Module II. Raspberry Pi, Python Development

1. Intro: What is the learning module about? For whom is it of interest? How will you learn?

This module is about **Raspberry Pi** device and **Python** development in order to interact with the the sensors installed in the Plant. It will be described how to work with this small computer and how to develop software to interact with the sensors and mechanism previously installed.

As the level of software development at this point is not very high , this is interested for the following courses (Spanish Vocational Education and Training Specialities).

- Technician in Telecommunications Installations (EQF4)
- Technician in Microcomputer Systems and Networks (EQF4)
- Higher Technician in Development of Web Applications (EQF5)
- Higher Technician in Multi-platform Applications Development (EQF5)

For those courses with non previous knowledge in software development It is required to give an extra introduction about programming basics and Linux OS commands.

In other countries, this also is to the interest of all the specialities related with electronics, and IT.

Once the devices are connected and installed (Module I), it is time to interact with them.

At this point the first goal will be testing all the sensors with their current data through the Python development.

So the steps will be :

1. Configuring and testing python.
2. Obtaining libraries to connect with sensors
3. Writing connection to sensors with python.
4. Showing results in console

2. Scenario: Narrative task which is presented in an authentic situation.

1. Raspberry Pi . Basics
2. Python IDLE (Hello world)
3. Adding sensors libraries to the project
4. Testing sensor with code

The knowledge learned in this module will be used for the global project Plant Irrigation. Using Raspberry Pi give us a powerful tool which allows future students and teacher to develop and increment the final project easily and scalable. I.E. Adding more sensors.

---

3. Development of the concrete tasks, the work plan, (international) division of work, ways of collaboration (Multidisciplinary or multinational) problem solving, implementation of the tasks

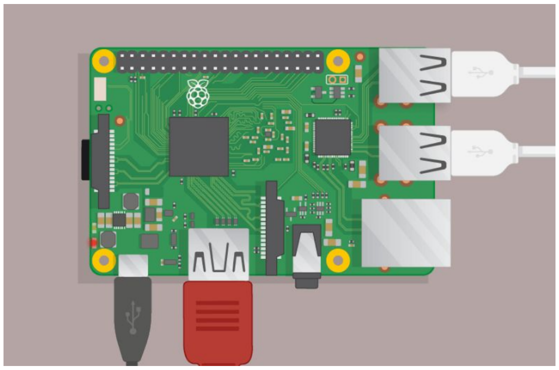
---

TASK 1 Raspberry Pi . Basics

Configure Raspberry Pi 3 with Rasbian OS.

Professional Competences:
- Use basic digital, analogue, and electromechanical components.
- Linux commands

The Raspberry Pi is a small compute.. You plug it into a monitor and attach a keyboard and mouse.



Requirements for the Task:

Hardware

A Raspberry Pi computer with an SD card

A monitor with a cable (and, if needed, an HDMI adaptor)

A USB keyboard and mouse

A power supply

Headphones or speakers (optional)

An ethernet cable (optional)

Software

Raspbian, installed via NOOBS

Installing Raspbian with NOOBS

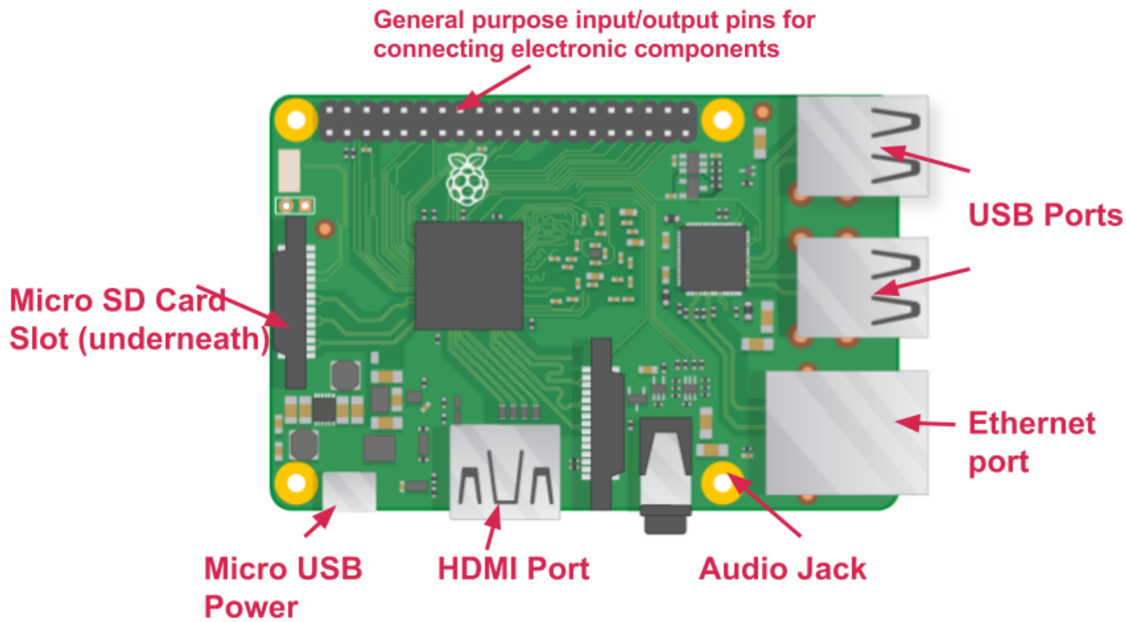
Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware.
http://raspbian.org

