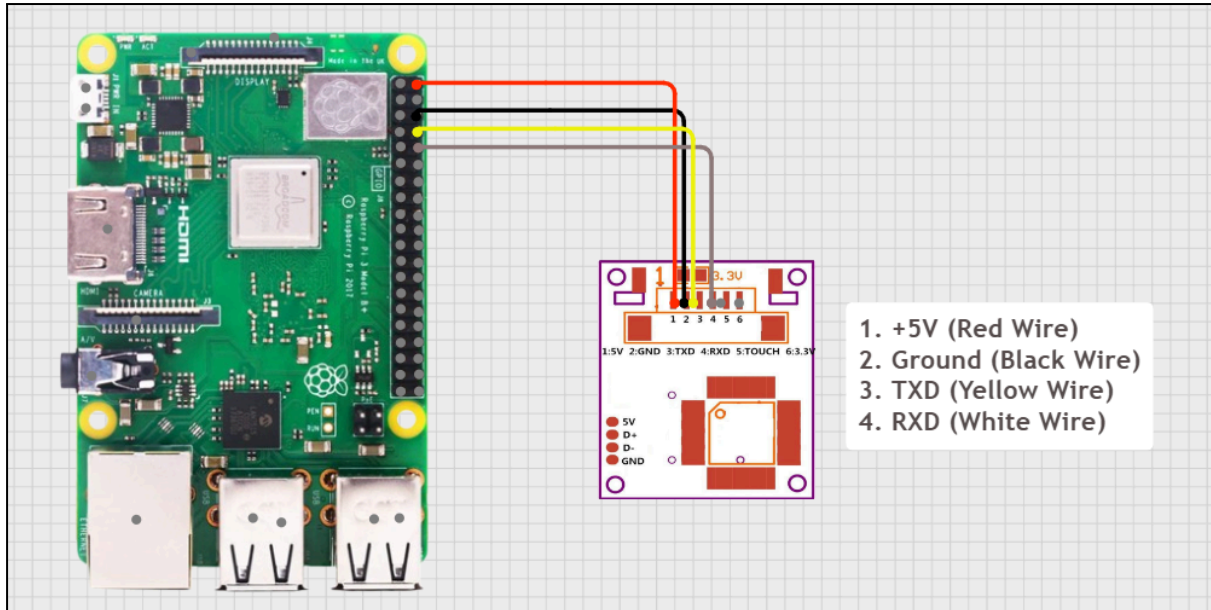


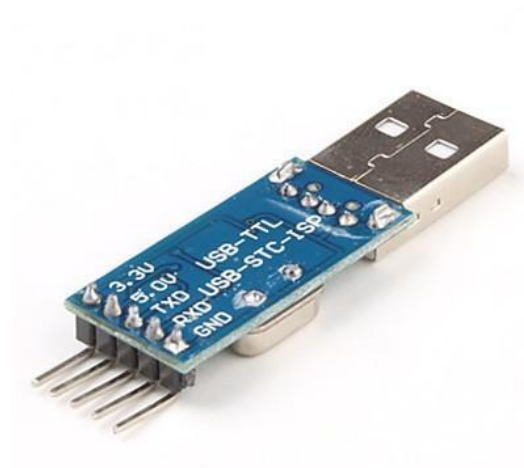
Practical 10

Aim: Fingerprint Sensor interfacing with Raspberry Pi



Pin Connections

Fingerprint module pins	Color	Wire	USB-TTL convertor
1: VCC	Red	Red	Connect it to USB-TTL(5V) : VCC
2: GND	Black	Black	Connect it to USB-TTL: GND
3: Tx	Yellow	Yellow	Connect it to USB-TTL: Rx
4: Rx	White	White	Connect it to USB-TTL: Tx



Enter the following commands in terminal.

- `sudo raspi-config`
- Go to Interfacing options -> I2C & SPI -> Enable it.
- `pip install pyfingerprint --break-system-packages`
- `ls /dev/ttyUSB*`

If required enter the following commands

- `sudo apt-get -f install`
- `sudo usermod -a -G dialout pi`

If any module error after doing these steps so do the below steps:

Install the module

`Sudo pip3 install pyfingerprint`

Then write the same above code in Python Thonny IDLE and save the Program with suitable name with the extension .py

Note : Don't forget to enable the the SPI and I2C.

Python Program

```
import time
from pyfingerprint.pyfingerprint import PyFingerprint
import RPi.GPIO as gpio # only needed if you later connect delete button

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('Exception message: ' + str(e))
exit(1)

# ----- Enroll New Finger -----
def enrollFinger():
    print('Waiting for finger...')
    while (f.readImage() == False):
        pass
    f.convertImage(0x01)
    result = f.searchTemplate()
    positionNumber = result[0]

    if (positionNumber >= 0):
        print('Finger already exists at position #' + str(positionNumber))
        time.sleep(2)
        return

    print('Remove finger...')
    time.sleep(2)

    print('Waiting for same finger again...')
    while (f.readImage() == False):
        pass

    f.convertImage(0x02)

    if (f.compareCharacteristics() == 0):
        print("Fingers do not match")
        time.sleep(2)
        return

    f.createTemplate()
    positionNumber = f.storeTemplate()
    print('Finger enrolled successfully!')
    print('New template position #' + str(positionNumber))
    time.sleep(2)

# ----- Search Finger -----
def searchFinger():
    try:
        print('Waiting for finger...')

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

        f.convertImage(0x01)

```

```

        result = f.searchTemplate()
        positionNumber = result[0]
        accuracyScore = result[1]

        if positionNumber == -1:
            print('No match found!')
            time.sleep(2)
            return
        else:
            print('Found finger at position #' + str(positionNumber))
            print('Accuracy score: ' + str(accuracyScore))
            time.sleep(2)

    except Exception as e:
        print('Operation failed!')
        print('Exception message: ' + str(e))
        exit(1)

# ----- Delete Finger -----
def deleteFinger():
    try:
        positionNumber = int(input('Please enter the template position you want to delete: '))

        if f.deleteTemplate(positionNumber) == True:
            print('Template deleted!')
            time.sleep(1)
            print('Currently used finger templates: ' + str(f.getTemplateCount()) + '/' +
                  str(f.getStorageCapacity()))
            time.sleep(1)

    except Exception as e:
        print('Failed to delete template!')
        print('Exception message: ' + str(e))

# ----- Main Menu -----
def mainMenu():
    while True:
        print("\n===== Fingerprint System Menu =====")
        print("1. Enroll Finger")
        print("2. Search Finger")
        print("3. Delete Finger")
        print("4. Exit")
        choice = input("Enter your choice: ")

        if choice == '1':
            enrollFinger()

```

```
elif choice == '2':
    searchFinger()
elif choice == '3':
    deleteFinger()
elif choice == '4':
    print("Exiting program...")
    break
else:
    print("Invalid choice! Please try again.")

# ----- Run Program -----
if __name__ == "__main__":
    mainMenu()
```