

SMART HOME ESP32

- [¿Qué es Smart Home con ESP32?](#)
- [Introducción](#)
 - [Objetivos y contenidos](#)
 - [Pensamiento computacional](#)
 - [Hardware ¿Qué es ESP32?](#)
 - [Software](#)
 - [Sensores](#)
 - [Actuadores](#)
- [Montaje](#)
 - [Preliminar](#)
 - [Las paredes](#)
 - [Montando las paredes](#)
 - [Conexiones](#)
- [Cómo programar la Smart Home ESP32](#)
- [STEAMAKERSBLOCKS](#)
 - [Qué es Steamakersblocks](#)
 - [Crear cuenta en Steamakerblocks](#)
 - [Cuentas alumnos](#)
 - [Connector](#)
 - [Empezando un proyecto](#)
 - [Mi primer programa](#)
- [PYTHON](#)

- [Descarga del programa](#)
- [Instalar el firmware](#)
- [Mi primer programa](#)
- [Instalando librería LCD en Micropython](#)
- [Librería MFRC](#)
- [Introducción al Python](#)

- [RETOS no IoT](#)
 - [Project 1.1 LED Flashing](#)
 - [Project 1.2 Breathing LED](#)
 - [Project 2.1 Read the Button](#)
 - [Project 2.2. Table Lamp](#)
 - [Project 3.1 Read the PIR Motion Sensor](#)
 - [Project 4.1 Play Happy Birthday](#)
 - [Project 5.1 Control the Door](#)
 - [Project 5.2 Close the Window](#)
 - [Project 6.1 RGB Sk6812](#)
 - [Project 7.2 Button Control Fan](#)
 - [Project 8.2 Dangerous Gas Alarm](#)
 - [Project 9.1 Temperature and Humidity Tester](#)
 - [Project 10.1 Open the Door](#)

- [RETOS Bluetooth](#)
 - [Un poco de teoría Bluetooth](#)
 - [Bluetooth en el ESP32](#)
 - [Reto de la APP hacia ESP32](#)
 - [Reto APP y ESP32 BIDIRECCIONAL](#)

- [Qué es IoT](#)
- [RETO IoT Blynk](#)
 - [¿Qué es Blynk?](#)



- [Blynk legacy y Blynk IoT](#)
- [Crear DASHBOARD en Blynk.io](#)
- [Programa en Steamakersblock](#)
- [Resultado](#)

- [RETO IoT Telegram](#)
 - [Crear bot en Telegram](#)
 - [Encontrar tu ID en Telegram](#)
 - [Código para enviar y recibir](#)
 - [Resultado](#)

- [Creditos](#)

¿Qué es Smart Home con ESP32?

Se trata de una maqueta con montón de sensores y actuadores muy bien diseñada para sacarle al máximo jugo a la domótica, programación y mundo maker.

Te recomiendo [ver la lista de componentes](#), muy variados y con gran potencial educativo

[smart_home.png](#)

Fuente [Innovadidactic](#) con permiso del comercial



Introducción

Introducción

Objetivos y contenidos

Objetivos

- Capacidad de desarrollar retos **STEAM** con un amplio nivel educativo: primaria hasta secundaria
 - Retos **STEAM** de nivel principiante con manejo de sensores
 - Retos **STEAM** de nivel medio adentrándose en las comunicaciones Bluetooth y wifi y almacenamiento de datos
 - Retos **STEAM** de nivel avanzado con prácticas IoT
- Realizar retos STEAM en un entorno **STEAMAKERSBLOCKS** amigable, sin complicaciones técnicas, multiplataforma y con lenguaje gráfico por bloques
 - Conocer **STEAMAKERSBLOCKS** una plataforma web que permite programar por bloques gráficos
 - Conocer las diferentes posibilidades de la programación por bloques de **STEAMAKERSBLOCKS**
 - Conocer las posibilidades didácticas de **STEAMAKERSBLOCKS** y su gestión de proyectos en el aula
- Conocer el kit robótico **TDR STEAM ESP32** que se propone, con un coste económico bajo pero de calidad para que un centro educativo lo pueda asumir para su aplicación en el aula (mínimo 12 por aula)
 - Conocer la placa **TDR STEAM ESP32**, sus sensores y actuadores
 - Aplicaciones prácticas del **TDR STEAM ESP32**

Contenidos

- **PLACA ESP32**
 - Placa ESP32, hardware y software
 - Sensores
 - Actuadores
 - Shield TDR Steam
- **STEAMAKERSBLOCKS**
 - **Retos**
 - **Telecomunicaciones**
 - **Bluetooth**
 - **IOT -WIFI**
 - MQTT
 - BLYNK IOT



Introducción

Pensamiento computacional

¿Dónde se encaja este robot? ¿Se puede comparar este robot con otros robots de otros cursos que hacemos desde CATEDU?

Esta es la hoja de ruta que proponemos, no se tiene que tomar al pie de la letra, pero intenta ayudar al profesorado que tenga una visión global de tanta oferta robótica:

Como se puede ver **SMARTHOME** tiene la ventaja de ser un proyecto maker con un amplio rango en programación en bloques y código y con prácticas avanzadas IoT es decir, un producto con muchas posibilidades.

Guía orientativa

https://docs.google.com/presentation/d/e/2PACX-1vQHiZvv1cGHet7eXVy-QcECY4Lj0k0I7ntDi8MevRWHQX-9myA0bfR5IofMeuGZkWD0Hw-Ob-MGoco_/embed?start=true&loop=true&delayms=3000

Tenemos un **grupo Telegram Robótica Educativa en Aragón**,
<https://t.me/roboticaeducativaaragon>



Introducción

Hardware ¿Qué es ESP32?

Primero : ¿Qué es Arduino?

Arduino es una tarjeta electrónica que integra básicamente a un microcontrolador y un conjunto de pines de conexión de entradas y salidas que permiten, mediante un determinado programa, interaccionar con el medio físico mediante sensores y actuadores electrónicos. De esta forma podrás crear tus propios proyectos tecnológicos, dotarlos de sensores que detecten magnitudes físicas como luz, calor, fuerza, etc... y en base a esa información, escribiendo un programa, activar otros dispositivos (actuadores) como pequeñas bombillas, ledes, servomotores, pequeños motores DC, relés, etc... Los sensores se conectan a los pines de entrada y los actuadores a los de salida.

¿Sabías que.... ? Uno de los co-creadores de Arduino es Español, de Zaragoza: **David Cuartielles** [+info](#)

Segundo ¿Qué es un microcontrolador?

Es un circuito integrado que se puede programar, o sea que puede ejecutar las órdenes que tenga almacenadas en su memoria. Tiene las tres funciones principales de un computador: la unidad central de proceso, memoria y entradas y salidas.

Arduino utiliza la marca ATMEL, y el modelo de microcontrolador depende del tipo de tarjeta, por ejemplo la tarjeta Arduino Uno utiliza el micro ATMEL MEGA 328P. Si quieres saber las entrañas de esta placa [aquí](#)

Tercero ¿Arduino tiene wifi?

Arduino (el clásico) no tiene wifi, y es importante esto para conseguir hacer prácticas IoT. Hay shields que permiten una conexión Ethernet o Wifi pero resultan caras. Otra opción es utilizar la versión del Arduino MKR1000 pero también resulta cara. A partir del 2023 lanzaron Arduino R4 Wifi pero realmente era un ESP32. Lo mejor es utilizar el ESP8266 para que a través de él nuestro Arduino pueda volcar o recibir datos a través de una Wifi.

Resumiendo brevemente, el **ESP8266** es un chip microcontrolador, es decir, no es un sensor, no es un actuador, no es una entrada/salida del Arduino sino que es, igual que el Arduino, es una placa electrónica montado en un módulo que tiene un microcontrolador (Tensilica L106 de 32bits) capaz de hacer cosas pero que tiene una característica importante: **Que tiene Wifi**, pero no lo

veas como un módulo Wifi para Arduino, sino como una placa electrónica completa, como el Arduino, incluso es su competencia.

El chip **ESP8266** se vende montado en un módulo, el más vendido es el **ESP01** y es el que se proporciona en el kit del curso **Arduinoblocks en el aula** de CATEDU.

[esp01-1.png](#)

Fuente Luis Llamas CC-BY-NC-SA [ver](#)

Por eso se habla indistintamente ESP8266 o el ESP01

Su bajo precio y su software libre permitió al mundo maker utilizar este chip. No sólo se puede utilizar para que el Arduino tenga acceso a Internet, sino también se han desarrollado multitud de módulos con el ESP8266, como veremos más adelante, el más interesante en su evolución es el módulo ESP-12E o el ESP32.

Pero sigamos con el ESP8266 montado en el módulo ESP01. Tiene unas pegas... no se diseñó para montarlo en el Arduino : **PRIMERO** La alimentación es 3V a 3.6V con picos de 200mA por lo que no puede conectarse directamente a la alimentación 3.3V y 50mA de Arduino.

SEGUNDO Consecuencia del máximo de 3.6V es que las entradas y salidas del ESP8266 no conviene conectarlas directamente a las entradas y salidas del Arduino que van a 5V.

Cuarto: ¿El ESP32 es?

Es una placa sucesora del ESP8266 de bajo coste y consumo que tiene el microprocesador Tensilica 32bits (variante el Xtensa LX6) que tiene Wifi y Bluetooth integrados. Como dice Luis Llamas en <https://www.luisllamas.es/esp32/> el ESP32 es el hermano mayor del ESP8266 con Wifi y Bluetooth.

[esp32.png](#)

Módulo ESP32 Fuente Luis Llamas CC-BY-NC-SA <https://www.luisllamas.es/esp32/>

Es posible programarlo usando código IDE de Arduino, pero también se puede programar con MicroPython.

Su principal característica es su potencial de uso en aplicaciones IoT

- Para saber más
 - Consideraciones sobre ESP32 <https://fgcoca.github.io/ESP32-STEAMakers/conesp32/>

“ Evolucion de búsquedas en Google de Arduino, Raspberry PI, ESP32 y MicroBit

-Arduino y Rpi tuvieron máximo en 2017-18. Desde entonces bajan.

-Arduino aguanta tipo

-Rpi cuesta abajo sin frenos

-ESP32 despegar [] [] [] []

-Microbit no, pese a que tuvo buen arranque (inmerecido?)

<pic.twitter.com/A2PI18qecZ>

— Luis Llamas (@LuisLlamas) [August 9, 2023](#)

Introducción

Software

Hemos visto que el ESP32 acepta código IDE Arduino, por lo tanto puede programarse como un Arduino, (acepta microPython, mBlock, etc...) pero nuestra recomendación es programarlo con **bloques gráfico** con dos opciones :

- **STEAMAKERSBLOCKS** (antes ARDUINOBLOCKS) es el programa que se usa en este tutorial
 - es **lento** cargando el programa y hay que tener bien instalado Connector
 - sólo online
 - permite programas **IoT**
 - tiene una buena **gestión de alumnado** ideal para envío de proyectos y corregirlos
 - tiene un canal de Telegram de apoyo al docente
https://t.me/innovadidactic_comunidad
- **MICROBLOCKS** El tutorial recomendado es:
<https://pedroruizf.github.io/steamakersmicroblocks>
 - no permite muchos programas IoT
 - es muy rápido al ejecutar y no da problemas de configuración
 - local o online

https://docs.google.com/presentation/d/e/2PACX-1vQb1Dv9wN9QK-F6V7yvwDoyzquqwWIGvlyVjr83Yk56kAoYD7bXlnYDm_tCQkeAgg/pubembed?start=falseloop=false&delayms=3000

Si quieres saber más sobre las diferencias entre programar con código, con bloques gráficos, en vivo, en carga... [aquí](#)

Introducción

Sensores

Esta sección es una visión rápida de las posibles **entradas** de los robots.

NO LEAS TODOS SINO LOS QUE TIENE TU ROBOT

Un poco de teoría...

Cualquier sistema de control podríamos decir que funciona de una manera similar a un ser humano, salvando las distancias. Nosotros recibimos la información del mundo exterior gracias a nuestros sentidos (oído, olfato, gusto, vista y tacto), nuestro cerebro procesa esa información y a través de nuestros músculos o de nuestra voz realizamos diferentes acciones. Pues lo mismo sucede con los sistemas de control, reciben información del exterior gracias a los diferentes **SENSORES**, procesan esa información en sus PLACAS CONTROLADORAS (sus cerebros) tales como Arduino y dan una respuesta utilizando sus diferentes **ACTUADORES**.

[2022-04-13 19_23_12-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

Un sensor es un objeto capaz de detectar magnitudes físicas o químicas y transformarlas en variables eléctricas. Los sensores o periféricos de entrada nos permiten obtener información del mundo real para utilizarla desde el programa de Arduino.

En la actualidad la cantidad de sensores disponibles es tan extensa como las variables que queramos medir, desde sensores de temperatura, humedad, luminosidad,... hasta acelerómetros, giroscopios, GPS,... pasando por detectores de gases, de pulsos cardiacos, sensores de efecto HALL,...

Tipos de sensores

- **DIGITAL:** un sensor digital sólo tiene dos estados: activado/desactivado, ON/OFF, 1/0, Alto/Bajo, ... En este caso conectaremos el sensor a una de las entradas digitales de Arduino para leer el estado.

Ejemplo: un pulsador es un tipo de sensor sencillo que sólo nos da dos estados, "pulsado o no pulsado". Conectado a la placa Arduino debe generar 0v en reposo y 5v al pulsarlo. De



esta forma desde el programa de Arduino podremos leer el estado del botón.

[2022-04-13 19_25_30-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

- **ANALÓGICO:** el sensor nos puede dar un rango de valores, normalmente se traduce en un valor de tensión o de corriente variable en función de la señal captada al sensor. En este caso conectaremos el sensor a una de las entradas analógicas de Arduino (A0,..., A5). El rango de entrada será una tensión entre 0v (GND) y 5v.

Ejemplo: Una fotorresistencia es un componente electrónico cuya resistencia disminuye con el aumento de intensidad de luz incidente. Su valor varía entre 0 y 5 v. la cantidad de valores que pueden leer las entradas analógicas de Arduino son de 10 bits es decir 1024 valores. De tal modo que 0 = 0 v. y 1023 = 5V.

[2022-04-13 19_26_06-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

- **DATOS:** el sensor ofrece su información a través de una interfaz de comunicación. La forma de comunicación puede ser por sistemas estándar como **I2C** o SPI o algunos sensores usan su propio protocolo para codificar la información y debemos realizar desde el software la decodificación correcta para interpretar los datos del sensor (normalmente los desarrolladores de este tipo de sensores ofrecen una librería software para Arduino que hace todo el trabajo).

Ejemplo: el sensor DHT11. Por un solo pin envía los datos de temperatura y humedad.

[2022-04-13 19_27_43-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

Sensores modulares.

En la actualidad existen infinidad de sensores que los fabricantes presentan en forma modular. Esto hace que su conexión y utilización sea mucho más sencilla que la tradicional, olvidándonos de resistencias, polaridades, cableados,... para su correcto funcionamiento.

Sensor pulsador

Es un sensor digital, que presenta dos estados; cuando se presiona el botón, emite una señal de bajo (0V), cuando suelta el botón, emite una señal de bajo alto (5V). [Datasheet](#)

Un ejemplo de uso

- [en el robot mClon](#)

[2022-04-13 21_58_38-Modulo pulsador con tapa, 6uds.pngimage-1649922433118.png](#)

Otra manera más "barata" de sustituir este módulo pulsador es poner un pulsador normal y una resistencia ($\pm 10k$), al pulsar se produce una entrada en el Arduino, hay dos configuraciones, que al pulsar se emita un 0 lógico (configuración **Pull up**) o que al pulsar emita un 1 lógico (configuración **Pull down**) [¿Por qué hay que poner una resistencia?](#)

[image-1650097141548.pngimage-1650097157889.png](#)

Lo "normal" es que al pulsar se emita un '1' configuración **Pull down**, pero hay pulsadores que funcionan **Pull up** y los llaman **lógica invertida**, por eso en la programación por bloques podemos encontrar esto:

[logicainvertida.png](#)

Sensor Táctil Capacitivo.

Este pequeño sensor puede "sentir" a las personas y el tacto y la retroalimentación de metales a un nivel de voltaje alto / bajo. Incluso aislado por alguna tela y papel, todavía puede sentir el tacto. Su sensibilidad disminuye a medida que la capa de aislamiento se hace más gruesa. En nuestra opinión lo preferimos frente al *Sensor pulsador* pues es muy económico, duradero y fiable.

Un ejemplo de uso en

- [Disparo láser en Arduino con ArduinoBlocks](#)
- [Apertura de puerta en Domotica con Arduino](#)

[2022-04-13 21_59_29-Interruptor táctil TTP223B.png](#)

Potenciómetro y joystick

Un potenciómetro es una resistencia variable, es decir, cambia de valor mecánicamente, lo tenemos en multitud de dispositivos. El **joystick** es internamente dos potenciómetros con un pulsador integrado en un solo mando.

[potenciometro-joystick.png2023-11-24 18_45_14-Quiero una - EchidnaShield.png](#)



Este sensor es **analógico**, su salida puede ser cualquier valor entre Vcc y GND (si está en divisor de tensión como en la placa Edubásica no llega a esos valores extremos), por lo tanto hay que conectarlo a una entrada analógica de Arduino y como cualquier entrada analógica, proporcionará valores entre 0 y 1023.

Ejemplos de uso:

- Arduino con código: [Mapeo del potenciómetro](#)
- Arduino con código: [Regular la luz con potenciómetro](#)
- [Arduinoblocks en el aula](#)
- En [Arduino con Echidna](#), con joystick
- [Domótica con Arduino con joystick](#)

Sensor Focélula LDR.

El uso de fotocélulas es muy común en nuestras vidas, las encontramos en el encendido automático de farolas, apertura de puertas,... La fotocélula es un semiconductor. Es ampliamente utilizado en campos de interruptores de control automático como cámaras, luces solares de jardín, lámparas de césped, detectores de dinero, relojes de cuarzo, tazas de música, cajas de regalo, mini luces nocturnas, interruptores de control de luz y sonido, etc. Es un sensor analógico dando valores entre 0 y 5V y como entrada analógica de un Arduino se traduce en un rango de 0 a 1023 valores.

