

# **ESCUELA POLITECNICA NACIONAL**

## **FACULTAD DE INGENIERIA ELECTRICA Y ELECTRONICA**

### **DISEÑO Y CONSTRUCCION DE TRES MINI ROBOTS EXPLORADORES COOPERATIVOS**

**PROYECTO PREVIO A LA OBTENCION DEL TITULO DE INGENIERO EN  
ELECTRONICA Y CONTROL**

**MAURICIO FERNANDO RODRIGUEZ BORJA**

(mauro\_drog@hotmail.com)

**SANTIAGO DAVID SANDOBALIN GUAMAN**

(sant\_d387@hotmail.com)

**DIRECTOR: MSc. CARLOS PATRICIO BURBANO ROMERO**

(pburbanor@hotmail.com)

**Quito, Septiembre 2013**

## DECLARACIÓN

Nosotros, Rodríguez Borja Mauricio Fernando y Sandobalín Guamán Santiago David, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

---

Rodríguez Borja Mauricio Fernando

---

Sandobalín Guamán Santiago David

## CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Rodríguez Borja Mauricio Fernando y Sandobalín Guamán Santiago David, bajo mi supervisión.

---

MSc. Patricio Burbano Romero  
DIRECTOR DEL PROYECTO

## AGRADECIMIENTO

Mis agradecimientos a mi familia por su apoyo, el esfuerzo y la confianza que depositaron en mi; a MSc. Patricio Burbano y a Phd. Andrés Rosales quienes fueron partícipes de la realización del presente Proyecto de Titulación, al Club de Robótica y a las personas que lo presidían que nos ayudaron con el espacio de trabajo y a mi compañero de proyecto de titulación por el trabajo realizado. A mis amigos por su apoyo, su preocupación, por sus consejos, por una palabra de aliento que nunca faltó y por los momentos compartidos durante toda mi vida estudiantil en esta prestigiosa universidad.

**Mauricio Rodríguez**

Agradecimientos a MSc. Patricio Burbano y a Phd. Andrés Rosales quienes nos han ayudado en la realización del presente Proyecto de Titulación, al Club de Robótica que nos ayudó con un espacio para la realización del proyecto, a mi familia por todo el apoyo y la paciencia, a mis amigos y amigas con quienes compartí varios momentos en mi vida y que de una u otra forma me han ayudado a seguir adelante y me han apoyado siempre.

**Santiago Sandobalín**

## DEDICATORIA

Este Proyecto de Titulación va dedicado a mis padres Fernando e Inés quienes me apoyan, por su esfuerzo, su trabajo y su empeño para que sus hijos salgamos adelante, a mis hermanos Santiago y Matías, que son parte importante de mi vida y Sebastián que me cuida desde el cielo. A mi familia por parte de mi padre y a mi familia por parte de mi madre, por estar pendientes y preocupados por mi; y a mis amigos por ser un apoyo importante para dar este paso en mi vida.

**Mauricio Rodríguez**

Este Proyecto de Titulación va dedicado a mi madre Guadalupe quien con su esfuerzo, trabajo y lucha diaria me ha permitido dar este gran paso en mi vida, a mis hermanas y hermano quienes con su apoyo y colaboración me han ayudado a seguir adelante a lo largo de estos años y a mis amistades quienes también son un pilar muy importante en la obtención de este logro.

**Santiago Sandobalín**

## CONTENIDO

<b>RESUMEN .....</b>	<b>ix</b>
<b>PRESENTACIÓN .....</b>	<b>x</b>
<b>CAPITULO 1 MARCO TEÓRICO .....</b>	<b>1</b>
1.1 INTRODUCCIÓN .....	1
1.2 ROBÓTICA Y MINI ROBÓTICA .....	2
1.3 ROBOTS MÓVILES CON RUEDAS .....	3
1.3.1 ESTRUCTURA GENERAL DE UN ROBOT MÓVIL .....	4
1.3.2 CONSIDERACIONES PARA EL MOVIMIENTO DE UN ROBOT MÓVIL .....	4
1.3.2.1 Grado de libertad (GDL).....	5
1.3.2.2 Tipos de ruedas .....	5
1.3.2.3 Centro instantáneo de rotación (ICR) o centro instantáneo de curvatura (ICC) .....	7
1.3.3 CONFIGURACIONES TÍPICAS PARA ROBOTS MÓVILES CON RUEDAS .....	8
1.3.3.1 Configuración diferencial .....	8
1.3.3.2 Configuración triciclo .....	9
1.3.3.3 Configuración Ackerman .....	9
1.3.3.4 Configuración sincronizada .....	10
1.3.3.5 Configuración omnidireccional .....	11
1.3.3.6 Configuración con múltiples grados de libertad .....	12
1.3.3.7 Tracción mediante orugas .....	12
1.4 ESTIMACIÓN DE LA POSICIÓN DE UN ROBOT MÓVIL.....	13
1.4.1 ESTIMACIÓN EXPLÍCITA DE LA POSICIÓN .....	15
1.4.1.1 Estimadores basados en medidas internas .....	15
1.4.1.2 Estimadores basados en estaciones de transmisión.....	16
1.4.1.3 Estimación mediante balizas.....	17
1.4.2 ESTIMACIÓN MEDIANTE PERCEPCIÓN DEL ENTORNO .....	18
1.5 ALGORITMOS DE RECORRIDO DENTRO DE UN LABERINTO.....	19
1.5.1 ALGORITMOS DE LA MANO DERECHA Y LA MANO IZQUIERDA.....	19
1.5.2 ALEATORIOS .....	20
1.5.3 RUTA PREDETERMINADA.....	20
1.5.4 PROCESAMIENTO DE INFORMACIÓN .....	20
1.5.5 ALGORITMO DE RECORRIDO RECURSIVO .....	21

1.5.6	ALGORITMO DE TARRY .....	21
1.5.7	ALGORITMO DE PLEDGE.....	22
1.5.8	ALGORITMO PLANTEADO POR LEE .....	22
1.6	COOPERATIVISMO .....	23
1.6.1	TIPOS DE CONTROL PARA SISTEMAS COOPERATIVOS.....	23
1.6.1.1	Control centralizado.....	23
1.6.1.2	Control distribuido.....	24
1.6.2	TIPOS DE SISTEMAS .....	24
1.6.2.1	Muy acoplados o colectivos.....	24
1.6.2.2	Poco acoplados o manadas.....	25
1.6.3	VENTAJAS DEL USO DE ROBOTS COOPERATIVOS .....	25
1.6.4	INCONVENIENTES EN EL USO DE ROBOTS COOPERATIVOS .....	25
1.7	COMUNICACIÓN POR RADIO FRECUENCIA .....	26
1.7.1	TECNOLOGÍA ZIGBEE .....	26
1.7.2	DISPOSITIVOS ZIGBEE .....	27
1.7.3	TOPOLOGÍAS ZIGBEE.....	28
<b>CAPITULO 2 DISEÑO Y CONSTRUCCIÓN DE LAS MINI PLATAFORMAS ROBÓTICAS.....</b>		<b>29</b>
2.1	DESCRIPCIÓN DE LOS MINIROBOTS EXPLORADORES COOPERATIVOS.....	29
2.1.1	COMPONENTES UTILIZADOS .....	30
2.1.1.1	Arduino Duemilanove.....	30
2.1.1.2	Módulos Arduino XBEE SHIELD .....	33
2.1.1.3	Módulos XBEE Serie 2.....	34
2.1.1.4	Sensores de distancia por ultrasonido HC - SR04 .....	35
2.1.1.5	Módulo L298N para los micromotores DC .....	38
2.1.1.6	Micromotores DC .....	39
2.1.1.7	Encoder de cuadratura .....	40
2.1.1.8	Baterías de Litio – Polímero (LiPo).....	41
2.1.2	DISEÑO DE LA ESTRUCTURA DEL MINIROBOT.....	42
2.2	CONSTRUCCIÓN DE LAS ESTRUCTURAS.....	43
2.3	ESTRUCTURAS DEFINITIVAS DE LOS MINIROBOTS .....	45
<b>CAPITULO 3 DESARROLLO DE LOS ALGORITMOS DE CONTROL, EVASIÓN, MAPEO Y COOPERACIÓN.....</b>		<b>47</b>
3.1	LOCOMOCIÓN DE LOS MINIROBOTS .....	47

3.1.1	ALGORITMOS Y CONTROL PI PARA EL AVANCE DE LOS MINIROBOTS .....	48
3.1.1.1	Controlador PID.....	48
3.1.1.2	Algoritmo de control para el avance del minirobot .....	52
3.1.2	CONTROL PARA GIROS DERECHA IZQUIERDA Y GIRO COMPLETO .....	54
3.1.2.1	Interrupción de programa por encoders .....	54
3.1.2.2	Giros .....	55
3.2	ALGORITMOS DE EXPLORACIÓN Y NAVEGACIÓN DENTRO DEL LABERINTO.....	57
3.2.1	ALGORITMO DE NAVEGACIÓN IMPLEMENTADO.....	57
3.3	OBTENCIÓN DE LA TRAYECTORIA Y LA RUTA CORTA.....	59
3.3.1	OBTENCIÓN DE LA TRAYECTORIA – ODOMETRÍA.....	59
3.3.2	OBTENCIÓN DE LA RUTA CORTA .....	62
3.4	TRATAMIENTO DE DATOS .....	66
3.4.1	CONFIGURACIÓN DE LOS MÓDULOS XBEE .....	66
3.4.1.1	Configuración del coordinador .....	69
3.4.1.2	Configuración de los routers.....	70
3.4.2	ENVÍO Y RECEPCIÓN DE DATOS .....	71
3.4.2.1	Envío de datos desde el arduino.....	71
3.4.2.2	Recepción de datos en la PC.....	72
3.5	INTERFAZ GRÁFICA DE MATLAB (GUIDE).....	76
3.6	ALGORITMOS IMPLEMENTADOS EN EL ARDUINO .....	80
<b>CAPITULO 4 PRUEBAS Y ANÁLISIS DE RESULTADOS.....</b>		<b>95</b>
4.1	CALIBRACIÓN DE LOS MINIROBOTS.....	95
4.1.1	CALIBRACIÓN DEL PI.....	95
4.1.2	CALIBRACIÓN DE LOS GIROS DEL MREC .....	98
4.1.3	AVANCE DEL MREC POR DIVERSAS ZONAS .....	99
4.2	PRUEBAS DE RED ZIGBEE .....	101
4.2.1	ENVIAR Y RECIBIR DATOS PUNTO – PUNTO.....	102
4.2.2	RECIBIR Y ENVIAR DATOS PUNTO – MULTIPUNTO .....	104
4.3	PRUEBAS DE EXPLORACIÓN Y MAPEAMIENTO .....	106
4.3.1	PRUEBAS CON UN MINIROBOT.....	106
4.3.2	PRUEBAS MULTIROBOT .....	112
4.3.2.1	Funcionamiento de los 3 minirobots.....	113
4.3.2.2	Mapeo de sectores diferentes del laberinto .....	113
4.4	PRUEBAS AL ALGORITMO PARA LA SOLUCIÓN DEL LABERINTO .....	118
<b>CAPITULO 5 CONCLUSIONES Y RECOMENDACIONES .....</b>		<b>120</b>



5.1	CONCLUSIONES .....	120
5.2	RECOMENDACIONES .....	121
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>		<b>123</b>

**ANEXOS**

## RESUMEN

Este proyecto de titulación trata sobre el diseño y la construcción de tres minirobots exploradores cooperativos, los cuales se utilizan para explorar y mapear un laberinto mediante la implementación de algoritmos de locomoción, exploración, obtención de la trayectoria recorrida y resolución del laberinto obteniendo una ruta corta.

Se realiza el diseño de algoritmos para la navegación de los minirobots dentro del laberinto, es decir, un algoritmo que le permita al minirobot ir por el centro del camino y girar en ángulos rectos. Un algoritmo para la exploración, donde el minirobot pueda reconocer la zona del laberinto en el cual se encuentra y así dar las señales correctas de control para continuar su camino con la regla de la mano izquierda. Un algoritmo para que cuando el minirobot termine de grabar la ruta por donde se desplazó, elimine los caminos largos y caminos sin salida, y pueda volver a realizar el recorrido pero con una ruta más corta.

La comunicación entre los minirobots y la PC se realizó con la programación de los módulos XBEE S2, donde los tres minirobots siempre se comunican con un coordinador que es la PC, para posteriormente hacer un análisis de los datos recogidos y así dibujar el entorno por donde los minirobots hayan recorrido.

Dentro del proyecto de titulación también se desarrolló una interfaz gráfica con la ayuda de MATLAB. Para la demostración del proyecto se han desarrollado varios menús donde los minirobots realizan diferentes actividades como por ejemplo: exploración de uno y de los tres minirobots dentro del laberinto, exploración y mapeo con un minirobot, exploración y mapeo con los 3 minirobots por sectores diferentes del laberinto; y, exploración y ruta corta del laberinto. La comunicación se realizó con la creación de una red zigbee.

## **PRESENTACIÓN**

El Proyecto de Titulación está distribuido de la siguiente manera:

En el Capítulo 1 se detallan los conceptos acerca de la robótica y de la mini robótica, así como también la teoría con respecto a los vehículos terrestres, un prototipo de los cuales se desarrolló en este proyecto. La estimación de la posición de los minirobots y el tipo de comunicación utilizado, también se detalla en este capítulo.

En el Capítulo 2 se desarrolla todo el diseño y la construcción de los minirobots, como también la revisión de las diferentes partes que constituyen a los minirobot.

En el Capítulo 3 se realiza todo el desarrollo del software, es decir todos los algoritmos para la navegación, exploración y comunicación de los minirobots, y también el diseño de la interfaz grafica (GUIDE) con la ayuda de MATLAB.

En la Capítulo 4 se exponen los resultados obtenidos al hacer las pruebas respectivas con los minirobots dentro de un entorno, en este caso un laberinto.

En el Capítulo 5 se presenta las conclusiones y las recomendaciones del proyecto de titulación.

En la parte final se encuentran los anexos: el costo del proyecto de titulación, manual de usuario, planos de los minirobots, hojas de datos de los componentes de los minirobots y los algoritmos del recorrido dentro de un laberinto.

# CAPITULO 1

## MARCO TEÓRICO

### 1.1 INTRODUCCIÓN

Los robots actualmente son producto de la ingeniería y por ende son concebidos para producir bienes y servicios, explotar recursos naturales, investigación y entretenimiento. Desde esta perspectiva son máquinas con las que se continúa una actividad que parte desde los propios orígenes de la humanidad, y que desde el comienzo de la Edad Moderna se fundamenta esencialmente en conocimientos científicos. El desarrollo de las máquinas ha sido fuertemente influido por el progreso tecnológico; que pasa desde máquinas que tienen como objetivo exclusivo la amplificación de la potencia muscular del ser humano, sustituyéndolo en su trabajo físico, a máquinas o instrumentos capaces de procesar información y desarrollar una actividad, al punto de tomar decisiones basándose en una lógica de programación establecida.

Desde sus inicios la robótica ha tenido una gran evolución, empezando por los robots manipuladores, sistemas mecánicos multifuncionales capaces de hacer una secuencia fija y que realizan sus actividades por medio de un sistema de control bastante sencillo, estos robots pertenecen a la primera generación. En la segunda generación de robots se encuentran los robots de aprendizaje, capaces de repetir una secuencia de movimientos que fueron ejecutados previamente por un operador. Para la tercera generación aparecen los robots con control sensorizado, varios sensores le dan información al robot de su entorno, esta información es procesada en una computadora o microcontrolador y por medio de un programa de control y actuadores realizan ciertas actividades, de una forma autónoma. En la cuarta generación de robots aparecen los robots inteligentes, que son similares a los anteriores pero que pueden tomar decisiones y que constantemente están aprendiendo, las decisiones tomadas y el control del proceso es en tiempo real.

En los últimos años se ha podido observar la gran evolución de los robots, teniendo robots cada vez más sofisticados y con habilidades impresionantes. Se han realizado robots capaces de simular movimientos físicos de una persona como caminar, recoger objetos y simular movimientos del rostro, demostrando emociones. También se ha dado un enfoque especial en ciertas comunidades de insectos para desarrollar el cooperativismo entre robots, imitando de manera aproximada a estas comunidades; insectos que cumplen una tarea sencilla y específica pero que de manera coordinada y organizada ejecutan funciones bastante complejas. Para ello ha influido el avance científico y tecnológico como la utilización de la microelectrónica, sensores que proveen de una mejor percepción del ambiente, actuadores con mejores características de funcionamiento y recursos como la inteligencia artificial, algoritmos genéticos y redes neuronales que amplían las características funcionales de los robots.

## **1.2 ROBÓTICA Y MINI ROBÓTICA[1]**

La definición de robot no está del todo establecida, no hay un concepto universal que defina esta palabra, aunque hay varias definiciones que han tenido bastante difusión.

La definición de la R.I.A (Robot Industry Association por sus siglas en inglés) es la siguiente:

“Un robot industrial, es un manipulador multifuncional reprogramable capaz de mover cargas, piezas, herramientas o dispositivos especiales, según trayectorias varias; programado para realizar trabajos diversos”

La definición de la B.R.A, Asociación de Robótica Británica es la siguiente:

“Un mecanismo diseñado para manipular y transportar piezas o útiles especiales, por medio de movimientos variables, programados para la ejecución de tareas específicas de manufactura”

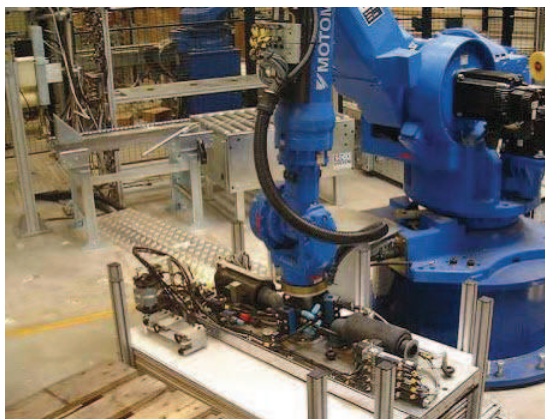


Figura 1.1 Brazo Automatizado Industrial[2]

Otra definición de interés para este trabajo de investigación es la de un mini robot, su definición aproximada es:

“Son aquellos robots de bajo costo cuya función primordial es la realización de tareas sencillas que pueden ejecutar con el mínimo de errores” [3]

Esta descripción es un tanto generalizada pues ahora pueden ejecutar funciones más avanzadas.

Existen cuatro tipos de mini robots:

- Terrestres
- Acuáticos
- Aéreos
- Espaciales

En el presente proyecto de titulación se estudia lo referente a los mini robots terrestres.

### **1.3 ROBOTS MÓVILES CON RUEDAS[4]**

Los robots móviles pequeños o microrobots por lo general están destinados a realizar tareas pequeñas con rapidez y precisión, simular actividades humanas y

de animales. La movilidad de los microrobots es una ventaja para realizar ciertas tareas, para trabajar de manera aislada o en equipo con otros robots.

Los microrobots tienen reducidas dimensiones con un bajo grado de control, el control que existe es para regular la velocidad de las ruedas de tracción mientras que adquieren un alto grado de complejidad cuando tienen que realizar tareas cooperativas.

Para que un robot realice automáticamente tareas de modo eficiente depende de la construcción mecánica y de la programación. La precisión en el desplazamiento se encuentra determinada por su sistema mecánico, mientras que la autonomía y la inteligencia depende del programa que gobierna las acciones del microrobot.

### **1.3.1 ESTRUCTURA GENERAL DE UN ROBOT MÓVIL**

Los robots móviles tienen un cuerpo o estructura mecánica, sensores, actuadores y un sistema inteligente.

- Estructura mecánica: estructura con ruedas, patas y orugas.
- Actuadores: motores, luces, brazos, ruedas y en definitiva cualquier elemento que permita la interacción con el entorno.
- Sensores: sonar, laser, cámaras y cualquier elemento que proporcione información del entorno.
- Inteligencia: Métodos y algoritmos.

### **1.3.2 CONSIDERACIONES PARA EL MOVIMIENTO DE UN ROBOT MÓVIL**

Para el movimiento de un robot móvil se deben considerar los grados de libertad, tipo de rueda y el centro instantáneo de rotación. Los grados de libertad y los tipos de ruedas son aspectos que intervienen en el proceso de control y análisis del movimiento del robot.

### 1.3.2.1 Grado de libertad (GDL)

Es cada uno de los movimientos de desplazamiento y rotación que puede realizar el robot.

- Un cuerpo que se mueve en dos dimensiones tiene 3 GDL, uno de rotación y 2 de traslación (ejes X, Y).
- Un cuerpo que se mueve en 3 dimensiones tiene 6 GDL, 3 rotaciones y 3 desplazamientos (ejes X, Y, Z).

De la capacidad y característica de desplazamiento que tenga un robot móvil se destacan los sistemas holonómicos y no holonómicos.

Un sistema es holonómico si es capaz de modificar su dirección instantáneamente y sin necesidad de rotar previamente.

Del mismo modo, un robot móvil con 2 ruedas es no holonómico ya que no puede moverse hacia la izquierda o la derecha. Siempre lo hace hacia delante en la dirección definida por la velocidad de sus ruedas.

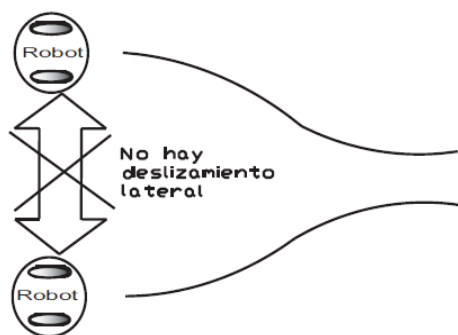


Figura 1.2 Sistema no holonómico[5]

### 1.3.2.2 Tipos de ruedas

Las ruedas proporcionan la capacidad de movilidad en un robot. Se clasifican de la siguiente manera:



**Rueda fija.** No posee articulación de dirección, por lo que su posición respecto a la estructura es fija.

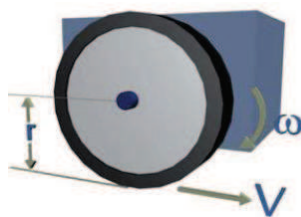


Figura 1.3 Rueda fija[5]

**Orientación centrada.** Tiene articulación de dirección, es decir, es orientable respecto a la estructura del robot móvil, pasando su eje de dirección por el centro de rotación de la rueda.

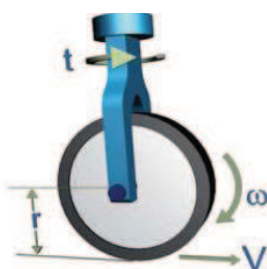


Figura 1.4 Orientación centrada[5]

**Orientación descentrada.** Tiene articulación de dirección, es decir, es orientable respecto a la estructura del robot móvil, no pasando su eje de dirección por el centro de rotación de la rueda.

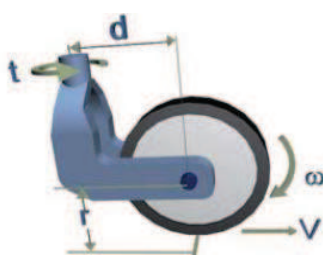


Figura 1.5 Orientación descentrada[5]

**Rueda sueca.** Es fija respecto a la estructura del robot móvil y posee rodillos entre la rueda y el suelo, con una determinada orientación fija respecto a la rueda.



Figura 1.6 Rueda sueca[5]

### 1.3.2.3 Centro instantáneo de rotación (ICR) o centro instantáneo de curvatura (ICC)

Se define como el punto por el cual cruzan los ejes de todas las ruedas; es el punto alrededor del cual el robot gira en un instante determinado. En la Figura 1.16 se muestra ICC para la configuración diferencial, ackerman y triciclo.

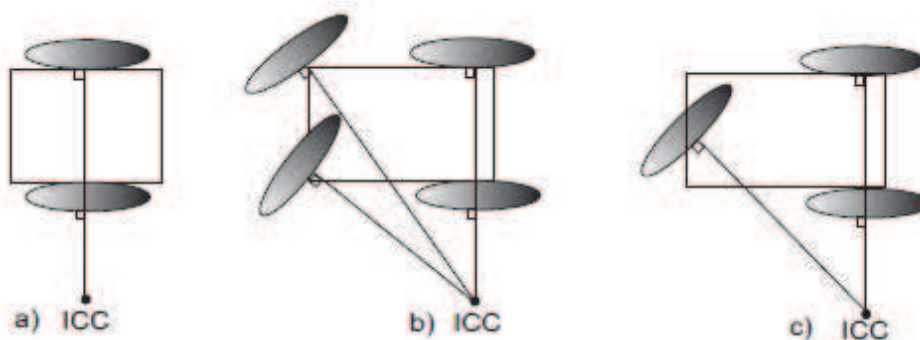


Figura 1.7 ICC en a) Configuración diferencial, b) Configuración ackerman, c) Configuración triciclo.[5]

### 1.3.3 CONFIGURACIONES TÍPICAS PARA ROBOTS MÓVILES CON RUEDAS

La configuración hace referencia a la forma en que se encuentran distribuidos los principales elementos que lo componen: plataformas, motores, ruedas. Las configuraciones para robots móviles con ruedas son diferencial, triciclo, ackerman, sincronizada, omnidireccional, con múltiples grados de libertad y movimiento mediante orugas.

#### 1.3.3.1 Configuración diferencial

Consta de dos ruedas colocadas en el eje perpendicular a la dirección del robot móvil. Cada rueda es controlada por un motor, de tal forma que el giro del robot móvil queda determinado por la diferencia de velocidad de las ruedas.

Uno de los problemas de esta configuración es mantener el equilibrio del robot móvil, para lo cual se le agregan ruedas de libre giro. Estas ruedas sirven para mantener horizontal al robot móvil y se orientan hacia la dirección del robot.

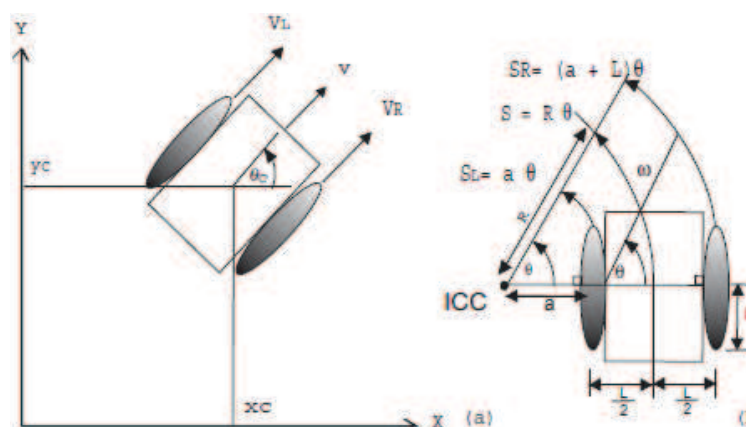


Figura 1.8 Modelo cinemático configuración diferencial[5]

Surge un problema cuando hay mas de tres apoyos en el robot movil, el cual es la perdida de tracción y graves errores en cálculos de odometría<sup>1</sup>.

### 1.3.3.2 Configuración triciclo

Consta de tres ruedas. Dos de las ruedas son pasivas y no estan acopladas a ningún motor, sirven de soporte. La dirección y tracción del robot son proporcionadas por la rueda delantera. En este tipo de configuración no se pueden realizar giros de 90 grados en cualquier dirección. El radio de curvatura es instantáneo, siempre se encuentra a lo largo del eje que cruza las dos ruedas traseras. Un problema que tiene la configuración triciclo es que el centro de gravedad del robot movil tiende a moverse lejos de la rueda delantera.

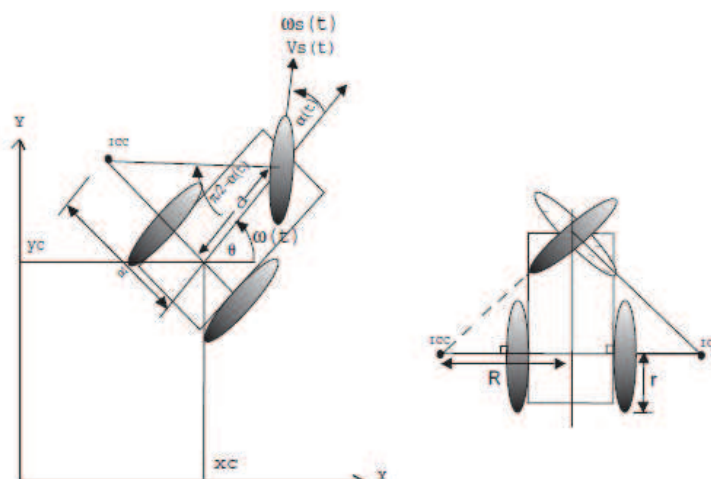


Figura 1.9 Modelo cinemático configuración triciclo.[5]

### 1.3.3.3 Configuración Ackerman

Posee dos ruedas traseras de tracción y dos ruedas delanteras para la dirección. Esta configuración esta creada para evitar el derrape<sup>2</sup> de las ruedas, lo que se

<sup>1</sup> Odometría es la técnica que se utiliza para calcular la distancia desplazada, midiendo el giro de las ruedas.

<sup>2</sup> Derrape es el deslizamiento que sufre un cuerpo móvil sobre una superficie sobre la cual no posee suficiente agarre.

consigue haciendo que la rueda delantera interior posea un ángulo  $\theta_i$  ligeramente mayor que el ángulo de la rueda exterior  $\theta_o$  (ver Figura 1.19) cuando el sistema se encuentra girando.

La configuración Ackerman, es usada en la industria automovilística y constituye un buen sistema de tracción, aplicable incluso para terrenos inclinados.

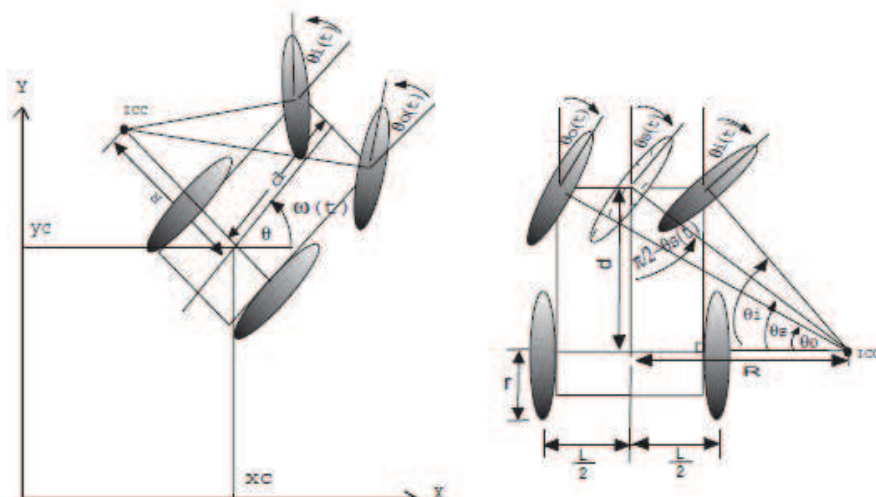


Figura 1.10 Modelo cinemático configuración Ackerman[5]

#### 1.3.3.4 Configuración sincronizada

Esta configuración consta de 3 o más ruedas, todas ellas acopladas mecánicamente y dotadas de tracción, de tal forma que todas rotan en la misma dirección y a la misma velocidad. La orientación de los ejes de rotación son controlables.

El ICC<sup>3</sup>, está siempre en el infinito, lo que simplifica el control. Cambiando la orientación de las ruedas, se puede manipular la dirección del ICC.

Este sistema mejora la odometría, reduciendo el deslizamiento de las ruedas, ya que todas las ruedas generan fuerzas con vectores de igual módulo y paralelos en todo momento.

<sup>3</sup> ICC siglas de Centro Instantáneo de Curvatura

La sincronización de las ruedas, se las realiza mecánicamente, aunque existen casos en que la sincronización es realizada electrónicamente, colocando un motor en cada rueda. El sistema sincronizado es holonómico<sup>4</sup>, posee dos motores uno para la dirección y otro para el movimiento. El robot no tiene la necesidad de girar para cambiar de dirección, ya que solo giran las ruedas.

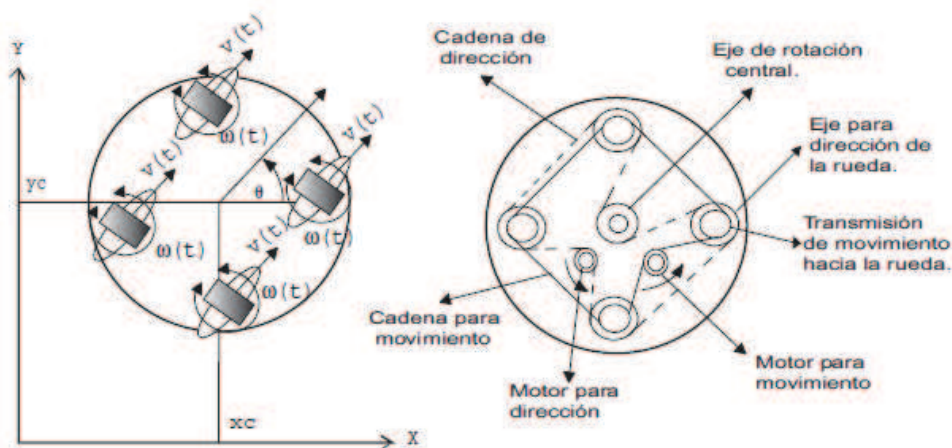


Figura 1.11 Modelo cinemático direccionamiento sincronizado[5]

### 1.3.3.5 Configuración omnidireccional

Este tipo de configuración está provista de ruedas omnidireccionales, lo que hace que los cálculos de odometría sean más complicados; pero el robot podrá moverse en cualquier dirección. La rueda omnidireccional se define como una rueda estándar, a la cual se le dota de una corona de rodillos. Los ejes de giro de los rodillos, son perpendiculares a la dirección normal de avance.

Al aplicarle una fuerza lateral, los rodillos giran sobre sí mismos y permiten que la componente de la velocidad en el eje x del robot móvil no sea nulo, y por tanto se elimina la restricción de no holomicidad.

<sup>4</sup> Holonómico es un sistema que es capaz de modificar su dirección instantáneamente sin necesidad de rotar previamente.

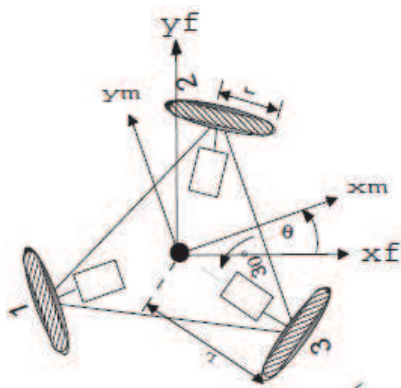


Figura 1.12 Modelo de la configuración omnidireccional[5]

### 1.3.3.6 Configuración con múltiples grados de libertad

Esta configuración tiene múltiples motores para la propulsión y dirección del vehículo, mejora mucho la movilidad aunque se dificulta el control. Esta configuración presenta un problema que es el incremento de deslizamiento en las ruedas, esto reduce la exactitud de la odometría. Es crítico coordinar el funcionamiento de todos los motores simultáneamente.

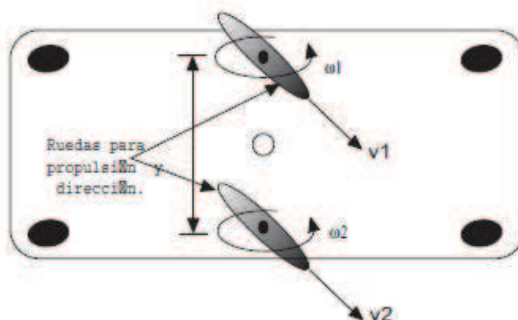


Figura 1.13 Vehículo con múltiples grados de libertad[5]

### 1.3.3.7 Tracción mediante orugas

En este tipo de configuración, las orugas sustituyen a las ruedas. Esta configuración es un caso especial de la configuración diferencial. El deslizamiento en los giros es muy grande, y se pierde exactitud en los cálculos de odometría. Es utilizado en lugares en donde el piso es muy irregular, y como no es posible saber

la posición exacta en la que se encuentra el robot son generalmente teleoperados. Son utilizados en búsqueda y rescate, levantamiento de bombas y en la industria nuclear.

Al poseer orugas el vehículo tiene una gran área de contacto, lo que impide hundimiento en la tierra suave y permite movilizarse por superficies irregulares. Para maniobrarlos se utiliza un direccionamiento mediante rodillos. La exactitud del direccionamiento viene a ser más dificultoso a altas velocidades, debido al deslizamiento y a la resistencia de las huellas desiguales, causadas por las fuerzas centrífugas.

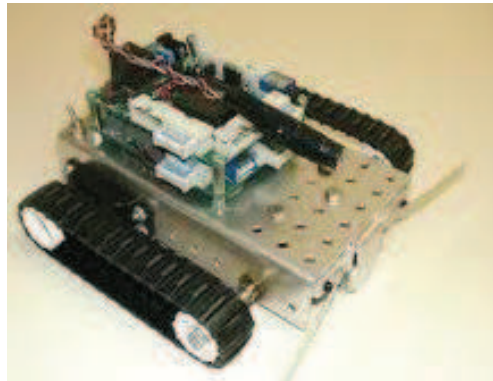


Figura 1.14 Tracción mediante orugas[6]

#### **1.4 ESTIMACIÓN DE LA POSICIÓN DE UN ROBOT MÓVIL[7]**

Un robot móvil consta de diversos subsistemas de locomoción, control de movimientos, percepción y planificación que interaccionan entre sí. El subsistema de percepción hace que el sistema sea capaz de interactuar con un entorno cambiante, así como poder reaccionar ante eventos inesperados, lo que exige la existencia de un sistema sensorial que suministre información sobre el entorno. Con la información recibida el robot realiza tres tareas fundamentales: estimar la posición y orientación, mantener actualizado el mapa del entorno y detectar posibles obstáculos.

Un robot móvil la mayoría de veces está equipado con más de un sensor para realizar sus tareas. Para que un robot pueda generar trayectorias y evitar



obstáculos tiene que ser capaz de determinar su localización con respecto a un sistema de referencia absoluto.

La estimación de la posición de un robot móvil viene dado por el tipo entorno, el conocimiento sobre el entorno, tipo de sensores que dispone y la tarea a realizar.

La mayoría de los robots móviles disponen de encoders, para detectar su movimiento, lo que les permite estimar a cada instante su posición, empleándose modelos de locomoción. Esta estimación no resulta muy conveniente para la mayoría de aplicaciones, ya que no es lo suficiente precisa. El problema es que los errores de estimación de la posición se van acumulando durante la navegación, por lo que se suelen usar sistemas de posicionamiento externo.



Figura 1.15 Sistemas para la estimación de la posición[7]

Para la estimación de la posición y orientación de un robot móvil autónomo se distinguen los estimadores explícitos y los estimadores basados en la percepción del entorno. Los estimadores explícitos proporcionan directamente la posición y orientación del robot por medio de medidas, sin tener que realizar procesamiento de información para la interpretación del entorno. Los estimadores basados en la percepción del entorno emplean sensores que suministran información del exterior, esta información se puede comparar con otros datos o modelos conocidos.

### **1.4.1 ESTIMACIÓN EXPLÍCITA DE LA POSICIÓN**

Se considera como estimación explícita a todos los sistemas capaces de estimar la posición sin que se realice una interpretación del entorno.

En los sistemas de estimación explícita se distingue la estimación basada en medidas internas y la estimación basada en estaciones de transmisión. La estimación basada en medidas internas trabaja solamente con sensores internos del robot móvil. Los sensores empleados pueden ser: giroscopios, encoders, detectores de norte, acelerómetros, tacómetros. La estimación basada en estaciones de transmisión se basa en el empleo de dos unidades, la unidad montada sobre el robot móvil y la unidad o unidades externas. La unidad montada sobre el robot móvil actúa como sensor receptor, mientras que las unidades externas actúan como emisores o transmisores de señales de referencia.

#### **1.4.1.1 Estimadores basados en medidas internas**

La posición y orientación puede obtenerse a partir de medidas internas como las vueltas que dan las ruedas, la velocidad, aceleración, cambios de orientación. Se distinguen entre estos los sistemas odométricos y los sistemas de navegación inercial.

##### *1.4.1.1.1 Sistemas odométricos*

La odometría se encarga de determinar la posición y orientación de un robot móvil a partir del número de pulsos obtenidos al momento que giran las ruedas. Se utilizan para ello codificadores ópticos de elevada precisión en al menos dos ruedas. La simplicidad y el bajo costo es una ventaja de la odometría, pero es necesario una calibración constante debido al desgaste y pérdida de presión de las ruedas o desajuste de los ejes. Esta técnica es vulnerable a las imperfecciones del suelo, al deslizamiento de las ruedas y a las variaciones de la

carcasa del robot móvil, aunque este último problema se puede corregir añadiendo un modelo para corregir las desviaciones introducidas. La idea fundamental de la odometría es la acumulación de los incrementos del movimiento en el tiempo, lo que produce una inevitable acumulación de errores. La acumulación de errores en la orientación causa grandes errores en la posición, los cuales se incrementan proporcionalmente con la distancia recorrida del robot. A pesar de las limitaciones la odometría es una parte importante de los sistemas de navegación.

#### 1.4.1.1.2 *Sistema de navegación inercial*

Este sistema obtiene la posición y orientación del robot móvil a partir de medidas de aceleración y ángulos de orientación. Para obtener la posición se integra la aceleración obteniendo la velocidad, la cual se integra para obtener la posición.

Para obtener la aceleración de los robots móviles se utilizan acelerómetros que suelen estar basados en sistemas pendulares, la precisión de los acelerómetros resulta crítica, ya que, los errores en la aceleración aunque sean pequeños afectan la posición, debido a la doble integración de la aceleración. Para medir la orientación se utilizan giroscopios e inclinómetros, también es posible medir el ángulo de orientación utilizando brújulas. Los sistemas de navegación inercial no son afectados por los problemas derivados de la interacción del robot móvil con el suelo, en este sistema se pueden corregir los efectos de las ondulaciones e irregularidades del terreno, haciendo en la práctica mucho más fiable y preciso que los sistemas basados en odometría, pero son más frágiles y caros.

#### **1.4.1.2 Estimadores basados en estaciones de transmisión**

Este tipo de sistema de posicionamiento es absoluto, se destaca por detectar la posición absoluta del robot móvil en un área suficientemente grande, sin necesidad de estructurar su entorno, lo que hace que el robot pueda moverse por escenarios muy diversos, recorriendo grandes distancias.

Estos sistemas constan de un receptor, que se encuentra colocado en el robot móvil, y de estaciones de transmisión de radio frecuencia RF. Dentro de este sistema se encuentran los sistemas de posicionamiento mediante estaciones fijas y los sistemas de posicionamiento mediante estaciones móviles.

Los sistemas de posicionamiento mediante estaciones fijas, se han empleado en la navegación de barcos, submarinos y aviones. Utilizan señales de radio de media y alta frecuencia que son emitidas por estaciones terrestres fijas.

Los sistemas de posicionamiento mediante estaciones móviles, son operados mediante satélites. Su utilización era en aplicaciones militares (misiles, aviación, submarinos) y en la navegación civil. Hoy en día se emplea un sistema similar pero mucho más potente denominado GPS (Global Positioning System), en el cual utilizando al menos 3 satélites, el receptor calcula por triangulación la altitud, latitud y altura del vehículo de forma instantánea y continua. También puede determinar la velocidad a partir del desplazamiento en frecuencias mediante el efecto Doppler.

#### **1.4.1.3 Estimación mediante balizas<sup>5</sup>**

Permite determinar la posición de un robot móvil en un entorno delimitado por balizas que se encuentran en posiciones conocidas. La posición del robot móvil es determinada de una forma mas o menos directa en base al principio de triangulación, a partir de medidas de distancias, ángulos o de la combinación de los dos.

La precisión y fiabilidad de este tipo de estimación va a depender del tipo de señal utilizada; infrarrojos, laser, radio, ultrasonidos, de las características del sensor y del número de balizas utilizadas. Las desventajas radican en la necesidad de configurar apropiadamente el entorno de trabajo, como de garantizar que el suficiente número de estas señales queden en todo momento libre de obstáculos y dentro del campo visual del sensor. Se debe tomar en cuenta, además, los

---

<sup>5</sup> Las balizas son objetos referenciales que permiten determinar la posición de algún objeto.

problemas que pueden originar las condiciones ambientales de iluminación y ruido, todo esto impide la utilización de esta técnica en entornos muy dinámicos o no estructurados.

#### **1.4.2 ESTIMACIÓN MEDIANTE PERCEPCIÓN DEL ENTORNO**

Consiste en dotar al robot móvil de un sistema sensorial capaz de proporcionar suficiente información del entorno, y con esta información de forma autónoma determinar la localización del robot móvil.

El sistema sensorial puede operar en base a diversos tipos de sensores; cámara CCD<sup>6</sup>, sonares, escáner, laser, etc. y seleccionando determinados tipos de datos u objetos a partir de un conjunto de información adquirida; marcas naturales, puntos de interés, entorno completo percibido. En cualquier caso, la localización del robot móvil se determina a través del emparejamiento de los datos extraídos del entorno por el sistema sensorial, con los datos previamente conocidos del entorno.

Los sensores utilizados para la navegación de un robot móvil pueden dividirse en dos grupos: activos y pasivos. Los sensores activos son aquellos que emiten algún tipo de energía al medio, luz infrarroja, ultrasonidos, luz láser, ondas de radio etc.

Los sensores denominados pasivos se limitan a captar la energía que existe en el medio, de este tipo son las cámaras CCD, las cuales perciben el entorno a través de la cantidad de luz que les llegan procedente de las fuentes luminosas o bien a través de reflexiones en los objetos del entorno.

---

<sup>6</sup> CCD son las siglas de Dispositivo de Carga Acoplada.



Figura 1.16 Robot móvil autónomo con sistema de sonares[5]

## **1.5 ALGORITMOS DE RECORRIDO DENTRO DE UN LABERINTO**

Si bien los algoritmos para este proyecto de titulación serán desarrollados por propia cuenta, se deben considerar los diferentes tipos de algoritmos que se tienen hasta el momento, tales como:

### **1.5.1 ALGORITMOS DE LA MANO DERECHA Y LA MANO IZQUIERDA[8]**

Estos algoritmos se basan en seguir una pared del laberinto (izquierda o derecha) y resultan efectivos en casi todos los casos, se conozca o no el laberinto. De esta forma se garantiza que la exploración realizada por el minirobot sea adecuada, el minirobot pasa por lo menos una vez por un camino del laberinto, y además lo hace de una forma ordenada, ya que siempre va a tomar las mismas decisiones para su desplazamiento.

Sus principales desventajas, radican en que no resuelven laberintos no conectados, si no se empieza desde una de las paredes exteriores del laberinto y por otro lado la trayectoria que siguen para encontrar la salida dista mucho de ser óptima, ya que recorre todas las opciones hasta encontrar la salida.

### **1.5.2 ALEATORIOS[8]**

Estos algoritmos pretenden llevar a cabo la exploración de manera aleatoria, es decir, el minirobot puede desplazarse a la derecha, o la izquierda, adelante, o atrás, en esta exploración no se tienen ningún tipo de inteligencia para mover el minirobot, solo se trata de avanzar en cualquier dirección hasta que la distancia se haya alcanzado o hasta que exista alguna colisión. Estos algoritmos son muy fáciles de implementar, pero tienen muy poca efectividad ya que el procesamiento de la información obtenida es muy poco o nulo, además de que desde un principio el minirobot puede navegar en dirección contraria a la que uno desearía y esto provocaría mayor tiempo de espera para llegar al objetivo.

