



UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
ESCUELA DE ELECTRÓNICA Y TELECOMUNICACIONES

“Trabajo de grado previo a la obtención del Título de Ingeniero en Electrónica y Telecomunicaciones”

TRABAJO DE GRADUACIÓN

Título del proyecto:

DISEÑO E IMPLEMENTACIÓN DE UN ROBOT PARA SERVICIO DE MENSAJERÍA INTERNA

AUTOR:

PABLO ANDRÉS FUENTES CAYAMBE

VIVIANA ODALIA PILCO CAJO

Director:

ING. Fabián Gunsha

Riobamba – Ecuador

AÑO 2014

Los miembros del Tribunal de Graduación del proyecto de investigación de título: **DISEÑO E IMPLEMENTACIÓN DE UN ROBOT PARA SERVICIO DE MENSAJERÍA INTERNA**, presentado por: **Pablo Andrés Fuentes Cayambe** y **Viviana Odalia Pilco Cajó** y dirigida por: **Ingeniero Fabián Gunsha**.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

Ing. Juan Carlos Cepeda
Presidente del Tribunal



Firma

Ing. Fabián Gunsha
Director del Proyecto



Firma

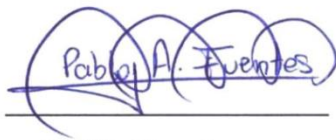
Ing. Aníbal Llanga
Miembro del Tribunal



Firma

AUTORÍA DE LA INVESTIGACIÓN

La responsabilidad del contenido de este Proyecto de Graduación, corresponde exclusivamente a: **Pablo Andrés Fuentes Cayambe, Viviana Odalia Pilco Cajo e Ingeniero Fabián Gunsha** y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo.



Pablo Fuentes

060402293-9



Viviana Pilco

060406822-1

AGRADECIMIENTO

Agradezco a Dios por cada nuevo día de vida, a mis padres por todo su esfuerzo realizado para darme una buena educación, a mi esposa e hija por el cariño y apoyo brindado, al Director y Asesor de tesis por las sugerencias efectuadas que permitieron el desarrollo exitoso del tema y a mi compañera Viviana Pilco por su dedicación y paciencia en la ejecución de este trabajo.

A todos mil gracias.

Pablo Andrés Fuentes Cayambe

AGRADECIMIENTO

Agradezco en primer lugar a Dios por bendecirme, darme la salud, fortaleza, y sabiduría necesaria para culminar una de mis metas tan anheladas.

A mis Padres quienes con sus consejos, valores y motivación constante me han permitido ser una persona de bien, además por siempre brindarme el apoyo absoluto, para poder realizar y terminar mis estudios con éxito. A mi director de tesis por su valiosa guía en este camino para la realización del proyecto de graduación.

Desde luego, mi agradecimiento a mi compañero Pablo Fuentes por su ayuda, paciencia, y contribución a la elaboración de la Tesis.

A mis amigos y todas las personas que estuvieron conmigo en los momentos difíciles y de felicidad, por su amistad, consejos y palabras de aliento para lograr tan ansiado sueño.

A todos ellos muchas gracias.

Viviana Odalia Pilco Cajo

DEDICATORIA

Dedico este trabajo a mis padres como muestra de agradecimiento por haber estado conmigo en el transcurso de mi desarrollo como profesional, a mi esposa e hija por ser la inspiración y la fuerza para salir adelante, por recibir su apoyo y cariño incondicional.

Pablo Andrés Fuentes Cayambe

DEDICATORIA

Con inmensa gratitud y amor dedico el presente trabajo a mis queridos Padres por ofrecerme confianza, respeto, por creer en mí, porque han pasado momentos difíciles siempre me han brindado su apoyo y amor incondicional. A mis hermanos por siempre acompañarme, estar presentes y ser el incentivo para seguir adelante.

A mi sobrino por formar parte de una de mis grandes alegrías en mi vida.

Y a todos aquellos que han colaborado para ver cristalizados mis aspiraciones y anhelos en la elaboración de la presente Tesis.

Viviana Odalia Pilco Cajo

ÍNDICE GENERAL

PORTADA	i
AUTORÍA DE LA INVESTIGACIÓN	iii
AGRADECIMIENTO I	iv
AGRADECIMIENTO II	v
DEDICATORIA I	vi
DEDICATORIA II.....	vii
ÍNDICE GENERAL.....	viii
ÍNDICE DE FIGURAS.....	xii
ÍNDICE DE TABLAS	xiv
RESUMEN.....	xv
SUMMARY	xvi
INTRODUCCIÓN	1
CAPÍTULO I.....	3
1.FUNDAMENTACIÓN TEÓRICA.....	3
1.1 La robótica.....	3
1.1.1 Clases de robots	3
1.2 Estructura general de un robot móvil	8
1.3 Sensor	9
1.3.1 Sensores de Posición y Detección	9
1.4 Microcontroladores	12
1.4.1 Características de los microcontroladores	12
1.4.2 Familias de microcontroladores.....	13
1.5 Android.....	15
1.5.1 Arquitectura de Android	15
1.6 App Inventor.....	16
CAPÍTULO II	18

2.METODOLOGÍA	18
2.1 TIPO DE ESTUDIO.....	18
2.2 POBLACIÓN Y MUESTRA	18
2.2.1 Población	18
2.2.2 Muestra	18
2.2.3 Hipótesis	19
2.3 OPERACIONALIZACIÓN DE VARIABLES.....	20
2.4 PROCEDIMIENTOS	21
2.4.1 Selección de elementos y dispositivos electrónicos	21
2.4.2 Diseño y construcción de la estructura	30
2.4.3 Diseño electrónico del robot.....	31
2.4.4 Diseño de circuitos impresos	40
2.4.5 Montaje del robot mensajero	40
2.4.6 Diseño del software	42
2.4.7 Diseño de aplicación.....	48
2.5 PROCESAMIENTO Y ANÁLISIS	60
2.5.1 Estudio y diseño del prototipo	61
2.5.2 Análisis de los sensores a utilizar	61
2.5.3 Diseño y administración de dispositivos	62
2.5.4 Comunicación del prototipo.....	62
2.5.5 Diseño de la interfaz para controlar al robot.....	62
2.5.6 Visualización de la interfaz.....	62
2.6 COMPROBACIÓN DE LA HIPÓTESIS	62
CAPÍTULO III.....	69
3.RESULTADOS.....	69
3.1 Pruebas en los Trayectos	69
3.1.1 Trayectoria 1	71
3.1.2 Trayectoria 2.....	75

3.1.3 Trayectoria 3	79
3.1.4 Trayectoria manual	83
3.1.5 Tiempos de Prueba de Trayectorias con y sin Obstáculos.....	86
CAPÍTULO IV	88
4.DISCUSIÓN	88
CAPÍTULO V	89
5.CONCLUSIONES Y RECOMENDACIONES.....	89
5.1 Conclusiones	89
5.2 Recomendaciones	91
CAPÍTULO VI.....	92
6.PROPOSTA.....	92
6.1 Título de la Propuesta.....	92
6.2 Introducción.....	92
6.3 Objetivos	92
6.3.1 General.....	92
6.3.2 Específicos	93
6.4 Fundamentación Científica – Técnica.....	93
6.5 Descripción de la Propuesta	94
6.6 Diseño Organizacional	94
6.7 Monitoreo y Evaluación de la propuesta.	95
CAPÍTULO VII	96
7.BIBLIOGRAFÍA	96
CAPÍTULO VIII.....	98
8.ANEXOS	98
8.1 Anexo 1	98
8.2 Anexo 2	105
8.3 Anexo 3	125
8.4 Anexo 4	129

8.5	Anexo 5	144
8.6	Anexo 6	147
8.7	Anexo 7	148

ÍNDICE DE FIGURAS

Figura 1. Robot explorador	4
Figura 2. Brazo Robótico	4
Figura 3. Robot Humanoide	5
Figura 4. Robot desactiva bombas	6
Figura 5. Esquema general de un robot móvil	8
Figura 6. Encoder	10
Figura 7. Sensor Ultrasónico	11
Figura 8. Microcontrolador	12
Figura 9. Arquitectura de Android	15
Figura 10. Esquema General de App Inventor	16
Figura 11. L298	22
Figura 12. Regulador de Voltaje	23
Figura 13. Evaluación de Sensores Ultrasónicos	23
Figura 14. Sensor Ultrasónico	24
Figura 15. Encoder	25
Figura 16. Colocación LCD	25
Figura 17. ATmega 2560	26
Figura 18. Bluetooth HC-06	27
Figura 19. Servomotor Truncado	28
Figura 20. Llanta	29
Figura 21. Buzzer	29
Figura 22. Dispositivo Celular	30
Figura 23. Diseño del Prototipo	31
Figura 24. Placa de Control	32
Figura 25. Puente H en Paralelo	33
Figura 26. Regulador L7805	34
Figura 27. Configuración Encoder	36
Figura 28. Buzzer de Alarma	37
Figura 29. Configuración Buzzer	38
Figura 30. Robot Mensajero	42
Figura 31. Diagrama de flujo trayectoria 1	43
Figura 32. Diagrama de flujo trayectoria 2	44

Figura 33. Diagrama de flujo trayectoria 3	45
Figura 34. Diagrama de flujo trayectoria manual	46
Figura 35. Diagrama de flujo control manual	47
Figura 36. Diagrama de flujo evasor.....	48
Figura 37. Pantalla inicial del proyecto.....	49
Figura 38. Entorno de Diseño	50
Figura 39. Diagrama de flujo Ingreso Clave.....	51
Figura 40. Diagrama de flujo operación del robot	52
Figura 41. Diagrama de flujo subrutina trayectoria 1	53
Figura 42. Diagrama de flujo subrutina trayectoria 2	54
Figura 43. Diagrama de flujo subrutina trayectoria 3	55
Figura 44. Diagrama de flujo subrutina trayectoria manual	56
Figura 45. Diagrama de flujo subrutina control manual	57
Figura 46. Diagrama de flujo subrutina evasor.....	58
Figura 47. Pantalla de Inicio	59
Figura 48. Pantalla de Control	60
Figura 49. Diagrama de procedimientos	61
Figura 50. Gráfica regiones críticas	67
Figura 51. Establecimiento del punto critico	68
Figura 52. Área Administrativa.....	70
Figura 53. Trayectoria 1	71
Figura 54. Trayectoria Promedio	75
Figura 55. Trayectoria 2.....	76
Figura 56. Trayectoria Promedio	79
Figura 57. Trayectoria 3.....	80
Figura 58. Trayectoria Promedio	83
Figura 59. Trayectoria Manual.....	84
Figura 60. Esquema Organizacional	95

ÍNDICE DE TABLAS

Tabla 1 . Familias de los Microcontroladores	14
Tabla 2. Operacionalización de la Variable Independiente	20
Tabla 3. Operacionalización de la Variable dependiente	21
Tabla 4.Comparación de Valores del Encoder y Formula	35
Tabla 5. Error absoluto en el encoder	36
Tabla 6. Datos Adquiridos de los tres Trayectos	63
Tabla 7. Datos Óptimos de Recorrido de los tres Trayectos.....	65
Tabla 8. Media de Muestras Aleatorias	65
Tabla 9. Pruebas de Trayectos	72
Tabla 10. Errores en distancia de la Trayectoria 1	73
Tabla 11. Error Promedio Total	74
Tabla 12. Trayectoria 1 Promedio.....	74
Tabla 13. Pruebas de Trayectos	76
Tabla 14. Errores en distancia de la Trayectoria 2.....	77
Tabla 15 . Error Promedio Total	78
Tabla 16. Trayectoria 2 Promedio.....	78
Tabla 17. Pruebas de Trayectos	81
Tabla 18. Errores en distancia de la Trayectoria 3.....	81
Tabla 19. Error Promedio Total	82
Tabla 20. Trayectoria 3 Promedio.....	82
Tabla 21. Prueba de Modo Manual	85
Tabla 22. Tiempo en Trayectoria 1	86
Tabla 23. Tiempo en Trayectoria 2.....	86
Tabla 24. Tiempo en Trayectoria 3	87

RESUMEN

El desarrollo del diseño e implementación de un robot para servicio de mensajería interna, fue realizado con el fin de mejorar el tiempo de entrega de archivos entre oficinas del área administrativa de la Facultad de Ingeniería.

El robot mensajero se propone como ayuda para el personal administrativo de la Universidad Nacional de Chimborazo, optimizando el tiempo que es requerido para realizar el trámite de un estudiante o actividad administrativa, que implique la entrega y recepción de documentos, además de un posible mejoramiento para el envío de otro tipo de archivos que permitan al personal cumplir con otras actividades.

Este trabajo fue desarrollado en varias etapas para su culminación como son: el estudio y diseño del prototipo a construir, la selección, análisis de los sensores y dispositivos a utilizar, el desarrollo de la interfaz de comunicación entre el robot y el usuario.

Cada etapa fue seguida meticulosamente, adquiriendo información del funcionamiento de los sensores y dispositivos alternos, que permiten un óptimo manejo del robot, evitando posibles errores y daños en el prototipo, por mal uso o conexiones erróneas, realizando así pruebas previas, para su correcta implementación en el dispositivo.

Finalmente la verificación de los datos adquiridos por los sensores y las trayectorias pre establecidas en el robot, dieron un promedio de la eficiencia que posee el dispositivo para realizar una trayectoria en el área administrativa, teniendo un margen de error e implementación aceptable.



Lic. Geovany Armas

22 de Octubre del 2014

SUMMARY

The design development and implementation of a robot for internal messaging service was developed in order to improve the delivery time of files among the administrative offices in the Faculty of Engineering.

The messenger robot is proposed as a help for the administrative staff at Universidad Nacional de Chimborazo, optimizing the time required for the processing of a student or administrative activity, it implies the delivery and reception of documents, and a possible improvement for sending other type of files that allow the staff to develop other activities.

For finishing this work, it was developed in several stages: the study and design of the prototype to be built, selection, analysis of sensors and devices to be used, the development of the communication interface between the robot and the user.

Each stage was meticulously followed, acquiring information about the performance of sensors and alternate devices which allow an optimal control of the robot, avoiding possible errors and damage to the prototype by misuse or erroneous connections, performing preliminary tests for the correct implementation of the device.

Finally, the verification of the data acquired by the sensors and the trajectories pre-established in the robot, produced an effectiveness average of the device for carrying out a trajectory in the administrative area. It had an acceptable error and implementation margin.



INTRODUCCIÓN

Con el avance de la tecnología, la aparición de la robótica en distintas áreas, han desarrollado diversas maquinas o dispositivos, Según (Rafael Barea, 2014) “estos dispositivos poseen algunas o todas de las siguientes capacidades: movilización, imitación, percepción, independencia e inteligencia artificial, contando para ello con sistemas de procesamiento de datos, control, actuadores y sensores que faciliten la percepción del entorno, permitiendo realizar trabajos específicos para los que fueron construidos” (p.4).

