

INICIACIÓN A LA ROBÓTICA CON MBLOCK



INDICE

FUNDAMENTOS SOBRE EL CONTROL Y LA ROBÓTICA

CONEXIÓN Y COMUNICACIÓN

FICHA 1: LED RGB Y CONEXIÓN SERIAL. Reto 1.

FICHA 2: USO DEL BOTÓN Y CONEXIÓN 2.4G. Reto2.

FICHA 3: ORDEN CONDICIONAL. Reto 3

FICHA 4: GENERAR SONIDOS. Reto 4 y 5.

FICHA 5: CREACIÓN DE BLOQUES. Reto 6.

FICHA 6: CONTROL DE LOS MOTORES. Reto 7.

FICHA 7: MUEVE EL ROBOT CON EL TECLADO. Reto 8 y 9.

FICHA 8: MUEVE EL ROBOT CON EL MANDO A DISTANCIA IR. Reto 10.

FICHA 9: ROBOT INDEPENDIENTE: MODO ARDUINO. Reto 11

FICHA 10: SENSOR DE ULTRASONIDOS: MEDIDA DE LA DISTANCIA. Reto 12 y13.

FICHA 11: SENSOR DE ULTRASONIDOS Y VARIABLES. Reto 14.

FICHA 12: MOTORES Y VARIABLES. Reto 15 y 16.

FICHA 13: DETECTOR DE LÍNEA. Reto 17.

FICHA 13: PROYECTOS FIN DE CURSO:

- a. SEGUIDOR DE LÍNEA. Reto 18.
- b. ROBOT LUCHADOR. Reto 19.
- c. CARRERAS DE ROBOTS. Reto 20.
- d. LABERINTO. Reto 21.
- e. CONCURSO DE HABILIDAD. Reto 22.

1. FUNDAMENTOS SOBRE EL CONTROL Y LA ROBÓTICA.

El funcionamiento de cualquier máquina o ROBOT responde al esquema de un SISTEMA EN LAZO CERRADO.

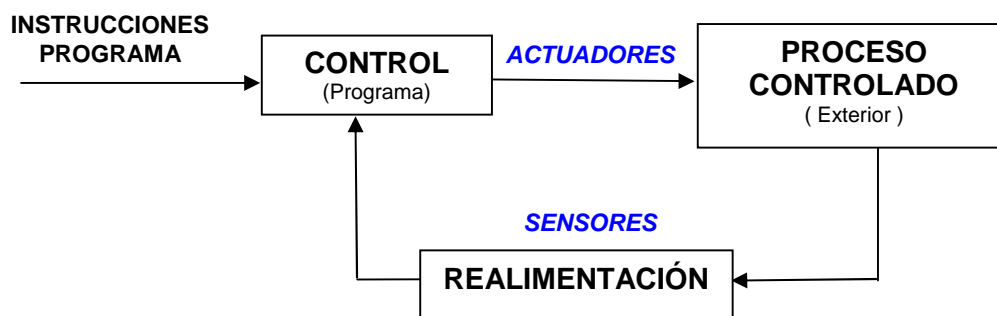
Cuando realizamos un programa, indicamos al robot que queremos que haga y cuando debe hacerlo.

La tarjeta controladora (placa **mCore**) es la responsable de guardar el programa y hacer que se cumpla. Ella es el CONTROL, el “cerebro” de nuestro robot.

Para que nuestro robot pueda cumplir con el programa necesita poder realizar acciones sobre su entorno. De eso se ocupan los ACTUADORES que son las “manos” de nuestro robot. Suelen ser motores (movimiento), resistencias (calor), Leds(luz), zumbadores (sonido),...

Del mismo modo, nuestro Robot no puede saber lo que sucede a su alrededor si no recibe información de su entorno. A este proceso lo llamamos realimentación. De eso se ocupan los SENSORES que son los “ojos” de nuestro robot. Le dicen qué está pasando a su alrededor: temperatura, objetos cercanos, posición en la que se encuentra, color de las cosas, etc.

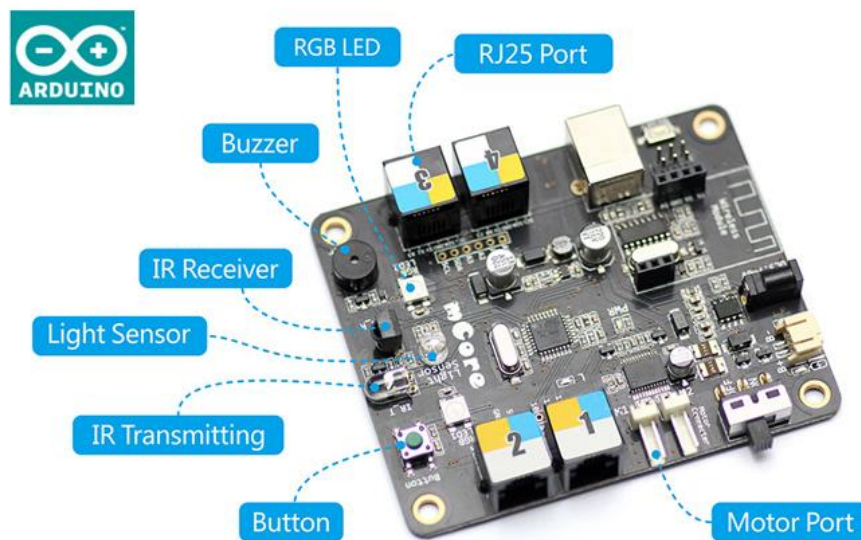
Si lo piensas, nosotros funcionamos de una forma similar. Ej: cuando cocinamos nuestro cerebro sabe lo que quiere hacer, pero necesitamos las manos para cocinar y los sentidos como el olfato, la vista,...para saber que tenemos que hacer en cada momento. Este sería un buen ejemplo de un sistema de control en bucle cerrado, como el de un robot.



