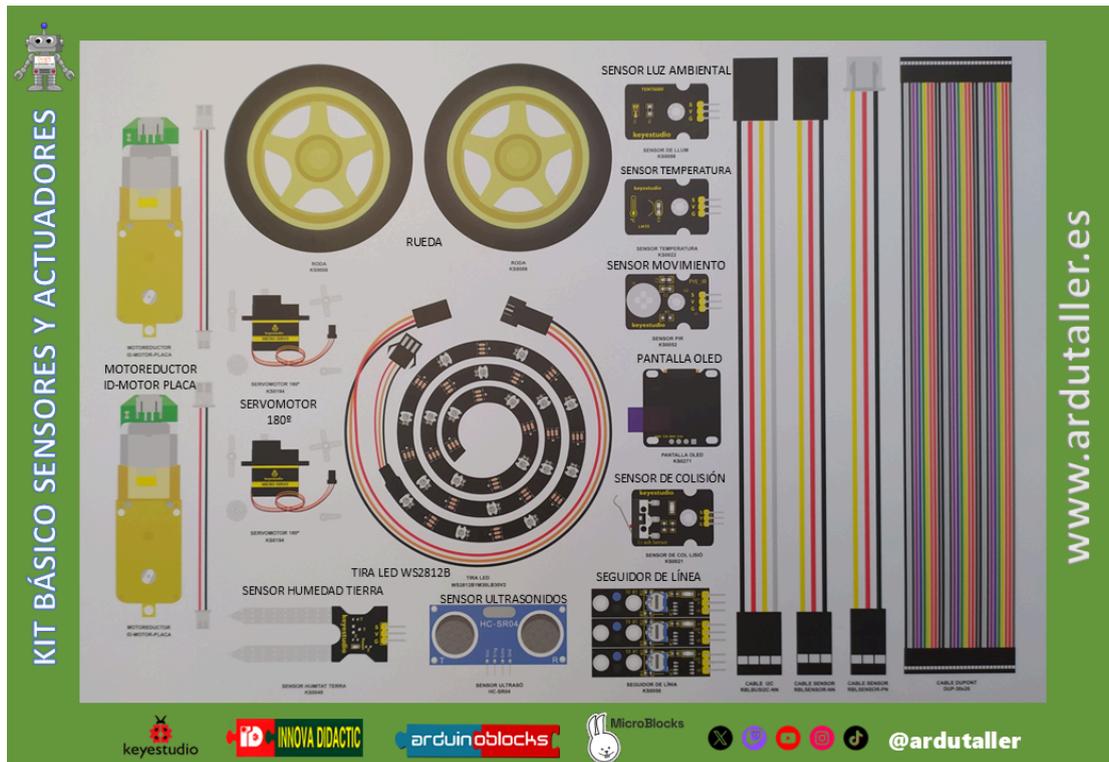




Este manual está optimizado para visualizarlo impreso o en pdf. Si estás accediendo desde un móvil o tablet, te recomendamos actives en el navegador la opción “visualizar como sitio para ordenador”

KIT BÁSICO SENSORES Y ACTUADORES



Editado por www.franciscojose.es y www.ardutaller.es
en colaboración con InnovaDidactic.

Este manual está en constante actualización

Si quieres este documento en .pdf puedes descargarlo [aquí](#). Es posible que no esté tan actualizado como el documento online.

Si ahora mismo estás usando una versión en pdf o impresa, te recomendamos que visites la versión [online](#) o escanees el código QR, para tener siempre disponible el documento actualizado. Este manual ha sido desarrollado por ardutaller, de manera desinteresada. Innovadidactic le ha proporcionado los materiales para hacer las pruebas pertinentes.



Programación realizada con:  

Basado en kit distribuido por: 

KIT BÁSICO SENSORES Y ACTUADORES

1 INTRODUCCIÓN

1.1 PRESENTACIÓN INTERACTIVA

1.2 HARDWARE

2. SOFTWARE DE PROGRAMACIÓN

2.1 Programación en arduinoblocks

2.2 Programación con MicroBlocks

3. ELEMENTOS DEL KIT BÁSICO

3.1 Moto-reductor + placa

3.2 Servomotor KS0194 180°

3.3 Rueda KS9008

3.4 Sensor humedad tierra KS0049

3.5 Tira de Led WS2812B1M30LB30V2

3.6 Sensor ultrasonidos HC-SR04

3.7 Sensor luz ambiental KS0098

3.8 Sensor temperatura KS0022

3.9 Sensor presencia PIR KS0052

3.10 Pantalla Oled KS0271

3.10.1 Mostrar mensajes en pantalla OLED

3.10.2 Mostrar imágenes en pantalla OLED

3.11 Sensor de colisión o final de carrera KS0021

3.12 Seguidor de línea KS0050

3.13 Cables I2C

3.14 Cables sensores

3.15 Cables Dupont

4. Conexiones IoT

4.1 Conexión a una red WIFI

4.1.1 Mostrar Fecha y Hora

4.2.1 Control con APP

1 INTRODUCCIÓN

El objetivo del presente documento, es presentar una aplicación desarrollada por InnovaDidactic en colaboración con Keyestudio, y que surge a raíz la necesidad de implantar en los institutos de Cataluña, un programa de aprendizaje centrándose en el uso de sensores y actuadores, y que puedan ser utilizados con cualquier microcontrolador. Así por ejemplo los elementos incluidos en esta maleta pueden ser utilizados desde los más básicos Arduino Uno, hasta la novedosa ESP32 micro:STEAMakers, pasando por otros como pueden ser la micro:Bit de o la ESP32 STEAMakers.

Este kit, no es el primero que se lanza con un objetivo similar, y seguro que habrás visto otros kits como los famosos [kits 37 en 1](#), que también han sido distribuidos por Innovadidactic y fabricados por keyestudio. Estos kits consistían en una caja de plástico en los que se almacenaban los sensores y actuadores.

El producto que se presenta en este manual, no consta de una caja que simplemente sirva para almacenar los componentes, sino que se trata de una maleta con varios huecos o departamentos, en los que van alojados los elementos.

A nivel organizativo, dentro de una clase de primaria o secundaria, esto aporta un valor añadido, pues con un simple vistazo, podemos ver si están todos los componentes, y si faltase alguno, podremos detectarlo fácilmente pues en la tapa de la maleta, tenemos una ilustración que nos indica dónde debe ir ubicado cada sensor y cada actuador.

El que estamos desarrollando ahora, es el kit básico, y en las próximas páginas, vamos a explicar uno por uno todos los elementos, así como el modo de uso de los mismos. En otro documento, analizaremos el kit avanzado.

A lo largo del desarrollo del presente manual, que como viene siendo habitual en el material desarrollado por ardutaller, será un documento vivo y constantemente actualizado a través de la versión del documento online, mostraremos no sólo como conectar los dispositivos a la placa, sino que también haremos pequeñas prácticas de programación con los programas de referencia de nuestra web, arduinoblocks o microblocks.

1.1 PRESENTACIÓN INTERACTIVA

Se ha habilitado en la plataforma genially, una presentación interactiva en la que mostramos todos y cada uno de los componentes, de manera que se pueda utilizar en las clases de primaria o secundaria, donde se utilicen los componentes de la maleta.

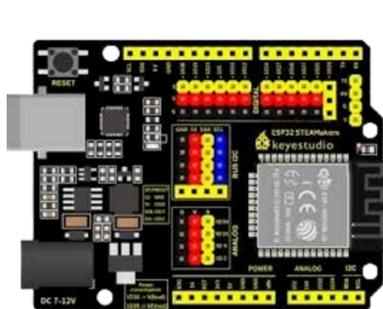


La presentación está disponible [aquí](#), y pulsando sobre cada uno de los elementos, podremos ampliar la información sobre el mismo.

