

Smart Agriculture Kit micro:bit

- [Introducción](#)
 - [¿Qué es Smart Agriculture Kit para micro:bit ?](#)
 - [Conexiones](#)
 - [Sensores](#)
 - [Actuadores](#)
 - [Objetivos y contenidos](#)
- [Jugando sólo con la micro:bit](#)
 - [Software : El entorno de trabajo de MakeCode](#)
 - [Nuestro primer programa](#)
 - [Retos básicos](#)
 - [Retos avanzados](#)
 - [Música y sonido con micro:bit](#)
 - [Inteligencia artificial Create AI](#)
- [Programas básicos con Smart Agriculture Kit](#)
 - [OLED](#)
 - [DHT11](#)
 - [Sonda sumergible DS18B20](#)
 - [Servo](#)
 - [Sensor de agua](#)

- [Humedad del suelo](#)
- [Led Neopixel](#)
- [Sensor de movimiento PIR](#)
- [Sensor distancia de ultrasonidos](#)

- [Ahora te toca a ti](#)
- [Bluetooth](#)
 - [BLUETOOTH un poco de teoría](#)
 - [Sólo con la microbit: BLUETOOTH programa en Makecode](#)
 - [BLUETOOTH programa Serial Bluetooth Terminal](#)
 - [Avanzado: BLUETOOTH con App Inventor. Extensiones](#)
 - [Avanzado: BLUETOOTH App Inventor programa. Sólo con la micro:bit](#)
 - [Ahora con la granja](#)

- [¿Qué es IoT?](#)
- [Conexión Wifi](#)
- [SMARTIO SPACE](#)
 - [Que es SMARTIOT](#)
 - [Instrucciones en Makecode](#)
 - [Visualización de la cantidad de luz](#)

- [ThingSpeak](#)
 - [ThingSpeak: Cuenta, canal y API KEY](#)
 - [Visualizar temperatura y humedad del DHT11por thinkspeak](#)
 - [SONDA DS18B20](#)
 - [POR QUÉ SE UTILIZA UNA TAZA PARA DESAYUNAR](#)

- [Muro](#)
- [Créditos](#)



Introducción

Introducción

¿Qué es Smart Agriculture Kit para micro:bit ?

Es un kit que [por menos de 100€](#) tenemos muchas posibilidades :

- **Sensores y actuadores** pensados en una domótica en un entorno de agricultura
 - Detectores de presencia
 - Detectores ambientales de temperatura y humedad
 - Detectores de suelo
 - Detectores sumergibles de temperatura y nivel de agua
 - Actuadores: Servo y luces
 - Además de los propios sensores y actuadores de la micro:bit
- **Visualización y registro de los datos**
 - Pantalla OLED
 - ESP32 que permite conexión a Internet, por lo tanto visualización de datos utilizando las técnicas más modernas de IoT
- [Instrucciones de seguridad](#)

[2026-02-06 15_38_40-WhatsApp.png](#)

Fuente Robotopia.es

Se suministran los elementos por separado

[2026-02-08 16_29_37-Greenshot.png](#) Fuente Robotopia.es

Opcionalmente se puede conseguir una maqueta de cartón, para ambientar educativamente todo el conjunto, [las instrucciones las tienes aquí](#) (no hagas caso de las conexiones)

[2026-02-08 16_30_47-Greenshot.png](#)

Fuente Robotopia.es

Introducción

Conexiones

La maqueta es fácil de montar, pero las conexiones requieren destreza. Las conexiones a la placa micro:bit que proponemos son las siguientes:

Elemento	Conexión	Observación
Termómetro sumergible DS18B20	P1	Analógico
Sensor de nivel de agua	P2	Analógico
Led Neopixel	P14	Datos
Sensor humedad del suelo	P4	Analógico
Servo	P9	Pulsos digitales
Sensor movimiento PIR	P3	Analógico
Sensor distancia ultrasonidos	P13	Pulsos digitales
Sensor DHT11	P15	Datos digitales

El Pin3 es compartido con la matriz de leds con la columna vertical.

Como el sensor PIR es pull up ([ver sensores](#)) cuando no detecta es un 1, **por lo tanto se enciende la columna vertical de la matriz de leds de la micro:bit** cuando no detecta movimiento.

Ojo, el fabricante [propone otra configuración](#) que no estamos de acuerdo pues hay que respetar que algunos sensores son analógicos por lo tanto tienen que estar en los pines analógicos de la micro:bit que son P1, P2, P3, P4 y P10.

2026-02-07 22_51_43-WhatsApp.png

Las conexiones hay que respetar la polaridad:

- El cable negro a masa, también marcado como **G** Ground
- El rojo el del medio que es la alimentación **V** Voltios
- El amarillo los datos, también marcado como **S** salida

[2026-02-08 16_35_42-WhatsApp.png](#)

La polaridad también se tiene que respetar en los diferentes elementos, pero es difícil equivocarse pues tiene una pestaña que impide su conexión al revés :

[2026-02-08 16_37_30-WhatsApp.png](#)

Los diferentes elementos se pueden unir a la maqueta de cartón con bridas, tornillos o alambres forrados:

[2026-02-08 16_40_15-WhatsApp.png](#)

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 Fococélula 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 decremента 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 :

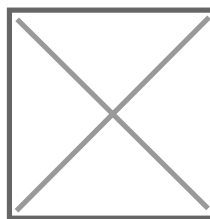
- Curso de [Domótica con Arduino](#)
- [Smart Agriculture Kit micro:bit](#)

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.

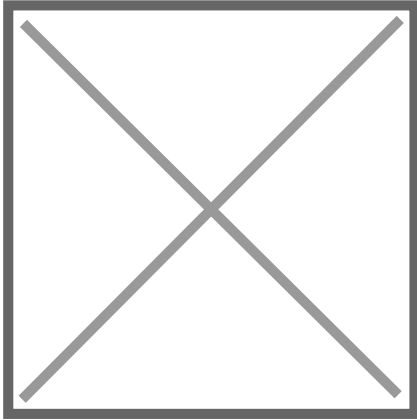


2022-04-14 10_07_05-1.3 Conexiones _ Librería CATEDU.png

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

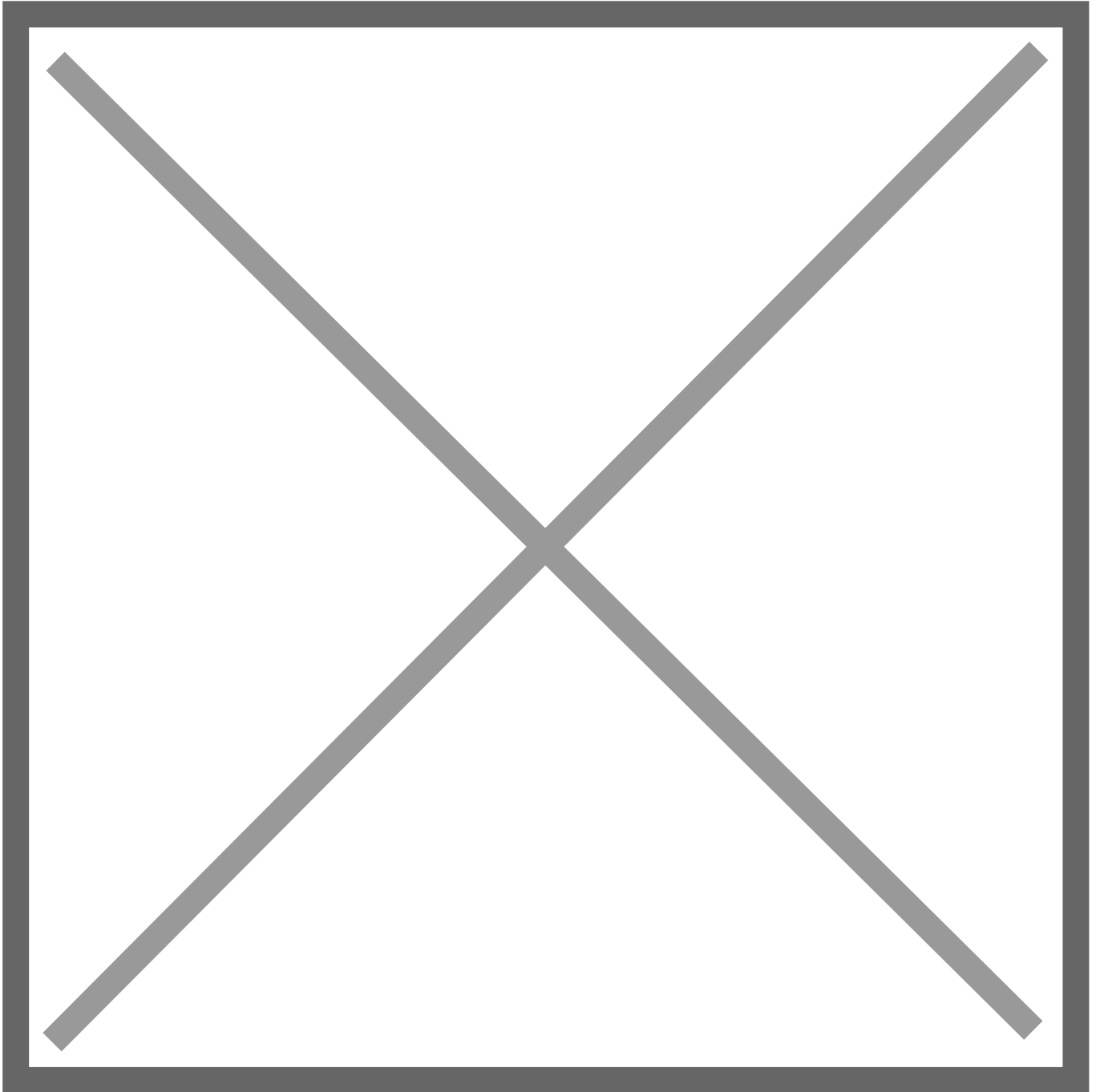
[Datasheet Sparkfun](#)

