ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

INFORME

DE MATERIA DE GRADUACIÓN

"Uso de Matlab y Simulink para el Control de
Robots y la Observación de Sensores de Tacto y Ultrasónico"

Previo a la obtención del Título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

INGENIERO EN ELECTRICIDAD ESPECIALIZACIÓN ELECTRÓNICA
Y AUTOMATIZACIÓN INDUSTRIAL

Presentada por:

Aguirre Patiño Martha Isabel
Cagua Ordoñez Jonathan
Criollo Ortiz Oswaldo Alfonso

GUAYAQUIL – ECUADOR

Año: 2009

A G R A D E C I M I E N T O

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

D E D I C A T O R I A

Martha
A mis padres y hermanas que son mi mayor inspiración

Jonathan
A mis padres y hermanos que son mi mayor inspiración

Oswaldo
A mi familia sobre todo a mis padres

TRIBUNAL DE GRADUACIÓN

Ing. Carlos Valdivieso A.
Director de Materia de Graduación

Ing. Hugo Villavicencio V.
Delegado del Decano

Ing. Jorge Aragundi
SUBDECANO DE LA FIEC

DECLARACION EXPRESA

“La responsabilidad del contenido expuestos en este Informe de Materia de Graduación nos corresponden exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

Aguirre Martha

Cagua Jonathan

Criollo Ortiz Oswaldo

RESUMEN

La evolución de la robótica en los últimos tiempos ha ido evolucionando para ir a paso con las nuevas tendencias tecnológicas, es así que constantemente aparecen nuevos avances tecnológicos y uno de estos son los denominados Legos Mindstorms NXT.

Estos Legos creados para niños desde los 9 años, permite su desenvolviendo mental, pero estas mismas herramientas son usados por personas con mayor experiencia en robótica para el desarrollo de procesos a nivel industrial para diferentes aplicaciones técnicas.

Es por ello que este proyecto desarrollará un sistema que permita el monitoreo de una determinada área de interés por parte del usuario, además el uso de una interfaz que permita al interesado realizar la adquisición de datos de una variable específica en este caso sensar el ruido presente en un determinado lugar y permitir la visualización de los datos en tiempo real.

Para lograr este fin se usará los conocimientos de Matlab y Simulink para la movilidad del Robot y el uso de GUI para la presentación de datos en una interfaz debidamente elaborada.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

INTRODUCCIÓN

Este proyecto de graduación tiene como objetivo destacar el uso de los sensores de tacto y ultrasónico de los LEGOS MINDSTORMS NXT bajo la programación de Simulink de Matlab con su respectiva aplicación para la adquisición de datos.

En el capítulo 1 se mencionan los objetivos y el alcance del Proyecto. También se realizó un estudio de soluciones similares en el mercado.

En el capítulo 2 se indicarán la base teórica de lo que se utilizó. Se mencionan el hardware y el software para el desarrollo de este proyecto.

En el capítulo 3 se presentará el diseño del proyecto, sus protocolos de comunicación, algoritmos de transmisión y de movilidad del robot.

En el capítulo 4 se mostrará la implementación del código en el robot y el funcionamiento de la interfaz gráfica.

Finalmente se presentan las conclusiones y recomendaciones que hemos obtenidos en la materia de graduación.

CAPITULO 1

1 DESCRIPCIÓN GENERAL DEL SISTEMA.

1.1 Introducción

Este proyecto tiene por objeto el destacar el uso de los sensores de tacto y ultrasónico de los LEGOS MINDSTORMS NXT bajo la programación de Simulink de Matlab, para lo cual se desarrolló una aplicación usando estos sensores y además se hizo uso del sensor de sonido para representar una variable analógica que nos proporcionará los datos necesarios para la adquisición y procesamiento de los mismos.

La aplicación consiste en un sistema de monitoreo de señales en este caso señales analógicas, que puede ser ruido, presión o temperatura, para este estudio se utilizará el ruido como variable a analizar.

Este sistema se ha desarrollado usando una de las innovaciones de la tecnología como lo es la robótica y consiste en un pequeño robot previamente armado dando la apariencia de un explorador, el mismo que se moverá en diferentes sectores y cada vez que el usuario lo requiera poder hacer la adquisición de la señal para su posterior graficación.

1.2 Antecedentes

Esta no es una de las primeras aplicaciones desarrolladas con el NXT, puesto que se han desarrollado automatismos simples hasta proyectos más ambiciosos. Como por ejemplo, una de las aplicaciones más básicas diseñadas por estos kits es el de seguidor de línea.

Figura 1.1 Diseño de un seguidor de línea.

Otra de las aplicaciones usadas con este kit es un robot estable sobre dos ruedas NXTWay-G que usa hábilmente los sensores proporcionados con el NXT.

Figura 1.2 Diseño de un NXTWay-G.

Es decir las aplicaciones que se les da a este tipo de kits son variadas, desde básicas para niños hasta más complejas para personas que han desarrollado interés en este campo de la robótica y les gusta innovar con este tipo de aplicaciones. Incluso cuando se adquiere el kit, en él incluye varios modelos para comenzar a desarrollarse como diseñadores de robots.

Como se ve, el potencial de procesamiento del NXT hace que las posibilidades sean variadas. De hecho, ya se ha realizado un experimento de control con un sistema coche-péndulo cuyos autores fueron Peter Gawthrop, miembro y profesor del Departamento de Ingeniería Mecánica de la Universidad de Glasgow, y Euan McGookin, también profesor y miembro del Departamento de Ingeniería Eléctrica y Electrónica de la misma universidad.

Estos tipos de kits se pueden usar con varios tipos software pero los estudiantes del laboratorio del primer semestre de la RWTH Aachen University, desarrollaron bloques de Matlab para usarlos con legos así como varios códigos para emplearlos en ellos. Este es el motivo por el cual nuestra aplicación fue realizada en este software, por lo que el robot explorador será diseñado en el ambiente de Matlab junto con aplicaciones de Simulink.

En la actualidad existen modelos similares al que se desarrollará en este proyecto como por ejemplo el robot "Spirit" que desde enero del 2004 empezó a enviar fotografías de la superficie de Marte, cuyo objetivo es la recolección y análisis de datos para determinar si en Marte alguna vez hubo agua, este fue nuestra motivación para diseñar robot explorador con capacidad de adquisición de datos.

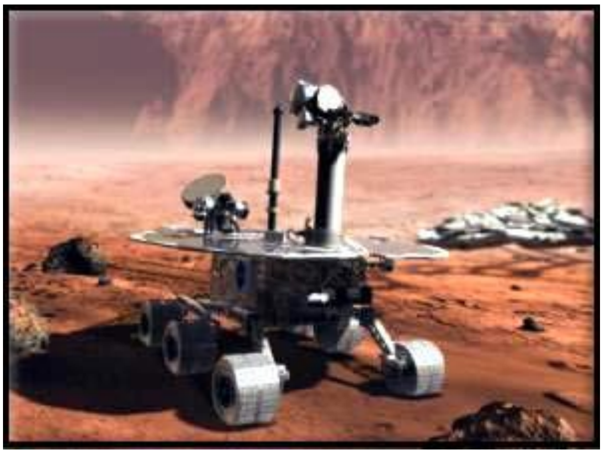


Figura 1.3 Diseño de robot "Spirit"

1.3 Descripción general del Proyecto

La aplicación que se ha desarrollado consiste en un pequeño robot explorador que se lo ha armado usando el Kit LEGO MINDSTORMS NXT, este robot permanecerá explorando en diferentes sectores evadiendo obstáculos con la ayuda de los sensores de tacto y ultrasónico y además realizará la adquisición de datos por parte del sensor de sonido que medirá el nivel de ruido en el ambiente, cabe recalcar que esto es solo una representación de una variable que se desea medir, el usuario cada cierto tiempo según lo requiera podrá realizar la adquisición de estos datos mediante la comunicación que se establecerá entre la computadora y el robot explorador.

Para la programación del robot y visualización de los datos se utilizó Simulink y Matlab con todas las herramientas que estos dos programas nos van a proporcionar, el desarrollo usado se lo estará detallando en capítulos posteriores.

1.4 Análisis y justificación del sistema

Para la realización de este proyecto hay que considerar el hecho de que se partió de un sistema robótico que cuenta con varios sensores y que permite la interacción con datos y variables que puedan ser manipuladas. El NXT cumplió con la arquitectura básica deseada para desde allí poder ampliar o destacar el uso de uno o varios sensores en este trabajo, los sensores utilizados son el ultrasónico y de tacto como principales protagonistas y el sensor de sonido que se lo ha utilizado para desarrollar una aplicación específica.

Aunque este grupo de investigación solo ha utilizado tres de los sensores que vienen en este kit, eso no significa que no se podrían utilizar los otros sensores para futuras innovaciones en otro tipo de aplicaciones.

A continuación se analizará el motivo por el cual se ha elegido la herramienta Simulink de Matlab para la programación de este lego en vez de utilizar su propio software de programación. Simulink de Matlab es una poderosa herramienta en el campo de la ingeniería debido a que cuenta con una variedad de toolbox e incluso hoy en día ya existen el toolbox para trabajar con estos legos, esto permite a los ingenieros incursionar en casi toda las ramas de la ingeniería ya sea para analizar y desarrollar complejos procesos industriales como para analizar de una gran variedad de señales entre otras aplicaciones. La gran fortaleza de este software le permitirá a este grupo de trabajo desarrollar otras posibles aplicaciones utilizando el mismo sistema en otro campo de la ingeniería debido a que con ayuda de este se podría realizar ampliaciones al sistema como otro tipo de sensores y más motores si es necesario, esta es la razón por el cual se ha utilizado este software para la programación de los legos y no su propio software.

1.5 Alcance del proyecto y soluciones similares

Se considera que un proyecto como este podría tener algunas aplicaciones en el campo de la industria debido a que no se lo ha orientado a una aplicación específica sino a un estándar el cual es el monitoreo de parámetros utilizando la innovación de la robótica.

Tan solo cambiar un sensor o hacer pequeñas innovaciones se lo puede aplicar para otro campo de la industria a parte de la que se ha especificado en este proyecto.

Es así que si se reemplaza el sensor de sonido con sensores que mida el nivel de oxígeno en el aire o gases nocivos se puede aplicar este sistema en la industria minera para garantizar la seguridad de un obrero antes y durante la extracción del mineral puesto que se lo puede utilizar para determinar el nivel de oxígeno seguro y si hay algún mineral o gas tóxico en un yacimiento mineral.

Otra posible aplicación sería en la industria pesquera ya que con algunas innovaciones como reemplazar el sensor ultrasónico con un radar y la tecnología vía bluetooth con tecnología wireless y además dándole direccionamiento a este equipo que se puede realizar con un giroscopio, se puede utilizar este sistema para hallar los bancos de peces.

CAPÍTULO 2

2 ANÁLISIS TEÓRICO

2.1 Fundamento Teórico.

Este proyecto está basado en el uso del lego mindstorms para el diseño de un robot explorador, haciendo uso de sensores como lo son el de tacto y el ultrasónico. Además para poder simular el sensor que va a adquirir datos hemos hecho uso del sensor de sonido. Siendo este una representación de uno de los tantos sensores que puede usar el lego. Pues lo que se analizará es una variable del tipo analógica que puede provenir de cualquier dispositivo que puede ser de presión o de temperatura, estos datos serán procesados por el lego y a la vez procesadas por el entorno Matlab junto con Simulink.

En este capítulo se mencionará detalles del lego así como sus sensores resaltando a los sensor de tacto y ultrasónico que son de interés de estudio en este proyecto. Además mencionaremos detalles del software, Matlab y Simulink, dando sus características más relevantes dentro de lo que es procesamiento de señales y adquisición de los mismos.

Es de importancia también mencionar el software que junto con Matlab y Simulink ayudarán a la programación del robot explorador.

2.1.1 Lego Mindstorms NXT

Lego Mindstorms NXT es una herramienta de desarrollo robótico que proporciona un gran número de posibilidades para los entusiastas de la robótica.

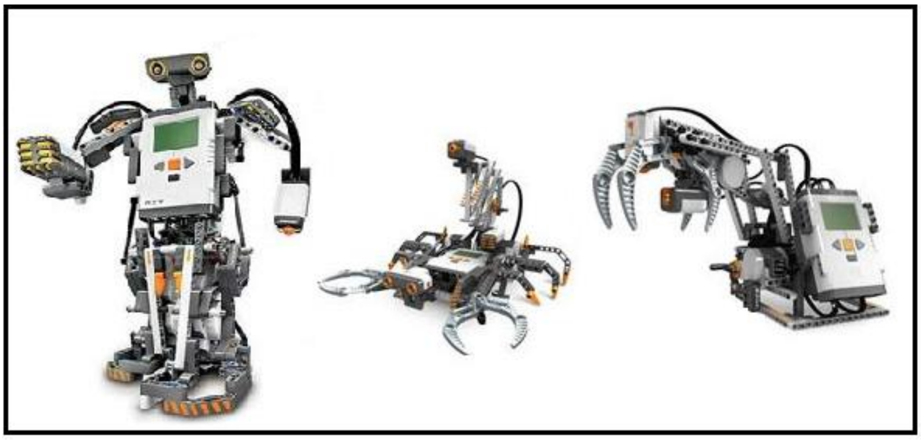


Figura 2.1 Lego Mindstorm.

Lego Mindstorms NXT, está recomendado desde los 10 a los 99 años, pudiendo construir y programar robots que hagan lo que nosotros deseamos. Una nueva opción de "Comienzo Rápido", hace más fácil construir y programar con éxito un robot en sólo 30 minutos, con su sistema de programación por USB o por vía Bluetooth.

Figura 2.2 Comunicación entre lego y computadora.

El software de programación de Lego Mindstorms NXT, tiene como principales características:

- Fácil de programar y usar en PC y MAC gracias a la tecnología de LabVIEW de National Instruments
- Interfaz intuitiva, basada en iconos, que se arrastran y se colocan en la pantalla para realizar programas
- 18 modelos de robot con instrucciones de construcción paso a paso para guiar a los principiantes en el proceso de construcción y programación de robots e inspirar a usuarios más expertos.

2.1.2 Componentes del NXT

Los componentes del lego Mindstorms NXT son:

El ladrillo o cerebro del NXT.

El ladrillo es el elemento fundamental del LEGO Mindstorms NXT. Incluye el CPU, el sistema de comunicaciones, los puertos de y salida y la interfaz de usuario, todo en un bloque de plástico solido de pequeño tamaño y peso.

Figura 2.3 Cerebro del Lego Mindstorms.

- El procesador principal posee un microprocesador Atmel de 32 bits ARM de 256 KB de Flash, 64 KB de RAM y un reloj de 48 MHz.
- El procesador secundario posee un microcontrolador Atmel de 8 bits AVR de 256 Kb de Flash, 64 Kb de RAM y un reloj de 48 MHz.

- Posee mayores capacidades de ejecución de programas, evitando que los procesos inherentes de varios paquetes de datos colisionen y produzcan errores y un posible error en la ejecución del software.
- Comunicación vía Bluetooth, de la clase II V2.0
- Comunicación mediante el puerto USB con una velocidad de 12 Mbit/s.
- Posee 4 puertos de entrada y 3 puertos de salida, mediante cable de 6 líneas.
- Tiene una LCD monocromática con resolución de 100 x 64 pixel.
- Un altavoz de 8 KHz, con una resolución de 8-bit y una resolución de 2-16 KHz de muestreo.
- Posee 4 botones para la interfaz de usuario.
- Alimentación es de 6 pilas AA.

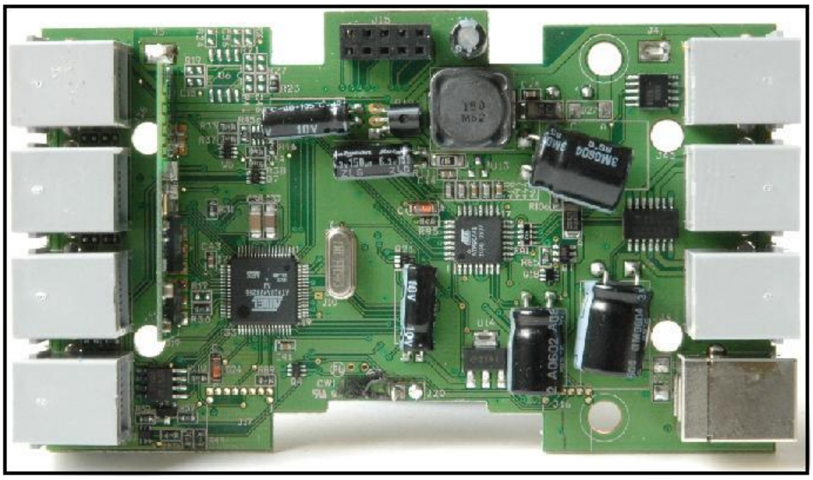


Figura 2.4 Parte interna del Cerebro del Lego.