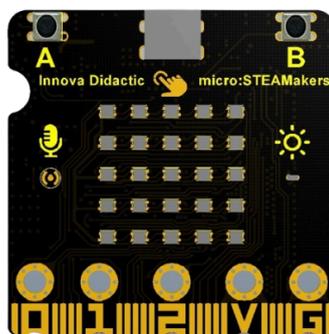
1.2 HARDWARE

Para la realización de las diferentes prácticas y pruebas de los sensores y actuadores, utilizaremos preferentemente las placas [ESP32STEAMakers](#) y [ESP32 micro:STEAMakers](#), ambas placas distribuidas por Innovadidactic. Para el caso de esta última, utilizaremos además la placa [shield:Bit](#), que nos permite añadir conexiones para conectar sensores y actuadores. Esta shield, es además compatible con micro:bit

Aunque nos centraremos en trabajar con las placas antes descritas, los componentes son totalmente compatibles con placas como arduino o micro:bit. En cuanto a la programación, la mayoría de bloques son perfectamente utilizables para la programación del resto de placas



ESP32 STEAMakers



micro:STEAMAKERS



Shiel:Bit

2. SOFTWARE DE PROGRAMACIÓN

2.1 Programación en **arduinoblocks**

Todos los programas que hagamos en este manual, se van a hacer dentro de la plataforma de programación educativa online, creada por [Juanjo López](http://Juanjo_López), www.arduinoblocks.com. Si ya has trabajado con esta plataforma, puedes pasar a la siguiente sección. Si por el contrario eres nuevo en esta plataforma, te dejamos algunos enlaces de nuestro canal que te pueden ayudar:

[PRIMEROS PASOS CON ARDUINOBLOCKS](#)

[REGISTRO EN ARDUINOBLOCKS Y PRIMER PROYECTO](#)

[GESTIÓN DE PROYECTOS EN ARDUINOBLOCKS](#)

Estos videos están basados en placas como Arduino Uno o Esp32 STEAMakers, pero los conceptos son aplicables a todas las placas compatibles con la plataforma.

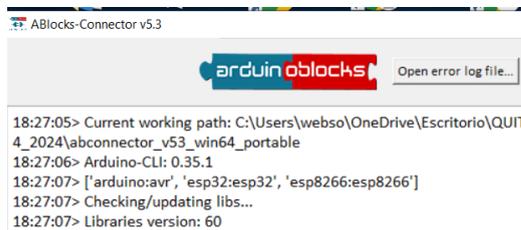
A lo largo del presente manual analizaremos la mayoría de los bloques disponibles en arduinoblocks, para esta nueva placa. Con el fin de no extendernos mucho, algunos bloques los veremos sólo a nivel informativo, profundizando sólo en aquellos que consideremos relevantes para la aplicación.

Para ampliar la información sobre los módulos, se recomienda visitar el Free Book Online editado por Juanjo López y que está disponible en este [enlace](#). Al margen de esto, en las diferentes prácticas, enlazaremos a las secciones correspondientes a cada bloque en dicho manual.

Para la programación de los sensores y actuadores en [arduinoblocks](#), trabajaremos sobre un proyecto en el iremos incluyendo ejemplos de los sensores y actuadores. El proyecto sobre el que trabajaremos es [éste](#)

Cuestiones a tener en cuenta en todos los proyectos:

Seleccionar correctamente la placa a la hora de crear el proyecto. En el desplegable que aparece donde definimos el proyecto, nos aparecen todas las placas compatibles. En el caso de nuestra placa sería →Tipo de proyecto: ESP32 micro:STEAMakers



Tener abierto el programa [AB-Connector](#), que será el software que nos permita pasar el programa desde nuestro navegador a la placa. Cuando tengamos conectada la placa a nuestro

ordenador, el sistema operativo, mediante el uso de este software, nos conectará la placa a través de un puerto COM. Es muy importante saber el puerto COM asignado. Este [video](#) se explica para el caso de windows.

Una vez sepamos qué puerto ha asignado el sistema operativo la placa, ya sólo nos queda elegir dicho puerto en arduinoblocks, cuando vayamos a transferir el programa. El AB-connector, debe estar siempre abierto, y funcionará en segundo plano. Sin él, será imposible la conexión de arduinoblocks con la placa.



Al [AB-Connector](#), está disponible para diferentes sistemas operativos, y cada vez que se ejecuta, se conecta a los servidores

de arduinoblocks, con el fin de actualizar todas las librerías del mismo.

Cabe indicar además, que en el momento de redacción de este documento, no están disponibles de manera oficial las bibliotecas de la shield:bit para arduinoblocks, por lo tanto, o bien usaremos las librerías existentes, o haremos la programación sin hacer uso de esa shield.

2.2 Programación con MicroBlocks

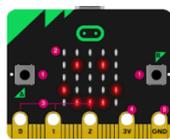
El acceso a la plataforma lo haremos a través de su web oficial <https://microblocks.fun> o descargando el software en el ordenador. Puedes encontrar información sobre esta plataforma en nuestra web.

Una de las ventajas que aporta MicroBlocks, es que permite que los cambios en el programa aparezcan de manera directa en la placa sin necesidad de estar constantemente transfiriendo el programa cada vez que hagamos una modificación. Esto es posible, gracias a un firmware que se instala en la placa, y que genera una máquina virtual, que permite la conexión en tiempo real con el PC. Además en las últimas versiones de MicroBlocks se ha implantado la posibilidad de programa placa sin cable, haciendo uso de la tecnología bluetooth (BLE).

Todas las versiones de microblocks, oficial, pilot e innovadidactic, están accesibles desde nuestra web. A lo largo del manual trabajaremos principalmente con la versión oficial, aunque es posible, que debido a la falta de alguna función en esa versión, utilizaremos las otras. Se indicará en la práctica

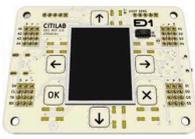


Estas son algunas de las placas compatibles con microblocks.



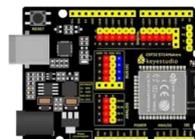
micro:bit

Programación placa desarrollada por la BBC



ED1

Programación Placa desarrollada por Citilab



STEAMakers

Programación placa diseñada por InnovaDidactic



micro:STEAMakers

Programación placa diseñada por InnovaDidactic

3. ELEMENTOS DEL KIT BÁSICO

A continuación procedemos a la descripción de todos los componentes incluidos en la maleta. En cada uno de los apartados, incluiremos accesos a la descripción oficial del fabricante, enlace a la web del distribuidor oficial así como cualquier otra información relevante. También se mostrarán ejemplos de programas en arduinoblocks y MicroBlocks. Si en algunos componentes, no está disponible algún enlace, es posible que no exista, o dicho apartado esté aún en proceso de redacción.

Para la programación de los sensores y actuadores en arduinoblocks, trabajaremos sobre un proyecto en el iremos incluyendo ejemplos de los sensores y actuadores. El proyecto sobre el que trabajaremos es [éste](#)

3.1 Moto-reductor + placa

El acceso a la página oficial del fabricante está disponible [aquí](#).

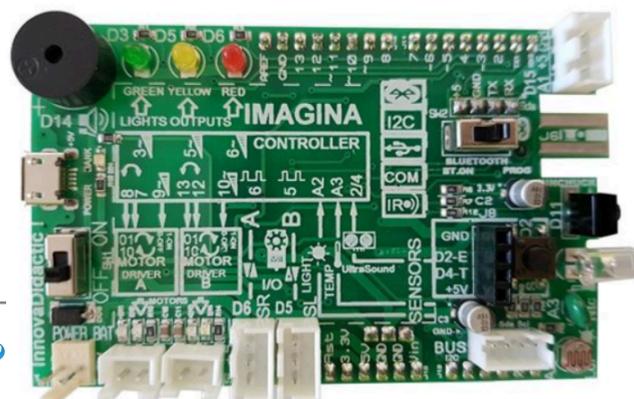
Este producto puede adquirirse de manera individual en la web de Innovadidactic [aquí](#).



