Book of Specifications

Project: Biometric Fingerprint Authentication

Yassine Khedher

[Khedheryassine268@gmail.com](mailto:Khedheryassine268@gmail.com)

ISTIC\_IOT-2A

**Introduction**

Not only in movies, fingerprint readers are seen more and more frequently. Such modules are often installed in home surveillance systems and are used for the simple but secure verification of persons. With such a Raspberry Pi Fingerprint Sensor you can also implement some other projects, such as secured hard drives, door locks etc.

One of the advantages is that passwords and / or number codes can be completely omitted. Although this is still possible, but it’s a lot more comfortable without. This project is about the connection as well as the reading, saving and verifying of imported fingerprints.

**Accessories**

* any CircuitPython microcontroller board or a computer that has GPIO and Python
* Fingerprint Sensor (check voltage)
* Serial USB converter with 3.3V and 5V connection
* Female-Female Jumper wires

In this Project, we are going to be using Raspberry Pi card (any model) with an AdaFruit Fingerprint Sensor using CircuitPython programming language

**Connection of the Fingerprint Sensor**

The USB adapter is labeled, but the fingerprint sensor cables are not. However, the cables have a clear color, which we can identify and connect to the USB converter. We only need four of the cables (if your fingerprint sensor has more, you can ignore the remaining colors):

* Red: Depending on the accepted voltage of the sensor (3.3V or 5V).
* White: RXD (Receiver).
* Green: TXD (Transmitter).
* Black: GND (Ground).

If your sensor needs a higher voltage than 3.3V (and the maximum value is equal to or greater than 5V), you can connect the red cable to the 5V pin.

**CircuitPython Fingerprint Library Installation**

To use the Fingerprint sensor, you'll need to install the Adafruit CircuitPython Fingerprint library on your CircuitPython board by running this command in the Console:

**pip3 install adafruit-circuitpython-fingerprint**

First make sure you are running the latest version of Adafruit CircuitPython for your board.

Next, you'll need to install the necessary libraries to use the hardware--carefully follow the steps to find and install these libraries from Adafruit's CircuitPython library bundle

Copy the necessary file from the library bundle to the **lib** folder on your **CIRCUITPY** drive:

**adafruit\_fingerprint.mpy**

Before continuing make sure your board's lib folder has the **adafruit\_fingerprint.mpy** file copied over.

Next connect to the board's serial REPL so you are at the CircuitPython >>> prompt.

**Python Installation of Fingerprint Library**

You'll need to install the Adafruit\_Blinka library that provides the CircuitPython support in Python.

Once that's done, from your command line run the following command:

**sudo pip3 install adafruit-circuitpython-fingerprint**

If your default Python is version 3 you may need to run 'pip' instead. Just make sure you aren't trying to use CircuitPython on Python 2.x, it isn't supported!

Now you can run the program with the following command:

**python3 fingerprint\_simpletest.py**

When you first start it up, you should get something like this:



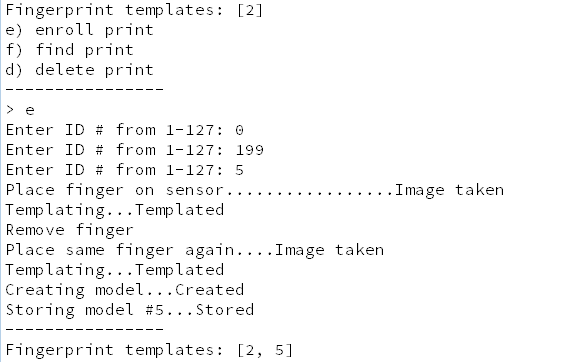
If you get an error like RuntimeError: Failed to read data from sensor it means something went wrong - check your wiring and baud rate!

This menu system is fairly simple, you have three things you can do

* Enroll print - you will use your finger to take images and 'store' the model in the sensor
* Find print - determine whether a fingerprint is known and stored
* Delete print - clear out a model

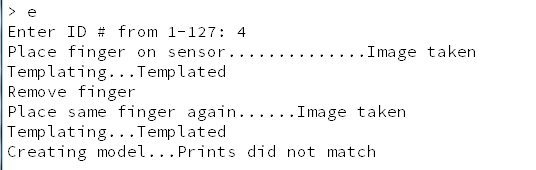
**Enrolling Prints**

Enrolling a finger print is easy. Type **e** to start the process. You'll need to select a location. The sensor can store up to 127 print locations. Pick a valid number, then place your finger twice to enroll.



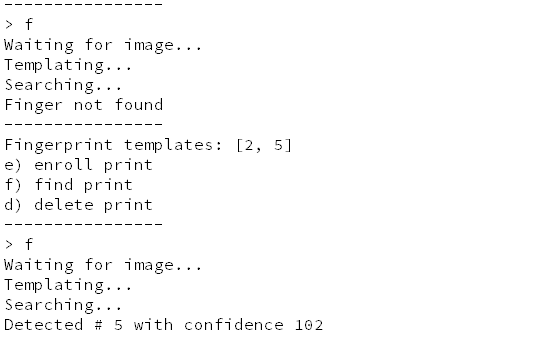
Note that after success, the **Fingerprint templates: [...]** printout will include the new template id.

If an error occurs, the sensor will give you an error, such as if the two prints don't match, or if it failed to store or generate a model:



**Finding Prints**

Once you've enrolled fingerprints you can then test them. Run the **f**ind command, and try various fingers! Once the fingerprint id identified it will tell you the location number, in this case **#5**



**Deleting Fingerprints**

If you made a mistake, you can remove fingerprint models from the database. For example, here's how to delete #5. Note the **Fingerprint templates: [...]** printout changes!

