



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

**" USO DE MATLAB Y SIMULINK PARA EL CONTROL  
DE ROBOTS Y LA OBSERVACION DE LOS SENSORES  
DE SONIDO Y ULTRASONICO."**

**INFORME DE MATERIA DE GRADUACIÓN**

**Previo a la obtención del Título de:**

**INGENIERO EN ELECTRICIDAD ESPECIALIDAD  
ELECTRÓNICA Y AUTOMATIZACIÓN INDUSTRIAL**

**Presentado por:  
Jimmy Eduardo Alvarado Ortiz.  
Juan Carlos Espinoza Alvarez.  
Diego Fernando Guñay Saldaña.**

**GUAYAQUIL – ECUADOR  
2009**

## AGRADECIMIENTOS

Primeramente a Dios quien ha sido mi guía y supo darme fuerzas en los momentos más difíciles.

A mis padres quienes me infundieron la ética y el rigor que guían mi transitar por la vida.

A todos mis familiares, quienes siempre me apoyaron incondicionalmente.

A mi director de tesis, por su asesoramiento científico y estímulo para seguir creciendo intelectualmente.

Jimmy Eduardo Alvarado Ortiz.

En primer Lugar a Dios que me ha dado la vida y las facultades necesarias para poder llevar a cabo este trabajo.

A mis padres que con su esfuerzo, constante motivación y apoyo moral me llevaron a la culminación exitosa de mi carrera universitaria.

A mi director de tesis por su amabilidad, buena disposición, paciencia, y tiempo dedicado para que este proyecto se llevara a cabo.

Juan Carlos Espinoza Alvarez.

### **A MIS PADRES:**

Que siempre me han dado su apoyo incondicional y a quienes debo este triunfo profesional, por todo su trabajo y dedicación para darme una formación académica y sobre todo humanista y espiritual.  
De ellos es este triunfo y para ellos es todo mi agradecimiento.

**Diego Fernando Guñay Saldaña.**

## DEDICATORIA

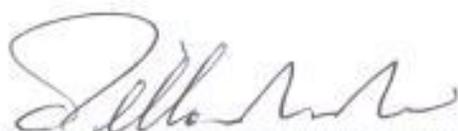
Dedico este proyecto y toda mi carrera universitaria a Dios por ser quien ha estado a mi lado en todo momento dándome las fuerzas necesarias para continuar luchando día tras día y seguir adelante rompiendo todas las barreras que se me presenten.

A mis padres, ya que gracias a ellos soy quien soy hoy en día, fueron los que me dieron ese cariño y calor humano necesario, son los que han velado por mi salud, mis estudios, mi educación alimentación entre otros, son a ellos a quien les debo todo, horas de consejos , de regaños, de reprimendas de tristezas y de alegrías de las cuales estoy muy seguro que las han hecho con todo el amor del mundo para formarme como un ser integral y de las cuales me siento extremadamente orgulloso.

## TRIBUNAL DE GRADUACIÓN



Ing. Carlos Valdivieso A.  
DIRECTOR DE TESIS



Ing. Hugo Villavicencio V.  
DELEGADO DEL DECANO



Ing. Jorge Aragundi  
SUBDECANO DE LA FIEC

## DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesis de Grado, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL"

(Reglamento de Graduación de la ESPOL)

-----  
Jimmy Alvarado

-----  
Juan Espinoza

-----  
Fernando Guñay

## RESUMEN

Este proyecto presenta como objetivo principal el desarrollo de un sistema de silla automática inteligente, que sea capaz de responder a las exigencias del usuario por medio de la intensidad con que sean emitidos los sonidos del dispositivo emisor de audio, independientemente de la distancia que esta se encuentre ubicada y será capaz de llegar al mismo a pesar de los obstáculos que se le presenten durante su recorrido.

En el primer capítulo de este texto se describen el alcance y las limitaciones que se puedan presentar durante la implementación de este sistema. Este capítulo nos proporciona una idea general del funcionamiento del sistema y de sus posibles fallas, así como también se mencionan soluciones similares utilizadas en implementaciones comerciales existentes en el mercado.

En el segundo capítulo se hace un análisis teórico de las tecnologías utilizadas en el diseño del prototipo. Estas tecnologías consisten en: un kit LEGO MINDSTORMS NXT, del mismo que se utilizarán los sensores ultrasónico y de sonido como principales dentro del desarrollo del prototipo de silla, el dispositivo bluetooth, el que cumple el importantísimo papel de protocolo de comunicación entre el robot y el software implementado. El software a ser utilizado es MATLAB-SIMULINK, el cual permite

desarrollar las condiciones de funcionamiento del prototipo de silla. Las características y herramientas de dicho Software se presentarán en este capítulo.

En el tercer capítulo se presentarán los detalles del software y hardware utilizados durante el desarrollo del sistema. Para hacer una descripción detallada del funcionamiento del sistema se recurrirá a un diagrama de bloques que explicará de manera sencilla la forma en que las distintas tecnologías implementadas interactúan de tal manera que sea posible un desenvolvimiento armónico, eficiente e integral de cada uno de los elementos que conforman al prototipo de silla inteligente.

Finalmente, el cuarto capítulo se refiere al funcionamiento del prototipo desarrollado a menor escala, esto es: simulación en tiempo real, datos adquiridos por el mismo a través del puerto de comunicaciones bluetooth tales como el tiempo que tarda el prototipo en encontrar al usuario y las decisiones tomadas del robot por medio de los datos adquiridos de los sensores .

# ÍNDICE GENERAL

RESUMEN

INDICE GENERAL

INDICE DE FIGURAS

INTRODUCCIÓN

## 1 DESCRIPCIÓN GENERAL DEL SISTEMA

1.1 Alcance y limitaciones del proyecto.....	2
1.1.1 Descripción del proyecto.....	3
1.1.2 Estrategia implementada.....	4
1.1.3 Limitaciones del proyecto.....	7
1.2 Análisis de soluciones existentes en el mercado.....	9

## 2 FUNDAMENTACIÓN TEÓRICA

2.1 Kit Lego Mindstorms NXT.....	10
2.1.1 Descripción.....	11
2.1.2 Micro-controlador.....	11
2.1.3 Entradas y salidas.....	11
2.1.4 Comunicaciones.....	12
2.1.5 Firmware.....	13
2.1.6 Motores.....	13
2.1.7 Piezas especiales.....	15

2.1.8	Ruedas.....	17
2.1.9	Alimentación eléctrica.....	17
2.1.10	Sensores de Lego Mindstorms NXT.....	18
2.2	MATLAB.....	22
2.2.1	Introducción.....	22
2.2.2	Herramientas de MATLAB para Lego Mindstorms NXT.....	23
2.3	Simulink.....	24
2.4	Bluetooth.....	25
<b>3</b>	<b>DISEÑO DEL SOFTWARE Y HARDWARE DEL PROTOTIPO DE SILLA</b>	
3.1	Diagrama de bloques del diseño propuesto.....	26
3.2	Proyecto de prueba previo a la implementación del prototipo de silla.....	46
3.3	Análisis del código de programación en MATLAB.....	50
<b>4</b>	<b>FUNCIONAMIENTO DEL SISTEMA</b>	
4.1	Implementación del sistema.....	61
4.2	Manejo y prueba del sistema.....	63
	<b>CONCLUSIONES</b>	
	<b>RECOMENDACIONES</b>	
	<b>ANEXOS</b>	
	<b>BIBLIOGRAFIA</b>	

## ÍNDICE DE FIGURAS

FIGURA 1.1	Prototipo de silla inteligente.....	4
FIGURA 1.2	Sistemas de comunicación con el robot.....	6
FIGURA 2.1	Bloque NXT.....	11
FIGURA 2.2	Motor NXT.....	14
FIGURA 2.3	Accesorios NXT y RXT.....	16
FIGURA 2.4	Ruedas NXT y RXT.....	17
FIGURA 2.5	Esquema interno básico de un sensor .....	19
FIGURA 2.6	Sensor Ultrasonico NXT.....	19
FIGURA 2.7	Sensor de sonido NXT .....	20
FIGURA 2.8	Sensor de tacto NXT.....	21
FIGURA 2.9	Sensor de luz NXT.....	21
FIGURA 2.10	Interacción NXT – MATLAB.....	23
FIGURA 2.11	Ejemplo de programación NXT en Simulink.....	24
FIGURA 2.12	Comunicación bluetooth .....	25
FIGURA 3.1	Diagrama del Diseño Propuesto.....	27
FIGURA 3.2	Componentes de la instalación del cygwin.....	28
FIGURA 3.3	Ventana del cygwin.exe.....	28
FIGURA 3.4	Instalador Cygwin .....	29
FIGURA 3.5	Instalando el Cygwin.....	29

FIGURA 3.6	Cygwin en modo default .....	30
FIGURA 3.7	Cygwin en modo default .....	30
FIGURA 3.8	Cygwin instalado .....	31
FIGURA 3.9	Demostración del Cygwin correctamente instalado .....	31
FIGURA 3.10	Archivos del instalador GNUARM .....	32
FIGURA 3.11	Ejecutando GNUARM.exe.....	32
FIGURA 3.12	Instalador GNUARM en cygwin .....	33
FIGURA 3.13	Librerías instaladas en el GNUARM.....	33
FIGURA 3.14	Instalando GNUARM.....	34
FIGURA 3.15	GNUARM Instalado .....	34
FIGURA 3.16	Archivos NextTool .....	35
FIGURA 3.17	Archivos en el lms_arm_jch.zip.....	35
FIGURA 3.18	Archivos juntos al NextTool.....	36
FIGURA 3.19	Cmd de Windows .....	36
FIGURA 3.20	Cmd de Windows con el NextTool.....	37
FIGURA 3.21	Comprobación del programa instalados .....	37
FIGURA 3.22	Archivos contenidos en nxtOSEK .....	38
FIGURA 3.23	Instalación de librería USB .....	38
FIGURA 3.24	Comprobando la librería USB .....	39
FIGURA 3.25	Archivos de eecrobotNXT .....	39

FIGURA 3.26 ErobotNXT ubicado en MATLAB .....	40
FIGURA 3.27 Ejecución de MATLAB .....	40
FIGURA 3.28 Ubicación de erobotnxtsetup.m en Matlab .....	41
FIGURA 3.29 Erobotnxtsetup.m .....	41
FIGURA 3.30 Ubicación de archivos mediante erobotnxtsetup.m .....	42
FIGURA 3.31 Ejemplos de programas elaborados en MATLAB .....	43
FIGURA 3.32 Software BlueSoleil 2.0.....	44
FIGURA 3.33 Conexión establecida entre PC y NXT .....	44
FIGURA 3.34 Prototipo de silla inteligente.....	45
FIGURA 3.35 Sensor ultrasónico .....	45
FIGURA 3.36 Sensor de sonido.....	46
FIGURA 3.37 Escorpión NXT.....	47
FIGURA 3.38 Escorpión NXT en funcionamiento.....	48
FIGURA 3.39 Programación del escorpión en NXT-G.....	50
FIGURA 4.1 Modelo del prototipo de silla.....	61
FIGURA 4.2 Prueba de Conectividad.....	63
FIGURA 4.3 Ejecución del programa "SILLA.m" en la ventana de comandos de MATLAB.....	64
FIGURA 4.4 Puesta en marcha del Sistema.....	65
FIGURA 4.5 Matriz de datos de "SILLA.m".....	66

## INTRODUCCIÓN

El progresivo avance de la tecnología ha permitido al ser humano desarrollar sistemas que le permitan realizar sus actividades cotidianas de una manera mucho más sencilla, rápida, eficiente y fácil; optimizando así el tiempo empleado para elaborar las mismas y cada vez logrando más actividades que le eran imposibles de realizar por sí mismo.

Con el fin de proporcionar facilidad a personas en general y la capacidad de contar con una silla inteligente que esté siempre a su alcance, se ha desarrollado un prototipo de silla inteligente a menor escala. Su característica principal es poseer la capacidad de búsqueda y seguimiento de una forma eficiente y precisa a la persona que requiera de esta, quedando a su disposición y siguiéndola a donde este desee brindando sus comodidades.

Para lograr nuestro propósito recurrimos al uso del LEGO MINDSTORMS NXT para la implementación física de nuestro prototipo de silla, también con la tecnología BLUETOOTH como protocolo de comunicación, se enviarán datos al prototipo de silla, utilizando también MATLAB-SIMULINK como software de programación para establecer las condiciones de funcionamiento y a la vez el muestreo de los resultados obtenidos. En el primer capítulo de este texto, presentado a continuación, se describirá en detalle el alcance y las limitaciones del presente proyecto.

# CAPÍTULO 1

## 1 DESCRIPCIÓN GENERAL DEL SISTEMA

### 1.1 Alcance y limitaciones del proyecto

El objetivo principal de este proyecto es poder desarrollar un sistema de silla automática inteligente y útil, capaz de responder a los requerimientos del usuario por medio de la intensidad con que sean emitidos los sonidos del dispositivo emisor de audio, indistintamente de la distancia que esta se encuentre ubicada y será capaz de llegar al mismo a pesar de los obstáculos que se le presenten durante su recorrido, para finalmente poder ser utilizada de la forma en que se desee. Además, promover el uso de MATLAB-SIMULINK para el control de robots, demostrando eficiencia y gran aporte al desarrollo tecnológico en nuestro medio.