Motoreductor Vigor BO1 de un eje (1:120, 90rpm), con placa y cable de conexión. Ref: BO1_120_placa

Marca: Vigor

- Reductora : 1:120
- Voltaje : 3.0~12.0V DC
- Sin carga : 40~180mA, 20~90 rpm
- Par : 0.8~5Kgf.cm



- Diámetro eje: 5.2mm
- Peso : 30.0 g
- Color: amarillo. A 6V gira aprox. a 55rpm.

La conexión del motor a la placa, en ningún caso se hará de manera directa, a menos que la placa especifique que puede conectarse. Para ello, utilizaremos shields específicas con los integrados correspondientes para el control del motor. Así por ejemplo podemos trabajar con la shield , [3dbot](#), con conexiones para placas compatibles con UNO, y que incluye drivers para dos motores o con la nueva placa desarrollada [shield-bit](#) que está preparada para conectar placas como micro:bit o micro:STEAMakers.

Los motores que se incluyen en el kit, a diferencia de los de otros fabricantes, traen incorporados una pequeña placa que permite conectar el motor a la placa donde se encuentra el driver que controla el microcontrolador. La conexión se hace a través de un cable que viene incluido.



Programación con [arduinoblocks](#)

Programación con [MicroBlocks](#)

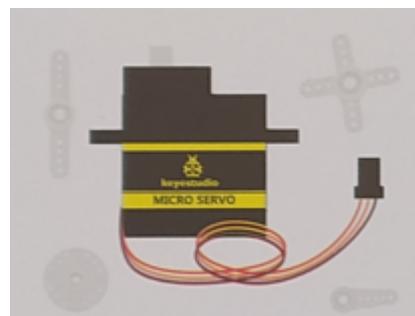
3.2 Servomotor KS0194 180°

El acceso a la página oficial del fabricante está disponible [aquí](#).

Este producto puede adquirirse de manera individual en la web de Innovadidactic [aquí](#).

El servomotor es un actuador rotativo de control de posición. Se compone principalmente de carcasa, placa de circuito, motor sin núcleo, engranaje y sensor de posición.

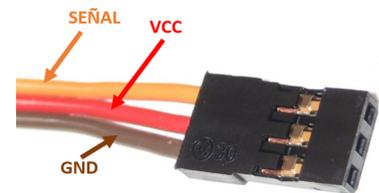
El microcontrolador envía una señal al servomotor. El motor tiene un circuito de referencia incorporado que emite una señal de referencia, un ciclo de 20 ms y un ancho de 1,5 ms. El motor compara la



tensión de polarización de CC adquirida con la tensión del potenciómetro y produce una diferencia de voltaje. El IC en la placa de circuito decidirá la dirección de rotación en consecuencia y conducirá el motor sin núcleo. El engranaje luego pasa la fuerza al eje. El sensor determinará si ha alcanzado la posición ordenada de acuerdo con la señal de realimentación.

Los servomotores se usan en sistemas de control que requieren tener y mantener diferentes ángulos. Cuando la velocidad del motor es definitiva, el engranaje hará que el potenciómetro gire. Cuando la diferencia de voltaje se reduce a cero, el motor se detiene. Normalmente, el rango del ángulo de rotación está entre 0-180 grados.

Los servomotores vienen con muchas especificaciones. Pero todos ellos tienen tres cables de conexión, que se distinguen por colores marrón, rojo, naranja (las diferentes marcas pueden tener un color diferente). Uno marrón es para GND, rojo uno para poder positivo, naranja para línea de señal.



El ángulo de rotación del servomotor se controla regulando el ciclo de trabajo de la señal [PWM \(Pulse-Width Modulation\)](#). El ciclo estándar de la señal PWM es 20 ms (50 Hz). Teóricamente, el ancho se distribuye entre 1ms-2ms, pero de hecho, está entre 0.5ms-2.5ms. El ancho corresponde al ángulo de rotación de 0 ° a 180 °. Pero tenga en cuenta que para un motor de marca diferente, la misma señal puede tener diferentes ángulos de rotación.

Programación con [arduinoblocks](#)

Programación con [MicroBlocks](#)

3.3 Rueda KS9008

El acceso a la página oficial del fabricante está disponible [aquí](#).

Este producto puede adquirirse de manera individual en la web de Innovadidactic [aquí](#).



Las ruedas tienen estas características técnicas:

Diametro 68mm Aproximadamente
Ancho 25mm Aproximadamente

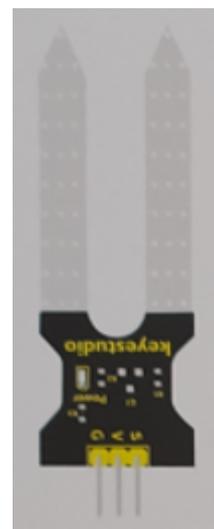
Estas ruedas irán colocadas en los motores vistos anteriormente, y por tanto la programación se hará de los mismos.

3.4 Sensor humedad tierra KS0049

El acceso a la página oficial del fabricante está disponible [aquí](#).

Este producto puede adquirirse de manera individual en la web de Innovadidactic [aquí](#).

Este es un sensor simple de humedad del suelo que tiene como objetivo detectar la humedad del suelo. Si el suelo no tiene agua, el valor analógico emitido por el sensor disminuirá, de lo contrario, aumentará. Si usa este sensor para hacer un dispositivo de riego automático, puede detectar si su botánica tiene sed para evitar que se marchite cuando salga. Con su uso, podemos conseguir un jardín inteligente.



El sensor se configura con dos sondas insertadas en el suelo, cuando la corriente atraviesa el suelo, el sensor obtiene un valor de resistencia al leer los cambios actuales entre las dos sondas y convertir dicho valor de resistencia en contenido de humedad. Cuanto mayor sea la humedad (menos resistencia), mayor será la conductividad del suelo.

La superficie del sensor se ha sometido a un proceso de metalización para prolongar su vida útil. Con la ayuda de este sensor, la planta puede decirte: ¡necesito agua!

Programación con [arduinoblocks](#)

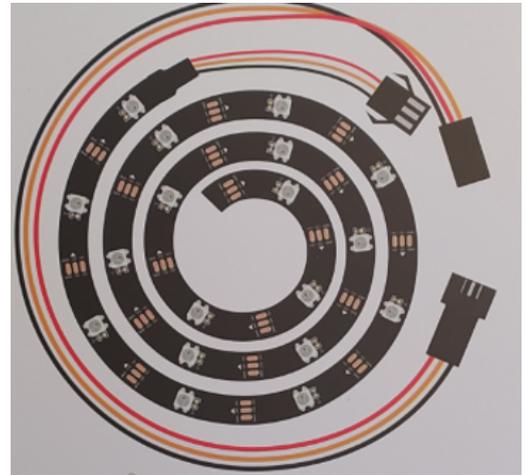
Programación con [MicroBlocks](#)

3.5 Tira de Led WS2812B1M30LB30V2

El acceso a la página oficial del fabricante está disponible aquí.

Este producto puede adquirirse de manera individual en la web de Innovadidactic [aquí](#).

Para una análisis mas completo de esta tira de led, también conocida como Neopixel, pues así la denominó en su día el fabricante Adafruit, podemos decir que se trata de “LEDs RGB direccionables individualmente”. Es decir, mediante programación podemos decir que leds que queremos que se encienda, y con que color RGB queremos que luzca. Es muy importante a la hora de conectar la tira, la corriente siga el sentido marcado por las flechas que aparecen en la tira de leds. Si quieres más información sobre el uso de tiras de leds, te invitamos a ver el [video](#) o consultar este [manual online](#).



