

Fingerprint Scanner UART

Adafruit compatible

Technical Manual Rev 1r0



The **Adafruit Fingerprint** is a **PC interface UART (TTL logic level)** an easy to use for your security project with biometric system, this all in one optical fingerprint sensor will make adding fingerprint detection and verification super simple. Typically used in safes there's a high powered DSP chip that does the image rendering, calculation, feature-finding and searching. Compatible to any gizduino microcontroller or MCU system with TTL serial and send packets of data to take photos, detect prints, hash and search. Can store up to 162 finger prints in the onboard FLASH memory.

General Specifications:

Input supply voltage: 3.3V

Operating current: <120mA

Signature File: 256 bytes

Template Files: 512 bytes

Store capacity: 162 templates

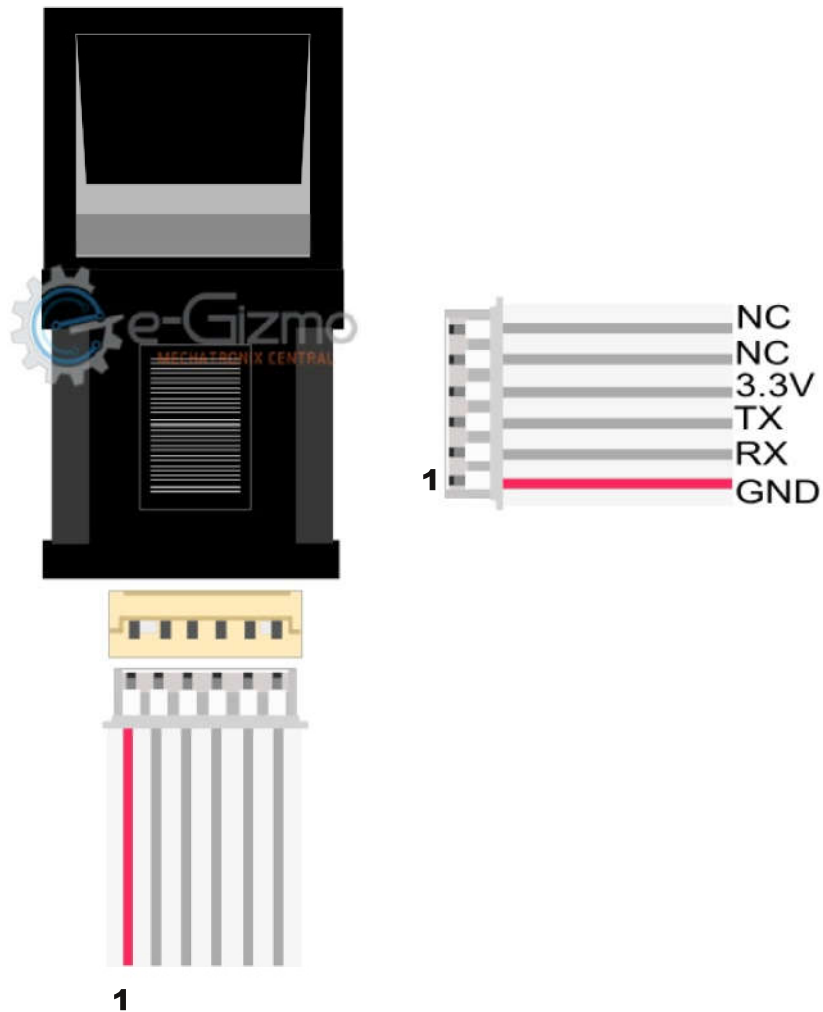
Security Level: Five (from low to high: 1,2,3,4,5)

PC interface: UART (TTL logic level) or USB1.1

Communications baud rate x N bps: 9600 N=1 to 12
(default value of N = 6, i.e 57600bps)

Module Dimensions: 44.5mmH-a /27mmH-b, 29mm Slant,
21mm W-side, 21mm W-front

PCB Dimensions: 21mm x 21 mm



**Figure 1: Fingerprint Scanner UART
Adafruit Compatible**

Wiring Connections:

Gizduino to FPS

| | |
|--------------|-----------------------|
| GND | Black wire (1) |
| D3 | Green wire (2) |
| D2 | Blue wire (3) |
| +3.3V | Red wire (4) |

