

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DISEÑO E IMPLEMENTACIÓN DE DOS ROBOTS DE COMPETENCIA (SEGUIDOR DE LÍNEA ESPECIALIDAD VELOCISTA, LABERINTO)

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA Y CONTROL**

**ROMO CAICEDO DIEGO ANTONIO
diegoromo_408@hotmail.com**

**TOCA TRUJILLO LUIS EDISON
luisedison15@gmail.com**

**DIRECTOR: NELSON GONZALO SOTOMAYOR OROZCO, MSc.
nelson.sotomayor@epn.edu.ec**

QUITO, Agosto 2016

DECLARACIÓN

Nosotros, Romo Caicedo Diego Antonio y Toca Trujillo Luis Edison, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado 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 normativa institucional vigente.

Romo Caicedo Diego Antonio

Toca Trujillo Luis Edison

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Romo Caicedo Diego Antonio y Toca Trujillo Luis Edison, bajo mi supervisión.

Nelson Sotomayor, MSc.
DIRECTOR DE PROYECTO

AGRADECIMIENTO

Agradezco a Dios por todas sus bendiciones, a mis padres por su infinito amor y apoyo incondicional.

También a Luis, un gran compañero y a nuestro director MSc. Nelson Sotomayor.

DIEGO.

AGRADECIMIENTO

A mis padres por ser un vivo ejemplo de amor, honor y humildad. A ustedes por todo el apoyo brindado no solo en este proyecto sino durante toda mi vida.

A todos mis amigos y amigas que he hecho a lo largo de esta etapa, con quienes he pasado experiencias irrepetibles e inolvidables.

A mi amigo y compañero de tesis, Diego y a nuestro director MSc. Nelson Sotomayor por su apoyo en el desarrollo de este proyecto.

“El hombre nunca sabe de lo que es capaz hasta que lo intenta”

Charles Dickens

LUIS.

DEDICATORIA

A mis padres, Antonio y Socorro, y a mi hermano David.

Los amo.

DIEGO.

DEDICATORIA

A mis padres Luis Humberto y Blanca Esther, por siempre haber estado junto a mí en cada momento de mi vida brindándome su cariño, enseñanzas y apoyo incondicional. A ustedes por todo su amor y consejos, que me han ayudado a crecer como una mejor persona. Y por todo su trabajo y empeño para que sus hijos salgamos adelante y podamos cumplir con nuestros sueños.

LUIS.

CONTENIDO

RESUMEN.....	1
PRESENTACIÓN.....	2
CAPÍTULO 1.....	3
FUNDAMENTOS TEÓRICOS	3
1.1 INTRODUCCIÓN.....	3
1.2 ROBOTS DE COMPETENCIA.....	3
1.2.1 FUNCIONAMIENTO DE UN ROBOT EN UNA COMPETENCIA....	4
1.2.1.1 Robot autónomo.....	5
1.2.1.2 Robot radio controlado.....	5
1.2.2 SEGUIDORES DE LÍNEA VELOCIDAD.....	6
1.2.2.1 Requerimientos técnicos del prototipo.....	7
1.2.2.2 Requerimientos técnicos de las pistas.....	7
1.2.3 ROBOT LABERINTO.....	7
1.2.3.1 Requerimientos técnicos del prototipo.....	8
1.2.3.2 Requerimientos técnicos de la pista.....	9
1.3 GENERALIDADES DEL ROBOT SEGUIDOR DE LÍNEA Y ROBOT LABERINTO.....	9
1.3.1 TIPOS DE SISTEMA DE LOCOMOCIÓN.....	10
1.3.1.1 Tracción diferencial.....	10
1.3.1.2 Triciclo.....	11
1.3.1.3 Ackerman.....	12
1.3.2 MOTORES.....	13
1.3.2.1 Micromotores.....	13
1.3.2.2 Motor de engranajes.....	14
1.3.3 SENSORES.....	14
1.3.3.1 Sensores para robot seguidor de línea velocidad.....	15
1.3.3.1.1 Sensor de reflectancia analógico (QTR-8A).....	15
1.3.3.1.2 Sensor de reflectancia digital (QTR-8RC).....	16
1.3.3.1.3 Matriz para seguidor de línea SparkFun.....	16
1.3.3.2 Sensores para robot laberinto.....	17
1.3.3.2.1 Sensores analógicos Sharp GP2Y0A51SK0F.....	17
1.3.3.2.2 Sensor ultrasónico Parallax PING.....	18
1.3.3.3 Encoder.....	19
1.3.3.3.1 Encoder magnético para micromotores.....	19
1.3.3.3.2 Encoder óptico para micromotores	20
1.3.3.3.3 Encoder óptico para rueda 42 x 19 mm.....	21
1.3.4 ROBOT SEGUIDORES DE LÍNEA COMERCIALES.....	21
1.3.4.1 Robot 3pi.....	21
1.3.4.2 Seguidor de línea del club Rutgers IEEE.....	22
1.3.4.3 Robomart.....	22

1.3.5	ROBOT LABERINTO COMERCIALES.....	23
1.3.5.1	Micromouse PICAXE "PICone".....	23
1.4	SELECCIÓN DE SISTEMA DE LOCOMOCIÓN.....	23
1.4.1	MODELO CINEMÁTICO DE LOS ROBOTS A IMPLEMENTARSE	24
1.5	COMPONENTES UTILIZADOS PARA LOS ROBOTS A IMPLEMENTARSE.....	28
1.5.1	COMPONENTES DEL SEGUIDOR DE LÍNEA VELOCIDAD.....	28
1.5.1.1	Sensor de reflectancia analógico (QTR-8A).....	28
1.5.1.2	Micromotor 10:1 HP de rotor extendido.....	29
1.5.1.3	Encoder magnético de cuadratura.....	30
1.5.2	COMPONENTES UTILIZADOS PARA EL ROBOT LABERINTO..	30
1.5.2.1	Sensores infrarrojos de 5 mm.....	30
1.5.2.2	Micromotor 50:1 HP.....	31
1.5.2.3	Encoders óptico de cuadratura.....	31
CAPÍTULO 2		33
DISEÑO E IMPLEMENTACIÓN DE LOS ROBOTS		33
2.1	INTRODUCCIÓN.....	33
2.2	ARQUITECTURA DE HARDWARE.....	33
2.2.1	DISEÑO MECÁNICO DEL ROBOT SEGUIDOR DE LÍNEA.....	34
2.2.1.1	Chasis.....	34
2.2.1.1.1	Forma.....	34
2.2.1.1.2	Dimensiones.....	35
2.2.1.2	Disposición de sensores.....	37
2.2.1.3	Motores y ruedas.....	37
2.2.2	DISEÑO MECÁNICO DEL ROBOT LABERINTO.....	40
2.2.2.1	Chasis.....	40
2.2.2.1.1	Forma.....	40
2.2.2.1.2	Dimensiones.....	42
2.2.2.2	Disposición de sensores.....	42
2.2.2.3	Motores y ruedas.....	44
2.3	DISEÑO ELECTRÓNICO DEL ROBOT SEGUIDOR DE LÍNEA.....	45
2.3.1	ARQUITECTURA.....	45
2.3.2	SISTEMA DE CONTROL.....	46
2.3.3	SISTEMA SENSORIAL.....	47
2.3.3.1	Sensores de línea.....	47
2.3.3.2	Encoder magnético.....	48
2.3.3.3	Medición de batería.....	48
2.3.4	CONTROLADOR DE POTENCIA.....	49
2.4	DISEÑO ELECTRÓNICO DEL ROBOT LABERINTO.....	52
2.4.1	ARQUITECTURA.....	52
2.4.2	SISTEMA DE CONTROL.....	53
2.4.3	SISTEMA SENSORIAL.....	53

2.4.3.1	Sensores de distancia.....	53
2.4.3.2	Encoder óptico.....	55
2.4.3.3	Medición de batería.....	56
2.4.4	CONTROL DE MOTORES.....	56
2.5	PERIFÉRICOS DE ENTRADA-SALIDA.....	58
2.5.1	PROGRAMADOR.....	58
2.5.2	COMUNICACIÓN.....	58
2.5.3	PULSADOR.....	58
2.5.4	INDICADORES.....	58
2.6	ROBOTS DISEÑADOS LISTOS PARA SU CONSTRUCCIÓN.....	59
CAPÍTULO 3		61
DESARROLLO DE LOS PROGRAMAS DE CONTROL		61
3.1	INTRODUCCIÓN.....	61
3.2	ARQUITECTURA SOFTWARE PARA ROBOT SEGUIDOR DE LÍNEA....	61
3.2.1	MÓDULO CONFIGURACIÓN DE PERIFÉRICOS.....	62
3.2.2	MÓDULO SENSORES.....	62
3.2.2.1	Función calibración de sensores.....	63
3.2.2.2	Función lectura y acondicionamiento de sensor de línea.....	63
3.2.2.3	Función cálculo del error de posición.....	65
3.2.2.4	Función sensar batería.....	66
3.2.2.5	Función cálculo de la distancia recorrida.....	66
3.2.3	MÓDULO MOTORES.....	68
3.2.3.1	Función control de motor derecho/izquierdo.....	68
3.2.4	MÓDULO INTERFAZ.....	69
3.2.5	MÓDULO COMUNICACIÓN.....	70
3.2.6	MÓDULO DE CONTROL.....	70
3.2.6.1	Características del sistema.....	70
3.2.6.2	Control Proporcional Integral Derivativo (PID).....	71
3.2.6.2.1	Implementación del controlador PID.....	72
3.2.6.2.2	Adición de control en lazo abierto.....	74
3.2.6.2.3	Arranque.....	74
3.2.6.3	Controlador Borroso.....	75
3.2.6.3.1	Borrosificación.....	75
3.2.6.3.2	Reglas de inferencia.....	78
3.2.6.3.3	Selección de reglas.....	79
3.2.6.3.4	Aplicación de la regla.....	80
3.2.6.3.5	Desborrosificación.....	80
3.2.7	PROGRAMA PRINCIPAL.....	80
3.3	ARQUITECTURA SOFTWARE PARA ROBOT LABERINTO.....	84
3.3.1	MÓDULO CONFIGURACIÓN DE PERIFÉRICOS.....	84
3.3.2	MÓDULO LECTURA DE SENSORES.....	85
3.3.2.1	Función acondicionamiento del sensor de pared.....	85

3.3.2.2	Función detectar paredes.....	87
3.3.2.3	Función obtener error de posición.....	87
3.3.3	MÓDULO MOTORES.....	88
3.3.4	MÓDULO INTERFAZ.....	88
3.3.5	MÓDULO DE CONTROL.....	88
3.3.5.1	Función avance con pared derecha.....	89
3.3.5.2	Función avanzar con dos paredes.....	90
3.3.5.3	Función avanzar una celda.....	91
3.3.5.4	Función girar a la derecha en su propio eje.....	93
3.3.5.5	Función giro 180° en su propio eje.....	94
3.3.5.6	Función girar a la derecha.....	94
3.3.5.7	Función posicionar con pared frontal.....	96
3.3.6	PROGRAMA PRINCIPAL.....	98
3.3.6.1	Algoritmo de la mano derecha o izquierda.....	99
3.3.6.2	Algoritmo de relleno de callejones sin salida.....	99
CAPITULO 4	107
PRUEBAS Y RESULTADOS	107
4.1	PRUEBAS EN EL ROBOT SEGUIDOR DE LÍNEA.....	107
4.1.1	PRUEBAS DE HARDWARE.....	107
4.1.2	PRUEBAS DE SOFTWARE.....	108
4.1.3	PRUEBAS EN PISTA.....	110
4.2	RESULTADOS DEL ROBOT SEGUIDOR DE LÍNEA.....	113
4.3	CARACTERÍSTICAS DEL ROBOT SEGUIDOR DE LÍNEA.....	114
4.4	PRUEBAS EN EL ROBOT LABERINTO.....	115
4.4.1	PRUEBAS DE HARDWARE.....	115
4.4.2	PRUEBAS DE SOFTWARE.....	116
4.4.3	PRUEBAS EN PISTA.....	116
4.5	RESULTADOS ROBOT LABERINTO.....	118
4.6	CARACTERÍSTICAS DEL ROBOT LABERINTO.....	118
4.7	COSTOS DE IMPLEMENTACIÓN DE LOS ROBOTS.....	119
CAPÍTULO 5	122
CONCLUSIONES Y RECOMENDACIONES	122
5.1	CONCLUSIONES.....	122
5.2	RECOMENDACIONES.....	124
REFERENCIAS BIBLIOGRÁFICAS	126
ANEXOS		

RESUMEN

En los últimos años, la robótica de competencia ha tenido una mayor acogida en nuestro país; existen muchos torneos a nivel nacional como a nivel internacional por esta razón se ha realizado una investigación del desarrollo de este tipo de robots, enfocándose en las categorías seguidor de línea velocidad y robot laberinto.

En el presente trabajo se diseña e implementa un robot seguidor de línea velocidad y un robot laberinto con las especificaciones técnicas del Concurso Ecuatoriano de Robótica; para el diseño de los robots se partió desde la elección de la configuración de locomoción, y se realiza un análisis del modelo cinemático empleado.

Para el diseño mecánico de los prototipos se toma en cuenta el ambiente en el que se van a desenvolver, y aspectos tales como: forma del chasis, peso del prototipo, distribución de elementos en el chasis, motor y llantas a utilizarse.

Se realiza el diseño de la electrónica de los prototipos, el cual está compuesto de un sistema de control, un sistema sensorial, un sistema de periféricos de entrada salida y un controlador de potencia con el que se controla los motores.

De acuerdo a la tarea que cada robot debe realizar, se desarrolla los algoritmos de control. El objetivo del seguidor de línea es recorrer un circuito en el menor tiempo posible, para cumplirlo se ha utilizado un controlador borroso y un controlador PID. El objetivo del robot laberinto es salir de un laberinto en el menor tiempo posible, es este caso se ha implementado el algoritmo de la mano derecha, el algoritmo de la mano izquierda y el algoritmo de relleno de callejones sin salida.

PRESENTACIÓN

En el presente proyecto se realiza el diseño y construcción de robots de competencia en las categorías robot seguidor de línea y robot laberinto, de esta manera se pretende afianzar las bases para la construcción y desarrollo de los mismos, permitiendo que los estudiantes tengan un mayor interés en elaborar sus prototipos.

El proyecto ha sido dividido en cinco capítulos: en el primer capítulo se presenta las generalidades de la robótica de competencia y se realiza un estudio bibliográfico de los robots, finalmente se escoge el sistema de locomoción más adecuado para la implementación de los prototipos y se presentan los elementos a utilizarse en su construcción.

La descripción de la parte mecánica de los robots y la ubicación de los elementos, así como el acondicionamiento de los sensores y los circuitos diseñados para el funcionamiento de los prototipos se presentan en el capítulo 2.

En el capítulo 3 se presenta la estructura del programa que controla a los prototipos y los diferentes algoritmos que utilizan para cumplir sus objetivos.

Las pruebas realizadas a los prototipos así como los resultados obtenidos en ellas se presentan en el capítulo 4; también se habla de los torneos en los que participaron los robots y los resultados obtenidos.

Finalmente en el capítulo 5, se encuentran las conclusiones y recomendaciones que servirán como aporte para futuras investigaciones y el continuo mejoramiento de los robots.

CAPÍTULO 1

FUNDAMENTOS TEÓRICOS

1.1 INTRODUCCIÓN

En este capítulo se presenta generalidades de la robótica de competencia, los principales torneos en el país, las categorías más populares; así como también algunas reglas, las cuales rigen para la mayoría de torneos en lo referente a las características técnicas de los prototipos y a los escenarios en los cuales estos se van a desenvolver.

Se realiza un estudio bibliográfico del robot seguidor de línea velocidad y del robot laberinto, revisando en esta parte los aspectos principales para la construcción de estos robots; también se revisa algunos robots seguidores de línea y laberinto comerciales.

Finalmente se escoge el sistema de locomoción más adecuado para la implementación de los robots y se realiza el análisis cinemático, detallando los componentes a usar en la construcción de los prototipos.

1.2 ROBOTS DE COMPETENCIA

Una competición de robótica es un evento donde participan prototipos de robots, los cuales pueden ser contruidos o adquiridos en casas comerciales y programados por los participantes para realizar una tarea específica cumpliendo con un reglamento preestablecido por la organización, con el fin de superar al resto de participantes. Las características mecánicas como electrónicas de los prototipos están definidas por los organizadores de las competencias.

En los torneos de robótica existen diferentes categorías entre las más populares se encuentran [1]:

1. Categoría Libre
2. Batalla de Robots

3. Carrera de Humanoides
4. Pelea de Bípedos
5. Robot Volador
6. Robot Transformers
7. Robot Bailarín
8. Robot Futbol
9. Robot Seguidor de línea
10. Robot Sumo
11. Robot Trepador
12. Robot Laberinto

En Ecuador se tiene muchos torneos de robótica que se realizan año a año, uno de los más importantes es el Concurso Ecuatoriano de Robótica (CER) en el cual compiten Universidades, Escuelas Politécnicas, e Institutos Tecnológicos del país.

Otro torneo que está tomando acogida en los últimos años es el Robot Games Zero Latitud, en este torneo participan delegaciones de varios países y se usan reglamentos similares al Concurso Ecuatoriano de Robótica con algunas modificaciones.

El RobotChallenge es uno de los torneos más importantes en el campo de la robótica en el mundo y se lo realiza en Viena, Austria desde el 2004. Este concurso tiene 14 categorías. A este torneo asisten delegaciones de muchas partes del mundo para demostrar su talento en la robótica.

1.2.1 FUNCIONAMIENTO DE UN ROBOT EN UNA COMPETENCIA

Ya se mencionó que en un torneo de robótica existen muchas categorías, cada una con su respectivo reglamento; en este marco a los robots que participan en los torneos se los puede clasificar en dos grandes categorías: Robots Autónomos y Robots Radio Controlados.

1.2.1.1 Robot autónomo [2]

Un robot autónomo tiene la capacidad de actuar en su entorno y cumplir con una tarea establecida sin intervención de un agente externo en base a la información que pueda recibir de su entorno mediante la utilización de sensores.

Otra característica importante en un robot autónomo es que debe poseer una fuente propia de energía; por lo general, en robots móviles la alimentación se realiza mediante baterías con lo que la autonomía de los prototipos es limitada por la capacidad de éstas; por lo que este es un punto muy importante que se debe tener en cuenta al momento de diseñar un prototipo.

Un robot autónomo tiene la capacidad de:

- Obtener información del entorno.
- Trabajar en entornos desconocidos.
- Moverse sin ayuda humana.
- Tener una fuente propia de energía.

En los torneos de robótica las categorías más populares en lo que se refiere a robots autónomos son: robots sumo, robots seguidores de línea, robots laberinto, entre otros.

1.2.1.2 Robot radio controlado [2]

Este tipo de robots necesitan de la intervención de un humano; las tareas de percepción del medio, planificación y acciones son ejecutadas por quien controla al robot. En algunos casos este tipo de robots también poseen sensores para proporcionar información extra al usuario (imágenes, distancia, velocidad) para que este pueda tomar mejores decisiones.

Dentro de los robots radio controlados las categorías que más acogida tienen en los torneos de robótica son: robot futbol, batalla de robot y mega sumo RC.

1.2.2 SEGUIDORES DE LÍNEA VELOCIDAD

Esta es una de las categorías más populares dentro de los torneos de robótica. El torneo de seguidores de línea velocidad se celebra cada año en el Concurso Ecuatoriano de Robótica (CER). La tarea del robot es seguir una pista marcada con una línea negra en un fondo blanco. La complejidad de la pista depende del reglamento de los organizadores del torneo. En la Figura 1.1 se puede observar la pista de la categoría robot seguidor de línea velocidad del Concurso Ecuatoriano de Robótica (CER) 2015 celebrado en La Universidad Santiago de Guayaquil.



Figura 1.1 Pista de la categoría robot seguidor de línea velocidad CER 2015 [3].

En la Figura 1.2 se muestra un robot seguidor de línea velocidad, este es uno de los diseños más populares en esta categoría.



Figura 1.2 Robot seguidor de línea velocidad [4].

El desarrollo de este proyecto se lo realizará bajo el reglamento del Concurso Ecuatoriano de Robótica (CER).

1.2.2.1 Requerimientos técnicos del prototipo [3]

Algunos de los requerimientos y restricciones de los prototipos son:

- “El prototipo debe ser totalmente autónomo, es decir que por ningún motivo puede ser manejado a control remoto, por computadora o cualquier dispositivo que se encuentre en el exterior.
- No existe limitaciones en el tamaño, altura o peso del prototipo.
- La fuente de energía para alimentar los circuitos electrónicos así como los motores del prototipo, será de cualquier tipo de baterías de corriente continua.
- El accionamiento del prototipo será en forma manual.
- El prototipo no puede tener ningún tipo de mecanismo de succión para aumentar la fuerza de contacto con el suelo.
- El prototipo podrá ser adquirido en casas comerciales y se privilegiara la programación y los algoritmos de control para llevar el robot a la meta en el menor tiempo posible.”

1.2.2.2 Requerimientos técnicos de las pistas [3]

Los escenarios para seguidor de línea velocidad tendrán fondo blanco y las pistas serán hechas con cinta aislante de color negro. Estas serán diseñadas a criterio de los organizadores y serán publicadas con un mes de anterioridad en el sitio web del concurso. La pista estará hecha con líneas rectas y curvas y no existirán discontinuidades ni intersecciones.

Como referencia se adjunta el reglamento completo en el Anexo C.

1.2.3 ROBOT LABERINTO

En esta categoría, el prototipo es autónomo y debe enfrentarse a la tarea de salir de un laberinto; al robot se lo ubica en la entrada de éste y el prototipo debe ser capaz de encontrar su salida en un tiempo máximo preestablecido. En la Figura 1.3 se muestra un robot laberinto.

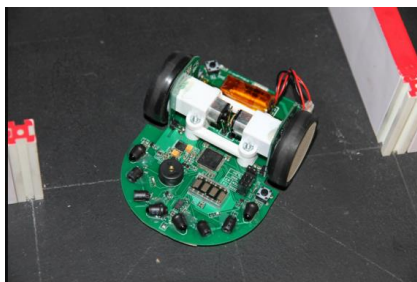


Figura 1.3 Robot laberinto [5].

El desarrollo de este proyecto se lo realizará bajo el reglamento del Concurso Ecuatoriano de Robótica (CER).

1.2.3.1 Requerimientos técnicos del prototipo [3]

Algunos de los requerimientos y restricciones de los prototipos son:

- “El robot no podrá tener dimensiones de largo y ancho mayores a 15 cm y altura de 20 cm, El robot no puede expandir sus dimensiones hacia ninguna dirección durante la competencia y no existe restricciones de peso.
- La fuente de energía para alimentar los circuitos eléctricos de control y los motores del robot será interna, considerándose cualquier tipo de baterías de corriente continua.
- El robot no puede tener ningún tipo de control externo, es decir, el robot tiene que ser totalmente autónomo.
- Se permite cualquier tipo de sensores para el sistema de navegación del robot, queda prohibido el uso de switches, fines de carrera u otros mecanismos de detección por contacto, por lo que el robot durante la resolución del laberinto no puede topar las paredes, solamente las ruedas pueden estar en contacto sobre la pista, el robot no puede dejar marcas o indicios del camino recorrido.
- Queda prohibido el uso de robots enteramente comerciales, sin embargo se puede utilizar tarjetas de desarrollo como partes del robot.”

1.2.3.2 Requerimientos técnicos de la pista [3]

La pista será construida con madera de 2 cm de espesor con un fondo blanco mate, las paredes tendrán un espesor y altura de 4 mm y 20 cm respectivamente y entre paredes de la pista habrá una distancia de 25 cm. La madera sobre la cual se construirá la pista tendrá dimensiones de 2 x 2 m con una tolerancia de $\pm 10\%$. La Figura 1.4 presenta un modelo de pista con el fin de indicar las dimensiones. Como referencia se adjunta el reglamento completo en el Anexo C.

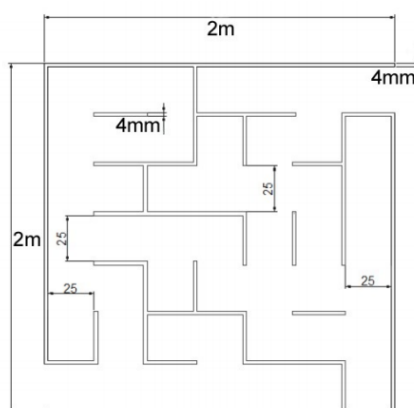


Figura 1.4 Laberinto modelo [3].

1.3 GENERALIDADES DEL ROBOT SEGUIDOR DE LÍNEA Y ROBOT LABERINTO

La robótica pone a prueba los conocimientos de los estudiantes al momento de diseñar y construir un prototipo, ya que al ser un área interdisciplinaria hay que tomar en cuenta muchos factores primero a nivel de hardware, como los siguientes:

- Sistema de locomoción
- Sensores
- Actuadores
- Baterías
- Ruedas
- Peso del prototipo

Y luego en el software, ya que hay que desarrollar un programa que controle al robot, tomando información del entorno por medio de los sensores, para luego procesarla y tomar decisiones, las cuales son enviadas a los actuadores para cumplir con la tarea para la cual el robot fue construido.

1.3.1 TIPOS DE SISTEMA DE LOCOMOCIÓN

El sistema de locomoción es el modo en que va desplazarse el robot, el más usado en robots de competencia es el de ruedas, ya que tiene las siguientes ventajas: mecánicamente sencillo, menor coste y permite alcanzar altas velocidades. Dentro de los sistemas de locomoción por ruedas existen diferentes configuraciones, las más comúnmente usadas son las siguientes: tracción diferencial, triciclo, y ackerman.

1.3.1.1 Tracción diferencial [6]

Esta configuración posee 2 ruedas colocadas paralelamente entre sí, cada rueda es controlada por un motor, los giros se lo realiza por diferencia de velocidades y sentidos. Ya que no es posible estabilizar la plataforma con solo dos ruedas se necesita de una o dos adicionales. Para desplazarse en línea recta ambas ruedas deben girar a la misma velocidad; para girar, una de las ruedas debe tener una velocidad diferente de la otra y para girar sobre el centro de gravedad ambas ruedas deben tener la misma velocidad pero sentidos contrarios. En la Figura 1.5 se puede observar la configuración de un robot diferencial.

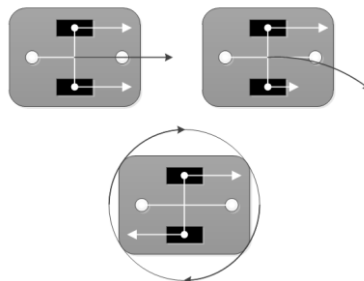


Figura 1.5 Conducción y rotación de un robot con configuración diferencial.

Ventajas:

- Sistema barato y fácil de implementar.
- Puede girar sobre su propio eje. El cual está localizado en el punto medio entre la distancia de las ruedas de tracción, por esta razón tiene una mejor respuesta ante curvas cerradas.
- Diseño simple porque solo se debe considerar la distancia entre las ruedas de tracción.

Inconvenientes:

- Para que el robot realice un movimiento rectilíneo se debe asegurar que las 2 ruedas giren a la misma velocidad.
- Tiene problemas con el equilibrio y debido a esto necesita de una tercera o hasta cuarta rueda de apoyo.

1.3.1.2 Triciclo [6]

Esta configuración consta de tres ruedas; se la utiliza de dos formas: la primera, en la que la rueda delantera se la utiliza para tracción y dirección y las ruedas posteriores de soporte; y la otra, en la que la rueda delantera se la utiliza para la dirección y las ruedas posteriores para tracción. En esta configuración la velocidad lineal esta desacoplada de la angular. El radio mínimo de giro está condicionado por la distancia entre la rueda delantera y el punto medio entre las ruedas traseras. En la Figura 1.6 se muestra un robot móvil tipo triciclo.

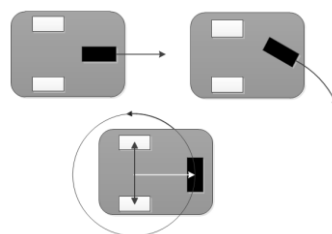


Figura 1.6 Conducción y rotación de un robot con configuración triciclo.

Ventajas:

- Fácil desplazamiento en línea recta.
- Se puede ir a alta velocidad siempre que el radio de curvatura de la pista sea mayor al del robot [6].

Inconvenientes:

- No tiene la capacidad de girar sobre su centro de gravedad.
- Su construcción mecánica es más complicada que la de tracción diferencial.

1.3.1.3 Ackerman [6]

Posee cuatro ruedas, dos traseras de tracción y dos delanteras que le permiten el cambio de dirección, es el sistema de locomoción de un automóvil. Para desplazarse en línea recta no hay problema ya que las llantas de atrás son controladas por un mismo motor (Figura 1.7).

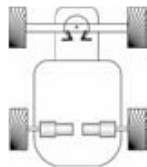


Figura 1.7 Robot ackerman [7].

Ventajas:

- Fácil desplazamiento en línea recta.
- Posee buena estabilidad ya que está apoyado en las cuatro ruedas en todo momento.

Inconvenientes:

- Se desliza en curvas.
- No puede girar sobre su centro de gravedad.
- Mecánicamente más complejo.

- Se debe reducir la velocidad al momento de entrar a una curva ya que el radio de curvatura del sistema no es tan pequeño.

1.3.2 MOTORES

Estos son los elementos finales del lazo de control que provocan movimiento en el robot, estos movimientos se realizan según las ordenes que envié el sistema de control. Existen varios modelos de motores con los cuales se podría trabajar, hay de varios tamaños, de diferentes velocidades, algunos traen caja reductora y otros no, pero en este tipo de robots lo que se busca son motores pequeños para que no ocupen mucho espacio en el robot, que sean livianos ya que el peso del robot es crucial para mejorar su rendimiento, especialmente en las curvas, ya que a mayor peso es más difícil que gire con rapidez, que tengan un buen torque y velocidad ya que esto va a permitir acelerar y frenar más rápido; todo esto se debe tomar en cuenta ya que el nivel de las competencias es tal que se gana o se pierde por décimas de segundo. A continuación se presentan algunos motores que existen en el mercado:

1.3.2.1 Micromotores [8]

Estos motores vienen integrados con una caja reductora, las relaciones de éstas las hay en una amplia gama desde 5:1 hasta 1000:1 y existe cuatro tipos de motores: de alta potencia con escobillas de carbón (HPCB), alta potencia (HP), media potencia (MP) y de baja potencia. La alimentación de estos motores esta entre 3 a 9 V. En la Figura 1.8 se observa un micromotor 10:1 HP.



Figura 1.8 Micromotor 10:1 alta potencia [8].

En esta misma línea de motores hay una serie que tiene la característica de que le sobresale un eje por la parte posterior, estos son llamados motores de rotor extendido; el eje es usado para colocar un encoder. En la Figura 1.9 se muestra un micromotor 10:1 HP con rotor extendido.



Figura 1.9 Micromotor 10:1 alta potencia con rotor extendido [9].

1.3.2.2 Motor de engranajes [10]

Como en los anteriores motores estos también vienen provistos de una caja reductora pero esta es plástica, la cual permite alcanzar una velocidad de 200 RPM a un voltaje de alimentación de 6 V. Este puede soportar voltajes de alimentación entre 3 a 6 V con un consumo de corrientes sin carga de 200 mA a 6 V (Figura 1.10).



Figura 1.10 Motor de engranajes [10].

1.3.3 SENSORES

La parte sensorial de un robot es muy importante, para su buen funcionamiento se necesita sensores, en el caso de este proyecto, para la detección de la línea, para la detección de una pared, sensores para medir la distancia que se ha recorrido, etc. Por lo que se necesita hacer una buena elección de los sensores a utilizarse.

1.3.3.1 Sensores para robot seguidor de línea velocidad

En lo que se refiere al robot seguidor de línea velocista lo primero que se debe hacer es detectar la línea, es decir, saber en dónde está la línea con respecto al robot. Para esto se usa sensores infrarrojos que están formados por un led y un fototransistor, el diodo led emite una radiación en el espectro de los infrarrojos, esta incide en el suelo y se da una reflexión, la cual es captada por el fototransistor, la cantidad de reflexión depende del color del suelo.

Para trabajar con estos se debe hacer el circuito correspondiente tanto para el emisor como para el receptor, si uno mismo se los va a diseñar; pero también hay placas de sensores comerciales que se puede adquirir.

Estos sensores se los puede trabajar en forma digital o en forma analógica, esto depende del tipo de estimación que se vaya realizar (forma de detectar la posición del robot con respecto a la línea) y de los elementos que se dispone para la construcción del robot. A continuación se va a ver algunos sensores comerciales:

1.3.3.1.1 Sensor de reflectancia analógico (QTR-8A) [11]

El módulo QTR-8A posee 8 sensores infrarrojos, cada sensor proporciona una salida de voltaje por separado. Este módulo de sensores se lo debe leer de forma analógica. El fabricante de estos sensores da la facilidad de usar librerías de Arduino para la adquisición de la señal de voltaje. En la Figura 1.11 se puede observar un sensor QTR-8A.

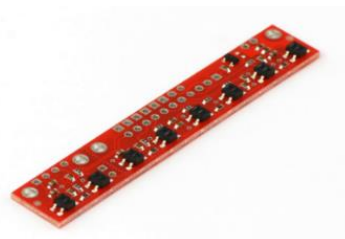


Figura 1.11 Sensores de reflectancia analógico QTR – 8A [11].

Características:

- Tensión de funcionamiento: 3.3 a 5 V.
- Formato de salida: 8 tensiones analógicas.

1.3.3.1.2 Sensor de reflectancia digital (QTR-8RC) [12]

Este módulo posee 8 sensores, estos sensores proporcionan una salida de tipo digital. Como ventaja, el fabricante proporciona librerías de Arduino para su utilización. En la Figura 1.12 se puede observar un sensor QTR-8RC.



Figura 1.12 Sensor de reflectancia digital QTR-8RC [12].

Características:

- Tensión de funcionamiento: 3.3 a 5 V
- Formato de salida: 8 señales digitales
- Peso: 3.09 g sin pines

1.3.3.1.3 Matriz para seguidor de línea SparkFun [13]

Es una placa con 8 sensores infrarrojos que han sido configurados para leerlos en forma digital. Entre otras cosas esta placa tiene leds en los cuales el usuario puede ver lo que se está detectando, tiene una interface i2c para la lectura de los sensores; también posee un potenciómetro para regular el nivel umbral de los sensores (Figura 1.13).



Figura 1.13 Matriz para seguidor de línea SparkFun [13].

Características:

- Tensión de funcionamiento: 5 V
- Corriente de alimentación: 185 mA
- Comunicación I2C

1.3.3.2 Sensores para robot laberinto

La tarea inicial aquí es detectar la pared y saber a qué distancia se encuentra el robot con respecto a esta. Hay que tener en cuenta que los reglamentos para esta categoría no permiten sensores por contacto. Para poder realizar esta tarea existen básicamente dos tipos de sensores que se podría utilizar: sensores infrarrojos o sensores ultrasónicos cada uno de ellos con sus respectivas ventajas e inconvenientes. En lo que se refiere a los sensores infrarrojos, se mide la distancia por medio de la cantidad de radiación infrarroja reflejada en la pared; al trabajar estos sensores de esta manera hay el problema de que estos se ven afectados por la luz ambiental. Por otra parte los sensores ultrasónicos, no se ven afectados por la luz ambiental ya que estos miden la distancia por medio del tiempo de vuelo que le toma al sonido en ir desde el emisor rebotar en la pared y regresar al receptor. Pero tiene el problema de que es más lenta la medición que en el sensor infrarrojo. Aquí también hay la opción de construirse los sensores o adquirirlos ya listos para su utilización.

1.3.3.2.1 Sensores analógicos Sharp GP2Y0A51SK0F [14]

Este es un sensor analógico infrarrojo de distancia, el cual proporciona una detección entre 2 a 15 cm con un rango de salida de voltaje de 2.5 a 0.4 V respectivamente. Cuando a la salida se tiene un voltaje pequeño los objetos están alejados, al tener un voltaje alto los objetos están cerca. La salida de voltaje del sensor se puede conectar a un convertidor analógico digital del microcontrolador. En la Figura 1.14 se puede apreciar el sensor.



Figura 1.14 Sensor analógico de distancia sharp GP2Y0A51SK0F [14].

Características:

- Tensión de funcionamiento: 4.5 V a 5.5 V
- Consumo de corriente medio: 12 mA
- Rango de medición de distancia: 2 a 15 cm
- Tipo de salida: voltaje analógico.
- Periodo de actualizaciones: 16.5 ± 4 ms
- Tamaño: 27 mm x 13.2 mm x 14.2 mm
- Peso: 2.7 g

1.3.3.2.2 *Sensor ultrasónico Parallax PING [15]*

Este es un sensor ultrasónico para la medición de distancia, la frecuencia de la señal ultrasónica que emite está por encima de la frecuencia de audición humana, se transmite un pulso desde el dispositivo y la distancia del objeto se determina midiendo el tiempo requerido para el retorno del eco. Su rango de detección está entre los 2 cm a 3 m. también posee un led el cual indica la activación o desactivación del sensor (Figura 1.15).



Figura 1.15 Sensor ultrasónico parallax PING [15].

Características:

- Tensión de funcionamiento: 4,5 V a 6 V.
- Consumo de corriente medio: 30 mA.

- Rango de medición de distancia: 2 cm a 3 m
- Tipo de salida: digital
- Periodo de actualizaciones: 16.5 ± 4 ms
- Tamaño: 22 mm x 46 mm x 16 mm

1.3.3.3 Encoder [2]

El encoder es un transductor que transforma un movimiento rotativo en pulsos digitales, para poder obtener información angular del dispositivo que está girando. A este tipo de sensores se los puede clasificar de varias maneras por ejemplo, según su funcionalidad se los puede clasificar en encoders absolutos y en incrementales y según la tecnología usada en su mecanismo estos pueden ser ópticos, magnéticos, inductivos o capacitivos.

Los encoders incrementales, por lo general entregan dos ondas cuadradas, las cuales están desfasadas 90 grados. Si se desea únicamente obtener la velocidad a la cual gira el motor se podría leer solo una salida, pero no se podría saber en qué sentido gira, es ahí donde entra en juego la otra señal ya que con las dos se puede determinar el sentido de giro.

En lo que se refiere a encoders absolutos, estos entregan a su salida un código binario que indica la posición absoluta del dispositivo que está girando. En el mercado existen varios modelos de encoder que se podría utilizar o incluso se podría fabricar uno artesanalmente.

1.3.3.3.1 Encoder magnético para micromotores [16]

Este es un encoder de tipo incremental, está especialmente diseñado para micromotores de rotor extendido. Este encoder da en su salida dos señales digitales en cuadratura las cuales se las puede conectar directamente al sistema de control. En la Figura 1.16 se muestran este tipo de encoder.

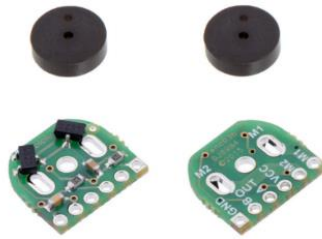


Figura 1.16 Encoder magnético para micromotores de rotor extendido [16].

Características:

- Voltaje de funcionamiento: 2.7 V a 18 V.
- Dos salidas digitales en cuadratura
- Pulsos por revolución: 12
- Peso: 1.0 g

1.3.3.3.2 Encoder óptico para micromotores [17]

Como en el anterior, también es un encoder del tipo incremental diseñado especialmente para micromotores de rotor extendido. A la salida de este se tiene dos señales sinusoidales desfasadas 90° por lo que es necesario usar un dispositivo externo para convertir la señal a digital (Figura 1.17).

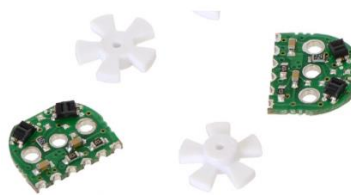


Figura 1.17 Encoder óptico para micromotores de rotor extendido [17].

Características:

- Dos sensores infrarrojos QRD 1114
- Voltaje de funcionamiento: 5 V
- Corriente de entrada: 12 mA
- Dos salidas analógicas desfasadas 90°

- Pulsos por revolución: 12
- Peso: 0.7 g

1.3.3.3 Encoder óptico para rueda 42 x 19 mm [18]

Es un encoder incremental usado específicamente con un tipo de rueda, la rueda de 42 x 19 mm, Para su funcionamiento usa una llanta la cual en su interior tiene ranuras y 2 sensores infrarrojos reflectantes (Figura 1.18).



Figura 1.18 Encoder óptico para rueda 42 x 19 mm [18].

Características:

- Voltaje de funcionamiento: 4.5 V a 5.5 V
- Corriente de entrada: 14 mA a 5 V
- Dos salidas digitales en cuadratura.
- Pulsos por revolución: 48

1.3.4 ROBOT SEGUIDORES DE LÍNEA COMERCIALES

1.3.4.1 Robot 3pi [19]

El Robot 3pi de Pololu ha sido diseñado como robot seguidor de línea y como un resuelve laberintos. El chasis de este robot tiene forma circular, con un diámetro de 9.5 cm y un peso de 83 g sin baterías. Se alimenta al robot con cuatro pilas AAA, los actuadores del robot son dos micromotores con caja reductora 30:1, en la Figura 1.19 se muestra al Robot 3pi. Este robot puede alcanzar velocidades de hasta 1 m/s.



Figura 1.19 Robot 3pi [19].

El cerebro del robot es un microcontrolador ATmega 328P que funciona a una frecuencia de 20 Mhz. Para programarlo se puede utilizar compiladores GNU C/C++, Atmel Studio o también el entorno de desarrollo Arduino.

Para la detección de la línea el robot tiene cinco sensores infrarrojos QTR ubicados en la parte central del robot.

1.3.4.2 Seguidor de línea del club Rutgers IEEE [20]

Este es un kit robótico que trae todos los componentes para armar un robot seguidor de línea, entre otros: una placa de sensores, motores y ruedas Solarbotics GM9; el cerebro de este robot es un Arduino Decimilia. El motor GM9 trae una caja reductora 143:1 con un voltaje de operación de entre 3 y 6 V alcanzando hasta 78 RPM sin carga (Figura 1.20). Para la programación de este prototipo se puede utilizar el entorno de desarrollo Arduino.

