UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE LOS LAGOS / DIVISIÓN DE ESTUDIOS DE LA BIODIVERSIDAD E INNOVACIÓN TECNOLÓGICA / DEPARTAMENTO DE CIENCIAS EXACTAS Y TECNOLÓGICAS / LABORATORIO DE MINI-ROBÓTICA

Manual de usuario para kit LEGO^{MR} 9794

SISTEMAS ROBOTICOS

LABORATORIO DE MINI-ROBOTICA

ENRIQUE DÍAZ DE LEÓN S/N COL. PASEOS DE LA MONTAÑA, LAGOS DE MORENO, JALISCO.
TEL. Y Fax: +52 (474) 742 36 78, 742 43 14 Fax ext. 6527
www.lagos.udg.mx

Contenido

I.- Introducción

II.- Generalidades

III.- Partes que lo componen

IV.- RCX

V.- Sensores

VI.- Estructuras Básicas

VII.- Lenguajes de Programación

VIII.-Uso de la Cámara Web

IX.- Prácticas Propuestas

Lenguaje ROBOLAB

1.- Movimiento hacia adelante

2.- Ajuste de posición

3.- Movimiento y paro mediante sensor de tacto

4.- Movimiento y paro mediante sensor de temperatura

5.- Movimiento y paro mediante sensor de luz

6.- Movimiento y paro mediante sensor giro

7.- Ciclos repetitivos

8.- Seguidor de línea

9.- Uso de la Cámara

10.- Esquivar obstáculo mediante el uso de la cámara

Lenguaje NQC

11.- Movimiento hacia adelante y atrás

12.- Ciclo repetitivo

13.- Haciendo espirales

14.- Control y Lógica

15.- Sensor de tacto

16.- Sensor de temperatura

17.- Sensor de proximidad

18.- Sensor de luz

19.- Rutinas y Subrutinas

20.- Seguidor de línea

X.- Referencias

Introducción

La Robótica es hoy en día una de las áreas con más impulso dentro de las aplicaciones de Ingeniería, por lo que los conocimientos generales de la lógica en programación aplicada a algún mecanismo, resultan indispensables para todos aquellas personas cuya formación implica el generar procesos óptimos de automatización o control.

¿Qué necesitamos para construir un robot? Antes que nada paciencia y ganas de aprender. Una vez que contamos con los materiales necesarios y los conocimientos básicos de lo que será nuestro robot, no debemos desesperarnos cuando no funcione, debemos analizar las posibles causas para solucionar el problema y jamás perder el espíritu de búsqueda en nuevas opciones (construcción y programación).

Este manual tiene como finalidad dar a conocer a los alumnos de la materia Sistemas Robóticos del CULagos las especificaciones y características principales del kit LEGO^{MR} 9794, tales como son partes que lo componen, posibles estructuras a realizar, tipos de programación, etc.

Aunque se genera para los alumnos de una asignatura en particular, no se descarta la opción de que algún interesado en el área lo use, ya que el tipo de conocimiento que aquí se presenta puede ser utilizado por cualquier persona con inquietud hacia el área de la programación y la Mini-Robótica.

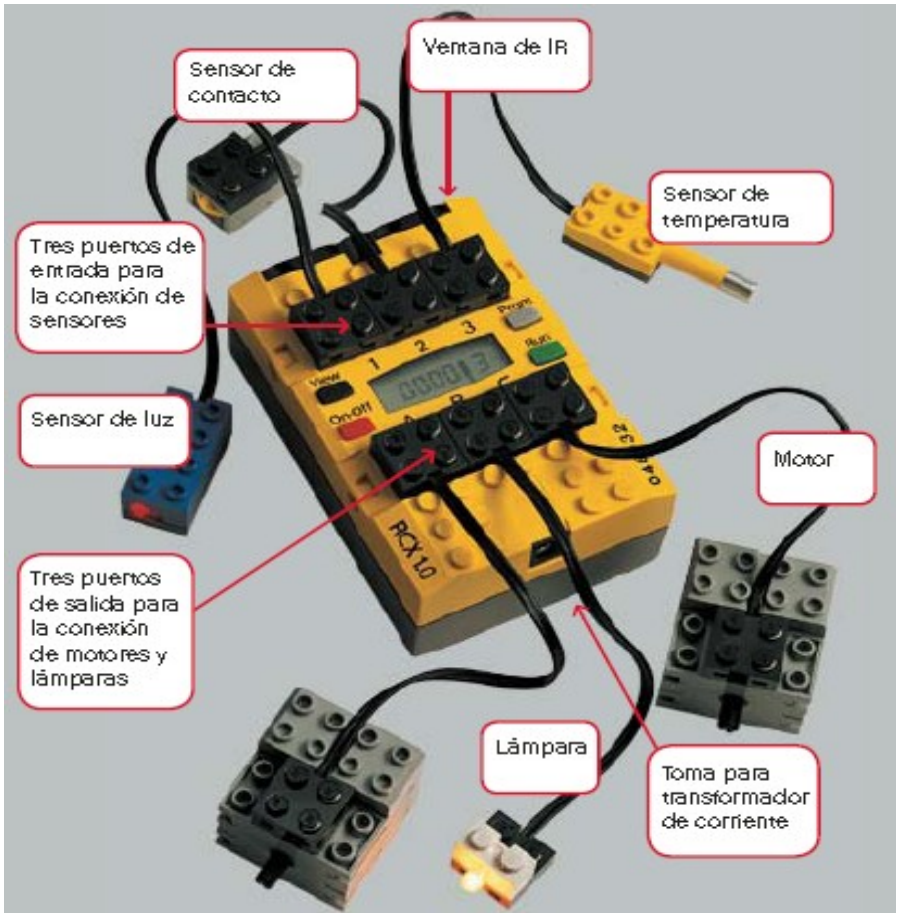
Ing. Diana Costilla López
Lab. Mini-Robótica

Generalidades

El kit LEGO[MR] 9794 fue el primero de la serie Mindstorms la que permite desarrollar las habilidades tanto de diseño mecánico como de programación estructurada y sobre todo análisis para resolución de problemas específicos con sus diversas restricciones. Las estructuras mecánicas a las cuales se adapte permitirán probar la programación generada para cumplir un objetivo determinado.

Partes que lo componen

Cada kit trae consigo un listado de piezas, dentro de las cuales destacan, el cerebro RCX, dos motores, sensor de contacto, sensor de luz, sensor de temperatura, lámpara, cables, torre usb de conexión mediante infrarrojos y el resto son piezas tales como engranes, ruedas, ejes, y coples.



RCX

El cerebro RCX es alimentado por 6 baterías AA de 1.5 v (no incluidas). Tiene 3 puertos para conectar los motores (A, B y C) y 3 puertos para conectar los sensores (1, 2, y 3). También tiene una ventana para conexión mediante IR, para comunicarse con otro RCX o con la PC y descargar de esta manera los programas generados.

El RCX tiene su pantalla display y además cuenta con 4 botones:

VIEW: negro
(*Este botón sirve para tomar las lecturas de los sensores)*
ON OFF: rojo
(Este botón sirve para seleccionar y también es para el encedido)
PRGM: gris
(Este botón permite seleccionar los programas que se encuentran en la memoria del RCX)
RUN: verde
(Este botón sirve para ejecutar los programas seleccionados)

Aparecerá un personaje cuando se encienda, si el programa es ejecutado mediante RUN, comenzará la animación y el personaje caminará.

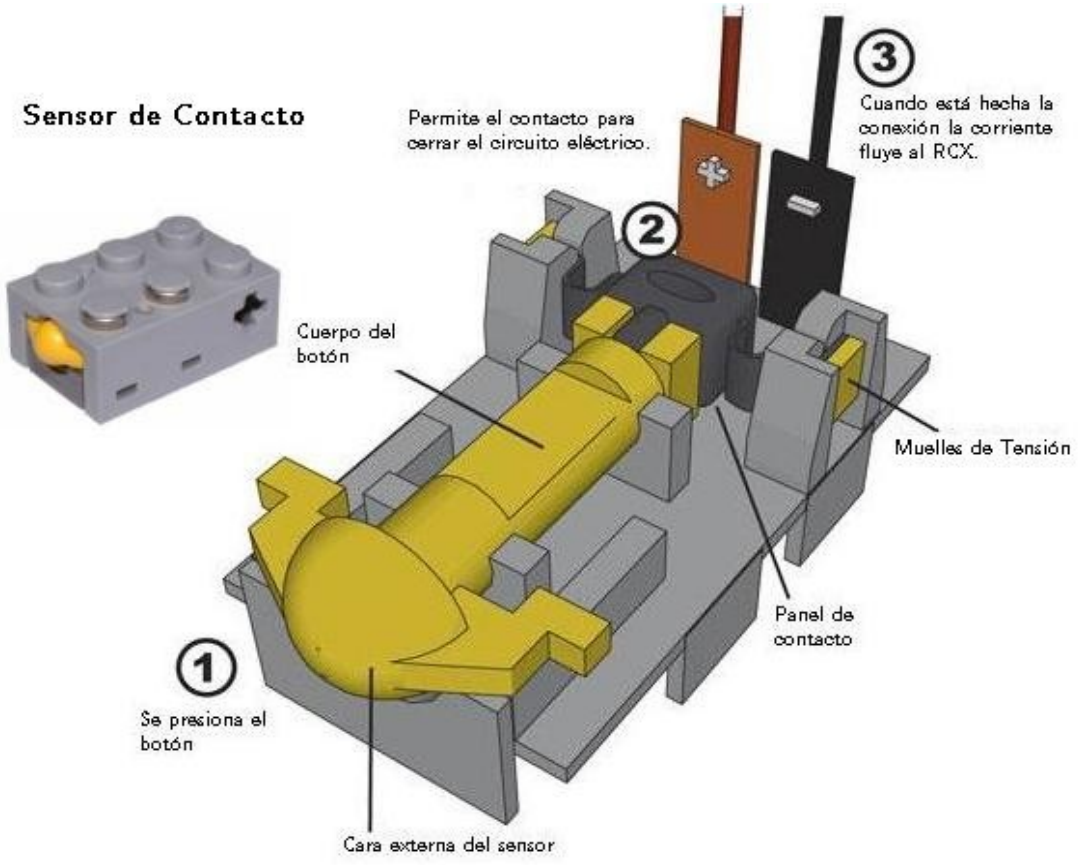
Sensores

Una de las partes fundamentales de la Robótica son los sensores ya que es mediante estos elementos que el sistema entra en contacto con el mundo exterior y permite al programador desarrollar respuestas a partir de las señales adquiridas. A continuación se muestra imagen y breve descripción del funcionamiento de cada uno de los sensores.

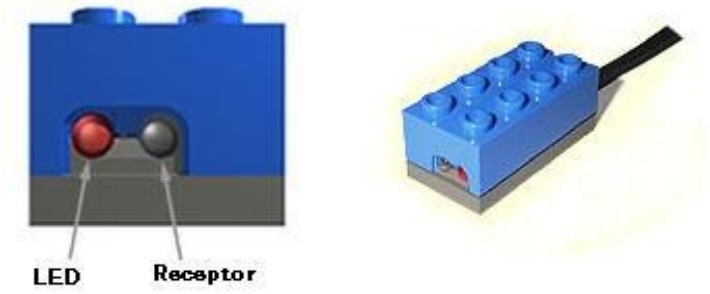
Los sensores que incluye el RCX son:

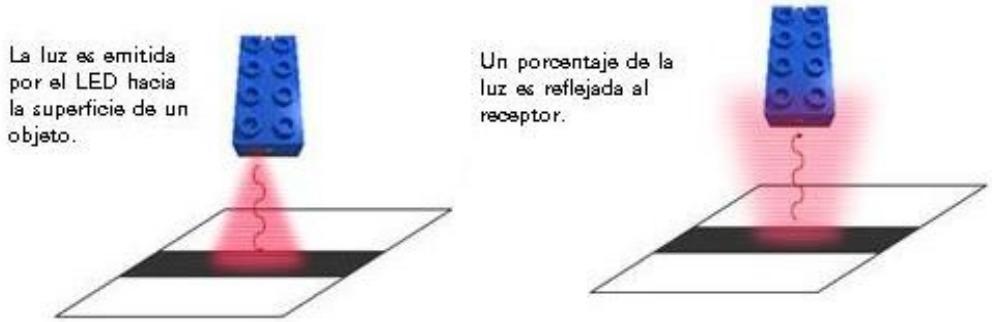
- Sensor de Conacto
- Sensor de Temperatura
- Sensor de Luz
- Sensor de Giro-Rotación (Servo-Motor)

Sensor de Contacto



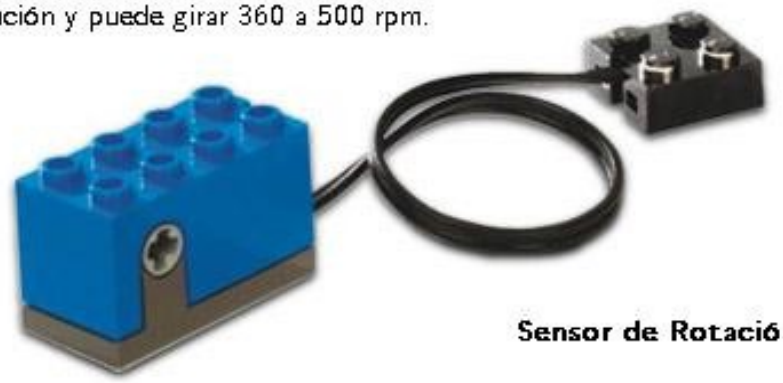
Sensor de Luz





El sensor analógico de temperatura tiene un rango de (-20 °C to +50 °C -4 °F to +122 °F)

El sensor digital de rotación, mide dirección y cantidad de rotaciones, velocidad angular y ángulos. Cuenta con 16 posiciones por revolución y puede girar 360 a 500 rpm.



Sensor de Rotación

Estructuras Básicas

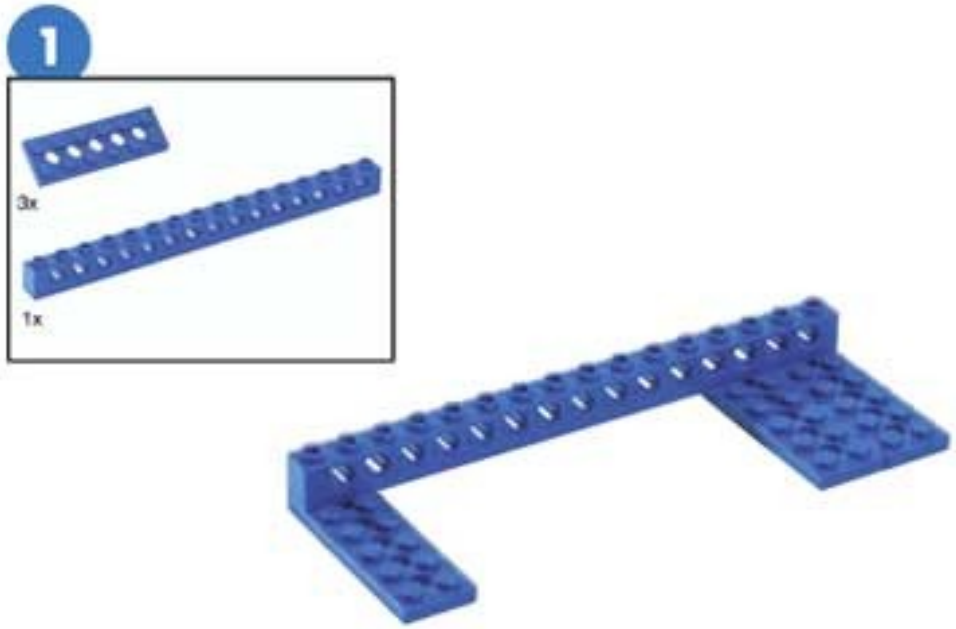
En cada kit se incluye un pequeño libro con la base para ensamblar un pequeño robot.

