Yefan GUO, Yiman LIN

Report IoT TP1

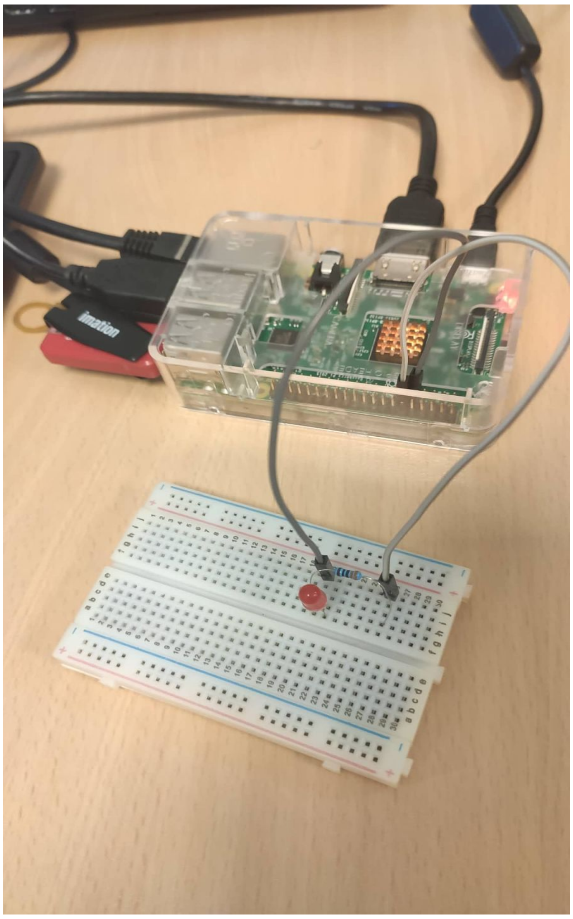
Directions: This document shows for each exercise of this practical work:
- how we set up the device using Raspberry Pi 3,
- and following, for some cases which may require a remote control by the user, screenshots in the console.

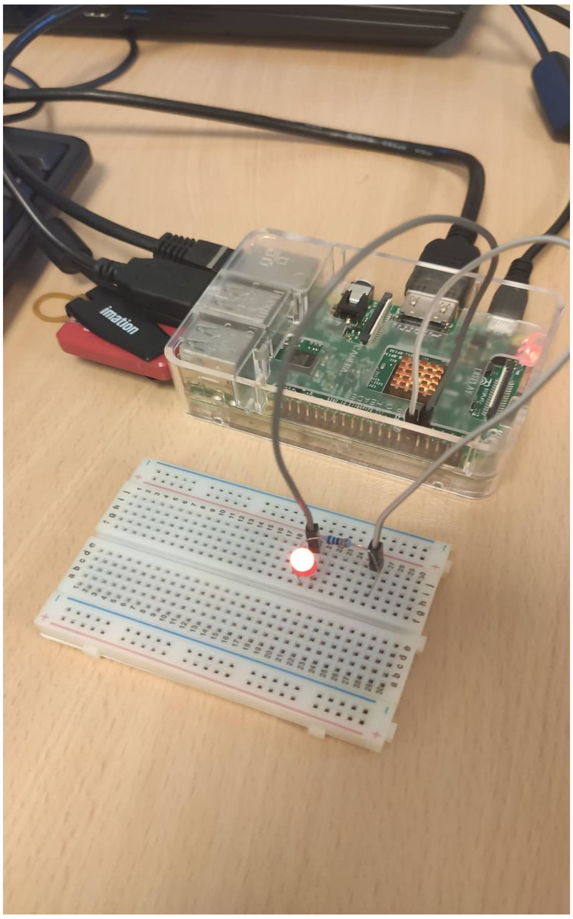
Exercise 01

Make a LED blink every 0.5 seconds.

Exercise 02

Redo exercise 01, 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.

