Serial.println(p);
  }
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK)  return -2;

  // found a match!
  Serial.print("Found ID #"); Serial.print(finger.fingerID);
  Serial.print(" with confidence of "); Serial.println(finger.confidence);
  return finger.fingerID;
}
```

Once successfully loaded, the following should show from the Serial Monitor.



```
Adafruit finger detect test
Found fingerprint sensor!
Sensor contains 2 templates
Waiting for valid finger...
0
0
```

Place your finger for verification. When placing an unenrolled fingerprint on the sensor, the word “Not Match” will show, and when an enrolled fingerprint is placed on the sensor, the word “Match” should show.



```
Adafruit finger detect test
Found fingerprint sensor!
Sensor contains 2 templates
Waiting for valid finger...
0
0
Not MatchNot MatchNot MatchNot MatchNot Match0
0
Found ID #5 with confidence of 77
Match
0
```

Congratulations! You have completed the Fingerprint Enrollment and Verification Project using the Fingerprint Reader Sensor (Optical) Model: AS608.

REFERENCES

Adafruit Optical Fingerprint Sensor by lady ada

- https://cdn-learn.adafruit.com/downloads/pdf/adafruit-optical-fingerprint-sensor.pdf

Interfacing AS608 Optical Fingerprint Sensor Module with Arduino by Amir Mohammad Shojaei

- https://electropeak.com/learn/interfacing-fpm10a-as608-optical-fingerprint-reader-sensor-module-with-arduino/

Fingerprint Reader Scanner/Sensor (optical) (Model: AS608) by Bitstoc Electronics

- https://www.bitstoc.com/product/1067/