INTERNET OF THINGS

LAB ASSIGNMENT:

"Raspberry"

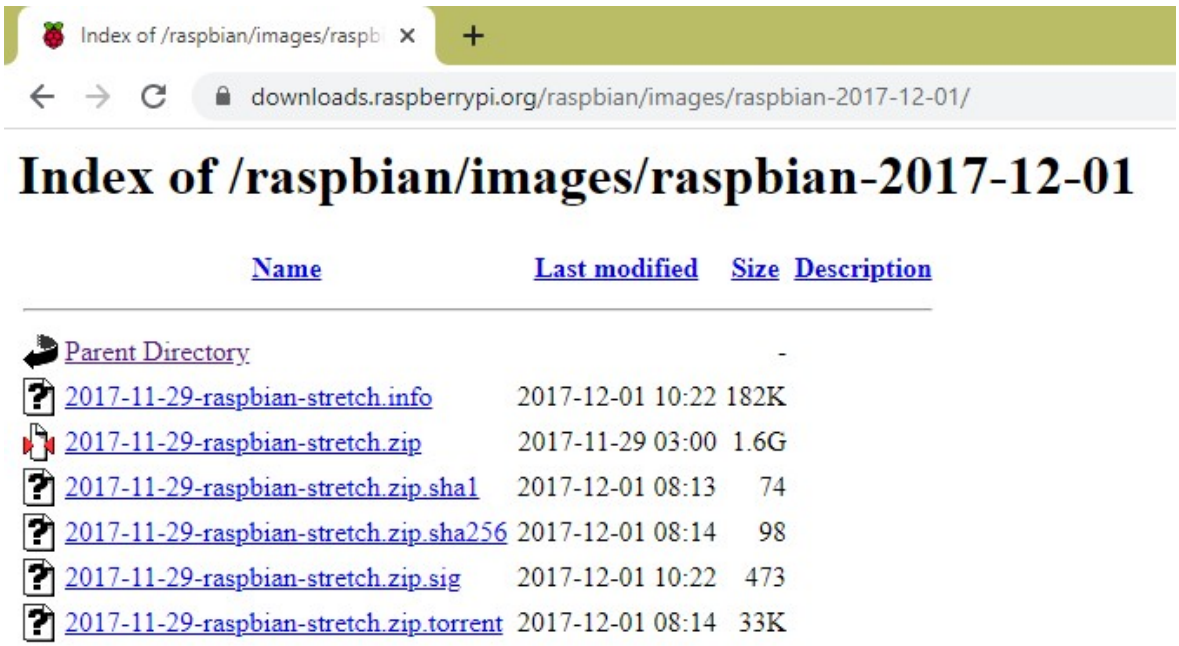
Name:	
ID:	
Date:	

- TOPICS ADDRESSED: Raspberry.
- EQUIPMENT AND MATERIALS REQUIRED: Personal computer connected to Internet. PDF report required can be filled directly as a form, or using free online version of pdfescape (`www.pdfescape.com`), mainly to insert images in the blank spaces provided. Find out your student ID since it will be necessary to correctly complete some of the tasks proposed.
- THEORETICAL CONCEPTS: Raspbian operating system. Python programming. Raspberry GPIO.
- ESTIMATED DURATION: 120 min.
- DESCRIPTION: Follow the instructions provided to learn about all the steps that involve working with a Raspberry Pi as an IoT node, from its configuration and set up, to its programming and its hardware interfacing capabilities.
- **ASSESSMENT:** Hand in a report of the work done and results achieved, using the present document as form template (Include your ID in the name of the file). **The student pledges by his/her honor that he/she has not given or received any unauthorized assistance on this assignment.** Cheating is a serious violation of the academic rules and the consequences can be grave and permanent.

