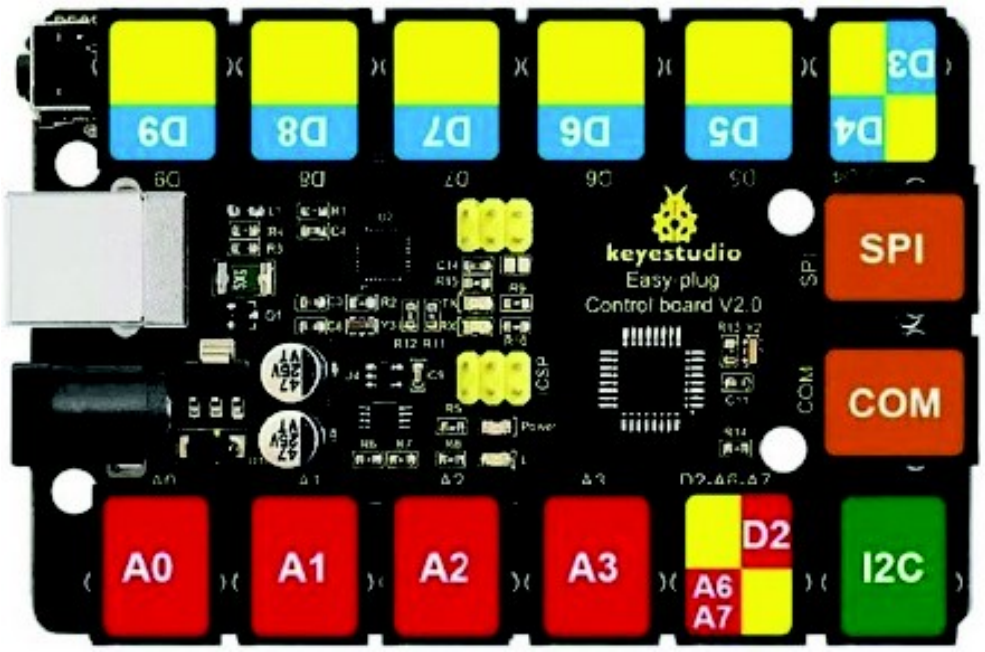
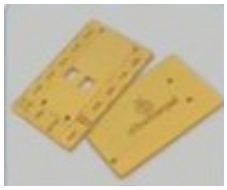
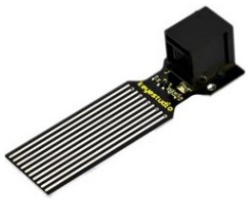
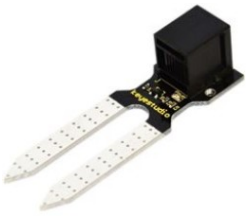
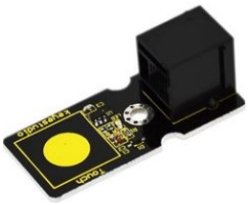
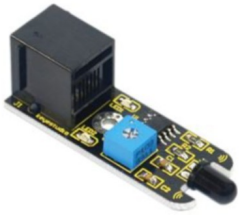
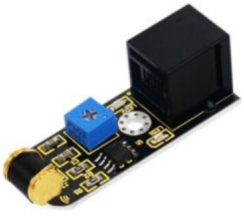
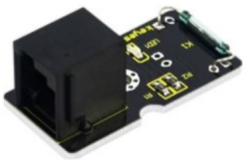
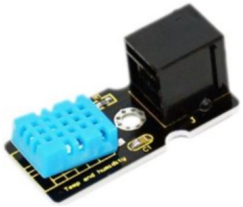
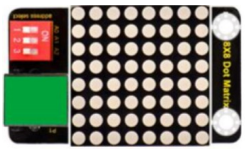
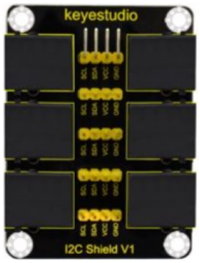
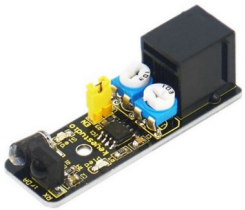
PROGRAMACIÓN DE LA PLACA EASYPLUG CON ARDUINOBLOCKS

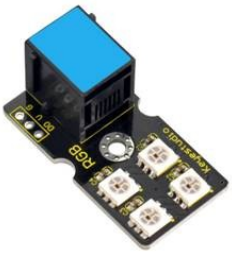
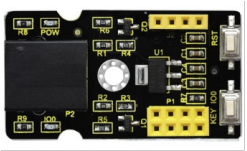
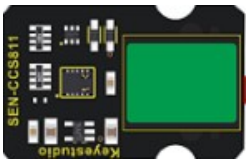
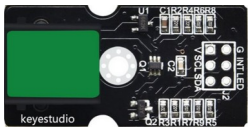
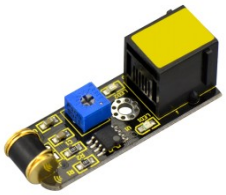


Versión: 1.2 - 10/22

Lista de elementos necesarios y ampliaciones

				
Placa EASY plug	Paneles acrílicos	Tornillos	Separadores	Cable USB
				
Cables RJ45	LED blanco	Zumbador activo	Zumbador pasivo	LED rojo, verde, amarillo o azul
				
Sensor analógico de temperatura	Sensor analógico de sonido	Sensor LDR	Detector de nivel de agua	Sensor de humedad del suelo
				
Potenciómetro rotatorio	Sensor de campo magnético Hall	Sensor de colisión	Botón pulsador	Botón táctil capacitivo
				
Sensor de golpe	Sensor de inclinación	Sensor de llama	Sensor de vibración	Modulo relé Reed

				
Sensor de temperatura LM35	DHT11	Pantalla LCD I2C		Pantalla OLED
				
Matriz 8x8 LEDs	Hub I2C	Sensor PIR	Módulo joystick	Sensor de Ultrasonido SR01 V2
				
Módulo relé	Módulo Bluetooth	Sensor de Ultrasonido SR01 V2	Sensor de luz ambiental TEMT6000	Sensor infrarrojo para evitar obstáculos
				
Servomotor	Adaptador Easy Plug a 3 Pines	Pantalla gráfica OLED 0.96" 128 x 64	Sensor de calidad del aire MQ-135	Módulo RTC DS3231

				
Aceleró metro de tres ejes ADXL345	Potenciómetro deslizante		Receptor de infrarrojos IR	Emisor de infrarrojos IR
				
Módulo ventilador L9110	Módulo Bluetooth	Módulo RGB con 4 Neopixel WS2812	Módulo ESP-01S WiFi + Bluetooth	Módulo WiFi ESP8266
				
CCS811 Sensor de eCO2 y TVOC	Sensor de color TCS34725	Relé Reed	Sensor de vibración o impacto	

Elementos y componentes Easy Plug

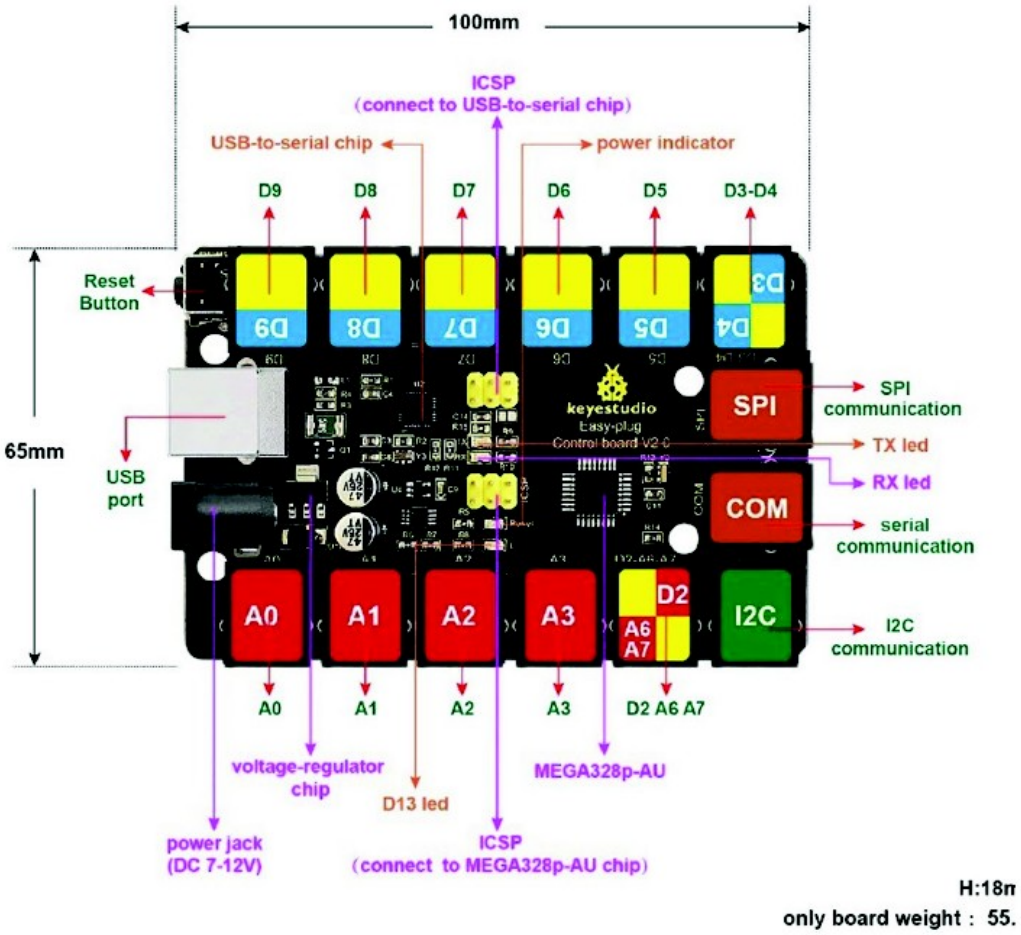
¿Que es la placa Easy Plug?

Es una placa donde podemos conectar diferentes sensores y actuadores de forma muy sencilla. Además, se pueden crear diferentes programas, proyectos y experimentos muy interesantes para niñas y niños de los últimos cursos de Educación Primaria así como, durante la etapa de educación secundaria.

¿Qué características técnicas tiene la placa Easy Plug?

La placa Easy Plug, es compatible con Arduino Uno. Tiene cinco puertos digitales (de entrada y salida), cuatro puertos de entrada analógica, un puerto SPI y un puerto I2C.

Sus conectores son pinos RJ-11 de cuatro hilos, lo que les hace mucho más fáciles para conectarlos y evitar malas conexiones.



ArduinoBlocks: programa, acceso e interfaz

¿Que es Arduino?

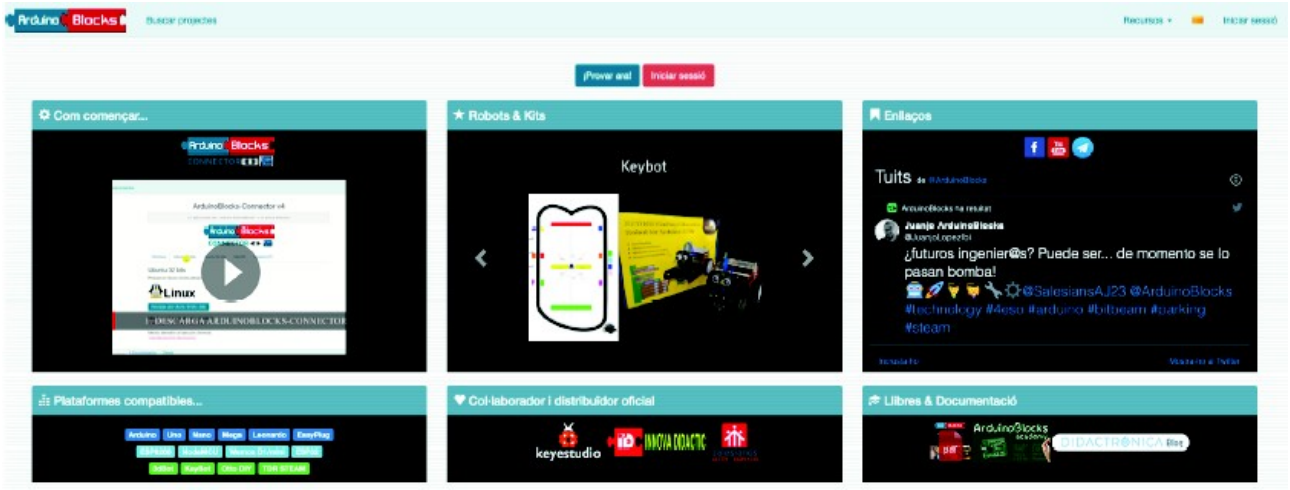
Arduino es una plataforma de prototipos de código abierto, basada en hardware y software flexibles y fáciles de utilizar. Su finalidad es que todo el mundo pueda generar proyectos y entornos interactivos mediante placas y sensores compatibles con Arduino.

¿Que es ArduinoBlocks?

Arduino se programa con lenguaje C++ y se necesita el IDE (Integrated Development Environment), que permite escribir el código. Programar con C++ puede resultar complejo y no es accesible para todos, por eso, Juanjo López creó ArduinoBlocks. Así pues, ArduinoBlocks es un lenguaje de programación por bloques. Los diferentes bloques de programación sirven para leer y escribir las entradas y salidas de la placa.

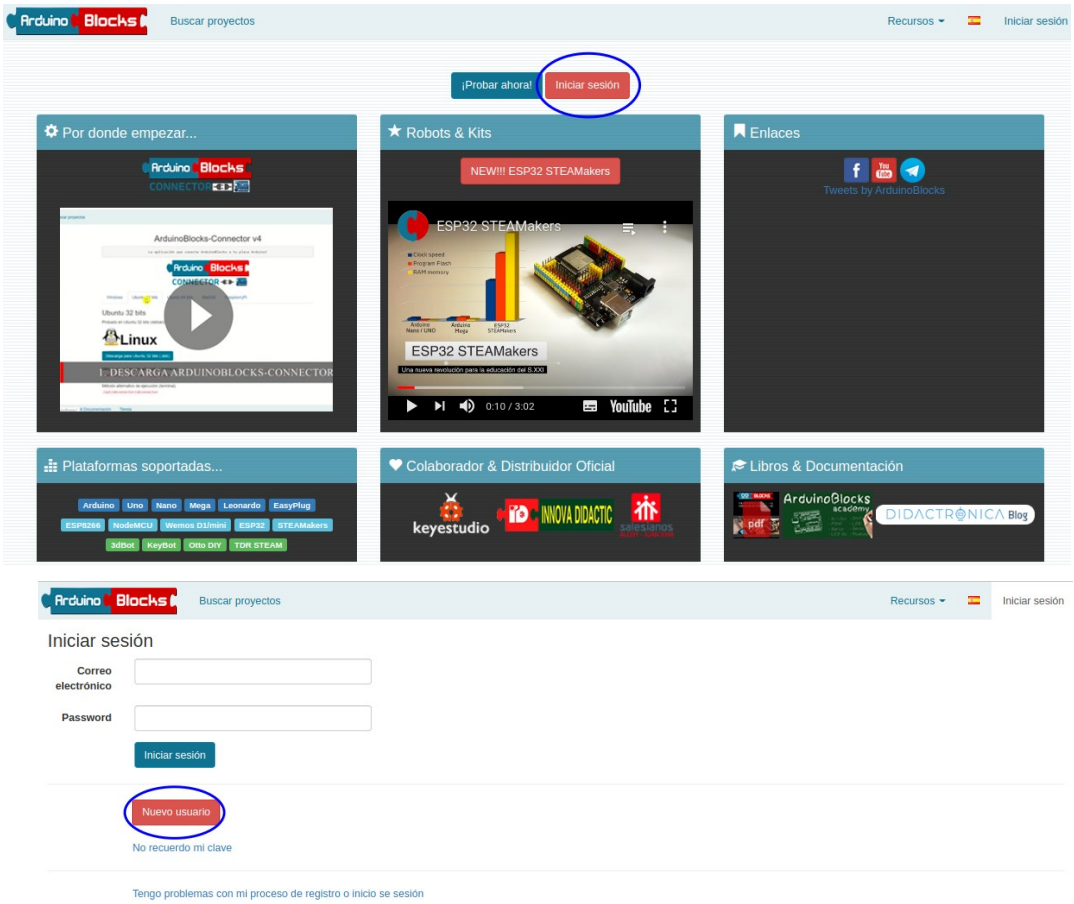
Encontramos ArduinoBlocks, en la siguiente página web: http://www.arduinoblocks.com.

Podemos registrarnos en la web, para guardar los proyectos, podemos añadir información a nuestros proyectos y ver proyectos de otros usuarios que hayan decidido compartirlos de forma pública.



¿Cómo se trabaja con ArduinoBlocks?

Para empezar a trabajar con ArduinoBlocks, es necesario registrarse y crear un nuevo usuario. Debemos acceder al botón "Iniciar sesión" para después, seleccionar la opción de nuevo usuario.



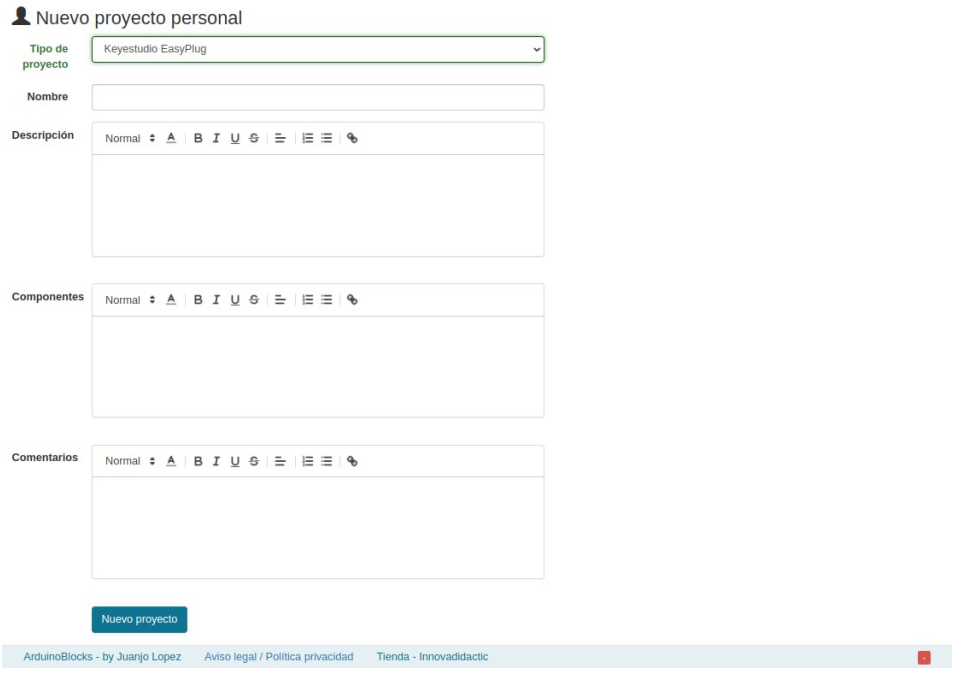
Seguidamente, se debe seleccionar "Empezar un proyecto nuevo". Nos aparecen tres opciones: proyecto personal, profesor o alumno.

Si escogemos el proyecto personal sólo podremos acceder nosotros, pero después lo podemos compartir si decidimos hacerlo público. En cambio, si seleccionamos el proyecto profesor, no se comienza un proyecto sino que se crea un código para que los alumnos puedan inscribirse en el proyecto. De esta forma, el profesor puede supervisar las programaciones de sus estudiantes. Por último, la opción de alumno es para unirnos al proyecto planteado por el profesor. Para aprender mas sobre Usuarios Gestionados basta con acceder a la presentación del enlace anterior que se adjunta como anexo a este documento.

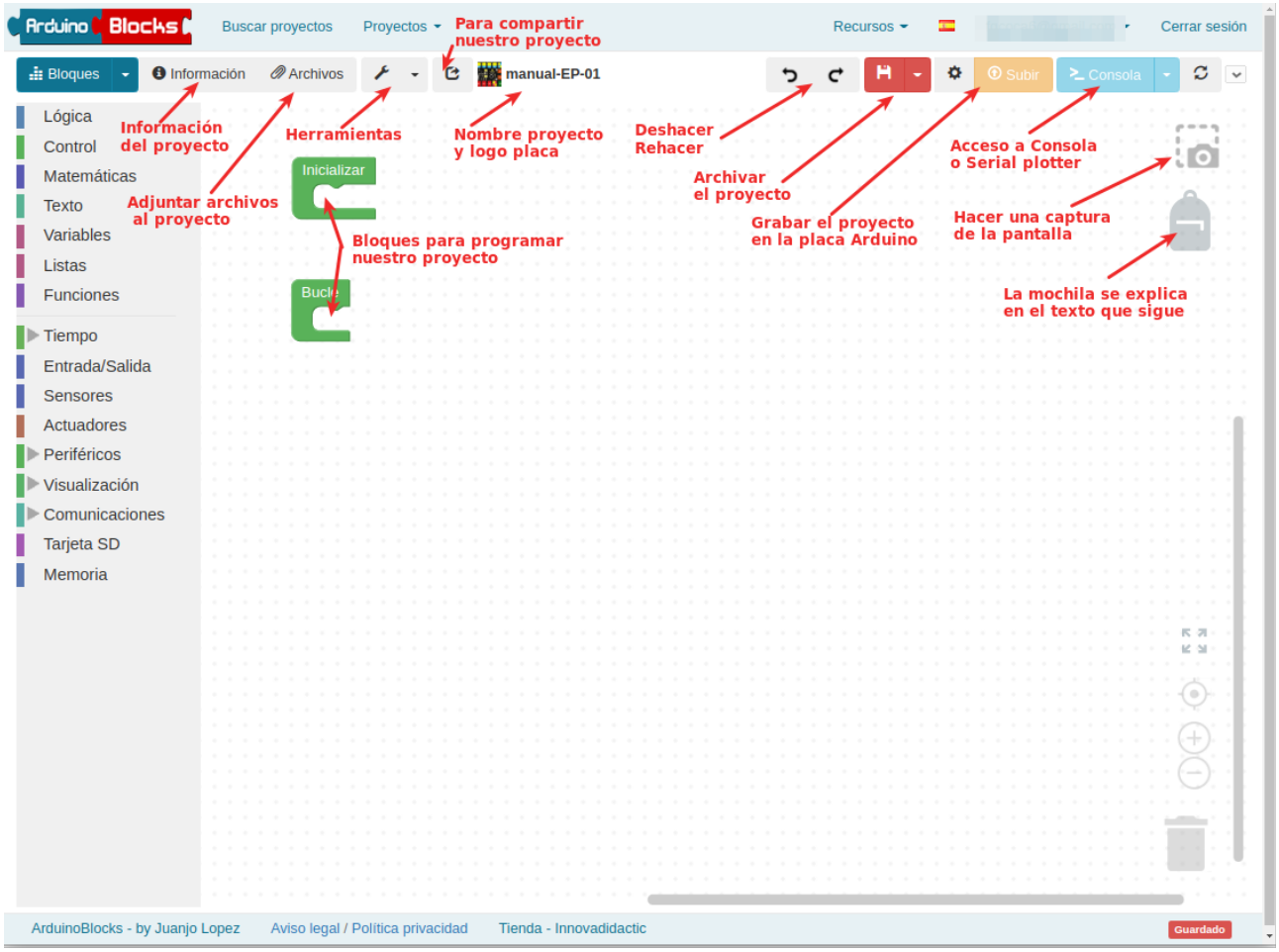
¿Cómo es la interfaz de programación de ArduinoBlocks?

Una vez hemos escogido la opción de proyecto personal, nos pregunta qué placa estamos utilizando y qué nombre queremos dar a nuestro proyecto. A su vez, podemos añadir información, como: descripción del proyecto, componentes que necesitamos y otras especificaciones en el apartado de comentarios.

En nuestro caso escogeremos: **Keyestudio EasyPlug.**



Una vez ponemos nombre, y aunque es opcional se recomienda cumplimentar los distintos apartados, y hacemos clic en el botón "*Nuevo proyecto*" entramos en el nuevo proyecto, nos encontramos con la siguiente pantalla:



Ya es posible guardar nuestros bloques favoritos de ArduinoBlocks en la mochila y llevarlos siempre con nosotros o usarla para compartir bloques entre nuestros proyectos. En este enlace tienes un video que lo explica de forma gráfica.

ArduinoBlocks Connector

ArduinobBloks genera el código de Arduino a partir de los bloques. El programa copia y sube nuestra programación a la placa gracias a la aplicación ArduinoBlocks Connector. Si no ejecutamos ArduinoBlocks Connector, podremos programar y acceder a la plataforma ArduionBlocks, pero no podremos subir nuestro programa a la placa. Así pues, es necesario instalarlo en nuestro ordenador, y ejecutarlo cuando hagamos uso de la plataforma.

Actualmente, está disponible la versión 5 para estos sistemas operativos: Windows, Ubuntu, MacOS, Chromebook y Raspberry Pi.

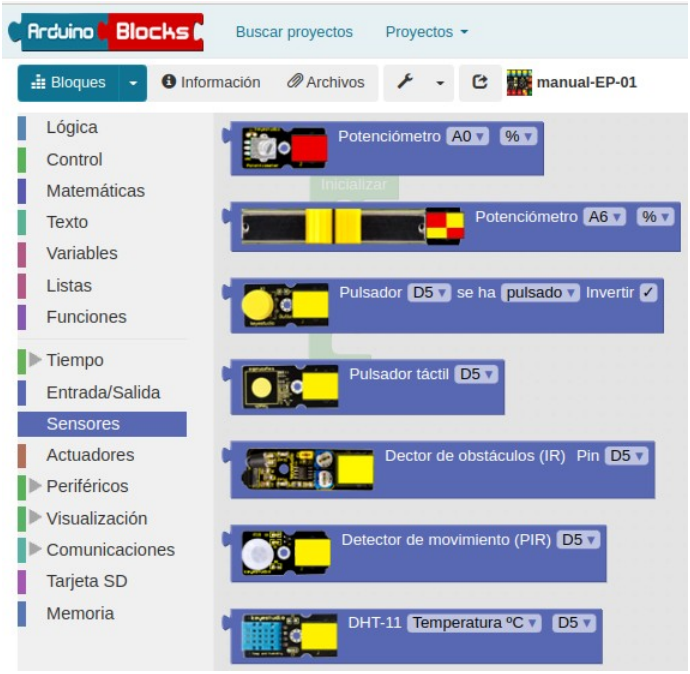
Sensores

¿Qué es un sensor?

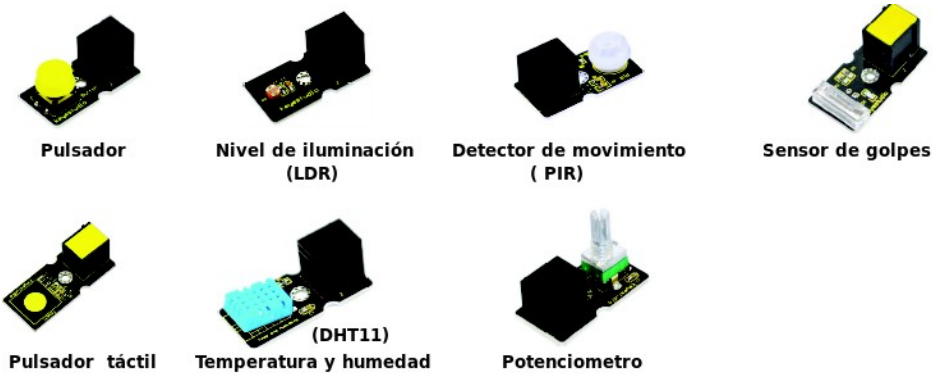
Los sensores permiten recabar información para que sea procesada para la placa y se accionen ciertas programaciones según nuestra finalidad.

¿Qué sensores hay?

Hay varios tipos de sensores. Por ejemplo, si vamos al apartado de "Sensores" del menú de herramientas de EasyPlug, existen varios. Algunos de ellos los vemos en la imagen siguiente:



¿Qué sensores programaremos?



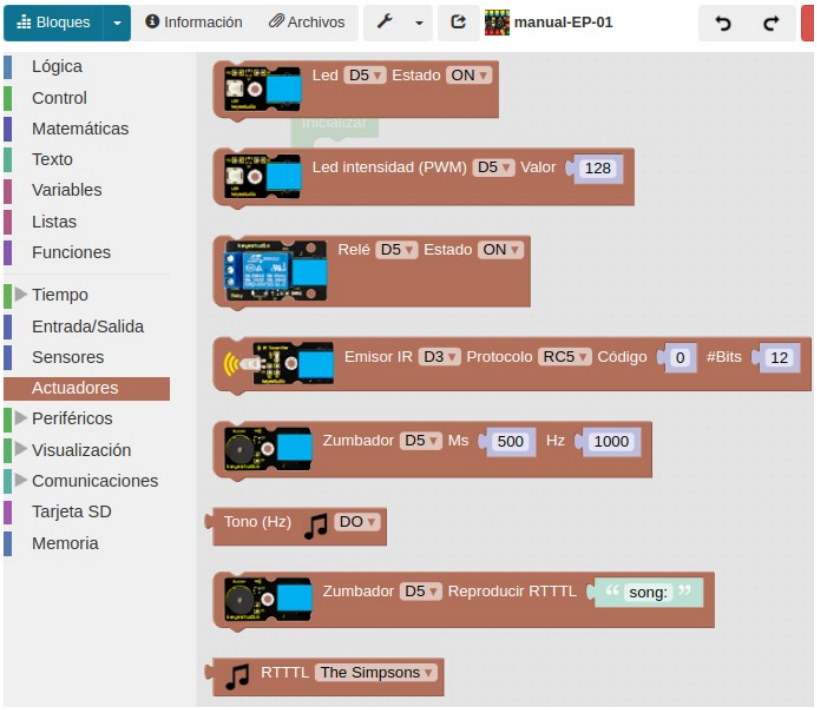
Actuadores

¿Qué es un actuador?

Los actuadores se accionan cuando la placa procesa la programación, que ésta puede estar influida por la recogida de datos de algún sensor.

¿Qué actuadores hay?

Hay varios tipos de actuadores. Por ejemplo, si vamos al apartado de "actuadores" del menú de herramientas de EasyPlug, existen varios. Algunos de ellos son:



¿Que actuadores programaremos?

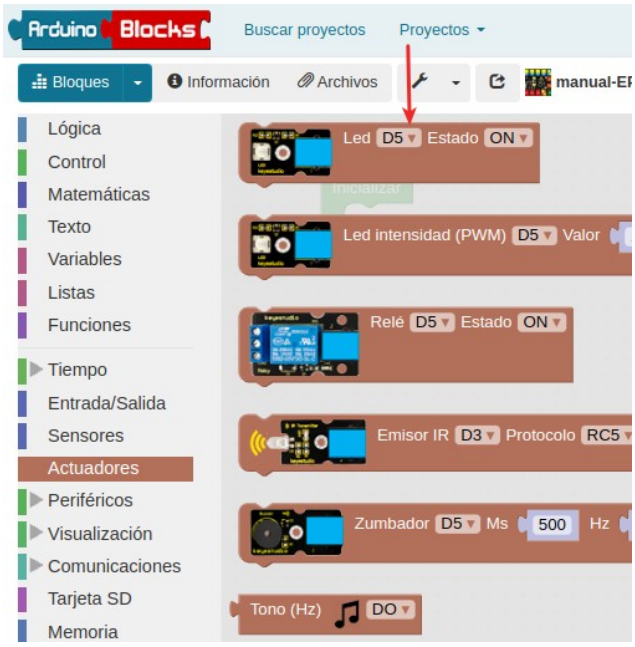
LED
Nivel iluminación
(PWM)

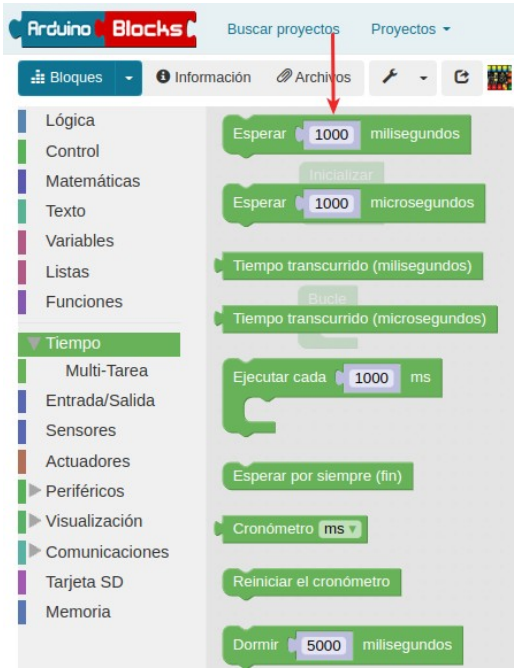
Zumbador

A01: LED

En esta primera práctica con EasyPlug aprenderemos a encender un LED y a programarlo para que se apague y encienda durante un tiempo determinado.

El LED es un actuador y por tanto, lo encontraremos en el apartado de actuadores. Si necesitamos programar un tiempo, deberemos ir al apartado de "Tiempo".





PRÁCTICA A01.1:

- Encendre un LED

En el ejemplo la espera en realidad no sirve para mucho y el LED permanecerá a nuestros ojos siempre encendido

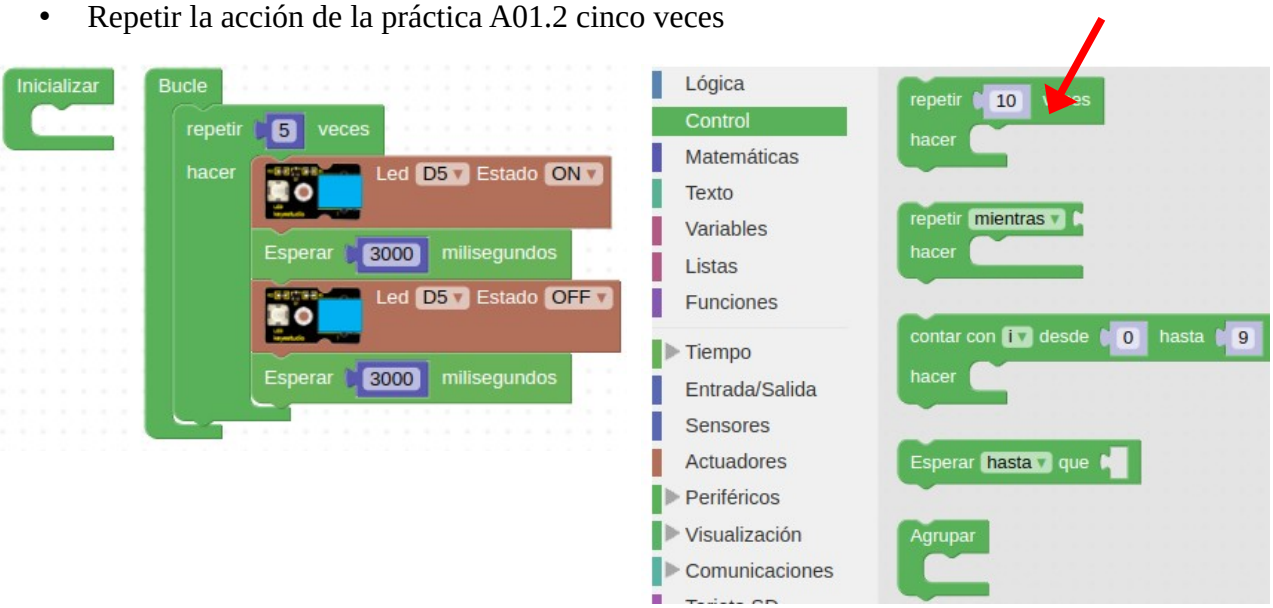
PRÁCTICA A01.2:

- Encendre un LED, esperar 3 segundo y apagar y LED

Deduce y comprueba el efecto de no poner el tiempo de retardo tras apagar el LED.

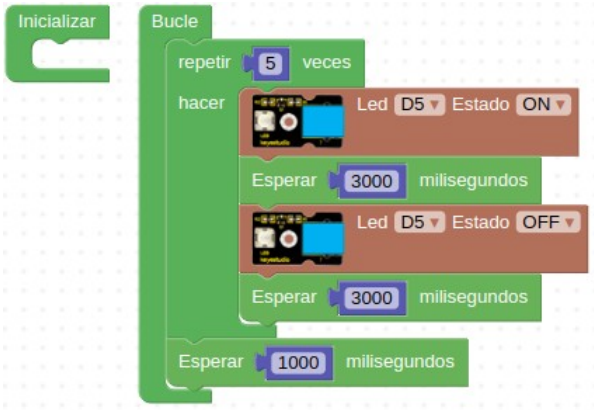
PRÁCTICA A01.3:

- Repetir la acción de la práctica A01.2 cinco veces



PRÁCTICA A01.4:

- Repetir la acción de la práctica A01.3 esperando un segundo entre repeticiones.



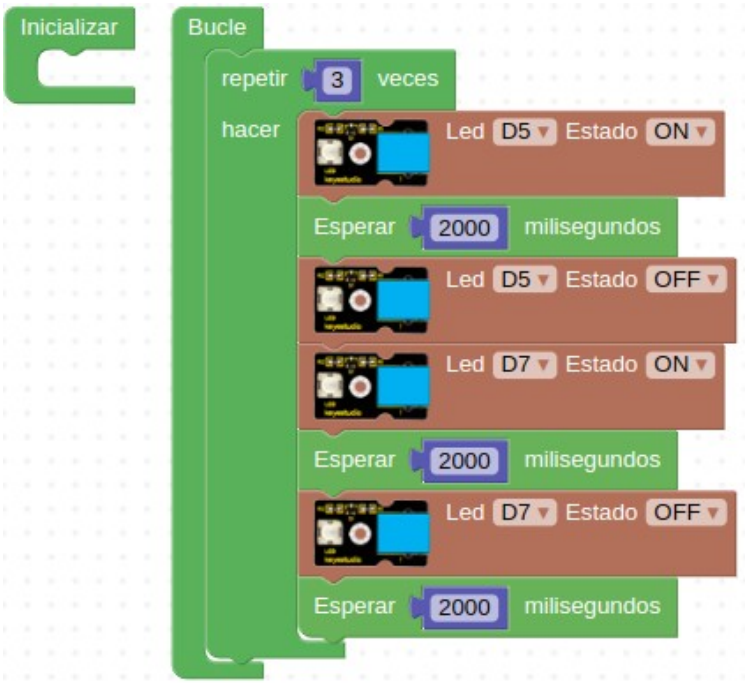
PRÁCTICA A01.5:

Conectamos dos LEDS a la placa:

- LED 1: D5

- LED 2: D7

- Se enciende LED 1 (D5) esperamos 2 segundos, se apaga el LED 1 y se enciende el LED 2 (D7) y esperamos 2 segundos y se apaga. Esta acción se repite 3 veces.



A02: Control de la intensidad de un LED utilizando PWM

Siguiendo con el uso de los LEDs, controlaremos la intensidad de un LED utilizando el PWM, que es el acrónimo de Pulse Width Modulation y significa Modulación por Anchura de Pulsos.

Las patillas de salida de Arduino sólo tienen dos estados posibles: ON y OFF, es decir, una corresponde a una salida de 5 V (ON) y una de 0 V (OFF). Si sólo hacemos uso de estas dos condiciones, sólo podemos realizar actividades como encender y apagar un LED. Gracias al PWM podemos programar un rango de valores de 0 a 255 que haga oscilar estas tensiones entre los 0 V y los 5 V. De esta forma podemos controlar la intensidad de un LED.

Encontramos el blog de programación en el apartado de actuadores:

PRÁCTICA A02.1:

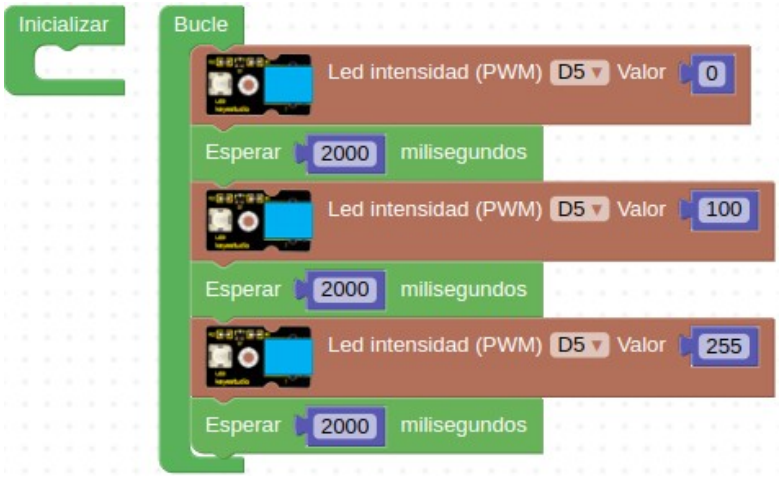
- Programamos el LED a un valor de 255 para que se encienda al máximo de iluminación y después de un segundo, se apague. Es decir, en valor 0.

Ejemplo similar al de la práctica A01.2.

PRÁCTICA A02.2:

A partir de la práctica A02.1 hacer que la intensidad del LED cambie.

- Cada 2000 milisegundos la intensidad del LED debe cambiar, aumentando de 0 a 100 y 255.



PRÁCTICA A02.3:

Para poder programar de una forma más práctica y efectiva, podemos hacer uso del bloque “Contar” situado en el apartado de "Control". Éste permite contar desde un determinado número hasta otro. A su vez, podemos decidir con qué frecuencia, es decir, cada dos números o cada diez.

En este bloque existe un nuevo concepto que es muy utilizado en programación: las variables. En esta actividad las utilizaremos de forma muy rápida, pero más adelante serán herramientas necesarias para que nuestra programación tenga sentido. Justamente, en nuestro bloque la variable se llama "i".

- La intensidad del LED debe aumentar de 25 en 25, desde 0 a 255.

El bloque con el nombre de la variable lo vamos a encontrar en “Variables”.

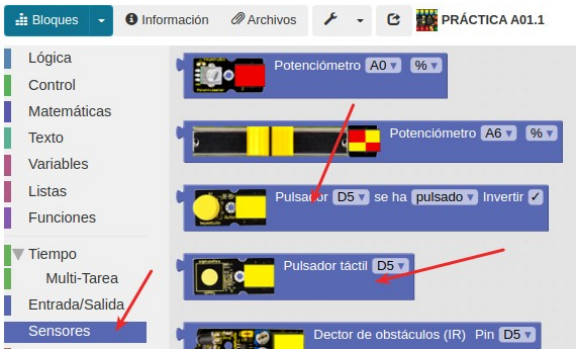
A03: Encender un LED con un pulsador

Continuando con el uso de LED, en las siguientes prácticas combinaremos dos pulsadores: el táctil y el pulsador normal.

El pulsador es un sensor digital, que tiene dos estados. Cuando se presiona el botón, emite una señal de nivel alto, es decir, 5 V. En cambio, cuando se suelta el botón, emite una señal de nivel bajo: 0 V.

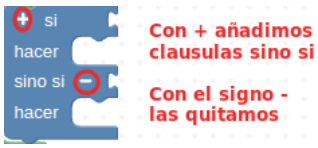
El pulsador táctil es un sensor táctil, tal y como explica su nombre puede "sentir" el tacto de forma muy sensible.

Los dos pulsadores son sensores, por tanto, los encontraremos en el apartado "Sensores". En el momento de la programación es muy importante tener en cuenta a qué puerto lo hemos conectado. Siempre se conectan a los puertos digitales, pero es necesario programar el número correcto.



A la vez también empezaremos a hacer uso de las funciones de "Lógica", como la condicional: "si… hacer". Este bloque de programación es uno de los pilares fundamentales en el mundo de la programación, puesto que permite evaluar estados, y según la condición, programar acciones.