Para utilizar esta resistencia haremos un **DIVISOR DE TENSIÓN** 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?

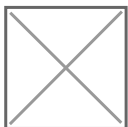
“ Un actuador es el nombre genérico que empleamos para cualquier **dispositivo capaz de realizar acciones en el mundo físico** y que podremos controlar desde un autómata o procesador como Arduino. En particular, usamos el nombre actuador para los dispositivos que son capaces de generar movimiento.

[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](#)

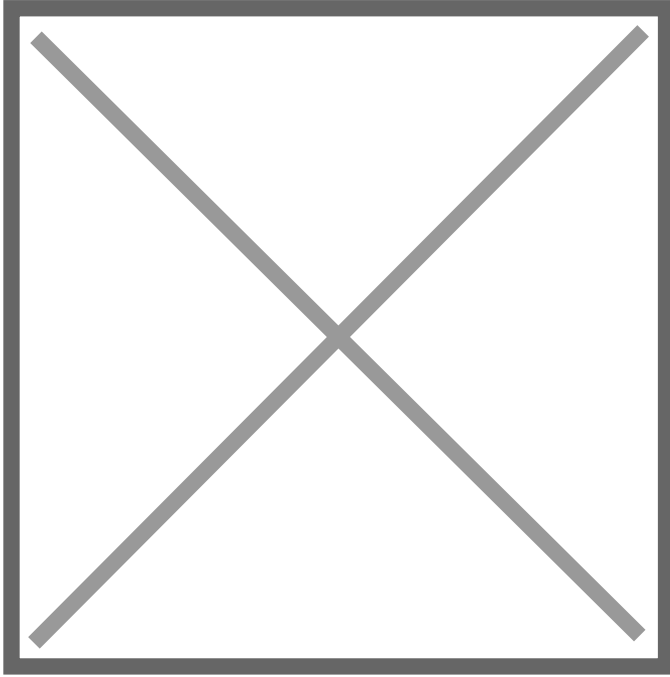


Motores con algo de potencia

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](#)

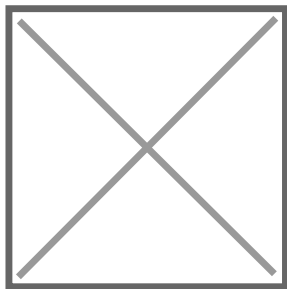
Ejemplos de uso lo puedes ver en

- [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



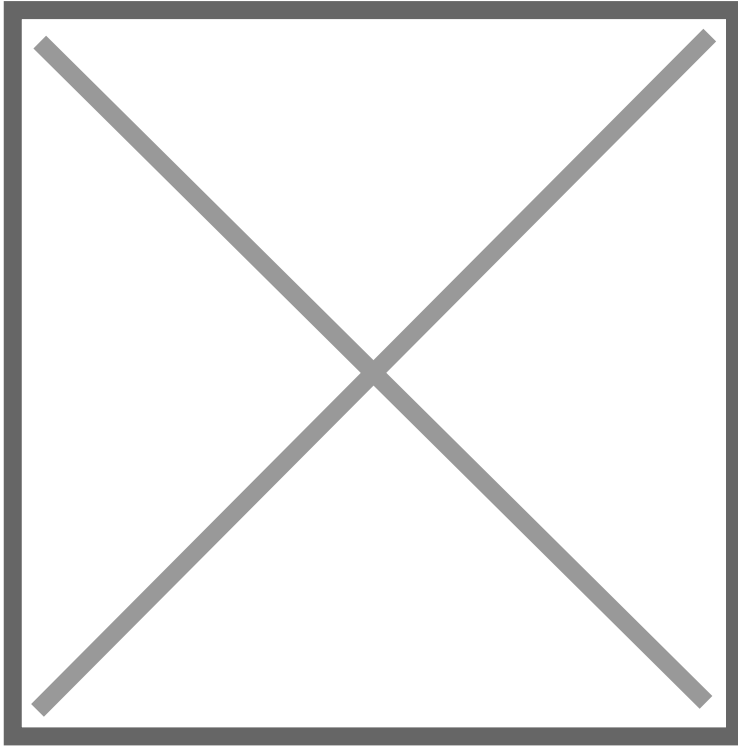
Imagen BY-NC-SA de [Luis Llamas](#)

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.

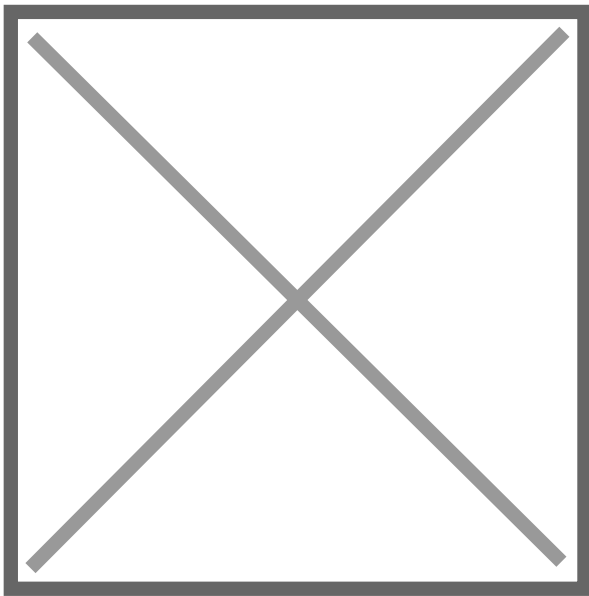
OTRAS SALIDAS

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>

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 y actuadores** con lenguaje de bloques.
 - Retos **STEAM** de nivel medio medio/avanzado en las comunicaciones **Bluetooth**.
 - Retos **STEAM** de nivel medio/avanzado con proyectos **IoT** (Internet de las cosas)

Contenidos

- ¿Qué es Smart Acriculture para micro:bit?
- Introducción sobre lenguajes, actuadores y sensores
- Jugando sólo **con la placa micro:bit** con lenguaje por **bloques Makecode**
- Jugando con la **maqueta** y lenguaje por **bloques Makecode**
- Jugando con **Bluetooth**
- Jugando con **IoT**

Jugando sólo con la micro:bit

Jugando sólo con la micro:bit

Software : El entorno de trabajo de MakeCode

“ El **punto de partida** para el estudio de un lenguaje de programación pasa inevitablemente por los **lenguajes gráficos** o de bloques que facilitan su estudio al mostrar de forma visual las instrucciones del algoritmo programado. (Morales, 2023)

Vamos a utilizar **MakeCode** ofrece un entorno integrado, simple e intuitivo de **codificación por bloques**. Hay que decir que en el entorno de MakeCode también están disponibles los lenguajes **Java** y **Python**, con la ventaja de que un programa desarrollado mediante bloques puede ser traducido instantáneamente a cualquiera de los otros dos lenguajes.

