Wroclaw University of Science and Technology

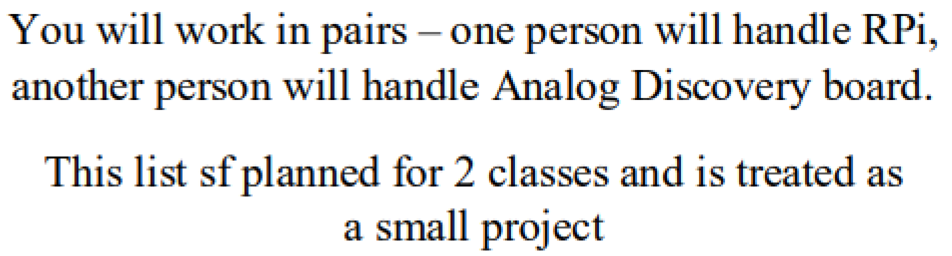
PYTHON
LABORATORY REPORT

Faculty of Electronics, Photonics and MicrosystemsPYTHON LABORATORY

Theme of class: Raspberry Pi, sensors, project

Student: Hayrettin Aycetin (276807)
Date of class: 29.12.2023 15:15-16:55
Group No:3
Submition Date:31.12.2023

Lab assistant: Aleksander Kubeczek, Alicja Kwaśny GRADE:

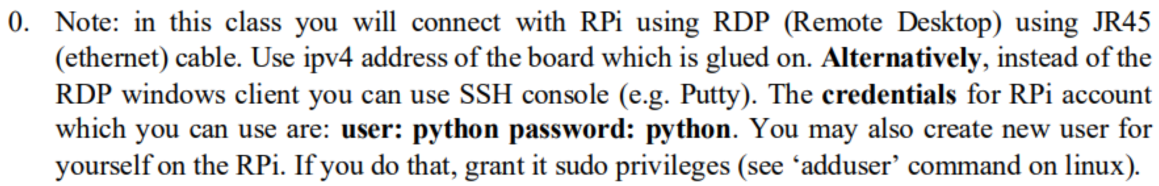


You will work in pairs – one person will handle RPi, another person will handle Analog Discovery board.

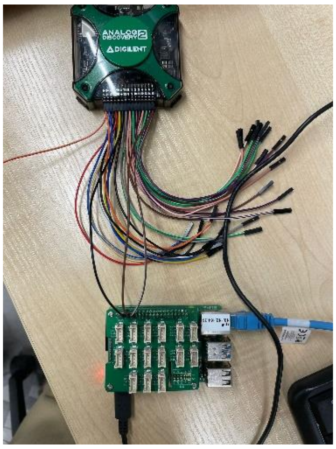
This list sf planned for 2 classes and is treated as a small project

In this project my pair was Yasin Aslan(276719).He had the Rasspery Pi connected to his pc and I had Analog Discovery board connected to mine.

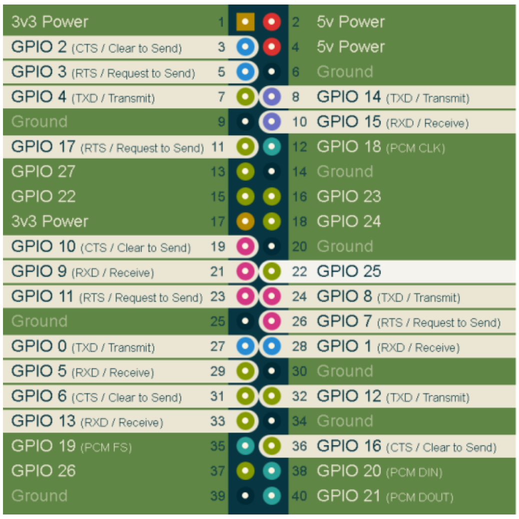
SETUP



0. Note: in this class you will connect with RPi using RDP (Remote Desktop) using JR45 (ethernet) cable. Use ipv4 address of the board which is glued on. **Alternatively**, instead of the RDP windows client you can use SSH console (e.g. Putty). The **credentials** for RPi account which you can use are: **user: python password: python**. You may also create new user for yourself on the RPi. If you do that, grant it sudo privileges (see ‘adduser’ command on linux).