Un ejemplos de uso :

- [el interruptor crepuscular del curso Arduino con ArduinoBlocks](#)
- [Medir la luz en Rover con Arduino](#)
- [Medir la oscuridad en Arduino con mBlock](#)
- [Hinchar un balón en Arduino con mBlock](#)

Una manera más económica de montar este sensor es utilizar una resistencia y un LDR:

- El LDR cuando más oscuridad, más resistencia
- En una configuración **PULL DOWN**, cuanto **más** luz, la resistencia del LDR baja, por lo tanto **más** tensión en A6

[embedded-image-YfsLGvOe.jpeg](#)

[2022-04-14 08_04_14-3.5 Cableado sensores _ Librería CATEDU.png](#)

Los módulos LDR que se venden suelen esta configuración Pull down, es decir, cuanto **más** luz, **más** tensión:

[image-1650370178627.png](#)

La instrucción con Arduinoblocks ya cuenta con esta configuración Pull downUp de que cuando **más** luz, **más** valor tiene la entrada analógica.

[2022-04-19 14_05_05-ArduinoBlocks.png](#)

Hay módulos LDR ya montados, que tienen componentes **activos** es decir, llevan circuitos electrónicos, transistores que amplifican etc... y dan la salida **digital** con un potenciómetro para definir el rango de luz que cambia de estado lógico. Puedes ver en la figura que tiene una salida digital **D0**.

[sensorluzarduinoigital.jpg](#)

O hay [algunos que tienen 4 pines](#) como en la figura que ofrecen las dos cosas: salida analógica **A0** y digital **D0**.

[sensorluzarduino.jpg](#)

Nosotros aconsejamos el divisor de tensión por tres razones: más barato, no implica gran circuitería y es visible su funcionamiento frente a estos encapsulados.

Sensor de Ultrasonidos.

Es un sensor digital de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 350 cm. Su uso es tan sencillo como enviar el pulso de arranque y medir la anchura del pulso de retorno.

No es un sensor preciso, con una ligera inclinación de la superficie ya da lecturas erróneas pero es muy barato

El más común es el **HC-SR04** que tiene 4 pines de conexión: **VCC Trig** (Disparo del ultrasonido)

Echo (Recepción del ultrasonido) y **GND** aunque en algunos modelos como el de [ElecFreaks](#) tiene 3 pines. Integra Trig y Echo en uno sólo.

La distancia se calcula con esta fórmula:

Distancia en cm = {(Tiempo en segundos entre Trig y el Echo) * (V.Sonido 34000 en cm/s)} / 2

Si programas en código, tienes que utilizar la fórmula anterior, previamente tienes que programar el cálculo del tiempo entre una emisión de un pulso en Trg y la respuesta en Echo.

Si utilizas la programación en bloques, no es necesario, seguro que hay un bloque que lo hace todo por ti ☐

[2022-04-14 08_10_43-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

[2022-04-14 08_11_21-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

Ejemplos de uso:

- [Alarma en Domótica con Arduino](#)
- [Piano invisible en Arduino con ArduinoBlocks,](#)
- [Sensor parking en Arduino con ArduinoBlocks](#)
- [Piano invisible en Arduino con mBlock](#)
- [Sensor parking en Arduino con mBlock](#)
- [Sensor de distancia de ultrasonidos con Picobricks](#)

Sensor DHT11 (Temperatura y Humedad).

Este sensor de temperatura y humedad **DHT11** nos permite [determinar las zonas de confort](#) para un rango de temperaturas entre 0°C y 50°C con un error de $\pm 2^\circ\text{C}$ y un rango de humedad entre 20 y 90 % $\pm 5\%$. Una salida digital para dos variables cómo lo hace? Tiene dentro un pequeño microprocesador que lanza por el bit de datos 40 bits en serie, los 16 primeros son la humedad (en BCD) y los 16 restantes es la temperatura (en BCD) los 8 restantes son de comprobación

Checksum (en binario) como la letra del DNI. Por ejemplo **0100 0111 0000 0011 0001 1001 0000 0000 0001 1000** es **0100 0111 0000 0011** = **47.03%** de humedad y **0001 1001 0000 0000** = **19.00°C** y la comprobación es la suma de $4+7+0+3+1+9+0+0=24=$ **11000**

Ejemplos de uso:

- [Medir H y T con Blink en Rover con Arduino](#)
- [Estación meteorológica Arduino con Arduinoblocks](#)
- [Arduinoblocks en el aula](#)
- [SMART HOME con Micro:bit](#)

2022-04-13 19_27_43-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png

No es un sensor con gran sensibilidad, pero para propósitos educativos cumple sus funciones. Por dentro tiene una resistencia NTC que decremanta su resistencia si aumenta la temperatura. Hay otros que van al revés, los PTC. Tanto los NTC como los PTC se llaman **thermistores**. Para la humedad, mide la capacidad de un condensador que es sensible a la humedad, o sea, un **sensor capacitivo**.

Tenemos dos opciones comerciales: **Encapsulado** que lo tienes preparado para conectar la alimentación y leer por el pin de datos, o **sin encapsular**, que hay que colocar una resistencia de aproximadamente 10k entre Vcc y Data

[dht11-pines.png](#)

Ejemplo de uso de un DHT11 sin encapsular:

[dht11sinencapsular.png](#)

Fuente Luis LLamas CC-BY-NC-SA <https://www.luisllamas.es/arduino-dht11-dht22/>

Si queremos mejorar la sensibilidad, podemos utilizar el **DHT22** que es igual pero de color blanco y más caro. Si lo que queremos es sólo la temperatura es mejor utilizar el **LM35D** que tiene un rango de temperaturas desde 0°C a 100°C con una sensibilidad de 2mV/°C

[lm35.jpg](#)

Es un sensor bastante mediocre, si necesitas una precisión el doble, te recomendamos el DHT22 que funciona exactamente igual pero es de color blanco y más caro ~3€. Ver <https://www.luisllamas.es/arduino-dht11-dht22/>

Sensor IR

Es un sensor para distancias cortas hasta 2cm y no da la distancia, simplemente si hay o no hay obstáculo, pero son muy baratos, unos 0.30€. [Aquí tienes un ejemplo de evita obstáculos en un rover marciano con Raspberry](#) Para saber más te recomendamos [esta página de Luis Llamas](#)

<https://sketchfab.com/models/6ad4f3afb83940fea95cd3846aa68a18/embed>

[IR Sensor Module for Arduino Projects | 3D Model](#) by [Veer AI](#) on [Sketchfab](#)

Sensor llama

Este sensor de llama se puede utilizar para detectar fuego u otras luces cuya longitud de onda se encuentra entre 760 nm ~ 1100nm.

Un ejemplo de su uso:

- [Alarma por fuego en Domótica con Arduino](#)

2022-04-14 08_06_56-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png

2022-04-14 08_07_38-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png

Sensor de Gas (MQ2).

Detecta gases inflamables : GLP, I-butano, propano, metano, alcohol, hidrógeno, humo... con más sensibilidad en algunos que en otros. Siempre detecta el conjunto. Son usados en electrónica de consumo y mercados industriales.

- **Sensibilidad** Tiene alta sensibilidad y se puede ajustar girando el potenciómetro.
- **Tiempo de respuesta:** Internamente posee un calentador para aumentar su temperatura y que estos gases reaccionen con la resistencia interna que tiene, por lo tanto tardan algo en responder la primera vez que se conectan, incluso horas en algunos modelos. Una vez calentados son rápidos en la respuesta.
- **Tipo de salida:** Analógico pero si tiene 4 pines como el de la figura, incorpora un pin digital.
- Ejemplos de uso:
 - [Smart Home Microbit](#)
 - [Smart Home ESP32](#)

2022-04-14 11_05_36-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png

Sensor de humedad de suelo.

“ Un higrómetro de suelo FC-28 es un sensor que mide la humedad del suelo. Son ampliamente empleados en sistemas automáticos de riego para detectar cuando es necesario activar el sistema de bombeo. El FC-28 es un sensor sencillo que mide la humedad del suelo por la variación de su conductividad. No tiene la precisión suficiente para realizar una medición absoluta de la humedad del suelo, pero tampoco es necesario para controlar un sistema de riego. Los valores obtenidos van desde 0 sumergido en agua, a 1023 en el aire (o en un suelo muy seco). Un suelo ligeramente húmedo daría valores típicos de 600-700. Un suelo seco tendrá valores de 800-1023.

Luis Llamas CC-NC-BY-SA <https://www.luisllamas.es/arduino-humedad-suelo-fc-28/>

[Se puede utilizar este sensor para hacer un dispositivo de riego automático, puede detectar si las plantas “tienen sed” y evitar que se marchiten.](#)

La corriente de trabajo del sensor es menor de 20mA. El voltaje de salida es de 0V (en el aire) a 2,3V (totalmente sumergido en agua).

- [Smart Agriculture Kit micro:bit](#)



2022-04-14 08_15_34-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png

Sensor de humedad.

Este sensor **analógico** está diseñado para identificar y detectar la presencia de agua y su cantidad. Puede servir para detectar el nivel de agua, para disparar una alarma en caso de una fuga de agua, también para hacer un limpiapalabrisas automático.... puedes ver un ejemplo de uso en :

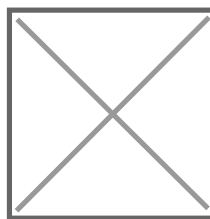
- Curso de [Domótica con Arduino](#)
- [Smart Agriculture Kit micro:bit](#)

Mide el volumen de agua caída a través de una serie de rastros de cables paralelos expuestos.

2022-04-14 08_19_22-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png
2025-11-08 18_13_20-.png

Sensor de efecto Hall.

Este es un sensor de inducción magnética. Detecta los materiales magnéticos dentro de un rango de detección de hasta 3 cm. El rango de detección y la fuerza del campo magnético son proporcionales. La salida es digital.

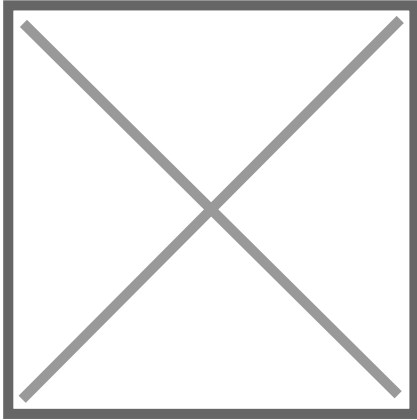


2022-04-14 10_07_05-1.3 Conexiones _ Librería CATEDU.png

Sensor Hall.

Edwin Helber Hall De Desconocido - [Popular Science Monthly Volume 64, Dominio público](#)

[Edwin Helbert Hall](#) descubrió en 1879 que en presencia de un campo magnético, un conductor que conduzca una corriente se le producía un campo eléctrico porque las cargas eléctricas se desviaban de su trayectoria principal, nuestro sensor simplemente mide ese campo eléctrico:



De [Luis Llamas](#) CC-BY-NC

El sensor tiene un led de color rojo que indica que hay una lectura de campo magnético. Un ejemplo de uso lo puedes ver aquí: [medir rocas magnéticas con el Rover con Arduino](#)

Sensor inclinación

Este sensor funciona al hacerle vibrar, emitiendo una señal digital de todo o nada. El módulo del sensor viene provisto de un potenciómetro para poder regularlo.

[2022-04-14 11_01_13-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

Sensor de golpe

Es un sensor digital que al ser golpeado este sensor envía una señal momentánea.

[2022-04-14 11_02_09-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

Sensor de pulso cardíaco.

Este módulo utiliza un LED infrarrojo (IR) ultrabrillante y un fototransistor para detectar el pulso en el dedo. Principio de funcionamiento: Se debe colocar el dedo entre el LED infrarrojo ultrabrillante (parte superior) mientras que el fototransistor, que queda en el otro lado, recoge la cantidad de luz transmitida. La resistencia del fototransistor variará levemente a medida que la sangre pase a través de su dedo.

[2022-04-14 11_03_47-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

Sensor de Alcohol.

Este sensor de gas analógico MQ-3 es adecuado para detectar alcohol. Se puede usar en un analizador de aliento. También tiene una alta sensibilidad al alcohol y baja sensibilidad a la bencina

(éter de petróleo). La sensibilidad se puede ajustar con el potenciómetro.

[2022-04-14 11_04_42-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

Sensor de CO2

Hay sensores que utilizan **el protocolo I2C**, este protocolo permite conexiones serie y pueden compartir el mismo cable pues cada elemento tiene una dirección diferente. Esto lo veremos en el Display LCD. Se identifican por los pines SDA y SCL

CCS811-KEYSTUDUUDIO.png

Resistencia Flex

Es una resistencia que cuanto más se dobla más resistencia ofrece, desde 25k hasta 125k

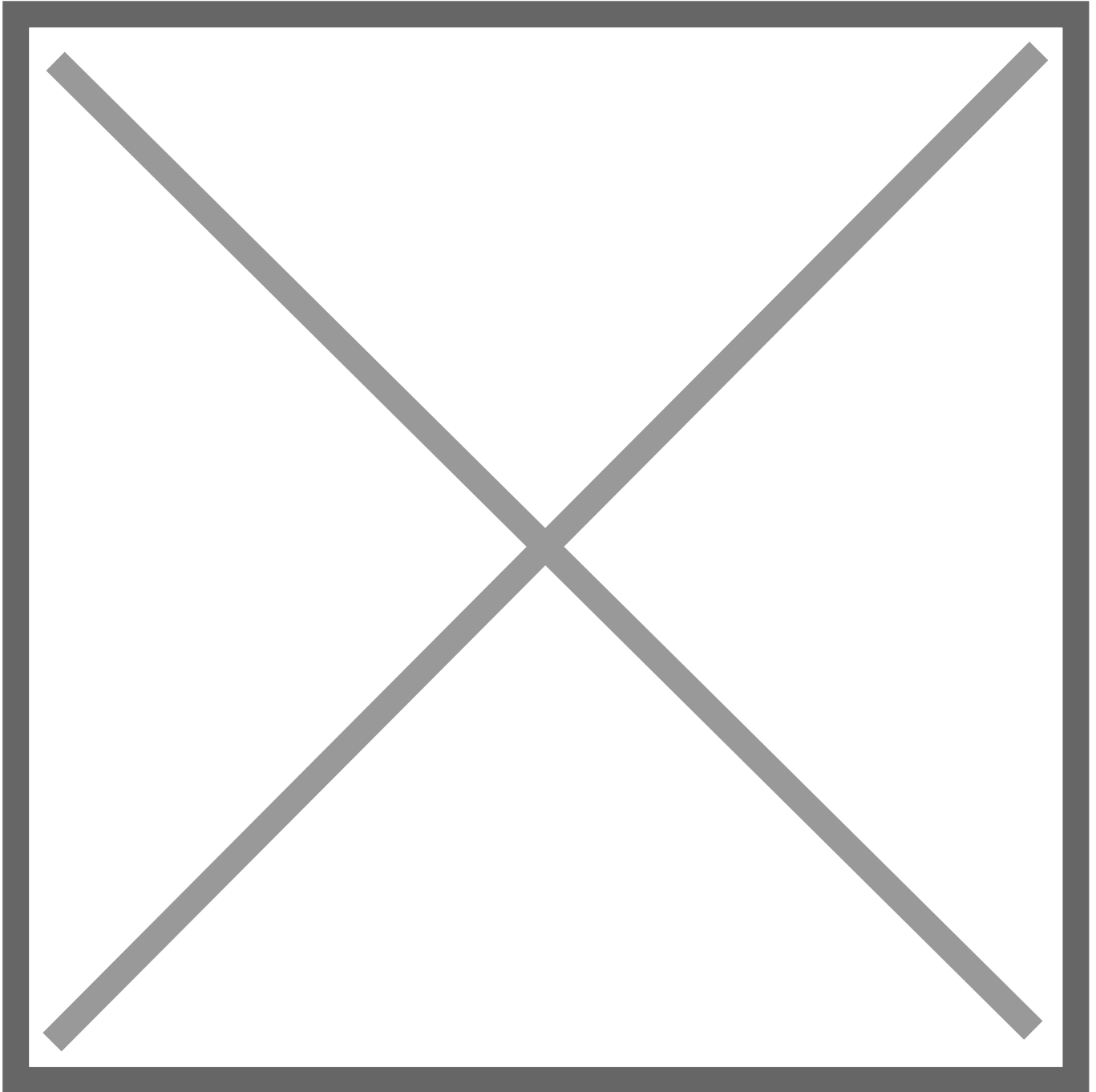
[Datasheet Sparkfun](#)

Para utilizar esta resistencia haremos un **DIVISOR DE TENSION** que consistirá en poner dos resistencias en serie y repartirá la tensión total entre 0V y 5V en las dos resistencias, **el punto medio** será un punto que tendrá una tensión variable en función de las dos resistencias, como la es variable, esa tensión es variable y ya tenemos la entrada **analógica**:



Es decir:

- La resistencia entre masa GND del ARDUINO (cable negro) y un punto en la placa protoboard
- ese punto medio conectarlo a una entrada analógica, por ejemplo A0 (cable amarillo)
- Una resistencia de valor parecida a la Flex de decenas de K entre ese punto y +5V (cable rojo en la foto)



Este sensor tiene posibilidades para usarlo en "ropa inteligente".

Sensor de movimiento

“ Los sensores infrarrojos pasivos (PIR) son dispositivos para la detección de movimiento. Son baratos, pequeños, de baja potencia, y fáciles de usar. Por esta

razón son frecuentemente usados en juguetes, aplicaciones domóticas o sistemas de seguridad.

Los sensores PIR **se basan en la medición de la radiación infrarroja**. Todos los cuerpos (vivos o no) emiten una cierta cantidad de energía infrarroja, mayor cuanto mayor es su temperatura. Los dispositivos PIR disponen de un sensor piro eléctrico capaz de captar esta radiación y convertirla en una señal eléctrica. En realidad **cada sensor está dividido en dos campos** y se dispone de un circuito eléctrico que compensa ambas mediciones. Si ambos campos reciben la misma cantidad de infrarrojos la señal eléctrica resultante es nula. Por el contrario, si los dos campos realizan una medición diferente, se genera una señal eléctrica.

De esta forma, si un objeto atraviesa uno de los campos se genera una señal eléctrica diferencial, que es captada por el sensor, y se emite una señal digital.

