**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**



**¨Parking Robot¨**

**Fundamentos de Robótica**

**Profesor:**

[Ronald Alberto Ponguillo Intriago](https://beta.sidweb.espol.edu.ec/courses/26134/users/35676)

**Proyecto Primer Parcial**

**Paralelo: 1**

**Grupo:**

David Cobo

Marco Mendoza

Julio Realpe

Wellington Martinez

**CONTENIDO**

1. [**Objetivos**](#h.owof4p88539o)
2. [**Requerimientos previos**](#h.9y3crrcyfsqo)
3. [**Fundamentos Teóricos**](#h.8gfuw2p2woe9)
   1. [Robot](#h.d8bzc7kl9qq0)
   2. [LEGO Mindstorms NXT](#h.hxtkh3rxrgwz)
   3. [MatLab y Simulink](#h.nfdiskqccxlt)
   4. [MatLab](#h.f2ctiqivxp11)
   5. [Simulink](#h.h6z9f3g9809w)
   6. [Sensor de luz](#h.vmzuupm8fddi)
   7. [Sensor de ultrasonido](#h.6ho87sqcy02s)
4. [**Diagrama de Flujo de la solución a programar**](#h.m3udxhftdr6r)
5. [**Descripción del programa**](#h.r1sm63yvttb7)
   1. [Diseño general](#h.813sg3dttji9)
   2. [Reconocimiento de los sensores de Tacto y de Sonido](#h.l6cz1wi4jkh)
   3. [Abrir la barra hasta 90 grados.](#h.cn0pnw2ul285)
   4. [Bucle hasta que detecte que la barra tenga 90 grados](#h.uea360tniny8)
   5. [Detectar si el carro está en medio de la línea para pasar](#h.m1o5txk06cy7)
   6. [Caso cuando el carro pasa de la barra](#h.8r9nmrdo88ho)
   7. [Caso cuando el carro se quede en medio.](#h.dowtcymng9ws)
6. [**Recomendaciones**](#h.33z65o961n7j)
7. [**Conclusiones**](#h.6bzns2w328hh)
8. [**Bibliografía**](#h.i4m15nd1x2su)

# 

# 

# Objetivos

· Programar el robot MindStorms NTX desde Matlab usando el plugin Simulink.

· Fomentar en el estudiante habilidades para programar un robot, para la realización de cualquier proyecto de uso personal o laboral.

· Aplicar los conocimientos adquiridos en clase para resolver los problemas que presente el robot con el entorno y ambiente que impidan su desarrollo correctamente

· Adquirir destreza en el uso de sensores de ultrasonido y luz

# Requerimientos previos

* Instalar MatLab 2014 como mínimo requerido, tiene mejor compatibilidad con las herramientas a usar posteriormente.
* Instalar el paquete [RWTH - Mindstorms NXT Toolbox](http://www.mindstorms.rwth-aachen.de/), es gratis y open-source, necesario para programar al robot.
* Instalar los drivers del robot para ser compatible con el programa.

# Fundamentos Teóricos

## Robot:

Es una máquina programable, cuyo fin es realizar ciertas tareas y operaciones, bajo la restricción de ciertas reglas.

## LEGO Mindstorms NXT:

LEGO Mindstorms NXT es un kit de robótica programables liberado por LEGO a finales de julio de 2006. El kit está basado en dos versiones:. La versión Retail Version (juego # 8527) y la Education Base Set (juego # 9797) que viene con el software de programación NXT-G, o LabVIEW opcionalmente para LEGO MINDSTORMS. Una variedad de lenguas no oficiales existen, como NXC, NBC, leJOS NXJ y RobotC. Una nueva versión de la serie, el nuevo Lego Mindstorms NXT 2.0, fue lanzado en 2009, con un sensor de color y otras capacidades mejoradas. Ahora hay un nuevo robot de Lego Mindstorms, llamado el EV3.



