

COMPONENTES LEGO

Antes de comenzar a construir un robot, es importante familiarizarse con los distintos componentes disponibles en el kit MindStorms de LEGO.

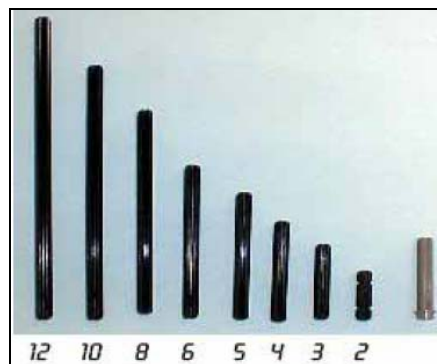
I.- Piezas

❖ Placas, Palancas y Bloques



Son los componentes estructurales. Las barras vienen en una variedad de largos y tienen agujeros para insertar ejes. Las placas son piezas planas que también vienen en una variedad de largos y anchos.

❖ Ejes



Los ejes se usan para colocar engranajes o ruedas. La longitud de los ejes varía entre 2 y 12 “unidades LEGO” de longitud.

❖ Golillas



Las golillas se colocan en los ejes para mantenerlos en su lugar. Las semi-golillas, también pueden ser usadas como pequeñas poleas ya que poseen una mayor fricción que la golilla completa y poseen una mejor sujeción.

❖ Conectores



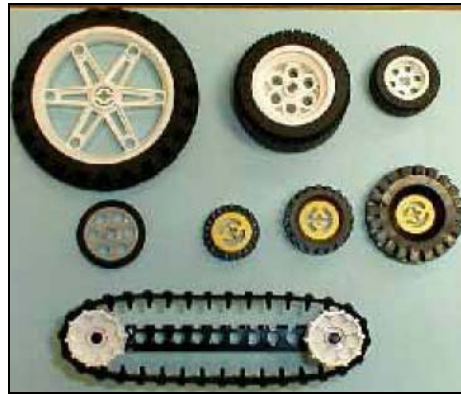
Los Conectores de Fricción permiten conectar barras; se insertan en los agujeros de las barras a conectar.

❖ Perfiles y Codos



Hay una variedad de perfiles y codos de soporte. Su función es permitir la interconexión de ejes y conectores de fricción.

❖ Ruedas



El kit contiene una variedad de ruedas de distinto tamaño, muy útil para lograr el desplazamiento del robot, en distintos tipos de terreno.

❖ Engranajes



Los engranes, son los que transforman el movimiento del motor en desplazamiento de del robot. Los engranes van directamente relacionados con el torque que ejerce las ruedas.

❖ Correas y poleas

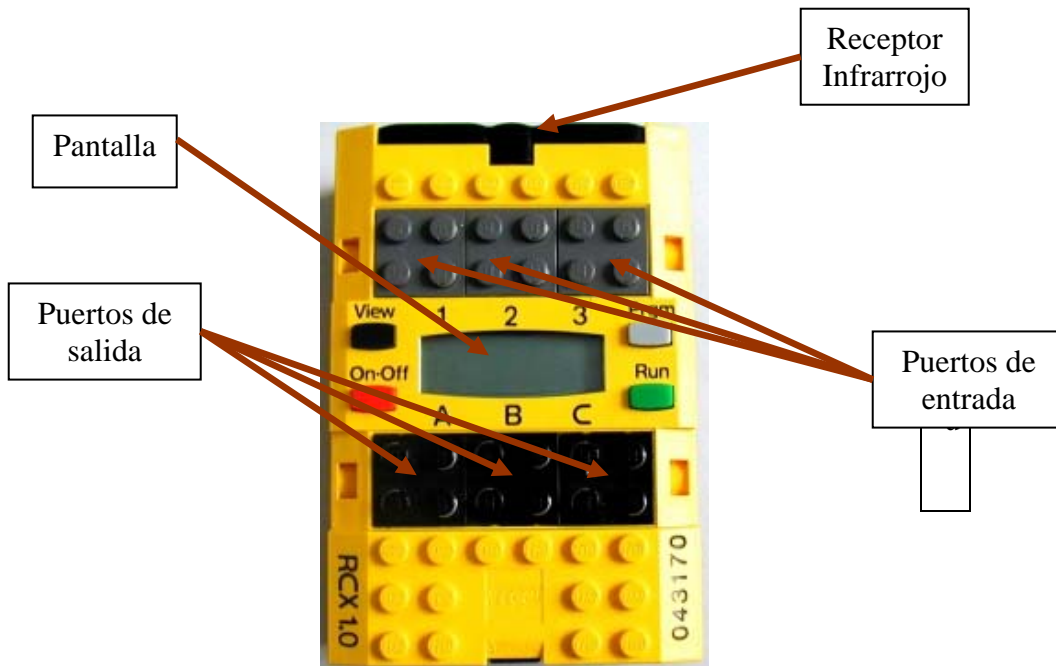


Las correas y poleas son otro medio de transferir movimiento rotacional.

II.- RCX



El **RCX** es el cerebro del robot. Almacena los programas computacionales, lee los sensores con la información de entrada y controla el movimiento de los motores.



III.- Motor



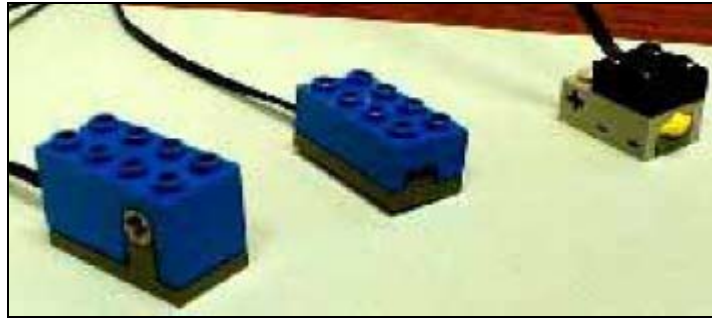
Los motores son los que generan el movimiento del robot. Transforma la electricidad provista por las baterías en el RCX, en movimiento rotacional.

❖ Conexión

Los motores se conectan a los puertos de salida A, B o C en el RCX. Es muy importante la forma en la cual se conectan los cables. La conexión determinará el sentido de la rotación. Si damos vuelta los cables, el motor girará en sentido contrario.



IV.- Sensores

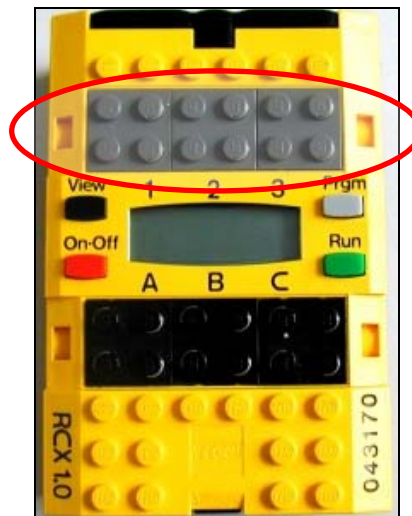


El kit incluye tres tipos de sensores:

- *Sensor rotacional*: Mide la rotación de un eje. El sensor mide 16 pasos, por cada vuelta del eje
- *Sensor de luz*: Reacciona según los cambios de luz. Devuelve un valor entre 0 (oscuro) y 100 claro
- *Sensor de tacto*: Reacciona al contacto entregando un valor binario, 1 apretado y 0 suelto

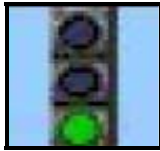
❖ Conexión

Los sensores se conectan en los puertos de entrada 1, 2 o 3 del RCX, como lo muestra la figura. Es muy importante chequear su funcionamiento antes de ser usado.