[2026-02-09 12_25_59-Greenshot.png](#)

El otro elemento restante para que todo funcione es **la óptica del sensor**. Básicamente es una cúpula de plástico formada por lentes de fresnel, que divide el espacio en zonas, y enfoca la radiación infrarroja a cada uno de los campos del PIR.

De esta manera, cada uno de los sensores capta un promedio de la radiación infrarroja del entorno. Cuando un objeto entra en el rango del sensor, alguna de las zonas marcadas por la óptica recibirá una cantidad distinta de radiación, que será captado por uno de los campos del sensor PIR, disparando la alarma.

[2026-02-09 12_27_59-Inbox - jjquintana@educa.aragon.es - Mozilla Thunderbird.png](#)

Luis Llamas CC-BY-NC-SA <https://www.luisllamas.es/detector-de-movimiento-con-arduino-y-sensor-pir/>

Puedes ver ejemplos de uso en robótica en :

- [Smart Home para Microbit](#)
- [Smart Agriculture Kit para Microbit](#)

[2022-04-16 10_24_22-sensor pir arduino at DuckDuckGo.png](#)[2025-11-08 18_11_48-.png](#)

Más sensibles son los **sensores de microondas**. Son un radar que por efecto Doppler pueden captar cualquier objeto en movimiento dentro de un alcance de 5-7 metros en cualquier dirección e independiente de su temperatura. Es un buen sensor para alarmas, activación de luz por presencia..... Para saber más [ver la página de Luis Llamas](#)

[image-1650097316026.png](#)

Su conexión es muy sencilla, es un detector digital que hay que alimentarlo como el resto de sensores.

[image-1650097889205.png](#)

Curiosamente, la gran ventaja técnica de los de microondas **es un gran inconveniente para usarlo en el aula**, con cualquier movimiento se dispara, luego **para clase es mejor el sensor PIR**

Esta página esta adaptada de [este enlace](#). José Andrés Echevarría @cantabRobots CC-BY-NC-SA.

Introducción

Actuadores

Esta página es un repaso rápido de las posibles **salidas** de los robots
NO LEAS TODOS SÓLO LOS QUE TIENE TU ROBOT

¿Qué es un actuador?

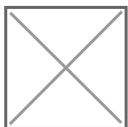
“ Un actuador es el nombre genérico que empleamos para cualquier **dispositivo capaz de realizar acciones en el mundo físico** y que podremos controlar desde un autómata o procesador como Arduino. En particular, usamos el nombre actuador para los dispositivos que son capaces de generar movimiento.

[Luis LLamas](#) CC-BY-NC-SA

ACTUADORES

Motores baja potencia

Con el Arduino podemos usar motores de corriente continua de juguete como en la figura, aptos para poner un ventilador. Ejemplo de uso [Smart Home para microbit](#)

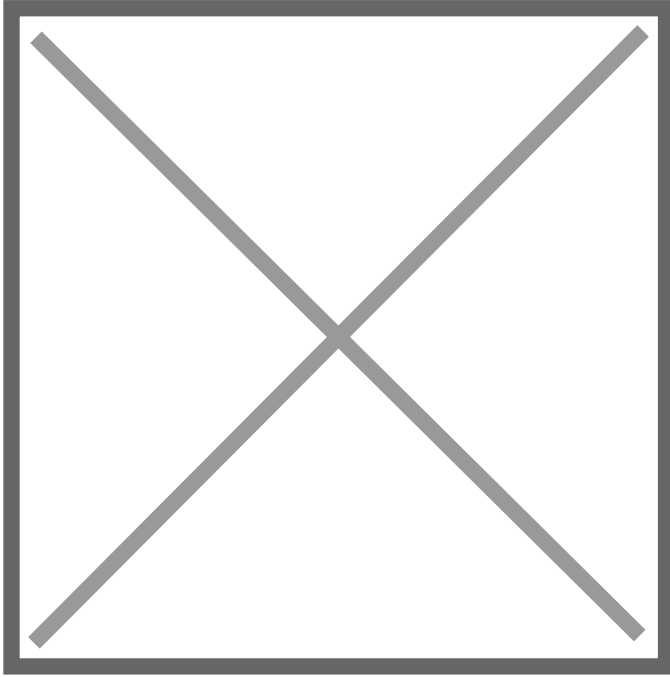


Motores con algo de potencia

Si quieres usar un motor para mover unas ruedas, ya pide más corriente, no se puede conectar directamente a la placa (Arduino, esp32 etc..). El de la figura es típico en robótica y lleva reductores dentro para bajar la velocidad :

[2022-04-16 10_40_41-motor arduino at DuckDuckGo.png](#)

Necesita una alimentación extra, como el LM298N y las pilas., o el B6612FNG o una shield específica.



CC-BY-NC [Luis Llamas](#)

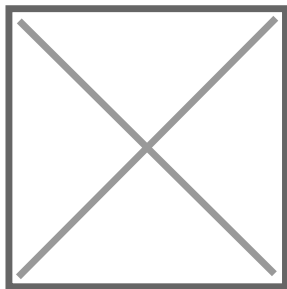
Ejemplos de uso lo puedes ver en

- [Curso mClon con nanoArduino con el B6612FNG](#)
- [Curso Rover con Arduino con la shield L293D ESP-12E para el NODE MCU que internamente tiene el LM298N](#)
- [Curso Arduino con código con LM298N](#) y también con un transistor

Los motores pueden (y deben) de estar conectados a **salidas PWM** de tal manera que se pueda regular la potencia y por lo tanto la velocidad.

Servos

Un [servo convencional](#) es un motor donde fijamos el ángulo desde 0º a 180º, pero si queremos una rotación, existen [servos rotatorios](#) que simplemente tienen su velocidad de rotación controlada



Ejemplos de uso de servos:

- [Curso Arduino con código](#)
- [Curso brazo robótico del mClon con nanoArduino](#)
- [Apertura de barrera por ultrasonidos en curso Arduino con ArduinoBlocks](#)
- [Tractor entrando en el corral Arduino con EchidnaShield](#)
- [Apertura de puerta Domótica con Arduino](#)
- [Servo con PicoBrick](#)
- [Apertura ventana y puerta en Smarth Home para microbit](#)
- [Smart Agriculture Kit para micro:bit](#)

Relés

Un relé es un interruptor activado por un electroimán, lo que permite independizar los circuitos. En el dibujo se ve que el circuito rojo de 220V está separado del verde, de sólo 1.5V. Pero es el circuito verde que al funcionar, hace que el electroimán mueva el interruptor del relé y encienda la bombilla. El objetivo es que he podido encender una bombilla de 220V sin tocar los 220V peligrosos. En el circuito verde, el interruptor puede ser un Arduino. Experimentalo en este [simulador](#).

- Un ejemplo con Arduino en [Luis LLamas](#)
- Un ejemplo de uso en [Picobricks](#)

2025-01-09 18_42_34-Simple Relay with bulb - Simulador De Circuito On Línea _ DCACLab.png

Electroimán

El electroimán no se puede conectar directamente, utilizaremos una etapa de potencia, por ejemplo el transistor el [IRF520N](#) que amplifique la señal del Arduino, pues las salidas de Arduino no tienen potencia para mover el electroimán. Para dar esa potencia utilizaremos otra fuente externa, unas pilas:

[image-1649970623453.png](#)

Las conexiones son :

- SIG del IRF520N a una salida digital por ejemplo D13
- VCC del IRF5020N al 5V del ARDUINO
- Los dos GND del IRF520N a GND del ARDUINO
- V+ y V- del IRF5020N al solenoide, da igual qué cable pues no tiene polaridad.
- VIN del IRF520N al VIN del ARDUINO (son los voltios de la pilas)

[image-1649970676310.png](#)

Motor paso a paso

Igual que el [electroimán](#), necesitamos un controlador que nos de la potencia necesaria para mover el motor, el ULN2003



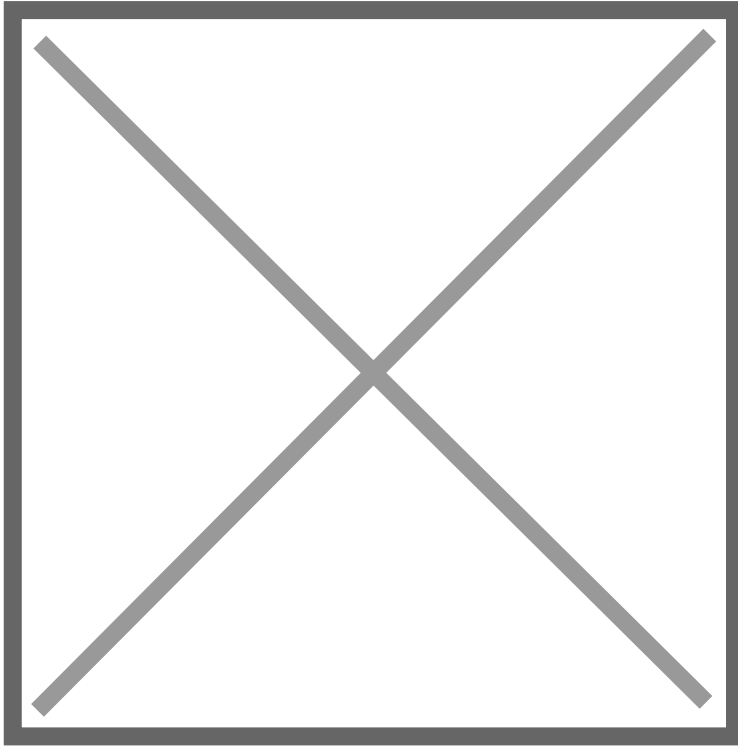
Imagen BY-NC-SA de [Luis Llamas](#)

También igual que el electroimán, necesitamos una potencia extra con las pilas:



Conexión :

- Cuatro pines digitales del ARDUINO al IN1,IN2,IN3,IN4 del ULN2003 por ejemplo D10,D11,D12,D13
- El conector blanco del ULN2003 al motor paso a paso
- El (+) del ULN2003 al Vin del ARDUINO
- El (-) del ULN2003 al GND del ARDUINO



La configuración más sencilla es la rotación simple en sentido horario (llamada fase1) :



Imagen BY-NC-SA de [Luis Llamas](#)

Es decir:

Paso	IN1=D10	IN2=D11	IN3=D12	IN4=D13
Paso 1	ON	OFF	OFF	OFF
Paso 2	OFF	ON	OFF	OFF
Paso 3	OFF	OFF	ON	OFF
Paso 4	OFF	OFF	OFF	ON

Si has entendido este ejemplo serás capaz de realizar las configuraciones que desees.

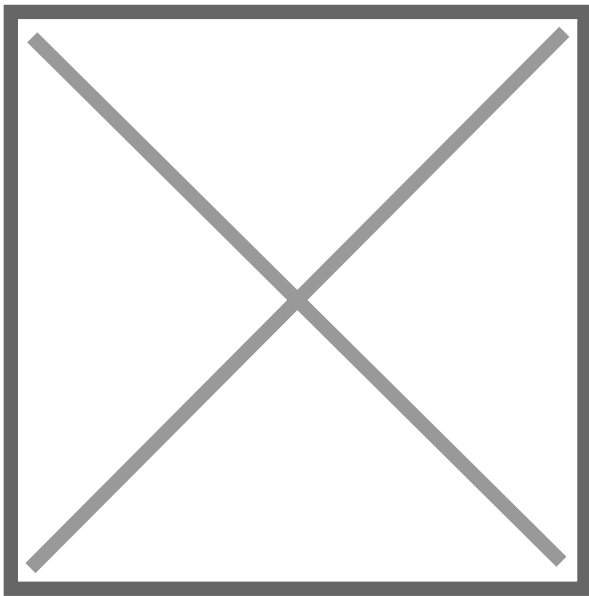
OTRAS SALIDAS

No son actuadores pues no representan movimiento pero son también **salidas** del Arduino como los actuadores, luego lo vemos en esta página

Buzzer activo

Reproduce un tono de una frecuencia determinada por el fabricante cuando recibe un '1' digital. Su conexión es muy simple:

- **GND** a GND y **VCC** a 5V (ojo que están a los dos extremos, marcados en rojo y en negro)
- **OUT** o también señalado como I/O a un pin digital, por ejemplo D13



Ejemplos de uso:

- Domótica con Arduino: [Apertura puerta con mBlock](#)
- Domótica con Arduino [Apertura puerta con código](#)

Buzzer pasivo

La diferencia de un buzzer activo con el pasivo es que el pasivo hay que mandar la onda que se tiene que reproducir, como Arduino no puede reproducir onda puras (senoidales analógicas), se le envía ondas cuadradas con la frecuencia que se pretende reproducir. Como se puede ver en la figura, no tiene elementos auxiliares para reproducir un tono, es simplemente un altavoz.

[zumbadorpasivo.png](#)

Ejemplos de uso



- [Timbre Arduino con Echidna](#)
- [Arduino blocks en el aula](#)

LEDS Y OTROS ELEMENTOS VISUALIZADORES

Led normal

Un diodo es la unión de dos semiconductores P-N que sólo permite la corriente en un sentido. Un diodo LED es un diodo, que al pasar esta corriente emite una luz. Hay de 3mm y de 5mm de grosor, transparentes, traslúcidos y de colores (realmente son traslúcidos tintados)

Tenemos pues que respetar su polaridad para que funcione:

[arduino-led-patillaje.png](#)

Fuente Luis LLamas CC-BY-NC [Encender LED con Arduino](#)

Si alimentamos el LED con una tensión inferior a su tensión de polarización directa $V_d \pm 1,4V-3,8V$ el led no luce. si alimentamos con una tensión superior, la corriente que circula se dispara por lo que se rompe. Conclusión: Hay que poner una resistencia limitante. Para calcularla te recomiendo la página de Luis Llamas [Encender LED con Arduino](#)

Aquí puedes ver un ejemplo de semáforo [en el curso Arduino con código](#)

[image-1650005625137.png](#)[image-1650005638493.png](#)

Se puede conectar directamente pues las salidas de Arduino tienen una limitación interna de 20mA como en este ejemplo <https://www.youtube.com/watch?v=EFFSLvIF9rY>

<https://www.youtube.com/embed/EFFSLvIF9rY>

Los LEDS igual que los motores pueden ser encendidos o todo o nada con salidas digitales 0 y 1 o salidas PWM de esta manera conseguimos luminosidades medias, en esta práctica con el potenciómetro se regula la intensidad de la luz de un LED

Led RGB

Se trata de un encapsulado que integra 3 leds de los tres colores básicos. Cada color básico se puede regular desde 0 a 255 de intensidad, luego tenemos 16 millones de tonalidades diferentes.

[image-1650007895867.png](#)

Un ejemplo de su uso lo puedes ver en

- [Domótica con Arduino, interruptor crepuscular](#)
- [Intermitente RGB en Arduino con ArduinoBlocks](#)

Existen LEDS RGB que se comunican unos a otros, ([ver este ejemplo en el curso mClon](#)) como las tiras de leds RGB que cambian de color todos simultáneamente.

Neopixel

La cinta LED RGB se basa en el chip controlador WS2812B de Neopixel con el cual podemos tener el control de cada pixel y de cada color, ya sea rojo, verde y azul a través de un solo cable de datos.

“ Los WS2811, WS2812 y WS2812B son LED que disponen de lógica integrada, por lo que es posible variar el color de cada LED de forma individual (a diferencia de las tiras RGB convencionales en las que todos los LED cambian de color de forma simultánea). Están basados en el LED 5050, llamado así porque tiene un tamaño de 5.0 x 5.0 mm.

Es un LED de bajo consumo y alto brillo, que incorpora en un único encapsulado los 3 colores RGB. La genial novedad del WS2812B (y resto de familia) es añadir un integrado dentro de cada LED, que permite acceder a cada pixel de forma individual. Por este motivo este tipo de LED se denominan “individual addressable”. Esto abre la puerta a un sinfín de aplicaciones y combinaciones, que van desde dotar de iluminaciones distintas zonas con una única tira, animaciones complejas, o incluso generar pantallas enteras de alta luminosidad....

A los LED WS2812B también se les denomina NeoPixel

Luis Llamas CC-BY-NS-SA <https://www.luisllamas.es/arduino-led-rgb-ws2812b/>

[2025-11-08 18_27_31-.png](#)



- [Smart Home para Microbit con 4 leds RGB Neopixel](#)
- [Smart Home para ESP32](#)
- [Smart Agriculture Kit micro:bit](#)

Láser

El diodo láser es un elemento motivador, barato y fácil de usar con el Arduino pues se activa digitalmente, si enviamos un 1 emite un rayo láser. Un ejemplo de uso lo puedes ver en la [Alarma por láser en Domótica con Arduino](#)



Si quieres saber más de este componente, te recomendamos [esta página de Luis Llamas](#).

Si tienes que comprar uno, es importante que no sea superior a 5mW, pues puede dañar permanentemente la retina del ojo [[+info](#)].

El modelo que proponemos es de **1mW**, no obstante, **EVITA SIEMPRE QUE EL LÁSER APUNTE A LOS OJOS** especialmente con niños.

[image-1649971311304.png](#)

Display LCD

Si queremos visualizar datos de forma independiente del ordenador, existen matrices de led que puedes ver en los cursos de mBot o mClon pero lo mejor es una pantalla Liquid Crystal Display y la mejor forma de conectar esta interfaz es en serie utilizando el **protocolo I2C**.

Ejemplo de uso :

- en [Arduino con código](#)
- [Arduinoblocks en el aula](#)
- [Smart Home para Microbit](#)

[image-1650096059762.png](#)

CC-BY-SA [Luis Llamas](#)

No confundas Display LCD con matriz de LEDs , o una OLED

Aclarando conceptos: Protocolo I2C

Es un protocolo de emitir los datos con sólo dos cables **SDA** (los datos) **y SCL** (reloj) y además pueden compartir varios dispositivos SLAVE) para ello la placa Arduino ESP32, microbit... (MASTER) tiene que saber **qué dirección tiene cada dispositivo IMPORTANTE:**

2025-11-08 18_35_50-.png

[Fuente Luis Llamas ArduinoI2C CC-BY-SA](#)

Aclarando conceptos: Lógica invertida

Igual que vimos en sensores, con las configuraciones Pull up y Pull down, las salidas de cualquier actuador **PUEDEN SER LÓGICA INVERTIDA**, mira en este NodeMCU (básicamente es un Arduino con wifi, aquí está conectado con el servidor Blynk) como funciona al revés, cuando apretamos, por lo tanto enviamos un uno a la salida, **se apaga la salida !!**

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

Montaje

Montaje

Preliminar

La casa nos lo venden a piezas, y es más difícil de montar que un mueble de Ikea

NO RECOMENDAMOS REALIZAR EL MONTAJE CON LOS ALUMNOS. La probabilidad de una rotura podría hacer la maqueta totalmente inservible.

