



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China





## Adafruit Optical Fingerprint Sensor

Created by lady ada



Last updated on 2016-03-07 06:04:10 PM EST

## Guide Contents

Guide Contents	2
Overview	3
Enrolling vs. Searching	5
Enrolling New Users with Windows	6
Searching with the Software	11
Wiring for use with Arduino	12
Enrolling with Arduino	15
Downloads	17
Support Forums	18

# Overview



Secure your project with biometrics - this all-in-one optical fingerprint sensor will make adding fingerprint detection and verification super simple. These modules are typically used in safes - there's a high powered DSP chip that does the image rendering, calculation, feature-finding and searching. Connect to any microcontroller or system with TTL serial, and send packets of data to take photos, detect prints, hash and search. You can also enroll new fingers directly - up to 162 finger prints can be stored in the onboard FLASH memory. There's a red LED in the lens that lights up during a photo so you know its working.

We like this particular sensor because not only is it easy to use, it also comes with fairly straight-forward Windows software that makes testing the module simple - you can even enroll using the software and see an image of the fingerprint on your computer screen. But, of course, we wouldn't leave you a datasheet and a "good luck!" - [we wrote a full Arduino library so that you can get running in under 10 minutes. The library can enroll and search so its perfect for any project \(<http://adafru.it/aRz>\)](#). We've also [written a detailed tutorial on wiring and use \(<http://adafru.it/clz>\)](#). This is by far the best fingerprint sensor you can get.

- **Supply voltage:** 3.6 - 6.0VDC

- **Operating current:** 120mA max
- **Peak current:** 150mA max
- **Fingerprint imaging time:** <1.0 seconds
- **Window area:** 14mm x 18mm
- **Signature file:** 256 bytes
- **Template file:** 512 bytes
- **Storage capacity:** 162 templates
- **Safety ratings** (1-5 low to high safety)
- **False Acceptance Rate:** <0.001% (Security level 3)
- **False Reject Rate:** <1.0% (Security level 3)
- **Interface:** TTL Serial
- **Baud rate:** 9600, 19200, 28800, 38400, 57600 (default is 57600)
- **Working temperature rating:** -20C to +50C
- **Working humidity:** 40%-85% RH
- **Full Dimensions:** 56 x 20 x 21.5mm
- **Exposed Dimensions** (when placed in box): 21mm x 21mm x 21mm triangular
- **Weight:** 20 grams

# Enrolling vs. Searching

There are basically two requirements for using the optical fingerprint sensor. First is you'll need to **enroll** fingerprints - that means assigning ID #'s to each print so you can query them later. Once you've enrolled all your prints, you can easily 'search' the sensor, asking it to identify which ID (if any) is currently being photographed.

You can enroll using the windows software (easiest and neat because it shows you the photograph of the print) or with the Arduino sketch (good for when you don't have a windows machine handy or for on-the-road enrolling).

# Enrolling New Users with Windows

The easiest way to enroll a new fingerprint is to use the Windows software. The interface/test software is unfortunately windows-only *but* you only need to use it once to enroll, to get the fingerprint you want stored in the module.

