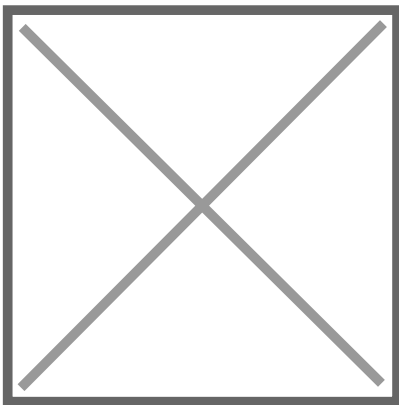


# 2 Programación en bloques mBlock

- [Instalando mBlock](#)
- [Dos formas de programar en mBlock](#)
- [Métodos para interactuar los objetos y el robot \(dispositivo\)](#)
- [Preparando mBlock y Arduino](#)
- [INTERRUPTOR CREPUSCULAR](#)
- [APERTURA PUERTA](#)
- [ALARMA AGUA](#)
- [JOYSTICK](#)
- [ALARMA LASER](#)
- [BLUETOOTH pasos previos](#)
- [BLUETOOTH](#)

# Instalando mBlock

**mBlock** es un programa especializado en el manejo de los robots de Makeblock ([ver cursos de mBot en Aularagon](#)), pero permiten muchas arquitecturas de placas pues los mismos robots de Makeblock estan basados en placas de hardware libre Arduino, ESP32...



Hay dos versiones, **online y offline**, las dos son buenas, no obstante

- Si tienes Windows o Mac preferimos la **versión offline** para no saturar la red en los centros, rapidez y comodidad.

- Si utilizas **Linux** o **Chromebook** tienes que usar la **versión online** obligatoriamente pues no hay versión instalable.

Tanto una opción como la otra, permite tener los proyectos en la nube de Makeblock, compartir, embeberlos, etc...

## Versión online

---

Para la versión web **primero hay que instalar mLink driver** nos vamos a

<https://mblock.cc/pages/downloads> y abajo nos encontramos mLink para descargar, **descarga el mLink correspondiente a tu sistema operativo** :

[mlink-1.png](#)

Licencia CC-BY -SA <https://docs.arduino.cc/tutorials/alvik/getting-started-mblock/>

Lo instalamos, y entonces pedirá instalar los **drivers** que son pequeños programas que comunican el PC con el hardware del robot

[mlink-drivers.png](#)

Licencia CC-BY -SA <https://docs.arduino.cc/tutorials/alvik/getting-started-mblock/>

Si utilizas **Chromebook** lee la siguiente [guía](#) para conectar el robot

Una vez instalado **ya podemos entrar en la web** <https://ide.mblock.cc/> para programar en bloques nuestro robot

Nota: La página <https://ide.mblock.cc/> no suele estar bien situada en los buscadores si ponemos *mBlock*, la página que suele salir es esta <https://www.mblock.cc/en/> y nos da dos opciones, trabajar con bloque, o con Python, elegimos bloques

2024-05-16 23\_50\_10-mBlock - One-Stop Coding Platform for Teaching and Learning.png

Si has entrado sin querer en la URL <https://ide.makeblock.com/> no permite el logueo en Europa

En resumen, la URL correcta es <https://ide.mblock.cc/>

En este editor **tenemos que loguearnos** podemos crear un nuevo usuario, utilizando una cuenta de Google/Apple o registrarnos con un correo electrónico

2024-05-16 23\_53\_10-mBlock Block-Based IDE- Coding for Beginners.png

## Versión offline Windows o Mac

---

Vamos a <https://mblock.cc/pages/downloads> y descargamos la versión correcta a nuestro sistema operativo.

[download-version.png](#)

Licencia CC-BY -SA <https://docs.arduino.cc/tutorials/alvik/getting-started-mblock/>

En windows hacemos doble click en el archivo ejecutable descargado

[installing-download.png](#)

Licencia CC-BY -SA <https://docs.arduino.cc/tutorials/alvik/getting-started-mblock/>

Lo instalamos, y entonces pedirá instalar los **drivers** que son pequeños programas que comunican el PC con el hardware del robot

[install-drivers.png](#)

Licencia CC-BY -SA <https://docs.arduino.cc/tutorials/alvik/getting-started-mblock/>

Algunas veces la ventana de driver de la figura anterior esta escondida en el fondo, hay que minimizar ventanas para acceder a ella.