Ojo, que quede bien claro que nosotros estamos para ayudarte en tu formación, no somos comerciales de este robot. O sea, esto mejor que no:



Pero te queremos animar:



Luego al final el resultado es muy satisfactorio :

[2025-10-21 23_52_30-WhatsApp.png](#)

Montaje

Las paredes

Tienes dos sitios para seguir todo el proceso de montaje:

- Página del comercial Innovadidactic
https://docs.innovadidactic.com/es/manuals/smart_home/mounting
- La documentación del fabricante en
<https://docs.keyestudio.com/projects/KS5009/en/latest/docs/index.html>

Vamos siguiendo los pasos, aquí sólo vamos a nombrarlos y decir algunas observaciones:

Pared principal

Los tornillos de la placa LCD son los largos estrechos, y hay que fijarse bien en la orientación, que los pines queden hacia la izquierda

El resto de componentes van con los tornillos normales :

[2025-10-22 08_13_22-Greenshot.png](#)

Fuente [Innovadidactic](#) con permiso del comercial

El sensor RFID puede ir por dentro como en la figura de arriba o exterior como esta foto:

[2025-10-21 23_20_16-WhatsApp.png](#)

Tejado

El techo no tiene muchas complicaciones y son los tornillos normales, tiene dos paneles :

[2025-10-22 08_17_56-Documento1 - Word.png](#)

Fuente [Innovadidactic](#) con permiso del comercial

Pared trasera

Todos son tornillos normales, menos el del portapilas, que debe de tener la cabeza plana (en caso contrario, impide la colocación de las pilas)



2025-10-22 08_18_33-Greenshot.png

Fuente [Innovadidactic](#) con permiso del comercial

También hay que tener en cuenta que el cable de alimentación tiene que salir hacia el borde de la pared para que llegue a la placa, es decir, orientar el cable del portapilas hacia la izquierda.

2025-10-21 23_26_09-WhatsApp.png

Detalle de los tornillos cabeza plana y colocación del cable portapilas

Pared lateral con ventana

Esta pared tiene su dificultad pues tiene un servo, con una rueda dentada transparente y una ventana transparente que se tiene que mover con libertad:

2025-10-22 08_22_26-Greenshot.png

Fuente [Innovadidactic](#) con permiso del comercial

La colocación de los tornillos de la ventana, van con unas tuercas autobloqueantes.

2025-10-21 23_37_01-WhatsApp.png

**NO TIENES QUE APRETAR MUCHO. LA VENTANA TIENE QUE TENER LIBERTAD DE MOVERSE
TAMPOCO TIENE QUE ESTAR SUELTO**

En <https://docs.keyestudio.com/projects/KS5009/en/latest/docs/index.html> recomiendan poner la rueda dentada, con 4 tornillos normales, que no pueden apretarse por el peligro de atravesar el tornillo a la otra parte y por lo tanto impedir su movimiento. Nosotros recomendamos usar los tornillos en punta los más pequeños, y como también lo utilizaremos en otro servo, y no hay 8, entonces con 2 es suficiente:

2025-10-21 23_41_04-WhatsApp.png

ATENCIÓN: NO PONER EL TORNILLO CENTRAL.

Pues ¿Con qué giro lo colocamos? En teoría estará a 0º por lo tanto ventana cerrada, tenemos que dejar la ventana cerrada pero cuando hagamos el reto 5.2 nos aseguramos que está en la posición correcta por lo tanto pondremos el tornillo.

SUELO CON LA PLACA

Es un suelo interior que contiene la placa ESP32 y el servo que moverá la puerta

[2025-10-22 08_23_21-Greenshot.png](#)

Fuente [Innovadidactic](#) con permiso del comercial

Igual que antes, recomendamos usar los tornillos en punta más pequeños que fijan muy bien el servo con la rueda. Como no hay suficientes ponemos 3

[2025-10-21 23_50_54-WhatsApp.png](#)

Montaje

Montando las paredes

Lo primero es hacer una **U** con la pared principal y las dos paredes que tiene ventanas (la que hemos montado el cristal que se mueve y otra pared sin ningún elemento electrónico). Todos los elementos se fijan con unas **T** de madera fáciles de poner aunque a veces se resisten.

[di1.png](#)

Fuente <https://docs.keyestudio.com/projects/KS5009/en/latest/docs/index.html>

Luego uniremos con

- La pared trasera que tiene el portapilas
- El suelo que tiene la placa (**OJO** no te olvides)
- Un lateral del tejado que tiene las luces

[di1.png](#)

Fuente <https://docs.keyestudio.com/projects/KS5009/en/latest/docs/index.html>

El siguiente elemento **NO LO PONDRÍA PUES DIFICULTA LAS CONEXIONES. DEJARLO PARA DESPUÉS DE REALIZAR EL CABLEADO.**

[di1.png](#)

Fuente <https://docs.keyestudio.com/projects/KS5009/en/latest/docs/index.html>

El siguiente elemento es difícil de visualizar, es unir la base con una puerta giratoria con el suelo que tiene la placa

[di1.png](#)

Fuente <https://docs.keyestudio.com/projects/KS5009/en/latest/docs/index.html>

A lo mejor con una foto, ayuda a cómo tiene que ser :

[2025-10-22 00_12_33-WhatsApp.png](#)

Finalmente encajamos la rueda dentada especial en la parte superior de la puerta giratoria.



di1.png

Fuente <https://docs.keyestudio.com/projects/KS5009/en/latest/docs/index.html>

Montaje

Conexiones

RECOMENDAMOS NO TENER PRISA Y UTILIZAR

- 1 BUENA ILUMINACIÓN
- 2 GAFAS - LUPA
- 3 PINZAS O TENAZAS PUNTA FINA
- 4 NO MONTAR EL TEJADO
- 5 CHICLES-CARAMELOS PARA ARMARSE DE PACIENCIA

[2025-10-21 23_43_54-WhatsApp.png](#)

En <https://docs.keyestudio.com/projects/KS5009/en/latest/docs/index.html> hay fotos para ver PASO A PASO TODAS LAS CONEXIONES con fotos bien claras de donde van cada sensor y actuador.

EL SENTIDO DE LOS CABLES ES MUY FÁCIL: LA MASA G - GND SIEMPRE EL CABLE NEGRO

Aquí sólo vamos a nombrar de forma escueta donde va cada elemento y qué cable utilizar:

ELEMENTO	A QUÉ PIN DEL ESP32	CABLE	OBSERVACIONES
Sensor humedad DHT11	IO17	3 polos corto 15cm	
LED amarillo	IO12	3 polos corto 15cm	
Sensor vapor	IO34	3 polos corto 15cm	
Ventilador	IN- AL IO18 IN+ AL IO19	4 polos suelos en los dos extremos	ver detalle conexión ventilador
Sensor PIR	IO14	3 polos corto 15cm	
Botón izquierdo	IO16	3 polos largo 20cm	ojo tiene que ser el largo
Botón derecho	IO27	3 polos largo 20cm	idem

ELEMENTO	A QUÉ PIN DEL ESP32	CABLE	OBSERVACIONES
RFID	IIC	4 polos	da igual a qué IIC lo pongas, tienes 4 conectores
LCD1602	IIC	4 polos	da igual a qué IIC lo pongas, tienes 3 libres, el otro lo has ocupado con el RFID
LEDS RGB	IO26	3 polos corto 15cm	
Sensor GAS	IO23	3 polos largo 20cm	ojo tiene que ser el largo
buzzer	IO25	3 polos largo 20cm	idem
servo ventana	IO5	tiene su cable	el marrón es la masa luego ya esta claro la forma de conectarlo
servo puerta	IO13	idem	idem
la alimentación del portapilas			sólo hay un sitio

Resulta difícil localizar los pines, aquí tienes una imagen ampliada del ESP32

[2025-10-22 00_45_49-WhatsApp.png](#)

Un resumen

[2025-10-22 08_25_30-.png](#)

Fuente [Innovadidactic](#) con permiso del comercial

Detalle conexión ventilador

Son 4 cables, viendo la foto :

[2025-10-22 00_27_51-WhatsApp.png](#)

- El negro masa G
- El blanco la alimentación V
- El gris al IN+
- El morado al IN-

Esto se tiene que conectar

1. bajar el cable de 4 polos
2. Desplazar el morado IN- al IO18
3. El gris IN+ al IO19
4. El blanco a V como es alimentación al macho rojo del IO19 (vale también el del IO18)
5. La masa al macho negro del IO19 (vale también el del IO18)

Los pasos 3-4-5 están alineados al IO19

El que esta separado es el paso 2

[2025-10-22 00_25_56-WhatsApp.png](#)

AHORA YA PUEDES PONER EL TEJADO.

[di1.png](#)

Cómo programar la Smart Home ESP32

No utilices su programa predeterminado

La placa viene con un programa predeterminado para comunicarse con la [app keyes IOT home](#)

2025-10-22 09_33_09-Inbox - jjquintana@educa.aragon.es - Mozilla Thunderbird.png

En la placa ESP32 [hay un programa que viene de fábrica y sólo sirve para comunicarse con la APP](#)

P: ¿Y qué se consigue didácticamente?

R: Pues nada

Si grabamos el programa de fábrica y hacemos funcionar la maqueta con la APP convertimos el Smart home ESP32 en un juguete en vez de un instrumento didáctico para fomentar el pensamiento computacional y aprender a programar.

¿Qué opciones tengo?

- Puedes utilizar programación por bloques, recomendado para primaria, y primer ciclo de secundaria.
Tienes muchas opciones de programas pero recomendamos **Steamakerblocks**
- O la programación por código, en ese caso puedes utilizar programación ArduinoIDE pero recomendamos **Python**

Para saber más sobre las diferencias y opciones de programación en bloques o programación en código para placas compatibles con Arduino, mira

<https://libros.catedu.es/books/programa-arduino-mediante-codigo/page/software>



STEAMAKERSBLOCKS

STEAMAKERSBLOCKS

Qué es Steamakersblocks

Esto no pretende ser un tutorial exhaustivo de STEAMAKERSBLOCKS, sino una guía rápida. STEAMAKERSBLOCKS es un programa que tiene muchas posibilidades. Si quieres saber más sobre ARDUINOBLOCKS tutoriales, ejemplos, foro.... te recomendamos

<http://arduinoblocks.didactronica.com/> o el libro [Arduino blocks - libros y tutoriales](#)

IMPORTANTE SABERLO: TENEMOS UN CHAT DE Ayuda en STEAMAKERSBLOCKS

hay chat de **Telegram** con una comunidad de profesores y técnicos de la empresa que apoya Arduinoblocks donde puedes encontrar proyectos, enlaces interesantes y lo más importante: Puedes preguntar tus dudas o problemas

https://t.me/innovadidactic_comunidad

logogrupotelegramarduinoblocks.jpg

¿Por qué una programación con bloques?

Arduino se programa en lenguaje C++ (con algunas variaciones para simplificarlo). Para programar normalmente se utiliza el IDE ("Integrated Development Environment"/"Entorno de Desarrollo Integrado") de Arduino, que permite escribir el código, compilar el programa (crear el programa binario para el procesador Arduino) y grabarlo en la placa Arduino a través del puerto USB. El IDE de Arduino se puede descargar [desde la web oficial](#). Es totalmente libre (José Andrés Echevarría @cantabRobots CC-BY-NC-SA)

2022-04-13 12_45_21-PROYECTO 00 CONOCEMOS ARDUINO. - PDF-XChange Viewer.png

(José Andrés Echevarría @cantabRobots CC-BY-NC-SA)

Sin embargo pensando en edades más tempranas se han desarrollado formas más sencillas e intuitivas de programar Arduino como son los **lenguajes de programación por bloques**. De todos estos lenguajes cabe destacar **STEAMAKERSBLOCKS**.

Gracias a este lenguaje visual podemos programar las placas Arduino sin necesidad de escribir ni una sola línea de código, de esta forma podemos empezar a realizar proyectos con Arduino de una forma muy rápida y a edades más tempranas. La única desventaja es que el lenguaje por código

tiene todo el potencial que requiere la programación de un experto.

Mismo programa creado con el IDE de Arduino en C++ (imagen de la izquierda) y con Arduinoblocks (imagen de la derecha).

(José Andrés Echevarría @cantabRobots CC-BY-NC-SA)

[2022-04-13 12_47_24-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

José Andrés Echevarría @cantabRobots CC-BY-NC-SA)

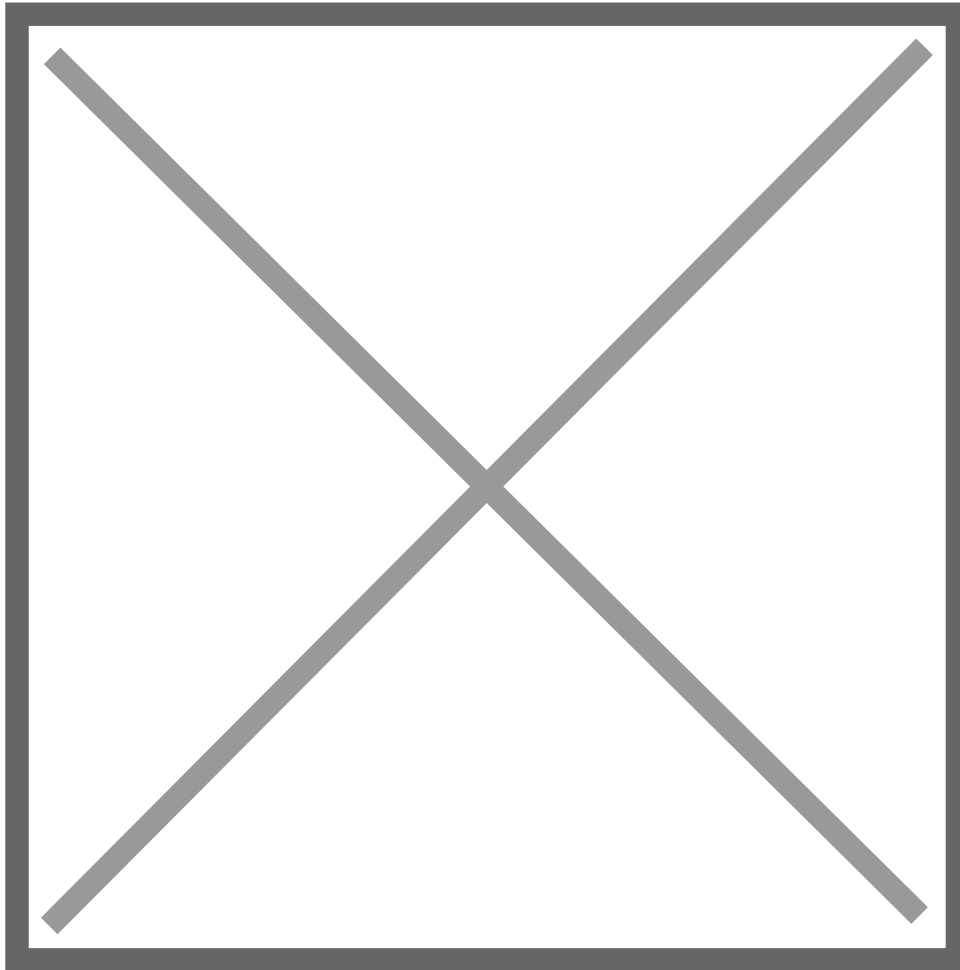
Para trabajar con Arduinoblocks debemos ir a su página web <http://www.arduinoblocks.com/> desde cualquier navegador y para cualquier sistema operativo (Windows, Linux, Mac). *(José Andrés Echevarría @cantabRobots CC-BY-NC-SA)*

STEAMAKERSBLOCKS (antes Arduinoblocks)



2025-10-22 10_31_17-.png antes

STEMAKERSBLOCKS es un programa creado por el profesor Juanjo López. Gracias a su entorno gráfico facilita la programación de placas Arduino a todos los niveles. Esta herramienta permite programar a personas sin conocimientos previos de programación, pero su versatilidad y potencia es tan grande que expertos programadores también pueden utilizarlo. *(José Andrés Echevarría @cantabRobots CC-BY-NC-SA)*



De [Juan José López Almendros](#) CC-BY-SA

La programación en **STEMAKERSBLOCKS** se realiza con bloques al estilo AppInventor o Scratch, *se puede utilizar a partir de 8 años*. No tenemos que escribir líneas de código y no nos permitirá unir bloques incompatibles evitando así posibles errores de sintaxis. La plataforma **STEMAKERSBLOCKS** genera, compila y sube el programa a la placa Arduino por medio de la conexión USB. Una vez subido el programa, la placa el Arduino no necesitará de la conexión al PC para funcionar pudiendo alimentarla con baterías o una fuente de alimentación para que funcione de forma autónoma.

STEMAKERSBLOCKS actualmente funciona con todos los navegadores de última generación: Mozilla Firefox, Google Chrome, Opera, Safari,... (José Andrés Echevarría @cantabRobots CC-BY-NC-SA)

Por otro lado, tal y como se describe en la [Wiki](#) de **Vitalinux**, **STEMAKERSBLOCKS** funciona perfectamente con todos los sistemas operativos, pudiendo ser fácilmente instalable en equipos individuales y a nivel de centro dentro del soporte de **Vitalinux**.