V.- ICONOS DE ROBOLAB

❖ Inicio del Programa



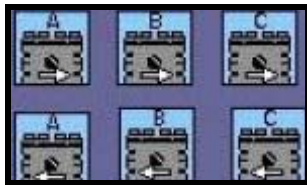
Este icono debe estar siempre presente al inicio de todos sus programas.

❖ Fin del Programa



Cualquier tarea dentro de un programa (tareas en un programa □ subrutinas)
Este icono debe estar al final de todos sus programas y al final de

❖ Encendido



Estos iconos encienden los motores y les indican que se muevan adelante o atrás (avanzar o retroceder).

❖ Nivel de Potencia



Estos iconos definen el nivel de potencia de un motor (de 1 a 5, de mínimo a máximo).

❖ Esperar por



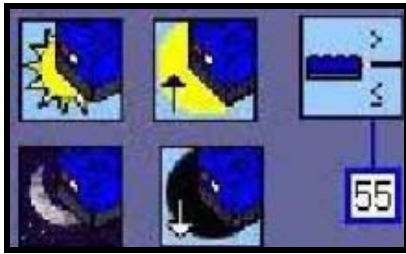
Estos iconos permiten apagar los motores.

❖ Sensor de contacto



Estos iconos permiten agregar un sensor de contacto al robot y administrarlo con el programa

❖ Sensor de Luz



Estos iconos permiten agregar un sensor de luz al robot y administrarlo con el programa.

❖ Sensor de Rotación



Estos iconos permiten agregar un sensor de rotación al robot y administrarlo con el programa.

❖ Relojes



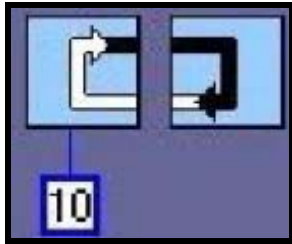
Estos iconos permiten controlar el tiempo de ejecución de tareas del programa.

❖ Contenedores



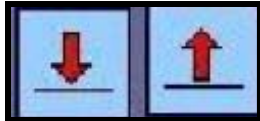
Estos iconos permiten el uso de variables en el programa. Las variables son números cuyo valor no sabemos sino hasta que el programa está siendo ejecutado (por ejemplo, un nivel de iluminación).

❖ Lazo FOR



Estos iconos permiten repetir el programa, o un trozo del mismo, por un cierto número de veces.

❖ **Salto**



Estos iconos permiten saltar desde un punto a otro del programa.

❖ **Multi-Tarea**



Permite dar inicio a la ejecución de dos tareas simultáneas en el programa. Cada tarea debe terminar con un signo ALTO (semáforo rojo). El símbolo inverso se denomina MERGE (fusión) y permite conectar dos tareas.

VI.- Programación

1ra Actividad.

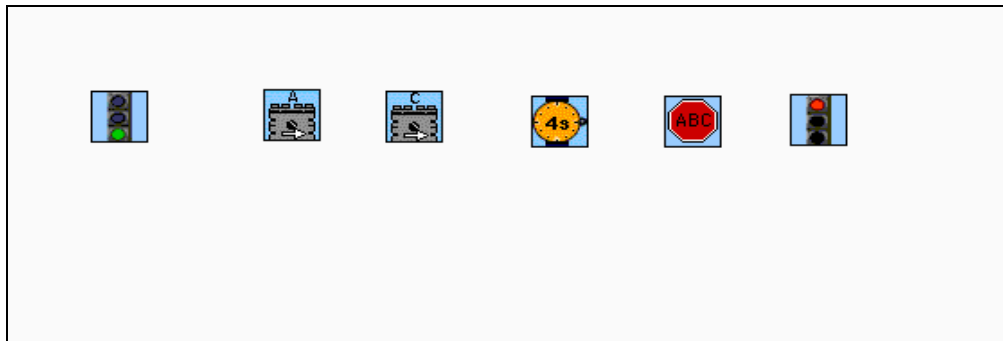
Tarea: Avanzar durante 4 segundos.

Realización:

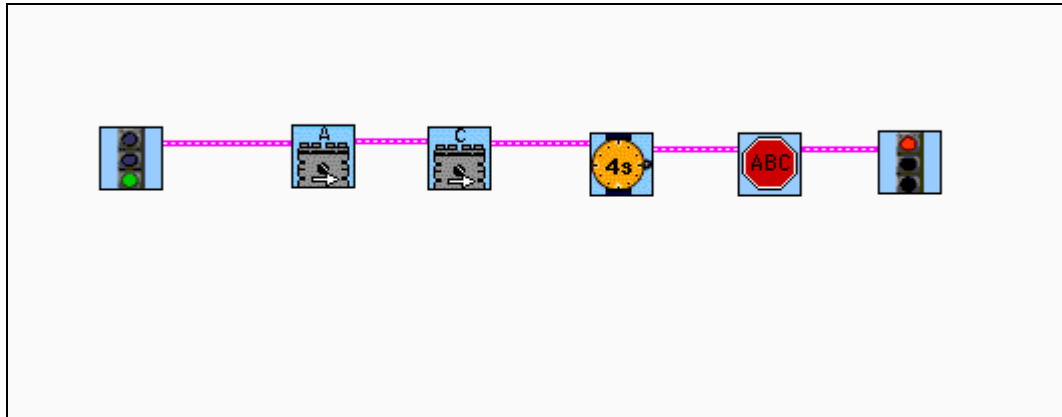
1. Arrastrar los iconos correspondientes a cada motor.



2. Arrastrar icono de esperar por y detenga los motores



3. Unir iconos por medio de la herramienta Wire o Carrete

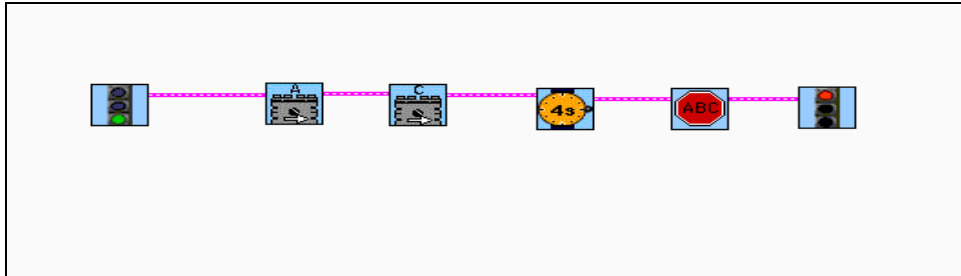


Actividad 2

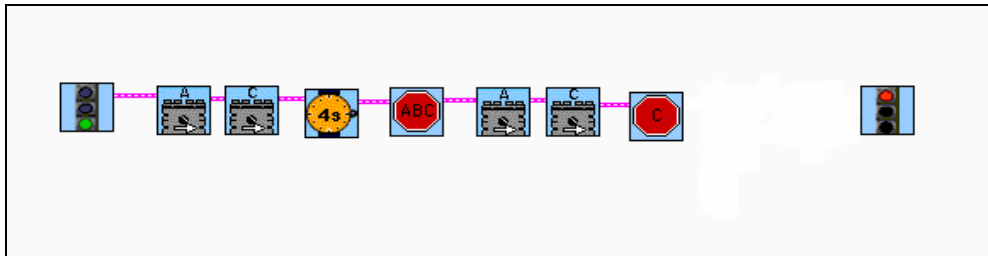
Tarea: avanzar durante 4 segundos y realizar un giro por 3 segundos a la derecha.

Realización.