Nuestro robot mBot tiene los siguientes elementos:

CONTROL (Cerebro)	ACTUADORES (Manos) <i>Actúan sobre proceso que estamos controlando.</i>	SENSORES (Ojos) <i>Informan al control de todo lo que pasa a su alrededor.</i>	
Placa mCore Donde grabamos el programa	<ul style="list-style-type: none"> • Motores para las ruedas • Servomotores (para movimientos exactos) • Buzzer (sonido) • Led RGB (luces de colores) • Matriz de Led (envían mensajes) 	<ul style="list-style-type: none"> • Ultrasonidos (mide distancias) • Sensor de color • Receptor Infrarrojos • Pulsador • Sensor de temperatura • Sensor de humedad • Sensor táctil • Sensor de ángulo (inclinómetro) 	<ul style="list-style-type: none"> • Sensor de sonido • Sensor de luz • Sensor brújula • Sensor de gas

La entrada y salida de información y órdenes del controlador se realizan por los PUERTOS RJ25.



COMO COMUNICARNOS CON MBOT

Nuestro robot puede funcionar de dos formas muy distintas: dependiente del ordenador (modo Scratch) o de forma autónoma (modo Arduino).

MODO SCRATCH (SC) la comunicación puede establecerse por cable USB, por 2.4 G (mediante una tarjeta insertada en el robot y un emisor conectado al PC) o por Bluetooth (con una tarjeta controlado por el teléfono o desde el ordenador). En este caso el robot puede ser controlado desde el teclado y siempre está conectado al PC.

MODO ARDUINO (Ar) creamos un programa que podemos subir al robot a través del cable USB utilizando la IDE de arduino (el programa de arduino que debemos tener instalado en nuestro PC). Cuando lo subimos el robot puede ser desconectado del ordenador y funcionará de forma independiente (aunque podamos enviarle órdenes por ejemplo desde el mando a distancia IR).

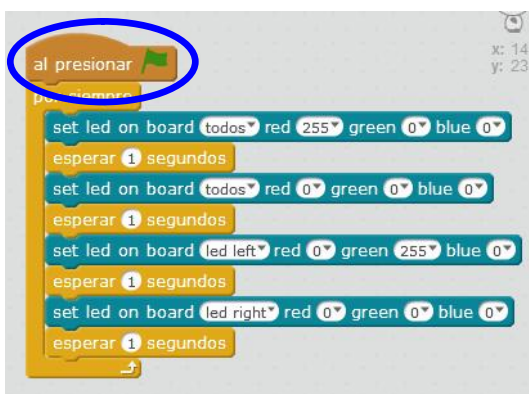
OJO:

- Sólo puede estar activa una de las conexiones para no tener problemas.
- Si el robot tiene cargado en memoria otro programa hay que borrarlo en conectar → actualizar fireware o conectar → restaurar programa predeterminado. Dejando que finalice totalmente la acción.

En modo SC el robot está conectado permanentemente con el PC bien sea por cable USB o por 2.4G.

En modo Arduino, podemos subir el programa al robot a través de la IDE Arduino (*hay que tener instalados los Driver - conectar → instalar driver arduino – y el programa de arduino en el pc*) y lo podemos desconectar porque será AUTÓNOMO.

SCRATCH



MBOT PROGRAM



Enciende los dos Led RGB a la vez en rojo. Espera 1 seg. Los apaga. Enciende el derecho en verde y lo apaga.

Modos de Funcionamiento del 2.4G serial (se cambian pulsando botón de la tarjeta)

- *Modo lento intermitente: si el led de la tarjeta parpadea lentamente es que tiene activada la función memoria y solo se conecta al emisor al que se conectó con anterioridad.*
- *Modo rápido intermitente: el led de la tarjeta parpadea rápidamente, la tarjeta se conectará al primer emisor 2.4G que detecte.*

FICHA 1: LED RGB Y CONEXIÓN SERIAL.


Nuestro Robot tiene en su placa base 2 Leds RGB. Un Led RGB emite luz en los tres colores primarios: rojo (Red), verde (Green), y azul (Blue). Al variar la intensidad de corriente de cada led se producen diferentes colores.

OBJETIVO: Aprender a conectar la placa Mbot al PC mediante el cable y nos iniciamos con la programación de mBot.

1. Conectamos mBot con el cable USB al PC y esperamos que la reconozca.
2. Abrimos mBlock y seleccionamos *Conectar* → *serial port* → *COM*..
3. Montamos este programa.
4. Pulsamos sobre la banderita para ejecutarlo en MODO SCRATCH.



Como funciona: se encenderán los 2 LED RGB en rojo durante 1 segundo y se apagarán.

Como se monta: La banderita  está en el menú **Eventos**. El led RGB de a bordo en **Robot**.



El tiempo en **Control:**

Este programa se repite una sola vez.



Repetimos el programa utilizando la orden de “repetir por 10 veces” que encontramos en el menú **Control**.



Logramos que los dos led se enciendan y apaguen en color rojo 10 veces.