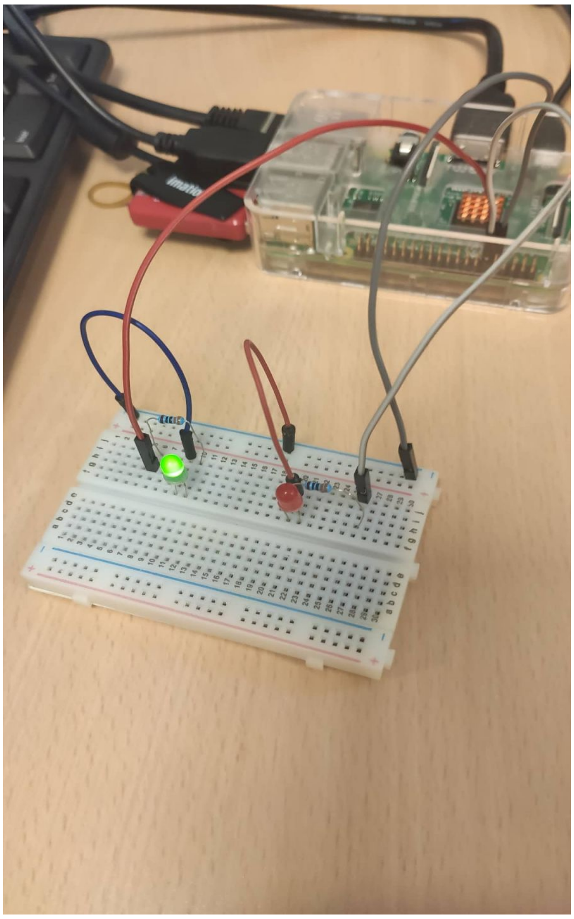




Yefan GUO, Yiman LIN

Exercise 03

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.



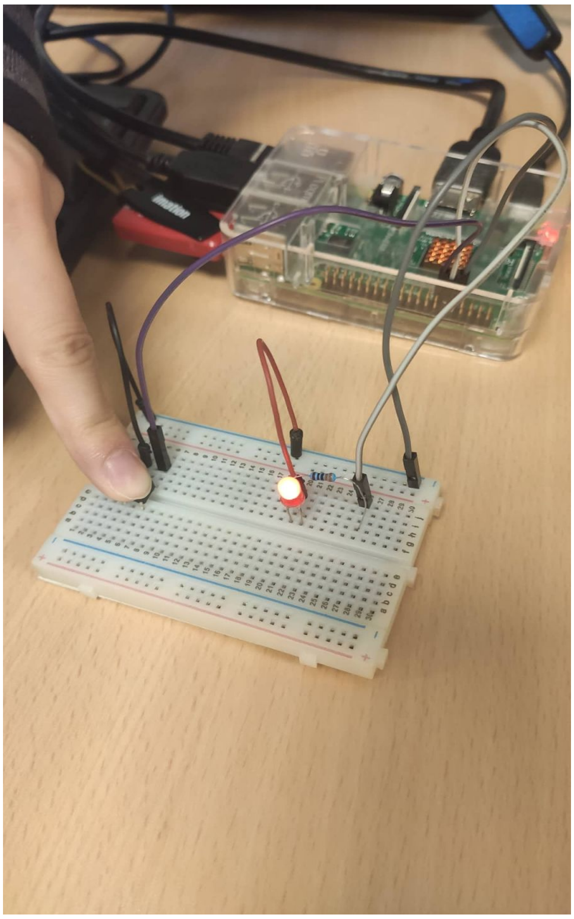
Yefan GUO, Yiman LIN

Exercise 04-a

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

Exercise 04-b

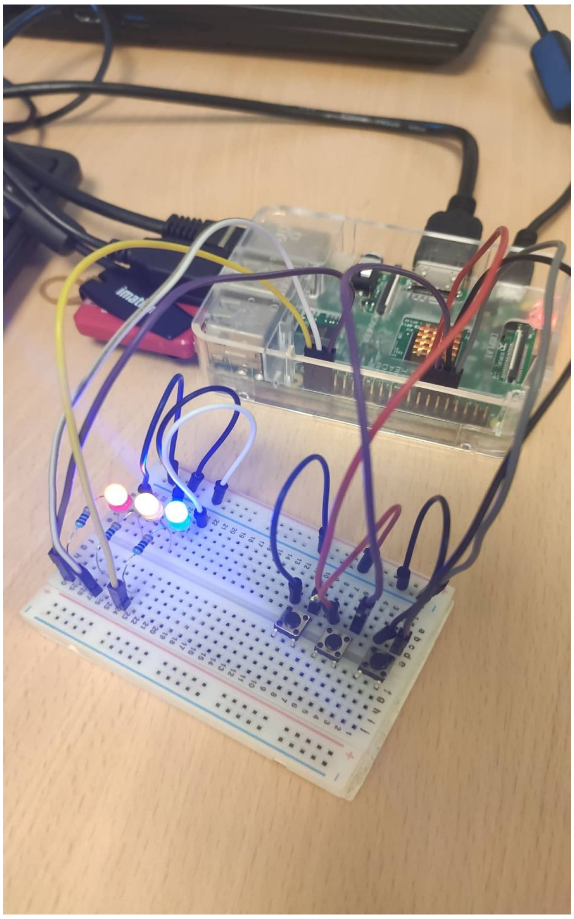
As instructed on Exercise 04a, make a new project where the user presses the button and the LED turns on. The LED is only turned OFF after the user presses the button again.
In sum: one press turns the light on, another press turns the light off.



