

Manual EasyArduino

The screenshot displays the EasyArduino software interface. At the top, the title "EasyArduino" is on the left, and navigation buttons "Borrar", "Guardar Arduino", "Guardar Proyecto", and "Cargar Proyecto" are on the right. A "Manual de Ayuda" link is also present. A sidebar on the left lists various components: Bloques, Inicio Programa, Servo, Robot 2 Servos, Actuadores, Sensores, Logica, Control, Matematicas, Texto, Variables, and Funciones. The main workspace shows two programs. The first program starts with "Inicio Programa", followed by a "si" block containing "Pulsar Boton" (PIN# 1), a "hacer" loop with "LED" (PIN# 2) and "Accion Encender", a "Esperar" block for 1000 milliseconds, and another "LED" (PIN# 2) with "Accion Apagar". The second program starts with "Inicio Programa", followed by a "Repetir mientras" block with "Detecta Movimiento" (PIN# 5), a "hacer" loop with "Bocina" (PIN# 8) and "Accion Encender", a "Esperar" block for 1000 milliseconds, and another "Bocina" (PIN# 8) with "Accion Apagar".

Para programar Arduino fácilmente

Por Jesús Lens Costa

INDICE

Contenido

1. Introducción	3
2. Primeros pasos	3
3. Encendiendo un LED	5
4. Parpadeo de un LED	5
5. Programando el botón	6
6. Programando el sensor de luz	6
7. Programando la bocina	7
8. Programando el servo de rotación continua	8
9. Programando el servo	8
10. Programando el sensor de ultrasonidos	9
11. Programando el sensor de temperatura	10
12. Programando el sigue líneas (Sensor de infrarrojo)	10
13. Programando el sensor de movimiento	11
14. Programando el potenciómetro	12
15. Controlador de motores Dual H-Bridge	12
16. Varios programas a la vez	13

1. Introducción

EasyArduino es una aplicación web que permite la elaboración de programas para Arduino sin necesidad de escribir el código con la sintaxis de sus órdenes. Esta herramienta está basada en la tecnología de programación mediante bloques funcionales tan extendida en la actualidad.

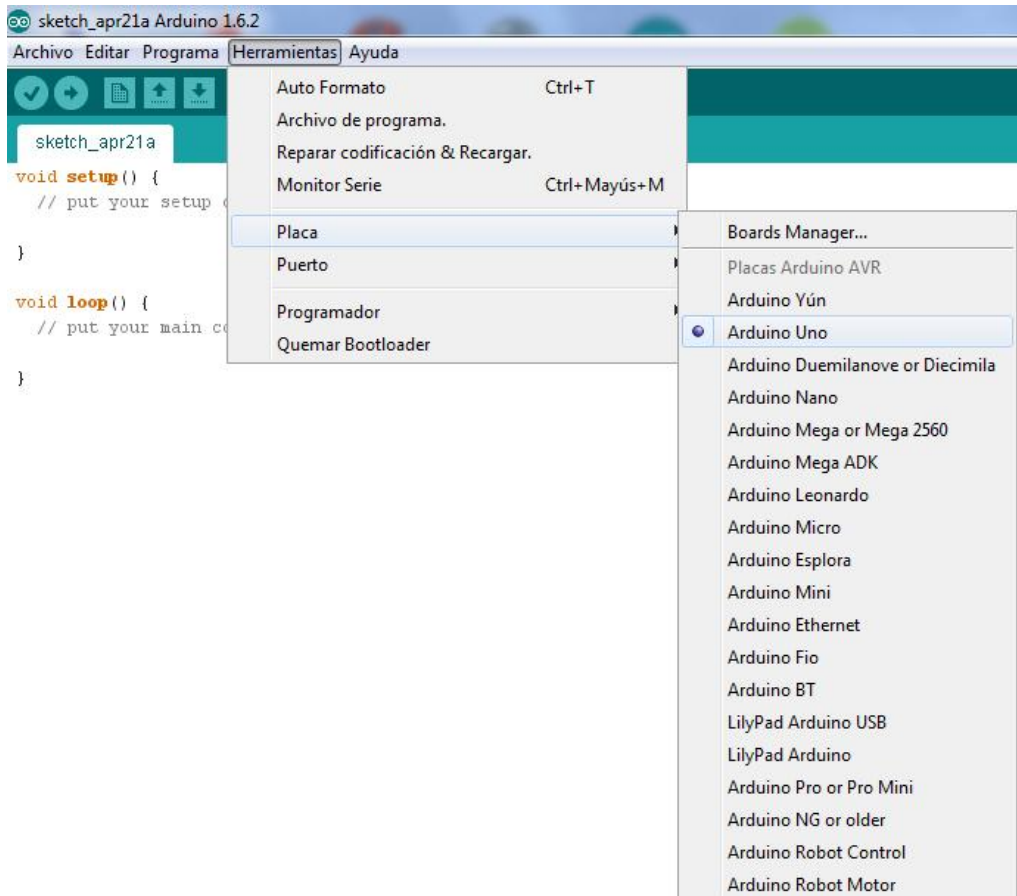
Esta posibilidad gráfica de elaborar programas es muy útil cuando estamos utilizando la Plataforma Arduino en niveles educativos. La facilidad de realizar la aplicación gráficamente permite que el alumno se dedique a pensar en el algoritmo más que en la corrección del código escrito.