RETO 1: Tienes que conseguir que se encienda primero el led derecho en color rojo 1 seg y se apague, luego el izquierdo en rojo 1 seg y se apague. Luego repite el proceso con el color verde y el color azul. Esto debe repetirse 5 veces. ¿Eres capaz de hacerlo?

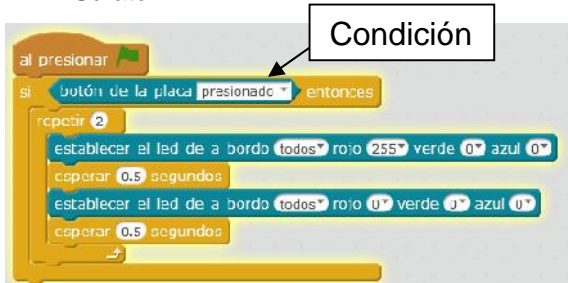
Como podemos lograr que el proceso se repita una y otra vez, por siempre? Mira en el menú **control** y cambia el programa. Graba un vídeo con tu móvil.

FICHA 2: BOTÓN, COMUNICACIÓN 2.4G.

La placa mBot tiene un pulsador o botón que nos permite interactuar con nuestro robot. Como hemos comentado, nuestro robot puede permanecer conectado al Pc bien por el cable USB o por comunicación 2.4G sin que se produzcan interferencias entre unos robots y otros.

OBJETIVO: Aprender a conectar la placa Mbot al PC por 2.4G y uso del botón.

1. Vamos a conectar la placa mBot con 2.4G, para no utilizar el cable. Desconectamos el cable USB. (La placa debe estar encendida)
2. Conectamos el emisor 2.4G a nuestro PC y esperamos que lo reconozca.
3. Pulsamos en *conectar* → 2.4G → *conectar*
4. Montamos el programa y pulsamos la bandera para que se ejecute. Si no funciona pulsamos *conectar* → *actualizar fireware*. Recuerda que estamos trabajando en modo Scratch.



Cuando pulsemos la bandera el programa se pone en marcha.

Cuando **pulsemos el botón** de la placa mBot, los led parpadean en Rojo dos veces.

El programa ha finalizado.

¿Cómo podemos hacer que este programa se repita?



Ahora, cuando pulsemos la bandera el programa se pone en marcha y siempre que pulsemos el botón los led parpadean en Rojo dos veces.

El programa no finaliza hasta que pulsemos el botón rojo de PARAR

PROGRAMAR ES PENSAR

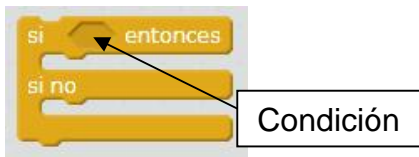
PENSAR Y EXPLICAR AL ROBOT, EN SU IDIOMA QUÉ QUEREMOS QUE HAGA.

RETO 2: Márcate tu propio objetivo, escríbelo en un papel e intenta llevarlo a cabo. Después sólo tienes que grabarlo en tu móvil explicando como funciona.

FICHA 3: ORDEN CONDICIONAL

En Robótica es muy importante la orden condicional. Consiste en que si se cumple una condición el robot hará una cosa y si no se cumple hará otra distinta. Esta orden es utilizada para que el robot tome decisiones.

2. Conectamos el emisor 2.4G a nuestro pc y esperamos que lo reconozca.
 3. Pulsamos en *conectar* → 2.4G → *conectar*
 4. Montamos el programa y pulsamos la bandera para que se ejecute.
- Si no funciona pulsamos *conectar* → *actualizar firmware*.



La encontraremos en el menú **CONTROL**.

* **Si...se cumple** la condición que queremos
...entonces: Hace lo que le decimos

* **Si no la cumple:** Hace esto otro



Este programa se repetirá por siempre:

Si pulso el botón se encenderán los 2 LED en blanco 3 seg y luego se apagaran.

Si no pulso el botón se encenderán en color ROJO.

Piensa: ¿Hasta cuando estarán en rojo?

RETO 3: Utiliza la orden "si...entonces, si no" del menú CONTROL

Queremos que cuando el botón esté pulsado:

- 1º Los 2 Led parpadeen 3 veces en rojo.
- 2º Led derecho parpadea 3 veces en verde.
- 3º Led izquierdo parpadea 3 veces en azul

Si el botón NO está pulsado:

Los 2 led se encienden en blanco durante 2 seg y se apaguen. Se quedarán apagados hasta que volvamos a pulsar el botón. Esto debe repetirse de forma indefinida, por siempre.

Todo el programa debe repetirse por siempre.

Graba un vídeo con tu programa utilizando el móvil.

FICHA 4: GENERAR SONIDOS.

Nuestro robot tiene un ZUMBADOR que emite sonidos más o menos agudos y de mayor o menor duración. Vamos a conectar la placa mBot con el cable USB y desconectar el 2.4G.

1. Conectamos el cable USB emisor *Conectar* → *serial port* → *COM.*
3. Montamos el programa y pulsamos la bandera para que se ejecute. Si no funciona pulsamos *conectar* → *actualizar firmware*.



Como vemos podemos variar la nota y el agudo de la misma. Con la orden repetir podemos hacer que esta se repita varias veces.

RETO 4: Queremos que se encienda el Led derecho en rojo reproduciendo un sonido y después el Led izquierdo en verde reproduciendo un sonido distinto. Esto debe repetirse 5 veces.

RETO 5: Utiliza la orden condicional y haz un programa que “se repita por siempre” .