### **1.5.3 RUTA PREDETERMINADA[8]**

Aquí se pretende presentar una ruta inicial a seguir, la cual se interrumpirá cuando el robot encuentre problemas para seguir avanzando y se procederá a realizar otra ruta a partir del punto en el cual se haya quedado el robot. Esto se hace sucesivamente hasta terminar con la exploración o la búsqueda. Este tipo de algoritmos suele tener mayor efectividad ya que la ruta predeterminada se puede orientar hacia el punto que se tenga como objetivo de tal forma que siempre se trate de ir línea recta hacia él.

### **1.5.4 PROCESAMIENTO DE INFORMACIÓN[8]**

Este tipo de algoritmos es el más efectivo, aquí se toman en cuenta todos los datos recabados hasta el momento. Desde el principio se trata de alcanzar el objetivo directamente como en el caso de una ruta predeterminada, después al encontrar obstáculos se hace una replaneación de la ruta tomando en cuenta los obstáculos encontrados, esta replaneación se lleva a cabo gracias a que se tienen nuevos datos acerca del área en la cual se realiza la búsqueda – exploración y se aprovechan para crear nuevas rutas que estén libres de colisión al menos con los

obstáculos hasta el momento encontrados. De esta forma se tomará como base un algoritmo que procese la información obtenida o que se tenga en el principio para dictar una ruta apropiada al minirobot, la planeación se llevará a cabo con la ayuda de un mapa que se irá generando conforme avance el minirobot o bien haya sido proporcionado con anterioridad.

El algoritmo funcionara bajo dos circunstancias, una para la búsqueda – exploración del área sin previo conocimiento y otra para la búsqueda – exploración mediante el procesamiento de la información obtenida a través de la exploración realizada anteriormente o con la ayuda de un mapa con que se cuente sobre el área. La exploración consiste en conocer la mayor parte posible de un ambiente del cual no se tiene ningún conocimiento. La búsqueda consiste en localizar un objetivo situado en el área, la cual puede o no haber sido explorada con anterioridad y el objetivo puede estar situado en cualquier lugar sin saber exactamente donde está.

### **1.5.5 ALGORITMO DE RECORRIDO RECURSIVO[8]**

Este algoritmo se basa en el recorrido de un camino, presentando fallo cuando se encuentra una pared y éxito si se encuentra una salida, en cualquier otro caso, se intenta recorrer de manera recursiva en cualquier de las 4 direcciones posibles. Cuando se intenta una nueva dirección se traza una línea y se borra una línea cuando se encuentra fallo. Este método encontrara siempre una salida, si es que existe, pero en la mayoría de los casos, no serán el camino más corto, y requiere recorrer una gran parte del laberinto para salir de él, además del hecho que de las 2 posibles salidas existentes en el laberinto, solo una de ellas es válida y no existe manera de descartar la otra utilizando este algoritmo.

### **1.5.6 ALGORITMO DE TARRY[8]**

Este método publicado por Tarry en 1895, crea un camino cíclico en el laberinto, pasando por cada cruce una y solo una vez en cada dirección. Iniciando en una



posición A, selecciona un camino cualquiera y lo marca de manera especial cuando llega al siguiente cruce, puede haber sido visitado o no, y lo marca acorde con ello. El algoritmo termina en el lugar de inicio, habiendo recorrido cada cruce, por lo menos una vez en cada dirección.

### **1.5.7 ALGORITMO DE PLEDGE[8]**

Consiste en mantener la dirección en que se está apuntando en cada momento, a pesar de su simplicidad, al igual que los algoritmos de la mano izquierda y derecha, no es aplicable a todo tipo de laberintos, puesto que en ciertos casos, el minirobot quedará ciclado en un solo camino.

### **1.5.8 ALGORITMO PLANTEADO POR LEE[8]**

Consta de dos partes, primero la creación del mapa del laberinto en memoria, por medio de recorridos sucesivos, durante estos recorridos, se anotan las paredes que tienen cada celda conforme el minirobot va avanzando, así, si la celda solo tuviera pared al norte, se le asigna un 8, pero si además tiene también pared al este, se le asigna un 10, 8 de la pared norte y 2 de la pared este. Se puede apreciar que se requiere una cantidad de memoria, de tamaño suficiente para, por lo menos, almacenar un arreglo con la información de cada celda del laberinto. Una vez que se tiene un mapa del laberinto, se asigna un número a cada celda, de la siguiente manera: se inicia colocando un 0 en cada una de las celdas del laberinto que se consideren como meta, se debe recorrer el laberinto columna por columna en repetidas ocasiones, para cada celda en la que esté el robot. Se busca en el mapa si dicha celda aun no está numerada y si las celdas a las que se puede llegar a partir de la actual ya se encuentran numeradas, si no es así, en esta visita no se hace nada, en caso de que por lo menos algunas de las celdas vecinas ya se encuentren numeradas, y la actual no, suponiendo que el número de Lee más pequeño de las celdas vecinas sea  $n$ , a la celda actual se le asigna

$n+1$ , este proceso se repite hasta que ya no pueda realizar cambios en el mapa, lo que supondrá que ya todas las celdas se encuentran numeradas.

Lo anterior supone que ya se tienen en memoria el mapa del laberinto, de no ser así, se puede modificar la generación de números, suponiendo que el laberinto solo tiene paredes en los extremos, y calculando los números, al cambiar de una celda a la siguiente, se verifica si en verdad no existen más paredes, en caso contrario, se actualiza el mapa y se recalcula los números. Una vez generados los números es fácil encontrar el camino dentro del laberinto. Y además se asegura que será siempre el camino más corto, puesto que estando en cualquier celda del laberinto basta con moverse a la celda vecina que contenga un número menor.

## **1.6 COOPERATIVISMO[9]**

En la actualidad, en el campo de la robótica se está produciendo un gran desarrollo en el campo de los robots cooperativos. Varios problemas son muy complicados para que los pueda resolver un solo robot, como mover algún objeto voluminoso, el manejo de un material peligroso, la exploración y cobertura de un campo.

La robótica colectiva busca diseñar sistemas compuestos de varios robots con el fin de resolver un problema conjuntamente. Los robots que forman parte de un sistema multirobot son sencillos en términos de diseño y control, lo que hace que sean menos costosos que sistemas de un solo robot especializado, además son adecuados para resolver un problema donde un solo robot es insuficiente.

### **1.6.1 TIPOS DE CONTROL PARA SISTEMAS COOPERATIVOS**

#### **1.6.1.1 Control centralizado**

Este tipo de control tiene una unidad central la cual es la encargada de la toma de decisiones. Los robots adquieren información a través de sus sensores, esta

información es enviada a la unidad central, que se encarga de tomar decisiones. La principal ventaja de este sistema es la eficiencia al obtener soluciones, debido a que la unidad central recibe mucha información.

Las desventajas pueden darse por congestión al recibir una gran cantidad de información de los robots y hacerse lento al momento de la toma de decisiones, además si la unidad central falla o deja de funcionar, deja de funcionar todo el sistema, ya que los robots pierden la capacidad de decidir.

### **1.6.1.2 Control distribuido**

En este caso cada robot puede decidir qué hacer por su cuenta, sin recibir órdenes de ninguna otra entidad. Este tipo de control es muy dinámico y rápido, pues no es necesario ni el envío ni el almacenamiento de información.

Si un robot deja de funcionar, todos los demás pueden seguir funcionando independientemente. Es difícil diseñar los algoritmos de programación, ya que tienen que ser sencillos y funcionar correctamente.

## **1.6.2 TIPOS DE SISTEMAS**

### **1.6.2.1 Muy acoplados o colectivos**

Todos los robots tienen su autonomía y además tienen la capacidad de comunicarse, y por lo tanto, de cooperación y coordinación. La inteligencia colectiva del sistema es la suma de la inteligencia de cada robot.

Cada robot debe tener la capacidad de transferir información a otro compañero y tener la suficiente inteligencia para pedir ayuda y coordinar movimientos con los demás individuos.

### 1.6.2.2 Poco acoplados o manadas

Se trata de sistemas en los que los robots funcionan independientemente del resto, sin tener en cuenta los movimientos o decisiones de los demás. Todo el control es distribuido, y cada robot se rige por reglas.

### 1.6.3 VENTAJAS DEL USO DE ROBOTS COOPERATIVOS

**Acciones mucho más rápidas.-** Al tener un solo robot, éste se tendría que encargar de realizar todas las tareas, reconocer el medio, enviar información, toma de decisiones, control de sus movimientos. Al dividir las tareas entre varios robots, el trabajo se realiza de una manera más rápida y eficiente.

**Tolerancia a fallos.-** Un fallo en el funcionamiento de uno de los robots puede ser subsanado por el resto del equipo.

### 1.6.4 INCONVENIENTES EN EL USO DE ROBOTS COOPERATIVOS

**Interferencia entre robots.-** Dependiendo de la programación y el control de los distintos robots que forman un sistema, la interferencia puede ser un problema, golpes, choques, distracciones.

**Coste de la comunicación.-** Los robots deben mantener una comunicación fluida entre ellos o con la unidad central, lo que complica su diseño, aumenta el peso y el consumo y provoca interferencia con otros sistemas de comunicación o con los sensores.

**Incertidumbre respecto a los otros robots.-** El sistema pierde eficacia cuando uno de los robots realiza un trabajo que ya ha sido realizado por un compañero suyo, o dejar de realizar una tarea pensando que ya ha sido realizada.

## 1.7 COMUNICACIÓN POR RADIO FRECUENCIA[10]

Una de las maneras más prácticas de trabajar con un robot móvil a distancia es programar sus funciones en una computadora y comandar al robot de manera remota por medio de algún enlace, este enlace en el caso de este proyecto de titulación es la comunicación con radiofrecuencia mediante la tecnología zigbee.

### 1.7.1 TECNOLOGÍA ZIGBEE

Zigbee es un conjunto de especificaciones basadas en el estándar IEEE 802.15.4 que define una serie de protocolos de comunicación a una velocidad de transmisión baja para redes inalámbricas personales de corto alcance y bajo consumo.

Zigbee opera a distintas velocidades dependiendo del lugar donde se encuentre, en Europa opera a 868 MHz, en Estados Unidos opera a 915 MHz y en el resto del mundo opera a 2,4 GHz.

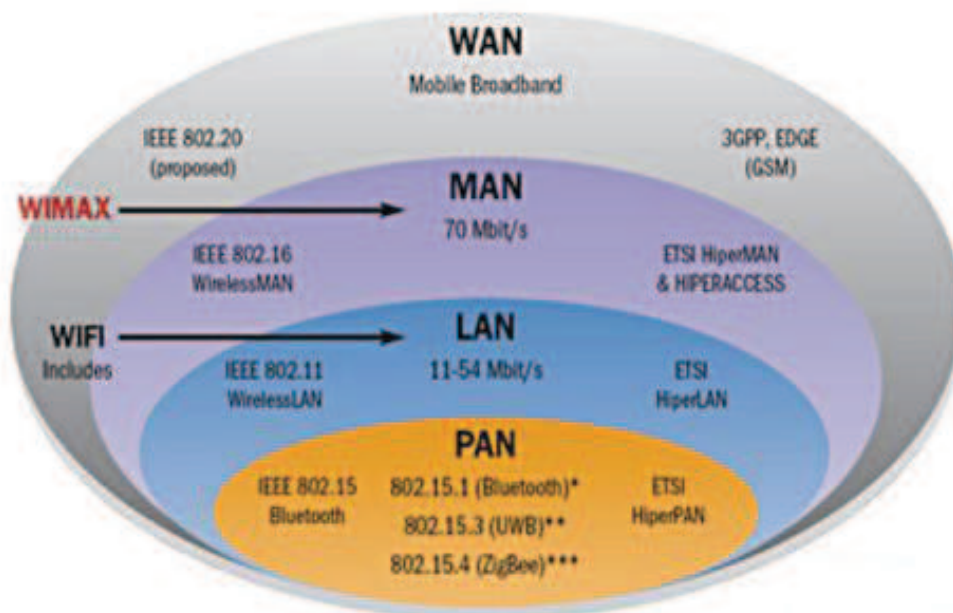


Figura 1.17 Tecnología en 2,4 GHz[11]

### 1.7.2 DISPOSITIVOS ZIGBEE

Para establecer una red Zigbee se tiene 3 dispositivos que son:

- **Coordinador:** en una red zigbee se debe tener uno y solo un coordinador, que es el encargado de controlar la red y enrutar los caminos hacia los demás dispositivos de la red.
- **Router:** es el encargado de interconectar los dispositivos que se encuentran separados en la red hacia el coordinador.
- **End device:** es el dispositivo final donde se encuentran los sensores o actuadores de la red, puede comunicarse directamente con el coordinador o con el router, pero no con otros end devices, puede estar dormido la mayor parte del tiempo ahorrando así energía, y aumentando la vida media de sus baterías.

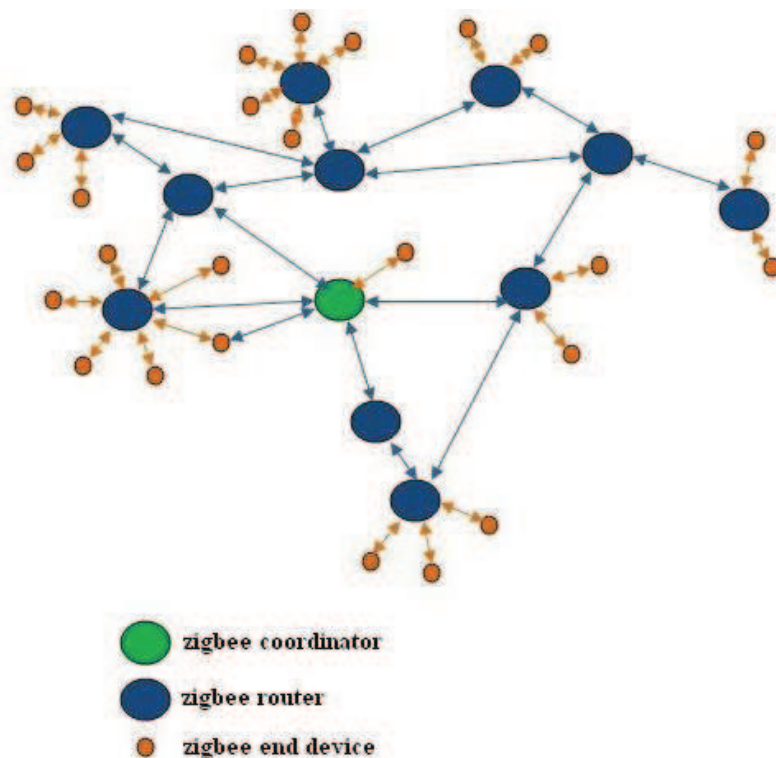


Figura 1.18 Dispositivos Zigbee[12]

### 1.7.3 TOPOLOGÍAS ZIGBEE

Zigbee permite realizar tres tipos de topologías de red:

- Topología en estrella; cuando el coordinador se encuentra en el centro de la red.
- Topología en árbol; cuando el coordinador se encuentra en la raíz de la red, de donde se ramifican los demás nodos.
- Topología en malla o mesh; cuando el coordinador y los nodos se pueden comunicar entre sí.

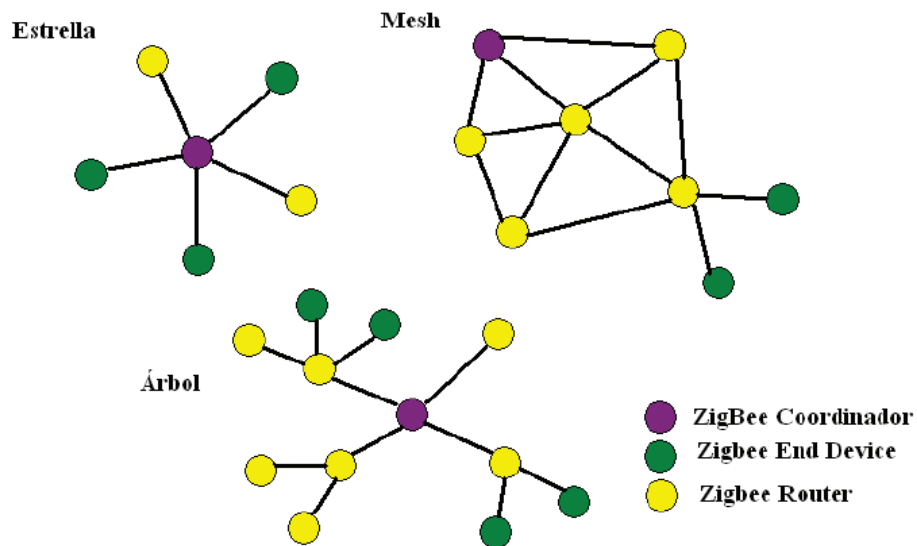


Figura 1.19 Topologías de red Zigbee[13]

## **CAPITULO 2**

### **DISEÑO Y CONSTRUCCIÓN DE LAS MINI PLATAFORMAS ROBÓTICAS**

El diseño y construcción de las mini plataformas robóticas fue realizado en base a criterios establecidos por conocimientos previos, por las dimensiones de los elementos que se van a utilizar, por cumplir los objetivos planteados y con el fin de realizar una estructura pequeña y adecuada para que el minirobot pueda desplazarse dentro de un laberinto.

#### **2.1 DESCRIPCIÓN DE LOS MINIROBOTS EXPLORADORES COOPERATIVOS**

Los minirobots exploradores cooperativos se desplazan a través de un laberinto de superficie regular, evitando chocarse con las paredes y reconociendo las rutas exploradas, con el fin de encontrar el punto de llegada que será la meta. Los tres minirobots exploran diferentes sectores del laberinto con el fin de mapear el laberinto en el menor tiempo posible. Cada minirobot recoge datos de su exploración y los envía a un dispositivo coordinador en un computador. Los datos son procesados con la ayuda computacional MATLAB, por medio de una interfaz gráfica donde se visualiza las rutas recorridas por los minirobots y la exploración de su entorno de una forma aproximada.

Los minirobots son autómatas capaces de desplazarse en diferentes direcciones dentro del laberinto, para ello utilizan tracción diferencial para el cambio de dirección y motores de corriente continua para la rotación de las ruedas. Para percibir su entorno utiliza sensores de distancia por ultrasonido, colocados en distintas posiciones dentro del minirobot, para evitar colisiones, tener datos de las paredes del laberinto que se utilizan para el algoritmo de desplazamiento y para encontrar nuevas rutas.



El control de los minirobots autómatas es por medio de un Arduino que es una placa ensamblada que tiene un microcontrolador ATMEGA 328P. Para el control de la velocidad y sentido de giro de los motores se utiliza un driver, el cual es controlado por señales PWM digitales entregadas por el Arduino, con las cuales se controla la velocidad angular de los motores, la habilitación de los mismos y el sentido de giro. Los minirobots cuentan con un sistema de encoders de cuadratura para tomar datos de velocidad de las ruedas, la cual sirve además para hacer cálculos de odometría.

Los robots se comunican con la PC donde se indican las rutas recorridas y la detección de las paredes del laberinto, para la comunicación de los robots se utiliza radiofrecuencia; la placa Arduino tiene la ventaja de ser compatible con drivers de comunicación entre dispositivos vía bluetooth o radiofrecuencia. En este caso se utilizan los módulos XBEE que son compatibles con el Arduino y que utilizan el protocolo Zigbee para la comunicación por radiofrecuencia.

## 2.1.1 COMPONENTES UTILIZADOS

### 2.1.1.1 Arduino Duemilanove[14]

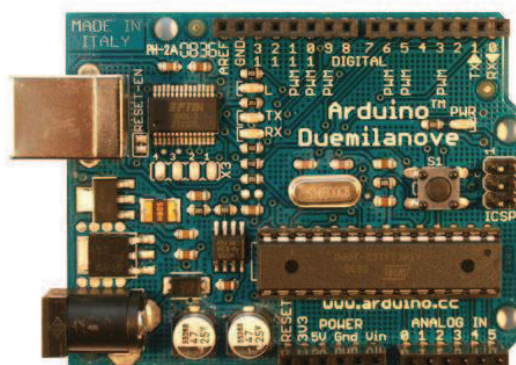


Figura 2.1 Arduino Duemilanove[14]

Un Arduino Duemilanove es una placa basada en el microcontrolador ATMEGA 328P, y es un producto de software libre, el cual consta de las siguientes características:

- Voltaje de funcionamiento: 5 V
- Voltaje de entrada: 7 – 12 V
- Voltaje de entrada (límite): 6 – 20 V
- Pines E/S digitales: 14 (6 proporcionan salida PWM)
- Pines de entrada analógica: 6
- Corriente por pin: 40 mA
- Corriente en pin 3.3 v: 50 mA
- Memoria flash: 32 KB de las cuales 2 KB se usa en el gestor de arranque
- SRAM: 2 KB
- EEPROM: 1 KB
- Frecuencia: 16 MHz

#### 2.1.1.1.1 Alimentación[14]

La alimentación del Arduino puede ser vía USB o con una fuente de alimentación o batería. La batería se puede conectar mediante un conector de 2.1 mm, con el centro positivo en el conector hembra de la placa. La placa puede trabajar con una alimentación externa de 6 a 20 V como límites mínimo y máximo respectivamente, pero el voltaje recomendado es entre 7 a 12 V.

La descripción de los pines de alimentación es la siguiente:

- **Vin:** es el voltaje de entrada a la placa cuando no se está utilizando los 5 V de la conexión USB. Se puede proporcionar voltaje mediante este pin, o, si ya existe el voltaje, se puede acceder al mismo mediante este mismo pin.
- **5 V:** es el voltaje ya regulado de la placa, que alimenta al microcontrolador y a los demás componentes, si se requiere un voltaje regulado de 5 V, se puede acceder mediante este pin.

- **3V3:** es una fuente de 3.3 V que provienen del chip FTDI<sup>7</sup> integrada en la placa.
- **GND:** pin de puesta a tierra de la placa.

#### 2.1.1.1.2 Descripción de pines Entrada/Salida[14]

La placa Arduino consta de los siguientes pines:

- **COMUNICACIÓN SERIAL:** para esta operación de la placa son utilizados los pines 0 y 1, para la recepción y transmisión de datos respectivamente, mediante el puerto serie TTL.
- **INTERRUPCIONES EXTERNAS:** para esto se utilizan los pines 2 y 3. Estos pines se pueden configurar para activar la interrupción en un valor alto (1L) o en un valor bajo (0L).
- **PWM:** la placa consta de 6 salidas de tipo PWM<sup>8</sup>, q son los pines 3, 5, 6, 9, 10 y 11. Proporcionan la señal PWM con 8 bits de resolución.
- **SPI<sup>9</sup>:** estos pines son utilizados para la comunicación SPI, los cuales están distribuidos, pin 10 (SS), pin 11 (MOSI), pin 12 (MISO), pin 13 (SCK).
- **6 PINES CON ENTRADAS ANALÓGICAS:** las entradas analógicas proporcionan una resolución de 10 bits, por defecto se mide de tierra a 5 voltios.
- **I2C:** estos pines son utilizados para la comunicación I2C, los cuales están distribuidos, pin 4 (SDA) y pin 5 (SCL).
- **AREF:** es utilizado como voltaje de referencia para las entradas analógicas.

---

<sup>7</sup> FTDI son las siglas de Dispositivo Internacional con Tecnología del Futuro

<sup>8</sup> PWM son las siglas de Modulación de Ancho de Pulso

<sup>9</sup> SPI por sus siglas en inglés de Serial Peripheral Interface, es un estándar de comunicaciones

#### 2.1.1.1.3 Programación[14]

La placa Arduino se puede programar a través del software Arduino. El cual se lo puede descargar desde la página principal de Arduino.cc.

El ATMEGA328P en la placa Arduino Duemilanove viene precargado con un gestor de arranque que permite cargar nuevo código sin necesidad de un programador por hardware externo. Se comunica utilizando el protocolo STK500 original. También se puede saltar el gestor de arranque y programar directamente el microcontrolador a través del puerto ISCP<sup>10</sup>.

#### 2.1.1.1.4 Comunicación[14]

La placa Arduino Duemilanove facilita la comunicación con la PC o con otros arduinos e incluso otros microcontroladores. El ATMEGA328P proporciona comunicación serial mediante el UART<sup>11</sup> TTL (5V), disponible a través de los pines digitales pin 0 (Recepción Rx) y pin 1 (Transmisión Tx). Un chip FTDI FT232RL integrado en la placa canaliza esta comunicación serial a través del USB y los drivers FTDI proporcionan un puerto serial virtual en la PC.

Los leds Rx y Tx de la placa parpadearan cuando se detecte comunicación transmitida través del chip FTDI y la conexión USB (no parpadearan si se usa la comunicación serial a través de los pines 0 y 1).

### 2.1.1.2 Módulos Arduino XBEE SHIELD

Los módulos “Arduino Xbee Shield”, son placas que están preparadas para ser utilizadas en la comunicación serial del Arduino.

---

<sup>10</sup> ISCP son las siglas de Programador Serial en Circuito

<sup>11</sup> UART son las siglas de Transmisión y Recepción Asíncrona Universal

Poseen un espacio donde se colocara la antena Xbee, dependiendo del alcance que se necesite para determinado proyecto, la antena Xbee podrá tener menor o mayor potencia, lo cual hará que su tamaño varié. Además, este modulo permite que el Arduino se comuniqué de forma inalámbrica mediante la radiofrecuencia.

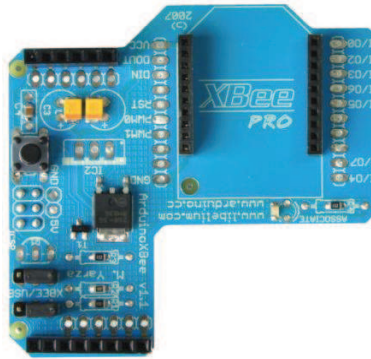


Figura 2.2 Arduino Xbee Shield[15]

### 2.1.1.3 Módulos XBEE Serie 2

Los módulos Xbee serie 2 proporciona interoperabilidad con dispositivos ZigBee de otros fabricantes, este módulo sólo es compatible con otros módulos Xbee Serie 2, no es compatible con la Xbee serie 1.



Figura 2.3 Módulo Xbee Serie 2[16]

#### 2.1.1.3.1 Características

- Es compatible sólo con la serie 2 de la familia

- Distancia de comunicación poco más larga: 400 pies (122 metros) línea de vista para el módulo de 1,25 mW y hasta 2 millas (3.2 Km) de línea de vista para el módulo de 60 mW
- Velocidad máxima de transmisión de hasta 1 Mbps
- Poco menor consumo de energía que la Serie 1
- Lleva 2 entradas A/D menos y dos E / S digitales más que la serie 1

#### 2.1.1.3.2 *Especificaciones Claves*

- Interfaz de datos con velocidad de 1200 bps - 1 Mbps
- Banda de 2,4 GHz de frecuencia (aceptada en todo el mundo)
- Rango de temperatura de funcionamiento industrial (-40C a 85C)
- Potencia de transmisión de 1,25 mW (1 dBm)
- Funciona con voltaje de 2,1 a 3,6 VCC; corriente de transmisión de 35 mA, corriente de recepción de 38 mA
- Corriente en power-down <1 uA
- Nivel de interfaz UART de 3.3V CMOS

#### 2.1.1.4 **Sensores de distancia por ultrasonido HC - SR04[17]**

Para la detección de obstáculos y percepción del entorno se utiliza tres sensores de distancia por ultrasonido, en la parte frontal de los minirobots, los sensores se ubicaron de tal manera que los minirobots no solo detecten los obstáculos que se encuentren al frente sino obstáculos que estén a los costados de la línea frontal que siguen los minirobots al desplazarse.

##### 2.1.1.4.1 *Funcionamiento*

Los sensores de distancia por ultrasonido utilizan ondas de sonido para su funcionamiento, a una frecuencia de 40kHz por lo que es imperceptible al oído humano, los sonidos que escuchamos están en el rango de 16Hz a 20KHz.

El sensor por ultrasonido emite un haz de ondas ultrasónicas, las cuales rebotan al chocar con un objeto y son receptadas por el mismo sensor. Se establece la distancia desde el sensor hacia objeto midiendo el tiempo que transcurre desde que se emite las ondas ultrasónicas hasta recibir el eco de estas ondas. La distancia se establece tomando en cuenta la velocidad del sonido y el tiempo medido.

Por software y sabiendo que la distancia es igual a la velocidad del sonido por el tiempo medido, se encuentra la distancia entre el sensor y el objeto, tomando en cuenta que el tiempo medido es el doble, tiempo de ida y de vuelta de la onda ultrasónica.



Figura 2.4 Sensor de ultrasonido HC - SR04[17]

El sensor de ultrasonido HC-SR04 tiene mejores resultados midiendo en un rango de 0 a 30 grados, como se muestra en la siguiente figura:

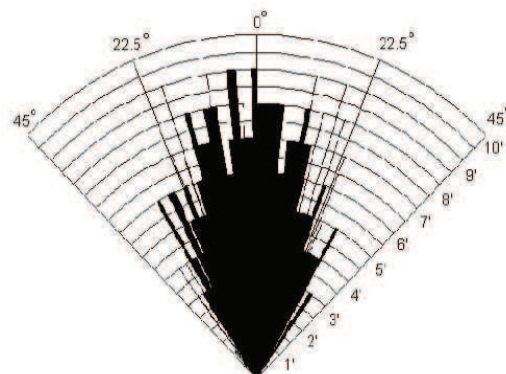


Figura 2.5 Test práctico de rendimiento[17]

#### 2.1.1.4.2 Modo de operación

Este modo emplea un pin para aplicar el pulso de inicio o Trigger y otro para leer la anchura del pulso del ECO medido. Tal y como se muestra en el diagrama de tiempos de la Figura 2.7, el modo de empleo es muy sencillo.

Externamente se aplica, por parte del usuario, un pulso de disparo o trigger de 10  $\mu$ S de duración mínima. Se inicia la secuencia. El módulo transmite un tren de pulsos o "burst" de 8 ciclos a 40KHz. En ese momento la señal de salida ECO pasa a nivel "1". Cuando la cápsula receptora recibe la señal transmitida como consecuencia de haber rebotado en un objeto (eco), esta salida pasa de nuevo a nivel "0". El usuario debe medir la duración del pulso de esta señal, es decir, el tiempo en que la señal eco se mantiene a "1".

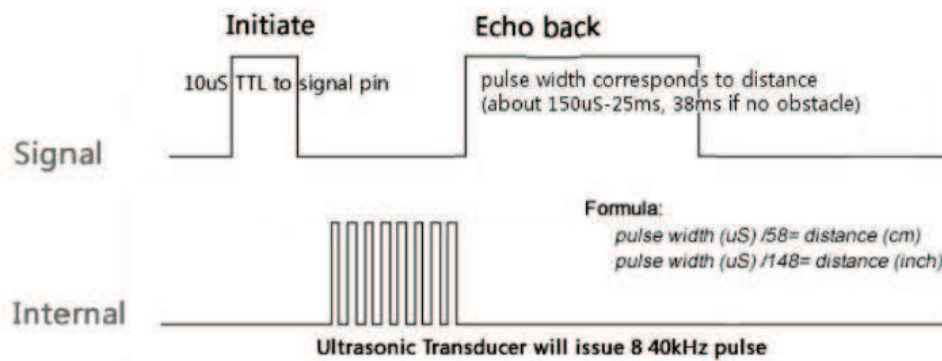


Figura 2.6 Diagrama de tiempos del funcionamiento del sensor de distancia por ultrasonido[17]

Con objeto de que el módulo se estabilice, ya que existen tres sensores ultrasónicos en el minirobot, se debe dejar un lapsus de tiempo de unos 10ms mínimo entre el momento en que la señal de eco pasa a "0" y un nuevo pulso de disparo que inicie el siguiente ciclo o medida. Esto permite realizar medidas cada 25ms o lo que es igual a 40 medidas por segundo.

La duración del pulso eco de salida varía entre 100 $\mu$ s y 25ms, en función de la distancia entre las cápsulas del módulo y el objeto. La velocidad del sonido es de



29.15  $\mu\text{s}/\text{cm}$  que, como realiza un recorrido de ida y vuelta, queda establecida en 58.30 $\mu\text{s}/\text{cm}$ .

Así pues el rango mínimo que se puede medir es de 1.7 cm (100 $\mu\text{s}/58$ ) y el máximo de 431 cm (25ms/58).

#### 2.1.1.4.3 *Especificaciones*

- Fuente de alimentación: 5 V
- Corriente de trabajo: 15 mA
- Corriente en reposo: < 2 mA
- Angulo efectivo: < 15 grados
- Distancia de alcance: 2 cm – 400 cm
- Resolución: 0,3 cm

#### 2.1.1.4.4 *Pines de conexión*

- Vcc
- Trigger (Disparo de ultrasonido)
- Echo (Recepción de ultrasonido)
- Gnd

#### 2.1.1.5 **Módulo L298N para los micromotores DC**

El módulo para el manejo de los micromotores DC tanto para su velocidad como para su dirección de giro contiene el integrado L298N o también llamado puente H. Este integrado es utilizado cuando la potencia consumida por los motores es baja o media.

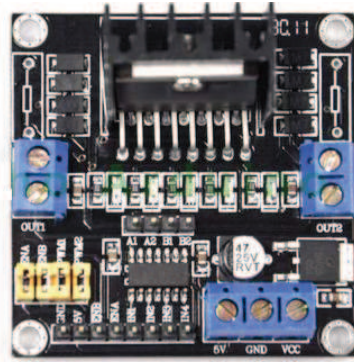


Figura 2.7 Módulo L298N Vista Superior

Las características principales son:

- Voltaje de funcionamiento desde 5 V hasta 35 V.
- Corriente máxima de 2 A.
- Voltaje terminal lógico de 5 V hasta 7 V.
- Voltaje de salida desde 1,5 V hasta el valor del voltaje terminal lógico.
- Máxima potencia de 20 W.
- Peso del módulo de 33 g.

#### 2.1.1.6 Micromotores DC

Los elementos utilizados para la movilidad de los minirobots son los micromotores con caja reductora metálica, estos pequeños micromotores están diseñados para funcionar con 6 V, que es su voltaje nominal; este tipo de motores pueden funcionar a tensiones por encima y debajo de este voltaje nominal, por lo que pueden operar cómodamente entre 3V a 9 V, (puede existir rotación del eje con valores de voltaje tan bajos como 0.5 V).

Para el control de los micromotores se utiliza la técnica conocida como PWM o modulación de ancho de pulso.



Figura 2.8 Micromotor DC[18]

Las características principales son: [18]

- Voltaje nominal de 6 V.
- Velocidad angular de 150 RPM a 6 V.
- Corriente que consume en vacío de 40 mA.
- Torque que otorga es de 1,7 Kg-cm
- Máxima corriente que consume con carga completa es de 700 mA.
- Relación de la caja reductora 150:1.

#### 2.1.1.7 Encoder de cuadratura

Este encoder de cuadratura está diseñado para ser usado con las ruedas todo terreno para micromotor, cuenta con dos sensores infrarrojos reflectantes los cuales leen las ranuras que tiene la rueda todo terreno; lo cual permite tener dos señales tipo tren de pulsos A y B por cada uno de sus dos canales desfasadas 90°.

La rueda posee 12 dientes, y el encoder en su funcionamiento en cuadratura puede tener una resolución máxima de 48 pulsos por vuelta, además la configuración en cuadratura permite leer la dirección de movimiento de la rueda. Cada sensor es análogo, y la señal se digitaliza por medio de un comparador para obtener salidas digitales, el encoder está calibrado para funcionar a 4.5 V pero puede re-calibrarse para 3.3 V [19].



Figura 2.9 Izquierda) Encoder de cuadratura, derecha) Encoder de cuadratura, rueda todo terreno, micromotor y soporte.[19]

Características:[19]

- Voltaje de Operación 4.5 V a 5.5 V
- Salida en cuadratura
- Corriente 14 mA a 5V
- Resolución 48 pulsos por vuelta

#### 2.1.1.8 Baterías de Litio – Polímero (LiPo)

Las baterías que se utilizan en los tres minirobots son las denominadas LiPo, estas baterías tienen la principal característica de tener una tasa de descarga grande, esto es, duran mucho más en comparación con las más comúnmente utilizadas baterías de Litio. Su tamaño y peso reducido hacen que este tipo de baterías sean las más utilizadas en robótica donde se necesita potencia y duración en una batería.

Las baterías LiPo se construyen por celdas, cada celda tiene un voltaje de 3,7 V, en el caso de los minirobots se necesitan baterías de 3 celdas (3S), esto da un voltaje de 11,1 V, y la corriente que entrega es de 1300 mAh. Se debe tener un cuidado especial con este tipo de baterías, ya que si el valor de la descarga por celda es menor a 3 V, en el interior de la batería ocurre una cristalización haciendo imposible y peligrosa la carga nuevamente de la batería.



Figura 2.10 Batería LiPo (Litio - Polímero)[20]

### 2.1.2 DISEÑO DE LA ESTRUCTURA DEL MINIROBOT

Para realizar la estructura de los minirobots se enfocó en varios aspectos tales como, configuración diferencial, dimensiones pequeñas tanto en el alto, largo y ancho del robot y una buena ubicación de los sensores ultrasónicos.

Se optó por la configuración diferencial ya que presenta buenas características como ser sencilla de implementar, al colocarla en una estructura y al momento de controlarla. Esta configuración permite controlar el cambio de dirección y velocidad de desplazamiento controlando dos motores DC, lo cual se hace con señales PWM. La modulación de ancho de pulso permite regular la velocidad de giro de los motores DC, se puede cambiar de dirección haciendo que uno de los motores gire más rápido que el otro, debido a la diferencia de velocidades el minirobot cambia de dirección.

Para realizar la estructura de los robots móviles se tomo en cuenta que las dimensiones debían ser pequeñas, ya que esto facilitaría el recorrido dentro de laberinto, además que el tamaño del laberinto es directamente proporcional con el tamaño del robot móvil. Debido a esto se trato de realizar una estructura donde los elementos estén debidamente ubicados y con el fin de ocupar el menor espacio posible.

Se consideró los elementos electrónicos y mecánicos que van a conformar el minirobot, cada minirobot constará de 3 sensores ultrasónicos, la placa del Arduino Duemilanove, el módulo Arduino Xbee Shield, la antena Xbee serie 2, una batería de LiPo de 1300 mAh de 11.1 V, el driver para los motores DC, dos micromotores DC, dos encoders de cuadratura acoplados a las dos ruedas para cada motor, la rueda de libre giro, conectores y cables.

Se determinó las dimensiones de cada uno de los elementos mencionados anteriormente y se los distribuyó de una manera adecuada para poder hacer una estructura compacta y capaz de desplazarse sin ningún inconveniente. Con una idea general de la estructura se diseñaron piezas donde se puedan empotrar los elementos y además se puedan cohesionar. Los diseños se los realizó en la herramienta computacional AUTOCAD<sup>12</sup> en dos dimensiones, tomando en cuenta el ancho del acrílico que es de 4mm, para posteriormente realizar un diseño aproximado en 3 dimensiones (Anexo 3).

El material que se utilizó para la elaboración de las estructuras de los minirobots es el acrílico debido a las características que este material presenta. El acrílico como una variación de plástico es manejable pero se necesitó una maquina especial para trabajarlo, ya que algunas piezas son muy pequeñas y necesitan de una buena precisión de corte.

## **2.2 CONSTRUCCIÓN DE LAS ESTRUCTURAS**

Una vez elaboradas las piezas se procedió al ensamblaje. Debido a que se llegó a realizar tres diseños previos a la estructura definitiva estos fueron denominados como MREC<sup>13</sup> – 01, MREC – 02 y MREC – 03. Para después escoger el diseño más adecuado. La construcción de los primeros minirobot se lo encuentra en el Anexo 6.

---

<sup>12</sup> AUTOCAD software computacional utilizado en el diseño asistido del dibujo de diferentes piezas o elementos ya sea en 2D o 3D.

<sup>13</sup> MREC, son las siglas de Mini Robot Explorador Cooperativo

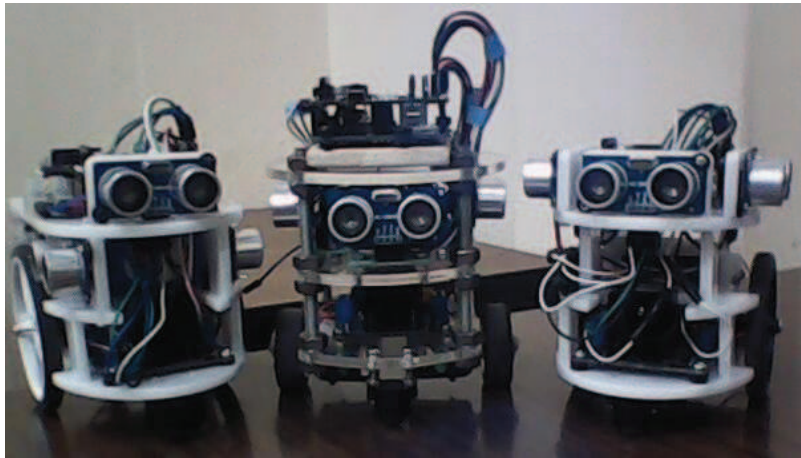


Figura 2.11 Minirobots con los cuales se realizaron las primeras pruebas

En la Figura 2.11 se muestran los primeros diseños. De izquierda a derecha se encuentran el MREC – 03, MREC – 01 y MREC – 02. Con los tres diseños elaborados se hizo las pruebas necesarias de desplazamiento por el laberinto. Obteniendo mejores resultados con el minirobot MREC – 03, ya que presenta una buena estabilidad, es compacto, se puede montar y desmontar debido a que es asegurado por medio de pernos y además las dimensiones son las adecuadas.

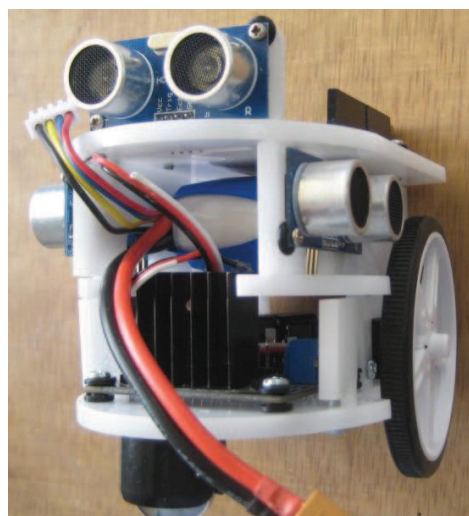


Figura 2.12 MREC – 03



### 2.3 ESTRUCTURAS DEFINITIVAS DE LOS MINIROBOTS

Después de realizar pruebas con estos diseños se logró un desplazamiento adecuado de los minirobots por el laberinto, pero no óptimo. Para cumplir con los objetivos planteados en el proyecto de titulación se implementó otros elementos como los encoders de cuadratura, los cuales necesitaban micromotores y unas llantas adecuadas, por lo cual se modifico otra vez la estructura del MREC – 03, que es la que mejor desempeño tuvo en las pruebas realizadas. Para esta nueva estructura se modifico la base inferior de la estructura MREC – 03, además de ciertas paredes y soportes, el resto de la estructura fue muy similar.



Figura 2.13 Base2 ensamblada

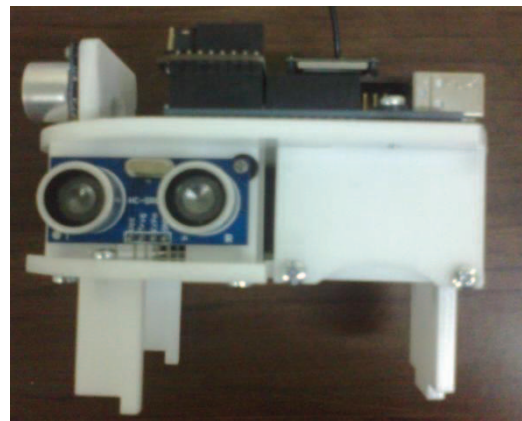


Figura 2.14 Base2 y base3 acopladas

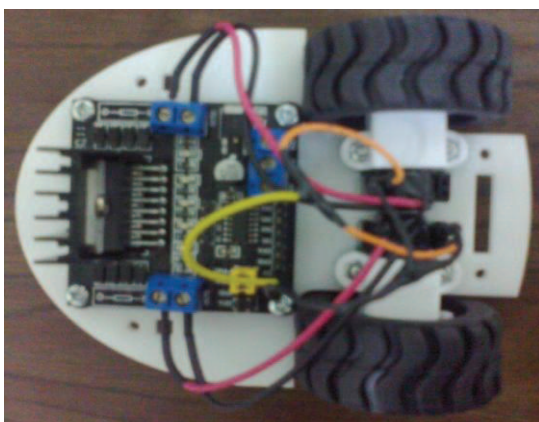


Figura 2.15 Base1 ensamblada

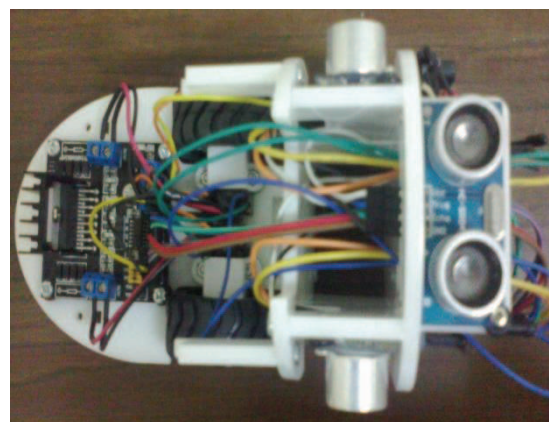


Figura 2.16 Ensamblado total y cableado



Una vez obtenida esta estructura se procedió a realizar dos estructuras similares, para que los tres minirobots tengan la misma estructura, con esto se logra que la programación dentro del microcontrolador sea la misma para los tres minirobots. La programación de cada minirobot cambia de manera muy ligera en ciertos valores de los controladores y ciertos elementos mecánicos ya que no todos responden de la misma manera.

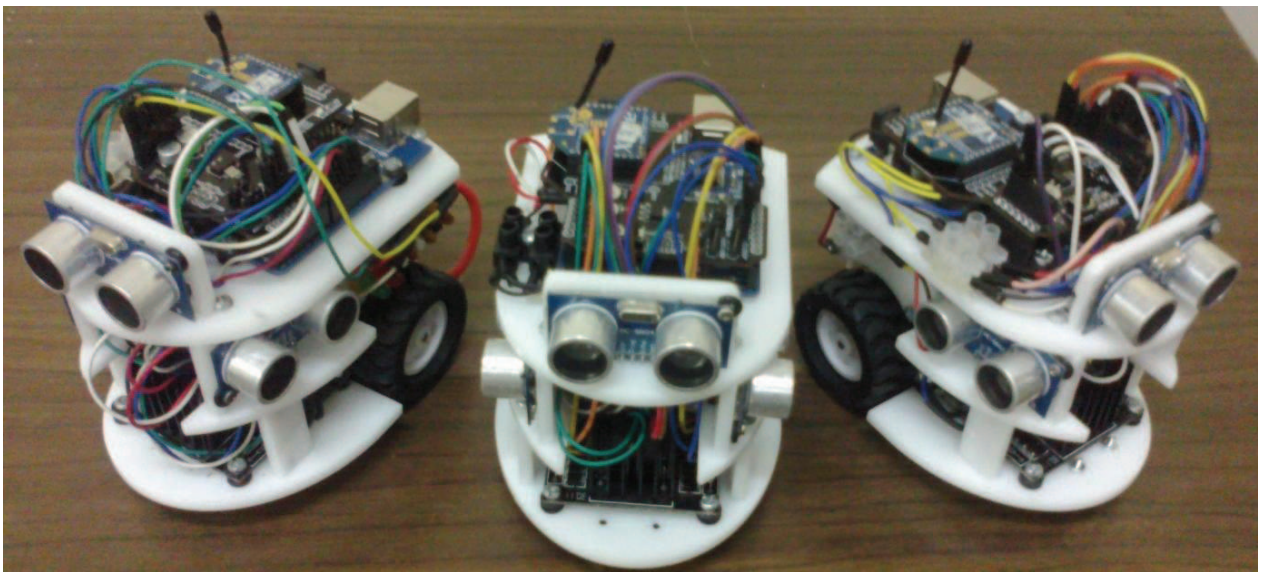


Figura 2.17 Estructuras finales de los MREC

## **CAPITULO 3**

### **DESARROLLO DE LOS ALGORITMOS DE CONTROL, EVASIÓN, MAPEO Y COOPERACIÓN**

Para desarrollar los algoritmos de control se enfocó en varios temas específicos, que son la locomoción de los MREC, algoritmos para una correcta exploración dentro del laberinto, obtención de la trayectoria y de la ruta corta, y el envío de datos adecuados hacia el computador para realizar el análisis de los mismos.