La tira que se incluye en el kit, no es muy grande, por lo que podremos alimentarla directamente desde el microcontrolador. Para tiras de mayor longitud, habría que utilizar alguna fuente alimentación externa. Se puede encontrar mas información en la siguiente [dirección](#) o en el [Github de Fede Coca](#).

Programación con [arduinoblocks](#)

Programación con [MicroBlocks](#)

3.6 Sensor ultrasonidos HC-SR04

El acceso a la página oficial del fabricante está disponible aquí.



Este producto puede adquirirse de manera individual en la web de Innovadidactic [aquí](#).

Sensor de Ultrasonidos HC-SR04, medidor de distancias a través de ultrasonidos *HC-SR04*.

Voltaje: 5 VDC

Corriente en reposo: < 2 mA

Salida: High 5V/Low 0V

Angulo: < 15°

Distancia: 2 a 450 cm

Resolución: 0,3 cm

Para poder trabajar con este sensor, necesitaremos usar además de la tensión de alimentación (VCC y GND), dos señales, una que generará un pulso (TRIG) y que será la señal que evitar el ultrasonidos, y otra señal (ECHO), que será la que nos diga, en función del tiempo que tarde en volver la señal, la distancia medida.

Programación con [arduinoblocks](#)

Programación con [MicroBlocks](#)

3.7 Sensor luz ambiental KS0098

El acceso a la página oficial del fabricante está disponible [aquí](#).

Este producto puede adquirirse de manera individual en la web de Innovadidactic [aquí](#).



Este dispositivo, a través del fotoresistor incluido y la circuitería que lo acompaña, permite de manera fácil ver la luminosidad existente. Lo que nos proporciona es una señal analógica.

Voltaje de suministro: + 5VDC 50mA

Tamaño: 36.5 * 16mm

Peso: 4g

Programación con arduinoblocks

Programación con MicroBlocks

3.8 Sensor temperatura KS0022

El acceso a la página oficial del fabricante está disponible [aquí](#).

Este producto puede adquirirse de manera individual en la web de Innovadidactic [aquí](#).

El sensor de temperatura se basa en el sensor de temperatura semiconductor LM35. Se puede usar para detectar la temperatura del aire ambiente. Este sensor ofrece un rango funcional entre 0 grados Celsius y 100 grados Celsius. La sensibilidad es de 10 mV por grado Celsius. El voltaje de salida es proporcional a la temperatura.



Especificación

Basado en el sensor de temperatura semiconductor LM35

Se puede utilizar para detectar la temperatura del aire ambiente

Sensibilidad: 10mV por grado Celsius

Rango funcional: 0 grados Celsius a 100 grados Celsius

Tamaño: 30 * 20mm

Peso: 3g

Programación con arduinoblocks

Programación con MicroBlocks

3.9 Sensor presencia PIR KS0052

El acceso a la página oficial del fabricante está disponible [aquí](#).

Este producto puede adquirirse de manera individual en la web de Innovadidactic [aquí](#).

El sensor de movimiento infrarrojo piroeléctrico puede detectar señales infrarrojas de una persona en movimiento o animal en movimiento, y la salida cambia de señal. Se puede aplicar a una variedad de ocasiones para detectar el movimiento del cuerpo humano. Los sensores infrarrojos piroeléctricos convencionales requieren un detector infrarrojo piroeléctrico del cuerpo, un chip profesional, un circuito periférico complejo, por lo que el tamaño es más grande, con circuitos complejos y menor confiabilidad. El sensor que incluye este sensor de movimiento infrarrojo piroeléctrico, está especialmente diseñado para el uso en placas basadas en Arduino o ESP32. Utiliza un sensor infrarrojo piroeléctrico de cuerpo digital integrado, tiene un tamaño más pequeño, una mayor fiabilidad, menor consumo de energía y un circuito periférico más simple.

Voltaje de entrada: 3.0-3.3V ~ 5V, 6V

Corriente máxima de trabajo: 100mA

Temperatura de trabajo: -20 ~ 85 °C

Voltaje de salida: Alto 3V, bajo 0V

Tiempo de retardo de salida (nivel alto): Aproximadamente 2.3 a 3 segundos

Ángulo de detección: menos de 80 ° en dirección horizontal y 55° en dirección vertical

Distancia de detección:

Indicador LED de salida de 3-5 metros (Cuando sale ALTO, estará ENCENDIDO)

Corriente límite del pin: 100mA

Tamaño: 30 * 20mm

Peso: 4g

Programación con [arduinoblocks](#)

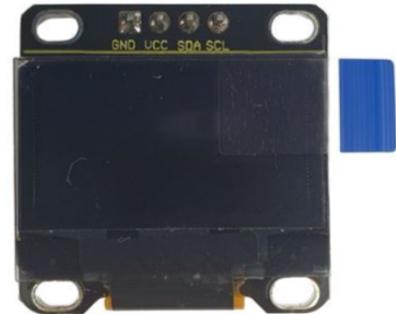
Programación con [MicroBlocks](#)

3.10 Pantalla Oled KS0271

El acceso a la página oficial del fabricante está disponible [aquí](#).

Este producto puede adquirirse de manera individual en la web de Innovadidactic [aquí](#).

OLED es la abreviatura de diodo emisor de luz orgánico. En el nivel microscópico, una pantalla OLED es una matriz de LED orgánicos que se iluminan cuando emiten energía. La tecnología antigua de LCD (pantalla de cristal líquido) utiliza polarizadores controlados electrónicamente para cambiar la forma en que la luz pasa o no pasa a través de ellos. Esto requiere una luz de fondo externa que ilumine toda la pantalla debajo. Esto consume mucha energía porque en el momento en que la pantalla está encendida, se debe proporcionar suficiente luz para todos los píxeles. La nueva tecnología OLED solo usa electricidad por píxel. Debido a que cada píxel crea su propia luz, solo los píxeles que están encendidos usan electricidad. Esto hace que la tecnología OLED sea muy eficiente. Además, la forma en que se construyen estos tipos de OLED permite que sean muy delgadas en comparación con la pantalla LCD.



Esta pantalla tiene la peculiaridad que se conecta utilizando la tecnología I2C, que permite la conexión de múltiples dispositivos haciendo uso de un bus compuesto por los dos cables de alimentación (VCC y GND) y los cables de datos (SCL y SDA).

Voltaje de suministro: 3.3V a 5V
 Número de píxeles: 128 × 64
 Profundidad del color: azul
 Modo de comunicación: IIC

Puedes encontrar mas información sobre el uso de este tipo de pantallas en el siguiente [video](#).

Programación con arduinoblocks

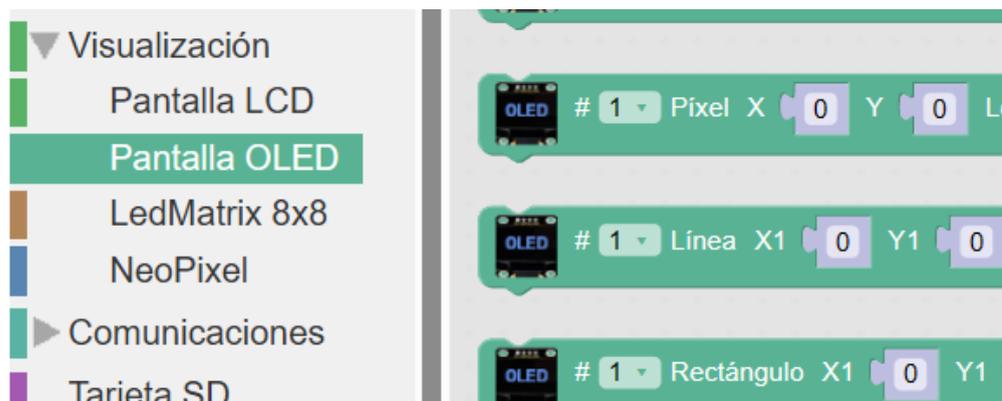
3.10.1 Mostrar mensajes en pantalla OLED

En el ejemplo anterior del sensor de temperatura y humedad, ya hemos dado algunas pinceladas para explicar cómo introducir los textos para ser mostrados en la pantalla OLED.