En el apartado de condiciones se pueden introducir factores como: estado de los sensores, comparaciones, igualdades, operaciones matemáticas. En cambio, en el apartado de acciones, podemos programar: encender un LED, enviar mensajes a la consola, escribir algo en alguna pantalla externa, etc.



También, se puede ampliar el bloque con más condicionales. Haciendo clic en el símbolo “+” y quitarlos con el signo “-”.

PRÁCTICA A03.1:

- Si hacemos clic en el pulsador el LED se enciende durante 2 segundos.

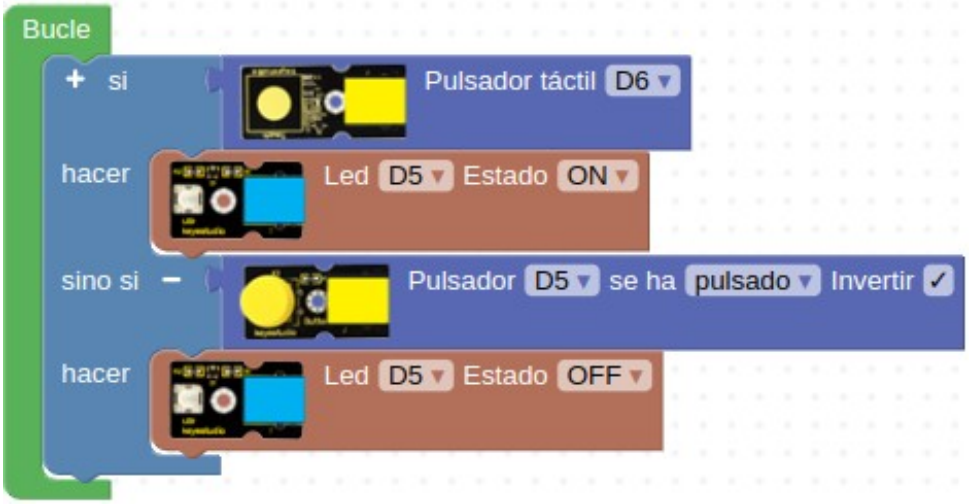
PRÁCTICA A03.2:

- Si hacemos clic en el pulsador táctil el LED se enciende durante 2 segundos.

PRÁCTICA A03.3:

Ahora utilizaremos los dos pulsadores

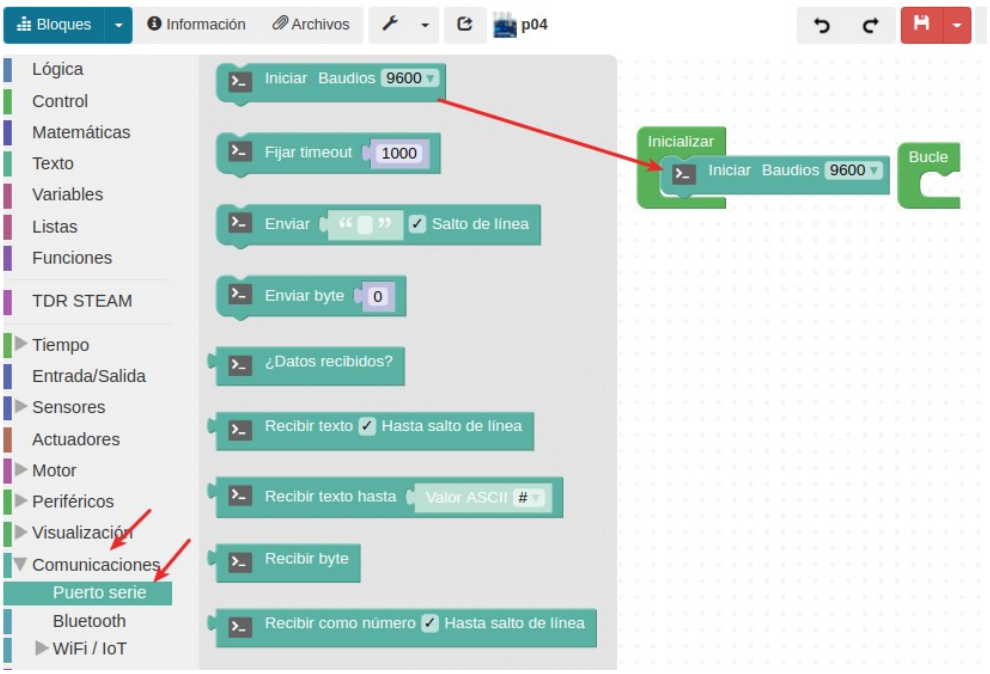
- Si hacemos clic en el pulsador táctil se enciende el LED. Sin embargo, si hacemos clic en el otro pulsador, se apaga.



A04: Consola o monitor serie

En esta práctica trabajaremos con la consola. Es una función del programa ArduinoBlocks que podemos programar para que se envíe información a la pantalla de nuestro ordenador.

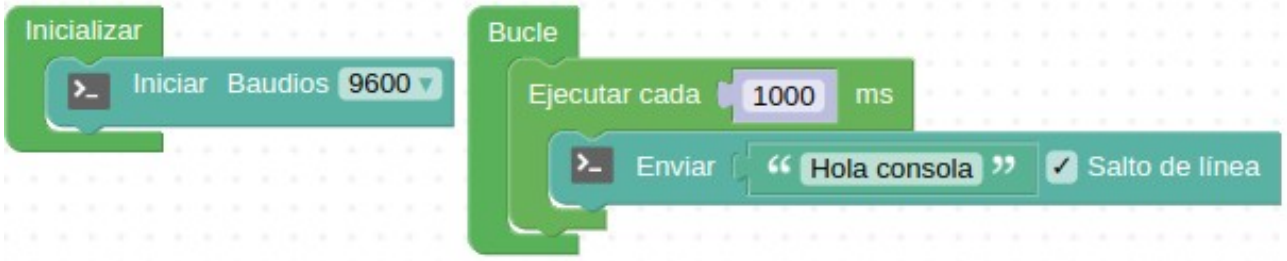
Para poder "enviar" esta información necesitamos inicializar el "Puerto serie". Por tanto, dentro del bloque inicializar pondremos: "Iniciar Bauds". En ese caso, para enviar mensajes a la consola con el valor 9600 es suficiente.



PRÁCTICA A04.1:

Comenzaremos esta primera actividad, enviando un mensaje a la consola, que diga "Hola consola". Es necesario inicializar el Puerto Serie, utilizando los Bauds.

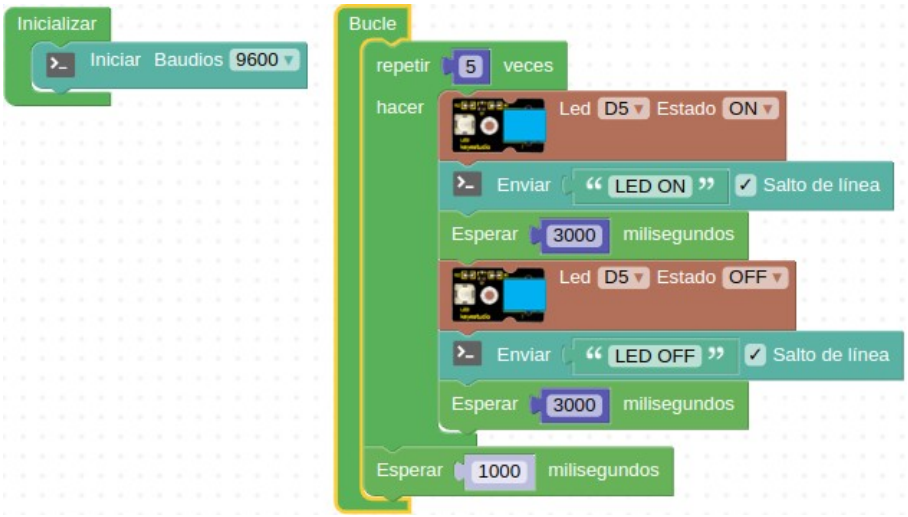
- Cada segundo, en la consola debe aparecer "Hola consola" con salto de línea.



PRÁCTICA A04.2:

Ahora que sabemos enviar un mensaje a la consola, combinaremos diferentes bloques de programación: control, actuadores, puerto serie y tiempo.

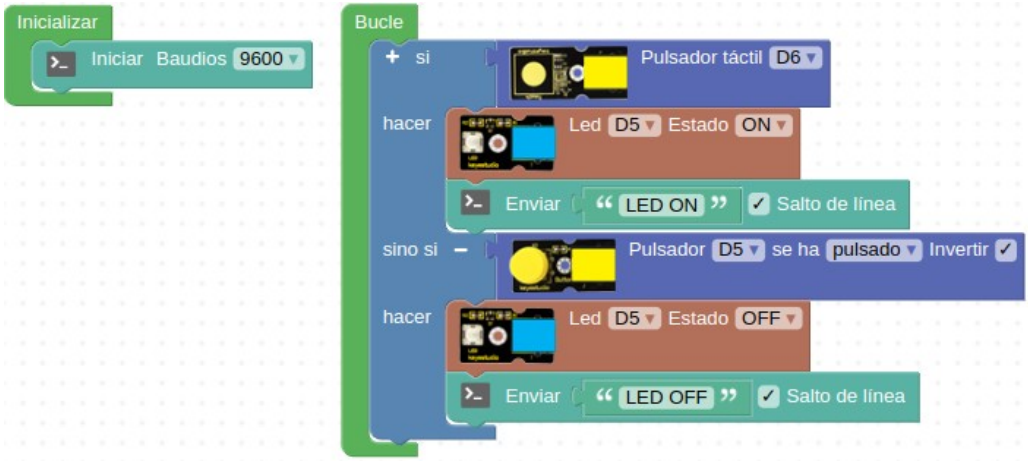
- Este programa se repite durante 5 veces y entre repetición y repetición hay un segundo de espera: se enciende el LED y automáticamente se envía a la consola "LED ON", durante 3 segundos. Luego se apaga, y en la consola aparece "LED OFF" durante 3 segundos.



PRÁCTICA A04.3:

Ahora programaremos utilizando los bloques de lógica.

- Si hacemos clic en el pulsador táctil, el LED se enciende y en la consola aparece "LED ON". En cambio, si hacemos clic en el pulsador normal, el LED se apaga y en la consola se escribe: "LED OFF".



A05: Sensor LDR

En esta quinta práctica aprenderemos a programar el Sensor de instensidad de luz o cómo se llama en ArduinoBlocks: Nivel de LUZ (LDR). Este sensor es necesario conectarlo a un puerto analógico.

El sensor Nivel de LUZ (LDR), también llamado Fotocélula, es muy común en nuestra vida cotidiana. Por ejemplo, cuando se encienden las farolas por la noche, las luces solares de jardín, los detectores de dinero, ...

Es un sensor analógico que obtiene valores entre 0 y 5V, concretamente de 0 a 1023.

En el apartado de bloques de programación, se encuentra en "Sensores".

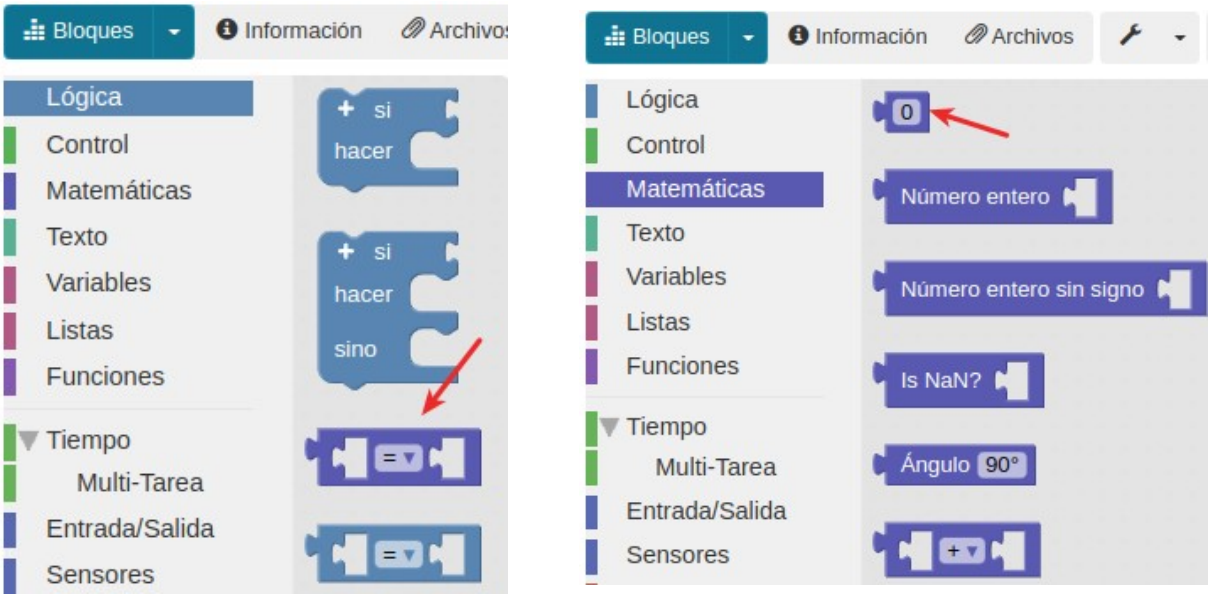
PRÁCTICA A05.1:

Antes de empezar la programación, necesitamos conocer qué valores marca el sensor en el lugar en el que trabajamos. Es decir, no podemos programar sin conocer el % de intensidad de luz que tenemos en nuestro entorno. Una vez obtenido este valor, podremos empezar las programaciones más complejas. Así pues:

- Ver en la consola el % de intensidad de luz cada 3 segundos.

PRÁCTICA A05.2:

Para realizar esta segunda práctica, necesitamos conocer el bloque de programación que nos permite igualar o comparar si un número es mayor o menor que otro. Además, utilizaremos el bloque que nos permite escribir números. En las siguientes imágenes vemos los diferentes bloques de programación:

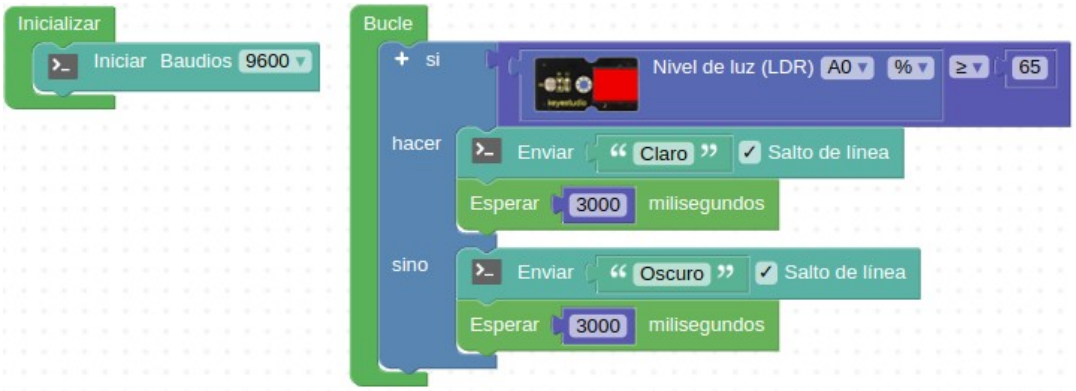


- Si la intensidad de luz es superior a un valor X (tomar como referencia el medido en la práctica anterior), en nuestro caso 65%, que en la consola aparezca "Claro".

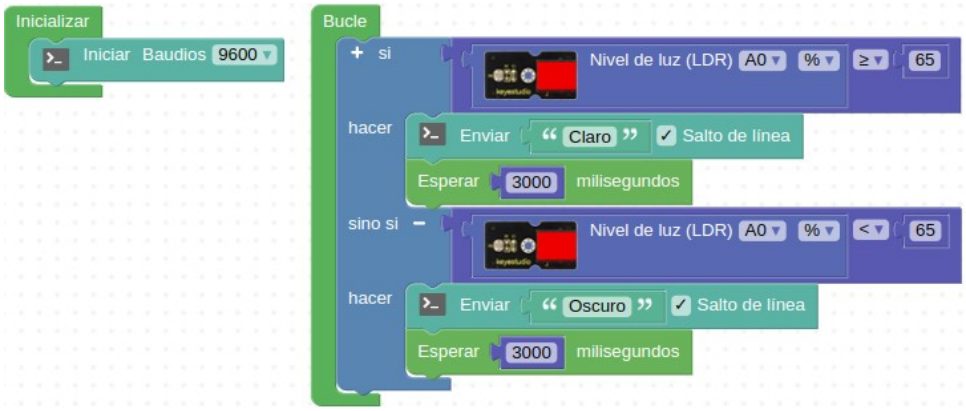
PRÁCTICA A05.3:

- Si la intensidad de luz es superior a un valor X, en nuestro caso 65%, que la consola escriba "Claro". Sin embargo, si es inferior al valor anterior, que la consola escriba "Oscuro".

Hay dos formas de programarlo, la primera necesitas especificar qué condición debe suceder para que la consola escriba "Oscuro".



En la segunda forma se entiende que todo lo que no sea la primera condición será la segunda.



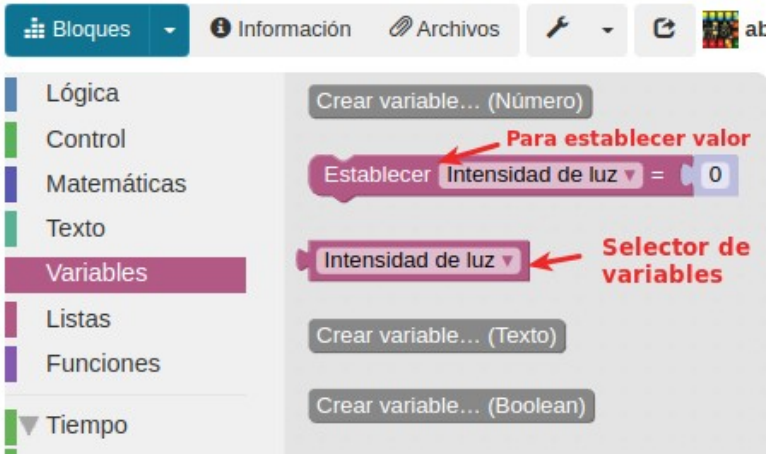
Hemos cambiado el bloque “si … hacer” por el bloque “si … hacer … sino”

PRÁCTICA A05.4:

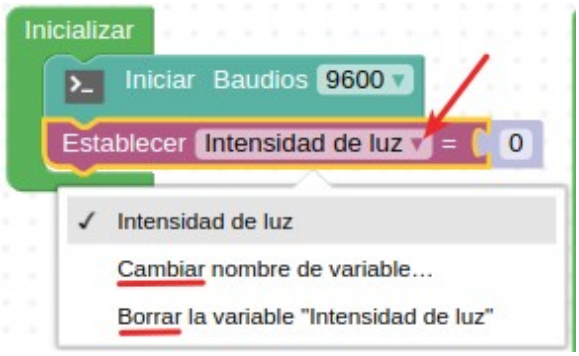
En esta cuarta práctica, volveremos a trabajar con las variables. Las variables son elementos comunes en programación. Cuando creamos una variable estamos dando un nombre a un dato o una lectura. Por ejemplo, cuando el sensor de intensidad de la luz detecta el % de luz de la habitación, toda la medida de los diferentes valores podemos agruparlos con la variable nombrada "Intensidad de luz". No es obligatorio utilizar variables, pero es una forma mucho más cómoda y comprensible de crear nuestros programas.

Para crear una nueva variable, debemos ir al apartado de variables y establecer una del tipo que necesitemos. Es decir, asignaremos un nombre a un conjunto de valores comunes. Cuando no tenemos ninguna variable creada en nuestro programa el aspecto que se muestra es el siguiente:

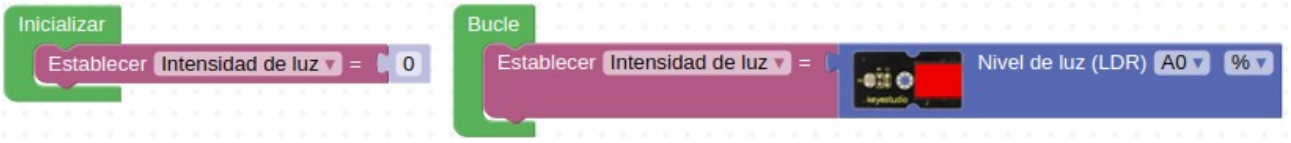
Cuando creamos una de tipo número llamada “Intensidad de luz” el aspecto cambia a:



Si necesitamos renombrar o eliminar una variable debemos mostrar el desplegable del bloque “Establecer valor” y escoger la opción que corresponda a nuestro caso



Es necesario establecer el valor de la variable en el bucle, y asociarla a un sensor que pueda agruparnos los valores. Lo habitual y recomendable en programación es definir e inicializar las variables a “0” o a un valor que nos interesa en el bloque “Inicializar” y en “Bucle” trabajar con ellas.



Repetiremos los mismos ejercicios que las prácticas anteriores pero utilizando la variable en lugar del bloque del sensor.

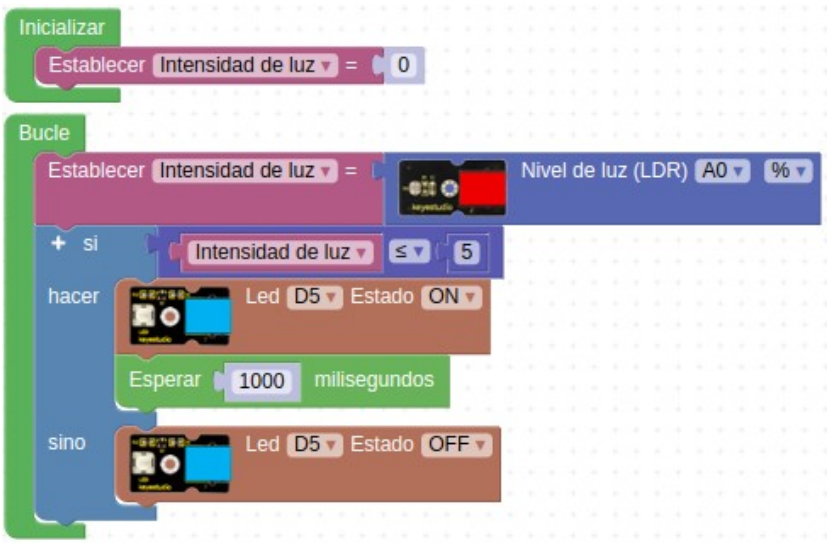
- Envía a la consola el % de intensidad de luz cada 3 segundos. En la consola debe aparecer escrito de la siguiente manera: Intensidad luz: XX,XX.

Para ello, podemos utilizar este nuevo bloque:

PRÁCTICA A05.5:

En este último proyecto utilizando el sensor de luz, imitaremos qué ocurre cuando por la noche las farolas de la calle se encienden. Por tanto, utilizaremos un LED y el sensor LDR.

- Ejemplifica lo que sucede por la noche con las farolas de la calle. Es decir, cuando la intensidad de la luz sea inferior o igual a un 5%, que el LED se encienda. Por el contrario, cuando sea superior al 5%, el LED se apaga.



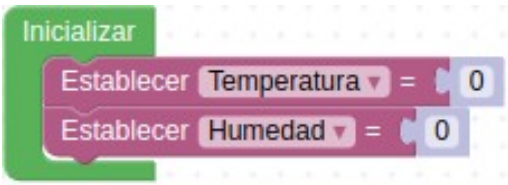
A06: Sensor DHT11

En esta sexta práctica aprenderemos a programar el Sensor de temperatura y humedad o cómo se llama en ArduinoBlocks: DHT11. Este sensor se conecta a un puerto digital.

El sensor DHT11 realmente es un sensor de temperatura y humedad. Tiene una salida de señal digital, que funciona en un rango de temperaturas entre 0 y 50 °C, con un error de 2ºC y un rango de humedad entre 20 y 90%, con un error de un 5%.

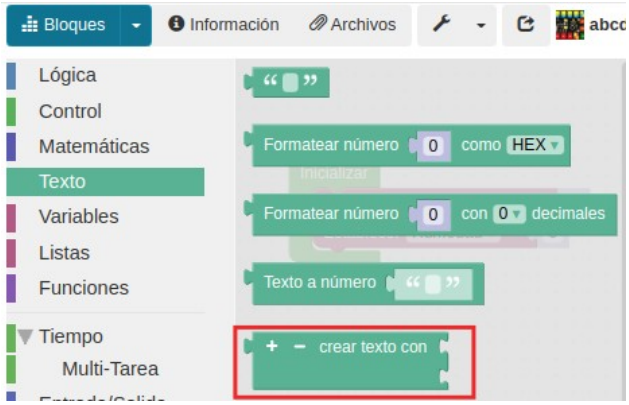
En el apartado de bloques de programación, se encuentra en "Sensores".

Antes de empezar la práctica, establecemos dos variables nuevas: una para la temperatura y otra para la humedad.

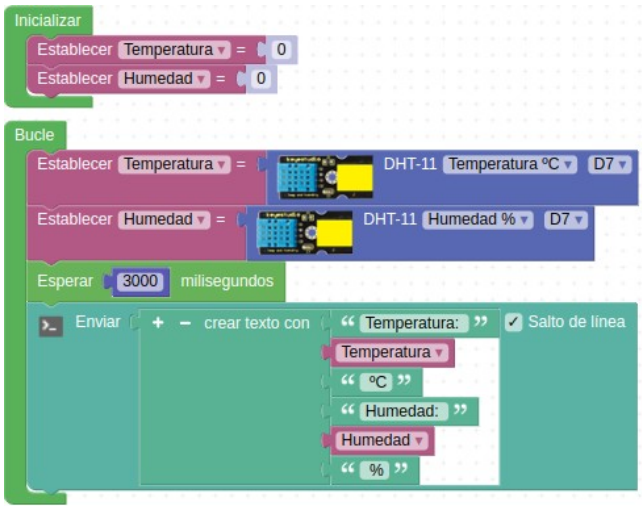


PRÁCTICA A06.1:

Esta primera práctica enviaremos los valores a la consola, es decir, los grados centígrados y el porcentaje de humedad de la habitación donde estamos. Recuerda y utiliza el bloque de programación "crear texto con", el cual te permite añadir diferentes mensajes en una misma línea de texto.

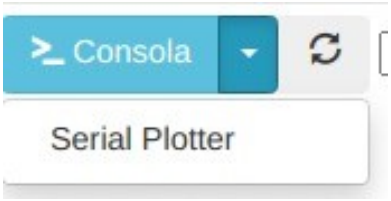


- Envía a la consola los valores de temperatura y humedad cada 3 segundos. En la consola debe aparecer:
  - Temperatura: X ºC
  - Humedad: X %

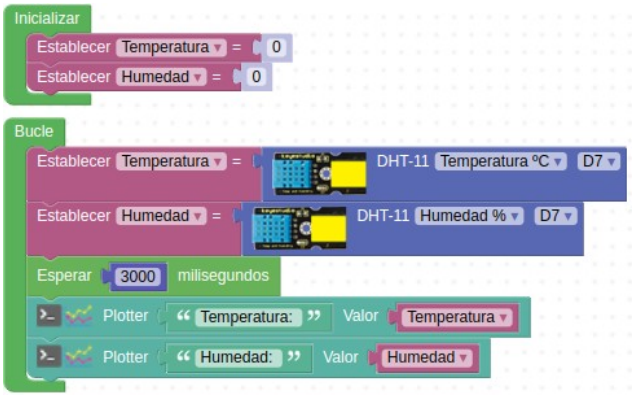


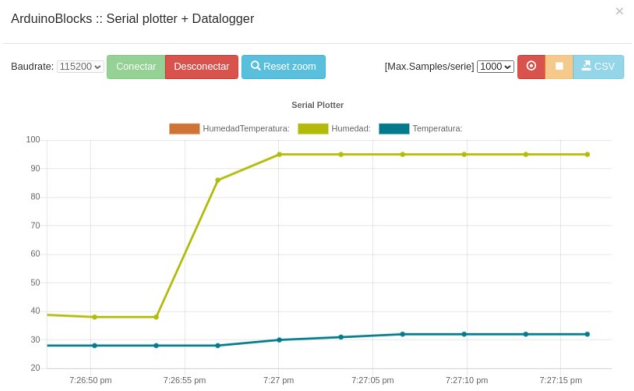
PRÁCTICA A06.2:

Seguidamente, trabajaremos con la herramienta Plotter. Esta es parecida a la consola, de hecho se encuentra en el desplegable de consola. Esta herramienta crea en tiempo real una estadística de los valores obtenidos. Para programarlo, lo encontraremos en el apartado de Puerto serie.



- Envía cada 3 segundos al Serial Plotter los valores de temperatura y humedad.

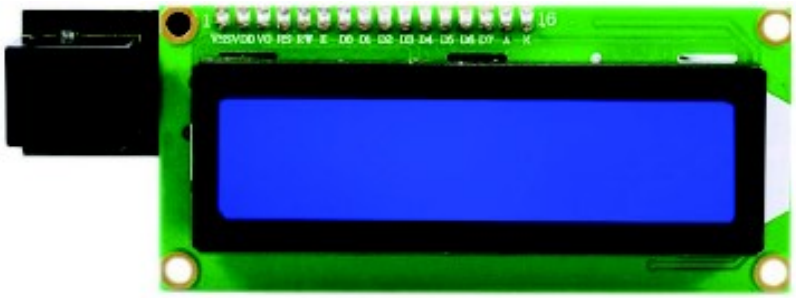




PRÁCTICA A06.3:

Otra forma de poder visualizar valores es haciendo uso de pantallas externas, es decir, pantallas conectadas a la placa Easy Plug. Tal y como podemos observar en el apartado de programación tenemos tres pantallas:

- Pantalla LCD: Es una pantalla LCD de 16 caracteres por 2 líneas con fondo azul y luz blanca. Se conecta al puerto I2C.



- Pantalla OLED: El nombre es una abreviatura de “Organic Light-Emitting Diode” que se traduce como "Diodo Emisor de luz orgánico". Una pantalla OLED es una matriz de LEDS orgánicos que se iluminan cuando emiten energía.

- LedMatrix 8x8: Es una matriz de de 8x8 LEDS.

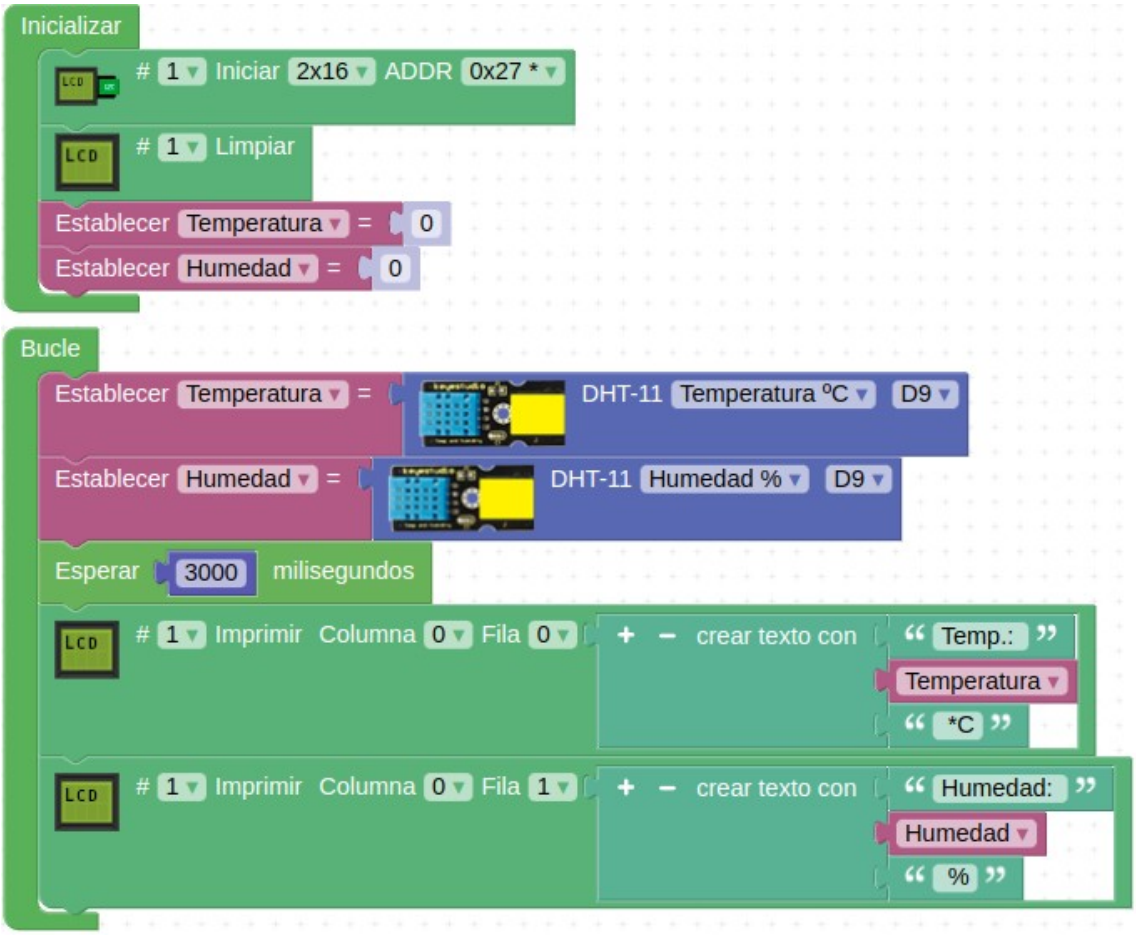
En esta práctica A06.3 enviaremos los mismos datos que hemos estado enviando a Consola y al Serial Plotter a nuestra pantalla LCD. Así pues, debes conectar la placa al puerto I2C.

Los bloques de programación de la pantalla son los de la imagen siguiente.

Hay que seleccionar el primero, y colocarlo en el bloque "Inicializar".

También aparecen los bloques que nos servirán para enviar la información a la placa. Es importante que establezcas qué fila quieres que se envíe la información. El número 0, ya cuenta como una opción.

- Envía a la pantalla LCD los valores de temperatura y humedad cada 3 segundos.

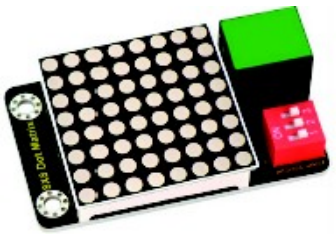


A08: Matriz de 8x8 LEDs

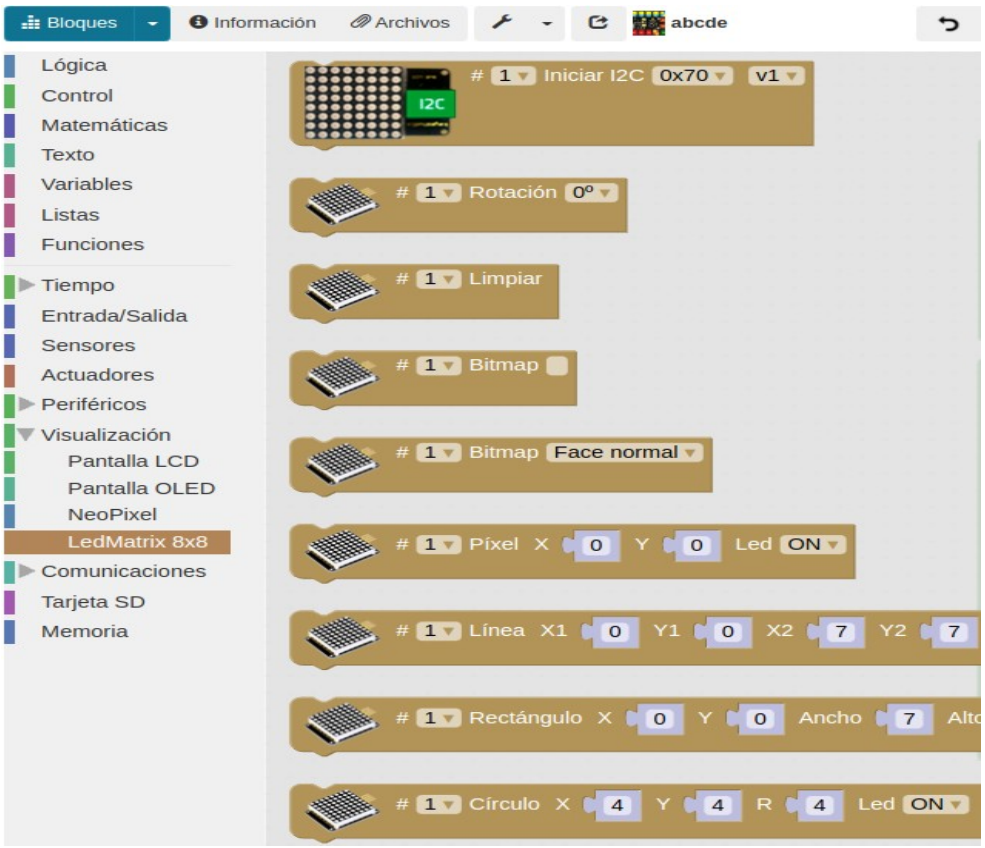
En esta práctica el objetivo es conocer la LEDMatrix 8x8 o también llamada matriz de LEDs.

La matriz de LEDs es una pantalla pequeña que tiene 64 LEDs y se conecta al puerto de comunicación I2C.

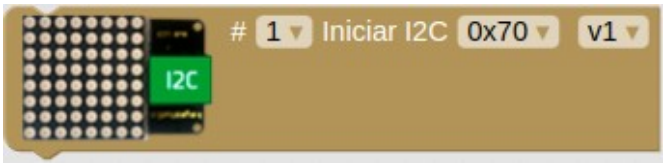
En esta pantalla podemos programar diferentes símbolos o elementos, como: caras, iconos, letras... Hay opciones prediseñadas desde ArduinoBlocks y también, existe la opción de crearlos personalizados.



En el apartado de bloques de programación, se encuentra en "LedMatrix 8x8". Existen diferentes opciones de programación, según nuestro objetivo:

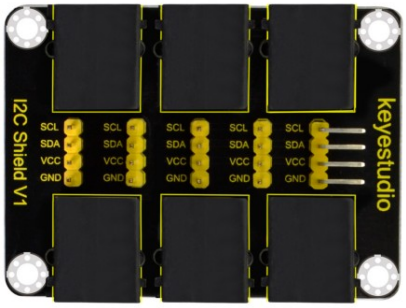


La primera tarea que debemos realizar cuando queremos hacer uso de la pantalla, es inicializarla. El primer bloque vemos que pone: inicializar I2C.



Hay una pestaña que nos deja escoger distintos números. Esto significa que podemos utilizar ocho diferentes haciendo uso de un Hub I2C.

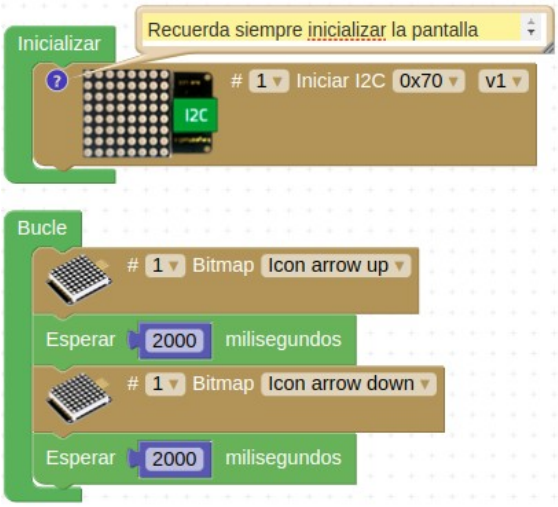
Un Hub sería como un "ladrón", que nos permite conectar varios dispositivos a la vez. Éste se conecta al puerto I2C.



PRÁCTICA A07.1:

En esta práctica introduciremos el uso de la LedMatrix. Lo que haremos es enviar un programa sencillo:

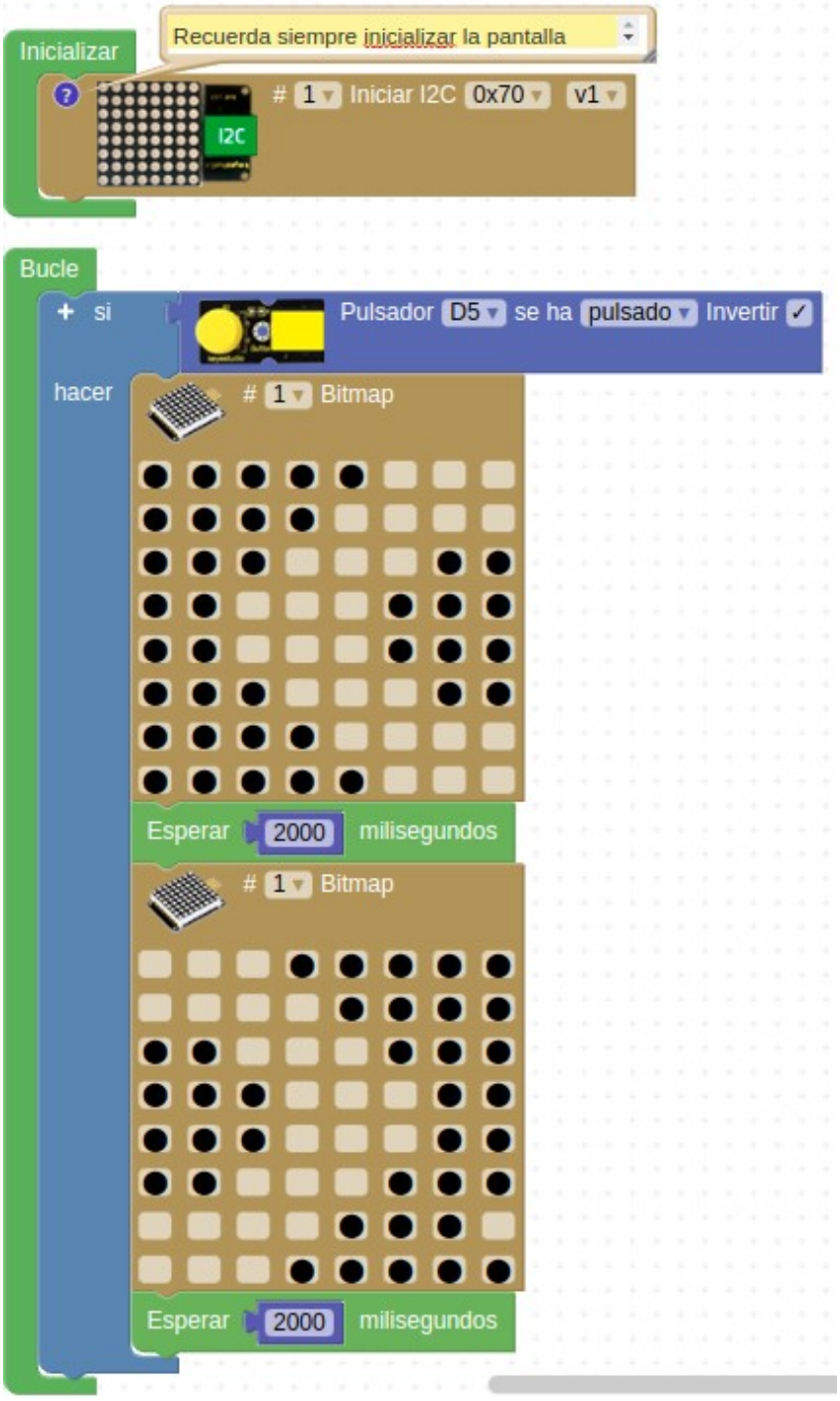
- La pantalla LedMatrix muestra una flecha hacia arriba, y después de dos segundos, muestra una flecha hacia abajo.