Aquí tienes un esquema de dónde se sitúa Makecode, encaja dentro de los programas de bloque se se carga en la placa

[2025-11-30 10_24_40-ESQUEMA LENGUAJES PROGRAMACIÓN - PowerPoint.png](#)

Para acceder al entorno basta con abrir en un navegador cualquiera el enlace:

<https://makecode.microbit.org/>. Si nuestro navegador tiene desactivadas las cookies o navega en modo incógnito, aparecerá un mensaje de advertencia:

[Advertencia guardado automático.jpg](#) Pulsando sobre el botón de **continuar** aparecerá la ventana principal de MakeCode. Para iniciar el entorno de programación habremos de pulsar sobre el botón de **nuevo proyecto**:

[Nuevo proyecto.jpg](#) MakeCode nos pedirá seguidamente **dar un nombre** cualquiera al proyecto y pulsar el **botón de crear**, tras lo cual se abrirá la ventana del editor.

[Entorno MakeCode.jpg](#)

Entorno Microsoft MakeCode. Captura de pantalla



Para trabajar con MakeCode no se requiere registro. No obstante, pulsando sobre el icono de la esquina superior derecha, situado en la barra de herramientas, es posible acceder a MakeCode desde una cuenta de Google o de Microsoft. Esta característica permite guardar en la nube los diseños realizados y acceder a ellos desde cualquier ordenador.

Podemos ajustar el **idioma de la interfaz** pulsando sobre la rueda dentada de la esquina superior derecha. Están disponibles 33 idiomas, incluyendo el español.

El panel de la izquierda representa el anverso de una placa micro:bit. Se trata de un **simulador**, que mostrará en tiempo real el efecto del programa en desarrollo. Nótese que no es necesario disponer de una placa real para realizar los primeros pasos en programación con micro:bit.

Con el puntero del ratón se puede interactuar con la placa y pulsar sus botones, agitarla o cambiar las condiciones simuladas de luz o de ruido. Por otro lado, los controles situados bajo la placa nos permiten:

[Controles simulador.jpg](#)

- Apagar o encender el simulador.
- Reiniciar el simulador, es decir, el programa en ejecución.
- Entrar en modo de depuración para localizar y solucionar errores.
- Activar o desactivar el altavoz del ordenador.
- Tomar una fotografía de la placa.
- Mostrar el simulador en pantalla completa.

El panel central es una **caja de herramientas** en forma de menú desplegable con todos los **bloques de programación** disponibles. Los bloques pueden ser arrastrados directamente con el ratón desde el menú. Cada color del menú se identifica con un tipo de acción o de estructura de control. Por ejemplo, el color verde se reserva para los bloques que permiten ejecutar acciones repetitivas o bucles.

[Menú bucles.jpg](#)

La **caja de búsqueda** colocada sobre el menú desplegable es muy útil para encontrar rápidamente el bloque más adecuado para realizar la acción deseada en cada momento.

El panel de la derecha es el **área de programación**. Aquí se escriben los programas arrastrando y encajando los bloques de código. Cuando se abre el navegador aparecen por defecto dos bloques azules de eventos del **menú básico**:

- **Al iniciar**, es la estructura que contendrá el algoritmo que será ejecutado por la placa sólo una vez, tras ser encendida o reiniciada.



- **Para siempre**, es el bucle que contendrá el algoritmo que se ejecutará indefinidamente, una y otra vez, cuando acabe la ejecución del algoritmo de inicio.

Pulsando con el botón derecho del ratón sobre un bloque se desplegará un **menú contextual** que dará varias opciones, como duplicar, borrar u obtener ayuda sobre el uso del bloque.[C](#)

[Menú contextual bloques.jpg](#)

Al colocar el puntero del ratón sobre un bloque, el editor mostrará una **etiqueta de ayuda** con una breve explicación sobre la función del bloque.

[Hints.png](#)

<https://makecode.microbit.org/reference> contiene explicaciones detalladas de los bloques de MakeCode.

En la parte inferior de la ventana de MakeCode se dispone la **barra de herramientas**, con botones para realizar acciones como descargar el programa en un placa micro:bit conectada al ordenador, guardar el programa en nuestro ordenador o en un repositorio en línea, hacer y deshacer y ampliar o reducir la imagen.

Jugando sólo con la micro:bit

Nuestro primer programa

El primer programa que se realiza cuando se comienza a aprender un nuevo lenguaje consiste casi siempre en presentar un **saludo por pantalla**. Micro:bit es más original, puesto que propone como primer programa un **corazón palpitante**.

Vamos a combinar el saludo inicial con el corazón. De esta forma ilustraremos el funcionamiento de los bloques de eventos **al inicio** y **para siempre**.

Se trata de seguir los pasos explicados en la página anterior y crear un programa llamado **Hola**. Tras la apertura del editor pulsaremos sobre el menú **Básico** y arrastraremos y encajaremos los bloques para configurar el siguiente programa:[Hola.jpg](#)

Para escribir el texto ¡Hola! basta con hacer clic sobre la caja blanca del bloque **mostrar cadena**. De igual forma, los iconos de los bloques **mostrar icono** y los números de los bloques **pausa (ms)** pueden seleccionarse pulsando sobre las flechas, acción que despliega menús de selección.

[Desplegable de bloque.jpg](#)

Como en el caso de la caja de texto, también puede teclearse directamente el número de milisegundos del bloque **pausa (ms)**.

¿Qué hace el programa? Cuando se enciende la placa, el **algoritmo de inicio mostrará una vez** mediante la matriz de LED la cadena de texto "¡Hola!". Como la pantalla es muy pequeña, micro:bit deslizará el texto horizontalmente, de derecha a izquierda.

Seguidamente, el **algoritmo principal** mostrará en pantalla el icono de un corazón grande y entrará en pausa 500 milisegundos, es decir, medio segundo. Pasado este tiempo, micro:bit mostrará un corazón pequeño y esperará otro medio segundo. Este algoritmo se ejecutará indefinidamente hasta que se apague la placa, se pulse el botón de reinicio o se acaben las pilas. El efecto visual será el de un **corazón palpitante**.

Nada más terminar de colocar el último bloque, el **simulador de MakeCode** mostrará en pantalla el resultado del programa en ejecución. Para reiniciar el programa hay que pulsar sobre el icono de reinicio, situado justo bajo el dibujo de la placa micro:bit.

[Saludo.gif](#)

Cargar el programa en micro:bit

Hay que conectar primero la placa al ordenador por medio del cable USB. Después de unos instantes, el sistema operativo, (Linux, Windows o MacOS) reconocerá a la tarjeta micro:bit conectada como si se tratase de una unidad USB con el nombre MICROBIT.

[Microbit USB Mint.jpg](#) *La unidad Micro:bit en el escritorio de Linux Mint*

Si la tarjeta es nueva, llevará cargado de fábrica un programa de demostración que empezará a ejecutarse nada más ser conectada al puerto USB. El programa mostrará mensajes, iconos en movimiento y sonidos y pedirá al usuario que calibre el compás magnético.

METODO RÁPIDO: EMPAREJAMIENTO (recomendable)

La primera vez que le damos a descargar, nos pide que conectemos la MICRO:BIT

[2025-11-02 19_03_44-WhatsApp.png](#)

La primera vez nos muestra la opción de EMPAREJAR

[2025-11-02 19_04_58-Greenshot.png](#)

Nos pregunta por la tarjeta que tiene que estar conectada al USB

[2025-11-02 19_06_36-Greenshot.png](#)


Una vez emparejado, ya sale la opción de descargar

[2025-11-02 19_08_18-Greenshot.png](#)

Empieza a cargarse

[2025-11-02 19_09_02-Greenshot.png](#)

Y YA ESTA

 La siguiente vez que utilices esa micro:bit automáticamente pasa a descargar de forma muy rápida.

METODO QUE NO FALLA : DESCARGAR COMO ARCHIVO Y COPIAR Y PEGAR EN LA NUEVA UNIDAD USB CREADA



Seguidamente hay que pulsar sobre el botón **Descargar**, acción que abrirá el diálogo de guardado de archivos del sistema operativo. Bastará con seleccionar la unidad MICROBIT y confirmar la descarga.

[Descargar.jpg](#)

Si tienes Vitalinux, te recomendamos visitar <https://libros.catedu.es/books/vitalinux-y-robotica/page/r8-microbit>

Mientras dure la transferencia de datos, el **LED ámbar** situado junto al conector USB de micro:bit **parpadeará rápidamente**, quedando fijo cuando el programa haya sido transferido.

