Co-funded by the
Erasmus+ Programme
of the European Union

Poitiers

Lycee Pilote
Innovant International

P H Y S I C S A N D
E N V I R O N M E N T

Topic Environment and society Measurment of light intensity.	Age >14	Country Poland	Date v.1 March 2020 v.2 October 2021

**The original version of the experiment (v.1) was developed using the Arduino IDE. With the advent of the Raspberry Pi Pico microcontroller and its great possibilities, the description of the experiment has been adapted to MicroPython (v.2), which will make it much easier for students to perform it.*

Function, realisation

Construction of a measuring instrument based on microcontroller and Python that measures the light intensity.

Hardware required

- Microcontroller Raspberry Pi Pico with Micropython 1.18
- Cytron Maker Pico docking station for Pico (or different docking station for Pico: Waveshare, Seeedstudio)
- BH1750 module (3.3V for Pico)
- 3V3 LCD 1602 I2C display from Seeedstudio with Grove Socket,
- PC or Mac computer.

Materials required

- 1 standard Grove wire, 1xGrove female wire
- micro usb 2.0 high speed cable(15cm or 30 cm)
- powerbank (for standalone version)
- paper for note

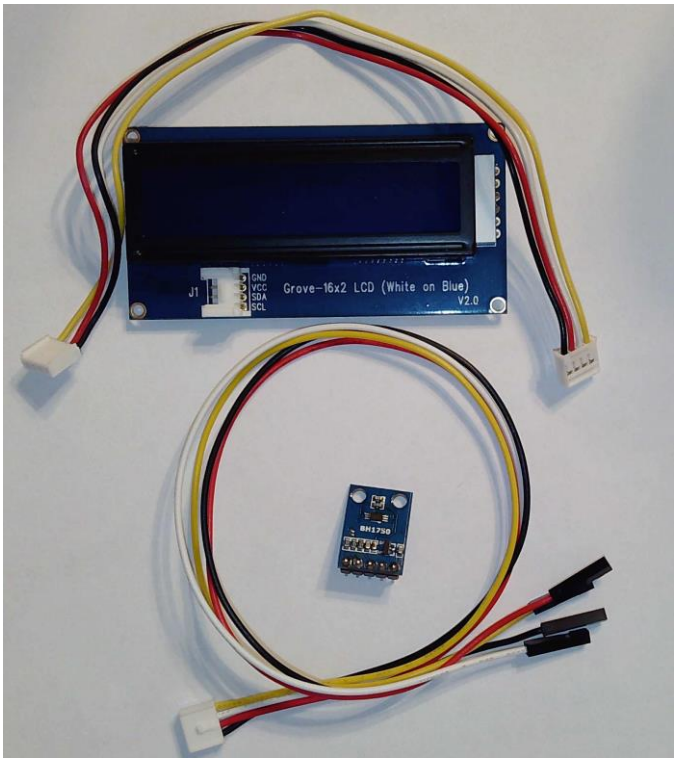
Software required

- Thonny App (thonny.org)
- library for BH1750 sensor: https://github.com/PinkInk/upylib/tree/master/bh1750/bh1750
- library for LCD: https://files.seeedstudio.com/wiki/Grove_Shield_for_Pi_Pico_V1.0/Libraries.rar

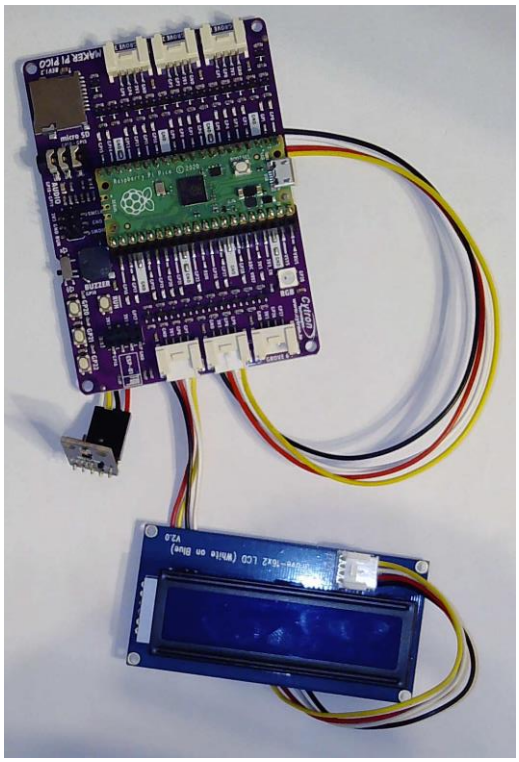
Using Thonny App You can save libraries inside lib folder in Pico memory.

