Universidad Nacional Autónoma de México
Facultad de Ingeniería
Laboratorio de Fundamentos Embebidos

Práctica 2:
Puertos GPIO Raspberry Pi

Integrantes:
Cárdenas Cárdenas Jorge
Garrido Sánchez Samuel Arturo
Murrieta Villegas Alfonso
Reza Chavarría Sergio Gabriel
Valdespino Mendieta Joaquín

Práctica 2: Puertos GPIO Raspberry Pi

Objetivo:

Comprobar el funcionamiento de los puertos GPIO en la tarjeta Raspberry pi, mediante la utilización del módulo gpiozero para Python.

Introducción:

El uso de microcomputadoras para aplicaciones como el IoT es sin duda una de las mayores tendencias en el mercado desde hace varios años, en la presente práctica a través del uso de bibliotecas para el funcionamiento y uso de los puertos de entrada y salida de nuestra Raspberry Pi es como realizaremos algunas aplicaciones básicas con dispositivos electrónicos básicas como push-bottons o leds.

Como parte también necesario de esta práctica, es fundamental conocer lenguajes como Python para poder inteeractuar con el hardware de nuestros ejercicios.

Desarrollo y resultados:

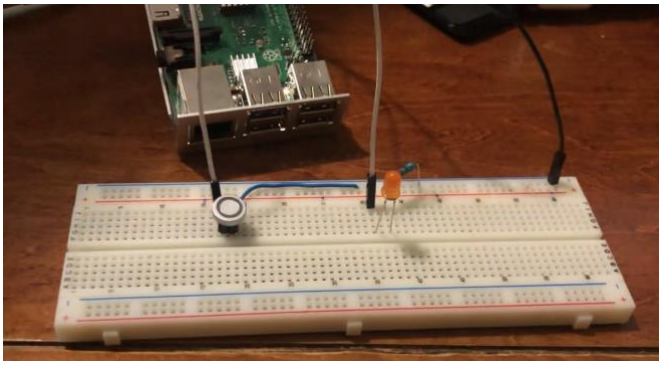
1. Actividades

A continuación, se muestran los resultados obtenidos de cada una de las actividades:

1. Ejercicio LED 1

```
1.  from gpiozero import LED
2.  from time import sleep
3.
4.  red = LED(17)
5.  while True:
6.    red.on()
7.    sleep(1)
8.    red.off()
9.    sleep(1)
10.
```

Se implementó el primer código el cual utiliza el LED con GPIO17. Realizará el encendido y apagado del LED con un retraso provocado por suspender el programa a partir de la función sleep.

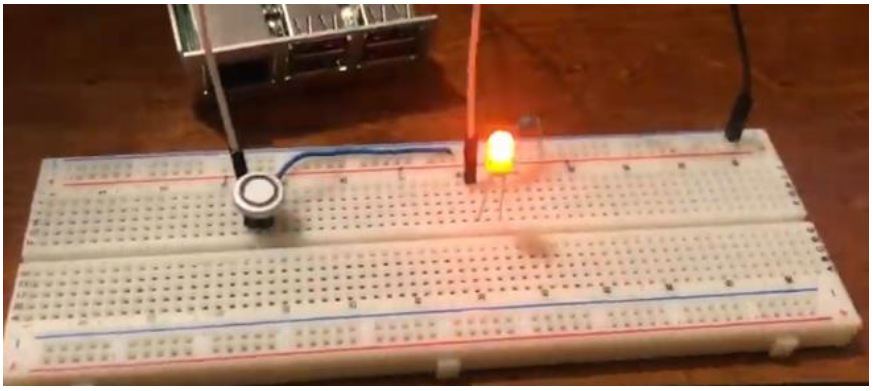


Circuito 1: Conexión del LED con la Raspberry Pi (GPIO17) y encendido y apagado del LED

2. Ejercicio LED 2

```
1.  from gpiozero import LED
2.  from signal import pause
3.  red = LED(17)
4.  red.blink()
5.  pause()
```

El código 2 generó la conexión con la GPIO 17. El resultado realizará la misma acción que el ejercicio anterior, el LED parpadeará y al final cause una pausa al final.

