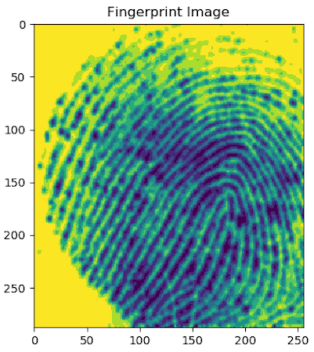
Fingerprint Password Manager



Jim McKeown 2023

Normal Operation

In normal use, the password manager box is connected to a USB port of the host computer and appears as an additional keyboard. The box waits for a valid fingerprint that has already been enrolled and if the fingerprint is recognized, sends the associated text (password) to the host.

Since most people have only ten fingers and some of those fingers may not have good readable fingerprints, the total number of possible fingers for an individual is limited to ten or less. Most people also have more than ten passwords. To work around this limitation, there are a number of things that can be done. One possibility is to use the box to store the master password of your online password manager such as BitWarden. All other passwords can then be accessed once your vault is unlocked. You will also need a few other offline passwords such as to initially login to your operating system account and to access your computer's BIOS.

Another option is to use each finger to store a password fragment and assemble the fragments at login time. For example, using three fingers: 0 - left thumb, 1 - right thumb, 2 - right index. Enroll whatever good fingers you have with random password fragments. Include upper and lower case letters and at least one number and one special character in each fragment:

0: aPwh7&
1: odBe3#
2: qmXt4?

Here is a partial list of results:

Account	Combination	Password
1	012	aPwh7&odBe3#qmXt4?
2	120	odBe3#qmXt4?aPwh7&
3	201	qmXt4?aPwh7&odBe3#
4	210	qmXt4?odBe3#aPwh7&
5	021	aPwh7&qmXt4?odBe3#
6	102	odBe3#aPwh7&qmXt4?
7	000	aPwh7&aPwh7&aPwh7&
8	001	aPwh7&aPwh7&odBe3#
9	002	aPwh7&aPwh7&qmXt4?

Using three fingers, there is a total of 27 (3^3) combinations. If you have ten readable fingers and use three fragments, you have 1000 (10^3) possible combinations. You just have to remember the finger sequence for each account.

Some variation of the above examples or something else may be suitable for you.

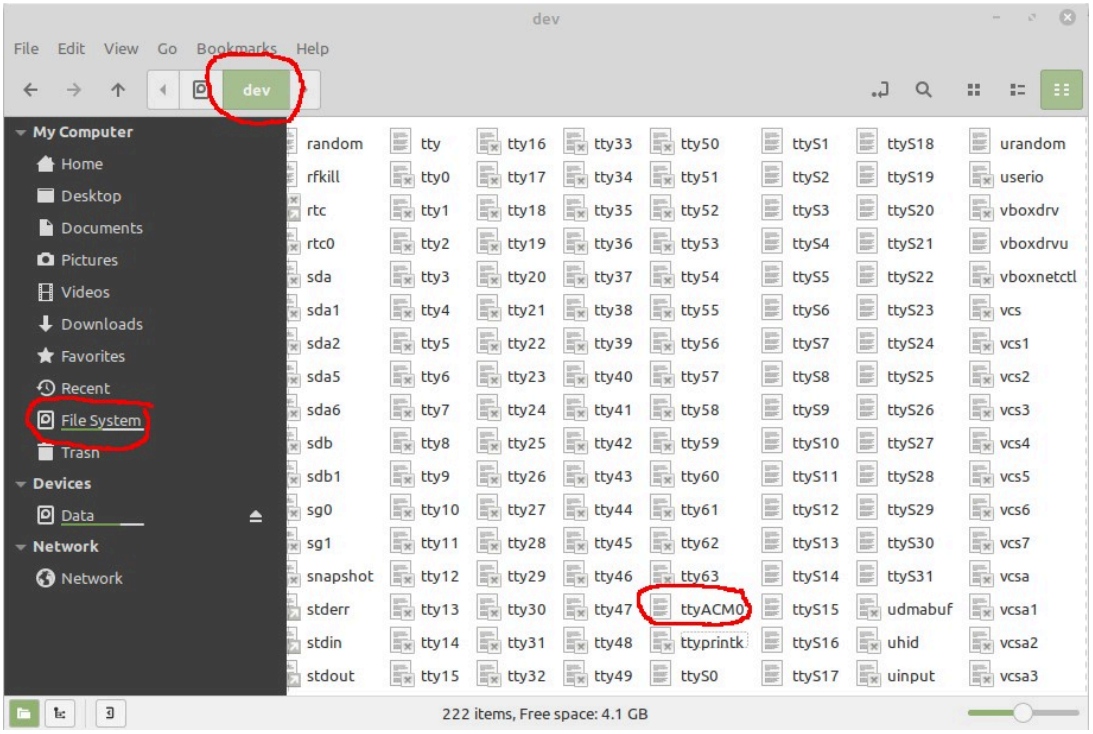
Connect a serial terminal emulator

Using a serial terminal emulator, you can:

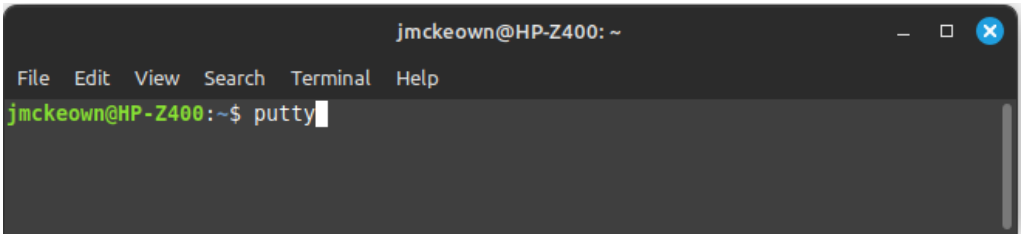
- Enroll new fingerprints
- Reset the device to clear all fingerprints and passwords
- Switch the device to passthru mode (communicate directly with the fingerprint scanning module)
- Edit passwords without disturbing fingerprint models
- Unenroll (delete) individual fingerprint models
- Get help and fingerprint module setting information

To setup PuTTY under Linux:

1. Plug the USB cable into a PC.
2. Determine the device assigned to the password manager. In Linux, open /dev directory. The device is likely ttyACMx. If there is more than one (ttyACM0, ttyACM1...) unplug the USB cable and see which device is removed. Then plug the USB cable back in and confirm that the device returns.



