13-10

Integrantes: Orbe
Chipugsi
Chasi

Fecha: 13 / 10 / 2025

Practica #1 LED / BCM

LED / BOARD

Práctica #1 LED + button / BCM Fecha: 14 / 10 / 2025

LED + button / BOARD

Práctica # 2 Fecha: 07 / 11 / 2025

funciones LED BCM / BOARD

funciones LED + button BCM / BOARD

Fecha: 19/10/2025

Fundamentos de Raspberry Pi y Python

Práctica 1

Anthony Fabricio Chipugsi Pilicita
achipugsip@est.ups.edu.ec
Cinthya Aracelly Orbe Muñoz
corbem@est.ups.edu.ec
Diego Mauricio Chasi Guamushig
dchasig@est.upd.edu.ec

I. Introducción

La Raspberry Pi es una computadora de placa reducida que permite la interacción con dispositivos electrónicos mediante sus pines GPIO (General Purpose Input/Output) [1]. El aprendizaje de su programación básica en Python constituye un primer paso hacia el desarrollo de sistemas embebidos e IoT, como prototipos de casas inteligentes. En esta práctica se buscó instalar Raspberry Pi OS, configurar Python y controlar un LED utilizando los modos de numeración BCM y BOARD, con el fin de comprender las diferencias entre ambos esquemas y reforzar la relación entre software y hardware.

II. Metodología

1. **Instalación y configuración del entorno**

- Se instaló **Raspberry Pi OS** en la tarjeta microSD [1].
- Se habilitó el acceso remoto mediante **SSH** utilizando PuTTY [2].
- Se actualizó el sistema con:

sudo apt update && sudo apt upgrade

- Se instaló la librería de control de pines GPIO:

sudo apt install python3-rpi.gpio

2. **Conexión del circuito**

- Se conectó un LED al pin **GPIO 18 (BCM)** y a tierra (GND) mediante una resistencia de 220 Ω.
- Se utilizó un protoboard y cables jumper para la conexión.

3. **Programación en Python**

- Se implementó un código para encender el LED en modo **BCM**.

```
 GNU nano 8.4
import RPi.GPIO as GPIO
import time

aGPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(18, GPIO.OUT)

while True:
        GPIO.output(18,True)
        time.sleep(1)
        GPIO.output(18, False)
        time.sleep(1)
```

Fig. 1. Código con puerto BCM

- Se modificó el código para modo **BOARD**, conectando el LED al pin físico correspondiente.

```
 GNU nano 8.4
import RPi.GPIO as GPIO
import time

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(12, GPIO.OUT)

while True:
        GPIO.output(12,True)
        time.sleep(1)
        GPIO.output(12, False)
        time.sleep(1)
```

Fig. 2. Código con puerto BOARD

III. Resultados

- El LED se encendió correctamente durante 5 segundos en ambos modos de numeración.

- Se comprobó que la diferencia entre BCM y BOARD radica únicamente en la forma de

identificar los pines, no en el funcionamiento eléctrico.

- El entorno remoto mediante SSH permitió programar y ejecutar los códigos sin necesidad de periféricos adicionales [2].

Tabla 1. Comparativa entre BCM y BOARD:

Característica	BCM	BOARD
Ejemplo usado	GPIO 18	Pin físico 12
Ventaja	Más preciso para programación avanzada	Más intuitivo para conexiones físicas
Resultado	LED encendido 5 s	LED encendido 5 s

IV. Discusión

La práctica permitió comprender los fundamentos de la programación en Raspberry Pi y la interacción con componentes electrónicos básicos [1]. El uso de Python y la librería RPi.GPIO facilitó el control del LED, mostrando la versatilidad de la plataforma para proyectos de IoT. La comparación entre BCM y BOARD evidenció que ambos esquemas son funcionales, pero requieren atención al identificar los pines. Además, el acceso remoto mediante SSH optimiza el trabajo en laboratorio [2], ya que elimina la necesidad de periféricos adicionales y permite un flujo de programación más eficiente.

V. Conclusiones

- La Raspberry Pi es una plataforma versátil para el aprendizaje de sistemas embebidos e IoT [1].
- La librería RPi.GPIO facilita el control de dispositivos físicos desde Python.
- La práctica permitió comprender las diferencias entre los modos BCM y BOARD, reforzando la importancia de identificar correctamente los pines antes de programar.
- El acceso remoto mediante SSH optimiza el trabajo en laboratorio [2], permitiendo programar sin necesidad de monitor ni teclado.

VI. Referencias

[1] Raspberry Pi Foundation, “Raspberry Pi Documentation,” Raspberry Pi, 2025. [Online]. Available: https://www.raspberrypi.com/documentation

[2] PuTTY, “PuTTY: a free SSH and Telnet client,” PuTTY Documentation, 2025. [Online]. Available: https://www.putty.org

VII. Anexos

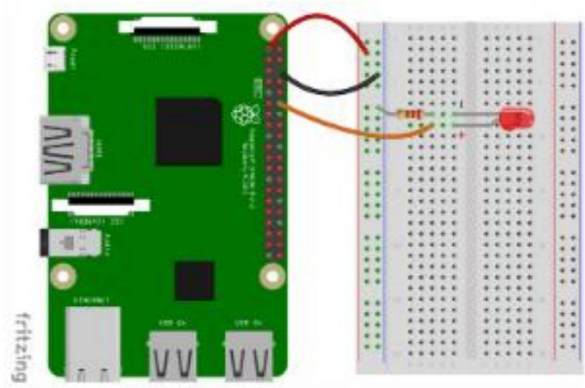

Fig. 3. Diagrama de conexión

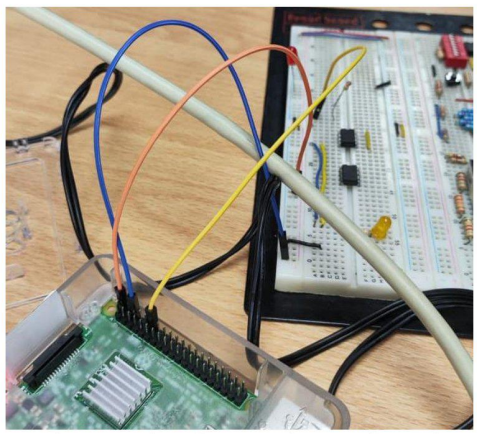

Fig. 4. Circuito en el protoboard