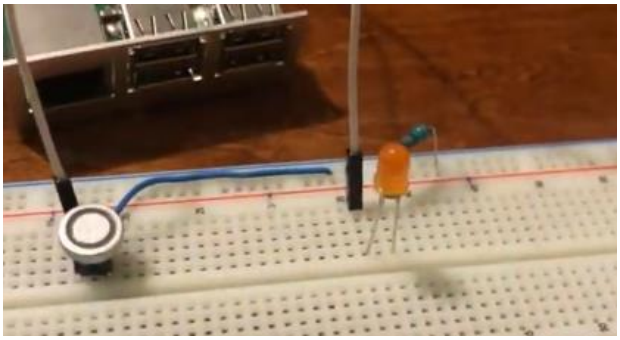


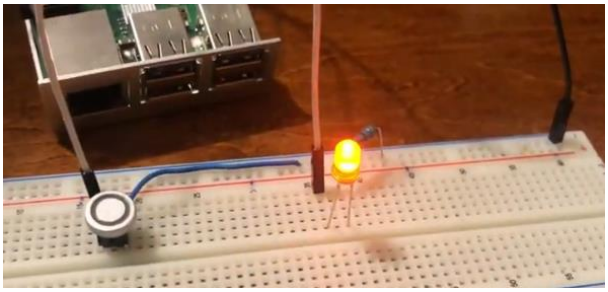
Circuito 2: Conexíon y parpadeo del LED

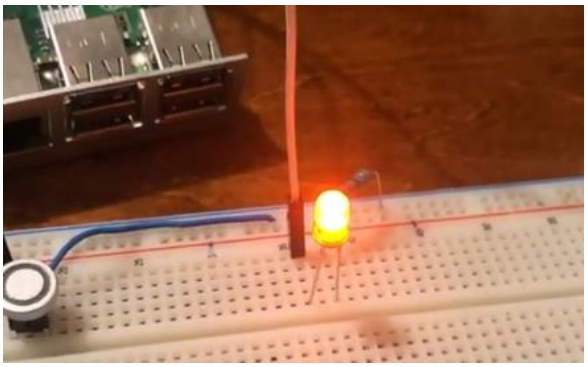
3. Ejercicio LED 3

```
1.  from gpiozero import PWMLED
2.  from time import sleep
3.  led = PWMLED(17)
4.  while True:
5.    led.value = 0
6.    sleep(1)
7.    led.value = 0.5
8.    sleep(1)
9.    led.value = 1
10.   sleep(1)
```

En el ejercicio 3 genera un encendido parcial con respecto al LED, un encendido parcial junto con una pausa entre los cambios.





