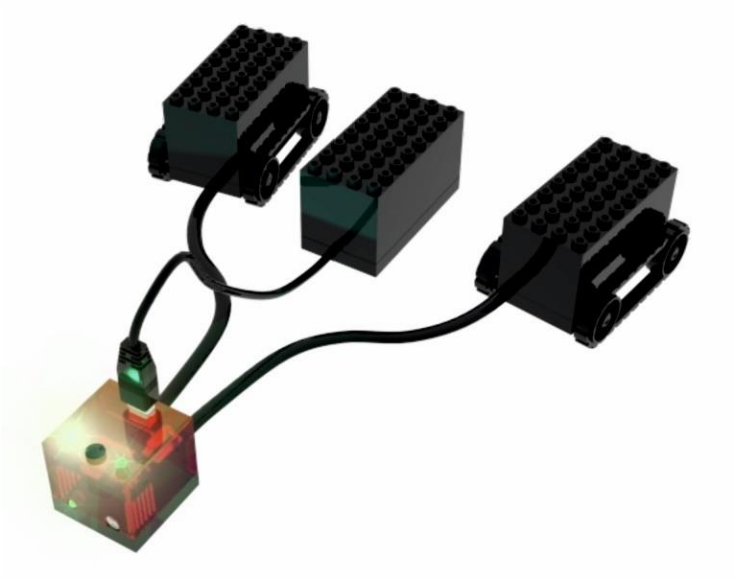
PROBOTlab® 4.0

Como se compone el sistema PROBOTS

Está compuesto por cuatro partes fundamentales

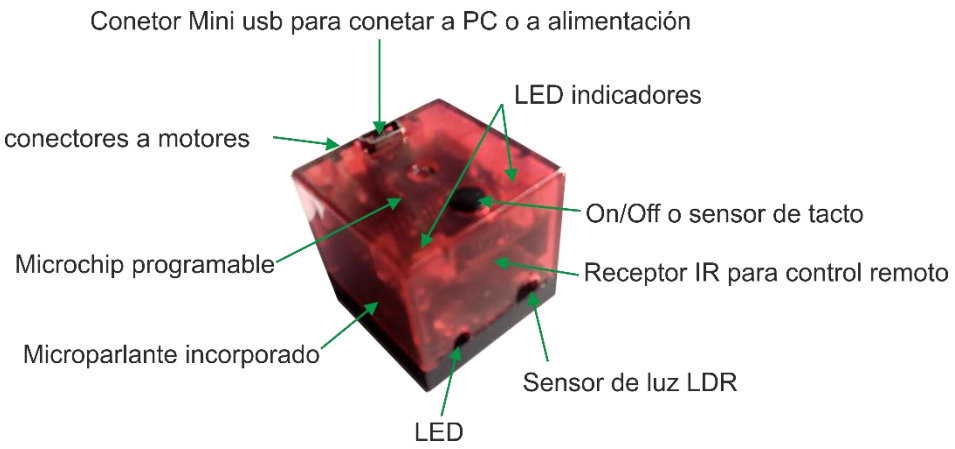
1. El e-brick (Ladrillo inteligente)
2. Dos motores de movimiento
3. El portapilas o pila recargable según kit

3. Los ladrillos constructivos



El e-brick (Ladrillo Inteligente)

El e-brick es el cerebro del sistema Probots. Este Microchip se compone de varias partes como ser: El sensor de Luz, el sensor de Tacto, el sensor Infrarojo,



los Leds, el Parlante. A continuación vemos un gráfico detallando las partes antes mencionadas.

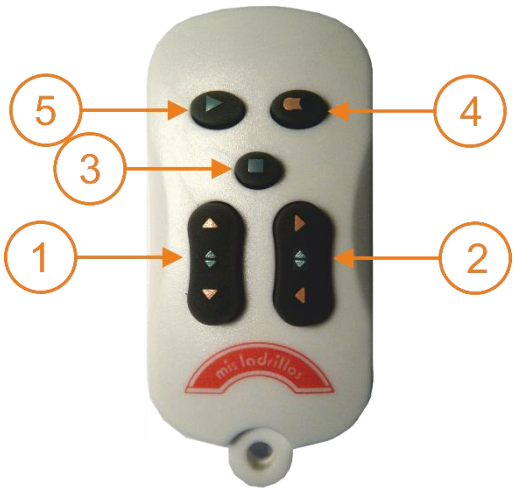
El e-brick puede recibir órdenes de dos maneras diferentes: una de ellas es a través del cable de conexión, desde una PC y la otra es a través del control remoto.

En el eBrick se almacenan los programas realizados con el ProbotLab y que fueron enviados al mismo a través del cable de conexión. También recibe las órdenes del control remoto tanto para mover al Probot como para programarlo.

Como usar el eBrick sin la PC

Podrás comandar los movimientos de tu Probot con control remoto, como así también se podrán ejecutar programas cargados previamente al e-brick. Conectalo a la batería y presiona cualquier tecla del control remoto. Cuando los leds dejen de parpadear, significa que has entrado al modo comando remoto.

Manejando el e-brick con remoto



El control remoto de Mis Ladrillos tiene 2 modos de funcionamieto.

Primer modo de funcionamiento:

1. Este botón hace que los motores avancen o retrocedan al mismo tiempo. Funciona de manera paulatina, es decir: Apretamos una vez, avanza lento, otra vez, acelera un poco más, otra, más rápido, se aprieta la tecla en sentido contrario y baja la velocidad. Cuando llega a velocidad 0 cambia el sentido. Si se quiera cambiar el sentido de una forma rápida apriete 3 (parar) y luego la tecla en el sentido deseado.
2. Este botón hace que los motores avancen o retrocedan en sentido contrario cada uno, de esta manera se genera el giro. El funcionamiento de la velocidad es igual a las teclas 1.
3. Parada
4. Se cambia al segundo modo de funcionamiento explicado más abajo
5. Ejecuta el programa que le haya bajado desde la PC y tenga grabado en su memoria.

Segundo modo de funcionamiento:

1. Con este botón se logra que el motor izquierdo avance o retroceda.
   El funcionamiento de la velocidad es igual a las teclas 1 del modo anterior.
2. Con este botón se logra que el motor derecho avance o retroceda.
   El funcionamiento de la velocidad es igual a las teclas 1 del modo anterior.
3. Para ambos motores. Si se desea parar un sólo motor, mientras el otro siga funcionando, se puede hacerlo rebajando la velocidad a cero, del motor que se desea parar.
4. Se vuelve al primer modo de funcionamiento
5. Ejecuta el programa que se haya grabado desde la PC. Una vez alimentado el ladrillo con la batería, los leds superiores comenzarán a destellar. La primer pulsación de este botón hará que los leds se apaguen. Luego de la segunda, estos destellarán 3 veces y comenzará a ejecutarse el programa grabado en la memoria.

Instalando el Probot Lab

El Probot Lab es el programa que le permitirá crear recorridos y asignarle tareas al Probot. Para instalar el programa de tener en cuenta los siguientes aspectos:

Requerimientos mínimos del Sistema

· Windows XP o superior
· 512 MB libres en disco rígido
· Lectora de CD
· Monitor color de 1024x768
· Puerto USB

Instalación de Probot Lab

Descargue el ProbotLab de nuestra web de misladrillos:

http://misladrillos.com/magento/index.php//soporte_p

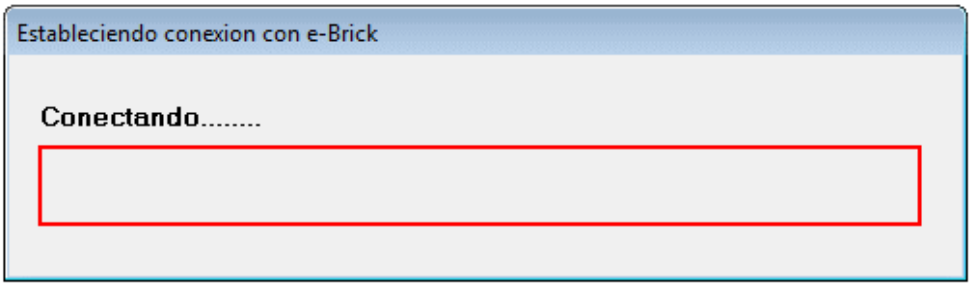
Configuración

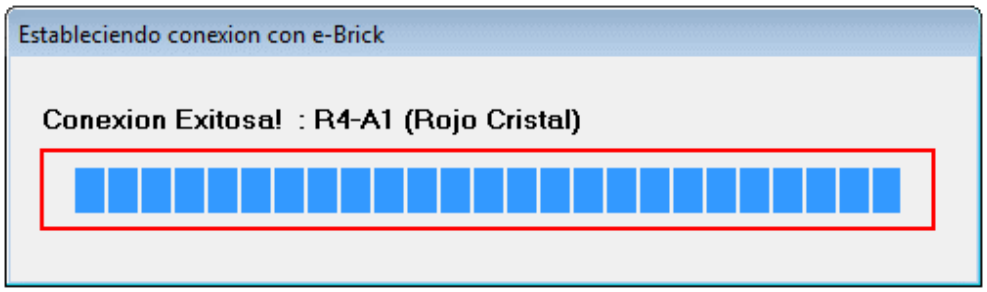
Para una correcta visualización de la interfaz del Probot Lab la pantalla debe estar configurada en 1024x768 con tamaño de fuente pequeñas, ambos cambios se realizan en el panel de control \Pantalla.

Como conectar el Ladrillo eBrick a la PC

El eBrick se comunica con la PC por el puerto USB. El cable de comunicación provisto tiene en un extremo un conector más pequeño que se conecta al eBrick y otro más grande que se conecta en el puerto USB PC.

Una vez que el eBrick está conectado a la PC hay encenderlo y esperar a que termine la secuencia de arranque (los 3 tonos que suenan al encenderlo). Luego, presionar la tecla **"Conectar"** que se encuentra en la barra de herramientas del *Probot Lab* y esperar a que la conexión se concrete. En la barra de información indicara **"Conectando…"** y finalmente avisara cuando el eBrick esté conectado.





Como cargar programas de la PC al eBrick

Una vez lograda la conexión ya estás listo para enviar programas de la PC al eBrick. **Probot Lab** tiene programas diseñados especialmente para cada uno de los modelos, los cuales se encuentran del lado derecho de la pantalla (Área de Programación Rápida). Click sobre la foto de nuestro modelo se abre una lista de programas. Solo tienes que seleccionar uno de ellos y en pantalla se abrirá el programa seleccionado. De esta manera podrás ver cómo está escrito el programa y modificarlo si deseas que funcione distinto. Una vez abierto el programa solo debes presionar la tecla CompiGraba de la barra de Herramientas y el programa será enviado al eBrick y comenzara a ejecutarse automáticamente por lo que se perderá la conexión del eBrick con la PC.
El modo de abrir programas a través del Área de Programación Rápida está limitada a cargar solo los programas de la lista. Para abrir un programa que se encuentra en una unidad de disco (CDROM o disco rígido) debes seleccionar “Abrir” en la barra de herramientas o bien en la opción “Archivo” de los menú desplegable y luego “Abrir”.

NOTAS

- **Cada vez que se ejecuta en programa en el ladrillo la conexión se pierde, por lo tanto para volver a bajar otro programa debes repetir el proceso de conexión antes explicado.**

- **Si al encender el ladrillo se le envía una orden por control remoto ya no permite conectarse a PC, para hacerlo deberás apagar y volver a encender el ladrillo y luego intentar conectarlo.**

Como programar

Primeros pasos en programación

¿Que es y para que sirve un programa?

Un programa consiste en una serie de instrucciones que se ejecutan ordenadamente. Un programa se podría comparar con una receta de cocina, donde se encuentran una lista de acciones a realizar, como por ejemplo: batir 6 huevos, derretir chocolate, hornear durante 30 minutos, etc.
En el caso del e-brick las acciones actúan sobre los motores, el parlante, las luces, y los sensores.

Conceptos Básicos

A continuación te explicamos algunos conceptos que vamos a utilizar a lo largo del manual y que van a ser útiles para que puedas comprender el funcionamiento del **ProbotLab**.
En la pantalla del **ProbotLab** los programas se leen de izquierda a derecha y de arriba hacia abajo.

Salidas
Vamos a llamar Salidas a los motores, las luces, el parlante y la salida de PC. Estas cuatro cosas que nombramos recién tienen en común que toman un dato del programa y generan una acción directamente relacionada con ese dato. Por ejemplo, el dato ***avanzar*** lo traduce en encender los motores en sentido de avance.

Entradas
Vamos a llamar entradas a los sensores (de luz, de tacto, de control remoto), la conexión a PC y el micrófono. Estas cosas tienen en común que las condiciones del lugar donde se encuentra el eBrick las afectan y en base a esa puede realizar una acción u otra. Por ejemplo El sensor de control remoto detecta una señal y puede hacer detener al Probot a hacer que realice otra acción. La información leída por los sensores es guardada en los contenedores.

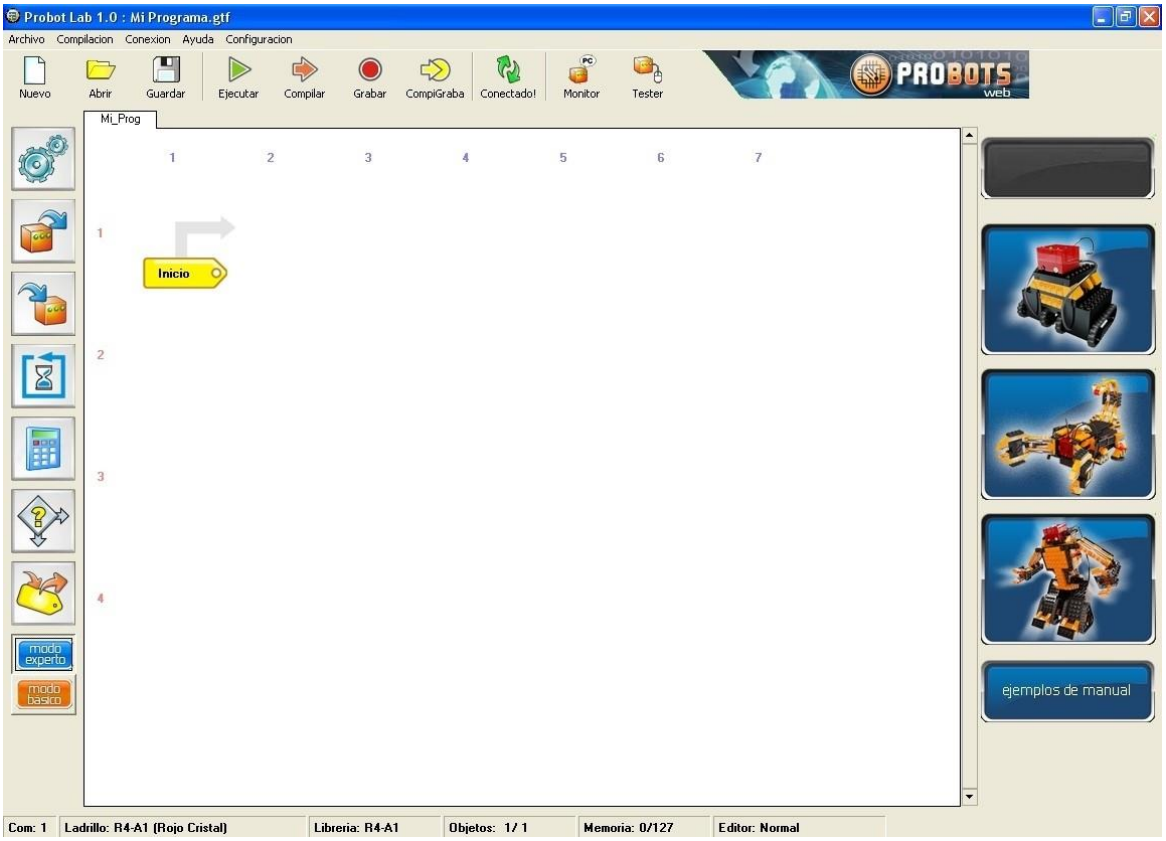
Contenedor
Los contenedores permiten guardar un valor numérico para usarlo cuando sea necesario. El Probot Lab maneja 5 contenedores que por simplicidad se lo identifica por colores. Los contenedores pueden cargarse con el valor recibido de un sensor, con un número fijo (constante), con un número al azar o con el resultado de una operación matemática. Usando contenedores también se le puede decir a una **Salida** como comportarse o a una **Espera por Tiempo** cuanto demorarse dependiendo del número que tenga guardado.