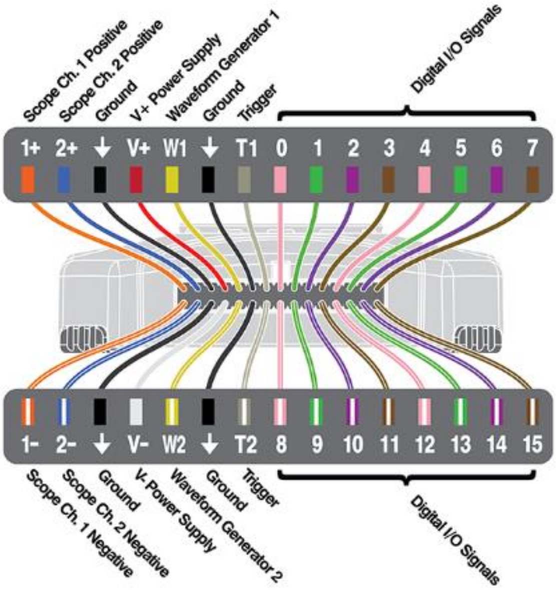


Raspberry Pi Pinout for UART

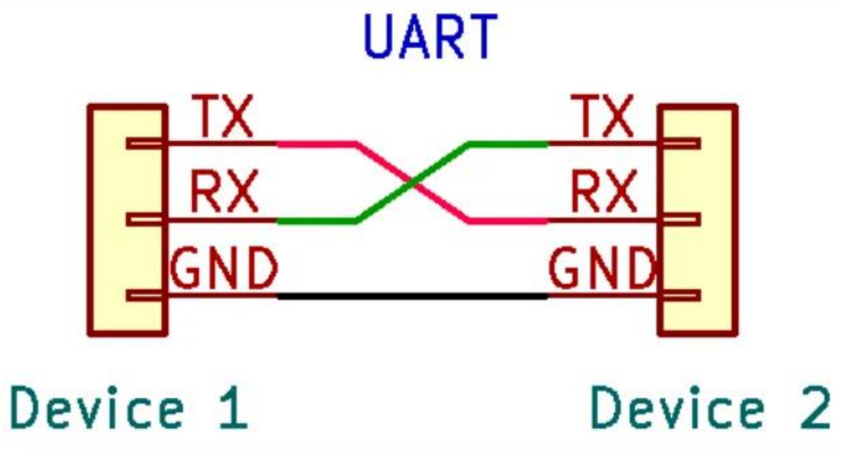


3v3 Power	1	2	5v Power
GPIO 2 (CTS / Clear to Send)	3	4	5v Power
GPIO 3 (RTS / Request to Send)	5	6	Ground
GPIO 4 (TXD / Transmit)	7	8	GPIO 14 (TXD / Transmit)
Ground	9	10	GPIO 15 (RXD / Receive)
GPIO 17 (RTS / Request to Send)	11	12	GPIO 18 (PCM CLK)
GPIO 27	13	14	Ground
GPIO 22	15	16	GPIO 23
3v3 Power	17	18	GPIO 24
GPIO 10 (CTS / Clear to Send)	19	20	Ground
GPIO 9 (RXD / Receive)	21	22	GPIO 25
GPIO 11 (RTS / Request to Send)	23	24	GPIO 8 (TXD / Transmit)
Ground	25	26	GPIO 7 (RTS / Request to Send)
GPIO 0 (TXD / Transmit)	27	28	GPIO 1 (RXD / Receive)
GPIO 5 (RXD / Receive)	29	30	Ground
GPIO 6 (CTS / Clear to Send)	31	32	GPIO 12 (TXD / Transmit)
GPIO 13 (RXD / Receive)	33	34	Ground
GPIO 19 (PCM FS)	35	36	GPIO 16 (CTS / Clear to Send)
GPIO 26	37	38	GPIO 20 (PCM DIN)
Ground	39	40	GPIO 21 (PCM DOUT)

