ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería Eléctrica y Computación

USO DE MATLAB Y SIMULINK PARA EL CONTROL DE ROBOTS Y LA OBSERVACIÓN DE SENSORES DE TACTO Y LUZ

INFORME DE MATERIA DE GRADUACIÓN

Previo a la obtención del Título de:

INGENIERO EN ELECTRICIDAD ESPECIALIZACIÓN ELECTRÓNICA Y AUTOMATIZACIÓN INDUSTRIAL

Presentado por:

Rosalía Aurora Pasmay Macías
Carlos Stalin Montero Orellana
Freddy Fernando Oviedo Moreno

GUAYAQUIL – ECUADOR
2009

AGRADECIMIENTO

A la ESPOL por la formación académica recibida en nuestra vida universitaria. A los buenos ingenieros por los conocimientos, el apoyo y la amistad impartida en todos estos años de estudio.

A todas las personas que de uno u otro modo colaboraron en la realización de este proyecto.

A los verdaderos amigos, que comparten las herramientas y el conocimiento.

Y un agradecimiento especial a todos aquellos que de uno u otro modo nos brindan las oportunidades de aprender.

DEDICATORIA

A Dios, Ser Supremo que guía nuestras vidas y que día a día nos llena de fortaleza para seguir adelante.

A nuestros padres, quienes han sido nuestro apoyo constante. Y nos han brindado a diario conocimiento intelectual y espiritual.

A nuestros hermanos quienes con su alegría fueron motivación constante para la finalización de este informe.

TRIBUNAL DE GRADUACIÓN

ING. CARLOS VALDIVIESO A.

PROF. DE LA MATERIA

ING. HUGO VILLAVICENCIO V.

DELEGADO DEL DECANO

ING. JORGE ARAGUNDI

SUBDECANO DE LA FIEC

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este Informe de graduación, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral".

(Reglamento de Graduación de la ESPOL)

Rosalía A. Pasmay Macías

Freddy F. Oviedo Moreno

Carlos S. Montero Orellana

RESUMEN

El proyecto explorador de franjas negras, es creado con fines de protección al medio ambiente en campos donde el desperdicio causa muchos problemas e inconvenientes a la vida que se encuentra a su alrededor y el ahorro trae beneficios económicos a la protección del mundo. Esto hizo que nos enfoquemos en las grandes empresas que se dedican a la obtención de petróleo y las cuales en su mayoría de veces son las causantes de daños, en especial en nuestro país.

El explorador de franjas negras, cuya construcción fue inspirado en un modelo a todo terreno, con miras a su automatización y comunicación a distancias donde el ser humano no intervenga pero que interactué en condiciones, decisiones y respuestas a los problemas que tenga en su trayectoria. Su comunicación será por medio de la tecnología Bluetooth y comandado por el software Matlab que ha sido habilitado con ayuda de un toolbox especial para mantener en comunicación con el explorador, cuyo funcionamiento es quien tome decisiones y sea en si el controlador a todos sus problemas.

El explorador es un diseño armado con ayuda de las herramientas o piezas físicas de Lego Mindstorms NXT y el uso del sensor de tacto que da al proyecto el encendido o movimiento a sus acciones, el sensor ultrasónico quien detecta presencia en frente de su trayectoria y el sensor de luz quien es el pionero en detectar franjas o manchas de color negra que en teoría representa las manchas de petróleo.

El proyecto se mantiene en comunicación con el ordenador en tiempo real, de esta manera sabremos si existen manchas negras ya que el explorador envía a cierto tiempo dos tipos de datos que representa la escala de grises medida por el sensor de luz y el dato referente a la presencia de objetos alrededor enviados por el sensor de ultrasónico.

Los datos son recopilados en una hoja del programa Excel donde se realiza una macro función y luego son analizadas y comparadas con las graficas obtenidas en Matlab.

Los resultados han sido proporcionados, mostrados y escritos en sus respectivas conclusiones y las observaciones de los muchos casos que se han presentado, ya que son útil a una mejora del proyecto especificadas en sus respectivas recomendaciones.

ÍNDICE GENERAL

3. DISEÑO E IMPLEMENTACION DE UN EXPLORADOR DETECTOR DE FRANJAS NEGRAS Y PRESENCIA

4. SIMULACION, PRUEBAS Y DATOS EXPERIMENTALES

CONCLUSIONES

RECOMENDACIONES

ANEXOS

BIBLIOGRAFÍA

ÍNDICE DE FIGURAS

INTRODUCCIÓN

El presente Proyecto forma parte de la materia de graduación de “Microcontroladores Avanzados” y consiste en el diseño de un “Explorador Detector de Franjas Negras y Presencia”. El principal objetivo es desarrollar un proyecto basado en la comunicación del robot Lego NXT Mindstorms y el software Matlab para destacar el uso de los sensores de tacto, luz y ultrasonido en una determinada aplicación.

Con el pasar del tiempo Matlab dejó de ser solo un apoyo para cursos relacionados con la Teoría de Matrices y se ha convertido en una poderosa herramienta tanto en el ámbito estudiantil como industrial; de la misma forma el Lego NXT Mindstorms dejó de ser un simple juego para niños y ahora es fuente importante para diversas aplicaciones gracias a la disposición de tecnología Bluetooth incorporada a su bloque NXT, permitiendo el envío de códigos de programación en un tiempo real al explorador.

CAPÍTULO 1

1 TECNOLOGIA UTILIZADA

1.1 Antecedentes

La Robótica, desde sus orígenes hasta nuestros días, ha evolucionado y experimentado cambios los cuales se han hecho más notorios en los últimos años, gracias al progreso de la tecnología la cual se ha convertido en parte indispensable para el desarrollo humano.

Dentro del campo de la Robótica, se pueden especificar varias generaciones:

- La primera se dio a principios de los años 60 y la constituyeron robots manipuladores que sólo podían realizar movimientos repetitivos. Sus sensores eran internos y sus movimientos eran limitados con poca precisión.
- La segunda se presenta a finales de los años 70, los cuales incorporan nuevos tipos de sensores que proporcionan al robot información del mundo exterior permitiéndole tomar decisiones limitadas y reaccionar ante el entorno de trabajo.

- La tercera, surgida en los últimos años, emplea la inteligencia artificial y potentes ordenadores para resolver problemas complejos e interpretar información procedente de avanzados sensores.

Todos estos progresos han permitido ampliar los campos de aplicación de la robótica respecto a los de su interés inicial, centrado principalmente en aplicaciones industriales.

Sin embargo, en los últimos años se tiende a desarrollar e implementar robots móviles que pueden realizar una gran cantidad de tareas en entornos distintos. El desarrollo tecnológico y la investigación han conducido a originar un nuevo tipo de robot conocido como robot de servicio, cuya finalidad principal es desarrollar tareas útiles para la sociedad, influyendo en la vida de muchas personas como es el caso de las sillas de ruedas semiautónomas que se guían mediante órdenes verbales, robots para alimentar a personas que carezcan de movilidad en los brazos.

Debido al impacto de la Robótica en la sociedad, en la última década investigadores e industrias han propuesto y desarrollado numerosos kits para la construcción de robots, diseñados para estimular el aprendizaje de contenidos relativos a la educación en áreas científicas como matemática, física, informática y mecánica.

Los kits incluyen pequeños motores, sencillos sensores, ruedas, engranajes, poleas y relés; entre otras cosas necesarias para poner en práctica la imaginación para la construcción de un robot. Productos como Lego Dacta y Lego CyberMaster incluyen cables o equipamientos de radio que posibilitan conectar el robot con un ordenador personal, lo cual permite al usuario controlar el invento.

Para enero del 2006 Lego anunció su versión Mindstorms NXT que permite al usuario construir robots autónomos con toda la capacidad de control localizada en el interior de la máquina. Lego Mindstorms NXT es un juego de robótica para niños y adultos, que posee elementos básicos de las teorías robóticas, como la unión de piezas y la programación de acciones, en forma interactiva.

Puede ser usado para construir un modelo de sistema integrado con partes electromecánicas controladas por computador. Prácticamente todo puede ser representado con las piezas tal como en la vida real, como un elevador o robots industriales.

1.1.1 Orígenes de Lego Mindstorms

La línea Lego Mindstorms fue uno de los resultados de la fructífera colaboración entre Lego y el MIT (Instituto Tecnológico de Massachusetts). Según este trato, Lego financiaría investigaciones del grupo de epistemología y aprendizaje del MIT sobre cómo aprenden los niños y a cambio obtendría nuevas ideas para sus productos.

El mentor del grupo, Seymour Papert, era un matemático interesado por la relación entre la ciencia, la adquisición del conocimiento y el desarrollo de la mente infantil. De hecho, el nombre del producto, Mindstorms, proviene del título de un libro suyo, llamado MindStorms: Children, Computers and Powerfull Ideas, en el que describe sus ideas respecto al empleo de las computadoras como impulsoras del aprendizaje.

La línea Mindstorms no fue el primer fruto de la relación entre Lego y el MIT, aunque sí el más exitoso. Con anterioridad Lego se había interesado por el lenguaje de programación Logo, fruto de este interés nación en 1986 Lego TC Logo el cual a pesar de alcanzar un relativo éxito comercial el sistema imponía restricciones tanto físicas como imaginativas.

Para principios de los años 90, se empezó a investigar la posibilidad de crear un programa de computadora que se conectara con una construcción de Lego pero el coste de la tecnología era demasiado alto y el mercado no era propicio. Por el año 1995 se comienza a desarrollar el bloque RCX, el cual contaba con un microcontrolador; llevando de esta manera a transformar las construcciones de Lego de estructuras estáticas a máquinas dinámicas que interactúan con el mundo.

Además del bloque RCX, hubo otros bloques programables que eventualmente se fueron desarrollando hasta lograr la versión definitiva del NXT, que se empezó a comercializar para enero del 2006.

1.2 Descripción de la Problemática

El diseño del proyecto basado en la comunicación entre el software Matlab y el robot Mindstorms NXT con la utilización de los sensores y motores que dispone se realizó dada la necesidad de analizar el funcionamiento y efecto que producen los sensores en una actividad industrial además de corroborar el gran alcance que ha logrado Matlab en relación a su comunicación con programas desarrollados en cualquier tipo de plataforma permitiéndole estar presente en los diseños de los procesos industriales.

Hoy por hoy, los sensores se han convertido en la parte vital para el desarrollo industrial. Están presentes en un proceso gobernando el correcto funcionamiento del mismo. Existen de todo tipo de acuerdo al uso y a la función que se les quiera asignar. Es muy importante analizar el comportamiento que tiene cada uno, los parámetros que maneja con sus respectivas limitaciones con el propósito de que cumplan satisfactoriamente la función que se les asigna.

En el presente trabajo se analiza el funcionamiento de los sensores de tacto, luz y de ultrasonido, los cuales tienen un gran número de aplicaciones. Es así como podemos observar el uso del sensor de luz en los procesos petroleros para evitar algún tipo de derrame que pueda suceder poniendo en peligro el equilibrio del medio ambiente. De la misma manera el sensor de ultrasonido es utilizado en el estudio de sistemas electrónicos orientados a evitar el choque de automóviles en las carreteras a través de determinar la distancia a la que se encuentra un auto respecto a los que se mueven a su alrededor.

Así como los procesos nombrados anteriormente, existen un gran número que día a día se desarrollan y otros que se encuentran en estudio; todos ellos bajo la premisa de un modelo. Es ahí donde Matlab con todas sus herramientas en especial Simulink desarrollan un papel importante, ya que en esta plataforma todo proceso puede ser modelado y luego simulado.

Realizar un estudio de todas las aplicaciones que Matlab permite realizar tomaría mucho tiempo, sin embargo en el presente trabajo pretendemos destacar sobre todo la facilidad de comunicación e interacción con otras plataformas de trabajo.

CAPITULO 2

2 FUNDAMENTACION TEORICA DE LOS RECURSOS UTILIZADOS

2.1 Fundamento Teórico

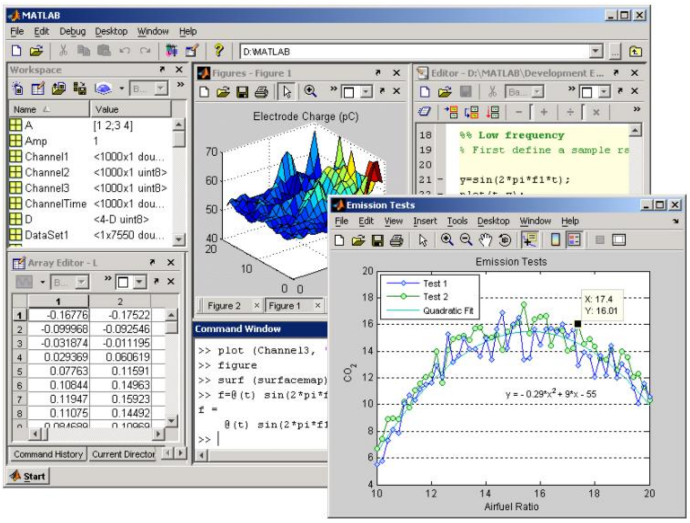
En el presente capítulo se detallan los recursos y herramientas necesarias para la implementación del robot explorador como es el uso del software Matlab y el hardware Lego Mindstorms NXT. Además se analiza la interacción entre ellos a través de un ordenador permitiendo el envío y recepción de datos mediante comunicación Bluetooth o USB.

2.2 Descripción del Software Matlab

MATLAB, abreviatura de las palabras MATrix LABoratory (Laboratorio de matrices), es un programa matemático que brinda un entorno de desarrollo integrado con un lenguaje de programación propio, conocido como lenguaje M. Está disponible para las plataformas Unix, Windows y Apple Mac OS X.

Entres sus utilidades básicas presenta: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware.

