

RETOS no IoT

- [Project 1.1 LED Flashing](#)
- [Project 1.2 Breathing LED](#)
- [Project 2.1 Read the Button](#)
- [Project 2.2. Table Lamp](#)
- [Project 3.1 Read the PIR Motion Sensor](#)
- [Project 4.1 Play Happy Birthday](#)
- [Project 5.1 Control the Door](#)
- [Project 5.2 Close the Window](#)
- [Project 6.1 RGB Sk6812](#)
- [Project 7.2 Button Control Fan](#)
- [Project 8.2 Dangerous Gas Alarm](#)
- [Project 9.1 Temperature and Humidity Tester](#)
- [Project 10.1 Open the Door](#)

Project 1.1 LED Flashing

<https://www.youtube.com/embed/EFZorJNhyPI>

SOLUCIÓN CON BLOQUES <https://www.steamakersblocks.com/web/project/2520299>

SOLUCION CON CODIGO

<https://docs.keyestudio.com/projects/KS5009/en/latest/docs/Python/Python.html#project-1-1-led-flashing>

Project 1.2 Breathing LED

En este código se va a utilizar la intensidad PWM para dar la sensación de subida y bajada gradual

<https://www.youtube.com/embed/VzL1NKdFmjc>

- Solución con bloques <https://www.steamakersblocks.com/web/project/2520329>
- Solución con código

<https://docs.keyestudio.com/projects/KS5009/en/latest/docs/Python/Python.html#project-1-2-breathing-led>

¿Qué son las salidas PWM?

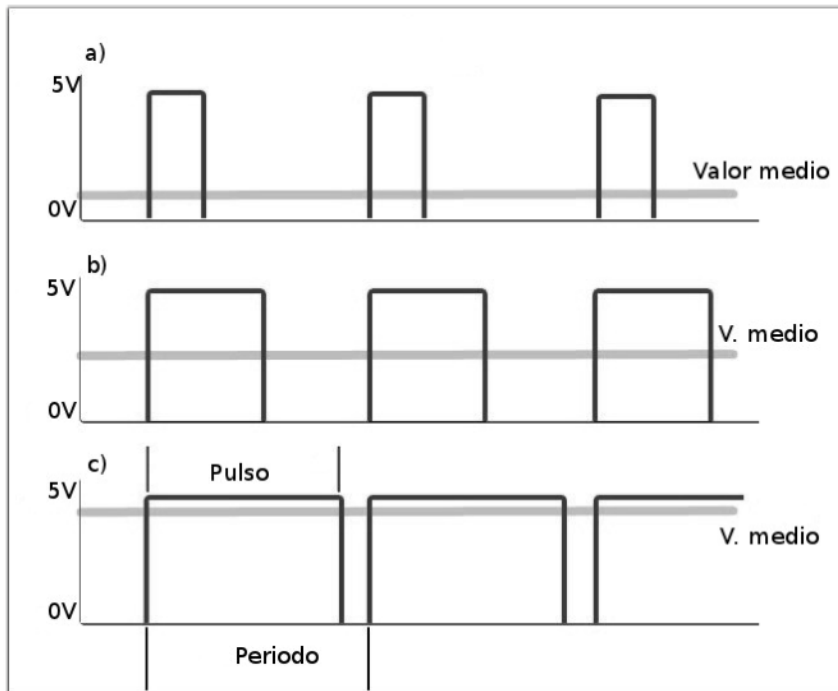
Arduino, ESP32, Micro:bit, PicoW... tienen **entradas** analógicas y digitales. Pero **salidas sólo digitales**.

Para **simular** una salida **analógica** entre 0V y 5V se utilizan señales digitales PWM. En Arduino sólo tiene 6 salidas pseudo-analógicas. En los pines digitales 3, 5, 6, 8, 10 y 11 son PWM

¿Qué es eso de PWM? La señal PWM (*Pulse Width Modulation, Modulación de Ancho de Pulso*) es una señal que utiliza el microcontrolador para generar una señal continua sobre el proceso a controlar. Por ejemplo, la variación de la intensidad luminosa de un led, el control de velocidad de un motor de corriente continua,...

Para que un dispositivo digital, microcontrolador de la placa Arduino, genere una señal continua lo que hace es emitir una señal cuadrada con pulsos de frecuencia constante y tensión de 5V. A continuación, variando la duración activa del pulso (ciclo de trabajo) se obtiene a la salida una señal continua variable desde 0V a 5V.

Veamos gráficamente la señal PWM:



Los pines digitales de la placa Arduino que se utilizan como salida de señal PWM generan una señal cuadrada de frecuencia constante (490Hz), sobre esta señal periódica por programación podemos variar la duración del pulso como vemos en estos 3 casos:

- La duración del pulso es pequeña y la salida va a tener un valor medio de tensión bajo, próximo a 0V.
- La duración del pulso es casi la mitad del período de la señal, por tanto, la salida va a tener un valor medio de tensión próximo a 2,5V.
- La duración del pulso se aproxima al tiempo del período y el valor medio de tensión de salida se aproxima a 5V.