1. Usaremos como base el programa hecho anteriormente



2. Acción de giro

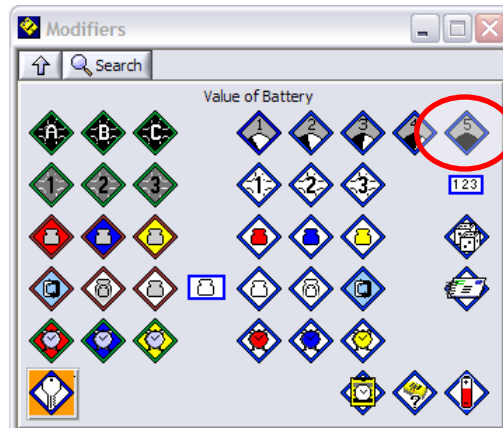


3. Al no existir el icono esperar por 3 segundos, es necesario crearlo, y se hace de la siguiente forma.

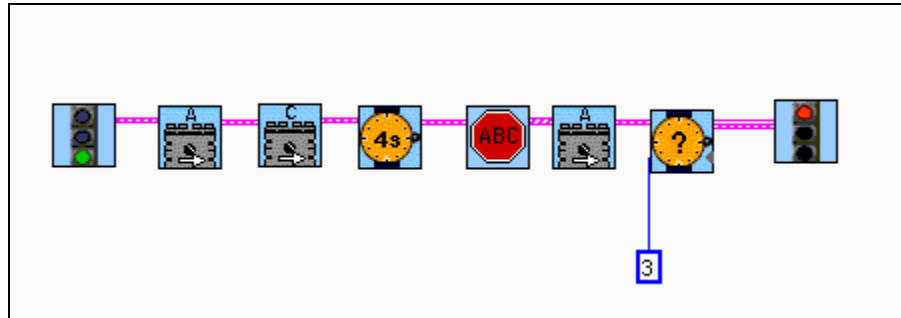
En esperar por, se selecciona el icono de esperar por tiempo.



Luego, por medio del modificador “constante numérica” se le introduce la cantidad de tiempo a esperar, en este caso 3 segundos.

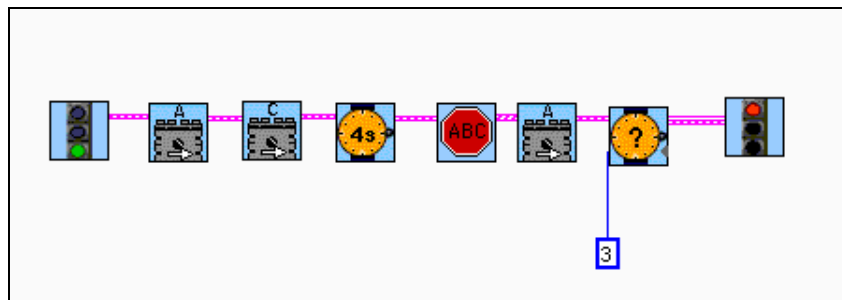


4. Uniendo todos los iconos, el programa queda de la séte forma:



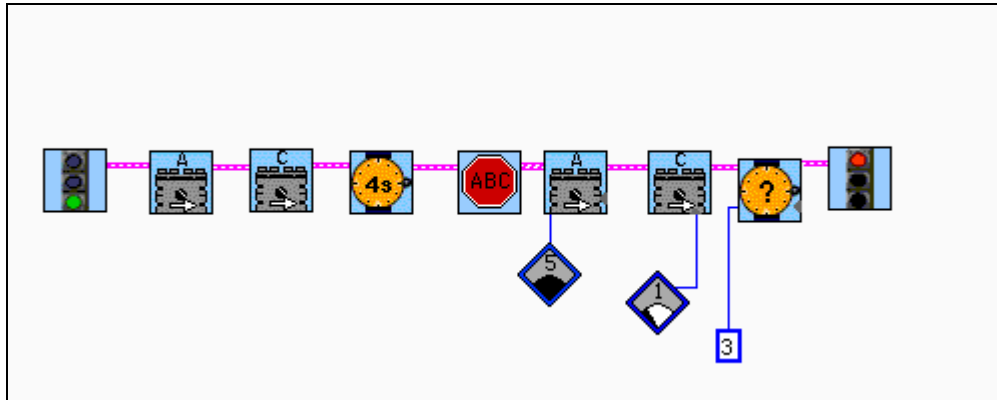
Observación: Para hacer girar un móvil, el cual posee ejes fijos (contrario a un auto, que puede direccional sus ruedas hacia donde desea dirigir), existen 3 formas las cuales se diferencian en el radio de giro.

- **Inmovilizar uno de los motores:** este es el caso del ejemplo recién desarrollado. Al estar ausente el motor C, es como si estuviera detenido, por lo tanto se desarrolla un movimiento de giro hacia la derecha. Al girar de esta manera, el móvil tiene un *radio de giro medio*.

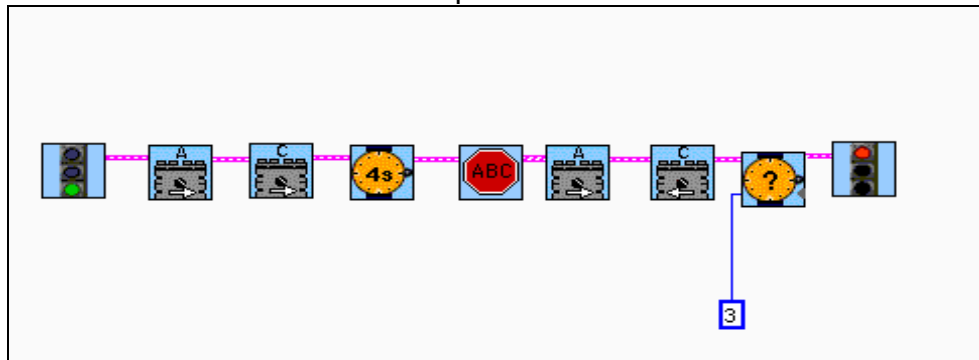


- **Cambiar la potencia de los motores:** A cada motor se le designa una potencia distinta que va desde la potencia 5 a la 1, la potencia de los motores se encuentra en modificadores, siendo esta última la más lenta. ejemplo, al motor A con potencia 5 y motor C con potencia 1, de esta forma el móvil girará hacia la derecha. Al girar de esta manera, el móvil tiene un *radio de giro largo*

Observación: Al no estar presente el icono de potencia, el programa va a asumir que la potencia del motor es la máxima, ósea 5



- **Motores en direcciones opuestas:** Esta es la forma más óptima de hacer girar un móvil con ejes fijos, ya que de esta manera se tiene un *radio de giro menor*, un ejemplo claro de la utilización de este mecanismo para girar son todas las máquinas orugas. Para programar esta forma de girar, basta con colocar los motores en direcciones opuestas.



Sensores

Los sensores son un componente crucial a cualquier máquina que se denomine un robot. Los sensores entregan información sobre el robot y el ambiente en el cual está interactuando, al computador (cerebro) del robot. El programa computacional del robot decide que hacer basándose en esa información y en sus propias instrucciones de tareas de alto nivel. En el kit MindStorms de LEGO posee 3 tipos de sensores:

- Sensor de Tacto
- Sensor de luz
- Sensor rotacional

❖ Sensor de Tacto

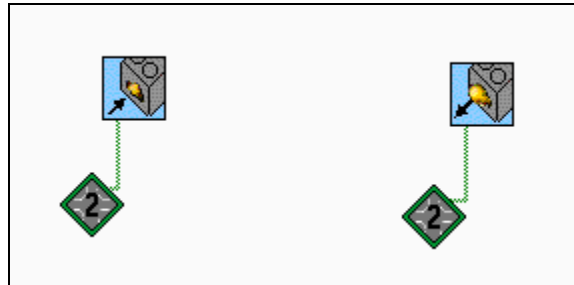
El sensor de contacto es un sensor que detecta el contacto con objetos en el ambiente del robot. El sensor de contacto le avisa al RCX si el botón está apretado o no.