Para poder desarrollar este tema se elaboró un prototipo de silla inteligente mediante el uso de LEGO MIDSTORMS NXT para la implementación física del mismo, complementando con los sensores que este kit posee y la constante comunicación y manipulación de los mismos a través de un software avanzado como lo es MATLAB-SIMULINK, el cual, mediante la programación y respectiva comunicación mediante BLUETOOTH, quedarán establecidos todos los mecanismos y las condiciones necesarias que deberá cumplir la silla para que

esta pueda detectar al usuario automáticamente y finalmente quede a su disponibilidad.

### 1.1.1 Descripción del Proyecto

Básicamente, nuestro objetivo principal será el desarrollo de un prototipo de silla inteligente, comandada por la intensidad que emitan los dispositivos de sonido, como se mencionó anteriormente, pudiendo brindar facilidades al usuario, principalmente la disposición de una silla para uso personal sin necesidad de ir en búsqueda de una.

A partir de este objetivo principal generalizado ya mencionado, se pueden derivar varios objetivos específicos, como por ejemplo, desarrollar un sistema para poder ayudar a personas discapacitadas para que se sientan lo suficientemente capaces e independientes en cuanto a la búsqueda y acceso de su silla; desarrollo de un sistema de sillas inteligentes capaces de seguir al usuario dentro de una biblioteca mientras el mismo está en búsqueda de algún libro; desarrollar un sistema de ayuda para las personas mayores, ya que debido a su edad, se sienten agotadas y necesitan descansar frecuentemente; entre otras.

Como características principales de este proyecto, tenemos:

- Garantizar al usuario la disponibilidad de una silla a su alcance, es decir, dentro de un límite establecido.

- Proporcionar una optimización del tiempo y esfuerzo a personas en general.
- Permitir al usuario disfrutar de la comodidad de una silla en cualquier lugar y en cualquier momento.



Figura 1.1 Prototipo de silla inteligente.

### 1.1.2 Estrategia Implementada

El diseño de un prototipo de silla automática inteligente comandada por la intensidad de sonido receptado, consiste básicamente en una constante interacción del usuario con la silla para que esta siempre se encuentre atenta al dispositivo emisor de sonido que el usuario posea en sus manos; al receptar la mayor intensidad de sonido, la silla procederá a avanzar en la dirección en la cual fue detectada dicha intensidad y, mediante el uso del sensor ultrasónico que esta

posee, se podrá movilizar libremente sin preocupación alguna de que se origine algún tipo de choque con algún objeto, ya que lo podrá esquivar.

Las condiciones y reglas que seguirá el prototipo de silla cuando se encuentra encaminada hacia el usuario son preestablecidas mediante la programación en el software MATLAB-SIMULINK; es aquí donde se implantará toda la estrategia de giros y recorridos de los motores que el prototipo posee para su movilización, mediante los datos obtenidos por parte de los sensores de sonido y ultrasónico respectivamente, permitiendo así al prototipo el cumplimiento de su objetivo principal mencionado previamente.

Todo lo que compete a la transmisión y recepción de datos, tanto por parte de los sensores anteriormente mencionados y el software, se lo realizará mediante el uso del protocolo de comunicaciones conocido como BLUETOOTH, ya que nos facilita de gran manera la comunicación entre el equipo móvil (prototipo de silla) y el equipo fijo (PC), además de eliminar cables y conectores entre los mismos; cabe recalcar que este protocolo no posee una amplia área de cobertura, lo que nos limita a maniobrar dentro de un espacio determinado.

Los detalles del diseño y funcionamiento del prototipo serán descritos en capítulos posteriores.

The MathWorks<sup>®</sup>  
Accelerating the pace of engineering and science

MATLAB<sup>®</sup>  
& SIMULINK<sup>®</sup>



Figura 1.2 Sistemas de comunicación con el robot.

El usuario podrá gobernar dicho prototipo de silla siempre y cuando este posea un dispositivo emisor de sonido de manera imprescindible, ya que únicamente de esta manera dicho prototipo ejecutará las órdenes que fueron enviadas en la programación desde MATLAB-SIMULINK.

Debido a que el sensor de sonido que posee el prototipo de silla única y exclusivamente detecta la intensidad del sonido y no de frecuencia, es necesario operar en un área con muy bajo nivel de ruido.

### 1.1.3 Limitaciones del Proyecto

Este prototipo contiene algunas limitantes que es recomendable tener en consideración al momento de implementar el mismo, a continuación presentamos algunas de estas limitantes y sus respectivas consecuencias:

- **Campo de visión de la silla.-** El sensor ultrasónico que posee LEGO MINDSTORMS NXT, posee un campo de visión limitado ya que este puede divisar obstáculos en el lugar siempre y cuando se encuentre al nivel de este sensor, caso contrario, no lo tomará en cuenta y por lo tanto la silla avanzaría como si no hubiera nada por delante y chocaría con el objeto haciéndola atascar, así mismo sucederá con los obstáculos que estén arriba del nivel tal como objetos guindados;
- **Sonidos emitidos por el dispositivo.-** El sensor de sonido que posee la silla funciona de acuerdo a la intensidad de sonido que sensa, es por esto que el lugar en que la silla se encuentre ubicada, tiene que ser un lugar tranquilo para que el sonido que emita el dispositivo pueda ser claro. Si estamos en un lugar muy ruidoso, el sensor de sonido no puede cumplir su función de rastreo y, por consecuencia, no llegará al destino especificado.

Además, es necesario establecer el tipo de sonido que se va a emitir. Este, deberá ser de una intensidad más o menos considerable para que dicho sensor pueda captarlo y cumplir con las especificaciones estipuladas dentro de la programación. También el problema que surgiría es que sólo se lo puede usar a distancias pequeñas ya que si la distancia es muy grande el sonido se podría confundir con cualquier tipo de ruido haciendo que la silla no pueda rastrear el sonido correcto.

- **Retraso de la silla en el momento del rastreo.-** Como sabemos, la silla tiene que esperar por dos condiciones para moverse: intensidad de sonido y libertad para moverse, es decir, que la distancia que vaya a recorrer no presente obstáculo alguno, por lo cual esta necesita generar varios tipos de movimientos para poder pasar el obstáculo, lo cual puede ser hasta cierto punto monótono si la silla se encuentra con varios obstáculos en el camino para llegar al destino. Recordemos que se toman varias consideraciones para establecer los movimientos del robot, ya sea en línea recta o para los giros, y en caso de presentarse dichos retrasos, puede afectar en la exactitud de los giros, sobre todo.

## 1.2 Análisis de soluciones similares existentes en el mercado

Actualmente en el mercado existe una gama de dispositivos y productos que pueden dar gran parte de la solución a los problemas planteados anteriormente ya que ofrecen optimización en nuestro prototipo, pero también incrementan sus costo, pero pueden mejorar la exactitud y rapidez de la silla automática haciéndola más eficiente.

Las soluciones posibles a los problemas planteados son:

- **Dispositivos de recepción de señales auditivas.-** Son varios tipos de dispositivos que se encuentran en esta categoría; el más aceptado dentro de la elaboración de este proyecto sería la utilización de un sensor de frecuencias, ya que pueden ayudar a ampliar el rango de utilización de la silla, permitiendo que el usuario pueda desplazarse a mas distancias sabiendo que la silla puede venir a él, eliminando así el problema de distorsiones emitidas por el ruido exterior.
- **Sensores de rastreo.-** Son sensores que tiene un gran campo de visión, los cuales visualizan casi toda el área alrededor de ellos para ejecutar ya sea una señal o un tipo de respuesta los cuales son muy precisos al momento del muestreo y rápidos en la captación de señales.

- **Mecanismos robóticos.-** Son circuitos que acoplados mecánicamente permiten al robot realizar más tipos de movimiento con más facilidad y con el menor tiempo posible sin mayor esfuerzo, lo cual optimiza el tiempo de rastreo del robot y da mayor facilidad de esquivar obstáculos en el menor tiempo posible.

## CAPÍTULO 2

### 2 FUNDAMENTACIÓN TEÓRICA

La implementación del prototipo de silla se realizó gracias al kit LEGO MINDSTORMS el cual trataremos de explicar con la brevedad posible, para que el usuario conozca el tipo de tecnología que se usará en este proyecto, a fin de que se pueda conocer los tipos de sensores que el LEGO ofrece y de las herramientas que pueden implementarse al mismo para poder desarrollar el proyecto en cuestión.

#### 2.1 Kit Lego Mindstorms NXT

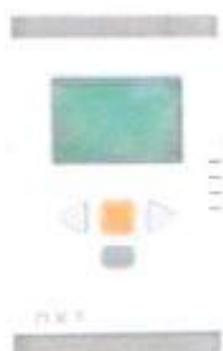


Figura 2.1 Bloque NXT

### 2.1.1 Descripción

El bloque NXT es la parte central del LEGO MINDSTORMS, ya que aquí se encuentra toda la parte lógica y electrónica que permite la mayoría de las acciones del robot, almacenándose hasta 5 programas que se pueden cargar en su memoria interna, y guardándose allí el firmware básico para el control de los distintos dispositivos que se pueden conectar al bloque.

### 2.1.2 Micro-Controlador

El micro-controlador que posee es un ARM7 de 32 bits, que incluye 256 Kb de memoria Flash y 64 Kb de RAM externa, 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. Su presentación es similar al Hitachi H8 ya que se encuentra en el circuito impreso del bloque, junto a la memoria FLASH.

### 2.1.3 Entradas y Salidas

En el bloque de NXT existen cuatro entradas para los sensores, además incluye el adaptador para que los sensores de RCX sean compatibles con NXT.

Las salidas de energía aún son tres localizadas en la parte posterior del bloque, haciendo que la conexión para los motores y partes móviles sean de más fácil acceso.

#### 2.1.4 Comunicaciones

El bloque de NXT puede comunicarse con el computador mediante la interfaz de USB que posee. Además, para comunicarse con otros robots en las cercanías posee una interfaz Bluetooth. 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.

### 2.1.5 Firmware

El firmware del LEGO MINDSTORMS consta de las instrucciones básicas que posee el bloque para hacer las distintas tareas que se le pueden programar en el bloque NXT.

Si no se carga el firmware, el robot queda en modo de arranque, lo cual hace que se pueda jugar con un programa que viene en forma nativa dentro del robot. Para cargar el firmware debe ejecutarse el programa adjunto y luego esperar cerca de 3 minutos para que se cargue completamente el firmware básico.

### 2.1.6 Motores

Los motores de la serie LEGO ROBOTICS han sido de tres tipos, los cuales son independientes al bloque, lo que entrega movilidad al sistema dinámico según las necesidades de construcción.



Figura 2.2 Motor NXT

Los motores desmontables son alimentados mediante cables que poseen conductores eléctricos que transmiten la energía a los inductores. Como son motores paso a paso, el sentido de conexión no entrega la misma dirección de movimiento.

Los motores integrados al bloque sin menos versátiles, pero no dependen de conexiones externas, lo cual visualmente ayuda al robot en su presentación.

El modelo NXT usa servo motores, los cuales permiten la detección de giros de la rueda, indicando los giros completos o medios giros, que es controlado por el software.

### **2.1.7 Piezas Especiales**

El LEGO MINDSTORMS, posee algunas piezas extras que permiten entregar flexibilidad y movimiento al robot que se esté construyendo.

Para clasificar las piezas, se sugiere una clasificación entre las piezas móviles, flexibles y de fijación, las cuales son las que incluye el LEGO MINDSTORMS para desarrollar cualquier robot en especial.



Figura 2.3 Accesorios NXT y RXT.

- **Piezas móviles:** Las piezas móviles que dispone LEGO MINDSTORMS se centran principalmente en la rotación de bloque, para lograr que las ruedas se muevan en un movimiento circular con respecto al bloque completo.
- **Piezas flexibles:** Las piezas flexibles permiten recrear una articulación de un sistema, donde se requiere que el robot deba realizar un movimiento no rígido en forma específica, como el brazo robot o el brazo clasificador de piezas. Las piezas flexibles por lo general son tubos de plástico capaces de conectarse con dos bloques que no se encuentren separados a una distancia mayor de 4 cm.
- **Piezas de fijación:** Las piezas de fijación, son aquellas que sirven para fijar los ejes de rotación producidos por las piezas de rotación, lo cual implica que son usadas en el centro de las ruedas que posee el Lego. Por lo general, son

tubos de 0.5 mm de diámetro el cual se puede poner en la punta de una barra que actúa como eje central de la rueda, fijando que la misma no se salga durante la ejecución de un programa.

### 2.1.8 Ruedas

Uno de los principales componentes de LEGO MINDSTORMS, y que le da mayor dinamismo a la construcción de robots son las ruedas, ya que permiten que el bloque lógico pueda moverse en un espacio real e interactúe con el medio que lo rodea.

Existen distintas versiones de ruedas, que vienen desde las llantas más anchas, que permiten mayor estabilidad y velocidad; hasta las ruedas más pequeñas que permiten el movimiento del robot en zonas más pequeñas. Se incluye además una cinta que simula el efecto de oruga que poseen los tanques, para que el usuario pueda crear tanques, o una cinta transportadora de objetos.



Figura 2.4 Ruedas NXT y RXT.

### 2.1.9 Alimentación Eléctrica

La alimentación eléctrica del bloque es mediante 6 baterías AA de 1,5 volts; las cuales se conectan en la parte posterior del bloque. Las baterías se conectan en paralelo y proporcionan energía tanto al bloque como a los motores que se conectan al mismo bloque.