Vamos a ver su funcionamiento con una colección de ejemplos.

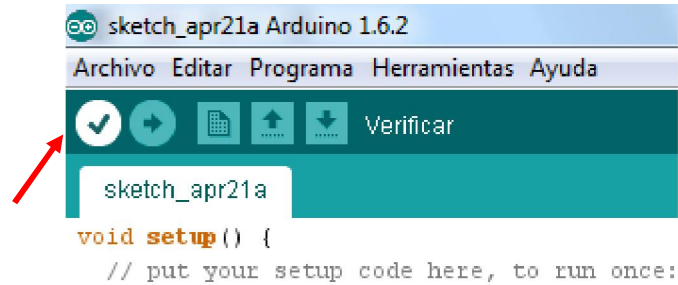
2. Primeros pasos

Para empezar a trabajar lo primero que tenemos que hacer es instalar el software necesario y para ello deberemos descargar e instalar el entorno Arduino siguiendo las instrucciones de <http://arduino.cc/en/Main/Software>

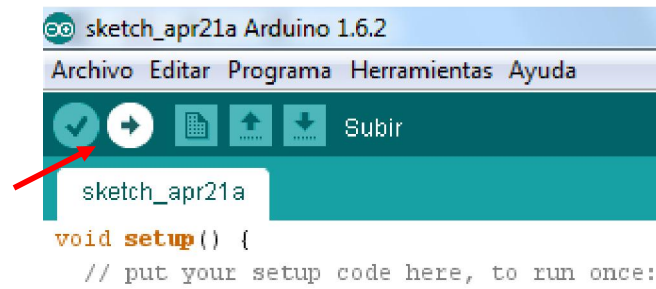
Conectamos la placa Arduino a un puerto USB de nuestro ordenador, a continuación abrimos el programa Arduino y en el menú Herramientas-Placa seleccionamos la placa que vamos a utilizar:



Ahora que tenemos todo listo podemos empezar a programar. En EasyArduino colocaremos los bloques en la secuencia deseada, si vamos a la pestaña Código veremos cómo se genera el código automáticamente. Una vez terminado le damos a GuardarArduino y nuestro programa se guardará en un archivo en la carpeta que hayamos seleccionado. Abrimos dicho archivo en el entorno Arduino y le damos a verificar:



Esto comprobará que no hay errores en la secuencia de código, una vez hecho esto ya podemos subir nuestro programa a la placa, para ello solo tendremos que pulsar el icono tal y como se puede ver en la siguiente figura:



3. Encendiendo un LED



1-Conectamos el LED en el PIN digital 1 de nuestra placa (puede ser cualquier otro).

2-Seleccionamos el bloque LED, le asignamos el pin digital 1 y escogemos la acción que deseamos que haga, en este caso: *Encender*.

4. Parpadeo de un LED

Este programa hace parpadear un LED, estará encendido durante 1 segundo, luego se apagará otro segundo y se volverá a encender, así sucesivamente.



1-Conectamos el LED en el PIN digital 1 de nuestra placa (puede ser cualquier otro).

2-Seleccionamos dos bloques LED, les asignamos el pin digital 1 y escogemos la acción que deseamos que hagan.