El sensor de contacto actúa como un interruptor normal. Al apretar el botón, éste cierra un circuito eléctrico y permite el flujo de corriente a través del sensor. EL RCX detecta este flujo de corriente y así se entera que el botón ha sido presionado. Al soltar el botón, el circuito se abre y cesa el flujo de corriente. La imagen a la izquierda muestra la forma correcta de conectar el cable al sensor de contacto.

❖ Programación

Los sensores de contacto son programados para detectar cuando los botones son presionados o liberados y de esta forma determinarle una tarea al robot. Usted necesita especificar a que puerto está conectado un sensor. El icono de la figura a la izquierda, indica al programa que debe esperar hasta que el sensor de contacto conectado al puerto 1 sea *presionado*. Mientras que la figura de la derecha espera hasta que el sensor de tacto sea *liberado*

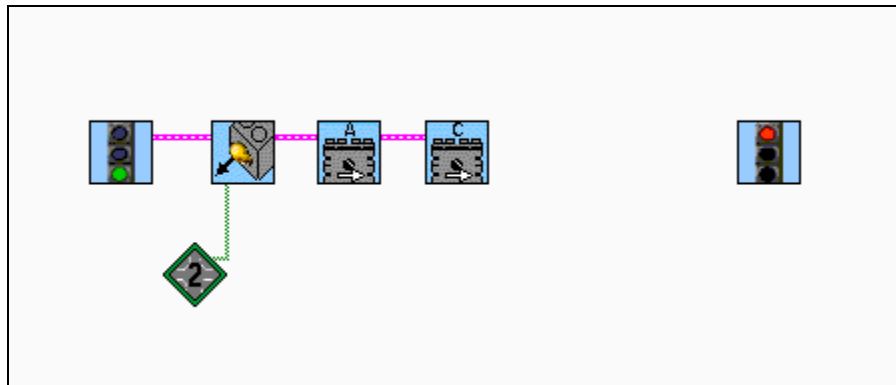


Actividad 1

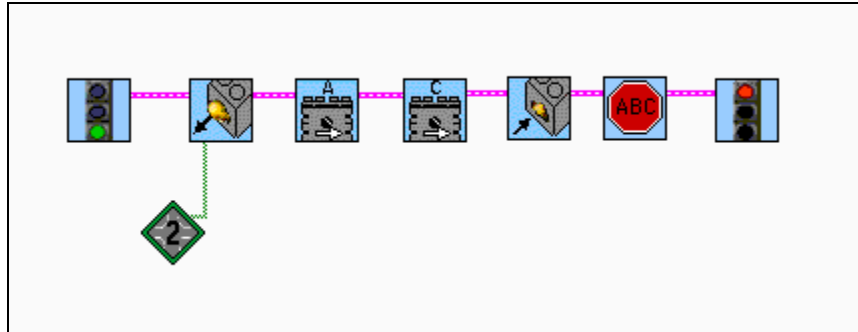
Tarea: Avanzar solo cuando el sensor de tacto se encuentre liberado

Realización

1. Tarea a realizar cuando el botón este liberado

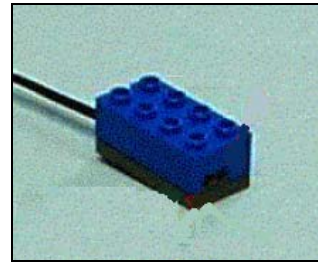


2. Adicionar la tarea cuando el botón este presionado



Sensor de luz

Un sensor de luz es un sensor que mide la cantidad de luz que recibe. Le entrega al RCX un número que varía entre 0 (oscuridad total) y 100 (muy brillante).



El sensor de luz tiene una fuente de luz propia, un *Diodo Emisor de Luz (LED)* rojo que ilumina una pequeña área al frente del receptor.

El sensor de luz puede determinar si esta viendo un trozo de papel blanco o negro. Cuando el sensor de luz está sobre papel blanco, lee un valor de 50. Cuando está sobre el papel Negro, mide un valor de 33 (valores aproximados).

❖ Programación

Los sensores de luz pueden ser programados para detectar un nivel de luz específico entre 0 (oscuro) y 100 (iluminado).

- Este icono le dice al programa que debe esperar hasta que el detector de luz detecte un nivel de luz por encima de un determinado umbral, en este caso, 50.
- Este icono le dice al programa que debe esperar hasta que el sensor de



luz halla detectado un nivel de luz por debajo de umbral que, en este caso, es de 50.



Los sensores de luz también pueden ser programados en un modo relativo.

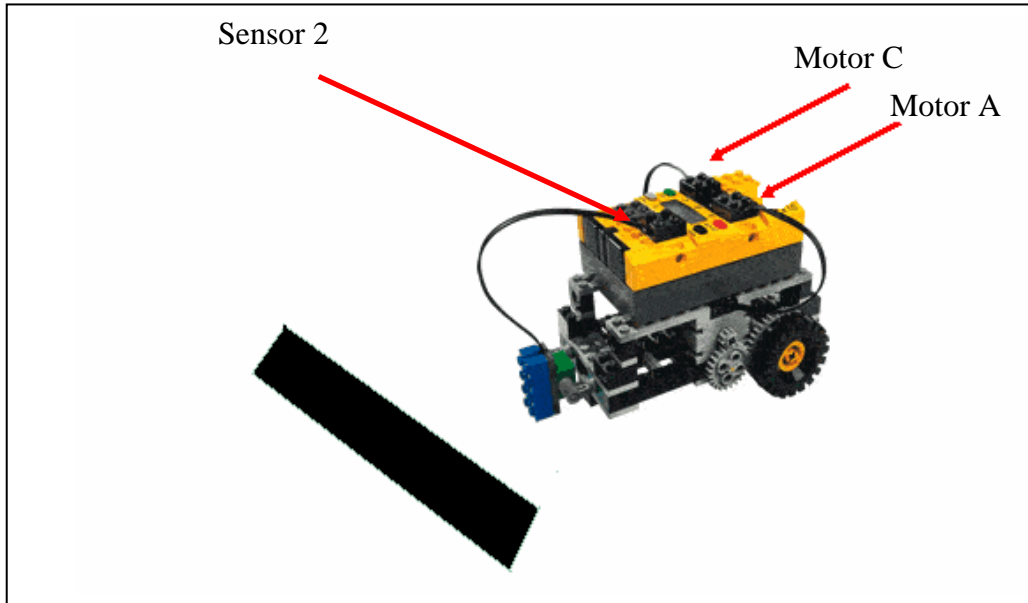
- Este icono le dice al programa que espere hasta que el nivel de luz sea un 5% superior al nivel actual de luz.



- Este icono le dice al programa que espere hasta que el nivel de luz sea un 5% menor que el nivel actual.