Analog Discovery Pinout



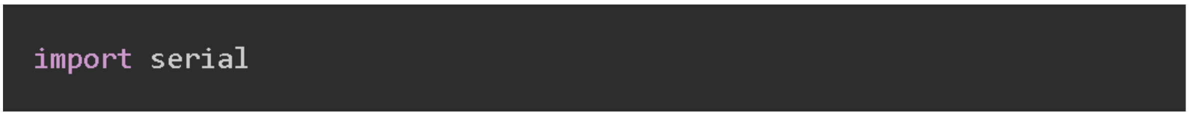
Comments : First we connected Raspberry Pi with using RDP (Remote Desktop) using JR45 (ethernet) cable then we used ipv4 address of the board which is glued on. The main difficulty for us was to determine the correct connections of Analog Discovery and Raspberry Pi. We connected as Input from Analog Discovery pin 7 to pin 8 on Raspberry Pi GPIO 8(TXD/TRANSMIT) ,as output from Analog Discovery we chose pin 15 and connected it to pin 10 on Raspberry Pi pin 10 GPIO15(RXD/Receive).Lastly we connected ground of Analog Discovery to pin 6(Ground) on Raspberry Pi. To understand this stage better we used description below which basically states that we need to connect inverse.



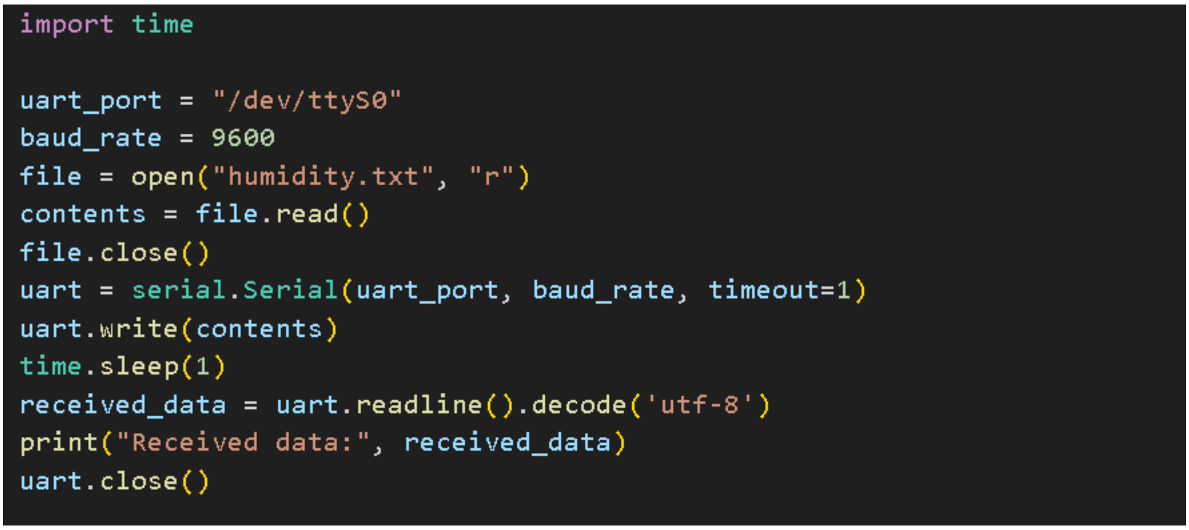
Python

1. Write a python program on RPi which sends and receives something via UART interface (configuration) to and from Analog Discovery board (also UART interface).
   Hints: first you need to enable UART in raspi-config. Then use wiringpi library. Edit the ProtocolUART.py file on your PC to use Analog Discovery board. Remember about connecting grounds of both boards.