Ejemplo en código ArduinoIDE y Arduino

Para ejecutar una señal PWM, es simplemente **analogWrite(analogOutPin, outputValor)**; donde analogOutPin es el número del Pin PWM, acuérdate que sólo puede ser uno de estos 6 : **3, 5, 6, 8, 10 y 11** y outputValor es el valor de la señal PWM pero **ojo desde 0 a 255** es decir si quieres el valor de 0V tienes que poner 0, si quieres el valor de 5V tienes que poner 255 y si quieres poner un valor medio, haz una regla de tres, por ejemplo 2.5V tienes que poner $255/2=127$ o 128 da igual

Otro ejemplo en Python con Micro:bit

pin16.write_analog(brillo) donde brillo puede ir de 0 a 255



Project 2.1 Read the Button

<https://www.youtube.com/embed/YrgTis942QU>

- SOLUCIÓN POR BLOQUES <https://www.steamakersblocks.com/web/project/2520373>
 - he optado por un retardo de 1 segundo para que se vea bien
- SOLUCIÓN CON CÓDIGO

<https://docs.keyestudio.com/projects/KS5009/en/latest/docs/Python/Python.html#project-2-1-read-the-button>

Project 2.2. Table Lamp

<https://www.youtube.com/embed/y5tsisEHaSQ>

- SOLUCIÓN CON BLOQUES <https://www.steamakersblocks.com/web/project/2520536>

- SOLUCIÓN CON CÓDIGO

<https://docs.keyestudio.com/projects/KS5009/en/latest/docs/Python/Python.html#project-2-2-table-lamp>

Project 3.1 Read the PIR Motion Sensor

<https://www.youtube.com/embed/drgaWBeH39k>

- SOLUCIÓN CON BLOQUES <https://www.steamakersblocks.com/web/project/2520553>
- SOLUCIÓN CON CÓDIGO

<https://docs.keyestudio.com/projects/KS5009/en/latest/docs/Python/Python.html#project-3-1-read-the-pir-motion-sensor>

Project 4.1 Play Happy Birthday

<https://www.youtube.com/embed/5c0SQh66JUk>

https://www.youtube.com/embed/ryl1_8k0qCM

- SOLUCIÓN CON BLOQUES <https://www.steamakersblocks.com/web/project/2521317>
- SOLUCIÓN CON CÓDIGO

<https://docs.keyestudio.com/projects/KS5009/en/latest/docs/Python/Python.html#project-4-1-play-happy-birthday>

Bueno, en la solución con bloques no es happy birthday pero podemos elegir la melodía que queramos



Si queremos crear un sonido RTTTL mirar <https://fgcoca.github.io/ESP32-STEAMakers/A05/>



Project 5.1 Control the Door

<https://www.youtube.com/embed/xW5coiRVthg>

- SOLUCIÓN POR BLOQUES <https://www.steamakersblocks.com/web/project/2520583>
- SOLUCIÓN POR CÓDIGO

<https://docs.keyestudio.com/projects/KS5009/en/latest/docs/Python/Python.html#project-5-1-control-the-door>

Project 5.2 Close the Window

Si llueve, que cierre la ventana. Se ha decidido poner como límite de humedad 50, con sólo humedecer un poco el dedo, supera ese umbral, y si esta poco mojado, se seca enseguida y baja de 50 con lo que se vuelve a abrir la ventana.