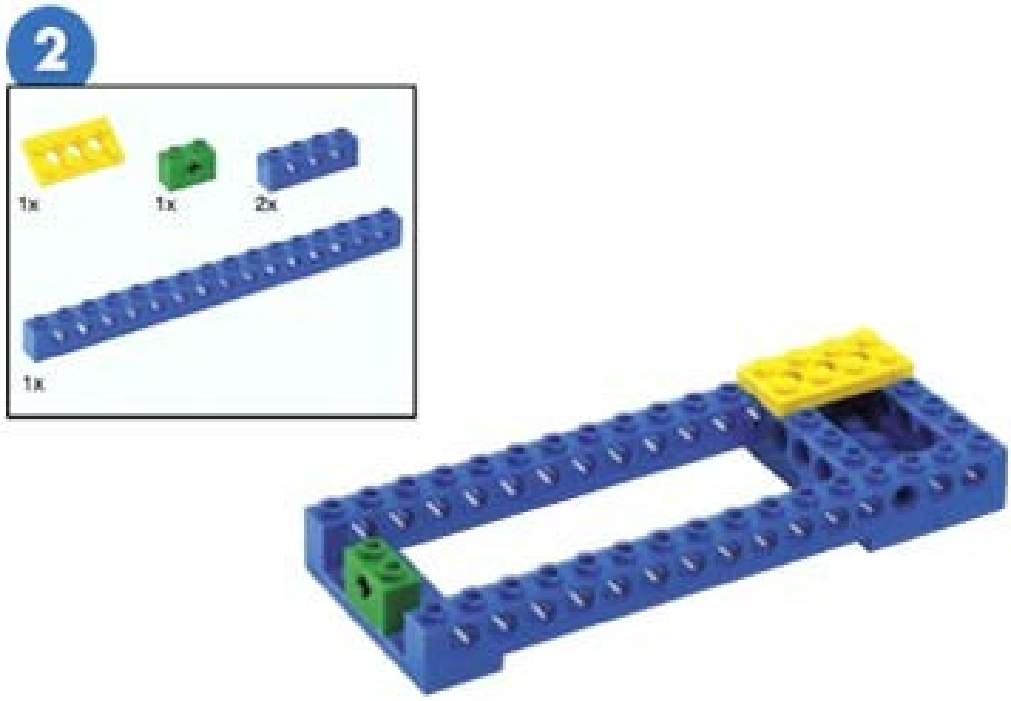
A continuación se anexan imágenes de una de las estructuras básicas, a partir de esta es posible conectar los sensores y partes mecánicas adicionales que le permitan al robot realizar sus tareas lo mejor posible.

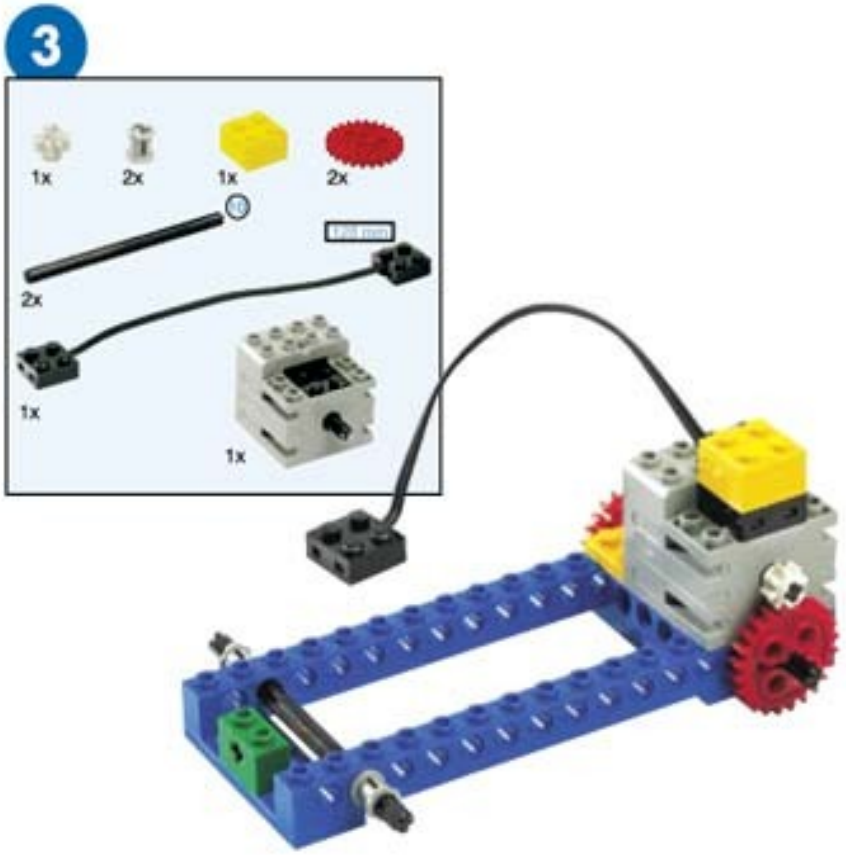
En cada imagen se muestran las piezas necesarias en número y tipo, así como el lugar donde ensamblar cada parte.

Cabe señalar que esta es una estructura de ejemplo y en la realidad la imaginación del usuario es la pauta a desarrollar mecanismos y elementos necesarios para cumplir las tareas asignadas.

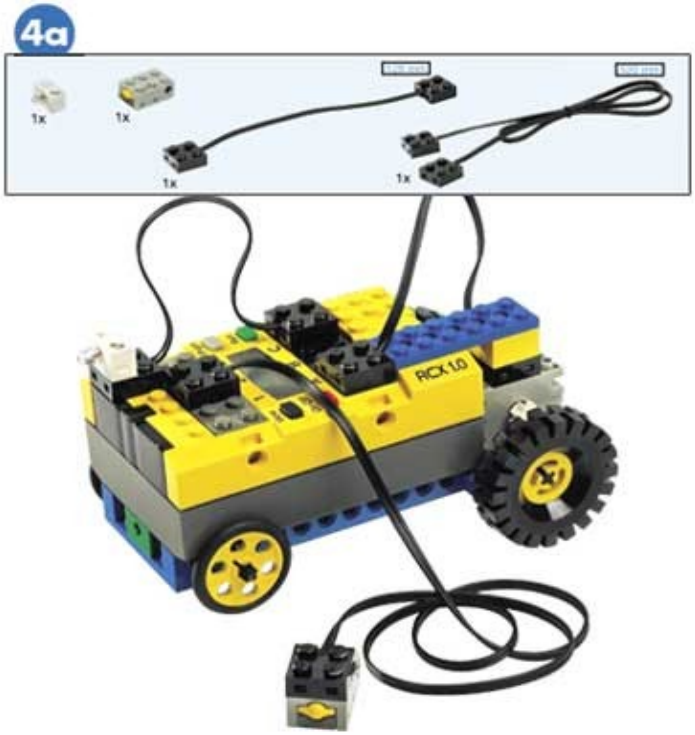
Es importante señalar que debes colocar las pilas (previamente cargadas) antes de montar el RCX ya que se evitará el tener que desarmar y armar la estructura.

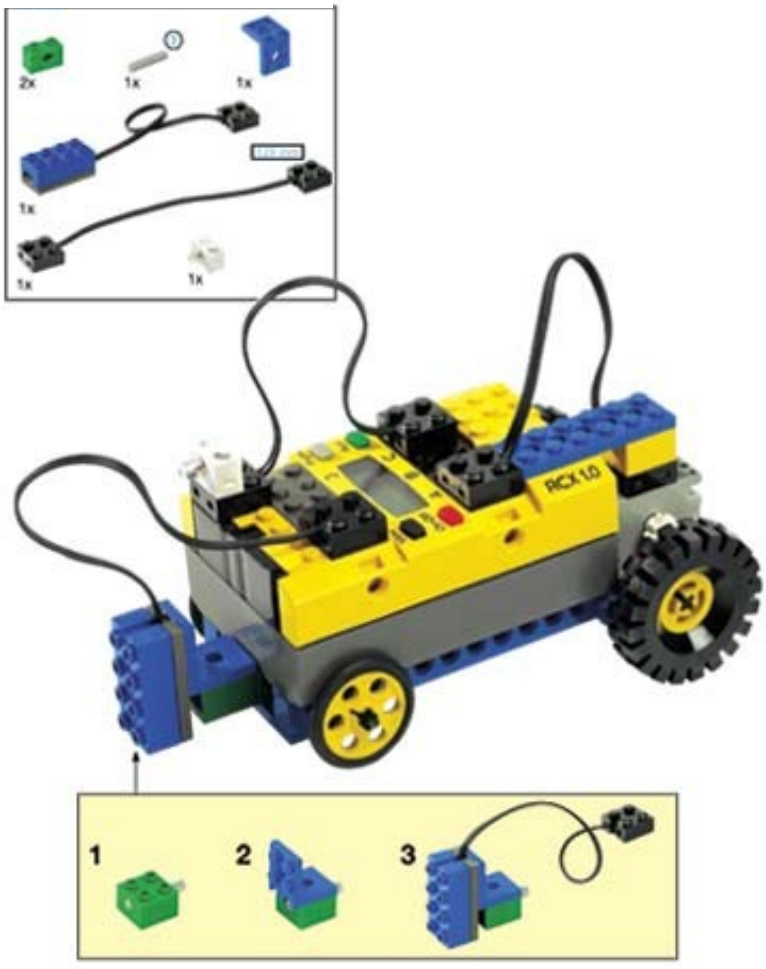





3
1x
2x
1x
2x
2x
1x
1x


4
2x
1x
2x
1x
RCX 1.0


4a
1x
1x
1x
1x
RCX 1.0


2x
1x
1x
1x
1x
1x
RCX 1.0
1
2
3

Lenguajes de Programación

Para programar los RCX abordaremos dos tipos de lenguajes, el visual del mismo nombre (ROBOLAB), basado en una plataforma LabVIEW, y el de código con base en C++.

ROBOLAB:

El software LEGO^{MR} RCX permite generar programas de una forma bastante fácil y amigable, ya que mediante un entorno visual podemos establecer una línea de comandos visuales (íconos) y definir especificaciones para cada uno de ellos.

Cuenta con 2 clasificaciones Programador e Investigador. En Programador se tienen dos niveles de avance, Piloto e Inventor, cada uno con 4 niveles internos con ejemplos. En Investigador también se cuenta con 5 niveles de programción.

Además permite formar ciclos y condiciones a partir de las señales obtenidas por los sensores, para así poder controlar el comportamiento de los motores. Puedes manejar funciones lógicas.

A continuación se anexa una lista de los principales íconos y su función:

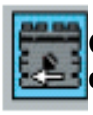
Comienzo y fin del programa

 **Comenzar**	Principio del programa, requerido como el primer comando en cada programa del inventor.
 **Final**	Final del programa, requerido como el comando de termino en cada programa del inventor.
 **Alto**	Detiene el encendido del RCX, puerto A.
 **Detener todas las salidas**	Detiene el encendido del RCX, puertos A, B, C.
 **Detener las salidas**	Detiene el encendido a los puertos especificados de RCX, por defecto - puertos A, B, C.
 **Motor A, giro hacia delante**	Dar vuelta al motor RCX del puerto A, encendido en la dirección delantera en poder máximo.
 **Motor A, giro hacia atrás**	Dar vuelta al motor RCX puerto A, encendido en la dirección contraria en poder máximo.

Salidas especificas

Lámpara Encender la lámpara, por defecto - todos los puertos, nivel 5 de la energía.

Giro hacia delante de motor Girar el motor, por defecto - todos los puertos, nivel 5 de la energía.

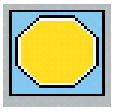


Giro hacia atrás de motor Dar vuelta al motor encendido en la dirección contraria, defecto - todos los puertos, nivel 5 de la energía.

Dirección del tirón Mover de un tirón en la dirección de la energía a los puertos especificados de RCX, por defecto - todos los puertos.

Enciende el sonido Encender un sonido en el RCX. Los sonidos disponibles son:

1-Tecleo dominante
2-BeepBeep
3-Barrido descendente
4- Barrido de levantamiento (ajuste por defecto)
5- Zumbido
6- Barrido de levantamiento rápido



Salidas flotantes Detiene el encendido de los puertos de salida y permiten que los dispositivos hagan girar a una parada.

¿Espera por ... ?

Sub-menu. Los iconos en este sub-menu especifican cuando los iconos del comando deben ejecutar el paro.

Para espera de 1 segundo. Espera 1 Segundo antes de continuar.

Espera por un tiempo. Esperar la cantidad de tiempo especificada, por defecto - 1 segundo.

Espera Tiempo al azar. Esperar una cantidad del tiempo al azar, por defecto - entre 0 y 5 segundos.

Espera para el empuje Esperar hasta que sensor del tacto se empuja hacia adentro, por defecto - entrar el puerto 1.

La espera para dejar ir

Esperar hasta que sensor del tacto sea lanzado, por defecto - entrar el puerto 1.

Espera para la luz

Esperar hasta que el sensor de luz lee un valor que sea más brillante que el número especificado, por defecto =55, puerto entrada 1.

Espera para la obscuridad

Esperar hasta que el sensor de luz lee un valor que sea más oscuro que el número especificado, defecto =55, puerto de entrada 1.

Espera para más brillante

Esperar el sensor de luz para leer un valor que sea mayor que el valor actual. Por defecto – entrada en el puerto 1,el aumento del valor de luz es de 5.

Espera para más oscuro

Esperar el sensor de luz para leer un valor que sea menos que valor actual. Por defecto – entrada en el puerto 1,la disminución del valor de luz es de 5.

Espera para la rotación Sin reajuste

Esperar hasta que el valor del sensor del ángulo es mayor que el número de las rotaciones especificadas (en 16ths de una rotación) en cualquier dirección. Este programa no pondrá a cero el sensor cada vez.

Espera para el aumento en sensor de la cámara fotográfica

Esperar hasta que el sensor de la cámara fotográfica lee un valor que sea mayor que el número especificado.

Espera para la disminución en sensor de la cámara fotográfica

Esperar hasta que el sensor de la cámara fotográfica lee un valor que sea menos que el número especificado.

Espera para aumentar Temperatura (c)

Esperar hasta que la temperatura es mayor que el número especificado. Por defecto - 30 centígrados encendido puerto 1 de la entrada.

Espera de RCX para Rotación

Esperar hasta que el valor del sensor del ángulo es mayor que el número de las rotaciones especificadas.
Por defecto - 16 (una rotación) encendido puerto 1 de la entrada.

 **Espera para el ángulo**	Esperar hasta que el valor del sensor del ángulo es mayor que el ángulo especificado (en cualquier dirección).Por defecto - 180 grados encendido motor 1 de la entrada.
 **Espera para el envase**	Esperar hasta que el envase es igual al número especificado. Por defecto - envase rojo igual a 1.
 **Espera para el contador de tiempo**	Esperar hasta que el contador de tiempo alcanza un valor especificado. Por defecto - contador de tiempo rojo igual a 1 segundo. ¡TÚ DEBES PONER A CERO EL CONTADOR DE TIEMPO PRIMERO!
 **Espera para el correo**	Esperar hasta que el correo recibido de otro RCX es igual al número especificado. Por Defecto - cualquiera número entero.
 **Modificantes Sub-menu**	Los iconos en este sub-menu especifican las localizaciones, los niveles de la energía, y los valores portuarios usados con iconos del comando.
 **Entrada 1**	Conectar el cable a este comando modificado para seleccionar el puerto de entrada 1.
 **Hacer salir A**	Conectar el cable a este comando modificado para seleccionar el puerto de salida A.
 **Accionar nivel 4**	Conectar este cable modificado entre un motor o una lámpara para fijar el nivel de energía a 4.
 **Constante numérica**	Conectar este cable modificado entre un sensor o medir el tiempo para fijar un valor constante.

Estructuras - salto y tierra

 **Saltos Sub-menu**	Los iconos en este sub-menu especifican en donde saltará y aterrizará el programa.
 **Salto**	Hacer el salto del programa a un lugar específico en la secuencia.
 **Tierra**	Este comando es donde el programa saltará cuando utilizas comando del salto rojo.

Valor del envase rojo El valor del envase rojo.

Envase rojo Conecta el cable al comando del envase para seleccionar el envase rojo.