La transferencia de archivos a micro:bit lleva algo de tiempo, especialmente si el programa es largo. Tras la transferencia, el programa empezará a ejecutarse inmediatamente sobre la tarjeta micro:bit.

La desventaja es que es un método LENTO y no se reduce la velocidad en los siguientes intentos, no como en el método anterior

¿Y ahora qué?

Pues simplemente la placa ejecutará el programa cargado cuando se le alimente.

Sólo es posible cargar un programa a la vez en micro:bit, por lo que cada nuevo programa se reescribe sobre el anterior.

Si se desea que micro:bit funcione desconectado del ordenador, simplemente hay que alimentarlo por medio de la caja de pilas o mediante una batería externa o un cargador USB. El programa se ejecutará de igual forma, pues se encuentra guardado en la memoria de la placa.

[embedded-image-aT6h3zXI.jpeg](#)

Jugando sólo con la micro:bit

Retos básicos

INDICE DE RETOS EN

<https://libros.catedu.es/books/microbit-car/chapter/retos-con-microbit>

[Panel LED](#)

[Jugando con los LED: bloques básicos de código](#)

[Sensor temperatura](#)

[Sensor luminosidad y sonido](#)

[Pulsadores](#)

[Acelerómetro](#)

[Brújula](#)

[Radio](#)

[Pines entrada y salida](#)

Jugando sólo con la micro:bit

Retos avanzados

Los tienes en <https://libros.catedu.es/books/microbit-car/chapter/retos-avanzados-con-microbit>

[Mejorando el termómetro](#)

[Aprender con un led la diferencia entre analógico y digital](#)

[Bloques de entradas: sonómetro, magnetómetro y acelerómetro con una sola línea de código](#)

[Nivel de burbuja usando el giroscopio](#)

[Lectura puerto USB por MakeCode o Coolterm](#)

Jugando sólo con la micro:bit

Música y sonido con micro:bit

Lo tienes en <https://libros.catedu.es/books/microbit-car/chapter/musica-y-sonido-con-microbit>

[Una pequeña melodía](#)

[Grabadora de sonidos](#)

[Fray Santiago](#)

[Controles de volumen, tempo y tono](#)

Jugando sólo con la micro:bit

Inteligencia artificial Create AI

Lo tienes en <https://libros.catedu.es/books/microbit-car/chapter/create-ai>

Qué es CreateAI

Es una aplicación web en [a https://createai.microbit.org/](https://createai.microbit.org/) ideada para realizar estos pasos: R...

Vestibles

Para proyectos con CreateAI es recomendable fijar la micro:bit a una pulsera. Fuente Ro-botic...

Paso 1 Collect data por Bluetooth y 1 micro:bit

Si trabajamos con un PC sin bluetooth no puedes usar esta página, mira esta página Conectamos nu...

Paso 2 Test model

Grabando una acción Vamos a darle una acción, elegimos un icono, uno nombre y una acción, en es...

Paso 3 Code

Una vez testado nuestro modelo pulsamos a Edit in MakeCode AHORA ES NUESTRO MOMENTO DE PROGRA...

Sin Bluetooth y 2 micro:bit

La idea es utilizar una micro:bit que haga de "puente inalámbrico" utilizando la conexión vía rad...

Entendiendo a mi peluche

Vamos a a reconocer diferentes gestos de un muñeco peluche. Extraído de <https://microbit.org/proj...>



Más proyectos CreateAI

Con el mismo principio de detectar con la AI movimientos con el acelerómetro, se pueden hacer div...

Programas básicos con Smart Agriculture Kit

Programas básicos con Smart Agriculture Kit

OLED

“ Un OLED (Organic light-emitting diode) es un tipo de LED en el que la capa emisiva es está formada por un compuesto orgánico que emite luz en respuesta a la electricidad...

...Las pantallas OLED tienen la ventaja de tener un consumo muy bajo, en torno a 20mA, dado que solo se enciende el pixel necesario y no requieren de backlight. Esto es especialmente interesante en aplicaciones que funcionan con baterías. Además, tienen una mejor visibilidad en ambientes luminosos, como bajo el sol. Aunque el pequeño tamaño de estas pantallas OLED de 0.96” pueden ser un problema para su correcta visualización.

Luis Llamas CC-BY-NC-SA <https://www.luisllamas.es/conectar-arduino-a-una-pantalla-oled-de-0-96/>

La pantalla OLED nos permitirá visualizar variables y mensajes muy fácilmente :

Para tener las instrucciones OLED instalaremos la extensión de ELECFREAKS

[2026-02-06 18_06_21-Greenshot image editor.png](#)

E instala estas instrucciones

[2026-02-08 00_24_01-WhatsApp.png](#)

En este programa visualiza el nivel de luz (el sensor está integrado en la placa microbit) en el OLED:

1. Inicializa la pantalla OLED como 128x64 píxeles
2. Limpia de anteriores mensajes
3. Enseña la frase (con salto de línea)
4. Enseña frase sin salto de línea
5. Enseña número con salto de línea

[2026-02-07 23_46_39-Greenshot.png](#)

El resultado:



2026-02-07 23_48_52-WhatsApp.png

<https://makecode.microbit.org/S69148-73800-11713-89082>

<https://makecode.microbit.org/#pub:S69148-73800-11713-89082>

Programas básicos con Smart Agriculture Kit

DHT11

La lectura de este sensor que es doble, lleva un protocolo que lo hemos visto en el apartado de **sensores** por lo que utilizaremos alguna función definida de alguna extensión.

La extensión recomendada es la que instala el fabricante ELECFREAKS en la extensión Environment

[2026-02-06 18_06_21-Greenshot image editor.png](#)

La instrucción esta un poco escondida

[2026-02-07 22_11_14-Greenshot.png](#)

Para ver su funcionamiento, vamos a visualizar su valor en la pantalla OLED :

[2026-02-07 22_13_43-Greenshot.png](#)

<https://makecode.microbit.org/S31229-07212-54331-00904>



<https://makecode.microbit.org/#pub:S31229-07212-54331-00904>

El resultado

2026-02-07 22_16_10-WhatsApp.png

Programas básicos con Smart Agriculture Kit

Sonda sumergible DS18B20

“ El sensor DS18B20 es un sensor barato y, sin embargo, bastante avanzado. Dispone de un rango amplio de medición de -55°C a $+125^{\circ}\text{C}$ y una precisión superior a $\pm 0.5^{\circ}\text{C}$ en el rango -10°C de $+85^{\circ}\text{C}$. Una de las ventajas del DS18B20 es que se comercializa tanto en un integrado TO-92 como en forma de sonda impermeable, lo que permite realizar mediciones de temperatura en líquidos y gases.

Luis LLamas CC-BY-NC-SA <https://www.luisllamas.es/temperatura-liquidos-arduino-ds18b20/>

Esto nos permite hacer mediciones en agua incluso hirviendo.

Recomendamos usar una extensión específica para esta sonda, buscar como DS TEMP

[2026-02-08 00_08_57-WhatsApp.png](#)

Esto nos instala estas dos instrucciones:

[2026-02-08 00_11_14-C__Users_HP_Downloads_2026-02-08 00_08_57-WhatsApp.png](#) - Greenshot ir

El siguiente programa:

1. Prepara OLED
2. Esta rutina permite visualizar por el OLED si hay algún error
3. Grabamos en la variable TEMPERATURA el valor que lee la sonda en el P1
4. Visualizamos el valor

[2026-02-08 00_18_11-WhatsApp.png](#)

<https://makecode.microbit.org/S42317-53460-45843-40235>


<https://makecode.microbit.org/#pub:S42317-53460-45843-40235>

Aquí podemos ver que marca cerca de 90°C en una infusión:

[2026-02-08 00_18_55-WhatsApp.png](#)

Programas básicos con Smart Agriculture Kit

Servo

Nos da mucho juego y permite muchos proyectos. El fabricante sólo proporciona uno  que sólo cuesta 2€ !!