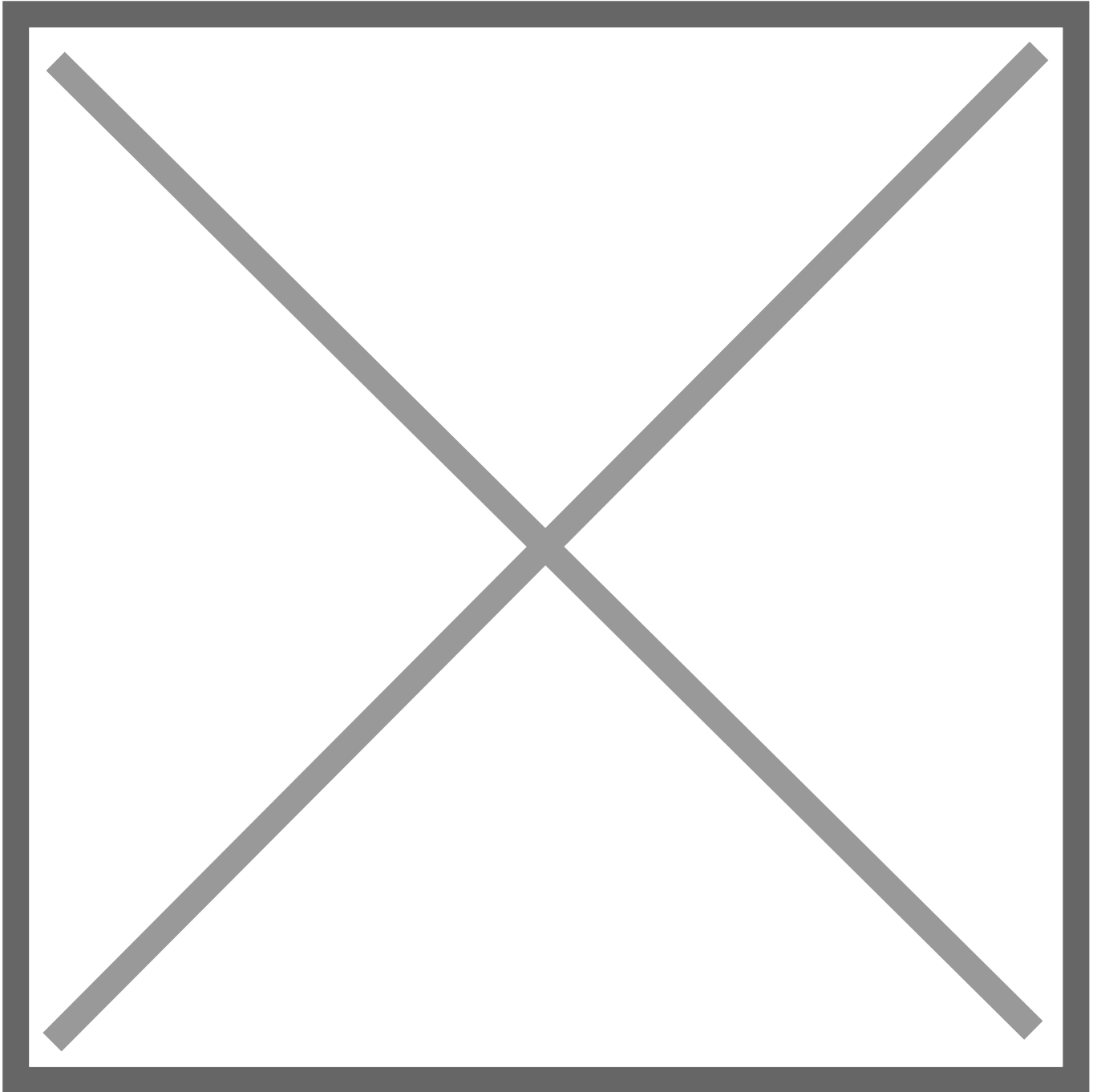
STEAMAKERSBLOCKS

Crear cuenta en Steamakerblocks

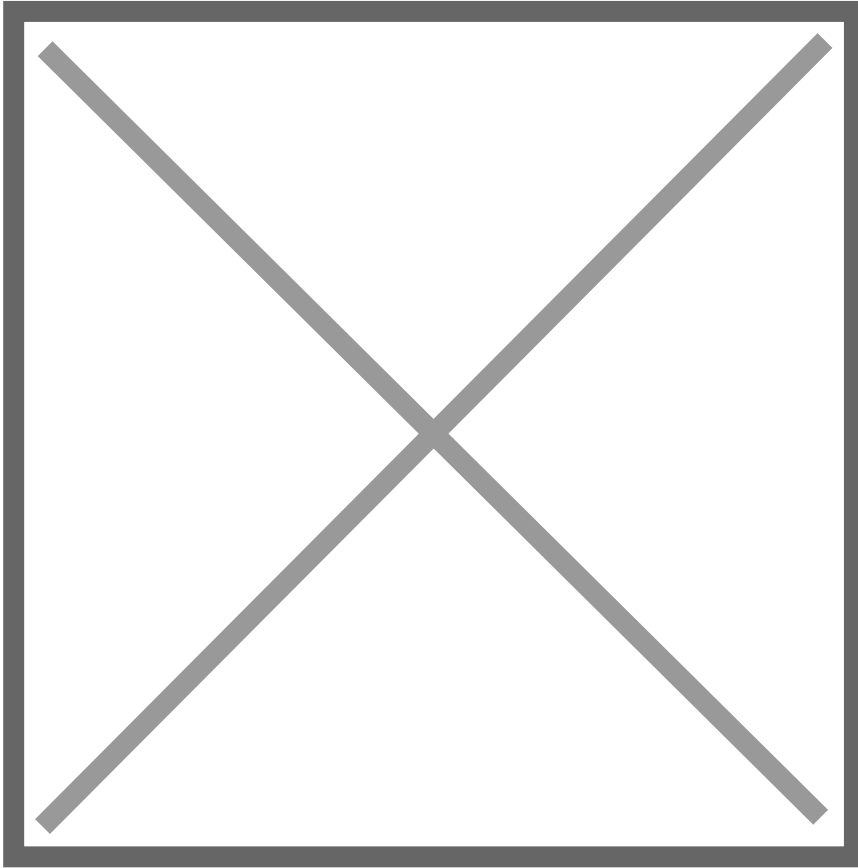
Registrándonos como usuarios de la plataforma **Steamakerblocks** podemos aprovechar todas estas posibilidades:

- Guardar tus proyectos en la nube de **Steamakerblocks** .
- Añadir información al proyecto: descripción, componentes utilizados, imágenes, etc.
- Añadir archivos adjuntos relacionados con el proyecto: esquemas, fotos, archivos para impresión 3D, aplicaciones, etc.
- Compartir proyectos con el resto del mundo.
- Importar proyectos compartidos por otros usuarios.
- Valorar y comentar proyectos.
- Programar directamente Arduino desde el propio navegador (con la aplicación: **Steamakerblocks -Connector**).
- Utilizar la consola serie desde el propio navegador.

Entramos en <https://www.steamakersblocks.com/> e iniciamos sesión



Y rellenamos el formulario



Autor José Andrés Echevarría @cantabRobots CC-BY-NC-SA

STEAMAKERSBLOCKS

Cuentas alumnos

Tal y como dice el tutorial de Juanjo López : *Permite a un usuario registrado con email, crear y administrar nuevas cuentas de usuario dentro de una organización, centro educativo o institución.*

- ♥Permite crear usuarios alumnos sin necesidad de ceder ningún tipo de datos
- ☐☐Puedes pasar proyectos a los alumnos, vacíos o empezados
- ☐☐Tú puedes ver los proyectos de los alumnos y ponerles comentarios

<https://www.youtube.com/embed/zdzKX0NX60Y>

- Si lo quieres en papel, te recomendamos el tutorial de Juanjo López son 12 diapositivas muy bien explicados
 - [Repositorio Github](#)
 - [Repositorio en steamakerbloks.com](#)
- Prueba con Catedu:
 - <https://www.steamakersblocks.com/>
 - usuario alumnox.catedu donde x de 1 a 20
 - contraseña [donde esta catedu en minúsculas]

STEAMAKERSBLOCKS

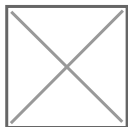
Connector

Espera !!! Aún no conectes tu placa (ESP32, Arduino, TDR STEAM...)

PRIMER PASO Descargar e instalar Connector

Para poder usar la herramienta **Steamakerblocks** tenemos que ejecutar antes **Connector**. Lo descargamos de la misma página de **Steamakerblocks** según el sistema operativo que usemos: Windows (**W7 E INFERIORES NO FUNCIONA**), Linux

Entra en <https://www.steamakersblocks.com/> y en Recursos, tienes la web para descargar este programa:



Lo descargamos y lo **instalamos**.

En el caso de tener equipos Vitalinux, es fácilmente accesible e instalable desde la aplicación **Vitalinux Play** o si se desea una instalación masiva en el centro a través de su página de [soporte](#):

[image-1667329017653.png](#)

SEGUNDO PASO: INSTALAR LOS DRIVERS

Si no hacemos estos pasos, cuando conectamos la placa, siempre sale en el COM1, le damos a subir y sale error

En <http://www.arduinoblocks.com/web/site/abconnector5> tenemos abajo ARDUINO SERIAL DRIVERS

RECOMENDAMOS EL PRIMER ENLACE Y EL TERCERO

2024-06-26 11_27_26-Coding with blocks ;).png

En el primero el instalador está en este enlace

https://cdn.sparkfun.com/assets/learn_tutorials/7/4/CDM21228_Setup.exe

2024-06-26 11_28_55-How to Install FTDI Drivers - SparkFun Learn.png

El segundo sólo si quieres utilizar Arduinos no oficiales, de fabricantes chinos, que tiene el CH340g y hay que leerse la página, paciencia

El tercero es necesario el 2102 si utilizas el ESP32 el instalador esta en este enlace, es una carpeta comprimida, la descomprimes y está el ejecutable instalador

https://www.silabs.com/documents/public/software/CP210x_Windows_Drivers.zip

2024-06-26 11_38_15-CP210x USB to UART Bridge VCP Drivers - Silicon Labs.png

concretamente hay que ejecutar este (al menos que el equipo sea muy viejo de 32bits)

2026-01-19 08_54_16-CP210x_Windows_Drivers - Explorador de archivos.png

COMPROBAR QUE DETECTA LA PLACA

Ahora **conectamos la placa** (ESP32, Arduino, NodeMCU, KeyStudio TDR STEAM...) a nuestro ordenador, y observamos si lo detecta, en Windows entramos en Administrador de dispositivos:

[administradordispositivos.png](#)

Y vemos que en los puertos COM se ha detectado correctamente la placa:

[puertos.png](#)

En el caso de que no aparezca, es que no se han instalado correctamente los *drivers* de Arduino. Entonces vamos a la página oficial de Arduino y descargamos el programa **ARDUINO IDE** : <https://www.arduino.cc/en/software> y lo instalamos. Al instalar este programa se instalan los drivers en nuestro ordenador. No hace falta ejecutarlo.

En el caso de equipos con sistema operativo Linux (como Vitalinux) el puerto serie tiene la forma **/dev/XXXX**

YA PUEDES EJECUTAR ARDUINOBLOCKS CONNECTOR



Ahora buscamos el programa ArduinoBlocks connector que hemos descargado e instalado en el primer paso y lo **ejecutamos**.

[2024-06-26 12_36_52-AB-Connector v5.3.png](#)

ATENCIÓN No podemos cerrar la ventana mientras utilizamos *Arduinoblocks*, la minimizamos simplemente.

En caso contrario, Arduinoblocks no se puede comunicar con nuestra placa Arduino, NodeMCU, KeyStudio, etc

YA PUEDES EJECUTAR ARDUINOBLOCKS

Entramos en la web ARDUINOBLOCKS <http://www.arduinoblocks.com/> nos logueamos e iniciamos un proyecto, Vemos que en el editor que aparece ya los puertos COM (si no te aparece, dale a la rueda actualizar)

Aparecen varios COM, **elegir el último que tiene que coincidir con el que has visto en el segundo paso**, no necesariamente es el COM más alto.

Si se queda una ruleta de espera demasiado tiempo, entonces, actualizar la página o darle a actualizar el botón 1 de la figura :

[arduinoblocks-com.jpg](#)

Una vez elegido el COM ya puedes darle al botón amarillo **SUBIR** cuando has realizado tu proyecto pero antes de subir, por si acaso dale a **guardar** el proyecto que has realizado.

En el caso de equipos con Linux veremos algo así:

[image-1667328938836.png](#)

¿Tengo que hacer los cuatro pasos cada vez?

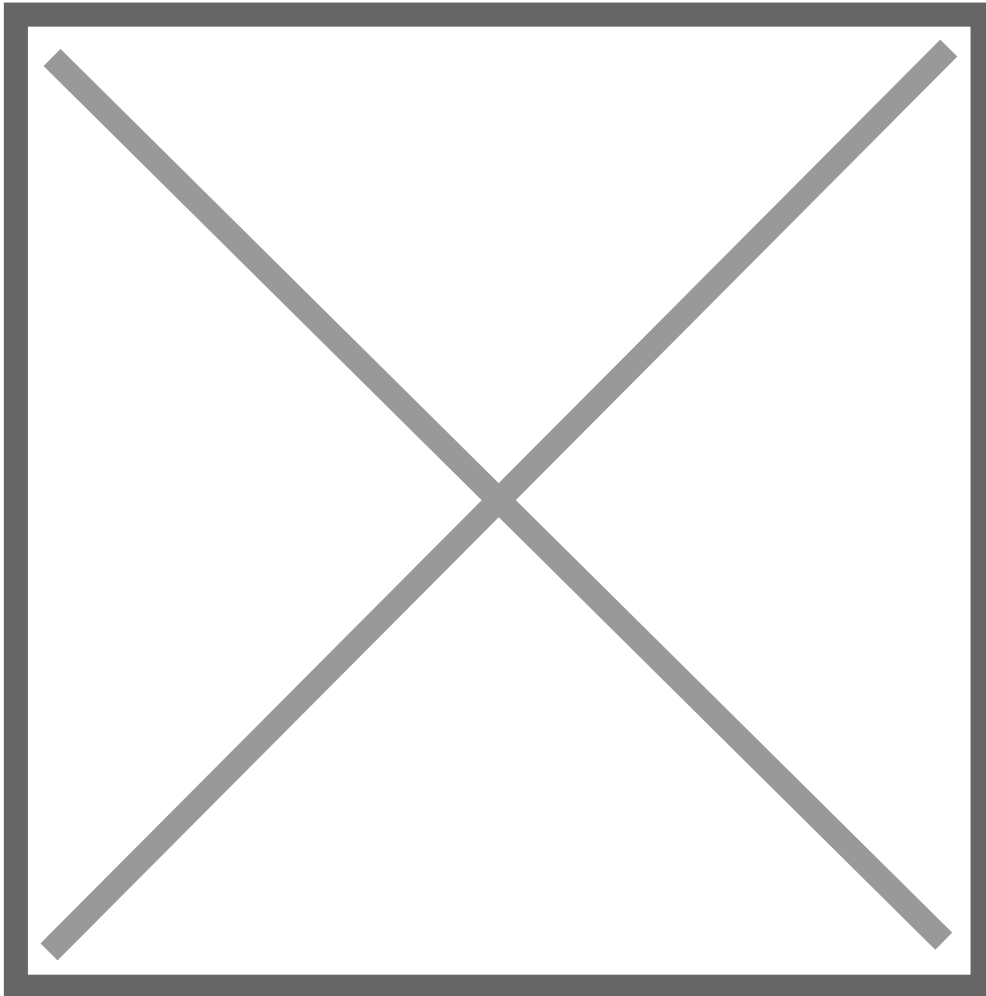
No, sólo la primera vez para asegurar los drivers del Arduino, las siguientes veces que te conectes lo único que tienes que hacer es el tercer y cuarto paso

IMPORTANTE: TENER EL SOFTWARE ARDUINOBLOCKS ACTUALIZADO para que funcionen los nuevos bloques que se incorporan en Arudinoblocks

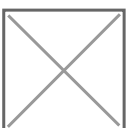
STEAMAKERSBLOCKS

Empezando un proyecto

Entramos en Proyectos y podemos ver nuestros proyectos creados como también empezar uno.



Y nos aparece tres opciones :



En esta ventana podremos elegir que tipo de proyecto vamos a realizar:

- **Proyecto Personal:** Iniciar un nuevo proyecto que sólo será accesible para el usuario. Posteriormente se puede compartir al resto de la comunidad si se desea.
- **Proyecto Profesor:** Iniciar un proyecto como profesor. De esta forma no se inicia un proyecto como tal, sino que se especifican los datos del proyecto y se genera un código para que los alumnos se puedan suscribir al proyecto. El profesor podrá supervisar y valorar los proyectos de sus alumnos.
- **Alumno:** De esta forma nos unimos a un proyecto planteado por el profesor. Nosotros realizaremos el proyecto como si de un proyecto personal se tratara, pero el profesor podrá supervisar y valorar nuestro trabajo.

Adaptado de [este enlace](#). José Andrés Echevarría @cantabRobots CC-BY-NC-SA

Lo primero que tenemos que elegir es para qué tipo de placa se hace el proyecto.

- En el caso de que estés con el kit de CATEDU **Rover marciano con Arduinoblocks** el tipo de proyecto es para **ESP8266 / NodeMCU**
- En el caso de que estés con el kit de CATEDU **Arduino con Arduinoblocks** el tipo de proyecto es para **Arduino UNO**
- En el caso de que estés con el kit de CATEDU **STEAMAKERSBLOCKS en el aula** tienes dos opciones totalmente válidas:
 - **ArduinoUno**
 - **ArduinoUno + Imagina TdR STEAM**
- En el caso de que estés con el kit de CATEDU **ESP32 en el aula** tienes dos opciones totalmente válidas:
 - **ESP32 STEAMakers**
 - **ESP32 STEAMakers + Imagina TdR STEAM**
- En el caso de que estés con el kit de CATEDU **SMARTHOME ESP32** tienes que elegir:
 - **ESP32 STEAMakers**
-

[placa-a-elegir-arduinoblocks.png](#)

ATENCIÓN luego **NO** se puede cambiar. Es decir, un proyecto realizado para un tipo de placa, no se puede cambiar a otro tipo de placa (la razón es simple: las instrucciones cambian)

Luego el nombre y el resto de campos es optativo pero **importante y buena costumbre** rellenarlos, sobre todo si el proyecto lo **compartimos**:

- Descripción



- Componentes
- Comentarios

Área de programación del proyecto

Este es el área sobre el que se trabaja en Arduinoblocks. En esta área arrastraremos y colocaremos los bloques que vamos a utilizar para crear nuestro programa.