First up, you'll want to connect the sensor to the computer via a USB-serial converter. The easiest way to do this is to connect it directly to the USB/Serial converter in the Arduino. To do this, you'll need to upload a 'blank sketch' this one works well for "traditional" Arduinos, like the Uno and the Mega:

```
// this sketch will allow you to bypass the Atmega chip
// and connect the fingerprint sensor directly to the USB/Serial
// chip converter.

// Red connects to +5V
// Black connects to Ground
// White goes to Digital 0
// Green goes to Digital 1

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

The "blank" sketch won't work for ATmega32u4-based Arduinos like the Leonardo and the Micro! Use the `Leo_passthru` sketch instead!

If you're using a Leonardo, Micro, Yun, or other ATmega32U4-based controller, use this `Leo_passthru` sketch instead of the "blank" sketch.

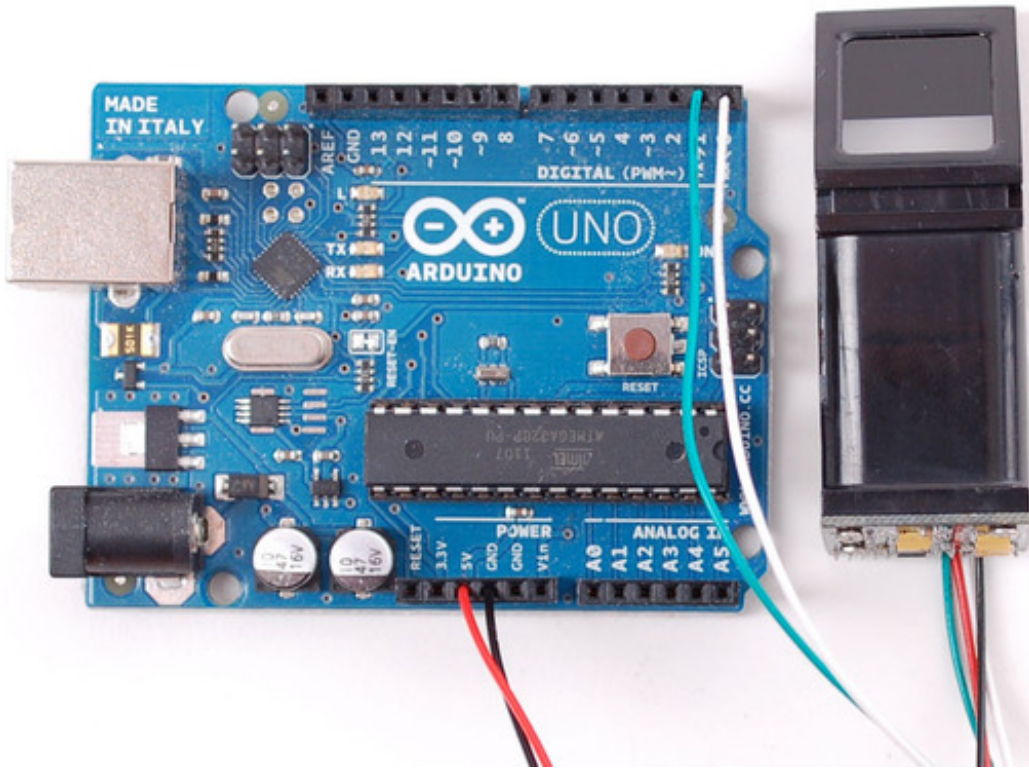
```
//Leo_passthru
// Allows Leonardo to pass serial data between
// fingerprint reader and Windows.
//
// Red connects to +5V
// Black connects to Ground
// Green goes to Digital 0
// White goes to Digital 1

