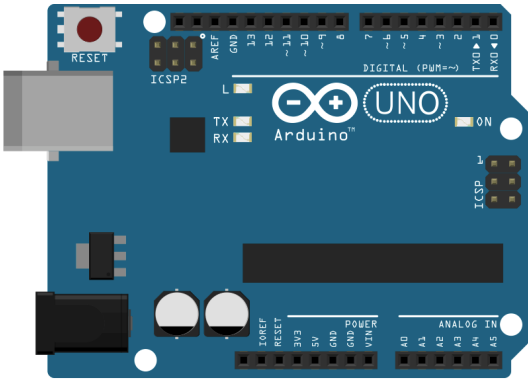
LIBRO DE PRÁCTICAS



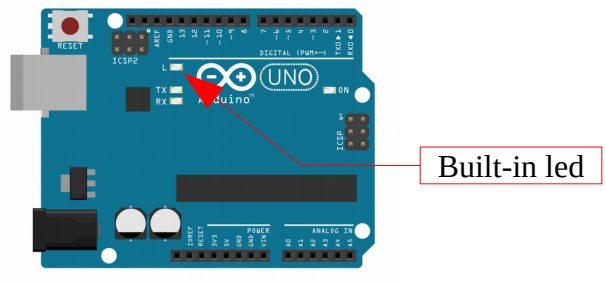
Juanjo López

Índice de prácticas

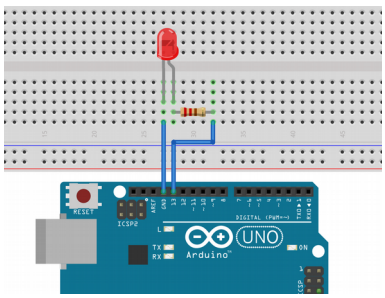
Built-In Led

Arduino incorpora un led en la placa conectado al pin 13 a través de una resistencia. De esta forma podemos hacer un primer test del funcionamiento de la placa sin tener que conectar ningún componente *adicional* a la placa Arduino.

Built-in led



Circuito equivalente al *built-in* led

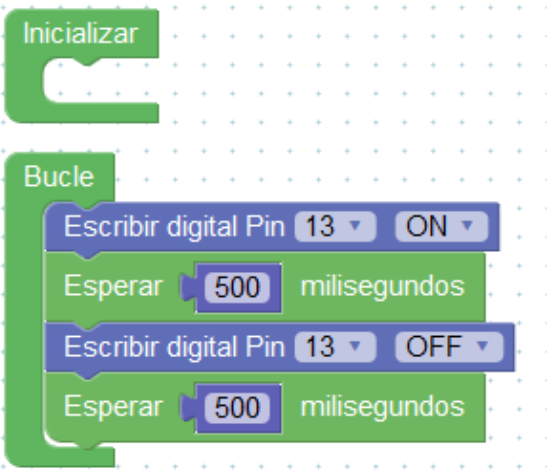


Built-In Led - 1

Parpadeo "built-in" led

CÓDIGO DE PROYECTO:

El pin 13 funcionará como salida. El programa hace parpadear el led integrado en la placa Arduino, de forma que permanece un tiempo encendido (ON) y otro tiempo apagado (OFF). El tiempo de retardo determinará la velocidad de parpadeo, se pueden probar distintos valores para ver la diferencia de velocidad en el parpadeo. Este proceso se repite infinitamente dentro del "bucle" principal de forma que el led nunca deja de parpadear.



Parpadeo del led 13 con retardo de 500 ms

Modifica el tiempo de retardo para:
100ms, 250ms, 2000ms, 5000ms

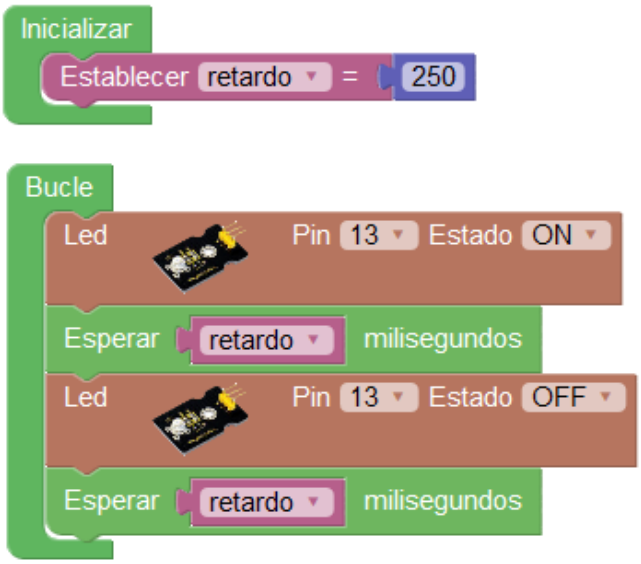
Built-In Led - 2

Parpadeo "built-in" led

CÓDIGO DE PROYECTO:

El bloque "Led" de los actuadores funciona exactamente igual que el bloque "Escribir Digital". Podemos usarlo de la misma forma.

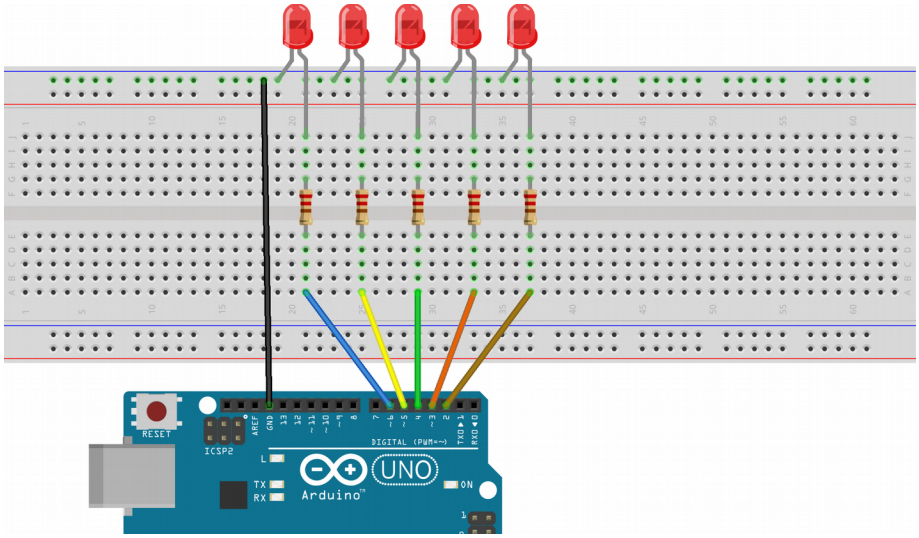
El siguiente programa realizar el parpadeo del led conectado en el pin 13 usando una variable para fijar el tiempo de retardo y utilizando el bloque "Led"



Modifica el valor de la variable en el bloque "Inicializar", prueba con estos valores 100ms, 250ms, 2000ms, 5000ms

Secuencias de leds

Conectaremos 5 leds a la placa Arduino en los pines 2, 3, 4, 5, 6 conectando una resistencia de 220 Ω antes del led ánodo del led (patilla larga) y todos los cátodos de los leds (patilla corta) se unirán directamente a GND:

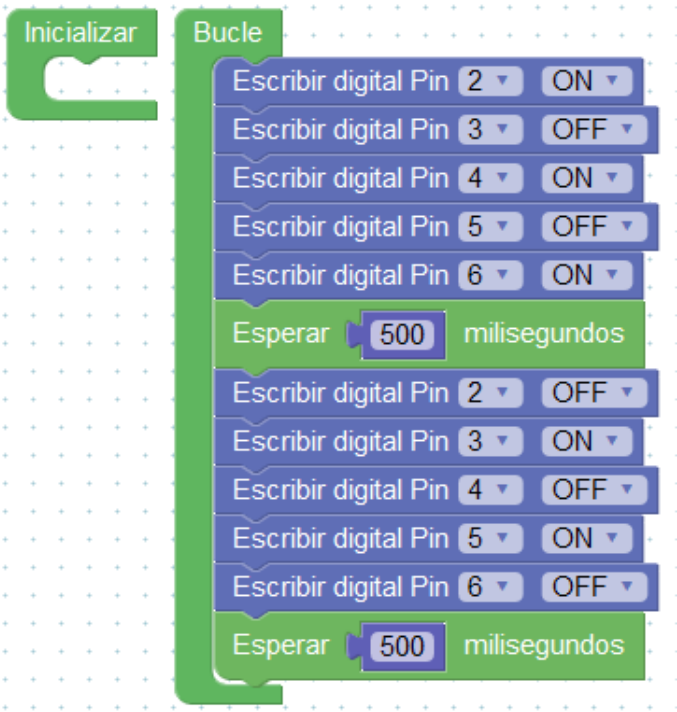


Secuencias de leds - 1

Pares / impares

CÓDIGO DE PROYECTO:

Encender leds pares y luego impares.

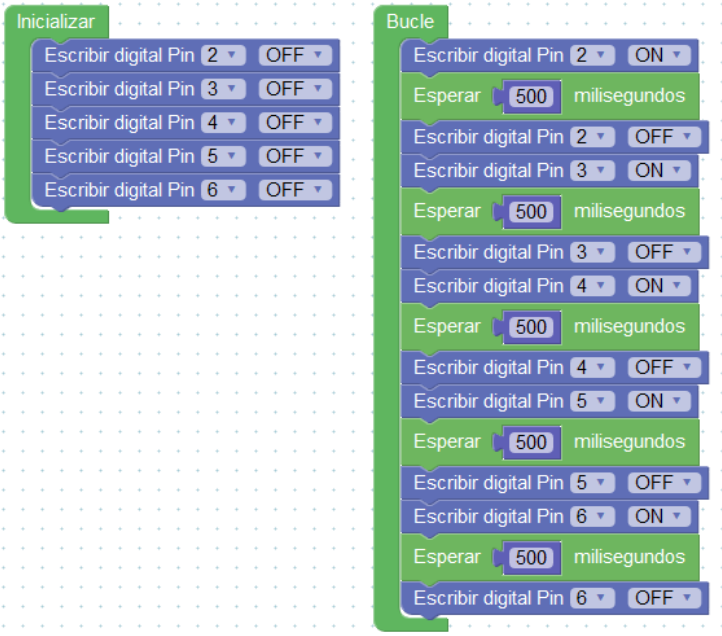


Secuencias de leds - 2

Deslizar led

CÓDIGO DE PROYECTO:

Encender un led consecutivamente uno detrás de otro de izquierda a derecha (sólo un led encendido). Al llegar al último vuelve a empezar la secuencia.



Secuencias de leds - 3

Encendido / apagado progresivo

CÓDIGO DE PROYECTO:

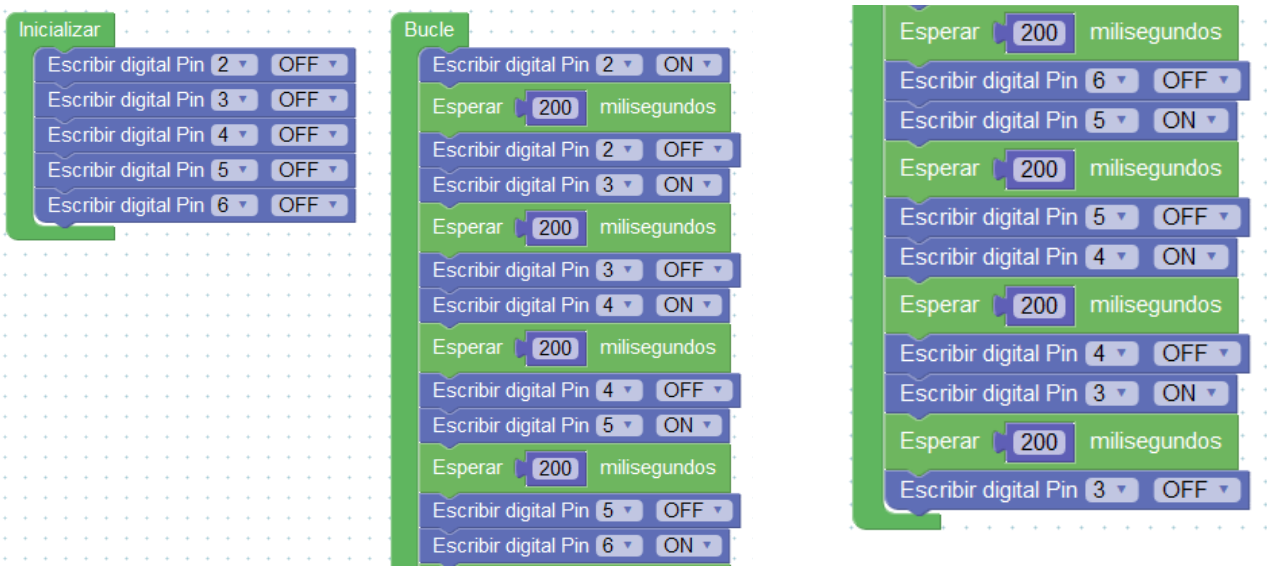
Encender los leds consecutivamente uno detrás de otro de izquierda a derecha hasta encederse todos, después se van apagando consecutivamente en el orden inverso que se han encendido hasta que se queden todos apagados y vuelve a empezar.

Secuencias de leds - 4

Coche fantástico

CÓDIGO DE PROYECTO:

Se ilumina un único led de lado a lado.

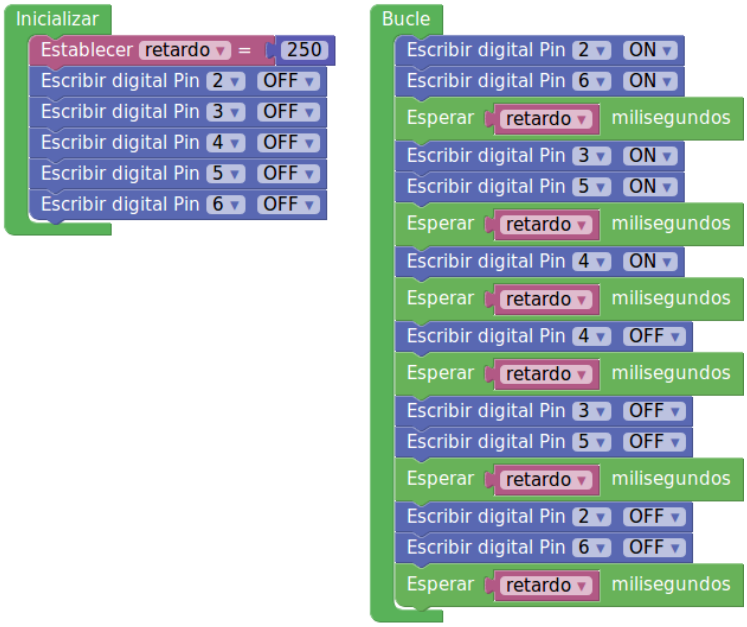


Secuencias de leds - 5

Dentro-Fuera

CÓDIGO DE PROYECTO:

Realiza una secuencia de leds que los leds se enciendan de fuera hacia dentro y luego al revés, la velocidad la podremos ajustar simplemente cambiando el valor de una variable.



Modifica el tiempo de retardo para:
100ms, 250ms, 2000ms, 5000ms

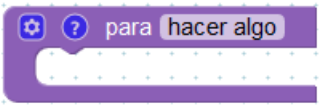
Secuencias de leds - 6

Repetición de secuencias 1+2+3

CÓDIGO DE PROYECTO:

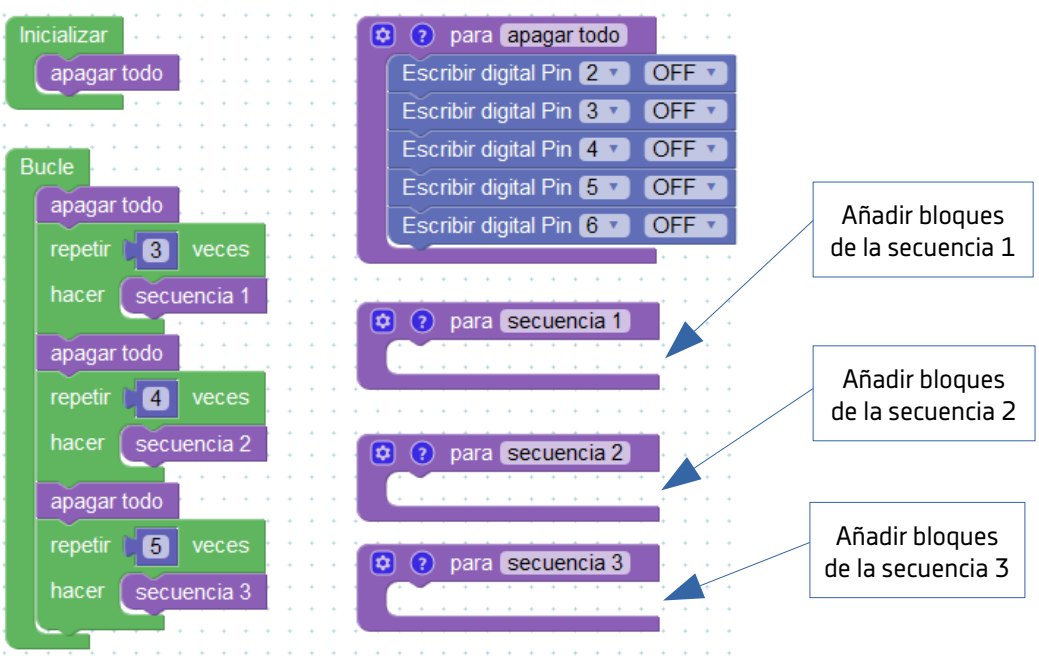
Las funciones nos permite agrupar bloques bajo un nombre y poder ejecutar todo el bloque simplemente añadiendo el bloque de llamada de la función.

Definición de la función

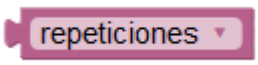


Llamada de la función

Crea una función “apagar todo” para iniciar apagar todos los leds, otra función para la “secuencia 1”, otra para la “secuencia 2” y otra para la “secuencia 3”. En el bucle principal ejecuta 3 veces la secuencia 1, 4 veces la secuencia 2 y 5 veces la secuencia 3 de forma continua.



Cambia el número de repeticiones de cada secuencia, y si te atreves...
Haz que cada secuencia se repita tantas veces como el valor de una variable

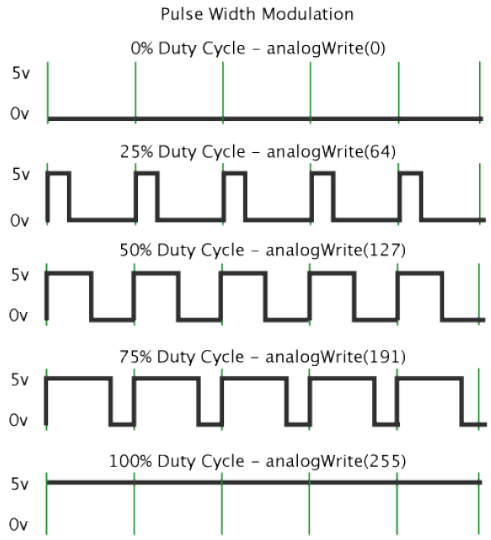


Salidas PWM

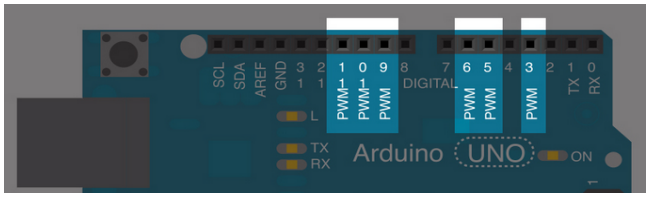
La modulación por ancho de pulsos (también conocida como PWM) de una señal o fuente de energía es una técnica en la que se modifica el ciclo de trabajo de una señal periódica (cíclica), para controlar la cantidad de energía que se envía a una carga. Con esta técnica podemos controlar la intensidad de leds, la velocidad de motores, etc.

Es la forma en la que Arduino genera una señal pseudo-analógica en sus salidas a partir de pulsos digitales para variar la energía que envía a través del pin correspondiente.

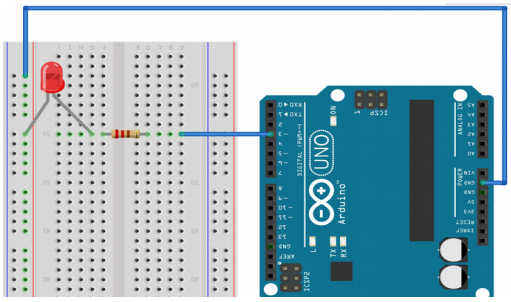
Gráfico de diferentes pulsos PWM (de menos a más intensidad)



Los pines que permiten el funcionamiento como PWM están señalados con el símbolo ~



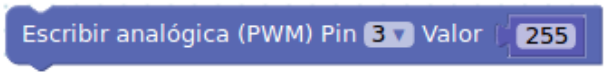
Conectamos un led al pin 3 de la siguiente forma:



Los valores que podemos escribir en el pin mediante PWM son de 0 a 255.

Salida PWM a su valor mínimo (0%)

Salida PWM a su valor máximo (100%)



Salidas PWM -1

Regular la intensidad de led

CÓDIGO DE PROYECTO:

Encendido de un led en varios pasos de intensidad.
Calcula la tabla con los valores que se deben escribir en el pin de salida PWM

Calcula los valores para cada paso:

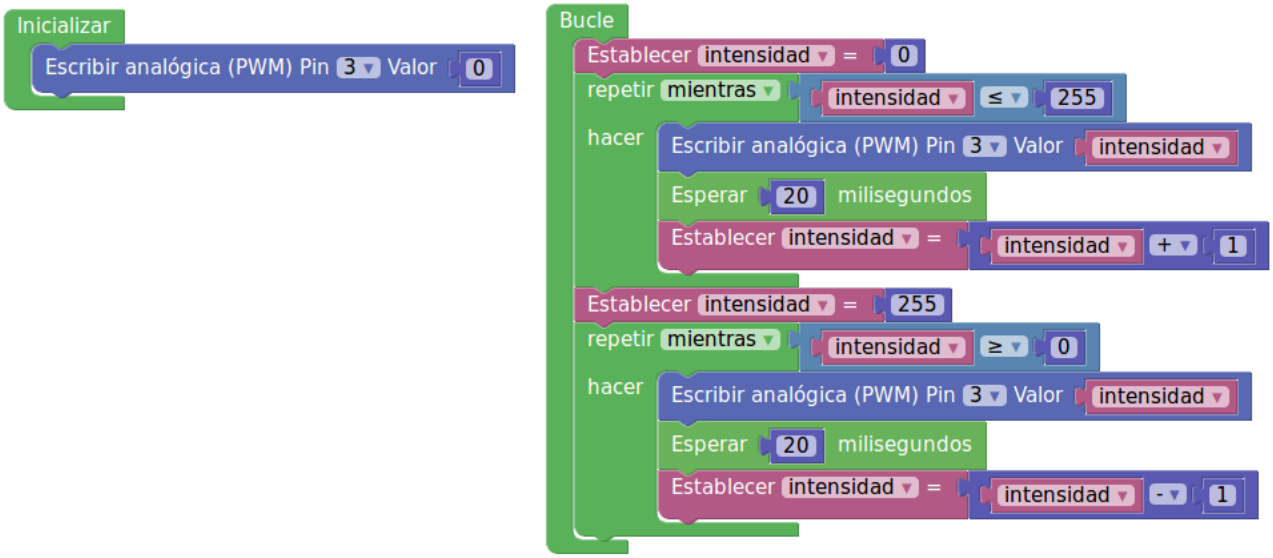
Paso	*% del Led*	**Valor PWM (AnalogWrite)**
1	*0%*	**0**
2	*20%*	**51**
3	*40%*	
4	*60%*	
5	*80%*	
6	*100%*	**255**

Salidas PWM -2

Efecto amanecer / anochecer

CÓDIGO DE PROYECTO:

Encendido y apagado suave del Led: Se escribirán los valores de 0 a 255 en el pin 3 de forma que se ilumine suavemente. Al llegar al 100% (255) se realizará el proceso al revés hasta llegar a 0.
(Modifica el valor de retardo para acelerar o ralentizar el proceso)

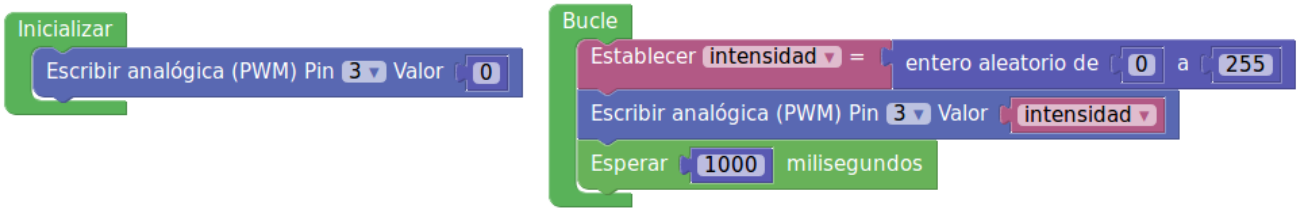


Salidas PWM -3

Intensidad del led aleatoria

CÓDIGO DE PROYECTO:

Realizar un programa que cada segundo cambie la intensidad del led de forma aleatoria.