Serial



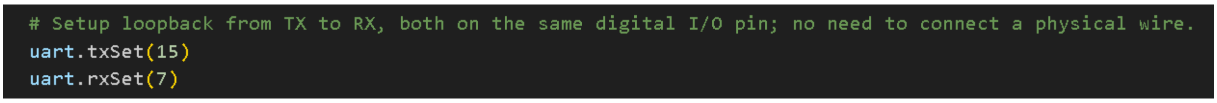
```
import serial
```



```
import time

uart_port = "/dev/ttyS0"
baud_rate = 9600
file = open("humidity.txt", "r")
contents = file.read()
file.close()
uart = serial.Serial(uart_port, baud_rate, timeout=1)
uart.write(contents)
time.sleep(1)
received_data = uart.readline().decode('utf-8')
print("Received data:", received_data)
uart.close()
```

Comments: We couldn’t use the code above because we lost time on understanding the setup and instead of serial we tried to use wiringpi library but it didn’t work properly.The code above first opens a file called "humidity.txt" and reads the contents into a variable. Then, it opens the serial port at 9600 baud rate and sends the contents of the file to the device .It waits for one second and then reads one line of data from the device and prints it to the console. Finally, it closes the serial port.Basically this code is used to communicate with Analog Discovery connected to a serial port.

ProtocolUart

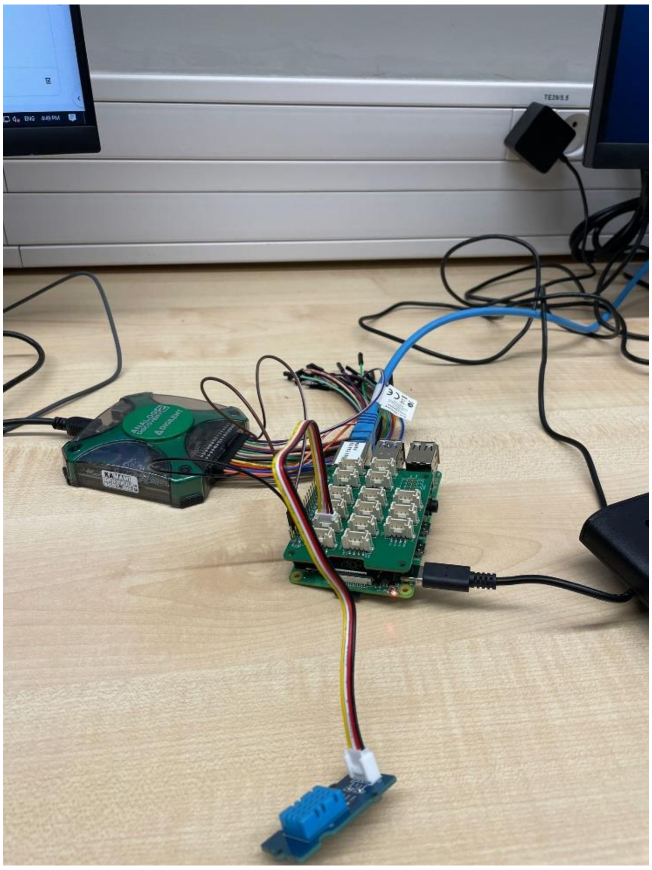


```
# Setup loopback from TX to RX, both on the same digital I/O pin; no need to connect a physical wire.
uart.txSet(15)
uart.rxSet(7)
```