Actividad

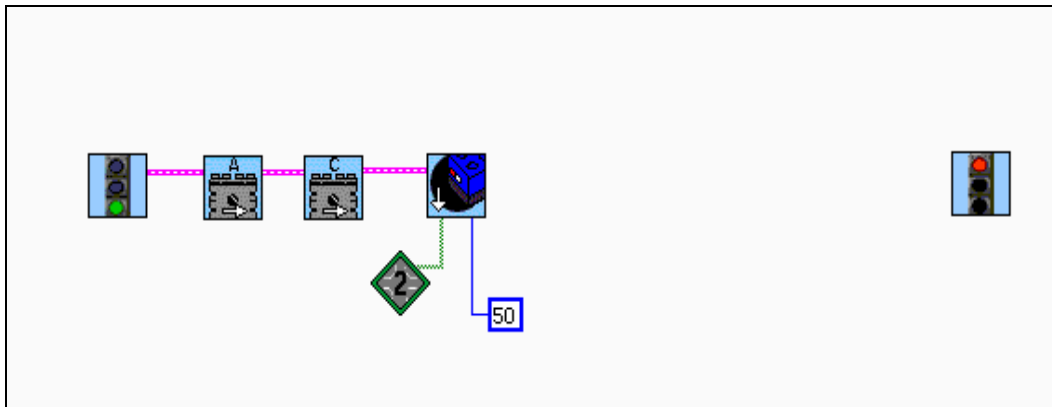


Tarea:
Que el Robo

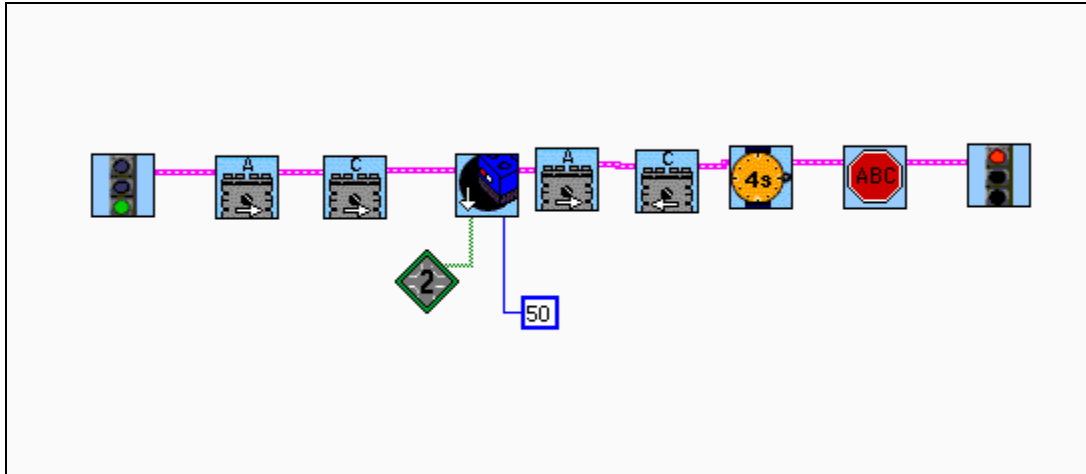
t avance hasta llegar a la línea negra y luego gire por 4 segundos a la derecha.

Realización.

1. Tarea a realizar hasta que el robot llegue a la línea negra, sabiendo con anterioridad que para el robot el color negro es 42.



2. Adicionar la tarea a realizar una vez que se encuentre con la línea negra



Observación: Un error muy común, a la hora de programar, es no determinar el puerto en el cual se encuentra el sensor

Sensor de rotación

El sensor de rotación es un sensor que es capaz de medir la rotación de un eje. Cuando el eje gira el sensor cuenta cuanto gira el eje. Cuenta 16 pasos por cada revolución completa del eje. El número de pasos o “clicks” es lo que el RCX lee como entrada.



El sensor de rotación es muy útil para medir y controlar cuán lejos el robot se mueve. Midiendo cuanto giran las ruedas y cual es la circunferencia de las ruedas, pueden calcular cuanto ha recorrido el robot

Actividad

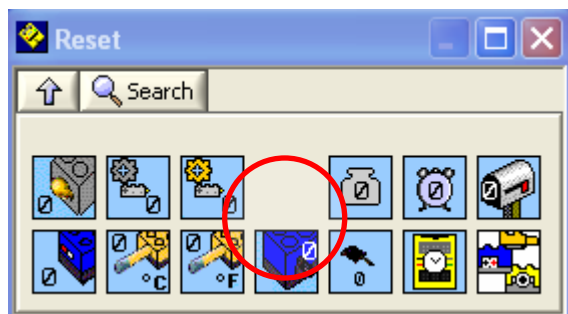
Tarea: EL motor A de 5 Vueltas

Realización

1. Es importante la puesta a cero, “RESET”, del sensor al comienzo del programa, para eso hay que ir a “reset”



2. Una vez dentro de reset, dirigirse a llevar a 0 el “sensor de rotación”



3. Realizamos la programación, sabiendo que el sensor cuenta clicks, y cada vuelta tiene 16 click, por lo tanto 5 vueltas son 80 click's

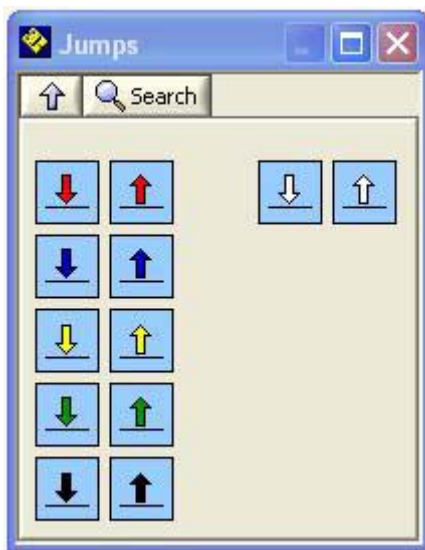


Una de las conductas más básicas de un robot, es que realice la repetición de una acción indefinidamente, para esto utilizaremos el comando Salto, también conocido entre los programadores como Go To.

Cuando el programa encuentra una flecha hacia arriba “salta” a la flecha hacia abajo y continúa la operación del programa desde allí.

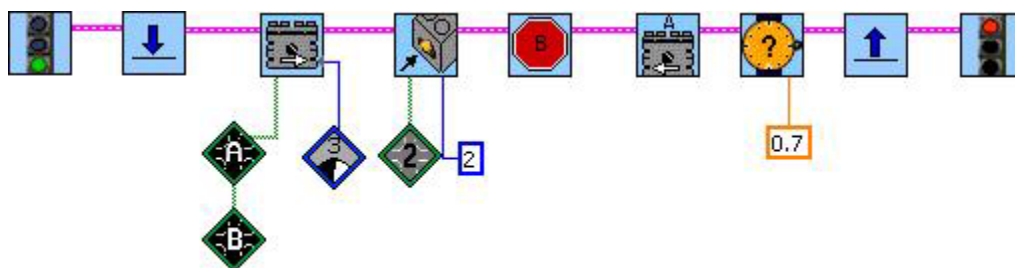
Es importante señalar que cuando el programa encuentra una flecha hacia abajo, no realiza ninguna acción.

Existen en RoboLab, cinco pares de flechas para realizar esta acción. Si es necesario en un programa mas de cinco saltos debemos utilizar las flechas blancas, con su correspondiente modificador numérico.



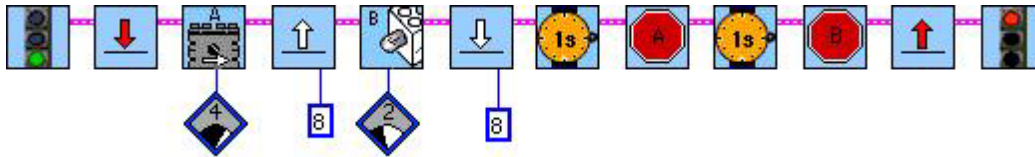
Actividad

El siguiente programa se repite indefinidamente ya que cada vez que el programa se encuentra con la flecha hacia arriba, salta al comienzo y así sucesivamente.



Tarea: ¿Qué significa el número 2, en el sensor de tacto que está conectado a la puerta de entrada 2?

Realización: En el siguiente programa determine en que momento se enciende la lámpara conectada en la puerta B y por cuanto tiempo.