Cuando pulso el botón suene un sonido y se enciendan los dos led en Rojo, después se apagan 1 segundo.

Si NO lo pulso suena otro sonido y se encienden en Azul, después se apagan durante 1 segundo.

El programa debe repetirse “por siempre”.

FICHA 5: CREACIÓN DE BLOQUES.

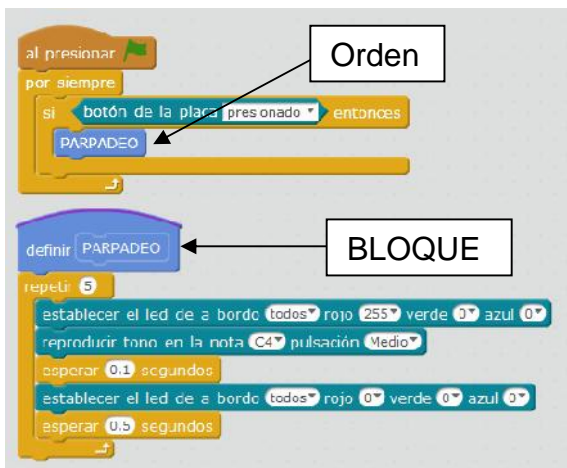
Es habitual que las órdenes que le damos a un programa sean largas y se repitan varias veces, para ordenar y aclarar el programa se utilizan los BLOQUES.

Un bloque es un conjunto de órdenes a las que asignamos un nombre. Una vez creado podemos utilizarlo cuantas veces queramos en el programa.

OBJETIVO: Aprender a crear BLOQUES y a utilizarlos.

Vamos a utilizar la placa mBot con el cable USB.

1. Verificamos que el 2.4 G está desconectado. *Conectar* → 2.4G → *desconectar* .
2. Conectamos el cable USB emisor *Conectar* → serial port → COM..
3. Montamos el programa y pulsamos la bandera para que se ejecute. Si no funciona pulsamos *conectar* → *actualizar firmware*. Trabajamos en modo Scratch.



Nos dirigimos al menú DATOS Y BLOQUES. Creamos un BLOQUE llamado parpadeo y creamos la secuencia.

Siempre que pongamos la orden PARPADEO repetirá todo lo que hemos indicado en el Bloque.

Pulsamos la banderita.

Al presionar el botón se pone en marcha el bloque parpadeo. PARPADEO: Se encienden los 2 led en rojo y se emite un sonido, se apagan. Se vuelve a repetir 5 veces.

RETO 6: Queremos diseñar un semáforo con nuestra placa Mbot.

Programa A Imagina que los Leds RGB de tu robot son un semáforo. Tienes que crear un bloque llamado “Parpadeo Amarillo” en el que parpadean en amarillo y se reproduce un sonido. El parpadeo debe repetirse 5 veces. ¿Cómo lograr el color amarillo con el led RGB? Busca en Internet.

Debes lograr que los colores se enciendan en el orden adecuado. Rojo, verde, parpadeo amarillo y que esto se repita.

Programa B Ahora vamos a añadir un botón y la orden condicional. Crea un bloque llamado “Peatones”.

Cuando pulsemos el botón debe activarse el bloque SEMAFORO que hace que permanezcan los led en color MORADO y emita un sonido que se repite 5 veces. Si no lo pulsamos, se ejecuta el programa A.



Más difícil todavía: Logra hacer un cruce de semáforos en el que el Led derecho sea un semáforo y el izquierdo otro.. Queremos que cuando uno esté en Rojo otro esté en verde. Piensa la secuencia de encendido de cada Led. Piénsalo bien, para que no existan accidentes, no pueden encenderse los dos en verde a la vez o provocarás un accidente de tráfico.

FICHA 6: CONTROL DE LOS MOTORES.

Nuestro robot tiene dos motores. Podemos hacer que giren hacia delante con una velocidad de 0 a 255 ó hacia atrás de 0 a -255.

OBJETIVO: Aprender a controlar los motores del robot.

1. Conectamos el 2.4G. *conectar* → 2.4G → *conectar*.
2. Montamos el programa, sujetamos en nuestra mano el robot y pulsamos la bandera para que se ejecute. Si no funciona pulsamos *conectar* → *actualizar firmware*.



El motor 1 se pone en marcha hacia delante a máxima velocidad durante 1 segundo. Después se para y se pone en marcha el motor 2 durante 1 segundo.

El proceso se repite 3 veces.

RETO 7: Coloca un objeto en el suelo, por ejemplo una lata de refresco. Al pulsar el botón el robot debe ponerse en marcha y dar 1 vuelta al objeto y, al finalizar debe emitir un sonido. Tendrás que hacer 3 bloques: AVANZA, GIRA y PARA. Realiza el programa y grábalo con tu móvil.

RETO 8: Haz que tu robot BAILE, SE ILUMINE Y EMITA SONIDOS. Escribe en un papel que movimientos quieres que haga tu robot, que colores quieres que tengan los LED y que sonidos quieres que emita mientras baila. Realiza el programa y grábalo con tu móvil.

FICHA 7: EL ROBOT SE MUEVE CONTROLADO CON EL TECLADO.

Siempre que trabajamos en modo Scratch, nuestro robot está conectado al ordenador, por el cable USB o bien por el emisor 2.4G. Si utilizamos el cable el robot, como es lógico, no podrá separarse del ordenador. Pero si lo conectamos con el 2.4G nuestro robot puede separarse. Sólo tenemos que decirle que queremos que haga cuando pulsamos una u otra tecla.