Número al azar Un número al azar entre 0 y 8.

Valor del puerto 1 El valor del puerto 1.

Contador de tiempo rojo Conecta el cable al comando contador de tiempo para seleccionar el contador de tiempo rojo.

Valor del contador de tiempo rojo El valor del contador de tiempo rojo.

Valor del correo El valor del correo.

Valor de soportes lógico inalterable El valor es el número de versión de los soportes lógico inalterable multiplicado por 100.

Valor de la batería El valor es el número del voltaje de la batería multiplicado antes de 1000.

Música

Sub-menu de la música Los iconos en este sub-menu especifican cómo las notas musicales deben ser escuchadas.

Nota C de la música Reproduce la nota musical en el RCX. Por defecto - cuarta nota en la escala estándar.

Resto Insertar pausa en la música.

Duración musical Especifica la duración de tiempo para reproducir una nota.

Sube una octava Conectar el cable al comando de música para subir los agudos por una octava o más octavas, si más de uno se conecta a la vez.

Carga el archivo de texto. Esta rutina agregará las notas musicales actualmente en un archivo en tu programa de inventor.

Estructuras - lazos

Sub-menu de los lazos

Los iconos en este sub-menu especifican donde los lazos del programa comenzarán y terminarán.

Comienzo del lazo

Comenzar una estructura del lazo. Por defecto - lazo dos veces.

Fin del lazo

Saltar de nuevo al comienzo del lazo un número especificado de tiempos.

Lazo del tacto

Comenzar un lazo que se repita mientras que el sensor del tacto sea empujado.

Colocar mientras que el sensor de la cámara fotográfica es Mayor que

Comienza un lazo que repita mientras que el valor del sensor de la cámara fotográfica es mayor que un número especificado.

Colocar mientras que el sensor de la cámara fotográfica es Menor que

Comienza un lazo que repita mientras que el valor del sensor de la cámara fotográfica es menor que un número especificado.

Estructuras - bifurcaciones y tareas

Fractura de la tarea

Comenzar una nueva tarea con este comando de funcionar tareas de función múltiples simultáneamente.

Sub-menu de las bifurcaciones

Los iconos en este sub-menu especifican donde el programa elegirá entre dos trayectorias y donde se combinará otra vez.

Bifurcación del sensor del tacto

Hacer que el programa elija entre una de las dos trayectorias dependiendo del estado del sensor del tacto. Omitir el puerto 1 de la entrada.

Unión de la bifurcación

Combinar las dos secuencias de una bifurcación juntas detrás. Debe ser utilizada con una bifurcación.

Bifurcación del sensor de la cámara fotográfica

Elegir una trayectoria dependiendo de si el valor del sensor de la cámara fotográfica es mayor que o menor-que un número especificado.

- Si el sensor de la cámara fotográfica es mayor que el valor especificado, el programa seguirá la secuencia superior.
- Si el sensor de la cámara fotográfica es menor que el valor especificado, el programa seguirá la secuencia inferior.

Bifurcación al azar

Hacer que el programa elija entre una de dos trayectorias aleatoriamente.

Estructuras – subrutinas

Crear la subrutina

Crear una nueva subrutina. La subrutina no funcionará en este punto del programa. Funcionará cuando el programa alcanza el icono del funcionamiento de la subrutina.

Correr la subrutina

Especificar donde funciona la subrutina en el programa.

Suprimir la subrutina

Suprimir las subrutinas especificadas en el RCX. Por defecto es para suprimir la subrutina 0.

Envase

Sub-menu del envase

Los iconos en este sub-menu manipulan los envases (las variables) y los valores dentro de ellos.

Agregar al envase

Agregar un número al envase. Por defecto - agregar 1 al envase rojo.

Quitar del envase

Restar un número del envase. Por defecto - restar 1 del envase rojo.

Llenar el envase

Fijar el envase a cierto valor. Por defecto - fijar el envase rojo a 1.

Tocar el envase

Fijar al envase el valor del sensor del tacto.

Envase del valor del contador de tiempo

Fijar al envase el valor del contador de tiempo. Por defecto-fijar el envase rojo al valor del contador de tiempo rojo.

Envase de fórmula

Fijar el envase a una fórmula.

Envase del estado del acontecimiento

Fijar el envase a cierto estado del acontecimiento. Dice si el acontecimiento esté en el punto bajo, estado normal o alto dependiendo de los umbrales del sistema.

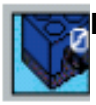
Envase del registro del acontecimiento

Fijar el envase a una copia del registro del pedacito de los acontecimientos acertados para la tarea actual.

Envase del sensor de la cámara fotográfica

Fijar el envase a cierto valor

Reajuste

 **Reajustar el Sub-menu**	Los iconos en este sub-menu reajustaron los envases, los contadores de tiempo, y los sensores a cero.
 **Vaciar el envase**	Reajustar el valor del envase a cero. Por defecto - fijar el envase rojo a cero.
 **Poner a cero el contador de tiempo**	Reajustar el valor del contador de tiempo a cero. Por defecto - fijar el contador de tiempo rojo a cero.
 **Poner a cero el sensor del ángulo**	Reajustar el sensor del ángulo a cero. Por defecto - puerto 1 de entrada.
 **Caja vacía**	Reajustar el valor de la caja de RCX a cero. Esto vacía la caja así que el correo puede ser recibido de otro RCX.
 **Poner cero al sensor del tacto**	Reajustar el sensor del tacto.
 **Poner cero al sensor de luz**	Reajustar el sensor de luz.
 **Poner cero al sensor de temperatura (Centígrado)**	Reajustar el sensor de temperatura de centígrados.
 **Poner cero al sensor de temperatura (Fahrenheit)**	Reajustar el sensor de temperatura de Fahrenheit.
 **Sensor de luz**	Recoger los datos del sensor de luz.
 **Sensor del tacto**	Recoger una cuenta de presión del sensor del tacto.
 **Sensor de temperatura**	Recoger los datos del sensor de temperatura.
 **Sensor de la rotación**	Recoger los datos del sensor del ángulo.
 **Adaptador del sensor**	Recoger los datos del adaptador del sensor.

1 sec	Fijar la tarifa del muestreo a 1 segundo entre cada punto de referencias.
Intervalo de la registración de datos	Fijar la tarifa del muestreo al intervalo usuario-especificando el tiempo entre cada punto de datos.
Intervalo basado en sensores del tacto	Recoger los datos cada vez que se lanza el sensor del tacto.
10 puntos	Recoger 10 puntos de referencias.
Los puntos de ajuste	Recoger tantos puntos de referencias según lo especificado.
Datos que entran	Recoger los datos durante el paso.
Datos que salen	No recoger los datos durante el paso.

Modificantes de la registración de datos

Muestra una décimo de segundo	Capturar los datos cada un décimo de un segundo.
Muestra un minuto	Capturar los datos cada minuto.
Muestra una hora	Capturar los datos cada hora.
Muestreo del tacto	Capturar los datos cada vez que se presiona el sensor del tacto.
Datos del sistema rojo	Identificar la localización en donde se almacenan los datos.
Con Muestra libre Tiempo de estampado	Capturar los datos cada vez que escribes en el sistema de datos y marcas los datos con el tiempo.

Registración de datos – investigador nivel 4

Sub-menu del investigador

Los iconos en este sub-menu controlan la operación de las funciones de la registración de datos del RCX.

Inicializar la registración del sensor de luz

Inicializar la toma de datos del sensor de luz y configura ajustes de la registración de datos. Por defecto ajustes: sensor en el puerto 1, datos del sistema rojo.

Inicializar la registración del sensor de tacto

Inicializar la toma de datos del sensor de tacto y configura ajustes de la registración de datos. Por defecto ajustes: sensor en el puerto 1, datos del sistema rojo

Inicializar la registración del sensor de temperatura

Inicializar la toma de datos del sensor de temperatura y configura ajustes de la registración de datos. Por defecto ajustes: sensor en el puerto 1, datos del sistema rojo

Inicializar la registración del sensor de rotación

Inicializar la toma de datos del sensor de rotación y configura ajustes de la registración de datos. Por defecto ajustes: sensor en el puerto 1, datos del sistema rojo

Inicializar la registración del sensor de tecleo

Inicializar la toma de datos del numero de tecleos del sensor del tacto y configura ajustes de la registración de datos. Por defecto ajustes: sensor en el puerto 1, datos del sistema rojo

Inicializar la registración del envase

Inicializar la toma de datos de un envase y configura ajustes de la registración de datos.

Inicializar la registración del contador de tiempo

Inicializar la toma de datos de un contador de tiempo y configura ajustes de la registración de datos.

Comenzar la registración de datos

Comenzar a capturar la registración de datos

Parar el registrar

Detener la captura de la registración de datos

Reasumir el registrar

Reasumir la captura de la registración de datos

Comenzar la registración de datos con Tecleos

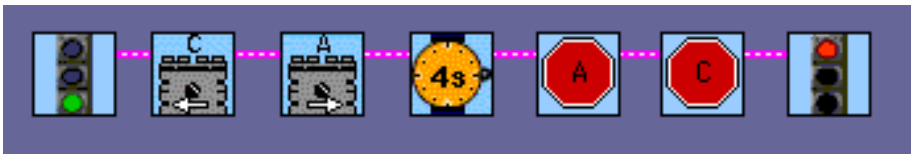
Comenzar a capturar o los datos y los tecleos de registración cada vez que se toma un punto de referencia.

Programa Ejemplo:

AVANZAR UN TIEMPO DETERMINADO

Un programa sencillo, al accionar este programa el robot avanzará y al pasar el tiempo establecido se detendrá hasta que volvamos a accionar el programa.

Para lograr el funcionamiento arriba mencionado tenemos el siguiente programa, en el cual comenzamos por colocar el ícono correspondiente al arranque, posteriormente los motores (se indican los conectados a los puertos A y C) y se indica que se moverán hacia adelante durante 4 segundos. Al concluir el tiempo ambos motores se detendrán y para concluir se indica el paro del programa. Este programa solo volverá a funcionar hasta que vuelva a seleccionarse desde el RCX y se indique ejecutar.

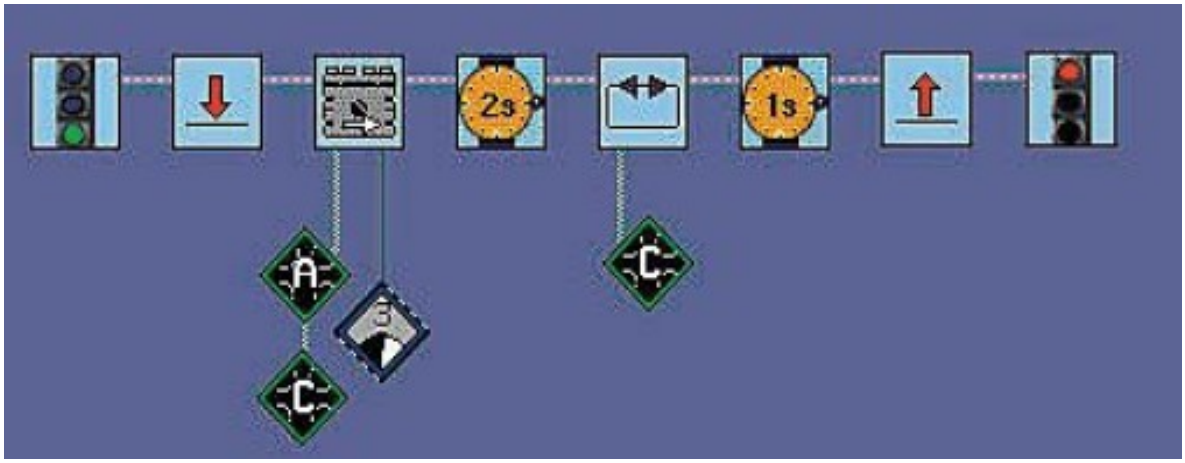


Programa Ejemplo:

CAMBIO DE DIRECCIÓN

Este programa permite ser descargado, avanzar los motores, cambiar de dirección, esto dependiendo del tiempo indicado. Dicho programa será recursivo.

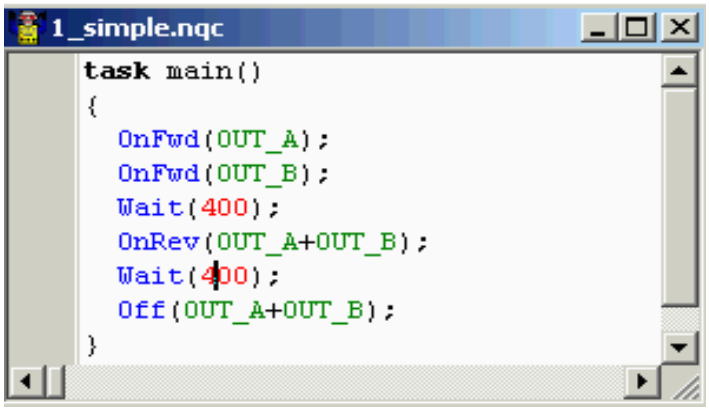
En este programa se están utilizando un ciclo (saltar y aterrizar) y además se está variando la potencia de los motores.



De esta forma pueden irse generando programas y tareas cada vez más complejas.

NQC: Lenguaje en código basado en el NBC que corren sobre C++, será necesario el software BricxCC para poder compilar y guardar con la extención correspondiente (*.nqc)

En la ventana del BricxCC podremos generar el código y compilarlo para ver que no tenga errores en la sintaxis, también podemos seleccionar el puerto en el cual debe buscarse el cerebro RCX para cargarle el programa correspondiente.



Programa Ejemplo: **Movimiento hacia adelante, y hacia atrás**

```
task main()
{
OnFwd(OUT_A);
OnFwd(OUT_C);
Wait(400);
OnRev(OUT_A+OUT_C);
Wait(400);
Off(OUT_A+OUT_C);
}
```

1. Se indica la primera tarea **task main ()**
2. Se abre la llave de la tarea.
3. La instrucción **OnFwd** (Encendido hacia adenlante) debe ir seguida por la salida (motor en el puerto, A, B, o C) que va realizar la función, tambien es posible indicar el nivel de velocidad del motor lo que se verá a continuación.
4. Para que espere un tiempo utilizamos **Wait** y se indican los segundos, para nuestro ejemplo 400 quiere decir 4 segundos completos.
5. La instrucción **OnRev** (Encendido en reversa) debe ir seguida por la salida (motor en el puerto, en este caso A y C dentro de una sola instrucción).
6. Nuevamente un tiempo de espera con **Wait** en 4 segundos.
7. La función **Off** será quien determine el paro de los motores, en el programa se indican las salidas A y C (puertos del motor) en una sola indicación.
8. Se cierra la llave de la tarea **task main ()**
9. Cada renglón debe concluir con punto y coma.

Programa Ejemplo: **Movimiento hacia adelante, y hacia atrás ***

```
task main()
{
SetPower(OUT_A+OUT_C,2);
OnFwd(OUT_A+OUT_C);
Wait(400);
OnRev(OUT_A+OUT_C);
Wait(400);
Off(OUT_A+OUT_C);
}
```

Exactamente el mismo programa pero ahora se define la velocidad de los motores en **SETPOWER** con 2, (la velocidad es variable de 0 a 7).

Programa Ejemplo: **Declaración de variables.**

A continuación se muestra un ejemplo de como definir variables, las cuales serán llamadas en lugar de un valor para los tiempos de espera.