#### **3.1 LOCOMOCIÓN DE LOS MINIROBOTS**

Para realizar el correcto desplazamiento de los minirobots se utiliza una configuración diferencial, ya que es un diseño sencillo, barato y fácil de implementar. El control se hace por medio de modulación de ancho de pulso a los micromotores, con eso el minirobot puede avanzar, girar hacia la derecha o izquierda y darse una vuelta completa.

El desplazamiento dentro del laberinto va a ser por medio de sus paredes por lo que va a tener ciertos desplazamientos en trayectorias rectas y giros hacia la derecha o izquierda en ángulos rectos. Para realizar estas acciones se tienen las medidas de los sensores de ultrasonido: del centro y laterales. Con estos sensores se puede captar la distancia de las paredes laterales en el caso de estar dentro de un pasillo y con el frontal se detiene el movimiento de los minirobots ante un obstáculo. Con los sensores de distancia también se observa posibles caminos y rutas por explorar. Es importante mencionar que esta configuración diferencial necesita un control especial para trayectorias rectas.

Debido a las características mecánicas de los micromotores, estos no reaccionan de la misma manera al mismo valor de PWM para los dos motores. Si se pone el mismo valor de PWM a ambas ruedas el minirobot se desplaza en línea recta por unos cuantos centímetros y luego empieza a salirse de la trayectoria recta. Debido a esto los MREC tiene implementado un controlador PI, que es un

controlador proporcional – integral, con el cual se logra que el robot se desplace en trayectorias rectas y si existen desviaciones de trayectoria que sea capaz de corregirse.

Los giros hacia la izquierda o derecha, y la vuelta completa también deben ser realizados de manera adecuada ya que existe mucha variación en los giros. En un principio los giros se hacían en base a tiempos pero las variaciones eran muy grandes por lo que se optó hacer los giros con cuentas de los encoders. De esta manera los giros mejoran bastante pero aun existen estas variaciones debido a las características físicas y mecánicas de las ruedas y su tracción dentro del laberinto.

### **3.1.1 ALGORITMOS Y CONTROL PI PARA EL AVANCE DE LOS MINIROBOTS**

Para el avance en trayectorias rectas del minirobot se utiliza un control PI, para esto necesita las medidas de distancia de los sensores laterales y del sensor frontal. Con la medida de los sensores laterales se hace que los minirobots mantengan una cierta distancia hacia las paredes del laberinto, es decir, que no se aleje ni se acerque hacia la pared. Con la medida del sensor frontal se hace que el minirobot se detenga cuando se encuentre con una pared y de ahí se tomará una decisión dependiendo del algoritmo de exploración.

#### **3.1.1.1 Controlador PID**

El PID es un control por realimentación, donde se calcula la desviación o error entre un valor medido y un valor deseado, para realizar una acción de corrección que ajuste el proceso. El control PID incluye tres acciones: proporcional (P), integral (I) y derivativa (D). Con estas acciones se puede realizar controles P, PI, PD y PID.

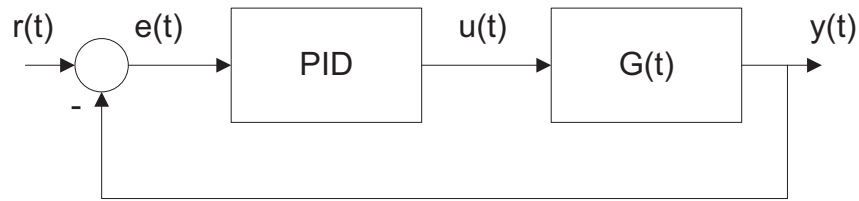


Figura 3.1 Diagrama de bloques general de un Control PID

**Acción de control proporcional.-** Entrega una salida del controlador que es proporcional al error, es decir  $u(t) = K_p \cdot e(t)$ . Donde  $K_p$  es una ganancia proporcional ajustable. Un control proporcional es adecuado para una planta estable, pero posee un desempeño limitado y un error en margen permanente (offset).

**Acción de control integral.-** Entrega una salida del controlador que es proporcional al error acumulado, debido a esto es un modo de controlar lento.  $u(t) = K_i \int_0^t e(\tau) d\tau$ . Tiene el propósito de disminuir y eliminar el error en estado estacionario. El control integral se utiliza para obviar el inconveniente de offset.

**Acción de control derivativa.-** Se manifiesta cuando hay una variación del error. Esta acción tiene un carácter de previsión, lo que hace más rápida a la acción de control, aunque tiene la desventaja de amplificar las señales de ruido y satura al actuador. La acción de control derivativa nunca se utiliza por sí sola, ya que es eficaz durante periodos transitorios.

Un controlador PID responde a la siguiente ecuación:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{d(e(t))}{dt} \quad \text{Ec 3.1}$$

En donde  $e(t)$  es el error de la señal,  $u(t)$  es la entrada de control del proceso,  $K_p$  es la ganancia proporcional,  $T_i$  es la constante de tiempo integral y  $T_d$  constante de tiempo derivativo.

En el dominio de la frecuencia, el controlador PID se escribe:

$$U(S) = K_p \left( 1 + \frac{1}{T_i S} + T_d S \right) E(S) \quad \text{Ec 3.2}$$

### 3.1.1.1.1 Controlador Digital PID [21]

La ecuación 3.2 para el PID en el dominio de la frecuencia se convierte en la ecuación 3.3 para un control PID discretizado:

$$U(z) = K_p \left[ 1 + \frac{T}{T_i (1-z^{-1})} + T_d \frac{(1-z^{-1})}{T} \right] E(z) \quad \text{Ec 3.3}$$

Donde  $T$  es el tiempo de muestreo. La ecuación anterior se representa como función de transferencia:

$$\frac{U(z)}{E(z)} = a + \frac{b}{1-z^{-1}} + c(1-z^{-1}) \quad \text{Ec 3.4}$$

Donde  $a = K_p$ ,  $b = \frac{K_p T}{T_i}$  y  $c = \frac{K_p T_d}{T}$

El controlador PID que se implementa tiene forma paralela, es decir:

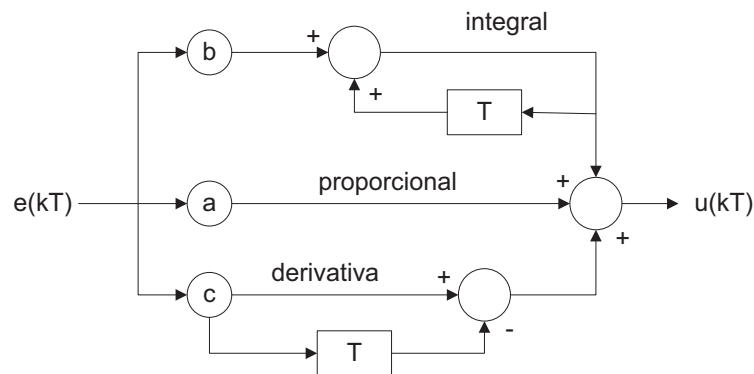


Figura 3.2 Diseño Paralelo de un Controlador PID

### 3.1.1.1.2 Algoritmo de programación PI implementado en el Arduino

Para la implementación del algoritmo de control PI se realiza ciertas modificaciones al control PID, con el fin de que sea adecuado para hacer que el robot avance en línea recta y a una distancia adecuada hacia una pared paralela al desplazamiento del minirobot. Este desplazamiento se va a dar cuando el minirobot este dentro de un pasillo generalmente y en ciertos desplazamientos que tiene que ser en trayectoria recta.

Se toma como señal de entrada la distancia de los sensores laterales y del sensor del frente. Si no hay un obstáculo en el frente los minirobots empiezan a moverse hacia adelante en una trayectoria recta, los sensores laterales le indican la distancia del robot hacia la pared, esta distancia va ser comparada con un setpoint, es decir que el robot busca compensar el movimiento de las ruedas para que el minirobot este a esta distancia especificada, para ello lo que controla es la velocidad de cada una de las ruedas derecha e izquierda.

El error va a ser la diferencia entre el setpoint y la medida entregada por el sensor lateral. Después se establece el cálculo del término proporcional y del término integral, el cálculo de la salida PI y con este valor se modifica la salida PWM hacia los micromotores. Estos cálculos se representan de mejor manera en el diagrama de bloques de la Figura 3.3:

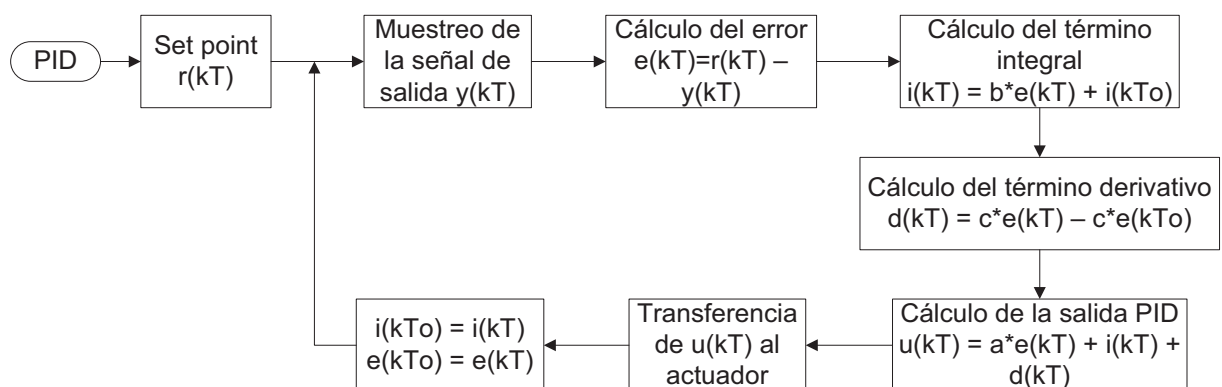


Figura 3.3 Algoritmo de programación PID digital[21]

### 3.1.1.2 Algoritmo de control para el avance del minirobot

Debido a que este algoritmo de control PI, tiene una entrada y una salida se hacen ciertas modificaciones para el control.

Si existe una pared en el lado derecho del robot y no en el lado izquierdo, el PI compensa tomando en cuenta la medida de distancia del sensor derecho y se controla el PWM que va hacia el motor de la derecha, la cual dependiendo de la distancia hacia la pared detectada, aumenta o disminuye el PWM para motor derecho, y mantiene constante el PWM del motor izquierdo; con esto se logra que el minirobot se aleje o se acerque a la pared de su derecha, tratando de mantener la misma distancia hacia esa pared. Si no existe pared en el lado derecho del robot y si en el lado izquierdo, el controlador PI utiliza la medida del sensor izquierdo para el cálculo del error y controla el PWM del motor izquierdo, además se mantiene constante el PWM para el micromotor de la derecha.

Si está dentro de un pasillo el controlador compensa utilizando la medida del sensor izquierdo y controla el PWM para el motor izquierdo.

Este algoritmo se puede explicar con el diagrama de flujo de la Figura 3.4. Cabe aclarar que el algoritmo se aplica para el avance del minirobot; este algoritmo tiene decisiones previas y decisiones posteriores, el algoritmo completo se analiza más adelante.

En el diagrama de flujo de la Figura 3.4: SF, SD, SI son las medidas de los sensores frontales, derecha e izquierda respectivamente; las medidas son en centímetros.

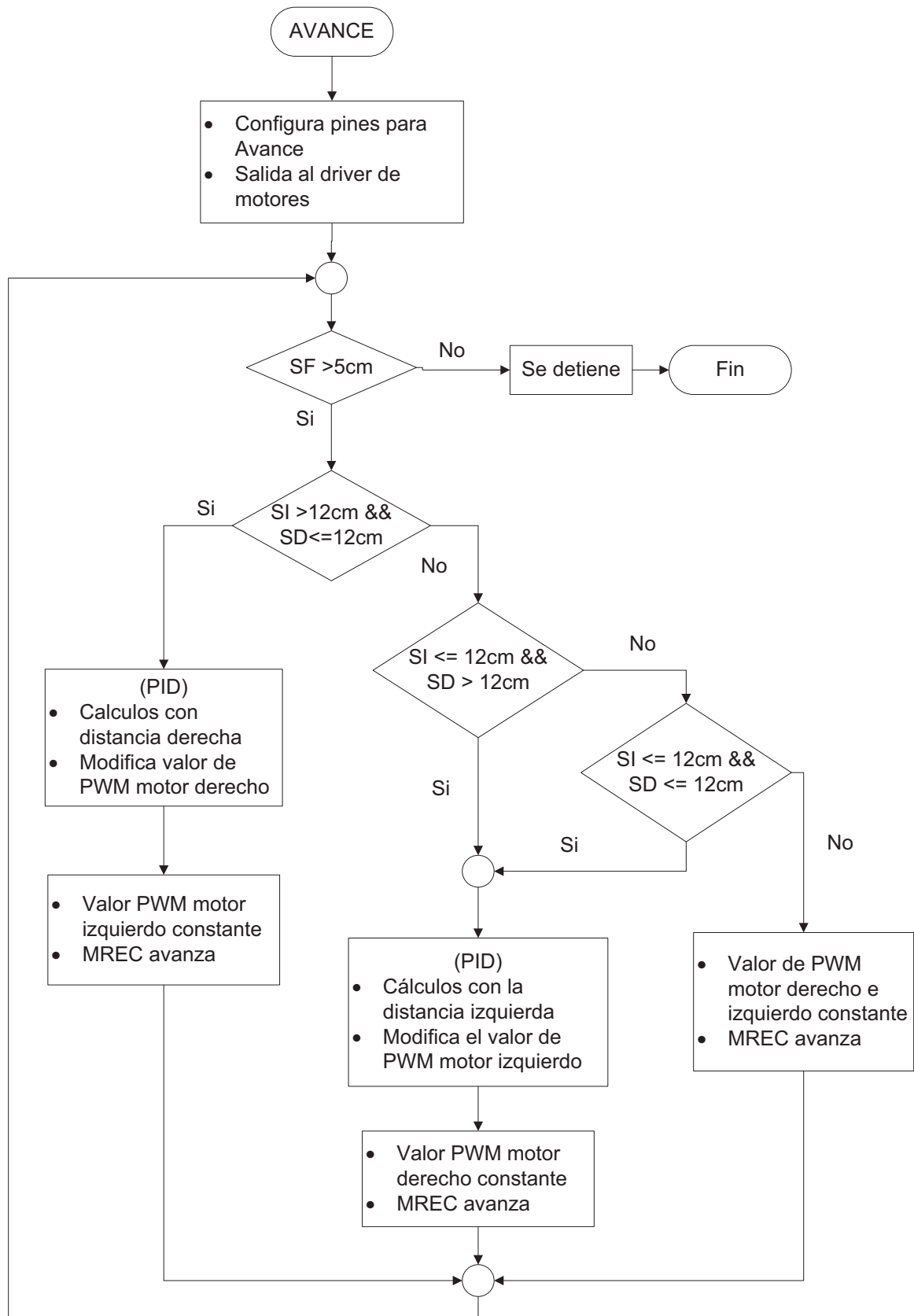


Figura 3.4 Algoritmo para el avance del MREC



### 3.1.2 CONTROL PARA GIROS DERECHA IZQUIERDA Y GIRO COMPLETO

Los giros en el minirobot son fundamentales para encontrar la ruta más corta y para un correcto desenvolvimiento dentro del laberinto; de ellos depende en gran parte, que los MREC no se pierdan dentro del laberinto, que funcione el algoritmo de exploración y el de la ruta más corta.

Los giros hacia la derecha, izquierda y vuelta completa van a ser limitados por pulsos de los encoders que tienen las llantas. Con esto se limitan los giros para que sean casi exactos aunque hay ciertas variaciones que no dependen de la electrónica sino del deslizamiento de las ruedas y de la respuesta de los micromotores.

Los encoders utilizados son de cuadratura por lo que presentan dos ondas de salida A y B desfasadas, con estas ondas se puede determinar en el arduino el sentido de giro de los motores y cuantos pulsos dan por vuelta.

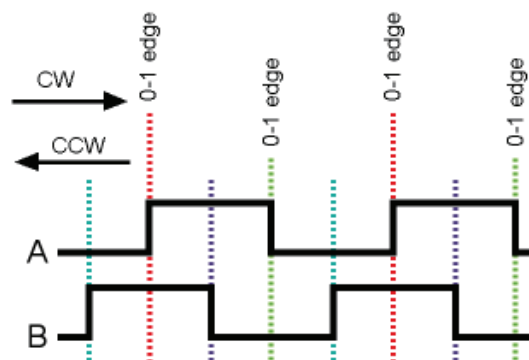


Figura 3.5 Formas de onda de los canales A y B del encoder[22]

#### 3.1.2.1 Interrupción de programa por encoders

Cuando el Arduino detecta una transición en el canal A, de un estado bajo a uno alto, de inmediato salta a una interrupción de programa “cuentaencoder”. Dentro de esta interrupción se lee el estado lógico de los dos canales A y B, si los dos canales tienen el mismo estado, alto o bajo entonces la rueda se mueve hacia adelante. Si los canales del encoder tienen estados diferentes entonces la rueda

se mueve hacia atrás. Esto se determina con las formas de onda de los canales del encoder.

Dentro de este algoritmo se declara una variable que va sumando las cuentas hacia adelante y las cuentas hacia atrás. Esto permite realizar el giro de los MREC hacia su derecha, hacia su izquierda y un giro completo; además se utiliza estas variables para realizar los cálculos para la odometría que se implementa para calcular la trayectoria por la cual se desplaza el minirobot.

El mismo algoritmo se realiza para las dos ruedas, derecha e izquierda.

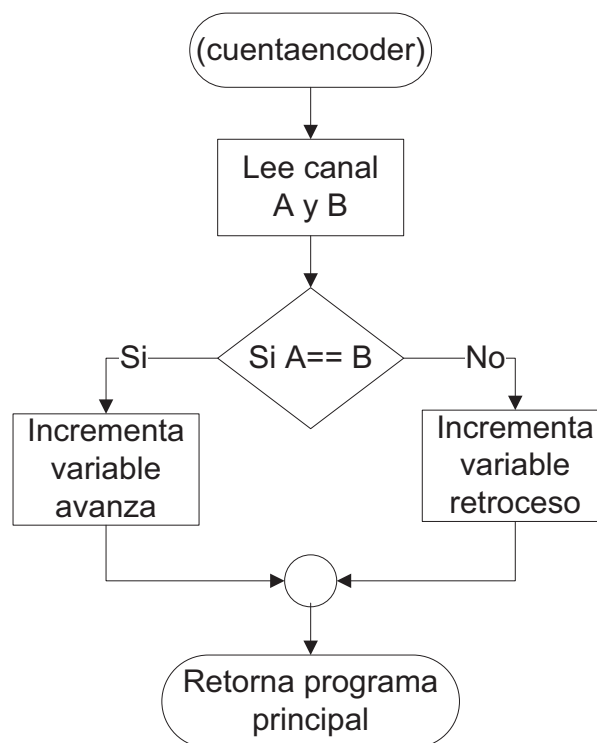


Figura 3.6 Algoritmo de interrupción de programa por encoders

### 3.1.2.2 Giros

Con el algoritmo anterior se cuenta los pulsos del encoder tanto en avance o retroceso de rueda, con esto es fácil realizar los giros del minirobot.

Para un giro hacia la izquierda lo que se hace es que la rueda derecha gire hacia adelante una cierta cantidad de pulsos y que la rueda izquierda gire hacia atrás unos cuantos pulsos, con esto se logra que el giro del minirobot sea por tracción diferencial y sea bastante preciso aunque si hay variaciones, es decir no gira exactamente en ángulo recto. Una vez realizada esta acción se detiene. Obviamente los pulsos serán proporcionados por los encoders de cada una de las ruedas.

Para un giro hacia la derecha se hace un proceso similar, pero en este caso la rueda de la derecha gira hacia atrás una cierta cantidad de pulsos y la rueda de la izquierda gira hacia adelante unos cuantos pulsos. Y de la misma forma para un giro completo la diferencia radica en la cantidad de pulsos que se hace girar a las ruedas.

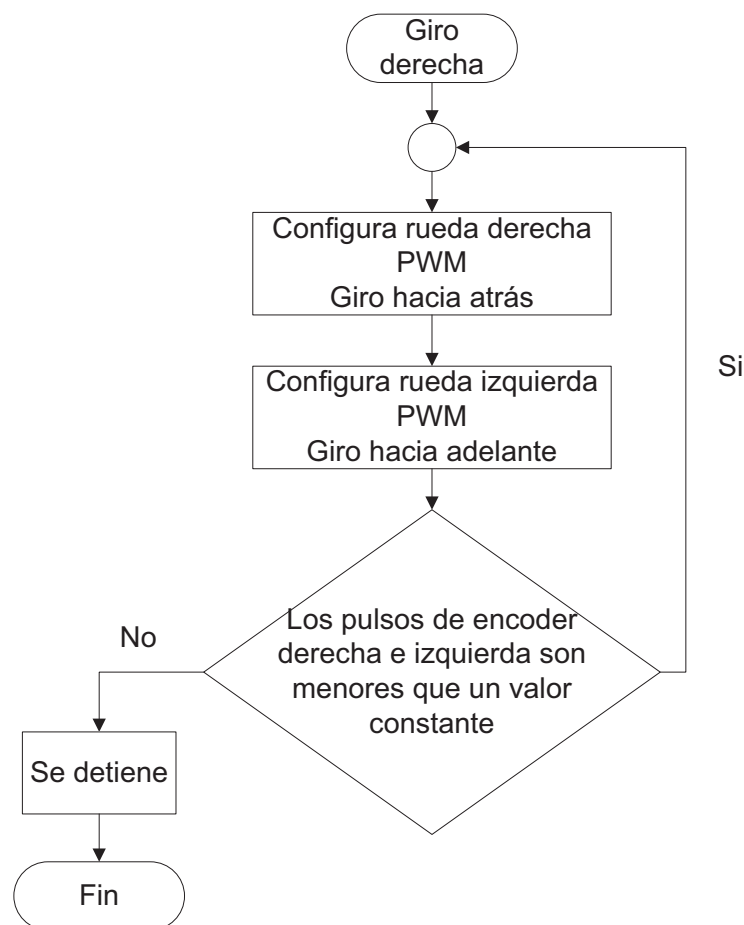


Figura 3.7 Algoritmo giro derecha

## 3.2 ALGORITMOS DE EXPLORACIÓN Y NAVEGACIÓN DENTRO DEL LABERINTO

La exploración se debe realizar de manera adecuada, ordenada y donde el minirobot recorra todos los caminos del laberinto, este algoritmo es muy importante ya que determina que el minirobot no se pierda dentro del laberinto, que pueda resolverlo en el menor tiempo posible y además que pueda guardar la ruta más corta para llegar a la meta.

### 3.2.1 ALGORITMO DE NAVEGACIÓN IMPLEMENTADO

Para realizar la navegación del minirobot dentro del laberinto se debe tener en cuenta el algoritmo de exploración (algoritmo de la mano derecha o izquierda) y las zonas que tiene un laberinto. Las zonas dentro del laberinto son 8, que son giros hacia la derecha o izquierda, intersecciones, caminos cerrados, cruces y pasillo; donde las intersecciones y cruces proporcionan dos posibilidades cada una.

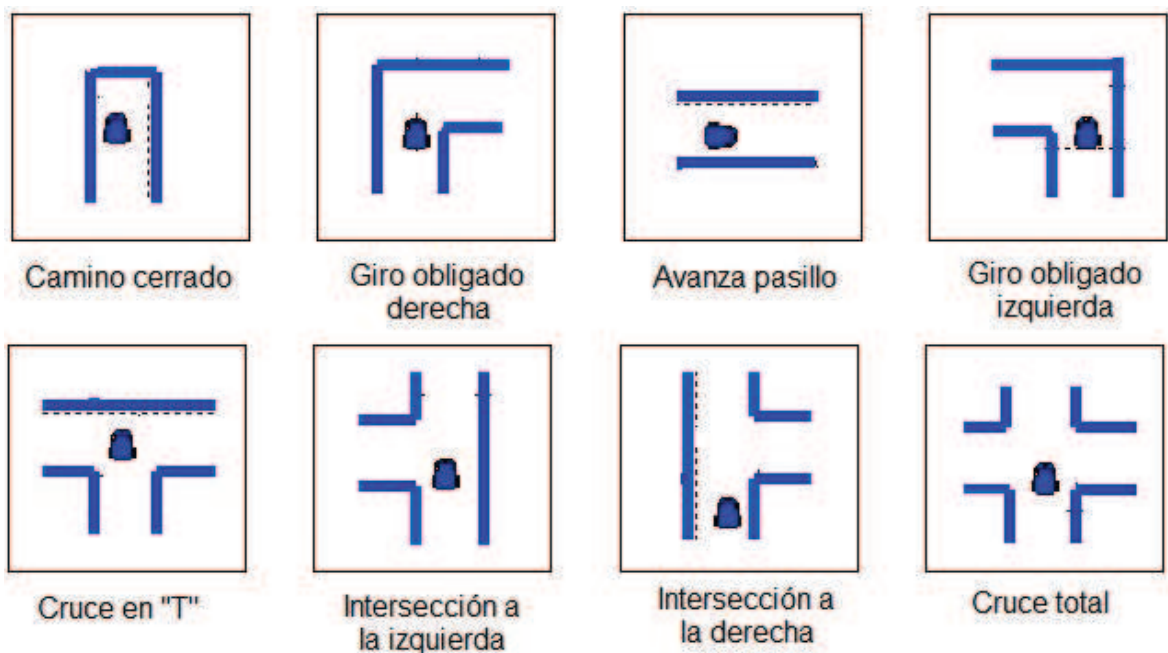


Figura 3.8 Zonas de operación dentro del laberinto

En la Figura 3.8 se puede observar las posibles zonas de operación dentro del laberinto, el algoritmo debe ser capaz de reconocer estas zonas y avanzar de acuerdo al algoritmo de la mano derecha o izquierda.

La primera zona es “camino cerrado”, para la identificación de esta zona se utilizan los sensores de distancia, donde el sensor frontal va a dar una lectura de pared en el frente, el sensor derecho una lectura de pared en su derecha y el sensor izquierdo de pared a la izquierda. La acción a tomar es comparar las medidas del lado derecho con el lado izquierdo para darse una vuelta completa por donde tenga mayor espacio para girar.

La segunda zona es “giro obligado derecha”, donde los sensores de distancia van a dar lecturas de pared al frente y a la izquierda y de camino libre hacia su derecha. La acción a tomar es girar a la derecha y avanzar hasta encontrar la nueva zona.

La tercera zona es “avanza pasillo”, donde el sensor del frente va a dar lectura de camino libre, y los sensores laterales lecturas de paredes a los costados. La acción a tomar es de “avance”, algoritmo que ya se explicó anteriormente, donde el minirobot va a avanzar por medio de un pasillo mientras va compensando su trayectoria a través de un control PI.

La cuarta zona es “giro obligado izquierda”, donde el sensor frontal y el sensor de la derecha van a dar lectura de pared y el sensor izquierdo de espacio sin pared. La acción a tomar es girar a la izquierda y avanzar hasta encontrar una nueva zona.

La quinta zona es “cruce en T”, donde el sensor del frente detecta una pared y los sensores laterales detectan posibles caminos cada uno, utilizando el algoritmo de la mano izquierda el minirobot va a girar a la izquierda y avanzar hasta salir de esta zona.

La sexta zona “intersección a la izquierda”, el sensor de la derecha detecta pared y los sensores del frente y de la izquierda posibles caminos, la acción que se toma de acuerdo al algoritmo de la mano izquierda es escoger el camino de su

izquierda. El minirobot avanza hasta situarse en la mitad de la intersección, gira a su izquierda y avanza hasta encontrar otra zona.

La séptima zona es “intersección a la derecha”, el sensor de la izquierda encuentra una pared, mientras que el sensor del frente y del costado derecho detecta posibles caminos. De acuerdo con el algoritmo de la mano izquierda el robot debe avanzar y no entrar en el camino de la derecha. Por lo que avanza por esta intersección compensando su trayectoria con la lectura del sensor izquierdo.

La octava zona es “cruce total”, en esta zona los tres sensores detectan posibles caminos, es decir encuentran 4 esquinas. Debido al algoritmo de la mano izquierda el minirobot se dirige a la izquierda, para ello avanza hasta situarse en medio del cruce, gira a la izquierda y avanza hasta encontrar una nueva zona.

Para este algoritmo de navegación se realizan varios ajustes, se incluyeron los algoritmos de avance, giros, PID, encoders que se explicaron anteriormente. El diagrama de flujo de este algoritmo se explica más adelante ya que se complementa con los algoritmos para la obtención de la trayectoria, la ruta corta y con la comunicación hacia el computador.

### **3.3 OBTENCIÓN DE LA TRAYECTORIA Y LA RUTA CORTA**

#### **3.3.1 OBTENCIÓN DE LA TRAYECTORIA – ODOMETRÍA[23]**

Para obtener la trayectoria se utiliza un sistema denominado odométrico. La odometría estima la posición y orientación de un móvil a partir del número de vueltas dadas por sus ruedas. Para ello se utiliza los encoders de cuadratura, que como ya se mencionó van acoplados a las ruedas de los minirobots. Para la estimación se requiere un registro odométrico de al menos un par de ruedas.

Un robot con configuración diferencial presenta ecuaciones cinemáticas del punto medio del eje, entre las ruedas activas, referidas a un sistema de referencia global (ver Figura 1.8); que son:

$$\dot{x} = v(t) \cos(\theta(t)) \quad \text{Ec 3.5}$$

$$\dot{y} = v(t) \text{sen}(\theta(t)) \quad \text{Ec 3.6}$$

$$\dot{\theta} = w(t) \quad \text{Ec 3.7}$$

Al integrar estas ecuaciones en un periodo de tiempo se encuentra la posición y la orientación de del minirobot:

$$x(t) = x(t_0) + \int_{t_0}^{t_f} v(t) \cos(\theta(t)) dt \quad \text{Ec 3.8}$$

$$y(t) = y(t_0) + \int_{t_0}^{t_f} v(t) \text{sen}(\theta(t)) dt \quad \text{Ec 3.9}$$

$$\theta(t) = \theta(t_0) + \int_{t_0}^{t_f} w(t) dt \quad \text{Ec 3.10}$$

Si el periodo de integración tiende a cero, entonces las integrales de las ecuaciones anteriores pueden ser reemplazadas por desplazamientos diferenciales  $\Delta x, \Delta y, \Delta \theta$ .

Si se mantiene una frecuencia de muestreo constante y elevada sobre la odometría del minirobot, se estima la posición y orientación mediante las siguientes ecuaciones:

$$x_k = x_{k-1} + \Delta x_k \quad \text{Ec 3.11}$$

$$y_k = y_{k-1} + \Delta y_k \quad \text{Ec 3.12}$$

$$\theta_k = \theta_{k-1} + \Delta \theta_k \quad \text{Ec 3.13}$$

Estas ecuaciones ayudan a encontrar la posición y orientación de los minirobots, pero para ello se necesita hacer cálculos con los encoders.

Para medir la distancia recorrida por cada rueda se toma en cuenta el diámetro de la rueda y el número de pulsos que entrega por vuelta cada encoder:

$$\text{Distancia1vuelta} = \pi * \phi_{rueda} \quad \text{Ec 3.14}$$

$$\mathbf{Resolución\_encoder} = \frac{\mathbf{Distancia1vuelta}}{\mathbf{\#pulsos\_vuelta}} \quad \text{Ec 3.15}$$

$$\mathbf{S_{recorrida}} = \frac{\mathbf{Distancia1vuelta}}{\mathbf{\#pulsos\_vuelta}} * \mathbf{\#pulsos\_recorridos} \quad \text{Ec 3.16}$$

Donde  $S_{recorrida}$  es la distancia recorrida por una rueda. Con las ecuaciones anteriores se obtuvieron las ecuaciones que se implementaron en el arduino, para conocer los desplazamientos de las ruedas derecha e izquierda y además la distancia recorrida por el centro del eje de tracción los minirobots:

$$\mathbf{S_{derecha}} = \mathbf{1.1} * \mathbf{\#pulsos\_recorridos} \quad \text{Ec 3.17}$$

$$\mathbf{S_{izquierda}} = \mathbf{1.1} * \mathbf{\#pulsos\_recorridos} \quad \text{Ec 3.18}$$

$$\mathbf{S_{centro}} = \frac{\mathbf{S_{derecha}} + \mathbf{S_{izquierda}}}{\mathbf{2}} \quad \text{Ec 3.19}$$

Donde el número de pulsos recorridos se va a obtener de la implementación de la interrupción de programa “cuentaencoder”, y la constante 1.1 se obtiene debido a que el minirobot tiene una resolución de 1.1cm de avance por cada pulso dado. Con una llanta de 4,2 cm de diámetro:

$$\mathbf{Resolución\_encoder} = \frac{\mathbf{Distancia1vuelta}}{\mathbf{\#pulsos\_vuelta}} = \frac{\mathbf{\pi * 4,2cm}}{\mathbf{12pulsos}} = \mathbf{1,1} \frac{\mathbf{cm}}{\mathbf{pulso}} \quad \text{Ec 3.20}$$

Para encontrar el ángulo de orientación del minirobot en un instante determinado se utiliza una fórmula aproximada que se obtiene de la fórmula trigonométrica  $S = \theta * R$ , donde  $S$  es longitud de arco y  $R$  es el radio de curvatura, (Figura 1.8), por lo tanto la fórmula se expresaría como:

$$\Delta\theta_k = \frac{S_{derecha} - S_{izquierda}}{d_{ruedas}} \quad \text{Ec 3.21}$$



Donde  $d_{ruedas}$  es la distancia que existe entre las dos ruedas de tracción.

Una vez obtenida la Ecuación 3.11 se puede obtener los valores de las otras ecuaciones de posición y orientación que se ven en las ecuaciones 3.7, las nuevas ecuaciones que son implementadas en el arduino quedan de esta manera:

$$\theta_k = \theta_{k-1} + \frac{S_{derecha} - S_{izquierda}}{d_{ruedas}} \quad \text{Ec 3.22}$$

$$x_k = x_{k-1} + S_{centro} * \cos(\theta_k) \quad \text{Ec 3.23}$$

$$y_k = y_{k-1} + S_{centro} * \sin(\theta_k) \quad \text{Ec 3.24}$$

Como se ve en las ecuaciones anteriores se muestra un término anterior al que se quiere obtener que va representado por el subíndice k-1, son valores iniciales de posición en x,y y orientación, los cuales van a ser configurados dentro del programa.[23]

### 3.3.2 OBTENCIÓN DE LA RUTA CORTA

Para encontrar la ruta corta, es importante que el algoritmo de exploración sea el adecuado, que los giros sean prácticamente exactos y que el recorrido se lo realice por el centro de los caminos dentro del laberinto, ya que con esto los minirobots no se pierden, en caso de no hacerlo van a proporcionar datos erróneos los cuales son utilizados para la ruta corta y para la interfaz gráfica dentro del MATLAB.

El algoritmo para la ruta corta se utiliza cuando se aplica el algoritmo de la ley de la mano derecha o izquierda. Este algoritmo va almacenando letras de decisiones que va tomando durante su desplazamiento en un vector, las letras que utiliza son: "R" de una decisión de avance recto, "V" de una decisión de media vuelta, "D" una decisión giro a la derecha, "I" de una decisión de giro a la izquierda.

Para ello se utilizan las zonas definidas en el algoritmo de exploración, en cada una de estas zonas se realiza un acción, y debido al algoritmo de la mano derecha o izquierda se toman ciertas decisiones en zonas donde hay dos o más posibilidades de operación. Las zonas donde se toma decisiones sobre a donde ir van a ser las zonas donde se asigna las letras de identificación.

La primera zona de “camino cerrado” asigna la letra “V”

La segunda zona “giro obligado derecha” no asigna ninguna letra al vector, ya que es un giro que necesariamente se tiene que dar para que los minirobots avancen.

La tercera zona es “avanza pasillo”, aquí tampoco se le asigna ninguna letra al vector.

La cuarta zona “giro obligado izquierda”, al igual que la segunda zona no asigna ningún valor al vector.

La quinta zona “cruce en T”, asigna la letra “I” en el caso de utilizar la ley de la mano izquierda, se pone “D” en caso de utilizar la ley de la mano derecha.

La sexta zona “intersección a la izquierda”, se asigna la letra “I” con el algoritmo de la mano izquierda y “R” con el de la derecha.

La séptima zona es “intersección a la derecha”, se asigna la letra “R” con el algoritmo de la mano izquierda y “D” con el de la derecha

La octava zona es “cruce total”, asigna la letra “I” en el caso de utilizar la ley de la mano izquierda, se pone “D” en caso de utilizar la ley de la mano derecha.

De esta forma se va asignando letras a un vector de las decisiones que tome el minirobot, pero todavía no se obtiene la ruta corta, para ello lo que se hace es el algoritmo en sí, que va verificando las letras almacenados en el vector y va identificando los caminos que no debió tomar. Cuando existen más de tres valores almacenados dentro del vector se hace la verificación, si el minirobot encontró un camino cerrado, es decir, si dentro del vector está la letra “V” se indica que el minirobot entró en un camino sin salida y se comprueba los valores anteriores y posteriores a la letra “V”. Se identifica las letras anterior y posterior a la letra V, y se le asigna un valor numérico similar al ángulo de rotación, “R” es 0, “D” es 90, “I”

es 270 y "V" es 180, se suma el valor de estas 3 letras y se saca el residuo de la división para 360 (que representa una vuelta completa), el residuo es nuevamente comparado, si es 0 entonces se le asigna la letra "R", si es 90 la letra "D", si es 180 la letra "V" y si es 270 la letra "I"; esta nueva letra va a representar el camino que debió escoger el minirobot y los valores anterior, posterior y la letra "V" va a ser reemplazada por la nueva letra obtenida[24].

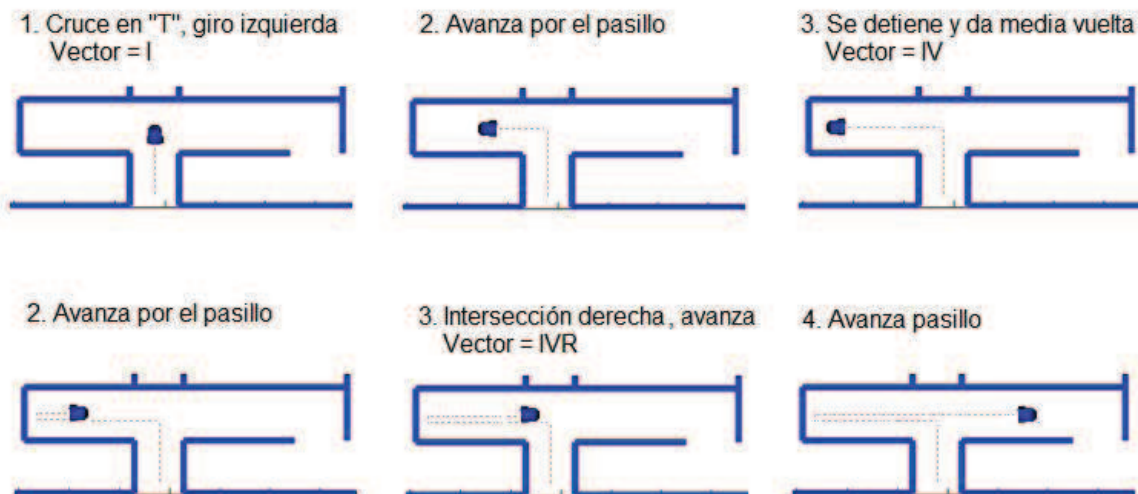


Figura 3.9 Ejemplo de exploración para obtener la ruta corta

En la Figura 3.9 se muestra caso por caso como se va asignando las letras a la variable vector, cuando las componentes del vector son iguales o mayores a tres empiezan a actuar el algoritmo de simplificación de ruta. A las letras del vector se les asigna los valores numéricos  $I = 270$ ,  $V = 180$  y  $R = 0$ , se suma el valor de las letras y se obtiene 450, este valor es dividido para 360, y el residuo es 90. El valor del residuo es comparado y para 90 se le asigna la letra D. El vector = IVR es reemplazado por vector = D, lo que quiere decir que en el cruce en T que detecta el minirobot inicialmente no debió girar a la izquierda sino a la derecha. Es importante que se de los reemplazos que realiza el algoritmo de simplificación de ruta, en este ejemplo la letra "D", ya que para el recorrido por la ruta corta se verifican las letras del vector y se realizan las acciones dependiendo de estas letras almacenadas.

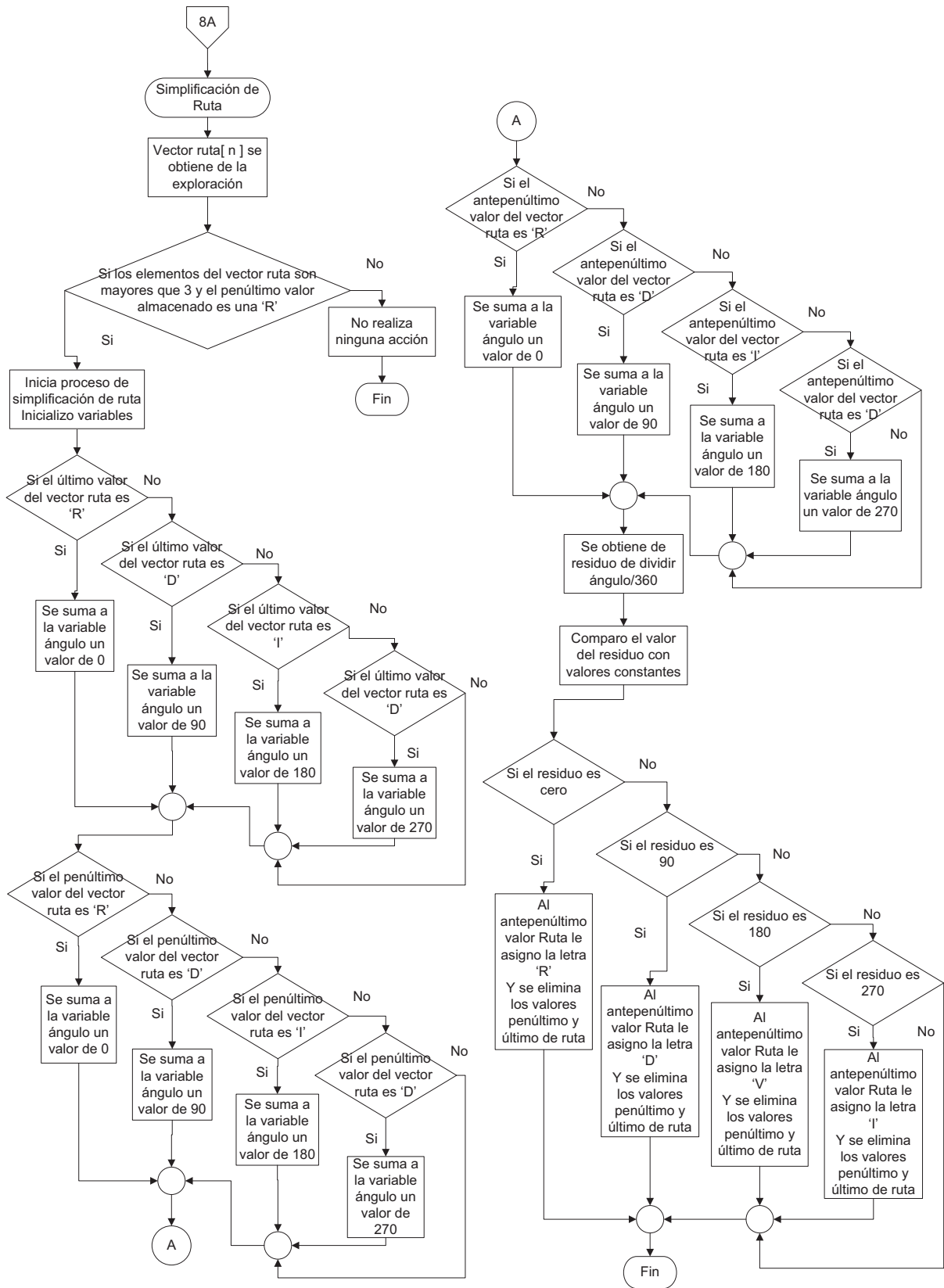


Figura 3.10 Algoritmo para obtener la ruta corta

### 3.4 TRATAMIENTO DE DATOS

Los datos que envían los minirobots hacia la PC se transmiten utilizando una red Zigbee formada por un coordinador y tres routers, los cuales son configurados mediante un software especializado de la compañía DIGI, los minirobots envían los datos de posición tales como, posición en X, posición en Y y la orientación del minirobot phi, y las medidas de las distancias de los sensores ultrasónicos del frente y de los lados, izquierda y derecha.

Para una buena comunicación en la red Zigbee se debe realizar una correcta configuración a las antenas Xbee, la configuración se muestra en el numeral 3.4.1, también los diferentes modos de comunicación y los tipos de conexión que tiene la red. Además utilizando el software MATLAB se hace un tratamiento de los datos que recibe el coordinador y que pasa a la PC para que el resultado sea una aproximación de los sectores que los minirobots van a mapear con la ayuda de los encoders y de los sensores ultrasónicos.

#### 3.4.1 CONFIGURACIÓN DE LOS MÓDULOS XBEE

Para realizar la configuración de los módulos xbee, se utiliza el programa X-CTU<sup>14</sup>.

En la Figura 3.11 se observa la primera pestaña (PC Settings) de programación del software, en esta página se configura el puerto por el cual se quiere hacer la comunicación, en este caso el puerto serie número 3 (COM3), también la velocidad de comunicación (9600 baudios), los bits de datos (8 bits de datos), la paridad (sin paridad) y los bits de parada (1 bit de parada). También se puede realizar una prueba o “test” a la antena xbee, para advertir si tiene algún desperfecto o si el puerto serie no se encuentra en buen estado. Esto se lo realiza mediante el botón “TEST/QUERY”.

---

<sup>14</sup> <http://www.digi.com/support/productdetail?pid=3352&osvid=57&type=utilities>

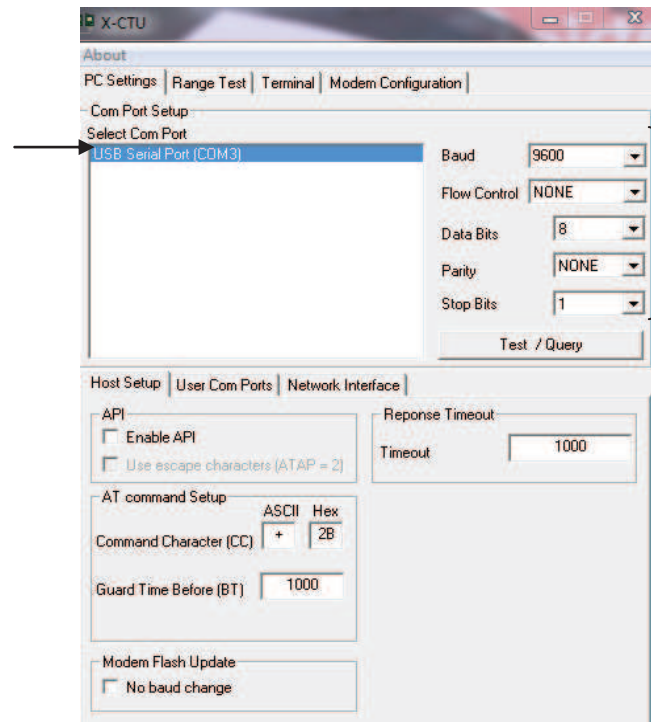


Figura 3.11 Software X – CTU.