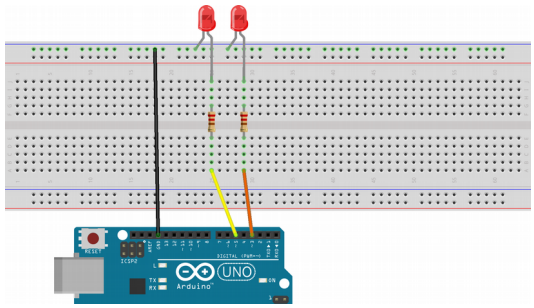


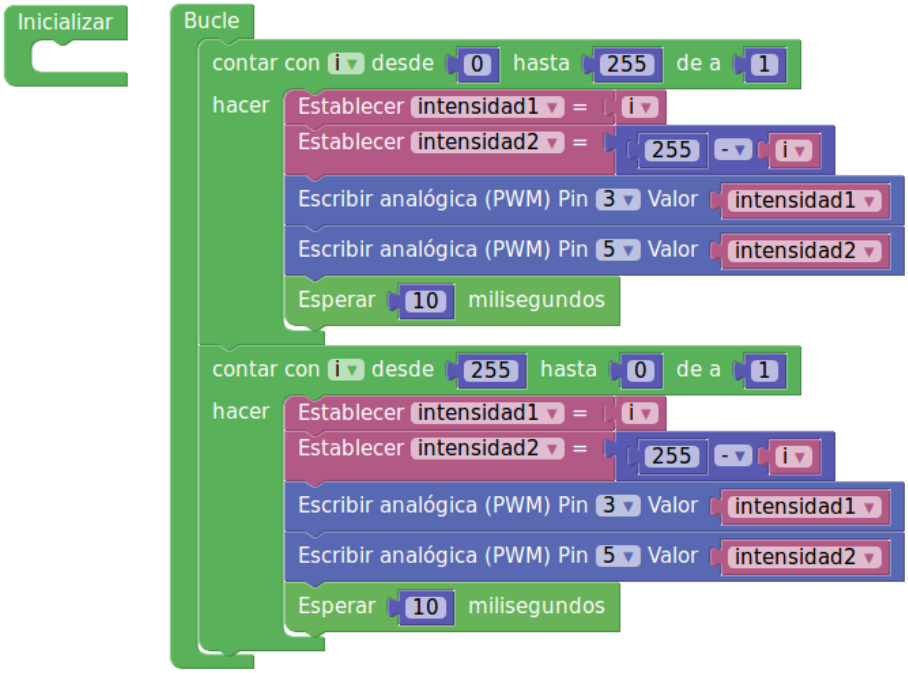
Salidas PWM -4

Controlar la intensidad de dos leds

CÓDIGO DE PROYECTO:

Realizar un programa con dos leds conectados a pines PWM, cuando la intensidad de uno sube la otra baja. Leds conectados a los pines 3 y 5.





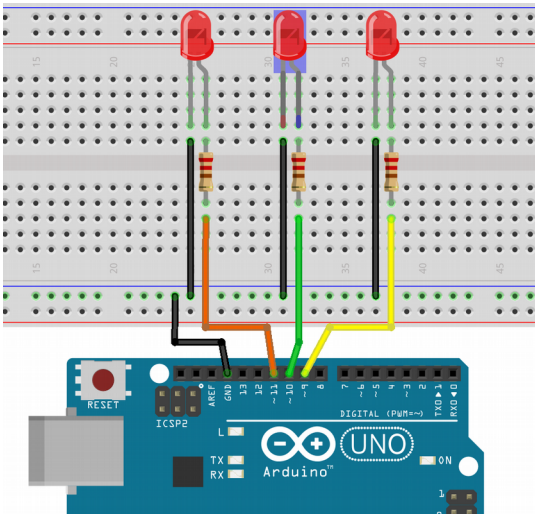
Salidas PWM - 5

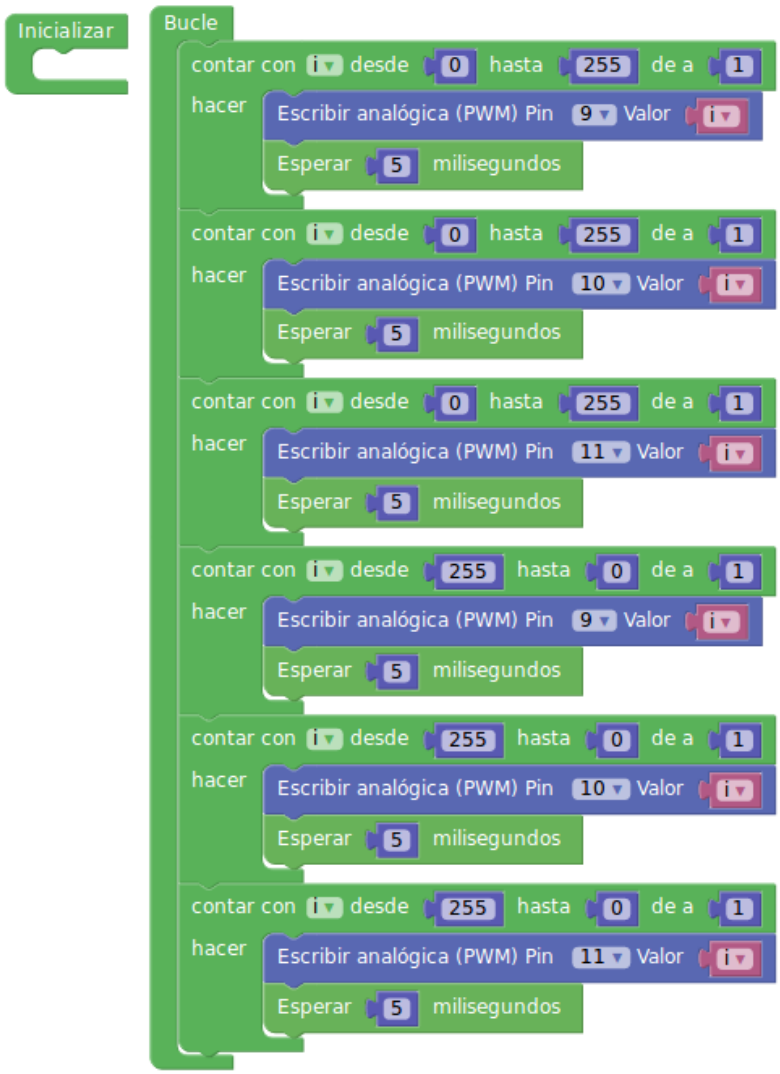
Secuencias con regulación de intensidad

CÓDIGO DE PROYECTO:

Conectar 3 leds en los pines 9,10 y 11.

Realizar una secuencia con aumento de intensidad de los leds progresivamente, luego va decrementando la intensidad también progresivamente.

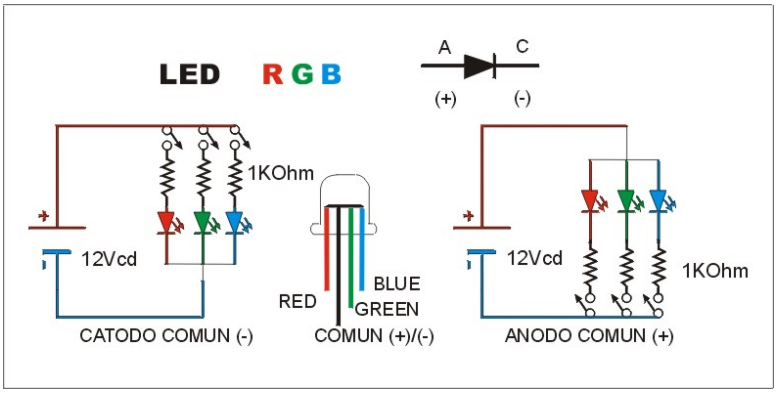


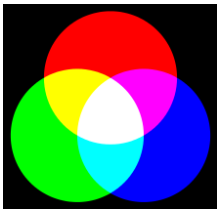


Led RGB

Un LED RGB es en realidad la unión de tres LEDs de los colores básicos, en un encapsulado común, compartiendo el cátodo (-) o el ánodo (+).

RGB: sigla en inglés de ***R**ed (Rojo), **G**reen (Verde) y **B**lue (Azul)*





Ejemplo de conexión de un led RGB de cátodo común a los pines 9~, 10~ y 11~

Led R

Arduino Pin 9 ~

Led G

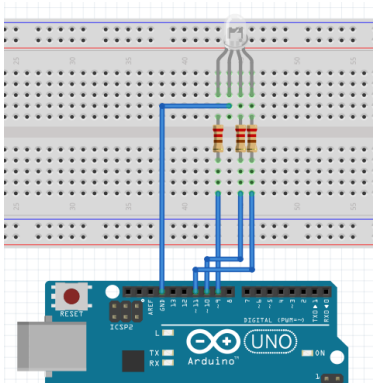
Arduino Pin 10 ~

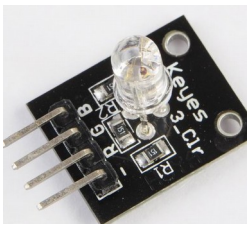
Led B

Arduino Pin 11 ~

Led GND

Arduino GND





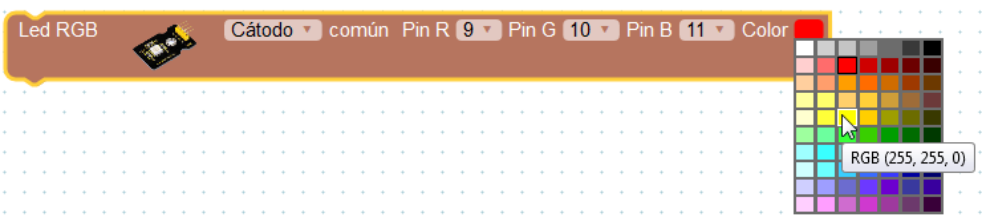
Módulo RGB
(con resistencias incorporadas)
B = Blue
G = Green
R = Red
- = GND

El led RGB podemos controlarlo de varias formas diferentes desde ArduinoBlocks:

Salidas digitales (ON/OFF)

Salidas analógicas (0...255)

Bloque Led RGB (Actuadores)

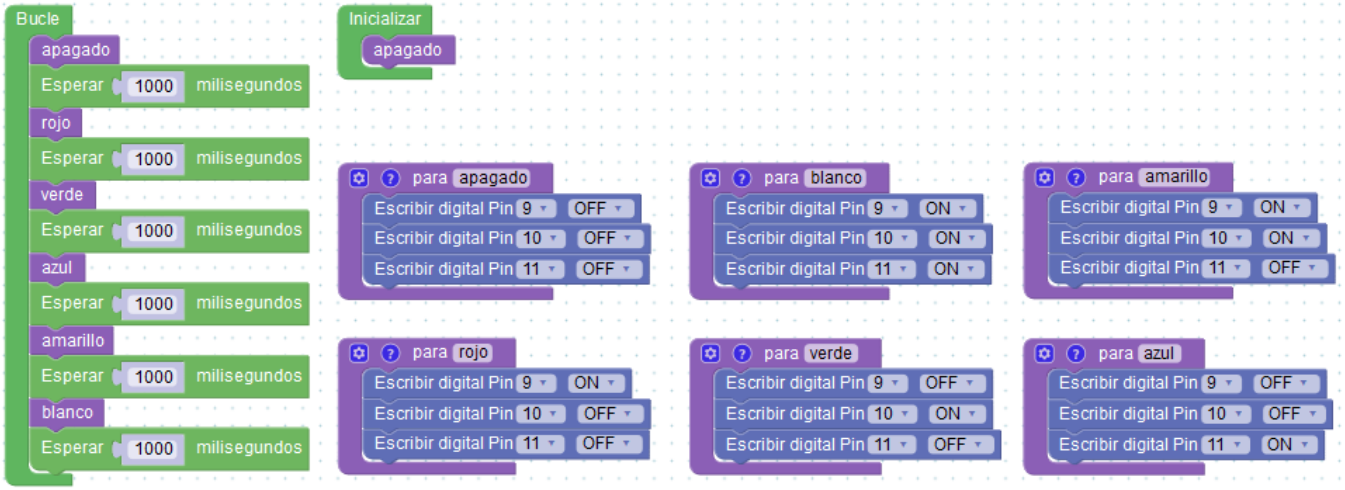


Led RGB - 1

Control ON/OFF del led RGB

CÓDIGO DE PROYECTO:

Con el uso de bloques para control de salidas digitales (ON/OFF) mostraremos el color verde, rojo, azul, amarillo (R+G), blanco (R+G+B) y negro (todo apagado) durante 1 segundo cada color. *(Led RGB de cátodo común)*

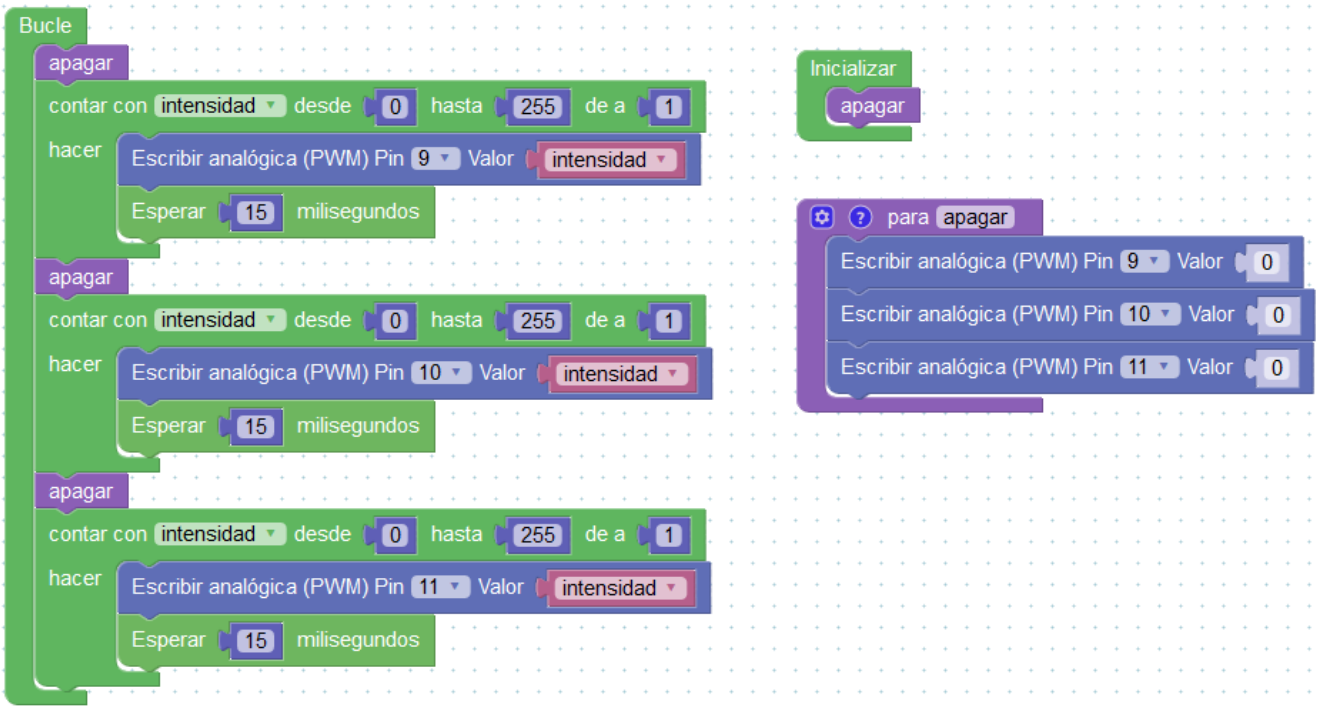


Led RGB - 2

Control PWM del led RGB

CÓDIGO DE PROYECTO:

Aumentaremos progresivamente la intensidad de Rojo, luego la de Verde y finalmente la del Azul
(Led RGB de cátodo común)

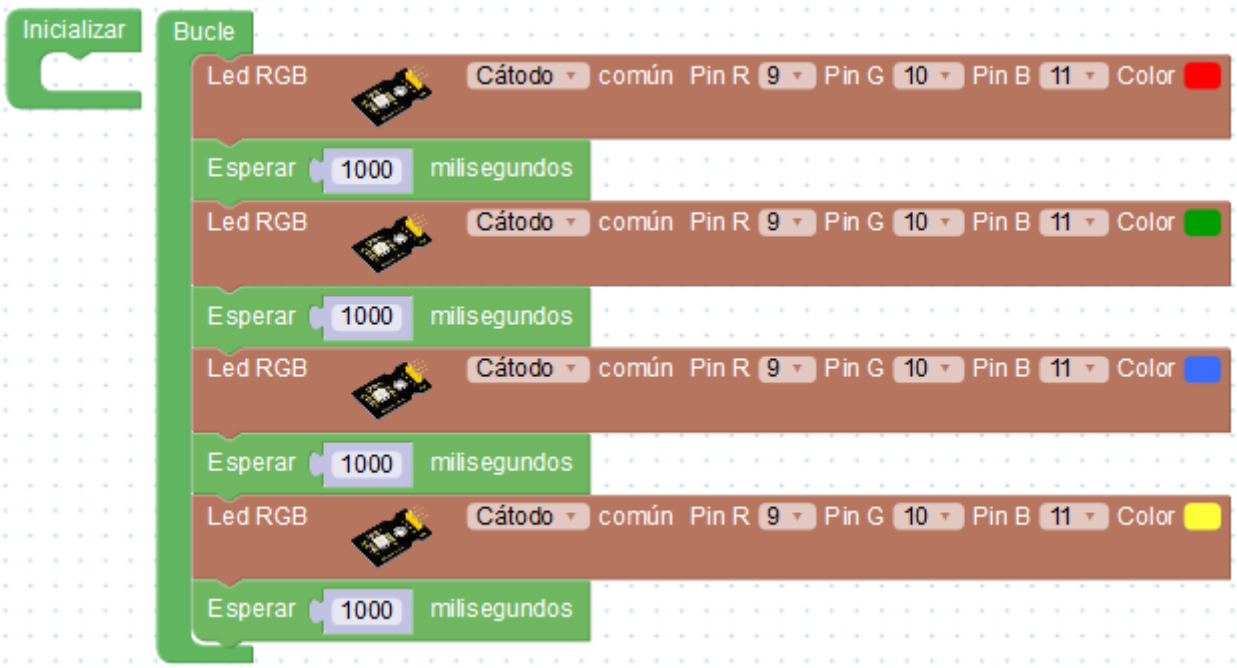


Led RGB - 3

Control con bloque de Led RGB

CÓDIGO DE PROYECTO:

Cambio secuencial de colores.



Led RGB - 4

Color aleatorio

CÓDIGO DE PROYECTO:

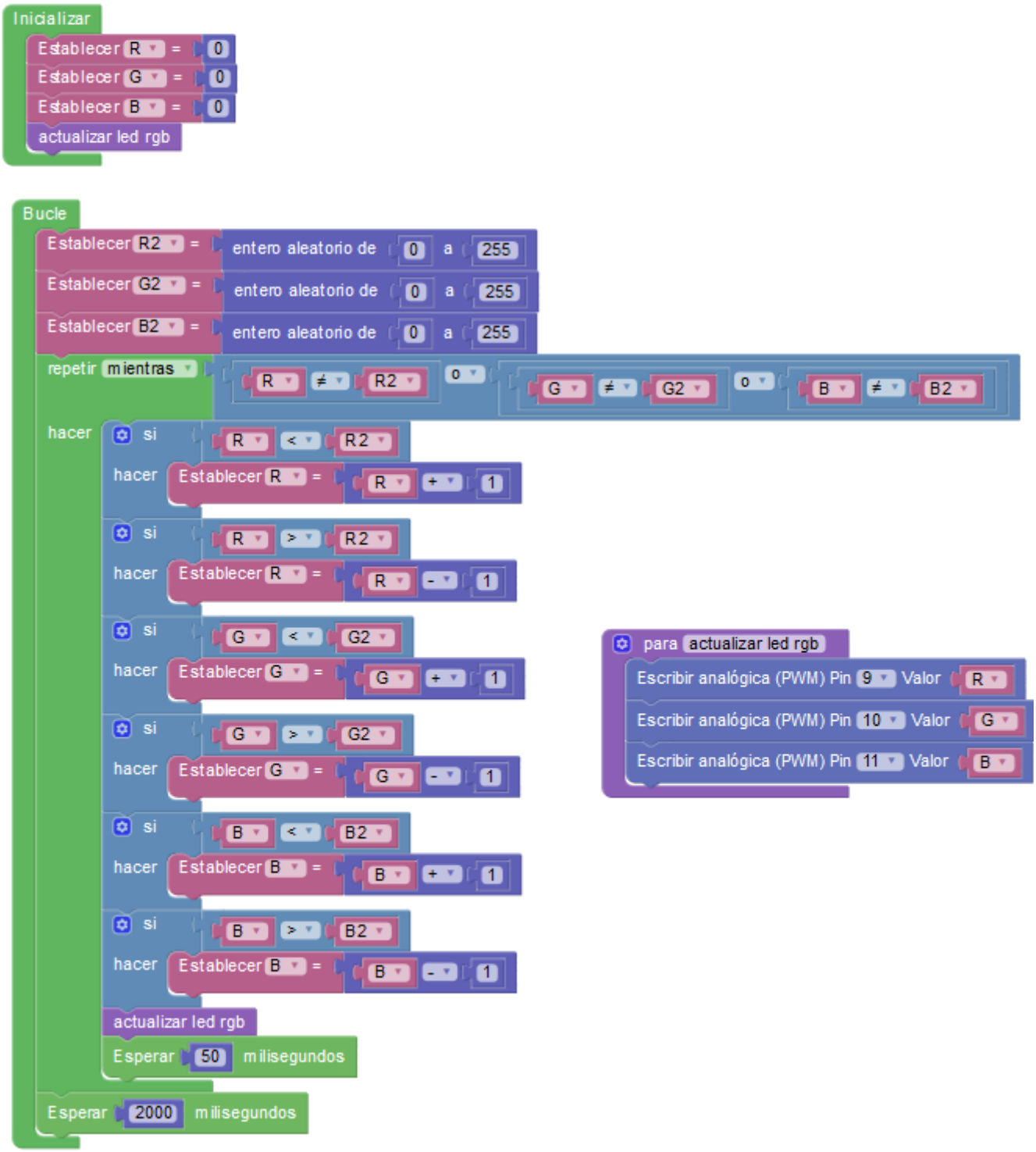
Cambio de color aleatoriamente cada 2 segundos

Led RGB - 5

Colores aleatorios con transición

CÓDIGO DE PROYECTO:

Genera colores RGB aleatorios y cambia del color actual al siguiente de forma suave.



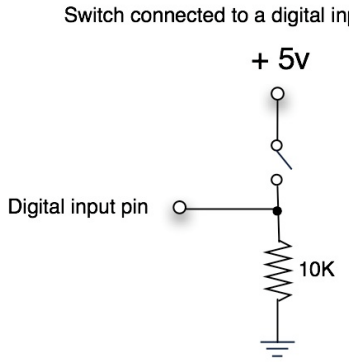
Entradas Digitales

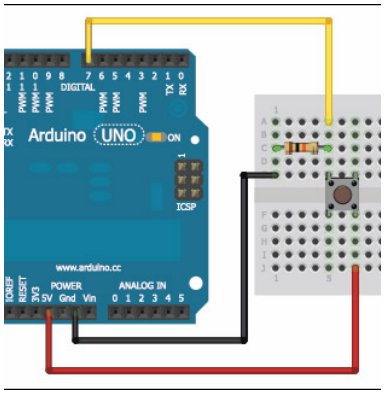
Arduino incorpora múltiples pines que pueden funcionar como entrada y salida. Vamos a probar a utilizar como entrada para leer datos de sensores externos.

Las entradas digitales permiten leer un valor ON / OFF según el voltaje que se aplique en el pin correspondiente.

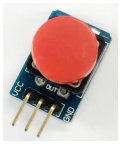
Si se aplica un voltaje menor de 2v se leerá un valor LOW (un "0" lógico)
Si se aplica un valor mayor de 3v se leerá un valor HIGH (un "1" lógico)

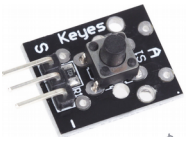
El esquema para conectar un pulsador/interruptor a una entrada digital de Arduino es:





En la mayoría de casos podemos utilizar un módulo de pulsador que incorpora la resistencia y simplifica las conexiones:





Algunos de estos módulos están conectados de forma invertida al esquema anterior, de forma que la entrada estará activa (HIGH/ON) en reposo y se desactivará (LOW/OFF) al pulsar,

Los bloques utilizados para leer el valor de una entrada digital o un pulsador son:

Pulsador o sensor con salida ALTO/HIGH/ON cuando está pulsado o activo

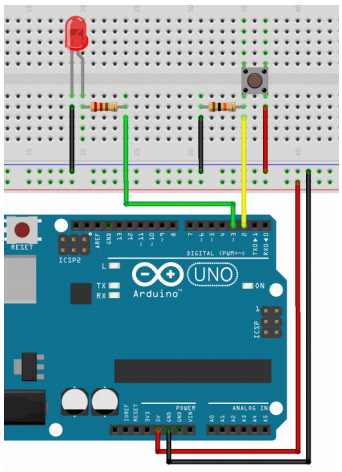
Pulsador o sensor con salida BAJO/LOW/OFF cuando está suelto o inactivo

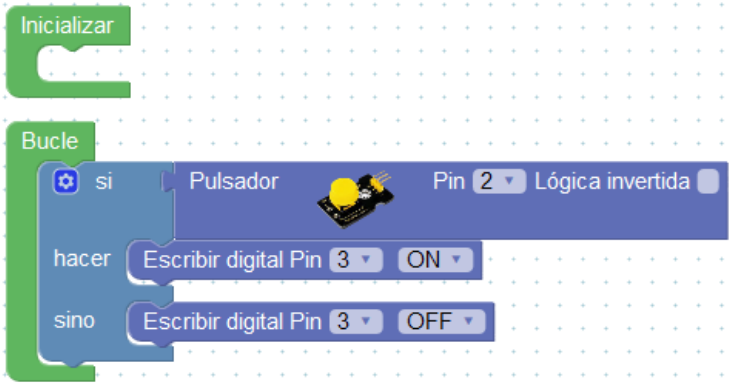
Entradas Digitales -1

Encender un led con pulsador

CÓDIGO DE PROYECTO:

Encenderemos un led mientras el pulsador esté pulsado, si no el led permacerá apagado



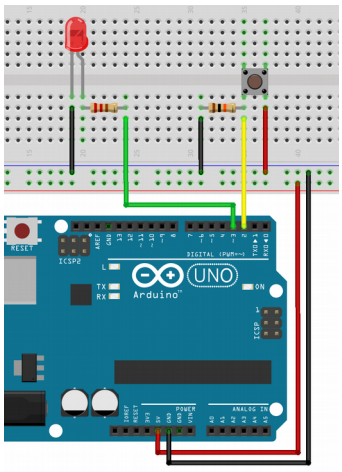


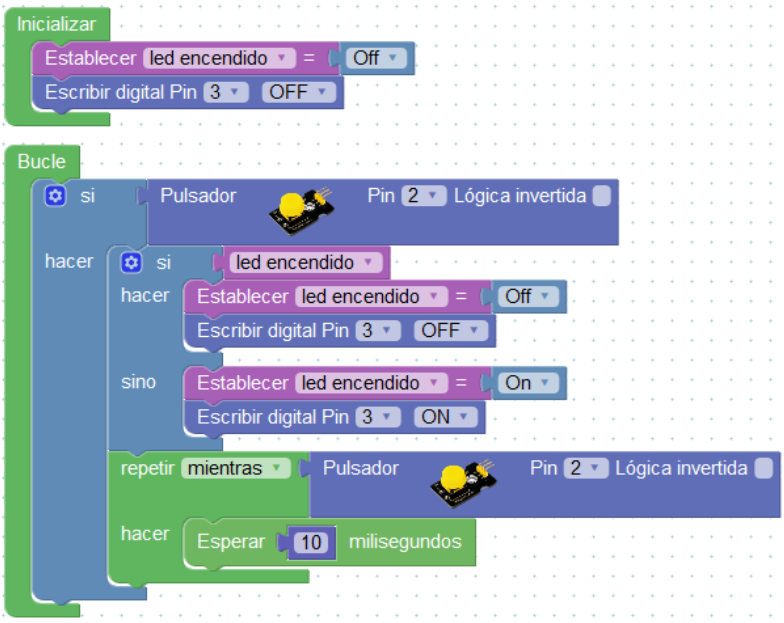
Entradas Digitales -2

Conmutación de un led con un pulsador

CÓDIGO DE PROYECTO:

Encender y apagar un led con un único pulsador.





