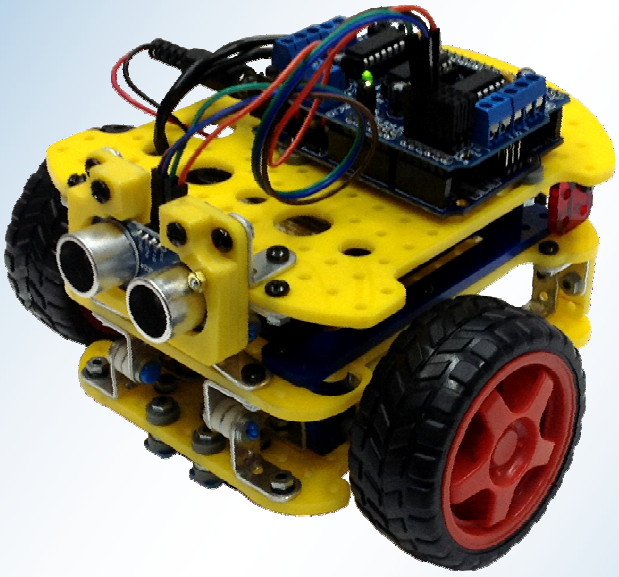


Manual de usuario del Ta-Bot

Guía de teoría y actividades

ROBOTGROUP
robótica para la acción

multiplo®

Autores
Octavio J. da Silva Gillig
Julián U. da Silva Gillig
Mónica J. Paves Palacios

ROBOTGROUP
robótica para la acción

Descargo de responsabilidad / Disclaimer

Él o los autores hacen el mayor esfuerzo posible por garantizar la exactitud de la información presentada en este documento. Sin embargo, ninguno del o los autor/es se responsabiliza/n por los errores o las inexactitudes que puedan aparecer en el mismo. La información contenida en este documento está sujeta a cambios sin previo aviso. Todos los productos, marcas y nombres de firmas o empresas mencionados en este documento son propiedad exclusiva de sus respectivos propietarios. Cualquier omisión o error es absolutamente no intencional.

ESTE TRABAJO, EL SOFTWARE O LOS ELEMENTOS QUE EVENTUALMENTE LO ACOMPAÑEN (SEAN ESTOS DE CUALQUIER CLASE) SON PROVISTOS POR LOS DUEÑOS DE LOS DERECHOS INTELECTUALES Y POR QUIENES CONTRIBUYERON A SU DESARROLLO "COMO SON", RENUNCIANDO ELLOS A CUALQUIER TIPO DE GARANTÍA EXPLÍCITA O IMPLÍCITA, INCLUYENDO, AUNQUE NO LIMITÁNDOSE, A LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN Y DE ADAPTACIÓN A PROPÓSITOS PARTICULARES. BAJO NINGUNA CIRCUNSTANCIA SERÁN LOS DUEÑOS DE LOS DERECHOS INTELECTUALES Y QUIENES CONTRIBUYERON AL DESARROLLO RESPONSABLES POR DAÑO ALGUNO, DIRECTO, INDIRECTO, INCIDENTAL, CASUAL, CAUSAL (INCLUYENDO PERO NO LIMITÁNDOSE A DAÑOS A LA VIDA Y/O A LA PROPIEDAD, PÉRDIDA DE DATOS, LUCRO CESANTE, INTERRUPCIÓN DE NEGOCIOS), AUNQUE EL MISMO OCURRA BAJO CUALQUIER TEORÍA DE DERECHO, PRODUCIDO EN CUALQUIER FORMA DE USO DE ESTE DESARROLLO O DESARROLLOS DE ÉL DERIVADOS, AÚN CUANDO SE AVISE O NO DE DICHO DAÑO O SU POSIBILIDAD.

Contenido

1. Introducción

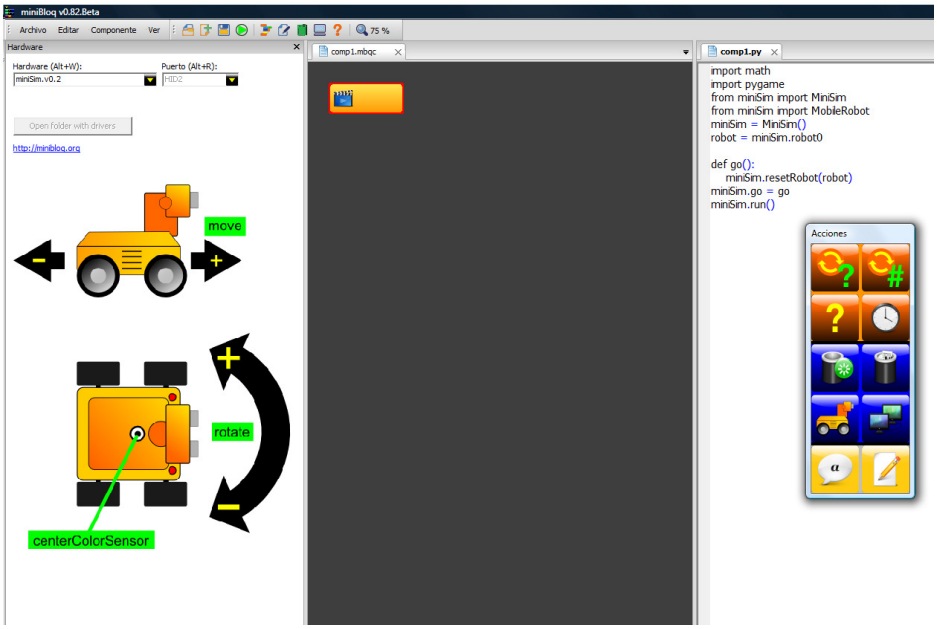
Introducción

El Ta-Bot es un robot de bajo costo fabricado por RobotGroup. Gracias a su precio accesible permite que más chicos (y no tan chicos) puedan acercarse al apasionante mundo de la Robótica. También es ideal para ser utilizado en el aula de clases ya que, al igual que otros productos fabricados por RobotGroup, puede ser programado con miniBloq, un entorno de programación de código abierto pensado especialmente para la educación.

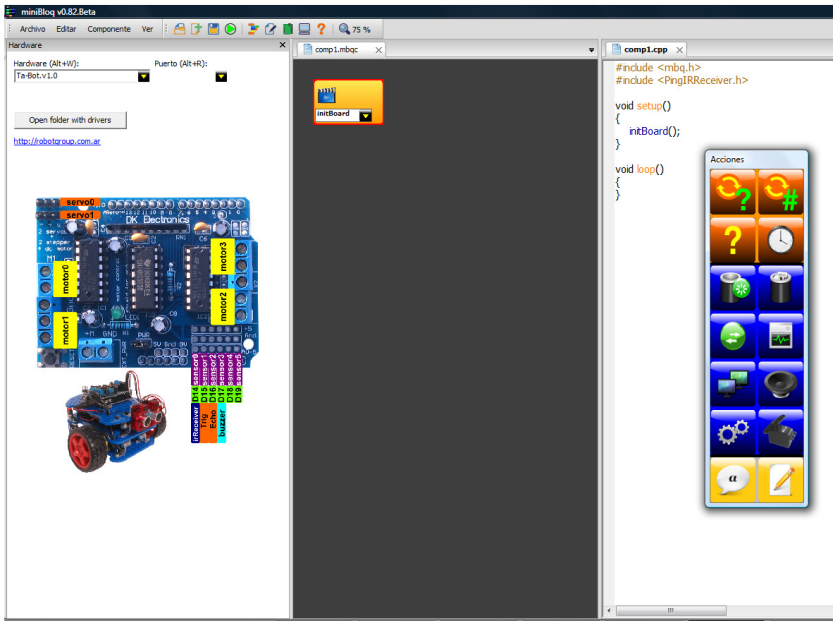
Empezar a usar miniBloq

El miniBloq es un software de código abierto desarrollado para programar robots. Su instalación es muy sencilla ya que, descargando el archivo miniBloq.v0.82.Beta.exe y siguiendo los pasos de instalación, ya se genera un acceso directo en el escritorio y se puede empezar a utilizar.

Lo primero que hay que hacer al abrir el miniBloq es seleccionar el robot que vamos a utilizar. Esto se hace a través del menú desplegable de la solapa "Hardware".



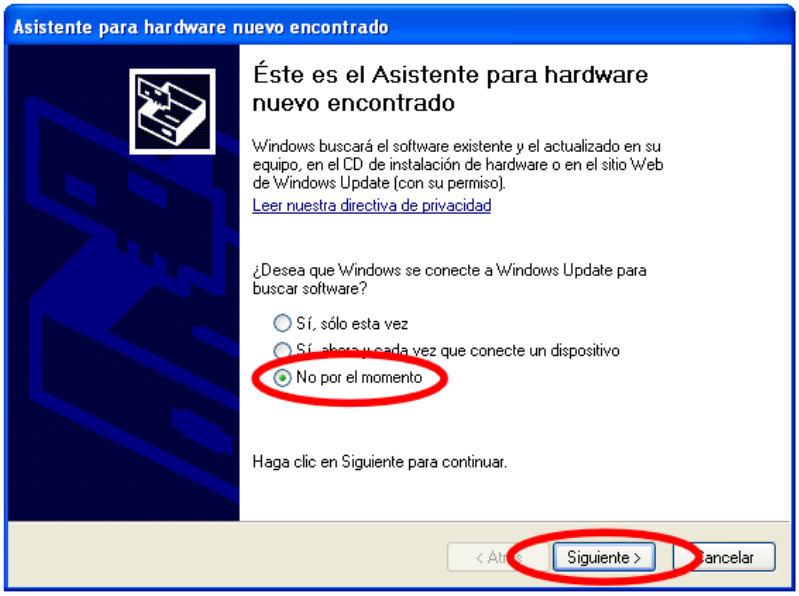
Una vez que hayamos seleccionado al robot correspondiente, que en nuestro caso será Ta-Bot.v1.0, miniBloq nos mostrará una foto del mismo junto a una imagen con el detalle de las conexiones del robot. Tal como se muestra en la siguiente figura.



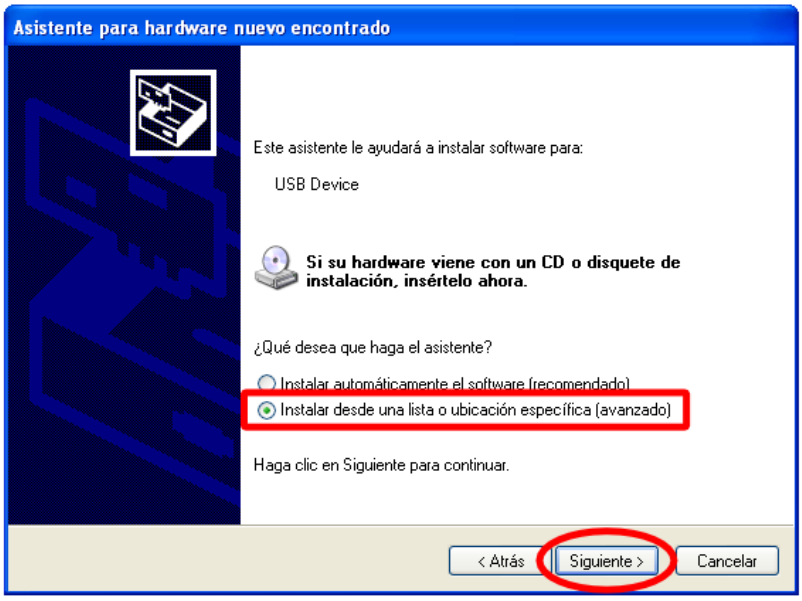
Instalación de drivers de comunicación

Una vez que tengamos instalado el entorno de programación miniBloq, tendremos que instalar los drivers de comunicación correspondientes a la placa controladora del robot para así poder comunicarlo con la computadora.

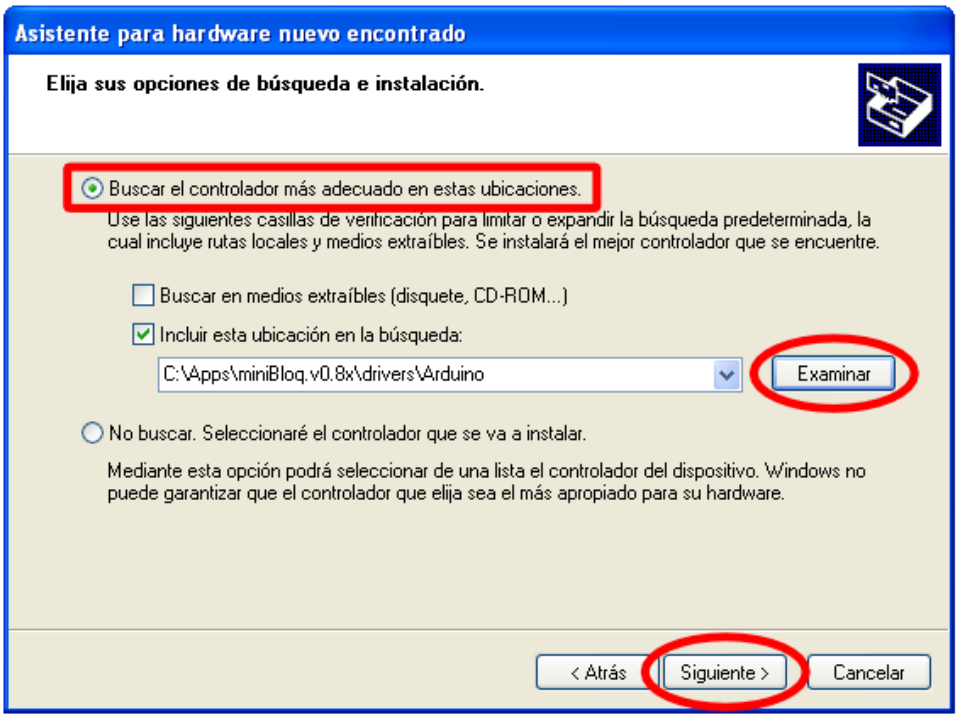
Lo primero que hay que hacer para instalar los drivers de comunicación es conectar el robot con la computadora utilizando un cable USB. En ese momento, veremos en la pantalla una ventana del "Asistente para hardware nuevo encontrado" en la que tendremos distintas opciones para seleccionar. En nuestro caso, tenemos que elegir "No por el momento" y luego presionar "Siguiente", como se muestra en la siguiente figura.