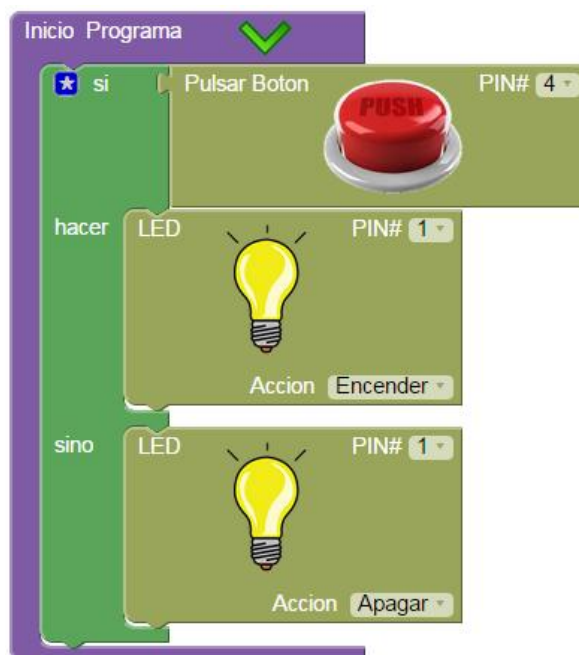
3-Colocamos el bloque Esperar tal y como muestra la figura y le asignamos un tiempo de espera, este será el tiempo que el LED esté encendido o apagado.

5. Programando el botón

El botón es un dispositivo digital de entrada que se activa cuando se pulsa con el dedo, permitiendo que circule por él la corriente, y se desactiva cuando se deja de pulsar. Puede devolver dos estados: 1 y 0. Es decir: pulsado y no pulsado.

Ejemplo: Encender un LED al pulsar el botón y apagarlo al soltarlo.

Al ejecutar este programa el LED comienza apagado. Cuando se pulsa el pulsador el LED se enciende y al soltarlo se vuelve a apagar.



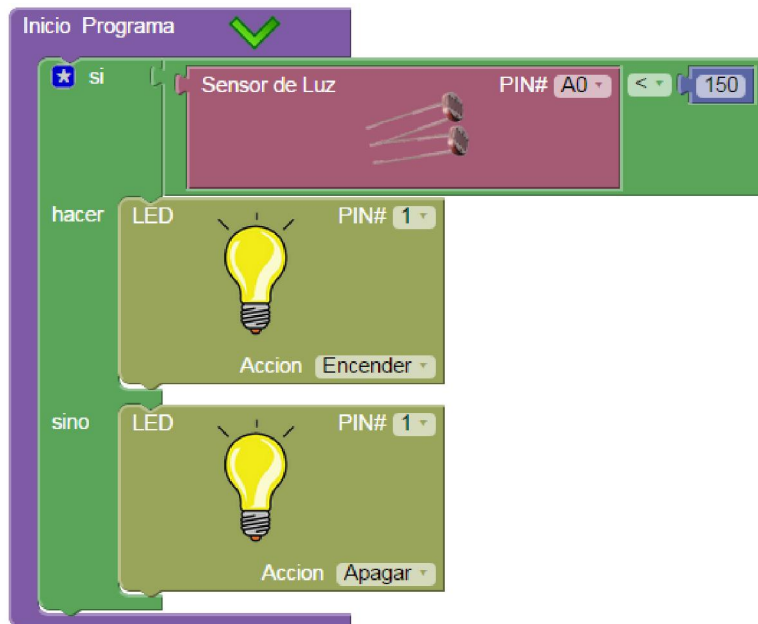
Nota: Para añadirle más opciones al bloque "Si - hacer" debemos pulsar sobre la crúz del recuadro azul e incorporar la nueva opción al bloque.

6. Programando el sensor de luz

El sensor de luz es un sensor que tal y como su nombre indica, detecta luz. Responde a los cambios en la intensidad de la luz y modifica su resistencia cuando ésta varía. El sensor devuelve valores comprendidos entre 0 y 500, aproximadamente; este valor dependerá del tipo de sensor. Se trata de un componente analógico, ya que devuelve un rango de valores, por lo tanto, deberá conectarse a uno de los pines analógicos (A0, A1,...). Valores cercanos a 0 indican oscuridad y cercanos a 500 indican luz.

Ejemplo: Encender un LED cuando el sensor de luz detecte poca luz

Primero tenemos que comprobar que hay poca luz. En ese caso debemos encender el LED que está conectado en el PIN digital 1. En caso de que la luz sea mayor de 150 el LED se apagará.