Bifurcaciones (IF's) y Saltos(Jumps)

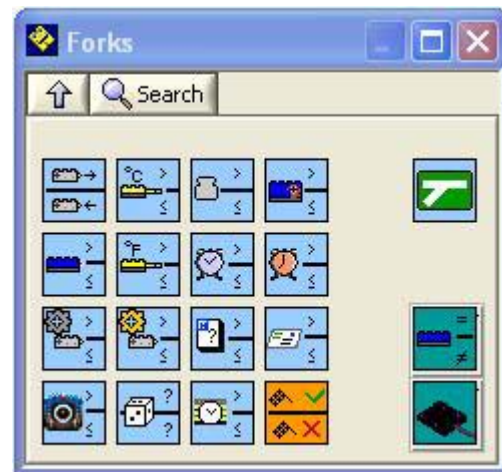
Las bifurcaciones se utilizan para evaluar una condición dentro del programa, y así poder realizar las acciones necesarias, de acuerdo a esa condición.

Las condiciones que pueden ser evaluadas son variadas, por ejemplo:

- Sensor de tacto
- Temperatura
- Reloj
- Correo
- Azar

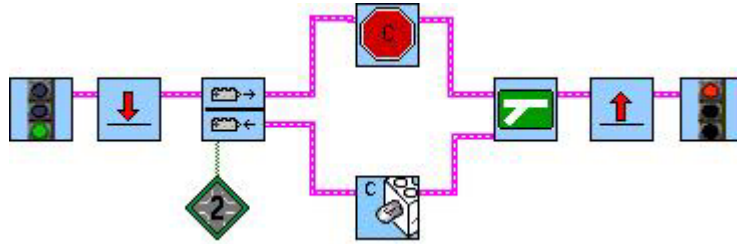


Es importante que al final de cada bifurcación, se encuentre el símbolo de cierre de esta.



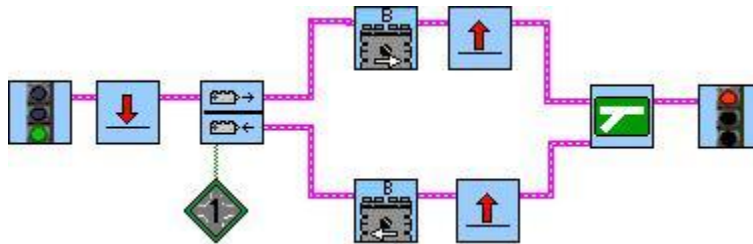
Actividad

Tarea: El verdadero poder de los saltos es en combinación con las bifurcaciones, veamos un ejemplo:



Realización: Cuando el sensor de tacto de la puerta 2, se active se encenderá la lámpara de la puerta C y se apagará cuando se desactive.

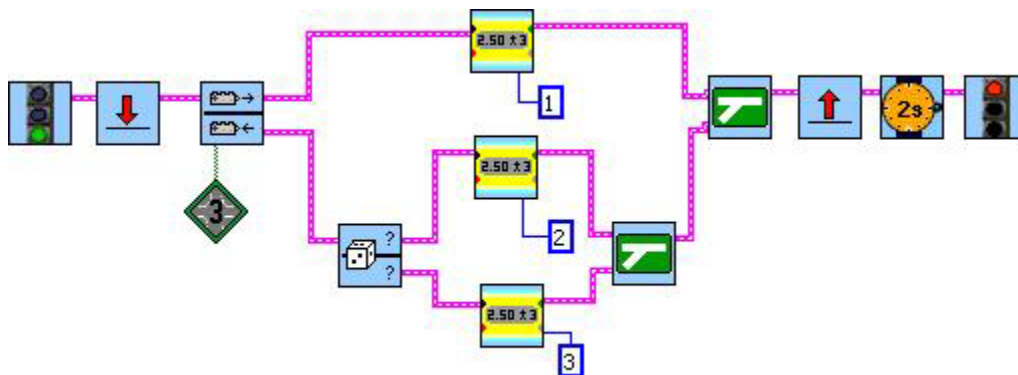
Observación: También se pueden combinar más de un salto del mismo color, pero sólo deberá existir un aterrizaje (flecha hacia abajo)



Tarea: ¿Que hace este robot?

Realización: Múltiples bifurcaciones también se permiten, pero teniendo cuidado de cerrarlos sin que se crucen.

En el siguiente ejemplo se muestra en pantalla un número que corresponde al camino que ha tomado el programa al evaluar una condición en cada bifurcación



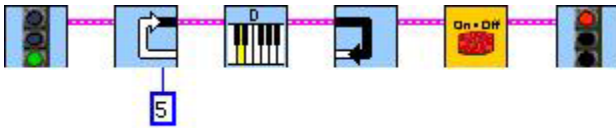
Observación: Utilice la opción de ayuda de RoboLab, para investigar que función realiza el dado en la segunda bifurcación.

Bucles(Loops)



Los bucles permiten efectuar iteraciones(repeticiones) de una sección de un programa, hasta que se satisfaga la condición esperada.

La estructura básica de un bucle o lazo es la siguiente:



En este caso la sección del programa donde se encuentra la nota musical, se repite cinco veces luego se ejecuta la siguiente instrucción, la cual apaga el RCX.

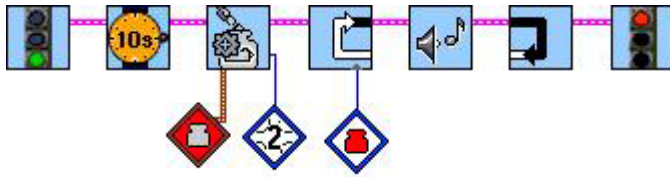
Es importante que cada bucle sea cerrado mediante el siguiente icono.



El sub-menú de bucles, tiene una gran variedad de condiciones que se pueden evaluar para determinar si el ciclo de repetición continúa o ha sido satisfecho.

Podemos evaluar:

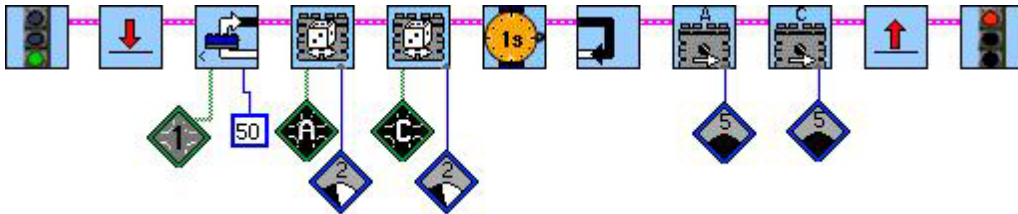
- Sensores
- Cointenedores
- Reloj
- Correo
- etc.



Observación: Este programa almacena en el contenedor rojo la cantidad de veces que presionamos el sensor de tacto, conectado en la puerta dos, luego utiliza ese valor para emitir la cantidad de “beeps” y termina. ¡Inténtelo!

Actividad

Tarea: Determine el comportamiento del robot:



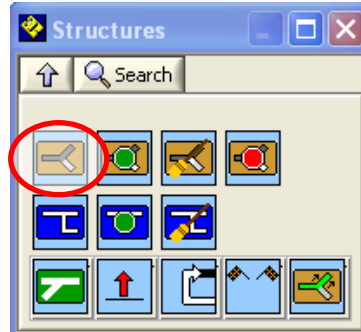
Tarea: El sensor de luz está colocado sobre el robot, y este se encuentra en una pieza oscura o en penumbra, luego encendemos la luz.

¿Qué hace el robot en la oscuridad?

¿Qué pasa cuando se enciende la luz y cuando se vuelve a apagar?