Los ángulos correctos para el servo son

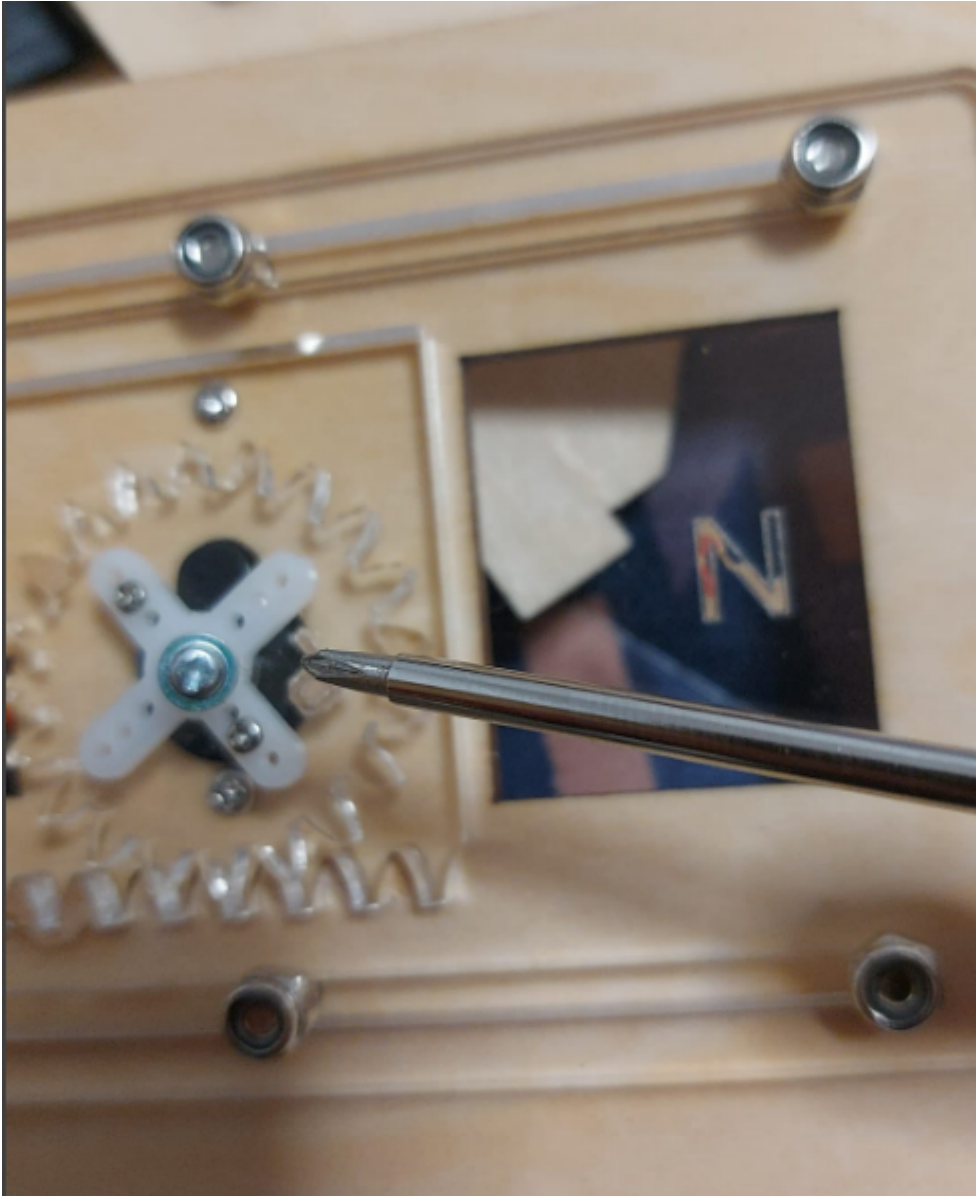
- 120º la ventana esta abierta
- 0º la ventana esta cerrada

<https://www.youtube.com/embed/tUgClbuZKTQ>

- SOLUCIÓN POR BLOQUES <https://www.steamakersblocks.com/web/project/2521324>
- SOLUCIÓN CON CÓDIGO

<https://docs.keyestudio.com/projects/KS5009/en/latest/docs/Python/Python.html#project-5-2-close-the-window>

AJUSTAMOS LA VENTANA PARA QUE SE CIERRE Y ABRA CORRECTAMENTE CON ESTOS ÁNGULOS Y YA PODEMOS PONER EL TORNILLO CENTRAL



Project 6.1 RGB Sk6812

<https://www.youtube.com/embed/CqLUw9FjAEs>

<https://www.youtube.com/embed/AujK40wtbx0>

- SOLUCIÓN CON BLOQUES <https://www.steamakersblocks.com/web/project/2521483>
- SOLUCIÓN CON CÓDIGO

<https://docs.keyestudio.com/projects/KS5009/en/latest/docs/Python/Python.html#project-6-1-rgb-sk6812>

Hemos puesto dos vídeos, uno con el programa de bloques y otro con el de código ¿Quién es quien?

Project 7.2 Button Control Fan

<https://www.youtube.com/embed/sK7B5ff6m4g>

- SOLUCIÓN POR BLOQUES <https://www.steamakersblocks.com/web/project/2521510>

- SOLUCIÓN CON CÓDIGO

<https://docs.keyestudio.com/projects/KS5009/en/latest/docs/Python/Python.html#project-7-2-button-control-fan>

Project 8.2 Dangerous Gas Alarm

El detector de gas MQ2 es sensible al gas butano, si tenemos vitro, probarlo con un encendedor de gas pero sin hacer la chispa.