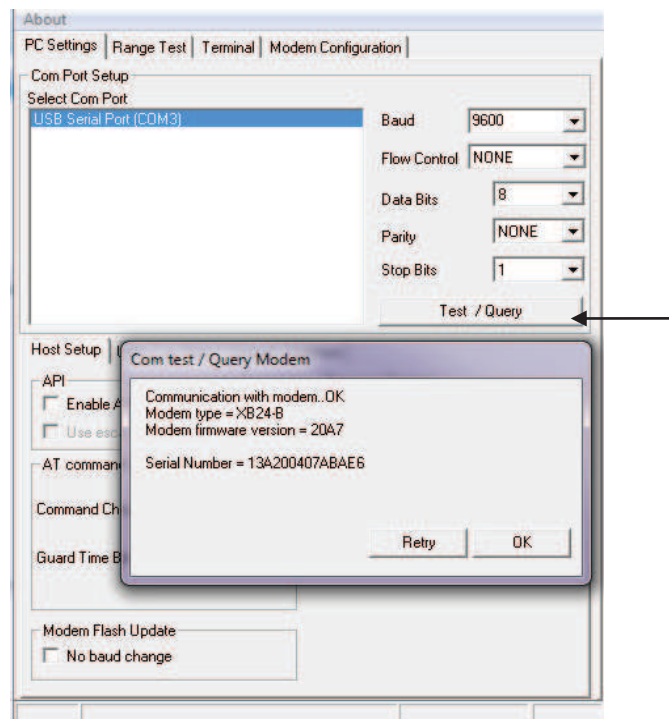


Figura 3.12 “Test” realizado a una antena Xbee

De la Figura 3.12 se tienen los datos de la antena xbee, como son, el tipo de modem (XB24-B), la versión de la firma (firmware 20A7), el estado de la comunicación (OK), y el serial de la antena (13A200407ABAE6) que es única, es decir, no existen dos antenas xbee con el mismo serial.

Para crear una red Zigbee se necesitan uno y solo un coordinador y varios routers o dispositivos finales. El coordinador se encarga de controlar la red y los caminos que deben seguir los datos para que los demás dispositivos se puedan comunicar. Los routers se encargan de pasar los datos hacia al coordinador y hacia los dispositivos finales si existieran en la red. Los dispositivos finales solo pueden enviar información hacia los routers o hacia el coordinador.

La red Zigbee tiene dos modos de operación, modo transparente y modo API, se utiliza el modo de operación transparente, debido a que es sencillo de configurar y cumple con los requerimientos que se necesita en el envío y recepción de datos, puesto que todo lo que pasa por el puerto UART<sup>15</sup> es enviado al módulo deseado. El modo API (interfaz de programación aplicada o interface programming application por sus siglas en inglés) es más complicado tanto en la configuración como en la aplicación en el proyecto de titulación, por esta razón solo se menciona y no se analiza a profundidad.

Para crear la red Zigbee dentro de la configuración de las antenas en modo transparente, se debe tener un mismo nombre de red (PAN ID), la comunicación debe hacerse por un mismo canal (CH) y las direcciones de destino (DH y DL) deben configurarse de acuerdo al tipo de conexión que se tenga en la red. Si se tiene una conexión punto – punto, es decir solamente dos antenas xbee, la dirección de destino de la una será la dirección de la origen de la otra y viceversa.

Pero en el proyecto de titulación se utiliza una conexión punto – multipunto y lo dicho en el párrafo anterior ya no aplica, ya que en este tipo de conexión se tienen un coordinador y tres routers.

---

<sup>15</sup> UART (Transmisor – Receptor Universal Asíncrono), es el componente clave del subsistema de comunicaciones series de un microcontrolador.

En el software X – CTU después de la primera pestaña de configuración, se tiene tres más a continuación, pero de esas pestañas se menciona la última que es la de configuración del modem (Modem Configuration) para preparar el coordinador y los routers de la red.

### 3.4.1.1 Configuración del coordinador

Para configurar el coordinador se debe ubicar en la cuarta pestaña del X – CTU y elegir el tipo de modem xbee (XB24 – ZB), además de la función que va a desempeñar el xbee (ZIGBEE COORDINADOR AT), la versión más actualizada del firmware (20A7 hasta el momento) y los siguientes parámetros:

- PAN ID = 1987
- SC = 0380
- DH = 0x0000
- DL = 0xFFFF

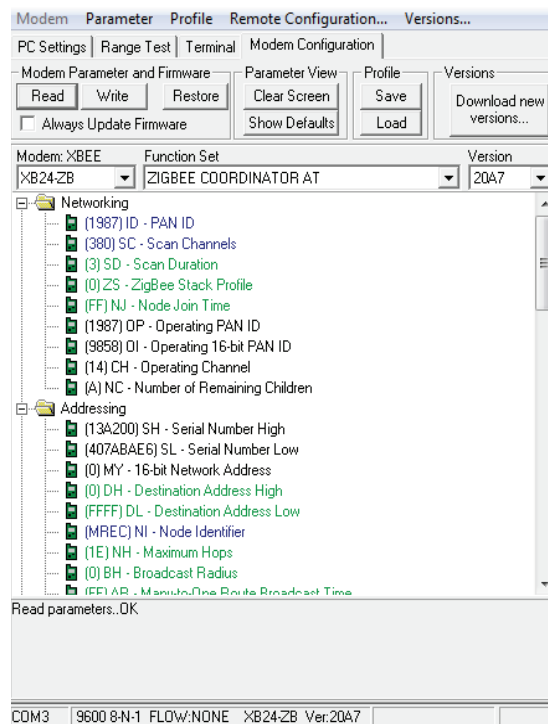


Figura 3.13 Configuración del Coordinador



A los parámetros mencionados en el párrafo anterior, se debe resaltar SC (scan channel por sus siglas en inglés) que es el scanner de canales, ayuda en la red a buscar los canales disponibles para la comunicación, con un valor hexadecimal de 0380 se pueden utilizar los canales 12, 13 y 14. En la red se utiliza un tipo de conexión punto – multipunto, por lo que se debe utilizar una dirección broadcast (0x0000FFFF donde DH=0x0000 y DL=0xFFFF) en el coordinador para que los datos que se envíen desde éste sean leídos por todos los routers o dispositivos finales en la red.

### 3.4.1.2 Configuración de los routers

Los routers se configuran de igual manera que el coordinador, la diferencia está en la versión del firmware (22A7 más actual al momento) y en la función que desempeña (ZIGBEE ROUTER AT), para los parámetros se tiene:

- PAN ID = 1987
- SC = 0380
- DH = 0x0000 y DL = 0x0000

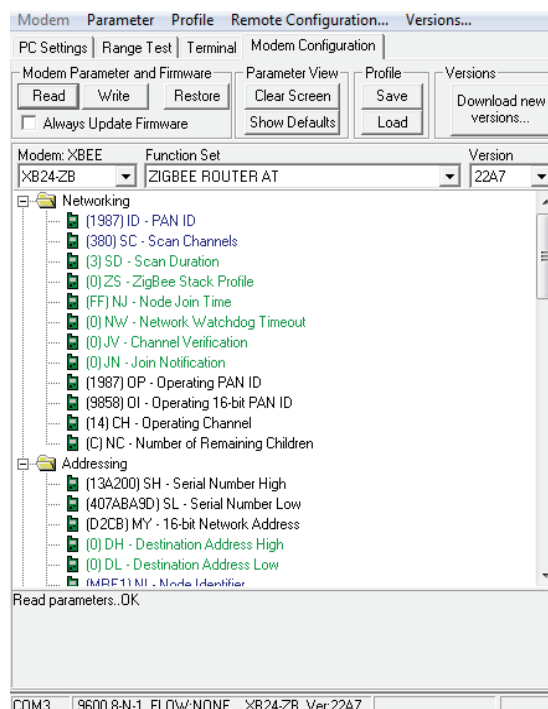


Figura 3.14 Configuración de un Router

Se debe tener muy en cuenta que los parámetros de PAN ID y de SC deben ser los mismos tanto en el coordinador como en los routers para que la red se pueda conectar, de otra forma no existiría conexión alguna. La dirección DH=0x00000000 y DL=0x00000000 que se tiene en los routers es para que se comuniquen directamente con el coordinador.

La misma configuración se repite para las otras dos antenas xbee que van a trabajar como routers.

### 3.4.2 ENVÍO Y RECEPCIÓN DE DATOS

Dentro del arduino se tiene una rutina donde se envía todos los datos hacia los pines de comunicación del microcontrolador y a su vez también hacia la antena xbee que esta como router, donde mediante radiofrecuencia los datos son transmitidos hacia el coordinador y al mismo tiempo mediante los pines de comunicación hacia el puerto serie de la PC para recibir los datos.

#### 3.4.2.1 Envío de datos desde el arduino

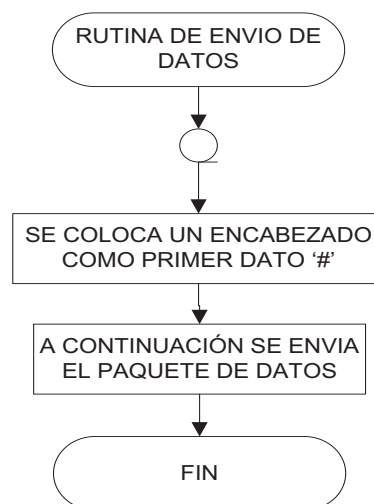


Figura 3.15 Rutina de envío de datos

Para identificar los datos de los tres minirobots se optó por colocar un encabezado en cada uno de los paquetes de datos, que se envía desde el arduino.

```

Serial.print('#');
Serial.print(",");
Serial.print(xint);
Serial.print(",");
Serial.print(yint);
Serial.print(",");
Serial.print(tetha);
Serial.print(",");
Serial.print(izquierda);
Serial.print(",");
Serial.print(centro);
Serial.print(",");
Serial.println(derecha);
delay(10);

```

Figura 3.16 Envío de datos con un encabezado, código de programación realizado en el arduino

En la figura 3.16 se muestra parte de la rutina de envío de datos desde el arduino, en la primera línea se encuentra el encabezado '#', y a continuación el paquete de datos donde están, la posición del minirobot en X, la posición del minirobot en Y, la orientación del minirobot TETHA, el valor de la distancia en centímetros del sensor ultrasónico de la izquierda, centro y derecha.

Esta rutina de envío estará activa en casi todo momento para que el minirobot proporcione los datos suficientes para el análisis en la PC. Para los otros dos minirobots los encabezados son los siguientes, '\$' y '%'.

#### 3.4.2.2 Recepción de datos en la PC

Después de que los datos son enviados hacia el coordinador, en la PC los datos llegan uno a continuación del otro.

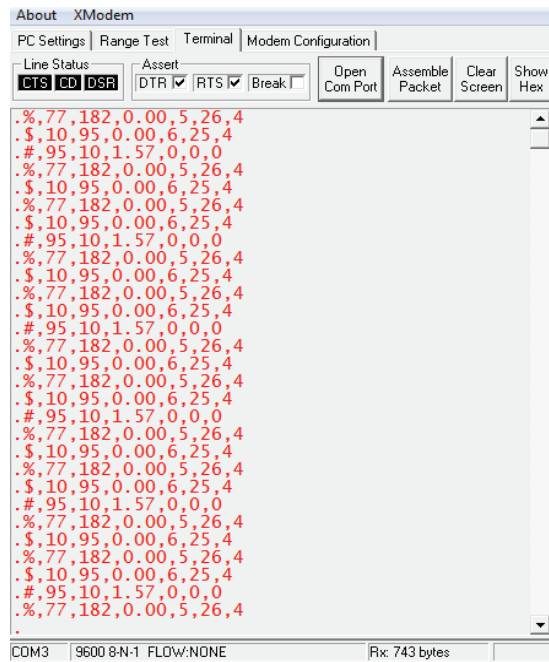


Figura 3.17 Ejemplo de recepción de datos con X – CTU (ver numeral 3.4.1)

De la figura 3.17 se tiene los paquetes de datos que llegan al coordinador, los mismos que como primer dato llevan el encabezado ('#', '\$' y '%') y después todos los datos que se necesitan en el siguiente orden: posición en X, posición en Y, orientación del minirobot THETA, el valor de los sensores izquierda, centro y derecha y por último el salto a la siguiente línea ó 'CR'<sup>16</sup>.

En MATLAB se toma los datos y se hace el análisis para que el resultado sea el mapeo de diferentes zonas del laberinto. Lo primero que se hace es escanear el puerto serie de la PC para verificar si existen datos en el buffer de entrada, a esos datos se les coloca un condicional antes de pasar, ya que algunos paquetes de datos se pierdan o lleguen incompletos, lo que se hace es calcular el tamaño del paquete de datos (en este caso 8) y verificar que el primer dato sea el encabezado predefinido antes ('#', '\$' ó '%'), solo si cumplen estas dos condiciones el paquete de datos puede pasar al siguiente análisis.

<sup>16</sup> CR (carriage return) es uno de los caracteres de control de la codificación ASCII, Unicode, o EBCDIC, que hace que se mueva el cursor a la primera posición de una línea.

```

dato=fscanf(s, '%c, %d, %d, %f, %d, %d, %d');
vector = length(dato);
if vector == 8 && (dato(1) == 35 || dato(1) == '#')
if vector == 8 && (dato(1) == 36 || dato(1) == '$')
if vector == 8 && (dato(1) == 37 || dato(1) == '%')

```

Figura 3.18 Escaneo del puerto serie y condiciones para el paquete de datos, código de programación en MATLAB.

Cuando el paquete de datos cumple con las condiciones, cada uno de los datos es guardado en vectores, cuyo tamaño es igual al número de muestras que se desea tomar. Para que el gráfico del mapeo sea aproximado al original los tres minirobots deben tener el mismo sistema de referencia o coordenadas que se le da al laberinto.

Para graficar la trayectoria de los minirobots se utiliza los datos de la odometría, haciendo una corrección en la orientación del minirobot.

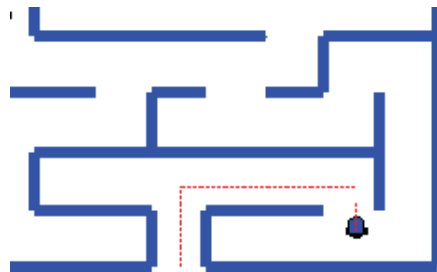


Figura 3.19 Simulación de la trayectoria de un minirobot (trayectoria en línea roja)

La odometría acumula mucho error según transcurre el tiempo, es por eso que se hacen correcciones en la orientación del minirobots.

El gráfico de las paredes del laberinto se hace a partir de cálculos que son función tanto de la odometría como del eje de referencia del laberinto y de los minirobots.

Las ecuaciones utilizadas son [25]:

$$\mathbf{salidaxn}(i) = \mathbf{p} + \mathbf{x}(i) + \mathbf{módulo} * \mathbf{cos} \theta \quad \text{Ec 3.25}$$

$$\mathbf{salidayn}(i) = \mathbf{p} + \mathbf{y}(i) + \mathbf{modulo} * \mathbf{sin} \theta \quad \text{Ec 3.26}$$

Donde:  $p$  es una constante que se suma o resta dependiendo del ángulo de inclinación del mini robot.

$n$  es el número de salida que corresponde según el dato

$i$  es el índice del vector donde se guarda los valores de salida.

$X$  y  $Y$  son las posiciones de X y Y entregadas por odometría.

$\theta$  es el ángulo que forma el rayo del ultrasonido con el eje X o Y

$módulo$  es el valor de la distancia de los sensores ultrasónicos.

Para cada uno de los datos de los minirobots se aplican las mismas ecuaciones, después de que los valores que se necesitan son calculados, en MATLAB se utilizan las funciones PLOT<sup>17</sup> y GETFRAME<sup>18</sup> para que el mapeamiento se pueda observar en la PC como una animación.

En la Figura 3.20 se muestra una simulación con los tres minirobots dentro del laberinto, en el siguiente capítulo se mostrarán los resultados experimentales que se lograron con los minirobots.

---

<sup>17</sup> PLOT función de MATLAB utilizada para graficar puntos de los vectores Y en función de X

<sup>18</sup> GETFRAME función de MATLAB que hace el efecto de animación sin el uso de mucha memoria

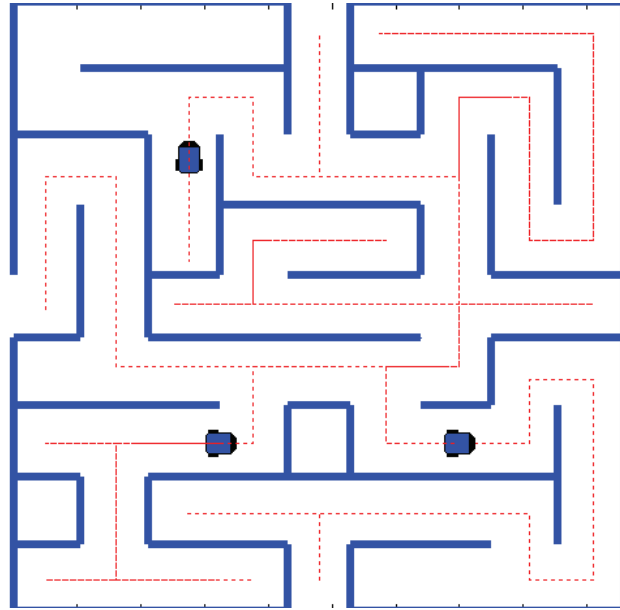


Figura 3.20 Simulación de los tres minirobots dentro del laberinto

### 3.5 INTERFAZ GRÁFICA DE MATLAB (GUIDE)[26]

La interfaz gráfica de MATLAB o GUI (Graphical User Interface), es la forma visual de comunicación entre el programa y el usuario que opera la PC, las principales ventanas de la GUIDE del proyecto de titulación son las siguientes:



Figura 3.21 Ventana de inicio



Figura 3.22 Ventana de menús

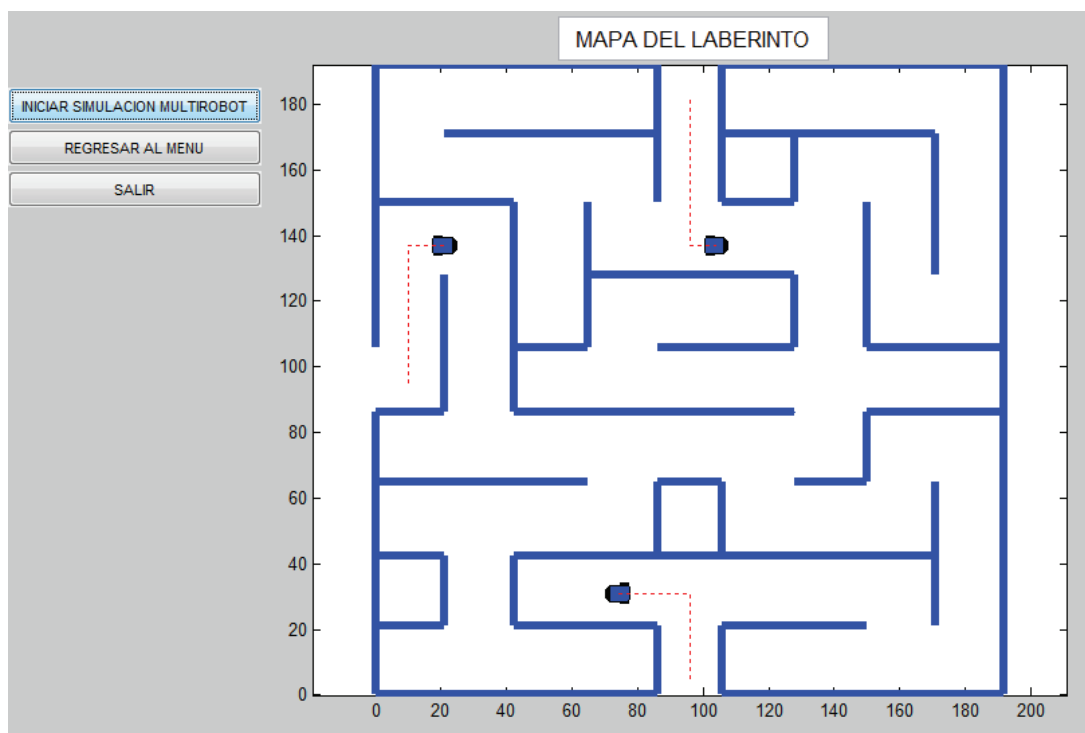


Figura 3.23 Ventana de simulación multirobot





Figura 3.24 Ventana de funcionamiento y control de los minirobots

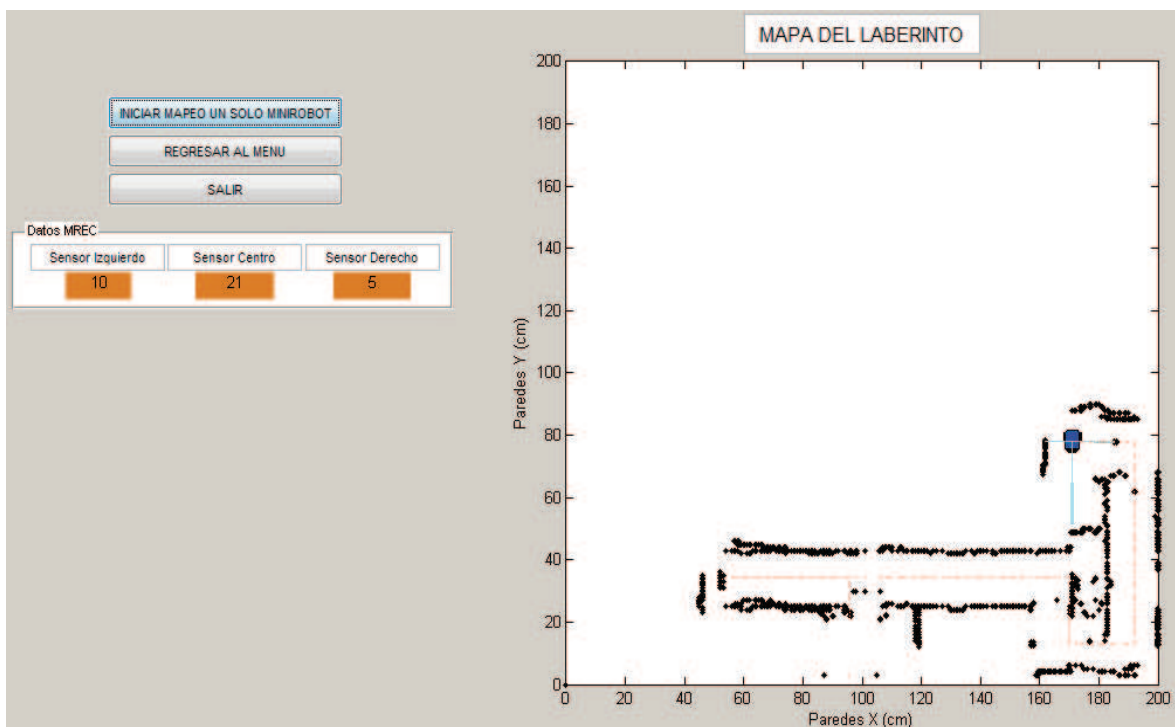


Figura 3.25 Ventana del mapeo de un solo minirobot

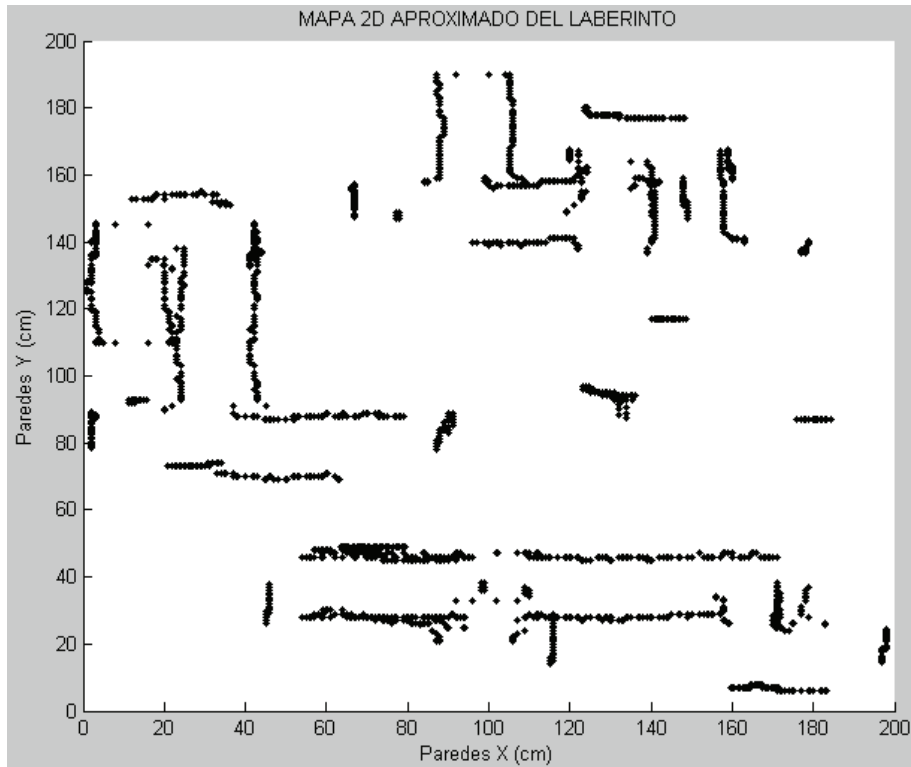


Figura 3.26 Ventana del mapeo multirobot



Figura 3.27 Ventana para ruta corta

### 3.6 ALGORITMOS IMPLEMENTADOS EN EL ARDUINO

Las siguientes figuras corresponden al algoritmo total implementado en el arduino.

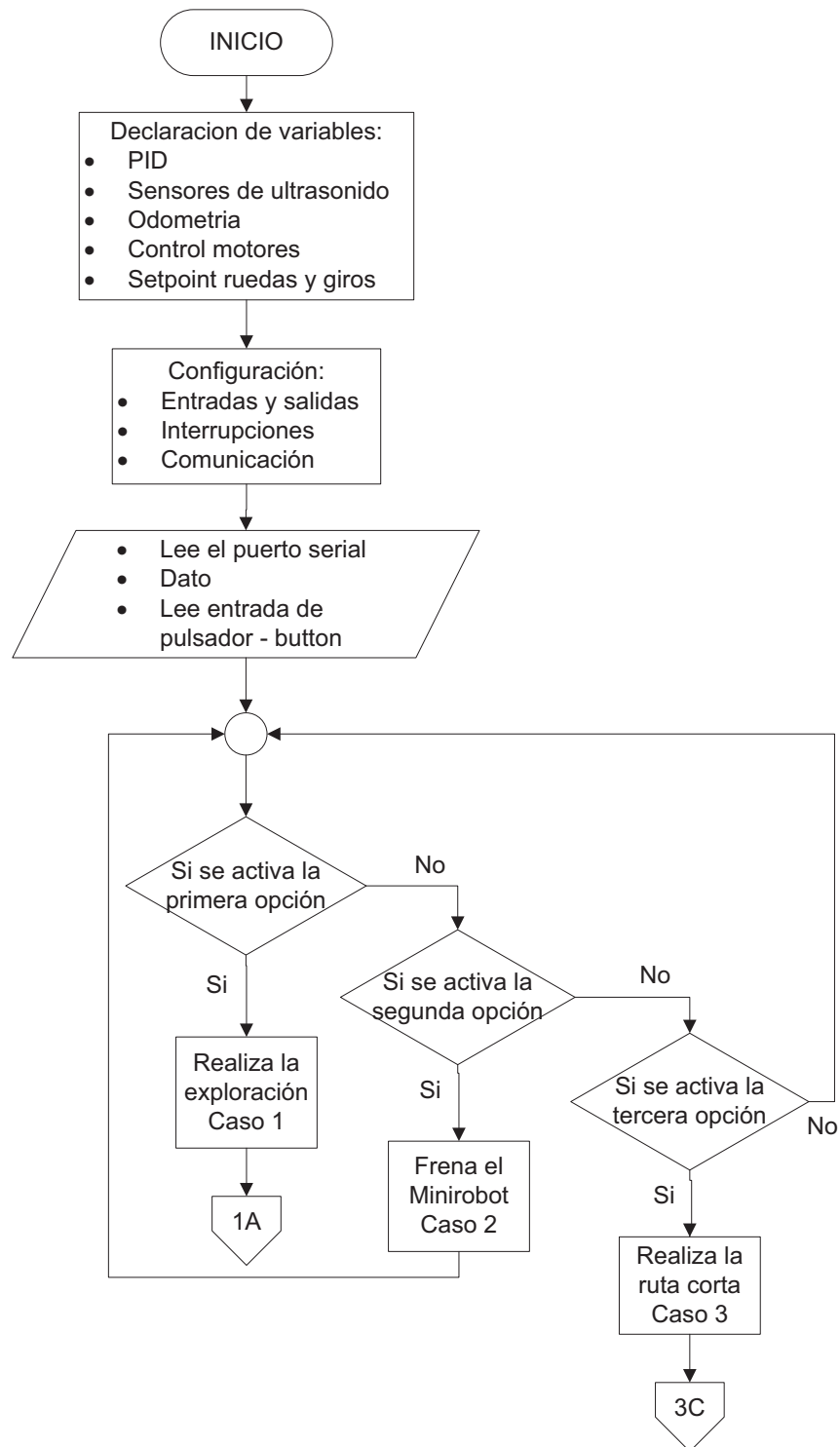


Figura 3.28 Algoritmo inicial, establece acciones a realizar

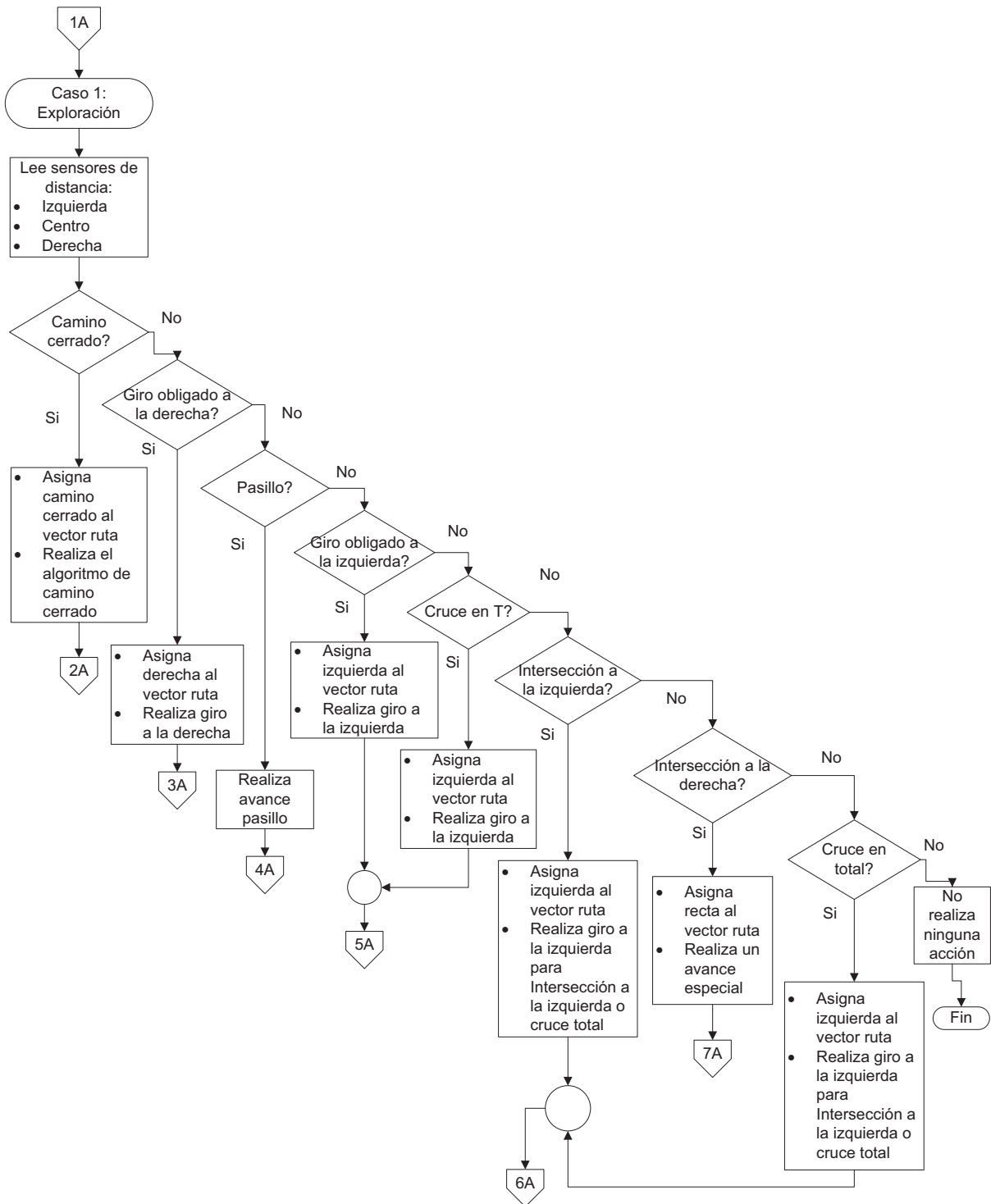


Figura 3.29 Algoritmo de exploración y establecimiento de casos de exploración

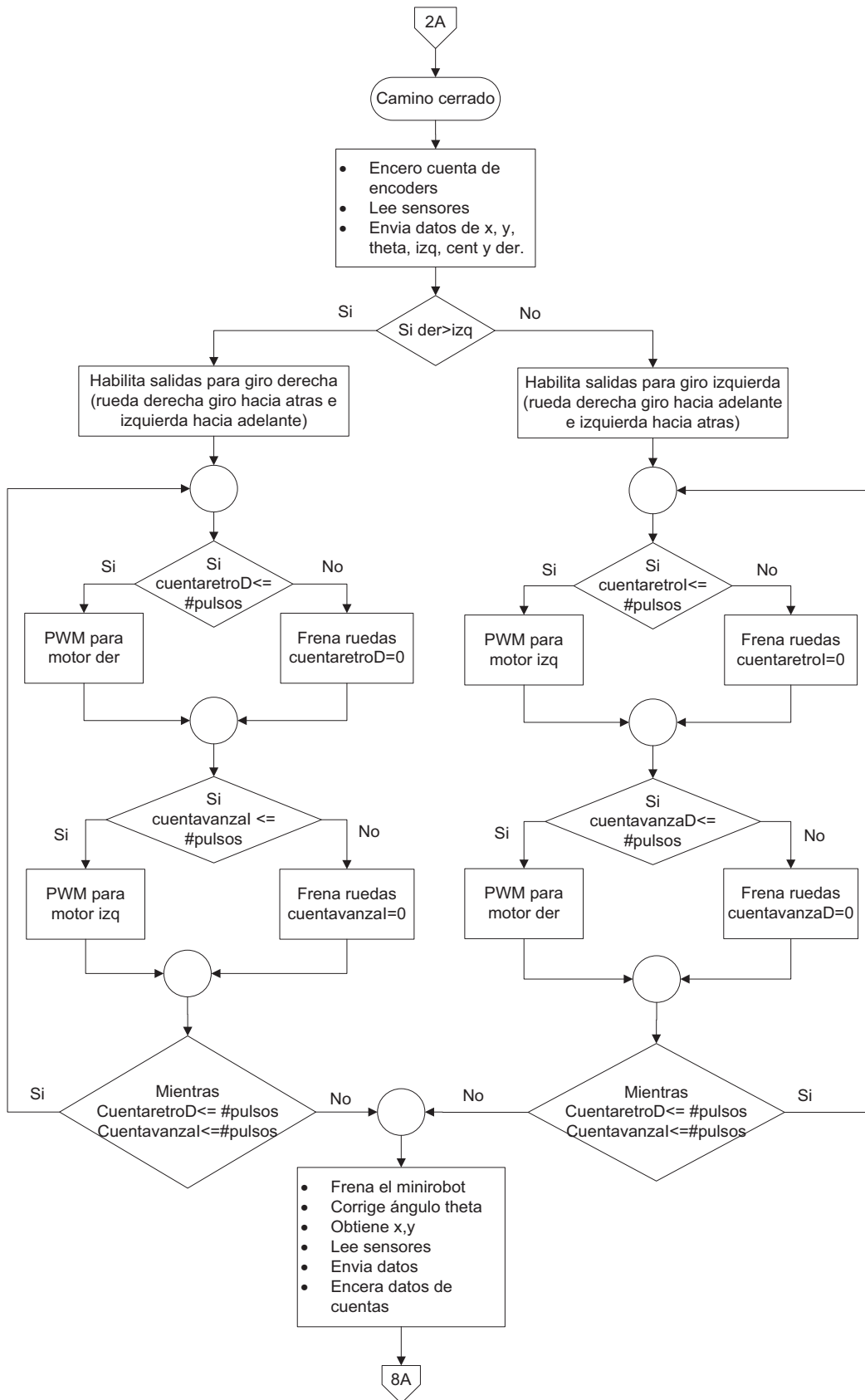


Figura 3.30 Algoritmo de camino cerrado

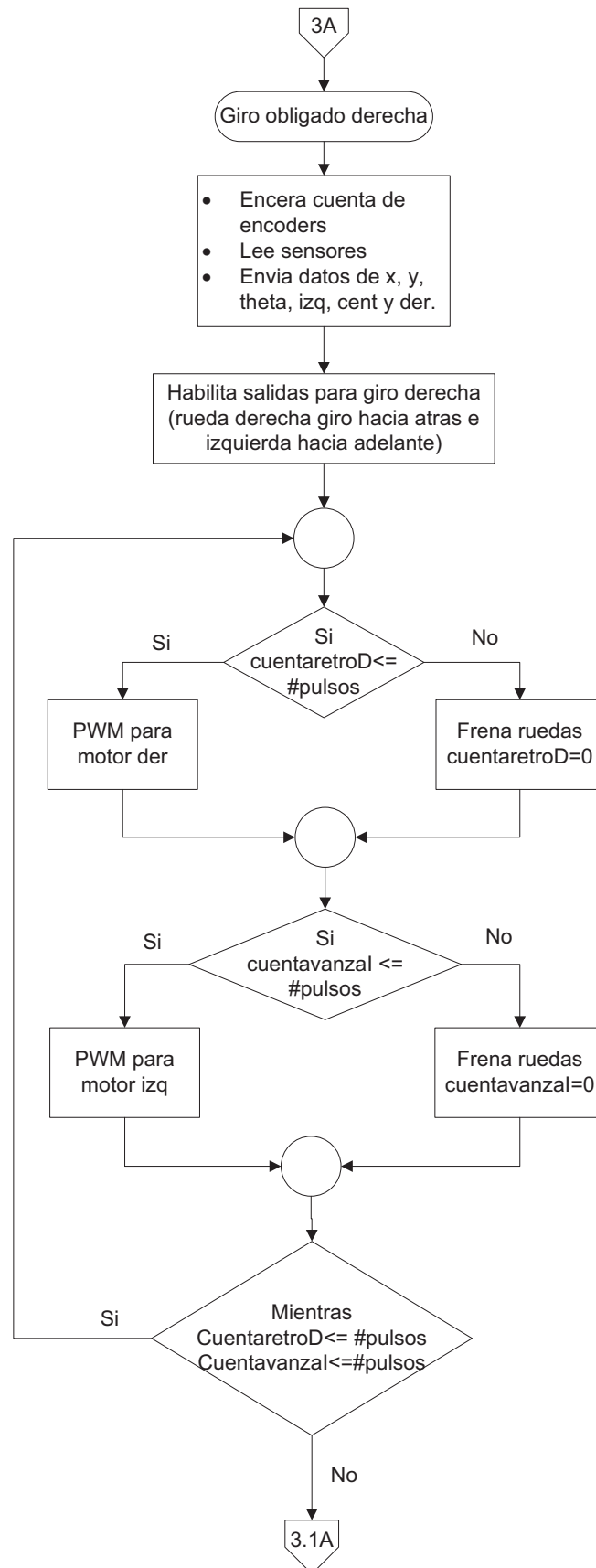


Figura 3.31 Algoritmo de giro obligado derecha (parte 1)

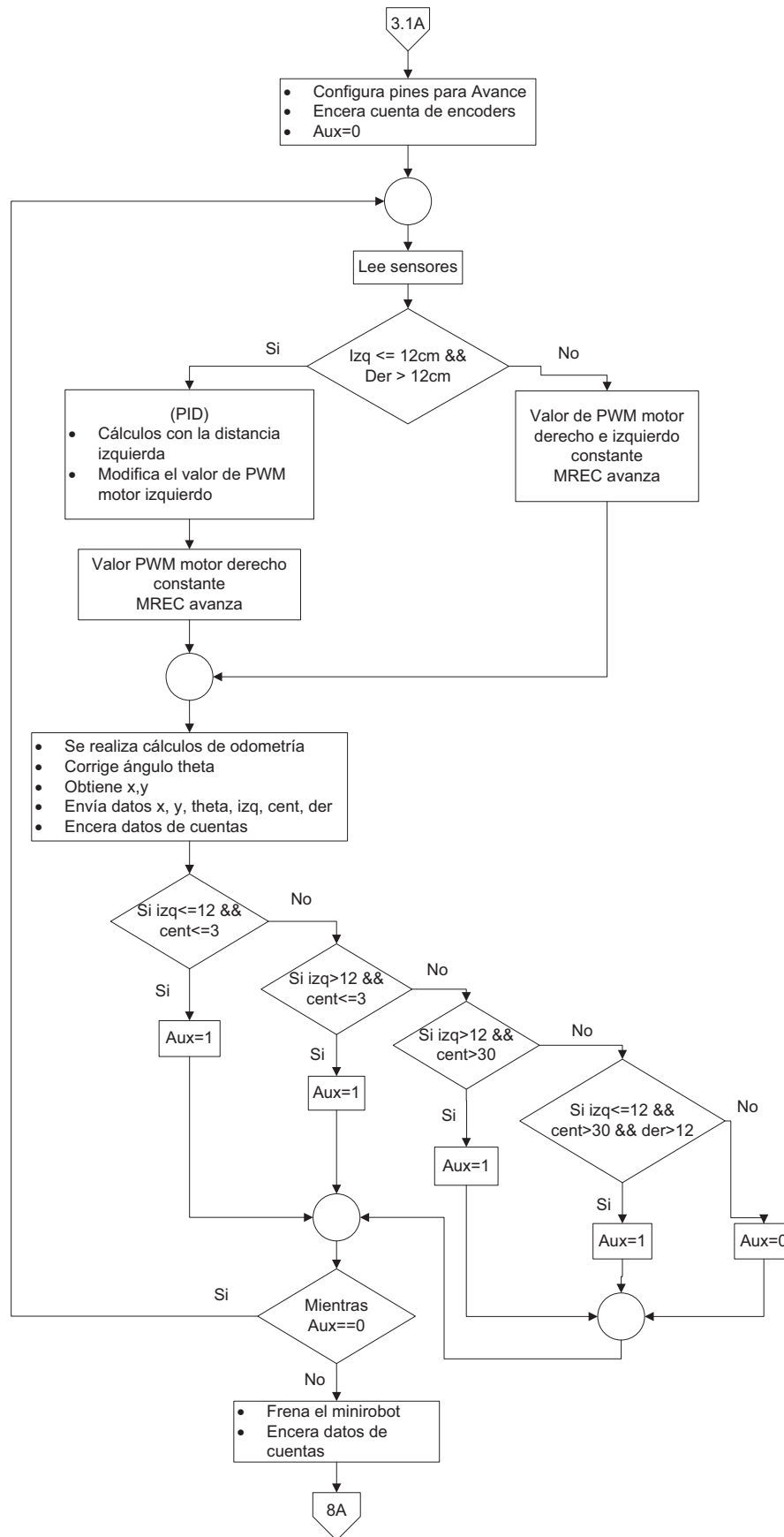


Figura 3.32 Algoritmo de giro obligado derecha (parte 2)

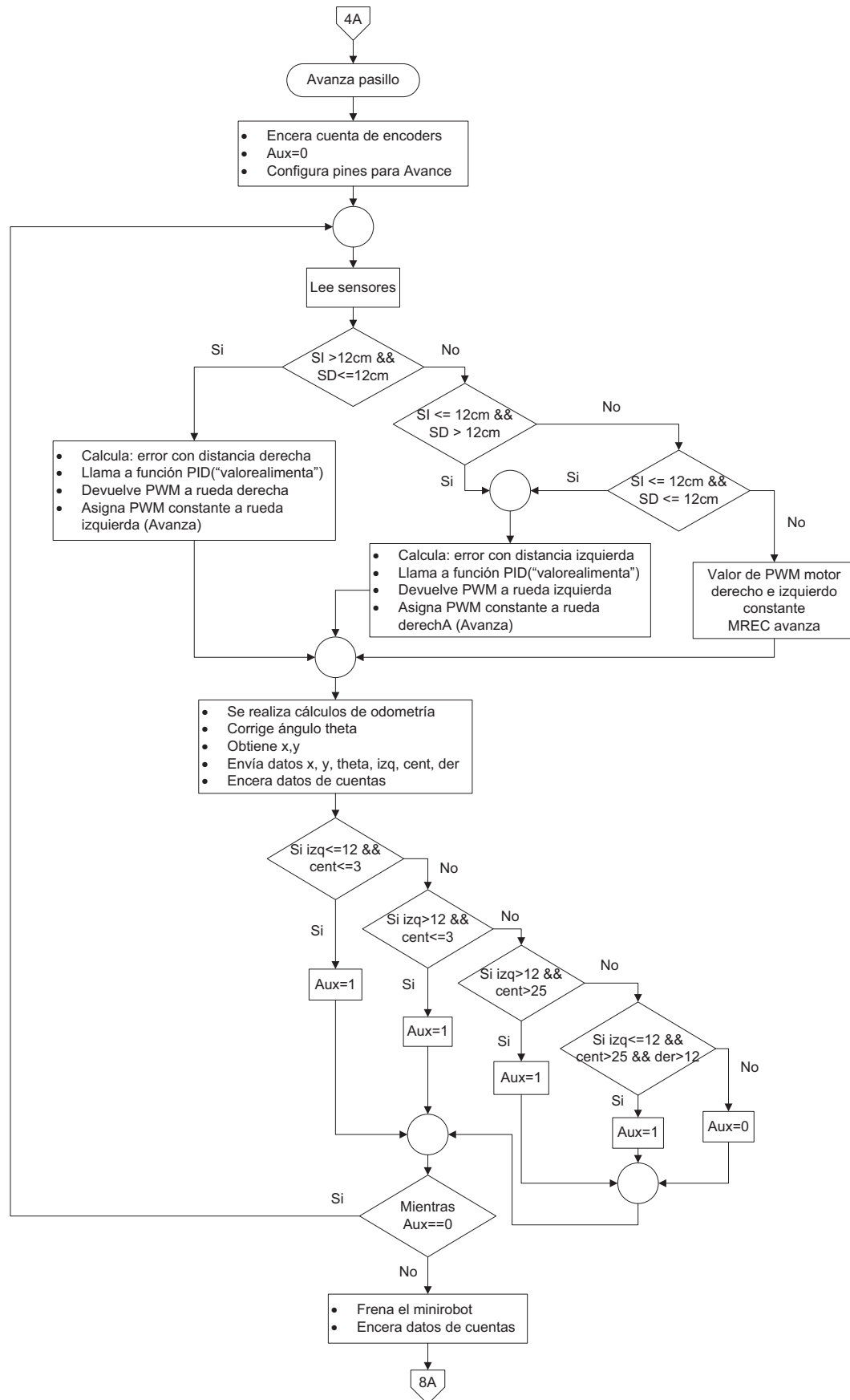


Figura 3.33 Algoritmo de avanza pasillo



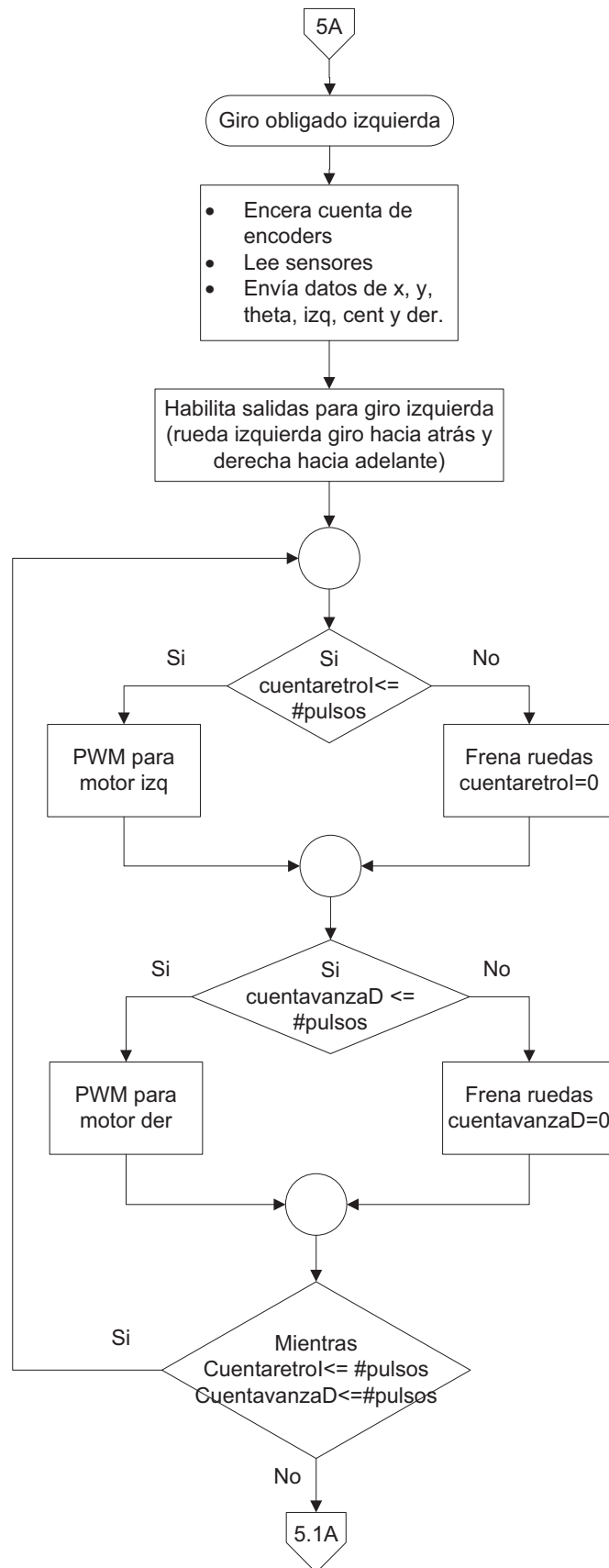


Figura 3.34 Algoritmo de giro obligado izquierda (parte 1)