Constantes
Las constantes son números fijos que a diferencia de los contendores, las constantes una vez escritas quedan inalterables dentro del programa.

La herramienta de programación PROBOT Lab

A continuación veremos el area de programación (Interfaz) del Probot Lab **El**

aspecto de la interfaz de Probot Lab es el siguiente:



La interfaz se divide en 6 áreas:

- Los Menús Desplegables · La Barra de Herramientas
- La Barra de Programación
- Las teclas de Programación Rápida
- El Área de Programación (zona donde se “dibuja” el programa para el eBrick)
- La Barra de Información

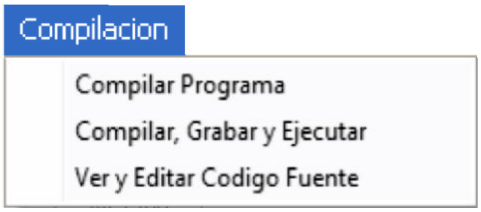
Menues Desplegables

El área de menues desplegables contiene 5 opciones y se encuentra en la parte superior de la Interfaz.

Menú Archivo

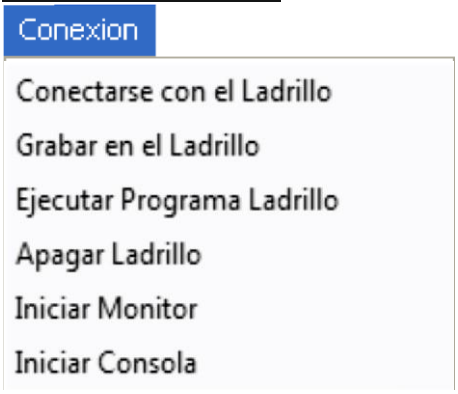
Nuevo: Limpia el área de programación para comenzar un nuevo programa.
Abrir: Abre y carga un programa grafico.
Guardar: Guarda el programa que se encuentra en el área de programación con el nombre actual del archivo.
Guardar como...: Guarda el programa que se encuentra en el área de programación con un nuevo nombre.
Tester: Prueba la funcionalidad del ladrillo.

Menu Compilación



Compilar Programa: Traduce el programa grafico en el lenguaje que entiende el eBrick (código fuente).
Compilar, Grabar y Ejecutar (CompiGraba): Traduce el programa grafico en el lenguaje de eBrick, lo graba en la memoria del Ladrillo y lo ejecuta. **Ver y Editar Codigo Fuente:** Permite ver y editar programas en el lenguaje de bajo nivel que entiende el eBrick. Con esta herramienta se pueden crear programas más potentes pero requiere mayores conocimientos de programación.

Menú Conexión



Conectarse con el Ladrillo: Después de conectar el cable de comunicación al ladrillo y a la PC y encenderlo debe usar esta opción para iniciar la conexión.
Grabar en el Ladrillo: Graba en la memoria del ladrillo el programa compilado, es decir que antes de usar esta función hay que compilar el programa grafico.
Ejecutar Programa Ladrillo: Una vez que se ha grabado un programa en la memoria del ladrillo esta función lo hace correr (lo ejecuta). Esta opción

funciona solo cuando está conectado a través del cable de conexión.

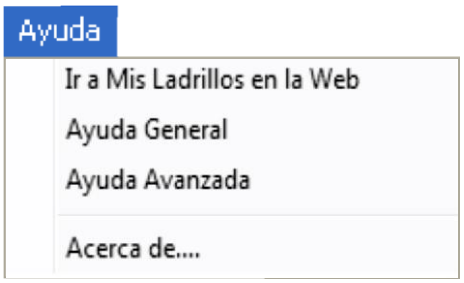
Resetear Ladrillo: Reinicia el ladrillo, borrando el programa que se encontraba guardado en el mismo. El reinicio del ladrillo es como si se apagara y se volviera a prender. Esta opción no funciona si el ladrillo está corriendo un programa grafico.

Apagar Ladrillo: Apaga el ladrillo, funciona igual que la tecla de encendidoapagado del ladrillo. Esta opción no funciona si el ladrillo está corriendo un programa grafico.

Iniciar Monitor: Abre la herramienta de comunicación con el ladrillo que permite intercambiar datos con los programas creados que usan mensajes "A VER PC".

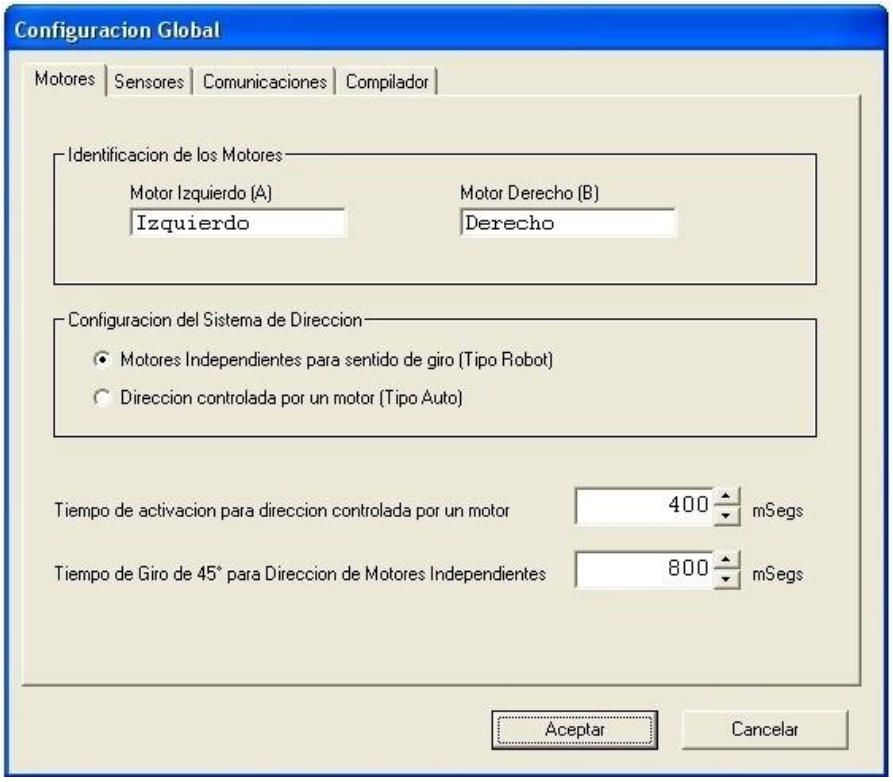
Iniciar Consola: Abre la herramienta de comunicación que permite manejar el ladrillo desde la PC.

Menú Ayuda



Configuración

Presionando aquí se abre la pantalla de configuración general del Sistema donde encontramos diferentes solapas.

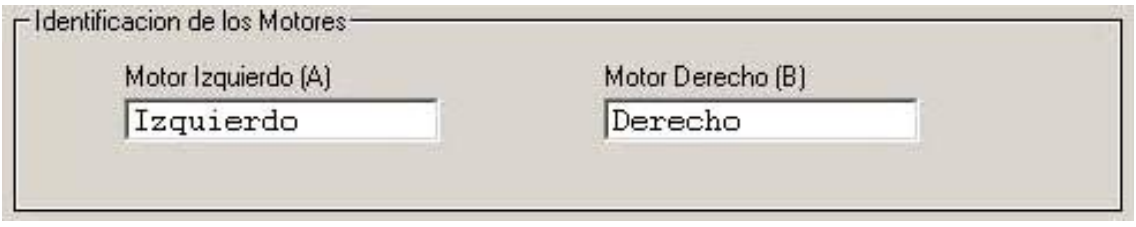


Solapas de la pantalla de configuración.

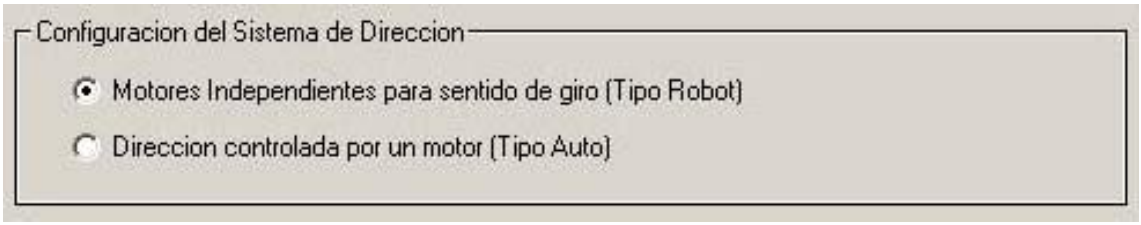
Motores

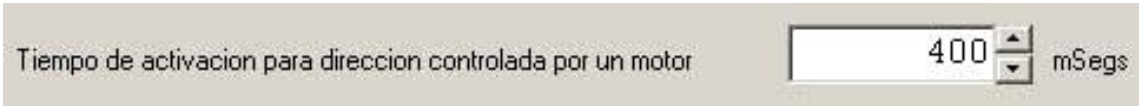
Identificación de Motores [1]

En esta parte nos da la posibilidad de ponerle nombre a los motores. Asi cuando estemos programando aparecerán con los nombres que le hemos puesto.
Los nombres pueden tener hasta un máximo de 10 caracteres y para nombrarlos hay que cliquear sobre el cuadro de texto. De esta manera aparecerá el cursor con el que se nombrara al motor.



Configuración del Sistema de Dirección [1]
Esta opción depende del modelo de Probot que tenga, de esta manera el programa sabrá como compilar las acciones de los motores dependiendo del tipo de dirección que esté usando el modelo. Por Ej. En el caso del Robot, este tiene un motor de cada lado donde para doblar se acciona un motor y se detiene el otro.
En el caso del Auto, el sistema de dirección que posee es de un motor que gira el tren delantero activándose durante un breve periodo de tiempo.





Tiempo de activación de dirección controlada por un motor [1]
Es el tiempo que se encenderá el motor que gira la dirección en los modelos tipo **automóvil**. El tiempo de giro seleccionado hay que pensarlo según el ángulo de giro, el estado de las baterías y la superficie donde se encuentra el Probot, de lo contrario se pueden notar fallas en la ejecución de los recorridos.

Tiempo de Giro de 45° para dirección de motores Independientes [1] Es el tiempo que los modelos con dirección de motores independientes, tipo robot, necesitan encender sus motores para girar 45°. Como en el otro modelo, este tiempo puede variar dependiendo del peso del modelo, del tipo de suelo donde se esté usando y el nivel de carga de las pilas. Cuando este tiempo no está bien ajustado se notaran fallas en la ejecución de recorridos.

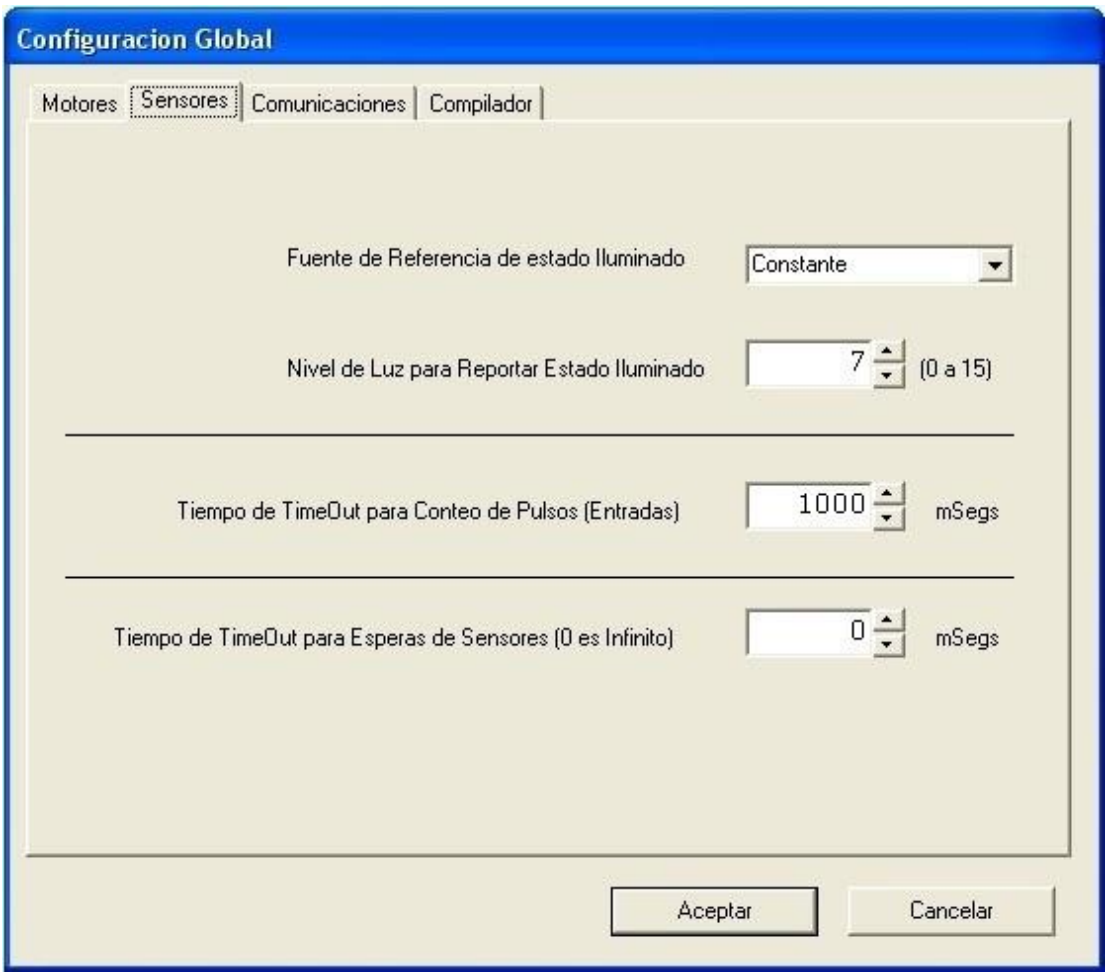
NOTA

Los tiempos de giro, tanto para el modelo Automóvil como para el modelo Robot, solo deben ser seteados cuando dentro de un programa utilizamos

la opción Recorrido, la cual permite crear un rec orrido especial dentro de otro general. En el programa General utilizamos para girar las opciones que se encuentran en la sección motores de la Barra de programación.

