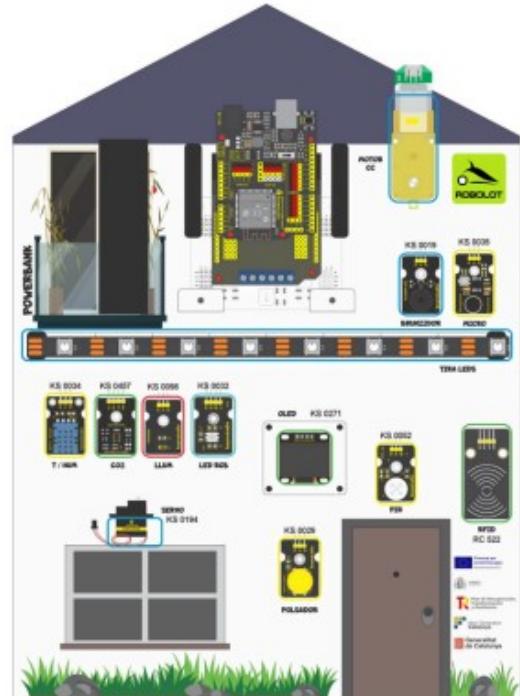
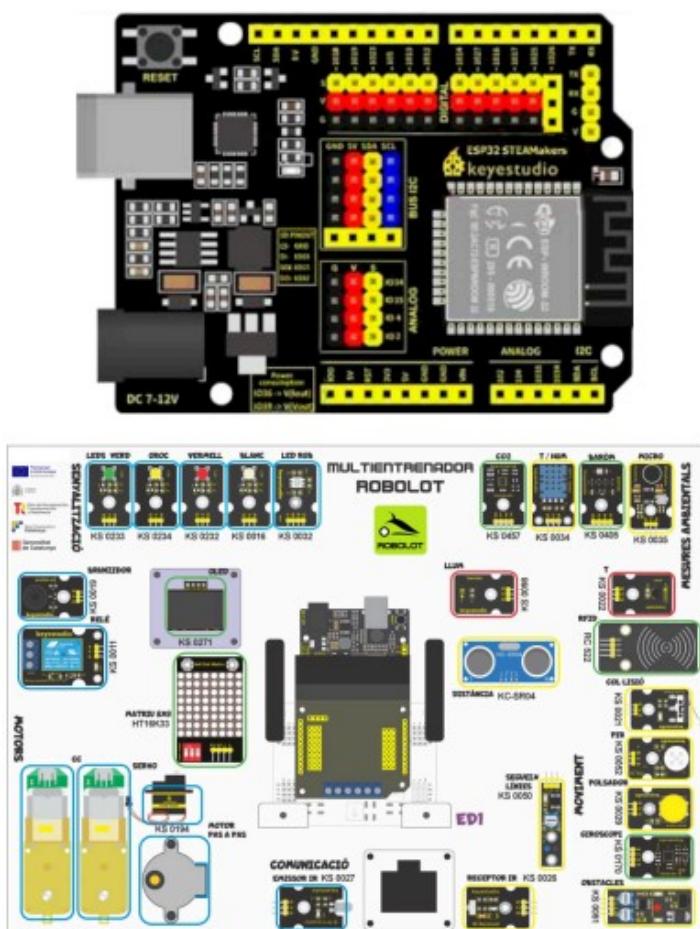


# **MANUAL DE PRÁCTICAS**

## **con ESP32 STEAMakers y arduinoblocks**



# ESP32 STEAMakers

Elaborado para la formación:

Retos de robótica con la placa ESP32 STEAMakers

Traducción de la versión 2.0 – Noviembre de 2024

**Autoria:** Cristina Blaya Góngora, Andreu Bonet Beltran, Josep Maria Filella Agulló, Sergi Horrillo Garcia, Toni Hortal Martínez, Miguel Pérez Manso, Marta Simon Bonet, Dani Soldevila Guitart y Alan Torrens Berenguer.

Traducción: Federico Coca



Esta obra está sujeta a una licencia de

[Atribución/Reconocimiento-NoComercial-CompartirlGual 4.0 Internacional](#)

## **Índice**

Introducción.....	1
Funcionamiento de un sistema de control programado.....	2
Elementos de los conjuntos de sensores y actuadores.....	4
Placa de control ESP32 STEAMakers.....	5
Programación con arduinoblocks.....	7
INSTALACIÓN DE ARDUINOBLOCKS.....	9
CREAR UNA CUENTA.....	10
P01. Encender un LED.....	11
P02. Encender un LED con un pulsador.....	15
P03. Sensor de luz (sensor analógico).....	22
P04. LED RGB.....	25
P05. Medida de la temperatura y la humedad con el DHT-11.....	29
P06. Pantalla OLED.....	33
P07. Matriz de LEDs 8x8.....	37
P08. Lista de reproducción con el zumbador.....	41
P09. Medir distancias con el sensor de ultrasonidos.....	45
P10. Controlar un servomotor.....	48
P11. Tira de LEDs NeoPixel.....	50
P12. Giroscopio y acelerómetro.....	54
P13. Bomba de agua.....	59
P14. Condiciones ambientales.....	61
P15. Motor de CC.....	64
P16. infrarojos + mando a distancia.....	69
P17. Emisor Infrarojos.....	72
P18. Detector de movimiento PIR.....	75
P19. Módulo RFID.....	78
P20. Seguidor de líneas.....	81
P21. Sensores integrados y consumo de energía.....	84
Glosario de términos de programación.....	88
Reconocimientos.....	88
Otras fuentes de documentación utilizadas.....	88
Imágenes y material gráfico.....	88

## Introducción

El presente manual surge del trabajo cooperativo de un equipo de profesores y profesoras de secundaria de toda Cataluña que durante el verano de 2024 se reunieron en la Garrotxa, gracias a la financiación del fondo Next Generation, con el objetivo de investigar y sacar provecho educativo a los materiales de robótica que habían llegado recientemente a los centros educativos. Estas personas, unidas por la pasión por llevar la programación y la robótica educativa a las aulas, analizamos las necesidades de formación docente que había que ofrecer para acompañar el proceso de implantación de dicha dotación. Una inversión tan grande en materiales de última generación no puede quedar olvidada en los armarios de los institutos por una falta de conocimiento del profesorado. Desgraciadamente los asesoramientos técnicos ofrecidos por el Departamento de Educación en los claustros a finales del curso pasado no han cubierto las necesidades del profesorado. Por ello, desde la Asociación Robolot Team se han preparado una serie de formaciones al profesorado para el curso 2024/25. Este manual es la guía práctica de una de estas formaciones.

Este documento es una recopilación no exhaustiva de prácticas para aprender a utilizar algunos de los materiales de la dotación de robótica educativa que llegaron a los centros en el curso 2023/24. En concreto trabaja con la placa electrónica de control ESP32 STEAMakers y los conjuntos de sensores y actuadores secundaria básico y avanzado (maleta verde y maleta roja). Todas las prácticas recogidas en el presente documento están pensadas para programarse con arduinoblocks. Este entorno de programación por bloques ha sido desarrollado por el profesor Juanjo López y permite aprender a programar de una forma visual y sencilla.

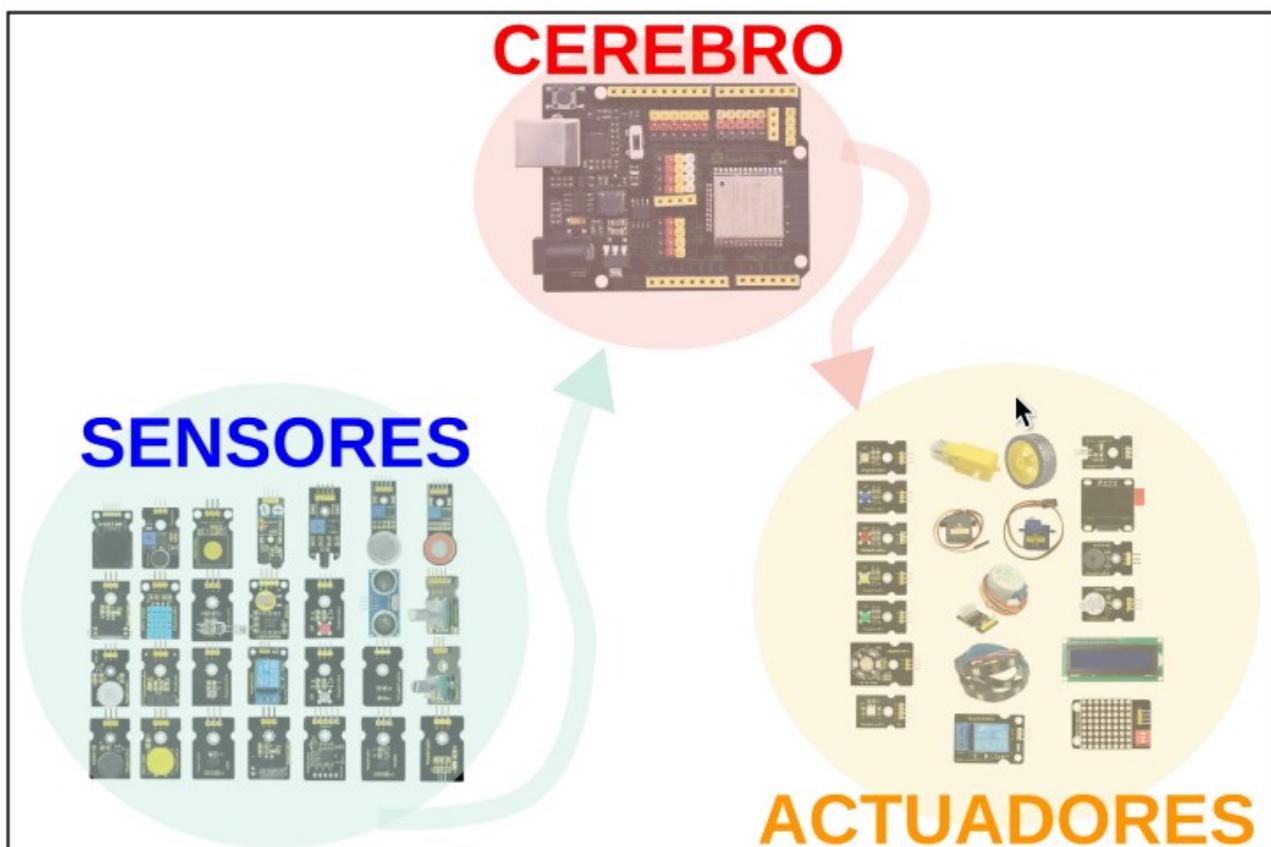
Las prácticas aquí desarrolladas, así como los retos de ampliación, están orientadas a la aplicación directa en las aulas de secundaria. La filosofía del Aprendizaje Basado en Retos quiere superar el modelo de clase magistral poniendo al alumno en el centro, y haciéndolo protagonista de un aprendizaje autorregulado, basado en los errores como parte del proceso.

El material recopilado en este manual contribuye a la adquisición de la competencia en pensamiento computacional por parte del alumnado. De forma paralela, potencia la relación entre las materias del ámbito STEAM y promociona la cultura maker entre el profesorado.



## Funcionamiento de un sistema de control programado

Un sistema de control programado funciona de forma similar a la de un ser humano. Cuando nuestro cerebro recibe información de los sentidos (oído, olfato, gusto, vista y tacto) analiza esta información, la procesa y da órdenes a nuestros músculos para realizar movimientos, dar estímulos a las cuerdas vocales para emitir sonidos, etc. Los 5 sentidos equivalen a entradas de información y la voz o músculos serían las salidas.

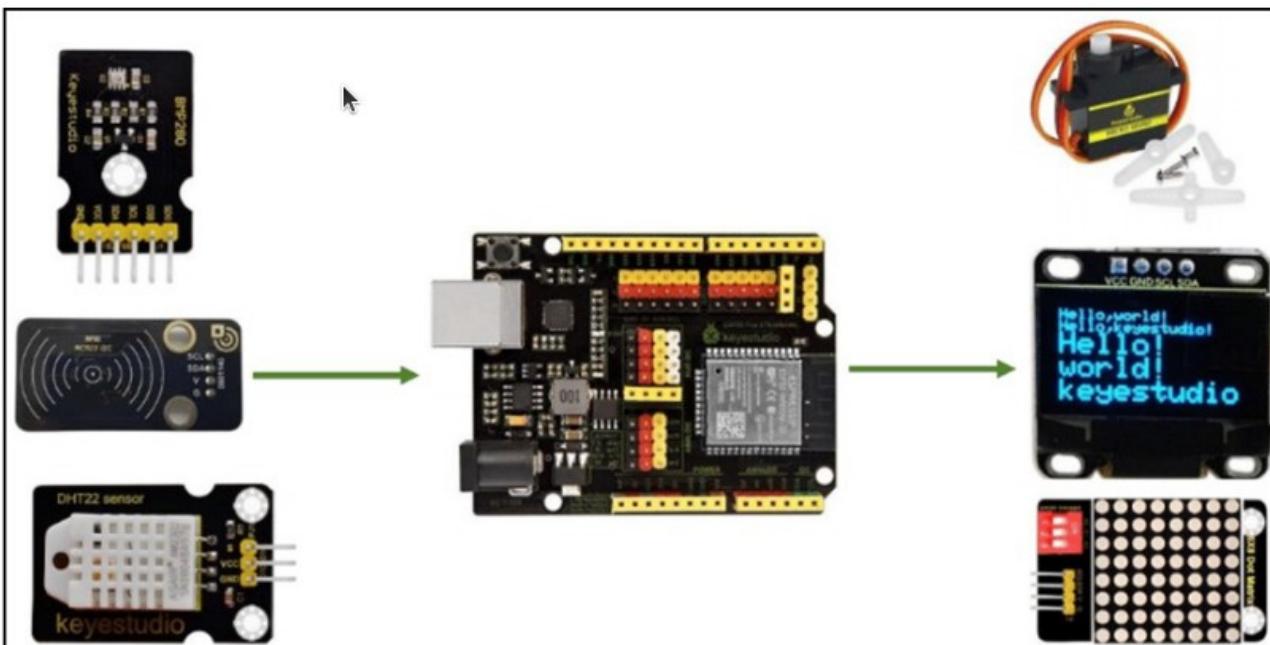
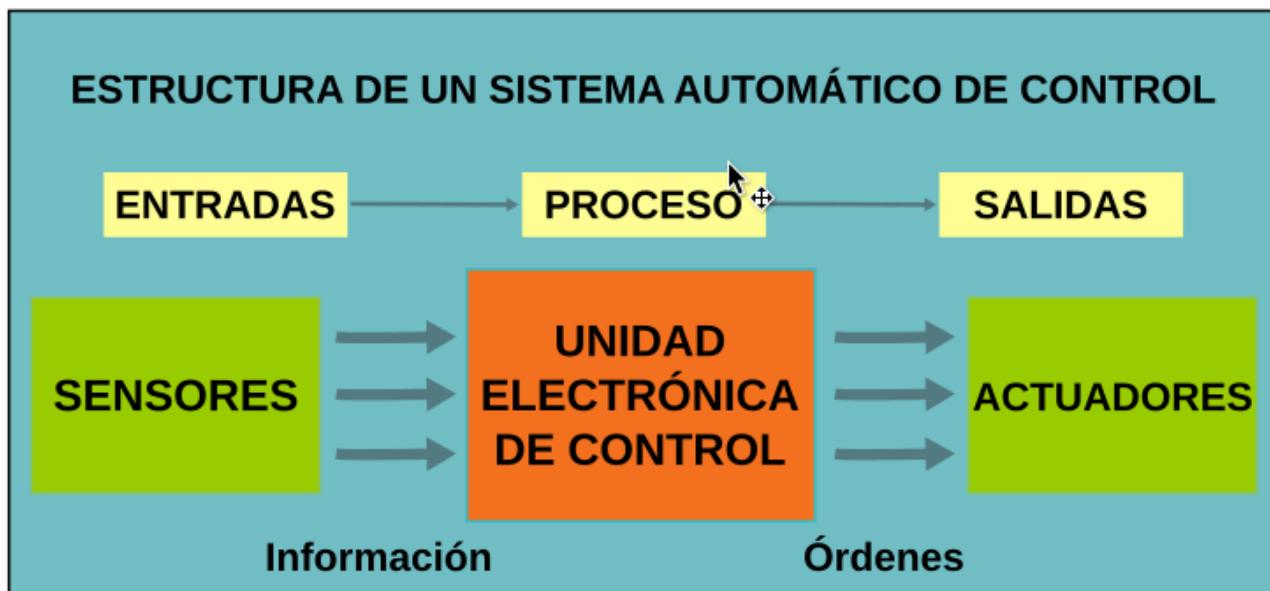


En el caso de un sistema de control programado, un microcontrolador realiza la función de cerebro. Este componente electrónico unas entradas de información donde se conectan los sensores de luz (LDR), temperatura (NTC), sonido, y también tiene salidas, donde se conectan los motores, LEDs, zumbadores, etc.



La diferencia principal es que, mientras que nuestro cerebro ha ido aprendiendo lo que debe hacer a lo largo de nuestra vida a base de estímulos, la memoria del sistema programado está vacía, es decir, no sabe lo que debe hacer y, por tanto, es necesario que nosotros le digamos cómo debe actuar en función de las señales que recibe de los sensores. Estos comandos se generan mediante el código del programa que introducimos en el sistema de control.

En todo sistema de control automático, los sensores (entradas) convierten a las magnitudes físicas, químicas, etc. en eléctricas, creando así la información. Esta información se envía a la unidad de control (proceso) que procesa la información y genera las órdenes mediante señales eléctricas que serán convertidas en los actuadores (salidas) en otros tipos de magnitudes.



## Elementos de los conjuntos de sensores y actuadores

### Sensores y actuadores ESO básicos



Ficha resumen

- Sensor de luz
- Sensor de humedad del suelo
- Sensor de temperatura
- Pantalla OLED
- Sensor de colisión o fin de carrera
- Sensor infrarrojo x3
- Sensor PIR
- Sensor de distancia por ultrasonidos
- Tira de 30 leds (1 m)
- Motor con rueda x2
- Servomotor x2
- Cables Dupont x30
- Cables GVS x5

### Sensores y actuadores ESO avanzados



Ficha resumen

- Sensor de sonido/micrófono
- Giroscopio y acelerómetro
- Sensor CO<sub>2</sub>
- Sensor de humedad y temperatura
- Sensor de presión barométrica
- LED blanco
- LED amarillo
- LED verde
- LED rojo
- LED RGB
- Zumbador
- Relé
- Bomba de agua
- Receptor IR y mando
- Emisor IR
- Pulsador
- Cables Dupont x30
- Cables GVS x5

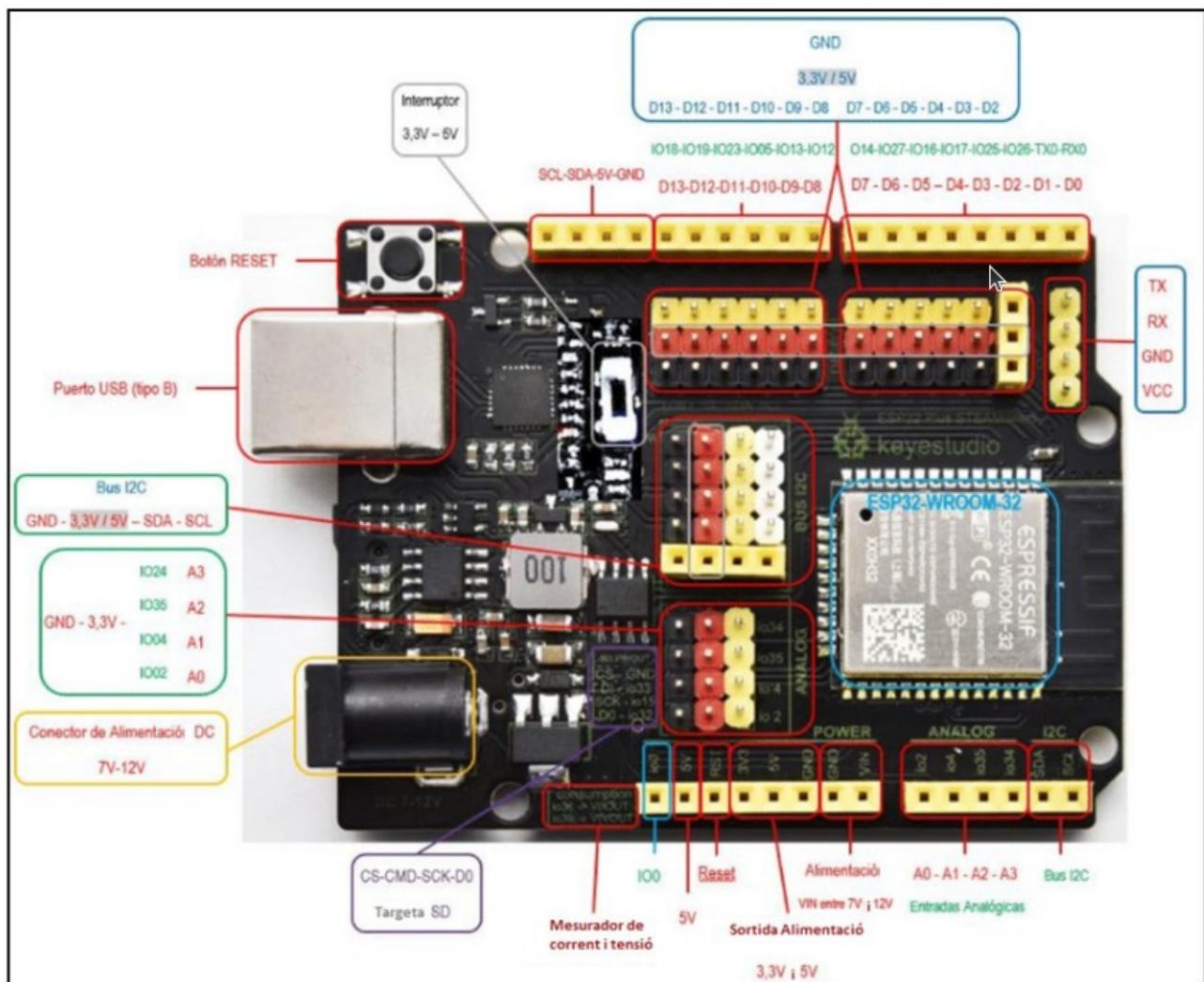
## Placa de control ESP32 STEAMakers

La placa está basada en una placa Arduino UNO. Arduino es una plataforma de prototipos electrónicos de código abierto (Open-Source) basado en hardware y software libre, flexible y fácil de usar, con una comunidad muy grande que genera muchísima información.

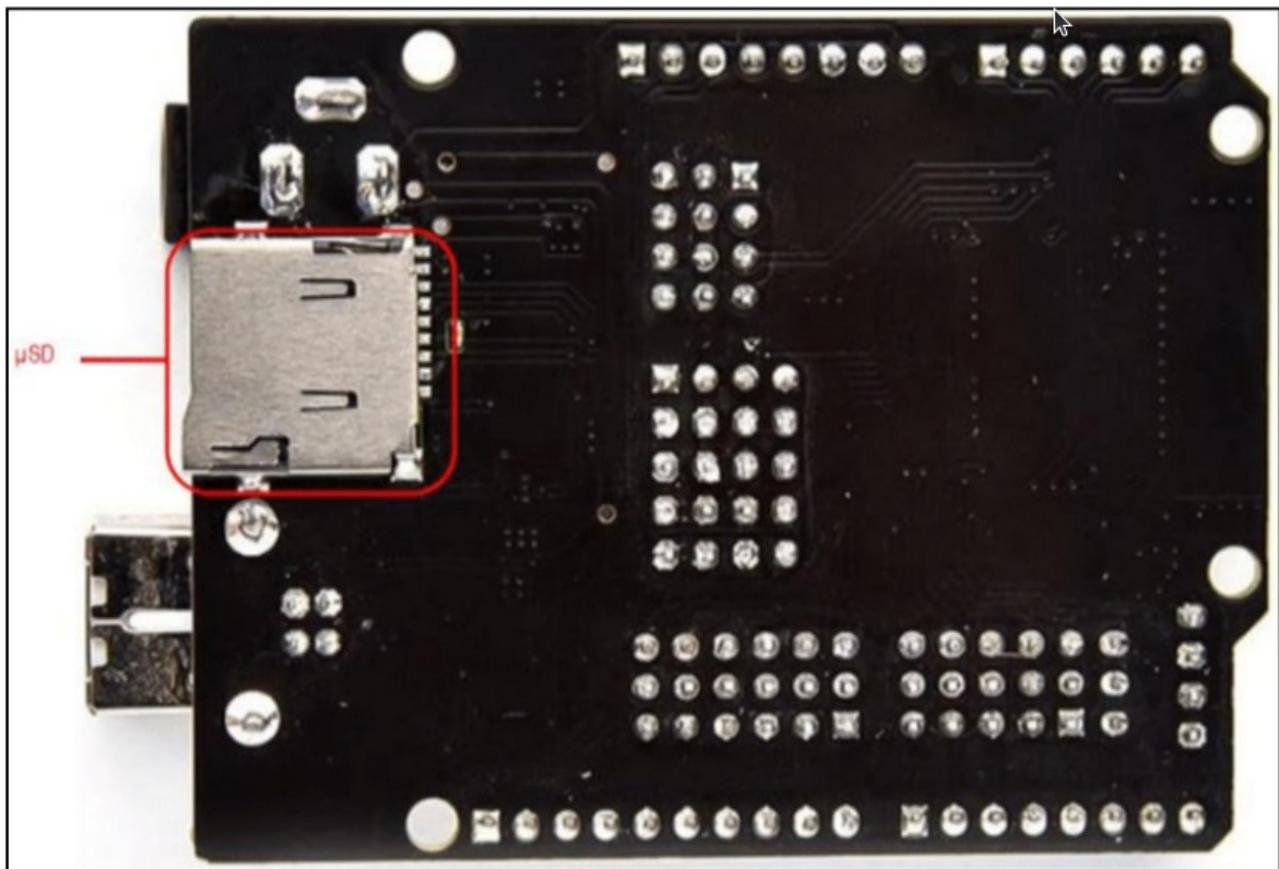
Esta plataforma permite crear distintos tipos de sistemas para diferentes usos. Puede ser utilizada por artistas diseñadores, ingenieros, profesores, alumnos, etc., y en general cualquier persona que esté interesada en crear objetos o entornos interactivos, prototipos, sistemas robóticos, etc.

La placa tiene los siguientes elementos.

Por la parte delantera:



Por la parte trasera de la placa:



Al ser Arduino un hardware libre, existen multitud de fabricantes que han desarrollado versiones basadas en Arduino. Uno de estos fabricantes es Keyestudio, que ha desarrollado junto al equipo Innova Didáctic una placa compatible con Arduino, pero basada en una ESP32. La nueva placa ESP32 Plus STEAMakers.

## Programación con arduinoblocks



Arduinoblocks es una plataforma web online, creada por el profesor Juanjo López, donde puede programar su placa de forma visual (consulte la lista adjunta de placas soportadas). Los diferentes bloques permiten leer diferentes entradas y enviar valores a las salidas de la placa, así como programar funciones lógicas, de control, etc.

La programación en arduinoblocks se hace con bloques al estilo de MicroBlocks. No hay que escribir líneas de código y no está permitido unir bloques incompatibles, evitando así posibles errores de sintaxis.

La plataforma arduinoblocks genera, compila y envía el programa a la placa por medio de la conexión USB.

Dependiendo del sistema operativo, será necesario disponer de los correspondientes drivers instalados y, siempre, ejecutar AB-connector para poder enviar el programa a la placa.

Una vez subido el programa a la placa, no será necesaria conexión lógica con el ordenador para funcionar. Puede alimentar la placa con baterías, la misma conexión USB o una fuente de alimentación para que funcione de forma autónoma.

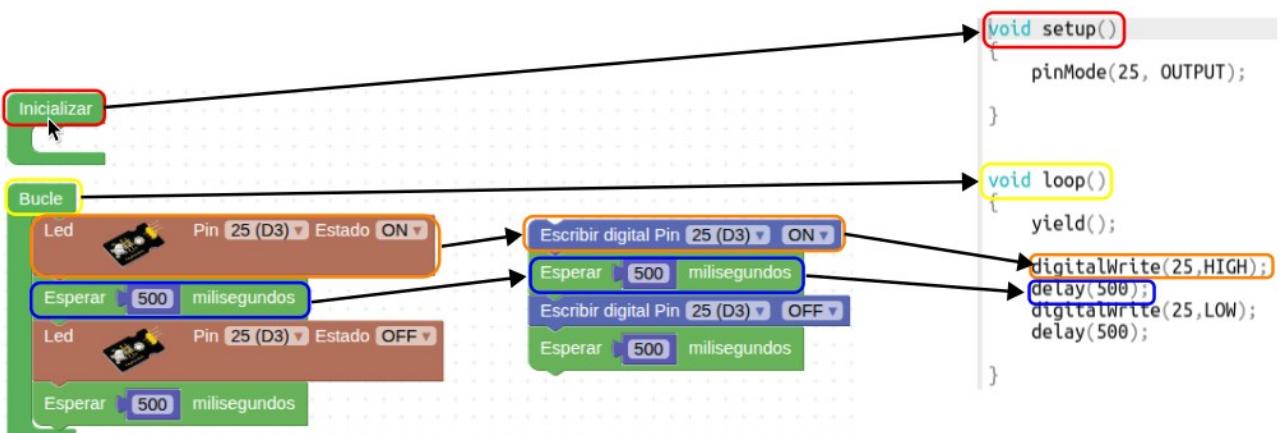
Arduinoblocks funciona actualmente con todos los navegadores de última generación: Google Chrome, Microsoft Edge, Mozilla Firefox, Opera, Safari, etc.