El principal componente en el kit es un ordenador en forma de ladrillo llamado NXT Intelligent Brick AKA(Ciara). Se puede obtener información de hasta cuatro sensores y control de hasta tres motores, a través de una versión modificada de cables RJ12, muy similar pero incompatible con cables telefónicos RJ11. El pin de plástico para sujetar el cable en el zócalo se mueve ligeramente a la derecha.

El “Brick” tiene una pantalla LCD monocromática de 100 x 60 píxeles y cuatro botones que se pueden utilizar para navegar por una interfaz de usuario utilizando menús jerárquicos. Cuenta con una de 32 bits ARM7TDMI-core microcontrolador Atmel AT91SAM7S256 con 256KB de memoria FLASH y 64KB de RAM, más un 8-bit Atmel AVR ATmega48 microcontrolador, y el apoyo bluetooth.

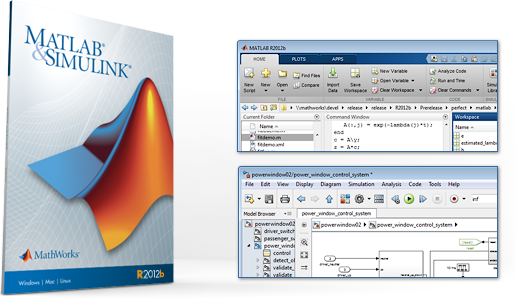
También cuenta con un altavoz y puede reproducir archivos de sonido a velocidades de muestreo de hasta 8 kHz. La energía es suministrada por 6 baterías en la versión de consumo del kit y por una batería recargable de iones de litio y un cargador en la versión educativa AA (1,5 V cada una).



## MatLab y Simulink

MATLAB es un lenguaje de programación de alto nivel para la computación numérica, adquisición de datos, y el análisis. Se puede utilizar para controlar robots LEGO NXT través de un puerto serie Bluetooth (comunicación del puerto serie es parte de la funcionalidad de base de MATLAB) o a través de una conexión USB; por ejemplo usando el RWTH - Mindstorms NXT Toolbox (gratis y de código abierto).

Simulink es un entorno de diagrama de bloques para el modelado y simulación de sistemas dinámicos. Usando Simulink, un usuario puede diseñar y simular los algoritmos de control y sistemas de LEGO, y, posteriormente, programar automáticamente el NXT LEGO o EV3. El soporte para la programación del Lego NXT o EV3 sólo requiere Simulink y está disponible sin cargo adicional.

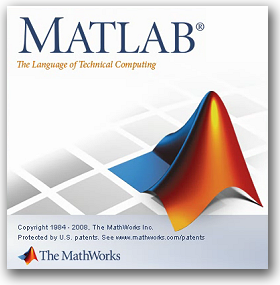


Estas herramientas se emplearon para la realización de la práctica, a continuación más detalles de sus funcionamiento:

## MatLab:

MATLAB es el lenguaje de alto nivel y un entorno interactivo utilizado por millones de ingenieros y científicos de todo el mundo. Se le permite explorar y visualizar las ideas y colaborar en todas las disciplinas, incluyendo la señal y el procesamiento de imágenes, comunicaciones, sistemas de control, y las finanzas computacional.

Se puede utilizar MATLAB en proyectos tales como el consumo de energía de modelado para construir las redes eléctricas inteligentes, desarrollo de algoritmos de control para vehículos hipersónicos, analizando los datos del tiempo para visualizar la trayectoria y la intensidad de los huracanes, y corriendo a millones de simulaciones para determinar la dosis óptima de antibióticos.



## Simulink:

Simulink® es un entorno de diagrama de bloques para la simulación multidominio y diseño basado en modelos. Es compatible con la simulación, generación automática de código, y pruebas continua y verificación de sistemas embebidos.

Simulink ofrece un editor gráfico, bibliotecas de bloques personalizables y solucionadores para el modelado y simulación de sistemas dinámicos. Está integrado con MATLAB®, lo que le permite incorporar algoritmos de MATLAB en modelos y resultados de la simulación de exportación a MATLAB para su posterior análisis.