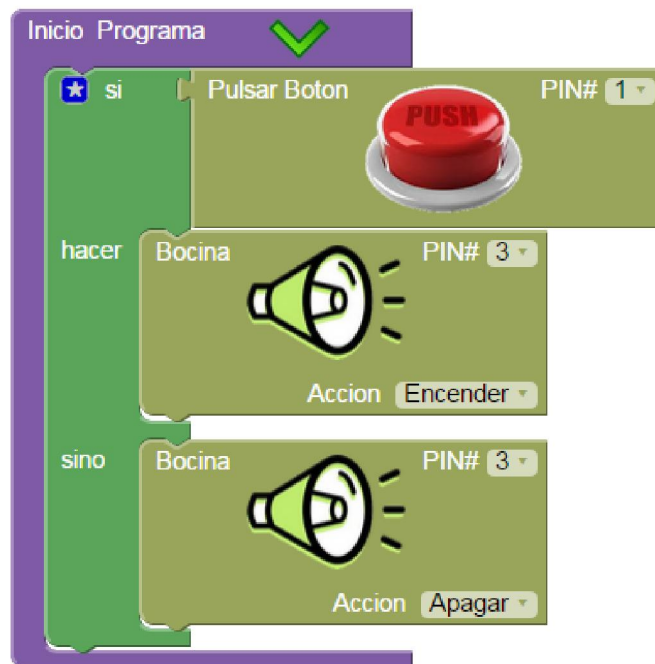


7. Programando la bocina

La bocina o zumbador es un componente de salida que emite zumbidos.

Ejemplo: Hacer sonar el zumbador al presionar el botón

Al ejecutar este programa la bocina no suena, está apagada. Cuando se pulsa el pulsador la bocina se enciende y empieza a sonar; al soltarlo se vuelve a apagar.

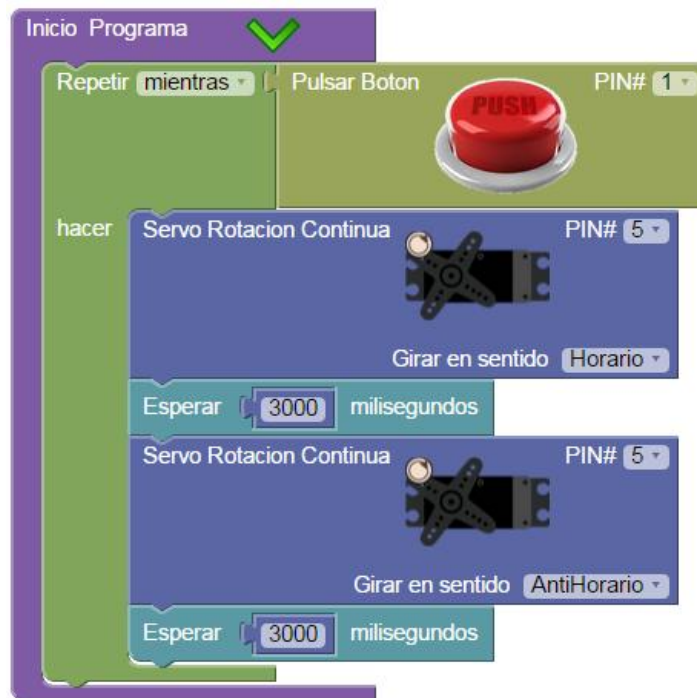


8. Programando el servo de rotación continua

Un servomotor de rotación continua es un servo cuyo funcionamiento se parece al de un motor convencional. Este servo no se detiene en una posición sino que puede girar en un sentido continuamente.

Ejemplo: Programar un servo de rotación continua para que gire en sentido horario durante 3 segundos y luego gire en sentido antihorario otros 3 segundos.

En este ejemplo usaremos el bucle de control “Repetir – mientras”, el bucle se estará ejecutando mientras pulsemos el botón.



9. Programando el servo

Un servo es un motor que es capaz de girar entre 0 y 180 grados.

Ejemplo: Programar un servo para moverlo a un ángulo de 60 grados y después a un ángulo de 120 grados.

El servo se colocará en la posición de 60 grados, esperará un segundo y pasará a la posición de 120 grados durante otro segundo. Repetirá esta secuencia una y otra vez.