void setup() {
  Serial1.begin(57600);
  Serial.begin(57600);
}

void loop()
```

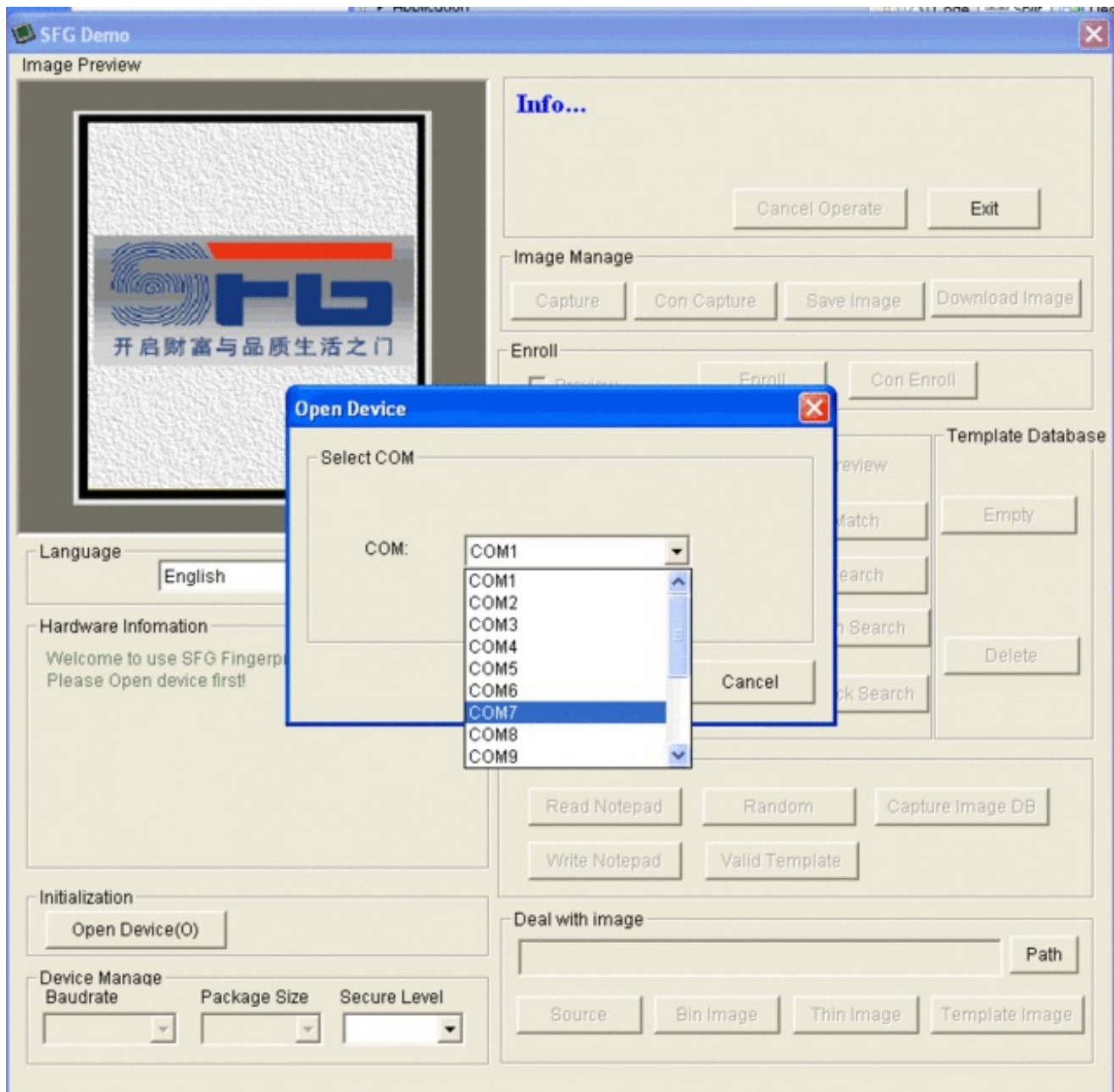
```
{  
  while (Serial.available())  
    Serial1.write(Serial.read());  
  while (Serial1.available())  
    Serial.write(Serial1.read());  
}
```

Wire up the sensor as described in the sketch comments **after** uploading the sketch. Since the sensor wires are so thin and short, we stripped the wire a bit and melted some solder on so it made better contact but you may want to solder the wires to header or similar if you're not getting good contact. When you plug in the power, you should see the red LED blink to indicate the sensor is working.

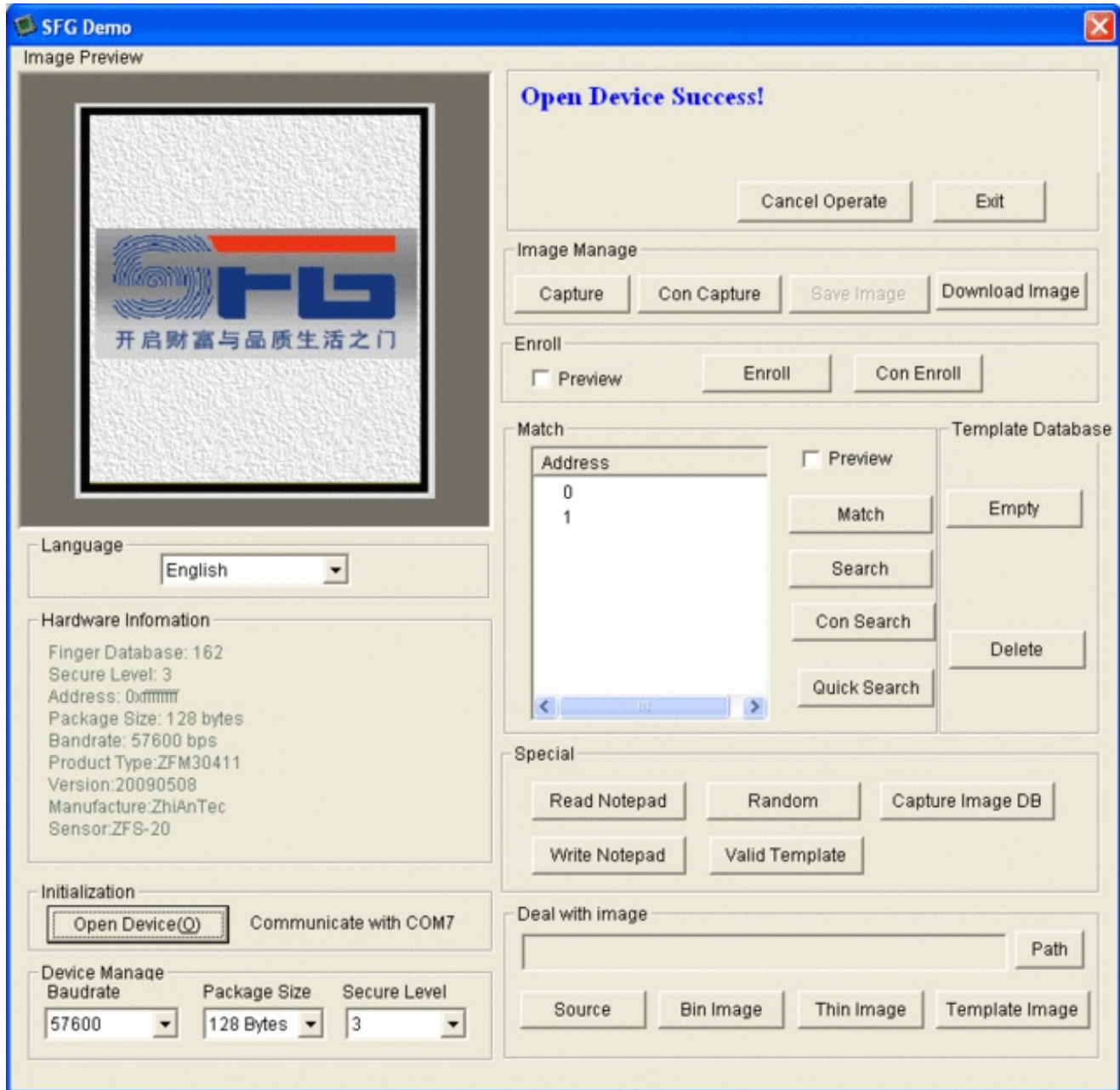


Start up the SFGDemo software and click **Open Device** from the bottom left corner. Select the **COM port** used by the Arduino.

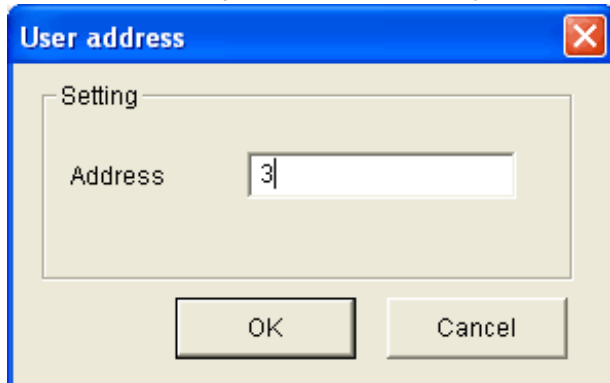