En MAC el archivo será un fichero mount con la extensión .dmg, por lo tanto hay que arrastar el icono a la carpeta de aplicaciones

[install-mac.png](#)

Licencia CC-BY -SA <https://docs.arduino.cc/tutorials/alvik/getting-started-mblock/>

# Dos formas de programar en mBlock

## DOS FORMAS DE PROGRAMAR EN MBLOCK

Como vemos, mBlock permite dos formas de programar :

[https://docs.google.com/presentation/d/e/2PACX-1vQb1Dv9wN9QK-F6V7yvwDoyzquqwWlGvlyVjr83Yk56kAoYD7bXlnYDm\\_tCQkeAgg/pubembed?start=false&loop=false&delayms=3000](https://docs.google.com/presentation/d/e/2PACX-1vQb1Dv9wN9QK-F6V7yvwDoyzquqwWlGvlyVjr83Yk56kAoYD7bXlnYDm_tCQkeAgg/pubembed?start=false&loop=false&delayms=3000)

## OPCIÓN Programación en vivo

El programa reside en el ordenador, y en la placa hay instalado un Firmware para ir escuchando y ejecutando lo que manda el ordenador.

- **VENTAJAS**

- Te permite interactuar el robot y el ordenador, por ejemplo podemos hacer que cuando el detector de humedad detecte agua, que salga por pantalla un fondo acuático, o que pulsando una tecla del teclado se encienda un LED en la placa...

- **DESVENTAJAS**

- **hay que cargar dentro del robot el Firmware exclusivo de mBlock** para que Arduino haga caso a mBlock
- Hay que tener **nuestro ordenador como intermediario, se come los recursos** y puede que nuestro programa vaya lento
- Por supuesto necesita tener ordenador conectado al robot, o sea, trabaja como un esclavo del ordenador.

## OPCIÓN Programación cargar a la placa

Todos los programas editores de Arduino (tanto los que programan con código como el Arduino IDE) como los editores de programas gráficos en bloque (mBlock, Snap4Arduino, Arduinoblocks, ...) permiten cargar el programa en la placa. Las ventajas y desventajas son las opuestas de trabajar en vivo.

## EN VIVO ¿Qué es eso?

---

Existe una posibilidad de utilizar la placa "en vivo" frente a "cargar" el programa en la placa. Es decir, interactuando con el ordenador. El programa está en el PC. En la placa hay un **firmware** que le dice que este a las órdenes del PC. De esta manera podemos por ejemplo:

- Enviar órdenes desde el ordenador a la placa.

Por ejemplo que al pulsar la tecla espacio que se encienda el led D13

- Enviar información desde la placa al ordenador

Por ejemplo que muestre por pantalla nos muestre la cantidad de luz, que registra el sensor LDR, etc...

Que nosotros sepamos, estos programas permiten la programación en vivo :

- **mBlock** placas: Arduino, Microbit, Raspberry Pi, ... robots de Makeblock: mBot, Cyberpi...

- **EchidnaScratch** CURSO DE ECHIDNA

- **Microblocks**

**VENTAJAS** LA PROGRAMACIÓN EN VIVO PERMITE MUCHO JUEGO Y POSIBILIDADES A LA HORA DE ELABORAR PROYECTOS

**INCONVENIENTES:** Necesitas el ordenador encendido y conectado al robot.

# Métodos para interactuar los objetos y el robot (dispositivo)

Si queremos que nuestro robot se pueda comunicar con el objeto o objetos que tenga mBlock, antes con mBlock 3.0 la comunicación era inmediata, fíjate en este script de una alarma:

[embedded-image-xGEexnUg.jpeg](#)

Mezcla en el mismo script:

- órdenes específicas de la placa arduino (set digital...)
- órdenes específicas del objeto que exista en mBlock (por defecto el oso panda) say ....  
switch costume to ....
- órdenes del fondo switch backdrop to ...