Esto ha llevado al desarrollo de robots móviles, Según (Miguel Sánchez, 2003) “los cuales responden a la necesidad de extender el campo de aplicación de la robótica, restringido inicialmente al alcance de una estructura mecánica anclada en uno de sus extremos” (p.3), incrementando la autonomía y limitando la intervención humana, creando así programaciones más complejas con trayectoria pre establecidas, según (Jaime Cid Monjaraz, 2003) “una vez realizada la planificación de la trayectoria, es necesario planificar movimientos concretos y controlar dichos movimientos para mantener al vehículo en la trayectoria planificada” (p.15), dando así la creación de robots móviles autómatas.

El presente proyecto expone el desarrollo de un robot para servicio de mensajería interna, este trabajo surge de la necesidad de optimizar el tiempo empleado en el área administrativa al entregar y receptor documentos para realizar un trámite o gestión.

La información obtenida mediante la recopilación de datos y resultados del robot mensajero, presentan el estudio que pretende diseñar e implementar un robot para el servicio de mensajería interna, así como la creación de un sistema de posicionamiento, la investigación de algoritmos para la generación de trayectorias y su correcta administración para la recepción de los diferentes documentos en los distintos entornos del área administrativa, de la Facultad de Ingeniería de la Universidad Nacional de Chimborazo.

La evaluación de dichos datos permiten la creación de un robot mensajero que consta de varias etapas, siendo la primera el diseño y selección de dispositivos para el prototipo, además de la evaluación de los actuadores, sensores y demás elementos utilizados, para las pruebas de funcionamiento, y la correcta ubicación de los mismos, que permiten el buen desempeño del robot.

Posteriormente empieza la construcción del robot, la programación para el procesamiento de datos, el control de entradas y salidas hacia los sensores y actuadores, así como la implementación de una interfaz de comunicación mediante la tecnología Bluetooth para comunicación de datos entre él y un dispositivo móvil.

Finalmente la creación de una aplicación para dispositivos móviles con el sistema operativo Android, la cual maneja el robot de una manera fácil, al ser intuitiva e interactiva, permitiendo el control y evaluación del dispositivo en las trayectorias previamente establecidas, donde el entorno no cambia, el cual puede presentar o no obstáculos hasta la culminación de la misma.

Estas etapas permitieron la creación del robot mensajero el cual tiene un comportamiento autónomo para tomar la mejor decisión y alcanzar su objetivo, para ello el robot tiene la capacidad obtener la información del entorno, conocer la ubicación del plano donde trabaja, localizarse y seguir la trayectoria, evitando los obstáculos, considerando las limitaciones físicas del mismo o del lugar, para llegar a un punto específico cuando la orden sea ejecutada.

CAPÍTULO I

1. FUNDAMENTACIÓN TEÓRICA

1.1 La robótica

La robótica es parte de la tecnología que trata del diseño, construcción, operación, estructura, y funcionalidad de los robots. De acuerdo con (Chiarani, Torres y Tapia, 2013) explican que la robótica combina distintas asignaturas como son: mecánica, electrónica, informática, inteligencia artificial y control.

Incluso la inteligencia artificial es aplicada con el propósito de comandar y controlar diferentes máquinas, siendo aptas para recibir y procesar la información exterior mediante su programación y sensores especializados.

Otro concepto de robótica de acuerdo a (Ricillo, 2012) “La robótica es un área interdisciplinaria, (...), básicamente es el desarrollo de máquinas que poseen cierta autonomía. Hay distintos grados, depende de la inteligencia que le va dando a cada uno; puede ser desde un brazo robótico que actualmente encontramos, por ejemplo en empresas automotrices hasta los humanoides.”(p.5)

1.1.1 Clases de robots

1.1.1.1 Robot móvil

El robot móvil posee la capacidad de trasladarse en un entorno determinado de acuerdo a la programación establecida.

Esta clase de robot es útil para realizar distintos tipos de trabajos, los cuales requieren tener una programación o trayectorias previamente establecidas que permiten al dispositivo desplazarse en un ambiente determinado, ejemplo de este, es el

robotenviado hacia el espacio para la captura de fotos y la toma de muestras espaciales como indica la figura 1.



Figura 1. Robot explorador

FUENTE:Robótica; construcción de robots móviles. [Imagen], obtenida 3 de julio del 2013, de <http://alextecnoso.wordpress.com/2011/12/20/robotica-1-construccion-de-robots-moviles-robotica-cableada-tecnologia-4o-eso/>

1.1.1.2 Robot industrial



Figura 2. Brazo Robótico

FUENTE: Herramientas Informáticas II. [Imagen], obtenida el 3 de julio del 2013, de <http://suarez-amador-guadalupe.blogspot.com/2011/05/tecnologia.html>

Como indica la figura 2, el robot industrial tiene diferentes movimientos programados, puede ser modificado por personas desde un ordenador, además posee la capacidad de manipular piezas y realizar diferentes tareas con movimientos secuenciales programados, es usado en la industria para procesos de fabricación. Los modelos más usados en la actualidad son los llamados brazos robóticos debido a su forma.

1.1.1.3 Robot humanoide

El robot humanoide adquiere su nombre de la similitud que existe con el ser humano, son fabricados con rasgos físicos, semejantes a los nuestros. Según (Baturone, 2011) manifiesta que esta clase de robot tiene la capacidad de poder interactuar con las personas y almacenar la información recopilada del entorno o la persona con la cual interactúa, realiza estas acciones mediante los diferentes tipos de sensores que están presentes a lo largo de su estructura, como indica la figura 3, esta clase de robots es el conocido como ASIMO de la compañía Honda.



Figura 3. Robot Humanoide

Fuente: Honda y su Robot Asimo. [Imagen], Obtenida el 3 de Julio del 2013, de <http://yomu.ws/honda-presenta-su-nuevo-robot-asimo/>

1.1.1.4 Robot teleoperador

El robot teleoperador es diseñado para recibir órdenes de un controlador en tiempo real. Según (Rodríguez, Sánchez y Atheortua, 2009) manifiesta que este robot obtiene el entorno mediante sensores especializados, que posibilitan la comunicación con el ser humano, puede ser comandado a grandes distancias y con gran precisión, siendo muy útil para situaciones potencialmente peligrosas como desarmar una bomba, adentrarse en medios contaminados. La figura 4 muestra un robot desactivador de bombas.



Figura 4. Robot desactiva bombas

Fuente: GOPE. [Imagen], Obtenida el 3 de julio del 2013, de <http://www.emol.com/noticias/nacional/2011/12/30/519389/gope-revisa-sucursal-bancaria-de-vitacura-tras-alerta-de-posible-bomba.html>

1.1.1.5 Los robots autónomos

El principal reto de la robótica móvil es la creación del comportamiento autónomo. Según (Constain, 2012) manifiesta que el robot al conocer su objetivo o lugar de destino, debe tomar la mejor decisión y alcanzarlo.

Este criterio es conocido como navegación, siendo uno de los principales temas de investigación en robótica móvil. Sin embargo, la navegación no es posible sin tener en cuenta cuatro bases principales: percepción, localización, planificación y mapeo.

- **Percepción:** Consiste en la información que adquiere el robot sobre el entorno, para eso emplea y procesa las señales de los sensores disponibles, tomando valores reales de cada espacio que está en el área que se desempeña.
- **Localización:** Hace referencia a la posición del robot en el plano. Es decir, debe saber la ubicación de todos los puntos que estructuran el plano donde trabaja. El problema de la localización será resuelto si la posición del robot la brindara un sistema GPS (Sistema De Posicionamiento Global). Sin embargo, la precisión de estos dispositivos tiene un error de unos cuantos metros, además, es indispensable que el robot este ubicado en un campo abierto donde no pierda la señal del dispositivo GPS. Además si el robot interactúa con personas, el robot debe poder identificarlos y definir su posición relativa con ellos.
- **Mapeo:** Hace mención a los datos del mapa o plano sobre el cual se traslada el robot, guardado previamente en su memoria. Esto proporciona al robot localizarse y realizar mejor la planificación de la trayectoria, evitando con mayor simplicidad los obstáculos y determinando la configuración del espacio.
- **Planificación:** Para llegar a un punto específico, partiendo de la ubicación real del robot. Este debe ser capaz de elegir el trayecto más eficaz para llegar al destino, evitando obstáculos durante el recorrido, considerando las limitaciones físicas del mismo o el entorno. Así, con un plano y el conocimiento del punto de llegada, el planteamiento de la trayectoria involucra identificar la ruta que hará que el robot llegue al punto de llegada cuando la orden sea ejecutada.

1.2 Estructura general de un robot móvil

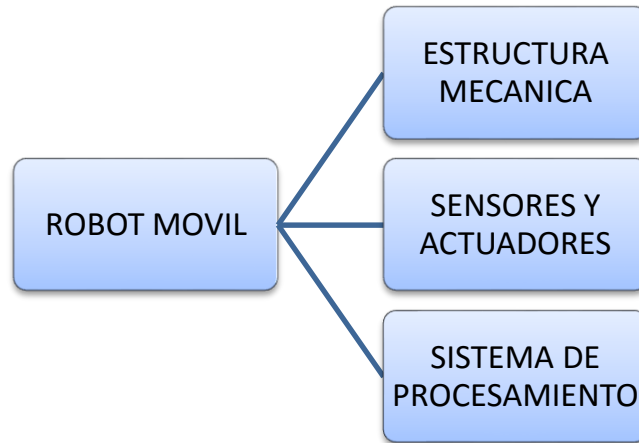


Figura 5. Esquema general de un robot móvil

Fuente: Los Autores

El esquema general de un robot móvil como indica la Figura 5, de acuerdo con (Ramírez, Ramírez, Ávila y Aguirre, 2013) resume lo siguiente:

- **Estructura mecánica:** Está formada por una serie de elementos estructurales sólidos, los cuales son contruidos, orientados al problema a resolver.
- **Sensores externos que captan una percepción del entorno:** Visión, tacto, audición, proximidad, etc.
- **Sensores internos que miden el estado de la estructura mecánica:** Giros, desplazamientos, velocidades, etc.
- **Actuadores:** Sistemas electromecánicos que generan fuerza y par para el movimiento, sistemas eléctricos, mecánicos o neumáticos.
- **Sistema inteligente o de procesamiento:** Es donde el robot envía las órdenes a los actuadores y reciben los datos o información de los sensores, asegura el funcionamiento correcto de los movimientos (bucles de control), planificación (trayectorias), etc. Son básicamente los procesadores, en ellos guardan los programas con los que va a funcionar el robot y así cumplir con sus tareas.

1.3 Sensor

Un sensor es un dispositivo eléctrico y/o mecánico diseñado para recibir información de una magnitud del exterior, transformando la magnitud física o química en un valor medible normalmente una señal eléctrica. La señal eléctrica es modificada por un sistema de acondicionamiento de señal, cuya salida es un voltaje. La magnitud puede ser: temperatura, presión, fuerza, inclinación, velocidad, etc.

Las principales características que poseen los sensores:

- **Rango de medida:** Es el valor mínimo y máximo en que el sensor puede captar la magnitud o variable con una mínima tolerancia de error.
- **Precisión:** Es el error de medida máximo esperado, entre la salida real comparada con la salida teórica.
- **Linealidad:** Es lineal, si existe una constante de proporcionalidad única que relaciona los incrementos de señal de salida con los correspondientes incrementos de señal de entrada, en todo el rango de medida.
- **Sensibilidad:** Un sensor tiene mayor sensibilidad, cuando mayor sea la variación de la salida producida por una determinada variación de entrada.
- **Resolución:** Indica la mínima variación de la variable de entrada que puede distinguirse a la salida.
- **Deriva:** Es la variación de la señal de salida, en un periodo de tiempo mientras conserva la variable y las condiciones ambientales, como la humedad, la temperatura y otras como el envejecimiento (oxidación, desgaste, etc.) del sensor.
- **Repetitividad:** Error esperado al repetir varias veces la misma medida de la variable

1.3.1 Sensores de Posición y Detección

A continuación algunos sensores de posicionamiento y detección.

1.3.1.1 Encoder

Un encoder o codificador es un dispositivo electrónico que permite modificar la posición angular de un eje establecido a datos o un código digital.

El encoder incremental proporciona una onda cuadrada cuya precisión dependerá de factores eléctricos y mecánicos, como es el error de división del disco, error de rodamiento y adherencia al suelo, factores de lectura de carácter electrónico y de tipo óptico como indica en la figura 6.

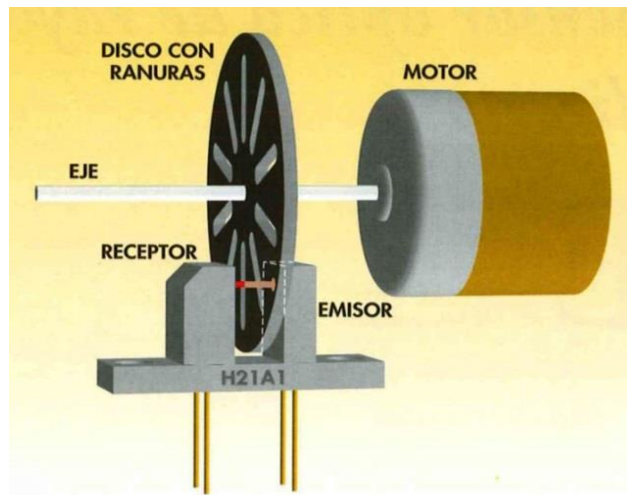


Figura 6. Encoder

Fuente: Sensor Óptico De Rayos Infrarrojos. [Imagen], Obtenida 3 de julio del 2013, de <http://www.comohacerturobot.com/Taller/taller-sensorOptico.htm>

1.3.1.2 Sensor ultrasónico

El sensor trabaja en función de los ultrasonidos (sonido), cuya frecuencia es mayor a la máxima expuesta a la audible por el oído humano desde 16 Hz a un aproximado de 20 KHz, a diferencia del ultrasonido que está estimado en los 40KHz.

De acuerdo con (Pérez, 2006) afirma:

El funcionamiento básico de los ultrasonidos, como medidores de distancia son observados en la figura 7, donde tiene un transmisor que emite un pulso de ultrasonido, que rebota sobre un determinado objeto y la reflexión de ese pulso es detectado por un receptor de ultrasonidos. (p.2)

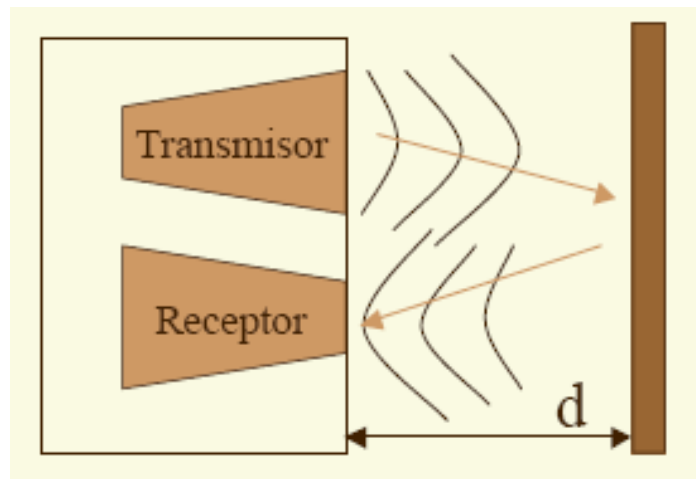


Figura 7. Sensor Ultrasónico

Fuente: Funcionamiento básico de los Ultrasonidos. [Imagen], Obtenido el 5 de Julio del 2013, de http://picmania.garcia-cuervo.net/recursos/redpictutorials/sensores/sensores_de_distancias_con_ultrasonidos.pdf=AFQjCNErsPCMkoralpg1vLgIHOMU4v-IuA&bvm=bv.70138588,d.cWc

Según (Pérez y Roldan, 2009) indican tres principios en los que se basa el funcionamiento de los sensores de ultrasonido:

- La propagación en medios homogéneos y no homogéneos.
- Reflexión en objetos móviles o fijos.
- Efecto Doppler. (p.3)

Princeton, con memoria/bus de datos y memoria/bus de programa compartidas.