PRÁCTICA A07.2:

En esta práctica introduciremos otra forma de diseñar símbolos en la pantalla y también haremos uso de un pulsador.

- Cuando se pulse el pulsador, en la pantalla debe aparecer un símbolo personalizado durante dos segundos y después otro distinto.



PRÁCTICA A07.3:

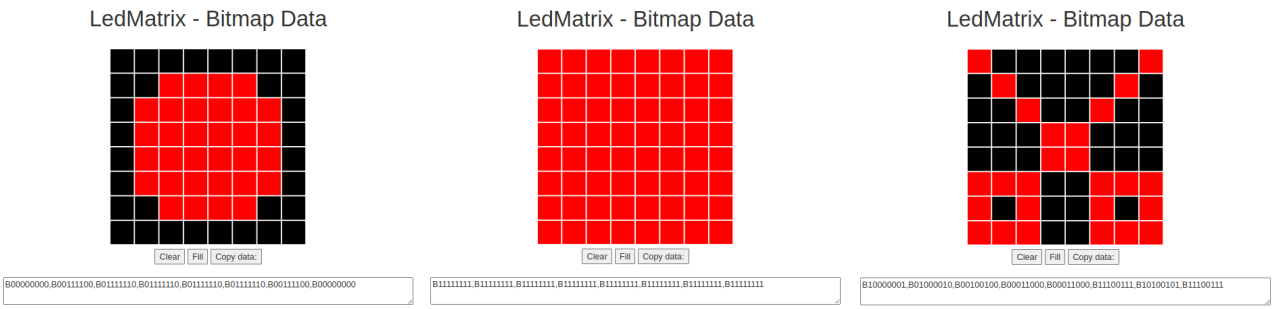
En esta tercera práctica recrearemos el famoso y conocido juego "Piedra, papel, tijera". Esta vez, aprenderemos otra forma de crear nuestros símbolos personalizados. También utilizaremos el pulsador para activar el juego y aprenderemos a crear una variable que elija aleatoriamente uno de los tres símbolos del juego.

- Cuando se pulse el pulsador, la pantalla muestra, de forma aleatoria, uno de los tres símbolos: piedra, papel o tijeras.

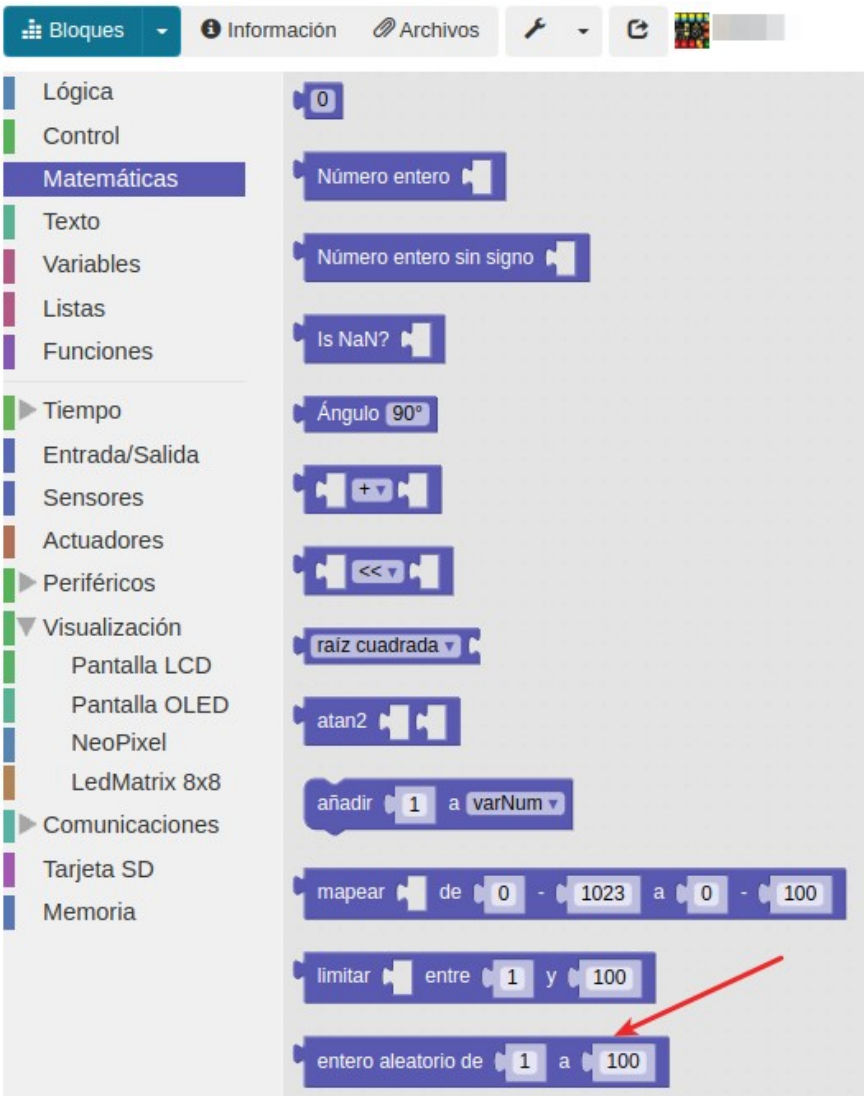
Lo primero que vamos a hacer es crear los tres símbolos: utilizando el "Bitmap". Para seleccionar los LEDs que queremos en estado ON, debemos dar los siguientes pasos:

1. Nos dirigimos a herramientas y escogemos según la imagen siguiente. También podemos hacer clic en el botón derecho del ratón sobre el bloque de programación, escoger la opción "ayuda". De cualquiera de las formas se nos abre otra pestaña en la que podemos ver un simulador de la pantalla.

2. Seleccionamos cada LED que queremos encender, así se nos devolverá de color rojo.
3. Por último, seleccionamos "Copy data" y dentro del cuadradito que está al lado de "bitmap", hacemos Ctrl+V y se nos engancha en código binario los LEDs que hemos seleccionado.
4. Creamos, las imágenes que vemos a continuación:



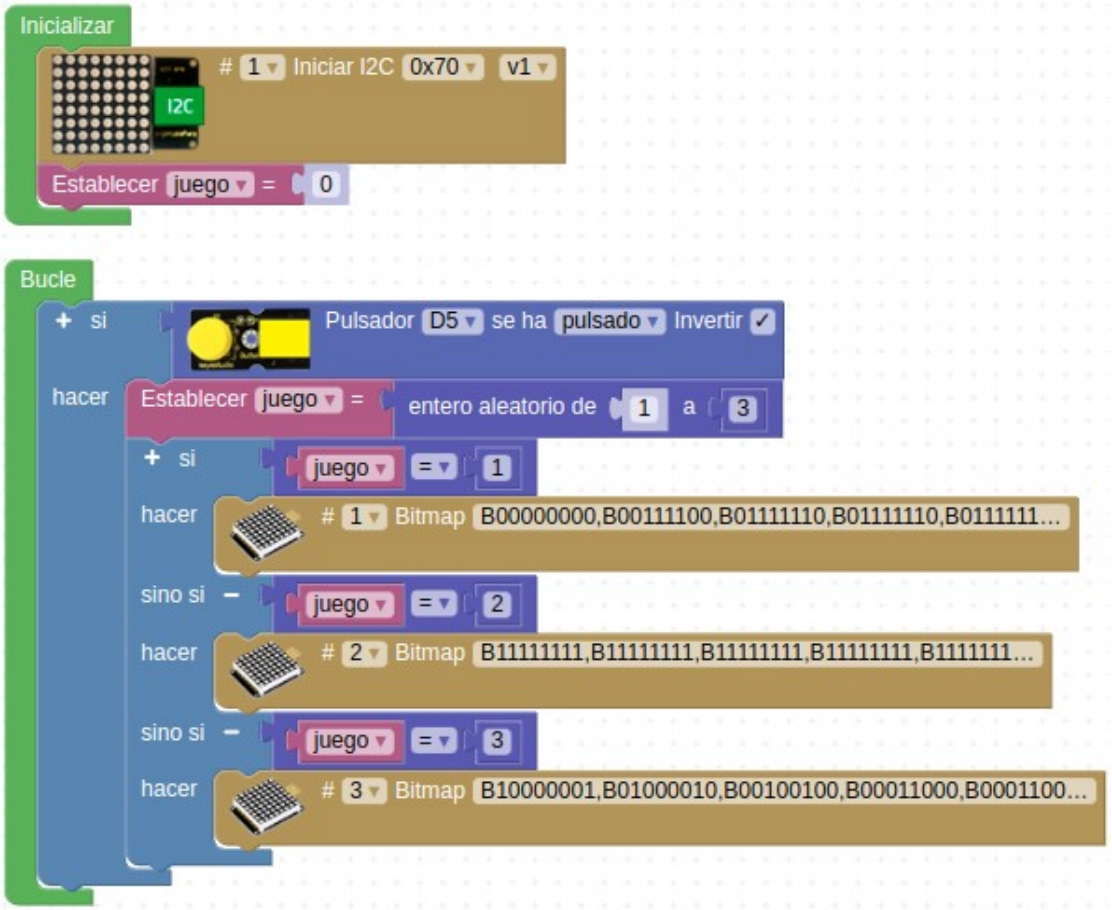
Para realizar esta práctica, vamos a necesitar crear una variable que llamaremos "Juego". Esta variable le asignamos tres números aleatorios, del 1 al 3. De este modo, asignaremos a cada número un símbolo. Es decir, que 1 es piedra, 2 es papel y 3 es tijeras. El bloque de programación: "entero aleatorio de x a x" lo encontramos en "Matemáticas". Éste nos permite escoger aleatoriamente el rango de números que nosotros establecemos.



Así pues, crearemos la variable "juego" estableciendo estos valores de 1 a 3.

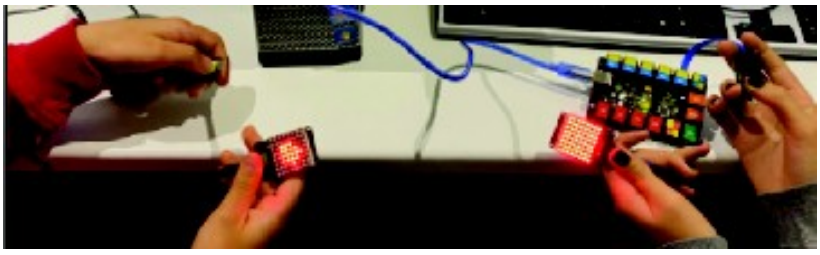
Una vez tenemos la variable creada y los símbolos, debemos seleccionar la opción de condicionales en el bloque de lógica, para establecer que si la variable "juego" es igual a 1, sea piedra. En cambio, si es igual a 2, sea papel o si es igual a 3 sea tijeras.

A continuación puedes ver la programación para elaborar el juego.



De este modo, cada vez que el jugador pulse el pulsador, el programa de forma aleatoria escogerá un número aleatorio entre 1, 2 y 3. Cuando lo elija, la pantalla mostrará el símbolo que corresponde al número seleccionado aleatoriamente.

En la imagen se muestran dos jugadores con el mismo programa.



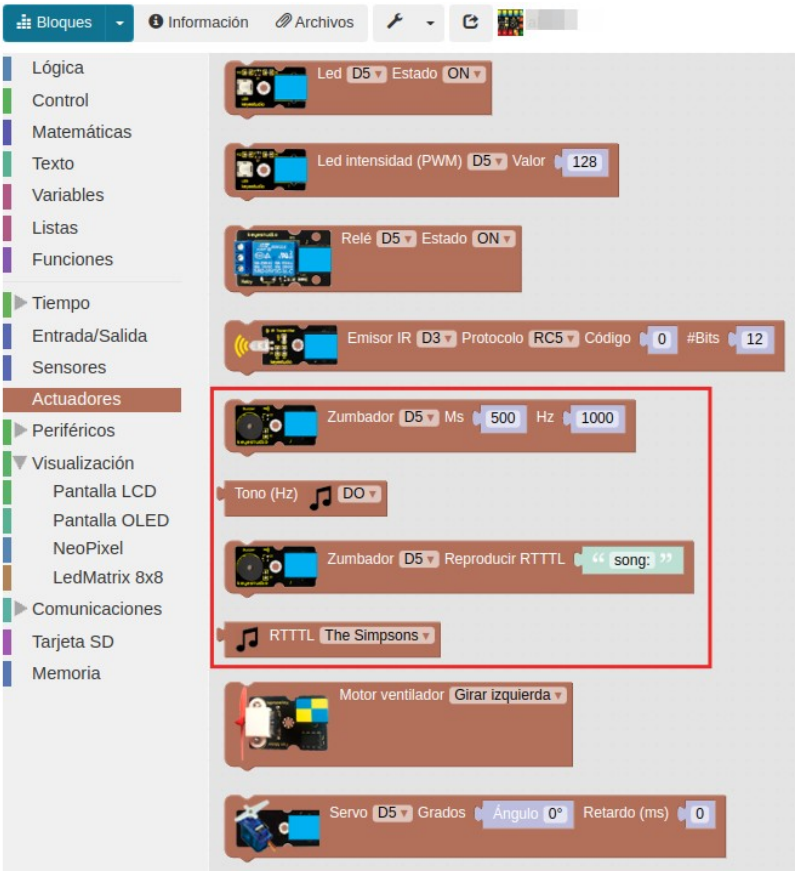
A08: Zumbador pasivo

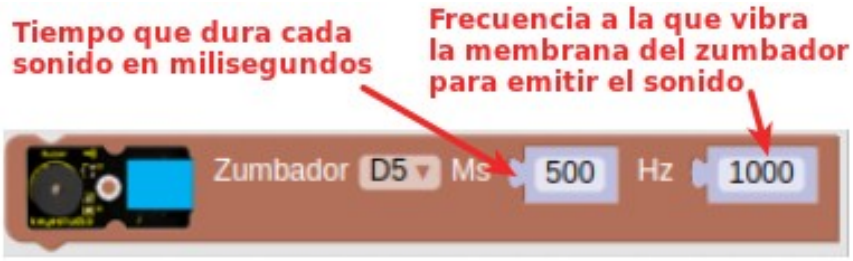
En la siguiente actividad trabajaremos con el zumbador, el cual es un actuador que se conecta a un puerto digital.

Este zumbador es pasivo, ya que no puede ser accionado por él mismo sino por pulsos externos de diferente duración y frecuencia. Puede ser utilizado como un pequeño "altavoz" que puede generar melodías musicales.

El sonido que emite el zumbador depende de la frecuencia de emisión del sonido. La frecuencia se entiende como el número de repeticiones por unidad de tiempo. El sonido se transmite en forma de onda. Por tanto, la frecuencia de un sonido es el número de oscilaciones por segundo.

En el apartado de bloques de programación, se encuentra en "actuadores". Hay diferentes opciones de programación, concretamente 4 bloques específicos, desde seleccionar una canción ya creada o decidir notas concretas.





PRÁCTICA A08.1:

En esta práctica crearemos la escala musical. Para ello puedes tener en cuenta la siguiente tabla.

Nota	Frecuencia	Nota	Frecuencia	Nota	Frecuencia
Do	261.6	Fa	349.2	La	440
Do#	277.2	Fa#	370	La#	466.2
Re#	293.7	Sol	392	Si	493,2
Mi	329.6	Sol#	415,3	Do	523,3

- Crea la escala de "Do"

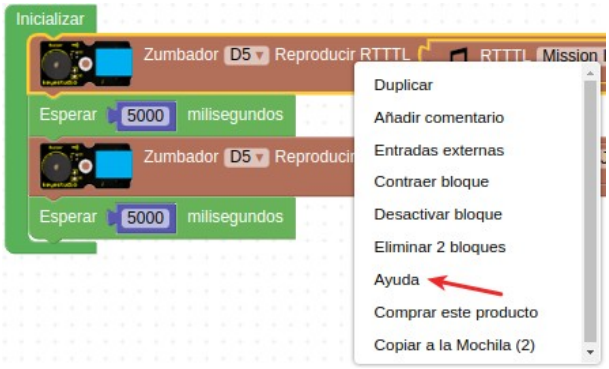
PRÁCTICA A08.2:

En esta práctica enviaremos al zumbador, una música ya diseñada por el programa.

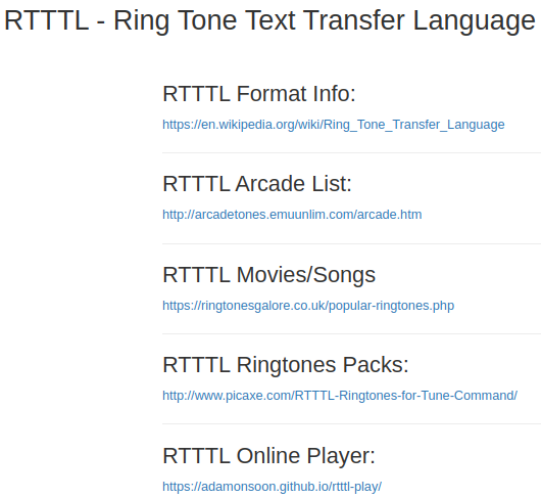
- El zumbador reproduzca una canción y tras una espera de 5 segundos, reproducirá otra.

¡Ahora puedes probar todas las canciones que quieras de la plataforma ArduinoBlocks!

Puedes buscar aún más canciones puedes hacer clic con el botón derecho sobre el bloque y seleccionando "ayuda".



Se muestran enlaces a distintas webs donde obtenerlas.



A09: Sensor PIR

En esta práctica, el objetivo es conocer el sensor de movimiento, también llamado PIR.

El sensor PIR es un sensor de movimiento, que tal y como dice su nombre puede detectar señales infrarrojas provenientes de una persona, animal u objeto en movimiento.

En el se basan las alarmas que detectan movimiento.

En el apartado de bloques de programación, se encuentra en "Sensores".

PRÁCTICA A09.1:

En esta práctica utilizaremos el sensor PIR y un LED.

- Si detecta movimiento que se encienda el LED sino que se apague.

PRÁCTICA A09.2:

En esta práctica reproduciremos una alarma. Así pues, utilizaremos el sensor PIR, y el zumbador.

- Si detecta movimiento que el zumbador emita un sonido similar a una alarma. Para ello, deberá emitir los dos tipos de sonido que se observan durante 500 milisegundos cada uno, repitiendo ciertas veces.

A10: Potenciómetro

En esta práctica el objetivo es conocer el potenciómetro.

El potenciómetro es un sensor de rotación analógico. Su voltaje se puede subdividir en 1024 o también puede dar valores en tanto por ciento.

En el apartado de bloques de programación se encuentra en “Sensores”.

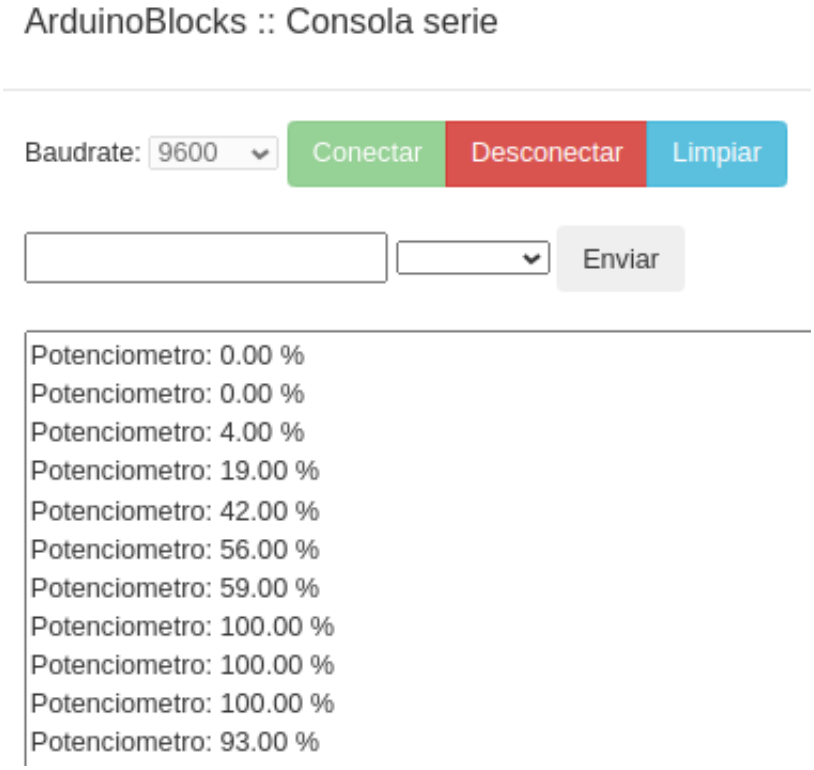
PRÁCTICA A10.1:

En esta práctica conoceremos cómo funciona el potenciómetro. Los valores que se recogen pueden calcularse en % o en numeración hasta 1023 (combinación binaria). En esta práctica observaremos los parámetros y el movimiento del potenciómetro a través de la consola.

Por tanto, crearemos una variable que se llame "potenciómetro" y la asociaremos al sensor potenciómetro.

- Que la consola muestre cada segundo cuál es el valor del potenciómetro.

Estos son los valores que muestra la consola serie, 0% (izquierda) para el potenciómetro en un extremo y 100% (derecha) para el otro.

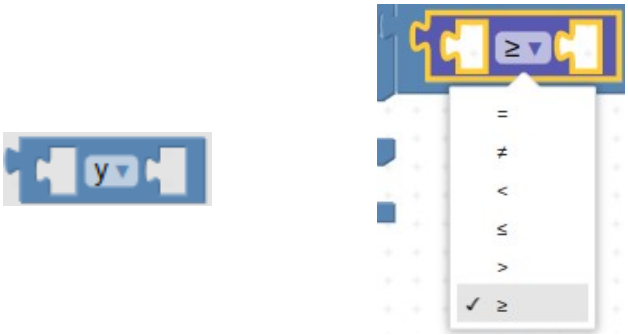


PRÁCTICA A10.2:

Ahora que conocemos cómo describir el valor que recoge el potenciómetro lo programaremos combinando con un zumbador:

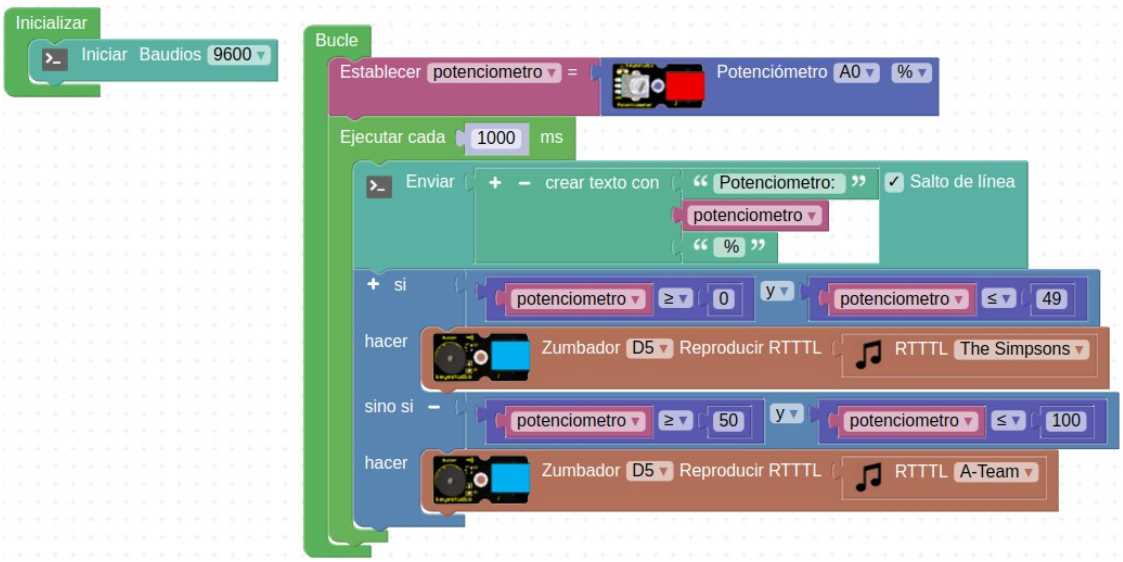
- Si el potenciómetro está entre 0 y 49% emitirá un sonido, en cambio, si está entre 50% y 100% emitirá otro sonido distinto.

Por tanto, ahora haremos uso de la función lógica "y" o “AND” y combinarlo con el de signos de mayor y menor. Es decir, estos dos bloques:



Vamos a programar el intervalo de 0 a 49% mediante una función AND y preguntando si potenciómetro es mayor o igual que 0 “Y” menor o igual que 49. Para el otro intervalo preguntamos si es mayor o igual que 50 “Y” menor o igual que 100.

En este programa es interesante visualizar lo que ocurre por la consola, pues así podemos visualizar el cambio de porcentaje. La programación final sería la siguiente:



PRÁCTICA A10.3:

En esta tercera práctica utilizaremos el potenciómetro para regular la intensidad luminosa de un LED. Recordemos que en la práctica 2, estuvimos trabajando el PWM. En aquellas prácticas descubrimos que se medía su rango, entre los valores 0 y 255, es decir, un total de 256 valores.

A la vez, sabemos que el potenciómetro puede darnos el rango de valores de 0 a 100% o también de 0 a 1023. Escogeremos esta última condición y crearemos una variable de nombre potenciómetro. Pero esta vez dividiremos los valores que recoja entre 4, ya que si dividimos 1023 entre 4, nos sale 255. De esta forma podemos ver la similitud: 255 (máxima intensidad del LED) y 1023 (máximo valor del potenciómetro).

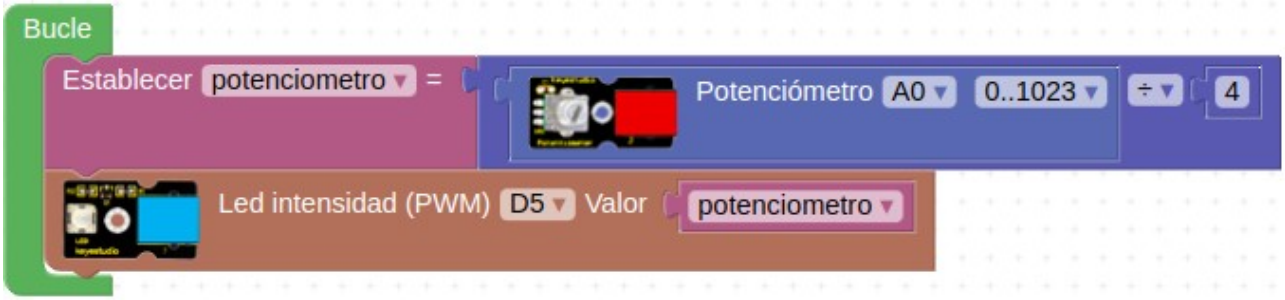
Empezamos creando la variable, para ello debemos ir al apartado de "Matemáticas" y seleccionar el de operadores. Éste tiene un desplegable y debemos escoger la división:

La variable que debe quedarnos esta vez es la siguiente:

Podemos comprobar esta nueva variable, enviando valores a la consola.

- Que a medida que vamos "girando" el potenciómetro, la intensidad del LED vaya subiendo.

Para hacer este programa vamos a utilizar el bloque “Led intensidad (PWM)” al que asignaremos el valor de la variable potenciómetro.

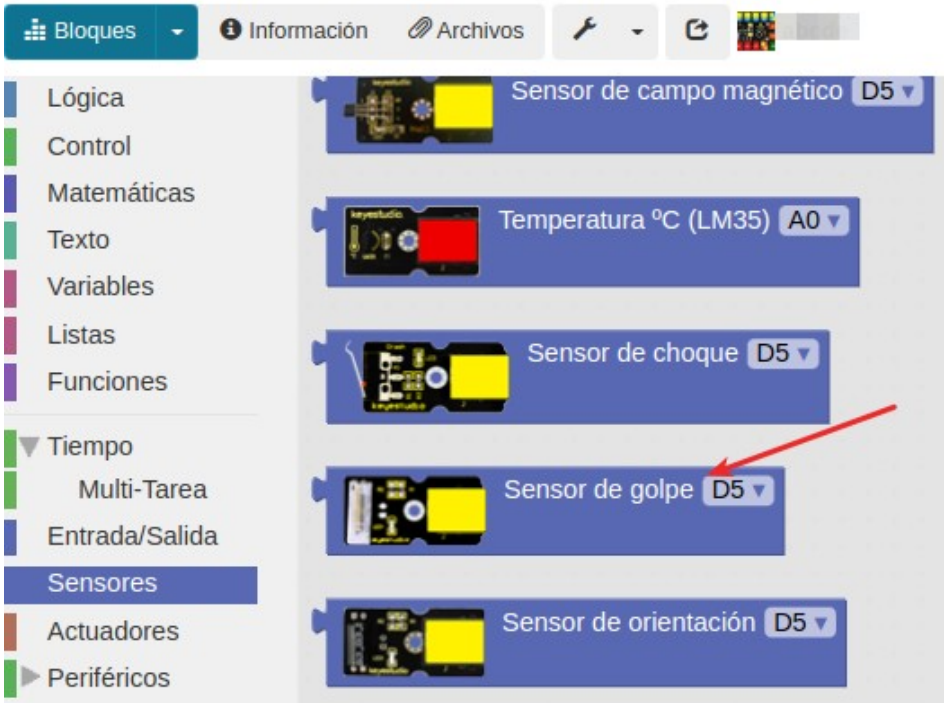


A11: Sensor de golpe

A lo largo de esta práctica aprenderemos a programar el sensor de golpe.

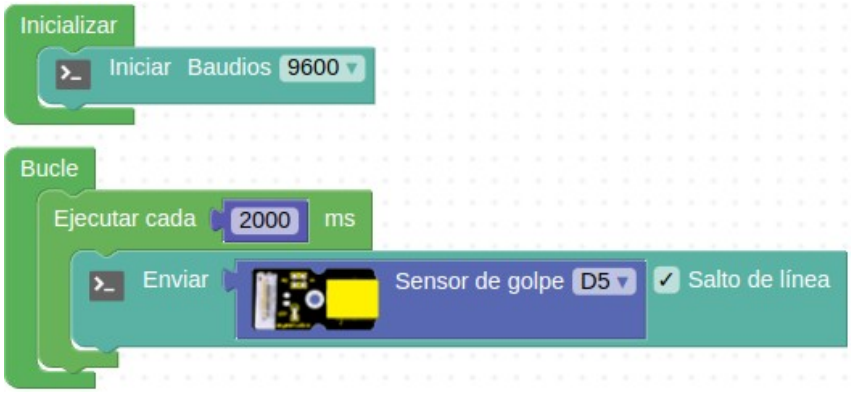
Es un sensor detector de golpes. Por tanto, cuando se golpea puede enviar una señal al instante. Se puede utilizar para sustituir el pulsador para encender un LED cuando golpeamos el sensor o también que cuando lo golpeamos se emita un sonido.

En el apartado de bloques de programación, se encuentra en "Sensores".



PRÁCTICA A11.1:

Al ser un sensor nuevo, antes de nada lo que haremos será "leerlo". Es decir, enviaremos sus valores a la consola.



Este sensor da cómo salida 1 y de forma instantánea, cuando se golpea, un 0. Un programa como el anterior va a originar que en la consola siempre tengamos el mismo valor: 1. El "problema" es que es una señal tan rápida y débil que no podemos visualizar su valor en la consola.

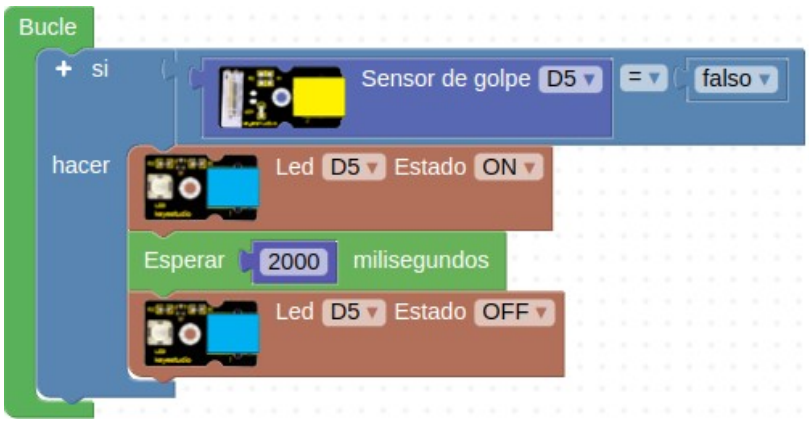
PRÁCTICA A11.2:

En esta práctica aprenderemos a encender un LED utilizando el sensor de golpe. Conectamos tanto el LED como el sensor de golpe a puertos digitales.

- Cuando damos un golpe, que el LED se encienda durante 2 segundos y que después se apague.

Sabemos que el sensor da 1 cuando no hay "golpe" y 0 cuando si hay "golpe". Por tanto, los valores de este sensor se mueve entre dos posibles: sí/no, verdad/falso, 1/0...

Así pues, en este programa debemos preguntar si el valor que devuelve el sensor de golpe es falso, es decir, que está a 0 y si es así que se encienda el LED. El programa es el siguiente:

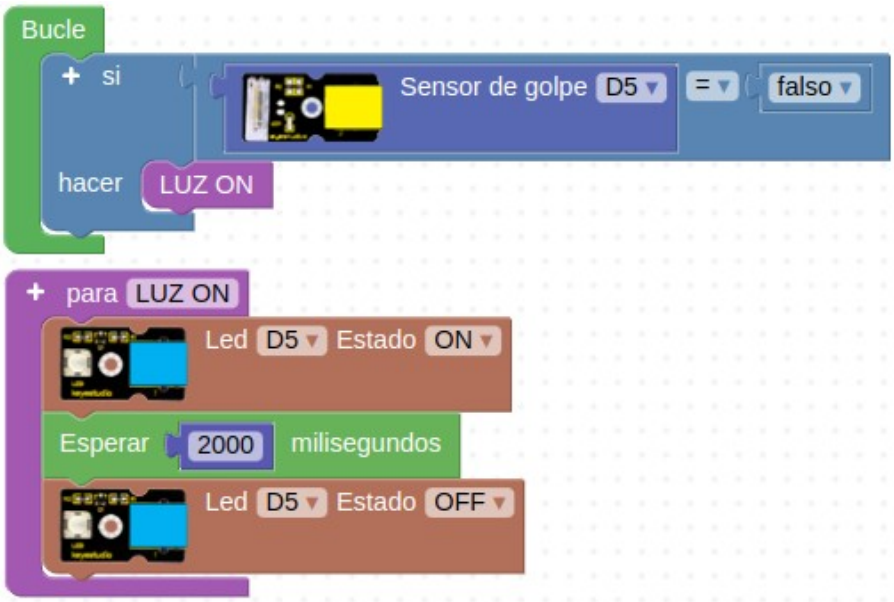


Siguiendo con esta práctica, realizaremos una pequeña modificación e introduciremos el concepto de "función".

ArduinoBlocks tiene un apartado llamado "funciones".

Para entender lo que es una función, podemos describirlo como un conjunto de instrucciones que las agrupamos bajo un nombre. Cuando creamos una función, automáticamente se genera un bloque de esa función y se puede insertar en cualquier momento del programa para invocarla. Es una forma de agilizar y hacer mas legible la programación y para que en programaciones más "grandes" nos quede todo un poco más ordenado.

En nuestro caso, pues, agruparemos la programación "LED ON, esperar 2000 milisegundos, LED OFF" en una función llamada LUZ ON.



A12: Sensor analógico de temperatura

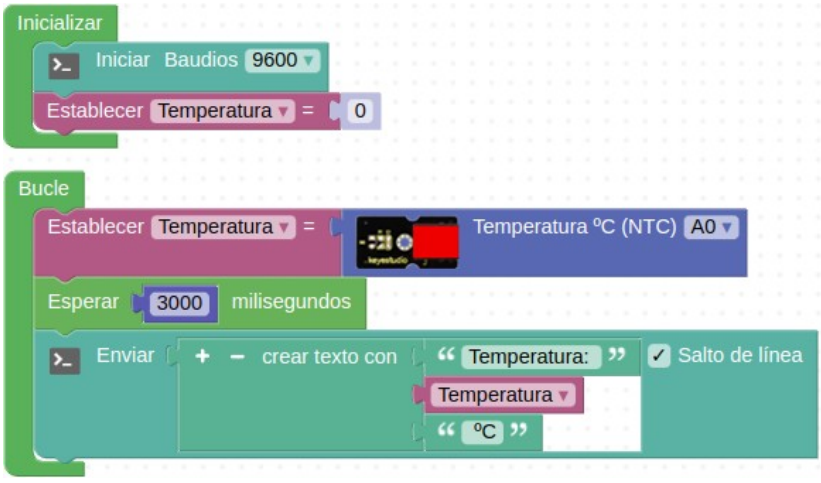
Un termistor es una resistencia cuyo valor depende la temperatura y es por tanto un sensor analógico que sirve para detectar temperatura.

El sensor analógico de temperatura está basado en un termistor y nos va a servir por tanto para medir la temperatura.

En el apartado de bloques de programación, se encuentra en "Sensores".

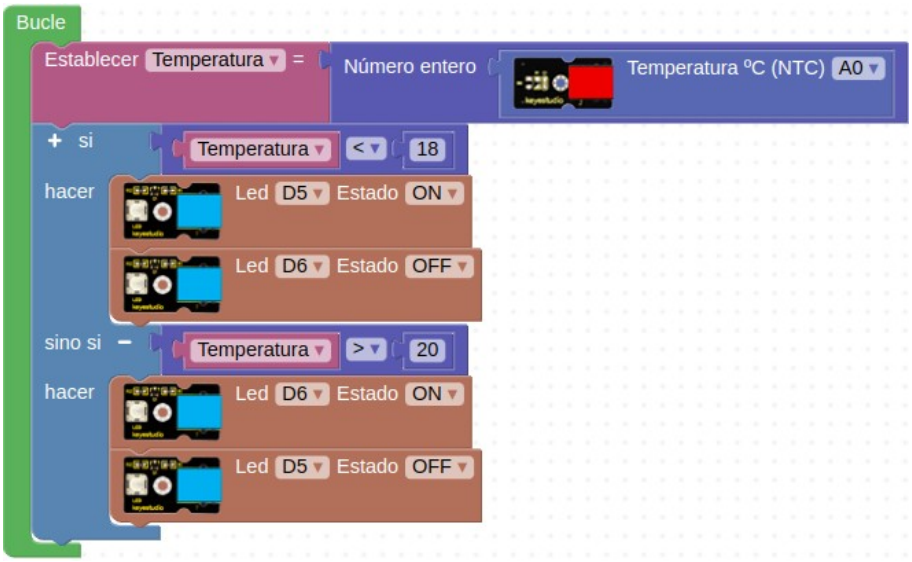
PRÁCTICA A12.1:

- Enviar a la consola serie la medida de temperatura cada tres segundos.



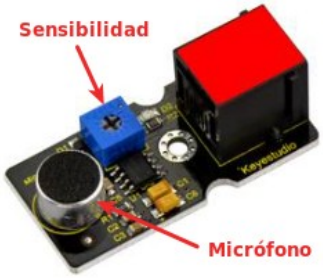
PRÁCTICA A12.2:

- Simular un termostato de forma que si la temperatura está por debajo de un valor de 18 ºC se encienda un LED que indica que se activa la calefacción y si está por encima de 20 ºC que se encienda un LED verde, se apague el rojo indicando que la estancia está a una temperatura superior a los 18 ºC.



A13: Sensor analógico de sonido

El sensor de sonido está dotado de un micrófono capaz de detectar el volumen de ruido ambiental. Se puede usar para hacer circuitos tipo interruptor operado por voz. Se trata de un sensor de tipo analógico.
La sensibilidad del sonido se puede ajustar mediante el potenciómetro.

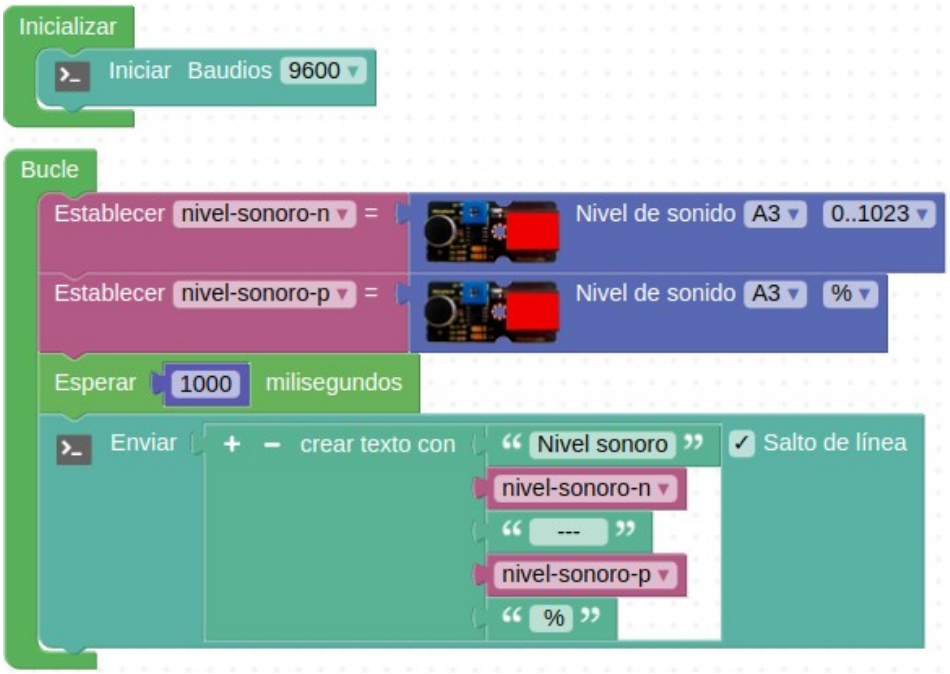


En el apartado de bloques de programación, se encuentra en "Sensores".

PRÁCTICA A13.1:

Lo primero que vamos a hacer es leer la salida del sensor y comprobar como cambia su sensibilidad al mover el potenciómetro.

- Lectura del sensor de sonido y ajuste de la sensibilidad mostrando los datos por consola tanto en valor numérico como en porcentaje.



PRÁCTICA A13.2:

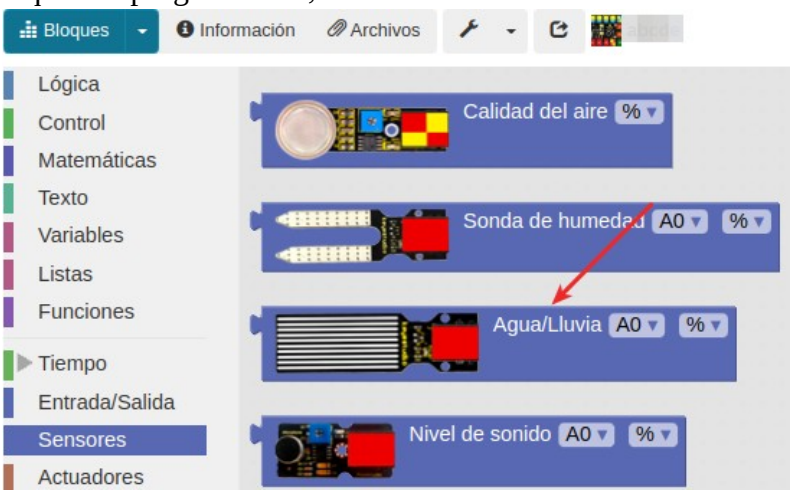
Usaremos el sensor de sonido como un interruptor.

- Cuando el nivel sonoro supere el 35% que se encienda un LED rojo.

A14: Detector de nivel agua

Este sensor está especialmente diseñado para identificar el nivel de agua y para detectar fugas de agua. Es posible medir el nivel a través de una serie de pistas expuestas. Realiza una conversión directa y proporcional entre la cantidad de agua y la salida analógica.