Una advertencia que se realiza en el manual de LEGO MINDSTORMS, es si el reemplazo demora más de 1 minuto, la información almacenada se pierde.

Se puede programar la desconexión de la alimentación eléctrica mediante el software incluido en el Kit del LEGO MINDSTORMS, donde se puede especificar el tiempo de apagado, desde 1 minuto hasta 99 minutos, e inclusive deshabilitar el apagado automático. Además, el mismo programa indica el nivel de carga que poseen las baterías del robot.

### 2.1.10 Sensores de Lego Mindstorms NXT

Un sensor es un dispositivo formado por células sensibles que detecta variaciones en una magnitud física (luz, temperatura, sonido, tamaño, velocidad, color, etc.) u otras alteraciones de su entorno, y las convierte en señales útiles para un sistema de medida o control.

Pueden ser de indicación directa (Ej. un termómetro de mercurio) o pueden estar conectados a un indicador (posiblemente a través de un convertidor analógico a digital, un computador y un display) de modo que los valores obtenidos puedan ser leídos por un humano.

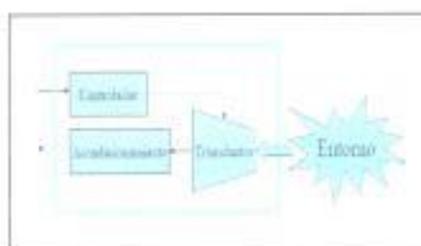


Figura 2.5 Esquema interno básico de un sensor.

- **Sensor Ultrasonico:** El sensor Ultrasonico tiene como función principal detectar las distancias y el movimiento de un objeto que se interponga en el camino del robot, mediante el principio de la detección ultrasonica. Este sensor es capaz de detectar objetos que se encuentren desde 0 a 255 cm, con una precisión relativa de  $\pm 3$  cm.



Figura 2.6 Sensor Ultrasonico NXT.

- **Sensor de sonido:** El sensor de sonido solo detecta la "cantidad" de sonido y no ningún tipo de tono o modulación, pero aun así hay muchas aplicaciones ingeniosas que se le pueden dar. Este sensor lee el sonido ambiental y nos regresa una medida de 0 a 100%. Podemos configurarlo para que lea Decibeles ajustados. En términos muy simples los decibeles ajustados solo incluye sonidos que el oído humano puede escuchar, al contrario de los decibeles normales que podría incluir frecuencias que no podemos escuchar pero que el sensor de sonido capta.



Figura 2.7 Sensor de sonido NXT.

- **Sensor de Contacto:** 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. En este caso, si la presión supera una medida estándar de 450, mostrado en la

pantalla de LCD, se considera que el sensor está presionado, de otro modo, se considera que está sin presión.



Figura 2.8 Sensor de tacto NXT.

- **Sensor de Luz:** Permite tomar una muestra de luz mediante un bloque modificado que un extremo trae un conductor eléctrico y por el otro una cámara oscura que capta las luces. Esta cámara es capaz de captar luces entre los rangos de 0,6 a 760 lux. Este valor lo considera como un porcentaje, el cual es procesado por el bloque lógico, obteniendo un porcentaje aproximado de luminosidad. Debido a que este sensor capta grados de luminosidad, no es capaz de distinguir colores, sólo captando la existencia del blanco (claridad), negro (oscuridad) y los tonos de grises que corresponden a los distintos porcentajes de luz existentes en el medio.



Figura 2.9 Sensor de luz NXT.

## **2.2 MATLAB**

### **2.2.1 Introducción**

MATLAB es una técnica de alto nivel de lenguaje de computación y de entorno interactivo para desarrollo de algoritmos, visualización de datos, análisis de datos y cálculo numérico. El uso del producto MATLAB, puede resolver problemas de computación técnica más rápida que con los lenguajes de programación tradicionales.

MATLAB posee una amplia gama de aplicaciones, incluyendo señales y procesamiento de imágenes, comunicaciones, diseño de control, de prueba y edición.

Además el entorno de MATLAB puede ampliar herramientas para resolver determinadas clases de problemas en distintas áreas de aplicaciones.

### **2.2.2 Herramientas de MATLAB para LEGO MINDSTORMS NXT**

**RWTH - Mindstorms NXT herramientas para MATLAB.** Estas herramientas han sido desarrolladas para el control de LEGO MINDSTORMS robots NXT con MATLAB a través de una conexión inalámbrica Bluetooth o vía USB.



**Figura 2.10** Interacción NXT – MATLAB.

Las funciones de las herramientas se basan en el LEGO MINDSTORMS NXT Bluetooth Protocolo de Comunicación para controlar el NXT ladrillo inteligente a través de una conexión inalámbrica Bluetooth o vía USB.

La principal ventaja de este concepto de control remoto le permite combinar las aplicaciones de robots con complejas operaciones matemáticas y visualización en MATLAB.

Estas herramientas ofrecen posibilidades ilimitadas para proporcionar a los robots de inteligencia artificial y otras mejoras, utilizar las funciones de MATLAB múltiples y cálculos para el procesamiento de señales digitales.

## 2.3 Simulink

Simulink es una plataforma para simulación multidominio y diseño basado en modelos de sistemas dinámicos. Proporciona un entorno gráfico interactivo y un conjunto de librerías de bloques personalizables que permiten diseñar, simular, implementar y probar una gran variedad de sistemas con variación temporal, entre los que se incluyen sistemas de comunicaciones, control, procesamiento de señales, vídeo e imagen.

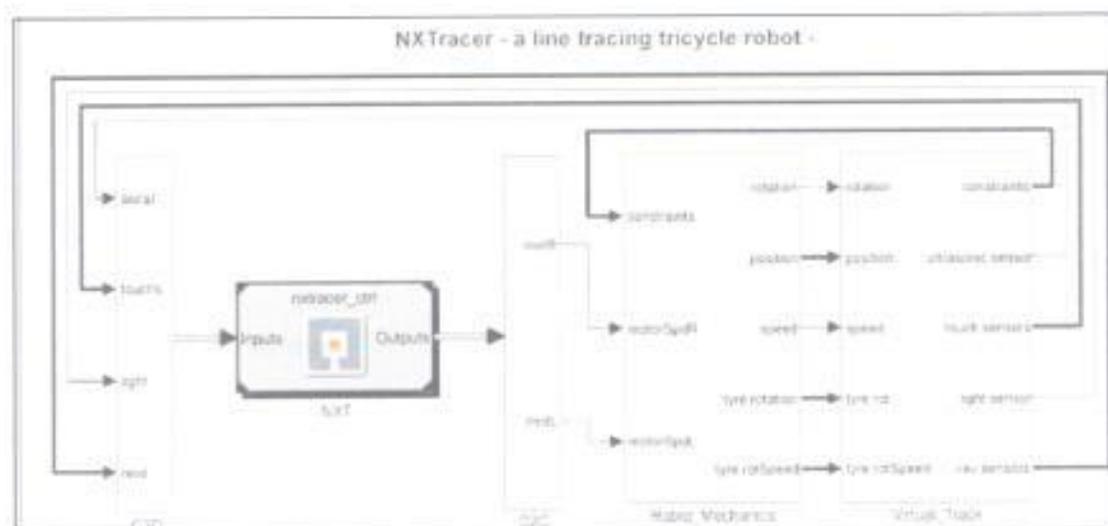


Figura 2.11 Ejemplo de programación NXT en Simulink.

## 2.4 Bluetooth

El LEGO MINDSTORMS NXT incluye un nodo inalámbrico Bluetooth que permite que el bloque NXT pueda comunicarse con otros dispositivos.



**Figura 2.12** Comunicación bluetooth

Bluetooth es una tecnología que permite enviar y recibir datos sin necesidad de utilizar cables. El uso de Bluetooth, puede intercambiar programas entre el NXT y otros NXT. Usted puede establecer una conexión inalámbrica entre su ordenador y su robot.

## CAPÍTULO 3

### 3 DISEÑO DEL SOFTWARE Y HARDWARE DEL PROTOTIPO DE SILLA

#### 3.1 Diagrama de bloques del diseño propuesto

Como se explicó en el capítulo 1, el objetivo principal de este proyecto es el desarrollo de un prototipo de silla, capaz de responder a los requerimientos del usuario por medio de la intensidad con que sean emitidos los sonidos del dispositivo emisor de audio, indistintamente de la distancia que esta se encuentre ubicada, siendo capaz de llegar al mismo a pesar de los obstáculos que se le presenten durante su recorrido.

En este capítulo describiremos en detalle el diseño del proyecto en general, tanto de del software y hardware utilizado para alcanzar los objetivos propuestos.

Para empezar, se describirá el diseño del hardware utilizado, luego del cual se procederá a explicar con detalles el funcionamiento del prototipo de silla para entender en su totalidad el comportamiento del sistema tratado.

El hardware del sistema está conformado por tres bloques, los mismos que interactúan entre sí para permitir a la silla ejecutar movimientos necesarios para el cumplimiento de su objetivo, los cuales se presentarán de manera comprensible.



**Figura 3.1** Diagrama del Diseño Propuesto.

El primer bloque, está conformado por un PC que, junto con el software MATLAB & SIMULINK, establecerá la programación explicada posteriormente para que la silla pueda lograr el objetivo requerido en cuestión.

Necesitamos, en primer lugar, acondicionar el software MATLAB & SIMULINK, ya que una vez instalado en sí no posee las herramientas necesarias para poder establecer correcto funcionamiento de la silla. Es por esto que se recurre a la instalación de varias librerías adicionales para poder enlazar el software MATLAB con la silla.

Inicialmente, necesitamos instalar el CYGWIN que se lo puede obtener de la página <http://www.cygwin.com/>, el cual permitirá que MATLAB almacene todos los datos que se procederán a instalar a continuación.

Nombre	Tamaño	Tipo	Fecha de modificación
http%3a%2f%2fcygwin.elite...		Carpeta de archivos	30/07/2009 18:05
setup	628 KB	Aplicación	19/07/2009 14:11
setup	13 KB	Documento de Write	23/07/2009 20:40
setup-1.7	593 KB	Aplicación	19/07/2009 14:08
setup.log.full	27 KB	Archivo FULL	23/07/2009 20:40

figura 3.2 Componentes de la instalación del cygwin.

Luego ejecutamos el archivo cygwin.exe



Figura 3.3 Ventana del cygwin.exe.

Una vez iniciado el proceso de instalación del cygwin, es necesario establecer las rutas en las cuales vamos a colocar los archivos que son componentes adicionales que necesita MATLAB para establecer la compatibilidad con el LEGO MINDSTORMS NXT.



Figura 3.4 Instalador Cygwin.

Es muy importante instalarlo en modo normal ya que se instalarán los archivos necesarios que necesita MATLAB; marcar la opción Unix / binary, la cual, permite que todos los archivos que se van a ejecutar se realicen de una forma binaria estableciendo así una compatibilidad en los archivos MATLAB con los del LEGO MINDSTORMS NXT.



Figura 3.5 Instalando el Cygwin.

Marcamos en Devel Default los último casilleros del 3.81 y procedemos con la instalación

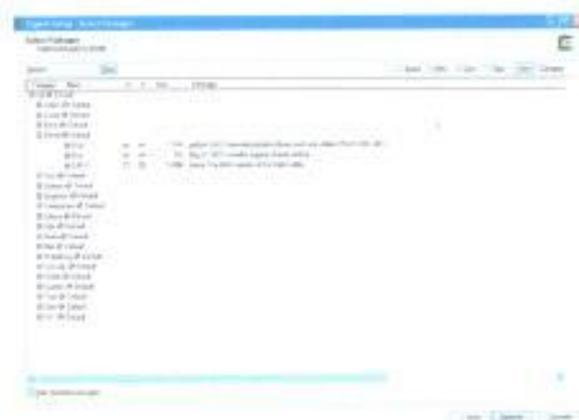


Figura 3.6 Cygwin en modo default

Luego del cual instalamos finalmente el cygwin.

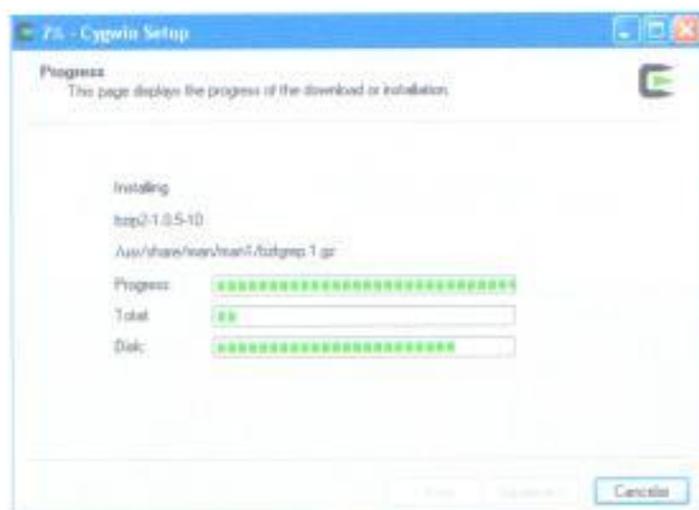


Figura 3.7 Cygwin en modo default



Figura 3.8 Cygwin instalado

Una vez instalado el cygwin, se comprobará su correcta instalación mediante el uso del mismo, escribiendo “which make” y el programa nos debe responder “/usr/bin/make”.