En el área de trabajo hay un Zoom (2) para ampliar o reducir la imagen, un icono para centrar (1) y un icono donde podremos borrar los bloques que no utilicemos (3).

2022-04-13 19_02_05-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png

Adaptado de [este enlace](#). José Andrés Echevarría @cantabRobots CC-BY-NC-SA

Las principales secciones del área de programación son las siguientes :

2022-04-13 19_03_07-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png

Adaptado de [este enlace](#). José Andrés Echevarría @cantabRobots CC-BY-NC-SA

Ver el código

ArduinoBlocks genera el código de Arduino a partir de los bloques. El programa se puede compilar y subir directamente a la placa Arduino gracias a la aplicación **ArduinoBlocks-Connector**, sin embargo si deseamos ver o descargar el código podemos realizarlo desde el área de bloques.

2022-04-13 19_05_37-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png

Adaptado de [este enlace](#). José Andrés Echevarría @cantabRobots CC-BY-NC-SA

Siempre, desde un **lenguaje de programación en bloques** podemos obtener su equivalente a **Código de Arduino IDE** (de hecho es lo que hacen los programas), y luego con las funciones de **Código de Arduino IDE** el software lo pasa a **lenguaje máquina** que es la que se graba el Arduino, **pero no al revés** es decir, no existen programas que dado un código máquina o código Arduino IDE lo pasen a bloques gráficos, (igual que no hay programas que lean el código máquina que hay grabado en un Arduino y lo pasen a código



Arduino IDE). Esto no es del todo 100% verdadero pues la Ingeniería inversa en informática trata pues de eso: obtener la fuente aunque sea parcial, pues si obtienes el código legible, puedes alterar lo que quieras.

Cuando compras un programa comercial, te dan el lenguaje máquina ilegible. Mientras que los programas de software libre se publica el código fuente legible para que todo el mundo pueda mejorarlo.

Por ejemplo en la siguiente figura, el programa gráfico **mBlock** que se utiliza en Arduino, mBot, etc... pasa sus instrucciones de lenguaje de programación de bloques parecido a Scratch a lenguaje de **Código de Arduino IDE** y Arduino IDE graba instrucciones binarias de **lenguaje máquina** al Arduino.

[embedded-image-j3gKXHlj.png](#)

¡¡A disfrutar!!

Consejo: Te recomendamos visitar el canal de Youtube de Arduinoblocks

<https://www.youtube.com/c/ArduinoBlocks>

STEAMAKERSBLOCKS

Mi primer programa

Vamos a borrar el programa que tiene predefinido, y simplemente vamos a apagar el dichoso ventilador y visualizar un texto por el Display para tener la seguridad de que tenemos el control

Abrimos Steamakerblocks un nuevo proyecto y elegimos la placa ESP32 STEAMAKER

[2025-10-22 09_58_45-Greenshot.png](#)

Escribimos este programa simple, que apaga al inicio el ventilador y en bucle representa la palabra CATEDU (también se podría poner en el bloque inicializar)

<https://www.steamakersblocks.com/web/project/2519579>

[2025-10-22 09_59_58-Configuración.png](#)

Y ya tenemos todo bajo control:

[2025-10-21 23_52_30-WhatsApp.png](#)

PYTHON

PYTHON

Descarga del programa

Existen muchos compiladores Python de micropython, recomendamos **Thomy**

Podemos descargarnos de la página oficial de Thomy <https://thonny.org/> o de la [página de keystudio](#)

2025-10-23 12_31_47-Descargas - Explorador de archivos.png

¿Qué es eso de "micro" python?

Como puedes ver [en este vídeo en 21:20](#) Python se compila **dentro del microcontrolador** es decir, dentro del ESP32. A diferencia con otros lenguajes, como el C++, el ordenador tiene el compilador, y se lo da ya en binario.

2024-07-04 18_44_27-(1) Exploring the Arduino Nano ESP32 _ MicroPython & IoT Cloud - YouTube.pr

Fuente [vídeo Exploring the Arduino Nano ESP32 | MicroPython & IoT](#)

¿Y a mi qué más me da?

Pues sí que importa....

Si programas ESP32 con Arduino IDE o Arduino Cloud o con Steamakersblock (que está basado en C++) **te has cargado el compilador Python que has puesto en "Actualizar firmware"** del ESP32 luego si quieres programar en Python, tienes que volver a **"Actualizar firmware"**

O sea, si pasas de [ArduilDE] o [Arduino Cloud] o [Steamakersblok] a Micropytho **tienes que volver a instalar el compilador Micropython**

¿Y con esto ya puedo crear mis programas con Micropython?

No, con esto tienes el compilador interpretador dentro del chip, pero necesitas un editor en tu PC y que se comunique con el Micropython del chip

PYTHON

Instalar el firmware

Dentro de Thomy nos vamos a

1. Herramientas
2. Opciones
3. Interprete
4. Instala o actualiza Micropython (ver apartado más abajo)
5. Interprete: ESP32
6. Puerto (el que sea COMx)

[python-home.png](#)

Instalar el firmware

Descargamos de la web https://micropython.org/download/ESP32_GENERIC/ el firmware más actual

[2025-10-23 12_28_29-Configuración.png](#)

Y ahora al pinchar en el punto 4 anterior nos sale este diálogo

[python-home2.png](#)

1. Pulsamos en las tres rayas horizontales
2. Seleccionamos la imagen Micropython que hemos descargado de https://micropython.org/download/ESP32_GENERIC/
3. Automáticamente se rellenan los campos
4. Instalar

La instalación dura unos 5 minutos pues **tiene que instalarse dentro del ESP32**

[2025-10-23 12_10_39-Opciones de Thonny.png](#)

PYTHON

Mi primer programa

Abrimos Thomy nuevo archivo y pegamos un código por ejemplo este

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

Al darle a GUARDAR hay que guardarlo en el dispositivo ESP32 pues recuerda que es un Micropython y el compilador esta dentro de la placa.

[python-home3.png](#)

Al darle al botón play pues suena ...

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

PYTHON

Instalando librería LCD en Micropython

Si vas a utilizar el display (cosa muy recomendable pues da mucho juego) tienes que instalar la librería que se descarga en

https://docs.keyestudio.com/projects/KS5009/en/latest/_downloads/2437db5a73e56669b3562ff78104e2f5/Resource-compression-package.7z

concretamente estas dos

[2025-10-23 12_42_28-Configuración.png](#)

Abre "Thonny", clic en "Este equipo" → "D:" → "2. Proyectos de Python" → "pj8_1_lcd1602".
Seleccione "i2c_lcd.py" y clic derecho para seleccionar "Subir a"

[2025-10-23 12_44_00-Configuración.png](#)

Te tiene que quedar las librerías puestas en la memoria del ESP32

[2025-10-23 12_44_29-Configuración.png](#)

PYTHON

Librería MFRC

Si vas a utilizar el MFRC tienes que instalar la librería que se descarga en

https://docs.keyestudio.com/projects/KS5009/en/latest/_downloads/2437db5a73e56669b3562ff78104e2f5/Resource-compression-package.7z

Abre "Thonny", clic en "Este equipo" → "D:" → "2. Proyectos de Python" → "pj8_1_lcd1602".
Seleccione "i2c_lcd.py" y clic derecho para seleccionar "Subir a"

[2025-10-23 12_44_00-Configuración.png](#)

Te tiene que quedar las librerías puestas en la memoria del ESP32

[2025-10-23 13_46_41-Mattermost Desktop App.png](#)

PYTHON

Introducción al Python

Esta es **una muy breve introducción al Python** como recordatorio de algunas instrucciones si ya has utilizado este lenguaje.

Si es la primera vez, te recomendamos que visites nuestro curso **PYTHON PARA TODOS**

Python for everybody por Charles R. Severance licencia CC-BY-NCSA que empieza desde cero.

Lenguajes, intérpretes y compiladores

Python es un lenguaje **de alto nivel** destinado a ser relativamente sencillo para que los humanos lean y escriban y para que los ordenadores lean y procesen. Otros lenguajes de alto nivel incluyen Java, C ++, PHP, Ruby, Basic, Perl, JavaScript y muchos más. El hardware real dentro de la Unidad Central de Procesamiento (CPU) no comprende ninguno de estos lenguajes de alto nivel.

La CPU entiende un idioma que llamamos **lenguaje de máquina**. El lenguaje de máquina es muy simple y francamente muy tedioso de escribir porque está representado en ceros y unos:

El lenguaje de máquina parece bastante simple en la superficie, dado que solo hay ceros y unos, pero su sintaxis es aún más compleja y mucho más compleja que Python. Muy pocos programadores escriben lenguaje de máquina. En su lugar, creamos varios traductores para permitir que los programadores escriban en lenguajes de alto nivel como Python o JavaScript y estos traductores convierten los programas al lenguaje de máquina para su ejecución real por parte de la CPU.

Estos traductores de lenguaje de programación se dividen en dos categorías generales: (1) intérpretes y (2) compiladores.

Un **intérprete** lee el código fuente del programa como está escrito por el programador, analiza el código fuente e interpreta las instrucciones sobre la marcha. Python es un intérprete y cuando ejecutamos Python de forma interactiva, podemos escribir una línea de Python (una oración) y Python la procesa de inmediato y está lista para que escribamos otra línea de Python.

```
>>> x = 6
>>> print(x)
6
```



```
>>> y = x * 7
>>> print(y)
42
>>>
```

Está en la naturaleza de un **intérprete** poder tener una conversación interactiva como se muestra arriba. A un **compilador** debemos entregarle todo el programa en un archivo, y luego ejecuta un proceso para traducir el código fuente de alto nivel al lenguaje de máquina y luego el compilador coloca el lenguaje de máquina resultante en un archivo para su posterior ejecución.

Variables

Las variables son como cajas que puedes meter valores. Y los valores pueden ser de varios **tipos** :

- **int** si son enteros
- **float** si tienen decimales
- **binario** Deben comenzar por 0b. Por ejemplo: 0b110, 0b11
- **string** son frases, son "**cadena**s" de caracteres entre "
- **bool** Solamente hay dos literales booleanos True o False
- **lista** Se pueden declarar variables que son conjuntos por ejemplo Colores = ["verde", "rojo", "naranja"]

Para crear una variable puedes usar cualquier palabra, x, y, z o Nombre_alumno ... pero algunas palabras no puedes usar, [ver](#)

Para visualizar variables puedes usar la **instrucción print** poniendo entre paréntesis el valor o variable que quieres visualizar.

En la siguiente ventana puedes dar al botón *play* y ver el resultado

<https://trinket.io/embed/python/47afa56dd668>

Modifica los valores como quieras, es un **intérprete**, juega y dale al play para ver el resultado

Como puedes ver se ha introducido un operador el + que realiza la suma del valor de x original (43) y se le incrementa una unidad resultando en la impresión un 44.

Cadenas

Cadenas son secuencias de caracteres, por ejemplo la palabra "banana"

[2025-04-05 09_29_39-Editing Page 6 Cadenas _ Librería CATEDU.png](#)

[fuente 'Python for Everybody' por Charles R. Severance](#)

Se puede obtener su longitud con la función len, o obtener un carácter ...

<https://trinket.io/embed/python/5e023b136db8>

Operadores

Este apartado de operadores es adaptado de Federico Coca [Guía de Trabajo de Microbit CC-BY-SA](#)

Los **operadores aritméticos** se utilizan para realizar operaciones matemáticas como sumas, restas, multiplicaciones, etc.

Operador	Descripción	Ejemplo
+	Suma o concatenación en textos	<code>5+3=8</code> , <code>"Hola" + "Mundo" = "Hola Mundo"</code>
-	Diferencia	<code>6-3=3</code>
*	Multiplicación	<code>3*3=9</code>
/	División	<code>6/2=3</code>
//	Parte entera de un cociente	<code>10//3=3</code>
%	Resto de un cociente	<code>10%3=1</code>
**	Potenciación	<code>5**2=25</code>

Los **operadores de asignación** se utilizan para asignar valores a variables.

Operador	Descripción	Ejemplo
=	Asignación	<code>x=4</code> , <code>a = a + 1</code>
+=	Suma y asignación	<code>x+=1</code> equivale a <code>x = x + 1</code>
-=	Diferencia y asignación	<code>x-=1</code> equivale a <code>x = x - 1</code>
=	Multiplicación y asignación	<code>x=3</code> equivale a <code>x = x * 3</code>
/=	División y asignación	<code>x/=3</code> equivale a <code>x = x / 3</code>

Operador	Descripción	Ejemplo
<code>%=</code>	Asignación de restos	<code>x%=3</code> equivale a <code>x = x % 3</code>
<code>**=</code>	Asignación de exponentes	<code>x**=3</code> equivale a <code>x = x ** 3</code>

Los **operadores de comparación** comparan dos valores/variables y devuelven un resultado booleano: Verdadero o Falso `True` o `False`.

Operador	Descripción	Ejemplo
<code>==</code>	Igual a	<code>2==3</code> retorna <code>False</code>
<code>!=</code>	Distinto de	<code>2!=3</code> retorna <code>True</code>
<code><</code>	Menor que	<code>2<3</code> retorna <code>True</code>
<code>></code>	Mayor que	<code>2>3</code> retorna <code>False</code>
<code><=</code>	Menor o igual que	<code>2<=3</code> retorna <code>True</code>
<code>>=</code>	Mayor o igual que	<code>2>=3</code> retorna <code>False</code>

Los **operadores lógicos** se utilizan para comprobar si una expresión es Verdadera o Falsa. Se utilizan en la toma de decisiones.

Operador	Descripción	Ejemplo
<code>and</code>	AND lógica	<code>a and b #True</code> si a y b son ciertos
<code>or</code>	OR lógica	<code>a or b #True</code> si a o b son ciertos
<code>not</code>	NOT lógica	<code>not a #True</code> si el operador a es falso
<code>in</code>	pertenencia	Devuelve True si pertenece
<code>no in</code>	no pertenencia	Devuelve True si no pertenece
<code>is</code>	identidad	Devuelve True si son iguales
<code>is not</code>	no identidad	Devuelve True si no son iguales

Los **operadores bit a bit** o bitwise actúan sobre los operandos como si fueran cadenas de dígitos binarios. Operan bit a bit:

Operador	Descripción	Ejemplo
<code>&</code>	AND bit a bit	<code>5&6 # 101 & 110 = 110 = 4</code>

Operador	Descripción	Ejemplo
	OR bit a bit	<code>5 6 # 101 110 = 111 = 7</code>
~	NOT bit a bit	<code>~3 # ~011 = 100 = -4</code>
^	XOR bit a bit	<code>5^3 # 101^011 = 110 = 6</code>
<<	Desplazamiento izquierda	<code>4<<1 # 100 << 1 = 1000 = 8</code>
>>	Desplazamiento derecha	<code>4 >> 1 # 100 >> 1 = 010 = 2</code>

Prueba, juega con este código:

<https://trinket.io/embed/python/502063b6b44b>

Comentarios en Python

Una sola línea : Escribiendo el símbolo almohadilla (#) delante del comentario.

Multilínea: Escribiendo triple comillas dobles (""") al principio y al final del comentario.

Entradas de teclado

Ya hemos visto salidas por pantalla con **print**, pero ahora con input puede leer variables del teclado, esto es mejor experimentarlo que leerlo :

<https://trinket.io/embed/python/34653253eb52>

Fíjate que hay que poner las líneas **x = float(x)** e **y = float(y)** para convertirlos a números decimales, en caso contrario las interpreta string y no puede multiplicar en Resultado, pero en el siguiente ejemplo **no es necesario en la variable cel** (celsius) pues se multiplica por números decimales 32.0 5.0 y 9.0

<https://trinket.io/embed/python3/5dbec1550b>

try y **except** son dos funciones que son *un seguro para el programador* por si el usuario en vez de teclear un número, mete un string o carácter

La sangría es importante en Python

La sangría se refiere a los espacios al comienzo de una línea de código. Mientras que en otros lenguajes de programación la sangría en el código es solo para facilitar la lectura, la sangría en Python es muy importante ya que se usa para indicar un bloque de código.

Condicionales

Las instrucciones **if: else:** son las que nos permiten realizar operaciones según las condiciones puestas. *Ojo con la sangría*

<https://trinket.io/embed/python/cc1aa3f917a7>

`\n` es un carácter especial que significa "Salto de página"

Bucles

- **while** ejecuta lo contenido en la sangría mientras sea verdadero la condición
- **for** ejecuta lo contenido en la sangría mientras y va recorriendo la variable dentro del rango creado

Para verlo mejor vamos a ver estos ejemplos

- EJEMPLO BUCLE WHILE
 - mientras n sea positivo va ejecutando : imprime n y lo decrementa
 - al decrementar llega un momento que deja de ser positivo y finaliza el bucle
- EJEMPLO BUCLE WHILE INFINITO
 - Es muy típico en robótica, todo el rato hace el bucle (en robótica para que lea los sensores y realice cosas en los actuadores) pero este ejemplo no esta en un robot sino en tu pc y no queremos que se quede "colgado" luego al teclear "fin" acaba gracias a la instrucción **break**
 - Fíjate que hay una instrucción **continue** para que pase a la siguiente iteración provocando que no imprime lo teclado



- EJEMPLO BUCLE FOR FRIENDS
 - Va recorriendo la variable friend dentro del conjunto lista friends
 - como puedes ver la diferencia entre for y while es que for además recorre la variable
- EJEMPLO BUCLE FOR
 - mientras n este en el rango de 0 a 5 se ejecuta

Venga pruébalo !!!