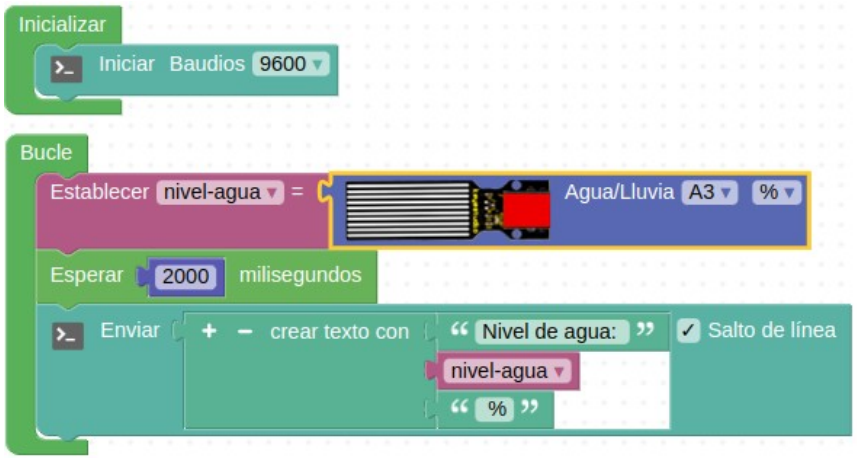
En el apartado de bloques de programación, se encuentra en "Sensores".



PRÁCTICA A14.1:

Vamos en primer lugar a leer los valores que da el sensor y mostrarlos por consola.

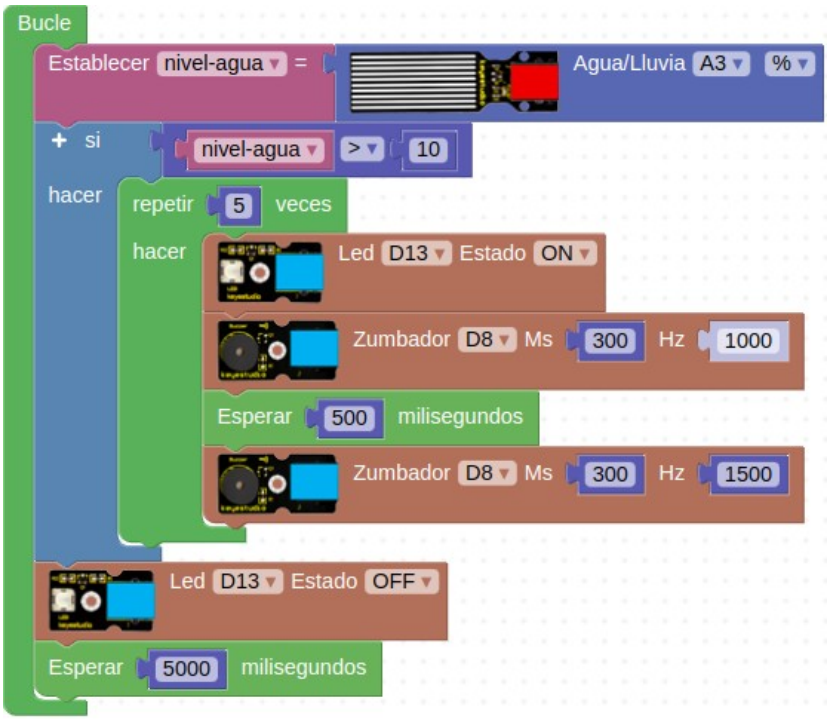
- Leer y mostrar los valores cuando el sensor está al aire y cuando se va introduciendo en el agua hasta llegar al máximo valor.



PRÁCTICA A14.2:

En esta práctica vamos a realizar una alarma óptico-acústica que nos indique que tenemos una fuga de agua.

- Cuando el sensor de agua detecte un nivel del 10% se accionará la alarma de manera intermitente cada 5 segundos.



A15: Sensor del nivel de humedad del suelo

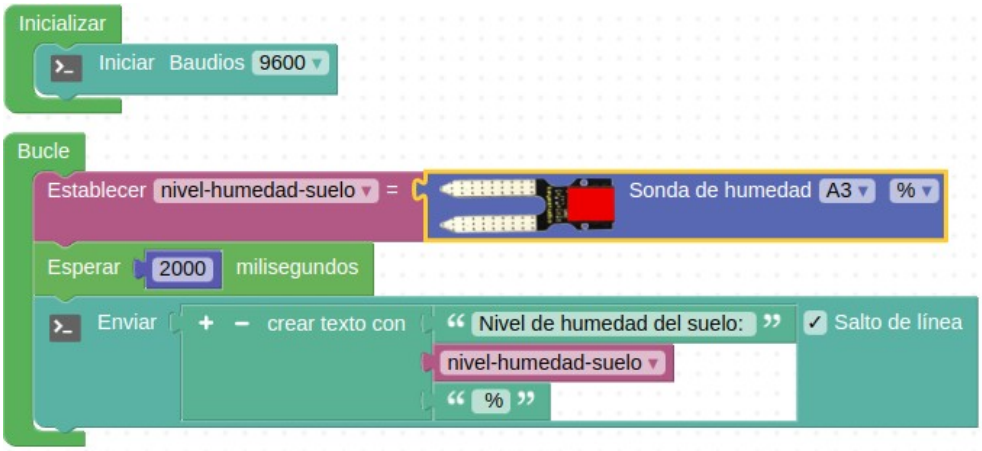
El sensor de humedad del suelo utiliza dos sondas para pasar corriente a través del suelo y leer la resistencia que sirve para obtener el nivel de humedad.
Si el suelo está muy húmedo habrá mas conducción de electricidad debido a la menor resistencia, mientras que un suelo seco conduce mal la electricidad porque presenta una resistencia mayor.
El dispositivo se puede usar para hacer un dispositivo de riego automático de plantas o jardines.

En el apartado de bloques de programación, se encuentra en "Sensores".

PRÁCTICA A15.1:

En esta práctica vamos a leer los valores que no entrega el sensor para distintos niveles de humedad.

- Leer y mostrar por consola el valor de humedad entregado por el sensor totalmente seco, en una tierra con poca humedad y en una tierra con mucha humedad.



A16: Sensor de campo magnético de efecto Hall

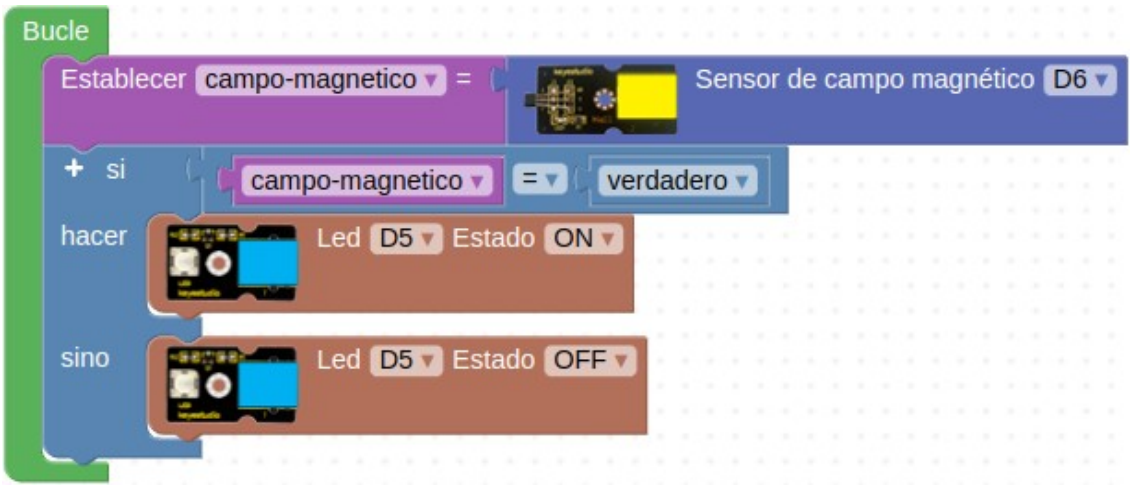
El sensor de campo magnético se basa en el efecto Hall y puede detectar sin contacto si existe un objeto que genera un magnético cerca de él. El resultado lo muestra a través de su salida digital.
El rango de detección y la fuerza del campo magnético son proporcionales.

En el apartado de bloques de programación, se encuentra en "Sensores".

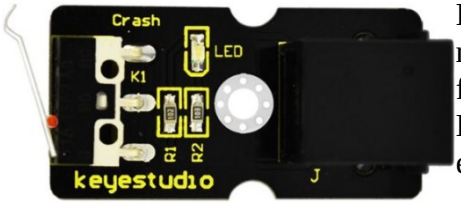
PRÁCTICA A16.1:

En esta práctica vamos a encender un LED cuando se detecta un campo magnético.

- Leer el valor booleano entregado por el sensor Hall y si es “1” encender un LED rojo y si es “0” mantenerlo apagado.



A17: Sensor de colisión



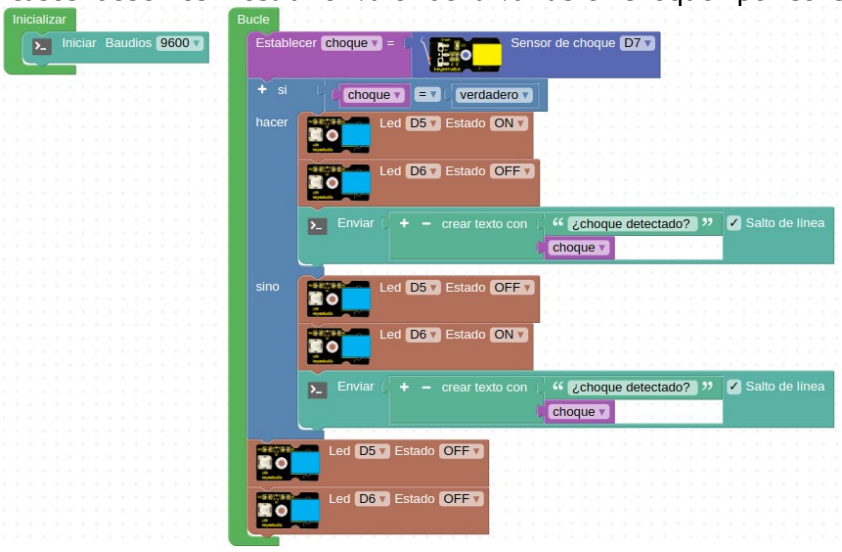
El sensor de colisión o choque, es un interruptor de acción rápida del tipo final de carrera que se activa con muy poca fuerza física.
Es un módulo de encendido y apagado digital necesario para la electrónica elemental.

En el apartado de bloques de programación, se encuentra en "Sensores".

PRÁCTICA A17.1:

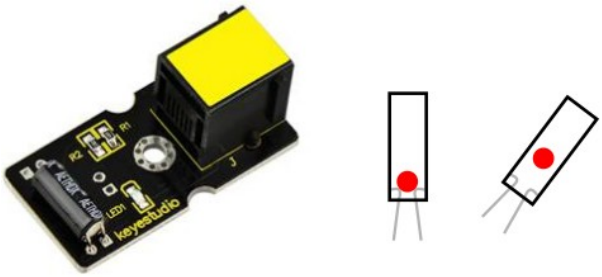
En esta práctica vamos a encender y apagar unos LEDs cuando se acciona el interruptor de choque al tiempo que mostramos por la consola serie el valor de la variable “choque”.

- Leer el valor booleano entregado por el sensor de choque y si es “1” encender un LED verde mientras el rojo se mantiene apagado y si es “0” apagamos el verde y encendemos el rojo. En ambos casos debemos mostrar el valor de la variable “choque” por consola.

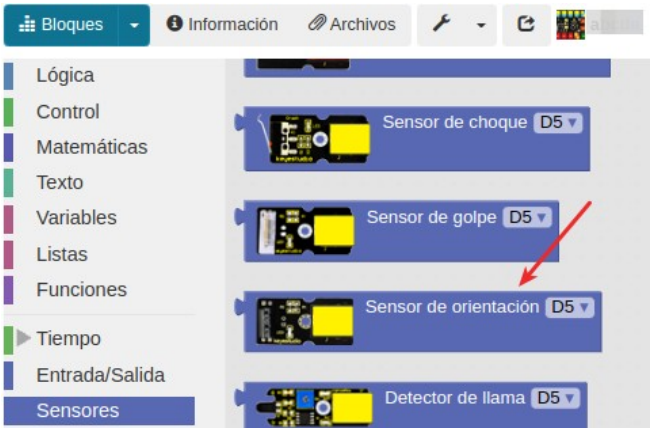


A18: Sensor de inclinación (tilt)

El interruptor de inclinación es el equivalente a un botón y se utiliza como entrada digital. Dentro del interruptor de inclinación hay una bolita metálica que hace contacto con los pines cuando el sensor está en posición vertical. Inclinando el sensor la bolita no toca los contactos, por lo que deja el circuito abierto. La idea es que cuando el sensor está horizontal está abierto y cuando se inclina se cierra, lo que permite detectar una determinada orientación. El aspecto y principio de funcionamiento lo vemos en la imagen siguiente.



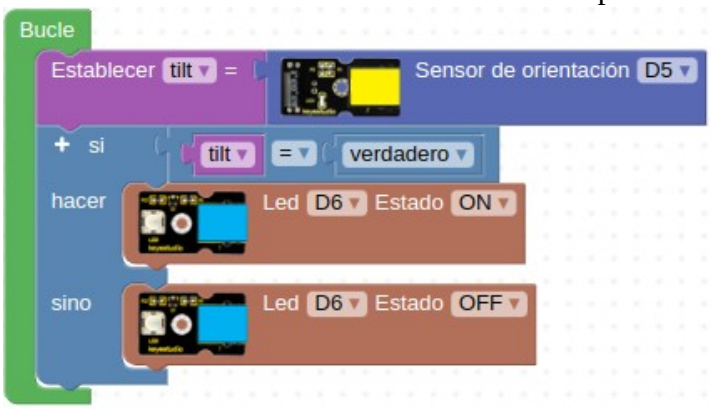
En el apartado de bloques de programación, se encuentra en "Sensores".



PRÁCTICA A18.1:

Vamos a detectar la posición no vertical del sensor de orientación.

- Encender un LED cuando el sensor de orientación no esté en posición vertical.



A19: Sensor de temperatura LM35

El LM35D es un sensor analógico de temperatura que tiene un rango de medida que va desde los -55 ºC hasta los 150 °C y una sensibilidad de 10 mV/ºC. Es decir, su salida es lineal y cada grado Celsius equivale a 10 mV, por lo tanto:

- 150 °C = 1500 mV
- -55 °C = -550 mV

Sus principales características son:

- Calibrado para dar la medición en Celsius
- Factor de escala lineal de ± 10 mV/°C
- Precisión de 0.5 °C (a 25°C)
- Adecuado para aplicaciones que trabajan en remoto

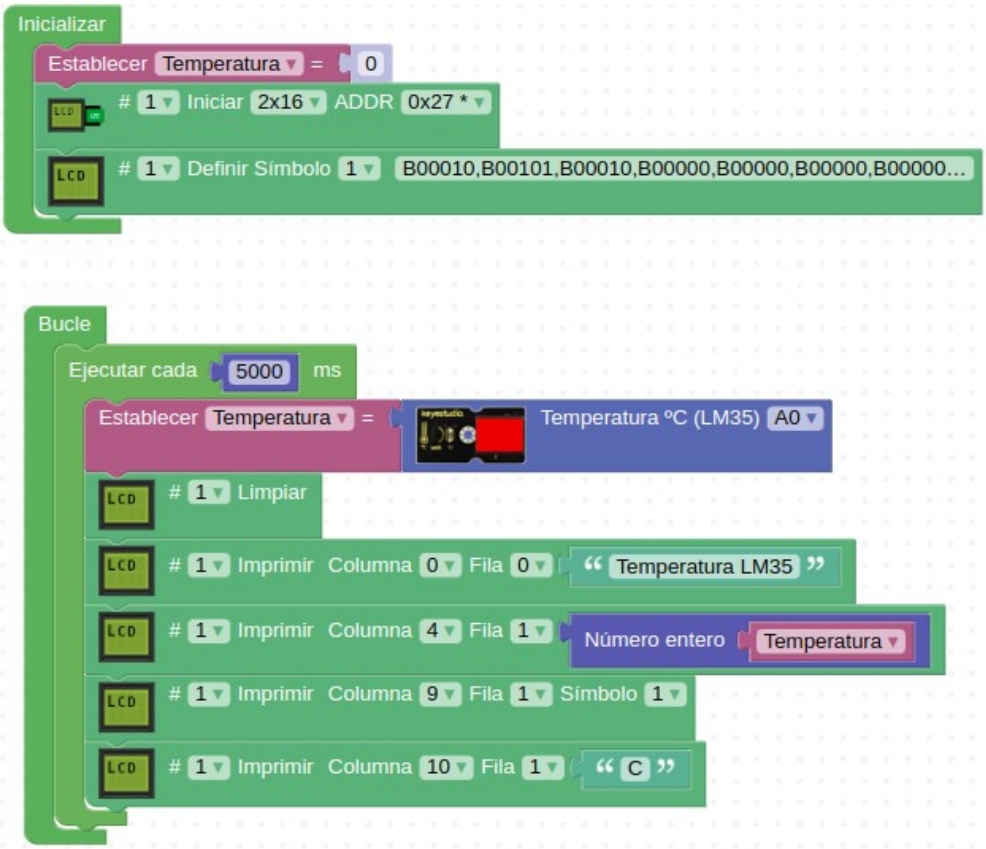
Su aspecto es:

En el apartado de bloques de programación, se encuentra en "Sensores".

PRÁCTICA A19.1:

Vamos a medir la temperatura ambiente de una habitación.

- Mostrar en una LCD la temperatura medida por un LM35.



A20: Sensor de llama

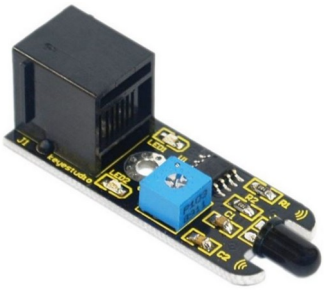
Es un sensor digital que puede detectar fuego o cualquier otra llama cuya luz tenga una longitud de onda entre 760 nm y 1100 nm, es decir, desde el rojo visible de las llamas hasta el infrarrojo de las mismas. Un robot contra incendios usará un detector de este tipo como sonda detectora de la fuente del fuego.

Sus principales características son:

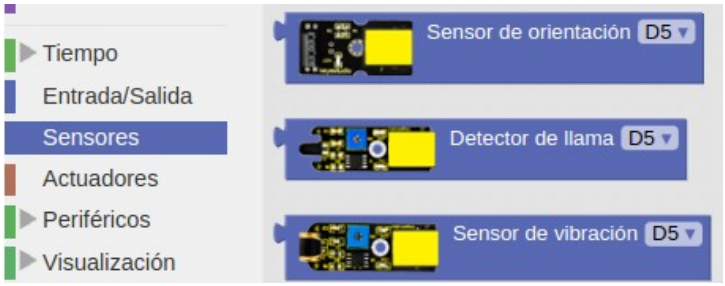
- Alimentación: 3.3V a 5V
- Ancho de banda espectral: 760nm a 1100nm
- Temperatura de trabajo: -25 °C a 85 °C

El potenciómetro sirve para ajustar la sensibilidad del sensor.

Su aspecto es:



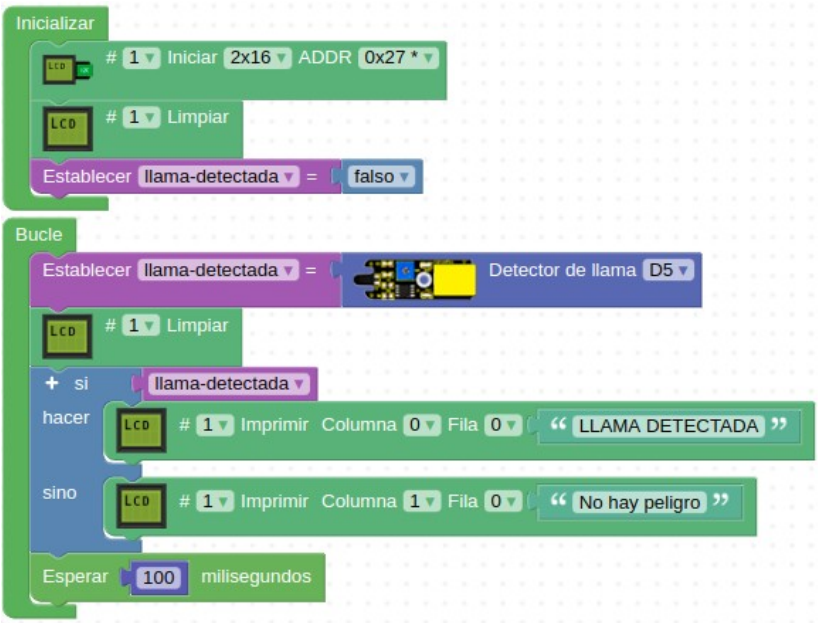
En el apartado de bloques de programación, se encuentra en "Sensores".



PRÁCTICA A20.1:

Vamos a detectar la llama de un encendedor a gas.

- Mostrar en una LCD el mensaje “LLAMA DETECTADA” cuando esto ocurra y mientras tanto el mensaje “No hay peligro”.



PRÁCTICA A20.2:

Se propone como actividad mejorar la anterior haciendo que cuando se detecta llama también se emite un sonido tipo sirena y que un LED parpadee.

A21: Sensor de luz ambiental TEMT6000

El sensor de luz ambiental TEMT6000 es un sensor analógico con mayor sensibilidad que la LDR. En teoría el TEMT6000 está adaptado según la sensibilidad del ojo humano. Es posible que con niveles de poca luz no sea muy sensible, y, al igual que el ojo humano, no reacciona bien a la luz infrarroja o ultravioleta.

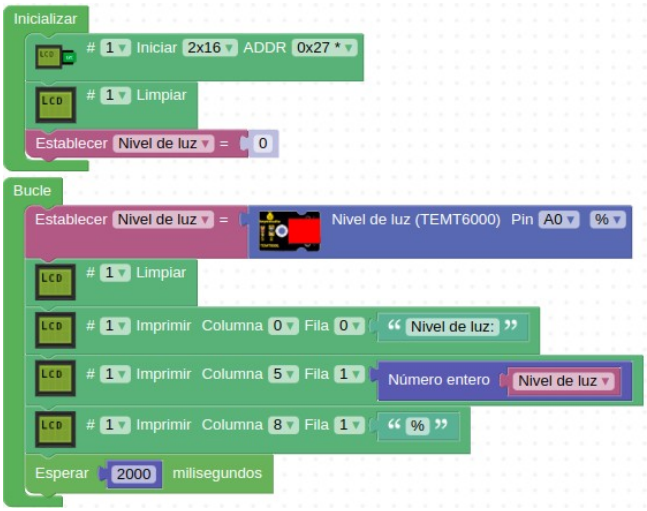
Su aspecto es:

En el apartado de bloques de programación, se encuentra en "Sensores".

PRÁCTICA A21.1:

Vamos a medir el nivel de luz ambiente de una habitación.

- Mostrar en una LCD el nivel de luz ambiental de una habitación expresado en %.



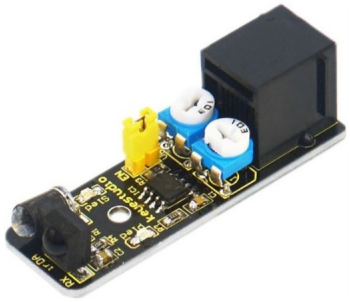
A22: Sensor infrarrojo para evitar obstáculos

Se trata de un sensor digital que entrega una lectura de nivel TTL. El sensor está equipado con función de ajuste de distancia con una gran adaptabilidad a la luz ambiental y es de alta precisión. Tiene un diodo de transmisión y un fototransistor de recepción de infrarrojos. Cuando el rayo infrarrojo lanzado por el LED transmisor encuentra un obstáculo el rayo se refleja y el fototransistor lo recibe, siendo esto procesado por un circuito comparador que detectará el obstáculo y el LED indicador de la placa se iluminará. Cuando se usa para detectar líneas básicamente el rayo emitido se refleja si el color es blanco y no hay reflexión si el color es negro puediendo así detectar estos colores de línea.

Se puede ajustar la distancia de detección con las dos resistencias variables de que está equipado en un rango efectivo entre 2 y 40 cm.

Básicamente se puede utilizar para evitar obstáculos y para robots seguidores de línea bien en color blanco o bien en color negro.

Su aspecto es:

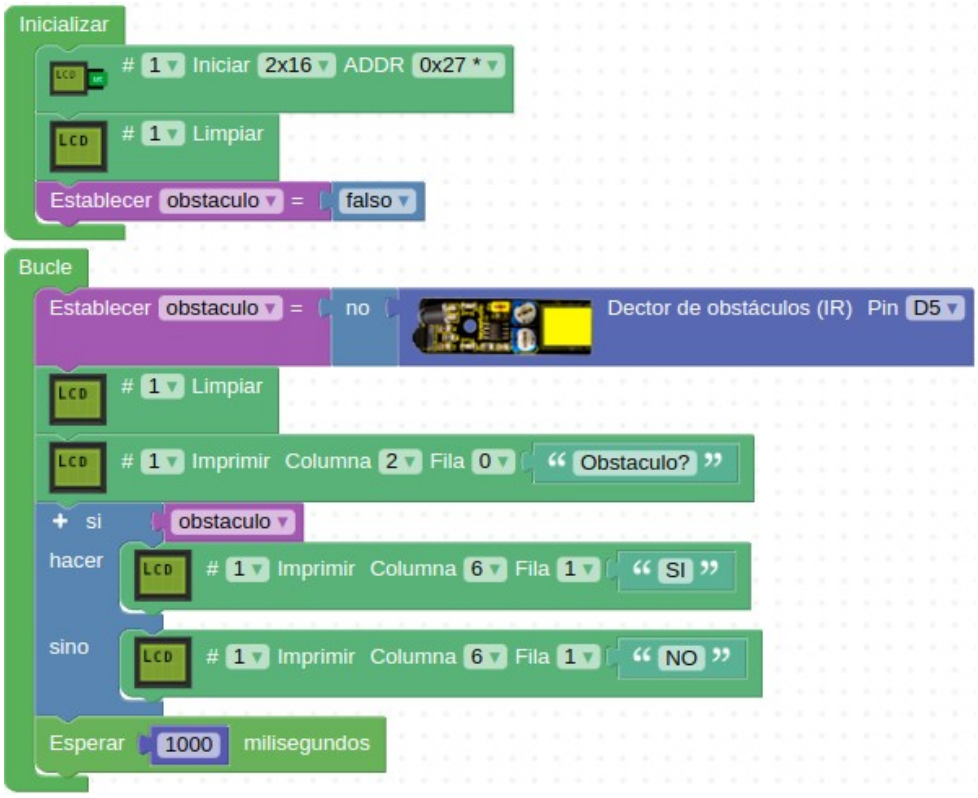


En el apartado de bloques de programación, se encuentra en "Sensores".

PRÁCTICA A22.1:

Vamos a realizar un detector muy básico de obstáculos.

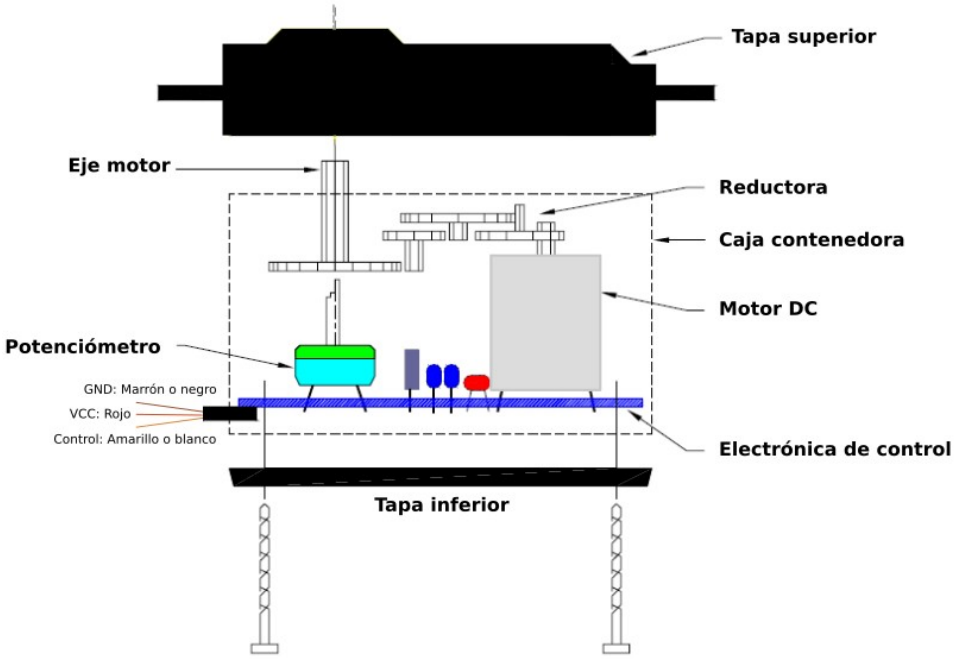
- Mostrar en una LCD el texto ¿Obstaculo? Y el mensaje SI o NO en función de que exista o no dicho obstáculo dentro del rango de detección del sensor.



A23: Servomotor

Un servomotor o abreviado servo es un motor especial que puede posicionar su eje en un ángulo determinado y lo puede mantener en esta posición. Los servos estándar suelen girar 180º, pero es habitual encontrar servos que giran 90º y otros 360º, que son los conocidos como servos de rotación continua. En el interior del mismo están ubicados tanto la electrónica de control como los engranajes reductores que a su vez pueden llevar o no topes físicos que marquen el ángulo de giro. Para su funcionamiento sólo necesita alimentación ser alimentados (conexiones GND y VCC o 5V) y una señal de control.

El interior de un servo lo vamos esquematizado en la imagen siguiente:



Su aspecto es:

Veamos su principio básico de funcionamiento: La electrónica de control del servomotor tiene un circuito de referencia incorporado que emite la señal de referencia, que es un ciclo de 20 ms con un ancho de pulso de 1,5 ms. Se compara la tensión de control recibida con la de referencia y se genera una diferencia de tensión. El circuito de control en la placa decidirá la dirección de rotación en consecuencia y accionará el motor. El sistema de engranajes o reductora convierten el giro del motor en un par de fuerza a través del eje. El sensor detecta que se ha alcanzado la posición enviada de acuerdo con la señal de retroalimentación. Cuando la diferencia de tensión existe el motor gira y cuando la diferencia se reduce a cero, el motor se detiene. Normalmente, el ángulo de rotación es de 0 a 180 grados.

El servomotor viene con un conector hembra de tres pines para tres cables de conexión, que se distinguen por los colores marrón, rojo y naranja (diferentes marcas pueden tener diferentes colores).

El ángulo de rotación del servomotor se controla regulando el ciclo de trabajo de la señal PWM cuyo estándar es de 20 ms (50 Hz).

Para poder conectar un servomotor a la placa Easy Plug debemos utilizar el adaptador que vemos en la imagen siguiente.

Hay que tener mucho cuidado de posicionar el conector del servo en los tres pines macho del adaptador para hacerlo en el orden correcto (el conector es reversible) o seguramente romperemos algo de manera irreversible.

En el apartado de bloques de programación, se encuentra en "actuadores".

Para controlar el servomotor, indicamos los grados de rotación (Ángulo de giro) que queremos y el tiempo de retardo, o tiempo que tarda en ir de una posición a otra.

PRÁCTICA A23.1:

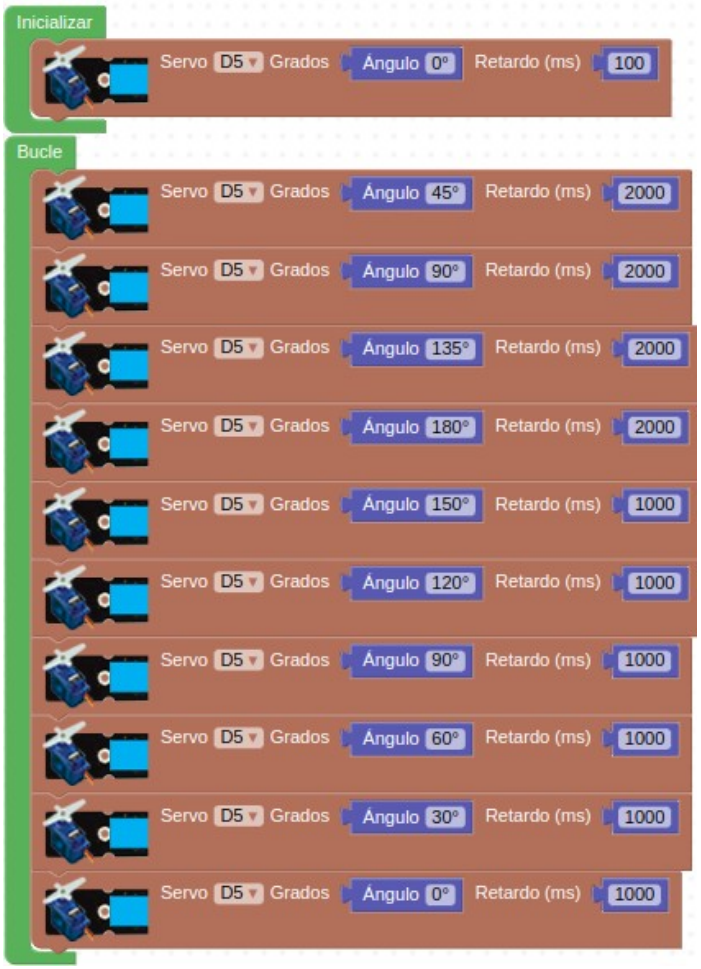
Vamos a posicionar el servo en su ángulo de 0º u origen. Esta práctica es siempre una buena idea cuando no sabemos cual es la posición 0º del servo.

- Programar, por primera vez el ángulo a 0º para descubrir el punto de origen y a partir de aquí montar alguna pala de las que vienen con el servo para poder visualizar la rotación del eje.

PRÁCTICA A23.2:

Una vez averiguada la posición inicial del servo vamos a hacer que este se mueva con la siguiente secuencia de ángulos: 0, 45, 90, 135, 180, 150, 120, 90, 60, 30, 0.

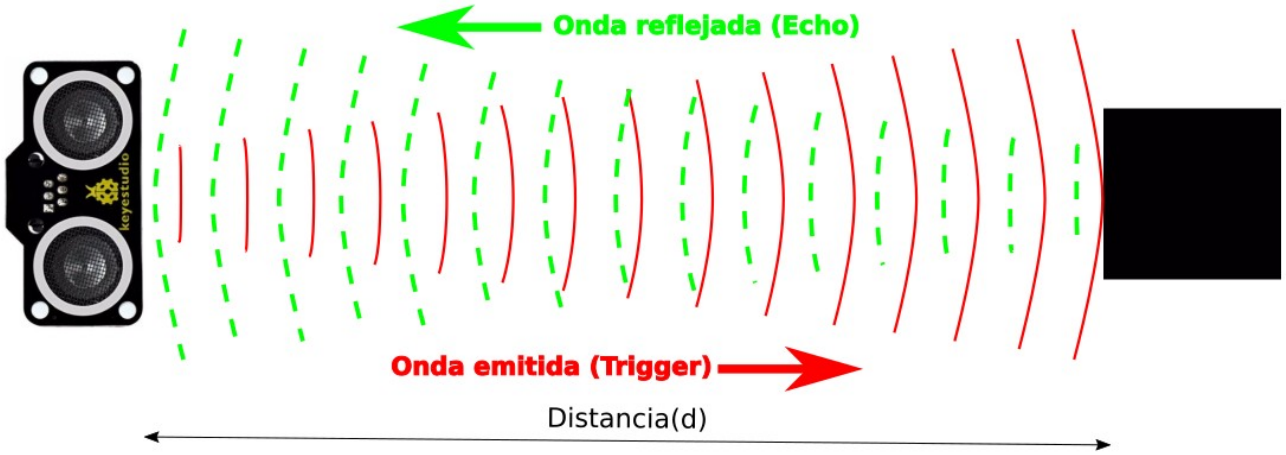
- Hacer que el servo gire de 0 a 180 en incrementos de 45º con un retardo de 2 segundos y de 180 a cero con un decremento de 30º y un retardo de un segundo.



A24: Sensor de Ultrasonido SR01 V2

Los sensores ultrasónicos utilizan un sonar para determinar la distancia desde el sensor al objeto. Este módulo utiliza un chip CS100A que puede medir distancias entre 4 cm y 300 cm siendo la medida precisa y estable. El módulo incluye el transmisor y el receptor ultrasónicos y su circuito de control. El dispositivo debe conectarse a dos pines, por lo que utilizaremos el conector etiquetado con D3 y D4. Esto se debe a que para el funcionamiento del sensor necesita dos pines, uno para emitir el ultrasonido (Trigger) y otro para recibirlo (Echo).

El principio de funcionamiento es el de la imagen siguiente:



El sensor lo que hace es medir el tiempo (t) microsegundos que tarda en recibir el eco del sonido emitido y como la velocidad (v) es conocida, se trata de la velocidad del sonido, que es de 340 m/s o 0.034 cm/us, la distancia vendrá dada por:

$$d = v \cdot t = 0.034 \left(\frac{cm}{us}\right) \cdot t\,(us) = 0.034 \cdot t\,(cm)$$

Aunque nosotros no debemos preocuparnos por esto puesto que el bloque ya no devuelve esta distancia medida en cm.

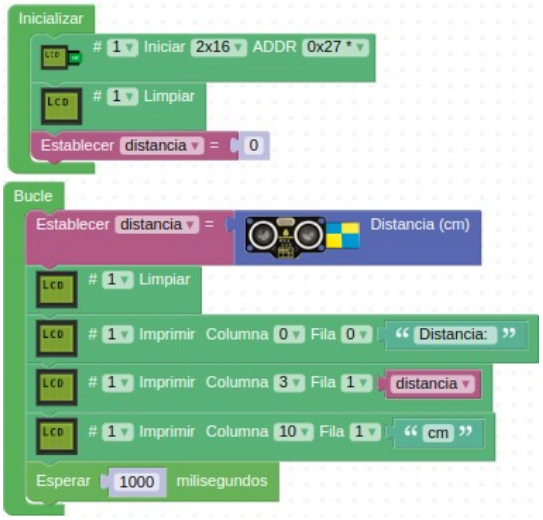
Su aspecto es:

En el apartado de bloques de programación, se encuentra en "Sensores".

PRÁCTICA A24.1:

Vamos a medir la distancia con el sensor de ultrasonidos y mostrar el resultado en una LCD.

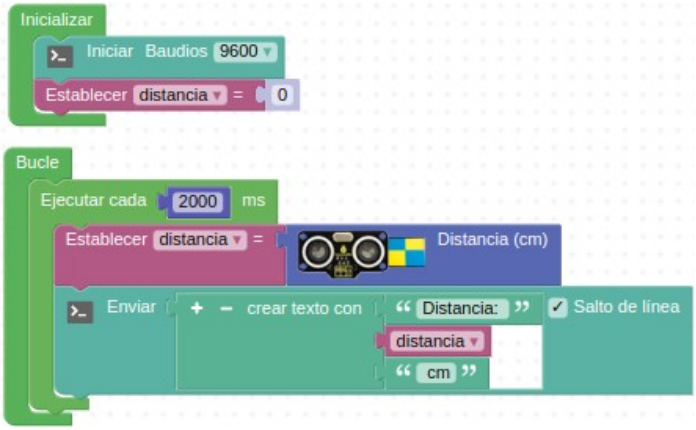
- Mostrar en una LCD la distancia que nos devuelve el sensor de ultrasonidos al poner frente al mismo un objeto a diferentes distancias.



PRÁCTICA A24.2:

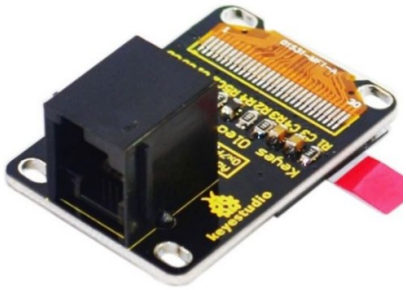
Vamos a medir la distancia con el sensor de ultrasonidos y mostrar el resultado en la consola.

- Mostrar por consola la velocidad medida por el sensor de ultrasonidos enviando un dato cada dos segundos.



A25: Pantalla gráfica OLED 0.96" 128 x 64

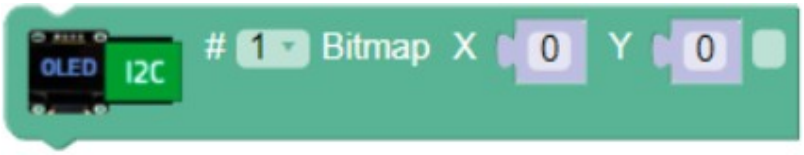
OLED es la abreviatura de (Organic Light-Emitting Diode) o diodo emisor de luz orgánico. Se dice orgánico debido a una película de carbono que se encuentra en el interior del panel, detrás del cristal protector. Cuando se colocan varias películas orgánicas entre dos conductores se puede hacer que cada pixel se ilumine de forma individual, haciendo muy eficiente este tipo de pantallas. Es decir, una pantalla OLED emite luz brillante propia al pasar corriente eléctrica. La OLED Easy Plug es de tipo I2C y su aspecto es:



En el apartado de bloques de programación, se encuentra en "Visualización" y tiene su propio menú de bloques:

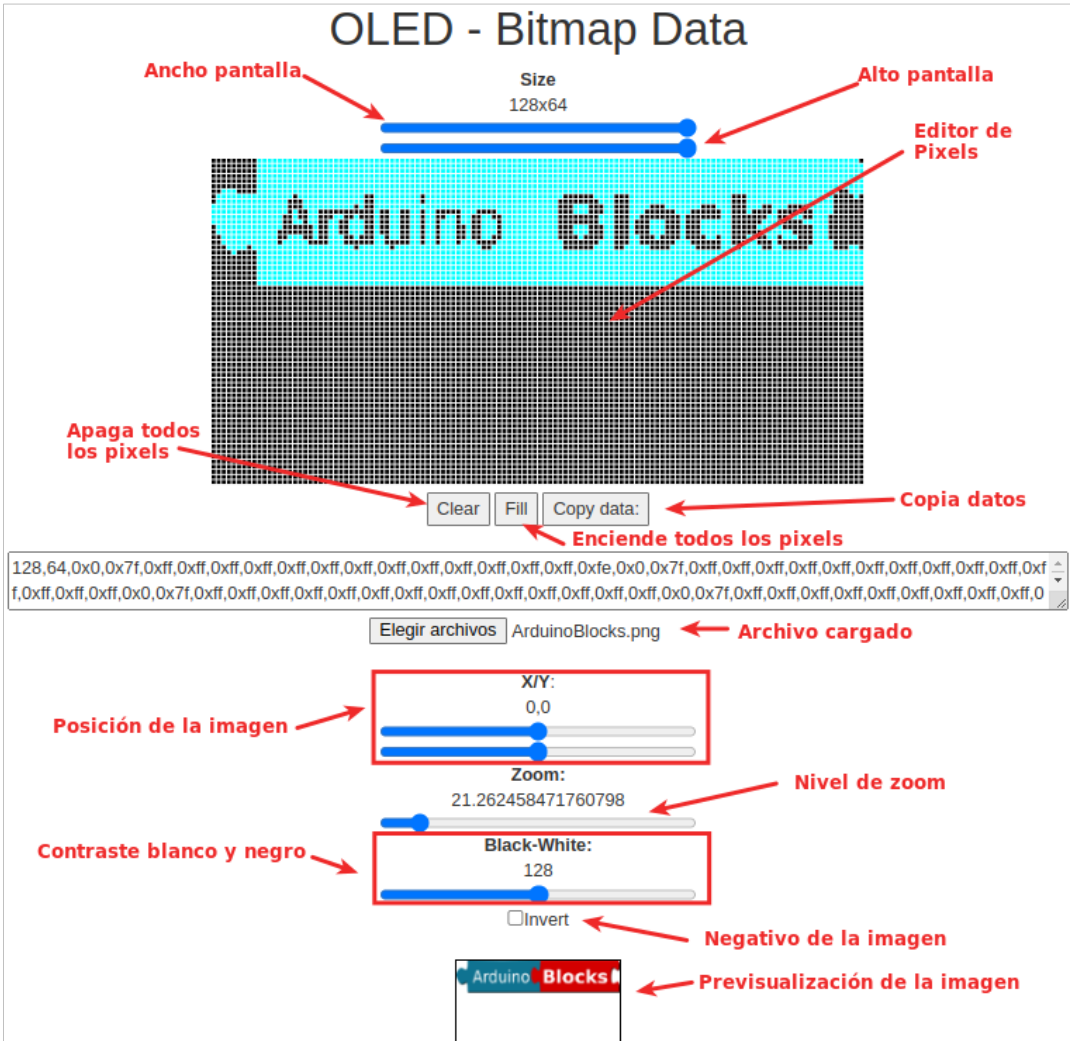
Antes de realizar actividades con la pantalla OLED es necesario conocer las herramientas disponibles en ArduinoBlocks para el trabajo con gráficos. Para poder mostrar gráficos debemos generar un mapa de bits que indique que pixeles estarán apagados y cuales encendidos. ArduinoBlocks dispone de una herramienta visual (OLED - Bitmap Data) que nos permite generar fácilmente el código de los bitmaps y de un bloque que nos permite mostrarlos.