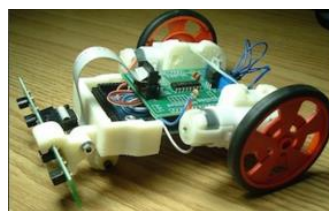


Figura 1.20 Seguidor de línea del club Rutgers IEEE [20].

1.3.4.3 Robomart [21]

Este es un kit robótico para la construcción de un seguidor de línea, tiene todo lo necesario para su construcción, entre otras cosas: dos motores de 300 RPM, el controlador de potencia para estos es el L293D, el cerebro del robot es un Atmega8,

dos sensores infrarrojos y un display LCD de 16x2. Este robot se aprecia en la Figura 1.21.



Figura 1.21 Robomart [21].

1.3.5 ROBOT LABERINTO COMERCIALES

1.3.5.1 Micromouse PICAXE “PICone” [22]

Este robot viene con todos los componentes necesarios para su ensamblaje. Usa sensores Infrarrojos para la detección de paredes. Motores DC con encoder para su movimiento, funciona a 64 MHz. En el kit vienen las instrucciones completas, videos de montaje y un programa básico incluido (Figura 1.22).

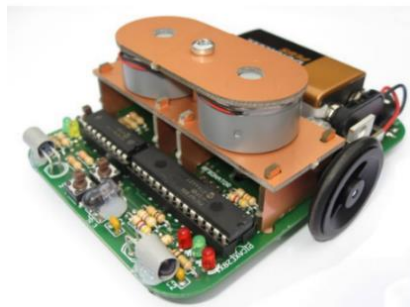


Figura 1.22 Micromouse PICAXE “PICone” [22].

1.4 SELECCIÓN DE SISTEMA DE LOCOMOCIÓN

Se tiene como precedente que las pistas en las cuales los robots se van a desenvolver son completamente planas el tipo de locomoción a usar es por ruedas, por las ventajas que este presenta.

En lo referente al robot laberinto, los principales movimientos de este son en línea recta, curvas un U a la derecha o izquierda, curvas de 90° y giros de 180° sobre el centro de gravedad del robot cuando este entra en un camino sin salida.

El robot seguidor de línea, se desplaza en línea recta y realiza giros en curvas de diferentes radios, el radio más pequeño que se puede encontrar es de 5 cm según el reglamento, todo esto a la mayor velocidad posible.

Teniendo en cuenta el tipo de movimientos que van a hacer los prototipos se escoge la configuración tipo tracción diferencial, debido a que permite realizar giros en curvas de radio pequeño que es lo que se necesita en ambos robots y es la única que permite realizar un giro sobre el centro de gravedad, el cual se necesita en el robot laberinto.

1.4.1 MODELO CINEMÁTICO DE LOS ROBOTS A IMPLEMENTARSE [2]

Para determinar el comportamiento del robot se parte de un modelo cinemático del mismo, en la Figura 1.23 se muestran las variables involucradas con el movimiento del robot.

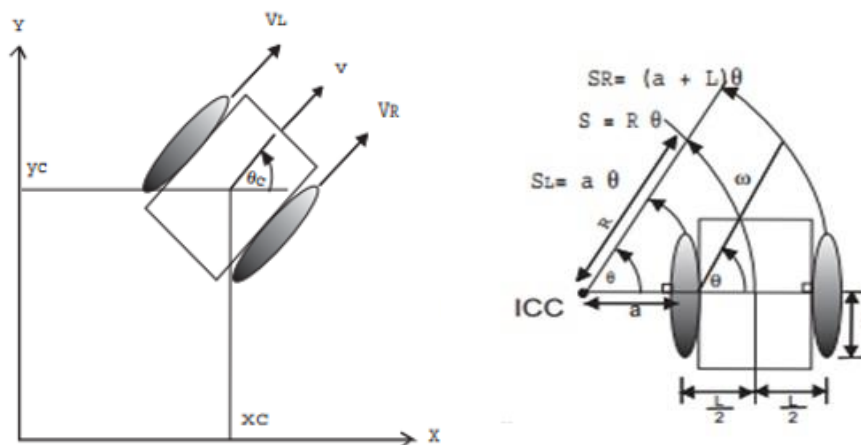


Figura 1.23 Robot móvil de tracción diferencial. Modelo cinemático [23].

θ_c = Ángulo del robot respecto a las coordenadas absolutas

I_{cc} = Centro instantáneo de curvatura

ω = Velocidad angular del robot con respecto a su centro

Ecuación general de la velocidad en una rueda:

$$velocidad = (radio * \omega) \quad (1.1)$$

De la ecuación anterior se calcula las velocidades de las ruedas derecha e izquierda

$$\text{Rueda derecha} \rightarrow V_L = r * \omega_L \text{ y Rueda izquierda} \rightarrow V_R = r * \omega_R \quad (1.2)$$

Se calcula la distancia recorrida por la rueda izquierda, por la rueda derecha y la distancia recorrida por el centro del robot:

$$S_L = a * \theta \quad \text{Distancia recorrida por la rueda izquierda} \quad (1.3)$$

$$S_R = (a + L) * \theta \quad \text{Distancia recorrida por la rueda derecha} \quad (1.4)$$

$$S = R * \theta \quad \text{Distancia recorrida por el centro del robot} \quad (1.5)$$

R es el radio de curvatura instantáneo del robot.

La ecuación (1.3) se reemplaza en la ecuación (1.4):

$$S_R = S_L + L * \theta \quad (1.6)$$

Para obtener ω se deriva la ecuación (1.6):

$$V_R = V_L + L * \omega \quad (1.7)$$

Se despeja ω de la ecuación anterior:

$$\omega = \frac{V_R - V_L}{L} \quad (1.8)$$

Se suma las ecuaciones (1.3) + (1.4) y se obtiene:

$$S_L + S_R = a * \theta + a * \theta + L * \theta = 2 * a * \theta + L * \theta$$

$$S_L + S_R = 2 * \theta * \left(a + \frac{L}{2} \right) \quad (1.9)$$

De la Figura 1.26:

$$R = a + \frac{L}{2} \quad (1.10)$$

Ahora al reemplazar (1.5) en la ecuación (1.9) se tiene:

$$S_L + S_R = 2 * \theta * R \quad \rightarrow S_L + S_R = 2 * S$$

$$S = \frac{S_L + S_R}{2} \quad (1.11)$$

Derivando la ecuación anterior

$$v = \frac{V_L + V_R}{2} \quad (1.12)$$

Cálculo de R:

Derivando la ecuación (1.5)

$$v = \omega * R \quad (1.13)$$

Ahora se reemplaza las ecuaciones (1.12) y (1.8) en la anterior:

$$\frac{V_L + V_R}{2} = \frac{V_R - V_L}{2} * R$$

$$R = \frac{L}{2} * \left(\frac{V_L + V_R}{V_R - V_L} \right) \quad (1.14)$$

Analizando la última ecuación, cuando las velocidades de las ruedas izquierda y derecha son iguales, R tiende al infinito, es decir que el robot va en línea recta. Si $V_R = -V_L$ el radio es cero por lo que el robot girará alrededor de su centro.

A partir de las ecuaciones (1.12) y (1.8):

$$s = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{V_L + V_R}{2} \\ \frac{V_R - V_L}{L} \end{bmatrix} \rightarrow \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{L} & \frac{1}{L} \end{bmatrix} \begin{bmatrix} V_L \\ V_R \end{bmatrix} \quad (1.15)$$

Al multiplicar el Jacobino $J(\theta)$ con s se obtiene:

$$\dot{P}_{(s)} = \begin{bmatrix} \dot{X}_c \\ \dot{Y}_c \\ \dot{\theta}_c \end{bmatrix} = \begin{bmatrix} \cos(\theta_c) & 0 \\ \text{sen}(\theta_c) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = J(\theta_c)s \quad (1.16)$$

Para obtener la posición se debe integrar la ecuación (1.16):

$$X_c = \int_0^t v(t) \cos(\theta_c) dT \quad (1.17)$$

$$Y_c = \int_0^t v(t) \text{sen}(\theta_c) dT \quad (1.18)$$

$$\theta_c = \int_0^t \omega(t) dT \quad (1.19)$$

Y el centro Instantáneo de curvatura (ICC) está localizado en:

$$I_{cc} = ((X_c - R * \text{sen}(\theta_c), Y_c + R * \cos(\theta_c))$$

1.5 COMPONENTES UTILIZADOS PARA LOS ROBOTS A IMPLEMENTARSE

Aunque el reglamento del Concurso Ecuatoriano de Robótica permite el uso de robots comerciales en el caso de la categoría seguidor de línea velocidad en éste trabajo los prototipos se los han diseñado de cero y solo se han adquirido los componentes necesarios.

1.5.1 COMPONENTES DEL SEGUIDOR DE LÍNEA VELOCIDAD

1.5.1.1 Sensor de reflectancia analógico (QTR-8A) [11]

Para la detección de la línea en la pista se usa una matriz de sensores QTR- 8A, la cual viene con 8 pares led-fototransistor, la separación entre sensores es de 9.52 mm. Proporciona 8 salidas analógicas con las cuales se detecta la posición de la línea. En la Figura 1.24 se muestra este tipo de sensor.

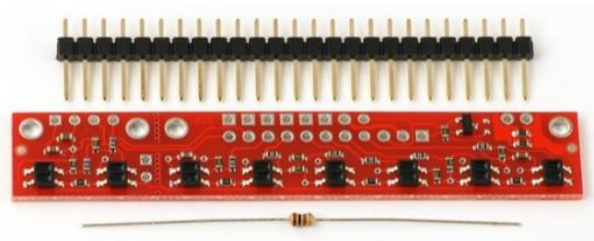


Figura 1.24 Sensores de reflectancia analógico QTR – 8A [11].

Características:

- Dimensión: 70 x 15 mm
- Consumo de corriente: 100 mA
- Formato de salida: 8 tensiones analógicas
- Rango de voltaje de salida: 0 V a la tensión suministrada
- Distancia óptima de detección: 3 mm
- Máxima distancia recomendada de detección: 6 mm
- Peso sin pines: 3.09 g

Pines de conexión:

- 8 salidas analógicas
- Led on (Activación de los sensores)
- Vcc
- Gnd

1.5.1.2 Micromotor 10:1 HP de rotor extendido [9]

Es un motorreductor metálico de corriente continua con escobillas, en el cual viene incorporada una caja reductora con relación 10:1, sus dimensiones son de 10x12x26 mm, la caja reductora de velocidad tiene una longitud de 9.2 mm y el eje al cual se debe acoplar la rueda tiene forma de D.

En la parte posterior del motor tiene una extensión del eje con una longitud de 4.5 mm y 1 mm de diámetro para la colocación de algún tipo de encoder. La tensión nominal de alimentación es de 6 V aunque también puede trabajar con voltajes mayores teniendo en cuenta de que con tensiones superiores a 9 V se reduce significativamente la vida útil de motor. En la Figura 1.25 se puede observar el micromotor de rotor extendido con el encoder.



Figura 1.25 Micromotor 10:1 HP de rotor extendido [9].

Características:

- Voltaje nominal: 6 V
- Velocidad: 3000 RPM
- Corriente en vacío: 120 mA
- Máxima corriente: 1600 mA
- Torque: 0.28 kg-cm

- Relación de la caja reductora 10:1
- Peso: 9.5 g

1.5.1.3 Encoder magnético de cuadratura [16]

Como ya se dijo este es un encoder de tipo incremental, utiliza un disco magnético de seis polos y dos sensores de efecto de Hall. A la salida se tiene dos señales digitales en cuadratura con lo cual se obtiene 12 pulsos por revolución del eje cuando se lee ambos bordes de los dos canales. La tensión de alimentación esta entre 2.7 y 18 V. En la Figura 1.26 se puede apreciar este tipo de encoder.



Figura 1.26 Encoder magnético de cuadratura [16].

1.5.2 COMPONENTES UTILIZADOS PARA EL ROBOT LABERINTO

1.5.2.1 Sensores infrarrojos de 5 mm

Para la medición de distancia se usa un sensor infrarrojo de 5 mm, el cual está compuesto por un emisor y un receptor. El circuito de acondicionamiento para éste se ha diseñado en el capítulo 2.

El emisor es un led infrarrojo que trabaja con voltaje de 1.7 V con un consumo de corriente de 100 mA como máximo.

El receptor es un fototransistor que conduce una cantidad de corriente eléctrica proporcional a la cantidad de luz que incide. El voltaje máximo inverso es de 1.3 V, este tiene un ángulo de visión de 20 grados [24]. En la Figura 1.27 se puede apreciar al sensor.

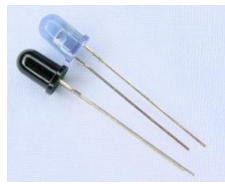


Figura 1.27 Receptor y emisor infrarrojo 5 mm [24].

1.5.2.2 Micromotor 50:1 HP

Para el robot laberinto se usa un motor de menor velocidad, la caja reductora de velocidad tiene una relación 50:1; este motor tiene las mismas características que el micromotor 10:1 HP, con la excepción de que este no trae una extensión del eje por la parte posterior [25] (Figura 1.28).



Figura 1.28 Micromotor 50:1 HP [25].

Características [25]:

- Voltaje nominal: 6 V
- Velocidad: 625 RPM
- Corriente en vacío: 120 mA
- Máxima corriente: 1600 mA
- Torque: 1.1 kg-cm
- Relación de la caja reductora 50:1
- Tamaño: 10 x 12 x 26 mm
- Peso: 9.5 g

1.5.2.3 Encoders óptico de cuadratura [18]

Es un encoder incremental que se usa con una rueda específica como se indicó en 1.3.3.3.3, para obtener los 48 pulsos por revolución en la rueda se tiene 12 ranuras y utiliza dos sensores infrarrojos de reflexión con los cuales se detectan las zonas

blancas (cuando se da reflexión) y oscuras (cuando no hay reflexión o es mínima) midiendo así el moviendo de las ranuras que se encuentran en la rueda. Los sensores están separados para a la salida tener dos señales cuadradas separadas entre sí 90 grados. La alimentación del encoder puede estar entre 4.5 y 5.5 V y consume 14 mA a 5 V. En la Figura 1.29 se muestra este tipo de encoder.



Figura 1.29 Encoder óptico de cuadratura [18].

CAPÍTULO 2

DISEÑO E IMPLEMENTACIÓN DE LOS ROBOTS

2.1 INTRODUCCIÓN

Este capítulo se presentan detalles de la construcción de los robots, para lo cual, lo primero que se hace es un análisis de la parte mecánica, tomando en cuenta las características que cada robot necesita para su correcto funcionamiento. Para el diseño se considera forma, dimensiones, peso, ubicación de sensores y motores; además se muestra la importancia de la distribución de los elementos en el chasis y cómo estos pueden afectar en el desempeño de los prototipos.

A continuación se detalla el diseño electrónico de: sensores, controladores de potencia, sistema de control y periféricos para interactuar con el robot. Finalmente se presentan los robots completamente diseñados.

2.2 ARQUITECTURA DE HARDWARE

El hardware es el conjunto de dispositivos que hacen posible el funcionamiento de un robot, este abarca todos los componentes mecánicos y electrónicos. Para los prototipos de este trabajo se necesita básicamente de un chasis, motores y electrónica; en esta última, los dos robots constan de los mismos componentes: sistema sensorial, sistema de control, alimentación y actuadores como se puede apreciar en la Figura 2.1 . En el diseño electrónico de cada prototipo se ve con más detalle estos puntos.

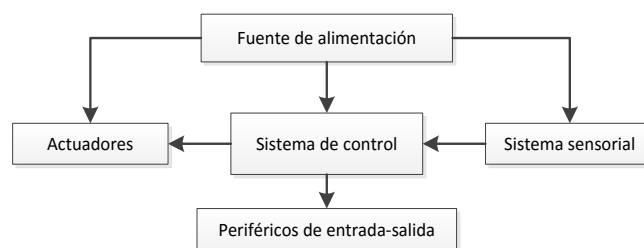


Figura 2.1 Arquitectura general de los robots.

2.2.1 DISEÑO MECÁNICO DEL ROBOT SEGUIDOR DE LÍNEA

Como se expuso en el capítulo anterior, se usa el sistema de locomoción diferencial para el diseño del robot por las ventajas que este proporciona. En la parte mecánica se considera los aspectos como: chasis, peso, ruedas, etc. A continuación se trata cada punto.

2.2.1.1 Chasis

El diseño del chasis es una parte muy importante en la construcción del robot, ya que se debe tener en cuenta: forma, tamaño, peso, distribución de elementos, distancia entre ruedas, distancia entre el eje de las ruedas y sensores, altura del chasis con respecto al suelo y materiales de fabricación, ya que todos estos influyen en el funcionamiento del robot.

2.2.1.1.1 Forma

Se tiene dos consideraciones al diseñar el chasis: diseñarlo de un material liviano y resistente y que este sirva de soporte para todos los elementos del robot (motores, sensores, batería, etc.) o que la placa que contiene los circuitos electrónicos sirva como chasis. Se escoge la segunda opción ya que de esta manera se disminuye el peso.

Teniendo en cuenta el objetivo de este robot, pasar la pista en el menor tiempo posible, es importante escoger la forma del chasis la cual va a estar determinada por la distribución de los elementos a los cuales brinda soporte; ya que una mala distribución hará que el robot pierda tracción y derrape al tomar una curva con rapidez, es por esto que se debe tomar en cuenta que el centro de gravedad de este quede cerca al centro del eje que une las ruedas.

Se considera los elementos más grandes y pesados para su correcta ubicación en el chasis; el elemento más pesado es la batería (aproximadamente 17 g y dimensiones de 47 x 17 x 12 mm) por lo que es necesario ponerla longitudinalmente al eje de las ruedas, como se aprecia en la Figura 2.2; toda la electrónica necesaria para el funcionamiento del robot, excepto los sensores, se la ubicó por debajo o a

los lados de la batería. Los sensores para la detección de la línea, se encuentran ubicados a cierta distancia del chasis soportado por un brazo.

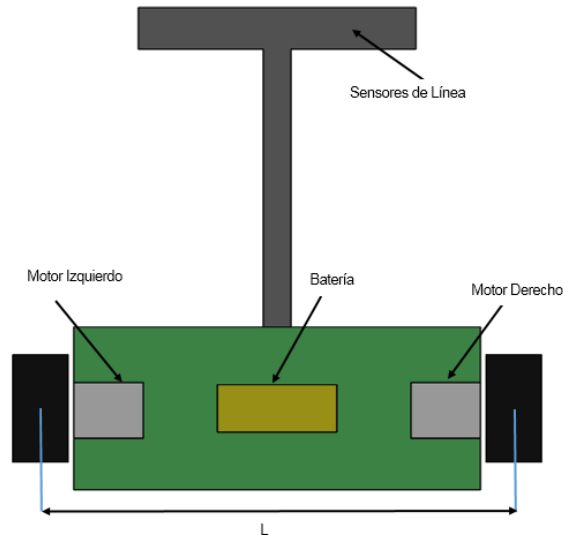


Figura 2.2 Chasis del robot seguidor y ubicación de elementos.

2.2.1.1.2 Dimensiones

Las dimensiones a considerar en el diseño del robot son la distancia entre las ruedas y la distancia entre eje de las ruedas y los sensores. Con respecto a la distancia entre ruedas, se puede observar en la ecuación (2.1) del modelo cinemático que ésta no debe ser muy grande, ya que el robot tarda más al girar y se necesitaría que las variaciones de las velocidades de las ruedas sean mayores.

$$\omega = \frac{V_R - V_L}{L} \quad (2.1)$$

ω = Velocidad angular del robot.

V_L = Velocidad de la rueda derecha.

V_L = Velocidad de la rueda izquierda.

L = Distancia entre las ruedas.

Otro aspecto importante con respecto a la distancia entre ruedas es el mínimo radio de curvatura que el robot puede tomar sin invertir el sentido de una rueda; este está determinado por la distancia entre las ruedas y es igual a $\frac{L}{2}$; esto se puede observar en (2.2) del modelo cinemático visto en la sección 1.4.1.

$$R = \frac{L}{2} * \left(\frac{V_L + V_R}{V_R - V_L} \right) \quad (2.2)$$

Teniendo en cuenta este y el punto anterior se diseñó el chasis de tal forma que la distancia entre las ruedas sea pequeña (se dejó únicamente el espacio suficiente para los motores y batería) con una distancia entre las ruedas de 12 cm, como se puede ver en la Figura 2.3. Así el robot será capaz de tomar curvas de hasta 6 cm de radio sin tener que invertir el giro de una llanta.

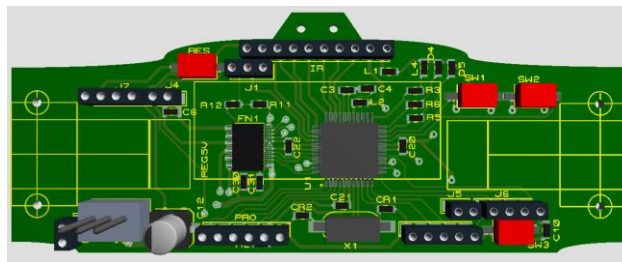


Figura 2.3 Chasis del robot seguidor de línea velocidad.

El siguiente punto importante de diseño es la distancia entre los sensores de línea y el eje de las ruedas, generalmente se une la placa de control con los sensores usando un brazo (Figura 2.4), se lo hace así ya que se necesita que haya una cierta distancia entre los sensores y la tracción para que el robot pueda ver con cierta anticipación la curva. El problema es el largo que debe tener el brazo, este depende de la pista, si es largo le permitirá al robot anticiparse a las curvas de mejor manera pero si las curvas son muy cerradas el robot no va a poder tomar la curva correctamente; por lo que si las curvas son cerradas el brazo deberá ser más corto; se realizó algunas pruebas en pista y la distancia con la que mejores resultados se tiene es de 15 cm. Teniendo en cuenta a la inercia rotacional, el brazo como los sensores de línea deben ser livianos para que el robot no tenga problemas en los giros.



Figura 2.4 Sensores y soporte de robot seguidor de línea.

2.2.1.2 Disposición de sensores

Hay varias formas de distribuir los sensores para la detección de la línea, las más usadas son: configuración en línea y configuración en V, pero la configuración en V es utilizada en robots seguidores rastreadores y la configuración en línea es usada para velocistas [26].

Para el seguimiento de la línea se lo podría hacer únicamente con 2 sensores aunque el desempeño no sería bueno, por lo que es necesario tener más sensores para poder determinar de forma más exacta la posición del robot con respecto a la línea, pero tampoco hay que irse al otro extremo ya que demasiada cantidad de sensores va hacer muy pesada la parte delantera con lo cual aumenta la inercia rotacional; en éste trabajo se realizaron pruebas con 4 y 6 sensores; con cuatro sensores el rango de detección de la línea fue muy pequeño (aproximadamente 14°) y no se pudo alcanzar el objetivo del tiempo, con 6 sensores ya se logró alcanzar este objetivo pero se quiso dar un rango de detección un poco más amplio y se dejó el diseño final con 8 sensores (rango de detección 28°).

2.2.1.3 Motores y ruedas

Los motores son una parte fundamental del robot, ya que de estos depende la velocidad con la que puede desplazarse, aquí se debe considerar el peso del robot y el radio de la ruedas para su correcta selección.

El radio de las ruedas influye significativamente en el desempeño del robot, ya que la velocidad máxima y el torque necesario para mover al robot están condicionados

por esta variable; si el radio es grande el desplazamiento del robot será mayor por cada revolución de los motores, por lo que la velocidad será mayor pero en contraparte se necesita un torque mayor para mover al robot. Si el radio es pequeño la velocidad máxima que el robot puede alcanzar va a ser menor pero también el torque necesario va a ser menor.

Para calcular el torque del motor requerido según [27] se necesita la masa total del robot, el radio de la rueda y la aceleración que se desee, para lo cual se aplica (2.3)

$$T = M * (a + g * \sin(\theta)) * r \quad (2.3)$$

Donde:

T : Torque del motor.

M : Masa total del robot.

a : Aceleración.

θ : Ángulo del plano.

g : Gravedad.

r : Radio de las ruedas.

Como datos se tiene, que la masa del robot es 170 g, se desea manejar una aceleración de 2 m/s^2 , el radio de la rueda es de 1 cm y el ángulo θ es cero ya que la pista es completamente plana. Usando (2.3) se tiene:

$$170 \text{ g} * \frac{1 \text{ kg}}{1000 \text{ g}} = 0.17 \text{ kg}$$

$$T = 0.17 \text{ kg} * 2 \text{ m/s}^2 * 0.01 \text{ m} = 0.0034 \text{ Nm}$$

$$T = 0.0034 \text{ Nm} * \frac{0.225 \text{ lb}}{1 \text{ N}} * \frac{16 \text{ oz}}{1 \text{ lb}} * \frac{100 \text{ cm}}{1 \text{ m}} * \frac{1 \text{ in}}{2.54 \text{ cm}}$$

$$T = 0.48 \text{ oz} - \text{in}$$

Existen varios motores que cubrirían este torque, teniendo en cuenta el catálogo de motores y que se necesita alcanzar altas velocidades, se usará el micromotor HP 10:1 con un torque de 4 oz-in y velocidad de 3000 RPM.

Entonces la velocidad máxima que se podría alcanzar se calcula con (2.4):

$$V = \frac{V \text{ RPM}}{60} * 2 * \pi * r \quad (2.4)$$

$$V = \frac{3000}{60} * 2 * \pi * r = 3.14 \text{ m/s}$$

Dando como resultado una velocidad muy elevada, por lo general no se alcanzan velocidades tan altas en las competencias ya que las pistas suelen tener curvas de radios pequeños y los robots no pueden desarrollar sus máximas velocidades. El motor al cual se hace referencia viene con una caja reductora de velocidad con relación 10:1.

A la salida de la caja reductora se tiene un eje en forma de D al cual se debe acoplar la rueda. Anteriormente se dijo que se iba a diseñar una rueda con un radio de 1 cm; en esta medida está considerado el rin de la rueda como su recubrimiento para así obtener mayor adherencia sobre la pista. Para la rueda se probó algunos materiales como caucho, silicona y espuma. En la Figura 2.5 se puede apreciar la rueda y el motor.

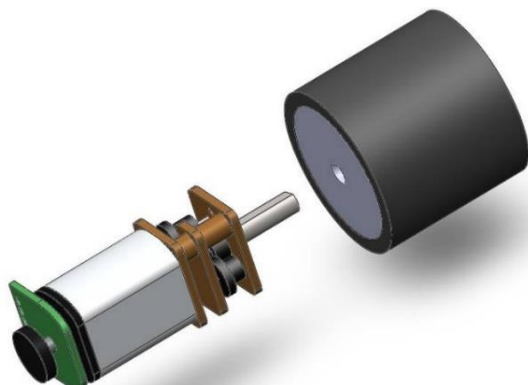


Figura 2.5 Motor y rueda.

2.2.2 DISEÑO MECÁNICO DEL ROBOT LABERINTO

Al igual que el anterior robot se usa el sistema de locomoción diferencial para su diseño, por lo que se diseña la parte estructural para que de soporte a todos los componentes del robot.

2.2.2.1 Chasis

El chasis del robot soporta todos los elementos que este posee, como: sensores, motores, batería, etc. Es importante realizar un buen diseño del mismo ya que por ejemplo, si es desproporcionado se podría tener problemas con los giros dentro del laberinto. Se considera para el diseño: forma, dimensiones, distancia entre las ruedas, distribución de la electrónica y ubicación de los sensores de pared.

2.2.2.1.1 Forma

La forma del chasis va a estar determinada por los elementos que éste va a albergar y por su distribución, debe considerarse que el centro de gravedad del robot este cerca al eje que une las ruedas. Hay varias formas que puede tomar el chasis y cumplir con lo dicho anteriormente. Para este robot se tomó la decisión de hacerlo redondo, ya que con ésta forma puede girar más libremente y la posibilidad de quedarse atascado con una pared en caso de choque es menor. En la Figura 2.6 se puede apreciar lo anteriormente dicho; se compara un chasis redondo con uno de forma cuadrada.

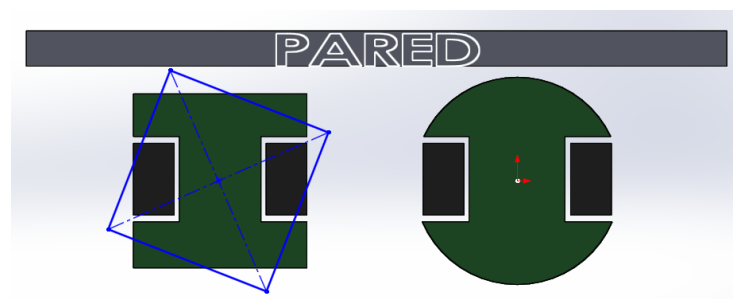


Figura 2.6 Influencia del chasis sobre el comportamiento del robot laberinto.

Para que el robot sea equilibrado y cumpla con lo ya dicho, el eje que une las ruedas pasa por el centro del chasis, ya que si esto no es así el robot al girar en su propio eje describirá una circunferencia de mayor tamaño aumentando las posibilidades de choque. En la Figura 2.7 se puede observar un ejemplo de lo que ocurriría si las ruedas se localizaran en la parte trasera del robot.

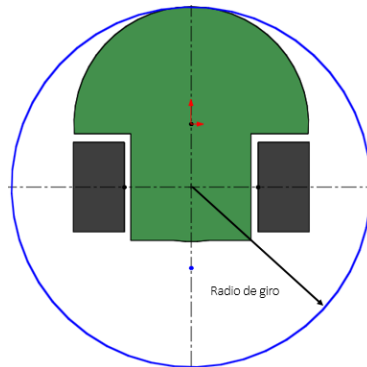


Figura 2.7 Robot laberinto con ruedas en la parte trasera.

También se tiene en cuenta el alojamiento de la batería (elemento más pesado del robot); es importante que la batería vaya en el centro, en medio de los motores, ya que esto ayudará a que el robot tenga mayor tracción. Al usar este chasis y tracción antes mencionado, se debe usar dos ruedas de soporte para mantener equilibrado al robot horizontalmente. En la Figura 2.8 se muestra todo lo antes dicho.

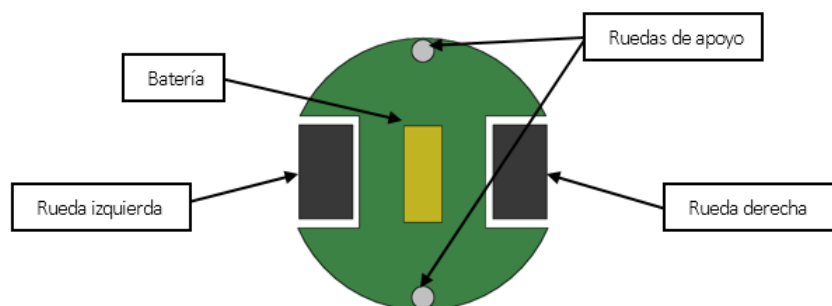


Figura 2.8 Robot laberinto distribución de ruedas y batería.

2.2.2.1.2 Dimensiones

Las dimensiones estarán limitadas por los reglamentos del CER el cual dice que debe ser menor a 15 x 15 cm con una altura no mayor a 20 cm. Por esta razón se optara por diseñar un robot pequeño para que pueda moverse libremente dentro del laberinto, especialmente en los giros de 180 grados y en U.

Que el robot sea pequeño también ayuda a que realice giros con menor probabilidad de choque, ya que el mínimo radio que puede girar éste sin tener que invertir el giro de una rueda es menor. Esto se puede apreciar en la ecuación (2.5) del modelo cinemático visto en la sección 1.4.1.

$$R = \frac{L}{2} * \left(\frac{V_L + V_R}{V_R - V_L} \right) \quad (2.5)$$

Es por esto que el chasis del robot no supera una circunferencia de 5 cm de radio, la distancia entre las ruedas es de 8 cm y la altura del robot es de 4 cm.

2.2.2.2 Disposición de sensores

Para este tipo de robots y dependiendo del tipo de algoritmo que se vaya a implementar solo sería necesario utilizar dos sensores de pared, uno para la pared frontal y el otro para detectar una pared lateral, pero este no es el caso, aquí se va implementar varios algoritmos así que se utilizó 4 sensores: dos para la detección de la pared frontal y dos para la detección de las paredes laterales.

En lo que se refiere a los sensores laterales la posición más intuitiva es ponerlos perpendicular a la pared lateral como se presenta en la Figura 2.9 (a); pero no es la mejor opción, porque se dificulta la implementación del control de posición que sirve para evitar chocar con las paredes. Al usar los sensores de esta forma no se puede discernir si el robot se está alejando o acercando a la pared, como se muestra en la Figura 2.9 (b) y en la Figura 2.9 (c) ya que estas lecturas del sensor dan un valor más alto del que debería ser en ambos casos.

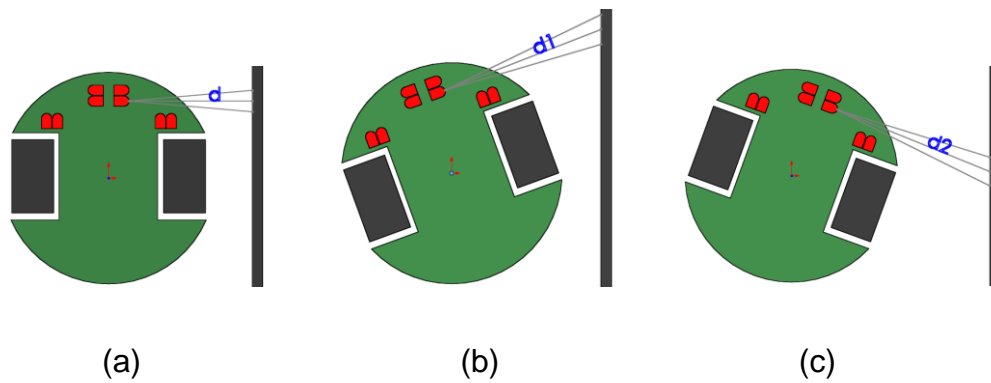


Figura 2.9 Disposición de sensores ortogonalmente. (a) Robot paralelo a la pared. (b) Robot con un ángulo de inclinación de 25° a la izquierda. (c) Robot con un ángulo de inclinación de 25° a la derecha.

Otra opción es colocar los sensores con un cierto ángulo como se puede ver en la Figura 2.10 (a), con esta configuración se conoce si el robot se está alejando o acercando a la pared, ya que al estar desviado como en la Figura 2.10 (b) proporciona un valor mayor del que debería ser y al estar desviado como en la Figura 2.10 (c) entrega un valor menor. Esta posición de los sensores permite un mayor rango de detección de la distancia. Otra ventaja que ofrece es que se puede prever si hay un camino libre a la derecha o a la izquierda con anticipación.

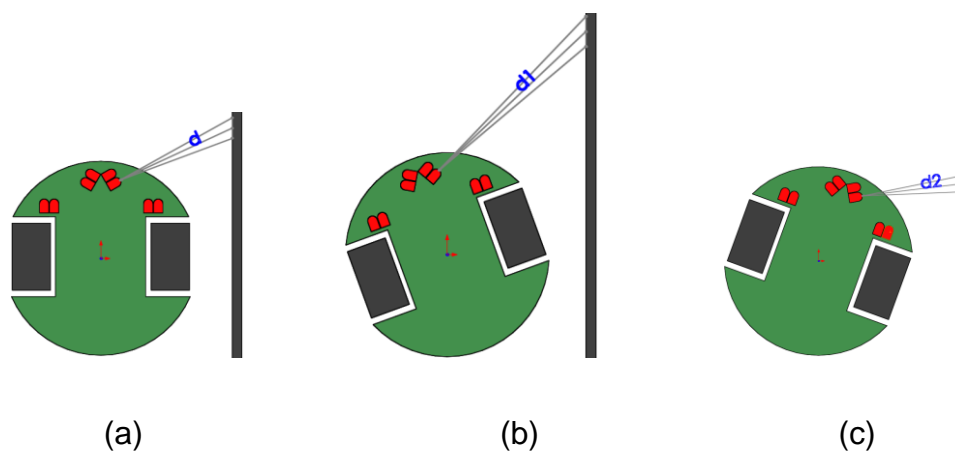


Figura 2.10 Disposición de sensores con un ángulo de 30° . (a) Robot paralelo a la pared. (b) Robot con un ángulo de inclinación de 25° a la izquierda. (c) Robot con un ángulo de inclinación de 25° a la derecha.

Para la selección del ángulo (α) de los sensores laterales (Figura 2.11), se tiene en cuenta las dimensiones de una celda del laberinto que mide 25 x 25 cm, se debe poner el eje de los motores en el centro de la celda y apuntar los sensores laterales justo al final de ésta, de esta manera se puede detectar con anticipación si existe un camino libre [28] y la distancia suficiente para poder realizar el giro. Para el robot, la medida de éste ángulo es de 34° .

Los sensores frontales son orientados perpendicularmente a la pared frontal, ya que con estos se mide la distancia que existe entre el robot y la pared. Se usa dos sensores ya que así se puede medir de mejor manera la distancia entre el robot y la pared evitando así que el robot se acerque demasiado a la pared frontal en caso de llegar con cierta desviación.

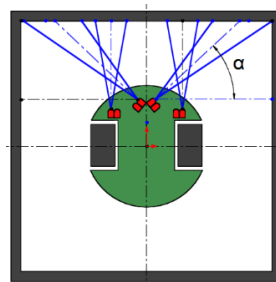


Figura 2.11 Disposición de los sensores laterales y frontales.

2.2.2.3 Motores y ruedas

Se realiza los cálculos necesarios para escoger el motor más adecuado para el robot laberinto. Como datos se tiene, que la masa del robot es 200 g, y se desea manejar una aceleración de 1 m/s^2 , el radio de la rueda es de 2 cm. Usando (2.3) se tiene:

$$200 \text{ g} * \frac{1 \text{ kg}}{1000 \text{ g}} = 0.2 \text{ kg}$$

$$T = 0.2 \text{ kg} * (1 \text{ m/s}^2) * 0.021 \text{ m} = 0.004 \text{ Nm}$$

$$T = 0.004 \text{ Nm} * \frac{0.225 \text{ lb}}{1 \text{ N}} * \frac{16 \text{ oz}}{1 \text{ lb}} * \frac{100 \text{ cm}}{1 \text{ m}} * \frac{1 \text{ in}}{2.54 \text{ cm}}$$

$$T = 0.5 \text{ oz} - \text{in}$$

Existen varios motores que pueden proporcionar este torque, se usa un micromotor HP 50:1 con un torque de 15 oz-in de 625 RPM. Entonces la velocidad máxima que se podría alcanzar se calcula con (2.4).

$$V = \frac{625}{60} * 2 * \pi * r = 1.37 \text{ m/s}^2$$

2.3 DISEÑO ELECTRÓNICO DEL ROBOT SEGUIDOR DE LÍNEA

2.3.1 ARQUITECTURA

El robot seguidor de línea está conformado de varios sistemas, los cuales se han pensado para que el robot tenga un buen desempeño, para que sea fácil su interacción con él y que la implementación de nuevos algoritmos como su depuración se la pueda hacer de manera eficiente. El robot, como se lo puede ver en la Figura 2.12, está compuesto de un sistema de control, de un sistema sensorial en el cual están los encoders, sensores de línea y medición de batería, un sistema de periféricos de entrada, salida y un controlador de potencia con el cual se accionan los motores.

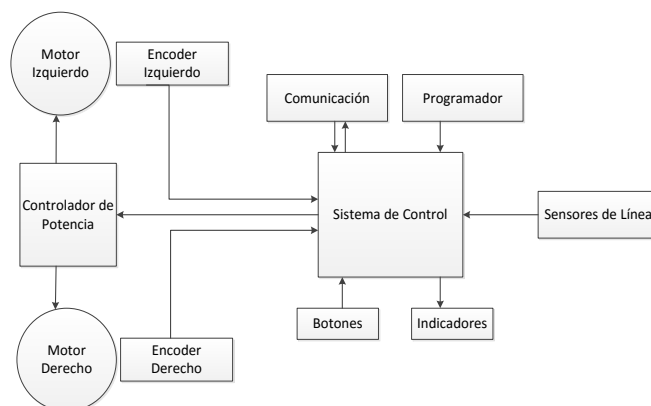


Figura 2.12 Arquitectura del robot seguidor velocidad.

En lo que se refiere al sistema de alimentación, se ha utilizado una batería de litio polímero, con un voltaje de 8.3 V, ésta alimenta a todos los sistemas anteriormente mencionados como se puede ver en la Figura 2.13.

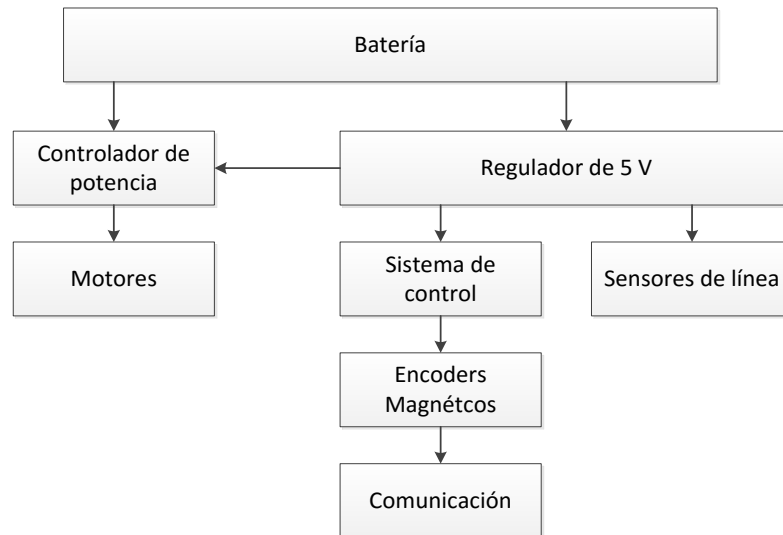


Figura 2.13 Alimentación de energía.