También se los puede comprobar con “ls” y tiene que mostrar “/usr/bin/make”.



Figura 3.9 Demostración del Cygwin correctamente instalado

Una vez instalado el cygwin, necesitamos instalar la segunda herramienta necesaria que se denomina GNU ARM el cual permitirá al Matlab poder establecer la librería que requerimos, la cual al ser instalada, necesariamente tiene que ser diseccionada en la carpeta cygwin con la siguiente ruta C:/cywing/GNUARM.

Ejecutamos GNUARM.exe

Nombre	Tamaño	Tipo	Fecha de modificación
bu-2.16.1_gcc-4.0.2-c-c++_...	25.453 KB	Aplicación	19/07/2009 15:00

**Figura 3.10** Instalador GNUARM



**Figura 3.11** Ejecutando GNUARM . exe.

Luego necesitamos colocar el GNUARM en la carpeta del cygwin.



Figura 3.12 Instalador GNUARM en cygwin

Aparecerá un cuadro de diálogo con los componentes que necesitamos instalar, desmarcamos los DLLs ya que fueron instalados previamente por el cygwin.

Luego, procederemos a desmarcar los cuatro últimos cuadros de la librería Little Indian y desmarcamos toda la librería Big Indian; a continuación debemos chequear un mensaje que se solicita 161.0 MB de espacio mínimo en el disco duro, hacemos clic en aceptar.

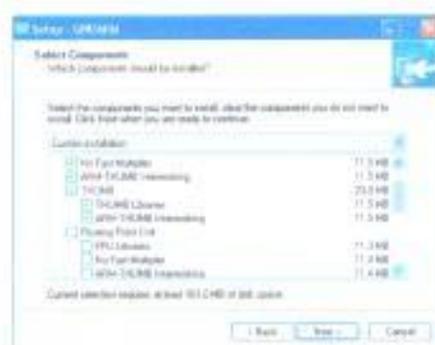


Figura 3.13 Librerías instaladas en el GNUARM.

Luego instalamos el GNUARM

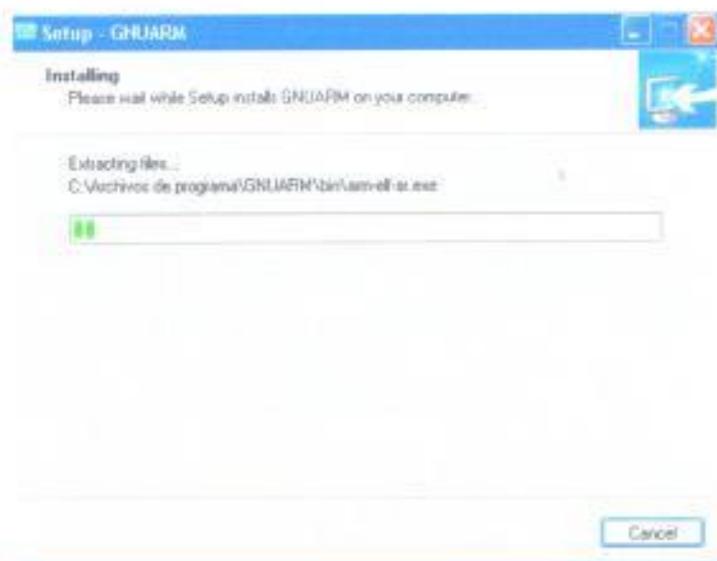


Figura 3.14 Instalando GNUARM.



Figura 3.15 GNUARM Instalado

Al finalizar la instalación, el programa me pedirá redireccionar los datos instalados lo cual no debemos hacer. Luego tendremos instalado la segunda herramienta para el MATLAB.

Luego continuamos con la instalación de la tercera herramienta, NeXTTool, la cual se la puede descargar de portal web:

<http://briexec.sourceforge.net/utilities.html>

Nombre	Tamaño	Tipo	Fecha de modificación
 nexttool	6 KB	Documento de texto	12/04/2007 13:30
 NeXTTool	159 KB	Aplicación	30/07/2007 13:29

**Figura 3.16** Archivos NextTool

Posteriormente se descomprime el archivo lms\_arm\_jch.zip

Nombre	Tamaño	Tipo	Fecha de modificación
 lms_arm_nbcnxc_107.rfw	256 KB	Archivo RFW	01/01/2009 17:30
 LMS_ARM_NBCNXC_105.a79	125 KB	Archivo A79	22/07/2008 10:39
 lms_arm_nbcnxc_105.rfw	256 KB	Archivo RFW	22/07/2008 10:39
 LMS_ARM_NBCNXC_106.a79	134 KB	Archivo A79	22/07/2008 10:42
 lms_arm_nbcnxc_106.rfw	256 KB	Archivo RFW	22/07/2008 10:43
 LMS_ARM_NBCNXC_107.a79	133 KB	Archivo A79	01/01/2009 17:30

**Figura 3.17** Archivos en el lms\_arm\_jch.zip.

Estos archivos los colocamos dentro de la carpeta NeXTTool

Nombre	Tamaño	Tipo	Fecha de modificación
LMS_ARM_NBCNXC_105.a79	125 KB	Archivo A79	22/07/2008 10:39
lms_arm_nbcnxc_105.rfw	256 KB	Archivo RFW	22/07/2008 10:39
LMS_ARM_NBCNXC_106.a79	134 KB	Archivo A79	22/07/2008 10:42
lms_arm_nbcnxc_106.rfw	256 KB	Archivo RFW	22/07/2008 10:43
LMS_ARM_NBCNXC_107.a79	133 KB	Archivo A79	01/01/2009 17:30
lms_arm_nbcnxc_107.rfw	256 KB	Archivo RFW	01/01/2009 17:30
nexttool	6 KB	Documento de texto	12/04/2007 13:30
NeXTTool	159 KB	Aplicación	30/07/2007 13:29

Figura 3.18 Archivos juntos al NeXTTool.

Luego de esto comprobamos si se ha realizado la instalación con éxito el cmd.

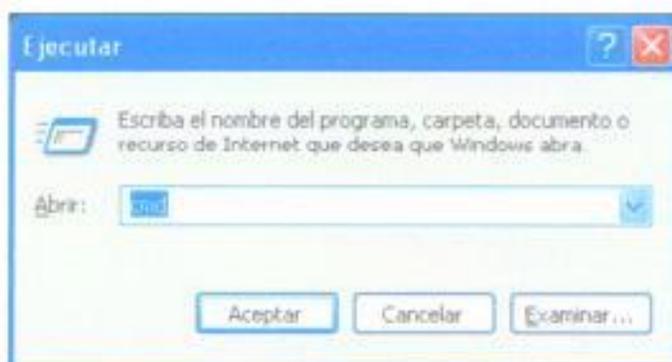
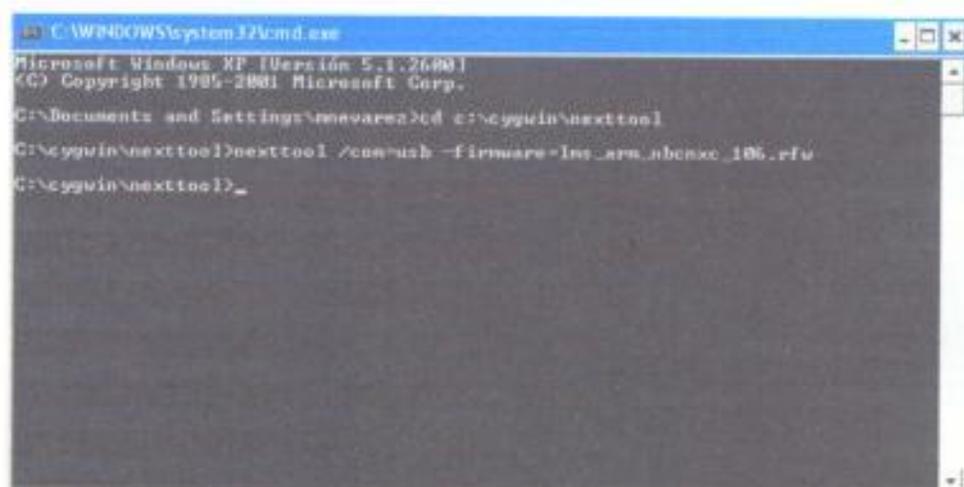


Figura 3.19 Cmd de Windows

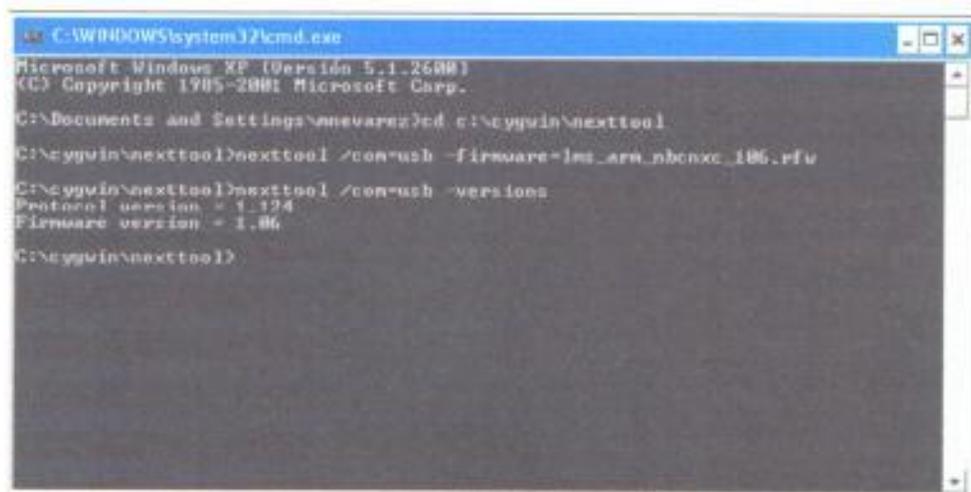


```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\mevare2>cd c:\cygwin\nexttool
C:\cygwin\nexttool>nexttool /con=usb -firmware=Imc_arm_abcncx_106.rfu
C:\cygwin\nexttool>
```