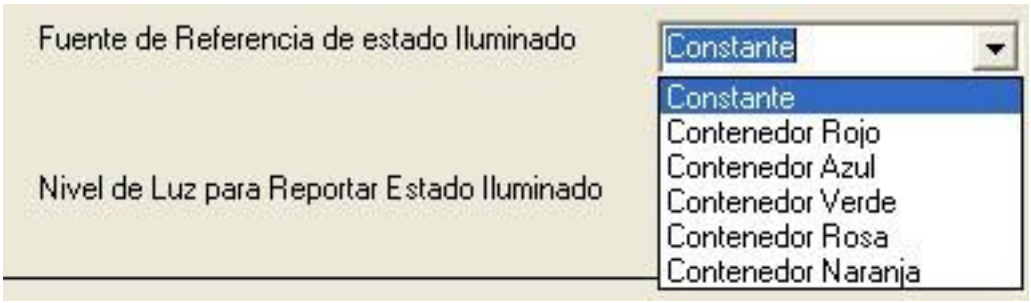
Sensores

Al presionar sobre esta solapa aparece la siguiente pantalla:



Fuente de Referencia para estado iluminado [1]

El sensor de Luz del eBrick tiene la capacidad de medir el nivel de luz que recibe devolviendo un valor de 0 (oscuro) a 15 (iluminado). Este parámetro permite seleccionar el valor de referencia que usara el programa en los bloques de *Esperar luz*, y comparar el nivel de luz. Al seleccionar la opción *"constante"* el valor esta dado por el valor ingresado en la opción "Nivel de Luz para Reportar…". las opciones *"contenedor xxxx"* usan como referencia un contenedor determinado que debe ser cargado por el programa.



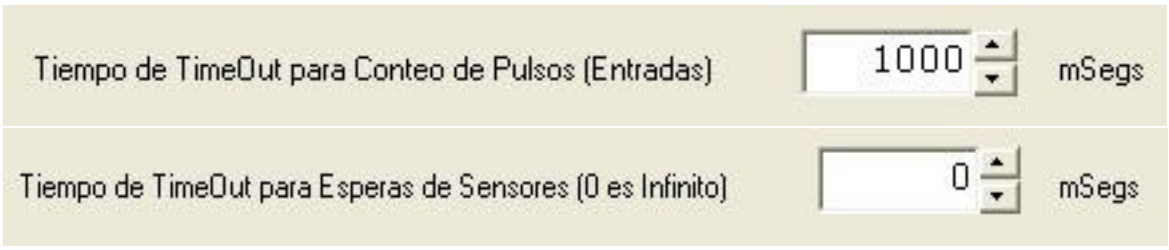
Nivel de Luz para Reportar Estado Iluminado [1]

En esta opción vamos a establecer los valores que indicaran a los bloques de programación el punto de quiebre entre oscuro e iluminado, para tomar decisiones. Por ejemplo, si el valor se fija en 7 y en una parte del programa existe un bloque de *"Esperar hasta que el sensor de Luz este Iluminado"* el programa se detendrá hasta que el nivel de luz supere el valor 7.

Esta opción sirve solo si seleccionamos como *Fuente de Referencia…* la opción ***Constante.***

Tiempo de TimeOut para Conteo de Pulsos (Entradas) [1]

El tiempo de TimeOut es el tiempo que el programa esperara a que un sensor responda. Superado este límite de tiempo el programa da por terminada la acción y continúa con el siguiente bloque evitando asi que el programa quede detenido indefinidamente. Por ejemplo, si el valor de TimeOut es de 2000 mseg (2 segundos) y en un programa se ejecuta un bloque de "Contar Pulsos de Sensor de Tacto", el bloque se quedara esperando cambios en el sensor durante 2 segundos, si transcurrido ese tiempo el sensor no cambia de estado (no incrementa el conteo) este bloque del programa se da por terminado y se sigue con el siguiente.



Comunicaciones

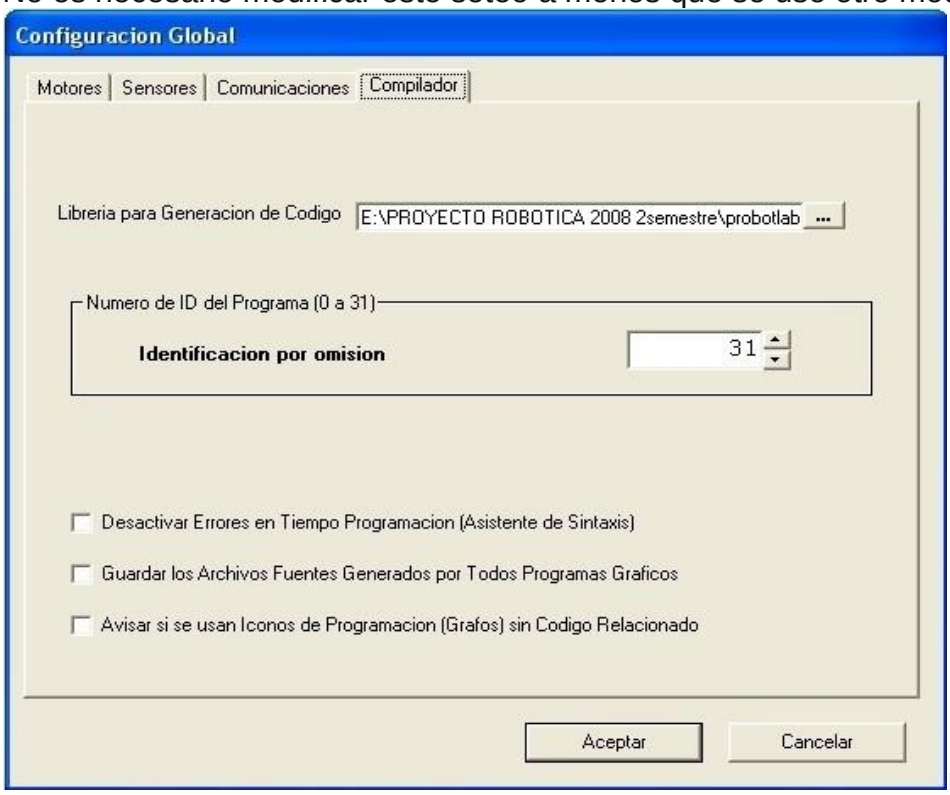
Conectar automáticamente antes de la acción CompiGrabar

Activando este control el Robot se conectara automáticamente al Probot Lab antes de hacer uso de la acción *"CompiGrabar"*

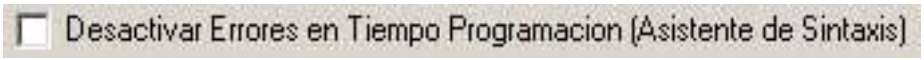
Compilador

Al seleccionar esta solapa se abre la siguiente pantalla:

No es necesario modificar este seteo a menos que se use otro modelo de ladrillo que

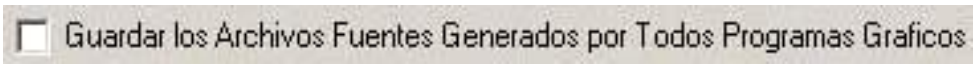


requiera una librería diferente. Es donde se guardan los códigos que se usan en el programa grafico para ser traducidos los



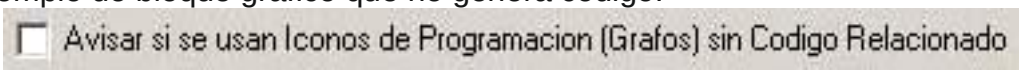
códigos de programación que recibe el eBrick, Si no se define una ubicación de librería correcta los programas generaran errores impredecibles.

Guardar los Archivos Fuentes Generados por Todos programas Gráficos [2]
Todos los programas gráficos generan un archivo fuente escrito en el lenguaje que entiende el eBrick, seleccionando esta opción todos los programas gráficos que se crean en el disco un archivo con el mismo nombre que el programa grafico pero con extensión .scr conteniendo el código fuente. Apagando esta opción estos archivos no se generan.



Avisar si se usan Iconos de Programación (Grafos) sin Código Relacionado [2]
Cada Bloque de programación gráfica al ser compilado se traduce a un lenguaje que el eBrick entiende, dependiendo del modelo del eBrick pueden existir bloques que no puedan ser manejados por el ladrillo, en esos casos ese bloque no genera código, es decir no se puede traducir al lenguaje del eBrick. Si esta opción esta seleccionada el

Probot Lab avisa cuando un bloque no se puede traducir y por la tanto no hace nada. Apagado no da ningún aviso.
Intentar usar un bloque de *"Reproducir Melodía"* en un modelo de ladrillo sin parlante es un ejemplo de bloque gráfico que no genera código.



NOTAS

- **Los parámetros indicados con [1] se guardan en el programa grafico y solo afectan al programa en donde fueron hechos los cambios.**

- **Los parámetros indicados con [2] se guardan en la configuración global del Probot Lab y afectan a todos los programas que se carguen.**

La Barra de Herramientas

En la barra de herramientas se Encuentran las opciones usadas más frecuentemente.

Nuevo
Limpia el área de programación para comenzar un nuevo programa.

Abrir
Abre y carga un programa.

Guardar (como)
Guarda el programa que se encuentra en el área de programación con un nuevo nombre.

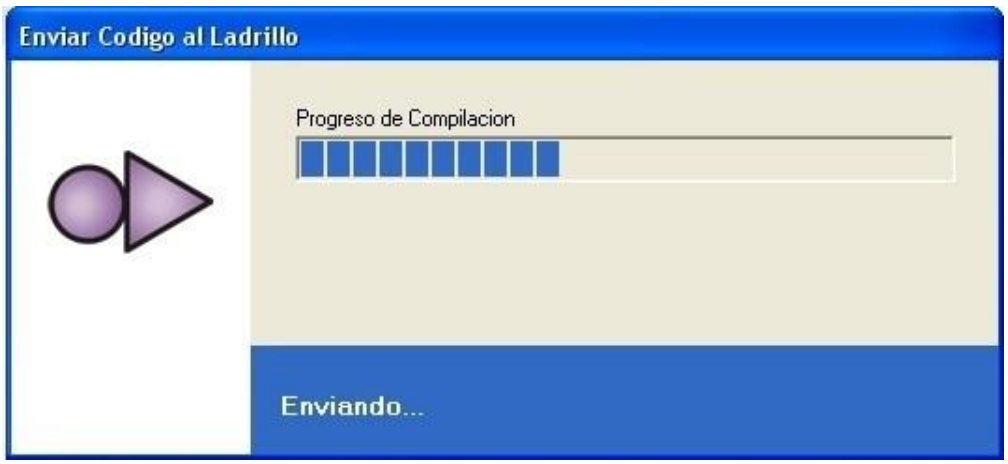
Ejecutar
Una vez que se ha grabado un programa en la memoria del eBrick esta función la hace correr (la ejecuta), siempre que el eBrick está conectado a la PC a través del cable de conexión.

Compilar

Traduce el programa al lenguaje que entiende el eBrick (código fuente).

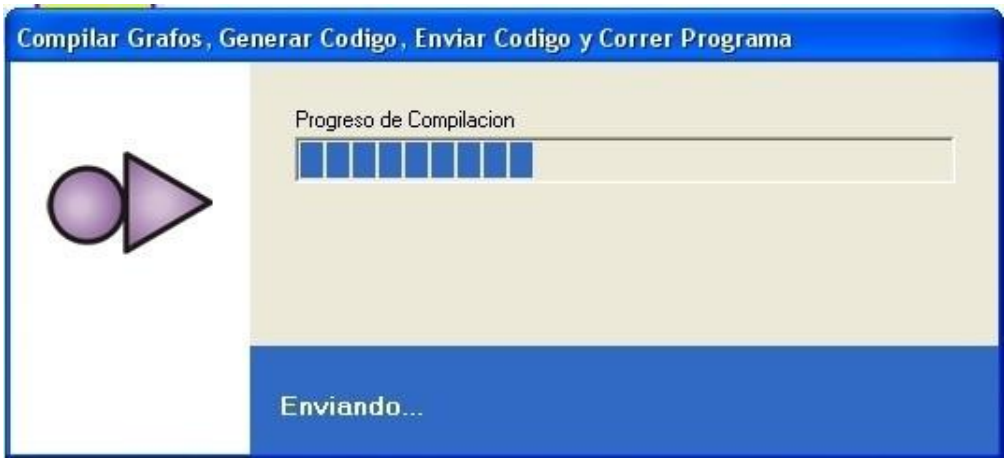
Grabar

Graba en la memoria del ladrillo el programa compilado, es decir que antes de usar esta función hay que compilar el programa gráfico.



CompiGraba

Traduce el programa grafico en el lenguaje de eBrick, lo graba en la memoria del Ladrillo y lo ejecuta.

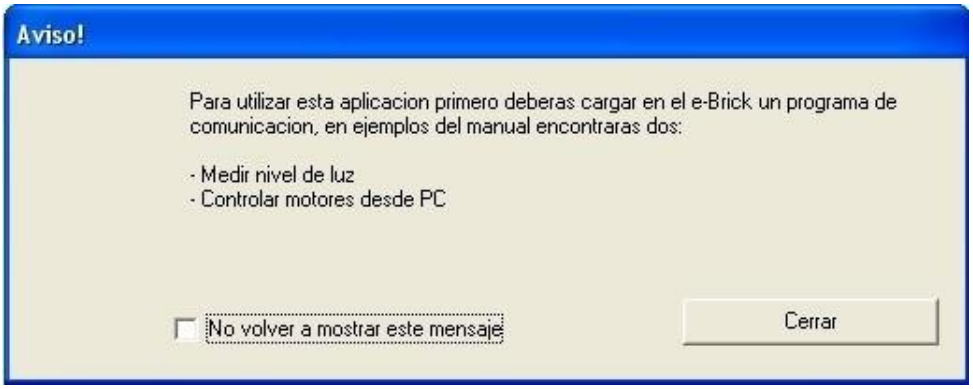


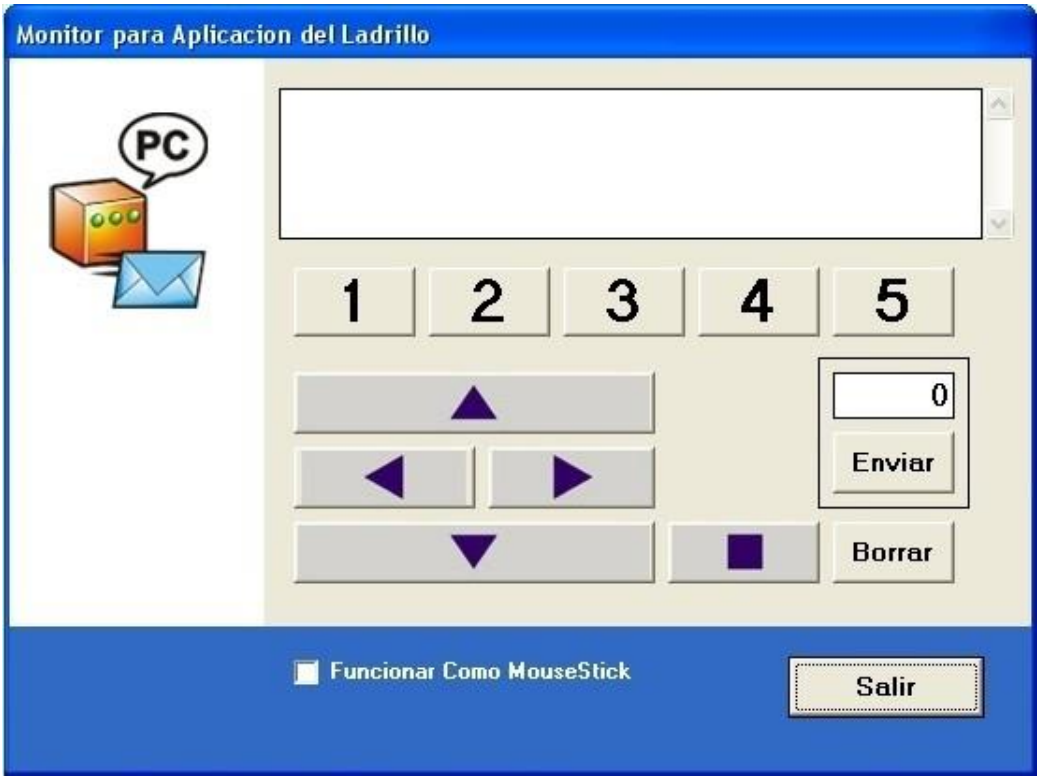
Conectar

Sirve para establecer la conexión entre el eBrick y la PC una vez conectado el cable y encendido el eBrick.