Setup Hardware

BH1750 module is a digital light sensor which uses the I2C interface. This allows it to be connected to the Raspberry Pi with only four wires. BH1750 is a Digital Ambient light sensor. It is easy to interface with a microcontroller, as it uses the I2C communication protocol. It consumes a



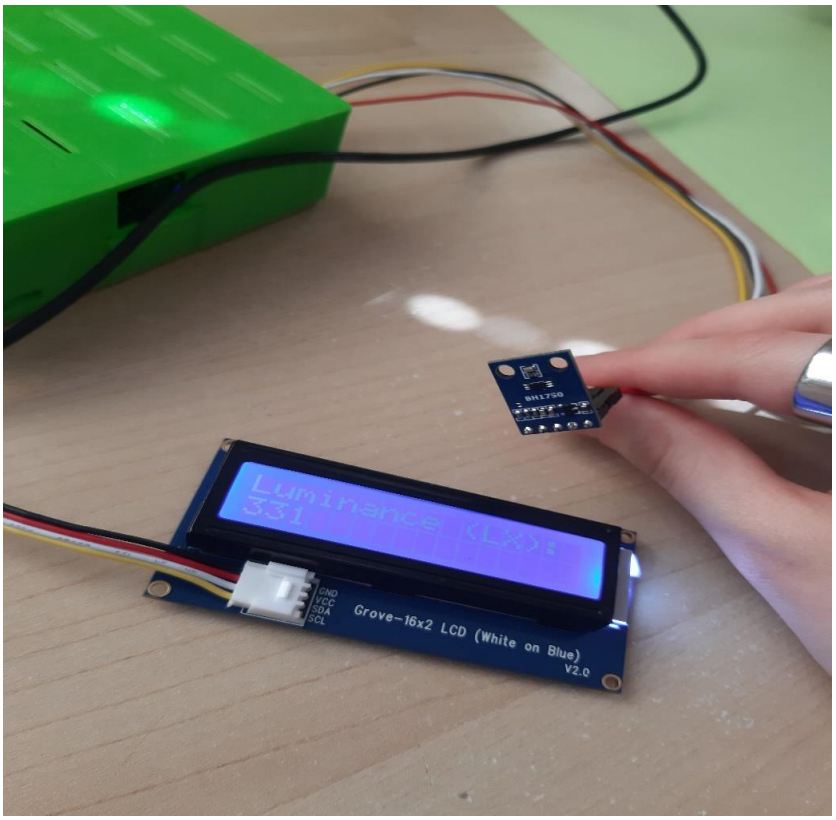
very low amount of current.



BH1750 sensor pins	Raspberry Pi Pico pins	Grove female wire
VCC	3V3	red
GND	GND	black
SCL	GP9	yellow
SDA	GP8	white

The Pico use 3.3V power (not 5!!!). We usually connect the red wire to pin 3V3 and black wire to the ground (GND). SDA and SCL are the pins used for I2C communication.
We suggest using the LCD screen to display the results. In this case, we can also build an autonomous, mobile system that can work without a computer. The use of grove cables makes the connection as easy as possible..

LCD 1602 pins	Raspberry Pi Pico pins	Grove standard wire
GND	GND	black
VCC	3V3	red
SDA	GP6	white
SCL	GP7	yellow



Raspberry Pi Pico in our 3D printed housing with Grove LCD and BH1750 Light sensor

Setup software

Below Python code for saving in microcontroller using Thonny App. Write the script in Thonny, then select "File/Save" and select Raspberry Pi Pico as destination. Remember please to add .py at the end of filename. Our proposal for filename: **bh1750lcd16x2.py**

```
from machine import I2C,Pin
from bh1750 import BH1750
from utime import sleep
from lcd1602 import LCD1602
#sensor BH1750 connected to GP8,GP9
i2c = I2C(0,scl=Pin(9), sda=Pin(8), freq=400000)
#i2c = machine.I2C(scl,sda)
s = BH1750(i2c)
#LCD16x2 display I2C: GP6,GP7
i2cbis = I2C(1,scl=Pin(7), sda=Pin(6), freq=400000)
#Init for LCD display
d = LCD1602(i2cbis, 2, 16)
d.display()
d.clear()
#set the place for print and printing on LCD
d.setCursor(0,0)
d.print('Measurment')
d.setCursor(0,1)
d.print('Light')
sleep(2)
d.clear()
while True:
    d.setCursor(0,0)
    d.print('Luminance (LX):')
    d.setCursor(0,1)
    d.print(str(int(s.luminance(BH1750.ONCE_HIRES_1))))
    sleep(2)
    d.clear()
```

Below You can find simple version of the code, which can be used with Thonny Plotter.Will be good to start with this simple version to check if everything works well.

```
from machine import I2C,Pin
from bh1750 import BH1750

#sensor BH1750 connected to GP8,GP9
i2c = I2C(0,scl=Pin(9), sda=Pin(8), freq=400000)

s = BH1750(i2c)

while True:
    print(int(s.luminance(BH1750.ONCE_HIRES_1)))

```

If you want have autonomous system then save the script on the Pico as with special name: **main.py.** This script will start automatically (No need to connect to a computer). In this case you can power the Pico from Powerbank.

The principle of operation of Digital Light Sensor

BH1750 Sensor board. When an ambient light is incident on the sensor the photo diode with a latency nearly equals to human eye absorbs the

light and produces an analog voltage signal which is further connected to Analog to Digital converter (ADC unit). Result signal is sent to Pico through I2C bus.

Technical specifications of BH1750:

- Operating voltage: 3V to 5V
- Measures Lux range with a high resolution from 1 to 65535lx.
- Illuminance to Digital Converter
- The influence of infrared is very small
- Small measurement variation (+/- 20%)
- Feature to select 2 different I2C addresses.
- Low power consumption

Definition of Luminous ,Lumen

Illuminance is a measure of how much luminous flux is spread over a given area. One can think of luminous flux (measured in lumens) as a measure of the total "amount" of visible light present, and the illuminance as a measure of the intensity of illumination on a surface

Lumen : The unit for the quantity of light flowing from a source in any one second (the luminous power, or luminous flux) is called the lumen. In our sensor we will take a reading from it in Lux which is equal to one lumen per square metre:
Lux = 1 Lm/m2 .

