

# Table of Contents

Programa arduino con Echidna	1.1
Tema 1 ¿Cómo utilizar Echidna?	1.2
1.1 ¿Qué es EchidnaShield?	1.2.1
1.2 Cómo se programa Echidna Shield	1.2.2
1.2.1 MONTAJE 1 SEMÁFORO con mBlock	1.2.2.1
1.2.2 MONTAJE 1 SEMÁFORO con Snap4Arduino	1.2.2.2
1.2.3 MONTAJE 1 SEMÁFORO mediante código	1.2.2.3
1.2.4 mBlock vs Snap4Arduino ¿cuál es el mejor?	1.2.2.4
1.2.4.1 Instrucciones específicas para Arduino	1.2.2.4.1
1.2.4.2 Importación de librerías	1.2.2.4.2
1.2.4.3 Formas de conexión con Arduino	1.2.2.4.3
1.2.4.4 Reutilizar proyectos desde Scratch	1.2.2.4.4
1.2.4.5 Entorno amigable	1.2.2.4.5
1.2.4.6 Guardar en la nube y compartir	1.2.2.4.6
1.2.4.7 IMPORTANTE Subir a Arduino	1.2.2.4.7
1.2.4.8 And winner is ...	1.2.2.4.8
1.2.5 Otros	1.2.2.5
1.3 Alimentación del Echidna	1.2.3
1.4 Hoja de Ruta	1.2.4
1.5 Kit de préstamo en CATEDU	1.2.5
Tema 2 Salidas de Echidna	1.3
2.1 MONTAJE 2 Luces psicodelicas	1.3.1
2.2 MONTAJE 3 Timbre	1.3.2
3 Entradas de Echidna	1.4
3.1 MONTAJE 4 Interruptor luz	1.4.1
3.2 El LDR en A5	1.4.2
3.2.1 Comprobar los límites	1.4.2.1
3.2.2 MONTAJE 5 Semáforo luminoso	1.4.2.2
3.2.3 MONTAJE 6 Piano luminoso	1.4.2.3
3.2.3.1 Cambios de escala	1.4.2.3.1
3.3 Joystick	1.4.3
3.3.1 MONTAJE 7 Telesketch	1.4.3.1
3.3.2 MONTAJE 8 Come bichos	1.4.3.2
3.4 Acelerómetro	1.4.4
3.4.1 MONTAJE 9 Helicóptero	1.4.4.1
3.4.2 MONTAJE 10 Mataliens	1.4.4.2
4 Modo Makey makey	1.5
4.1 ¿Qué es?	1.5.1
4.2 MONTAJE 11 Piano con bananas	1.5.2

---

4.3 MONTAJE 12 Joystick plastilina	1.5.3
5 Extensiones	1.6
5.1 MONTAJE 13 Encender con el móvil (muy difícil)	1.6.1
5.2 MONTAJE 14 Encendido sensible	1.6.2
5.3 MONTAJE 15 Alarma láser	1.6.3
5.4 MONTAJE 16 Semáforo distancia	1.6.4
5.5 Servo	1.6.5
5.6 MONTAJE 17 Tractor entrando al corral	1.6.6
Muro	1.7
Grupo Robotica Educativa Aragon	1.8
Créditos	1.9

---

# Programa arduino con Echidna

En este curso pretende mostrarte el potencial educativo de este sencillo escudo.

- **1. Cómo utilizarlo**, veremos que está adaptado a la programación en primaria o primeros cursos de secundaria con programación gráfica
- **2. Salidas** De forma fácil y sin electrónica ya podemos jugar
- **3. Entradas** El joystick, acelerómetro y el LDR son elementos motivadores para la programación y la imaginación.
- **4. Makey Makey** tienes dos placas en una. Se abre otra puerta a las capacidades de nuestros pequeños genios.
- **5. Extensiones** no nos quedamos cortos, podemos construir sencillos robots y proyectos con poco cableado.

Todos los programas de este curso se encuentran en este repositorio: <https://github.com/JavierQuintana/Echidna>



Fuente <http://echidna.es/>



Fuente <http://echidna.es/>

# CENTRO ARAGONÉS de TECNOLOGÍAS para la EDUCACIÓN



**CATEDU**

# Tema 1 ¿Cómo utilizar Echidna?

En este tema veremos:

- **1.1** **Qué es Echidna**
- **1.2** **Qué programa** vamos a utilizar para programarlo
- **1.3 Comparativa** de otros robots y propuestas para enseñar el pensamiento computacional en nuestras clases.

Todos los programas de este curso se encuentran en este repositorio: <https://github.com/JavierQuintana/Echidna>



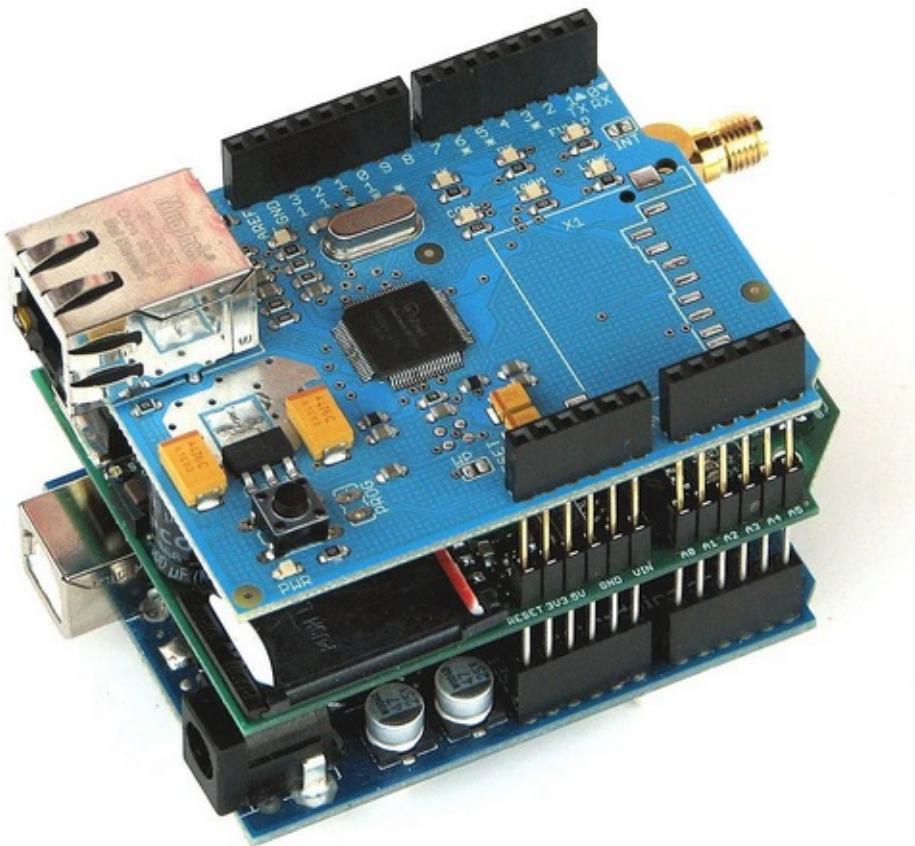
Source: <http://echidna.es/>

## 1.1.1 ¿Qué es EchidnaShield?

### 1.1.1.1 Primero ¿qué es Arduino? ¿qué es una Shield?

**¿Qué es Arduino?** Pues no podemos explicarlo en este curso, *suponemos que lo conoces*, te recomendamos que [visites nuestro curso Arduino con código y Edubasica en esta página](#), si la lees verás que Arduino tiene un inconveniente: Arduino es una placa microcontroladora orientado para las entradas y las salidas tanto analógicas como digitales, pero.... "**tienes que ponerlas**", evidentemente, y eso implica cableado y electrónica.

Una solución es utilizar una Shield, está adaptado para ponerlo encima e incluso hay posibilidad utilizar varias a la vez.



Enrique Crespo CC-BY-SA

Hay muchas en el mercado:

- Para las comunicaciones:
  - [EthernetShield](#)
  - [WifiShield](#)
  - [GSM Shield móvil GPRS](#)
- Para salidas concretas
  - [Motores](#)
  - [Relés RelayShield](#)
- Para entradas concretas
  - [OpenEnergy Medidor Medidor de magnitudes eléctricas](#)
- Para facilitar la conexión de sensores y actuadores
  - [Grove Shield](#) donde hay una **familia sensores y actuadores de conectar y listo** asociados, por cierto... veremos [en la unidad 5](#) que Echidna tiene 3 I/O digitales y una analógica para estos sensores, es decir, es también *una versión pequeña Grove Shield* que nos permitirá crear buenos proyectos .

Hay tantas Shields comerciales que la lista es interminable..., prueba poner las palabras *Shield Arduino* en tu buscador. Pero centrémonos, lo que nos interesa aquí es el punto educativo: Hay otras están orientadas para **el uso educativo**:

- [Edubásica](#) que en Aularagón tenemos un [curso dedicado](#) a esta Shield creada por docentes y para docentes.
- Hay otra muy básica pero muy barata [enlace1](#), [enlace2](#), [Tutorial de uso de Javier Fernandez Panadero](#).
- ¿Y cómo no?: nuestra estrella en este curso [Echinda](#)

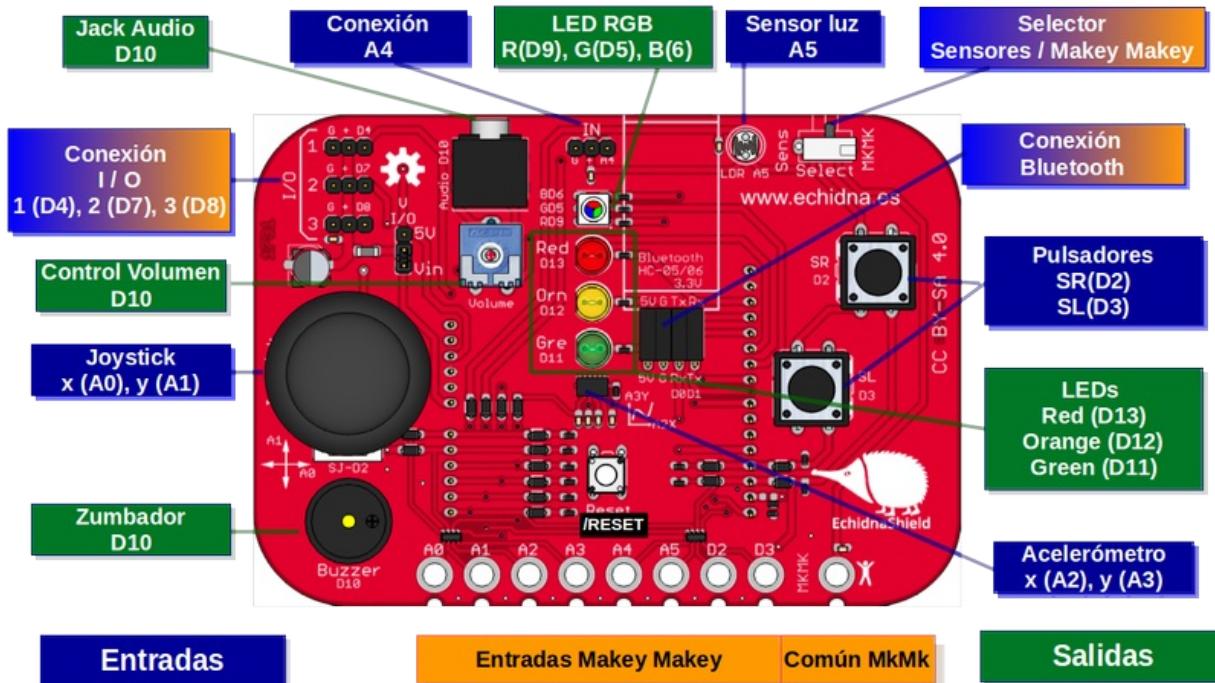
También hay que nombrar “*entrenadores*” educativos para Arduino, “Kits de principiantes”, etc.. los hay con muchos sensores y otros muy compactos, visita por curiosidad uno: [Evive](#), pues explican muy bien las ventajas de utilizar una Shield/entrenador frente a usar la electrónica pura.

### 1.1.2 Ahora sí: ¿Qué es Echidna?

Es una Shield de Arduino diseñada para facilitar la programación del Arduino en los últimos cursos de primaria y primeros de secundaria, pensado en minimizar el cableado de la electrónica (ya lo sabemos pues es una Shield) y enfocado a nivel educativo y al lenguaje por símbolos

*¿Por qué hemos elegido esta Shield?* Por que tiene un buen equilibrio calidad/precio sencillez/potencial orientado a últimos cursos de primaria, primeros de secundaria:

- Es **OpenHardware**, por lo tanto es un proyecto con garantías de estabilidad, libre, colaborativo y vivo, con la misma filosofía que la placa Arduino.



Source: Web oficial Echidna- documentación

- **Salidas acertadas:**
  - Diodos Red D13, Orange D12 y Green D11.
  - Diodo de 3 colores RGB gobernados por D9, D5 y D6 respectivamente.
  - Audio en D10.
- **Entradas o sensores acertados:**
  - Joystick (estupendo para hacer proyectos atractivos) xy con A0 y A1
  - Acelerómetro (idem) xy en A2 y A3
  - Luz LDR en A5
  - Botones digitales D2 y D3
- **Es MakeyMakey:** ¡Dos Shields en una!: Sensores y [MakeyMakey](#) por lo tanto da más potencial a nuestros proyectos.(Tenemos un [curso en Aularagon de MakeyMakey](#))

- **Conexión a periféricos:**

- Bluetooth que da más potencial a nuestros proyectos.
- Tiene pines para conexión de otros elementos como relés pero no pueden pasar de 300mA para más potencia es mejor utilizar Shield adaptados para ello como Edubásica.

**¿Estás preparado? adelante !!!**

## 1.2 Cómo se programa Echidna Shield

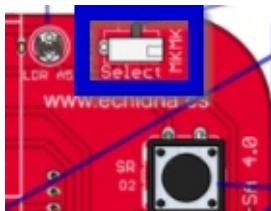
Tenemos dos opciones:

- Programación mediante **lenguajes gráficos**: símbolos, gráficos... tipo [Scratch](#)
  - **Ventajas**: Mucho más sencillo e intuitivo, ideal para principiantes en programación como es nuestro caso entre primaria y secundaria.
  - **Inconvenientes**: No se llega a aprovechar todas las posibilidades del Arduino.
  - **Programas**: [Snap4Arduino](#), [mBlock](#), [ArduinoBlocks](#), [Bitbloq](#), [S4A](#)...
- Programación mediante **código**:
  - **Ventaja**: Se aprovecha todo el potencial de programación, puesto que se controla todas las variables
  - **Inconvenientes**: Como toda programación en código, tiene su dificultad y abstracción.
  - **Programas**: [ArduinoIDE](#).

Como este curso trata del Echidna, que es una shield sencilla, nos decantamos por la programación mediante lenguajes gráficos, no obstante vas a ver las diferencias en el siguiente apartado.

Vamos a enseñarte **tres formas** de hacer un mismo programa: **Un semáforo**

Nota: Acuerdate en toda esta sección de poner la Echidna en modo Sensor



## 1.2.1 MONTAJE 1 SEMAFORO CON MBLOCK

### PRIMERO QUÉ ES MBLOCK



\*

Es un programa propietario de la empresa MakeBlock basado en Scratch 2.0 Open Software para sus robots.

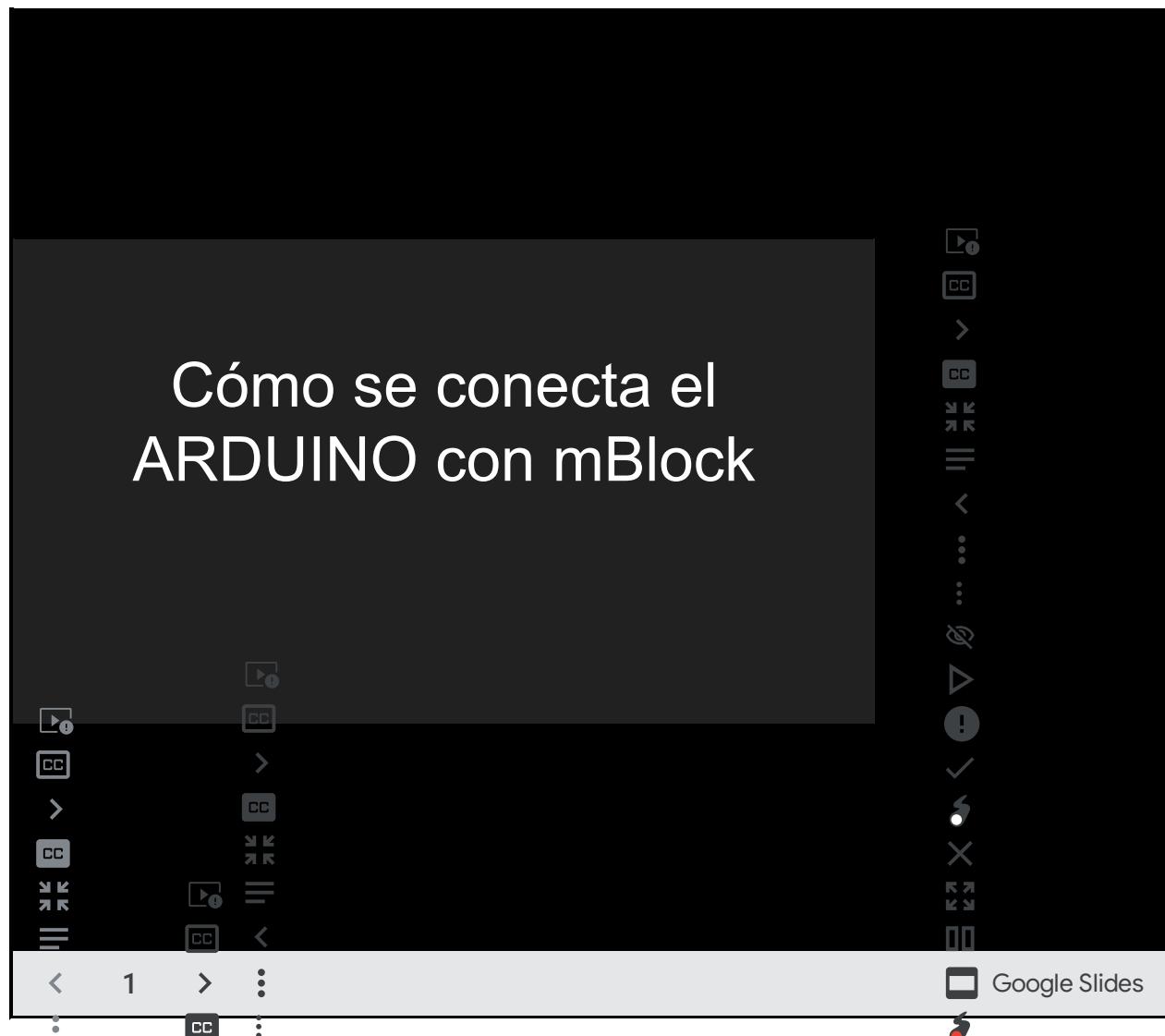
Aunque la anterior definición hace pensar que es un programa que no nos interesa, todo lo contrario, pues los robots de Makeblock están basados en Arduino por lo tanto se programa realmente el corazón de Arduino.

Que sea un software propietario, tampoco nos tiene que echar para atrás, pues es gratuito, multiplataforma y nos asegura el mantenimiento y las actualizaciones.

Su página de descarga es <http://www.mblock.cc/>

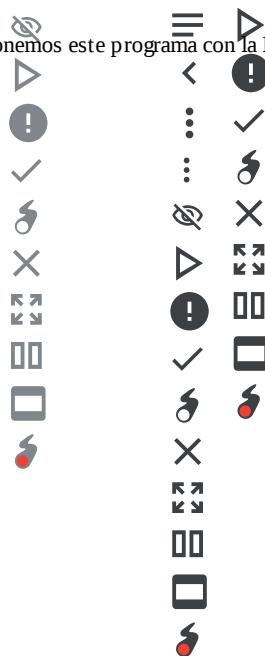
### SEGUNDO LA CONFIGURACIÓN DE MBLOCK

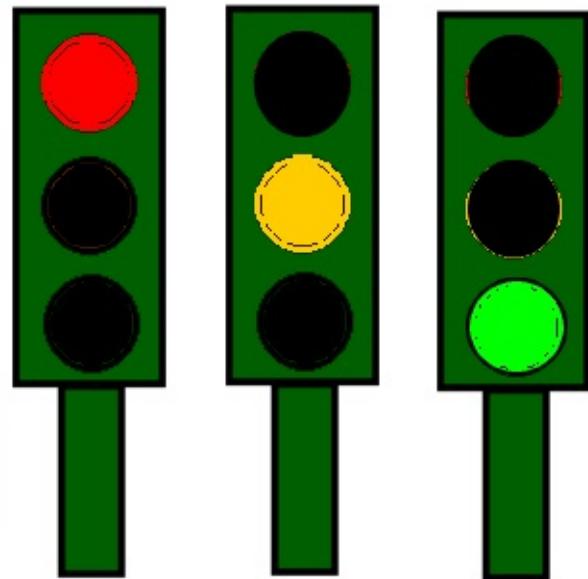
En esta presentación se explica CÓMO SE CONECTA MBLOCK CON EL ARDUINO



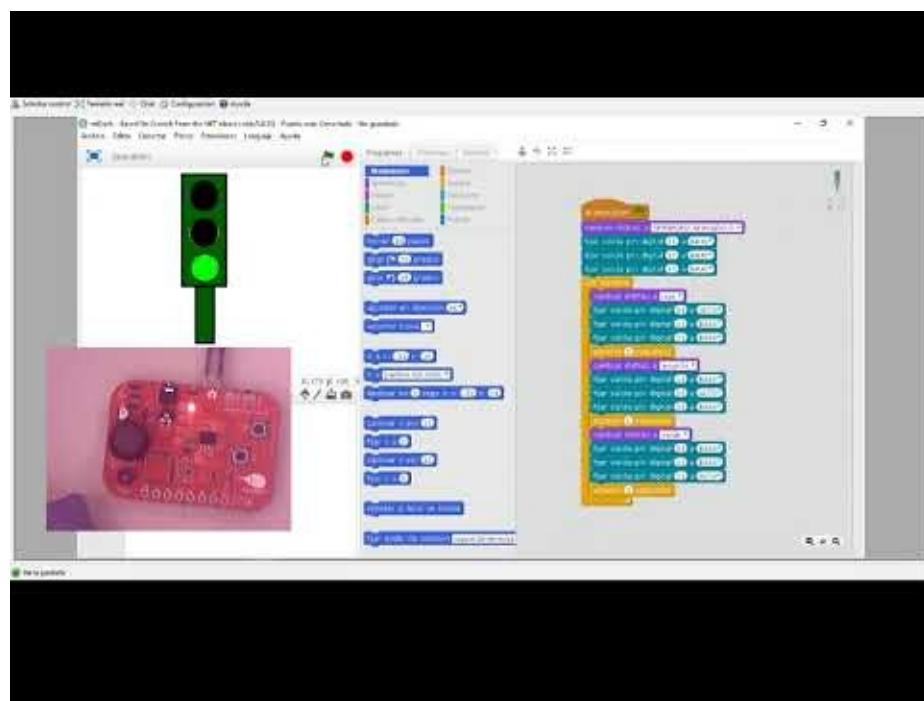
### TERCERO VER EL EJEMPLO DEL SEMÁFORO Y ECHIDNA

Ponemos este programa con la Echidna y en mBlock hemos cambiado el disfraz del oso panda por tres semáforos:





Y este es el resultado: [Video](#)



[Video link](#)

## 1.2.2 MONTAJE 1 SEMAFORO CON SNAP4ARDUINO

### PRIMERO QUÉ ES SNAP4ARDUINO



Es también un programa gráfico basado en Scratch, multiplataforma, libre y abierto y permite muchas versiones de placa, bibliotecas, etc... evolución del S4A

Usa el firmware standard [Firmda](#) ¿que es un firmware ? es un programa que se graba y se ejecuta en una placa hardware, en el caso del firmware Firmda, hay que instalarlo en el Arduino para que se comunique con Snap4Arduino. (en mBlock también hay un firmware pero está contenido en el propio mBlock y se instala cuando le damos a Conectar-Actualizar Firmware). Si todo esto del firmware no lo entiendes del todo, no pasa nada, es la explicación de los pasos que se explica en apartado de la configuración, no es necesario para programar, pero si lo entiendes, mejor.

El programa se puede descargar de su web <http://snap4arduino.rocks/>

En principio es la aplicación más acorde con la filosofía de la Shield Echidna.

### SEGUNDO LA CONFIGURACIÓN DE SNAP4ARDUINO

En esta presentación se enseña COMO CONFIGURAR SNAP4ARDUINO para que se comunique con nuestro Arduino y la Echidna

¿Cómo configurar mi Snap4Arduino para que se entienda con mi EchidnaShield?

O cómo cargar el Firmware que tenga las librerías y configuraciones necesarias para que el Arduino se comunique con Snap4Arduino

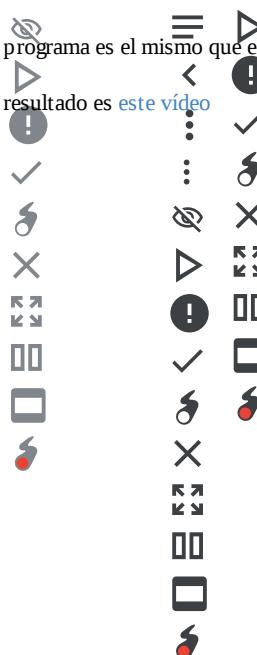
CATEDU CENTRO ARAGONÉS de TECNOLOGÍAS para la EDUCACIÓN

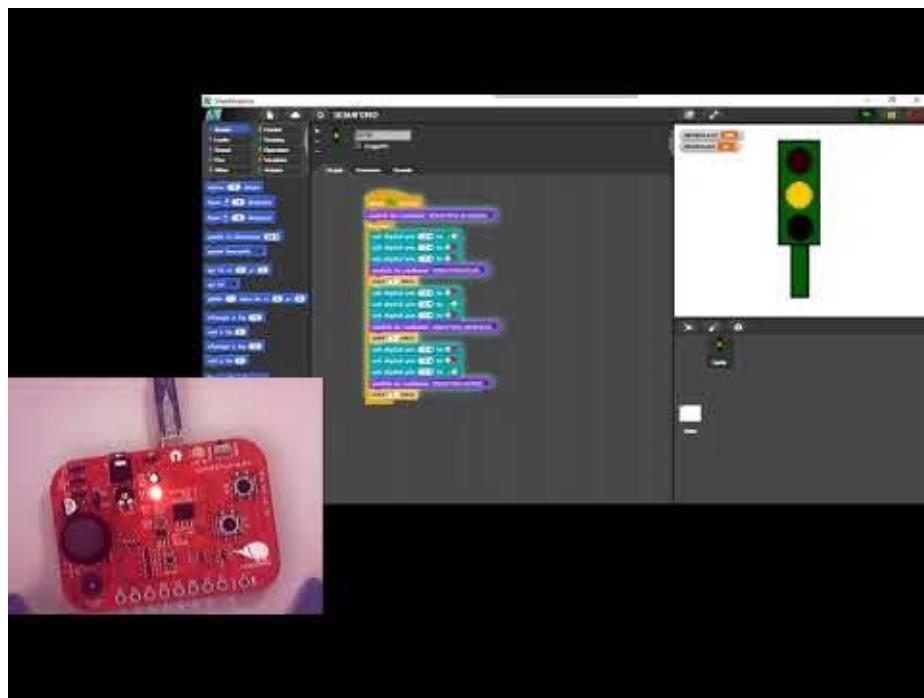
Google Slides

### TERCERO EL EJEMPLO SEMÁFORO CON ECHIDNA

El programa es el mismo que el anterior, con los mismos disfraces y se puede abrir desde Snap4Arduino desde [esta URL](#)

El resultado es [este vídeo](#)





[Video link](#)

## 1.2.3 MONTAJE 1 SEMÁFORO MEDIANTE CÓDIGO

### PRIMERO QUÉ ES "MEDIANTE CODIGO"

Es la forma de trabajar de forma profesional el Arduino: con su lenguaje código, en realidad, los otros lenguajes traducen el programa

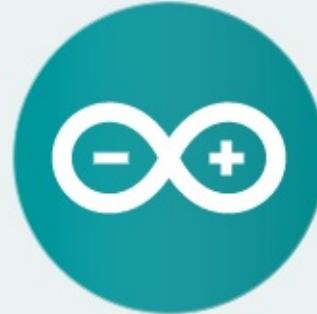


gráfico en lenguaje código Arduino, es decir son meros intermediarios.