Para utilizarlo **no hace falta ninguna extensión** en pines tenemos las instrucciones correspondientes

Vamos a hacer un programa para que veas **que sólo puede girar de 0º a 180º**

1. Creamos una variable **ángulo** y al principio le asignamos el valor de 0 por lo tanto obligamos que empiece con ese valor
2. Fijamos el ángulo del servo que esta en el pin P9 al valor de la variable ángulo dentro del bucle para siempre. Una pausa breve de ms **que podemos bajar si queremos que vaya más rápido**
3. si la variable ángulo supera los 180 que vuelva a "resetearse" a 0
4. la variable ángulo se va incrementando de 10 en 10 con otra pausa. Estos valores se pueden cambiar para conseguir la velocidad deseada

2026-02-08 10_41_35-C__Users_HP_Downloads_2026-02-08 10_25_18-WhatsApp.png - Greenshot in

<https://makecode.microbit.org/S99415-22198-23132-97526>

<https://makecode.microbit.org/#pub:S99415-22198-23132-97526>

RETO: Crear un programa que alimente automáticamente a los animales, mediante una programación semanal, es decir dado una fecha, que se active el servo que abrirá la compuerta del silo durante un tiempo determinado. **Solución**

RETO: Crear un programa que proteja las plantas de la luz directa, es decir si la luz es mayor que un cierto valor, se active el servo que pondrá una sombra a la planta. **Solución**

Programas básicos con Smart Agriculture Kit

Sensor de agua

“ Este tipo de sensores **detectan la presencia de lluvia por la variación de conductividad** del sensor al entrar en contacto con el agua. *Luis Llamas CC-BY-NC-SA* <https://www.luisllamas.es/arduino-lluvia/>

Para este sensor **no utilizaremos ninguna extensión especial**, la instrucción **lectura de pin analógico** correspondiente es suficiente y fiable, no tiene protocolo de datos como los otros sensores :

1. Inicializamos el OLED
2. Grabamos en una variable nueva AGUA el valor del pin analógico P2
3. Visualizamos la lectura

[2026-02-08 00_29_18-WhatsApp.png](#)

<https://makecode.microbit.org/S42209-00876-15805-47351>

<https://makecode.microbit.org/#pub:S42209-00876-15805-47351>

Aquí vemos el resultado

[2026-02-08 00_35_39-WhatsApp.png](#)

RETO: VISUALIZALO EN FORMA DE TANTO POR CIENTO:

La variable analógica va de 0 a 1024 si quieres que visualice valores de 0 a 100 tienes que "mapear" o "cambio de variable" una forma de hacerlo es con la instrucción. (también hay otra de mapear en "Pines").

O de forma casera, con una regla de 3, es decir multiplicarlo por $100/1024$

[2026-02-08 00_38_34-WhatsApp.png](#)

Programas básicos con Smart Agriculture Kit

Humedad del 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/>

Te proponemos este sencillo programa para poder calibrar y medir las humedades del suelo :

1. Inicializamos el OLED
2. Grabamos en una variable la lectura del pin P4
3. Visualizamos en el OLED la lectura

Para este sensor **no utilizaremos ninguna extensión especial**, la instrucción **lectura de pin analógico** correspondiente es suficiente y fiable, no tiene protocolo de datos como los otros sensores :

[2026-02-08 10_05_17-WhatsApp.png](#)

<https://makecode.microbit.org/S85526-52667-76006-02288>

<https://makecode.microbit.org/#pub:S85526-52667-76006-02288>

2026-02-08 10_08_19-WhatsApp.png

RETO: VISUALIZALO EN FORMA DE TANTO POR CIENTO:

La variable analógica va de 0 a 1024 si quieres que visualice valores de 0 a 100 tienes que "mapear" o "cambio de variable" una forma de hacerlo es con la instrucción. (también hay otra de mapear en "Pines").

O de forma casera, con una regla de 3, es decir multiplicarlo por $100/1024$

2026-02-08 00_38_34-WhatsApp.png

RETO: Este sensor junto con el servo pide a gritos un proyecto de riego automático: ¿Te atreves?

<https://www.youtube.com/embed/xm27jGEysfl>

Programas básicos con Smart Agriculture Kit

Led Neopixel

Los leds Neopixels son "chulos" cuando hay muchos, pues los datos van pasando de un led a otro y tiene mucho juego pero en este caso el fabricante sólo ha puesto uno ☐

“

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/>

Para utilizar este elemento tenemos que descargarnos la Extensión Neopixel y tendremos las instrucciones adecuadas.

Vamos a hacer UNA LUZ CREPUSCULAR PARA LA GRANJA es decir que cuando sea de noche se encienda la luz

1. Inicializamos el led neopixel en el pin P14 con 1 led y formato GRB (se refiere al orden de envío de las instrucciones, primero el green, luego red y luego blue, hay otro formato que es más común el RGB)
2. Si la luz ambiental (detectado por el sensor integrado en la placa micro:bit) es menor que 50 (un valor arbitrario para detectar si es de noche o es de día)



3. Apaga la luz si es de día
4. Enciende la luz si es de noche

2026-02-08 10_25_18-WhatsApp.png

<https://makecode.microbit.org/S47979-61604-21560-37629>

<https://makecode.microbit.org/#pub:S47979-61604-21560-37629>

Programas básicos con Smart Agriculture Kit

Sensor de movimiento PIR

“

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/>

Para este sensor **no utilizaremos ninguna extensión especial**, la instrucción **lectura de pin analógico** correspondiente es suficiente y fiable, no tiene protocolo de datos como los otros sensores :

Vamos a hacer una alarma sencilla:

1. Inicializamos el OLED
2. Si la lectura del sensor es mayor que 100 (valor arbitrario) es que hay "alguien"
3. Si hay alguien
 1. lo visualizo por el OLED
 2. sale la palabra INTRUSO
 3. y reproduce un tono de alarma
4. Si no hay nadie
 1. sale el valor por el OLED
 2. sale que no hay nadie
5. una breve pausa para no saturar la placa

2026-02-08 10_53_55-C__Users_HP_Downloads_2026-02-08 10_25_18-WhatsApp.png - Greenshot in

PROPUESTA: Ajusta el valor arbitrario del paso 2 para que detecte personas pero no animales

<https://makecode.microbit.org/S47327-01515-95700-67184>

<https://makecode.microbit.org/#pub:S47327-01515-95700-67184>

RETO: VISUALIZALO EN FORMA DE TANTO POR CIEN:

La variable analógica va de 0 a 1024 si quieres que visualice valores de 0 a 100 tienes que "mapear" o "cambio de variable" una forma de hacerlo es con la instrucción. (también hay otra de mapear en "Pines").

O de forma casera, con una regla de 3, es decir multiplicarlo por $100/1024$

[2026-02-08 00_38_34-WhatsApp.png](#)

Programas básicos con Smart Agriculture Kit

Sensor distancia de ultrasonidos

Hardware

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 Trig 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](#)

Software

La extensión recomendada es la que instala el fabricante ELECFREAKS en la extensión Environment

[2026-02-06 18_06_21-Greenshot image editor.png](#)

Y en una sección que pone Octopus

[2026-02-08 11_07_48-WhatsApp.png](#)

Vamos a hacer un programa que visualice las mediciones:

1. Preparamos la OLED
2. En una variable nueva DISTANCIA recoge la medida del sensor que está conectado en el pin P13



3. visualiza la DISTANCIA en el OLED

2026-02-08 11_08_48-WhatsApp.png

<https://makecode.microbit.org/S66928-56344-60970-73771>

<https://makecode.microbit.org/#pub:S66928-56344-60970-73771>

En la foto me he colocado a aproximadamente 11cm de los "ojos"

2026-02-08 11_11_43-WhatsApp.png

Ahora te toca a ti

Una vez visto los programas básicos para aprender cada una de las piezas **te toca a ti realizar el puzle**

[jonny-gios-SqjhKY9877M-unsplash.jpg](https://www.unsplash.com/photo-1511362768113-045f58d47711)

Foto de [Jonny Gios](#) en [Unsplash](#)

¿Qué queremos decir? La *Smart agriculture kit para micro:bit* NO esta diseñada para programar cada vez un solo un elemento. Si fuese así no sería necesario ni maqueta. Hace falta idear proyectos que integran las diferentes piezas que consolidan un verdadero pensamiento computacional.

¿Por qué NO desarrollamos estos proyectos en este curso? Por que es la parte que el profesor con el alumnado tienen que desarrollar, idear desde cero. Si publicamos las soluciones, sería ver y copiar.