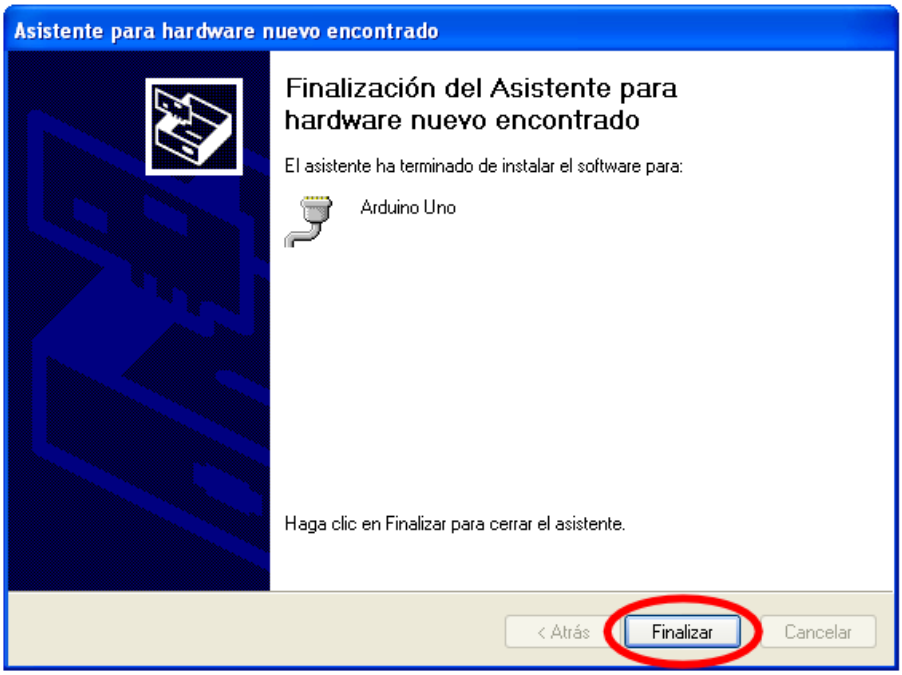
A continuación, el asistente nos preguntará acerca de la ubicación de los drivers para poder instalarlos. Seleccionamos la opción "Instalar desde una lista o ubicación específica (avanzado)". Luego presionamos "Siguiente". Como se muestra en la siguiente imagen.



Luego, seleccionamos la opción "Buscar el controlador más adecuado en estas ubicaciones." y seleccionamos la ubicación en la que el "Asistente para hardware nuevo encontrado" buscará los archivos necesarios. En el caso del Ta-Bot hay que seleccionar la carpeta "..miniBloq.v0.8x\drivers\Arduino". Como se muestra en la siguiente figura.

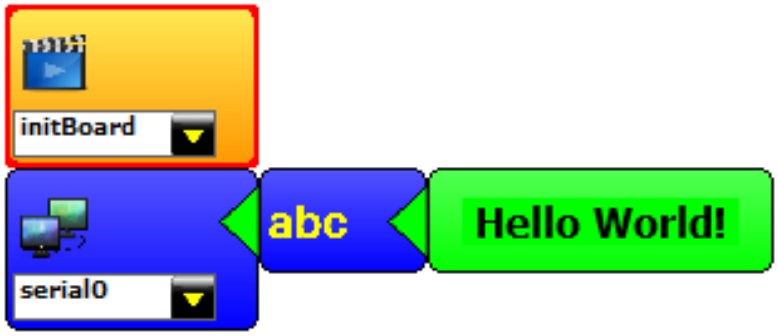


Al finalizar este proceso, veremos el siguiente cartel por pantalla indicándonos que la instalación se realizó correctamente. Presionamos "Finalizar".



Ejemplo de prueba en miniBloq

Para probar la comunicación entre el Ta-Bot y el entorno miniBloq, vamos a realizar un pequeño programa de prueba que nos servirá para asegurarnos de que el robot y la computadora se están comunicando correctamente. El programa final se verá como se muestra a continuación:

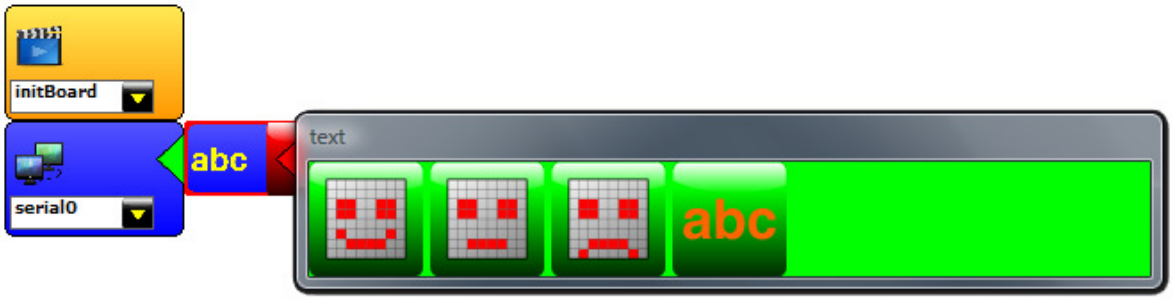


Explicación del programa de prueba, paso a paso

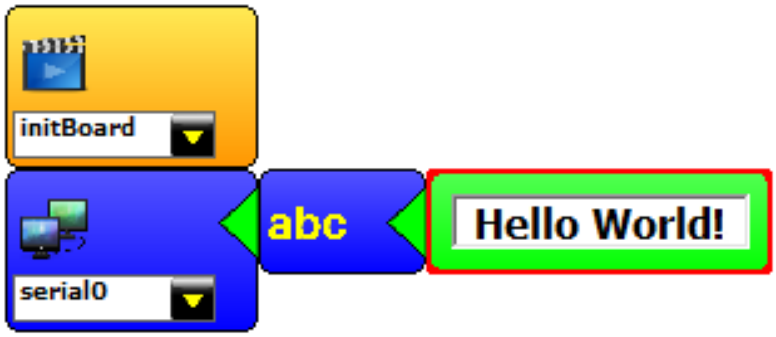
Este programa mostrará por pantalla la frase "Hello World!" utilizando la comunicación serial a través del cable USB. Lo primero que hay que hacer es presionar el botón que genera el bloque que nos permitirá enviar datos a través del puerto USB. El mismo es el que se muestra seleccionado en la imagen que está a continuación.

Luego, se presiona el triángulo rojo del bloque de comunicación con lo cual podremos ver las distintas opciones que tenemos para asignarle al bloque en cuestión. Como se muestra a continuación, se abrirá un menú desplegable.

Seleccionamos el botón que tiene las letras "abc" ya que mandaremos un texto como mensaje desde la placa a la computadora. Nuevamente, tenemos que agregar un bloque entre las opciones de las que disponemos al presionar el triángulo rojo de la última orden elegida.



Seleccionamos la opción que tiene las letras "abc" la cual nos permitirá ingresar un texto a elección para ser transmitido. A continuación, ingresamos el texto en el especio en blanco que nos generó. En nuestro ejemplo, el texto ingresado es "Hello World!", como se muestra en la imagen.

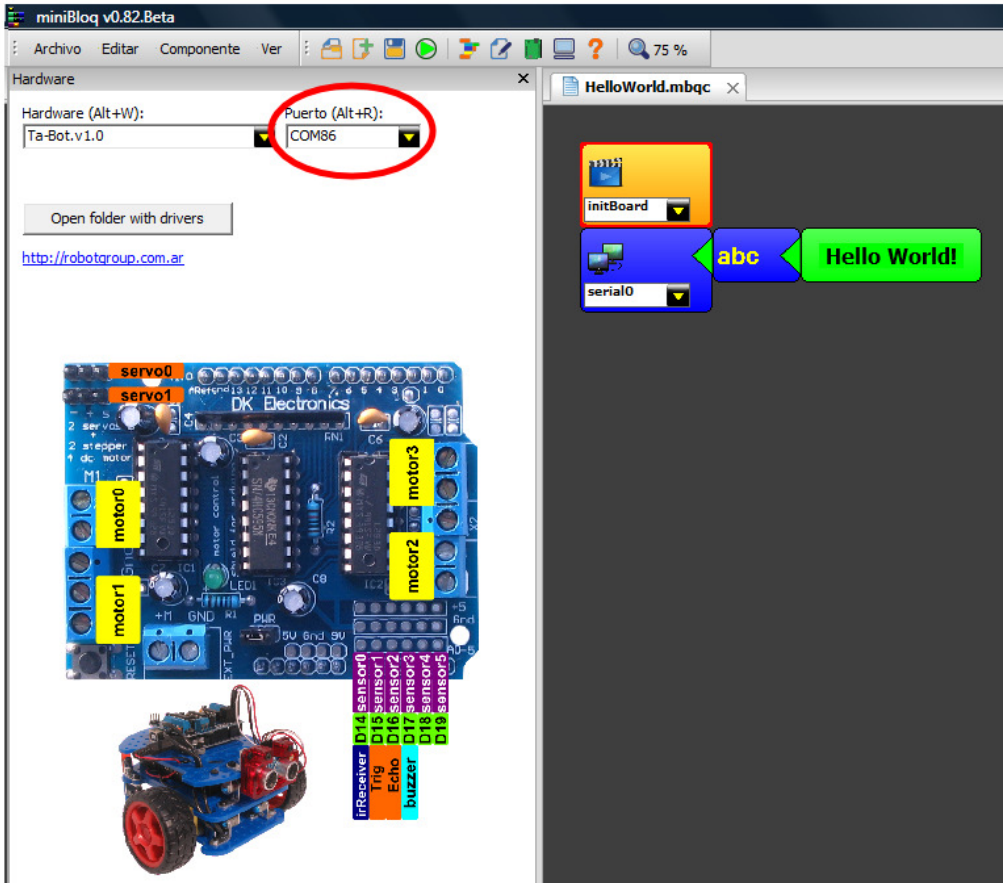


Descarga del programa a la placa

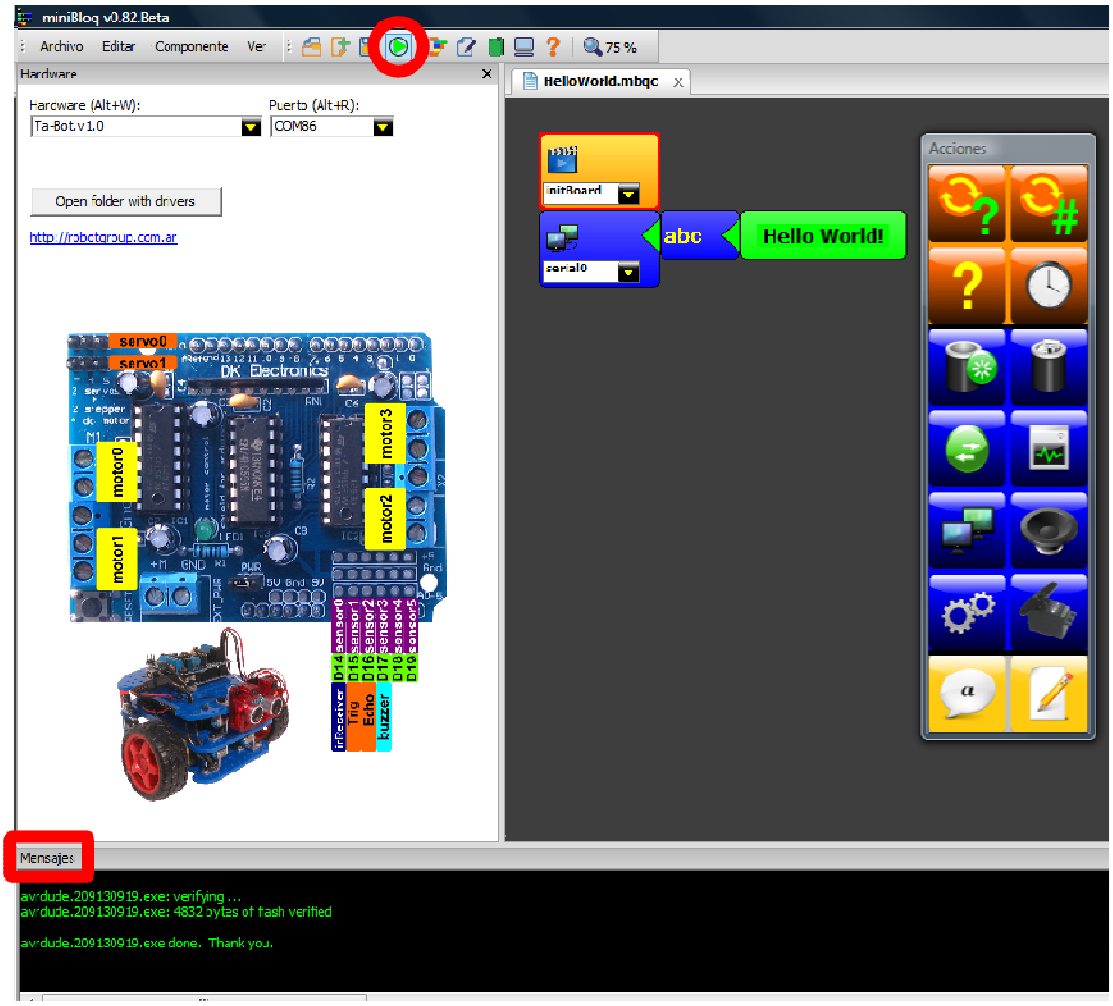
Una vez que terminamos de escribir el programa que queremos que el robot ejecute, teniendo seleccionado al Ta-Bot en la solapa "Hardware", seguimos los siguientes pasos para descargar nuestras órdenes a la placa controladora.

1- Conectamos el robot a la computadora a través del cable USB. En este momento vamos a notar que se enciende un LED verde en la placa superior del robot y, si miramos debajo del mismo, vamos a ver que la placa controladora también encendió un LED verde y, además, uno rojo.

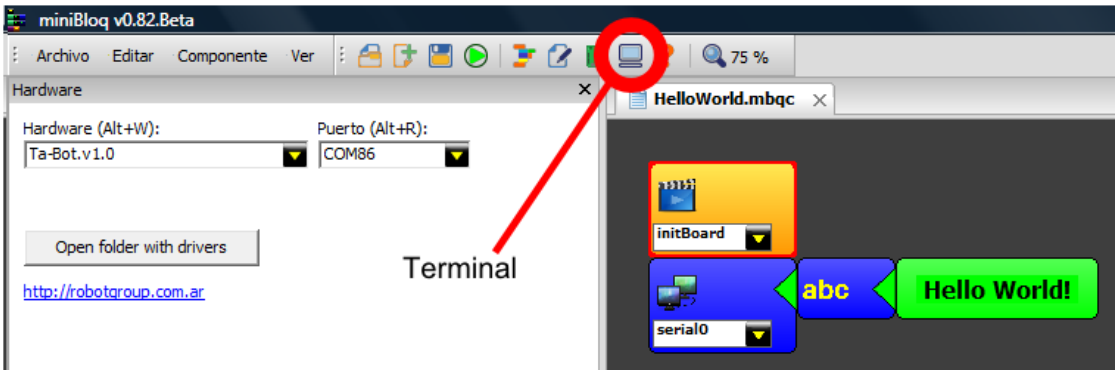
2- Seleccionamos el puerto correspondiente en el menú "Port (Alt+R)". Por lo general, el puerto generado para comunicar el robot con la computadora es el más alto ya que es el último que abrió la computadora. En el ejemplo de la imagen es el "COM86".



3- Una vez configurada la conexión entre el robot y la computadora, presionamos el botón "Ejecutar" el cual hará que miniBloq revise que el programa no tenga errores de compilación y luego, si está todo bien, lo descargará a la placa controladora del robot. En la parte inferior de la pantalla se abrirá una pantalla de mensajes que nos informará el estado del proceso de compilación y descarga.



4- Luego de que el programa se cargó en la placa controladora del robot, presionamos el botón "Terminal" que abrirá la pantalla de comunicaciones.