2.3.2 SISTEMA DE CONTROL

En la implementación de los robots se usó el microcontrolador ATmega 324PA (Figura 2.14), éste es un microcontrolador de 8 bits de alto rendimiento y bajo consumo de energía. El voltaje de alimentación de éste está entre 2.7 a 5.5 V, tiene 32 Kbyte de memoria de programa, 1 Kbyte de memoria EEPROM, y 2 Kbyte de memoria SRAM; a continuación se nombra los periféricos que este posee [29]:

- ✓ Dos temporizadores de 8 bits con pre escalamiento separado y modo de comparación.
- ✓ Un temporizador de 16 bits con pre escalamiento separado, modo de comparación y modo de captura.
- ✓ Contador en tiempo real con Oscilador separado.
- ✓ 6 Canales PWM.
- ✓ 8 Canales ADC de 10 bits, Modo diferencial con ganancia seleccionable x1, x10 o x 200.
- ✓ Dos puertos seriales USART programables.
- ✓ Interfaz serial SPI maestro – esclavo.

- ✓ Watchdog Timer programable con oscilador independiente.
- ✓ Comparador analógico en el mismo chip.

La configuración de pines se muestra en la Figura 2.14:

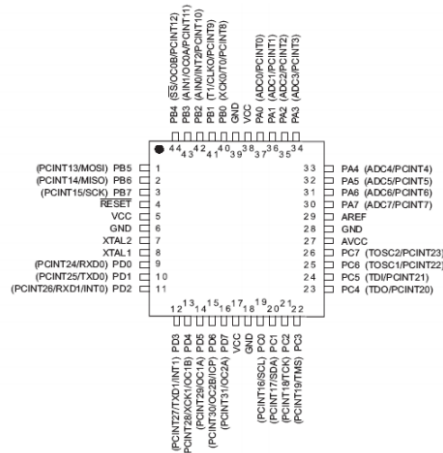


Figura 2.14 Distribución de pines ATMEGA 324PA TQFP [29].

2.3.3 SISTEMA SENSORIAL

2.3.3.1 Sensores de línea [11]

La detección de la línea se la realiza por medio de sensores infrarrojos, se utiliza la placa de sensores QTR-8A, que está especialmente diseñada como un sensor de línea pero también puede funcionar como un sensor de proximidad.

El módulo trae 8 pares emisor receptor espaciados entre si 9.525 mm; la salida de cada sensor (fototransistor) está conectado a una resistencia de pull-up para obtener un voltaje analógico que está entre 0 y 5 V.

En lo que respecta a los emisores, están colocados en pares para reducir el consumo de corriente, los leds son controlados por un MOSFET, es decir que los emisores se los enciende o se los apaga por medio de la compuerta del MOSFET. Para limitar la corriente en los emisores se coloca una resistencia para que circule 25 mA por lo que el consumo total de la placa es de alrededor de 100 mA.

La separación de los sensores se la hace de tal manera que solo dos sensores puedan estar sobre la pista (cinta aislante negra de 20 mm).

2.3.3.2 Encoder magnético

Se usó un par de encoders magnéticos, uno para cada rueda, estos vienen diseñados para que se acoplen perfectamente al motor por la parte posterior, a estos sensores se lo debe alimentar con un voltaje entre 2.7 a 18 V, estos tienen dos salidas digitales en cuadratura, en 0L el voltaje de la salida será 0 V y en 1L el voltaje será de Vcc (Figura 2.15).

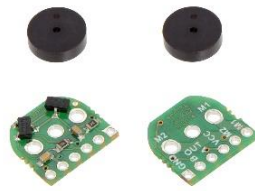


Figura 2.15 Encoders magnéticos.

2.3.3.3 Medición de batería

Para el buen funcionamiento del robot es necesario que la batería tenga un nivel de voltaje adecuado para que tanto la parte de control como la de potencia funcionen perfectamente. Debido a que el voltaje de la batería excede al voltaje que soporta el microcontrolador, no es posible medirlo directamente, por lo que se implementó un divisor de voltaje (Figura 2.16).

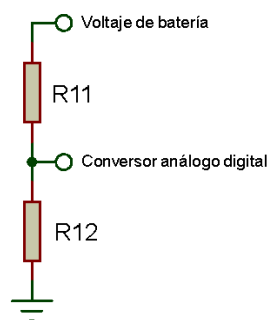


Figura 2.16 Divisor de voltaje para medición de batería.

La salida de voltaje se obtiene por medio de la ecuación (2.6).

$$\text{Salida del divisor} = \frac{R11}{R11 + R12} * \text{Voltaje de batería} \quad (2.6)$$

El voltaje de la batería completamente cargada es 8.3 V, para la medición se redujo el voltaje a niveles que soporta el microcontrolador; se reduce a 4.15 V por lo que las resistencias R11 y R12 deben ser iguales. Para evitar el consumo excesivo de corriente se escoge un valor de 10 k Ω para las resistencias.

2.3.4 CONTROLADOR DE POTENCIA

Se va a utilizar un micromotor de 3000 RPM, revisando las especificaciones de este, se tiene que el máximo consumo de corriente es de 1.6 A a 6 V pero en el robot se lo va a alimentar con un voltaje de 7.5 V por lo que se midió y el consumo de corriente es de 2.1 A máximo. Entonces es necesario una interfaz entre el microcontrolador y el motor para poder controlarlo, después de revisar algunas opciones existentes en el mercado, se optó por el controlador de potencia TB6612FNG, este trae dos puentes H. Algunas de sus especificaciones son [30]:

- Voltaje de motor: 4.5 a 13.5 V.
- Voltaje lógico: 2.7 a 5.5 V.
- Salida de corriente máxima por canal: 3.2 A.
- Salida de corriente continua: 1.2 A.
- Máxima frecuencia de PWM: 100 kHz.

La distribución de pines del TB6612FNG se muestra en la Figura 2.17:

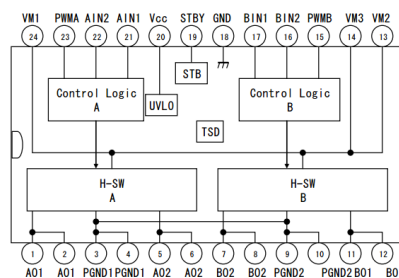


Figura 2.17 Controlador de motores TB6612FNG [30].

Para mayor detalle se adjunta el manual en el Anexo D. En la Tabla 2.1 se indica la descripción de cada pin.

Tabla 2.1 Descripción de los pines del TB6612FNG [30].

Número	Nombre del pin	Descripción
1	AO1	Salida 1 del canal A
2	AO1	
3	PGND1	Tierra voltaje de motor
4	PGND2	
5	AO2	Salida 2 del canal A
6	AO2	
7	BO2	Salida 2 del canal B
8	BO2	
9	PGND2	Tierra voltaje de motor
10	PGND2	
11	BO1	Salida 1 del canal B
12	BO1	
13	VM2	Voltaje de motor
14	VM3	
15	PWMB	Entrada de pwm del canal B
16	BIN2	Entrada 2 del canal B
17	BIN1	Entrada 1 del canal B
18	GND	Tierra voltaje lógico
19	STBY	"L"= standby
20	Vcc	Voltaje lógico
21	AIN1	Entrada 1 del canal A
22	AIN2	Entrada 2 del canal A
23	PWMA	Entrada de pwm del canal A
24	VM1	Voltaje de motor

Dos señales de entrada IN1 e IN2 se usan para controlar el motor en cuatro formas diferentes: giro horario, giro anti horario, freno rápido y freno lento; las dos salidas que posee este circuito integrado se las puede controlar de forma independiente por medio de AIN1, AIN2 y PWMA para la salida A y BIN1, BIN2 y PWMB para la salida B, éste además trae una entrada extra la cual se la puede usar para poner el chip en activo o en standby. En la Tabla 2.2 se indica el funcionamiento del controlador.

Tabla 2.2 Control del integrado TB6612FNG.

Entrada				Salida		
IN1	IN2	PWM	STBY	OUT1	OUT2	Modo
H	H	H/L	H	L	L	Freno rápido
L	H	H	H	L	H	Giro sentido anti horario
		L	H	L	L	Freno rápido
H	L	H	H	H	L	Giro sentido horario
		L	H	L	L	Freno rápido
L	L	H	H	OFF Alta impedancia		Freno lento
H/L	H/L	H/L	L	OFF Alta impedancia		Espera

En la Tabla 2.3 se detalla las conexiones del microcontrolador al integrado TB6612FNG.

Tabla 2.3 Conexiones del microcontrolador al TB6612FNG.

Motor derecho	
Pin del microcontrolador	Pin del controlador de potencia
PD4	AIN1
PD5	AIN2
PB3	PWMA
Motor izquierdo	
PB0	BIN1
PB1	BIN2
PB4	PWMB

Para absorción del ruido es necesario poner capacitores en la alimentación del controlador de potencia (Figura 2.18); en el manual de éste recomienda el uso de 0.1 y 10 (uF) [30]:

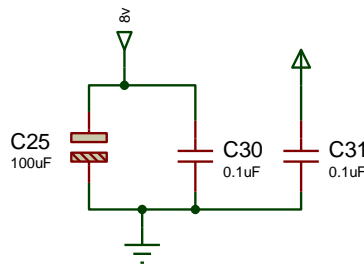


Figura 2.18 Capacitores recomendados para el TB6612FNG.

El inconveniente de trabajar con motores es que estos causan ruido, éste puede afectar a la lectura de sensores o incluso puede afectar el funcionamiento del microcontrolador. La forma más efectiva de eliminar este ruido es con el uso de capacitores, para este motor recomienda el fabricante utilizar capacitores cerámicos de 0.1 uF [31]. En la Figura 2.19 se muestra los capacitores de los motores.

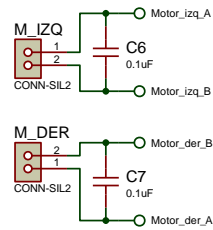


Figura 2.19 Capacitores de los motores en el robot seguidor de línea velocidad.

2.4 DISEÑO ELECTRÓNICO DEL ROBOT LABERINTO

2.4.1 ARQUITECTURA

El robot laberinto está conformado de varios sistemas que interactúan entre sí para un funcionamiento óptimo, este está compuesto de periféricos de entrada y salida, sistema de control, sistema sensorial y sistema de potencia para el control de motores. En la

Figura 2.20 se puede observar todos los sistemas que este posee.

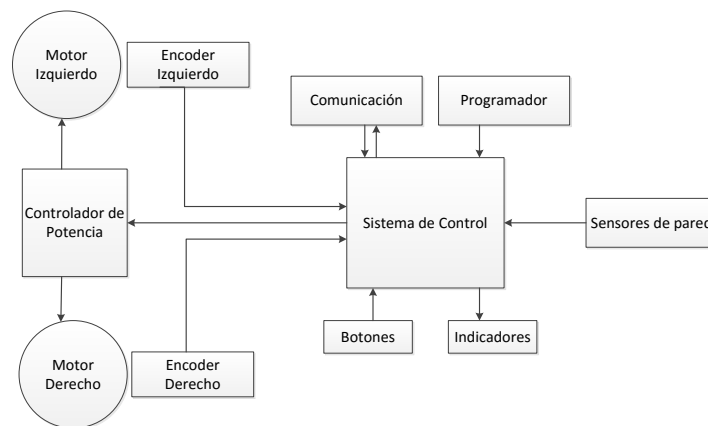


Figura 2.20 Arquitectura del robot laberinto.

En el sistema de alimentación, se utiliza una batería de litio polímero, con un voltaje de 8.3 V, ésta alimenta a los sistemas anteriormente mencionados como se puede observar en la Figura 2.21.

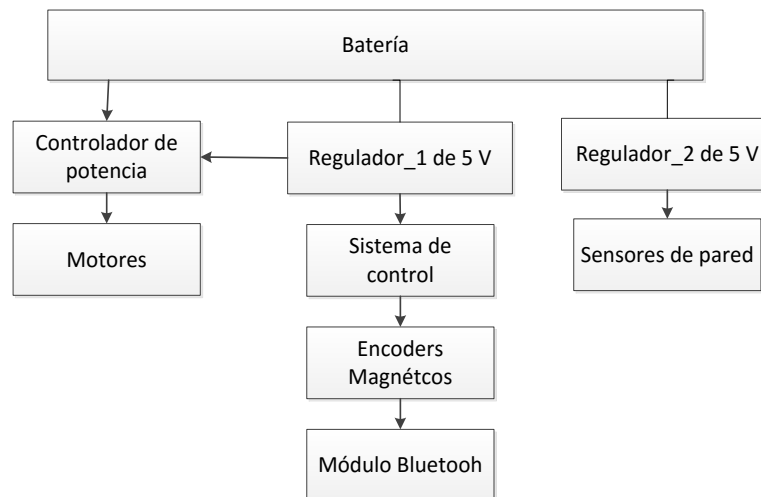


Figura 2.21 Alimentación de energía.

2.4.2 SISTEMA DE CONTROL

Al igual que en el robot seguidor se usa un microcontrolador ATmega 324PA que posee las características ya antes expuestas.

2.4.3 SISTEMA SENSORIAL

2.4.3.1 Sensores de distancia

Para la navegación dentro del laberinto el robot utiliza sensores infrarrojos, los cuales son acondicionados para tener una correcta lectura de distancia en un rango de 3 a 30 cm.

El sensor utilizado está constituido por un led infrarrojo (IR333) como emisor y un fototransistor (PD333) que se lo usa como receptor.

En la Figura 2.22 se muestra el circuito de acondicionamiento del sensor infrarrojo de distancia. Para su diseño primero se debe seleccionar la corriente que pasa a través del diodo infrarrojo (D2), teniendo en cuenta que la máxima corriente que este puede soportar es de 100 mA, se usa un valor de 80 mA.

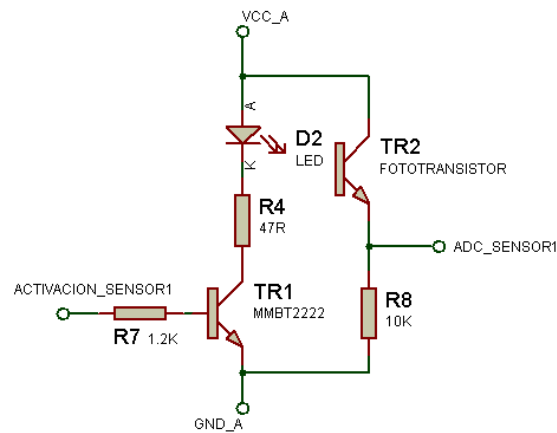


Figura 2.22 Acondicionamiento del sensor infrarrojo.

Donde:

Voltaje de la fuente $V_{CC_A} = 5\text{ V}$.

Voltaje del diodo a 100 mA $V_{D2} = 1.6\text{ V}$.

Corriente a través del diodo $I_{D2} = 80\text{ mA}$.

$$R4 = (V_{CC_A} - V_{D2})/I_{D2}$$

$$R4 = (5\text{ V} - 1.6\text{ V})/80\text{ mA}$$

$$R4 = 42.5\ \Omega \Rightarrow R4 \approx 47\ \Omega$$

Después de dimensionar la resistencia R4, se utiliza un transistor NPN (TR1) que se conecta al pin del microcontrolador a través de R7, permitiendo el control de encendido y apagado del diodo infrarrojo.

Si se conecta el diodo infrarrojo directamente a un pin del microcontrolador este no proporciona la corriente necesaria para su correcto funcionamiento.

Tomando en cuenta que la corriente I_{D2} también circula por TR1 se selecciona un transistor (MMBT2222) el cual soporta hasta una corriente máxima de 150 mA a través de Colector-Emisor. Con los datos del transistor se realiza el cálculo de R7.

Donde:

Voltaje de salida del pin $V_{Pin} = 5 V$.

Voltaje Base – Emisor $V_{BE} = 0.7 V$.

Ganancia de corriente mínima $\beta = 30$.

$$Si I_{D2} = I_C$$

$$R7 = ((V_{Pin} - V_{BE}) * (\beta)) / I_C$$

$$R7 = ((5 V - 0.7 V) * (30)) / 80 mA$$

$$R7 = 1.6 k\Omega \Rightarrow R7 \approx 1.2 k\Omega$$

Finalmente se calcula R8, ésta se encuentra en serie con el foto transistor (TR2); la corriente que circula por TR2 afecta al rango de detección del sensor, se probaron con varias intensidades de corriente para obtener una mejor medida de distancia; con 0.5 mA se obtuvo el rango de detección deseado.

Donde:

Corriente a través de TR2 es $I_{Q2} = 0.5 mA$

$$R8 = V_{CC_A} / I_{Q2}$$

$$R8 = 5 V / 0.5 mA \quad R8 = 10 k\Omega$$

2.4.3.2 Encoder óptico

Este es un encoder incremental el cual sirve para medir la rotación de la llanta y sentido de giro, posee 12 dientes (Figura 2.23 (b)), con lo cual se puede conseguir hasta 48 pulsos por revolución. El circuito está provisto de dos sensores infrarrojos adecuadamente localizados y acondicionados para detectar el blanco y negro de la llanta.

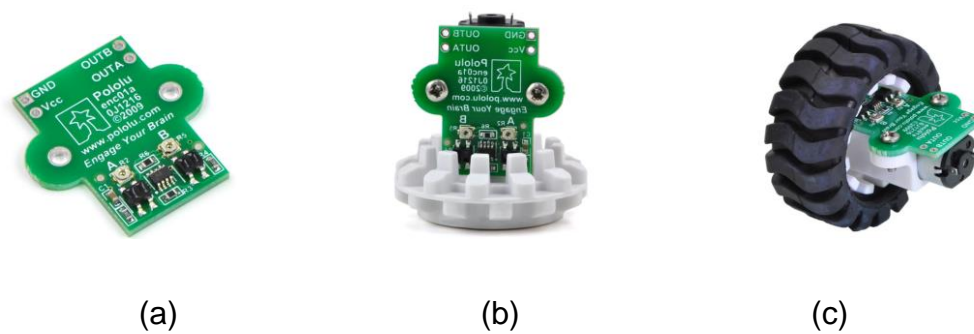


Figura 2.23 Encoder óptico 42 x 19 mm. (a) Placa del encoder y sensores infrarrojos. (b) Encoder montado sobre los dientes. (c) Encoder montado dentro de la llanta [18].

El encoder proporciona dos señales desfasadas 90° . En la Figura 2.24 se muestra las señales que proporciona el sensor.

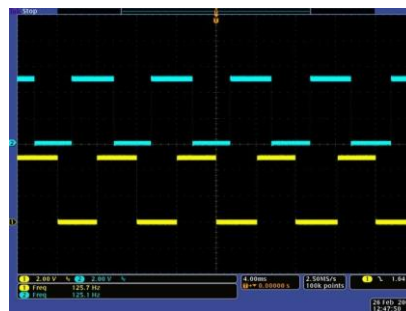


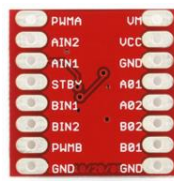
Figura 2.24 Captura del osciloscopio de la salida del encoder [18].

2.4.3.3 Medición de batería

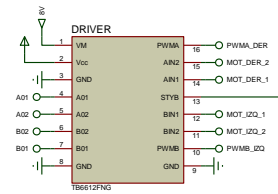
Al igual que en el anterior prototipo es necesario que la batería esté con un nivel de voltaje adecuado por lo que se implementó el mismo circuito del robot seguidor de línea.

2.4.4 CONTROL DE MOTORES

Para hacer el control de los motores se usa el módulo TB6612FNG al igual que en el robot seguidor de línea velocidad, con la diferencia que en este usa la placa disponible en el mercado que se muestra en la Figura 2.25 (a), debido a que en el robot laberinto se dispone de más espacio, y por facilidad de remplazo si fuera necesario.



(a)



(b)

Figura 2.25 Controlador de potencia TB6612FNG para micromotores DC. (a) Módulo disponible en el mercado. (b) Conexiones del módulo en el robot laberinto [30].

En la Tabla 2.4 se muestra las conexiones del microcontrolador al módulo TB6612FNG.

Tabla 2.4 Conexiones del microcontrolador al módulo TB6612FNG.

Motor derecho	
Pin del microcontrolador	Pin del controlador de potencia
PD6	AIN1
PD7	AIN2
PB3	PWMA
Motor izquierdo	
PD5	BIN1
PD4	BIN2
PB4	PWMB

Para la atenuar los efectos de ruido que causan los motores del robot se usa capacitores cerámicos de 0.1 μF conectados en paralelo a cada motor [31]. En la Figura 2.26 se muestran los capacitores de los motores.

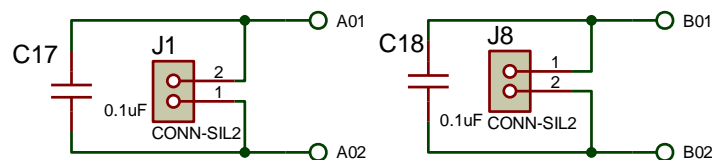


Figura 2.26 Capacitores de los motores en el robot laberinto.

2.5 PERIFÉRICOS DE ENTRADA-SALIDA

2.5.1 PROGRAMADOR

La programación del microcontrolador se lo hace por medio de ISP, por lo cual se sacó los pines (Vcc, GND, MISO, MOSI, SCK Y RST) del microcontrolador hacia una bornera para poder conectar el programador.

2.5.2 COMUNICACIÓN

La parte de comunicación se lo hace con un Módulo Bluetooth HC-06, este módulo es un conversor de puerto Serie a Bluetooth, solo puede funcionar como esclavo, versión 2.0, tiene un alcance de hasta 10 m [32]. La comunicación se la hace a 9600 baudios. Esta sirve para realizar una depuración de los programas que se escriben y saber lo que está pasando dentro del microcontrolador.

2.5.3 PULSADOR

Los pulsadores en el robot seguidor se los utiliza para poder interactuar con él, ordenarle que calibre los sensores, que mida el voltaje de la batería, etc. El robot posee tres pulsadores: SW1, SW2 y START que están conectados a los pines PC1, PC0 y PD6 respectivamente.

En el robot laberinto se dispone de tres pulsadores P1, P2 y P3 que están conectados a los pines PB0, PB1 y PB2 respectivamente. Estos son utilizados para escoger la estrategia de solución del laberinto, para iniciar los movimientos etc.

2.5.4 INDICADORES

En esta parte se ha considerado el uso de diodos led para poder interactuar con el robot; se usó tres leds, estos están conectados a los pines PC2, PC3 y PC4 del microcontrolador.

En el robot laberinto se tiene tres indicadores, los cuales están conectados a los pines PA0, PA1 y PC2. Estos se usan para mostrar de manera visual las acciones

que el robot está realizando en ese momento, como: giro derecha, giro izquierda y giro 180°, etc. En la Figura 2.27 se puede ver el circuito del led.

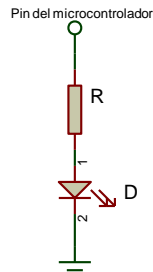


Figura 2.27 Indicadores visuales.

A continuación se hace los cálculos necesarios para el dimensionamiento de la resistencia.

$$R = \frac{V_{CC} - \text{Voltaje}_{\text{diodo}}}{\text{Corriente}_{\text{diodo}}}$$

$$R = \frac{5\text{ V} - 2\text{ V}}{4\text{ mA}} = 750\ \Omega \quad R \approx 1\text{ k}\Omega$$

En el Anexo B se encuentran los circuitos completos del robot seguidor y del robot laberinto.

2.6 ROBOTS DISEÑADOS LISTOS PARA SU CONSTRUCCIÓN

A continuación se muestran los robots diseñados tomando en cuenta todo lo dicho anteriormente. El robot seguidor de línea tiene una distancia entre ruedas de 12 cm, la distancia entre el eje de las ruedas y los sensores es de 15 cm y su peso es 170 g. En la Figura 2.28 se muestra el robot seguidor de línea.

Para el robot laberinto, el chasis del robot tiene una envergadura de 10 cm, la distancia entre las ruedas es de 8 cm y la disposición de los sensores y batería como anteriormente se mencionó; el peso total del robot es de 200 g (Figura 2.29).

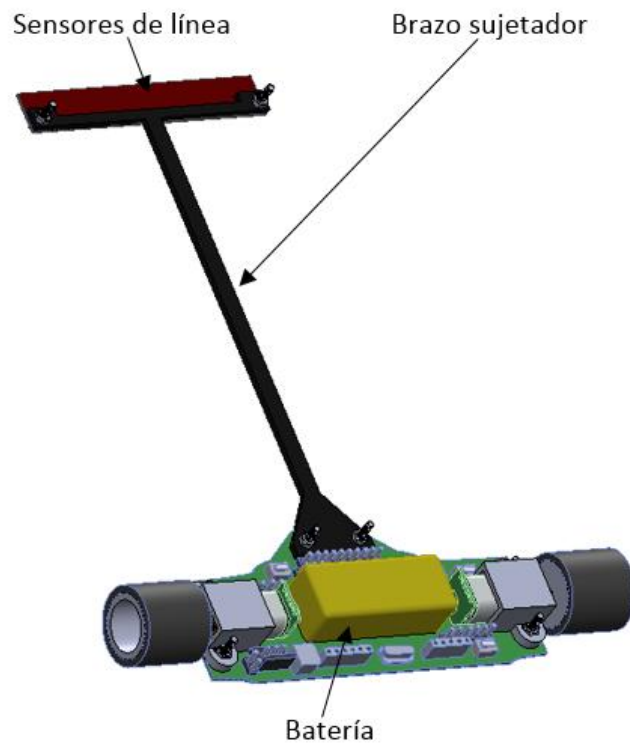


Figura 2.28 Robot seguidor de línea velocidad.

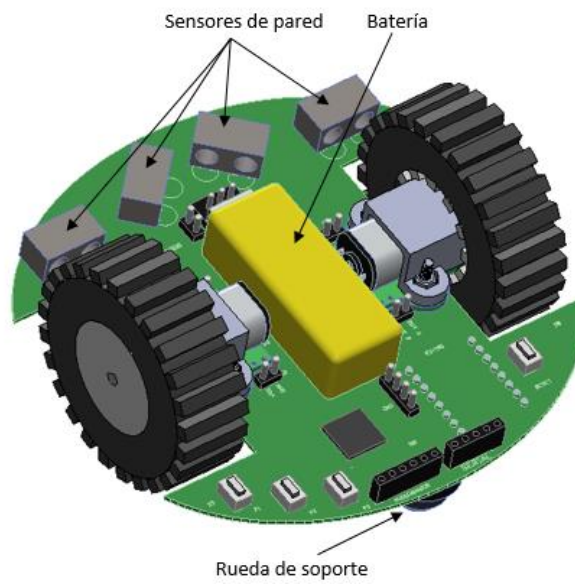


Figura 2.29 Robot laberinto.

CAPÍTULO 3

DESARROLLO DE LOS PROGRAMAS DE CONTROL

3.1 INTRODUCCIÓN

En este capítulo se explica el desarrollo de los programas, los algoritmos para controlar las diferentes partes de los robots y los algoritmos que les permiten a los robots cumplir con sus objetivos (seguir la línea en el caso del robot seguidor y resolver el laberinto en el caso del robot laberinto).

Para el desarrollo de los programas se utilizó el entorno Atmel Studio 7 y para la codificación se utilizó el lenguaje de programación C.

3.2 ARQUITECTURA SOFTWARE PARA ROBOT SEGUIDOR DE LÍNEA

Con el hardware ya listo, ahora es necesario un programa para poder utilizarlo y controlarlo. Se han creado varios módulos o librerías en las cuales están las funciones que se necesitan para el control del robot y que son invocadas desde un programa principal llamado seguidor de línea. La implementación del programa se lo hizo de ésta manera porque permite mejorar el orden del código, ya que a éste se lo distribuye en varios archivos, otra ventaja de hacerlo de esta manera es que contribuye al trabajo en equipo y brinda mayor facilidad en la detección de posibles fallos en el código.

Los módulos que se han escrito son 6, los que se ha denominado: módulo configuración de periféricos, módulo interfaz, módulo sensores, módulo motores, módulo de comunicación y módulo de control. En la Figura 3.1 se puede observar la arquitectura del robot seguidor de línea.

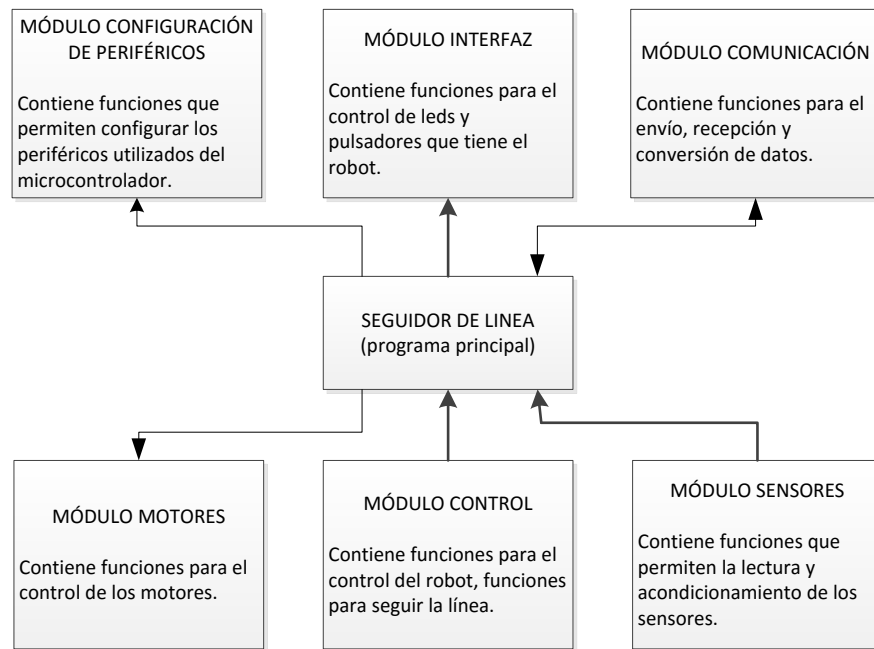


Figura 3.1 Arquitectura de software del robot seguidor de línea.

3.2.1 MÓDULO CONFIGURACIÓN DE PERIFÉRICOS

Este módulo contiene varias funciones las cuales permiten configurar los periféricos utilizados del microcontrolador: una función para configurar los pines, una para configurar el conversor analógico digital, una para configurar las interrupciones, una para el timer que se encarga de la generación de las señales para controlar los motores (modulación por ancho de pulso) y una función para el timer que se encarga del tiempo de muestreo.

3.2.2 MÓDULO SENSORES

Esta librería contiene varias funciones, que permiten leer los sensores de línea, acondicionarlos y proveer la realimentación que el algoritmo de control necesita para seguir la línea. También posee una función que mide el voltaje de la batería y por último una que es usada para obtener la distancia que el robot recorre; las más importantes se las detalla a continuación.

3.2.2.1 Función calibración de sensores

Esta función permite calibrar los sensores de línea, el objetivo de la calibración es hacer que todos los sensores se comporten de manera similar y se adapten a las condiciones de luz. Para la calibración se coloca al robot sobre la pista y se lo hace girar de modo que todos los sensores pasen sobre las superficies blanca y negra; la función guarda los valores máximo y mínimo de cada sensor en la memoria. En la Figura 3.2 se observa el diagrama de flujo de la función calibración de sensores.

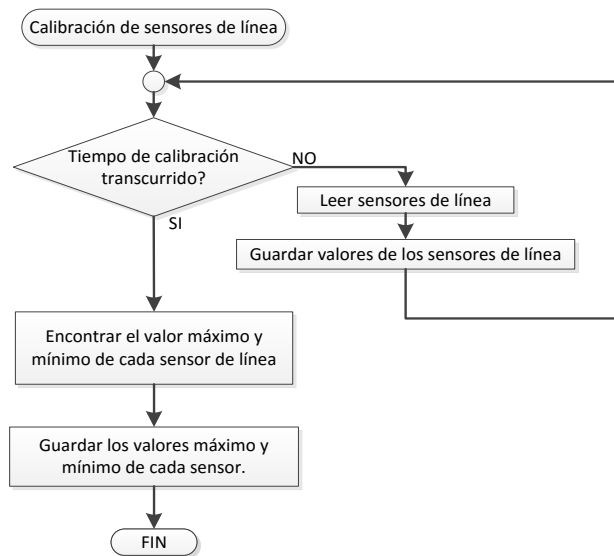


Figura 3.2 Diagrama de flujo de la función calibración de sensores.

3.2.2.2 Función lectura y acondicionamiento de sensor de línea

Lo que hace esta función es tomar los valores máximo y mínimo de cada sensor que fueron establecidos previamente por la función calibración de sensores y realiza un escalamiento entre 0 y 1000, como se puede ver en la Figura 3.3, donde 0 corresponde al valor mínimo (el sensor está sobre una superficie blanca) y 1000 corresponde al valor máximo (el sensor está sobre una superficie negra).

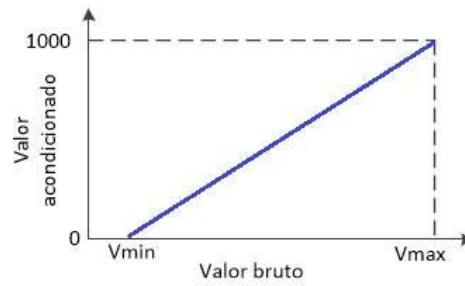


Figura 3.3 Escalamiento de sensores.

Para el acondicionamiento de cada sensor se utiliza la ecuación (3.1):

$$V_N = \frac{1000 * (V_B - v_{min})}{v_{max} - v_{min}} \quad (3.1)$$

Donde:

V_N : Valor acondicionado

V_B : Valor entregado por el sensor

En la Figura 3.4 se observa el diagrama de flujo de la función lectura y acondicionamiento de sensor de línea.

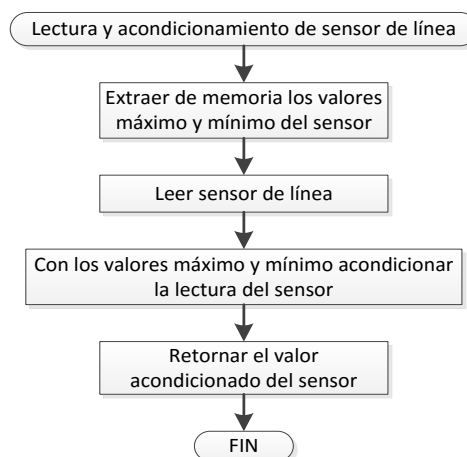


Figura 3.4 Diagrama de flujo de la función lectura y acondicionamiento de sensor de línea.

3.2.2.3 Función cálculo del error de posición

Por medio de esta función se obtiene el error de posición del robot con respecto a la línea, esta función toma los valores ya acondicionados de cada sensor y mediante un promedio ponderado (3.2) se realiza el cálculo del error.

$$Error\ e(n) = \frac{\sum_0^7 L_i * (2i)}{L_i} - 7 \quad (3.2)$$

Donde:

L_i : I-ésimo sensor de línea.

Los sensores están numerados de izquierda a derecha. En la ecuación (3.2), la sustracción por 7 se la realiza para que error cero ocurra cuando el centro del arreglo de sensores este sobre la línea. Un valor de cero indica que el robot está centrado, para valores positivos el robot esta desviado a la izquierda de la línea y para valores negativos esta desviado a la derecha. Además la función indica cuando el robot está sobre la línea o cuando éste se desvió a la izquierda o a la derecha. En la Figura 3.5 se muestra el diagrama de flujo de la función.

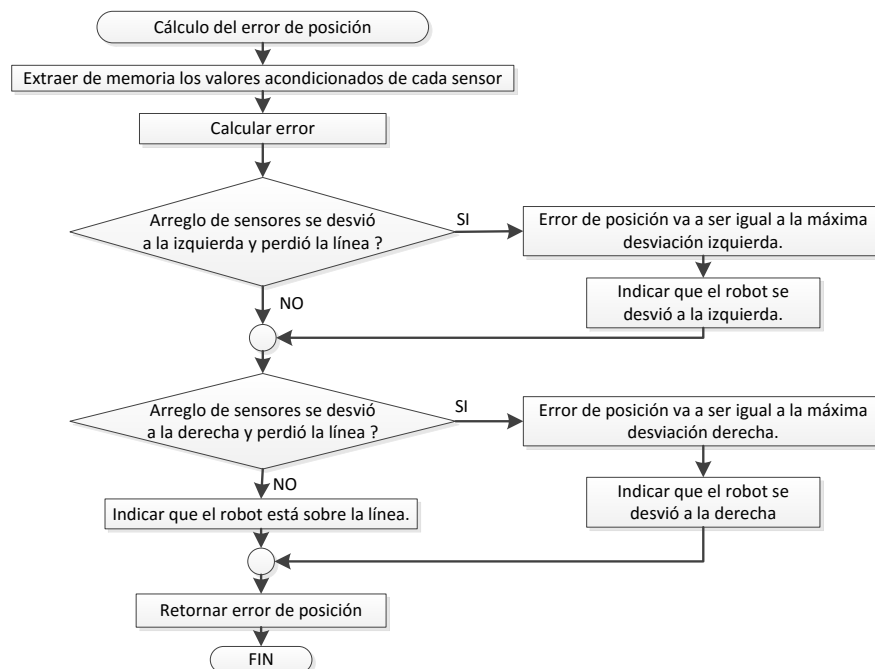


Figura 3.5 Diagrama de flujo de la función cálculo del error de posición.

3.2.2.4 Función sensor batería

Esta función permite leer el voltaje que tiene la batería y calcular una constante con la cual se calcula el voltaje medio que se suministra a los motores para que éstos no cambien su comportamiento en un rango de voltaje de la batería. La constante se la calcula con la ecuación (3.3):

$$\text{ajuste de voltaje} = \frac{\text{voltaje máximo de motor}}{\text{voltaje de batería}} \quad (3.3)$$

El voltaje máximo de alimentación de los motores es 7.5 V y la batería completamente cargada tiene 8.3 V por lo que el rango de operación del robot es de 0.8 V; en la Figura 3.6 se observa el diagrama de flujo de la función sensor batería.

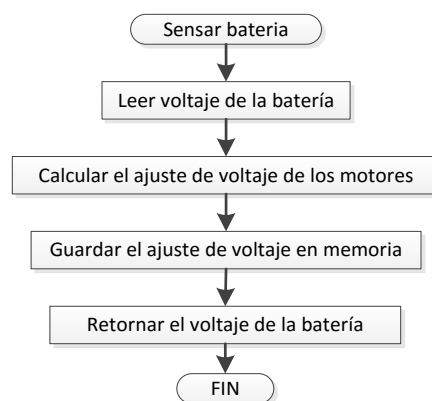


Figura 3.6 Diagrama de flujo de la función sensor batería.

3.2.2.5 Función cálculo de la distancia recorrida

Para medir la distancia que recorre el robot se utiliza un encoder incremental cuya resolución es de 12 pulsos por revolución cuando se cuentan los flancos de los dos canales. Para el cálculo de la distancia se lleva una cuenta por rueda para luego transformar ésta a unidades de distancia (cm). La distancia recorrida se calcula con la ecuación (3.4):

$$d_r = N_p * \frac{2 * \pi * R}{p} * \frac{1}{n} \quad (3.4)$$

Donde:

d_r : Distancia recorrida.

N_p : Número de pulsos.

R : Radio de la rueda = 1 cm.

p : Resolución del encoder = 12

n : Relación de engranes = 10

El número de conteos de cada rueda se lo hace en variables enteras de 16 bits sin signo con lo cual la máxima distancia que se puede medir antes que las variables se desborden se calcula con la ecuación (3.5)

$$\text{distancia máxima} = (2^{16} - 1) * \frac{2 * \pi * R}{p} * \frac{1}{n} = (2^{16} - 1) * \frac{2 * \pi * 1}{12} * \frac{1}{10} \quad (3.5)$$

$$\text{distancia máxima} = 3431.4 \text{ cm} = 34.31 \text{ m}$$

La distancia máxima es lo suficientemente grande con lo cual no habría peligro de desbordamiento ya que por lo general las pistas no llegan a tener esa distancia en su recorrido.

En la Figura 3.7 se puede observar el diagrama de flujo de la función cálculo de la distancia recorrida.

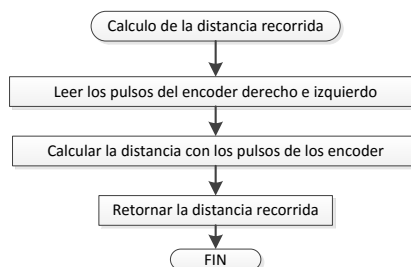


Figura 3.7 Diagrama de flujo de la función cálculo de la distancia recorrida.

3.2.3 MÓDULO MOTORES

Este módulo contiene funciones que permiten controlar la velocidad de cada motor por medio del voltaje que se les suministra a éstos, también permite escoger su sentido de giro. Para variar el voltaje de los motores se utiliza una modulación por ancho de pulso (PWM), la cual es generada por uno de los timers del microcontrolador.

3.2.3.1 Función control de motor derecho/izquierdo

Estas funciones se encargan de la generación de las señales que van al controlador de potencia (TB6612FNG) y permiten accionar los motores. En la Tabla 3.1 se resume las señales que necesita cada motor y en la Tabla 3.2 se detalla la lógica del control de motores.

Tabla 3.1 Señales de control de motores.

Motor derecho	
Pin del microcontrolador	Pin del controlador de potencia
PD4	AIN1
PD5	AIN2
PB3	PWMA
Motor izquierdo	
PB0	BIN1
PB1	BIN2
PB4	PWMB

Tabla 3.2 Lógica del control de motores.

Entrada				Salida		
IN1	IN2	PWM	STBY	OUT1	OUT2	Modo
H	H	H/L	H	L	L	Freno rápido
L	H	H	H	L	H	Giro sentido anti horario
		L	H	L	L	Freno rápido
H	L	H	H	H	L	Giro sentido horario
		L	H	L	L	Freno rápido
L	L	H	H	OFF Alta impedancia		Freno lento
H/L	H/L	H/L	L	OFF Alta impedancia		Espera

Se escribieron dos funciones, una para el motor derecho y otra para el izquierdo. A la función se le pasa como argumento la velocidad de giro del motor, para valores positivos del argumento el motor girará en sentido anti horario y para valores negativos en sentido horario.