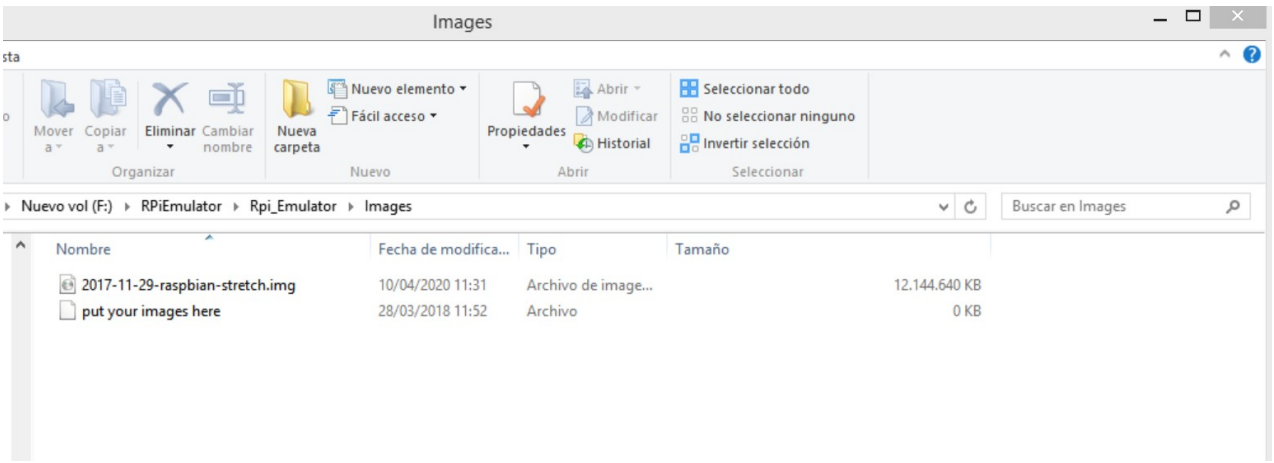
STEPS:

1) Install the `Rpi Emulator` from the zip file provided (simply decompress it in a directory).

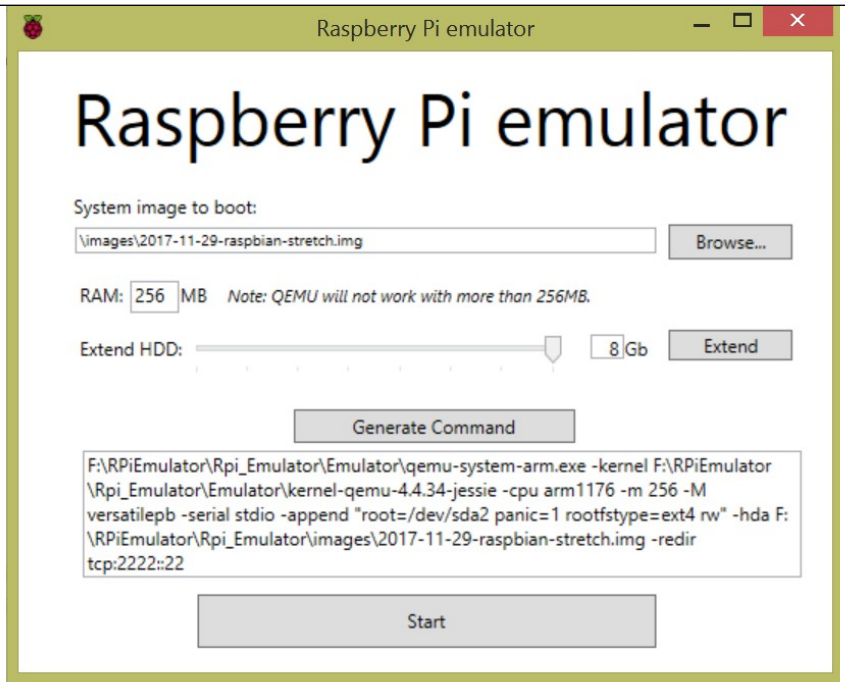
2) Download a Raspbian image (zip file) from the repository https://downloads.raspberrypi.org/raspbian/images/ . Specifically, download the image whose month's unit coincides with your ID's units. For example, if your ID is 15, you can download any file that matches the template `raspbian-XXXX-X5-XX`.