Para este ejemplo, volveremos al ejemplo que vimos anteriormente en arduinoblocks, y trabajaremos con una función. Es muy importante, en el bloque “inicializar”, indicar la dirección en la que está conectada la pantalla.

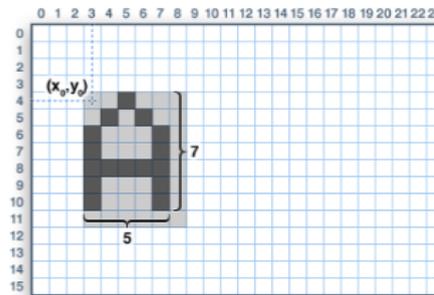
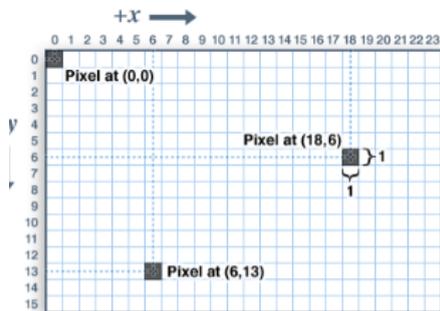


A continuación, buscamos los bloques correspondientes a la pantalla OLED:



En este apartado, tenemos muchos bloques nos permitirá poder definir el mensaje a mostrar en la pantalla.

Los bloques son muy intuitivos, sólo tendríamos que hacer alguna puntualización a la hora de definir las coordenadas donde se ubica el primer elemento a mostrar.



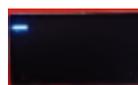
La coordenada X corresponde a la dirección en la horizontal y la Y corresponde a la dirección vertical. En decir, haciendo analogía con una hoja de cálculo, la X sería la columna y la Y la fila.

Según podemos ver en la imagen anterior, el punto definido estaría en la Columna (x) 18 y en la fila (y) 6.

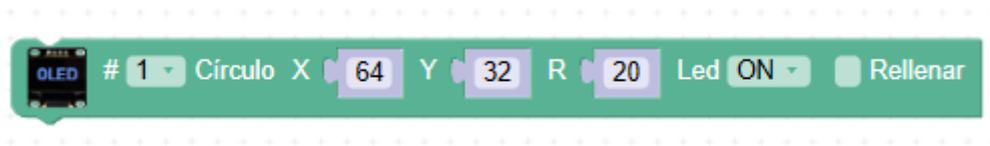
Así por ejemplo, si lo que queremos es encender ese pixel, el bloque que a utilizar sería el siguiente:



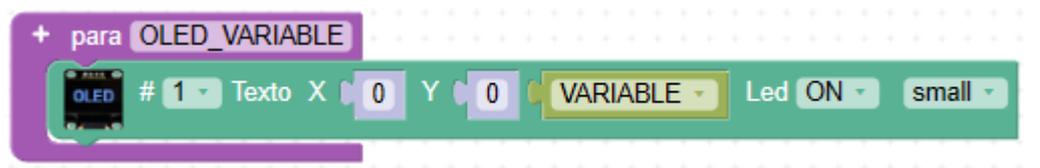
Si por ejemplo queremos dibujar una línea que empiece en el pixel (2,10), y que tenga una longitud de 10 pixels, este sería el bloque. Tendríamos que decirle que la línea va desde el pixel (2,10) hasta el pixel (12,10)



Si lo que queremos mostrar es un círculo, tendremos que indicarle cual es el centro del mismo y el radio:



Si lo que queremos mostrar es por ejemplo el valor de una variable será tan simple como utilizar el siguiente bloque, y dentro del apartado correspondiente, introducir dicha variable



Siguiendo la misma línea, podemos ir definiendo diferentes elementos en la pantalla.

A continuación vamos a ver el bloque que quizás sea mas interesante, sobre todo debido a que vamos a necesitar de varios elementos para su definición.

3.10.2 Mostrar imágenes en pantalla OLED

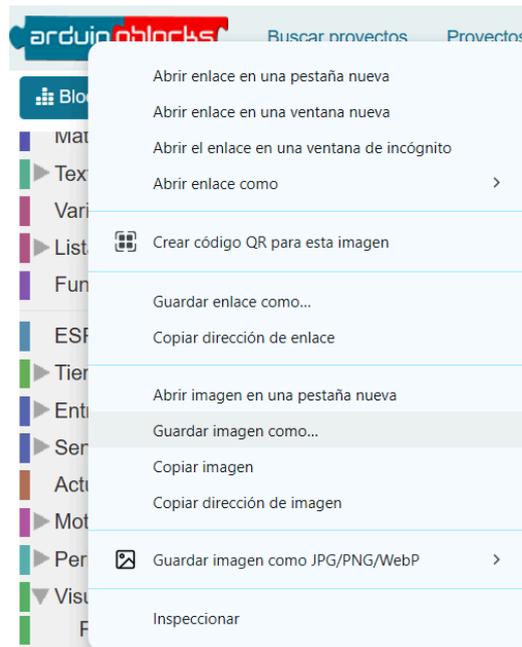
Trabajaremos con este bloque, en el que mediante una secuencia de texto, le indicaremos que pixeles son los que tendremos que encender para crear la imagen.

Podemos pensar que esto va a ser una tarea difícil, pero gracias a la función que tienen implementada arduinoblocks, será muy sencillo.

Lo primero que vamos a necesitar es una imagen en blanco y negro que podamos introducir en el programa. Si trabajamos con windows, podemos utilizar el programa Paint.