Yefan GUO, Yiman LIN

Exercise 05

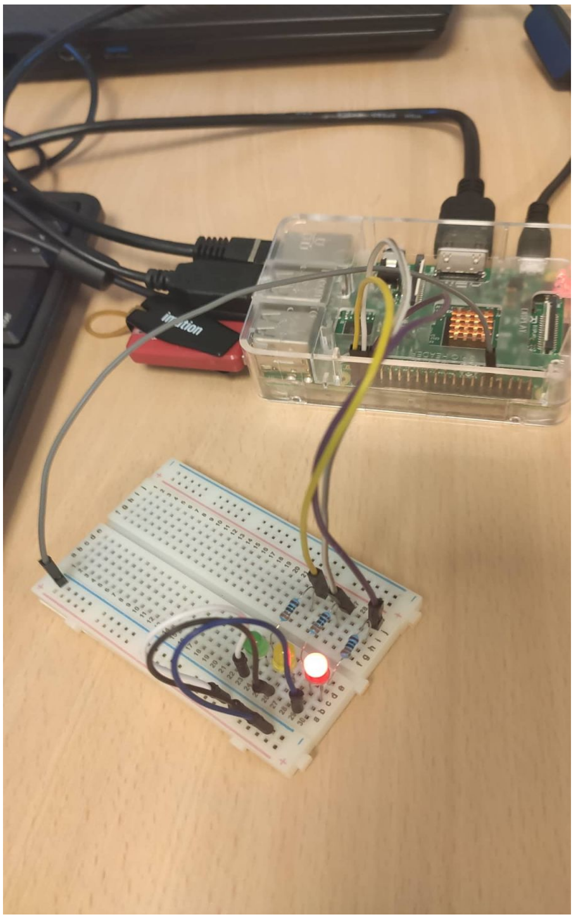
Redo Exercise 04 using 3 buttons and 3 LEDs. Each button, respectively, will turn on/off one LED.



Yefan GUO, Yiman LIN

Exercise 06

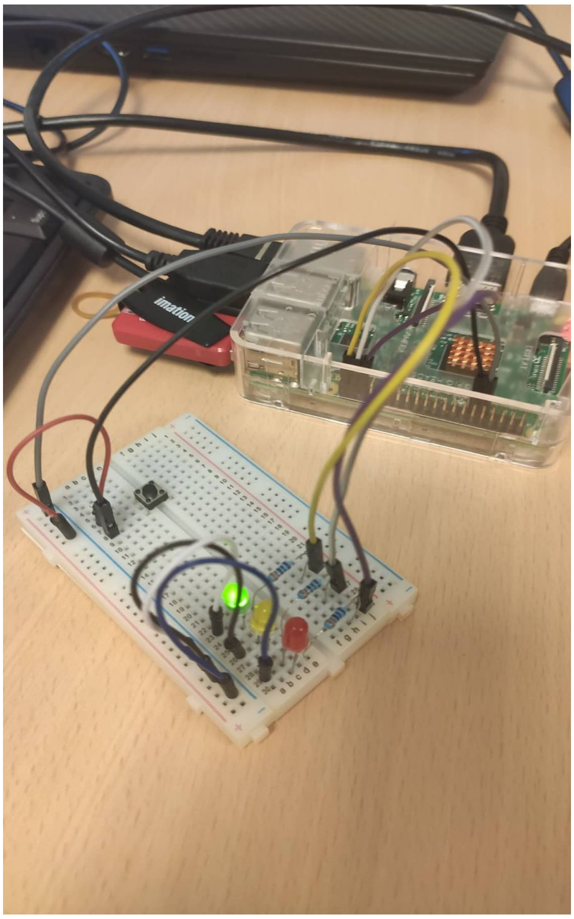
Simulate a traffic light using the red, yellow and green LEDs,



Yefan GUO, Yiman LIN

Exercise 07

Redo exercise 06 altering the functionality of the traffic light. The traffic light will have the red light on, and it will only change colors when the button is pressed.