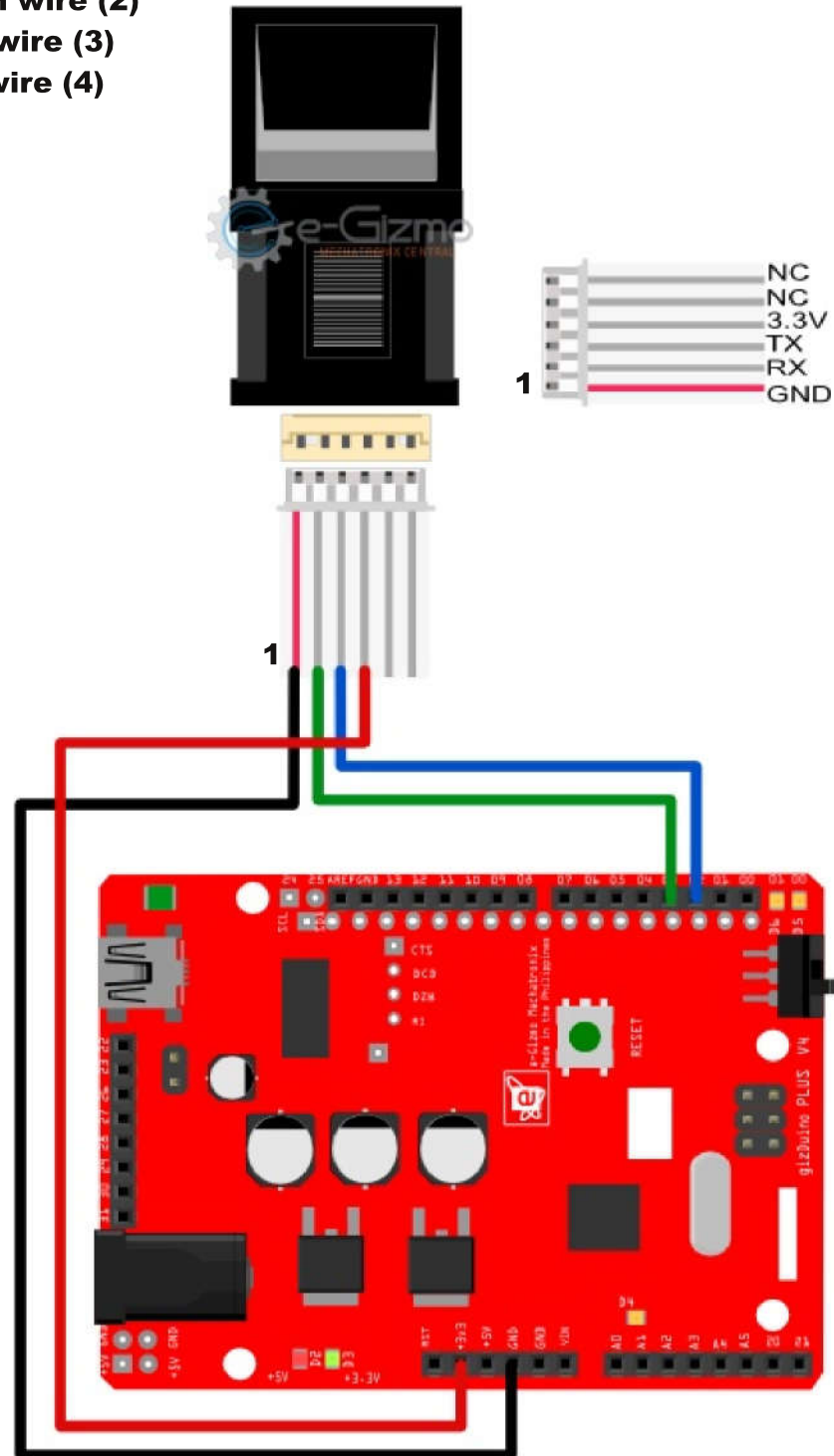


Figure 2: Wiring diagram with gizduino PLUS.