El programa se puede descargar de su web oficial, aunque también hay una versión online <https://www.arduino.cc/>

### SEGUNDO LA CONFIGURACIÓN

Aquí no hay que instalar ningún Firmware, pues el código, o sea tú programa, es el mismo “firmware”. Digamos que no necesitas intermediarios si tratas con el agricultor ;)

### TERCERO EL EJEMPLO SEMÁFORO CON CÓDIGO

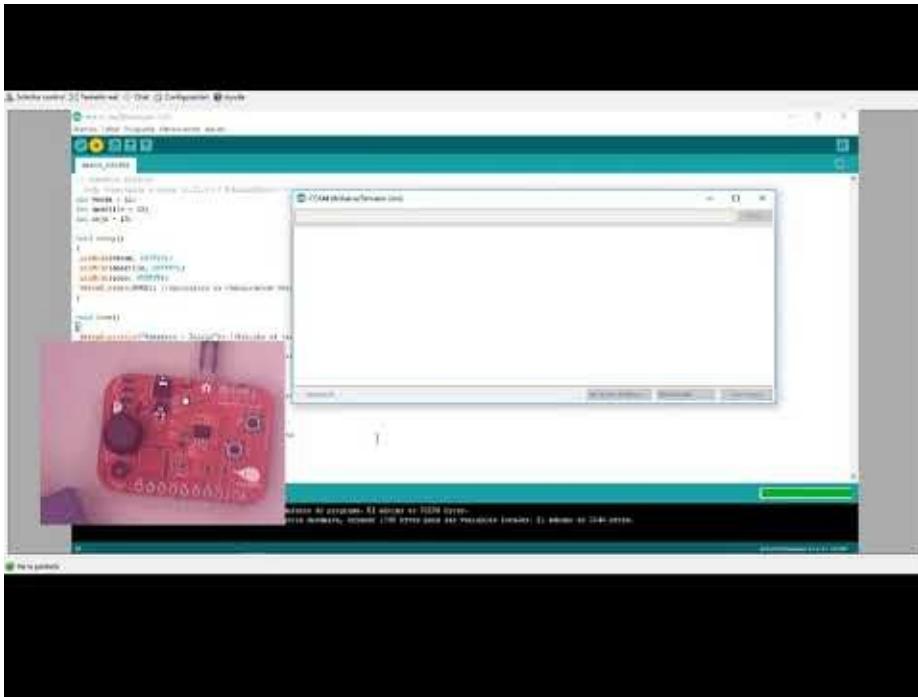
Aquí es donde vemos que la programación no es apropiada para Primaria, y la explicación de cada línea necesitaría un curso entero. ([Como el que hay en Aularagón !](#) pero si eres un poco *pito* seguro que eres capaz de leerlo intuitivamente sin problemas.

```
/* Semáforo Arduino
   Leds conectados a pines 11,12,13 = EchidnaShield */
int verde = 11;
int amarillo = 12;
int rojo = 13;

void setup()
{
  pinMode(verde, OUTPUT);
  pinMode(amarillo, OUTPUT);
  pinMode(rojo, OUTPUT);
  Serial.begin(9600); //inicializa la comunicación Serial
}

void loop()
{
  Serial.println("Semáforo - Inicio"); //Escribe el texto
  digitalWrite(verde, HIGH);
  Serial.println("Semáforo - Verde"); //Escribe el texto
  delay(2000);
  digitalWrite(verde, LOW);
  digitalWrite(amarillo, HIGH);
  Serial.println("Semáforo - Amarillo"); //Escribe texto
  delay(1000);
  digitalWrite(amarillo, LOW);
  digitalWrite(rojo, HIGH);
  Serial.println("Semáforo - Rojo"); //Escribe el texto
  delay(2000);
  digitalWrite(rojo, LOW);
}
```

Este programa se escribe (o copia y pega ¡es un texto !!) en el software del Arduino y se carga en la placa, en [el vídeo](#) se ve muy bien:



[Video link](#)

## Desventaja

En este programa queremos que te fijes en una **desventaja**: La interacción con el ordenador se pierde frente a mBlock y Snap4Arduino:

- En los lenguajes gráficos, tenemos a **interacción con el ordenador** igual que en Scratch: si te fijas en la pantalla el dibujo del semáforo va cambiando de disfraz para representar los colores de la luz que se enciende a la vez que en el Arduino.
- En un programa grabado en el Arduino **perdemos esa interacción**, lo máximo que podemos visualizar es una ventana donde se representa en formato texto qué es lo que le está pasando al Arduino (en el ejemplo del semáforo sale "Semáforo - Verde, Semáforo - Rojo ....").

Esto lo vemos como una **desventaja** desde el punto de vista de la enseñanza de la programación pues perdemos el potencial de interactuar con los elementos del ordenador crear personajes, disfraces, sonidos, teclado..., es decir **el sprite** (el oso panda que aparece por defecto en mBlock, la flecha que aparece por defecto en Snap4Arduino o el que nosotros creamos, como este ejemplo el dibujo del semáforo).

## Ventaja

Pero... (siempre hay un “pero” para estropear la fiesta) la programación en código tiene una **ventaja**: Se graba en el Arduino, no hay ningún intermediario entre nuestro programa y el Arduino (en mBlock y Snap está el ordenador), quien manda en el Arduino es nuestro programa, no nuestro ordenador, esto se traduce en: **rapidez !**, en mBlock veremos que podemos grabar nuestro programa en Arduino, pero perdemos la interacción con el sprite, como lo veremos [más adelante](#).

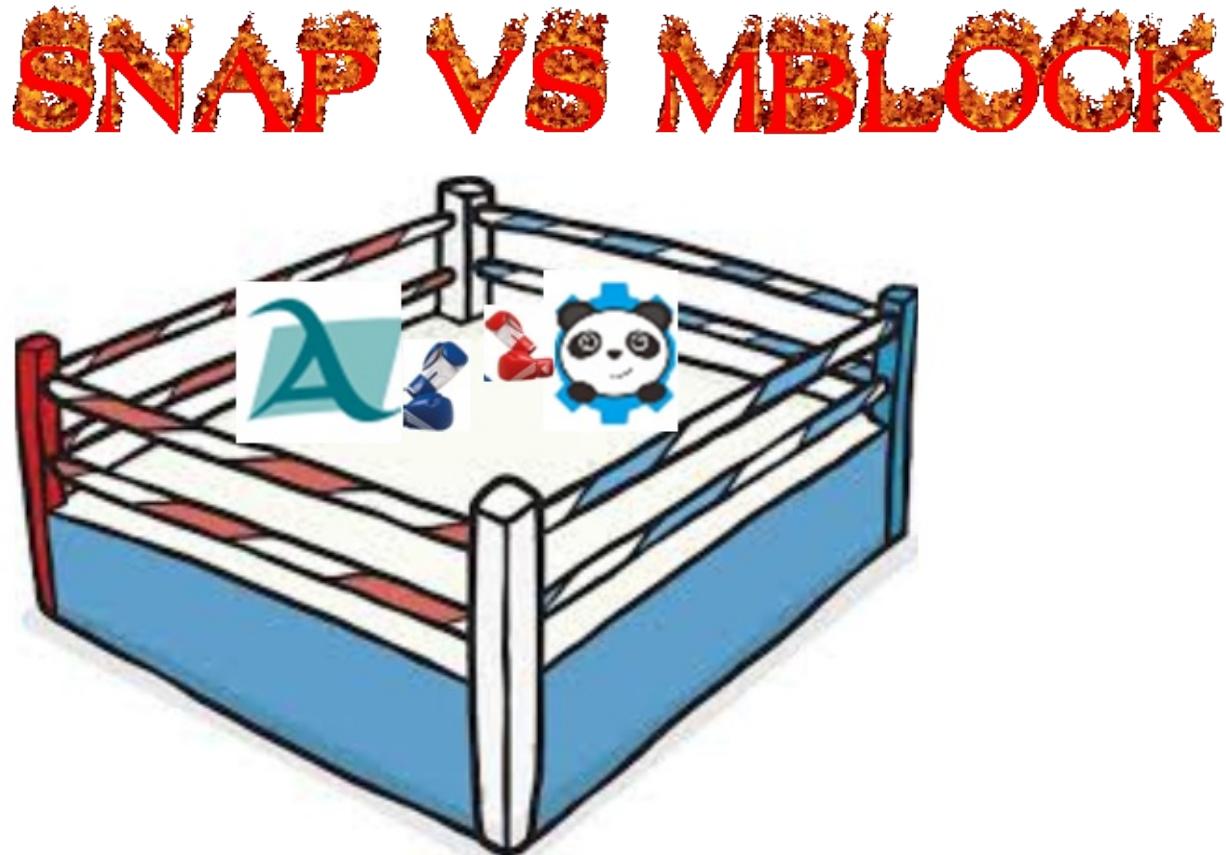
## Conclusión

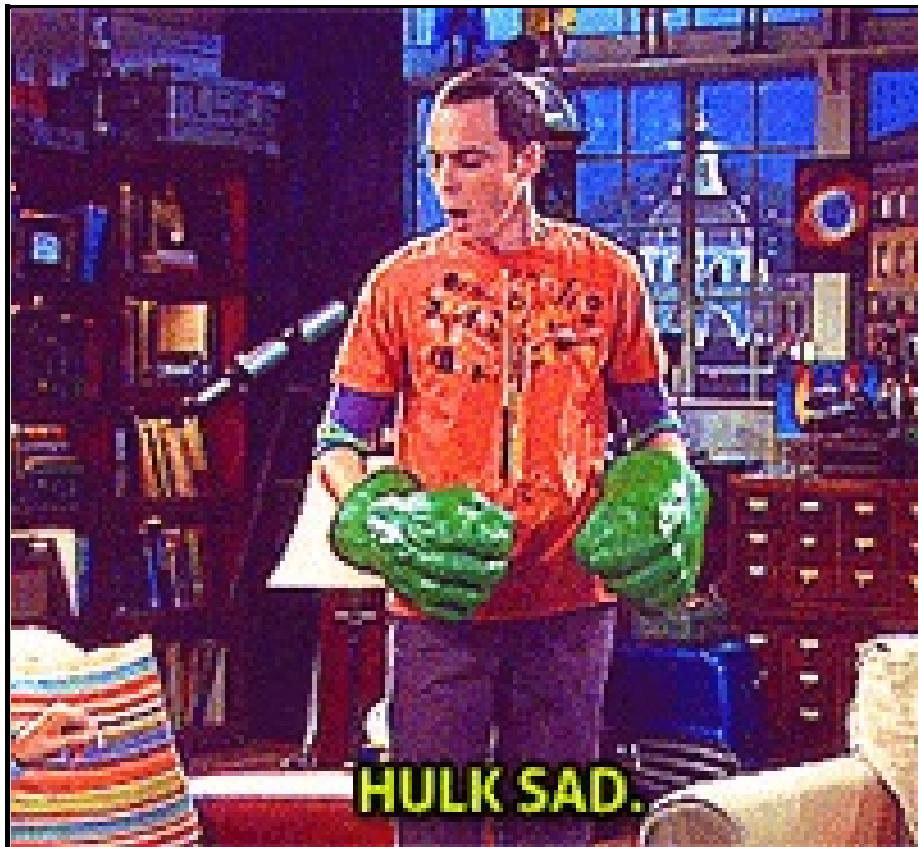
Profesionalmente es mejor utilizar lenguaje con código pero en la enseñanza es mejor el lenguaje gráfico, además lo hemos dicho en el apartado [¿Cómo se programa Echidna?](#) es una Shield Educativa que lo lógico es utilizar un lenguaje adaptado al nivel educativo: Gráfico. Pero ... ¿cuál?

## 1.2.4 mBlock vs Snap4Arduino

Conclusión: mejor el lenguaje gráfico

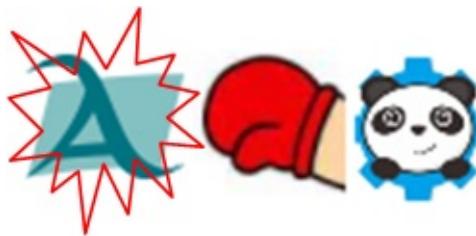
¿Pero cual? Siempre es mejor moverse en estándares, y actualmente el lenguaje gráfico que está más extendido y donde hay más proyectos es **Scratch**, luego vamos a centrarnos en dos programas que se basan en Scratch: mBlock y Snap pero ¿cuál es mejor? Snap4Arduino o mBlock



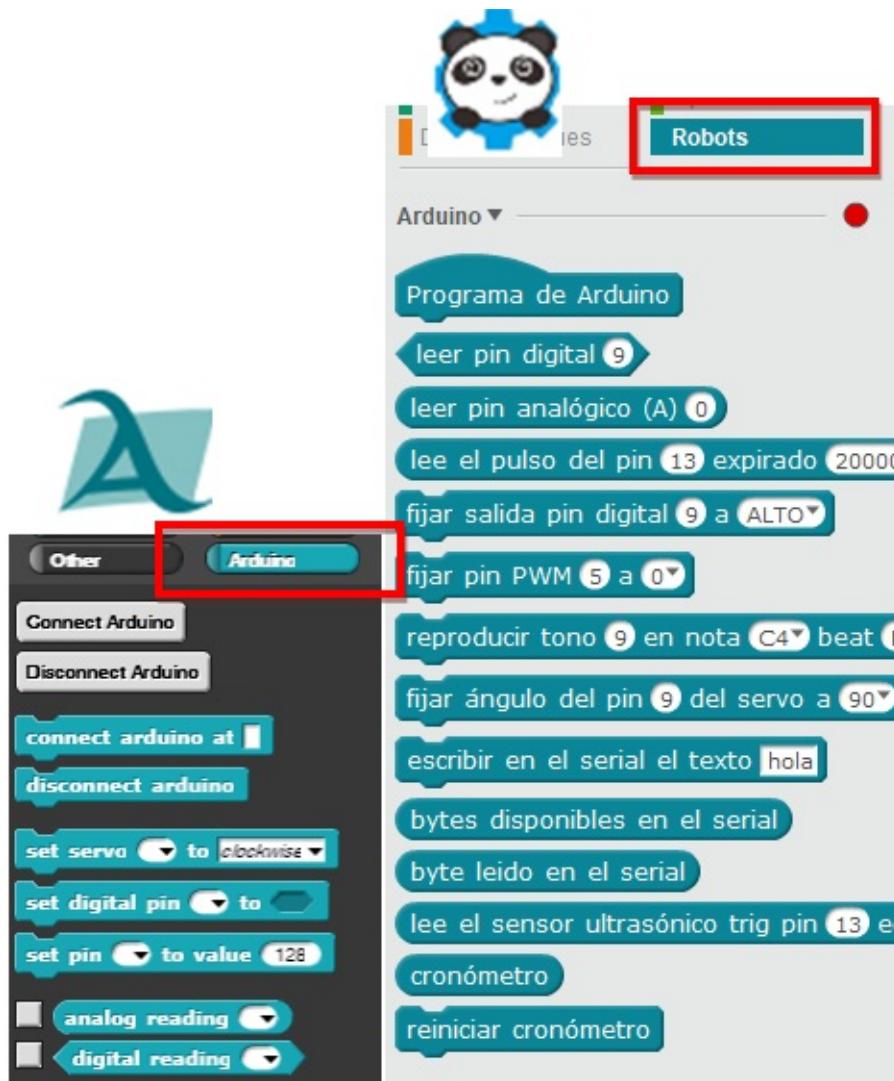


via GIPHY

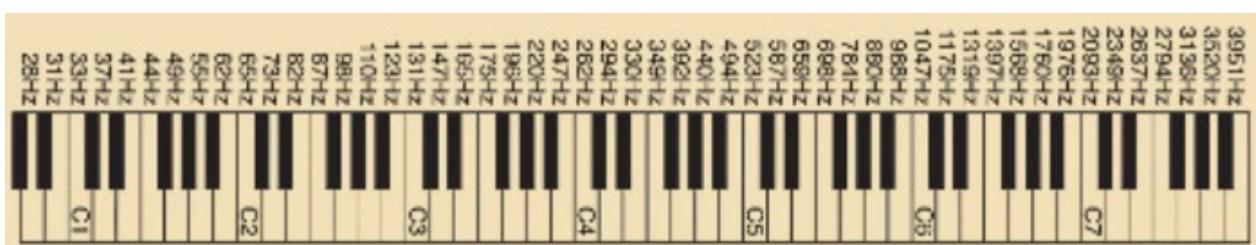
### 1.2.4.1 Instrucciones específicas para Arduino



Si entramos en la sección de instrucciones específicas para Arduino, en Snap4Arduino sólo tenemos las instrucciones básicas simples, a partir de estas piezas tienes que hacer el puzzle. Mientras que en mBlock tenemos más instrucciones que nos simplifican los programas pues la misma instrucción implican varias de las simples básicas:

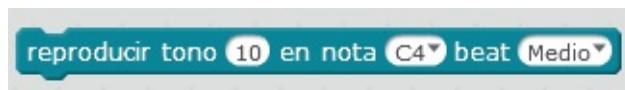


Esto parece una tontería pero no lo es: por ejemplo, supongamos que queremos reproducir el tono C4 es más o menos 262Hz:



Luego tiempo que dura la onda es  $=1/262 = 0.0038$  segundos es decir el tiempo de “encendido HIGH” es  $0.0038/2= 0.0019$ seg y el tiempo de “apagado LOW” es  $0.0035/2= 0.0019$ seg

o sea... que para hacer esta instrucción en mBlock



necesitas hacer todo esto en Snap4Arduino :

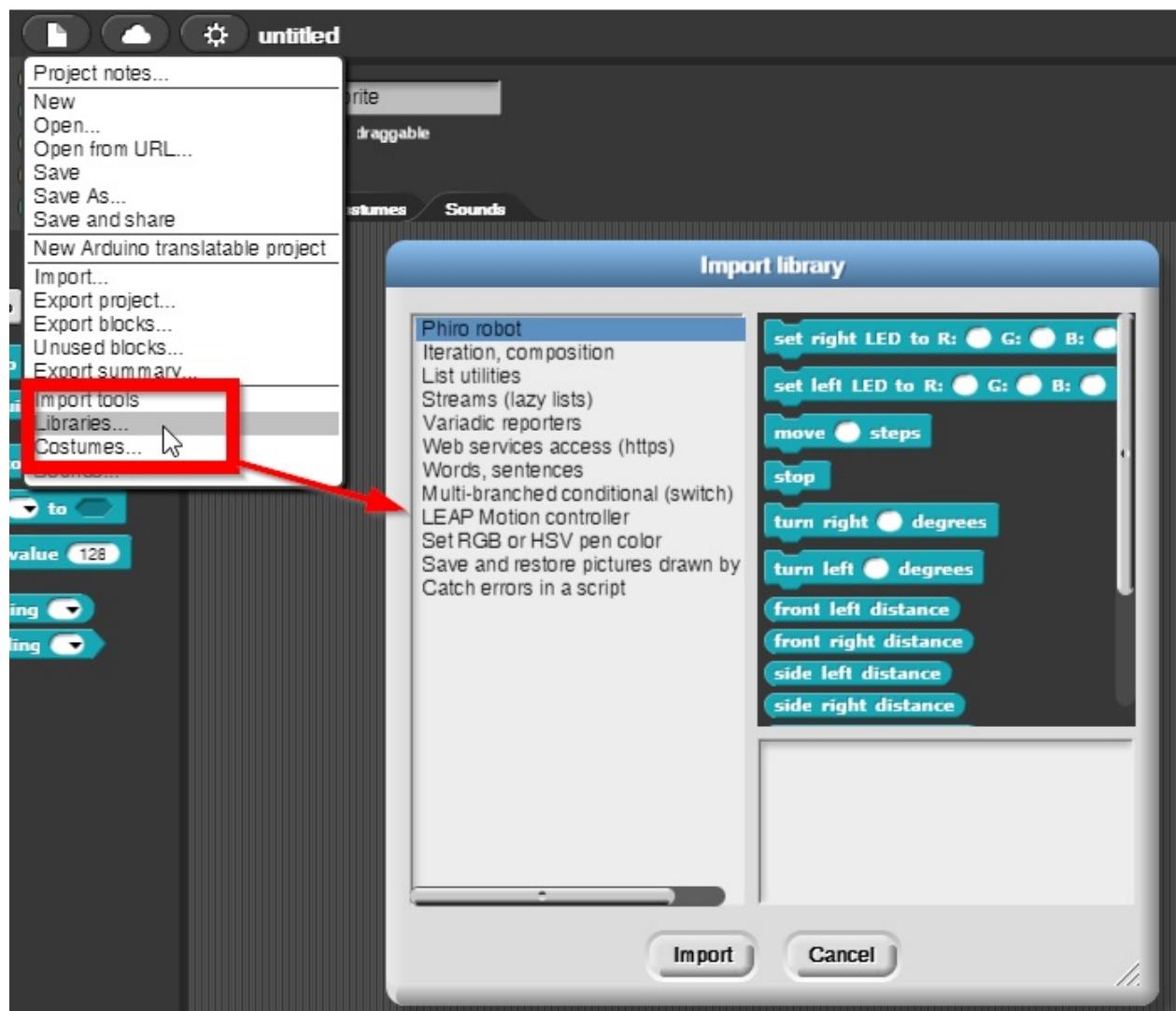


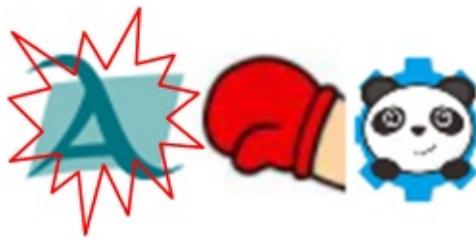
### 1.2.4.2 Importación de librerías



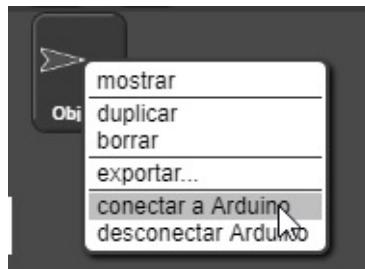
Snap4Arduino puede importar instrucciones o librerías que pueden aumentar el repertorio de instrucciones.

mBlock también tiene en Extensiones-Administrar extensiones pero no son tan específicas para el Arduino.



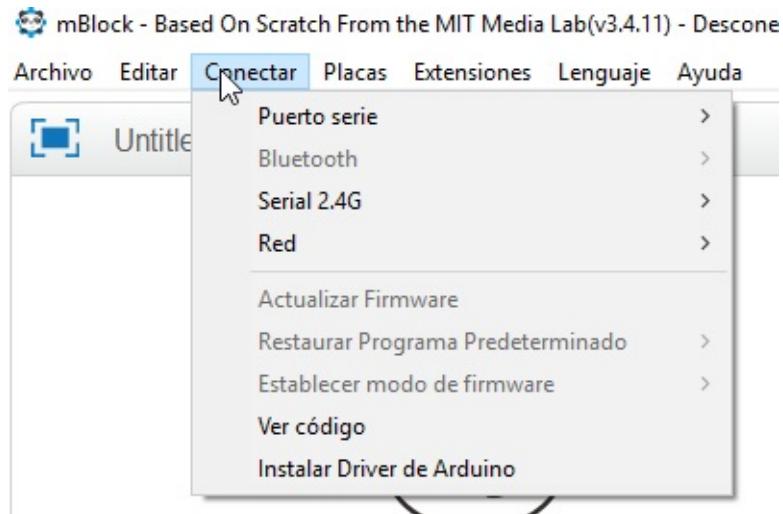


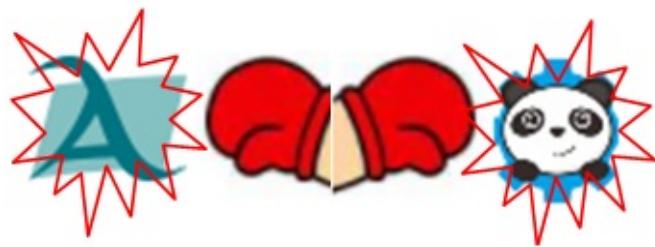
Vamos a fijarnos las diferentes posibilidades que tiene de conectarse el programa con el Arduino



En Snap4Arduino hay que dar con el botón derecho en el sprite y sólo permite conexiones puerto serie.

Mientras que en mBlock hay un menú con la opción de Conexiones y diferentes caminos para conectarse: Puerto serie, Bluetooth, 2.4G .. nos permite por ejemplo tener nuestro Arduino conectado de forma remota, necesario para ciertos proyectos, ejem un coche





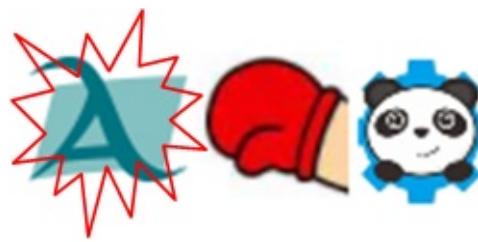
Si vemos un proyecto en [Scratch](#) que nos gusta y lo queremos reutilizar lo descargamos y si lo queremos reutilizar...

**En mBlock** lo podemos abrir, pero **no** siempre va bien.

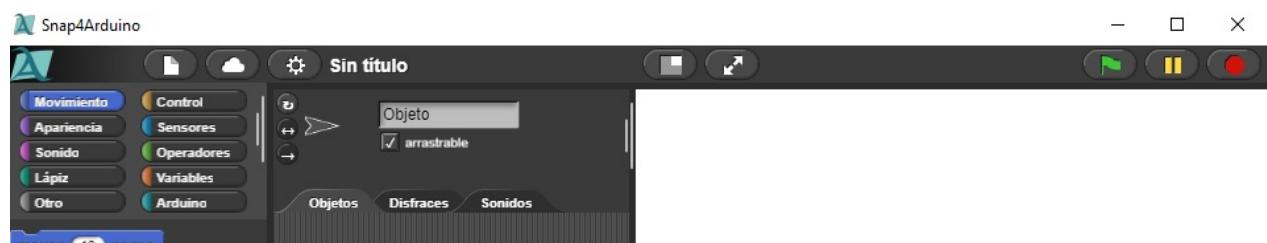
**En Snap4Arduino** no acepta ficheros Scratch, pero hay conversores online como [Snapin8r](#), y también da problemas.

Conclusión: tortazo a los dos !!

Lo veremos mejor más adelante en [esta sección](#).



En Snap4Arduino no existe un menú arriba, las opciones no se encuentran tan inmediatamente

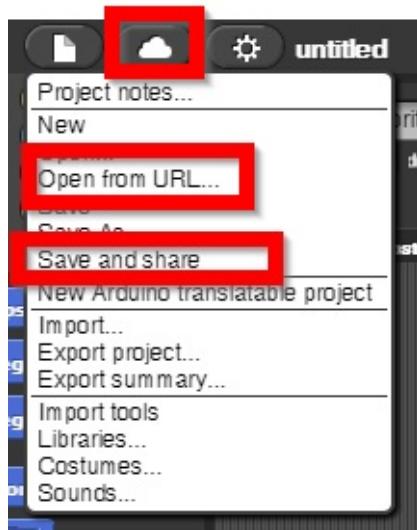


Mientras que en mBlock es más intuitivo:





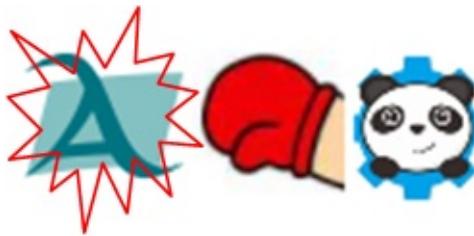
Snap4Arduino permite grabar tus programas en la nube, esto es una gran ventaja, pues permite al alumnado estar en cualquier ordenador, y seguir trabajando con sus programas que están en su repositorio en Internet.



Incluso puede grabar y compartir con otra persona el proyecto, (la otra persona recibe una copia del programa, no el original), esto permite publicar en un blog por ejemplo los programas. [Ver como](#).

mBlock sólo trabaja en modo local.