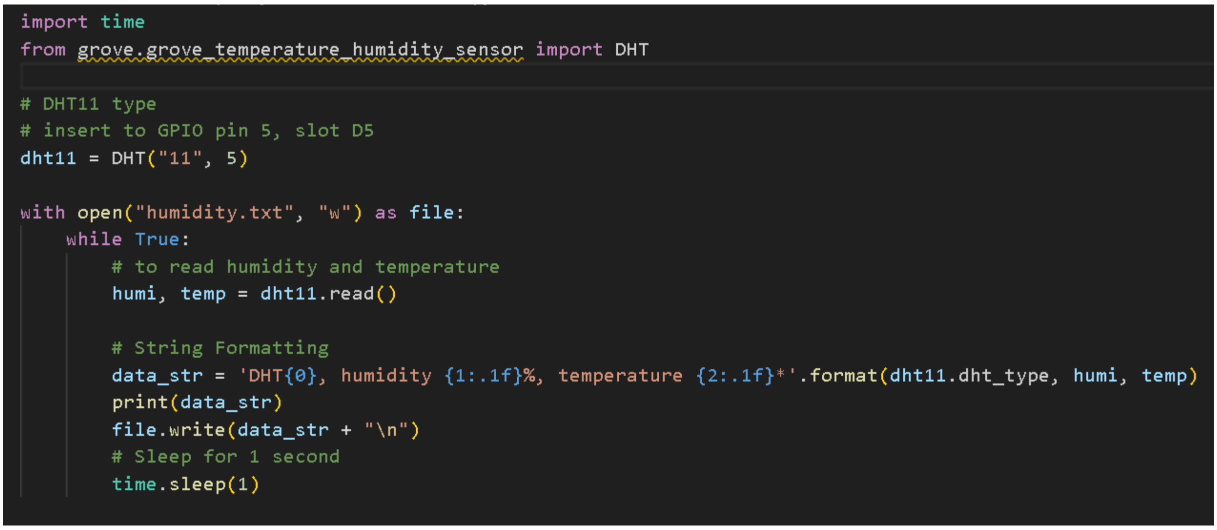
Comments: This whole code was provided before so basically we set our transmit to output pin of Analog Discovery which is 15 and receive to input pin of Analog Discovery but of course we connected with Raspberry Pi opposite way.

2. Choose one sensor, either analog or digital. Use it with RPi.

Comments : We have chosen temperate and humidity sensor and connected as below to our Raspberry Pi slot D5 because it was told like that in the code which we took from github.