Adafruit_Fingerprint Library

Download the library from adafruit Github account.
<https://github.com/adafruit/Adafruit-Fingerprint-Sensor-Library>

Adding Library

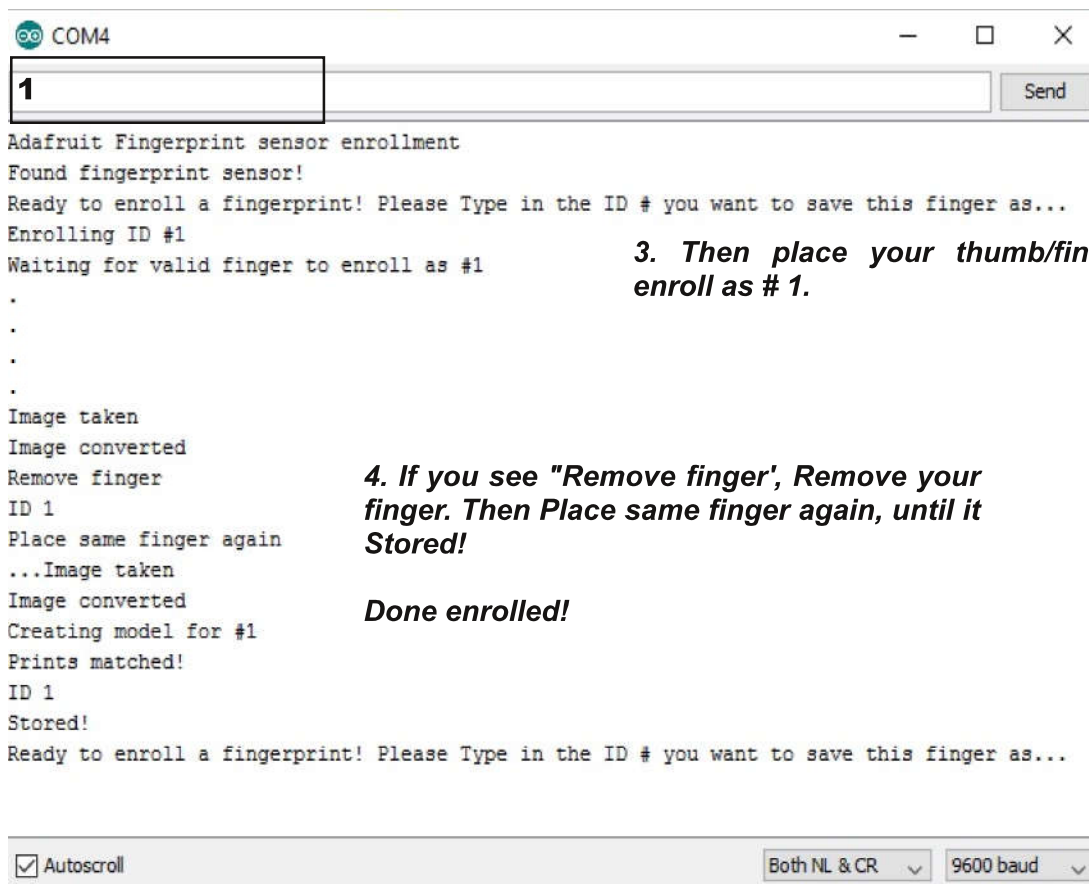
Go to My Documents> Arduino folder> libraries> add the extracted zip file. Folder file name: Adafruit_Fingerprint

Upload the Adafruit_Fingerprint> enroll.ino

Then open the Serial monitor as shown below.
When the FP sensor is found and ready to enroll.

1. Type here the ID# you want to save.

2. Press enter or send it.



The screenshot shows the Arduino Serial Monitor window for COM4. The input field contains the number '1'. The output text shows the following sequence of events:

```
Adafruit Fingerprint sensor enrollment
Found fingerprint sensor!
Ready to enroll a fingerprint! Please Type in the ID # you want to save this finger as...
Enrolling ID #1
Waiting for valid finger to enroll as #1
.
.
.
.
Image taken
Image converted
Remove finger
ID 1
Place same finger again
...Image taken
Image converted
Creating model for #1
Prints matched!
ID 1
Stored!
Ready to enroll a fingerprint! Please Type in the ID # you want to save this finger as...
```

At the bottom of the window, the 'Autoscroll' checkbox is checked, and the baud rate is set to 9600 baud.