En mBlock podemos poner la vista de Modo Arduino Y NOS MUESTRA NUESTRO PROGRAMA ESCRITO EN LENGUAJE GRAFICO CONVERTIDO A LENGUAJE CÓDIGO



Esto nos permite **SUBIR AL ARDUINO** es decir, nuestro programa grabarlo en el Arduino como si fuera un Firmware y ganar velocidad aquí tienes una captura y puedes observar cómo traduce el programa gráfico en código:

```

1 #include <Wire.h>
2 #include <SoftwareSerial.h>
3
4
5 double angle_rad = PI;
6 double angle_deg = 1;
7 double tiempo;
8
9 void setup() {
10   tiempo = 0.00175;
11   pinMode(2, INPUT);
12   pinMode(10, OUTPUT);
13   pinMode(3, INPUT);
14 }
15
16 void loop(){
17   if(digitalRead(2) == HIGH)
18     for(int i=0; i<1000; i++)
19       digitalWrite(10, HIGH);
20   else
21     delay(tiempo);
22 }
  
```

**QUEREMOS EXPLICARTE LO IMPORTANTE QUE ES ESTO** El anterior programa funciona bien si lo hacemos subir al Arduino, pero si lo hacemos de forma normal (cambiamos el evento “Programa Arduino” por evento bandera):



#### NO FUNCIONA BIEN

¿por qué?: por la lentitud.

Si apretamos el pulsador 2 le pedimos al ordenador que encienda y apague D10 a una velocidad de 0.00175 segundos esto lo tiene que procesar el ordenador, pasa por el cuello de botella del puerto serie y al final se ejecuta en el Arduino a una velocidad mucho más baja, más o menos 0.5 segundos, en vez de oírse un tono, se oye tut,tut,tut,tut....

Si apretamos el pulsador 3 sí que se oye bien, porque mBlock manda el código del bucle de golpe para que se oiga bien.

Volveremos a este programa varias veces, no te preocupes si no lo has entendido del todo bien.

**Desventajas** de subir un programa al Arduino:

1. **NO PODEMOS INTERACTUAR CON EL ORDENADOR** es decir, no podemos hacer que el sprite (el oso por defecto) se mueva o que al pulsar una tecla del ordenador... porque EL ARDUINO ES INDEPENDIENTE DEL ORDENADOR, incluso podemos desconectarlo y sigue funcionando !!!
2. **Nos hemos cargado el Firmware** que tenía antes !! luego al finalizar tu programa, tenemos que volver a configurar el Arduino para que haga caso al programa mBlock visto en [esta presentación](#).

**Ventajas** de subir un programa al Arduino:

1. **Rapidez**, tenemos la ventaja del lenguaje con código, el programa se ejecuta a la velocidad del Arduino sin tener el ordenador como intermediario.
2. **Independencia**: Alimentamos el arduino con una batería !! y funciona !! esto es esencial para proyectos que impliquen movimiento y no queremos tener el ordenador encendido para que funcione.

### Aunque...: Rapidez en la simulación

Aunque Snap4Arduino no sube el programa, sólo simula, es mucho más rápido en la simulación que mBlock



En este [apartado](#) veremos las diferencias.

Nosotros le vamos a dar el premio a mBlock

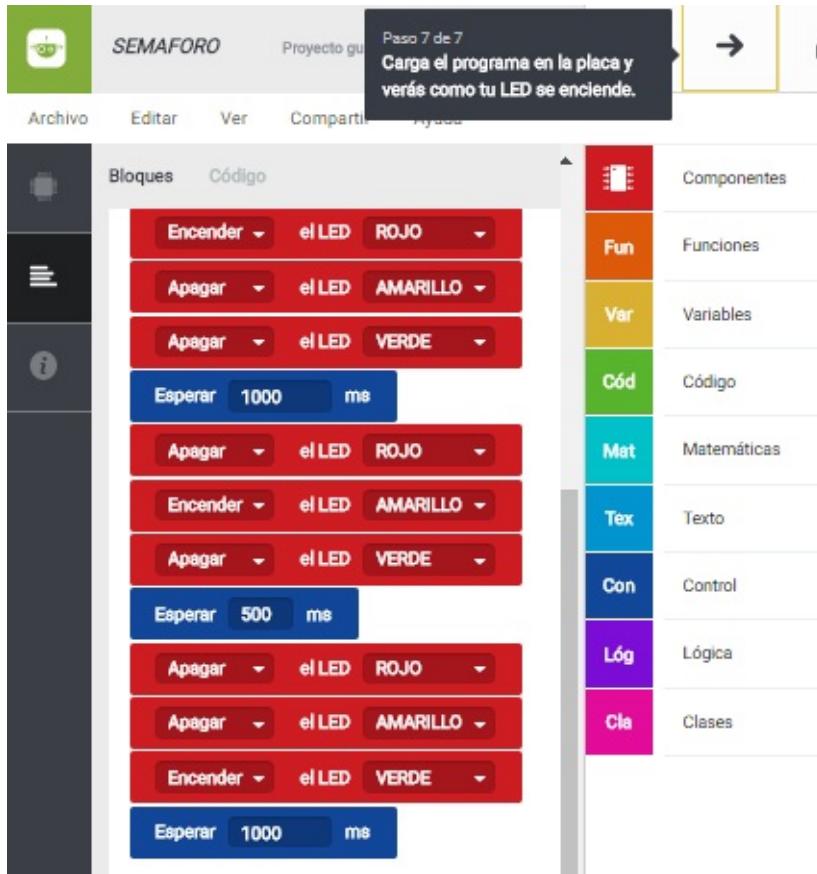


Todo es cuestión de gustos, las capturas y descargas de programas de este curso serán en mBlock, pero algunas veces utilizaremos Snap4Arduino cuando nos interesa rapidez en la simulación.,.

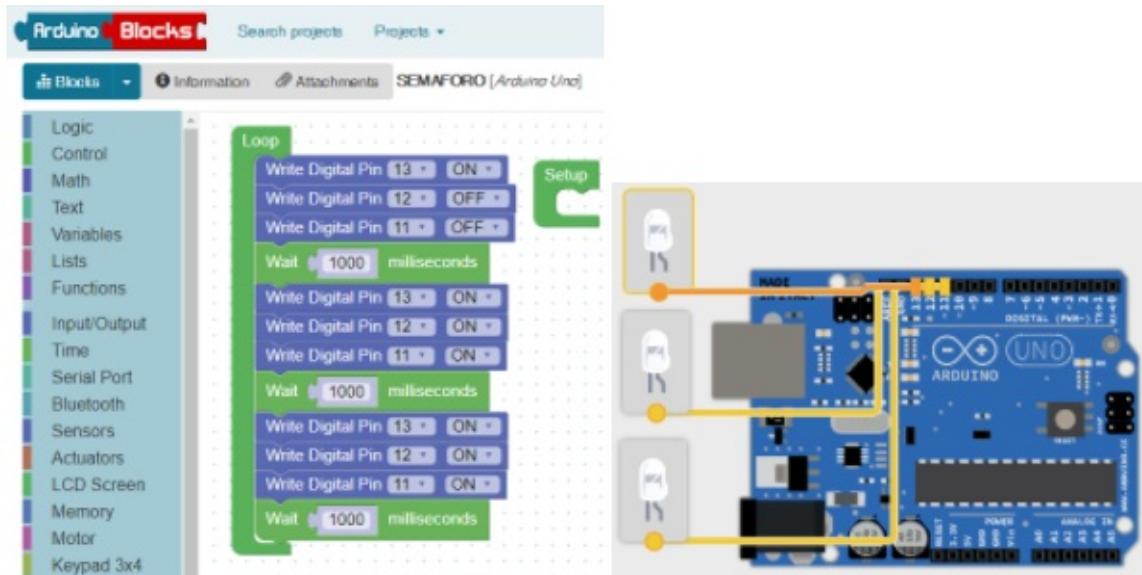
Ten en cuenta que si te gusta Snap4Arduino puedes hacer este curso pero que tendrás que currarte [las instrucciones más elaboradas que tiene mBlock](#) y no podrás descargar tu programa al Arduino para que sea independiente [tal y como hace mBlock](#).

Existen otros programas gráficos, el ArduinoBlocks, Bitblock... pero no tienen la posibilidad de iteraccionar con Sprites del ordenador, ni tienen el mismo entorno del referente Scratch, por lo que no se utilizarán en este curso, pero son posibilidades interesantes, os mostramos dos capturas de pantalla del mismo programa semáforo visto anteriormente, fíjate que aquí no hay *Sprite*:

Con Bitbloq:



Con ArduinoBlocks:



Y un favorito nuestro el **VISUALINO** pues a la vez que haces programación gráfica, va enseñando el código a la derecha:

Ayuda

```
#include <IFT.h>

/** Global variables */
IFT TFTscreen = IFT(10, 9, 8);

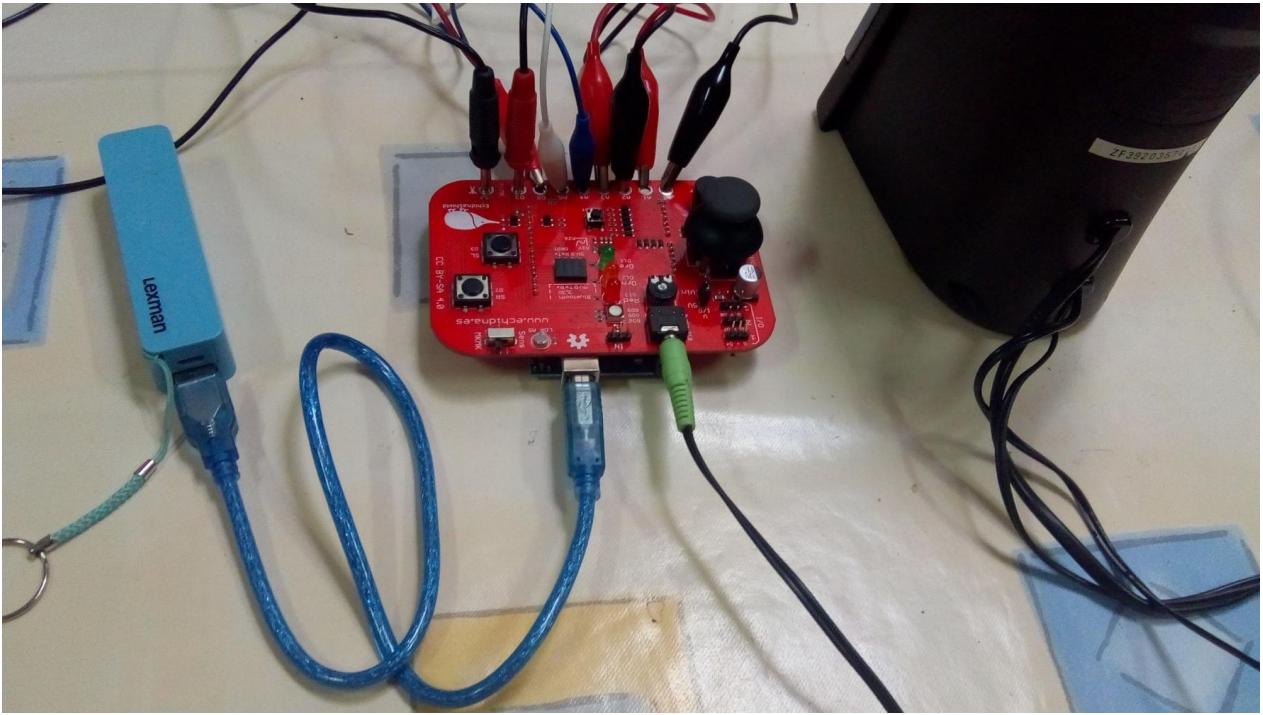
/** Function declaration */
void setup()
{
    TFTscreen.begin();
    TFTscreen.background(0, 0, 0);

    pinMode(A0, INPUT);
}

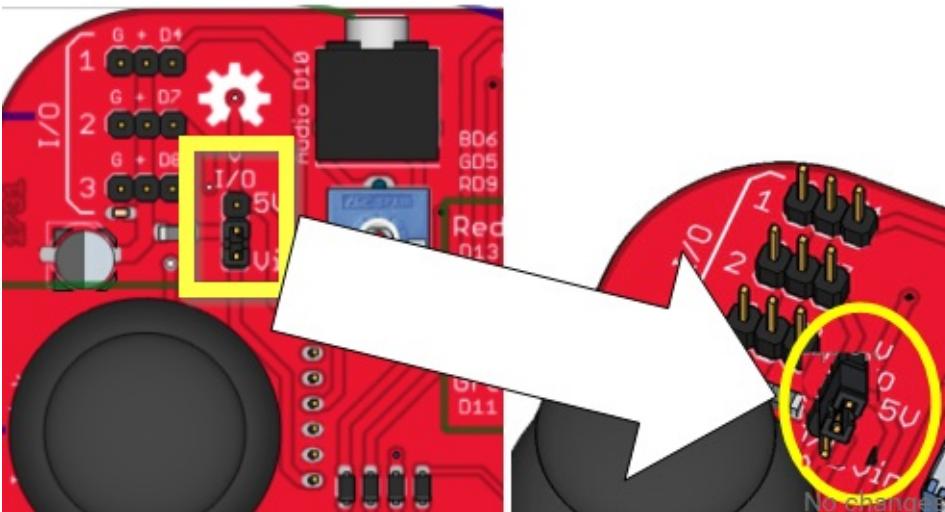
void loop()
{
    int Luz=(6.0 * analogRead(A0) * 1000)/1000;
    TFTscreen.stroke(255, 255, 255);
    TFTscreen.text(Luz_c_str(),0,0);
    TFTscreen.stroke(0, 0, 0);
    TFTscreen.text(Luz_c_str(),0,0);
}
```

## Alimentacion USB:

El Echidna **normalmente se alimenta a través de la alimentación USB** del Arduino, ya sea cuando está conectado con el ordenador, o cuando se utiliza un PowerBank por ejemplo, o cargador de móvil (esto ocurre cuando Arduino tiene el programa grabado en él y puede vivir sin el ordenador).



Lo que hay que hacer es que el jumper que tiene arriba a la izquierda tenerlo conectado a 5V:



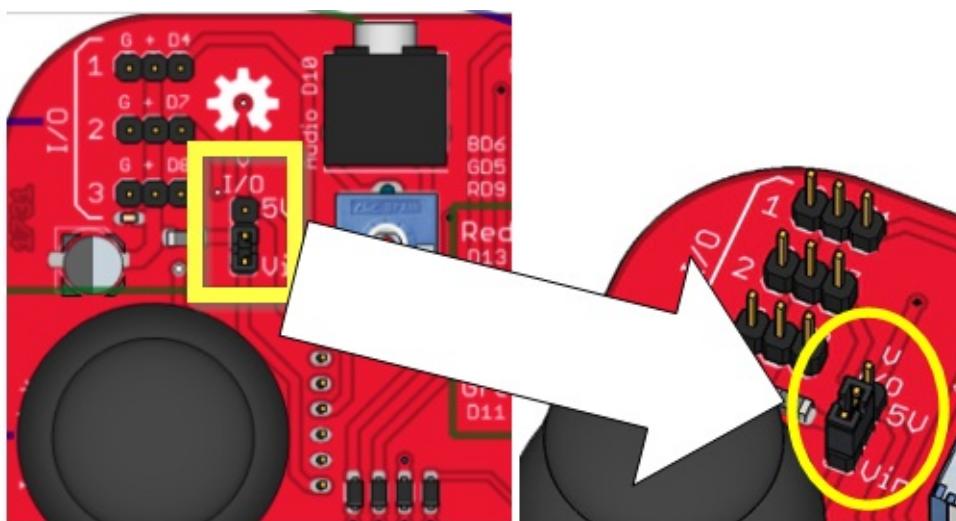
**No utilizar la alimentación 5V cuando los servos o dispositivos a controlar consuman más de 300mA**, para no sobrepasar el regulador del Arduino, en este caso utilizar la alimentación externa:

## Alimentacion externa:

Arduino se puede alimentar con un portapilas o fuente de alimentación, sin ningún problema, con un rango desde 9V hasta 12V.



Pero hay que decirle al Echidna que la alimentación no es por 5V sino por **Vin**, pues el Arduino también se alimenta por Vin, esto se hace cambiando el jumper:



#### Nota:

Es posible que en ocasiones nos interese utilizar **las dos alimentaciones a la vez**, por ejemplo queremos usar el Arduino dependiendo del ordenador, porque queremos por ejemplo que interactue con el *Sprite oso panda del mBlock* (por lo tanto está conectado al cable USB) pero *las extensiones que queremos conectar necesitan potencia* (más de 300mA) por lo tanto conectamos también un portapilas o fuente de alimentación, así pues estamos obligados de decirle al Echidna que alimente las extensiones por Vin cambiando el jumper a Vin.

## 1.3 Hoja de Ruta

¿Dónde se encaja este robot? ¿se puede comparar este robot con otros robots de otros cursos que hacemos desde CATEDU?

Esta [es la hoja de ruta](#), no se tiene que tomar al pie de la letra, pero intenta ayudar al profesorado que tenga una visión global de tanta oferta:

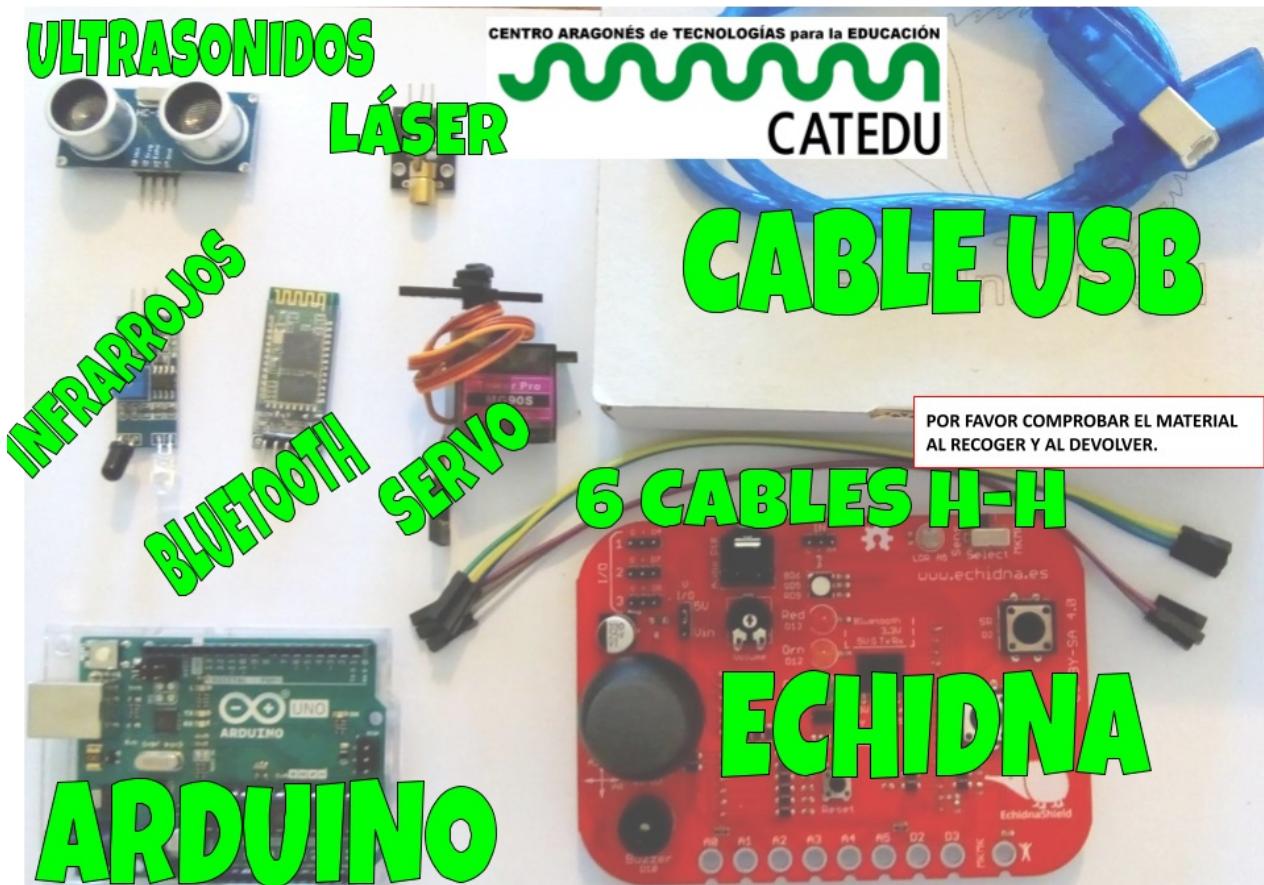
### RoboTICa

Oferta de formación en Pensamiento computacional del Centro Aragonés de Tecnologías para la Educación.



## 1.5 Kit de préstamo

Para hacer este curso te hemos preparado este kit de préstamo, donde tienes todo lo necesario para realizar este tutorial, menos unas bananas para hacer un [piano](#) :)



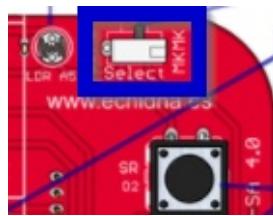
Ahora déjate de teorías, y vamos a jugar !!



via GIPHY

## Tema 2 Salidas de Echidna

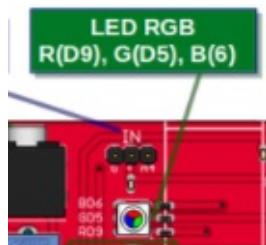
Nota: Acuerdate en toda esta sección de poner la Echidna en modo Sensor



Todos los programas de este curso se encuentran en este repositorio: <https://github.com/JavierQuintana/Echidna>

## 2.1 MONTAJE 2 LUCES PSICODELICAS

Ya hemos visto las luces LED con el [semáforo](#), ahora vamos a ver el **LED RGB**



Está conectado a las salidas digitales D5, D6 y D9 y si te fijas, en el Arduino tienen el símbolo ~ ¿Qué significa esto? Que son señales PWM. Si has leído [¿Qué es un Arduino?](#) habrás leído que es una señal digital pero los pulsos pueden variar su ancho de tal manera que la media puede ser una tensión entre 0 y 5V, que se gobiernan con esta instrucción:



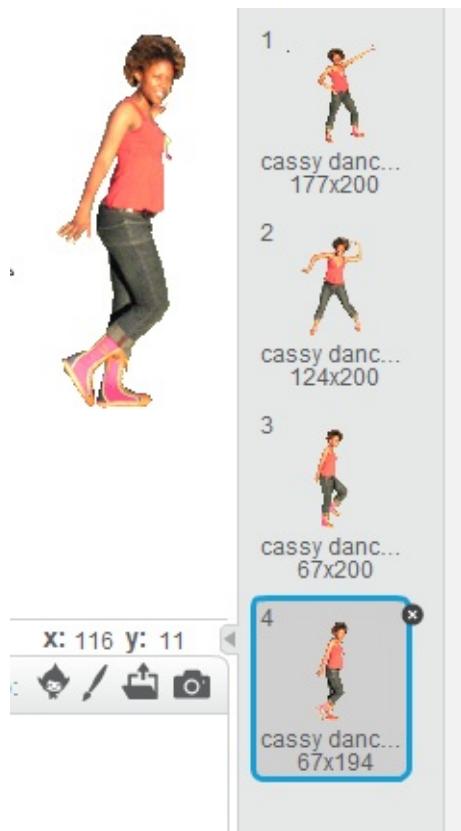
Donde 0 sería 0V y el valor 255 sería el valor máximo de tensión que en nuestro caso es 5V aunque puede ser otra tensión si se alimenta Arduino con una alimentación externa Vin.

### RETO

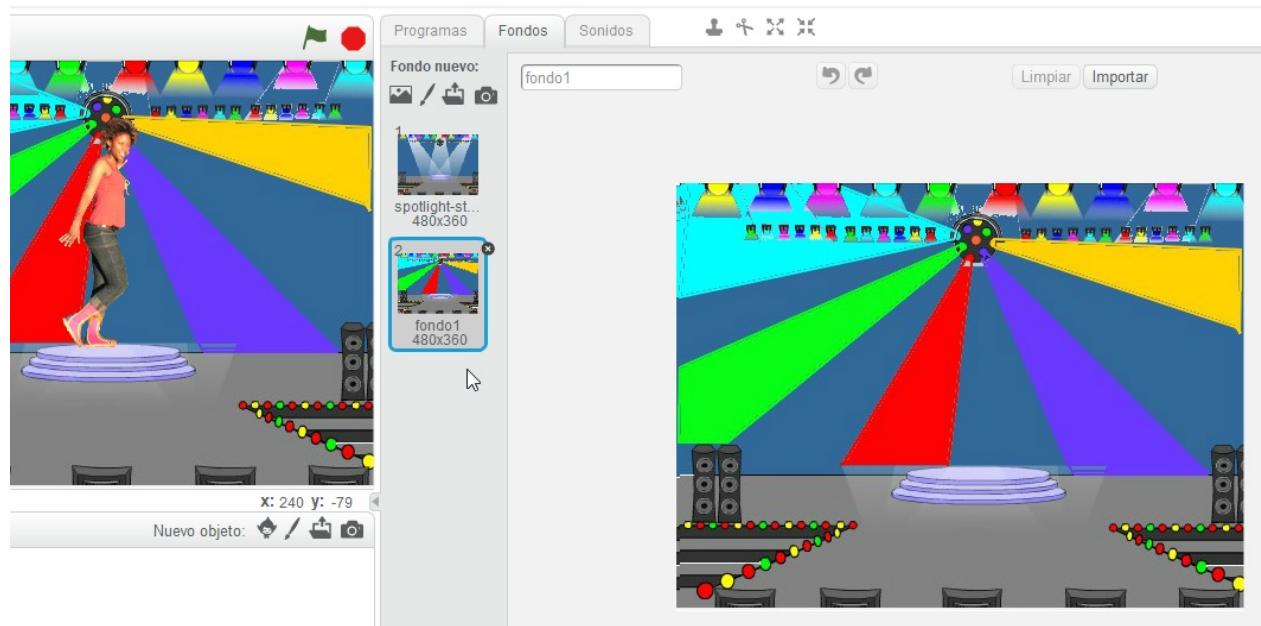
Vamos a realizar un programa con mBlock que se iluminen todas las luces de colores de forma aleatoria, además vamos a poner un fondo, un personaje y música para que parezca una discoteca.