Multitarea

El RCX tiene la habilidad de ejecutar más de una tarea a la vez, esta capacidad se llama Multitarea (Multitask), para este fin utilizamos el “Divisor de Tareas”

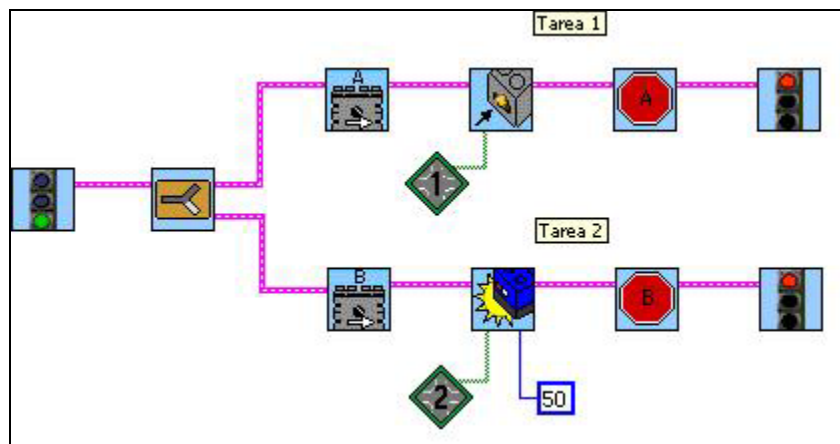


Este comando permite básicamente ejecutar dos tareas al mismo tiempo, lo cual es muy útil cuando queremos monitorear dos sensores al mismo tiempo, el siguiente programa muestra dos tareas independientes para controlar los motores A y B, uno con el sensor de tacto y otro con el sensor de luz

Actividad

Tarea: Realizar 2 tareas independientes para controlar los motores A y B, uno con el sensor de tacto y otro con el sensor de luz.

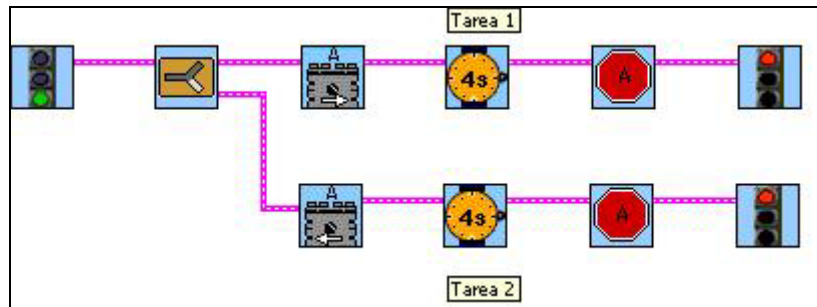
Realización:



Observación: Debido a que las tareas son independientes, aún cuando un motor se detenga y termine su tarea, el otro seguirá girando.

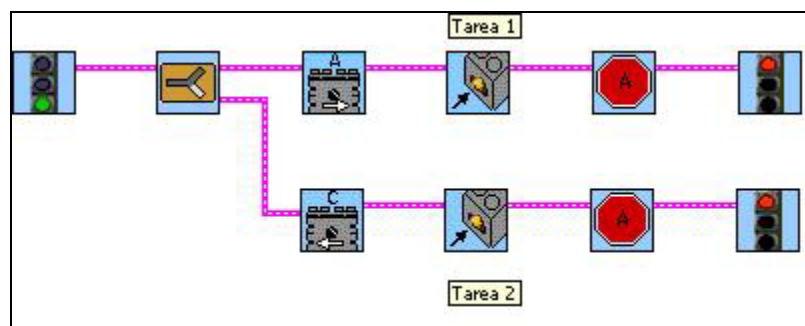
Conflictos

La habilidad de ejecutar programas independientes es un arma de doble filo, dado que los programas comparten los mismos dispositivos. Si una tarea ejecuta el comando “Motor A hacia adelante” y otra tarea ejecuta el comando “Motor A hacia atrás”, al mismo tiempo, es difícil determinar que ocurrirá. La resolución de conflictos (ver figura)

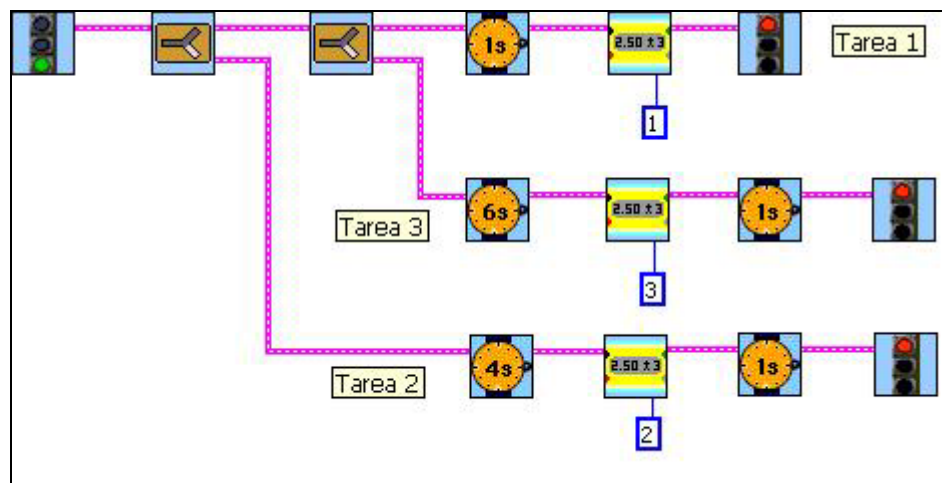


Mientras los dispositivos de salida (motores y lámparas), pueden causar conflictos, no hay problema en compartir recursos de entrada en diferentes tareas.

En este programa, las dos tareas utilizan sin problemas en sensor de tacto, conectado en el puerto de entrada 1, para detener los motores A y C.



En el siguiente programa se muestra como se asigna la numeración de las tareas.

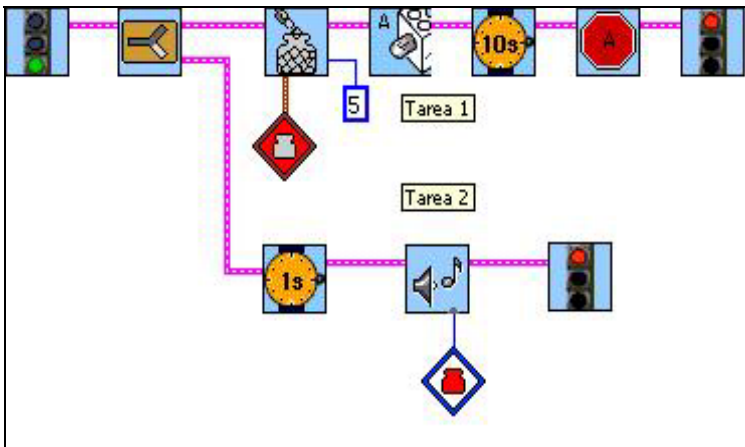


Actividad

Analice que mostrará el panel, en cada segundo, luego prográmelo y compruebe su predicción.

Multitarea con contenedores

Así como se pueden compartir los sensores entre las diferentes tareas, también se puede acceder a los contenedores sin problemas.

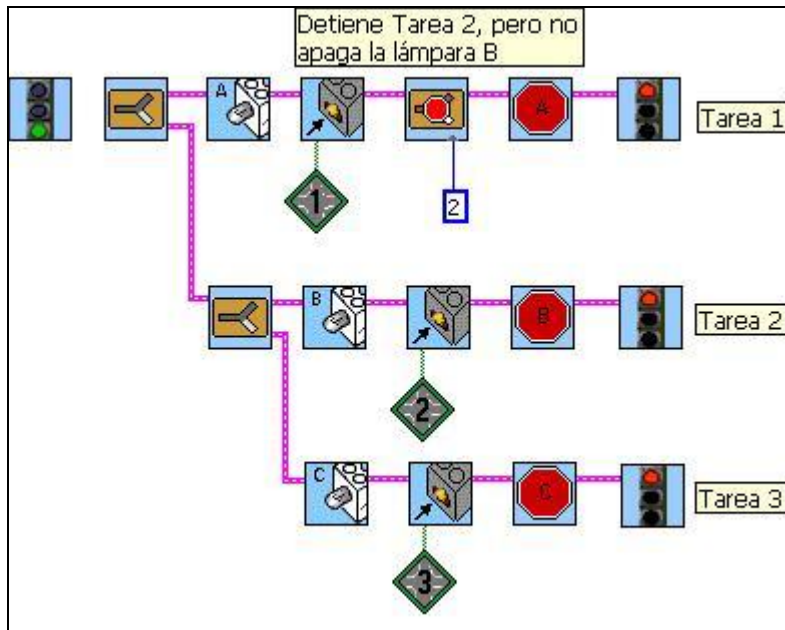


Observación: Durante la ejecución de la tarea 1, se llena el contenedor con el valor 5, y se usa en la tarea 2, para determinar el tipo de sonido.