OBJETIVO: Aprender a controlar nuestro robot utilizando el teclado del ordenador.

Primero vamos a probarlo utilizando el cable USB y después el 2.4G serial.

1. Conectamos el cable USB emisor *Conectar* → *serial port* → *COM*..
2. Montamos el programa, sujetamos en nuestra mano el robot y pulsamos la bandera para que se ejecute. Si no funciona pulsamos *conectar* → *actualizar firmware*.



Sujeta el robot en tu mano para que no salga corriendo.
PULSA la **bandera verde**

Cuando pulsamos la **tecla a**:

Los dos motores avanzan a máxima velocidad uno para adelante y otro para atrás, girando.

Si pulsamos la **tecla s**:

Los dos motores se paran.

Desconecta el cable USB y conecta el 2.4G y Conectamos el 2.4G. *conectar* → *2.4G* → *conectar* .

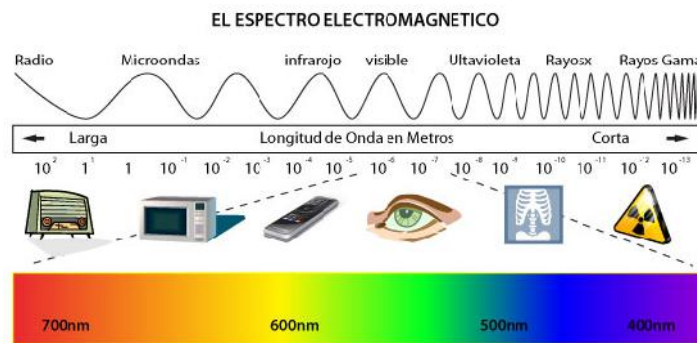
Prueba de nuevo el programa.

RETO 9: Haz un programa de modo que con las teclas del ordenador: arriba, abajo, derecha e izquierda, podamos controlar totalmente el movimiento del robot utilizando el 2.4 G. Cuando avance deben encenderse los 2 led en Rojo, cuando retroceda los 2 en verde. Si gira a la izquierda sólo se enciende el led izquierdo y si va hacia la derecha, sólo el derecho. No olvides que tendrás que pararlo de alguna forma.

FICHA 8: EL ROBOT SE MUEVE CONTROLADO CON EL MANDO A DISTANCIA IR.

Nuestro mandos a distancia funcionan emitiendo LUZ INFRAROJA. Las ondas electromagnéticas (radio, luz, microondas, rayos x,.. se caracterizan por su frecuencia. Sólo una parte de ellas es visible por el ojo humano (las de frecuencia intermedia). Las de alta frecuencia pueden transportar mucha energía y son dañinas, como los rayos X.

La luz infrarroja IR, es una luz que nuestro ojo no puede ver, y podemos utilizar para transmitir información.



La luz infrarroja es adecuada para hacer mandos a distancia porque:

- Utilizan luz en una frecuencia que no es peligrosa o dañina.
- Tiene relativamente poco alcance, lo suficiente para alcanzar la televisión o a nuestro robot.
- Como es una luz, si colocamos un objeto entre el mando y el diodo recetor de la luz, la señal no llegará. Se dice que “el mando debe ver al robot”.
- Otro problema es que si la luz llega a otro robot que tiene el mismo programa podemos mover el robot equivocado.

que trabajamos en modo Scratch, nuestro robot está conectado al ordenador, por el cable USB o bien por el emisor 2.4G. Si utilizamos el cable el robot, como es lógico, no podrá separarse del ordenador. Pero si lo conectamos con el 2.4G nuestro robot puede separarse. Sólo tenemos que decirle que queremos que haga cuando pulsamos una u otra tecla.

Primero vamos a probarlo utilizando el cable USB y después el 2.4G serial.

1. Comprueba que el cable USB está desconectado, conecta el 2.4G y pulsa *conectar* → 2.4G → *conectar*.



En el menú **botón → del mando IR presionado** está en el menú ROBOTS.

Sujeta el robot en tu mano para que no salga corriendo. PULSA la **bandera verde**

Cuando pulsamos la **tecla → del mando a distancia**:

Los dos motores avanzan a máxima velocidad.

Si pulsamos la **tecla A**:

Los dos motores se paran.

RETO 10: Repite el programa que controlaba el robot con el teclado del ordenador pero ahora debes **controlarlo con el mando a distancia**. Recuerda que si apuntas con tu mando a otro robot, puedes moverlo sin querer.



FICHA 9: ROBOT INDEPENDIENTE: MODO ARDUINO.

Es importante que comprendas que el Robot continúa conectado al ordenador. Depende de él. NO ES INDEPENDIENTE. Si apagamos el ordenador el programa se borra. Para el quede en el robot habrá que subirlo en el MODO ARDUINO.

Ahora que ya lo tienes hecho vamos a subirlo y grabarlo en su memoria.

Es obligatorio utilizar el cable USB.

1. Comprueba que está desconectado el 2.4G.
2. Conectamos el cable USB emisor *Conectar* → *serial port* → *COM.*

3. Realizamos el programa. Observa que hemos cambiado la banderita  por  si no hacemos esto no funciona.

3. Pulsamos en el menú EDITAR→MODO ARDUINO. La placa controladora de nuestro robot es una placa Arduino modificada. Tiene una memoria en la que puedes guardar el programa en un código de programación similar al C. El programa en código C, aparecerá en la derecha.