3) Uncompress the Raspbian image and copy it to the directory `Images` of the RPI-Emulator (save a copy of the original image file to restore it in case something goes wrong when running the emulator):



4) Run the emulator (`RPI Emulator.exe`) and edit its splash screen (shown below). First browse and select the Raspbian image file to boot (dowloaded in a previous step), and then select the maximum size for the HDD, 8 Gb. Press `Generate Command`, and then `Start`:



5) Insert below a screenshot of your splash screen just before clicking on `Start`, as in the previous example:

6) Open a terminal window and execute the command `ifconfig`. Insert below a screenshot of the command's result:

7) Install in your personal computer the working environment for Python **PyCharm** (Community version, free).
8) Create a new project, called `ProjectID`, where `ID` is your student ID.
9) Create a new file for the main program by means of `File ⇨ New ⇨ Python File`. Use your initials for the program name.
10) Develop the following programs in Python using the PyCharm environment:
    a) Write a program in Python that gets from the user his/her full name, and then shows the name entered in reverse order, skipping one every two letters. Include the source code of the final program below:

```
# Method in order to reverse the name
def reverseName(name):
    nameLength = len(name)
    reversedName = [''] * nameLength  # Initialize a list with the same length as the name

    for i in range(nameLength):
        # -1 because we count nameLength from 0 onwards
        reversedName[i] = name[nameLength - 1 - i]
    # join()-call on empty string ''which concatenates all elements off the list
    return ''.join(reversedName)

def main():
    full_name = input("Enter your full name: ") #input() gets user input from the console
    reversed_name = reverseName(full_name)
    print("Reversed name:", reversed_name)

#ensures that the program is only run when the script is run directly, not when it is
imported
```

b) Write a program that asks the user for 20 numerical values, and by means of the `filter` function, shows all the values entered between 10 and 20. Insert below a screenshot where the program output and the source code can both be seen simultaneously:

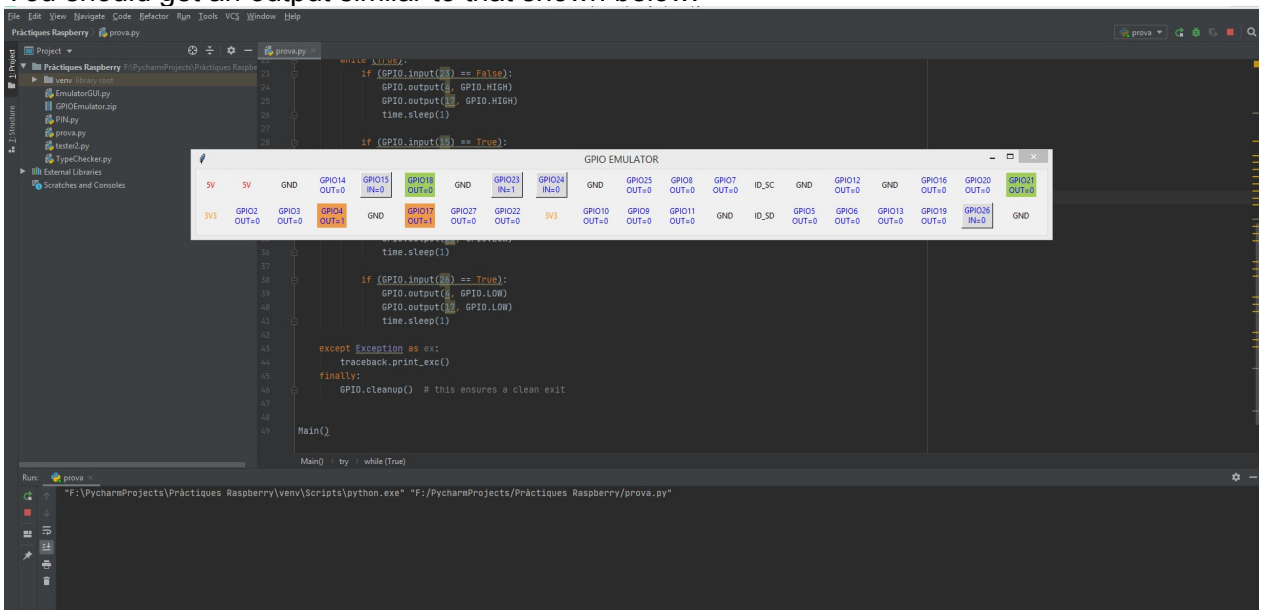
```
AMOUNTOFVALUES = 20

def userInput():
    # define a empty list
    user_input = []
    # loop from 0 to 19
    for i in range(AMOUNTOFVALUES):
        # convert the user input to an integer
        value = int(input(f"Enter value {i + 1}: "))
        # append the value to the list
        user_input.append(value)
    return user_input

def filterValues(user_input):
    # value: what is extractes, for value in user_input: which values to extract # if 10 <= value <= 20 condition which has to be met
    filtered_values = [value for value in user_input if 10 <= value <= 20]
    return filtered_values
```