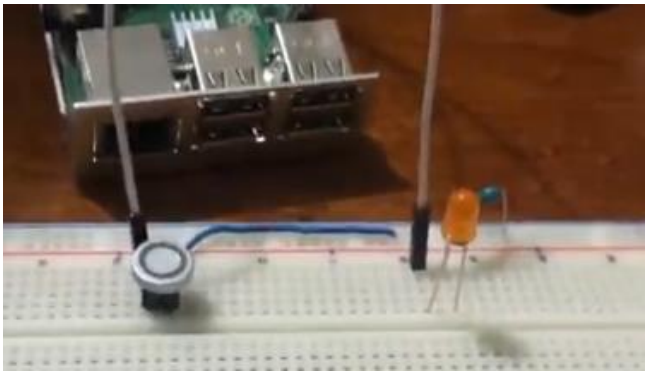


Circuito 3: Apagado, encendido parcial y total de LED

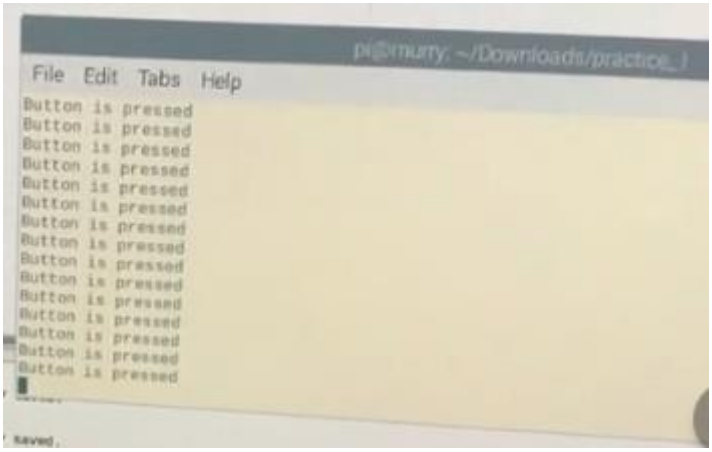
4. Ejercicio Push Button 1

```
1.  from gpiozero import Button
2.  button = Button(2)
3.  while True:
4.    if button.is_pressed:
5.              print("Button is pressed")
6.    else:
7.              print("Button is not pressed")
```

En este ejercicio se realizó el manejo de las entradas, esto a partir de un botón y tener respuesta en la tarjeta. El GPIO utilizado para el botón es el GPIO 2. Dependiendo de la acción se le tomará la respuesta en la consola.



Circuito 4: Conexión con el Boton la Raspberry con el GPIO2

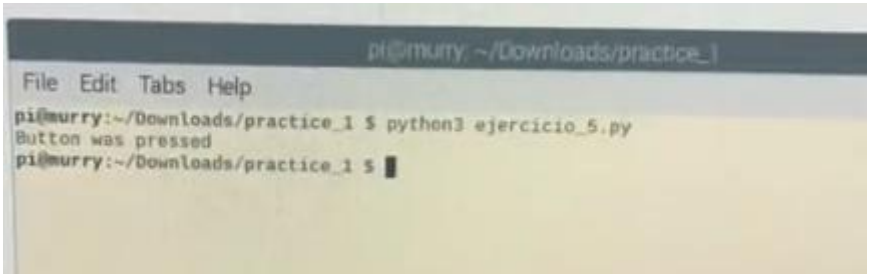


Circuito 5: Mensaje del estado del botón en consola.

5. Ejercicio Push Button 2

```
1.  from gpiozero import Button
2.  button = Button(2)
3.  button.wait_for_press()
4.  print("Button was pressed")
```

A partir de la función wait_for_press() el programa se quedará en pausa hasta que el botón asignado sea oprimido.



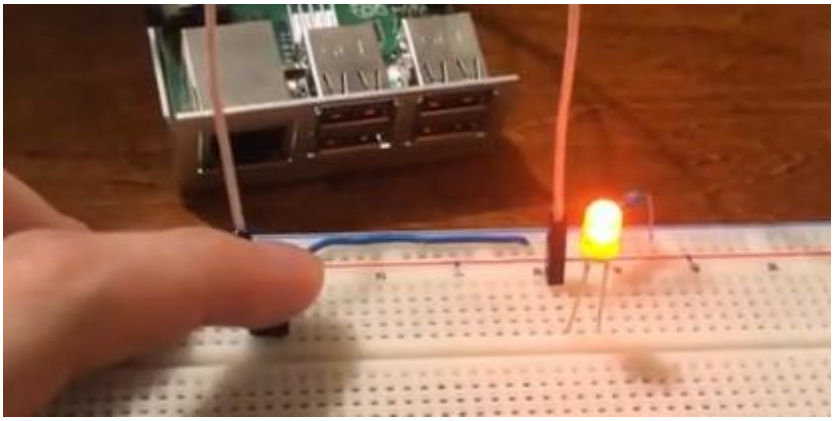
Circuito 6: Continuación del código a partir de oprimir el botón

6. Ejercicio Push Button y LED 1

```
1.  from gpiozero import LED, Button
```

```
2.  from signal import pause
3.  led = LED(17)
4.  button = Button(2)
5.  button.when_pressed = led.on
6.  button.when_released = led.off
7.  pause()
```

En este ejercicio se manejaron ambos elementos, LED y el Botón. A partir de apretar el boton el led encenderá, y al soltarlo el LED se apagará.