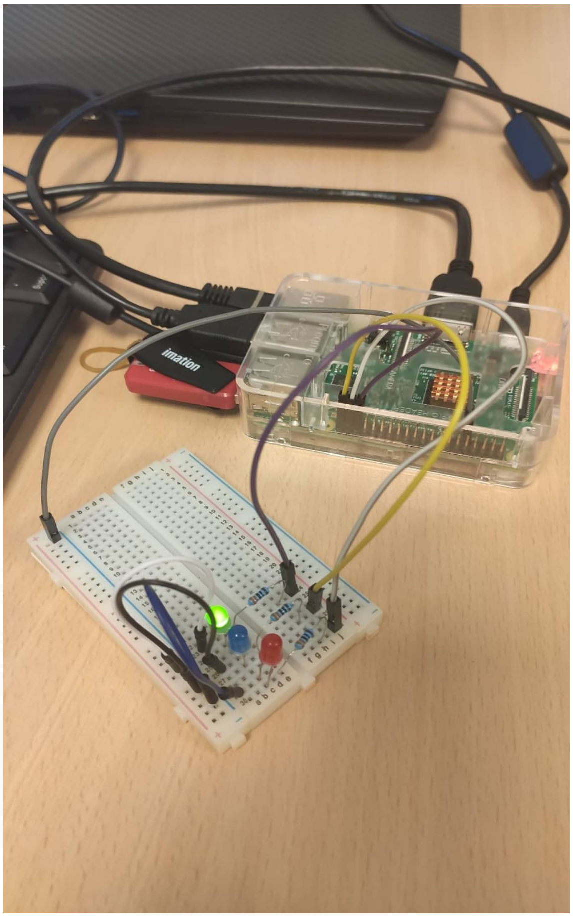


Yefan GUO, Yiman LIN

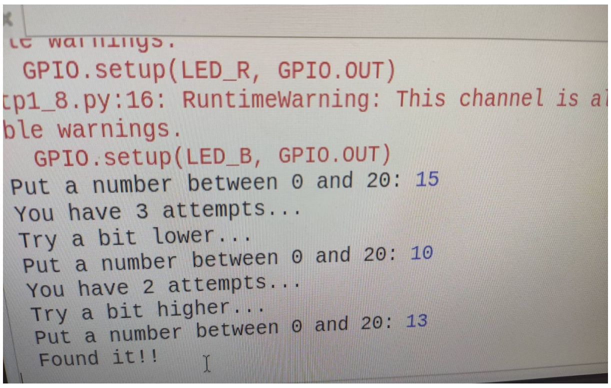
Exercise 08

Game - Secret Number
This application challenges the user to try to finda number among 1 and 20 by the console. 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.

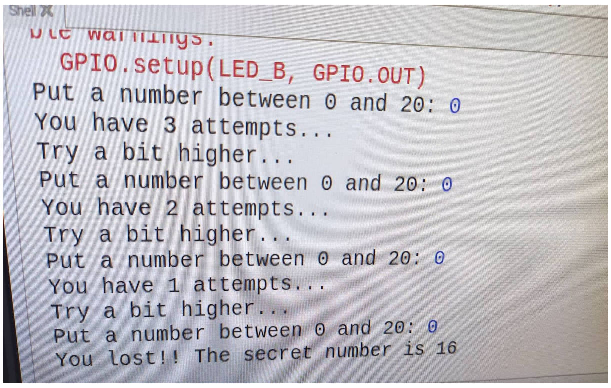


Output when secret found



```
le warnings.
  GPIO.setup(LED_R, GPIO.OUT)
tp1_8.py:16: RuntimeWarning: This channel is a
ble warnings.
  GPIO.setup(LED_B, GPIO.OUT)
Put a number between 0 and 20: 15
You have 3 attempts...
Try a bit lower...
Put a number between 0 and 20: 10
You have 2 attempts...
Try a bit higher...
Put a number between 0 and 20: 13
Found it!!
```