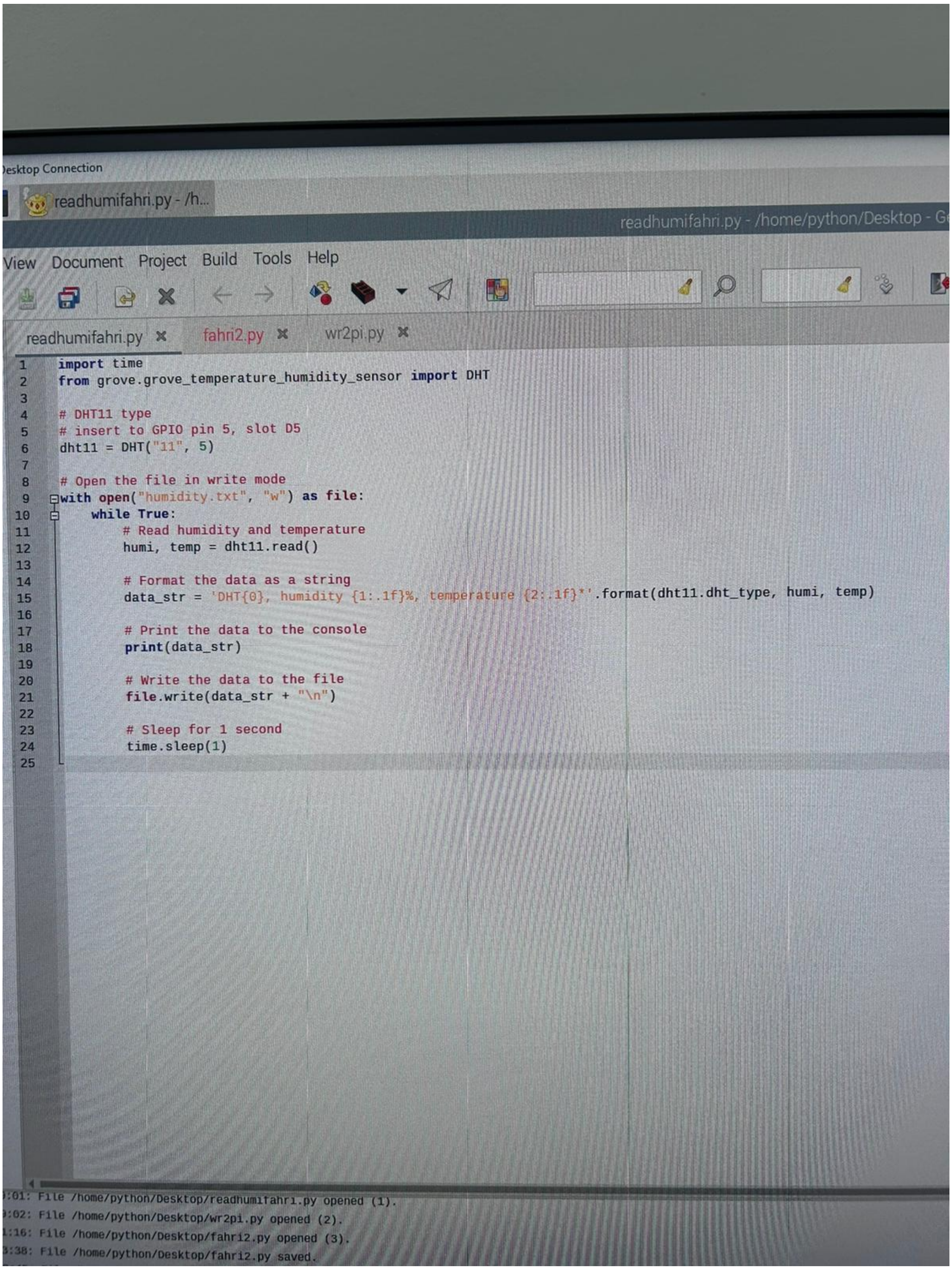


3. Read data from chosen sensor and store it is a file (type and extension of your choice).



```python
import time
from grove.grove_temperature_humidity_sensor import DHT

# DHT11 type
# insert to GPIO pin 5, slot D5
dht11 = DHT("11", 5)

with open("humidity.txt", "w") as file:
    while True:
        # to read humidity and temperature
        humi, temp = dht11.read()

        # String Formatting
        data_str = 'DHT{0}, humidity {1:.1f}%, temperature {2:.1f}*'.format(dht11.dht_type, humi, temp)
        print(data_str)
        file.write(data_str + "\n")
        # Sleep for 1 second
        time.sleep(1)
```

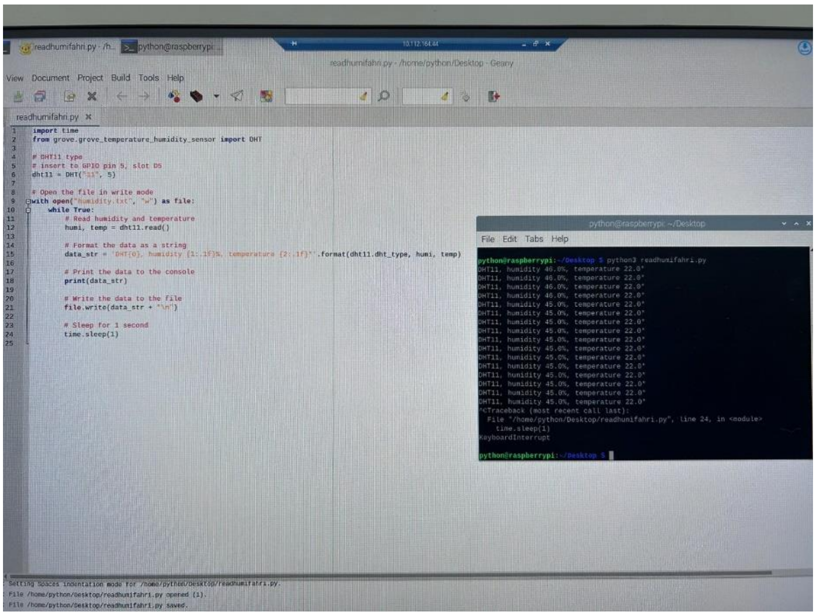
Above I have written the code in a more clear way and readable in Visual Studio Code but I didn’t import humidity sensor library to my computer with pip install so that’s why I have that error there.