GPIO emulator

11) Create a new PyCharm project in your computer and write down the directory where the project's files are located.
12) There is a software emulator of the Raspberry GPIO available at the site: https://roderickvella.wordpress.com/2016/06/28/raspberry-pi-gpio-emulator/ . Downlad the zip file of the emulator and decompress the files in the directory of your project. With this simulator it is possible to use the following functions of the RPi.GPIO library: `GPIO.setmode()`, `GPIO.setwarnings()`, `GPIO.setup()`, `GPIO.input()`, `GPIO.output()`.
13) Run the demo program included at the same web site and make sure everything works fine. You should get an output similar to that shown below:



14) Using the previous emulator in the same way as in the example, develop a program that turns GPIO14 and GPIO04 ON and OFF with a period of 1s using the `sleep` function of the `time` module. Insert the resulting code below:

```
import RPi.GPIO as GPIO
import time

# Set GPIO mode
GPIO.setmode(GPIO.BCM)

# Define GPIO pins
pin_14 = 14
pin_04 = 4

# Set up GPIO pins
GPIO.setup(pin_14, GPIO.OUT)
GPIO.setup(pin_04, GPIO.OUT)

try:
    while True:
        # Turn GPIO14 ON and GPIO04 OFF
        GPIO.output(pin_14, GPIO.HIGH)
        GPIO.output(pin_04, GPIO.LOW)
        time.sleep(1)  # Sleep for 1 second

        # Turn GPIO14 OFF and GPIO04 ON
        GPIO.output(pin_14, GPIO.LOW)
```

15) A distance measure sensor based on ultrasounds (generates an ultrasound pulse and measures the time elapsed until the echo is received back) has the following pins:

- VCC: Power in.
- TRIG: Trigger input.
- ECHO: Echo output.
- GND: Ground.

If we assume the pins are correctly connected, and specifically the control/data ones:

- TRIG: Activates the sensor. Connected to an GPIO output.
- ECHO: The pulse echo is received here. Connected to an GPIO input.

Develop a program for Raspberry using the GPIO emulator that implements distance measurement using this sensor, in a similar way as done in www.instructables.com/id/Simple-Arduino-and-HC-SR04-Example/. Choose inputs and outputs at your will. Insert below a screenshot of the source code and the program output:

```
import time
import random

# Define GPIO pins
TRIG = 23  # GPIO pin for TRIG
ECHO = 24  # GPIO pin for ECHO

# Function to simulate distance measurement
def measure_distance():
    # Simulate sending trigger pulse
    print("Triggering sensor...")
    time.sleep(0.00001)

    # Simulate receiving echo pulse
    echo_time = random.uniform(0.01, 0.1)  # Random echo time between 10ms to 100ms
    print(f"Received echo after {echo_time} seconds")

    # Calculate distance (speed of sound = 343 m/s)
    distance = echo_time * 343 / 2  # Divide by 2 as the pulse travels to and from the object
    return distance

# Main function
def main():
    print("Starting distance measurement...")
    while True:
        distance = measure_distance()
        print(f"Distance: {distance:.2f} cm")
        time.sleep(1)  # Delay for 1 second before the next measurement

if __name__ == "__main__":
    main()
```

TRIG pin:
- initiate the measurement process by sending out ultrasonic sounds
- when it receives a high-level signal, it sends out a high-signal wave

ECHO pin:
- receive the echo of the transmitted ultrasonic pulse
- When the ultrasonic pulse encounters an obstacle and reflects back to the sensor, the ECHO pin generates a pulse whose duration is proportional to the time taken for the pulse to travel to the obstacle and back

Explanation of 343/2
- The speed of sound in air at room temperature is approximately 343 meters per second (m/s)
- The ultrasonic pulse travels from the sensor to the object and then reflects back to the sensor. This means the pulse travels a round-trip distance
- To calculate the one-way distance from the sensor to the object, we divide the total distance traveled (round-trip distance) by 2