Finger Print Sensor, which we used to verify the identity of a person for various purposes. Nowadays we can see fingerprint-based systems everywhere in our daily life.

One of the advantage of fingerprint-based systems is that passwords and / or number codes can be completely omitted.

**Hardware Requirements**

1. Raspberry Pi Model A/B/B+
2. Fingerprint Module
3. Serial USB Converter
4. Jumper Wires

Here, I am using **R307 fingerprint module** with **Serial USB Converter.**

R307 Fingerprint Module consists of optical fingerprint sensor, Supply voltage: DC 4.2 ~ 6.0V

|  |  |  |
| --- | --- | --- |
| **Pin No** | **Pin Name** | **Details** |
| **1** | 5V | Regulated 5V DC |
| **2** | GND | Common Ground |
| **3** | TXD | Data output - Connect to MCU RX |
| **4** | RXD | Data Input - Connect to MCU TX |
| **5** | TOUCH | Active Low output when there is touch on sensor by finger |
| **6** | 3.3V | Use this wire to give 3.3V to sensor instead of 5V |

# Connect fingerprint module to USB Serial converter.

|  |  |
| --- | --- |
| **Fingerprint Module** | **USB Serial Converter** |
| **5V** | **5.0 V** |
| **GND** | **GND** |
| **TXD** | **TXD** |
| **RXD** | **RXD** |
| **TOUCH** | - |
| **3.3V** | - |

Note: Check USB to Serial Converter pins according to your model. And connect accordingly.

* **Software Requirements**

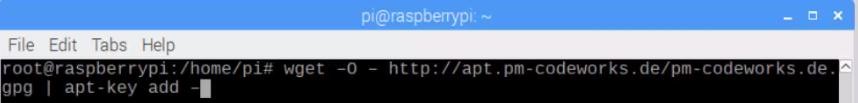
1. Raspbian Stretch OS

# 1. Now, just connect fingerprint module to Raspberry Pi USB port by using USB to Serial converter.

**Step 1: To install this library, root privileges are required. So login with root user.**



# Step 2: Download some required packages using wget command

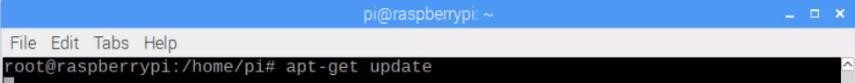


wget –O – <http://apt.pm-codeworks.de/pm-codeworks.de.gpg>| apt-key add –

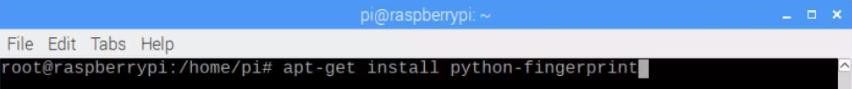
wget [http://apt.pm-codeworks.de/pm-codeworks.list -P /etc/apt/sources.list.d/](http://apt.pm-codeworks.de/pm-codeworks.list%20-P%20/etc/apt/sources.list.d/)



**Step 3: Update the Raspberry Pi**



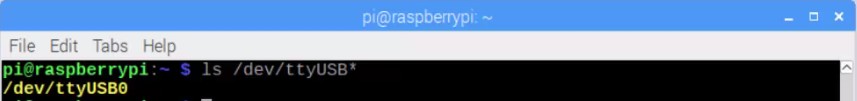
# Step 4: Install the downloaded finger print sensor library



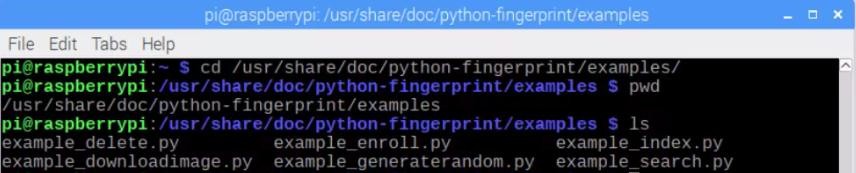
**Step 5: To return to the normal shell (under the** Pi user**), type** exit



**Step 6: Now check** USB port **on which your** finger print sensor **is connected.** Use this **USB port** in our **Python script.**



# Step 7: Now go to the examples directory. (/usr/share/doc/python-fingerprint/examples/)



**Step 8: Run sample file, to test to see if the sensor is detected and ready for access**



The above data should appear, which allows you to display the positions under which an imprint is stored by selecting a page (0-3).

If here you get **Exception message**, then something is wrong with the cabling or the sensor. Check it again.

**Step 9:** Now execute other scripts, to make sure Fingerprint module is working.

|  |  |
| --- | --- |
| **Script** | **Usage** |
| **example\_index.py** | **Shows template index table.** |
| **example\_enroll.py** | **Stores new fingerprint** |
| **example\_delete.py** | **Deletes a fingerprint from sensor** |
| **example\_search.py** | **Search for recorded fingerprint** |
| **example\_downloadimage.py** | **Read fingerprint and download it.** |
| **example\_generaterandom.py** | **Generates 32 bit random number.** |

Run **example\_enroll.py** script to store new fingerprint



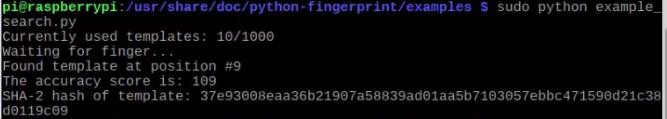
Put your finger on the glass surface, wait for the instruction in the terminal and remove your finger as soon as it is written there. Afterwards you have to put your finger a second time for the verification and the imprint is stored in the next number.

Now, Run **example\_search.py** script to see whether our finger is recognized.



Put the **same finger** on glass surface.

If the fingerprint is detected, it displays below message.



Execute same script again, this time use other finger which is not previously stored. If fingerprint is not detected, then gives **"No match Found"** message



**So, our fingerprint module is working properly.**