Output when all the attempts used up

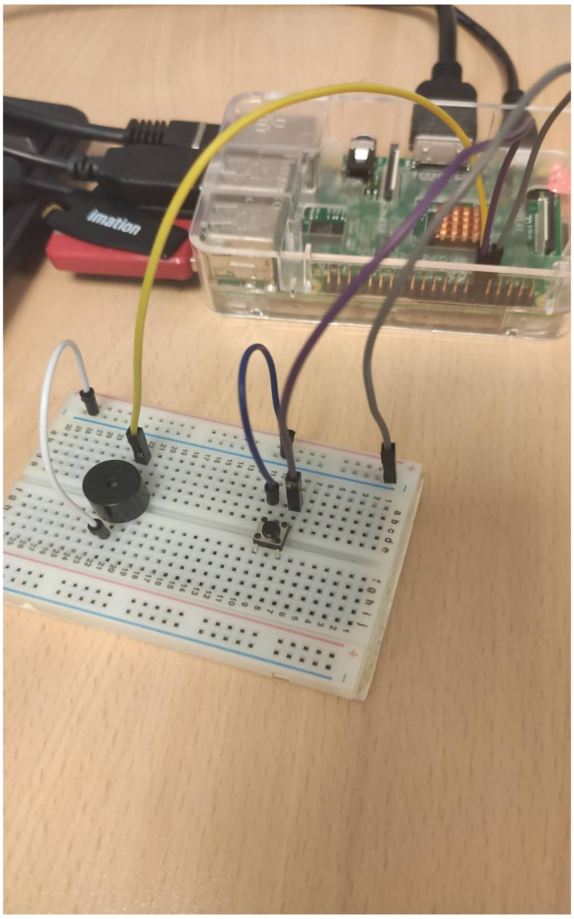


```
ble warnings.
  GPIO.setup(LED_B, GPIO.OUT)
Put a number between 0 and 20: 0
You have 3 attempts...
Try a bit higher...
Put a number between 0 and 20: 0
You have 2 attempts...
Try a bit higher...
Put a number between 0 and 20: 0
You have 1 attempts...
Try a bit higher...
Put a number between 0 and 20: 0
You lost!! The secret number is 16
```

Yefan GUO, Yiman LIN

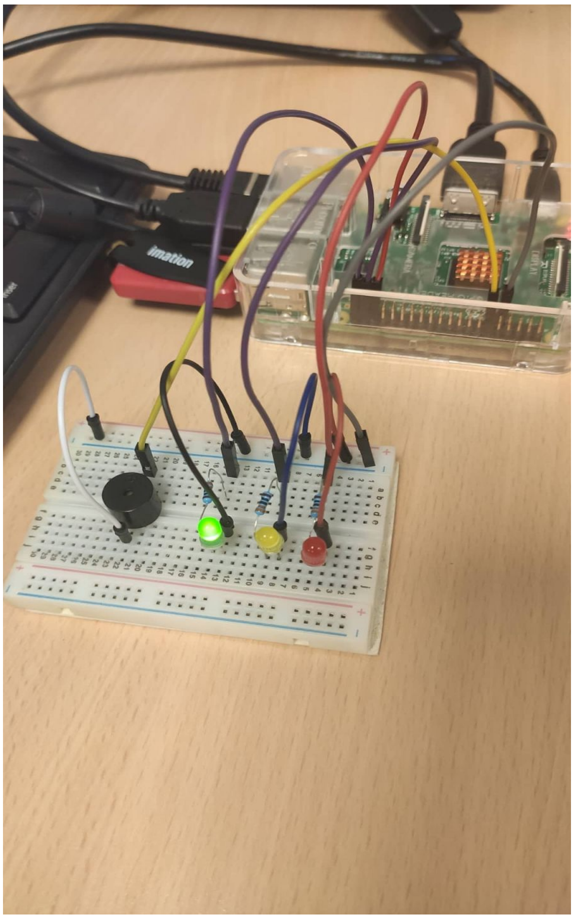
Exercise 09

Design a system composed of a buzzer and a button. When the button is pressed, the buzzer must activate a beep.