HW Requirements: Micro SD card with NOOBS (New Out Of Box Software)
Using NOOBS is the easiest way to install Raspbian on your SD card.

Downloads NOOBS in :
http://www.raspberrypi.org/downloads/

Follow the instructions
https://projects.raspberrypi.org/en/projects/noobs-install

Rasberry Pi. Hardware Basic Knowledge:



USB : To connect mouse and keyboard or other components as USB drive.

SD card slot: OS Installed and Files stored.

Ethernet port: To connect Raspberry Pi to network via cable. Also wifi connection is available.

Audio jack: headphones or speakers connection.

HDMI port: Monitor connection.

Micro USB power connector: power supply connection. Tip: Connect this port in the last position, once you have connected all the others.

GPIO ports: electronic components connections.

Raspberry Pi Connection
SD Card with Rasbian versión installed (via NOOBS).
Connect all the ports (Keyboard, Mouse, Monitor, Power).
Connect Micro USB power connector. Automatically you will see Rasbian on your monitor screen loaded.

Raspberry Pi, is a Linux OS. The files created will be stored in the SD Card.
Terminal can be used with Linux commands.
Wifi or LAN connection is very easy. The same way as in every Linux distribution.
Rasberry Pi, has several apps preinstaled (for different uses).

TASK 2 Python in Raspberry Pi, First Steps

For this task , we will use the program language Python (easy to read and write) and with Raspberry Pi lets you connect your project to the real world through sensor.

Python syntax is very clean, with an emphasis on readability .
First step is opening the IDLE from the desktop.

IDLE gives you a REPL (Read-Evaluate-Print-Loop) which is a prompt you can enter Python commands in to. As it's a REPL you even get the output of commands printed to the screen without using print.

You can use variables if you need to but you can even use it like a calculator. For example:

```
ēēē ú č û
ü
ēēē fYe] đ í3YjY`í
ēēē í(]ddg í č fYe]
ì(]ddg 3YjY`ì
```

IDLE also has syntax highlighting built in and some support for autocompletion. You can look back on the history of the commands you've entered in the REPL with Alt + P (previous) and Alt + N (next).

Some languages use curly braces { and } to wrap around lines of code which belong together, and leave it to the writer to indent these lines to appear visually nested. However, Python does not use curly braces but instead requires indentation for nesting. For example a for loop in Python:

```
^gj a af jYf_]ÝúùÞÆ
    hjaflÝí(]ddgíÞ
```

The indentation is necessary here. A second line indented would be a part of the loop, and a second line not indented would be outside of the loop. For example:

```
^gj a af jYf_]ÝûÞÆ
    hjaflÝí!íÞ
    hjaflÝí"íÞ
```

would print:

```
!
"
!
"
```

whereas the following:

```
^gj a af jYf_]ÝûÞÆ
    hjaflÝí!íÞ
hjaflÝí"íÞ
```

would print:

```
!
```

```
!
"
```

VARIABLES

To save a value to a variable, assign it like so:

```
fYe] đ í"gZí
Y_] đ úþ
```

Note here I did not assign types to these variables, as types are inferred, and can be changed (it's dynamic).

```
Y_] đ úþ
Y_] čđ ú  ø af[j]e]fl Y_] Zq ú
hjaflÝY_]Þ
```

This time I used comments beside the increment command.