3. In Terminal, open a serial terminal emulator application such as PuTTY.

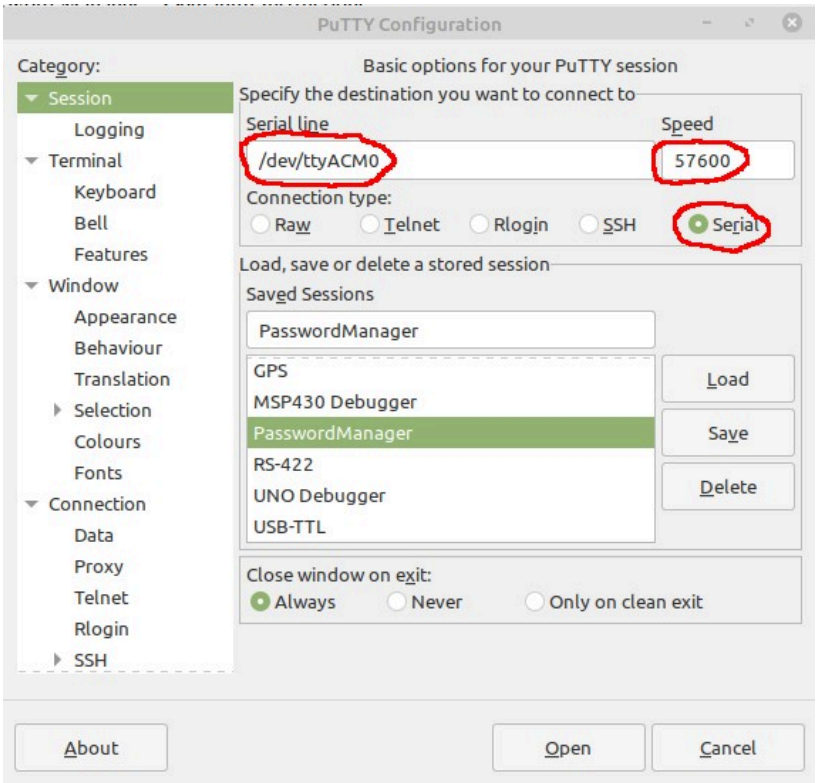


4. Change Session settings to:

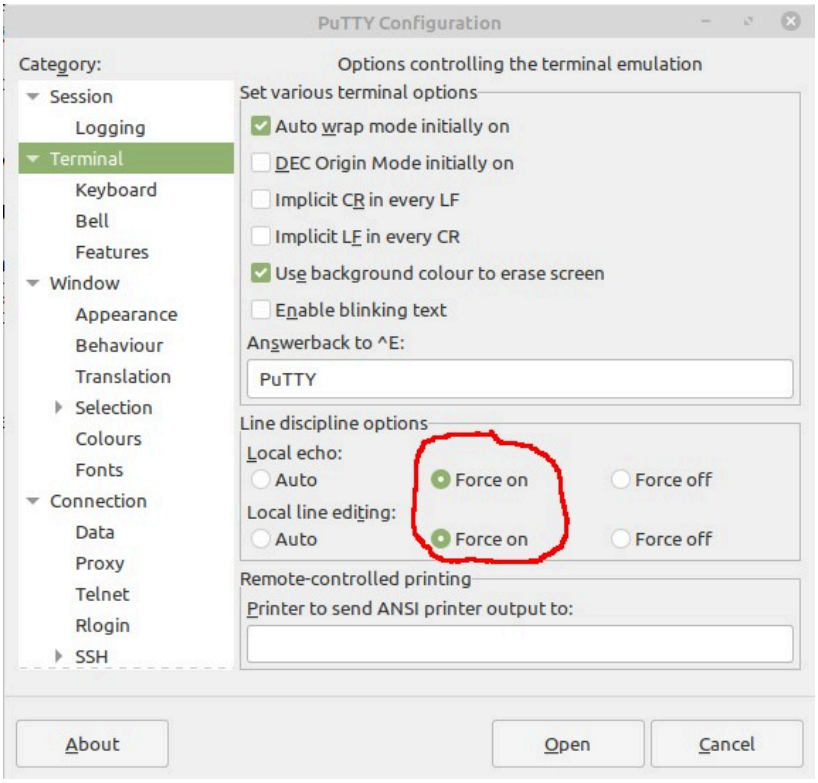
   Destination: dev/ttyACM0 (or the device from step 2)

   Speed 57600

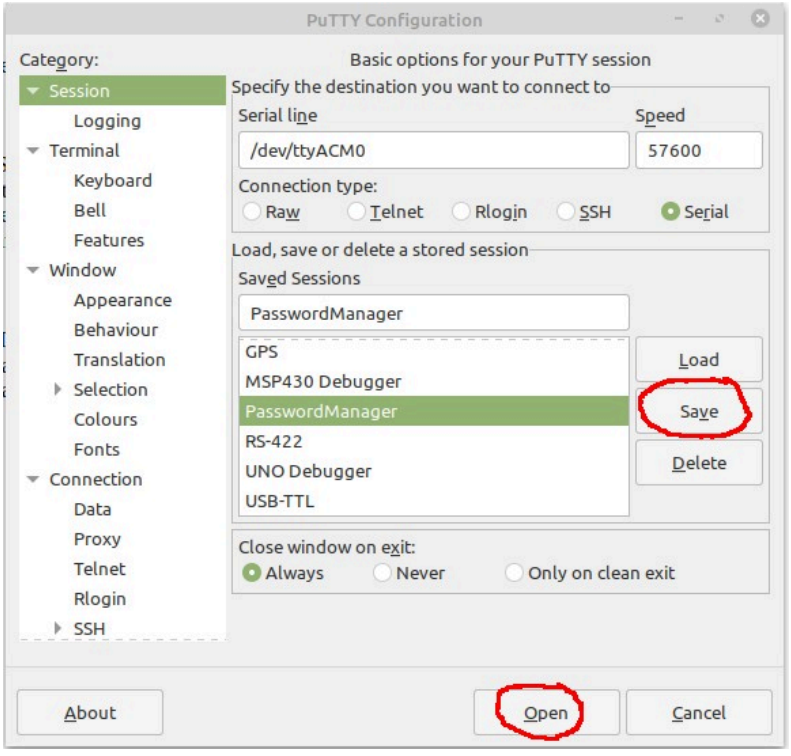
   Connection type: Serial



5. Change Terminal settings to:
   Local echo: Force on
   Local line editing: Force on
   All other settings are default.