### Solución

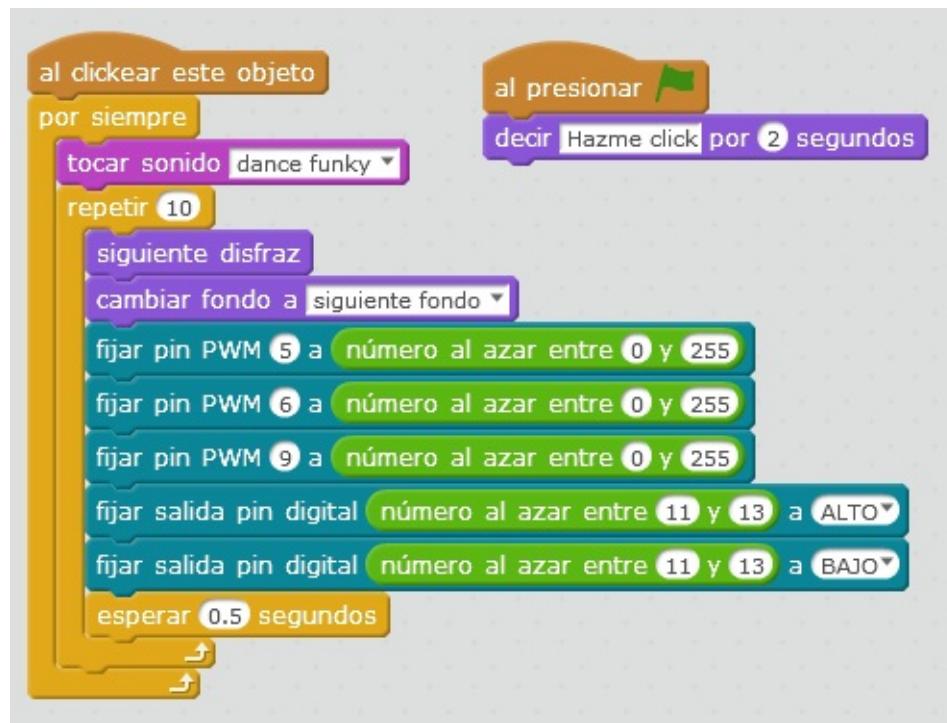
Vamos a utilizar un sprite de la biblioteca que tiene mBlock “cassy” que tiene ya los disfraces apropiados de baile:



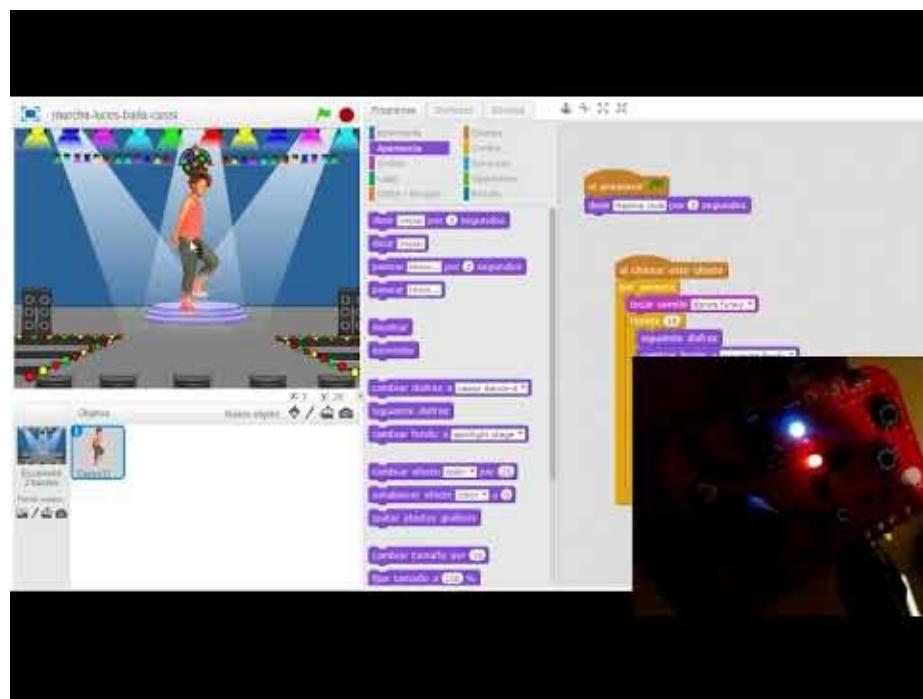
De fondo, utilizaremos el escenario de la biblioteca también “Spotlight” pero añadiremos otro pintando el mismo pero modificándolo de forma que parezca que salgan luces (aquí te ponemos el reto de copiar el escenario en un editor de imágenes, aunque sea el Paint y poner los haces de luces):



Y ya nos queda programar en el Sprite cassy para cumplir el enunciado:



El resultado es el siguiente video:



[Video link](#)

## 2.2 MONTAJE 3 TIMBRE

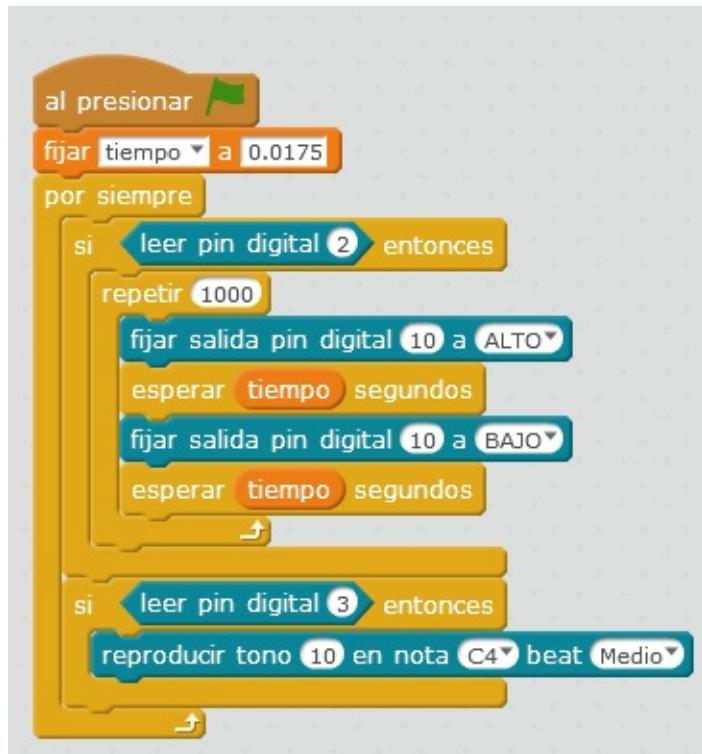
**RETO** Queremos que hagas un programa en el Echidna que:

- Cuando se pulsa D2 suene una onda cuadrada de 282Hz o sea una duración 0.035 segundos (0.0175seg el bajo y 0.0175seg el alto).
- Si se pulsa el botón D3 que suene el tono C4 que es aproximadamente esa frecuencia.

### SOLUCIÓN

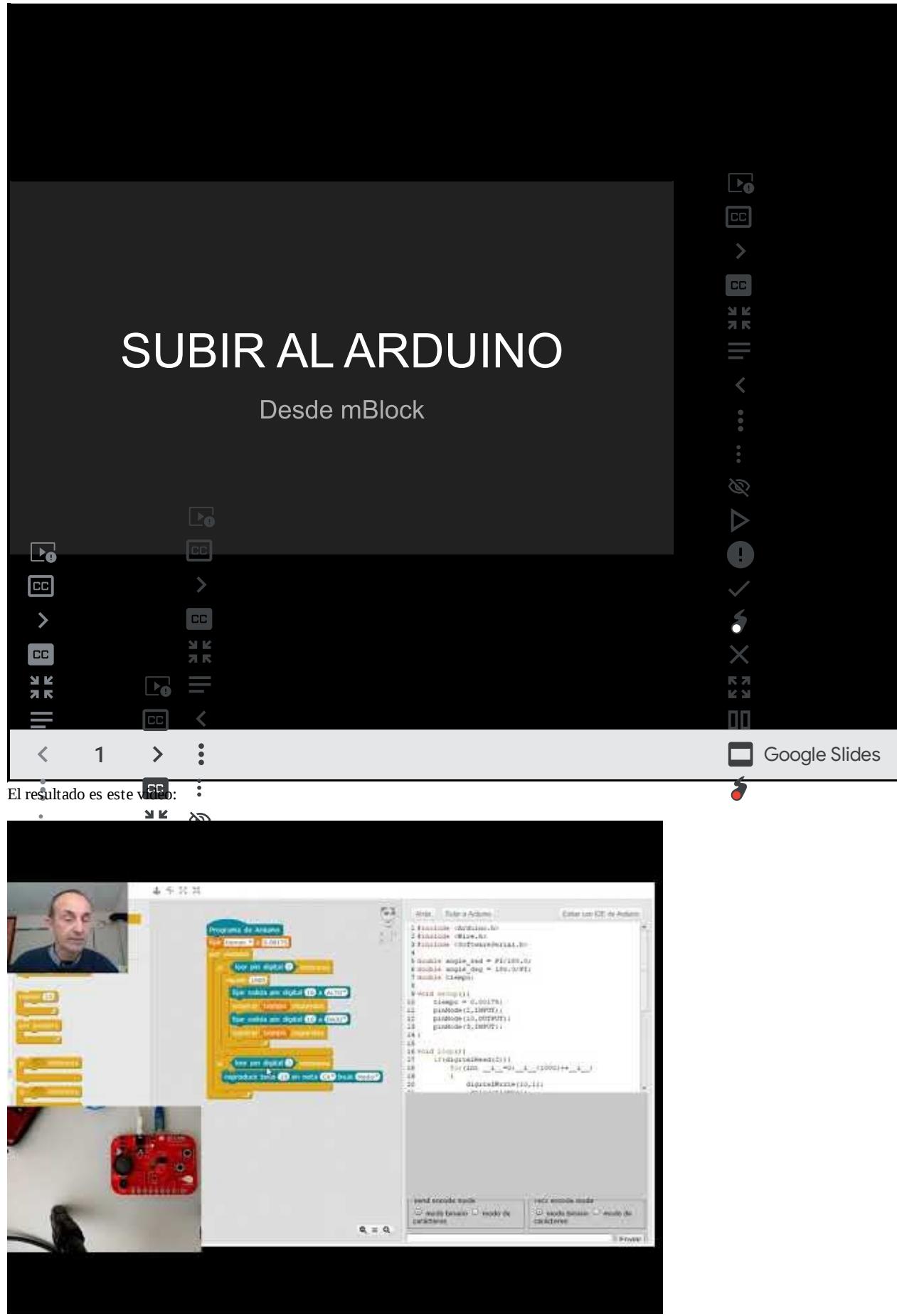
Solución

La solución seguro que te sonará [pues lo hemos visto en el Tema 1](#).



Si lo ejecutamos desde el ordenador, y pulsamos el botón D2 se oye un tut, tut, tut por los auriculares, ya dijimos en [este apartado](#) que la culpa lo tiene tu ordenador: No es capaz de enviar mBlock órdenes al Arduino a la velocidad de 0.00175 segundos. Si pulsamos el botón D3 se oye el tono C4 sin problemas.

La solución consiste en subir al arduino, ya lo hemos visto en el Tema 1 ¿no te acuerdas? vuelve a ver esta [presentación](#):

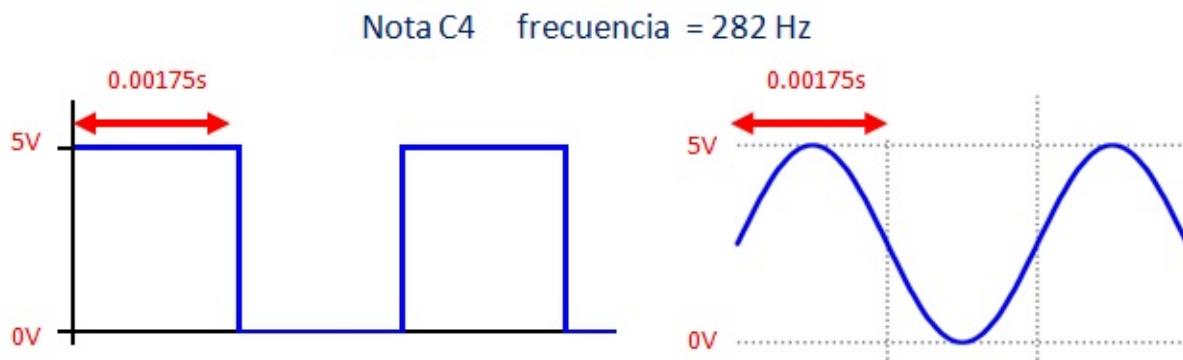


## Video link

Si pulsamos D2 el sonido es más agudo que pulsando D3 que corresponde a la nota C4 ¿por qué? porque la señal es cuadrada.

Más explicación

Los cálculos del tiempo de subida y bajada  $t = 0.00175\text{s}$  calculados anteriormente están bien hechos ¿por qué no reproduce bien el tono de la nota C4? por esto:

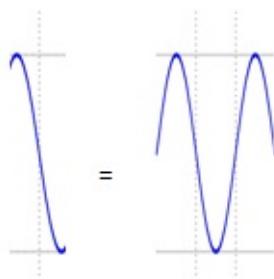


No es lo mismo una onda analógica sinusoidal de 282 Hz que una onda digital cuadrada que es lo que se reproduce en la salida digital D10

¿Y por qué se oye más agudo? Vamos a fijarnos en una transición por ejemplo la de bajada de 5V a 0V ¿A qué se parece más esa transición?

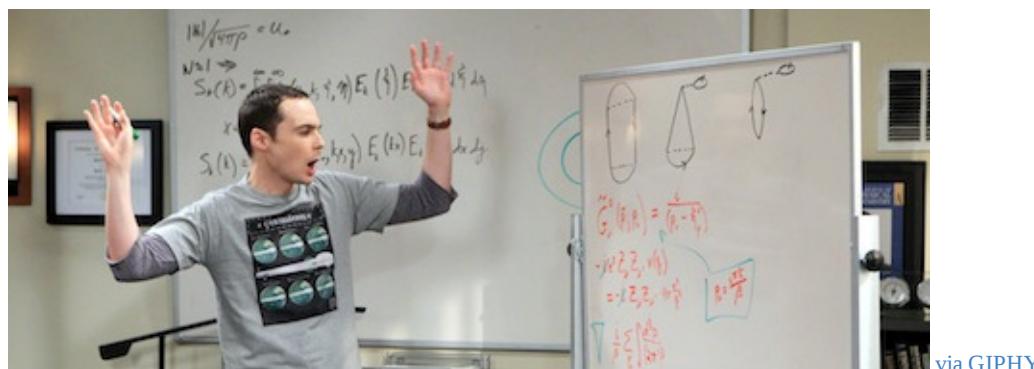


Evidentemente a la segunda, y esa pendiente tan vertical corresponde más a ondas agudas:



Por lo tanto se oye más agudo, por eso es

Ya sé que el anterior razonamiento, si lo lee algún físico, le producirá un rechinar de dientes



pues para los frikis: ... la transformada Fourier de una onda cuadrada produce armónicos de orden superior, incluso infinitos si fuese perfecta..., ja!, pero es mejor ser simple que ser incomprendible.

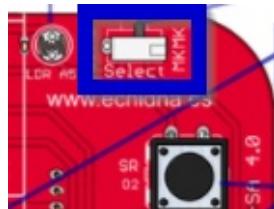


## 3 Entradas de Echidna

Aquí es donde vamos a sacar todo el jugo a esta Shield.

Nota: Si has hecho el anterior programa Grabar a Arduino, recuerda dejarlo preparado para que haga caso a mBlock: Conectar-Actualizar Firmware.

Nota: Acuerdate en toda esta sección de poner la Echidna en modo Sensor



Todos los programas de este curso se encuentran en este repositorio: <https://github.com/JavierQuintana/Echidna>

## 3.1 MONTAJE 4 INTERRUPTOR LUZ.

El anterior programa ya es un ejemplo de uso de estos botones digitales que están conectados a los pines digitales 2 y 3 de Arduino. Ten en cuenta que sólo pueden leer niveles lógicos.

### RETO1

Al pulsar el botón D2 se enciende los 3 leds del Echidna y al soltarlo se apagan. El programa “tunealo” un poco, quitando el oso como disfraz y añadiendo dos que sean Leds

#### SOLUCION RETO1:

Previamente hemos descargado de Internet o realizado por nosotros mismos (otra vez ponemos a prueba tus habilidades en el Paint) dos imágenes de leds y se lo añadimos al disfraz del oso (suponemos que sabes algo de Scratch) :



La solución es fácil:



### RETO2

Al pulsar D2 tiene que encenderse los leds, y **sólo se apagaran** si se vuelve a pulsar D2. O sea, *utilizar sólo un pulsador para encender y apagar los leds*\_.

## SOLUCION RETO2:

## UN POCO DE PARÉNTESIS TEÓRICO

Fíjate en el enunciado del RETO1: “al pulsar el botón D2 se enciende y al soltarlo se apaga” ES UNA MÁQUINA LÓGICA pues el estado de la máquina sólo depende de las entradas (en este caso de un botón): Pulsar la entrada (botón D2) produce una salida concreta (encender leds).

*CONTINUAMOS ...*

Pero tal y como está redactado, el RETO2 tiene que memorizar el estado anterior, no es trivial el enunciado “Al pulsar D2 tiene que encenderse los leds, y sólo se apagan si se vuelve a pulsar D2.” ES UNA MÁQUINA SECUENCIAL pues el estado de la máquina depende de las entradas y de lo que ha pasado antes. Pulsar la entrada (botón D2) NO produce una salida concreta (depende si estaba apagado o encendido anteriormente).

No pasa nada si no lo entiendes del todo, es teoría.

La programación se complica **necesitamos añadir una variable que recuerde lo que ha pasado antes** la vamos a llamar `_encendido` que recordará si está encendido los leds o no:



¿qué pinta ese “esperar 0.5s? je, je, je.... te vamos a dejar que lo experimentes tú, quitalo, pruébalo varias veces y verás...

Muchos de nuestros aparatos electrónicos se encienden y se apagan con el mismo botón, así que a partir de ahora aprecia que su funcionamiento no es trivial.

## 3.2 El LDR en A5

Primero de todo ¿Qué es un LDR? Vamos a la Wikipedia:



Fotorresistor

*Una fotorresistencia es un componente electrónico cuya resistencia disminuye con el aumento de intensidad de luz incidente. Puede también ser llamado fotorresistor, fotoconductor, célula fotoeléctrica o resistor dependiente de la luz, cuyas siglas, LDR, se originan de su nombre en inglés light-dependent resistor. Su cuerpo está formado por una célula fotorreceptora y dos patillas.*

W [More at Wikipedia \(ES\)](#)

Lo verás en el Echidna arriba un poco a la derecha, y está conectado a la entrada analógica del Arduino A5 y según <http://echidna.es/> los valores van desde 20 en ausencia de luz, hasta 1.000 con mucha luz.

### 3.2.1 Comprobar los límites

No os creáis al pie de la letra los límites oficiales del LDR! Los componentes electrónicos no son ideales, cada uno es particular, probar esos límites en vuestro Echidna, para ello vamos a utilizar nuestro mBlock y que nuestro simpático oso panda nos diga esos valores, con este sencillo programa:



El valor que leemos con máxima luz (utilizando una linterna) y el valor de máxima oscuridad (a tapar con el dedo, no te compliques) no llegan a los límites que se marcan oficiales en [Echidna](#) EN TU CASO PUEDEN SER OTROS !! pero parecidos.

### 3.2.2 MONTAJE 5 SEMÁFORO LUMINOSO

**RETO** Vamos a reutilizar esfuerzos: reutilizar el semáforo visto en [esta sección](#) pero vamos a hacer que se iluminen los colores según la luz:

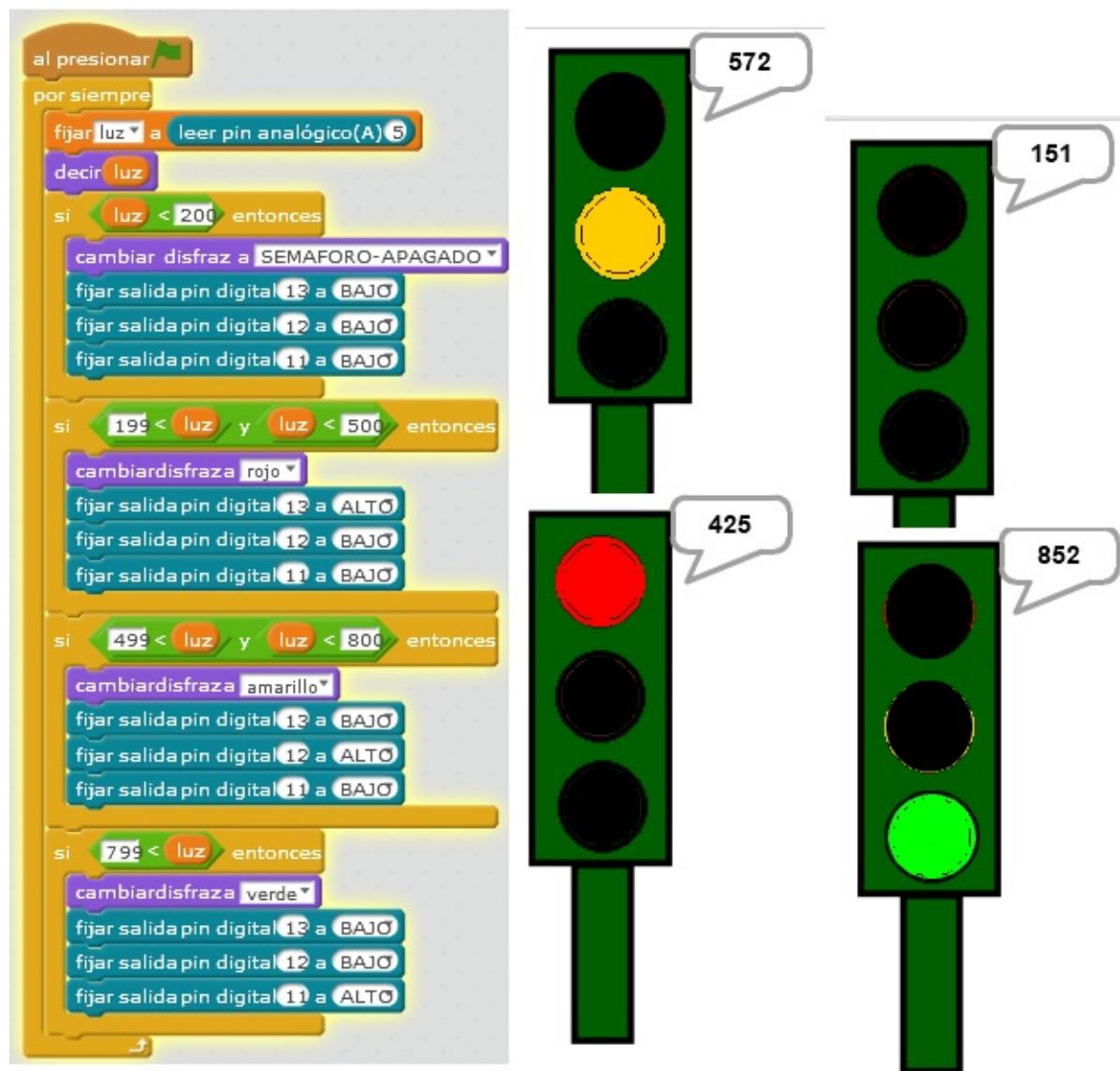
Semáforo	Luz
Todo apagado	Mucha oscuridad
Rojo	Oscuridad
Amarillo	Luz normal
Verde	Mucha luz

#### SOLUCIÓN

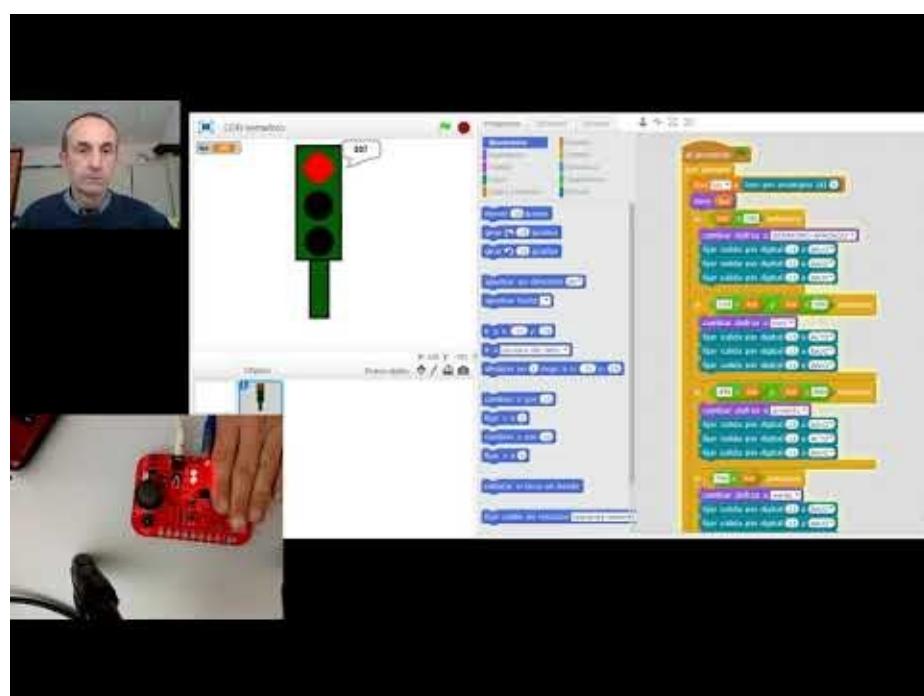
Según nuestros límites medidos en el anterior programa fijamos nuestras propias fronteras para cumplir el enunciado (puedes poner otros valores parecidos):

Semáforo	Luz	Límite inferior	Límite superior
Todo apagado	Mucha oscuridad	---	199
Rojo	Oscuridad	200	499
Amarillo	Luz normal	500	799
Verde	Mucha luz	800	---

El programa sería el siguiente, donde hemos capturado los semáforos en los tres estados utilizando una linterna y nuestra mano para oscurecer el LDR:



El resultado lo puedes ver en [este vídeo](#):



[Video link](#)

## MONTAJE 6 PIANO LUMINOSO

**Nuestro RETO** es el siguiente:

*Realizar un programa que suene una nota diferente según la luz*

Parece sencillo pero no lo es ¿por qué? Porque hay cambios de escala: el LDR nuestro trabaja con valores distintos al de las notas, por lo tanto hay que hacer UN CAMBIO DE ESCALA, y esto necesita un apartado diferente, te recomendamos ver [3.2.3.1 Cambios de escala](#)

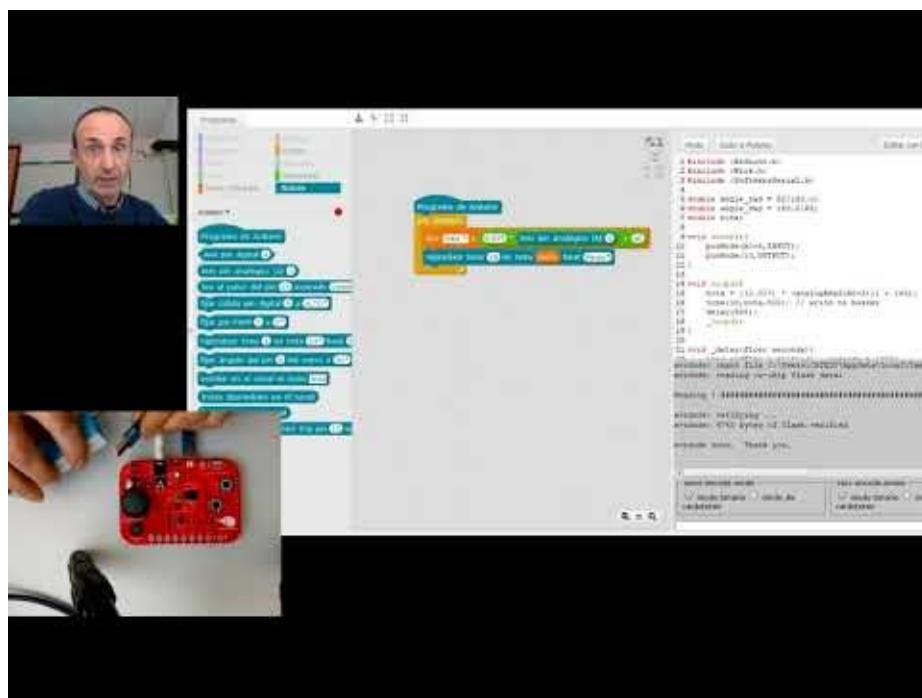
Solución

El programa es pues el siguiente



¿Por qué lo hacemos con la opción de subir a Arduino? Porque la simulación va lenta (se oye tut-tut-tut) si lo subes al Arduino reproduce el tono correctamente. [VER COMO SUBIR AL ARDUINO](#).

El resultado lo puedes ver en [este vídeo](#):

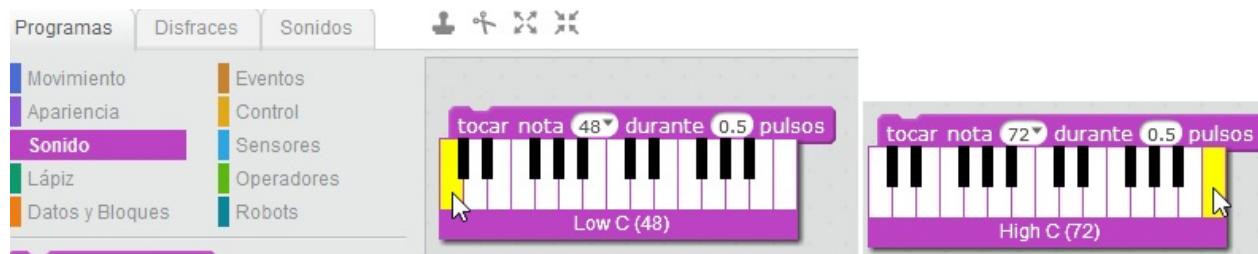


[Video link](#)

## Cambios de escala

Nos encontramos que:

- Los valores de entrada, es decir, el LDR trabaja con unos límites, que según hemos visto en el apartado comprobar los límites, el nuestro va de 108 a 982, llamaremos a esta **variable de entrada X**
- Los valores de salida son los tonos, que mBlock trabaja con la norma americana (la europea es el típico Do-Re-Mi-Fa-Sol) y los americanos, son valores numéricicos que van desde 48 hasta 72 (lo puedes comprobar con la instrucción “tocar nota” que está en Programas-Sonido). Llamaremos a la **variable salida de notas Y**:



Problema: ¿cómo convertimos X en Y?