El bloque para mostrar un bitmap es el siguiente:



A la herramienta OLED - Bitmap Data podemos acceder por cualquiera de los métodos que vemos en la imagen siguiente:

En la imagen vemos la pantalla OLED - Bitmap Data y la explicación de sus partes

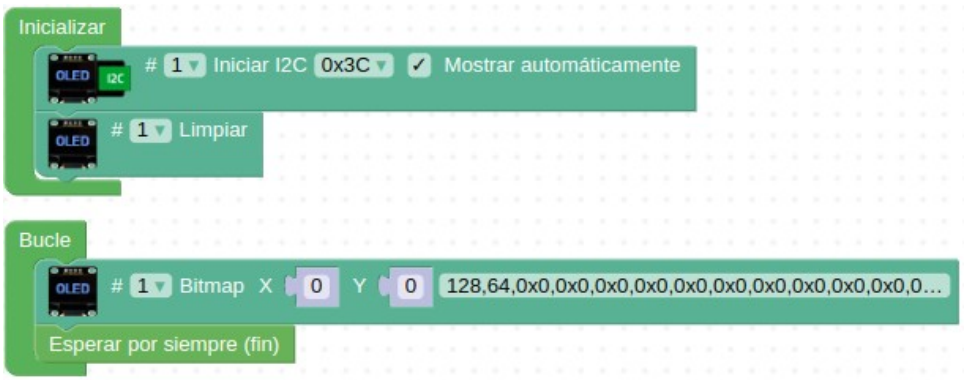


En la imagen siguiente tenemos el bloque de mostrar bitmap con los datos del logo de ArduinoBlocks copiados en el mismo.

PRÁCTICA A25.1:

Vamos a mostrar en una pantalla OLED un logotipo.

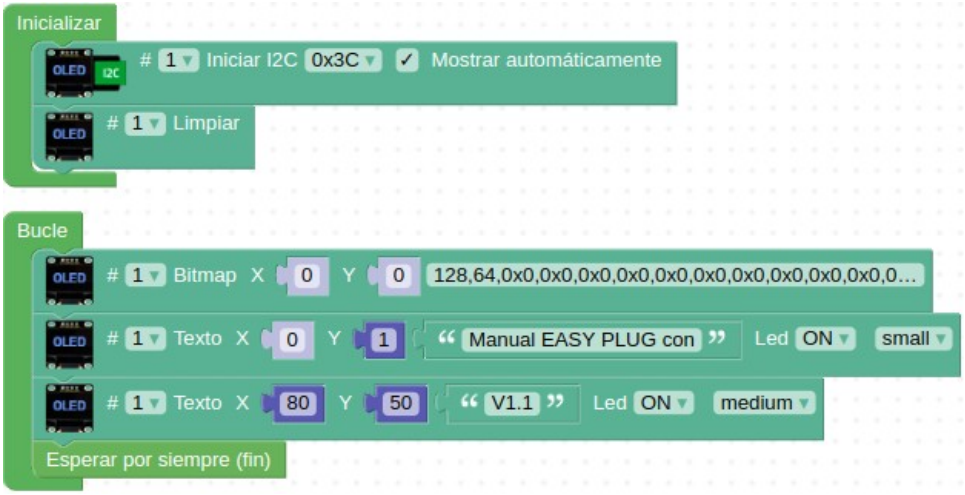
- Mostrar en una OLED el logotipo de ArduinoBlocks.

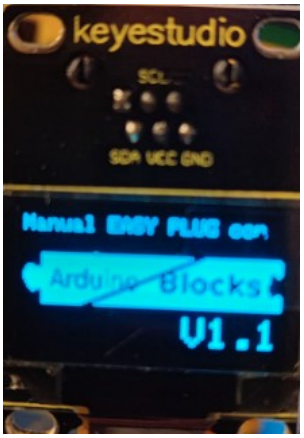


PRÁCTICA A25.2:

Vamos a mostrar en una pantalla OLED un logotipo y un texto.

- Mostrar en la pantalla OLED el mismo logotipo que en la práctica A25.1 situando en la parte superior el texto “Manual EASY PLUG con” en tamaño ‘Small’ y en la parte inferior el texto “V1.1”.





A26: Módulo relé

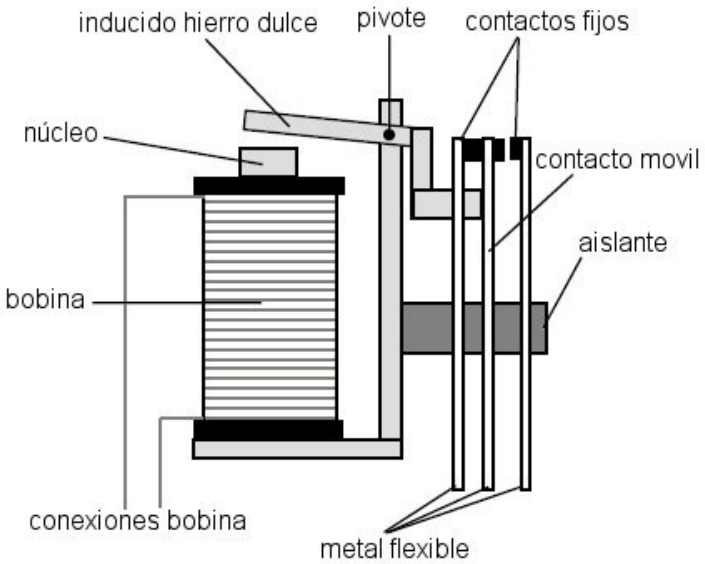
Un relé es básicamente un interruptor mecánico controlado eléctricamente de forma que a través de una pequeña tensión y corriente se puede controlar la apertura y cierre de sus contactos mecánicos donde se puede trabajar con tensiones y corrientes elevadas.

El módulo incluye un LED que nos indica si el relé está accionado cuando se enciende.

El modelo de relé que lleva este módulo es el SRD-05VDC-SL-C que pressenta las siguientes características:

- En AC (corriente alterna): 250Vac / 10A
- En DC (corriente continua): 30Vdc / 10A o bien 250Vdc / 1A
- Tensión de la bobina: 5V, compatible con señales TTL para su excitación.
- Contactos: 3 pines, 1 Común + 1 contacto NA (normalmente abierto) + 1 contacto NC (normalmente cerrado).

Esquemáticamente un relé de este tipo se puede parecer a la imagen siguiente:

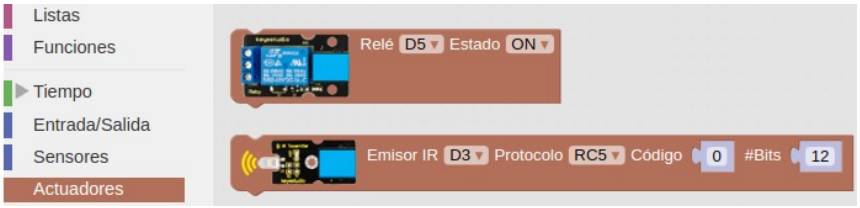


Fuente: wikipedia

En el enlace podemos ver una animación del funcionamiento de un relé.

Su aspecto es:

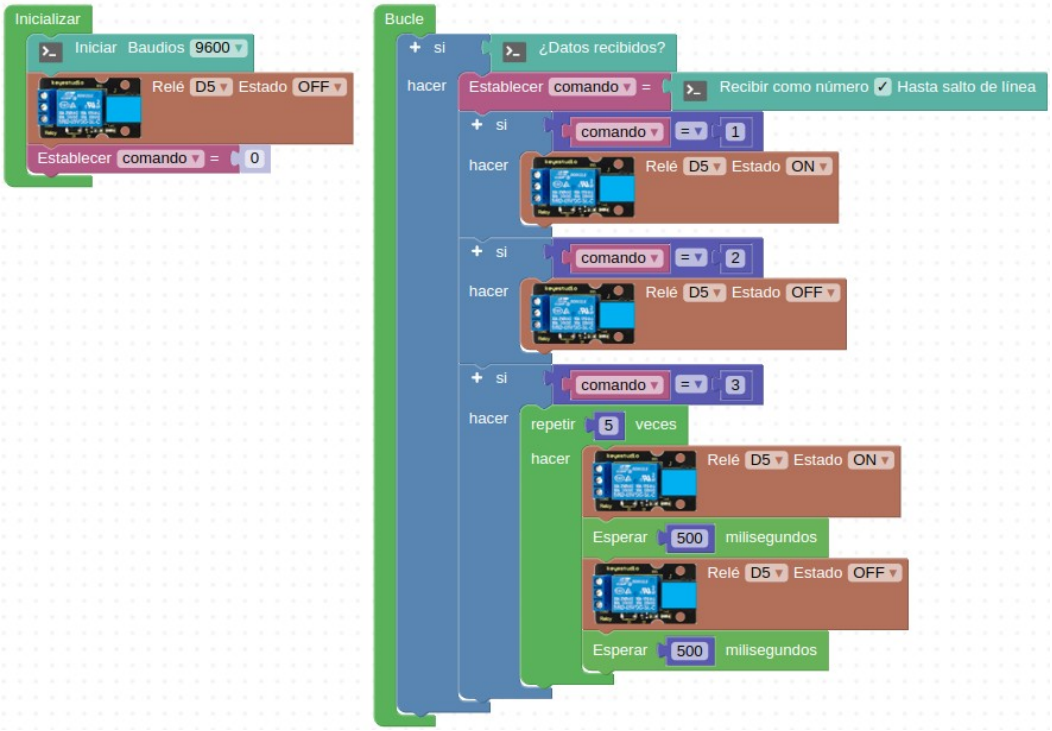
En el apartado de bloques de programación, se encuentra en "Actuadores".



PRÁCTICA A26.1:

Vamos a controlar el accionamiento de un relé desde nuestro ordenador a través de la consola serie.

- Conectar un módulo relé y vamos a hacer que funcione a través de los siguientes comandos que enviaremos desde la consola: 1 –> ON, 2→ OFF y 3 → Conmutar ON/OFF a 500 ms.



El aspecto de la consola lista para enviar el comando de encendido es:

A27: Módulo joystick

El módulo se componen de dos potenciómetros, uno para el movimiento del eje X y otro para el movimiento del eje Y asociados a entradas analógicas y además incorpora un pulsador asociado a un pin digital. En la placa EASY PLUG existe un terminal de conexionado marcado en rojo y amarillo que son los pines A6, A7 y D2. Este mismo tipo de conexión la lleva el módulo joystick con la siguiente relación de conexionado:

- El eje X se conecta a A6
- El eje Y se conecta a A7
- El pulsador o eje Z se conecta a D2

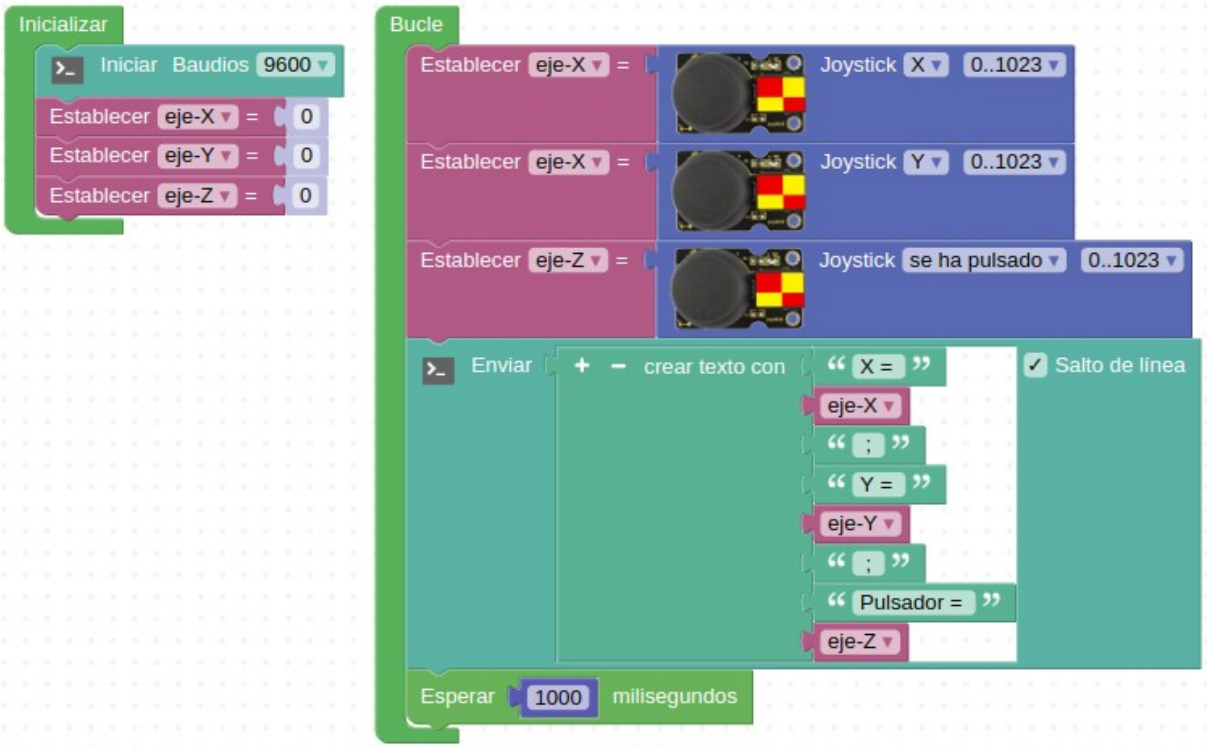
Su aspecto es:

En el apartado de bloques de programación, se encuentra en "Sensores".

PRÁCTICA A27.1:

Vamos a mostrar por consola los valores leídos del sensor.

- Mostrar en la consola el valor de cada uno de los ejes en función de la posición del mando del joystick y también indicar si se ha accionado el pulsador.



Una captura de la consola donde se ven los cambios es:

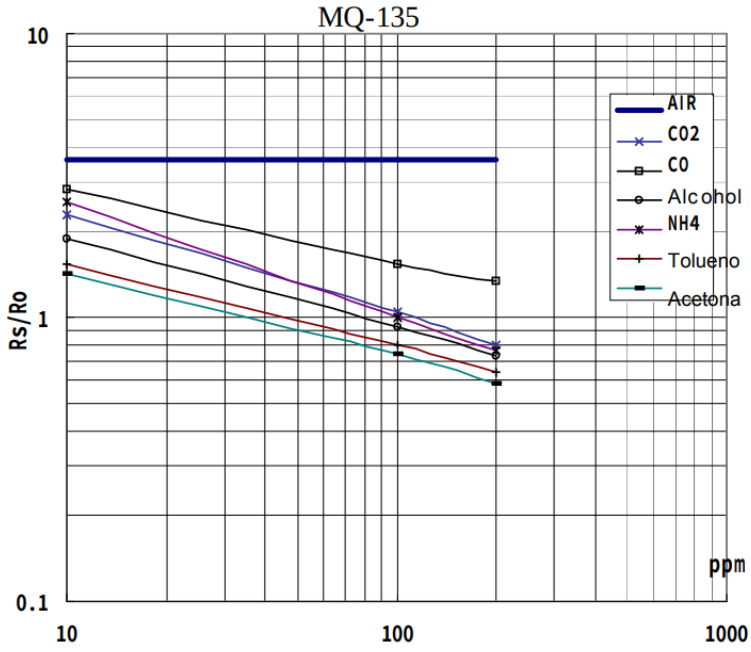
A28: Sensor MQ-135

El MQ-135 dispone de una lámina de SnO_2 como material sensible a los gases y basa su funcionamiento en que el SnO_2 tiene baja conductividad eléctrica en el aire limpio. Su funcionamiento es muy sencillo, según aumente la cantidad de contaminantes en el aire también aumentará su conductividad eléctrica y viceversa.

El MQ-135 tiene una sensibilidad alta a la presencia de amoniaco, sulfuros, vapores de benceno, humo y otros gases nocivos para la salud. El módulo dispone de una resistencia ajustable que permite ajustar la sensibilidad del mismo.

En el datasheet del sensor tenemos un ábaco de curvas que nos muestra las características típicas de sensibilidad del MQ-135 para varios gases. Las curvas se dan a una temperatura de 20 ºC, una humedad del 65% bajo una concentración de O_2 del 21% y una RL=20kΩ con:

- Ro: resistencia del sensor a 100 ppm de NH_3 en aire limpio.
- Rs: resistencia del sensor a varias concentraciones de gases.



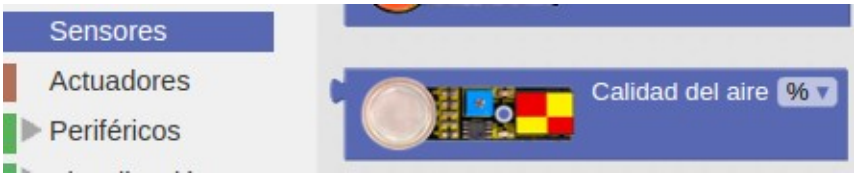
El sensor requiere para su funcionamiento un calentamiento previo de unos 30 segundos y cuando se utiliza por primera vez se recomienda que este sea de entre 8 y 12 horas para que el calor que se genera durante este tiempo queme los residuos del proceso de fabricación. No es recomendable tocar el sensor una vez conectado ya que puede quemar.

Los módulos MQ son sensibles a más de un gas y en diferentes proporciones por lo que no sirven para identificar la presencia de un gas especifico.

Dado que el sensor va a estar en ambientes con presencia de gases que pueden ser inflamables el mismo se encierra en una cápsula de dos capas de malla de acero inoxidable que asegura que el elemento calefactor interno no cause una explosión. Esta malla sirve también de filtro para las partículas en suspensión evitando que entren al interior de la cámara y que solamente penetren a ella los gases. El sistema calefactor se realiza mediante una bobina de níquel cromado que es la que se recubre del dióxido de estaño sensible a gases combustibles.

Su aspecto es:

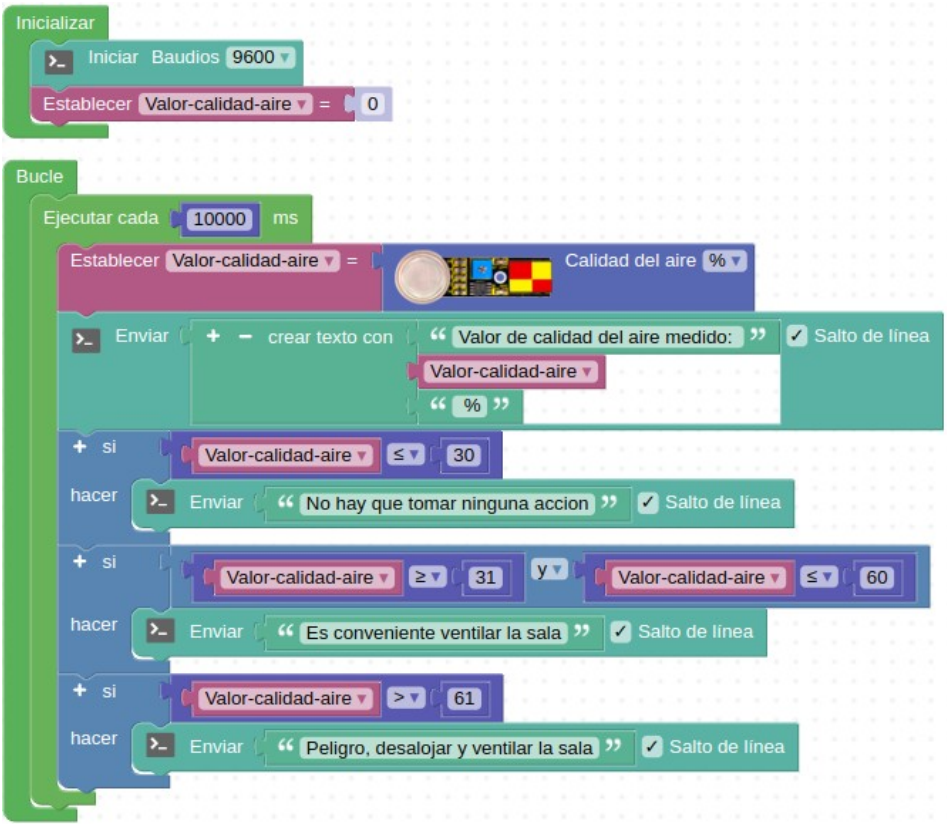
En el apartado de bloques de programación, se encuentra en "Sensores".



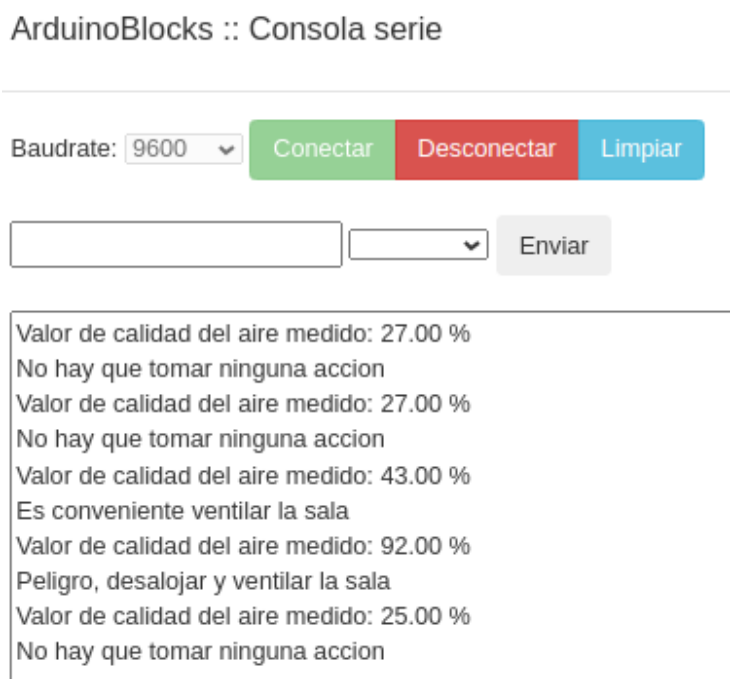
PRÁCTICA A28.1:

Vamos a medir la presencia de gases en el ambiente de una habitación.

- Mostrar en la consola mensajes relativos a la calidad del aire de una habitación en función de que se detecte un determinado nivel de presencia de gases en la misma. El bloque nos entrega la medida en % y vamos a estimar que si el valor medido está por debajo del 20% la calidad del aire es buena, si está entre e 21% y el 60% es aceptable y que si el porcentaje supera el 60% la calidad del aire es mala. Asociaremos a cada grupo de valores los mensajes de “No hay que tomar ninguna acción”, “Es conveniente ventilar la sala” y “Peligro, desalojar y ventilar la sala”.



Ante presencia de gases o alcoholes la consola muestra lo siguiente:

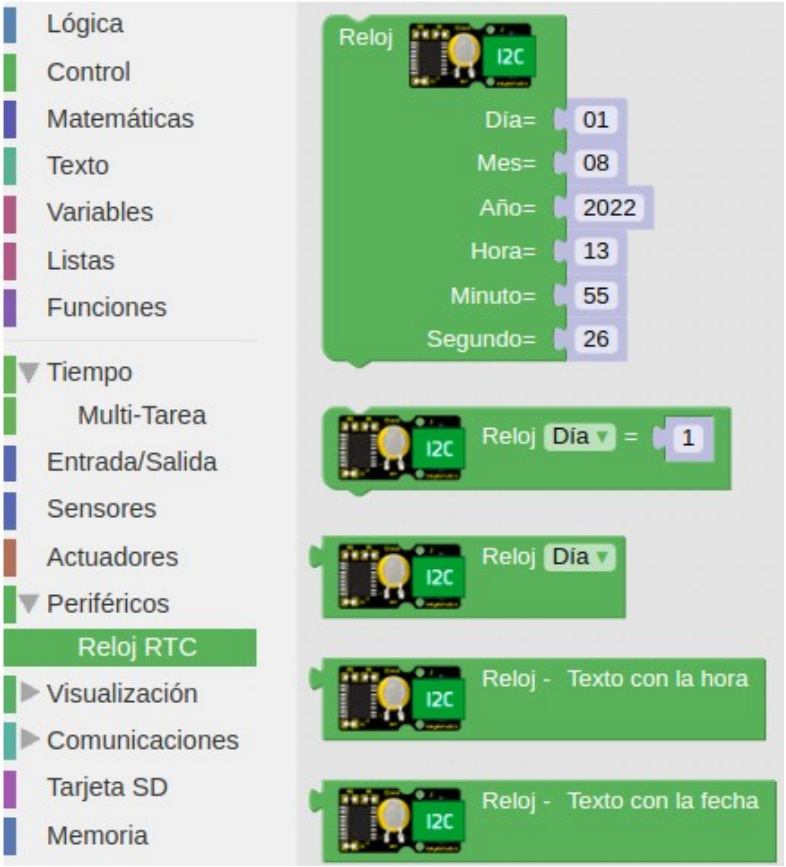


A29: Módulo RTC DS3231

El circuito integrado DS3231 en el que se basa el módulo es un reloj I2C en tiempo real (RTC de Real Time Clock) muy preciso con un oscilador de cristal compensado con la temperatura que nos devuelve la hora y la fecha. El dispositivo incorpora una batería encargada de mantener la hora y fechas ajustadas cuando se interrumpe la alimentación principal al mismo.

Su aspecto es:

En el apartado de bloques de programación, se encuentra en "Periféricos" y tiene sus propios bloques.



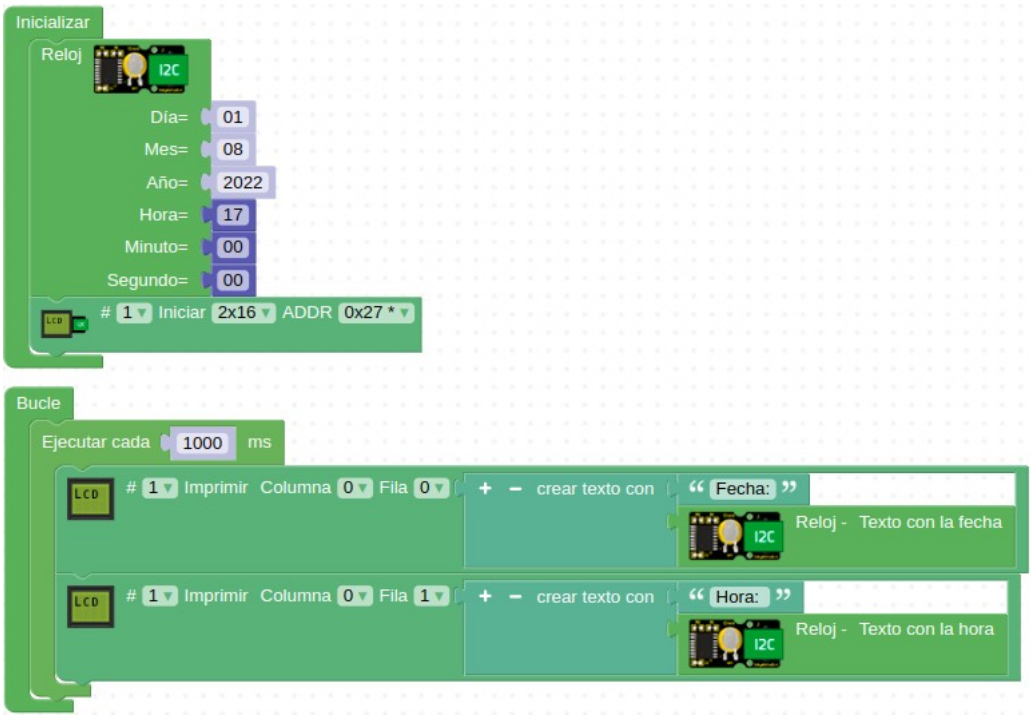
PRÁCTICA A29.1:

Ajuste de fecha y hora del módulo RTC DS3231

- Ajustar la fecha y hora actuales o las deseadas en el RTC y mostrar sus valores leídos del dispositivo.

En esta ocasión vamos a utilizar dos dispositivos I2C por lo que se hace necesario el uso del Hub I2C para conectar ambos a la única entrada de que dispone la placa.

Lógicamente el ajuste solamente lo debemos realizar una vez y a partir de ahí no poner el bloque de configuración en nuestro programa y que sea la batería la encargada de mantener la fecha y la hora. El valor que lee el bloque para la fecha y la hora por defecto es el del ordenador.



El aspecto de la LCD justo después de subir el programa y transcurridos unos minutos sin alimentar tras los cuales volvemos a conectar el sistema.

A30: Acelerómetro de tres ejes ADXL345

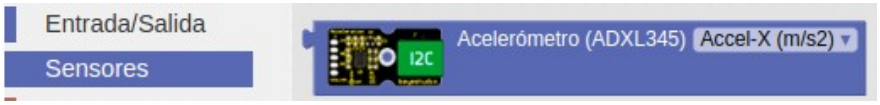
El módulo ADXL345 es un acelerómetro MEMS (MicroelEctroMechanical Systems) o sistema microelectromecánico de 3 ejes con capacidad de medición de hasta ±16g de fuerza gravitacional.

Los datos de salida los entrega en formato digital en complemento a dos de 16 bits y se puede acceder a ellos a través de una interfaz digital I2C.

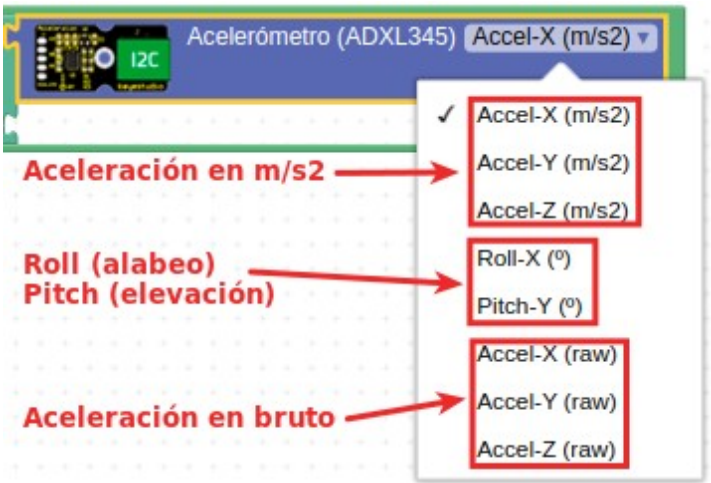
El ADXL345 es ideal para medir la aceleración estática de la gravedad en aplicaciones de detección de inclinación, así como la aceleración dinámica resultante del movimiento o impacto.

Su aspecto es:

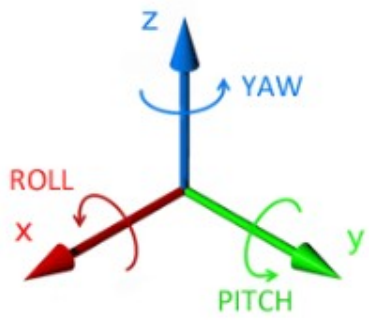
En el apartado de bloques de programación, se encuentra en "Sensores".

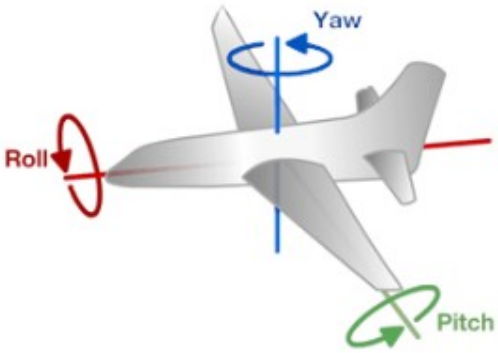


Dispone de un solo bloque en el que se pueden seleccionar las opciones que vemos en la imagen siguiente:



Los conceptos que vemos en la imagen se refieren a la rotación del eje X (Roll) y la rotación del eje Y (Pitch) y se ha prescindido del eje Z (Yaw). Para aprender mas sobre estos conceptos visita este enlace al blog de Luis Llamas. En la explicación de los conceptos básicos de este tipo de dispositivos se muestra este gráfico donde se comparan los conceptos de movimientos angulares aplicados a una aeronave.

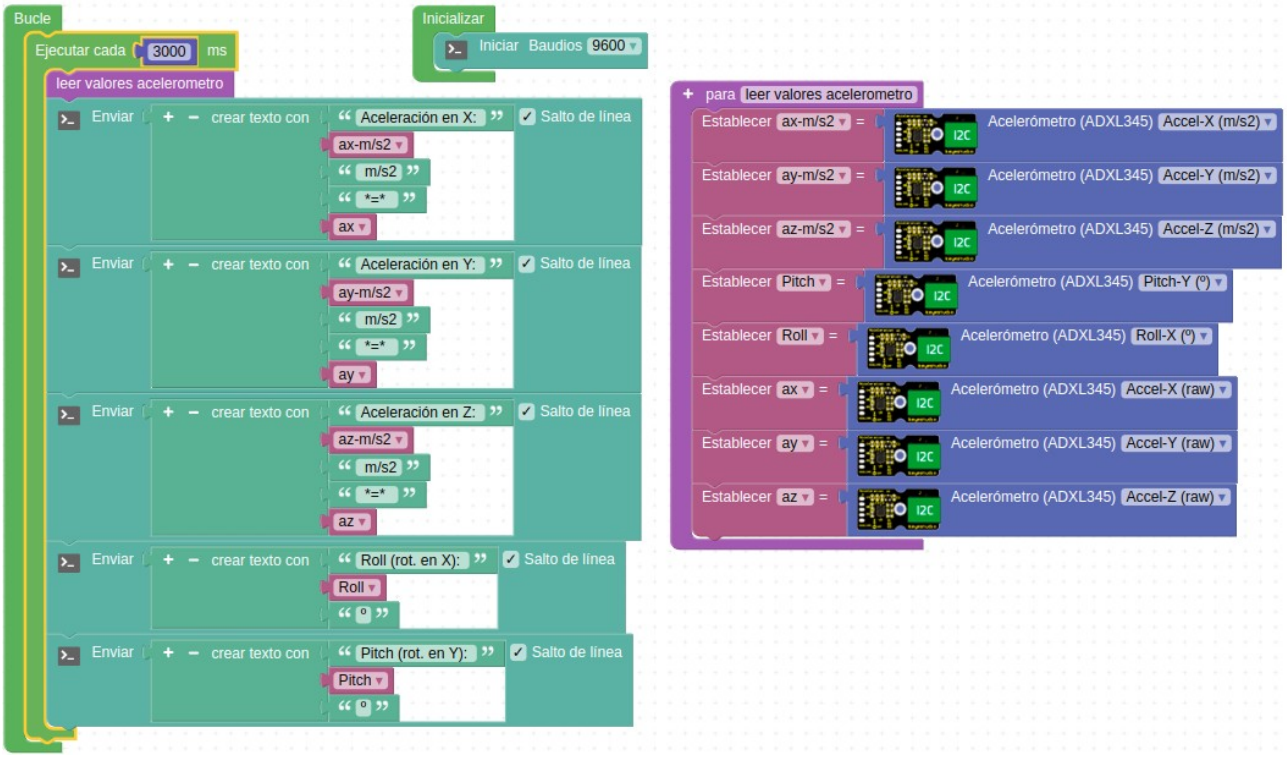




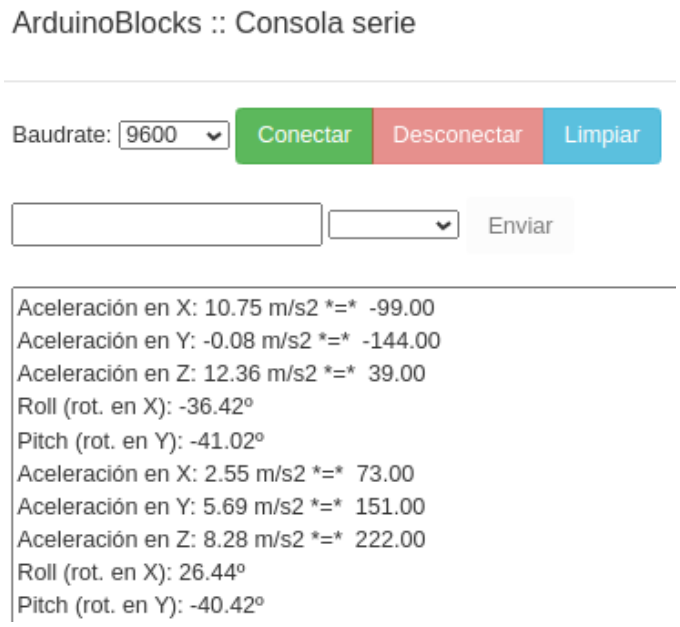
PRÁCTICA A30.1:

Vamos a mostrar por consola el valor de todas las opciones de que dispone el bloque.

- Mostrar en la consola cada una de las opciones del bloque del acelerómetro para poder hacerlas cambiar moviendo el mismo y ver los resultados que obtenemos.



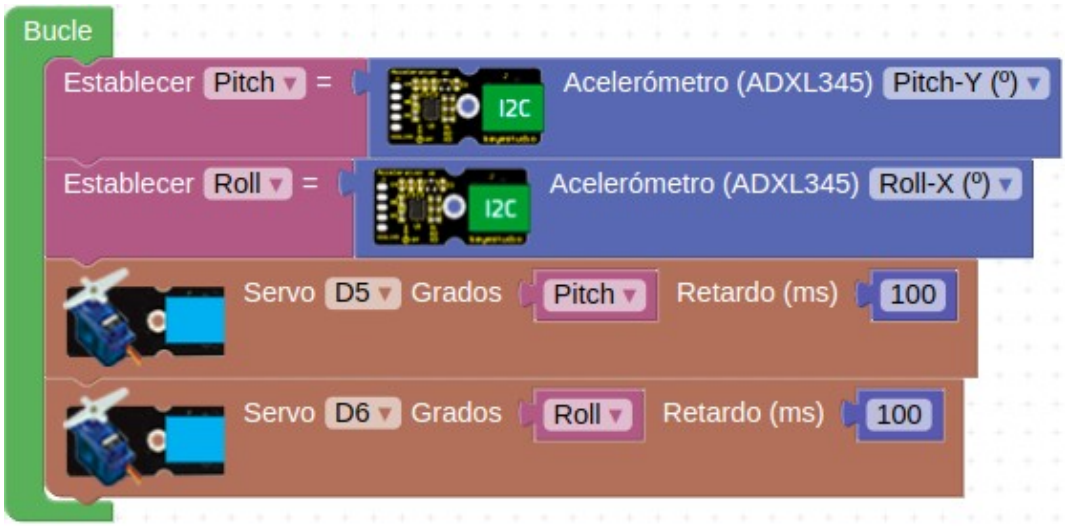
En la imagen siguiente vemos algunos resultados obtenidos en la consola.



PRÁCTICA A30.2:

Vamos a programar dos grados de libertad de un brazo robot.

- Realizar el programa para controlar un sistema como el de la imagen, que es un soporte para servos con 2 DOF (grados de libertad) para el montaje tipo cardan de una cámara para un robot teledirigido.



A31: Potenciómetro deslizante

El potenciómetro deslizante o slide es una resistencia variable a la que podemos cambiar su valor deslizando un cursor por una resistencia a lo largo de toda su longitud. Muy característico de mesas de mezclas de audio. En realidad se trata de un potenciómetro doble montados de forma que cuando uno aumenta su resistencia el otro la disminuya y viceversa.

Su aspecto es:

El color con el que viene marcado nos indica que es un sensor analógico pero en realidad funciona con dos pines analógicos y el único conecto de la placa que los tiene es el marcado en rojo y amarillo (D6 y D7) que son los únicos pines que deja conectar el bloque de ArduinoBlocks.

En realidad se trata de una salida analógica dual que nos entrega una señal analógica variable entre 0 y VCC, es decir valores entre 0 y 1023 (o 0 y 100%). La suma de los dos valores analógicos que da el potenciómetro es 1023.

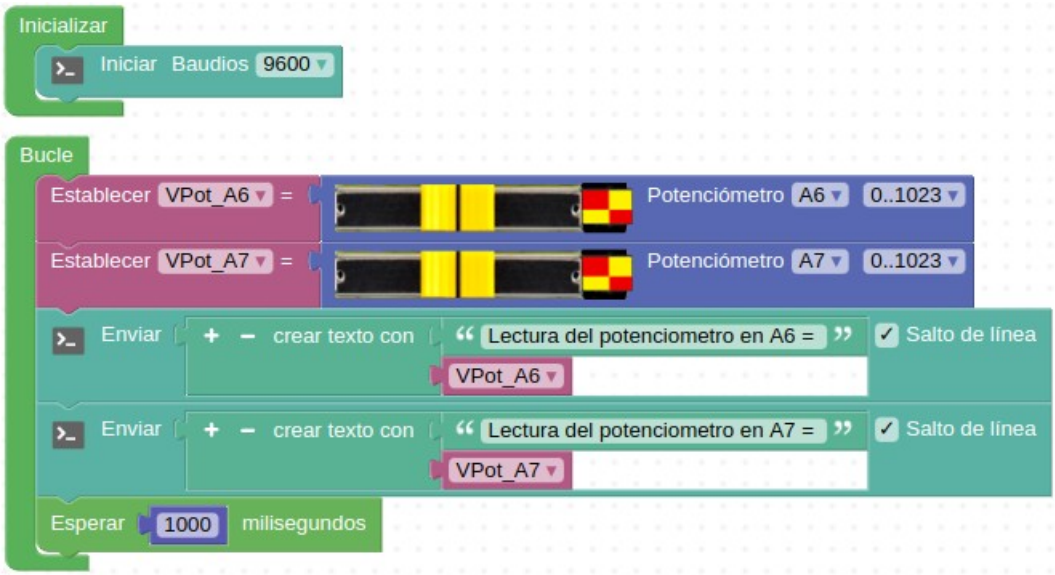
Hay posibilidad de soldar dos conectores de 3 pines de paso 2,54 mm en el módulo que permite su conexión con otros dispositivos.

En el apartado de bloques de programación, se encuentra en "Sensores".

PRÁCTICA A31.1:

Vamos a medir las variaciones de las resistencias del potenciometro.

