

# Laboratorio 13

## Competencias a desarrollar

Practicar el manejo de los puertos GPIO de la Raspberry Pi como entradas y salidas, utilizando conexiones a leds y botones y diseñando un programa que controle dichas entradas y salidas.

## Materiales

- Raspberry Pi, cargador y tarjeta SD
- 1 protoboard
- 3 leds
- 3 resistencias de 270Ω
- 1 push-button (botón)
- 1 resistencia del 10k Ω
- Jumpers macho-hembra
- Jumpers macho-macho

## Circuitos Básicos

### Continuidad en un *Protoboard*

El *protoboard* es una herramienta que permite armar y modificar circuitos rápida y prácticamente. Esta herramienta se encuentra en diversos tamaños y consiste de una matriz de agujeros cuyas filas tienen continuidad entre sí tal como lo muestra la figura 1. Comúnmente, en un *protoboard* se utilizan resistencias, capacitores, leds, botones, circuitos integrados, entre otros para armar circuitos más complejos. Una de las mayores ventajas de los *protoboards* es que su estructura de agujeros permite probar circuitos sin necesidad de que éste sea permanente, pues se pueden conectar muchos dispositivos e intercambiarlos fácilmente.

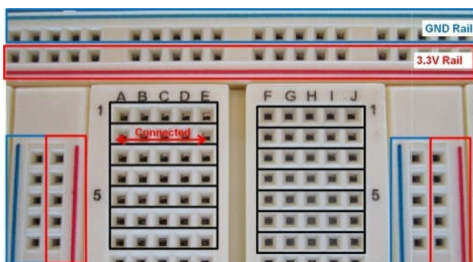


Figura 1: Continuidad en un protoboard. Fuente: <http://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/turing-machine/two.html>

### Conexión de un LED en Protoboard

Un LED (Light Emitting Diode por sus siglas en inglés), es un dispositivo electrónico que emite luz cuando se conecta una fuente de voltaje en sus terminales. Es importante notar que el led, por ser un diodo, debe conectarse en la orientación adecuada para que funcione. La figura 2 muestra la forma correcta de conectar un diodo led.

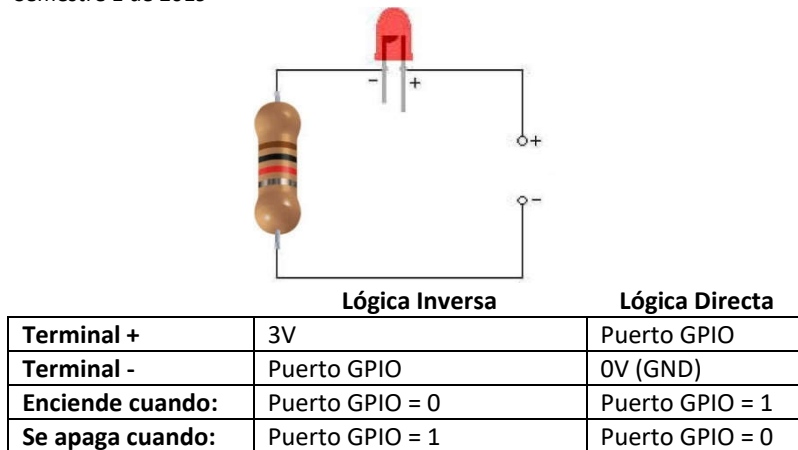


Figura 2: Conexión de un LED. Fuente: <http://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/turing-machine/two.html>

## Conexión de un Switch

Al trabajar con circuitos, existen varios tipos de interruptores que se pueden utilizar como entradas al sistema. Uno de los interruptores más utilizados por su simpleza y funcionalidad es el conocido “push-button”. Este interruptor consta de por lo menos 2 pines que están aislados normalmente y al presionar el botón los pines se conectan. Existen dos configuraciones para conectar el botón, el **“pull-down” implementa la lógica directa**, el **“pull-up” implementa la lógica inversa**. Las conexiones adecuadas del botón se muestran en la figura 3. Es muy importante verificar que el botón está conectado correctamente para evitar accidentes.



Figura 3: Conexiones de un push button en configuraciones pull down (izquierda) y pull up (derecha).