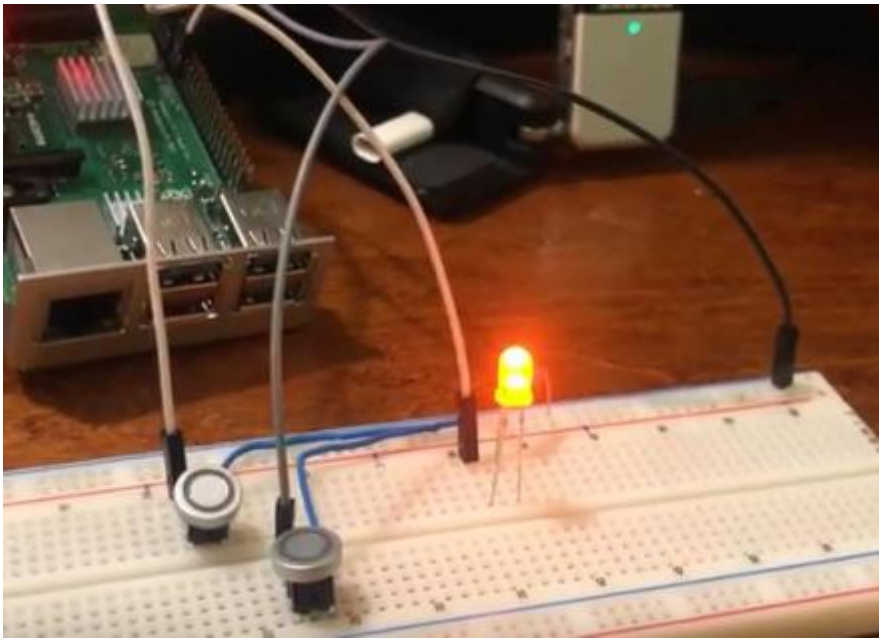


Circuito 7: Encendido del LED a partir de oprimir el botón

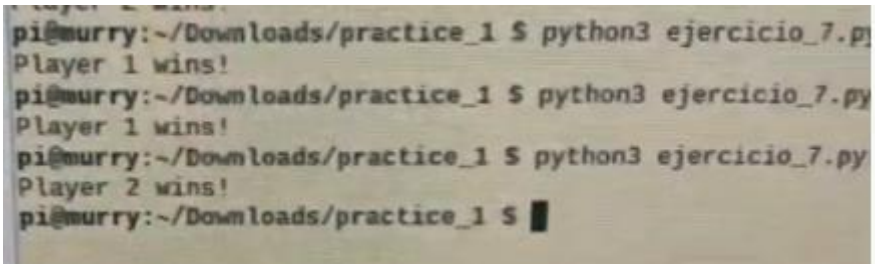
7. Ejercicio Push Button y LED 1

```
1.  from gpiozero import Button, LED
2.  from time import sleep
3.  import random
4.  led = LED(17)
5.  player_1 = Button(2)
6.  player_2 = Button(3)
7.  time = random.uniform(5, 10)
8.  sleep(time)
9.  led.on()
10. while True:
11.   if player_1.is_pressed:
12.             print("Player 1 wins!")
13.             break
14.   if player_2.is_pressed:
15.             print("Player 2 wins!")
16.             break
17. led.off()
```

El manejo del ejercicio utiliza 2 botones (conexión con el GPIO 2 y GPIO 3). A partir de encender el LED después de una suspensión aleatoria se debe de oprimir un botón. Dependiendo que botón sea oprimido, la respuesta se mostrará en consola.