3. Then place your thumb/fingerprint to enroll as # 1.

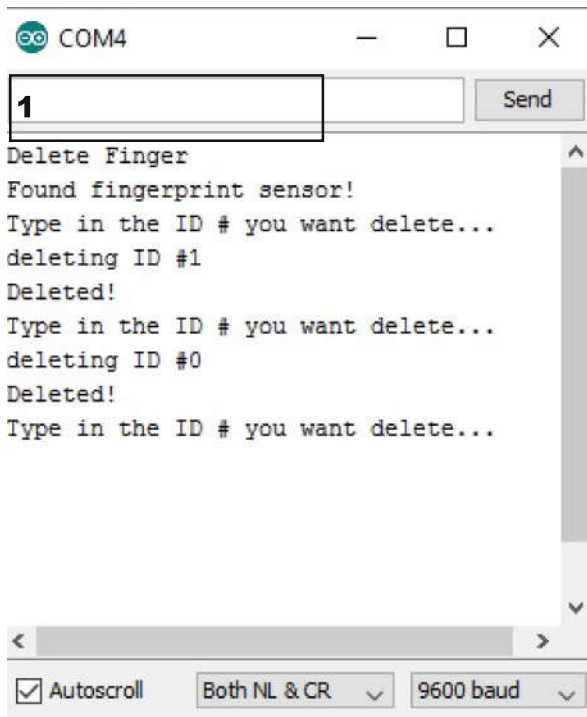
4. If you see "Remove finger", Remove your finger. Then Place same finger again, until it Stored!

Done enrolled!

Upload the Adafruit_Fingerprint> delete.ino

Then open the Serial monitor as shown below.
When the FP sensor is found and ready to delete.

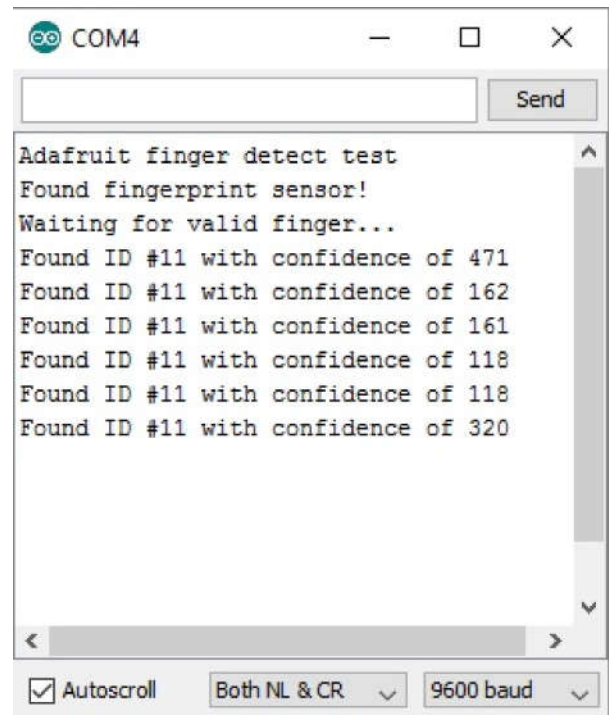
1. Type here the ID# you want to delete.



Upload the Adafruit_Fingerprint> fingerprint.ino

Then open the Serial monitor as shown below.
When the FP sensor is found and ready to identify.

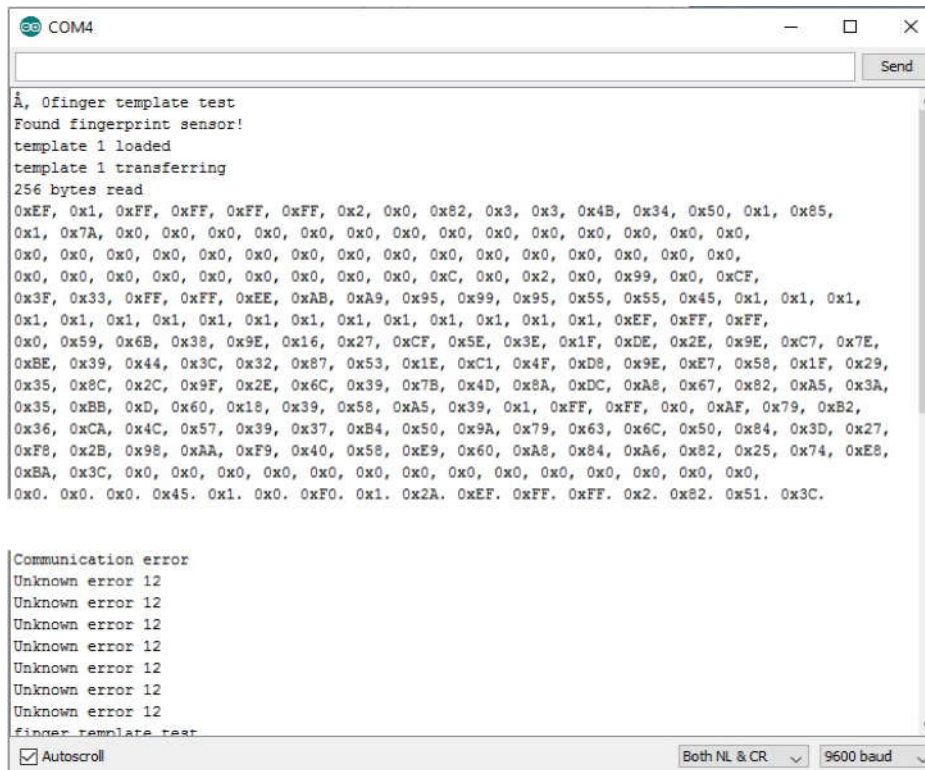
1. Place your finger/thumb that you've been enrolled and it will show up on the serial monitor the # ID with confidence value.