Con mBlock 5.0 **YA NO SE PUEDE**, pero tenemos unos trucos

## MÉTODO UTILIZAR VARIABLES GLOBALES

Se pueden crear variables, en cualquier objeto, y las lee cualquier objeto,

**Este método se utiliza con la opción EN VIVO**

[variablesglobales.jpg](#)

De esta manera si creamos una variable **frase** para todos los objetos:

[frase3.jpg](#)

Podemos usarla en el robot

[frase1.jpg](#)

y el programa del objeto que queramos, en este caso el oso panda lo puede visualizar

[frase2.jpg](#)

# MÉTODO UTILIZAR MENSAJES

Cualquier objeto tiene a su disposición enviar mensajes a los otros

**Este método se utiliza con la opción EN VIVO**

[mensajes.jpg](#)

# MÉTODO EXTENSIÓN BROADCAST = TRANSMITIR MENSAJES

**Este método se utiliza con la opción EN CARGA**

Es parecido al anterior, hay que ir al + que hay abajo para instalar extensiones

[arduino-pelotahinchadistancia3.png](#)

Buscar la extensión "Broadcast" e instalarla

[broadcast.jpg](#)

Se instala primero descargándola con el + aquí

[broadcast1.png](#)

Y luego añadir una vez descargada

[broadcast2.png](#)

Entonces aparecen unas nuevas instrucciones

[broadcast3.png](#)

y se instalan unas instrucciones extras parecidas a las anteriores pero más potentes

- En el mismo mensaje podemos transmitir valores asociados
- Funciona **EN MODO CARGA**
  - Esto es muy útil pues hay instrucciones que sólo se pueden utilizar en modo CARGA, de esta manera podemos pasar valores de la placa electrónica a los objetos del ordenador (oso panda o lo que sea) simplemente teniendo conectado la placa con el ordenador.

El resto de objetos trabajan en modo vivo, es decir, si cambias un bloque, automáticamente se ven los efectos

A lo otros objetos TAMBIÉN hay que instalar la extensión BROADCAST

**DESVENTAJA** no se pueden transmitir mensajes de objetos a la placa. Sólo de la placa a los objetos

# Preparando mBlock y Arduino

Hay que hacer dos cosas

- Cuando te descargas **mBlock** por defecto está preparado para utilizarlo con los robots (mBot), hay que decirle que no vamos a conectar un robot, que la placa será **Arduino UNO**
- Nuestro **ARDUINO** no pueden entender mBlock por lo que hay que grabarle un software dentro (un software que se graba en una placa hardware se llama **FIRMWARE**, no se borra cuando se apaga) y este firmware entiende lo que le manda mBlock, es decir, si conectamos otro Arduino y no le hemos grabado el firmware, no entenderá a mBlock.

Lo mejor un tutorial muy breve:

[https://docs.google.com/presentation/d/e/2PACX-1vRLqEoJCT355xMCeCXsd0Wc7JcJRk9JkwLCyzEPb\\_h1S2IbYKK7OdUf33yWRIXq216Zgh9Da7-glOq1/embed?start=false&loop=false&delayms=3000](https://docs.google.com/presentation/d/e/2PACX-1vRLqEoJCT355xMCeCXsd0Wc7JcJRk9JkwLCyzEPb_h1S2IbYKK7OdUf33yWRIXq216Zgh9Da7-glOq1/embed?start=false&loop=false&delayms=3000)

# INTERRUPTOR CREPUSCULAR

## Conocimientos previos

### SENSOR DE LUZ LDR

El LDR es una resistencia que varía su valor con la luz, cuanto más OSCURO más grande es su valor, por lo tanto por la ley de Ohm  $V= I \cdot R$  este módulo nos da una señal analógica (voltios) que aumenta con la oscuridad. Para saber más del LDR te recomendamos [esta página de Luis Llamas](#).

Los valores analógicos en el Arduino van desde 0 a 1023, luego hay que elegir un valor umbral, el cual, a partir de ese valor es un valor digital 1 o un 0, nosotros hemos elegido 500.

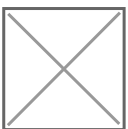
Por hardware puedes regular el potenciómetro que tiene el módulo para que produzca el cambio cuando lo desees:



### MODULO LED RGB

Este módulo tiene 4 pines que podemos proporcionar valores analógicos desde 0V a 5V para conseguir diferentes colores:

Pines B = Blue G = Green R = Red (-) = GND



Recuerda que Arduino tiene entradas digitales, entradas analógicas, salidas digitales, **pero no tiene salidas analógicas**. En vez de ello tiene salidas PWM ¿Qué no sabes lo que es? ☐

eso es porque te has saltado un capítulo anterior.

De momento conectaremos el pin G (Green) a una salida digital normal (pin 7), luego sólo se puede encender o apagar.

Más adelante usaremos los colores Blue y Red a salidas PWM

En [esta página](#) puedes encontrar más detalles de este módulo y un ejemplo curioso de utilización sin cables.