COMMENTS

Comments are ignored in the program but there for you to leave notes, and are denoted by the hash # symbol. Multi-line comments use triple quotes like so:

```
ííí
4`ak ak Y n]jq kaehd] 0ql`gf hjg_jYe l`Yl hjaflk í(]ddgíÇ
4`Ylìk Ydd al \g]kÇ
ííí
hjaflÝí(]ddgíÞ
```

LISTS

Python also has lists (called arrays in some languages) which are collections of data of any type:

```
fmeZ]jk đ ßúÄ ûÄ üà
```

Lists are denoted by the use of square brackets [] and each item is separated by a comma.

ITERATION

Some data types are iterable, which means you can loop over the values they contain. For example a list:

```
fmeZ]jk đ ßúÄ ûÄ üà
```

```
^gj fmeZ]j af fmeZ]jkÆ
    hjaflÝfmeZ]jÞ
```

This takes each item in the list numbers and prints out the item:

```
ú
û
ü
```

Note I used the word number to denote each item. This is merely the word I chose for this - it's recommended you choose descriptive words for variables - using plurals for lists, and singular for each item makes sense. It makes it easier to understand when reading.

Other data types are iterable, for example the string:

```
\g_ëfYe] đ í").'/í

^gj [`Yj af \g_ëfYe]Æ
    hjaflÝ[`YjÞ
```

This loops over each character and prints them out:

```
"
)
.
'
/
```

RANGE

The integer data type is not iterable and trying to iterate over it will produce an error. For example:

```
^gj a af üÆ
    hjaflÝaÞ
```

will produce:

```
4qh]%jjgjÆ ìaflì gZb][l ak fgl al]jYZd]
```

However you can make an iterable object using the range function:

```
^gj a af jYf_]ÝüÞÆ
    hjaflÝaÞ
```

range(5) contains the numbers 0, 1, 2, 3 and 4 (five numbers in total). To get the numbers 1 to 5 use range(1, 6).

LENGTH

You can use functions like len to find the length of a string or a list:

```
fYe] đ í*Yea]í
hjaflÝd]fÝfYe]ÞÞ  ø þ

fYe]k đ ßí"gZíÄ í*Yf]íÄ í*Ye]kíÄ í!da[]íà
hjaflÝd]fÝfYe]kÞÞ  ø ý
```

IF STATEMENTS

You can use if statements for control flow:

```
fYe] đ í*g]í

a^ d]fÝfYe]Þ ē üÆ
    hjaflÝí.a[] fYe]ÄíÞ
    hjaflÝfYe]Þ
]dk]Æ
    hjaflÝí4`Ylìk Y k`gjl fYe]ÄíÞ
    hjaflÝfYe]Þ
```

Python files in IDLE

To create a Python file in IDLE, click File > New File and you'll be given a blank window. This is an empty file, not a Python prompt. You write a Python file in this window, save it, then run it and you'll see the output in the other window.

For example, in the new window, type:

```
f đ ù

^gj a af jYf_]ÝúÄ úùúÞÆ
    f čđ a
```

```
hjaflÝí4`] kme g^ l`] fmeZ]jk ú lg úùù akÆíÞ
hjaflÝfÞ
```

Then save this file (File > Save or Ctrl + S) and run (Run > Run Module or hit F5) and you'll see the output in your original Python window.

Executing Python files from the command line

You can write a Python file in a standard editor like Vim, Nano or LeafPad, and run it as a Python script from the command line. Just navigate to the directory the file is saved (use cd and ls for guidance) and run with python, e.g. python hello.py.

Python Documentation

Full documentation for Python is available at https://www.python.org/doc/

TASK 3 Adding sensors libraries to the project

Installing Python libraries

APT

Some Python packages can be found in the Raspbian archives, and can be installed using APT, for example:

```
km\g YhlÈ_]l mh\Yl]
km\g YhlÈ_]l afklYdd hql`gfÈha[Ye]jY
```

This is a preferable method of installing things as it means that the modules you install can be kept up to date easily with the usual `km\g YhlÈ_]l mh\Yl]` and `km\g YhlÈ_]l mh_jY\]` commands.