**Upload the Adafruit_Fingerprint>
show_fingerprint_templates.ino**

Then open the Serial monitor as shown below.
When the FP sensor is found and ready to
template test.

**1. Place your finger/thumb that you've been
enrolled and it will show up on the serial
monitor the data with bytes read in hex
format.**



```

COM4
A, 0finger template test
Found fingerprint sensor!
template 1 loaded
template 1 transferring
256 bytes read
0xEF, 0x1, 0xFF, 0xFF, 0xFF, 0xFF, 0x2, 0x0, 0x82, 0x3, 0x3, 0x4B, 0x34, 0x50, 0x1, 0x85,
0x1, 0x7A, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0,
0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0,
0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0xC, 0x0, 0x2, 0x0, 0x99, 0x0, 0xCF,
0x3F, 0x33, 0xFF, 0xFF, 0xEE, 0xAB, 0xA9, 0x95, 0x99, 0x95, 0x55, 0x55, 0x45, 0x1, 0x1, 0x1,
0x1, 0x1, 0x1, 0x1, 0x1, 0x1, 0x1, 0x1, 0x1, 0x1, 0x1, 0x1, 0xEF, 0xFF, 0xFF,
0x0, 0x59, 0x6B, 0x38, 0x9E, 0x16, 0x27, 0xCF, 0x5E, 0x3E, 0x1F, 0xDE, 0x2E, 0x9E, 0xC7, 0x7E,
0xBE, 0x39, 0x44, 0x3C, 0x32, 0x87, 0x53, 0x1E, 0xC1, 0x4F, 0xD8, 0x9E, 0xE7, 0x58, 0x1F, 0x29,
0x35, 0x8C, 0x2C, 0x9F, 0x2E, 0x6C, 0x39, 0x7B, 0x4D, 0x8A, 0xDC, 0xA8, 0x67, 0x82, 0xA5, 0x3A,
0x35, 0xBB, 0xD, 0x60, 0x18, 0x39, 0x58, 0xA5, 0x39, 0x1, 0xFF, 0xFF, 0x0, 0xAF, 0x79, 0xB2,
0x36, 0xCA, 0x4C, 0x57, 0x39, 0x37, 0xB4, 0x50, 0x9A, 0x79, 0x63, 0x6C, 0x50, 0x84, 0x3D, 0x27,
0xFB, 0x2B, 0x98, 0xAA, 0xF9, 0x40, 0x58, 0xE9, 0x60, 0xA8, 0x84, 0xA6, 0x82, 0x25, 0x74, 0xE8,
0xBA, 0x3C, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0,
0x0. 0x0. 0x0. 0x45. 0x1. 0x0. 0xF0. 0x1. 0x2A. 0xEF. 0xFF. 0xFF. 0x2. 0x82. 0x51. 0x3C.

Communication error
Unknown error 12
Unknown error 12
Unknown error 12
Unknown error 12
Unknown error 12
Unknown error 12
Unknown error 12
finger template test
☒ Autoscroll
Both NL & CR
9600 baud
  
```

Upload the Adafruit_Fingerprint> blank.ino

Blank sample code is compose of

```
void setup(){}  
void loop(){}  

```

Upload this code then make sure your connections from gizduino with the FPS is correct. Please see the wiring connections (Figure 3).