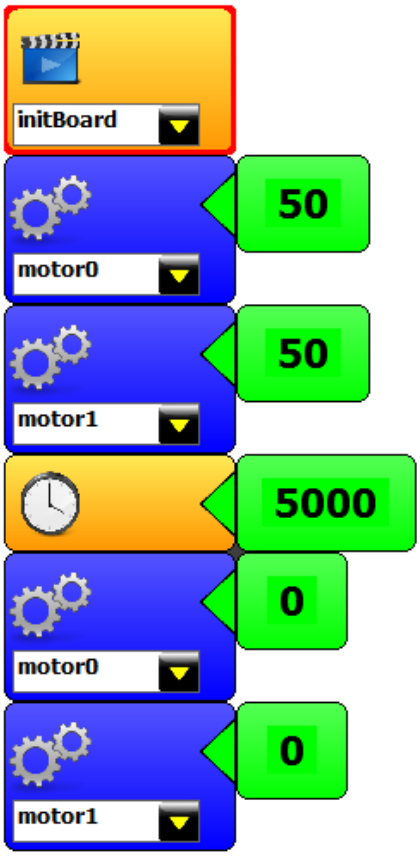


5- La pantalla de comunicaciones se abrirá debajo de la ventana "Hardware". Para ver el mensaje que nos manda el robot, presionamos el botón "Abrir" para iniciar la transmisión de datos. Finalmente, la salida del programa se verá como se muestra en la imagen que está a continuación.

2. Motores

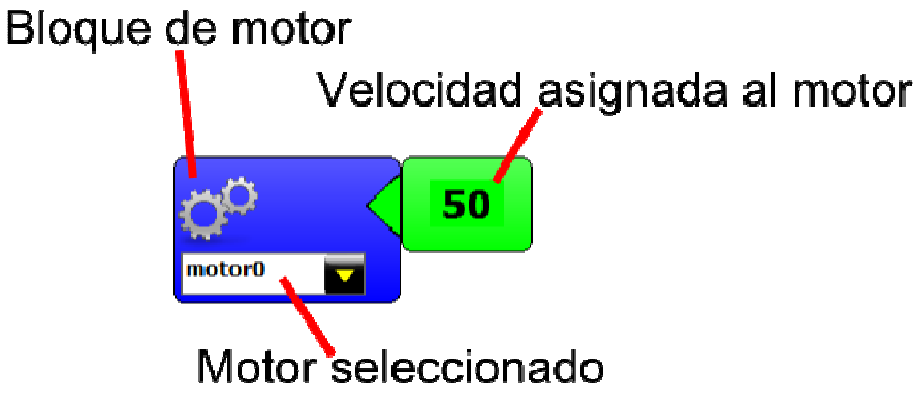
Actividad 2.1: avanzar en línea recta

En esta primera actividad queremos hacer que el robot avance por un tiempo determinado en línea recta y que luego se detenga. Utilizando el entorno de programación miniBloq es muy sencillo. Simplemente hay que hacer el siguiente programa:

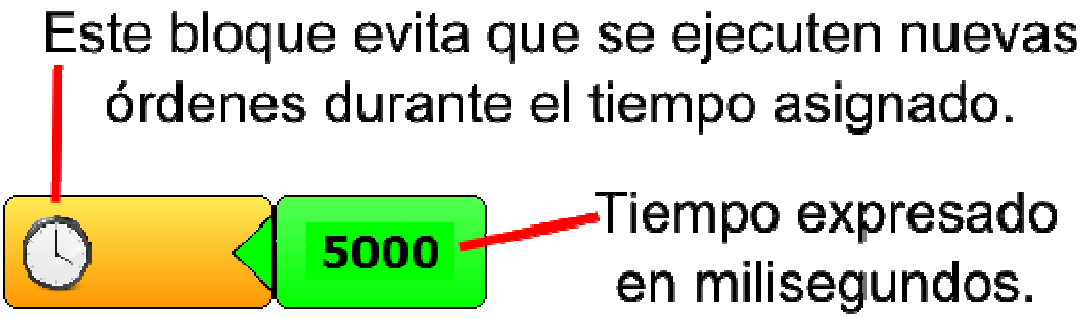


Explicación de la actividad 2.1 hecha en miniBloq

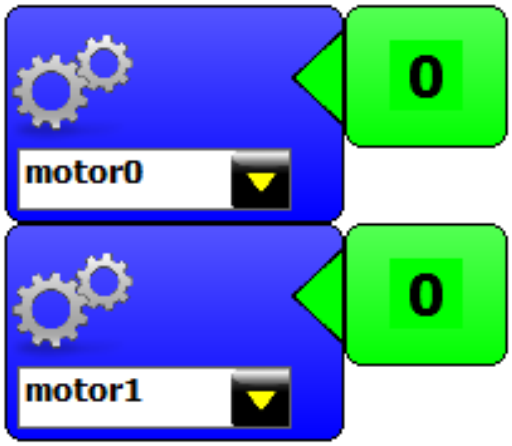
Lo primero que hacemos es seleccionar el bloque de motor. Dentro de este, seleccionamos el número de motor que queremos que avance. En este caso, el motor 0. El motor 0 es el que está conectado mediante un cable de dos hilos a la conexión designada como motor0 en el gráfico que se encuentra en la solapa "Hardware". Por último, con el símbolo "#" (numeral), asignamos la velocidad ingresándola como un número.



Como queremos que el robot avance durante un tiempo determinado y luego frene, asignamos velocidad 50 al motor 1 y luego le asignamos tiempo. Para determinar la duración del avance seleccionamos el bloque que contiene un reloj al que le asignamos el tiempo correspondiente. En nuestro ejemplo escribimos el número "5000" porque queremos que avance durante 5 segundos. El tiempo está expresado en milisegundos.

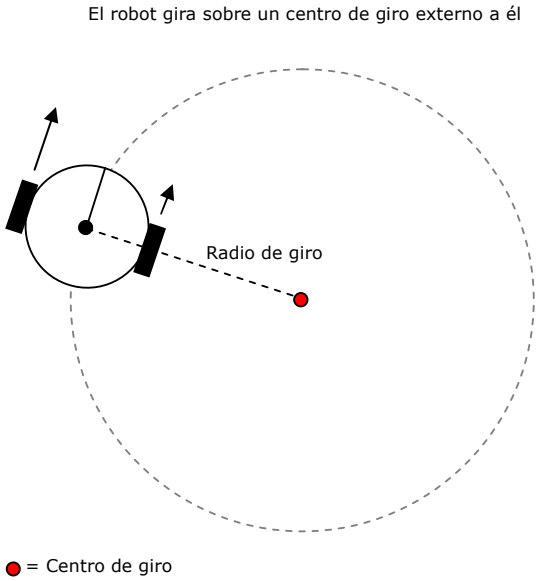


Por último, le asignamos potencia igual a cero a cada motor para frenar al robot. De no agregar estos dos bloques, el robot seguirá avanzando sin detenerse aunque se haya establecido el tiempo en los bloques anteriores.

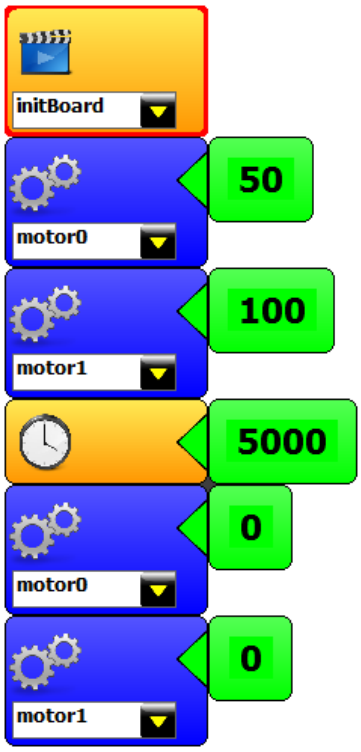


Actividad 2.2: dibujar una circunferencia

En esta actividad queremos hacer que el robot dibuje una circunferencia. Tal como se muestra en la figura que está a continuación, el centro de giro del robot será el punto central de la circunferencia que este dibuje.



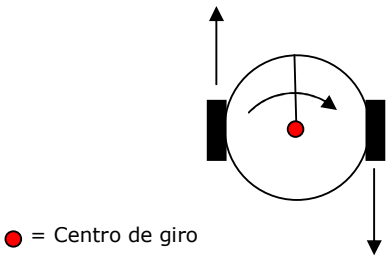
Para realizar esta actividad utilizando el entorno miniBloq, tenemos que utilizar los siguientes bloques. Hay que tener en cuenta que, para que el robot gire, un motor avanzará más rápido que el otro pero los dos tendrán velocidad.



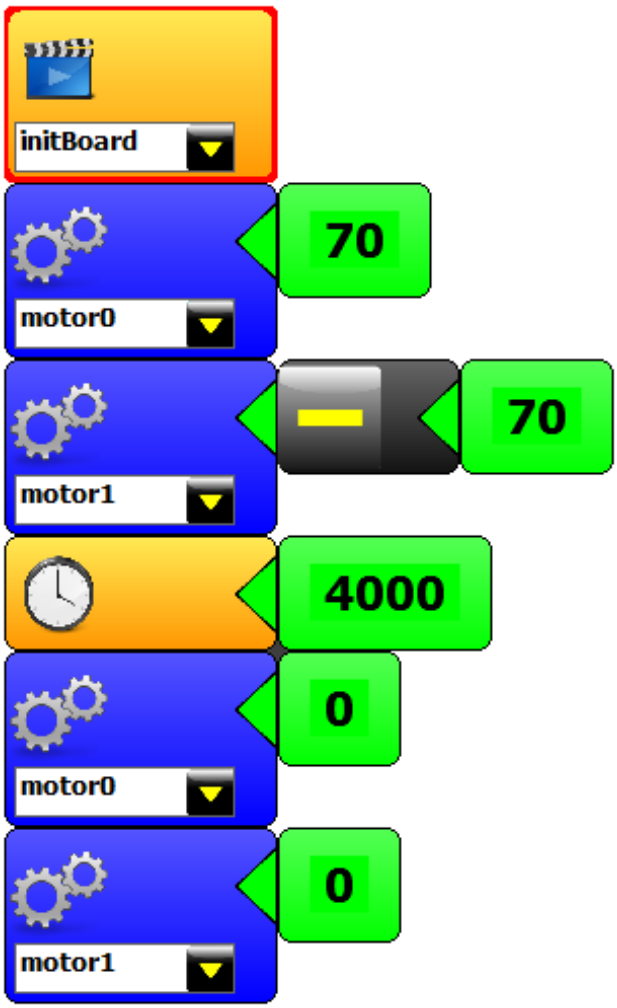
Actividad 2.3: hacer que el robot gire sobre su propio eje

En esta actividad queremos hacer que el robot gire en el lugar donde está sin desplazarse. Tal como lo muestra el esquema que está a continuación, tendremos que hacer que una rueda avance y la otra retroceda, pero que ambas lo hagan a la misma velocidad.

El robot gira sobre su propio centro



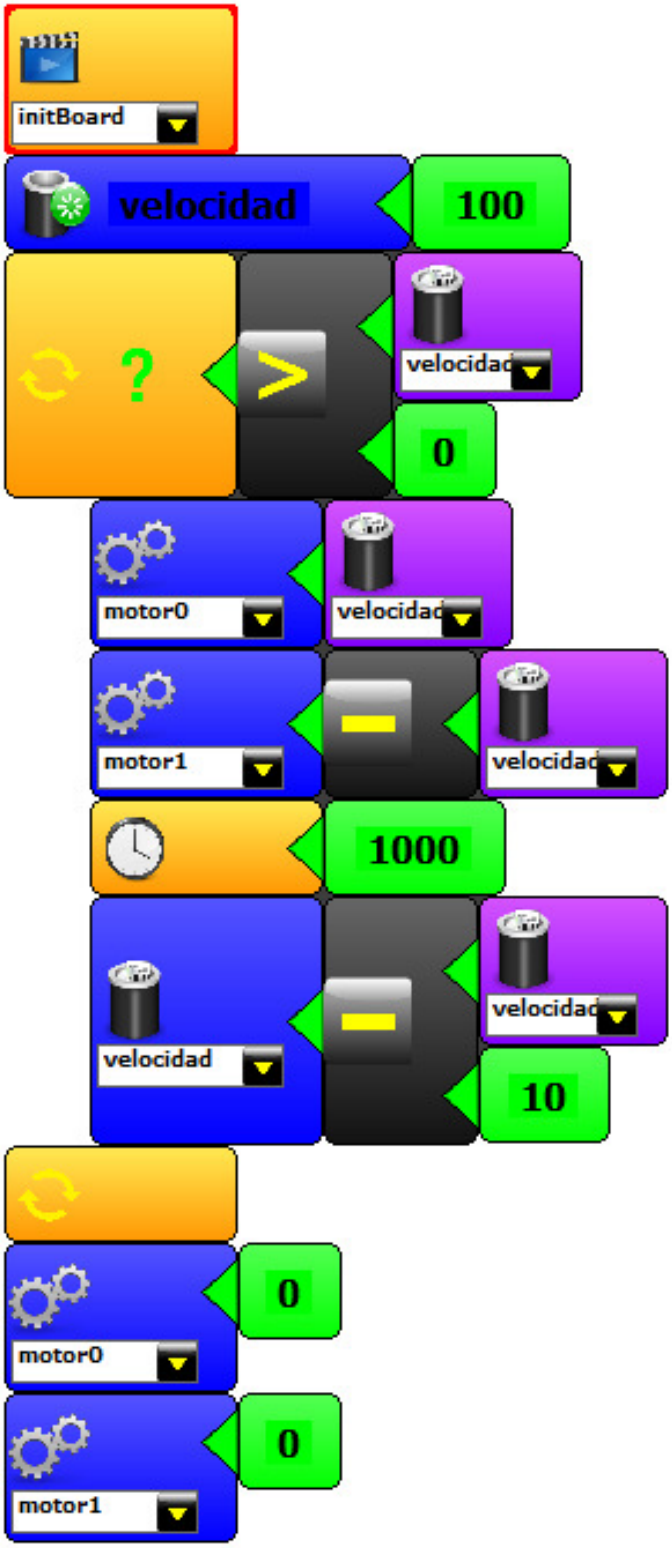
Para realizar esta misma actividad utilizando el entorno miniBloq, tenemos que agregar los bloques que se muestran a continuación.



En este caso, utilizamos un signo negativo para hacer que uno de sus motores avance en sentido contrario al otro. Para esto, tenemos que seleccionar la opción marcada como "-(x)", tal como se muestra en la imagen que está a continuación.