Propuestas

Todas las que la imaginación pueda, aquí te dejamos unas evidentes que pide a gritos la maqueta :

1. **RIEGO AUTOMÁTICO** realizar un programa que detecte la humedad del suelo, y cuando llegue a cierto margen de "sequía" active el servo para regar. El servo vuelve a su posición cuando el detecto de humedad vuelve a recuperar la humedad desesada.
2. **DETECCIÓN DEL NIVEL DE AGUA DEL DEPOSITO.** Utilizar el detector de agua que si no hay suficiente nivel en el depósito de riego, y que avise mediante una señal (acústica o luminosa o ambas) de que falta agua en el depósito.
3. **RIEGO AUTOMÁTICO + DETECCIÓN DE NIVEL DE AGUA DE DEPOSITO** jo! que guay juntar los dos anteriores proyectos !! , Si el nivel del agua del depósito baja de un determinado valor, que no riegue (o sea el servo que permanezca quieto)
4. **REGULACION TEMPERATURA EN LA GRANJA** si la temperatura ambiente (por el DHT11) es elevada, que se active el servo que abre una ventana. Que cierre cuando la temperatura es baja.
5. **RIEGO + CONTROL TEMPERATURA**, necesitaríamos dos servos, no lo proporciona el kit pero [un servo cuesta 2€](#)
6. **REGULACIÓN DE CONTROL DE TEMPERATURA EN LUGARES CRÍTICOS** Por ejemplo en una granja se desea que el lugar donde se almacenan las vacunas no sobrepase ciertos valores límites. La sonda termométrica DS18B20

Bluetooth

Bluetooth

BLUETOOTH un poco de teoría

ONDAS

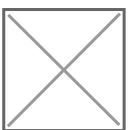
Una onda es una señal que se propaga por un medio. Por ejemplo el sonido, que es una onda mecánica que viaja usando el aire o cualquier otro material. Pero en el caso de las señales eléctricas pueden ser enviadas por el cable o a través del vacío (no necesitan un medio para transmitirse).

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

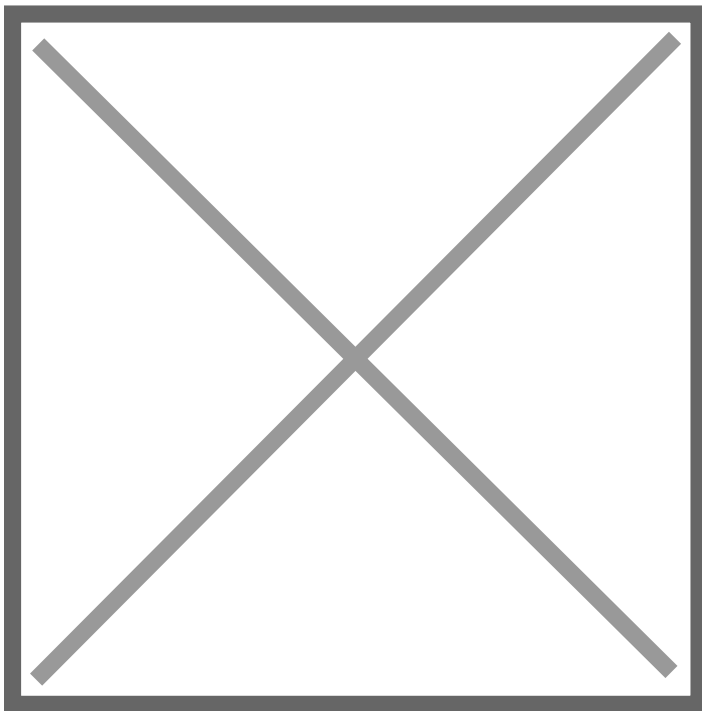


también se encuentran todas las redes WIFI que usamos a diario. En concreto funcionan a 2,4GHz. (Un G son 10^9) luego entre FM y Microondas.

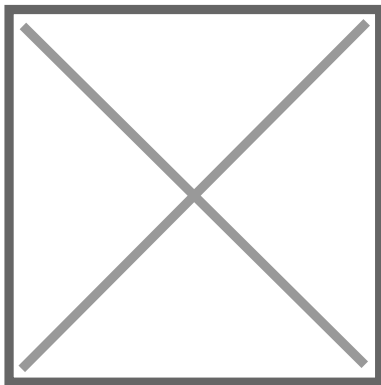
¿Sabías que?

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](#)

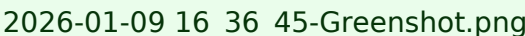
2025-12-05 08_05_39-WhatsApp.png

Bluetooth

Sólo con la microbit: BLUETOOTH programa en Makecode

Realizamos un programa que :

- En **inicio**
 - se active el servicio UART para el envío y recepción de mensajes,
 - muestra un mensaje del nombre de la micro:bit, ver más abajo
- **Al conectar Bluetooth** que muestre un check
- **Al desconectar Bluetooth** que muestre X
- **Al recibir datos**, hasta # (puede ser otro carácter) que muestre la frase recibida
- **Al presionar el botón A**
 - Que muestre un mensaje
 - Que muestre la temperatura

¿Para qué mostrar el nombre de la micro:bit? Para saber a qué micro:bit conectarte. En una clase con muchas micro:bit es importante este dato. El nombre del equipo está en 

<https://makecode.microbit.org/S60585-58735-21378-05922>



<https://makecode.microbit.org/#pub:S60585-58735-21378-05922>

Bluetooth

BLUETOOTH programa Serial Bluetooth Terminal

Entramos con el móvil a Google Play e instalamos esta aplicación

https://play.google.com/store/apps/details?id=de.kai_morich.serial_bluetooth_terminal

2025-11-02 21_50_49-Greenshot.png

Tiene la ventaja de

- **Enviar** mensajes
- **Recibir** mensajes
- Permitir conexiones **BLUETOOTH LE** (Low emission) **que es lo que utiliza MICRO:BIT**

Entramos en **Devices** y en **Bluetooth LE** y nos conectamos a la Micro:bit

[unnamed.webp](#)

Una vez conectado, podemos:

- enviar un mensaje, que como hemos definido anteriormente en Makecode tiene que ir entre #
- recibir un mensaje, se visualizará lo que nos envíe la micro:bit que en Makecode lo hemos programado al apretar el botón A

<https://www.youtube.com/embed/H0HDVPmX-tE>

Bluetooth

Avanzado: BLUETOOTH con App Inventor. Extensiones

Descargas e instalación

Necesitamos estas extensiones para poder crear una APP que pueda enviar y recibir con nuestra micro:bit

Página de descargas <https://iot.appinventor.mit.edu/#/>

[2025-11-02 22_05_10-.png](#)

Una vez descargadas, vamos al APP INVENTOR <https://ai2.appinventor.mit.edu> y las instalamos en extensiones :

[2025-11-02 22_08_33-Mattermost Desktop App.png](#)

Una vez instaladas, se visualizan como extensiones abajo del menú. Las dos últimas son las que utilizaremos:

[2025-11-02 22_10_51-.png](#)

La extensión Bluetooth tiene diversas funciones que tienes su descripción [aquí](#) (English). Para poder instalarla, vamos al APP INVENTOR <https://ai2.appinventor.mit.edu>

Bluetooth

Avanzado: BLUETOOTH App Inventor programa. Sólo con la micro:bit

En DESIGNER

incorporamos:

1. **HorizontalArrangement** para que los botones queden alineados horizontalmente
2. **Botones**
 1. Scan
 2. Stop
 3. Conectar
 4. Desconectar
3. **Label** que dirá el estado de la conexión. Lo llamaremos **LabelEstado**
4. **ListView** que lo llamaremos **ListBLE** donde mostrará los diferentes dispositivos Bluetooth LE que detecta
5. **TextBox** para poner el texto que queramos a enviar a micro:bit
6. Un **botón Enviar** el texto anterior
7. Un **Label** que lo llamaremos **LabelTextoRecibido** que mostrará el mensaje desde micro:bit
8. Añadimos los elementos de las extensiones que hemos instalado anteriormente
 1. Microbit_UART_Simple
 2. BluetoothLE

[2025-11-02 22_13_06-Mattermost Desktop App.png](#)

En Blocks

Cuando escaneemos, que el elemento empiece el escaneado y la lista se vuelva visible, además de que LabelEstado diga que esta escaneando

[2025-11-02 22_24_04-Greenshot.png](#)

Si ha encontrado un dispositivo, que lo vaya añadiendo a la lista ListBLE



[2025-11-02 22_25_21-Mattermost Desktop App.png](#)

Cuando le digamos que pare, simplemente se lo mandamos al dispositivo y LabelEstado lo informa

[2025-11-02 22_24_44-Mattermost Desktop App.png](#)

Cuando le demos a conectar, pues conecta con el seleccionado en ListBLE y LabelEstado informa

[2025-11-02 22_25_49-Greenshot.png](#)

Si conecta, pues LabelEstado informa y ListBLE no es necesaria por lo tanto se oculta, pues entorpece la visión

[2025-11-02 22_26_36-Greenshot.png](#)

Si queremos desconectar, pues le decimos al elemento BluetoothLE que desconecte

[2025-11-02 22_27_31-Greenshot.png](#)

Si se ha desconectado (voluntariamente al dar al botón anterior, o involuntariamente pues el dispositivo se ha desconectado, o esta muy lejos... etc) que informe

[2025-11-02 22_28_08-Mattermost Desktop App.png](#)

Si apretamos el botón enviar, le enviamos el texto que esta en TextBox entre "#" pues así lo hemos definido en el programa Makecode

[2025-11-02 22_30_07-Mattermost Desktop App.png](#)

Si se ha recibido un mensaje, pues que lo visualice, pero primero comprueba que el mensaje no este vacío

NOTA el mensaje "**message**" lo arrastras desde la instrucción "**when..**" tal y como señala la línea roja

[2025-11-02 22_30_57-Mattermost Desktop App.png](#)

[basicoBluetoothLE.aia](#)

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.