This code is for SFGDemo rev2.0 from Adafruit website. "Compatible in windows only test software"

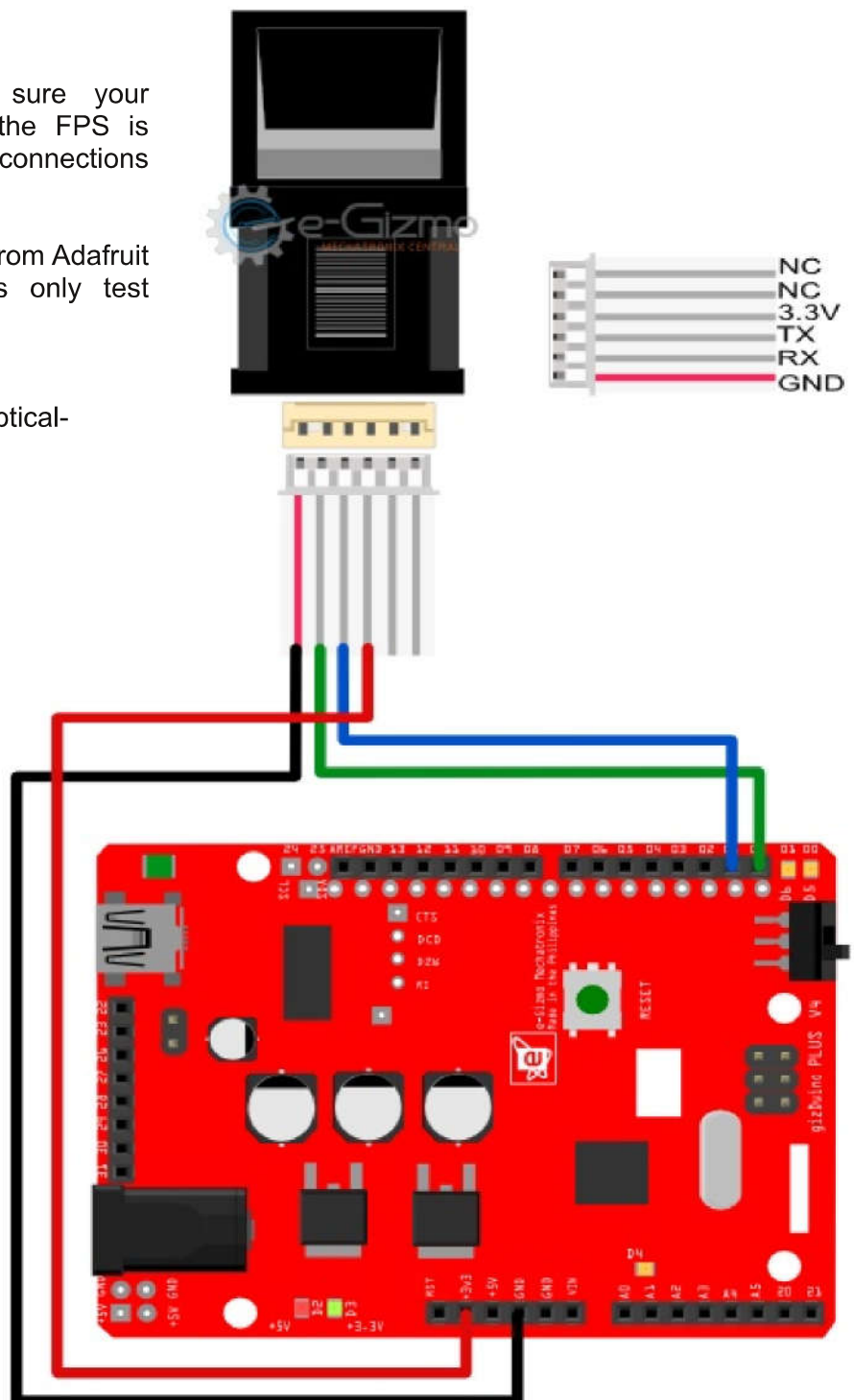
Here's the link:
<https://learn.adafruit.com/adafruit-optical-fingerprint-sensor/downloads>