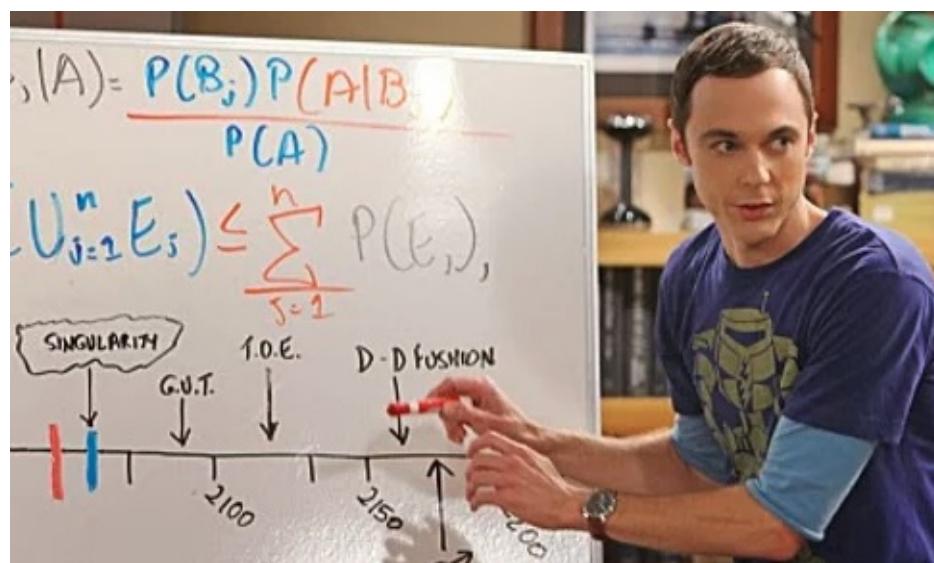
Matemáticamente es una recta con una pendiente m y una ordenada n :

$$Y = m \cdot X + n$$

Para calcular m y n tenemos que utilizar un sistema de ecuaciones dadas las condiciones límites de X e Y:

- Cuando X = 108 quiero que Y valga 48:  $48 = m \cdot 108 + n$
- Cuando X = 982 quiero que Y valga 72:  $72 = m \cdot 982 + n$

Ala! dos ecuaciones y dos incógnitas: calcula m y n



Buen ejercicio para los alumnos para que vean matemáticas aplicadas

No obstante, como esto lo repetiremos muchas veces los cambios de escala, hemos confeccionado [ESTA HOJA DE CÁLCULO](#) que te lo puedes descargar y facilita las cosas: Ponemos en las celdas amarillas los valores límites :

- $X_1 = 108 \quad X_2 = 982$
- $Y_1 = 48 \quad Y_2 = 72$

Y nos da los valores m y n automáticamente:

A	B	C
<b>HOJA DE CALCULO DE CAMBIO DE ESCALAS</b>		
' = m X + n <b>CAMBIA LO AMARILLO</b>		
Valores que tengo de x		
X1=	108	
X2=	982	
Valores que quiero de Y		
Y1=	48	
Y2=	72	
<b>RESULTADOS</b>		
m =	0,02745995	
n =	45,0343249	

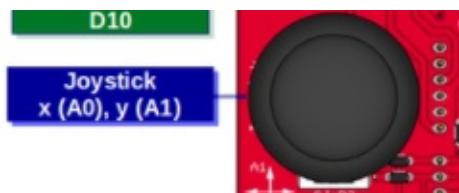
Luego la fórmula para el cálculo de la Y (las notas) en función de la luminosidad X es:

$$Y = 0.027 * X + 45$$

### 3.3 Joystick

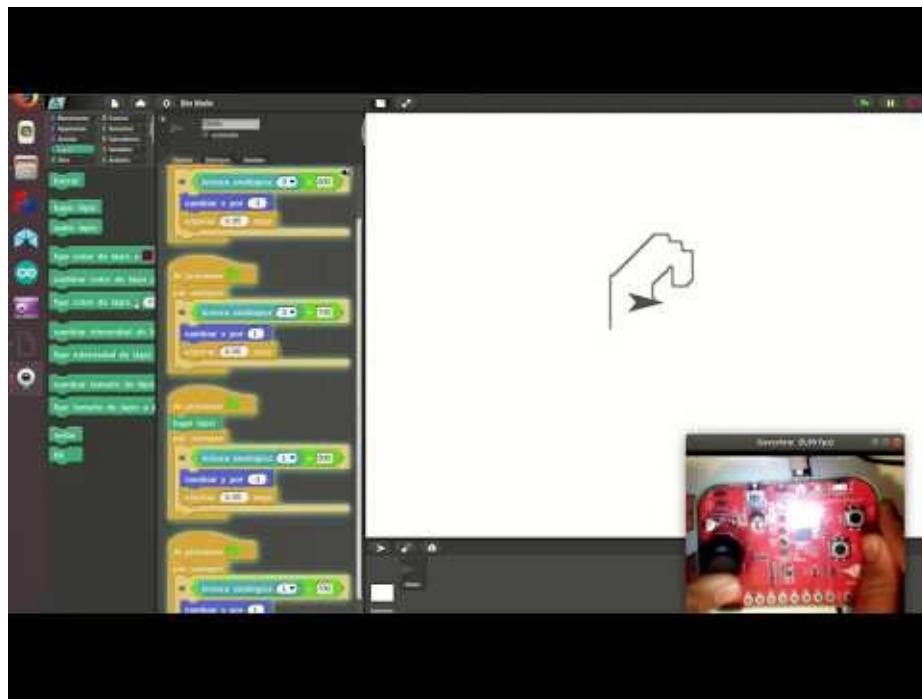
El Joystick está conectado a los pines A0 y A1 y su valor varía desde 100 hasta 1024.

*El pin A0 controla el eje X El pin A1 controla el eje Y*



## MONTAJE 7 TELESCKET

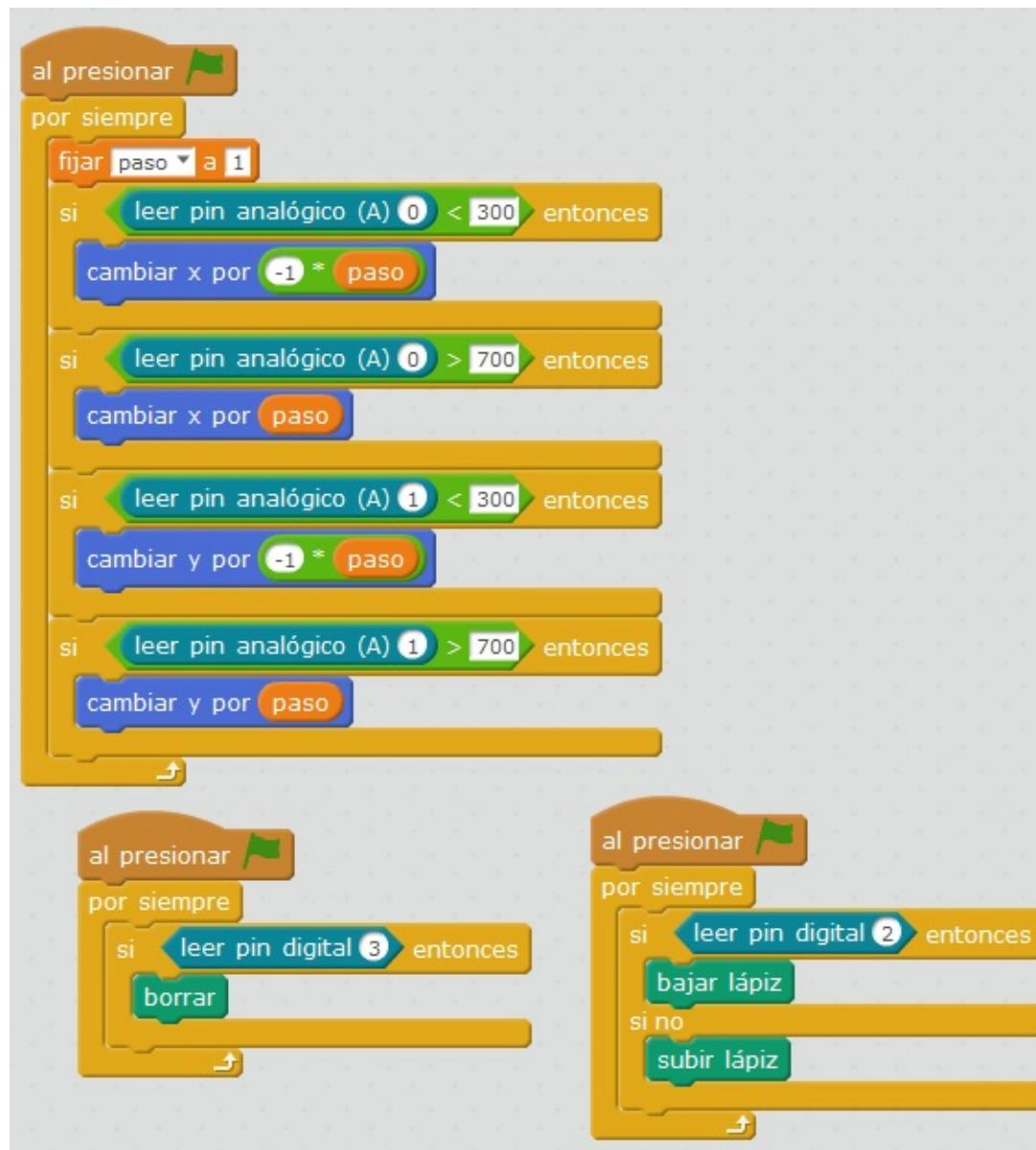
**RETO** Vamos a realizar el mismo ejemplo que [este vídeo de Jorge Lobo](#) pero en vez de realizado en Snap4Arduino lo vamos a hacer en mBlock.



[Video link](#)

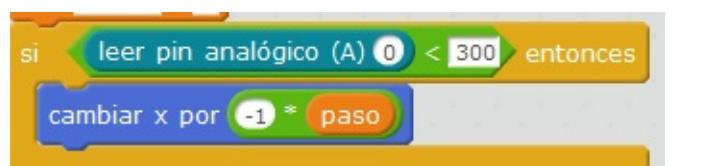
Solución

El programa :

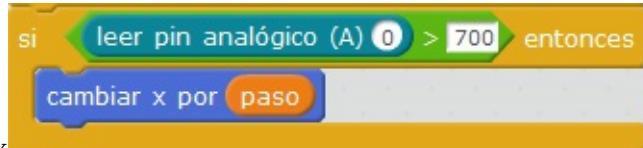


simplemente vamos a comentar las condicionales principales:

- Si A0 es menor que 300 es que estás inclinando el Joystick hacia la izquierda, luego el Sprite tiene que modificar su variable x hacia la derecha, es decir decrementando su valor

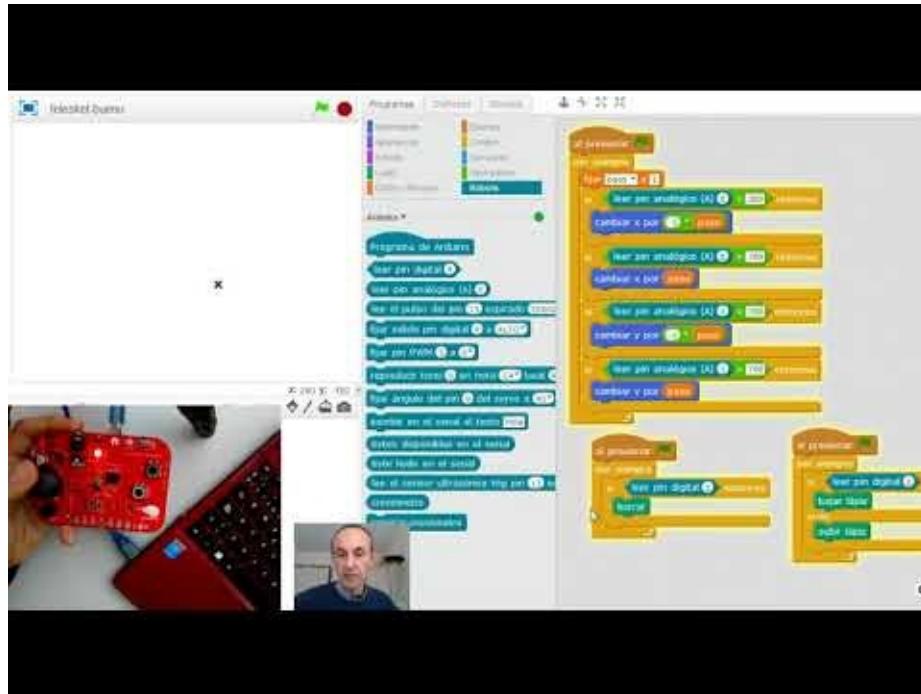


- Si A0 es mayor que 700 es seguro que estás inclinando el Joystick hacia la derecha luego hay que incrementar el valor de la ordenada



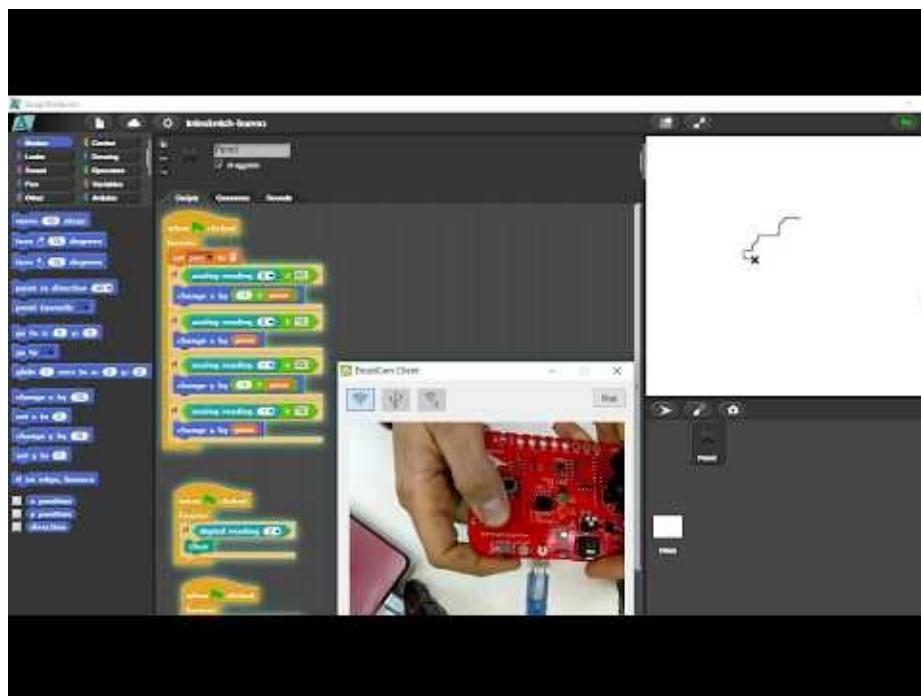
- Para los valores del eje Y es igual, cambiando A0 por A1

**VAMOS LENTO** El resultado lo puedes ver en [este video](#):



## Video link

Por curiosidad puedes ver el mismo programa en **Snap4Arduino** en este vídeo:



## Video link

Puedes ver que es mucho **más rápido** que mBlock.

## MONTAJE 8 COMEBICHOS

No podemos dejar el Joystick sin hacer un videojuego !!



El siguiente **RETO** es: *Mover un Sprite "Bat" con el Joystick para atrapar un Beetle*

- Bat se tiene que mover con el joystick por toda la pantalla.
- Beetle se mueve horizontalmente de el borde derecho hasta el borde izquierdo, y cuando llega al izquierdo, vuelve a aparecer en el derecho (y aleatoriamente desde cualquier altura)



Solución

El programa te lo puedes descargar [aquí](#)

El bicho lo haremos mover bastante rápido y al azar en el eje Y para que lo tenga difícil Bat:



Y bat tiene que moverse con el joystick



Los valores de  $m$  y  $n$  lo hemos calculado con la [HOJA DE CÁLCULO](#) que comentamos anteriormente, los valores límites que hemos puesto son:

Para el eje X e Y :0-715

HOJA DE CALCULO DE CAMBIO DE ESCALAS	
$x' = m x + n$	
Valores que tengo de $x$	
$x_1 =$	100
$x_2 =$	1024
Valores que quiero de $x'$	
$x'_1 =$	-220
$x'_2 =$	220
RESULTADOS	
$m =$	0,47619048
$n =$	-267,619048

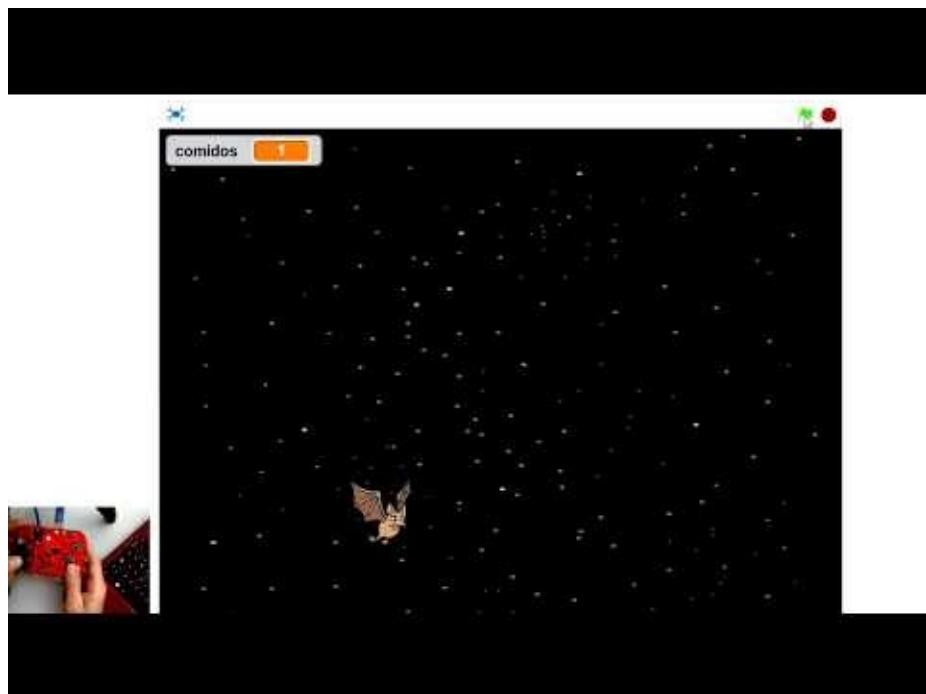
Para el eje Y:

HOJA DE CALCULO DE CAMBIO DE ESCALAS	
$x' = m x + n$	
Valores que tengo de $x$	
$x_1 =$	100
$x_2 =$	1024
Valores que quiero de $x'$	
$x'_1 =$	-150
$x'_2 =$	150
RESULTADOS	
$m =$	0,32467532
$n =$	-182,467532

Además hemos añadido la puntuación y el cambio de disfraz para que parezca que aletea:

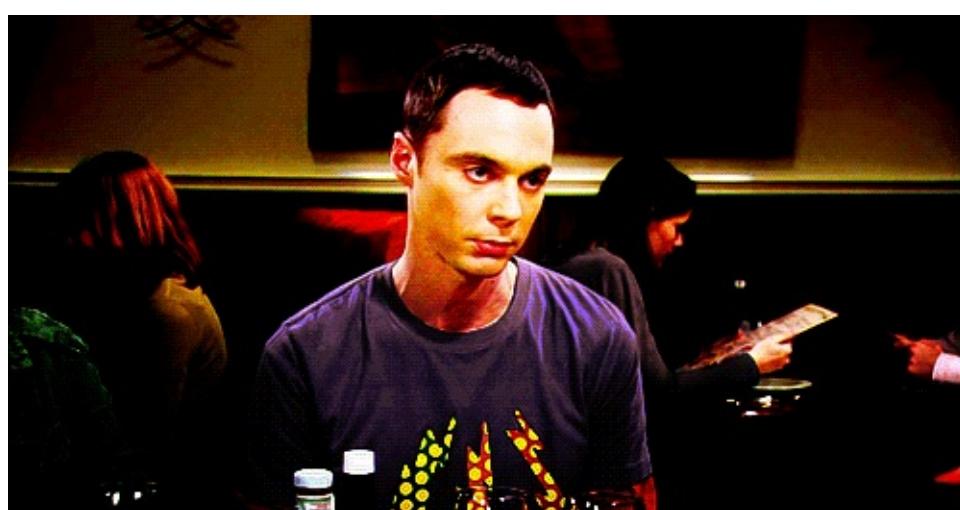


El resultado es:



[Video link](#)

Lo sé .... soy bastante malo !!





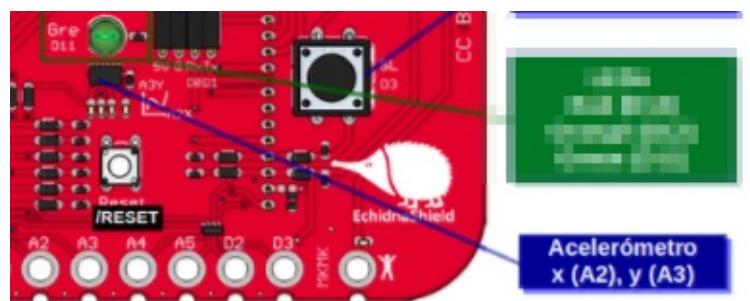
## 3.4 Acelerómetro

El acelerómetro tiene el mismo efecto que el Joystick, nos da valores en el eje X o eje Y, en el caso del Joystick lo daba la inclinación de la palanca, en este caso lo da la inclinación del mismo Echidna.

Está conectado a los pines analógicos:

- A2 nos da la inclinación en el eje X
- A3 nos da la inclinación en el eje Y

Los valores van desde 250 hasta 500



## MONTAJE 9 HELICOPTERO

### RETO

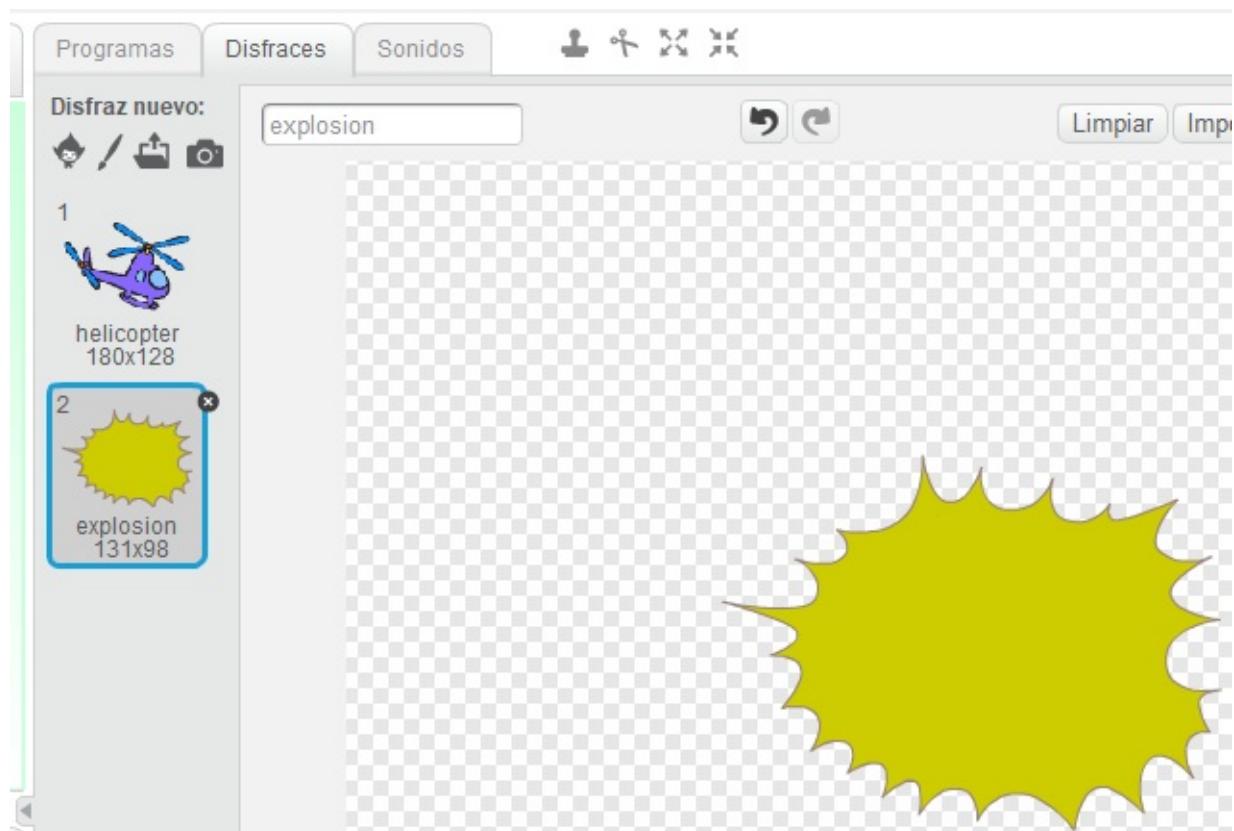
Vamos a realizar un sencillo videojuego: Mover un helicóptero, el movimiento de un sprite con el acelerómetro y esquivando edificios que se mueven de derecha a izquierda dando la sensación de que el helicóptero vuela.

El reto te lo complicamos con dos premisas:

- Reutiliza el "sprite" edificio copiándotelo desde Scratch
- El helicóptero explota si choca con un edificio

Solución sprite helicóptero moviéndose según acelerómetro

El sprite lo vamos a seleccionar de la biblioteca: Helicopter y le vamos a añadir otro disfraz que sea una explosión:



, la manera más sencilla es elegir otro disfraz de la biblioteca, por ejemplo sol, y borrar la cara feliz que tiene:



Vamos a hacer que se mueva según el acelerómetro:

Posición X: La pantalla de mBlock tiene desde -240 hasta 240 y hemos visto que los márgenes de A2 del acelerómetro va desde 250 hasta 500 luego hacemos el cambio de escala con la [HOJA DE CÁLCULO](#) que comentamos anteriormente:

<b>HOJA DE CALCULO DE CAMBIO DE ESCALA</b>	
$Y = m X + n$	CAMBIA LO AMA
Valores que tengo de x	
X1=	250
X2=	500
Valores que quiero de Y	
Y1=	-240
Y2=	240
RESULTADOS	
m =	1,92
n =	-720

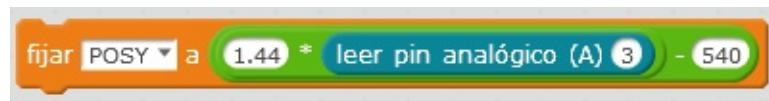
luego la fórmula para la posición X es (creando una variable que se llame POSX) :



E igualmente para la posición Y pero en este caso los márgenes de la pantalla van desde -180 a 180 luego:

<b>HOJA DE CALCULO DE CAMBIO DE ESCALA</b>	
$Y = m X + n$	CAMBIA LO AM
Valores que tengo de x	
X1=	250
X2=	500
Valores que quiero de Y	
Y1=	-180
Y2=	180
RESULTADOS	
m =	1,44
n =	-540

luego la fórmula para la posición Y es (creando una variable que se llame POSY):



El programa para el helicóptero es:

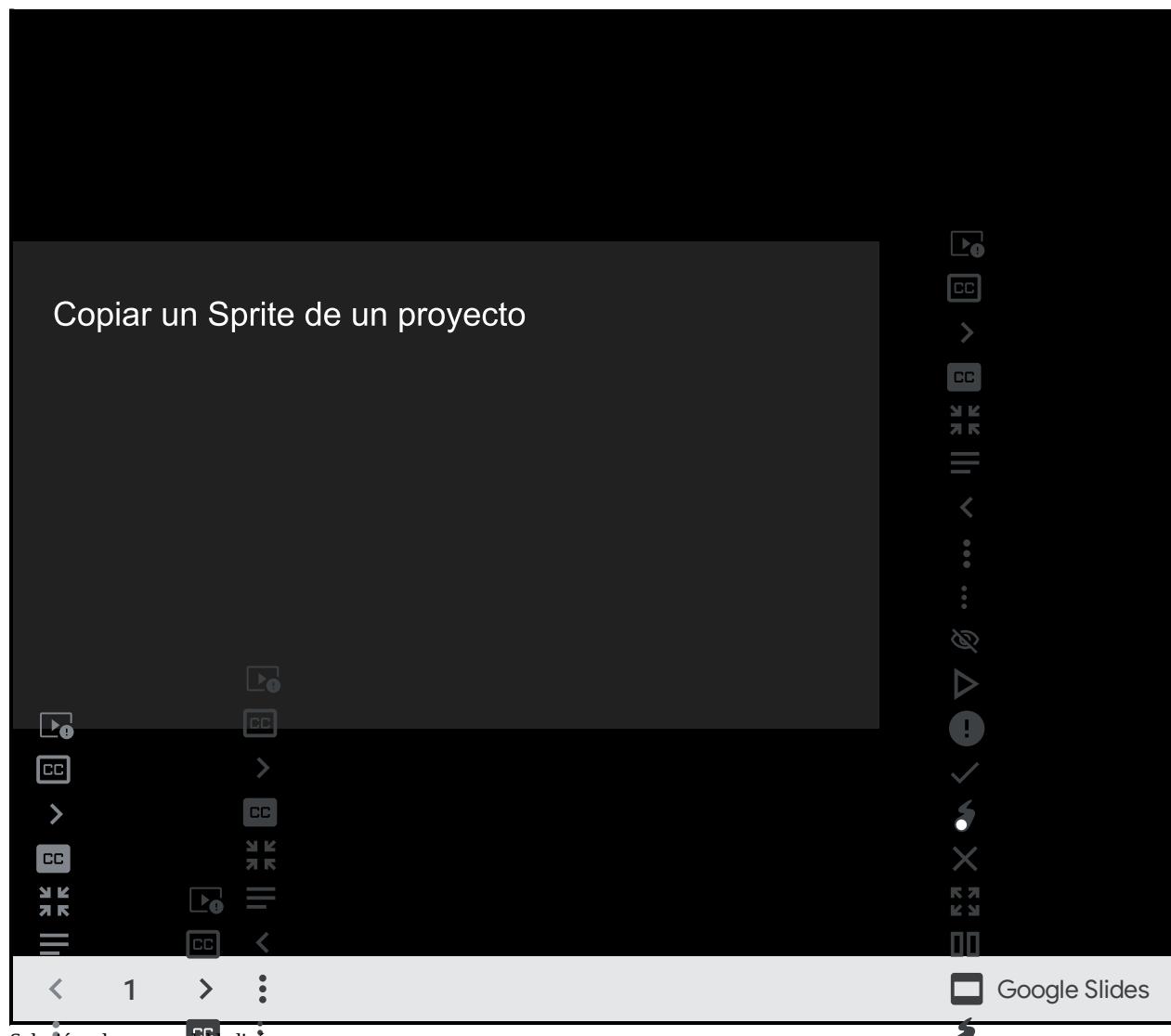


Solución edificio, cómo se reutiliza un sprite

Reutilizamos un poco: Nos copiamos un Sprite.