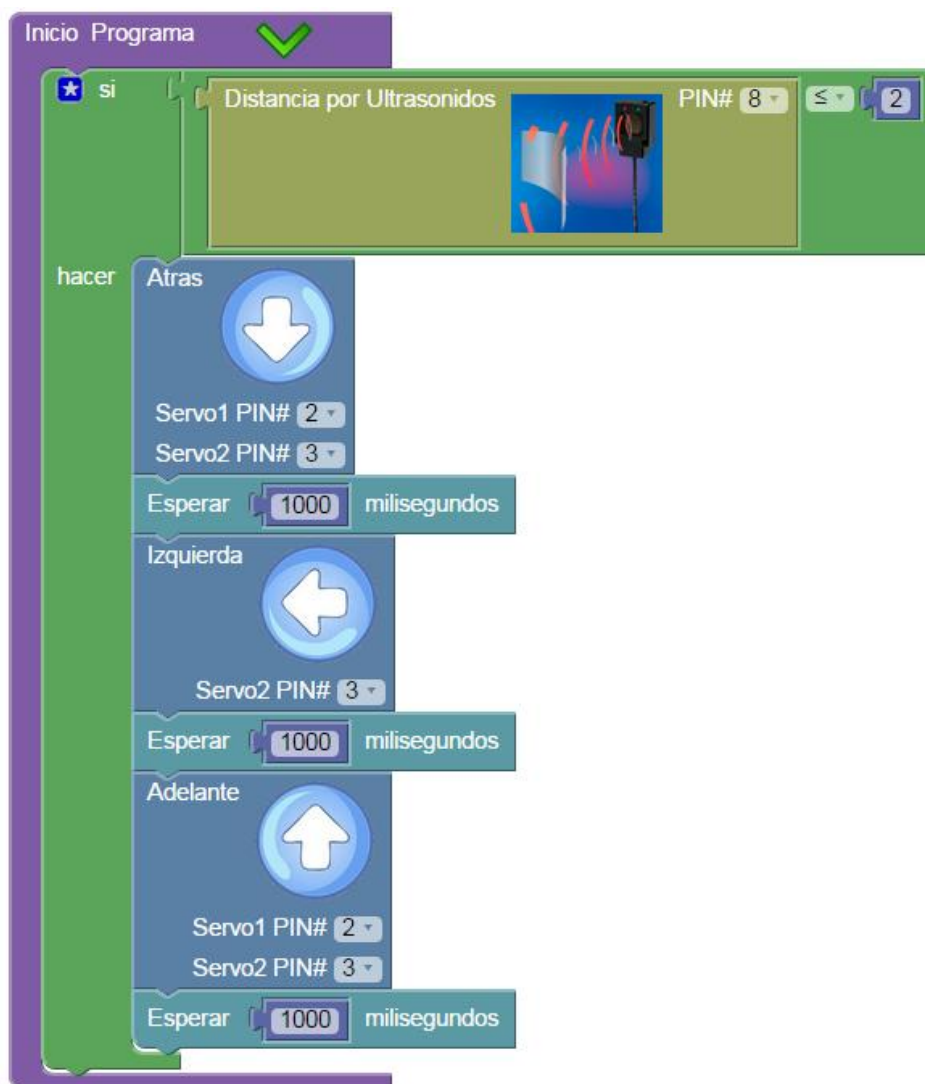


10. Programando el sensor de ultrasonidos

Los sensores de ultrasonidos son dispositivos que utilizan ondas sonoras de ultrasonidos para saber a qué distancia está un objeto. Su funcionamiento es así: se emite una onda de ultrasonido, esa onda se propaga, y al encontrarse con un obstáculo rebota. Cuanto más tiempo tarde en rebotar más lejos estará el objeto.

Ejemplo: Programar un robot con dos servos para que cuando esté a punto de chocar con un objeto paré, dé marcha atrás, gire a la izquierda y continúe su camino.