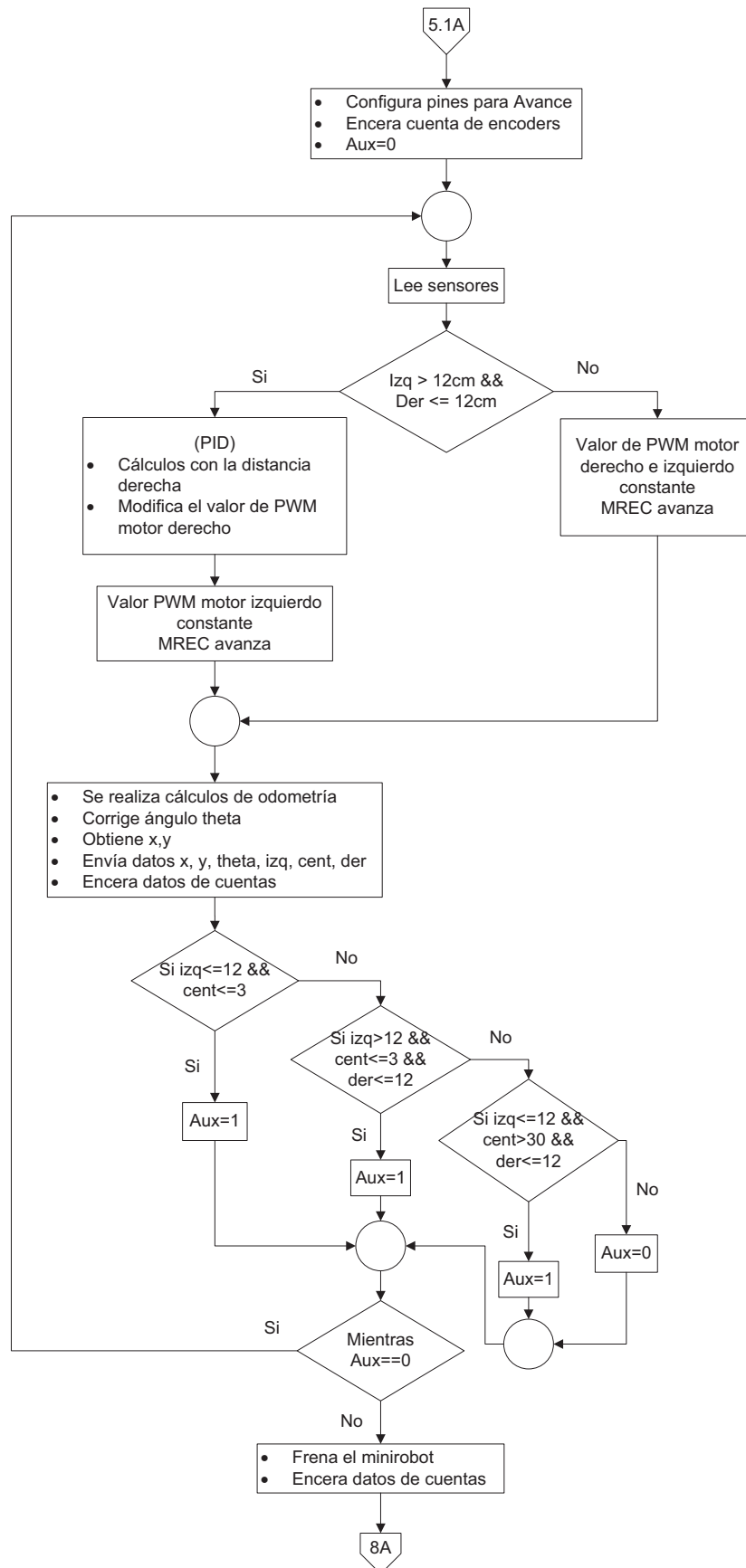


Figura 3.35 Algoritmo de giro obligado izquierda (parte 2)

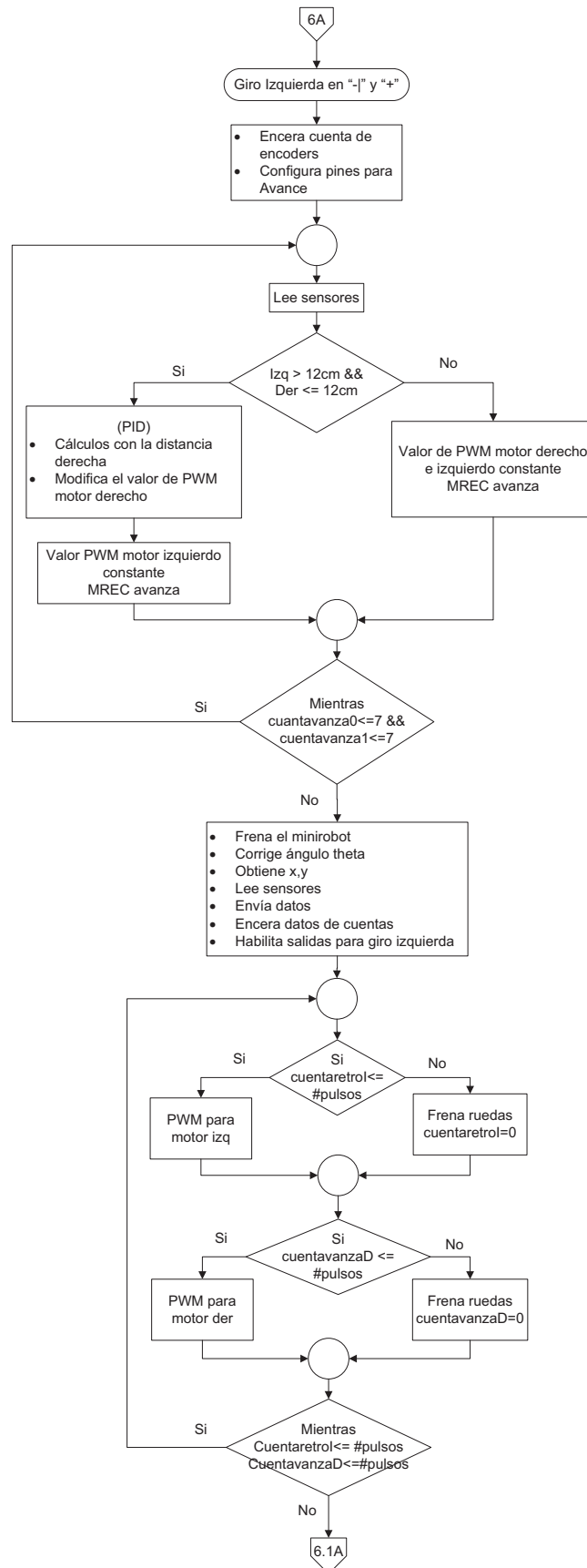


Figura 3.36 Algoritmo de giro izquierda en "-" y "T" (parte 1)

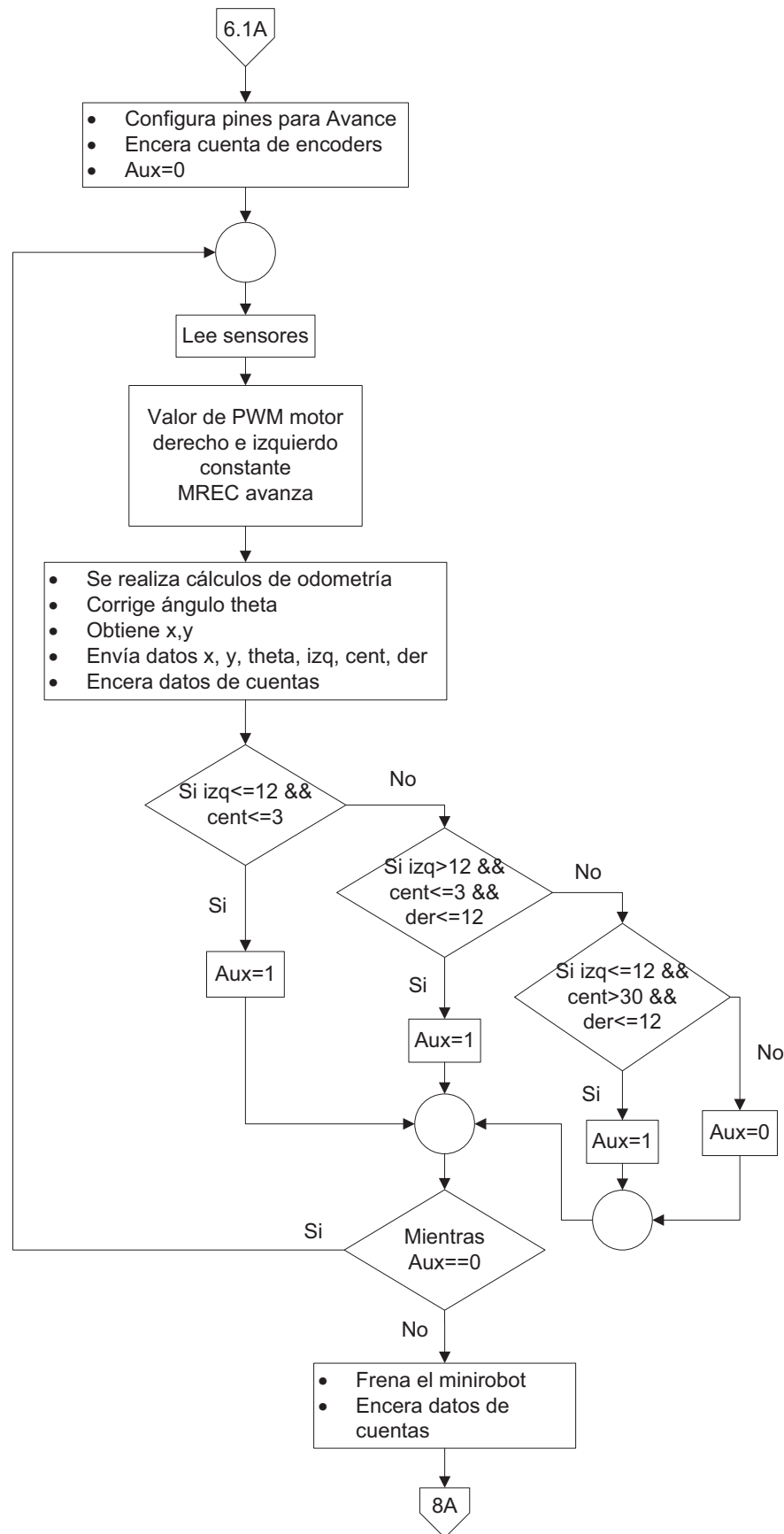


Figura 3.37 Algoritmo de giro izquierda en “-|” y “T” (parte 2)

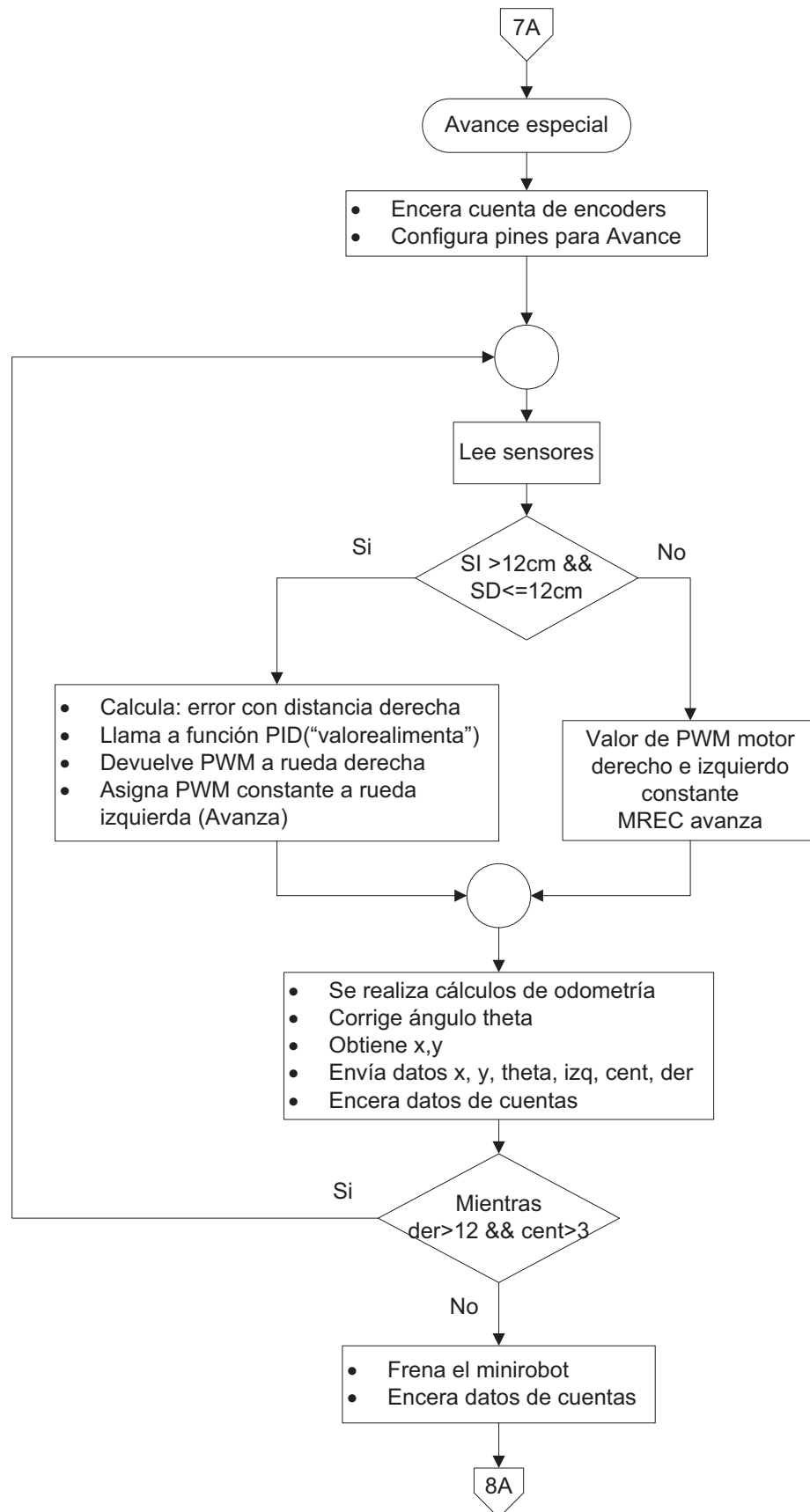


Figura 3.38 Algoritmo de avance especial, intersección derecha

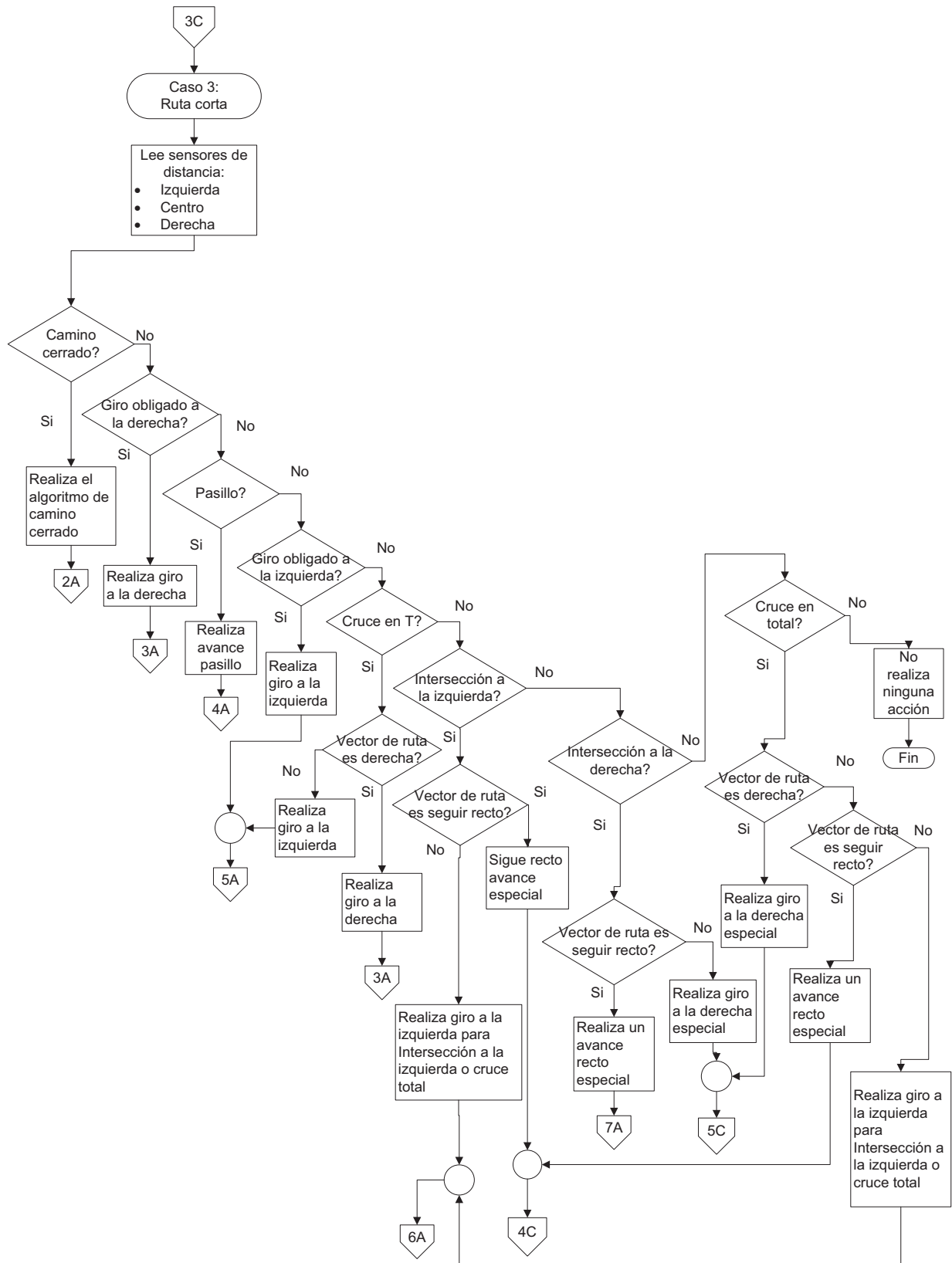


Figura 3.39 Algoritmo de ruta corta y establecimiento de caminos a seguir

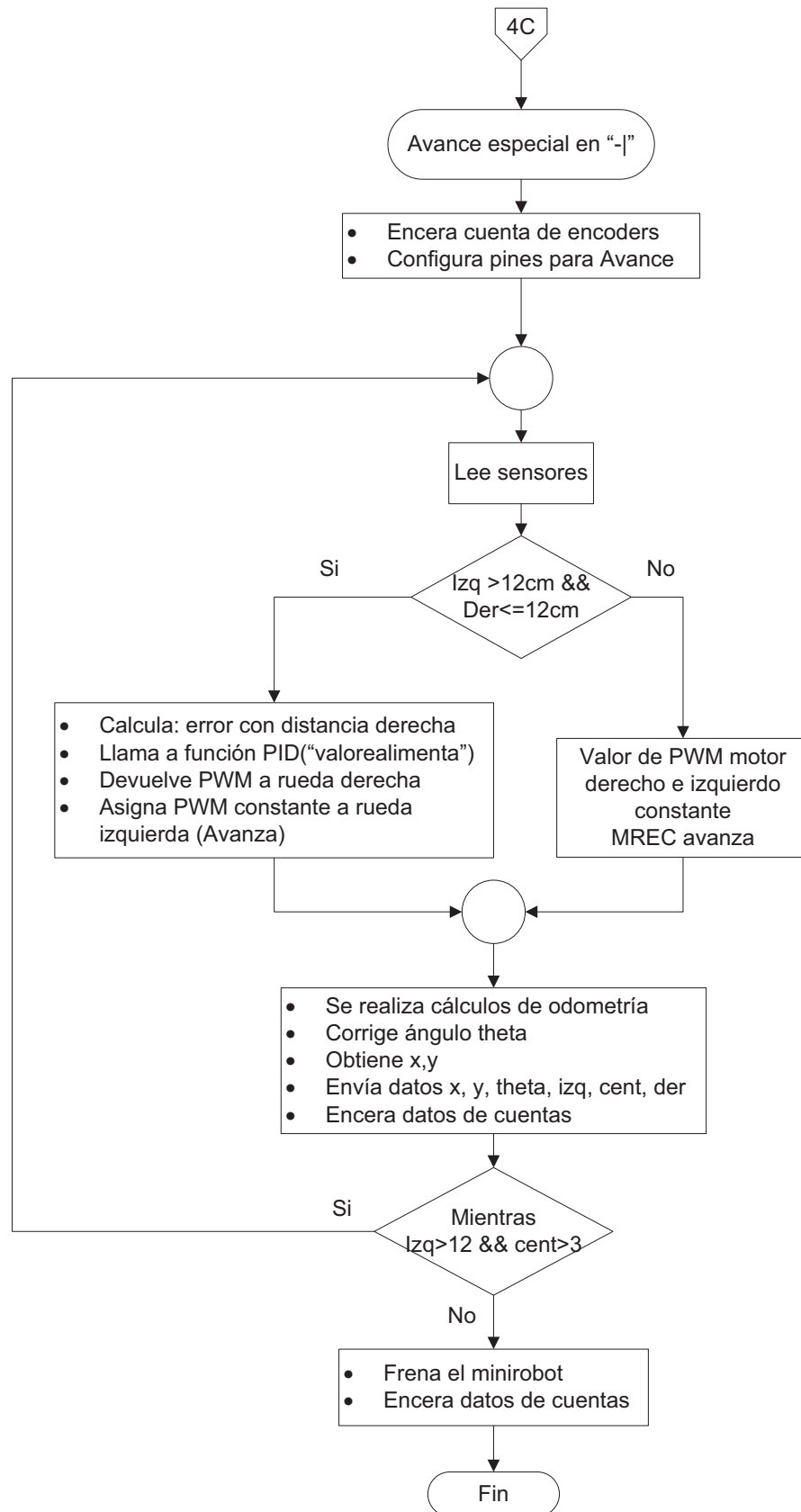


Figura 3.40 Algoritmo de avance especial en “-|” (intersección derecha)

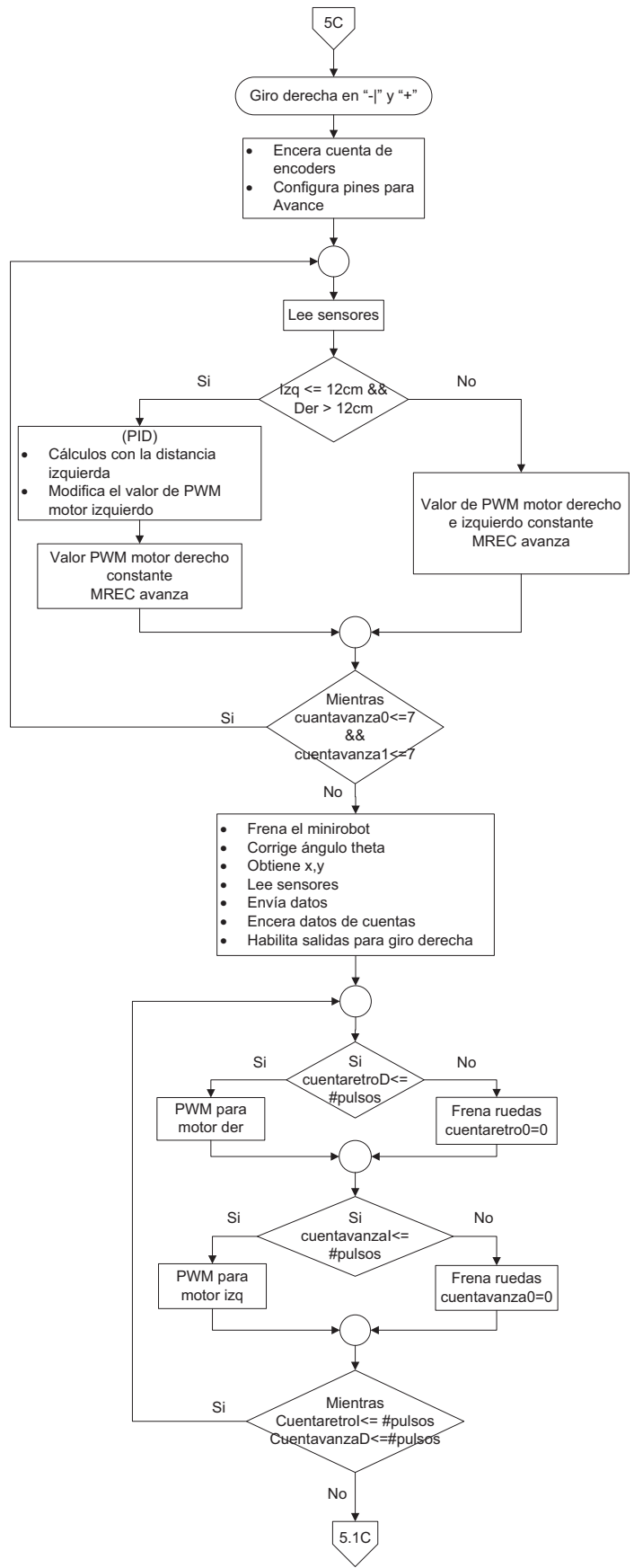


Figura 3.41 Algoritmo de giro derecha en "-" y "+" (parte 1)



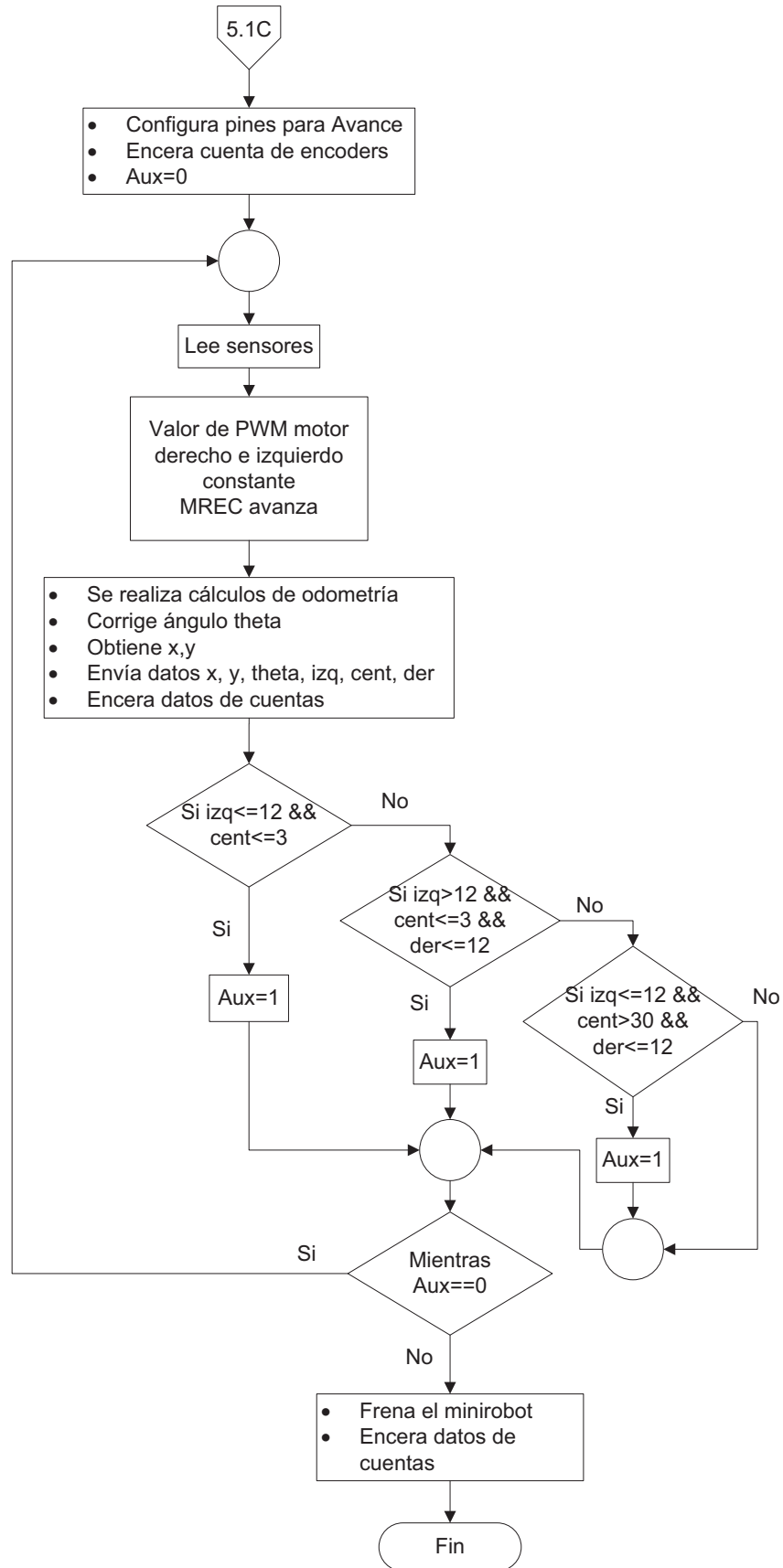


Figura 3.42 Algoritmo de giro derecha en “-|” y “T” (parte 2)

## CAPITULO 4

### PRUEBAS Y ANÁLISIS DE RESULTADOS

Las pruebas se hicieron de acuerdo a como se fue realizando el programa y dependiendo de los cambios realizados durante el proceso de elaboración del Proyecto de Titulación.

#### 4.1 CALIBRACIÓN DE LOS MINIROBOTS

La calibración de los minirobots va a ser en el aspecto de la locomoción de los minirobots, es decir, en la manera de avanzar por los pasillos del laberinto y en los giros para las diversas zonas que se tiene. La locomoción es un aspecto muy importante ya que de esto dependerá una buena exploración, la obtención de datos adecuados para ser procesados luego y la resolución del laberinto obteniendo el camino más corto.

##### 4.1.1 CALIBRACIÓN DEL PI[27]

Un robot móvil para su desplazamiento necesita de un sistema de telemando o de percepción del entorno mediante sensores, para este proyecto de titulación se necesita un buen sistema de percepción del entorno ya que va a atravesar un laberinto donde la distancia entre paredes es pequeña, de 20cm. Para obtener una trayectoria de avance recta es necesario que los minirobots se guíen por las paredes que observan a los costados. Es necesario recordar que los minirobots compensan con la medida del sensor de la izquierda o con la medida del sensor de la derecha, dependiendo de la zona en la que se encuentre dentro del laberinto.

La calibración del PI se hizo en base a varios ajustes de las constantes  $K_p$  y  $T_i$ . Se partió del conocimiento de orden de la magnitud de dichas constantes mediante simulaciones de controladores digitales[25]. Se realizaron cálculos

matemáticos de la señal enviada al PWM para tener idea de la señal de control que va a tener el motor, que se muestran en la Tabla 4.1, con valores iniciales de  $K_p$  y  $T_i$  para una referencia de 5 cm, que es la distancia desde el sensor lateral hacia la pared, obteniendo resultados no muy satisfactorios. Las pruebas realizadas fueron para una compensación con el valor de distancia del sensor izquierdo y el valor de PWM a modificar para el motor izquierdo, mientras que el valor de PWM en el motor derecho es constante.

Tabla 4.1 Cálculos para el control PID

		Valores de PID					
		Kp	Ti	Tm	a=Kp	b=(Kp*Tm)/Ti	
		0,5	0,02	0,01	0,5	0,25	
Medida de sensores		Valores calculados					
Sensor izquierdo	Sensor derecho	Error	Valor proporcional (a*error)	Valor integral (b*error + valor de la integral anterior)	Valor total (Valor. proporcional + valor integral)	Valor PWM (PWM + Valor. total)	Integral anterior
4	6	1	0,5	0,25	0,75	40,75	0,25
4	6	1	0,5	0,5	1	41	0,5
5	5	0	0	0,5	0,5	40,5	0,5
6	4	-1	-0,5	0,25	-0,25	39,75	0,25
7	3	-2	-1	-0,25	-1,25	38,75	-0,75

Como la prueba que se muestra en la Tabla 4.1 se realizaron más pruebas del mismo tipo, obteniendo resultados poco satisfactorios, pero permitieron validar las señales que se envían al PWM y finalmente hacer el ajuste del controlador. Se realiza solo un controlador PI ya que la acción de control derivativa al facilitar una respuesta más rápida puede producir un mal posicionamiento del minirobot dentro del laberinto lo que ocasiona una toma de mediciones erróneas.

Se utilizaron estos valores en el programa del arduino y se notó claramente que el control no era el adecuado, por lo que se procedió a variar los valores de  $K_p$  en

un principio y se mantuvo el mismo valor de  $Ti$ . La prueba se realizó acercando el minirobot mucho a la pared izquierda. Obteniendo los siguientes resultados:

- $Kp = 0,5$  → Compensa, se corrige la posición pero no se estabiliza en el centro y se acerca a la pared derecha hasta chocarse.
- $Kp = 0,1$  → No compensa, la corrección de la posición fue mínima.
- $Kp = 0,8$  → Compensa, corrige la posición pero su respuesta es muy lenta.
- $Kp = 1,2$  → Compensa, corrige la posición mejorando la respuesta.

Con los resultados anteriores se observa que mejora el control aumentando el valor de la constante proporcional, y se optó por realizar modificaciones solo al valor proporcional, para luego encontrar el valor adecuado para la constante integral. Se varió la constante de acción de control proporcional  $Kp$  hasta llegar al valor de 10 que fue el adecuado, es decir, un valor con el cual sea notoria la corrección de la trayectoria y que se estabilice de buena manera.

La acción de control proporcional corrige el valor de PWM pero muchas veces se pasa del valor a corregir o no llega al valor de corrección, debido a esto se implementó la acción integral, la cual a través del tiempo va a seguir corrigiendo la velocidad de la rueda izquierda hasta que el minirobot avance por el centro del pasillo, para ello, una vez establecido el valor de  $Kp$  se empezó a variar el valor de  $Ti$  hasta encontrar el valor adecuado de 50. Los valores con los que se trabajó inicialmente no fueron los adecuados pero al realizar el ajuste final se obtuvo el comportamiento deseado.

El control que se implementa es un control PI, capaz de corregir su posición, estabilizarse de buena manera y en un tiempo adecuado, es importante que el control no sea demasiado rápido ni brusco ya que en su afán de corregir su posición el minirobot puede girar demasiado y obtener medidas erróneas de los sensores de distancia, haciendo que el control trabaje inadecuadamente

#### 4.1.2 CALIBRACIÓN DE LOS GIROS DEL MREC

Los giros fueron un aspecto muy crítico en un principio, ya que se necesita tener giros de 90 y 180 grados aproximadamente. Se inició colocando giros controlados por tiempo, es decir que se daba la acción de giro y se esperaba de un retardo dentro del arduino. De esta manera los giros siempre eran mayores o menores a 90 y a 180 grados, ya que la variable de tiempo es inadecuada, y no se obtuvo buenos resultados debido a que los giros eran tan imprecisos que los minirobots se chocan con las paredes, no precisamente al momento de girar sino al momento de avanzar; y esto debido a que los sensores no quedan perpendiculares a las paredes del laberinto, donde se tiene una correcta lectura de la distancia, sino quedan desubicados y esto provoca lecturas erróneas.

Para que los giros sean precisos se realizó giros controlados por los encoders de cuadratura, estos encoders brindan una cantidad específica de pulsos por cada giro completo de la rueda, con este principio se condicionó que las ruedas giren una cantidad de pulsos determinada y se detengan. Por ejemplo para un giro a la derecha se hizo que la rueda derecha gire hacia atrás 4 pulsos de encoder derecho y que la rueda izquierda gire hacia adelante 4 pulsos de encoder izquierdo y luego se detienen ambas ruedas. Con esto se consigue giros bastante exactos, para un giro de 180 grados solo se aumenta la cantidad de pulsos de cada rueda.

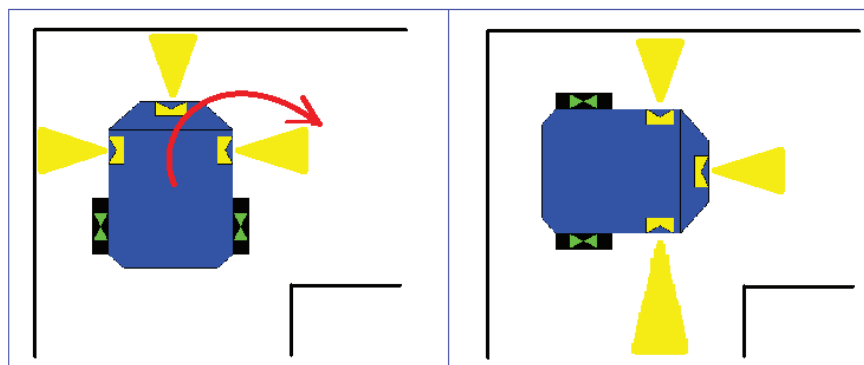


Figura 4.1 Giro correcto a la derecha, lectura de sensores adecuado

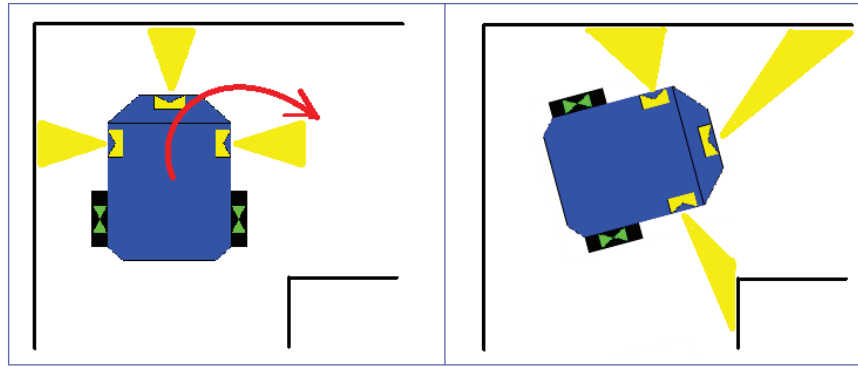


Figura 4.2 Giro incorrecto a la derecha, medida de sensores errónea

La manera de calibrar los giros es muy simple, ya que es necesario controlar la cantidad de pulsos de encoder por giro realizado, es decir, en el programa se puede establecer 3, 4 o 5 pulsos de encoders por giro. Al realizar las pruebas se observa que con 3 pulsos los giros se aproximan a 90 grados, al probar con 5 pulsos de encoder los giros superan los 90 grados y con 4 pulsos se obtiene un giro muy aproximado a los 90 grados.

#### 4.1.3 AVANCE DEL MREC POR DIVERSAS ZONAS

Como se menciona en el Capítulo 3 el laberinto tiene varias zonas de exploración, las cuales se las puede observar en la Figura 3.8. Es importante aclarar esto debido a que una buena exploración depende del reconocimiento y el traslado a través de éstas zonas.

Los minirobots tienen que ser capaces de reconocer la zona en la que se encuentran y desplazarse por ella hasta salir de esa zona y entrar en otra, esto se debe a que los giros que el minirobot realiza pueden hacerle percibir que entra en otra zona, cuando aún sigue en la primera zona detectada.

En la Figura 4.3 se muestra que el minirobot detecta un cruce en “T”, gira hacia la izquierda y avanza hasta encontrar una nueva zona que va a ser el avance pasillo, esta es la manera adecuada de detectar una zona, salir de ella y encontrar una nueva zona.



Figura 4.3 Giro y avance adecuado a la izquierda en la zona cruce en "T"

En la Figura 4.4 se muestra el mismo ejemplo anterior pero con un avance inadecuado, el minirobot detecta un cruce en "T", gira hacia la izquierda y en lugar de avanzar toma los valores de los sensores de distancia, éstos le dan una percepción de una nueva zona la cual va a ser una intersección a la izquierda, y en lugar de avanzar el minirobot va a volver a girar hacia la izquierda y va a avanzar hasta encontrar una nueva zona que es el pasillo, como se puede observar el minirobot toma una decisión inadecuada y regresa por el mismo camino por el cual entró.

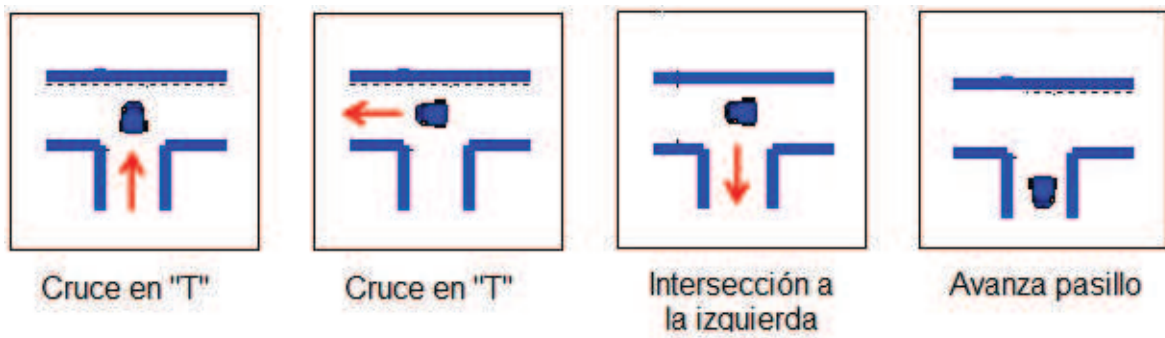


Figura 4.4 Zona detectada con avance inadecuado

Para realizar unos avances correctos por las diversas zonas del laberinto lo que se hizo es hacer pruebas para cada caso y establecer por medio de programación condiciones adecuadas para que el minirobot detecte una zona y sea capaz de salir de esta. En casos como el mencionado es importante que se den los

avances luego del giro, estos avances en un principio se los realizó por cuentas de encoders de las ruedas pero daban problemas, ya que el minirobot pierde la percepción del entorno por el mismo hecho de avanzar con las cuentas de los encoders, debido a esto se realizó algoritmos de control que permiten avanzar y detectar por medio de los sensores de distancia una nueva zona, de esta manera no se pierde la percepción del entorno.

Dentro de esta sección implícitamente esta el avance por diversas zonas con el algoritmo de la mano izquierda, es decir, se calibra de manera adecuada los minirobots para que tengan unos buenos avances pero que se manejen por el algoritmo de la mano izquierda en zonas donde los minirobots tienen que decidir. Para ello las pruebas que se realizaron fueron visuales, se coloca a uno de los minirobots en una entrada del laberinto y se observa que avance de buena manera y que en las zonas de decisión escoja los caminos de acuerdo al algoritmo de la mano izquierda.

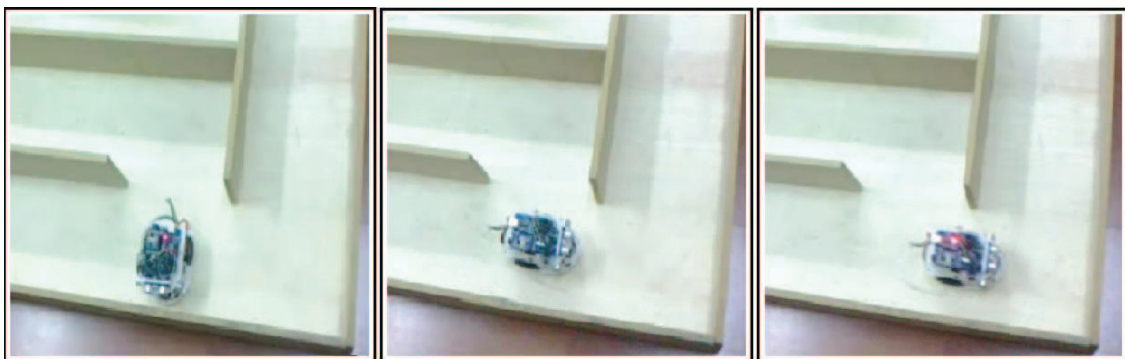


Figura 4.5 Pruebas de exploración

## 4.2 PRUEBAS DE RED ZIGBEE

Para la comunicación con la red zigbee se realizaron varias pruebas entre 2 módulos xbee es decir una comunicación punto – punto y con los 4 módulos xbee que consta en el proyecto de titulación haciendo una comunicación punto – multipunto con un coordinador y 3 routers.



## 4.2.1 ENVIAR Y RECIBIR DATOS PUNTO – PUNTO

Una red zigbee siempre necesita de un coordinador y en este caso se une con un router o con un end device. Para las pruebas que se realizaron se utilizó un router, para enviar datos del minirobot y para recibir la orden de empezar o detenerse.