En la Figura 3.8 se muestra el diagrama de flujo de la función que controla al motor derecho.

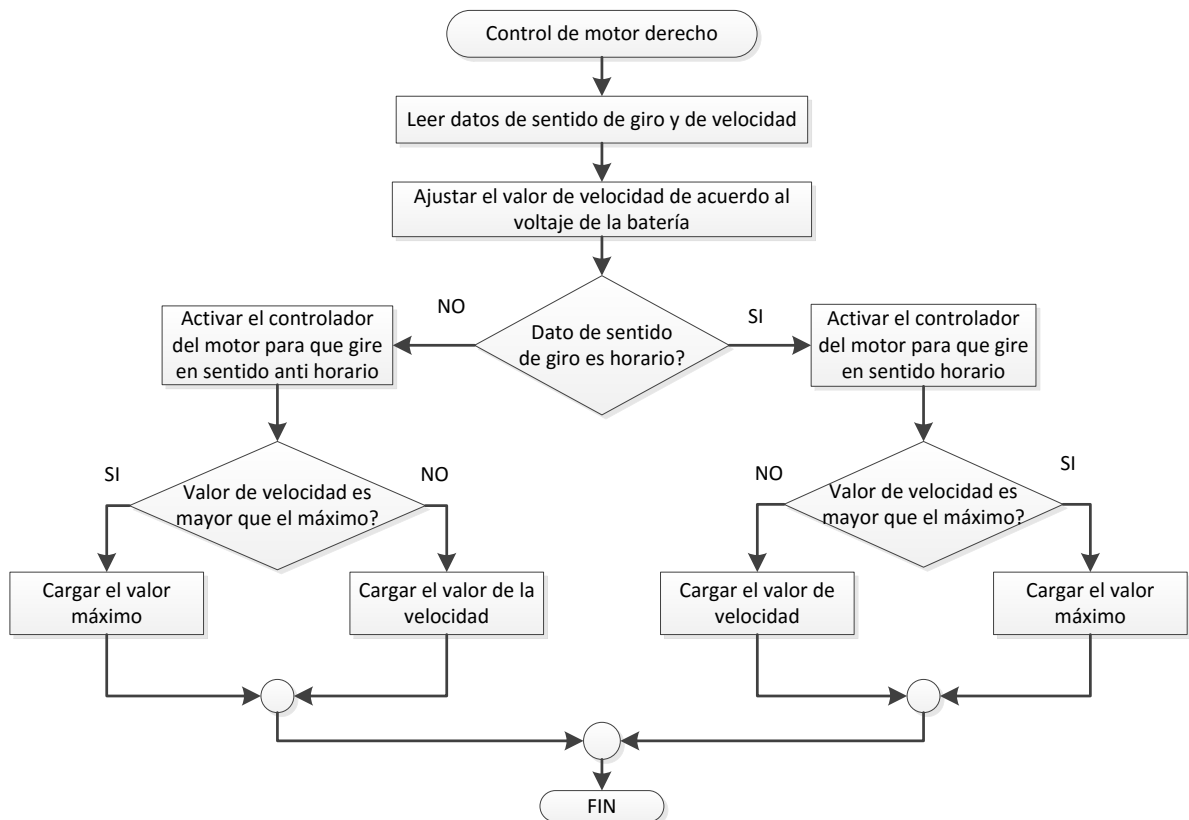


Figura 3.8 Diagrama de flujo de la función control de motor derecho.

3.2.4 MÓDULO INTERFAZ

El módulo interfaz es el encargado de la interacción entre el robot y el usuario, mediante funciones que permiten obtener información de los pulsadores y funciones que permiten controlar los leds.

3.2.5 MÓDULO COMUNICACIÓN

Este módulo es el encargado de la comunicación entre el robot y un dispositivo remoto (pc o smartphone) para lo cual se utiliza el puerto USART (universal synchronous and asynchronous serial receiver and transmitter) del microcontrolador y el módulo bluetooth HC06. La comunicación que se realiza es asíncrona a 9600 baudios, sin paridad y con un bit de parada. El módulo contiene funciones que permiten el envío, recepción y conversión de datos.

3.2.6 MÓDULO DE CONTROL

Este módulo contiene las funciones que permiten controlar al robot para que siga la línea, los controladores a implementarse son un proporcional Integral derivativo (PID) y un borroso. El objetivo es que el prototipo siga la línea a alta velocidad y que el error con respecto a la línea sea cercano a cero pero en realidad el único parámetro a tomar en cuenta en el rendimiento de controlador es el tiempo que le toma al prototipo en recorrer la pista.

3.2.6.1 Características del sistema [33]

El error en estado estacionario del sistema es usualmente usado para clasificarlo en un tipo, es decir, el número de integradores que tiene, se considera una realimentación unitaria.

La prueba del error en estado estable es hecha contando el número de sobre impulsos que se dan a la izquierda y derecha de la línea. Para iniciar se pone al robot con la máxima desviación que puede leer el arreglo de sensores.

Cuando el robot es colocado en una línea recta, el robot oscila a ambos lados de la línea con igual magnitud, esto indica que el error en estado estacionario es cero para una línea recta. Cuando el robot es colocado a seguir una línea curva las oscilaciones del robot ya no son centradas respecto a la línea, la oscilación es mayor en la dirección opuesta a la curva. Esto indica que existe un error en estado estacionario cuando la línea es curva.

Ahora, se puede concluir que, cuando el sistema tiene cero o pequeña perturbación (como una entrada paso) el error de estado estacionario es cero (error de posición); cuando la perturbación es grande (análogo a una entrada rampa o parabólica) tiene error de estado estacionario (error de velocidad o aceleración). Por lo que se puede decir que el sistema con el que se está tratando es de tipo 1 o mayor.

También al prototipo se le hicieron algunas pruebas en lazo abierto según [33];

Se pone al robot exactamente en el centro de una línea recta y en la dirección de ésta, como no hay ningún esfuerzo de control, los motores giran a la misma velocidad y el robot se mueve con error de cero. Si hay alguna circunstancia como que la línea no sea completamente recta, que el robot no haya estado bien alineado o que se dé una diferencia de velocidad entre las ruedas se necesita un esfuerzo de control (la velocidad de uno de los motores debe ser reducida).

Ahora, se pone al robot alejado del centro de la línea por una cierta distancia, con el fin de reducir el error se da al robot un esfuerzo de control, pero el ejercicio constante de ese esfuerzo hace que el robot rebase la línea.

Teniendo en cuenta las consideraciones anteriores, se puede ver que el bucle de control se está integrando, es decir, el esfuerzo de control es integrado por el sistema, la respuesta de bucle abierto del sistema se parece a la de un sistema de posición de un servo por lo tanto las técnicas de control de estos pueden ser aplicables al robot, la más usada es el controlador proporcional derivativo (PD).

3.2.6.2 Control Proporcional Integral Derivativo (PID) [33]

Se parte de la ecuación de un controlador proporcional integral derivativo (PID) en forma paralela, ecuación (3.6):

$$u(t) = k_p * e(t) + k_I * \int_0^t e(t) * dt + k_D * \frac{d e(t)}{dt} \quad (3.6)$$

Para poderlo aplicar se la discretizó, ecuación (3.7). El término que representa el tiempo de muestreo se lo simplifica debido a que es una constante que se multiplica por las constantes k_I y k_D .

$$u(n) = K_p * e(n) + K_I * \sum_0^n e(n) + K_D * (e(n) - e(n - 1)) \quad (3.7)$$

Donde: $K_I = k_I * t_s$ y $K_D = \frac{k_D}{t_s}$

Las velocidades de los motores se establecen en función de dos parámetros, la salida del controlador y la velocidad del robot (velocidad crucero), las ecuaciones (3.8), (3.9), (3.10) y (3.11) representan las velocidades de cada motor.

Si el error es negativo:

$$motor_{izquierdo} = velocidad_{robot} + u(n) \quad (3.8)$$

$$motor_{derecho} = velocidad_{robot} \quad (3.9)$$

Si el error es positivo:

$$motor_{izquierdo} = velocidad_{robot} \quad (3.10)$$

$$motor_{derecho} = velocidad_{robot} - u(n) \quad (3.11)$$

De lo mencionado anteriormente se mira que cuando el robot se desvía a la derecha (error negativo), el motor derecho se mantiene con la velocidad de crucero mientras que el motor izquierdo reduce la velocidad para que el robot se mueva a la izquierda; y cuando esta desviado a la izquierda (error positivo) el motor izquierdo se mantiene con la velocidad de crucero y el motor derecho reduce su velocidad.

3.2.6.2.1 Implementación del controlador PID

La implementación directa del controlador propuesto en la sección anterior tiene un problema con la parte integral ya que funciona bien cuando la pista se compone de una curva simple pero cuando ésta tiene varias curvas y rectas el robot se desestabiliza.

Para resolver los problemas que provoca la implementación directa de la acción integral se realiza una modificación, se toma únicamente los últimos 10 valores para esta acción en lugar de todos [33], ecuación (3.12):

$$u(n) = K_p * e(n) + K_I * \sum_{n-9}^n e(n) + K_D * (e(n) - e(n - 1)) \quad (3.12)$$

Esto permite un esfuerzo de control integral para atacar el error en estado estable que se da en las curvas a la vez que también da la adaptabilidad a los cambios de curvatura de la pista.

Se ha creado una función llamada PID cuyo diagrama de flujo se muestra en la Figura 3.9. Es una función a la cual se le pasa como argumentos el error, la constante proporcional, constante derivativa, constante integral, velocidad de cruceo del robot, velocidad del motor derecho y velocidad del motor izquierdo.

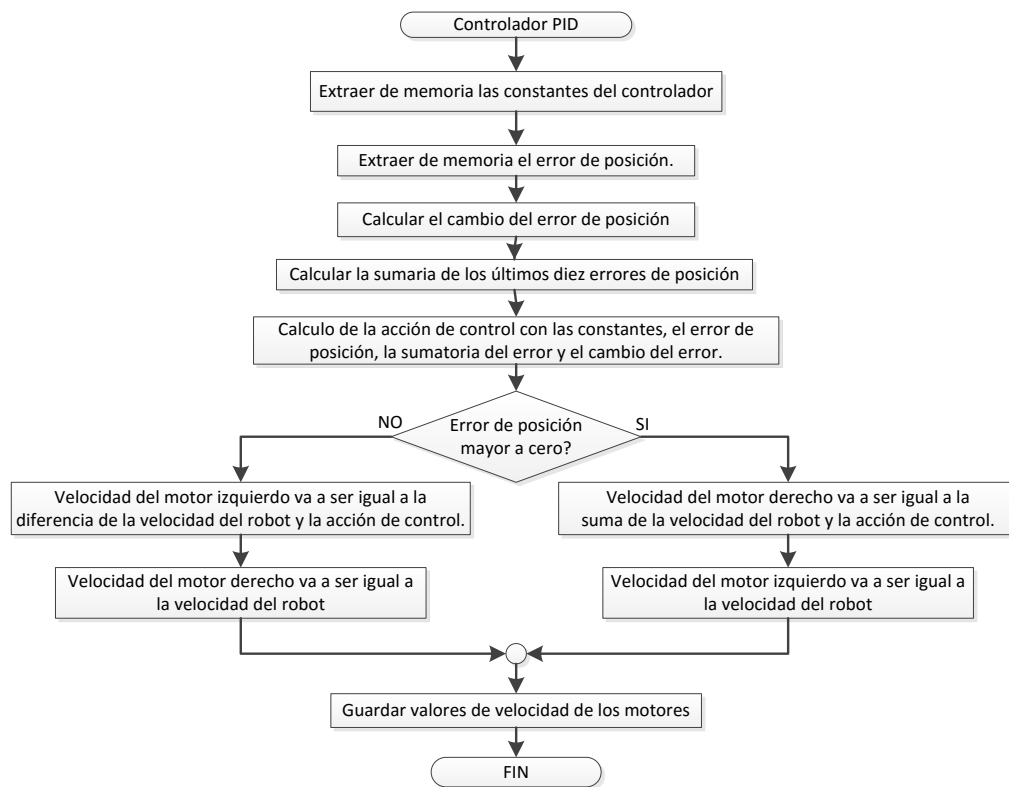


Figura 3.9 Diagrama de flujo del controlador PID.

3.2.6.2.2 Adición de control en lazo abierto

El control PID que se implementó funciona bien para líneas rectas y curvas moderadas pero cuando las curvas son de radios pequeños (radio menor a 30 cm) el desempeño del prototipo no es muy bueno. Esto debido a que cuando el robot pierde la línea hace que el error se sature. Bajo estas circunstancias el esfuerzo de control producido por el PID no puede hacer girar al robot lo suficientemente rápido, para resolver este problema se utiliza un esquema de control de lazo abierto [33].

Esta idea consiste en setear ciertas velocidades a los motores dependiendo del sentido de la curva y de su radio hasta que el robot detecta nuevamente la línea y puede seguir con el lazo de control normal.

3.2.6.2.3 Arranque

En el desarrollo de este proyecto se ha participado ya en varios torneos, y de la experiencia obtenida se ha pensado en hacer que el robot corra a diferentes velocidades, se quiere que el robot arranque a la mayor velocidad posible pero hay que tener cuidado ya que si en el arranque se acciona los motores al máximo el robot se vuelve inestable y pierde pista por lo que se ha implementado una curva de arranque como la que se muestra en la Figura 3.10.

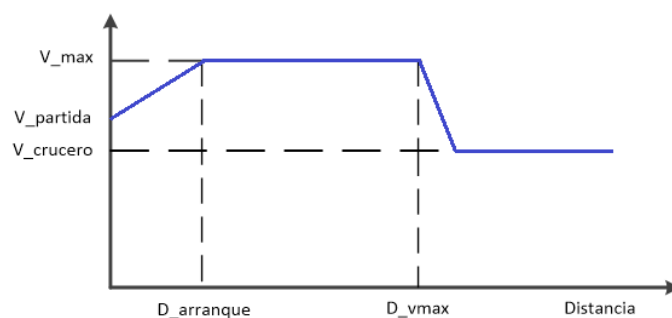


Figura 3.10 Curva de velocidad en el arranque.

En donde la distancia de arranque ($D_{arranque}$), la distancia de velocidad máxima (D_{vmax}), la velocidad de partida ($V_{partida}$), la velocidad de cruceo ($V_{crucero}$, velocidad con la que el robot recorre el resto de la pista) y velocidad máxima (V_{max}) se las ajusta de acuerdo a la pista del torneo.

3.2.6.3 Controlador Borroso

El sistema de control consta de dos entradas y una salida, donde las entradas son: el error (e) que es la desviación actual del robot con respecto a la línea menos la desviación que se desea (desviación cero) y Δe que representa el cambio del error. La variable de salida u representa la diferencia de velocidades de las ruedas para que el robot siga la línea y tenga un error de cero (sin desviación); en la Figura 3.11 se indica el esquema de control.

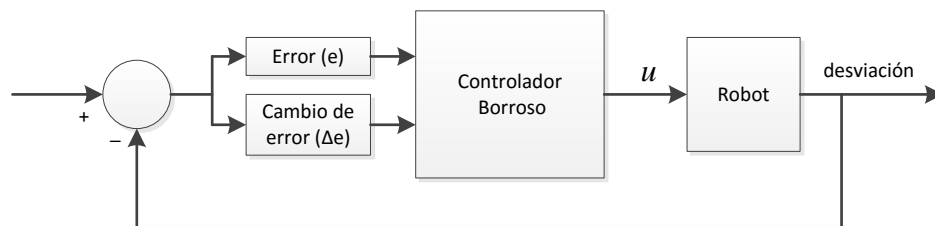


Figura 3.11 Esquema de control borroso para el robot seguidor de línea.

En el diseño de este controlador se empleará la siguiente notación:

E_i : Términos lingüísticos asociados a la variable error.

ΔE_i : Términos lingüísticos asociados a la variable cambio del error.

U_i : Términos lingüísticos asociados a la variable acción de control.

u_{E_i} : Función de pertenencia asociada a la etiqueta lingüística E_i .

$u_{\Delta E_i}$: Función de pertenencia asociada a la etiqueta lingüística ΔE_i .

$u_{U_i}(u)$: Función de pertenencia asociada a la etiqueta lingüística U_i .

w_i : Peso de la i -ésima regla.

3.2.6.3.1 Borrosificación

Consiste en calcular el valor de pertenencia de las variables de entrada (e , Δe) a las etiquetas lingüísticas (E_i , ΔE_i) utilizando las funciones de pertenencia. Para esto primero es necesario establecer el universo de discurso para cada una de las

variables. Teniendo en cuenta que el error (e) está descrito por la ecuación (3.13), el universo de discurso está en el rango de -14° a 14° .

$$e = \text{referencia} - \text{desviación} \quad (3.13)$$

Donde 14° es la máxima desviación que puede captar el arreglo de sensores cuando éste se desvía a la izquierda y -14° es la máxima desviación a la derecha; cuando el error es cero el prototipo no tiene desviación.

Ahora, el cambio del error Δe está descrito por la ecuación (3.14).

$$\Delta e = \frac{e - e_1}{T} \quad (3.14)$$

Dónde:

T : Tiempo de muestreo.

e_1 : Error anterior a e .

La entrada Δe puede tener valores positivos (e mayor que e_1) y negativos (e_1 mayor que e) y el universo de discurso es desde -1400 a 1400.

Para la definición del número de etiquetas lingüísticas que se debe usar no existe una regla formal, todo depende del diseñador [34]. Para la variable error se utiliza tres funciones de membresía cuyas etiquetas lingüísticas son: negativo (N), cero (C) y positivo (P). Los nombres que se les dé a las etiquetas no son determinantes en el desempeño del controlador borroso, éstas solo ayudan a identificar a cada función de membresía.

Se usó funciones de membresía tipo triangulares ya que son las de menor coste computacional teniendo en cuenta que el controlador se va a implementar en un microcontrolador, en la Figura 3.12 se puede apreciar las funciones de membresía de la variable error.

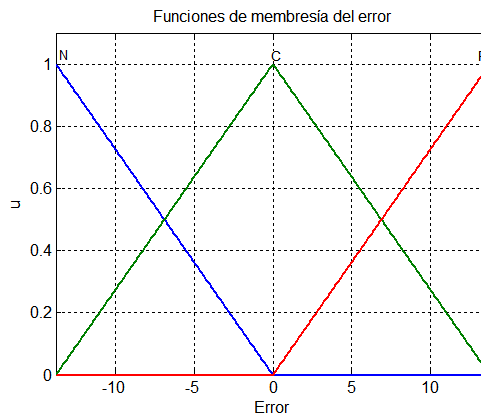


Figura 3.12 Funciones de membresía del error.

Para la variable cambio del error (Δe) se utilizó tres etiquetas lingüísticas, negativa (N), cero (C) y positiva (P): En la Figura 3.13 se muestra las funciones de membresía.

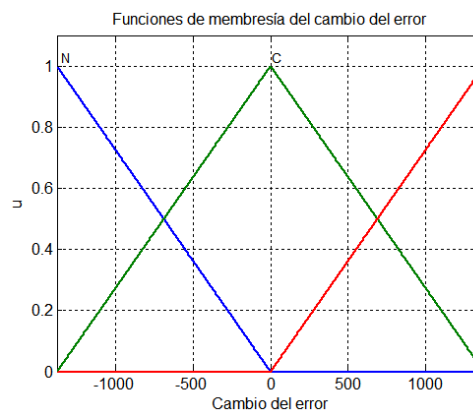


Figura 3.13 Funciones de membresía del cambio del error.

En esta parte también se definen las funciones de membresía de la acción de control: negativa muy grande (NMG), negativa grande (NG), negativa mediana (NM), negativa pequeña (NP), cero (C), positiva pequeña (PP), positiva mediana (PM), positiva grande (PG) y positiva muy grande (PMG). En la Figura 3.14 se puede apreciar las funciones de membresía para la acción de control.

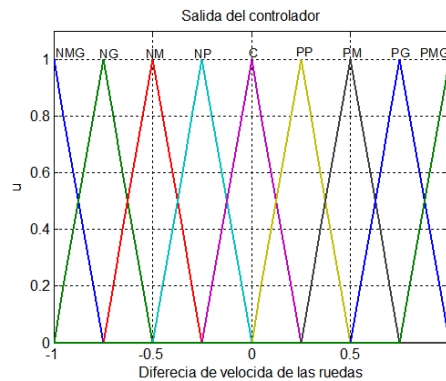


Figura 3.14 Funciones de membresía de la salida del controlador.

Con el diseño que se propone, cuando el error es positivo (desviación a la izquierda) la rueda derecha girará a menor velocidad que la izquierda, cuando el error es negativo (desviación a la derecha) la rueda izquierda girará a menor velocidad que la derecha, y cuando el error sea cero las ruedas girarán a la misma velocidad.

3.2.6.3.2 Reglas de inferencia

Para establecer las reglas de control es necesario cubrir todos los posibles valores que pueden darse en las entradas del controlador y así asegurar que en su salida exista un valor concreto y exacto que promueva una acción de control óptima [35].

El número de entradas y el número de funciones de membresía por cada entrada influye en el número de reglas que puede tener el controlador borroso, por lo que si se tiene en cuenta que el diseño posee dos entradas y tres funciones de membresía por cada una de ellas, van a existir 9 reglas que pueden ser activadas.

1. Si el error es negativo y el cambio del error es negativo.
2. Si el error es negativo y el cambio del error es cero.
3. Si el error es negativo y el cambio del error es positivo.
4. Si el error es cero y el cambio del error es negativo.
5. Si el error es cero y el cambio del error es cero.
6. Si el error es cero y el cambio del error es positivo.
7. Si el error es positivo y el cambio del error es negativo.
8. Si el error es positivo y el cambio del error es cero.
9. Si el error es positivo y el cambio del error es positivo.

Se toma en cuenta todas las combinaciones posibles de entradas al controlador con el objetivo de que no queden huecos ya que el controlador no sabría qué hacer, y la acción de control no sería la adecuada. Ahora se necesita decidir qué acción debe ejecutar el controlador para cada una de las nueve posibilidades [35].

Aquí se utiliza las etiquetas lingüísticas para de una manera sencilla describir la acción de control que necesita el controlador borroso para funcionar adecuadamente, por ejemplo:

1. Si el error es negativo y el cambio del error es negativo, entonces la acción de control aplicada es negativa muy grande.
2. Si el error es negativo y el cambio del error es cero, entonces la acción de control aplicada es negativa grande.

Por lo tanto las reglas de inferencia quedan como sigue:

1. Si e es N y Δe es N, entonces u es NMG.
2. Si e es N y Δe es C, entonces u es NG.
3. Si e es N y Δe es P, entonces u es NM.
4. Si e es C y Δe es N, entonces u es NP.
5. Si e es C y Δe es C, entonces u es C.
6. Si e es C y Δe es P, entonces u es PP.
7. Si e es P y Δe es N, entonces u es PM.
8. Si e es P y Δe es C, entonces u es PG.
9. Si e es P y Δe es P, entonces u es PMG.

3.2.6.3.3 Selección de reglas

El grado de cumplimiento de la premisa puede tomarse como el producto de las condiciones (3.15), tomándose éste valor para la conclusión final, es decir que se toma como operador lógico para la intersección la función producto. Cada regla tiene como peso w , donde, w es el producto de los respectivos grados de pertenencia de e y Δe a las etiquetas correspondientes a esa regla.

$$w_i = u_{Ei}(e) * u_{\Delta Ei}(\Delta e) \quad (3.15)$$

3.2.6.3.4 Aplicación de la regla

Hasta aquí se dispone de un conjunto de reglas y de un peso de cada una para la conclusión final, para calcular la conclusión de cada una de las reglas se debe multiplicar la función primitiva por el peso, Ecuación (3.16).

$$u_{U_i}(u) = w_i * u_{U_i}(u) \quad \forall u, \forall i \quad (3.16)$$

El resultado final de la aplicación de todas las reglas es una serie de conjuntos borrosos con sus respectivas funciones de pertenencia. Con el fin de obtener un único conjunto a partir de los anteriores se utiliza la función máximo (3.17).

$$u_U(u) = \max(u_{U_i}(u)) \quad \forall i \quad (3.17)$$

3.2.6.3.5 Desborrosificación

Como salida se tiene un conjunto borroso pero éste no se lo puede aplicar a los actuadores directamente por los que es necesario una etapa de desborrosificación para poder obtener un valor numérico a partir del conjunto borroso de salida [36], existen varios métodos para esto, en este trabajo se empleó el método del centroide, ecuación (3.18); en la que la salida es el centro de gravedad del área total resultante; es el método que más información toma en cuenta para el cálculo y contribuye a que la salida se mueva suavemente.

$$cg = \frac{\sum y u_D(y)}{\sum u_D(y)} \quad (3.18)$$

3.2.7 PROGRAMA PRINCIPAL

El programa principal se lo ha desarrollado de tal forma que tenga todo lo necesario para el cumplimiento de su objetivo y que la interacción con él sea fácil. La parte principal del programa son los algoritmos que permiten seguir la línea, pero además de esto el programa permite verificar de manera rápida el estado de algunas partes del robot (motores, sensores de línea y encoders) en caso de sufrir algún accidente, además se puede comunicar con un dispositivo externo (una computadora o smartphone), para enviar por ejemplo datos de los sensores o el estado de la batería y receptar las constantes del controlador.

Para la implementación se ha utilizado las librerías y funciones que se vieron en secciones anteriores (Figura 3.15). El programa consta de un menú principal, en el que están las opciones: calibración de sensores, menú de revisión breve del robot y menú de carrera. En el menú de revisión breve del robot se tiene: comprobación de motores, comprobación de encoders y medición de batería; en el menú de carrera se tiene las opciones de controlador borroso y controlador PID.

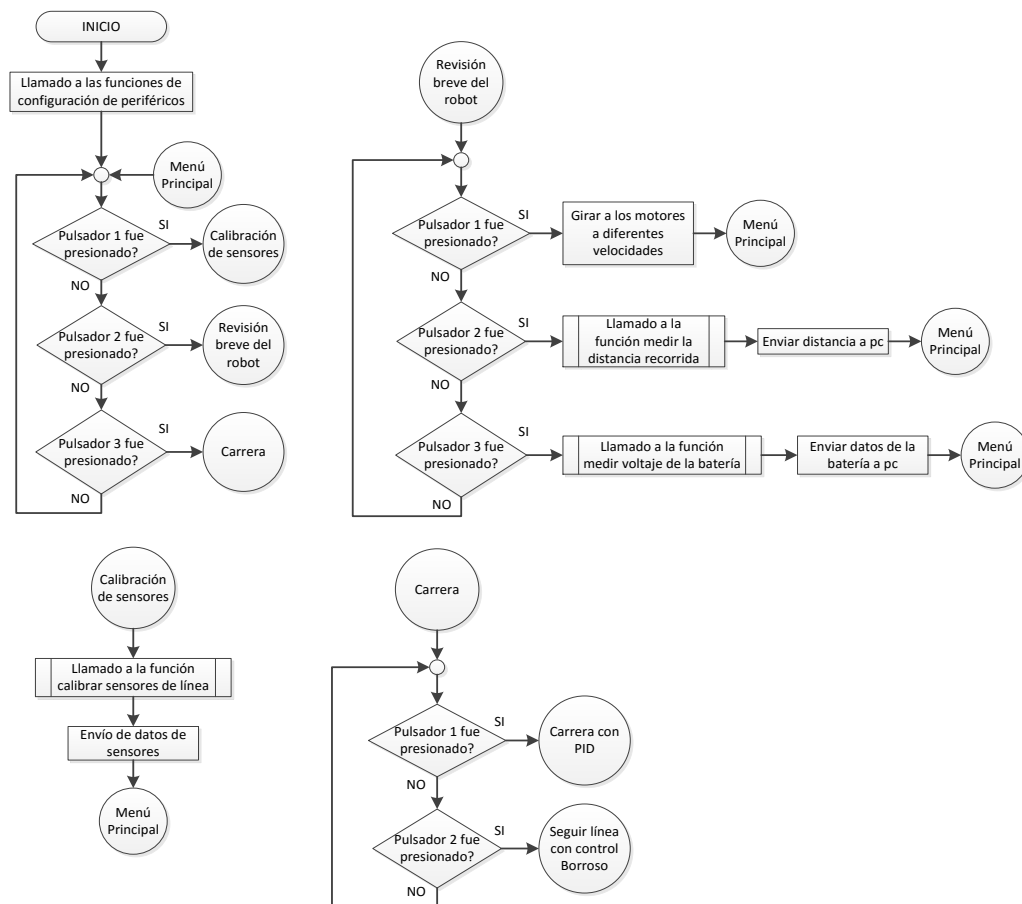


Figura 3.15 Diagrama de flujo del programa principal.

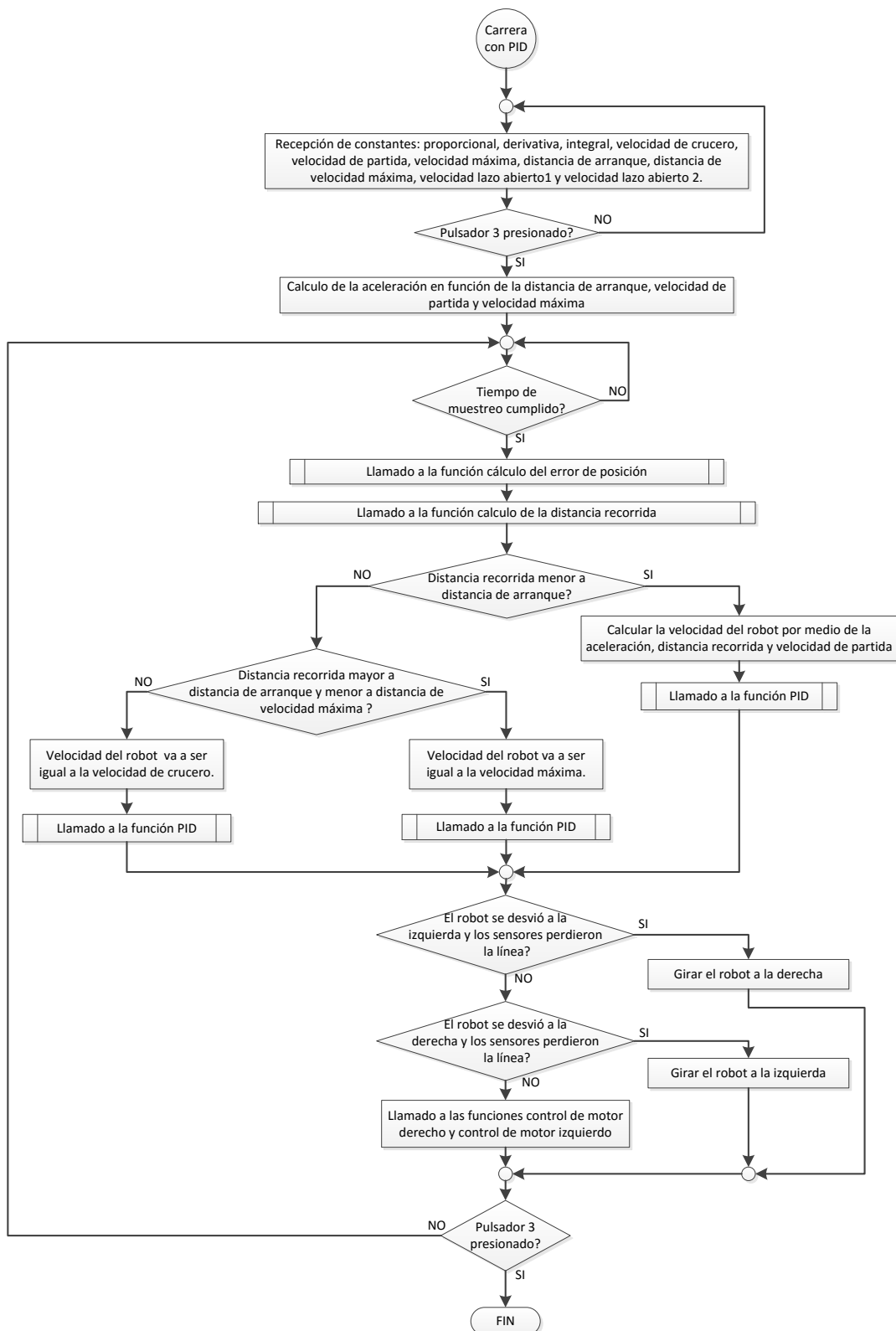


Figura 3.16 Diagrama de flujo del controlador PID.

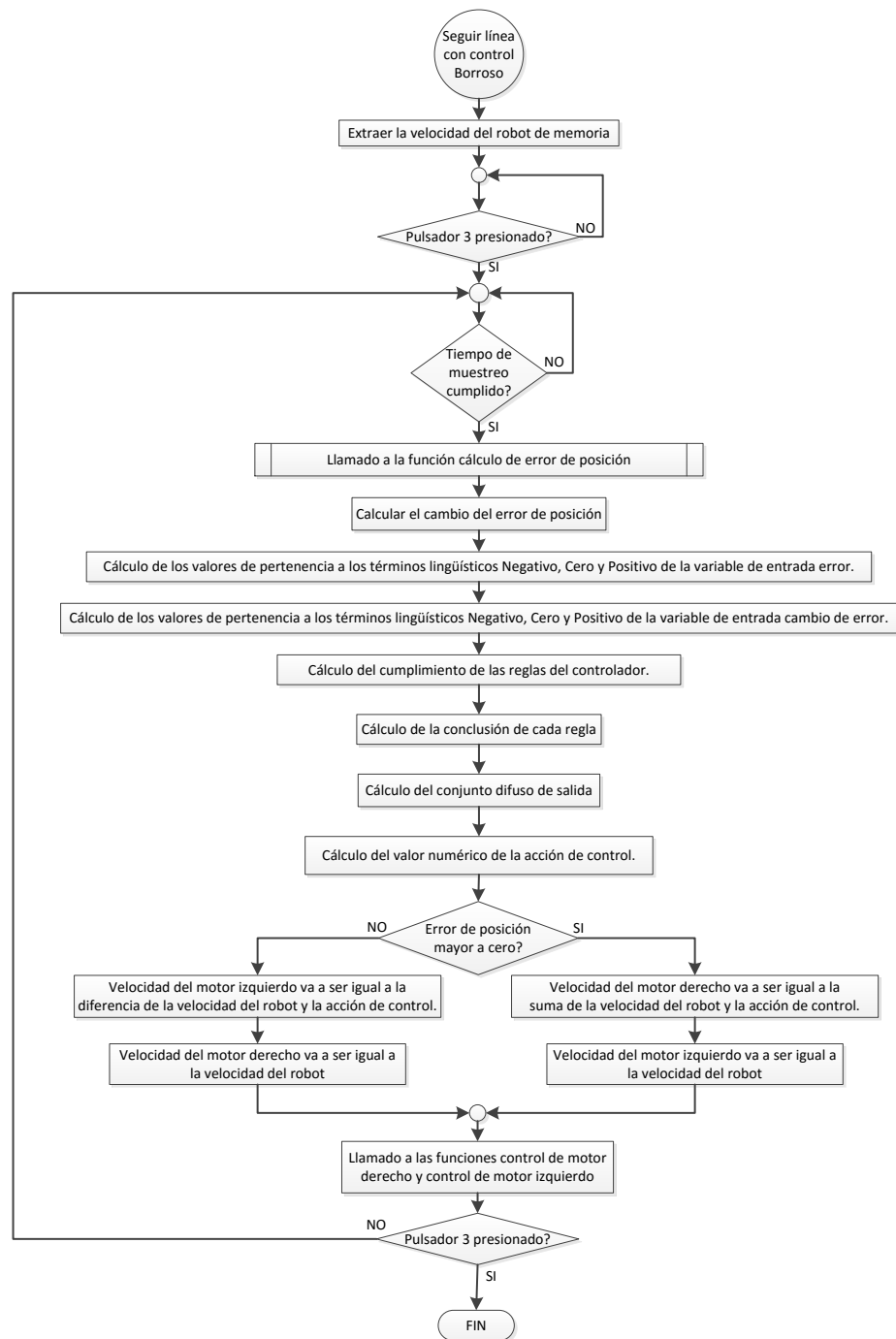


Figura 3.17 Diagrama de flujo del controlador borroso.

Como se puede observar en los diagramas de flujo, cuando corre con PID se necesita que se le envíen las constantes; en el manual de usuario (anexo A) se indica cómo hacerlo.

3.3 ARQUITECTURA SOFTWARE PARA ROBOT LABERINTO

En el desarrollo del proyecto se hizo que las partes constitutivas del programa principal de cada robot sean similares para poder reutilizar código. En la Figura 3.18 se observa la arquitectura de software del robot laberinto.

Es así que los módulos de configuración de periféricos, módulo de motores, módulo de interfaz, y módulo de comunicación son similares. En el módulo de interfaz se agregó una función la cual permite controlar un buzzer. Los módulos que son diferentes son el de sensores y el de control.

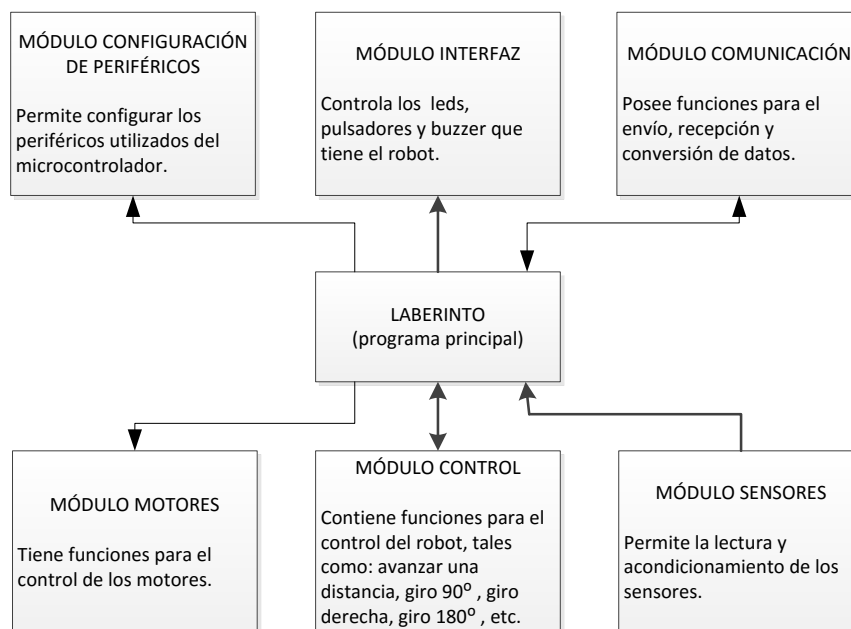


Figura 3.18 Arquitectura de software del robot laberinto.

3.3.1 MÓDULO CONFIGURACIÓN DE PERIFÉRICOS

Al igual que en el robot seguidor de línea, el módulo configuración de periféricos es usado en la configuración de pines, en la inicialización de timers y en la configuración de conversor analógico digital e interrupciones.

3.3.2 MÓDULO LECTURA DE SENSORES

Este módulo contiene varias funciones, por medio de las cuales se obtiene la información necesaria de los sensores, es decir: obtener la distancia del robot a la pared, detectar que paredes están cerca, calcular el error de posición y medir el voltaje de la batería.

3.3.2.1 Función acondicionamiento del sensor de pared

Esta función realiza la lectura del sensor infrarrojo y transforma la señal de voltaje que éste da a unidades de distancia (milímetros). La función también se encarga de disminuir los efectos de la luz ambiente sobre los sensores.

Primero se trata de disminuir los efectos de la luz ambiental sobre el sensor, para lo cual se lee el sensor con el emisor apagado (proporciona una medida de la luz ambiental), luego se lee el sensor con el emisor encendido (proporciona una medida de la luz ambiental más la luz infrarroja del led emisor), el valor del sensor con el emisor apagado se resta del valor del sensor con el emisor encendido dando como resultado el valor de la luz infrarroja del led emisor reflejada en la pared quitando la luz ambiental, esto ayuda a evitar que los sensores sean afectados por los cambios de la luz ambiental.

El valor obtenido solo es una medida de la reflexión de la luz infrarroja emitida por el led por lo que se debe transformar a unidades de distancia, para lo cual se utiliza una regresión. Para la realización de la regresión se toman datos del voltaje que da el sensor y de la distancia de éste con respecto a la pared (los datos se tomaron cada 10 mm), a los sensores se los acondicionó para medir una distancia de hasta 30 cm con lo cual se puede prever una pared con más de una celda de anticipación (una celda mide 25 x 25 cm).

En la Figura 3.19 se puede apreciar el comportamiento del sensor, en el eje de las abscisas se tiene el valor del conversor análogo digital y en el eje de las ordenadas la distancia con respecto a la pared.

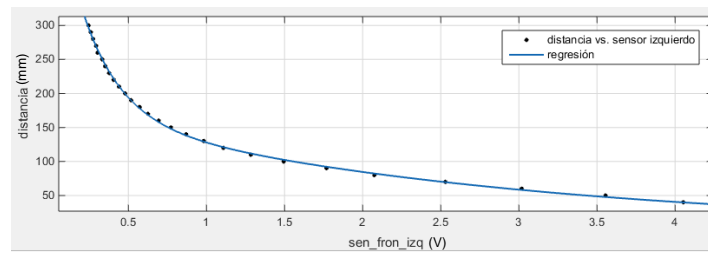


Figura 3.19 Distancia en mm vs lectura del sensor frontal derecho.

Al obtener la curva, se puede ver claramente la formación de una función exponencial. Con ayuda de Matlab se realizó una regresión para obtener la ecuación que mejor se ajusta a los datos obtenidos, la ecuación resultante es (3.19).

$$f(x) = a * e^{b*x} + c * e^{d*x} \quad (3.19)$$

La ecuación tiene cuatro parámetros: a, b, c y d que se los calcula con Matlab y que permiten ajustar la curva para que concuerde con los datos obtenidos. Entonces, la lectura que da el sensor es ingresada en la ecuación (3.19) dando como resultado la distancia del sensor con respecto a la pared. En la Figura 3.20 se observa el diagrama de flujo de la función acondicionamiento del sensor de pared.