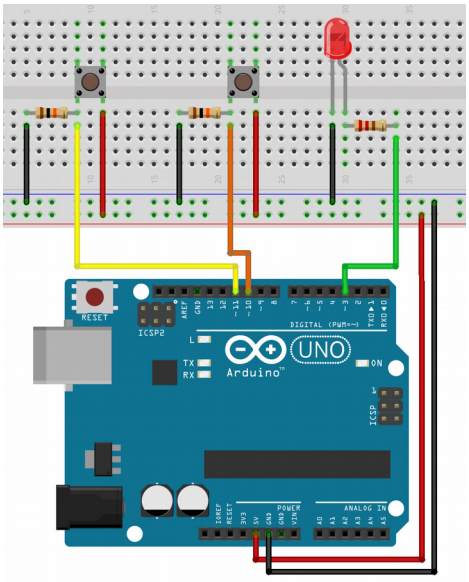
¿Para qué sirve el bloque “repetir mientras...” ?
¿Qué pasa si lo quitamos?

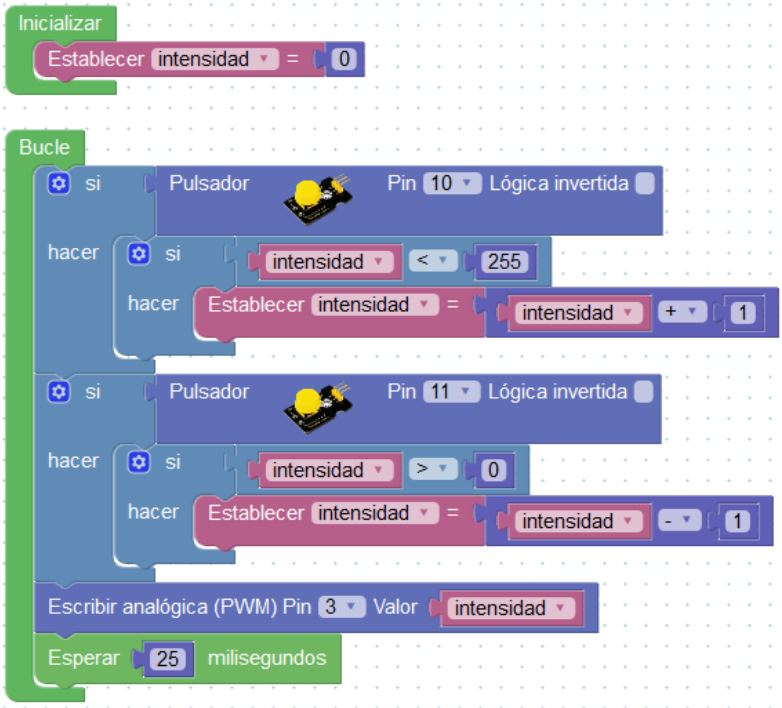
Entradas Digitales -3

Control de intensidad con dos pulsadores

CÓDIGO DE PROYECTO:

Conectaremos dos pulsadores y un led. Un pulsador aumentará la intensidad del led y otra la disminuirá.



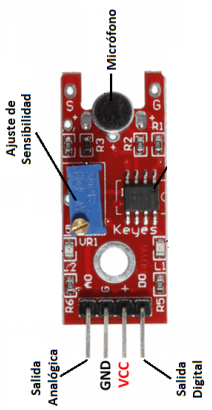


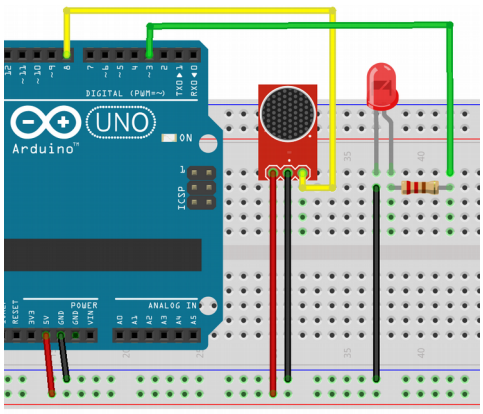
Entradas Digitales -4

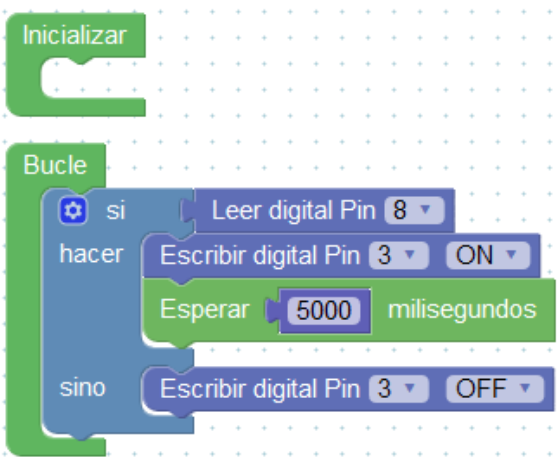
Control de led con palmada

CÓDIGO DE PROYECTO:

Utilizaremos un sensor de sonido con salida digital (0/OFF sin sonido y 1/ON cuando detecta sonido).
El programa encederá el led durante 5s cuando detecte un sonido fuerte.
(Comprueba los pines del sensor, seguramente con coinciden con el del esquema. Debes conectar la salida digital D0 del sensor al pin 8)







Entradas Digitales - 5

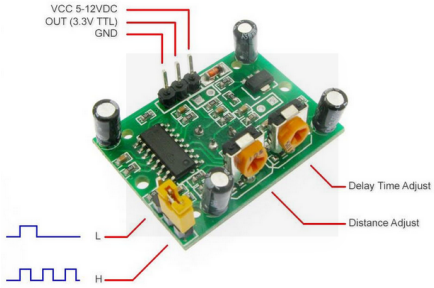
Detector de movimiento PIR

CÓDIGO DE PROYECTO:

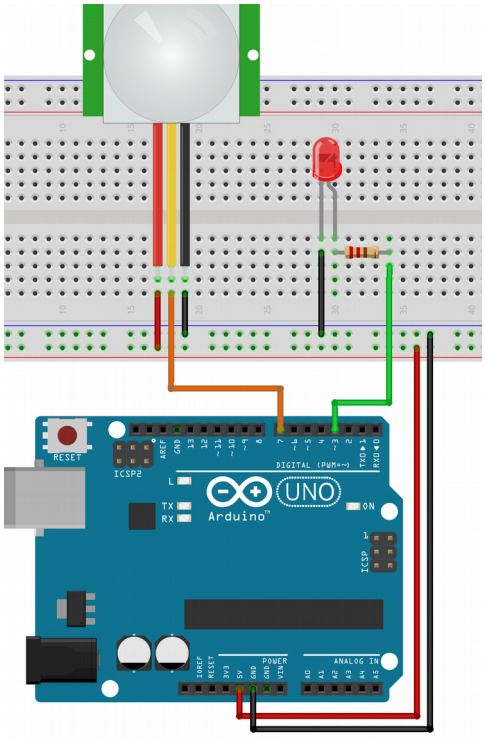
El sensor de movimiento PIR (Passive Infrared) o Pasivo Infrarrojo, reaccionan sólo ante determinadas fuentes de energía tales como el calor del cuerpo humano o animales. Reciben la variación de las radiaciones infrarrojas del medio ambiente que cubre. Es llamado pasivo debido a que no emite radiaciones, sino que las recibe. Estos captan la presencia detectando la diferencia entre el calor emitido por el cuerpo humano y el espacio alrededor.

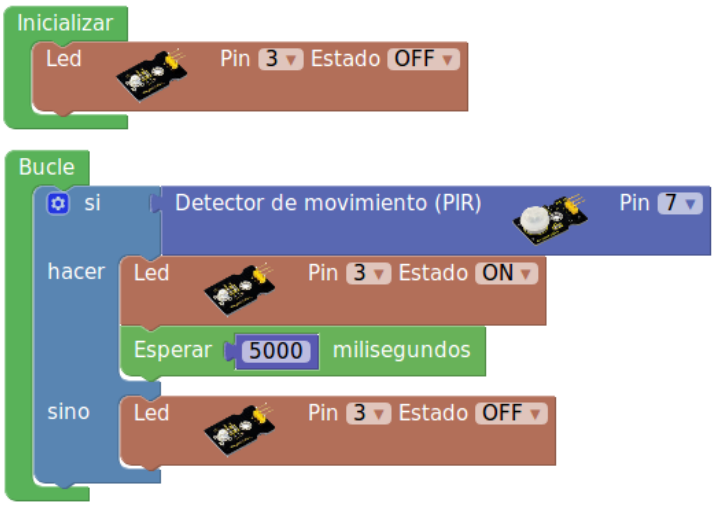
Sensor PIR / Módulo sensor PIR

Sensor PIR ajustable



Realizar un programa que encienda durante 5s un led conectado en el pin 3 cuando detecte movimiento. El sensor de movimiento PIR lo conectaremos al pin 7.





Comunicación Serie

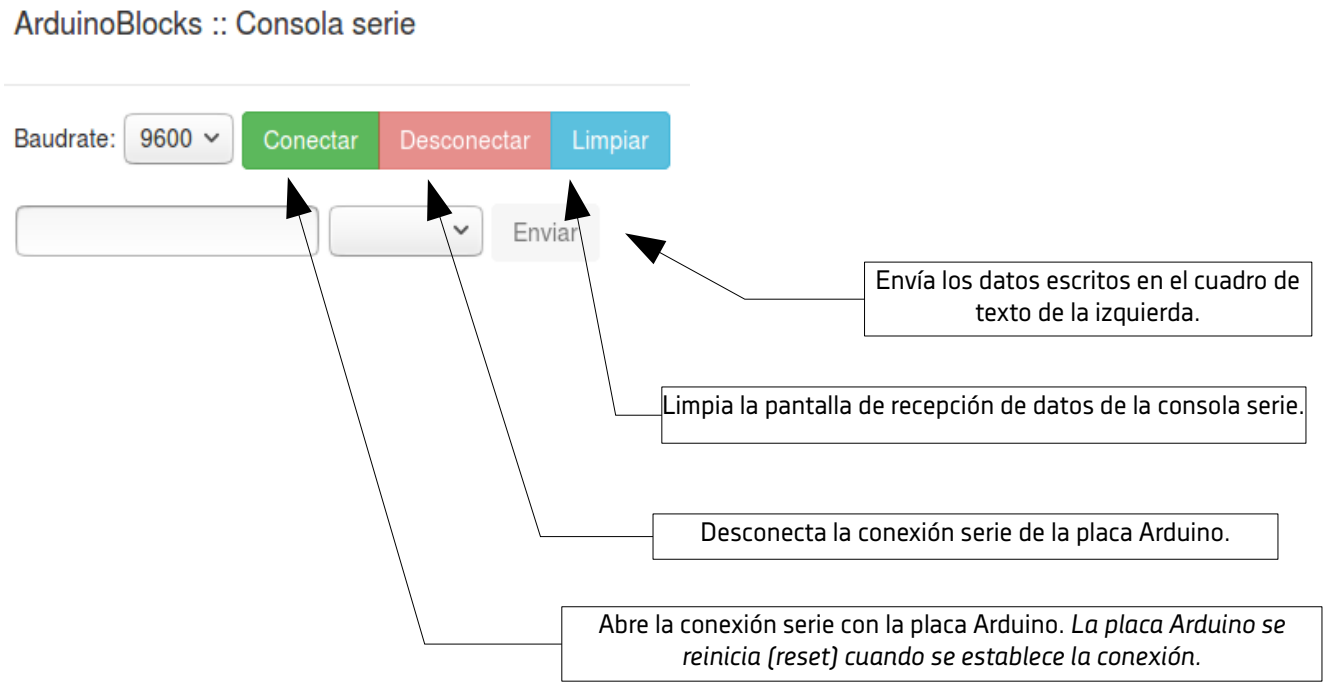
Arduino incorpora una conexión serie que permite conexión con el PC (o con otros muchos dispositivos). Esta es la misma conexión se utiliza para subir el programa al Arduino. Utilizando esta conexión podemos enviar información desde Arduino al PC y al revés.

Consola de ArduinoBlocks

Para mostrar la información recibida en el PC y poder enviar datos al Arduino lo más fácil es utilizar un terminal serie o **consola serie.**

La consola serie permite enviar datos desde el PC a la placa Arduino. Y al revés, permite recibir y visualizar los datos recibidos desde la placa Arduino.

Opciones de la consola serie de ArduinoBlocks:



Comunicación Serie-1

Enviar mensajes desde Arduino

CÓDIGO DE PROYECTO:

El programa enviará mensajes de texto desde Arduino, para visualizar los datos recibidos desde la conexión serie utilizaremos la consola que incorpora ArduinoBlocks.

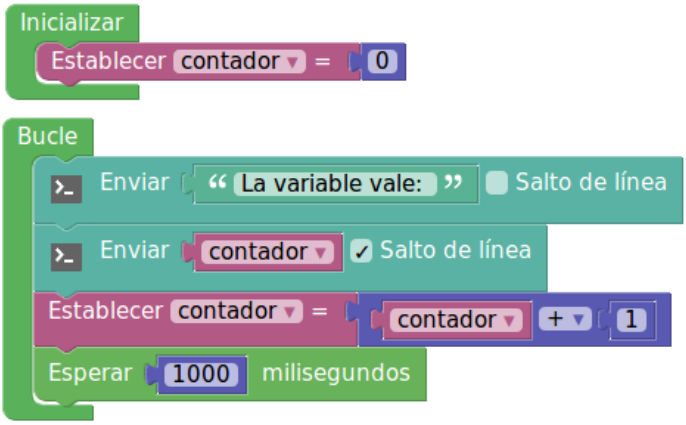
Prueba a utilizar otros programas de terminal / consola serie:
https://sourceforge.net/projects/realterm/
https://sourceforge.net/projects/hypeterminal/

Comunicación Serie-2

Visualizar el valor de una variable

CÓDIGO DE PROYECTO:

Vamos a ver como enviar el valor de una variable, en este caso una variable que va incrementándose y mostramos en la consola serie su valor.



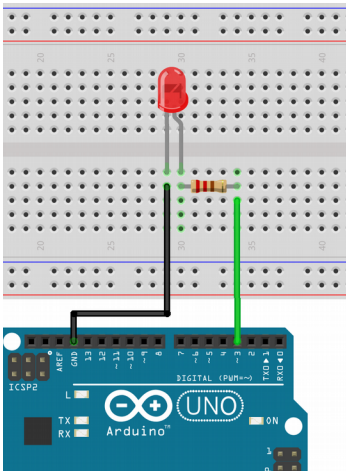

```
La variable contador vale: 1.00
La variable contador vale: 2.00
La variable contador vale: 3.00
La variable contador vale: 4.00
La variable contador vale: 5.00
La variable contador vale: 6.00
La variable contador vale: 7.00
La variable contador vale: 8.00
La variable contador vale: 9.00
```

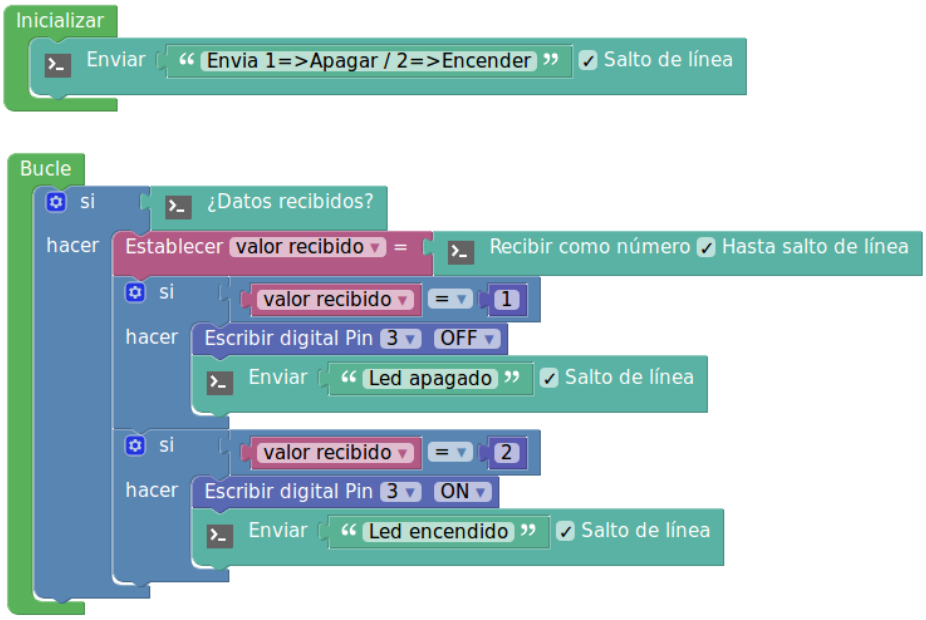
Comunicación Serie-3

Encendido de un led desde el PC

CÓDIGO DE PROYECTO:

Al recibir el valor 1 desde la consola apagaremos el led, al recibir el 2 lo encenderemos.



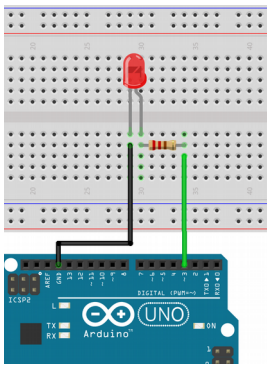


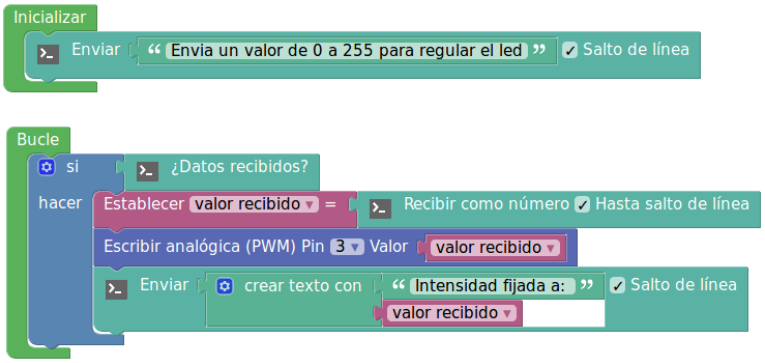
Comunicación Serie-4

Regulación de intensidad de led desde PC

CÓDIGO DE PROYECTO:

Conectar un led al pin 3. Recibir un número desde el ordenador a través del puerto serie. El número recibido debe estar entre 0 y 255 y se escribirá en la salida analógica (PWM) del pin 3.



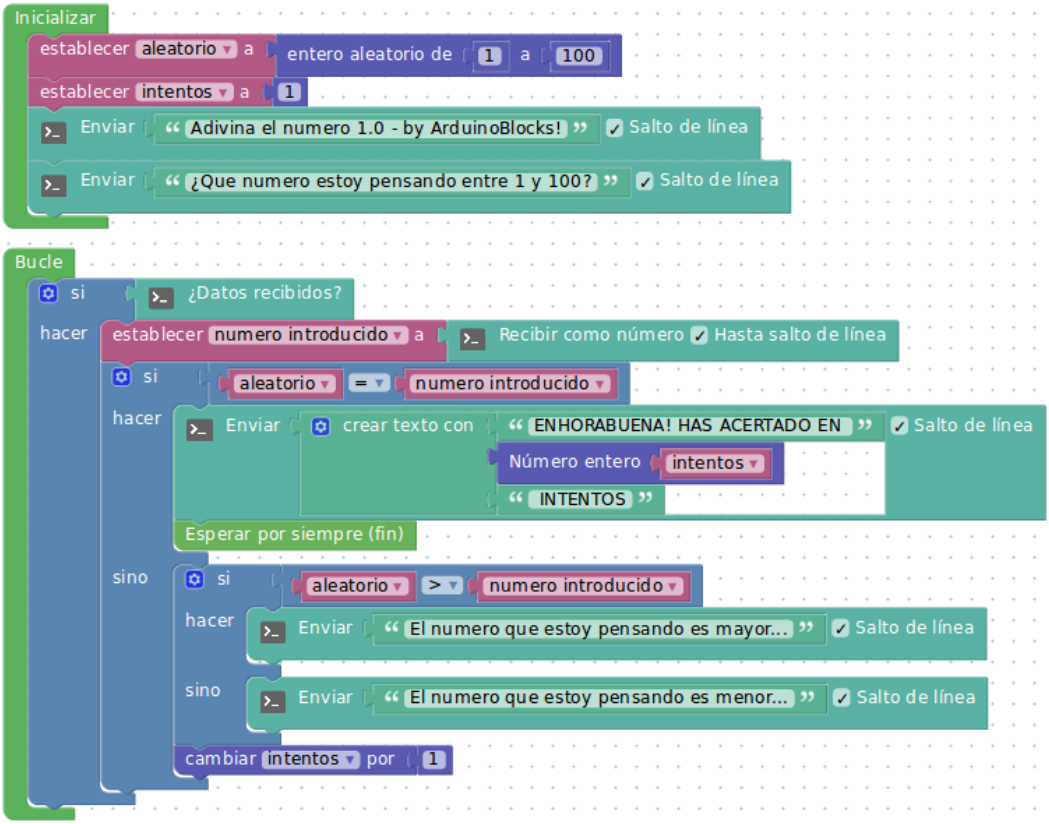


Comunicación Serie-5

Juego: Adivina el número

CÓDIGO DE PROYECTO:

Realizaremos un juego donde la placa Arduino "pensará" un número al azar entre 1 y 100.
Desde la consola iremos diciendo números y nos dirá si el número secreto es mayor o menor hasta que lo adivinemos y nos muestre el número total de intentos que hemos usado.

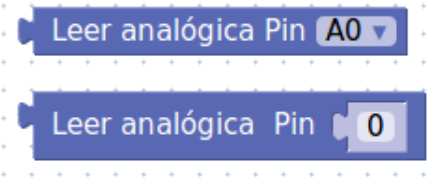


Entradas Analógicas

Arduino incorpora 6 pines que pueden funcionar como entradas analógicas, vamos a probar cómo podemos leer datos de sensores externos a través de ellas.