Hemos elegido que si detecta gas sale una advertencia por el display, suena un tono y hace funcionar el ventilador

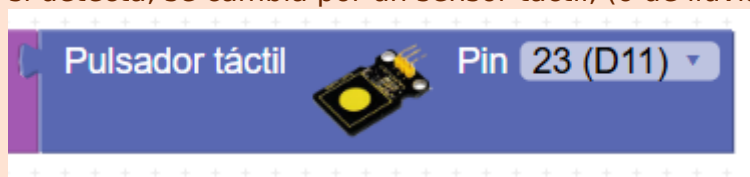
<https://www.youtube.com/embed/UMf1K0L-ZPs>

También detecta el **alcohol**, como se puede ver en el curso de SMART HOME PARA MICROBIT

<https://libros.catedu.es/books/smart-home-para-microbit/page/gas>

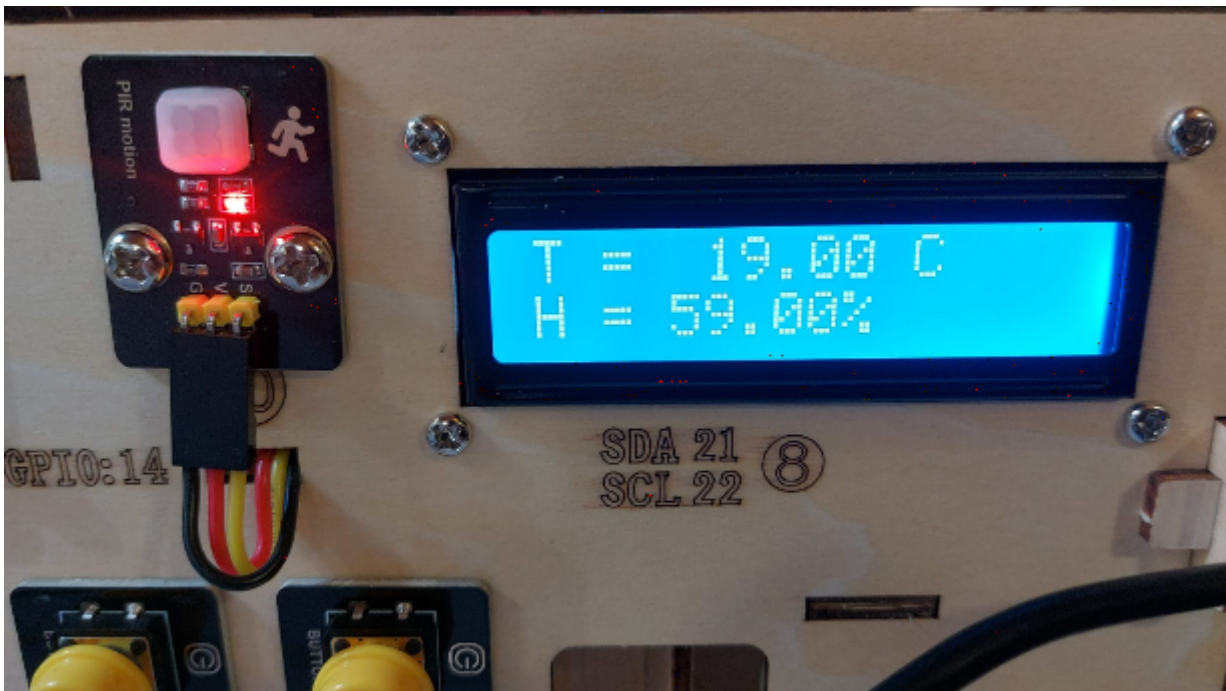
<https://www.youtube.com/embed/hSX2yizTSH4>

ATENCIÓN: EN STEAMAKERSBLOCK no se encuentra este sensor de gas, pero como es un sensor de todo o nada pull down, es decir digital, que emite un 1 si no detecta y emite un 0 si detecta, se cambia por un sensor táctil, (o de lluvia o de cualquier cosa que haga 1 o 0)



- SOLUCIÓN CON BLOQUES HOME <https://www.steamakersblocks.com/web/project/2523009>
- SOLUCIÓN CON CÓDIGO
<https://docs.keyestudio.com/projects/KS5009/en/latest/docs/Python/Python.html#project-8-2-dangerous-gas-alarm>

Project 9.1 Temperature and Humidity Tester



- SOLUCION POR BLOQUES <https://www.steamakersblocks.com/web/project/2523126>
- SOLUCION CON CÓDIGO

<https://docs.keyestudio.com/projects/KS5009/en/latest/docs/Python/Python.html#project-9-1-temperature-and-humidity-tester>

Project 10.1 Open the Door

<https://www.youtube.com/embed/zlfVCRCNZ68>

- SOLUCIÓN CON BLOQUES <https://www.steamakersblocks.com/web/project/2523179>

- SOLUCIÓN CON CÓDIGO

<https://docs.keyestudio.com/projects/KS5009/en/latest/docs/Python/Python.html#project-10-1-open-the-door>