## RETO

- Cuando el sensor LDR detecte oscuridad
  - El led RGB se enciende verde
  - Cambiamos a fondo tipo "noche"
- En caso contrario
  - el led RGB está apagado
  - el fondo es día

## Solución esquema

- El módulo LDR lo conectamos al pin analógico A0
- El LED RGB lo conectamos
  - Pin 5 digital - Blue
  - Pin 6 digital - Red
  - Pin 7 digital - Green



## Solución Video

[https://www.youtube.com/embed/c\\_eyV8aCdvk](https://www.youtube.com/embed/c_eyV8aCdvk)

## Solución programa

El programa lo tienes aquí <https://planet.mblock.cc/project/3255351>

Utilizaremos la técnica variables globales <https://libros.catedu.es/books/domotica-con-arduino/page/programacion-mblock>

### **Objeto placa arduino**

Nota: el pin digital 3 bajo es por si tienes puesto el buzzer, para que no suene, lo veremos más adelante.

[interruptor-crepuscular1.jpg](#)

### **Objeto panda**

[interruptor-crepuscular2.jpg](#)

# APERTURA PUERTA

## Conocimientos previos

### SERVOMOTORES

Visita [esta página](#) para que veas dos vídeos y entiendas la diferencia entre:

- Servos convencionales
- Servos de rotación continua

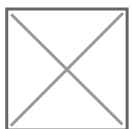


Tranquilo, que el servo que nos interesa es un **servo convencional** donde simplemente se fija el ángulo con la instrucción en mBlock



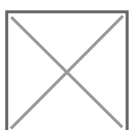
Si quieres saber más sobre servos te aconsejamos visitar [esta página de Luis Llamas](#).

### INTERRUPTOR TÁCTIL



No tiene mucho misterio: detecta una acumulación de carga y dispara un pulso positivo. Si quieres saber más mira [esta página de Luis Llamas](#).

### BUZZER ACTIVO





La diferencia con el pasivo es que no es necesario enviarle pulsos para que emita una frecuencia, sólo tenemos que dar la orden y él reproduce un tono.

**Ojo que funciona con lógica negativa** es decir:

- si queremos que **suene** tenemos que enviar un **LOW**.
- si queremos que **no suene** tenemos que enviar un **HIGH**

Si quieres saber más de este componente visita [esta página](#)

## RETO

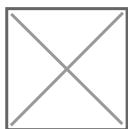
Cuando se pulse el interruptor táctil ....

- Se abre la puerta
- Al cabo de 5 segundos, tiempo suficiente para entrar
  - Se avisa que la puerta se va a cerrar con 3 pulsos buzzer y cambiando el color el sprite del Panda de mBlock
  - Se cierra la puerta

(sería como una llave táctil)

### Solución esquema

- pin 2 digital: El interruptor táctil touchless.
- pin 3 digital: El buzzer activo.
- pin 4 digital: Servo de la puerta.



### Solución Video

<https://www.youtube.com/embed/bBA3GEhLmRg>

### Solución programa

El programa lo tienes aquí <https://planet.mblock.cc/project/3251877>

Se utiliza la técnica de envío de mensajes <https://libros.catedu.es/books/domotica-con-arduino/page/programacion-mblock>

### **El dispositivo ARDUINO**

[domotica-aperturapuerta1.png](#)

### **El objeto OSO PANDA**

[domotica-aperturapuerta2.png](#)

El objeto **puerta** tiene este sencillo programa:

[domotica-aperturapuerta3.png](#)

# ALARMA AGUA

## Conocimientos previos

### SENSOR DE AGUA



Es un sensor muy sencillo: Al mojar los conductores, aumenta la conductibilidad y se traduce en una salida HIGH en su pin de salida **S**.

Si quieres saber más de este sensor visita [esta página de Luis Llamas](#).

Con sólo tener los dedos húmedos ya lo detecta.

Hay que tener en cuenta que sólo queremos detectar AGUA o NO AGUA, o sea digital. Hay otros sensores de humedad que están indicados para medir la humedad del suelo, (tienen forma de dos pinchos para clavarlos) en estos casos que queremos medir la cantidad de humedad la señal es más precisa y analógica. [Ver](#)

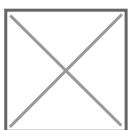
## RETO

Si detecto agua : Aviso de agua en el fondo y en el objeto Panda y cierro la puerta pues entra el agua de la lluvia con el mismo procedimiento que en apartado anterior.