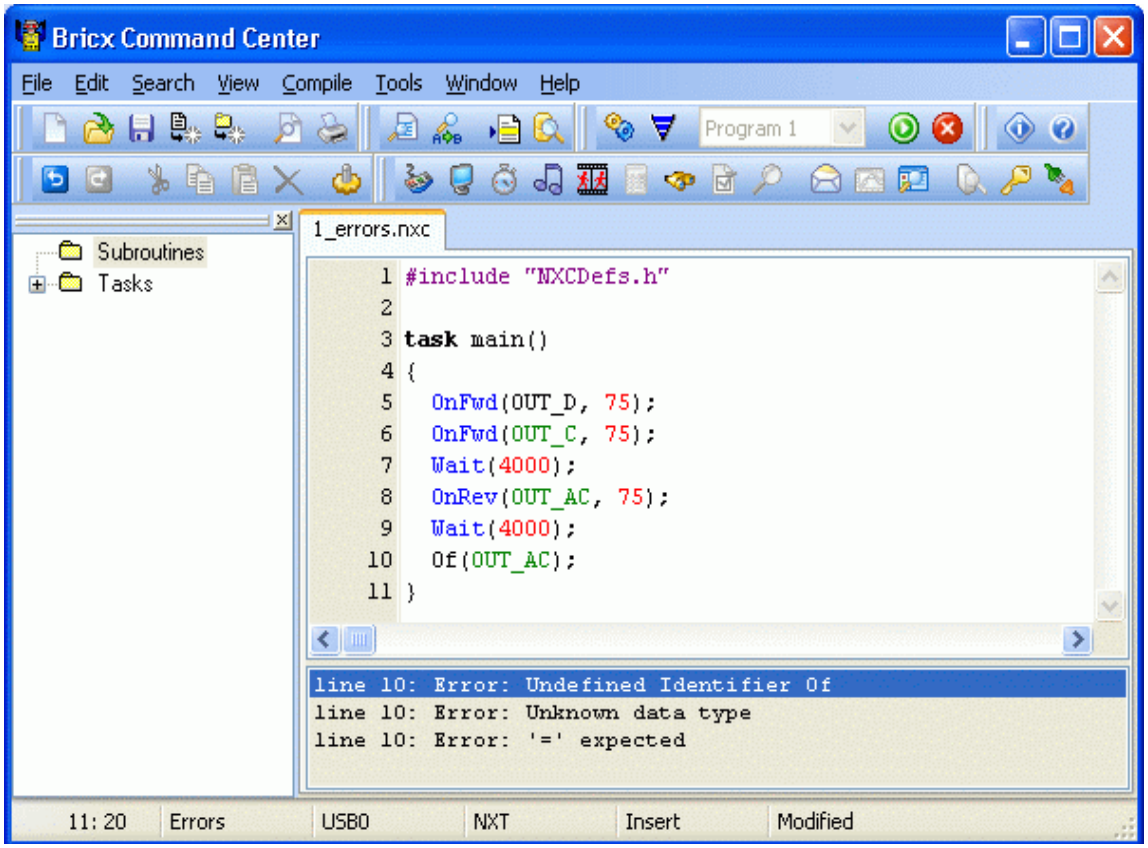
```
#define MOVE_TIME 100
#define TURN_TIME 85
task main()
{
OnFwd(OUT_A+OUT_C);
Wait(MOVE_TIME);
OnRev(OUT_C);
Wait(TURN_TIME);
Off(OUT_A+OUT_C);
}
```

Para compilar y bajar el programa

En la imagen anterior se muestran los controles (iconos) para compilar, bajar al RCX, correr y detener el programa, respectivamente.

Para correr el programa de forma automática se debe oprimir **Ctrl+F5**.

En caso de generar algun error, se mostrará como en la siguiente pantalla:



Al igual que la mayoría de los compiladores se muestra la línea en la que existe algún error de sentencia o comunmente tipográfico.

A continuación se presentan algunos de los Macros o terminos principales, para ver el resto consultar el manual (**NQC N**ot **Q**uite **C**).

Sensores.-

Hay tres sensores que internamente están numerados 0, 1 y 2. Puede crear confusión ya que externamente son 1, 2 y 3. Por esto están definidos como SENSOR_1, SENSOR_2 y SENSOR_3.

Tipos y Modos.-

Los puertos de los sensores son capaces de recibir diferentes tipos de sensores, así que deberá indicarse en el programa el tipo de sensor que está conectado a cada puerto.

Tipo de Sensor	Significado
SENSOR_TYPE_NONE	Sensor generador pasivo
SENSOR_TYPE_TOUCH	Sensor de Contacto

SENSOR_TYPE_TEMPERATURE	Sensor de Temperatura
SENSOR_TYPE_LIGHT	Sensor de Luz
SENSOR_TYPE_ROTATION	Sensor de Rotación

Modo del Sensor	Significado
SENSOR_MODE_RAW	Valor bruto de 0-1023
SENSOR_MODE_BOOL	Valor booleano 0 o 1
SENSOR_MODE_EDGE	Cuenta el número de transiciones booleanas
SENSOR_MODE_PULSE	Cuenta el número de periodos booleanos
SENSOR_MODE_PERCENT	Valor porcentual de 0-100
SENSOR_MODE_FAHRENHEIT	Grados Farenheit (para el RCX solamente)

Configuración de sensor	Tipo	Modo
SENSOR_TOUCH	SENSOR_TYPE_TOUCH	SENSOR_MODE_BOOL
SENSOR_LIGHT	SENSOR_TYPE_LIGHT	SENSOR_MODE_PERCENT
SENSOR_ROTATION	SENSOR_TYPE_ROTATION	SENSOR_MODE_ROTATION
SENSOR_CELSIUS	SENSOR_TYPE_TEMPERATURE	SENSOR_MODE_CELSIUS
SENSOR_FAHRENHEIT	SENSOR_TYPE_TEMPERATURE	SENSOR_MODE_FAHRENHEIT
SENSOR_PULSE	SENSOR_TYPE_TOUCH	SENSOR_MODE_PULSE
SENSOR_EDGE	SENSOR_TYPE_TOUCH	SENSOR_MODE_EDGE

El puerto debe ser una constante numérica

SetSensor(sensor, configuración): fija el tipo y modo del sensor .
SetSensorType(sensor, tipo): fija el tipo de sensor.
SetSensorMode(sensor, modo): fija el modo de sensor.
ClearSensor(sensor): limpia el sensor (Ajusta a cero).
SensorValue(n): indica la lectura procesada del sensor n, donde n es 0, 1 o 2.

SensorType(n): indica el tipo de configuración del sensor n.

SensorMode(n): indica el modo actual del sensor.

SensorValueBool(n): indica el valor booleano del sensor nfija el tipo de sensor.

SensorValueRaw(n): indica el valor bruto del sensor n.

SetSensorLowerLimit(valor): fija el valor mínimo del sensor de luz. (*El valor puede ser una expresión*)

SetSensorUpperLimit(valor): fija el valor máximo del sensor de luz. (*El valor puede ser una expresión*)

SetSensorHysteresis(valor): fija el valor de histéresis del sensor de luz. (*El valor puede ser una expresión*)

CalibrateSensor( ): lee el valor actual del sensor de luz.

Funciones básicas.-

Modo	**Significado**
OUT_OFF	La salida está apagada (el motor no puede girar)
OUT_ON	La salida está encendida (el motor puede funcionar)
OUT_FLOAT	El motor seguirá girando hasta detenerse por si solo

Dirección	**Significado**
OUT_FWD	El motor gira hacia adelante
OUT_REV	El motor gira hacia atrás
OUT_TOGGLE	El motor invierte el sentido de giro

SetOutput(salidas, modo): fija las salidas en el modo especificaco.

SetDirections(salidas, modo): fija las salidas en la dirección especificada.

SetPower(salidas, potencia): fija las salidas en el nivel de potencia o velocidad especificada (0-7, o puede utilizarse OUT_LOW, OUT_HALF y OUT_FULL).

OutputStatus(n): indica la configuración actual de la salida para el motor n, 0,1 o 2.

Otras funciones.-

On(salidas): enciende la salida indicada.

Off(salidas): apaga la salida indicada.

Float(salidas): convierte las salidas en flotantes.

Fwd(salidas): activa las salidas hacia adelante.

Rev(salidas): activa las salidas hacia atrás.

Toggle(salidas): cambia la dirección de las salidas.

OnFwd(salidas): arranca las salidas.

OnRev(salidas): igual a OnFwd pero en la dirección opuesta

OnFor(salidas, tiempo): enciende las salidas durante un tiempo determinado.

EnableOutput(salidas):una función de ayuda para habilidar la salida específica.

DisableOutput(salidas): una función de ayuda para deshabilitar la salida especificada.

InvertOutput(salidas): una función de ayuda para invertir la dirección de la salida especificada.

ObvertOutput(salidas): una función de ayuda para regresar la dirección de la salida indicada hacia adelante.

Sonido.-

PlaySound(sonido): toca uno de los 6 sonidos preestablecidos para RCX. El argumento sonido debe ser una constante. Las siguientes constantes son pre-definidas para usarse con PlaySound: SOUND_CLICK, SOUND_DOUBLE_BEEP, SOUND_DOWN, SOUND_UP, SOUND_LOW_BEEP, SOUND_FAST_UP.

PlayTone(frecuencia, duración): toca un tono simple para la frecuencia y duración indicada.

Display LCD.-

Modo	Contenido del LCD
DISPLAY_WATCH	Muestra el reloj del sistema
DISPLAY_SENSOR_1	Muestra el valor del sensor 1
DISPLAY_SENSOR_2	Muestra el valor del sensor 2
DISPLAY_SENSOR_3	Muestra el valor del sensor 3
DISPLAY_OUT_A	Muestra la configuración de la salida A
DISPLAY_OUT_B	Muestra la configuración de la salida B
DISPLAY_OUT_C	Muestra la configuración de la salida C

El RCX tiene 7 diferente recuencia, duración): toca un tono. La frecuencia esta en Hz. La duración esta en milisegundo.

SelectDisplay(modo**)** Selecciona un modo de display.

Comunicaciones.-

ClearMessage(valor,precisión**)** Borra el buffer de mensajes.

SendMessage(mensaje**)** Envía un mensaje por infrarrojos. Mensaje puede ser una expresión, pero el RCX sólo puede enviar mensajes con un valor entre 0 y 255, por lo tanto sólo los 8 bits menores del argumento serán usados.

SetTxPower(potencia**)** Establece la potencia para la transmisión de infrarrojos.

Temporizadores.-

ClearTimer(n**)** Pone a cero el temporizador especificado.
Timer(n**)** Devuelve el valor actual del temporizador especificado (con una resolución de 100 ms).

Contadores.-

ClearCounter(n**)** Pone a cero el contador n. n tiene que ser 0 ó 1 para el Scout, 0-2 para el RCX2.
IncCounter(n**)** Incrementa el contador n en 1. n tiene que ser 0 ó 1 para el Scout, 0-2 para el RCX2.
DecCounter(n**)** Decrementa el contador n en 1. n tiene que ser 0 ó 1 para el Scout, 0-2 para el RCX2.
Counter(n**)** Devuelve el valor del contador n. n tiene que ser 0 ó 1 para el Scout, 0-2 para el RCX2.

Registro de datos.-

CreateDatalog(tamaño**)** Crea un registro de datos del tamaño especificado (que debe ser constante). Un tamaño de 0 elimina el anterior sin crear uno nuevo.
AddToDatalog(valor**)** Añade el valor, que puede ser una expresión, al registro de datos. Si el registro de datos está lleno la llamada no tiene efecto.
UploadDatalog(comienzo, contador**)** Inicia y transmite un número de datos igual a contador, comenzando por el valor de comienzo. Este comando no es muy útil ya que el ordenador es el que normalmente comienza la transmisión.

Características generales.-

Wait(tiempo**)** Hace parar una tarea durante un tiempo especificado (en centésimas de segundo). El argumento tiempo puede ser una expresión o una constante.
StopAllTasks( ) Detiene todas las tareas que están ejecutándose. Esto hará parar el programa completamente, por lo tanto, cualquier comando que siga a éste será ignorado.
Random(n**)** Devuelve un número aleatorio entre 0 y n. n debe ser una constante.
SetSleepTime(minutos**)** Establece el numero de minutos (que debe ser constante) que estará activo hasta que se duerma. Especificando 0 minutos deshabilita esta

opción.
SleepNow(n**)** Fuerza a que se duerma. Sólo funciona si el tiempo para que se duerma no es cero.

Características específicas del RCX.-

Program( ) Número del programa que está seleccionado.
Watch( ) Devuelve el valor del reloj del sistema en minutos.
SetWatch(horas,minutos**)** Establece en el reloj de sistema un número específico de horas y minutos. Horas debe ser un número constante entre 0 y 23 ambos incluidos. Minutos debe ser una constante entre 0 y 59 ambos incluidos.
SetScoutMode(modo**)** Pone el Scout en modo stand-alone(0) o power (1). Dentro de un programa sólo tiene sentido utilizarla para ponerlo en modo stand-alone, ya que para que pueda ejecutarse un programa NQC debe estar en modo power.

Referencia rápida de Instrucciones.-

Instrucción	Descripción
while (cond) cuerpo	Ejecuta el cuerpo cero o más veces mientras la condición es verdadera
do (cuerpo) **while** (cond)	Ejecuta el cuerpo cero o más veces mientras la condición es verdadera
until (cond) cuerpo	Ejecuta el cuerpo cero o más veces hasta que la condición sea verdadera
break	Salir dell cuerpo de un while/do/until
continue	Saltarse la siguiente iteración del cuerpo de un while/do/until
repeat (expresión) cuerpo	Repetir el cuerpo un número determinado de veces
if (cond) acción1 **if** (cond) acción1 **else** acción2	Ejecuta acción1 si la condición es verdadera. Ejecuta acción2 (si existe) si la condición no se cumple.
start nombre_de_tarea	Inicia la tarea especificada
stop nombre_de_tarea	Detiene la tarea especificada
function (argumentos)	Llama una función utilizando los correspondientes argumentos
var = expresión	Evalúa una expresión y la asigna a una variable
var += expresión	Evalúa una expresión y la suma a una variable
var -= expresión	Evalúa una expresión y la resta de una variable
var *= expresión	Evalúa una expresión y la multiplica por una variable
var /= expresión	Evalúa una expresión y divide la variable por ella
var |= expresión	Evalúa una expresión y realiza una comparación OR bit a

	bit con la variable dejando el resultado en esta última
var &= expresión	Evalúa una expresión y realiza una comparación AND bit a bit con la variable dejando el resultado en esta última
return	Devuelve el control de la función al llamador
expresión	Expresión a evaluar

Referencia rápida de Condiciones.-

Condición	Significado
true	Siempre verdadero
false	Siempre falso
expr1 == expr2	Comprueba si las expresiones son iguales
expr1 != expr2	Comprueba si las expresiones son diferentes
expr1 < expr2	Comprueba si una de las expresiones es inferior a la otra
expr1 < = expr2	Comprueba si una expresión es inferior o igual a la otra
expr1 > expr2	Comprueba si una expresión es superior a la otra
expr1 >= expr2	Comprueba si una expresión es superior igual a la otra
! condition	Negación lógica de la condición
cond1 && cond2	AND lógico entre dos condiciones (verdadero si y sólo si las dos condiciones son verdaderas)
cond1 | cond2	OR lógico (excluyente) entre dos condiciones (verdadero si y sólo si al menos una de las condiciones es verdadera)

Referencia rápida de Expresiones.-

Valor	Descripción
Número	Valor constante
Variable	Nombre de una variable
Timer (n)	Valor del temporizador n, donde n está comprendido entre 0 y 3
Random (n)	Número aleatorio entre 0 y n
SensorValue (n)	Valor actual del sensor n, donde n está comprendido entre 0 y 2
Watch ( )	Valor del reloj del sistema
Message ( )	Valor del último mensaje IR recibido

Referencia rápida de Operadores.-

Operador	Descripción	Asociatividad	Restricción	Ejemplo
abs ( ) sign ( )	Valor absoluto Signo de operando	n/a		abs (x) sign (x)
++ --	Incremento, Disminución	Izquierda Izquierda	Sólo variables Sólo variables	x++ o ++x x-- o --x
- ~ !	Menos unario Negación bitwise (unario) Negación lógica	Derecha Derecha Derecha	Sólo constantes	-x ~123 !x
*, /, %	Multiplicación, división, modulo	Izquierda		x * y
+, -	Suma, resta	Izquierda		x + y
<<, >>	Desplazamiento a derecha e izquierda	Izquierda	La medida del desplazamiento ha de ser constante	x<<4
<, > <=, >=	Operadores relacionales	Izquierda		x < y
= =, !=	Igual a, no igual a	Izquierda		x = = 1
&	AND bit a bit	Izquierda		x & y
^	XOR bit a bit	Izquierda		x ^ y
|	OR bit a bit	Izquierda		x | y
&&	AND lógico	Izquierda		x && y
||	OR lógico	Izquierda		x || y
? :	Valor condicional	n/a		x==1 ? y : z