Sensores

- Servo Motores Interactivos, los cuales permiten la detección de giros de la rueda, indicando los giros completos o medios giros, que es controlado por el software
- Sensor de giro incorporado para unos movimientos precisos.
- Sensor Ultrasónico: Son los "ojos" del robot, y miden la distancia, los movimientos y detectan los objetos.
- Sensor de Sonido: Son los "oídos" del robot y permiten a los robots reaccionar antes comando de sonido y tonos.
- Sensor de Tacto: Son los "dedos" del robot, reaccionado ante la presión o relajación de este sensor.
- Sensor Óptico: Detecta los diferentes colores y la intensidad de luz

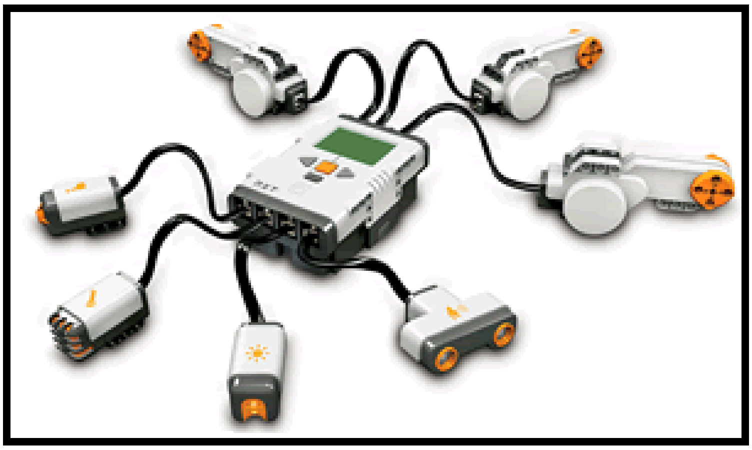


Figura 2.5 Sensores y Motores del Lego Mindstorms.

Consta de 519 piezas de gran versatilidad, para una fácil y más estable construcción, manteniendo las capacidades de funcionalidad y movimiento.

Comunicación del NXT

El bloque de NXT puede comunicarse con el computador mediante la interfaz de USB que posee, la cual ya viene en la versión 2.0. Además, para comunicarse con otros robots en las cercanías posee una interfaz Bluetooth que es compatible con la Clase II v 2.0. Esta conectividad con Bluetooth no tan sólo permite conectarse con otros bloques, sino también con computadores, palms, teléfonos móviles, y otros aparatos con esta interfaz de comunicación.

Dentro de las posibilidades de conexión se encuentran:

- Conectar hasta tres dispositivos distintos,
- Buscar y conectarse a otros dispositivos que posean Bluetooth.
- Recordar dispositivos con los cuales se ha conectado anteriormente para conectarse más rápidamente.

- Establecer el bloque NXT como visible o invisible para el resto de los dispositivos.

2.1.3 Lenguajes de programación

La programación del Lego Mindstorms se realiza mediante el software que se adjunta en el empaque original, el cual trae el firmware del robot y un programa que emula un árbol de decisiones, para los cuales, el usuario debe programar las acciones a seguir por el robot. El software se encuentra dividido por cada tipo de robot que se puede construir, y que viene recomendado en el empaque.

Una de las principales características de este software de programación, es su entorno visual, el cual emula la construcción por bloques, dando la posibilidad a cualquier usuario aprendiz acostumbrarse rápidamente a la programación de bloque.

Este lenguaje permite las instrucciones secuenciales, instrucciones de ciclos e instrucciones de decisiones, éstas últimas, basadas en los datos reportados por los sensores que se puede añadir al robot.

Este lenguaje de programación no será el que se usará para el desarrollo del proyecto, en vez de este software se empleará Matlab junto con Simulink, con forme se haga el desarrollo del proyecto se irá mencionando las características de Matlab y el empleo de éste en los lego Mindstorms.

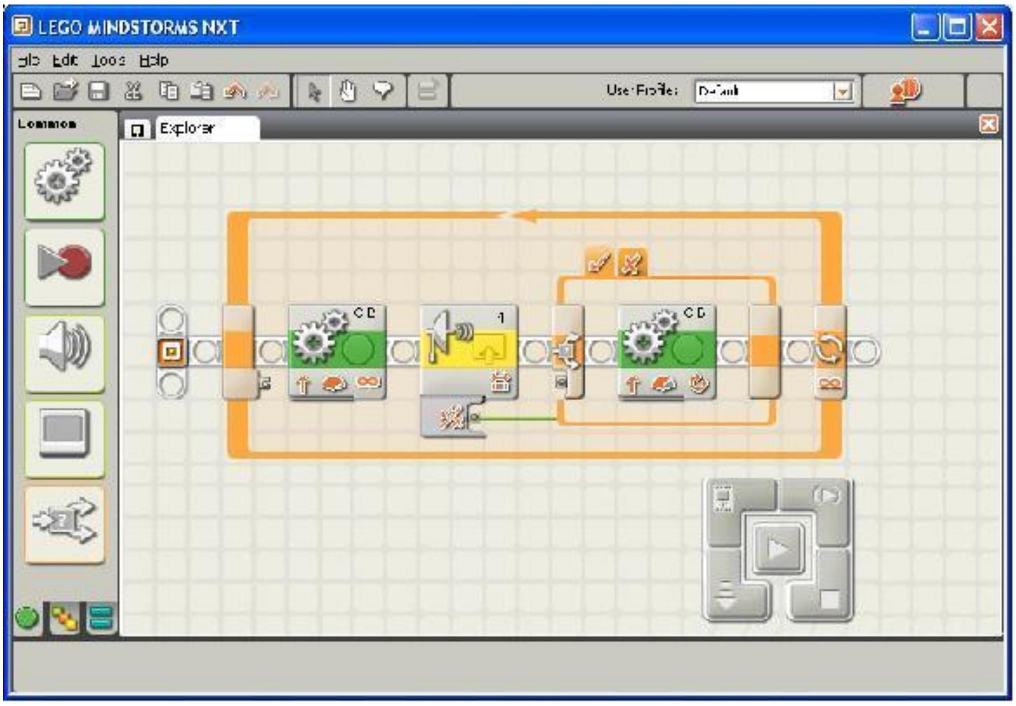


Figura 2.6 Software del Lego Mindstorms.

El bloque del Lego Mindstorms como un producto de hardware y software integrado, puede ser programado con varias interfaces, pero todos logrando el mismo fin. Existen varios lenguajes para programar los NXT, como lo son:

- NXC: Lenguaje parecido a C.
- LeJOS NXJ: Lenguaje para java.

- NXT_Python: Es un módulo de Python.
- LabVIEW:Se pueden desarrollar nuevos bloques.
- MATLAB y Simulink: Para controlar el robot via bluetooth.
- NBC: Lenguaje ensamblador.

Como se menciona el lego Mindstorms puede trabajar bajo cualquiera del software antes detallados, el programa que se empleará en el desarrollo de este proyecto será Matlab junto con Simulink y conforme se vaya desarrollando el proyecto se irá mencionando las características de Matlab y el empleo de éste en los legos Mindstorms.

2.1.4 Requisitos mínimos del Sistema

Los requerimientos para hacer uso de este kit de desarrollo en el sistema operativo Windows son:

- Procesador Intel Pentium o compatible 800 MHz mínimo.
- Windows XP o Home Edition con Service Pack 2
- 256 MB de RAM mínimo.
- 300 MB de espacio de disco duro.
- XGA display (1024x768)
- 1 USB port.

- CD-ROM drive
- Compatible Bluetooth.

2.2 Herramientas de desarrollo

Entre las herramientas de desarrollo que se utilizó en el proyecto se encuentra los siguientes:

Software:

- Matlab 2008ª
  - Simulink
  - GUI (Interfaz gráfica de Matlab)
- ECRobot

Hardware

- Lego Mindstorms NXT

2.2.1 Matlab 2008

El nombre MATLAB proviene de Matrix Laboratory. MATLAB es un lenguaje de alta performance para cálculo técnico. El mismo integra cálculo, visualización y programación en un entorno de fácil utilización en donde los problemas y las soluciones son expresados en una notación matemática familiar.

Los usos más típicos incluyen:

- Cálculos Matemáticos.
- Desarrollo de Algoritmos.
- Modelado, simulación y prototipos.
- Gráficas Científicas e Ingenieriles.

Pero MATLAB llego mucho más allá, hasta llegar a ser un lenguaje de computación técnica de alto nivel, entorno de desarrollo para la gestión de código, archivos y datos, con herramientas interactivas para desarrollo de algoritmos, visualización de datos, análisis de datos como los que aplicaremos en nuestra tesis.

MATLAB tiene una amplia gama de aplicaciones que incluyen procesamiento de señales e imágenes, comunicaciones, diseño de sistemas de control, sistemas de prueba y medición. La aplicación que le daremos en este proyecto será para la programación y adquisición de datos a través del Lego Mindstorms NXT.

MATLAB posee un amplio rango de herramientas para modelar sistemas de control, análisis, simulación y procesamiento de prototipos, MATLAB es el sistema ideal para desarrollar sistemas avanzados de control.

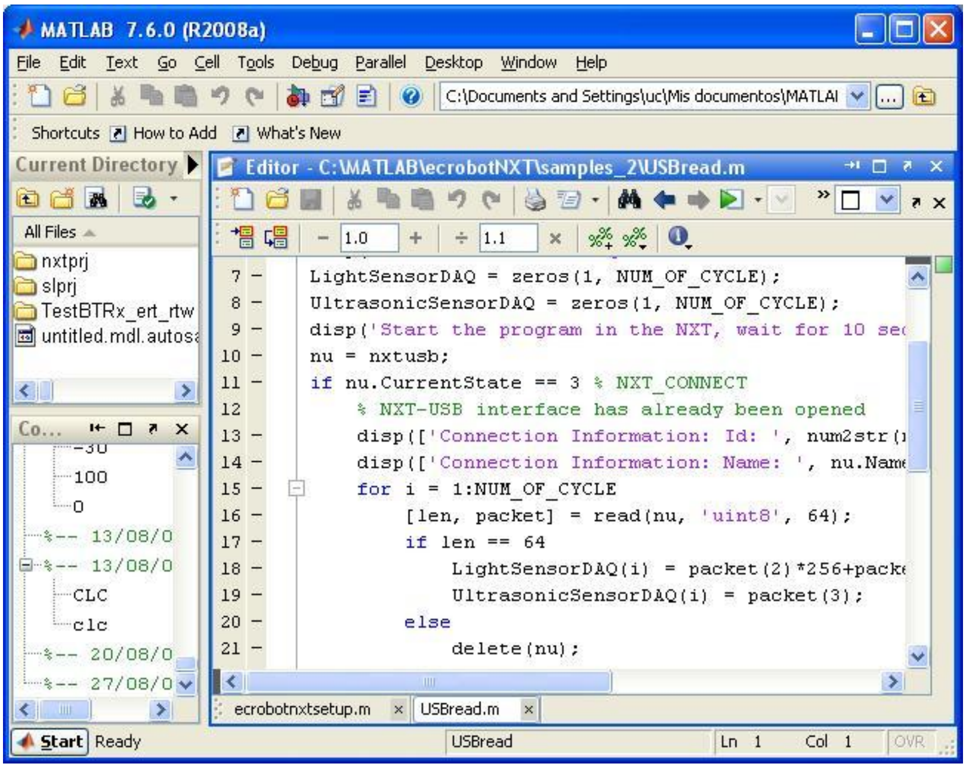


Figura 2.7 Interface del Programa de Matlab.

MATLAB dispone en la actualidad de un amplio abanico de programas de apoyo especializados, denominados Toolboxes, que extienden significativamente el número de funciones incorporadas en el programa principal. Estos Toolboxes cubren en la actualidad prácticamente casi todas las áreas principales en el mundo de la ingeniería y la simulación, destacando entre ellos el 'toolbox' de proceso de imágenes, señal, control robusto, estadística, análisis financiero, matemáticas simbólicas, redes neuronales, lógica difusa, identificación de sistemas, simulación de sistemas dinámicos, etc. es un entorno de cálculo técnico, que se ha convertido en estándar de la industria, con capacidades no superadas en computación y visualización numérica.

Para poder generar el código ANSI-C y poder enviar al LEGO MINDSTORM NXT el programa realizado y que lo ejecute según este diseñado, se utilizó el Real Time Workshop, propio de MATLAB que permitirá generar el código C autónomo para desarrollar y probar algoritmos modelados en Simulink.

Es precisamente Real-Time Workshop el que facilita la conversión del programa desde Simulink a un archivo de ejecución que luego se carga en el Lego mediante comunicación USB, para luego poder ejecutarlo según lo que se haya diseñado en Simulink.

MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario - GUI).

2.2.2 Simulink

Simulink es una librería de Matlab que proporciona al usuario un ambiente interactivo para modelar una amplia variedad de sistemas dinámicos, pudiendo ser estos lineales, no lineales, discretos, de tiempo continuo y sistemas mixtos. Permite realizar diagramas de bloques con operaciones click-and-drag, cambiar parámetros del modelo y visualizar resultados durante una simulación.

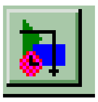
MATLAB y su grupo de Toolboxes ofrecen un conjunto amplio de herramientas de ingeniería y matemática para definir algoritmos, analizando datos y visualizando resultados. Juntos, SIMULINK y MATLAB proveen un entorno integrado para construir modelos versátiles y simular dinámicos, diseñando y testeando ideas nuevas.

La simulación se puede realizar desde menús descolgables o desde la línea de comandos de Matlab. Los resultados pueden ser vistos durante la simulación usando osciloscopios (Scopes) o bloques gráficos (Graph blocks); o grabarlos en un archivo o transferidos al espacio de trabajo de Matlab para su posterior análisis o procesamiento.

Existe un Toolbox especial, diseñado para manipular todos los sensores del LEGO MINDSTORM NXT, su modo de uso, características se lo estará describiendo en capítulos posteriores.

Simulink es la plataforma que se utilizó para la realización de la programación del sistema del robot explorador, haciendo uso de su Toolbox específico así como el resto de paquetes de bloques para realizar operaciones conjuntas entre sensores.

Para la utilización de Simulink es necesario familiarizarse con la manipulación de bloques y la construcción de modelos. Para esto es necesario también conocer la variedad de bloques existentes. Y por último manejar las herramientas de análisis provistas por Simulink.

Para ejecutar Simulink se deberá tipear Simulink a continuación del símbolo del sistema en Matlab o bien hacer click sobre el icono  que se encuentra en la barra de herramientas de la ventana principal de Matlab. Una vez que se cargue el programa aparecerá una ventana como la siguiente:

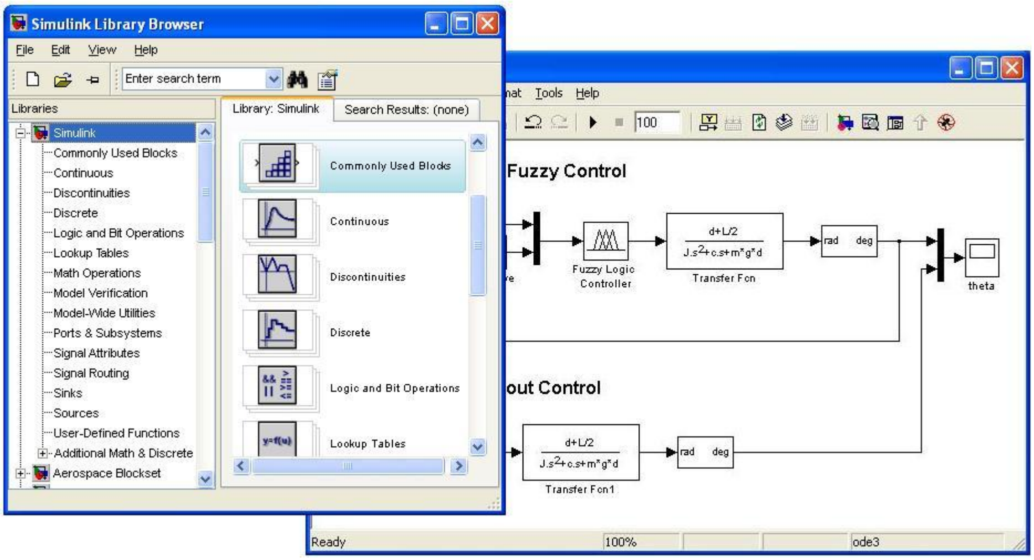


Figura 2.8 Interfaz del Programa Simulink de Matlab.