La configuración para este tipo de comunicación se la presenta en la Figura 4.6:

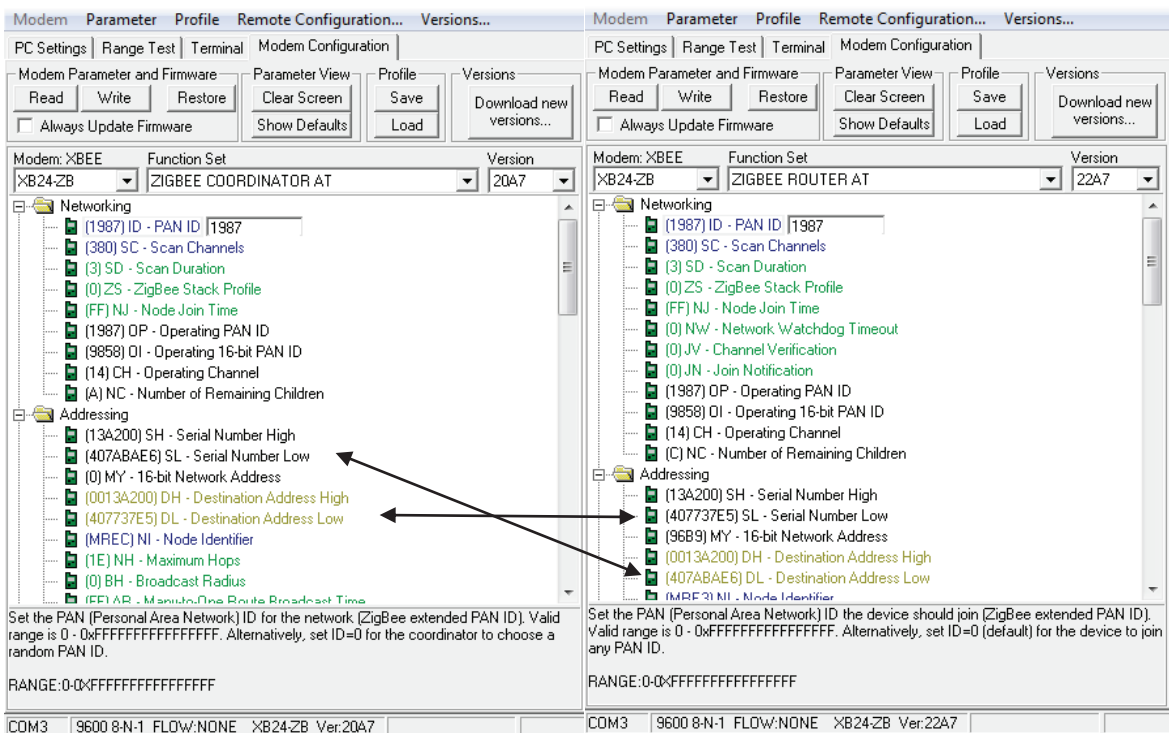


Figura 4.6 Configuraciones de Coordinador y Router

Como se observa en la Figura 4.6, la principal configuración para realizar una comunicación punto – punto, es que la dirección del coordinador es la dirección de destino del router y viceversa, es decir los números SH y SL del coordinador se los coloca en DH y DL del router, y los números SH y SL del router se los coloca en DH y DL del coordinador.

Para las pruebas se utiliza la pestaña Terminal del X – CTU, ahí los datos en color azul son los datos enviados y los datos en color rojo (Figura 4.7) son los datos recibidos.

En la primera prueba se comunicó distintos datos desde el coordinador que esta conectando en el puerto 3 (COM 3) y el router que está conectado en el puerto 4 (COM 4). En la Figura 4.7 se observan los datos enviados y recibidos por la red zigbee.

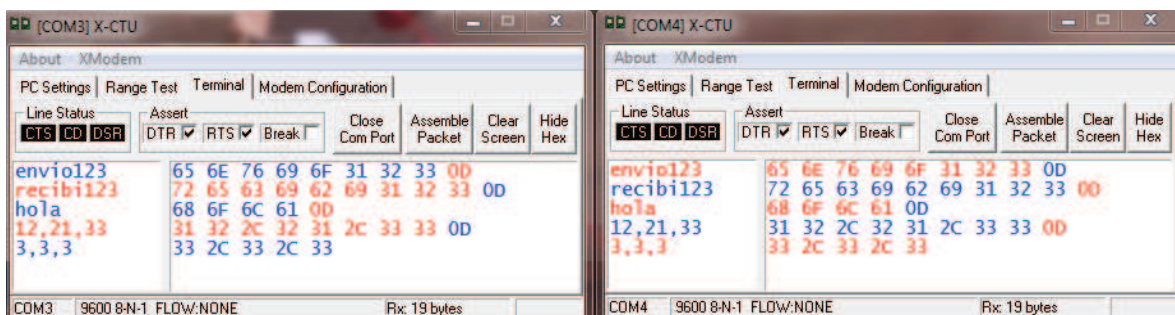


Figura 4.7 Comunicación punto – punto con distintos datos

En la segunda prueba se comunicó al minirobot con el coordinador para obtener los datos de odometría (x, y, theta) y los valores de sensores ultrasónicos (izquierda, centro, derecha).

Cuando el coordinador envía la letra 'b' el MREC empieza la exploración y envía los datos, cuando se quiere detener al minirobot se debe enviar nuevamente la letra 'b', esto es debido a la programación y las condiciones hechas dentro del arduino.

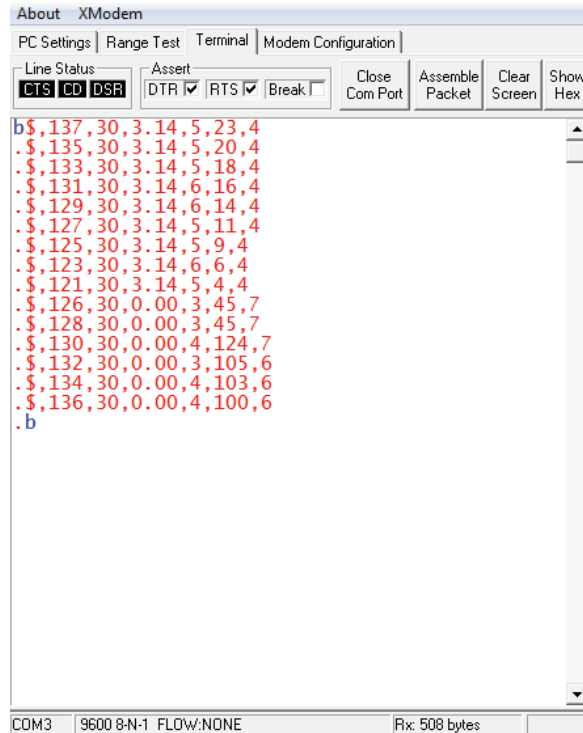


Figura 4.8 Comunicación punto – punto con datos del minirobot

#### 4.2.2 RECIBIR Y ENVIAR DATOS PUNTO – MULTIPUNTO

Para una conexión punto – multipunto, es decir, un coordinador y 2 o más routers o end device's, la configuración es distinta, ya que se tienen varias direcciones de destino a los cuales el coordinador se debe comunicar, y también todos los routers se deben comunicar con el coordinador, debido a esto, la dirección que se debe poner en el coordinador es una dirección broadcast, para configurar esta dirección se debe colocar en DH = 0x0000 y en DL= 0xFFFF.

De la misma manera en los routers se debe colocar la dirección DH = 0x0000 y DL = 0x0000, como se observa en la Figura 4.9, la configuración está hecha entre el coordinador (COM 3) y un router (COM4).

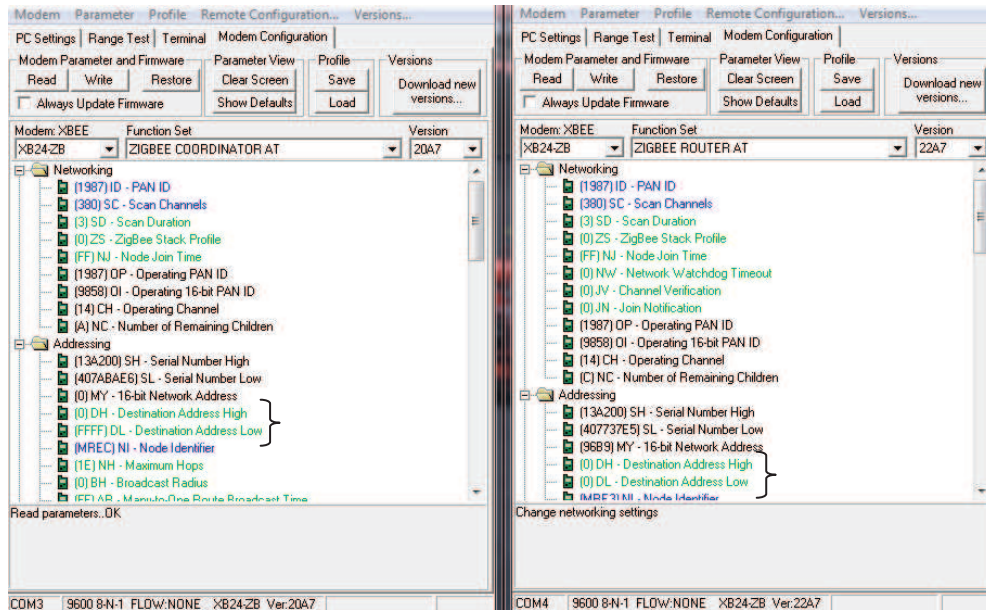


Figura 4.9 Configuraciones de Coordinador y Router

Para las pruebas se tomaron los datos de los 3 minirobots, en la figura a continuación se puede ver los datos enviados en color azul ('a', 'b', 'c') y los datos recibidos en color rojo, datos de odometría (x, y, theta) y valores de los sensores de ultrasonido (izquierda, centro, derecha).

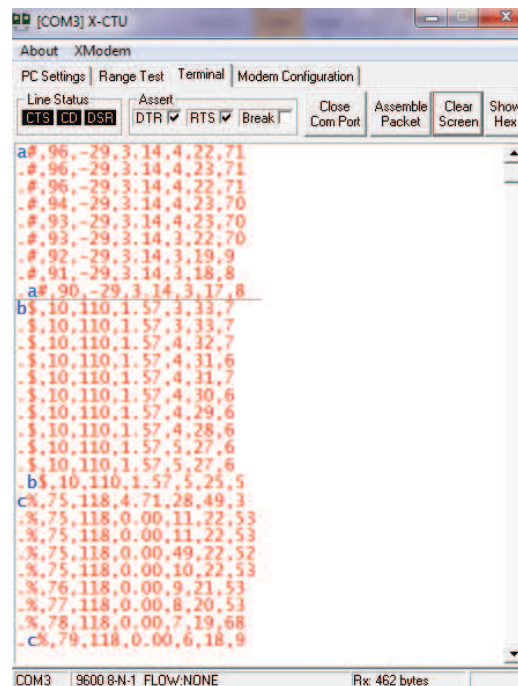


Figura 4.10 Comunicación punto – multipunto con los 3 minirobots

### 4.3 PRUEBAS DE EXPLORACIÓN Y MAPEAMIENTO

Para las pruebas de exploración y mapeamiento se enfocó en un buen recorrido de los minirobots por diversas zonas, que cumplan con el algoritmo de la mano izquierda y además que los datos que proporcionen sean adecuados para poder trabajar con ellos en el MATLAB.

#### 4.3.1 PRUEBAS CON UN MINIROBOT

Se realizaron pruebas para dibujar en el MATLAB la trayectoria del minirobot y su entorno, dentro de una trayectoria recta inicialmente. Los datos obtenidos desde el MREC fueron de odometría, es decir, la posición del minirobots dependiendo del recorrido que hace y de lo que observa en su recorrido que van a ser los valores de medidas obtenidas por los sensores de distancia.

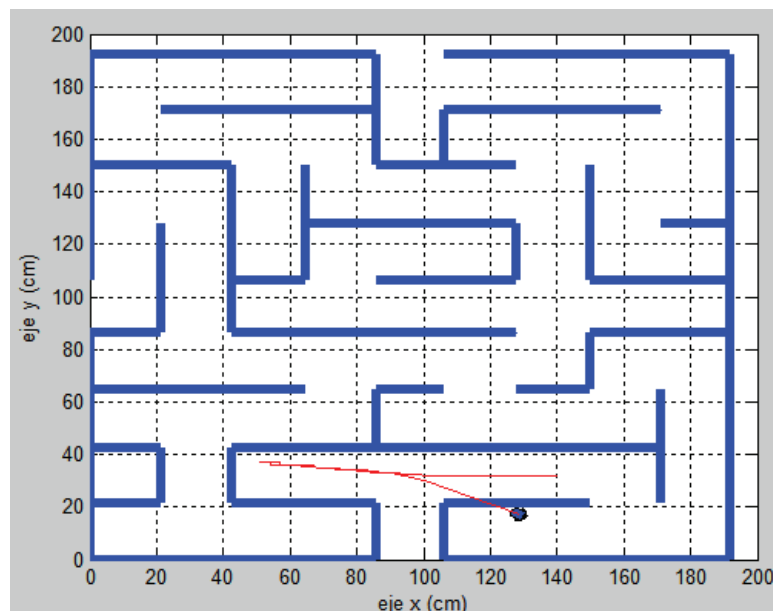


Figura 4.11 Primera prueba de recorrido en trayectoria recta

En esta primera prueba se coloca al MREC en la posición (140, 30) de los ejes x-y respectivamente. El robot avanza con la orientación inicial de 180 grados, por lo

que avanza hasta llegar al punto (50, 30) donde encuentra un “camino cerrado”, el MREC se da la vuelta y regresa al mismo punto. Como se puede observar según los datos ingresados al MATLAB el minirobot al final de su recorrido se llega a cruzar una pared lateral, pero eso en la realidad no fue así, sino que existe un problema que es la acumulación del error por odometría, el cual envía datos con error al MATLAB. Se realizó más pruebas con las mismas condiciones y los errores fueron aleatorios y en muchos casos similares. Estos errores aparecen cuando el MREC trata de estabilizar su recorrido en medio del pasillo.

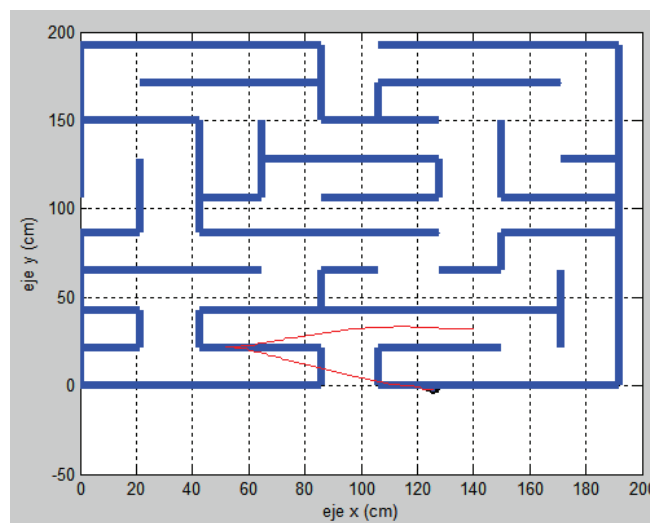


Figura 4.12 Segunda prueba de recorrido en trayectoria recta

En la Figura 4.12 se observa que el error de la trayectoria empieza luego de comenzar el recorrido, y una vez que se da el error se acumula de manera permanente, la trayectoria debe ser prácticamente recta pero los datos enviados muestran que tienen mucho error. Debido a estos errores se realizaron correcciones por software en la parte de odometría, con estas correcciones se mejoró de manera satisfactoria los datos obtenidos para ser enviados hacia el Matlab. Aunque igual existe un cierto error de odometría que ya no es muy grande, por el mismo hecho de utilizar odometría siempre va a existir acumulación de error.

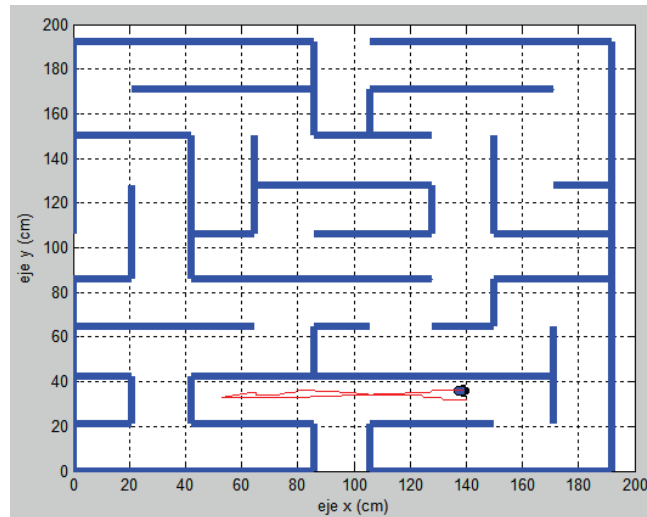


Figura 4.13 Primera prueba con correcciones de odometría por software

En la Figura 4.13 se observa claramente que mejoran los datos de odometría, se puede observar que el MREC casi va por el centro del pasillo, muy similar a como ocurre en la realidad. Las siguientes pruebas son con mayor tiempo de recorrido, es decir que el robot envía más datos y además también realiza giros, ya no solamente una trayectoria recta.

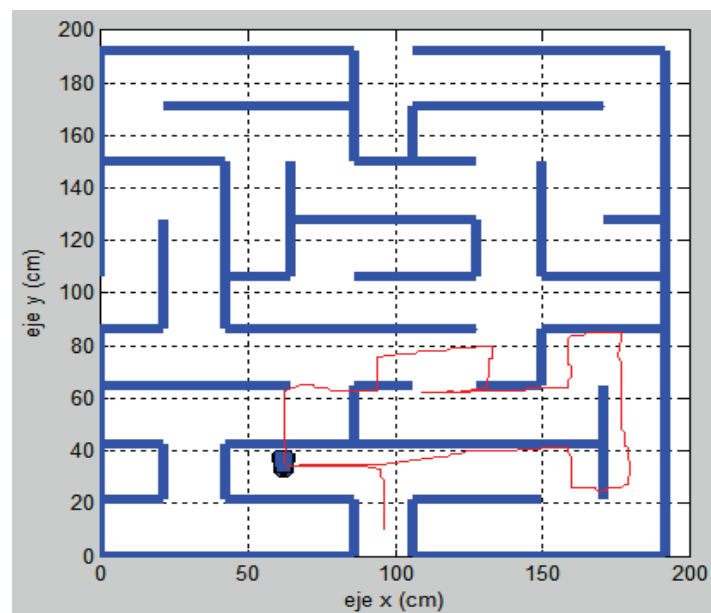


Figura 4.14 Trayectoria del MREC, recorrido con correcciones de odometría

En la figura 4.14 se muestra la primera prueba con más tiempo de envío de datos y con giros, como se puede apreciar en un principio los datos son bastante acertados pero mientras va avanzando el MREC los errores van aumentando, y después de dar los giros parece que el minirobot se monta sobre las paredes. El robot empieza su recorrido en  $(95,10,\pi/2)$  que son los puntos en el eje x,y y el ángulo de orientación. Como se observa los datos no son los adecuados, por lo que se realizó nuevas modificaciones a los cálculos odométricos con el fin de mejorar aun más la trayectoria.

Para mejorar la odometría, es decir, que los cálculos realizados dentro del arduino sean los adecuados se realizó avances de una cierta cantidad de pulsos y se midió cuantos centímetros recorrió por avance, con esto se calibra la resolución del encoder (cm/pulsos), constante que es ingresada en la programación del arduino. La calibración de odometría para cada robot se puede ver en las siguientes tablas:

Tabla 4.2 Calibración de la resolución de los encoders (cm/pulso) para el MREC 1

<b>MREC 1</b>				
<b># pulsos</b>	<b>Distancia recorrida</b>	<b>Dist/# pulsos</b>	<b>Promedio por muestras</b>	<b>Promedio</b>
8	12,00	1,50	1,49	1,38
8	12,50	1,56		
8	11,60	1,45		
8	11,60	1,45		
12	16,80	1,40	1,35	
12	16,60	1,38		
12	16,00	1,33		
12	15,50	1,29		
16	21,50	1,34	1,30	
16	21,00	1,31		
16	20,00	1,25		
16	21,00	1,31		



Tabla 4.3 Calibración de la resolución de los encoders (cm/pulso) para el MREC 2

<b>MREC 2</b>				
<b># pulsos</b>	<b>Distancia recorrida</b>	<b>Dist / #pulsos</b>	<b>Promedio por muestras</b>	<b>Promedio</b>
8	11,30	1,41	1,44	1,34
8	11,80	1,48		
8	11,40	1,43		
8	11,60	1,45		
12	15,20	1,27	1,30	
12	16,00	1,33		
12	16,20	1,35		
12	15,00	1,25		
16	20,30	1,27	1,27	
16	20,00	1,25		
16	20,40	1,28		
16	20,30	1,27		

Tabla 4.4 Calibración de la resolución de los encoders (cm/pulso) para el MREC 3

<b>MREC 3</b>				
<b># pulsos</b>	<b>Distancia recorrida</b>	<b>Dist / #pulsos</b>	<b>Promedio por muestras</b>	<b>Promedio</b>
8	12,10	1,51	1,53	1,40
8	12,60	1,58		
8	12,00	1,50		
8	12,30	1,54		
12	16,00	1,33	1,35	
12	16,30	1,36		
12	16,40	1,37		
12	16,00	1,33		
16	21,20	1,33	1,32	
16	21,00	1,31		
16	21,40	1,34		
16	21,00	1,31		

Luego de realizar las respectivas calibraciones en odometría se procedió a realizar más pruebas de recorrido como la que se muestra en la Figura 4.15.

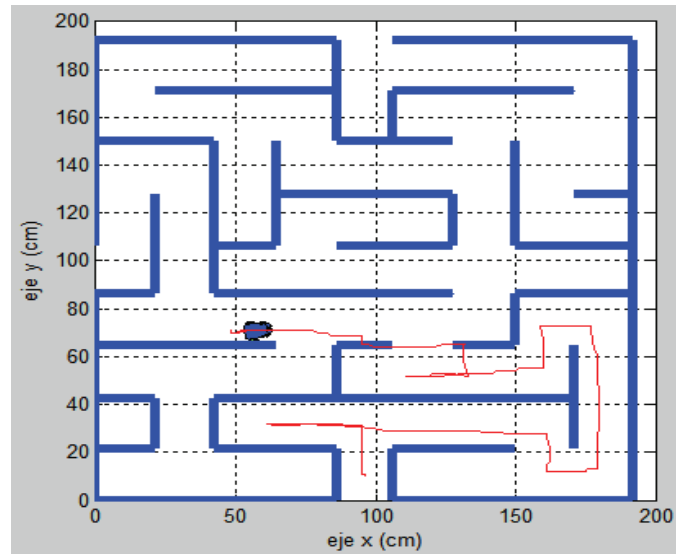


Figura 4.15 Trayectoria del MREC, recorrido con mejoras de odometría

Luego de obtener una trayectoria bastante adecuada se realizó el envío de datos de sensores de distancia, con los cuales en lugar de tener solo la trayectoria y un laberinto ya dibujado en MATLAB. Ahora se tiene la trayectoria y la percepción de los sensores al momento de desplazarse por el laberinto, obteniendo la exploración y visualización de las zonas recorridas.

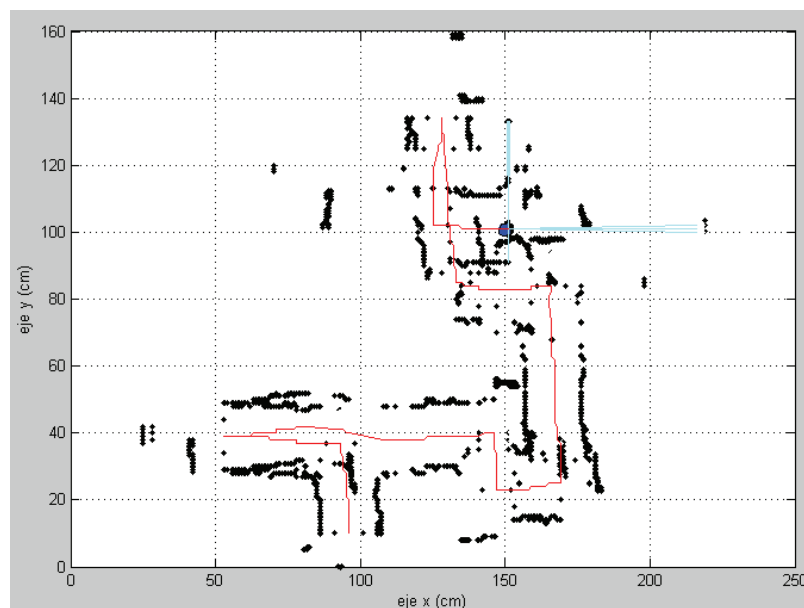


Figura 4.16 Exploración y visualización de la trayectoria y del entorno

Como se puede observar en la Figura 4.16 los datos de trayectoria y percepción de los sensores son bastante buenos pero todavía presentan errores. Estos datos mejoraron mucho con las modificaciones que se realizaron en los algoritmos para avances y recorrido de los minirobots en diferentes zonas y con ciertas correcciones por software en los cálculos odométricos.

En la Figura 4.17, en el gráfico de la izquierda, se muestra un recorrido ideal, es el recorrido que debe hacer uno de los MREC y se muestra el laberinto con las dimensiones reales a escala; y en el gráfico de la derecha se muestra un recorrido real, es el gráfico que se obtiene en MATLAB de las medidas tomadas por el minirobots, trayectoria y medidas de sensores, también a escala. Como se puede observar los gráficos son bastante aproximados, obviamente con las limitaciones de los elementos utilizados.

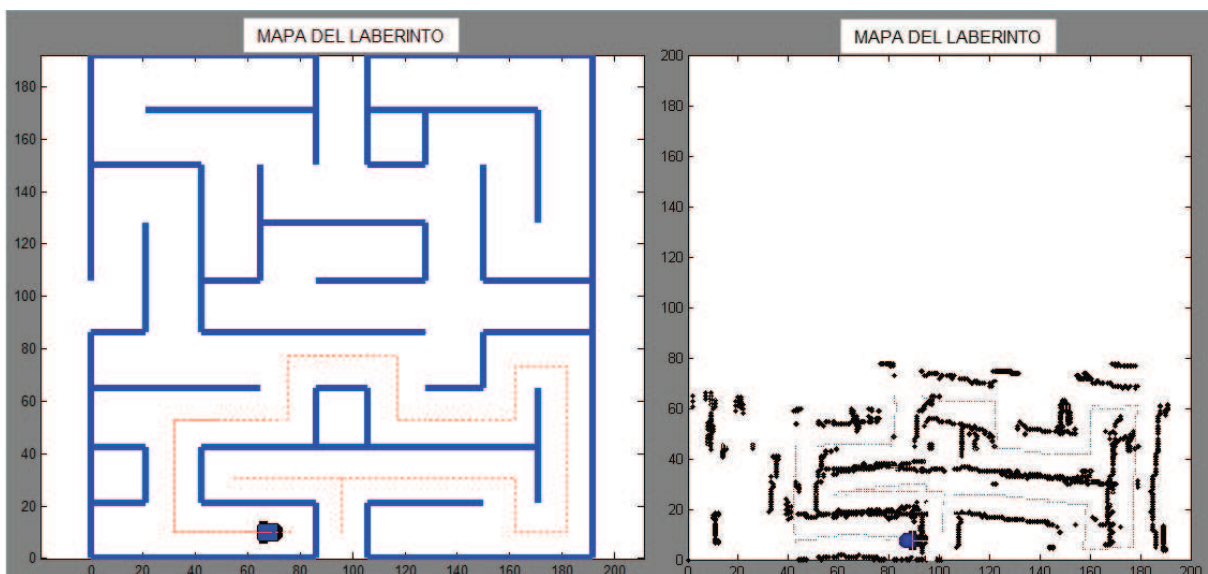


Figura 4.17 Comparación de un recorrido ideal (Izquierda) con un recorrido real y datos reales (Derecha).

### 4.3.2 PRUEBAS MULTIROBOT

Las pruebas hechas con los 3 minirobots constan de 2 partes, la primera parte para comprobar las rutas que siguen en su funcionamiento mediante una

exploración y así contrastar con la simulación hecha en MATLAB y la segunda parte el mapeo de los diferentes sectores del laberinto.

#### 4.3.2.1 Funcionamiento de los 3 minirobots

El funcionamiento dentro del laberinto de los 3 minirobots se lo puede hacer con la ayuda de la ventana de funcionamiento y control de los minirobots (Figura 3.24) que se encuentra en la interfaz grafica de MATLAB (numeral 3.5).

En la figura 4.18 se observa la simulación hecha con las rutas que deben seguir los minirobots dentro del laberinto.

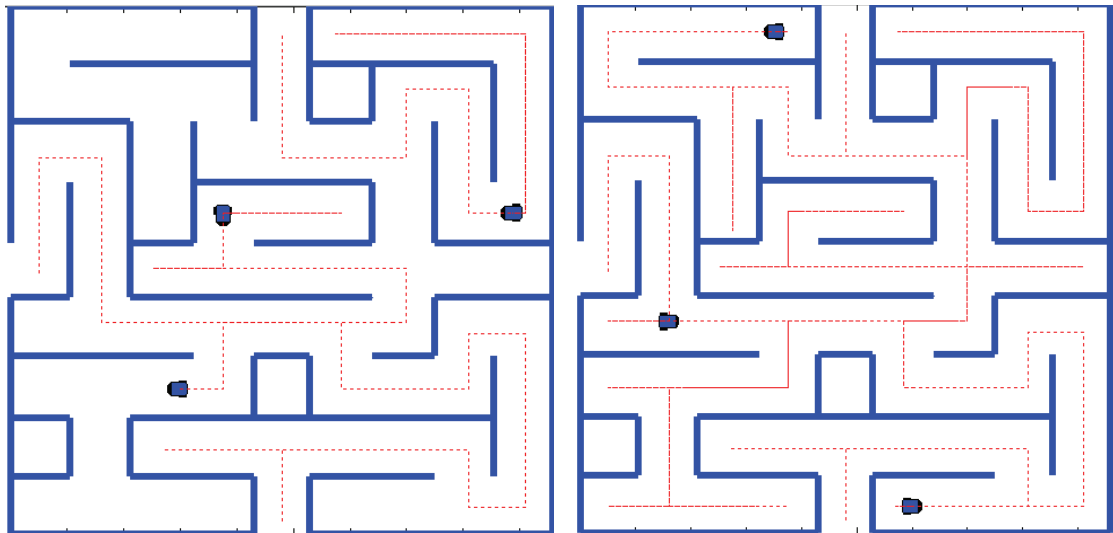


Figura 4.18 Izquierda) Proceso de mapeo Derecha) Mapeo terminado

Cuando un minirobot termina de explorar uno de los sectores del laberinto, se detendrá.

#### 4.3.2.2 Mapeo de sectores diferentes del laberinto

Para el mapeo multirobot de diferentes sectores del laberinto, al mismo se lo ha dividido en 3 sectores, cada minirobot tendrá que terminar su trabajo de explorar y

enviar datos para graficar el sector en la PC para que otro minirobot entre en funcionamiento de la misma manera.

Los resultados de los diferentes sectores mapeados se muestran a continuación, con un gráfico con la ruta y el dibujo del sector ideales y el otro gráfico con datos reales.

➤ SECTOR 1

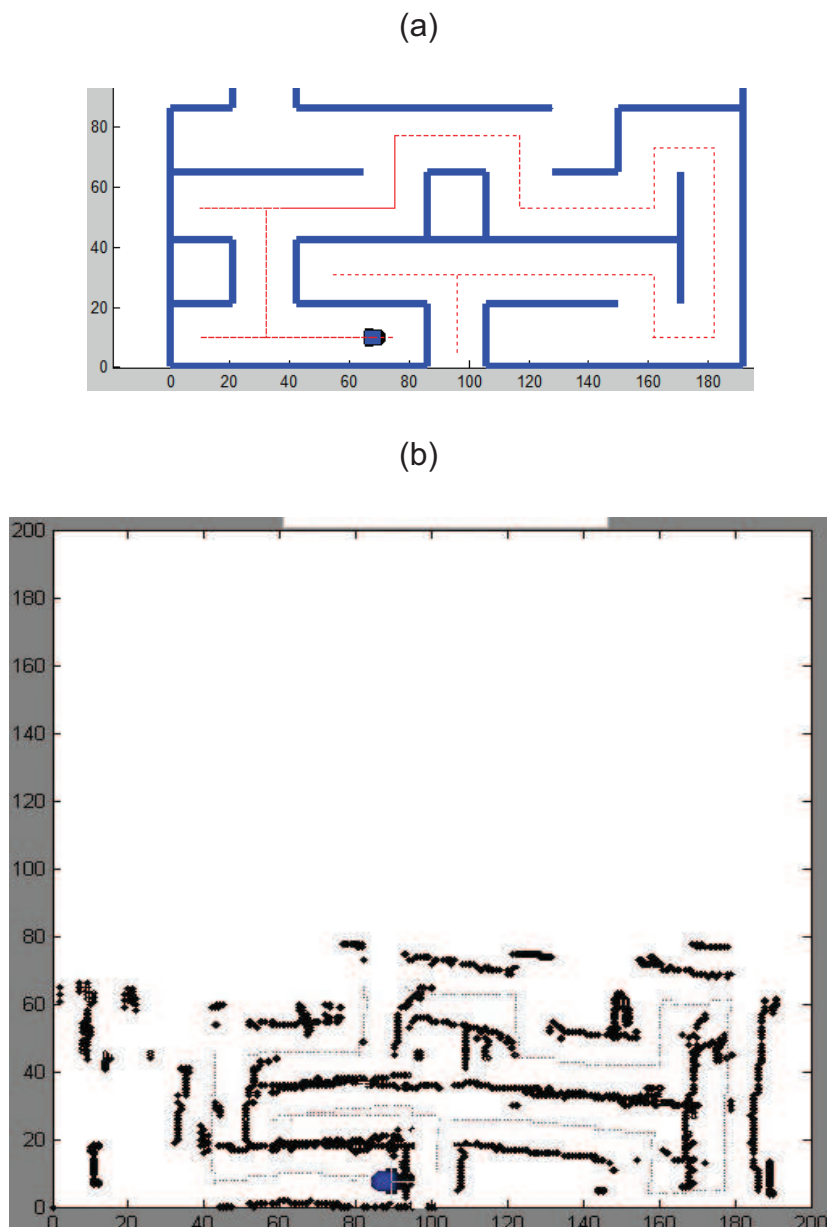
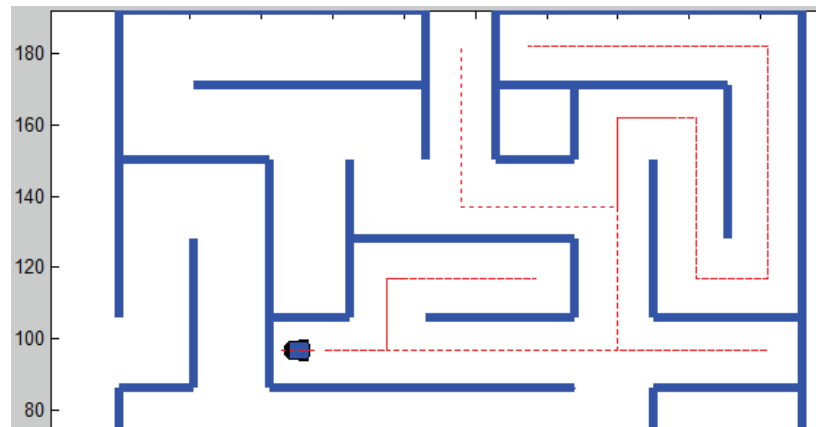


Figura 4.19 Mapeo Sector 1 del laberinto (a) ruta y dibujo del laberinto ideales (b) ruta y dibujo del laberinto con datos reales

## ➤ SECTOR 2

(a)



(b)

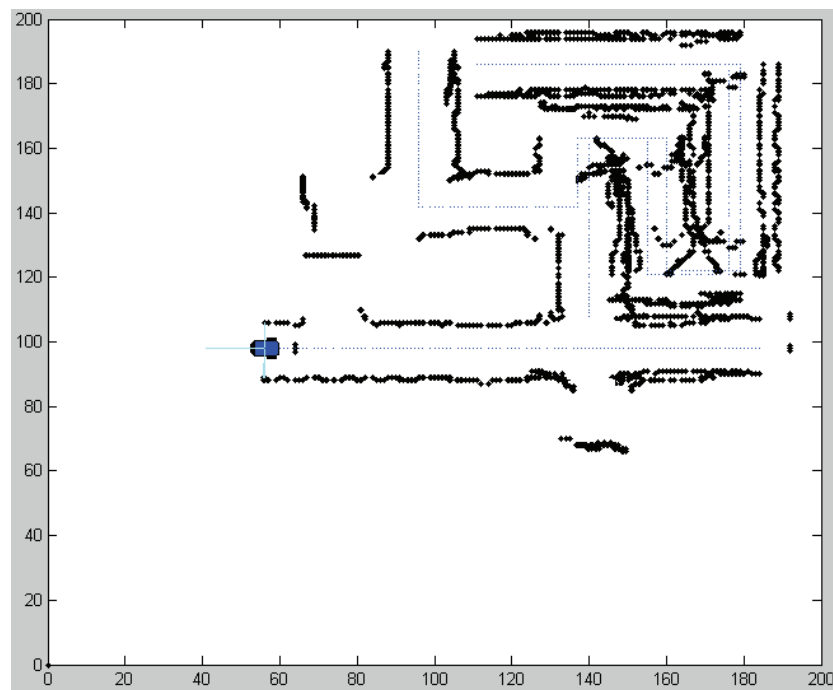


Figura 4.20 Mapeo Sector 2 del laberinto (a) Ruta y dibujo del laberinto ideales (b) Ruta y dibujo del laberinto con datos reales

## ➤ SECTOR 3

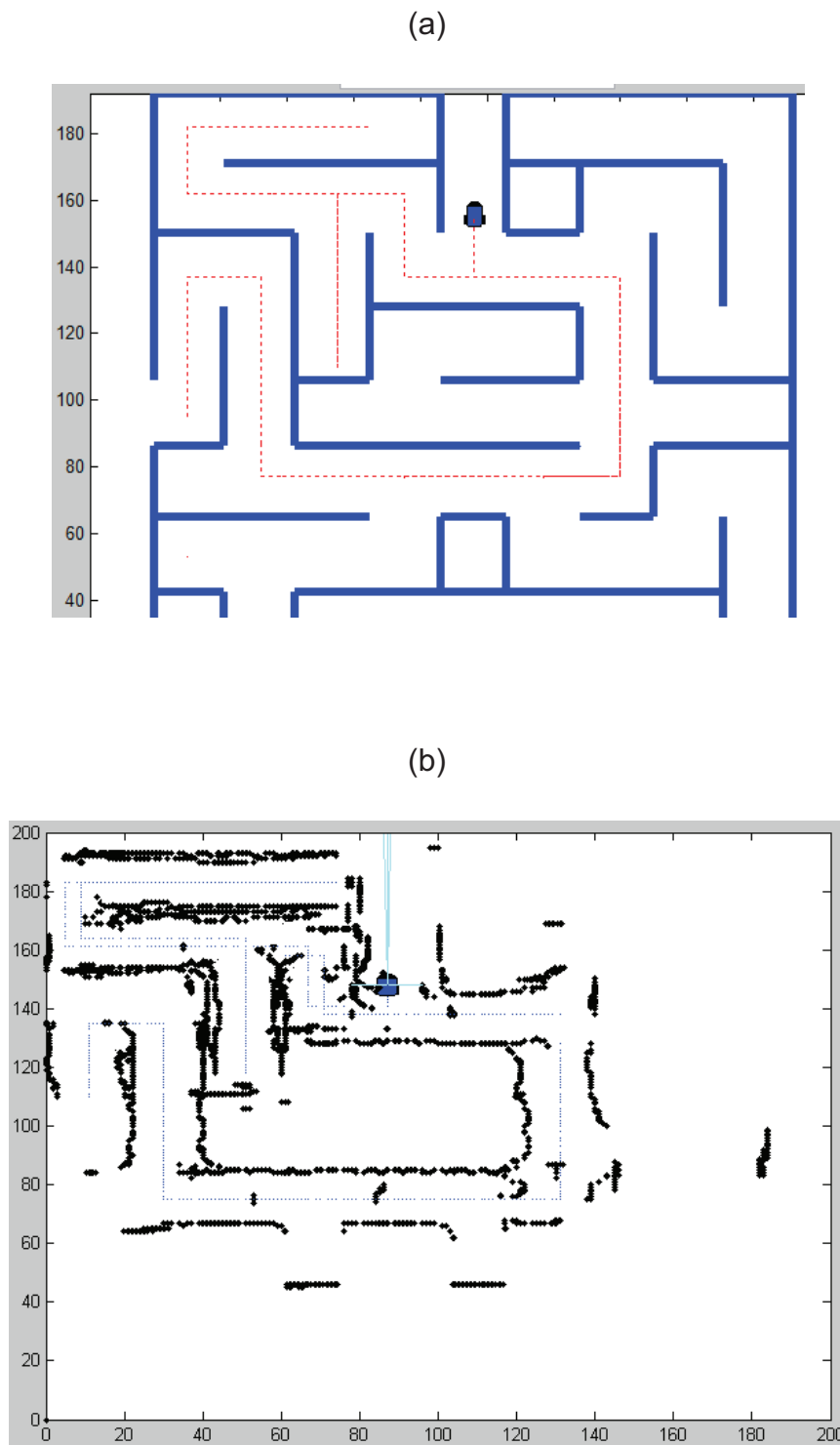


Figura 4.21 Mapeo Sector 3 del laberinto (a) Ruta y dibujo del laberinto ideales (b) Ruta y dibujo del laberinto con datos reales.

De las pruebas realizadas se puede apreciar que los tres minirobots actuando de manera cooperativa exploran el entorno de manera más eficiente, esto quiere decir que actuando coordinadamente se toma el menor tiempo para explorar el laberinto. Cuando la exploración termina, todos los datos de los 3 sectores se procesan con MATLAB para unirlos y formar un mapa aproximado del laberinto mediante la cooperación de los 3 minirobots.

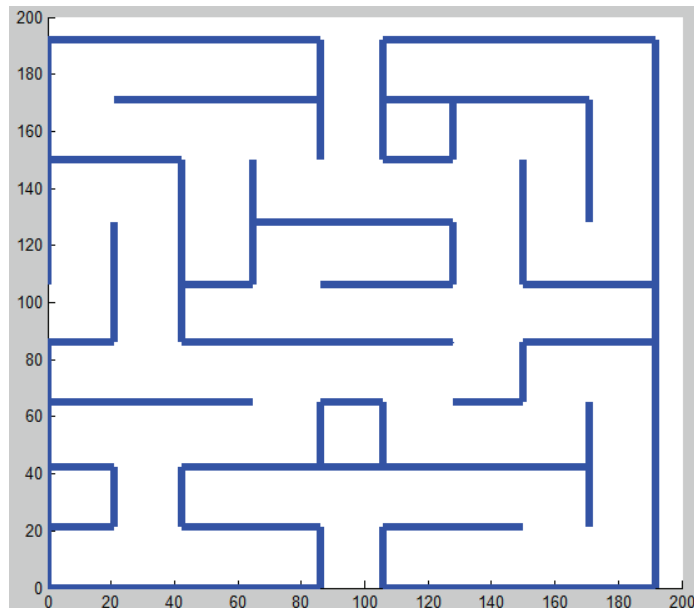


Figura 4.22 Laberinto 2D simulado en MATLAB.

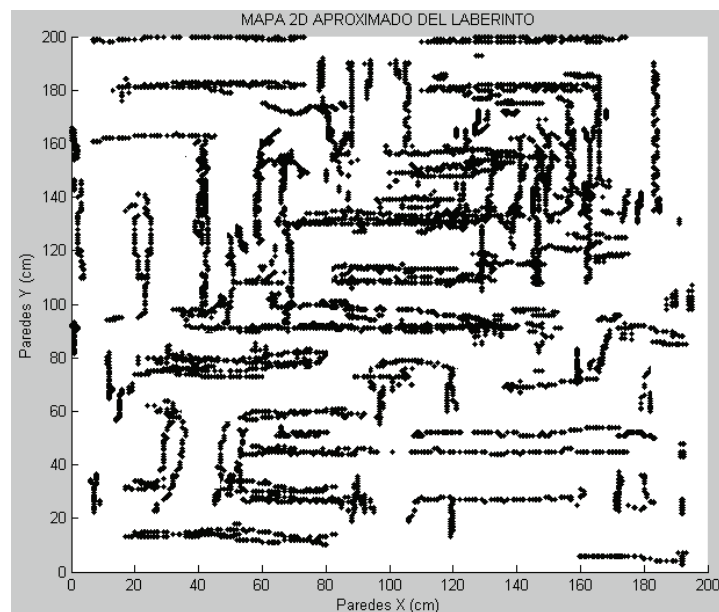


Figura 4.23 Laberinto 2D aproximado después de la exploración cooperativa.



#### 4.4 PRUEBAS AL ALGORITMO PARA LA SOLUCIÓN DEL LABERINTO

Para encontrar la solución del laberinto es necesario que los minirobots tengan una buena locomoción, un adecuado avance dentro de las zonas del laberinto y que exploren con el algoritmo de la mano izquierda de manera eficiente.

Al probar este algoritmo se observó algunas fallas que se presentaron en el avance adecuado dentro de las zonas del laberinto, fallas que a simple vista eran imperceptibles, ya que por la programación y por el algoritmo de la mano izquierda el minirobot parecía que realizaba de buena manera estas acciones.

Estas fallas se hicieron notorias cuando se estableció que el minirobot envíe las letras de las decisiones que toma en el transcurso de su recorrido hacia el MATLAB, como comprobación para tener la certeza de que si estaba realizando bien las acciones de control. Cada decisión tomada por el minirobot es enviada al MATLAB, giro a la izquierda la letra 'I', giro a la derecha 'D', vuelta 'V' e intersección a la derecha 'R', como se explica en el capítulo 3, cuando se explica el algoritmo de la ruta corta.

Con las pruebas realizadas se pudo observar que el minirobot tomaba malas decisiones y que en ciertas zonas como los giros obligados a la derecha o a la izquierda donde no se asignan letras también enviaba letras, lo que representaba que el minirobot detectaba mal la zona donde se encuentra (Figura 3.9)

Para corregir estos problemas se realizaron modificaciones en la programación de avance dentro de las diferentes zonas del laberinto, haciendo cumplir que el minirobot entre y salga de una zona del laberinto, que se explica anteriormente en el numeral 4.1.3. De esta manera se resuelve el laberinto atravesándolo por la ruta más corta que es almacenada dentro de un vector que indica las decisiones correctas, luego de la primera exploración.

Si se desea ver el funcionamiento de la ruta corta, ingrese a esta dirección URL<sup>19</sup>.