### Placas soportadas

- UNO
- NANO / ATmega328
- NANO / ATmega328 (new bootloader)
- MEGA / 2560
- Leonardo
- UNO + Imagina TdRSTEAM
- UNO + Imagina 3DBot
- Keyestudio EasyPlug
- Keyestudio KeyBot
- Keyestudio KidsIoT
- ESP32 STEAMakers
- ESP32 STEAMakers + Imagina TdR STEAM
- ESP32 STEAMakers + Imagina 3DBot
- ESP32 micro:STEAMakers
- ESP32 / WROOM
- ESP8266 / NodeMCU v2
- ESP8266 / WeMos D1
- Otto DIY / Nano
- Otto DIY / Nano (new bootloader)



Podemos programar arduinoblocks de diferentes maneras ya que tiene múltiples bloques. También permite exportar el código por el IDE de Arduino.



## INSTALACIÓN DE ARDUINOBLOCKS

Arduinoblocks trabaja online, funciona con todos los navegadores de última generación: Google Chrome, Microsoft Edge, Opera, Safari, etc. La programación no requiere ninguna instalación específica, pero sí se necesita un programa para poder enviarlo al microcontrolador de la placa. Este programa se llama AB-Connector.

- Para ordenadores de dotación:

Buscarlo en el portal correspondiente.

- Para ordenadores personales:

[Enlace](#)

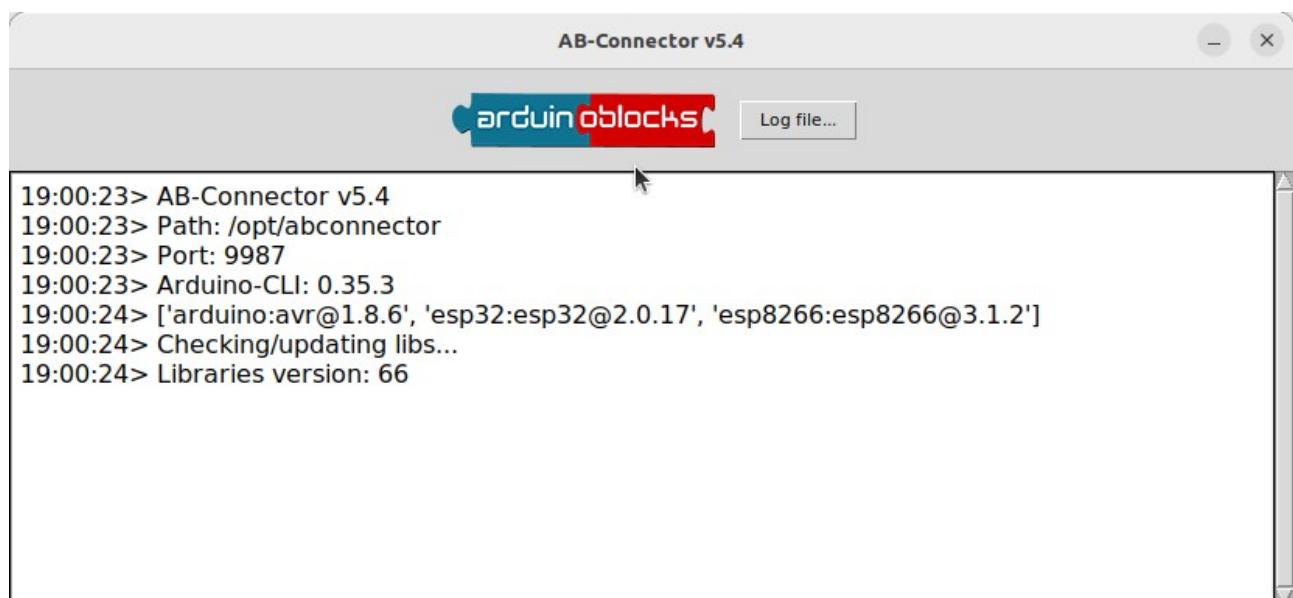
El icono que se ve después de la instalación es el siguiente:



Deberemos tener siempre abierto este programa cuando enviamos el código a la placa o cuando queramos hacer comunicaciones por la consola o el Serial Plotter de arduinoblocks.

Primero deberemos crear una cuenta con arduinoblocks y después instalar el software AB-Connector para poder comunicar el ordenador con la placa a programar.

Abrimos el AB-Connector y vemos la siguiente pantalla:



AB-Connector v5.4

arduinoblocks Log file...

```
19:00:23> AB-Connector v5.4
19:00:23> Path: /opt/abconnector
19:00:23> Port: 9987
19:00:23> Arduino-CLI: 0.35.3
19:00:24> ['arduino:avr@1.8.6', 'esp32:esp32@2.0.17', 'esp8266:esp8266@3.1.2']
19:00:24> Checking/updating libs...
19:00:24> Libraries version: 66
```

## CREAR UNA CUENTA



Tengo problemas con mi proceso de registro o inicio de sesión



Captcha

iztfeec



## P01. Encender un LED



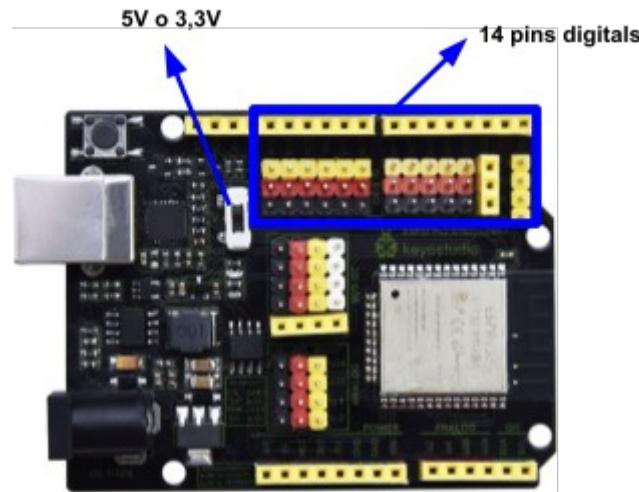
### DESCRIPCIÓN

- Encender y apagar un led de forma intermitente
- Regular la intensidad de un LED

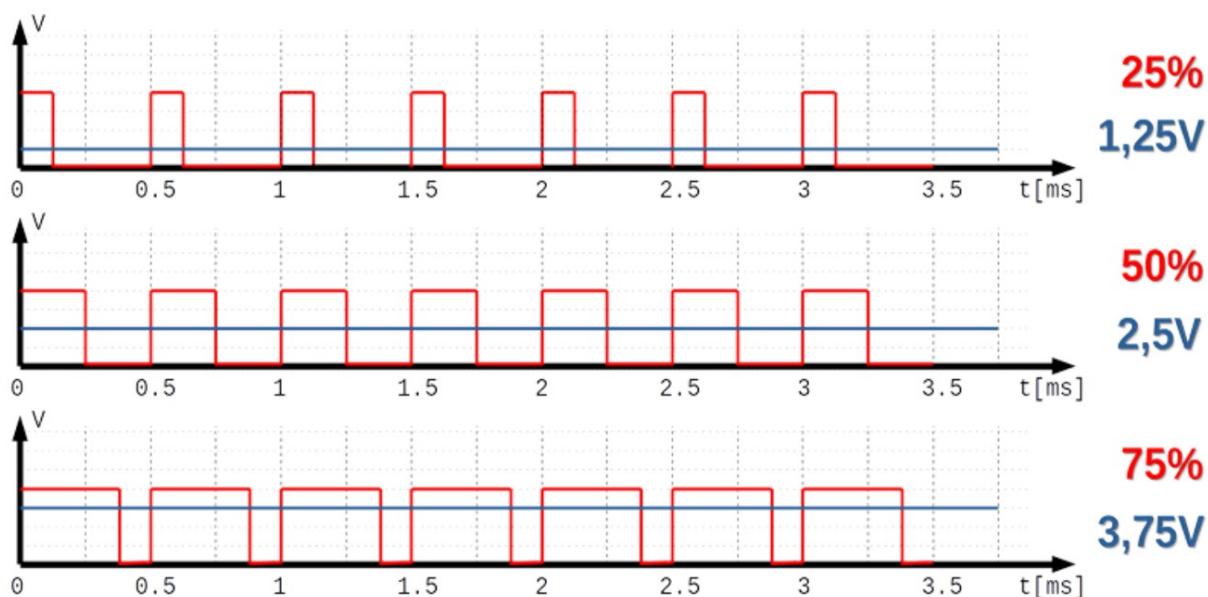
### CONOCIMIENTOS

- Salida digital: Es un pin del microcontrolador que puede ser configurado para enviar señales de salida binariaS, es decir, señales que sólo pueden tener dos estados: alto (HIGH) o bajo (LOW).

La placa tiene un interruptor para configurar el nivel alto en 5V o 3,3V. el nivel bajo son 0V.



- Salida analógica: Podemos configurar algunos pines para generar señales analógicas utilizando la técnica PWM (Modulación por Ancho de Pulso). El PWM permite simular una salida analógica con un pin digital cambiando la cantidad de tiempo que el pin está encendido (HIGH) o apagado (LOW) durante un ciclo. El ciclo de trabajo es el porcentaje de tiempo que la señal está encendida. Esto permite controlar el brillo de los leds o la velocidad de los motores ajustando la energía que se entrega al dispositivo.



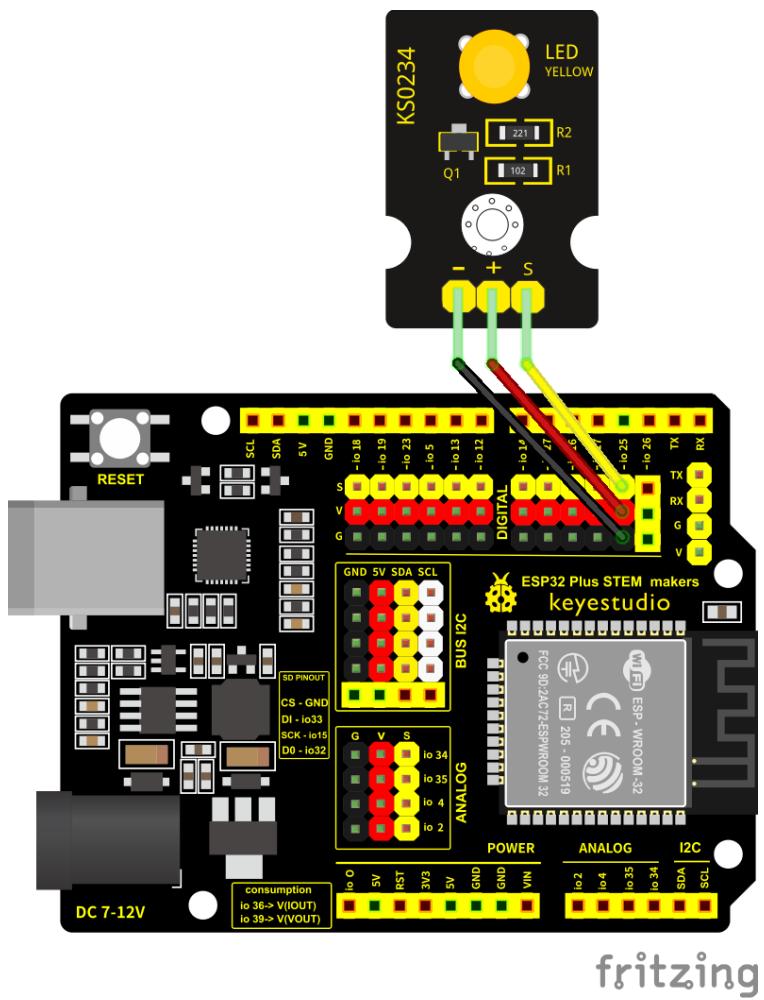
## LISTA DE ELEMENTOS

- 1 placa ESP32 STEAMakers
- LEDs

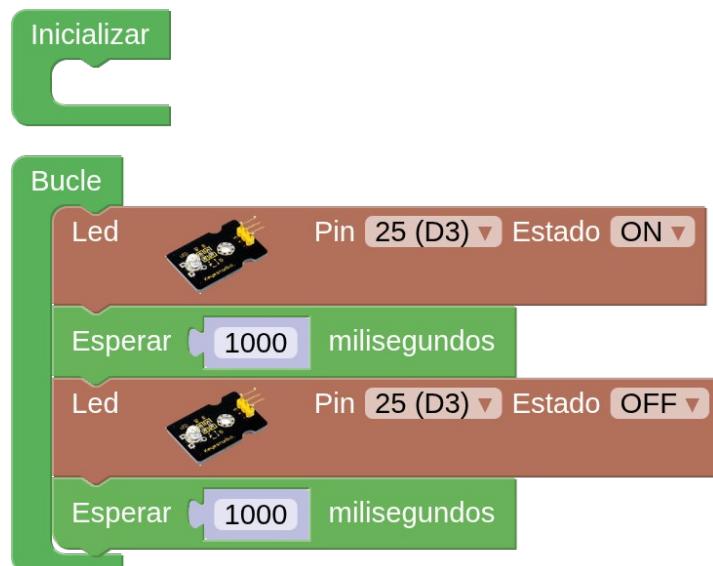


## MONTAJE

El montaje es el mismo para las dos partes de la práctica.



## PROGRAMACIÓN 1.1

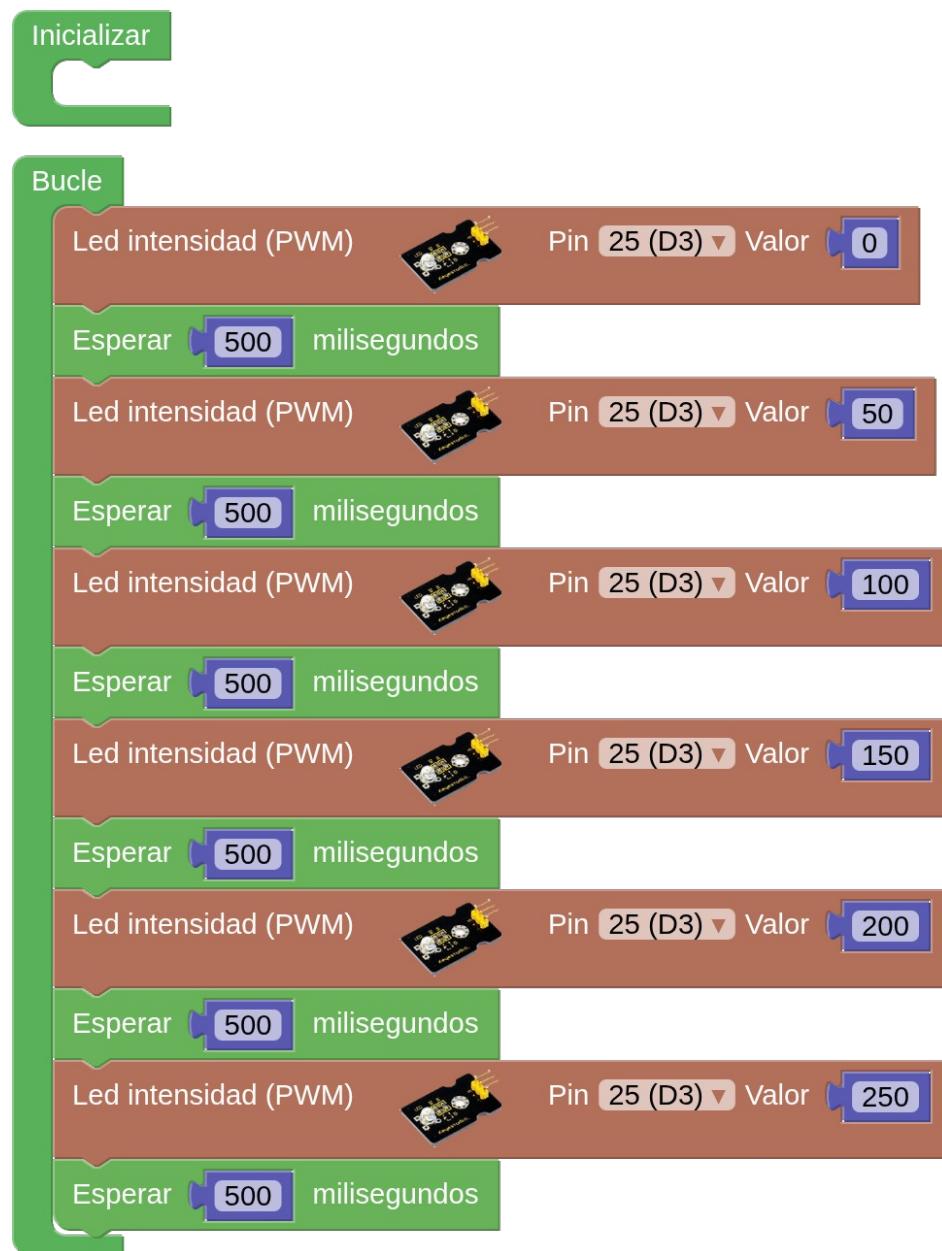


El LED se deberá de encender y apagar con una cadencia de un segundo.

Prueba a cambiar los tiempos de espera que como ves están medidas en milisegundos.



## PROGRAMACIÓN 1.2



El LED deberá aumentar el brillo progresivamente. El valor que enviamos a la salida PWM (“analógica”) es un valor de 8 bits y, por tanto, puede tomar valores decimales de entre 0 y 255 en decimal. ( $2^8=256$  valores posibles). Un valor de 0 equivale a 0 V y un valor 255 equivale a un valor de 5 o 3,3 V (dependiendo del valor con el que trabajemos).

### RETOS AMPLIACIÓN

- Semáforo: Simula el funcionamiento de un semáforo de tráfico utilizando los leds rojo, amarillo y verde. Programa los leds para encenderlos en secuencia (verde, amarillo, rojo) con intervalos de tiempo adecuados.



## P02. Encender un LED con un pulsador



### DESCRIPCIÓN

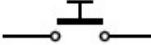
- Encender y apagar un LED mediante un pulsador digital.

### CONOCIMIENTOS

En el siguiente reto utilizaremos el pulsador. Previamente, debemos recordar la diferencia entre un pulsador y un interruptor. Un interruptor es un dispositivo que abre o cierra el paso de la corriente eléctrica, por ejemplo, los interruptores de la luz de nuestras casas, cada vez que los pulsamos cambian de estado y permanecen en él hasta ser pulsados de nuevo. Sin embargo, un pulsador sólo se activa mientras dure la pulsación volviendo a su estado inicial en el momento en el que se deja de pulsar.

Hay dos tipos de pulsadores:

- NA (Normalmente Abierto): Cuando no se pulsa el pulsador, el circuito está abierto (sin conexión). Al pulsarlo, el circuito se cierra y activa la función o acción asociada.
- NC (Normalmente Cerrado): Cuando no se pulsa el pulsador, el circuito está cerrado (con conexión). Al pulsarlo, el circuito se abre e interrumpe la función o acción asociada.

		
Símbolo del interruptor	Símbolo del pulsador NA	Símbolo del pulsador NC

En el multientrenador encontramos pulsadores digitales que pueden programarse para comportarse como lo harían estos tres componentes eléctricos.



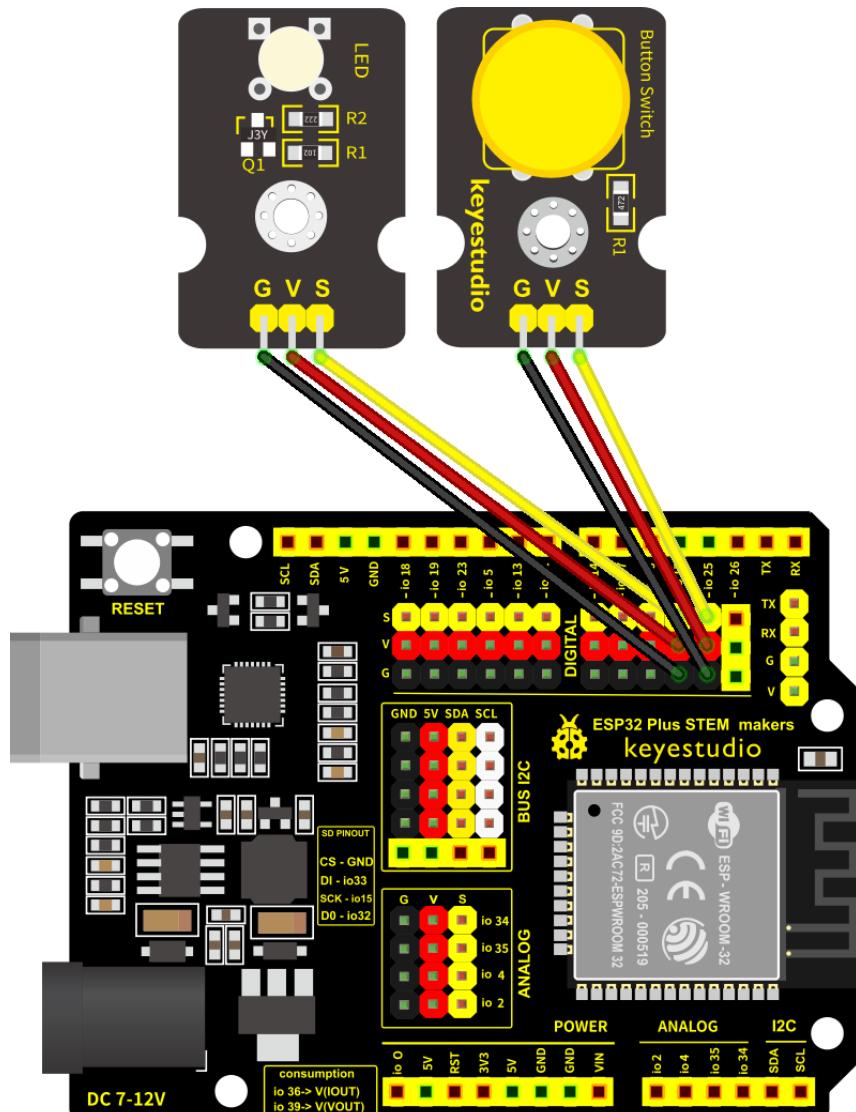
## LISTA DE ELEMENTOS

- 1 placa ESP32 STEAMakers
- LED
- Pulsador digital
- 2 cables Dupont VGS HH



## MONTAJE

Conectamos el pulsador digital al pin IO25 (D3) y el LED al IO17 (D4)



fritzing



## PROGRAMACIÓN 2.1: PULSADOR

Ahora realizaremos un programa en el que al accionar el pulsador se encienda el LED y se apague cuando lo dejamos de pulsar, es decir queremos programar el pulsador digital para que se comporte como un pulsador NA.

En el menú de arduinoblocks encontramos 2 bloques distintos de pulsador en el apartado de sensores.

Empezaremos con el primer bloque.



Para realizar este programa necesitamos conocer unas de las funciones más utilizadas en programación. Las funciones del menú Lógica con los bloques condicionales.

- Condicional:

Un condicional es una estructura de programación que permite que el código tome decisiones basadas en ciertas condiciones. Esto significa que el programa puede ejecutar distintos bloques de código según si una condición es cierta o falsa. Es como si el programa preguntara: "Si ocurre esto, ¿qué debo hacer?"

Bloque condicional simple	Bloque condicional completo

El funcionamiento con el bloque condicional simple es el siguiente: si se cumple la condición incluida en su primer apartado, entonces se realiza la acción incluida en su segundo apartado. De lo contrario, no se hace nada.

En cambio, con el bloque condicional completo, en caso de que no se cumpla la condición se realiza una acción distinta.

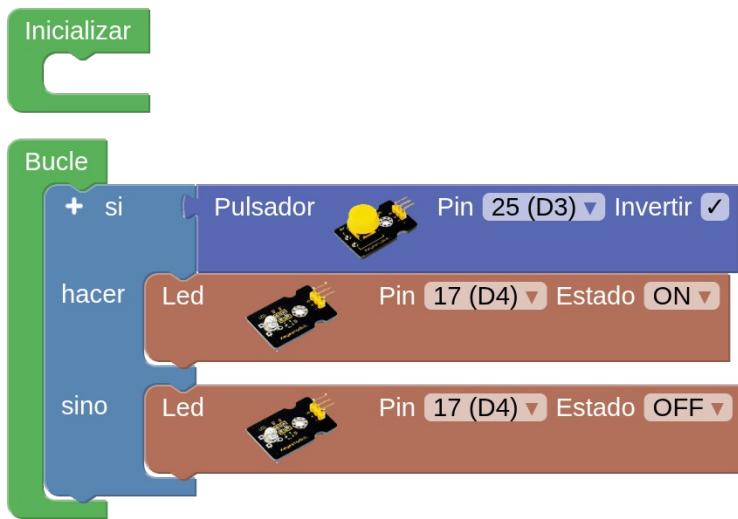
En el apartado de condiciones se puede introducir multitud de factores: estado de sensores (analógicos o digitales), comparaciones, igualdades, operaciones matemáticas, etc. Debemos tener en cuenta que siempre estamos trabajando con condiciones cierto/falso.

Para la práctica que nos ocupa utilizamos el primer bloque del pulsador dentro del bloque lógico condicional Si.... hacer... y el programa quedaría como la imagen.



Si hacemos esto no se apaga el LED nunca, esto no es lo que queremos, el problema es que en ningún momento del programa decimos que el LED deba estar en la situación de OFF.

Utilizaremos otro bloque condicional más completo que incluye el comportamiento en caso de que no se dé la condición inicial.



El LED se pondrá en marcha mientras accionemos el pulsador y se apagará al soltarlo, comportamiento normalmente abierto, NA. Si probamos el mismo programa pero sin marcar la opción invertir veremos que se comporta como un pulsador normalmente cerrado, NC.

## PROGRAMACIÓN 2.2: INTERRUPTOR

Ahora realizaremos un programa que encienda un LED cuando accionemos el pulsador y se mantendrá así hasta que lo pulsemos de nuevo, momento en que se apagará, es decir programaremos el pulsador digital para que se comporte como un interruptor eléctrico.

Para programar esta práctica necesitamos utilizar variables.

- Variables:

Una variable en programación es un elemento que utilizamos para almacenar información que puede cambiar mientras se ejecuta el programa.

Piensa en una variable como una caja con una etiqueta: puedes poner en ella diferentes valores y cambiar estos valores siempre que quieras.

Por ejemplo, si estamos programando un juego, podrías tener una variable llamada "puntos" para guardar la puntuación del jugador. Cada vez que el jugador gana puntos, el valor de la variable "puntos" cambia para reflejar la nueva puntuación. Así, una variable es una forma de guardar y gestionar datos que pueden variar durante la ejecución de un programa.

Para definir una variable en arduinoblocks se debe seleccionar el menú “variables”, donde se pueden crear tres tipos:

- Número: almacena valores numéricos
- Texto: almacena texto
- Boolean: almacena un estado binario ON u OFF (cierto o falso)





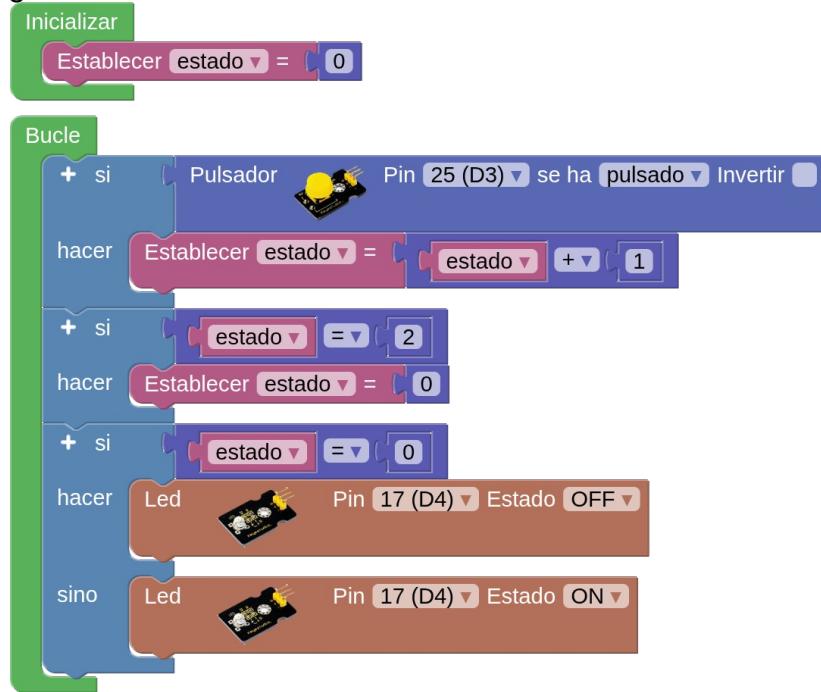
El sistema nos permite asignar a las variables un nombre simbólico, como por ejemplo “temperatura\_exterior”, “velocidad”, “posición\_servo\_1”,... para facilitar su uso. (Aunque arduinoblocks permite crear variables donde su nombre puede contener espacios en blanco, hay que acostumbrarnos a no utilizarlos, al igual que otros caracteres problemáticos como por ejemplo la “á” o la “ñ”, ya que en programación por código no se puede hacer).