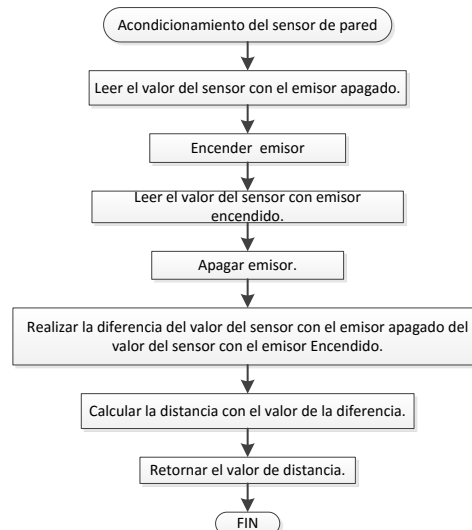


Figura 3.20 Diagrama de flujo de la función acondicionamiento del sensor de pared.

3.3.2.2 Función detectar paredes

Esta función realiza una medición de la distancia de cada sensor a la pared y en base a una distancia referencia de cada una se determina si existe o no pared (Figura 3.21).

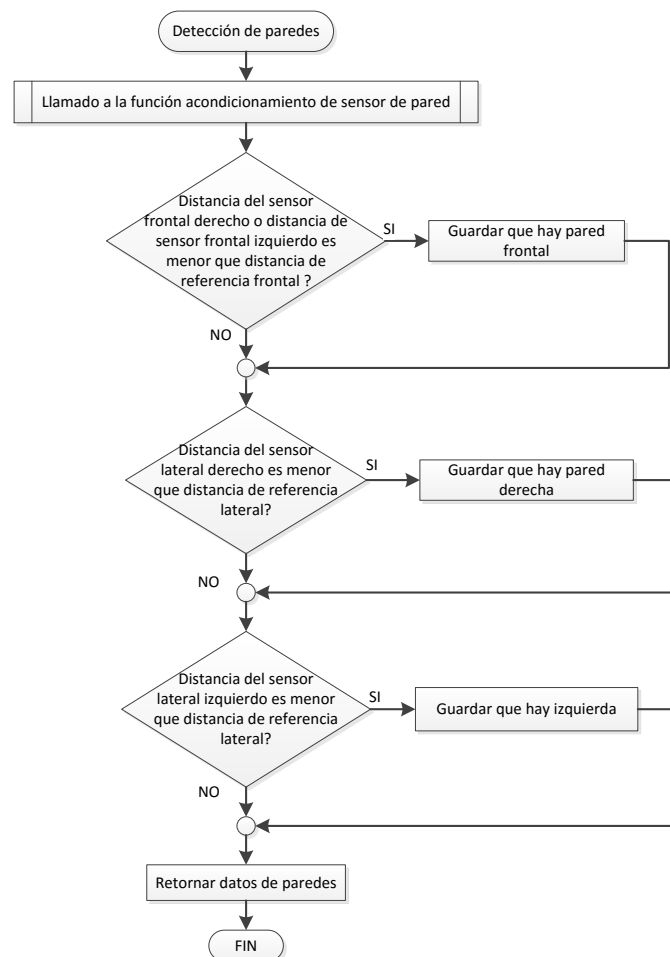


Figura 3.21 Diagrama de flujo de la función detección de paredes.

3.3.2.3 Función obtener error de posición

Esta función retorna el error de posición (en milímetros) que tiene el robot con respecto a una pared. Para calcular el error se lee la distancia que tiene el sensor con respecto a la pared y se lo resta del set point, el cual es fijado por una función en el programa principal (sección 3.3.6).

En la Figura 3.22 se observa el diagrama de flujo de la función obtener error de posición con respecto a la pared derecha.

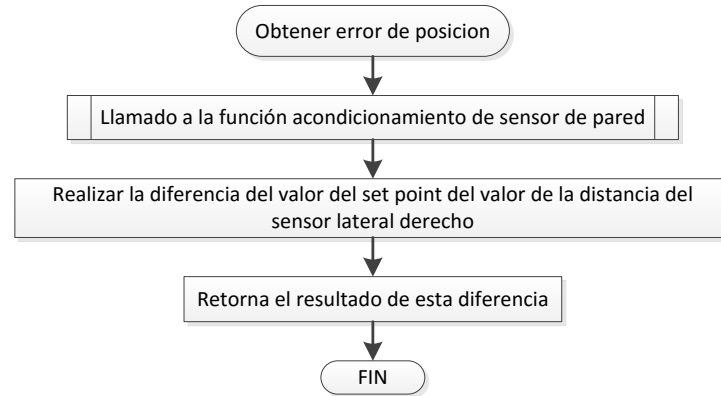


Figura 3.22 Diagrama de flujo de la función obtener error de posición derecho.

3.3.3 MÓDULO MOTORES

Éste módulo es similar al del robot seguidor de línea, se utiliza el mismo diagrama de flujo, lo único que cambia son los pines con los cuales se controla al controlador de potencia.

3.3.4 MÓDULO INTERFAZ

Este módulo es el encargado de la interacción entre el usuario y el prototipo, tiene funciones que permiten obtener información de los pulsadores y funciones que permiten controlar el buzzer y los leds.

3.3.5 MÓDULO DE CONTROL

Este módulo contiene las funciones que permiten controlar al robot y desplazarse por el laberinto. Las funciones que tiene este módulo son: avance con pared derecha, avance con pared izquierda, avance con dos pares, avanzar una celda, girar a la derecha en su propio eje, girar a la izquierda en su propio eje, giro 180° en su propio eje, girar a la derecha con un radio, girar a la izquierda con un radio y posicionar con pared frontal.

3.3.5.1 Función avance con pared derecha

Esta función permite avanzar al robot por el centro de la celda teniendo como referencia la pared derecha (Figura 3.23). Para controlar el avance del robot se mide el error de posición con respecto a la pared derecha y se ejecuta un controlador proporcional derivativo (PD) que proporciona una acción de control. Para setear las velocidades de las ruedas se tiene en cuenta dos parámetros, la acción de control y la velocidad de cruceo (la velocidad de cruceo es velocidad a la cual se desplaza el prototipo) de acuerdo con las ecuaciones (3.20) y (3.21):

$$motor_{izquierdo} = velocidad_{robot} + acción\ de\ control \quad (3.20)$$

$$motor_{derecho} = velocidad_{robot} - acción\ de\ control \quad (3.21)$$

De lo dicho, cuando el error es positivo (el robot está a la derecha del centro de la celda) la rueda izquierda girará a una velocidad menor que la derecha, y viceversa cuando el error es negativo (el prototipo está desviado a la izquierda del centro de la celda).

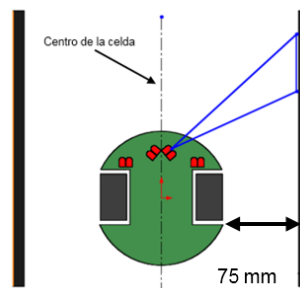


Figura 3.23 Desplazamiento con pared derecha.

En la Figura 3.24 se muestra el diagrama de flujo para la función avance con pared derecha. La función avanzar con pared izquierda es similar a la anterior pero en ésta se toma como referencia la pared izquierda.

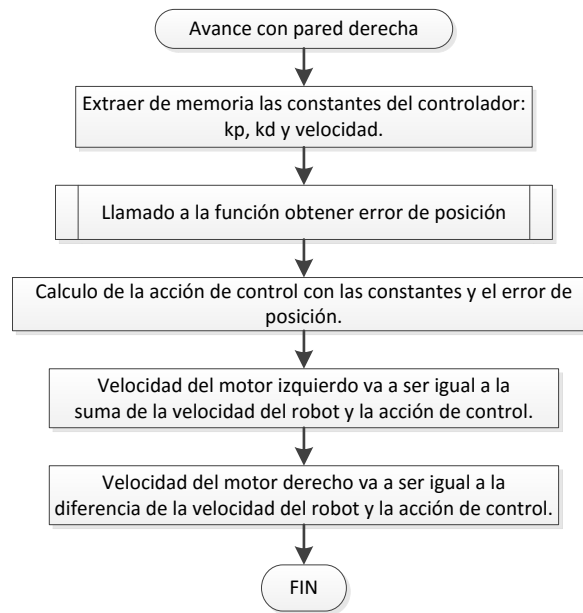


Figura 3.24 Diagrama de flujo de la función avance con pared derecha.

3.3.5.2 Función avanzar con dos paredes

Esta sirve para recorrer tramos del laberinto que tengan dos paredes laterales (Figura 3.25), lo que se hace es ejecutar un controlador proporcional derivativo, la entrada al controlador es el error de posición, el cual se calcula teniendo como referencia las paredes laterales, de acuerdo a la ecuación (3.22):

$$\text{error de posición} = \text{distancia derecha} - \text{distancia izquierda} \quad (3.22)$$

Con la acción de control resultante se setea las velocidades de los motores conforme las ecuaciones (3.23) y (3.24):

$$\text{motor}_{\text{izquierdo}} = \text{velocidad}_{\text{robot}} + \text{acción de control} \quad (3.23)$$

$$\text{motor}_{\text{derecho}} = \text{velocidad}_{\text{robot}} - \text{acción de control} \quad (3.24)$$

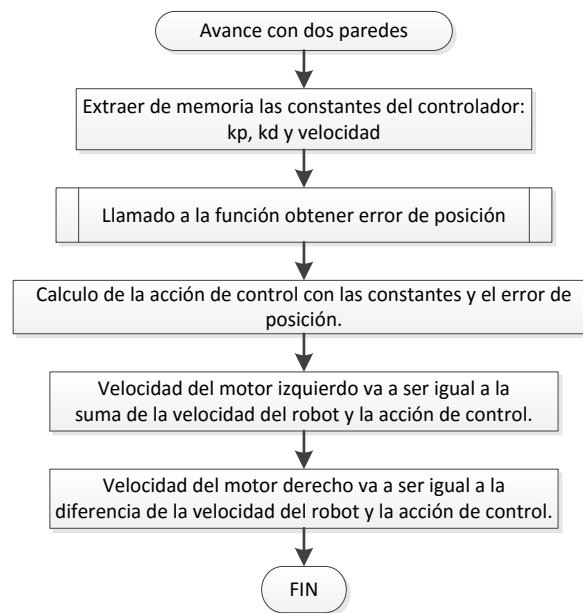


Figura 3.25 Diagrama de flujo de la función avance con dos paredes.

3.3.5.3 Función avanzar una celda

Esta función permite al robot avanzar en línea recta una distancia. Para realizar este desplazamiento se mide la distancia de cada rueda mientras el robot avanza a una velocidad constante, al llegar a la distancia deseada o al detectar una pared en frente se detienen los motores.

Para medir la distancia que recorre el robot se utiliza un encoder incremental, cuya resolución es de 24 pulsos por revolución cuando se cuentan los dos flancos de un canal. Para el cálculo de la distancia se lleva una cuenta por rueda para luego transformar ésta a unidades de distancia mm.

La distancia recorrida se calcula con la ecuación (3.25).

$$d_r = N_p * \frac{2 * \pi * R}{p} \quad (3.25)$$

Donde:

d_r : Distancia recorrida.

N_p : Número de pulsos.

R : Es el radio de la rueda = 20 mm.

p : Resolución del encoder = 24.

En la Figura 3.26 se observa el diagrama de flujo de la función avanzar una celda.

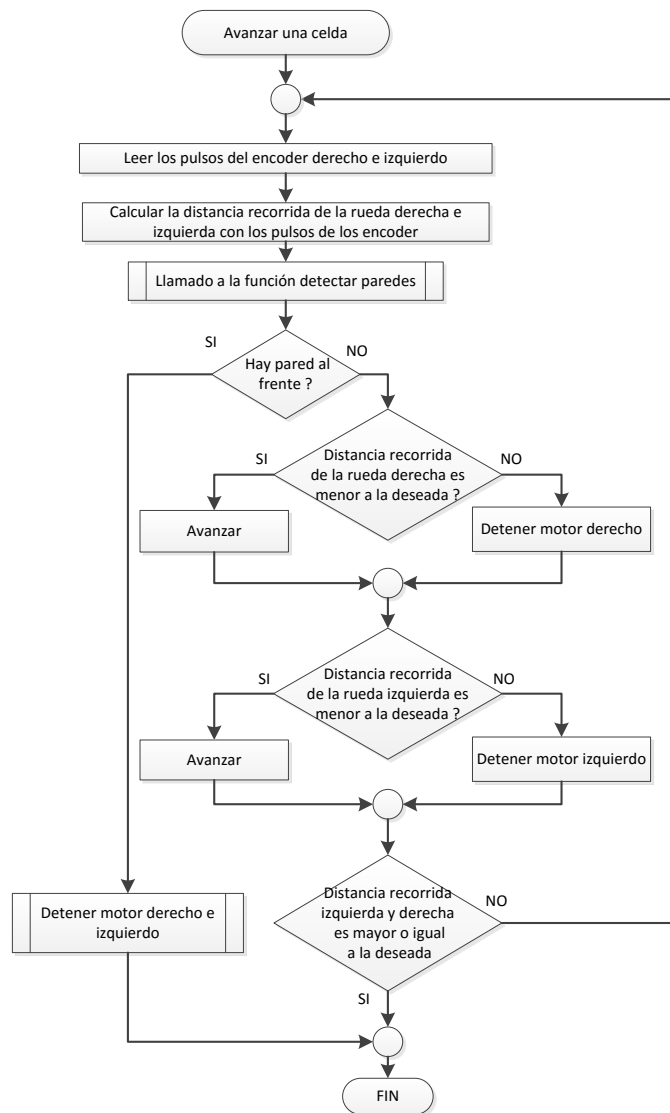


Figura 3.26 Diagrama de flujo de la función avanzar una celda.

3.3.5.4 Función girar a la derecha en su propio eje

Esta función permite al robot girar 90° a la derecha en su propio eje. Para realizar el giro se setea un valor de velocidad igual en los dos motores pero con sentido contrario. Mientras el robot gira se mide el ángulo, al llegar al ángulo de giro seteado (90°) los motores se detienen.

Para medir el ángulo de giro se usa los encoders, se lleva la cuenta del número de pulsos de cada rueda para luego transformar esta cuenta a grados de giro. Los grados que gira el robot se calcula con la ecuación (3.26).

$$G_g = N_p * \frac{360^\circ}{\frac{D_R * \pi}{d_{RP}}} \quad (3.26)$$

Donde:

G_g : Grados de giro.

N_p : Número de pulsos.

D_R : Distancia entre las ruedas = 80 mm.

d_{RP} : Distancia recorrida por pulso del encoder.

La distancia recorrida por pulso se calcula con la ecuación (3.27).

$$d_{RP} = \frac{2 * \pi * R}{p} \quad (3.27)$$

Donde:

R : Radio de la rueda.

p : Resolución del encoder.

En la Figura 3.27 se observa el algoritmo para girar a la derecha en su propio eje. La función de girar a la izquierda en su propio eje se realiza de manera similar considerando que los motores deben girar en sentido contrario.

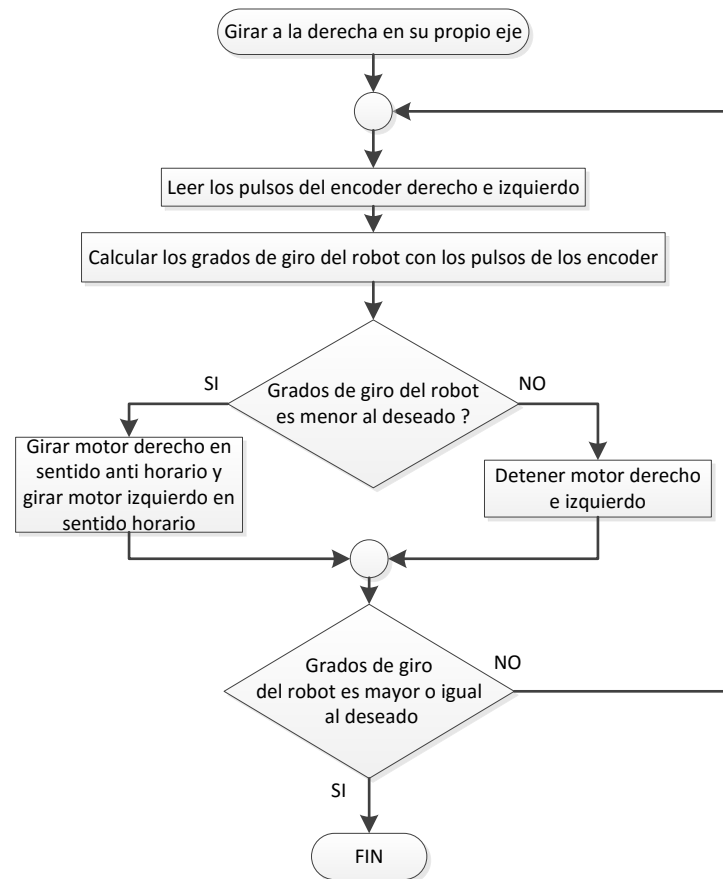


Figura 3.27 Diagrama de flujo de la función giro a la derecha en su propio eje.

3.3.5.5 Función giro 180° en su propio eje

Realiza un giro de 180° en su propio eje, el procedimiento para realizar el giro es similar al de la función girar a la derecha en su propio eje, teniendo en cuenta que se deberá cambiar ángulo de giro deseado a 180°.

3.3.5.6 Función girar a la derecha

La función girar a la derecha con un radio le permite al robot realizar un giro con un radio de curvatura igual a 125 mm lo que significa que el robot va a girar por el centro de la celda como se observa en la Figura 3.28. Para calcular las velocidades

que se necesita para realizar el giro con el radio de curvatura deseado se usa la ecuación (3.28).

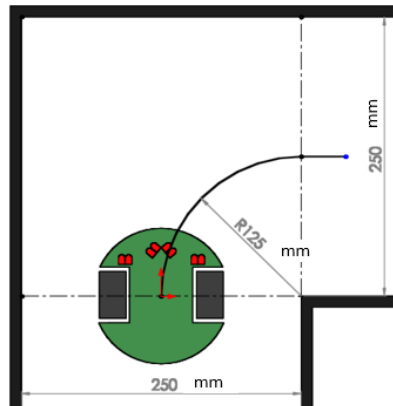


Figura 3.28 Curva de 90° con un radio de giro.

$$R = \frac{L}{2} * \left(\frac{V_L + V_R}{V_R - V_L} \right) \quad (3.28)$$

Donde:

R : Radio de curvatura = 125 mm.

V_L : Velocidad de la rueda izquierda.

V_R : Velocidad de la rueda derecha.

L : Distancia entre las ruedas = 80 mm.

Despejando V_R se tiene:

$$V_R = V_L * \left(\frac{R + \frac{L}{2}}{R - \frac{L}{2}} \right) \quad (3.29)$$

Para calcular la relación que deben tener las velocidades de las ruedas se reemplazan los datos y se obtiene:

$$V_R = V_L * \left(\frac{125 + \frac{80}{2}}{125 - \frac{80}{2}} \right) = 1.941 * V_L$$

$$V_R = 1.941 * V_L$$

De lo anterior, se tiene que para girar a la derecha la velocidad de la rueda izquierda debe ser 1.9141 veces la velocidad de la rueda derecha; para realizar un giro a la izquierda, la velocidad de la rueda derecha es 1.9141 veces la velocidad de la rueda izquierda. En la Figura 3.29 se observa el diagrama de flujo de la función girar a la derecha con un radio.



Figura 3.29 Diagrama de flujo de la función girar a la derecha con un radio.

3.3.5.7 Función posicionar con pared frontal

El objetivo de ésta función es que el robot siempre conserve una distancia con la pared frontal y que en caso de llegar con cierta desviación el robot corrija ese error y quede completamente paralelo a la pared (Figura 3.30).

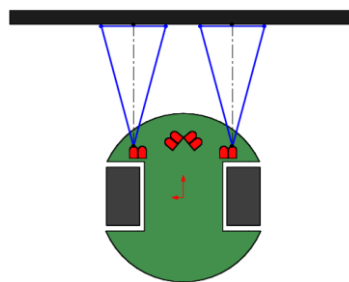


Figura 3.30 Posicionar el robot paralelo a la pared frontal.

Para que el robot este bien posicionado los sensores frontales deben medir la misma distancia, para esto se ejecuta un controlador PD para cada llanta durante un tiempo determinado, el sensor frontal derecho da la realimentación para

controlar la rueda derecha y el sensor frontal izquierdo da para la rueda izquierda. El error se calcula con las ecuaciones (3.30) y (3.31):

$$\text{error rueda derecha} = \text{set point} - \text{distancia sensor frontal derecho} \quad (3.30)$$

$$\text{error rueda izquierda} = \text{set point} - \text{distancia sensor frontal izquierdo} \quad (3.31)$$

El controlador de la rueda izquierda entrega una acción de control u_1 con el cual se setea la velocidad del motor izquierdo de acuerdo a la ecuación (3.32):

$$\text{velocidad izquierda} = -u_1(n) \quad (3.32)$$

Para la rueda derecha se tienen otro controlador, el cual proporciona una acción de control u_2 con la cual se setea la velocidad de la rueda derecha de acuerdo a la ecuación (3.33):

$$\text{velocidad derecha} = -u_2(n) \quad (3.33)$$

De lo dicho, cuando la parte izquierda del robot está muy cerca de la pared, el sensor frontal izquierdo medirá una distancia menor que el set point, por lo que el error va a ser positivo y la rueda izquierda va a tener que retroceder para alcanzar el set point. Cuando queda muy alejado de la pared, el error va a ser negativo y la rueda izquierda va a tener que avanzar para alcanzar el set point. El control para la rueda derecha funciona de igual manera.

El diagrama de flujo de la función se puede observar en la Figura 3.31.

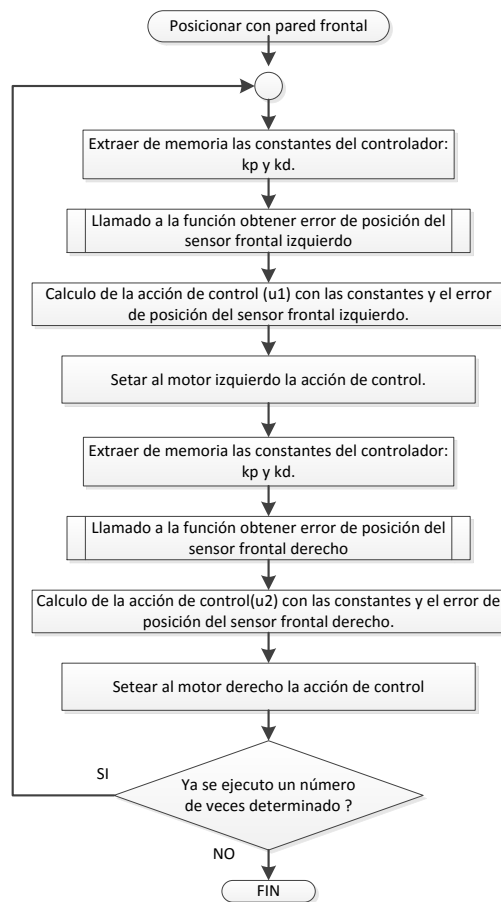


Figura 3.31 Diagrama de flujo de la función posicionar con pared frontal.

3.3.6 PROGRAMA PRINCIPAL

El programa principal se ha diseñado de tal manera que permita al usuario un fácil manejo e interacción con el robot y a la vez pueda cumplir el objetivo de resolver laberintos. Posee tres algoritmos de solución de laberinto: algoritmo de la mano derecha, algoritmo de la mano izquierda y algoritmo de relleno de callejones sin salida. Además tiene funciones que comprueban el funcionamiento correcto de los sensores, motores y encoders. Para la implementación se han utilizado todos los módulos y funciones que se describieron anteriormente. En la Figura 3.32 se puede observar el diagrama de flujo del programa principal.

3.3.6.1 Algoritmo de la mano derecha o izquierda [37]

Este algoritmo realiza el seguimiento de una pared ya sea ésta derecha o izquierda dependiendo de lo escogido. Para el algoritmo de la mano derecha siempre seguirá la pared derecha mientras la tenga, si no existe la pared derecha el robot girará a la derecha (Figura 3.33). Se debe realizar dicho proceso de forma recursiva hasta solucionar el laberinto.

El algoritmo de la mano izquierda funciona de manera similar pero usando la pared izquierda. Estos algoritmos son los más utilizados para resolver laberintos que tengan la entrada y salida en las paredes externas y no posean islas en su interior. En la Figura 3.34 se muestra el diagrama de flujo del algoritmo de la mano izquierda.

3.3.6.2 Algoritmo de relleno de callejones sin salida [37]

El algoritmo tiene como objetivo rellenar todos los callejones que lleven al robot a un camino que provoque un giro de 180° , se realiza un primer recorrido con un algoritmo de detección de caminos sin salida (Figura 3.35), en el cual se va guardando en un vector las intersecciones que tengan dos o más posibles caminos y los caminos sin salida que existan en el recorrido.

Después de obtener este vector se usa un algoritmo de optimización del camino (Figura 3.36), el cual devuelve un vector en el que están eliminados todos los caminos sin salida; al obtener el vector optimizado, la segunda vez que el robot recorra el laberinto se usa la rutina de resolución del laberinto evitando caminos sin salida (Figura 3.37), cada que llegue a una intersección el robot revisa en el vector de memoria que giro debe realizar y así evita entrar innecesariamente a caminos ayudando a disminuir el tiempo de resolución del laberinto.

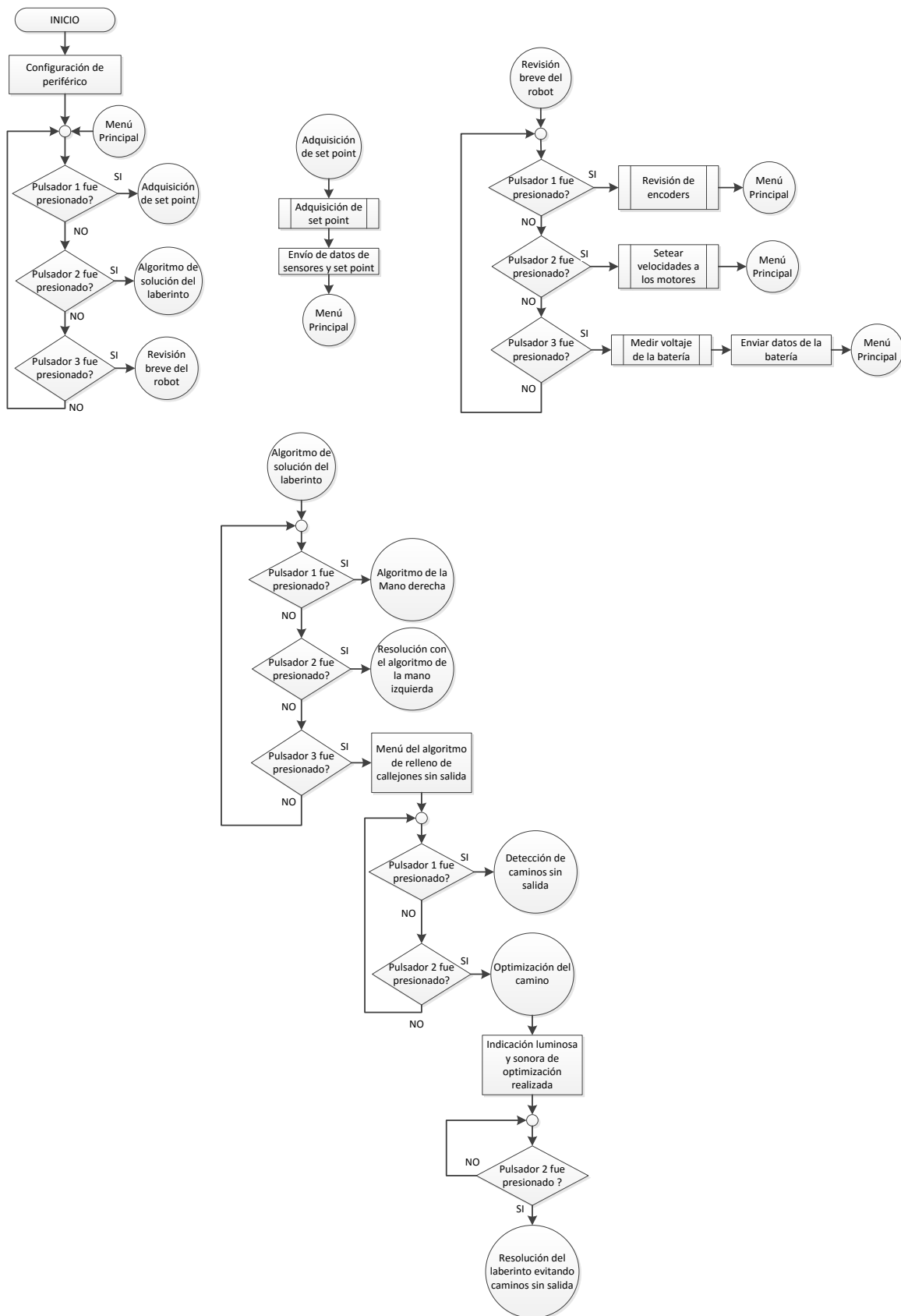


Figura 3.32 Diagrama de flujo del programa principal.

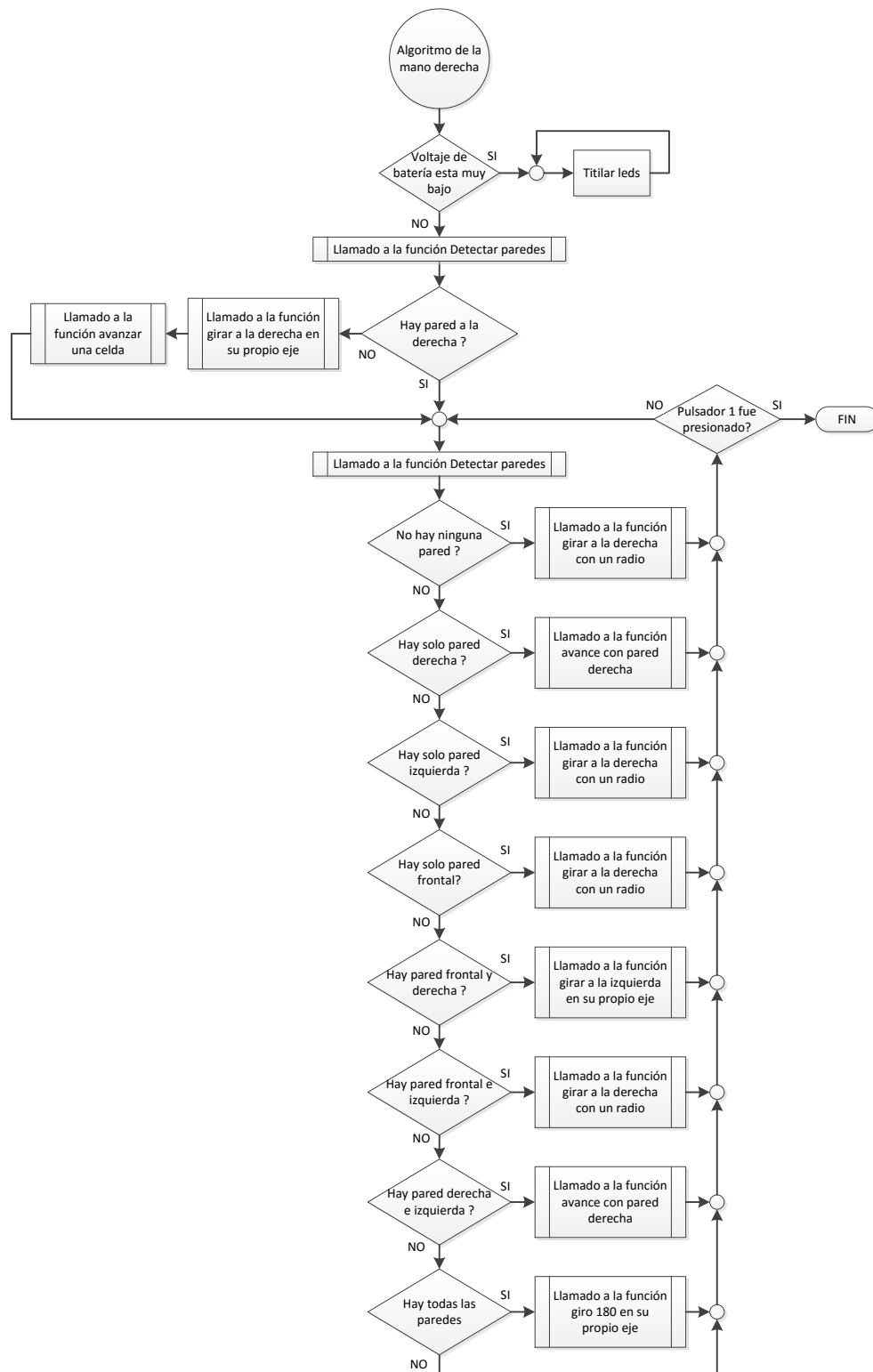


Figura 3.33 Diagrama de flujo del algoritmo de la mano derecha.

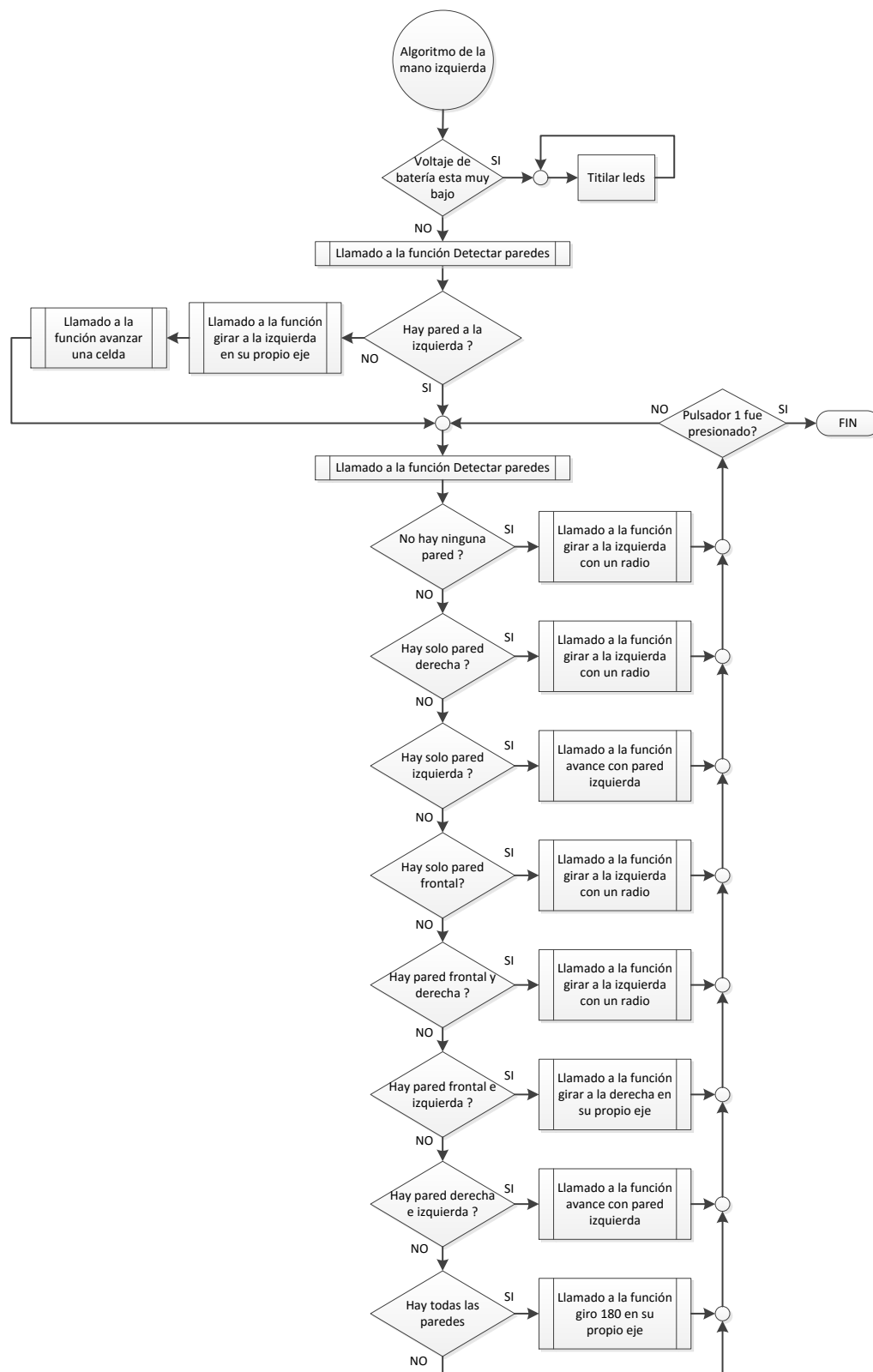


Figura 3.34 Diagrama de flujo del algoritmo de la mano izquierda.

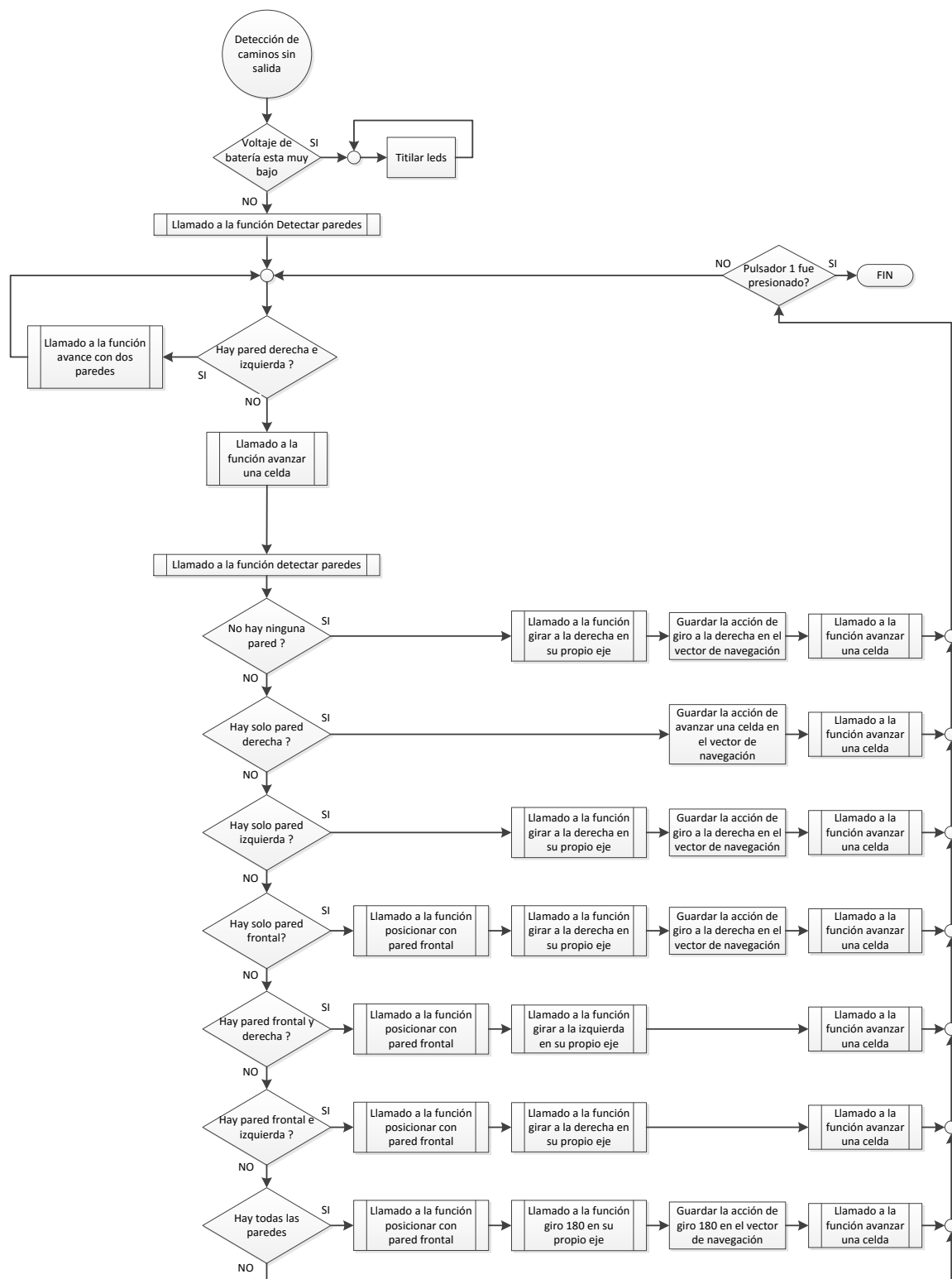


Figura 3.35 Diagrama de flujo del algoritmo de detección de caminos sin salida.