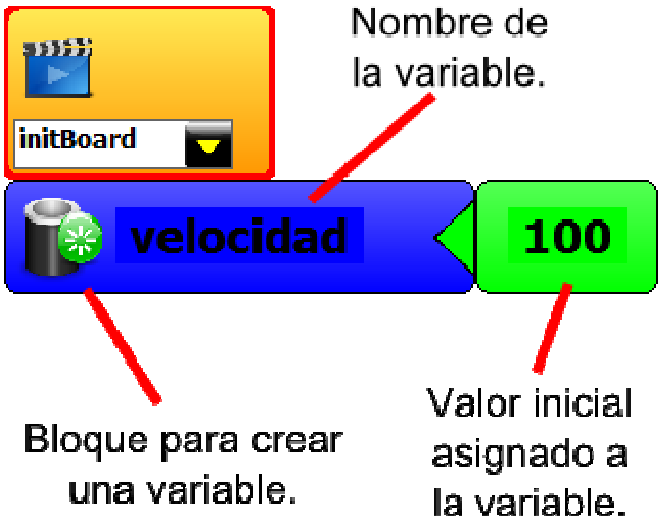
Actividad 2.4: hacer que el robot desacelere hasta frenar

En esta actividad queremos que el robot empiece girando sobre su centro a una alta velocidad y luego comience a frenarse hasta detenerse. Para eso, podemos usar los siguientes bloques:

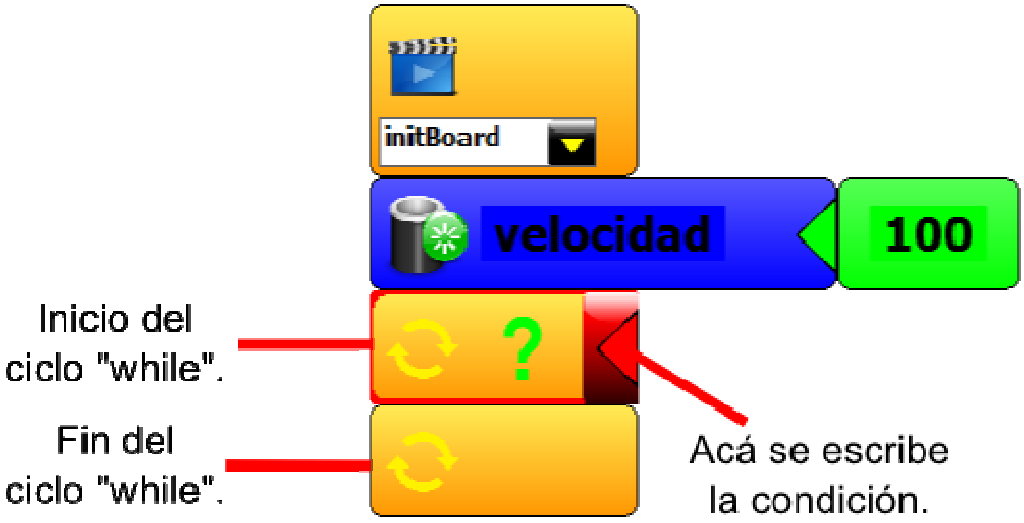


Explicación de la actividad 2.4: Variables en programación

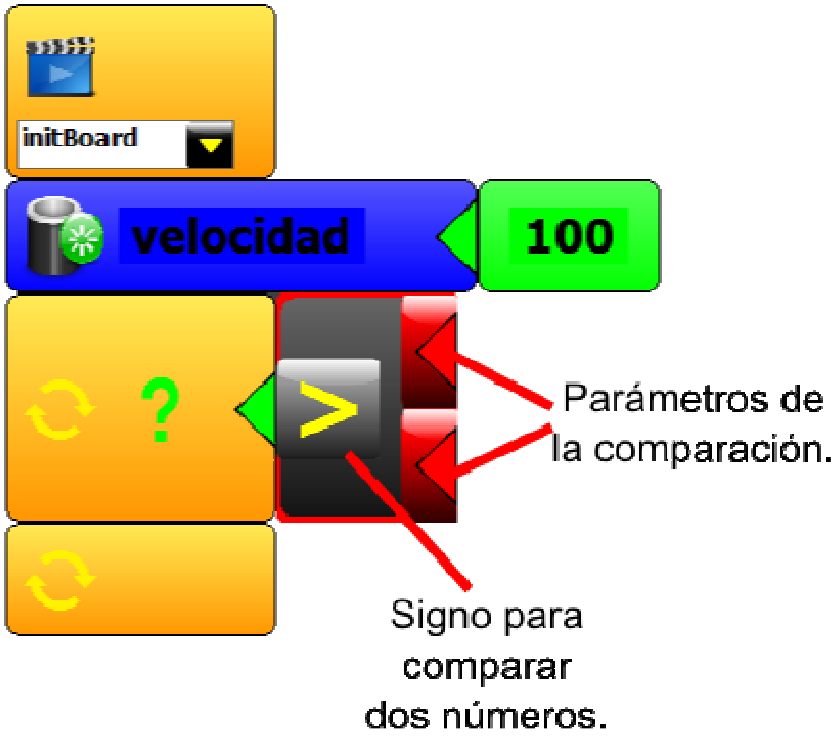
En programación, una variable es simplemente un espacio de memoria en el que guardamos algo. Para acceder a ese algo, lo hacemos a través del nombre que le asignamos a la variable cuando la creamos. Luego, podemos modificar el contenido de ese espacio de memoria tantas veces como sea necesario. Por ejemplo, si quisiéramos hacer una variable que represente la velocidad de los motores, podríamos declararla de la siguiente manera:



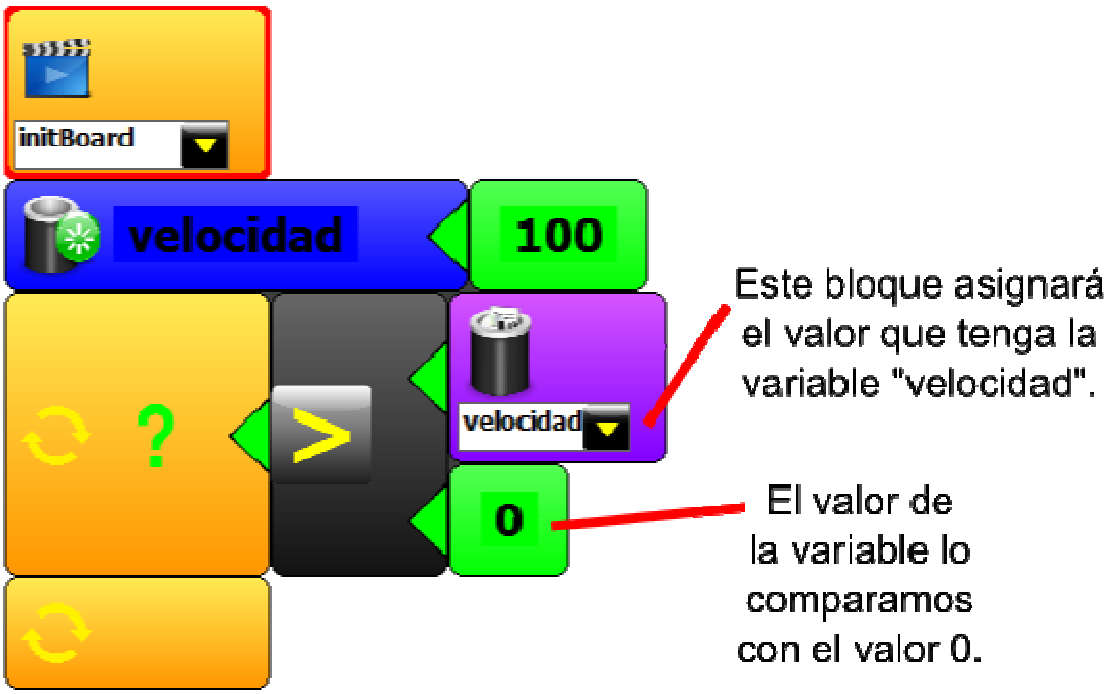
Para continuar con el código, agregamos un ciclo "while". Este nos generará dos bloques en vez de uno ya que, todo lo que quede encerrado entre estos dos bloques, se ejecutará siempre que la condición asignada a dicho ciclo sea verdadera. La condición se agrega presionando el triángulo rojo del bloque que tiene el signo de interrogación.



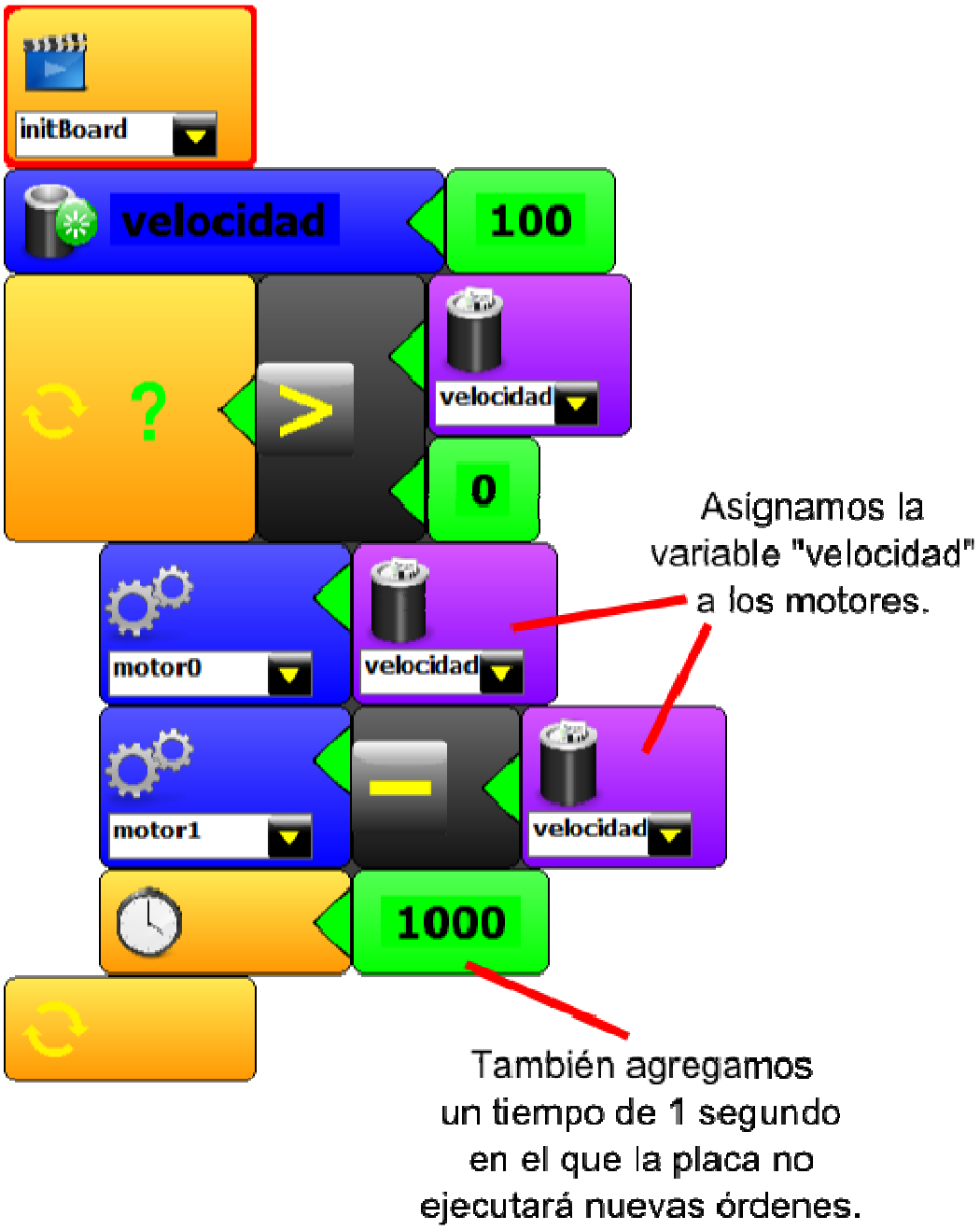
Al primer bloque del ciclo le agregamos una condición presionando la flecha roja. En nuestro caso, para la condición seleccionaremos en el menú desplegable un signo de comparación aritmética.



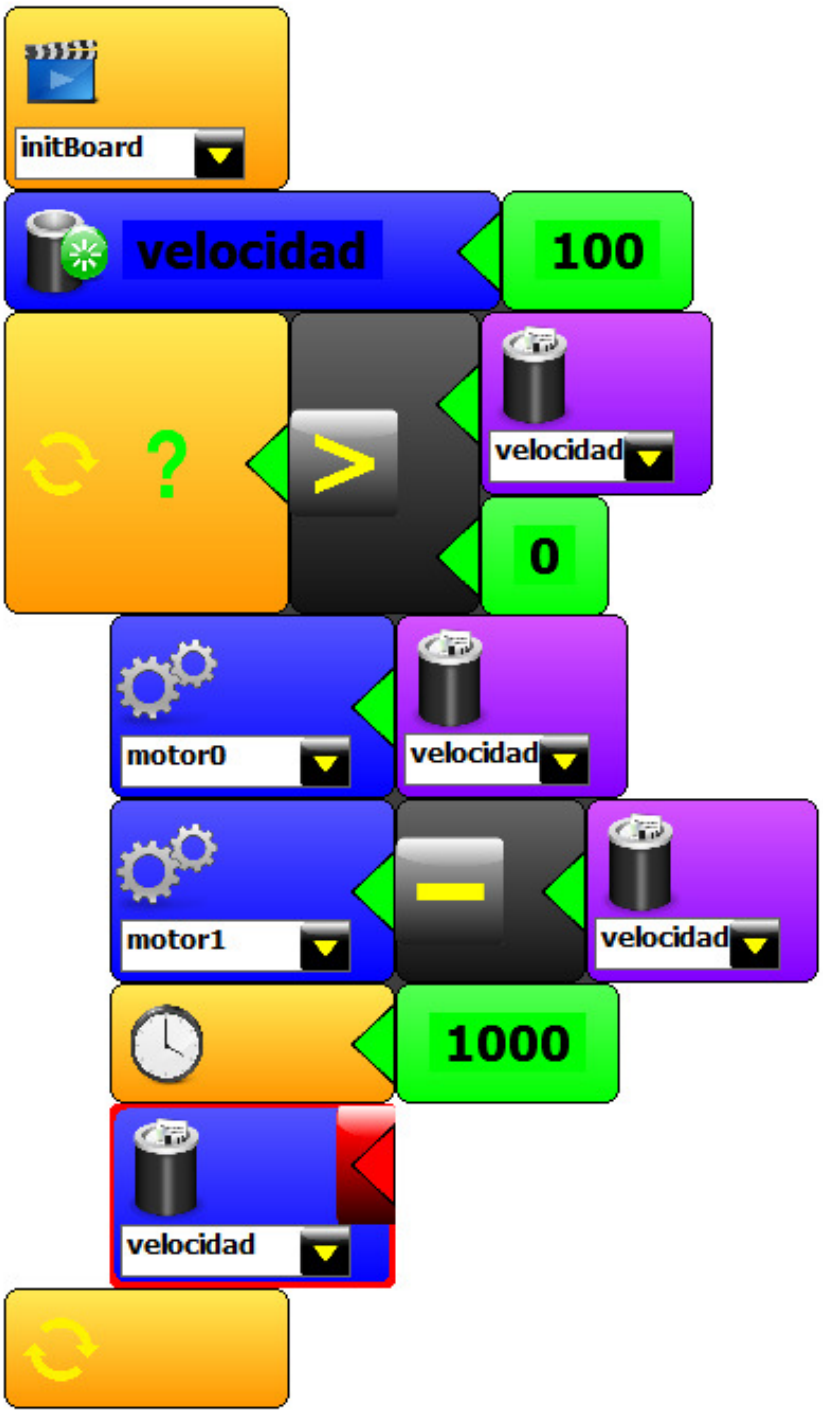
Una vez que agregamos el signo para la comparación, completamos ambos lados de la misma. Uno de los parámetros que recibirá esta comparación será la variable en sí, el otro, un número que representará el límite que queremos alcanzar. En este caso, ese límite será cero ya que queremos que el robot decremente su velocidad hasta llegar a detenerse.



Dentro del ciclo, asignamos la velocidad guardada en nuestra variable a los motores. Uno con un signo negativo y otro sin él. Luego, agregamos un tiempo igual a 1 segundo para que la placa no ejecute nuevas órdenes.