PIP

Not all Python packages are available in the Raspbian archives, and those that are can sometimes be out of date. If you can't find a suitable version in the Raspbian archives you can install packages from the Python Package Index (also known as PyPI). To do so, use the pip tool (which is installed with the python-pip package in Raspbian:

```
km\g YhlÈ_]l afklYdd hql`gfÈhah
km\g hah afklYdd kaehd]bkgf
```

GPIO

Using Python on the Raspberry Pi opens up the opportunity to connect to the real world through the Pi's GPIO pins. This can be done with the RPi GPIO library. It is preinstalled on recent Raspbian images, but if you have an older one you can install it with:

```
km\g YhlÈ_]l afklYdd hql`gfÈjhaÇ_hag
```

or

```
km\g YhlÈ_]l afklYdd hql`gfüÈjhaÇ_hag
```

In older versions of Raspbian, you'll need root access to access the GPIO pins, so run sudo python, sudo ipython or sudo idle &, but in newer versions, if the user is in the gpio group, you can run it normally.

In your Python script (or in the REPL), import the GPIO module, set the board mode to that of your preference, set up the pins you want to use and turn them on:

```
aehgjl 20aÇ'0)/ Yk '0)/

'0)/Çk]leg\]Ý'0)/Ç"#-Þ  ø k]l ZgYj\ eg\] lg "jgY\[ge

'0)/Çk]lmhÝúÃÄ '0)/Ç/54Þ  ø k]l mh haf úÃ
'0)/Çk]lmhÝúāÄ '0)/Ç/54Þ  ø k]l mh haf úā

'0)/ÇgmlhmlÝúÃÄ úÞ  ø lmjf gf haf úÃ
'0)/ÇgmlhmlÝúāÄ úÞ  ø lmjf gf haf úā
```

TASK 4 Testing sensor with code

In this task we are going to develop a simple connection with a temperature sensor and show the data every second in the console.

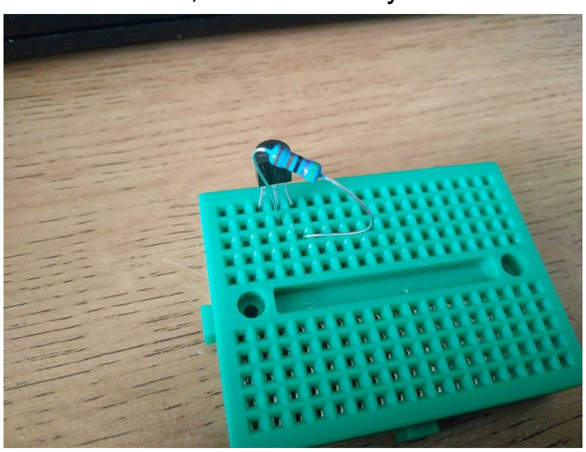
Material needed:
Raspberry Pi
DS18B20 Temperature Sensor
4.7K Ohm Resistor (Colour Code: Yellow Purple Red Gold)
Breadboard
3 x Female to male jumper cables.
1 x Male to Male jumper cable (Optional)

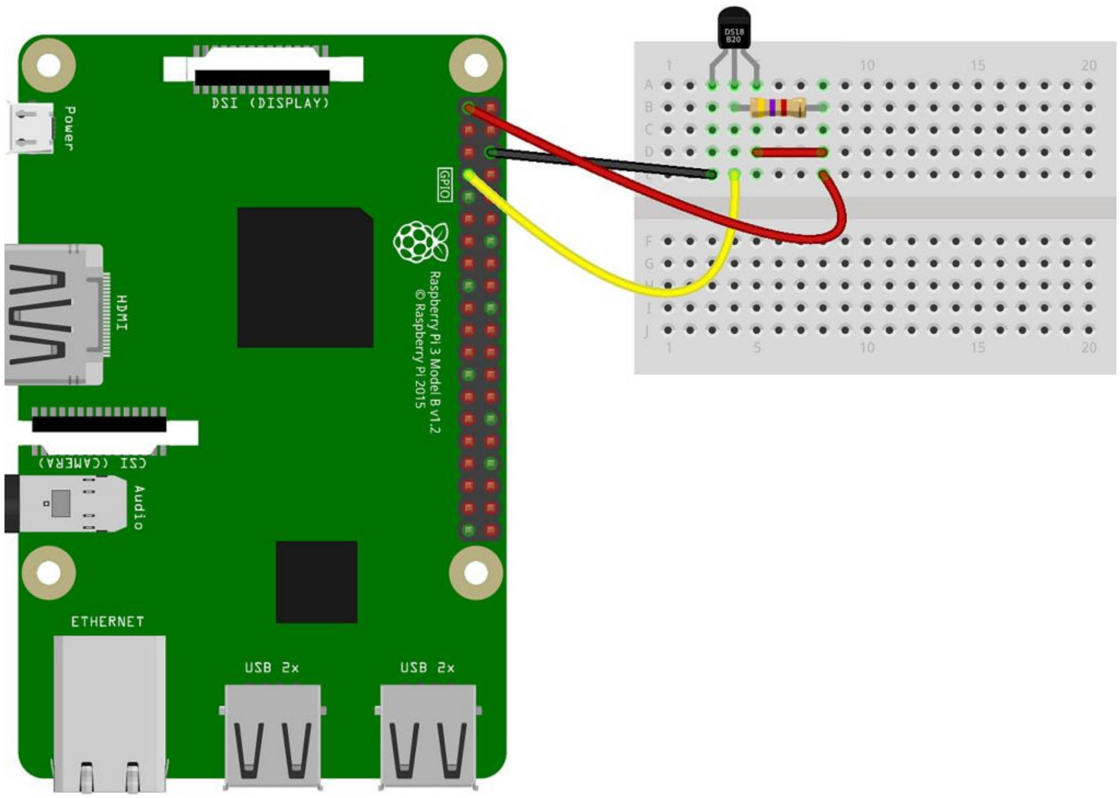
Connections
Raspberry Pi turned off.
DS18B20 placed into the breadboard , the flat side faces you.