- Mostrar en la consola el valor como lectura analógica de 0 a 1023 de cada una de las resistencia del potenciómetro deslizante.



PRÁCTICA A31.2:

Controlar un servo 0-180º mediante el potenciómetro deslizante.

- Realizar el mapeo de una variable que mide las variaciones de resistencia de un potenciómetro deslizante de forma que haga girar a un servo entre 0 y 180º en función del valor de la resistencia del mismo.

A32: Módulos receptor y emisor de infrarrojos IR

- **Receptor IR**

El receptor de infrarrojos es un módulo con funciones de recepción, amplificación y demodulación para que entregue una señal digital demodulada en frecuencia de 38KHz. Normalmente se utiliza junto con el módulo transmisor IR o algún mando a distancia universal o un mando a distancia como el del enlace.

Su aspecto es:

En el apartado de bloques de programación, se encuentra en "Sensores".

- **Emisor IR**

Un transmisor de infrarrojos es simplemente un diodo LED que genera luz infrarroja. Por ejemplo, en un control remoto infrarrojo al presionar un botón se envía una señal al LED IR, que convierte la señal en un haz de luz infrarroja. El dispositivo receptor detecta la luz con un fotodiodo IR y la convierte en una señal eléctrica a través de un circuito integrado, controlando así sus acciones.

Los transmisores infrarrojos se utilizan ampliamente como medio de comunicación inalámbrica mediante controles remotos para televisores y otros dispositivos electrónicos.

Este módulo transmisor IR se empareja con el módulo receptor IR y sus principales características son:

- Frecuencia central infrarroja: 850nm a 940nm
- Ángulo de emisión infrarroja: 20 grados
- Distancia de emisión infrarroja: 1.3m (5V 38Khz)

Su aspecto es:

En el apartado de bloques de programación, se encuentra en "Actuadores".

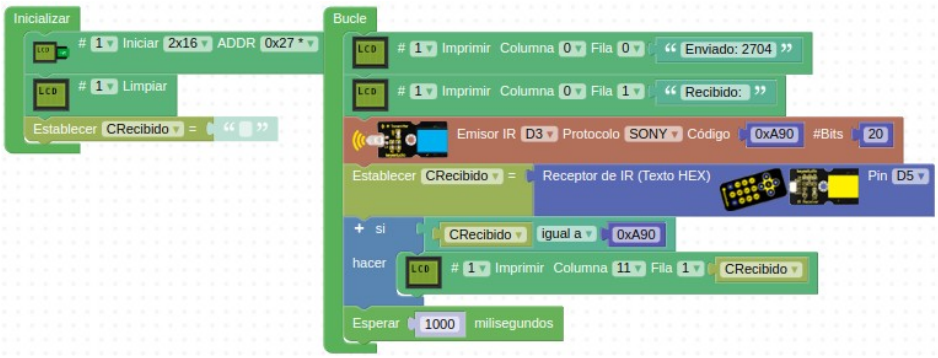
El bloque permite seleccionar el protocolo al que pertenece el código a enviar que pongamos en el campo correspondiente y solamente permite conectar el módulo al pin D3.

Para obtener la información de un mando IR cualquiera y decodificar el protocolo, número de bits y código de cada tecla podemos seguir este tutorial de Adafruit sobre la utilización de su librería de IR para el IDE de Arduino.

PRÁCTICA A32.1:

Vamos a enviar un código desde el módulo emisor, lo visualizamos en la LCD, lo recibimos en el receptor y lo visualizamos también en la LCD.

- Enviar el código Sony 0xA90 (2704 en decimal) desde el módulo emisor y comprobar si el módulo receptor lo recibe mostrando los resultados en una pantalla LCD.

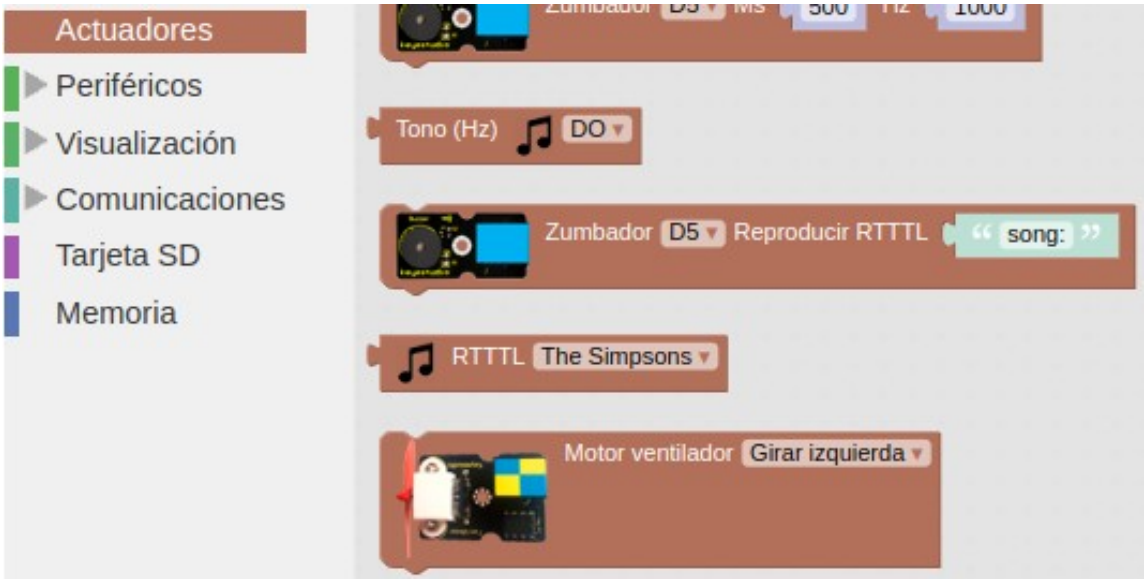


A33: Módulo ventilador L9110

El módulo se componen de un motor de corriente continua a cuyo eje se puede aplicar una pequeña hélice que se puede usar como mini ventilador o como hélice en si misma. El módulo incorpora el controlador L9110 para poder seleccionar el sentido de giro.

Su aspecto es:

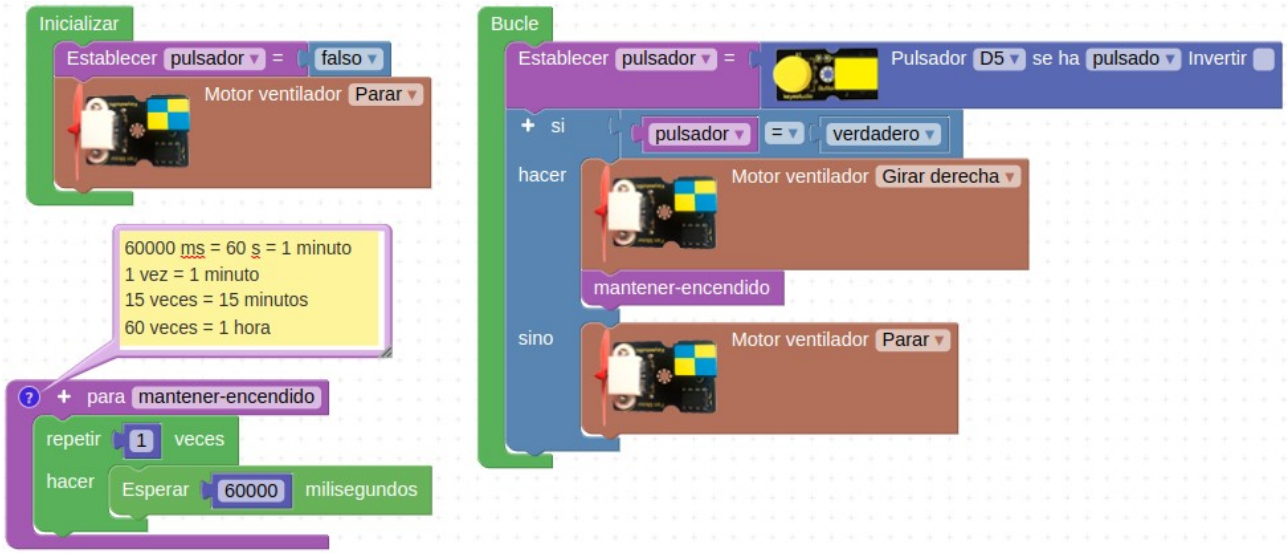
En el apartado de bloques de programación, se encuentra en "Actuadores".



PRÁCTICA A33.1:

Vamos a hacer que el motor gire a derechas y se detenga transcurrido un tiempo que prefijamos a través de una función creada a tal efecto.

- Implementar un ventilador que se acciona mediante un pulsador y que se mantiene en funcionamiento durante el tiempo que prefijemos.



Actividad propuesta. PRÁCTICA A33.2:

Hacer que el motor gire en los dos sentidos para crear un ventilador de techo con posiciones de verano e invierno. El efecto ventilador verano e invierno se basa en lo siguiente: el funcionamiento más habitual de un ventilador de techo es impulsar el aire hacia abajo, directamente por debajo de él girando en sentido horario. Si queremos mover el aire caliente próximo al techo el ventilador debe girar en sentido contrario a las agujas del reloj, es decir, impulsando el aire hacia arriba y succionando por su parte inferior. De forma que el aire caliente se desplaza lateralmente, baja por las paredes hacia el suelo y vuelve a subir por el centro del ventilador.

- Implementar un ventilador que podamos configurar el modo de funcionamiento como modo verano y modo invierno pulsador.

A34: Módulo relé Reed

Un interruptor de lengüeta o reed switch o relé reed es un interruptor eléctrico activado por un campo magnético.

Cuando los contactos están normalmente abiertos se cierran en la presencia de un campo magnético; cuando están normalmente cerrados se abren en presencia de un campo magnético.

Este módulo incluye un relé reed que cierra sus contactos cuando se expone a un campo magnético y están abiertos si no hay campo magnético presente. es un pequeño dispositivo llamado interruptor de láminas en el módulo. Esto lo hace útil para montarlo por ejemplo en una puerta con propósito de activar una alarma o como interruptor.

Sus principales características son:

- Voltaje de funcionamiento: 3,3 V a 5 V DC
- Corriente de trabajo: ≥20mA
- Temperatura de trabajo: -10 ℃ a ＋50 ℃
- Distancia de detección: ≤10 mm

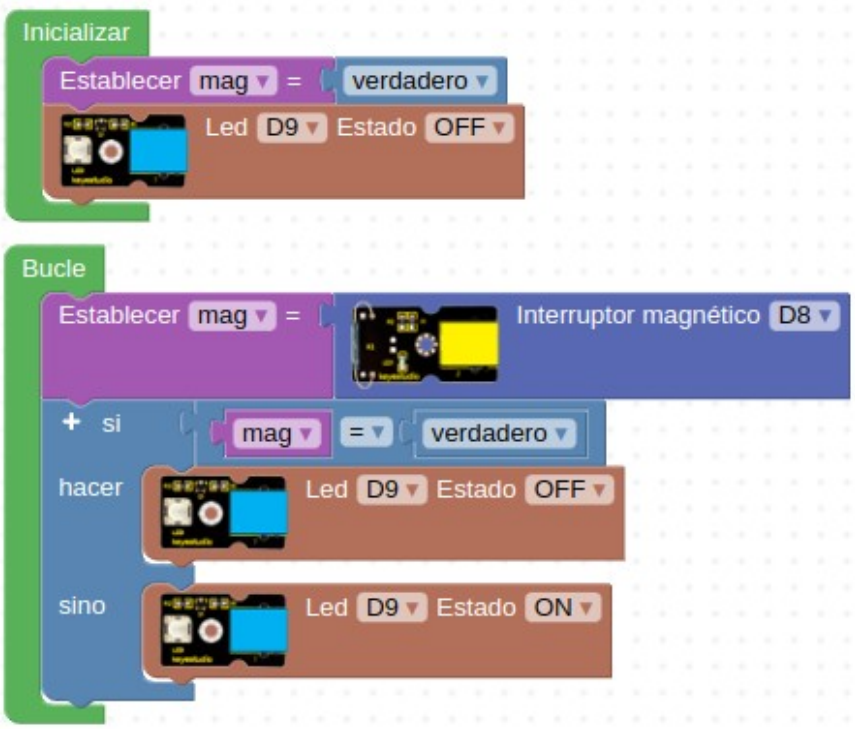
Su aspecto es el siguiente:

En el apartado bloques de programación lo encontramos en “Sensores”.

PRÁCTICA A34.1:

Vamos a montar un sistema que cuando acerquemos un imán se encienda un diodo LED.

- Implementar un programa que haga funcionar al relé reed como detector magnético en una supuesta puerta de un armario. Cuando la puerta está cerrada se detecta el campo magnético y el diodo LED está apagado y cuando se abre este debe apagarse.

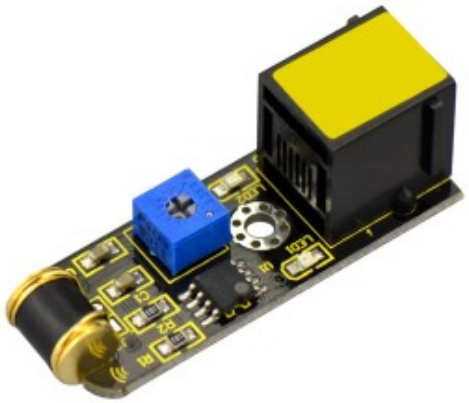


A35: Sensor de vibración

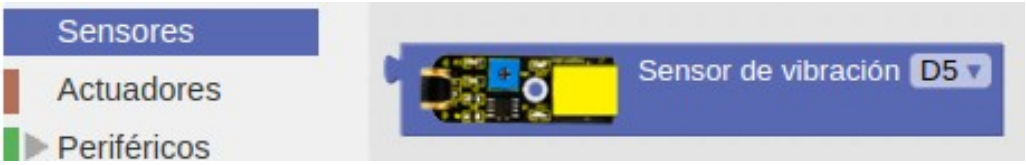
Se trata de un sensor digital que cuando se somete a vibración cambia su estado de salida. El módulo detector de vibración consta de un resorte y un poste conductor central que, ante golpes y vibraciones, el resorte reacciona desplazándose de su centro y cerrando el circuito.

A pesar de la simplicidad, se puede aprovechar para contar pasos, alarma de advertencia de colisión, etc.

Su aspecto es el siguiente:



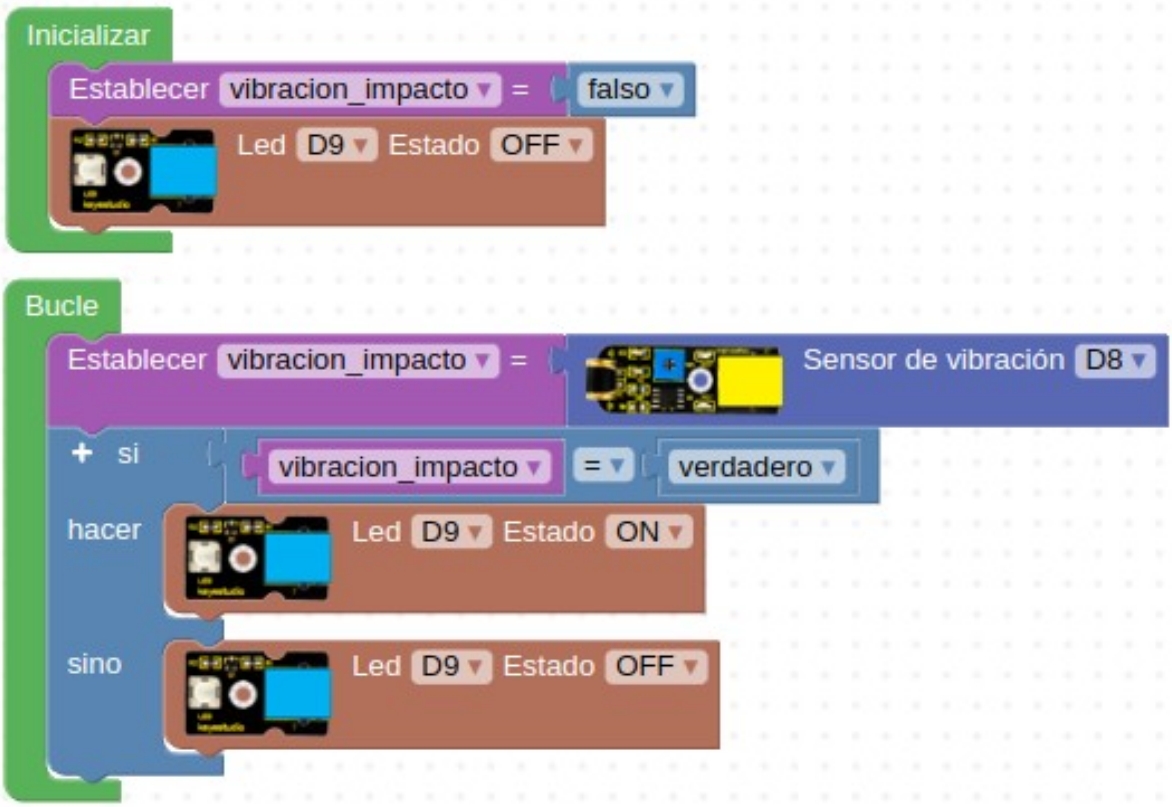
En el apartado bloques de programación lo encontramos en “Sensores”.



PRÁCTICA A35.1:

Vamos a detectar vibraciones o impactos encendiendo un LED cuando se producen.

- Implementar un programa que haga funcionar al sensor como detector de vibraciones o impactos de forma que cuando estos ocurran se encienda el LED.



A36: Módulo Bluetooth

El módulo Bluetooth 2.0 permite transmitir fácilmente datos en serie. La frecuencia operativa se encuentra entre las bandas de frecuencia ISM (Industrial, Scientific and Medical) de 2,4 GHz, es decir en uso industriales, científicos y médicos.

Trabaja según el estándar Bluetooth 2.1+EDR. El Endpoint Detection and Response (conocido por su siglas en inglés EDR) es una herramienta que proporciona monitorización y análisis continuo del endpoint y la red. Tiene una antena incorporada que proporciona señales de alta calidad. Sus principales características son:

- Protocolo Bluetooth: Bluetooth 2.1+ EDR
- Protocolo USB: USB v1.1/2.0
- Frecuencia de funcionamiento: banda ISM de 2,4 GHz
- Modo de modulación: GFSK (modulación por desplazamiento de frecuencia de Gauss)
- Potencia de transmisión: ≤ 4dBm, clase 2
- Sensibilidad: ≤-84dBm a una tasa de error de bits del 0,1 %
- Velocidad de transferencia: asíncrona: 2,1 Mbps (máx.)/160 kbps; Síncrono: 1Mbps/1Mbps
- Característica de seguridad: Autenticación y encriptación
- Configuración admitida: puerto serie Bluetooth

Su aspecto es:

Hay dos tipos de módulos Bluetooth básicos, los modelos HC-05 y HC-06. La diferencia entre ellos es que el HC-05 es maestro-esclavo, lo que significa que además de recibir conexiones desde un ordenador, tablet o móvil Android, también es capaz de generar conexiones hacia otros dispositivos bluetooth. El HC-06 solo puede funcionar en modo esclavo. En BlueTooth HC-05 y HC-06 hay disponible mas información sobre estos módulos. El Bluetooth Easy Plug es HC-06.

Un módulo de comunicaciones Bluetooth tiene las 4 conexiones siguientes que son las que se implementan en el conector RJ11:

- Vcc: Alimentación 5V (+)

- Gnd: Alimentación 0V (-)
- Tx: Transmisión de datos
- Rx: Recepción de datos

Los pines Tx y Rx son precisamente los que utiliza la placa su programación por lo que cuando utilicemos este módulo debemos tener muy presente la siguiente advertencia:

¡ MUY IMPORTANTE ! Para cargar los programas es imprescindible desconectar el módulo Bluetooth, si lo dejamos conectado impedirá la carga a través del puerto serie (Tx y Rx).

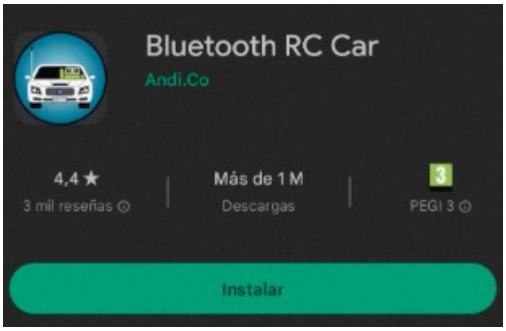
En el apartado de bloques de programación, se encuentra en "Comunicaciones".

La entrada Bluetooth dispone de los bloques que vemos en la imagen.

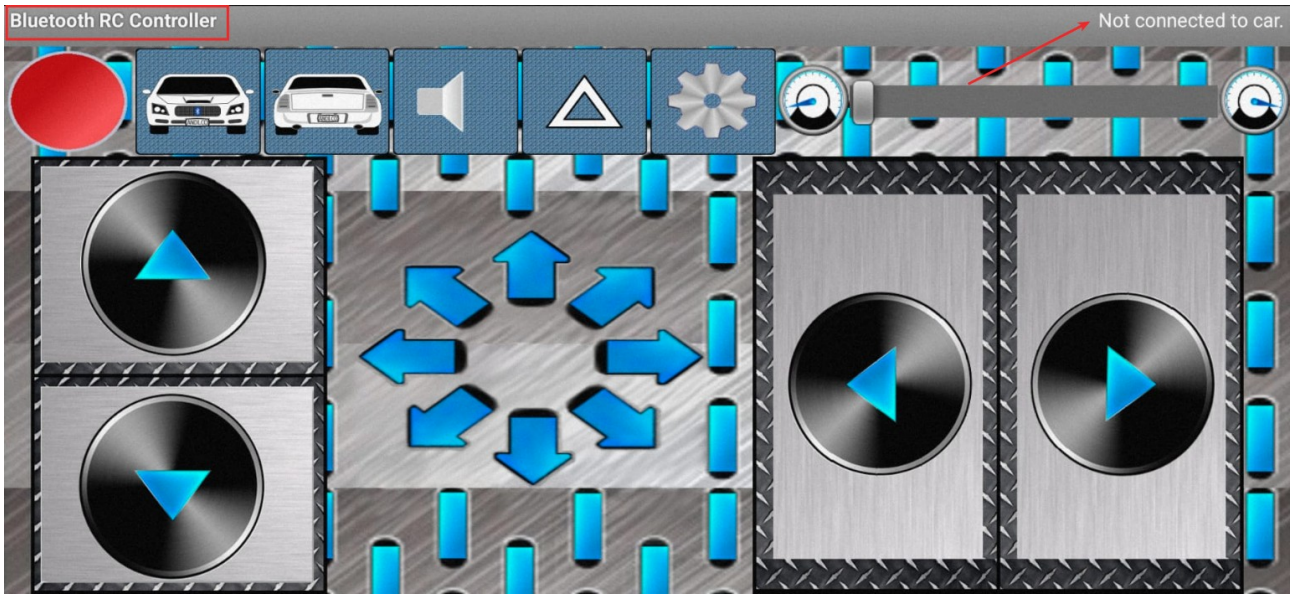
Un programa como el siguiente nos permitirá trabajar con comandos AT y el módulo Blutooth para poder configurarlo en sus distintas opciones.

Para trabajar en el control por Bluetooth hacen faltan dos cosas: la placa Easy Plug con el módulo Bluetooth conectado y un programa y un móvil Android con una aplicación que permita el control. Para ello podemos crear nuestra propia aplicación para Android utilizando por ejemplo APPInventor , que es una plataforma online en la que programar nuestras propias aplicaciones de modo muy similar a ArduinoBlocks.

Pero no vamos a entrar en esto por ahora y lo que haremos será utilizar una aplicación ya existente como Arduino Bluetooth RC Car. Este es el aspecto que presenta su búsqueda para instalarla:

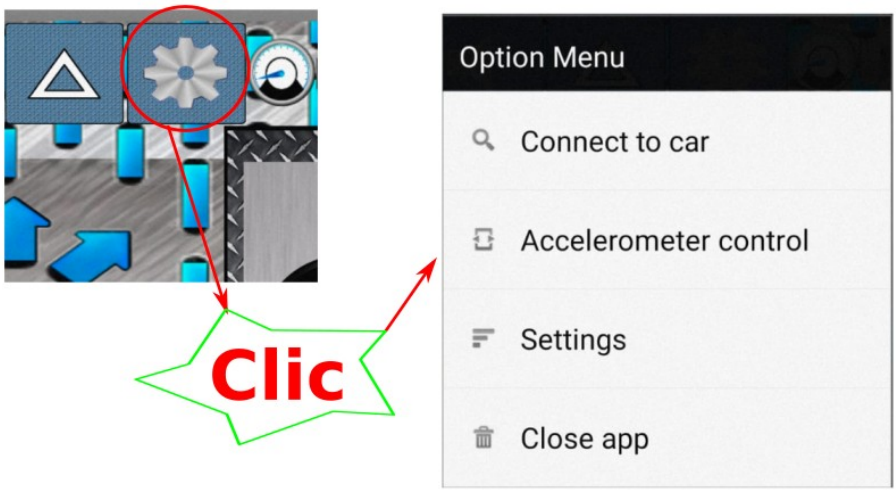


Una vez instalada e iniciada la aplicación nos muestra la siguiente ventana:

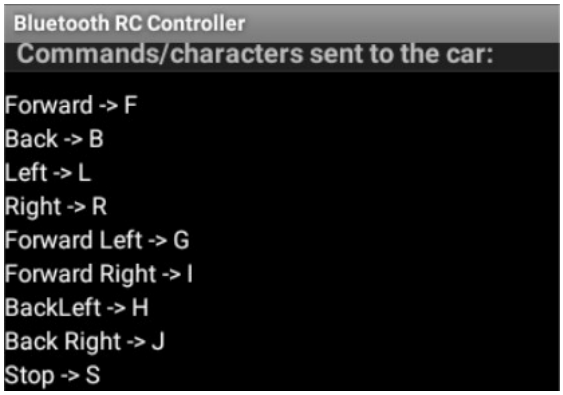


Aunque es una interfaz muy completa para el control de un coche es sencilla de utilizar e intuitiva, razón por la que la vamos a usar para probar nuestro Bluetooth.

Si hacemos clic sobre el botón del engranaje (menú) se nos abren las siguientes opciones:



Si clicamos sobre “Settings” nos aparece una ventana en la que se nos informa de los comandos/caracteres que enviamos desde la aplicación. En la imagen vemos parte de la información que nos da “Setting”.



Por ejemplo, cuando pulsamos el botón “Forward” (adelante) se envia el carácter “F” que en una letra del código ASCII por lo que en ArduinoBlocks debemos leer precisamente valores ASCII. El código ASCII pronunciado como “aski” (American Standard Code for Information Interchange o Código Estándar estadounidense para el Intercambio de Información) adoptado internacionalmente y que en sus primeros 128 símbolos básicamente define los números, letras mayúsculas y minúsculas y los principales signos de puntuación y que son comunes en casi todo el mundo. En la parte extendida es donde se definen los caracteres especiales de cada idioma, pero en esto ya no hay tanta unanimidad.

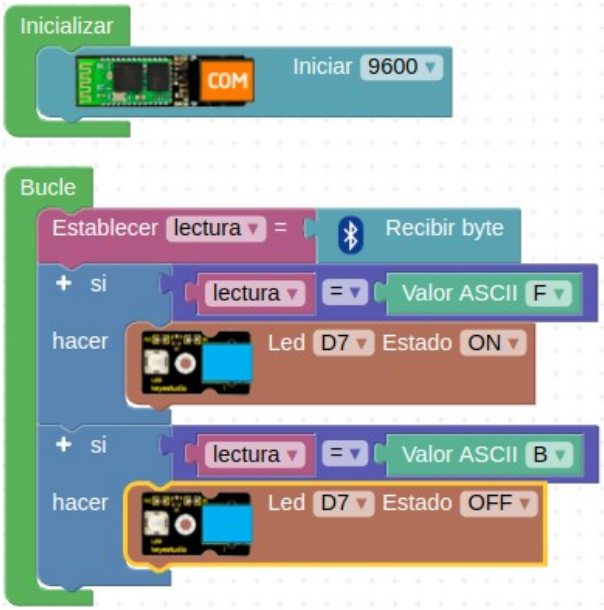
El bloque encargado de traducir los datos leídos en bytes es “Valor ASCII” y se encuentra al final del menú “Texto”.

Con el módulo Bluetooth conectado a alimentación realizamos las tareas de emparejamiento habituales entre el móvil y el módulo HC-06. Habitualmente la contraseña de emparejamiento del módulo es 1234.

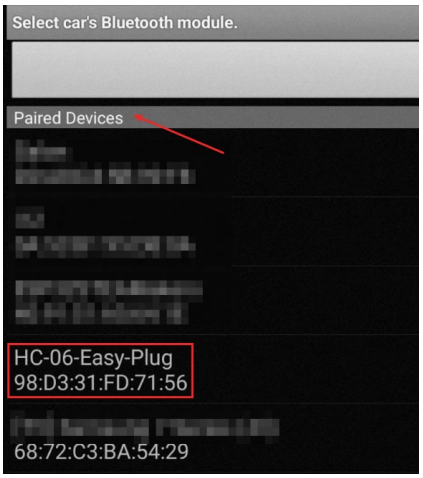
PRÁCTICA A36.1:

Vamos a enviar comandos desde la APP para recibirlos via Bluetooth.

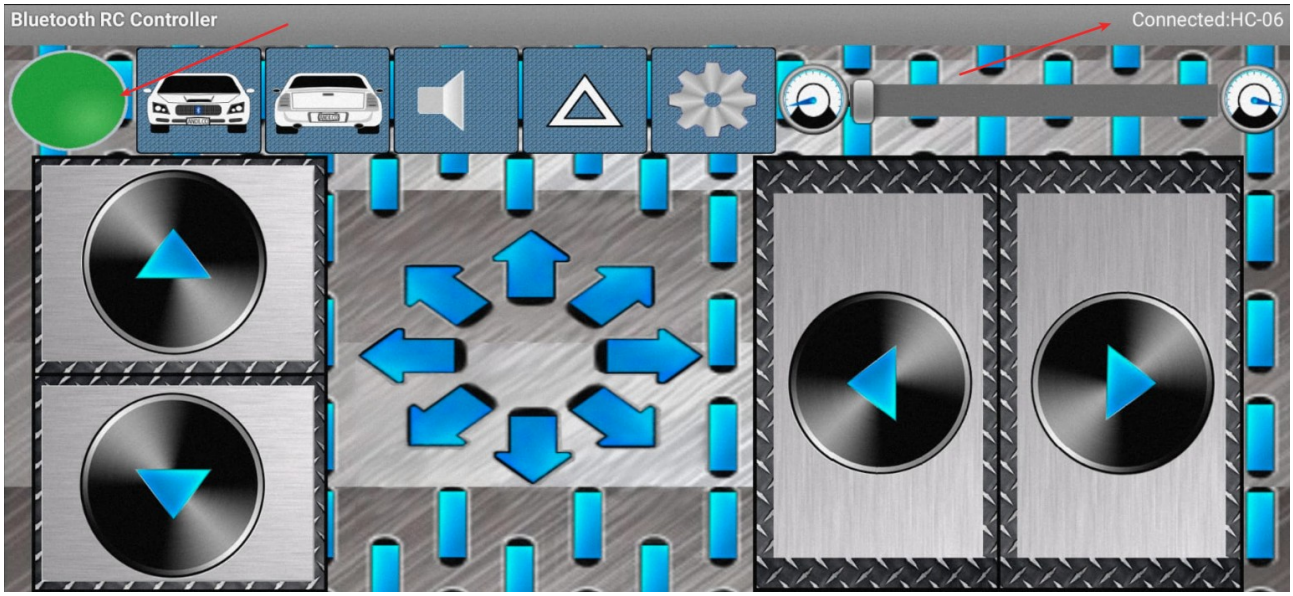
- Encender un LED utilizando el comando “Forward” y apagarlo con el comando “Back” .



Desconectamos el módulo Bluetooth y cargamos el programa en la placa. Volvemos a conectar el módulo HC-06 y desde el menú de la APP seleccionamos “Connect to car” para que se muestre la lista de dispositivos Bluetooth emparejados. Localizamos el módulo y lo seleccionamos.

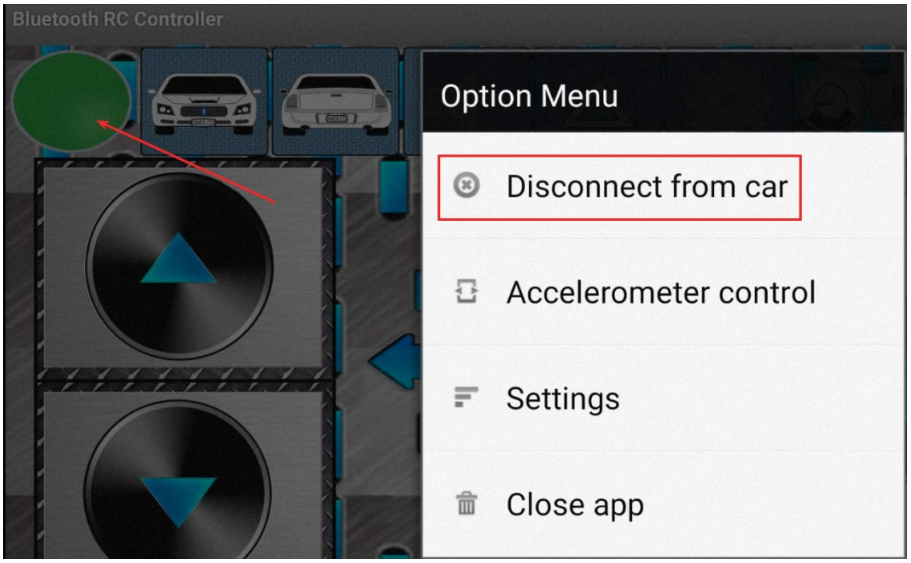


Tras unos instantes el circulo rojo que parpadea en la esquina superior izquierda se queda fijo y en color verde, la APP nos informa de que está conectada al HC-06 y el LED del módulo que estaba parpadeando se queda fijo.



Ya podemos probar los botones y comprobar el funcionamiento del programa.

Para dejar de estar conectados sin abandonar la APP podemos seleccionar “Disconnect from car” del menú. Lógicamente cuando cerramos la aplicación también se produce la desconexión.



A37: Módulo LED RGB con 4 LEDs Neopixel WS2812

¿Que es Neopixel? Neopixel es una marca creada por Adafruit Industries para referirse a algunos LEDs RGB que son direccionables individualmente, es decir LEDs que cuentan con un circuito lógico integrado dentro de si mismos, y este circuito es el que hace posible controlar con un solo pin digital el color de cada LED en una secuencia de LEDs encadenados en tiras, círculos o matrices. No todos los LEDs que son direccionables individualmente son Neopixel, solo lo son los basados en alguno de estos controladores: WS2812, WS2811 o SK6812.

Cada LED cuenta con los siguientes 4 pines:

- GND
- 5V – Vcc
- DIN – Pin que recibe la información del color
- DO – Pin que entrega la información del color

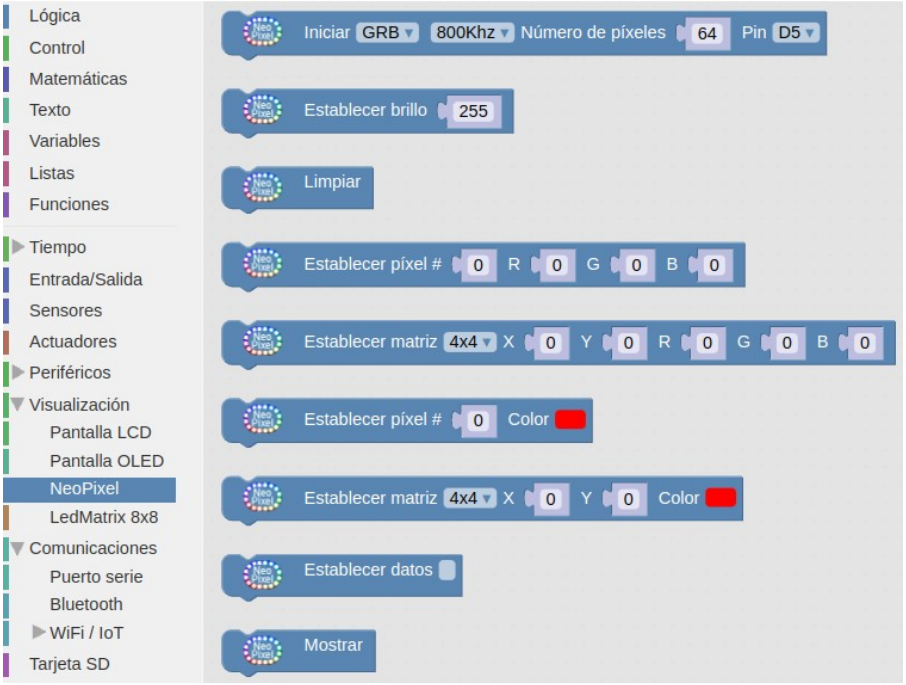
El módulo RGB basado en 2812 es una matriz de 2x2 formada por diodos LED 5050 y la circuiteria necesaria para ser direccionados. Se sincronizan gracias al oscilador interno de alta precisión de que disponen. Cada uno de los LEDs tiene una “dirección única” y es RGB, siendo todo gestionado por un solo cable de datos y el controlador integrado en el mismo elemento.

Sus principales características son:

- Potencia: 100 mW
- Fuente de luz: LEDs RGB 5050 SMD
- Modelo de circuito integrado: 4 x WS2812
- Niveles de gris: 256
- Ángulo de iluminación: 180°

Su aspecto es:

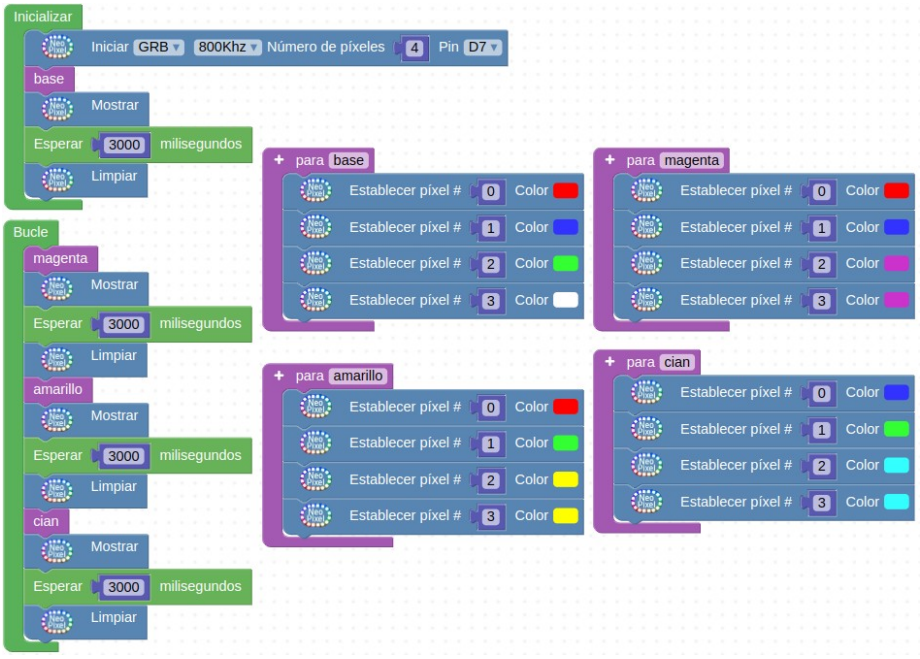
En el apartado de bloques de programación, se encuentra en "Visualización" y tiene su propio grupo de bloques.



PRÁCTICA A37.1:

Vamos a poner en funcionamiento el módulo matriz de 2x2 haciendo un test de colores.

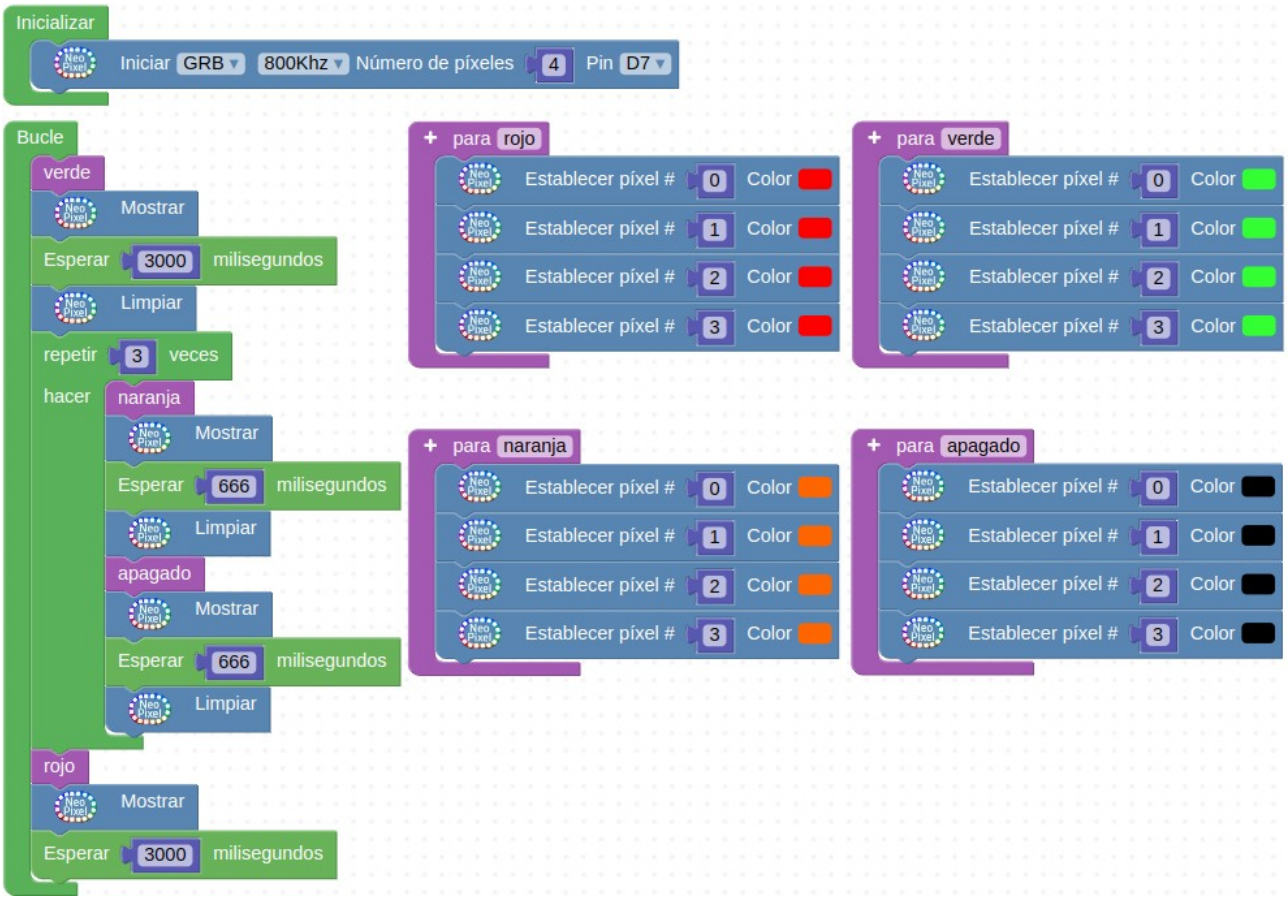
- Haremos un programa que realice un test de los cuatro diodos LED.



PRÁCTICA A37.2:

Una práctica de semáforos con la matriz de 2x2.

- Haremos un semáforo sencillo que permanentemente cambie el color entre los tres habituales.