Funciones

Función	Descripción	Ejemplo
SetSensor(sensor, config)	Configura un sensor	SetSensor(SENSOR_1, SENSOR_TOUCH)
SetSensorMode(sensor, modo)	Modo de configuración de sensor	SetSensor(SENSOR_2, SENSOR_MODE_PERCENT)
SetSensorType(sensor, tipo)	Tipo de configuración de sensor	SetSensor(SENSOR_2, SENSOR_TYPE_LIGHT)
ClearSensor(sensor)	Borra el valor de un sensor	ClearSensor(SENSOR_3)
On(salidas)	Conecta una o más salidas	On(OUT_A + OUT_B)
Off(salidas)	Desconecta una o más	Off(OUT_C)

	salidas	
Float(salidas)	Detiene las salidas sin freno	Float(OUT_B)
Fwd(salidas)	Establece las salidas para dirección de avance	Fwd(OUT_A)
Rev(salidas)	Establece las salidas para dirección de retroceso	Rev(OUT_B)
Toggle(salidas)	Invierte la dirección de las salidas	Toggle(OUT_C)
OnFwd(salidas)	Arranca con dirección de	OnFwd(OUT_A)
OnRev(salidas)	Arranca con dirección de retroceso	OnRev(OUT_B)
OnFor(salidas, tiempo)	Arranca por un determinado número de 1/100 de segundo. El tiempo puede ser una expresión.	OnFor(OUT_A)
SetOutput(salidas, modo)	Configura el modo de salida.	SetOutput(OUT_A, OUT_ON)
SetDirection(salidas, dir)	Configura la dirección de salida	SetDirection(OUT_A, OUT_FWD)
SetPower(salidas, potencia)	Configura el nivel de potencia de salida (0-7). La potencia puede ser una expresión.	SetPower(OUT_A, 6)
Wait(time)	Espera durante el tiempo especificado en 1/100 de segundo. El tiempo puede ser una expresión.	Wait(x)
PlaySound(sonido)	Genera el sonido especificado (0-5).	PlaySound(SOUND_CLICK)
PlayTone(frec, duración)	Toca un tono de una especificada frecuencia durante el tiempo especificado (en 1/10 de segundo)	PlayTone(440, 5)
ClearTimer(temporizador)	Reinicia el temporizador (0-3) al valor 0	ClearTimer(0)
StopAllTasks( )	Detiene todas las tareas que se estén ejecutando	StopAllTasks( )
SelectDisplay(modo)	Selecciona uno de los 7 modos de display: 0: reloj	SelectDisplay(1)

	del sistema, 1-3: valor del sensor, 4-6: configuraciones de salidas. Modo puede ser una expresión.	
SendMessage(mensaje)	Envía un mensaje IR (1-255). Mensaje puede ser una expresión.	On(OUT_A + OUT_B)
ClearMessage( )	Borra el buffer de mensajes IR	ClearMessage( )
CreateDatalog(tamaño)	Crea un Nuevo registro de datos del tamaño dado.	CreateDatalog(100)
AddToDatalog(valor)	Añade un valor al registro de datos. El valor puede ser una expresión.	AddToDatalog(Timer (0))
SetWatch(horas, minutos)	Establece el valor del reloj del sistema.	SetWathc(1, 30)
SetTxPower(hi_lo)	Configura la potencia del transmisor de infrarrojos en alta o baja potencia.	SetTxPower(TX_POWER_LO)

Constantes RCX

Configuraciones de sensor para SetSensor( )	SENSOR_TOUCH, SENSOR_LIGHT, SENSOR ROTATION, SENSOR_CELSIUS, SENSOR_FAHRENHEIT, SENSOR_PULSE, SENSOR_EDGE
Modo para SetSensorMode( )	SENSOR_MODE_RAW, SENSOR_MODE_BOOL, SENSOR_MODE_EDGE, SENSOR_MODE_PULSE, SENSOR_MODE_ PERCENT, SENSOR_MODE_CELSIUS, SENSOR_MODE FAHRENHEIT, SENSOR_MODE_ROTATION
Tipo para SetSensorType( )	SENSOR_TYPE_TOUCH, SENSOR_TYPE_TEMPERATURE_ SENSOR_TYPE_LIGHT, SENSOR_TYPE_ROTATION
Salidas para On( ), Off( ), etc	OUT_A, OUT_B, OUT_C
Modos para SetOutput( )	OUT_ON, OUT_OFF, OUT_FLOAT
Direcciones para SetDirection( )	OUT_FWD, OUT_REV, OUT_TOGGLE
Potencia de salida para	OUT_LOW, OUT_HALF, OUT_FULL

SetPower( )	
Sonidos para PlaySound( )	SOUND_CLICK, SOUND_DOUBLE_BEEP, SOUND_DOWN, SOUND_UP, SOUND_LOW_BEEP, SOUND_FAST_UP
Modos para SelectDisplay( )	DISPLAY_WATCH, DISPLAY_SENSOR_1, DISPLAY_SENSOR_2, DISPLAY_SENSOR_3, DISPLAY_OUT_A, DISPLAY_OUT_B, DISPLAY_OUT_C
Nivel de potencia Tx para SetTxPower( )	TX_POWER_LO, TX_POWER_HI

Palabras clave

Existen algunas palabras reservadas por el compilador de NQC para sí mismo. Es un error utilizar cualquiera de estas como nombre de funciones, tareas o variables. Las palabras reservadas son las siguientes;

_sensor,	abs,	asm,	break,	const,
continue,	do,	else,	false,	if,
inline,	int,	repeat,	return,	sign,
start,	stop,	sub,	task,	true,
void,	while.			

Uso de la Cámara Web

Nivel PILOTO.-

Seleccionar la Cámara

ROBOLAB recuerda el tipo de cámara que se ha seleccionado, y lo utiliza cada vez que corre el programa.

 **Configuración de la Cámera**
Es posible ajustar los valores tales como Hue(), Saturation(saturación), y Brightness(brillo), así como el tipo de iluminación de fondo.
La ventana de configuración es diferente para cada tipo de cámara, pero cuentan con caracterísitcas similares.

 **Pausa de la Cámara**
Utilizando el botón de pause puede congelarse la imagen.
Esta imagen permanece hasta que el botón es deseleccionado, de otra forma la imagen es actualizada constantemente por la cámara.

 **Guardar Imagen**
Es posible capturar una imagen. Se abrirá una nueva ventana para definir el nombre y elegir el destino. Se grabará como mapa de bits (bmp) tanto en PC como en Mac.

Nota: en el nivel PILOTO, no hay información de la cámara en el programa ni en el RCX, es decir, solo estará conectada a la PC mediante USB._

Nivel INVENTOR.-

Para seleccionar la cámara en INVENTOR, debe elegirse desde el menú Project / Select Camera.

Hay algunas cosas importantes en el uso de la cámara en INVENTOR, en este nivel el programa y la cámara puede tanto operar completamente independiente o el RCX puede ser programado para responder a lo que la cámara ve. Si el RCX es programado para utilizar la información de la cámara, deberá permanecer dentro del rango del transmisor IR.

El nivel 4 de Inventor puede utilizar la cámara como sensor tanto como una entrada visual.

Accesando a la Cámara en Inventor
Para utilizar la cámara en Inventor, se necesita la ventana del Centro de Visión, la cual se abre desde el menú Project / Vision Center.

Centro de Visión
La ventana centro de visión provee un formato para ver la imagen de la cámara y para configurarla como un sensor que provee entrada al programa RCX. El centro de visión tiene tres velocidades de comunicación: Sin Conexión con el RCX, Modo de Conexión Lenta y Modo de Conexión Rápida.

Velocidad de Comunicación

La velocidad de Comunicación de datos entre el sensor de la cámara y la computadora dependen de la complejidad del sensor de la cámara, la velocidad del CPU, el número de ubicaciones del sensor de cámara, y la comunicación del transmisor IR.

Debe seleccionarse el tipo de comunicación apropiado para las necesidades de programación.

Sin Conexión con el RCX
Este modeo de operación no involucra comunicación entre el RCX y la computadora. En este modo la actualización de la imagen es más rápida.

Modo Conexión Lenta
Este modo de operación es usado para enviar los paquetes de valores especificos en la configuración del sensor de la cámara al RCX. En esta configuración, el RCX puede pedir a la computadora grabar una imagen.

Modo Conexión Rápida
Este modo de operación es usado para enviar continuamente los paquetes de valores al RCX. En esta configuración, el RCX no pide a la computadora grabar una imagen, incrementando el nivel de la tranferencia de datos.

Prácticas Propuestas

A continuación se presentan ejemplos prácticos para utilizar lo explicado anteriormente a fin de que el manejo y programación del kit LEGO^{MR} 9794 se domine. De esta forma se tiene la capacidad de realizar diversas tareas y aplicaciones, que pueden surgir de un desafío o como solución a una problemática en particular.

Las primeras diez están enfocadas al uso del lenguaje gráfico ROBOLAB y las otras diez se adentrarán en la esctructura de programación y control del robot mediante el lenguaje en código NQC.

Cabe mencionar que estas prácticas son ejemplos básicos los que permitirán adquirir habilidades para posteriormente realizar construcciones, programas y robots mucho más complejos.

PRACTICA 1

Movimiento hacia delante

Objetivo:

El alumno identificará la programación gráfica mediante ROBOLAB, al programar el mini-robot para que se desplace hacia adelante.

Teoría preliminar:

La programación mediante ROBOLAB es basada en LabVIEW y se presenta en un entorno gráfico, se deberán conocer los íconos y sus respectivas tareas así como las variables que cada uno presenta. Para una información más detallada consulta la parte de Generalidades.

Equipo Requerido:

1 Kit LEGO 9794

Procedimiento:

- Construir el básico ***Driving Base o Taskbot*.** (Pags. 16-21)
- En el RCX, programar la secuencia mostrada a continuación.
- Cargar el programa en el cerebro.
- Correr el programa y analiza su comportamiento.

Esquema gráfico del Programa:

Fig. 1-1 Programa de movimiento hacia el frente.

Problemas Propuestos:

- Pensando en el comportamiento obtenido, deberá modificarse el programa para que avance el triple de la distancia que recorrió originalmente.
- Ahora cambiar el programa para que se mueva hacia atrás en lugar de hacia adelante, pero solo la mitad de la distancia.
- ¿Puede generarse un programa para que avance 20 cm, retroceda 10 cm y avance nuevamente 10 cm?
- Proponer una aplicación a este tipo de funcionamientos.
- Plantear un funcionamiento distinto, generar su programa y entregar su justificación.

Producto:

- Deberá mostrarse el mini-robot funcionando.
- Entregar los esquemas en archivo a más tardar la próxima sesión y conservarlos para impresión en el Trabajo Integrador al final de la Unidad.

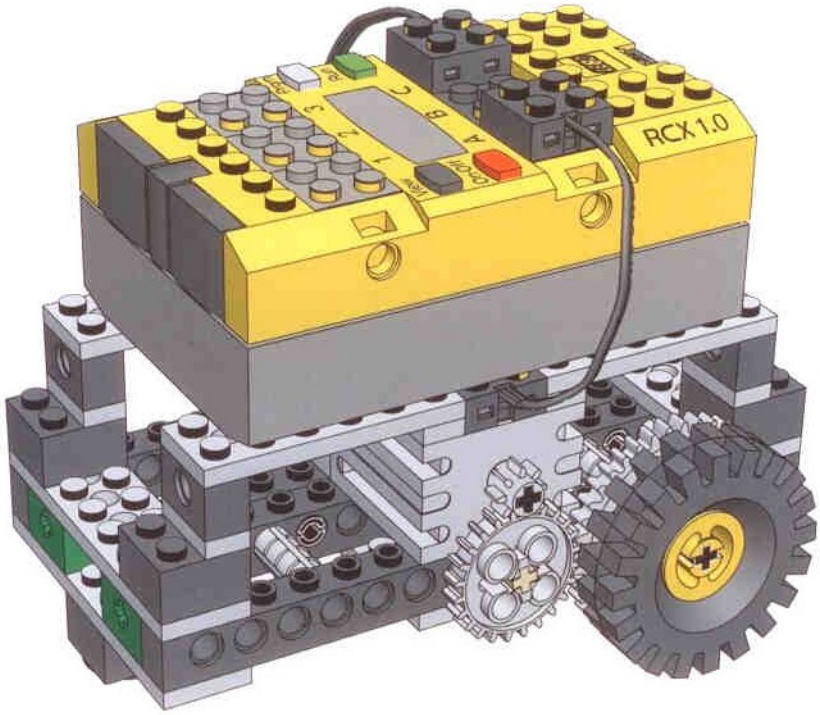


Fig 1-2 Driving Base

PRACTICA 2

Ajuste de Posición

Objetivo:

El alumno programará al mini-robot para que cambie de orientación, dependiendo de su posición original.

Teoría preliminar:

La orientación es una de las partes primordiales de la Robótica, la cual puede ser previamente programada o dependiente de alguna señal adquirida, en esta práctica se programará un cambio en la posición del robot.

En la Robótica móvil es de suma importancia el poder determinar la ubicación final del robot lo que se consigue mediante sensores, y giros correcto de los motores.

Equipo Requerido:

1 Kit LEGO 9794

Procedimiento:

- Construir el básico ***Driving Base o Taskbot*.**
- En el ROBOLAB, programar la secuencia mostrada a continuación.
- Cargar el programa en el cerebro.
- Correr el programa y analiza su comportamiento.
- Realizar los desafíos que solicitan.

Esquema gráfico del Programa:

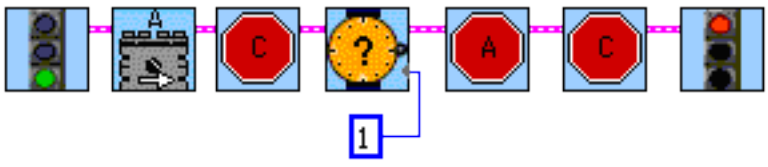


Fig. 2-1 Programa de movimiento para cambiar orientación.

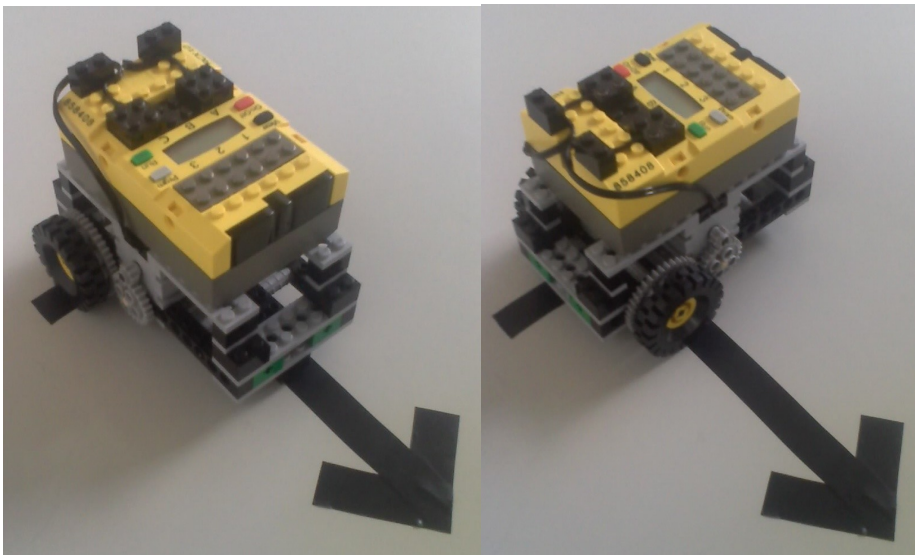


Fig. 2-2 Posición original (A) y Posición una vez que ha girado (B).