Experiment Evaluation

System testing:
-measure the light intensity in various situations: sunlight, torch, normal lightning, sensor being covered with something,
-find dependency between luminance and distance from light source,
-compare the obtained results with the results of the mobile application,
-change the delay (sleep) and observe the effect on the results,
-change the Python script to plot graph based on results (using internal ploter from Thonny App),
-you can use the built-in system to check if the lighting in the room is good enough according to the standards,
-you can check if the lighting in the desk you are working on is sufficient.

Write down your results:

situation	result (lux)	situation	result (lux)

This sensor is mainly used in LCD displays, TVs, Monitors and mainly in Mobile displays to automatically adjust the screen brightness according to the outdoor light conditions. This sensor can also used to Turn on/Off the lights.

Topic: Environment and society Measurement of humidity and temperature with DHT11	Age >14	Country Poland	Date v1 March 2020 v.2 October 2021

**The original version of the experiment (v.1) was developed using the Arduino IDE. With the advent of the Raspberry Pi Pico microcontroller and its great possibilities, the description of the experiment has been adapted to MicroPython (v.2), which will make it much easier for students to perform it.*

Function, realisation

Build experimental system based on DHT11 sensor, which can measures air parameters: temperature and humidity.
The temperature and humidity DHT11 sensor is the basic device for measuring temperature and humidity. This sensor is used in many devices to measure their temperature. It is widely known and used all over the world.

Hardware required

- Microcontroller Raspberry Pi Pico with Micropython 1.18 or above,
- Cytron Maker Pico docking station for Pico
- DHT11 sensor **with module** or DHT11 sensor from Seeedstudio with Grove socket. (Notice: on the market you can buy also sensor without module, but in this case you need resistor, breadboard and connection to Pico is not so easy.),
- PC or Mac computer,
- 3V3 LCD 1602 I2C display Seeedstudio with Grove socket.

Materials required

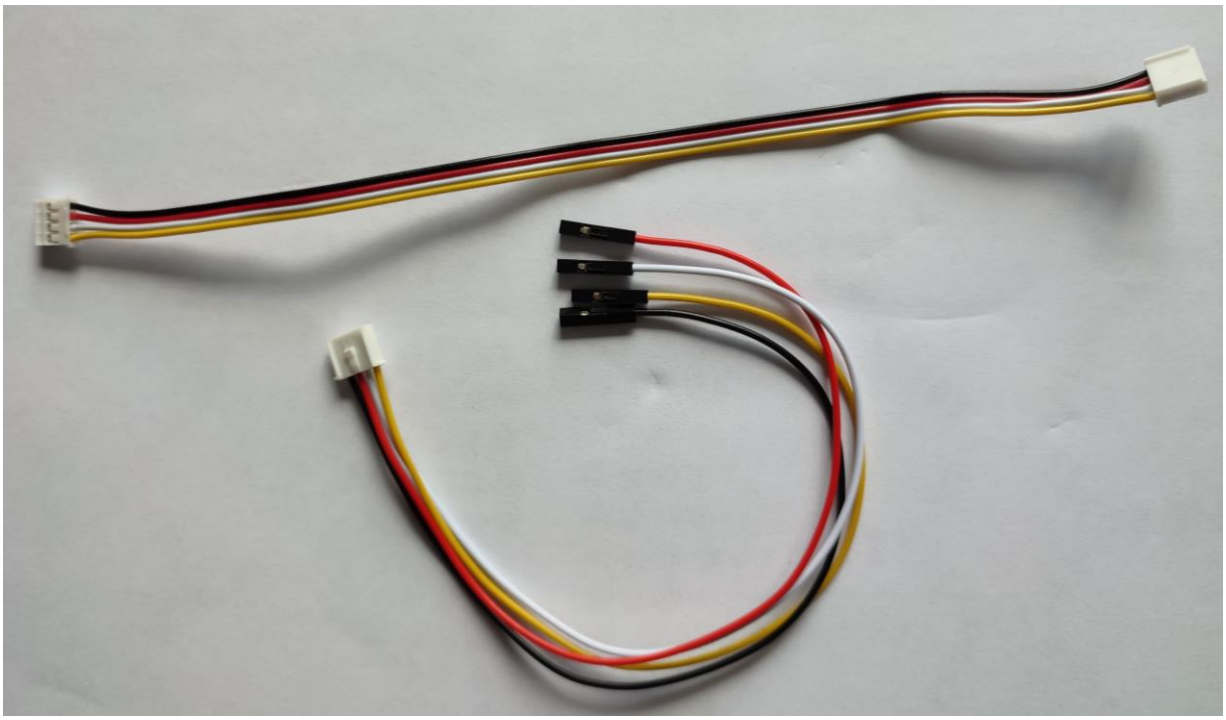
2xGrove standard wire (or 1 Grove standard Wire and 1 Grove-Female wire, depend on sensor selection),
micro usb 2.0 high speed (15cm or 30 cm)
paper for note

Software required:

-Thonny App (thonny.org)
-library for DHT11 sensor is a standard Micropython library

- library for LCD: https://files.seeedstudio.com/wiki/Grove_Shield_for_Pi_Pico_V1.0/Libraries.rar

DHT11 sensor with Grove socket (on the left) and cheaper DHT11 with module on the right.