A38: Módulo WiFi mas ESP01

En Easy Plug no existe como tal el módulo ESP-01 basado en el microcontrolador ESP8266 que permite realizar comunicaciones vía WiFi. Pero para facilitar la conexión de este módulo a la placa si se dispone de un zócalo que facilita su montaje.

En este apartado vamos a hablar de como enviar datos a la nube, empezando así a trabajar el Internet de las Cosas. La idea aquí no es profundizar en conceptos técnicos como Broker, MQTT, Mosquitto, etc. Pero si que vamos a iniciarnos en el tema IoT.

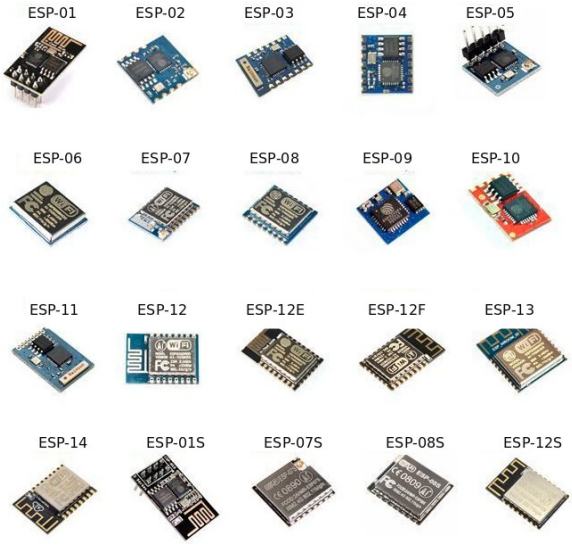
- **¿Qué es la Internet de las Cosas?**

La expresión “Internet de las cosas” o IoT (del inglés, Internet of Things), internet de todas las cosas o internet en las cosas, hace referencia al uso que hacen de Internet los dispositivos

conectados (las cosas) para comunicarse sin intervención humana directa. También se usa el término IoT para referirse al conjunto de estos dispositivos conectados comunicando "entre máquinas" (M2M) sin requerir interacción humana. No existe una Internet especial o diferente para "las cosas", en principio, se opera con las mismas redes que en otros usos de Internet. En la imagen siguiente se grafica la conexión de elementos con la nube a través de la red.

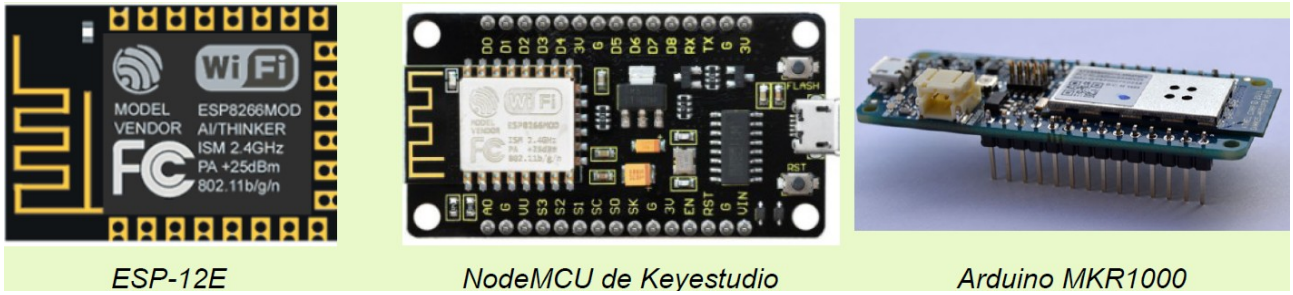
El IoT sirve para monitorización y control electrónico y toma de decisiones inteligente. Es decir, conectar dispositivos a Internet permite enviar y recibir información usando una infraestructura global y así poder monitorizar y/o controlar automáticamente y a distancia multitud de contextos. Por ejemplo, en entornos urbanos (ciudades inteligentes) se puede conocer la actividad humana, del medio ambiente, del tráfico… tanto para informar de su estado (como las plazas de aparcamiento disponibles, la densidad o velocidad de la circulación, la contaminación…) como para tomar decisiones de forma manual o automática para optimizar los recursos disponibles (iluminación, riego de jardines, funcionamiento de los semáforos…).

Para conectar a la red las placas Arduino, sus clones o basadas en el mismo se usan los módulos ESP8266. El ESP8266 es un chip de bajo costo WiFi que contiene su propio microcontrolador, un Tensilica L106 de 32-bit. Es un chip de bajo bajo coste y reducido tamaño.



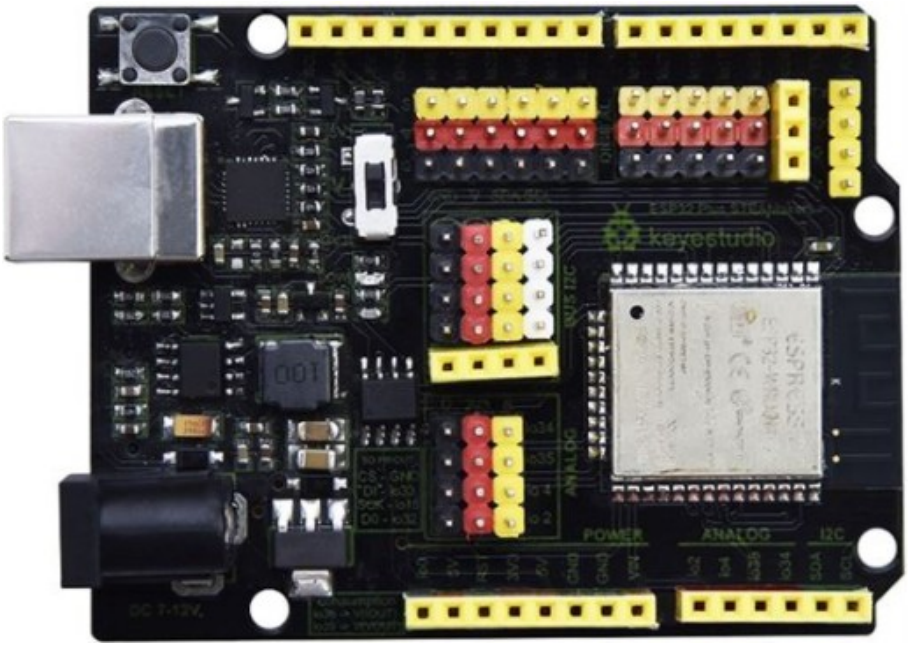
El chip ESP8266 forma parte de multitud de módulos comerciales a través de diferentes soluciones con distintas funciones, pines, tamaños y precios. El más simple es el ESP-01, que solo dispone de 2 puertos digitales y necesita un adaptador USB para poder ser programado y es el que vamos a usar aquí.

Hay otros muchos sistemas como por ejemplo ESP-12E de ESP8266, la placa de desarrollo NodeMCU, también basada en el ESP-12E, Además, hay placas con WiFi que usan otros chip WiFi diferentes, como la Arduino MKR1000.



ESP-12E *NodeMCU de Keyestudio* *Arduino MKR1000*

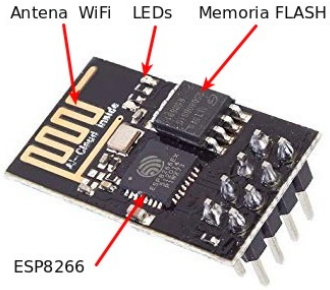
Un modelo destacable es la placa ESP32 STEAMakers basada en la CPU de dos núcleos Xtensa LX6 con arquitectura de 32 bits y una frecuencia de trabajo de 160 MHz que integra comunicaciones WiFi y Bluetooth y tiene el formato de un Arduino UNO.



- **Introducción al módulo ESP8266 ESP-01**

El módulo WiFi ESP-01 es uno de las más populares y económicos, pese a no ser el más potente ni versátil. El ESP-01 trae instalado una versión de firmware con la que podemos comunicarnos con el ESP8266 mediante comandos AT (los veremos a continuación) a través del puerto serie. Este tipo de comunicación nos va a permitir crear un puente entre la placa de control del proyecto y el ESP8266, consiguiendo así conectar a una red WiFi y dar un primer paso de gigantes en el mundo IoT.

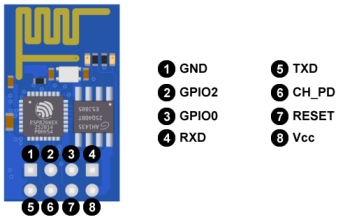
Veamos el aspecto del ESP-01 y algunas de sus características:



- ESP8266 es el microcontrolador del módulo ESP-01.
- La memoria flash es la BG25Q80A
- Los LEDs informan de si está encendido o no y de la transmisión de datos (Tx y Rx).
- La antena WiFi para la conexión a una internet.
- Los pines permiten conectar alimentación, sensores, …

Toda la información en el datasheet del fabricante

Los pines están distribuidos de la siguiente forma:



1 - GND

2 - Pin digital número 2

3 - Pin digital número 0

4 - RXD es el pin por donde se van a recibir los datos del puerto serie. Trabaja a 3,3 V. También puede ser el pin digital GPIO3

5 - TXD es el pin por donde se van a transmitir los datos del puerto serie. Trabaja a 3,3 V. También puede ser el pin digital GPIO1

6 - CH_PD es el pin para apagar y encender el ESP-01: si lo ponemos a 0 V (LOW) se apaga, y a 3,3 V (HIGH) se enciende.

7 - RESET pin a 0V resetea el ESP-01

- Vcc es el pin de alimentación. Funciona a 3,3V y admite un máximo de 3,6 V. La corriente suministrada debe ser mayor que 200 mA.

Datos obtenidos de https://programarfacil.com

GPIO (del inglés, General Purpose Input Output) son entradas o salidas de propósito general, o sea pines digitales.

El ESP-01 soporta comunicación I2C, por lo que, pese a tener solo un par de GPIOs, podemos conectarle multitud de sensores y actuadores a través del mencionado bus de datos I2C.

- **Programación del módulo ESP8266 ESP-01**

El ESP-01 dispone de un microcontrolador y una memoria donde poder almacenar programas, luego es un dispositivo programable en si mismo. Cargar programas en el dispositivo es algo mas complejo de lo que hemos visto hasta ahora dado que tiene dos modos de operación, el modo flash o de ejecución y el modo UART o de grabación y debemos ser nosotros quienes activemos un modo u otro, cosa que hasta ahora el entorno ArduinoBlocks ha sido el encargado de hacerlo. Los modos de operación se configuran a través de los puertos GPIO0 y GPIO2.

Para programar el ESP-01 hay que usar los pines Rx y Tx para transmitir los datos a la memoria Flash, donde se almacenará el sketch o programa.

En las placas de control los pines Rx y Tx están en los pines D0 y D1 respectivamente y también son los que se utilizan para cargar programas a una velocidad de 115200 baudios, así que si estos pines los ocupamos con el ESP8266 no podremos cargar programas en nuestra placa. Ahora bien, es posible utilizar otros pines para usar WiFi y evitar este problema, pero el resto de pines digitales solamente trabajan a 9600 baudios y el ESP-01 por defecto viene a 115200, luego para utilizarlo en pines distintos a los D0 y D1 de la placa tendremos que reprogramarlo para que la velocidad sea de 9600 baudios.

Aunque sea bastante técnico debemos exponer cómo configurar los dos modos de funcionamiento aunque sea de forma breve.

• Modo de funcionamiento UART. Para cargar un programa en el ESP-01 debemos o bien encenderlo o bien resetearlo pero teniendo los siguientes estados de pines:

◦ GPIO0 = 0 (nivel bajo o LOW = 0 V)

◦ GPIO2 = 1 (nivel alto o HIGH = 3,3 V = Vcc).

Recordemos siempre que el ESP8266 trabaja con niveles lógicos de 3,3 V.

El pin GPIO2 está por defecto a HIGH, ya que tiene un pull-up interno, por lo que podemos dejarlo simplemente desconectado.

• Modo de funcionamiento Flash. Para ejecutar un programa en el ESP-01 una vez cargado debemos tener la siguiente configuración de pines:

◦ GPIO0 = 1

◦ GPIO2 = 1

Tanto el GPIO0 como el GPIO2 están por defecto a HIGH, ya que ambos tienen un pull-up interno, por lo que podemos dejarlos simplemente desconectados.

En este momento reflexionamos sobre el uso de Rx y Tx ya que si los estamos utilizando para cargar el programa y en la placa y GPIO0 y GPIO2 para indicar el modo de trabajo ¿cómo conectamos los sensores y actuadores al ESP-01?. Veamos:

1. Rx y Tx los utilizamos para cargar el programa. Una vez finalizada la carga los podemos utilizar como pines de entrada y salida digitales.

2. Los modos de trabajo se indican cuando se resetea o reinicia la placa. Una vez que tengamos el modo de ejecución podemos conectar cualquier componente a estos pines.

Los programas los podemos subir a nuestro ESP-01 mediante el IDE de Arduino, que no vamos a explicar aquí, o por medio de un convertidor USB-serie igual o similar al de la imagen.

Se trata de un escudo o shield para el módulo WiFi ESP-01 que está provisto de un chip conversor de USB a puerto serie, en concreto el CH340G.

El proceso de reprogramación con esta placa es bastante sencillo pero requiere de comando AT que pasamos a introducir antes de continuar.

- **Comando AT en el ESP8266**

El ESP-01 viene por defecto con el firmware AT ai-thinker V0.9.2.4.

Los módems venían con un conjunto de comandos que permiten que nos podamos comunicar con ellos para configurarlos y que lo podamos hacer a través del puerto serie de ordenador al que están conectados. A estos comandos se les llama AT (de attention).

Después de cada comando AT, el ESP8266 espera los caracteres especiales de nueva linea <CR><LF> para ejecutar el comando. El carácter no imprimible CR (del inglés, Carriage Return) significa retorno de carro y LF (del inglés, Line Feed) es salto de línea. El origen de la nomenclatura está en las máquinas de escribir.

En el enlace tenemos un pdf de la empresa (Espressif Systems) con el juego de comandos AT para el ESP8266.

En la tabla siguiente damos un resumen de los comandos con una información ampliada de los que vamos a usar para la configuración inicial que necesitamos.

Comando	Respuesta	Parámetros
AT - Probar iniciación correcta		
AT	OK	—
AT+RST - Reinicia el módulo		
AT+RST	OK	—
AT+UART_DEF – Configuración por defecto grabada en la memoria flash		
AT+UART_DEF? Consulta configuración	+UART_DEF:<baudrate>,<databits>,<stopbits>,<parity>,<flow control> OK	• <baudrate>: velocidad UART en baudios • <databits>: bits de datos ‣ 5: bit de datos 5 ‣ 6: bit de datos 6 ‣ 7: bit de datos 7 ‣ 8: bit de datos 8 • <stopbits>: bits de parada ‣ 1: bit de parada 1 ‣ 2: bit de parada 1.5 ‣ 3: bit de parada 2 • <parity>: bit de paridad ‣ 0: Ninguna ‣ 1: Impar ‣ 2: Par • <flow control>: control de flujo ‣ 0: no habilitado ‣ 1: habilitado RTS ‣ 2: habilitado CTS ‣ 3: habilitados ambos RTS y CTS
AT+UART_DEF=<baudrate>,<databits>,<stopbits>,<parity>,<flow control> Establece configuración	OK	

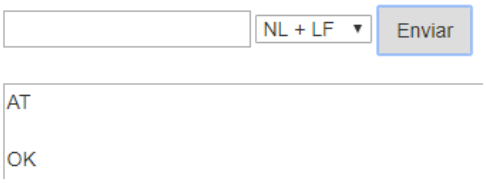
- **Preparación del módulo ESP-01**

Primero conectamos el módulo ESP8266 a la shield USB-serie y este a su vez a un puerto USB de nuestro ordenador. Nos aseguramos de que el interruptor está en modo Flash Boot.

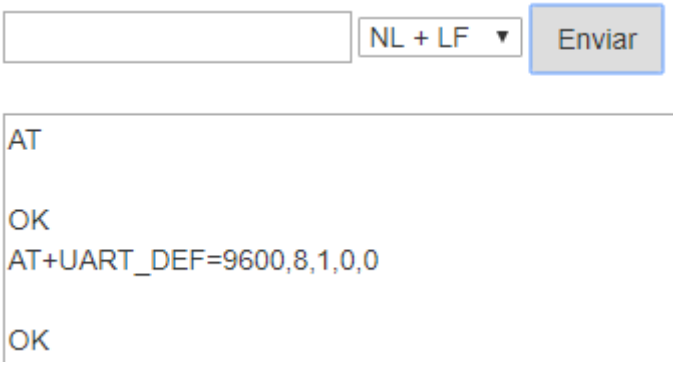
Entramos en ArduinoBlocks con ArduinoBlocks-Connector en funcionamiento, realizamos un refresco de la lectura del puerto para que detecte la shield del módulo WiFi si es necesario, abrimos la consola, escogeremos la opción de 115200 en baudrate y también la de NL + LF (NL + LF es igual que CR + LF) para comunicarnos con el módulo WiFi.

Hacemos clic en Conectar y conectamos la consola, escribimos "AT" en ella y clic en enviar. La situación es la de la imagen siguiente:

Si todo es correcto debe respondernos "OK". Si responde algo sin sentido o no contesta, significa que está configurado en alguna otra velocidad. En este segundo caso deberemos cambiar la opción de baudrate y repetir la operación con diferentes velocidades hasta que nos responda "OK". La situación correcta en la consola de comandos es la siguiente:



Una vez que nos responda "OK", le enviamos el texto "AT+UART_DEF=9600,8,1,0,0" y nos debe responder otra vez "OK".



Con esto hemos cambiado la velocidad a 9600 baudios mediante el comando AT+UART_DEF que nos permite cambiar la definición a 9600 baudios, con 8 bits de datos, 1 bit de parada, sin paridad y sin habilitar el control de flujo.

Una vez que hemos configurado el módulo WiFi a la velocidad que nos interesa debemos crear un canal en un servidor MQTT donde enviar los datos y visualizarlos. Esto lo vamos a hacer a través de la web de https://thingspeak.com, pero antes de nada vamos a describir que es MQTT.

- **Una introducción al IoT a través de MQTT**

Si queremos trabajar en tareas de IoT debemos de utilizar algún protocolo de comunicación y hoy por hoy el principal es MQTT, pero antes de citar protocolos vamos a tratar algunos conceptos necesarios. Un protocolo de comunicación no es otra cosa que una serie de normas definidas para que dos o más dispositivos puedan comunicarse entre si de forma comprensible para ambos.

  - **Requisitos del IoT**

Estamos bastante acostumbrados a realizar un tipo de comunicación denominada M2M (machine-to-machine) utilizando internet, pero cuando trabajamos en IoT debemos establecer una serie de requisitos que hacen que la comunicación M2M no sea la mas adecuada.

Algunos de estos requisitos son:

• Cantidad: se puede llegar a tener un gran número de dispositivos diferentes, desde sensores, actuadores, servidores, etc.

• Escalabilidad: los sistemas deben permitir añadir o eliminar dispositivos sin que el sistema global resulte afectado.

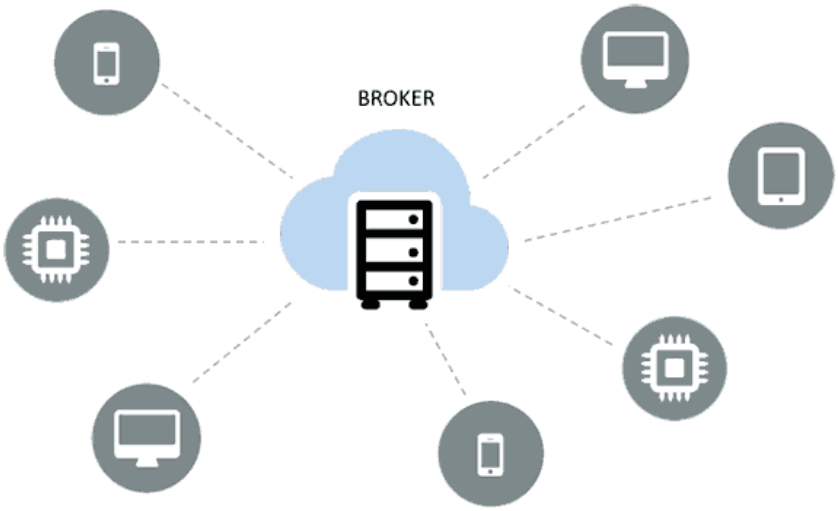
• Variedad: normalmente necesitaremos que el sistema funcione con diferentes sistemas operativos, lenguajes de programación y el mayor número posible de dispositivos.

• Simultaneidad: gran cantidad de comunicaciones simultaneas, lo que requiere respuestas rápidas para lo que es necesario que los mensajes transmitidos sean lo mas cortos posibles.

• Seguridad: internet no es un sitio muy seguro y estos dispositivos van a estar conectados a internet controlando dispositivos físicos.

• Accesibilidad: tendremos que trabajar en condiciones muy variadas en lo que se refiere a ancho de banda, firewall, direccionamiento,…

La solución mas común consiste en disponer un servidor denominado ‘broker’, o a veces ‘Router’, que será el que reciba los mensajes de los dispositivos emisores y los distribuirá a los receptores.



El broker va a tener una dirección IP fija y será accesible para todos los dispositivos, puede mantener un registro de los dispositivos conectados, recibir y distribuir mensajes y establecer filtros de destinatarios. Esto permite algo fundamenta, y es que cada dispositivo no dependa del resto de dispositivos.

Veamos algunos conceptos que nos permitan entender las infraestructuras de los servicios IoT:

• Message Queue o cola de mensajes. En este tipo el broker genera una cola de mensajes única para cada uno de los clientes que inician la subscripción.

• Message Service o servicio de mensajería. En este tipo el broker distribuye inmediatamente los mensajes filtrados por algún criterio a los clientes conectados. A diferencia de Message Queue, los mensajes entregados mientras el cliente está desconectado se pierden.

• Publish/Susbcribe (PubSub). Se trata de un sistema de mensajería donde el 'Subscriber' informa al broker de que quiere recibir un tipo de mensajes y el 'Publisher' entonces puede publicarlos.

• Router Remoder Procedure Calls (RRPC). Se trata de la ejecución remota de procedimientos donde 'Callee' comunica al broker que proporciona un procedimiento y el 'Caller', puede llamar a este procedimiento.

Algunos de los protocolos para IoT son:

• MQTT (del inglés Message Queuing Telemetry Transport) es un protocolo PubSub de servicio de mensajería que actúa sobre TCP. Es ligero y fácil de implementar resultando apropiado para dispositivos de baja potencia, tan habituales en IoT. TCP es un protocolo de control de transmisión (del inglés, Transmission Control Protocol) fundamental en Internet.

• AMQP (Advanced Message Queuing Protocol) es un protocolo PubSub de cola de mensajes que asegura la confiabilidad e interoperabilidad necesaria en aplicaciones corporativas.

• STOMP (Streaming Text Oriented Messaging Protocol). Es un protocolo sencillo que emplea HTTP y mensajes de texto.

• XMPP (Extensible Messaging and Presence Protocol). Es un protocolo abierto basado en XML diseñado para aplicaciones de mensajería instantánea.

• WAMP (Web Application Messaging Protocol). Es un protocolo abierto que se ejecuta sobre WebSockets, y provee tanto aplicaciones de PubSub como RRPC.

• CoAP (Constrained Application Protocol) es un protocolo pensado para emplearse en dispositivos de IoT de baja capacidad.

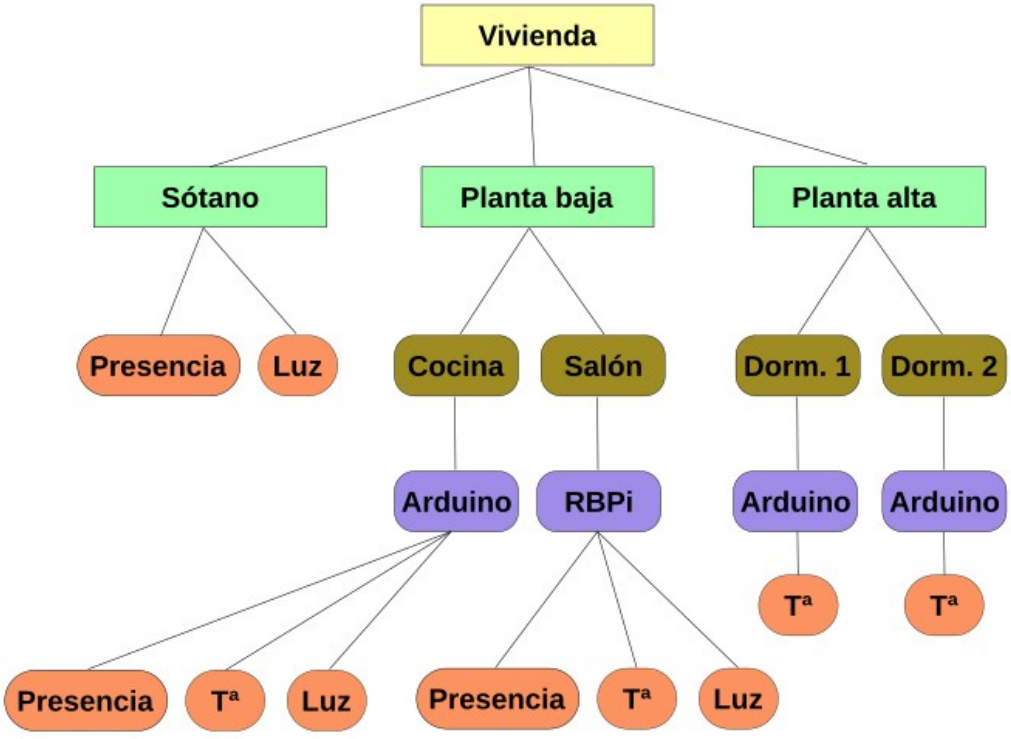
- **MQTT**

Dentro de una arquitectura de MQTT, es muy importante el concepto topic (tema en español) ya que la comunicación se realiza a través de topics debiendo estar los emisores y receptores subscritos a un topic común para poder establecer la comunicación.

Este tipo de arquitectura permite que la comunicación pueda ser de uno a uno o de uno a muchos.

Los topics tienen estructura jerárquica pudiendo establecer relaciones padre-hijo de manera que cuando nos suscribimos a un topic padre podemos recibir también la información de sus hijos.

En un ejemplo lo podemos ver más claramente.



Un topic se representa mediante una cadena con las jerarquias separadas por /. Por ejemplo:

- Vivienda/Planta baja/Cocina/Arduino/Luz
- Vivienda/Planta alta/Dorm.1/Arduino/Temperatura

De esta forma podemos suscribirnos a un topic concreto o a varios, por ejemplo:

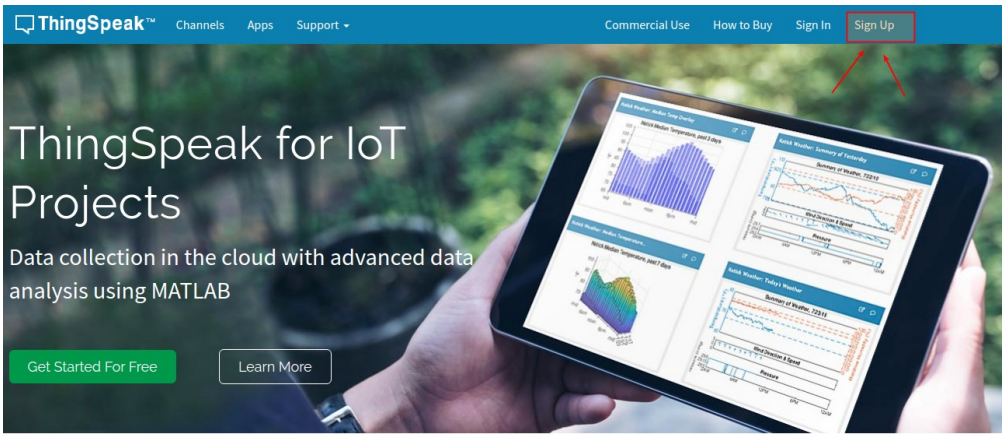
- Un topic: Vivienda/Planta baja/Cocina/Arduino/Luz
- Varios topics: Vivienda/Planta baja/#

Existen básicamente tres tipos de brokers, los privados, los públicos y los locales. A continuación citamos alguno de los más utilizados:

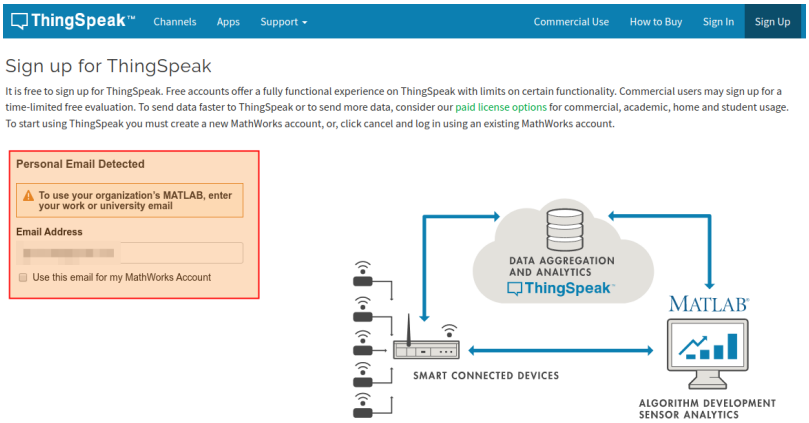
1. Private MQTT Broker: solamente los dispositivos que establezcamos pueden publicar o suscribirse a un topic. Se utiliza en producción y prototipado. Algunos de ellos son:
   - Azure de Microsoft
   - AWS de Amazon
   - CloudMQTT disponible en: https://www.cloudmqtt.com/plans.html
   - ThingSpeak de Mathworks (MATLAB)
2. Public MQTT Broker: cualquier dispositivo puede publicar y suscribirse a topics. Algunos de ellos son:
   - Eclipse: Enlace - Dirección del broker
   - Mosquitto: Enlace - Dirección del broker
   - HiveMQ: Enlace - Dirección del broker
   - Thingspeak: Enlace – Dirección del broker
3. Si queremos instalar nuestro propio broker en una raspberry o PC, sin duda Mosquito es la opción más extendida.

- **Creación de un canal en un servidor MQTT**

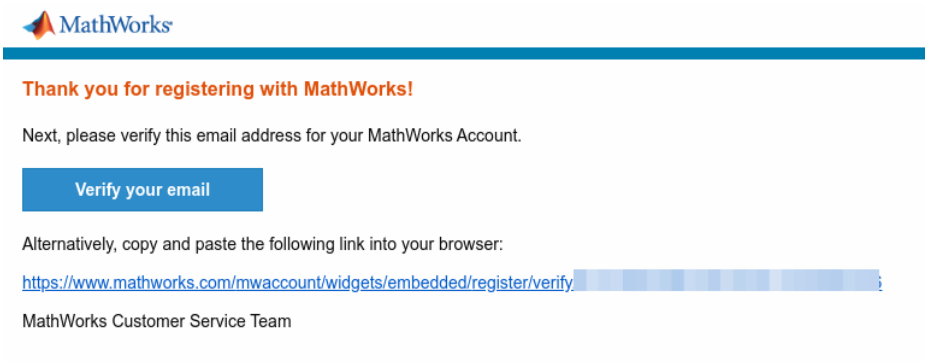
En nuestro caso vamos a utilizar Thingspeak en su versión pública. Lo primero que debemos hacer es crear una cuenta, para lo que clicamos en Sign Up. Seguidamente, introducimos un correo electrónico válido y el resto de datos que nos pide. Lo vemos en la imagen siguiente:



Debemos aceptar que use la dirección de correo electrónico para nuestra cuenta de MathWorks, como se ve en la imagen siguiente:

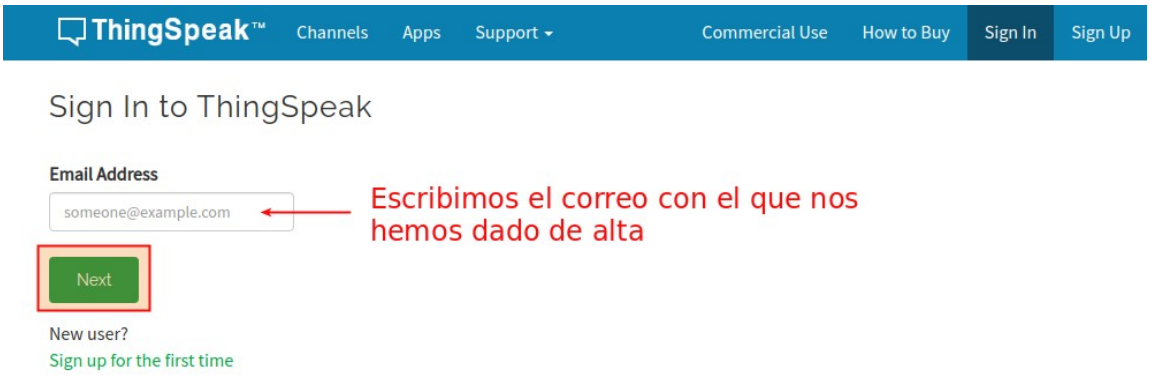


Tras esto recibiremos un correo en la dirección que hemos dado para confirmar la creación de la cuenta y confirmarla. Debemos ver algo similar a la imagen siguiente:

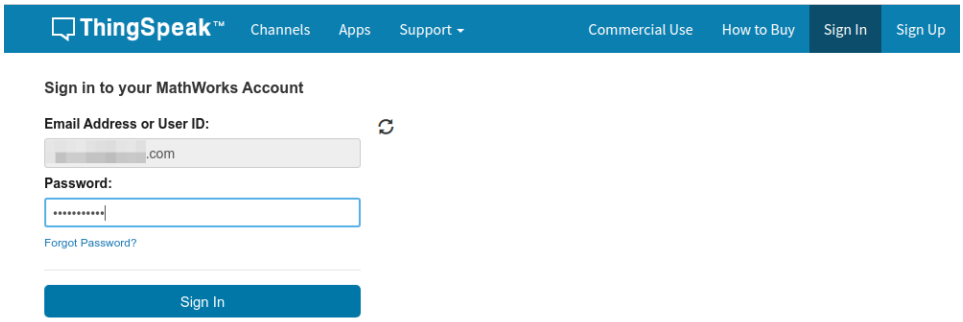


Un vez confirmada la dirección de correo electrónico, volvemos a la página donde estábamos y hacemos click en continuar. Nos pedirá un nombre de usuario y una contraseña que usaremos a partir de ahora para iniciar sesión.

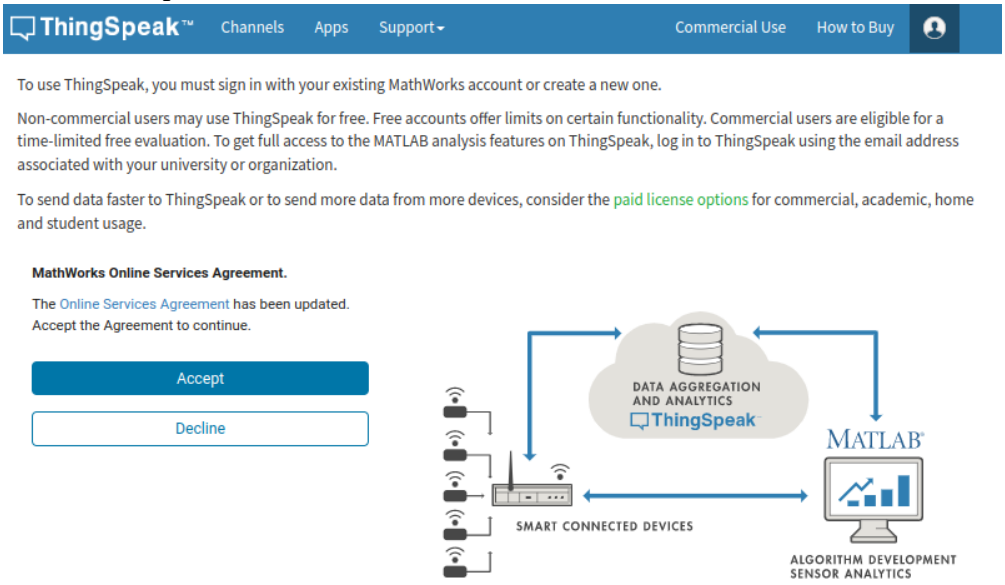
Finalizada la creación de la cuenta iniciamos sesión desde la página principal haciendo clic en “Sign In”. Veremos una ventana como la de la imagen siguiente:



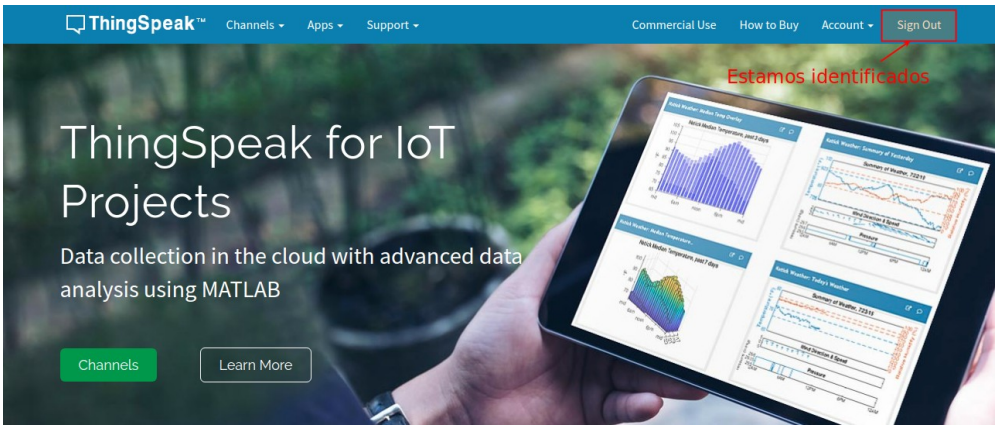
Escribimos nuestra contraseña y hacemos clic en el botón Sign in.



Se nos mostrará una página con información de uso y los acuerdos de MathWorks para sus servicios online que debemos aceptar.

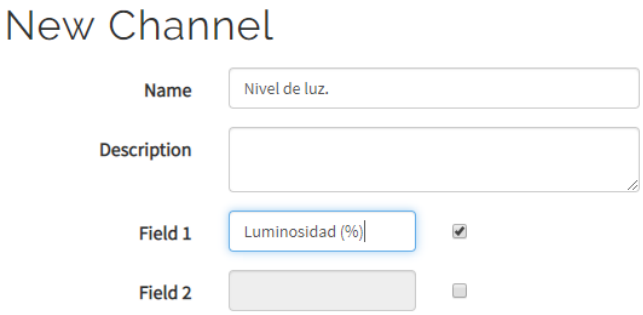


Con esto entramos en el broker y está todo listo para iniciar el trabajo.



- **Pasos iniciales con MQTT**

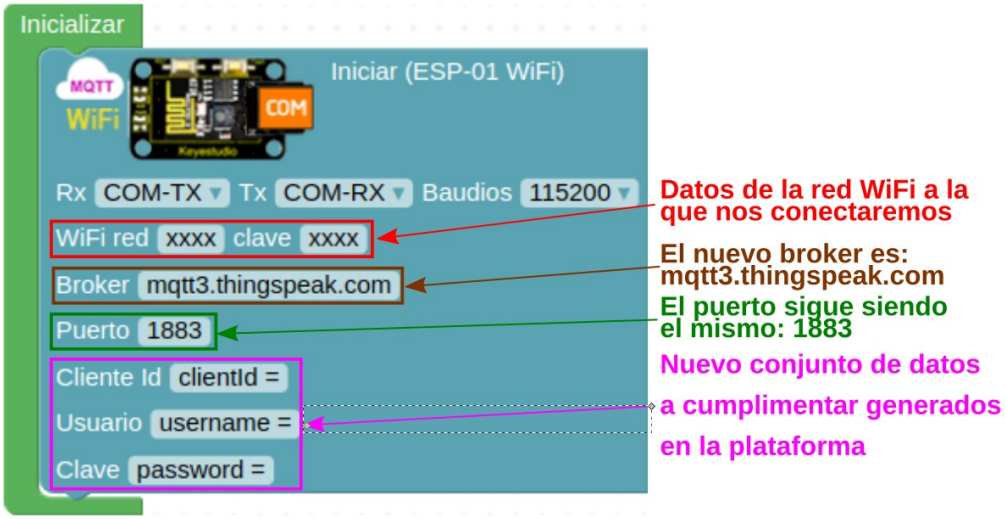
En ThingSpeak el funcionamiento está basado en canales por lo que, una vez logueados, lo primero que vamos a hacer es crear un nuevo canal con las siguientes características:



Si nos desplazamos hacía abajo en la misma página encontramos este botón:

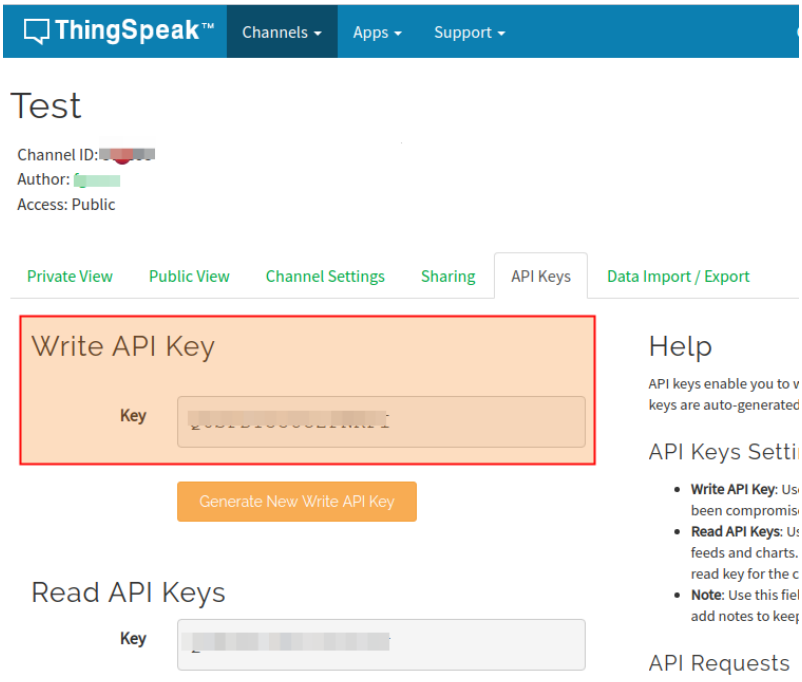
Y será en este canal donde veremos los datos una vez hagamos nuestro programa en ArduinoBlocks.

El bloque de inicialización de la conexión con el broker para la versión Easy Plug lo vemos en la imagen siguiente:



El bloque para publicar datos ahora simplemente lleva el número de identificación del canal (Channel ID) y el campo (Field) que corresponda con la variable a publicar que seleccionamos con el desplegable. Es muy importante respetar el orden de los campos establecido (según los programas a importar y la comunicación de correo realizada para comunicar los datos de configuración) para que los resultados sean correctos.