ECRobot

Embedded Coder Robot NXT es un entorno de desarrollo basado en Matlab y Simulink para lego Mindstorms NXT. ECROBOT ofrece la programación en bloques en Simulink para el control del NXT, también hace uso del Real-Time Workshop propio de Matlab para generar el código ANSI-C y luego hacer la implementación usando el NXTOSEK con el hardware real NXT.

Una vez que un programa de aplicación nxtOSEK se ejecuta desde la RAM del NXT, sobrescribe los datos del firmware del NXT estándar, por lo que no es posible volver al firmware de NXT cuando termina el programa de aplicación nxtOSEK. Esta es una restricción de la ejecución del NXTOSEK

Embedded Coder Robot NXT utiliza software anexos para configurar el entorno de desarrollo. La mayoría de ellos (excepto los productos MATLAB) están disponibles de forma gratuita (código abierto). Como condición previa, MATLAB se debe instalar en el directorio que no tiene espacios. El software adicional que se deben instalar de manera correcta para el normal funcionamiento del sistema:

- Cygwin
- GNU ARM
- NeXTTool
- LibUsb driver

Una vez que se instala adecuadamente todo el software adicional, en Simulink se agrega un nuevo blockset, “ECRobot NXT Blockset”, que es propio para aplicaciones de lego mindstorms NXT que serán los usados en este proyecto.

Este nuevo blockset permite obtener bloques para realizar la programación que se necesita para los NXT, la manera como se lo realiza el diseño del sistema con estos nuevos bloques es de igual forma como si se estuviera usando un blockset cualquiera de SIMULINK, por el método de click-and-drag.

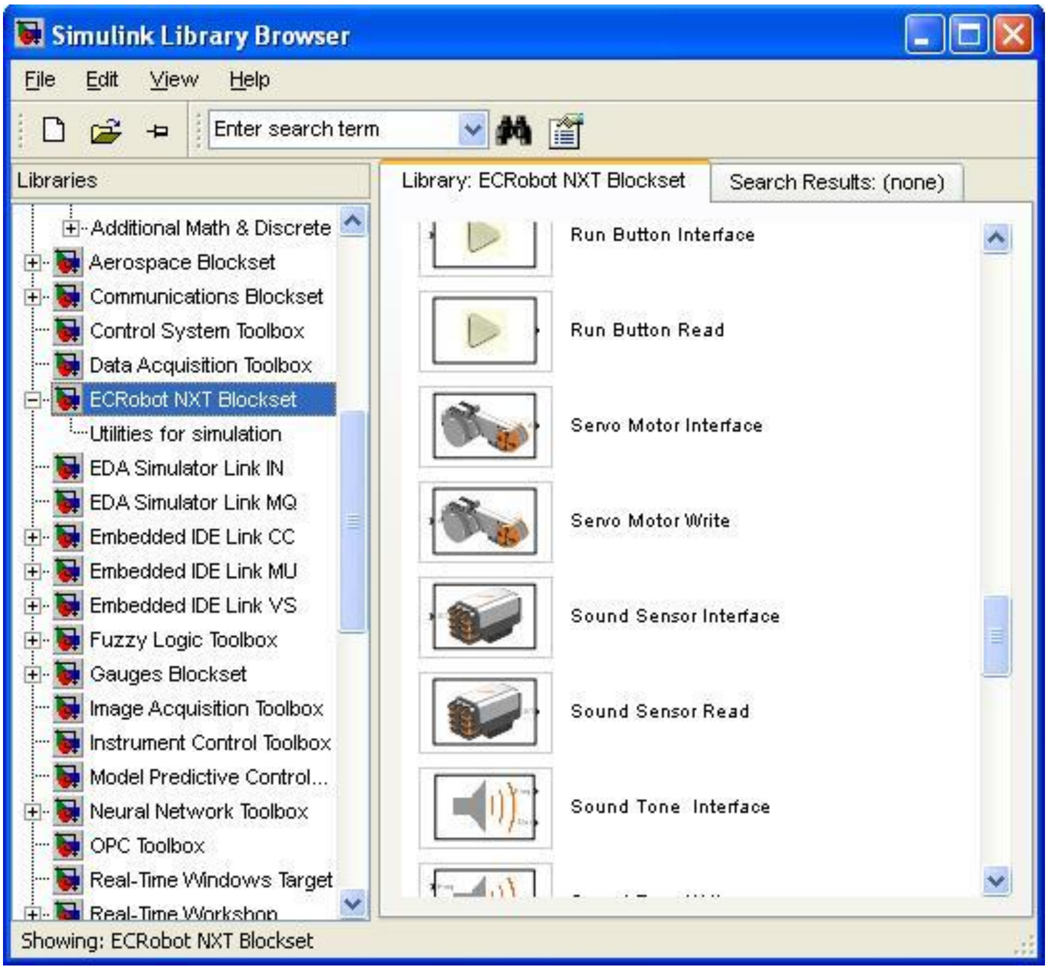


Figura 2.9 Paquete de ECRobot con bloques para Lego NXT.

2.3 Componentes del sistema.

El diseño del sistema de este proyecto está basado en el uso de los lego Mindstorms para obtener un robot explorador, pero para ello detallaremos características importantes de componentes que intervienen en el sistema.

Uno de los componentes básicos del diseño lo conforma el lego NXT, y su parte central lo es el “Ladrillo Programable” de LEGO, conocido como Brick, que no es más que una microcomputadora programable que puede enviar señales a los puertos de salida y recibir información de sensores conectados a sus puertos de entrada. El NXT realiza las funciones de una interfaz recolectora de datos, que transfiere al computador la información proveniente de los Sensores.

Los sensores que utiliza el kit el Lego Mindstorms NXT son:

- Servos motores.
- Sensor Óptico.
- Sensor ultrasónico.
- Sensor de tacto
- Sensor de sonido.

Para el desarrollo de este proyecto se hará uso de servos motores y como sensores de estudio será el de tacto y ultrasónico, además como sensor que representa el ingreso de un valor analógico para poder realizar la adquisición de datos del mismo, haremos uso del sensor de sonido. A continuación un breve detalle de los sensores que se usarán en el proyecto.

2.3.1 Servos Motores.

Los servos motores que suministran el NXT son tres motores de corriente continua y funcionamiento PWM. El uso de un motor paso a paso pudiera parecer en principio más interesante, pero los del NXT son tradicionales de corriente continua. Aún más, y a pesar de su reducido tamaño, esconden en su interior un complejo sistema de reducción por tren de engranajes y un sensor de rotación de tipo tacométrico.

El servo motor alcanza sin carga una velocidad de 160 rpm, y hasta 120 rpm con una carga de 11.5 N.cm, alimentado a 9 V. Es capaz de entregar un par máximo de 25 N.cm a 60 rpm, aunque no es recomendable exceder de los 15 N.cm en periodos prolongados (puede producir fatiga al actuador).

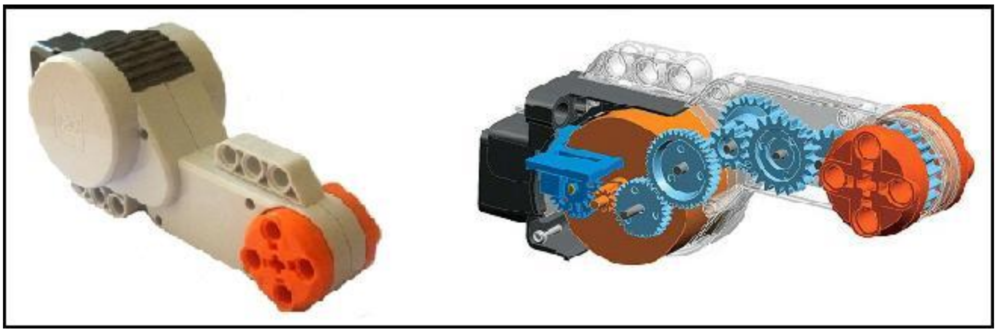


Figura 2.10 Servo Motor con imagen de su parte interna.

2.3.2 Sensor Ultrasónico

El sensor Ultrasónico al contrario del resto de sensores, no devuelve los valores en ninguna escala ni porcentaje, sino en unidades reales, bien en centímetros o en pulgadas.

Su principal función es detectar las distancias y el movimiento de un objeto que se interponga en el camino del robot, mediante el principio de la detección ultrasónica. Este sensor es capaz de detectar objetos que se encuentren desde 0 a 255 cm, con una precisión relativa de +/- 3 cm.

Mediante el principio del eco, por ultrasonido a 40 KHz, el sensor es capaz de recibir la información de los distintos objetos que se encuentren en el campo de detección. El sensor funciona mejor cuando las señales ultrasónicas que recibe, provienen de objetos que sean grandes, planos o de superficies duras. Los objetos pequeños, curvos o suaves, como pelotas, pueden ser muy difíciles de detectar. Si en el cuarto se encuentra más de un sensor ultrasónico, los dispositivos pueden interferir entre ellos, resultando en detecciones pobres.

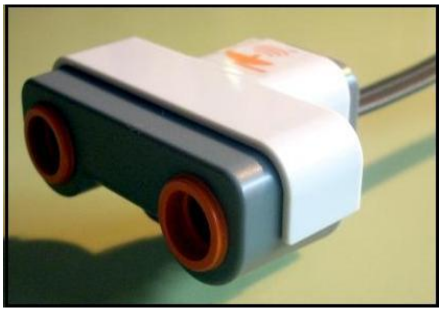


Figura 2.11 Sensor Ultrasónico.

2.3.3 Sensor de Tacto

El sensor de tacto permite detectar si el bloque que lo posee ha colisionado o no con algún objeto que se encuentre en su trayectoria inmediata. Al tocar una superficie, una pequeña cabeza externa se contrae, permitiendo que una pieza dentro del bloque cierre un circuito eléctrico comience a circular energía, provocando una variación de energía de 0 a 5 V.

Por lo tanto el sensor de presión simplemente devuelve una señal digital binaria cuando se activa y desactiva. Normalmente se usa en conjunto con más piezas de LEGO para aumentar el alcance táctil del robot.

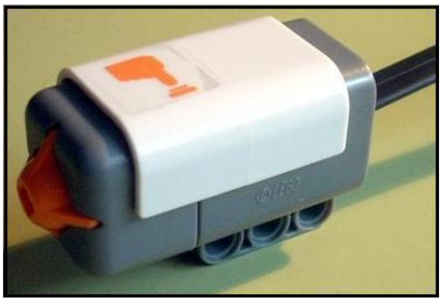


Figura 2.12 Sensor de Tacto.

2.3.4 Sensor de Sonido

El sensor acústico puede configurarse para devolver los valores de medida en decibelios (dB) o decibelios ajustados (dBA), estando esta última unidad de medida ponderada a los niveles audibles por el oído humano.

La sensibilidad máxima se encuentra en los 90 dB (aproximadamente el mismo nivel sonoro que una calle ruidosa con mucho tráfico). La respuesta a intensidades en decibelios crecientes es aproximadamente exponencial.

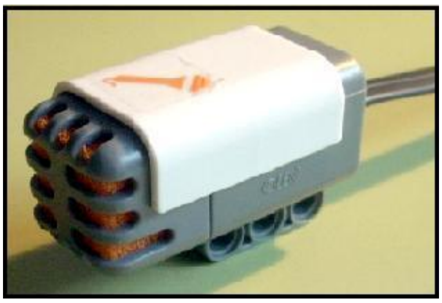


Figura 2.13 Sensor de Sonido.

2.4 Fundamentos de la tecnología aplicada

Para este proyecto vamos a establecer una comunicación con el puerto COM de la computadora establecido por el software del Bluetooth y lego mindstorms NXT, de esta manera nosotros podremos hacer una adquisición de datos. Esos detalles serán de interés en el próximo capítulo.

A continuación características relevantes de esta tecnología.

Bluetooth

Bluetooth, es una especificación industrial para Redes Inalámbricas de Área Personal que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz.

Los principales objetivos que se pretenden conseguir con esta norma son:

- Facilitar las comunicaciones entre equipos móviles y fijos.
- Eliminar cables y conectores entre éstos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales.

Los dispositivos que con mayor frecuencia utilizan esta tecnología pertenecen a sectores de las telecomunicaciones y la informática personal, como PDA, teléfonos móviles, computadoras portátiles, ordenadores personales, impresoras o cámaras digitales.

Gracias a este protocolo, los dispositivos que lo implementan pueden comunicarse entre ellos cuando se encuentran dentro de su alcance. Las comunicaciones se realizan por radiofrecuencia de forma que los dispositivos no tienen que estar alineados y pueden incluso estar en habitaciones separadas si la potencia de transmisión lo permite. Estos dispositivos se clasifican como "Clase 1", "Clase 2" o "Clase 3" en referencia a su potencia de trasmisión, siendo totalmente compatibles los dispositivos de una clase con las de las otras.

Figura 2.14 Comunicación entre el brick y computador.

CAPÍTULO 3

3 DISEÑO E IMPLEMENTACIÓN DEL PROYECTO.

3.1 Detalles del diseño propuesto.

Este capítulo muestra el diseño y la implementación del software y hardware para alcanzar cada uno de los objetivos propuesto. En primera instancia se describirá de una manera muy general el diseño del hardware utilizado para luego mostrar la parte medular del proyecto que es la implementación del software.

La parte del hardware está compuesta por un computador, el cual dispone de un bluetooth dongle por el cual se realizará la adquisición de datos que ha sido debidamente procesada y enviada por medio del robot explorador.

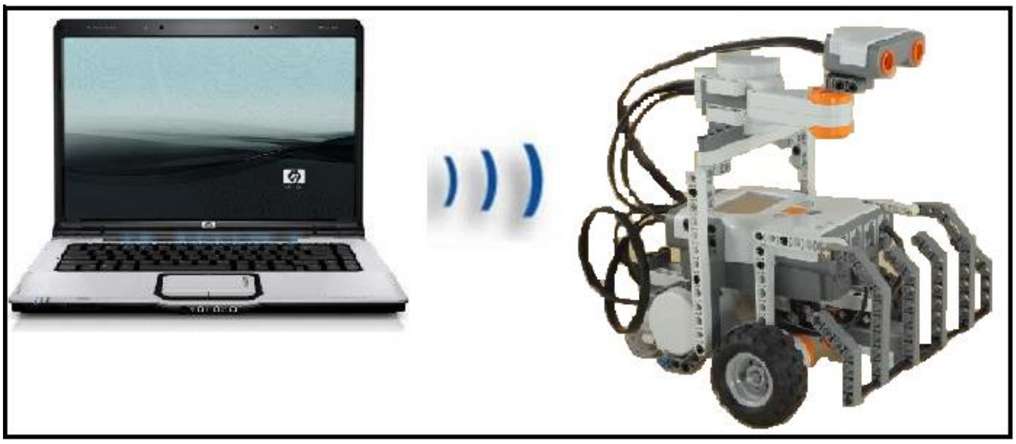


Figura 3.1 Comunicación Bluetooth entre el lego y computadora.

El diseño del robot explorador lego fué basado en una implementación encontrada en la siguiente página de internet: http://www.nxtprograms.com/explorer/index.html

Los pasos a seguir para armar el explorador se encontrarán en esta página, así como detalles del programa desarrollado en el propio software de lego Mindstorms, la diferencia radica en que al diseño se le agregó las modificaciones adecuadas para colocar el sensor de sonido, además el software usado para programar la funcionalidad del mismo fue realizado en Simulink de Matlab así como el programa para lograr el procesamiento de los datos adquiridos por parte del sensor.