Monitor

Abre la herramienta de comunicación con el ladrillo que permite intercambiar datos con los programas que usan mensajes a VER PC.

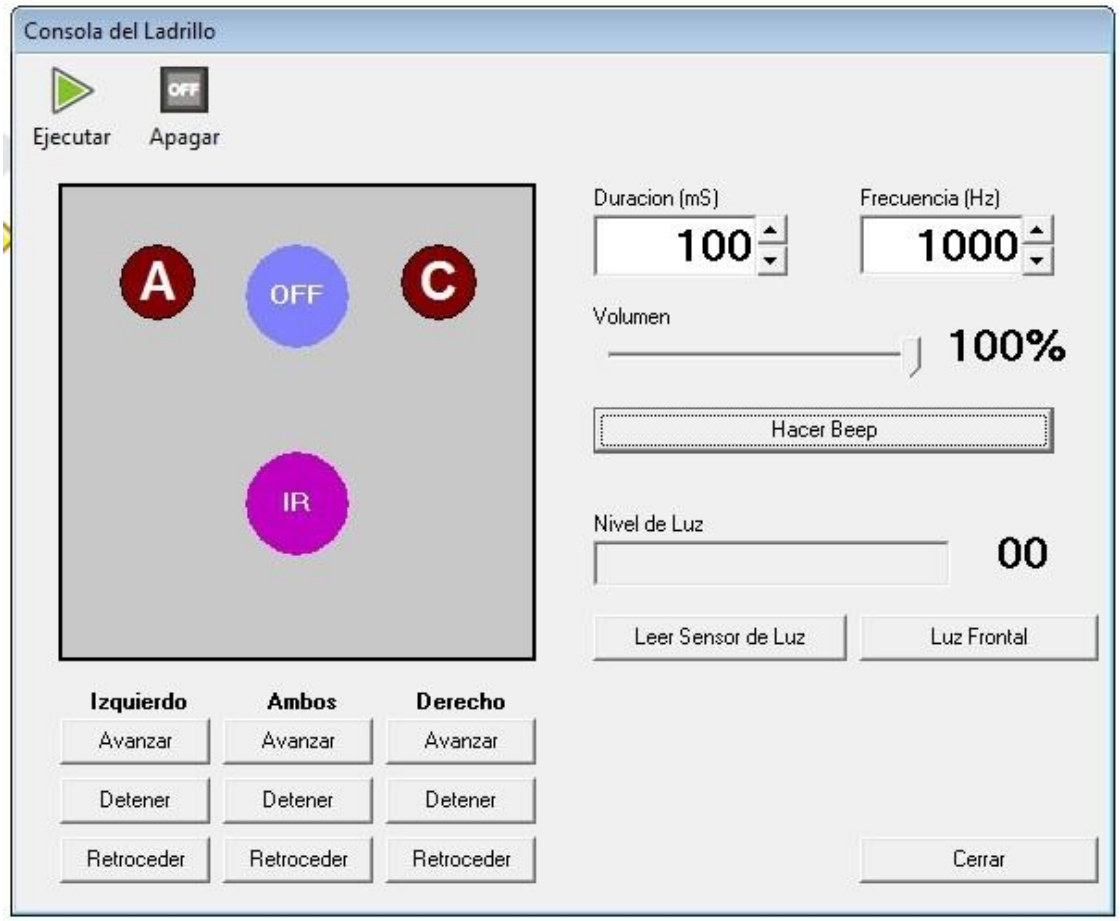




Desde esta interfaz, puedes dirigir los movimientos del sistema PROBOTS, selecciona los programas, conecta el eBrick física y virtualmente con el programa, compigraba, **no desconectes** el eBrick de la computadora.

Consola

Abre la herramienta de comunicación que permite manejar el ladrillo desde la PC. con esta herramienta puedes hacer un test de cada una de las partes del sistema PROBOTS.

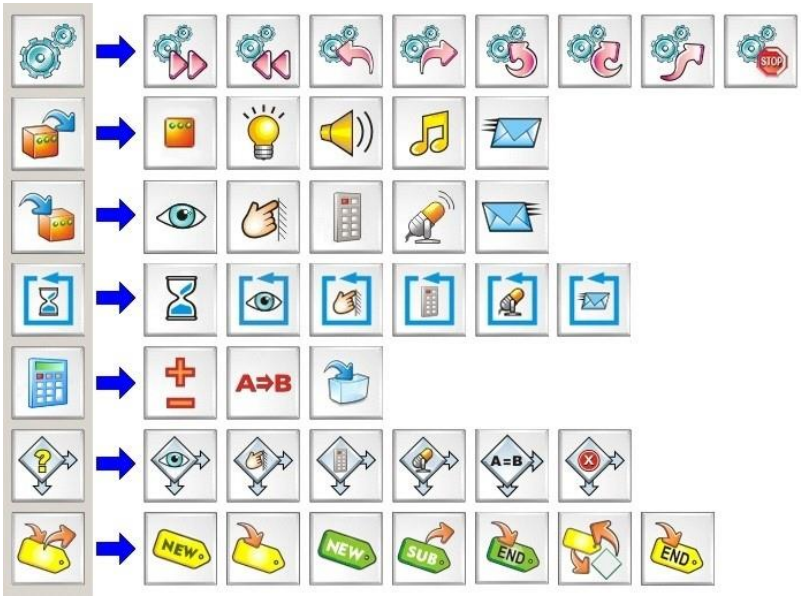


La Barra de Programación

Bueno cada vez estamos más cerca de empezar a programar y hacer que nuestro Probot se mueva y haga las cosas que nosotros queremos. En esta parte lo que vamos a explicar es que significan cada uno de los botones de la barra de programación y que acciones encierra cada uno de ellos.

Los botones que componen la barra de programación agrupan funciones con las que se crean los bloques de programación. Al presionar cualquiera de estos botones se despliegan menús con funciones especificas del grupo elegido, como por ejemplo, al presionar “Motores” se despliega un menú conteniendo las funciones de “Avanzar”, “Retroceder”, “Doblar”, etc.

La imagen que sigue muestra todos los menús que se despliegan de la barra de programación



A continuación se describe cada botón de la barra de programación, las funciones que se despliegan de cada una y los bloques de programación que generan.

Motores

Avanzar
Enciende los motores en sentido de avance. Al accionar este botón se abre una nueva pantalla donde permite seleccionar individualmente que motor se desea encender.

Retroceder
Enciende los motores en sentido de retroceso. Permite seleccionar individualmente que motor se desea encender.

Doblar (A la Izquierda)
Enciende los motores haciendo que el modelo doble hacia la izquierda.

Doblar (A la Derecha)
Enciende los motores haciendo que el modelo doble hacia la derecha.

Girar (En Sentido AntiHorario)
Enciende los motores haciendo que el modelo gire sobre si mismo en sentido antihorario.

Girar (En Sentido Horario)
Enciende los motores haciendo que el modelo gire sobre si mismo en sentido horario.

Recorrido
Esta opción permite *"dibujar"* un recorrido dentro del programa que estamos creando. Este recorrido será ejecutado cuando el programa llegue a este bloque y una vez finalizado continua con el programa normalmente.

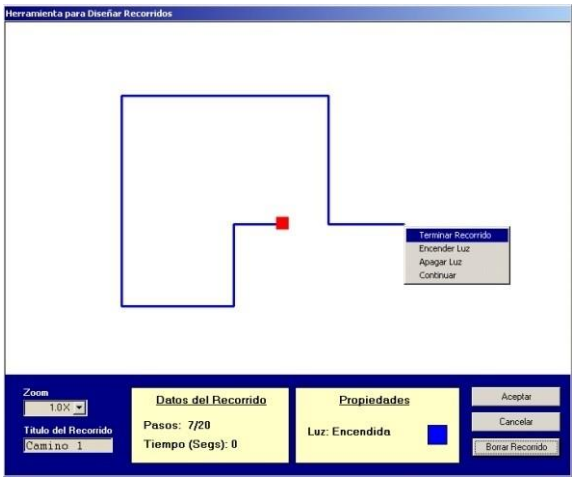
Configuración del recorrido
El sistema de recorrido se adapta a 2 tipos de sistema de dirección:

· Un motor independiente de cada lado (robot o escorpión)
· Un motor de tracción y uno de dirección (Vehículos)

Dependiendo del tipo de tracción el generador de recorrido genera un programa distinto, por lo tanto, es necesario indicarle que tipo de tracciones estamos usando. Esto se hace en el menú desplegable *"Configuración",* seleccionando la *"Solapa Motores"* y dentro de esta solapa ir al área de *"Configuración del Sistema de Dirección".*

En el sistema de dirección tipo Robot-Escorpión los tiempos de giros pueden variar dependiendo del estado de las pilas y el tipo de piso sobre el cual se mueva el modelo. Para ajustar estos tiempos y que los recorridos sean lo más parecido a los dibujados puedes cambiar el *"tiempo de giro para 45°"* desde el menú desplegable *"Configuración",* seleccionado la *"Solapa Motores"* y cambiando el valor indicado como *"Tiempo de Giro de 45° para Dirección de Motores Independientes".*

Área de Programación de recorrido



En la pantalla de recorrido nos encontraremos con un cuadrado rojo en el centro de la misma, este cuadrado representa al Probot. Ahora moviendo el mouse aparecerá una línea punteada que representa el movimiento que realizara el Probot en nuestro recorrido. Para terminar una línea y comenzar con una nueva tenemos que presionar el botón izquierdo del mouse, de esta manera comenzaremos una nueva línea a partir de la anterior. Para finalizar el recorrido tenemos que cliquear el botón derecho del mouse, apareciendo así un menú con las siguientes opciones que salen en el gráfico.

Cuando ya finalizamos presionamos sobre esta opción, de esta manera aparecerá un cuadrado indicando el final del recorrido. Para crear el bloque de “Recorrido” y volver a la pantalla de programación cliquear en “Aceptar”.

Encender Luz

Esta opción sirve para hacer que el Probot realice todo el recorrido o una parte de él con la luz frontal del eBrick encendida. Cuando seleccionamos la luz encendida las líneas del recorrido aparecen de color Azul.

Apagar Luz

Al igual que en la opción anterior podemos decirle al Probot que realice el recorrido completo o parte de él con la luz del eBrick apagada. Cuando seleccionamos esta opción las líneas del recorrido aparecerán de color Negro.

Continuar

Esta permite que continuemos dibujando nuestro recorrido.

Ahora te mostramos la parte inferior de la Pantalla Recorrido donde se encuentran algunas opciones más y ciertos datos informativos referidos a nuestro recorrido.

Zoom

Esta opción nos permite hacer zoom sobre el recorrido que estamos creando. Hay tres niveles de zoom 1.0X, 1.5X y 2.0X.

Titulo del Recorrido
Acá podemos nombrar nuestro recorrido. Cliqueamos sobre el nombre que esta por defecto y aparecerá el cursor para que podamos nombrarlo. De esta manera cuando se genere el bloque en el programa al poner el cursor sobre el aparecerá como ayuda rápida dicho nombre.

Datos del recorrido
Este cuadro es informativo y lo que nos dice es la cantidad de segmentos que hemos creado y la cantidad de máxima de segmentos permitida. El máximo permitido es de 20 Pasos (segmentos), pero puede variar según el recorrido.

Tiempo en segundos
Nos muestra cuantos segundos tardara en recorrer cada segmente creado.

Propiedades
Este cuadro también es informativo y nos indica mientras creamos el recorrido, si la luz está encendida o no.

Aceptar: Luego de finalizar el recorrido, para crear el bloque y volver al programa hay que presionar aquí.
Cancelar: Sale de la pantalla recorrido y vuelve al programa sin crear el bloque
Borrar recorrido: Borra el recorrido que se encuentra en la pantalla recorrido y nos permite seguir dibujando.

NOTA:
Para realizar recorridos es aconsejable usar el robot, ya que la tracción del escorpión suele tener movimientos meno s precisos y los recorridos no se ejecutaran debidamente.

Detener
Apaga los motores. Permite seleccionar individualmente que motor se desea encender. **Salidas**

Pantalla
Esta opción modifica el estado de los tres LEDs de la cara superior del eBrick y permite elegir el origen de datos que afectan a los LEDs. ¿Qué significa el origen de datos? Bueno llamamos origen de datos al lugar de donde proviene la variable que le indicara al eBrick que luz encender y cual apagar. Los orígenes posibles son, los Contenedores, una Constante, la opción Encender y la opción Apagar.

Origen de Datos: Contenedor (Rojo, Azul, Verde, Rosa y Naranja)

Si seleccionamos como origen a los contenedores tenemos que tener en cuenta que valores se encuentran guardados en los mismos, ya que de acuerdo a esos valores el eBrick encenderá o apagara algunos de los LEDs. A continuación vemos una tabla donde especifica los LEDs que se encienden de acuerdo al valor existente en el contenedor.

Los LEDs se encienden y se apagan según esta Tabla:

Valor del Contenedor	LEDS	
	Izq. [A]	Der. [C]
0	Apagado	Apagado
1	Apagado	*Encendido*
2	Apagado	Apagado
3	Apagado	*Encendido*
4	*Encendido*	Apagado
5	*Encendido*	*Encendido*
6	*Encendido*	Apagado
7	*Encendido*	*Encendido*

Origen de Datos: Constante

Eligiendo esta opción nos da la posibilidad de seleccionar los LEDs a encender (A, C) y apaga los no seleccionados.

Origen de Datos: Encender

Esta otra opción enciende los LEDs seleccionados (A, C) y no afecta los no seleccionado.

Origen de Datos: Apagar

Apaga los LEDs seleccionados (A, C) y no afecta los no seleccionado.

Luces

Modifica el estado del LED frontal de alto brillo del eBrick y permite elegir el origen de datos que afecta a los LED.

Origen de Datos: Contenedor (Roja, Azul, Verde, Rosa y Naranja) El LED se enciende si el valor contenido en el Contenedor es 1 (o si es impar) y se apaga si es 0 (o si es par).

Origen de Datos: Encender Enciende el LED.

Origen de Datos: Apagar Apaga el LED.

Sonido

Crea un bloque que produce un tono en el parlante del eBrick y permite seleccionar el modo de generar el tono.

Fuente: Constante

Genera el tono con la frecuencia, duración y volumen determinados con los controles que se encuentran en la ventana de “Salida de Sonido”.

Fuente: Contenedor (función Avanzada)

Para generar sonidos se requieren de 4 contenedores, 2 para el tiempo de duración y 2 para el tono a generar.

Rosa y Naranja: Frecuencia “tono agudo o grave”= (1/ ((Rosa + Naranja) * 8uS ‟microsegundos″))
Los contenedores Rosa y naranja determinan las frecuencia (mas baja, mas grave. Mas alta, mas aguda) Azul y Verde: Duración del tono en tiempo=(((Azul*256)+Verde)*(Rosa+Naranja)* 8uS “microsegundos″) Los contenedores Azul y Verde el tiempo que dura el sonido.

Para simplificar los cálculo al crear un bloque de sonido la pantalla calcula automáticamente los valores para el tono pedido.

Fuente: Sonido de OK

Produce un tono corto de alta frecuencia.

Fuente de Sonido de Error

Produce un tono largo y de baja frecuencia.

Notas
- Las funciones de sonido siempre cambian el contenido de las variables Rosa y Naranja.
- Mientras el tono se escucha el ladrillo no puede ejecutar otra instrucción.

Música
Abre la herramienta generadora de melodías. Los bloques creados con esta opción contiene una serie de notas que forman una melodía. Para agregar una nota en la melodía seleccione la nota o frecuencia, la duración y el volumen de la misma y presione “Agregar Nota”, una vez terminada la melodía presione “Aceptar” para que el bloque de programación aparezca en el programa.