Los contenedores son variables globales, por lo que se pueden usar en cualquier lugar independiente de las tareas.

Importante

- No se debe mezclar tareas, sólo se pueden mezclar bifurcaciones.
 - La División de tareas es básicamente la iniciación de un nuevo programa
 - Cada tarea debe tener al final su semáforo rojo
 - Evite conflictos, entre dispositivos de salida.
 - No salte entre tareas
 - No sobrepase la división de tareas en un salto.
-



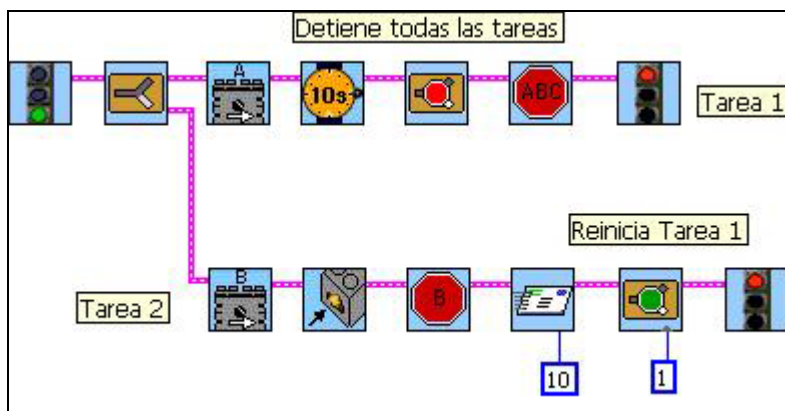
En el programa de Detención de Tareas, se especifica qué tarea detener, específicamente la tarea 2.

Actividad

Investigue, qué pasa con cada tarea en todos los casos.



El comando Comenzar Tareas, reinicia la tarea especificada.



En este programa, la tarea 1 detendrá todas las tareas si trascurren 10 segundos. Sin embargo si el sensor de tacto es presionado antes que se cumplan los 10 segundos, se detendrá el motor B, enviará el valor 10 por el puerto infrarojo y luego reiniciará ambas tareas, desde el comando División de Tareas.

Observación: El programa seguirá en ejecución, si presionamos el sensor de tacto, al menos cada 10 segundos.

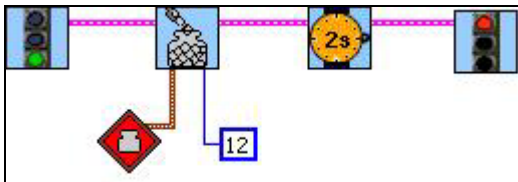
Contenedores



Los Contenedores son lo que los programadores de computadores llaman variables, estos permiten crear programas más flexibles, que los que se pueden desarrollar usando constantes (Modificador Constante numérica).

Principios Básicos

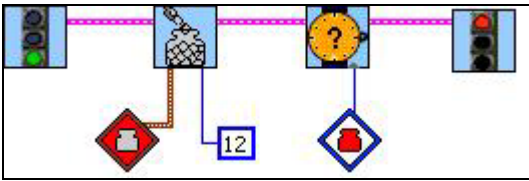
Existen tres contenedores básicos, rojo, azul y amarillo. Los contenedores sólo almacenan números enteros, es decir no almacenan números con punto flotante, el programa redondea hacia abajo cualquier resultado con decimales, es decir 1.9 se almacena como 1.



Observación: Este programa, almacena el número 12 en el contenedor rojo.

Para poder usar los contenedores en forma más útil, introduciremos un nuevo concepto, Valor del Contenedor. Pensemos en los contenedores como el color del frasco, (frasco rojo, por ejemplo) y en el valor del contenedor como en el contenido del frasco. (el frasco rojo contiene en número 12).

Contenedor	Valor del Contenedor
Identifica el contenedor	Identifica el contenido del contenedor



Observación: Este programa utiliza el valor del contenedor rojo (12), para que la espera sea 12 segundos.

Operaciones con Contenedores

Como se puede apreciar en el sub-menú de contenedores, hay una gran variedad de opciones y operaciones que pueden ser realizadas con los contenedores, desde llenar el contenedor con un valor hasta construir una fórmula, pasando por operaciones matemáticas, lógicas, valores de los sensores, temporizadores, etc.

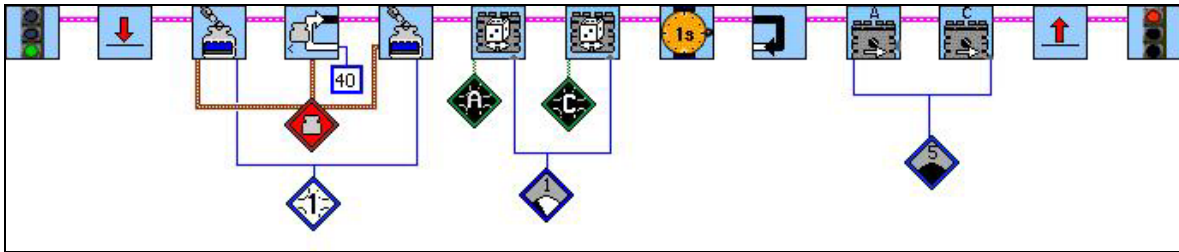
El siguiente ejemplo, muestra el uso valor al azar (random) en el contenedor amarillo para determinar el tipo de sonido a emitir.



En el siguiente programa, llenamos el contenedor amarillo con un número al azar, entre 0 y 6, lo mostramos en la pantalla y luego lo utilizamos para elegir el sonido a emitir por la bocina del RCX.

Actividad

Descubra, que hace el siguiente programa llamado “La cucaracha” y por qué se llamará así.



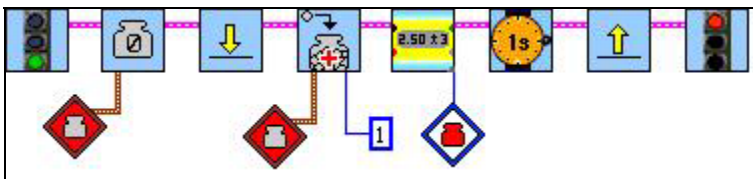
Importante

Es conveniente poner en cero(reset) los contenedores antes de usar, ya que esto nos asegura su uso correcto.

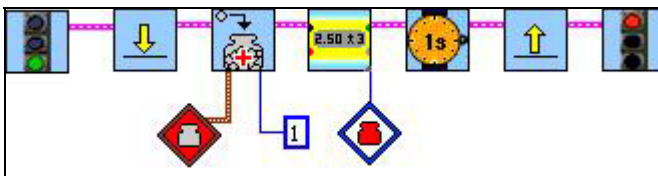


Para ello usamos el siguiente comando

Diferencias



En este programa cada vez que presionamos la tecla RUN, la cuenta comienza de cero.



En cambio en este, la cuenta comienza, desde el último valor almacenado en el contenedor rojo.