Problemas Propuestos:

- En la figura 2-2, se muestran las posiciones A y B:
- Debe modificarse el programa indicado en la figura 2-1 para que el robot gire desde A hacia B primero hacia la derecha. Y posteriormente para que gire de A hacia B girando hacia la izquierda.
- Ahora cambiar el programa para que haga los giros mientras se mueve hacia atrás.
- ¿Puede generarse un programa para que al llegar al punto B, espere 10 segundos y gire nuevamente hacia el punto A?
- Proponer una aplicación a este tipo de funcionamientos.
- Plantear un funcionamiento distinto, generar su programa y entregar su justificación.

Producto:

- Deberá mostrarse el mini-robot funcionando.
- Entregar los esquemas en archivo a más tardar la próxima sesión y conservarlos para impresión en el Trabajo Integrador al final de la Unidad.

PRACTICA 3

Movimiento y paro mediante sensor de Tacto

Objetivo:

El alumno analizará el funcionamiento de los sensores de proximidad, detectando obstáculos mediante tacto, a la vez que propone aplicaciones para los mismos.

Teoría preliminar:

Una de las partes fundamentales de la Robótica son los sensores ya que es mediante estos elementos que el sistema en contacto con el mundo exterior y permite al programador desarrollar respuestas a partir de las señales adquiridas. Para una información más detallada consulta la parte correspondiente a sensores.

Equipo Requerido:

1 Kit LEGO 9794
2 sensores de tacto

Procedimiento:

- Construir el básico ***Driving Base o Taskbot*.**
- En el ROBOLAB, programar la secuencia mostrada a continuación.
- Cargar el programa en el cerebro.
- Correr el programa y analiza su comportamiento.
- Realizar los desafíos que solicitan.

Esquema gráfico del Programa:

El programa es sencillo, simplemente para probar el sensor de tacto.

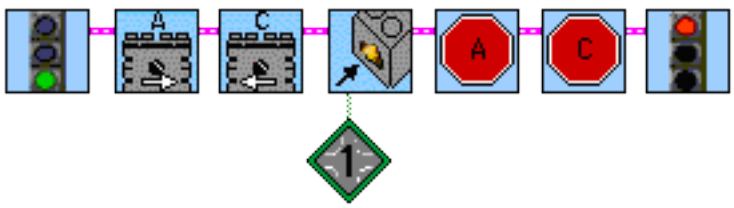


Fig. 3-1 Programa para detener los motores mediante sensor de tacto.

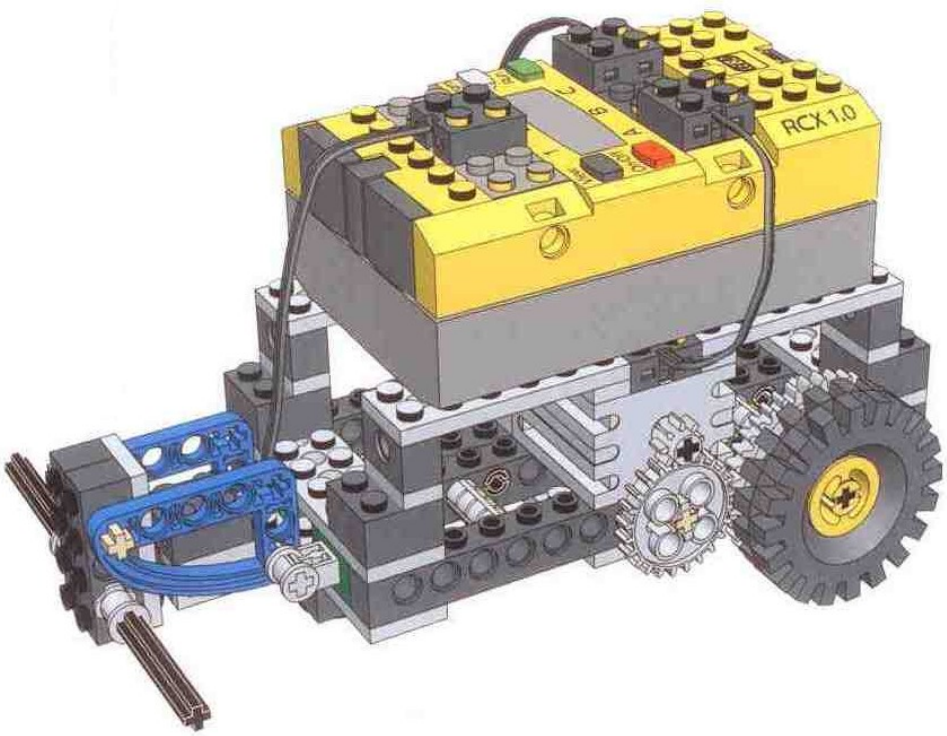


Fig. 3-2 Configuración para adicionar el sensor de Tacto

Problemas Propuestos:

- Debe analizarse el cambio en el funcionamiento cuando se modifica el programa de avance, para que se mueva hacia delante hasta que encuentre algún obstáculo, una vez que lo detecte, se detenga, retroceda un poco y gire.
- Adaptar el ***Driving Base*** con 2 sensores de tacto y generar un programa para que detecte utilizando dos puertos, tanto en la parte frontal como posterior.
- Proponer una aplicación distinta a la anterior, generar un programa y entregar su justificación.

Producto:

- Deberá mostrarse el mini-robot funcionando.
- Entregar los esquemas en archivo a más tardar la próxima sesión y conservarlos para impresión en el Trabajo Integrador al final de la Unidad.

PRACTICA 4

Movimiento y paro mediante sensor de Temperatura

Objetivo:

El alumno analizará el funcionamiento de los sensores de temperatura, utilizará dos tipos de unidades, y propondrá aplicaciones para los mismos.

Teoría preliminar:

Una de las partes fundamentales de la Robótica son los sensores ya que es mediante estos elementos que el sistema en contacto con el mundo exterior y permite al programador desarrollar respuestas a partir de las señales adquiridas. El sensor de temperatura puede tener modo RAW además generar lecturas en Celcius y Fahrenheit. Para una información más detallada consulta la parte correspondiente a sensores.

Equipo Requerido:

1 Kit LEGO 9794
1 sensor de temperatura

Procedimiento:

- Construir el básico ***Driving Base o TaskBot*.**
- En el ROBOLAB, programar la secuencia mostrada a continuación.
- Cargar el programa en el cerebro.
- Correr el programa y analiza su comportamiento.
- Realizar los desafíos que solicitan.

Esquema gráfico del Programa:

El programa es para probar el sensor de temperatura.

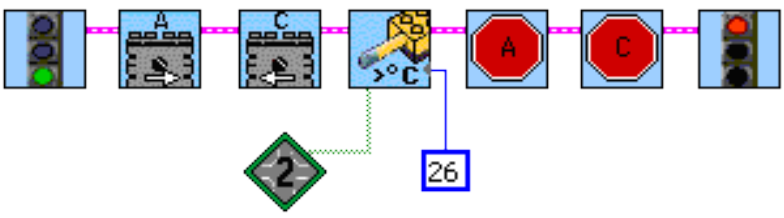


Fig. 4-1 Programa para detener los motores mediante sensor de temperatura.

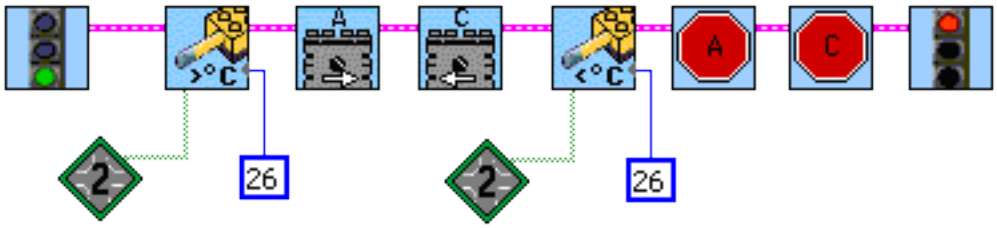


Fig. 4-2 Programa de movimiento y paro mediante sensor de Temperatura.

Problemas Propuestos:

- Deberá analizarse el cambio en el funcionamiento cuando se hace la modificación al programa, explica los cambios y como se indican mediante los íconos.
- Generar un programa para que al detectar una temperatura elevada, el robot se detenga, retroceda un poco, gire 180° y se mueva hacia delante otra vez hasta que detecte una temperatura menor; cuando la temperatura aumente de nuevo, el robot debe realizar el proceso anterior, y así sucesivamente.
- Proponer una aplicación distinta a la anterior, generar un programa y entregar su justificación.

Producto:

- Deberá mostrarse el mini-robot funcionando.
- Entregar los esquemas en archivo a más tardar la próxima sesión y conservarlos para impresión en el Trabajo Integrador al final de la Unidad.

PRACTICA 5

Movimiento y paro mediante sensor de Luz

Objetivo:

El alumno analizará el funcionamiento del sensor óptico, sus estados activo y pasivo, a la vez que propone aplicaciones para los mismos.

Teoría preliminar:

Una de las partes fundamentales de la Robótica son los sensores ya que es mediante estos elementos que el sistema en contacto con el mundo exterior y permite al programador desarrollar respuestas a partir de las señales adquiridas. El sensor de luz puede funcionar en modo activo o pasivo, es decir, puede emitir y recibir la señal, o simplemente ser receptor. Para una información más detallada consulta la parte correspondiente a sensores.

Equipo Requerido:

1 Kit LEGO 9794
1 sensor de luz

Procedimiento:

- Construir el básico ***Driving Base*.**
- En el RCX, programar la secuencia mostrada a continuación.
- Cargar el programa en el cerebro.
- Correr el programa y analiza su comportamiento.
- Realizar los desafíos que solicitan.

Esquema gráfico del Programa:

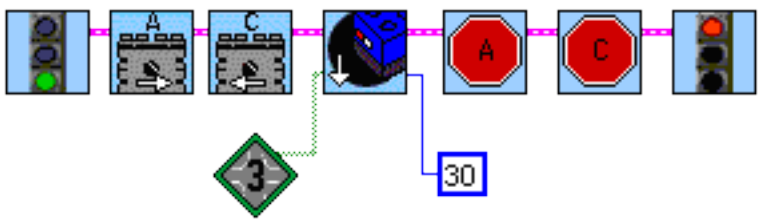


Fig. 5-1 Programa para detener los motores mediante sensor de luz (modo Activo).

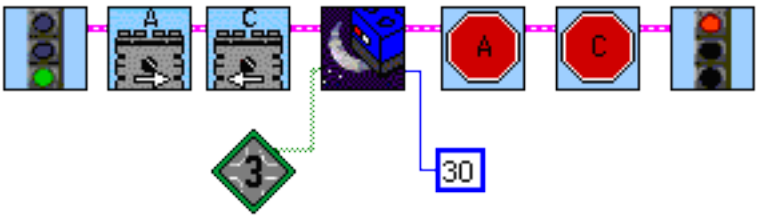


Fig. 5-2 Programa de movimiento y paro mediante sensor de luz (modo Pasivo).

Problemas Propuestos:

- Generar un programa que detecte un cambio de color en el piso (puede ser de blanco a negro o cualquier color), para entonces retroceder un poco y girar.
- Trazar un cuadrado amplio con cinta y dejar al mini-robot dentro funcionando, la idea es que no salga del margen establecido.
- Proponer una aplicación distinta a la anterior, generar un programa y entregar su justificación.

Producto:

- Deberá mostrarse el mini-robot funcionando.
- Entregar los esquemas en archivo a más tardar la próxima sesión y conservarlos para impresión en el Trabajo Integrador al final de la Unidad.

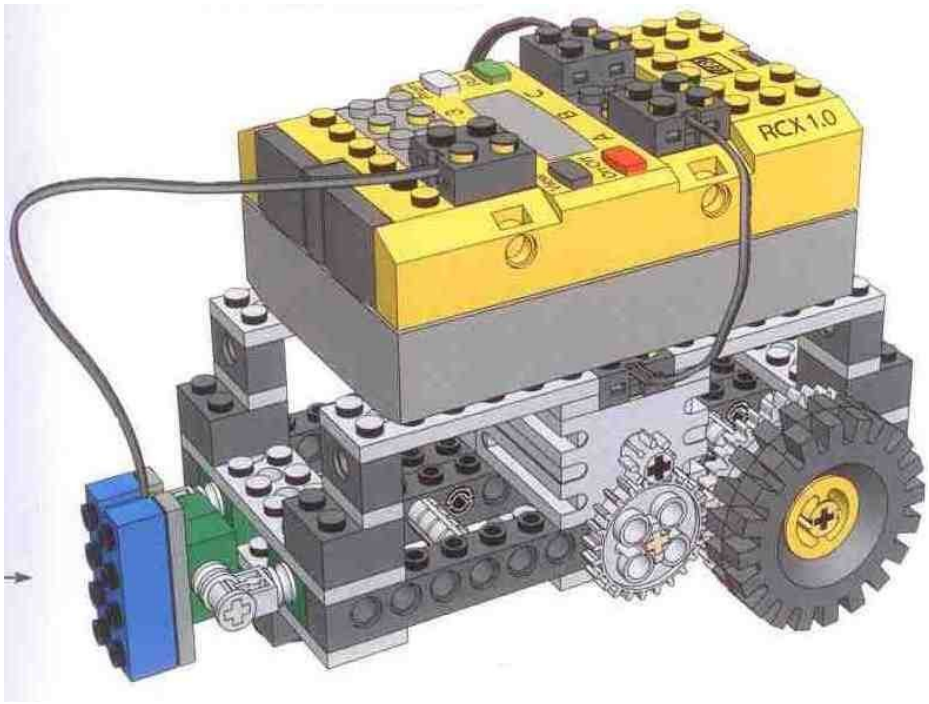


Fig. 5-3 Mini-Robot con el sensor de luz acoplado al frente.

PRACTICA 6

Movimiento y paro mediante sensor de Giro

Objetivo:

El alumno analizará el funcionamiento del sensor de giro, a la vez que propone aplicaciones para los mismos.

Teoría preliminar:

Una de las partes fundamentales de la Robótica son los sensores ya que es mediante estos elementos que el sistema en contacto con el mundo exterior y permite al programador desarrollar respuestas a partir de las señales adquiridas. Para una información más detallada consulta la parte correspondiente a sensores.

Equipo Requerido:

1 Kit LEGO 9794
1 sensor de giro

Procedimiento:

- Construir el básico ***Driving Base*.**
- En el RCX, programar la secuencia mostrada a continuación.
- Cargar el programa en el cerebro.
- Correr el programa y analiza su comportamiento.
- Realizar los desafíos que solicitan.

Esquema gráfico del Programa:

El siguiente programa es para probar y calibrar el sensor de giro.

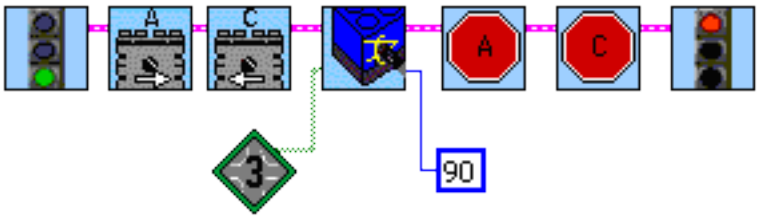


Fig. 6-1 Programa para detener los motores mediante sensor de giro.

Problemas Propuestos:

- Deberá analizarse el cambio en el funcionamiento conforme se modifican los valores de giro en el sensor.
- Generar un programa que mida los grados avanzados al hacer que el robot gire una vuelta completa sobre sí mismo.
- Proponer una aplicación distinta a la anterior, generar un programa y entregar su justificación.

Producto:

- Deberá mostrarse el mini-robot funcionando.
- Entregar los esquemas en archivo a más tardar la próxima sesión y conservarlos para impresión en el Trabajo Integrador al final de la Unidad.

PRACTICA 7

Ciclos repetitivos

Objetivo:

El alumno utilizará ciclos en diferentes programas a fin de analizar la recursividad y utilidad de los mismos.

Teoría preliminar:

Para la programación en ROBOLAB los ciclos(LOOPS) se forman por saltos, los que indican donde comienza y termina la parte a repetir, también es posible indicar el número de veces que deberán realizarse los saltos, una vez que termine el conteo el programa seguirá con la siguiente instrucción fuera del ciclo.