Además dispone de dos herramientas adicionales que incrementan sus recursos de utilización, los cuales son Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario – GUI). Adicionalmente se puede aumentar la capacidad de MATLAB con las cajas de herramientas (toolboxes); y las de Simulink con los paquetes de bloques (blocksets).



Name	Value
A	[1 2;3 4]
Amp	1
Channel1	<1000x1 dou...
Channel2	<1000x1 uint8>
Channel3	<1000x1 uint8>
ChannelTime	<1000x1 dou...
D	<4-D uint8>
DataSet1	<1x7550 dou...

	1	2
1	-0.16776	-0.17522
2	-0.099968	-0.092546
3	-0.031874	-0.011195
4	0.029369	0.060619
5	0.07763	0.11591
6	0.10844	0.14963
7	0.11947	0.15923
8	0.11075	0.14492

```
>> plot (Channel3, '
>> figure
>> surf (surfacemap)
>> f=@(t) sin(2*pi*f
f =
   @(t) sin(2*pi*f1
>> |
```

```
%% Low frequency
% First define a sample re

y=sin(2*pi*f1*t);
```

Fig. 2.2: Visualización del entorno de Matlab

De esta manera, MATLAB permite el desarrollo de un gran número de aplicaciones tales como: procesamiento de señales e imágenes, comunicaciones, diseño de sistemas de control, sistemas de prueba y medición, modelado y análisis financiero y biología computacional.

2.2.1 Descripción de la herramienta Simulink

Simulink es una herramienta para el modelaje, análisis y simulación de una amplia variedad de sistemas físicos y matemáticos. Soporta tanto sistemas lineales como no lineales en tiempo continuo, muestreados, híbridos y sistemas muestreados a diferentes frecuencias. Como una extensión de MATLAB, adiciona características específicas a los sistemas dinámicos, mientras conserva toda la funcionalidad del propósito general de MATLAB, generando archivos con extensión .mdl (de model).

Permite controlar los atributos de las señales y parámetros asociados con un determinado modelo a través de dos fases de uso: la definición del modelo y el análisis del modelo. La definición del modelo hace referencia a la construcción del modelo a partir de elementos básicos tales como integradores, bloques de ganancia o servomotores. El análisis del modelo significa realizar la simulación, linealización y determinar el punto de equilibrio de un modelo previamente definido.

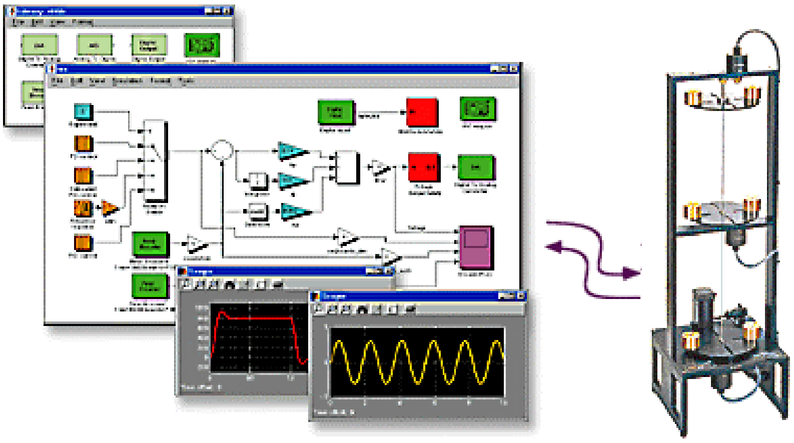


Fig. 2.2.1: Visualización del entorno de Simulink

Se emplea principalmente en Ingeniería Electrónica, en temas relacionados con el procesamiento digital de señales (DSP), involucrando temas específicos de ingeniería biomédica, telecomunicaciones, entre otros. También es muy utilizado en Ingeniería de Control y Robótica.

2.3 Descripción del hardware Lego NXT Mindstorms

Lego NXT MINDSTORMS es una versión mejorada a partir de Lego Mindstorms RCX, que generalmente se considera la precursora de los bloques programables de Lego. Debido a la comercialización de los bloques programables, Lego vendió la generación NXT en dos versiones: Retail Version y Education Base Set. Además, Lego dispuso de varios kits para desarrolladores según las características de los programas que estuvieran desarrollando.

Fig. 2.3: Componentes principales de Lego NXT MINDSTORM

Cualquiera que sea el tipo de kit a utilizarse, todos están constituidos por 3 partes fundamentales: el bloque NXT, motores y sensores; y finalmente el software con sus respectivos medios de comunicación.

2.3.1 Bloque NXT

Es el cerebro del robot MINDSTORMS, formado por una memoria capaz de almacenar y ejecutar las diferentes operaciones que el usuario requiera. Sus componentes principales son:

- Tres puertos de salida para la conexión de motores: A, B y C.
- Cuatro puertos de entrada para conectar sensores: 1, 2, 3 y 4.

- Puerto USB para la carga y descarga de programas desde el ordenador.
- Altavoz
- Botón naranja: ON/ENTER/RUN
- Flechas de desplazamiento
- Botón gris oscuro: Clear/ Go back
- Display

2.3.2 Motores y Sensores

El kit NXT MINDSTORMS utiliza 3 servos motores, los cuales permiten la detección de giros de cada rueda gracias a la incorporación de un sensor de rotación, indicando los giros completos o medios giros en grados. Su accionamiento está condicionado según el programa realizado por el usuario. Adicionalmente el sensor de rotación permite establecer diferentes velocidades para los motores. Complementariamente, el usuario puede hacer uso de cuatro tipo de sensores: contacto, sonido, luz y ultrasónico.

A continuación se detalla los 3 sensores que se utilizan en el proyecto.

2.3.2.1 Sensor de Contacto

El sensor de contacto brinda al robot un sentido de tacto, cuando éste es pulsado o soltado por algo. El sensor puede contar con una o varias pulsaciones.

El pulsador del sensor es grande y tiene mayor recorrido lo que es más adecuado para su función. También presenta un agujero en forma de cruz en la parte delantera del pulsador que facilita la aplicación para cierto tipo de montaje como detectores de obstáculos.

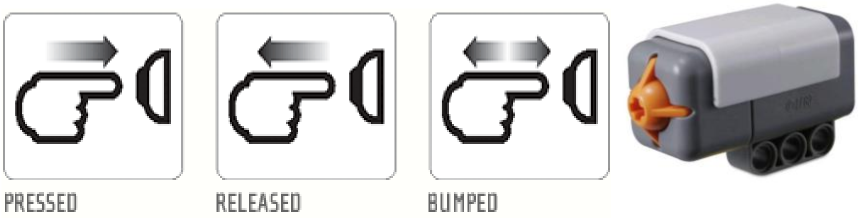


Fig. 2.3.2.1: Sensor de Contacto y sus posibles movimientos

Puede ser utilizado en varias aplicaciones y de diversas maneras, por ejemplo en un brazo robótico, en el cual al incorporar el sensor de contacto permita la detección o presencia de algún objeto agarrado. También se lo puede utilizar como un comando que permita ejecutar alguna acción como: caminar, hablar, tocar un a puerta y hasta encender un televisor.

2.3.2.2 Sensor de Luz

El sensor de luz es uno de los sensores que le brinda visión al robot. Este sensor permite la distinción entre la luz y la oscuridad. Puede leer la intensidad de la luz en una habitación y medir la intensidad de la luz de las superficies de colores.

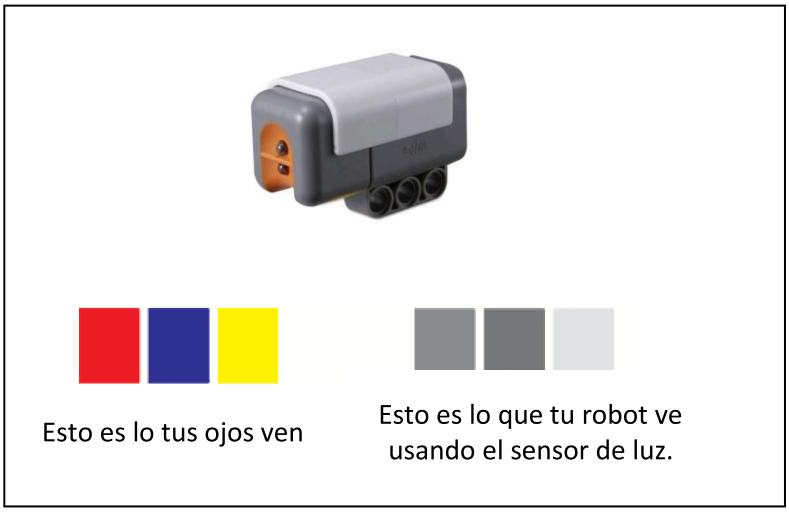


Fig. 2.3.2.2: Diferencia entre visión del usuario ante la del robot

Puede ser utilizado para hacer un robot de alarma; cuando un intruso entre la luz de la habitación se enciende y el robot puede reaccionar para defender su propiedad.

También se lo puede usar en un seguidor de línea Este sensor puede trabajar tanto en modo reflexión o como medidor de nivel de luz.

2.3.2.3 Sensor Ultrasónico

El sensor ultrasónico permite a su robot detectar objetos. Es capaz de medir las distancias desde 0 hasta 255 centímetros con una precisión de +/- 3 cm. También puede sensar en unidades de pulgadas.

El cálculo de la distancia lo realiza en relación directa al tiempo que tarda una onda de sonido para golpear un objeto y volver como un eco. Si el tiempo es pequeño, el objeto está cerca; si el tiempo es grande, el objeto está lejos y si el tiempo es mayor que un determinado valor, el objeto está fuera de alcance.

Fig. 2.3.2.3: Sensor Ultrasónico

Los objetos de gran tamaño con superficies duras devuelven las mejores lecturas; mientras que los objetos de tela suave, curveados, delgados o pequeños pueden ser difíciles para el sensor de detectar.

Su mayor aplicación es la detección de obstáculos, permitiendo que el robot evite tropezar o chocar con algún objeto que se encuentre a una distancia determinada con respecto al robot.

2.3.3 Software y medios de comunicación

Las opciones que tenemos para programar este dispositivo son varias: por un lado tenemos las herramientas comercializadas por Lego y por otro las comercializadas por terceras empresas en las que abarcan software con un entorno gráfico o con un lenguaje de programación.

2.3.3.1 Mindstorms NXT Software

Este software permite al usuario diseñar el programa en forma gráfica y subirlos al NXT, vía USB o conectividad Bluetooth; su programación es sencilla y muy vistosa. El software Lego MINDSTORMS NXT incorpora la construcción y las instrucciones de programación para 4 diferentes modelos.

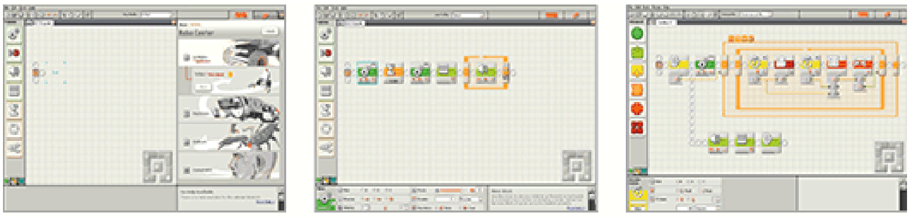


Fig. 2.3.3.1a: En torno de Trabajo del Software MINDSTORMS NXT

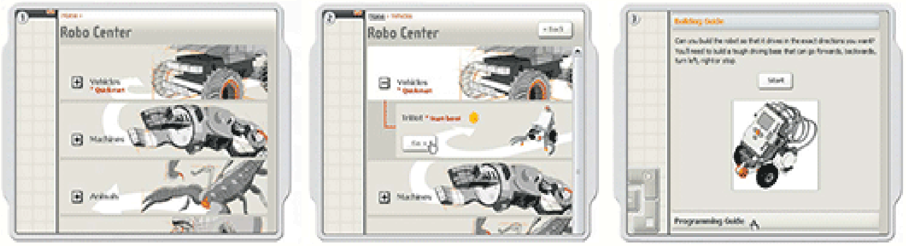


Fig. 2.3.3.1b: Ayuda para el Trabajo del Software MINDSTORMS NXT

2.3.3.2 Comunicación Inalámbrica Bluetooth

El Lego MINDSTORMS NXT incluye un nodo inalámbrico Bluetooth que habilita el NXT y brinda comunicación con otros dispositivos Bluetooth.

Fig. 2.3.3.2: Comunicación Bluetooth

Bluetooth es una tecnología que permite que sea posible enviar o recibir datos sin usar cables. Usando Bluetooth se puede intercambiar programas entre tu NXT y otros NXT's , establecer una conexión inalámbrica entre un ordenador y el robot permitiendo cargar programas al NXT en forma instantánea estando en otra localización distinta al ordenador.

2.4 Comunicación Matlab - Lego NXT Mindstorms

Para el control del Lego NXT MINDSTORMS desde MATLAB a través de la comunicación inalámbrica Bluetooth o vía USB se puede hacer uso de las siguientes herramientas:

- **RWTH-Mindstorms NXT Toolbox**: Toolbox de comunicación vía Bluetooth para MATLAB, desarrollado en el Instituto de Imagen y Visión por Ordenador RWTH. Constituye un software de código abierto que permite el diseño y desarrollo de funciones de alto nivel aprovechando la interacción entre sensores y motores del Lego NXT.

  No se recomienda el uso para el control de un robot en tiempo real debido a los retardos existentes en la comunicación que son muy grandes comparados con los de la conexión USB.

- **Fantom driver:** Driver básico para la comunicación del NXT vía USB. Se instala con el entorno de desarrollo NXT-G.
- **NXTender:** Software desarrollado de manera independiente que permite al NXT controlar el ordenador, emulando un teclado o un ratón.

- **Simulink LEGO MINDSTORMS NXT Code:** Ha sido desarrollado usando Simulink, Real-Time Workshop y Real-Time Workshop Embedded Coder, convirtiéndose en un lenguaje de máquina que puede ser descargado para ejecutarse en el Lego MINDSTORMS NXT.