En caso que nos ocupa, definimos una variable de tipo numérico: “estado” que almacenará el estado del LED. Si el valor es 0 el LED debe estar apagado, si es 1, encendido. Utilizamos una operación de suma para cambiar el valor de la variable cada vez que se pulse el pulsador. Como sólo trabajaremos con el valor 0 y el 1, cuando llegue a 2 le volvemos a dar el valor 0.

En el programa que faremos a continuación necesitamos asegurarnos de que cada vez que accionamos el pulsador sea detectado como una sola señal ya que realizaremos un contador. Por tanto debemos utilizar el segundo bloque de pulsador, que toma la señal “filtrada”, es decir, sólo detecta una pulsación bien al ser pulsado o al ser soltado.



Este sería el programa resultante:



## RETOS DE AMPLIACIÓN

- Prueba a realizar el programa de la práctica 2.1 con el otro bloque del pulsador, fíjate que en este hay dos posibles acciones se ha pulsado y se ha soltado.
- Cambio de color: programa el pulsador de forma que cada vez que los ACCIONES se encienda un LED diferente en una serie repetitiva. Para realizar esta ampliación debes conectar a la placa electrónica el resto de leds del multientrenador.



## P03. Sensor de luz (sensor analógico)



### DESCRIPCIÓN

- Mostrar el valor del sensor de luz a través de la consola serie.
- Encender un LED automáticamente cuando oscurezca (interruptor crepuscular)

### CONOCIMIENTOS

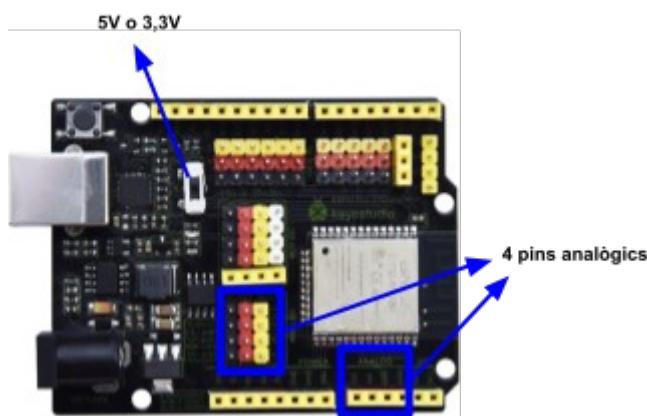
- Entrada analógica:

Imagina un control de volumen en casa. Puedes girar el botón para hacer sonar la música más fuerte o más baja. A diferencia de un interruptor que sólo tiene dos estados (arranque y apagado), el control de volumen permite ajustar el sonido a muchos niveles distintos. Los pines de entrada analógica de la placa ESP32 STEAMakers funcionan de forma similar.

Un pin de entrada analógica lee voltajes que pueden tener muchos valores distintos entre 0V y 5V (o 3,3V). Este pin convierte el voltaje analógico en un número digital que puede entender la placa ESP32 STEAMakers. Este número digital va desde 0 hasta 4095. Estos valores se deben a que las entradas analógicas de la placa son transformadas a señal digital gracias a conversores analógico-digital (ADC) de 12 bits de resolución. Con 12 bits, el rango de valores es  $2^{12}=4096$ . Estos valores van de 0 a 4095 (4096 niveles en total).

Así, cuando el pin lee 0V, el número digital será 0. Cuando lee 5V (o 3,3V), el número digital será 4095. Cualquier valor entre 0V y 5V (o 3,3V) se convertirá en un número entre 0 y 4095. Esta función nos permite leer con mucha precisión la intensidad de la luz.

En resumen, un pin de entrada analógica es como un control de volumen que puede detectar muchos niveles, no sólo encendido y apagado. Convierte un voltaje entre 0V y 5V en un número digital entre 0 y 4095 para que la placa ESP32 STEAMakers pueda entenderlo y utilizarlo.

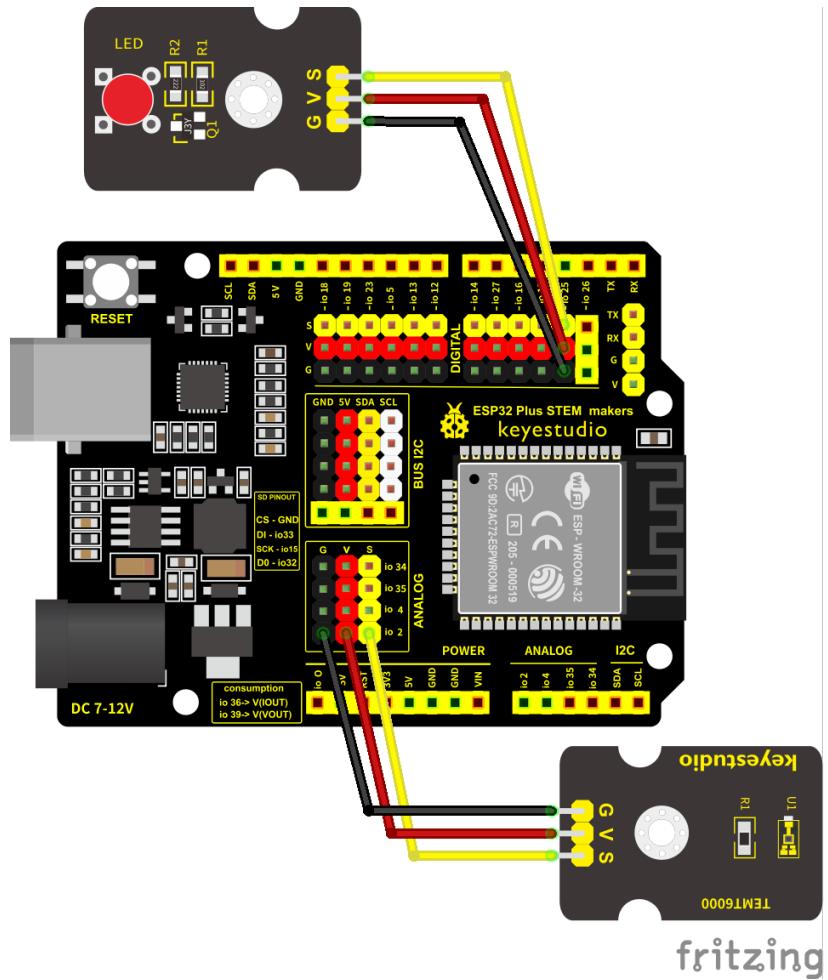


### LISTA DE ELEMENTOS

- 1 placa ESP32 STEAMakers
- Sensor de luz TEMT6000
- LED (2<sup>a</sup> práctica)



## MONTAJE



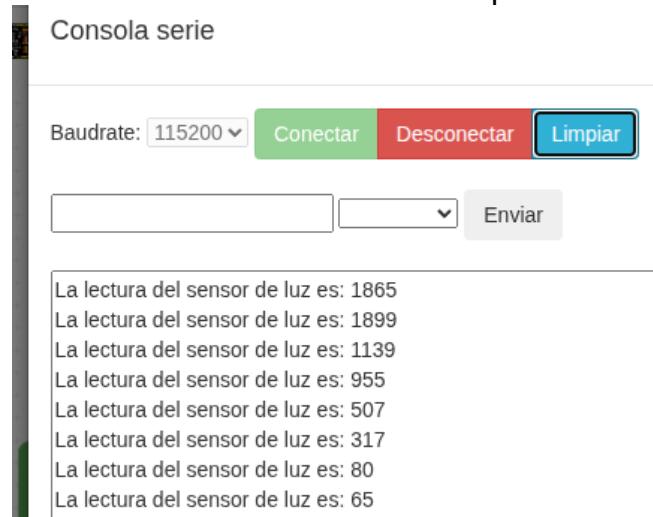
## PROGRAMACIÓN 3.1



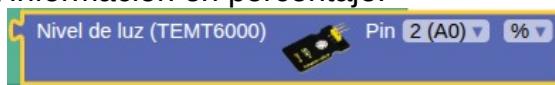
Después de cargar el programa tendrás que abrir la consola de comunicación serie a una velocidad de 115200. Esto hará que la información del sensor se envíe.



Entonces podrás visualizar en tu ordenador los valores que envía el sensor.

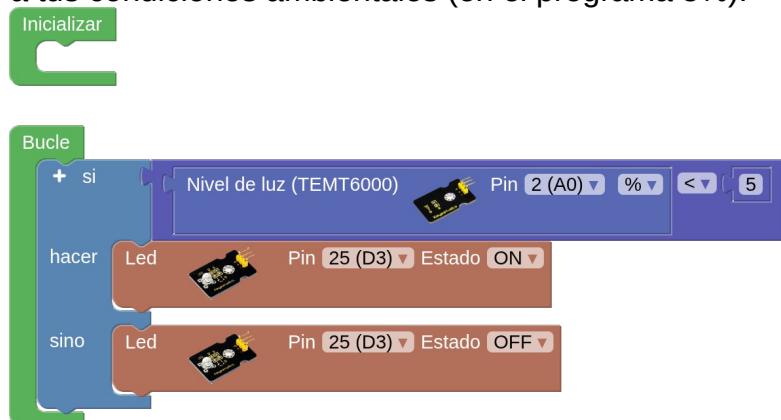


También puedes ver esta información en porcentaje.



## PROGRAMACIÓN 3.2

El LED se encenderá automáticamente cuando oscurece. Deberás establecer el valor umbral adecuado a tus condiciones ambientales (en el programa 5%).



## RETOS DE AMPLIACIÓN

- Juego de reflejos: Crea un juego donde los jugadores deben reaccionar rápidamente poniendo la mano sobre el sensor LDR cuando un LED se enciende. El juego enciende el LED de forma aleatoria, y los jugadores deben cubrir el sensor de luz con la mano tan rápidamente como sea posible. El tiempo de reacción se puede medir y mostrar en la consola serie para ver quién ha sido el más rápido.
- Alarma de luz: activa un zumbador cuando el nivel de luz cae por debajo de un cierto umbral.
- Detector de paso: Contar el número de elementos que cruzan el haz de luz. Se entiende que en función de las condiciones particulares de la luz ambiental deberá manipularse el umbral.
- Ahorro de Energía en iluminación: Reduce la intensidad de las luces cuando hay suficiente luz natural, utilizando el sensor de luz para detectar la luz ambiental.

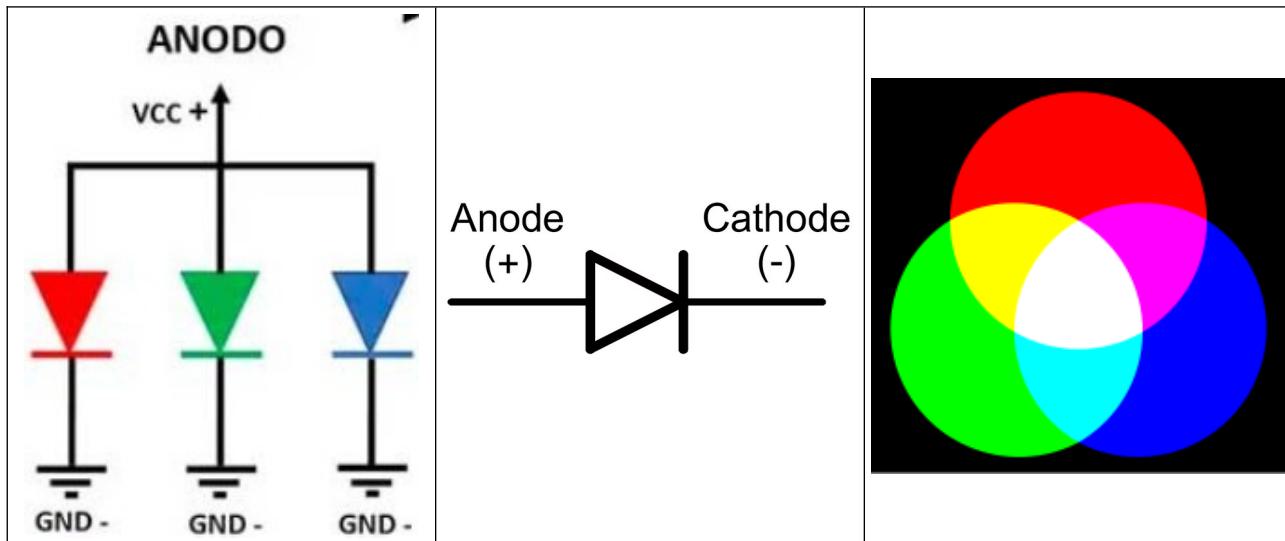
**OBJETIVOS**

- Encender un LED RGB en diferentes colores
- Hacer diferentes combinaciones de colores

**CONOCIMIENTOS**

- LED RGB:

Un led RGB es un led que incorpora en un mismo encapsulado tres leds. Las siglas RGB corresponden a: R (Red=rojo), G (Green=verde) y B (Blue=azul). Con estos tres colores, en óptica, se puede formar cualquier otro color, ajustando de forma individual la intensidad de cada uno.

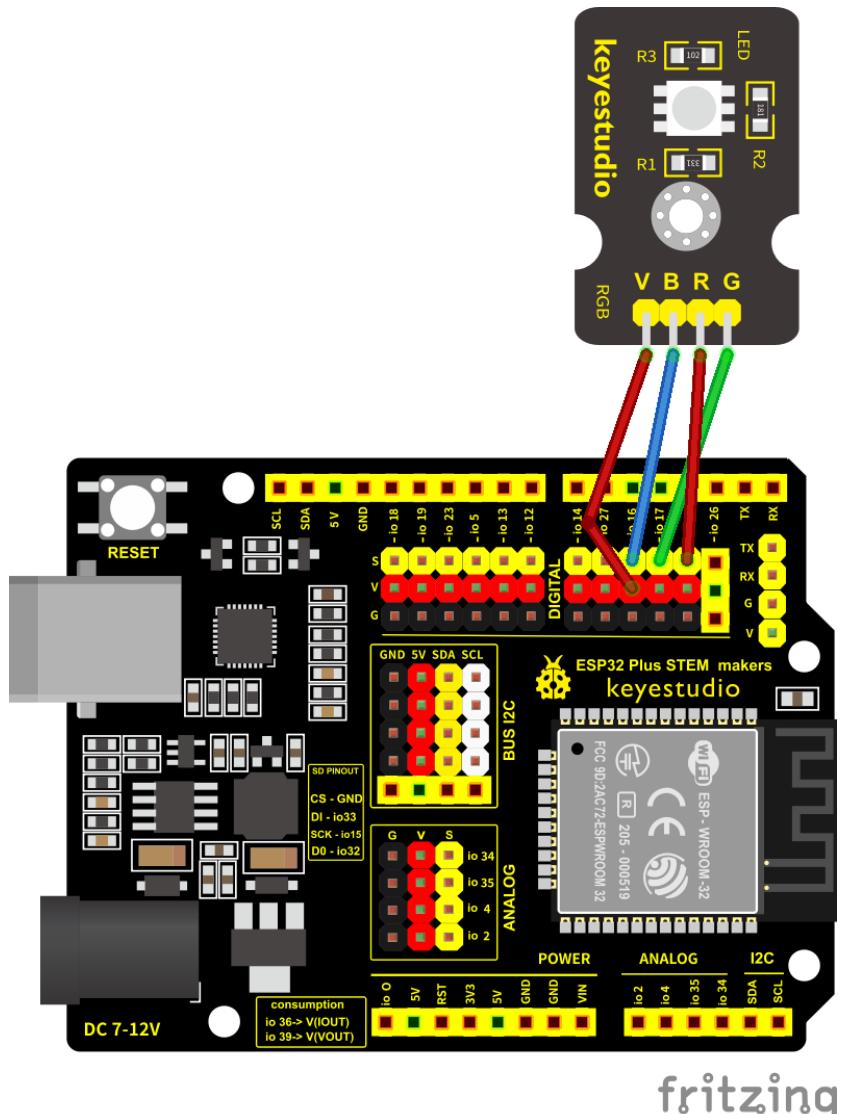


En Arduino, cada uno de estos leds podría tomar 256 colores distintos, es decir, el Rojo podría ir desde 0 hasta 255, el Verde de 0 a 255 y el Azul de 0 a 255, en total un led RGB podría dar más de 16,5 millones de colores distintos.

**LISTA DE ELEMENTOS**

- 1 placa ESP32 STEAMakers
- LED RGB
- Pulsador

## MONTAJE



fritzing

## PROGRAMACIÓN 4.1



Puedes probar este programa para ver que solamente aparece el color rojo.  
Revisa que está en ánodo común.

Puedes hacer diferentes combinaciones para probar colores diferentes.

Observa [esta web](#) y encontrarás algunas de estas combinaciones.



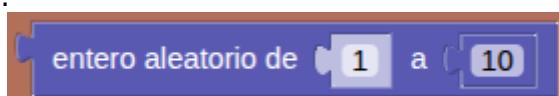
También hay un bloque en arduinoblocks que permite tomar directamente el color de una paleta.



## CONOCIMIENTOS

- Número aleatorio entre dos valores:

El bloque de número aleatorio entre dos valores es una herramienta de programación que genera un número de forma aleatoria dentro de un rango específico. Este rango se define por dos valores: un valor mínimo y un valor máximo. Cada vez que se utiliza este bloque, devuelve un número diferente dentro del rango indicado.



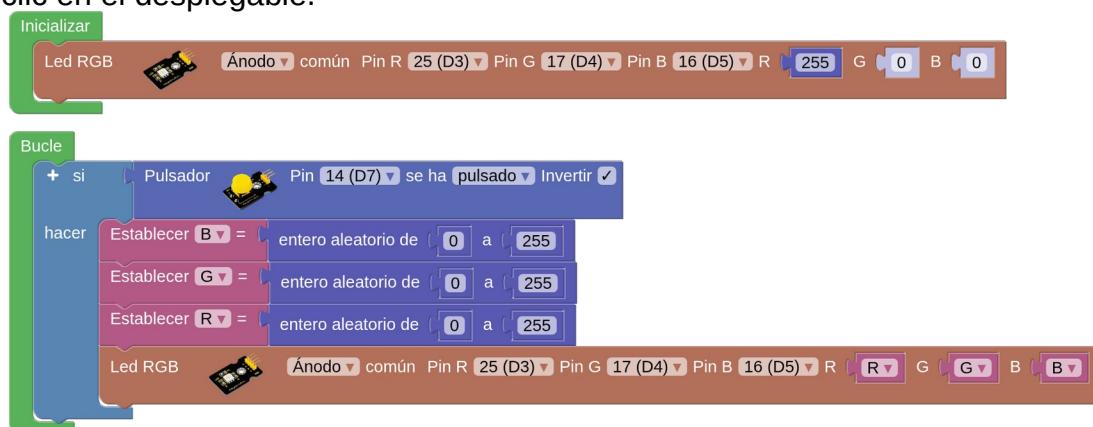
Ejemplo: Si configuramos el bloque para generar un número aleatorio entre 1 y 10, cada vez que el programa se ejecute, el bloque puede devolver cualquier número entero entre 1 y 10, como 3, 7 o 10. Este bloque es muy útil para juegos, simulaciones y otras aplicaciones donde se necesite variabilidad o imprevisibilidad.

## PROGRAMACIÓN 4.2

Intentaremos hacer un programa para mostrar un color aleatorio cada vez que se pulse el pulsador.

Se deben crear tres variables, una para cada LED.

Habrás observado que en el menú no aparecen todas las variables que has definido, se queda con la última variable en pantalla, pero las demás están guardadas, basta con hacer clic en el desplegable.



Para poder observar correctamente los colores, puede poner una hoja en blanco sobre el LED RGB.

## RETOS DE AMPLIACIÓN

- Atenuación Suave de Colores: Crea un efecto de transición suave entre colores con el LED RGB. Programa el LED RGB para que cambie de color de forma gradual y suave en un ciclo continuo. Como si haces un recorrido por los diferentes colores del Arco Iris.
- Semáforo: Simula el funcionamiento de un semáforo utilizando los colores rojo, amarillo y verde. Programa el LED RGB para que cambie de color en secuencia con intervalos de tiempo específicos.
- Juego de Memoria de Colores: Crea un juego donde los jugadores deben recordar y repetir una secuencia de colores. Programa el LED RGB para mostrar una secuencia de colores que los jugadores deben repetir pulsando botones.
- Código Morse: Usa el LED RGB para transmitir mensajes en código Morse cambiando los colores. Programa el LED RGB para mostrar mensajes en código Morse con distintos colores por puntos y rayas.



## P05. Medida de la temperatura y la humedad con el DHT-11



### OBJETIVOS

- Leer valores de temperatura y humedad utilizando el sensor DHT11.

### CONOCIMIENTOS

El sensor DHT11 permite realizar la lectura de la temperatura y de la humedad utilizando el mismo pin digital.

Este sensor mide temperaturas en un rango de acción de 0°C a +50°C con un error de +/- 2°C y la humedad relativa entre 20% y 90% con un error de +/-5%. No es un sensor con gran sensibilidad, pero cumple nuestros objetivos sobradamente.

El sensor de temperatura es un termistor tipo NTC. Un termistor es un tipo de resistencia (componente electrónico) cuyo valor varía en función de la temperatura de una forma más acusada que una resistencia común.



Su funcionamiento se basa en la variación de la resistividad que presenta un semiconductor con temperatura.

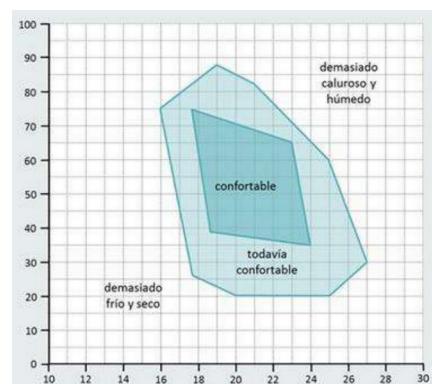
El término proviene del inglés “thermistor”, que es un acrónimo de las palabras Thermally Sensitive Resistor (resistencia sensible a la temperatura).

Hay dos tipos fundamentales de termistores:

- Los que tienen un coeficiente de temperatura negativo (en inglés Negative Temperature Coeficiente o NTC), los cuales decrementan su resistencia a medida que aumenta la temperatura.
- Los que tienen un coeficiente de temperatura positivo (en inglés Positive Temperature Coeficiente o PTC), los cuales incrementan su resistencia a medida que aumenta la temperatura.

### Zona de confort higrotérmico

Puede definirse confort térmico, o más propiamente confort higrotérmico, como la ausencia de malestar térmico. En fisiología, se dice que existe confort higrotérmico cuando no deben intervenir los mecanismos termorreguladores del cuerpo para una actividad sedentaria y con una indumentaria ligera. Esta situación puede registrarse mediante índices que no deben ser sobre pasados para que no se pongan en funcionamiento los sistemas termorreguladores (metabolismo, sudoración y otros).



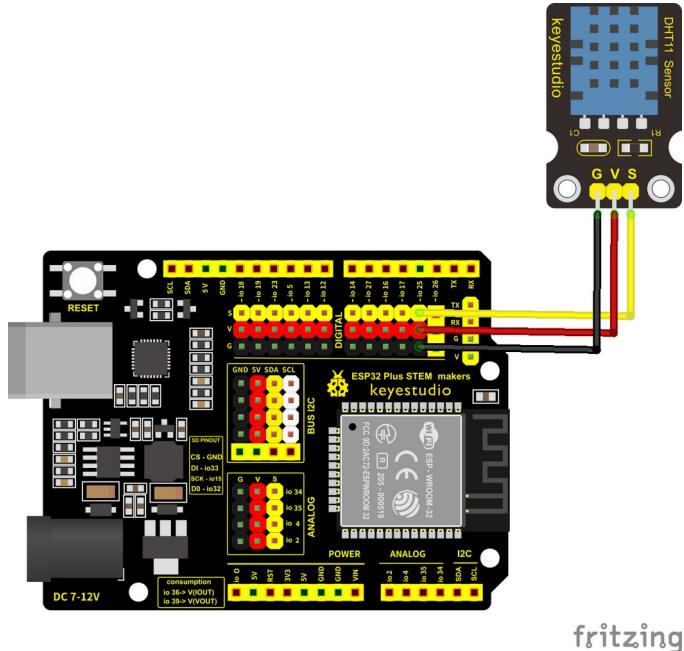
Según la imagen adjunta vamos a marcar unos puntos de temperatura y humedad en los que estaremos dentro de la zona de confort térmico, dentro de una zona de medio confort y fuera de la zona de confort.



## LISTA DE ELEMENTOS

- 1 placa ESP32 STEAMakers
- Sensor DHT11

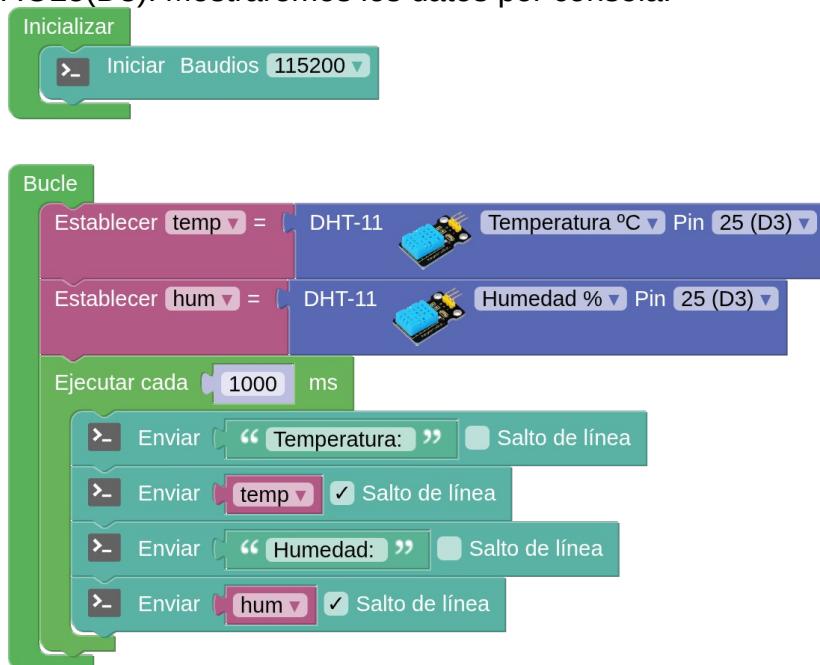
## MONTAJE



fritzing

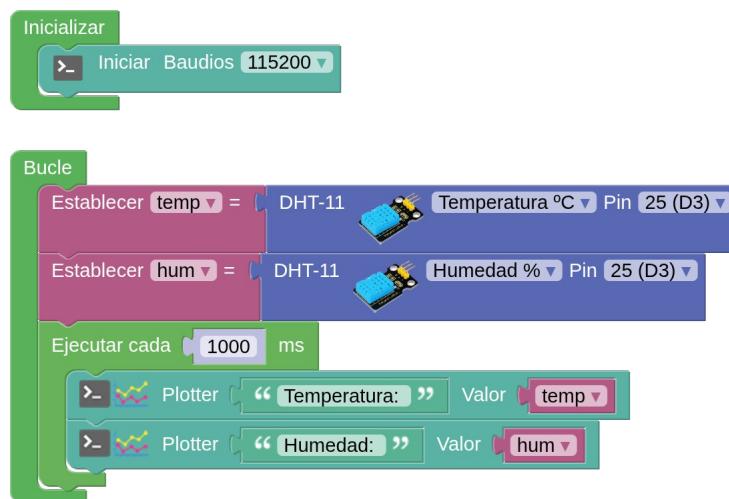
## PROGRAMACIÓN 5.1

En esta actividad controlaremos la temperatura y la humedad con el sensor DHT11 conectado al pin IO25(D3). Mostraremos los datos por consola:



## PROGRAMACIÓN 5.2

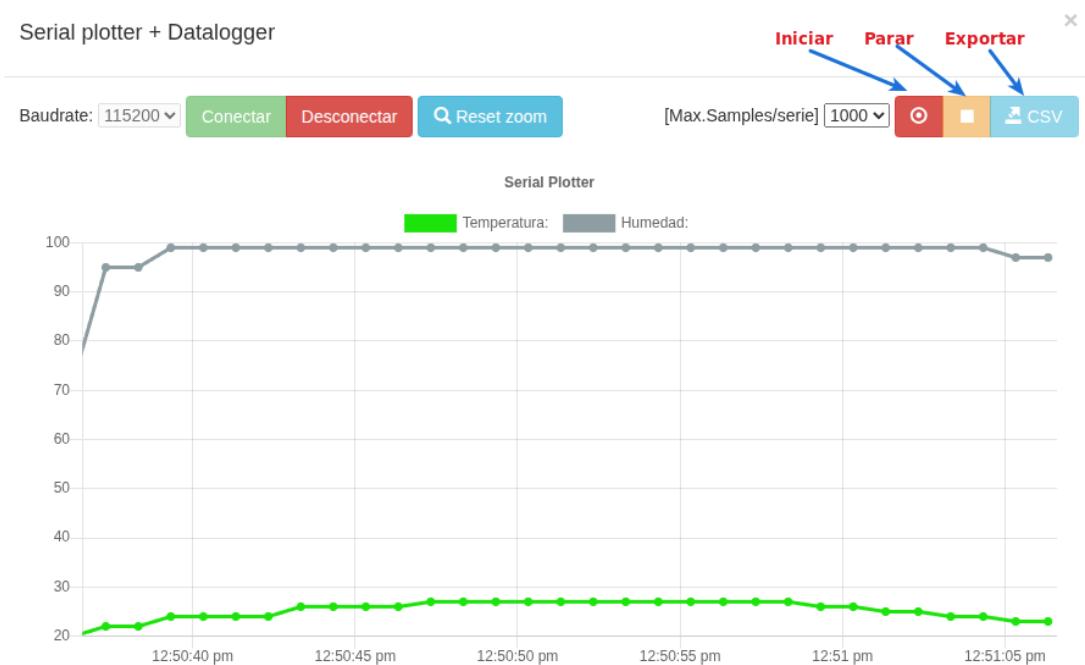
Mostrar los datos de temperatura y humedad en el plotter:



Enviamos el programa a la placa y activamos el Serial Plotter a través del desplegable que hay junto a Consola.