### Solución esquema

- Pin D8 : Sensor de agua

El esquema el sensor de agua es uno de humedad, pero no había otro en Fritzing



### Solución Vídeo

<https://www.youtube.com/embed/VqUBA445Two>

### **Solución programa**

El programa lo tienes aquí <https://planet.mblock.cc/project/3255438>

Se utiliza la técnica de variables globales <https://libros.catedu.es/books/domotica-con-arduino/page/programacion-mblock>

Dispositivo ARDUINO

[domotica-agua1.png](#)

Objeto Oso panda

[domotica-agua2.png](#)

# JOYSTICK

## Conocimientos previos

### JOYISTICK

Yo no sé qué tiene este componente que llama la atención a los chavales



Así que vamos a aprovecharlo: Tiene 2 potenciómetros según los ejes X e Y y un interruptor central. Las dos primeras salidas son analógicas, el *microrruptor* (cuando presionas el joystick) lo trataremos también analógico, pues nos faltarán entradas y salidas digitales cuando lleguemos al final y queramos poner un Bluetooth.

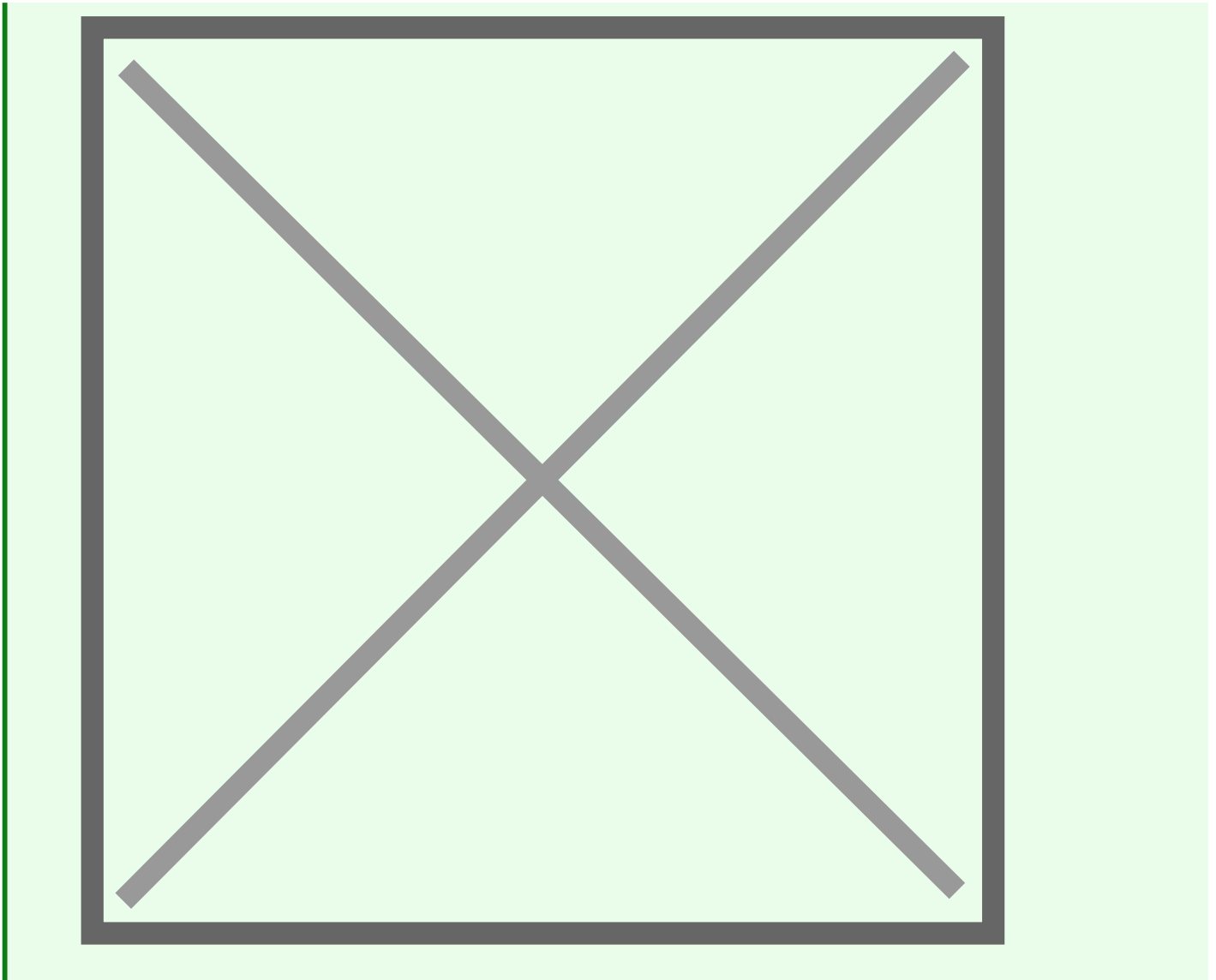
Si quieres saber más de este componente visita [esta página. de Luis LLamas](#)