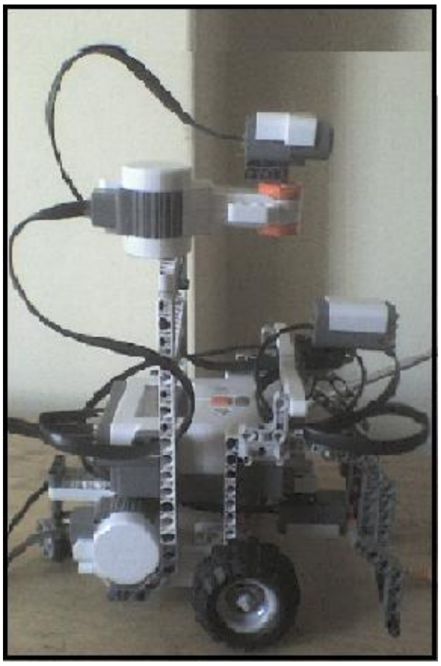


Figura 3.2 Diseño del robot explorador.

Como se ha mencionado anteriormente el Lego Mindstorms tiene la característica de comunicarse por medio de Bluetooth, está característica será utilizada para lograr la adquisición de datos enviados desde el sensor de sonido que tiene el robot al bluetooth dongle que tiene la computadora previamente instalado correctamente con su software respectivo.

La parte del software es la parte principal del proyecto pues es la que permite realizar los programas respectivos para la movilidad del explorador, para la adquisición de datos y además la interfaz que se realizará para el usuario.

El software que se usará para lograr todo estos fines es Matlab, el cual fue detallado en el capítulo anterior mostrando sus características y requerimientos necesarios el cual será de utilidad para este proyecto.

Matlab es una buena herramienta para el manejo de datos en tiempo real y manipulación de los mismos, y son estas las características las que nos permitirán el desarrollo de este proyecto, además vale resaltar que el programa del funcionamiento del robot explorador fue diseñado en Simulink que es una herramienta propia de Matlab.

Para poder hacer uso del lego mindstoms en Simulink se instaló todo el software necesario para este fin, de esa manera poder tener los bloques para lograr la programación.

ECRobot es un toolbox para Matlab, el que nos permite hacer uso de los bloques creados exclusivamente para el manejo de los Lego Mindstorms, estos bloques son los utilizados para el diseño del programa en Simulink que da la funcionalidad para el robot explorador.

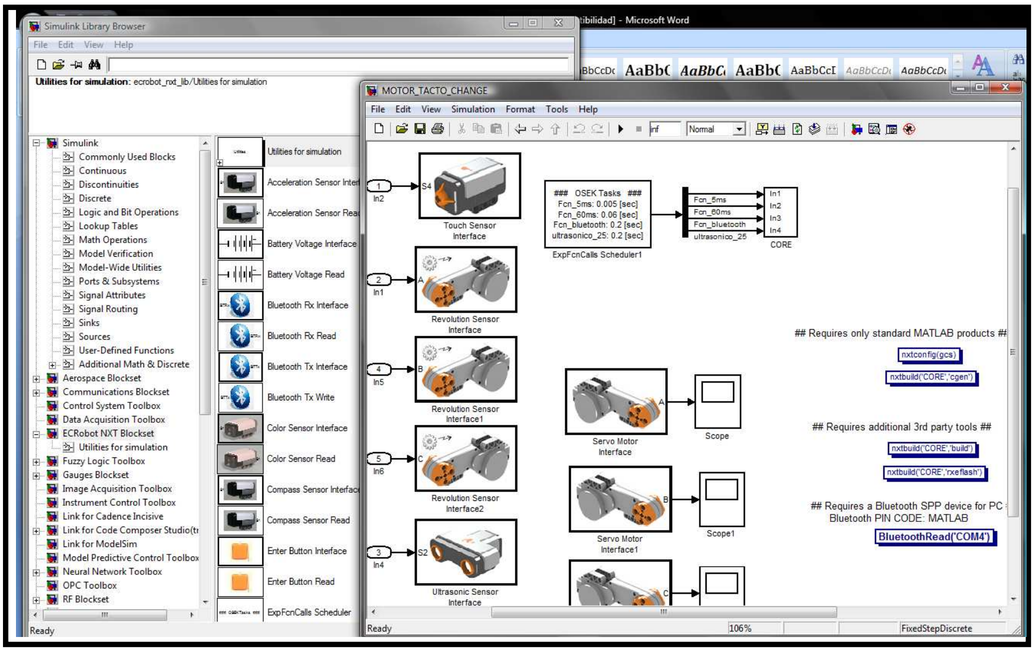


Figura 3.3 Bloques que proporciona ECRobot.

3.2 Diseño del Sistema

El diseño del sistema está basado en el siguiente diagrama de bloque, en el que se muestra las señales de entrada del sistema así como las señales de salida.

Las señales de entrada están representados por los sensores de tacto y ultrasónico, además del sensor que simulará la señal analógica que se procesará, esta señal será la que proviene del sensor de sonido.

Las señales de salida serán las que se encuentra en los motores A, B y C, que permitirán la movilidad del robot explorador, además para lograr la adquisición de los datos se usará el dispositivo bluetooth del lego para enviar los datos a la interfaz gráfica diseñada para el usuario.

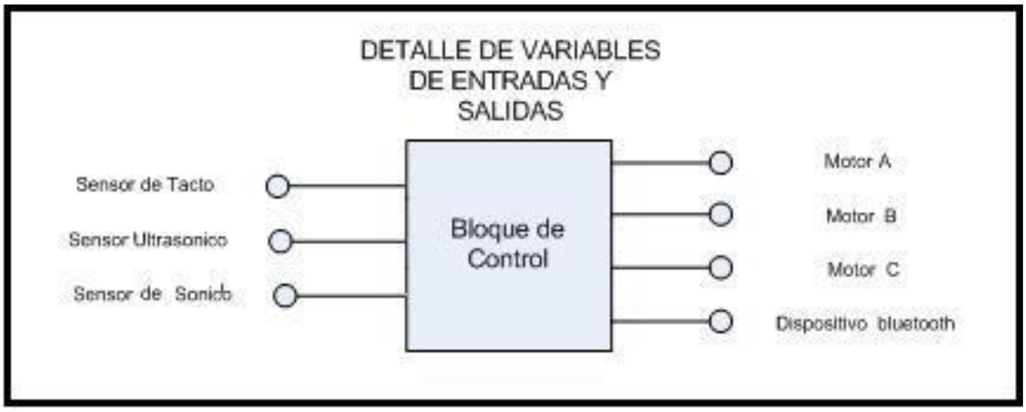


Figura 3.4 Bloque del Sistema propuesto.

3.3 Análisis del programa en Simulink.

Para el diseño del programa que permitirá la movilidad del robot explorador se utilizó los bloques de Simulink cargados para el lego para realizar el respectivo procesamiento de los datos adquiridos por cada sensor.

Para este análisis lo hemos divido en dos partes, en la primera se presentará el diseño de la movilidad del robot explorador explicado en detalle y la segunda parte demostrará la forma de programación gráfica empleado para la adquisición de datos.

3.3.1 Programa para la movilidad del robot explorador

A continuación se describe en detalle el código que genera el movimiento del explorador en Simulink.

Esta es toda la programación desarrollada en Simulink, aquí se detalla todos los boques del lego que proporciona Simulink, así como los bloques que representan el sensor ultrasónico, sensor de tacto, los motores y el dispositivo bluetooth.

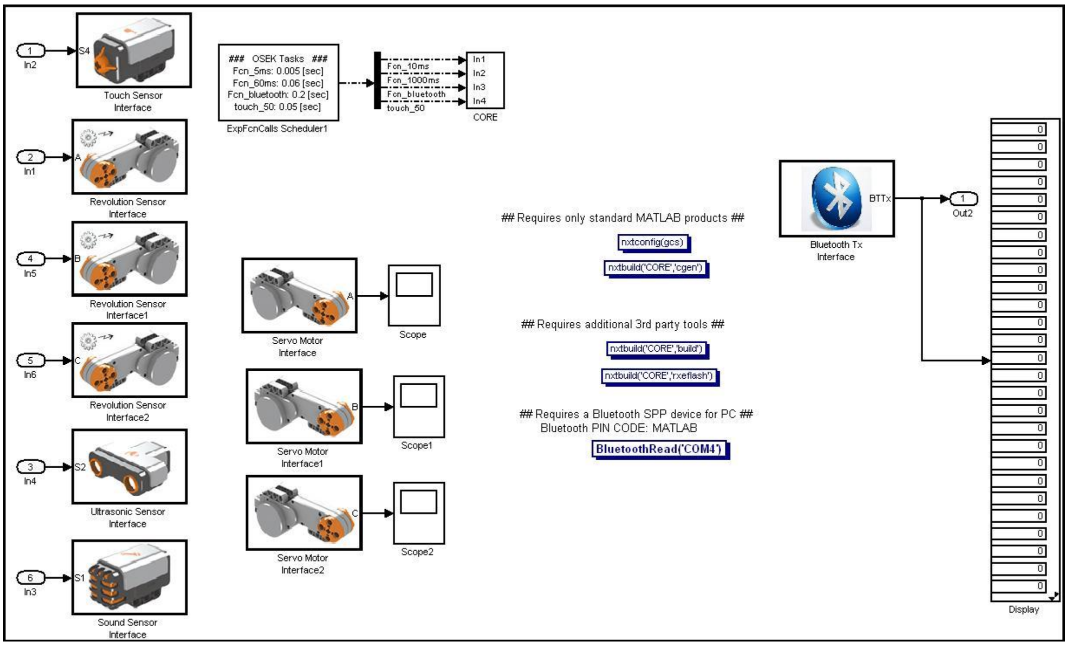


Figura 3.5 Programación gráfica del Sistema en Simulink.

A continuación una descripción de los bloques empleados en el diseño:

El bloque **ExpfcnCalls Scheduler1** sirve para declarar cada cuanto tiempo efectúa la funciones dentro del bloque CORE.

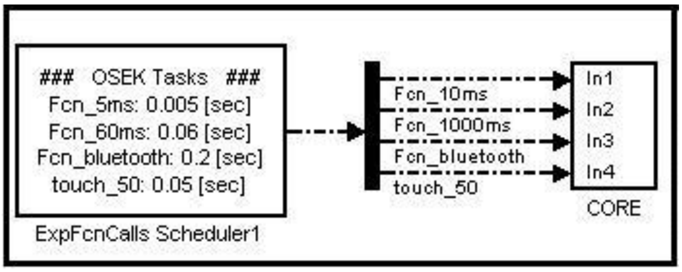


Figura 3.6 Bloque ExpfcnCalls Scheduler1.

Estos tres bloques son los que permite el movimiento del explorador.

Dentro del bloque CORE se encontrarán nuevos bloques que servirán para la programación del robot explorador.

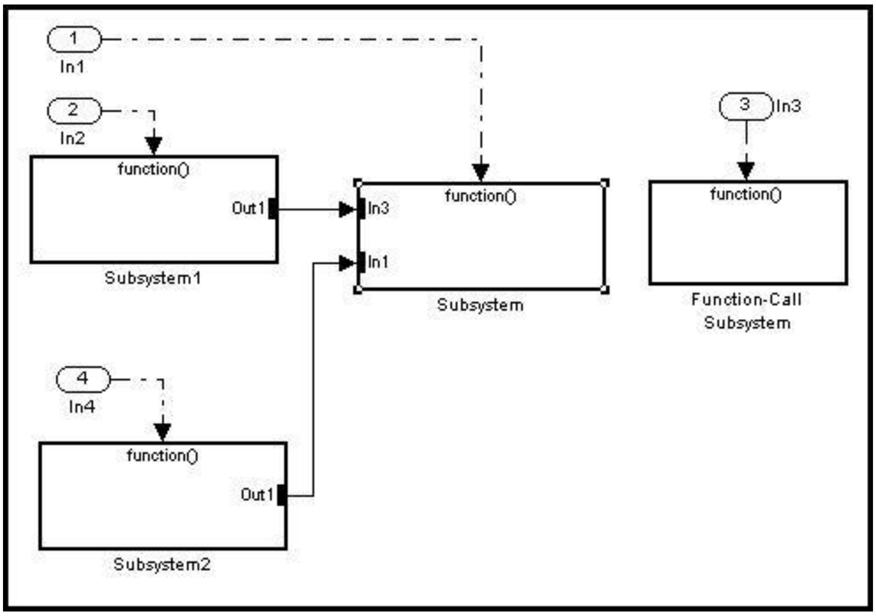


Figura 3.7 Bloques internos del Bloque CORE.

El bloque **Subsystem1** es el bloque encargado de generar una variable externa al proceso, y evitar que ciertos bloques de la función Subsystem queden seteados, lo cual no permite que el proceso se repita de forma correcta siempre.

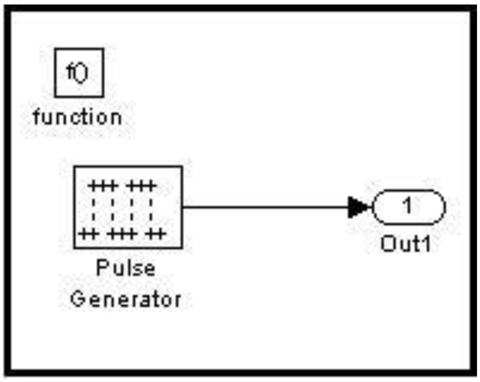


Figura 3.8 Bloques internos del Bloque Subsystem1.

En el bloque **Subsystem2** se efectúa la lectura del sensor ultrasónico.

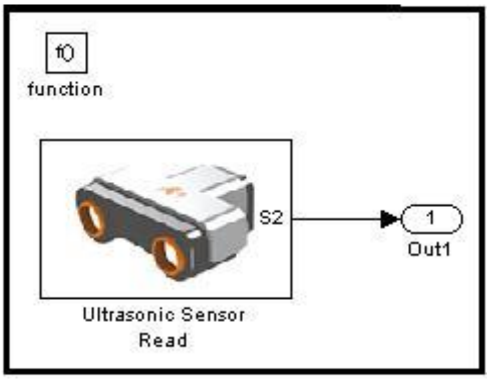


Figura 3.9 Bloques internos del Bloque Subsystem2.

El bloque **Subsystem** es el corazón del movimiento del explorador debido que aquí se encuentran las funciones que permiten controlar y direccionar al explorador para que escoja la ruta más adecuada

A continuación se procede a detallar los bloques internos del bloque Subsystem.

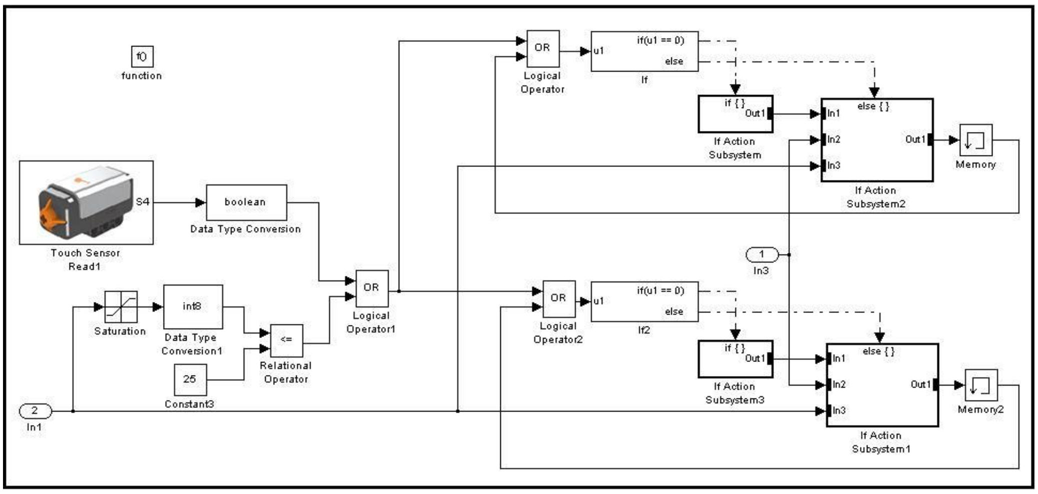


Figura 3.10 Bloques internos del Bloque Subsystem.

En el bloque **if Action Subsystem** es el que se ejecuta cuando el explorador no detecta obstáculo con el sensor ultrasónico y de tacto. If Action Subsystem y if Action Subsystem3 son bloques similares con la única diferencia que If Action Subsystem actúa sobre el Motor A y If Action Subsystem3 sobre el Motor B, entonces solo se detalla if Action Subsystem, a continuación se examina if Action Subsystem.