On above illustration on the top standard Grove Wire, on the bottom female Grove wire for sensor connection.

Setup Hardware

Connections have been simplified thanks to the docking station and Grove cables. It is much harder to make a mistake or damage.

DHT 11 Sensor connection:

DHT 11 pins	Raspberry Pi Pico pins	Grove wire
Signal	GP9	yellow
not connected	GP8	white
power (+)	3V3	red
ground (-)	GND	black

The pins can be changed according to your needs, but in this case you must change the code.
All functions of Raspberry Pi Pins, you can find on the website: pico.pinout.xyz.

LCD I2C screen connection:

LCD Pins	Raspberry Pi Pico pins	Grove wire
SCL	**GP7**	**yellow**
SDA	**GP6**	**white**
VCC	**3V3**	**red**
GND	**GND**	**black**

The Pico uses a 3.3V power supply (not 5V !!!). Usually we connect the red wire to the 3V3 pin and the black wire to the ground (GND).

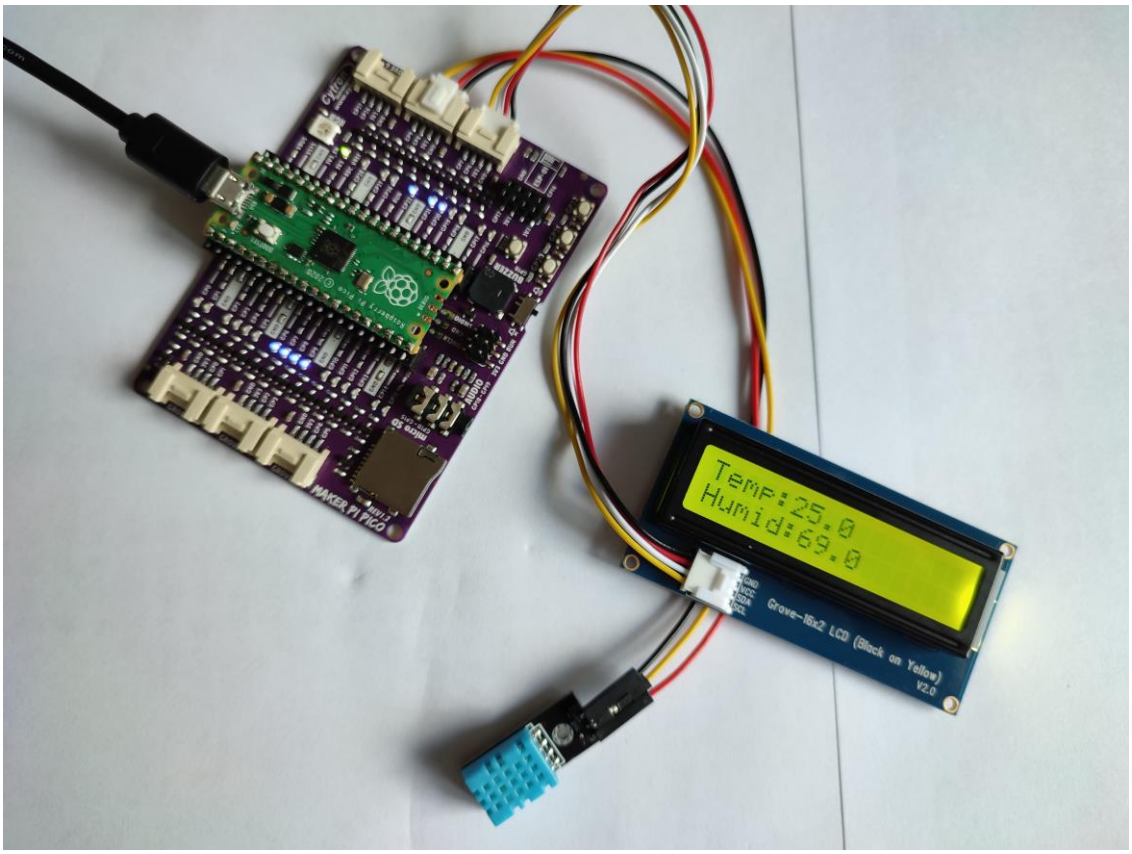
Setup software

The script use special libraries, but only standard Micropython libraries, we are not allowed to install libraries.
The script is very simple, very easy to understand, as are the scripts that read data from analog sensors.

