

Samsung-Innovation-Campus-OT23 /
C04_A01[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#) [Set](#)

0 stars 0 forks 0 watching Activity

Private template repository

main



Branches Tags



mrodriguez257 Update README.md ...

5 days ago 3

[View code](#)

Capitulo 04 - Raspberry Pi con Python

Actividad 01

1. Escribe un programa que lea cuando un botón se presiona y escriba en un archivo de texto la fecha y hora de cuando se presionó.
2. Escribe un programa que encienda una serie de LEDs, al leer la petición de un usuario escrita en un archivo de texto. Por ejemplo, en el archivo de texto deberá de aparecer 1110, esto indicaría que se deberán de encender los leds 1, 2 y 3 mientras que el 4 no.
3. Escribe un programa que mida la distancia a través de un sensor ultrasónico, encienda uno de tres LEDs cuando se encuentre dentro de un rango de distancia específico y guarde la información en un archivo *.txt con el valor de la distancia con mensaje de alerta y fecha de cuando se leyó. Si la distancia calculada es inferior a 10cm, se enciende el LED rojo y se escribe el siguiente mensaje de alerta: MUY CERCA. Cuando es superior a 30, se enciende el LED verde y se escribe el siguiente mensaje de alerta: EXCELENTE. De lo contrario, se enciende el LED ámbar y se escribe el siguiente mensaje de alerta: CERCA.
4. Escribe un programa donde se lea el uso de memoria ram y la guarde en un archivo *.txt

Para leer la memoria ram en porcentaje usar la siguiente linea.

`psutil.virtual_memory().available * 100 / psutil.virtual_memory().total`

Adicionalmente se deben de prender los 4 leds en este orden:

- Si la ram disponible es menor a 25% : 1000
- Si la ram disponible es menor a 50% : 1100
- Si la ram disponible es menor a 75% : 1110
- Si la ram disponible es menor al 90% : 1111

*No olviden siempre conectar una resistencia en serie a cada LED. *Revisen la conexion correcta del sensor ultrasonico (sección de abajo) *No olviden utilizar el formato para el encabezado en cada archivo generado.

— . .

☰ README.md



- Utilizar los mismos GPIOs para los LEDs de los ejercicios.
- Utilizar la libreria GPIO Zero para los ejercicios.
- Subir cada ejercicio en diferentes ramas de su repositorio (sean consistentes al momento de nombrar las ramas).
- Hacer merge de todas las ramas a la rama principal Main.
- Todos los integrantes deberán hacer al menos un commit por actividad.

Formato para encabezado [↗](#)

"""

Archivo: nombreArchivo.py

Equipo: #

Integrantes del equipo: Integrante #1

Integrante #2

Integrante #3

Fecha: Fecha de creación

Descripcion: Copiar la instrucción del ejercicio.

"""