### SALIDAS PWM

Como Arduino no tiene valores de salida analógicos, utilizaremos los pines PWM (~) para simular tensiones variables de salida.

De esta manera la luz Roja y Azul serán variables pudiendo crear el ambiente que queramos.

**¿No sabes lo que es PWM (~)?** eso es que no te has leído los capítulos anteriores



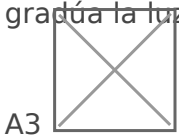
## RETO

- Si muevo en el eje X el Joystick la luz RGB se enciende AZUL proporcionalmente a la inclinación del mando Joystick.
- Si muevo en el eje Y el Joystick la luz RGB se enciende ROJO proporcionalmente a la inclinación del mando Joystick.
- Si pulso el Switch central
  - Si la puerta estaba cerrada la abre
  - Si la puerta estaba abierta, la cierra, con los avisos correspondientes.
- Además vamos a complicar el reto Y TIENE QUE INCLUIR
  - Lo visto en interruptor crepuscular:
    - Si es luz < 500 se hace de noche, tanto en el fondo com el oso panda
    - se enciende la luz verde del led RGB

- Lo visto en la apertura y cierre de la puerta con el sensor táctil exterior
  - si se pulsa el sensor táctil exterior, se abre la puerta, tanto la real, como la del objeto
  - se mantiene abierta durante 5 segundos
  - avisa que va a cerrar la puerta con tres avisos, el led13 parpadeando y el objeto avisando
  - se cierra la puerta, tanto la real como la del objeto
- Lo visto en alarma agua, pero modificado
  - si hay agua objeto gotera aparece y suena pitido

## Solución esquema

Al ser un componente analógico lo conectaremos en las entradas analógicas: \* La salida eje X que gradúa la luz azul en A1 \* La salida eje Y que gradúa la luz roja en A2 \* La salida Switch central en



## Solución vídeo

<https://www.youtube.com/embed/jPvQin9iFTE>

## Solución esquema

El programa lo tienes aquí <https://planet.mblock.cc/project/3255492>

Se ha utilizado el truco DE ENVIOS MENSAJES BROADCASTS ver

<https://libros.catedu.es/books/domotica-con-arduino/page/programacion-mblock>

Elegimos esta opción pues el programa es muy complejo para ejecutarlo EN VIVO es mejor cargar, y que la placa Arduino envíe los mensajes correspondientes. Si se realiza EN VIVO da problemas

Para ir simplificando se ha utilizado la creación de bloques, y cada mensaje que envía la placa Arduino lo recogerá un objeto

## DISPOSITIVO ARDUINO

Ejecuta siempre los 4 bloques



[DOMOTICA-JOYSTICK1.png](#)

Bloque NOCHE

[DOMOTICA-JOYSTICK2.png](#)

Bloque AGUA

[DOMOTICA-JOYSTICK3.png](#)

Bloque PULSADOR

[DOMOTICA-JOYSTICK4.png](#)

Bloque ABERPUERTA-Y-DESPUESICERRA

[DOMOTICA-JOYSTICK5.png](#)

Finalmente bloque JOYSTICK -RGB que lee los pines de entrada A2 y A3.

Son valores desde 0 hasta +1024 pero la posición central del Joystick está en 500, luego tenemos que contar desde 500 hasta 1024, luego hay que convertirlos a 0-255

Esos valores ya normalizados transmitirlos a los pines digitales donde está conectado el pin rojo y azul del led RGB por PWM

[DOMOTICA-JOYSTICK6.png](#)

### **OBJETO OSO PANDA**

recoge la LUZ

[DOMOTICA-JOYSTICK7.png](#)

### **EI OBJETO PUERTA**

Que recoge los mensajes del sensor táctil

[DOMOTICA-JOYSTICK9.png](#)

### **OBJETO WATER**

recoge el mensaje gotera y aparece si hay

[DOMOTICA-JOYSTICK10.png](#)



# ALARMA LASER

## Conocimientos previos

El diodo láser es un elemento motivador, barato y fácil de usar con el Arduino pues se activa digitalmente:



Si quieres saber más de este componente, te recomendamos [esta 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 SIEMPRE QUE EL LÁSER APUNTE A LOS OJOS** especialmente con niños.

