	TP
C. RODRIGUES	**Internet of Things**

TP 1 : Questions

Goals:

1. **Learn to manipulate a Raspberry Pi**
2. **Learn to assemble and use different devices with a Raspberry Pi**
3. **Implement simple applications in Python, using sensors and actuators connected to the Raspberry**

II – Exercises:

1. **Before proceeding to the exercises, you MUST complete the tutorial examples to understand how to assemble and manipulate the devices we are going to use during this course.**
2. **You will deliver a report that will contain the python code and a picture or sketch (preferable) of the model you assembled for each question.**

Exercise 01: Redo LED Tutorial, but instead of a fixed time, read the interval time from the user. Note: use the Python instruction x = input("Enter a valor:") to prompt and read a value from the user. Also, when you use the instruction int(X) you can convert any value to integer.

Exercise 02: Prototype a system with 2 LEDs and alternate the blink between them. When LED 1 is on, LED 2 is off; when LED 2 is on, LED 1 is off… and so on.

Exercise 03-a: Make a new project with a button and one LED. When the button is pressed the LED must light up.

Exercise 03-b: As instructed on Exercise 03-a, make a new project where the user presses the button, and the LED turns on. The LED is only turned OFF after the user presses the button again. In sum: one press turns the light on, another press turns the light off.

Exercise 04: Redo Exercise 03 using 3 buttons and 3 LEDs. Each button, respectively, will turn on/off one LED. You may use the traffic light device.

Exercise 05: Simulate a traffic light using the red, yellow, and green LEDs. You may use the traffic light device.

Exercise 06: Redo exercise 05 altering the functionality of the traffic light. The traffic light will start with the red light on, and it will **only** change colors when the button is pressed.

Exercise 07: Game - Secret Number: This application defies the user to try discovery a number among 1 and 20 by the console/terminal. The user will guess any number among the values 1 and 20. The user has 4 attempts. If the user wins, the green LED will turn on. If the user informs a lower number than the secret one, the red LED blinks, and if the user informs a greater number than the secret one, the blue LED blinks. Note: the instruction random.random() returns a float number among 0 and 1. So, to take a random number among 0 and 20, follow the instruction: random.random() * 20. For more information about random command, check this link: https://docs.python.org/2/library/random.html.

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

secret = random.random() * 5
x = int(input("Inform  a number [1..5]:"))

if (x == secret):
    print("You found it")
else:
    print("Sorry, but you lose!")
```
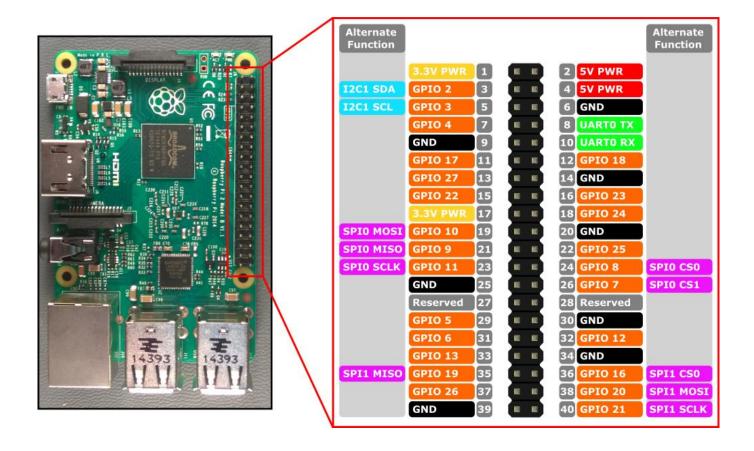
Exercise 08: Redo exercise 05 adding the follow functionality: when the red LED turns on, the buzzer must beep 3 times; when a yellow LED turns on, the buzzer must beep 2 times; when a green LED turns on, the buzzer must beep one single time.

Exercise 09: Build a system composed of one LDR, one buzzer, and one LED. This system works as follows: when the environment (e.g., room, classroom, bedroom, etc.) is bright, the buzzer and LED, which are connected in the Raspberry PI, must be off. However, when the environment is dark (i) the buzzer must beep for 2 seconds and (ii) the LED must be turned on until the environment is bright again.

Exercise 10: Using the RGB LED, make a new project following the Table MENU (below). The system shows the MENU with possible colors and options. Your program must read the user choice, perform the action and go back to the MENU to wait for another user order. It ends after the user selects the option 0 from the Menu.

MENU	3.Blue ON	6. Green OFF	9. Random color
1. Red ON	4. Blue OFF	7. Turn ON ALL	0. Exit
2. Red OFF	5. Green ON	8. Turn OFF ALL	

For more information about Raspberry Pi GPIO ports, check:

https://pinout.xyz