Figura 3.20 Cmd de Windows con el NextTool

Para verificarlo del mismo necesitamos ejecutar los siguientes comandos mostrados para observar la versión instalada y su protocolo.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\mevare2>cd c:\cygwin\nexttool
C:\cygwin\nexttool>nexttool /con=usb -firmware=Imc_arm_abcncx_106.rfu
C:\cygwin\nexttool>nexttool /con=usb -version
Protocol version = 1.124
Firmware version = 1.06
C:\cygwin\nexttool>
```

Figura 3.21 Comprobación del programa instalados.

Nuestro siguiente paso es descomprimir el `nxtOSEK` en el que se encuentran los siguientes archivos

Nombre	Tamaño	Tipo	Fecha de modificación
bin		Carpeta de archivos	29/04/2009 10:02
c++		Carpeta de archivos	29/04/2009 10:02
ecrobot		Carpeta de archivos	01/05/2009 08:35
lejos_nxj		Carpeta de archivos	29/04/2009 10:02
samples_c		Carpeta de archivos	29/04/2009 10:02
samples_c++		Carpeta de archivos	29/04/2009 10:02
samples_jsp		Carpeta de archivos	29/04/2009 10:02
toppers_jsp		Carpeta de archivos	29/04/2009 10:02
toppers_osek		Carpeta de archivos	29/04/2009 10:02
README	8 KB	Documento de texto	01/05/2009 09:29

Figura 3.22 Archivos contenidos en `nxtOSEK`

Una vez instalados los archivos anteriores procedemos con la instalación de la librería USB.

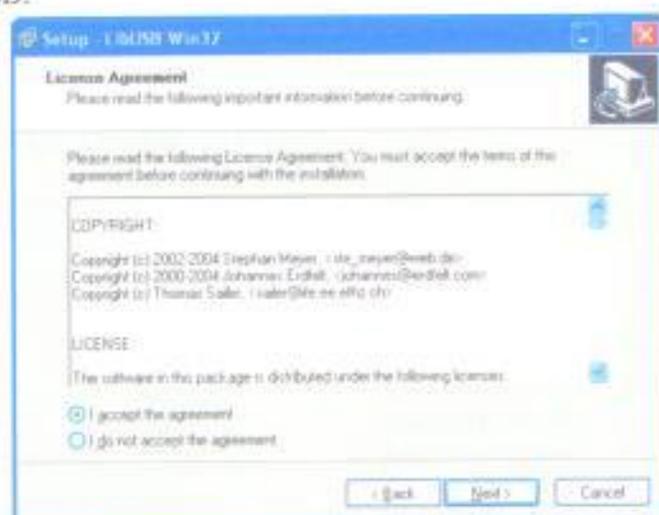


Figura 3.23 Instalación de librería USB

El cual nos tiene que mostrar la siguiente pantalla.



Figura 3.24 Comprobando la librería USB

Luego de esto se procede a la colocación de los archivos de la carpeta e robot n xt el cual tiene los siguientes archivos:

Nombre	Tamaño	Tipo	Fecha de modificación
docs		Carpeta de archivos	10/03/2009 13:55
environment		Carpeta de archivos	10/03/2009 11:26
nxt2nxt		Carpeta de archivos	28/12/2008 14:00
nxtmouse		Carpeta de archivos	28/12/2008 14:00
nxttracer		Carpeta de archivos	28/12/2008 14:37
nxtremocon		Carpeta de archivos	28/12/2008 14:38
nxtway		Carpeta de archivos	28/12/2008 14:38
samples		Carpeta de archivos	17/03/2009 18:13
ECRobot Installation Instructi...	518 KB	Adobe Acrobat 7.0 ...	10/03/2009 14:03
ecrobotnxtsetup.m	8 KB	Archivo M	24/03/2009 10:08

Figura 3.25 Archivos de e robotNXT

Estos archivos son ubicados dentro de la carpeta MATLAB

Nombre	Tamaño	Tipo	Fecha de modificación
bin		Carpeta de archivos	26/06/2009 3:54
ecrobotNXT		Carpeta de archivos	26/06/2009 6:01
extern		Carpeta de archivos	26/06/2009 3:41
help		Carpeta de archivos	26/06/2009 3:41
ia		Carpeta de archivos	26/06/2009 3:41
java		Carpeta de archivos	26/06/2009 3:41
help		Carpeta de archivos	26/06/2009 3:42
licenses		Carpeta de archivos	26/06/2009 4:17
notebook		Carpeta de archivos	26/06/2009 3:42
rtw		Carpeta de archivos	26/06/2009 3:45
simulink		Carpeta de archivos	26/06/2009 3:45
stateflow		Carpeta de archivos	26/06/2009 3:45
sys		Carpeta de archivos	26/06/2009 3:56
toolbox		Carpeta de archivos	26/06/2009 4:10
uninstall		Carpeta de archivos	26/06/2009 4:12
work		Carpeta de archivos	26/06/2009 4:11
licenses	73 KB	Documento de texto	23/01/2008 22:58
MATLAB R2009a	1 KB	Acceso directo	26/06/2009 4:12
patents	1 KB	Documento de texto	23/01/2008 22:21
trademarks	1 KB	Documento de texto	26/12/2007 16:02
licencia	2.616 KB	Reserva	02/09/2009 22:13

Figura 3.26 EcrobotNXT ubicado en MATLAB

Ahora ya tenemos listo a MATLAB para poder ejecutar los archivos NXT, ahora podemos abrir MATLAB para poder realizar nuestro proyecto en cuestión y nos colocamos en la ventana de trabajo



Figura 3.27 Ejecución de MATLAB

Luego de ingresar en MATLAB, procedemos a redireccionar los archivos que ubicamos anteriormente, ejecutando un programa que nos permite asignar las carpetas directamente llamado **ecrobotxtsetup.m**

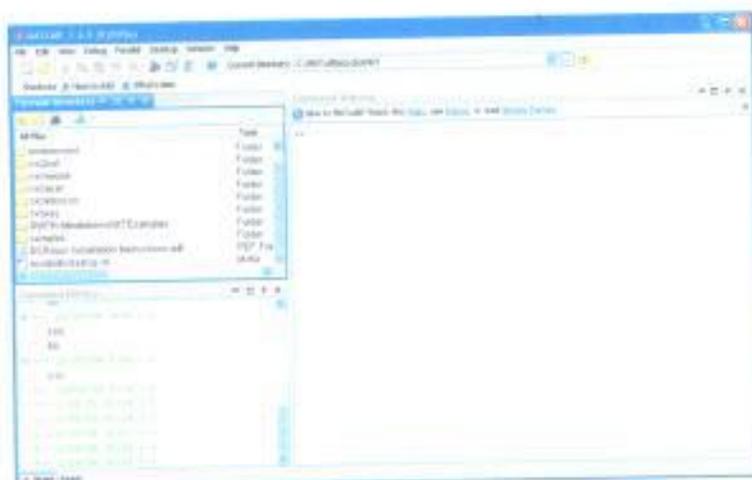


Figura 3.28 Ubicación de ecrobotxtsetup.m en Matlab

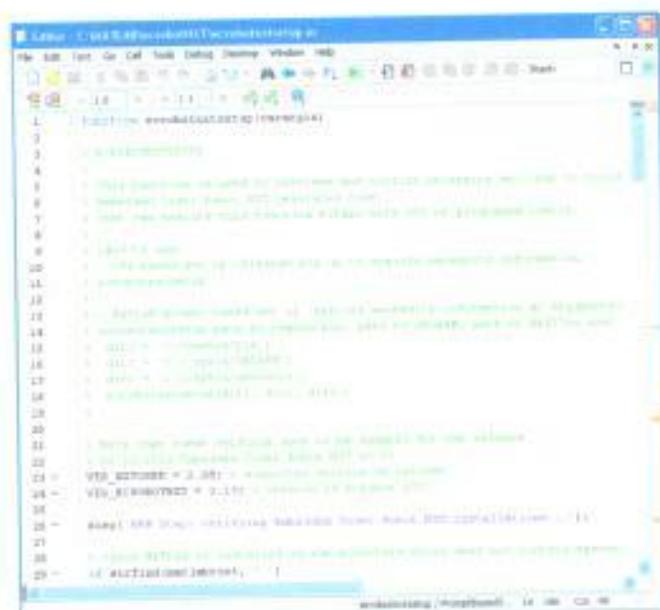


Figura 3.29 ecrobotxtsetup.m

El `ecrobotsetupnxt.m` le permite a MATLAB direccionar los archivos instalados de una manera definitiva. Luego de ejecutar el programa automáticamente pedirá al usuario que direccione manualmente las carpetas de los programas instalados: Cygwin/bin, GNUARM, y el `nexttool`.

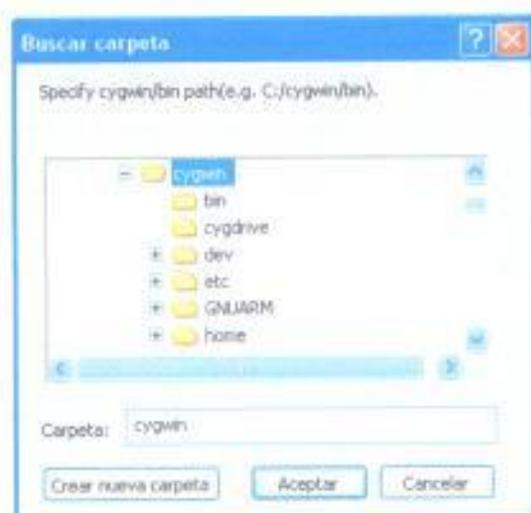
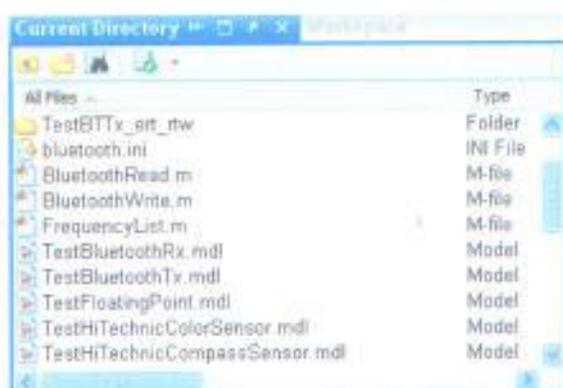


Figura 3.30 Ubicación de archivos mediante `ecrobotnxtsetup.m`

Luego de esto ya tenemos enlazado a MATLAB con el LEGO MINDSTORMS para poder realizar la programación del proyecto en cuestión.

Una vez seguidos estos pasos se procede a la comprobación de los datos en MATLAB. Para su comprobación vamos a la ventana de carpetas en el workspace y buscamos la carpeta "samples"



**Figura 3.31** Ejemplos de programas elaborados en MATLAB

Luego de esto podemos ver varios ejemplos elaborados en MATLAB & SIMULINK para poder ejecutarlos y así podemos desarrollar el programa para el prototipo de silla inteligente.

El segundo bloque, el dispositivo bluetooth, consiste en un protocolo de transmisión para enviar la programación detallada anteriormente desde MATLAB hacia el prototipo de silla.

La configuración de dicho dispositivo, será de la siguiente manera:

- Conectar el dispositivo al PC y la instalación del software que este posee denominado BLUESOLEIL 2.0
- Luego del cual aparecerá la siguiente pantalla:

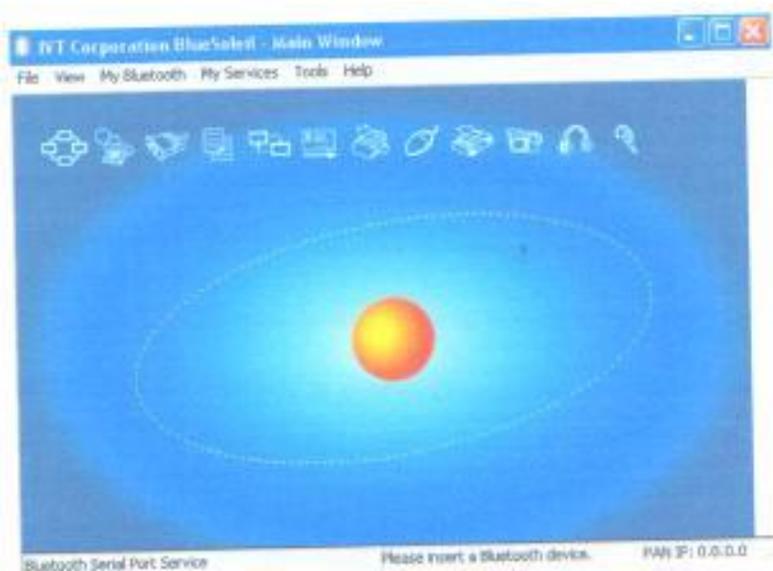


Figura 3.32 Software BlueSoleil 2.0

- Luego de esto necesitamos establecer la conexión entre el PC y el bloque NXT.

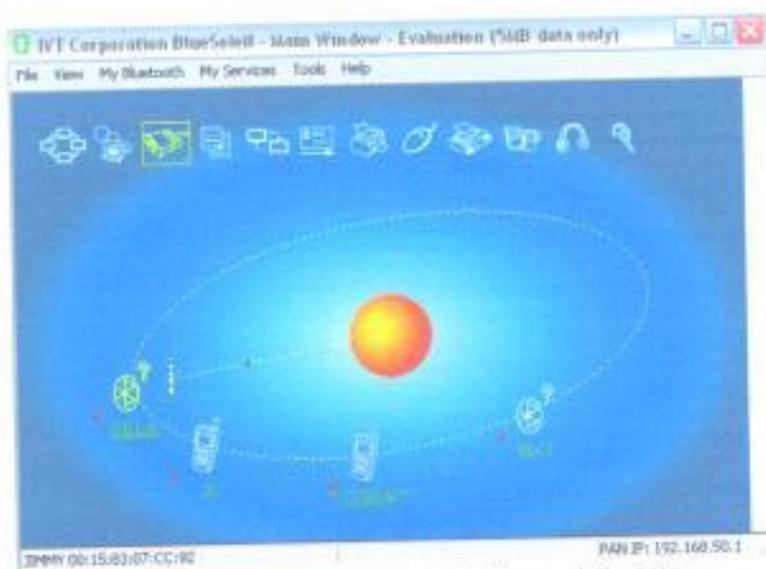


Figura 3.33 Conexión establecida entre PC y NXT.

Una vez realizado esto procedemos a enviar el programa elaborado en MATLAB hacia el bloque NXT para así tener lista nuestra silla para su funcionamiento.

Nuestro último bloque es el prototipo de silla con su sensor ultrasónico y de sonido, los cuales funcionarán de acuerdo a la programación realizada.

El ensamblado con el LEGO de la silla es el siguiente.



**Figura 3.34** Prototipo de silla inteligente

El sensor ultrasónico estará a cargo de las condiciones de distancia para que la silla pueda llegar al usuario de una manera óptima.



**Figura 3.35** Sensor ultrasónico

El sensor de sonido realizará la tarea de búsqueda del dispositivo de sonido, para encaminar la silla al lugar requerido.



Figura 3.36 Sensor de sonido.

### 3.2 Proyecto de prueba previo a la implementación del prototipo de silla

Este proyecto fue elaborado mediante el uso del kit LEGO MINDSTORMS NXT. Mediante este proyecto, se demostrará el funcionamiento de algunos sensores que incluye el kit de LEGO para familiarizarse tanto con el equipo (hardware) y su programación (software).

Los escorpiones o alacranes son un orden de arácnidos con los pedipalpos en forma de pinza y un aguijón venenoso en el extremo del cuerpo. Por lo general, son seres insociables que pocas veces entran en contacto con otros de su misma especie.

Basándonos en este principio de comportamiento, hemos diseñado nuestro robot para que responda de la misma manera, es decir, simulando el comportamiento normal que tendría un escorpión.



**Figura 3.37** Escorpión NXT.

El robot fue construido mediante el uso de 3 sensores: el ultrasónico, el de tacto y el de sonido.

Su funcionamiento es el siguiente:

Mediante el uso del sensor ultrasónico, el escorpión tendrá la capacidad de "observar" objetos que se encuentren frente a él. Podrá distinguir objetos y realizar una acción dependiendo de la distancia que éste se encuentre. Está programado para distinguir objetos en 3 rangos de distancias: mayores a 50cm, entre 50cm y 15cm, y menores a 15cm. Si detecta algún objeto a distancias mayores a 50 cm, avanzará normalmente hacia adelante; entre 50cm y 15cm,

retrocederá rápidamente por un instante; y a distancias menores a 15cm, procederá a "picar".

El sensor de tacto se encuentra ubicado al final de la cola, como especie de aguijón venenoso; su función será detectar el instante en que algún objeto o persona fue "picado" por el escorpión. Luego de haber realizado su picadura, como gesto de burla, se reirá de la acción que realizó.

El sensor de sonido tiene una particularidad; en esta ocasión fue usado para detectar sonidos fuertes, haciendo que el escorpión se "asuste", demostrando como reacción una picadura instantánea.



Figura 3.38 Escorpión NXT en funcionamiento.

Básicamente, la programación consta de un lazo infinito, y dentro de dicho lazo se encuentran los switches para el sensor ultrasónico y el de sonido.

Inicialmente pregunta, en el switch ultrasónico, si la distancia que está el objeto en frente es mayor a 50cm. En caso de ser cierto, entra al switch del sonido, en el

cual pregunta cuál es la intensidad del sonido. En caso de ser fuerte, los motores que controlan el movimiento del arácnido pararán, y el motor que controla la cola con el aguijón se activará, esperando a que el sensor de tacto sea presionado para luego regresar a su posición original; luego de esto procederá a reírse.

Ahora, en caso de que el objeto se encuentre menor a 50cm, entra al siguiente switch ultrasónico, el cual pregunta si el objeto se encuentra a una distancia mayor a 15cm. En caso de ser correcto, los motores que mueven el arácnido girarán en sentido contrario por 2 segundos, haciendo que el escorpión retroceda como medida de prevención. En caso de que el objeto se encuentre a una distancia menor a 15cm, procederá con la acción de igual manera cuando la intensidad del sonido era fuerte, es decir, los motores que controlan el movimiento del arácnido pararán, y el motor de la cola con el aguijón se activará, esperando a que el sensor de tacto sea presionado para luego regresar a su posición original; luego de esto procederá a reírse.

Y así continúa el ciclo de manera infinita.



Figura 3.39 Programación del escorpión en NXT-G

### 3.3 Análisis del código de programación en MATLAB

Explicación del código por partes y al final un resumen del mismo