Si pulsas en “**subir a arduino**” tarda un rato pero el programa se guarda en la memoria del robot y quedará ahí aunque apaguemos el ordenador.

4. Si no funciona pulsamos *conectar*→ *actualizar fireware*.



Sujeta el robot en tu mano para que no salga corriendo. DESCONECTA EL CABLE USB, ya no necesitamos estar conectados al ordenador. El robot es INDEPENDIENTE. Guarda el programa y cierra el programa para que puedas ver que no está conectado al ordenador.


Cuando pulsamos la **tecla a**:

Los dos motores avanzan a máxima velocidad uno para adelante y otro para atrás, girando.

Si pulsamos la **tecla s**:

Los dos motores se paran.

RETO 11: Cambia el programa de control con el mando a distancia para que podamos controlar el

robot de forma INDEPENDIENTE del ordenador. . Conecta el cable USB, comprueba que no está conectado por 2.4G y conéctalo por cable. Pulsamos en el menú EDITAR→MODO ARDUINO. Pulsa en “**subir a arduino**”, tarda un rato.

AHORA: Intenta llevar tu ROBOT a su APARCAMIENTO evitando los obstáculos.

FICHA 10: SENSOR DE ULTRASONIDOS: MEDIDA DE LA DISTANCIA.

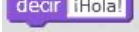
Los sensores de ultrasonido son detectores de proximidad que pueden demarcar objetos sin llegar a tocarlos. Pueden medir desde centímetros hasta metros. El sensor emite un sonido, que se rebota en el objeto y regresa. La **distancia** dependerá del tiempo que tarde en regresar. Su funcionamiento es similar a lo que hacen los murciélagos.

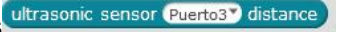
OBJETIVO: Aprender medir la distancia hasta un objeto con nuestro robot.

Primero vamos a probarlo utilizando el cable USB.

1. Conectamos el cable USB emisor *Conectar* → *serial port* → *COM..*
2. Montamos el programa, sujetamos en nuestra mano el robot y pulsamos la bandera para que se ejecute. Si no funciona pulsamos *conectar* → *actualizar fireware*.



Utilizaremos la orden  que encontramos en el menú **APARIENCIA**.

Dentro introducimos  que encontramos en el menú **ROBOTS**.

Comprueba que tu sensor de ultrasonido está conectado al puerto 3. El robot mide la distancia y la dice. Repite este proceso por siempre.

RETO 12: Observa este programa tiene un problema y tienes que encontrarlo. Queremos que nuestro sensor de ultrasonidos mida la distancia a un objeto. Si la distancia que mide es menor de 10 cm, se encienden los leds en rojo y **se para el motor**. Si la distancia es mayor de 10



Cm se **apagan los led** y el motor 1 avanza. Esto debe repetirse por siempre.

¿Por qué no hace lo que queremos? Solucióvalo

RETO 13: Ya lo has solucionado? Ahora queremos que conectes el robot con 2.4G. Debe avanzar recto, cuando esté a menos de 10 cm de una pared debe pararse y los led deben parpadear en rojo 5 veces. Si movemos el robot con la mano y la distancia es mayor de 10 cm debe volver a repetir el proceso.

FICHA 11: SENSOR DE ULTRASONIDOS Y VARIABLES.

Ya eres un gran programador. Tienes que utilizar correctamente todas las herramientas de programación. Una de ellas son las VARIABLES. Una variable es una especie de “cajón” donde almacenamos una información que utilizamos cuando necesitamos.

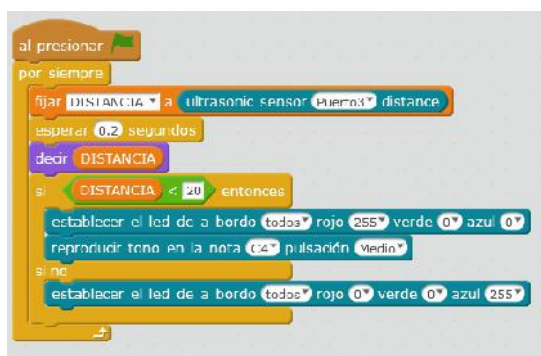
Por ejemplo, podemos crear una variable llamada DISTANCIA donde almacenamos lo que va midiendo nuestro sensor de ultrasonidos. La variable podemos utilizarla cuando la necesitemos.

Para crear una VARIABLE, entramos en el menú *DATOS Y BLOQUES*.

Los sensores de ultrasonido son detectores de proximidad que pueden demarcar objetos sin llegar a tocarlos. Pueden medir desde centímetros hasta metros. El sensor emite un sonido, que se rebota en el objeto y regresa. La **distancia** dependerá del tiempo que tarde en regresar. Su funcionamiento es similar a lo que hacen los murciélagos.

OBJETIVO: Medir la distancia con el ultrasonido y almacenar el dato en una VARIABLE.

1. Conectamos el cable USB emisor *Conectar* → *serial port* → *COM..*
2. Montamos el programa, sujetamos en nuestra mano el robot y pulsamos la bandera para que se ejecute. Si no funciona pulsamos *conectar* → *actualizar firmware*.



Creamos una VARIABLE llamada DISTANCIA, en ella guardamos la distancia que mide nuestro robot. Nos esperamos un instante para que “piense” y, nos DICE la distancia que mide.

Explica que hace después con tus palabras.

Monta el programa y comprueba si has acertado.