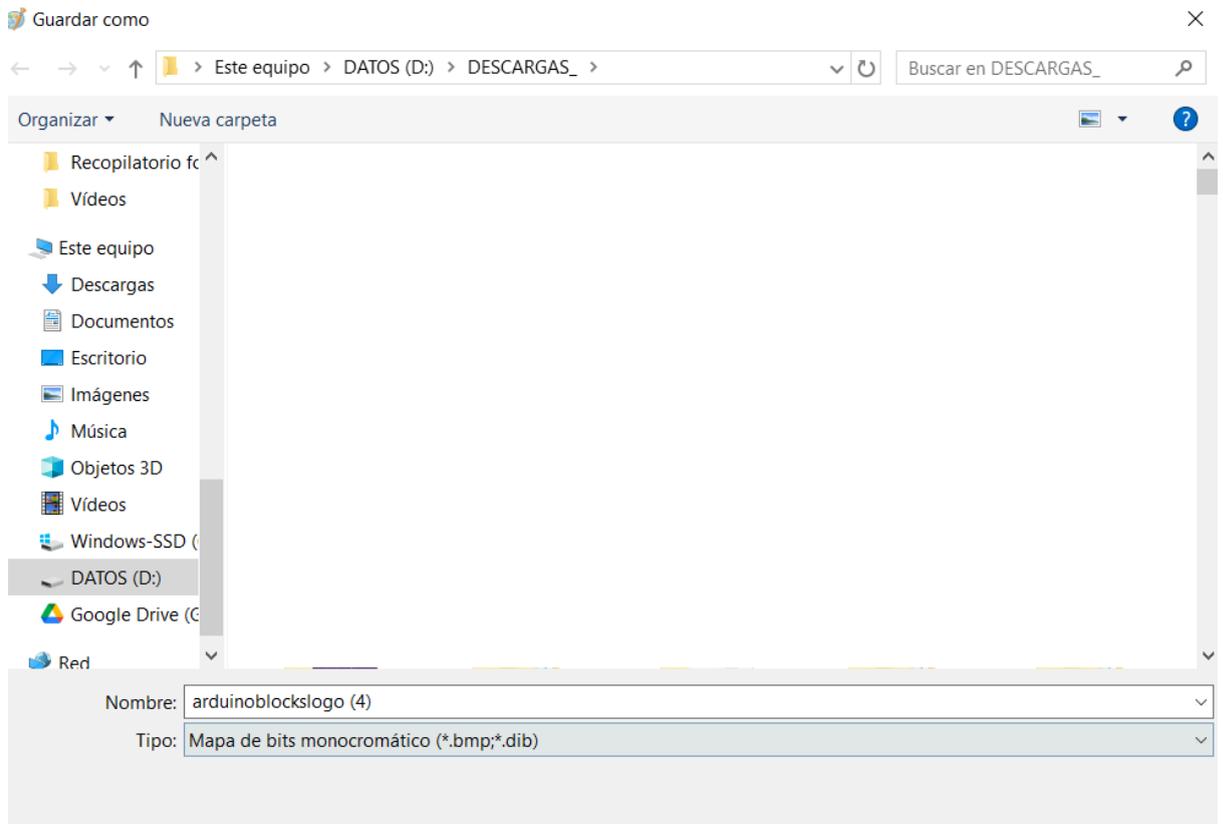
En estos caso vamos a coger el logo de arduinoblocks, y lo vamos a poner en la pantalla.

Accedemos a la pantalla principal y guardamos la imagen.

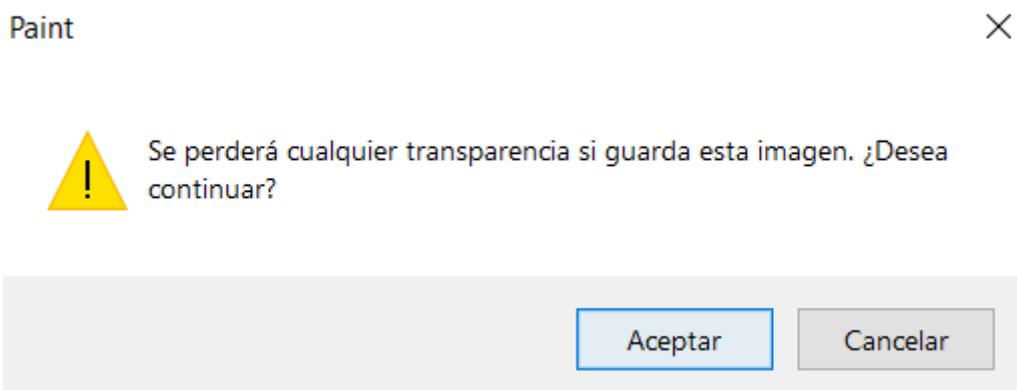


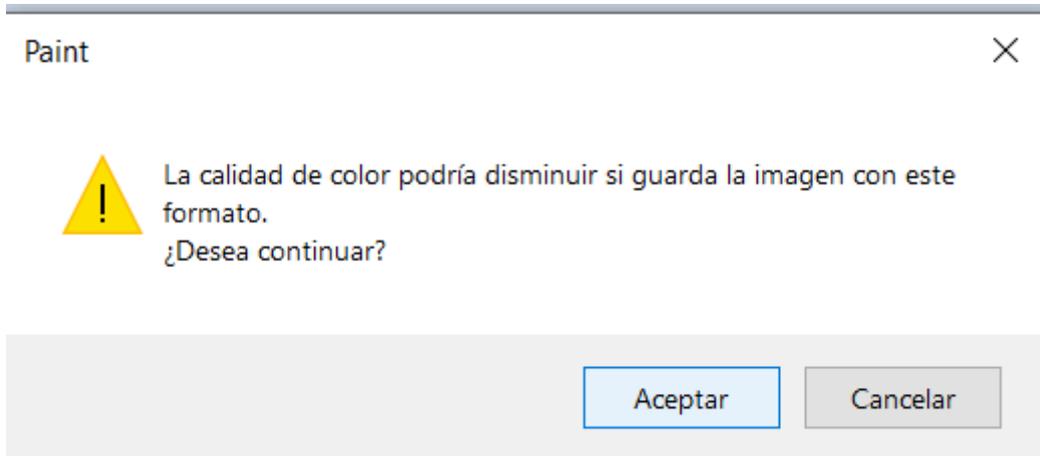
Guardamos la imagen en un lugar donde luego podamos rescatarla, y procedemos a abrir la imagen en el Paint de Windows.

Una vez abierta la imagen la guardamos en formato .bmp monocromático.

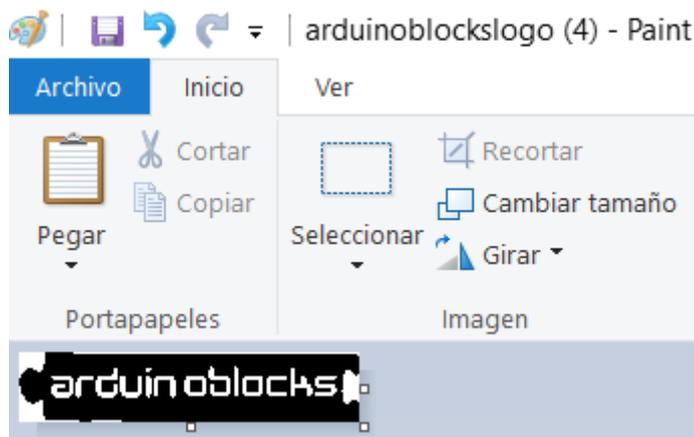


Si nos sales esta advertencia, las aceptamos.





La imagen quedaría de esta forma. La guardamos y en el próximo paso, tendremos que abrirla.



Ahora ya podemos volver a arduinoblocks, y buscamos la siguiente función, que será la que nos ayude a generar el código de la imagen, con los pixeles que vamos a encender.

Clear: apaga todos los píxeles

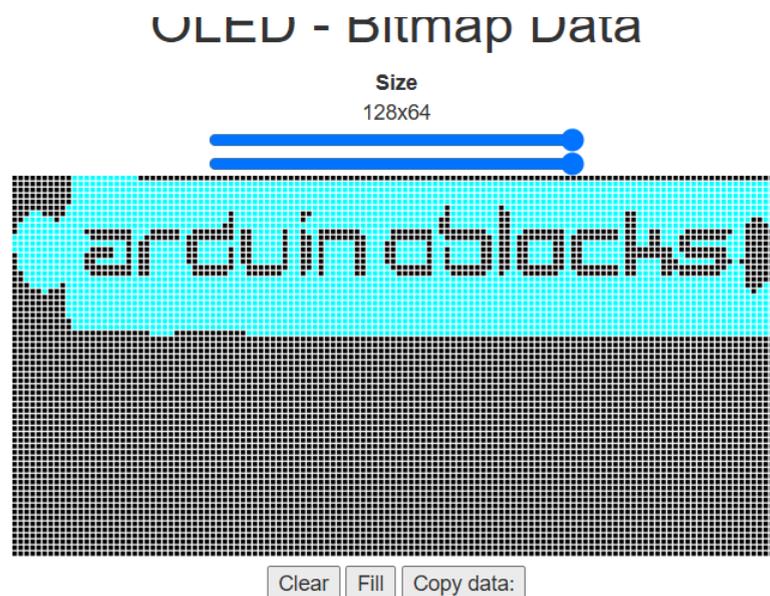
Fill: enciende todos los píxeles

Copia data: lo usaremos para copiar el código que se genera abajo, y que será en el que utilizemos para definir la imagen

Elegir archivos, nos permitirá elegir el archivo con el que queremos trabajar.