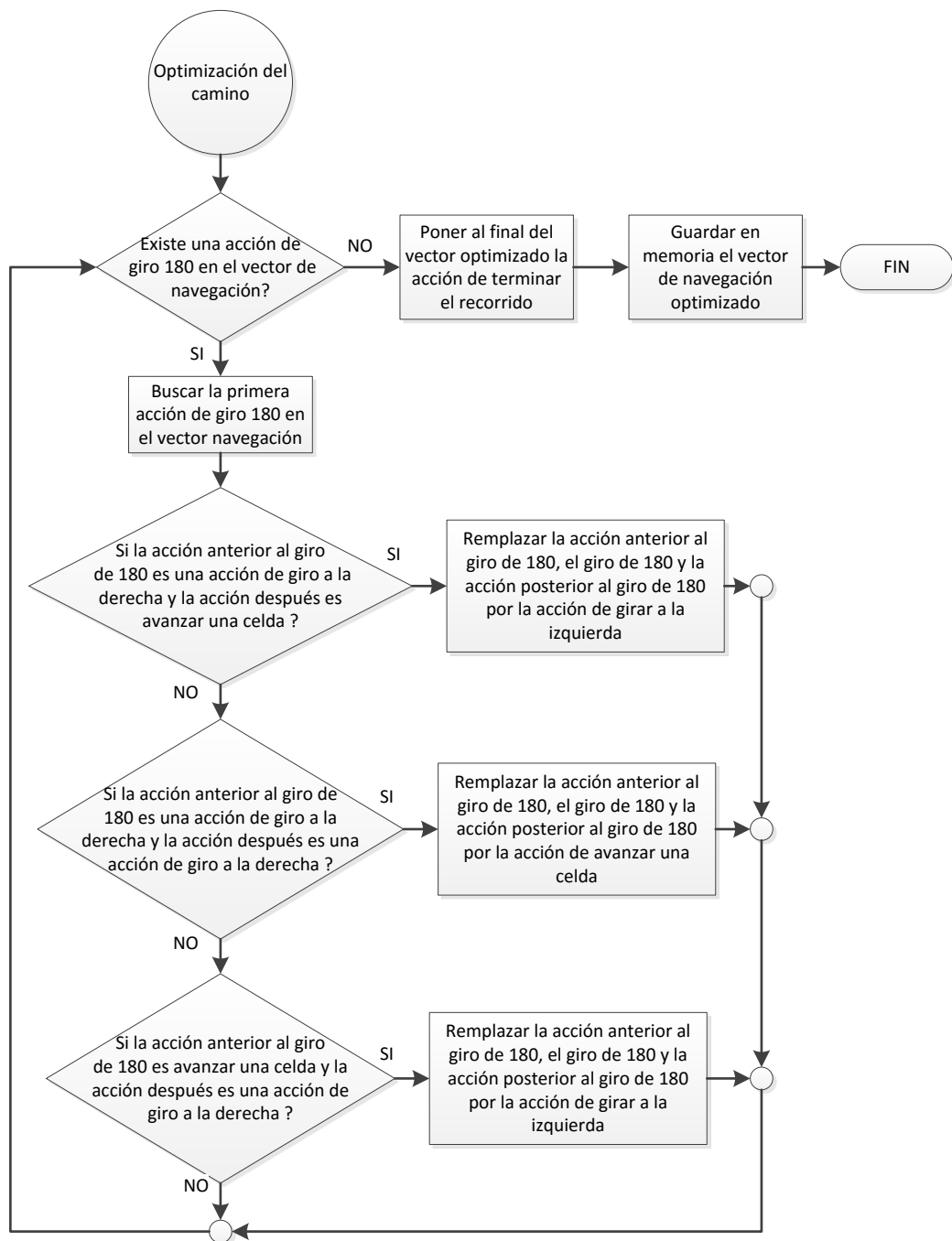


Figura 3.36 Diagrama de flujo de la función optimización del camino.

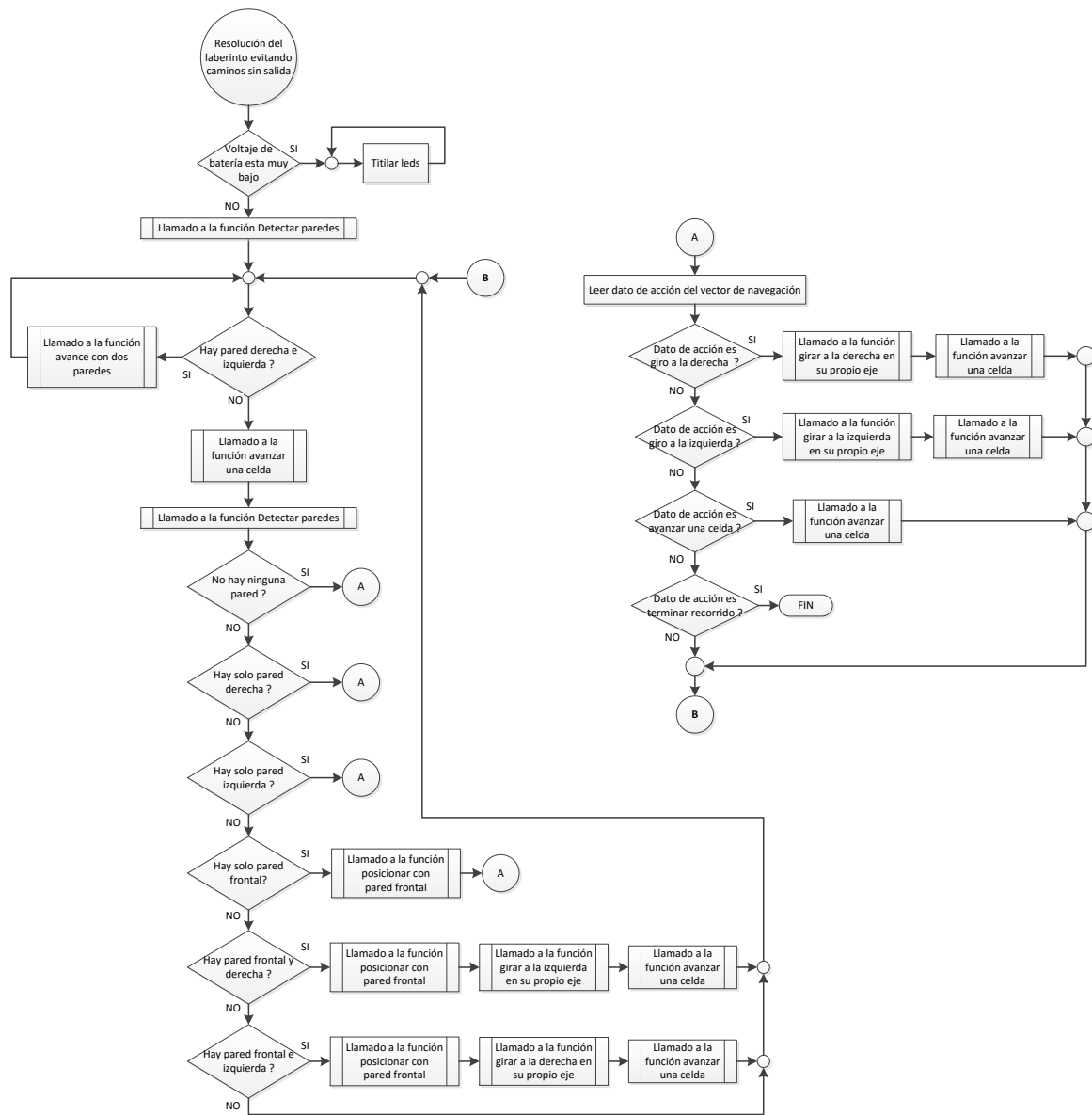


Figura 3.37 Diagrama de flujo de la función recorrido con memoria.

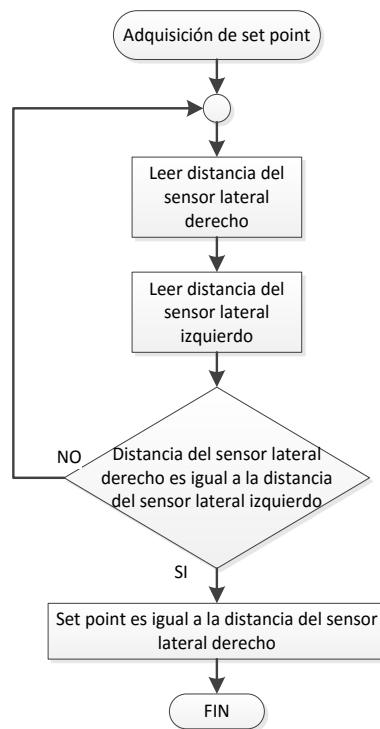


Figura 3.38 Diagrama de flujo de la función adquisición de set point.

CAPITULO 4

PRUEBAS Y RESULTADOS

En este capítulo se presenta las pruebas realizadas en los prototipos que indican el correcto funcionamiento de ellos. Como parte de las pruebas, también se resume la participación de los prototipos en varios torneos y los resultados obtenidos.

4.1 PRUEBAS EN EL ROBOT SEGUIDOR DE LÍNEA

Al robot seguidor de línea se le hacen las pruebas en el hardware, en el software y su funcionamiento en conjunto (pruebas en pista).

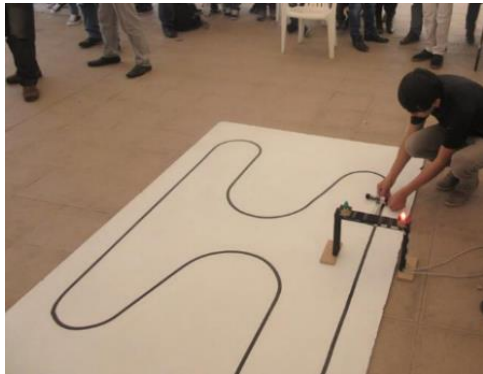
4.1.1 PRUEBAS DE HARDWARE

Esta parte contempla la electrónica del prototipo, la parte mecánica se la trata en las pruebas en pista; se realizaron varios cambios en el diseño de la placa de control especialmente en los encapsulados de los elementos ya que de las pruebas realizadas en prototipos anteriores la tarjeta era demasiado grande y pesada lo cual incidía negativamente en el desempeño de los prototipos (Figura 4.1), por lo que en el diseño final todos los elementos de la placa de control son de montaje superficial además se agregó un conector para comunicación.

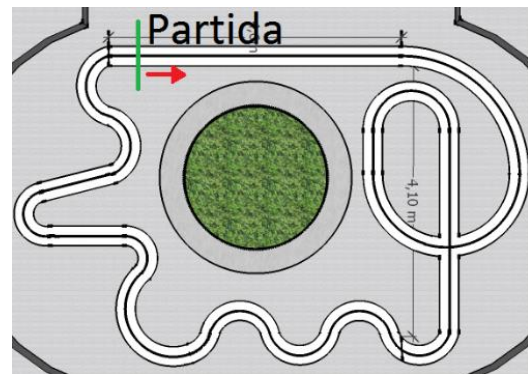


Figura 4.1 Prototipos anteriores, los robots eran muy pesados y tenían problemas de adherencia.

De la experiencia obtenida en la participación de torneos se le incorporó al diseño de los prototipos encoders para la medición de distancia, el objetivo es que el robot en el arranque vaya a una velocidad mayor que en el resto del recorrido con lo cual se optimizaría varias décimas de segundo (dependiendo de la pista), ya que por ejemplo en varios de los torneos asistidos en el arranque se tenía rectas largas y en algunos casos seguidas de curvas con radio muy grande (Figura 4.2).



(a)



(b)

Figura 4.2 Pistas de competencias. (a) Pista UMEBOT X. (b) Pista Torneo Universidad de Nariño.

4.1.2 PRUEBAS DE SOFTWARE

Conforme se iba avanzando y mejorando el hardware, el software cambiaba para adaptarse. La primera mejora sustancial que se le hizo al programa fue en la forma de leer los sensores de línea, al inicio se los leía en forma digital en donde el sensor arrojaba un uno lógico si se encontraba sobre la línea y cero lógico si se encontraba fuera de ésta, con lo cual se tenía que programar todas las combinaciones para obtener el error. Para programar las combinaciones se debía tener en cuenta el ancho de la línea y la separación entre sensores.

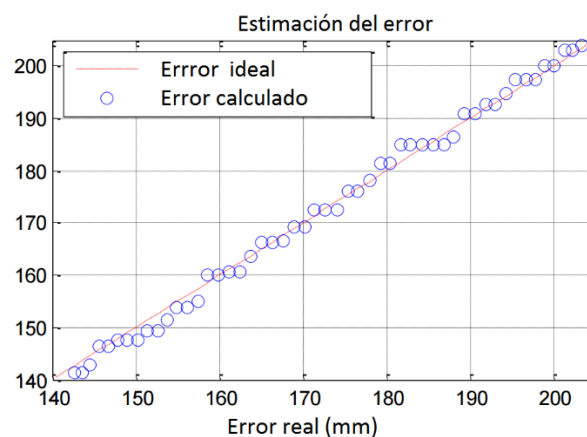
En la Tabla 4.1 se puede observar las combinaciones posibles para 5 sensores con un ancho de pista de 19 mm y una separación entre sensores de 9 mm.

Tabla 4.1 Estimación del error en forma digital.

S0	S1	S2	S3	S4	Error
1	0	0	0	0	-4
1	1	0	0	0	-3
0	1	0	0	0	-2
0	1	1	0	0	-1
0	0	1	0	0	0
0	0	1	1	0	1
0	0	0	1	0	2
0	0	0	1	1	3
0	0	0	0	1	4

El inconveniente que se tiene al leer los sensores de ésta manera es que el error tiene saltos muy grandes lo cual no es deseable para el controlador, ya que esto contribuirá a que el robot tienda a oscilar.

Considerando el inconveniente que tiene el método anterior se mejoró el algoritmo para la obtención del error, ahora a los sensores se los lee en forma analógica y se calcula el error por medio de un promedio ponderado. Existen varios métodos más para el cálculo, como: interpolación lineal e interpolación cuadrática pero el promedio ponderado es la mejor opción ya que su estimación es la que menor error tiene [38]. En la Figura 4.3 se puede observar el error real de posición y la estimación mediante el algoritmo del promedio ponderado.

**Figura 4.3** Estimación del error por promedio ponderado.

Otro avance importante en el control del robot fue la adición del control en lazo abierto cuando el arreglo de sensores pierde completamente la línea, ya que el

robot solo con el control PID le costaba tomar curvas de radio pequeño (curvas de radio menor a 30 cm), y con éste control se pudo solucionar este problema.

También se fue mejorándolo en cuanto a su funcionalidad, ya que por ejemplo las primeras versiones para poder realizar ajustes en el controlador se necesitaban volver a programar el microcontrolador, ahora se lo hace por medio de comunicación serial.

En la Figura 4.4 se puede observar el envío de las constantes del controlador mediante Hercules setup utility el cual es un programa de libre acceso con el que se realiza la comunicación serial.

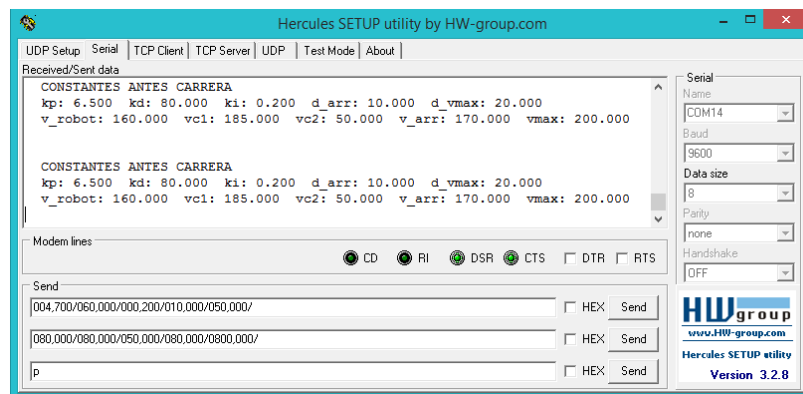


Figura 4.4 Envío de constantes al robot.

4.1.3 PRUEBAS EN PISTA

Una vez que toda la electrónica está funcionando perfectamente y que el programa que controla al robot no tiene problemas se procedió a probarlo en pista, para lo cual se tomó como referencia la pista del Concurso Ecuatoriano de Robótica 2015 (Figura 4.5). La pista es completamente plana, no tiene intersecciones ni discontinuidades, está hecha con cinta aislante negra en un tablero triplex de color blanco.

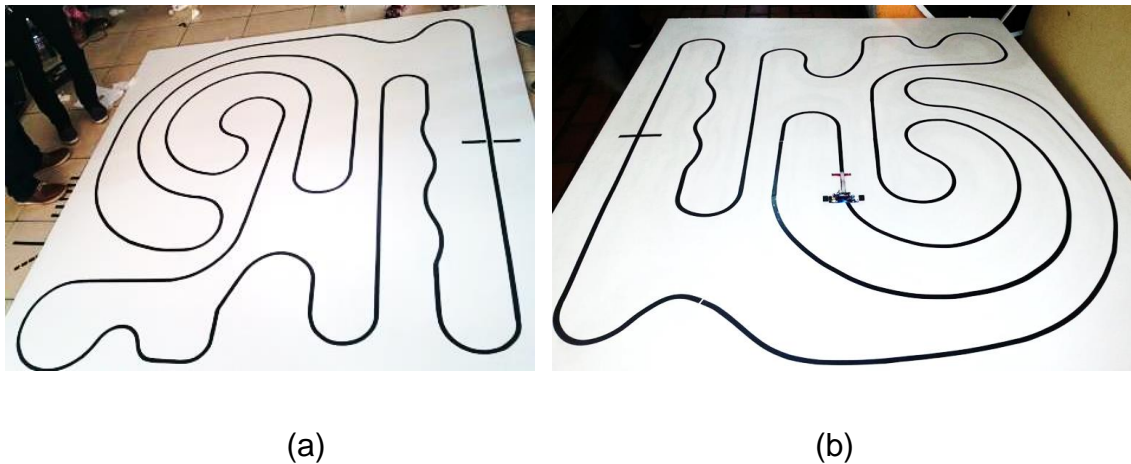


Figura 4.5 Pista del seguidor de línea. (a) Pista del CER 2015. (b) Pista recreada para pruebas.

El desempeño del robot se lo evalúa por medio del tiempo que le toma recorrer la pista (el objetivo es recorrer la pista en el menor tiempo posible), el robot no debe salirse de la pista ni acortar camino (el reglamento no lo permite).

Lo primero que se probó fueron las ruedas, se probó con algunos materiales como: caucho, silicona, goma y espuma. Las pruebas arrojaron que las ruedas de goma, silicona y caucho (Figura 4.6) no son las adecuadas ya que éstas no dan suficiente adherencia al prototipo, no importa los ajustes que se le haga al controlador, no es posible incrementar la velocidad del robot ya que éste derrapa fácilmente y pierde pista, otra desventaja que tienen este tipo de ruedas es que les afecta mucho la suciedad (polvo) que puede estar en la pista por lo que hay que limpiarla continuamente.



Figura 4.6 Ruedas. (a) Ruedas de silicona. (b) Ruedas de goma.

Las mejores ruedas para el prototipo fueron las de espuma (Figura 4.7), éstas dieron la suficiente adherencia para poder incrementar la velocidad del prototipo además que a éstas no les afecta demasiado la suciedad que pueda haber en la pista.



Figura 4.7 Ruedas de espuma.

El otro factor que tiene un papel importante en el desempeño del robot es la distancia entre los sensores y el eje de las ruedas, se calibró y probó al prototipo con diferentes distancias, distancias que van desde los 5 cm hasta los 16.5 cm (éstas pruebas se hicieron con las ruedas de espuma). La distancia con la que mejores resultados se obtuvo fue con 15 cm (Figura 4.8).

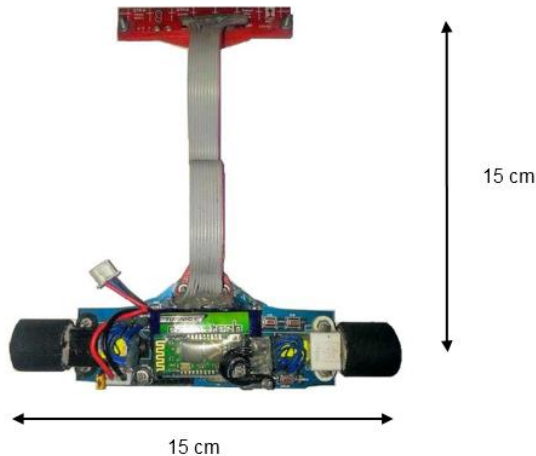


Figura 4.8 Distancia entre sensores y ejes de las ruedas.

Para distancias mayores, el prototipo le cuesta girar en las curvas con lo cual se pierde algunas décimas de segundo. Para distancias menores los tiempos no fueron mejores aunque a simple vista pareciera que el robot fuera más rápido. En

la Tabla 4.2 se presenta los resultados obtenidos para algunas distancias del eje de las ruedas a los sensores. El objetivo para el robot seguidor de línea es superar el tiempo del ganador del Concurso Ecuatoriano de Robótica 2015 el cual recorrió la pista en 11.4 segundos y de las pruebas realizadas al prototipo se puede afirmar que se cumplió el objetivo.

Tabla 4.2 Tiempos obtenidos con diferentes distancias de los sensores al eje de las ruedas.

Distancia de los sensores al eje de las ruedas	Control PID	Controlador Borroso
9 cm	11.2 s	18.2 s
13.5 cm	10.9 s	17.9 s
15 cm	10.6 s	16,4 s
16.5 cm	10.8 s	17.3 s

De la Tabla 4.2 se puede apreciar que el controlador con el que mejores resultados se obtuvo fue el PID ya que los tiempos en recorrer la pista son menores.

4.2 RESULTADOS DEL ROBOT SEGUIDOR DE LÍNEA

Como parte de las pruebas en el desarrollo de este proyecto se ha participado en varios torneos, tanto a nivel nacional como internacional, en la mayoría de ellos se obtuvo buenos resultados. Se participó en:

- El Torneo de Robótica IEEE Universidad de Nariño en la ciudad de San Juan de Pasto, 4 de noviembre del 2015, con dos prototipos los cuales fueron eliminados en segunda ronda.
- Torneo de robótica UMEBOT X en la Escuela Politécnica Nacional, 2 de diciembre del 2015, con dos prototipos, uno de ellos obtuvo el segundo lugar.
- Concurso Ecuatoriano de Robótica, 9 de diciembre del 2015, con 2 prototipos, uno de ellos se obtuvo el tercer lugar.
- Torneo Innova Week en la ciudad de Quito, 26 de marzo del 2016, con dos prototipos y se obtuvo el primer y segundo lugar.

- Torneo UTNATION en la Universidad Técnica de Norte, 28 de abril del 2016, se obtuvo el primer lugar.

4.3 CARACTERÍSTICAS DEL ROBOT SEGUIDOR DE LÍNEA

En la Figura 4.9 se muestra el diseño final del robot seguidor de línea.

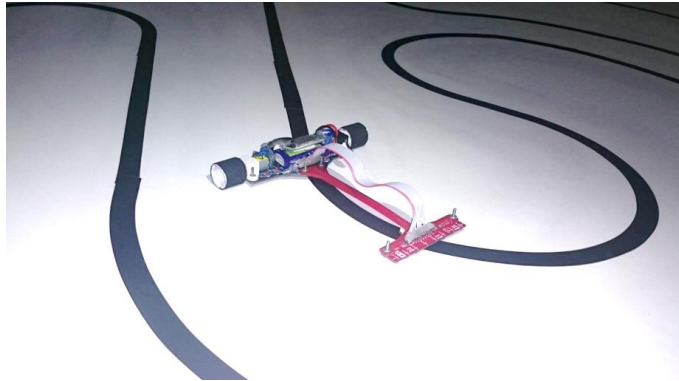


Figura 4.9 Robot seguidor de línea.

Características:

- Largo: 16.4 cm.
- Ancho: 15 cm.
- Alto: 2.5 cm.
- Peso: 170 g.
- La configuración del robot es tipo tracción diferencial.
- Alimentación: Una batería de corriente continua.
- En la parte sensorial: tiene un arreglo de ocho sensores infrarrojos para la detección de la línea y un par de encoders magnéticos instalados uno en cada motor para medir la distancia que el robot avanza.
- El cerebro del robot es un microcontrolador ATmega 324PA.
- Para la interacción con el robot se tiene cuatro pulsadores y cuatro leds; además tiene un conector para comunicación serial.

4.4 PRUEBAS EN EL ROBOT LABERINTO

Al robot laberinto se le hacen las pruebas en el hardware, en software y su funcionamiento en conjunto (pruebas en pista).

4.4.1 PRUEBAS DE HARDWARE

Implica comprobar el funcionamiento mecánico y de los circuitos diseñados. Todos los circuitos necesarios para el funcionamiento se los ha diseñado en una sola placa que a la vez servirá de chasis para el robot. En la placa se tiene el circuito de la alimentación, microcontrolador, sensores de pared, pulsadores, leds, etc. Al armar el robot no se tuvo ningún inconveniente con la placa de circuitos, todo funcionó de acuerdo a lo diseñado y no se realizó ningún cambio.

Al inicio se optó por diseñar los encoders y acoplarlos a las llantas pero esto dio problemas, las señales que daban los encoders eran malas, les afectaba la luz ambiental por esta razón se decidió usar encoders comerciales que tenían ya sus propias ruedas y aseguraban un mejor funcionamiento.

También se probó el funcionamiento del robot con los sensores laterales perpendiculares a la pared pero se tuvo problemas con el desplazamiento debido a que con la medida que éstos proporcionaban, el control no sabía la dirección de la desviación del robot lo cual no permitía incrementar la velocidad de navegación en el laberinto.

Es por esto que se colocaron los sensores con un ángulo de 34° ya que de ésta manera la medida tiene más significado para el control del robot, el robot ya sabe si se está acercando o alejando de la pared con lo cual se pudo incrementar la velocidad de navegación del robot.

Después de realizar algunas pruebas de desplazamiento del robot, se notó que las llantas al ser de caucho y tener un labrado no poseían la adherencia necesaria, el robot derrapaba muy fácilmente dando problemas especialmente en los giros, por esta razón se cambió el recubrimiento de la llanta por espuma con lo cual se solucionó el problema (Figura 4.10).

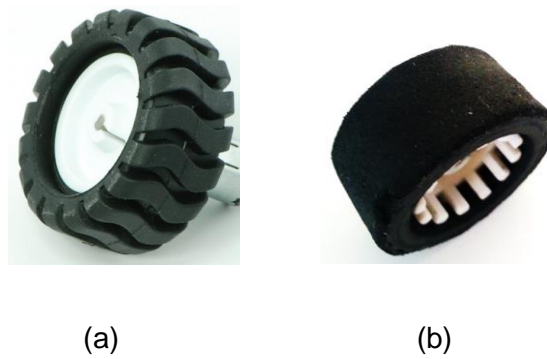


Figura 4.10 Ruedas. (a) Rueda original. (b) Rueda modificada.

4.4.2 PRUEBAS DE SOFTWARE

En el diseño del programa principal y el módulo de control se realizaron múltiples cambios y mejoras al algoritmo hasta obtener un mejor funcionamiento. De esta manera se pudo conseguir que el seguimiento de las paredes del laberinto y los giros sean los adecuados para los algoritmos de solución implementados, evitando choques con las paredes del laberinto.

4.4.3 PRUEBAS EN PISTA

Una vez construido el prototipo y haber probado el correcto funcionamiento de hardware y software, se llevó a cabo las pruebas en pista para comprobar el desempeño del robot en un laberinto.

Las pruebas se realizaron en base a la pista del Concurso Ecuatoriano de Robótica 2015 (Figura 4.11). El objetivo para el robot es solucionar un laberinto en el menor tiempo posible sin topar las paredes, ya que recibiría una penalización de acuerdo al reglamento. La dimensión del laberinto es de 2 x 2 m y una distancia entre paredes de 25 cm. El ganador del Concurso Ecuatoriano de Robótica 2015 resolvió el laberinto en un tiempo de 30 segundos. Para las pruebas primero se usó el algoritmo de la mano derecha. Con este algoritmo el robot consiguió superar el laberinto en un tiempo aproximado de 25 s, es posible bajar este tiempo incrementando la velocidad de navegación pero al hacerlo la probabilidad de choque aumenta lo cual no es deseado ya que de acuerdo al reglamento se recibe una penalización.

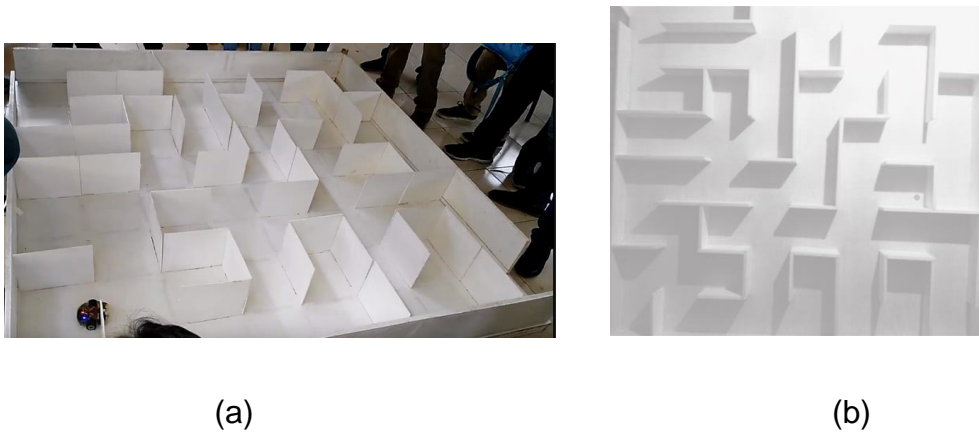


Figura 4.11 Pista de robot laberinto. (a) Pista del CER 2015. (b) Pista recreada para pruebas.

Después se usó el algoritmo de la mano izquierda, con lo cual el robot consiguió solucionar el laberinto en un tiempo de 21 s. Aunque el algoritmo anteriormente probado funciona de forma similar se tiene resultados diferentes en tiempo debido a que la distancia que recorre el robot para resolver el laberinto es menor por uno de ellos.

Finalmente con el algoritmo de relleno de callejones sin salida se tiene que recorrer el laberinto dos veces, debido a que la primera vez el robot memoriza las intersecciones y caminos sin salida que existan en su recorrido, la segunda vez que lo recorre evita entrar a esos caminos sin salida para poder resolverlo en un menor tiempo. Con este método se obtuvo un tiempo de 35 s en el primer recorrido, después, al realizar el segundo recorrido (sin recorrer callejones sin salida) se obtuvo un tiempo de 26 s. En la Tabla 4.3 se muestra la comparación en tiempo que se demora en solucionar el laberinto con los algoritmos implementados. Se debe tener en cuenta que el tiempo que se demora en solucionar el laberinto por los diferentes algoritmos dependerá del diseño de éste.

Tabla 4.3 Comparación en tiempos de los diferentes algoritmos de resolución.

Algoritmos de resolución	Tiempo
Algoritmo de la mano derecha	25 s
Algoritmo de la mano izquierda	21 s
Algoritmo de relleno de caminos sin salida	26 s

4.5 RESULTADOS ROBOT LABERINTO

El prototipo ha participado en algunos concursos en los cuales se han obtenido buenos resultados y también se ha ganado experiencia para realizar mejoras en el mismo. Se participó en:

- Torneo de robótica UMEBOT X que se llevó acabo el 2 de diciembre del 2015 en Escuela Politécnica Nacional en el cual el prototipo fue eliminado.
- Concurso Ecuatoriano de Robótica el 9 de diciembre del 2015, el prototipo obtuvo el primer lugar.

Después de realizar todos los cambios necesarios en software, y haber realizado las pruebas necesarias, se puede decir que los objetivos planteados al inicio de este proyecto de titulación fueron conseguidos satisfactoriamente, obteniendo un buen funcionamiento del robot con todos los algoritmos de resolución implementados.

4.6 CARACTERÍSTICAS DEL ROBOT LABERINTO

En la Figura 4.12 se puede ver al robot laberinto terminado. Las características principales del robot se las detallan a continuación.

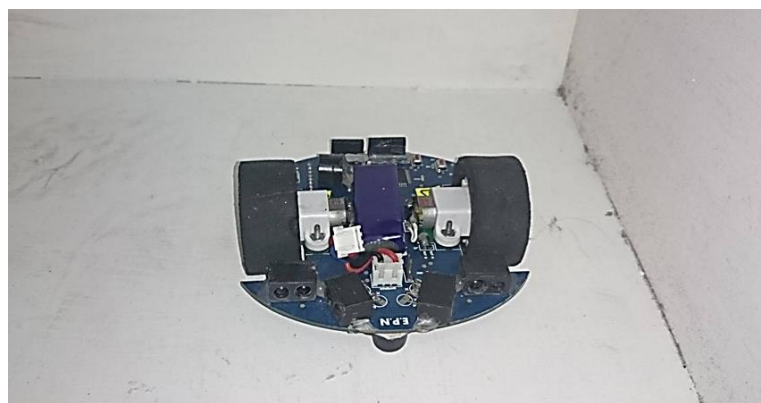


Figura 4.12 Robot laberinto.

Características:

- Largo: 10 cm.

- Ancho: 10 cm.
- Alto: 4 cm.
- Peso: 200 g.
- Tracción diferencial, la cual permite realizar giros en su propio eje.
- Alimentación: Una batería de corriente continua.
- Posee cuatro sensores infrarrojos. Dos para la detección de paredes laterales y dos para la detección de la pared frontal. Con un rango de medición de distancia máxima de cada sensor de 300 mm.
- Incorpora un encoder en cada rueda el cual le permite avanzar distancias o realizar giros con mayor precisión.
- El cerebro del robot es un microcontrolador ATmega 324PA.
- Para la interacción con el usuario tiene cuatro leds, cuatro pulsadores y un buzzer; además posee un conector para la comunicación serial.

4.7 COSTOS DE IMPLEMENTACIÓN DE LOS ROBOTS

Para calcular el costo de la implementación de los robots de competencia, primero se presenta una lista de los materiales necesarios para la construcción de los prototipos (Tabla 4.4 y Tabla 4.6).

Para el cálculo también se tiene en cuenta el costo de ingeniería en cual está incluido el diseño, construcción y desarrollo de los programas de control. Considerando que el proyecto fue planificado para 6 meses y que se laboraron 40 horas a la semana y un costo por hora de \$6 USD se calcula el costo de ingeniería en \$5760 USD, pero éste valor representa el costo de la construcción de los dos robots por lo que hay que dividir éste en cada robot.

En el robot seguidor de línea se utilizó alrededor del 60% del tiempo por lo que su costo de ingeniería es de \$3456 USD y el costo del robot laberinto es de \$2304 USD.

Tabla 4.4 Costos de materiales robot seguidor de línea.

Descripción	Cantidad	Valor USD
Tarjeta principal	1	\$ 15,00
Microcontrolador ATmega 324PA	1	\$ 6,00
Motores HP 10:1	2	\$ 50,00
Soporte de motores	1	\$ 7,00
Llantas	2	\$ 15,00
Encoder	2	\$ 10,00
Resistencias	7	\$ 0,21
Capacitores	18	\$ 3,60
Bobinas	2	\$ 0,20
Leds	4	\$ 1,00
Cristal	1	\$ 0,80
Pulsadores	4	\$ 1,20
Sensor de línea	1	\$ 15,00
Controlador de motores TB6612FNG	1	\$ 6,00
Regulador 7805	1	\$ 1,50
Conectores	2	\$ 1,00
Batería 7.4 V	1	\$ 15,00
Soporte impreso	1	\$ 10,00
Estaño		\$ 2,00
Pomada para soldar		\$ 3,00
Varios		\$ 10,00
	Total	\$ 173,51

Para calcular el costo total del robot seguidor de línea se suma el costo de materiales y el costo de ingeniería (Tabla 4.5).

Tabla 4.5 Costo total robot seguidor de línea.

Descripción	Cantidad	Valor USD
Materiales	1	\$ 173.51
Ingeniería	1	\$ 3456
	Total	\$ 3629.51

Tabla 4.6 Costos de materiales robot laberinto.

Descripción	Cantidad	Valor USD
Tarjeta principal	1	\$ 15,00
Microcontrolador ATmega 324PA	1	\$ 6,00
Motores HP 50:1	2	\$ 50,00
Soporte de motores	1	\$ 7,00
Llantas	2	\$ 15,00
Encoder	2	\$ 30,00
Resistencias	20	\$ 0,60
Capacitores	18	\$ 3,60
Bobinas	2	\$ 0,20
Leds	4	\$ 1,00
Cristal	1	\$ 0,80
Pulsadores	4	\$ 1,20
Transistor 2N3904	1	\$ 0,10
Transistor MMBT2222	4	\$ 0,80
Sensor infrarrojo emisor y receptor	4	\$ 6,00
Controlador de potencia TB6612FNG	1	\$ 6,00
Regulador 7805	2	\$ 3,00
Conectores	2	\$ 1,00
Buzzer	1	\$ 0,50
Batería 7.4 V	1	\$ 15,00
Ruedas locas	2	\$ 7,00
Estaño		\$ 2,00
Pomada para soldar		\$ 3,00
Varios		\$ 10,00
	Total	\$ 184,80

Para calcular el costo total del robot laberinto, se suma el costo de materiales con el costo de ingeniería (Tabla 4.7).

Tabla 4.7 Costo total robot laberinto.

Descripción	Cantidad	Valor USD
Materiales	1	\$ 184,80
Ingeniería	1	\$ 2304
	Total	\$ 2488.8

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

En este capítulo se menciona las conclusiones a las que se ha llegado con el presente proyecto de titulación, que son resultado de la implementación de múltiples prototipos antes de llegar al diseño final y de lo aprendido en las competencias en las que se ha participado. Se realizan recomendaciones que servirán como aporte para futuras investigaciones y el continuo mejoramiento de los robots.

5.1 CONCLUSIONES

- El controlador que mejores resultados dio en el robot seguidor de línea fue el PID ya que la puesta a punto del robot es más rápida frente al controlador borroso lo cual fue fundamental en los torneos de robótica ya que el tiempo que se tiene para calibraciones es limitado. Al PID se lo calibra con las constantes proporcional, integral y derivativa en contraparte del controlador borroso en donde no hay constantes para calibrar, en éste se debe cambiar las reglas de inferencia o las funciones de membresía lo cual conlleva más tiempo.
- La implementación del control en lazo abierto cuando el arreglo de sensores pierde completamente la línea permitió al robot mejorar su rendimiento ya que el robot pudo tomar las curvas de radio pequeño (radio menor a 30 cm) de mejor manera y así disminuir el tiempo en recorrer la pista.
- Para el cálculo del error de posición se utilizó un promedio ponderado ya que este es el algoritmo que menor error de estimación presenta lo cual es fundamental ya que el robot seguidor podrá seguir la línea de mejor manera.
- Se mejoró la lectura del error de posición del robot seguidor con respecto a la línea mediante la calibración de los sensores haciendo que éstos se comporten de manera similar y se adapten a las condiciones de luz del lugar.

- Una mala elección de las ruedas en el robot seguidor de línea hará bajar el rendimiento del prototipo sin importar los ajustes que se le hagan al controlador, el tiempo en el cual el prototipo supera la pista va a ser alto ya que la velocidad del robot debe ser baja para que no derrape y pierda pista.
- Se mejoró la lectura de distancia de los sensores infrarrojos del robot laberinto por medio de un algoritmo que permite eliminar en un gran porcentaje la luz ambiental, ya que ésta tiene componentes infrarrojos que afectan al correcto funcionamiento de los sensores.
- La ubicación de los sensores laterales de pared es de suma importancia ya que influye en la rapidez con la que el robot navega por el laberinto, en éste trabajo se ubicó a los sensores con un ángulo de 34° , de tal forma que le permita saber al controlador si el robot se está acercando o alejando de la pared.
- Con respecto a la navegación del robot laberinto es necesario realizarla por el centro de la celda, minimizando la probabilidad de choque con las paredes, para esto se implementó un controlador que con el cual se obtuvo una navegación correcta sin ningún choque.
- Los tres algoritmos implementados se basan en el seguimiento de paredes, con los cuales se lograron resultados favorables para la solución de laberintos con las especificaciones del concurso ecuatoriano de robótica.
- El tiempo de resolución de un laberinto siempre dependerá del diseño del laberinto de cada concurso, ya que cada algoritmo tiene una forma de resolver.
- El robot laberinto con el algoritmo de relleno de callejones sin salida se ve limitado a trabajar a bajas velocidades, debido a que posee una forma de navegación más compleja que el de mano derecha o izquierda, ya que este

reconoce en todo su trayecto las paredes que tiene a su alrededor. Es por esto que este algoritmo podría demorarse un mayor tiempo en resolver un laberinto.

- Para el buen funcionamiento de los robots y que éstos alcancen un nivel competitivo la parte mecánica como la de control deben hacer sinergia ya que si una falla los resultados no van a ser buenos.

5.2 RECOMENDACIONES

- Cuando se realiza un robot de competencia (seguidor de línea, robot laberinto), lo más recomendable es desarrollarlo por etapas, primero con el diseño e implementación de la placa de control y comprobar que ésta funciona correctamente. También es recomendable desarrollar el programa de control por partes para ir probando que todo está bien, con lo cual el desarrollo se haría ordenadamente y después no se invertiría mucho tiempo en la búsqueda de problemas en el código.
- Para un buen funcionamiento de los prototipos se recomienda dar mantenimiento a sus motores, verificar que éstos se encuentran debidamente ajustados y las cajas reductoras no tienen problemas.
- Para la participación en un torneo de robótica llevar los prototipos a punto para que en éste no se pierda tiempo haciendo ajustes de última hora.
- Justo antes de la participación de los robots en un torneo desconectar los módulos bluetooth ya que estos dispositivos no son permitidos durante la competencia.
- Para trabajos futuros en el robot seguidor de línea se sugiere mejorar la mecánica del mismo utilizando una turbina con la cual se aumenta la adherencia a la pista, en lo que se refiere al control se puede realizar una

identificación de la pista, para tomar las curvas a velocidades seguras e ir a máxima velocidad en rectas.

- Para tener una mejor navegación con el robot laberinto, se recomienda el uso de una unidad de medición inercial (IMU), para realizar los giros con mayor precisión y evitar un mal posicionamiento dentro del laberinto.
- Para trabajos futuros en el robot laberinto se puede mejorar la navegación del robot con la identificación de caminos rectos, de ésta manera en las curvas iría a una velocidad segura, con lo cual no sufriría choques y en las rectas aceleraría.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Robot Challenge, Winners of RobotChallenge 2016, Robot Challenge, 13 Marzo 2016. [En línea]. Available: <https://www.robotchallenge.org/robotchallenge/resultate-2016/>. [Último acceso: 24 Marzo 2016].
- [2] J. S. Tercero, Cibernética aplicada Robots educativos, México: Alfaomega, 2009.
- [3] CER, Reglamento para la categoría "Seguidor de línea", Concurso Ecuatoriano de Robótica 2014, 9 Diciembre 2014. [En línea]. Available: <http://www.cer2015.com/index.php/bases-del-concurso>. [Último acceso: 18 Diciembre 2015].
- [4] J. Alonso, Proyecto Reborn, Hiyalife, 15 Julio 2014. [En línea]. Available: <http://hiyalife.com/meemo/95e747ede9>. [Último acceso: 20 12 2015].
- [5] N. B. Kiat, Proyecto futura, Micromouse USA, 29 noviembre 2011. [En línea]. Available: http://micromouseusa.com/?page_id=1342. [Último acceso: 20 diciembre 2015].
- [6] B. García García, Diseño y construcción de un robot móvil controlado con técnicas de lógica difusa implementadas en un FPGA, México D.F: Instituto Politécnico Nacional, 2012.
- [7] S. Ortigoza, Una panorámica de los robots móviles, Publicaciones urbe, 12 Febrero 2007. [En línea]. Available: <http://publicaciones.urbe.edu/index.php/telematique/article/viewArticle/833/2037>. [Último acceso: 12 Enero 2015].
- [8] Pololu, High-Power (HP) Micro Metal Gearmotors, Pololu, 1 Enero 2001. [En línea]. Available: <https://www.pololu.com/product/999>. [Último acceso: 12 Enero 2016].
- [9] Pololu, Micro Metal Gearmotors with Extended Motor Shafts, Pololu, 1 Enero 2001. [En línea]. Available: <https://www.pololu.com/product/2211>. [Último acceso: 12 Enero 2016].
- [10] C. Electrónica, Data sheet A139 Gearmotors, Carrod Electrónica, 1 Marzo 2010. [En línea]. Available: <http://www.carrod.mx/products/motores-reductor-48-1-65-rpm>. [Último acceso: 12 Enero 2016].
- [11] Pololu, QTR-8A Reflectance Sensor Array, Pololu, 1 Enero 2001. [En línea]. Available: <https://www.pololu.com/product/960>. [Último acceso: 21 Diciembre 2015].