---

<sup>19</sup> [http://www.youtube.com/watch?v=TPx-ZqZ\\_eJs&feature=share](http://www.youtube.com/watch?v=TPx-ZqZ_eJs&feature=share)

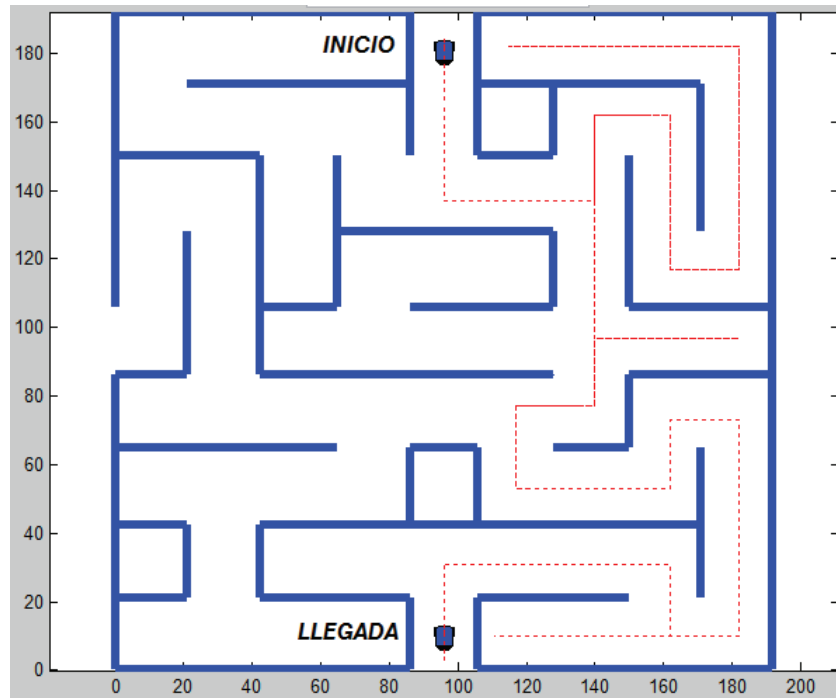


Figura 4.24 Exploración de los caminos en el laberinto

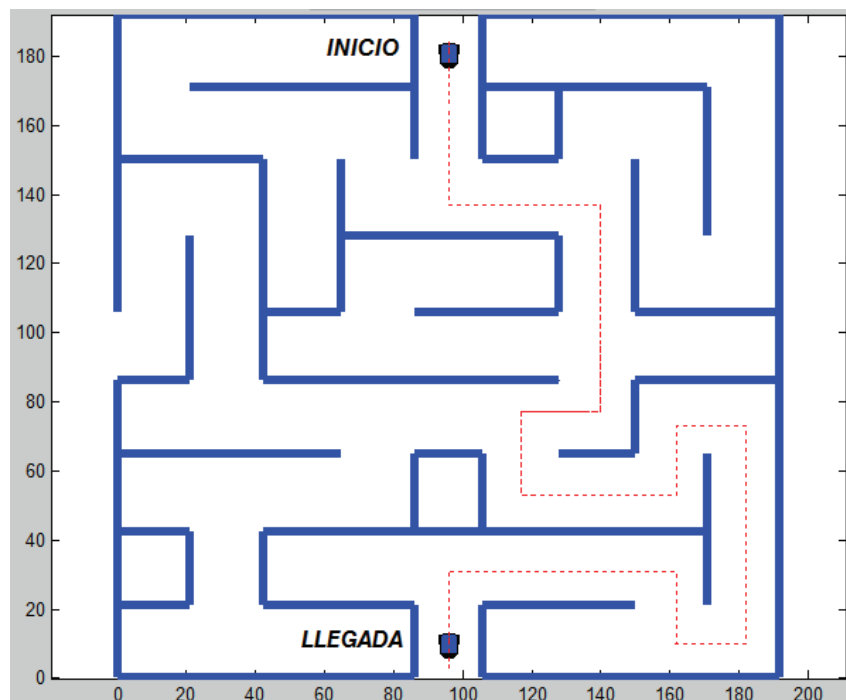


Figura 4.25 Algoritmo de ruta corta después de la primera exploración.

## **CAPITULO 5**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **5.1 CONCLUSIONES**

Se puede concluir que se cumplieron los objetivos planteados en el Proyecto de Titulación, puesto que se consiguió realizar tres mini plataformas robóticas con tracción diferencial que puedan cumplir con ciertas órdenes de exploración de manera eficiente.

Se puede dibujar el entorno explorado usando MATLAB en una PC, para así tener una buena aproximación del entorno por el cual esta desplazándose el minirobot.

El trabajo que puede realizar un solo minirobot es muy bueno, debido a su tamaño o a su velocidad, pero no es eficiente, ya que el tiempo que emplea en la exploración de un entorno es muy largo. Es por esto que la otra parte del proyecto de titulación fue el cooperativismo para lograr una eficacia en la exploración de un entorno y se logró con los 3 minirobots, bajar el tiempo de exploración de un entorno en este caso de un laberinto de  $3.68 \text{ m}^2$ , demostrando así una ventaja del cooperativismo.

Se consiguió realizar la exploración y obtención de la ruta corta de un laberinto, en la cual es indispensable un buen recorrido de los minirobots, es decir que los minirobots durante su desplazamiento vayan por la mitad de los caminos del laberinto, que tengan una velocidad adecuada, que los giros sean precisos. Para lograr esto fue necesario tener una buena estructura física del minirobots, es decir que tenga una medida idónea, que los elementos utilizados como sensores, encoders, motores y la placa arduino sean adecuados y de excelentes algoritmos de control para llevar a cabo todas las tareas.

Un aspecto importante que se trató en el proyecto fue la trayectoria recorrida por los minirobots, la cual se obtuvo utilizando odometría. La odometría conlleva un error en la obtención de la trayectoria por lo que se realizó ajustes, calibraciones

en los minirobots y también correcciones por software, debido a que la acumulación de errores en la orientación causa grandes errores en la posición, los cuales se incrementan proporcionalmente con la distancia recorrida del minirobot. Por este motivo al realizar la exploración con un solo minirobot, de todo el laberinto, se obtiene mayor error en la posición, ya que un minirobot realiza un recorrido bastante extenso, y al realizar la exploración los tres minirobots se obtiene menos error en la posición por el hecho de realizar cada uno un sector del laberinto. Esta es una ventaja adicional del cooperativismo.

Se consiguió implementar la comunicación punto – multipunto para realizar el cooperativismo, de manera que se configuró una red zigbee que proporciona una comunicación entre un único coordinador (PC) y sus routers (minirobots), los datos son enviados hacia el coordinador para que a través del mismo se comunique hacia la PC o se regrese a cualquier de los minirobots para cumplir una orden.

## **5.2 RECOMENDACIONES**

Este proyecto de titulación podrá ser utilizado como un inicio para investigaciones futuras en las que puedan realizar aplicaciones de exploración en otros entornos y realizar tareas de búsqueda.

Para obtener una posición más exacta del minirobot dentro del laberinto se puede utilizar otro método para estimar la posición del minirobot además del sistema odométrico utilizado. Puede ser una estimación de la posición mediante balizas, que permite determinar la posición de un robot móvil en un entorno delimitado por balizas que se encuentran en posiciones conocidas. La posición de los minirobot se determinaría en base al principio de triangulación, a partir de medidas de distancias, ángulos o de la combinación de los dos. La precisión y fiabilidad de este tipo de estimación dependerá del tipo de señal utilizada; infrarrojos, laser, radio, ultrasonidos, de las características del sensor y del número de balizas utilizadas.

Para el manejo y programación de los minirobots, se recomienda primero leer al manual de usuario que está incluido en el anexo 2 de este proyecto de titulación, ya que el uso indebido del arduino duemilanove y de los módulos Xbee S2 puede provocar un fallo en el funcionamiento de los minirobots.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] M. Perezagua, «Conceptos e introduccion sobre robotica,» 27 Enero 2012. [En línea]. Available: [http://twileshare.com/uploads/teoria\\_robotica.pdf](http://twileshare.com/uploads/teoria_robotica.pdf). [Último acceso: 11 Abril 2012].
- [2] Hector, «HectorNavarroBlog,» 10 Marzo 2011. [En línea]. Available: <http://hectornavarroblog.blogspot.com/2011/03/campeones-del-mundo-en-robots.html>. [Último acceso: 30 Mayo 2012].
- [3] I. J. C. Téllez, «MINI - ROBÓTICA COSTRUCCIÓN DE PROTOTIPOS,» [En línea]. Available: <http://www.clubse.com.ar/news/news11/notas/nota10.htm>. [Último acceso: 15 Octubre 2012].
- [4] A. Ollero Baturone, Robótica: manipuladores y robots móviles, Barcelona: MARCOMBO, 2001.
- [5] N. Cardenas y G. Simbana, DISEÑO Y CONSTRUCCION DE UN ROBOT MOVIL DIDACTICO PARA ESTUDIOS EXPERIMENTALES DE ROBOTICA COOPERATIVA, Cuenca: Universidad Politecnica Salesiana, 2007.
- [6] JDLOPE, 9 Febrero 2001. [En línea]. Available: <http://jdlope.tripod.com/hermanos.html>. [Último acceso: 27 Enero 2013].
- [7] G. Javier, «Estimación de la Posición de un Robot Móvil,» [En línea]. Available: [http://www.robotica-urjc.es/doctorado/gonzalez-estimacion\\_de\\_la\\_posicion-96.pdf](http://www.robotica-urjc.es/doctorado/gonzalez-estimacion_de_la_posicion-96.pdf). [Último acceso: Julio 2012].
- [8] P. E. Acuña Carlos, «DISEÑO Y CONSTRUCCION DE UN ROBOT MOVIL QUE PUEDA DESPLAZARSE DENTRO DE UN LABERINTO,» Escuela Politecnica Nacional, Quito, Agosto 2006.
- [9] L. d. S. Rodrigo, *Seminario de Diseño y Construcción de Microrobots*.
- [10] R. Faludi, Building Wireless Sensor Network, Primera Edición ed., O'Reilly,

2011.

- [11] Carla, «ITGS,» 20 Mayo 2009. [En línea]. Available: <http://itgscsolano.blogspot.com/>. [Último acceso: Septiembre 2012].
- [12] RFWIRELESS, 2011. [En línea]. Available: [http://www.rfwireless-world.com/Tutorials/Zigbee\\_tutorial.html](http://www.rfwireless-world.com/Tutorials/Zigbee_tutorial.html). [Último acceso: Octubre 2012].
- [13] J. Logares, Abril 2011. [En línea]. Available: <http://www.javierlongares.com/arte-en-8-bits/introduccion-a-zigbee-y-las-redes-de-sensores-inalambricas/>. [Último acceso: Enero 2013].
- [14] Arduino, «Arduino Duemilanove,» 2010. [En línea]. Available: <http://arduino.cc/es/Main/ArduinoBoardDuemilanove>. [Último acceso: Marzo 2012].
- [15] Gravitech, «Electronic Expermiental Solution,» 2006. [En línea]. Available: <http://www.gravitech.us/arxbsh.html>. [Último acceso: Noviembre 2012].
- [16] BricoGeek, «BricoGeek,» 2005. [En línea]. Available: <http://www.bricogeek.com/shop/43-xbee-2mw-serie-25-con-antena.html>. [Último acceso: Noviembre 2012].
- [17] Cytron Technologies, «HC - SR04 User's\_Manual,» Noviembre 2012. [En línea]. Available: [https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL\\_pfa39RsB-x2qR4vP8saG73rE/edit?pli=1](https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL_pfa39RsB-x2qR4vP8saG73rE/edit?pli=1).
- [18] Pololu Corporation, «Pololu Robotics & Electronics,» [En línea]. Available: <http://www.pololu.com/catalog/product/2368/specs>. [Último acceso: 10 Febrero 2013].
- [19] Pololu Corporation, «Pololu Robotics & Electronics,» [En línea]. Available: <http://www.pololu.com/catalog/product/1217%20target=>. [Último acceso: 10 Febrero 2013].
- [20] HobbyKing, «HobbyKing,» 2012. [En línea]. Available:

[http://www.hobbyking.com/hobbyking/store/\\_\\_9494\\_\\_Turnigy\\_1300mAh\\_4S\\_20C\\_Lipo\\_Pack.html](http://www.hobbyking.com/hobbyking/store/__9494__Turnigy_1300mAh_4S_20C_Lipo_Pack.html). [Último acceso: Enero 2013].

- [21] Grupo de Investigación en Tecnologías de la Información y las Comunicaciones, «MÉTODO BÁSICO PARA IMPLEMENTAR UN CONTROLADOR DIGITAL PID EN UN MICROCONTROLADOR PIC PARA DESARROLLO DE APLICACIONES A BAJO COSTO,» [En línea]. Available: [http://www.edutecne.utn.edu.ar/microcontrol\\_congr/industria/MTODOB~1.PDF](http://www.edutecne.utn.edu.ar/microcontrol_congr/industria/MTODOB~1.PDF). [Último acceso: 9 Noviembre 2012].
- [22] Arduino, «Arduino Playground,» [En línea]. Available: <http://playground.arduino.cc/Main/RotaryEncoders>. [Último acceso: 10 Enero 2013].
- [23] Cuentos Cuánticos, «Robótica: Estimación de posición por odometría,» 15 Diciembre 2011. [En línea]. Available: <http://cuentos-cuanticos.com/2011/12/15/robotica-estimacion-de-posicion-por-odometria/>. [Último acceso: 12 Diciembre 2012].
- [24] Pololu Corporation, «Robot Pololu 3pi Guía de usuario,» [En línea]. Available: <http://www.pololu.com/file/0J137/Pololu3piRobotGuiaUsuario.pdf>. [Último acceso: 10 Febrero 2013].
- [25] A. Rosales, *APUNTES DE CONTROL COMPUTARIZADO*, Quito, 2012.
- [26] G. Diego y Z. Cristian, «PROGRAMACION E IMPLEMENTACION DE UN PANTOGRAFO CNC PARA OXICORTE MEDIANTE EL USO DE MATLAB BASADO EN LA IMPORTACION DE UN ARCHIVO EN FORMATO GRAFICO,» Escuela Politecnica Nacional, Quito, 2012.
- [27] P. H. Analia, «Universidad Nacional de San Juan,» [En línea]. Available: <http://dea.unsj.edu.ar/control1/apuntes/resumen%20controladores.pdf>. [Último acceso: Octubre 2012].



## ANEXO 1

### GUIDE (GRAPHICAL USER INTERFACE)

Una interfaz GUI contiene menús, botones, cuadros de texto, y otros más que se explican en los siguientes párrafos.

Cuando se crea una GUI se obtienen los siguientes archivos:

**Archivo Fig:** es el archivo que contiene la descripción de todos los componentes de la interfaz.

**Archivo .m:** es el archivo donde se hace la programación para el control del GUI con los llamados a las funciones o callbacks. Para crear una GUI en MATLAB basta con escribir la palabra guide en el Comand Windows o mediante el icono en la pantalla principal:

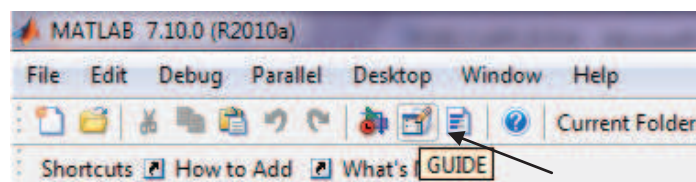


Figura A1.1 Icono para iniciar un GUIDE

Después saldrá el cuadro de diálogo que se muestra en la Figura A1.2:

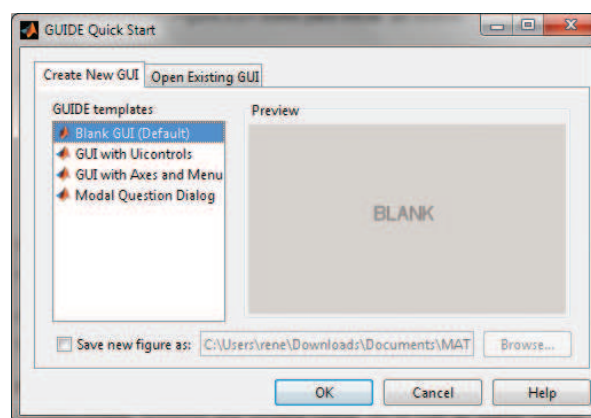


Figura A1.2 Cuadro de diálogo para crear una GUIDE

El cuadro de diálogo muestra dos opciones, una para crear una nueva GUIDE y otra para abrir una GUIDE existente.

Al elegir la opción de crear una nueva GUIDE, se tiene que elegir de entre las siguientes cuatro opciones:

**Blank GUI (Default):** es la opción de GUIDE en blanco que viene predeterminada, presenta una ventana nueva, en el cual se puede diseñar una nueva interfaz.

**GUI with Uicontrols:** esta opción presenta un ejemplo en el cual se calcula la masa, dada la densidad y el volumen, en alguno de los dos sistemas de unidades.

**GUI with Axes and Menu:** esta opción es otro ejemplo el cual contiene las opciones Open, Print y Close.

También tiene un pop-up menu, un push button y un objeto axes, puede ejecutarse el programa eligiendo alguna de las seis opciones que se encuentran en el menú despegable.

**Modal Question Dialog:** se muestra en la pantalla un cuadro de diálogo común, el cual consta de una pequeña imagen, una etiqueta y dos botones Yes y No, dependiendo del botón que se presione, el GUIDE retorna el texto seleccionado. En una nueva GUIDE se tiene la siguiente pantalla:

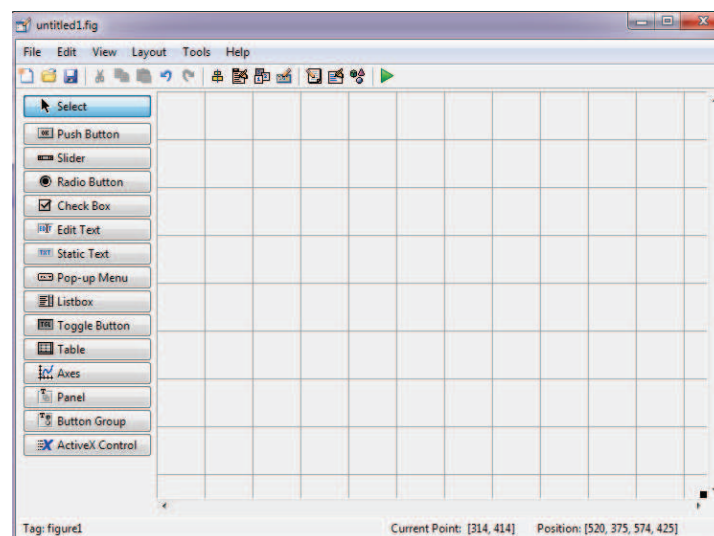


Figura A1.3 Pantalla principal de una nueva GUIDE

En la Figura A1.3 se observa los principales botones que son utilizados en la creación de una nueva GUIDE, así como las barras de herramientas. Cada botón del GUIDE cumple con una función específica que se muestra a continuación:

**Select:** con la ayuda del ratón de la PC se puede mover por cualquier lado de la interfaz y seleccionar los elementos que se quieren mover o borrar.

**Push Button:** genera una acción cuando se presiona con la ayuda del click del ratón sobre ellos, y la rutina programa en el archivo .m se ejecuta.

**Slider:** presenta una lista de opciones de las cuales podemos elegir solo una.

**Radio button:** presenta varias opciones de las cuales se puede elegir una sola.

**Check box:** indica el estado de una opción, se lo hace con un visto.

**Edit text:** es una caja donde se edita el texto.

**Static text:** es una caja donde se muestra un string de texto.

**Pop – up menú:** da una lista de opciones para elegir.

**List box:** muestra una lista deslizable.

**Toggle button:** muestra dos estados que pueden ser 'on' u 'off'

**List box:** muestra un listado de palabras.

**Axes:** permite mostrar gráficos en general.

**Button group:** permite exclusividad de selección con los radio button.

Cada uno de los botones enumerados antes tiene su editor de propiedades, que se lo puedo obtener al hacer doble click izquierdo sobre cualquiera de los botones.

En el presente proyecto de titulación se utilizan los botones: push button, axes, static text, edit text y button group.

## ANEXO 2

### MANUAL DE OPERACIÓN DE LOS MINIROBOTS

Para un correcto funcionamiento de los minirobots se debe tener en cuenta el siguiente manual de operación, que esta dividido en dos partes, la primera se trata acerca de la programación tanto del arduino duemilanove como de las antenas Xbee S2 y la segunda acerca del manejo y puesta en marcha de los minirobots y de la interfaz en MATLAB.

#### PROGRAMACIÓN DE ARDUINO DUEMILANOVE Y DE ANTENAS XBEE S2

##### 1. ARDUINO DUEMILANOVE

Para programar al arduino duemilanove se debe tener el programa ARDUINO instalado en la PC, se puede descargar de la página: <http://arduino.cc/es/Main/Software>.

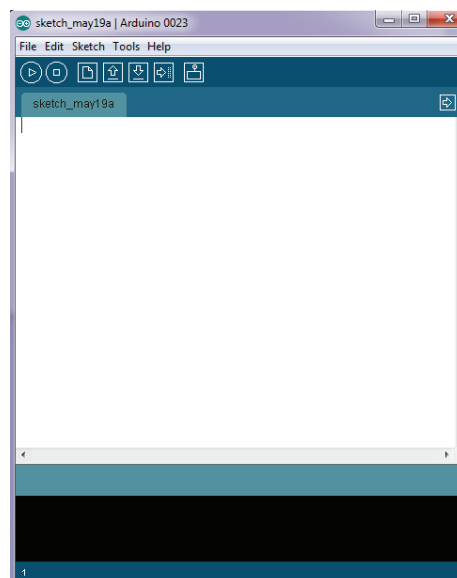


Figura A2.1 Software Arduino

Una vez con el programa se debe abrir un nuevo programa con extensión .pde

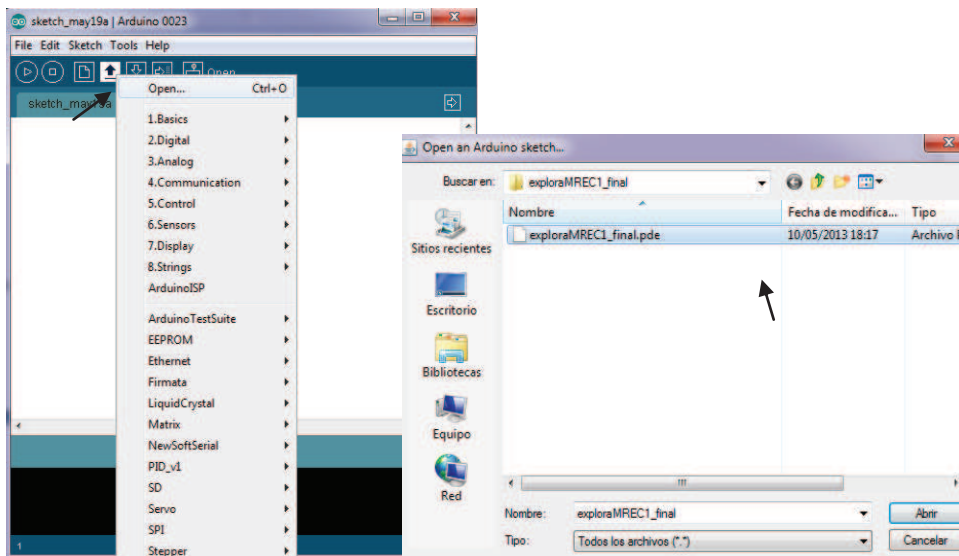


Figura A2.2 Ventana para abrir un programa

Cuando el programa está abierto, se debe verificar que no existen errores en la programación, se debe dar un click en el botón de VERIFY.

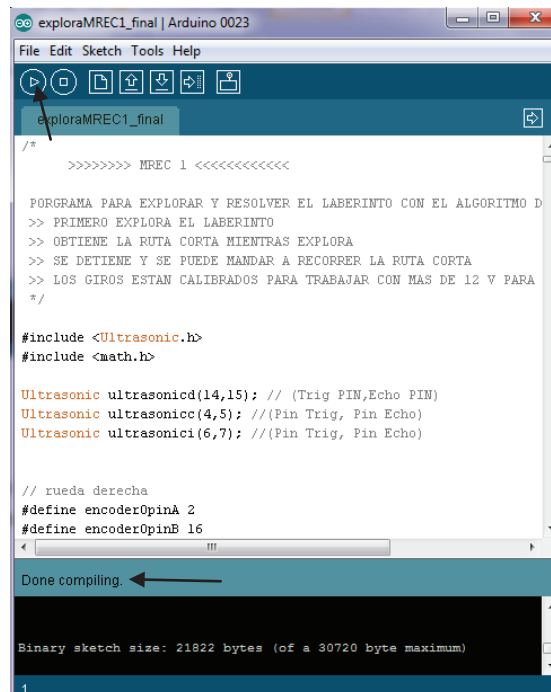


Figura A2.3 Ventana de verificación de errores



## 2. ANTENAS XBEE S2

Para la programación de las antenas xbee debe tener el software X –CTU en la PC o se lo puede descargar de la página.

<http://www.digi.com/support/productdetail?pid=3352&osvid=57&type=utilities>, una vez en el programa la pantalla principal es:

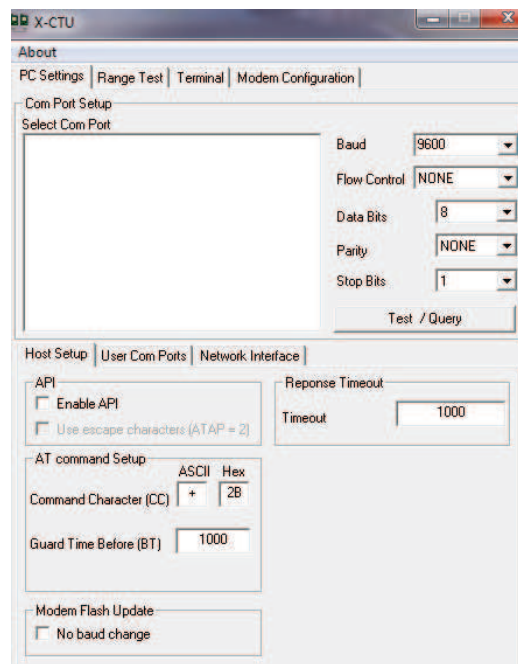


Figura A2.6 Software X – CTU

Para la programación de las antenas xbee con el arduino, los jumpers del modulo de comunicación deben cambiar de posición de XBEE a USB, además en el arduino se debe escribir las líneas de programación mostradas en la Figura A2.7:

```
void setup () {
}
void loop() {
}
```

Figura A2.7 Programación en arduino para programar las antenas xbee



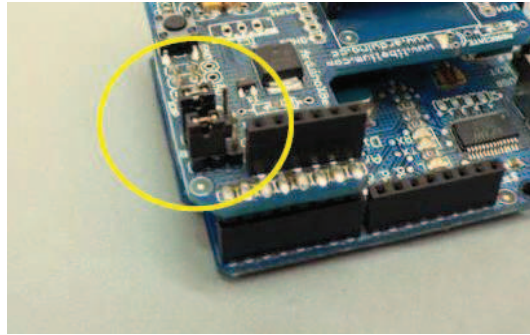


Figura A2.8 Jumpers cambiados de XBEE a USB

Cuando se hayan completado los pasos descritos anteriormente, se debe conectar el USB al puerto del arduino y con el programa X – CTU abierto, se debe dar un click en TEST/QUERY para verificar que la antena se conectó correctamente.

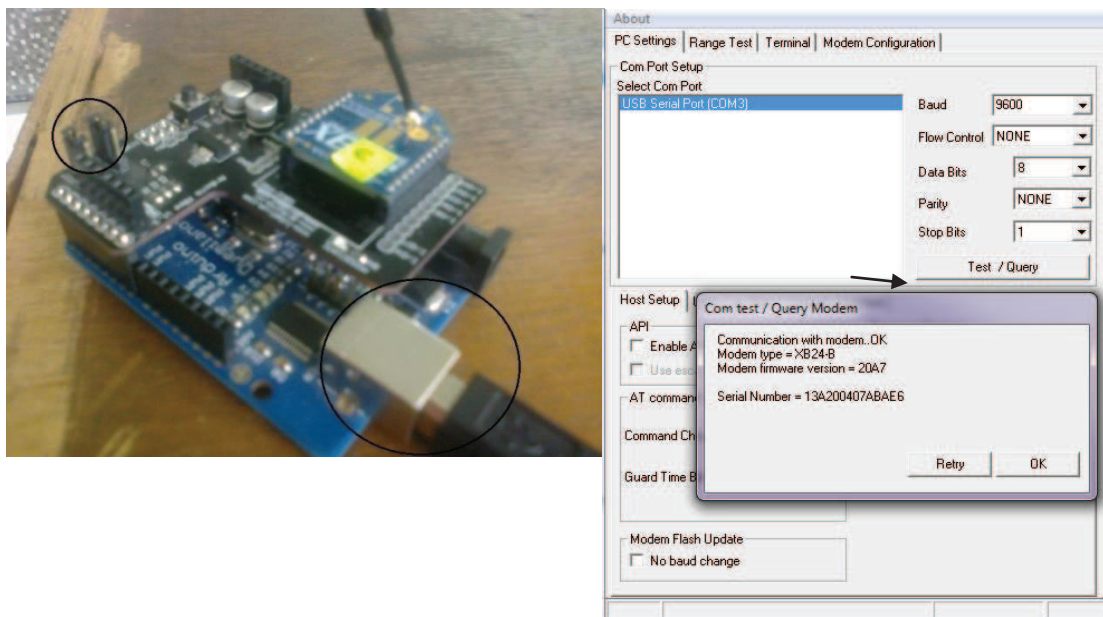


Figura A2.9 Ventana TEST/QUERY

En la pestaña MODEM CONFIGURATION se debe elegir las características de las antenas, tales como: tipo de antena (modem), la función que va a cumplir la antena (coordinador, router o end device) y la versión de firmware.



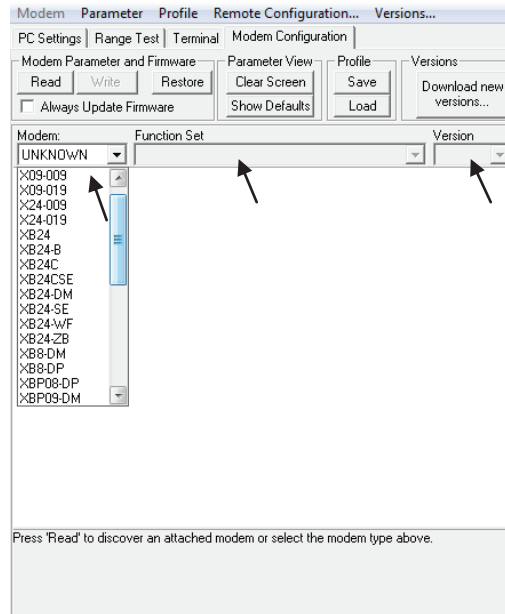


Figura A2.10 Pestaña MODEM CONFIGURATION

Después de elegir toda la configuración, se debe dar un click en WRITE para que la programación esté completa. Como sugerencia se puede dar un click en ALWAYS UPDATE FIRMWARE antes de quemar el programa para mantener el firmware actualizado.

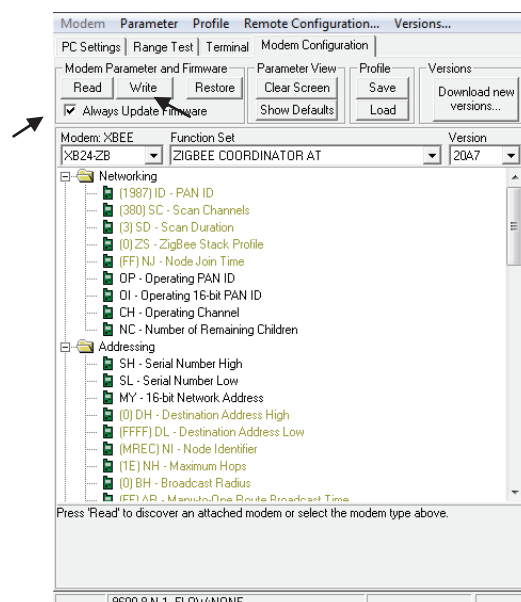
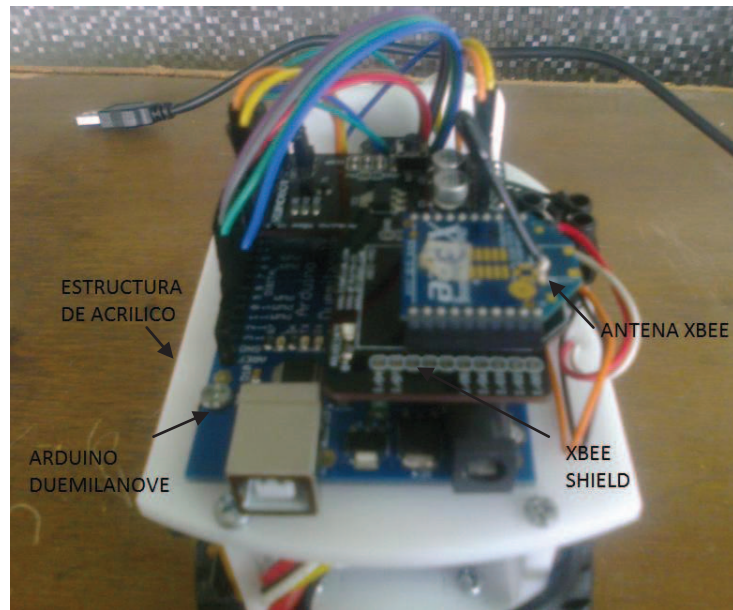


Figura A2.11 Ventana final de programación de las antenas xbee

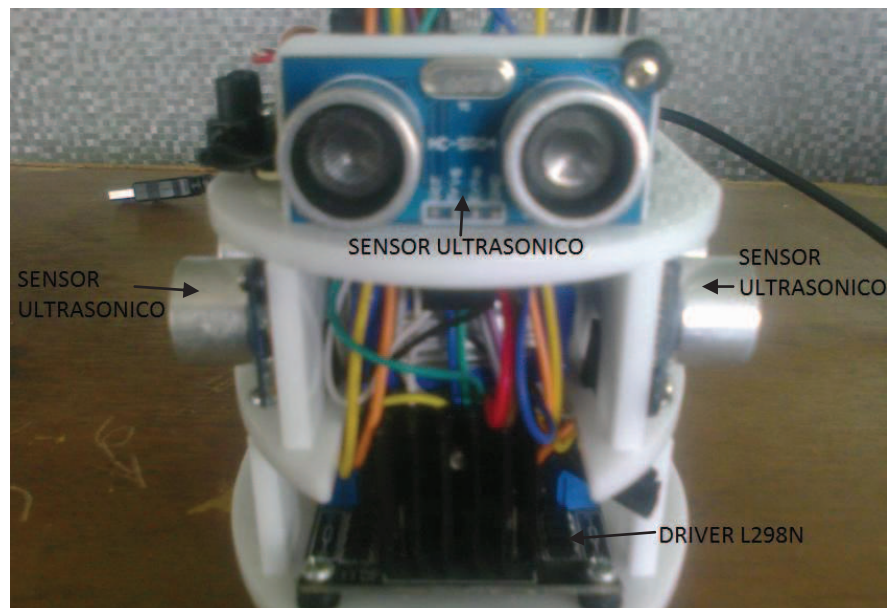
## MANEJO Y PUESTA EN MARCHA DE LOS MINIROBOTS

El diseño final de los minirobots es el mismo, los 3 tienen la misma estructura y la misma disposición de los elementos. A continuación se muestra en la Figura A2.12 el diseño final del minirobot y sus partes constitutivas.

(a)



(b)



(c)

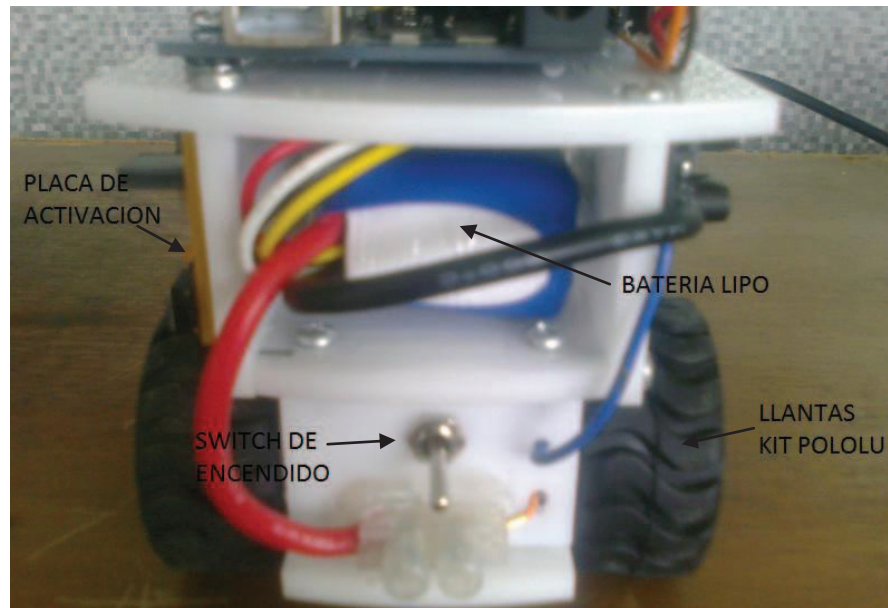


Figura A2.12 Partes constitutivas del minirobot en (a), (b) y (c)

Para que el minirobot se encienda, se debe levantar la palanca del switch de encendido hacia arriba, esperar unos 5 segundos hasta que la antena xbee entre en funcionamiento, después de este tiempo se debe aplastar el pulsador mostrado en la Figura A3.13:

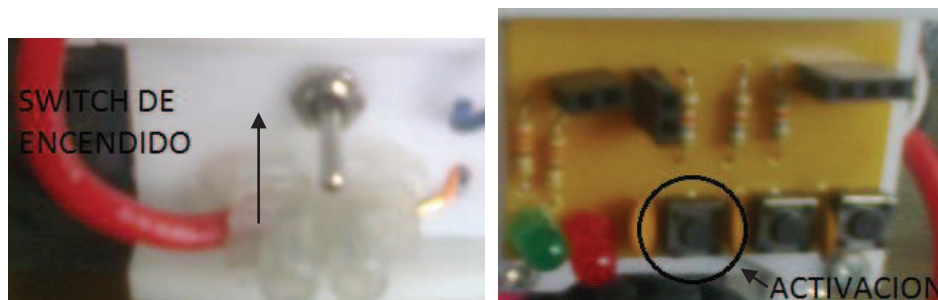


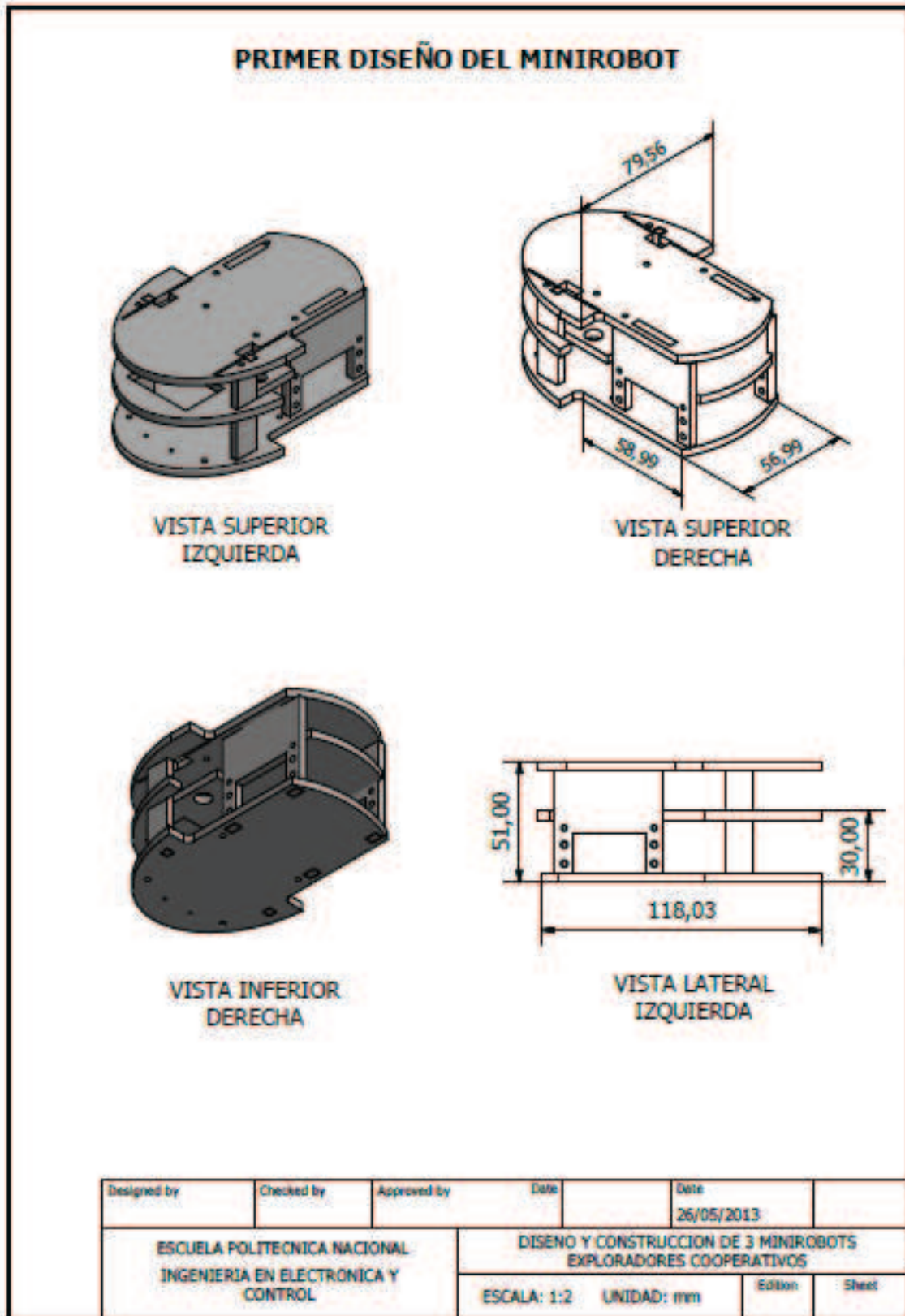
Figura A2.13 Encendido y puesta en marcha.

Si el funcionamiento que se desea con los minirobots es con la comunicación hacia la PC, se debe instalar el MATLAB y leer los archivos .m y .fig del CD de los programas del proyecto de titulación.

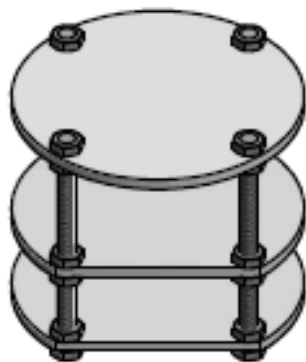
Las baterías utilizadas en los minirobots son del tipo LIPO (litio – polímero), son formadas por celdas, en el caso del proyecto de titulación de 3 celdas, cada una de un valor de 3.7 voltios, son muy comunes en radiocontrol pero cuando el valor de la celda cae por debajo de los 3 voltios, cada celda entra en un proceso de cristalización y la batería no podrá ser recargada, es por eso que la recomendación es que el valor mínimo en el cual la batería de cada minirobot debe operar es de 9 voltios porque cada batería tiene un valor de 11.1 voltios, si la batería está llegando a este límite mínimo se debe recargar enseguida.

## ANEXO 3

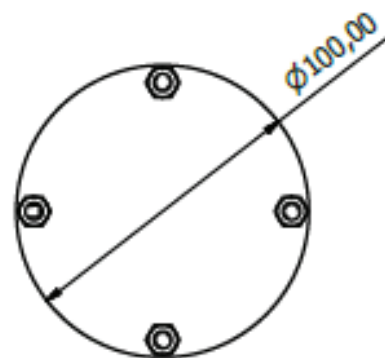
## PLANOS DEL DISEÑO DE LAS ESTRUCTURAS



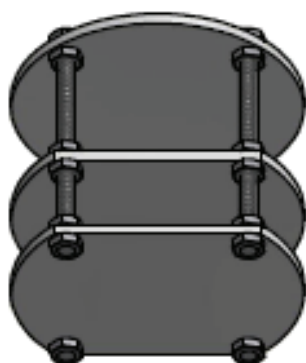
## SEGUNDO DISEÑO DEL MINIROBOT



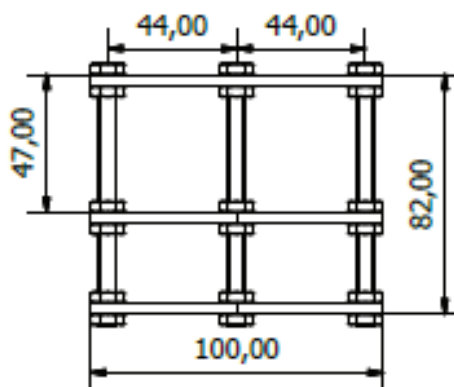
VISTA SUPERIOR  
IZQUIERDA



VISTA SUPERIOR



VISTA INFERIOR  
IZQUIERDA

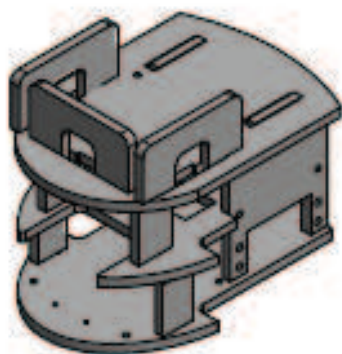


VISTA LATERAL  
DERECHA

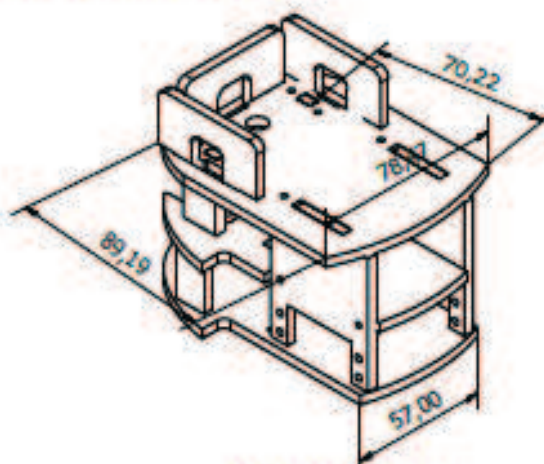
Designed by	Checked by	Approved by	Date	Date	
				26/05/2013	
ESCUELA POLITECNICA NACIONAL INGENIERIA EN ELECTRONICA Y CONTROL			DISEÑO Y CONSTRUCCION DE 3 MINIROBOTS EXPLORADORES COOPERATIVOS		
ESCALA: 1:2		UNIDAD: mm		Edición	Sheet



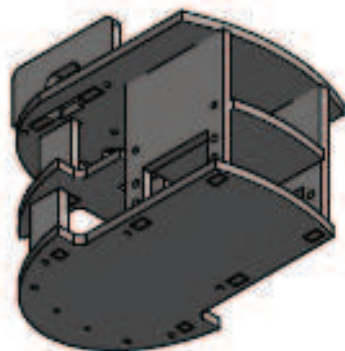
### TERCER DISEÑO DEL MINIROBOT



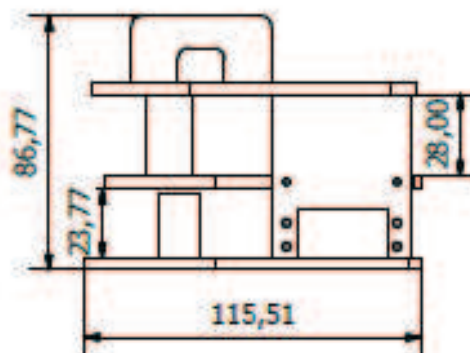
VISTA SUPERIOR  
IZQUIERDA



VISTA SUPERIOR  
DERECHA



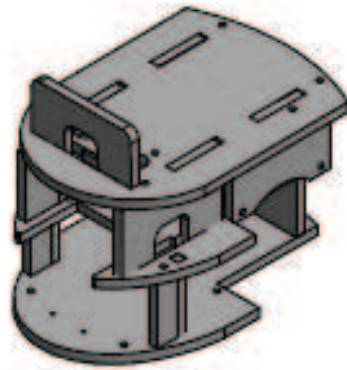
VISTA INFERIOR  
DERECHA



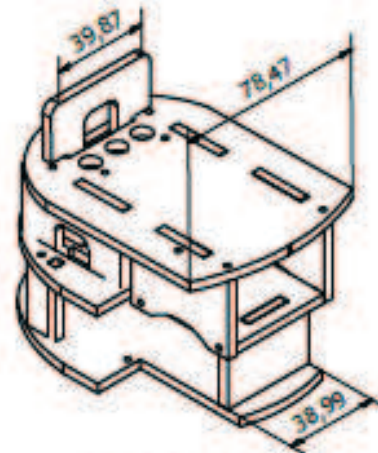
VISTA LATERAL  
IZQUIERDA

Designed by	Checked by	Approved by	Date	Date					
				27/05/2013					
ESCUELA POLITECNICA NACIONAL INGENIERIA EN ELECTRONICA Y CONTROL			DISEÑO Y CONSTRUCCION DE 3 MINIROBOTS EXPLORADORES COOPERATIVOS						
			ESCALA: 1:2	UNIDAD: mm	<table border="1"> <tr> <td>Edicion</td> <td>Sheet</td> </tr> <tr> <td></td> <td></td> </tr> </table>	Edicion	Sheet		
Edicion	Sheet								

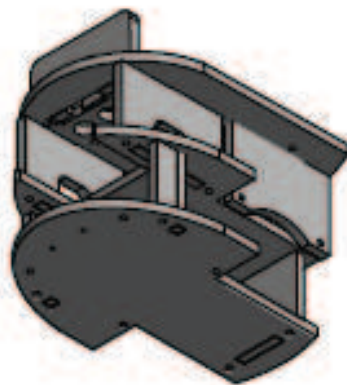
## DISEÑO FINAL DEL MINIROBOT



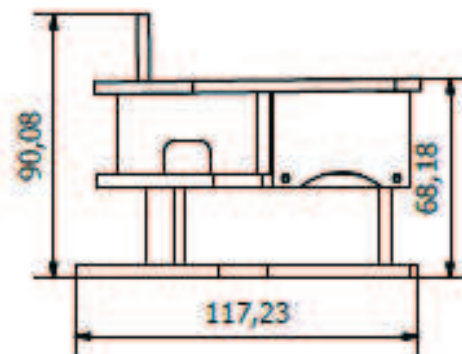
VISTA SUPERIOR  
IZQUIERDA



VISTA SUPERIOR  
DERECHA



VISTA INFERIOR  
IZQUIERDA



VISTA LATERAL  
IZQUIERDA

Designed by	Checked by	Approved by	Date	Date	
				26/05/2013	
ESCUELA POLITECNICA NACIONAL INGENIERIA EN ELECTRONICA Y CONTROL			DISEÑO Y CONSTRUCCION DE 3 MINIROBOTS EXPLORADORES COOPERATIVOS		
			ESCALA: 1:2	UNIDAD: mm	Edición Sheet



## ANEXO 4

## HOJAS DE DATOS DE LOS COMPONENTES DE LOS MINIROBOTS



Tech Support: [services@electfreaks.com](mailto:services@electfreaks.com)

### Ultrasonic Ranging Module HC - SR04

#### Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The module includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
  - (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
  - (3) IF the signal back, through high level, time of high output IO duration is the time from sending ultrasonic to returning.
- Test distance = (high level time \* velocity of sound (340M/S)) / 2,

#### Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

#### Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
Measuring Angle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL level signal and the range in proportion
Dimension	45*20*15mm

Figura A4.1 Hoja de datos sensor ultrasónico (primera parte)



### Timing diagram

The Timing diagram is shown below. You only need to supply a short 10 $\mu$ s pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion. You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula:  $\mu\text{s} / 58 = \text{centimeters}$  or  $\mu\text{s} / 148 = \text{inch}$ ; or: the range = high level time \* velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.