## Reto

- **ALARMA**

- Si activo la alarma el láser tiene que encenderse
  - Una vez activada si se corta el láser, por lo tanto el valor del LDR sube, la alarma se dispara.
- Si desactivo la alarma
  - La alarma se apaga si se ha disparado, también por pantalla
  - El laser de apaga.

- Mantenemos la alarma **agua**

- Si se detecta agua, suena un aviso, también por pantalla

- Mantenemos el **pulsador exterior**

- Si se pulsa, la puerta se abre y se mantiene 5seg
- Se cierra automáticamente pero antes avisa, con un led y por pantalla

- Mantenemos el **Joystick**

- Si se mueve, se visualiza el led RGB colores azul y verde
- si se pulsa se abre la puerta



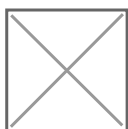
No mantenemos la programación del interruptor crepuscular pues necesitamos el LDR para la alarma



## Solución maqueta

En este caso **hay que tener cuidado con la fijación del láser y el LDR para que apunte al LDR** y otra cuestión son los cables: al instalarse fuera de la casa hay que utilizar cables largos o añadir dos M-H consecutivos.

- Hemos utilizado dos fijaciones con alambre el LDR para que quede fijo.
- En el láser hemos fijado con una chincheta para fijar el láser además de alambre
- Truco, si la maqueta va a estar fija, una buena gota con la pistola de pegamento es mano de santo para que se quede fijo



## Solución conexiones eléctricas

- La alimentación del láser y el del LDR igual que en los casos anteriores
- El control del LÁSER a la salida digital D10
- La lectura del LDR a la entrada analógica A0
- El resto: Joystick, detector fuego, sensor de agua, igual que las páginas anteriores

## Solución vídeo

<https://www.youtube.com/embed/3O7UwfKRlrI>

## Solución programa

El programa lo tienes aquí <https://planet.mblock.cc/project/3255982>

Principalmente el difícil es el bloque Alarma

[DOMOTICA-ALARMA.png](#)

Los otros objetos

Objeto puerta

[DOMOTICA-ALARMA2.png](#)

Objeto gota

[DOMOTICA-ALARMA3.png](#)

Objeto Policia

[DOMOTICA-ALARMA4.png](#)

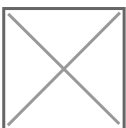
## Otra opción

En vez de un láser, con un SENSOR DISTANCIA POR ULTRASONIDOS



Este sensor mide las distancias utilizando el eco:

*Un ojo marcado con la T es un altavoz: Emite un sonido ultrasónico fuera de lo audible*  
El otro *ojo* marcado con una R es un micrófono que detecta el pulso emitido por T.



Por software hay que calcular la distancia utilizando la fórmula  $v=e/t$  donde  $v$  es la velocidad del sonido.

Si quieres saber más de este sensor mira esta página de [Luis Llamas](#).

su código en mBlock es muy sencillo:



Podríamos usar uno más profesional, por ejemplo [DETECTOR DE MOVIMIENTO CON ARDUINO Y SENSOR PIR](#), pero su ángulo de sensibilidad es tan abierto que todo el rato se dispara. (un rollo si se utiliza en clase).

#### Reto alternativo

- Si se pulsa el botón de activación
  - Si la alarma no está activada
    - **Activa** la alarma, es decir *está vigilando*.
  - Si la alarma está activada
    - **Desactiva** la alarma, *deja de vigilar*.
  - Si la alarma está disparada
    - Anula el disparo y desactiva la alarma
- Si la alarma está activada:
  - Está encendido el led verde para indicar que *está vigilando*.
  - Si detecta un *intruso* a menos de 10 cm
    - Se **dispara** la alarma, es decir se enciende la luz roja y el buzzer de forma intermitente, no se apaga hasta que se pulsa el interruptor.