- **Memoria de Programa:** Es una memoria ROM (*Read-Only Memory*), EPROM(*Electrically Programmable ROM*), EEPROM (*Electrically Erasable/Programmable ROM*) o Flash que almacena el código del programa que típicamente puede ser de 1 kilobyte a varios megabytes.
- **Memoria de Datos:** Es una memoria RAM (*Random Access Memory*) que típicamente puede ser de 1, 2, 4, 8, 16, 32 kilobytes.
- **Generador del Reloj:** Usualmente un cristal de cuarzo de frecuencias que genera una señal oscilatoria de entre 1 a 40 MHz, o también resonadores o circuitos RC.
- **Interfaz de Entrada/Salida:** Puertos paralelos, seriales (UARTs, *Universal Asynchronous Receiver/Transmitter*), I2C (*Inter-Integrated Circuit*), Interfaces de Periféricos Seriales (SPIs, *Serial Peripheral Interfaces*), Red de Área de Controladores (CAN, *Controller Area Network*), USB (*Universal Serial Bus*).
- **Otras opciones:**
 - Conversores Análogo-Digitales (A/D, *analog-to-digital*) para convertir un nivel de voltaje en un cierto pin a un valor digital manipulable por el programa del microcontrolador.
 - Moduladores por Ancho de Pulso (PWM, *Pulse-Width Modulation*) para generar ondas cuadradas de frecuencia fija pero con ancho de pulso modificable.(p.3,4.)

1.4.2 Familias de microcontroladores

Como muestra la tabla 1, están indicadas las principales familias de microcontroladores de las siguientes empresas: Atmel, Freescale, Holtek, Intel, National Semiconductor, Microchip, NXP Semiconductors, Renesas, STMicroelectronics, Texas Instruments y Zilog.

Tabla 1. Familias de los Microcontroladores

Empresa	8 bits	16 bits	32 bits
Atmel	AVR (mega y tiny), 89Sxxxx familia similar 8051		SAM7 (ARM7TDMI), SAM3 (ARM Cortex-M3), SAM9 (ARM926) AVR32
Freescale (antes Motorola)	68HC05, 68HC08, 68HC11, HCS08	68HC12, 68HCS12, 68HCSX12, 68HC16	683xx, PowerPC, ColdFire
Holtek	HT8		
Intel	MCS-48 (familia 8048) MCS51 (familia 8051) 8xC251	MCS96, MXS296	X
National Semiconductor	COP8	X	X
Microchip	Familia 10f2xx Familia 12Cxx Familia 12Fxx, 16Cxx y 16Fxx 18Cxx y 18Fxx	PIC24F, PIC24H y dsPIC30FXX, dsPIC33F con motor dsp integrado	PIC32
NXP Semiconductors (antes Philips)	80C51	XA	Cortex-M3, Cortex-M0, ARM7, ARM9
Renesas (antes Hitachi, Mitsubishi y NEC)	78K, H8	H8S, 78K0R, R8C, R32C/M32C/M16C	RX, V850, SuperH, SH-Mobile, H8SX
STMicroelectronics	ST 62, ST 7		STM32 (ARM7)
Texas Instruments	TMS370	MSP430	C2000, Cortex-M3 (ARM), TMS570 (ARM)
Zilog	Z8, Z86E02		

Fuente: Microcontrolador. [Tabla], obtenida el 7 de julio del 2013, de <http://es.wikipedia.org/wiki/Microcontrolador>

1.5 Android

Android es un sistema operativo creado en función de Linux para diversos dispositivos móviles como teléfonos, Tablets, google Tv y otros.

Según (Baz, Ferreira, Álvarez y García, 2009) aseguran. “El sistema busca, nuevamente, un modelo estandarizado de programación que simplifique las labores de creación de aplicaciones (...), busca que los programadores desarrollen sus creaciones una única vez y así ésta sea compatible con diferentes terminales”. (p.6)

1.5.1 Arquitectura de Android

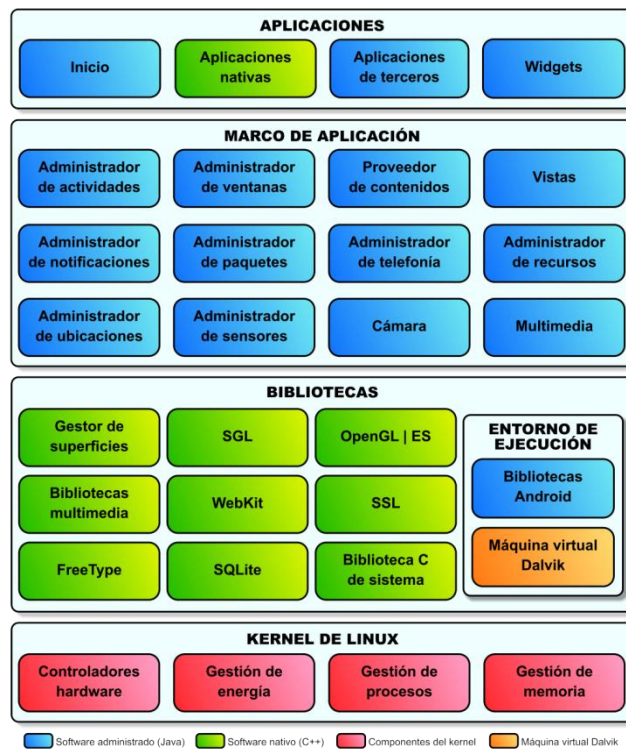


Figura 9. Arquitectura de Android

Fuente: Arquitectura de Android. [Imagen], obtenida el 8 de julio del 2013, de <http://columna80.wordpress.com/2011/02/17/arquitectura-de-android/>

En la figura 9 se observa, la arquitectura interna del sistema operativo de Android está compuesta de 4 capas: aplicaciones, almacén de aplicaciones, librerías y el Kernel/ Linux.

Según (Báez, et al., 2012) describe las capas de la siguiente forma:

“La capa que forma parte del propio Kernel de Linux, donde Android puede acceder a diferentes controladores, las librerías creadas para el desarrollo de aplicaciones Android, la siguiente capa organiza los diferentes administradores de recursos, y por último, la capa de las aplicaciones a las que tiene acceso”.(p.4)

1.6App Inventor

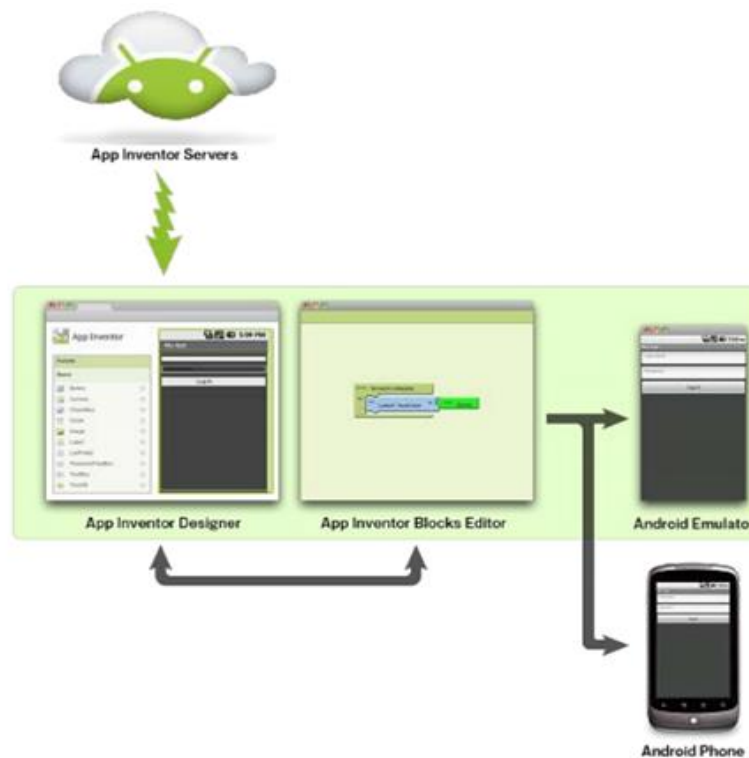


Figura 10. Esquema General de App Inventor

Fuente: App Inventor. [Imagen], obtenida el 7 de julio del 2013, de <http://www.kramirez.net/Robotica/Material/Presentaciones/AppInventor.pdf>

App inventor fue desarrollado por Google Labs, es una utilidad web que permite crear aplicaciones de forma visual e intuitiva, para el sistema operativo Android, sin la necesidad de programar en Java. Emplea una interfaz gráfica constituida por un conjunto de herramientas básicas que posibilita al usuario mover, soltar y enlazar una serie de bloques para crear la aplicación. Google termino con el desarrollo en el año 2011, transfiriendo el código al Instituto de Tecnología de Massachusetts (MIT), el cual puso a disposición del público de manera libre y gratuita.

Como indica la figura 10, App Inventor está compuesto de servidores, los cuales trabajan en línea, además consta del editor de diseño, editor de bloques y emulador.

Según la Universidad Autónoma del Occidente (UAO, 2012) expone las funciones de cada uno de sus componentes:

- **Entorno de diseño:** Permite la creación de los proyectos y su interfaz gráfica.
- **Editor de bloques:** Es el componente que “construye” la aplicación a partir de los bloques definidos en App Inventor, que implementan las estructuras de programación, funciones, etc.
- **Emulador:** Es el componente que permite la emulación de la aplicación móvil construida. Aún tiene ciertas limitaciones para el proceso de simulación, por ello es recomendable disponer de un teléfono de pruebas con plataforma Android.

El concepto de proyecto es manejado como en otras herramientas de programación y los archivos son guardados en los servidores de Google, de tal manera que el usuario puede retomar su proyecto desde prácticamente cualquier computador que tenga conexión a Internet y que cuente con una máquina virtual Java para ejecutar los componentes de la herramienta de programación. (p.3)

CAPÍTULO II

2. METODOLOGÍA

2.1 TIPO DE ESTUDIO

Investigativa.- Es un proceso de adquisición de información, conocimientos previos sobre electrónica y diseño de prototipos robóticos, además de métodos de posición y control de un robot autónomo mediante un sistema sustentable.

Activa o Aplicada.- Esta investigación realizada a la par del desarrollo del funcionamiento del dispositivo, al evaluar y calibrar constantemente los datos de los sensores adquiridos en cada rutina ejecutada en diversos escenarios y posibles superficies de prueba.

Campo.- Con las diversas pruebas desarrolladas en el escenario a emplearse y la obtención de datos mediante la implementación del mismo dando paso a la evaluación y análisis del correcto funcionamiento de los sensores y dispositivos de medición para un óptimo desarrollo en el entorno a aplicarse.

Además que la realización de este proyecto de grado prestara un servicio para mejorar la eficiencia en el área administrativa de la Universidad Nacional de Chimborazo.

2.2 POBLACIÓN Y MUESTRA

2.2.1 Población

Son los datos adquiridos de las diversas pruebas realizadas, en función de la distancia recorrida de cada uno de los trayectos establecidos.

2.2.2 Muestra

La muestra está establecida por el cálculo de población infinita o desconocida.

$$n = \frac{Z_{\alpha}^2 \cdot p \cdot q}{i^2} \quad (1)$$

Z_{α} Valor correspondiente a la distribución de gauss, $Z_{\alpha=0.05} = 1,96$

p Prevalencia esperada del parámetro a evaluar, de un 90%

$q = 1-p$

i Es el error que se prevé cometer, de un 10%

$$n = \frac{(1,96)^2 \cdot (0,9) \cdot (0,1)}{(0,1)^2}$$

$$n = \frac{0,345744}{0,01}$$

$$n = 34,57$$

$$n \approx 35$$

2.2.3 Hipótesis

“La creación del robot de mensajería interna ejecutará las trayectorias diseñadas desenvolviéndose eficazmente en tiempo y distancia, optimizando el tiempo de entrega- recepción de documentos del personal administrativo en el entorno de la Facultad de Ingeniería”

Al plantear la hipótesis del trabajo de graduación, están considerados los datos adquiridos, al realizar los múltiples trayectos establecidos en el prototipo y su precisión al culminar un recorrido, tomando en consideración posibles obstáculos presentes en el área a desempeñarse.

2.3 OPERACIONALIZACIÓN DE VARIABLES

Para el desarrollo de la operacionalización de variables, entro en consideración las variables independientes y dependientes del tema expuesto y su respectiva hipótesis.

A continuación como muestra la tabla 2, está indicada la operacionalización de la variable independiente del tema de tesis.

Tabla 2. Operacionalización de la Variable Independiente

VARIABLES	DIMENSIONES	INDICADORES	ÍTEMS
<p>Variable Independiente:</p> <p>Trayectoria</p>	<p>Precisión en la ejecución de un trayecto.</p>	<p>Pulsos de medición de Distancia (metros)</p> <p>Segundos</p> <p>Rango de Alcance (dBm)</p>	<p>Sensor Ultrasonico y Fototransistor</p> <p>Tiempo</p> <p>Comunicación Serial (Bluetooth)</p> <p>Aplicación Android</p>
<p>Dispositivo móvil</p>	<p>Empleo del Hardware y Software desarrollado.</p>	<p>APK en java</p>	

Fuente: Los Autores

Por consiguiente en la tabla 3, indica la operacionalización de la variable dependiente los cuales serán comprobados mediante procesos estadísticos para así cumplir los objetivos planteados en este trabajo.

Tabla 3. Operacionalización de la Variable dependiente

VARIABLES	DIMENSIONES	INDICADORES
<p>Variable Dependiente:</p> <p>Robot móvil para servicio de mensajería interna.</p>	<p>De acuerdo al requerimiento y entorno a aplicarse dentro de un área específica</p>	<p>Indicador Positivo</p> <p>Cumple con la Trayectoria Establecida</p> <p>Indicador negativo</p> <p>No Cumple con la Trayectoria Establecida</p>

Fuente: Los Autores

2.4 PROCEDIMIENTOS

Para el diseño e implementación de un robot para el servicio de mensajería, existe una serie de procesos a seguir, como se observa a continuación.