esktop Connection
readhumifahri.py - /h...
readhumifahri.py - /home/python/Desktop - G
View Document Project Build Tools Help
readhumifahri.py
fahri2.py
wr2pi.py

```
import time
from grove.grove_temperature_humidity_sensor import DHT

# DHT11 type
# insert to GPIO pin 5, slot D5
dht11 = DHT("11", 5)

# Open the file in write mode
with open("humidity.txt", "w") as file:
    while True:
        # Read humidity and temperature
        humi, temp = dht11.read()

        # Format the data as a string
        data_str = 'DHT{0}, humidity {1:.1f}%, temperature {2:.1f}*'.format(dht11.dht_type, humi, temp)

        # Print the data to the console
        print(data_str)

        # Write the data to the file
        file.write(data_str + "\n")

        # Sleep for 1 second
        time.sleep(1)

```

```
:01: File /home/python/Desktop/readhumifahri.py opened (1).
:02: File /home/python/Desktop/wr2pi.py opened (2).
:16: File /home/python/Desktop/fahri2.py opened (3).
:38: File /home/python/Desktop/fahri2.py saved.
```

Here is the code that we wrote in Raspberry Pi.



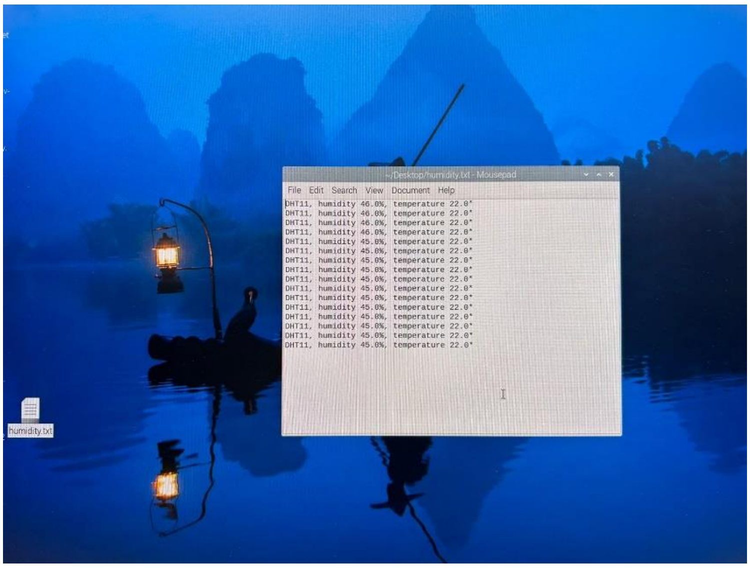
```python
import time
from grove.grove_temperature_humidity_sensor import DHT

# DHT11 type
# insert to GPIO pin 5, slot D5
dht11 = DHT("11", 5)

# Open the file in write mode
with open("humidity.txt", "w") as file:
    while True:
        # Read humidity and temperature
        humi, temp = dht11.read()

        # Format the data as a string
        data_str = 'DHT{0}, humidity {1:.1f}%, temperature {2:.1f}*'.format(dht11.dht_type, humi, temp)

        # Print the data to the console
        print(data_str)

        # Write the data to the file
        file.write(data_str + "\n")

        # Sleep for 1 second
        time.sleep(1)
```

```
python@raspberrypi:~/Desktop $ python3 readhumifahri.py
DHT11, humidity 46.0%, temperature 22.0*
DHT11, humidity 46.0%, temperature 22.0*
DHT11, humidity 46.0%, temperature 22.0*
DHT11, humidity 46.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
^CTraceback (most recent call last):
  File "/home/python/Desktop/readhumifahri.py", line 24, in <module>
    time.sleep(1)
KeyboardInterrupt

python@raspberrypi:~/Desktop $
```

```
Setting Spaces indentation mode for /home/python/Desktop/readhumifahri.py.
File /home/python/Desktop/readhumifahri.py opened (1).
File /home/python/Desktop/readhumifahri.py saved.
```