  Con este control integrado usted puede:

  - Programar los motores y sensores en forma precisa y coordinada permitiendo la ejecución en tiempo real.
  - Realizar el modelo, visualización y prueba del robot usando software de Simulink en animación 3D.

CAPITULO 3

3 DISEÑO E IMPLEMENTACIÓN DE UN EXPLORADOR DETECTOR DE FRANJAS NEGRAS Y PRESENCIA

3.1 Diseño del Explorador

La construcción de este detector de franjas negras y presencia se llevó a cabo dada la necesidad de implementar un prototipo en que se destaquen la utilidad de los sensores de tacto, luz y ultrasonido del Lego NXT Mindstorms; así como la comunicación del robot con un software de gran alcance como es Matlab.

Para el efecto se diseñó un explorador que llamamos Ciclón al cual se le acopló tres servomotores responsables del movimiento del robot y los tres sensores descritos anteriormente que condicionan y limitan la trayectoria del mismo.

El objetivo de ciclón es detectar la presencia de algún objeto que se encuentre a una distancia determinada y evitar la colisión con el mismo. Así mismo su trayectoria se limita ante la detección de una franja de color oscuro, instante en el cual ciclón cambia de dirección con la finalidad de que regrese a su trayectoria permitida. El movimiento será continuo hasta que el sensor de tacto sea activado, instante en el cual se detendrá; simulando un control de apagado manual.

El diseño se enfoca a semejar una determinada aplicación industrial gobernada por motores donde su accionamiento y correcto funcionamiento sea condicionado por sensores. En el anexo E se encuentra detallado la construcción del explorador.

3.1.1 Disposición de los Servomotores

El robot dispone de tres servomotores que ejecutan diferentes acciones. Están colocados de tal forma que sea posible desarrollarse el movimiento continuo y giro del explorador.

El giro de cada uno de los servomotores puede ser establecido mediante un valor constante del ángulo de giro o mediante un tiempo de recorrido.

Así mismo, la potencia de trabajo puede ser establecida en un rango de -100 a 100.

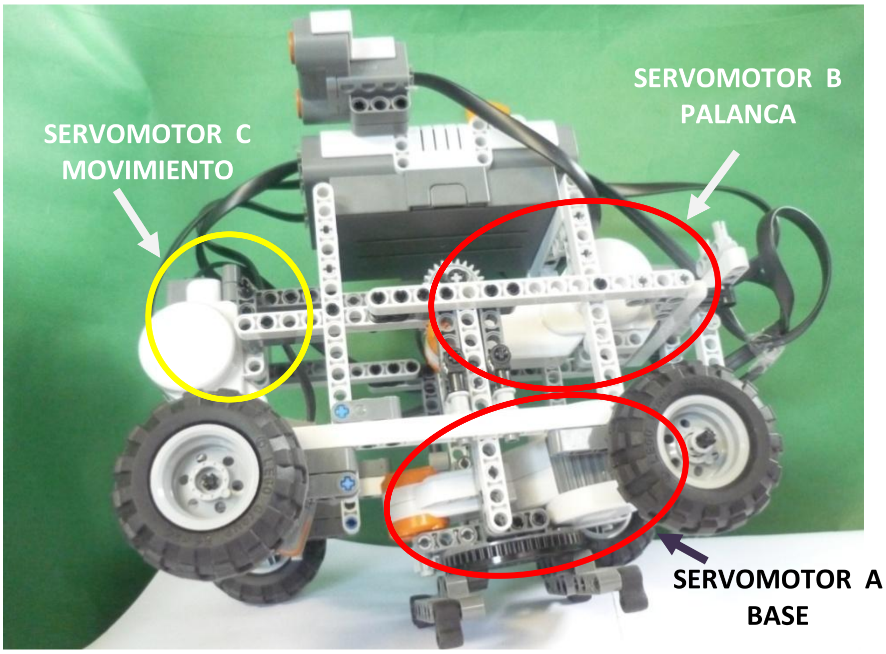


Fig. 3.1.1: Ubicación de Servomotores en el Explorador

El Servomotor B está ubicado en la parte superior posterior del robot, Se lo denomina Palanca debido a que permite el desplazamiento vertical del Motor C, logrando con esto la elevación y descenso del robot para el cambio de dirección del mismo. Debido a la fuerza que debe desarrollar por la función que realiza se le acopló un sistema de engranajes.

El Servomotor C o también llamado Servo de Movimiento está ubicado en la parte frontal del robot. Sus giros permiten un desplazamiento continuo del explorador.

El Servomotor A o base está ubicado en la parte inferior del robot, su activación permite que el robot gire sobre su propio eje durante un tiempo determinado para lograr un cambio de dirección en su trayectoria. Al igual que el servo B también tiene acoplado un sistema de engranajes; además que funciona a una potencia mayor que los otros dos servos debido al trabajo exigente que realiza.

3.1.2 Disposición de los Sensores

El robot dispone de tres sensores que van a condicionar el movimiento de los servomotores; los cuales son: tacto, luz y ultrasonido.

Están colocados de tal forma que sea posible detectar de manera adecuada las variables que gobiernan y ejecutan la parte fundamental de la programación.

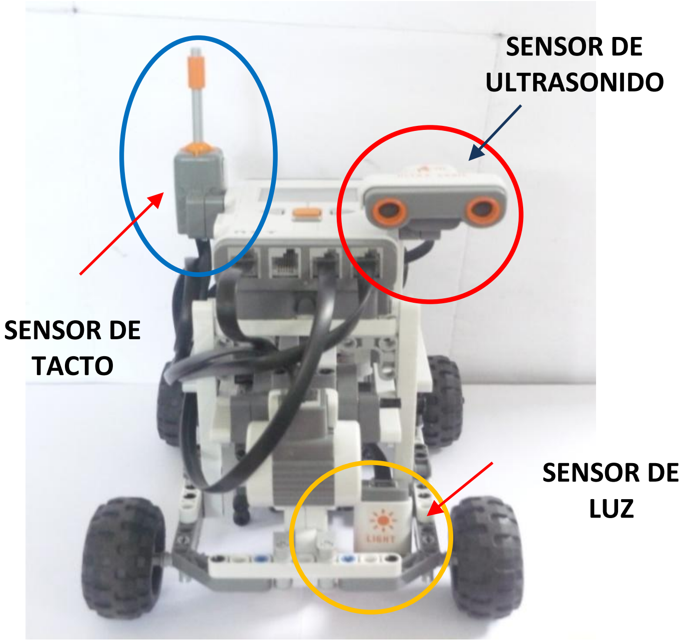


Fig. 3.1.2: Ubicación de los Sensores en el Explorador

3.1.2.1 Utilización del Sensor de Luz

El Sensor de Luz se encuentra ubicado en la parte inferior izquierda del robot a un centímetro del suelo; en el lado frontal del mismo. Su función es detectar los cambios de tonalidades que se presenten en el transcurso de su camino.

El Sensor tiene un valor definido en escala de grises y cada vez que detecte el dato, este será registrado como la variable light en la programación. Se debe tener muy en cuenta que el sensor debe ser colocado lo más próximo al suelo y estar fijo para así evitar variaciones en la toma de datos.

Fig. 3.1.2.1: Ubicación del Sensor de Luz en el Robot

3.1.2.2 Utilización del Sensor de Ultrasonido

El Sensor de Ultrasonido se encuentra ubicado en la parte superior izquierda del robot; en el lado frontal del mismo y a unos 20 centímetros del suelo. Su función es detectar la presencia de algún objeto que obstaculice el recorrido normal del robot.

El Sensor tiene un valor definido por la distancia a la que se encuentre con respecto a un determinado objeto y es almacenada en la variable dist. Se debe considerar el tamaño de los objetos que servirán como obstáculos para la ubicación correcta del sensor.

3.1.2.3 Utilización del Sensor de Tacto

El sensor de tacto se encuentra ubicado en la parte superior derecha, del lado posterior del robot. Su función es de interrumpir la comunicación entre el robot y el ordenador, así al presionarlo el robot se detiene.

Su activación es reflejada en la variable touch. Debe ser ubicado en un lugar de fácil acceso para el usuario.

Fig. 3.1.2.3: Ubicación de los Sensores de Tacto y Ultrasonido

3.2 Diseño de la Pista

El modelo de la pista debe ser diseñado de tal manera que permita realizar las pruebas para obtener los resultados pertinentes con respecto al objetivo del proyecto.

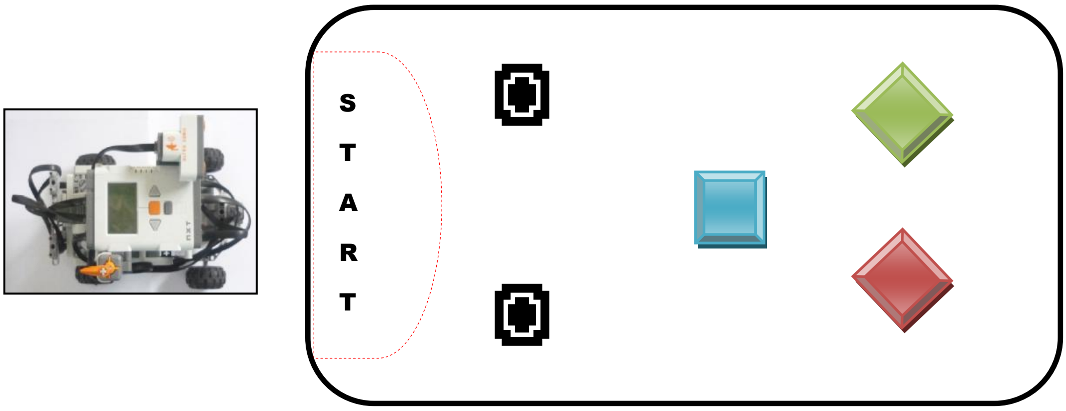


Fig. 3.1.2.3: Diseño de la Pista

El tamaño aproximado de la misma es de 1.5 m^2 cuya característica es que posee una franja negra de 3 cm. de grosor en su contorno. Y en la parte interna tendrá obstáculos de diferentes formas con un alto mayor a los 20cm. También posee franjas de diferentes colores en el interior con fines de comparación, registro y análisis del proyecto en estudio. En uno de sus extremos se indica la entrada “ START ”.

3.3 Funcionamiento del Explorador

Inicialmente, el explorador (NXT) recibe el programa elaborado en Matlab, en forma inalámbrica a través de la tecnología bluetooth. Luego el explorador se moviliza con la ayuda del servomotor C en forma sincronizada e indefinidamente.

En el transcurso del recorrido el sensor de luz va registrando datos en un rango de 0-1024 / 10bits, los cuales varían de acuerdo a las diferentes tonalidades de la escala de grises. El recorrido normal es para valores inferiores a 500. Una vez que detecte un valor mayor al antes mencionado, que correspondería a los colores oscuros el programa desactiva al servomotor C.

Con la desactivación del servo C, el robot recibe la orden de energizar al servomotor B, el cual girará durante 1.0 s. dicho giro producirá la elevación del cuerpo del robot. Una vez que se cumpla este tiempo el servo B se detendrá y simultáneamente se accionará el servomotor A.

El servo A girará durante 1.2 s permitiendo el giro del robot sobre su propio eje, con el propósito de cambiar de dirección. Transcurrido este tiempo el servo A se deshabilita mientras el servo B vuelve a activarse produciendo un giro contrario al realizado anteriormente con la finalidad de regresar el cuerpo del robot a su altura original. En el anexo A se detalla el Diagrama ASM del funcionamiento del explorador y se analiza el significado de cada una de sus entradas y salidas.

Fig. 3.3: Reacción del robot ante un obstáculo

El sensor de luz como el ultrasónico se encuentran captando datos en todo momento los cuales son almacenados en sus respectivas variables. Una vez que el robot termina de ejecutar el proceso anteriormente descrito y se encuentra con la presencia de un nuevo obstáculo o está a punto de ingresar hacia un terreno de color oscuro vuelve a realizar el mismo proceso hasta encontrar las condiciones apropiadas para su libre recorrido.

La comunicación entre el robot y el ordenador será continua en todo momento; sin embargo puede ser interrumpida por el usuario al presionar el sensor de tacto sólo en el instante en el que las llantas del robot vuelven a topar el suelo después de realizar el cambio de dirección.