A jugar...

<https://www.youtube.com/embed/ZS5d-xrDVcA>

Bluetooth

Ahora con la granja

Una vez visto cómo podemos controlar la micro:bit usando Bluetooth, vamos a hacer lo mismo con la granja

1. Empezamos instalando la **Extensión Bluetooth** que hemos visto en las páginas anteriores
2. También pondremos la Extensión **lot-environment** kit de Elecfreak para poder usar la OLED, DHT11 etc.. de la granja tal y como lo hemos trabajado en el capítulo [Programas básicos con Smart Agriculture Kit](#)
3. En el bucle **Al inicial**, inicializamos la **OLED**, y también inicializamos el **Bluetooth** pero aquí el nombre de dispositivo que lo visualice por la OLED que es más cómodo, y así sabemos a qué microbit tenemos que conectarnos si estamos en una clase con muchos: [2026-02-09 16_44_51-Greenshot.png](#)
4. Añadimos el bucle "**Al conectar Bluetooth**" y "**Al desconectar Bluetooth**" que también lo hemos visto en las páginas anteriores, pero esta vez le decimos que los mensajes lo visualice por la OLED [2026-02-09 16_44_07-Greenshot.png](#)
5. Vamos a jugar ahora que si detecta agua en el sensor de nivel de agua, que mande un mensaje de inundación : [2026-02-09 16_46_38-Greenshot.png](#)
6. Vamos a jugar más y si aprieto el botón A de la micro:bit que salga un mensaje y además que nos diga la temperatura y la humedad de la granja: [2026-02-09 16_47_45-Greenshot.png](#)
7. No podemos abandonar el reto sin poner comunicación en el otro sentido, es decir de la APP a la granja: Vamos a hacer que si se envía desde la APP #abre# pues que el servo se ponga a 0º y se envía desde la APP #cierra# pues que el servo se mueva a 180º. El servo podría estar unido a una puerta, ventana, a un silo de alimentación... [2026-02-09 16_50_35-Greenshot.png](#)

El proyecto <https://makecode.microbit.org/S87280-53966-04815-61300>



<https://makecode.microbit.org/#pub:S87280-53966-04815-61300>

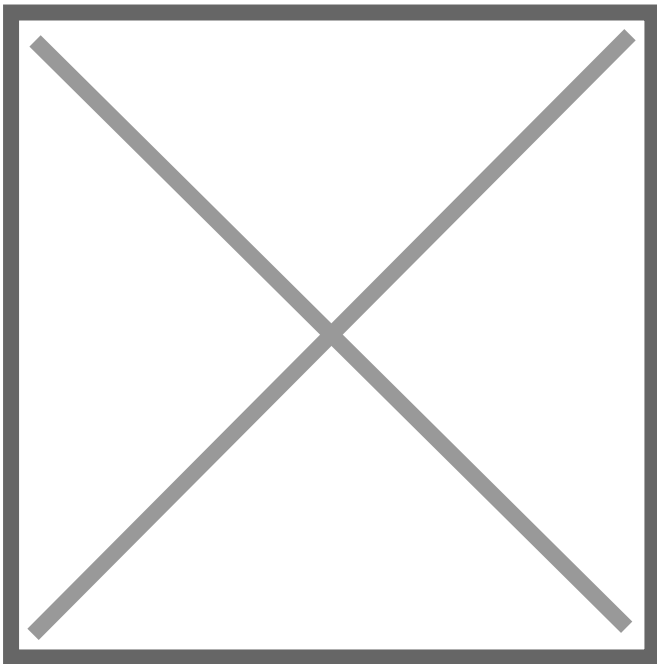
Y el resultado :

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

¿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 Wilgenbroed 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](#)

Conexión Wifi

Hardware

Para poder utilizar las características IoT necesitamos conectar nuestra Micro:bit a Internet, pero la placa micro:bit **no tiene Wifi** para lograrlo se hace con la **placa IoT:bit**. En la placa tiene un **ESP32** (que es una evolución del **ESP8266**) , y podemos ver :

- El pin de recepción lo tiene en el Pin 8
- El pin de transmisión lo hace por el Pin 12

[2026-02-06 17_54_48-WhatsApp.png](#)

OJO TENEMOS QUE ALIMENTAR LA PLACA pues el ESP32 no se puede alimentar a través de micro:bit.

[2026-02-08 19_57_00-WhatsApp.png](#)

SOFTWARE

Para conectarlo a Internet usaremos una extensión en *Makecode IoT environment*:

[2026-02-06 18_06_21-Greenshot image editor.png](#)

Y se nos instalan nuevas instrucciones, nos vamos a fijar en las verdes ESP8266 :

[2026-02-06 18_07_52-Greenshot.png](#)

Y las utilizamos para poner en el bucle inicio :

1. Establecemos que el ESP32 tiene los pines de recepción y transmisión por los números definidos anteriormente
2. Ponemos el nombre de nuestra wifi y su contraseña (a veces los routers "capan" estas conexiones, por lo que recomendamos usar la conexión compartida, zona wifi o broadcast del móvil)
3. Si no hay conexión (pues no es instantáneo) entramos en un bucle con una cara triste
4. Si se sale del bucle anterior, es que estamos conectados y mostramos cara sonriente

[2026-02-06 17_57_43-Greenshot.png](#)

Y de paso preparamos la OLED que es muy útil para visualizar variables

2026-02-07 09_23_10-Greenshot.png

1. Definimos OLED 128x64 pixeles
2. Ponemos la OLED en blanco
3. Definimos la ESP32 con los pines de transmisión y recepción en P8 y P12
4. Ponemos la clave de la Wifi
5. Mientras no este conectado muestra cara triste
6. Sale del bucle, luego esta conectado, muestra cara contenta
7. Mensaje en la OLED que esta preparado



SMARTIO SPACE

SMARTIO SPACE

Que es SMARTIOT

Antes se llamaba KidsIoT y está adaptado a los alumnos, ahora se llama

<https://www.smartiot.space/>

2026-02-07 08_40_20-Greenshot.png

Vamos a ver uno ejemplo con la cantidad de luz

Podemos hacerlo con **cualquier tipo de sensor** ponemos estos dos ejemplos pero puede ser con la humedad, con el nivel de agua, con la distancia, con la temperatura, humedad, sonda, presencia.....

Crear cuenta

Creamos una cuenta, esto implica que los alumnos tienen que tener un email o que los alumnos utilicen la cuenta del profesor

2026-02-07 08_41_14-Greenshot.png

Crear un dispositivo y sus campos

Una vez logueados, **creamos un dispositivo nuevo**

2026-02-07 08_43_48-Greenshot.png

Rellenamos con el nombre que queramos poner al dispositivo y ponemos tantos campos como queramos, según las variables que queramos visualizar o controlar.

2026-02-07 08_45_04-C__Users_HP_Downloads_2026-02-07 08_27_38-.png - Greenshot imi

SMARTIO SPACE

Instrucciones en Makecode

Para conectarlo a Internet usaremos una extensión en Makecode IoT environment:

[2026-02-06 18_06_21-Greenshot image editor.png](#)

Y se nos instalan nuevas instrucciones :

[2026-02-07 09_30_53-Greenshot.png](#)

SMARTIO SPACE

Visualización de la cantidad de luz

Bucle INICIO

Ponemos las típicas instrucciones de **Inicio**

[2026-02-07 09_23_10-Greenshot.png](#)

1. Definimos OLED 128x64 pixeles
2. Ponemos la OLED en blanco
3. Definimos la ESP32 con los pines de transmisión y recepción en P8 y P12
4. Ponemos la clave de la Wifi
5. Mientras no este conectado muestra cara triste
6. Sale del bucle, luego esta conectado, muestra cara contenta
7. Mensaje en la OLED que esta preparado

Bucle por siempre..