Las entradas analógicas permiten leer el voltaje que se le aplica como entrada. Ese voltaje podrá variar entre 0 y 5v. El valor del voltaje leído se convierte a un valor numérico comprendido entre 0 y 1023

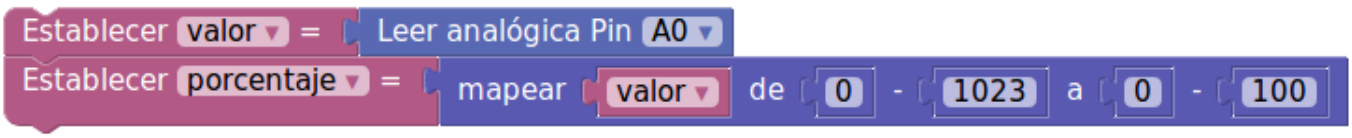
Bloques para leer una entrada analógica:



Los bloques de lectura de una entrada analógica devolverán un valor entre 0...1023

Voltios en la entrada	*Valor leído*
0v	0
2.5v	512
5v	1023

Otro bloque interesante es el “Mapear”, que permite cambiar el rango del valor leído. Por ejemplos si quiero cambiar el rango leído de 0...1023 a 0..100 puedo “mapearlo” de la siguiente manera:

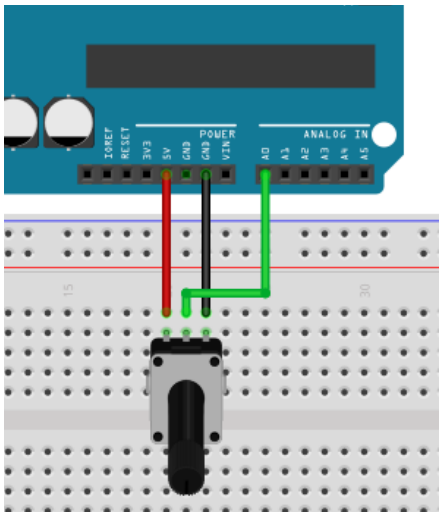


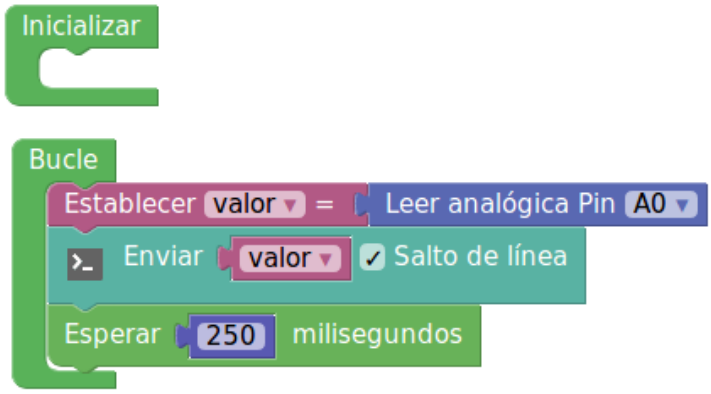
Entradas Analógicas -1

Leer la posición de un potenciómetro

CÓDIGO DE PROYECTO:

Leeremos el valor de la entrada analógica donde está conectada el potenciómetro y lo enviaremos a la consola serie para poder visualizarlo en el PC



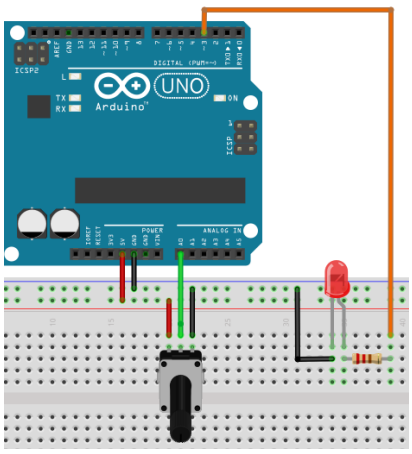


Entradas Analógicas - 2

Regular la intensidad de led con un poteciómetro

CÓDIGO DE PROYECTO:

Leeremos el valor de un potenciómetro (0...1023) y lo mapearemos a un valor proporcional entre 0 y 255 para regular un led conectado al pin 3 como salida PWM.



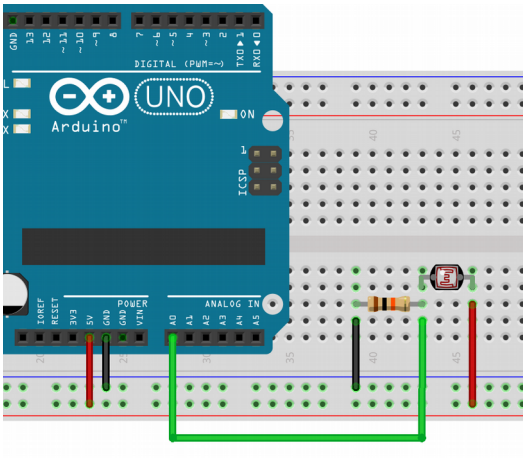
Entradas Analógicas - 3

Medidor de luz ambiente

CÓDIGO DE PROYECTO:

Conectaremos una resistencia LDR a la entrada analógica A0, a través de la cual mediremos el nivel de luz ambiente detectado.

Enviaremos el valor leído a través de la conexión serie para visualizarlo en la consola serie cada 2s

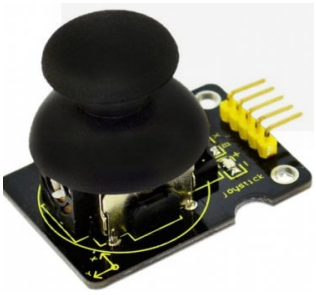


Entradas Analógicas - 4

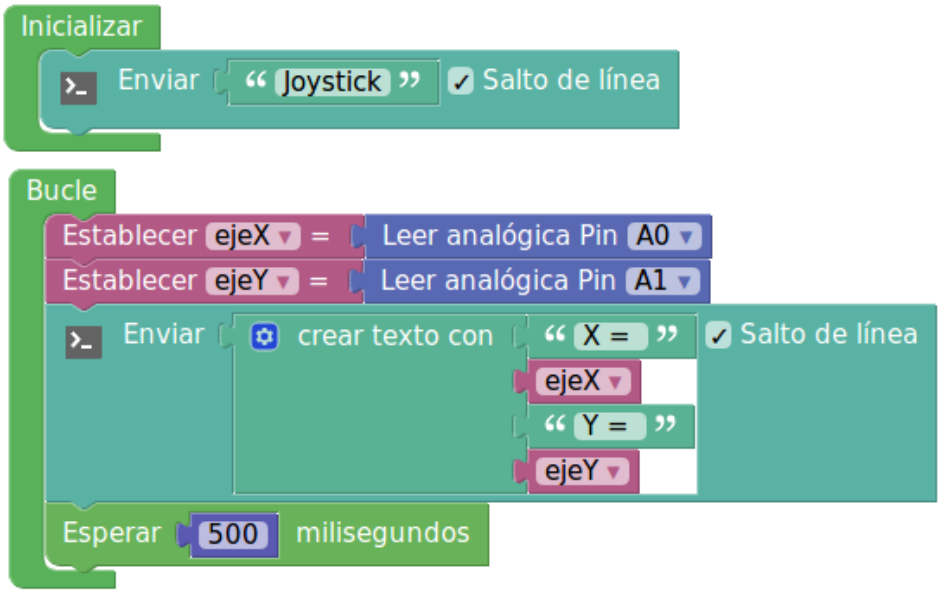
Joystick

CÓDIGO DE PROYECTO:

Los módulos tipo “joystick” para Arduino se componen de dos potenciómetros, uno para el movimiento del eje X y otro para el movimiento del eje Y. También en el mismo módulo suelen incorporar un pulsador.



GND / -	GND = 0v
VCC / 5V / +	VCC = 5v
Vrx / X	Potenciómetro del eje X (a un pin analógico)
Vry / Y	Potenciómetro del eje Y (a un pin analógico)
SW	Botón (a un pin digital)

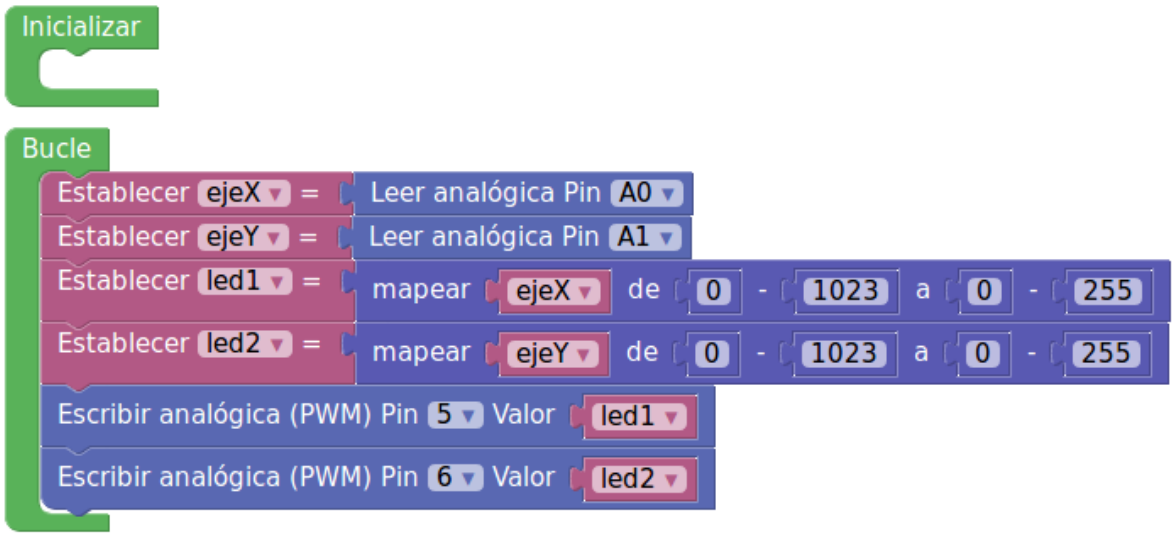


Entradas Analógicas - 5

Control de dos leds con joystick

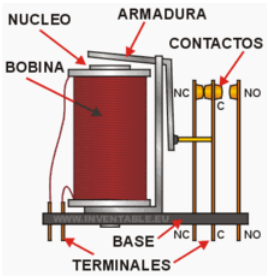
CÓDIGO DE PROYECTO:

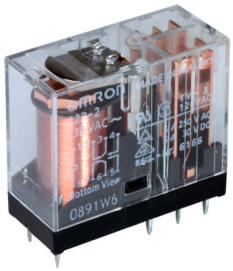
Conectar un joystick (X => A0 / Y => A1) y dos leds a los pines 6 y 7 respectivamente. Un led variará su intensidad con el eje X y otro con el eje Y. En posición de reposo del joystick los dos leds deberán estar iluminados a la mitad de intensidad aproximadamente.



Relé

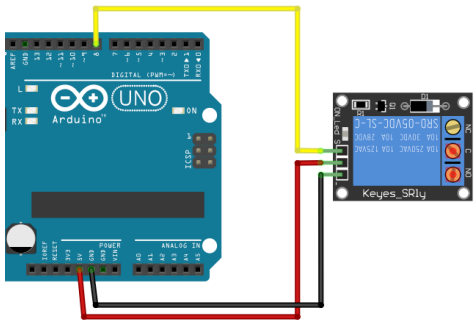
El relé es un dispositivo electromagnético. Funciona como un interruptor controlado por un circuito eléctrico en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes





La conexión del relé la realizaremos a través de una salida digital, pues sólo tenemos dos estados ON / OFF (relé activo o no)

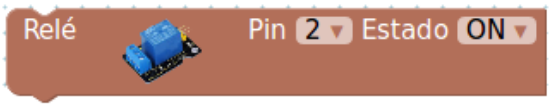
Lo más habitual es conectar un módulo relé que simplifica la conexión pues ya tiene todo los elementos necesarios para la conexión del relé directa a Arduino. Sólo debemos conectar 5V (VCC) , GND y la señal de entrada de activación del relé que estará conectada al pin de Arduino correspondiente.



Para activar el relé en el programa Arduino simplemente utilizaremos la instrucción para activar el pin conectado al módulo relé de forma digital (ON / OFF)

En el apartado salida tenemos un módulo de relé que internamente hace la misma función que la instrucción “escribir digital”

Estas dos instrucciones realizan la misma función: Activar la salida del pin 2

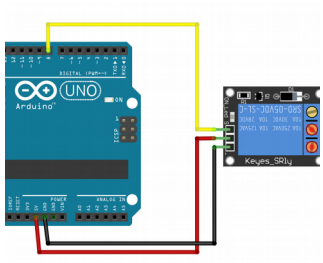


Relé - 1

Intermitencia con relé

CÓDIGO DE PROYECTO:

Realizar un montaje con el módulo de relé conectado al pin 8, realizaremos un programa que active y desactive el relé en períodos de un 1s.

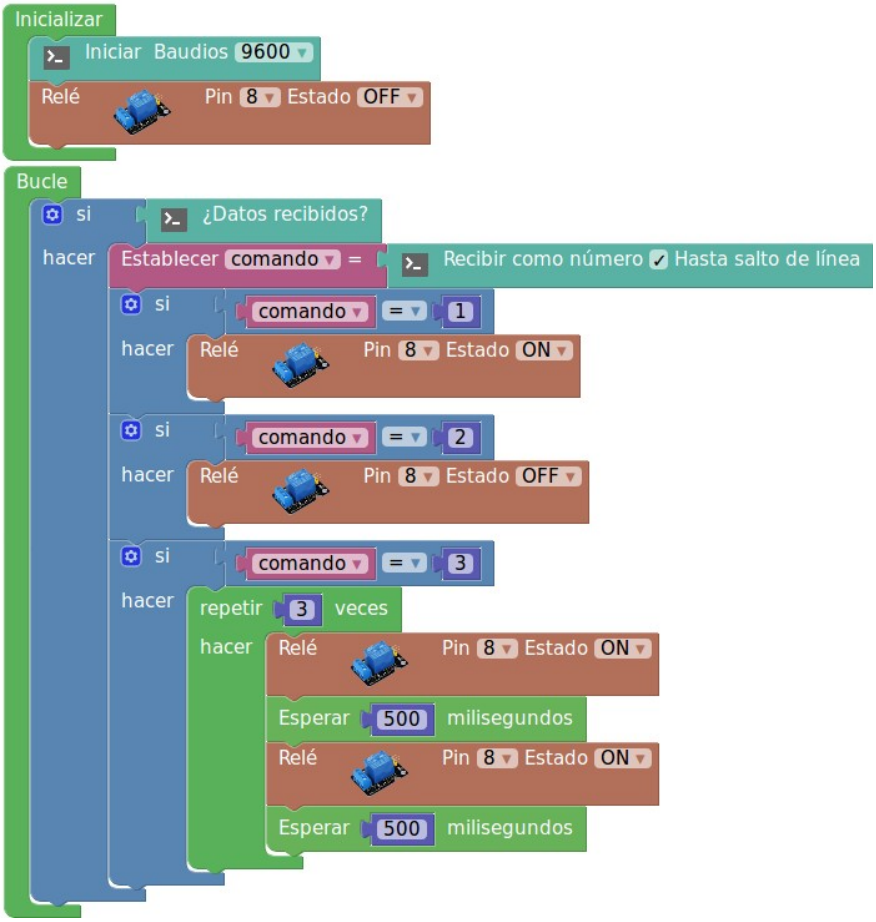


Relé - 2

Control de relé desde el PC

CÓDIGO DE PROYECTO:

Encenderemos/apagaremos el relé conectado al pin 8 a través de la consola del puerto serie.
1 = encender / 2 = apagar / 3 = parpadear

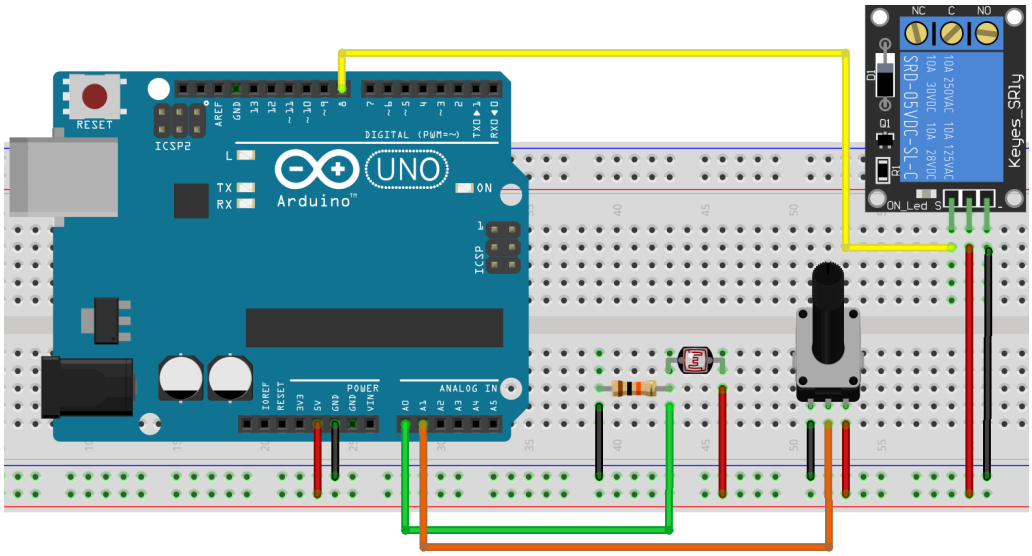


Relé - 3

Relé activado por nivel de luz (ajustable)

CÓDIGO DE PROYECTO:

Realizaremos un programa que active el relé cuando el nivel de luz sea mayor que el valor leído del potenciómetro y lo desactive en caso contrario.

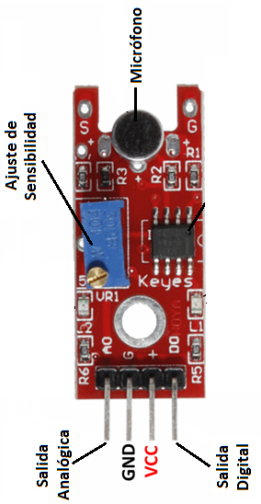


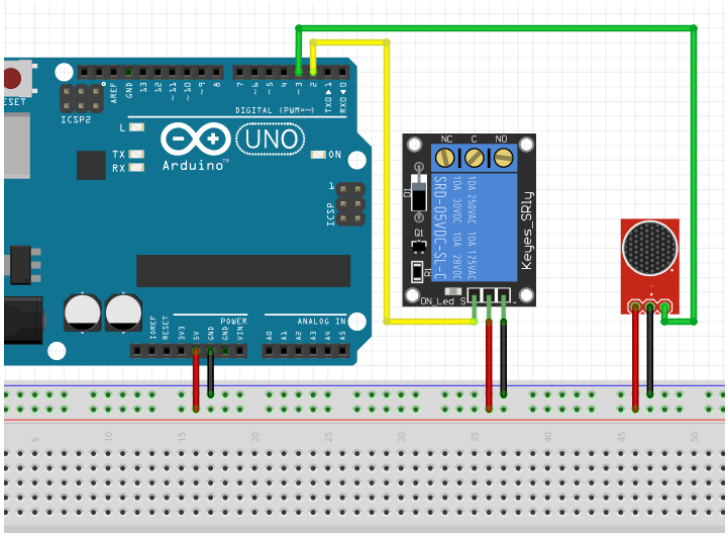
Relé - 4

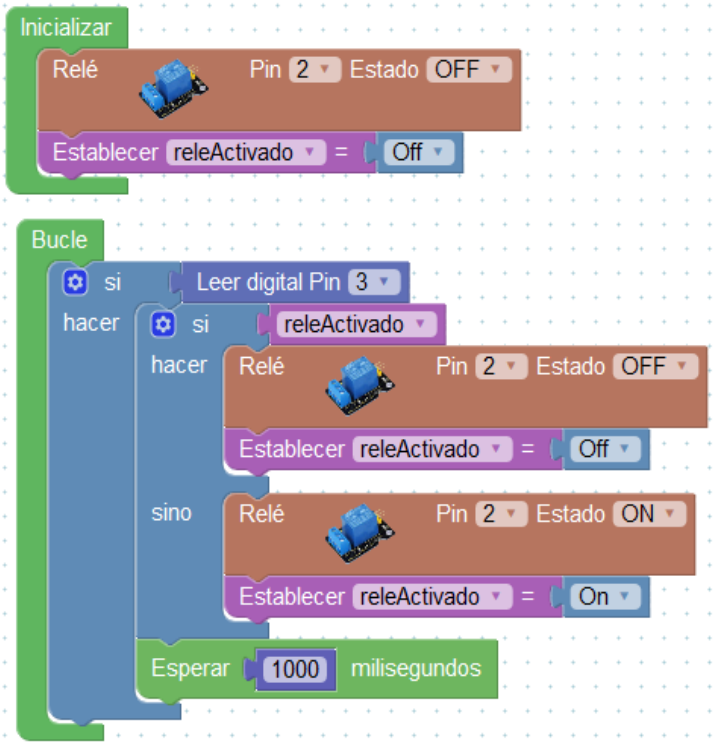
Control de un relé con sonido

CÓDIGO DE PROYECTO:

Realizaremos un control de activación y desactivación de un relé con sonido (una palmanda por ejemplo)
Conectaremos un sensor de sonido al pin 3 y el relé al pin 2.







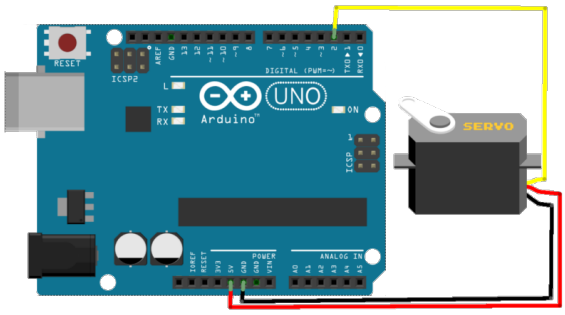
Servomotor

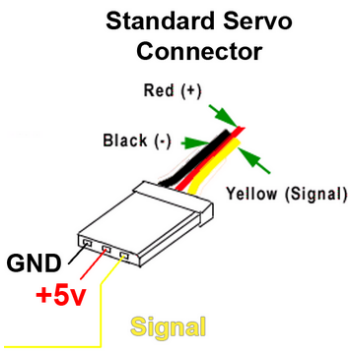
Un servomotor (normalmente llamado simplemente "servo") es un dispositivo similar a un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición

Está conformado por un motor, una caja reductora y un circuito de control.

Los servomotores hacen uso de la modulación por ancho de pulsos (PWM) para controlar la dirección o posición de los motores de corriente continua.

microservo 9g





Ángulo → Posición en grados donde mover el servo. Depende del modelo del servo podremos moverlos en un rango de ángulo. **Los microservos 9g (utilizados habitualmente) funcionan entre 0º y 180º**

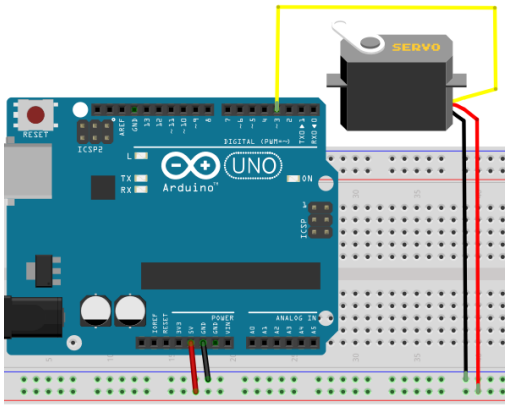
Retardo (ms)→ Retardo que se añade para esperar a que el servo se mueva hasta la posición indicada. Puede ser 0, pero deberemos controlar en el resto del programa el retardo hasta el próximo movieminto para darle tiempo a llegar a la posición indicada.

Servomotor - 1

Movimiento básico

CÓDIGO DE PROYECTO:

El programa realiza movimientos simples del servomotor a distintas posiciones de forma secuencial. El objetivo es comprobar el funcionamiento correcto del servo y determinar sus posiciones mínima y máxima. Prueba con distintos valores de "retardo" para ver el resultado.



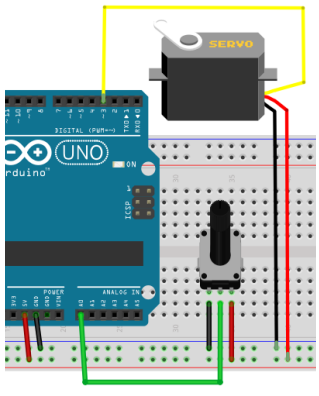
Realiza una secuencia similar parando en 5 posiciones y esperando 4s en cada una
Posiciones de ejemplo: 10º , 50º, 80º, 130º, 170º

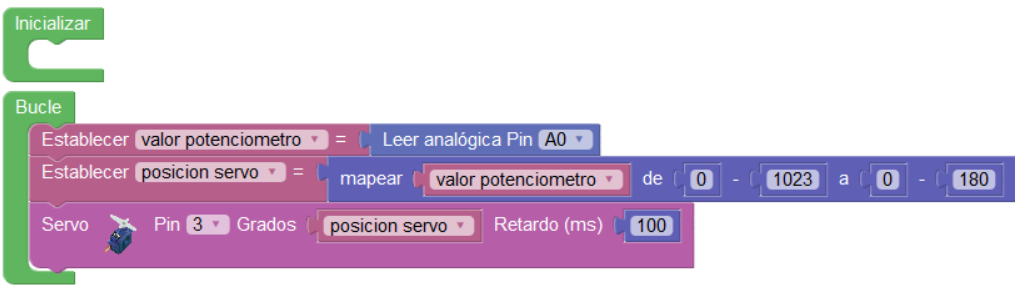
Servomotor - 2

Control de servo con potenciómetro

CÓDIGO DE PROYECTO:

El servo conectado en el pin 3 se moverá a la posición indicada por un potenciómetro conectado al pin A0.
El valor leído del servo va de 0...1023 por lo que se debe mapear a un valor proporcional entre 0...180º antes de posicionar el servo.



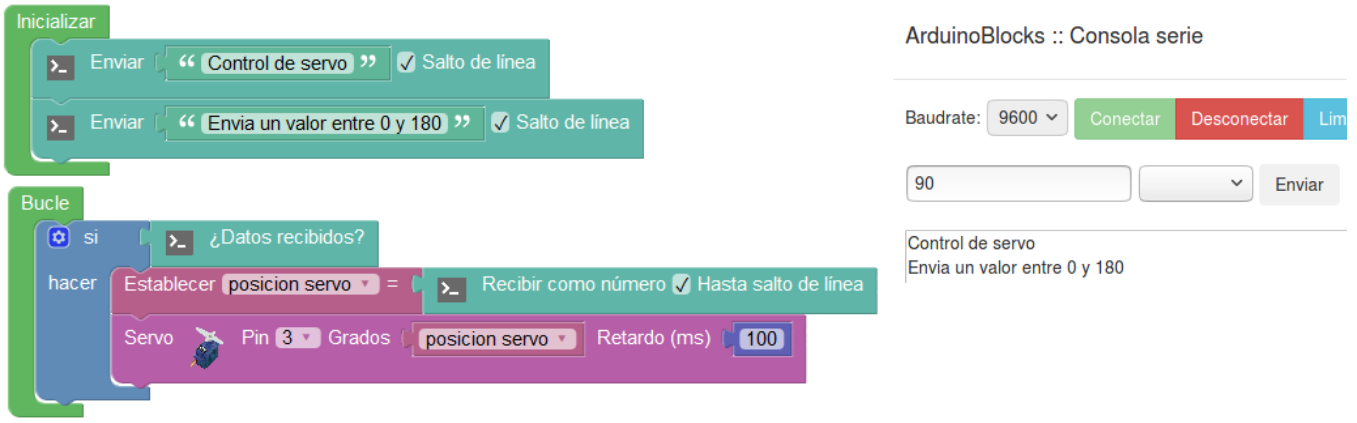


Servomotor - 3

Control de servo desde PC

CÓDIGO DE PROYECTO:

Con el mismo montaje que la práctica *Servomotor-1* realizaremos un programa que reciba un número desde el puerto serie (utlizaremos la consola serie para enviar el número). El número deberá ser entre 0 y 180 y moverá el motor a dicha posición.