3.4 CÓDIGO DE PROGRAMACIÓN EN MATLAB

```
%% Clean up
% Close previous handles (if existing)
COM_CloseNXT all
```

```
%% Set up Matlab
clear all
close all
format compact
```

```
%% Banderas
a=0;
b=0;
c=0;
d=0;
m=0;
t=0;
```

```
%% Set up ports
portUS       = SENSOR_1;    % SENSOR DE LUZ
portUS2      = SENSOR_4;    % SENSOR DE ULTRASONICO
portUS3      = SENSOR_3;    % SENSOR DE TACTO
portMotor    = MOTOR_A;     % MOTOR BASE
portMotor2   = MOTOR_B;     % MOTOR PALANCA
portMotor3   = MOTOR_C;     % MOTOR DE MOVIMIENTO
MotorSpeed  = 100;
```

```
%% Open connection
handle = COM_OpenNXT('bluetooth.ini','check');
COM_SetDefaultNXT(handle);
```

```
%% SENSOR ULTRASONICO ***ABRIR COMUNICACION****
 OpenUltrasonic(SENSOR_4);
```

```
%% SENSOR DE LUZ ***ABRIR COMUNICACION****
 OpenLight(SENSOR_1,'ACTIVE');
```

```
while true

while true
     t=t+1;
    light(t)=GetLight(SENSOR_1)
    m=m+1;
    tiempo(t)=m;
    dist(t)=GetUltrasonic(SENSOR_4)
```

```
    SetMotor(MOTOR_C);
    SetPower(50);
    SendMotorSettings();
```

```
  if ((GetLight(SENSOR_1)<500)||(GetUltrasonic(SENSOR_4)<30))
      a=1;
end
```

```
%% **Graficas**
    subplot(2,1,1)
    plot (t,light(t),'-rs','LineWidth',2,'MarkerEdgeColor','k','MarkerFaceColor','g','MarkerSize',5)
    title('Grafica Sensor de LUZ')
    xlabel('tiempo')
    ylabel('escala de grises')
    grid on
    hold on


    subplot(2,1,2)
    plot (t,dist(t),'-bs','LineWidth',2,'MarkerEdgeColor','k','MarkerFaceColor','g','MarkerSize',5)
    title('Grafica Sensor de Presencia')
    xlabel('tiempo')
    ylabel('Distancia')
    grid on
    hold on

while a==1
    StopMotor(MOTOR_C, 'brake');
    pause(0.5) %retardo de 0.5 segundo

    SetMotor(MOTOR_B);
    SetPower(75);
    SendMotorSettings();
    pause(1)
    StopMotor(MOTOR_B, 'brake');
```

```
    c=1;
    break
end

while c==1
    pause(0.5) %retardo de 0.5 segundo
    SetMotor(MOTOR_A);
    SetPower(90);
    SendMotorSettings();
    pause(1.2)
    StopMotor(MOTOR_A, 'off');
    d=1;
   break
end

while d==1
```

```
 %% SENSOR TACTO ABRIR COMUNICACION
   OpenSwitch(SENSOR_3);
   touch = GetSwitch(SENSOR_3)
```

```
 %% Regreso de palanca
    pause(0.5) %retardo de 0.5 segundo
    SetMotor(MOTOR_B);
    SetPower(-58);
    SendMotorSettings();
    pause(1)
    StopMotor(MOTOR_B, 'brake');
    b=1;
    c=0;
    a=0;
    break

   end
```

```
if (b==1) %cerrar comunicacion USB
    b=0;
    d=0;
end
```

```
%% SENSOR TACTO ABRIR COMUNICACION
   OpenSwitch(SENSOR_3);
   touch = GetSwitch(SENSOR_3)
if (touch==1)


    light = light'
    xlswrite('datos.xls',light, 'finalgraf1', 'D4')
    tiempo= tiempo'
    xlswrite('datos.xls',tiempo, 'finalgraf1', 'C4')
    dist=dist'
    xlswrite('datos.xls',dist, 'finalgraf1', 'E4')


COM_CloseNXT(handle);

end

end
end
```

3.4.1 DETALLE DE COMANDOS USADOS

Dentro del toolbox que brinda el Instituto de Imagen y Visión RWTH, se encuentran las herramientas necesarias entre comandos, demos y variables que permiten al programador diseñar diversos códigos abiertos al desarrollo de funciones de alto nivel en Matlab.

Una de las cosas más importantes dentro del código es fijar el modo de envío de la información ya sea por red inalámbrica bluetooth o USB.

El siguiente comando permite la conexión vía USB. El primer puerto que éste encuentre será usado.

```
handle = COM_OpenNXT()
```

Mientras que al comunicarse en forma inalámbrica se define `'bluetooth.ini'` y al ejecutarse el comando se configurará el puerto de conexión para no cometer errores en el momento de envió de la información.

```
handle = COM_OpenNXT('bluetooth.ini','check');
```

Para inicializar el modo de entrada de los diferentes sensores que se encuentran conectados al bloque NXT se especifican los comandos siguientes y se detalla el puerto en el que se encuentra direccionado.

- `OpenUltrasonic(SENSOR_4);` El sensor ultrasónico se conecta al puerto 4 del bloque NXT.
- `OpenSwitch(SENSOR_3);` El sensor de contacto se conecta al puerto 3 del bloque NXT.
- `OpenLight(SENSOR_1,'ACTIVE');` El sensor de luz se conecta al puerto 1 del bloque NXT y estará activado hasta que otra señal lo desactive.

La activación de cada uno de los servomotores está definida de la siguiente manera:

```
SetMotor(MOTOR_C);
%       SetPower(76);
%       SetAngleLimit(720);
SendMotorSettings();
```

El código permite la activación del motor, seteando constantemente el valor de corriente enviado al puerto al que se encuentra conectado el servo. Se establece una potencia de acuerdo a la aplicación que se realice y su giro puede ser limitado por un valor angular o por un tiempo de ejecución.

CAPITULO 4

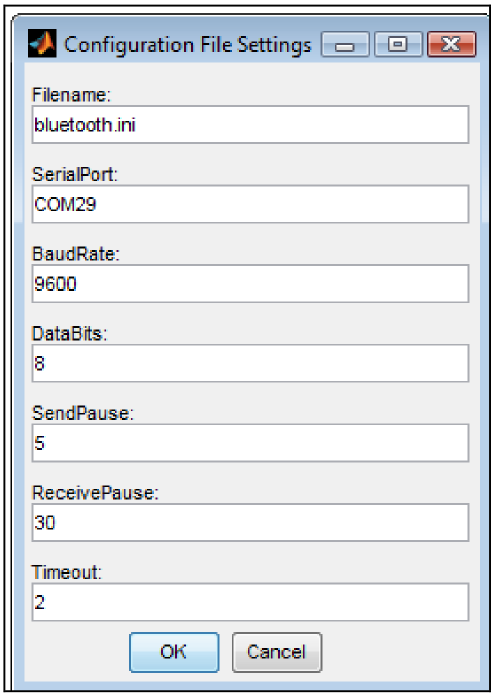
4 SIMULACIÓN, PRUEBAS Y DATOS EXPERIMENTALES

4.1 Transferencia de Datos

Para la interacción entre el ordenador y el NXT se consideró el uso de la tecnología bluetooth, siendo este el más apropiado para el propósito de este proyecto; ya que la comunicación a través de USB limita el funcionamiento independiente considerando que el control del mismo será operado en tiempo real utilizando una capacidad de memoria insignificante cuya ventaja será aprovechada por las múltiples transmisiones y adquisiciones de datos.

Matlab proporciona un toolbox que facilita y habilita la comunicación entre el ordenador y el NXT. En el interior del mismo se encuentra un comando que permite modificar los parámetros necesarios para el uso del bluetooth. Se lo habilita desde el Command Window de Matlab ejecutando >>COM_MakeBTConfigFile. inmediatamente se presenta una ventana para la configuración respectiva.

Entre los parámetros necesarios que se requieren, se destaca: la configuración del puerto serial el cual varía según el puerto de comunicación que el usuario utilice a través de un dispositivo externo, la velocidad de transmisión de datos, los bits de datos transmitidos y los tiempos de retardo de envió y recepción de información.



Filename: bluetooth.ini
SerialPort: COM29
BaudRate: 9600
DataBits: 8
SendPause: 5
ReceivePause: 30
Timeout: 2

Fig. 4.1: Configuración de parámetros en Matlab

4.2 Prueba de Sensores y Servomotores

Las prueba que se realiza a los sensores y servomotores son en forma individual mediante la cual se analiza el funcionamiento del sensor ante una señal externa. También se puede adquirir los datos necesarios para el condicionamiento del proyecto.

4.2.1 Servomotores

En la prueba realizada a los servos, se observa la variación de velocidades con respecto a su frecuencia. Al motor A se le asigna un valor menor de frecuencia por esto su rotación es más rápida. Caso contrario sucede con el Motor C, donde su rotación es lenta debido al alto valor de frecuencia asignado.

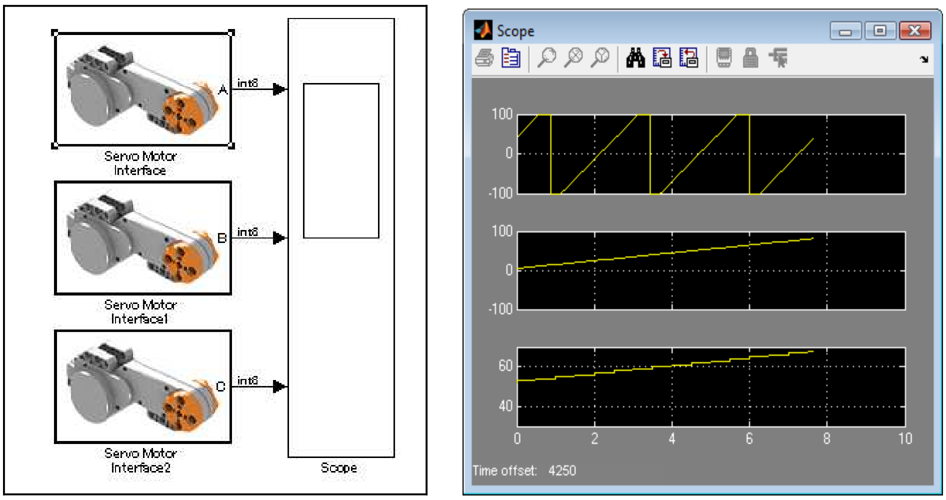


Fig. 4.2.1: Prueba para los Servomotores

4.2.2 Sensores

Mediante el análisis de los sensores se define que el de tacto envía una señal digital dada por 1 o 0, el sensor de luz registra datos de 0 -1024 / 10bits y el ultrasónico valores en decibeles.

4.3 Adquisición de Datos

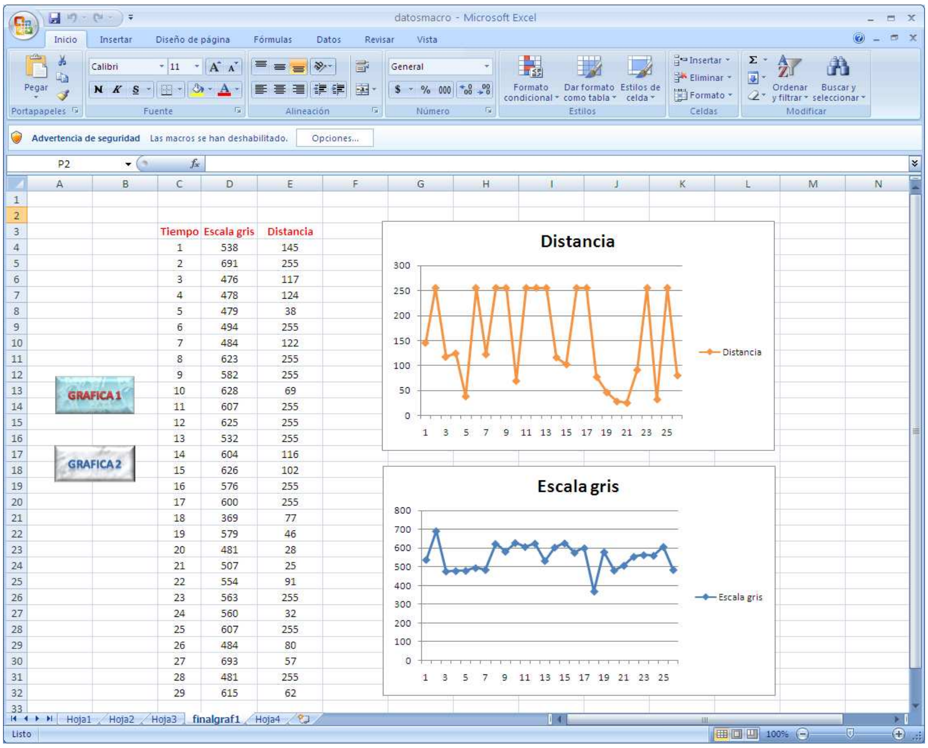
Se tiene una visión amplia hacia campos que requieren ser controlados mediante un registro de datos, entonces la meta ha sido enfocar el proyecto hacia un campo real donde el almacenamiento de los mismos permita analizar y mejorar el control que realiza la aplicación.

El proceso de adquisición de datos se lo realiza cuando el proyecto está en ejecución y los sensores se encuentran activados, los datos detectados por el explorador con la ayuda de sus dispositivos son enviados hacia el ordenador con Matlab ejecutando. La información que adquiere Matlab es enviada mediante el comando >> xlswrite `('datos.xls', light, 'finalgraf1', 'D4')` `a Excel.` Se detalla las variables que se definen en el comando:

- `'datos.xls':` Es el nombre del archivo en Excel.

- `light:` Variable en forma de matriz que contiene datos del sensor analizado.
- `'finalgraf1':` Nombre de la hoja de datos de Excel.
- `'D4':` Ubicación donde comienza a registrar el primer dato adquirido.

Para la implementación de la base de datos requerimos los valores registrados por los sensores de luz y ultrasónico en tiempo determinados; ya que estos son importantes para la ejecución y objetivo del proyecto.



Advertencia de seguridad Las macros se han deshabilitado. Opciones...

Tiempo	Escala gris	Distancia
1	538	145
2	691	255
3	476	117
4	478	124
5	479	38
6	494	255
7	484	122
8	623	255
9	582	255
10	628	69
11	607	255
12	625	255
13	532	255
14	604	116
15	626	102
16	576	255
17	600	255
18	369	77
19	579	46
20	481	28
21	507	25
22	554	91
23	563	255
24	560	32
25	607	255
26	484	80
27	693	57
28	481	255
29	615	62

Fig. 4.3: Prueba de adquisición de datos

Luego de ejecutar el comando antes mencionado en Matlab, se puede verificar en Excel los datos registrados en una de las pruebas realizadas. Cada dato es registrado durante un segundo de tiempo. En el anexo C, se encuentran los datos obtenidos de una segunda prueba.

Posteriormente, se realiza una macro función en Excel, la cual permite visualizar los datos adquiridos en forma gráfica, ésta es diferenciada con los resultados proporcionados por Matlab. Esta herramienta es útil para el análisis del funcionamiento correcto del explorador. En el anexo D, se muestran las gráficas determinadas en la segunda prueba.