Permite configurar las tareas y/o eventos que se requiere que el robot realice, a partir de diagramas de bloques. Se emplearon los sensores de luz, tacto y de ultrasonido para la llevar a cabo la realización de la práctica, los cuales se describen a continuación:

## Sensor de luz:

Son sensores que emplean fotoresistencias, las cuales son resistencias que van variando de acuerdo a la intensidad de la luz que incide en estas, esta resistencia es muy alta cuando el sensor no detecta luz (está cubierto o la distancia de la luz hacia el sensor es muy grande) y decrece a medida que recibe mayor luz e intensidad.

La variación de valores comprendida en el sensor de luz comprende entre 0 y 100 (de baja intensidad hacia alta intensidad).



El sensor de luz se puede emplear de varias formas. las cuales se describen a continuación:

* Puede emitir una frecuencia de luz y posteriormente medir qué porcentaje ha recibido con respecto al emitido. Esto se puede emplear para detectar el brillo de las superficies donde impacta la luz emitida.
* Se puede emplear para detectar la luz del ambiente, conociendo así, si es de día o de noche, si un cuarto tiene un foco prendido o no, etc.

## Sensor de ultrasonido:

Se emplea básicamente para detectar obstáculos o también para medir distancias, puede detectar objetos que estén a distancias entre 0 y 255 cm con una precisión de +/-3 cm (margen de error).

Su funcionamiento es idéntico al de un radar o un sonar, al enviar frecuencias de sonido muy altas y medir el rebote de dichas ondas (eco).



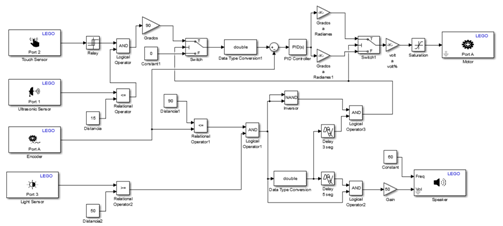
# Diagrama de Flujo de la solución a programar





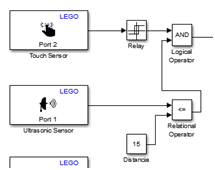
# Descripción del programa

## Diseño general



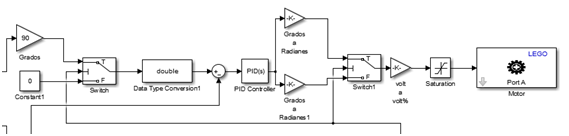
## 

## Reconocimiento de los sensores de Tacto y de Sonido



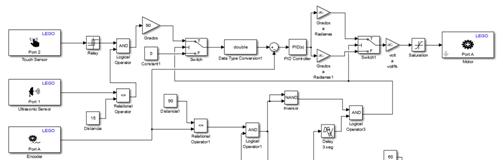
El sensor de tacto envía una señal y mantiene dicha señal en alto con el Relay hasta terminar el programa, mientras que paralelamente se envía una señal de sonido para detectar si el carro está frente a la barra, dicha señal detecta si el carro está a 15 cm de la barra. Ambos sensores envían verdadero si detectan su respectiva señal y se procede al siguiente proceso después de salir verdadero con el operador AND.

## Abrir la barra hasta 90 grados.



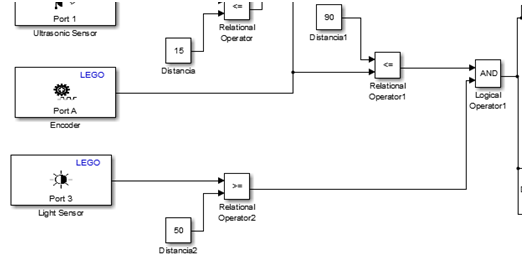
Para abrir la barra después de detectar las dos señales, se procede a rotar la barra 90 grados, para esto se hacen varias transformaciones y asignaciones como tipo de dato, grados a radianes, etc., con este dato ya convertido se pasa al siguiente proceso, si el carro aun no pasa se abrirá la barra, esto es en un sentido, cuando el carro ya haya pasado el switch cambia, lo que hace que gire la barra 90 grados, pero en sentido contrario, es decir para cerrar; este proceso depende del valor de detección del carro, ya que el switch depende de dicho valor para hacer girar la barra en dos sentidos (abrir y cerrar).