<https://trinket.io/embed/python/f797a1eaea48>

Funciones

No vamos a entrar en detalle, pero observa el siguiente código

- **FUNCIONES PREDEFINIDAS** Si observas, la primera línea llama a importar una librería externa, **import math** donde math es un fichero que tienen funciones predefinidas, vamos a utilizar una de ellas, la raíz cuadrada **sqrt** luego para llamar a esa función que esta definida dentro de math se hace con la instrucción **math.sqrt**
- **FUNCIONES DEFINIDAS POR TI** em este caso, se utiliza la palabra **def** para crear una función, que le vamos a pasar tres argumentos a, b y c y para finalizar la función usamos **return** para devolver el valor que queremos obtener

<https://trinket.io/embed/python/900fd133a2a9>

Para saber más de Python

CURSO PYTHON FOR EVERYBODY en español	ver
Curso completo de Python 222pag pdf (*)	Descargar
Curso completo de Python 422pag (*)	Descargar
Curso completo de Python desde 0 (*)	Ver
Curso de Python desde 0 (*)	Ver
Manual de referencia Python (*)	Ver
Programación en Python (*)	Ver



Trabajando con ficheros en Python (*)	Ver
Programación orientada a objeto en Python (*)	Ver
un manual para aquellos usuarios con previos conocimientos de Python, como la programación modular y orientada a objetos. También algunos conocimientos de las librerías tkinter (Para crear interfaces gráficos y SQLite3 (para gestionar bases de datos). (*)	Descargar

(*) Agradecimientos a Pere Manel <http://peremanelv.com>

RETOS no IoT

RETOS no IoT

Project 1.1 LED Flashing

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

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

SOLUCION CON CODIGO

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

RETOS no IoT

Project 1.2 Breathing LED

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

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

- Solución con bloques <https://www.steamakersblocks.com/web/project/2520329>
- Solución con código
<https://docs.keyestudio.com/projects/KS5009/en/latest/docs/Python/Python.html#project-1-2-breathing-led>

¿Qué son las salidas PWM?

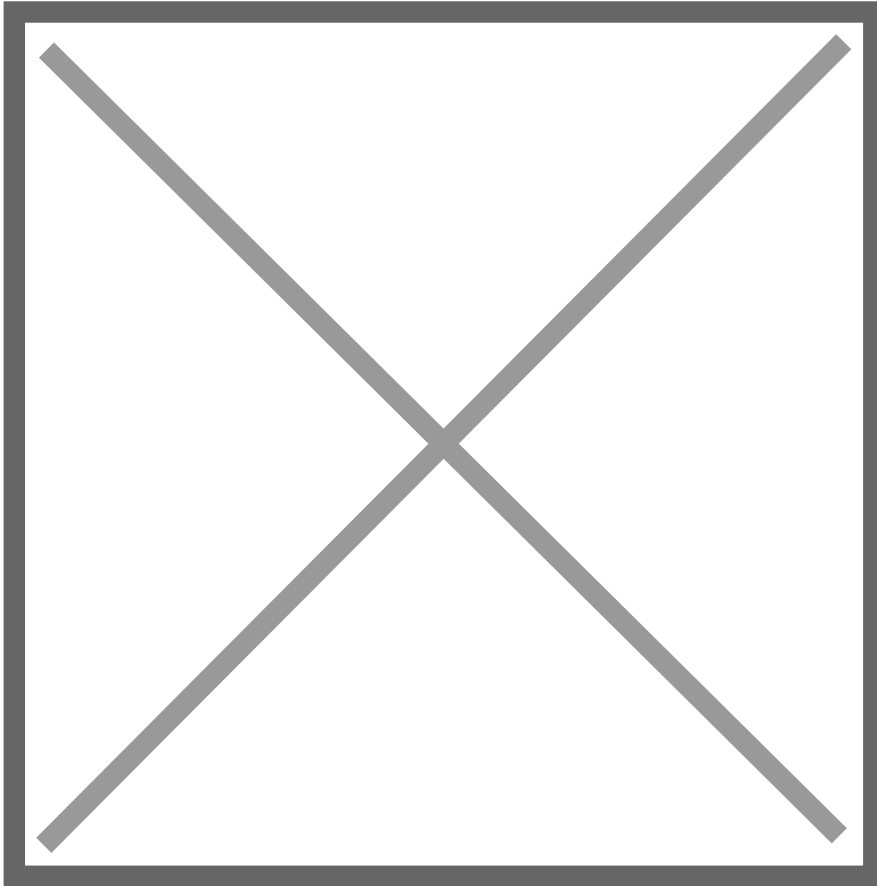
Arduino, ESP32, Micro:bit, PicoW... tienen **entradas** analógicas y digitales. Pero **salidas sólo digitales**.

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

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

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

Veamos gráficamente la señal PWM:



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

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

Ejemplo en código ArduinoIDE y Arduino

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

Otro ejemplo en Python con Micro:bit

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

RETOS no IoT

Project 2.1 Read the Button

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

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

- SOLUCIÓN CON CÓDIGO

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

RETOS no IoT

Project 2.2. Table Lamp

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

- SOLUCIÓN CON BLOQUES <https://www.steamakersblocks.com/web/project/2520536>
- SOLUCIÓN CON CÓDIGO
<https://docs.keyestudio.com/projects/KS5009/en/latest/docs/Python/Python.html#project-2-2-table-lamp>

RETOS no IoT

Project 3.1 Read the PIR Motion Sensor

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

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

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

RETOS no IoT

Project 4.1 Play Happy Birthday

<https://www.youtube.com/embed/5c0SQh66Jk>

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

- SOLUCIÓN CON BLOQUES <https://www.steamakersblocks.com/web/project/2521317>
- SOLUCIÓN CON CÓDIGO
<https://docs.keyestudio.com/projects/KS5009/en/latest/docs/Python/Python.html#project-4-1-play-happy-birthday>

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

[2025-10-22 19_22_02-Mattermost Desktop App.png](#)

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

RETOS no IoT

Project 5.1 Control the Door

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

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

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

RETOS no IoT

Project 5.2 Close the Window

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

Los ángulos correctos para el servo son

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

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

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

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

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

2025-10-22 21_09_24-WhatsApp.png

RETOS no IoT

Project 6.1 RGB Sk6812

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

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

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

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

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

RETOS no IoT

Project 7.2 Button Control Fan

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

- SOLUCIÓN POR BLOQUES <https://www.steamakersblocks.com/web/project/2521510>
- SOLUCIÓN CON CÓDIGO
<https://docs.keyestudio.com/projects/KS5009/en/latest/docs/Python/Python.html#project-7-2-button-control-fan>

RETOS no IoT

Project 8.2 Dangerous Gas Alarm

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

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

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

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

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

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

ATENCIÓN: EN STEAMAKERSBLOCK no se encuentra este sensor de gas, pero como es un sensor de todo o nada pull down, es decir digital, que emite un 1 si no detecta y emite un 0 si detecta, se cambia por un sensor táctil, (o de lluvia o de cualquier cosa que haga 1 o 0)
2025-10-23 13_29_13-WhatsApp.png

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

RETOS no IoT

Project 9.1 Temperature and Humidity Tester

2025-10-23 13_25_25-WhatsApp.png

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

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

RETOS no IoT

Project 10.1 Open the Door

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

- SOLUCIÓN CON BLOQUES <https://www.steamakersblocks.com/web/project/2523179>
- SOLUCIÓN CON CÓDIGO
<https://docs.keyestudio.com/projects/KS5009/en/latest/docs/Python/Python.html#project-10-1-open-the-door>

RETOS Bluetooth

RETOS Bluetooth

Un poco de teoría Bluetooth

ONDAS

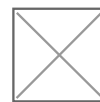
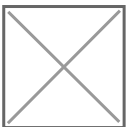
Una onda es una señal que se propaga por un medio. Por ejemplo el sonido, que es una onda mecánica que viaja usando el aire o cualquier otro material. Pero en el caso de las señales eléctricas pueden ser enviadas por el cable o a través del vacío (no necesitan un medio para transmitirse).

Dependen de 3 parámetros principalmente:

- **Amplitud:** altura máxima de la onda. Hablando de sonido representaría el volumen. Si nos referimos a una onda eléctrica estaríamos representando normalmente el voltaje.
- **Longitud de onda λ :** distancia entre el primer y último punto de un ciclo de la onda (que normalmente se repite en el tiempo).
- **Frecuencia f :** Número de veces que la onda repite su ciclo en 1 segundo (se mide en hertzios).
- **Periodo T** es simplemente es la inversa de la frecuencia. $T=1/f$

La relación entre ellas es muy fácil pues las ondas electromagnéticas viajan a la velocidad de la luz c y si velocidad es espacio/tiempo luego $c = \lambda/T$ luego $c = \lambda * f$

Dentro del espectro electromagnético encontramos diferentes tipos de señales dependiendo de las características de su onda.



TRANSMISIÓN INALÁMBRICA: BLUETOOTH.

- Hoy en día, este grupo está formado por miles de empresas y se utiliza no sólo para teléfonos sino para cientos de dispositivos.
- Bluetooth es una red inalámbrica de corto alcance pensada para conectar pares de dispositivos y crear una pequeña red punto a punto, (sólo 2 dispositivos).
- Utiliza una parte del espectro electromagnético llamado "**Banda ISM**", reservado para fines no comerciales de la industria, área científica y medicina. Dentro de esta banda

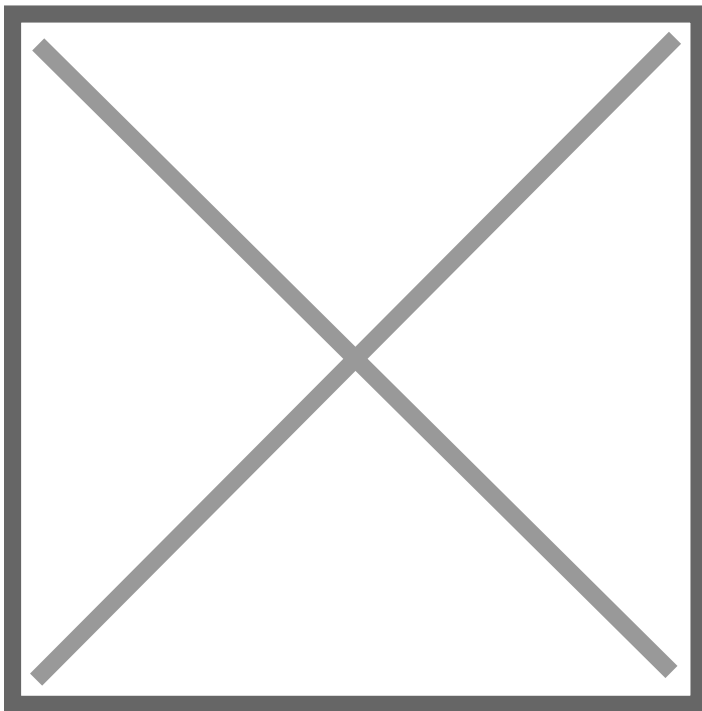


también se encuentran todas las redes WIFI que usamos a diario. En concreto funcionan a 2,4GHz. (Un G son 10^9) luego entre FM y Microondas.

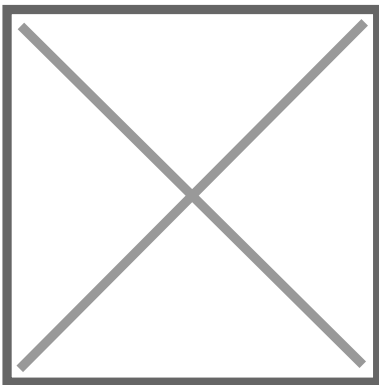
¿Sabías que?

Su curioso nombre viene de un antiguo rey Noruego y Danés, y su símbolo, de las antiguas ruinas que representan ese mismo nombre.

Hay 3 clases de bluetooth que nos indican la máxima potencia a la que emiten y por tanto la distancia máxima que podrán alcanzar:



También es muy importante la velocidad a la que pueden enviarse los datos con este protocolo:



Mbps : Mega Bits por segundo. MBps: Mega Bytes por segundo.
 kb = 1.024 b M = 1.024 k G = 1.024 M

¿Te atreves a calcularlo ?

¿Cuántos ciclos por segundo tendrán las ondas que están en la **Banda ISM**? ¿Cuál es el periodo de esas ondas?

Solución

a) $f = 2.4\text{G}$

b) $\lambda = c/f = 12.5\text{cm}$ o sea, las antenas tendrían que ser de esta longitud. Hay muchos trucos para reducirla, una de ellas es la forma de serpiente que puedes ver en el HC-06

¿Te atreves a calcularlo...?

¿A qué distancia y cuanto tiempo tardarían en enviarse los siguientes archivos por Bluetooth?

1. Un vídeo de 7Mb usando versión 2 clase 2
2. Una imagen de 2.5Mb usando versión 3 clase 1
3. Un archivo de texto de 240KB usando versión 1.2 clase 1

Solución

1) $7\text{Mb} / 3\text{Mbs} = 2.3 \text{ seg.}$

2) $2.5\text{Mb} / 24\text{Mbs} = 0.1 \text{ seg.}$

3) $240 \text{ kB } 8\text{b/B} = 1.920 \text{ kb}$ $1.920 \text{ kb} / 1.024 = 1.875 \text{ Mb}$ $1.875\text{Mb} / 1\text{Mbs} = 1.875 \text{ seg.}$

¿Bluetooth clásico o Bluetooth Low Energy = BLE?

Es un protocolo similar al clásico Bluetooth pero diseñado a consumir menos potencia manteniendo funcionalidad. Su popularidad ha crecido en multitud de dispositivos

En robótica, el clásico device que utiliza BLE es la **Micro:bit**. Aunque la Micro:bit no tiene Wifi integrada, posee una radiofrecuencia que podemos configurar para Bluetooth (hay que elegir, o utilizar sus comandos de Radio o utilizar comandos de Bluetooth)

Por eso a la hora de elegir la APP tienes que tener en cuenta:

- Si acepta Bluetooth clásico o BLE
- Que la APP acepte leer datos desde el robot como enviar



Nosotros hemos elegido uno sencillo que cumple las dos condiciones (hay muchas APPs) [Serial Bluetooth Terminal](#)

2025-12-05 08_05_39-WhatsApp.png

RETOS Bluetooth

Bluetooth en el ESP32

A diferencia del [Arduino que necesita un módulo exterior el HC06](#), el **ESP32 tiene el Bluetooth integrado**

Para inicializar el Bluetooth, STEAMAKERBLOCKS tiene instrucciones específicas, lo primero es definir un nombre a la señal Bluetooth para que los otros dispositivos (tu móvil) lo encuentre, por ejemplo en la figura ESP32 STEAMakers

[2025-10-24 08_37_02-Mattermost Desktop App.png](#)

El bloque **Fijar timeout** significa que fijamos el tiempo en la transmisión de datos por bluetooth. Por defecto está fijado en 1000 ms pero podemos reducirlo a 100 ms como en la figura para que funcione más fluido.

RETOS Bluetooth

Reto de la APP hacia ESP32

Enunciado

Si recibe un código, que encienda el ventilador. Por ejemplo:

- si recibe un 1 que se encienda el ventilador.
- si recibe otro código, apaga el ventilador

Este enunciado parecido a otro del curso [ESP32 en el aula](#)

Programa

El programa es sencillo: <https://www.steamakersblocks.com/web/project/2523984>

[2025-10-24 08_37_02-Mattermost Desktop App.png](#)

[2025-10-24 08_57_50-Mattermost Desktop App.png](#)

La APP Arduino Bluetooth Control

Esta APP es para comunicación PLACA->APP es decir, el Bluetooth será como receptor de datos nuestra placa electrónica (slave) y el móvil (master) como emisor de datos.

Esta APP permite enviar datos pero de distintas formas

[2026-01-19 10_45_00-WhatsApp.png](#)

Arduino Bluetooth Control

Esta APP es muy completa y configurable, [aquí para descargarla de Google Play](#).





El código de programa que tenemos que cargar en la placa se basa en escuchar de forma continua el puerto serie. **Cuando llegue el dato, se ejecutará la acción que le indiquemos.** ¡¡así de sencillo !!

<https://giphy.com/embed/xTilzoyw4Yh3mRM5DG>

Vincular móvil

Hay que vincular nuestro móvil y nuestra APP de Android con el Arduino, para ello sigue [este sencillo tutorial](#):

<https://docs.google.com/presentation/d/e/2PACX-1vT0vG1z61MuZ XKm diw4ga7z15FIQfeussqDNYzMauJSZUU2G2NIL7M-JjXb4PFT4YTigj9Yal8PzHmR/embed?start=false&loop=false&delayms=3000>

El resultado

<https://www.youtube.com/embed/zlHzjxNET24>

RETOS Bluetooth

Reto APP y ESP32 BIDIRECCIONAL

Enunciado

Ahora vamos a realizar la comunicación bidireccional. El propósito es que si recibe un código, que encienda y apague los leds, pero además vamos añadir comunicación hacia la APP del móvil, que nos envíe información de los sensores. Un ejemplo de enunciado sería:

- APP->ES`32
 - si recibe un ON que se encienda el led
 - si recibe un OFF que se apague
- APP<-ESP32:
 - Que envíe a la APP el valor de la temperatura del DHT11.
 - Que envíe a la APP el valor del sensor lluvia.

Además lo vamos a complicar un poco más:

- que visualice los valores de temperatura y lluvia en el display LCD
- Que si lluvia es mayor de 50 que cierre la ventana (en caso contrario que este abierta)
- Que también lo diga el display LCD
- Y también la APP

Este enunciado parecido al programa en [ESP32 en el Aula](#)

Programa

El programa ya es más complejo: <https://www.steamakersblocks.com/web/project/2524455>

2025-10-24 15_28_52-.png

La APP en el móvil

La comunicación con la placa en este caso no es tan sencilla como en el caso anterior, no nos vale las APPs que hay en Google Play que envían un código y en paz, ahora tenemos que tener una APP que si reciben un código que hagan algo. Esto ya es **exclusivo** por lo tanto **hay que programar una APP que cumpla los requisitos. En nuestro caso lo hemos hecho en APP INVENTOR**

<https://appinventor.mit.edu/>

NO ES EL OBJETIVO DE ESTE CURSO FORMARTE EN APP INVENTOR

Pero si no conoces el programa, te recomendamos que vayas a estos contenidos que de una manera muy fácil aprenderás a crear programas con APP INVENTOR y exportarlos al móvil

<https://libros.catedu.es/books/app-inventor>

En la parte de **DESIGNER**

[2025-10-24 15_30_25-WhatsApp.png](#)

En la parte de **Blocks**

[2025-10-24 15_31_28-WhatsApp.png](#)

Puedes descargar la APP e importarlo a APP INVENTOR : [HOMEBLUETOOTHBIDIRECCIONAL.aia](#)

Aquí tienes los pasos de su realización

https://docs.google.com/presentation/d/e/2PACX-1vTRRqw_hTIQhfUlWzW7T_Uo6sTSvZxy5YgLfADLBsM5Wb3J6PUrEU1F-PWef_FW0FaoyRjIC9hCGd8-/pubembed?start=false&loop=false&delayms=3000

La APP a tu móvil

Tienes dos opciones

- **EN VIVO CONNECT - AI COMPANION** esta opción es la más rápida, y realmente lo simula a través de la APP INVENTOR.
 - Tienes que tener instalada la APP MIT AI2 COMPANION
 - Se le pasa el código de tu APP a la APP
- **OTRAS OPCIONES**
 - Ver <https://appinventor.mit.edu/explore/ai2/setup>

OPCIÓN EN VIVO AI COMPANION

Instalas la [APP MIT AI2 COMPANION](#)

[APP-MIT.png](#)

En APP INVENTOR

[conect-ai-companion.png](#)

Y sale un código y un QR asociado al código

[cod-ai-companion.png](#)

Abrimos la [APP MIT AI2 COMPANION](#) y metemos el código anterior (o lo escaneamos con el QR)

[ai-companion2.jpg](#)

En APP INVENTOR verás que sale una barra de progreso enviando tu APP a tu móvil. Cuando termina automáticamente lo ejecuta.