#### Conexiones

- Entradas y salidas digitales
  - D3 Buzzer
  - D5 Blue de led RGB
  - D6 Red de led RGB
  - D7 Green de led RGB
  - D12 Echo del sensor de ultrasonidos
  - D13 Trg del sensor de ultrasonidos
- Entradas y salidas analógicas
  - A4 Pulsador

#### Video

<https://www.youtube.com/embed/RB7K16FhHlg>



El programa lo puedes descargar [aquí](#) pero realizado en mBlock3

# BLUETOOTH pasos previos

## Conocimientos previos

Tienes que visitar las siguientes páginas de la [Unidad 4 Comunicaciones con Arduino](#):

- ¿Qué es el [HC-06](#)?
- [La APP](#) que tienes que instalar
- [Vincular tu móvil con el HC-06](#) con tu móvil

Aprende a configurar los botones de la APP !! diapositiva 12 pero en vez de Up, Down, Right y Left los que se establezcan en el reto

<https://docs.google.com/presentation/d/e/2PACX-1vT0vG1z61MuZXKmdiw4ga7z15FIQfeussqDNYzMauJSZUU2G2NIL7M-JjXb4PFT4YTigj9YaI8PzHmR/embed?start=false&loop=false&delayms=3000>

[Configuración avanzada](#) pues **nosotros somos pitos, y conectaremos el HC-06 en los pines digitales D11 y D12** y no en D0 y D1 pues están ocupados con la comunicación del ordenador. El RX del HC-06 en 11 y el TX del HC-06 en el 12

[https://docs.google.com/presentation/d/e/2PACX-1vSCv2j25rIQxh6pHjMM4n4AXttDDetQPL93qMrYfQO2p-BVC6tSzeRVgU7nVq4\\_pXEnLedrvF7LTM4V/embed?start=false&loop=false&delayms=3000](https://docs.google.com/presentation/d/e/2PACX-1vSCv2j25rIQxh6pHjMM4n4AXttDDetQPL93qMrYfQO2p-BVC6tSzeRVgU7nVq4_pXEnLedrvF7LTM4V/embed?start=false&loop=false&delayms=3000)

## En mBlock INSTALAR LA EXTENSIÓN HC-06

Entramos en Extensión

[DOMOTICA-BLUETOOTH1.png](#)



Y ponemos en el buscador HC-06 nosotros hemos elegido esta, pero podría ser otra, es cuestión de experimentar

[DOMOTICA-BLUETOOTH2.png](#)

IMPORTANTE

- SOLO FUNCIONA CARGANDO EL PROGRAMA en vivo no
- NO PUEDES USAR ESTAS INSTRUCCIONES DENTRO DE UN BLOQUE no entiendo por qué, si alguien lo consigue que me lo diga

[DOMOTICA-BLUETOOTH3.png](#)

## Un ejemplo: Apertura simple de la puerta

Conectamos:

- RX del HC-06 en 12
- TX del HC-06 en 11
- El servo de la puerta en D4

y el código es:

<https://planet.mblock.cc/project/3481632>

[mblock-bluetooth-simpleapertura.png](#)



# BLUETOOTH

## Reto

Controlar nuestra casa con el móvil, para ello vamos a definir los siguientes comandos:

COMANDO	VOZ	DATO	descripción
Comando 1	abrir	A	abrir la puerta y cierra automáticamente
Comando 2	puerta	P	abrir/cerrar la puerta
Comando 3	alarma	L	activar/desactivar la alarma
Comando 4	pit	T	hace un pit
Comando 5	rojo	R	enciende luz interior roja
Comando 6	azul	B	enciende luz interior azul
Comando 7	apaga	O	apaga luces interiores

Lo tienes que hacer así:

## Solución conexiones eléctricas

## Solución vídeo

<https://www.youtube.com/embed/UGP6Oq95dfE>

## Solución programa

El programa lo tienes aquí : <https://planet.mblock.cc/project/3259600>

Todo este código, si lo pongo en un bloque, NO FUNCIONA, no sé por qué, así que a lo bruto en el bloque principal

[DOMOTICA-BLUETOOTH4.png](#)

Total que el programa del dispositivo Arduino queda bastante monstruoso

[DOMOTICA-BLUETOOTH5.png](#)

Objeto puerta

[DOMOTICA-BLUETOOTH6.png](#)

Objeto gotera

[DOMOTICA-BLUETOOTH7.png](#)

Objeto policia

[DOMOTICA-BLUETOOTH8.png](#)