2.4.1 Selección de elementos y dispositivos electrónicos

Los elementos requeridos para el presente proyecto son los siguientes:

2.4.1.1 L298

Como indica la figura 11 el L298, es un circuito integrado de 15 terminales basado en el estándar TTL (lógica transistor a transistor), está formado de dos puentes H de

potencia que soporta una intensidad de hasta 2 A (amperios) por canal, la etapa de salida es una configuración en puente y sus salidas pueden conducir una carga inductiva en modo común o diferencial, dependiendo del estado de las entradas.

Los puentes son accionados por medio de 4 entradas y 2 habilitadores, cuando el habilitador tiene un pulso alto las entradas definen el estado del puente, pero si el habilitador está en bajo inhibe el puente. Por sus características y al poder incrementar la intensidad hasta 4 A, realizando una conexión en paralelo de sus salidas, se trabajó con este elemento, que permite controlar motores de corriente continua con PWM (modulación por ancho de pulsos), además de controlar el giro, ya sea hacia adelante, atrás o parar.

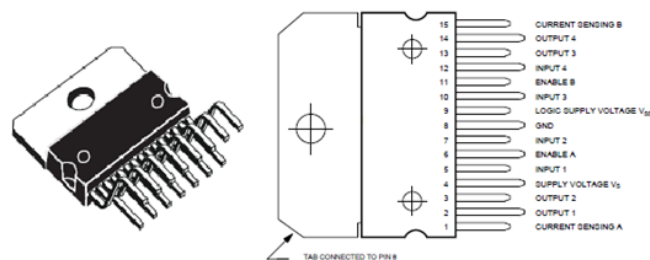


Figura 11. L298

Fuente: Los Autores

2.4.1.2 L7800 - Regulador de voltaje

Es un dispositivo que mantiene una tensión fija y estable en su salida, por ello al ser necesario mantener la tensión de 5 V (voltios), para alimentar el puente H, fue escogido el encapsulado de la serie LM 7805, este en un circuito integrado con la apariencia de un transistor, posee tres terminales, el primero corresponde a la entrada de voltaje no regulado desde 5 V hasta 18 V, el otro terminal es la salida de tensión regulada a 5 V y el último va conectado hacia tierra como indica la figura 12.

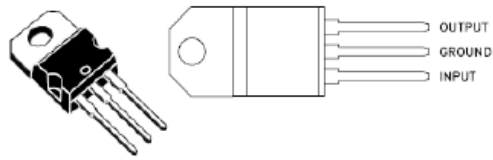


Figura 12. Regulador de Voltaje

Fuente: Los Autores

2.4.1.3 Ultrasonidos

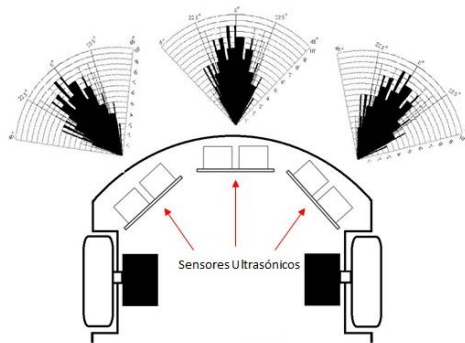


Figura 13. Evaluación de Sensores Ultrasonicos

Fuente: Los Autores

El sensor ultrasónico HC-Sr04 fue seleccionado, dado que su margen de error era bajo y su capacidad al evaluar objetos comprendía una distancia de cinco metros y un mínimo de dos centímetros; Al ser estos evaluados y determinando su forma de adquisición de datos que era en un rango de 30° y analizando la existencia de algún punto ciego en el transcurso de una trayectoria y evitar un posible error en detectar un obstáculo o impedimento, se colocó los sensores como muestra la figura 13.

A su vez estos sensores analizan objetos sólidos, de forma regular y en lo posible fijos porque que trabajaban con pulsos de ultrasonido que rebotan en el objeto, permitiendo determinar la distancia del obstáculo presente como muestra la figura 14, además del análisis mediante la configuración y programación, para la evasión del obstáculo.

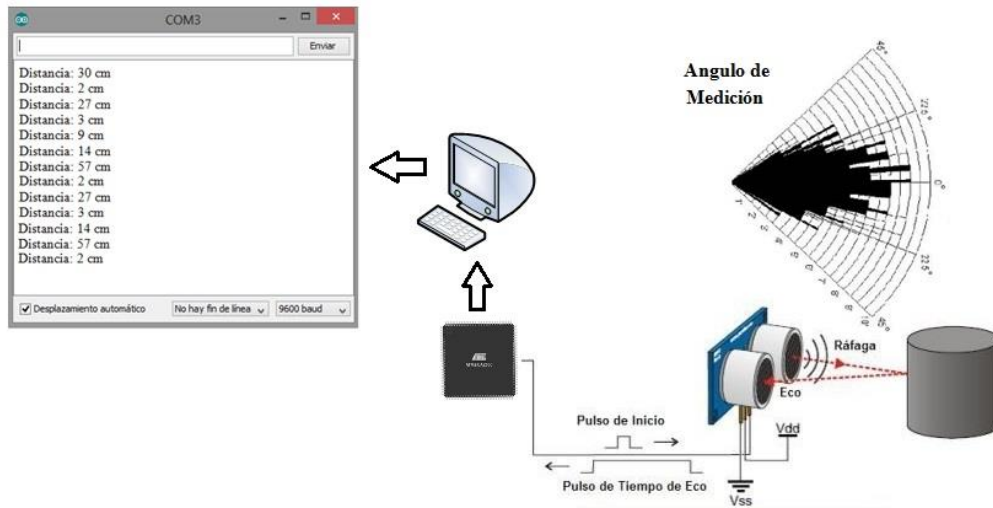


Figura 14. Sensor Ultrasónico

Fuente: Los Autores

2.4.1.4 Fototransistor o encoder óptico

El fototransistor está compuesto por el fotoemisor que es un diodo infrarrojo y un fotodetector formado por un fototransistor, su principal función es detectar la presencia de objetos.

Al realizar la configuración mostrada en la figura 15, obtiene en su salida una respuesta en el tiempo, cuando el fototransistor detecte un objeto este representara un voltaje en alto pero si no hay detección es bajo. Además como principal ventaja es el error mínimo en la detección considerando las recomendaciones de la hoja de especificaciones.

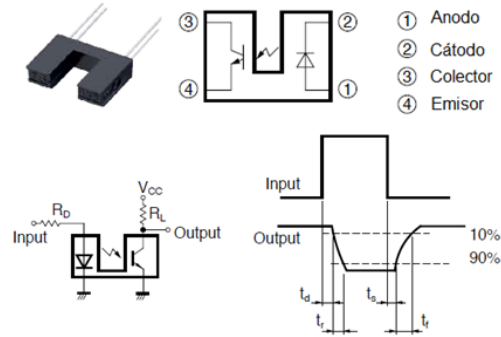


Figura 15. Encoder

Fuente: Los Autores

2.4.1.5 Pantalla de cristal líquido (LCD)

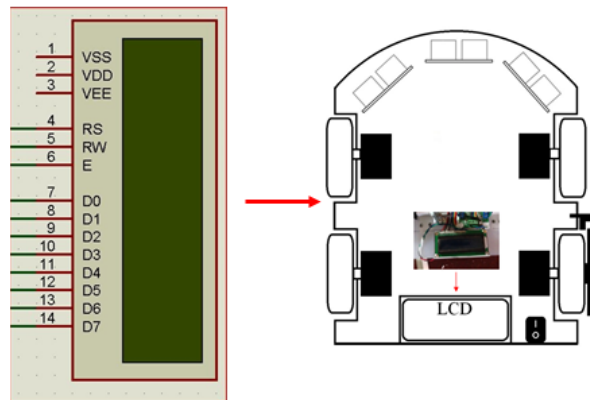


Figura 16. Colocación LCD

Fuente: Los Autores

Como muestra la figura 16, la necesidad de conocer la etapa que está realizando el robot o saber la distancia que recorrió, fue colocado un dispositivo de visualización de datos como es un LCD de 16x2, el cual está en la parte posterior del robot.

2.4.5.6 Microcontrolador ATmega 2560

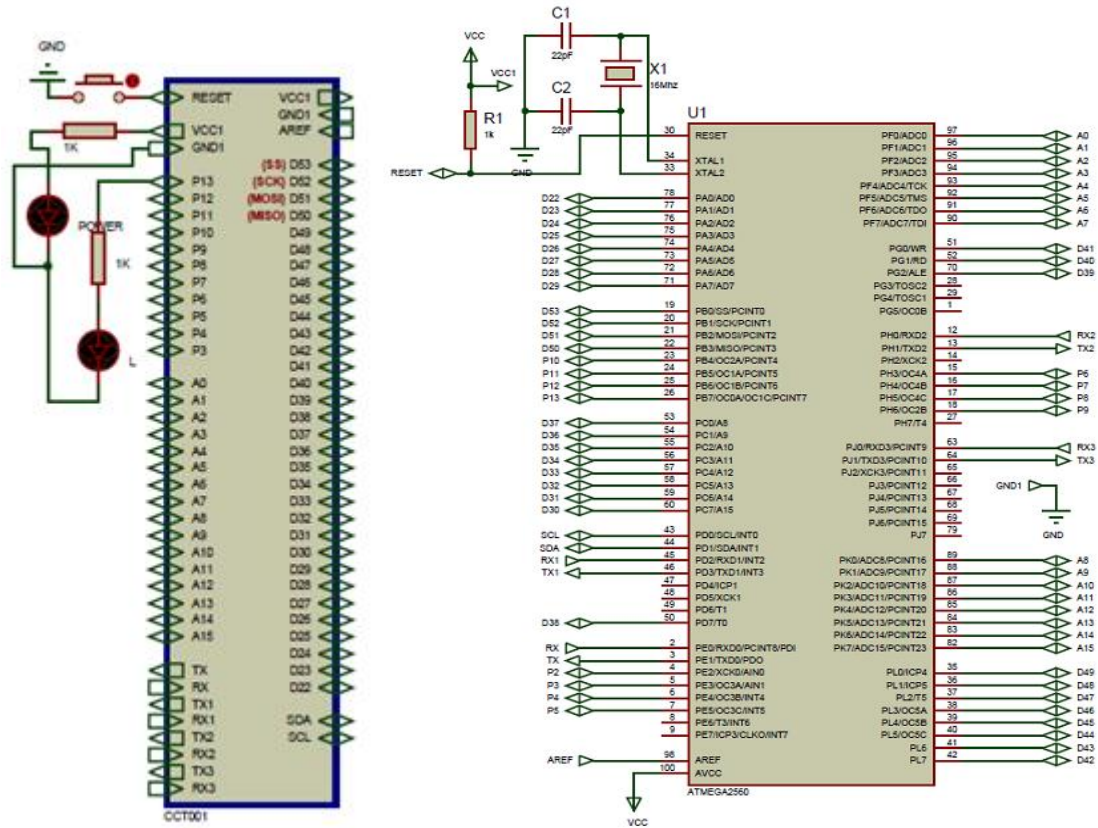


Figura 17. ATmega 2560

Fuente: Los Autores

El microcontrolador es el encargado de obtener, procesar y facilitar los datos necesarios para ejecutar las funciones del robot, está configurado de acuerdo a nuestros requerimientos como indica la figura 17, además tiene una gran capacidad de procesamiento y memoria, es el indicado dado que contempla el manejo de varios dispositivos y la administración innumerable de datos para el cumplimiento de las funciones.

2.4.5.7 Bluetooth

Debido a la necesidad de controlar el robot y no ser factible la utilización de pulsadores en el mensajero, siendo este un método rustico, se analizó otras posibilidades, considerando el control mediante una aplicación, la cual tiene que comunicarse por un medio confiable y seguro para la transmisión de datos, es así que por las características del dispositivo bluetooth HC-06 como muestra la figura 18, fue el idóneo para esta comunicación, el mismo que receptorá datos para el funcionamiento del robot, este dispositivo tiene la configuración de esclavo.



Figura 18. Bluetooth HC-06

Fuente: Tutorial Modulo Bluetooth HC-06. [Imagen], obtenida el 18 de julio de 2013, de <http://blog.make-a-tronik.com/tutorial-modulo-bluetooth-hc-06-parte-1/>

2.4.1.8 Servomotores

Para la movilización del robot se determinó trabajar con servomotores, dado el bajo peso, consumo de voltaje y corriente, debido a estas particularidades son más prácticos para el uso del mensajero, los servomotores trabajan en un rango de 180° de giro al poseer una parte electrónica, de tal manera fueron truncados; es decir fue

retirada su parte electrónica, eliminando el impedimento de giro y logrando a su vez manejarlos como simples motores de corriente continua.

A continuación como muestra la figura 19 está ilustrados pasos sencillos para truncar un servomotor y a su vez utilizarlo como un motor ordinario.



Figura 19. Servomotor Truncado

Fuente: Los Autores

- **Llantas**

La selección de las llantas dependió de factores como la adherencia para los distintos pisos donde realiza las pruebas y desempeña su trayecto, es así que fueron elaboradas 4 llantas de madera con un diámetro de 11cm, un grosor de 3cm, además de un recubrimiento de caucho que no permitiera deslizamiento en superficies lisas, dando así una adherencia aceptable para el andar del robot.

En cuanto al acople a utilizar en los servomotor fueron variados de acuerdo al tipo de llanta utilizada, como muestra la figura 20.

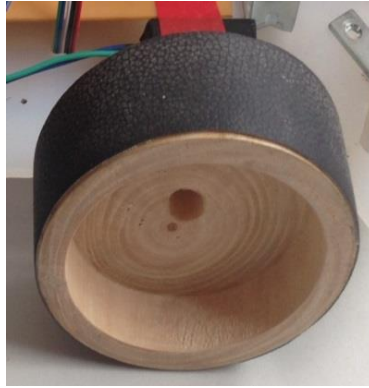


Figura 20. Llanta

Fuente: Los Autores

2.4.1.9 Buzzer



Figura 21. Buzzer

Fuente: Elementos Electrónicos. [Imagen], obtenida el 8 de julio del 2013, de http://www.futurlec.com/Buzzers_Flying_Leads.shtml

Como muestra la figura 21, se utilizó un buzzer de 5 V, para el tono de alarma del robot mensajero, el cual consta de un electroimán y una lámina metálica internamente, este a su vez fue conectado a un pin digital del microcontrolador.

2.4.1.10 Elementos electrónicos

Están conformados por los diversos elementos necesarios para la configuración de los dispositivos, como son resistencias, diodos, capacitores, cables de conexión, etc.

2.4.1.11 Dispositivo celular

Para el desarrollo de la aplicación fue necesario determinar un dispositivo de control para el robot mensajero, es así que fue requerido, un dispositivo celular como indica la figura 22, al ser el mismo de fácil adquisición y uso, el cual posee como software operativo Android, un sistema fácil de trabajar a nivel de aplicación y con múltiples plataformas para programar como es App inventor.