## Instrucciones

En parejas asignadas en Canvas realice lo siguiente en lenguaje ensamblador ARM:




Realizar el corrimiento cíclico del bit activo (de izquierda hacia derecha), en una serie de 3 leds, con una pulsación inicial de un *push-button* conectado a un puerto GPIO configurado como una entrada. Al iniciar el programa los 3 leds están apagados y al pulsar el *push-button* se habilita el corrimiento del led encendido en un ciclo, al pulsar una segunda vez, solamente al mismo led, se detiene el ciclo y se apagan todos los leds.

1. En un *protoboard* conecte los leds con la resistencia de 270Ω como se indica en la figura 2 con la lógica que más le convenga. También conecte el *push-button* con la resistencia de 10kΩ como lo indica la figura 3 e implemente la lógica que le quede mejor.
2. Escoja 4 puertos GPIO y conecte los 3 leds y el *push-button* utilizando *jumpers* macho-hembra.
3. Diseñe la subrutina **GetGpio** sugerida en el material de la semana 16 en un archivo independiente al main, para leer el estado de un puerto de entrada.
4. Diseñe el programa main.s que configure los puertos GPIO de los leds y *push-button* utilizando la biblioteca GPIO acorde con su Raspberry y que realice el corrimiento cíclico del bit activo en los 3 leds, manejado por el botón como se indica al inicio.

## Material a entregar

1. Todos los archivos .s
2. Enlace a video donde se observe a los dos integrantes del grupo explicando el funcionamiento de su programa en la Raspberry.

## Evaluación

Criterios de evaluación	Nivel 3 Experto 	Nivel 2 Aprendiz 	Nivel 1 Novato 
<b>Ejercicios previos al laboratorio</b> <b>15%</b>	El estudiante presentó el ejercicio previo de conexión del GPIO como puertos de entrada y salida, demostrando su funcionamiento. <b>15%</b>	No aplica	No aplica
<b>Funcionamiento del Programa</b> <b>45%</b>	El programa funciona con todos sus requerimientos: funcionamiento de los puertos GPIO de entrada y salida. <b>45%</b>	El programa funciona al menos en el 80% de sus requerimientos. <b>35%</b>	El programa funciona en menos del 50% de sus requerimientos. <b>10%</b>
<b>Uso de subrutinas e implementación de GetGpio</b> <b>20%</b>	El programa está estructurado con al menos una subrutina(s) <b>-GetGpio-</b> que usa el <b>estándar ABI</b> y que hace que el código sea más eficiente, comprensible y ordenado. La subrutina está en un archivo diferente al <i>main</i> . <b>20%</b>	El programa está estructurado con al menos una subrutina(s) <b>-GetGpio-</b> pero NO USA el <b>estándar ABI</b> y/o la subrutina está en el mismo archivo del <i>main</i> . <b>10%</b>	No se implementó la subrutina GetGpio. <b>0%</b>
<b>Orden y cableado del Circuito (grupal)</b> <b>5%</b>	El orden del circuito permite la fácil identificación de las rutas de datos; todas las conexiones se están completamente ordenadas. <b>5%</b>	El 70% del circuito está ordenado y se identifican conexiones hacia los leds y el botón. <b>3%</b>	El circuito está completamente desordenado y no se identifican las rutas de datos. <b>0%</b>
<b>Documentación y orden del programa</b> <b>5%</b>	La documentación incluye encabezado y comentarios representativos en los bloques de código más importantes. Los nombres de las variables son significativos. La presentación del programa es muy clara y ordenada, y utiliza una tabulación adecuada. <b>5%</b>	La documentación incluye encabezado, pero no hay comentarios representativos en los bloques de código más importantes. Algunos nombres de las variables son significativos. La presentación del programa es aceptable, y utiliza una tabulación más o menos adecuada. <b>4%</b>	Falta más del 50% de la documentación del código Los nombres de las variables no expresan ningún significado. La presentación del programa es confusa y desordenada. No hay tabulación de las instrucciones. <b>0%</b>
<b>Diario reflexivo</b> <b>10%</b>	El estudiante realizó una reflexión completa con aportes significativos para mejorar su aprendizaje en/antes de la fecha solicitada y contestó todas las preguntas. <b>10%</b>	El estudiante realizó una reflexión con aportes significativos para mejorar su aprendizaje en/antes de la fecha solicitada y contestó el 80% de las preguntas. <b>8%</b>	El estudiante realizó una reflexión con muy pocos aportes significativos para mejorar su aprendizaje en/antes de la fecha solicitada y contestó menos del 50% de las preguntas. <b>2%</b>