The black jumper cable goes from GND, which is the third pin down on the right column to the first pin of the DS18B20.

The yellow jumper cable goes from the fourth pin down on the left column and is connected to the middle pin of the DS18B20.

The red jumper cable goes from the top left pin of the Raspberry Pi to the far right pin of the DS18B20.
The Resistor connects the RIGHT pin to the MIDDLE pin.
This is called a pull up resistor and is used to ensure that the middle pin is always on.
In the diagram it is used a spare red wire to show this connection. But using the resistor to make the connection, is the best way.





Configuring the Raspberry Pi

Now attach keyboard, mouse, HDMI and power to your Raspberry Pi and boot to the desktop. Two steps to enable our DS18B20 for use.

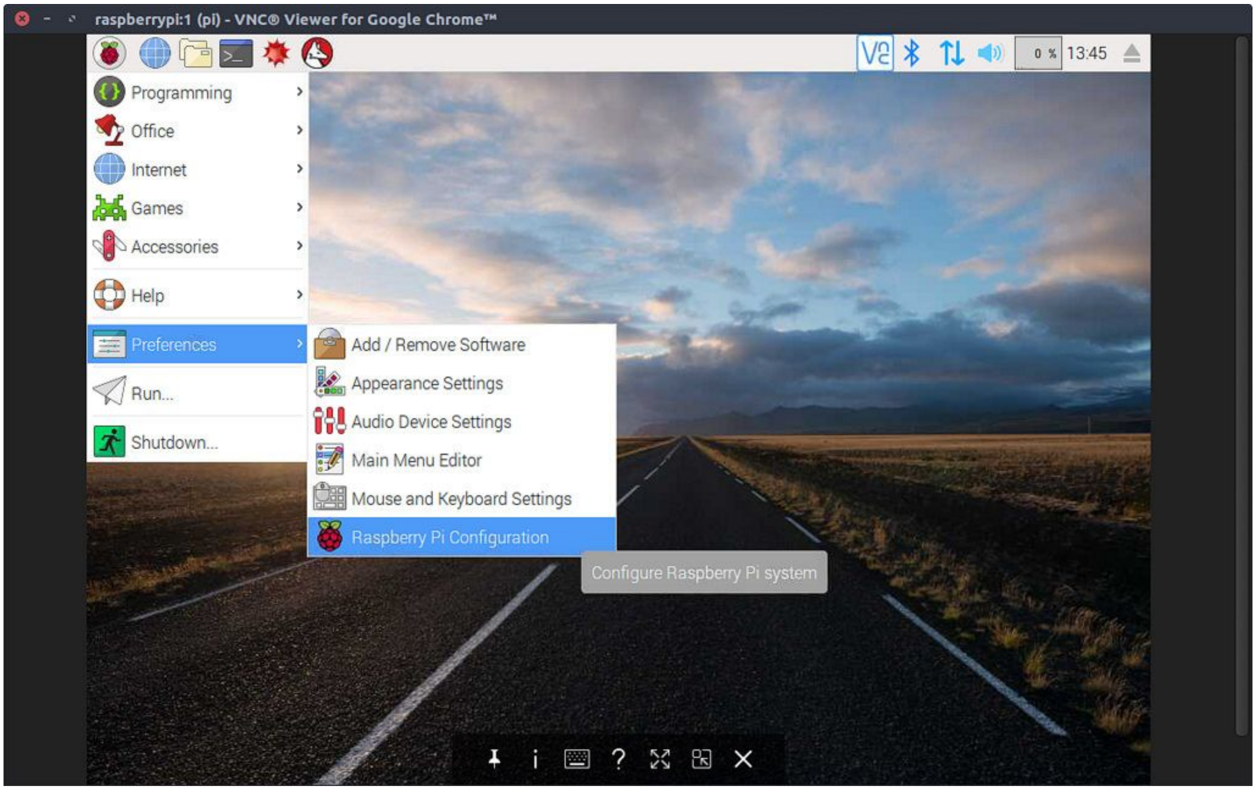
Install the Python Library

Firstly install a Python library, pre-written code that enables the Python code that we shall later write to talk to the sensor.
The Python library is called w1thermsensor and to install it is needed to use the Terminal.