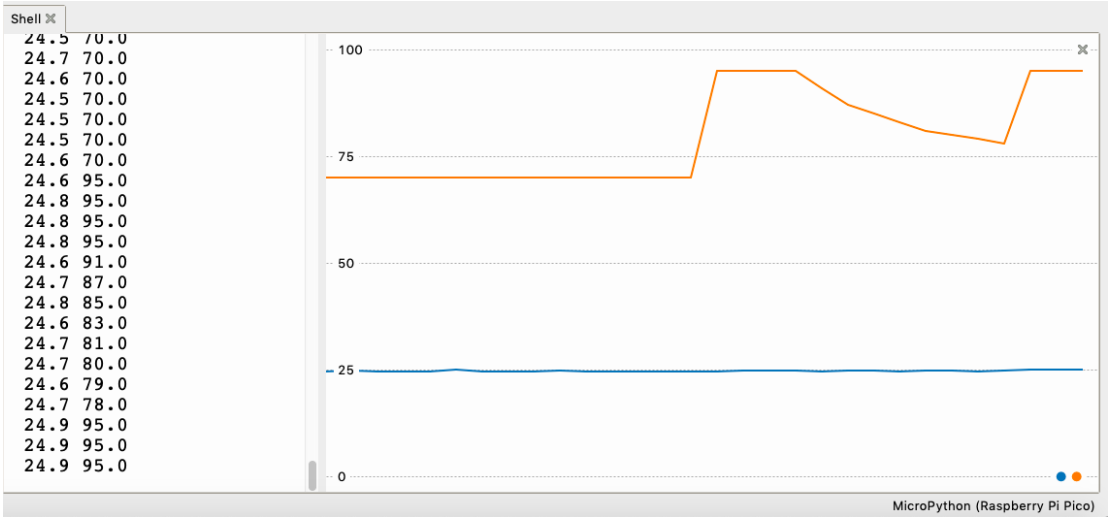
```
from lcd1602 import LCD1602
from dht11 import *
from machine import I2C,Pin
from utime import sleep
# 1 LCD 16x2 display initialisation
i2c = I2C(1,scl=Pin(7), sda=Pin(6), freq=400000)
d = LCD1602(i2c, 2, 16)
d.display()
# 1 LCD display
# sensor connected to PIN 9
dht = DHT(9)

while True:
    #reading from the sensor
    temp,humid = dht.readTempHumid()#temp:  humid:
    sleep(1)
    d.clear()
    # set the place on LCD for printing results
    d.setCursor(0,0)
    d.print("Temp:"+str(temp))
    #the line below allows us to observe the results in the console and plot with Plotter
    print(temp, humid)
    d.setCursor(0,1)
    d.print("Humid:"+str(humid))
    sleep(1)
```

This script allows you to plot the measurement results with a plotter. To do this, select "View / Plotter" from the Thonny menu.

Demonstration of the system operation:



Terminal window and Plotter window with results of reading:



Go Go further- ideas, testing:

-test the system by finger touching

-place a heat source nearby.

The sensor is not waterproof !!!

-You can use this simple example to build weather station

-You can build the system with logger and use outside

-design a system that sends the sensor result to a website or via bluetooth to a mobile application on your phone.

Topic: Environment and society Measurement of pressure	Age >14	Country Poland	Date v1 March 2020 v.2 October 2021

**The original version of the experiment (v.1) was developed using the Arduino IDE. With the advent of the Raspberry Pi Pico microcontroller and its great possibilities, the description of the experiment has been adapted to MicroPython (v.2), which will make it much easier for students to perform it.*

Function, realisation

Construction of a measuring instrument based on microcontroller and Python that measures pressure (also temperature and humidity).

Hardware required

- Microcontroller Raspberry Pi Pico with Micropython 1.18
- Cytron Maker Pico docking station for Pico
- BME280 module (3.3V for Pico)
- 3V3 LCD 1602 I2C display from Seeedstudio with Grove Socket,
- PC or Mac computer.

Materials required

- 1 standard Grove wire, 1xGrove female wire
- micro usb 2.0 high speed cable(15cm or 30 cm)
- powerbank (for standalone version)
- paper for note

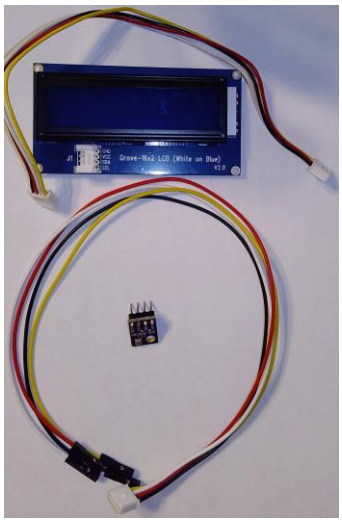
Software required

- ThonnyApp (thonny.org)
- library for the BME280 sensor: https://github.com/SebastianRoll/mpy_bme280_esp8266/blob/master/bme280.py, display of units has been removed from the library
- library for LCD: https://files.seeedstudio.com/wiki/Grove_Shield_for_Pi_Pico_V1.0/Libraries.rar

Use Thonny app to save libraries inside lib folder in Pico memory.