Ponemos estas instrucciones

[2026-02-07 09_29_37-Greenshot.png](#)

1. El TOKEN que puedes ver en la ilustración de más adelante dónde está
2. Comprobamos que estamos conectados
3. Creamos una variable nueva LUZ y le asignamos la lectura del sensor LUZ
4. Determinamos que en SmartIoT el dato1 es la variable LUZ
5. Lo subimos a SmartIoT
6. Enseñamos también la variable LUZ en el OLED
7. Un check para visualizar que se ha subido
8. una pausa para no saturar al servidor
9. Una aspa en el caso de que no estemos conectados

Resultado en SMARTIOT

Entramos en la web logueados y vemos que muestra una gráfica según la luz que incide a la placa:



También puedes ver dónde está el TOKEN

[2026-02-07 08_27_38-.png](#)

EXPERIMENTA

Ahora cambia LUZ POR CUALQUIER OTRO SENSOR Y MIRA LOS CAMBIOS EN SMARTIOT



ThingSpeak

ThingSpeak

ThingSpeak: Cuenta, canal y API KEY

Creamos una cuenta en <https://thingspeak.mathworks.com/>

[image-1654161787395.01EZM1.png](#)

CREAR CANAL

Una vez que hemos creado la cuenta ThingSpeak le damos a crear canal

[2026-02-06 18_26_32-Greenshot.png](#)

Y rellenamos los campos, un nombre y que campos tendrá por ejemplo si queremos visualizar dos variables temperatura y humedad :

- Field 1 temperatura
- Field 2 humedad

Etc..

Y nos queda algo así

[image-1654161810157.ZPM0M1.png](#)

API KEY

Es importante este dato, está en la pestaña "API Keys".

- **Write API Key:** código identificativo para enviar los datos.

[image-1654161839609.png](#)

PUBLICO

Por último haz el canal público

[2026-02-06 18_41_26-Greenshot.png](#)

Vamos a ver unos ejemplos

- Temperatura y humedad con el DHT11
- Temperatura con la sonda DS18B20

Podemos hacerlo con **cualquier tipo de sensor** ponemos estos dos ejemplos pero puede ser con la humedad, con el nivel de agua, con la distancia, con la luz, presencia.....

ThingSpeak

Visualizar temperatura y humedad del DHT11 por thinkspeak

Realizamos un programa que:

1. Inicializamos y conectamos a la Wifi como hemos visto
2. Nos conectamos a Thingspeak Una pausa de 2 segundos
3. Si estamos conectados a Thingspeak pues procedemos a:
4. Leemos los datos de temperatura y humedad y lo almacenamos en unas variables nuevas llamadas TEMPERATURA y HUMEDAD
5. Hacemos una comprobación de que se cumplan las siguientes condiciones, que temperatura sea mayor de - 30 y la humedad mayor que 0. Esto es debido a que el sensor DHT11 frecuentemente da errores y los valores que devuelve son -999 y 0. Con esta comprobación nos aseguramos que los valores leídos son correctos
6. Grabo por el canal que he creado de Thinkspeak dos campos, el primero la temperatura, y el segundo la humedad
IMPORTANTE HAY QUE PONER EL API KEY DE ESCRITURA
7. Subo a Thinkspeak
8. muestro un check

2026-02-08 20_01_54-(8) WhatsApp — Mozilla Firefox.png

<https://makecode.microbit.org/S21198-38927-33091-18971>



<https://makecode.microbit.org/#pub:S21198-38927-33091-18971>

el resultado lo puedes ver en <https://thingspeak.mathworks.com/channels/1700342>

2026-02-06 17_40_01-Greenshot.png

ThingSpeak

SONDA DS18B20

Vamos a ver otro ejemplo, en este caso lo visualiza por el OLED y lo envía a Thingspeak para ver la evolución, vamos a poner una taza de agua caliente, la sonda conectada al Pin 1 :

[2026-02-06 19_51_58-WhatsApp.png](#)

El programa:

1. Inicializamos el OLED
2. Inicializamos y conectamos la Wifi
3. Si hay error en la sonda que lo visualice
4. Conectamos a Thingspeak
5. Si estamos conectados, almacenamos en una variable nueva TEMPERATURA la lectura de la sonda DS18B20
6. lo asociamos al campo 1 de nuestro canal
7. lo subimos al servidor
8. lo visualizamos en el oled y mostrar un check

[2026-02-06 19_52_25-WhatsApp.png](#)

Proyecto <https://makecode.microbit.org/S52558-17116-07305-38404>



<https://makecode.microbit.org/#pub:S52558-17116-07305-38404>

ThingSpeak

POR QUÉ SE UTILIZA UNA TAZA PARA DESAYUNAR

VELOCIDAD DE TRANSFERENCIA DE CALOR

Igual que velocidad es espacio partido por tiempo. La velocidad de transferencia de calor es

$$v = Q / t$$

- **Q** es el calor transmitido en calorías
- **v** la velocidad de transmisión de calor en cal/seg
- **t** es el tiempo en segundos

Si un material tiene una velocidad de transmisión de calor muy alta, (por ejemplo los metales) es mal aislante y si es muy baja es que es buen aislante del calor.

El calor se puede medir con la fórmula

$$Q = m \text{ cesp } (T_f - T_o)$$

- **m** masa en gr que en el caso del agua pondremos la mitad de una taza o un vaso, luego aproximadamente son 100ml que son aproximadamente 100 gr
- **T_f** la temperatura final en °C
- **T_o** la temperatura inicial en °C
- **cesp** calor específico que en el caso del agua es 1 cal/gr °C

PONEMOS UNA TAZA

A la mitad de agua, bien caliente, y con la sonda DSB18 y Thingspeak y medimos :

[2026-02-06 19_51_58-WhatsApp.png](#)

El resultado fue:

[2026-02-06 20_12_01-WhatsApp.png](#)

Podemos también exportar como CSV los datos



[2026-02-06 20_19_41-Greenshot.png](#)

y en un fichero excell importarlos como texto, delimitado por coma y espacios

[2026-02-06 20_20_54-Libro1 - Excel.png](#)

Y obtenemos las medidas:

[2026-02-06 21_15_59-Libro1 - Excel.png](#)

16 minutos con 32 segundos son 992 segundos

Calculamos el calor

$$Q = m \text{ cesp } (T_f - T_o) = 100 * 1 * 22.4616 = 2246.16 \text{ calorías perdidas}$$

Calculamos la velocidad de transferencia de calor

$$v = Q / t = 2246.16 / 992 = 2.26 \text{ cal/seg}$$

PONEMOS UN VASO DE CRISTAL

La misma cantidad de agua (la vertemos en un vaso), bien caliente, y con la sonda DSB18 y Thingspeak y medimos :

[2026-02-06 20_32_21-WhatsApp.png](#)

y hacemos igual, el resultado es:

[2026-02-06 20_33_07-Greenshot.png](#)

[2026-02-06 21_17_34-Libro1 - Excel.png](#)

15 minutos con 37 segundos son 937 segundos

Calculamos el calor

$$Q = m \text{ cesp } (T_f - T_o) = 100 * 1 * 31.9375 = 3193.75 \text{ calorías perdidas}$$

Calculamos la velocidad de transferencia de calor

$$v = Q / t = 3193.75 / 937 = 3.41 \text{ cal/seg}$$

CONCLUSIÓN



Pues que si utilizamos un vaso, perdemos 1.14 calorías cada segundo que si utilizáramos la taza. Por eso desayunamos con taza y no con vaso de cristal, para que se mantenga los líquidos calientes.

2026-02-06 21_21_17-Libro1 - Excel.png

AGRADECIMIENTOS a José Francisco Muñoz www.almeribot.com que la idea de este experimento es suya. Libro IoT con micro:bit

Muro

Pon aquí ejemplos tuyos o de los demás <https://padlet.com/CATEDU/smart-agriculture-kit-para-micro-bit-15lbmbamv9txumnd>

<https://padlet.com/embed/15lbmbamv9txumnd>

Made with **Hecho con Padlet**

Créditos

Autores:

- **Javier Quintana Peiró** CATEDU 2026

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](#)