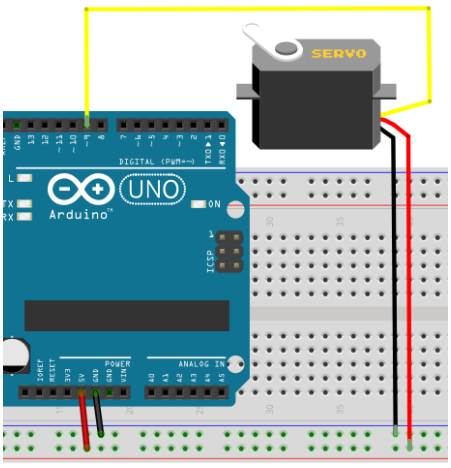
Servomotor - 4

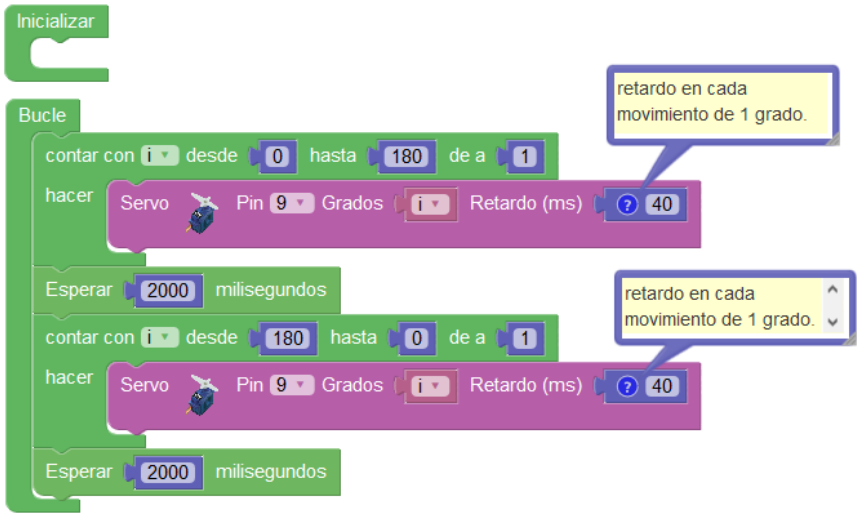
Movimiento progresivo

CÓDIGO DE PROYECTO:

Realizaremos un programa que mueva el servo suavemente y con una velocidad regulada con retardos. Se aumenta de 1 grado en 1 grado desde 0 hasta 180º grados y luego al revés de 180 a 0º.

El retardo en cada movimiento del servo determinará la velocidad.
Ejemplo: con un retardo de 40ms x 180 posiciones (grados) = 7200 ms, el servo tardará 7,2s de un extremo al otro.





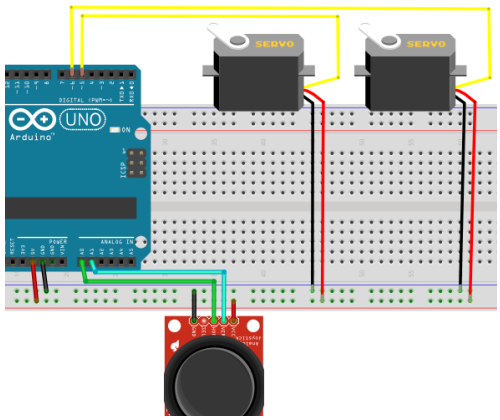
Servomotor - 5

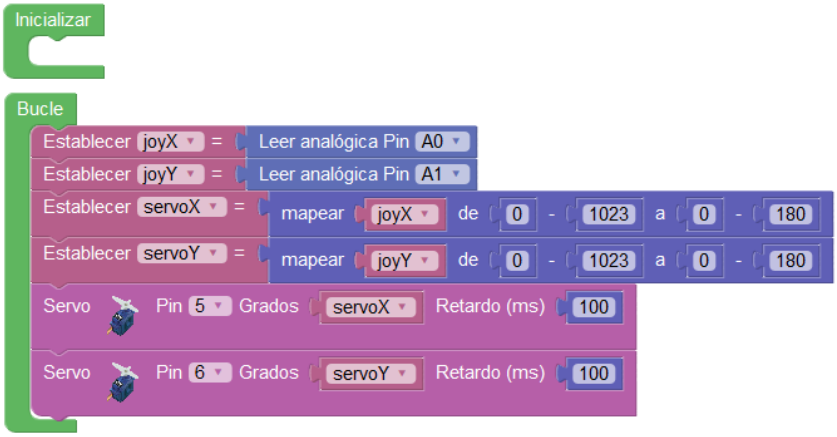
Control de dos servos con joystick

CÓDIGO DE PROYECTO:

Con la ayuda de un josytick:
Pin VRX → Pin A0
Pin VRY → Pin A1

Moveremos dos servos:
Servo 1 → Pin ~5
Servo 2 → Pin ~6





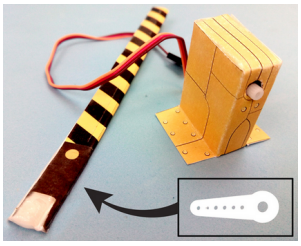
Para subir nota...
¿serías capaz de realizar un movimiento progresivo
de cada servo controlado con el joystick?

Servomotor - 6

Control de un servo con un pulsador

CÓDIGO DE PROYECTO:

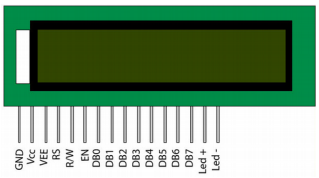
Diseña un proyecto para controlar un servo con un pulsador.
El proyecto funcionará simulando la apertura de una barrera de seguridad (movida por el servo). Al pulsar el botón la barrerá subirá y se mantendrá levantada durante 5s y automáticamente después se bajará.



Pantalla LCD

La pantalla de cristal líquido (LCD) es un dispositivo empleado para la visualización de contenidos o información de una forma gráfica, mediante caracteres, símbolos o pequeños dibujos dependiendo del modelo. Está gobernado por un microcontrolador interno el cual dirige todo su funcionamiento.

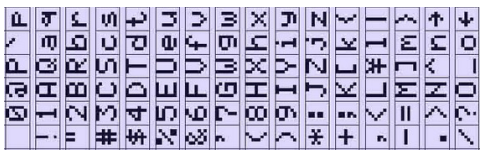
La pantalla puede ser de 16x2 (16 caracteres de ancho y 2 líneas) o 20x4 (20 de ancho y 4 líneas)



Conexiones de una pantalla LCD

GND (o VSS)	Conexión a 0V / GND
VCC	Conexión a 5V
VEE	Corresponde al pin de contraste, lo regularemos con un potenciómetro de 10K conectado a Vdd.
RS	Corresponde al pin de selección de registro de control de datos (0) o registro de datos(1)
R/W	Corresponde al pin de Escritura(0) o de Lectura(1).
EN	Corresponde al pin Enable o de habilitación. Si EN=0 esto quiere decir que el LCD no esta activado
DB0...DB7	Bus de datos bidireccional. La comunicación con el LCD podemos hacerlo utilizando los 8 bits del bus de datos(D0 a D7) o empleando los 4 bits mas significativos del bus de datos(D4 a D7)
Led +	Led de luz de fondo (opcional)
Led -	

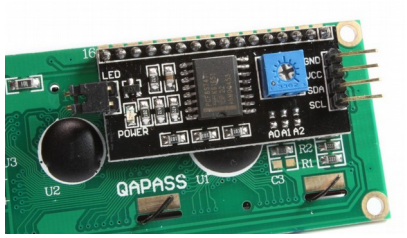
Caracteres imprimibles en el display LCD:

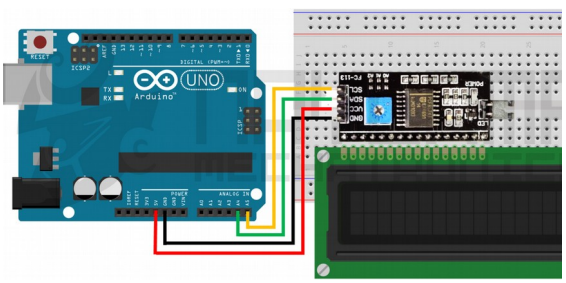


Las pantallas LCD tienen la opción de definir hasta 8 caracteres personalizados, como por ejemplo:

La conexión de la pantalla LCD a Arduino se puede realizar de dos formas:

- Conexión usando 4 bits de datos + Señales de control RS / EN
- Conexión usando un módulo I2C. Simplifica la conexión y permite conectar el LCD sólo con 2 cables al bus I2C (pins A4 y A5 de Arduino).



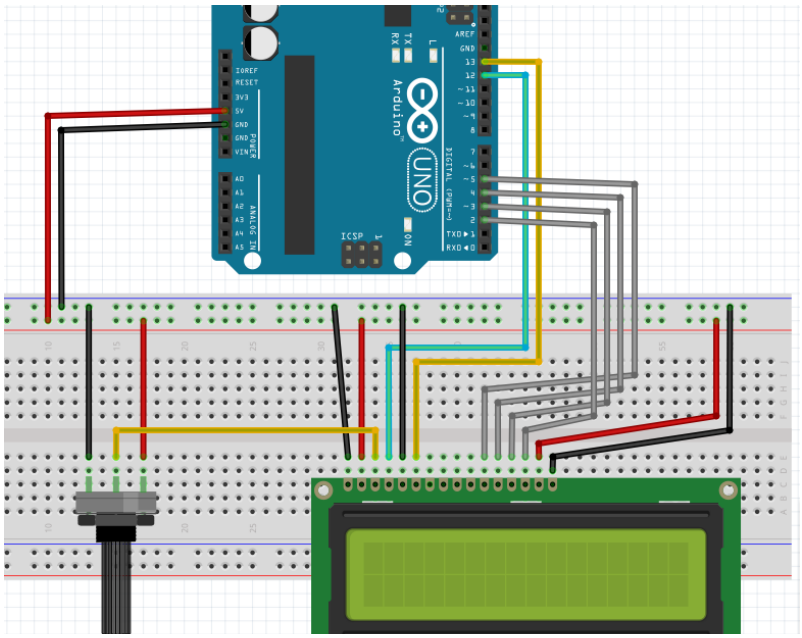


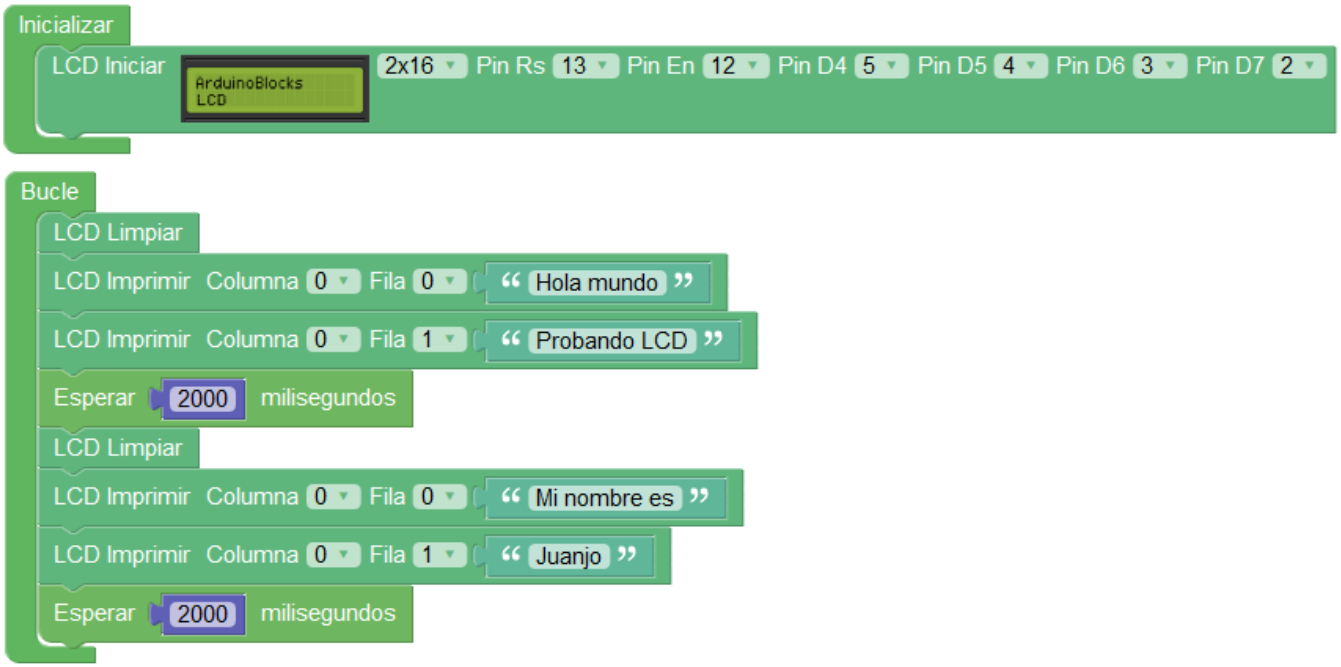
Pantalla LCD - 1

Mensajes en LCD / conexión de 4 bits + RS + EN

CÓDIGO DE PROYECTO:

Conectaremos una pantalla LCD según el siguiente esquema y realizaremos un programa para mostrar mensajes de prueba cada 2 segundos.





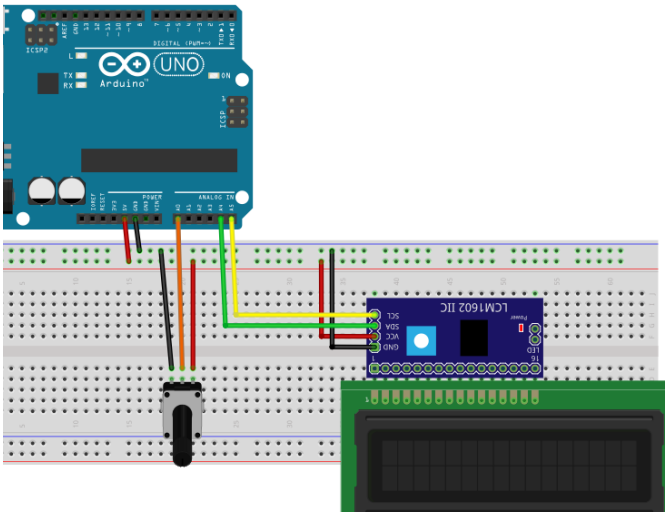
Modifica el programa anterior para mostrar 5 mensajes a tu elección

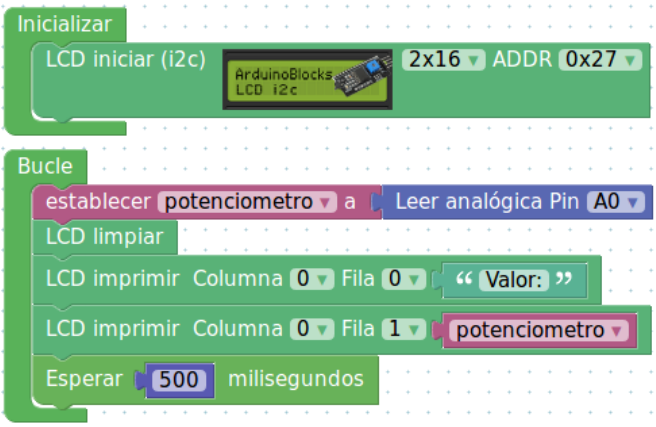
Pantalla LCD - 2

Visualizar la posición de un potenciómetro (%) / LCD (i2c)

CÓDIGO DE PROYECTO:

Conectar la pantalla LCD con el módulo I2C (A4=SDA / A5=SCL).
Conectar un potenciómetro a la entrada analógica A0.



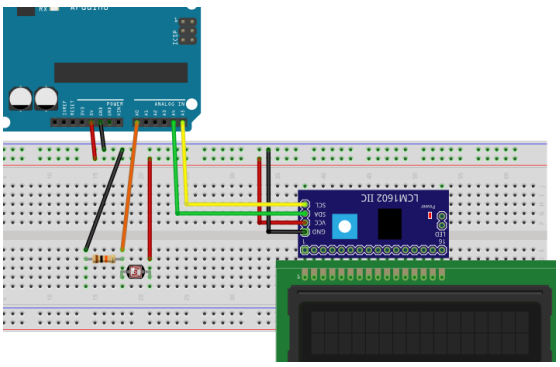


Pantalla LCD - 3

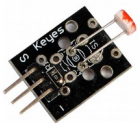
Medidor de luz ambiente / LCD (i2c)

CÓDIGO DE PROYECTO:

Conectar la pantalla LCD con el módulo I2C.
Conecta un LDR a la entrada A0 (con una resistencia de 10k)



Si disponemos de módulos LDR listos para usar podemos simplificar el circuito (el módulo lleva la resistencia incorporada, sólo debemos conectar 5V, GND y la Señal al pin A0):



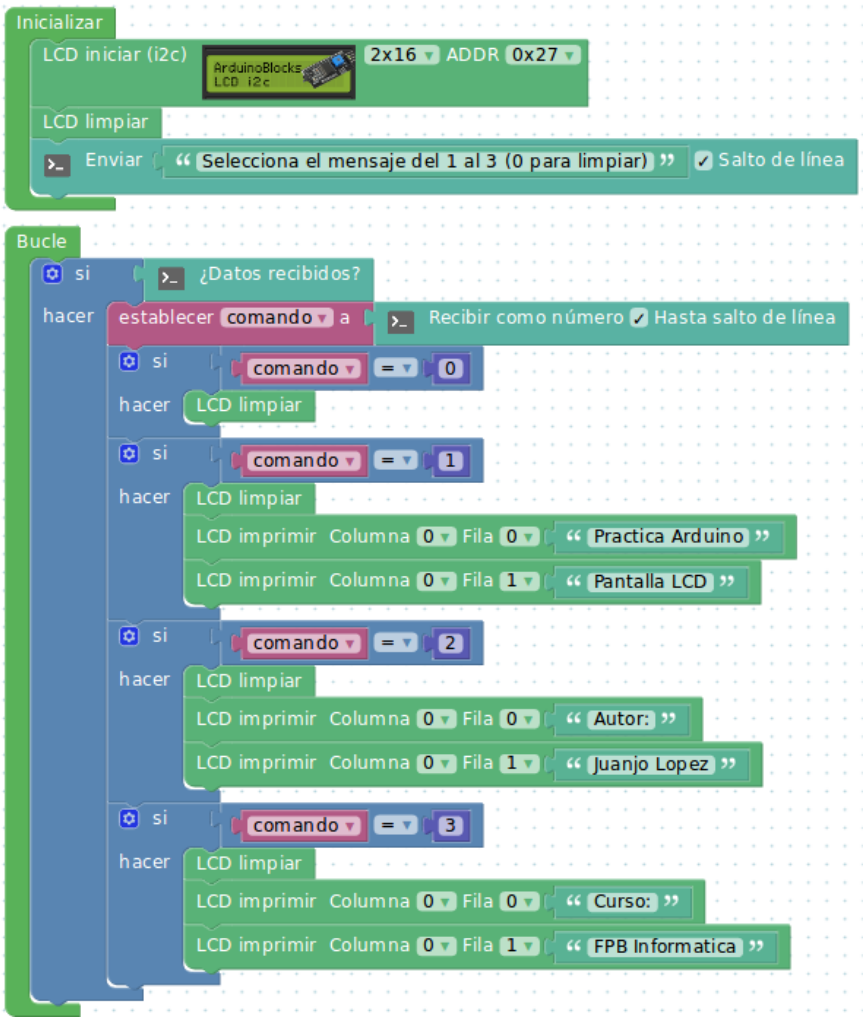
Pantalla LCD - 4

Mensajes seleccionado desde PC / LCD (i2c)

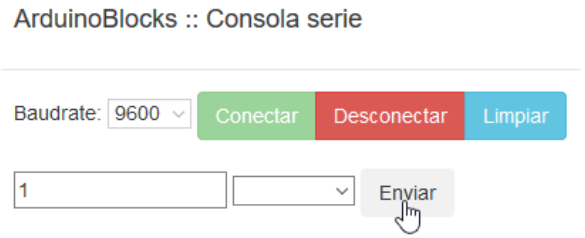
CÓDIGO DE PROYECTO:

Realiza un programa que reciba un número desde la consola serie.

Si recibe el valor “0” → borra la pantalla LCD
Si recibe el valor “1” → mensaje: “Practica Arduino” “Pantalla LCD”
Si recibe el valor “2” → mensaje: “Autor:” “tu nombre”
Si recibe el valor “3” → mensaje: “Curso:” “tu curso“

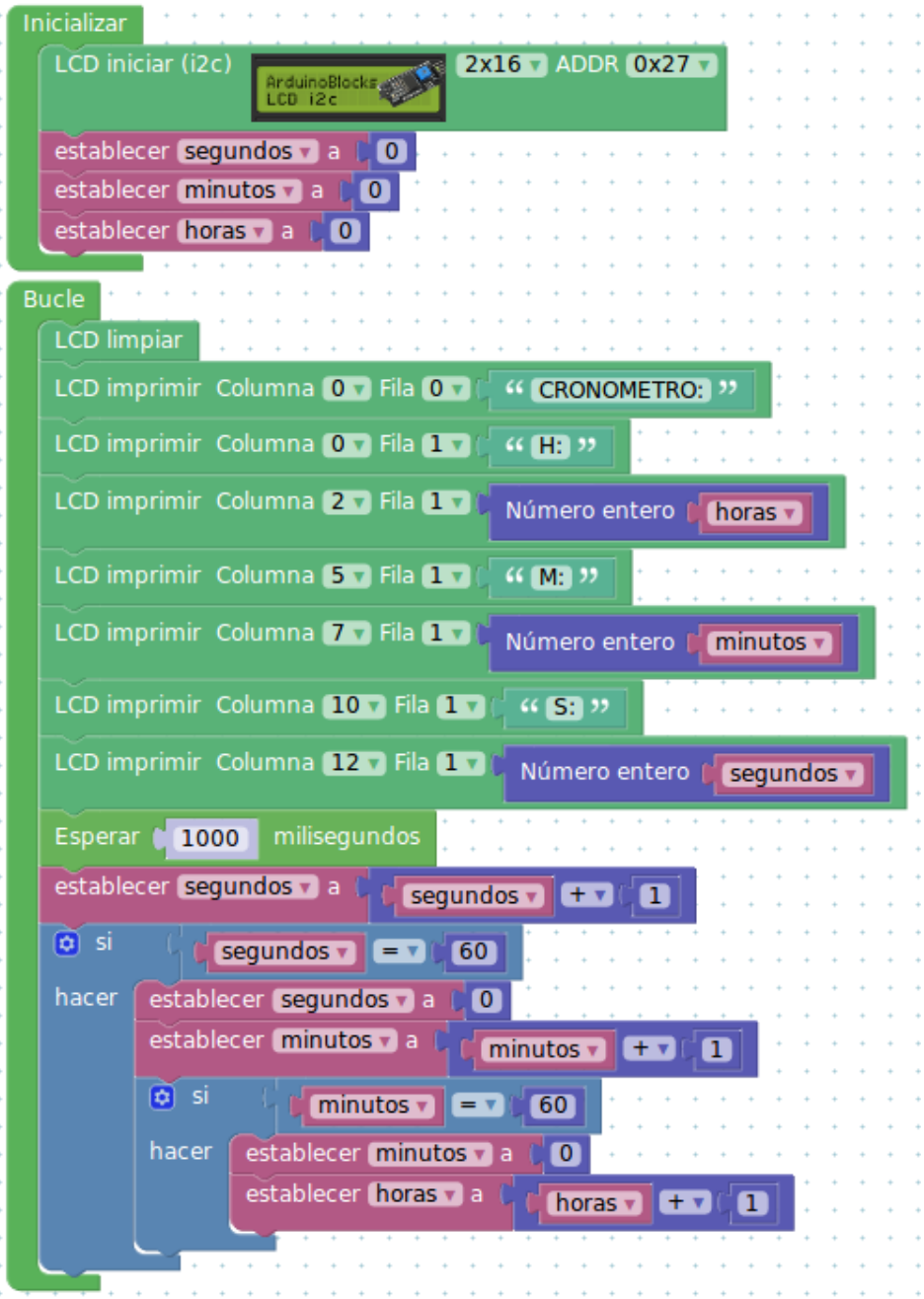


Ejemplo: envío del valor “1” por la consola para visualizar el primer mensaje:



Pantalla LCD - 5 Cronómetro / LCD (i2c) CÓDIGO DE PROYECTO:

Realizamos un montaje sencillo, simplemente conectando la pantalla LCD con el módulo i2c.
El programa mostrará por la pantalla LCD un cronómetro que contará segundos, minutos y horas.