RETO 14: prepárate, este será un gran reto. Queremos que nuestro robot funcione de forma INDEPENDIENTE y avance. Si encuentra un obstáculo a menos de 10 cm debe de evitarlo, retrocediendo un poco y cambiando de dirección. Nunca debe de estrellarse. Utiliza los sonidos y los led.

Piensa antes de programar que es lo que debe hacer.

Antes de subirlo en MODO ARDUINO, prueba el programa en modo Scratch (con la banderita) es más rápido. Para cargarlo tienes que usar el cable USB

RETO 15: Ahora queremos más, queremos que el robot entre en un aparcamiento que hemos construido con una caja de cartón. Debe pararse dentro hacer un ruido y regresar al punto de inicio. ¿Es difícil no?

FICHA 12: MOTORES Y VARIABLES.

Habrás visto que la velocidad de nuestro robot sólo puede ser 0, 50, 100 y 255. No podemos utilizar velocidades intermedias. Pues si podemos utilizarlas con las VARIABLES.

OBJETIVO: Variar la velocidad del robot con una VARIABLE.

1. Vamos a utilizar la conexión 2.4G para probar los programas y cuando funcionen correctamente, utilizaremos el cable USB para subir el programa en MODO ARDUINO.
2. Montamos el programa, sujetamos en nuestra mano el robot y pulsamos la bandera para que se ejecute. Si no funciona pulsamos *conectar* → *actualizar firmware*.



En este programa creamos una VARIABLE llamada VELOCIDAD. Le damos un valor inicial de 90.

Usamos el valor de la variable=90 como velocidad del motor1.

Hacemos que se mueva a esa velocidad durante 1 seg.

Ahora hacemos que el valor de la variable aumente en +55 y hacemos que el motor acelere a esa velocidad.

El motor irá aumentando su velocidad y al final se parará durante 2 segundos.

El proceso volverá a comenzar.

RETO 16: Crea dos variable. Una llamada DISTANCIA Y otra VELOCIDAD. Tienes que lograr que el robot avance a toda velocidad, cuando encuentre un obstáculo a menos de 80 cm, reducirá su velocidad a 200, si el objeto está a menos de 50 cm irá a 150, si está a menos de 20 cm irá a 100 si está a menos de 5 cm debe pararse. Realizar un sonido 5 veces, retroceder a máxima velocidad, girar de espaldas y volver a avanzar, repitiendo el proceso. Grábalo todo con tu móvil. Piensa antes de programar que es lo que debe hacer.

Antes de subirlo en MODO ARDUINO, prueba el programa en modo Scratch (con la banderita) es más rápido. Para cargarlo tienes que usar el cable USB.

¿Serás capaz de hacerlo? Quien sabe...

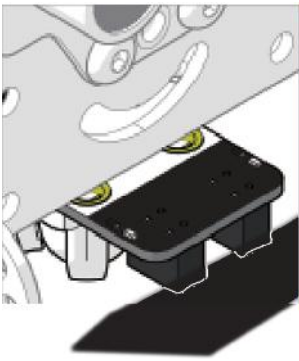
FICHA 13: DETECTOR DE LÍNEA.

Nuestro robot puede utilizar muchos sensores que, como vimos, le envían información. Hay detectores de humo, humedad, temperatura, inclinación,...

Vamos a ver un sensor que detecta si el robot está sobre una línea negra. Está en la parte delantera y tiene dos detectores uno a la derecha y otro a la izquierda. Emiten una luz infrarroja y con el reflejo saben si están sobre un color negro o un color blanco. Pueden mandar al robot un valor 0, 1, 2 ó 3. Veamos cuando nos dan cada uno de ellos:

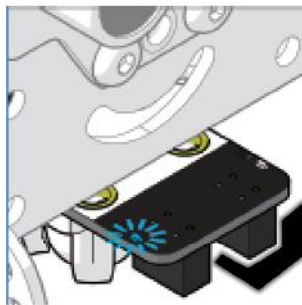
VALOR 0

Los 2 detectores ven color negro.
Están sobre línea.



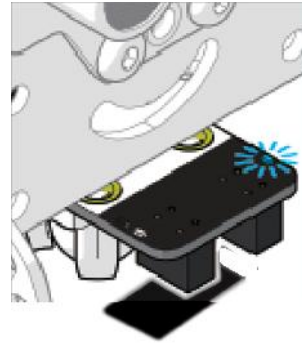
VALOR 1

Derecho sobre blanco.
Izquierdo sobre negro.



VALOR 2

Derecho sobre Negro.
Izquierdo sobre blanco.



VALOR 3

Los 2 detectores ven color blanco.
No Hay línea negra.



1. Vamos a utilizar la conexión 2.4G para probar los programas y cuando funcionen correctamente, utilizaremos el cable USB para subir el programa en MODO ARDUINO.
2. Montamos el programa, sujetamos en nuestra mano el robot y pulsamos la bandera para que se ejecute. Si no funciona pulsamos *conectar* → *actualizar firmware*.

```

al presionar bandera
por siempre
  fijar SensorLinea a sigue-lineas Puerto2
  si SensorLinea = 0 entonces
    establecer el led de a bordo todos rojo 0 verde 0 azul 0
    establecer el led de a bordo todos rojo 255 verde 0 azul 0
  si SensorLinea = 1 entonces
    establecer el led de a bordo todos rojo 0 verde 0 azul 0
    establecer el led de a bordo todos rojo 0 verde 255 azul 0
  si SensorLinea = 2 entonces
    establecer el led de a bordo todos rojo 0 verde 0 azul 0
    establecer el led de a bordo todos rojo 0 verde 0 azul 255
  si SensorLinea = 3 entonces
    establecer el led de a bordo todos rojo 0 verde 0 azul 0
    establecer el led de a bordo todos rojo 255 verde 255 azul 255
  
```

En este programa creamos una VARIABLE llamada SensorLinea y le asignamos el valor que esté leyendo nuestro sensor de línea en cada momento.

