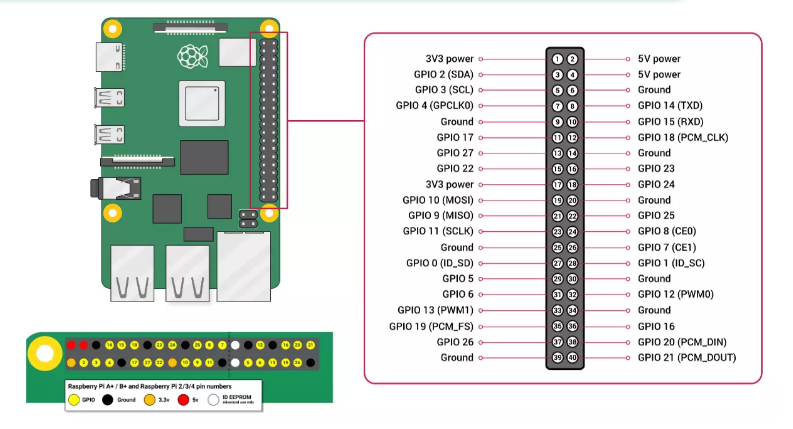
**Explicación sobre GPIO**

Esta conexión GPIO (General Purpose Input Output) es un conjunto de pines que varían en número según la versión de la Raspberry Pi. De todos modos, la forma de organizarlos siempre es la misma y se distribuyen en dos filas. Cada uno de estos pines tienen una función específica o son de propósito general.



**Módulo control de salida GPIO**

En esta ocasión veremos cómo se activan la salida GPIO de las Raspberry para después sustituirlos por actuadores para el encendido y apagado de accesorios en la casa como lo son la televisión, iluminación, prendido de ventiladores, apertura de puertas entre otros.

En este programa haremos parpadear 4 LEDs con la Raspberry Pi; usando el lenguaje de programación Python, y los puertos GPIO. Con “parpadear” un LED, esperar un tiempo, apagarlo, y repetirlo hasta el infinito.

Estos módulos se unirán con los dos anteriores (reconocimiento de voz, reproducción de sonido) en un solo programa para el control de actuadores por medio de la muñeca lele muñeca inteligente

**Control de puertos GPIO de Raspberry**

**Requerimientos:**

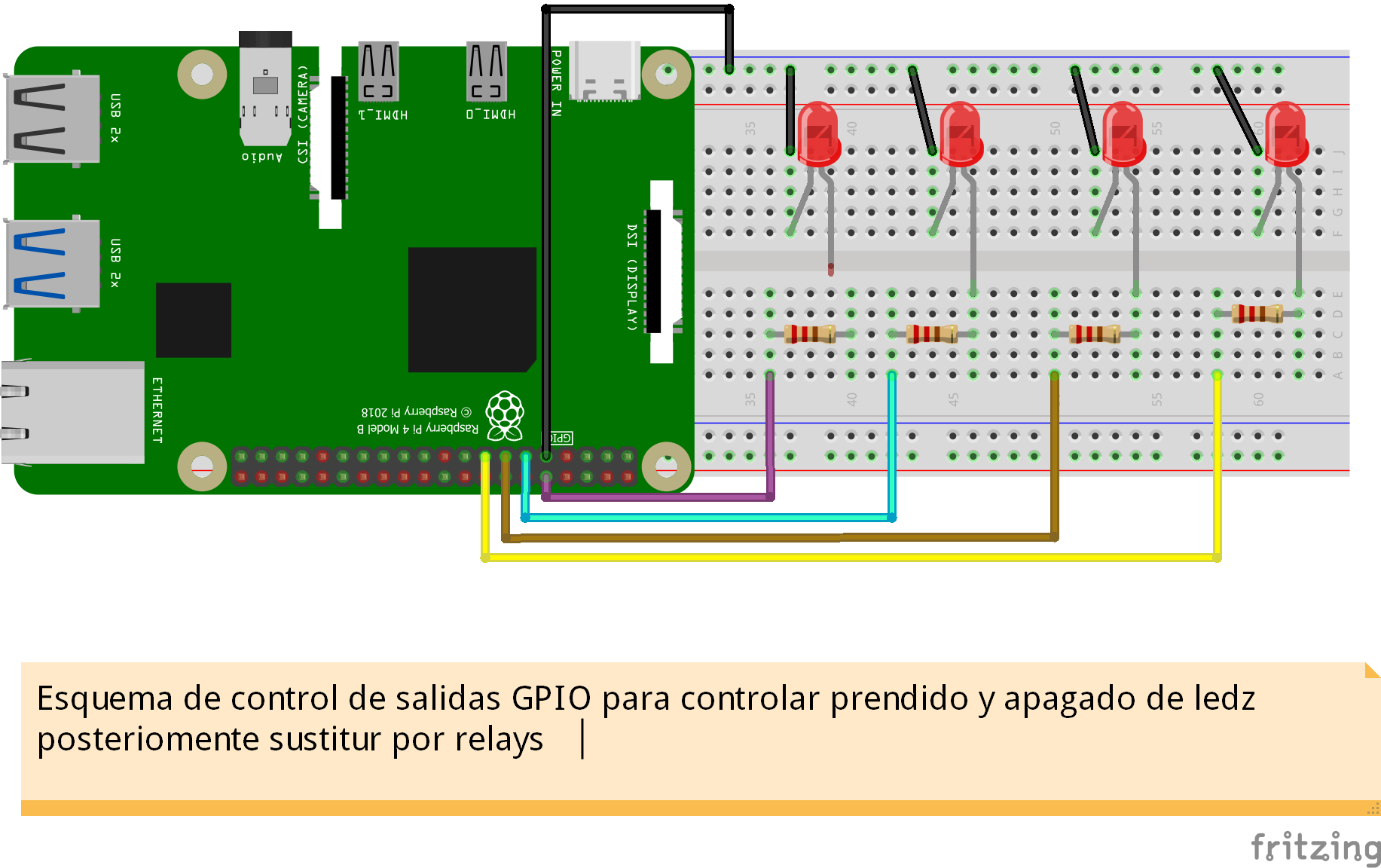
* Raspberry pi 4
* 4 resistencias 220 ohm o de 47 ohm
* 4 leds