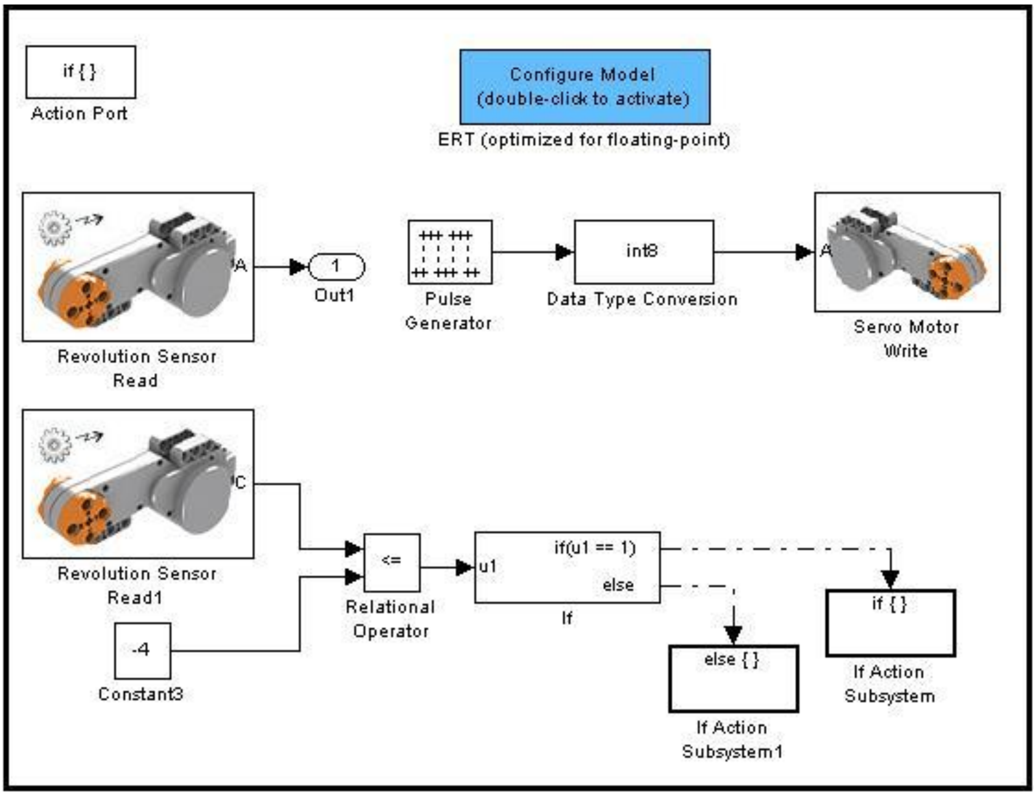


Figura 3.11 Bloques internos del Bloque if Action Subsystem.

La siguiente bloque se utilizo para generar un pulso PWM para el Motor A, el bloque ERT fue necesario debido a que sin este, el software no podía compilar.

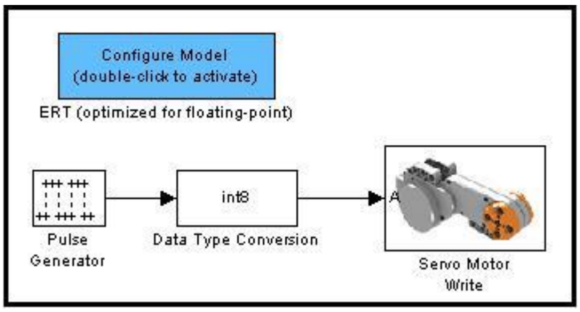


Figura 3.12 Bloques internos del Bloque if Action Subsystem.

Se realizo una lectura del Motor A la cual fue almacenada en una variable de salida de este bloque para ser utilizada posteriormente.

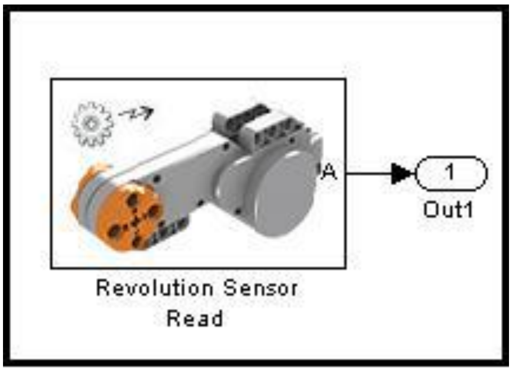


Figura 3.13 Bloques internos del Bloque if Action Subsystem.

Esta función es la que se utilizó para dejar al sensor ultrasónico en su posición inicial.

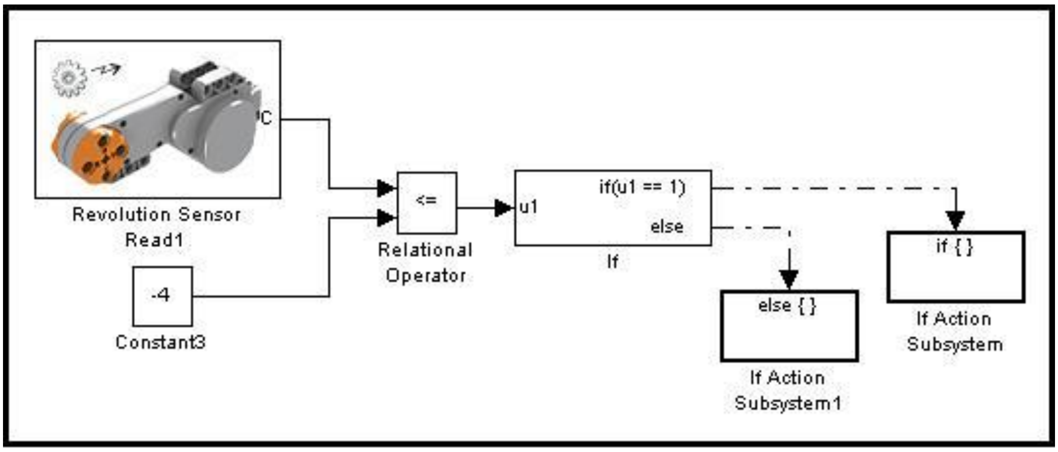


Figura 3.14 Bloques internos del Bloque if Action Subsystem.

El bloque **If Action Subsystem** genera el PWM para el Motor C.

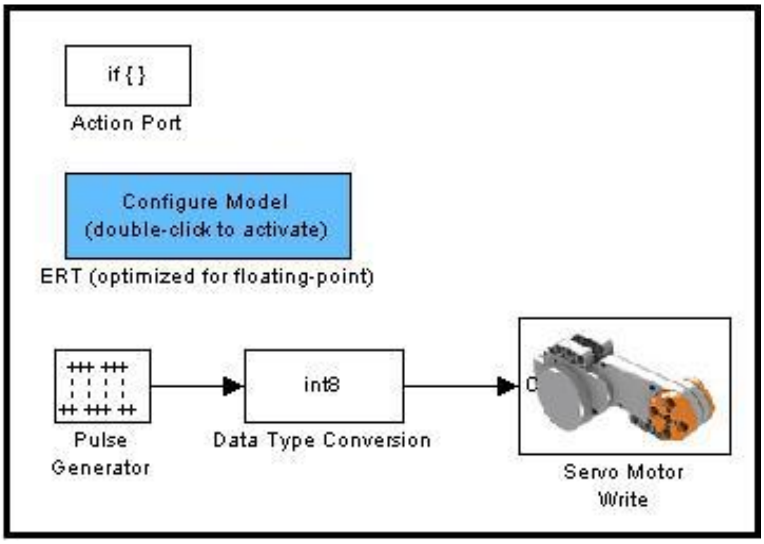


Figura 3.15 Bloques internos del Bloque if Action Subsystem.

El bloque **If Action Subsystem1** se utilizo para detener el Motor C y lograr mayor precisión.

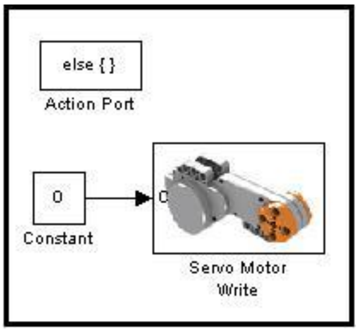


Figura 3.16 Bloques internos del Bloque if Action Subsystem1.

Ahora se analizará los otros dos bloques del corazón del movimiento del explorador **If Action Subsystem2** y **If Action Subsystem1**, al igual que antes estos dos bloques son muy parecidos con la única diferencia que If Action Subsystem2 actúa sobre el Motor A y If Action Subsystem2 sobre el Motor B

Otro punto importante fue la necesidad de agregar un bloque llamado Memory debido a que sin este no se podía realizar una retroalimentación en la aplicación puesto que surgía un error en la compilación de la aplicación.

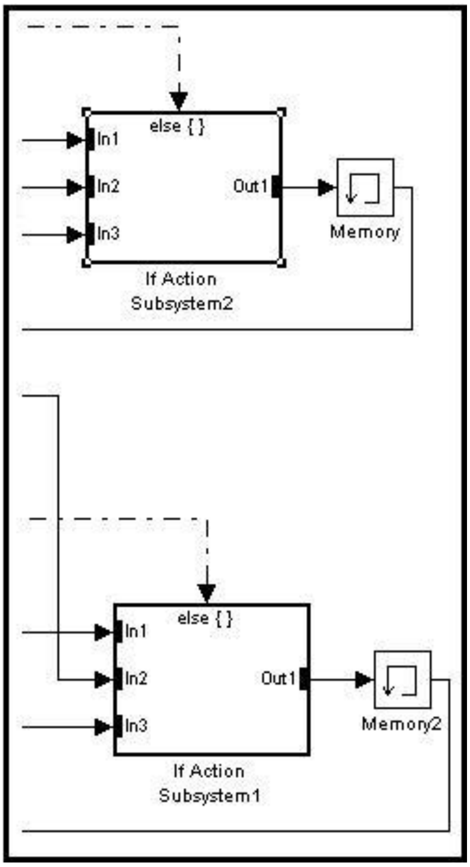


Figura 3.17 Bloques internos del Bloque if Action Subsystem1.

A continuación se detalla el bloque **If Action Subsystem2**, If Action Subsystem2 se divide en dos bloques que son If Action Subsystem y If Action Subsystem1 como se puede apreciar en la siguiente figura.

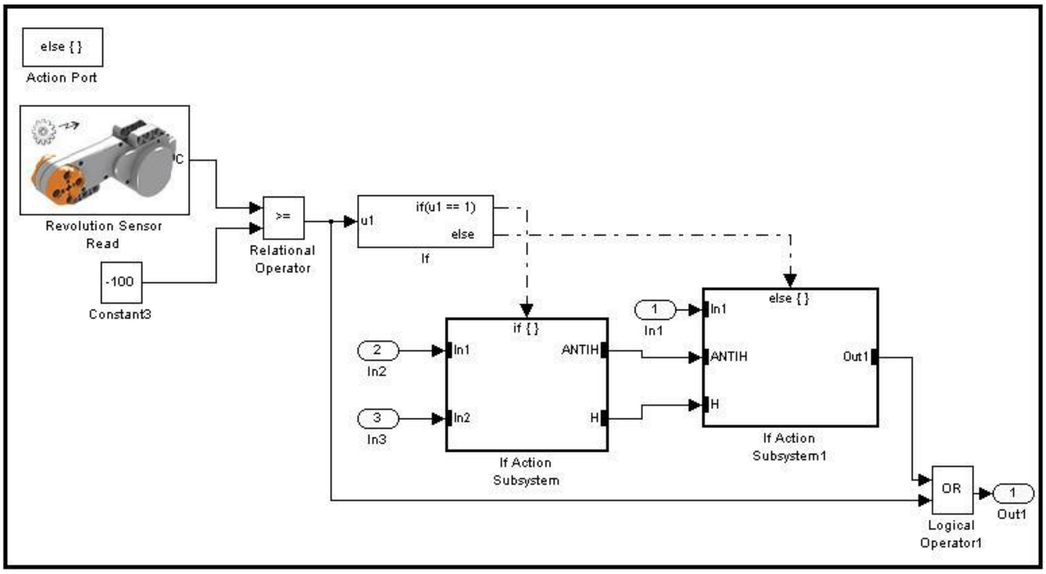


Figura 3.18 Bloques internos del Bloque if Action Subsystem2.

El contenido de **If Action Subsystem** sepuede apreciar a continuación.

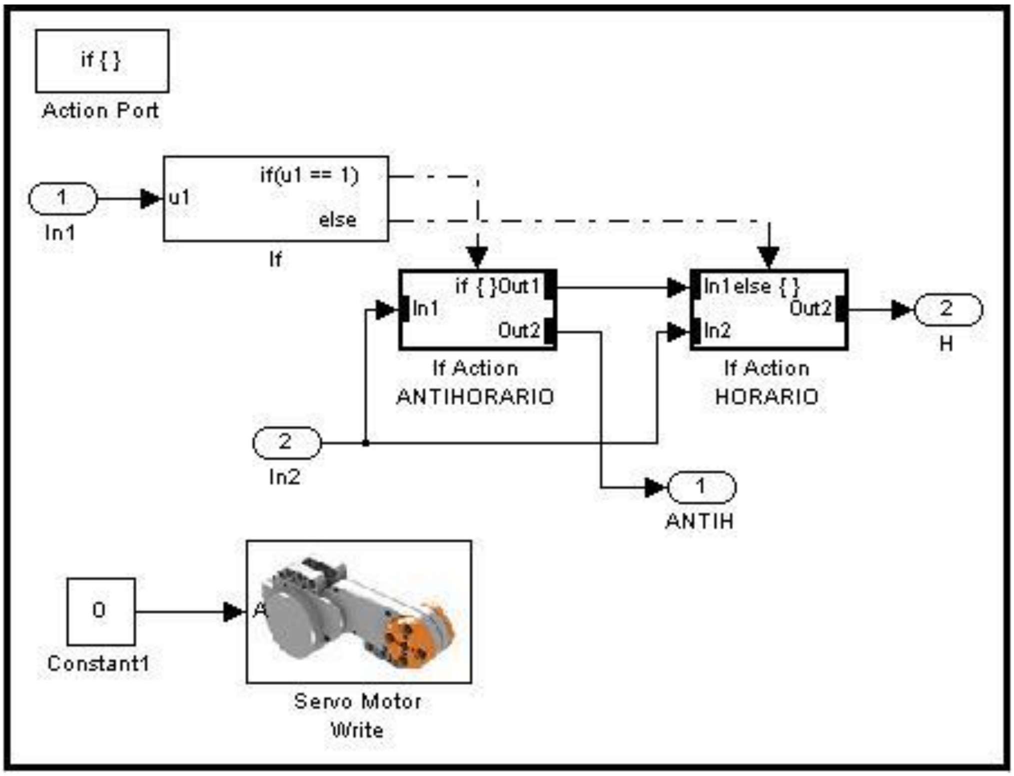


Figura 3.19 Bloques internos del Bloque if Action Subsystem.

El bloque **If Action ANTIHORARIO** es el encargado de hacer rotar el sensor ultrasónico mediante el Motor C hacia la derecha y también registra en la variable Out2 el dato que lee el sensor ultrasónico para utilizarlo en la decisión de giro del explorador, el contenido de este bloque es el siguiente.