Mensajes
Crea un bloque de programación que envía el valor de una variables por medio del cable de conexión a PC. Para recibir y mostrar fácilmente los valores que se envían a la PC con esta función, Probot Lab incluye la función “Monitor”.
Los valores enviados se pueden mostrar en Decimal (base 10), Hexadecimal (base 16), Binario (base 2) y ASCII. Para posicionar en la pantalla del Monitor que muestra los valores enviados, el bloque de mensaje incluye las siguientes opciones: · Borrar Pantalla Antes: Antes de mostrar el valor borra la pantalla del monitor

· Posiciona en Origen: Muestra el valor en la esquina superior izquierdo de la pantalla del monitor.

· Bajar de Línea: Muestra el valor una línea mas abajo que el anterior.

· Mostrar Sin Acción: Muestra el valor detrás del ultimo enviado.

Entradas

Luz
Crea un bloque de programación que lee el estado del sensor del luz de eBrick. Con esta función se pueden crear 2 tipos de bloques dependiendo del que opción seleccione de “Tipo de Entrada”. Seleccionando “Contar Pulsos” el bloque de programación generado cuenta cambios de luz de iluminado a oscuro, y numero de cambios se guarda en el Contenedor seleccionado, de este tipo de bloques termina cuando se vence el tiempo de TimeOut que se configura en “Sensores”.
Seleccionando “Intensidad” el sensor de luz envía el valor iluminación a el Contenedor indicado, el valor va de 0 (oscuro) a 15 (iluminado).

Tacto
El bloque de programación generado por esta opción cuenta el numero de veces que se presiona la tecla de encendido-apagado y deja el valor en el contenedor seleccionado, de este tipo de bloques termina cuando se vence el tiempo de TimeOut que se configura en “Sensores”. Esta función desactiva la tecla de encendido -apagado del ladrillo por lo tanto la única forma de apagarlo es desconectando las pilas.

Control Remoto
Crea un bloque de programación que lee el estado del sensor de control remoto del eBrick. Con esta función se pueden crear 2 tipos de bloques dependiendo de que opción seleccione como “Tipo de Entrada”. Seleccionando “Contar Pulsos” el bloque de programación generado cuenta cambios de estados del sensor y dicho numero se guarda en el contenedor seleccionado, de este tipo de bloques termina cuando se vence el tiempo de TimeOut que se configura en “Sensores”. Seleccionando “Datos” el eBrick lee los códigos que envía el control remoto “Mis Ladrillos” y los guarda en el contenedor seleccionado, de este modo se puede identificar la tecla que se presiona del control remoto según el numero devuelto.

Micrófono
Crea un bloque de programación que lee el estado del sensor del micrófono externo. Con esta función se pueden crear 2 tipos de bloques dependiendo del que opción seleccione de “Tipo de Entrada”. Seleccionando “Contar Pulsos” el bloque de programación generado cuenta cambios sonido-silencio, y numero de cambios se guarda en el contenedor seleccionado, de este tipo de bloques termina cuando se vence el tiempo de TimeOut que se configura en “Sensores”. Seleccionando “Frecuencia” el micrófono devuelve un valor proporcional a la frecuencia del sonido que escucha, ese valor es devuelto en el contenedor seleccionado.

Mensajes
En el bloque que crea esta opción, el programa se detiene hasta recibir un mensaje de la PC (a través del cable de conexión) , el valor recibido se guarda en el contenedor seleccionado.

Esperar…

Tiempo
Produce un bloque de demora. El tiempo de la demora puede definirse con una constante (un valor fijo) o un contenedor. La unidad de la espera puede prefijarse con las opciones de “Base de Tiempo”. Las bases permitidas son: 1 Seg, 0,1 Seg y 1 mSeg.

Para saber cuanto dura una demora multiplique el numero por la base, ejemplo: Valor= 43 y Base= 0,1 Seg produce una demora de 4,3 segundos.

Luz
Esta opción crea un bloque que deja al programa esperando cierto estado del sensor de luz del eBrick. Si el “Tipo de Espera” seleccionado es “Mientras”, el programa se detiene mientras el sensor este iluminado, si selecciona “Hasta” el programa se detiene hasta que el sensor este iluminado. El nivel de luz que se considera iluminado se setea en la pantalla de configuración del Probot Lab, solapa “Sensores”.

Tacto
Esta opcion crea un bloque que deja al programa esperando cierto estado del interruptor de encendido - apagado del eBrick. Si el “Tipo de Espera” seleccionado es “Mientras”, el programa se detiene mientras el interruptor este presionado, si selecciona “Hasta” el programa se detiene hasta que el interruptor este presionado.

Control Remoto
Esta opcion crea un bloque que deja al programa esperando cierto estado del sensor de control remoto del eBrick. Si el “Tipo de Espera” seleccionado es “Mientras”, el programa se detiene mientras en el sensor hay señal de control remoto, si selecciona “Hasta” el programa se detiene hasta que el sensor de control remoto vea señal.

Micrófono
Esta opción crea un bloque que deja al programa esperando cierto estado del micrófono externo. Si el “Tipo de Espera” seleccionado es “Mientras”, el programa se detiene mientras hay sonido, si selecciona “Hasta” el programa se detiene hasta que el micrófono escuche algún sonido.

Mensaje
Esta opción crea un bloque que deja al programa esperando cierto estado de la conexión del ladrillo con la PC. Si el “Tipo de Espera” seleccionado es “Mientras”, el programa se detiene mientras la PC envía datos, si selecciona “Hasta” el programa se detiene hasta que la PC envía algún mensaje. El mensaje recibido se pierde.

Cálculos

Matemáticas
Permite hacer operaciones de suma y resta entre variables.. En las operaciones matemáticas el contenedor resultado siempre es el mismo que el primer operando, el

segundo operando puede ser un contenedor o una constante. La forma genérica de estas operación es la siguiente: A = A +/- B

Lógica
Permite hacer operaciones lógicas entre variables y constante. En las operaciones lógicas el contenedor resultado siempre es el mismo que el primer operando, el segundo operando puede ser un contenedor o una constante. Las operación lógicas disponibles son: OR, AND y XOR. La forma genérica de estas operación es la siguiente:

A = A or/and/xor B **Cargar**

Variable
Permite cargar variables (contenedores). Una variable se puede cargar con una constante, un valor al azar u otra variable.

Decisión

Generalidades de las Decisiones
Todas la decisiones tienen dos respuestas posibles a la pregunta que hacen: si y no. Una de esas respuestas hacen que el programa ejecute el próximo bloque y la otra salte a una etiqueta, creando un bifurcación en la secuencia del programa. Las identificaciones de las salida (si y no) se pueden intercambiar usando la opción “Continuar por No, Saltar por Si”.

Por Luz
Crea un bloque de decisión que pregunta: Esta el sensor de luz iluminado? El nivel de luz que se considera iluminado se setea en la pantalla de configuración del Probot Lab en la solapa de “Sensores”.

Por Tacto
Crea un bloque de decisión que pregunta: Esta el interruptor de encendidoapagado? Esta función desactiva la tecla de encendido-apagado del ladrillo por lo tanto la única forma de apagarlo es desconectando las pilas.

Por Control Remoto
Crea un bloque de decisión que pregunta: Hay señal de control remoto?

Por Sonido
Crea un bloque de decisión que pregunta: Hay sonido?

Comparar

Genera un bloque que compara un contenedor con un contenedor o una constante. Las combinaciones posibles son las siguientes:

Pregunta	A	comparador	B
El contenedor A es igual a el contenedor B?	Contenedor	=	Contenedor
El contenedor A es distinta de contenedor B?	Contenedor	?	Contenedor
El contenedor A es mayor que el contenedor B?	Contenedor	>	Contenedor
El contenedor A es menor que el contenedor B?	Contenedor	<	Contenedor
El contenedor A es mayor o igual que el contenedor B?	Contenedor	?	Contenedor
El contenedor A es menor o igual que el contenedor B?	Contenedor	?	Contenedor

Pregunta	A	comparador	B
El contenedor A es igual a la Constante B?	Contenedor	=	Constante
El contenedor A es distinta de Constante B?	Contenedor	?	Constante
El contenedor A es mayor que la Constante B?	Contenedor	>	Constante
El contenedor A es menor que la Constante B?	Contenedor	<	Constante
El contenedor A es mayor o igual que la Constante B?	Contenedor	?	Constante
El contenedor A es menor o igual que la Constante B?	Contenedor	?	Constante

Por Error

El bloque que crea esta opción pregunta si hubo error en la ultima operación de suma realizada. Las variables pueden contener valores de 0 a 255. Pero que ocurre si una operación de suma da mas de 255? En ese caso el eBrick genera un estado interno de Error que puede ser manejado con este bloque de decisión. En el caso de la resta el indicador de Error opera a revés, si la operación de resta da un resultado valido indica

error, pero si el valor esta fuera del rango valido indica que no hay error, por lo tanto debe usarse lógica inversa en este caso.

Saltos

Etiqueta

Crea una nueva etiqueta. Una etiqueta indica una posición en el programa identificada con un nombre, y los bloque de programación que pueden saltar (como las decisiones o lo saltos) usan como destino de esos saltos el nombre de una etiqueta creada previamente. Hay ciertas reglas que se deben tener en cuentas con las etiquetas:

- No puede haber dentro de un modulo de programación 2 etiquetas con el mismo nombre.
- El nombre “Inicio” como etiqueta no es valido porque ya existe una etiqueta con ese nombre (que es la primera instrucción de todos los programas)
- Los saltos que hacen referencia a una etiqueta que no existe generan errores de compilación.ç

Saltar

El bloque generado por esta función fuerza un salto a una etiqueta. Si después de un salto el programa continua, el bloque siguiente al salto debe ser una etiqueta.

Nueva Sub

Al crear una nueva subrutina, Probot Lab abre un nuevo espacio de programación donde se puede escribir un subprograma que se pueden llamar desde el programa principal. Las subrutinas hacen los programas mas leíbles y mas compactos.

Ir a Sub

Esta función genera un bloque que llama a una subrutina, cuando la subrutina se termina el programa continua ejecutando el bloque siguiente al llamado de la subrutina.

Fin de Sub

Crea un bloque de fin de subrutina. Las subrutinas siempre deben terminar con este bloque para retornar al punto del programa donde la subrutina fue llamada. Si una subrutina no termina con este bloque genera un error de compilación.

Repetir

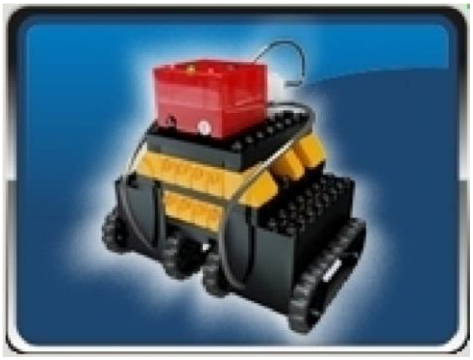
Crea un bloque de realiza las siguientes operaciones: le resta 1 a el contenedor seleccionado (Repetición) y si el contenedor no es 0 salta a la etiqueta, si es 0 ejecuta el próximo bloque. Al hacer esta secuencia el bloque de “Repetir” salta a la etiqueta tantas veces como lo indique el contenedor de repetición.

Fin de Programa

Crea un bloque que indica fin de programa. Este bloque no puede usarse en una subrutina. No es obligatorio el uso de esta función pero su uso hace mas claros a los programas. El eBrick tiene las siguientes formas de terminar un programa:

- Apagar el Ladrillo
- Devolver el mando a la PC (vuelve a estado “conectado”)
- Reiniciar el Ladrillo (se resetea sin hacer la secuencia de leds inicial)
- Entrar en un ciclo sin fin (el ladrillo solo se puede apagar) Saltar a Inicio (Repetir el programa)

Las teclas de Programación Rápida



Al cliquear los dibujos de los modelos del área de programación (estas imágenes son de lista de programas listos para usar para ese modelo. Para usar los programas, seleccione el que desea usar de la lista, conecte el ladrillo a la PC, enciéndalo y presione *“Conectar”* de la barra de herramientas, finalmente presione *“CompiGraba”* y el programa estará corriendo en el ladrillo.

Nota:

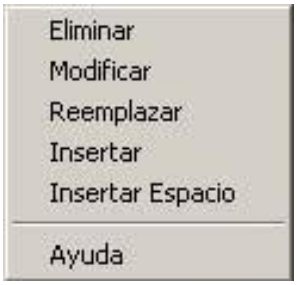
Las imágenes de los modelos cabian según versión de ProbotLab

El Área de Programación

En el área de programación aparecen los bloques que identifican la acción que con las siguientes opciones:

Herramientas de edición de bloques de programación

Cliqueando con el mouse sobre un bloque de programación aparece un menú **Eliminar:**



Borra el bloque seleccionado.

Modificar: Permite editar las propiedades del bloque seleccionado.

Reemplazar: Marca el bloque seleccionado para reemplazarlo por uno nuevo, una vez seleccionado elija otro bloque de la barra de programación y ese bloque reemplazara el marcado, para cancelar la operación hay una tecla de cancelación en la barra de información (extremo derecho).

Insertar: Marca el bloque seleccionado para insertar uno nuevo antes del seleccionado, una vez seleccionado elija otro bloque de la barra de programación y ese bloque aparecerá delante del marcado, para cancelar la operación hay una tecla de cancelación en la barra de información (extremo derecho).

Insertar espacio: Inserta un espacio en blanco en el lugar del bloque seleccionado corriendo todos los bloques siguientes una posición hacia delante. Esta función permite separa partes para hacer mas leíble un programa.

Ayuda: Abre una pantalla de ayuda detallada del bloque seleccionado. Dejando el cursor del mouse sobre cualquier bloque de programa aparecerá la descripción de la operación que dicho bloque realiza y entre corchetes el numero de bytes que ocupa en memoria.

La Barra de Información

En esta zona de la interfaz muestra información general del Probot Lab y del programa que se esta dibujando.

- **USB:** Tipo de conexión que esta usando para comunicarse con el ladrillo
- **Ladrillo:** Modelo del ladrillo que esta conectado
- **Librería:** Modelo del ladrillo para el cual genera programas (para que todo funcione bien debe ser el mismo modelo que el del ladrillo que esta conectado) • **Objetos:** Numero de bloque visibles / Numero de bloques totales
- **Memoria:** Memoria en uso / Memoria Disponible
- **Editor:** Indica el modo en que esta trabajando el editor de programas:
  Normal : los bloques se agregan al final
  Reemplazar : cambia el bloque seleccionado por uno nuevo
  Insertar : agregar un nuevo bloque delante del seleccionado

Como Mover tu PROBOT

A continuación se muestra como crear diversos programas (algunos simples y otros no tanto)

Programa 1

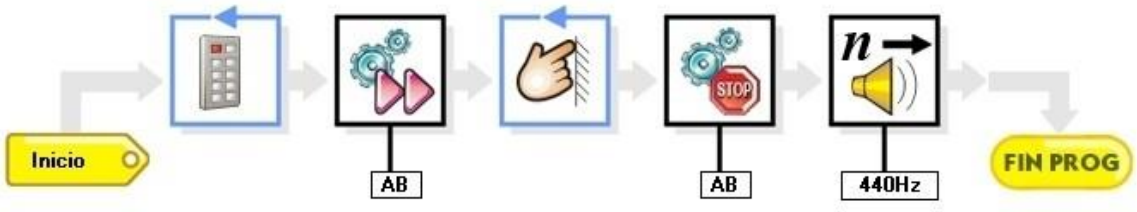
Al correr este programa el modelo avanzara durante 2 segundos luego gira en sentido opuesto a las agujas del reloj durante 1 segundo y finalmente se detiene apagando el ladrillo.