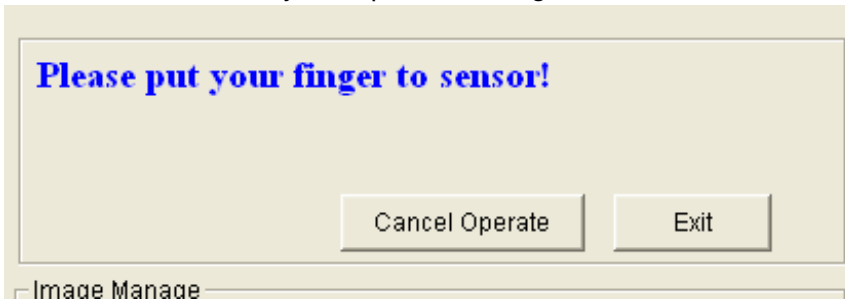
And press OK when done. You should see the following, with a blue success message and some device statistics in the bottom corner. You can change the baud rate in the bottom left hand corner, as well as the "security level" (how sensitive it is) but we suggest leaving those alone until you have everything running and you want to experiment. They should default to 57600 baud and security level 3 so set them if they're wrong



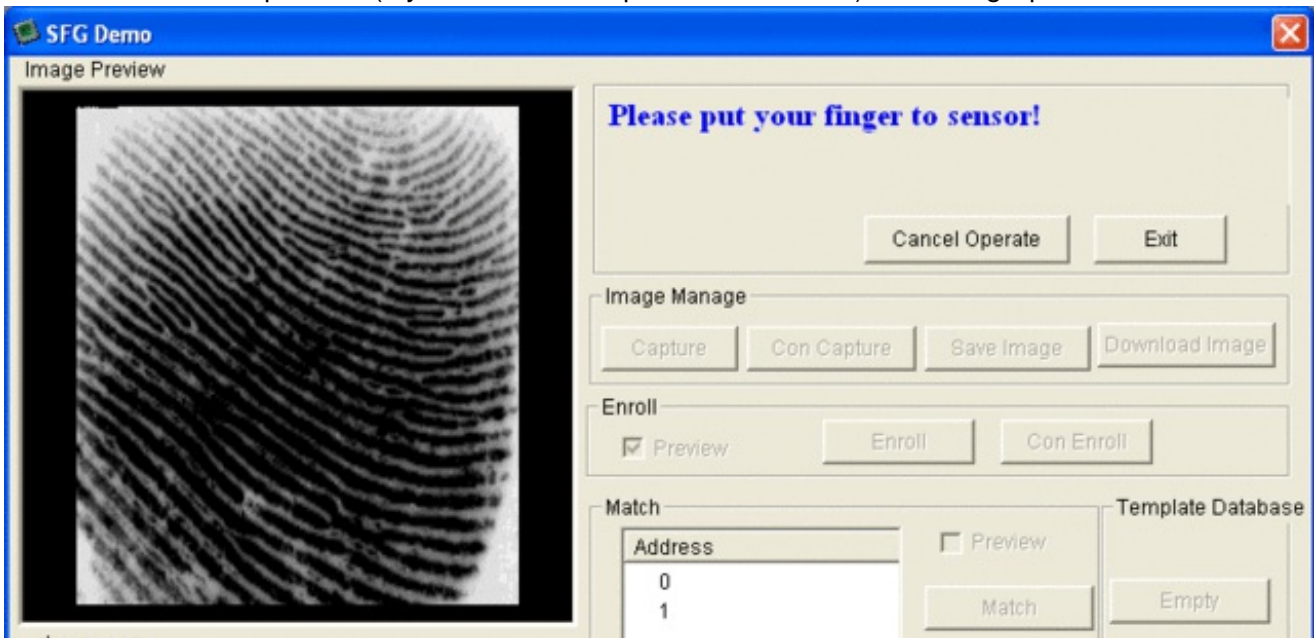
Lets enroll a new finger! Click the **Preview** checkbox and press the **Enroll** button next to it (**Con Enroll** means 'Continuous' enroll, which you may want to do if you have many fingers to enroll). When the box comes up, enter in the ID # you want to use. You can use up to 162 ID numbers.



The software will ask you to press the finger to the sensor

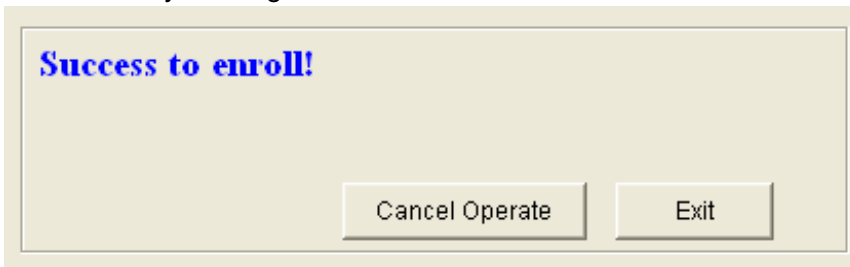