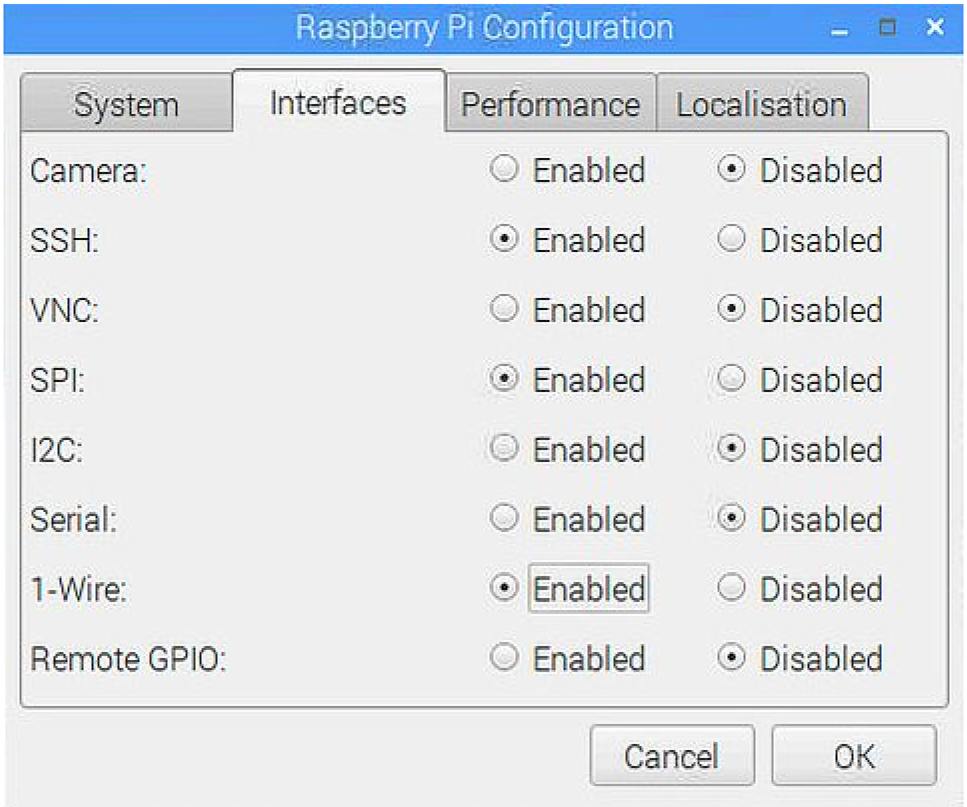
```
km\g hahü afklYdd oúl`]jek]fkgj
```

Enable the Interface

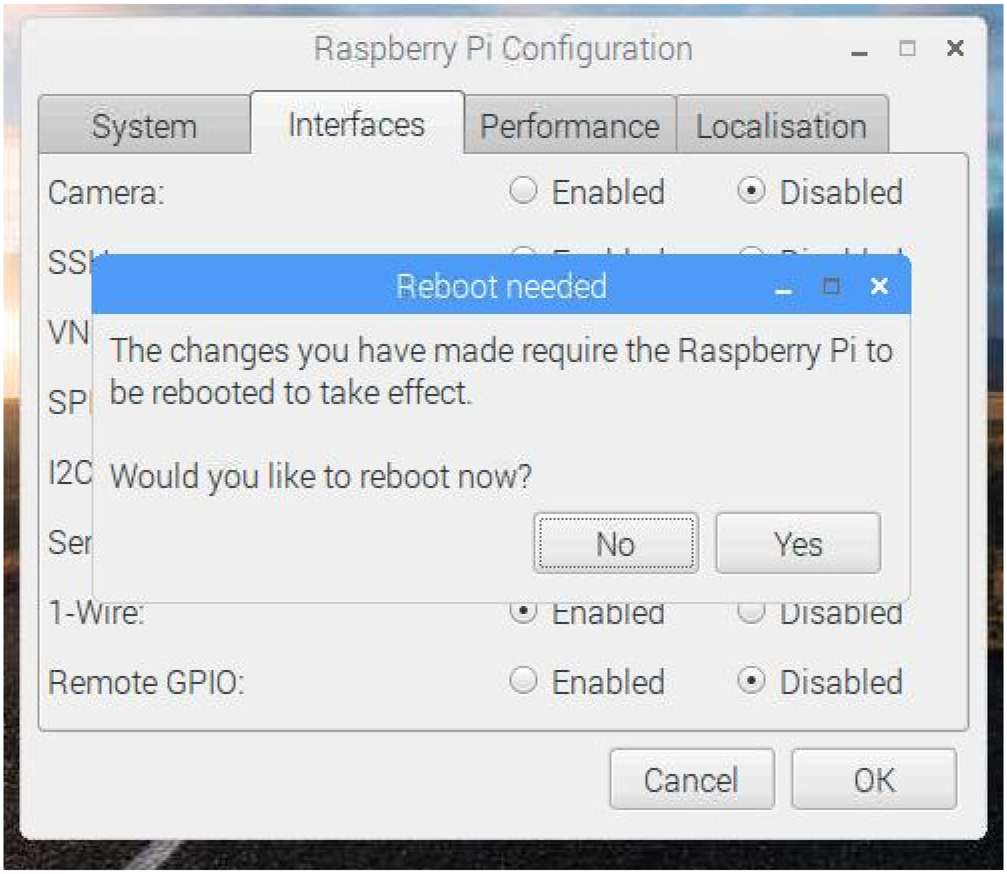
The DS18B20 uses a 1 wire serial interface, this is the middle pin of the sensor, that is connected to the Raspberry Pi via the yellow wire in the diagram. It is needed to tell Raspberry Pi that it is used this pin and to do that it is needed to use the **Raspberry Pi Configuration tool**, founded in the **Preferences menu**.



Ant then enables the **1-Wire** interface



And reboot needed



Python Code

The project will take the temperature from DS18B20 sensor every one second and print it to the screen. It is an infinite loop.

Needed to use **Python3 IDLE**

click on **File >> New** to create a new blank document. **File >> Save** call the project temperature-sensor.py

Import the libraries
First step in any Python project that uses libraries is to import the used libraries.
Time Library → to control how often the sensor data is collected
w1thermsensor Library → to enable the project to talk to the sensor.

```
aehgjl lae]
^jge oúl`]jek]fkgj aehgjl 7ú4`]je3]fkgj
```

Sensor

Then create an object to store a connection to the sensor.

```
k]fkgj đ 7ú4`]je3]fkgjÝþ
```

The loop

To get data from the temperature sensor every second, and run forever. It is used a while True loop.

```
o`ad] 4jm]Æ
```

Important, that next lines of code are indented.

First get the current temperature from the **DS18B20** sensor, and then store it in a variable called temperature.

```
    l]eh]jYlmj] đ k]fkgjÇ_]lël]eh]jYlmj]Ýþ
```

Ant then format the data and print it to the screen.

```
    hjaflÝí4`] l]eh]jYlmj] ak Čk []dkamkí Č l]eh]jYlmj]þ
```

And finally wait for 1 second between taking a temperature reading.

```
    lae]Çkd]]hÝúþ
```