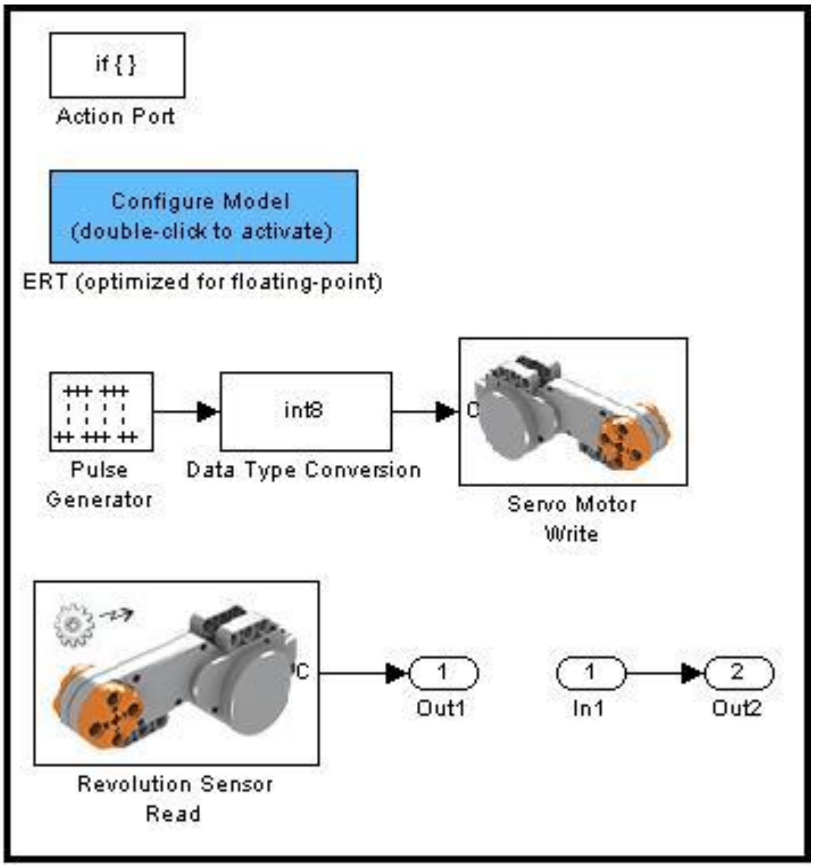


Figura 3.20 Bloques internos del Bloque If Action ANTIHORARIO.

El bloque **If Action HORARIO** es el encargado de hacer rotar el sensor ultrasónico hacia la izquierda mediante el Motor C y también registra en una variable Out2 el dato que lee el sensor ultrasónico para utilizarlo en la decisión de giro del explorador, su contenido es el siguiente figura.

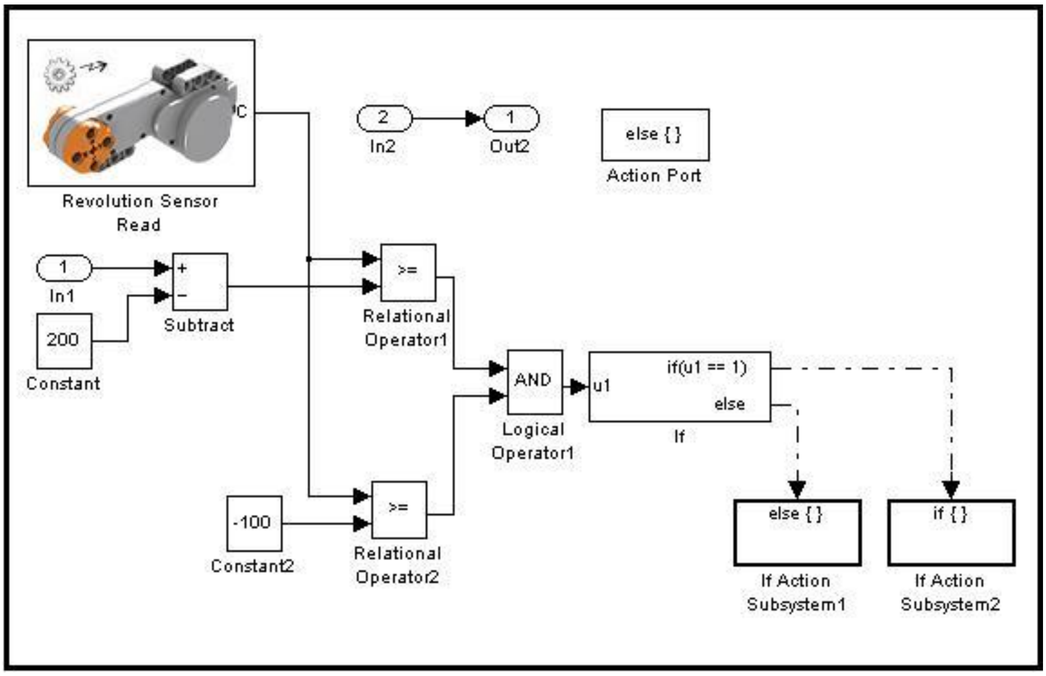


Figura 3.21 Bloques internos del Bloque If Action HORARIO.

El contenido de este bloque genera un PWM para el Motor C en If Action Subsystem2 y en If Action Subsystem1 frena al Motor C a un determinado ángulo de rotación.

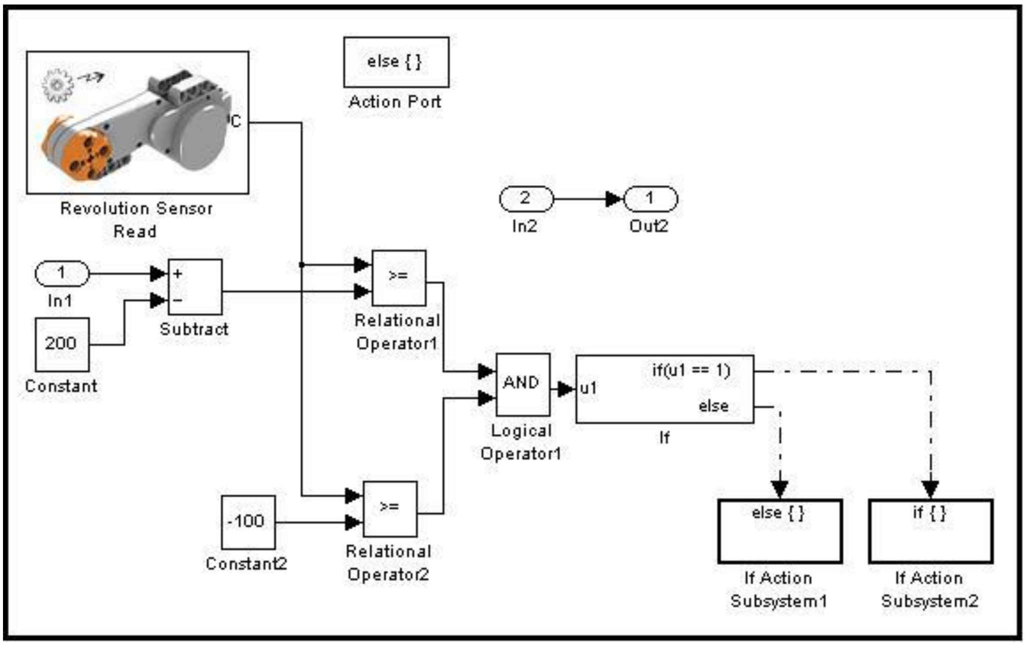


Figura 3.22 Bloques internos para frenar al Motor C.

Ahora se analizara el contenido de **If Action Subsystem1** de la siguiente figura, que es el bloque que se en donde el explorador decide a que dirección ir cuando detecta un obstáculo ya sea por el sensor de tacto o ultrasónico

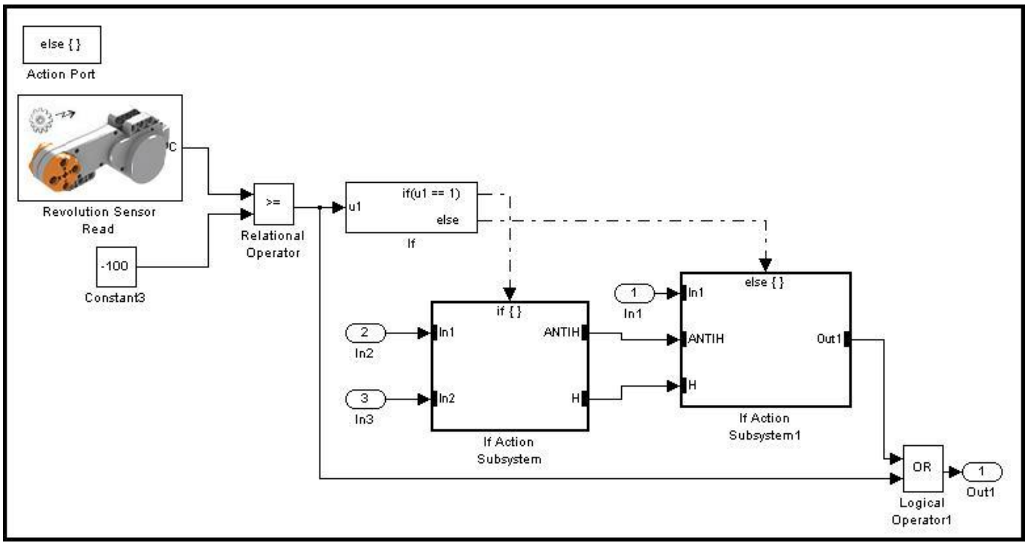


Figura 3.23 Bloques internos del Bloque If Action Subsystem1.

El bloque de **If Action Subsystem1** se tiene el proceso que detiene el Motor A y en If Action Subsystem se genera un PWM para el Motor A y se decide a que dirección el explorador debe ir.

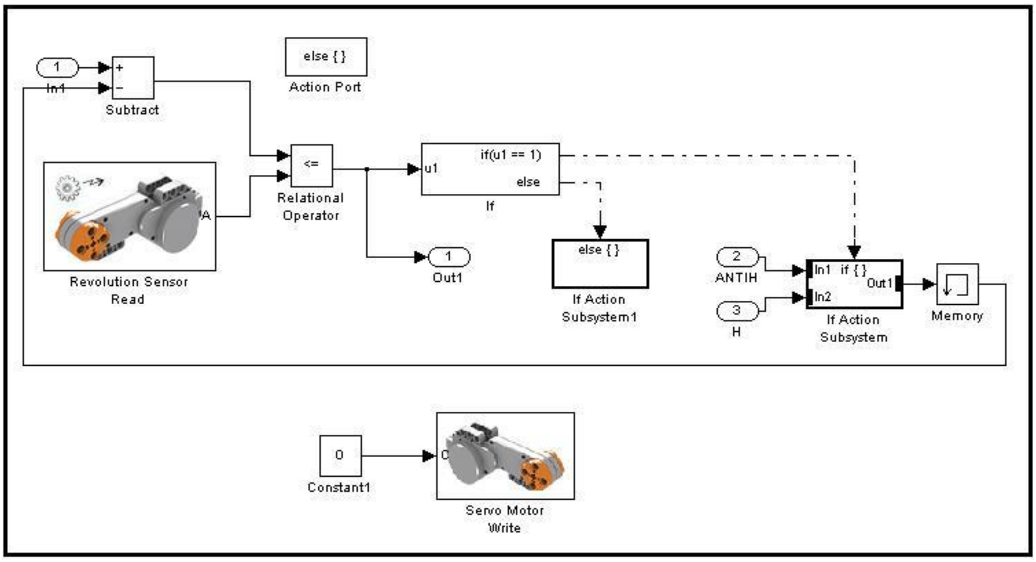


Figura 3.24 Bloques internos del Bloque If Action Subsystem1.

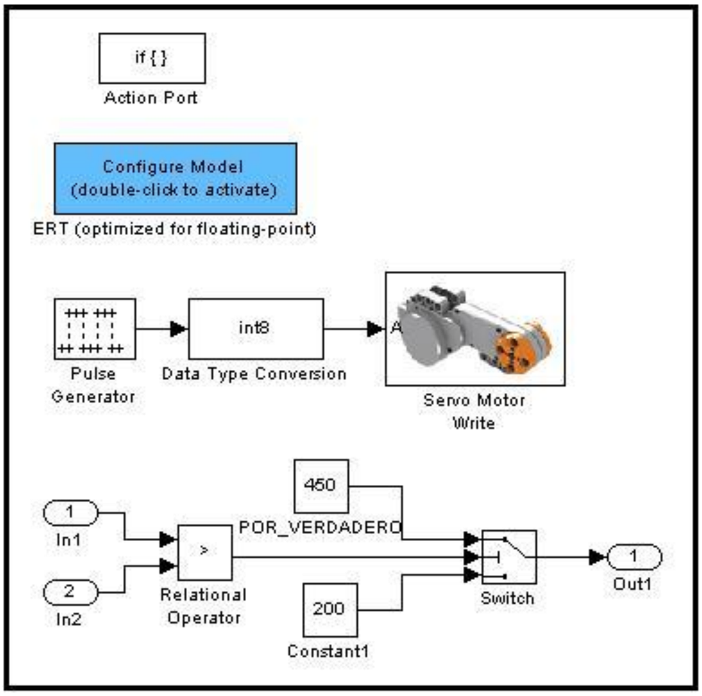


Figura 3.25 Bloques internos del Bloque If Action Subsystem.

3.3.2 Programa desde Simulink para el envío de datos por parte del robot explorador

Para el envío de datos desde Simulink se usó el bloque de bluetooth, de esta manera la adquisición de datos también es incluida en el programa realizado por simulink. En donde consta la interface del bluetooth así como la interface del sensor de sonido.

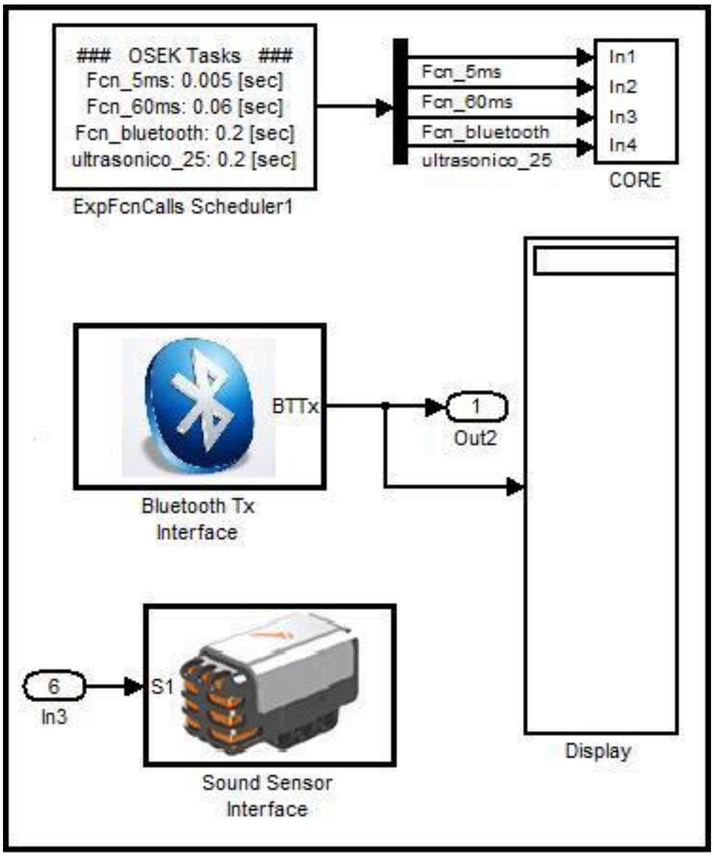


Figura 3.26 Bloques para la adquisición de datos.

La parte del programa que permite adquirir datos sensados por parte del sensor de sonido, es el que se encuentra en el bloque interno Function Calls System se muestra a continuación.

En él se muestra el procedimiento empleado para obtener los datos del sensor y la conversión realizada para enviarlos vía bluetooth. Cada indicar que la comunicación se realiza de la siguiente manera: se envía 4 paquetes de 8 bits y a eso se le agrega un bit de inicio y uno de parada.

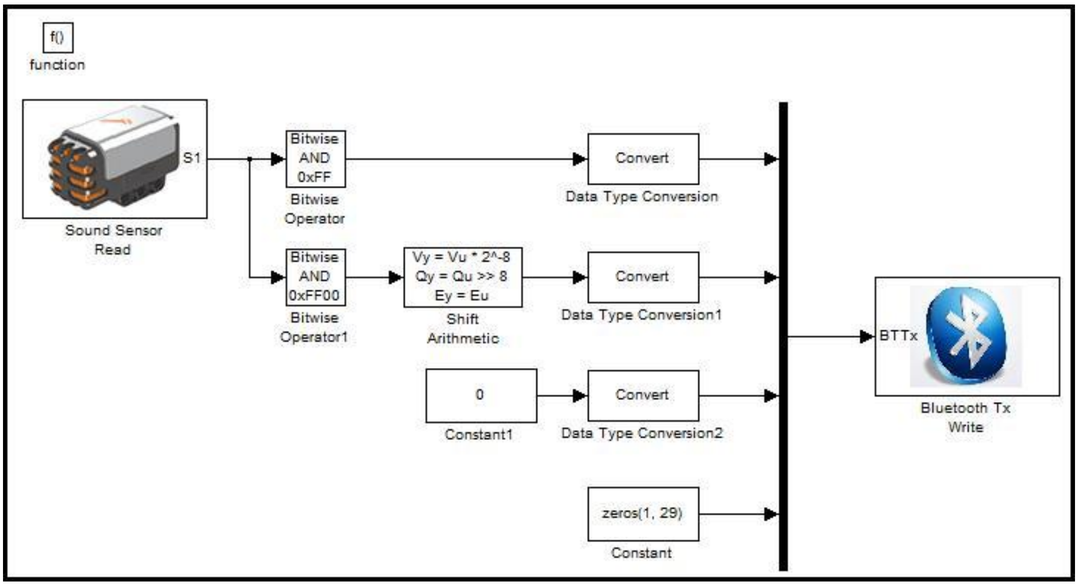


Figura 3.27 Bloques internos para la adquisición de datos.