Podemos grabar los datos en formato csv. Para ello debemos iniciar la captura de datos y finalizarla para poder generar los archivos csv de cada variable:



Se creará un archivo .csv para cada variable que grabemos. En este ejemplo se crearán 2 archivos .csv, un archivo con los datos registrados de temperatura y otro con los datos registrados de humedad.



## RETOS DE AMPLIACIÓN

- Semáforo de zona de confort: Definir un rango de confort y diseñar los límites con un led RGB con el que indicaremos estas zonas:
  - Led RGB en ROJO; fuera de la zona de confort.
  - Led RGB en VERDE; en la zona de confort.

Zona VERDE:

- Humedad entre 40% y el 65%.
- Temperatura entre 18°C y 24°C.

El led brillará en VERDE dentro de los parámetros de la Zona VERDE, para el resto el led estará parpadeando en color ROJO.

Para realizar este programa necesitaremos varios bloques del menú de Lógica. Necesitaremos evaluar una condición Lógica y utilizar conjunciones y disyunciones.

- Y: se cumple si ambos operandos son verdaderos.
- O: se cumple si alguno de los dos operandos es verdadero.



## P06. Pantalla OLED

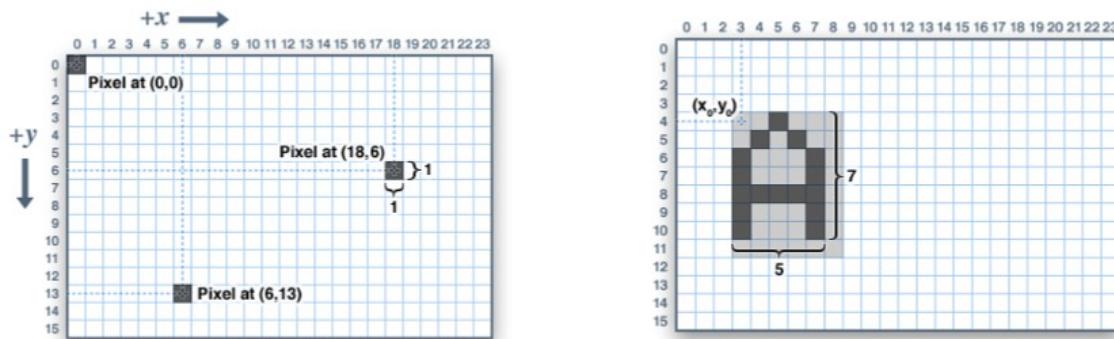
### OBJETIVOS

- Mostrar información por la pantalla OLED en tiempo real.

### CONOCIMIENTOS

La pantalla OLED la conectaremos al puerto de comunicaciones I2C respetando el orden de las conexiones.

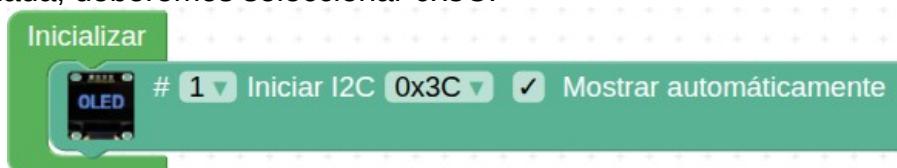
La pantalla que vamos a utilizar es una pantalla OLED de 0,96 pulgadas de 128x64 pixels.



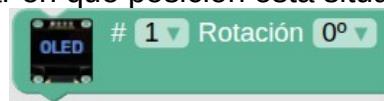
OLED es la abreviatura de diodo emisor de luz orgánico. A nivel microscópico, una pantalla OLED es una matriz de leds que se iluminan cuando emiten energía. La nueva tecnología OLED sólo utiliza electricidad para cada píxel que queramos iluminar. Esto hace que la tecnología OLED sea muy eficiente. Además, la forma en que se construyen este tipo de OLED permite que sean muy delgadas en comparación con la antigua pantalla LCD.

Existen una serie de bloques específicos para controlar la pantalla.

- Primero configuraremos la dirección I2C de la pantalla OLED. En el caso de la pantalla utilizada, deberemos seleccionar 0x3C.



- Rotación: Permite indicar en qué posición está situada la pantalla



- Limpiar: Elimina todo el contenido de la pantalla.



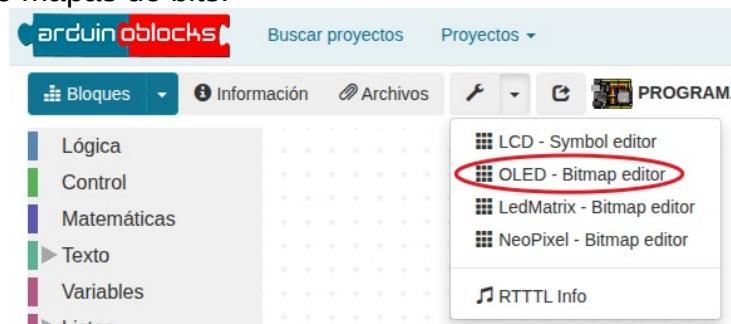
- Texto: Muestra un texto en la posición indicada. Dado que la pantalla es monocroma, podemos indicar si el texto debe mostrarse en positivo (encendiendo pixeles en la pantalla) o en negativo (apagando pixeles). Además podemos especificar 3 tamaños de letra en la pantalla.



- Para sacar una imagen por pantalla lo podemos hacer a través de un mapa de bits, que podemos cargar desde un archivo o editarlo desde el editor de mapas de bits.

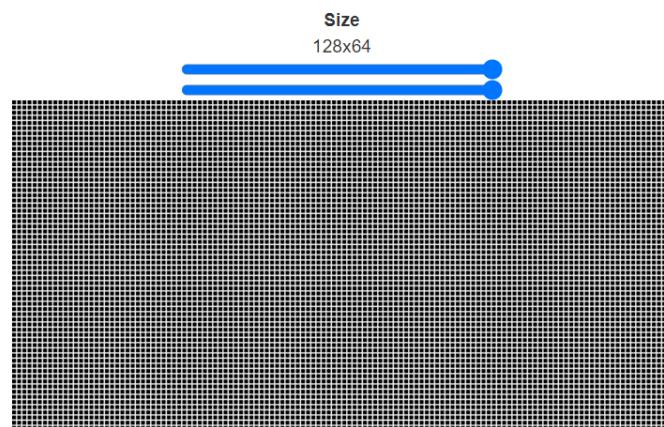


Abrimos el editor de mapas de bits.



- Al final, tanto si lo cargamos desde un archivo como si lo editamos, es necesario copiar y pegar el mapa de bits en el bloque de programación.

## OLED - Bitmap Data



**Clear** **Fill** **Copy data:**

**Elegir archivos** Ningún archivo seleccionado

- Para dibujar un rectángulo debes darle el primer vértice, anchura y altura del rectángulo.



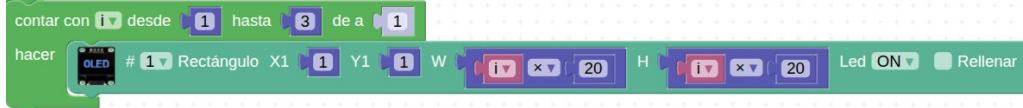
- Para dibujar un círculo debes darle el centro y el radio.



- Para dibujar una línea debes indicarle el punto inicial y el final.



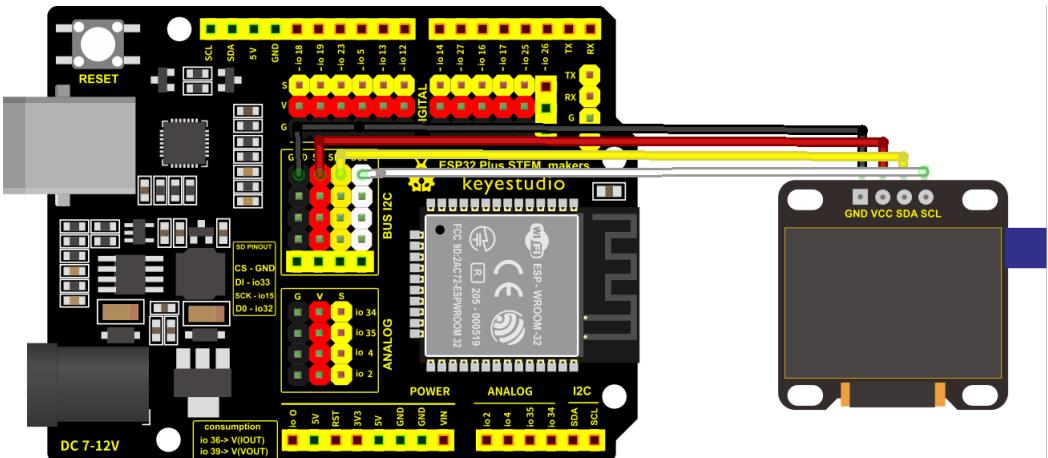
- Para dibujar diferentes figuras con una razón de proporcionalidad podemos utilizar un contador como variable.



## LISTA DE ELEMENTOS

- 1 placa ESP32 STEAMakers
- Pantalla OLED

## MONTAJE

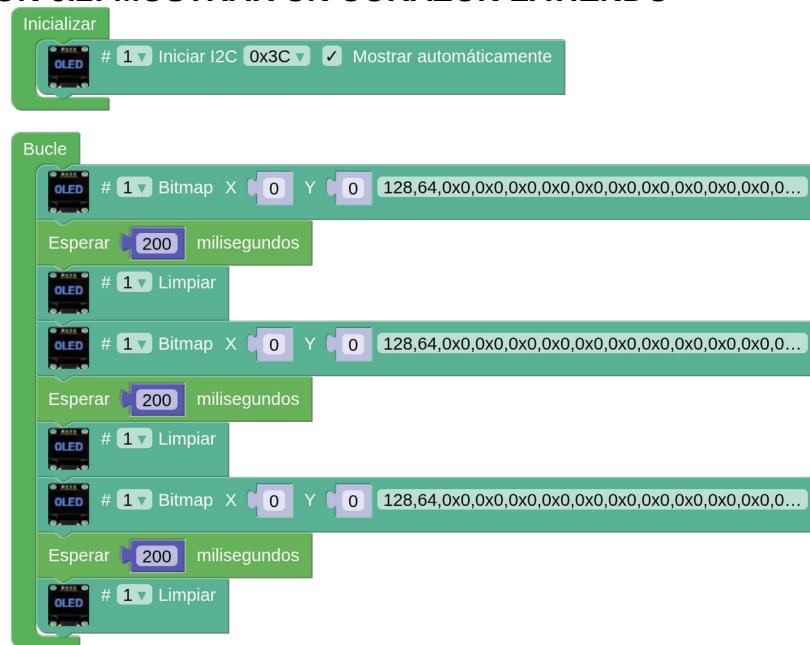


fritzing

## PROGRAMACIÓN 6.1: MOSTRAR UN MENSAJE



## PROGRAMACIÓN 6.2: MOSTRAR UN CORAZÓN LATIENDO



## PROGRAMACIÓN 6.3: DIBUJAR FIGURAS POR COORDENADAS

- Dibujaremos 3 cuadrados de diferentes tamaños que comparten un vértice, 3 círculos concéntricos de diferente radio y un triángulo dados sus vértices.



## **RETOS DE AMPLIACIÓN**

- Con pulsadores, seleccionar la información que queremos ver por pantalla.
  - Mostrar información, utilizando diferentes tamaños de texto.
  - Cargar una imagen Bitmap.
  - Dibujar líneas dirigidas con pulsadores (o con el joystick)
  - Mostrar temperatura y humedad en la pantalla OLED
  - Hacer contador de turnos de una tienda (aumento uno cuando se acciona un pulsador) y mostrarlo por la pantalla.

## P07. Matriz de LEDs 8x8



### OBJETIVOS

- Mostrar información EN la matriz de 8x8 LEDs en tiempo real.

### CONOCIMIENTOS

Las matrices de LEDs permiten controlar 64 LEDs monocromos (normalmente rojos, azules o verdes) dispuestos en forma de matriz de 8 filas y 8 columnas. Para controlar los LEDs la matriz tiene un chip controlador (HT16K33) que gestiona los mismos y se comunica con la placa controladora mediante el bus I2C.

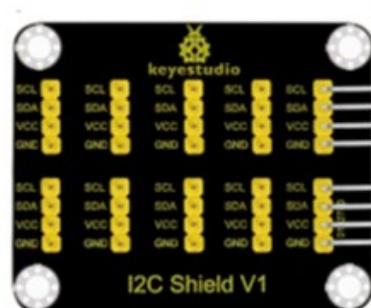
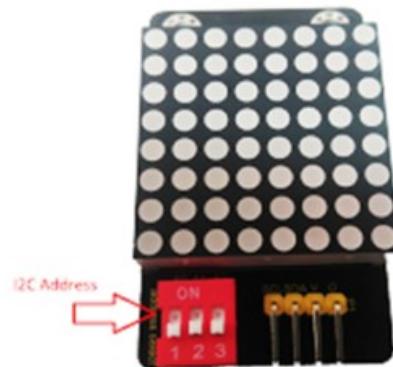
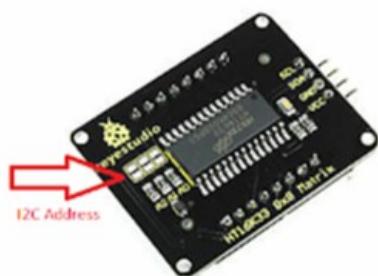


Podemos conectar varias matrices de LEDs en el mismo bus I2C, para controlarlas individualmente debemos cambiar la dirección de cada una de ellas.

Configuración de la dirección I2C mediante pinos soldados

Configuración de la dirección I2C mediante micro-switches

Hub I2C para ramificar y conectar varios dispositivos en el mismo bus I2C



- Iniciar: Inicializa una matriz en una dirección en concreto. Posteriormente podremos referirnos a ella mediante el ID #



- Rotación: Permite ajustar la orientación de la matriz LED.



- Limpiar: Apaga todos los LEDs de la matriz.



- Bitmap: Permite establecer un mapa de bits para dibujar en la matriz led. Mediante la opción "Ayuda" del bloque podemos acceder al editor de mapa de bits.





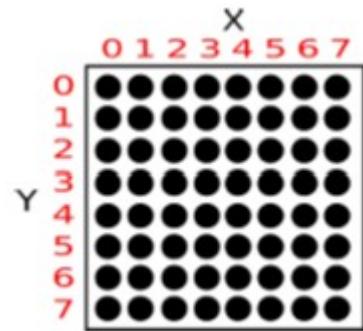
- Bitmaps (predefinidos): Mediante este bloque podemos cargar bitmaps predefinidos de forma muy sencilla.



- Píxel: Enciende o apaga un led en concreto indicándole sus coordenadas.



Sistema de coordenadas de la matriz led 8x8



- Línea: Permite dibujar una línea recta entre dos coordenadas en la matriz led.

Punto origen: X1, Y1

Punto destino: X2, Y2



- Rectángulo: Dibuja un rectángulo indicando las coordenadas de una esquina y la contraria.

Esquina 1: X1, Y1

Esquina 2: X2, Y2



- Círculo: Dibuja un círculo indicando las coordenadas del centro y el número de píxeles de radio.

Centro: X, Y

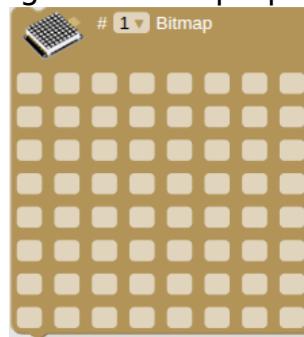
Radio: R



- Texto: Dibuja un texto en la matriz led. Utilizando coordenadas fuera de las visibles, podemos realizar textos con desplazamiento con una única matriz led.



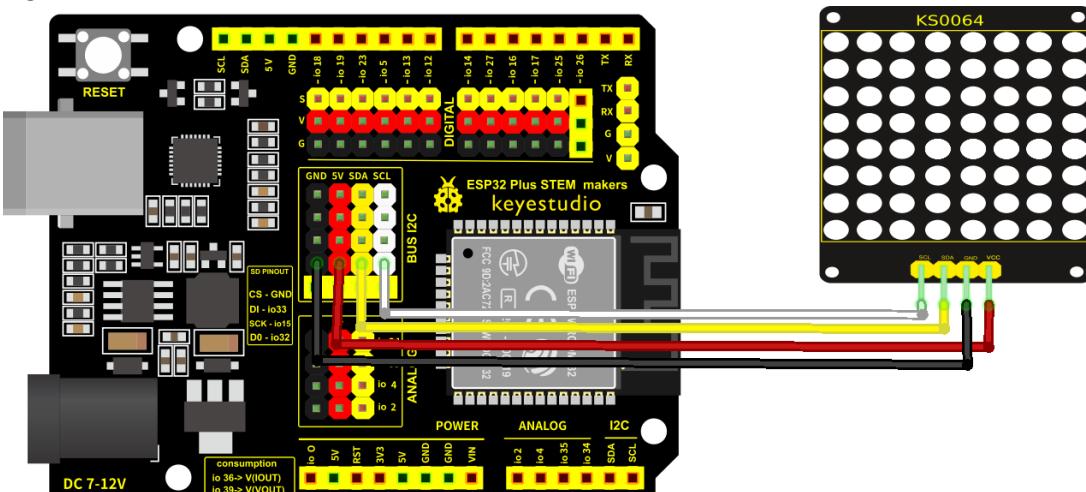
- Bitmap: Permite definir un mapa de bits directamente marcando los píxeles encendidos y apagados en el propio bloque de una forma visual.



## LISTA DE ELEMENTOS

- 1 placa ESP32 STEAMakers
- Matriz de LEDs 8x8
- Pulsador

## MONTAJE

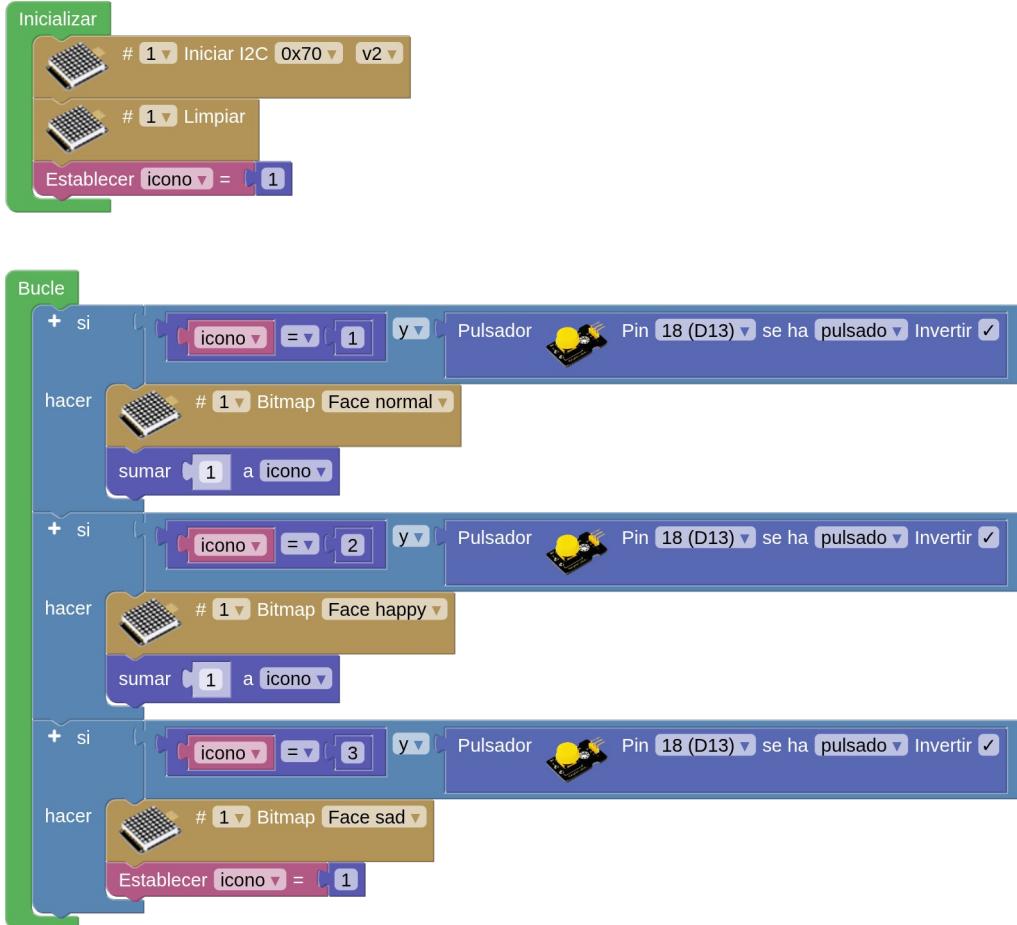


fritzing



## PROGRAMACIÓN 7.1

Haremos un programa que vaya mostrando distintos iconos. Cada vez que pulsamos el pulsador la matriz LED muestra un ícono diferente.



## RETOS DE AMPLIACIÓN

- Dibujar figuras sencillas.
- Mostrar un mensaje de texto en movimiento



## P08. Lista de reproducción con el zumbador



### OBJETIVOS

- Crear una lista de reproducción de música.

### CONOCIMIENTOS

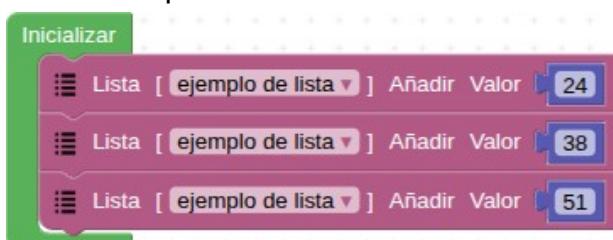
- El zumbador (*Buzzer* en inglés) es un transductor electroacústico que produce un sonido o zumbido continuo o intermitente. En función de si se trata de un zumbador Activo o Pasivo, este zumbido será del mismo tono o lo podremos variar. Sirve como mecanismo de señalización o aviso y se utiliza en múltiples sistemas, como en automóviles o en electrodomésticos.

Para controlar el zumbador pasivo y hacer que suenen diferentes notas, necesitamos introducir ondas cuadradas de diferentes frecuencias en el polo positivo del componente y poner el polo negativo a tierra, para conectarlo a un pin digital de la placa.

Lo podemos controlar reproduciendo un sonido a cierta frecuencia y durante un cierto tiempo, o bien, mediante el bloque de reproducir una melodía RTTTL:

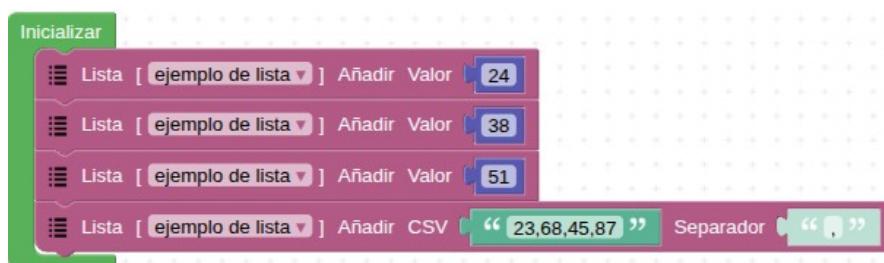


- Las listas de datos nos permiten almacenar un listado de valores y acceder a ellos por su posición en la lista. Las listas pueden ser de tipo numérico o de texto. La creamos dándole un nombre y entonces podemos ir añadiendo elementos en cualquier momento.



Posición	0	1	2
Valor	24	38	51

Aquí hemos creado una lista de tipos numérica con 3 valores, ordenados como se ve en la tabla. Con las listas dinámicas podemos añadir elementos con formato CSV separados por comas:



Así la lista anterior de ejemplo quedaría de la siguiente forma:

Posición	0	1	2	3	4	5	6
Valor	24	38	51	23	68	45	87

En una lista podemos obtener el valor de una posición (desde la 0 hasta el número de elementos -1 en la lista) con el siguiente bloque (en el ejemplo tenemos 7 elementos, podemos acceder a ellos del 0 al 6):



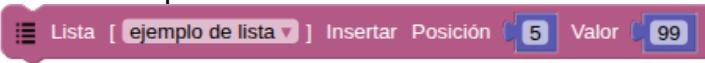
Para saber el número de elementos que tenemos en una lista podemos usar el bloque “Número de elementos”.



Para modificar un elemento de la lista sin alterar el resto:



Para añadir un elemento en un punto determinado de la lista:



Con este último bloque se desplazan los elementos a partir del nuevo una posición hacia arriba. Vemos cómo quedaría la lista de ejemplo:

Posición	0	1	2	3	4	5	6	7
Valor	24	38	51	23	68	99	45	87

→ Elemento nuevo  
→ Elemento desplazado

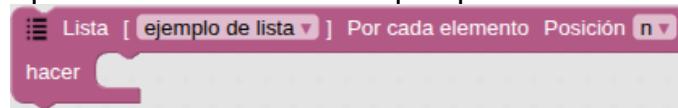
Y la acción contraria que sería eliminar un elemento de la lista, con el consecuente desplazamiento hacia atrás de los siguientes elementos a la posición eliminada.



Posición	0	1	2	3	4	5	6
Valor	24	38	23	68	99	45	87

→ Elementos desplazados

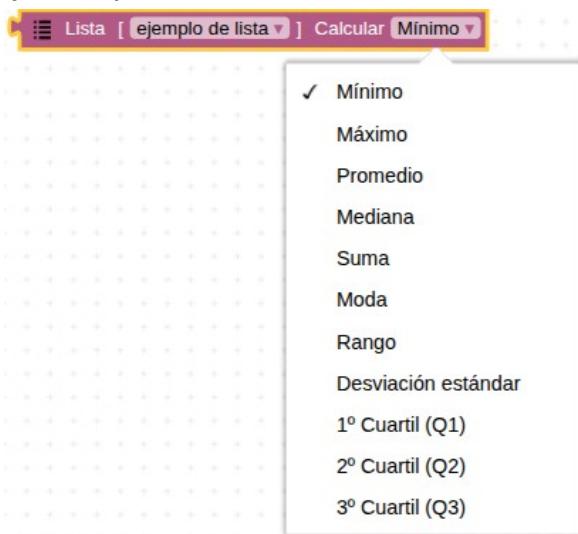
Para recorrer una lista podemos utilizar este bloque que crea un bucle local:



Con este bloque podemos ordenar los elementos de la lista de menor a mayor, o a la inversa:



También tenemos la posibilidad de obtener algunos valores estadísticos de la lista creada con el bloque “calcular” y sus opciones:



En la tabla siguiente se hace un breve resumen de los conceptos estadísticos que se pueden calcular.