## Bucle hasta que detecte que la barra tenga 90 grados



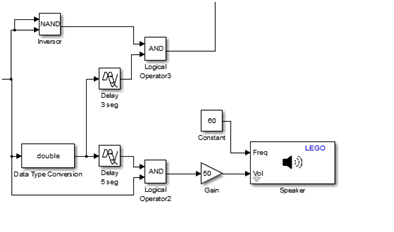
El bloque Port A da la funcionalidad de un bucle que se mantenga en falso hasta que la barra llegue a 90 grados, este dato booleano servirá para el siguiente proceso y como condición para avanzar con la funcionalidad del robot.

## Detectar si el carro está en medio de la línea para pasar



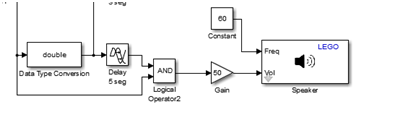
Usando el dato anterior (barra en 90 grados) como verdadero y que el sensor de luz detecte que el carro aun esta en medio, se hará una operación AND para validar que ambas señales sean verdaderas, así se continuará con el siguiente proceso.

## Caso cuando el carro pasa de la barra



En caso de que el evento sea en condiciones normales, es decir que el carro pase de la barra, girará 90 grados de vuelta, este proceso se lo explicó anteriormente, el dato de este proceso, es decir verdadero si paso el carro, sirve para el otro proceso de girar 90 grados en sentido contrario (cerrar) cambiando esto con un switch que previamente se explicó. El proceso termina ahí y un contador aumentará en uno por cada carro que pase.

## Caso cuando el carro se quede en medio.



Este proceso ocurre cuando se da un valor de falso en caso de quedar en medio el carro, en este proceso se esperan 5 segundos con el Delay, y con el operador AND, se espera que los valores sean verdaderos cuando haya pasado los 5 segundos que el carro no avanza y el dato que previamente era verdadero, que es si se abrió la barra, en caso de que pasen los 5 segundos, se activará una alarma.

Cuando el carro avance, los valores de las señales cambiarán, cambiará a verdadero el proceso de avanzar el carro y ya no ejecutará el proceso de la alarma, se ejecutará el proceso de cierre de la barra (giro de 90 grados en sentido contrario).

**Diseño del Robot**

Para este proyecto se diseñó un robot según la figura:



Donde se observan dos sensores en la parte frontal y uno en la parte de atrás:



# Recomendaciones

* Instalar Matlab con Simulink sin errores y con la librería de Lego NXT, para esto, instalar la versión 2014 como mínimo para no tener problemas con la librería, ya que al no ejecutarse correctamente, afecta el tiempo de trabajo en el desarrollo del proyecto.
* Investigar el funcionamiento de Simulink, este programa usa un lenguaje de programación de bloques parecido a Lego Mindstorms.

# Conclusiones

* Para llevar a cabo las acciones es preciso manejar vehículos con forma similar, ya que el sensor podría confundir su forma con la forma de otro objeto y no llevar a cabo la acción esperada.
* Si se presiona varias veces el botón que emite la orden de levantar el brazo robótico, es posible que el programa presente conflictos de tiempo, y realice acciones en tiempos inesperados.
* Pudimos concluir que el código que se creó para realizar la práctica funciona correctamente si se siguen los pasos establecidos en la secuencia de instrucciones de la misma, sin embargo, podría esperarse eventos inesperados por parte de la ejecución de las órdenes del robot si la práctica se lleva a cabo bajo otras condiciones de terreno y ambiente, esto ocurre debido al funcionamiento de los sensores de ultrasonido y de luz.

# Bibliografía

* <http://rbtnxt.blogspot.com/2009/01/el-sensor-de-sonido.html>
* <https://es.wikipedia.org/wiki/Rob%C3%B3tica>
* <http://www.monografias.com/trabajos31/robotica/robotica.shtml>
* http://www.mathworks.com/lego