Para el envío de datos se aplicó una AND para enviar el valor de la conversión analógico digital del sensor de sonido en dos tramas de 8 bits.

Y se aplicó un multiplexador para enviar todos los datos en una gran trama de 32 bits y a eso se le adiciona un bit de inicio y uno de parada.

3.4 Diseño de la aplicación realizada en Matlab.

Para la aplicación hecha en Matlab utilizamos la herramienta GUI que nos permitirá desarrollar una interfaz a nivel de usuario. Las ventajas que trae esta herramienta es que se puede enlazar cada botón con ciertas instrucciones tanto para la adquisición o graficación de los datos.

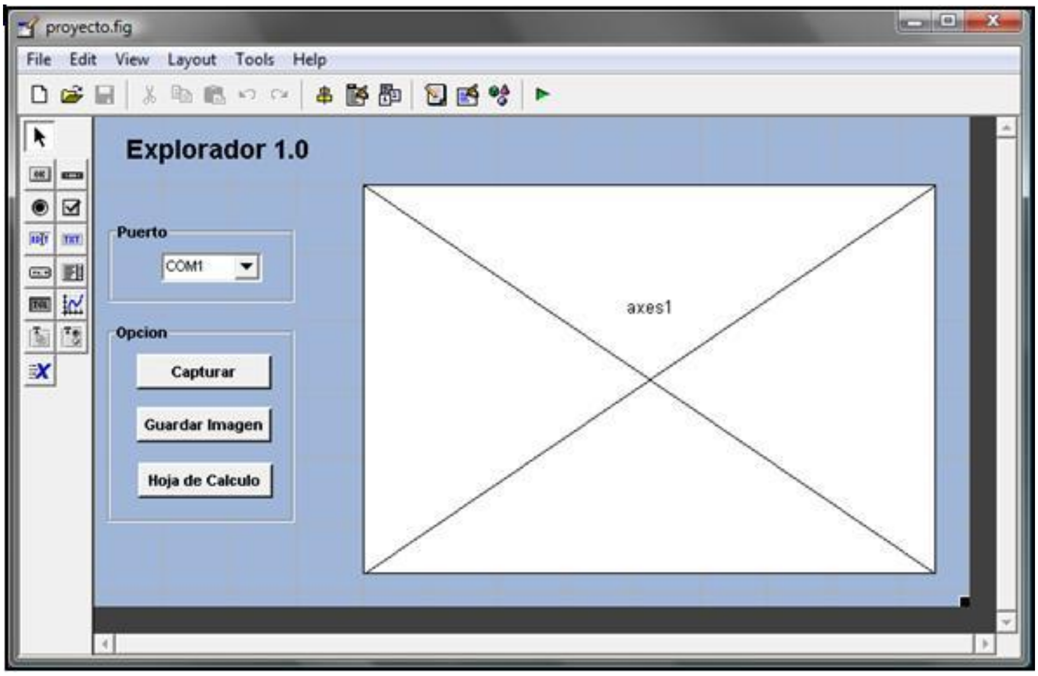


Figura 3.28 Ventana de GUI para la adquisición.

3.4.1 Desarrollo del programa en GUI

Para la interface de usuario se divide en algunas funciones de la cual se va a detallar y comentar cada una.

El botón capturar que se encuentra en la interface es la que se servirá para capturar los datos enviados por el Lego.

```
function btnCapturar_Callback(hObject, eventdata, handles)
global myPuerto
PACKET_SIZE = 34;   %la cantidad de bit recibidos
NUM_OF_CYCLE = 50;  %Es la cantidad de muestras a tomar
PERIOD = 0.2;       %es el periodo con el cual se va adquirir los datos
%El try-catch se lo usa para evitar errores
try
    %w32serial es la funcion que realiza la configuracion
    %del puerto a trabajar
    s = w32serial(myPuerto, 'BaudRate', 128000);
    %Se encarga de abrir el puerto a trabajar
    fopen(s);
catch
    %Se encarga de cerrar el puerto a trabajar
    fclose(s);
    disp(['Failed to connect with: ' port]);
    errordlg({'Puerto','No', 'Disponible'},'Error')
end
```

La función btnCapturar_Callback como se puede apreciar está compuesta por una variable global que nos servirá para saber con qué puerto se va a trabajar, PACKET_SIZE es la longitud de la trama del protocolo de comunicación, PERIOD esta variable representa el periodo con el que vamos a estar adquiriendo los datos.

El try catch se aplica para evitar errores al abrir un puerto de comunicación, la función que se usa para configurar el puerto de comunicación es w2serial que es propio del ecrobotNXT y fopen se aplica para abrir el puerto de comunicación, fclose se lo aplica para cerrar el puerto de comunicación a trabajar.

```
disp('Ready for DAQ via Bluetooth. Run your NXT!');
%El for es usado para adquirir un numero de muestras
for i = 1:NUM_OF_CYCLE
    packet = fread(s, PACKET_SIZE, 'uint8');
    %Comparamos la longitud del tamaño de trama q adquirimos
    if length(packet) == PACKET_SIZE
        SoundSensorDAQ(i) =1023-( packet(4)*256+packet(3));
        %UltrasonicSensorDAQ(i) = packet(5);
    end
end
```

Lo que se realiza en este fragmento de código es la adquisición y almacenamiento de datos en la variable SoundSensorDAQ para luego ser graficada en el axes, como se puede apreciar la variable PACKET es el paquete que contiene el valor enviado por el Lego y se le realizá el debido proceso para usar la variable.

```
a=SoundSensorDAQ';%Hacemos transpuesta a la matriz para graficar en excel
fclose(s);%cerramos el puerto
disp('DAQ is finished.');
assignin('base', 'SoundSensorDAQ', SoundSensorDAQ);
%Con plot mostramos la grafica en GUI
plot(PERIOD:PERIOD:NUM_OF_CYCLE*PERIOD, SoundSensorDAQ(1:NUM_OF_CYCLE));
title('Sound Sensor A/D ');
%Esta funcion es para graficar en EXCEL y establecemos las celda donde
%seran enviado los datos
xlswrite('datos',a(1:NUM_OF_CYCLE),'A1:A50');
grid on;
guidata(hObject, handles);
```

En estas líneas se encuentra tanto la parte de graficación en el axes como en la hoja de Excel. Se realizó la transpuesta a la variable SoundSensor para hacerlo una columna y enviarla a Excel.

```
%
function btnGuardar_Callback(hObject, eventdata, handles)
formatos = {'*.bmp','BMP (*.bmp)'};
[nomb,ruta] = uiputfile(formatos,'GUARDAR Datos');
if nomb==0, return, end
% Crear nueva figura
figura = figure;
% Unidades y posición
unidades  = get(handles.axes1,'Units');
posicion   = get(handles.axes1,'Position');
objeto_2  = copyobj(handles.axes1,figura);
% Modificar la nueva figura
set(objeto_2,'Units',unidades);
set(objeto_2,'Position',[15 5 posicion(3) posicion(4)]);
% Ajustar la nueva figura
set(figura,'Units',unidades);
set(figura,'Position',[15 5 posicion(3)+30 posicion(4)+10]);
% Guardar la gráfica
saveas(figura,[ruta nomb])
%Cerrar figura
close(figura)
```

La función btnGuardar se realizó para grabar una imagen .bmp de los datos que se muestran en el axe y en un futuro todas estas imágenes puedan ser almacenado en una base de datos.

Esta función tiene la característica de generar una imagen que tenga una similitud con la gráfica mostrada en el axe con su respectivos puntos de X y Y.

Para almacenar esta imagen se tiene que indicar el nombre de la imagen a almacenar.

```
function selectCom_Callback(hObject, eventdata, handles)
global myPuerto
fun =get(handles.selectCom,'Value');
switch fun
     case 1
        myPuerto='COM1';
     case 2
        myPuerto='COM2';
     case 3
        myPuerto='COM3';
     case 4
        myPuerto='COM4';
     case 5
        myPuerto='COM5';
     case 6
        myPuerto='COM30';
end

guidata(hObject, handles);
```

En esta función se hace el respectivo almacenamiento del Com seleccionado en la interface, para que pueda ser usado al momento de realizar la comunicación.

CAPÍTULO 4

4 SIMULACION Y PRUEBAS EXPERIMENTALES.

4.1 Implementación y pruebas del Proyecto

El diseño básicamente es del robot explorador, en el diseño consta los sensores de tacto y ultrasónico que son de nuestro interés y se encuentran colocados en su posición correspondiente para aprovechar todas sus características.

Figura 4.1 Diseño del robot explorador.

En la implementación del programa en Simulink, eso se lo realizó de manera directa con el robot explorador pues el procedimiento a seguir básicamente es realizar la programación como lo hemos detallado en el capítulo anterior para luego proceder a enviarlo vía USB al lego Mindstorms, más específicamente al cerebro del mismo.

Después de que el cerebro del robot explorador ya tiene el programa se procede a la comunicación entre el robot explorador y la computadora para ello se utilizó el software que brinda el propio bluetooth para lograr el enlace.

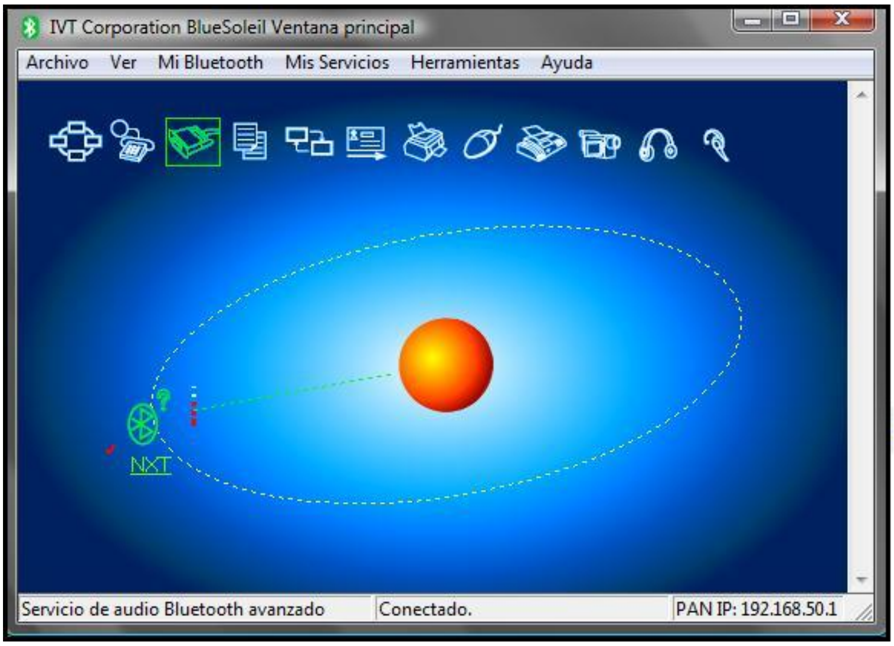


Figura 4.2 Software del bluetooth para el enlace de comunicación.

En la parte de adquisición de datos se tomó muy en cuenta la frecuencia a trabajar ya que las primeras pruebas salían datos erróneos de valores muy altos que hacían que la gráfica tenga un porcentaje de error muy alto, la solución a este problema fue cambiarle el tiempo de envío de recepción de datos tanto en el robot como en la aplicación.

En la interfaz que se realizó en GUI se muestran los datos obtenidos, esta gráfica presenta los datos que el sensor de sonido sensa por un tiempo de 20 segundos, todo esto después que el usuario presiona el botón de **Capturar.**

La elaboración de esta interfaz fue desarrollada en detalle en el capítulo anterior, aquí ya mostramos la pantalla que se genera cuando se pone a correr el programa, sin necesidad de manejar directamente Simulink, la imagen que se muestra está relacionada con el nivel de ruido presente en el ambiente a medida que el robot explorador se encuentra en movimiento.

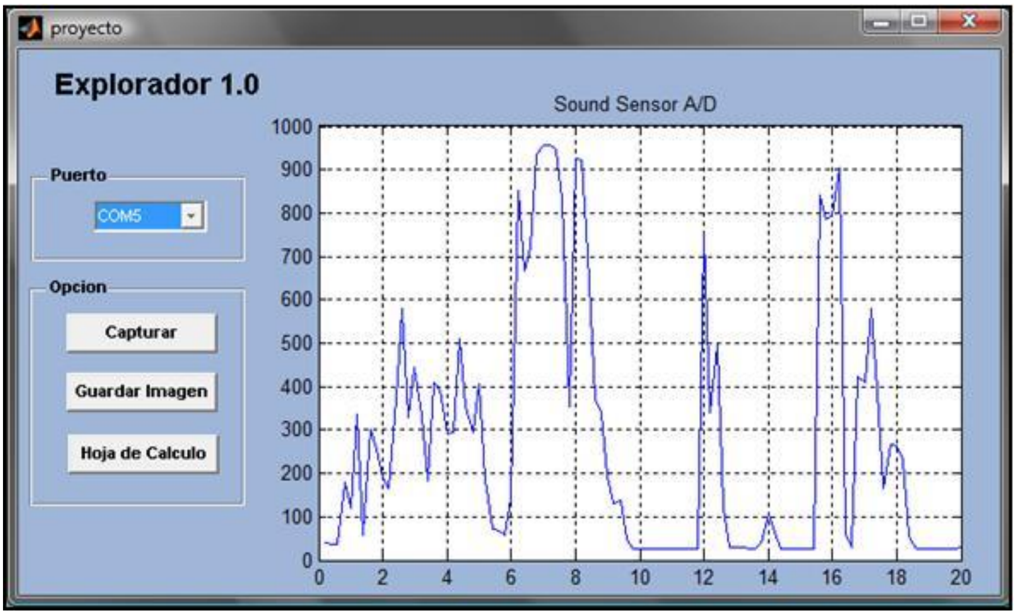


Figura 4.3 Interfaz para la adquisición de datos.

Cabe indicar que para la ejecución de la interfaz debemos escoger de manera correcta el COM de comunicación, es por eso que en la interfaz optamos por colocar una lista desplegable para que sea el mismo usuario el que elija el COM que se abrió al proceder con el enlace Bluetooth.

Como una opción para el usuario optamos por colocar un botón de **Guardar Imagen,** lo que permite guardar los datos obtenidos en un directorio que el mismo establece.

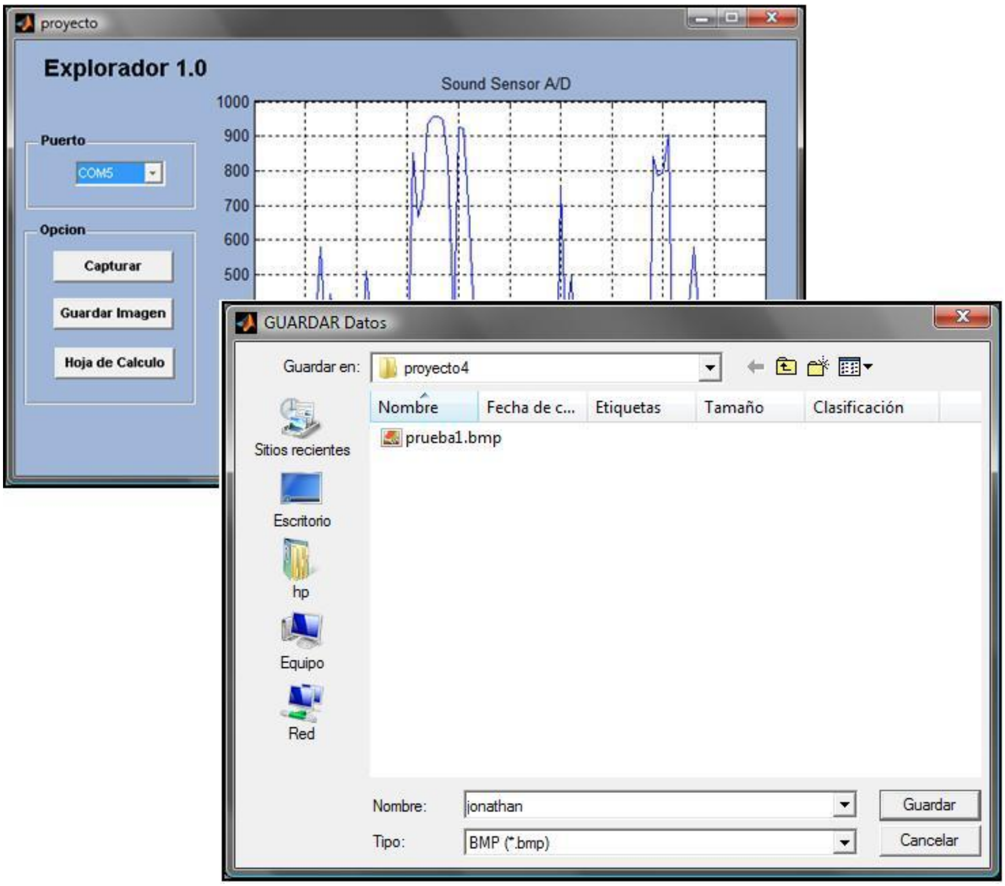


Figura 4.4 Ventana que permite Guardar Imagen.