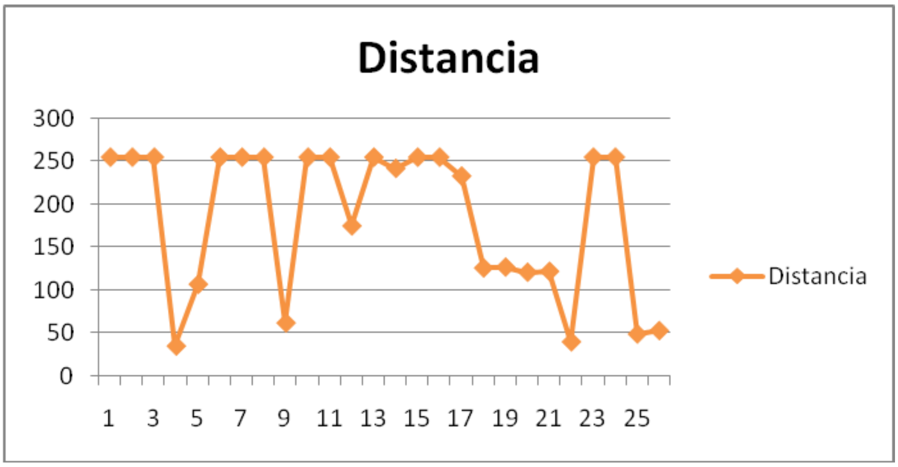


Fig. 4.3a: Valores registrados en Excel por el sensor ultrasónico. Prueba 1

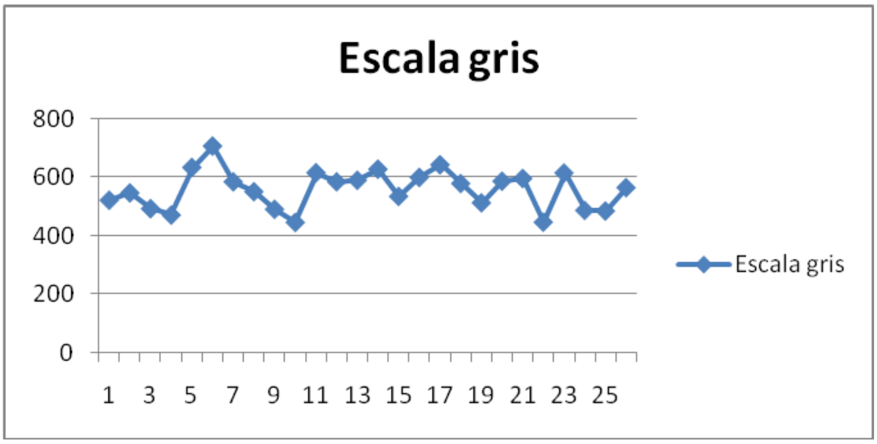


Fig. 4.3b: Valores registrados en Excel por el sensor de luz. Prueba 1

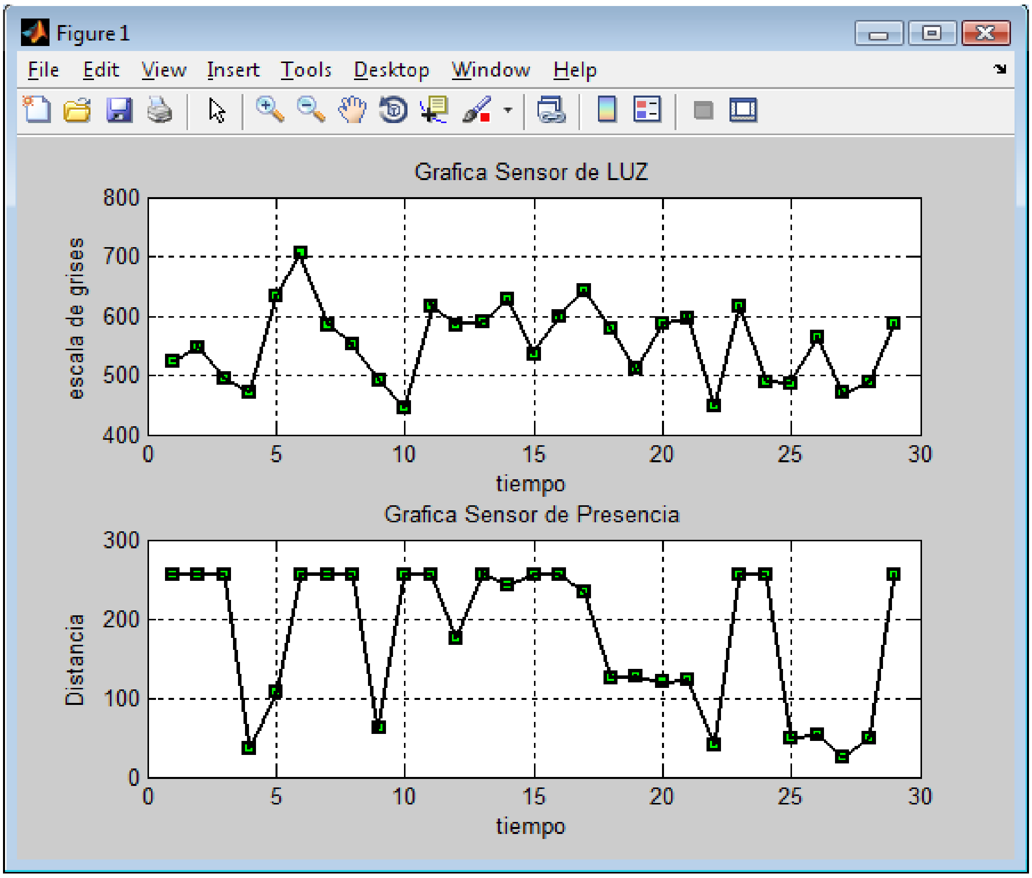


Fig. 4.3: Valores en Matlab d el sensor de luz y ultrasónico. Prueba 1

4.4 PRACTICA REALIZADA ANTES DEL PROYECTO FINAL

Introducción

Esta práctica consistió en realizar un proyecto con fines de aprendizaje con respecto a Lego Mindstorms NXT y con la utilización de su software. El proyecto realizado se trata de un robot buscador de franjas negras y cuyo objetivo es la detección y el rechazo de ésta.

El pequeño explorador consiste de una estructura similar al proyecto de graduación en curso cuya diferencia es que este posee dos tipos de sensores: tacto y de luz.

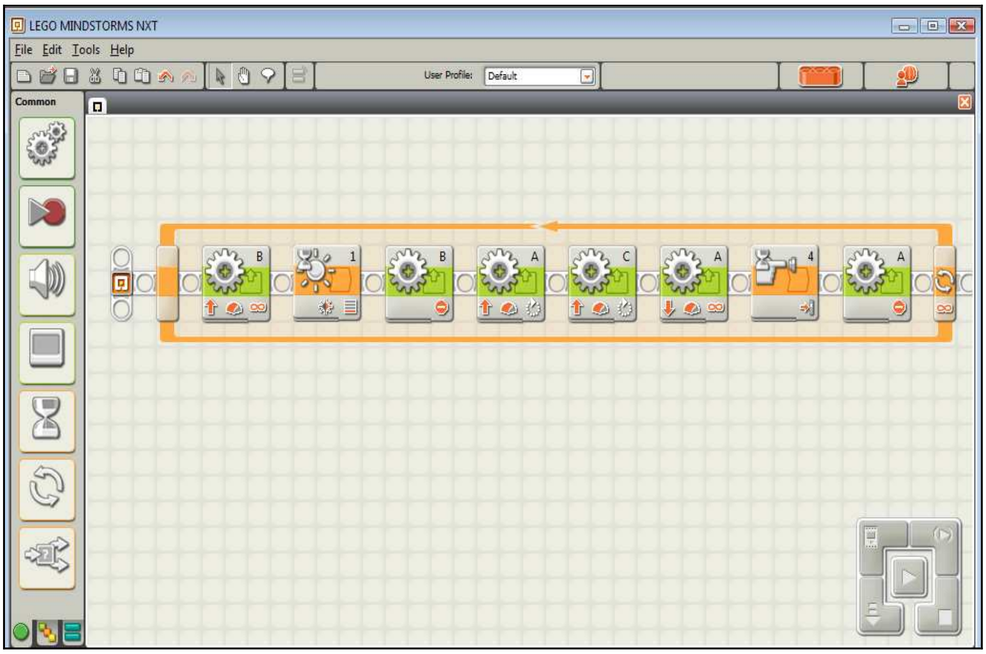
Como campo de acción, se utilizó la pista que el mismo kid de Lego proporciona cuya área es factible y amigable al ambiente de Lego.

Proceso del programa

El sistema se encuentra manejado por un lazo while, iniciando el encendido del motor B quien mueve el explorador. Después pregunta si se detecta franjas de color negro, si es así para el motor B y se enciende el motor A en sus grados respectivo hasta cuento este completo.

El siguiente paso será en encender el motor C, ya que en este caso observamos en el proyecto el movimiento circular a sus grados respectivos y enseguida se enciende de nuevo el motor A en sentido opuesto hasta que el sensor de tacto detecte el tope y luego el programa inicia otra vez dándole al proyecto nuevamente la movilidad

Entorno gráfico



User Profile: Default

Fig. 4.4:   Programación en Software LEGO MINDSTORMS NXT

Parámetros principales

Motor B: Posee parámetros como una potencia de 75 (power) y en forma indefinida.

Motor A: Posee parámetros como una potencia de 75 (power) a 600 grados en brake, se utiliza en dos sentidos, cuando de la orden subir y luego de bajar.

Motor C: Posee parámetros como una potencia de 75 (power) a 630 grados en brake.

Estructura

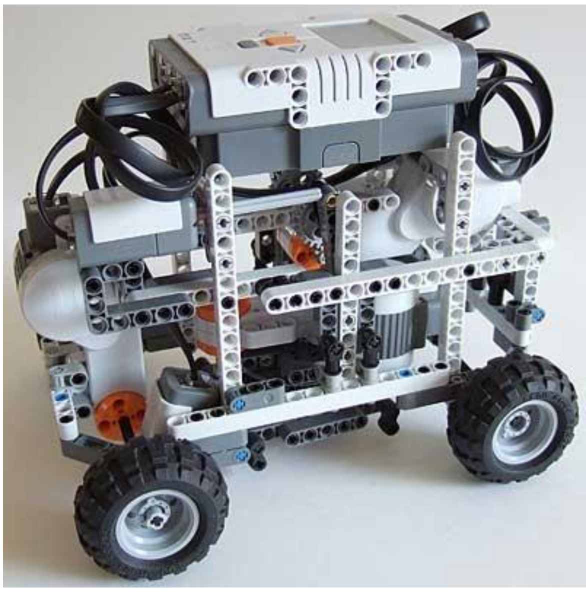


Fig. 4.4: Estructura Final de Cicló

Conclusiones

1. Se logró implementar un explorador con la capacidad de detectar franjas o líneas de tonalidades oscuras y evitar el choque contra algún objeto cercano. Debido a las propiedades que necesita el explorador y aprovechando las ventajas que nos brinda el bloque NXT, se decidió comunicar el ordenador con el NXT a través de la tecnología bluetooth.

2. Se aplicaron las bases adquiridas y se mejoraron los conocimientos en cuanto a la programación en Matlab que es un programador de alta calidad para el desarrollo de extensas aplicaciones en diversos campos tecnológicos. Se verificó las limitaciones que presenta simulink en cuanto al retardo significativo en la toma de datos en tiempo real.

3. Durante la implementación del proyecto, se pudo visualizar las diferentes formas en que el Lego Mindstorms NXT permite ser manipulado, esto facilita y amplía el campo de aplicaciones en las que puede ser utilizado. Así como el lego puede ser un sencillo juego para niños, también puede llegar a ser una parte muy significativa para cualquier proceso industrial.

4. Es factible la adquisición de datos que se analizan en Matlab hacia otro software, en este caso hacia Excel, que nos recopila la información y permite el estudio y análisis de las gráficas que se pueden realizar aquí a través de una macro función con las ejecutadas directamente en Matlab.

5. Los motores de giro y palanca son condicionados a realizar giros con respecto a un breve retardo de tiempo, ya que esto hace que el giro sea un poco preciso en tiempo y posición, porque en el programa cuando se le aplica a grados este tiene un poco a perder su posición ya que se utiliza sistema de engranaje para proporciónale a cada motor fuerza. Eso implico buscar una solución en la posición final para el desplazamiento del objeto, base o llantas que se desea aplicar en su debido tiempo y situación.

6. Una de las ventajas que presenta la codificación en Matlab es el realizar extensos programas que no ocupan memoria en el bloque NXT, permitiendo realizar numerosas pruebas sin eliminar archivos constantemente y aplicar funciones extensas sin que su capacidad de memoria sea limitada. Mientras que un archivo de Simulink al ser ejecutable limitaría la memoria del NXT.

Recomendaciones

1. Al momento de diseñar el robot se debe considerar una ubicación adecuada para cada uno de los sensores teniendo en cuenta que los datos recibidos por éstos pueden variar con respecto a los reales. Su ubicación debe ser fija y de fácil acceso para el usuario. Además se debe tener en cuenta los parámetros de cada uno con sus respectivas limitaciones.