Para subir nota...
Realiza un cronómetro de "cuenta atrás", empezando en un valor de hora, minuto y segundos que vaya descontando hasta llegar a 0h 0m 0s

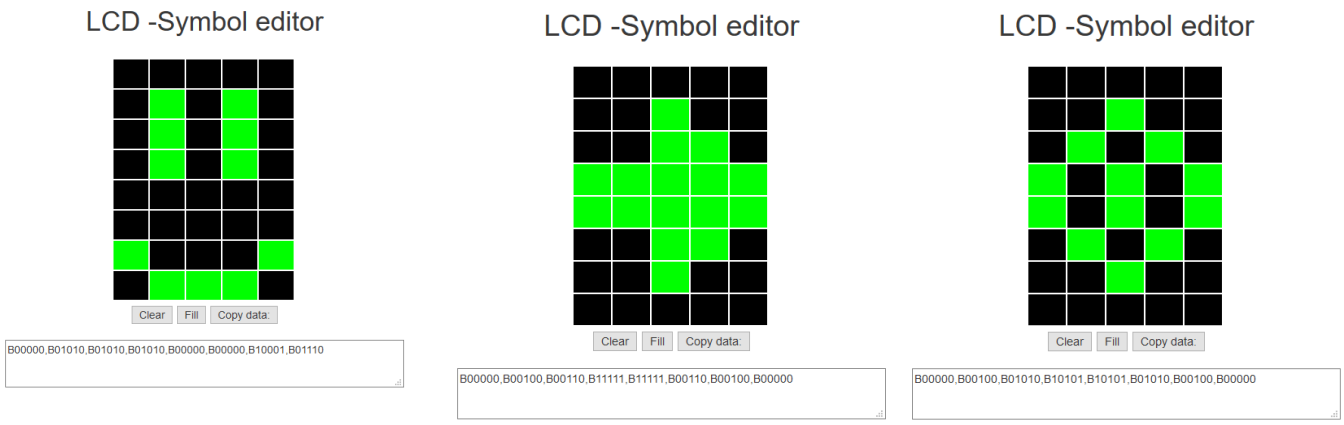
Pantalla LCD - 6

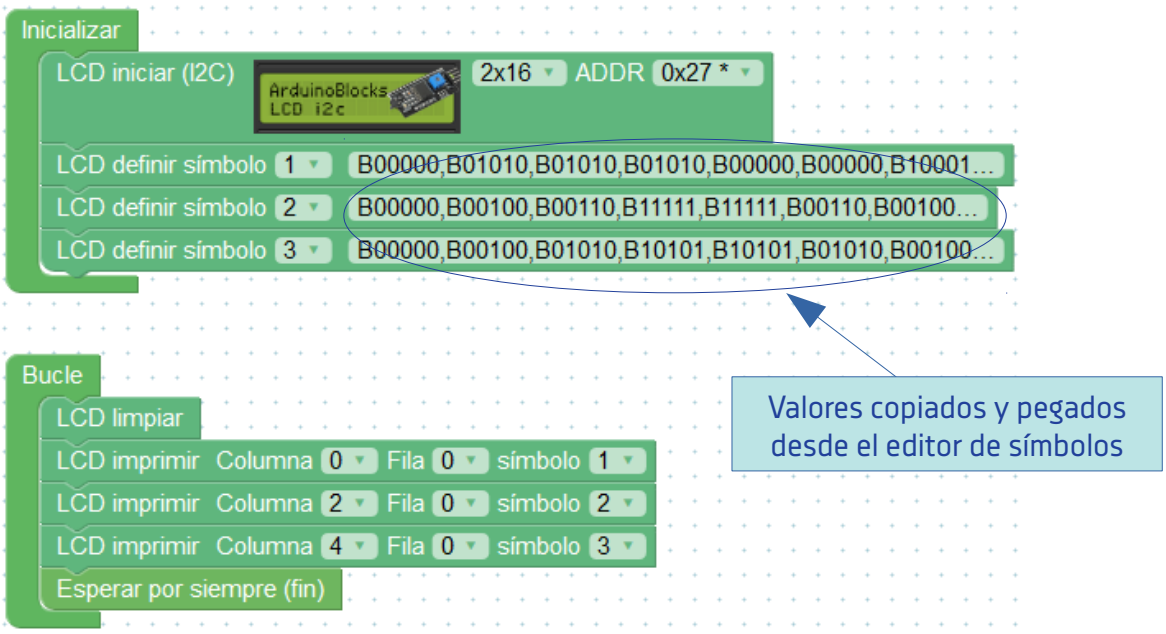
Definir símbolos personalizados / LCD (i2c)

CÓDIGO DE PROYECTO:

Gracias al editor de mapas de bits de ArduinoBlocks podemos definir fácilmente un nuevo símbolo para mostrar en el LCD. La pantalla LCD permite personalizar hasta 8 símbolos.

http://www.arduinoblocks.com/web/help/chareditor





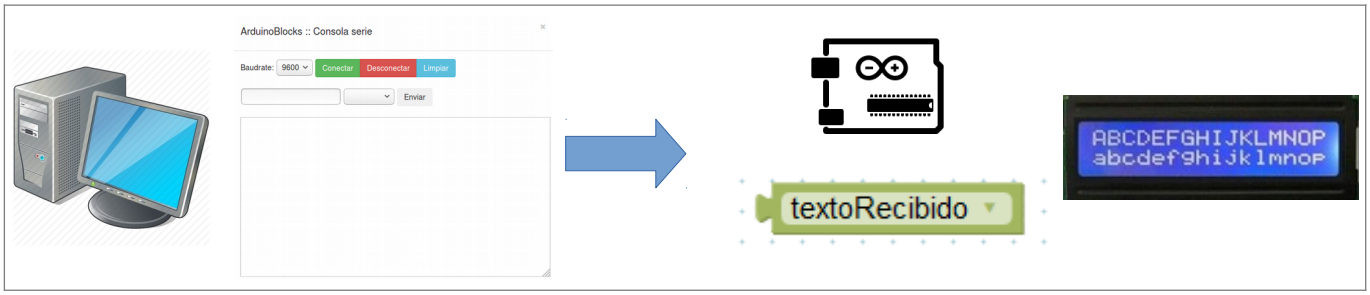
Pantalla LCD - 7

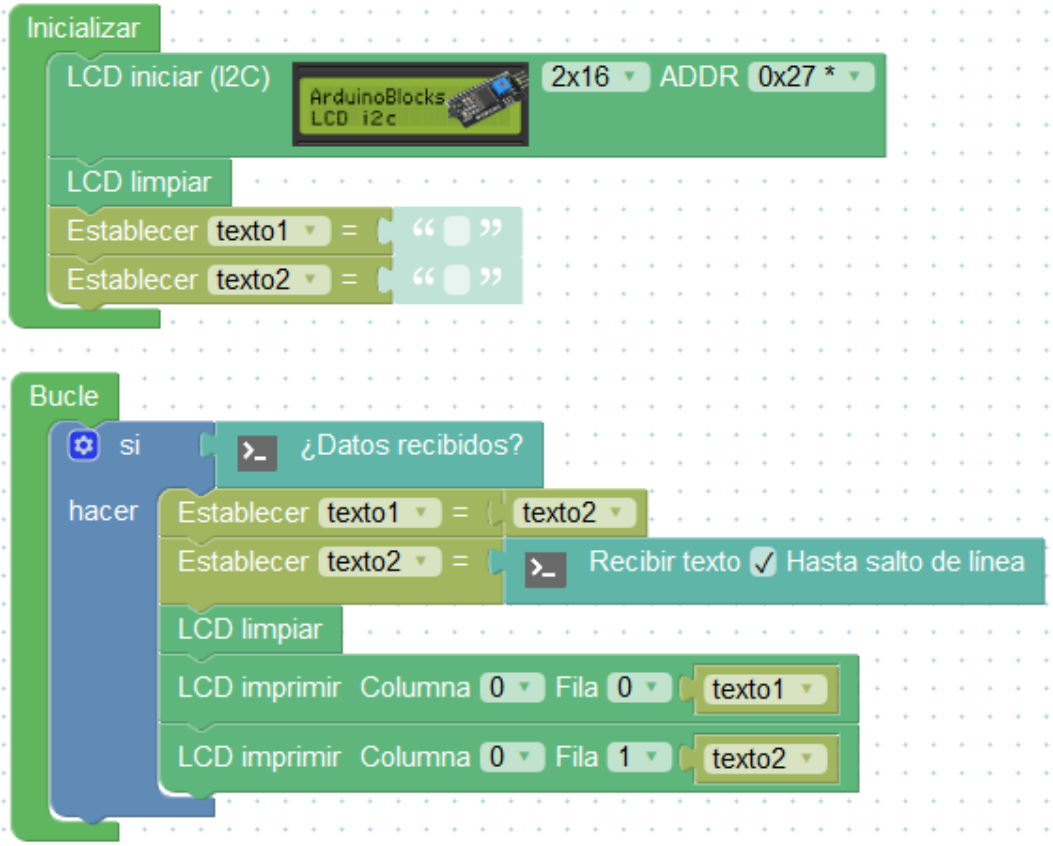
Enviar texto desde consola / LCD (i2c)

CÓDIGO DE PROYECTO:

El siguiente programa recibe textos a través de la conexión serie. El texto recibido lo inserta en la línea inferior del LCD (fila 1). El texto que había anteriormente en la fila 1 sube a la fila superior (fila 0).

Debemos tener la precaución de no enviar textos de más de 16 caracteres de longitud para que así quepan correctamente en la pantalla LCD.



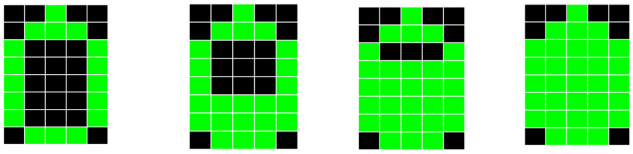


Pantalla LCD - 8 Animación de carga de batería / LCD (i2c) CÓDIGO DE PROYECTO:

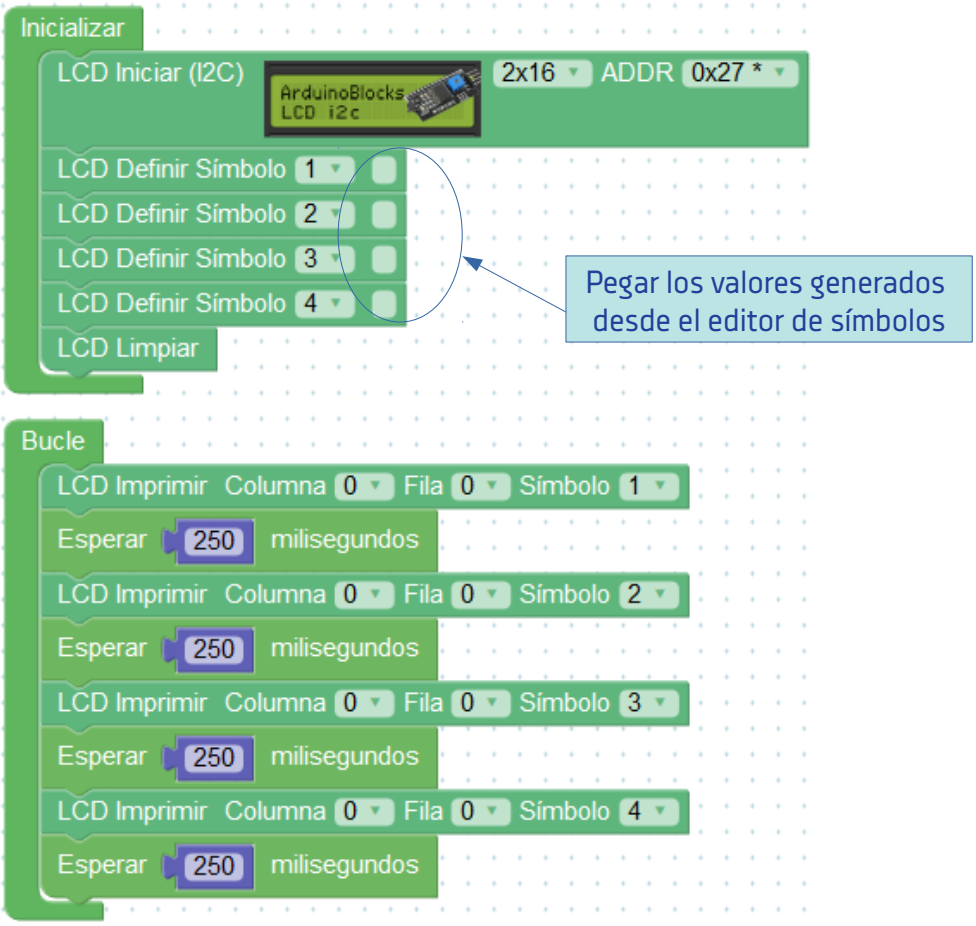
Definiendo 4 símbolos personalizados realizar la animación simulando la carga de una batería.

Ejemplo de animación:

Ejemplo de símbolos definidos para LCD:



Programa (completar la definición de los símbolos desde el editor de símbolos):



¿Te atreves a diseñar estos símbolos para la pantalla LCD?
WiFi, Altavoz, Bluetooth, Nube, Whatsapp, E-Mail, Home, ...

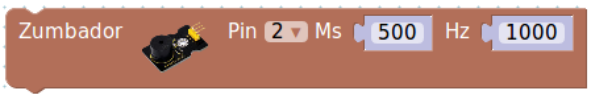
Zumbador

Un zumbador pasivo permite reproducir tonos con diferentes frecuencias (el sistema utilizado por ArduinoBlocks es similar al PWM pero por software). De esta forma podemos generar señales acústicas con el tono deseado y combinando duración, frecuencias y pausas podremos reproducir sencillas melodías.

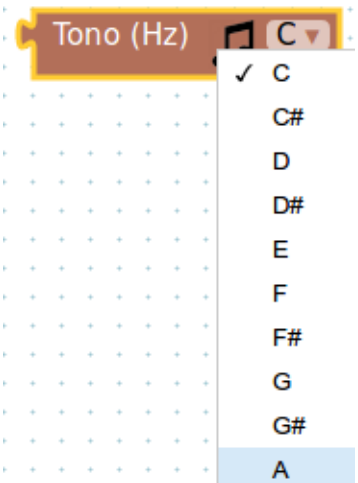
Zumbador pasivo

Módulo de zumbador listo para conectar

Para controlar el zumbador utilizaremos estos bloques:



Reproduce el tono con frecuencia de 1000Hz en el zumbador conectado en el pin 2 durante 500 ms



Con este bloque podemos seleccionar la frecuencia correspondiente a las notas musicales, en lugar de indicar los Hz de forma numérica directamente.

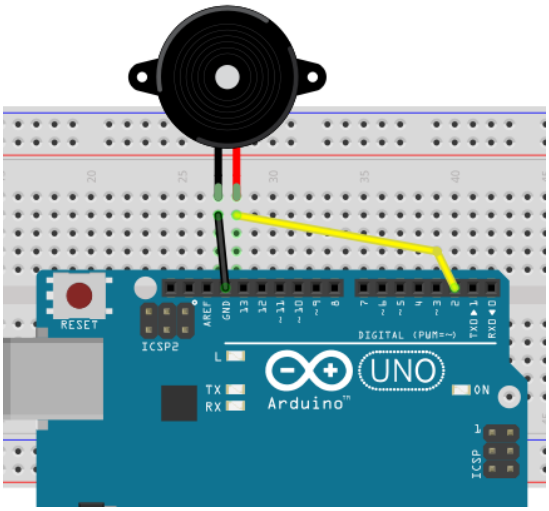
Nota	Nombre	Frecuencia
B4	SI	493.88
A4#	LA#	466.16
A4	LA	440.00
G4#	SOL#	415.30
G4	SOL	392.00
F4#	FA#	369.99

Zumbador - 1

Escala musical

CÓDIGO DE PROYECTO:

Reproducir escala musical



Zumbador - 2

Frecuencia progresiva

CÓDIGO DE PROYECTO:

Aumento de frecuencia progresivamente desde 40Hz hasta 2000Hz

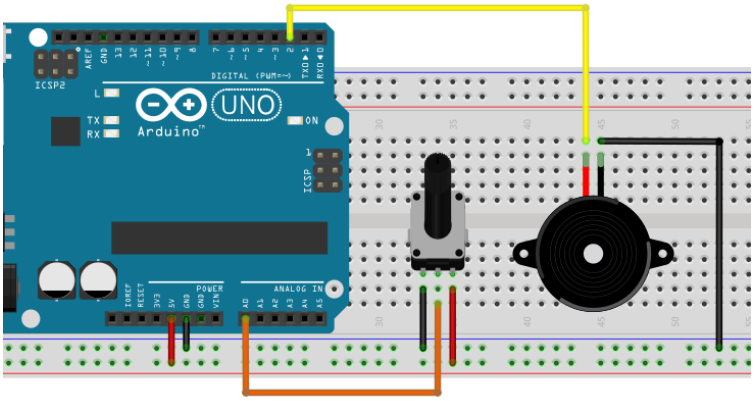
Modifica el programa anterior para que empiece con un tono de 10kHz y vaya bajando hasta 1Khz

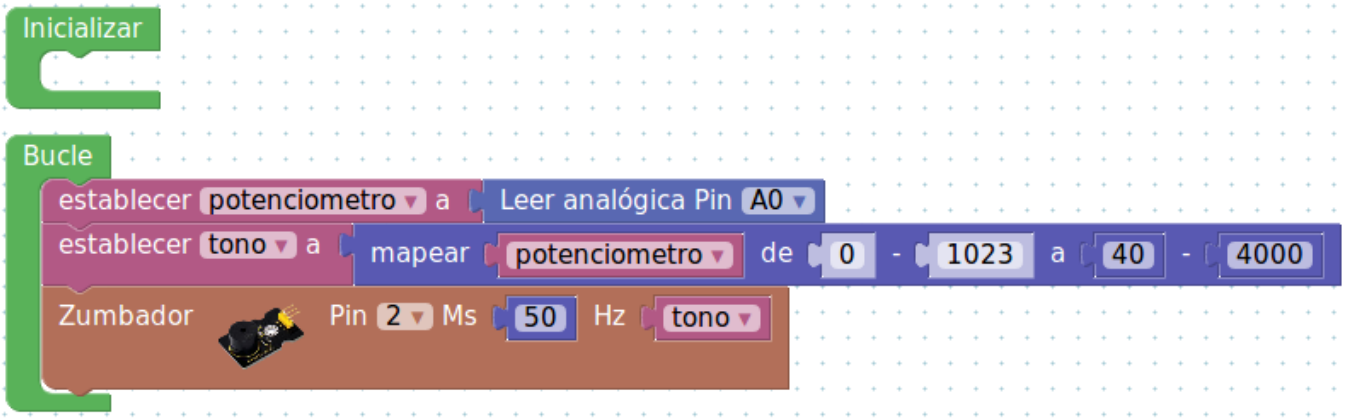
Zumbador - 3

Ajuste de tono con potenciómetro

CÓDIGO DE PROYECTO:

Con la ayuda de un potenciómetro conectado a una entrada analógica ajustaremos el valor de la frecuencia a reproducir en el zumbador entre 40 y 4000 Hz



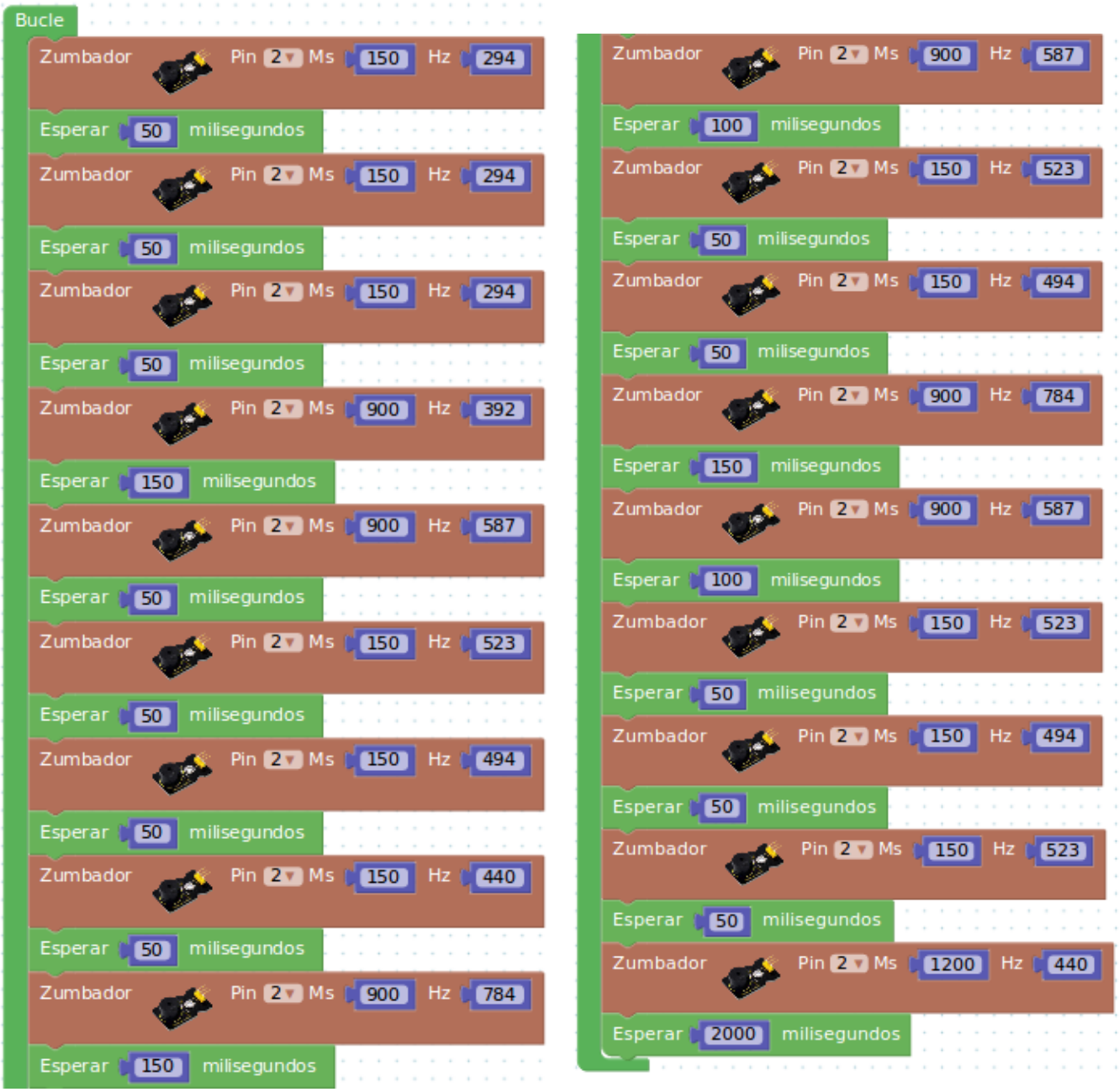


Modifica el programa para que el potenciómetro varíe el valor del tono entre 1000Hz y 2000Hz

Zumbador - 4

Melodía

CÓDIGO DE PROYECTO:



¿Reconoces la melodía?

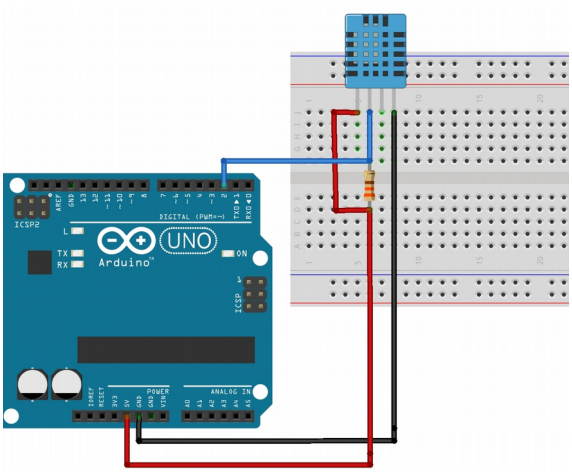
DHT11 - Sensor de temperatura/humedad

El DHT11 es un sensor de humedad/temperatura de bajo costo y de media precisión. Proporciona una salida de datos digital. Entre sus ventajas podemos mencionar el bajo coste y el despliegue de datos digitales. Entre las desventajas el DHT11 solo lee valores enteros, por lo que no podemos leer temperaturas con decimales.

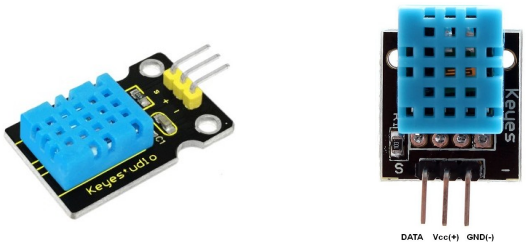
Rango de temperatura: de 0 a 50º (resolución 1ºC)
Rango de humedad: de 20 a 90% (resolución 1%)

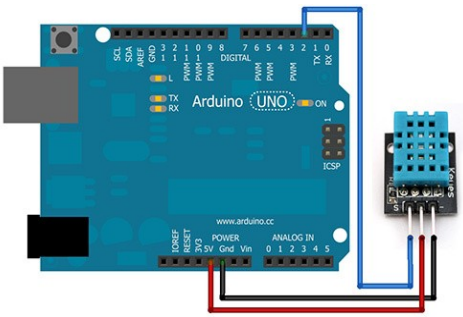
El pin GND y el VCC del sensor se conectan en sus correspondientes pines en Arduino (GND y +5V). El pin "data" se conecta a un pin digital para leerlo (si no es un módulo hay que añadir la resistencia de 10K)

Si queremos usar el sensor directamente debemos añadir una resistencia

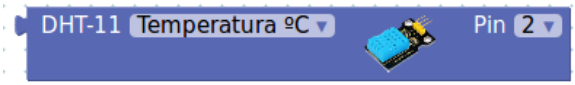


Módulos listos para conectar

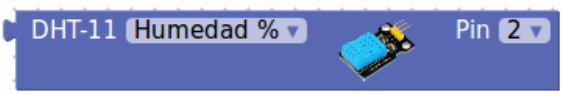




Para obtener el valor de temperatura o humedad utilizaremos los siguiente bloques:



Obtiene la temperatura (ºC) del sensor.



