Practical No: 06

**Aim:** Fingerprint Sensor interfacing with Raspberry Pi.

**Hardware Required:**

1. Raspberry Pi 3B+
2. Ethernet Cable
3. Monitor
4. HDMI to VGA convertor
5. Micro SD card (any class, best is class 10)
6. Adaptor with 5v 2A
7. USB mouse
8. USB keyboard
9. LEDs (Green, Red)
10. Figure Print Sensor (R370)
11. A USB TTL adapter
12. Buzzer

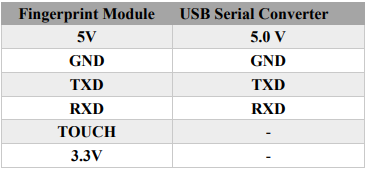
**Software Required:**

1. Thonny Python

**Procedure:**

1. **Hardware Setup:**

* Connect according to the figure.



****

1. **Software Setup:**

* Open Terminal and type in the following commands
  + sudo bash
* 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/)
* apt-get update
* apt-get install python-fingerprint --yes
* Open Thonny Python and type the following code

*from time import sleep*

*from gpiozero import LED*

*from pyfingerprint.pyfingerprint import PyFingerprint*

*authorizedLed=LED(21)*

*unAuthorisedLed=LED(20)*

*index=0x01*

*try:*

*f = PyFingerprint('/dev/ttyUSB0', 57600, 0xFFFFFFFF, 0x00000000)*

*if ( f.verifyPassword() == False ):*

*raise ValueError('The given fingerprint sensor password is wrong!')*

*except Exception as e:*

*print('The fingerprint sensor could not be initialized!')*

*print('Exception message: ' + str(e))*

*exit(1)*

*print('Currently used templates: ' + str(f.getTemplateCount()) +'/'+ str(f.getStorageCapacity()))*

*def readFingurePrint(ind):*

*print('Waiting for finger...')*

*## Wait that finger is read*

*while ( f.readImage() == False ):*

*pass*

*f.convertImage(ind)*

*result = f.searchTemplate()*

*positionNumber = result[0]*

*return positionNumber*

*try:*

*positionNumber=readFingurePrint(index)*

*if ( positionNumber == -1 ):*

*print('No match found!')*

*unAuthorisedLed.blink(on\_time=0.2,off\_time=0.2)*

*sleep(10)*

*unAuthorisedLed.off()*

*else:*

*authorizedLed.blink(on\_time=0.6,off\_time=0.6)*

*print('Found template at position #' + str(positionNumber))*

*sleep(10)*

*except Exception as e:*

*print('Operation failed!')*

*print('Exception message: ' + str(e))*

*exit(1)*