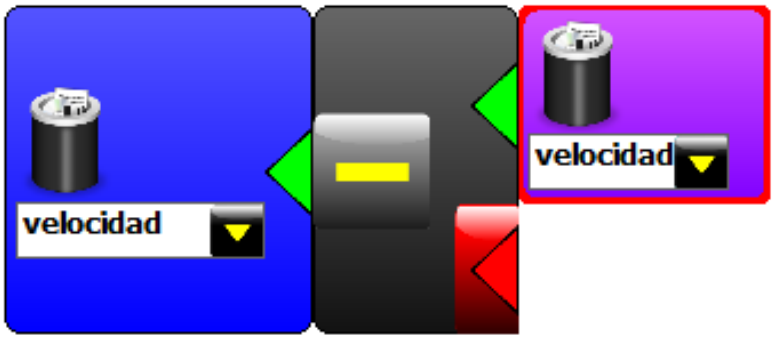


Si el código queda como está, el ciclo se ejecutará siempre y la velocidad de los motores no se reducirá nunca. Por eso, lo que tenemos que hacer a continuación es modificar el valor de la variable de tal manera que, por cada ejecución del ciclo, su valor se reduzca. Para eso, lo primero que tenemos que hacer es seleccionar el bloque con el que asignamos un nuevo valor a la variable "velocidad".

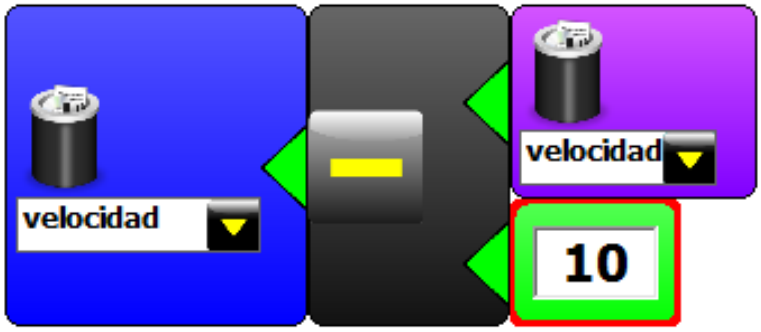


A continuación, a la variable "velocidad" le asignamos una resta como parámetro. No hay que confundir el bloque que se usa para restar del bloque que se usa para cambiar el signo de un número.

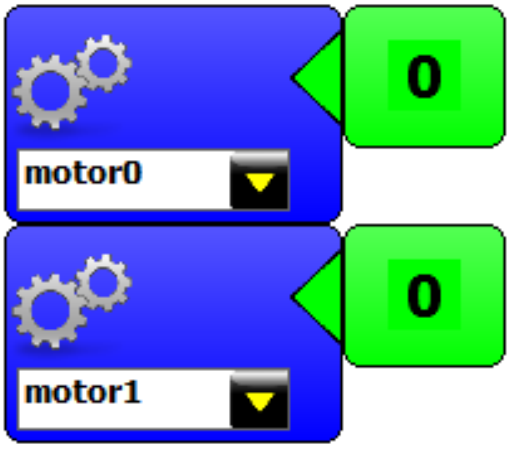
Como la resta necesita dos parámetros numéricos, asignamos primero a la variable "velocidad" ya que es el valor que queremos que se reduzca.



El segundo parámetro de esta resta es el número que servirá para modificar a la variable "velocidad". En nuestro caso, elegimos "10".



Por último, y por fuera del ciclo, le pedimos al robot que frene ambos motores.



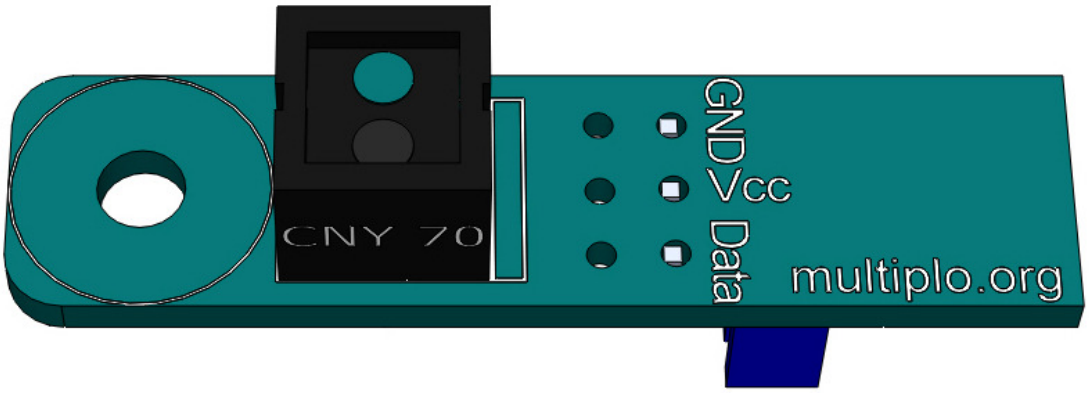
3. Sensores

Sensores Infrarrojos (IR)

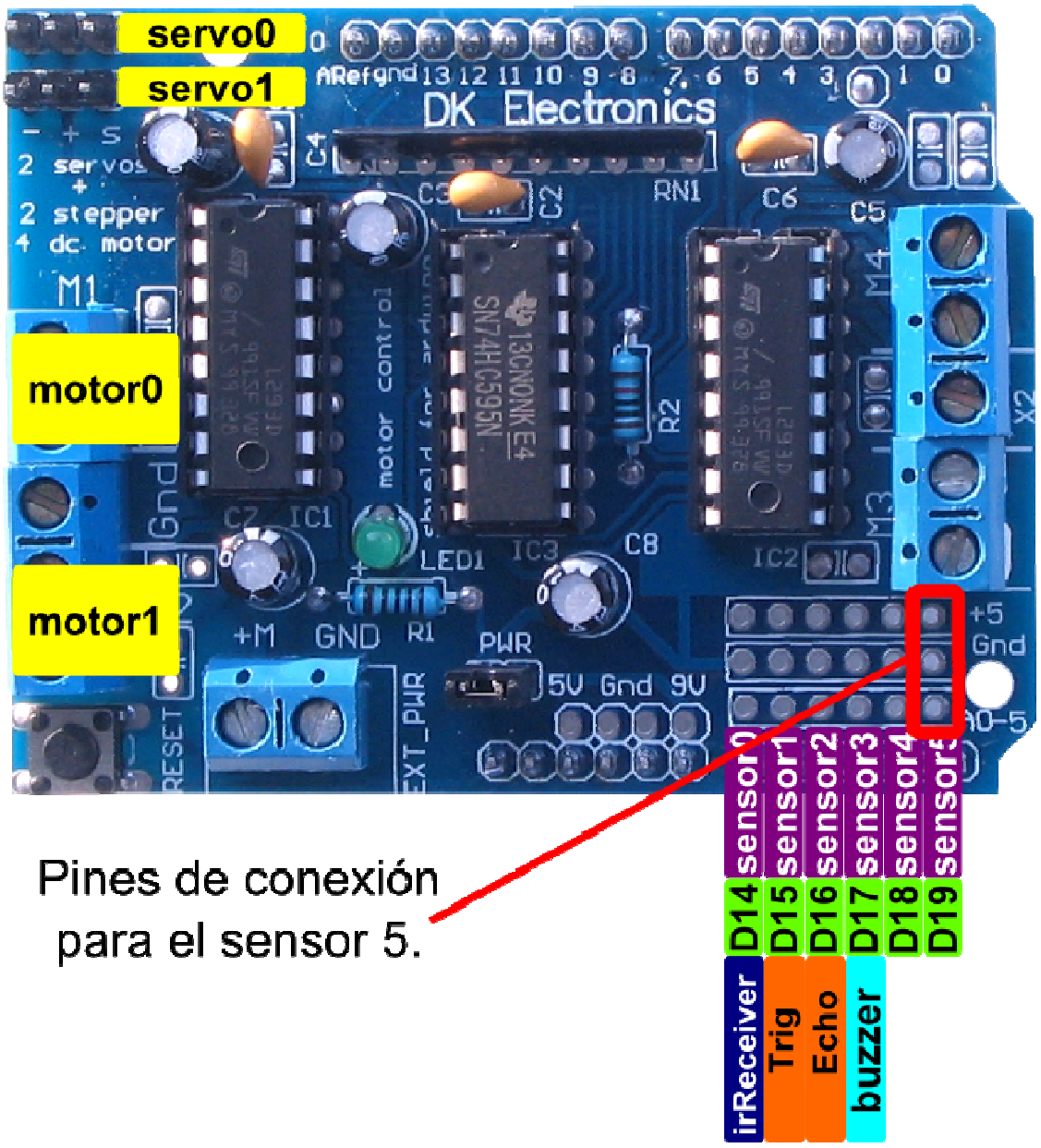
Los sensores IR CNY 70 funcionan emitiendo un haz de luz infrarrojo invisible. Este, luego de rebotar contra un objeto, vuelve al sensor, el cual capta la intensidad del regreso. Esta intensidad es convertida en un número real con el cual trabajaremos en las actividades que proponemos a continuación.

Actividad 3.1: testeo de un sensor IR CNY 70

En esta actividad vamos a testear el funcionamiento de un sensor infrarrojo (IR). Lo primero que tenemos que hacer es conectar el sensor correctamente a la placa. Para ello, notemos que los sensores IR tienen tres pines de conexión nombrados como "GND", "Vcc" y "Data". Como se muestra en la siguiente figura.



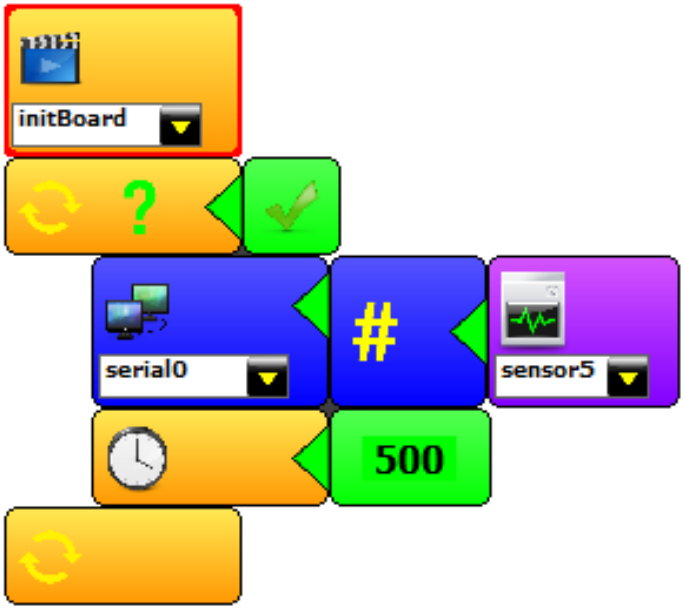
Como tenemos tres pines en el conector del sensor usaremos un cable con tres hilos o tres cables individuales para conectar este tipo de sensores a la placa. Por lo tanto, los cables se conectarán en un extremo al sensor y en el otro extremo a la placa. En el esquema que se encuentra en el entorno miniBloq tenemos numeradas las entradas/salidas de sensores. En nuestro caso, vamos a probar un sensor conectado a A5 por lo que lo llamaremos como sensor 5 desde el software.



Para esta prueba, vamos a conectar los cables de la siguiente manera:

Color del cable	Pin del sensor	Conector de la placa
ROJO	Vcc	+5
NEGRO	GND	Gnd
BLANCO	Data	A5

Una vez que el sensor está conectado, bajamos el siguiente programa a la placa controladora del robot.

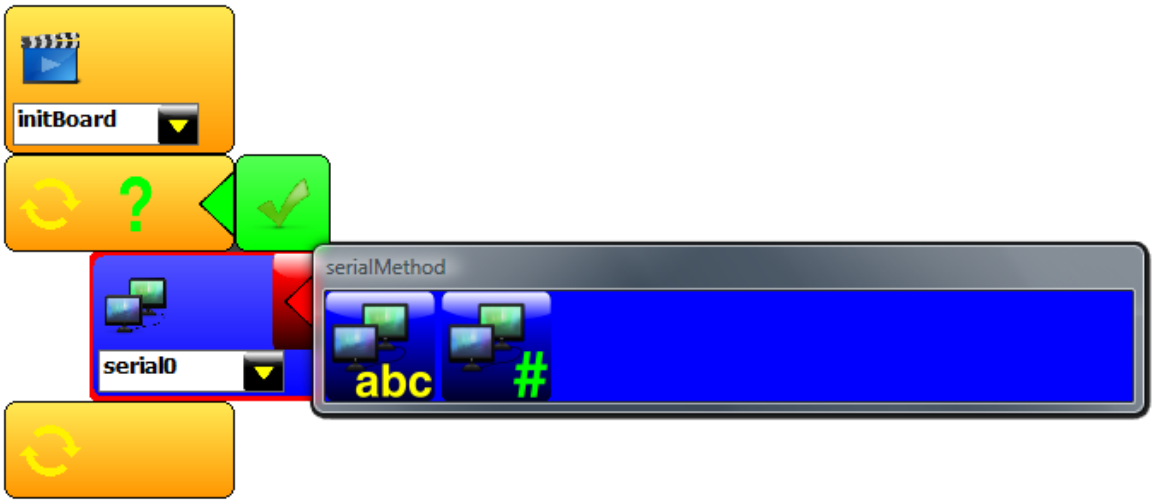


Una vez descargado el programa a la placa y teniendo el robot conectado a la computadora, presionamos el botón "Terminal" y luego el botón "Abrir" para ver lo que comunica el robot.

Explicación de la actividad 3.1: testeo de un sensor IR

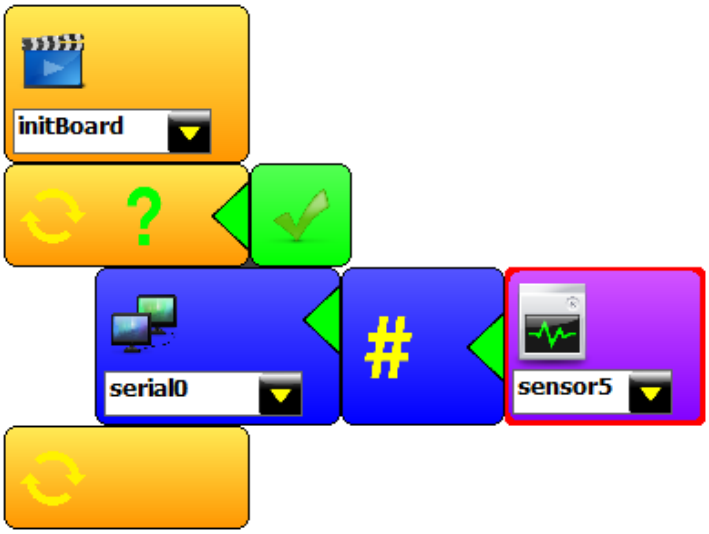
Lo primero que hay que hacer para testear un sensor es crear un ciclo infinito. Esto nos permitirá hacer distintas mediciones sin tener que resetear la placa. El primer bloque que seleccionamos es el que utilizamos para hacer ciclos "while" y luego seleccionamos la tilde para indicarle que la condición será siempre verdadera.