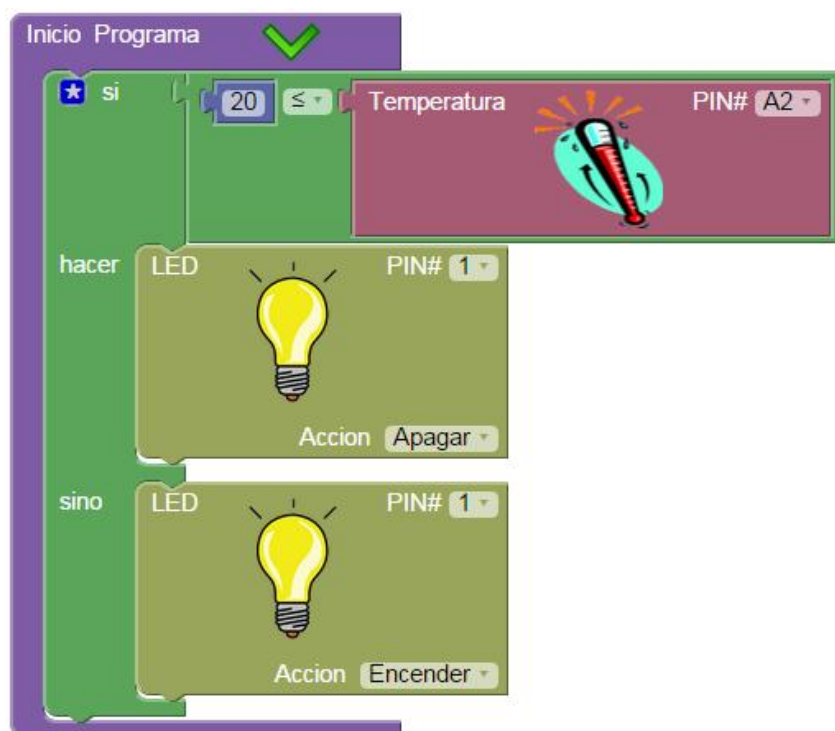
Si la distancia medida por el sensor de ultrasonidos es menos o igual que 2 cm, el robot dará marcha atrás y luego girará a la izquierda. Si la distancia es mayor de 2 cm el robot se moverá hacia adelante. Como nuestro programa se ejecuta una y otra vez, el robot seguirá siempre hacia adelante a menos que la distancia al objeto sea menor o igual que 2 cm.



11. Programando el sensor de temperatura

El sensor de temperatura es un sensor que mide la temperatura en grados centígrados. Se trata de un componente analógico, ya que devuelve un rango de valores, por lo tanto, deberá conectarse a uno de los pines analógicos (A0, A1,...).

Ejemplo: Encender un LED cuando el sensor de temperatura indique que hay menos de 20 grados y apagarlo cuando hay 20 o más.

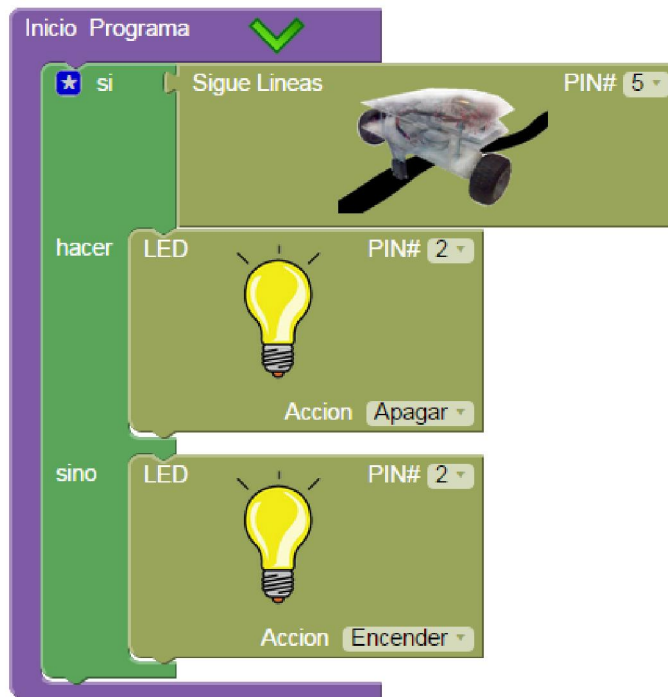


12. Programando el sigue líneas (Sensor de infrarrojo)

El sensor infrarrojo (IR) es un dispositivo que detecta la luz reflejada, por lo que es capaz de diferenciar entre blanco y negro. Es un componente digital que solo devuelve dos valores. Cuando detecta blanco devuelve 1 y cuando detecta negro devuelve 0.

Ejemplo: Encender un LED cuando el sensor detecta negro.

Si el sensor devuelve 0 no será línea negra, por lo tanto se encenderá el LED; en caso contrario se apagará.