Vamos a copiar un poco el código de otra persona: Vamos a [scratch](#)

Y seguimos [estos pasos](#)



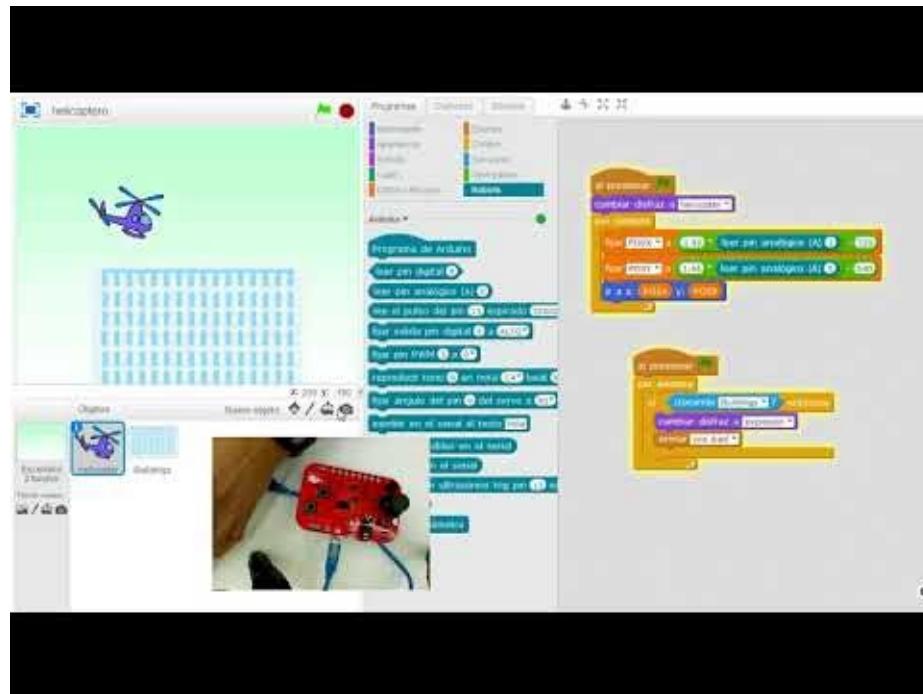
Solución a la muerte del helicóptero

Vamos a poner que si toca un edificio, “explota” (o sea, cambia el disfraz a explosión), además envía el mensaje “you died” que vemos que el script del edificio lo necesita para esconderse una vez terminado el juego:



El resultado se puede ver en este vídeo:





[Video link](#)

### MEJORAS

Ya sé que colocando el helicóptero siempre arriba, siempre ganas!!

[via GIPHY](#)

Te propongo que añadas además de los edificios, unos pájaros que tengan el mismo comportamiento pero que aleatoriamente aparezcan a diferente altura, para complicar un poco la vida del helicóptero.

### 3.4.2 MONTAJE 10 MATA-ALIENS

Esta vez no vamos a ser originales, vamos a ser un poco vagos y nos vamos a copiar el programa de otra persona. La [web de Scratch](#) tiene un repositorio de millones de proyectos, podemos seleccionar uno, y adaptarlo a nuestra simpática Echidna.

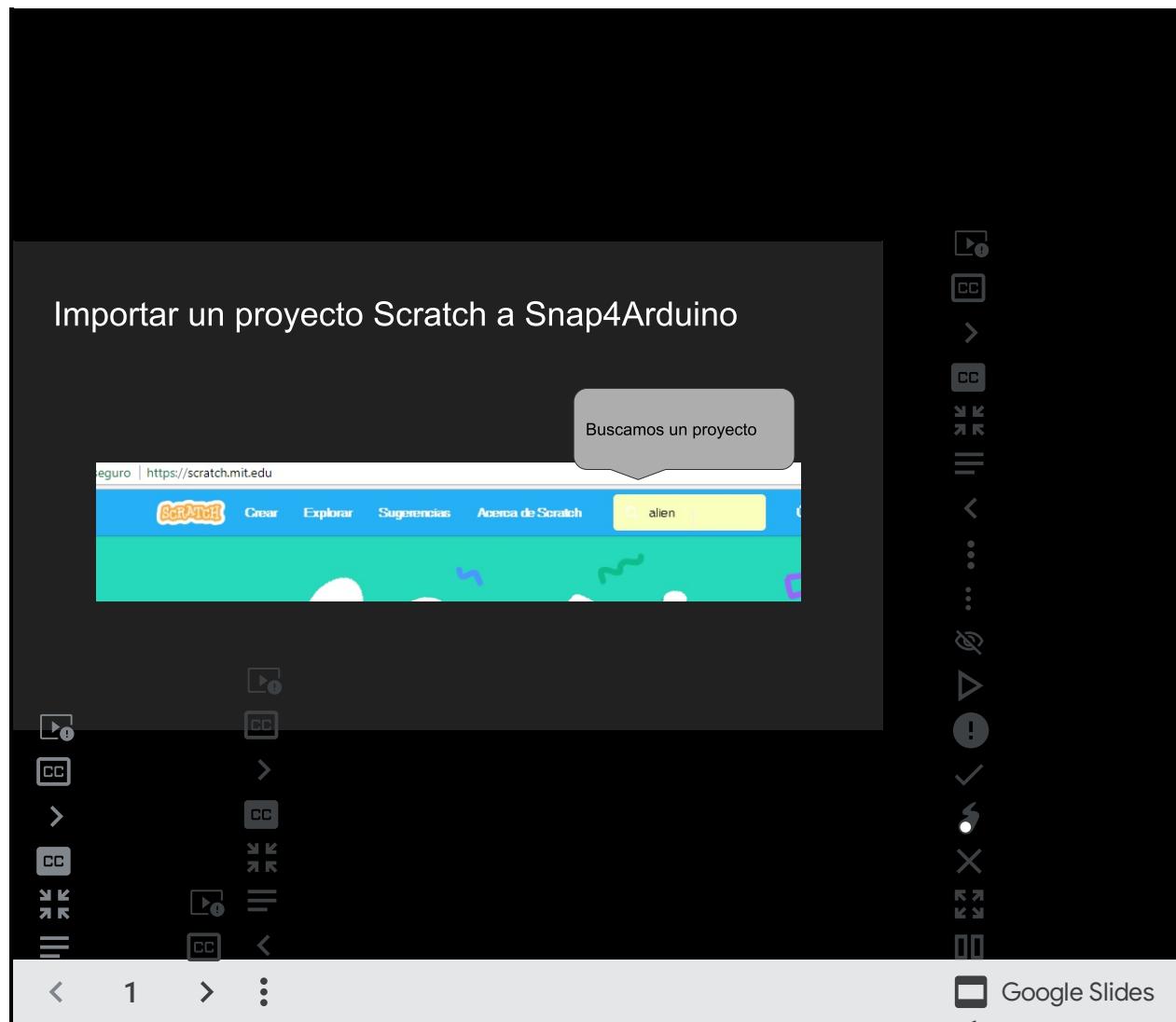


**RETO** Vamos a ver un ejemplo con este proyecto <https://scratch.mit.edu/projects/29744/>. El reto que te lanzamos es que funcione la nave espacial con nuestro acelerómetro de la Echidna.

#### SOLUCION CON MBLOCK

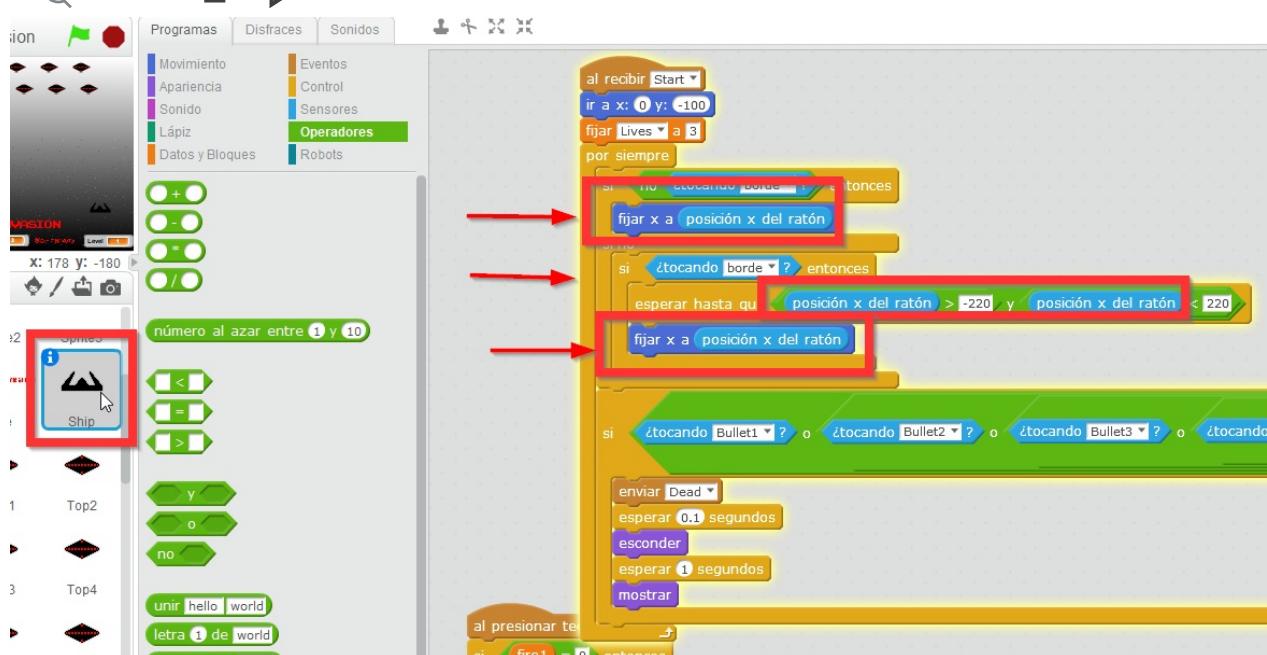
Importar un proyecto de Scracth a mBlock

En [esta presentación](#) se muestra cómo se importa un proyecto a mBlock:

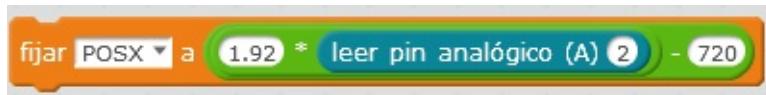


Altear el código para que funcione con el acelerómetro

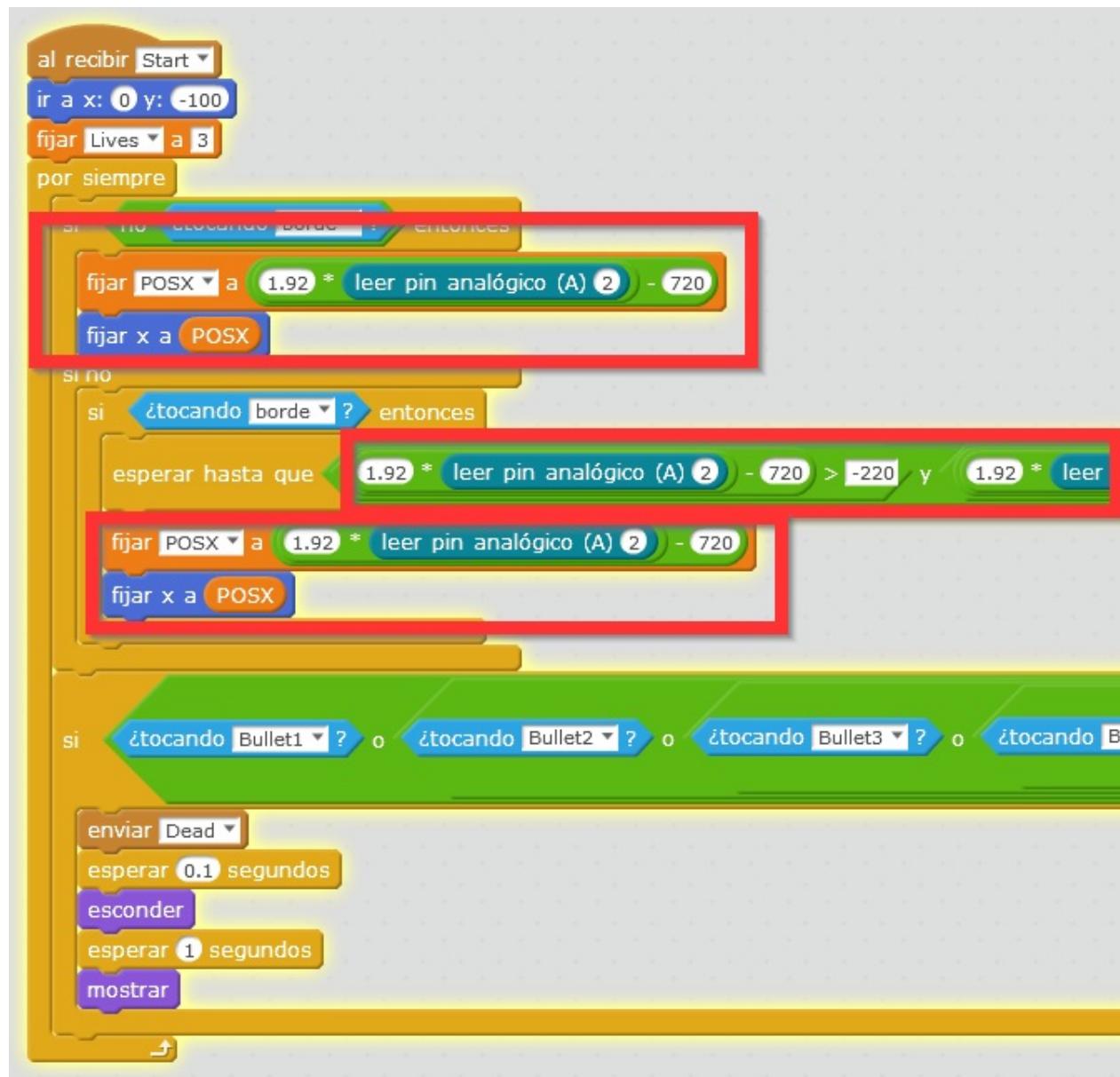
Una vez importando los fijamos en el código que hay que modificar, está claro que lo que hay que tocar es el Sprite "Ship" que es el que queremos mover con el acelerómetro y modificar el "posicionX" que se realiza con el ratón y cambiarlo por el acelerómetro:



Perooo como siempre: la posición del ratón es la misma que las coordenadas de la pantalla de mBloc y la x va desde -240 hasta 240 y nuestro acelerómetro de A2 va desde 250 hasta 500 luego hacemos el cambio de escala con la [HOJA DE CÁLCULO](#) anteriormente:



Y cambiamos el código:

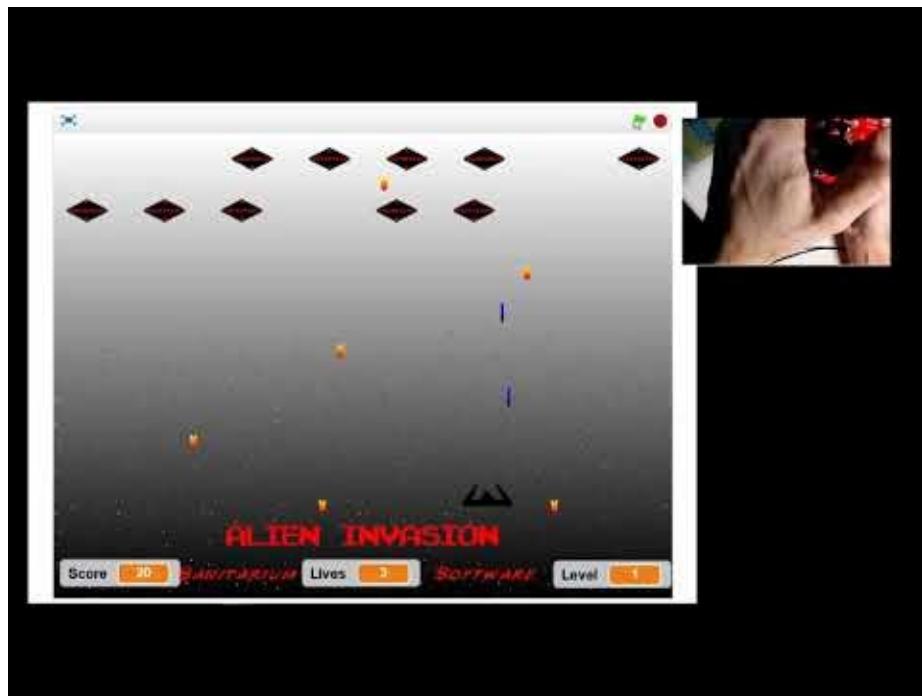


Un poco más, disparar con D2

También vamos a cambiar el disparador “espacio” por pin digital 2 para disparar con el botón D2 del Echidna:



El resultado en este vídeo:



[Video link](#)

### SOLUCIÓN CON SNAP4ARDUINO

Este programa vemos que va lento, **RECOMENDAMOS** que para la ejecución de proyectos grandes desde el ordenador Snap4Arduino va más rápido, por lo que vamos a ver esta alternativa:

Importar un proyecto de Scratch a Snap4Arduino

Para importar un proyecto desde Scratch es un poco más complicado, ver [esta presentación](#):

Importar un proyecto Scratch a Snap4Arduino

The slide shows a screenshot of a presentation. At the top, there is a dark header with the text "Importar un proyecto Scratch a Snap4Arduino". Below this is a screenshot of a web browser displaying the Scratch website (<https://scratch.mit.edu>). A search bar at the top right contains the word "alien". A speech bubble from the browser says "Buscamos un proyecto". The main content area shows a green stage with a white alien sprite. On the left, there is a vertical stack of various Scratch blocks. At the bottom, there is a navigation bar with arrows and a "Google Slides" icon.

Cambios en el código en Snap4Arduino

Los cambios en el código son los mismos que los anteriores.



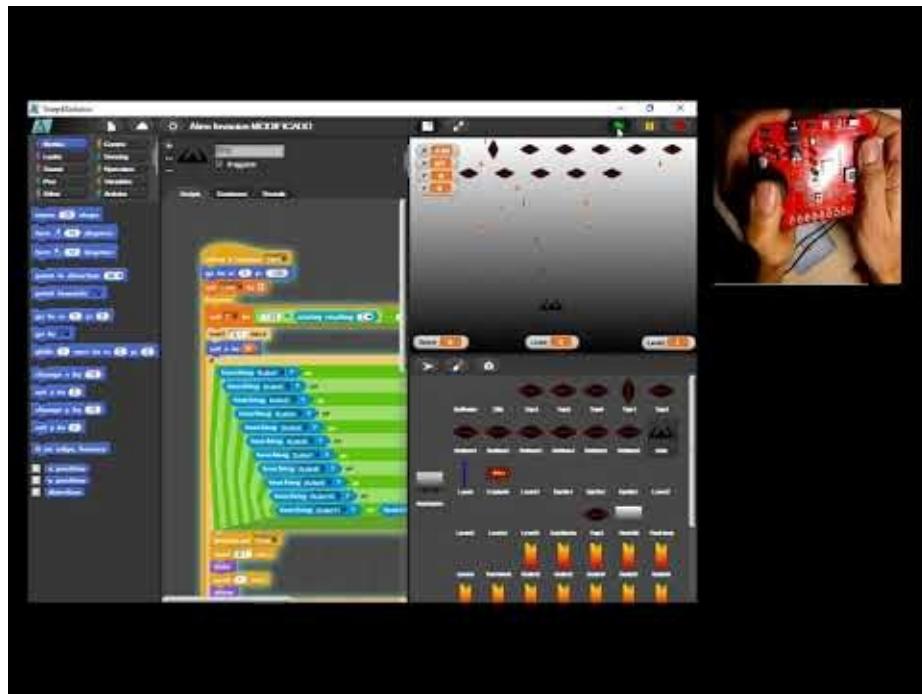
Recura que en [esta presentación](#) se enseña cómo configurar Snap4Arduino para que se comunique con nuestro Arduino y la Echidna.

El programa lo puedes abrir en Snap4Arduino en esta [URL](#)

El Sprite que tienes que conectar con Arduino es la nave esa con cuernos.

El resultado lo puedes ver en este video

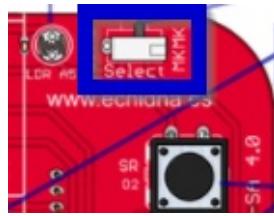




[Video link](#)

## 4 Modo Makey makey

En esa sección acuérdate de cambiar el Echidna en modo **MakeyMakey**



Todos los programas de este curso se encuentran en este repositorio: <https://github.com/JavierQuintana/Echidna>

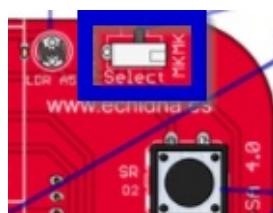
## 4.1 ¿Qué es?

Es una placa que se inventó en 2010 que permite conectar cualquier cosa que conduzca mínimamente la electricidad en entrada por el puerto USB. Para que lo entiendas bien, mira este vídeo:



[Video link](#)

Nuestra placa **tiene esta función**, pero en vez de ser una entrada por el puerto USB es una entrada por Arduino que podemos leer, luego en esta sección cambia nuestra placa Echidna el modo sensor a modo makey makey



Tiene las siguientes entradas:



- **ANALOGICAS**
  - Desde A0 hasta A5 que tenemos que detectarlas si superan un cierto “umbral” que dependerá de lo que conectemos.
- **DIGITALES**
  - D2 y D3 aquí es fácil, 0 o 1
- **GND** o sea el conector que tiene que conectarse a una parte de tu cuerpo para crear el puente eléctrico.

Entonces... ¿qué ventajas tiene esta placa con respecto a la original?:

- Más **barata** (importante)
- **2x1** sí, sí, piénsalo: Tienes en la misma placa una Shield educativa con sensores y a la vez una Makey Makey,

- **Autonomía:** Makey necesita un ordenador y nuestra Echidna necesita Arduino ¿quien gana en simplicidad? si hacemos que nuestro programa se graba al Arduino: *¡La portabilidad es total!!*

*Nota:*

En [Aularagón](#) tienes un curso de manejo de la Makey Makey original con el curso Scratch avanzado y Makey Makey

## 4.2 MONTAJE 11 Piano con bananas

Vamos a hacerlo !! Ten en cuenta que mBlock utiliza la notación americana de notas, y a cada nota la vamos a corresponder a una entrada Makey Makey (utilizando por ejemplo el acorde 4):

Nota americana	C4	D4	E4	F4	G4	A4	B4	C5
Nota Europea	Do	Re	Mi	Fa	Sol	La	Si	Do
Makey	A0	A1	A2	A3	A4	A5	D2	D3

Como las entradas analógicas pueden variar desde 0 a 1024 vamos a fijar como umbral en 500 (si es muy sensible lo subimos, si no nos hace caso lo bajamos, a nosotros nos ha funcionado bien 500).

El código lo subiremos al Arduino pues mBlock responde muy lentamente, si no te acuerdas cómo se hacía mira esta [presentación](#).

Solución

El programa es muy sencillo pero repetitivo:



Conectamos cada salida de Makey Makey a unos plátanos y la salida del auricular a unos altavoces de ordenador, en la foto puedes ver que se han utilizado unos cables con cocodrilos, y al subirlo al Arduino no hace falta tener el ordenador, alimentamos el Arduino con un PowerBank o incluso con pilas.

**ATENCIÓN VER 1.3 ALIMENTACION DE LECHIDNA**

En el kit que te prestamos para hacer este curso no están incluidos los plátanos, así que cálzate y baja a comprar unos. ¿no tienes perras? pues utiliza recursos humanos:



[Video link](#)



[Video link](#)

Esta noche prepara un karaoke con tus amigos.



## 4.3 MONTAJE 12 Plastilina

Con plastilina conductora podemos hacer interesantes proyectos. ¿Cómo se hace la plastilina conductora? Aquí tienes una receta:

<http://www.comofuncionainternet.net/circuitos-con-plastilina/>

Una pega: No dura nada, a los pocos días se endurece o se pudre.

Podemos hacer interesantes proyectos, por ejemplo:

### RETO

Que un helicóptero se mueva según unos botones hechos de plastilina:

- Botón con forma de ⇒
- Botón en forma de ↑
- Botón en forma de ↓
- Botón en forma de ←

### Solución



- A3 conectado a la plastilina con forma de ⇒
- A1 conectado con la plastilina en forma de ↑
- A5 conectado con la plastilina en forma de ↓
- A2 conectado con la plastilina en forma de ←

El umbral se ha cambiado según la sensibilidad del botón.