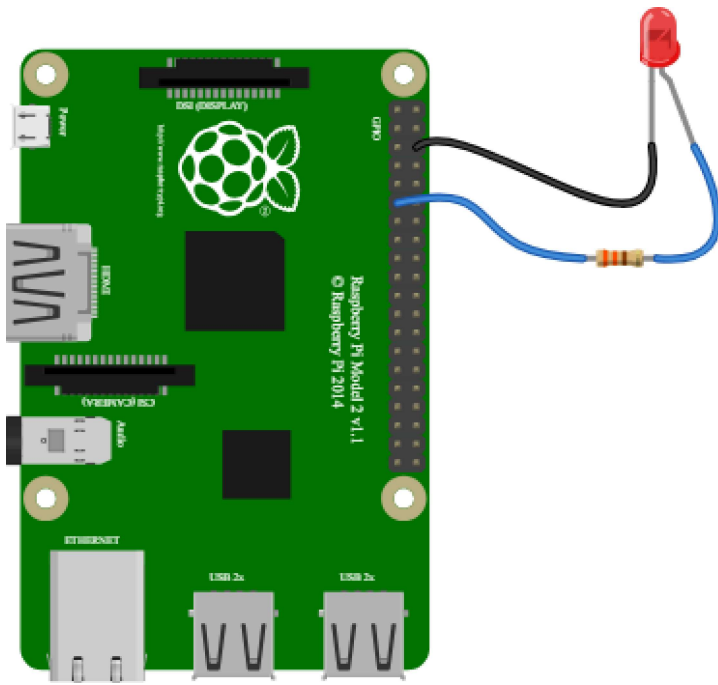


Recursos [↗](#)

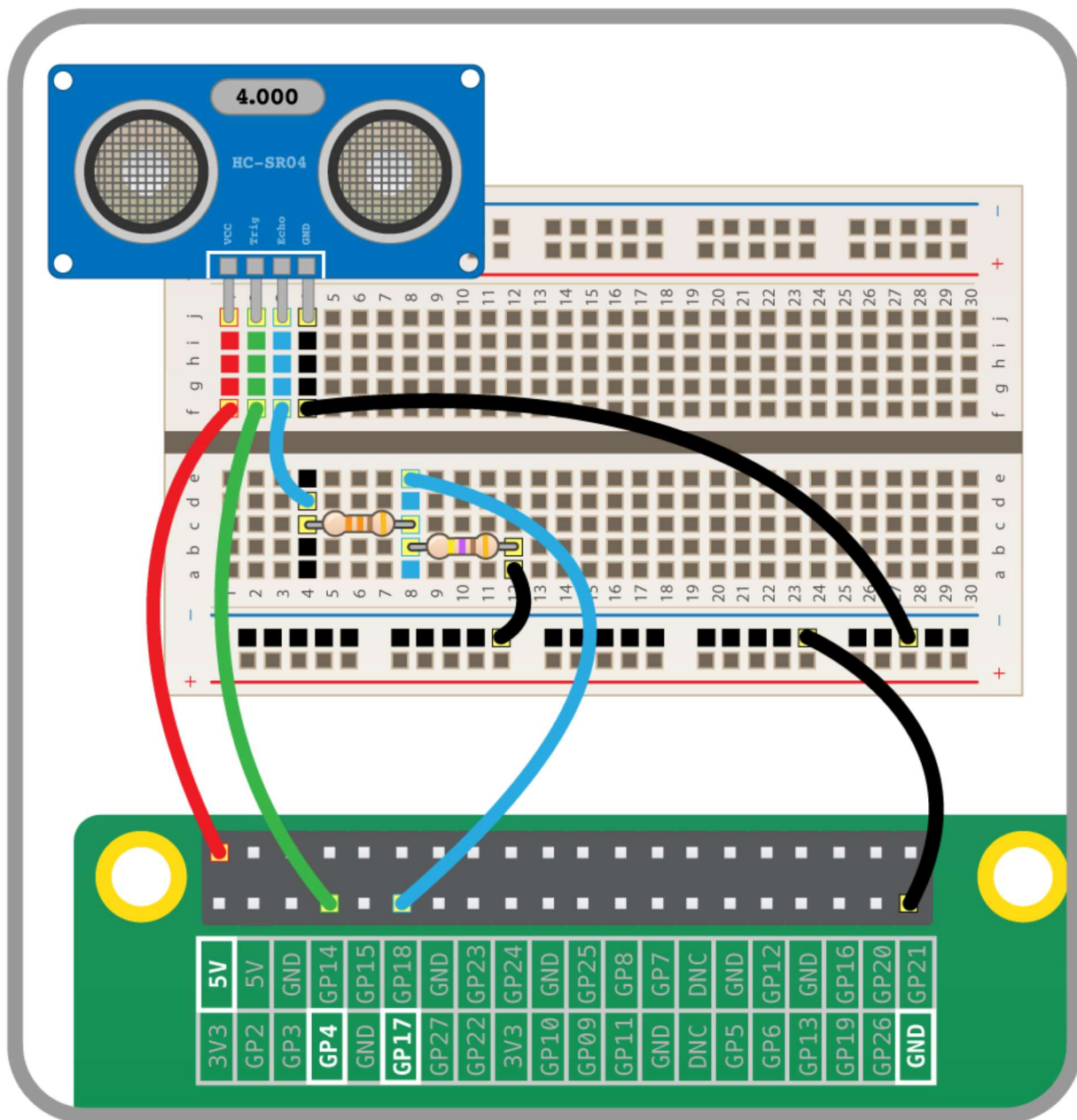
Documentacion GPIO Zero [↗](#)