<b>Tipo</b>	<b>Descripción</b>	<b>Ejemplo</b>	<b>Resultado</b>
Mínimo	Valor más pequeño de un conjunto.	1, 2, 2, 3, 4, 7, 9	<b>1</b>
Máximo	Valor más grande de un conjunto.	1, 2, 2, 3, 4, 7, 9	<b>9</b>
Promedio	La media o promedio de un conjunto de datos se encuentra al sumar todos los números en el conjunto de datos y luego al dividir entre el número de valores en el conjunto.	1, 2, 2, 3, 4, 7, 9 28/7	<b>4</b>
Mediana	Valor medio que separa las mitades mayores y menores de un conjunto de datos	1, 2, 2, <b>3</b> , 4, 7, 9	<b>3</b>
Suma	La suma de los números de un conjunto.	1, 2, 2, 3, 4, 7, 9	<b>28</b>
Moda	Valor más frecuente en un conjunto de datos.	1, <b>2</b> , <b>2</b> , 3, 4, 7, 9	<b>2</b>
Rango	La media aritmética de los valores más altos y más bajos de un conjunto	(1 + 9) / 2	<b>5</b>
Desviación estándar	La variación o dispersión en la que los datos individuales difieren de la media.	1, 2, 2, 3, 4, 7, 9 $\sqrt{52/6}$	<b>2.94</b>
Suma = $(1-4)^2 + (2-4)^2 + (2-4)^2 + (3-4)^2 + (7-4)^2 + (9-4)^2 = 9+4+4+1+9+25 = 52$ ; Elementos – 1= 7 - 1 = 6			
Primer cuartil (Q1)	Los cuartiles son tres valores que dividen una muestra en, aproximadamente, 4 partes iguales cuando los datos están ordenados de menor a mayor. 1, 2, 2, 3, 4, 7, 9	Posición = $(n+1)/4 = (7+1)/4 = 2$	<b>2</b>
Segundo cuartil (Q2)		Posición = $(n+1)/2 = (7+1)/2 = 4$	<b>3</b>
Tercer cuartil (Q3)		Posición = $3(n+1)/4 = 3(7+1)/4 = 6$	<b>7</b>

Por último, podemos eliminar toda la lista con el siguiente bloque:



Esto resulta importante para liberar el espacio de memoria de la placa cuando no sea estrictamente necesario.

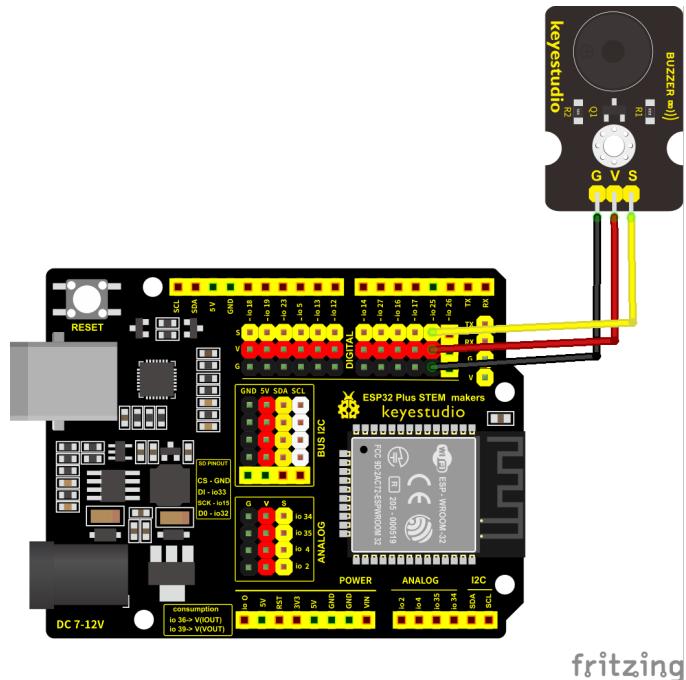
Estos bloques también están disponibles para las listas de texto, en color verde, con algunas funciones específicas diferentes como ordenar alfabéticamente en lugar de ordenar de mayor a menor, y sin la parte estadística.



## LISTA DE ELEMENTOS

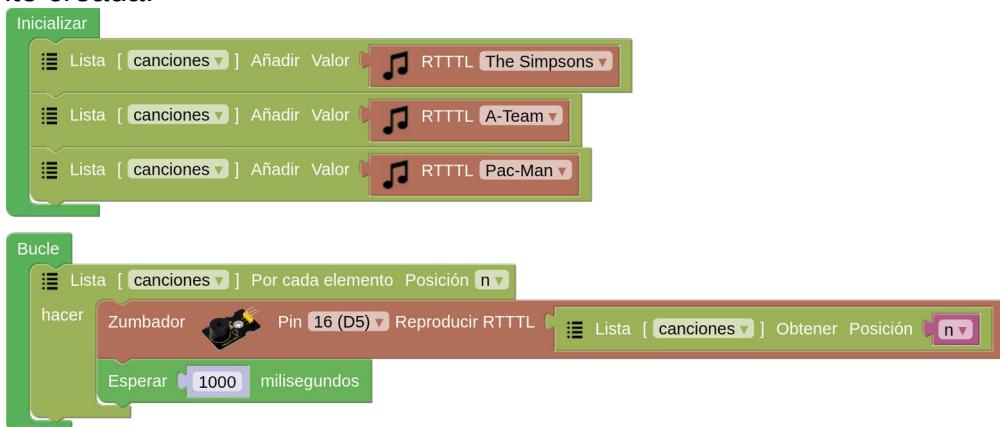
- 1 placa ESP32 STEAMakers
- Zumbador

## MONTAJE



## PROGRAMACIÓN 8.1

Queremos programar la placa y el zumbador para que reproduzcan una lista de canciones previamente creada:



## RETOS DE AMPLIACIÓN

- Hacer que no suene la canción, o la canción siguiente, hasta que se pulse el pulsador.
- Que salga por la pantalla OLED, el nombre de la canción



## P09. Medir distancias con el sensor de ultrasonidos



### OBJETIVOS

- Mostrar la distancia desde el sensor hasta un obstáculo en la consola serie.
- Mostrar la distancia desde el sensor hasta un obstáculo en una pantalla OLED.
- Activar una alarma cuando se detecta un objeto a menos de 10 cm.

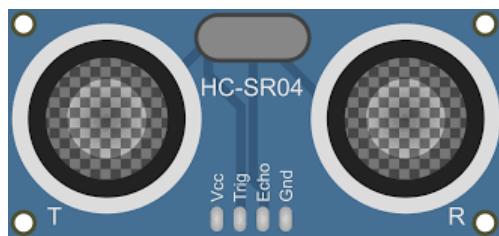
### CONOCIMIENTOS

- Sensor de distancia por ultrasonidos:

El sensor de distancia HC-SR04 utiliza ultrasonidos para medir la distancia a un objeto.

Funciona de la siguiente manera:

1. Enviar la Señal: El sensor envía un pulso ultrasónico mediante el pin Trig (trigger, que significa disparo).
2. Recibir el Eco: Cuando el pulso choca contra un objeto, se refleja y vuelve al sensor. El pin Echo detecta ese eco.
3. Medir el Tiempo: El sensor mide el tiempo que tarda el eco en volver.
4. Calcular la Distancia: Utiliza el tiempo medido para calcular la distancia al objeto.

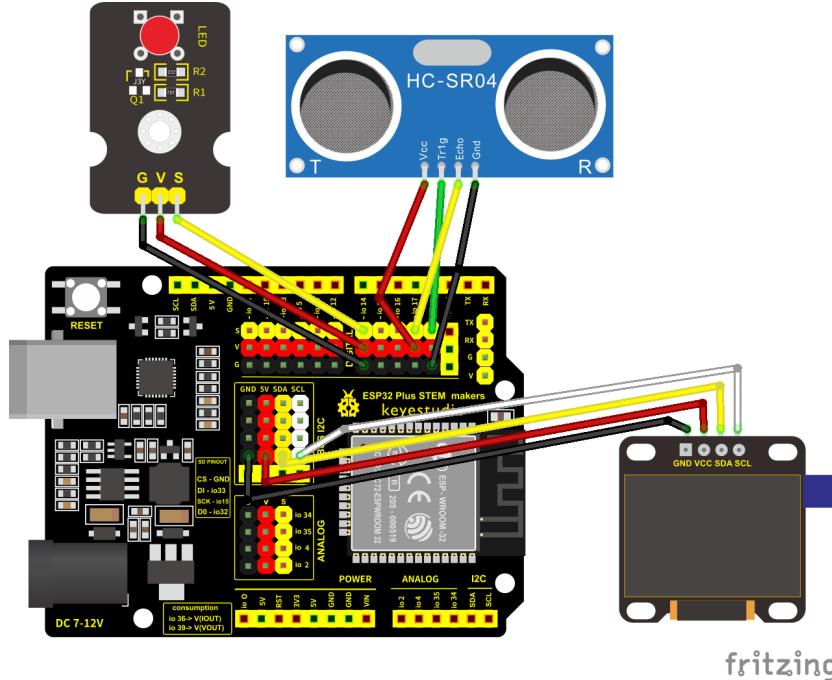


### LISTA DE ELEMENTOS

- 1 placa ESP32 STEAMakers
- Sensor ultrasonidos HC-SR04
- Pantalla OLED
- LED

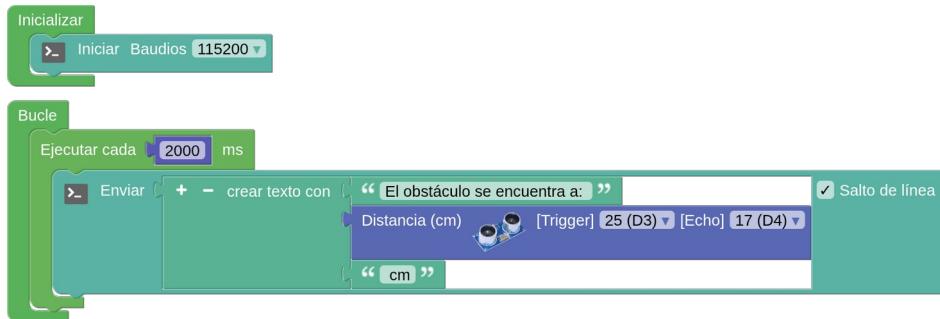


## MONTAJE



fritzing

## PROGRAMACIÓN 9.1



Debemos abrir la consola para poder verificar la distancia a la que se encuentra el obstáculo.

Consola serie

Baudrate: 115200 ▾ Conectar Desconectar Limpiar

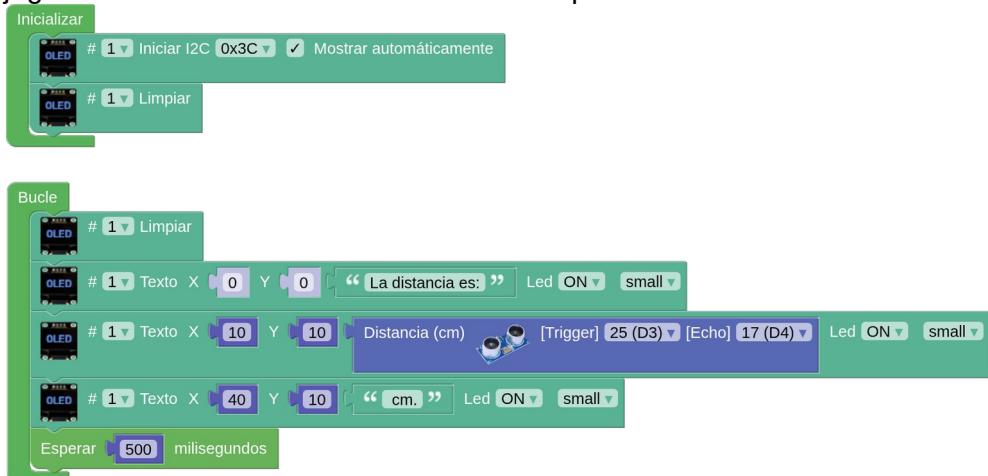
Enviar

El obstáculo se encuentra a: 180.00 cm  
El obstáculo se encuentra a: 11.00 cm  
El obstáculo se encuentra a: 29.00 cm  
El obstáculo se encuentra a: 33.00 cm  
El obstáculo se encuentra a: 51.00 cm  
El obstáculo se encuentra a: 63.00 cm

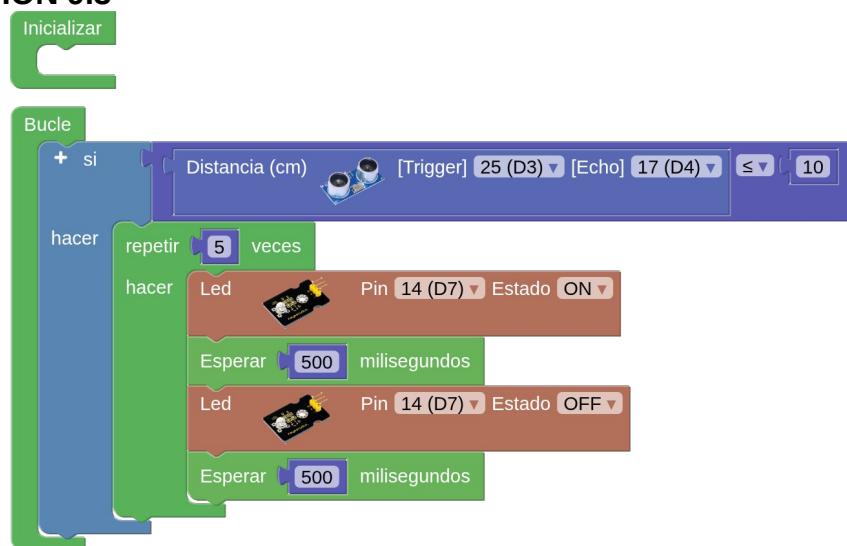


## PROGRAMACIÓN 9.2

Para presentar la información en la pantalla OLED deberemos iniciarla. Recordemos que sólo acepta mensajes de texto en inglés, por lo que no acepta acentos. Si desea que la letra sea mayor tendrá que jugar con los valores de coordenadas de la pantalla OLED.



## PROGRAMACIÓN 9.3



## RETOS DE AMPLIACIÓN

- Alarma de proximidad acústica: Crear una alarma que se active cuando alguien se acerca demasiado al sensor de ultrasonidos. La alarma emitirá un sonido que aumentará de frecuencia a medida que la persona se acerque más al sensor, advirtiendo así de su proximidad. Sería el típico ejemplo de cuándo te acercas mucho a una obra de arte en un museo y se genera una alarma sonora por proximidad.
- Theremin: Crear un Theremin sencillo utilizando un sensor de ultrasonidos y un zumbador. Theremin es un instrumento musical electrónico que se toca sin contacto físico, y en este proyecto la distancia entre la mano y el sensor de ultrasonidos determina la frecuencia del sonido producido por el zumbador.
- Medidor de altura: Medir de manera estimativa a los alumnos posicionando el sensor de ultrasonidos.
- Identificador de alturas: imagina que desea saber quién está delante de la puerta, si encuentra a una persona de estatura pequeña que el sistema ponga una canción infantil, si encuentra a una persona de mayor altura que ponga una canción diferente.

## P10. Controlar un servomotor

### OBJETIVOS

- Controlar el movimiento de un servomotor.

### CONOCIMIENTOS

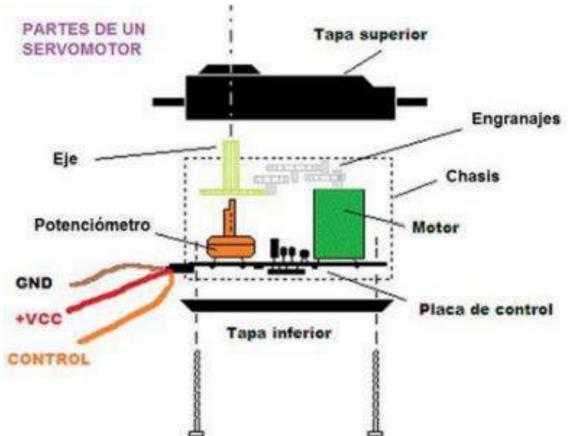
Un SERVO es un tipo de motor utilizado normalmente en electrónica. Son cómodos de utilizar, ya que podemos indicar directamente el ángulo de rotación deseado (suelen tener un ángulo de giro entre 0 a 180°).

- Por un lado, nos permite mantener la posición que indicamos, siempre que esté dentro del rango de operación del dispositivo.
- Por otra parte nos permite controlar la velocidad de giro, podemos hacer que antes de que se mueva a la siguiente posición espere un cierto tiempo.

Internamente, un servo está constituido por un motor de corriente continua, acoplado a una reductora para reducir la velocidad de giro, junto con la electrónica necesaria para controlar su posición.

Normalmente, se dispone de un potenciómetro unido al eje del servo, que permite al mismo conocer la posición del eje. Esta información la trata un controlador integrado que se encarga de ajustar y actuar sobre el motor para alcanzar la posición deseada.

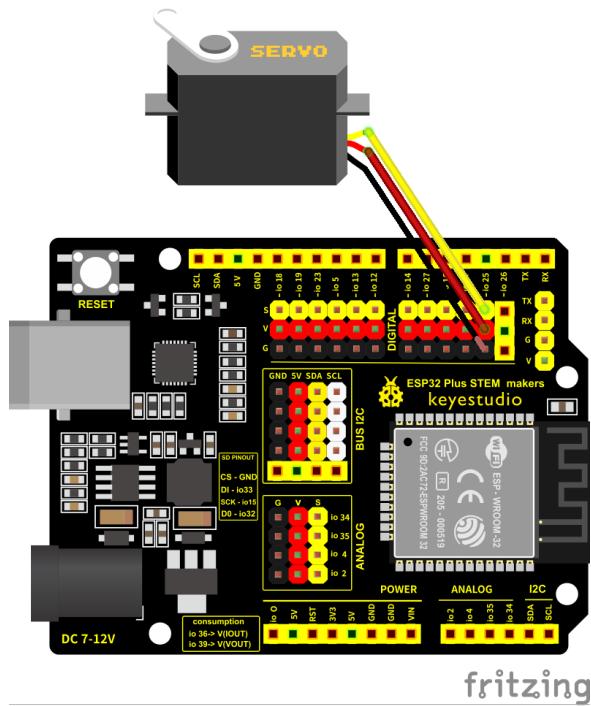
Conectar un servo es sencillo. El servo dispone de tres cables, dos de alimentación (GND y +VCC) y uno de señal de control.



### LISTA DE ELEMENTOS

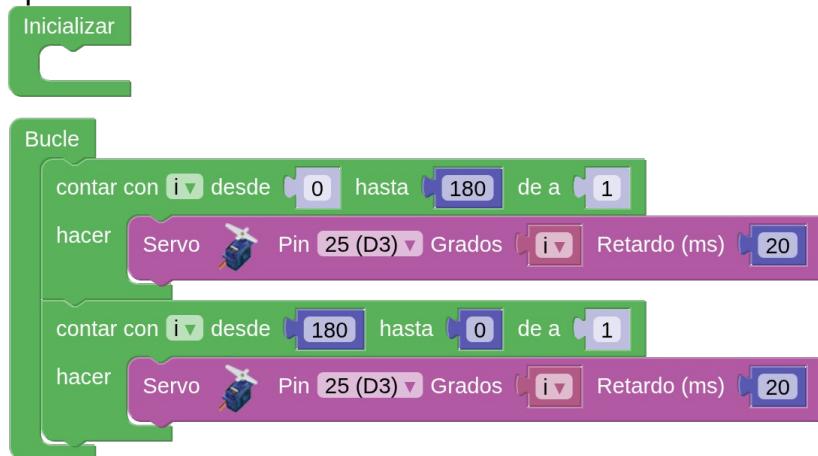
- 1 placa ESP32 STEAMakers
- 1 servomotor

## MONTAJE



## PROGRAMACIÓN 2.1: PULSADOR

Generar un movimiento continuo del servo desde la posición 0º hasta 180º y volver del mismo modo a la posición 0º.



## RETOS DE AMPLIACIÓN

Mover el servomotor 10º cada vez que pulsamos el pulsador.

- Si el motor llega a 180º, cada vez que pulsamos el pulsador disminuiremos 10 grados en la posición.
- Si el motor llega a 0º, cada vez que pulsamos el pulsador aumentaremos 10 grados la posición.



## P11. Tira de LEDs NeoPixel

1

### OBJETIVOS

- Encender progresivamente los LEDs de la tira
- Crear el efecto del coche fantástico
- Programación: listas, bucles y contadores.

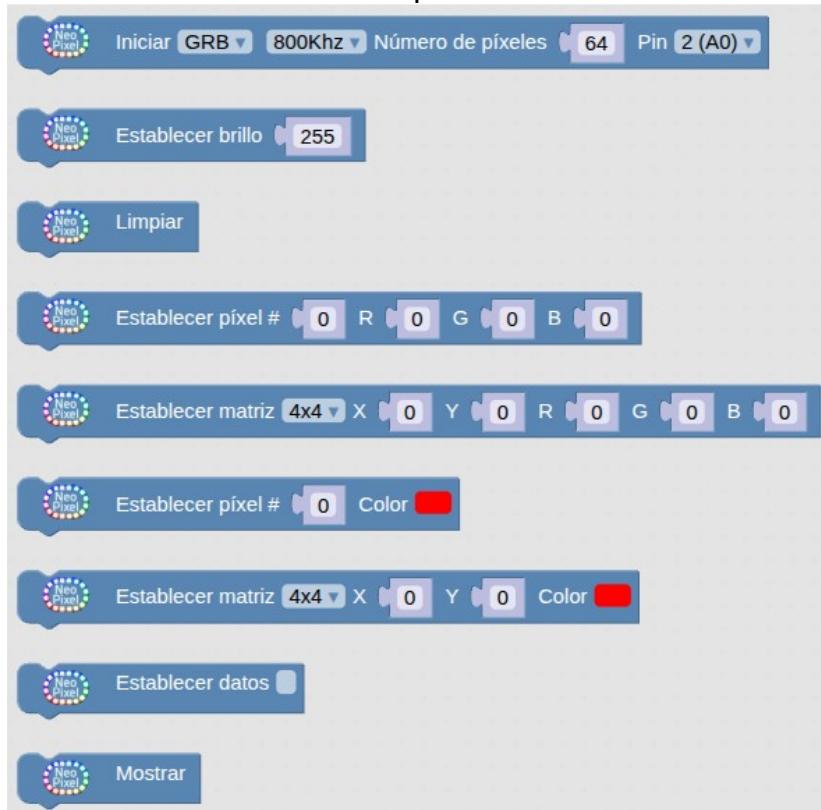
### CONOCIMIENTOS

Los NeoPixel son LEDs RGB inteligentes que incorporan su propio microcontrolador dentro de cada LED. Normalmente, se encuentran en forma de tira de LEDs conectados uno tras otro, pudiendo haber tiras de decenas o cientos de LEDs.

Un LED NeoPixel tiene 4 pines de contacto:

- VCC: alimentación a 3,3 V o 5 V
- GND: 0 V
- DIN (DI): Entrada de información. Conectado a un pin de salida digital.
- DOUT (DO): Salida de información. Conectado al siguiente LED o tira de LEDs.

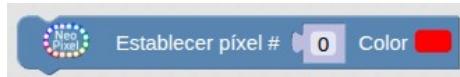
Los bloques de control se encuentran en el apartado de “Visualización” → NeoPixel



Para inicializar la tira, disponemos del bloque donde se define el número de LEDs (píxeles) y el pin al que la conectamos:



Define el color de un píxel indicando el número de LED según su posición. De 0 en adelante:



Para apagar todos los píxeles que estén conectados:



Es necesario actualizar los datos enviados a NeoPixel, ya que las operaciones no se reflejan hasta que se ejecuta el bloque “mostrar”.



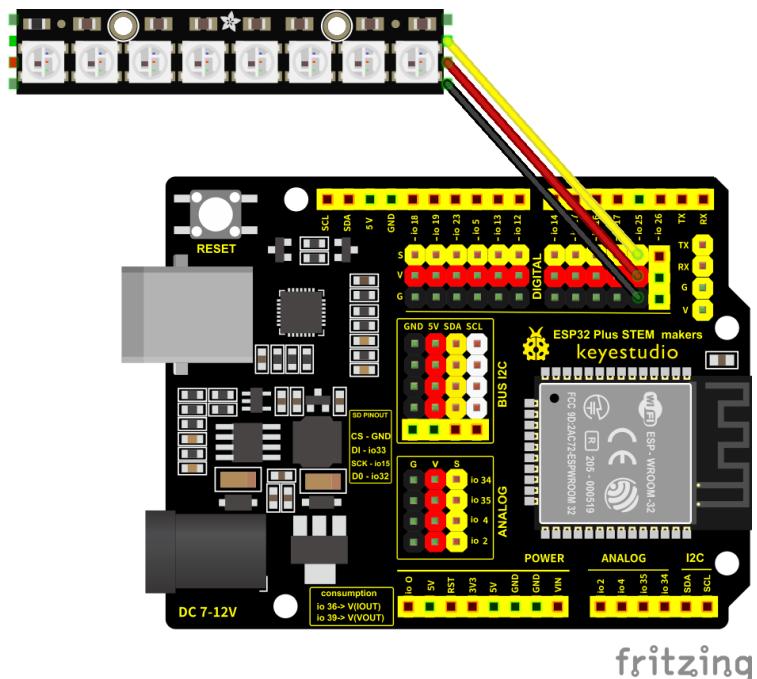
- Contar: Realiza un bucle contando con una variable índice “i”. Se define un valor de inicio, un valor de fin y los incrementos que se realizarán en cada iteración del bucle. Dentro del bucle podremos utilizar esta variable “i”, que arduinoblocks crea automáticamente.



## LISTA DE ELEMENTOS

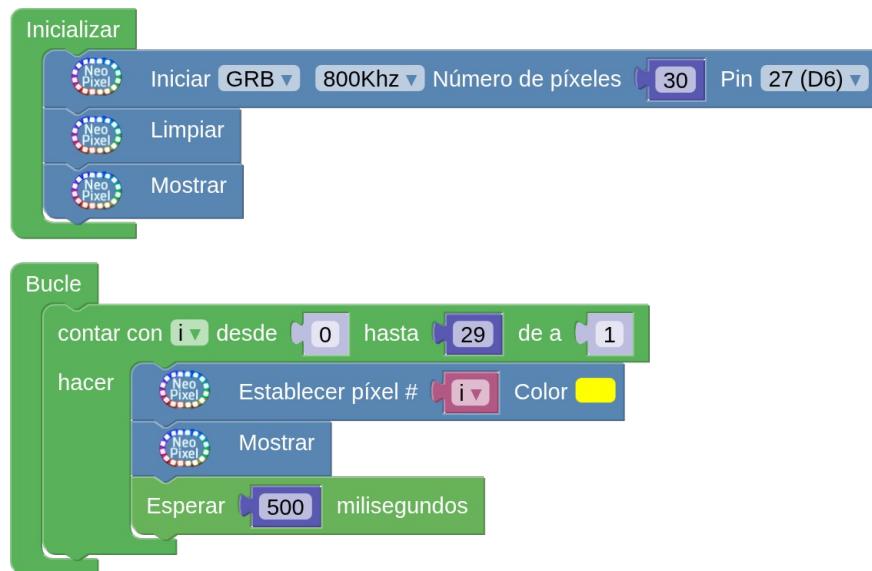
- 1 placa ESP32 STEAMakers
- 1 Tira de LEDs NeoPixel

## MONTAJE



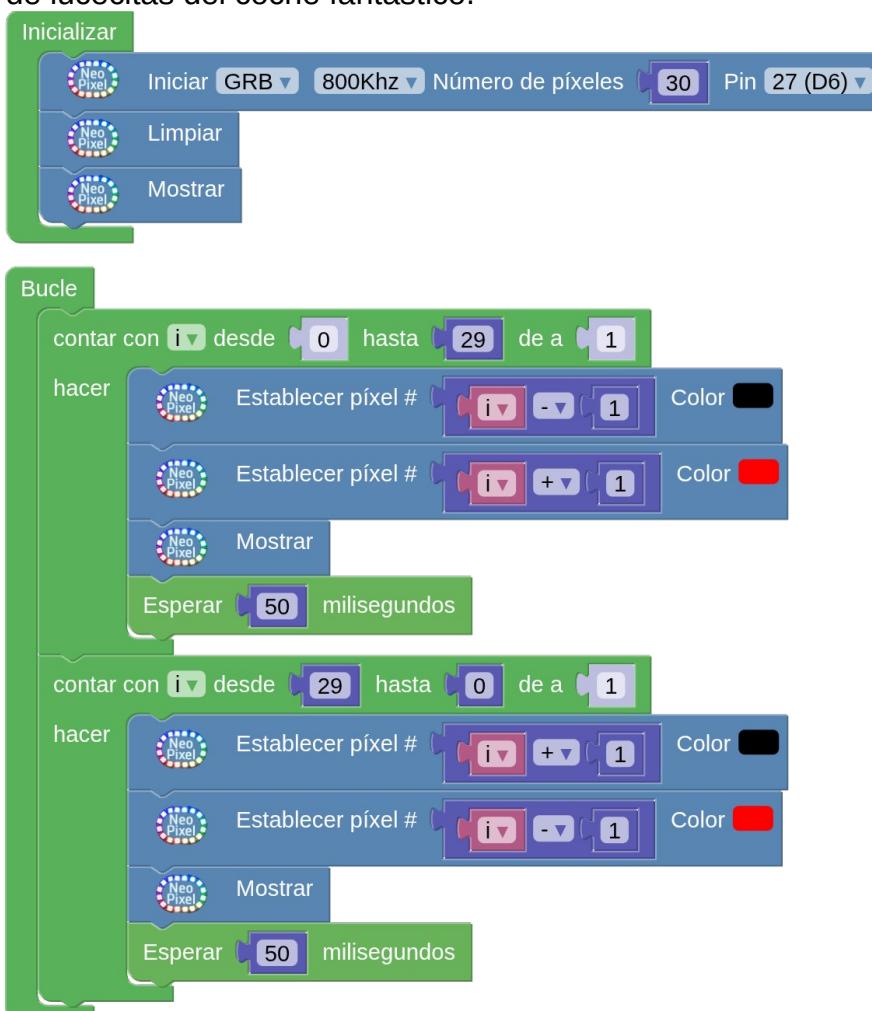
## PROGRAMACIÓN 11.1

Intentamos poner en marcha los LEDs de la tira NeoPixel de forma secuencial utilizando un contador.