Technical specifications of BME280:

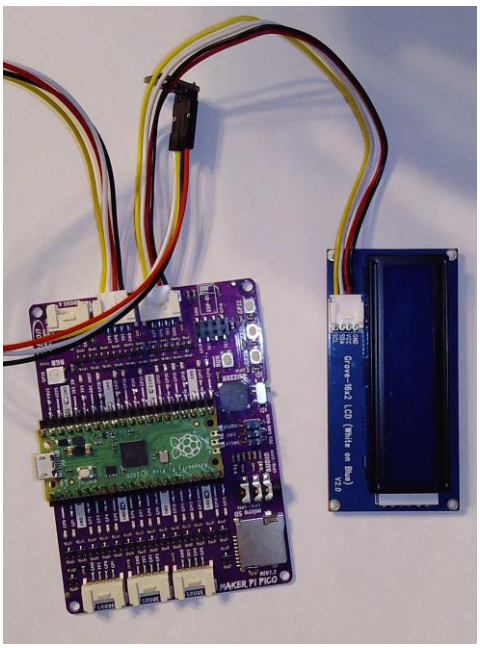
- Supply voltage: 3.3 V



- Temperature measurement range: -40 °C to 85 °C
- Accuracy: ± 1 °C
- Humidity measuring range: 10 to 100% RH
- Accuracy: ± 3% RH
- Pressure measuring range: 300 to 1100 hPa
- Accuracy: ± 1 hPa

Setup Hardware:

Humidity, temperature and pressure sensor powered with the voltage of 3.3 V with I2C It is easy to interface with a microcontroller. It consumes a very low amount of current.



BME280 sensor pins	Raspberry Pi Pico pins	Grove female wire
VCC	3V3	red
GND	GND	black
SCL	GP7	yellow
SDA	GP6	white

The Pico use 3.3V power (not 5!!!). We usually connect the red wire to pin 3V3 and black wire to the ground (GND). SDA and SCL are the pins used for I2C communication.
We suggest using the LCD screen to display the results. In this case, we can also build an autonomous, mobile system that can work without a computer. The use of grove cables makes the connection as easy as possible.

LCD 1602 pins	Raspberry Pi Pico pins	Grove standard wire
GND	GND	black
VCC	3V3	red
SDA	GP6	white
SCL	GP7	yellow

Raspberry Pi Pico in our 3D printed housing with Grove LCD and BME280 pressure sensor

Setup software

Below Python code for saving in microcontroller using Thonny App. Write the script in Thonny, then select "File/Save" and select Raspberry Pi Pico as destination. Remember please to add .py at the end of filename. Our proposal for filename: **bme280lcd16x2.py**

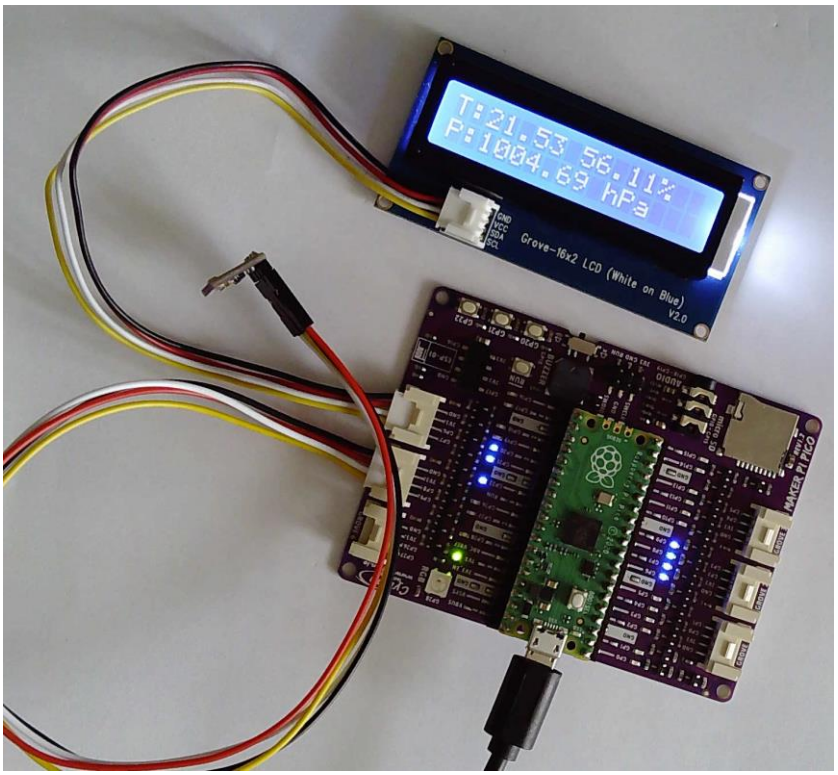
```
from machine import Pin, I2C
from bme280 import BME280
import utime
from lcd1602 import LCD1602
#I2C Bus for bme280
i2c = I2C(0, scl=Pin(9), sda=Pin(8), freq=400000)
#sensor initialisation
bme = BME280(i2c=i2c, address=0x76)
#I2C Bus for LCD
i2cbis = I2C(1, scl=Pin(7), sda=Pin(6))
#LCD display initialisation
d = LCD1602(i2cbis, 2, 16)
d.display()

print("temp. C, pressure hPa,humidity %")
while True:
    # clear LCD from previous results
    d.clear()
    #set the place for print and printing on LCD
    d.setCursor(0,0)
    d.print("T:"+str(bme.values[0]))
    d.setCursor(0,1)
    d.print("P:"+str(bme.values[1]) +" hPa")
    d.setCursor(8,0)
    d.print(str(bme.values[2]) +"%")
    print(bme.values[0],bme.values[1],bme.values[2])
    utime.sleep(5)
```

Below You can find simple version of the code, which can be used with Thonny Plotter.Will be good to start with this simple version to check if everything works well.

```
from machine import Pin, I2C
from bme280 import BME280
import utime
#I2C Bus initialisation
i2c = machine.I2C(0, scl=Pin(9), sda=Pin(8))
#sensor initialisation
bme = BME280(i2c=i2c, address=0x76)
print("temp. C, pressure hPa,humidity %")
while True:
    print(bme.values[0],bme.values[1],bme.values[2])
    utime.sleep(2)
```

If you want have autonomous system then save the script on the Pico with special name: **main.py.** This script will start automatically (No need to

connect to a computer). In this case you can



power the Pico from powerbank.