Circuito 8: Conexión con el LED y los 2 botones con la Raspberry



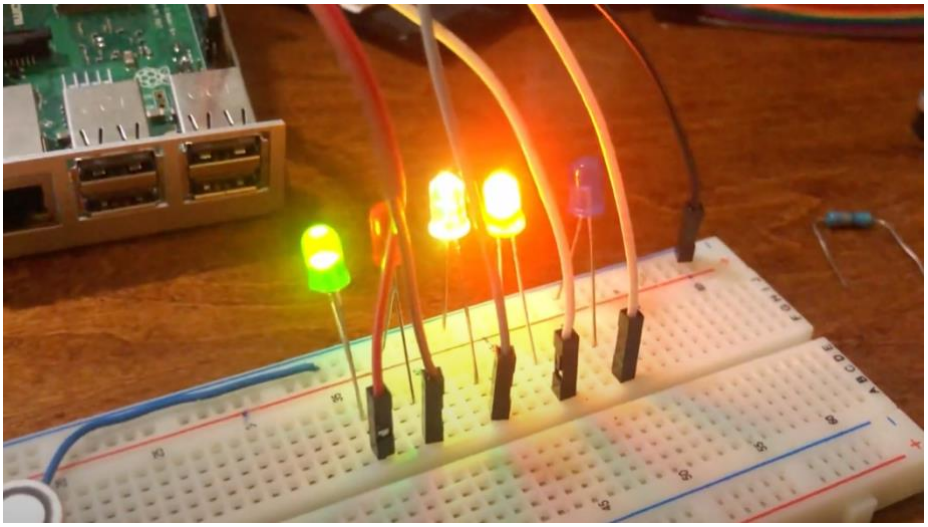
Circuito 9: Respuestas obtenidas al ejecutar y oprimir los 2 botones en varios casos

8. Ejercicio Barra de leds

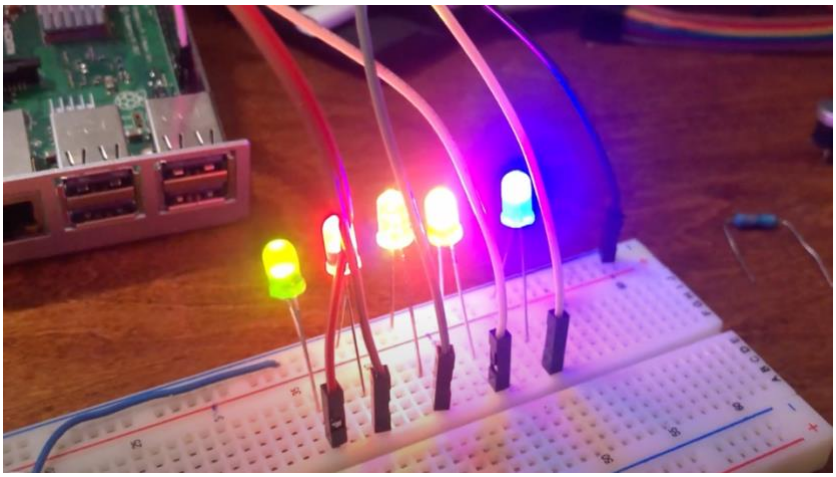
```
1.   from gpiozero import LEDBoard
2.  from time import sleep
3.  from signal import pause
4.  leds = LEDBoard(5, 6, 13, 19, 26)
5.  leds.on()
6.  sleep(1)
7.  leds.off()
8.  sleep(1)
9.  leds.value = (1, 0, 1, 0, 1)
10. sleep(1)
11. leds.blink()
12. pause()
```

Mediante el uso del objeto LEDboard es como pudimos emplear un arreglo de leds e instanciarlos en cada uno de los puertos de nuestra Raspberry (Solo los puertos declarados).

A través de los métodos off y blink es como este arreglo de leds pudimos darles una serie de encendido y apagado que resultara en algo visualmente llamativo, a continuación se muestran los resultados obtenidos:



Circuito 9: Primer encendido de leds antes del primer apagado



Circuito 10 Encendido de leds tras el primer apagado

2. Ejercicios (Semáforos)

1. Programa para semáforo

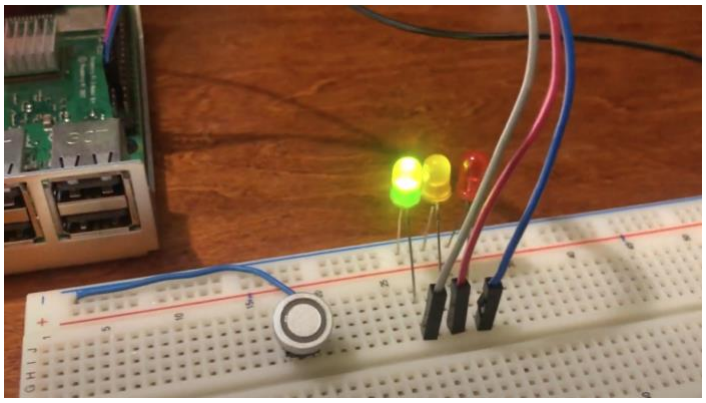
Para el desarrollo del semáforo nos basamos principalmente en el ejercicio del board-led y del push-botton , donde a través de dos ciclos while es como vamos iterando cada uno de los estados de un semáforo, para este caso en concreto solamente son 3 estados, verde(1,0,0) , naranja (0.1.0) y rojo (0,0,1).

Sin embargo, para dar un efecto más realista agregamos un efecto de parpadeo o “blink”, realizando una transición de naranja a todos apagados para dar el efecto de que se cambiará próximamente a rojo.

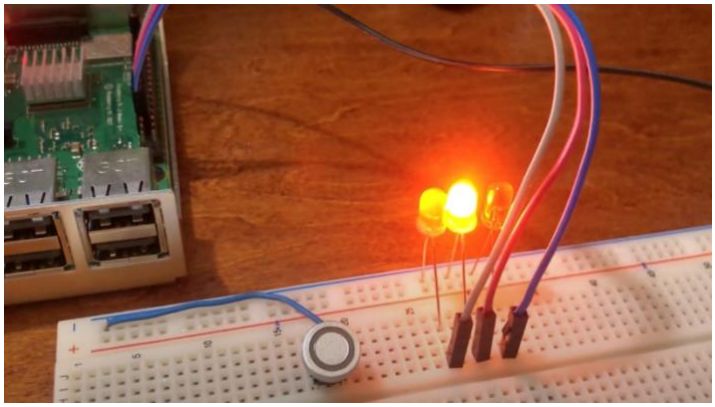
A continuación se muestra nuestra solución realizada en Python:

```
1.  from gpiozero import LEDBoard
2.  from time import sleep
3.  leds = LEDBoard(5,6,13)
4.  while True:
5.      leds.value = (1,0,0)
6.      sleep(5)
7.      x=5
8.      while (x > 0):
9.          leds.value = (0,1,0)
10.         sleep(1)
11.         leds.value = (0,0,0)
12.         sleep(1)
13.         x = x - 1
14.     leds.value = (0,0,1)
15.     sleep(5)
16.     x=5
17.     while (x > 0):
18.         leds.value = (0,0,1)
19.         sleep(1)
20.         leds.value = (0,0,1)
21.         sleep(1)
22.         x = x - 1
```

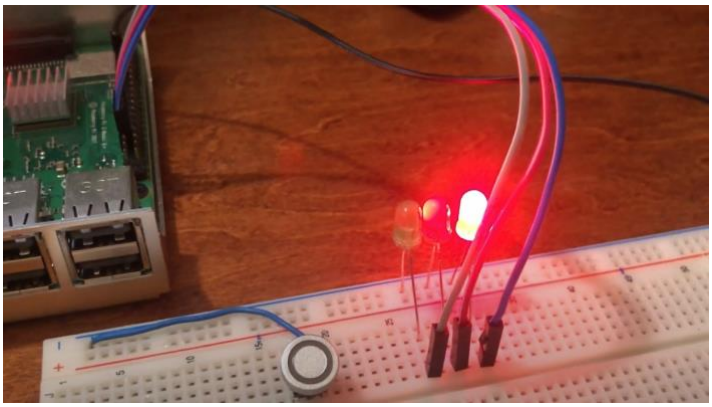
A continuación, se muestran los resultados del semáforo mediante los leds:



Circuito 11 : Se observa el primer estado donde se muestra el avanzar a los autos mediante un led verde



Circuito 12: Se observa el cambio al segundo estado, donde es una advertencia a los autos (led de color naranja)



Circuito 13: Se observa el cambio al último estado, donde es el parar a los autos (Led color rojo)