Equipo Requerido:

1 Kit LEGO 9794

Procedimiento:

- Construir el básico ***Driving Base o Taskbot*.**
- En el RCX, programar la secuencia mostrada a continuación.
- Cargar el programa en el cerebro.
- Correr el programa y analiza su comportamiento.
- Realizar los desafíos que solicitan.

Esquema gráfico del Programa:

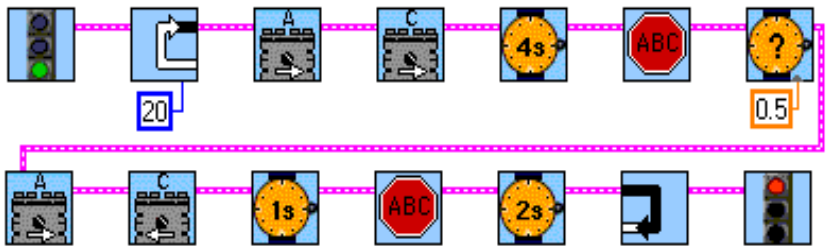


Fig. 7-1 Programa de accionamiento y paro en ciclo repetitivo.

Problemas Propuestos:

- Debe analizarse el funcionamiento del programa anterior..
- Generar programas con ciclos recursivos para los ejemplos de prácticas anteriores.
- Proponer una aplicación distinta a las anteriores, generar un programa y entregar su justificación.

Producto:

- Deberá mostrarse el mini-robot funcionando.
- Entregar los esquemas en archivo a más tardar la próxima sesión y conservarlos para impresión en el Trabajo Integrador al final de la Unidad.

PRACTICA 8

Seguidor de línea

Objetivo:

El alumno utilizará un programa seguidor de línea, con uno o dos sensores ópticos, así como las condiciones lógicas que esto implica.

Teoría preliminar:

Para la programación de un seguidor de línea deben tenerse en cuenta las condiciones lógicas que se generan de los posibles juegos de lecturas obtenidas por los sensores de luz.

Equipo Requerido:

1 Kit LEGO 9794
1 ó 2 sensores de luz

Procedimiento:

- Construir el básico ***Driving Base o Taskbot*.**
- En el RCX, programar la secuencia mostrada a continuación.
- Cargar el programa en el cerebro.
- Correr el programa y analiza su comportamiento.
- Realizar los desafíos que solicitan.

Esquema gráfico del Programa:

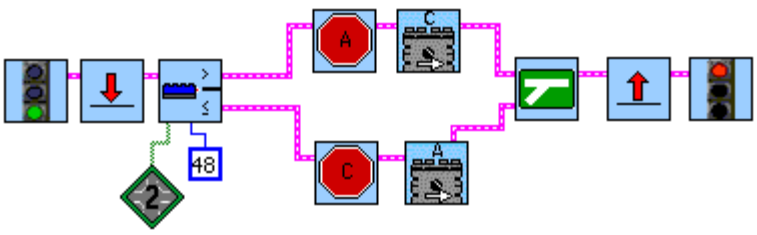


Fig. 8-1 Programa seguidor de línea, con un sensor.

Problemas Propuestos:

- Debe analizarse el funcionamiento del programa anterior.
- Generar un programa para que siga la línea utilizando dos sensores de luz.
- Proponer una aplicación distinta a las anteriores, generar un programa y entregar su justificación.

Producto:

- Deberá mostrarse el mini-robot funcionando.
- Entregar los esquemas en archivo a más tardar la próxima sesión y conservarlos para impresión en el Trabajo Integrador al final de la Unidad.

PRACTICA 9

Uso de la cámara

Objetivo:

El alumno utilizará la cámara web con el RCX a través del ROBOLAB. Se utilizará la cámara para contar cuantos puntos muestra un dado después de haber rodado y el RCX producirá un sonido el número de veces correspondiente al número mostrado.

Teoría preliminar:

El centro de visión en ROBOLAB muestra la imagen actual cualdo el plano de color es seleccionado. La imagen procesada para este ejemplo, se amacenará en el contenedor rojo. El valor se muestra a un lado. Si el contenedor seleccionado, la imagen en la ventana es procesada para dicho contenedor.

En esta imagen los tres puntos son visibles. Es importante tener en cuenta que si se esta utilizando la cámara como un sensor de entrada, el RCX debe estar en el rango del transmisor IR.
La cámara es un sensor interesante y único. La imagen puede procesarse en la PC para generar un valor. Este valor es un número el cual puede ser usado de varias formas. Los valores se almacenan en los contenedores rojo, azul y amarillo. Estos valores que fueron determinados en la computadora son enviados al RCX, el RCX puede entonces actualizar los valores en el programa que este ejecutando. Esto permite una respuesta en tiempo real del RCX a la entrada visual de la cámara.

Imagen de la Cámara

Imagen Procesada

Equipo Requerido:

1 Kit LEGO 9794
1 cámara web

Procedimiento:

- Construir el básico ***Driving Base o Taskbot*.**
- En el RCX, programar la secuencia mostrada a continuación.
- Cargar el programa en el cerebro.
- Correr el programa y analiza su comportamiento.
- Realizar los desafíos que solicitan.

Esquema gráfico del Programa:

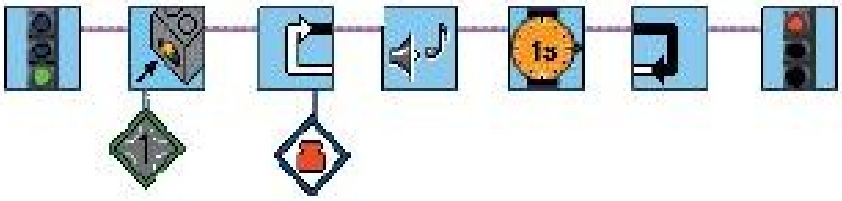


Fig. 9-1 Programa para contar puntos en la superficie de un dado.

Problemas Propuestos:

- Debe analizarse el funcionamiento del programa.
- Proponer una aplicación distinta a las anteriores, generar un programa y entregar su justificación.

Producto:

- Deberá mostrarse el mini-robot funcionando.
- Entregar los esquemas en archivo a más tardar la próxima sesión y conservarlos para impresión en el Trabajo Integrador al final de la Unidad.

PRACTICA 10

Esquivar obstáculos

Objetivo:

El alumno utilizará la cámara web con el RCX a través del ROBOLAB, para determinar cual parte del camino esta bloqueado y hacer que el RCX logre esquivarlos.

Teoría preliminar:

Si el procesao de imagen es configurado para asumir que los obstáculos son oscuros, la imagen es procesada para ver si el área con mayor iluminación es la de la izquierda, centro o derecha. Estos valores son enviados como valores de contenedor. Cuando el RCX recibe el nuevo valor del sensor, el programa que esta corriendo directamente puede ajustar la dirección del movimiento para que el RCX se dirija al área más iluminada.

Para lograr este funcionamiento deberá realizarse la programación en ROBOLAB en Nivel INVENTOR.

Equipo Requerido:

1 Kit LEGO 9794
1 cámara web

Procedimiento:

- Construir el básico ***Driving Base o Taskbot*.**
- En el RCX, programar la secuencia mostrada a continuación.
- Cargar el programa en el cerebro.
- Correr el programa y analiza su comportamiento.
- Realizar los desafíos que solicitan.

Problemas Propuestos:

- Debe analizarse el funcionamiento del programa.
- Proponer una aplicación distinta a las anteriores, generar un programa y entregar su justificación.

Esquema gráfico del Programa:

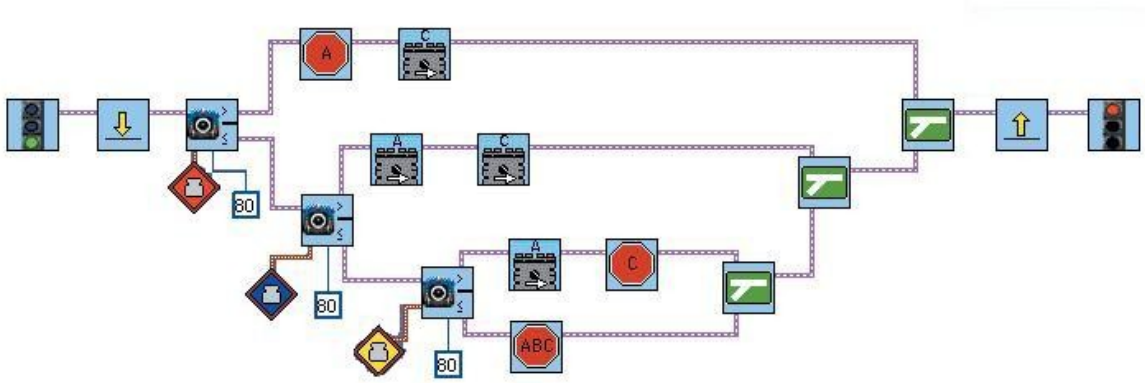


Fig. 10-1 Programa para esquivar obstáculos.

Producto:

- Deberá mostrarse el mini-robot funcionando.
- Entregar los esquemas en archivo a más tardar la próxima sesión y conservarlos para impresión en el Trabajo Integrador al final de la Unidad.

PRACTICA 11

Movimiento hacia adelante y atrás

Objetivo:

El alumno identificará la programación mediante NQC, al programar en lenguaje código el mini-robot para que se desplace hacia adelante y hacia atrás.

Teoría preliminar:

En la programación NQC se define la tarea a realizar mediante comandos o palabras definidas, también debe definirse quien realiza la acción en este caso los motores, los que se manejan como salidas OUT_A y OUT_C, si se requiere pueden escribirse en una sola instrucción, OUT_A+OUT_C. Deberá especificarse el tiempo de espera o el número de rotaciones, dependiendo la secuencia a realizar. Para más detalles de las estructuras y comandos básicos consulta la parte de Generalidades.

Equipo Requerido:

1 Kit LEGO 9794

Procedimiento:

- Construir **TASKBOT** básico.
- En el BricxCC, programar la secuencia mostrada a continuación.
- Cargar el programa en el cerebro RCX.
- Ejecutar el programa y analizar su comportamiento.

Código del Programa:

Primer programa

```
task main()
{
      OnFwd(OUT_A);
      OnFwd(OUT_C);
      Wait(400);
      OnRev(OUT_A+OUT_C);
      Wait(400);
      Off(OUT_A+OUT_C);
}
```

Segundo programa

```
task main()
{
      SetPower(OUT_A+OUT_C,2);
      OnFwd(OUT_A+OUT_C);
      Wait(400);
      OnRev(OUT_A+OUT_C);
      Wait(400);
      Off(OUT_A+OUT_C);
}
```

Este programa modifica la velocidad de los motores, la potencia es un valor numérico del 0 al 7.

Problemas Propuestos:

- A continuación, se deberá modificar el programa, para que avance el triple de la distancia que recorre el primer programa.
- Cambiar el programa para que se mueva hacia atrás en lugar de hacia adelante, pero solo la mitad de la distancia original.
- ¿Puede generarse un programa para que avance 20 cm, retroceda 10 cm y avance nuevamente 10 cm?
- Proponer una aplicación a este tipo de ejemplos.
- Plantear un ejercicio distinto, generar su programa y entregar su aplicación.

Producto:

- Debe mostrarse el mini-robot funcionando.
- Entregar los códigos en archivo a más tardar la próxima sesión y conservarlos para impresión en el Trabajo Integrador al final de la Unidad.

PRACTICA 12

Ciclo repetitivo

Objetivo:

El alumno programará en lenguaje NQC ciclos que se repitan, para realizar funciones específicas tales como giros y trayectorias con el Mini-Robot.

Teoría preliminar:

Definir variables es fundamental en la programación de cualquier lenguaje código, el tener que aplicar ciertas operaciones en varias ocasiones, puede evitarse si se nombran los resultados, ya que pueden llamarse más adelante, permitiendo agilizar el proceso y la ubicación de las partes dentro del programa, además de que sirven como referencia para el programador.

El generar ciclos que pueden repetirse, permite llevar a cabo tareas mientras se incrementan las variables hasta obtener un valor deseado.

Equipo Requerido:

1 Kit LEGO 9794

Procedimiento:

- Construir **TASKBOT** básico.
- En el BricxCC, programar la secuencia mostrada a continuación.
- Cargar el programa en el cerebro RCX.
- Ejecutar el programa y analizar su comportamiento.
- Realizar los problemas propuestos que se solicitan.

Código del Programa:

Primer programa

```
#define TIEMPO_AVANCE 100      // define la variable con nombre y valor

#define TIEMPO_GIRO 80


task main()
{
      OnFwd(OUT_A+OUT_C);
```

```
        Wait(TIEMPO_AVANCE);  // usa la variable para detenerse
        OnRev(OUT_C);
        Wait(TIEMPO_GIRO);
        Off(OUT_A+OUT_C);
    }
```

Segundo programa

```
#define TIEMPO_AVANCE 100      // define la variable con nombre y valor
#define TIEMPO_GIRO 80

    task main()
    {
        repeat(4)              // comienza un ciclo que se repetiá 4 veces
        {
        OnFwd(OUT_A+OUT_C);
        Wait(TIEMPO_AVANCE);
        OnRev(OUT_C);
        Wait(TIEMPO_GIRO);
        }
    Off(OUT_A+OUT_C);
    }
```

Problemas Propuestos:

- A continuación, se deberá modificar el primer programa. Para que el robot avance 20 cm y realice un giro de 180 grados hacia la izquierda, avance otros 20 cm y gire de 90 hacia la derecha.
- Modificar ahora el programa para que haga los giros mientras avanza hacia atrás.
- Generar un programa para que el Mini-Robot se mueva en un cuadrado perfecto, repitiendo el ciclo 5 veces.
- Plantear un ejercicio distinto, generar el programa y entregar su aplicación.

Producto:

- Debe mostrarse el mini-robot funcionando.
- Entregar los códigos en archivo a más tardar la próxima sesión y conservarlos para impresión en el Trabajo Integrador al final de la Unidad. mostrar el mini-robot funcionando.

PRACTICA 13

Haciendo espirales

Objetivo:

El alumno programará en lenguaje código NQC espirales que se repitan, y manejará variables aleatorias con el Mini-Robot.

Teoría preliminar:

Es común definir variables, pero ¿qué sucede cuando le pedimos al robot que genere valores aleatorios?, el programa se portará de una forma que no esperamos, o al menos que no podemos predecir.

Equipo Requerido:

1 Kit LEGO 9794

Procedimiento:

- Construir **TASKBOT** básico.
- En el BricxCC, programar la secuencia mostrada a continuación.
- Cargar el programa en el cerebro RCX.
- Ejecutar el programa y analizar su comportamiento.
- Realizar los problemas propuestos que se solicitan.

Código del Programa:

Primer programa

```
#define TIEMPO_GIRO 85

int tiempo_avance;  // define la variable de tipo entero
task main()
{
      tiempo_avance = 20;  // indica el valor inicial
      repeat(50)
      {
      OnFwd(OUT_A+OUT_C);
      Wait(tiempo_avance);  // usa la variable para detenerse
      OnRev(OUT_C);
```

```
      Wait(TIEMPO_GIRO);
      move_time += 5; // incrementa la variable
      }
Off(OUT_A+OUT_C);
}
```

Segundo programa

```
int tiempo_avance, tiempo_giro;
task main()
{
      while(true)
      {
      tiempo_avance = Random(60);
      tiempo_giro = Random(40);
      OnFwd(OUT_A+OUT_C);
      Wait(tiempo_avance);
      OnRev(OUT_A);
      Wait(tiempo_giro);
      }
}
```

Problemas Propuestos:

- Debe analizarse el segundo programa y aplicar este tipo de comportamiento al primer programa.
- Proponer una aplicación a este tipo de ejercicios, y entregar su justificación.

Producto:

- Debe mostrarse el mini-robot funcionando.
- Entregar los códigos en archivo a más tardar la próxima sesión y conservarlos para impresión en el Trabajo Integrador al final de la Unidad.