Experiment Evaluation

System testing:

- measure pressure, humidity and temperature,
- test the system in vacuum,
- test the system in a mine or in a cave during the tour
- observe pressure and temperature changes as you hike to the mountains
- use a balloon to observe changes in temperature, pressure and humidity
  record changes in air parameters throughout the day (find our "data logger" instructions)
- change the scripts, to measure and display/plot only one parameter from the list: temperature, humidity, pressure

Write down your results:

situation	pressure	temperature	humidity

This sensor can be used in many situations. May be also the base for built weather station.

Topic: Environment and society Measurement of soil moisture	Age >12	Country Poland	Date v1 March 2020 v.2 October 2021

**The original version of the experiment (v.1) was developed using the Arduino IDE. With the advent of the Raspberry Pi Pico microcontroller and its great possibilities, the description of the experiment has been adapted to MicroPython (v.2), which will make it much easier for students to perform it.*

Function, realisation

Build experimental system which measures humidity of soil and other things.

This project is of great importance due to the lack of water and the need for its rational use. The wide application of similar solutions in agriculture can reduce losses in plant production and lower costs.

Hardware required

- Microcontroller Raspberry Pi Pico with Micropython 1.18 or above
- Cytron Maker Pico docking station for Pico
- Grove moisture sensor v1.4
- PC or Mac computer
-

Materials required

- 1 long grove wire
- micro usb 2.0 high speed (15cm or 30 cm)
- paper for note
- plant in a pot

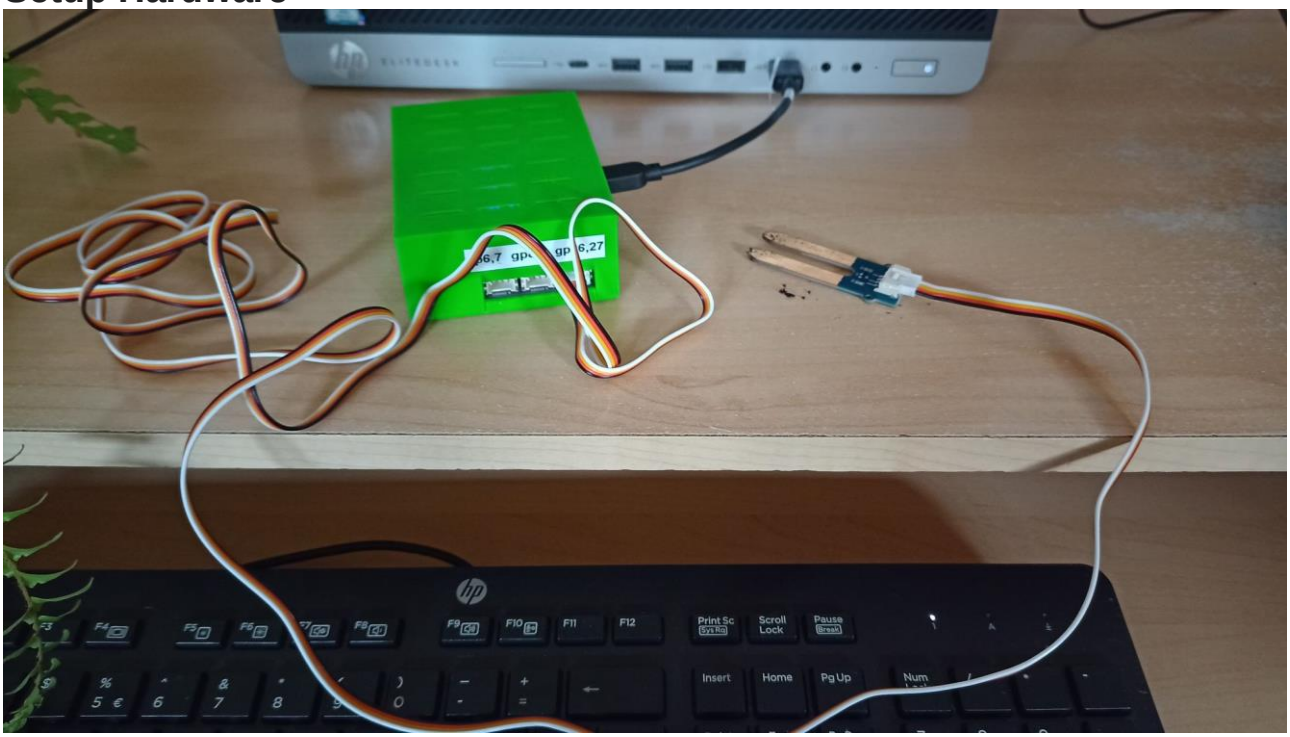
- water

Software required

- Thonny App (thonny.org)