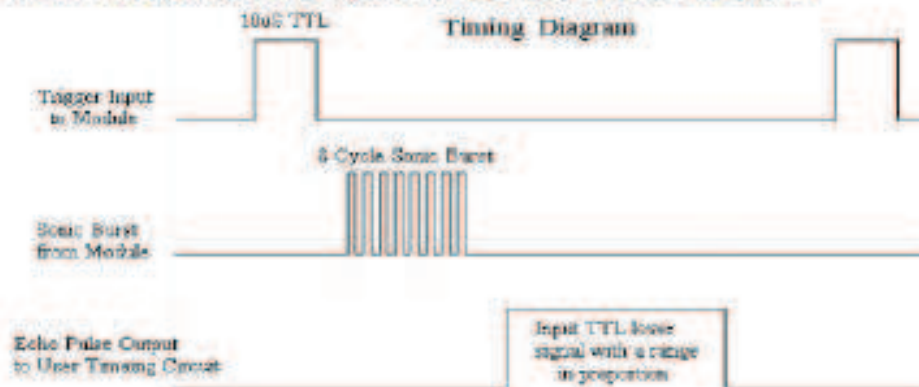


Figura A4.2 Hoja de datos sensor ultrasónico (segunda parte)



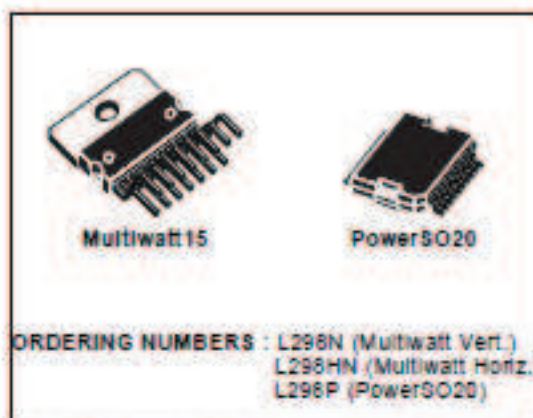
L298

## DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 48 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

## DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

## BLOCK DIAGRAM

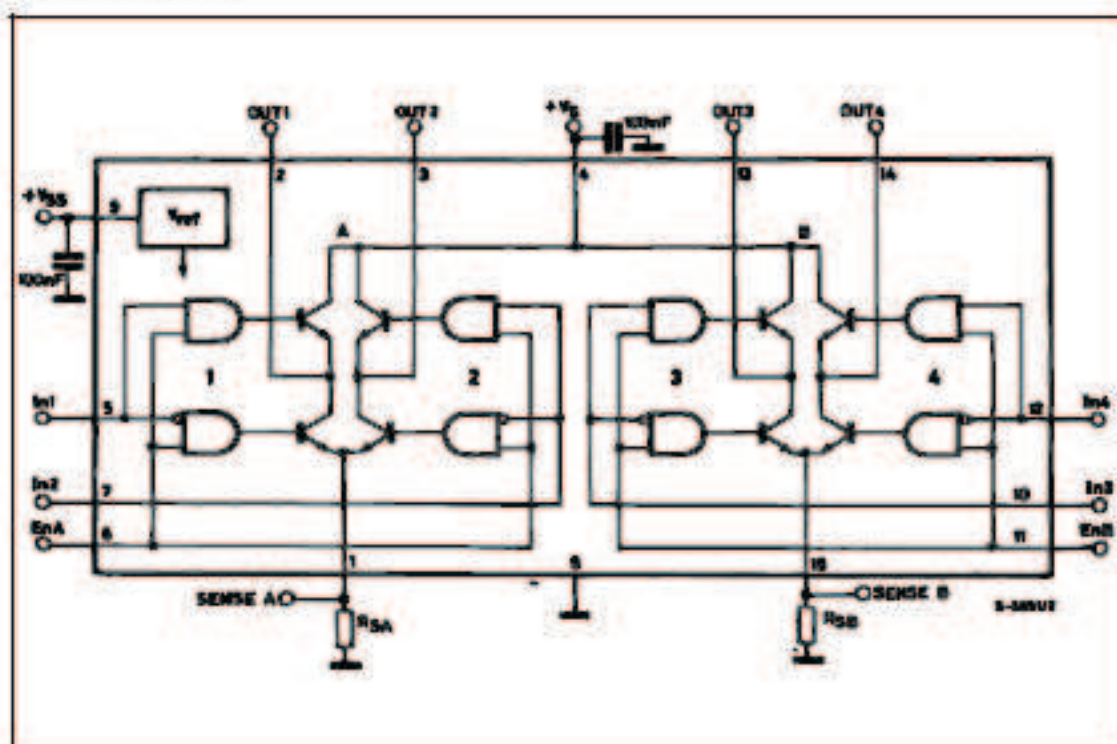


Figura A4.3 Hoja de datos L298N (primera parte)

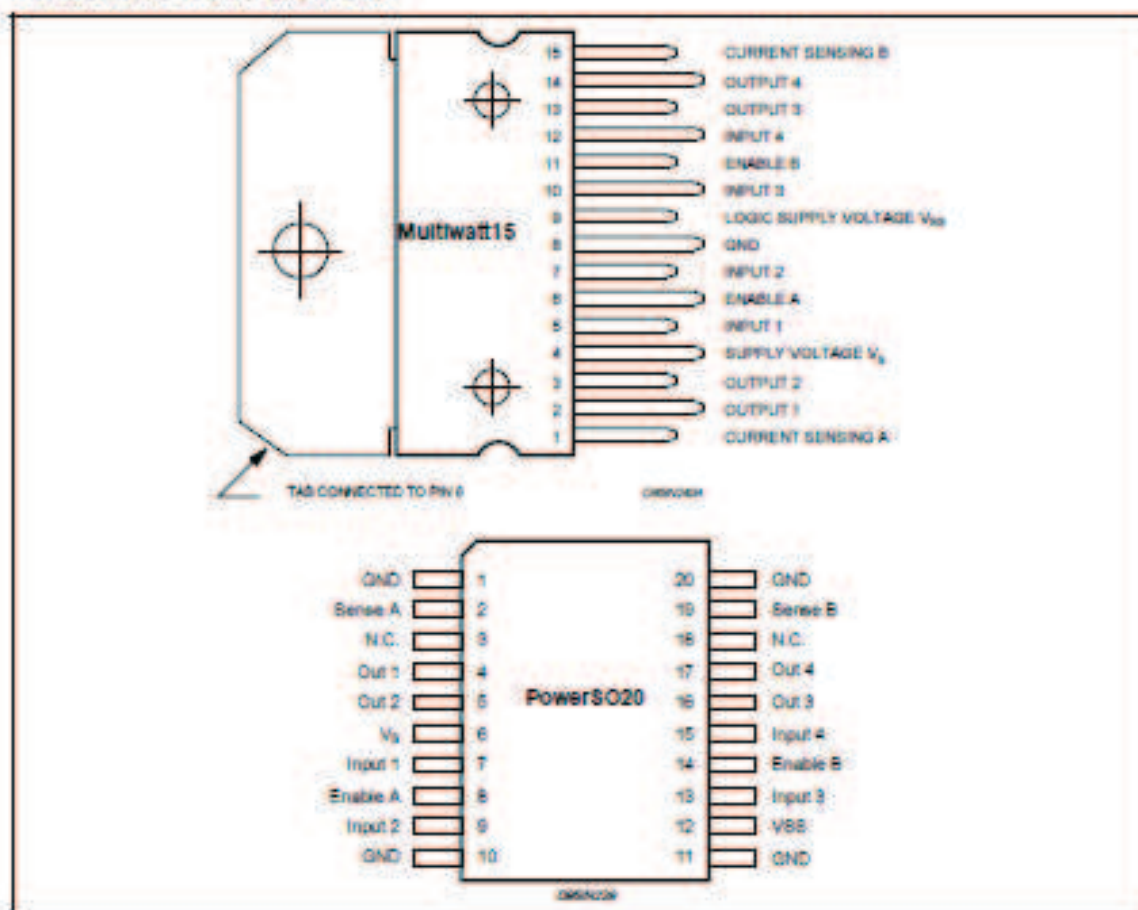


## L298

## ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
$V_S$	Power Supply	50	V
$V_{SS}$	Logic Supply Voltage	7	V
$V_i, V_{en}$	Input and Enable Voltage	-0.3 to 7	V
$I_o$	Peak Output Current (each Channel)		
	- Non Repetitive ( $t = 100\mu s$ )	3	A
	- Repetitive (80% on -20% off; $t_{on} = 10ms$ )	2.5	A
	-DC Operation	2	A
$V_{sense}$	Sensing Voltage	-1 to 2.3	V
$P_{tot}$	Total Power Dissipation ( $T_{case} = 75^\circ C$ )	25	W
$T_{op}$	Junction Operating Temperature	-25 to 130	$^\circ C$
$T_{stg}, T_j$	Storage and Junction Temperature	-40 to 150	$^\circ C$

## PIN CONNECTIONS (top view)



## THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th(j-c)}$	Thermal Resistance Junction-case	Max.	-	3	$^\circ C/W$
$R_{th(j-a)}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	$^\circ C/W$

(\*) Mounted on aluminum substrate

Figura A4.4 Hoja de datos L298N (segunda parte)

## PIN FUNCTIONS (refer to the block diagram)

MW. 15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	$V_{DS}$	Supply Voltage for the Power Output Stages. A non-Inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	$V_{SS}$	Supply Voltage for the Logic Blocks. A 100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
-	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS ( $V_{DS} = 42V$ ;  $V_{SS} = 5V$ ,  $T_J = 25^\circ C$ ; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$V_{DS}$	Supply Voltage (pin 4)	Operative Condition	$V_{IH} + 2.5$		45	V
$V_{SS}$	Logic Supply Voltage (pin 9)		4.5	5	7	V
$I_S$	Quiescent Supply Current (pin 4)	$V_{en} = H$ ; $I_L = 0$		13	22	mA
		$V_I = L$		50	70	mA
		$V_I = H$			4	mA
		$V_{en} = L$				mA
$I_{SS}$	Quiescent Current from $V_{SS}$ (pin 9)	$V_{en} = H$ ; $I_L = 0$		24	36	mA
		$V_I = L$		7	12	mA
		$V_I = H$			6	mA
		$V_{en} = L$				mA
$V_{IL}$	Input Low Voltage (pins 5, 7, 10, 12)		-0.3		1.5	V
$V_{IH}$	Input High Voltage (pins 5, 7, 10, 12)		2.3		$V_{SS}$	V
$I_{L1}$	Low Voltage Input Current (pins 5, 7, 10, 12)	$V_I = L$			-10	$\mu A$
$I_{H1}$	High Voltage Input Current (pins 5, 7, 10, 12)	$V_I = H$ & $V_{SS} - 0.6V$		30	100	$\mu A$
$V_{en} = L$	Enable Low Voltage (pins 6, 11)		-0.3		1.5	V
$V_{en} = H$	Enable High Voltage (pins 6, 11)		2.3		$V_{SS}$	V
$I_{en} = L$	Low Voltage Enable Current (pins 6, 11)	$V_{en} = L$			-10	$\mu A$
$I_{en} = H$	High Voltage Enable Current (pins 6, 11)	$V_{en} = H$ & $V_{SS} - 0.6V$		30	100	$\mu A$
$V_{CEsat (H)}$	Source Saturation Voltage	$I_L = 1A$ $I_L = 2A$	0.95	1.35 2	1.7 2.7	V
$V_{CEsat (S)}$	Sink Saturation Voltage	$I_L = 1A$ (5) $I_L = 2A$ (5)	0.85	1.2 1.7	1.6 2.3	V
$V_{CEsat}$	Total Drop	$I_L = 1A$ (5) $I_L = 2A$ (5)	1.80		3.2 4.9	V
$V_{sense}$	Sensing Voltage (pins 1, 15)		-1 (1)		2	V

Figura A4.5 Hoja de datos L298N (tercera parte)



## L298

## ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$T_1$ (V)	Source Current Turn-off Delay	0.5 V <sub>i</sub> to 0.9 I <sub>L</sub> (2); (4)		1.5		μs
$T_2$ (V)	Source Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (2); (4)		0.2		μs
$T_3$ (V)	Source Current Turn-on Delay	0.5 V <sub>i</sub> to 0.1 I <sub>L</sub> (2); (4)		2		μs
$T_4$ (V)	Source Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (2); (4)		0.7		μs
$T_5$ (V)	Sink Current Turn-off Delay	0.5 V <sub>i</sub> to 0.9 I <sub>L</sub> (3); (4)		0.7		μs
$T_6$ (V)	Sink Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (3); (4)		0.25		μs
$T_7$ (V)	Sink Current Turn-on Delay	0.5 V <sub>i</sub> to 0.9 I <sub>L</sub> (3); (4)		1.5		μs
$T_8$ (V)	Sink Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (3); (4)		0.2		μs
f <sub>c</sub> (V)	Commutation Frequency	I <sub>L</sub> = 2A		25	40	KHz
$T_1$ (V <sub>sat</sub> )	Source Current Turn-off Delay	0.5 V <sub>sat</sub> to 0.9 I <sub>L</sub> (2); (4)		3		μs
$T_2$ (V <sub>sat</sub> )	Source Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (2); (4)		1		μs
$T_3$ (V <sub>sat</sub> )	Source Current Turn-on Delay	0.5 V <sub>sat</sub> to 0.1 I <sub>L</sub> (2); (4)		0.3		μs
$T_4$ (V <sub>sat</sub> )	Source Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (2); (4)		0.4		μs
$T_5$ (V <sub>sat</sub> )	Sink Current Turn-off Delay	0.5 V <sub>sat</sub> to 0.9 I <sub>L</sub> (3); (4)		2.2		μs
$T_6$ (V <sub>sat</sub> )	Sink Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (3); (4)		0.35		μs
$T_7$ (V <sub>sat</sub> )	Sink Current Turn-on Delay	0.5 V <sub>sat</sub> to 0.9 I <sub>L</sub> (3); (4)		0.25		μs
$T_8$ (V <sub>sat</sub> )	Sink Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (3); (4)		0.1		μs

1) Sensing voltage can be -1 V for t < 50 μsec; in steady state V<sub>sat</sub> min > -0.5 V.

2) See fig. 2.

3) See fig. 4.

4) The load must be a pure resistor.

Figura A4.6 Hoja de datos L298N (cuarta parte)

## Features

- High Performance, Low Power AVR<sup>®</sup> 8-Bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 32K Bytes of In-System Self-Programmable Flash program memory
  - 1K Bytes EEPROM
  - 2K Bytes Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Optional Boot Code Section with Independent Lock Bits
    - + In-System Programming by On-chip Boot Program
    - + True Read-While-Write Operation
  - Programming Lock for Software Security
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package
    - + Temperature Measurement
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Byte-oriented 2-wire Serial Interface (Philips I<sup>2</sup>C compatible)
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 23 Programmable I/O Lines
  - 32-lead TQFP, and 32-pad QFN/MLF
- Operating Voltage:
  - 2.7V - 5.5V for ATmega328P
- Temperature Range:
  - Automotive Temperature Range: -40°C to +125°C
- Speed Grade:
  - 0 - 8 MHz @ 2.7 - 5.5V (Automotive Temp. Range: -40°C to +125°C)
  - 0 - 16 MHz @ 4.5 - 5.5V (Automotive Temp. Range: -40°C to +125°C)
- Low Power Consumption
  - Active Mode: 1.5mA @ 3V - 4MHz
  - Power-down Mode: 1 µA @ 3V



8-bit **AVR<sup>®</sup>**  
Microcontroller  
with 32K Bytes  
In-System  
Programmable  
Flash

ATmega328P

Automotive

Preliminary

7810A-AVR-11/06



Figura A4.7 Hoja de datos ATMEGA 328P





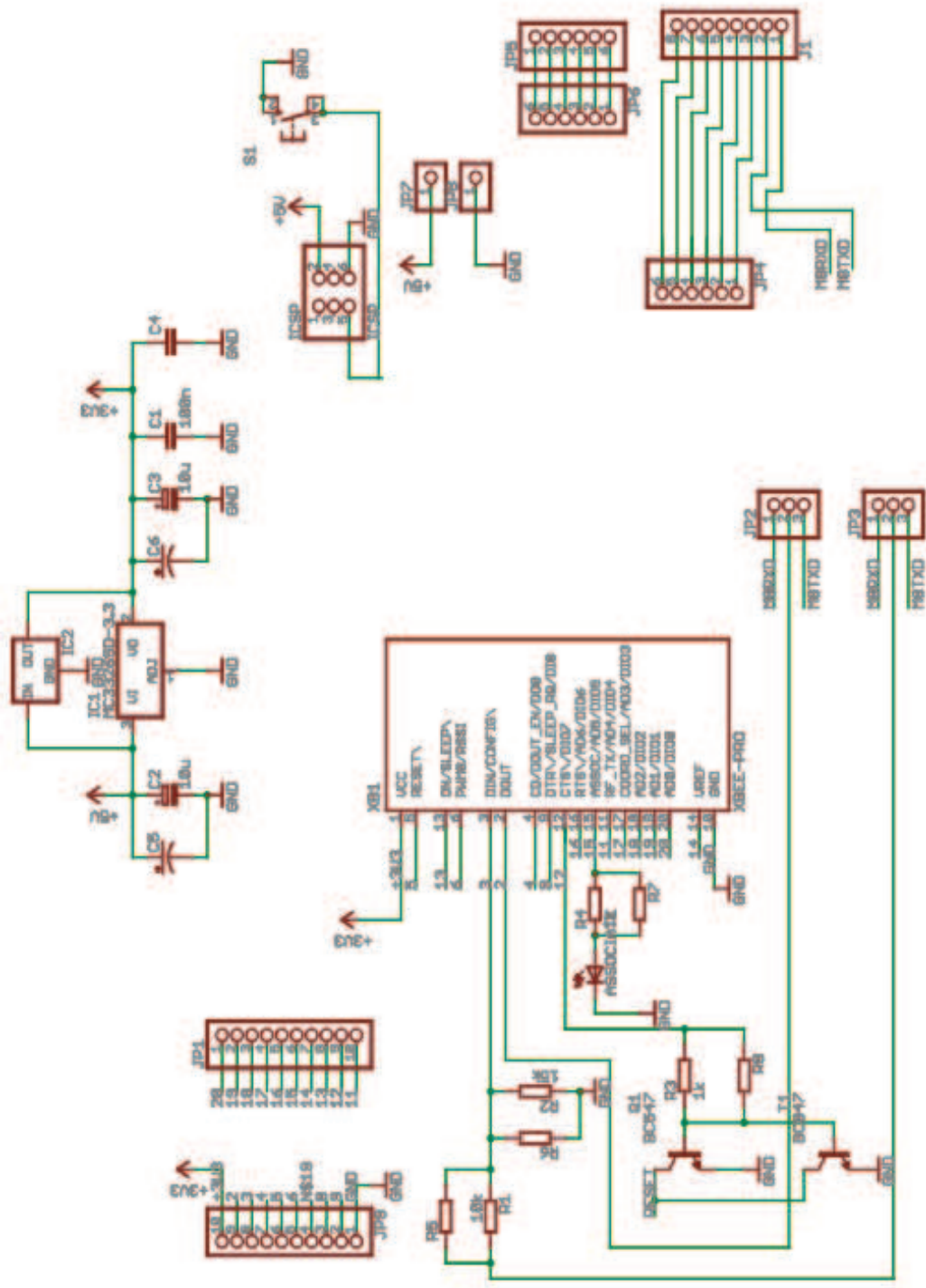


Figura A4.9 Diagrama circuital del Xbee Shield (módulo para la comunicación)

## 1. XBee Series 2 OEM RF Modules

The XBee Series 2 OEM RF Modules were engineered to operate within the ZigBee protocol and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between remote devices.

The modules operate within the ISM 2.4 GHz frequency band.



### 1.1. Key Features

#### High Performance, Low Cost

- Indoor/Urban: up to 133' (40 m)
- Outdoor line-of-sight: up to 400' (120 m)
- Transmit Power: 2 mW (+3 dBm)
- Receiver Sensitivity: -95 dBm

RF Data Rate: 250,000 bps

#### Advanced Networking & Security

Retries and Acknowledgements  
 DSSS (Direct Sequence Spread Spectrum)  
 Each direct sequence channel has over 65,000 unique network addresses available  
 Point-to-point, point-to-multipoint and peer-to-peer topologies supported  
 Self-routing, self-healing and fault-tolerant mesh networking

#### Low Power

XBee Series 2

- TX Current: 40 mA (@3.3 V)
- RX Current: 40 mA (@3.3 V)
- Power-down Current: < 1  $\mu$ A @ 25°C

#### Easy-to-Use

No configuration necessary for out-of box RF communications

AT and API Command Modes for configuring module parameters

Small form factor

Extensive command set

Free X-CTU Software  
 (Testing and configuration software)

**Free & Unlimited Technical Support**

Figura A4.10 Hoja de datos de la antena Xbee S2 (primera parte)

Specification	XBee Series 2
<b>Performance</b>	
Indoor/Urban Range	up to 133 ft. (40 m)
Outdoor RF line-of-sight Range	up to 400 ft. (120 m)
Transmit Power Output (software selectable)	2mW (+3dBm)
RF Data Rate	250,000 bps
Serial Interface Data Rate (software selectable)	1200 - 230400 bps (non-standard baud rates also supported)
Receiver Sensitivity	-95 dBm (1% packet error rate)
<b>Power Requirements</b>	
Supply Voltage	2.8 – 3.4 V
Operating Current (Transmit)	40mA (@ 3.3 V)
Operating Current (Receive)	40mA (@ 3.3 V)
Power-down Current	< 1 uA @ 25°C
<b>General</b>	
Operating Frequency Band	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)
Operating Temperature	-40 to 85° C (Industrial)
Antenna Options	Integrated Whip, Chip, RPSMA, or U.FL Connector
<b>Networking &amp; Security</b>	
Supported Network Topologies	Point-to-point, Point-to-multipoint, Peer-to-peer & Mesh
Number of Channels (software selectable)	16 Direct Sequence Channels
Addressing Options	PAN ID and Addresses, Cluster IDs and Endpoints (optional)
<b>Agency Approvals</b>	
United States (FCC Part 15.247)	Pending
Industry Canada (IC)	Pending
Europe (CE)	Pending

Figura A4.11 Hoja de datos de la antena Xbee S2 (segunda parte)

Figure 1-03. XBee Series 2 RF Module Pin Number  
(top sides shown - shields on bottom)

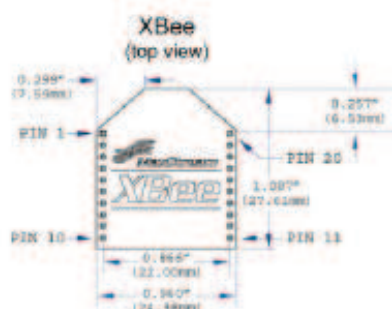


Table 1-02. Pin Assignments for the XBee Series 2 Modules  
(Low-asserted signals are distinguished with a horizontal line above signal name.)

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / CONFIG	Input	UART Data In
4	DIO8	Either	Digital I/O 8
5	RESET	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI / DIO10	Output	PWM Output 0 / RX Signal Strength Indicator / Digital IO
7	PWM / DIO11	Either	Digital I/O 11
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	DIO4	Either	Digital I/O 4
12	CTS / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / SLEEP	Output	Module Status Indicator
14	[reserved]	-	Do not connect
15	Associate / DIO5	Either	Associated Indicator, Digital I/O 5
16	RTS / DIO6	Either	Request-to-Send Flow Control, Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

#### Design Notes:

- Minimum connections: VCC, GND, DOUT & DIN
- Minimum connections to support firmware upgrades: VCC, GND, DIN, DOUT, RTS & DTR
- Signal Direction is specified with respect to the module
- Module includes a 30k Ohm resistor attached to RESET
- Several of the input pull-ups can be configured using the PR command
- Unused pins should be left disconnected

Figura A4.12 Hoja de datos de la antena Xbee S2 (tercera parte)



## ANEXO 5

### COSTOS DEL PROYECTO DE TITULACIÓN

En el proyecto de titulación se utilizaron varios componentes que fueron traídos del exterior debido a su alto coste y también porque no se encontraban en el país.

Los precios que se colocan con (\*) son los precios sin el coste de importación, lo cual se aumentará al final de la tabla A5.1.

Tabla A5.1 Precios de los componentes del proyecto de titulación

CANTIDAD	COMPONENTE	PRECIO UNITARIO	TOTAL
4	Arduino Duemilanove ATMEGA328P	\$ 25,00	\$ 100,00*
15	Sensores Ultrasónicos HC – SR04	\$ 3,50	\$ 52,50*
4	Módulos XBEE SHIELD	\$ 12,00	\$ 48,00*
4	Antenas XBEE Series 2	\$ 30,00	\$ 120,00*
3	Driver L298N manejo de motores DC	\$ 12,00	\$ 36,00*
6	Micromotores DC 150:1	\$ 19,50	\$ 117,00
3	Kits encoders POLOLU	\$ 49,50	\$ 148,50
3	Baterías LIPO 11.1 v/1,3 Ah	\$ 9,50	\$ 28,50*
1	Cargador de baterías LIPO	\$ 7,80	\$ 7,80*
---	Cables de conexión macho – hembra	---	\$ 10,00
---	Cables de conexión hembra – hembra	---	\$ 10,00
---	Acrílico para las estructuras de los minirobots	---	\$ 80,00
---	Corte de acrílicos	---	\$ 50,00
---	Resistencias, capacitores, pulsadores, baquelitas, leds, conectores macho y hembra.	---	\$ 10,00

Sumando las cantidades con el asterisco se obtiene: \$ 392,80

A este precio se debe aumentar el 5% de impuesto a la salida de capitales, lo cual es:

$$\$ 392,80 * 5\% = \$ 19,64$$

Sumando todos los precios da un total de:

$$\$ 392,80 + \$ 19,64 + \$ 425,50 = \$ \mathbf{837,94}$$

Por lo que el precio de cada minirobot está estimado en:

$$\$ \frac{837,94}{3} = \$ \mathbf{279,31}$$

## ANEXO 6

### CONSTRUCCIÓN DE MINIROBOTS PREVIOS AL MINIROBOT DEFINITIVO

#### CONSTRUCCIÓN DEL MINI ROBOT MREC – 01

El minirobot consta de tres bases hechas en acrílico de 4mm de espesor, donde se realizaron perforaciones en distintos lugares para la sujeción mediante pernos y tuercas de los elementos que consta el mini robot, así como también para la sujeción de las bases mediante un tornillo sinfín de ¼ de pulgada.

En la base inferior van ubicados y sujetos las llantas, el driver L298D para los motores, los motores y la rueda de libre movimiento.

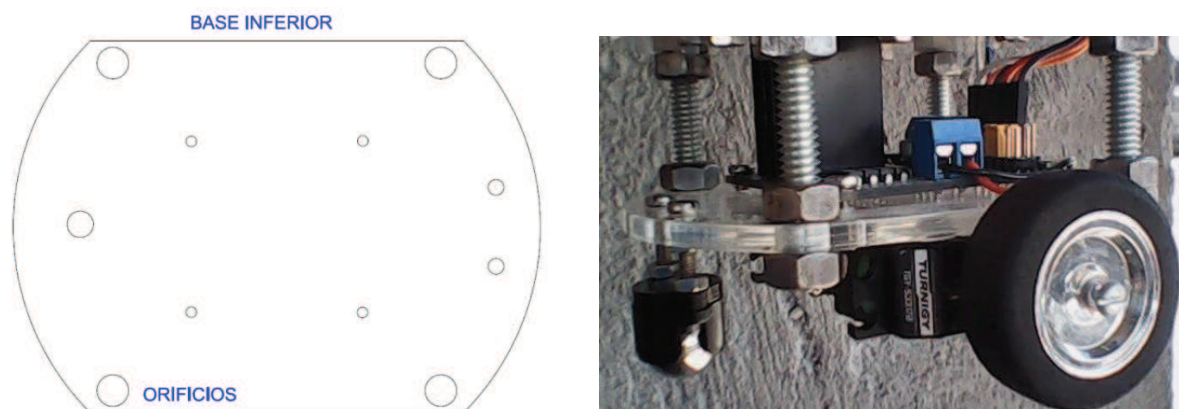


Figura A6.1 Izquierda) Base inferior dibujada en AUTOCAD Derecha) Base inferior con los elementos

En la base intermedia se ubican los tres sensores ultrasónicos con sus respectivas placas de conexión y sujeción, además se encuentra el espacio en el medio de la base donde va ubicada la batería de LiPo.

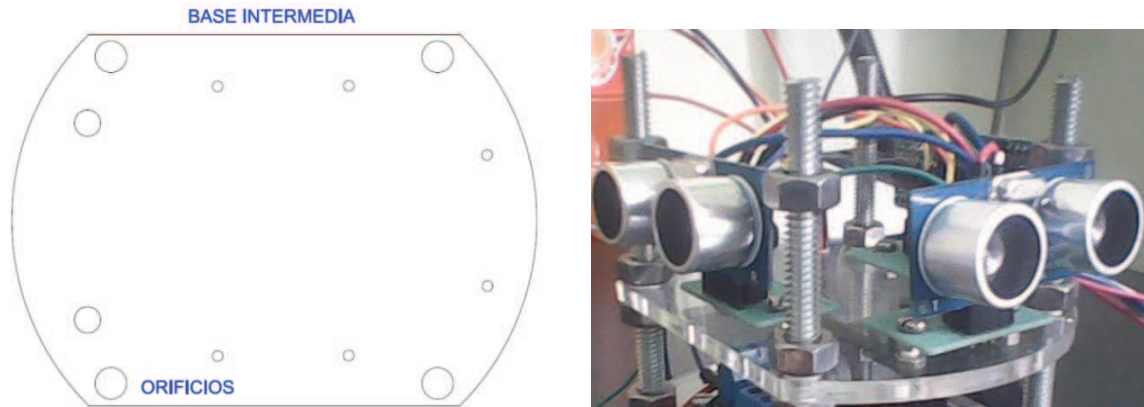


Figura A6.2 Izquierda) Base intermedia dibujada en AUTOCAD Derecha) Base intermedia con los elementos

En la base superior está ubicado el arduino duemilanove, sujetado a una madera de balsa para no tener problema con las tuercas ubicadas a los lados de la base, arriba del arduino está ubicado el módulo Arduino Xbee Shield, así como también, la antena Xbee Series 2.

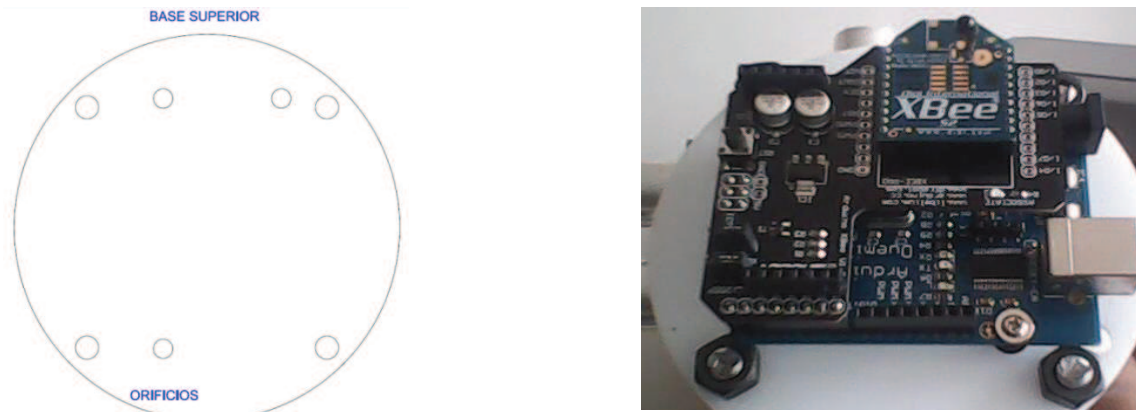


Figura A6.3 Izquierda) Base superior dibujada en AUTOCAD Derecha) Base superior con los elementos

Todas las bases se unen mediante 4 tornillos sinfin sujetos con tuercas.



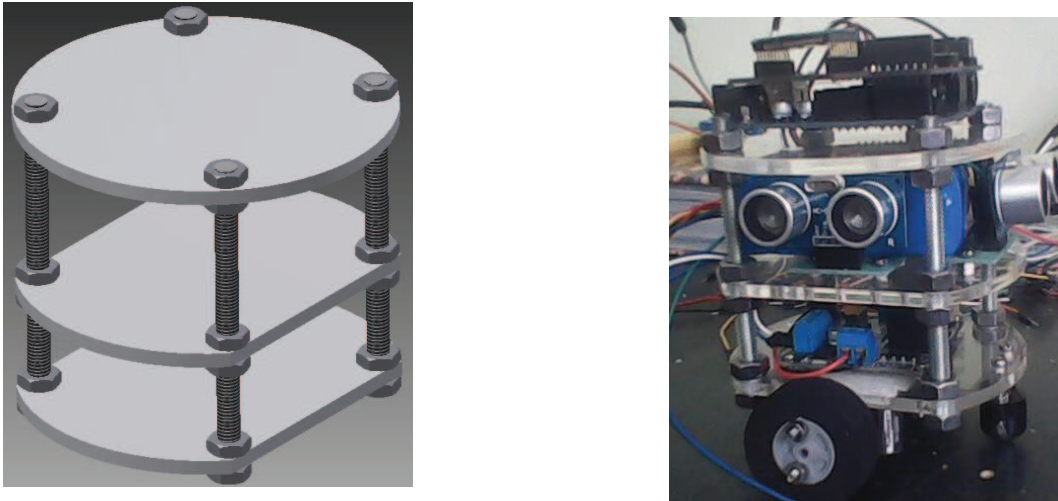


Figura A6.4 Izquierda) Bases ensambladas en INVENTOR Derecha) Bases ensambladas con los elementos

## CONSTRUCCIÓN DEL MINIROBOT MREC – 02

Se elaboró tres bases, la base inferior es capaz de sostener a los motores DC, el driver de los motores y la rueda de libre giro. Esta base tiene varias perforaciones, que posteriormente será donde se coloquen los soportes para el driver L298D para los motores y además donde se empernarán las paredes y soportes para el resto de la estructura.

Para realizar el ensamblado previamente se machueló todas las bases, los soportes y las paredes, se utilizó pernos M3 para la cohesión.

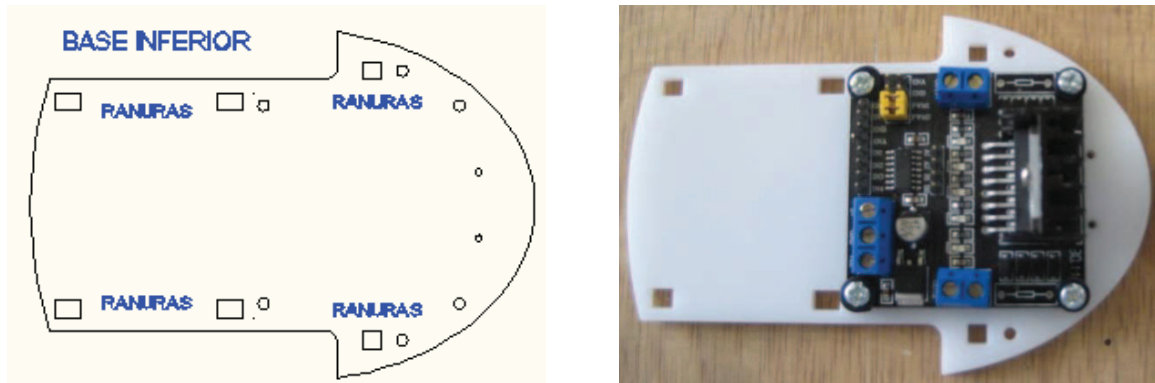


Figura A6.5 Izquierda) Base inferior dibujada en AUTOCAD, derecha) Base inferior ensamblada vista lateral superior

La base intermedia aloja a la batería, permite que el driver de la base inferior no tenga problemas de acoplamiento, además consta de ranuras para soportes y paredes.

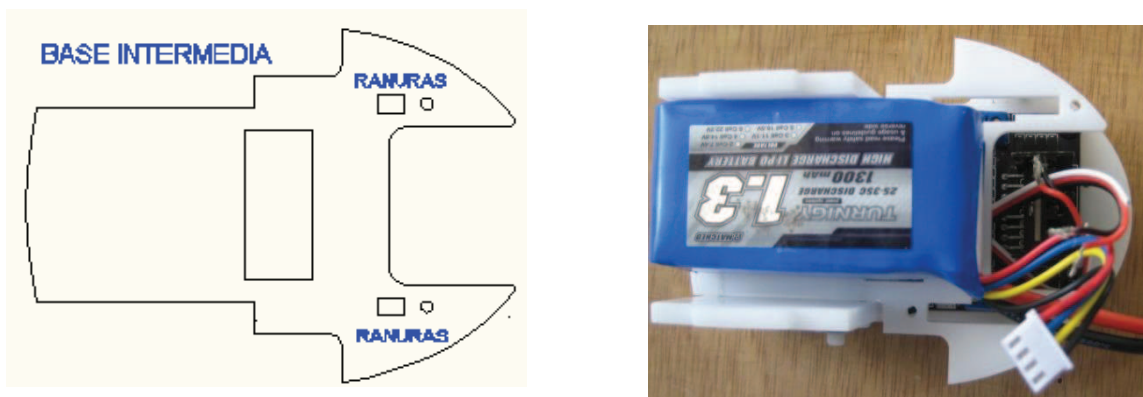


Figura A6.6 Izquierda) Base intermedia dibujada en AUTOCAD, derecha) Base intermedia ensamblada vista superior

En la base superior se coloca la placa del Arduino, el módulo Xbee y la antena, ésta base está sostenida por las paredes que se colocaron en las bases inferiores, además esta base es el soporte para los tres sensores ultrasónicos.

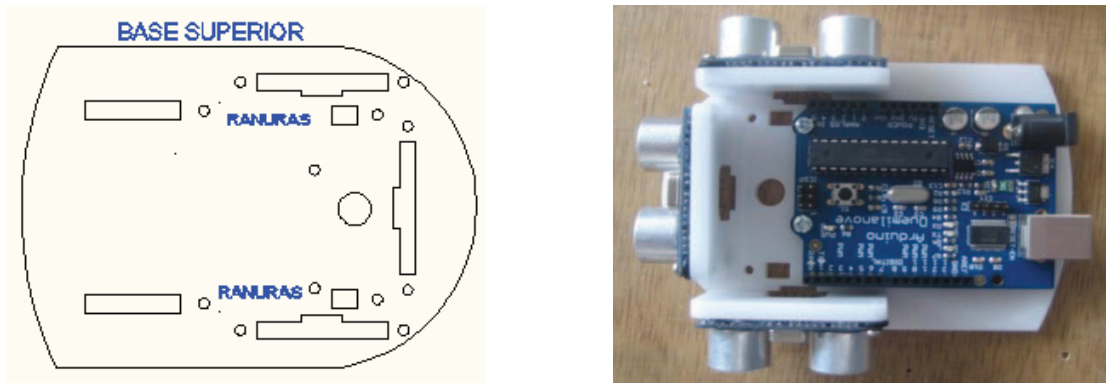


Figura A6.7 Izquierda) Base superior dibujada en AUTOCAD, derecha) Base superior ensamblada, vista superior

Además de las bases se elaboró paredes y soportes para formar de manera adecuada la estructura del minirobot, estas paredes y soportes van empotradas en las ranuras de las bases y aseguradas por medio de pernos. Finalmente se colocan y aseguran las llantas, conectores y se realizan las conexiones necesarias.



Figura A6.8 Mini robot MREC – 02

### CONSTRUCCIÓN DEL MINI ROBOT MREC – 03

Para realizar esta estructura se procedió a empotrar el driver L298D del los motores en la base inferior, se colocó la rueda de libre giro y los motores, además se añadió los soportes frontales para que pueda sostener la Base intermedia. Esta base inferior es idéntica a la base inferior del minirobot MREC – 02. Figura A6.6

La base intermedia aloja a la batería, permite que el driver de la base inferior no tenga problemas de acoplamiento, además consta de ranuras para soportes y paredes. Esta base también acopla a los dos sensores ultrasónicos laterales, esta base intermedia es fundamental ya que sostiene a la base superior y además sirve de unión con la base inferior debido a las paredes y soportes que están adheridas a esta base.

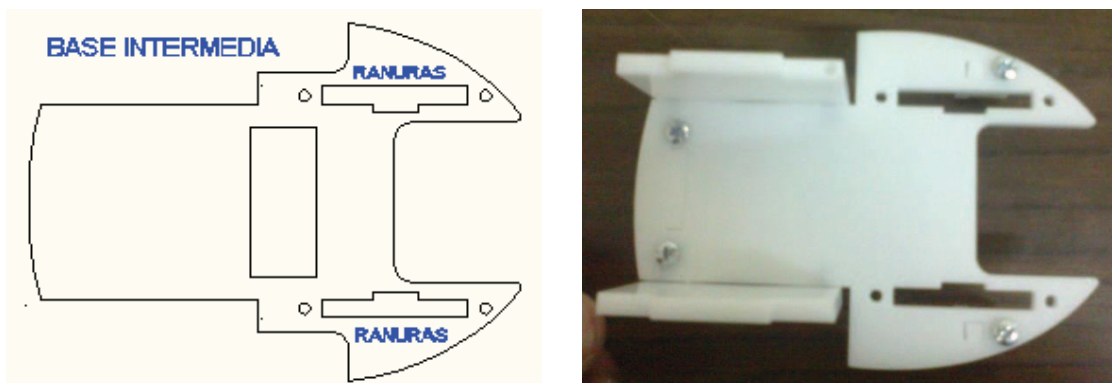


Figura A6.9 Izquierda) Base intermedia dibujada en AUTOCAD, derecha) Base intermedia ensamblada vista superior

En la base superior se coloca las placas del Arduino, el módulo Xbee y la antena, ésta base está sostenida por las paredes que se colocaron en las bases inferiores. Además la Base superior tiene al sensor ultrasónico delantero.

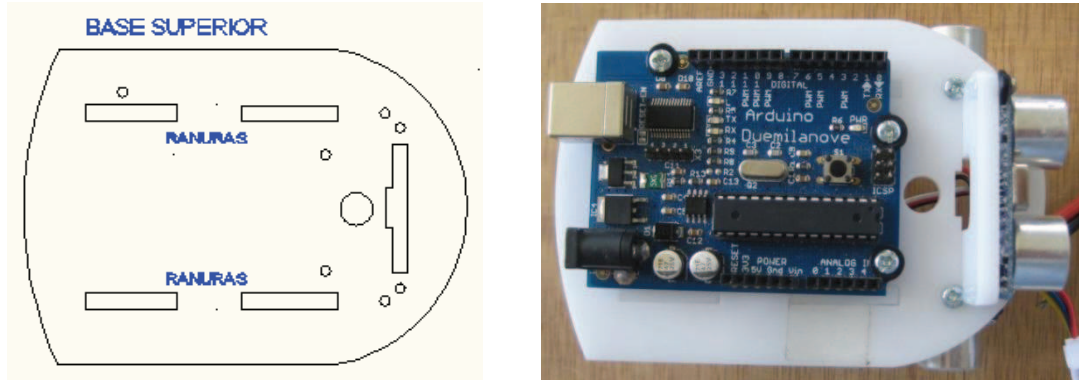


Figura A6.10 Izquierda) Base3 dibujada en AUTOCAD, derecha) Base3 ensamblada vista superior

Finalmente se ensamblan las tres bases, se aseguran con los pernos y se realiza las conexiones adecuadas.

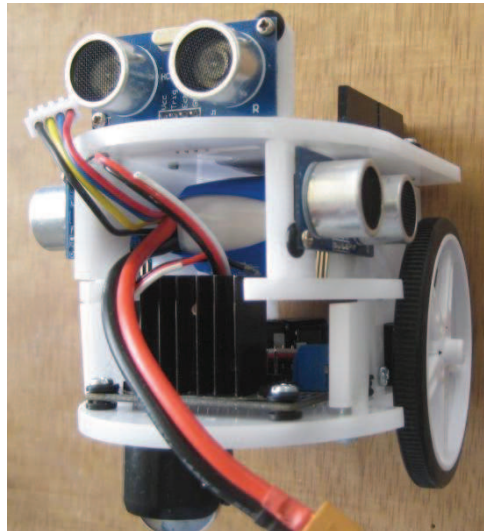


Figura A6.11 Estructura MREC – 03, vista frontal