El programa consiste en 7 bloques, cada bloque realiza una funcion especifica dentro de la secuencia del programa. Analicemos este programa paso a paso para comprender su funcionamiento:

Bloque 1
Inicia el programa, todos los programas de Probot Lab arrancan en esta etiqueta.
Bloque 2
Enciende motores en sentido de avance, (A es el motor Izquierdo y B el derecho), dado que debajo se indican las letras A y B se encenderán ambos motores. **Bloque 3**
El programa se detiene durante 2 segundos (los motores siguen en funcionamiento durante este periodo). **Bloque 4**
Enciende los motores para girar en sentido anti-horario. **Bloque 5**
El programa se detiene durante 1 segundo. **Bloque 6**
Apaga motores, (A es el motor Izquierdo y B el derecho), dado que debajo se indican las letras A y B se apagaran ambos motores. **Bloque 7**
Fin del programa, este bloque puede funcionar de varias maneras, en este caso fue configurado para apagar el ladrillo. (para ver como esta configurado ubica el cursor del mouse sobre el bloque y la ayuda te indicara como termina el programa). **Del ejemplo anterior podemos destacar**

· Todos los programas tienen un punto de inicio bien definido, para Probot Lab es la etiqueta de Inicio, todos los programas que escribas comenzaran en este punto · Los motores tienen “memoria” propia en el sistema eBrick, es decir que si les damos una orden –por ejemplo Avanzar- los motores seguirán avanzando sin importar que operación esta realizando el ladrillo programable –en el caso del ejemplo una espera- y solo cambian de estado cuando se ejecuta otro bloque de motor.
Los programas tienen un punto de finalización, en este lenguaje grafico el uso de Fin de Programa no es obligatorio, pero para evitar que tus programas realicen acciones inesperadas es recomendable terminarlos en un bloque de “Fin de Programa”.

Ejemplo 2



Este programa hace que el ladrillo inteligente quede detenido hasta que se presione cualquier tecla de un control remoto (puede ser el control de un televisor, DVD o similar). Al recibir la orden del control remoto el modelo avanza hasta que un objeto choque y presione el sensor de tacto del eBrick, en ese momento ambos motores se detendrán, se emitirá un tono y la secuencia se repetirá desde el comienzo.

Bloque 1
Inicia el programa, todos los programas de Probot Lab arrancan en esta etiqueta.

Bloque 2
El eBrick se detiene hasta que se detecte una señal de control remoto.

Bloque 3
Enciende motores en sentido de avance, (A es el motor Izquierdo y B el derecho), dado que debajo se indican las letras A y B se encenderán ambos motores.

Bloque 4
El programa se detiene Hasta que se detecte un objeto con el sensor de tacto.

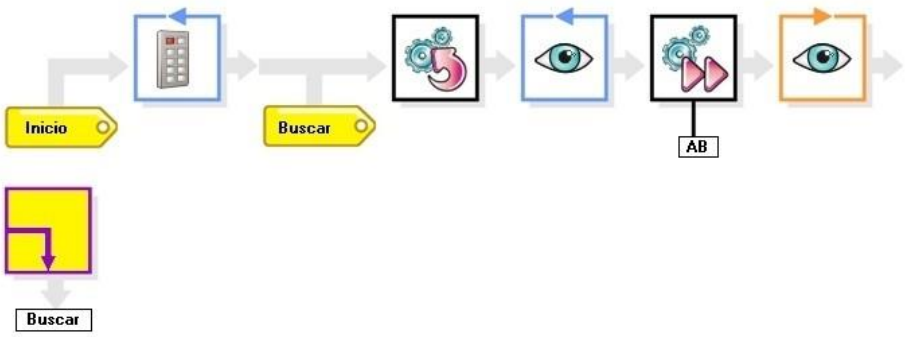
Bloque 5
Apaga motores, (A es el motor Izquierdo y B el derecho), dado que debajo se indican las letras A y B se apagaran ambos motores.

Bloque 6
El eBrick emite un tono de 440Hz durante medio segundo (0,5 seg).

Bloque 7
Fin de Programa, en este caso el fin de programa funciona repitiendo todo desde el inicio.

Del ejemplo anterior podemos destacar:

- Se pueden hacer esperas con un sensor, a diferencia del ejemplo 1 que las demoras eran fija (por tiempo) este programa se detiene hasta que el sensor de control remoto detecte un señal. Como concepto general en Probot Lab hay 2 tipos de esperas contra sensores: Esperar Hasta que (Recuadro celeste) y Esperar Mientras que (Recuadro naranja).
- Para cambiar el tiempo de espera del control remoto se puede cambiar en “Configuracion\Sensores\Tiempo de TimeOut para Conteo de Pulsos”.



Ejemplo 3

Este programa hace que el ladrillo inteligente quede detenido hasta que se presione cualquier tecla de un control remoto (puede ser el control de un televisor, DVD o similar). Al recibir la orden del control remoto el modelo comienza a girar en sentido anti-horario hasta que encuentra una luz brillante, al encontrarla comienza a avanzar si la luz desaparece volverá a girar buscando luz nuevamente.

Bloque 1
Inicia el programa, todos los programas de Probot Lab arrancan en esta etiqueta.

Bloque 2
El eBrick se detiene Hasta que se detecte una señal de control remoto.

Bloque 3
Etiqueta “Buscar”.

Bloque 4
El modelo gira sobre si mismo en sentido ant-horario.

Bloque 5
Espera Hasta que el sensor de luz encuentre un punto iluminado.

Bloque 6
Enciende motores en sentido de avance, (A es el motor Izquierdo y B el derecho), dado que debajo se indican las letras A y B se encenderan ambos motores.

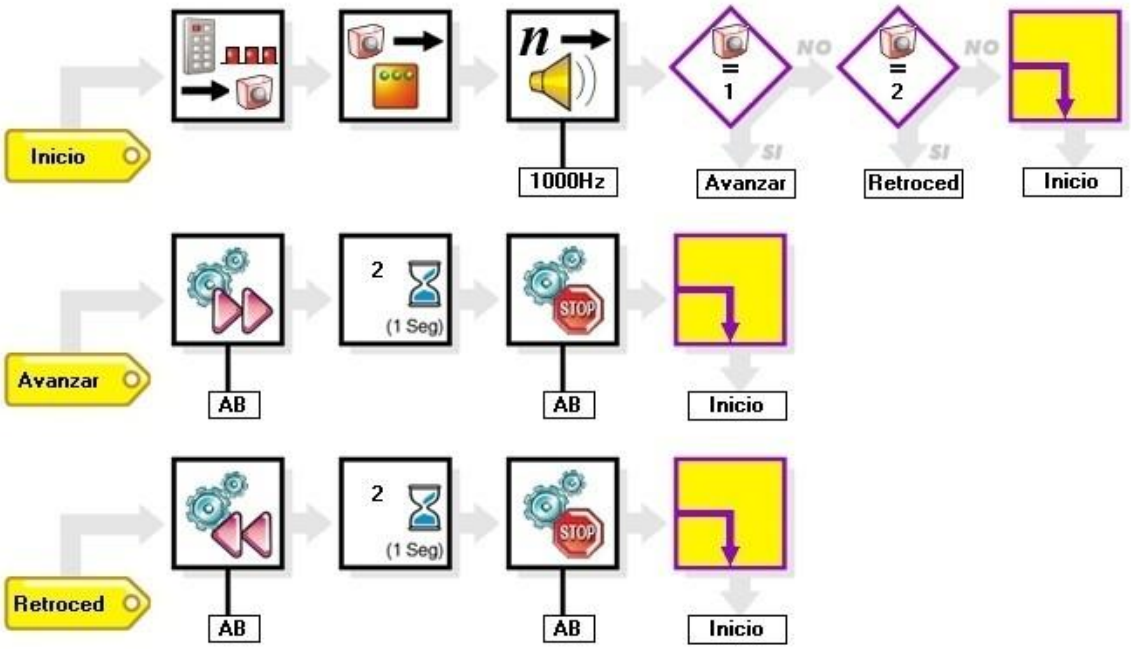
Bloque 7
Espera mientras el sensor este iluminado (busca oscuridad).

Bloque 8
Salta a la etiqueta “Buscar”.

Del ejemplo anterior podemos destacar:

- Aparece el bloque Etiqueta, las etiquetas son los destinos de los saltos y tienen las siguientes particularidades:
  - No puede haber 2 etiquetas con el mismo nombre
  - No ocupan memoria en el ladrillo
  - El nombre de las etiquetas puede tener hasta 8 caracteres

Este programa se ejecuta sin fin, la única forma de detenerlo es apagando el ladrillo. El ultimo bloque del programa es un salto, cuando un bloque salta lo que esta haciendo es cambiar el orden en que se ejecutan las instrucciones, en vez de ejecutar la siguiente retoma la ejecución desde la etiqueta que esta llamando, en este caso “Buscar”.



Ejemplo 4 (censores y decisiones)

Este programa cuenta las veces que se presiona un tecla en un control remoto, si se presiono una vez el modelo avanza durante 2 segundos, si se presiono 2 veces el modelo retrocede 2 segundos. Antes de ejecutar la acción hace un bip y enciende los leds indicando la orden recibida.

Bloque 1

Etiqueta de Inicio el programa.

Bloque 2

El eBrick se detiene esperando señal de control remoto, cada señal recibida hace que el contenedor rojo se incremente en 1, si durante un tiempo (1) no se recibió ninguna señal sale con el contendor rojo en 0.

Bloque 3

Envía a la pantalla de Leds en contenido del contendor rojo, el valor es mostrado en binario según la siguiente tabla.

Bloque 4

Produce un tono de 1000Hz durante 0,2 segundos (cuanto mas alta es la frecuencia mas agudo es el sonido).

Bloque 5

El contenedor rojo es igual a 1? Si la respuesta es Si salta a “Avanzar”, si es No pasa al próximo Bloque.

Bloque 6
El contenedor rojo es igual a 2? Si la respuesta es Si salta a “Retroced”, si es No pasa al próximo Bloque.

Bloque 7
Salta a Inicio (repite todo lo anterior).

Bloque 8
Etiqueta “Avanzar”.

Bloque 9
Enciende motores en sentido de avance, (A es el motor Izquierdo y B el derecho), dado que debajo se indican las letras A y B se encenderán ambos motores.

Bloque 10
El ladrillo detiene el programa durante 2 segundos (los motores siguen funcionando).

Bloque 11
Apaga motores, (A es el motor Izquierdo y B el derecho), dado que debajo se indican las letras A y B se apagaran ambos motores.

Bloque 12
Salta a Inicio.

Bloque 13
Etiqueta “Retroced”.

Bloque 14
Enciende motores en sentido de Retroceso, (A es el motor Izquierdo y B el derecho), dado que debajo se indican las letras A y B se encenderán ambos motores.

Bloque 15
El ladrillo detiene el programa durante 2 segundos (los motores siguen funcionando).

Bloque 16
Apaga motores, (A es el motor Izquierdo y B el derecho), dado que debajo se indican las letras A y B se apagaran ambos motores.

Bloque 17
Salta a Inicio

Del ejemplo anterior podemos destacar:

· Los sensores pueden leerse y el valor recibido es almacenado en un contenedor, dependiendo de la naturaleza del sensor se puede leer intensidad, datos o conteo de pulsos como en este ejemplo.
Probot Lab permite leer los siguientes datos de los sensores:

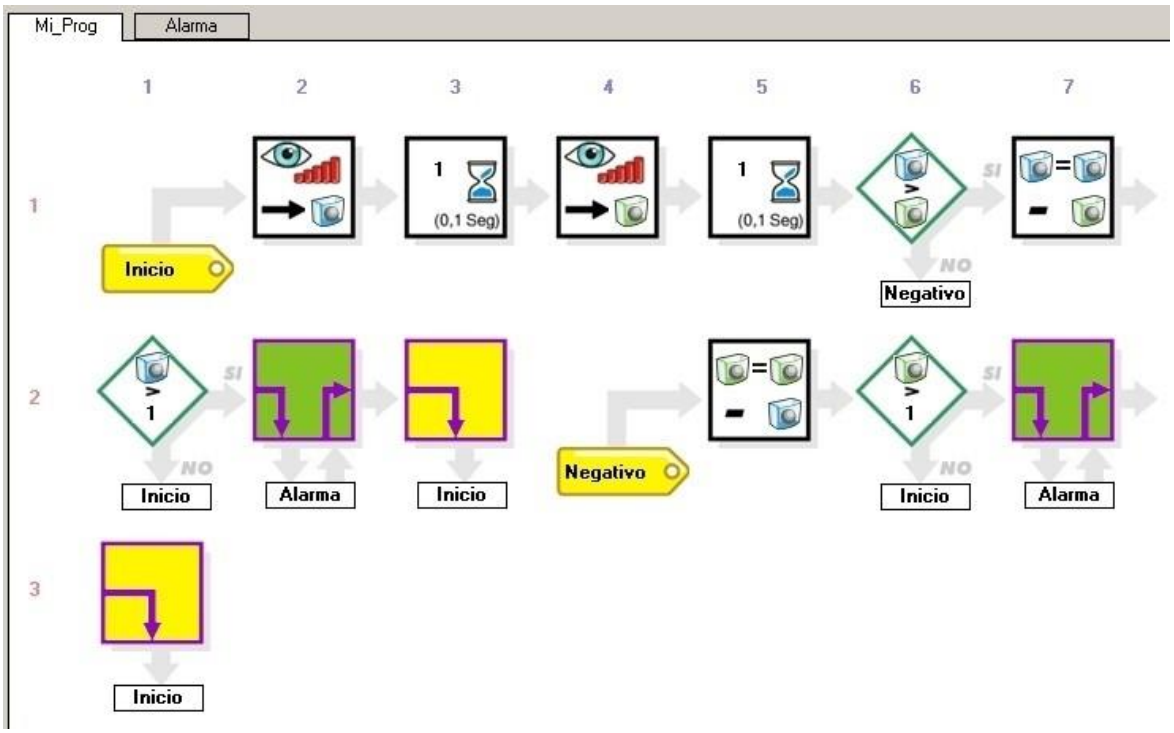
- Conteo de pulsos de control remoto
- Datos de control remoto (solo para controles de sistema Probot)
- Nivel de intensidad de brillo del sensor de luz
- Conteo de pulsos del sensor de luz
- Conteo de pulsos del sensor de choque (interruptor de encendido-apagado)

- Datos de la PC vía cable de comunicación
- Frecuencia de sonido en microfono (versiones que soportan micrófono full)
- Conteo de pulsos de sonido en el micrófono (versiones que soportan micrófono) Cuando los sensores cuentan pulsos (como en este ejemplo) tienen un tiempo en cual sino se detecta ninguno nuevo pulso terminan de esperar y ejecutan el próximo bloque, dicho tiempo se puede configurar en el menu desplegable “Configuración\Sensores\ Tiempo de TimeOut para conteo de pulsos”.