2. Programa para semáforo con interrupción

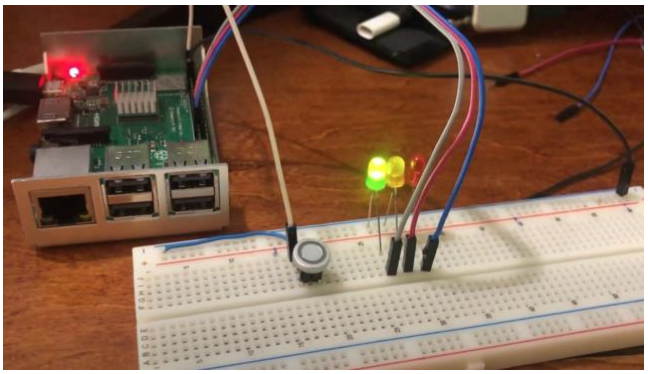
Para el desarrollo de este ejercicio se empleó directamente el anterior ejercicio con la diferencia de que se empleó un objeto de tipo Button con el cual se condicionó cada uno de los ciclos while para de estar forma siempre tener contemplado el estado del botón y así poder realizar una interrupción que nos llevará directamente a un estado en el que solo se tuviera el led rojo encendido (El estado en el que se apran los autos para dejar pasar a las personas).

A continuación se muestra nuestra solución realizada en Python:

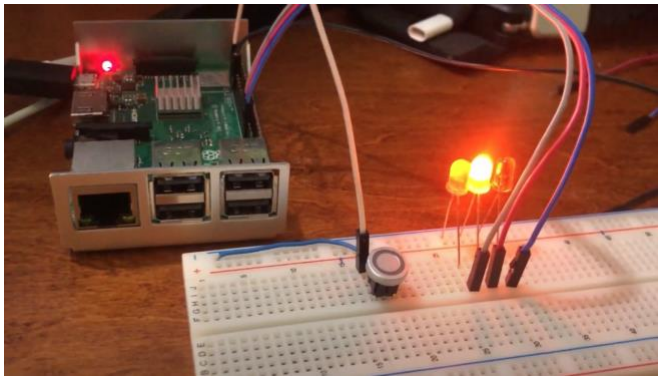
```
1.  leds = LEDBoard(5,6,13)
2.  boton = Button(2)
3.  n=0
4.  while True:
5.      while n==0 and boton.is_pressed == 0:
6.          leds.value = (1,0,0)
7.          sleep(5)
8.          x=5
9.          while (x > 0 and boton.is_pressed == 0):
10.             leds.value = (0,1,0)
11.             sleep(1)
12.             leds.value = (0,0,0)
13.             sleep(1)
14.             x = x - 1
15.         leds.value = (0,0,1)
16.         sleep(5)
```

```
        x=5
        while (x > 0 and boton.is_pressed == 0):
            leds.value = (0,0,1)
            sleep(1)
            leds.value = (0,0,1)
            sleep(1)
            x = x - 1
    else:
        print("interruption")
        leds.value = (0,0,1)
        sleep(5)
```

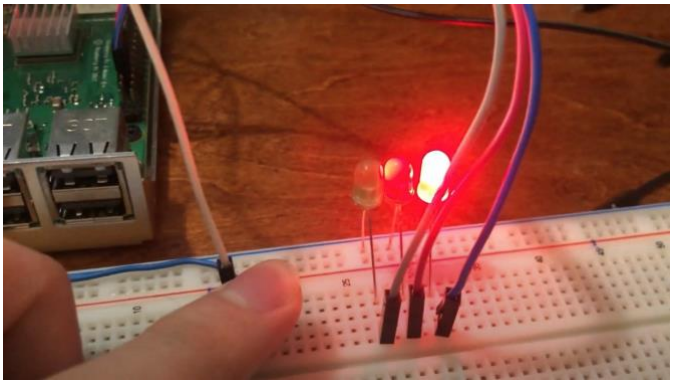
Por otro lado, a continuación se muestran los resultados del semáforo mediante los leds:



Circuito 14: Se observa el primer estado donde se muestra el avanzar a los autos mediante un led verde



Circuito 15: Se observa el cambio al segundo estado, donde es una advertencia a los autos (led de color naranja)



Circuito 16: Se observa el cambio al tercer estado tras oprimir el botón de cambió de estado en el semáforo

NOTA: Para poder contemplar el efecto de paro o cambio de estado mediante el botón , se sugiere revisar el vídeo de la práctica.