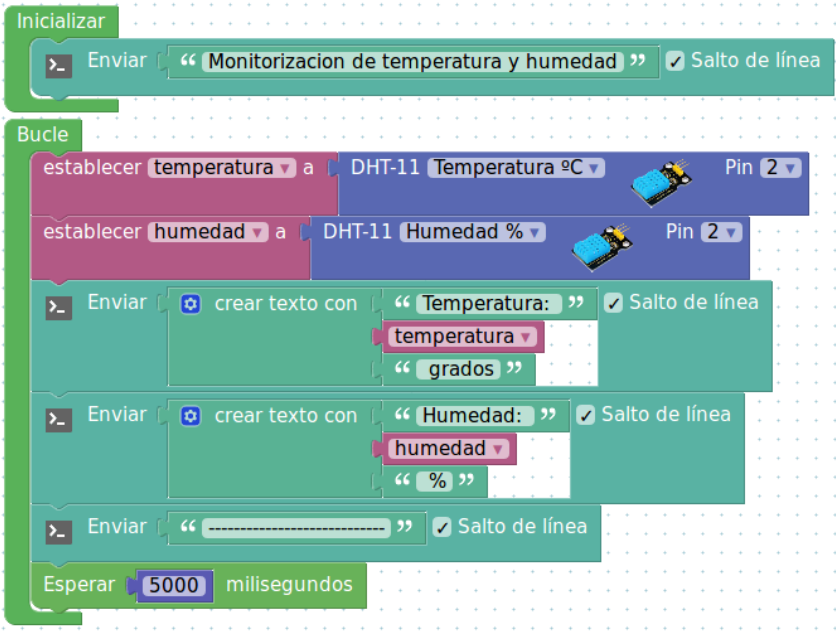
Obtiene la humedad relativa del aire (%) del sensor

DHT11 - 1

Mostrar temperatura y humedad por consola serie

CÓDIGO DE PROYECTO:

Conectando el sensor DHT-11 al pin 2 enviaremos el valor de temperatura y humedad cada 5s por la conexión serie para poder visualizarla desde la consola serie de ArduinoBlocks.

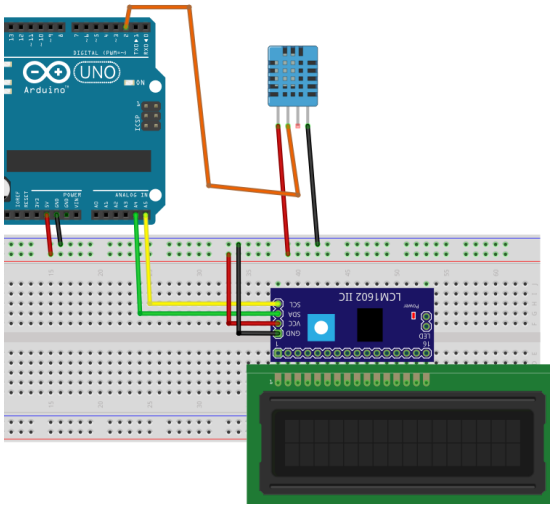


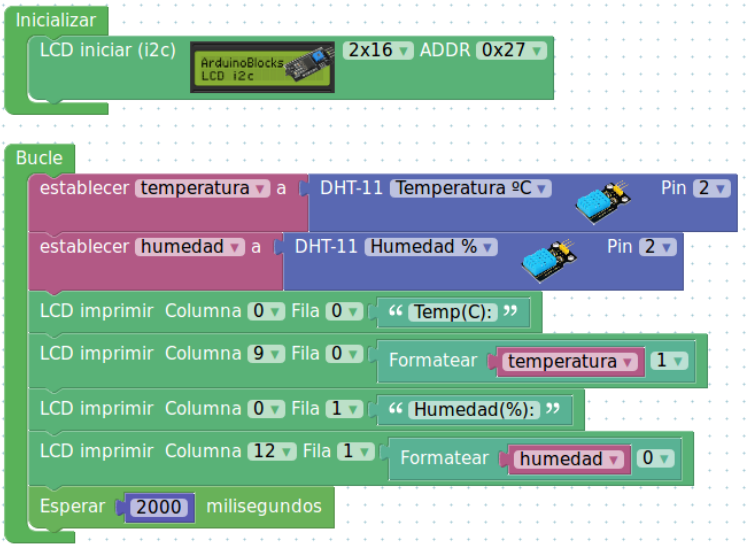
DHT11 - 2

Mostrar temperatura y humedad en pantalla LCD (I2C)

CÓDIGO DE PROYECTO:

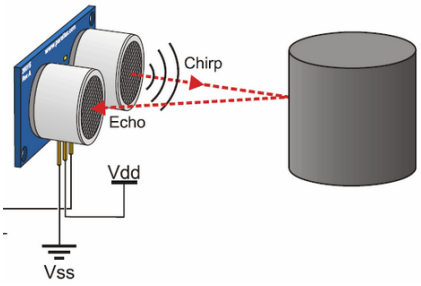
Conectando el sensor DHT-11 al pin 2 mostraremos el valor de temperatura y humedad en una pantalla LCD con conexión i2c.

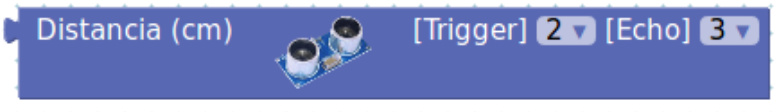




HC SR04 - Sensor de distancia por ultrasonidos

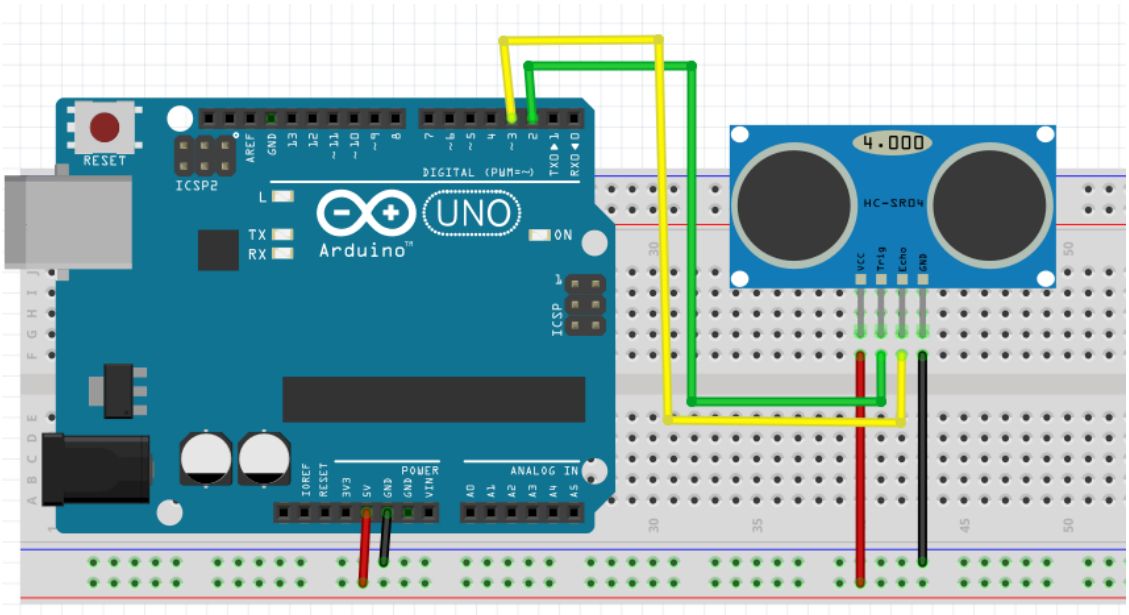
El sensor HC-SR04 permite detectar si hay objetos u obstáculos cercanos obteniendo la distancia a la que se encuentran. Su funcionamiento se basa en un emisor y receptor de ultrasonidos. El emisor emite una secuencia de impulsos de ultrasonidos y mide el tiempo hasta que el sensor los recibe (porque rebotan contra un objeto). Con el tiempo medido se calcula la distancia a la que se encuentra el objeto (si no hay ningún objeto, no se produce el "eco" y no se detecta nada)





Obtiene la distancia medida por el sensor. Si no se detecta ningún objeto devolverá valor 0.

Ejemplo de conexión del sensor de distancia por ultrasonidos HC-SR04 a Arduino:

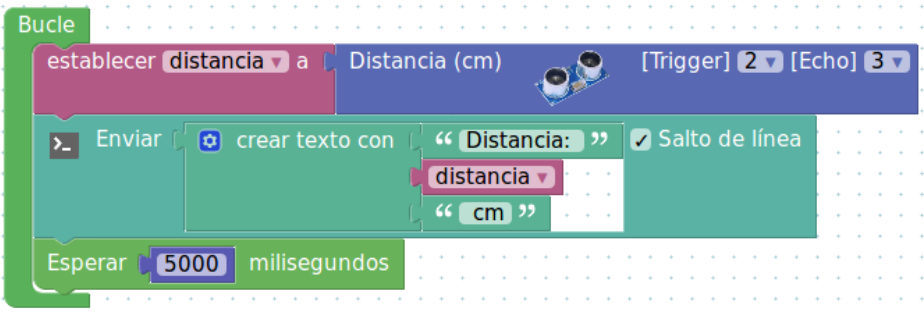


HC SR04 - 1

Mostrar distancia medida por la consola serie

CÓDIGO DE PROYECTO:

Conectar el sensor HC-SR04 al pin 2 (trigger) y 3 (echo) de Arduino.
Envía cada 5 segundos la distancia medida por el sensor por la conexión serie.

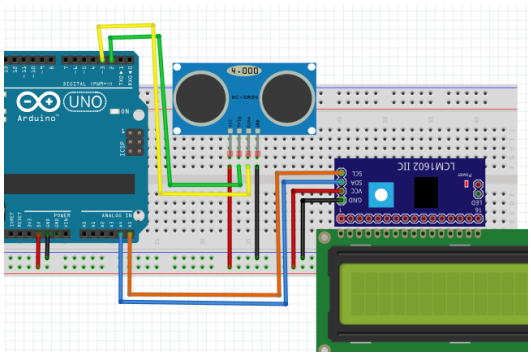


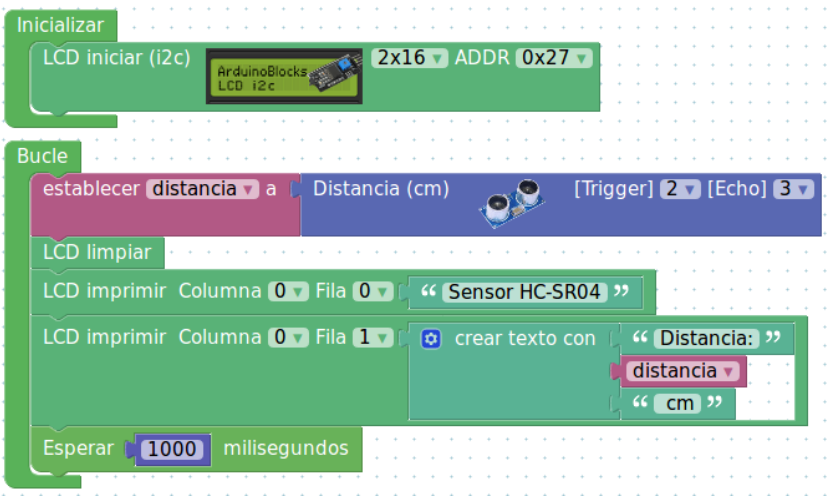
HC SR04 - 2

Mostrar distancia en pantalla LCD (I2C)

CÓDIGO DE PROYECTO:

Conectar el sensor HC-SR04 al pin 2 (trigger) y 3 (echo) de Arduino. Conectar la pantalla LCD con conexión i2c. Mostrar por la pantalla LCD la distancia detectada:

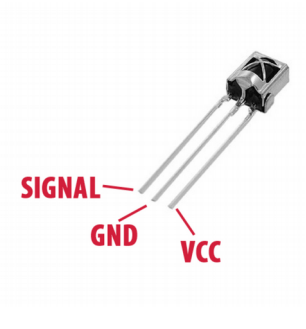




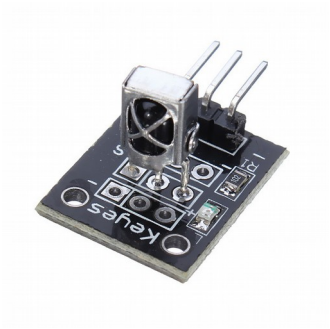
Receptor IR

La radiación infrarroja o IR es un tipo de radiación electromagnética y térmica, de mayor longitud de onda que la luz visible, pero menos que la de las microonda. El ojo humano no es capaz de ver esta longitud de onda, por lo que los IR son invisibles para nosotros.

El sensor AX-1838HS, permite recibir una señal infrarroja y convertirla en pulsos de señales eléctricas que en Arduino con el apropiado software podemos decodificar e interpretar.



Sensor AX-1838HS



Módulo listo para conectar

Mando IR genérico

Podemos reutilizar viejos mandos en nuestros proyectos

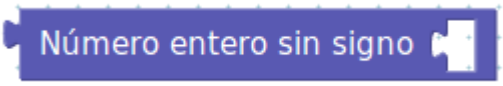
Obtiene el código recibido por el sensor IR. Este valor depende del tipo de mando IR utilizado.

ArduinoBlocks realiza el proceso de decodificación automáticamente, pero debemos saber que no todos los mandos de control remoto de IR utilizan el mismo tipo de códigos.

Los protocolos soportados son: *RC5, RC6, NEC, Panasonic, Sony, JVC, Samsung, Whynter, Aiwa, LG, Sanyo, Mitsubishi,Denon y Pronto* (muchas marcas reutilizan protocolos de otras marcas, con los protocolos anteriores decodificaremos prácticamente cualquier mando de control IR)

Si no se recibe ningún código válido el bloque devuelve valor “0”

Normalmente el valor recibido por el bloque IR lo almacenaremos en una variable. El valor recibido es un valor de 32 bits sin signo, por lo que lo recomendable a la hora de tratar el valor es ajustarlo con el bloque “Número entero sin signo”

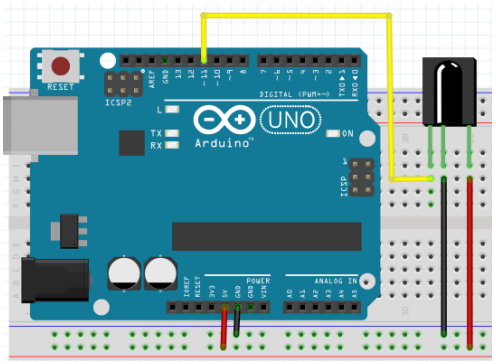


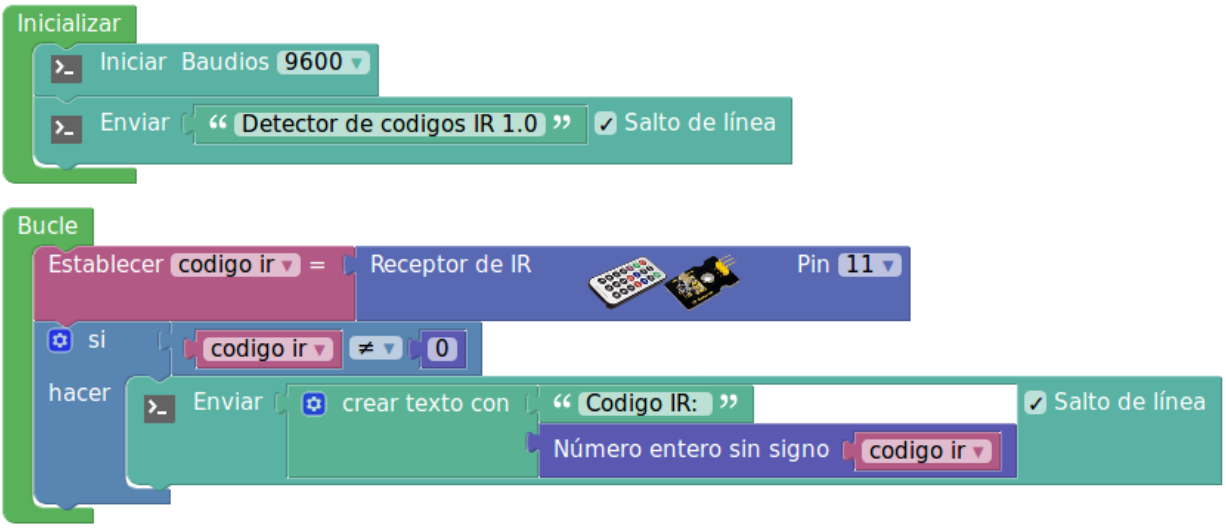
Receptor IR - 1

Mostrar códigos recibidos por consola serie

CÓDIGO DE PROYECTO:

Conectamos el sensor IR al pin 11 de Arduino.
Los códigos recibidos se mostrarán en la consola serie de ArduinoBlocks.
Prueba con distintos mandos a distancia para ver los códigos recibidos.





Apunta los códigos obtenidos:

Mando/Tecla	*Código*

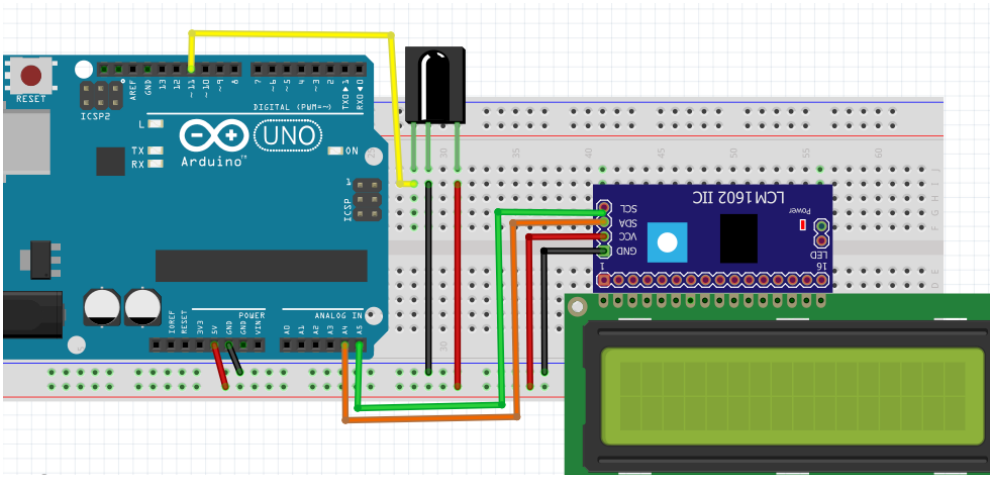
Mando/Tecla	*Código*

Receptor IR - 2

Mostrar código IR recibido en pantalla LCD

CÓDIGO DE PROYECTO:

Con la ayuda de una pantalla LCD conectada por I2C mostraremos los códigos de cualquier mando IR compatible. Este montaje es de gran utilidad para detectar los códigos de cada tecla de los mandos IR para utilizarlos en proyectos posteriores.



Receptor IR - 3

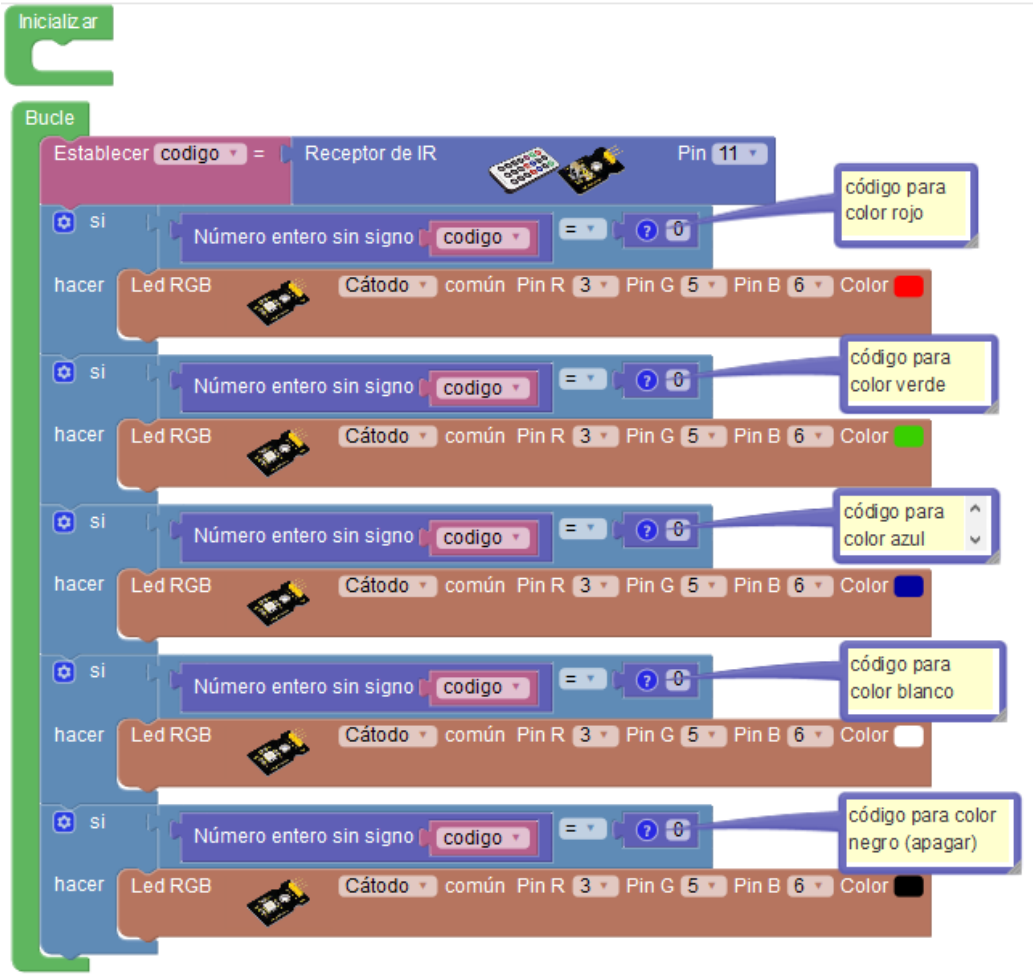
Control de led RGB con mando IR

CÓDIGO DE PROYECTO:

Con la ayuda del *programa-1* o el *programa-2* apunta el código correspondiente a 5 botones de un mando a distancia.

Botón	Código	Función a realizar
		Apagar el led
		Poner led en rojo
		Poner led en verde
		Poner led en azul
		Poner led en blanco

Conecta un led RGB a los pines 3, 5 y 6
Realiza este programa con los códigos capturados y anotados en la tabla anteriores



Receptor IR - 4

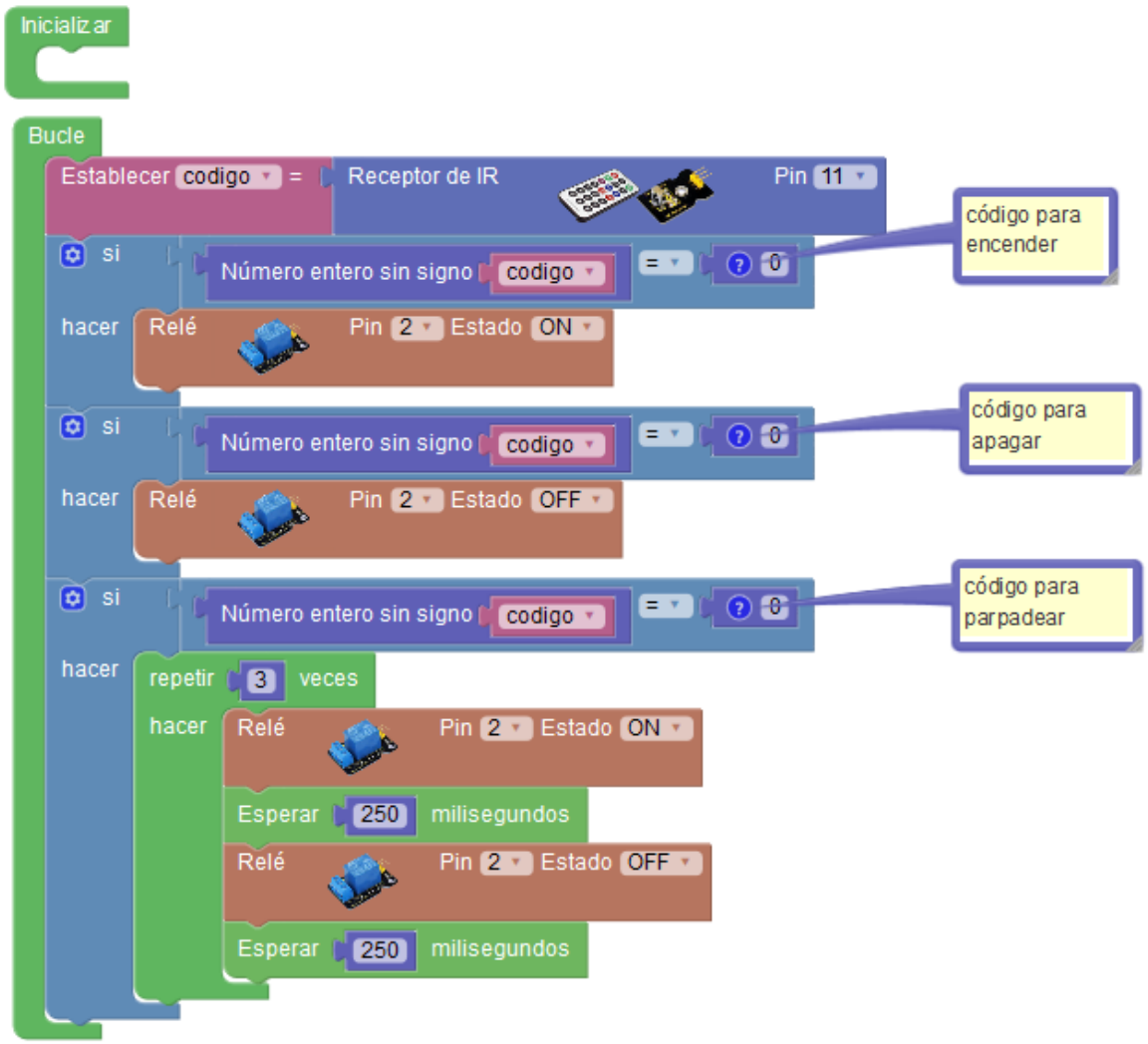
Control de relé con mando IR

CÓDIGO DE PROYECTO:

Con la ayuda del *programa-1* o el *programa-2* apunta el código correspondiente a 3 botones de un mando a distancia.