El resultado en este vídeo:

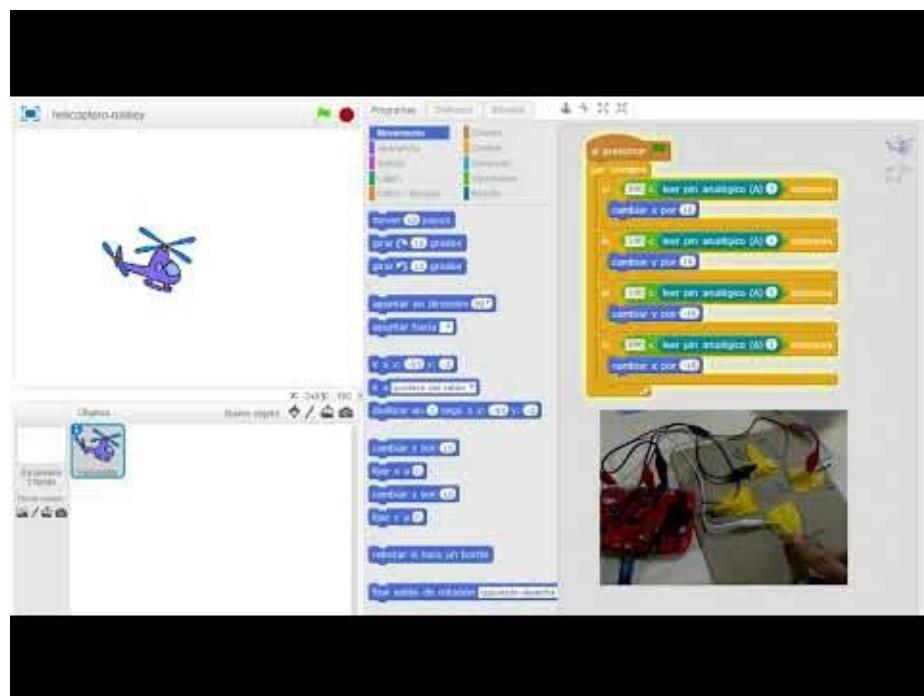


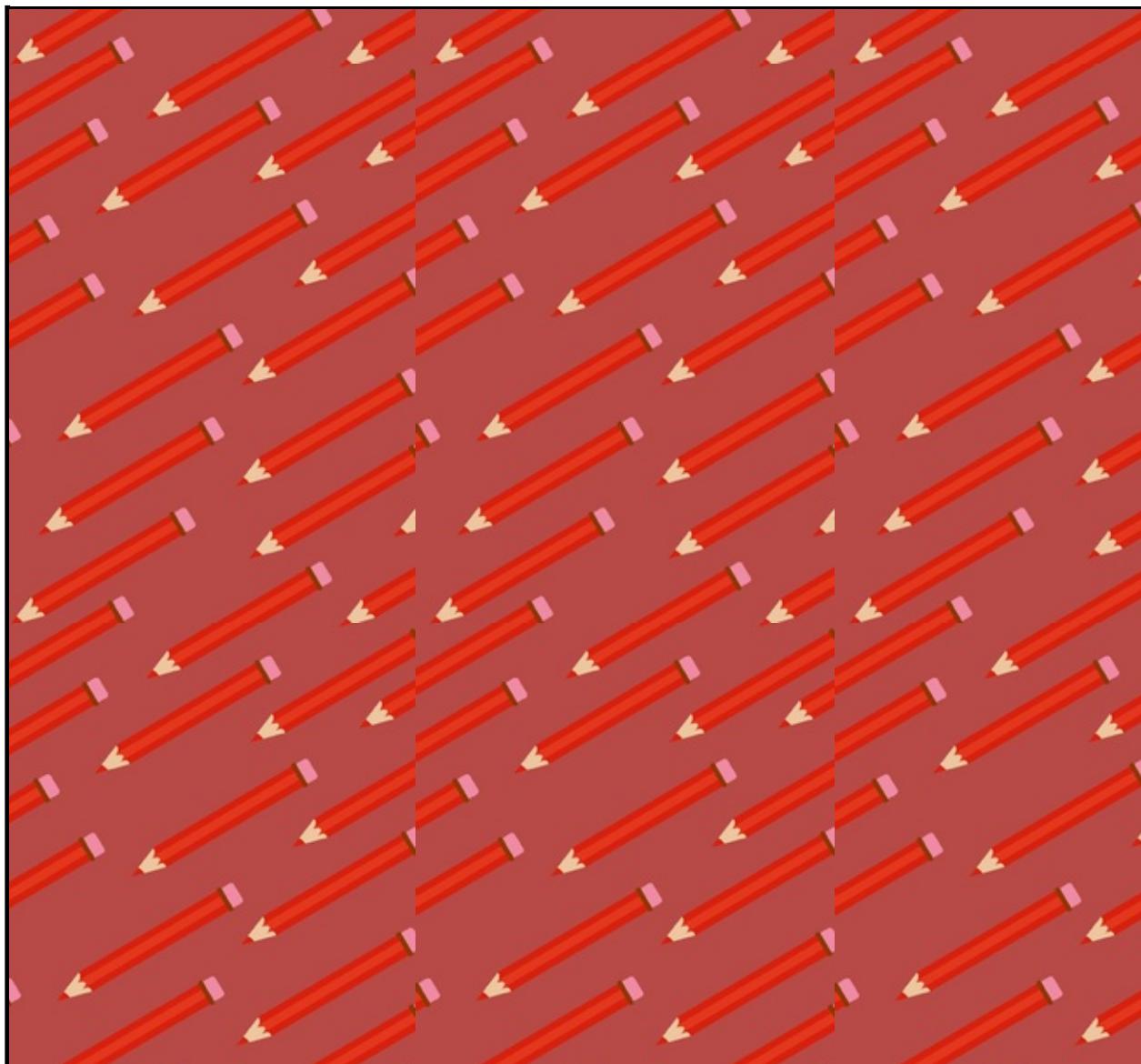
Imagen - YouTube video thumbnail

[Video link](#)

**OTROS RETOS** Con la plastelina podemos hacer manualidades tipo conecta y que por la pantalla salga el resultado, [aquí tienes un ejemplo](#):



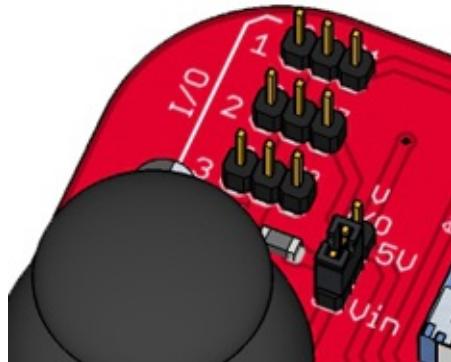
Las posibilidades son muchas, cuelga tus experiencias en este [muro](#)



Made with **padlet**

## 5 Extensiones

Ahora vamos a utizar las extensiones 3 digitales y una analógica:



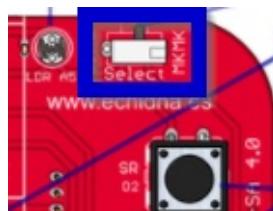
Para conexión de otros elementos como relés pero no pueden pasar de 300mA de lo contrario dañas al Arduino.

- Entrada analógica A5
- Salidas/Entradas digitales D4, D7 y D8

La alimentación de estas I/O se puede elegir entre la salida de 5V por el Arduino y Vin con la tensión que estemos alimentando al Arduino utilizando el jumper que ya vimos [en su momento](#).

---

Nota: Acuerdate en toda esta sección de poner la Echidna en modo Sensor



Todos los programas de este curso se encuentran en este repositorio: <https://github.com/JavierQuintana/Echidna>

---

Ahora con las extensiones puedes aumentar más tus proyectos robóticos. Por ejemplo en este vídeo el Echidna se ha colocado en un chasis con unos motores y controlado por el Bluetooth del móvil. Este ejemplo del robot-coche no lo trataremos aquí pues es de un nivel superior, que se trata mejor en [Arduino con código](#)



[Video link](#)

Pero sólo está el límite de tu imaginación para convertir al Echidna el algo más poderoso ...

## 5.1 MONTAJE 13 ENCENDER CON EL MÓVIL Bluetooth

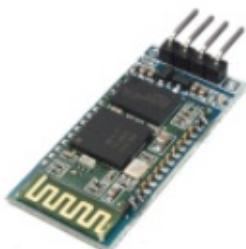
### ATENCIÓN: MUY DIFÍCIL

#### 5.1.1 Módulo HC-06

Echidna tiene un conector preparado para conectar un módulo de Bluetooth



Nosotros utilizaremos un JY-MCU o **HC-06** muy común y barato. .

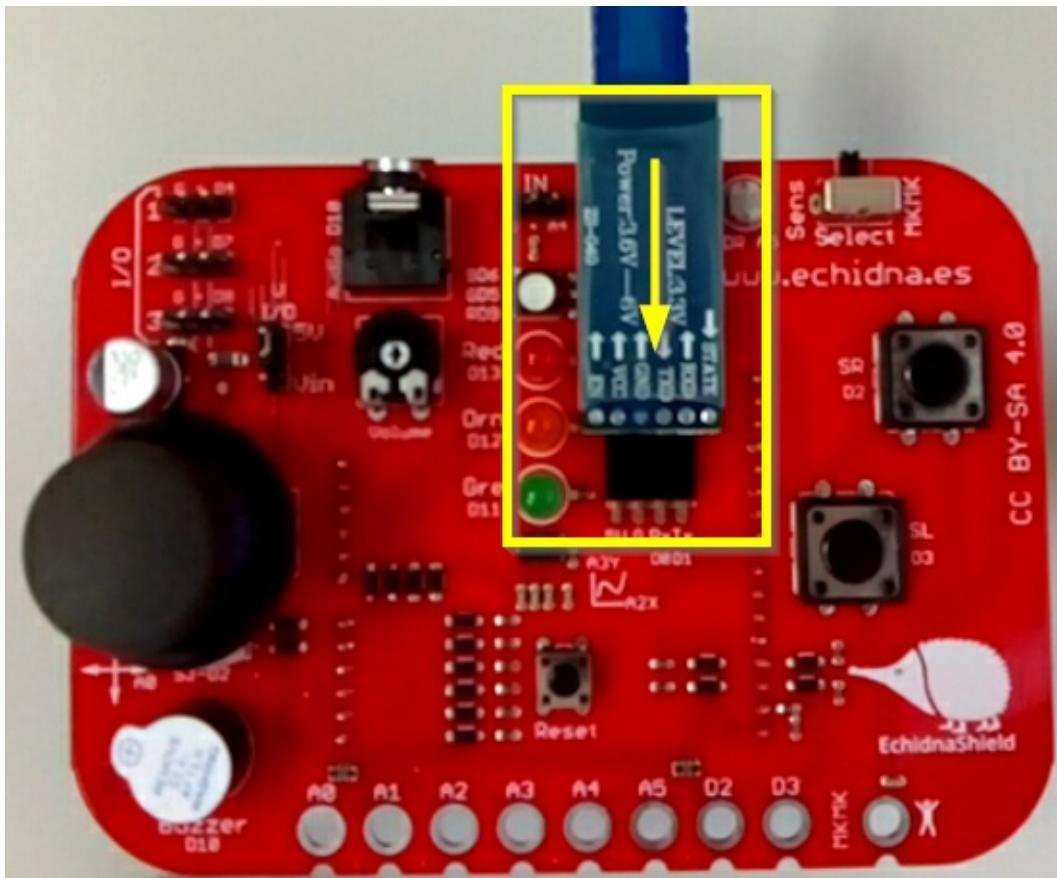


Te recomendamos estas páginas:

- [Teoría de Bluetooth](#)
- [Cómo se comunica con un Arduino](#)

Para conectar el HC-06 lo hacemos hacia abajo de modo que coincida los pines:

Pines del HC-06	Pines del Echidna	Pines del Arduino
Vcc	5V	5V
GND	GND	GND
RX	TX	D1
TX	RX	D0



Nosotros vamos a utilizar la APP BlueControl:

- ver cómo funciona
- ver cómo se vincula con el móvil



### 5.1.2 Problema número 1: ocupamos el puerto serie

Si has leído [Cómo se comunica con un Arduino](#) habrás visto que ocupamos LOS MISMOS PINES D0 Y D1 QUE UTILIZA EL ARDUINO PARA COMUNICARSE POR EL PUERTO SERIE CON EL ORDENADOR.

Esto crea un problema: No se puede tener conectado el HC-06 mientras nos comunicamos el ordenador con el Arduino.

Solución: **Pues quítalo**, y luego cuando acabes de descargar el programa en el Arduino, **pues lo pones**.

Bah!! ¿sólo era eso? pues no, que nos ocupe el puerto serie nos fastidia: ¿puedes interaccionar con el Sprite? por ejemplo ¿puedes hacer que el oso panda de mBlock se mueva según el mando de BlueControl?...**NO**

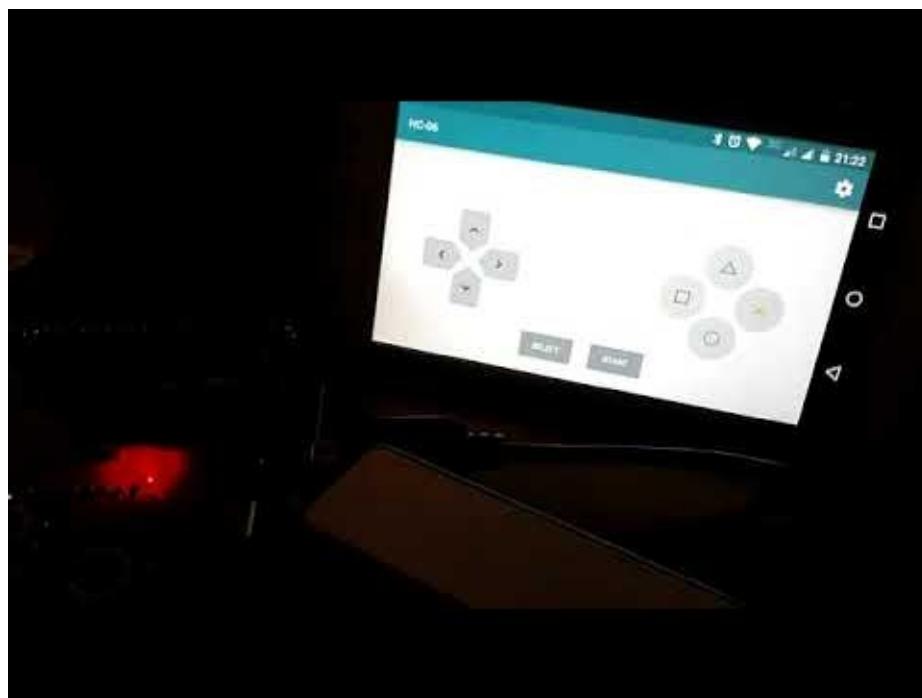
### 5.1.3 Problema 2 la velocidad del puerto es elevada

Al programar con mBlock fija la velocidad del puerto serie a 115200 baudios, y nuestro HC-06 soporta 9600

Solución: Bajarlo manualmente, **un rollo**, tenemos que salir de mBlock, editarlo en Arduino IDE .... esto se ve mejor en el ejemplo siguiente.

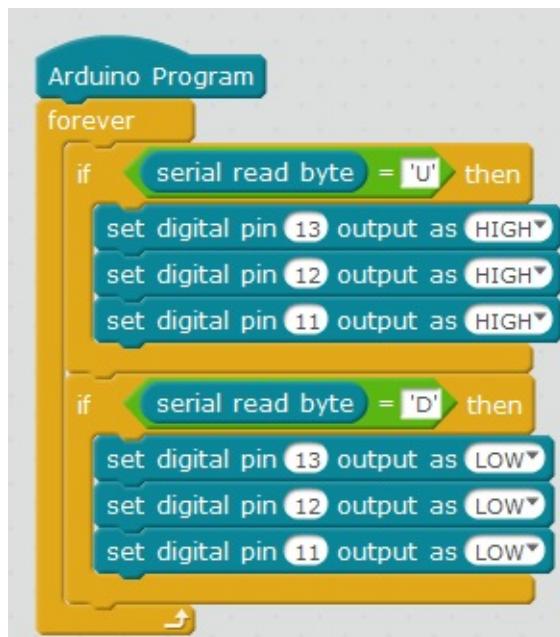
### 5.1.4 Reto: Encendido y apagado de LEDs con el móvil

Vamos a ejecutar este pequeño programa, que al apretar el botón de arriba se encienden los leds y al apretar el de abajo se apagan:



[Video link](#)

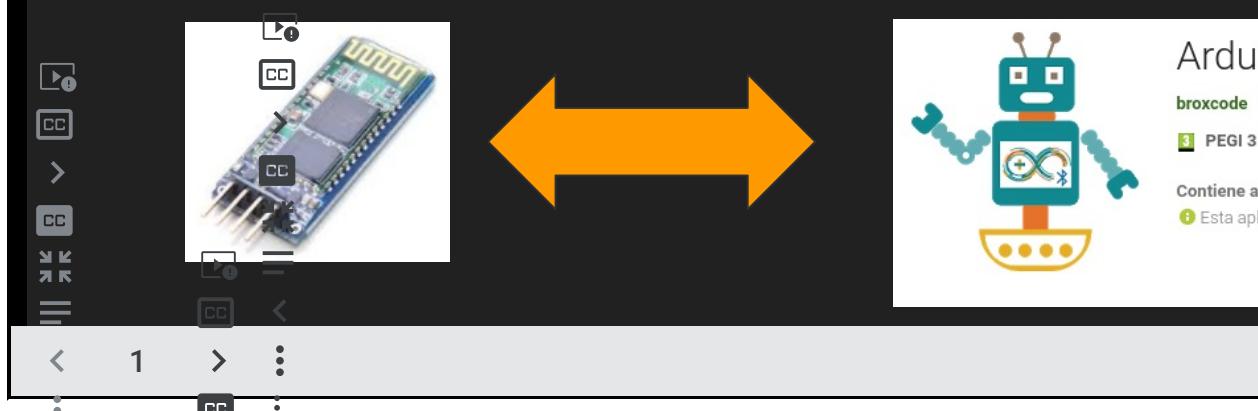
Solución



**Primero hay que vincular el móvil con la APP**

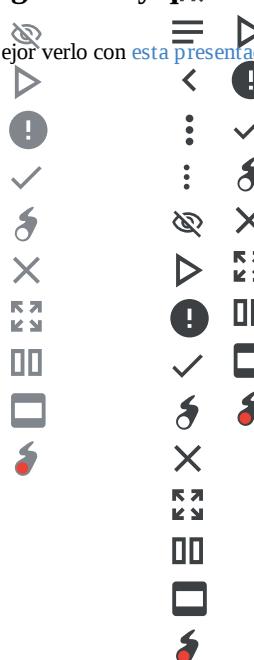
# ¿Cómo emparejar el Bluetooth del Arduino móvil y la APP

Utilizamos el bluetooth JY-



Segundo hay que seguir el programa solucionando los problemas anteriores

Mejor verlo con [esta presentación](#):

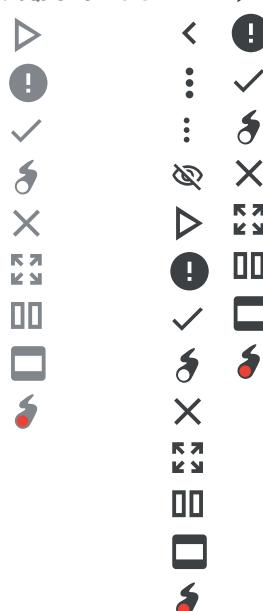


# Encendido de leds por bluetooth

Arduino y Echidna

O sea, ya podemos jugar con el móvil y con nuestro Echidna !!!

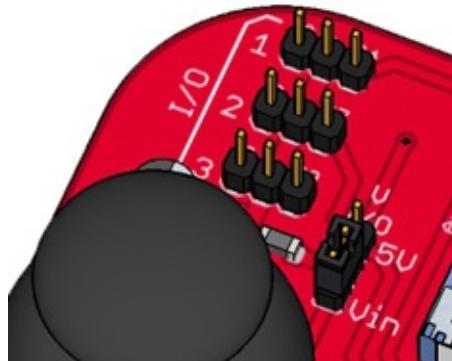
Por ejemplo.. se podría hacer un coche teledirigido, el programa lo tienes en el repositorio: <https://github.com/JavierQuintana/Echidna> y el video en el [muro](#)



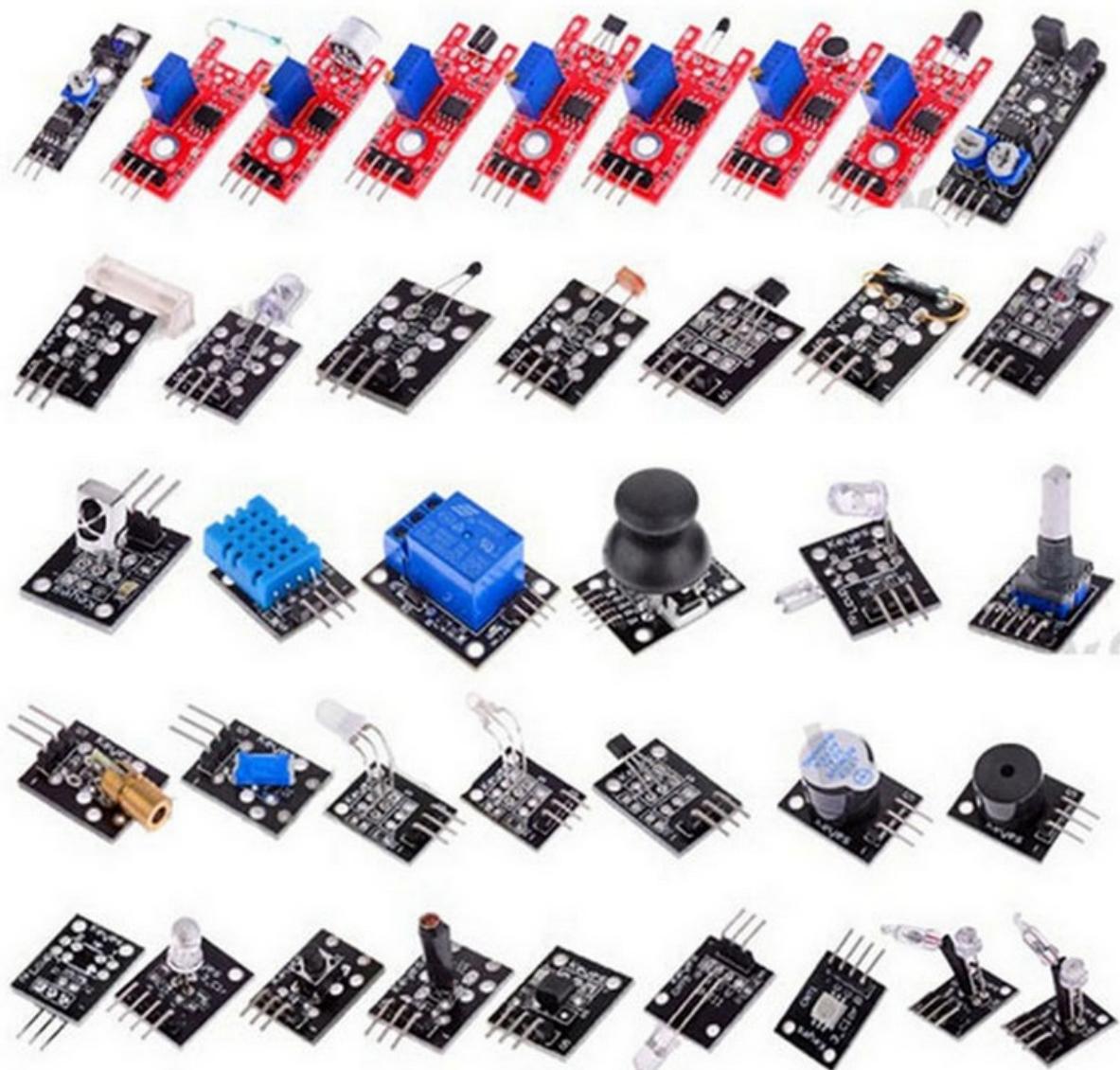
Google Slides

## 5.2 MONTAJE 14 ENCENDIDO SENSIBLE Sensor INFRAROJOS

Vamos a provechar las conexiones digitales que tiene Echidna preparado para conectar módulos exteriores, si te fijas está preparado para conectar fácilmente estos módulos pues facilita la alimentación, GND y el pin digital



Hay muchos módulos para conectar, y todos tienen la misma configuración en sus pines : GND, + , I/O, donde I/O es el pin digital o analógico de entrada o salida, dependiendo del sensor, y como puedes ver en la figura, hay mucha variedad (busca en Internet sensores para Arduino)

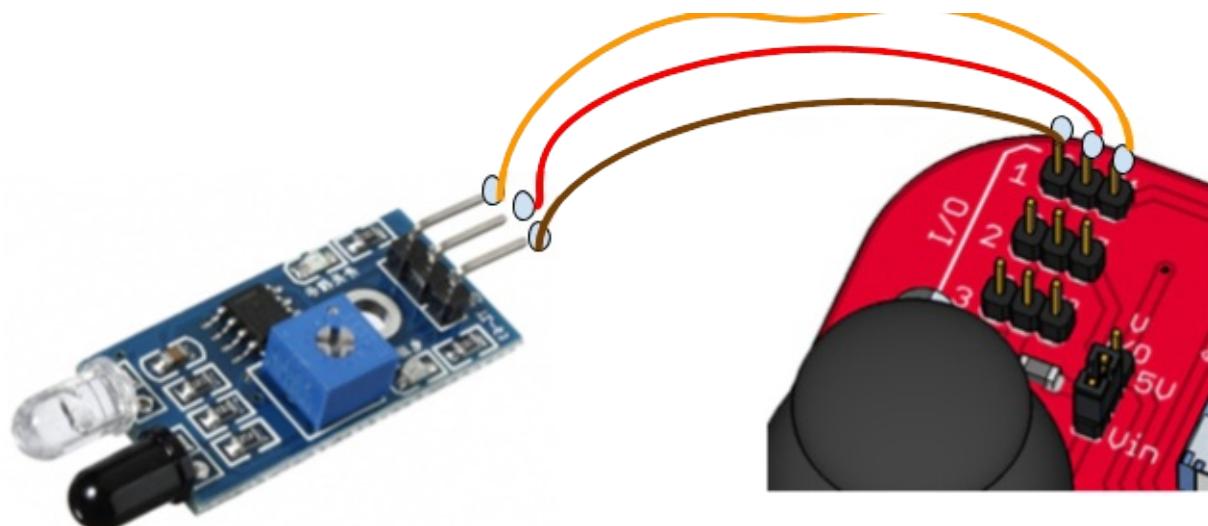


hay para a empezar con uno sencillo de entrada digital que es muy útil: el sensor Infrarrojos



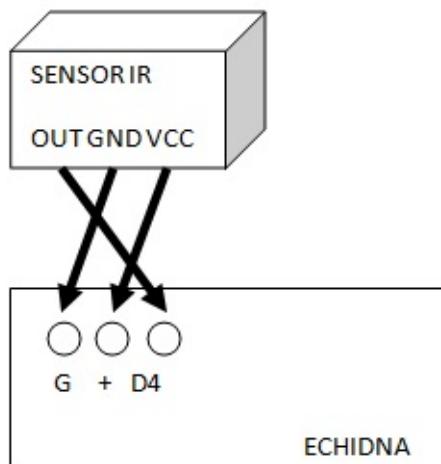
Para ver más información de este sensor te recomendamos [esta página](#).

En el Echidna arriba a la izquierda tienes 3 conectores digitales a elegir, nosotros elegimos el primero D4 luego el pin OUT del sensor se conecta al D4, el resto en el mismo orden



Nota: El potenciómetro es para ajustar la sensibilidad

**OJO Hay sensores que tiene los pines en otro orden FIJATE de lo contrario el sensor se pondrá a arder**

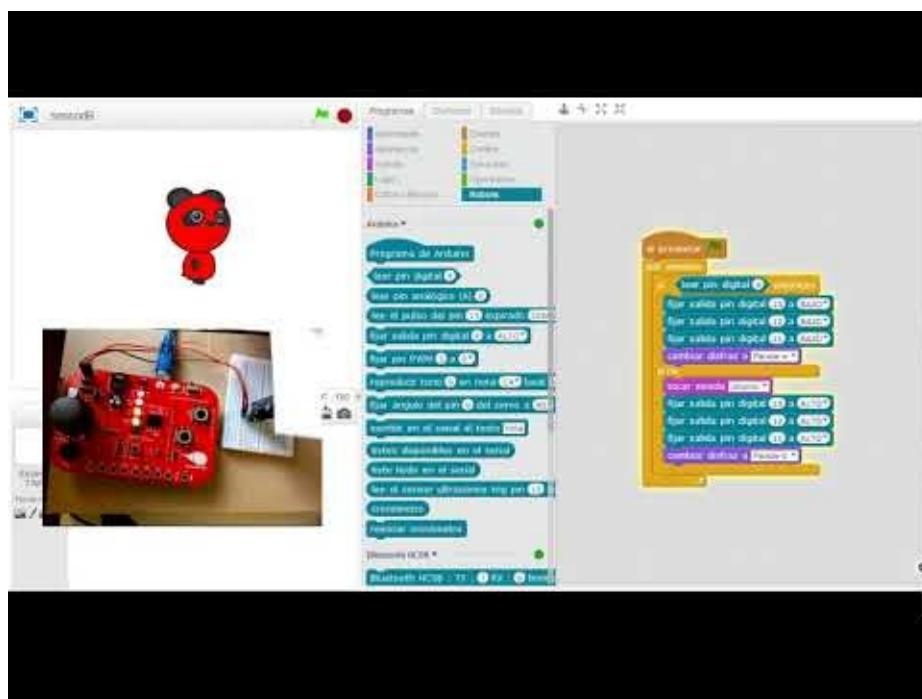


Nota: Hay sensores con 4 pines, que permiten alimentar el diodo IR de forma independiente, pero tienen un jumper que inutiliza el 4 para sólo utilizar los 3 pines.