Dependiendo del valor que tenga la variable el robot hará una cosa u otra.

El proceso se repite de forma continua, **por siempre**.

Prueba el programa y comprueba como funciona porque será muy útil a partir de ahora.

RETO 17: ESTAS ATRAPADO!!

El programa comenzará cuando pulsemos el botón. Debemos crear una variable llamada SensorLinea. Cuando uno de los detectores vea línea negra, debe retroceder, girar a la derecha y volver a avanzar. El robot **NO PUEDE ESCAPAR** del círculo negro. Tendrás que crear también varios bloques: AVANCE, RETROCESO Y GIRO.

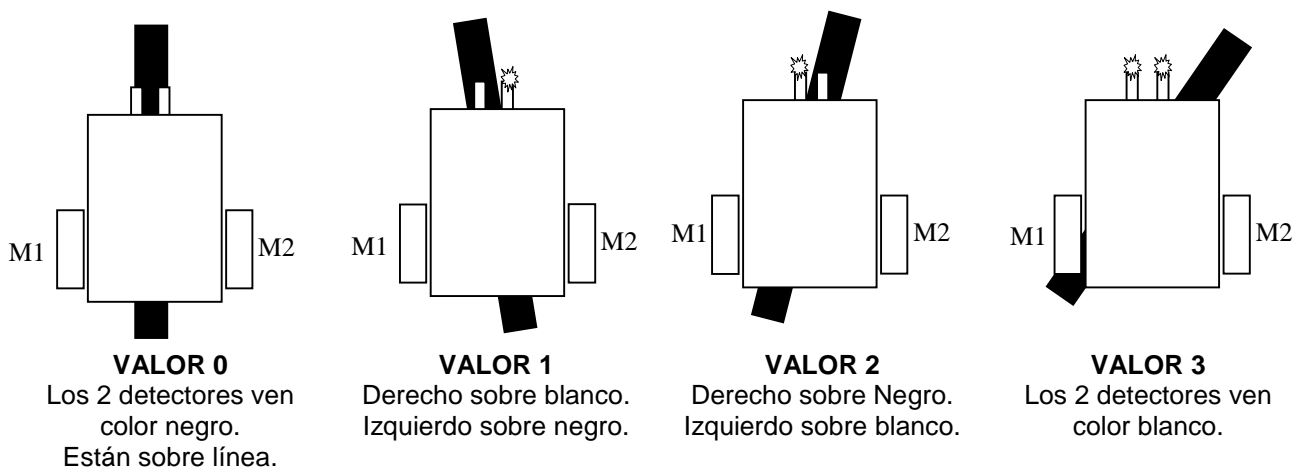
Antes de subirlo en MODO ARDUINO, prueba el programa en modo Scratch (con la banderita) y 2.4G, es más rápido. Recuerda que para cargarlo tienes que usar el cable USB.

FICHA 14: PROYECTO FIN DE CURSO.

Te propongo varios proyectos para que utilices todo lo que has aprendido. Puedes hacerlos todos o pensar en uno propio que sea totalmente original. Al final haremos un concurso con los mejores diseños.

RETO 18: SEGUIDOR DE LÍNEA

Al pulsar el botón de inicio, queremos que nuestro robot sea capaz de seguir una línea negra con curvas sin salirse de ella. Piensa que deben hacer los motores en cada caso. Si van muy rápido se saldrán con facilidad del camino.



RETO 19: ROBOT LUCHADOR

Vamos a organizar un concurso de lucha entre robots. Tu robot tiene que moverse dentro del círculo negro sin salirse de él. Cuando vea o detecte otro robot debe lanzarse a por él y empujarlo fuera del círculo. Intenta programar el robot para que cuando esté fuera del círculo porque otro robot le ha empujado, si no logra entrar de nuevo en un tiempo, se quede quieto. Ha sido vencido y reproducirá los “sonidos de la muerte”.

RETO 20: CARRERAS DE ROBOTS

Demuestra tu habilidad y rapidez conduciendo tu robot por una pista con curvas. Dibuja un camino con dos líneas paralelas (como una carretera). Ahora diseña programa que controle tu robot con el teclado del ordenador portátil o por el mando a distancia IR, si se sale más de 3 veces de la carretera debe pararse aunque no termine el recorrido. El conductor que recorra el camino más rápidamente será el ganador del gran premio.

RETO 21: LABERINTO. Construye un laberinto utilizando cajas de cartón. El robot debe lograr salir en el menor tiempo posible. Para ello, cuando encuentre una pared, deberá detectar hacia donde debe dirigirse para poder continuar avanzando, hasta lograr salir.

RETO 22: CONCURSO DE HABILIDAD. Tienes que realizar un programa en modo arduino utilizando el mando a distancia. Con el mando a distancia tienes que llevar en modo manual, el robot hasta una zona de “inicio” evitando los obstáculos que te coloque el profesor por el camino. Después debes pulsar uno de los botones del mando a distancia y el robot debe llegar EL SÓLO a su aparcamiento y pararse dentro. Ganará el grupo que lo haga en el menor tiempo posible.