- custom library not required

Setup Hardware



Sensor pin	Pico Pin	Grove wire
VCC	3V3	red
GND	GND	black
SIG	GP27	yellow
not connected	GP26	white

The reader use 3V3 power (not 5V!!!). Usually we connect red wire to pin 3V3 and black wire to the ground (GND)

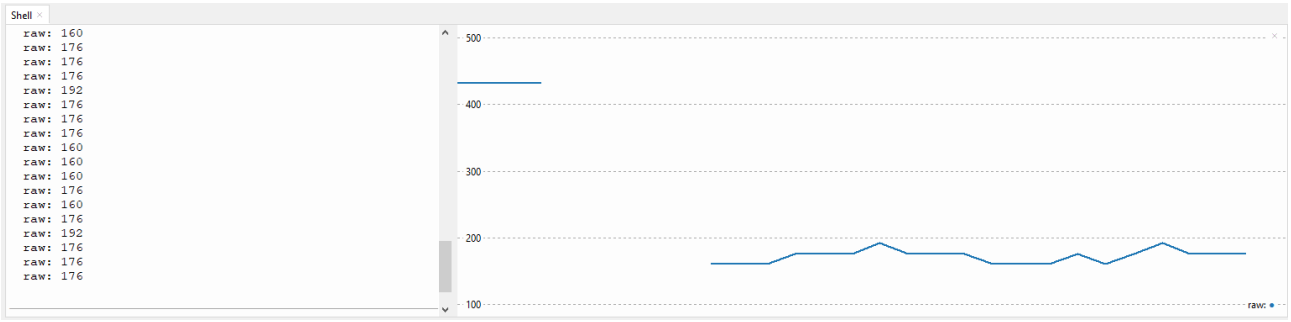
Setup software

The file analogwatersensor.py which contain script for soil testing we must run on our Pico microcontroller. We can do this with Thonny software.

The script for testing

```
import utime
import machine
#Grove Water sensor connected to Cytron Maker ADC1
analog_value = machine.ADC(1)

while True:
    sensor_raw = analog_value.read_u16()
    print("raw:", sensor_raw)
    utime.sleep(2)
```

Terminal window with results of reading



Possible results are in the scope: 0-65535

Description of the sensor

Grove - Soil Moisture Sensor can measure soil moisture for plants. The soil moisture sensor consists of two probes that allow the current to pass through the soil and then obtain resistance values to measure soil moisture content. It can be used to decide if the plants in a garden need watering. You can also use soil moisture sensors in gardens to automate watering plants. It can be used very easily by just inserting the sensor into the soil and reading the output using ADC. You can define the level at which the alarm should be sent or the watering can be started by a relay.

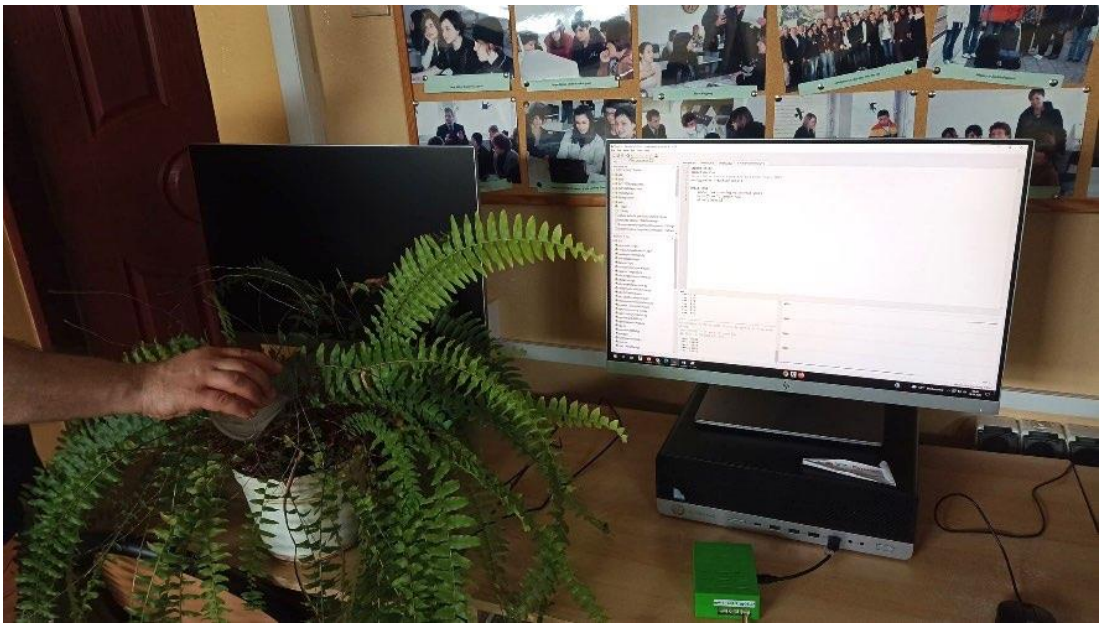
Experiment Evaluation

System testing:

-put sensor into different soils

-run the script

-read the results from Thonny’s terminal



Write down Your results

plant	average result	description of the state from observation	result after adding to the pot small glass of the water

Go further:

The script can be modified to give information on a scale of very dry, dry, normal, wet, very wet.
You can also make sure that this information is passed over the network (SMS, website, LoRa WAN).