PRACTICA 14

Control y Lógica

Objetivo:

El alumno programará activicades utilizando funciones lógicas en lenguaje código NQC que se repitan, además de que manejará variables aleatorias con el Mini-Robot.

Teoría preliminar:

Las secuencias pueden programarse mediante ciclos y funciones lógicas, aquí se presenta unas de las principales, las condiciones (if-else), (while), (do) las cuales permiten desarrollar actividades cada vez más complejas.

Equipo Requerido:

1 Kit LEGO 9794

Procedimiento:

- Construir **TASKBOT** básico.
- En el BricxCC, programar la secuencia mostrada a continuación.
- Cargar el programa en el cerebro RCX.
- Ejecutar el programa y analizar su comportamiento.
- Realizar los problemas propuestos que se solicitan.

Código del Programa:

Primer programa

```
#define TIEMPO_AVANCE 100
#define TIEMPO_GIRO 85

     task main()
     {

          while(true)
          {
               OnFwd(OUT_A+OUT_C);
               Wait(TIEMPO_AVANCE);
               if (Random(1) >= 0)
               {
                    OnRev(OUT_C);
```

```
                }
                else
                {
                      OnRev(OUT_A);
                }
                Wait(TIEMPO_GIRO);
          }
     }
```

Segundo programa

```
int tiempo_avance, tiempo_giro, tiempo_total;
     task main()
     {
          tiempo_total = 0;
          do
     {
          tiempo_avance = Random(100);
          tiempo_giro = Random(100);
          OnFwd(OUT_A+OUT_C);
          Wait(tiempo_avance);
          OnRev(OUT_C);
          Wait(tiempo_giro);
          tiempo_total += tiempo_avance;
          tiempo_total += tiempo_giro;
          }
     while (tiempo_total < 2000);
     Off(OUT_A+OUT_C);
}
```

Problemas Propuestos:

- Deberán analizarse estos programas y aplicar este tipo de estructura de código a programas generados anteriormente en ROBOLAB, por lo menos a dos de ellos.
- Proponer una aplicación a este tipo de ejemplos, y entregar su aplicación.

Producto:

- Debe mostrarse el mini-robot funcionando.
- Entregar los códigos en archivo a más tardar la próxima sesión y conservarlos para impresión en el Trabajo Integrador al final de la Unidad.

PRACTICA 15

Sensor de Tacto

Objetivo:

El alumno programará en lenguaje código NQC el uso del sensor de tacto para el Mini-Robot.

Teoría preliminar:

Para el lenguaje código NQC se deben indicar los puertos donde están conectados los sensores, se especifican como SENSOR_1, SENSOR_2 y finalmente SENSOR_3. En este caso el sensor de tacto estará conectado en el puerto 1.

Equipo Requerido:

1 Kit LEGO 9794

Procedimiento:

- Construir **TASKBOT** básico.
- En el BricxCC, programar la secuencia mostrada a continuación.
- Cargar el programa en el cerebro RCX.
- Ejecutar el programa y analizar su comportamiento.
- Realizar los problemas propuestos que se solicitan.

Código del Programa:

Primer programa

```
task main()
{
      SetSensor(SENSOR_1,SENSOR_TOUCH);
      OnFwd(OUT_A+OUT_C);
      until (SENSOR_1 == 1);
      Off(OUT_A+OUT_C);
}
```

Segundo programa

```
task main()
{
      SetSensorTouch(SENSOR_1);
      OnFwd(OUT_A+OUT_C);
      while (true)
            {
            if (SENSOR_1 == 1)
            {
            OnRev(OUT_A+OUT_C); Wait(30);
            OnFwd(OUT_A); Wait(30);
            OnFwd(OUT_A+OUT_C);
            }
      }
}
```

Problemas Propuestos:

- Analizar estos programas y aplicar este tipo de estructuras de código a las actividades realizadas previamente con el sensor de tacto programado para ROBOLAB, por lo menos a dos de ellos.
- Proponer una aplicación a este tipo de ejemplos, entregar la justificación.

Producto:

- Deberá mostrarse el mini-robot funcionando.
- Generar el código en NQC de los programas realizados anteriormente para ROBOLAB.
- Entregar los esquemas en archivo a más tardar la próxima sesión y conservarlos para impresión en el Trabajo Integrador al final de la Unidad.

PRACTICA 16

Sensor de Temperatura

Objetivo:

El alumno programará en lenguaje código NQC el uso del sensor de temperatura y almacen de datos para el Mini-Robot.

Teoría preliminar:

El modelo RXC permite guardar datos en una lista, la que será accesible desde la PC, en ella se pueden almacenar datos del sensor de temperatura o de los temporizadores. Esto es posible creando el DATALOG, el cual podrá ser alimentado en el transcurso de la ejecución del programa. Para accesar a los datos almacenados debe abrirse el apartado TOOLS en BricxCC.

Equipo Requerido:

1 Kit LEGO 9794
1 sensor de temperatura

Procedimiento:

- Construir **TASKBOT** básico.
- En el BricxCC, programar la secuencia mostrada a continuación.
- Cargar el programa en el cerebro RCX.
- Ejecutar el programa y analizar su comportamiento.
- Realizar los problemas propuestos que se solicitan.

Código del Programa:

Primer programa

```
#define EXP_TIME 600

task main()
      {
      SetSensorType(SENSOR_2, SENSOR_TYPE_TEMPERATURE);
      SetSensorMode(SENSOR_2, SENSOR_MODE_RAW);
      CreateDatalog(600);
      PlaySound(0);
        repeat (EXP_TIME)
        {
            AddToDatalog(SENSOR_2);
```

```
        Wait(100);        // medición en centésimas de segundo
    }
  PlaySound(0);
  }
```

Problemas propuestos:

- Analizar este programas y modificar el progtrama para que tome lecturas en Celcius y posteriormente en Farenheit.
- Proponer una aplicación a este tipo de ejemplo, y entregar la justificación.

Producto:

- Deberá mostrarse el mini-robot funcionando.
- Generar el código de los programas en NQC.
- Entregar los esquemas en archivo a más tardar la próxima sesión y conservarlos para impresión en el Trabajo Integrador al final de la Unidad.

PRACTICA 17

Sensor Proximidad

Objetivo:

El alumno programará en lenguaje código NQC el uso de los sensores de luz y el puerto infrarrojo a fin de obtener el funcionamiento de un sensor de proximidad para el Mini-Robot.

Teoría preliminar:

Hasta el momento no hay en el mercado sensor ultrasónico para el modelo RCX, solo para el NXT, pero puede generarse un arreglo mendiante el sensor de luz (en modo raw) y el puerto infrarrojo del RCX, el programa se describe a continuación.

Equipo Requerido:

1 Kit LEGO 9794
1 sensor luz

Procedimiento:

- Construir **TASKBOT** básico. **El sensor de luz deberá estar encima del puerto IR apuntando hacia adelante.**
- En el BricxCC, programar la secuencia mostrada a continuación.
- Cargar el programa en el cerebro RCX.
- Ejecutar el programa y analizar su comportamiento.
- Realizar los problemas propuestos que se solicitan.

Código del Programa:

Primer programa

```
int nivelanterior;          //Almacena el nivel previo

task envia_senal()          //No utilizaremos la ñ, para evitar
                            conflictos en el programa
{
     while(true)
     {SendMessage(0); Wait(10);}
}
```

```
task revisa_senal()
{
     while(true)
     {
     nivelanterior = SENSOR_2;
     if(SENSOR_2 > nivelanteior+200)
     {OnRev(OUT_C); Wait(85); OnFwd(OUT_A+OUT_C);}
}
}
task main(){
     SetSensorType(SENSOR_2, SENSOR_TYPE_LIGHT);
     SetSensorMode(SENSOR_2, SENSOR_MODE_RAW);
     OnFwd(OUT_A+OUT_C);
     start envia_senal;
     start revisa_senal;
}
```

Problemas Propuestos:

- Analizar este programa y aplicar este tipo de estructuras de código a otras actividades tales como:
- Generar un programa que al detectar un objeto a 30 cm el robot se detenga, retroceda un poco, gire 180 grados, y se mueva hacia adelante otra vez, en caso de escuchar sonido realizará el ciclo nuevamente.
- Construir un laberinto y programa la ruta para que el mini-robot lo atraviese sin hacer contacto con las paredes.
- Proponer una aplicación a este tipo de ejemplos, y entregar la justificación.

Producto:

- Deberá mostrarse el mini-robot funcionando.
- Entregar los esquemas en archivo a más tardar la próxima sesión y conservarlos para impresión en el Trabajo Integrador al final de la Unidad.

PRACTICA 18

Sensor de Luz

Objetivo:

El alumno programará en lenguaje código NQC el uso del sensor de luz para el Mini-Robot.

Teoría preliminar:

Para el lenguaje código NQC se deben indicar los puertos donde están conectados los sensores, se especifican como SENSOR_1, SENSOR_2 y finalmente SENSOR_3.
En este caso el sensor de luz estará conectado en el puerto 2.

Equipo Requerido:

1 Kit LEGO 9794
1 sensor de Luz

Procedimiento:

- Construir **TASKBOT** básico.
- En el BricxCC, programar la secuencia mostrada a continuación.
- Cargar el programa en el cerebro RCX.
- Ejecutar el programa y analizar su comportamiento.
- Realizar los problemas propuestos que se solicitan..

Código del Programa:

Primer programa

```
#define INTENSIDAD 40

task main()
{
     SetSensor(SENSOR_2,SENSOR_LIGHT);
     OnFwd(OUT_A+OUT_C);
     while (true)
     {
     if (SENSOR_2 > INTENSIDAD)
          {
```

```
            OnRev(OUT_C);
            Wait(10);
            until(Sensor(SENSOR_2) <= INTENSIDAD);
            OnFwd(OUT_A+OUT_C);
            }
      }
}
```

Problemas Propuestos:

- Analizar este programa y aplicar este tipo de estructura de código a las actividades realizadas previamente para ROBOLAB con el sensor de luz, por lo menos a dos de ellos.
- Proponer una aplicación a este tipo de ejemplo, y entregar la justificación.

Producto:

- Deberá mostrarse el mini-robot funcionando.
- Generar el código para NQC de los programas realizados anteriormente con ROBOLAB.
- Entregar los esquemas en archivo a más tardar la próxima sesión y conservarlos para impresión en el Trabajo Integrador al final de la Unidad.

PRACTICA 19

Rutinas y Subrutinas

Objetivo:

El alumno aplicará diversas estructuras de programación (Rutinas y Subrutinas) en código NQC para controlar el movimiento del Mini-Robot.

Teoría preliminar:

Tareas-Rutinas: En la programación NQC pueden usarse 10 tareas (rutinas) diferentes, cada una tiene un nombre único. La Tarea ***main*** debe estar siempre al principio, el resto se ejecutarán solamente cuando se indique, la tarea principal deberá concluir antes que las demás comiencen.

Subrutinas: Algunas veces se requiere una parte del código varias veces dentro de un programa, en este caso se coloca una subrutina y se le da un nombre. Se podrá ejecutar esta parte de código con solo llamarla dentro de una rutina, el RCX soporta 8 subrutinas.

Equipo Requerido:

1 Kit LEGO 9794
1 sensor de tacto

Procedimiento:

- Construir **TASKBOT** básico.
- En el BricxCC, programar la secuencia mostrada a continuación.
- Cargar el programa en el cerebro RCX.
- Ejecutar el programa y analizar su comportamiento.
- Realizar los problemas propuestos que se solicitan.

Código del Programa:

Primer Programa

```
task main()
{
     SetSensor(SENSOR_1,SENSOR_TOUCH);
     start revisa_sensor;
```

```
      start mover_en_cuadrado;

task mover_en_cuadrado()
{
      while (true)
      {
      OnFwd(OUT_A+OUT_C); Wait(100);
      OnRev(OUT_C); Wait(85);
      }
}

task revisa_sensor()
{
      while (true)
      {
      if (SENSOR_1 == 1)
      {
      OnRev(OUT_A+OUT_C); Wait(50);
      OnFwd(OUT_A); Wait(85);
      star mover_en_cuadrado
      }
      }
}
```

Segundo Programa

```
sub gira( )
{
      OnRev(OUT_C); Wait(340);
      OnFwd(OUT_A+OUT_C);
}

task main()
{
      OnFwd(OUT_A+OUT_C);
      Wait(100);
      gira();
      Wait(200);
      gira();
      Wait(100);
      gira();
      Off(OUT_A+OUT_C);
}
```

Problemas Propuestos:

- Generar un programa que utilice al menos 2 subrutinas, puede ser usando cualquiera de los sensores.
- Modificar el programa para que el Mini-Robot sea capaz de usar 3 subrutinas.
- Proponer una aplicación industrial para este tipo de sistema y entrega su justificación.

Producto:

- Deberá mostrarse el mini-robot funcionando.
- Entregar los esquemas en archivo a más tardar la próxima sesión y conservarlos para impresión en el Trabajo Integrador al final de la Unidad.

PRACTICA 20

Seguidor de línea

Objetivo:

El alumno aplicará diversas estructuras de programación en código NQC para controlar el movimiento del Mini-Robot para que siga la línea. Deberán analizarse las distintas condiciones lógicas implicadas.

Equipo Requerido:

1 Kit LEGO 9794
1 o 2 sensores de luz

Procedimiento:

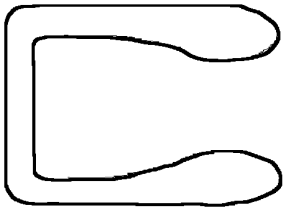
- Construir **TASKBOT** básico.
- En el BricxCC, programar la secuencia mostrada a continuación.
- Cargar el programa en el cerebro RCX.
- Ejecutar el programa y analizar su comportamiento.
- Realizar los problemas propuestos que se solicitan.

Problemas Propuestos:

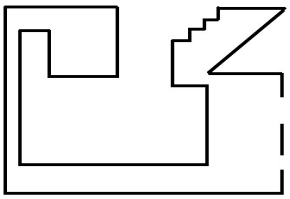
- Generar un programa que sea capaz de detectar la línea usando 1 sensor de luz.
- Posteriormente agregar un segundo sensor y generar un nuevo programa para que siga la línea utilizando 2 sensores.
- Modificar el programa para que el Mini-Robot sea capaz de seguir cualquiera de las pistas que se muestran a continuación.
- Proponer una aplicación industrial para este tipo de sistema y entrega su justificación.

Producto:

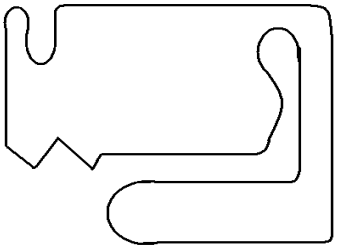
- Deberá mostrarse el mini-robot funcionando.
- Entregar los esquemas en archivo a más tardar la próxima sesión y conservarlos para impresión en el Trabajo Integrador al final de la Unidad.



Pista ejemplo 1



Pista ejemplo 2



Pista ejemplo 3

Para un siguiente nivel de capacidad en el manejo del sistema NXT se proponen modificaciones a las prácticas y retos tales como utilizar el sistema de sonidos para crear melodías, de dibujo, comunicación mediante Infrarrojo, y otras formas de controlar y sincronizar motores, para los usuarios expertos.

Referencias

Para mayor información respecto a los temas aquí mencionados:

Using ROBOLAB (LEGO - National Instruments 2002) Martha N. Cyr, Ph.D. Director of Engineering Educational Outreach, Tufts University, Massachusetts, US

Guía de inicio rápido sobre robótica y control computacional con LEGO® MINDSTORMS™ for Schools (2004) LEGO Educational division

Programación de LEGO robots utilizando NQC (Version 3.03 Oct 2 de 1999) por Marc Overmars

Guía del programador de NQC (Versión 2.5 a4) por Dave Baum

www.legomindstorms.com

http://www.legoedwest.com

Para temas avanzados y diversas construcciones podrás encontrar varios sitios y foros dedicados a proyectos de Mini-Robótica.

Con la colaboración de:

Fidencio Costilla Hermosillo
Luis Antonio Marizcal Barba