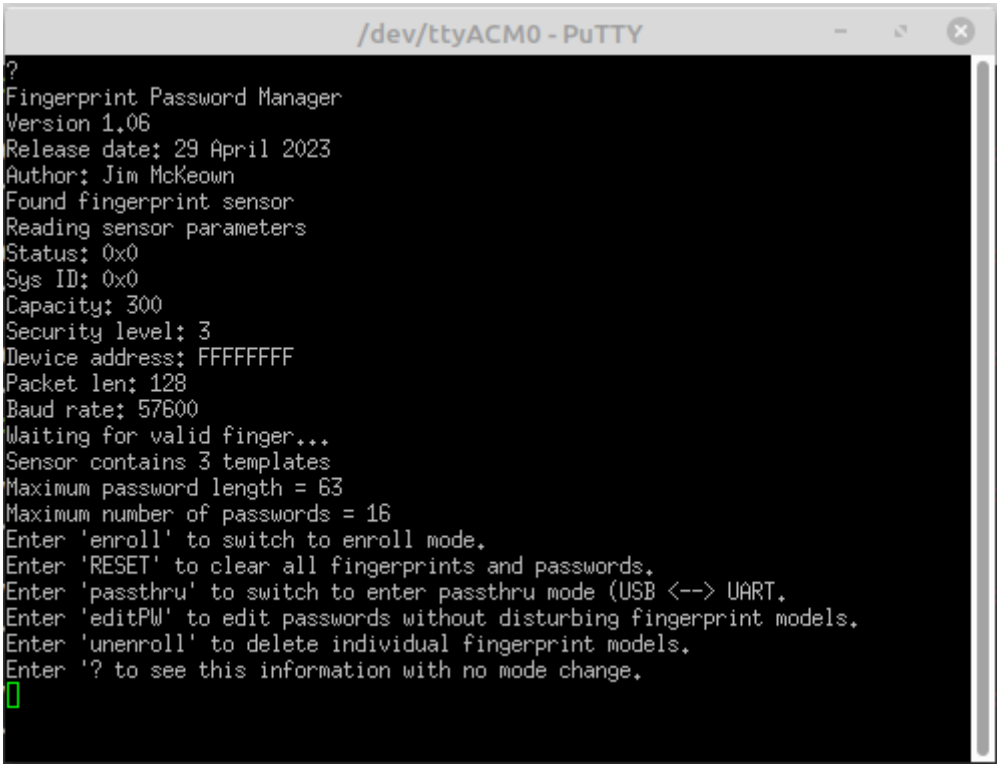


6. Save the configuration with a meaningful name so you can skip steps 2, 4, and 5 next time. Load the
   configuration if you have already saved it.



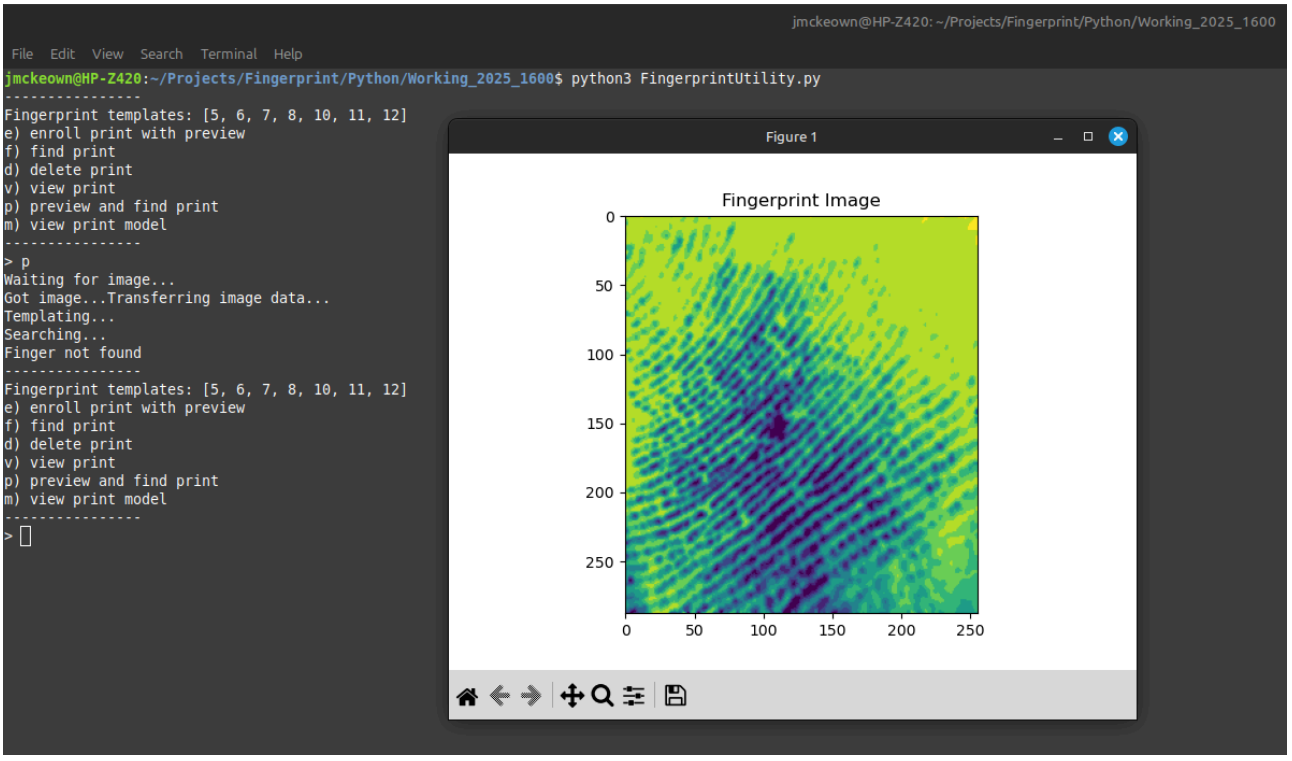
7. Select “Open”.