Luego, dentro del ciclo que acabamos de crear, agregamos un bloque de comunicación y a este le asignamos un bloque que le indicará que el valor que recibirá será numérico.

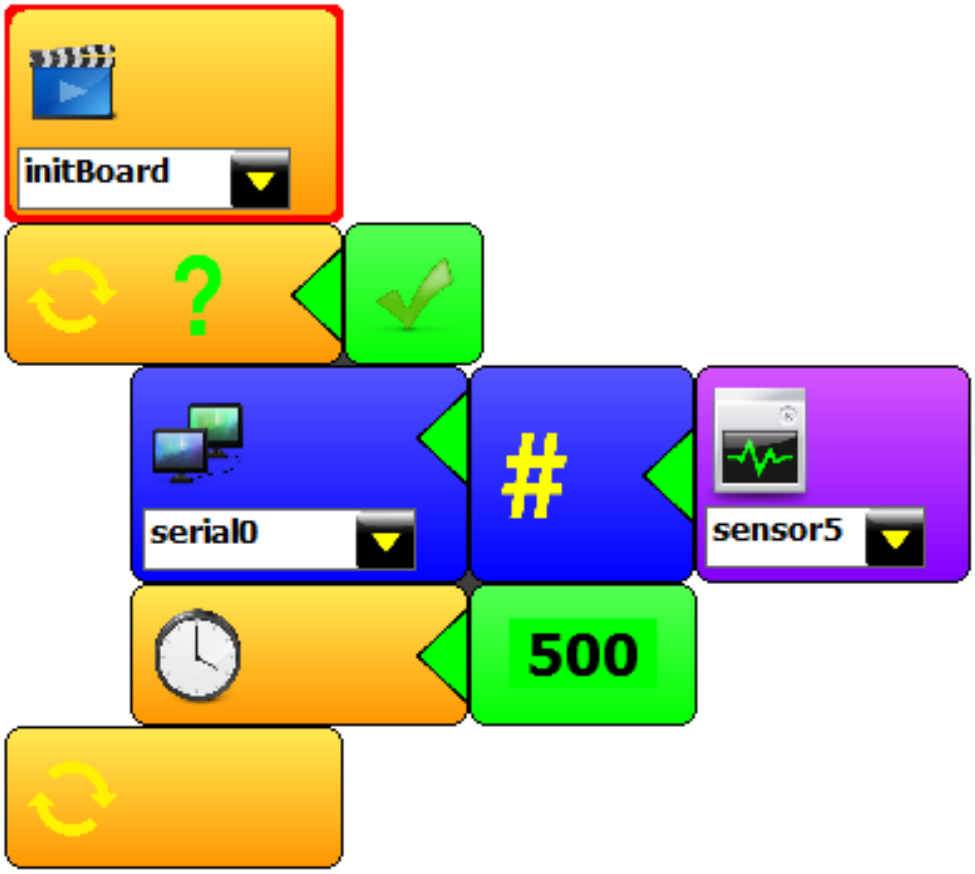


A continuación, asignamos como parámetro numérico lo detectado por una entrada.

En el menú desplegable del bloque seleccionamos la opción "sensor5" que será nuestra entrada.

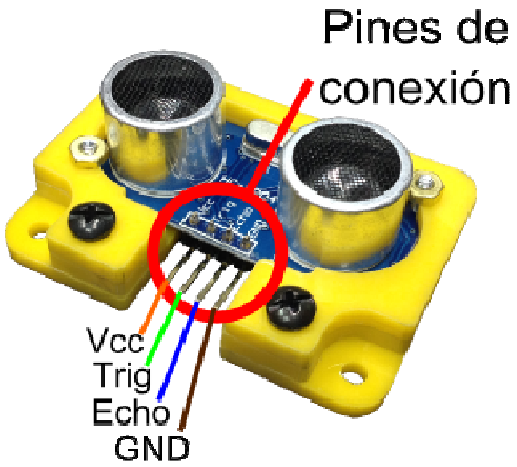


Por último, agregamos un pequeño tiempo de espera y con eso terminamos el código del programa de testeo. Es interesante notar que este tipo de programas nos servirá para testear otros tipos de sensores. Lo que tendremos que tener en cuenta de querer cambiar el sensor que utilizamos es que esté seleccionado en el bloque conrrespondiente y que esté correctamente cableado en la placa del robot.



Actividad 3.2: testeo de un sensor ultrasónico

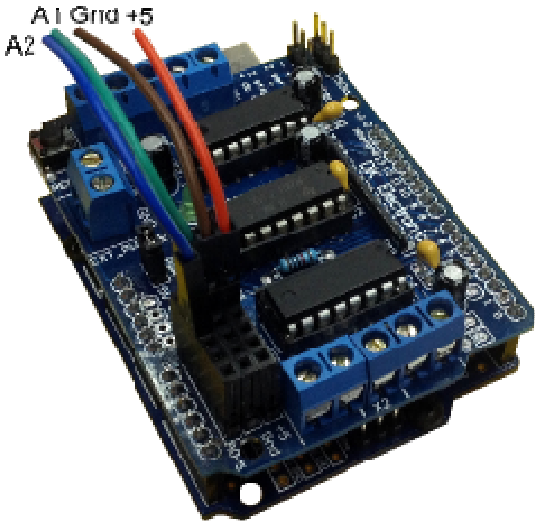
Para testear el sensor ultrasónico HC-SR04 que viene incluido con el robot Ta-Bot, usaremos un programa similar al anterior pero, por las características de este sensor en particular, tendremos que hacerle algunas modificaciones. Antes que nada, vamos a notar que el sensor HC-SR04 tiene 4 pines de conexión mientras que el sensor IR CNY-70 tiene sólo 3. Por lo tanto, tendremos que usar un cable de 4 hilos o 4 cables individuales.



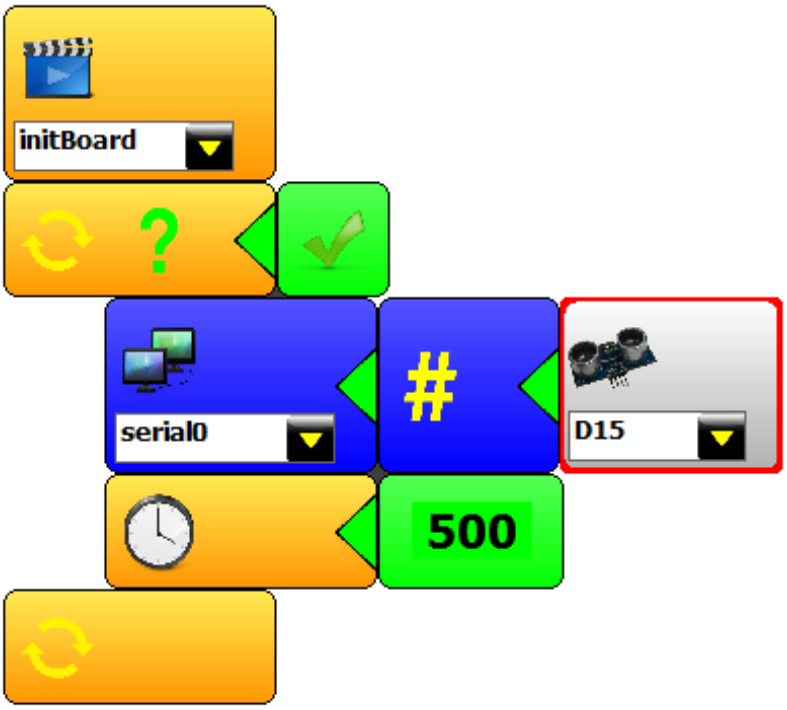
Para testear este sensor, vamos a conectar los cables de la siguiente manera:

Color del cable	Pin del sensor	Conector de la placa
NARANJA	Vcc	+5
VERDE	Trig	A1
AZUL	Echo	A2
MARRÓN	GND	Gnd

Por lo tanto, los cables conectados a la placa quedarán como se muestra en la siguiente figura.

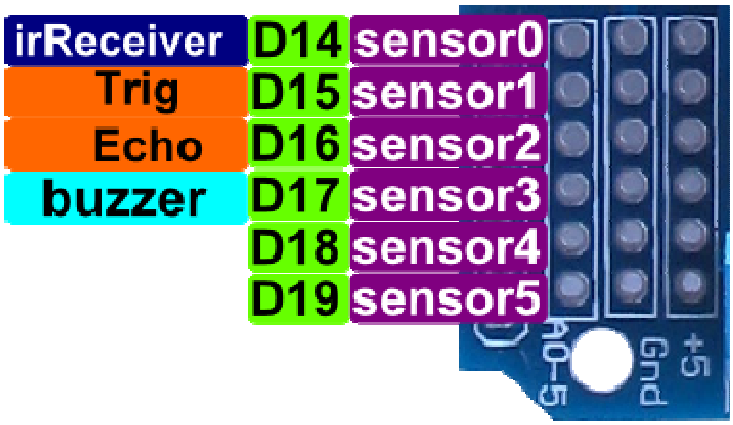


Una vez conectado el sensor en los pines correspondientes, armamos el siguiente código en miniBloq:

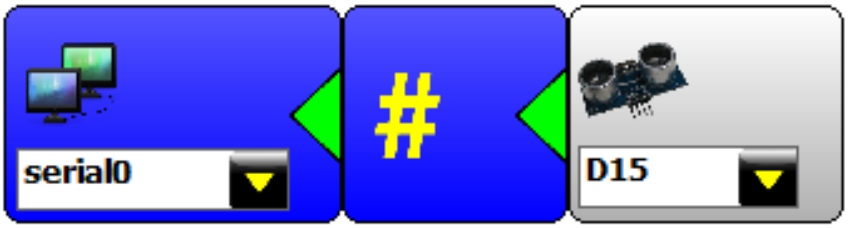


A diferencia del código anterior, en este caso, en vez de asignar un valor tomado de una entrada analógica se asignamos el valor detectado por un sensor específico: el HC-SR04.

Una vez seleccionado el bloque específico para sensor HC-SR04, seleccionamos el pin al que lo conectamos. Si nos fijamos en la imagen de miniBloq que muestra la numeración de los pines, notaremos que, junto a "sensor1" dice "D15" sobre fondo verde y, a su izquierda, "Trig" sobre fondo anaranjado.

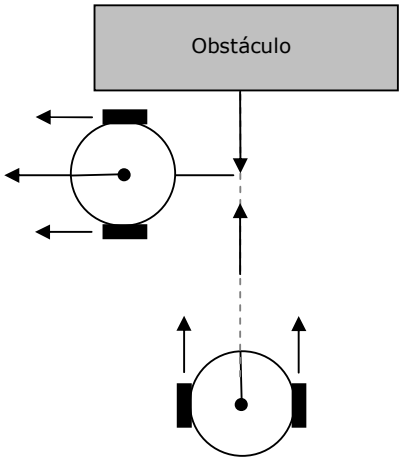


Por lo tanto, para el bloque del sensor HC-SR04 tendremos que seleccionar el pin D15 que corresponde al denominado "Trigger".

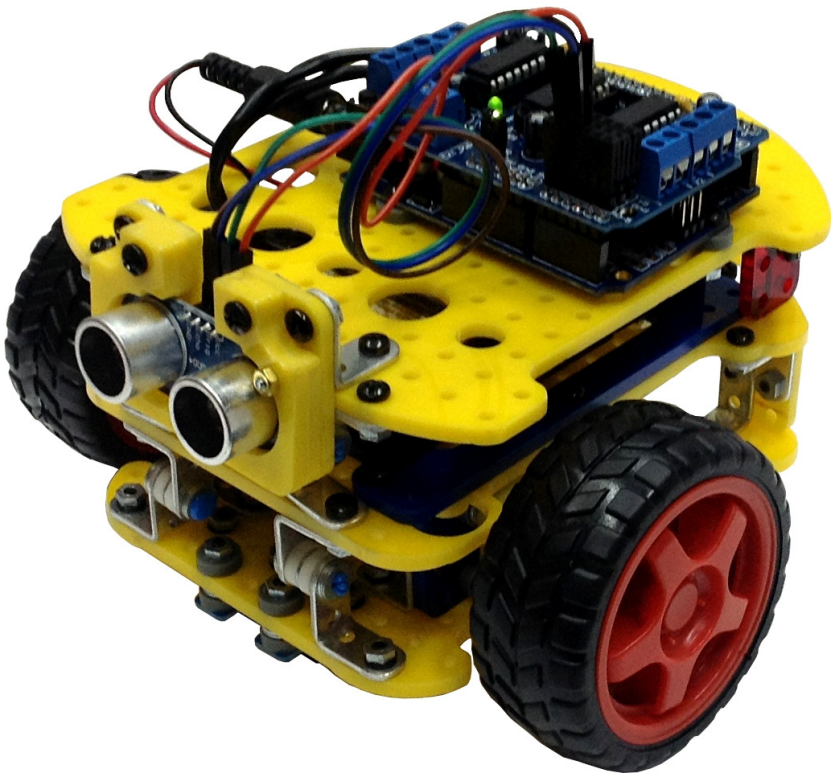


Actividad 3.3: avanzar hasta detectar un obstáculo

¿Cómo haría un robot para esquivar un obstáculo usando un único sensor montado en su parte frontal tal como lo muestra la siguiente figura? Para realizar esta actividad, utilizaremos el sensor HC-SR04 que testeamos en la actividad anterior.

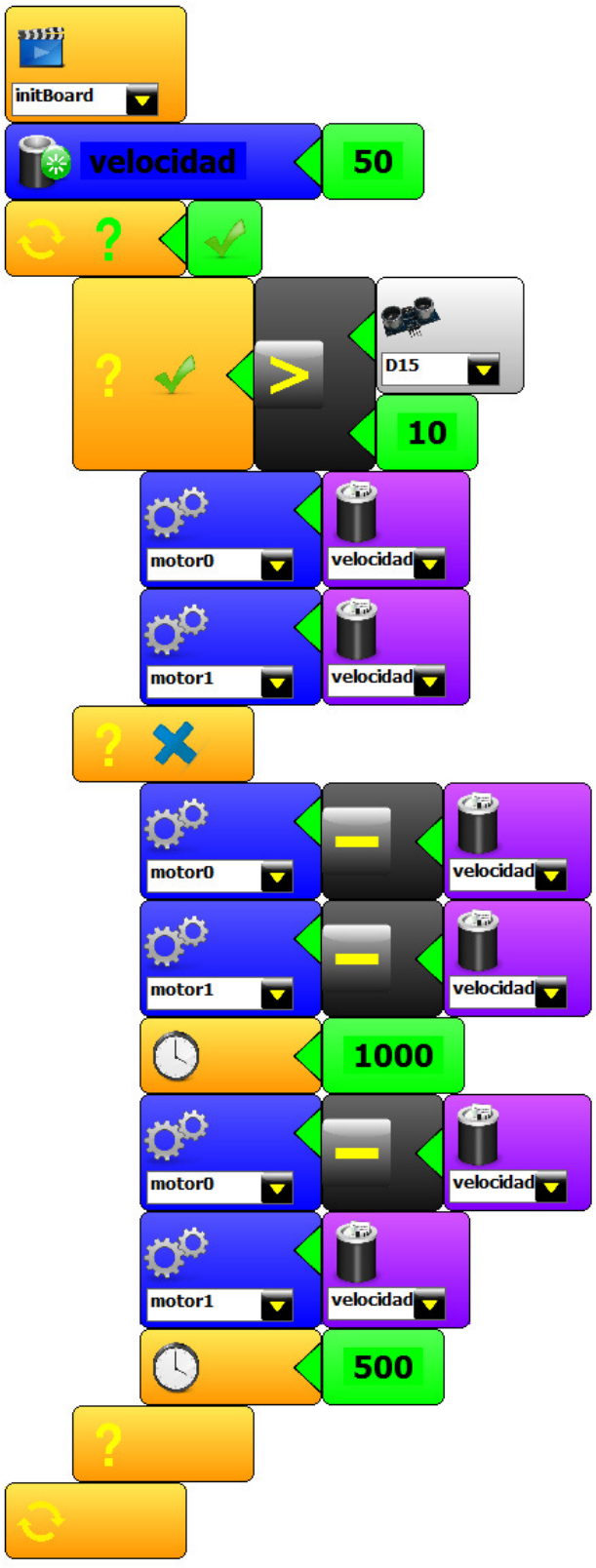


Como vemos en la imagen que está a continuación, el Ta-Bot ya viene con el sensor HC-SR04 en su parte frontal. Conectado como se indica en la actividad 3.2 podremos programarlo utilizando miniBloq.