Conclusiones:

Cárdenas Cárdenas Jorge

Esta práctica nos permitió comprender el funcionamiento básico de los puertos GPIO de entrada y salida de una Raspberry Pi y la forma de interactuar con ellos desde un lenguaje de alto nivel como lo es Python, el cual mediante el módulo gpiozero nos permite leer o escribir valores a determinados puertos. Comprender como interactuar con los puertos de entrada y salida es algo fundamental al trabajar con una plataforma como lo es Raspberry, ya que la tarjeta por sí sola no serviría de gran cosa si no se le pudieran conectar sensores o dispositivos para interactuar con un ambiente o usuario, como en este caso lo fueron los led y push button.

Cada uno de los ejercicios realizados nos permitieron observar por un lado las conexiones físicas en la tarjeta (los pines) y por otro comprender la forma en que se debe programar, así como las variables y métodos de los que se disponen para poder interactuar con la tarjeta.

Garrido Sánchez Samuel Arturo

En la práctica hemos podido verificar por primera vez cómo es el funcionamiento de los puertos de las Raspberry Pi, utilizando el puerto de salida de voltaje 5V ubicado en la posición 17 de un modelo 4B. Luego, utilizando el lenguaje de programación Python junto con la biblioteca gpio python utilizamos dichos puertos con instrucciones determinadas, como lo es encender leds. Esto nos permite manipular elementos de hardware a través de lenguajes de alto nivel. Finalmente, este mismo proceso lo podemos utilizar en demás proyectos colocando mayores condiciones que se adapten a nuestros requerimientos.

Murrieta Villegas Alfonso

En la presente práctica aprendimos a interactuar con los puertos de entrada y salida de nuestra Raspberry Pi empleando una biblioteca de Python, además, mediante conceptos básicos de dispositivos electrónicos es como resolvimos los diferentes ejercicios que se nos pedía a lo largo de la práctica.

Por otro lado, de manera implícita en algunos ejercicios se nos pedía realizar programas donde se considerarán interrupciones que pudieran llevar a nuestros pequeños sistemas a estados alternativos o emergentes al que estaban ejecutando.

Por último, a su vez nos familiarizamos con los distintos pins de nuestra tarjeta para a futuro facilitarnos el uso de sus distintos puertos

Reza Chavarria Sergio Gabriel

La práctica proporciono la forma en cómo se da la conexión e interacción entre la Raspberry Pi y los elementos externos que utilizaremos en las futuras práctica, esto a partir del manejo de los puertos GPIO.

Además del manejo externo, también se revisó el manejo de las entradas y salidas a partir de la programación en Python. Con esto poder variar las acciones de las salidas a partir del comportamiento de las entradas, en estos casos del led y los push button respectivamente.

Esto es importante debido a que los sistemas embebidos manejarán de manera indispensable la interacción con el ambiente a partir de puertos de entrada y salida.

Valdespino Mendieta Joaquín

En la presente practica pudimos aplicar y comprender el funcionamiento de la comunicación por medio de los puertos de la raspberry Pi, con los elementos de circuitos, como los leds, o los push button que usaremos a lo largo del semestre para realizar diversos aplicativos, estos puerto denominados GPIO, estos puertos en entrada o salida fueron habilitados mediante python, con uso de conceptos de microcomputadoras, como las interrupciones, en este caso aplicados en un lenguaje de alto nivel, es indispensable manejar estos elementos para realizar proyectos mas complejos mas adelante.

Referencias:

1. GPIO ZERO Docs. *gpiozero*. Recuperado el 21 de Septiembre de 2021, de https://gpiozero.readthedocs.io/en/stable/ https://www.chiark.greenend.org.uk/~sgtatham/putty/
2. Raspberry Pi. *Raspberry Pi OS*. Recuperado el 11 de Septtiembre de 2021, de https://www.raspberrypi.org/software/