Botón	Código	Función a realizar
		Apagar relé
		Encender relé
		Encender y apagar relé 3 veces

Conecta el módulo de relé al pin 2. Programa para controlar el relé:



Receptor IR - 5

Control de servo con mando IR

CÓDIGO DE PROYECTO:

Conectar un servomotor al pin 3.
Con un mando IR situar al servo en diferentes posiciones:

Código tecla mando IR	Posición servo
	0º
	45º
	90º
	135º
	180º

Receptor IR - 6

Notas musicales con mando IR

CÓDIGO DE PROYECTO:

Conectar un zumbador al pin 5.
Con cada tecla de un mando a distancia IR haremos que suene una nota musical diferente durante 1s.

Código tecla mando IR	Nota musical
	C
	C#
	D
	D#
	E
	F
	F#
	G
	G#
	A

Tareas

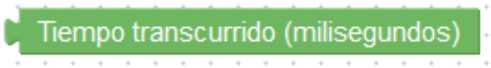
Arduino no posee un sistema operativo y tampoco incorpora ningún otro sistema que gestione la multitarea como estamos acostumbrados en otros entornos de programación (por ejemplo en Scratch), el programa se ejecuta directamente por el microcontrolador y somos nosotros en el propio programa los que tenemos que gestionar la manera de ejecutar varias tareas simultaneamente intentando simular un sistema multitarea.

La teoría de un sistema multitarea es dividir los procesos en pequeños bloques, cada uno de estos bloques debe realizar un pequeño trabajo (en poco tiempo) y dejar paso al siguiente sin bloquear el funcionamiento. Al realizar este proceso de forma continua y a mucha velocida el resultado final es que todos los bloques se ejecutan simultáneamente.

¿Qué debemos evitar dentro de los bloques de las tareas?

- Bloques de tipo “esperar”
- Bucles o repeticiones muy largas
- Condiciones que bloqueen la ejecución de la tarea

Para implementar un sistema sencillo de tareas utilizaremos el bloque “tiempo transcurrido”. Este bloque nos permite obtener el tiempo en ms que ha pasado desde el inicio del programa (reset).

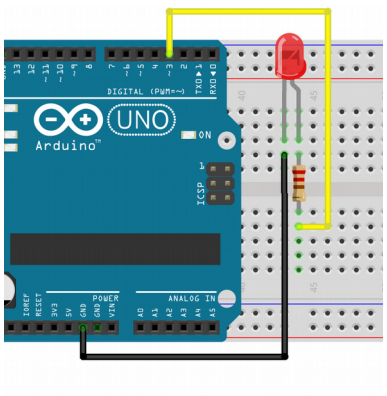


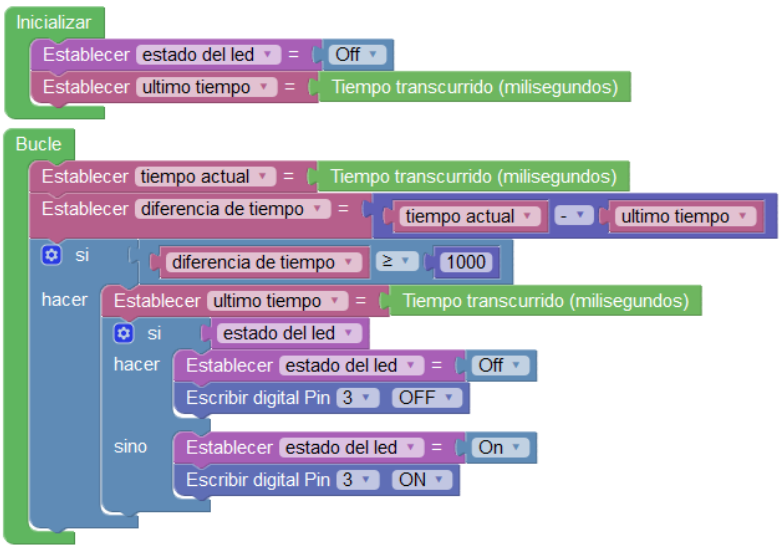
Tareas - 1

Parpadeo led cada 1s

CÓDIGO DE PROYECTO:

El programa comprobará si han pasado 1000 ms (o más) desde la última ejecución de la tarea, si es así ejecutará el código de encender/apagar el led.





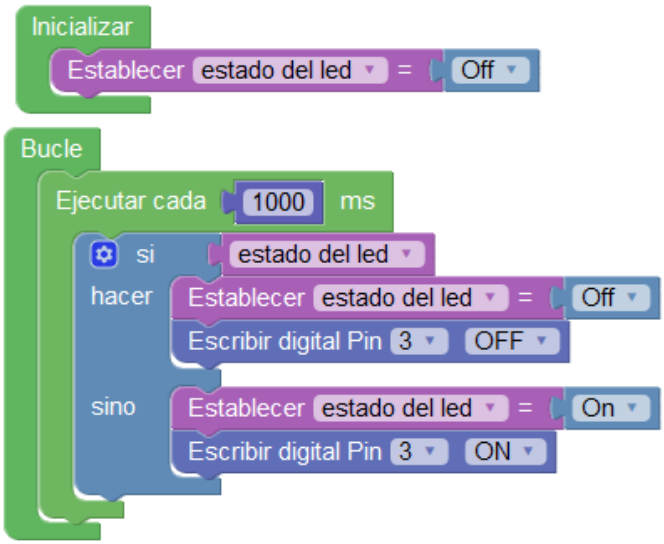
Tareas - 2

Bloque de tareas simplificado

CÓDIGO DE PROYECTO:

Todo el programa anterior de la práctica *Tareas-1* se puede simplificar con un bloque que incorpora ArduinoBlocks llamado "*ejecutar cada*", donde le especificamos cada cuanto tiempo se deben ejecutar los bloques contenidos en su interior.

Programa equivalente a la práctica *Tareas-1*



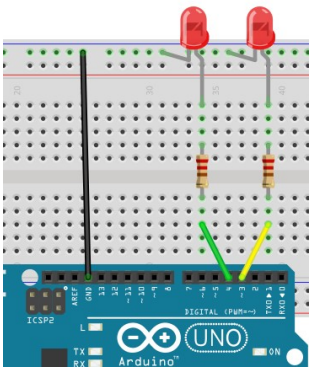
Tareas - 3

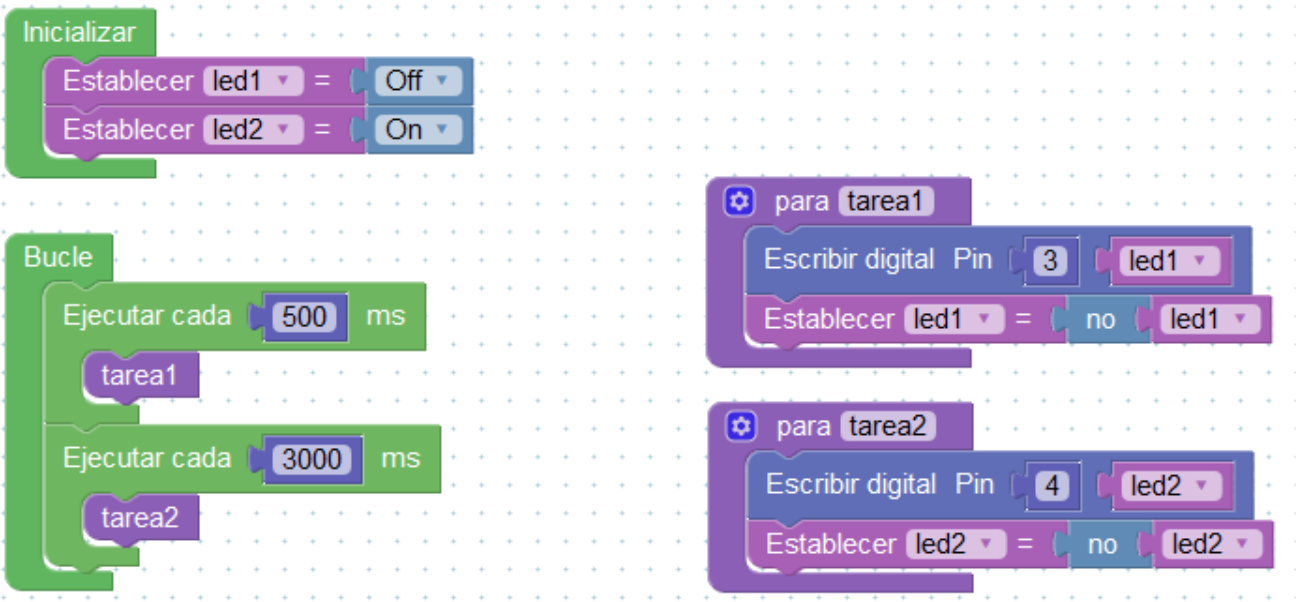
Dos tareas a la vez

CÓDIGO DE PROYECTO:

Utilizando bloques "ejectura cada" vamos a realizar dos tareas simultáneas

- Tarea 1 (cada 500ms): hará parpadear un led conectado al pin 3 cada 500 ms
- Tarea 2 (cada 3000ms): hará parpadear un led conectado al pin 4 cada 3000 ms



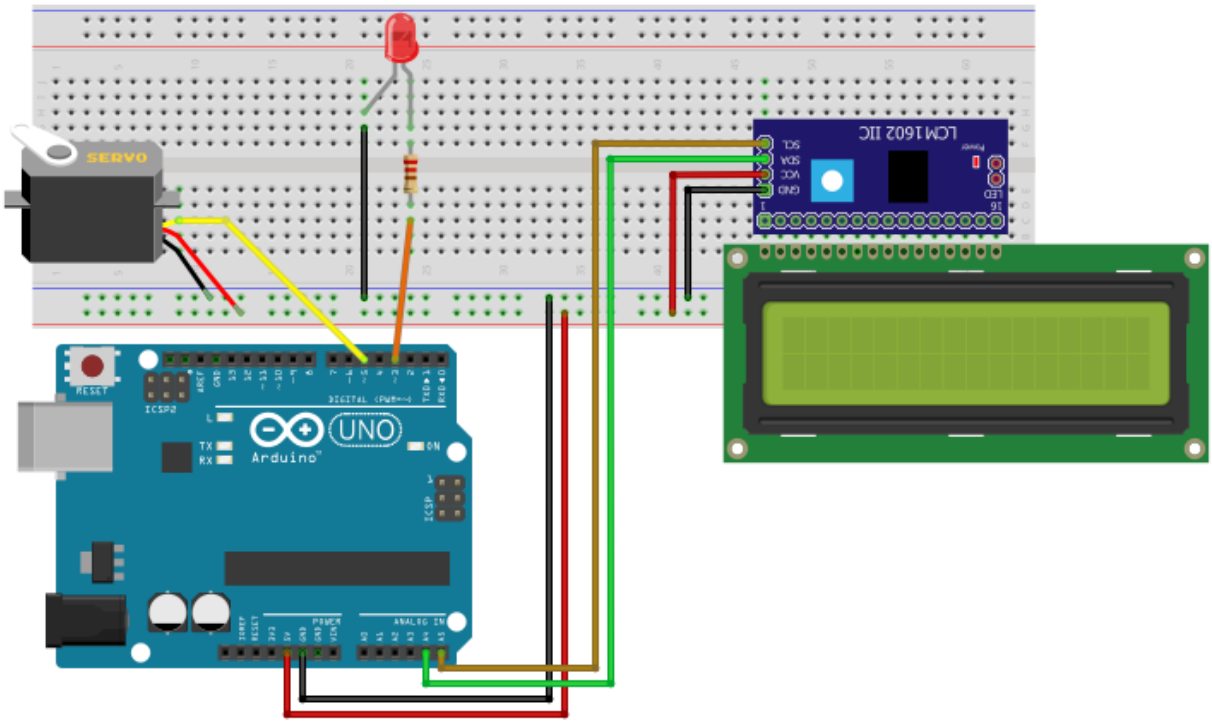


Tareas - 4

Tres tareas a la vez

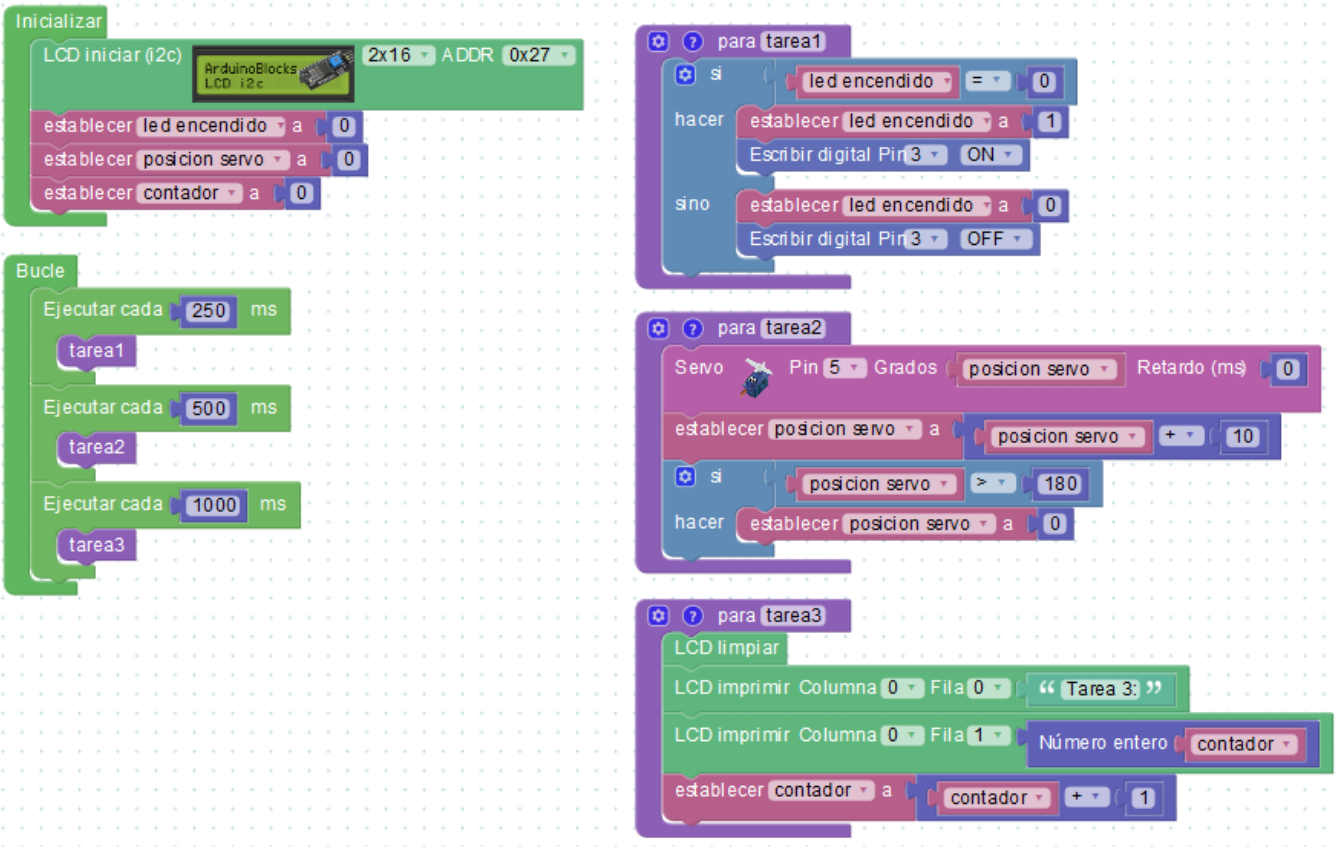
CÓDIGO DE PROYECTO:

Conecta la pantalla LCD con el módulo i2c, un led al pin 3 y un servo al pin 5.



El programa debe realizar 3 tareas:

- Tarea 1: Cada 250 ms parpadea el led
- Tarea 2: Cada 500 ms el servo se mueve 30 grados
- Tarea 3: Cada 1000 ms en la pantalla se muestra el valor de un variable que va aumentando

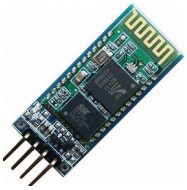


Modifica el tiempo de cada tarea en el programa anterior...

Bluetooth: Comunicación con módulo Bluetooth HC-06

El sistema Bluetooth permite comunicaciones inalámbricas entre dispositivos a una distancia máxima de unos 100m (según la potencia del módulo utilizado).

El módulo Bluetooth HC-06 que vamos a utilizar permite "simular" una conexión serie estándar a través del protocolo inalámbrico Bluetooth de una forma muy sencilla (protocolo RFCOMM/SPP).

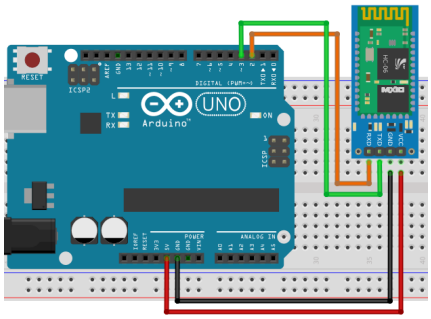


La conexión con la placa Arduino utiliza 2 pines, uno para RX y otro para TX. ArduinoBlocks implementa un puerto serie por software para comunicar con el módulo Bluetooth y así no interferir con el puerto serie integrado en Arduino UNO (pines 0 y 1) utilizado también para la programación del microcontrolador de Arduino.

Para conectar con el módulo Bluetooth necesitaremos un dispositivo con conexión Bluetooth (smartphone, tablet o pc con conexión bluetooth) y una aplicación de consola/terminal serie.

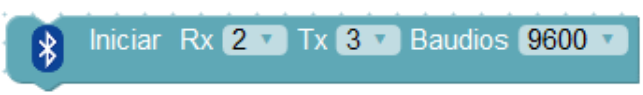
En Android podemos encontrar aplicaciones como: "Bluetooth Terminal", "BlueTerm", "BlueTerm2", ...

Ejemplo de conexión a los pines 2,3

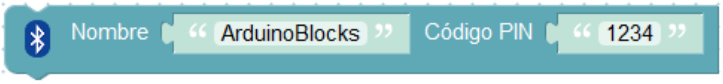


Consola serie bluetooth "BlueTerm" en dispositivo móvil Android

Los bloques para uso de la comunicación Bluetooth son exactamente iguales a los bloques de comunicación serie vistos en la práctica 6. Además hay dos bloques para inicializar y ajustar la configuración interna del módulo HC-06 (nombre y pin):



Configura la conexión con el módulo Bluetooth y la velocidad de la conexión.



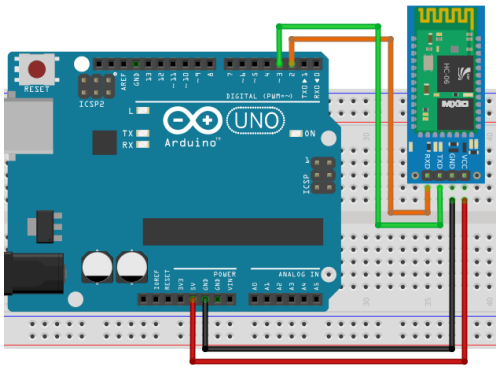
Configura el módulo internamente para fijarle el nombre y el código de emparejamiento.

Bluetooth - 1

Envío de mensajes via Bluetooth

CÓDIGO DE PROYECTO:

El programa enviará varios mensajes mediante el módulo Bluetooth y los visualizaremos en una aplicación de consola/terminal Bluetooth en un dispositivo móvil (smartphone o tablet)



Aplicación BlueTerm (Android)

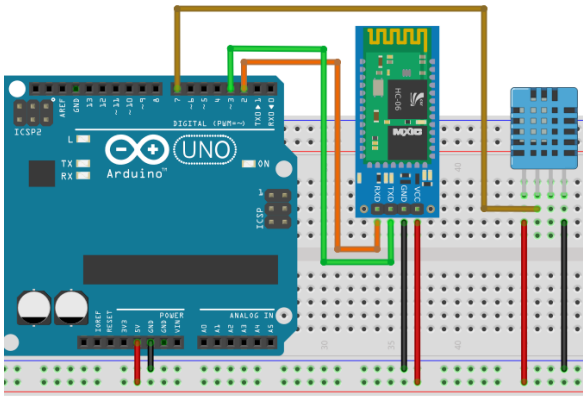
Existen muchas otras aplicaciones de terminal Bluetooth para dispositivos Android: Bluetooth Terminal, Arduino bluetooth controller, BlueTerm 2, ...

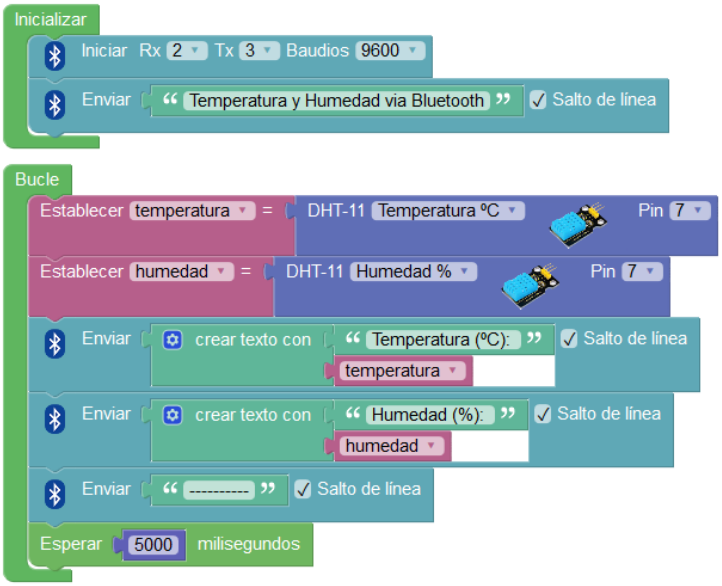
Bluetooth - 2

Envío de temperatura y humedad via Bluetooth

CÓDIGO DE PROYECTO:

Leeremos la temperatura y humedad de un sensor DHT11.
Enviaremos la información de temperatura y humedad cada 5 segundos para poder visualizarla remotamente en el terminal Bluetooth de un smartphone.





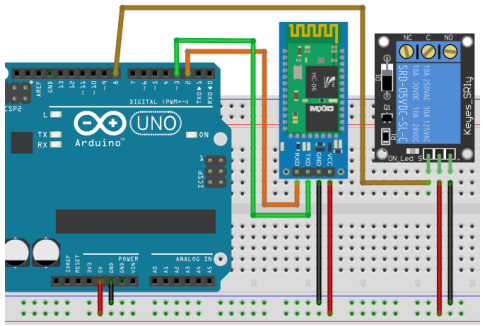
Bluetooth - 3

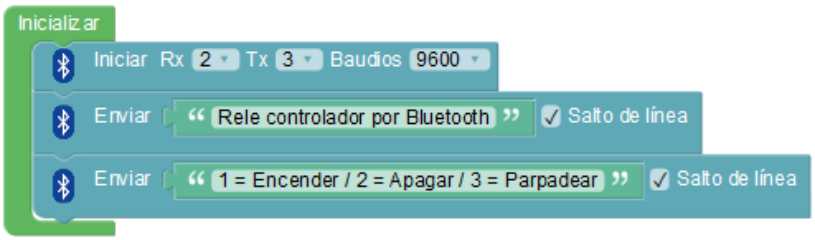
Control de relé desde el móvil

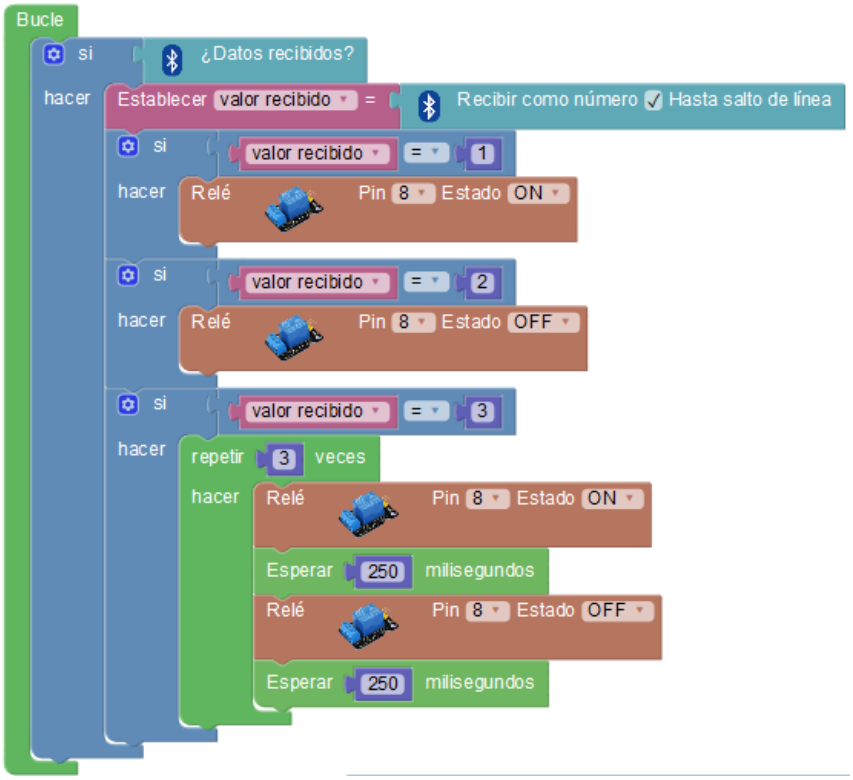
CÓDIGO DE PROYECTO:

Conectaremos un relé en el pin 8 y el módulo Bluetooth como anteriormente a los pines 2,3.
En este caso enviaremos un valor numérico desde la consola serie bluetooth del smartphone.

Si la placa Arduino recibe un valor "1" enciende el relé, un "2" apaga el relé, un "3" intermitente.







Bluetooth - 4

Control desde móvil por voz (Android)

CÓDIGO DE PROYECTO:

Con el mismo montaje de la práctica anterior y utilizando una aplicación específica realizaremos el control por voz de la práctica anterior.

El funcionamiento es exactamente el mismo, cuando reconoce cada palabra envía un un valor numérico a través de la conexión Bluetooth:

“encender”	Envía valor 1
“apagar”	Envía valor 2
“parpadear”	Envía valor 3

www.arduinoblocks.com/web/apk/ArduinoBlocks_ControlVoz.apk