Figura 22. Dispositivo Celular

Fuente: Samsung Galaxy. [Imagen], obtenida el 8 de julio del 2013, de <http://gadgets.ndtv.com/samsung-galaxy-star-544>

2.4.2 Diseño y construcción de la estructura

El material de la estructura fue seleccionado debido a las fallencias encontradas en otro material como la madera, en las que por el peso y fragilidad no fueron óptimas para la construcción y elaboración del robot; al tratar de simplificar el

diseño se trabajó con acrílico, un material más resistente y fácil de utilizar, el cual fue recortado y adaptado con una estructura de metalde acuerdo a las medidas especificadas en la figura 23, esto facilito la implementación de un portapapeles y otros dispositivos.

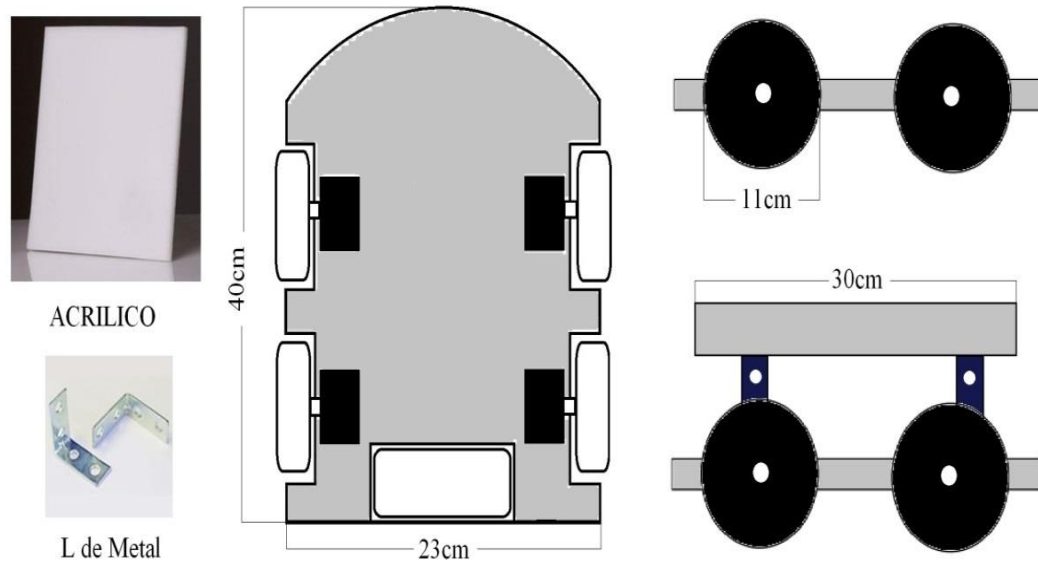


Figura 23. Diseño del Prototipo

Fuente: Los Autores

2.4.3 Diseño electrónico del robot

El diseño electrónico es la construcción y ubicación óptima de los dispositivos que comprenden el robot mensajero, también en esta etapa se realizará un breve análisis de los circuitos electrónicos que fueron implementados en el robot.

2.4.3.1 Circuito control general

El circuito de control fue diseñado con el fin de administrar todos los dispositivos electrónicos mediante el microcontrolador ATmega 2560, al cual está

conectado cada elemento mediante un pin, para ser controlado y administrado según el requerimiento de la programación y el robot, como la sección del encoder, comunicación bluetooth, sensores ultrasónicos, lcd, Buzzer y señales de control hacia los servomotores como indica la figura 24.

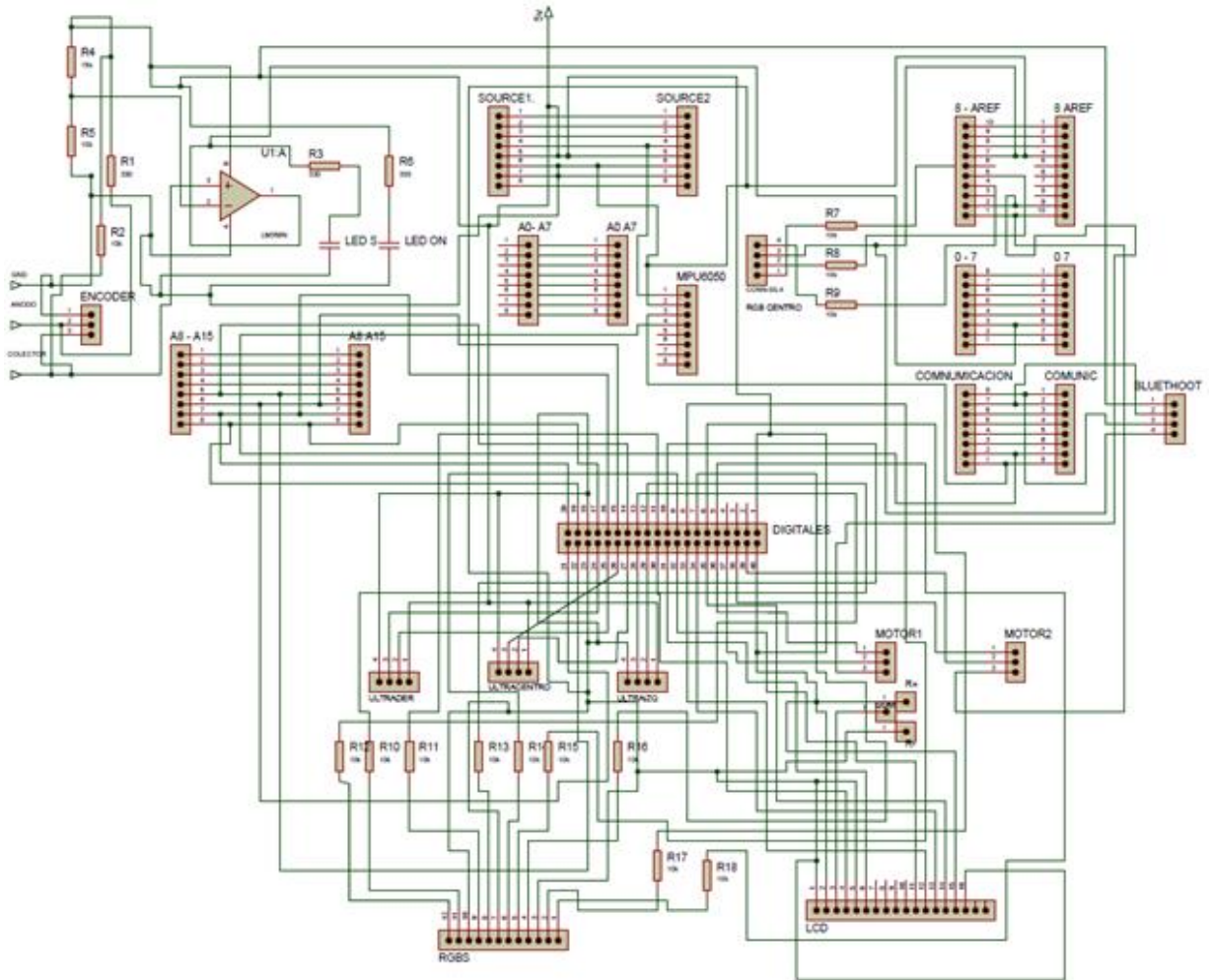


Figura 24. Placa de Control

Fuente: Los Autores

2.4.3.2 Circuito puente H

El control y manejo de los servomotores es administrado por un puente H, el cual permite a los servomotores girar en ambos sentidos y controlar a su vez la velocidad mediante pulsos PWM, dichos servos fueron administrados en paralelo cada lado, para de esta manera controlar dos servomotores por un puente H, se conectó los puentes H en paralelo, debido a que cada canal soporta un máximo de 2A y al estar dos servomotores en paralelo consumirán más del amperaje establecido, además de contar con un regulador de voltaje que permitirá mantener una tensión constante en nuestro robot, como muestrala figura 25.

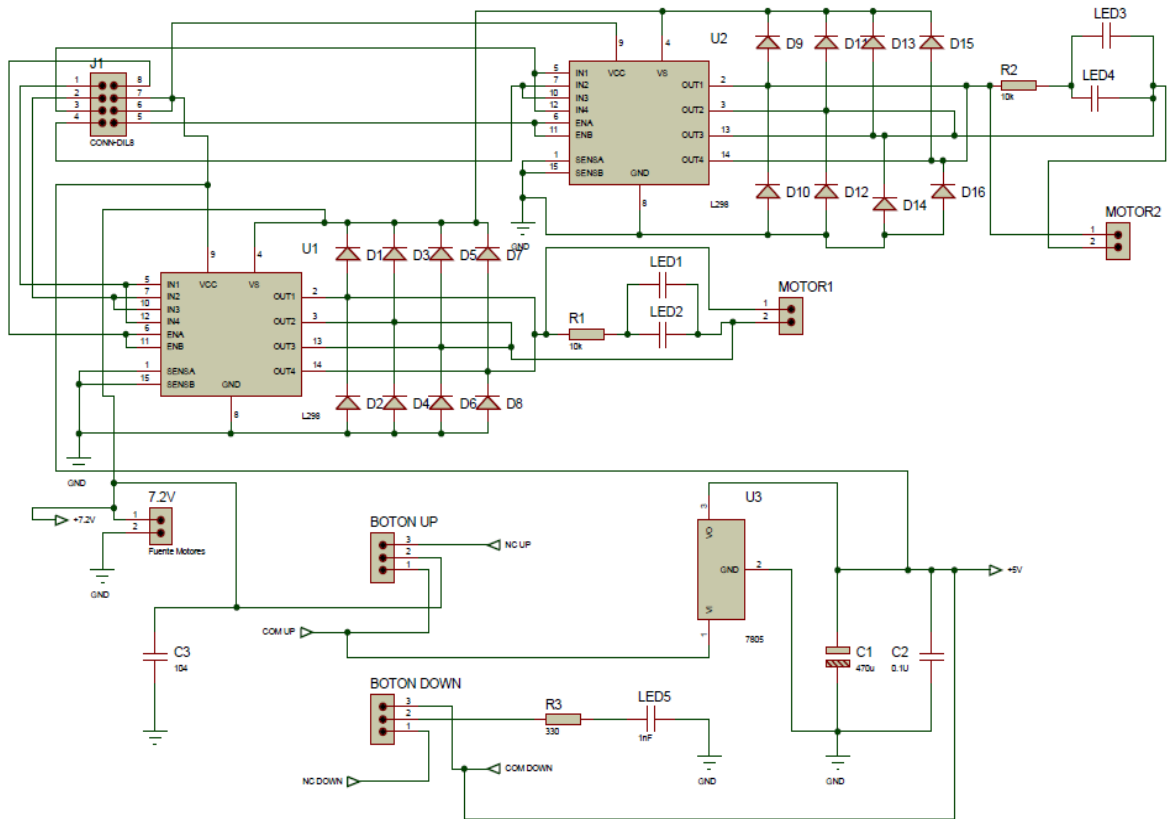


Figura 25. Puente H en Paralelo

Fuente: Los Autores

Este circuito presenta una protección de diodos, los cuales evitan la realimentación de corriente hacia el integrado L298, permitiendo la circulación de la corriente en un solo sentido en base al funcionamiento del driver

El driver requiere ser polarizado con 5 V para su funcionamiento interno, por lo que se incorpora un regulador de voltaje 7805 para extraer 5V de la misma alimentación de los servos, las entradas enable del driver son señales PWM que vienen desde el microcontrolador y hacen un control de voltaje para controlar la velocidad de los servomotores.

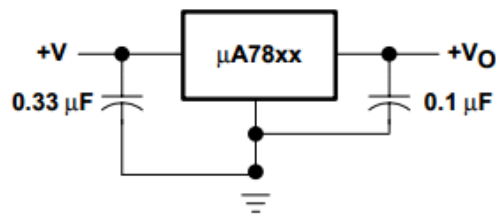


Figura 26. Regulador L7805

Fuente: Los Autores

El regulador L7805 que indica la figura 26, tiene capacitores en la entrada y salida del regulador, que funcionan como filtros para estabilizar la señal de voltaje.

2.4.3.3 Circuito encoder

Se trabajó con un encoder óptico, es decir un contador de pulsos que mediante un cálculo matemático permite saber la distancia recorrida del dispositivo, como muestra las siguientes ecuaciones:

Distancia Recorrida En Un Pulso

$$Distancia\ Recorrida\ en\ 1\ pulso = \frac{Rotacion\ Rueda \times Circunferencia}{Pulsos\ Totales\ de\ la\ Llanta} \quad (2)$$

$$Distancia\ Recorrida\ en\ 1\ pulso = \frac{R.R \times (\pi \times Diámetro)}{Pulsos\ Totales\ de\ la\ Llanta} \quad (3)$$

A continuación la tabla 4, muestra la comparación de los resultados teóricos obtenidos mediante la fórmula y los reales.

Distancia recorrida en un pulso

$$D.R.\ en\ 1\ pulso = \frac{1 \times (\pi \times 11\ cm)}{16}$$

$$D.R.\ en\ 1\ pulso = \frac{34.5575\ cm}{16}$$

$$D.R.\ en\ 1\ pulso = 2.15\ cm$$

Distancia recorrida total en una vuelta

$$D.R.Total = D.R.\ en\ 1\ pulso \times 16\ pulsos\ totales \quad (4)$$

$$D.R.Total = 2.15\ cm \times 16$$

$$D.R.Total = 34.4\ cm$$

Tabla 4.Comparación de Valores del Encoder y Formula

Vueltas de llanta	Encoder	Formula
Primera vuelta	35.6 cm	34.4 cm
Segunda vuelta	35.8 cm	34.4 cm
Tercera vuelta	35.5 cm	34.4 cm
Total Recorrido en 3 Vueltas	106.9 cm	103.2 cm

Fuente: Los Autores

Los errores establecidos mediante el encoder se muestran a continuación en la tabla 5, considerando que el error de la fórmula es 0%.

Tabla 5. Error absoluto en el encoder

Error absoluto en el Encoder			Error Total en 1m
1^{ra} vuelta	$35.6 - 35.633 = -0.33$	$(0.33/35.633) \times 100\% = 0.926\%$	1.767%
2^{da} vuelta	$35.8 - 35.633 = 0.167$	$(0.167/35.633) \times 100\% = 0.468\%$	
3^{ra} vuelta	$35.5 - 35.633 = -0.133$	$(0.133/35.633) \times 100\% = 0.373\%$	

Fuente: Los Autores

Cabe mencionar que el error absoluto del encoder incrementara de acuerdo a la distancia recorrida del robot, dando así un cálculo que permitía determinar la distancia en un trayecto establecido, y la utilización de la misma para determinar trayectos fijos o variables, para el envío y recepción de archivos.

A continuación en la figura 27, indica el diseño del circuito.