You can then see a preview (if you clicked the preview checkbox) of the fingerprint.



You will then have to repeat the process, to get a second clean print. Use the same finger!

On success you will get a notice.



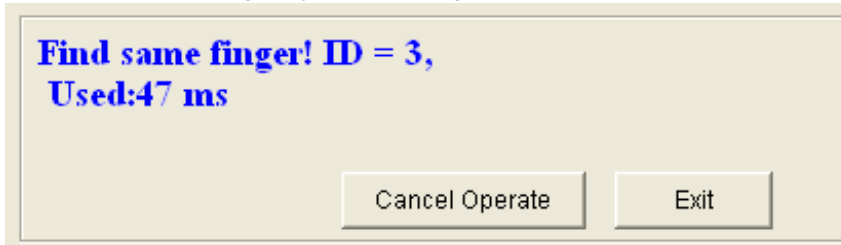
If there's a problem such as a bad print or image, you'll have to do it again.

# Searching with the Software

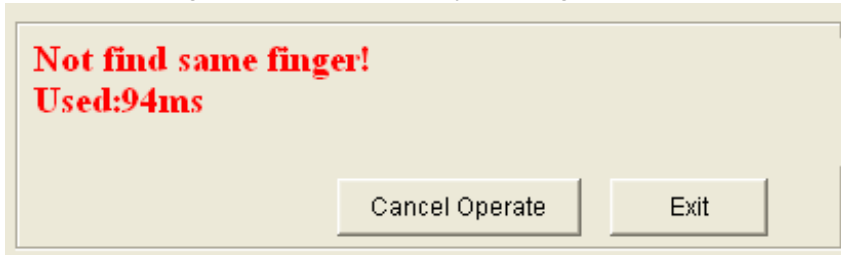
Once you have the finger enrolled, it's a good idea to do a quick test to make sure it can be found in the database. Click on the **Search** button on the right hand side.

When prompted, press a different/same finger to the sensor.

If it is the same finger, you should get a match with the ID #



If it is not a finger in the database, you will get a failure notice.



# Wiring for use with Arduino

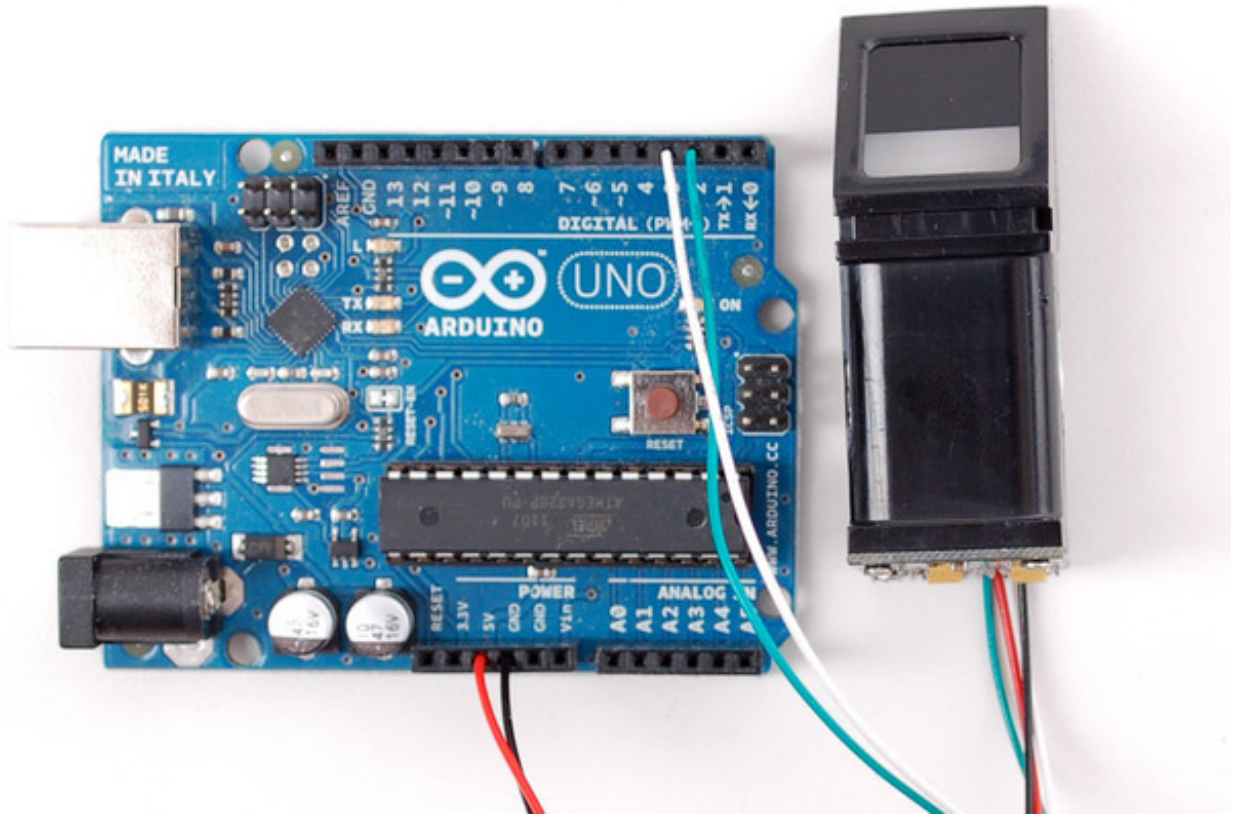