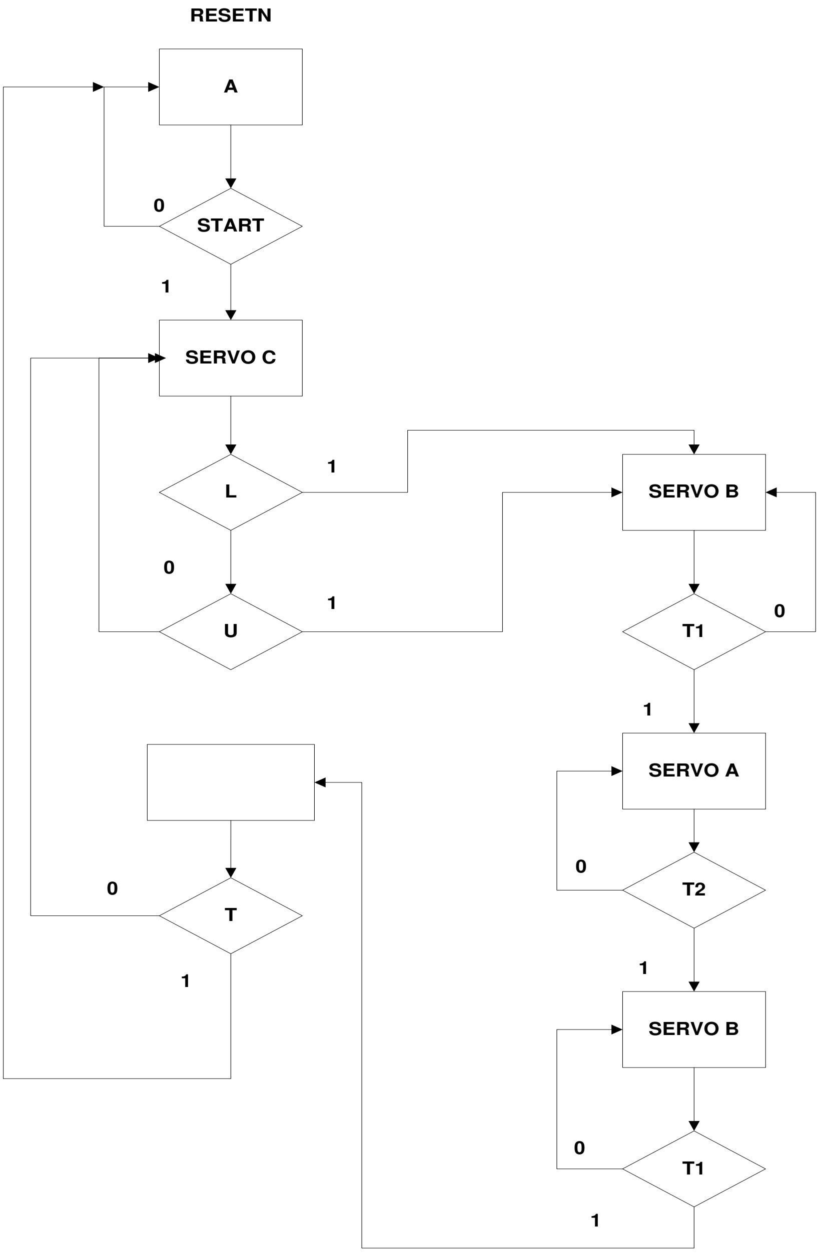
2. Al realizar la comunicación vía Bluetooth tener presente el puerto de comunicación asignado al ordenador y los parámetros correspondientes a la velocidad de recepción y envío de datos. Cualquier dispositivo externo de Bluetooth puede ser utilizado en el ordenador para lograr la comunicación con el NXT; sea el propio de LEGO o de otra marca.

3. En el momento de la construcción del explorador se debe colocar las piezas de forma tal que en el momento del giro el centro de masa del mismo esté encerrado en el área limitada por sus ruedas, evitando de esta manera que el explorador pierda su equilibrio. Considerar la potencia máxima que soporta cada servomotor.

4. Es de gran ayuda colocar comentarios en las líneas fundamentales de la programación pues esto será útil tanto para el programador como para el usuario.

ANEXO A

DIAGRAMA ASM DEL CONTROLADOR



ESPECIFICACIONES DEL ASM

START: Señal que habilita la comunicación entre el ordenador y el NXT cuando el programa ‘ Mfile ’ se ha ejecutado.

U: Señal que representa al sensor ultrasónico (aproximación), cual ejecuta la siguiente función si el valor registrado es igual o no al establecido.

T: Señal que representa al sensor de tacto, cual ejecuta la siguiente función si ha sido pulsado.

L: Señal que representa al sensor de luz, cual ejecuta la siguiente función si el valor registrado es igual o no al establecido.

T1: Señal que indica el tiempo que el servomotor B estará en movimiento.

T2: Señal que indica el tiempo que el servomotor A estará en movimiento.

A: Señal que indica que no existe comunicación entre el ordenador y el NXT.

MOTOR A: Señal que indica que el servo A (base) está en movimiento.

MOTOR B: Señal que indica que el servo B (palanca) está en movimiento.

MOTOT C: Señal que indica que el servo C (movimiento) está en movimiento.

ANEXO B

DATOS OBTENIDOS EN MATLAB DE LA PRUEBA 2

```
light =
  538
dist =
  145
touch =
    0
touch =
    0
light =
  538   691
dist =
  145   255
touch =
    0
touch =
    0
light =
  538   691   476
dist =
  145   255   117
touch =
    0
touch =
    0
light =
  538   691   476   478
dist =
  145   255   117   124
touch =
    0
touch =
    0
```

```
light =
  538   691   476   478   479
dist =
  145   255   117   124    38
touch =
    0
touch =
    0
light =
  538   691   476   478   479   494
dist =
  145   255   117   124    38   255
touch =
    0
touch =
    0
light =
  538   691   476   478   479   494   484
dist =
  145   255   117   124    38   255   122
touch =
    0
touch =
    0
light =
  538   691   476   478   479   494   484   623
dist =
  145   255   117   124    38   255   122   255
touch =
    0
touch =
    0
light =
  538   691   476   478   479   494   484   623   582
dist =
  145   255   117   124    38   255   122   255   255
touch =
    0
```

```
touch =
     0
light =
   538   691   476   478   479   494   484   623   582   628
dist =
   145   255   117   124    38   255   122   255   255    69
touch =
     0
touch =
     0
light =
   538   691   476   478   479   494   484   623   582   628   607
dist =
   145   255   117   124    38   255   122   255   255    69   255
touch =
     0
touch =
     0
light =
   538   691   476   478   479   494   484   623   582   628   607   625
dist =
   145   255   117   124    38   255   122   255   255    69   255   255
touch =
     0
touch =
     0
light =
   538   691   476   478   479   494   484   623   582   628   607   625   532
dist =
   145   255   117   124    38   255   122   255   255    69   255   255   255
touch =
     0
touch =
     0
light =
   538   691   476   478   479   494   484   623   582   628   607   625   532   604
dist =
   145   255   117   124    38   255   122   255   255    69   255   255   255   116
```

```
touch =
     0
touch =
     0
light =
   538   691   476   478   479   494   484   623   582   628   607   625   532   604   626
dist =
   145   255   117   124    38   255   122   255   255    69   255   255   255   116   102
touch =
     0
touch =
     0
light =
   538   691   476   478   479   494   484   623   582   628   607   625   532   604   626
576
dist =
   145   255   117   124    38   255   122   255   255    69   255   255   255   116   102
255
touch =
     0
touch =
     0
light =
   538   691   476   478   479   494   484   623   582   628   607   625   532   604   626
576   600
dist =
   145   255   117   124    38   255   122   255   255    69   255   255   255   116   102
255   255
touch =
     0
touch =
     0
light =
  Columns 1 through 17
   538   691   476   478   479   494   484   623   582   628   607   625   532   604   626
576   600
  Column 18
   369
```

```
dist =
  Columns 1 through 17
   145   255   117   124    38   255   122   255   255    69   255   255   255   116   102
255   255
  Column 18
    77
touch =
     0
touch =
     0
light =
  Columns 1 through 17
   538   691   476   478   479   494   484   623   582   628   607   625   532   604   626
576   600
  Columns 18 through 19
   369   579
dist =
  Columns 1 through 17
   145   255   117   124    38   255   122   255   255    69   255   255   255   116   102
255   255
  Columns 18 through 19
    77    46
touch =
     0
touch =
     0
light =
  Columns 1 through 17
   538   691   476   478   479   494   484   623   582   628   607   625   532   604   626
576   600
  Columns 18 through 20
   369   579   481
dist =
  Columns 1 through 17
   145   255   117   124    38   255   122   255   255    69   255   255   255   116   102
255   255
  Columns 18 through 20
    77    46    28
```

```
touch =
    0
touch =
    0
light =
  Columns 1 through 17
   538   691   476   478   479   494   484   623   582   628   607   625   532   604   626
576   600
  Columns 18 through 21
   369   579   481   507
dist =
  Columns 1 through 17
   145   255   117   124    38   255   122   255   255    69   255   255   255   116   102
255   255
  Columns 18 through 21
    77    46    28    25
touch =
    0
touch =
    0
light =
  Columns 1 through 17
   538   691   476   478   479   494   484   623   582   628   607   625   532   604   626
576   600
  Columns 18 through 22
   369   579   481   507   554
dist =
  Columns 1 through 17
   145   255   117   124    38   255   122   255   255    69   255   255   255   116   102
255   255
  Columns 18 through 22
    77    46    28    25    91
touch =
    0
touch =
    0
```

```
light =
  Columns 1 through 17
   538   691   476   478   479   494   484   623   582   628   607   625   532   604   626
576   600
  Columns 18 through 23
   369   579   481   507   554   563
dist =
  Columns 1 through 17
   145   255   117   124    38   255   122   255   255    69   255   255   255   116   102
255   255
  Columns 18 through 23
    77    46    28    25    91   255
touch =
     0
touch =
     0
light =
  Columns 1 through 17
   538   691   476   478   479   494   484   623   582   628   607   625   532   604   626
576   600
  Columns 18 through 24
   369   579   481   507   554   563   560
dist =
  Columns 1 through 17
   145   255   117   124    38   255   122   255   255    69   255   255   255   116   102
255   255
  Columns 18 through 24
    77    46    28    25    91   255    32
touch =
     0
touch =
     0
light =
  Columns 1 through 17
   538   691   476   478   479   494   484   623   582   628   607   625   532   604   626
576   600
  Columns 18 through 25
   369   579   481   507   554   563   560   607
```

```
dist =
  Columns 1 through 17
   145   255   117   124    38   255   122   255   255    69   255   255   255   116   102
255   255
  Columns 18 through 25
    77    46    28    25    91   255    32   255
touch =
     0
touch =
     0
light =
  Columns 1 through 17
   538   691   476   478   479   494   484   623   582   628   607   625   532   604   626
576   600
  Columns 18 through 26
   369   579   481   507   554   563   560   607   484
dist =
  Columns 1 through 17
   145   255   117   124    38   255   122   255   255    69   255   255   255   116   102
255   255
  Columns 18 through 26
    77    46    28    25    91   255    32   255    80
touch =
     0
touch =
     0
light =
  Columns 1 through 17
   538   691   476   478   479   494   484   623   582   628   607   625   532   604   626
576   600
  Columns 18 through 27
   369   579   481   507   554   563   560   607   484   693
dist =
  Columns 1 through 17
   145   255   117   124    38   255   122   255   255    69   255   255   255   116   102
255   255
  Columns 18 through 27
    77    46    28    25    91   255    32   255    80    57
```

```
touch =
    0
touch =
    0
light =
  Columns 1 through 17
   538   691   476   478   479   494   484   623   582   628   607   625   532   604   626
576   600
  Columns 18 through 28
   369   579   481   507   554   563   560   607   484   693   481
dist =
  Columns 1 through 17
   145   255   117   124    38   255   122   255   255    69   255   255   255   116   102
255   255
  Columns 18 through 28
    77    46    28    25    91   255    32   255    80    57   255
touch =
    0
touch =
    0
light =
  Columns 1 through 17
   538   691   476   478   479   494   484   623   582   628   607   625   532   604   626
576   600
  Columns 18 through 29
   369   579   481   507   554   563   560   607   484   693   481   615
dist =
  Columns 1 through 17
   145   255   117   124    38   255   122   255   255    69   255   255   255   116   102
255   255
  Columns 18 through 29
    77    46    28    25    91   255    32   255    80    57   255    62
touch =
    0
touch =
    1
light =
   538
```

```
   691
   476
   478
   479
   494
   484
   623
   582
   628
   607
   625
   532
   604
   626
   576
   600
   369
   579
   481
   507
   554
   563
   560
   607
   484
   693
   481
   615
 tiempo =
     1
     2
     3
     4
     5
     6
     7
     8
     9
```

```
    10
    11
    12
    13
    14
    15
    16
    17
    18
    19
    20
    21
    22
    23
    24
    25
    26
    27
    28
    29
dist =
   145
   255
   117
   124
    38
   255
   122
   255
   255
    69
   255
   255
   255
   116
   102
   255
   255
```

77
46
28
25
91
255
32
255
80
57
255
62

ANEXO C

TABLA DE ADQUISICIÓN DE DATOS DE LA PRUEBA 2

Tiempo	Escala gris	Distancia
1	538	145
2	691	255
3	476	117
4	478	124
5	479	38
6	494	255
7	484	122
8	623	255
9	582	255
10	628	69
11	607	255
12	625	255
13	532	255
14	604	116
15	626	102
16	576	255
17	600	255
18	369	77
19	579	46
20	481	28
21	507	25
22	554	91
23	563	255
24	560	32
25	607	255
26	484	80
27	693	57
28	481	255
29	615	62