```
%INICIANDO PROGRAMA
```

```
disp('INICIANDO PROGRAMA...');
COM_CloseNXT all
clear all
close all
```

```
%CREANDO CONSTANTES
```

```
disp('CREANDO BASE DE DATOS...');
S = SENSOR_1;
TACTO = SENSOR_2;
US= SENSOR_4;
tiempomax = 2.5;
distmax = 31;
giro90 = 280;
giro180 = 1.35*giro90;
flag = 0;
prev = 400;
dir = 1;
```

```
i = 1;
matriz = 0;
```

#### %INICIANDO COMUNICACION CON NXT

```
disp('INICIANDO COMUNICACION CON PROTOTIPO...');
handle = COM_OpenNXT('bluetooth.ini','check');
COM_SetDefaultNXT(handle);
StopMotor('all','off');
ResetMotorAngle(MOTOR_B);
ResetMotorAngle(MOTOR_C);
OpenUltrasonic(US);
OpenSwitch(TACTO);
disp('INICIO COMPLETADO!');
```

#### %LAZO INFINITO

```
while true
    sonido = 0;
    derecha = 0;
    izquierda = 0;
```

```
% flag = 0
```

#### % INDICA OPERACION DE GIRO CON OPERACION DE

#### % SENSOR DE INTENSIDAD DE SONIDO

```
while flag == 0
    SetMotor(MOTOR_B);
    SyncToMotor(MOTOR_C);
    SetPower(23);
    SetAngleLimit(0);
    SetTurnRatio(dir*100);
    SendMotorSettings();
    pause(2);

    OpenSound(S,'DBA');
    while true
        if GetSound(S) > prev
            sonido = round((GetSound(S)/1023)*100);
            break
        end
        if GetSwitch(TACTO) == 1
```

```

        sonido = 0;
        derecha = 0;
        izquierda = 0;
        break
    end
end
end
CloseSensor(S);
StopMotor('all','off');
ResetMotorAngle(MOTOR_B);
ResetMotorAngle(MOTOR_C);
flag = 1;
end

if GetSwitch(TACTO) == 1
    sonido = 0;
    derecha = 0;
    izquierda = 0;
    break
end

% flag = 1
% INDICA OPERACION DE SENSOR ULTRASONICO
while flag == 1
    SetMotor(MOTOR_B);
    SyncToMotor(MOTOR_C);
    SetPower(70);
    SetAngleLimit(0);
    SetTurnRatio(0);
    SendMotorSettings();

    tictic(1);
    while (toctoc(1) < tiempomax)
        if GetUltrasonic(US) < distmax
            NXT_PlayTone(440, 1000);
            flag = 2;
            break
        end
    end
    flag = 0;
end
StopMotor('all','off');
```

```

    ResetMotorAngle(MOTOR_B);
    ResetMotorAngle(MOTOR_C);
end

if GetSwitch(TACTO) == 1
    sonido = 0;
    derecha = 0;
    izquierda = 0;
    break
end

% flag = 2
% OPERACION EN CASO DE PRESENCIA DE OBSTACULOS
while flag == 2
    SetMotor(MOTOR_B);
    SyncToMotor(MOTOR_C);
    SetPower(35);
    SetAngleLimit(giro90);
    SetTurnRatio(100);
    SendMotorSettings();
    pause(1)
    WaitForMotor(GetMotor)
    pause(1)
    StopMotor('all','off');
    ResetMotorAngle(MOTOR_B);
    ResetMotorAngle(MOTOR_C);
    derecha = GetUltrasonic(US);

    SetMotor(MOTOR_B);
    SyncToMotor(MOTOR_C);
    SetPower(35);
    SetAngleLimit(giro180);
    SetTurnRatio(-100);
    SendMotorSettings();
    pause(1)
    WaitForMotor(GetMotor)
    pause(1)
    StopMotor('all','off');
    ResetMotorAngle(MOTOR_B);
    ResetMotorAngle(MOTOR_C);

```

```

izquierda = GetUltrasonic(US);

if derecha > izquierda
    SetMotor(MOTOR_B);
    SyncToMotor(MOTOR_C);
    SetPower(35);
    SetAngleLimit(giro180);
    SetTurnRatio(100);
    SendMotorSettings();
    pause(1)
    WaitForMotor(GetMotor)
    pause(1)
    StopMotor('all','off');
    ResetMotorAngle(MOTOR_B);
    ResetMotorAngle(MOTOR_C);
    flag = 3;
else
    flag = 4;
end
end

if GetSwitch(TACTO) == 1
    sonido = 0;
    derecha = 0;
    izquierda = 0;
    break
end

% flag = 3
% INDICA QUE SE DIRIGE HACIA LA DERECHA Y
% SIRVE PARA RECORDAR QUE LUEGO DEBERÁ GIRAR HACIA
% LA IZQUIERDA PARA SEGUIR EL CAMINO PREVIAMENTE
% ESTABLECIDO

while flag == 3
    SetMotor(MOTOR_B);
    SyncToMotor(MOTOR_C);
    SetPower(70);
    SetAngleLimit(0);
    SetTurnRatio(0);

```

```

SendMotorSettings();

tic(2);
while toctoc(2) < 1
    if GetUltrasonic(US) < distmax
        break
    end
end

StopMotor('all','off');
ResetMotorAngle(MOTOR_B);
ResetMotorAngle(MOTOR_C);

SetMotor(MOTOR_B);
SyncToMotor(MOTOR_C);
SetPower(35);
SetAngleLimit(giro90-20);
SetTurnRatio(-100);
SendMotorSettings();
pause(1)
WaitForMotor(GetMotor)
pause(1)
StopMotor('all','off');
ResetMotorAngle(MOTOR_B);
ResetMotorAngle(MOTOR_C);

if GetUltrasonic(US) > distmax
    flag = 0;
    break
else
    SetMotor(MOTOR_B);
    SyncToMotor(MOTOR_C);
    SetPower(35);
    SetAngleLimit(giro90-20);
    SetTurnRatio(100);
    SendMotorSettings();
    pause(1)
    WaitForMotor(GetMotor)
    pause(1)
    StopMotor('all','off');

```

```

    ResetMotorAngle(MOTOR_B);
    ResetMotorAngle(MOTOR_C);

```

```
end
```

```
end
```

```
if GetSwitch(TACTO) == 1
```

```
    sonido = 0;
```

```
    derecha = 0;
```

```
    izquierda = 0;
```

```
    break
```

```
end
```

```
% flag = 4
```

```
% INDICA QUE SE DIRIGE HACIA LA IZQUIERDA Y
```

```
% SIRVE PARA RECORDAR QUE LUEGO DEBERÁ GIRAR HACIA
```

```
% LA DERECHA PARA SEGUIR EL CAMINO PREVIAMENTE
```

```
% ESTABLECIDO
```

```
while flag == 4
```

```
    SetMotor(MOTOR_B);
```

```
    SyncToMotor(MOTOR_C);
```

```
    SetPower(70);
```

```
    SetAngleLimit(0);
```

```
    SetTurnRatio(0);
```

```
    SendMotorSettings();
```

```
    tictic(3);
```

```
    while toctoc(3) < 1
```

```
        if GetUltrasonic(US) < distmax
```

```
            break
```

```
        end
```

```
    end
```

```
    StopMotor('all','off');
```

```
    ResetMotorAngle(MOTOR_B);
```

```
    ResetMotorAngle(MOTOR_C);
```

```
    SetMotor(MOTOR_B);
```

```
    SyncToMotor(MOTOR_C);
```

```
SetPower(35);
SetAngleLimit(giro90-20);
SetTurnRatio(100);
SendMotorSettings();
pause(1)
WaitForMotor(GetMotor)
pause(1)
StopMotor('all','off');
ResetMotorAngle(MOTOR_B);
ResetMotorAngle(MOTOR_C);

if GetUltrasonic(US) > distmax
    flag = 0;
    break
else
    SetMotor(MOTOR_B);
    Sync ToMotor(MOTOR_C);
    SetPower(35);
    SetAngleLimit(giro90-20);
    SetTurnRatio(-100);
    SendMotorSettings();
    pause(1)
    WaitForMotor(GetMotor)
    pause(1)
    StopMotor('all','off');
    ResetMotorAngle(MOTOR_B);
    ResetMotorAngle(MOTOR_C);
end
end

if GetSwitch(TACTO) == 1
    sonido = 0;
    derecha = 0;
    izquierda = 0;
    break
end

dir = -dir;
if GetSwitch(TACTO) == 1
    sonido = 0;
```

```

    derecha = 0;
    izquierda = 0;
    break
end

matriz(i,1) = sonido;
matriz(i,2) = derecha;
matriz(i,3) = izquierda;
i = i+1;
end

%CERRAR CONEXION
CloseSensor(S);
CloseSensor(US);
CloseSensor(TACTO);
COM_CloseNXT(handle);

matriz(i,1) = sonido;
matriz(i,2) = derecha;
matriz(i,3) = izquierda;

