Experiment - 7

(Working with BMP - 280 sensor)

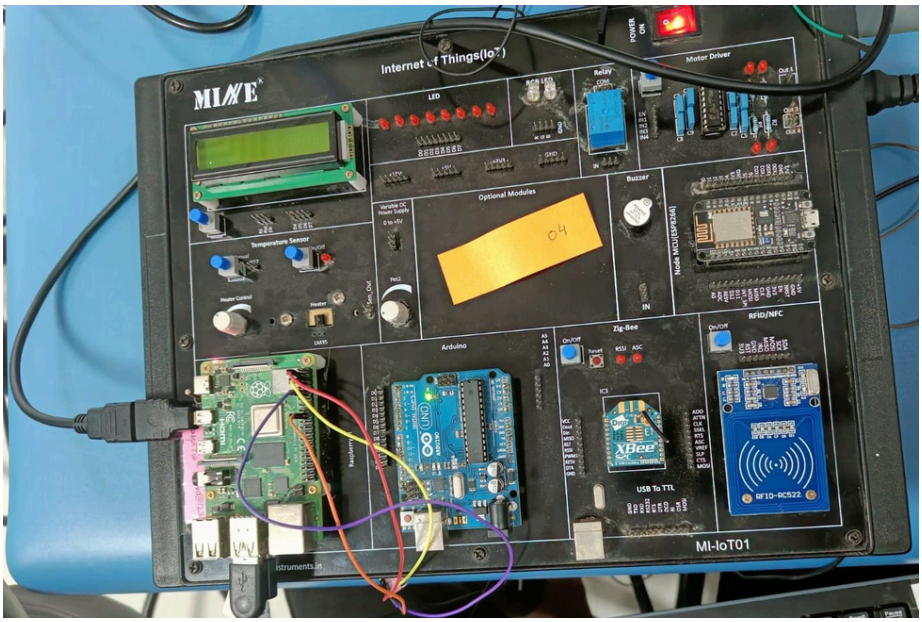
Aim = To set the client server communication with raspberry pi 4 using UDP protocol to transmit BMP280 data from wifi.

Equipments Required:

1. Raspberry Pi-4 Microcontroller Board
2. VGA cable, VGA to Hdmi and Hdmi to micro-usb cables.
3. BMP-280 sensor
4. Jumper wires
5. Monitor
6. Mouse
7. Keyboard

Methodology:

Circuits connections are shown below. Inbelow connection VCC of bmp-280 is connected to 3.3V pin of raspberry pie board and both ground pins are connected together, SCL pin and SDA pins of bmp-280 are connected to GPIO-2 and GPIO-3 pins of raspberry pie 4 board.



Results:- The drive link which contains the photo of results (temperature and pressure) in csv file is :-[Drive link]

Conclusion:- We learnt about the working of BMP-280 and also learnt how to make its connection with raspberry pie, we also understood the code which helps to link BMP-280 and raspberry pie 4 board to obtain data of pressure and temperature.

Code:- The code for server annd client is written in THONNY software as shown below:-

Server code:

```
import socket
import json
import csv
host = '10.205.2.245'
port = 8000

def setupServer():
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    print("Socket created.")
    try:
        s.bind((host, port))
    except socket.error as msg:
        print(msg)
    print("Socket bind comlete.")
    return s

def setupConnection():
    s.listen(1) # Allows one connection at a time.
    conn, address = s.accept()
    print("Connected to: " + address[0] + ":" + str(address[1]))
    return conn


def dataTransfer(conn):
    # A big loop that sends/receives data until told not to.
    while True:
        # Receive the data
        data = conn.recv(1024) # receive the data
        data = data.decode('utf-8')
        print(data)
        # Split the data such that you separate the command
        # from the rest of the data.
```

```
        dataMessage = data.split(' ', 1)
        command = dataMessage[0]
        dataext = json.loads(data)
        fieldname = dataext.keys()
        with open('/home/redpitaya/Documents/meow.csv', mode = 'a', newline='') as file:
            writer = csv.DictWriter(file, fieldnames = fieldname)

            file.seek(0,2)
            if file.tell()==0:
                writer.writeheader()

            writer.writerow(dataext)
            print()
    # conn.close()


s = setupServer()

while True:
    conn = setupConnection()

    while(True):
        dataTransfer(conn)
```

Client code:-

```
#!/usr/bin/env python

import time

from bmp280 import BMP280

try:
    from smbus2 import SMBus
except ImportError:
    from smbus import SMBus
```

```
import socket

host = '10.205.2.245'

port = 8000

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

s.connect((host, port))

print("""temperature-and-pressure.py - Displays the temperature and pressure.

Press Ctrl+C to exit!

""")

# Initialise the BMP280

bus = SMBus(1)

bmp280 = BMP280(i2c_dev=bus)

while True:

    temperature = bmp280.get_temperature()

    pressure = bmp280.get_pressure()

    degree_sign = u"\N{DEGREE SIGN}"

    format_temp = "{:.2f}".format(temperature)

    print('Temperature = ' + format_temp + degree_sign + 'C')

    format_press = "{:.2f}".format(pressure)

    print('Pressure = ' + format_press + ' hPa \n')


        jsonData = "{ \"temperature\":" + "{:.2f}".format(temperature) + ", \"pressure\":" + "
{:.2f}".format(pressure) + "}"


    s.send(str.encode(jsonData))

    time.sleep(4)
```