Cuando entramos en uno de los canales creados en el broker disponemos de una serie de pestañas para trabajar y configurar dicho canal, en concreto en la pestaña API Keys encontraremos la información necesaria para el bloque anterior, es decir la ID del canal y la llave API para publicar, tal y como vemos en la imagen siguiente:



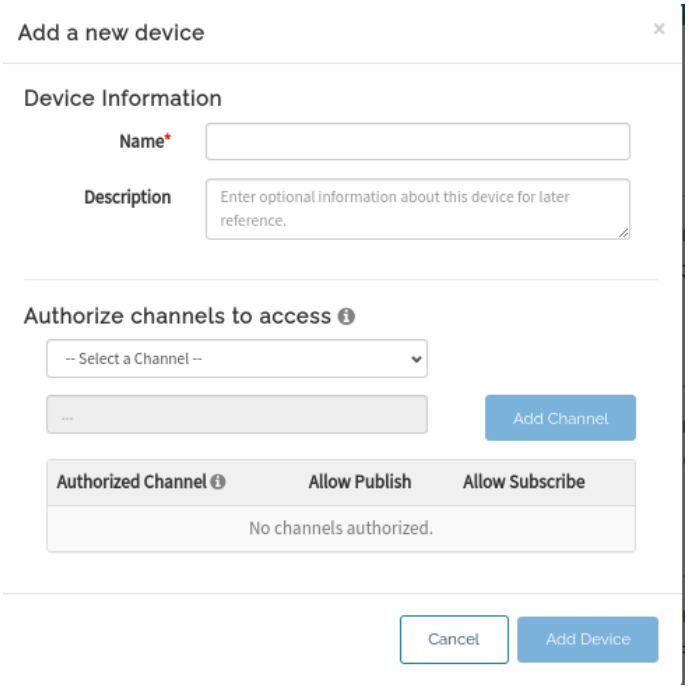
Entre cada envio de datos debemos poner un bloque esperar de como mínimo 16 segundos. Esto es muy importante porque la versión gratuita de ThingSpeak sólo permite subir datos con ese intervalo de tiempo.

- **Añadir canales a MQTT en ThingSpeak**

El proceso es sencillo y se lanza desde el menú horizontal de ThingSpeak, tal y como vemos en la imagen siguiente:

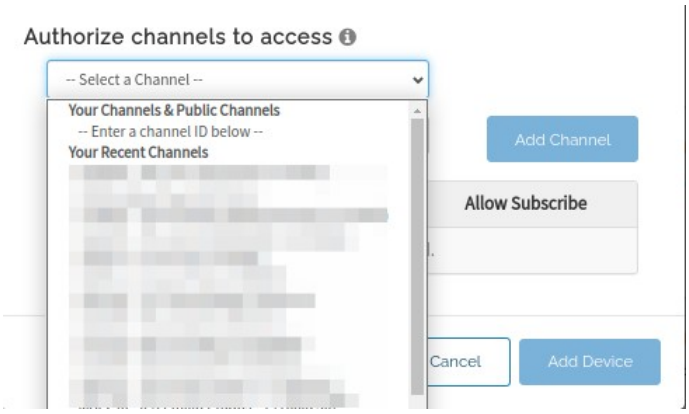
Es conveniente copiar o guardar los datos indicados para tenerlos a mano, en especial la contraseña, porque en el momento que salgamos de esta ventana ya no la podremos recuperar aunque si regenerar.

Tras hacer clic en MQTT se nos mostrará una nueva ventana con los canales añadidos (en blanco si no hemos añadido ninguno) y un botón verde con la leyenda “Add a new device” que nos va a permitir añadir nuevos dispositivos. En la imagen siguiente vemos la ventana citada.



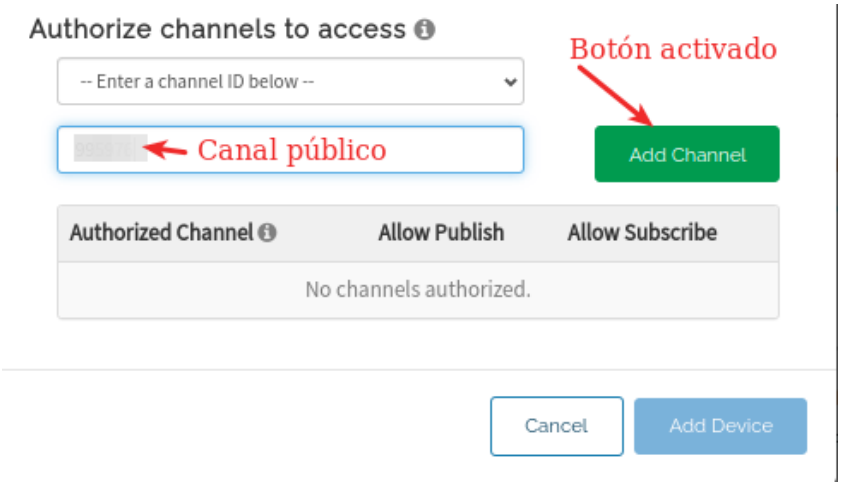
En “Device information” damos los datos identificativos del dispositivo MQTT, siendo “Name” un campo obligatorio a cumplimentar y “Description” un campo opcional para dar mas detalles del dispositivo.

La parte realmente importante es “Authorize channels to access que vemos seguidamente. Si hacemos clic en el desplegable se nos mostrará una ventana con el listados de dispositivos candidatos en “Your Recent Channels”. Si el canal que queremos añadir está en la lista lo seleccionamos con un clic y continuamos el proceso como veremos posteriormente.

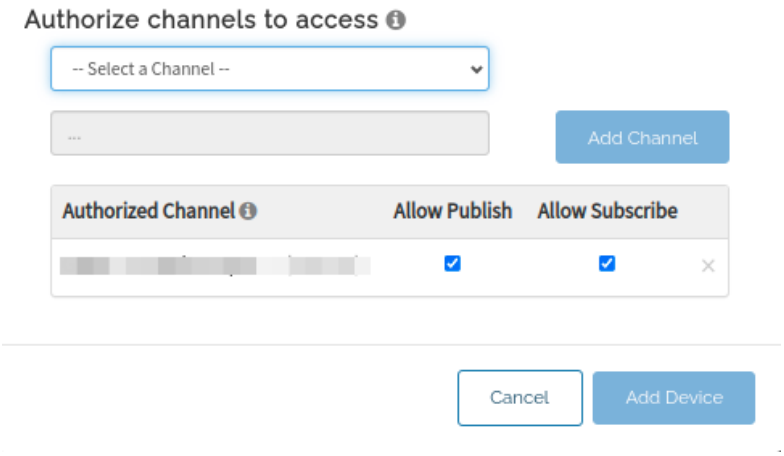


Si el canal no está en la lista escogemos la opción marcada en la imagen siguiente:

En la ventana que se muestra ahora tenemos habilitado el campo para introducir un “Channel ID” válido que nos habilitará el botón “Add Channel”.



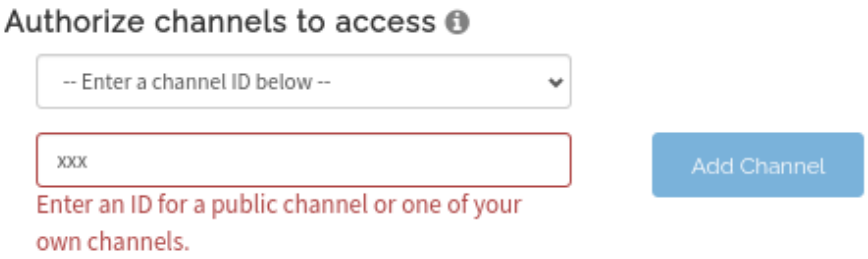
Al hacer clic en el botón verde se añade el canal como dispositivo MQTT y veremos algo similar a la ventana siguiente:



La configuración por defecto permite publicar en el dispositivo “Allow Publish” y permite suscripciones “Allow Subscribe” y ambos podemos deseleccionarlos o dejarlos como están según nuestras necesidades. El aspa a la derecha nos permite eliminar el canal como dispositivo MQTT.

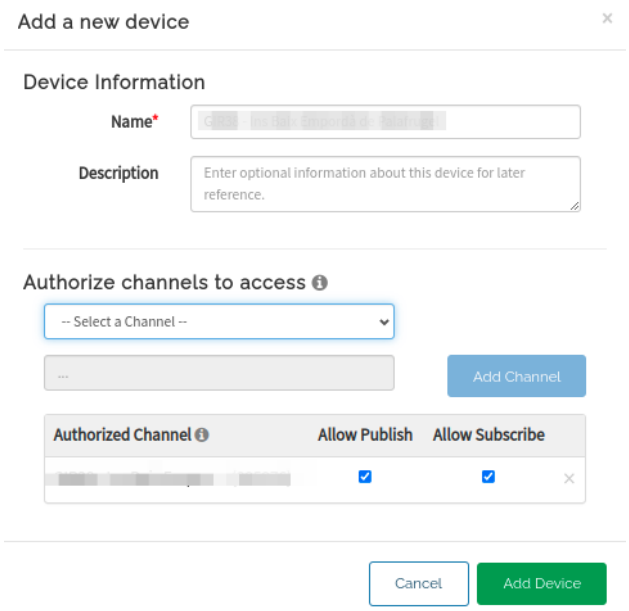
El proceso cuando seleccionamos un canal de la lista que nos muestra es totalmente similar al descrito cuando introducimos un número válido de canal.

Si en el campo para introducir un ID de canal tecleamos algo no válido se mostrará un mensaje como el que vemos en la imagen siguiente y el botón “Add Channel” permanece inactivo.

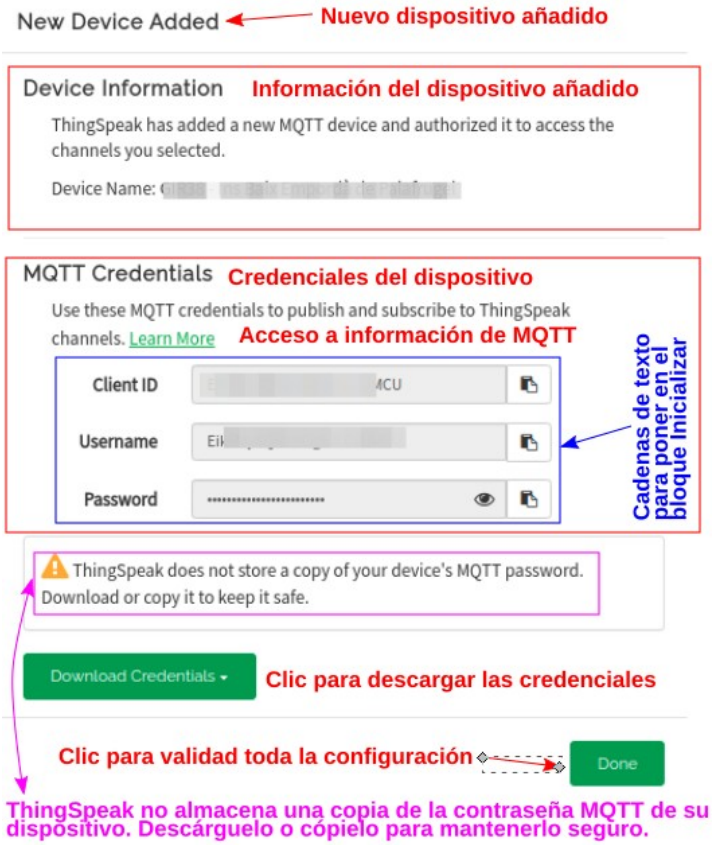


Literalmente el error nos indica que introduzcamos el ID de un canal público o el de uno de nuestros propios canales.

Cuando toda la información introducida es correcta se activa el botón “Add Device” (ver imagen siguiente) y si hacemos clic el canal queda añadido como dispositivo MQTT en ThingSpeak.



Tras hacer clic en el botón “Add Device” se nos muestra una ventana como la de la imagen siguiente, donde se han realizado las descripciones pertinentes de sus apartados.



El nuevo dispositivo se mostrará en la venta “MQTT Devices” de la forma que vemos en la imagen siguiente:

En esta ventana podemos eliminar el dispositivo o editar su configuración, para, por ejemplo regenerar la contraseña.

- **Configuraciones en ArduinoBlocks**

El primer bloque que vamos a configurar es el que tenemos en Inicializar. Es el bloque encargado de establecer la configuración de nuestra conexión WiFi con el módulo ESP8266 que vamos a utilizar.

Configuramos el bloque de la siguiente forma:

• WiFi red: debemos poner el nombre de nuestra conexión WiFi

• Clave: tecleamos la contraseña de acceso a nuestra red WiFi

• Broker: tecleamos el nombre del broker que es mqtt3.thingspeak.com

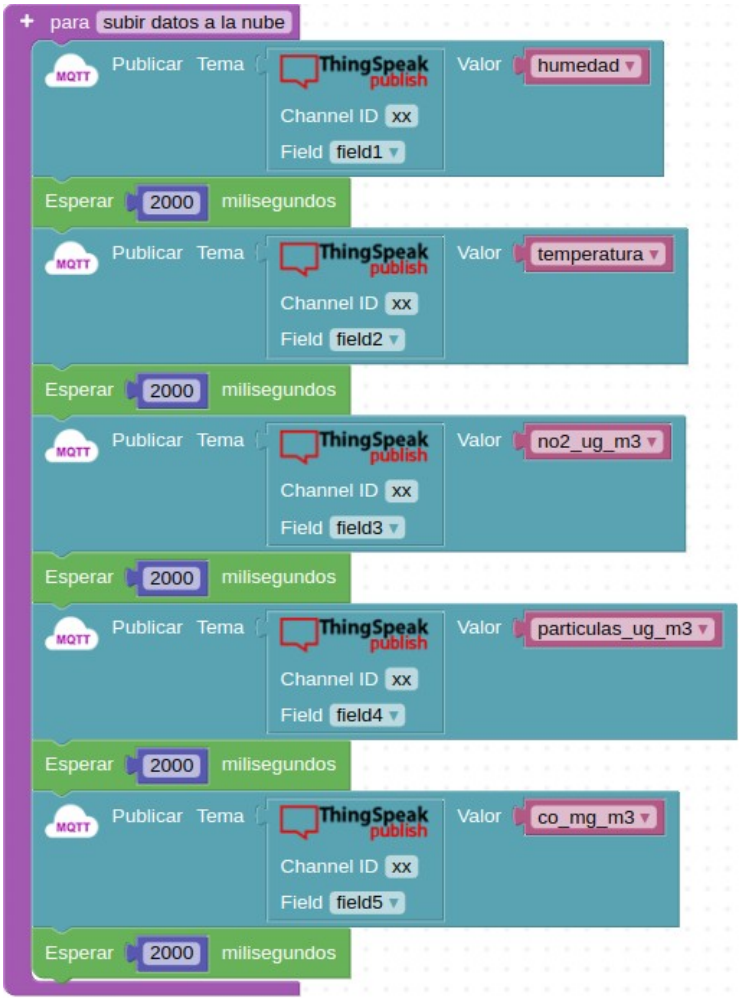
• Cliente Id: sustituimos “clientId =” por el valor generado en la plataforma al crear el dispositivo MQTT “Devices -> MQTT y Add a new device”.

• Usuario: sustituimos “username =” por el valor generado en la plataforma.

• Clave: sustituimos “password =” por el valor generado en la plataforma.

El resto de casillas las dejamos como vemos en la imagen anterior.

En segundo lugar, configuramos los bloques contenidos en la función subir datos a la nube. El bloque Publicar Tema solamente requiere del identificador del canal “Channel ID” y que seleccionemos el campo para cada caso tal y como se muestra en la imagen siguiente:



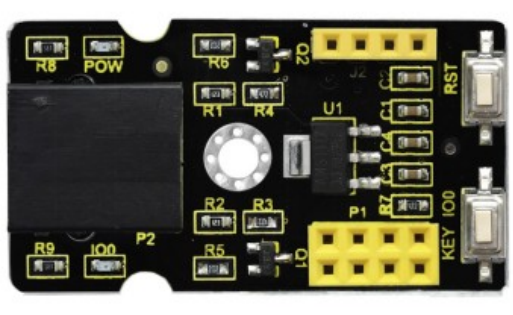
Debemos configurar para cada parámetro a subir:

• Channel ID: Es el número identificador del canal en el que vamos a escribir los datos que enviemos.

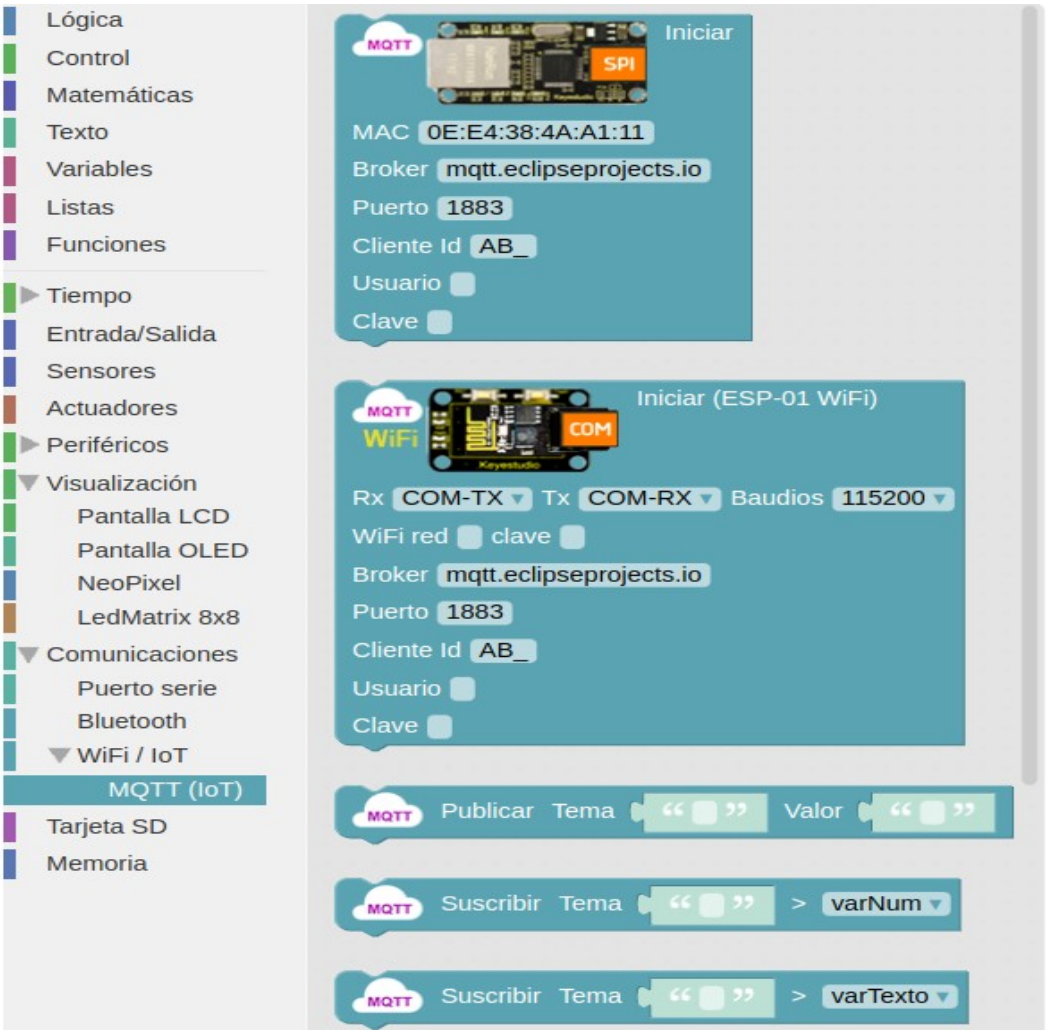
• Field: Es el campo seleccionado para mostrar.

Ambos datos se obtienen de la información suministrada por el servidor ThingSpeak para cada canal y no serán alterados salvo que se detecte una total inactividad del canal, bien por estar desatendido o bien porque hemos dejado de utilizarlo.

El aspecto de los elementos a utilizar para la conexión WiFi es:



En el apartado de bloques de programación, se encuentra en "Comunicaciones → WiFi → MQTT (IoT)".



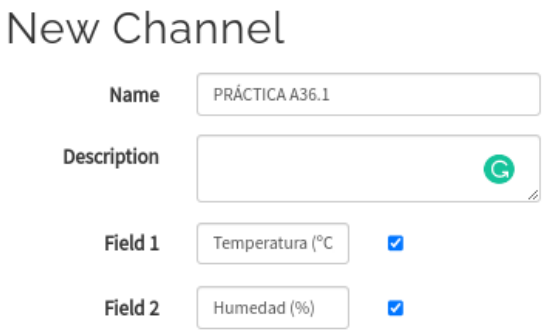
PRÁCTICA A38.1:

Vamos a medir la temperatura ambiente y la humedad de una habitación y publicarlas en Thingspeak.

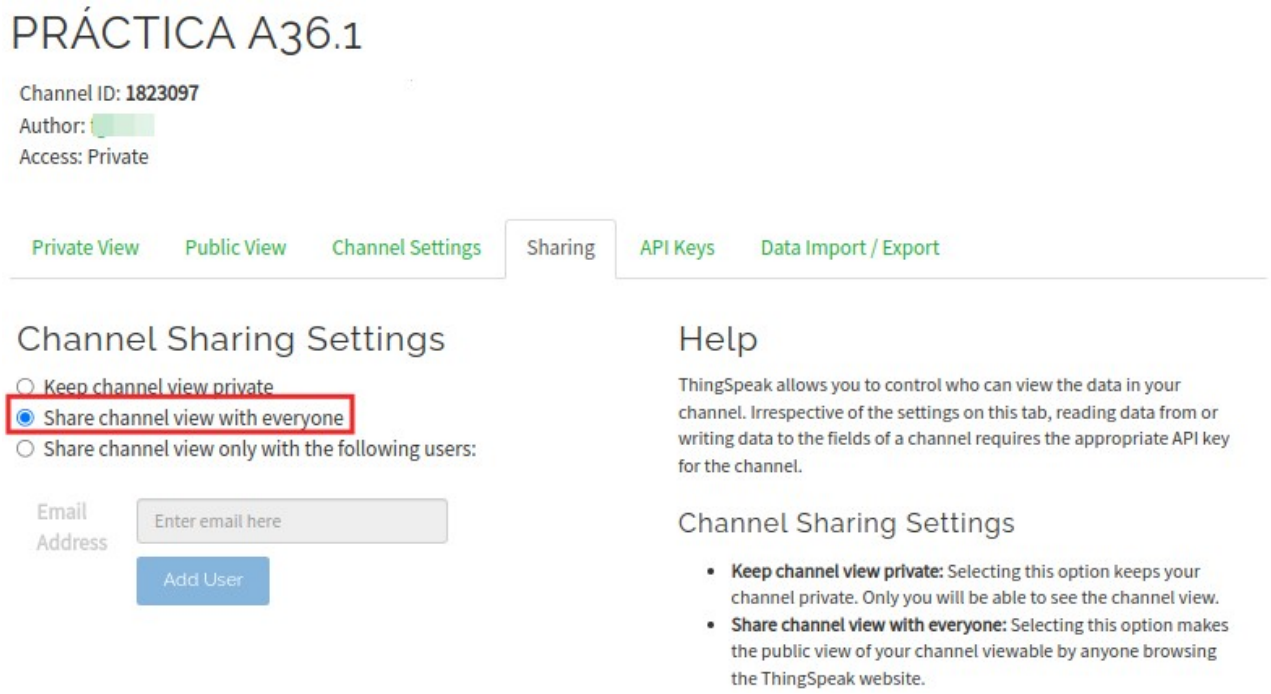
- Crear el programa en ArduinoBlocks y publicar en Thingspeak la temperatura y la humedad medidas por un sensor DHT22. Necesitaremos crearnos una cuenta si no disponemos de ella, crear el canal y crear el dispositivo MQTT.

Primero: Thingspeak

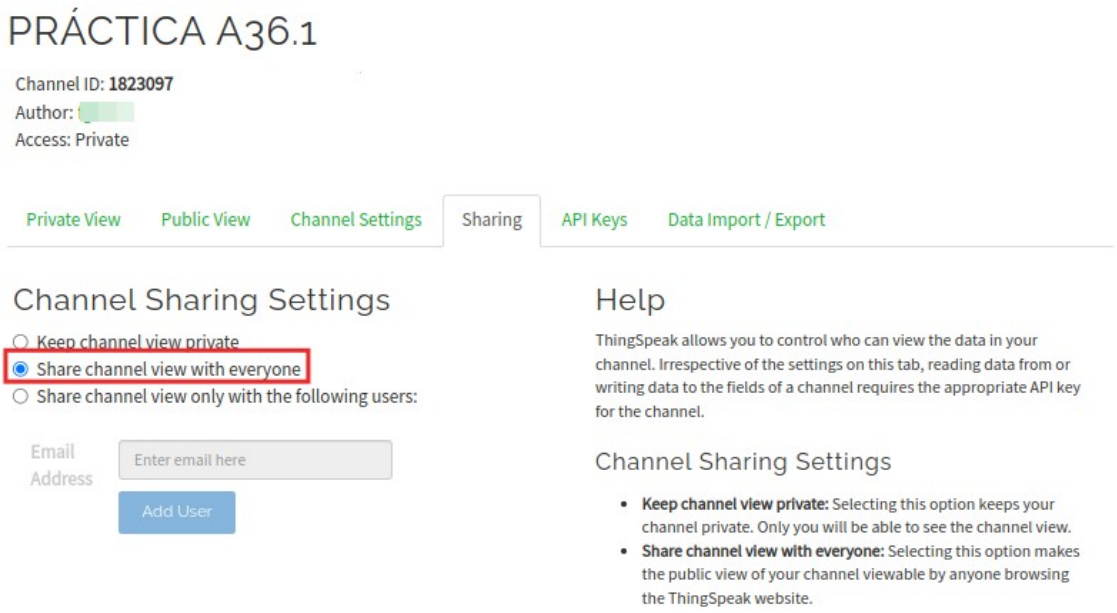
Crearemos el canal tal y como vemos en la imagen. Salvamos los cambios con el botón que hay al final.



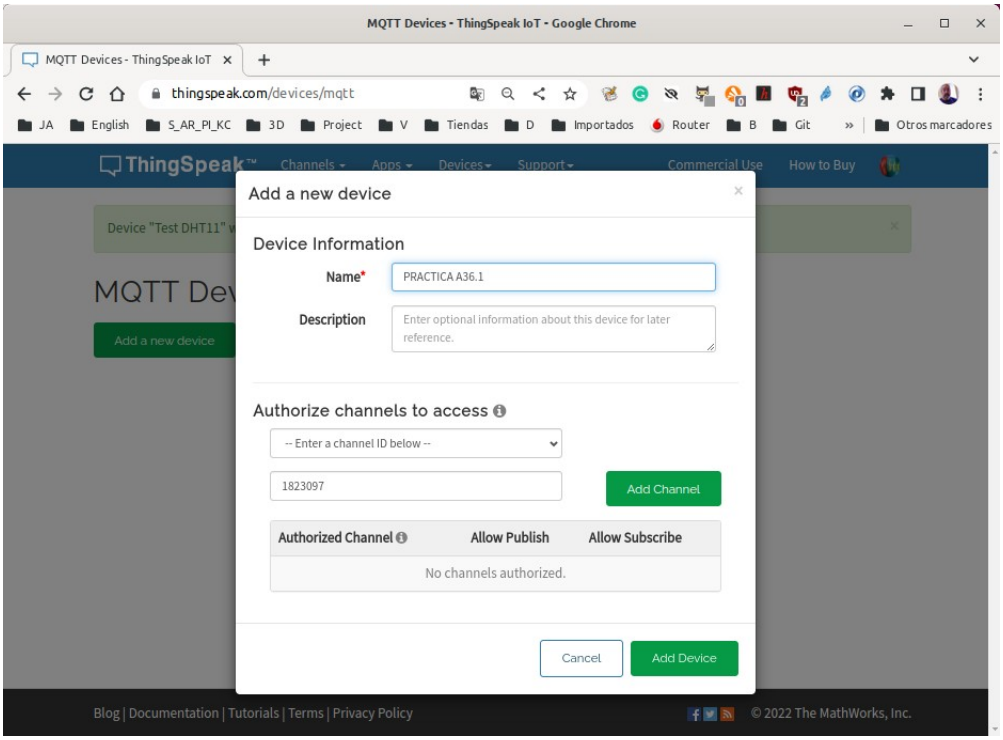
Hacemos que el canal sea público.



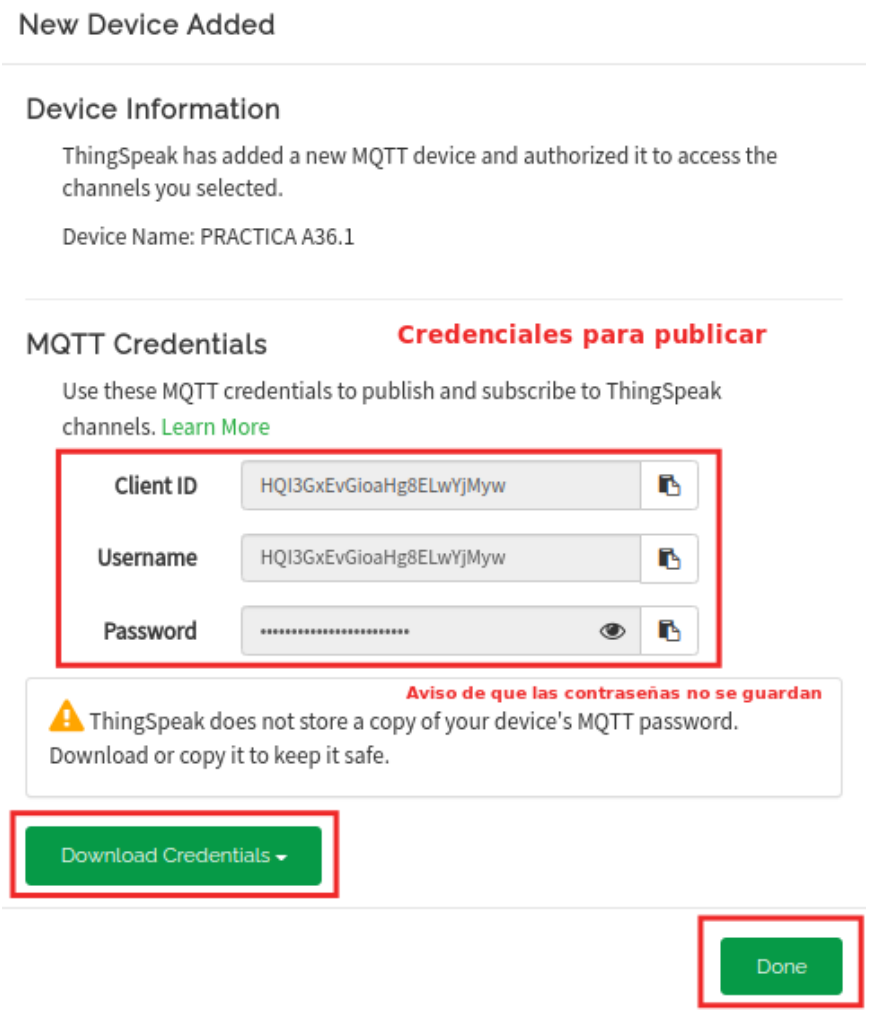
Vemos el apartado API Keys.



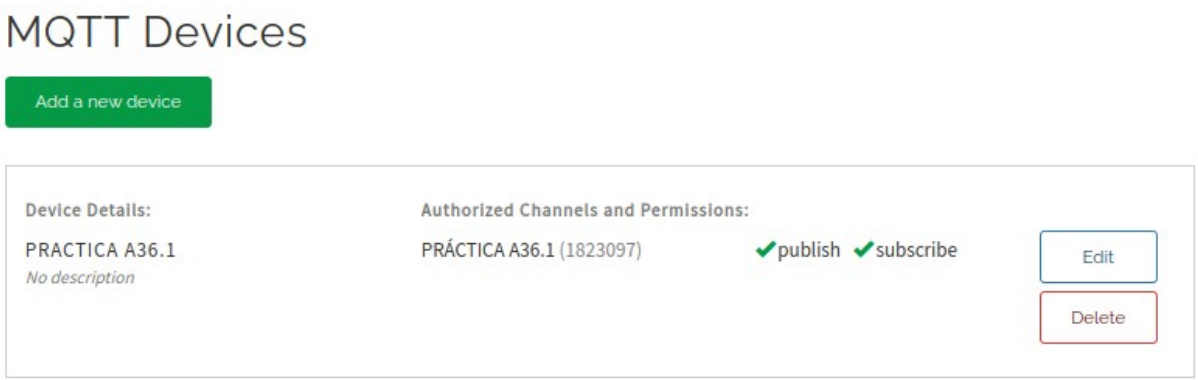
Añadimos un nuevo dispositivo MQTT:



Una vez añadido se nos muestran los datos de las credenciales para poder publicar y que debemos preservar bien copiando cada campo o bien descargando las credenciales.



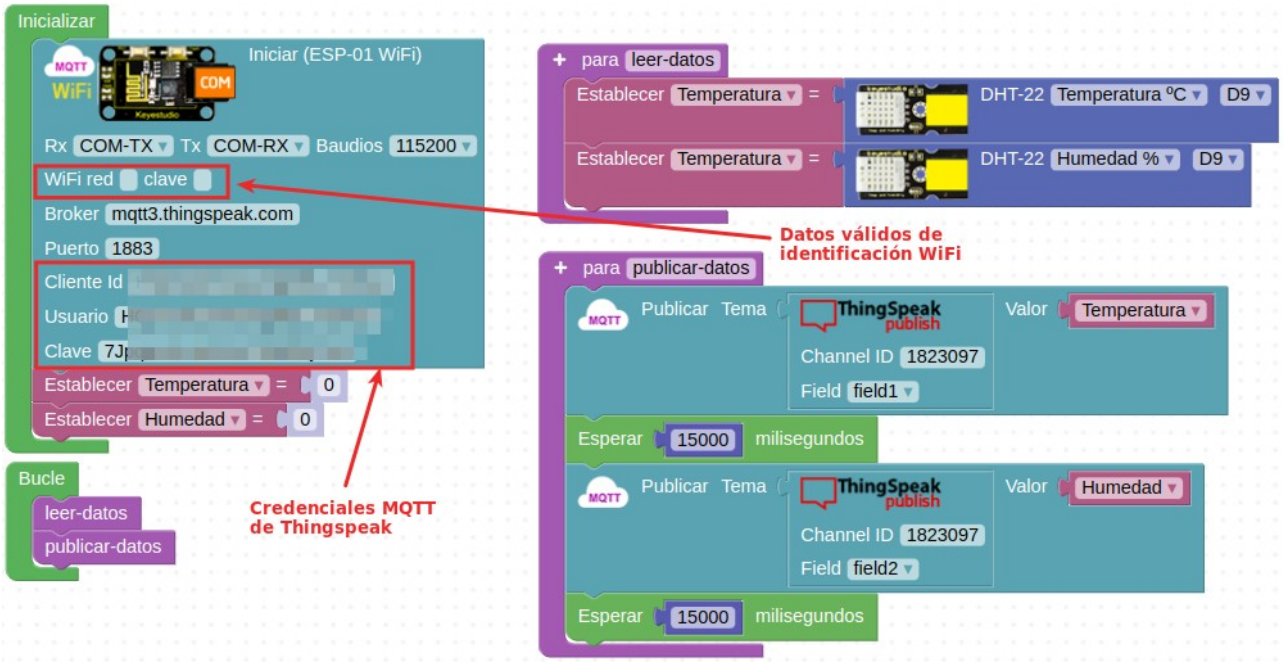
Al pulsar el botón Done se nos muestra el dispositivo MQTT creado:



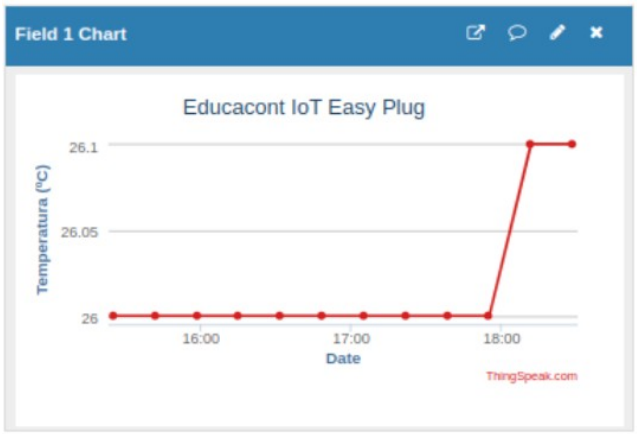
Segundo: ArduinoBlocks

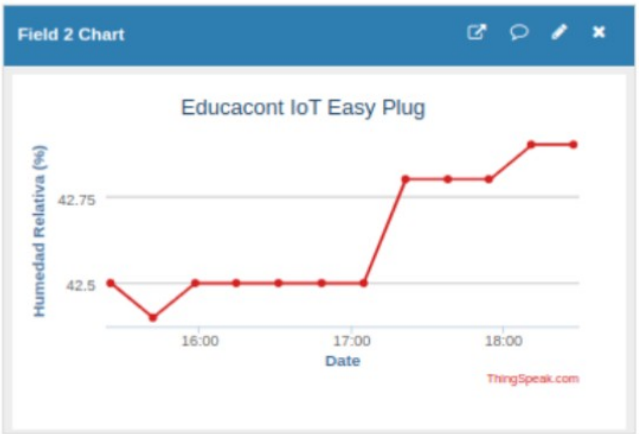
Vamos ahora a crear el programa para grabarlo en la placa y comenzar a emitir datos y subirlos al canal creado.

Un programa como el siguiente:



Nos arrojará resultados similares a los siguientes:





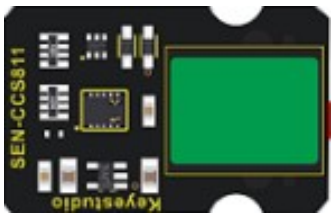
A39: CCS811 Sensor de eCO2 (Dióxido de Carbono Equivalente) y TVOC

El sensor de gas CCS811B puede detectar una amplia gama de Compuestos Orgánicos Volátiles (VOCs del inglés Volatile Organic Compound) y está diseñado para monitorizar la calidad del aire.

El sensor devuelve una lectura de VOC Totales (TVOC) y una lectura de dióxido de carbono equivalente (eCO2).

Para aprender mas sobre los conceptos de VOC y eCO2 y como utilizar el sensor se aconseja visitar la web sobre Semáforo óptico-acústico de CO2 y nivel de ruido creada por el Club Robótica Granada en su entrada Algunos conceptos sobre CO2 y VOC

Su aspecto es:



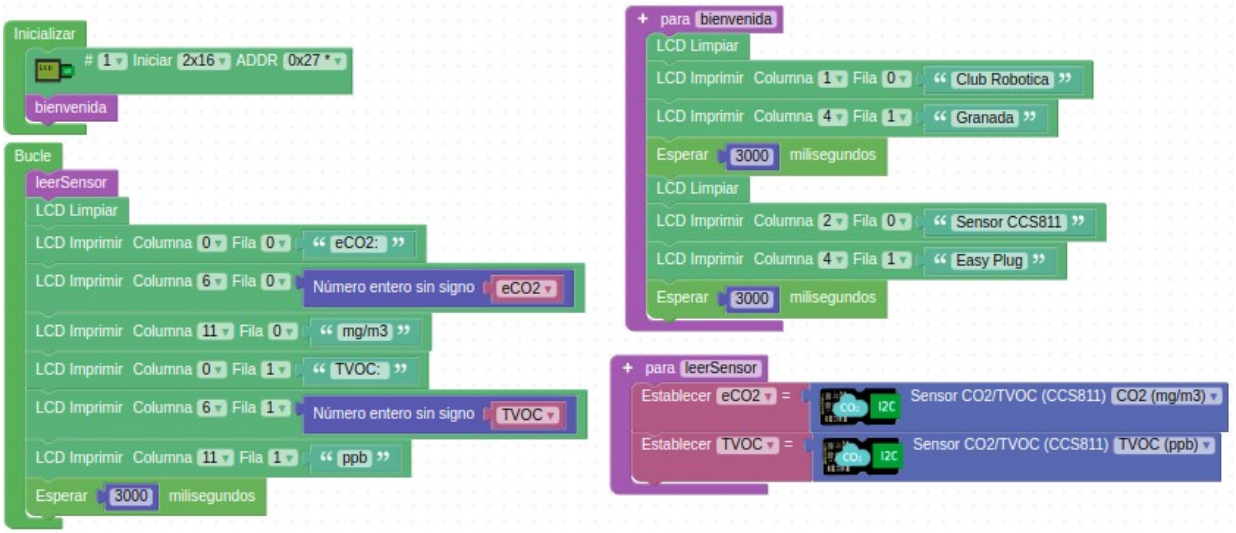
En el apartado de bloques de programación, se encuentra en "Sensores".

Se puede seleccionar entre ppm o mg/m³ como unidades para el CO2 y los VOC solamente se pueden mostrar en ppb.

PRÁCTICA A39.1:

Vamos a medir la calidad del aire de una habitación.

- Mostrar en una LCD un mensaje de bienvenida, otro relativo a lo que hace el programa y los datos medidos por el sensor CCS811.



Un programa como este arroja unos resultados en la LCD como los que vemos en las imágenes siguientes:

Mensaje de bienvenida y presentación

Mensaje relativo al sensor que se va a usar

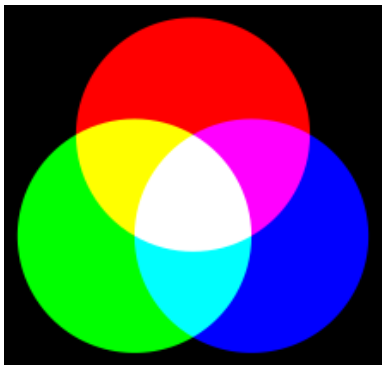
Valores iniciales medidos

Valores medidos con el sensor expuesto a un gas

Valores medidos después de dejar ventilar unos instantes

A40: Sensor de color TCS34725

El modo RGB es un estándar de colores en la industria, que se obtiene cambiando los tres colores fundamentales: Rojo (Red), Verde (Green) y Azul (Blue) y superponiéndolos entre sí. Este estándar incluye casi todos los colores que la visión humana puede percibir y es uno de los sistemas de colores más utilizados. En la imagen vemos como se comporta el modo aditivo de estos colores.



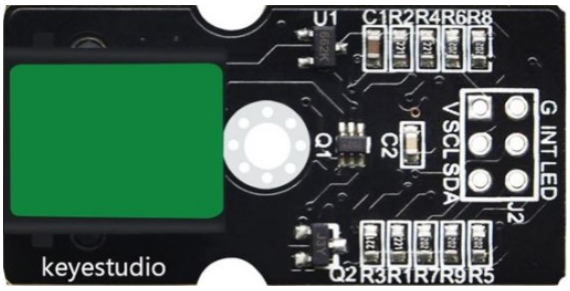
El sensor de color TCS34725 es un sensor de reconocimiento de colores RGB que puede reconocer el color de la superficie de un objeto a través de la detección óptica.

Para evitar la interferencia del entorno y aumentar la precisión, lleva incorporada una placa de protección que es un filtro de luz infrarroja, de modo que su influencia en la medida se minimiza haciendo que el reconocimiento del color sea más preciso.

El sensor también incorpora 4 LED de color amarillo rodeando al mismo, que garantizan que el sensor pueda usarse en condiciones de luz ambiental.

Trabaja a una frecuencia de reloj de 0 a 400 Khz y la distancia de detección va de 3 a 10 mm.

Su aspecto es:

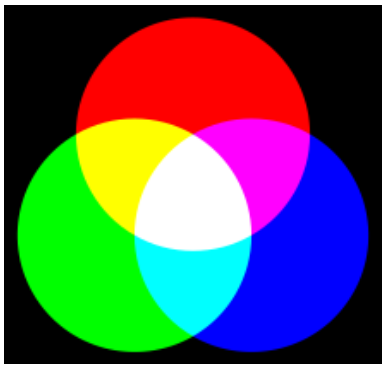


En el apartado de bloques de programación, se encuentra en "Sensores".

PRÁCTICA A40.1:

Vamos a reconocer diferentes colores.

- Mostrar en la consola el color detectado de los posibles del modo aditivo RGB de la imagen.



Los nombres de los colores son: Rojo, Verde, Azul, amarillo, Cian, Violeta y Blanco

Como demostración del bloque