Exercise 10

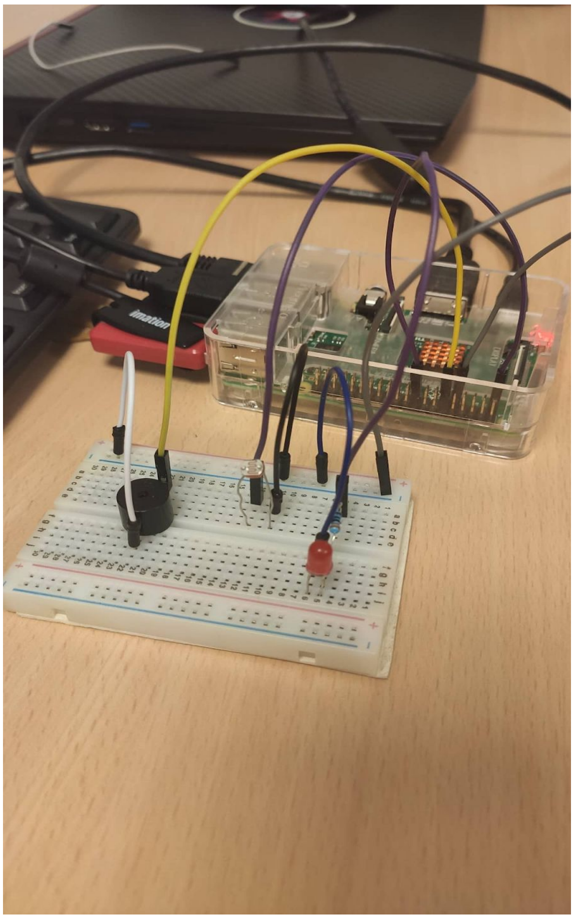
Redo exercise 06 adding the follow functionality: when the red LED turns on, the buzzer must beep 3 times; when a yellow LED turns on, the buzzer must beep 2 times; when a green LED turns on, the buzzer must beep one single time.



Yefan GUO, Yiman LIN

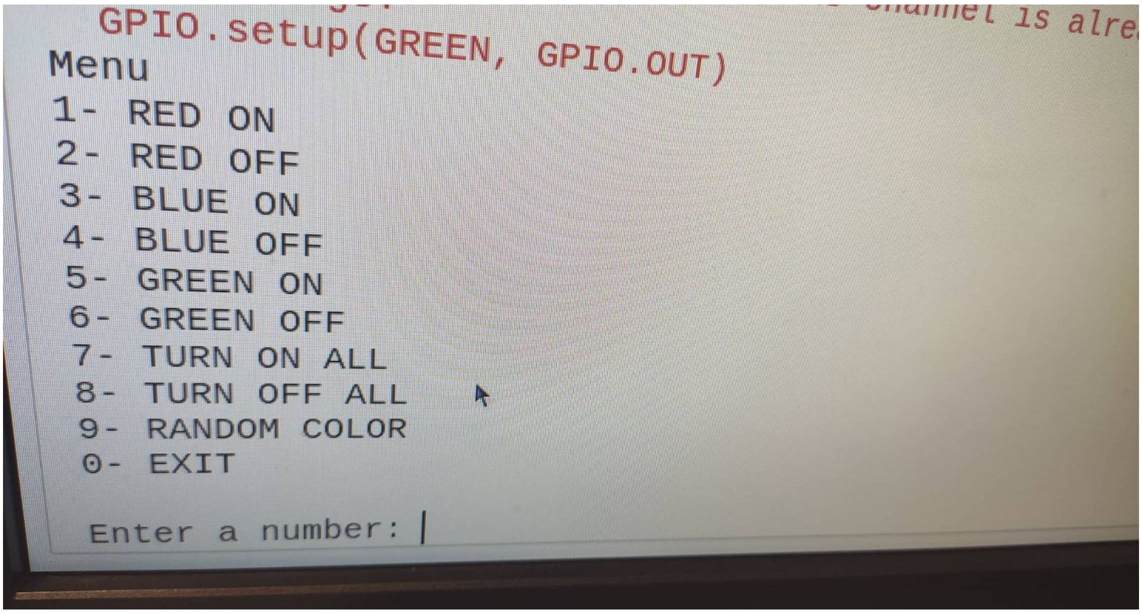
Exercise 11

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 12

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.



```
                                              channel is alre
  GPIO.setup(GREEN, GPIO.OUT)
Menu
1- RED ON
2- RED OFF
3- BLUE ON
4- BLUE OFF
5- GREEN ON
6- GREEN OFF
7- TURN ON ALL
8- TURN OFF ALL
9- RANDOM COLOR
0- EXIT

Enter a number: |
```