disp('ADQUIRIENDO BASE DE DATOS ACTUALIZADA...');
xlswrite('BASE DE DATOS.xls', matriz, 'SILLA', 'A2');
disp('GUARDADA EN
"c:\MATLAB\ecrobotNXT\RWTHMindstormsNXT\BASE DE DATOS.xls");

```

Inicialmente, se procede con los comandos básicos de inicio de programa: asegurarse de tener conexiones cerradas en caso de que se hayan ejecutado programas previos, eliminar las variables existentes para liberar memoria en el sistema y cerrar gráficos.

Luego, se procede con la creación de constantes que se utilizarán a lo largo de la programación.

A continuación, se procede con el inicio de la comunicación por bluetooth, mediante el comando `COM_OpenNXT`, el seteo inicial de los motores del prototipo y la inicialización de los sensores ultrasónico y de tacto.

Ahora, el cuerpo del programa, se divide en 5 etapas, las que hemos denominado como "flag". Todas estas etapas se encuentran ubicadas dentro de un lazo infinito.

**Primera Etapa (flag = 0):** Indica la operación de giro indefinido en su propio eje, esperando por respuesta de sonido. Luego de haber receptado una alta intensidad de sonido, procederá a parar de girar.

**Segunda Etapa (flag = 1):** Indica la operación del sensor ultrasónico. Procederá a avanzar en línea recta esperando por 2 tipos de respuesta: tiempo o presencia de obstáculo. Al recibir cualquiera de estas 2 respuestas, procederá a detenerse y continuará nuevamente con la Primera Etapa en caso del tiempo, o Tercera Etapa en caso de presencia de obstáculo.

**Tercera Etapa (flag = 2):** Indica la operación en caso de presencia de obstáculo. Sensará la distancia existente tanto a su lado derecho e izquierdo, comparará ambas distancias, y se dirigirá hacia la de mayor valor. En caso de dirigirse hacia la derecha, procederá con la Cuarta Etapa, y con la Quinta Etapa si se dirige hacia la izquierda.

**Cuarta Etapa (flag = 3):** Indica que se dirige hacia la derecha y servirá para recordar que luego deberá girar a la izquierda para continuar su camino.

**Quinta Etapa (flag = 4):** Indica que se dirige hacia la izquierda y servirá para recordar que luego deberá girar a la derecha para continuar su camino.

Cabe recalcar que al finalizar cada etapa siempre estará sensando el estado del sensor de tacto; en caso de ser presionado ("1" lógico), saldrá del lazo infinito.

Finalmente, al terminar la ejecución de todas las etapas, inicia con la adquisición de datos, tanto de intensidad de sonido, distancias medidas hacia la derecha y hacia la izquierda de la trayectoria recorrida, guardando dichos valores en una matriz llamada "matriz". Dicha matriz, luego de haber presionado el sensor de tacto, es enviada hacia Excel, como un historial de eventos realizados.

# CAPÍTULO 4

## 4 FUNCIONAMIENTO DEL SISTEMA

### 4.1 Implementación del Sistema

Como se indicó anteriormente el sistema diseñado en cuestión, consta de tres bloques, el primero consta del software MATLAB-SIMULINK, el segundo es el protocolo de comunicación Bluetooth con el cual se comunica el software con la silla; y el último es el prototipo de silla con sus motores y sensores.



Figura 4.1 Modelo del prototipo de silla.

Como se dijo anteriormente el software MATLAB-SIMULINK debe tener de los archivos adicionales explicados en el capítulo tres para su óptimo

funcionamiento, luego del cual se elabora un programa en el software explicado anteriormente, diseñado para que el hardware pueda desempeñarse de una manera óptima y realice las funciones en cuestión.

El programa establece las condiciones necesarias para que la silla pueda buscar a la persona a una cierta distancia, y pueda llegar al usuario a pesar de los obstáculos que se le presenten en el camino estando siempre a disposición del usuario. Luego de la obtención del programa debidamente compilado, se utiliza el dispositivo bluetooth para la transferencia de información al prototipo de silla para el inicio de su funcionamiento.

El prototipo de silla implementado se lo ha diseñado de la mejor manera para que pueda realizar giros libremente sin riesgo de caídas o trabas de la misma, además de la correcta ubicación del sensor de sonido lejos de los motores para que estos no interfieran con ruido al sensor y de un sensor ultrasónico en la parte de abajo para el chequeo constante de obstáculos. Mediante el uso del sensor de tacto agregado, servirá para proceder con la adquisición de datos de sonido y distancia y exportarlos en una tabla en un archivo de Excel.

## 4.2 Manejo y Prueba del Sistema

Al haber elaborado el programa en MATLAB-SIMULINK, necesitamos la correcta instalación del dispositivo bluetooth, chequeando que el mismo establezca la conexión entre el dispositivo bluetooth y el prototipo de silla para poder realizar la comunicación entre el programa de MATLAB-SIMULINK y el cerebro de la silla.

Luego de establecer la debida conexión, se procederá a verificar las características en el archivo "bluetooth.ini", principalmente el puerto serial de comunicación. Este, deberá ser el mismo que establece el BLUESOLEIL para permitir una correcta comunicación (en este caso COM8). Luego de esto, se procederá a llamar al programa "SILLA.m" en la ventana de comandos en MATLAB.

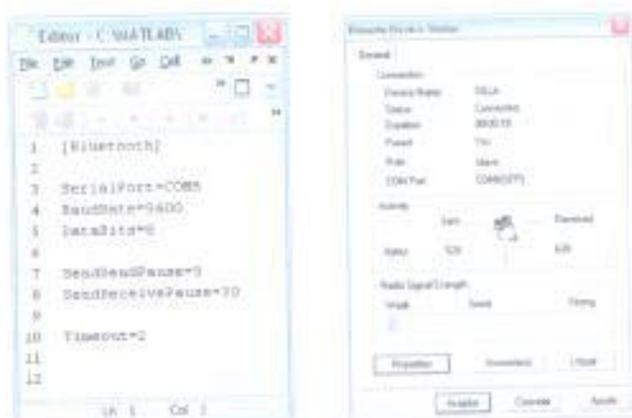


Figura 4.2 Prueba de Conectividad.

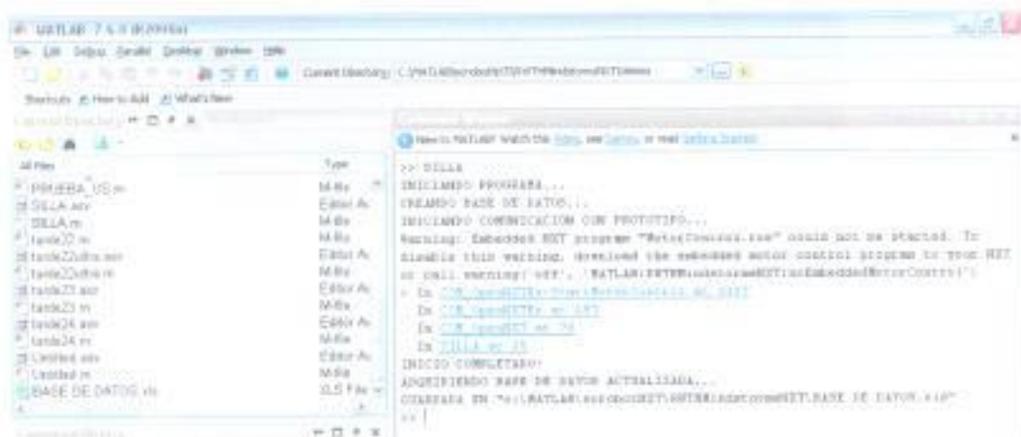


Figura 4.3 Ejecución del programa “SILLA.m” en la ventana de comandos de MATLAB.

Una vez funcionando el programa, en primer lugar, la silla girará indefinidamente para poder determinar el lugar en el cual se va a producir la mayor intensidad de sonido.

Luego de haber detectado la mayor intensidad de sonido, la cual puede ser emitida por una persona o por medio de dispositivos sonoros, el prototipo de silla detiene sus motores para luego dirigirse hacia el lugar donde se produjo el sonido.

Si el prototipo de silla se encuentra con algún obstáculo en el camino, procede a detectarlo deteniéndose frente a él, para luego realizar un muestreo de distancias localizando la mayor para dirigirse hacia esa dirección, tomando en cuenta que el rango de sensado de distancias es de  $180^\circ$ .



**Figura 4.4** Puesta en marcha del Sistema.

Ya que el prototipo de silla ha decidido avanzar en dirección que sensó una mayor distancia, procede a sensar cada cierto tiempo predeterminado hacia la misma dirección para verificar si el objeto sigue presente. En caso de ser cierto, continúa avanzando en esa dirección y repite el sensado; caso contrario, quedará disponible girando indefinidamente esperando nuevamente por la mayor intensidad de sonido.

Así mismo con este mecanismo la silla puede seguir al usuario a cualquier lugar donde este desee disponiendo siempre de la silla a cualquier lugar donde la persona vaya.

Además, la persona puede exportar los datos de sensado de la silla para tener información sobre la intensidad de los sonidos y distancias sensados por el prototipo de silla en la trayectoria que esta realizó, poniéndolos en un archivo Excel, estableciendo un historial de acciones realizadas por el prototipo de silla, por medio del sensor de tacto.

The image shows a screenshot of a Microsoft Excel spreadsheet. The spreadsheet has a header row with three columns labeled 'SONIDO', 'DENSIDAD', and 'DISTANCIA'. Below the header, there are five rows of data. The first row contains the values 42, 0, and 0. The second row contains 31, 39, and 42. The third row contains 40, 0, and 0. The fourth row contains 30, 33, and 112. The fifth row contains 47, 0, and 0. The spreadsheet is displayed in a window with a standard Windows interface, including a menu bar and a toolbar.

	SONIDO	DENSIDAD	DISTANCIA
1	42	0	0
2	31	39	42
3	40	0	0
4	30	33	112
5	47	0	0

Figura 4.5 Matriz de datos de "SILLA.m".

## CONCLUSIONES

1. El Sistema creado a través de la elaboración de un prototipo de silla demuestra ser una excelente herramienta, útil para las personas que se les imposibilita moverse fácilmente y que necesitan de una silla para poder desplazarse de un lugar a otro, y además, disponer del prototipo de silla aún cuando esta se encuentre a cierta distancia de la persona, proporcionando la seguridad al usuario de que el prototipo de silla regrese a él automáticamente.
2. El proyecto nos demuestra que puede crearse una silla capaz de ejecutar acciones que el usuario desee, la cual puede ser usada no sólo por personas con discapacidad sino también para la creación de sistemas inteligentes de prototipo de silla, logrando que las mismas se organicen, almacenen y estén siempre a disposición de las personas cuando estas las requieran, sin necesidad de que un grupo de personas las organice manualmente.
3. El software diseñado e implementado para el funcionamiento del prototipo de silla nos permite poseer la capacidad diseñar programas sencillos y muy útiles, para que puedan ser aplicados en ciertos dispositivos eléctricos o electrónicos, y a la vez estableciendo una comunicación entre el programa y el dispositivo electrónico via

bluetooth, logramos tener sistemas inteligentes capaces de ahorrar esfuerzo y optimizar tiempo a las personas en general.

4. El kit LEGO MINDSTORMS NXT resulta una herramienta muy flexible en cuanto a la variedad de diseños posibles a ser elaborados, demostrando así que se puede crear una infinidad de modelos permitiendo el desarrollando mecanismos cada vez más óptimos para un mejor desempeño del prototipo de silla que el propuesto en este proyecto, mejorando su desempeño y logrando movimientos más precisos.

## RECOMENDACIONES

1. Es necesario que el usuario del prototipo de silla tenga en cuenta la capacidad de alcance limitada del dispositivo bluetooth ya que esta se puede aplicar a distancias cortas para su óptimo funcionamiento, limitando el área de trabajo y además conocer que este tipo de comunicación produce determinados retardos al insertar el programa en el prototipo de silla .
2. Se recomienda precaución cuando se empleen sensores de sonido ya que estos son muy sensibles en su funcionamiento, ya que han sido diseñados para sensar intensidades de sonido, por lo tanto, al ensamblar el robot se necesita que el sensor esté alejado de cualquier tipo ruido como en este caso son los motores de la silla.
3. Es necesario tomar en cuenta el terreno en el cual la silla se va a desarrollar y la intensidad con que van a funcionar los motores del prototipo, ya que en terrenos muy lisos y con máxima fuerza de funcionamiento en los motores, estos pierden su precisión por deslizamiento de las llantas, es por esto que el terreno e intensidad de motores llevan una estrecha relación.

4. Existe el riesgo de que los obstáculos que se presenten al prototipo de silla estén fuera de la capacidad que posee el sensor ultrasónico y esta no pueda funcionar correctamente, por lo cual se recomienda el análisis de la ubicación y tipo de obstáculos, lo que permitirá a su vez una óptima ubicación del sensor ultrasónico en el prototipo, ampliándose su campo de visión y muestreo.
  
5. El sensor de sonido utilizado en la elaboración de este proyecto, no explota la capacidad máxima del prototipo de silla con respecto a la detección de la intensidad del mismo; por lo que se puede recomendar el uso de un mayor número de dichos sensores, u otro tipo de sensor que sea capaz de detectar frecuencias de sonidos, en lugar de su intensidad.

## **ANEXO**

### **TECNOLOGIA BLUETOOTH**



El nombre procede del rey danés y noruego Harald Blátand cuya traducción al inglés sería Harold Bluetooth (Diente Azul, aunque en lengua danesa significa 'de tez oscura') conocido por buen comunicador y por unificar las tribus noruegas, suecas y danesas. De la misma manera, Bluetooth intenta unir diferentes tecnologías como las de las computadoras, los teléfonos móviles y el resto de periféricos. El símbolo de Bluetooth es la unión de las runas nórdicas análogas a las letras B y H: (Hagall) y (Berkanan).

En 1994, Ericsson inició un estudio para investigar la viabilidad de una nueva interfaz de bajo costo y consumo para la interconexión vía radio entre dispositivos como teléfonos móviles y otros accesorios. El estudio partía de un largo proyecto que investigaba unos multi-comunicadores conectados a una red celular, hasta que se llegó a un enlace de radio de corto alcance, llamado MC link.

Conforme este proyecto avanzaba se fue haciendo claro que éste tipo de enlace podía ser utilizado ampliamente en un gran número de aplicaciones, ya que tenía como principal virtud que se basaba en un chip de radio.

## VERSIONES BLUETOOTH

- \* Bluetooth v.1.1
- \* Bluetooth v.1.2
- \* Bluetooth v.2.0
- \* Bluetooth v.2.1

La versión 1.2, a diferencia de la 1.1, provee una solución inalámbrica complementaria para co-existir Bluetooth y Wi-Fi en el espectro de los 2.4 GHz, sin interferencia entre ellos.

La versión 1.2 usa la técnica "Adaptive Frequency Hopping (AFH)", que ejecuta una transmisión más eficiente y un cifrado más seguro, provee una más rápida configuración de la comunicación con los otros dispositivos bluetooth dentro del rango del alcance, como pueden ser PDA, computadoras portátiles, computadoras de escritorio, Headsets, impresoras y celulares.

La versión 2.0, creada para ser una especificación separada, principalmente incorpora la técnica "Enhanced Data Rate" (EDR) que le permite mejorar las velocidades de

transmisión en hasta 3Mbps a la vez que intenta solucionar algunos errores de la especificación 1.2.

La versión 2.1, simplifica los pasos para crear la conexión entre dispositivos, además el consumo de potencia es 5 veces menor.