Procedemos a cargar la imagen y quedaría en la pantalla así:

Si la cogemos tal cual sale, vemos que no está ni centrada, ni coge en la pantalla completa. A continuación, con las barras que aparecen debajo, definiremos el tamaño y centraremos la imagen.



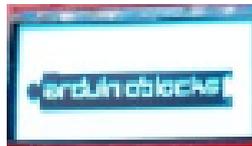
Debajo de la imagen, nos aparecen estos controles:

Con las barras X e Y, definiremos el centrado de la imagen.

Con zoom, ampliaremos/reduciremos la imagen

Con black-white, podremos variar la definición de la imagen.

Será cuestión de actuar sobre estos elementos para conseguir que la imagen quede centrada y bien definida.



En el siguiente [ejemplo](#), podremos ver más aplicaciones de la pantalla, pudiendo aprovechar la conexión Wifi del dispositivo, y mostrándolo en pantalla.

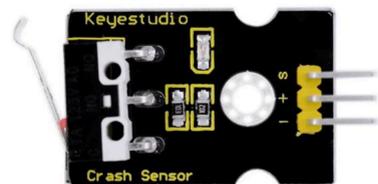
Programación con [MicroBlocks](#)

3.11 Sensor de colisión o final de carrera KS0021

El acceso a la página oficial del fabricante está disponible [aquí](#).

Este producto puede adquirirse de manera individual en la web de Innovadidactic [aquí](#).

El sensor de colisión, también conocido como final de carrera, es un módulo de entrada y salida digital necesario para el aprendizaje electrónico elemental. Su uso además es muy utilizado en la industria para controlar hasta dónde debe abrir por ejemplo una cancela o bien maquinarias para la detección de productos. A nivel mecánico y eléctrico es como el de un pulsador de los que tenemos por ejemplo para activar un timbre de nuestras casas.



1. Si la colisión ocurre al frente de donde está instalado el módulo de colisión, el módulo emite una señal de bajo nivel; sin colisión, emite señal de alto nivel.
2. El módulo reserva el orificio de montaje M3, conveniente para la fijación en un automóvil.
3. Tamaño del módulo: 3.1cm * 2.1cm
4. Con la luz indicadora del interruptor, si hay colisión, la luz está encendida; sin colisión, la luz está apagada.

Programación con [arduinoblocks](#)

Programación con [MicroBlocks](#)

3.12 Seguidor de línea KS0050

El acceso a la página oficial del fabricante está disponible [aquí](#).

Este producto puede adquirirse de manera individual en la web de Innovadidactic [aquí](#).

Este sensor de seguimiento de línea puede detectar líneas blancas en líneas negras y negras en blanco. La señal de seguimiento de línea única proporciona una señal de salida estable TTL para una línea más precisa y más estable. La opción multicanal se puede lograr fácilmente instalando sensores robóticos de rastreo de línea requeridos.

Fuente de alimentación: + 5V

Corriente de funcionamiento: <10mA

Rango de temperatura de funcionamiento: 0 ° C ~ + 50 ° C

Interfaz de salida: interfaz de 3 hilos (1 - señal, 2 - potencia, 3 - fuente de alimentación negativa)

Nivel de salida: nivel TTL

Tamaño : 41.7 * 10.7mm

Peso: 3g

Programación con [arduinoblocks](#)

Programación con [MicroBlocks](#)

3.13 Cables I2C

El acceso a la página oficial del fabricante está disponible [aquí](#).

Este producto puede adquirirse de manera individual en la web de Innovadidactic [aquí](#).

Programación con [arduinoblocks](#)

Programación con [MicroBlocks](#)

3.14 Cables sensores

El acceso a la página oficial del fabricante está disponible [aquí](#).

Este producto puede adquirirse de manera individual en la web de Innovadidactic aquí.

Programación con [arduinoblocks](https://www.arduino.cc/en/software/ArduinoIDE)

Programación con [MicroBlocks](https://www.microblocks.com)

3.15 Cables Dupont

El acceso a la página oficial del fabricante está disponible aquí.

Este producto puede adquirirse de manera individual en la web de Innovadidactic aquí.

Programación con [arduinoblocks](https://www.arduino.cc/en/software/ArduinoIDE)

Programación con [MicroBlocks](https://www.microblocks.com)

4. Conexiones IoT

Si hay algo que caracteriza a las placas STEAMakers, es su función IoT. Esta capacidad viene fundamentada en el microcontrolador ESP32, que como ya hemos hablado en varias ocasiones, trae implementada la comunicación Wifi y Bluetooth. Esta nos permitirá conectar la placa a un dispositivo móvil mediante la conexión inalámbrica por Bluetooth, o conectarlo a internet a WiFi con todas las posibilidades que esto trae asociadas. Para el ejemplo que vamos a desarrollar, conectaremos la pantalla por I2C a una micro:STEAMakers.

En los siguientes apartados, vamos a aprender a utilizar los diferentes bloques disponibles para convertir nuestra placa en un dispositivo IoT.

4.1 Conexión a una red WIFI

Programación con [arduinoblocks](https://www.arduino.cc/en/software/ArduinoIDE)

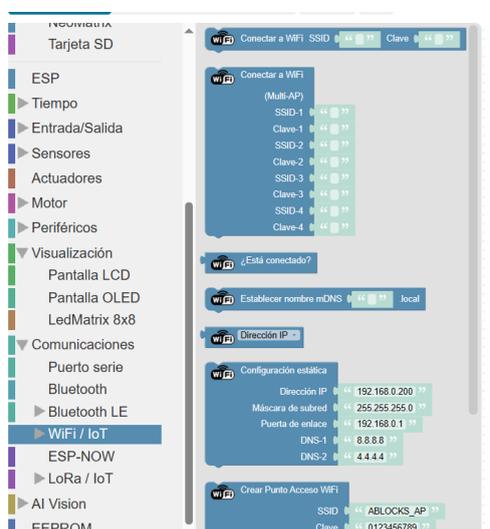
Lo primero que tendremos que hacer, es conectar nuestra placa a una red Wifi, y se conecte a internet. Como norma general, sólo será posible conectarse a redes wifi estándar, es decir de 2.4 GHz, no estando permitida la conexión a Wifi 5G.

Si no disponemos de wifi, podremos crear un punto de acceso en nuestro terminal móvil, y aprovechar la conexión de datos del dispositivo para poder conectar nuestra placa a internet.

Los módulos para la conexión wifi, están en el siguiente apartado:

Comunicaciones → Wifi/IoT

Una vez dentro de estos bloques, tendremos dos bloques que nos permitirán conectar la placa a una conexión Wifi. El primer bloque mostrado, será para una única red wifi, el segundo, nos permite definir una serie de conexiones, de manera que irá probando, hasta que pueda conectar a alguna de ellas.



En los bloques SSID, pondremos el nombre la red wifi y en Clave pondremos la contraseña.

Otra cuestión interesante, será asignar una dirección IP al dispositivo, y no depender de que el router la asigne de manera automática. De esta forma, para futuras aplicaciones, sabremos la dirección IP del dispositivo. Es muy recomendable, antes de poner una IP fija, asegurarnos que no la está utilizando otro dispositivo.

Si asignamos la dirección IP de un equipo que ya está en uso, nos podemos encontrar con que no se conecte de manera correcta. Si no es algo muy importante el hecho de mantener la IP, se recomienda ponerla automática, y de alguna forma, como ejemplo, mostrándola en pantalla, obtener dicha dirección.