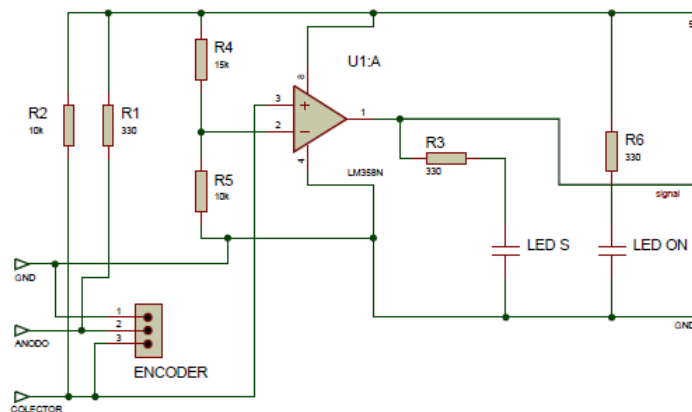


Figura 27. Configuración Encoder

Fuente: Los Autores

Este circuito está configurado como comparador de voltaje en lazo abierto, eso quiere decir que tiene ganancia infinita cuando la salida sea positiva entrega $V_{CC} = 5\text{ V}$ cuando la salida sea negativa entrega 0 V .

A continuación el cálculo del voltaje de referencia, para que el opam de un pulso ya sea alto o bajo.

$$V_{ref} = V_{CC} * \frac{R_4}{R_4 + R_5} \quad (5)$$

$$V_{ref} = 5V * \frac{15K\Omega}{15K\Omega + 10K\Omega}$$

$$V_{ref} = 5V * \frac{15K\Omega}{25K\Omega}$$

$$V_{ref} = 3V$$

Cuando la tensión de entrada V_{in} sea mayor que la de referencia, el $V_{in} - V_{ref}$ dará un número positivo y por lo tanto la tensión de salida estará a nivel alto (cercano a $5V$), cuando la tensión de entrada V_{in} sea menor que la de referencia, el $V_{in} - V_{ref}$ dará un número negativo y por lo tanto la salida estará a nivel bajo (cercano a 0 voltios).

2.4.3.4 Circuito buzzer

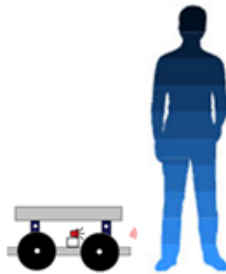


Figura 28. Buzzer de Alarma

Fuente: Los Autores

Este circuito presenta un transistor NPN 2222A que trabaja en corte y saturación, un buzzer que opera a 5V y necesita 20mA para funcionar, además de una resistencia interna de 250 ohm, como indica la figura 29.

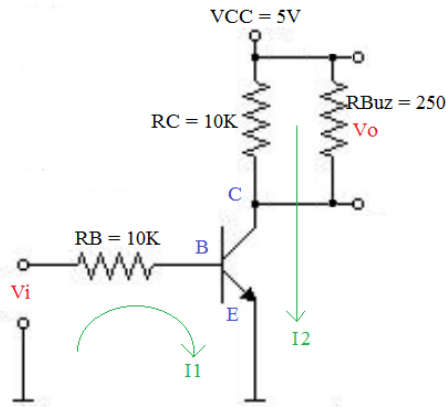


Figura 29. Configuración Buzzer

Fuente: Los Autores

Para este circuito es necesario conocer la resistencia de la base, para lo cual establece un valor a la resistencia $R_C = 10k$, que mediante un cálculo permitirá conocer el valor de R_B , como está indicado en el cálculo 6, 7 y 8:

$$R_{Ceq} = \frac{1}{\frac{1}{R_C} + \frac{1}{R_{Buz}}} \quad (6)$$

$$R_{Ceq} = \frac{1}{\frac{1}{10K} + \frac{1}{250}}$$

$$R_{Ceq} = \frac{1}{\frac{41}{10K}}$$

$$R_{Ceq} = \frac{10000}{41}$$

$$R_{Ceq} = 243.90 \Omega$$

Cuando la salida sea un “1”, habrá 5V en la entrada del circuito y se desea que esa condición sature al transistor es decir una corriente levemente superior a la que produce que $V_{CE} = V_{CEsat}$.

$$V_{CEsat} = V_{CC} - (\beta * I_B) * R_C \quad (7)$$

$$V_{CEsat} = V_{CC} - \beta \left(\frac{V_i - 0.7V}{R_B} \right) * R_C$$

$$0.3V = 5V - 100 \left(\frac{5V - 0.7V}{R_B} \right) * 243.90 \Omega$$

Despejando R_B , se obtiene el máximo valor con el que alcanza la saturación dada una entrada “1”:

$$R_B = \left(\frac{\beta_{min} * R_C}{V_{CC} - V_{CEsat}} \right) * (V_i - 0.7V) \quad (8)$$

$$R_B = \left(\frac{100 * 243.90 \Omega}{5V - 0.3V} \right) * (5V - 0.7V)$$

$$R_B = \left(\frac{24390 \Omega}{4.7V} \right) * (4.3V)$$

$$R_B = 22.31 K\Omega$$

El valor ideal de la resistencia es $22.31 K\Omega$, pero se optó por conectar una resistencia $10 K\Omega$, que es el estándar que manejan los microcontroladores para salidas digitales.

Otro aspecto importante es que entre los terminales de entrada B y E , se observa el comportamiento de un diodo, de modo que existirá una diferencia de potencial V_{BE} que rondará los $0,7 V$. Si dicha tensión cae por debajo de este valor, la corriente

I_B será prácticamente nula, dando lugar al corte del transistor y consecuentemente también lo será I_C .

2.4.4 Diseño de circuitos impresos

El diseño de circuitos impresos o placas fue realizado en proteus 8, tomando en consideración una serie de pasos para su elaboración, seleccionado los elementos a utilizar y los requerimientos del robot.

- El primer paso es realizar un esquema del circuito requerido para cumplir con una función determinada.
- Con el esquema del circuito, se ingresa al software para el diseño de placas, en donde empieza un nuevo proyecto y la búsqueda de los elementos en proteus, considerando que cada elemento debe tener PCB para la elaboración de la placa.
- Se realiza la conexión entre los elementos según el esquema antes establecido.
- Después ingresando a ARES de proteus, empieza el diseño de pistas, donde están todos los elementos antes conectados en un esquema acomodado, evitando puentes y sobre posiciones de elementos.
- Finalmente selecciona el grosor de las pistas para cada conexión, en especial para voltaje y tierra las cuales serán más gruesas que las demás, por el voltaje de ingreso y la corriente que circula por dichas pistas.

La placa de control y la placa del puente H se puede observar en el anexo 5.

2.4.5 Montaje del robot mensajero

El robot tuvo una serie de pasos y observaciones, las cuales establecieron la óptima posición de cada uno de los elementos que conforman el robot mensajero.

- **Servomotores:** Los servomotores fueron ubicados simétricamente a los costados del robot para su movilización y distribución del peso como muestra la figura 30.

- **Llantas:** Las llantas realizadas en madera y con un recubrimiento de caucho, fueron colocadas mediante un acople a cada uno de los servomotores como indica la figura 30.
- **LCD:** Fue incorporado a la estructura del robot para la visualización del recorrido realizado por parte del móvil como muestra la figura 30.
- **Encoder:** Está incorporado en la llanta posterior derecha, la misma que tiene adaptada un disco con ranuras que permiten el conteo al encoder.
- **Sensores Ultrasónicos:** Están colocados mediante “L’s” de metal en la parte delantera del robot con una separación de 30° entre cada sensor para evitar posibles puntos ciegos.
- **Placa de Control:** Es considerado el cerebro del robot móvil ya que en esta recae toda la programación y rutinas que debe cumplir el dispositivo, está ubicada en la parte central del robot como muestra la figura 30.
- **Puente H:** Fue colocado en la parte inferior del robot, para una mejor administración de dispositivo, ya que la placa de control está ubicada en la parte central del mismo.
- **Bluetooth:** Esta colocado en la parte delantera, para una mejor recepción de señales del robot, y que no exista interferencial con algún dispositivo del robot.
- **Buzzer:** Fue incorporado en la parte superior de la placa para emitir una alarma al llegar a su destino o encontrar algún obstáculo.
- **Alimentación:** Esta distribuida en dos partes, la primera está ubicada en la parte inferior del vehículo y proporciona una alimentación 7.2 V y 1.3 A hacia la placa de control, sensores, pantalla de visualización y demás dispositivos del robot; la segunda está ubicada al extremo izquierdo del robot, y alimenta a los servomotores con 7.2 V y 2.3 A mediante el puente H, todo esto en un arreglo de pilas en serie, que permiten mantener la corriente y voltaje necesario para el dispositivo.

Todos estos elementos conforman el robot mensajero, el cual culmina de la siguiente forma y presentación como indica la figura 30.

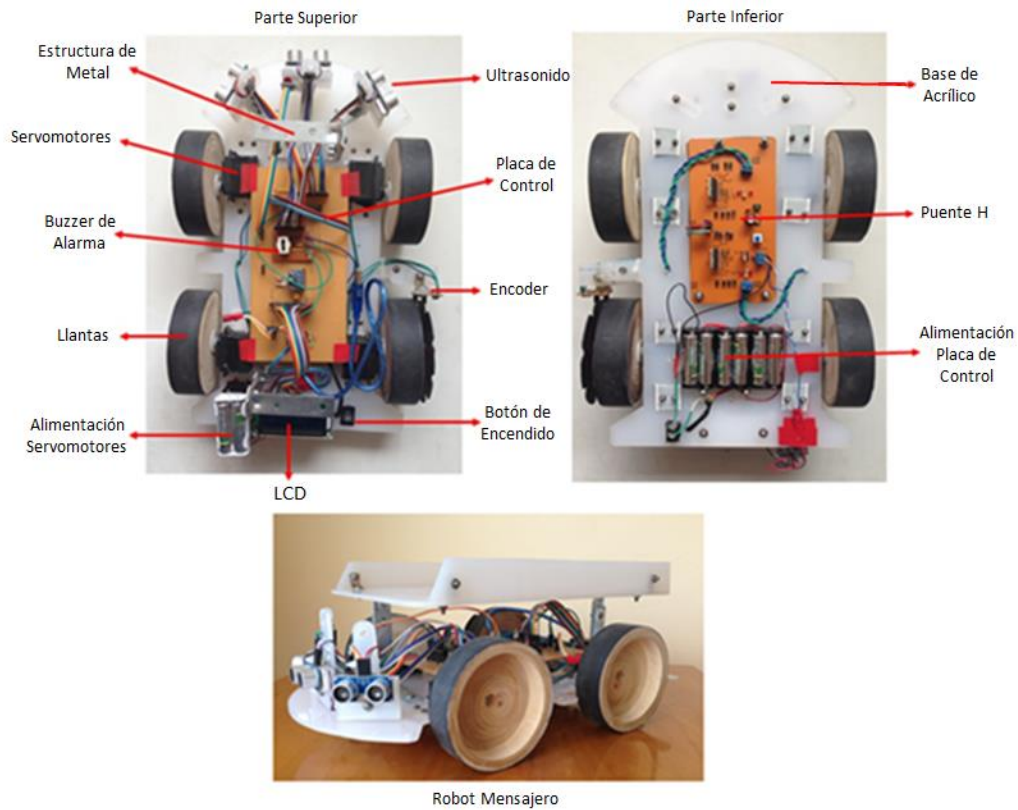


Figura 30. Robot Mensajero

Fuente: Los Autores

2.4.6 Diseño del software

La programación del robot esta especificada en base a funciones, las cuales son administradas posteriormente en una aplicación como muestra el anexo 2, que permite un óptimo funcionamiento del robot mensajero de acuerdo a los requerimientos deseados.

2.4.6.1 Programación microcontrolador

Los parámetros y datos obtenidos en los escenarios establecidos fueron analizados, para desarrollar un programa que funcionara correctamente con el manejo de la aplicación, todo esto desarrollado en una plataforma para programar el

microcontrolador como es ARDUINO, a continuación el programa del robot mensajero con sus diversas etapas y funciones, las cuales desempeñadas correctamente implican un óptimo funcionamiento en diferentes entornos.

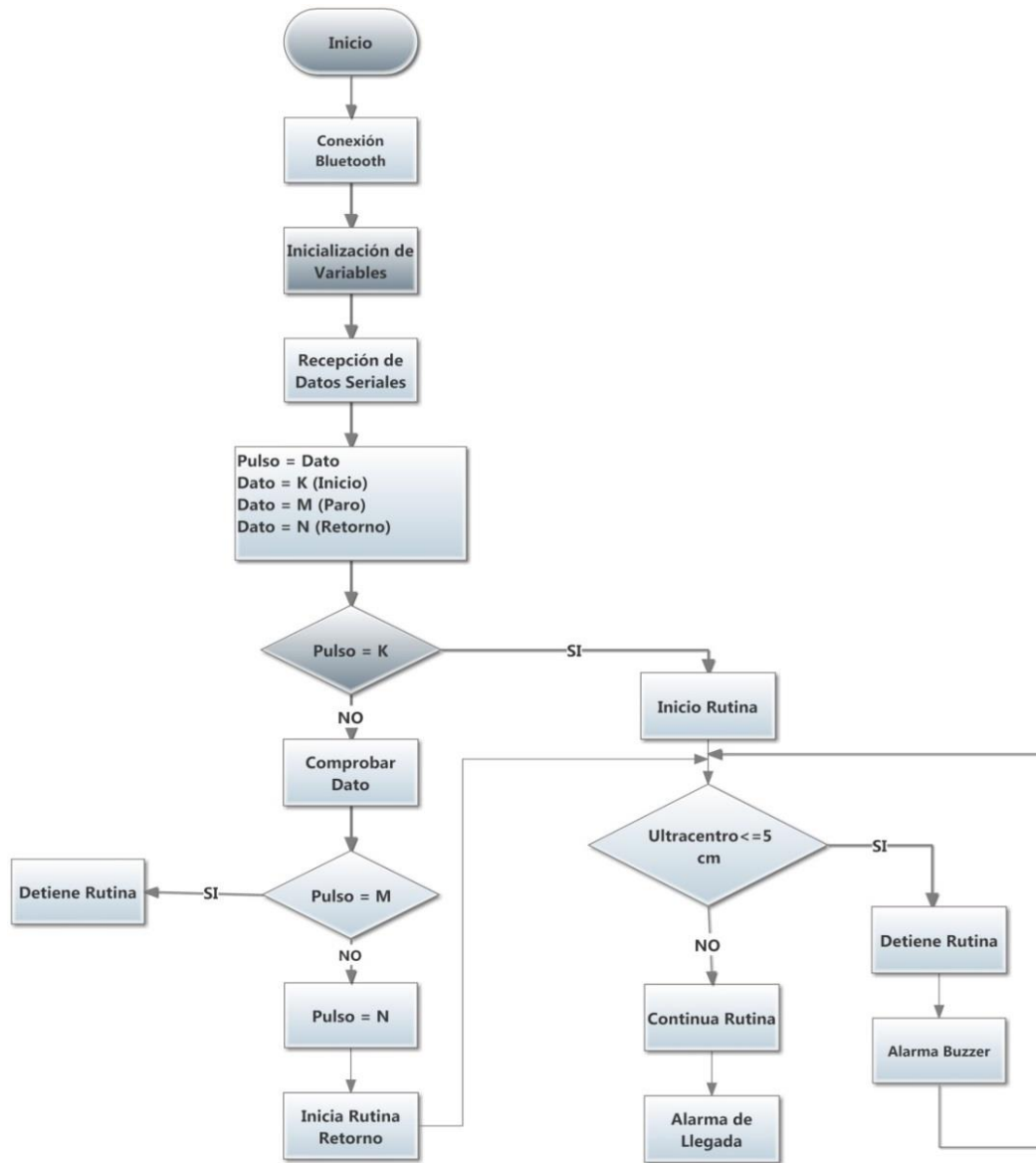


Figura 31. Diagrama de flujo trayectoria 1

Fuente: Los Autores

Las primeras tres rutinas como muestran los diagramas de flujo 31, 32 y 33, son trayectorias pre establecidas en la memoria del microcontrolador las cuales son cumplidas paso a paso sin obviar ningún detalle del trayecto hasta culminarlo.