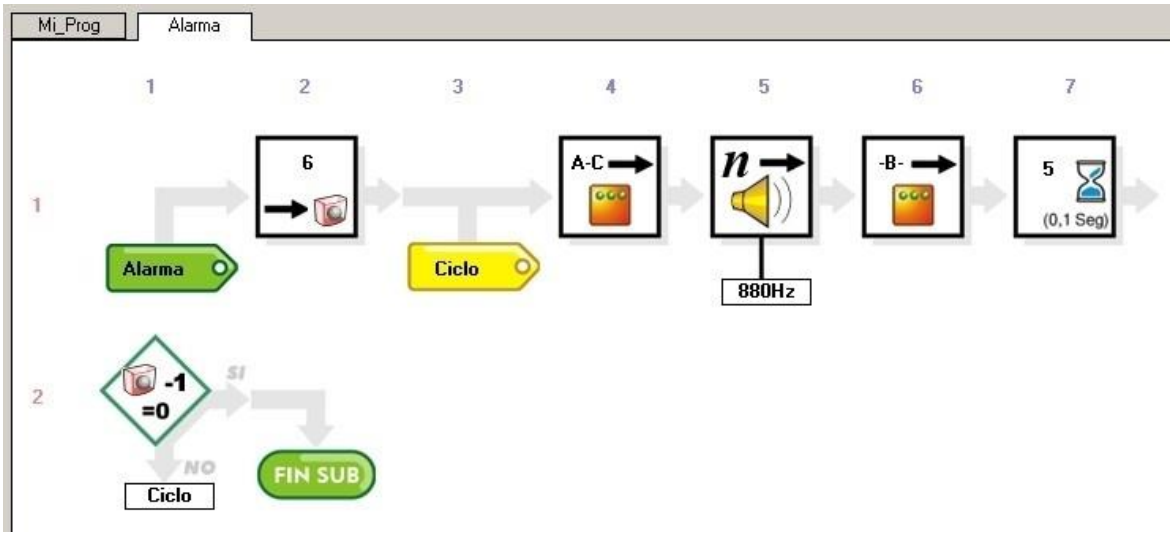
· Aparece el bloque de decisión, es decir que se hace una pregunta (el valor de un contenedor, el estado de un sensor, etc) y en función de la respuesta se ejecuta el próximo bloque o salta a una etiqueta.
Los bloques de decisión pueden seleccionarse para saltar si la respuesta es “Si” (rombo violeta) o bien si es “No” (rombo verde).

Ejemplo 5 (uso de subrutinas)

Cuerpo principal del programa



Subrutina Alarma



Este programa hace que el ladrillo programable funcione como una alarma detectora de movimiento. Cuando el sensor de luz detecta una variacion emite un bip que se repite 6 veces y luego vuelve a funcionar como sensor.

Cuerpo principal del programa

Bloque 1
Etiqueta de Inicio el programa

Bloque 2
Lee el nivel de señal en el sensor de luz y almacena ese valor en el contenedor azul.

Bloque 3
Espera 0,1 segundos

Bloque 4
Lee el nivel de señal en el sensor de luz y almacena ese valor en el contenedor verde.

Bloque 5
Espera 0,1 segundos

Bloque 6
Si el contenedor azul tiene un valor mayor que el verde pasa al próximo bloque, sino salta a la etiqueta “Negativo”

Bloque 7
Hace la siguiente operación matemática: Azul = Azul – Verde

Bloque 8
Si el contenedor azul es mayor que 1 ejecuta el próximo bloque (porque detecto una variación de luz), sino salta a la etiqueta “Inicio”.

Bloque 9
Llama a la subrutina “Alarma”

Bloque 10
Salta a la etiqueta “Inicio”

Bloque 11
Etiqueta “Negativo”

Bloque 12
Hace la siguiente operación matemática: Verde = Verde – Azul

Bloque 13
Si el contenedor verde es mayor que 1 ejecuta el próximo bloque (porque detecto una variación de luz), sino salta a la etiqueta “Inicio”.

Bloque 14
Llama a la subrutina “Alarma”

Bloque 15
Salta a la etiqueta “Inicio”

Subrutina “Alarma”

Bloque 1
Etiqueta de Inicio de Subrutina “Alarma”
Bloque 2
Cargar el contendor rojo con el numero 6. (numero de veces que va a repetir el ciclo)
Bloque 3
Etiqueta “Ciclo”
Bloque 4
Enciende los Leds de la punta de la pantalla

Bloque 5
Emite un tono durante medio segundo
Bloque 6
Enciende el Led central de la pantalla
Bloque 7
Hace una demora de medio segundo
Bloque 8
Le resta 1 al contendor rojo, si es resultado es 0, pasa al próximo bloque, sino salta a la etiqueta “Ciclo”. Esto hace que se salte tantas veces a “Ciclo” como indique el valor del contendor rojo. **Bloque 9**
Fin de Subrutina, retoma la ejecución desde el bloque que sigue al llamado.

Del ejemplo anterior podemos destacar:

· Surge un elemento de programación nuevo, *la subrutina*. Una subrutina es un pequeño programa, por lo tanto tiene inicio y fin propio. La principal particularidad de una subrutina es que cuando termina de ejecutarse retorna al bloque que la llamo, de modo que el programa principal se sigue ejecutando normalmente una vez terminada la subrutina. El uso de subrutinas permite hacer programas mas fáciles de leer ya que permite dividir el programa en partes mas fáciles de analizar y hace que los programas ocupen menos memoria ya que una secuencia que se usa mas de una vez puede convertirse en una subrutina y ser llamada tantas veces como se la necesite, como se puede ver en el ejemplo, donde la subrutina “Alarma” se llama dos veces.

· Dentro de la subrutina “Alarma” se usa por primera vez el bloque de Repetir. Usando este bloque se puede generar un ciclo que se repite tantas veces con indique el contenido de un contener. Funciona de la siguiente maneja: cuando el bloque se ejecuta le resta 1 al contenedor seleccionado, luego pregunta si el contendor quedo en 0, si es asi ejecuta el próximo bloque, sino salta a la etiqueta. Siempre que se termina el ciclo el contenedor queda en 0.

Notas:

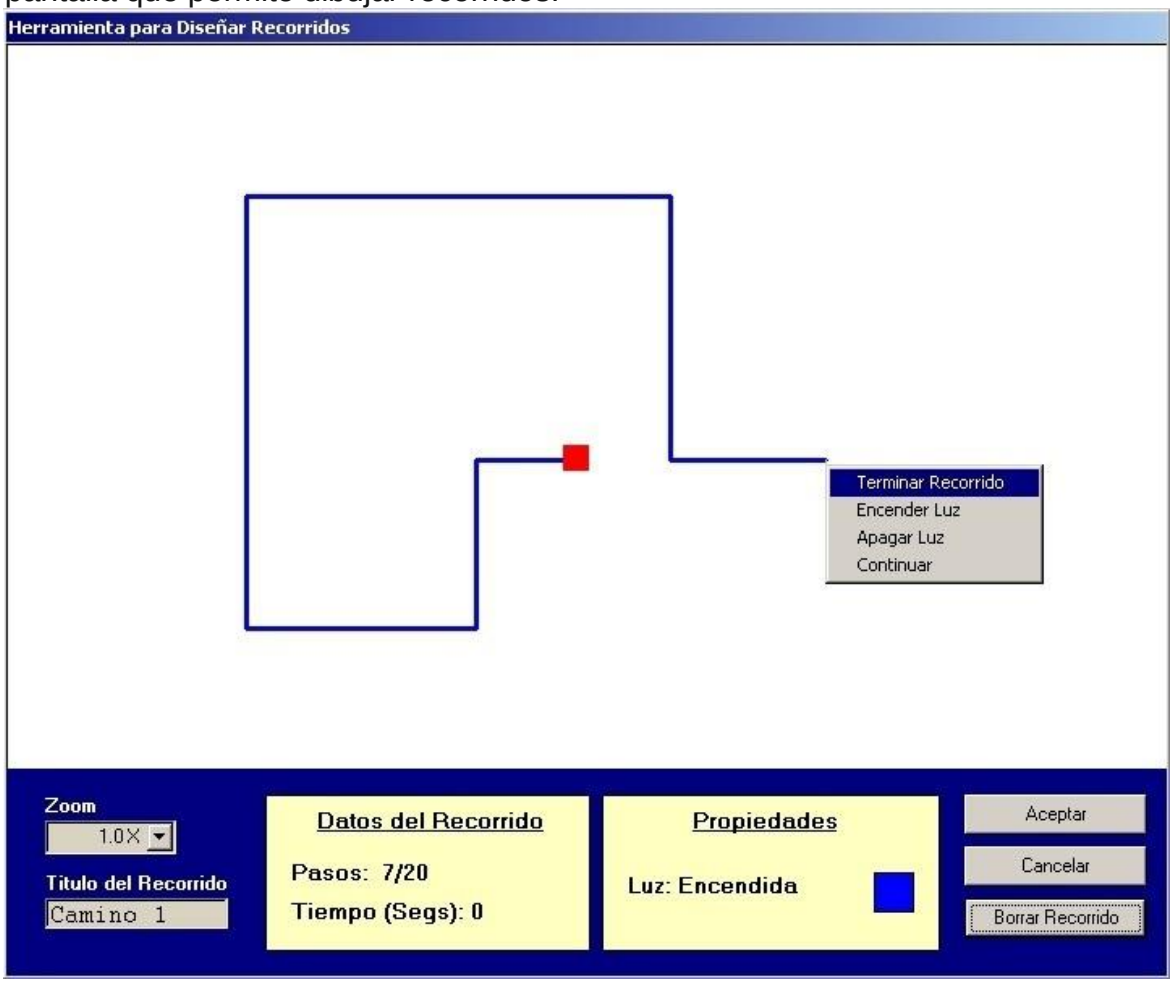
- No se pueden crear 2 subrutinas con el mismo nombre
- No se puede llamar a una subrutina desde otra subrutina Las subrutinas SIEMPRE deben terminar en “fin de sub”

Bloques con Herramientas integradas

Motores

Recorrido

Al seleccionar el bloque de Recorrido (dentro de la motores) se abrirá la siguiente pantalla que permite dibujar recorridos:



En la pantalla de recorrido encontraras una área donde se pueden dibujar líneas, cliqueando con el botón izquierdo del mouse se termina una línea y se inicia una nueva, con el botón derecho aparecerán las siguientes opciones:

Terminar Recorrido
Encender Luz
Apagar Luz
Continuar

Al *terminar el recorrido* ambos motores se apagaran y el bloque se da por terminado. Usando los controles de luz el modelo puede cambiar el estado de la luz frontal del eBrick mientras se realiza el recorrido. *Continuar* permite seguir dibujando el recorrido.

Cuando el trazo es azul la luz frontal estará encendida, cuando es negro estará apagada.

En la parte inferior se encuentra el control “Zoom”, modificando este valor se puede ampliar el recorrido generado, es decir que mantiene el dibujo pero lo hace mas pequeño o mas grande dependiendo del valor de zoom seleccionado.
Al recorrido puede asignarsele un nombre de modo que cuando se genere el bloque en el programa al poner el cursor sobre el aparece como ayuda rápida dicho nombre.

En el área de “Datos del Recorrido” podemos ver cuantos segmentos tiene el dibujo (el numero máximo de segmentos es 20 y puede variar dependiendo del recorrido) y debajo el tiempo que le llevara al modelo recorrer esa distancia.

Configuración del recorrido

El sistema de recorrido se adapta a 2 tipos de sistema de dirección:

- Un motor independiente de cada lado (robot o escorpión)
- Un motor de tracción y uno de dirección (Vehículos)

Dependiendo del tipo de tracción el generador de recorrido genera un programa distinto por lo tanto es necesario indicarle que tipo de tracciones estamos usando, esto se hace en el menú desplegable “Configuración”, seleccionado la solapa motores y dentro del área de “Configuración del sistema de Dirección”.

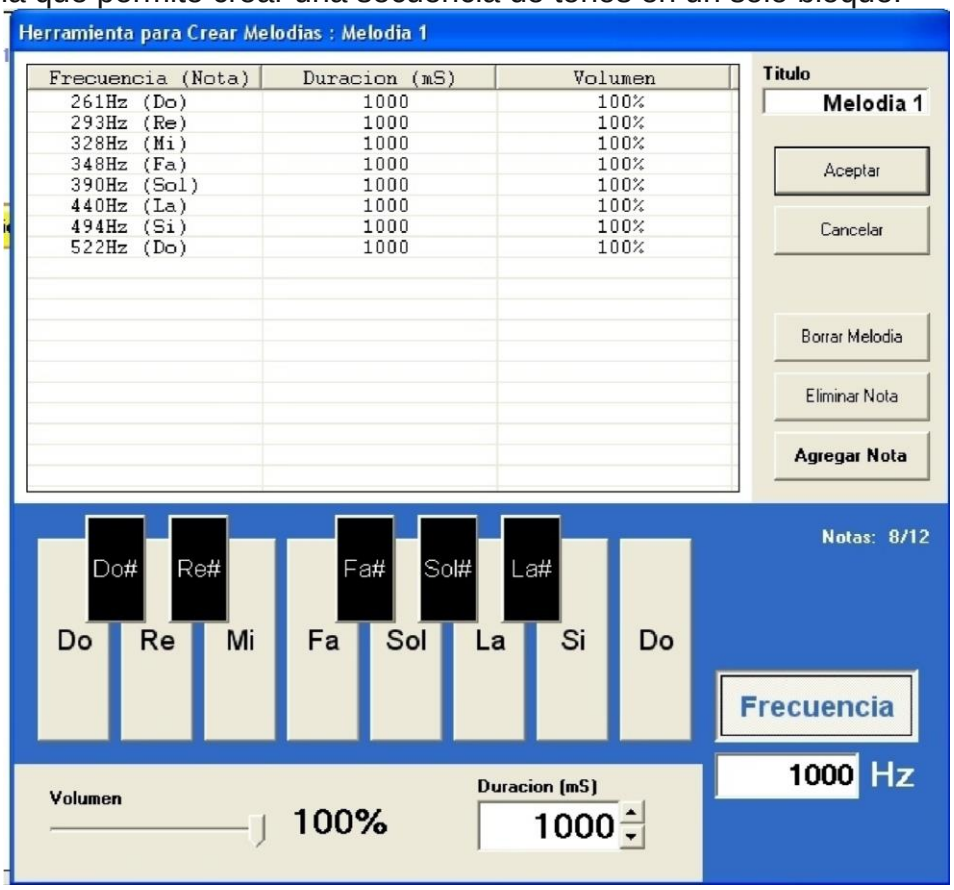
En el sistema de dirección tipo robot-escorpión los tiempos de giros pueden variar dependiendo del estado de las pilas y el tipo de piso sobre el cual se mueva el modelo, para ajustar estos tiempos y que los recorridos sean lo mas parecido a los dibujados puedes cambiar el tiempo de giro para 45° esto se hace en el menú desplegable “Configuración”, seleccionado la solapa motores y cambiando el valor indicado como “Tiempo de Giro de 45° para Dirección de Motores Independientes”.

Nota:
Para realizar recorridos es aconsejable usar el robot, ya que la tracción del escorpión suele tener movimientos menos precisos y los recorridos no se ejecutaran debidamente.

Sonido

Melodía

Al seleccionar el bloque de melodía (dentro de Salidas) se abrirá la siguiente pantalla que permite crear una secuencia de tonos en un solo bloque:



Las teclas de la parte inferior de la pantalla permiten seleccionar la nota que desea generar o bien fijar una frecuencia arbitraria, una vez elegido el tono presione “Agregar Nota” y en la lista aparecerán los datos de la nota agregada. En la zona inferior gris están los controles de dirección del tono y volumen. Si desea eliminar la ultima nota presione la tecla “Borrar Nota”.