<https://gpiozero.readthedocs.io/en/latest/index.html>

Ejemplo conexion LED [🔗](#)



Ejemplo conexion Sensor Ultrasonico [🔗](#)

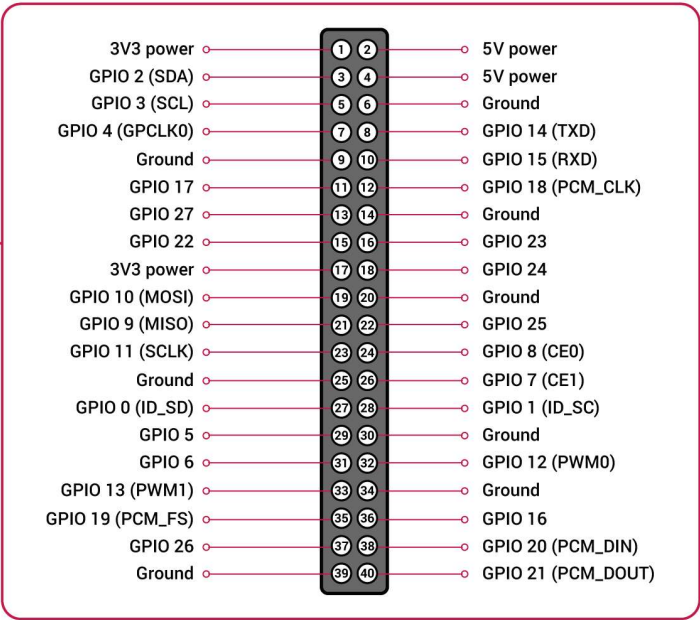
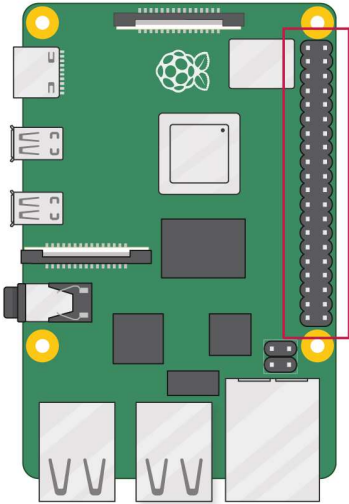


El sensor de distancia requiere dos pines GPIO: uno para el disparo (marcado como TRIG en el sensor) y otro para la respuesta (marcado como ECHO en el sensor). Sin embargo, se requiere un divisor de voltaje para asegurarse de que los 5V del pin ECHO no dañen la Raspberry Pi.

Conecta tu sensor siguiendo las siguientes instrucciones:

- Conecta el pin GND del sensor a un pin de tierra en la Raspberry Pi.
- Conecta el pin TRIG del sensor a un pin GPIO.
- Conecta un extremo de una resistencia de 330 ohmios al pin ECHO del sensor.
- Conecta un extremo de una resistencia de 470 ohmios al pin GND del sensor.
- Conecta los extremos libres de ambas resistencias a otro pin GPIO. Esto forma el divisor de voltaje necesario.
- Finalmente, conecta el pin VCC del sensor a un pin de 5V en la Raspberry Pi.

GPIO [🔗](#)



Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)