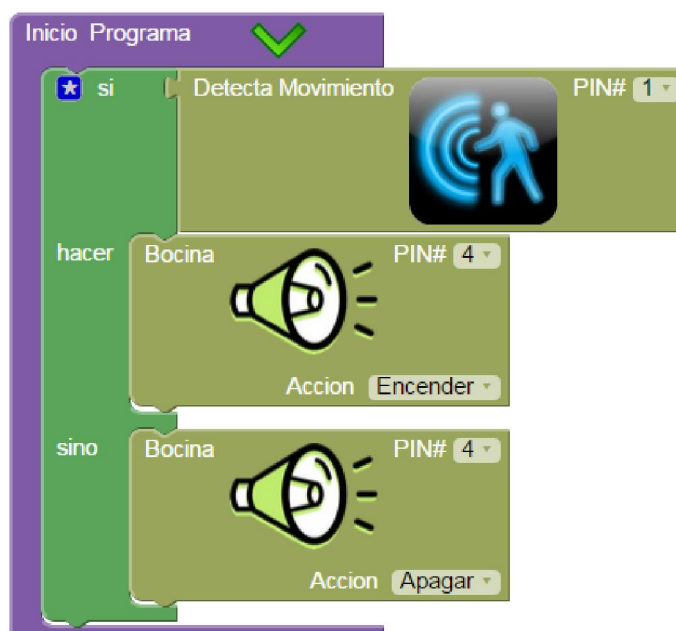


13. Programando el sensor de movimiento

El sensor de movimiento es un dispositivo que detecta si algo se mueve. Es componente digital que solo devuelve dos valores. Cuando detecta movimiento devuelve 1 y cuando no, devuelve 0.

Ejemplo: Encender la bocina cuando el sensor detecta movimiento.

Si el sensor detecta que algo se mueve devolverá 1, y por lo tanto sonará la bocina, en caso contrario se apagará.

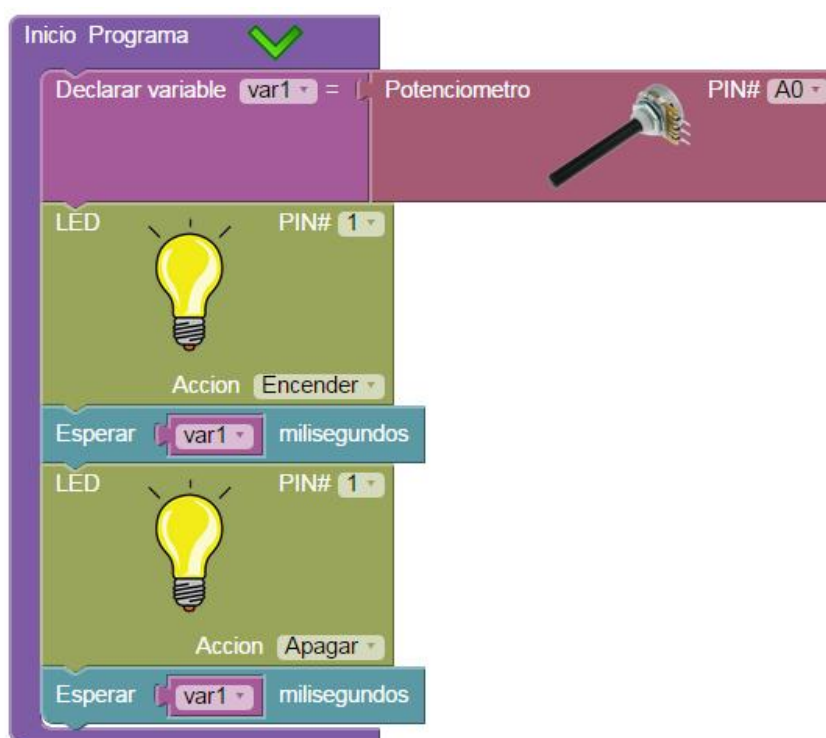


14. Programando el potenciómetro

Un potenciómetro es un componente electrónico que permite cambiar la intensidad de la corriente que circula por un circuito.

Ejemplo: Variar el tiempo de parpadeo de un LED.

En esta ocasión vamos a declarar una variable para guardar el valor que devuelva el potenciómetro. Este valor será el tiempo que usemos para el parpadeo del LED. Como se puede ver en la figura, el LED se enciende durante el tiempo indicado según el valor del potenciómetro, y se apaga durante el mismo tiempo. Si nosotros giramos el potenciómetro haremos que cambie ese tiempo y, por lo tanto, variará la velocidad de parpadeo.