## LISTA DE APLICACIONES

- \* Conexión sin cables entre los celulares y equipos de manos libres y kit para vehículos.

- \* Red inalámbrica en espacios reducidos donde no sea tan importante un ancho de banda grande.

- \* Comunicación sin cables entre la computadora y dispositivos de entrada y salida. Mayormente impresora, teclado y mouse.

- \* Transferencia de ficheros entre dispositivos vía OBEX.

\* Transferencia de fichas de contactos, citas y recordatorios entre dispositivos vía OBEX.

\* Reemplazo de la tradicional comunicación por cable entre equipos GPS y equipamiento médico.

\* Controles remotos (tradicionalmente dominado por el infrarrojo).

\* Enviar pequeñas publicidades desde anunciantes a dispositivos con Bluetooth. Un negocio podría enviar publicidad a teléfonos móviles cuyo Bluetooth (los que lo posean) estuviera activado al pasar cerca.

\* Las consolas Sony PlayStation 3 y Nintendo Wii incorporan Bluetooth, lo que les permite utilizar mandos inalámbricos.

## **EN QUÉ CONSISTE LA TECNOLOGÍA BLUETOOTH**

Últimamente, el panorama de los móviles está salpicado por la resonancia de una palabra de la que, en muchos casos, desconocemos su significado. Esa palabra es Bluetooth.

Anuncios de operadoras, de marcas de móviles, manos libres, ordenadores y hasta automóviles nos ensalzan las bondades de su producto o servicio incluyendo el reclamo Bluetooth, como la panacea del siglo. Pero, ¿qué es y para qué sirve?

Bluetooth es una tecnología o, como dicen sus creadores, un estándar global de comunicaciones sin cables, que hace posible la transmisión de voz y datos entre diferentes equipos a través de un enlace de radiofrecuencia. Con ello, se facilitan las comunicaciones entre equipos móviles y fijos, y se eliminan los molestos cables y conectores entre éstos.

Sus antecedentes se remontan a 1994, año en el que la compañía Ericsson puso en marcha un estudio para analizar e investigar la viabilidad de un interfaz vía radio para la interconexión entre teléfonos móviles y otros equipos que, a su vez, tuviera un bajo coste, un mínimo consumo y que no precisara de cables.

La idea era tan sugerente que pronto despertó el interés de otras grandes compañías como Nokia, IBM, Toshiba e Intel quienes, junto a Ericsson, crearon en 1998 el grupo conocido por las siglas SIG (grupo de interés especial). De este modo, se aseguraba la interoperabilidad entre equipos de distintos fabricantes. Pero también perseguía otros objetivos no menos ambiciosos, ya que los objetivos eran que el sistema operara en todo

l mundo, que consumiera poca energía (porque debería ir integrado en equipos alimentados por baterías), y que la conexión soportara voz y datos (por tanto, aplicaciones multimedia).

Desde entonces hasta la fecha, Bluetooth se ha convertido en una norma común mundial para la conexión inalámbrica y se prevé que sea utilizada en millones de teléfonos móviles, PCs, ordenadores portátiles y toda una gama de aparatos eléctricos.

## **APLICACIONES ILIMITADAS**

Visto lo anterior, las posibilidades de conexión inalámbrica que Bluetooth ofrece son prácticamente ilimitadas. Además, como la radiofrecuencia que se utiliza está disponible a escala mundial, Bluetooth puede ofrecer acceso rápido y seguro a una conexión inalámbrica en todo el mundo. Con un potencial así, no es extraño que esta tecnología o estándar vaya a convertirse en una de las tecnologías de más rápida adopción de la historia, según asegura la propia Nokia.

De momento, las aplicaciones de Bluetooth a la vida diaria son amplias, siempre y cuando las distancias sean cortas y la información a transmitir no sea excesiva. He aquí unos ejemplos de ellas:

### ➤ *Telefonía*

Por un lado, los móviles que disponen de Bluetooth permiten conectarse con otros terminales compatibles para compartir la agenda, intercambiar melodías o logos, fotografías, etc. Y todo ello a un coste cero, ya que no se utiliza la red para ello. Pero, por otro lado, un mismo teléfono lo podemos utilizar como fijo (si se encuentra dentro del radio de acción del punto de acceso), como móvil (sin nos encontramos fuera del radio de acción), como medio de acceso a nuestros contactos, correo electrónico, etc. Asimismo, mediante unos auriculares inalámbricos podemos acceder a nuestro portátil o PDA y realizar llamadas como si se tratara de un teléfono.

### ➤ *Oficina*

Con Bluetooth los molestos y engorrosos cables conectados a nuestro ordenador desaparecen. Así, el teclado, el ratón, el disco duro, la impresora o cualquier periférico pueden comunicarse mediante esta tecnología sin la necesidad de cables. Además, se cuenta con la ventaja de la sincronización continua, ya que los dispositivos Bluetooth mantienen la información sincronizada constantemente, de tal modo que si modificamos en nuestro ordenador unos datos y éstos estaban almacenados en nuestro PDA, aquí también se modificarán automáticamente.

### ➤ *Automoción*

A la vista de los spots publicitarios de televisión, está claro que el sector automovilístico apuesta fuerte por esta tecnología. La primera aplicación en este ámbito son los dispositivos manos libres para el móvil. Pero las posibilidades que se barajan van más allá: desde sistemas de navegación, hasta lectores de DVDs., pasando por consolas de videojuegos y altavoces.

### ➤ *Ocio*

Auriculares inalámbricos para escuchar música, juguetes teledirigidos por control remoto o mandos a distancia para los televisores son algunas de las aplicaciones de Bluetooth en el campo del ocio; aplicaciones que, en un futuro, se verán incrementadas hasta lo inimaginable.

### ➤ *Vivienda*

Igualmente seductoras serán las aplicaciones que Bluetooth tendrá en los hogares y, en concreto, en el ámbito de los electrodomésticos. De momento, ya existen ollas a presión capaces de comunicarse con la vitrocerámica y regular el tiempo y potencia de cocción, incluso apagarse cuando ésta se ha terminado. Encender la luz, subir o bajar las persianas, poner en marcha la lavadora o el lavavajillas, son algunas cosas que son posibles de hacer a través de una llamada de teléfono con Bluetooth.

De momento, y hasta que Bluetooth adquiera una mayor amplitud y se imponga a las demás tecnologías, para poder hacer uso de ella es preciso disponer de un chip o pequeña tarjeta instalada en cada dispositivo. Si éste no cuenta con ello, no será posible añadirla.

## **CÓMO FUNCIONA**

El estándar Bluetooth, del mismo modo que WiFi, utiliza la técnica FHSS (Frequency Hopping Spread Spectrum, en español Espectro ensanchado por saltos de frecuencia), que consiste en dividir la banda de frecuencia de 2.402 - 2.480 GHz en 79 canales (denominados saltos) de 1 MHz de ancho cada uno y, después, transmitir la señal utilizando una secuencia de canales que sea conocida tanto para la estación emisora como para la receptora.

Por lo tanto, al cambiar de canales con una frecuencia de 1600 veces por segundo, el estándar Bluetooth puede evitar la interferencia con otras señales de radio.

## **PRINCIPIO DE COMUNICACION**

El estándar Bluetooth se basa en el modo de operación maestro/esclavo. El término "piconet" se utiliza para hacer referencia a la red formada por un dispositivo y todos los dispositivos que se encuentran dentro de su rango. Pueden coexistir hasta 10 piconets dentro de una sola área de cobertura. Un dispositivo maestro se puede conectar simultáneamente con hasta 7 dispositivos esclavos activos (255 cuando se encuentran en

modo en espera). Los dispositivos en una piconet poseen una dirección lógica de 3 bits, para un máximo de 8 dispositivos. Los dispositivos que se encuentran en el modo en espera se sincronizan, pero no tienen su propia dirección física en la piconet.



Fig. A.1 Dispositivo Maestro

En realidad, en un momento determinado, el dispositivo maestro sólo puede conectarse con un solo esclavo al mismo tiempo. Por lo tanto, rápidamente cambia de esclavos para que parezca que se está conectando simultáneamente con todos los dispositivos esclavos. Bluetooth permite que dos piconets puedan conectarse entre sí para formar una red más amplia, denominada "scatternet", al utilizar ciertos dispositivos que actúan como puente entre las dos piconets.

### COMO SE ESTABLECEN LA CONEXIONES?

El establecimiento de una conexión entre dos dispositivos Bluetooth sigue un procedimiento relativamente complicado para garantizar un cierto grado de seguridad, como el siguiente:

Durante el uso normal, un dispositivo funciona en "modo pasivo", es decir, que está escuchando la red.

El establecimiento de una conexión comienza con una fase denominada "solicitud", durante la cual el dispositivo maestro envía una solicitud a todos los dispositivos que encuentra dentro de su rango, denominados puntos de acceso. Todos los dispositivos que reciben la solicitud responden con su dirección.

El dispositivo maestro elige una dirección y se sincroniza con el punto de acceso mediante una técnica denominada paginación, que principalmente consiste en la sincronización de su reloj y frecuencia con el punto de acceso.

De esta manera se establece un enlace con el punto de acceso que le permite al dispositivo maestro ingresar a una fase de descubrimiento del servicio del punto de acceso, mediante un protocolo denominado SDP (Service Discovery Protocol, en español Protocolo de descubrimiento de servicios).

Cuando esta fase de descubrimiento del servicio finaliza, el dispositivo maestro está preparado para crear un canal de comunicación con el punto de acceso, mediante el protocolo L2CAP.

Según cuáles sean las necesidades del servicio, se puede establecer un canal adicional, denominado RFCOMM que funciona por el canal L2CAP, para proporcionar un puerto serial virtual. De hecho, algunas aplicaciones se han diseñado para que puedan conectarse a un puerto estándar, independientemente del hardware utilizado. Por ejemplo, se han diseñado ciertos programas de navegación en carretera para la conexión con cualquier dispositivo GPS Bluetooth (GPS significa Global Positioning System [Sistema de posicionamiento global], un sistema de localización geográfica por satélite para encontrar las coordenadas geográficas de un dispositivo móvil o de un vehículo).

El punto de acceso puede incluir un mecanismo de seguridad denominado emparejamiento, que restringe el acceso sólo a los usuarios autorizados para brindarle a la piconet cierto grado de protección. El emparejamiento se realiza con una clave cifrada comúnmente conocida como "PIN" (PIN significa Personal Information Number [Número de identificación personal]). Para esto, el punto de acceso le envía una solicitud de emparejamiento al dispositivo maestro. La mayoría de las veces se le solicitará al usuario que ingrese el PIN del punto de acceso. Si el PIN recibido es correcto, se lleva a cabo la conexión.

En el modo seguro, el PIN se enviará cifrado con una segunda clave para evitar poner en riesgo la señal.

Cuando el emparejamiento se activa, el dispositivo maestro puede utilizar libremente el canal de comunicación establecido.

## **PERFILES BLUETOOTH**

El estándar Bluetooth define un cierto número de perfiles de aplicación (denominados perfiles Bluetooth) para definir qué tipos de servicios ofrece un dispositivo Bluetooth. Por lo tanto, cada dispositivo puede admitir múltiples perfiles. A continuación encontrará una lista de los principales perfiles Bluetooth:

Perfil de distribución de audio avanzado (A2DP)

Perfil de control remoto de audio y video (AVRCP)

Perfil básico de imagen (BIP)

Perfil básico de impresión (BPP)

Perfil de telefonía inalámbrica (CTP)

Perfil de red de marcado (DUNP)

Perfil de fax (FAX)

Perfil de transferencia de archivos (FTP)

Perfil de acceso genérico (GAP)

Perfil genérico de intercambio de objetos (GOEP)

Perfil de sustitución de cable de copia impresa (HCRP)

Perfil manos libres (HFP)

Perfil de dispositivo de interfaz humana (HID)

Perfil de auricular (HSP)

Perfil de intercomunicador (IP)

Perfil de acceso LAN (LAP)

Perfil de objeto push (OPP)

Perfil de redes de área personal (PAN)

Perfil de acceso SIM (SAP)

Perfil de aplicación de descubrimiento de servicio (SDAP)

Perfil de sincronización (SP): se utiliza para sincronizar el dispositivo con un administrador de información personal (abreviado PIM).

Perfil de puerto de serie (SPP)

## BIBLIOGRAFÍA

- [1]. Gasperi Michael; Hurbain Philippe; Hurbain Isabelle, Extreme NXT – Entending the *LEGO MINDSTORMS NXT*, edition Technology in Action Press, 2007
- [2]. Lego Mindstorms, Página HTML,  
<http://mindstorms.lego.com>.
- [3]. Wikipedia, Lego Mindstorms , Página HTML,  
[http://es.wikipedia.org/wiki/Lego\\_Mindstorms](http://es.wikipedia.org/wiki/Lego_Mindstorms).
- [4]. Wikipedia, Matlab, Página HTML,  
<http://es.wikipedia.org/wiki/MATLAB>.
- [5]. Matlab, Principios Básicos, Página HTML,  
<http://www.monografias.com/trabajos5/matlab/matlab.shtml>.
- [6]. Simulink, Principios Básicos, Página HTML,  
<http://www.mathworks.es/products/simulink>.
- [7]. Simulink, Principios Básicos, Página HTML,  
<http://www.monografias.com/trabajos22/aplicaciones-matlab/aplicaciones-matlab.shtml>.
- [8]. Matlab-Simulink, Generalidades, Página HTML,  
<http://www.mathworks.co.kr/matlabcentral/fileexchange/?term=tag%3A%22simulink%22>.
- [9]. Bluetooth, introducción, Página HTML,  
<http://es.wikipedia.org/wiki/Bluetooth>.
- [10]. Protocolo de comunicaciones bluetooth, Página HTML,  
<http://www.xataka.com/otros/mas-de-un-millardo-de-dispositivos-bluetooth>.