**Librerías a utilizar**

1. RPi.GPIO

Esta librería RPi. GPIO ya se encuentran instaladas en el sistema operativo

**Esquema grafico**

****

**Descripción de la conexión**

1. El pin de salida Ground se conecta en el GPIO 6
2. El primer led se conecta en el GPIO 10
3. El segundo led se conecta en el GPIO 11
4. El tercer led se conecta en el GPIO 13
5. El cuarto led se conecta en el GPIO 15
6. Las resistencias se conectan en la pata positivo del y el puerto GPIO correspondiente al led

**A continuación, el código:**

Python

*1 import RPi.GPIO as GPIO*

*2 import time*

*3 ESPERA = 0.5*

*4 rel1 = 10*

*5 rel2 = 11*

*6 rel3 = 13*

*7 rel4 = 15*

*8 GPIO.setmode(GPIO.BOARD)*

*9 GPIO.setup(rel1, GPIO.OUT)*

*10 GPIO.setup(rel2, GPIO.OUT)*

*11 GPIO.setup(rel3, GPIO.OUT)*

*12 GPIO.setup(rel4, GPIO.OUT)*

*13 while True:*

*14 GPIO.output(rel1, GPIO.HIGH)*

*15 GPIO.output(rel2, GPIO.HIGH)*

*16 GPIO.output(rel3, GPIO.HIGH)*

*17 GPIO.output(rel4, GPIO.HIGH)*

*18 time.sleep(ESPERA)*

*19 GPIO.output(rel1, GPIO.LOW)*

*20 GPIO.output(rel2, GPIO.LOW)*

*21 GPIO.output(rel3, GPIO.LOW)*

*22 GPIO.output(re4, GPIO.LOW)*

*23 time.sleep(ESPERA)*

Explicación:

Línea 1: Realizamos el import de *RPi.GPIO*  como GPIO

Línea 2: importamos la librería time para manejar los tiempos

Línea 3: Asignamos un tiempo de espera de medio segundo

Línea 4 a 7: Asignamos los pines GPIO que utilizaremos

Línea 8: Asignamos un modo de uso a la tarjeta

Línea 9 a 12: Asignamos los pines como salida

Línea 13: creamos un siclo infinito

Línea 14 a17: mandamos un valor al pin y con este voltaje se enciende el led

Línea 13: esperamos medio segundo

Línea 19 a 22: quitamos el valor al pin, sin voltaje se apaga el led

Línea 23: esperamos medio segundo

Consideraciones:

Se recomienda hacer las conexiones necesarias sin tener la raspberry encendida para evitar un corto circuito y dañar la tarjeta Raspberry

Se pueden utilizar resistencias con otro valor, para este caso solo se requieren para no quemar el led

Este código fuente es un extracto de la página de Publicado por parzibyte en febrero 9, 2020