15. Controlador de motores Dual H-Bridge

El controlador de motores es una placa que nos permitirá mover uno o dos motores hacia delante o hacia atrás, y controlar su velocidad a través de los pines de Arduino.

Los pines para controlar la velocidad y la dirección son los marcados como EA, I1, I2, EB, I3, I4. Los tres primeros sirven para controlar el motor A, y el resto el B. En el caso del Motor A: I1 e I2 sirven para controlar la dirección. Si I1 está en ALTA e I2 en BAJA, el motor avanza hacia delante. Si están al revés el motor avanza hacia atrás. EA sirve para controlar la velocidad del motor con lo que podemos avanzar a más o menos velocidad. Los pines del Motor B funcionan de la misma forma.

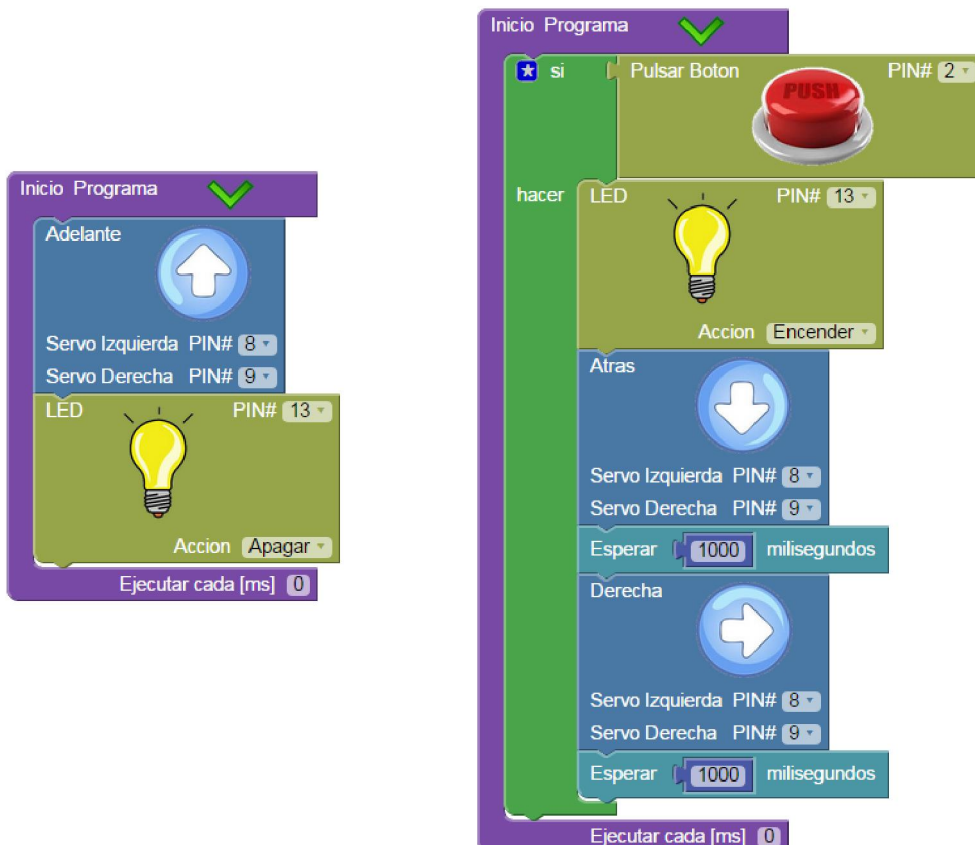
Ejemplo: Conectar dos motores al controlador y aplicarle movimientos.



16. Varios programas a la vez

Para crear varios programas que se ejecuten a la vez solo tenemos que utilizar diferentes bloques de "Inicio Programa".

Ejemplo: Mover el robot; si se pulsa el botón encender el led y aplicar otros movimientos.



Los dos programas correrán a la vez, de modo que aunque se esté moviendo el robot debido al primer programa, si pulsamos pulsar el botón se encenderá el LED y el robot se desplazará hacia atrás y después girará a la derecha. En cuanto soltemos el botón el robot volverá a ir hacia adelante.

Usando varios bloques de inicio programa podremos crear programas orientados a eventos. Como hemos visto en este ejemplo, el primer programa estará corriendo continuamente, y el segundo estará a la espera de que suceda un evento, en este caso, que se pulse un botón.