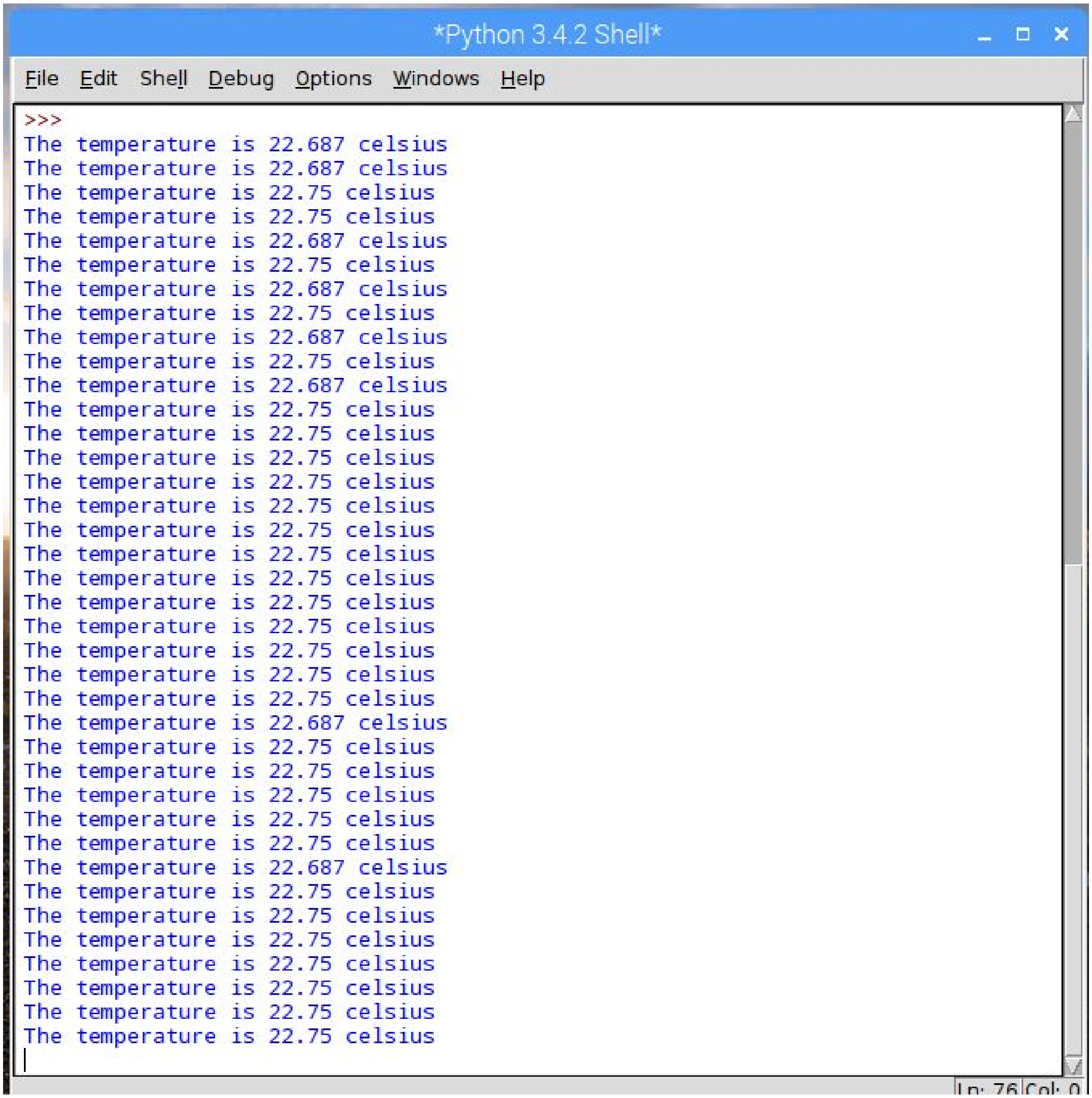
Complete Code

```
aehgjl lae]
^jge oúl`]jek]fkgj aehgjl 7ú4`]je3]fkgj
k]fkgj đ 7ú4`]je3]fkgjÝþ

o`ad] 4jm]Æ
    l]eh]jYlmj] đ k]fkgjÇ_]lël]eh]jYlmj]Ýþ
    hjaflÝí4`] l]eh]jYlmj] ak Čk []dkamkí Č l]eh]jYlmj]þ
    lae]Çkd]]hÝúþ
```

Running the code

Run >> Run Module the Python Shell window pop up and start displaying temperature data.



4. Assessment of training success

The main assesment in this training is to develop the basics skills in the second submodule of the project Plant Irrigation.

The main goals in the students learning is to manage Sensors data with Raspberry Pi and Python.

Once this goal is achieved the student will be able to do the same with the global project and understand the following steps.

---

5. Meta-cognitive self-reflexion and evaluation of the learning process

At this point we have to discuss with the students which other possibilities are with this technologies.
It has to be a discussion forum to first search for different sensors in ADAFruit and then give them possible profesional IOT Solutions.