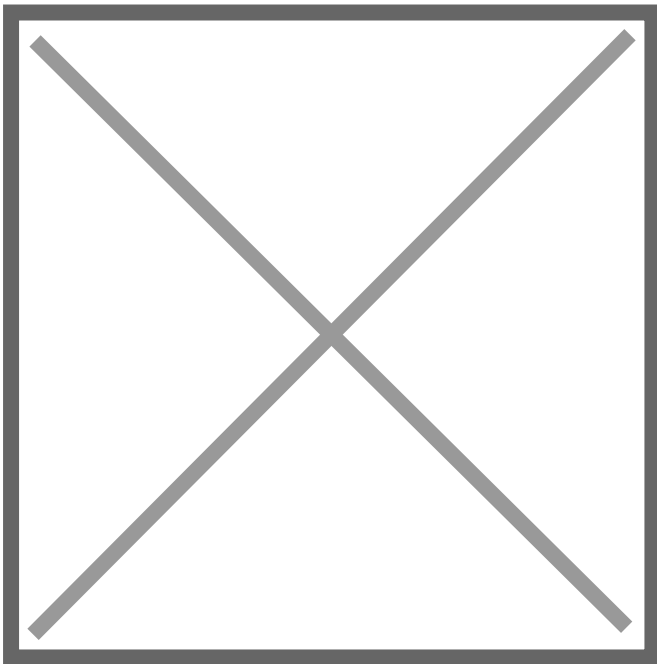
El resultado

<https://www.youtube.com/embed/m64C0wb3jKs>

Qué es IoT

El **Internet de las cosas** (Internet of Thing IoT) describe objetos físicos —o grupos de estos— con sensores, capacidad de procesamiento, software y otras tecnologías que se conectan e intercambian datos con otros dispositivos y sistemas a través de internet u otras redes de comunicación. El Internet de las cosas se ha considerado un término erróneo porque los dispositivos no necesitan estar conectados a la Internet pública. Sólo necesitan estar conectadas a una red y ser direccionables individualmente

[Fuente Wikipedia IoT Internet de las cosas CC-BY-SA](#)



[De Drawed by Wilgengebroid on FlickrTranslated by Prades97 CC BY-SA 3.0](#)

Estamos hablando de dispositivos que se conectan a internet de forma desatendida, por vía hardware (o mejor dicho firmware) a diferencia de un ordenador, tablet o móvil, donde tienes que configurar por software el dispositivo y hay un diálogo entre usuario y dispositivo sobre el uso de Internet (el software solicita tal página web, tales datos etc por voluntad del usuario o por diálogo con el usuario) Aquí los dispositivos están ya configurados de los datos que se comunican. Es decir "conectar y olvidar".

Piensa en la diferencia entre un enchufe inteligente y un ordenador, el primero es lo que se considera dentro de IoT

Desventajas: El acceso a Internet de dispositivos caseros puede generar problemas a nivel mundial:

- [el caso Mirai](#)
- [aspiradores que nos espían](#)

IoT en los cursos de Aularagón

- **Blynk:** lo que nos gusta de esta herramienta es que es casi "instantánea" o "síncrona". Esto es imprescindible con ciertos robots como el **Rover Marciano con Arduino**. Necesitamos que "gire" para evitar un obstáculo, no podemos esperar !!! Veremos con **BLYNK** un protocolo que entre el dispositivo electrónico (nuestro robot) y nosotros (en ordenador, en una APP en el móvil) la comunicación es instantánea, gracias a un servidor que hará de intermedio, que puede ser local (BLYNK LEGACY) o en Internet (BLYNK IoT).
 - **Blynk legacy** es la que se va a trabajar en
 - [Rover Marciano con Arduino](#)
 - [Arduinoblocks en el aula](#)
 - [ESP32 en el aula](#)
 - **Blynk IoT** es la que se va a trabajar con
 - [En ESP32 en el aula](#)
 - [En Smart Home ESP32](#)
- **ThinkSpeak y SmartioSpace**
 - [Smart Agriculture Kit para Micro:bit](#)
- **MQTT** El emisor envía datos, se almacenan en un servidor, y cuando puede, lo vuelca al cliente. Cliente y emisor pueden ser el dispositivo electrónico y nosotros o viceversa. Veremos que esto es lo que hace el protocolo **MQTT** y está tremendamente extendido por lo barato y fácil que es. Hace que los servidores no estén tan ocupados, por lo tanto hay varios proveedores que ofrecen este servicio gratuitamente. Hay robots como los que tienen la placa **TDR STEAM IMAGINA** que envía datos de temperatura, humedad, .. y pueden recibir datos pero no precisan de esta exigencia instantánea como un rover.
 - [ESP32 EN EL AULA](#)
 - [En Smart Home ESP32](#)
- **TELEGRAM**
 - [ESP32 EN EL AULA](#)
 - [En Smart Home ESP32](#)



- **Arduino cloud IoT**
 - [Arduino Alvik](#)
- **Cyberpi y mBot2**
 - [IoT con Cyberpi](#)

RETO IoT Blynk

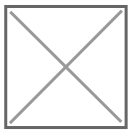
RETO IoT Blynk

¿Qué es Blynk?

Blynk pertenece al mundo del Internet de las cosas IoT que has visto anteriormente. Hay dos opciones en IoT.

- MQTT ocupa pocos datos pero es lento
- Instantaneo: Blynk Aquí utilizaremos Blynk para comunicarnos casi instantáneamente. Con el **Rover con Arduino** esto es imprescindible ¡¡no queremos que se estrelle!! con **TdR STEAM** no es tan imprescindible, pero si queremos interactuar con él (encender sus leds, hacer un pitido...) con MQTT puede ser desesperante.

Blynk nos posibilita la conexión de un hardware conectado a Internet: Arduino, Raspberry, NodeMCU, ESP32 ... con la APP Blynk (hay dos APPs que veremos IoT y Legacy) que puedes poner diferentes elementos de control.



O sea, se necesitan tres cosas:

1. Un dispositivo para poner la **APP** por ejemplo un móvil Smartphone o una tableta, la APP hay dos tipos:
 1. **Blynk IoT**
 2. **Blynk Legacy**
2. El dispositivo o **robot** que esté preparado para conectarse por Internet IoT Hardware
 1. En el caso del Rover con Arduino es el NodeMCU
 2. En el caso de Arduinoblocks en el aula es un Arduino con el ESP01 o ESP8266
 3. En el caso de ESP32 en el aula es un ESP32 que tiene la wifi integrada.
3. Una conexión entre los dos
 1. En el caso de **Blynk IoT** lo proporciona la empresa, con un servidor remoto en Internet
 2. En el caso de **Blynk Legacy** lo proporcionamos nosotros con una Raspberry que hará de servidor local en nuestra red local. (en los cursos de Catedu para que no te entretengas con esto, te proporcionamos un servidor nuestro).



RETO IoT Blynk

Blynk legacy y Blynk IoT


Antes de junio 2022 existía **Blynk legacy** en los servidores de [Blynk Inc](#) pero a partir de esa fecha, ha dejado de funcionar pasando a **Blynk IoT** ([ver noticia](#)) esto crea varios inconvenientes:

VENTAJAS E INCONVENIENTES DE USAR BLYNK LEGACY

- Compatible con todas las placas en Arduinoblocks
- Muchos centros y proyectos utilizan en sus servidores locales **BLYNK LEGACY** por lo que ARDUINOBLOCKS mantendrá en el futuro su compatibilidad con Blynk legacy.
- La APP se instala en en Google Play (pero con riesgo que ya no dejen de ofertarlo) o fuera de Google Play
- El servidor hay dos opciones
 - En tu centro hacer un **SERVIDOR LOCAL** lo más fácil es con una Raspberry, en el curso se enseña cómo hacerlo.
 - En un servidor de CATEDU SÓLO PARA LOS CURSOS DE ROBOTICA
- Al utilizar **BLYNK LEGACY** con tu propio servidor tiene la ventaja que no tiene ninguna limitación de widgets y tienes el control de los alumnos, pero si el servidor es local, no se puede utilizar por ejemplo desde casa.

VENTAJAS E INCONVENIENTES DE USAR BLYNK IoT

- **BLYNK IoT** tiene limitaciones, la mayoría de widgets son de pago pero los esenciales: switch, slider, led, label, gauge son gratuitos.
- Es una buena opción para que los alumnos puedan trabajar en casa con su robot, estamos a la espera que ARDUINOBLOCKS sea compatible con **BLYNK IoT ACTUALMENTE SÓLO ES POSIBLE CON LA PLACA ESP32 STEAMAKER** (OJO hay que tener [Arduinoconnector actualizado](#))

“ Ya puedes probar los bloques para el nuevo Blynk IoT [@blynk_app](#) con la placa [#esp32steamakers](#) de [@Innovadidactic](#) y [@ArduinoBlocks](#)  [#iot](#) [#blynk](#) [#esp32](#) [#steamakers](#) [#steam](#) [#dashboard](#) pic.twitter.com/kzPNkSpbAV

— ArduinoBlocks (@ArduinoBlocks) [November 19, 2022](#)



RETO IoT Blynk

Crear DASHBOARD en Blynk.io

Vamos a crear un DASHBOARD o PANEL DE CONTROL para controlar nuestro ESP32.

AÑADIR DEVICE

1. Vamos a **Devices**
2. **Create New**
[2026-01-19 16_48_26-Greenshot.png](#)
3. Buscamos **ESP32**
4. Elegimos **Quick start**

[2026-01-19 16_49_59-.png](#)

Nos saldrá un código que tiene **tres cosas importantes** que tienes que guardar (En download te lo baja en formato .ino que lo puede leer el bloc de notas)

1. **ID DE LA PLANTILLA**
2. **NOMBRE DE LA P LANTILLA**
3. **TOCKEN**

[2025-10-23 22_07_24-Devices - Blynk.Console — Mozilla Firefox.png](#)

En el siguiente diálogo quiere meter el código en el ESP32, dar a **CANCELAR** pues eso lo hará Steamakersblocks

DATASTREAMS O FLUJO DE DATOS

Una vez creado el **Device**, le damos dos clicks y nos sale el **Template**. Al darle a **EDIT** tenemos opción de ir añadiendo las variables asociadas a los pines virtuales, TE RECOMIENDO BORRAR LAS QUE TE PONE Y AÑADIR TÚ LOS PINES VIRTUALES por ejemplo en este caso DE ESP32 KEYSTUDIO TDR STEAM IMAGINA

NOMBRE	PIN VIRTUAL	TIPO	UNIDAD	MIN	MAX
--------	-------------	------	--------	-----	-----



LED ROJO	V1	entero		0	1
SIMPSON	V2	entero		0	1
LCD	V3	CADENA			
PULSADOR	V4	entero		0	1
TEMPERATURA LM35	V5	entero	°C	0	100
TEMPERATURA DHT11	V6	entero	°C	0	100
HUMEDAD	V7	entero	%	0	100
SONIDO	V8	entero		0	4095

Ves añadiendo las variables que quieras visualizar o gobernar, dándoles las unidades, valores min, max correspondientes

No utilices ni el potenciómetro ni el sensor de luz, tal y como vimos en Sensores y actuadores de la placa Imagina TDR STEAM, cuando hay comunicación Wifi, están inutilizados.

[2026-01-19 21_27_09-Greenshot.png](#)

TEMPLATE PANEL WEB

Una vez creado el **Device**, le damos dos clicks

[2025-10-23 22_10_43-Devices - Blynk.Console — Mozilla Firefox.png](#)

Si le damos dos clicks sale el **Template**

- En los tres puntos horizontales le podemos cambiar el nombre
- En el Edit podemos poner nuestros elementos

[2025-10-23 22_11_48-Devices - Blynk.Console — Mozilla Firefox.png](#)

Vamos a ir añadiendo **WIDGETS ASOCIADOS A ESOS DATASTREAMS** para crear tu panel de control, algunos son de pago ☐☐

[2026-01-19 21_54_28-.png](#)

Los puedes redimensionar, borrar, etc... En el botón de la rueda dentada vas asociando el Widget con el pin virtual V1, V2, ... V8

[blynk-gif.gif](#)

Fuente <https://blynk.io/>

APP BLYNK IOT

Hay una APP sencilla, que al loguearse YA APARECE EL DEVICE lo único que hay que hacer es el TEMPLATE asociado, y es muy sencillo

[2025-10-23 22_27_43-Greenshot.png](#)

Podemos visualizar y crear las plantillas en el móvil, para ello sigue las instrucciones

CASO ESP32 SMART HOME

[smart_home.png](#)

ESP32 SMARTHOME he puesto de pines virtuales

- **TEMPERATURA** asociado al pin virtual **V1** entero unidades °C mínimo 0 máximo 100 (por ejemplo)
- VENTILADOR asociado al pin virtual V2
- HUMEDAD asociado al pin virtual V3

[2025-10-23 22_19_22-.png](#)

Podemos poner multitud de widgets, vamos a simplificar con

- ESP32 al Template: Temperatura y humedad
- Template al ESP32 un botón para encender y apagar el ventilador

[2025-10-23 22_22_02-Greenshot.png](#)

RETO IoT Blynk

Programa en Steamakersblock

Ya tenemos :

- Template ID
- Template Name
- Token

Lo ponemos junto con los datos de nuestra Wifi

[2025-10-23 22_40_13-Greenshot.png](#)

Para los datos de la ESP32 Temperatura y humedad al template:

[2025-10-23 22_48_34-Greenshot.png](#)

Y para los datos de Template hacia el ESP32, el botón ventilador

[2025-10-23 22_42_15-Greenshot.png](#)

RETO IoT Blynk

Resultado

<https://www.youtube.com/embed/Ps6bnVRFIcE>



RETO IoT Telegram

RETO IoT Telegram

Crear bot en Telegram

[2023-05-30 14_37_46-BotFather - \(68\).jpg](#)

Entramos en nuestro Telegram y chateamos con el creador de los bots: **@BotFather** y nos saldrá esta pantalla:

[2023-05-30 14_32_01-BotFather - \(68\).jpg](#)

Si tecleamos **/start** nos sale las diferentes opciones

[2023-05-30 14_33_30-BotFather - \(68\).jpg](#)

Para crear un nuevo bot, tecleamos **/newbot** y nos preguntará el nombre del bot

Por cierto, el nombre tiene que acabar con las letras **bot**,

NOS PROPORCIONARÁ EL **TOKEN** DEL ROBOT, QUE TOMAREMOS NOTA

[2023-05-30 14_39_59-BotFather - \(68\).jpg](#)

Si tecleamos **/mybots** nos sale los diferentes bots creados y al pulsar en uno de ellos nos salen sus opciones

[2023-05-30 14_36_25-BotFather - \(68\).jpg](#)

RETO IoT Telegram

Encontrar tu ID en Telegram

Buscar tu ID : chat privado

En este caso para que mi Bot me envié mensajes a mi usuario de Telegram directamente busco mi ID.

Vamos a chatear con **@myidbot**

[2023-05-30 14_54_31-IDBot - \(68\).jpg](#)

y le preguntamos por nuestro identificador con **/getid**

TOMAMOS NOTA DE NUESTRO IDENTIFICADOR ID

[2023-05-30 14_47_06-.jpg](#)

Buscar ID de un grupo

En este caso tendríamos que añadir a **@myidbot** al grupo y ejecutar el comando en el chat del grupo **/getgroupid** saldrá un identificador **negativo**

Una vez conseguido el ID podemos eliminar **@myidbot** del grupo

RETO IoT Telegram

Código para enviar y recibir

Bucle Inicio

1. Lo primero es conectar con la wifi
2. Y lo segundo conectar con el **Bot** que hemos creado, gracias al **TOKEN** con la instrucción **INICIAR API TOKEN**
3. Enviamos al **ID** un mensaje de bienvenida con las instrucción **Enviar a Chat ID Mensaje ... Formato**

[2025-10-24 00_06_08-JavierArduino @ Javier Quintana \(6998\).png](#)

Hemos añadido algunos elementos de control, como el LCD

ATENCIÓN si en el mensaje a enviar pones caracteres no standard, puede ser que no funcione. Por ejemplo a mi no me ha funcionado enviar °C

Enviar y recibir

En el evento **TELEGRAM - NUEVO MENSAJE RECIBIDO** se puede utilizar el comando **TELEGRAM MENSAJE** en el siguiente programa se puede ver como en la pantalla del LCD aparecerá quien es el remitente y qué mensaje

Luego envía la temperatura si el usuario de Telegram envía la palabra *temp*

[2025-10-24 00_07_15-JavierArduino @ Javier Quintana \(6998\).png](#)

RETO IoT Telegram

Resultado

<https://www.youtube.com/embed/ZWRA3I0D6sU>

Creditos

Autor:

- CATEDU Octubre 2025 por Javier Quintana Peiró
-

Cualquier observación o detección de error en soporte.catedu.es

Los contenidos se distribuyen bajo licencia **Creative Commons** tipo **BY-NC-SA** excepto en los párrafos que se indique lo contrario.

[image-1648462225402.gif](#)

[image-1648462299882.png](#)

[image-1648462361893.png](#)