- [12] Pololu, QTR-8RC Reflectance Sensor Array, Pololu, 1 Enero 2001. [En línea]. Available: <https://www.pololu.com/product/961>. [Último acceso: 12 Enero 2016].
- [13] Sparkfun, SparkFun Line Follower Array, Sparkfun, 31 Julio 2007. [En línea]. Available: <https://www.sparkfun.com/products/13582>. [Último acceso: 21 Diciembre 2015].
- [14] Pololu, Sharp GP2Y0A51SK0F Analog Distance Sensor, Pololu, 1 Enero 2001. [En línea]. Available: <https://www.pololu.com/product/2450>. [Último acceso: 12 Enero 2016].
- [15] Pololu, Parallax PING Ultrasonic Sensor, Pololu, 1 Enero 2001. [En línea]. Available: <https://www.pololu.com/product/1605>. [Último acceso: 12 Enero 2016].
- [16] Pololu, Magnetic Encoder Pair Kit for Micro Metal Gearmotors, Pololu, 1 Enero 2001. [En línea]. Available: <https://www.pololu.com/product/2598>. [Último acceso: 12 Enero 2016].
- [17] Pololu, Optical Encoder Pair Kit for Micro Metal Gearmotors, Pololu, 1 Enero 2001. [En línea]. Available: <https://www.pololu.com/product/2590>. [Último acceso: 12 Enero 2016].
- [18] Pololu, Encoder for Pololu Wheel 42x19mm, Pololu, 1 Enero 2001. [En línea]. Available: <https://www.pololu.com/product/1217>. [Último acceso: 12 Diciembre 2015].
- [19] Pololu, 3pi Robot, Pololu, 1 Enero 2001. [En línea]. Available: <https://www.pololu.com/product/975>. [Último acceso: 2 Enero 2016].
- [20] The Rutgers IEEE Robotics club, Line Following Kit, IEEE, 1 Enero 2011. [En línea]. Available: <http://ieee.rutgers.edu/line-following-kit>. [Último acceso: 31 Diciembre 2015].
- [21] Robomart, Robomart White Line Follower Robot, Robomart, 4 Febrero 2014. [En línea]. Available: <https://www.robomart.com/robomart-white-line-follower-robot>. [Último acceso: 31 Diciembre 2015].
- [22] PICAXE, Data sheet PICAXE, PICAXE, 3 Marzo 2011. [En línea]. Available: <http://www.picaxe.com/Hardware/Robot-Kits/PICAXE-PICone-Micromouse/>. [Último acceso: 31 Diciembre 2015].
- [23] G. Simbana y N. Cardenas , Diseño y construcción de un robot movil didactico para estudios experimentales de robotica cooperativa, Cuenca: Universidad Politécnica Salesiana, 2007.

- [24] Everlight, Technical Data sheet Infrared LED, Everlight, 5 Abril 2012. [En línea]. Available: <http://www.madnesselectronics.com/producto/emisor-y-receptor-infrarrojo/>. [Último acceso: 13 Enero 2016].
- [25] Pololu, 50:1 Micro Metal Gearmotor HP, Pololu, 1 Enero 2001. [En línea]. Available: <https://www.pololu.com/product/998>. [Último acceso: 13 Enero 2016].
- [26] P. F. Laspidea, Diseño e implementación de un robot para la automatización de un almacén, Pamplona, E.T.S. de Ingeniería Industrial, 2014, pp. 16-17.
- [27] A. Neal, Tips for selecting DC motors for your mobile robot, Servomagazine, 20 Enero 2010. [En línea]. Available: http://www.servomagazine.com/uploads/issue_downloads/pdf/Tips%20For%20Selecting%20DC%20Motors%20For%20Your%20Mobile%20Robot.pdf. [Último acceso: 31 Diciembre 2015].
- [28] K. Lima, Sensores de pared, Micromouse Brasil, 11 Marzo 2015. [En línea]. Available: <http://kleberufu.wix.com/micromousebrasil#!Sensores-de-Pared/c218b/550063760cf27b8ab267f6e6>. [Último acceso: 31 Diciembre 2015].
- [29] Atmel Corporation, Datasheet ATmega 324PA, San José, California: Atmel Corporation, 2015.
- [30] Toshiba, Data sheet Driver IC for dual DC motor, Toshiba, 9 Junio 2008. [En línea]. Available: <https://www.adafruit.com/images/product-files/1944/TB6612FNG%20datasheet.pdf>. [Último acceso: 10 Junio 2015].
- [31] Pololu, Dealing with Motor Noise, Pololu, 23 Marzo 2001. [En línea]. Available: <https://www.pololu.com/docs/0J15/9>. [Último acceso: 12 Junio 2015].
- [32] HC Information Technology, Data sheet HC-06, HC Information Technology, 3 Julio 2005. [En línea]. Available: <http://www.electronicoscaldas.com/modulos-rf/482-modulo-bluetooth-hc-06.html>. [Último acceso: 13 Junio 2015].
- [33] M. M. M. A. k. I. E. Vikram Balajia, ScienceDirect, 3 Diciembre 2015. [En línea]. Available: <http://www.sciencedirect.com/science/article/pii/S1877050915038302>. [Último acceso: 10 Enero 2016].
- [34] P. V. Medina Caicedo, Estudio del desempeño del motor de inducción usando un controlador difuso, Quito: Escuela Politécnica Nacional, 2007.
- [35] M. A. Gonzáles Torres, Inteligencia artificial en robots móviles de competencia para su aplicación en el diseño de un agente inteligente en un

- robot seguidor de trayectorias, Instituto Politécnico Nacional, 10 Enero 2009. [En línea]. Available: <http://tesis.ipn.mx/bitstream/handle/123456789/2593/ROBOTSMOVILES.pdf?sequence=1>. [Último acceso: 16 Abril 2016].
- [36] W. Pedrycz, Fuzzy control and Fuzzy systems, 1989. [En línea]. [Último acceso: 3 Enero 2016].
- [37] A. J. Barragán Piña, Síntesis de sistemas de control borroso estables por diseño, Universidad de Huelva, Departamento de Ingeniería Electrónica, 07 Julio 2009. [En línea]. Available: <http://rabida.uhu.es/dspace/bitstream/handle/10272/2629/b15480549.pdf?sequence=1>. [Último acceso: 05 Enero 2016].
- [38] W. Pullen, Maze Classification, Astrolog, 20 Noviembre 2015. [En línea]. Available: <http://www.astrolog.org/labyrnth/algrithm.htm>. [Último acceso: 3 Diciembre 2015].
- [39] C.-S. L. H.-H. H. S.-H. C. & C.-Y. L. Juing-Huei Su, An intelligent line-following robot project for introductory robot courses, Lunghwa University of Science and Technology, 20 Enero 2010. [En línea]. Available: [http://www.wiete.com.au/journals/WTE&TE/Pages/Vol.8,%20No.4%20\(2010\)/9-15-SU-J-H.pdf](http://www.wiete.com.au/journals/WTE&TE/Pages/Vol.8,%20No.4%20(2010)/9-15-SU-J-H.pdf). [Último acceso: 11 Enero 2016].

ANEXO A

MANUAL DE USUARIO

A.1 Características del robot seguidor de línea

El seguidor de línea es un robot autónomo para competencias (Figura A.1), este robot sigue una línea negra en un fondo blanco, para el seguimiento de la línea el robot cuenta con dos algoritmos de control para hacerlo.

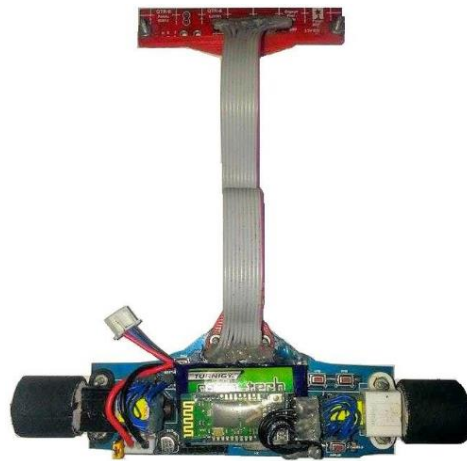


Figura A.1 Robot seguidor de línea

Características:

- Largo: 16.4 cm.
- Ancho: 15 cm.
- Alto: 2.5 cm
- Peso: 170 g.
- La configuración del robot es tipo tracción diferencial.
- Alimentación: Una batería de corriente continua.
- En la parte sensorial: tiene un arreglo de ocho sensores infrarrojos para la detección de la línea y un par de encoders magnéticos instalados uno en cada motor para medir la distancia que el robot avanza.
- El cerebro del robot es un microcontrolador ATmega 324PA.

- Para la interacción con el robot se tiene cuatro pulsadores y cuatro leds, además consta de un conector para comunicación serial.

A.2 Funcionamiento del prototipo

Antes de accionar el robot verificar que el voltaje de la batería esté por encima de 7.5 V, si esto es así colocar el jumper de encendido. En la Figura A.2 y Figura A. 3 se observa las partes del prototipo.

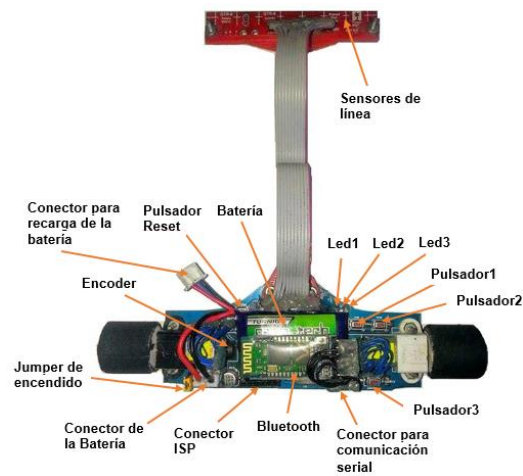


Figura A.2 Parte superior del robot seguidor de línea.

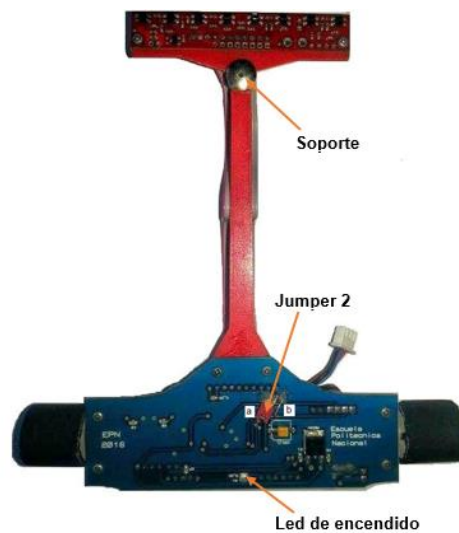


Figura A. 3 Parte inferior del robot seguidor de línea.

En caso de que se vaya a utilizar la comunicación se debe emparejar al bluetooth con un dispositivo externo (pc o Smartphone), la clave es: 1234. En el computador para la comunicación se puede utilizar cualquier programa como por ejemplo: hércules, hyperterminal, labview, etc.

Nota importante: antes de hacer correr el robot se debe leer el voltaje de la batería y calibrar al menos una vez los sensores de línea.

A.2.1 Calibración de sensores.

Antes de hacer correr al prototipo se necesita calibrar los sensores para que éstos se adapten a las condiciones del lugar. Para realizar la calibración se debe seguir con los siguientes pasos:

1. Encender al robot si está apagado, éste se inicia en el menú principal del programa.
2. Colocar al robot sobre la pista.
3. Presionar el pulsador1; los leds titilan por un segundo.
4. Se debe girar manualmente al robot de tal forma que los sensores pasen por las superficies blanca y negra; el usuario sabrá que el robot se está calibrando porque el led 2 está titilando y dejará de hacerlo cuando se termine la calibración. En esta parte el usuario podrá ver los datos de calibración si lo desea ya que el robot envía éstos a través de un dispositivo bluetooth a una pc. Terminado el envío el programa retorna al menú principal.

A.2.2 Carrera del robot

En esta parte se encuentran los algoritmos que tiene el robot para seguir la línea; cuenta con un controlador borroso y un controlador PID.

A.2.2.1 Controlador borroso

1. Encender al robot si está apagado, éste se inicia en el menú principal del programa.
2. Presionar el pulsador3, los leds titilan por un segundo.

3. Presionar el pulsador2, los leds titilan por un segundo.
4. Colocar al robot sobre la pista.
5. Presionar el pulsador3; el robot arrancará cuando el pulsador deje de estar presionado.
6. Para detener el robot presionar el pulsador3, el cual envía al menú principal.

A.2.2.2 Controlador PID

1. Encender el robot si está apagado, éste se inicia en el menú principal del programa.
2. Presionar el pulsador3, los leds titilan por un segundo.
3. Presionar el pulsador1, los leds titilan por un segundo.
4. A continuación el robot esperará que se le envíe las constantes del controlador (en caso de que no se le envíe, el prototipo corre con constantes por default). Las constantes que se envían son: proporcional, derivativa, integral, distancia de arranque, distancia de velocidad máxima, velocidad crucero del robot, velocidad de lazo abierto uno, velocidad de lazo abierto dos, velocidad de partida y velocidad máxima.

Las constantes proporcional, integral, derivativa son con las que se calibra el PID.

La velocidad crucero va a ser la velocidad con la que normalmente el robot va a recorrer la pista.

Las constantes: velocidad de lazo abierto1 y velocidad de lazo abierto2 son las velocidades que se setean a los motores para el control de lazo abierto; la velocidad de lazo abierto2 se setea a la rueda que va por la parte interna de la curva y la velocidad de lazo abierto1 se setea a la rueda que va por la parte externa.

Las últimas cuatro constantes: distancia de arranque (D_{arranque}), distancia de velocidad máxima (D_{vmax}), velocidad de partida (V_{partida}) y velocidad máxima (V_{max}) son para el arranque del robot. En la Figura A.4 se muestra la gráfica del arranque donde se utiliza las constantes mencionadas.

2. Presionar el pulsador2, los leds titilan por un segundo.
3. Colocar el jumper2 en la posición b.
4. Presionar el pulsador3; el usuario se podrá dar cuenta que la batería fue medida porque se encienden los tres leds del robot. En esta parte el usuario podrá ver los datos de la batería si lo desea ya que el robot los envía a través de un dispositivo bluetooth.
5. Retornar el jumper2 a la posición a. Terminado el proceso el programa retorna al menú principal.

A.2.3.2 Revisión de motores

1. Encender el robot si está apagado, éste se inicia en el menú principal del programa.
2. Presionar el pulsador2, los leds titilan por un segundo.
3. Presionar el pulsador1, en ésta parte las ruedas giran a diferentes velocidades con lo cual se podría descartar un posible fallo en los motores o sus conexiones. Terminado el proceso el programa retorna al menú principal.

A.2.3.3 Revisión de encoders

1. Encender el robot si está apagado, éste se inicia en el menú principal del programa.
2. Presionar el pulsador2, los leds titilan por un segundo.
3. Volver a presionar el pulsador2; para esta parte se hace recorrer manualmente al robot una distancia conocida y se verifica que los encoders estan bien porque la distancia recorrida debe ser igual a la que se envía a través del dispositivo bluetooth. Terminado el envío el programa retorna al menú principal.

MANUAL DE USUARIO DEL ROBOT LABERINTO

A.3 Características del robot laberinto

Este robot es autónomo, construido especialmente para competencias de resolución de laberintos. En la Figura A.6 se puede observar el prototipo.

Este cuenta con diferentes algoritmos para la resolución de laberintos, estos son: algoritmo de la mano derecha, algoritmo de la mano izquierda y algoritmo de relleno de callejones sin salida.

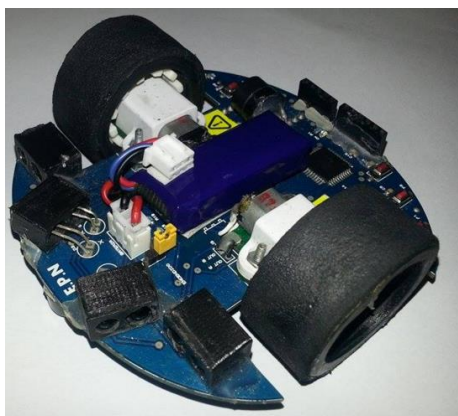


Figura A.6 Robot laberinto

Características:

- Largo: 10 cm.
- Ancho: 10 cm.
- Alto: 4 cm.
- Peso: 200 g.
- Tracción diferencial la cual permite realizar giros en su propio eje.
- Alimentación: Una batería de corriente continua.
- Posee cuatro sensores infrarrojos. Dos para la detección de paredes laterales y dos para la detección de la pared frontal. Con un rango de medición de distancia máxima de cada sensor de 30 cm.
- Incorpora un encoder en cada rueda el cual le permite avanzar distancias o realizar giros con mayor precisión.
- Usa un microcontrolador Atmel ATmega 324PA.
- Para la interacción con el usuario tiene cuatro leds, cuatro pulsadores y un buzzer; además posee un conector para la comunicación serial.

A.4 Funcionamiento del prototipo

Para la utilización primero se debe realizar la adquisición de set point, para asegurar el funcionamiento correcto del robot. Después de esto se puede usar cualquier algoritmo de solución. En la Figura A.7 se puede ver todos los componentes que el robot posee y sus respectivas ubicaciones.

Primero se debe encender al robot, para esto se conecta el jumper de encendido. Al encenderse el prototipo se encenderá el led de encendido y el programa del robot irá al menú principal donde se tiene las siguientes opciones: adquisición de set point, algoritmos de resolución, chequeo del prototipo. Éstas serán descritas más adelante.

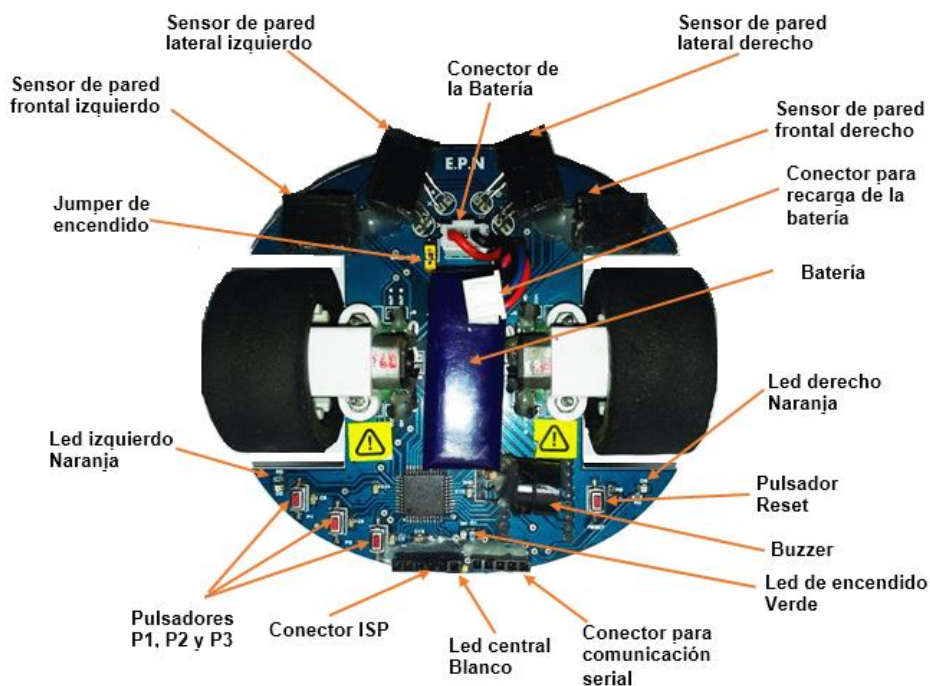


Figura A.7 Ubicación de los componentes en el robot.

En caso de que se vaya a utilizar la comunicación se debe emparejar al bluetooth con un dispositivo externo (pc o Smartphone), la clave es: 1234. En el computador, para la comunicación se puede utilizar cualquier programa como por ejemplo: hércules, hyperterminal, labview, etc.

A.4.1 Adquisición de set point

Se recomienda realizar una adquisición de set point cada vez que el prototipo vaya a un laberinto nuevo, debido a que éste puede tener diferente ancho de las celdas. Para realizar la adquisición de set point se debe hacer los siguientes pasos:

1. Encender el robot si está apagado, éste se inicia en el menú principal del programa.
2. Colocar al robot entre dos paredes laterales del laberinto.
3. Presionar el pulsador P1.
4. Posicionar al robot en el centro de la celda.
 - Si el led derecho esta encendido mover el robot hacia la izquierda.
 - Si el led izquierdo esta encendido mover el robot hacia la derecha.
 - Realizar las acciones anteriores hasta colocar el prototipo en el centro de la celda ahí se encenderá el led central y tomará automáticamente el valor de set point.
5. El robot envía mediante comunicación serial la distancia que mide cada sensor y el valor de set point. Terminado el envío el programa retorna al menú principal.

A.4.2 Algoritmos de resolución

Aquí se encuentran los algoritmos de resolución del laberinto que son: algoritmo de la mano derecha, algoritmo de la mano izquierda y algoritmo de relleno de callejones sin salida.

A.4.2.1 Algoritmo de la mano derecha

Para el uso del algoritmo de la mano derecha, previamente se debe realizar la adquisición de set point y continuar con los siguientes pasos:

1. Encender el robot si está apagado, éste se inicia en el menú principal del programa.
2. Presionar P2, enciende el led central y se escucha una señal auditiva.
3. Colocar el robot a la entrada del laberinto.

4. Presionar P1 para iniciar la resolución del laberinto por el algoritmo de la mano derecha.
5. Para detener el robot presionar el pulsador1 el cual envía al menú principal.

A.4.2.2 Algoritmo de la mano izquierda

Para el uso del algoritmo de la mano izquierda, previamente se debe realizar la adquisición de set point y continuar con los siguientes pasos:

1. Encender el robot si está apagado, éste se inicia en el menú principal del programa.
2. Presionar P2, se enciende el led central y se escucha una señal auditiva.
3. Colocar el robot a la entrada del laberinto.
4. Presionar P2 para iniciar la resolución del laberinto por el algoritmo de la mano izquierda.
5. Para detener el robot presionar el pulsador1 el cual envía al menú principal.

A.4.2.3 Algoritmo de relleno de callejones sin salida

Para el uso del algoritmo de relleno de callejones sin salida, primero se realiza un recorrido por el laberinto donde el robot guarda en un vector las intersecciones y caminos sin salida que encuentre, para lo cual se debe realizar los siguientes pasos:

1. Encender el robot si está apagado, éste se inicia en el menú principal del programa.
2. Presionar P2, se enciende el led central y se escucha una señal auditiva.
3. Presionar P3, se enciende el led derecho y se escucha una señal auditiva.
4. Colocar el robot a la entrada del laberinto.
5. Presionar P1 para iniciar la resolución del laberinto.
6. Para detener el robot presionar el P1, el cual envía al menú principal.

Después de haber adquirido el vector se debe optimizarlo antes de realizar el recorrido final, para la optimización del vector se debe realizar lo siguiente:

7. Presionar P2, se enciende el led central y se escucha una señal auditiva.
8. Presionar P3, Se enciende el led derecho y se escucha una señal auditiva.
9. Presionar P2 para que el robot optimice el vector.
10. Esperar unos segundos mientras el robot optimiza el vector, realice un titileo de leds y haga una señal auditiva con la cual confirma que el vector fue optimizado.

Para realizar el recorrido final donde el robot evita entrar a los callejones que no tienen salida se siguen los siguientes pasos:

11. Colocar el robot a la entrada del laberinto.
12. Presionar P2, el robot recorrerá el laberinto y al terminar el recorrido se detendrá solo.

A.4.3 Chequeo del robot

En el chequeo del robot se tiene funciones que comprueban el funcionamiento de encoder, motores y voltaje de la batería.

A.4.3.1 Revisión de encoders

Comprueba el correcto funcionamiento de los encoders, los cuales son utilizados para realizar giros y avances rectilíneos.

1. Encender el robot si está apagado, éste se inicia en el menú principal del programa.
2. Presionar el pulsador P3.
3. Presionar el pulsador P1 para ingresar a la rutina de revisión de encoders.
4. Mover la llanta izquierda y comprobar que el led izquierdo titile mientras gira la llanta.
5. Mover la llanta derecha y comprobar que el led derecho titile mientras gira la llanta.
6. Terminado el proceso el programa retorna al menú principal.

A.4.3.2 Revisión de motores

Al entrar a esta función las ruedas giran a diferentes velocidades con lo cual se podría descartar un posible fallo en los motores o sus conexiones.

1. Encender el robot si está apagado, éste se inicia en el menú principal del programa.
2. Presionar el pulsador P3 para pasar a un submenú.
3. Presionar el pulsador P2 para ingresar a la rutina revisión de motores.
 - Comprobar que las dos ruedas comiencen a girar con esto se asegura que no haya problema en alguna rueda.
4. Terminado el proceso el programa retorna al menú principal.

A.4.3.3 Medición de voltaje de la batería

Comprueba el voltaje que tiene la batería en ese instante.

1. Encender el robot si está apagado, éste se inicia en el menú principal del programa.
2. Presionar el pulsador P3.
3. Presionar el pulsador P3 para ingresar a la rutina de medición de voltaje.
 - El robot medirá el voltaje de la batería y si esta tiene un voltaje menor a 7.5 dará una alerta auditiva la cual significa que debe recargarse inmediatamente.
4. El robot envía mediante comunicación serial el valor de voltaje de la batería. Terminado el envío el programa retorna al menú principal.

ANEXO B

PLACAS Y DIAGRAMAS ESQUEMÁTICOS

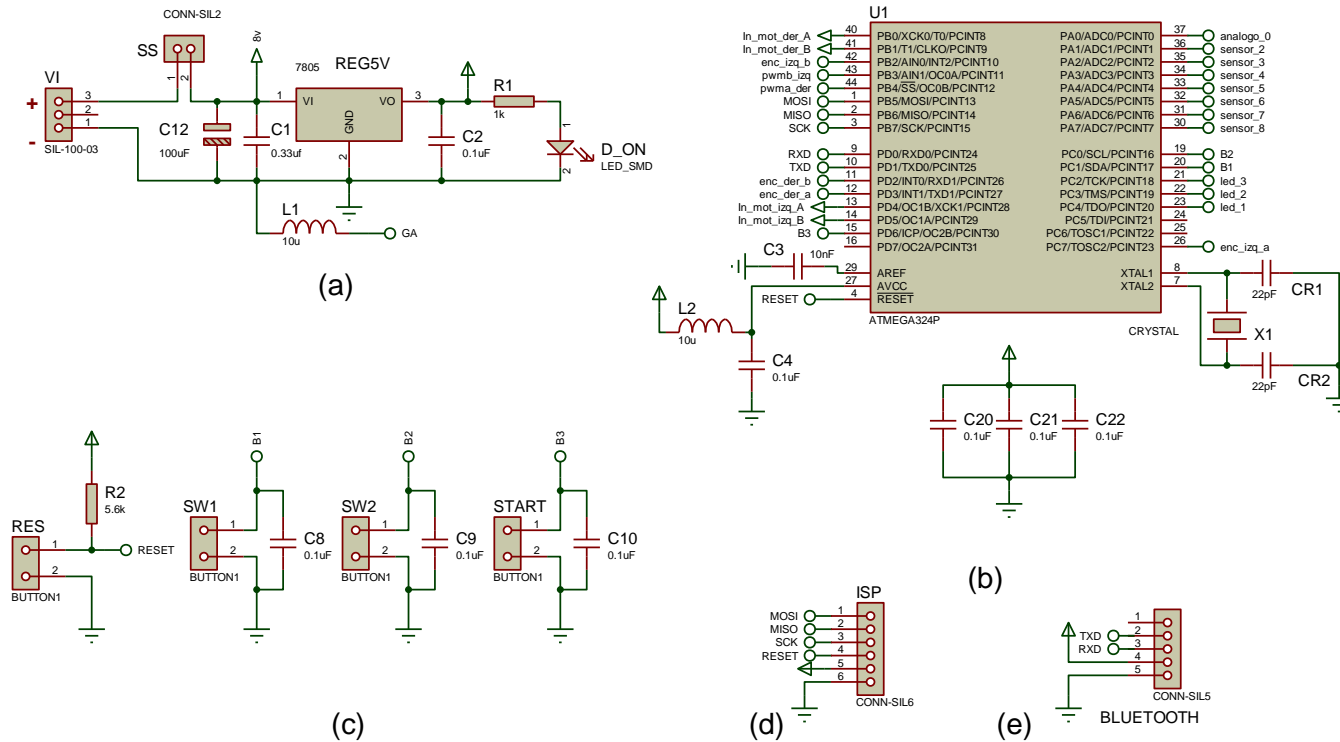


Figura B.1 Diagrama esquemático del robot seguidor de línea. (a) Fuente de alimentación. (b) Microcontrolador ATmega 324PA. (c) Pulsadores. (d) Programador. (e) Conector para comunicación serial.

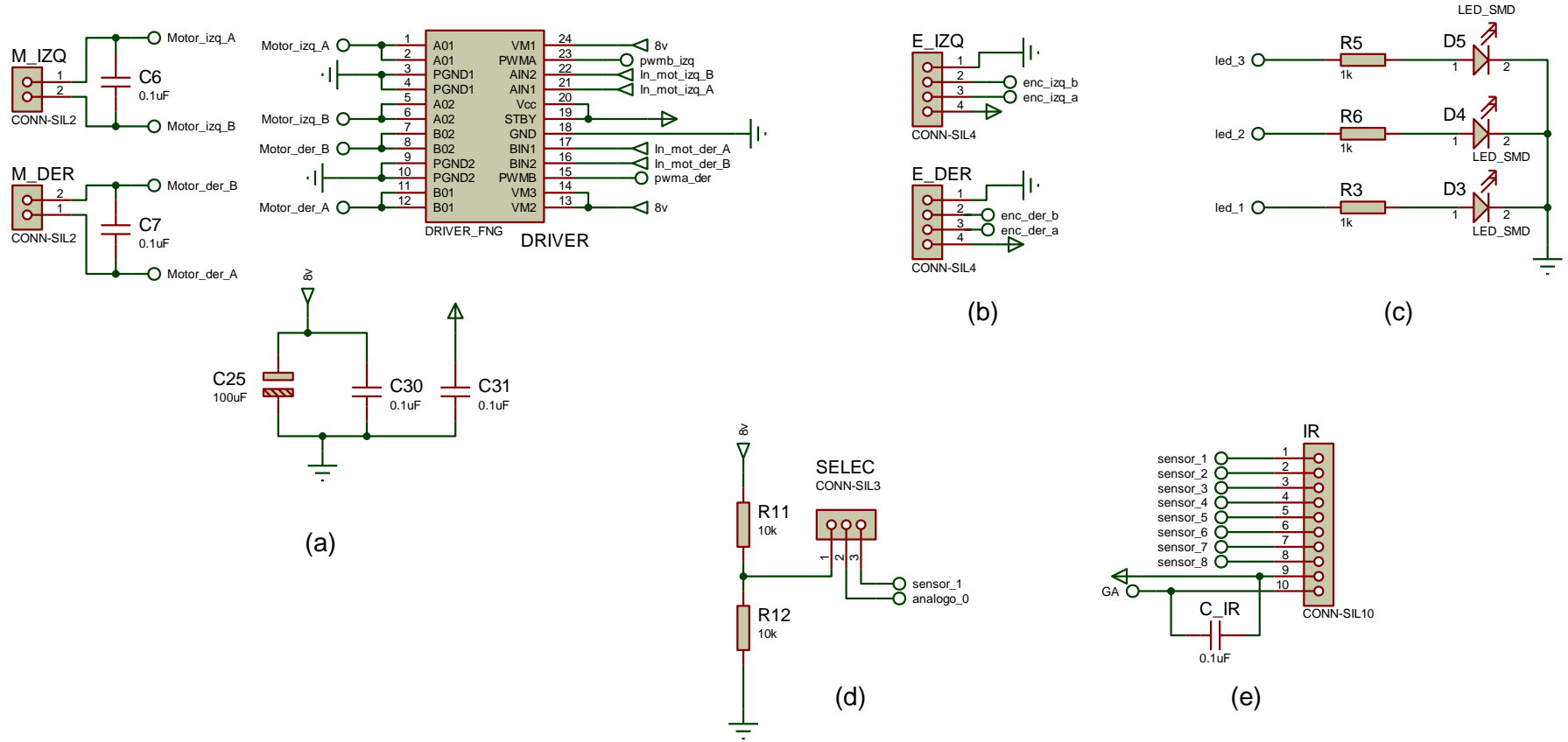


Figura B.2 Diagrama esquemático del robot seguidor de línea. (a) Controlador de potencia. (b) Conectores para encoder. (c) Leds. (d) Sensor de voltaje. (e) Conector para el sensor de línea.

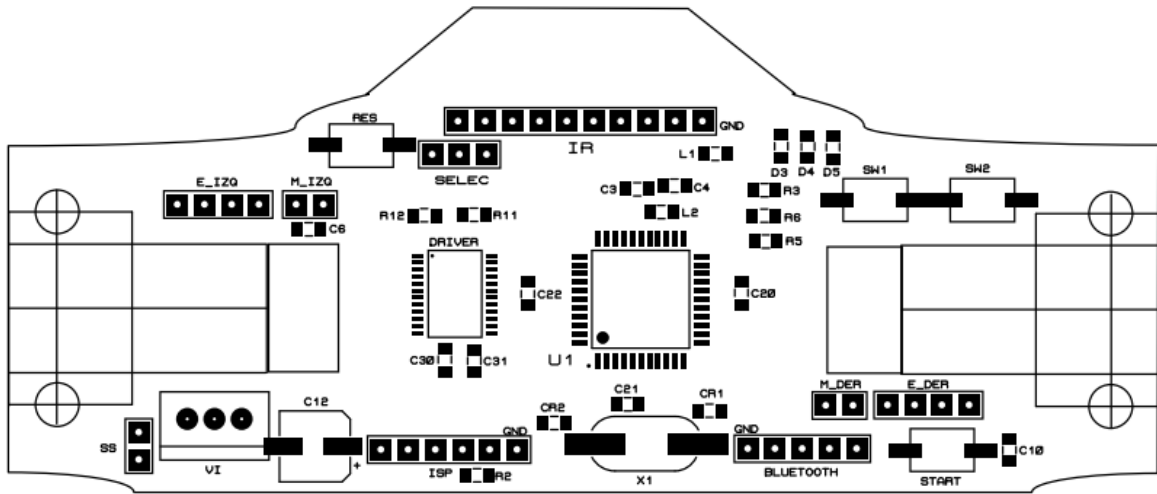


Figura B.3 Plano de posicionamiento superior, tarjeta del robot seguidor de línea.

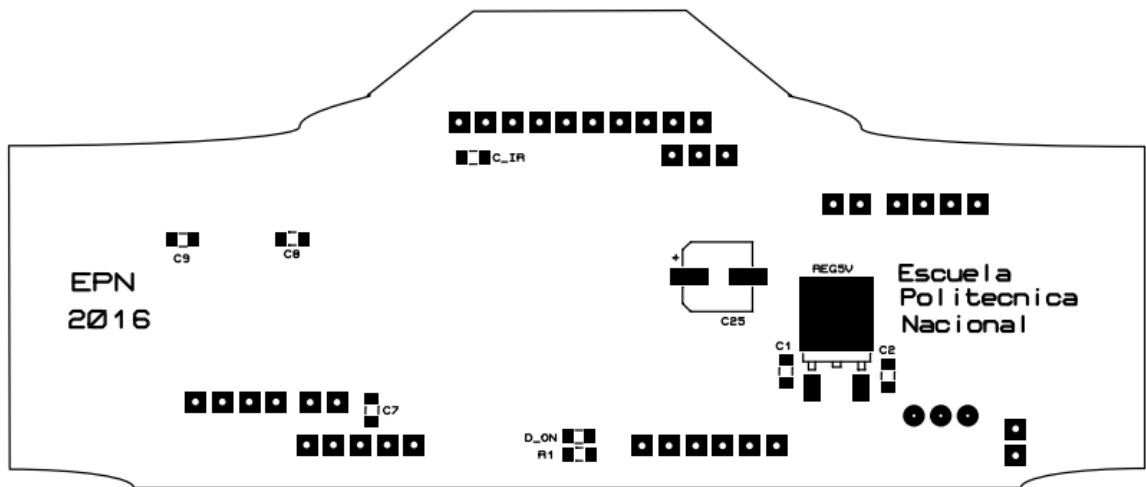


Figura B.4 Plano de posicionamiento inferior, tarjeta del robot seguidor de línea.

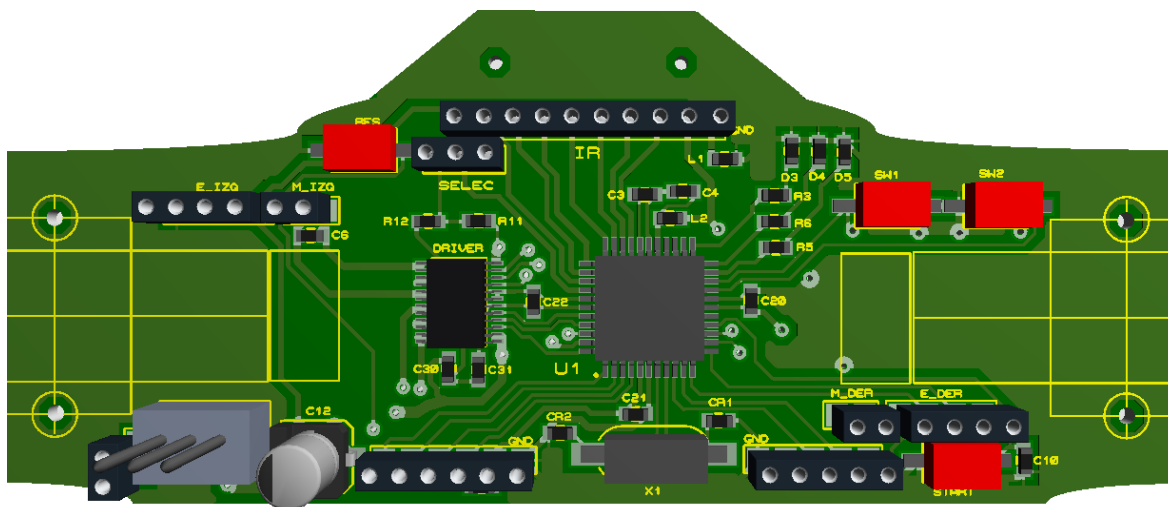


Figura B.5 Tarjeta de control robot seguidor de línea, vista superior.

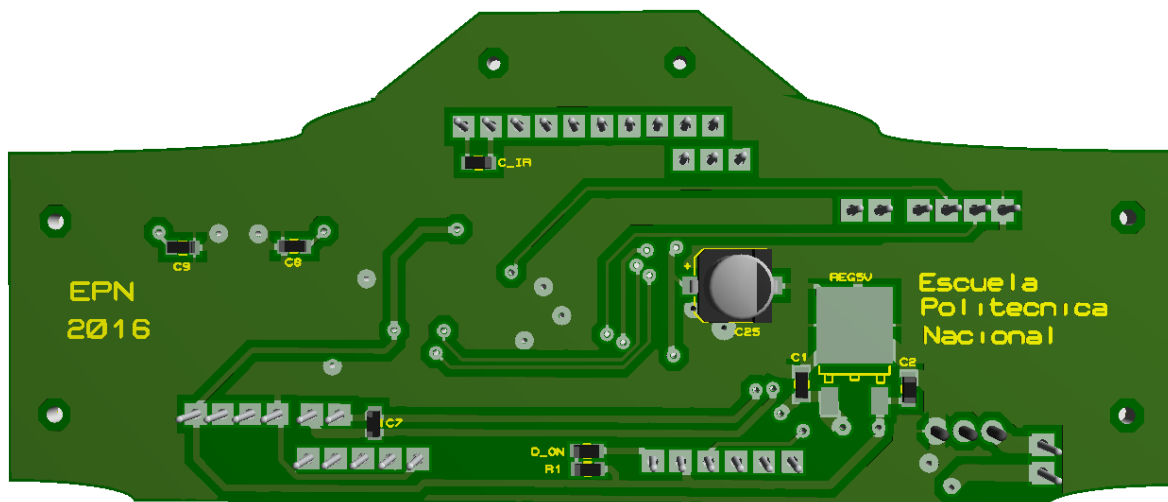


Figura B.6 Tarjeta de control robot seguidor de línea, vista inferior.

Tabla B.1 Lista de componentes, tarjeta del robot seguidor de línea.