## PROGRAMACIÓN 11.2

Simular la tira de luces del coche fantástico.



## RETOS DE AMPLIACIÓN

- Encender los LEDs, empezando por el central e ir hacia los extremos.
- Semáforo de ruido: Hacer que la tira de LEDs se ponga en marcha de un color u otro, en función del ruido. Utilizaremos el sensor de sonido, que debe conectarse a una entrada analógica, por ejemplo el IO25/D3, y funciona con el bloque de leer la entrada analógica.

Establecer Sonido = Leer analógica Pin 25 (D3)

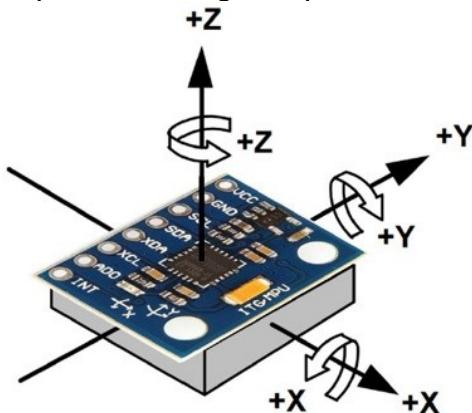


**OBJETIVOS**

- Representación gráfica del ángulo de giro.
- Control de los LEDs de una tira en función del ángulo de inclinación del dispositivo. Según la inclinación de la placa que se pongan en marcha los LEDs de la tira.
- Programación:
  - Funciones
  - Mapear

**CONOCIMIENTOS**

Un giroscopio es un elemento que mide los giros que se realizan en cada uno de los ejes.



- Acelerómetro: es un sensor que mide la aceleración. Esta aceleración puede ser la causada por el movimiento o gravedad. En otras palabras, un acelerómetro puede decirte lo rápido que algo está cambiando su velocidad o dirección, así como la orientación del objeto respecto a la Tierra.
- Función:  
Una función es un bloque de código que ejecuta una tarea específica y que puede ser reutilizado varias veces dentro del programa. Las funciones ayudan a organizar el código en partes más pequeñas y manejables, permitiendo así una programación más clara y eficiente.



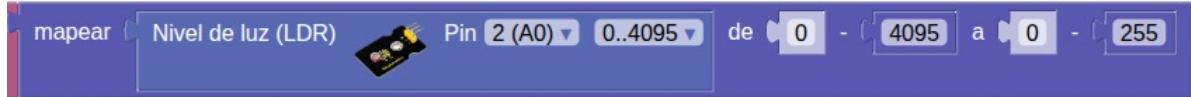
Cambiamos el nombre de la función. Y aparecerá bajo estos dos bloques la función que acabamos de declarar.

- Bloque mapear:  
El bloque mapear convierte un valor de un rango a otro rango. Esto es útil para ajustar los valores de sensores y actuadores para que trabajen correctamente juntos.

## Parámetros del Bloque:

- Valor: El valor que quieras convertir.
- De Min: El valor mínimo del rango original.
- De Max: El valor máximo del rango original.
- A Min: El valor mínimo del nuevo rango.
- A Max: El valor máximo del nuevo rango.

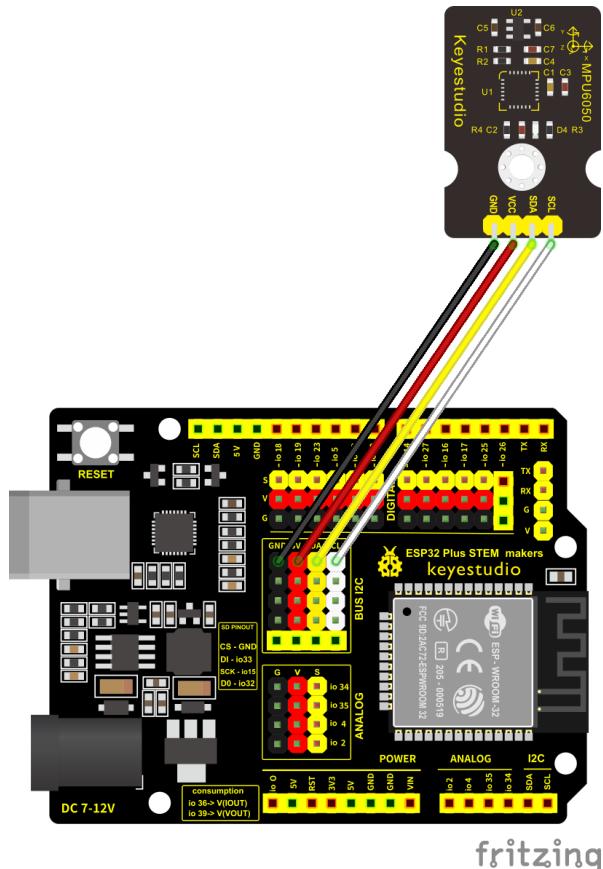
Ejemplo: Si un sensor lee valores entre 0 y 4095 y quieres controlar un LED que acepta valores entre 0 y 255, puedes utilizar el bloque mapear para convertir los valores.



## LISTA DE ELEMENTOS

- 1 placa ESP32 STEAMakers
- Giroscopio
- Tira de 30 LEDs NeoPixel

## MONTAJE



## PROGRAMACIÓN 12.1

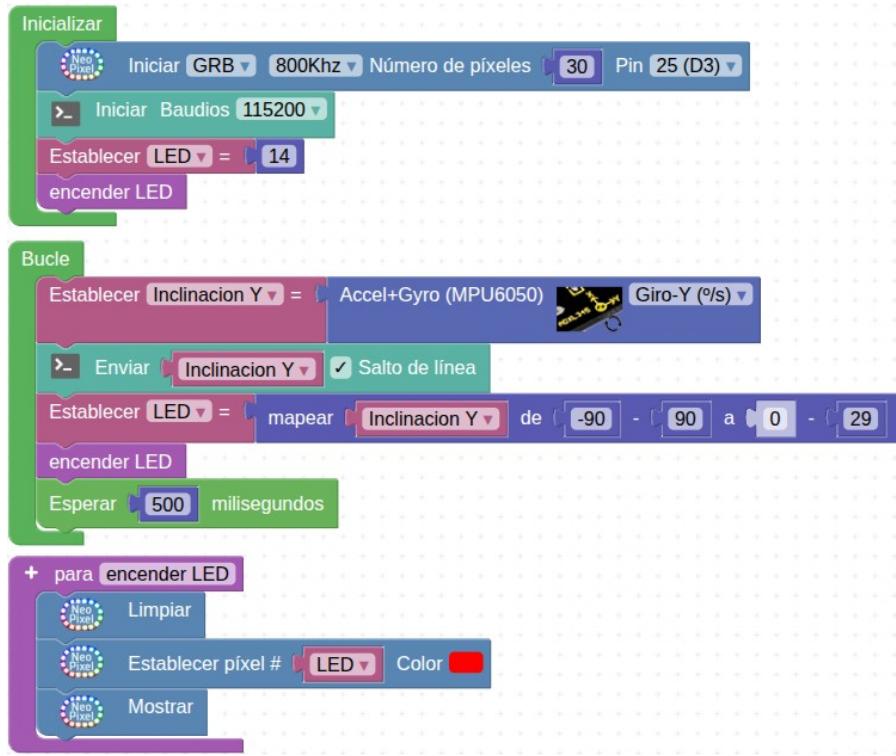
Esta primera práctica nos permite entender el funcionamiento del giroscopio. Observamos que cuando giramos la placa ésta cambia y genera un pico positivo o negativo dependiendo del ángulo de giro de la placa, y rápidamente, se estabiliza. Es muy importante que el tiempo de espera sea pequeño para captar bien la señal.



## PROGRAMACIÓN 12.2

Este programa introduce las funciones y el mapeado. Obtenemos del giroscopio el valor de la inclinación en el eje Y. Para verificar su funcionamiento, hemos utilizado el puerto serie que nos permite ver los datos que está captando, para hacernos idea de los valores. A consecuencia de sus valores observamos que sería bueno realizar un mapeado para obtener unos valores más adecuados para ponerlos en la tira de NeoPixels. El margen de trabajo es de -90 a +90 grados en el entrenador, como disponemos de una tira de 30 NeoPixels, lo escalamos a esta medida. Si dispone de una tira de NeoPixels sin cortar, coloque sobre el 14 o 15 el centro para probar su estabilidad.

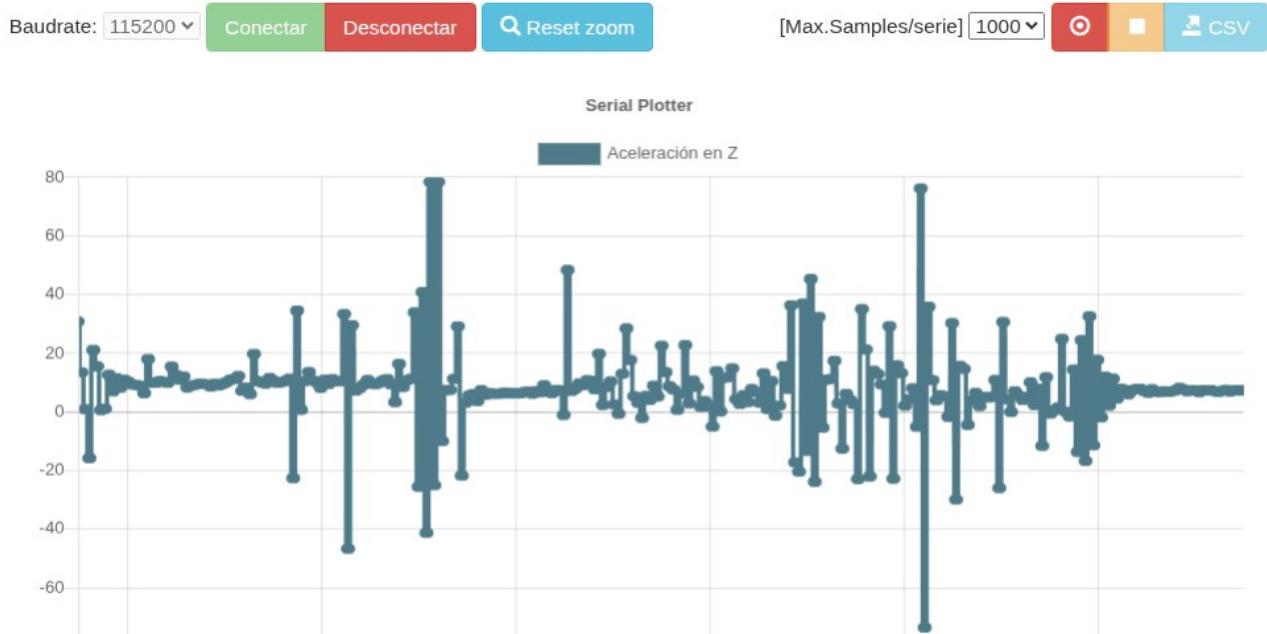




## PROGRAMACIÓN 12.3

Para representar gráficamente los valores del acelerómetro en el eje Z (arriba y abajo) utilizamos la consola del puerto serie.





## RETOS DE AMPLIACIÓN

- Detectar saltos: para realizar una carrera de saltos en un tiempo determinado, cada salto permite utilizar la tira de LEDs para visualizar el total de saltos.
- Podómetro sencillo que cuente los pasos.
- Detector de caída.
- Control de motor: Utilizar el giroscopio para controlar un motor, que podría servir para controlar la dirección (volante) de un coche.
- Detector de movimiento de un objeto o de una persona.



## P13. Bomba de agua



### OBJETIVOS

- Conectar la bomba de agua dependiendo de la humedad del suelo.

### CONOCIMIENTOS

La bomba de agua es un motor estanco de 3.3V que normalmente se combina con el relé que lo abre o lo cierra según las condiciones dadas.



Un relé es un aparato o dispositivo destinado a controlar un circuito determinado de una cierta potencia (generalmente cerrar o abrir el circuito) mediante un circuito auxiliar de pequeña potencia y, por tanto, con una corriente muy débil.



El circuito tiene tres pines NO (Normally Open), COM (COMmon contact) y NC (Normally Closed):

1. NO: este pin está normalmente abierto, y se conecta a la placa ESP32 al pin de corriente de 3.3V
2. COM: este pin es el que se conecta con los otros módulos que queremos controlar, en nuestro caso, el rojo de la bomba de agua.
3. NC: este pin normalmente está cerrado y es el que utilizaremos con el ESP32 para cerrar el circuito.

El circuito auxiliar es el que conectamos a la placa ESP32 para transmitir la apertura/cierre del circuito.

El sensor de humedad de suelo se utiliza para medir el contenido de agua en el suelo, controlar la humedad del suelo, regar los cultivos y proteger los bosques. Es un sensor de tipo analógico.

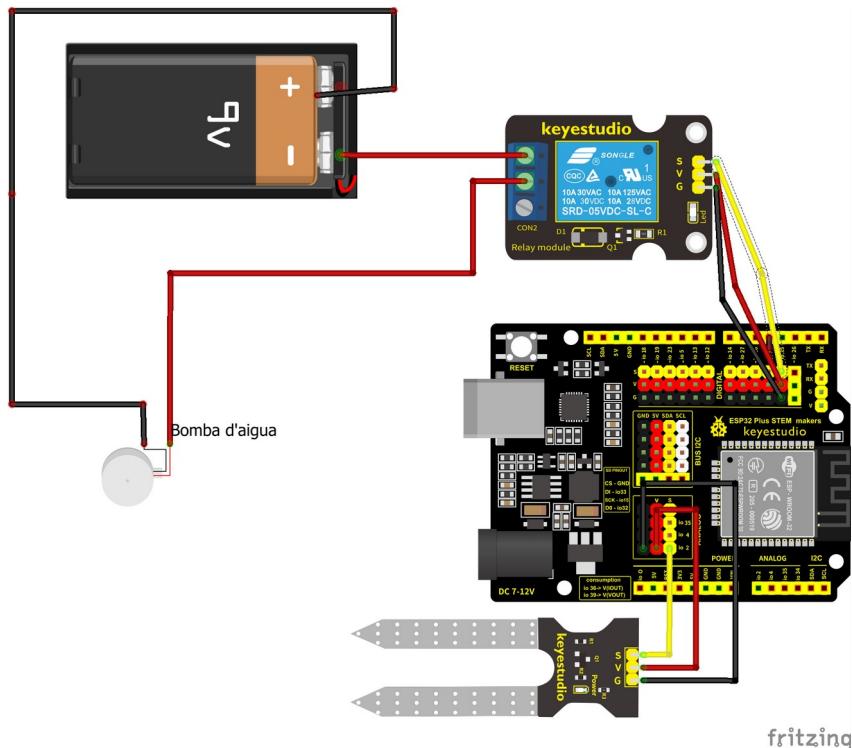


## LISTA DE ELEMENTOS

- 1 placa ESP32 STEAMakers
- 1 cable macho-macho para la conexión del relé
- Relé (relay Module)
- Bomba de agua
- Sensor de humedad del terreno



## MONTAJE



## PROGRAMACIÓN 13



## RETOS DE AMPLIACIÓN

- Sistema de Riego con Previsiones Meteorológicas
- Sistema de Riego Temporizado: Activar la bomba de agua a intervalos regulares programados, independientemente de las lecturas de los sensores. Por ejemplo, la bomba se activa durante 10 minutos cada 2 horas.

## P14. Condiciones ambientales



### OBJETIVOS

- Conocer los elementos que pueden conformar una estación ambiental, para recoger datos: presión atmosférica, altitud, calidad del aire, luz, temperatura y humedad.
- Enviar los datos a través de la consola serie.
- Grabar datos en una tarjeta MicroSD.
- Recuperar datos de una tarjeta MicroSD.

### CONOCIMIENTOS

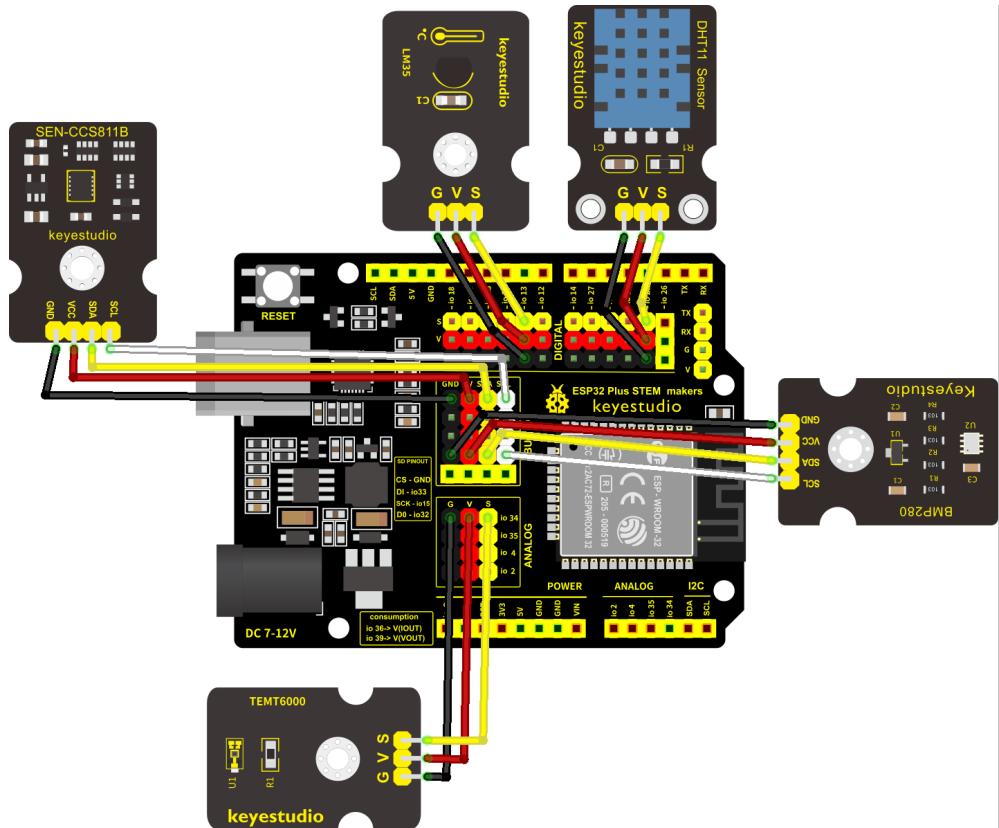
Reconocer el funcionamiento de los distintos sensores.

### LISTA DE ELEMENTOS

- 1 placa ESP32 STEAMakers
- Sensor de presión barométrica BMP280
- Sensor de temperatura y humedad DHT-11
- Sensor de temperatura LM35
- Sensor de luz TEMT 6000
- Sensor de CO<sub>2</sub> CCSS811B
- Tarjeta MicroSD



### MONTAJE

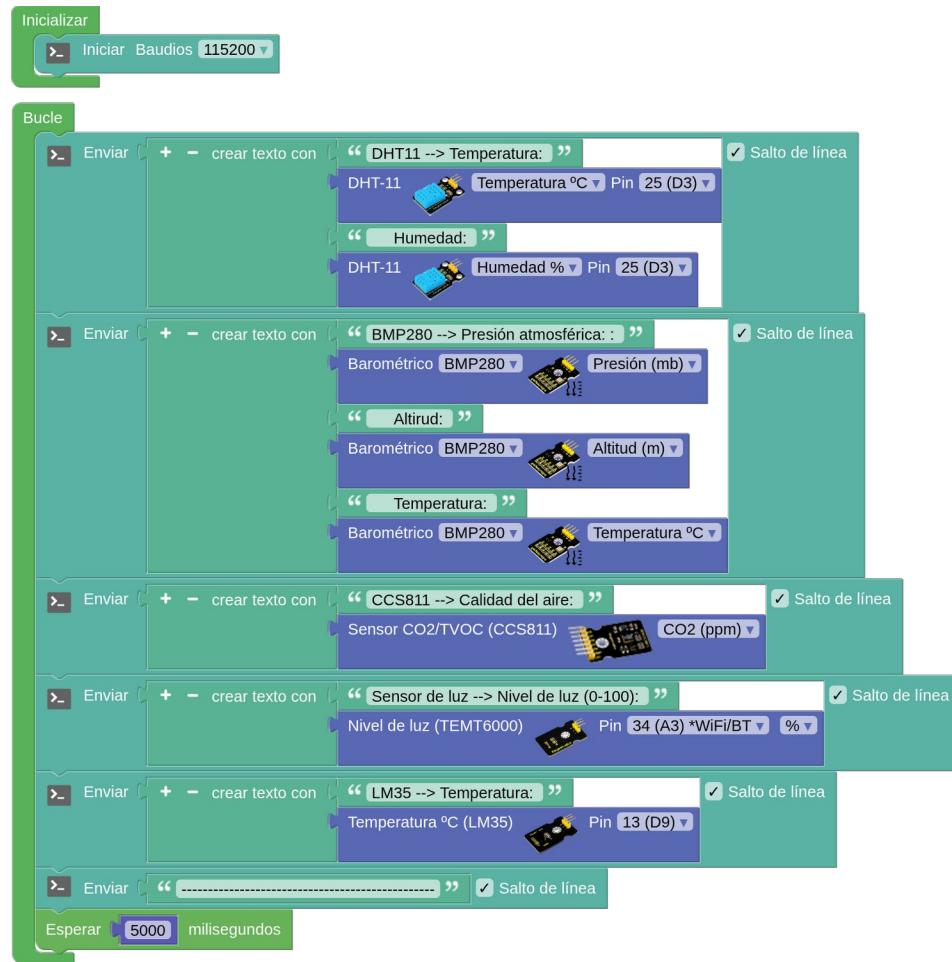


fritzing



## PROGRAMACIÓN 14.1

Para leer los valores de los sensores enviamos sus datos a la consola.



Consola serie

Baudrate: 115200 Conectar Desconectar Limpiar

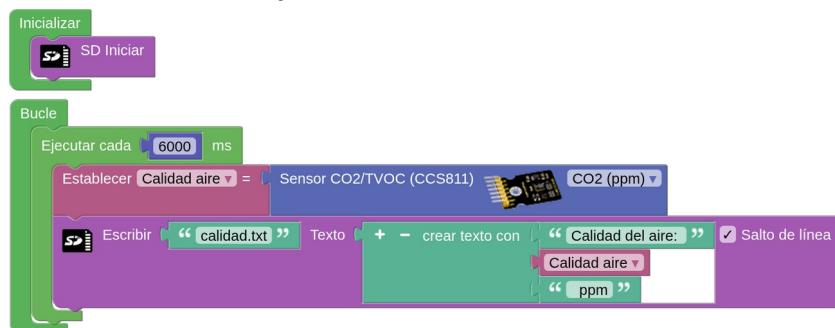
Enviar

```
DHT11 --> Temperatura: 23.00  Humedad: 57.00
BMP280 --> Presión atmosférica: : 939.26  Altitud: 635.06  Temperatura: 22.78
CCS811 --> Calidad del aire: 400.00
Sensor de luz --> Nivel de luz (0-100): 29
LM35 --> Temperatura: 22.30
-----
DHT11 --> Temperatura: 23.00  Humedad: 57.00
BMP280 --> Presión atmosférica: : 939.29  Altitud: 634.81  Temperatura: 22.78
CCS811 --> Calidad del aire: 400.00
Sensor de luz --> Nivel de luz (0-100): 29
LM35 --> Temperatura: 22.20
-----
DHT11 --> Temperatura: 23.00  Humedad: 57.00
BMP280 --> Presión atmosférica: : 939.30  Altitud: 634.74  Temperatura: 22.78
CCS811 --> Calidad del aire: 400.00
Sensor de luz --> Nivel de luz (0-100): 29
LM35 --> Temperatura: 22.20
```



## PROGRAMACIÓN 14.2

Grabar los resultados en una tarjeta SD.



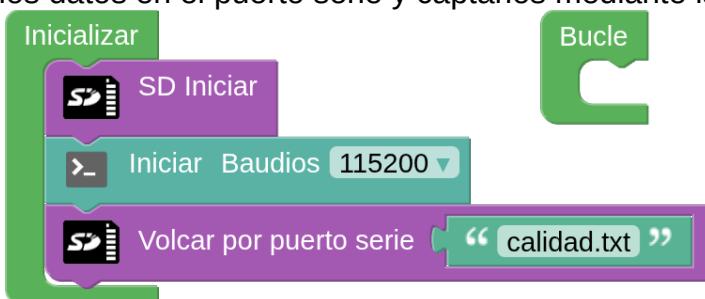
A continuación hemos sacado la tarjeta de la placa y hemos leído el valor en el ordenador:

A screenshot of a text editor window titled '\*calidad.txt'. The file contains a log of 23 entries, each consisting of 'Calidad del aire:' followed by a value in ppm. The values range from 400.00 to 472.00. The text editor interface includes buttons for 'Abrir', 'Guardar', and 'X', and status bars showing the file path 'PRUSA\_MK3S /m...', line count 'Ln 33, Col 30', and insertion mode 'INS'.

Linea	Calidad del aire (ppm)
1	472.00
2	418.00
3	425.00
4	464.00
5	464.00
6	515.00
7	486.00
8	495.00
9	472.00
10	442.00
11	445.00
12	445.00
13	436.00
14	421.00
15	418.00
16	418.00
17	421.00
18	418.00
19	411.00
20	409.00
21	409.00
22	405.00
23	400.00

## PROGRAMACIÓN 14.3

Si queremos volcar los datos en el puerto serie y captarlos mediante la consola.



## RETOS DE AMPLIACIÓN

- Seguidor solar: Ajusta la posición de un panel solar para maximizar la luz solar recibida en el día.
- Monitorización de ruido ambiental: Registra los niveles de ruido en diferentes períodos del día.
- Riego automático: Activa un sistema de riego cuando se detecta que el suelo está seco.
- Monitorización de plantas: Envía notificaciones cuando las plantas necesitan ser regadas.
- Robot autónomo avanzado que detecte fuegos: Que utilice sensores de distancia y temperatura para navegar y reaccionar de forma inteligente en diferentes situaciones.
- Detector de fiebre.



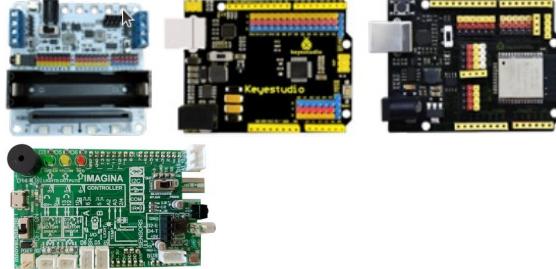
## P15. Motor de CC

### OBJETIVOS

- Controlar el funcionamiento de 1 o 2 motores de corriente continua.



### CONOCIMIENTOS

	<b>Motor con rueda</b>	<b>Actuador</b>
	Motor con reductora, encapsulado con protección, con cable polarizado y rueda adaptable. Voltaje 3.0 a 12.0 V DC. 1 eje (reductora 1: 120, 90 rpm), con placa. A 6V gira aprox. a 55 rpm. Diámetro del eje 5.2 mm. Rueda: diámetro 68 mm y Ancho 25 mm.	Compatible con: 
	<b>Voltaje</b> DC 3 a 6 V	<b>Tipo de señal</b> Digital
<b>Tipo de conexión</b>  Conector JST. Para la micro:shield, conectar clavija o añadir cables de conexión macho-macho.		

Los motores de corriente continua tienen las siguientes características:

Las salidas de la placa STEAMakers no proporcionan suficiente corriente para controlar un motor de corriente continua (proporcionan unos 50 mA y un motor puede consumir unos 1000 mA) por lo que necesitaremos realizar un pequeño circuito bien con un transistor, bien con un driver para controlar una corriente mucho mayor.

### LISTA DE ELEMENTOS

- 1 placa ESP32 STEAMakers
- 2 Motores DC con ruedas
- Protopboard con transistor NPN + batería 9V
- Controlador de motor L298N/L293D + batería 9V
- Shield IMAGINA 3DBot Arduino



## MONTAJE CON TRANSISTOR NPN

Podemos utilizar un transistor NPN en modo corte/saturación que permita, como un interruptor, el paso de una intensidad de corriente muy superior desde una fuente de alimentación auxiliar.