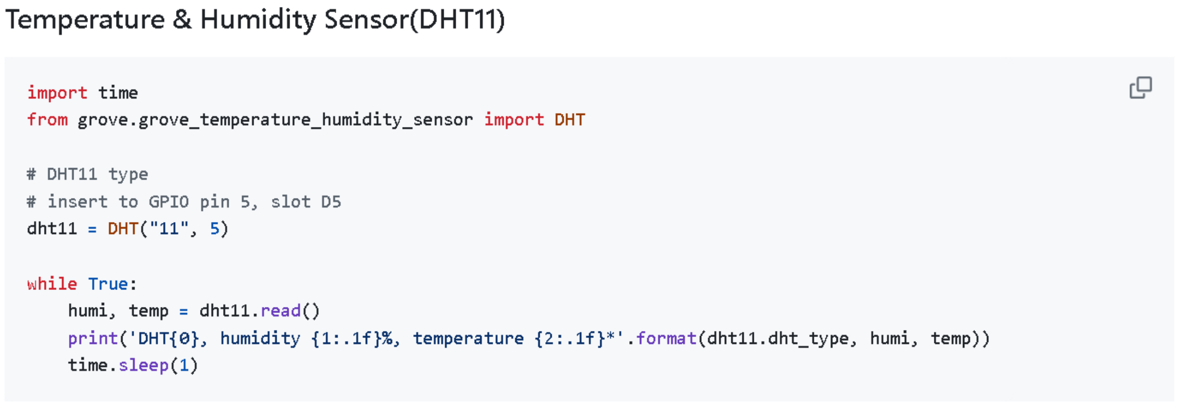
Here we are running the code and taking the measurements from our sensor.

Here is the text file that we store our measurements.



```
DHT11, humidity 46.0%, temperature 22.0*
DHT11, humidity 46.0%, temperature 22.0*
DHT11, humidity 46.0%, temperature 22.0*
DHT11, humidity 46.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
DHT11, humidity 45.0%, temperature 22.0*
```

Comments : In this part we had base of our script from GitHub(which is below) but we changed it a bit. Firstly we have opened a file called “humidity.txt” in write mode ,formatted our data and assigned it to data_str. Then using .write notation we wrote our data to our file.



Temperature & Humidity Sensor(DHT11)

```
import time
from grove.grove_temperature_humidity_sensor import DHT

# DHT11 type
# insert to GPIO pin 5, slot D5
dht11 = DHT("11", 5)

while True:
    humi, temp = dht11.read()
    print('DHT{0}, humidity {1:.1f}%, temperature {2:.1f}*'.format(dht11.dht_type, humi, temp))
    time.sleep(1)
```

Above is the code from Github to use our sensor.

4. Send the file to your partner's device (Analog Discovery) and display it using simple GUI. The basic setup should look more-less like this:

   digital/analog sensor → RPi app → via UART → Analog Discovery → Windows app with GUI

Comments : Unfortunately as I mentioned before we couldn't send our file from Raspberry Pi to our Analog Discovery Device but If we had more time I think we could do it with the codes that we wrote here.

CONCLUSION

In this lab we have learned how two devices are sending each other files via UART method. We also understood how Raspberry Pi works with sensors and how it interacts with different devices.