Cabe indicar, que la mayoría de los router, en la primera conexión, asignan la IP, en función de la dirección MAC del dispositivo, por lo que siempre nos asignará la misma IP:

En los próximos ejemplos, aprenderemos a mostrar en la pantalla OLED, el nombre de la wifi, la ip, la dirección MAC....

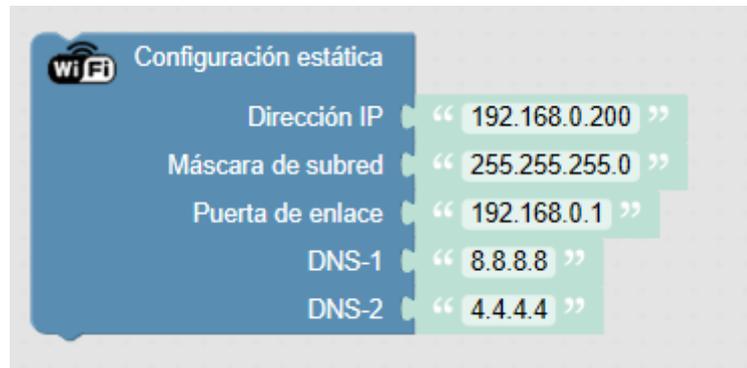
El router con el que trabajo habitualmente, trabaja en el rango 192.168.16.xx, por lo que ese será el rango que utilice en los ejemplos.

El siguiente bloque nos servirá para poder visualizar información sobre la conexión del dispositivo:



Esta será toda la información que podremos obtener de la conexión.

Si decidimos asignar la dirección IP al dispositivo, deberemos de dar la siguiente información:



También podremos asignarle un nombre, para en caso de querer acceder al dispositivo desde la red, no tener que acordarnos de la dirección IP. Para ello usaremos este bloque. Habría que poner nombre.local, y accederíamos de la misma forma que si usamos la dirección IP. :



4.1.1 Mostrar Fecha y Hora

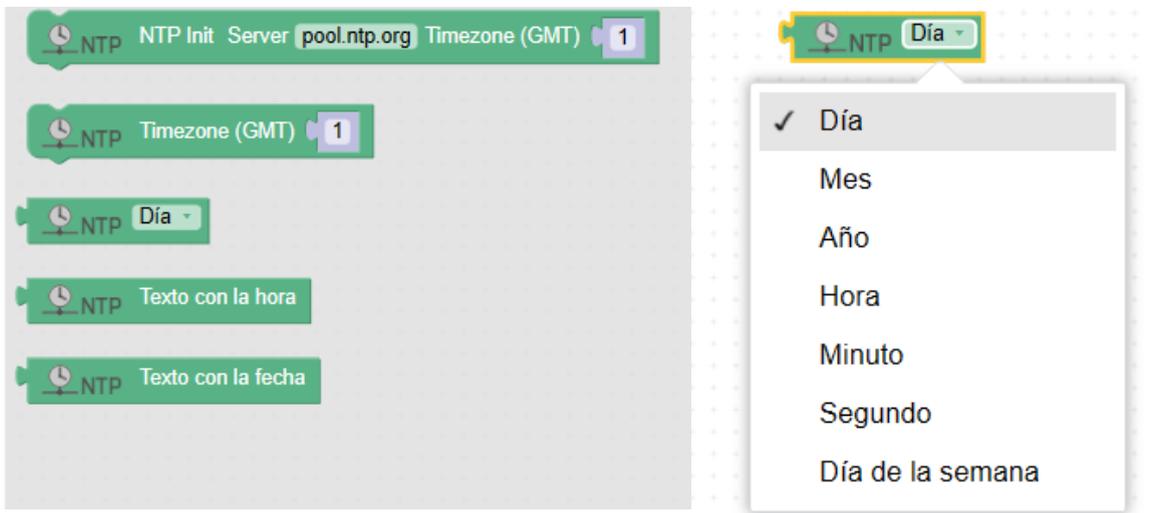
Podemos obtener la fecha y la hora en nuestro dispositivo, con sólo disponer de conexión a internet. Accederemos a un servidor NTP, que aloja la hora de internet, y lo tendremos automáticamente.

Wifi IoT → NTP Client

Después de conectarnos al wifi, conectaremos al servidor NTP, e indicaremos la zona horaria. En nuestro caso sería +2. Este bloque lo pondremos en inicializar.



Utilizando los siguientes bloques podremos obtener toda la información sobre la fecha y la hora:



Si utilizamos los siguientes bloques, obtendremos la información de la fecha y la hora en formato internacional. Si queremos mostrarlo en un formato que sea como estamos acostumbrados a visualizarlo, podemos agruparlas como queramos, utilizando los bloques unir texto, e incluso podremos alojarlo en una variable de tipo texto, para así poder usarlo más fácilmente.

En los siguientes bloques, podemos ver cómo hemos creado dos funciones, y dentro de estas unas variables en las que alojaremos los valores obtenidos del servidor de hora NTP, y lo mostraremos en el formato: dd/mm/aaaa y hh:mm.

```

+ para FECHA
  Establecer FECHA = + - crear texto con
    NTP Día
    "/"
    NTP Mes
    "/"
    NTP Año
+ para HORA
  Establecer HORA = + - crear texto con
    NTP Hora
    ":"
    NTP Minuto
    
```

En el siguiente ejemplo, vamos a mostrar en la pantalla OLED, la wifi a la que estamos conectada, la dirección IP, la dirección MAC, la fecha, la hora y el logotipo de arduinoblocks que habíamos hecho en un ejemplo anterior.

Comenzaremos con el bloque inicializar:

```

Inicializar
  WIFIS_CASA
  NTP NTP Init Server pool.ntp.org Timezone (GMT) 2
  # 1 Iniciar I2C 0x3C Mostrar automáticamente
  # 1 Limpiar
  # 1 Mostrar

+ para WIFIS_CASA
  Conectar a WiFi
  (Multi-AP)
  SSID-1 "M...o"
  Clave-1 "p...l"
  SSID-2 "se...10"
  Clave-2 "JL...DR"
  SSID-3 ""
  Clave-3 ""
  SSID-4 ""
  Clave-4 ""
  
```

A continuación definiremos las funciones con las que generamos la fecha y la hora:

```

Bucle
  FECHA
  HORA
  # 1 Texto X 0 Y 0 + - crear texto con " WiFi: " Led ON small
    WiFi WiFi-SSID
  # 1 Texto X 0 Y 10 + - crear texto con " IP: " Led ON small
    WiFi Dirección IP
  # 1 Texto X 0 Y 20 + - crear texto con " MAC: " Led ON small
    WiFi Dirección MAC
  # 1 Texto X 0 Y 30 FECHA Led ON small
  # 1 Texto X 60 Y 30 HORA Led ON small
  
```

```

+ para FECHA
  Establecer FECHA = + - crear texto con
    NTP Día
    " / "
    NTP Mes
    " / "
    NTP Año

+ para HORA
  Establecer HORA = + - crear texto con
    NTP Hora
    " : "
    NTP Minuto
    
```

Y para terminar, exponemos el “bucle”, en el que indicamos cómo mostrar las variables en la pantalla, y mostramos el bitmap del logotipo.

Para evitar parpadeos en la pantalla, y puesto que no hemos contemplado los “segundos”, pondremos un refresco de pantalla cada 60000 ms (60 segundos).

Este sería el resultado:



Podemos ver analizando las dos imágenes, el detalle de dirección MAC. La que nos muestra la pantalla coincide la que viene en la etiqueta de fabricación. La dirección MAC, es una dirección única del dispositivo.

4.2 Conexión por Bluetooth

4.2.1 Control con APP