Código en miniBloq de la actividad 3.3

El código necesario para realizar la actividad en miniBloq es el que se muestra a continuación.



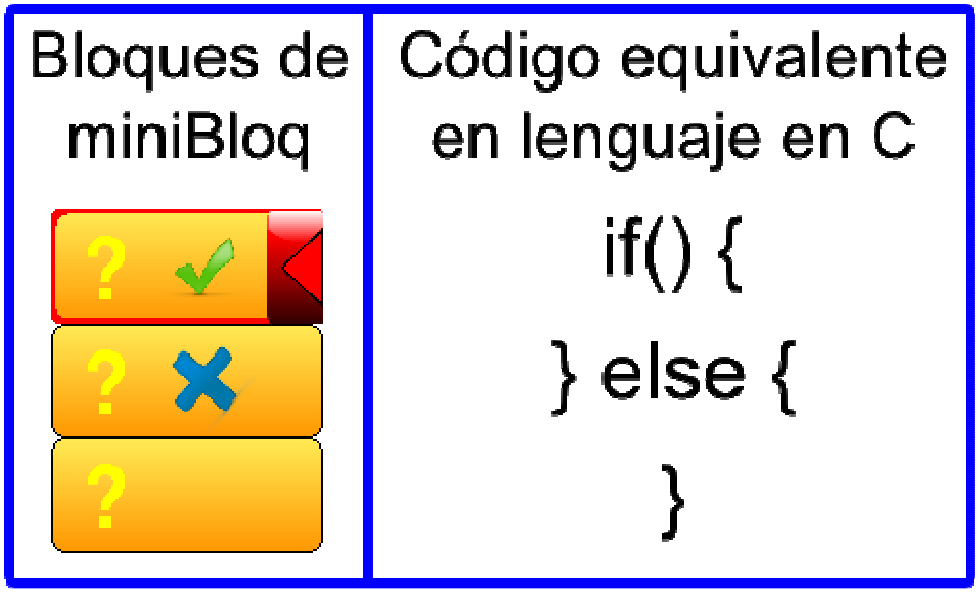
Explicación del código de la actividad 3.3

Para destacar en este código tenemos las siguientes situaciones:

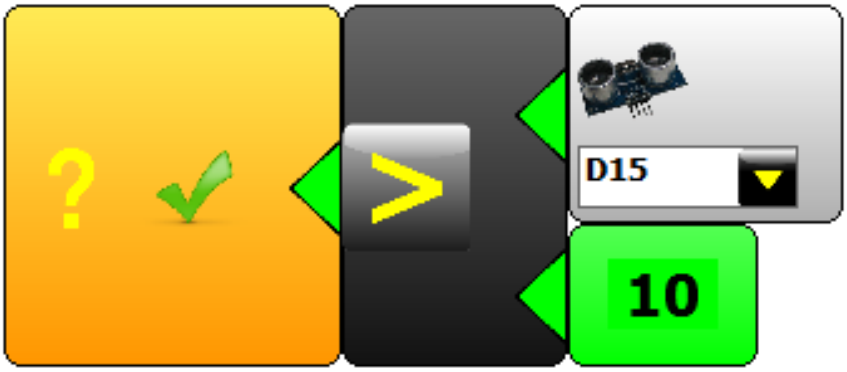
1- La velocidad de los motores la asignamos a través de una variable creada al comienzo del programa. De querer modificarla, solamente tendremos que cambiar el valor que agregamos al bloque que genera la variable.

2- Creamos un ciclo infinito dentro del cual estarán todas las órdenes que queremos que el robot ejecute. Las decisiones las tomará dentro de este bucle.

3- Dentro del ciclo infinito usaremos una estructura de control que le permitirá al robot decidir qué hacer dependiendo de lo que mida el sensor HC-SR04. Esta estructura es el equivalente a la sentencia "if" en lenguaje C. Al seleccionar el bloque correspondiente a esta estructura miniBloq generará 3 bloques: uno para establecer la condición que se evaluará, otro para el código que se ejecutará cuando la condición sea falsa y un tercer bloque que delimitará el código.

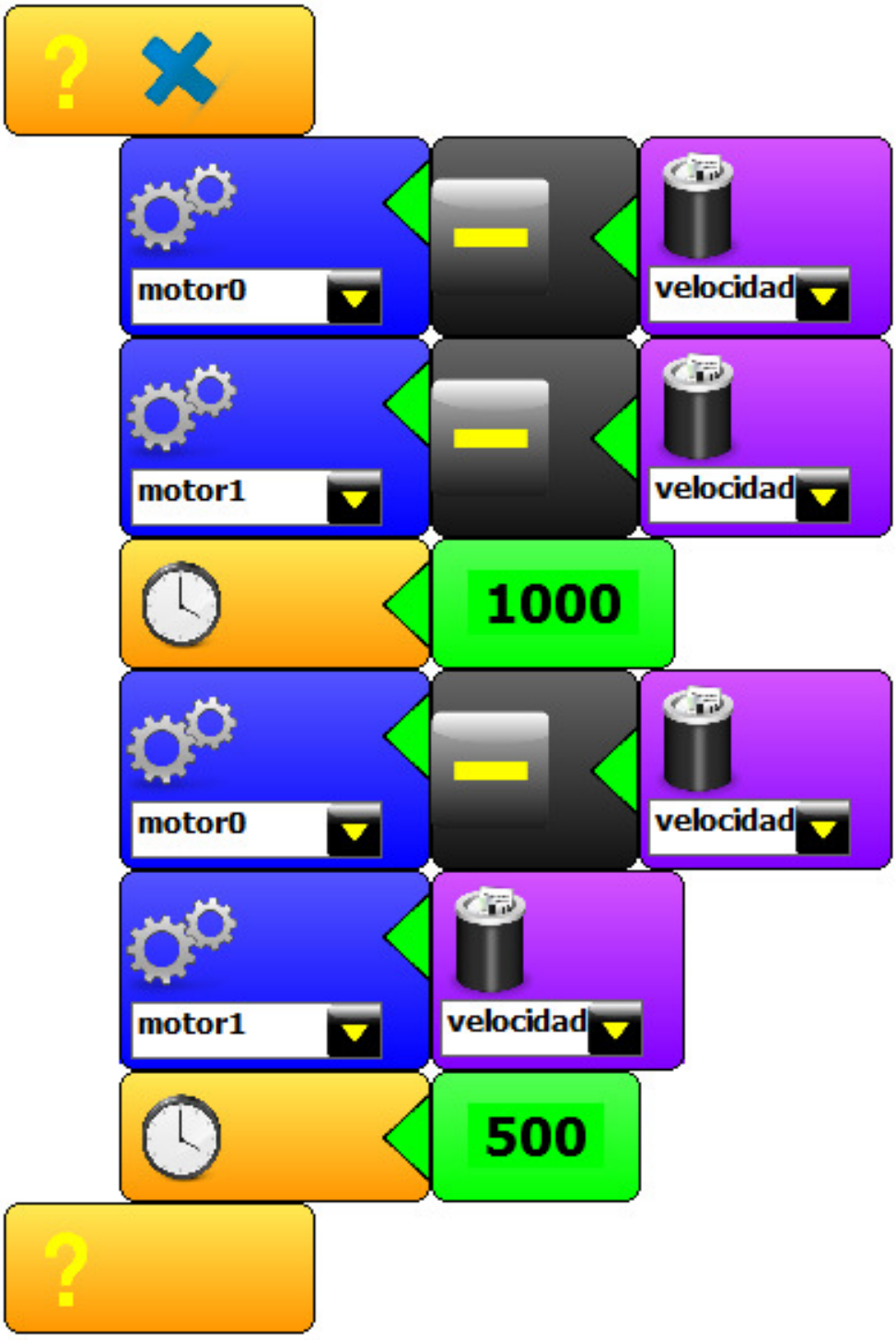


4- La condición, de ser verdadera, hará que se ejecute el código que se encuentra entre el bloque que posee la tilde verde y el que posee la cruz azul. En nuestro ejemplo, esta condición será una comparación entre el valor detectado por el sensor y un número que elegimos nosotros (10).

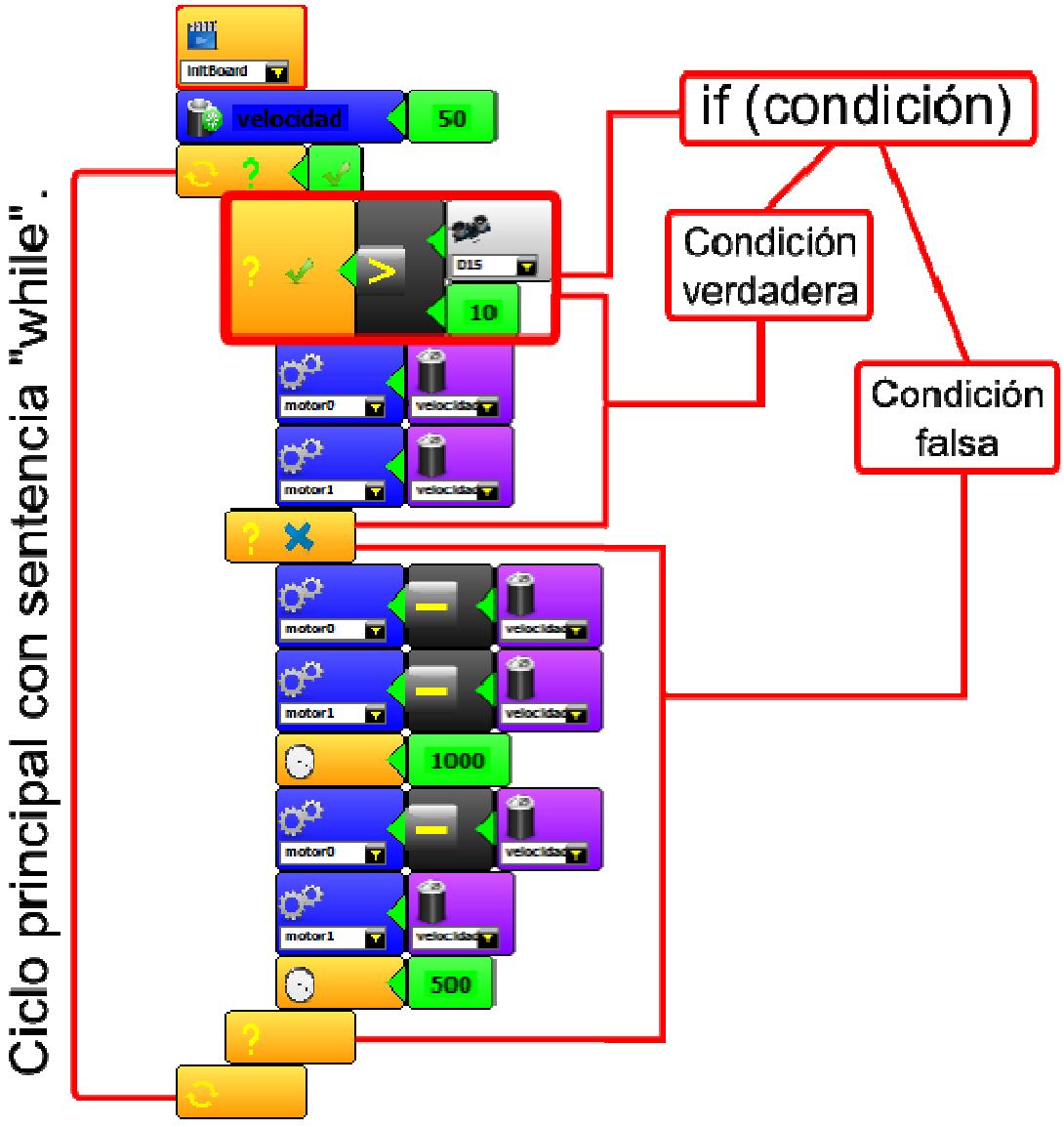


5- Tengamos en cuenta que el bloque del sensor toma un valor detectado y lo convierte en un número. Ese número es el asignado a esta comparación. Luego, si el valor detectado por el sensor en cuestión es mayor a 10, el robot ejecutará el código que se encuentra a continuación. En nuestro caso, si el sensor detectó un valor "mayor a 10" asumimos que está lejos de un obstáculo y, por lo tanto, deberá avanzar hacia adelante.

6- Luego, entre el bloque con la cruz azul y el bloque que cierra a la estructura de control que usamos para decidir, encerramos el código que usaremos cuando el robot detecte un obstáculo. En este caso, el sensor frontal habrá detectado un valor "menor o igual a 10".