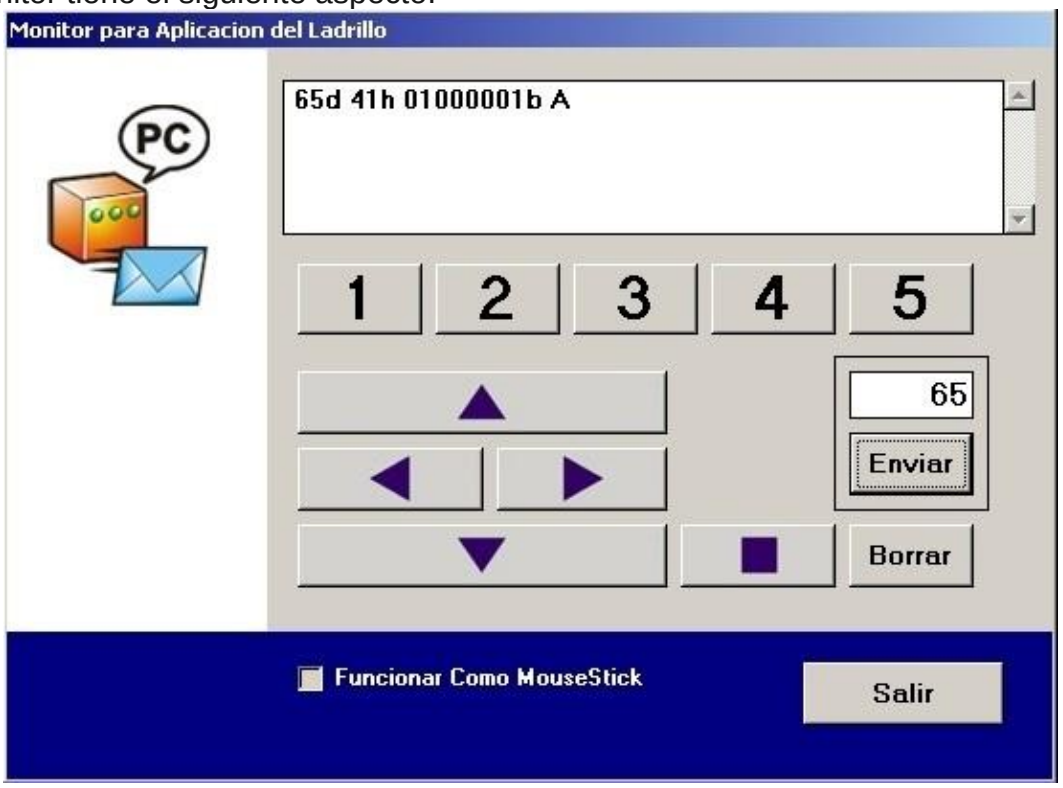
A la melodía se le puede asignar un nombre de modo que cuando se genere el bloque en el programa al poner el cursor sobre el, aparecerá como ayuda rápida dicho nombre. El nombre de la melodía se encuentra en la esquina superior derecha de la pantalla (Titulo).

Una vez terminada la melodía presione la tecla “Aceptar” y el nuevo bloque con la melodia se generara. Un bloque de melodia puede tener hasta 12 notas.

Programación Avanzada

¿Como Buscar errores en un programa?

En los programas mas elaborados pueden surgir errores de programación difíciles de resolver, para estos casos los programas pueden enviar mensajes a la PC para ver los valores de los contenedores e insertar indicadores para poder seguir el flujo del programa.
Para esto el Probot Lab tiene instrucciones de mensaje que funcionan en conjunto con la pantalla de monitor que muestra los valores enviados por el eBrick. La pantalla monitor tiene el siguiente aspecto:



En el programa “Medir Luz”, podemos ver como el ladrillo puede enviar un valor a la PC. Para ver dicho valor cargue y ejecute el programa en el ladrillo y luego presione la tecla “Monitor” de la barra de herramientas y en la zona de pantalla del monitor (área blanca) podra ver un valor de 1 a 15 que es proporcional a la cantidad de luz que llega al sensor de luz del eBrick.

Los bloques que envían mensajes pueden transmitir el valor de un contenedor en decimal, hexadecimal, binario o ASCII. La ubicación en la pantalla del monitor puede manejarse con las siguientes opciones:

Borrar Pantalla Antes (antes de enviar el contenedor)
Bajar Línea
Posicionar en Origen (ubica el valor en la esquina superior izquierda de la pantalla)
Mostrar sin Acción (pone el valor a continuación del anterior)

¿Como controlar un programa desde la PC?

Asi como es posible enviar mensaje del ladrillo a la PC como se explico en la sección anterior, es posible enviar datos de la PC al ladrillo y usar esos datos desde un programa.

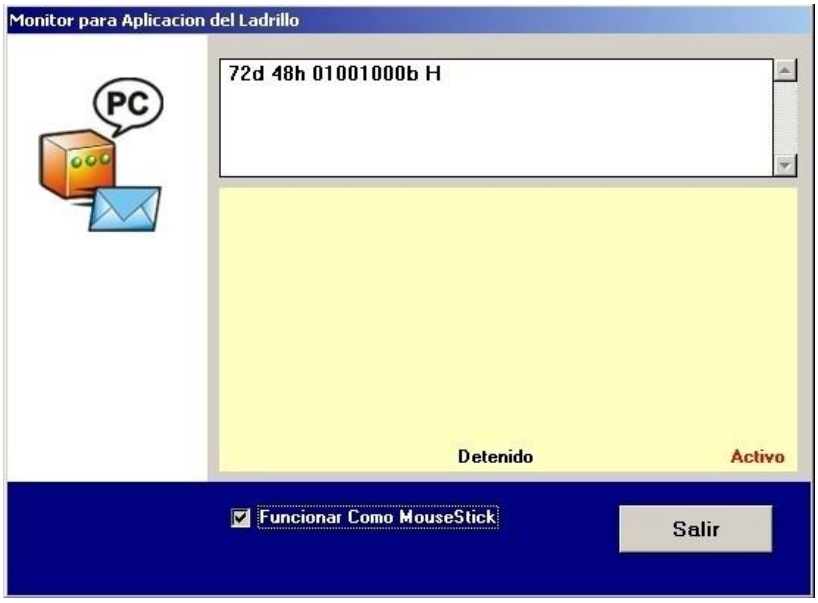
El programa “Controlar Motores PC” permite manejar los motores desde el monitor del Probot Lab. Cada tecla del monitor genera un código que se envía por la interfaz serial al ladrillo.
Los códigos de las teclas son las siguiente:

Tecla	Codigo
Tecla 1 a 5	01d a 05d
Arriba	34d
Izquierda	36d
Derecha	32d
Abajo	38d
Detener	40d
Enviar	El valor que se encuentra en la caja de texto ubicada arriba de la tecla (0d a 255d)

Los bloques de recepción de mensajes se quedan detenidos hasta que llega un mensaje de la PC, dicho mensaje (que siempre es un valor numérico entre 0 y 255) se almacena en el contenedor seleccionado.

Como usar el MouseStick

Seleccionado la opción “Funcionar como MouseStick” en la pantalla monitor las teclas son reemplazadas por un rectángulo amarillo que es sensible al movimiento del mouse que se ve así:



Si el cursor del mouse se mueve sobre el rectángulo el monitor ira enviando valores al ladrillo indicando el sentido del movimiento. Usando este recurso puede hacer programa que se controlen con solo desplazar el mouse.

Movimiento	Codigo
Adelante	34d
Adelante-Izquierda	35d
Izquierda	36d
Izquierda-Atras	37d
Atras	38d
Atras-Derecha	39d
Derecha	32d
Derecha-Adelante	33d
Detenido	40d

Manejo avanzado del Sensor de Luz

El sensor de luz del eBrick permite leer el nivel de brillo sobre el y muchos programas usan esta característica. Pero al cambiar las condiciones de iluminación el las que se ejecuta un programa que usa la luz como información para operar, pueden surgir problemas que hagan que el modelo realice acciones no esperadas. Para resolver problemas asociados al nivel de iluminación hay que conocer cuan iluminado esta el ambiente en el que el modelo va a funcionar, un modo de saberlo es midiendo el nivel de luz y luego modificar el programa para que use ese valor de luz, aunque es una solución posible no es la mas aconsejable ya que requiere modificar el programa dependiendo de donde se lo va a usa. De todas maneras siempre es bueno saber que valor devuelve el sensor de luz en un ambiente promedio para tener una referencia, esto lo puede hacer usando el programa “Medir Luz”:

Este programa envía el valor leído por el sensor de luz al monitor del Probot Lab.

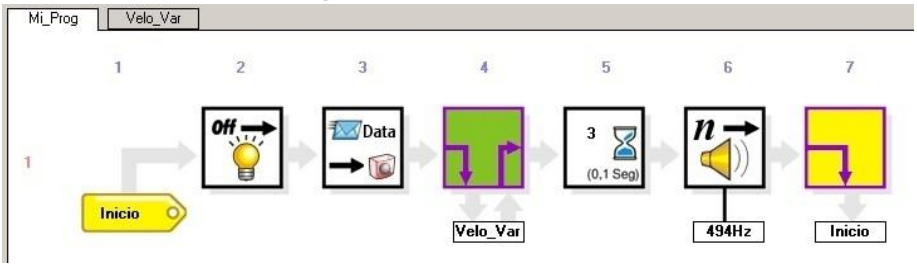
La solución mas aconsejable es usar un sistema de autocalibracion que haga que el ladrillo tome la luz ambiente al comenzar el programa de modo que si cambian las condición de iluminación solo habrá que correr el programa nuevamente para que se ajuste a las nuevas condiciones. Para hacer esto debe ir al menu desplegable “Configuración”, solapa “Sensores” y en la “Fuente de Referencia de estado Iluminado” seleccionar un contenedor (es decir no usar la opción “Constante”). El contenedor seleccionado será la referencia del nivel de luz que usara el programa. Luego debe hacer que su programa al iniciarse lea el nivel de luz y lo almacene en el mismo contenedor que selecciono en configuración. Este contenedor no debe ser usado para otra función ya que de hacerlo se perderá la información del nivel de luz inicial.

Una vez hecho esto todas las esperas y decisiones con luz (que tienen el ojo en el bloque) usaran como referencia el contenedor seleccionado.

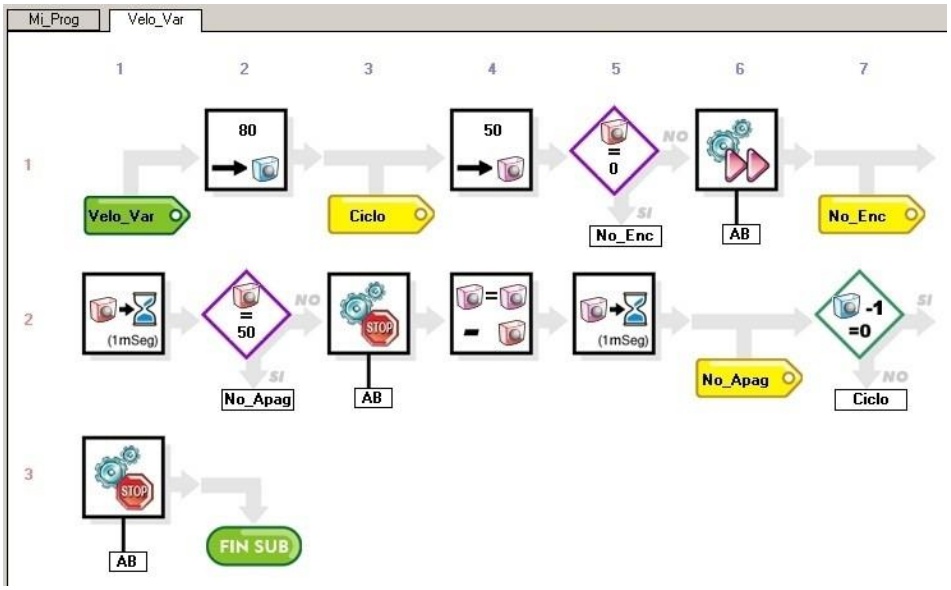
Manejo avanzado de Motores

El sistema de motores de eBrick no usa partes electromecánicas en su controlador lo cual permite controlar la velocidad desde los programas. La idea de variar la velocidad del motor es simple: se enciende y apaga el motor muy rápido, cuanto mayor es el tiempo que esta encendido, mas rápido funcionara, en cambio si el tiempo de apagado es mayor funcionara mas lentamente. El programa “Regulador de Velocidad” muestra como controlar la velocidad de ambos motores a la vez desde la PC:

Cuerpo principal de “Regulador de Velocidad”



Subrutina Velo_Var



Enviando desde el monitor de la PC un valor entre 0 y 50 los motores funcionaran durante unos 4 segundos con una velocidad proporcional al numero enviado. Para controlar al velocidad de los motores el eBrick queda en un lazo enviando ordenes a los motores a gran velocidad por lo que no puede hacer operaciones de esperas ni nada que lo detenga, de ser asi el proceso de velocidad variable fallaría. Lo que se puede hacer el leer sensores en el medio del proceso de control de velocidad u ejecutar otras acciones como encender o apagar leds, pero no se pueden emitir sonidos ni esperar mensajes de PC ya que estas operación detienen el programa.

Manejo Avanzado del Sonido

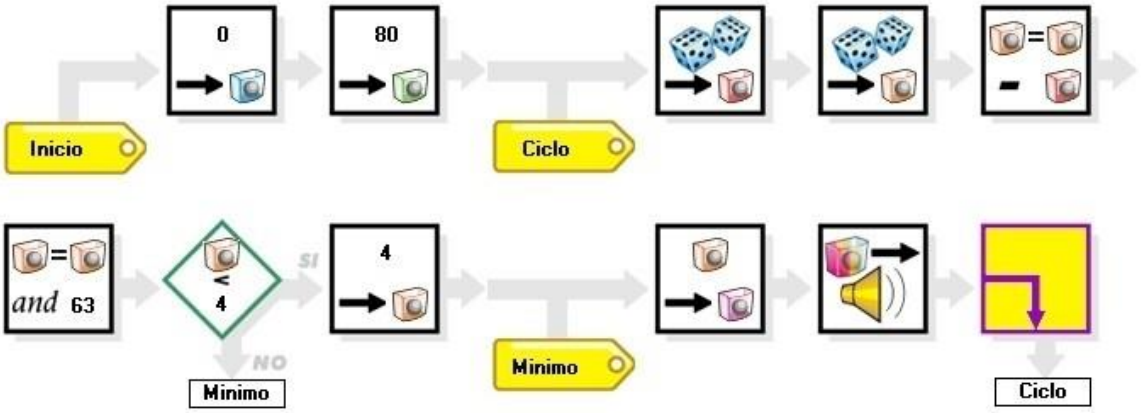
Hasta ahora el sonido se ha usado con tonos fijado por constantes, es decir tonos que siempre que se ejecutan producen el mismo sonido. Probot Lab tiene la opción de genera tonos definidos por contenedores, esto quiere decir que se pueden generar tonos variables, ya sea por una operación matemática, un sensor o un valor al azar.

Para generar sonidos se requieren de 4 contenedores, 2 para el tiempo de duración y 2 para el tono a generar.

Rosa y Naranja: Frecuencia “tono agudo o grave”= (1/ ((Rosa + Naranja) * 8uS “microsegundos”))
Los contenedores Rosa y naranja determinan las frecuencia (mas baja, mas grave. Mas alta, mas aguda)
Azul y Verde: Duración del tono en tiempo=(((Azul*256)+Verde)*(Rosa+Naranja)* 8uS “microsegundos”)
Los contenedores Azul y Verde el tiempo que dura el sonido.

Para simplificar los cálculo al crear un bloque de sonido la pantalla calcula automáticamente los valores para el tono pedido.

En la opción de ejemplo de programación rápida hay 3 ejemplos de manejo de sonido por variables. En los 3 ejemplos la duración del tono es fija (por simplicidad) y luego se entra en bucle de repetición donde las variables de frecuencia se modifican de algún modo creando tonos que van variando en cada repetición. Las operación matemáticas con el contenedor naranja limitan y adaptan el valor para que los tonos generados estén dentro de cierto rango de sonido. Aquí se puede ver el programa “tonos aleatorios” que genera sonidos tomando la frecuencia de un valor al azar.



En el ejemplo anterior se usan 2 generadores al azar para bajar la posibilidad de que los tonos se repitan y la función “AND” se usa para limitar el valor del generador aleatorio.

Notese que cuanto más grande es el valor de los contenedores Rosa-Naranja mássche i baja es la frecuencia producida (más grave es el sonido) y que la duración del tono también es proporcional a la frecuencia, para un mismo valor de los contenedores Azul-Verde.

Para que la función de sonido se ejecute correctamente nunca debe cargar con 0 los contendores Rosa-Naranja.