ANEXO D

GRAFICAS DE LOS RESULTADOS DE LA PRUEBA 2

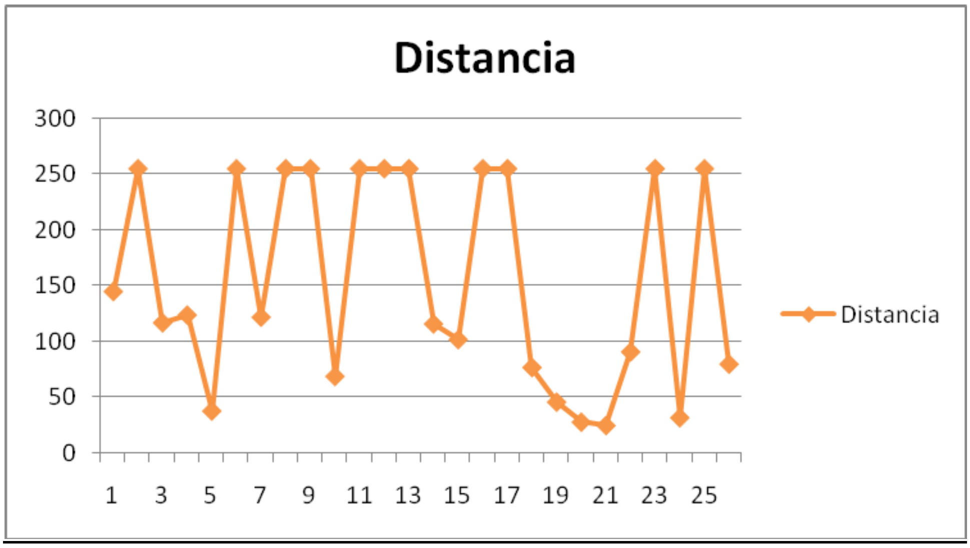


Foto 1: Distancias registradas

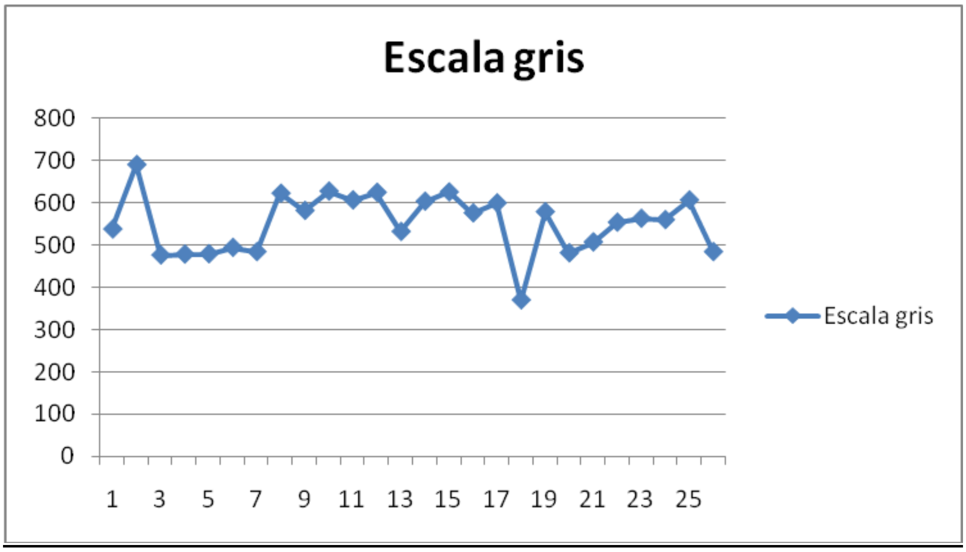


Foto 2: Niveles de luz registrados

GRAFICAS DE LOS RESULTADOS DE LA PRUEBA 2 EN MATLAB

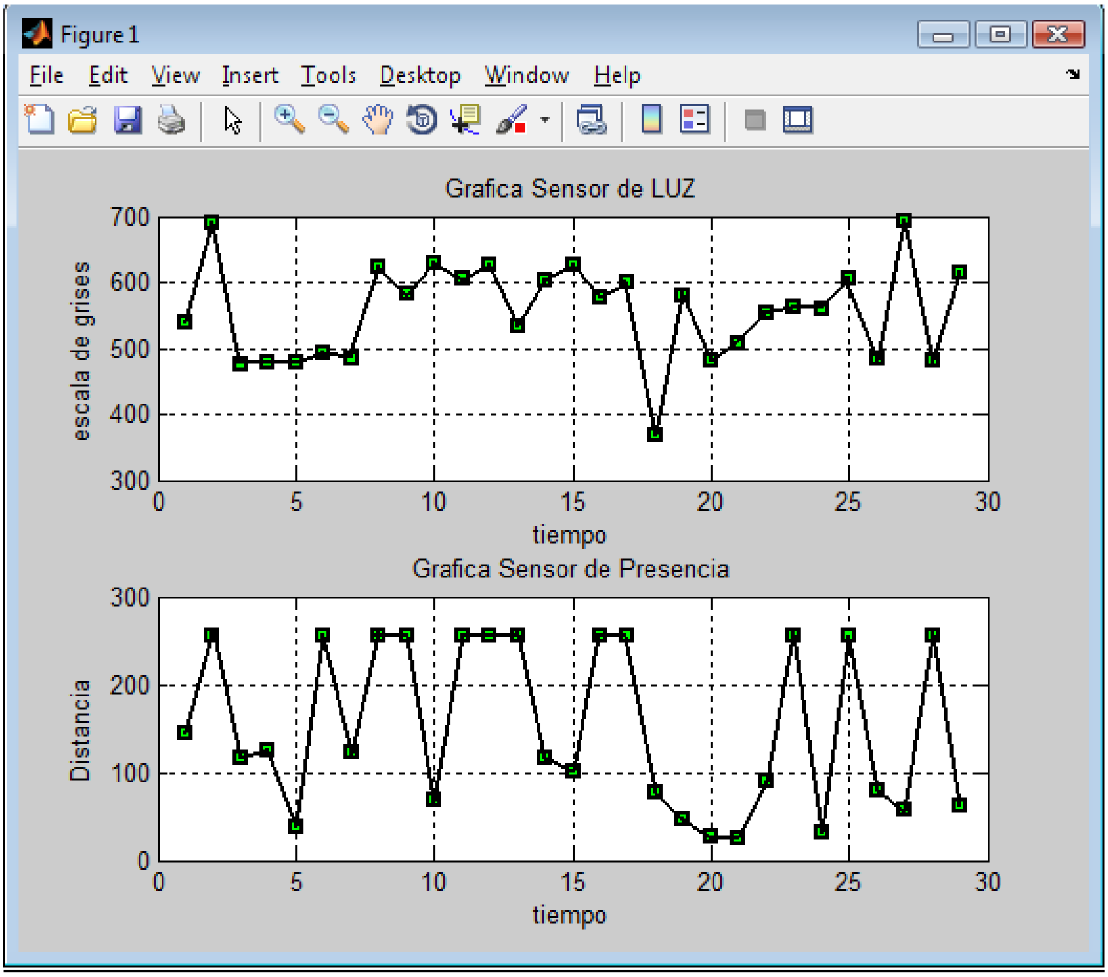
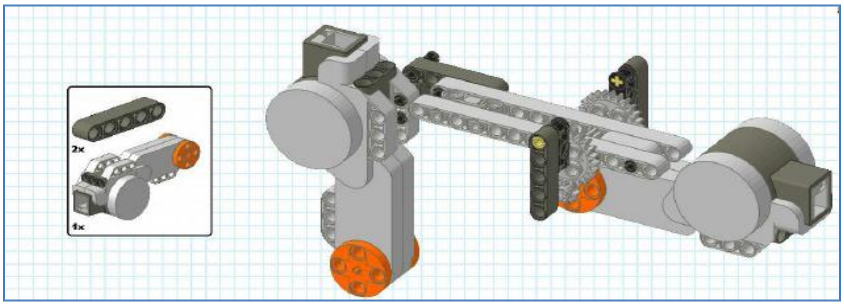


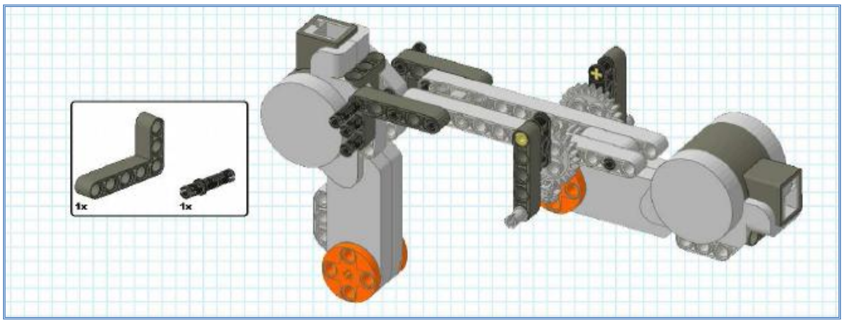
Foto 3: Registro de Datos en Matlab

ANEXO E

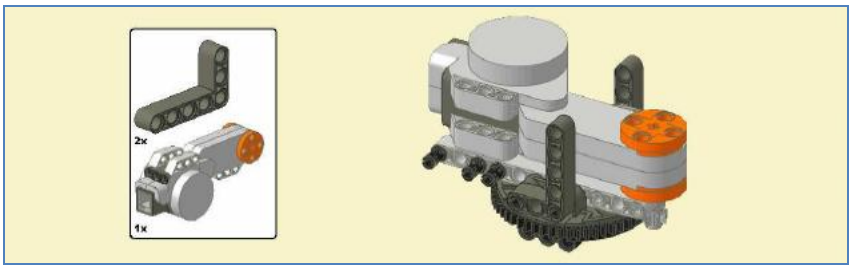
IMPLEMENTACIÓN PASO A PASO

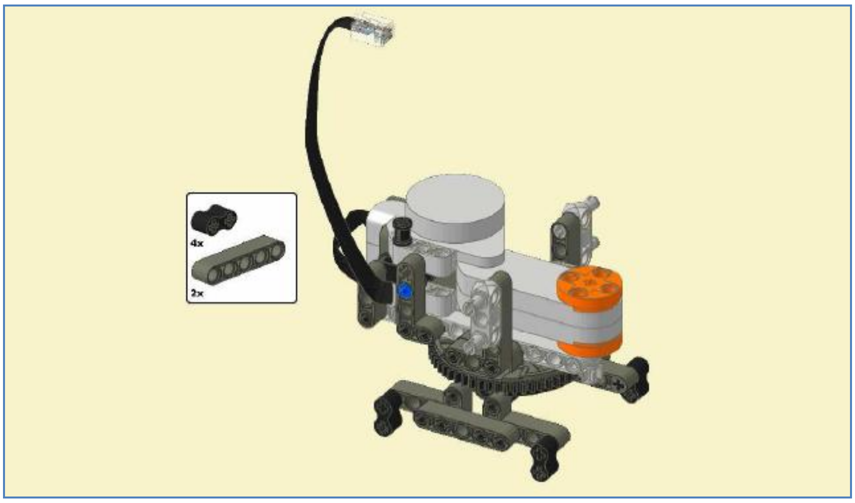
1. Paso: construcción y unión de la estructura de los motores de movimiento y palanca que se encuentra en la parte internar, ubicada en el centro del robot.



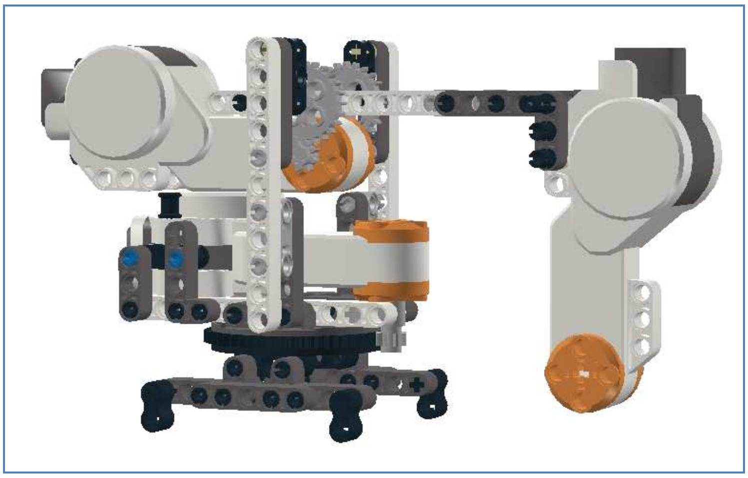


2. Paso: Construcción de la base del motor que ayuda a realizar su movimiento en forma horizontal, utilizando el motor base.



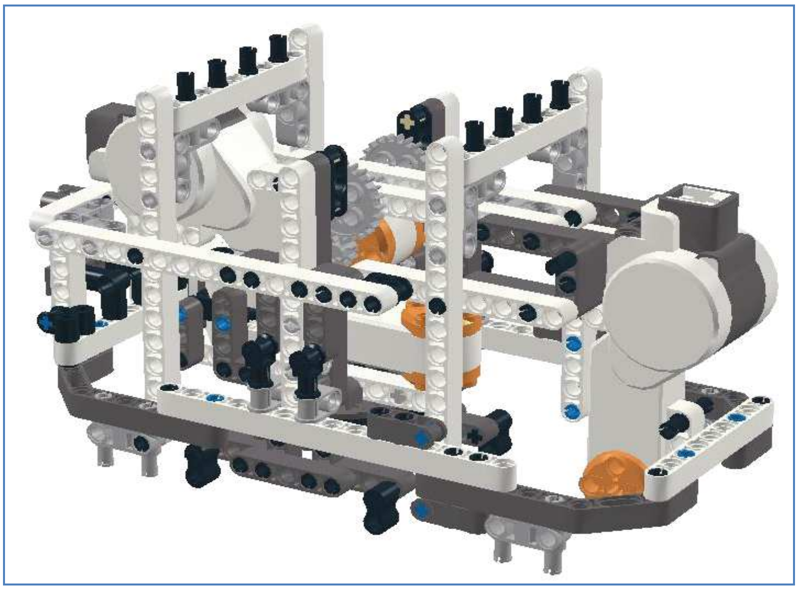


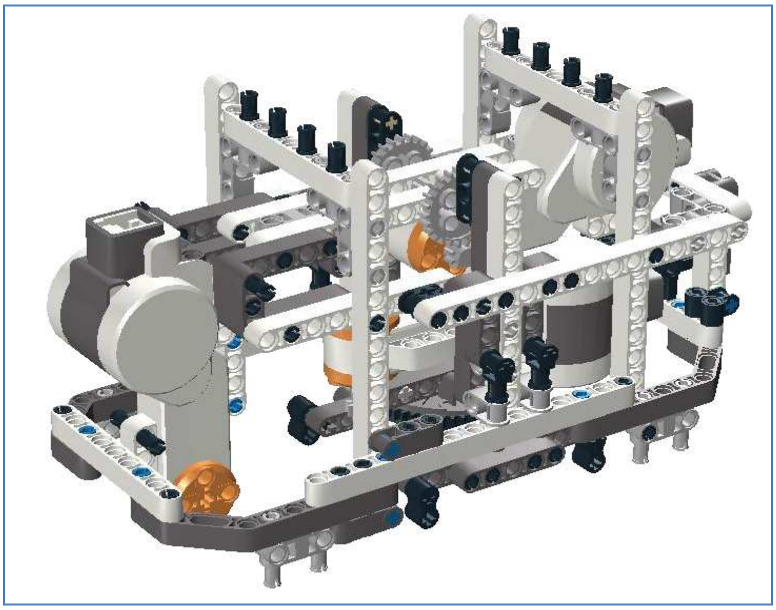
3. Paso: unión entre la parte de la construcción base y los motores (paso1 y paso2).