Realizamos un pequeño programa muy fácil que detecta si hay un obstáculo o no:



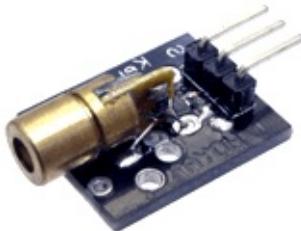
El resultado es:



[Video link](#)

## 5.3. MONTAJE 15 ALARMA LASER

Hemos visto anteriormente una entrada digital sencilla con el sensor IR, ahora vamos a ver una salida digital muy simple, buscando siempre la motivación en nuestros proyectos: El diodo Laser.



Si quieras saber más de este componente, te recomendamos la página de [Luis Llamas](#). Si tienes que comprar uno, te recomendamos que no sea superior a 5mW, pues puede dañar permanentemente la retina del ojo [[+info](#)]. El modelo que te proponemos es de **1mW**, no obstante, **EVITA QUE EL LÁSER APUNTE A LOS OJOS** especialmente con niños.

**RETO** Te proponemos que realices un programa para desactivar una **ALARMA LASER**.

- La alarma está protegida por un haz laser (que enfoca a nuestro querido [LDR del Echidna](#)), si se corta el haz la alarma suena.
- El disparo va a ser simulada en el ordenador, no queremos queremos que alertar a los vecinos, un sonido en el ordenador y en la pantalla una imagen en el ordenador.
- Para desactivar la alarma, pulsamos la tecla espacio, y nos pide introducir el código (tienes 10 segundos para darle más entusiasmo, sino se dispara), si has acertado, el laser se apaga y ya puedes entrar a tu lindo hogar.

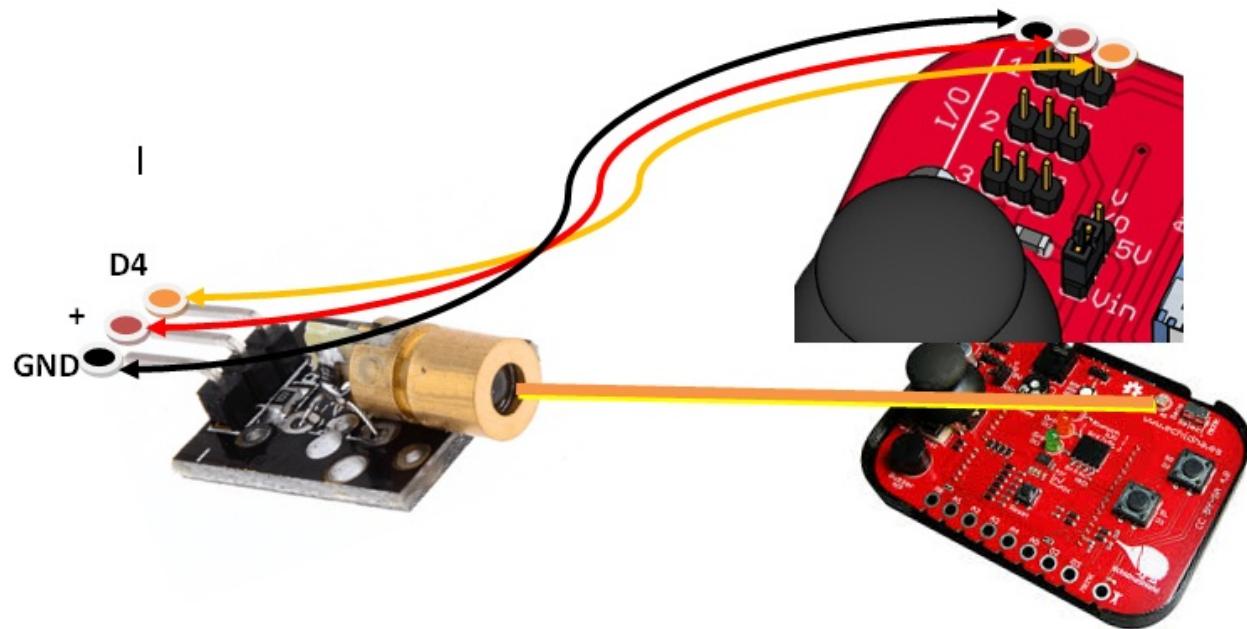
Luego vamos a no ponernos nerviosos para desactivar la alarma y mantener la serenidad.



[via GIPHY](#)

**SOLUCION**

**Conexiones** son sencillas, el laser al D4 por ejemplo y hay que tener maña para que apunte al LDR:



**El script que controla si se corta el haz laser que se dispare** es un condicional que lee el LDR en A5, mientras sea mayor que 900 es que tiene una intensidad de luz muy grande, es decir, le está enfocando el laser, la variable \_PARACRONO \_es en el caso de que se desactive la alarma, el laser se apagará luego que no se crea que se corta el haz. Si se corta el haz envía un mensaje \_Booom \_que lo leeran los demás scripts. Se aprovecha este script para ENCENDER EL LASER pin digital 4 alto:



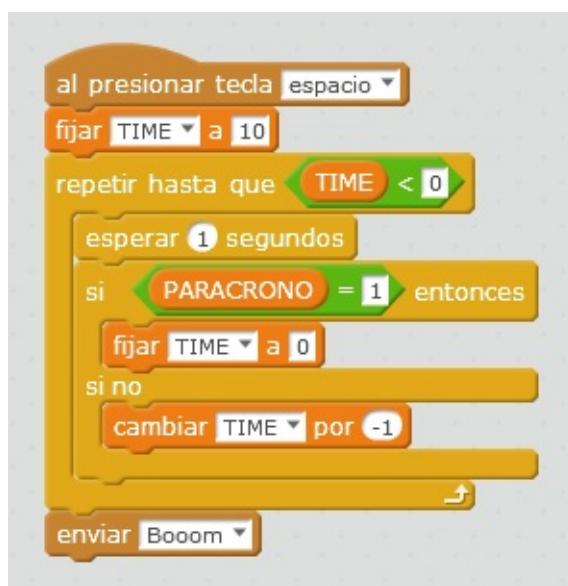
**El script que controla la desactivación de la alarma mediante código** es el siguiente. Si se acierta con el código secreto 666, se envía el mensaje \_Uffff \_al resto de scripts:



**El script que desactiva la alarma** es cuando reciba el mensaje \_Ufff \_por lo tanto para el cronómetro y apaga el laser:

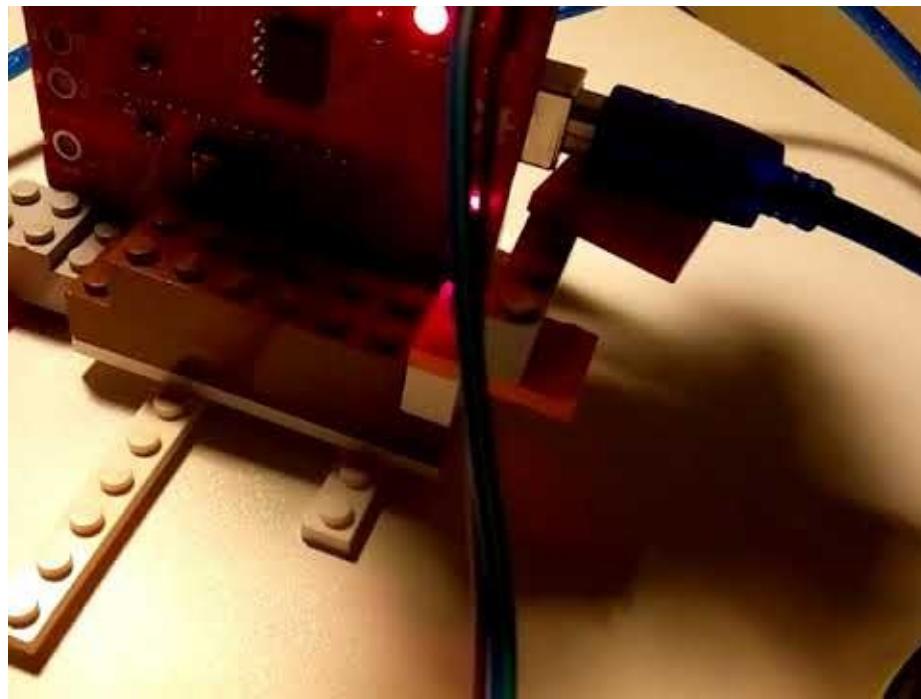


**El script del tiempo** cuenta de forma descendente, y si finaliza envía el mensaje Boom al resto de scripts, sólo se para si PARACRONO=1



El resto: esconder y mostrar scripts y disfraces se omiten por simplicidad, el programa lo puedes descargar desde el repositorio.

El resultado es



[Video link](#)

## OTRO RETO

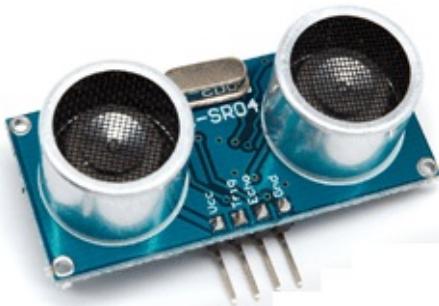
Si se corta el haz, hay 10 segundos para desactivarlo con código, sino, se dispara. No ponemos la solución ¡¡ponlo tú en [el muro !!](#)

Seguro que se te ocurren muchas ideas...



## 5.4 MONTAJE 16 SEMAFORO DISTANCIA Ultrasonidos

Queremos poner un ejemplo de un sensor que tenga 4 pines, barato y que puede darnos mucho juego pues nos da información de la distancia en la que se encuentra un objeto.



Funciona por eco entre la señal que se emite por Trg y la que se recibe por Echo y para su utilización requiere utilizar la fórmula de conversión de tiempo a espacio con la fórmula de la velocidad del sonido... tranqui !! no lo vamos a hacer, pues ya mBlock tiene una función especial para ello sin utilizar fórmulas y **nos da directamente la distancia en cm**, pero si quieres saber más sobre este sensor, te recomendamos la página de [Luis Llamas](#):

The screenshot shows the mBlock 3 interface with the following details:

- Top Bar:** Shows tabs for "Programas", "Disfraces", "Sonidos", and "Robots". The "Robots" tab is highlighted with a blue background.
- Left Sidebar:** Shows categories: Movimiento, Apariencia, Sonido, Lápiz, Datos y Bloques, Eventos, Control, Sensores, Operadores, and Robots.
- Arduino Section:** Shows "Arduino ▾" and "Programa de Arduino".
- Script Area:** Displays the following sequence of blocks:
  - leer pin digital 9
  - leer pin analógico (A) 0
  - lee el pulso del pin 13 expirado 20000
  - fijar salida pin digital 9 a ALTO
  - fijar pin PWM 5 a 0
  - reproducir tono 9 en nota C4 beat
  - fijar ángulo del pin 9 del servo a 90
  - escribir en el serial el texto hola
  - bytes disponibles en el serial
  - highlighted block:** lee el sensor ultrasónico trig pin 13
  - cronómetro
  - reiniciar cronómetro

Como se necesitan 4 pines, y las extensiones tienen 3 utilizaremos alguno libre.

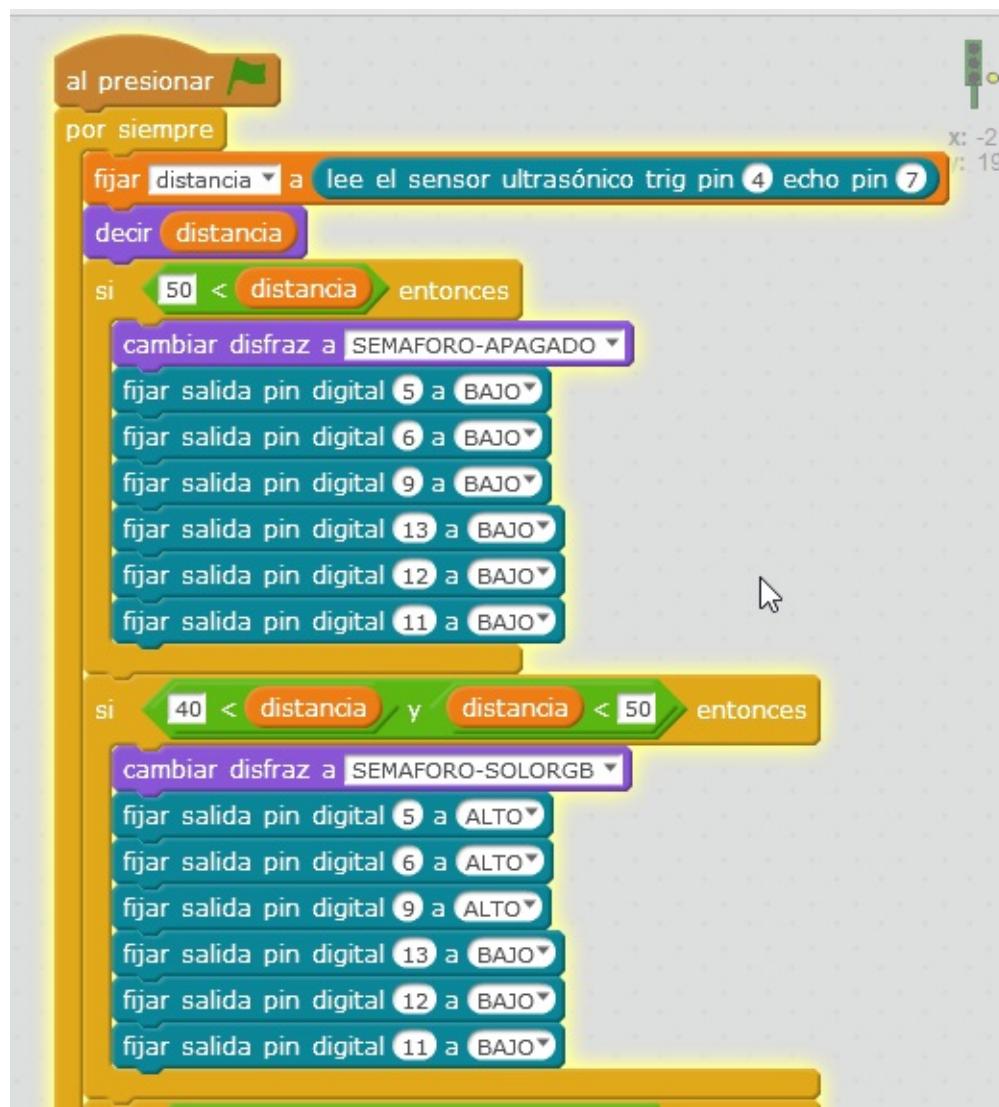
**RETO RADAR LUMINOSO** Realizar un programa que a medida que se acerque un objeto, se enciendan más luces, reutilizar el sprite del semáforo para visualizarlo.

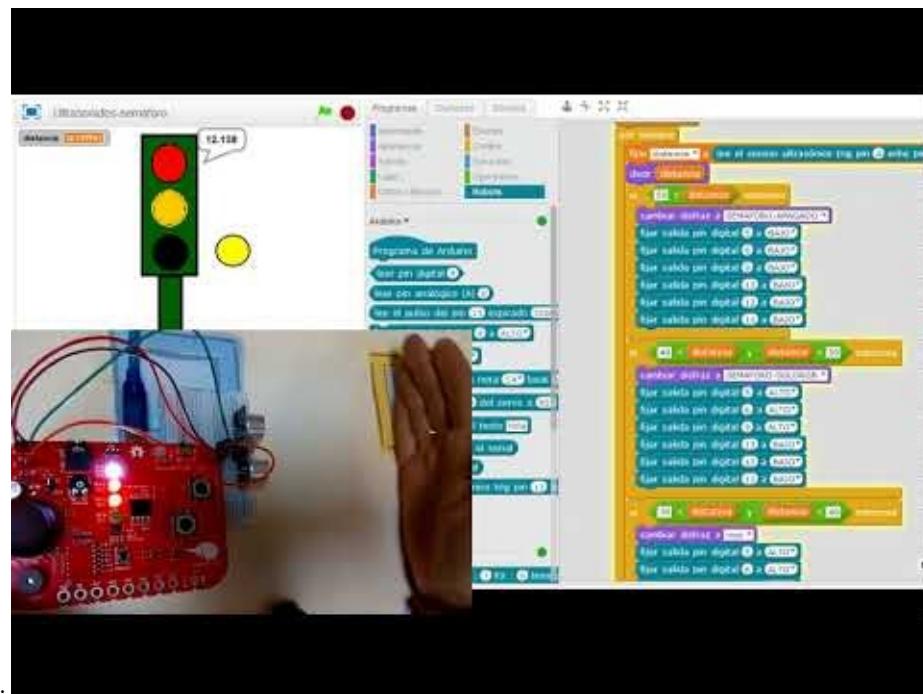
Solución

El programa es simplemente recoger la distancia con la instrucción que hemos señalado antes. La conexión que vamos a realizar entre los pines I/O de Echidna y el sensor ultrasonidos HC-SR04 va a ser:

- el D4 en Trig
- el D7 en Echo
- el '+' en Vcc
- el G en GND

Por lo tanto el programa es (sólo se visualiza una parte, el resto es continuar con las condiciones, hemos puesto de límites 50cm, 40cm, 30cm, 20cm y 10cm para ir encendiendo luces)





El resultado es :

[Video link](#)

Los siguientes retos, aunque las imágenes se ve que no están hechos con Echidna, da igual, es simplemente conectar el trig y echo en los pines D4 y D7 del Echidna y utilizar la instrucción de "lee el sensor ultrasónico trig pin 4 echo pin 7" (o utilizar otro orden o el D8 si te pones revelde y cambiar los números anteriores) ¿Te atreves a hacerlos todos ?



[via GIPHY](#)

**RETO HINCAR UNA PELOTA** Pon de sprite una pelota y que se hinche a medida que acercas un objeto al ultrasonidos. [Solución](#)

**RETO PIANO INVISIBLE**

Que suene una nota según la distancia del objeto. [Solución](#)

**RETO RADAR CON INTERMITENCIA DE UN LED**

Cuanto más cerca está un objeto, más rápido un led se enciende y apaga. [Solución](#)

**RETO SENSOR PARKING**

Cuanto más cerca está un objeto, más rápido suena un pitido intermitente [Solución](#)

## 5.5 Servo

El servo es un motor que podemos controlar el ángulo de giro, hay diferentes clases, pero nos vamos a centrar en este que tiene un precio/calidad aceptable, el MG90S.



Este servo se controla con una salida digital, y se les indica el ángulo de giro, que puede ir desde 0° hasta 180° (no permite otro margen). Si elegimos otro más barato, puede tener deriva en los extremos ([ver vídeo con HD-144A](#))

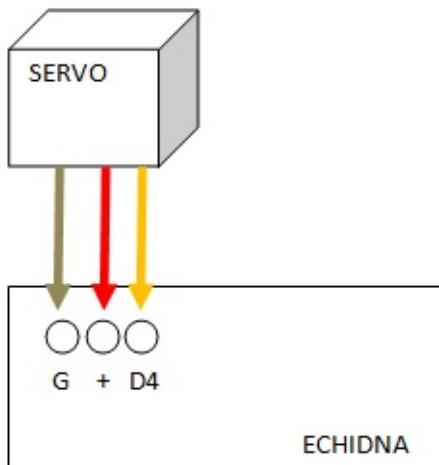
Aprende más sobre servos en esta página de [Luis Llamas](#).

La instrucción para los servos en mBlock es muy fácil, simplemente le indicamos en qué pin digital está conectado y el ángulo que deseamos:

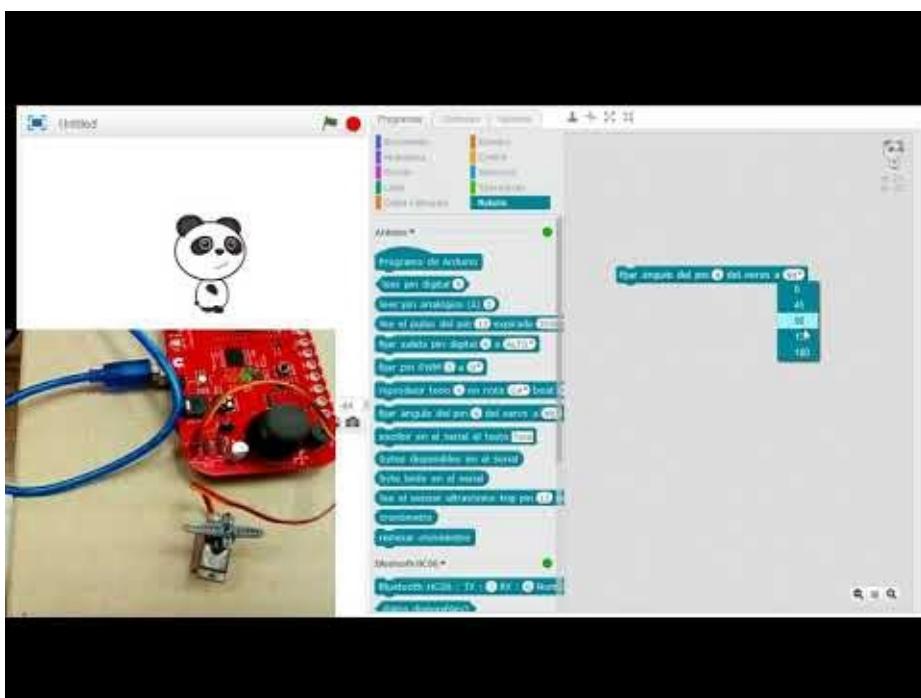
 A screenshot of the mBlock visual programming environment. The interface shows a sidebar with 'Arduino' selected, and a main workspace with a sequence of blocks. The blocks are:
 

- 'Programa de Arduino'
- 'leer pin digital 9'
- 'leer pin analógico (A) 0'
- 'lee el pulso del pin 13 expirado 20000'
- 'fijar salida pin digital 9 a ALTO'
- 'fijar pin PWM 5 a 0'
- 'reproducir tono 9 en nota C4 beat 16'
- fijar ángulo del pin 4 del servo a 90** (This block is highlighted with a yellow rectangle)
- 'fijar ángulo del pin 9 del servo a 90'
- 'escribir en el serial el texto hola'

y ponlo bien, el marrón indica la masa:



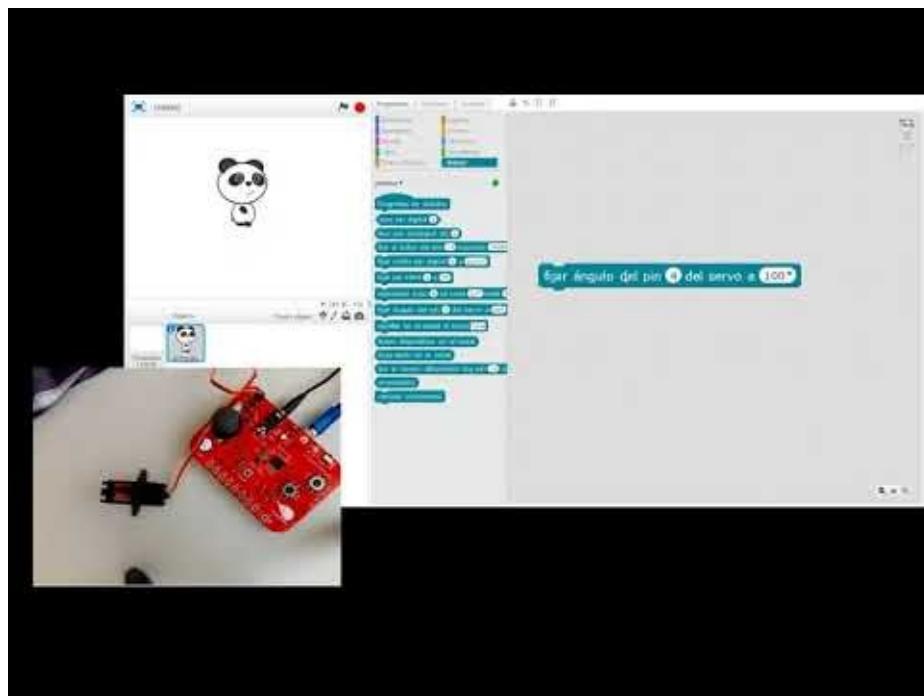
El resultado lo puedes ver en este vídeo, donde puedes observar que para valores límites 0° y 180° hace un poco de vibración pues trabaja forzado:



[Video link](#)

Mira la diferencia con un **servo de rotación continua**, fíjate como:

- Los extremos 0° y 180° es a máxima velocidad, pero un sentido u otro.
- 90° es parado.
- Un valor intermedio es menos velocidad (se ve el ejemplo 80° y 100°)
- Si tiene deriva, (cosa frecuente) hay un potenciómetro para ajustar.



[Video link](#)

## 5.6 MONTAJE 17 Tractor entrando al corral

Te proponemos un reto utilizando dos elementos vistos en esta unidad, para dar un poco de rienda suelta a tu imaginación de la cantidad de proyectos que se pueden hacer.

### Reto

Construir un proyecto donde la barrera (hecha de cartón y fijada al servo) está bajada y el semáforo en rojo. Si el sensor IR detecta un vehículo, tiene que abrir la barrera durante 5 segundos y el semáforo en verde. Antes de cerrar, se encenderá el semáforo en naranja para advertir que se va a cerrar la barrera.



[Video link](#)

### Solución

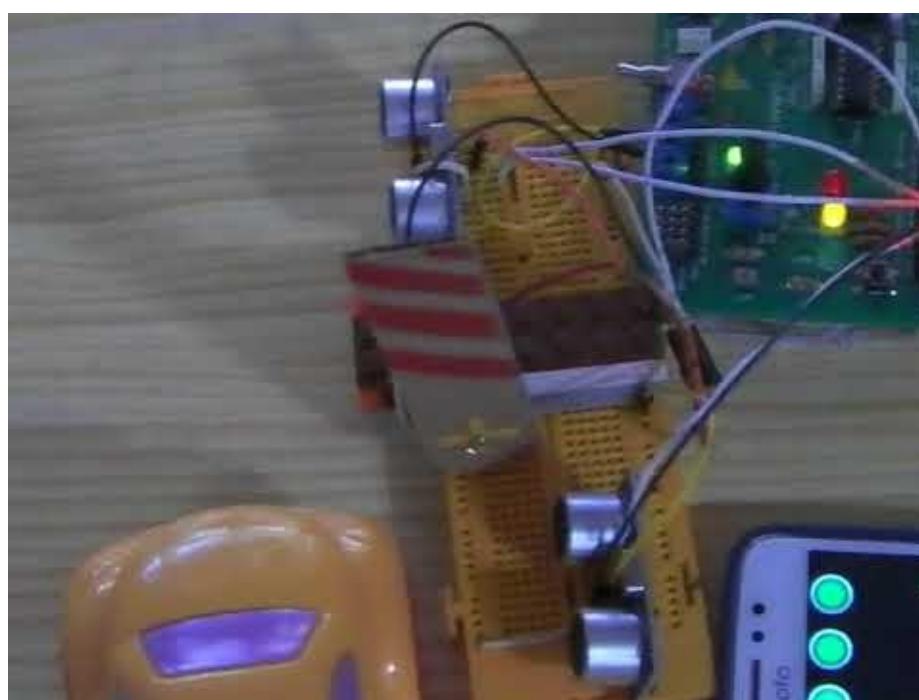
Conectamos por ejemplo el servo al D4 y el sensor de IR al D7



## Otro reto más difícil

Este ya es para los "sobresalientes":

Podemos hacer que la barrera se abra con Bluetooth, o que también baje si el sensor de Ultrasonidos en el otro lado detecta que el vehículo ya ha pasado. Aunque el vídeo está hecho con la Shield Edubásica perfectamente se puede hacer con Echidna, y en nuestro caso podemos sustituir un sensor ultrasonidos con un sensor IR.



[Video link](#)



■ ■

## Grupo ROBOTICA EDUCATIVA EN ARAGÓN

Tenemos un grupo en Telegram de profesorado interesado en la Robótica Educativa en Aragón, si estás interesado en unirte, envía un mensaje por Whatsapp o Telegram a CATEDU 623197587 y te enviaremos un enlace.



## Créditos

- 2018 por [CATEDU](#) (Javier Quintana Peiró).
- Imágenes de Echidna y capturas: <http://echidna.es/>
- Imágenes embebidas de GIFPY están en estos [términos](#).

Cualquier observación o detección de error por favor aquí [soporte.catedu.es](mailto:soporte.catedu.es)

Los contenidos se distribuye bajo licencia Creative Commons tipo BY-NC-SA.



# GOBIERNO DE ARAGÓN

Departamento de Educación,  
Cultura y Deporte

**CATEDU**   
CENTRO ARAGONÉS de TECNOLOGÍAS para la EDUCACIÓN



Esperamos que hayas disfrutado de la robótica con Echidna.



via GIPHY