COMPONENTES	VALORES
RESISTENCIAS	
R1, R3, R5, R6	1kohm
R2	5.6kohm
R11, R12	10kohm
CAPACITORES	
C1	0.33uF
C2, C4, C6-C10, C20-C22 ,C30 ,C31 ,C_IR	0.1uF
C3	10nF
C12, C25	100uF
CR1, CR2	22pf
INTEGRADOS	
U1	ATMEGA 324PA
REG 5V	7805
Controlador de potencia	TB6612FNG
DIODOS	
D3	Amarillo
D4	Verde
D5	Rojo
D_ON	Azul
BOBINAS	
L1, L2	10uH
CONECTORES	
BLUETOOTH	Comunicación serial
E_DER	Encoder derecho
E_IZQ	Encoder izquierdo
IR	Sensores de línea
ISP	Programador
M_DER	Motor derecho
M_IZQ	Motor izquierdo
SS	Encendido
SELECT	Sensor de batería
VI	Alimentación
PULSADORES	
RES, START, SW1, SW2	SW_029
CRISTAL	
X1	16 MHz

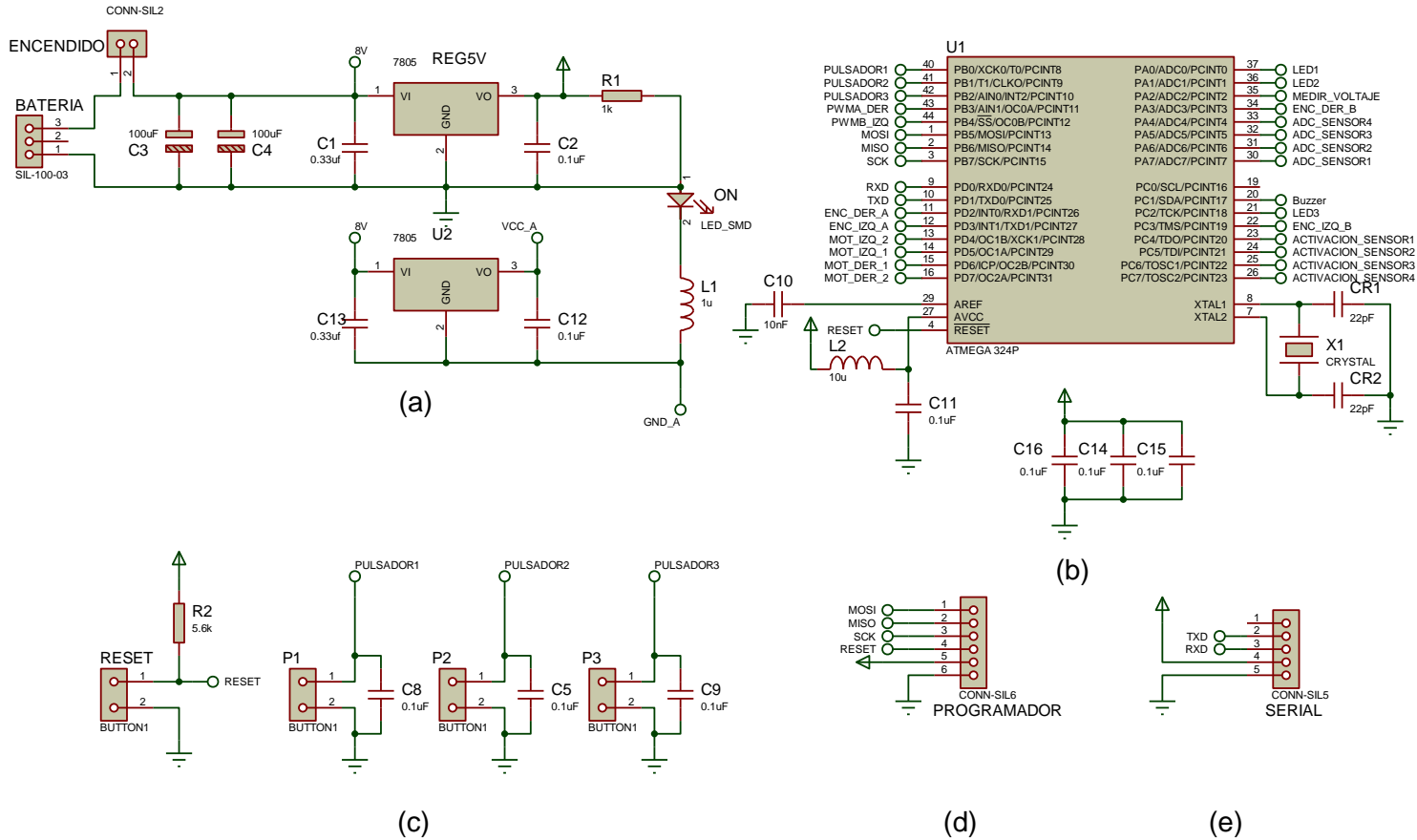


Figura B.7 Diagrama esquemático del robot laberinto. (a) Fuente de alimentación. (b) Microcontrolador ATmega 324PA. (c) Pulsadores. (d) Programador. (e) Conector para comunicación serial.

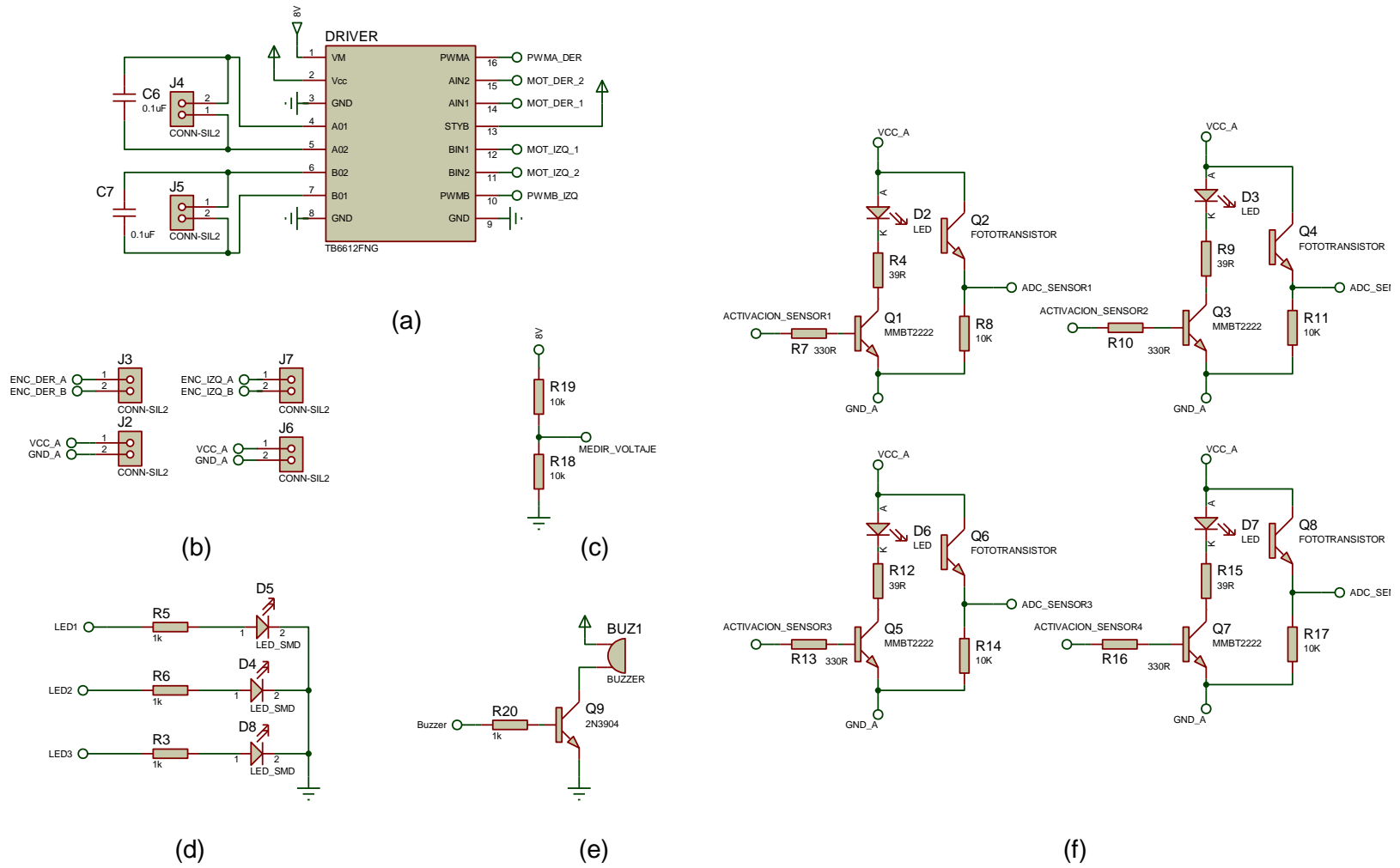


Figura B.8 Diagrama esquemático del robot laberinto. (a) Controlador de potencia. (b) Conectores para encoder. (c) Sensor de voltaje. (d) Leds. (e) buzzer. (f) sensores de distancia.

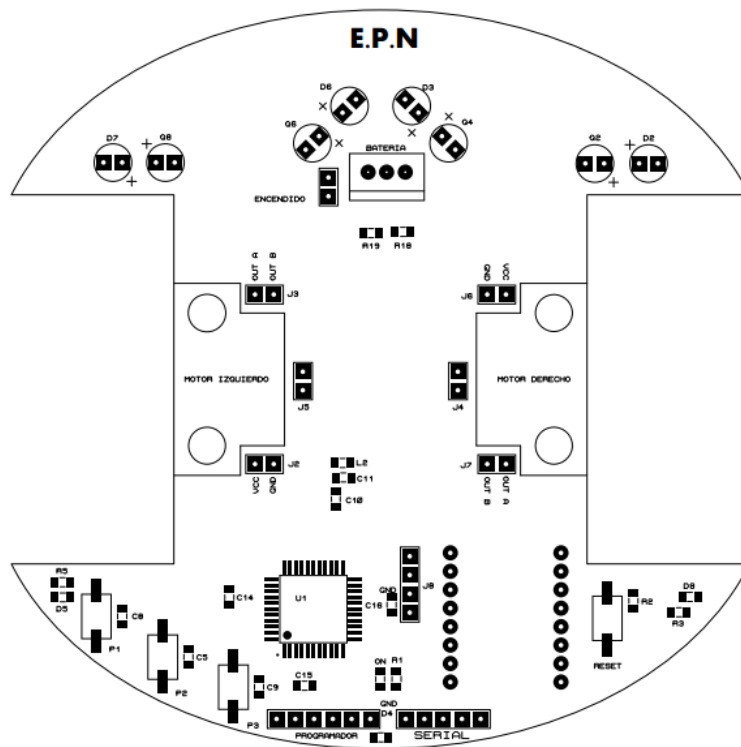


Figura B.9 Plano de posicionamiento superior, tarjeta del robot laberinto.

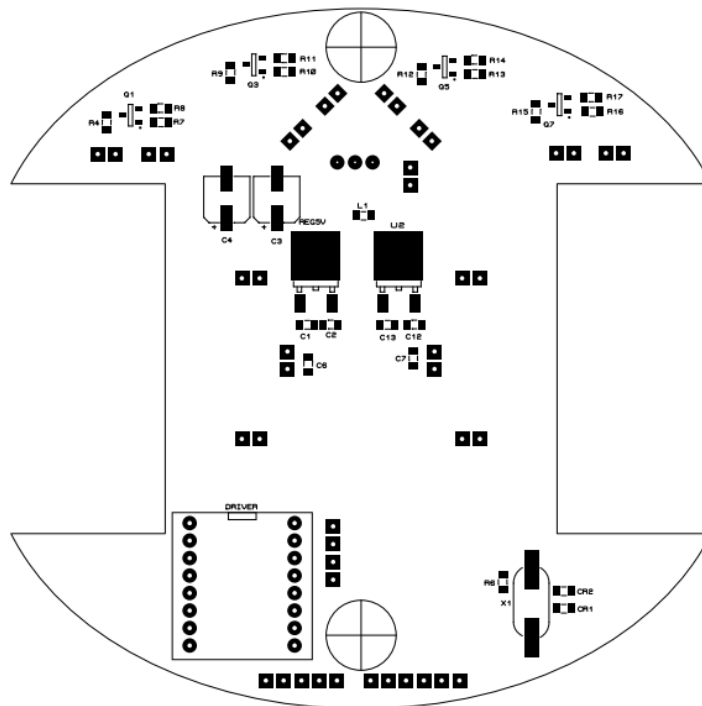


Figura B.10 Plano de posicionamiento inferior, tarjeta del robot laberinto.

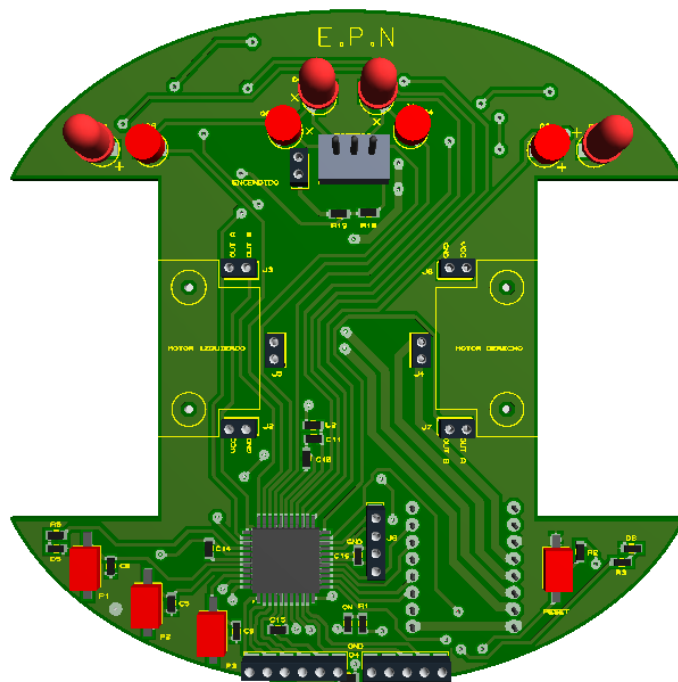


Figura B.11 Tarjeta de control robot laberinto, vista superior.

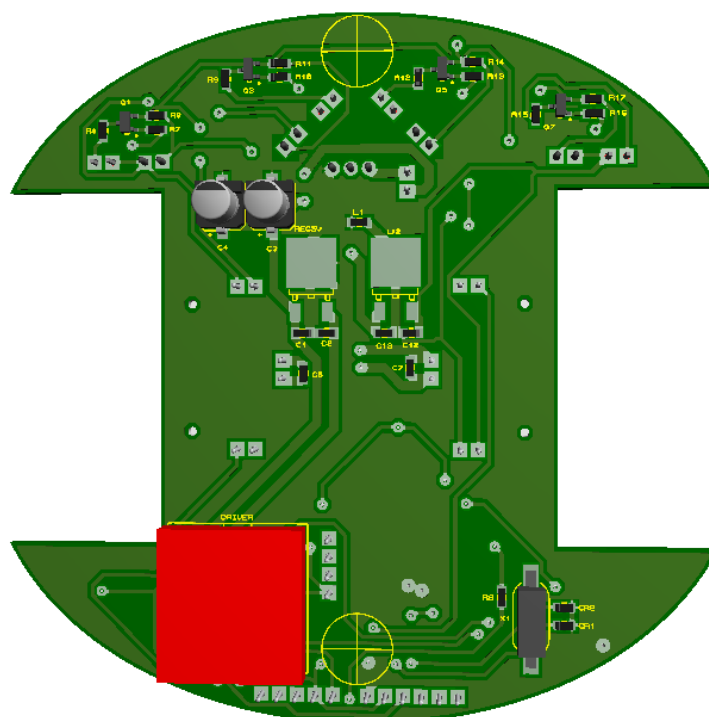


Figura B.12 Tarjeta de control robot laberinto, vista inferior.

Tabla B.2 Lista de componentes, tarjeta del robot laberinto.

COMPONENTES	VALORES	COMPONENTES	VALORES
RESISTENCIAS		DIODOS	
R1, R3, R5, R6, R20	1kohm	D2,D3,D6,D7	IR333
R2	5.6kohm	D4	Amarillo
R4, R9, R12, R15	39ohm	D5	Blanco
R7, R10, R13, R16	330ohm	D8	Amarillo
R8, R11, R14, R17-R19	10kohm	ON	Verde
CAPACITORES		BOBINAS	
C1,C13	0.33uF	L1,L2	10uH
C2,C5-C9,C11,C12,C14-C16	0.1uF	CONECTORES	
C10	10nF	SERIAL	Comunicación serial
C3, C4	100uF	J6,J7	Encoder derecho
CR1, CR2	22pf	J2,J3	Encoder izquierdo
INTEGRADOS		PROGRAMADOR	
U1	ATMEGA 324PA	J4	Motor derecho
U2, REG 5V	7805	J5	Motor izquierdo
Controlador de potencia	TB6612FNG	ENCENDIDO	Encendido
TRANSISTORES		J8	
Q1, Q3, Q5, Q7	MMBT2222	BATERÍA	Alimentación
Q2, Q4,Q6,Q8	PD333-3B	PULSADORES	
Q9	2N3904	RESET,P1,P2,P3	SW_029
CRISTAL		BUZZER	
X1	16 MHz	BUZ1	BUZZER

ANEXO C

REGLAMENTO DEL ROBOT SEGUIDOR DE LÍNEA Y LABERINTO

CONCURSO ECUATORIANO DE ROBÓTICA CER2015 REGLAMENTO PARA LA CATEGORÍA “SEGUIDOR DE LINEA”



CAPÍTULO 1: DE LAS DISPOSICIONES GENERALES.

Artículo 1.1 La categoría Seguidor de Línea se desarrollará en dos modalidades: seguidor de línea DESTREZA y seguidor de línea VELOCIDAD.

Artículo 1.2 Cada delegación bajo autorización del representante podrá inscribir como máximo 2 equipos para cada modalidad, los cuales estarán conformados por un máximo de 2 estudiantes.

Artículo 1.3 El jurado calificador podrá aplicar en cualquier circunstancia el presente reglamento y tendrá las atribuciones necesarias para decidir cualquier aspecto o eventualidad que no esté contemplada en el mismo.

Artículo 1.4 Todos los participantes deberán acogerse a lo estipulado en el Reglamento General en cuanto a inscripciones, participación y penalizaciones generales.

Artículo 1.5 El presente reglamento es una evolución sujeta a mejoras continuas que toma como referencia reglamentos presentados por las universidades anfitrionas en eventos anteriores y concursos realizados a nivel mundial.

CAPÍTULO 2: DE LOS REQUERIMIENTOS TÉCNICOS DEL PROTOTIPO.

Artículo 2.1 Los prototipos de ambas modalidades deberán cumplir los requerimientos técnicos comunes que se detallan a continuación:

- El prototipo debe ser autónomo en su totalidad, es decir que por ningún motivo puede ser manejado a control remoto, por computadora o cualquier dispositivo que se encuentre en el exterior.
- No existe limitaciones en el tamaño, altura o peso del prototipo.

- La fuente de energía para alimentar los circuitos electrónicos así como los motores del prototipo, será de cualquier tipo de baterías de corriente continua.
- El accionamiento del prototipo será en forma manual una vez que lo indique el juez, los robots no pueden tener partes en movimiento (ruedas), antes de la señal de salida.
- El prototipo no puede tener ningún tipo de mecanismo de sección para aumentar la fuerza de contacto con el suelo.

Artículo 2.2 Para la modalidad destreza los prototipos deberán constar de hardware y software diseñado por los participantes y/o tarjetas de desarrollo, el tipo de controlador del sistema es libre, se prohíbe el uso de prototipos conformados únicamente con kits de desarrollo para robots.

Artículo 2.3 Para la modalidad velocidad, el prototipo podrá ser adquirido en casas comerciales y se privilegiará la programación y los algoritmos de control para llevar el robot a la meta en el menor tiempo posible.

CAPÍTULO 3: DE LOS REQUERIMIENTOS TÉCNICOS DE LAS PISTAS.

Artículo 3.1 Los escenarios tanto para seguidor de línea velocidad como para seguidor de línea destreza tendrán fondo blanco y las pistas serán hechas con cinta aislante de color negro. Estas serán diseñadas a criterio de los organizadores y serán publicadas con un mes de anterioridad en el sitio web del concurso.

Artículo 3.2 Las especificaciones de diseño de la pista para la modalidad destreza son las siguientes:

- Curvas cerradas a un ángulo no menor de 30°.

- Discontinuidades en la trayectoria máxima de 5 cm.
- Líneas rectas y cruzadas, además se tendrá desvíos.

Artículo 3.3 Las especificaciones de diseño de la pista para la modalidad velocidad son las siguientes:

- La pista estará hecha con líneas rectas y curvas.
- No existirá discontinuidades ni intersecciones.

CAPÍTULO 4: DEL DESARROLLO DE LAS COMPETENCIAS.

Artículo 4.1 En el preámbulo de la competencia los robots deberán ser sometidos a verificación, si el jurado calificador observara el incumplimiento de alguno de los puntos detallados en el capítulo 2, el robot será descalificado.

Artículo 4.2 Todos los prototipos deberán tener sus baterías totalmente cargadas, antes de entregar a los organizadores, después de esto, no podrán hacer ninguna modificación o acoplamiento, etc.

Artículo 4.3 Los prototipos estarán situados y resguardados en el área de jueces. Los prototipos serán entregados a sus respectivos dueños al finalizar cada ronda. Una ronda es una fase donde participan todos los clasificados, únicamente clasificarán los robots que terminen el recorrido de la pista.

Artículo 4.4 En el desarrollo de la competencia, se contará con marcadores electrónicos, los cuales indicarán el tiempo de llegada de cada participante.

Artículo 4.5 El orden de participación se define mediante un sorteo interno de los organizadores.

Artículo 4.6 Si el participante no acude al llamado, automáticamente perderá su turno.

Artículo 4.7 El participante ubicará al robot en la posición de inicio y deberá activarlo cuando el juez lo indique. Una vez que comience su marcha el robot, se iniciará el cronómetro, éste se detendrá cuando el prototipo finalice el circuito trazado.

Artículo 4.8 En la modalidad destreza cada participante cuenta con 3min para recorrer la pista, una vez finalizado este tiempo, se detendrá el cronómetro y la participación del estudiante finalizará.

Artículo 4.9 En la modalidad velocidad cada participante cuenta con 1min para recorrer la pista, una vez finalizado este tiempo, se detendrá el cronómetro y la participación del estudiante finalizará.

Artículo 4.10 Se permitirá el reemplazo de baterías y cualquier otro dispositivo, solo después de cada ronda.

Artículo 4.11 En ambas modalidades cada participante contará con dos oportunidades para realizar el recorrido. Se considerará el menor tiempo de las dos participaciones.

Artículo 4.12 El prototipo está obligado a permanecer dentro de la pista y seguir la trayectoria marcada durante toda la carrera. Si el robot sale de la pista y vuelve de nuevo al mismo punto de la pista por sí mismo, puede continuar la carrera. Si el robot se sale de la pista completamente o permanece inmóvil durante 5 segundos, la participación se dará por terminada inmediatamente. En el caso de que el robot se salga y acorte camino, el intento se declarará fallido.

Artículo 4.13 El participante no podrá tocar el prototipo mientras éste se encuentre haciendo la trayectoria, en caso de que esto suceda, automáticamente pierde un turno. Solo podrá tocarlo cuando inicie o termine el recorrido.

Artículo 4.14 En ambas categorías, si ninguno de los participantes puede completar la trayectoria en los tiempos máximos establecidos, el jurado declarará nula la categoría y no existirá ganador.

Artículo 4.15 Para que exista una siguiente etapa deberá existir al menos tres prototipos que hayan terminado el circuito dentro del tiempo establecido. Si el jurado lo decide, la final se puede realizar en una nueva pista. En caso de que ocurra un empate, se utilizará el sentido contrario de la pista para desempatar la competencia.

CAPÍTULO 5: DETERMINACIÓN DEL GANADOR DE LA COMPETENCIA.

Artículo 5.1 El jurado calificador declarará al ganador de la competencia cuando en la etapa final un prototipo haya alcanzado la meta en un tiempo menor al de los contrincantes. Para ocupar el segundo y tercer lugar los prototipos respectivos deberán alcanzar la meta, caso contrario el jurado calificador declarará un único ganador, sin que haya segunda y tercera ubicación.

Artículo 5.2 Una vez finalizadas las competencias, el jurado calificador publicará entre los presentes el nombre del robot ganador y la universidad a la que pertenece.

CONCURSO ECUATORIANO DE ROBÓTICA CER2015 REGLAMENTO PARA LA CATEGORÍA "ROBOT LABERINTO"



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL



CAPÍTULO 1: DE LAS DISPOSICIONES GENERALES

Artículo 1.2 Cada delegación bajo autorización del representante podrá inscribir como máximo 2 equipos, los cuales estarán conformados por un máximo de 2 estudiantes.

Artículo 1.3 El jurado calificador podrá aplicar en cualquier circunstancia el presente reglamento y tendrá las atribuciones necesarias para decidir cualquier aspecto o eventualidad que no esté contemplada en el mismo.

Artículo 1.4 Todos los participantes deberán acogerse a lo estipulado en el Reglamento General en cuanto a inscripciones, participación y penalizaciones generales.

Artículo 1.5 El presente reglamento es una evolución sujeta a mejoras continuas que toma como referencia reglamentos presentados por las universidades anfitrionas en eventos anteriores y concursos realizados a nivel mundial.

CAPÍTULO 2: DE LOS REQUERIMIENTOS TÉCNICOS DEL PROTOTIPO.

Artículo 2.1 Todos los robots deberán sujetarse a las especificaciones técnicas detalladas en el presente capítulo. El incumplimiento de alguno de estos puntos será motivo de descalificación del robot ya sea en el preámbulo o durante la competencia.

- **Dimensiones:** Las dimensiones del robot están en función de las dimensiones de la pista, por lo tanto el robot no podrá tener dimensiones de largo y ancho mayores a 15cm y la altura de 20cm. El robot no puede expandir sus dimensiones hacia ninguna dirección durante la competencia.
- **Peso:** No existe restricciones de peso.

- **Alimentación:** La fuente de energía para alimentar los circuitos eléctricos de control y los motores del robot será interna, considerándose cualquier tipo de baterías de corriente continua
- **Control:** El robot no puede tener ningún tipo de control externo, es decir, el robot tiene que ser totalmente autónomo
- **Funcionamiento en la pista:** Se permite cualquier tipo de sensores para el sistema de navegación del robot, queda prohibido el uso de switches, fines de carrera u otros mecanismos de detección de contacto, es decir durante la navegación el robot no deberá tocar las paredes caso contrario recibirá una amonestación. Por otra parte solamente las ruedas pueden estar en contacto sobre la pista, el robot no puede dejar marcas o indicios del camino recorrido.

Artículo 2.2 Queda prohibido el uso de robots enteramente comerciales, sin embargo se puede utilizar tarjetas de desarrollo como partes del robot.

CAPÍTULO 3: DE LAS ESPECIFICACIONES TÉCNICAS DE LA PISTA.

Artículo 3.1 La pista será construida con madera de 5cm de espesor con un fondo blanco mate, las paredes tendrán un espesor y altura de 4mm y 20cm respectivamente y entre paredes de la pista habrá una distancia de 25cm. La madera sobre la cual se construirá la pista tendrá dimensiones de 2mx2m con una tolerancia de $\pm 10\%$. La figura 1 presenta un modelo de la pista con el fin de indicar las dimensiones, más no el modelo a utilizar en la competencia.

Artículo 3.2 Todas las intersecciones serán de 90° y el prototipo debe acoplarse para funcionar con la iluminación ambiental.

Artículo 3.3 El día anterior a la organización del evento el jurado calificador diseñará la pista y la pondrá a disposición de los organizadores, los cuales deberán construirla para el día indicado en el cronograma

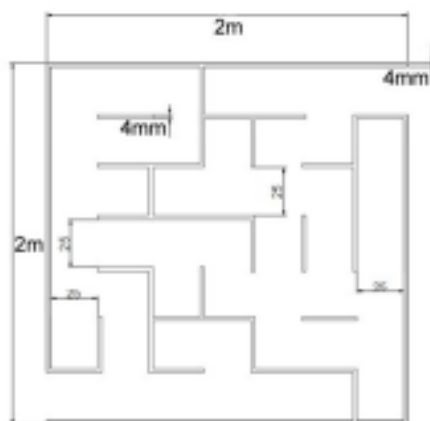


Figura 1. Dimensiones de la pista para la categoría Robot Laberinto

CAPÍTULO 4: DEL DESARROLLO DE LA COMPETENCIA.

Artículo 4.1 El concurso del robot de laberinto consiste en completar un recorrido de manera autónoma en el menor tiempo posible.

Artículo 4.2 En el preámbulo de la competencia el jurado calificador observará que los robots cumplan con las especificaciones que se establecen en los artículos del capítulo 2, además se observará la funcionalidad del mismo. Constituye motivo de descalificación en el preámbulo de la competencia los siguientes aspectos:

- Incumplimiento de las especificaciones técnicas
- El no presentarse en el preámbulo de la competencia tal cual dictamina el reglamento general.
- La no funcionalidad del robot
- La presencia de adhesivos, figuras o escritos que atenten con la moral

Artículo 4.3 El jurado llamará a pista a los participantes de acuerdo a las llaves establecidas, si este no se presentare el jurado hará un segundo

llamado transcurridos cinco minutos, si la situación persiste el participante quedará eliminado.

Artículo 4.4 El robot tiene dos intentos con un máximo de 5 minutos cada uno para completar el recorrido.

Artículo 4.5 Cada intento puede finalizar bajo las siguientes circunstancias:

- Con la culminación del recorrido por parte del prototipo.
- Cuando se haya cumplido el tiempo establecido.
- Cuando el participante lo decida e informe al jurado calificador.
- Cuando el jurado calificador observe que el robot no tiene movilidad durante un tiempo de 20 segundos.

Artículo 4.6 Los dos intentos se realizarán de manera consecutiva, sin que existe un período de tiempo para reparaciones o modificaciones. Si el participante lo decide puede no hacer uso de su segundo intento, este particular deberá informar al jurado calificador luego de haber culminado la primera participación.

Cuadro 1
PUNTAJES CATEGORÍA ROBOT LABERINTO

Parámetro	Puntajes
Terminar el recorrido	40 puntos
Recorre si retornar a la partida	10 puntos
Mantiene su movilidad durante todo el recorrido	10 puntos
Evita pasar más de 3 veces por el mismo recorrido	10 puntos
Termina el recorrido en menos de 1 minuto en cualquier intento	20 puntos
Por cada vez que toca la pared durante el recorrido	-10 puntos

Artículo 4.7 El desempeño del robot será evaluado mediante la modalidad de puntajes, el detalle de los parámetros evaluados y su respectivo puntaje se pueden ver en el cuadro 1.

Artículo 4.8 El desempeño del robot será evaluado por el jurado calificador y se considerará el puntaje más alto que se obtuvo de los dos intentos. Si existen empates de puntaje el ganador se determinará en base al robot que cumplió el recorrido en menor tiempo. Si los robots que llegaren a empatar no culminaron su recorrido total

el desempate se dará en base al mayor alcance que tuvieron.

Artículo 4.9 El responsable de un equipo puede alegar cualquier motivo de sospecha de incumplimiento de normativa por parte de su contrincante a cualquiera de los jueces. Si se produce una confirmación de dichas sospechas, el juez tendrá la facultad de eliminar al equipo implicado y sancionarlo de acuerdo a la puntuación general.

Artículo 4.10 En el caso de que el jurado descubra algún dispositivo que viole la reglamentación como por ejemplo algún dispositivo de radio frecuencia, el robot será eliminado y la universidad amonestada en cuanto a la premiación general.

ANEXO D HOJAS DE DATOS

CONTROLADOR DE POTENCIA

TOSHIBA

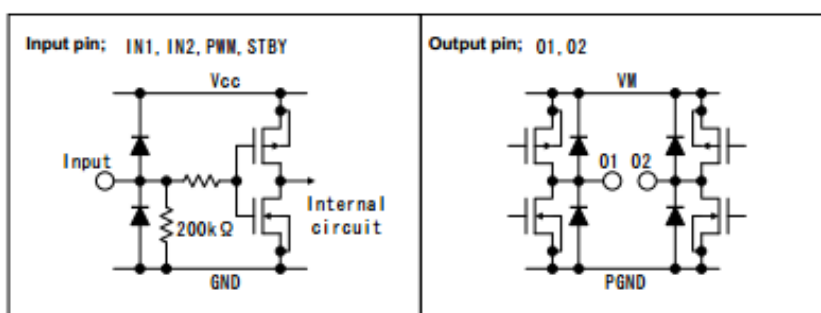
TB6612FNG

Absolute Maximum Ratings (Ta = 25°C)

Characteristics	Symbol	Rating	Unit	Remarks
Supply voltage	V _M	15	V	
	V _{CC}	6		
Input voltage	V _{IN}	-0.2 to 6	V	IN1,IN2,STBY,PWM pins
Output voltage	V _{OUT}	15	V	O1,O2 pins
Output current	I _{OUT}	1.2	A	Per 1 ch
	I _{OUT (peak)}	2		tw = 20 ms Continuous pulse, Duty ≤ 20%
		3.2		tw = 10 ms Single pulse
Power dissipation	P _D	0.78	W	IC only
		0.89		50 mm × 50 mm t=1.6 mm Cu ≥ 40% in PCB mounting
		1.36		76.2 mm × 114.3 mm t=1.6 mm Cu ≥ 30% in PCB mounting
Operating temperature	T _{opr}	-20 to 85	°C	
Storage temperature	T _{stg}	-55 to 150	°C	

Operating Range (Ta=-20~85°C)

Characteristics	Symbol	Min	Typ.	Max	Unit	Remarks
Supply voltage	V _{CC}	2.7	3	5.5	V	
	V _M	2.5	5	13.5	V	
Output current (H-SW)	I _{OUT}	—	—	1.0	A	V _M ≥ 4.5 V
		—	—	0.4		4.5 V > V _M ≥ 2.5 V Without PWM Operation
Switching frequency	f _{PWM}	—	—	100	kHz	



TOSHIBA

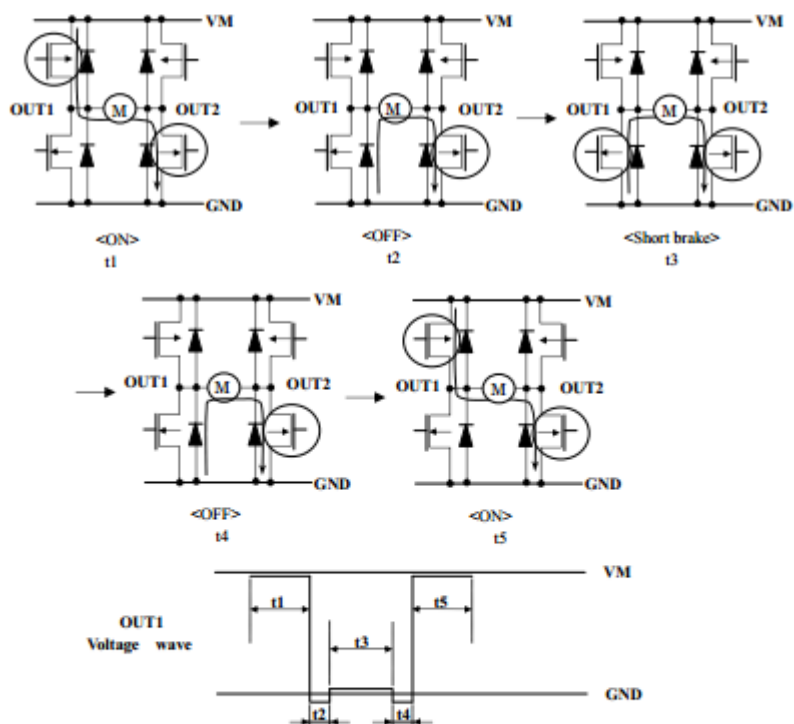
TB6612FNG

H-SW Control Function

Input				Output		
IN1	IN2	PWM	STBY	OUT1	OUT2	Mode
H	H	H/L	H	L	L	Short brake
L	H	H	H	L	H	CW
		L	H	L	L	Short brake
H	L	H	H	H	L	CW
		L	H	L	L	Short brake
L	L	H	H	OFF (High impedance)		Stop
H/L	H/L	H/L	L	OFF (High impedance)		Standby

H-SW Operating Description

- To prevent penetrating current, dead time t_2 and t_4 is provided in switching to each mode in the IC.



EMISOR INFRARROJO IR333



Technical Data Sheet

5mm Infrared LED , T-1 3/4

IR333/H0/L10

Features

- High reliability
- High radiant intensity
- Peak wavelength $\lambda_p=940\text{nm}$
- 2.54mm Lead spacing
- Low forward voltage
- Pb free
- The product itself will remain within RoHS compliant version.

Descriptions

- EVERLIGHT'S Infrared Emitting Diode(IR333/H0/L10) is a high intensity diode , molded in a blue transparent plastic package.
- The device is spectrally matched with phototransistor , photodiode and infrared receiver module.

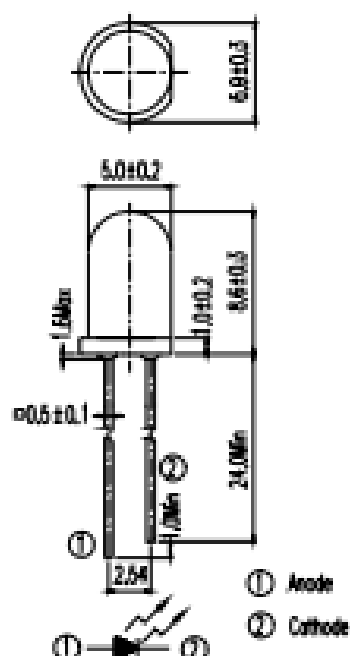


Applications

- Free air transmission system
- Infrared remote control units with high power requirement
- Smoke detector
- Infrared applied system

Device Selection Guide

LED Part No.	Chip	Lens Color
	Material	
IR	GaAlAs	Blue


IR333/H0/L10**Package Dimensions**

- Notes: 1.All dimensions are in millimeters
2.Tolerances unless dimensions $\pm 0.25\text{mm}$

Absolute Maximum Ratings ($T_a=25^\circ\text{C}$)

Parameter	Symbol	Rating	Units
Continuous Forward Current	I_F	100	mA
Peak Forward Current	I_{FP}	1.0	A
Reverse Voltage	V_R	5	V
Operating Temperature	T_{opr}	-40 ~ +85	$^\circ\text{C}$
Storage Temperature	T_{stg}	-40 ~ +85	$^\circ\text{C}$
Soldering Temperature	T_{sld}	260	$^\circ\text{C}$
Power Dissipation at(or below) 25 $^\circ\text{C}$ Free Air Temperature	P_d	150	mW

- Notes: *1: I_{FP} Conditions--Pulse Width $\leq 100\mu\text{s}$ and Duty $\leq 1\%$.
*2:Soldering time ≤ 5 seconds.

RECEPTOR INFRARROJO PD333



Technical Data Sheet

5mm Silicon PIN Photodiode , T-1 3/4

PD333-3B/H0/L2

Features

- Fast response time
- High photo sensitivity
- Small junction capacitance
- Pb free
- The product itself will remain within RoHS compliant version.

Descriptions

PD333-3B/H0/L2 is a high speed and high sensitive PIN photodiode in a standard 5 φ plastic package. Due to its black epoxy the device is sensitive to infrared radiation.



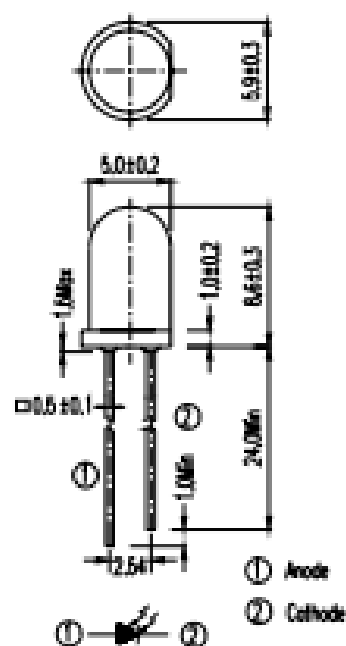
Applications

- High speed photo detector
- Security system
- Camera

Device Selection Guide

LED Part No.	Chip	Lens Color
	Material	
PD	Silicon	Black



PD333-3B/H0/L2
Package Dimensions


- Notes: 1.All dimensions are in millimeters
 2.Tolerances unless dimensions ± 0.25 mm

Absolute Maximum Ratings (Ta=25℃)

Parameter	Symbol	Rating	Units
Reverse Voltage	V_R	32	V
Power Dissipation	P_d	150	mW
Lead Soldering Temperature	T_{sol}	260	℃
Operating Temperature	T_{opr}	-40 ~ +85	℃
Storage Temperature	T_{sig}	-40 ~ +85	℃

Notes: *1:Soldering time ≤ 5 seconds.

MICROCONTROLADOR ATMEGA324PA

Features

- High-performance, Low-power AVR[®] 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
- Nonvolatile Program and Data Memories
 - 16/32/64K Bytes of In-System Self-Programmable Flash
Endurance: 10,000 Write/Erase Cycles
 - Optional Boot Code Section with Independent Lock Bits
In-System Programming by On-chip Boot Program
True Read-While-Write Operation
 - 512B/1K/2K Bytes EEPROM
Endurance: 100,000 Write/Erase Cycles
 - 1/2/4K Bytes Internal SRAM
 - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - 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
Differential mode with selectable gain at 1x, 10x or 200x⁽¹⁾
 - Byte-oriented Two-wire Serial Interface
 - Two Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - 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 RC 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
 - 32 Programmable I/O Lines
 - 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages
 - 2.7 - 5.5V for ATmega164P/324P/644P
- Speed Grades
 - ATmega164P/324P/644P: 0 - 8MHz @ 2.7 - 5.5V, 0 - 16MHz @ 4.5 - 5.5V
- Power Consumption at 8 MHz, 5V, 25°C for ATmega644P
 - Active mode: 8 mA
 - Idle mode: 2.4 mA
 - Power-down Mode: 0.8 µA

Notes: 1. Differential Mode is not recommended above 85°C.



**8-bit AVR[®]
Microcontroller
with 16/32/64K
Bytes In-System
Programmable
Flash**

**ATmega164P
ATmega324P
ATmega644P**

Automotive

Summary