Esquema de conexión	Montaje en protoboard

## PROGRAMACIÓN 15.1

Activar el giro del motor:

Escribir digital Pin 25 (D3) ON

Activar el giro controlando la velocidad:

Escribir analógica (PWM) Pin 25 (D3) Valor 200

Aumentar progresivamente la velocidad:

```

contar con velocidad desde 100 hasta 255 de a 1
hacer
    Escribir analógica (PWM) Pin 25 (D3) Valor velocidad
    Esperar 100 milisegundos
  
```

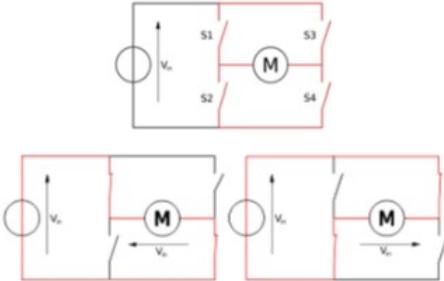
## MONTAJE CON CONTROLADOR DE MOTORES L298n/L293D

Para poder controlar los motores CC, podemos utilizar también el controlador L298N por su facilidad de utilización. El controlador L298N puede trabajar con tensiones de hasta 35V y una intensidad de hasta 2A por cada canal, es decir puede controlar hasta 4A en total y unos 25W.

Este módulo tiene otra característica muy importante y nos permite controlar la dirección de giro de los motores, es decir, podremos hacer que giren en ambos sentidos.

Además, nos permite obtener una tensión del módulo de 5V, perfecta para alimentar a nuestra placa ESP32 sin utilizar ninguna otra fuente adicional. Eso sí, este regulador sólo funciona si alimentamos el módulo con una tensión máxima de 12V.

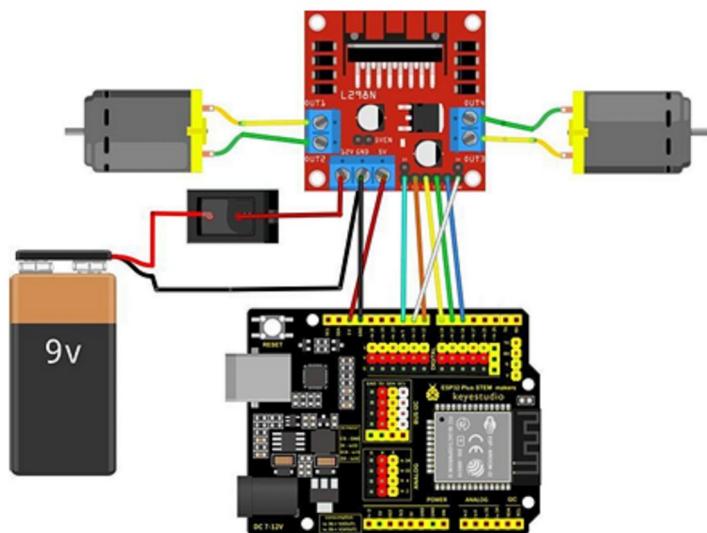


Esquema de puente en H para controlar la dirección de giro de un motor	Módulo típico con configuración en puente en H para control de motores DC
	

La entrada Vin admite tensiones entre 3V y 35V, y justo a su derecha en la imagen tenemos el pin que debemos conectar a GND. La tercera conexión de este grupo es Vlogic que suministra 5V si el jumper del regulador está cerrado.

Los pines IN1 hasta IN4 nos sirven para controlar el sentido de giro de los motores (1 y 2 para el motor de la salida A y 3 y 4 para el motor de la salida B). Funcionan de forma que si IN1 está HIGH e IN2 está LOW, el motor A gira en un sentido y si están a la inversa el motor cambia el sentido de giro.

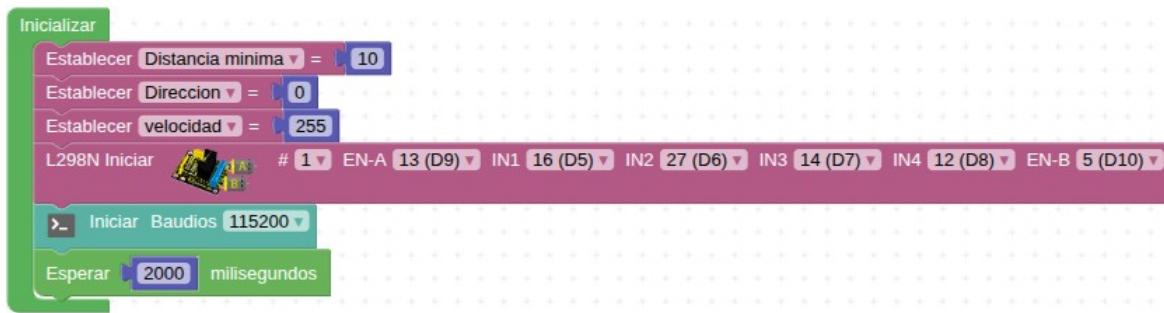
Para controlar la velocidad de giro de los motores debemos quitar los puentes que hay en los pines ENA y ENB y conectar éstos a dos salidas PWM de la placa. De esta forma, enviamos un valor entre 0 y 255 que controlará la velocidad de giro de los motores. Si tenemos los puentes montados, girará siempre a igual velocidad.



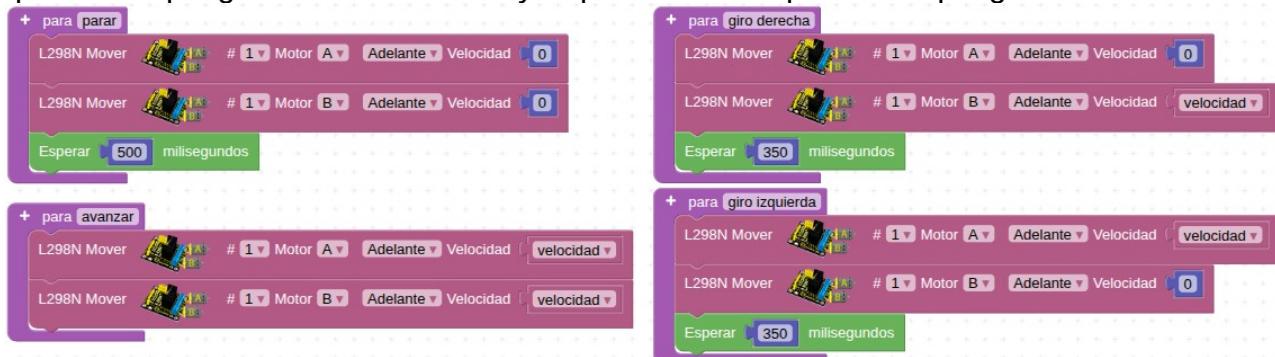
## PROGRAMACIÓN 15.2

Al Inicializar, iniciamos el controlador L298N indicando los pines de la placa ESP32 a los que está conectado, así como la consola serie de nuestra placa. La consola la iniciamos para poder ver alguna lectura del sensor de ultrasonidos y poder ajustar la distancia a la que nuestro coche tendrá que girar. También se pueden crear tres variables que después serán útiles durante el programa.

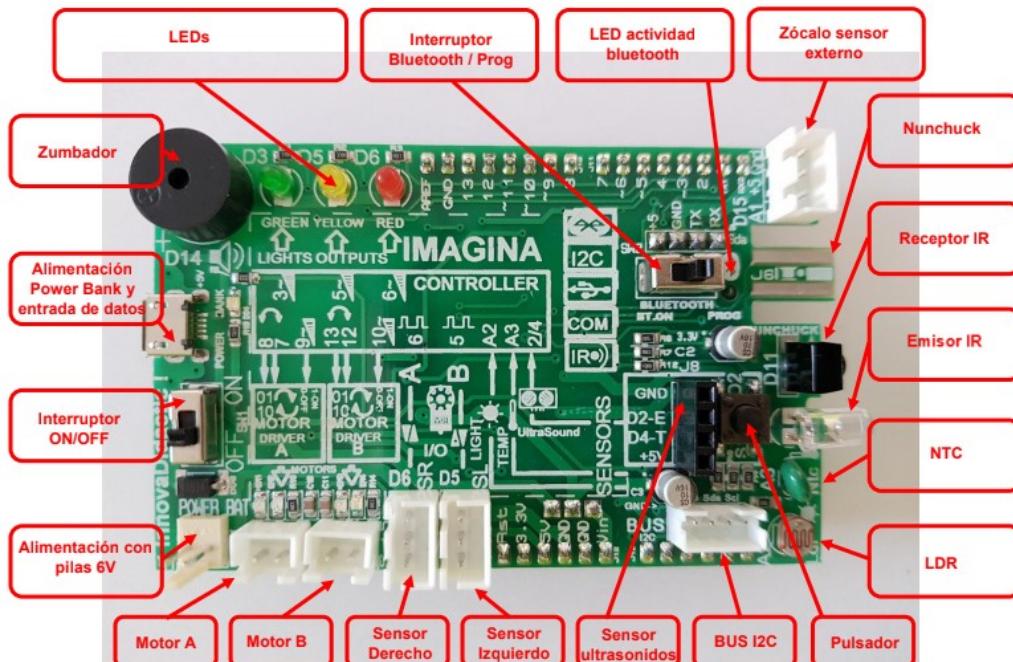




Se pueden crear también las cuatro funciones principales de nuestro programa para evitar escribir cada vez todas las órdenes. Estas funciones son cuatro: avanzar, parar, girar a la izquierda y girar a la derecha. Estas funciones se montan a partir de instrucciones del controlador L298N. Podemos elegir cuál de los dos motores estamos comandante, si queremos que gire adelante o atrás y a qué velocidad queremos que gire.



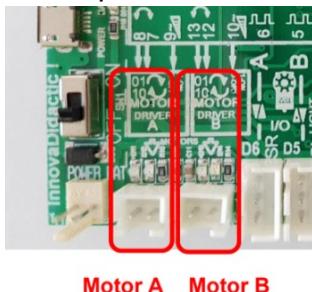
## MONTAJE CON LA SHIELD IMAGINA 3DBOT ARDUINO



Ubicación de componentes



La placa Imagina 3DBot Arduino ya lleva incorporado un “driver” o controlador de motores de CC, con él podemos realizar el control de dos motores de corriente continua de hasta 2A Amperios de consumo.



Utilizaremos la shield Imagina 3DBot Arduino conectada a la placa ESP32 STEAMakers. Este shield dispone, entre otras conexiones, de 2 conectores JST para motores.

Para ampliar información sobre esta placa shield puede consultar el siguiente [enlace](#).

## PROGRAMACIÓN 15.3

arduinoblocks Buscar

Bloques Información

- Lógica
- Control
- Matemáticas
- Texto
- Variables
- Listas
- Funciones
- 3dBot

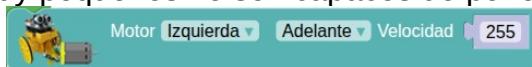
Creamos un nuevo proyecto del tipo ESP32 STEAMakers + Imagina 3DBot.

En el submenú 3dBot de arduinoblocks podemos observar que existen diferentes instrucciones específicas para controlar el motor A (o izquierdo) donde podemos escoger qué acción hará si girar hacia adelante o hacia atrás.

Cambiando la selección del motor, también sirve para controlar las mismas acciones para el motor B (o derecho).

En total podemos hacer funcionar cualquiera de los dos motores, en cualquiera de los sentidos, parar e incluso ajustar la velocidad de rotación.

En “Velocidad”, valores muy pequeños no son capaces de poner en marcha el motor.



Si introducimos el programa siguiente observaremos que el motor A va hacia delante a una velocidad rápida durante 0.2 segundos, después a una velocidad más lenta otros 0.2 segundos y finalmente estará parado 0.2 segundos y repetirá el proceso.



## RETOS DE AMPLIACIÓN

- Hacer que se mueva el multientrenador: Montar los motores en los laterales del multientrenador y programarlo para que avance como un robot.
- Reloj: crear un reloj con los dos motores, uno que haga de aguja minutera y el otro de las horas.



## P16. infrarojos + mando a distancia



### OBJETIVOS

- Encender y apagar LEDs con el mando.
  - 1 → LED AMARILLO
  - 2 → LED ROJO
  - 3 → LED VERDE
  - 4 → LED BLANCO
- Control de motores con infrarojos

### CONOCIMIENTOS

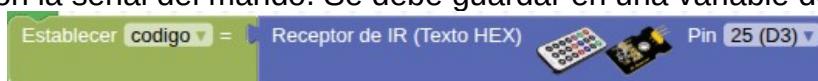
Gran parte de los electrodomésticos utilizan mandos a distancia de infrarrojos, como los televisores o los equipos musicales. El sensor infrarrojo es un dispositivo optoelectrónico capaz de medir la radiación electromagnética infrarroja de los cuerpos en su campo de visión. Todos los cuerpos emiten cierta cantidad de radiación, ésta resulta invisible para nuestros ojos, pero no para estos aparatos electrónicos, puesto que se encuentran en el rango del espectro justo por debajo de la luz visible.

En el caso del receptor de infrarrojos (IR) permite codificar los protocolos de señales de pulsos infrarrojos utilizados por los mandos a distancia.

El mando a distancia contiene un circuito interno, un procesador y un led que emite la señal infrarroja.

La señal infrarroja transmite el código correspondiente al botón pulsado del mando a distancia y lo transmite al dispositivo en forma de una serie de impulsos de luz infrarroja. El receptor IR los recibe y pasa a un procesador que decodifica la serie de 0 y 1 en un dato de texto, que procesaremos en nuestro programa.

En el menú Sensores → Receptor IR, encontramos los bloques que permiten leer el dato que nos llega con la señal del mando. Se debe guardar en una variable de texto.



Para comprobar el código que se está recibiendo de cada tecla del comando, podemos enviar este dato a la consola.

Una vez que sabemos el código recibido, podemos programar las diferentes funcionalidades.

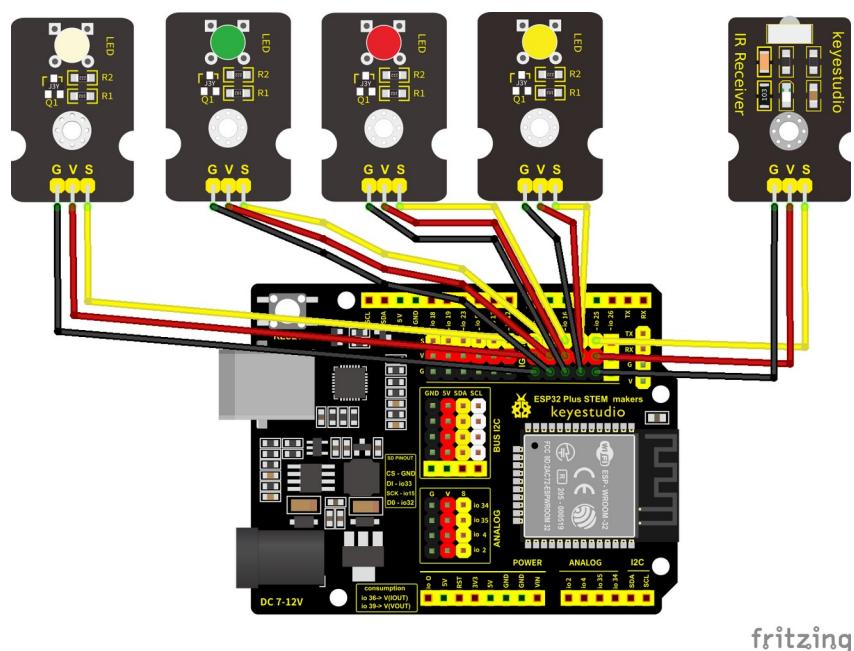
### LISTA DE ELEMENTOS

- 1 placa ESP32 STEAMakers
- Mando a distancia
- Receptor IR
- Pila de botón CR2025/2032



## MONTAJE

LED	ENTRADA ESP32 STEAMakers
Sensor IR	D3/IO23
Amarillo	D4/IO17
Rojo	D5/IO16
Verde	D6/IO27
Blanco	D7/IO14



fritzing



## PROGRAMACIÓN 16



## RETOS DE AMPLIACIÓN

- **Comandar motores con IR**

Podemos simular que los motores de corriente continua que tenemos son los de un robot y diseñar funciones de movimiento del robot, que se activan con las teclas del mando.

Conectamos los motores con la shield Imagina 3DBot Arduino y abrimos un proyecto del arduinoblocks, tipos: ESP32 STEAMakers + Imagina 3DBot.

El mando en la placa Imagina 3DBot Arduino 3DBot, está conectado a la entrada D11/IO23 de la placa, y el proyecto dispone ya de los bloques del 3DBot, que controlan los movimientos de los motores.

- **Descifrar señales de otros comandos y reutilizarlas**

Cogemos cualquier mando de luces LED que tengamos por casa y desciframos los códigos que envía cada botón. Reutilizamos este mando para gobernar a los actuadores de nuestro Kit.



## P17. Emisor Infrarrojos



### OBJETIVOS

- Copiar un botón del mando desde una placa UNO y que lo recibamos desde una STEAMakers, para poner en marcha un LED.

### CONOCIMIENTOS

Mediante el módulo emisor IR, mando a distancia. podemos emitir códigos de infrarrojos como un mando a distancia.

Haremos un programa para la placa emisora, en este caso, la Keyestudio UNO, utilizando el blog que está en el menú Actuadores.

Es necesario conectar el emisor al pin 3 de la placa, escribir el protocolo del mando que queremos clonar y el número de bits que utiliza.



El mando utiliza un protocolo NEC con número decimal de 32 bits. Por tanto, nosotros enviaremos un número en decimal, que el receptor nos codifica en Hexadecimal.

El sistema hexadecimal (abreviado hex) es un sistema numérico con base 16. Se representa normalmente utilizando los símbolos del 0 al 9 y de la A a la F. Por ejemplo, el número decimal 79, cuya representación en sistema binario es 01001111, puede escribirse como 4F en hexadecimal.

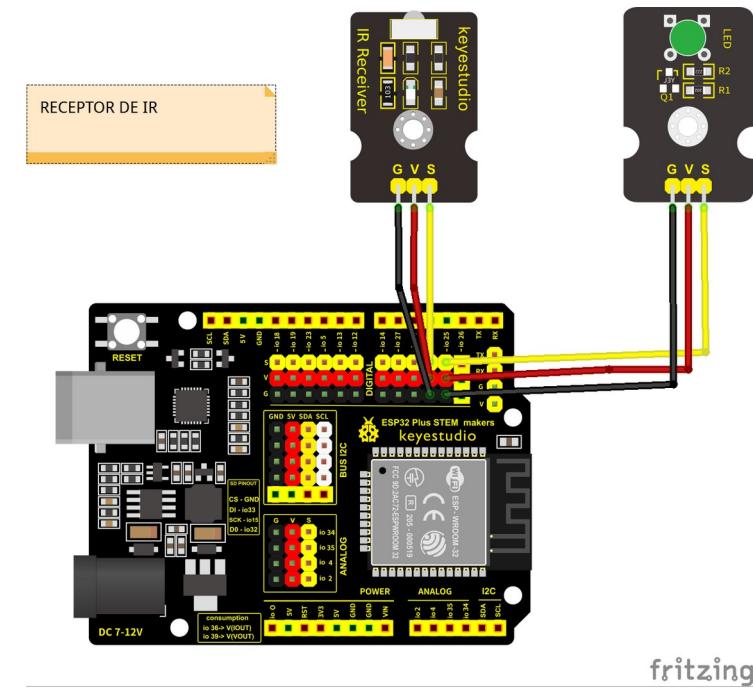
Si el código del mando en Hexadecimal es 00FF6897, será necesario que envíemos el número 16738455 en decimal.

### LISTA DE ELEMENTOS

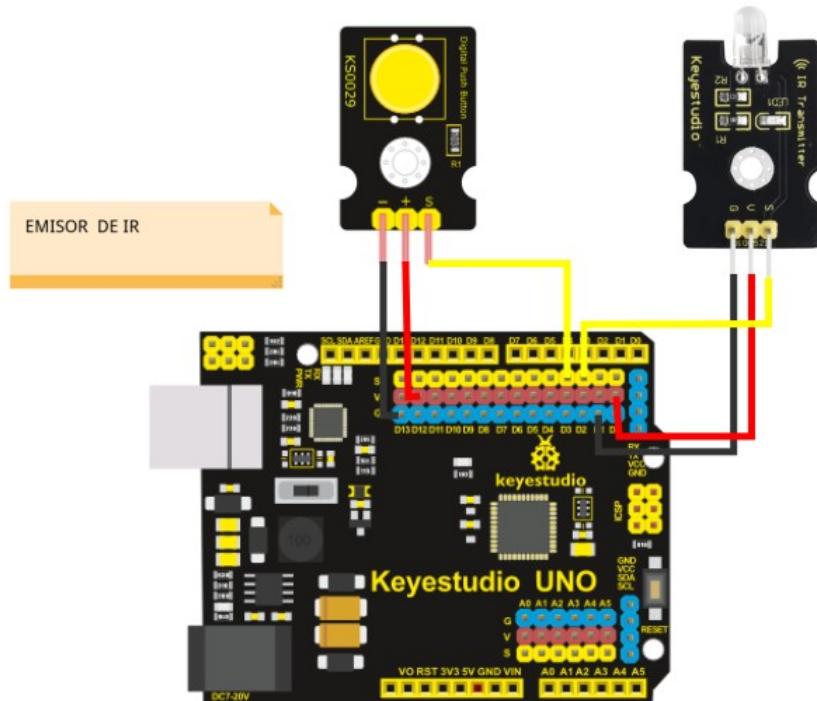
- Placa ESP32 STEAMakers
- Placa Keyestudio UNO
- Emisor IR
- Receptor IR
- Pulsador
- LED



## MONTAJE



fritzing



Nota: El montaje del emisor se puede realizar con una placa Keystudio UNO

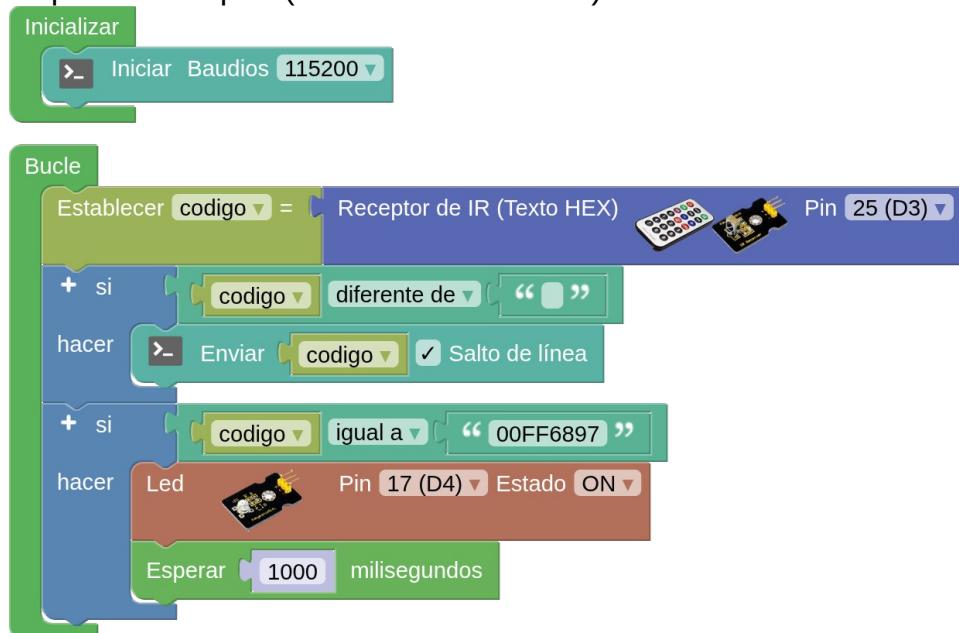


## PROGRAMACIÓN 17

Programación para el emisor (Keyestudio UNO)



Programación para el receptor (ESP32 STEAMakers)



## RETOS DE AMPLIACIÓN

- Robot controlado a distancia:** Controlar los movimientos del 3DBot a través de distintos sensores del panel de control.



## P18. Detector de movimiento PIR

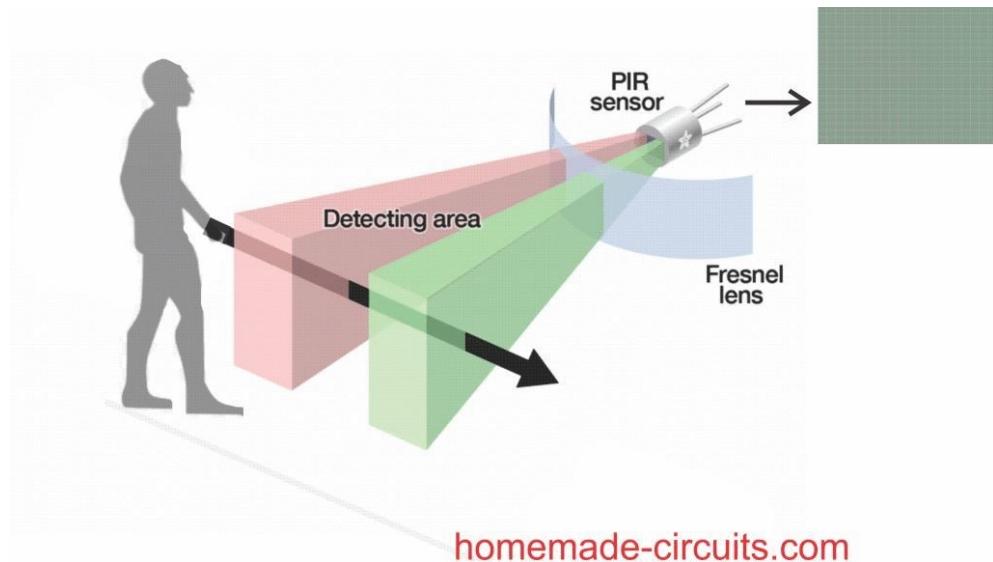


### OBJETIVOS

- Encender y apagar un LED mediante el detector de movimiento PIR.

### CONOCIMIENTOS

En el siguiente reto utilizaremos el detector de movimiento, sensor de infrarrojos pasivo, (PIR siglas en inglés). Este sensor funciona detectando las emisiones infrarrojas de los cuerpos que emiten personas o animales debidas al calor corporal, a mayor calor mayor radiación.



El funcionamiento interno del dispositivo se basa en un sensor piroeléctrico capaz de detectar las radiaciones infrarrojas y transformarlas en una señal eléctrica. En realidad, cada sensor PIR se divide en dos campos, y lo que se detecta es la diferencia entre ellos. Si la señal recibida por ambos campos es diferente se produce corriente eléctrica y si es igual, no.

La otra parte del conjunto es la óptica, necesaria para que todo funcione, se trata de una cúpula de polietileno, conocida como lente de Fresnel, que concentra la luz en un punto central o foco para facilitar la lectura por parte del sensor.

Debe tenerse en cuenta que el rango de detección y el tiempo de respuesta del [detector de movimiento PIR de Keyestudio](#) está predeterminado y no se puede modular como en otros sensores PIR más potentes.



El tiempo de respuesta es de unos 3 segundos, lo que significa que puede retrasar la ejecución de programas complejos.