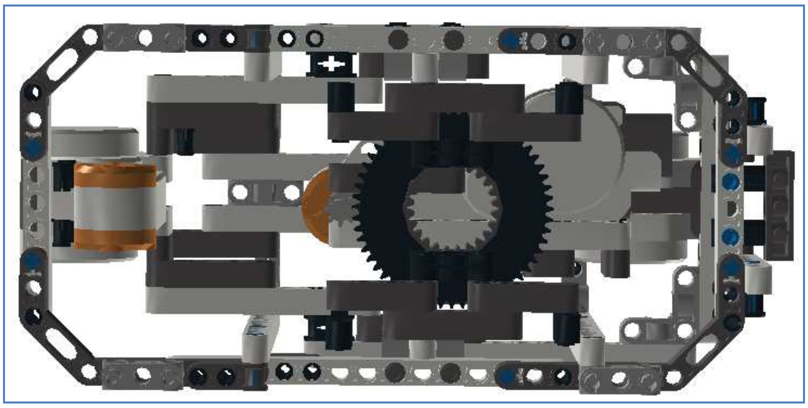
4. Paso: Armado de las partes laterales, que permiten la unión total del cuerpo y a la vez son útil para guiar el acenso y descenso de la base del motor.

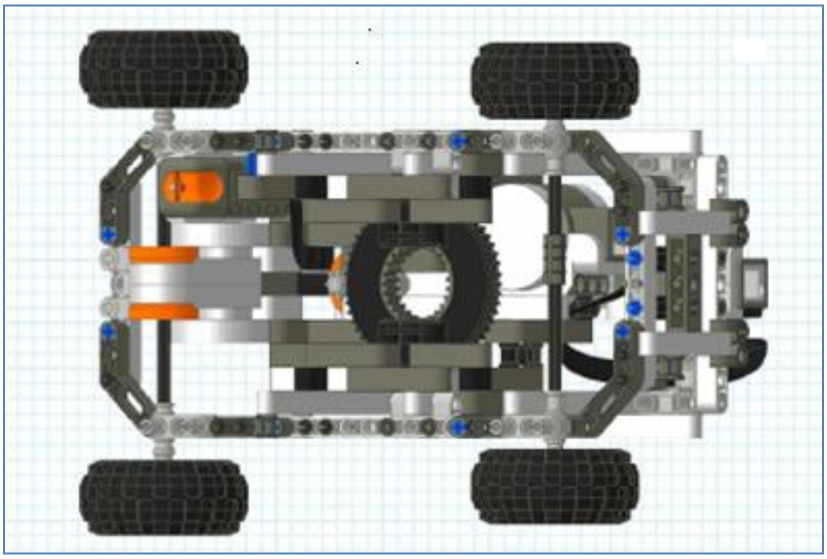
5. Paso: Construcción parte frontal y trasera, que sirven para la protección y equilibrio con fines a su movimiento horizontal y sobretodo vertical.

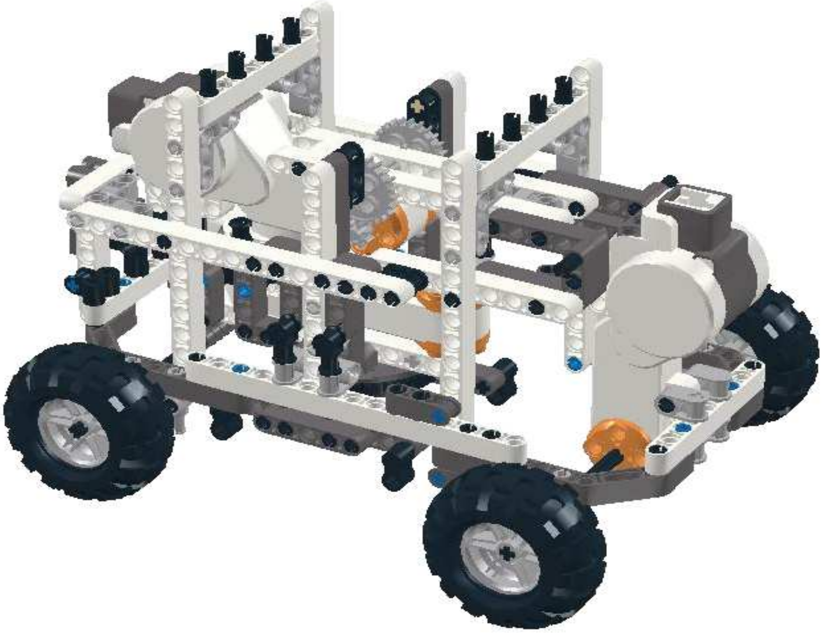


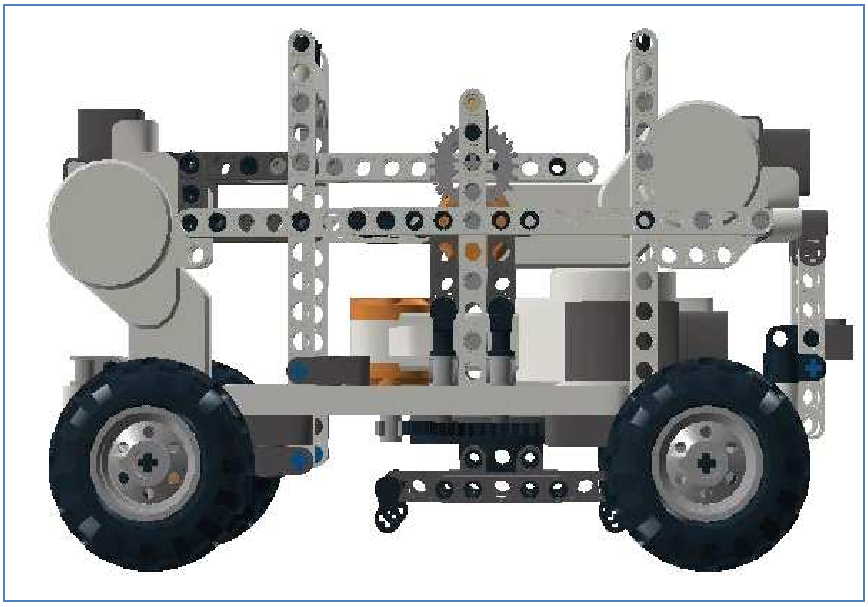


6. Paso: Construcción de las ruedas y ubicación de las misma.

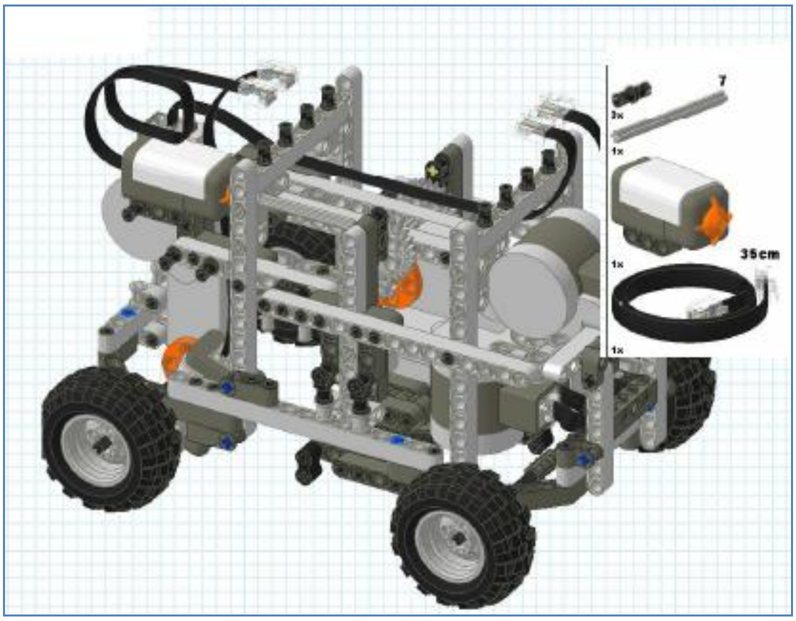


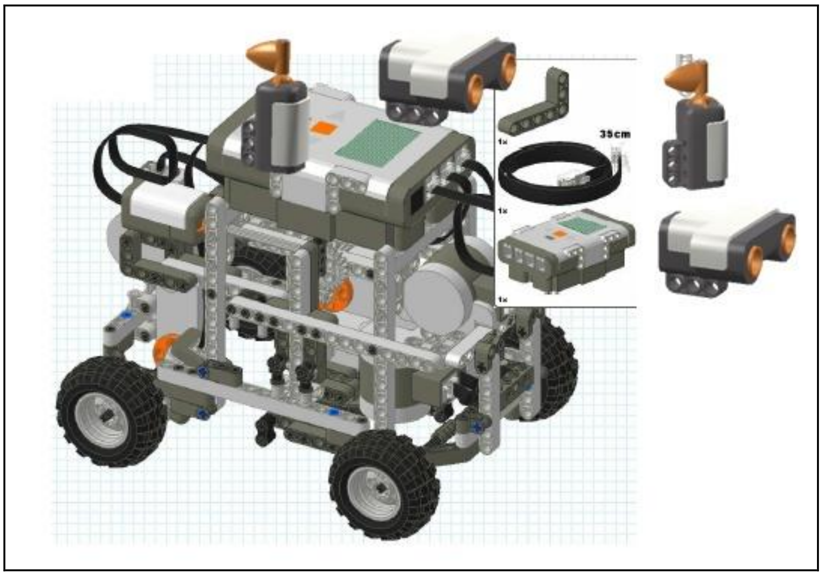




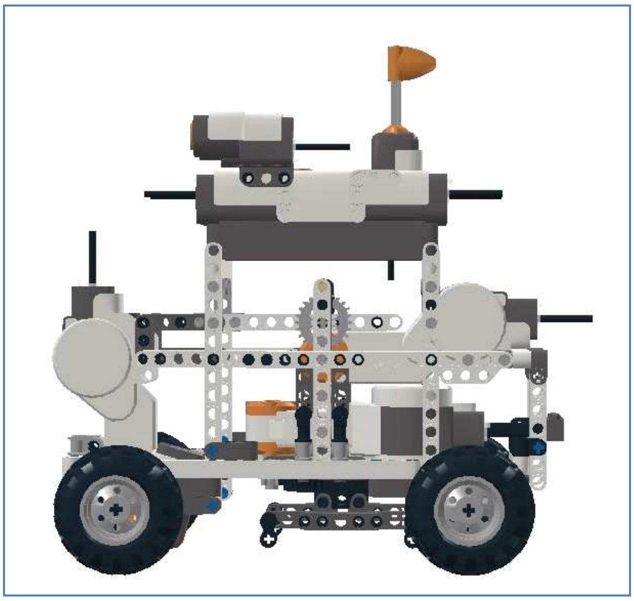


7. Paso: ubicación de los sensores y la conexión de los puertos de entrada de cada dispositivo en comunicación con el NXT.





8. Paso: Construcción final del proyecto explorador “CICLON”, en diferentes perspectivas.



BIBLIOGRAFÍA

[1]. Floyd Kelly James, *LEGO MINDSTORMS NXT-G Programming Guide,* edición Technology in Action Press, 2007

[2]. Gasperi Michael; Hurbain Philippe; Hurbain Isabelle, *EXTREME NXT Extending the LEGO MINDSTORMS NXT to the Next Level*, edición Technology in Action Press, 2007

[3]. Ferrari Mari; Ferrari Giulio; Hempel Ralph, *BUILDING ROBOTS WITH LEGO Mindstorm,*, edición Syngress Publishing. Inc.,2002

[4]. Estudio de las posibilidades didácticas en ingeniería de control del LEGO Mindstorms NXT, Archivo PDF, http://repositorio.bib.upct.es/dspace/bitstream/10317/822/1/pfc2797.pdf

[5]. Plataforma para la realización de trabajos prácticos de Mecatrónica basada en robots lego, Archivo PDF, http://www.cea-ifac.es/actividades/jornadas/XXIX/pdf/302.pdf

[6]. Introducción a Matlab y Simulink, Archivo PDF, http://www.esi2.us.es/~fabio/apuntes_matlab.pdf

[7]. Wikipedia, Lego Mindstorms, Página HTML, http://es.wikipedia.org/wiki/Lego_Mindstorms

[8]. Lego Mindstorms NXT en Español, Bluetooth, Página HTML, http://rbtnxt.blogspot.com/2009/01/bluetooth.html

[9]. Lego, Sensor de Tacto, Página HTML, http://mindstorms.lego.com/eng/Overview/Touch_Sensor.aspx

[10]. LEGO MINDSTORMS NXT Toolkit for MATLAB and Simulink, Página HTML, http://www.mathworks.com/programs/mindstorms/