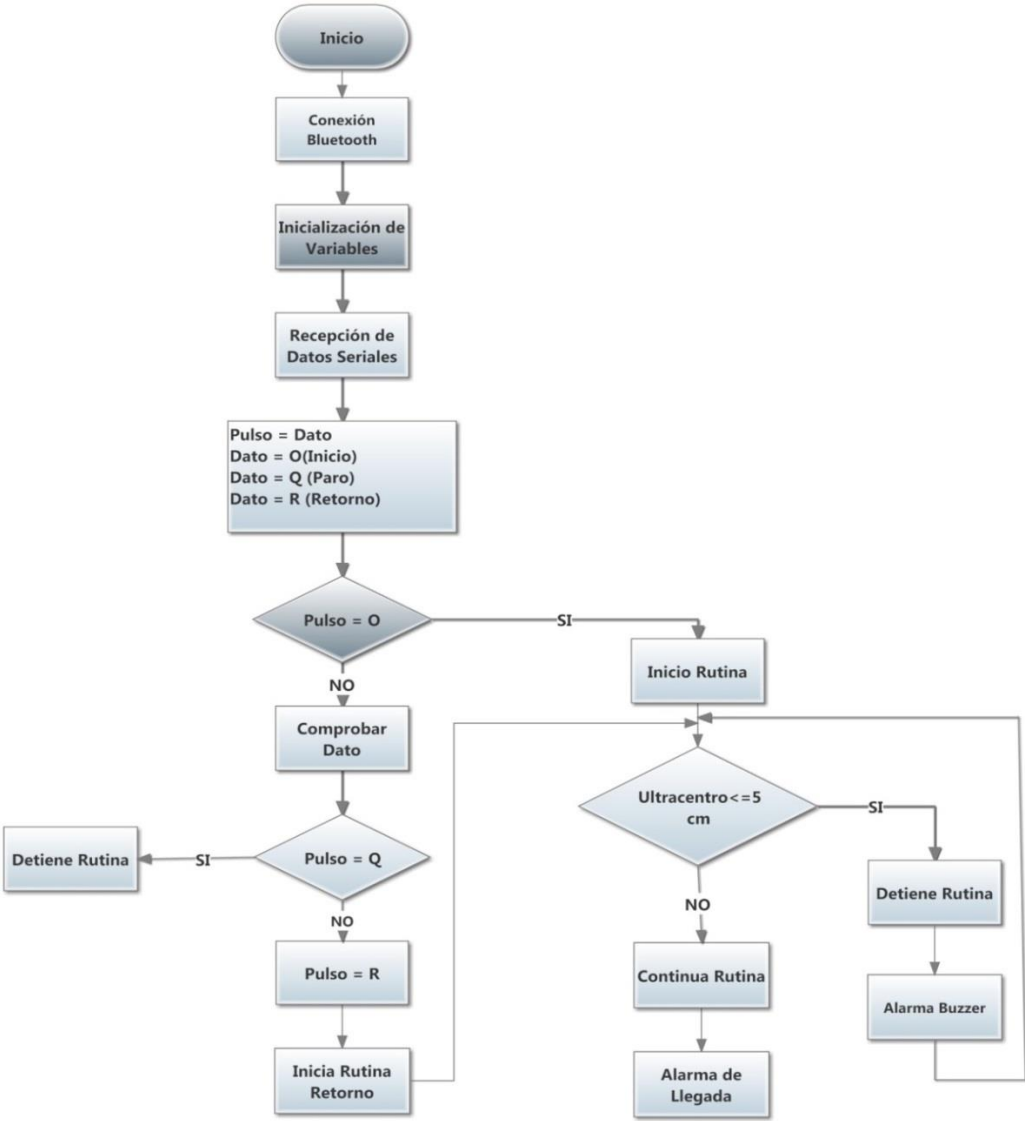


Figura 32. Diagrama de flujo trayectoria 2

Fuente: Los Autores

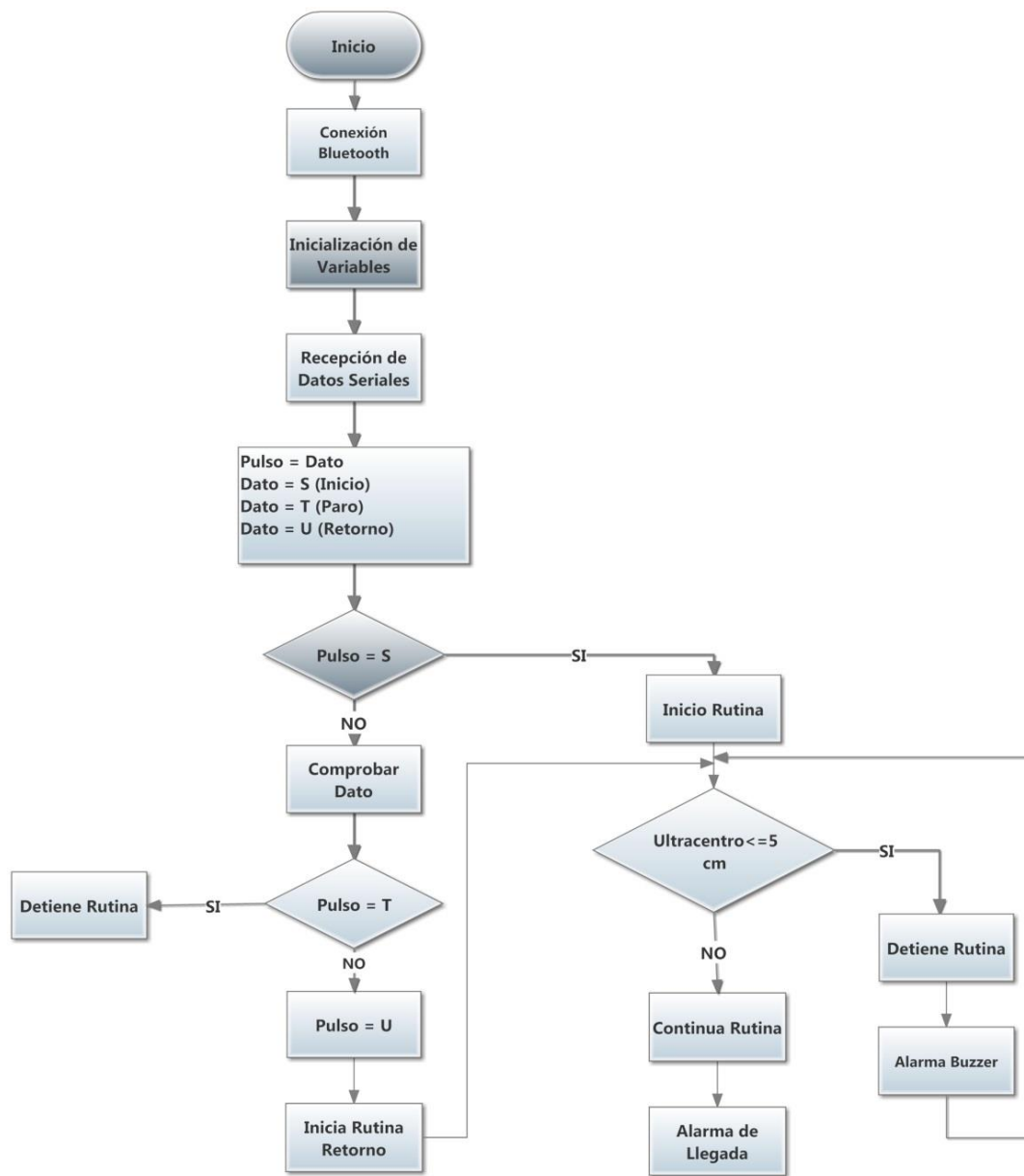


Figura 33. Diagrama de flujo trayectoria 3

Fuente: Los Autores

La rutina cuatro comprende una trayectoria que almacena datos en la memoria del microcontrolador, permitiendo al robot realizar una trayectoria distinta a las ya establecidas, como muestra el diagrama 34.

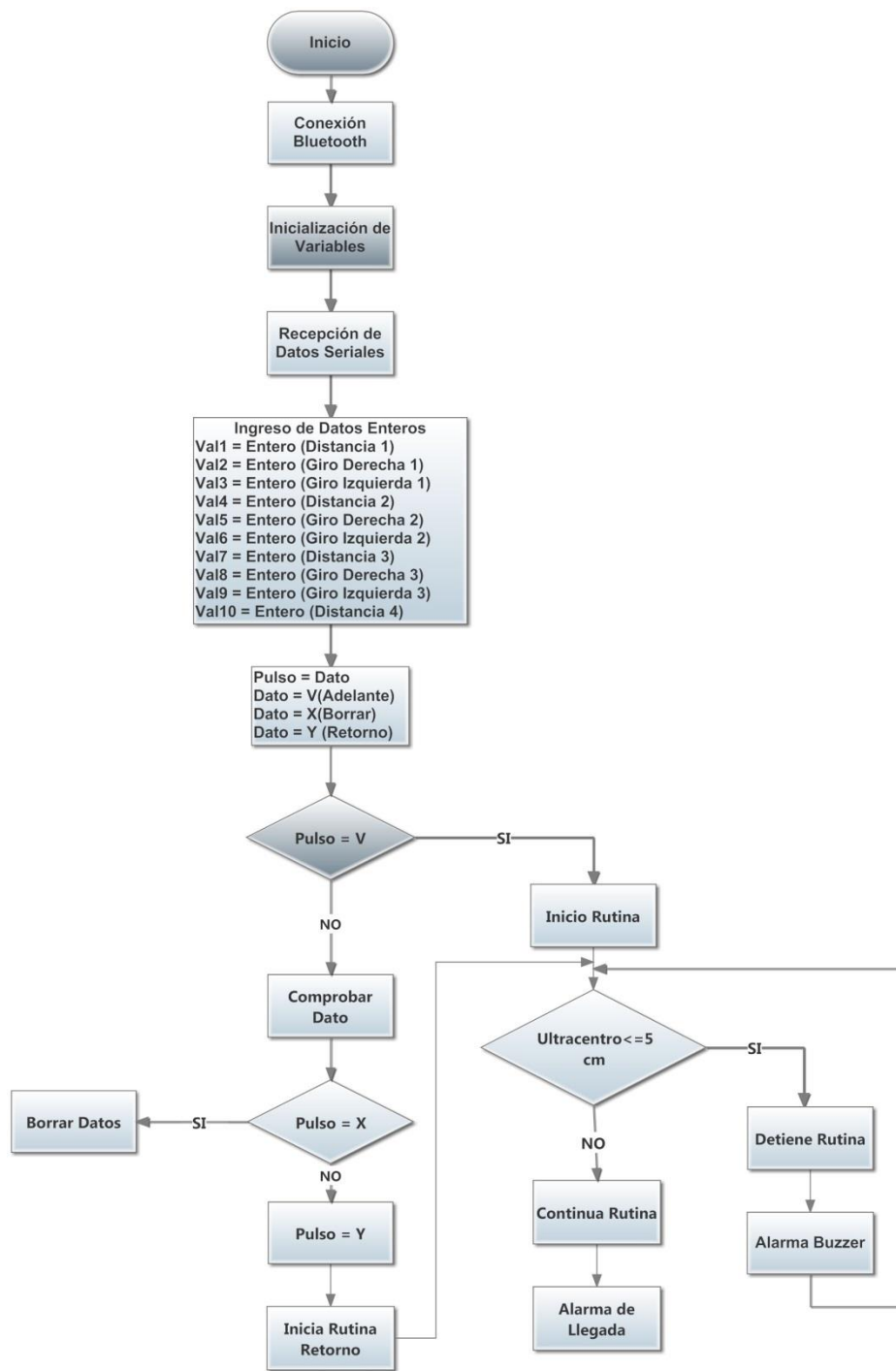


Figura 34. Diagrama de flujo trayectoria manual

Fuente: Los Autores

La figura 35 muestra el diagrama de control remoto del robot mensajero establecido en el microcontrolador, el cual ejecuta las funciones adelante, derecha, izquierda, atrás y detener.

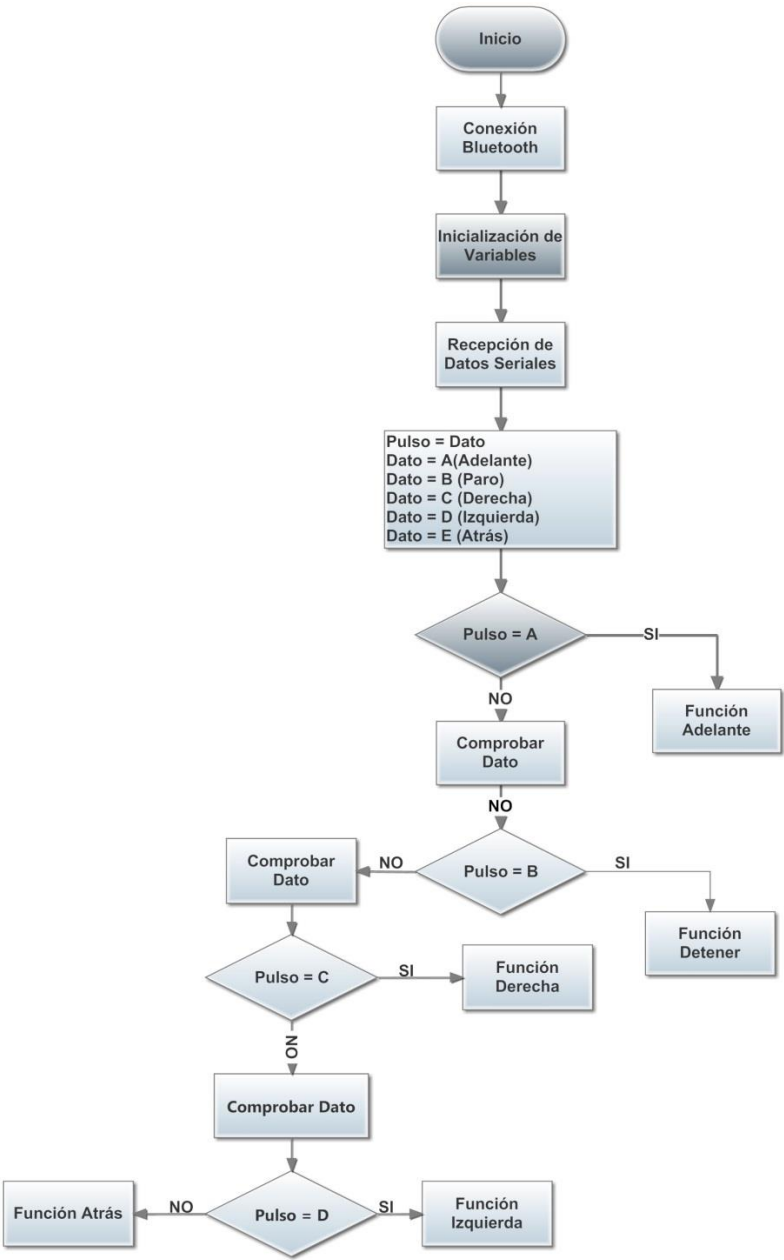


Figura 35. Diagrama de flujo control manual

Fuente: Los Autores

La figura 36 indica la estructura de la función evasor en el robot, la cual desempeña una comprobación de bandas cada 5 segundos, para poder evitar un obstáculo delante o al lado izquierdo o derecho escogiendo la mejor opción de salida para el dispositivo.