De esta manera el usuario podrá tener guardado todos los datos que se van adquiriendo por parte del robot explorador. Esta imagen se guarda en formato .bmp, esto se pudo obtener gracias a los comandos propios en la programación de GUI, obteniendo imágenes en donde se puede apreciar tanto el tiempo como los datos máximo y mínimo del sensor.

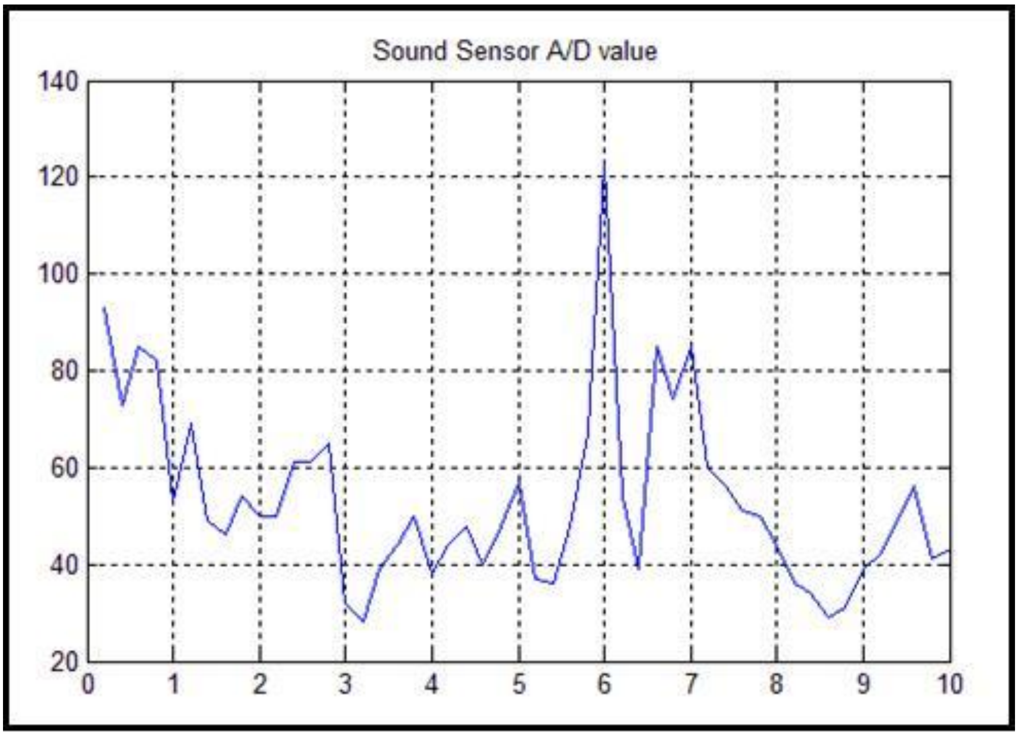


Figura 4.5 Imagen .bmp que se guarda.

Otra opción que le añadimos en la programación en GUI es la de guardar los datos en una hoja de Excel por medio del botón **Hoja de Excel**, y poder obtener una gráfica de los datos obtenidos.

Esto se lo realiza por medio de xlswrite que forma parte de los comandos de GUI que permite la interacción con hojas de EXCEL.

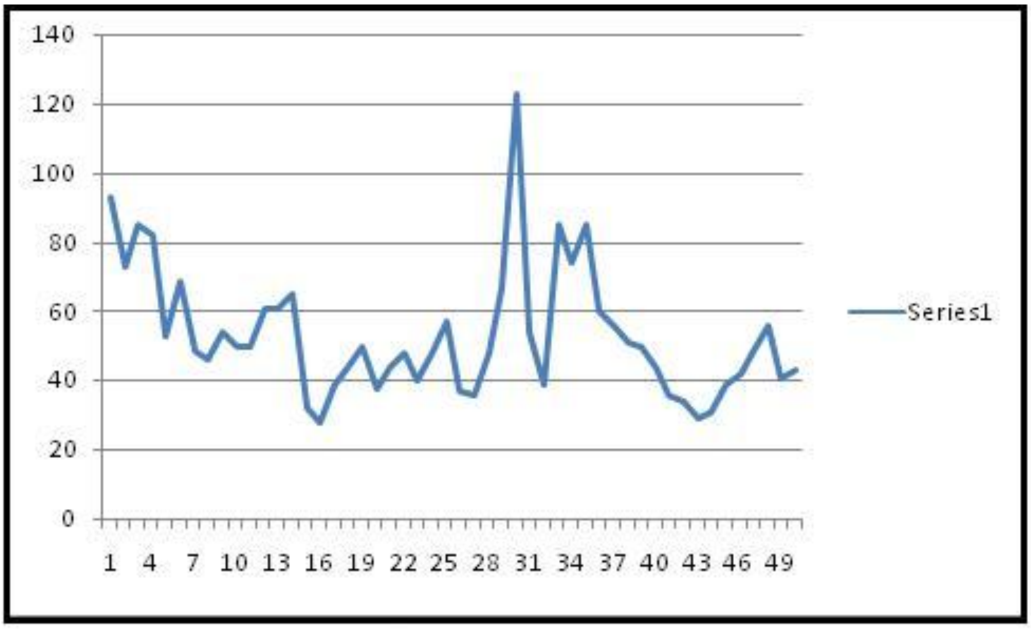


Figura 4.6 Gráfica que se obtiene de Excel.

El único problema con este tipo de gráfica es que lo hace en función del número de datos adquiridos y no en función del tiempo, aún así es una buena alternativa para el procesamiento de datos.

CONCLUSIONES

1. Con el desarrollo del proyecto llegamos a la conclusión que la elección de Simulink para la programación del Lego NXT fue correcta, ya que este no presentó un nivel de dificultad muy grande y hace que el robot explorador no tenga una dependencia con una base o computador para cumplir su propósito de movilidad, sin que afecte el factor de distancia.
2. Se concluyó que Matlab es una óptima herramienta para el desarrollo de procesos con el lego Mindstorms, pues por medio del bluetooth podemos enviarle un sin número de órdenes y éste las cumple a cabalidad, la única desventaja de trabajar con un archivo .m de Matlab es la demora del arranque del programa.
3. La adquisición que se realizó gracias a las herramientas de Matlab y Simulink permitió hacer la lectura eficiente del sensor del robot explorador, para luego con la herramienta de GUI presentar dichos datos en una interfaz que permita al usuario manejar las opciones que esta interfaz le permite.

4. Se logró utilizar todas las herramientas necesarias para desarrollar un sistema en Simulink de tal manera que controle el robot explorador, siendo ECRobot una de las herramientas indispensables para alcanzar este fin, pues permitió agregar un toolbox de Lego NTX en Simulink para realizar la programación necesaria para lograr el objetivo planteado.
5. Por medio de las prácticas realizadas se conoció que en la parte del muestreo la frecuencia no sea muy alta por lo que nos dió errores a la hora de receptar datos por el motivo de que el convertidor analógico digital propio del lego está dentro de un microcontrolador y este tiene un retardo para dicha conversión, por lo que se obtuvo lecturas erróneas que llegó a valores de 0, con las muestras que se desarrolló se conoció que el tiempo de muestreo adecuado es de 0.2 segundos, este tiempo fué el que no nos trajo errores a la hora de trabajar y obtener las muestras.

RECOMENDACIONES

1. Cuando se desarrolló el software para dar movimiento al LEGO el algoritmo solo realizaba de forma correcta la función una sola vez y luego hubo fragmentos del proceso que no se desarrollaban, este problema surgió debido a que ciertos segmentos del proceso quedaban validados al término de éste. La solución a este problema fue utilizar una variable externa al proceso, de esta manera se pudo concretar el desarrollo de esta aplicación en Simulink.
2. Otro problema que surgió al momento de realizar el software para la aplicación fué que hubo que hacer la lectura del sensor ultrasónico en un bloque diferente puesto que este sensor no trabajaba de forma eficiente a la misma frecuencia de muestreo de los demás bloques esto se debe a que el sensor necesita un determinado tiempo para realizar una lectura precisa.

3. Otra solución para el problema antes mencionado fué la de hacer coincidir la frecuencia de muestreo de los bloques de lectura del sensor ultrasónico y la del movimiento del explorador, pero al hacer esto se perdía precisión en los movimiento de los motores, es por esta razón que no se lo realizó de esta forma.
4. Si es que queremos el robot tenga un rango de exploración más elevado es necesario cambiar el medio de comunicación con la computadora, para este caso puede ser los módulos zigbee o módulos wireless que pueden tener un alcance de hasta 300 metros y la implementación de estos módulos en el lego no tienen un nivel muy alto de dificultad ya que se podría utilizar los conocimientos adquiridos en la materia para realizar una comunicación I2C entre estos módulos y el lego y se podría usar un microcontrolador de intermediario entre los dispositivos para que se realice la comunicación.

ANEXO A

Pruebas del Hardware y Software de Lego Mindstorms

En las primeras pruebas que se realizó antes de desarrollar el proyecto de la materia de graduación, fué el desarrollo del mismo robot explorador en el software del Lego Mindstorms, para lo cual se siguió los siguientes pasos:

1. Se conectó cada uno de los sensores en diferentes puertos y se probó con el mismo menú que trae el lego Mindstorms (ladrillo) y cada una de las pruebas fueron satisfactorias.
2. Se probó los motores del kit, tanto para realizar movimientos como para colocarlos en una posición específica, para lo cual se utilizó el programa que trae incluido el ladrillo para hacer dichas operaciones.
3. Otras de las pruebas realizadas fue probar la conexión que tenía el lego con la computadora, se hizo uso del puerto USB para dicha comunicación y se puedo apreciar con satisfacción la lectura de los sensores y de los motores.

4. El siguiente paso que se realizó después de las pruebas del kit fué cargarle un programa básico utilizando el software que viene incluido en el kit, el cual no presentó ningún problema porque ser un programa educativo para niños y el nivel de dificultad es muy bajo

Luego de lograr un manejo adecuado del software del propio lego, se realizó el robot explorador para verificar su funcionamiento y manipulación de los bloques del software, y desarrollar creatividad con las partes del lego, para luego sí desarrollarlo en Matlab y Simulink.

El robot explorador fue desarrollado de tal manera que se logró el siguiente diseño para poder verificar su funcionamiento.

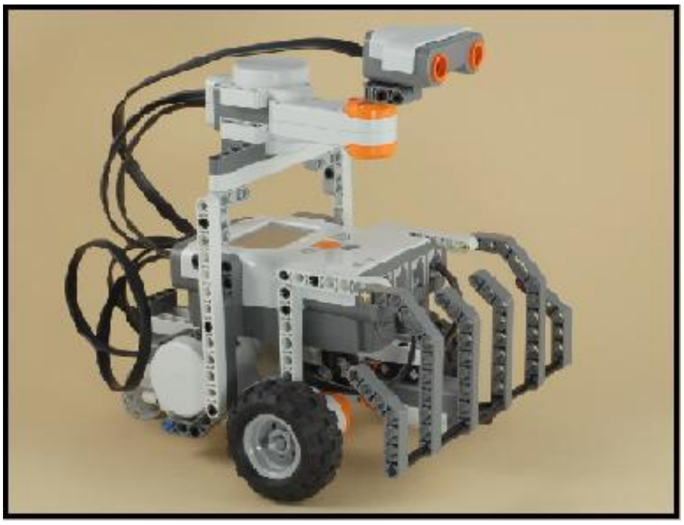


Foto 1: Diseño del Robot Explorador.

Una vez desarrollado el robot se procederá a realizar el programa para cargarlo en el cerebro del robot lego, a continuación la pantalla que aparece cuando se carga un programa al cerebro.

Foto 2: Ventana que indica cargar programa.

El diagrama que se usará será el que se muestra a continuación, hará uso de los señores de tacto y ultrasónico así como de los motores para su movimiento, cada uno tendrá opciones que permitan generar un programa que de tal manera creen una secuencia de instrucciones.

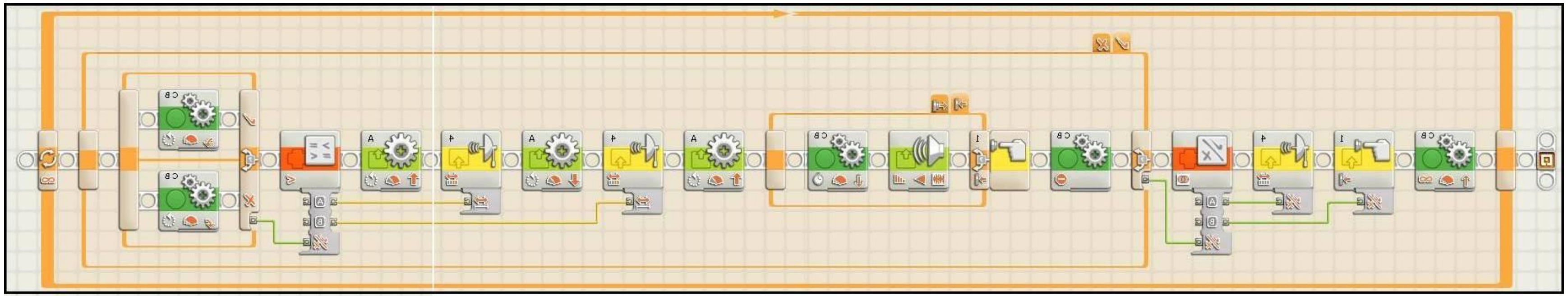


Foto 3: Diseño del programa para el Robot Explorador.

BIBLIOGRAFIA

[1]. Excel con Matlab, Página HTML,

http://www.robotshop.ca/hitec-hs311-servo.html

[2]. Excel con Matlab, Página HTML,

http://www. blogs.mathworks.com

[3]. Lego Mindstorms, Página HTML,

http://www.wikipedia.org/wiki/Lego_Mindstorms

[4]. NXT características, Página HTML,

www. electrosystemss.blogspot.com

[5]. Robot spirit, Página HTML,

www. eduteka.org/RoboticaSondas.php

[6]. Lego con Matlab, Página HTML,

www. blogs.mathworks.com

[7]. Firmware de lego, Página HTML,

www. mindstorms.lego.com

[8]. Legos-OSEK, Página HTML,

www. mindstorms.lego.com

[9]. Legos-ECRobot, Página HTML,

http://lejos-osek.sourceforge.net/api.htm

[10]. Legos-nxtosek, Página HTML,

http://lejos-osek.sourceforge.net/nxtOSEK_hack.htm

[11]. Simulink y Excel, Archivo PDF,

www.tecnun.es/asignaturas/.../aprendainf/matlab70/mat71.pdf

[12]. MECHATRONICS, Archivo PDF,

www.microchip.com

[13]. Introducción a MATLAB / SIMULINK, Archivo PDF,

www.fceia.unr.edu.ar/dsf/files/TP_Introd.pdf

[14]. Aprenda Matlab, Cátedra D.S.F, Archivo PDF,

www.tecnun.es/asignaturas/.../aprendainf/matlab60/matlab60.pdf