8. Enter "?". You should get a response similar to the example below.



Follow the instructions shown in the terminal window as in the example above.

Python Utility



If you would like to have visual feedback of the quality of the captured fingerprint image, you can use the Python open-source utility included in this project for the enroll process. The Python utility automatically puts the box in 'passthru' mode to send and receives USB data directly to and from the fingerprint scanning module. The box stays in 'passthru' mode until it is unplugged.

Download files 'FingerprintUtility.py' and 'adafruit_fingerprint.py' from the project repository: https://github.com/jim-mckeown/fingerprint-scanner-arduino/

Put both files in the same folder.

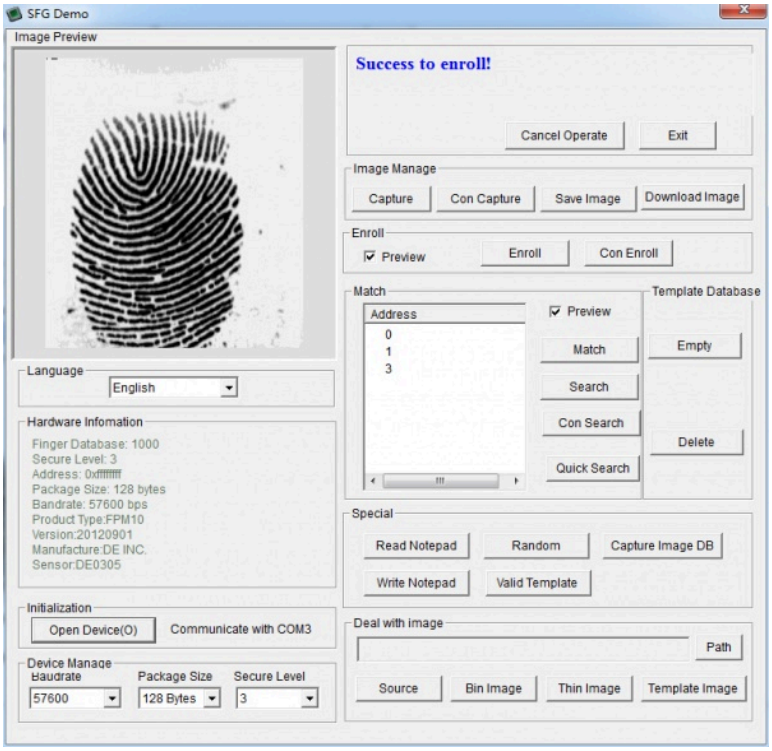
In a terminal window, navigate to the folder with the two files.

Run: python3 FingerprintUtility.py

Follow the prompts.

The Python utility should work under Windows but has only been tested under Linux.

SFG Demo Windows Application



If you are running Windows, there is a closed-source utility that shows the captured fingerprint image. Open PuTTY before running this program to put the box in 'passthru' mode to allow direct communication with the fingerprint module.

You can download the application here:

http://www.adafruit.com/datasheets/SFGDemoV2.0.rar

Extract SFGDemoV2.0.rar and run SFGDemo.exe