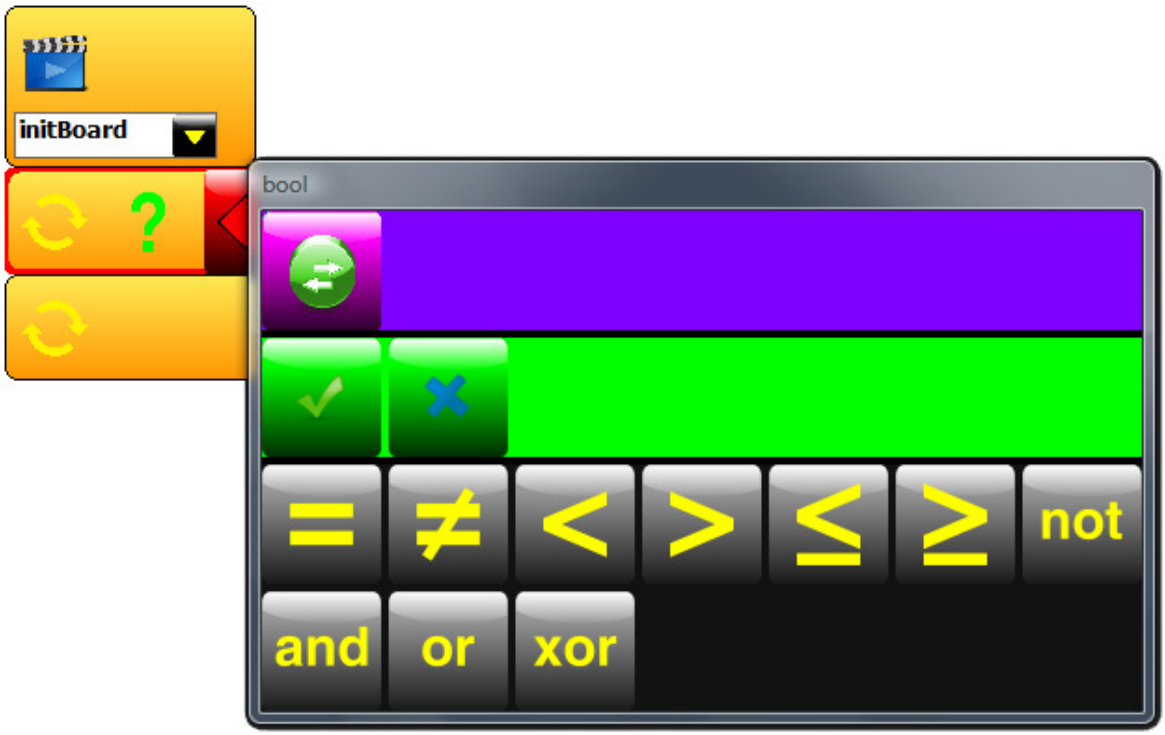
Por último, analicemos la lógica de este programa para entender mejor cómo funcionan las estructuras de control. Lo primero que hicimos fue crear un ciclo infinito con el bloque que genera la sentencia "while". Dentro de este bucle, utilizamos la sentencia "if" para que, cada vez que se ejecute el ciclo principal, el robot decida entre dos posibilidades. De esta manera, si los resultados de las detecciones del sensor son mayores a 10, el robot solamente ejecutará las órdenes encerradas entre los dos primeros bloques de la sentencia "if". Pero cuando el sensor detecte un obstáculo, el programa ignorará las órdenes que hacen avanzar al robot para ejecutar los bloques que se encuentran a continuación del bloque que representaría al "else" (bloque con cruz azul).



Operadores lógicos en miniBloq

Los operadores lógicos los utilizamos cuando queremos combinar varias condiciones dentro de una misma estructura de control. Por ejemplo, en el caso de querer que se ejecute un determinado bloque de código solo si dos condiciones son verdaderas al mismo tiempo, utilizaremos un operador lógico que signifique "y". En el caso de querer que se ejecute un bloque ante la ocurrencia de "al menos una" de dos condiciones, utilizaremos un operador lógico que signifique "o". También podremos pedirle al procesador que ejecute ciertas órdenes cuando "no" sea cierta una condición.

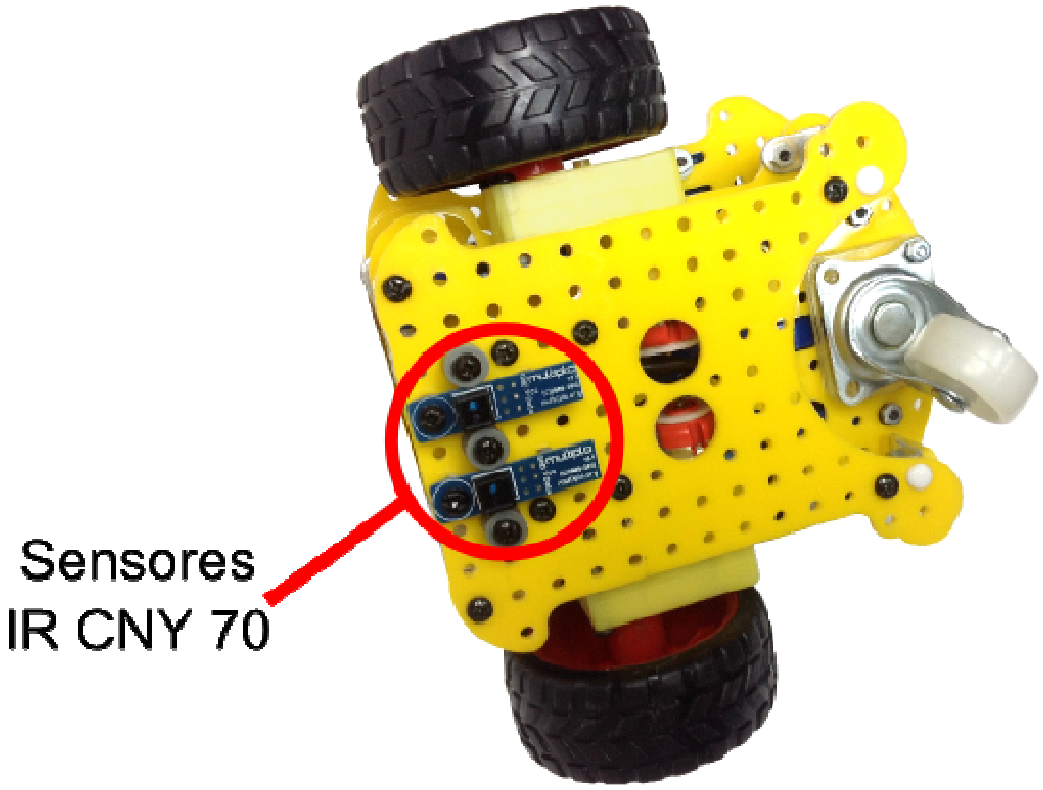
En el entorno miniBloq disponemos de operadores lógicos. Estos se usan dentro de las condiciones de las estructuras de control al igual que los operadores matemáticos y los booleanos. En la imagen que se muestra a continuación, podemos ver la lista de operadores que se despliega junto a un ciclo como el while. Los operadores lógicos están al final y son "not", "and", "or" y "xor".



Actividad 3.4: hacer que el robot siga una línea

En esta actividad vamos a programar al robot para que, usando dos sensores IR CNY 70, pueda seguir una línea negra pintada sobre un fondo blanco.

1- Sensores IR CNY 70: Lo primero que hay que hacer es adaptar al robot para que un par de sensores IR apunten hacia el suelo. Hay que tener en cuenta que los IR seleccionados tendrán que estar muy cercanos al piso ya que, si se los pusiera muy alejados, podrían no darse los resultados esperados. El robot Ta-Bot ya viene con los sensores montados en la posición indicada.



2- Cableado: A uno de los sensores IR lo cablearemos como en la actividad 3.1: "Testeo de un sensor IR". Al otro, lo cablearemos en los pines que están al lado de estos, tal como se muestra en la siguiente tabla.

Color del cable	Pin del sensor	Conector de la placa
ROJO	Vcc	+5
NEGRO	GND	Gnd
BLANCO	Data	A5
VIOLETA	Vcc	+5
GRIS	GND	Gnd
AMARILLO	Data	A4

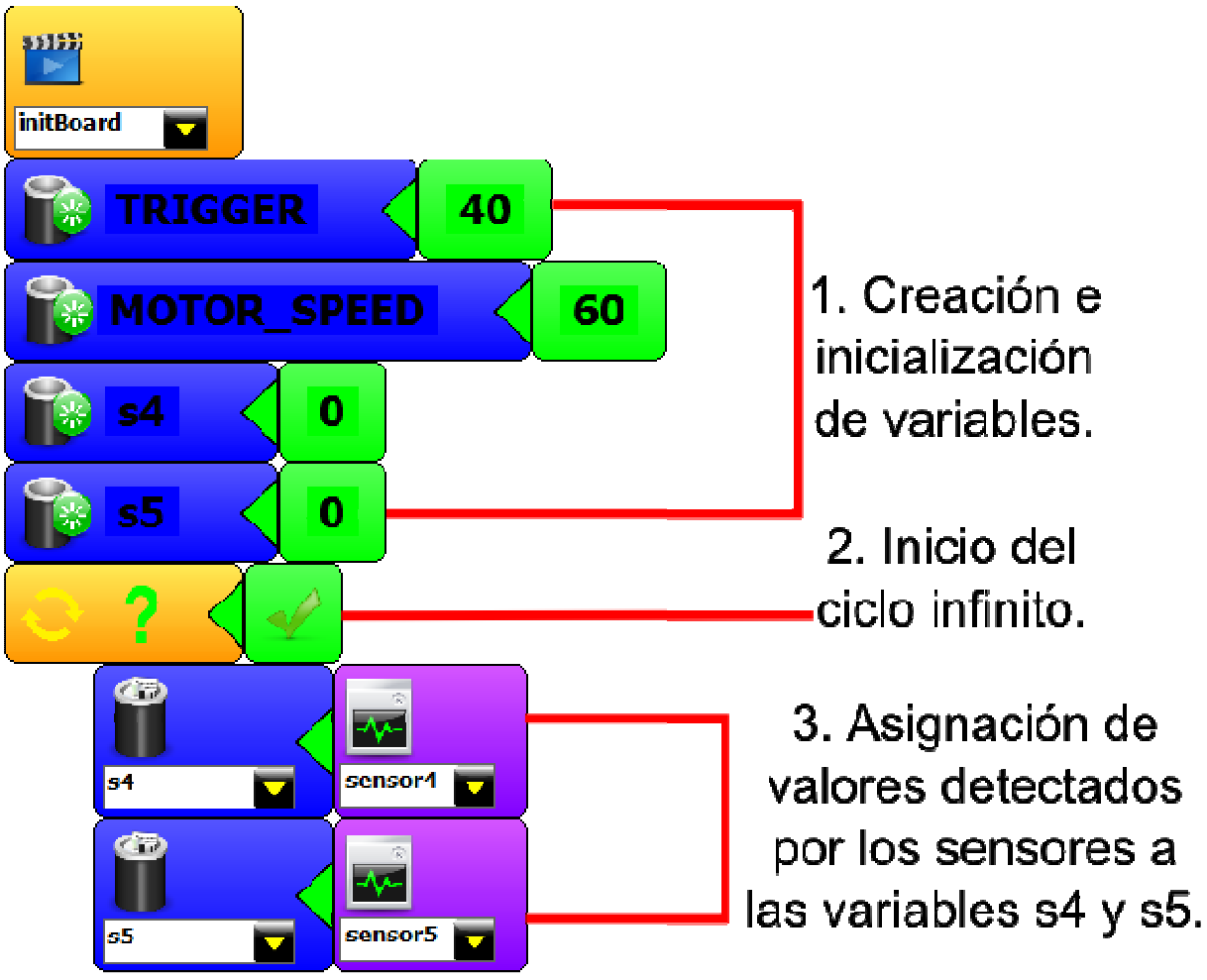
3- Programa: una vez que el robot esté preparado, descargamos el siguiente programa a la placa del robot. El mismo se muestra en partes por su longitud.

En la primera parte del programa haremos tres cosas:

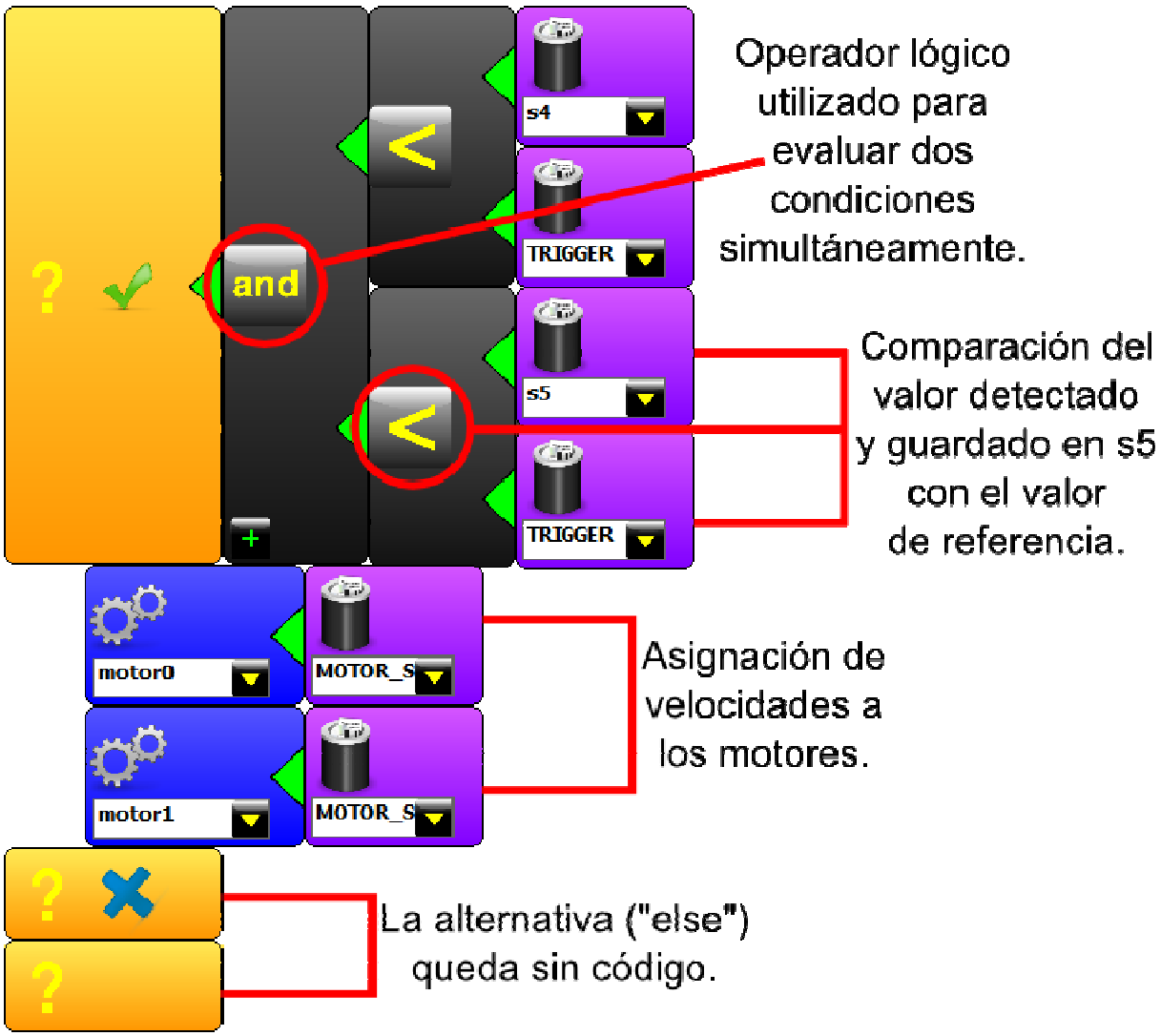
(1) Creación de variables para velocidad y manejo de sensores. Las variables s4 y s5 las inicializamos con el valor "0" ya que serán modificadas luego en cada ejecución del ciclo principal.

(2) Inicialización del ciclo infinito usando "while".

(3) Dentro del bucle, asignamos a las variables s4 y s5 el valor detectado por los sensores 4 y 5 respectivamente.



A continuación de la primera parte del programa, tendremos que agregar tres estructuras de control "if". Cada una de estas evaluará una situación distinta basada en la combinación de detecciones de los dos sensores. A continuación, mostramos la primera de las tres en la cual, si los dos sensores detectan un valor menor al guardado en la variable "TRIGGER", se asigna a los dos motores la misma velocidad para que el robot avance en línea recta. Este caso corresponde al momento en el que el robot se encuentra sobre la línea negra.



Los dos casos siguientes son similares. En uno de ellos se evalúa la situación en la cual un sensor está detectando un valor alto (blanco) y el otro un valor bajo (negro). Dicha situación nos indica que el robot necesita frenar el motor que está del mismo lado que el sensor que detectó "negro".