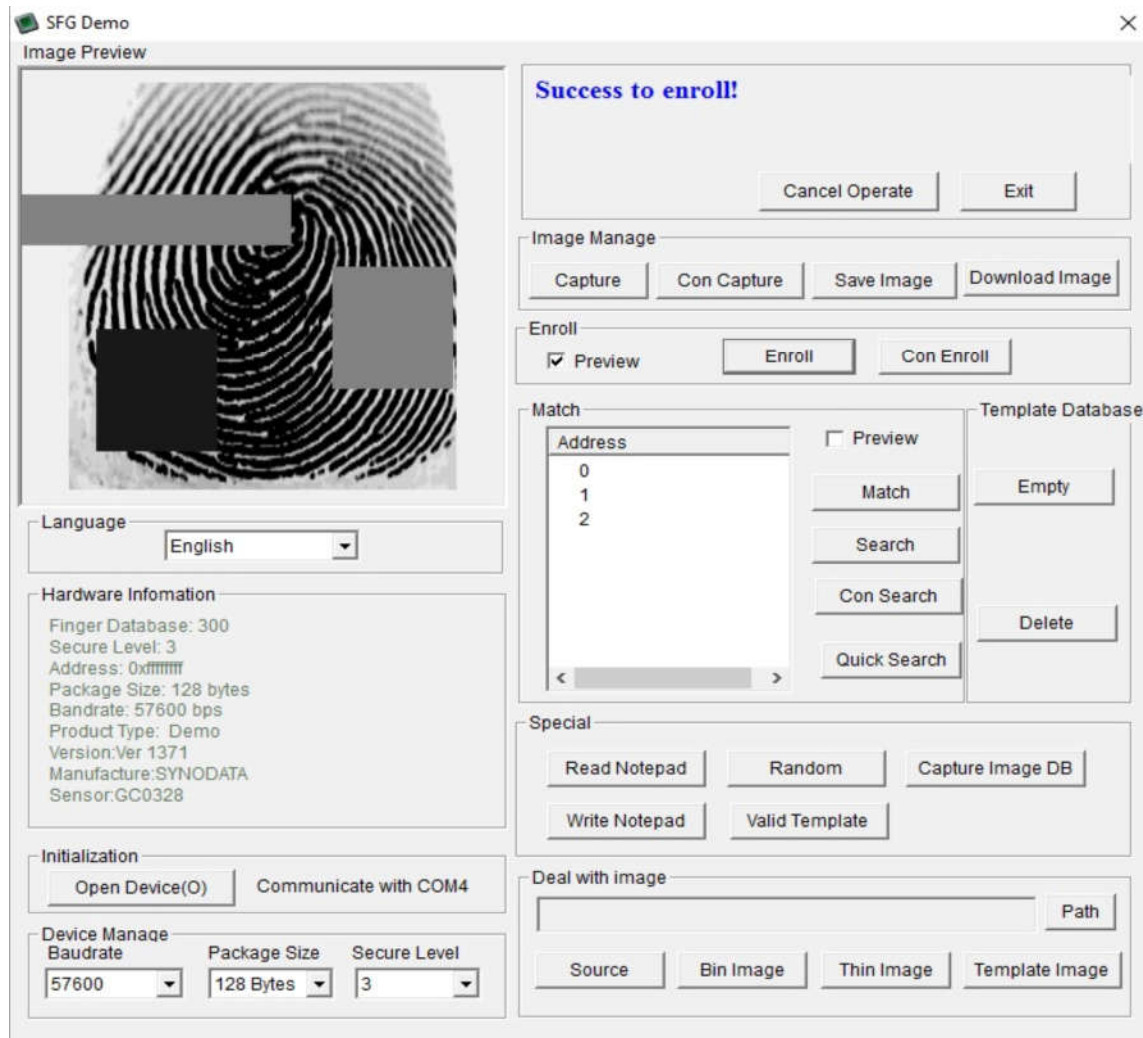
Wiring Connections:

Gizduino to FPS

| | |
|--------------|-----------------------|
| GND | Black wire (1) |
| D0/RX | Green wire (2) |
| D1/TX | Blue wire (3) |
| +3.3V | Red wire (4) |

Figure 3:
Sample Diagram to
communicate with COM#
using SFG Demo Software.





1. Once you've done constructing the wiring connections and uploading the code. Click "Open Device(O)" then a new window will show SELECT the COM PORT NO.

Note: If it is doesn't connecting, we advised to change your COM port. or if it is up to COM10 to COM99, change it to COM2 to COM4.

**2. If it is connected, and successfully "Communicate with COM#". Set up this on the "Device Manage" section.
Baudrate: 57600
Package Size: 128Kbytes
Secure Level: 3**

3. In "Enroll" section. Check the "Preview", then Click "Enroll". Another window will show up, if the address has already been taken. its up to you to overwrite on it or select/ type your address no. you want.

Just click ok and wait ,up to 8 - 10secs until the transferring is done and for the second attempt for finishing. The message will show up "Success to enroll!".