Once you've tested the sensor, you can now use it within a sketch to verify a fingerprint. We'll need to rewire the sensor. Disconnect the green and white wires and plug the green wire into digital **2** and the white wire to digital **3**. You can change these pins later but for now, use the default pins.

**In the diagrams below we show the wires plugged directly into the Arduino. However, this does not work well because the wires are so thin and they don't make contact. You should solder thicker solid core wires to each wire, to make good contact**

**It is normal for the sensor to blink the red LED quickly once powered, after that the LED will be off until you've started to request data from it**

Leonardo users: The Leonardo does not support software serial on pins 2 & 3. Use pins 8 and 9 and change the example code to match.

Mega users: The Mega does not support software serial on pins 2 & 3. Use pins 10 and 11 and change the example code to match.



Next, [download the Adafruit Fingerprint sensor library from github \(http://adafruit.it/aRz\)](http://adafruit.it/aRz). To download click the ZIP button in the top bar, rename the uncompressed folder **Adafruit\_Fingerprint**. Check that the **Adafruit\_Fingerprint** folder contains **Adafruit\_Fingerprint.cpp** and **Adafruit\_Fingerprint.h**. Place the **Adafruit\_Fingerprint** library folder your **/libraries/** folder. You may need to create the libraries subfolder if its your first library. Restart the IDE.

Once you've restarted you should be able to select the **File**→**Examples**→**Adafruit\_Fingerprint**→**fingertest** example sketch. Upload it to your Arduino as usual. Open up the serial monitor at 9600 baud and when prompted place your finger against the sensor that was already enrolled.

You should see the following:

```
COM7
fingertest
Found fingerprint sensor!
Waiting for valid finger...
Found ID #3 with confidence of 157
```

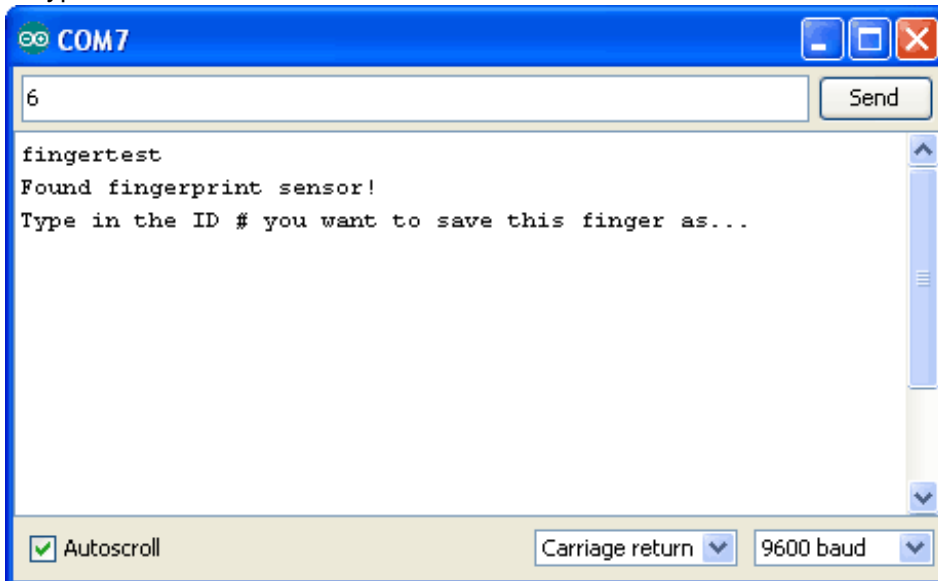
The 'confidence' is a score number (from 0 to 255) that indicates how good of a match the print is, higher is better. Note that if it matches at all, that means the sensor is pretty confident so you don't have to pay attention to the confidence number unless it makes sense for high security applications.

If you want to have a more detailed report, change the **loop()** to run **getFingerprintID()** instead of **getFingerprintIDez()** - that will give you a detailed report of exactly what the sensor is detecting at each point of the search process.

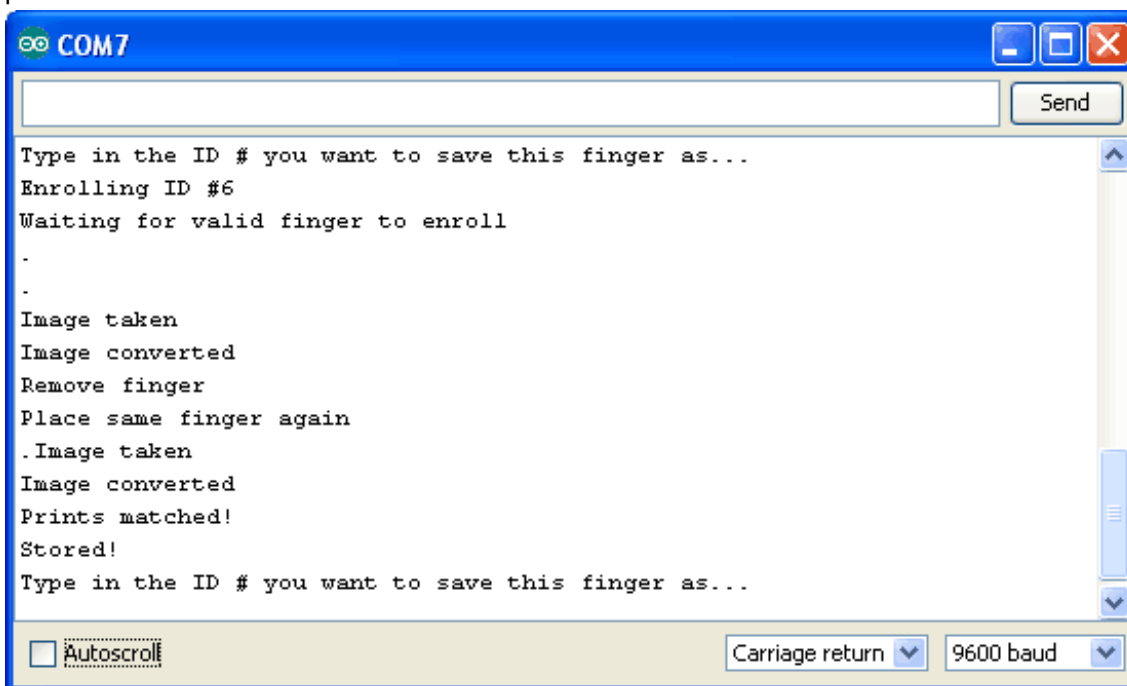
# Enrolling with Arduino

We did put together a simple sketch for enrolling a new finger via Arduino - its not as easy to use as the Windows program but it does work. Run the **File**→**Examples**→**Adafruit\_Fingerprint**→**enroll** sketch and upload it to the Arduino, use the same wiring as above.

When you open up the serial monitor, it will ask for you to type in the ID to enroll - use the box up top to type in a number and click Send.



Then go through the enrollment process as indicated. When it has successfully enrolled a finger, it will print **Stored!**





Don't forget to do a search test when you're done enrolling to make sure its all good!

# Downloads

- [Arduino interface library on github \(http://adafru.it/aRz\)](http://adafru.it/aRz)
- [User Manual \(http://adafru.it/aRD\)](http://adafru.it/aRD)
- [Datasheet \(its not really a great datasheet and its Chinese but its better than nothing\) \(http://adafru.it/aRB\)](http://adafru.it/aRB)
- [English version of the User Manual \(http://adafru.it/aRD\)](http://adafru.it/aRD)
- ["SFGDemo" Windows-only test software \(http://adafru.it/aRC\)](http://adafru.it/aRC)

# Support Forums

Support Forums (<http://adafru.it/forums>)