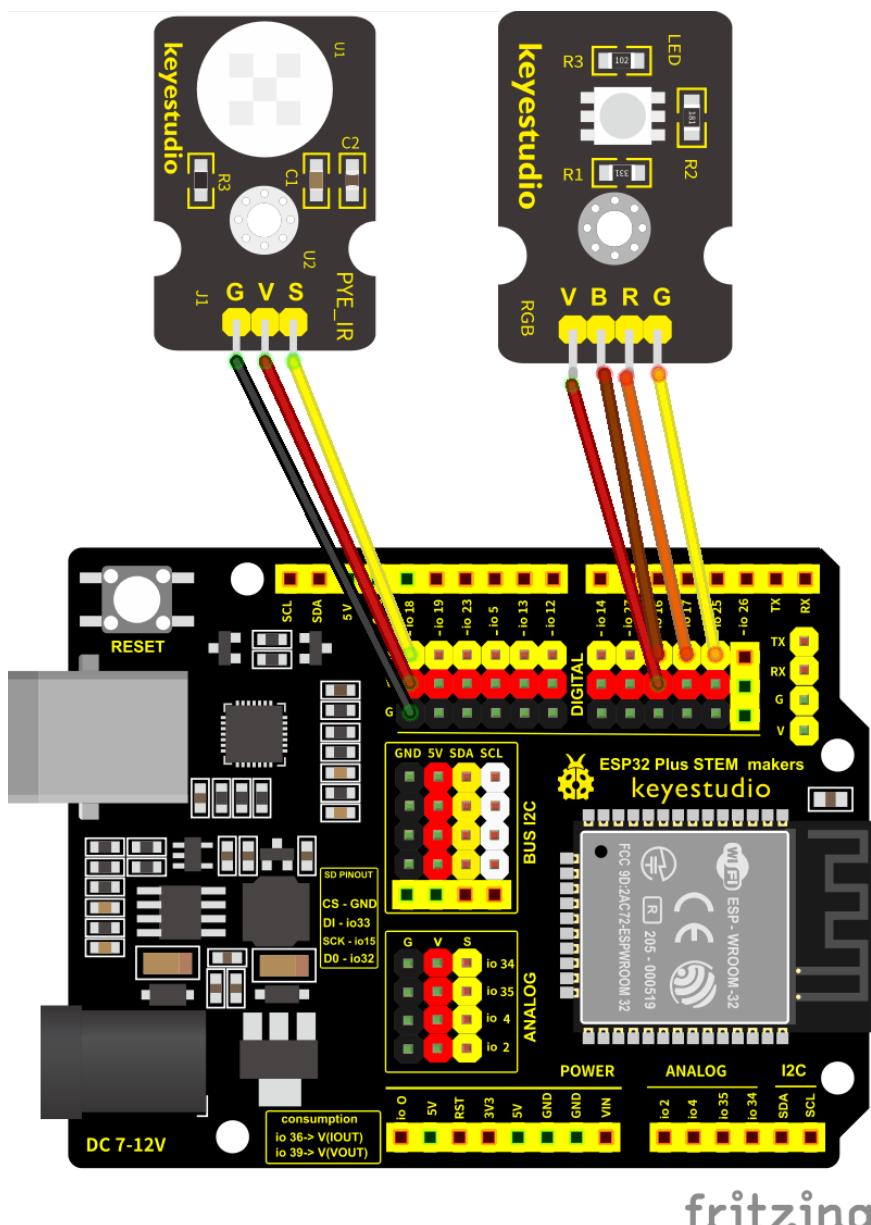
## LISTA DE ELEMENTOS

- 1 placa ESP32 STEAMakers
- Detector de movimiento PIR
- LED



## MONTAJE

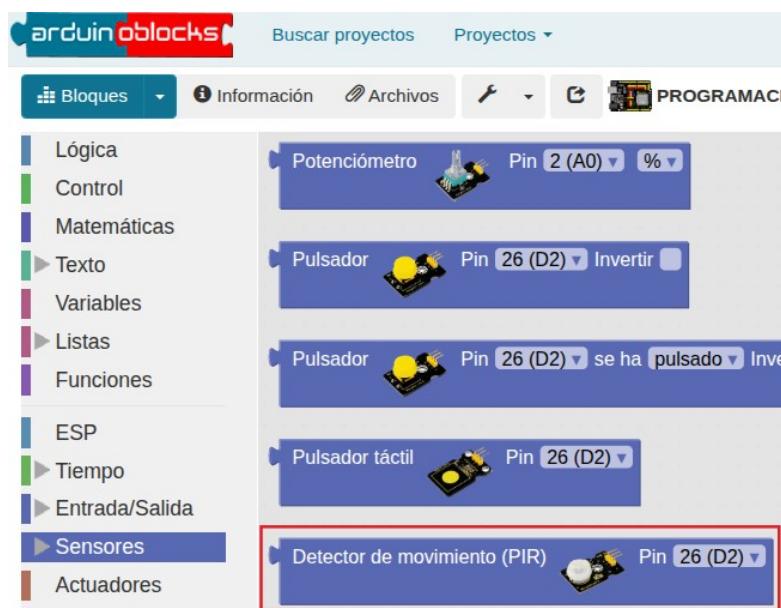
Conectamos el detector de movimiento PIR en el pin IO18 (D13) y el LED RGB debemos conectarlo de manera diferente, el cable que sale del positivo V al rojo del D5, y las señales de cada color así: Azul en el pin amarillo del D5, Red en el pin amarillo del D4, Green en el pin amarillo del D3.



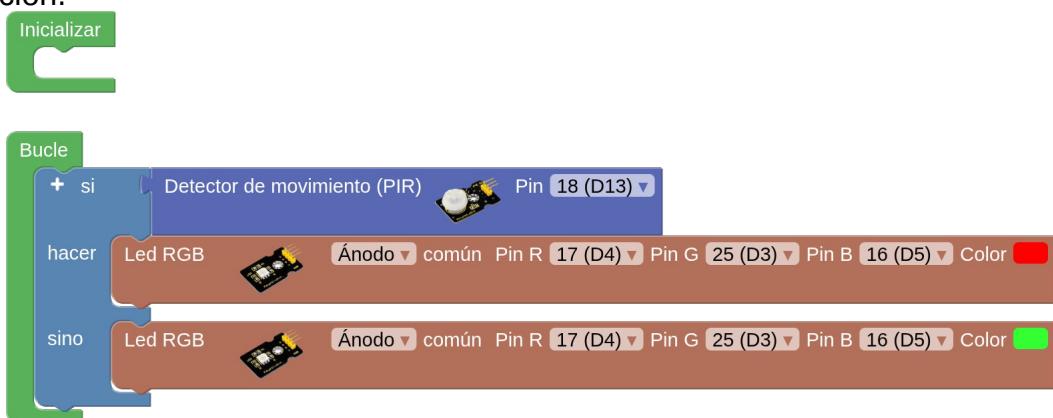
## PROGRAMACIÓN 18

Ahora realizaremos un programa en el que al pasar por delante del detector de movimiento se encienda el LED RGB con color rojo y cuando deje de detectar se quede en verde.

En el menú de arduinoblocks encontramos el bloque del detector de movimiento PIR en el apartado de sensores. Los bloques correspondientes al LED RGB ya se han explicado en prácticas anteriores.



Simplemente hacemos un bloque condicional, en el que la condición es que si el sensor detecta movimiento, la acción es poner en rojo el LED, y ponerlo en verde si no se cumple la condición.



## RETOS DE AMPLIACIÓN

- Ahorra energía con la luz del pasillo: podemos utilizar este sensor para ahorrar energía, si hacemos que controle la luz de un pasillo. Normalmente la luz estará apagada y cuando el sensor PIR detecte movimiento que se encienda la luz durante 20 segundos (tiempo que una persona tarda en atravesar el pasillo).

## P19. Módulo RFID



### OBJETIVOS

- Apertura de una puerta controlada con RFID: Al acercar la tarjeta al módulo RFID, el servomotor girará simulando que se abre una puerta.

### CONOCIMIENTOS

El lector de tarjetas está compuesto por un módulo de radiofrecuencia y un campo magnético de alto nivel. El transpondedor (el tag o tarjeta) es un dispositivo de detección, que no contiene batería. Sólo contiene pequeños circuitos integrados y medios para almacenar datos y antenas para recibir y transmitir señales.

Para leer los datos de la tarjeta, es necesario acercarla al lector. Éste generará un campo magnético, que produce electricidad acorde con la ley de Lenz, activando así el dispositivo.



Para saber qué código contiene cada tarjeta o tag, leeremos el código con el sensor RFID conectado por I2C. El valor leído, lo guardamos en una variable de texto y lo enviamos al puerto serie, o a la pantalla OLED para poder visualizarlo.

Los bloques para controlarlo se encuentran en el menú periféricos → RFID (I2C).

El siguiente programa nos permite inicializar el módulo, leer la tarjeta, guardar el código en una variable y visualizarlo en la consola.

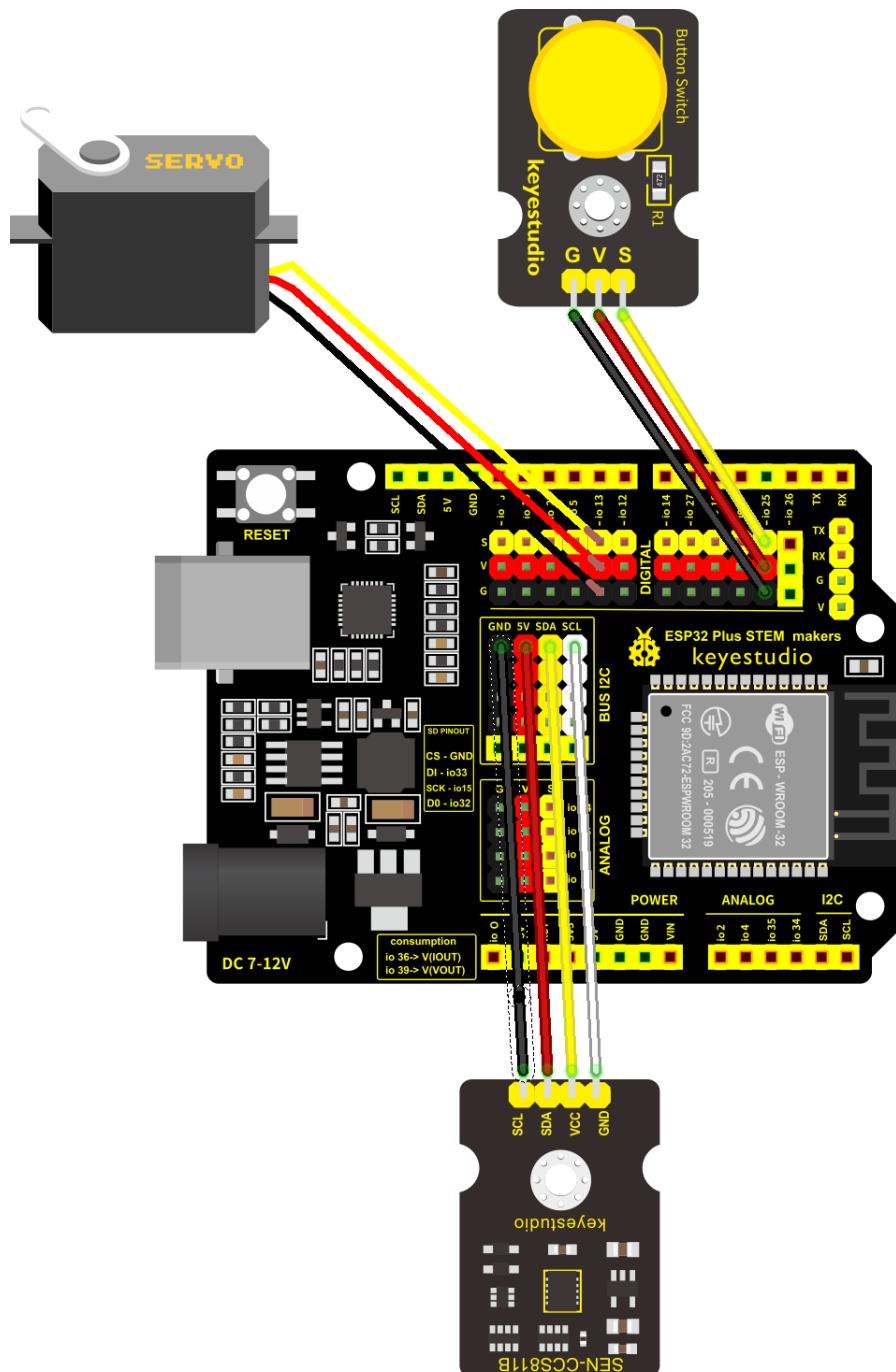


### LISTA DE ELEMENTOS

- 1 placa ESP32 STEAMakers
- Módulo RFID con tag o tarjeta
- Pulsador
- Servomotor



## MONTAJE



fritzing



## PROGRAMACIÓN 19



## RETOS DE AMPLIACIÓN

- **Visualización:** La pantalla OLED nos indica el estado de la puerta.
- **Puerta de garaje:** La puerta se abre con la tarjeta y se cierra si, pasado cierto tiempo, el sensor de colisión está activado. Este sensor se trabaja igual que el pulsador, con un bloque de entrada digital. Sin pulsar la palanca da ON y al pulsar da OFF.

## P20. Seguidor de líneas

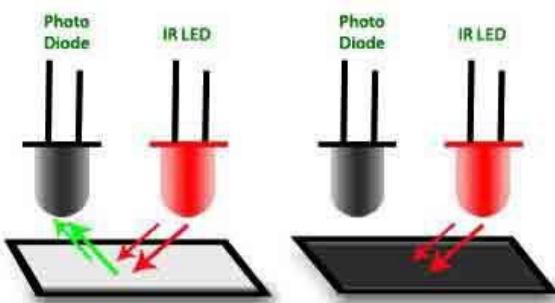


### OBJETIVOS

- Emplear el sensor de infrarrojos tipo detector digital de líneas negras y blancas como un contador de líneas de un papel impreso.

### CONOCIMIENTOS

El sensor de líneas KS0050 está formado por dos sensores infrarrojos TCRT5000. Su principio de funcionamiento es el de emitir una señal infrarroja (emisor) y detectar la cantidad reflejada (receptor). En caso de que el color sea negro, este color absorbe toda la luz pero si es de color blanco la refleja. Por tanto, si el color es negro devolverá un valor Alto (1) y si es blanco devolverá un valor Bajo (0). La altura de detección está comprendida entre 0,5 y 3 cm.



En la parte superior existe un potenciómetro para poder ajustar la sensibilidad. Junto al potenciómetro hay un pequeño LED rojo que se enciende cuando el sensor detecta el color blanco y se apaga con el color negro.

### LISTA DE ELEMENTOS

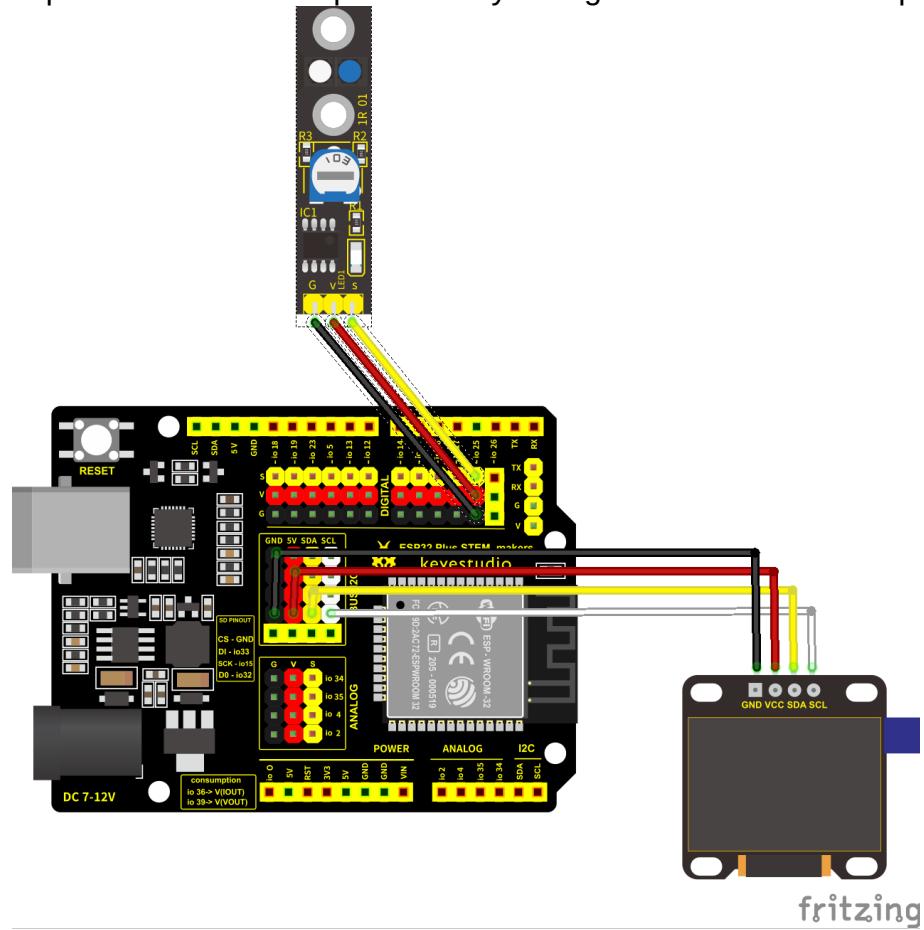
- 1 placa ESP32 STEAMakers
- Sensor seguidor de líneas
- Pantalla OLED
- Hoja con líneas negras

Imprimir el archivo:  
P20 - Seguidor de líneas - plantilla imprimir.pdf



## MONTAJE

Conectamos la pantalla OLED a un puerto I2C y el seguidor de líneas en el pin digital D3.



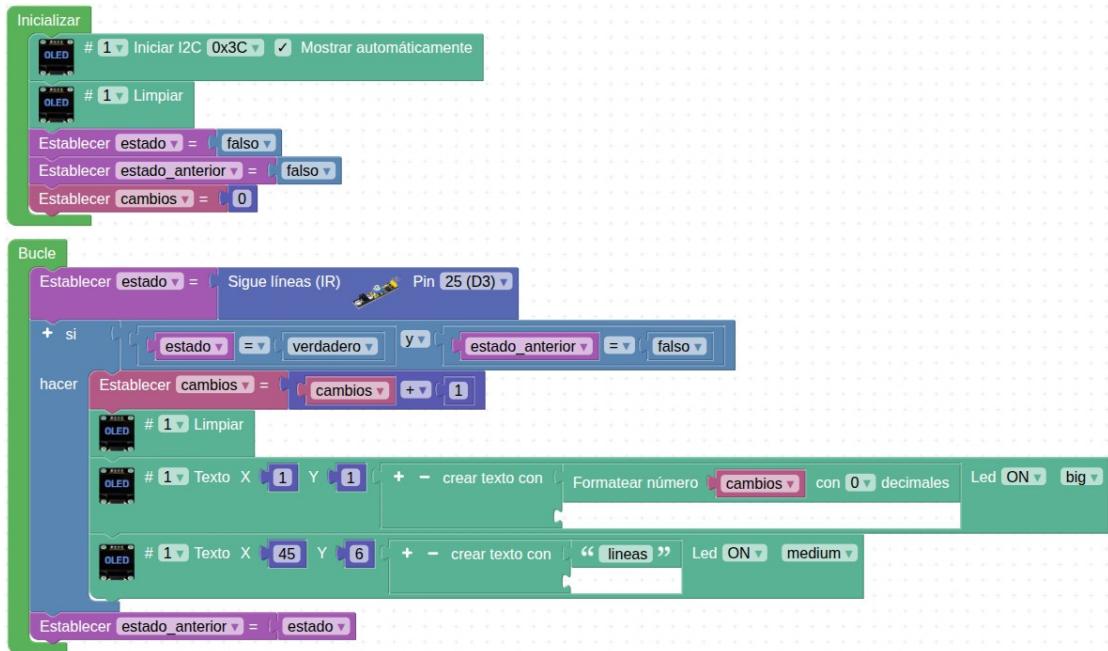
## PROGRAMACIÓN 20

Para realizar este programa debemos tener en cuenta que el sensor transmite señal de manera continua siempre que está detectando un color u otro, por tanto si queremos hacer un contador de líneas lo que debemos programar que se cuente son los cambios de estado, no la señal. Por eso necesitamos crear dos variables booleanas "estado" y "estado\_anterior" con las que podemos comprobar cuándo hay un cambio de estado. Además necesitaremos una tercera variable de tipo numérico "cambios" para realizar el sumatorio de los cambios de estado, ésta será la que mostramos en pantalla.

En inicializar declaramos las variables e iniciamos la pantalla OLED, limpiamos la pantalla por si queda algo anterior.

En el bucle asociamos el valor de la variable "estado" a la lectura del sensor, y después comprobamos si ese valor es diferente que el anterior. Si es diferente significa que ha cambiado de blanco a negro, en cuyo caso sumamos 1 a la variable "cambios" cada vez que se recorre el bucle se vuelve a dar a la variable "estado anterior" el valor de la variable "estado".





Una vez transferido el programa pasamos la hoja mirando hacia abajo por encima del lector. Debemos intentar no hacerlo demasiado rápido y que no pasen las manos por encima porque pueden dar fallos.

## RETOS DE AMPLIACIÓN

- **Ampliación 1: Escala musical.** Modificar el programa anterior de forma que cada vez que pase una línea el zumbador reproduzca una nota musical diferente. Se puede hacer una escala desde el DO hasta el SI, y volver a empezar.
- **Ampliación 2: Contador de vueltas.** Diseñar una rueda con un radio de color negro, imprimirla en cartulina y ensamblarla en el motor de CC.



### OBJETIVOS

- Conocer y utilizar los sensores integrados y el sistema de medida de consumo de energía en la placa ESP32 STEAMakers.

### CONOCIMIENTOS

La placa ESP32 STEAMakers lleva incorporado:

- Un sensor de efecto hall
- Un sistema de medida de intensidad y voltaje

Con el sensor de efecto hall detectamos variaciones de campos magnéticos en la proximidad de la placa.

Para explicar el funcionamiento del sistema medidor de consumo de energía incorporado en la placa ESP32 STEAMakers, es necesario tener en cuenta varios conocimientos teóricos que cubren los aspectos fundamentales de la electrónica y la física:

- Intensidad de Corriente (I): La cantidad de carga eléctrica que pasa por un conductor por unidad de tiempo. Se mide en amperios (A).
- Voltaje (V): La diferencia de potencial eléctrico entre dos puntos. Es la fuerza que impulsa a los electrones a través de un circuito. Se mide en voltios (V).
- Resistencia (R): La oposición que ofrece un material al paso de corriente eléctrica. Se mide en ohmios ( $\Omega$ ).

La ley de Ohm establece que la intensidad de la corriente que pasa por un conductor entre dos puntos es directamente proporcional a la diferencia de potencial entre estos puntos e inversamente proporcional a la resistencia del conductor:

$$I = \frac{V}{R}$$

Potencia Eléctrica (P): La cantidad de energía consumida por unidad de tiempo. Se mide en vatios (W) y se calcula con la fórmula:

$$P = V \cdot I$$

Energía (E): La capacidad para realizar trabajo. En el contexto eléctrico, se mide en julios (J) o, más comúnmente, en kilowatios hora (kWh). La fórmula para calcular la energía es:

$$E = P \cdot t$$

Donde t es el tiempo durante el cual la potencia ha sido consumida.

Si la tensión de alimentación de la placa ESP32 STEAMakers baja de los 4,8V, ésta no funcionará correctamente (sobre todo la transmisión WiFi). Es recomendable realizar una comprobación de la tensión que está entregando el puerto USB del ordenador o la fuente externa de energía que se esté utilizando.

Dentro de arduinoblocks, los bloques para trabajar con estos sensores los encontramos



en 'Sensores' → 'Integrados', y son los que vemos en la siguiente figura:

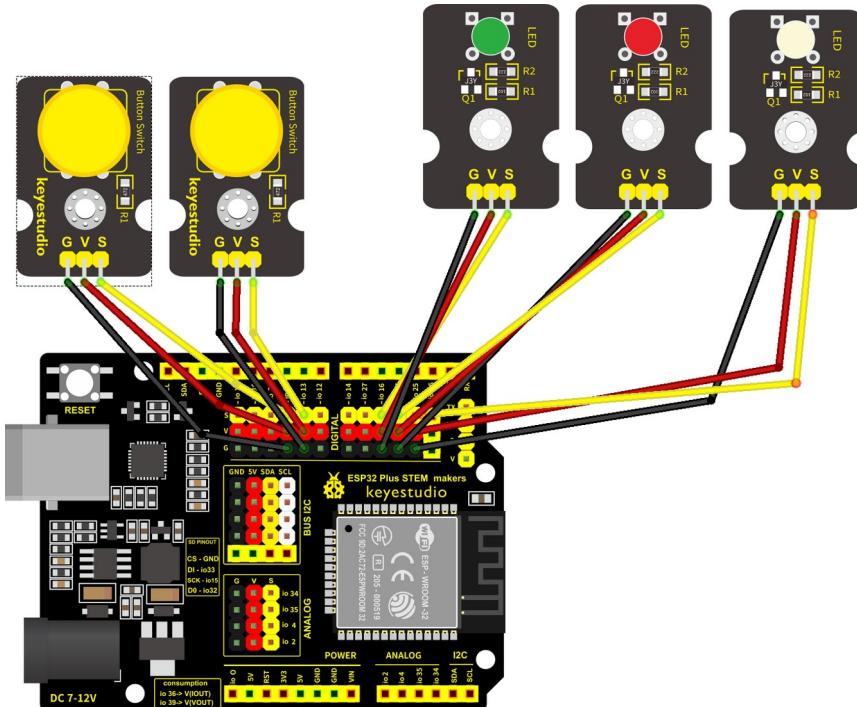


## LISTA DE ELEMENTOS

- 1 placa ESP32
- 1 pulsador
- 1 sensor de colisión (hará la función de pulsador)
- 3 LEDs



## MONTAJE



## PROGRAMACIÓN 21.1



Sensor de campo magnético: Podemos observar que al colocar un objeto metálico cerca de la STEAMakers el valor de proximidad aumenta y al alejarlo disminuye.

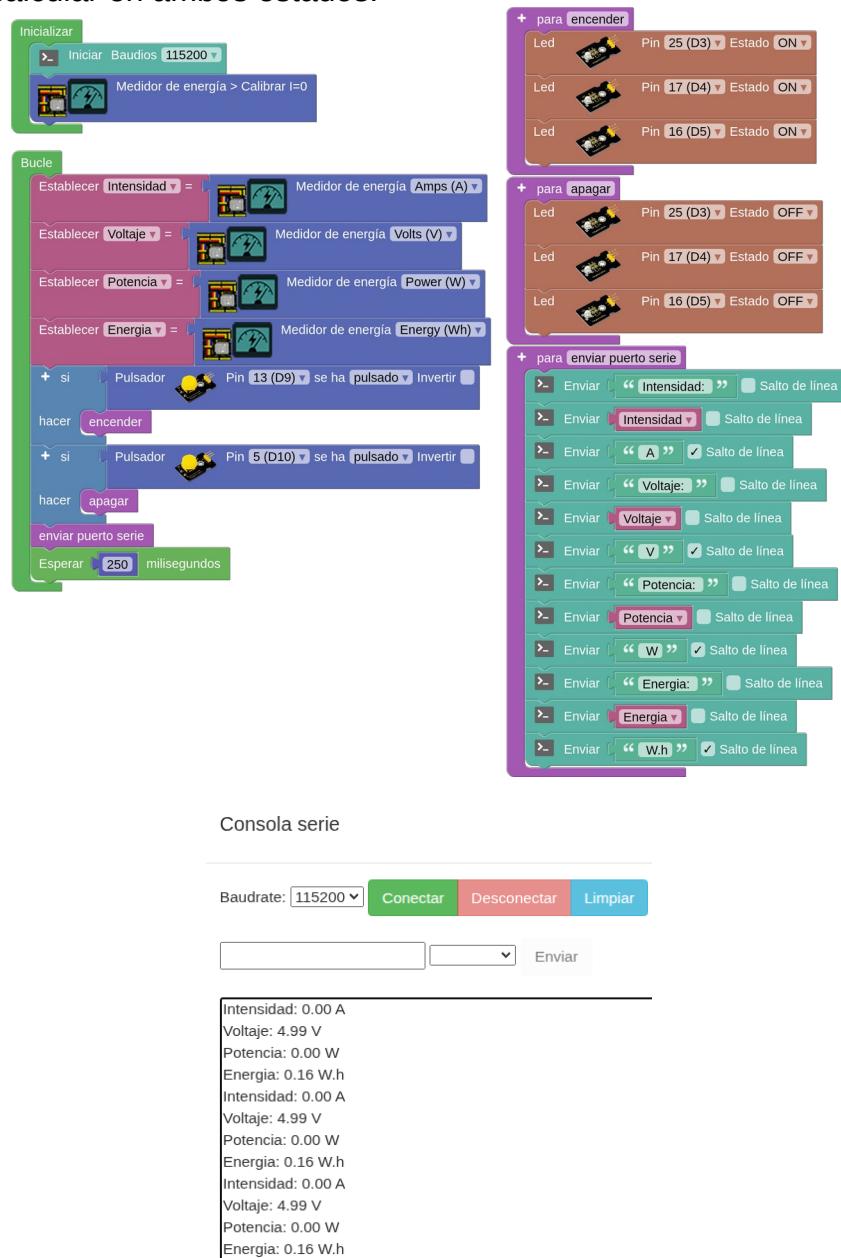


PROGRAMACIÓN 21.2

**Leer la tensión de alimentación de la placa:**

## PROGRAMACIÓN 21.3

Controlar el encendido y apagado de 3 LEDs mediante 2 pulsadores (o un pulsador y controlar el estado con una variable booleana). Comparar los 4 parámetros de consumo que podemos calcular en ambos estados.



## RETOS DE AMPLIACIÓN

- Guardar los datos de los sensores internos:** graficar las 4 magnitudes y recogerlas en un archivo .csv.
- Consumo por colores:** Comparar el consumo de energía si conectamos la tira de leds neopixel y, cuando ponemos en marcha los LEDs lo hacemos:
  - Todos en rojo
  - todos de color verde
  - todos de color azul
  - todos de color blanco



## Glosario de términos de programación

Bucle: [P11](#)

Condicional: [P02](#)

Funciones: [P12](#)

Listas: [P08](#)

Mapear: [P12](#)

Operadores aritméticos

Operadores lógicos

Variables: [P02](#)

Números aleatorios: [P04](#)

## Reconocimientos

Este manual está elaborado en base a documentos anteriores, reconocer y agradecer el trabajo de:

- Physical Computing. ESP32 Plus STEAMakers, Imagina TDR STEAM, arduinoblocks. Versión 4.0. Fernando Hernández e Innova Didactic.
- GUÍA DE USUARIO PLACA ESP32 STEAMakers. Innova Didactic.

## Otras fuentes de documentación utilizadas

<https://www.luisllamas.es/>. Luis Llamas. Ingeniería, informática y diseño.

## Imágenes y material gráfico

- P18: [Imagen funcionamiento PIR](#). Lavina Saldhana, bajo una licencia Creative Commons Atribución-CompartirIgual 4.0 internacional.
- [Imagen del sensor KS0349](#) de P20. Wiki Keystudio.
- [Imagen del sensor RFID](#) P19 con i2C. Innova didactic.