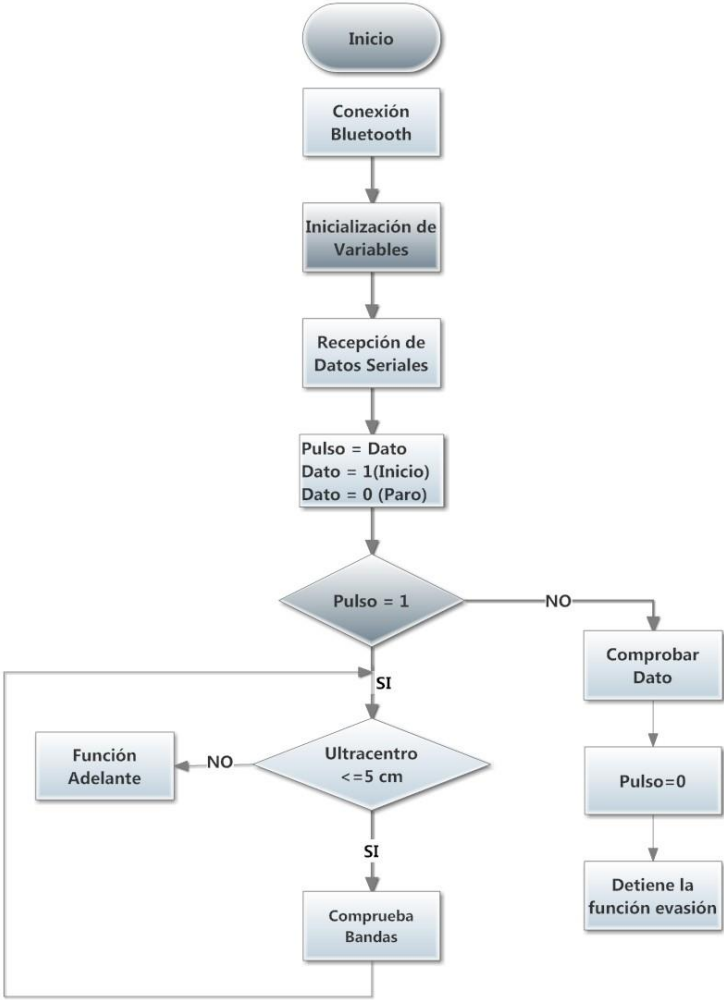


Figura 36. Diagrama de flujo evasor

Fuente: Los Autores

2.4.7 Diseño de aplicación

Para consolidar la programación establecida en el robot es necesario del desarrollo de una interfaz, que permita al usuario controlar el dispositivo móvil y

desempeñar correctamente cada una de sus funciones, mediante una plataforma o software libre de desarrollo como es Android.

La interfaz de usuario fue creada en este sistema operativo móvil, utilizando la plataforma web App Inventor que es de fácil acceso, además de ser una herramienta gráfica e intuitiva, cuya programación se realiza mediante bloques, es compatible con sistemas operativos Mac OS X, GNU/Linux o Windows y las aplicaciones creadas en esta plataforma funcionan en cualquier teléfono o dispositivo que tenga Android.

Los requisitos para poder utilizar App Inventor son los siguientes:

- Tener un computador con uno de los siguientes navegadores web: Mozilla Firefox 3.6 o superior, Apple Safari 5.0 o superior, Google Chrome 4.0 o superior, Microsoft Internet Explorer 7 o superior
- Disponer de una cuenta de correo electrónico en Gmail.
- Tener actualizado el JRE (Java RuntimeEnvironment) en el computador sino descargarlo e instalarlo desde: <http://java.com/es/download/index.jsp>
- Para realizar el diseño de la aplicación, se ingresa a la plataforma mediante el siguiente link: <http://beta.appinventor.mit.edu/>, seguir los pasos indicados hasta que observen la pantalla inicial del proyecto como muestra la figura 37.

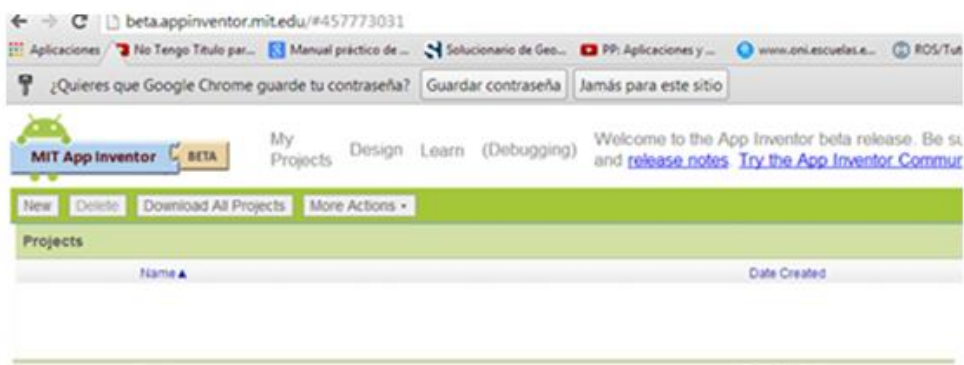


Figura 37. Pantalla inicial del proyecto

Fuente: Los Autores

La aplicación está construida en el entorno de diseño Design donde selecciona los múltiples elementos que forman parte de la interfaz, como muestra la figura 38, la cual está compuesta de 5 partes principales descritas brevemente a continuación:

- La paleta tiene los elementos que agrega elementos a la aplicación, tales como: imágenes, cuadros de texto, botones, animaciones, multimedia, sensores,...etc.
- Visor indica el aspecto visual que tendrá la aplicación en el móvil.
- Componentes muestra la lista de todos los elementos utilizados.
- Media muestra las todas las imágenes y sonidos agregados en el proyecto.
- Propiedades indicara las características del elemento seleccionado en el visor.

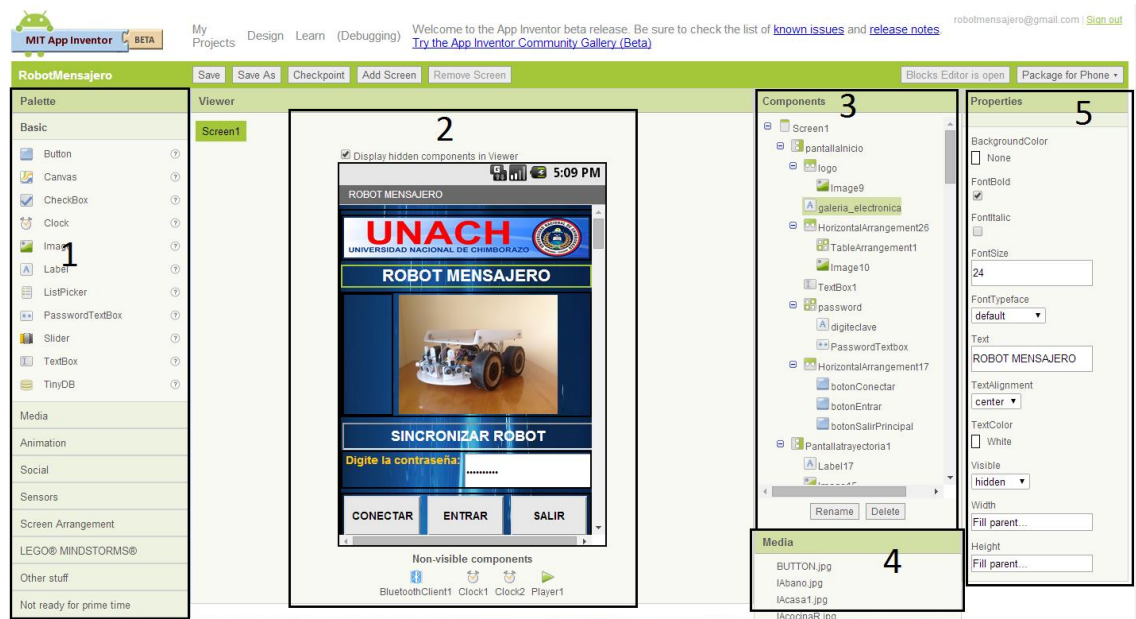


Figura 38. Entorno de Diseño

Fuente: Los Autores

En la aplicación se creó una pantalla principal o screen y nueve sub-pantallas para su correcta visualización y manejo, nombradas según la función que van a realizar sin espacio entre los nombres, por ejemplo pantallaInicio, pantallaTrayectoria1,

pantallaControl, etc. en las cuales ingresa cuadros de texto, imágenes, botones, audio, cuadro de contraseña, cuadros de ingreso de datos, reloj y bluetooth.

Con el editor de bloques se ha programado las funciones que van a ejecutarse en la aplicación, mediante la utilización de los bloques de definición, texto, lógicos, control, texto y color.

2.4.7.1 Programación en la plataforma Android

La aplicación requiere de los parámetros y datos establecidos en el microcontrolador del robot, para de esta manera crear una App que maneje correctamente al mensajero, todo esto desarrollado en una plataforma para programar, como es APP Inventor, la cual está indicada en el anexo 3.

Por seguridad y uso exclusivo del personal administrativo, el ingreso a la aplicación, requiere insertar una clave de seguridad como indica el diagrama 39.

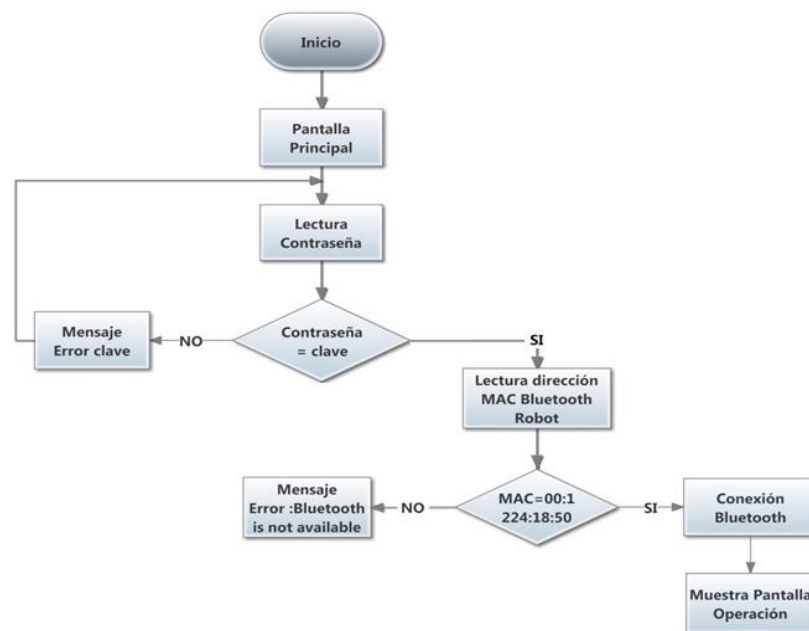


Figura 39. Diagrama de flujo Ingreso Clave

Fuente: Los Autores

El diagrama principal de operación del robot, trabaja a nivel de aplicación y requiere la selección de una subpantalla en la pantalla de principal, para que ingrese a la función requerida, como indica el diagrama de flujo en la figura 40.

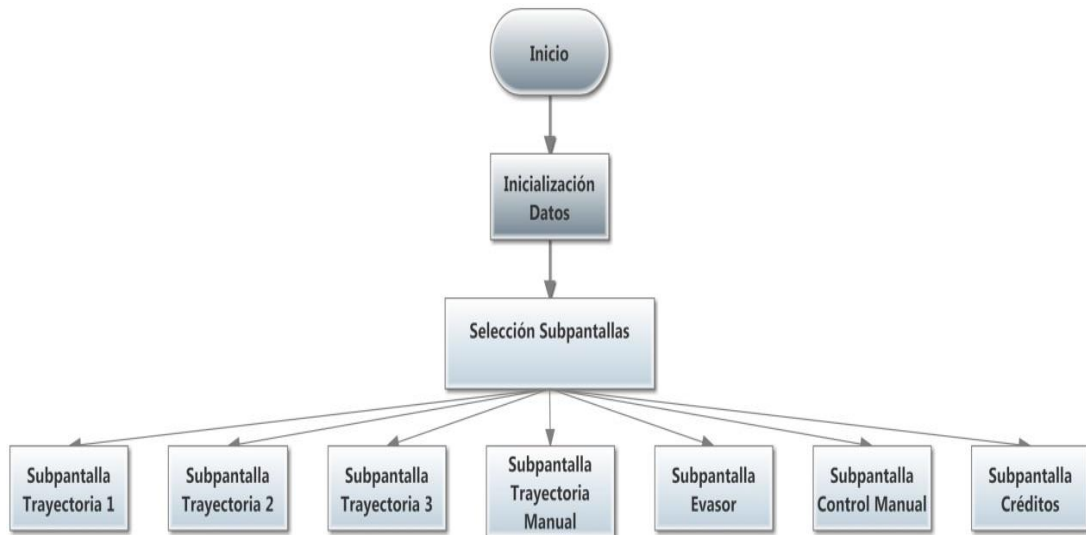


Figura 40. Diagrama de flujo operación del robot

Fuente: Los Autores

El menú principal de operación comprende a su vez de seis subrutinas las cuales realizan un trabajo independiente una de otra de acuerdo al requerimiento solicitado por el usuario. Estos trayectos establecidos, pueden ser utilizados mediante la selección correcta de la subrutina, las cuales están pre cargadas en la memoria del microcontrolador.

A continuación se observa el diagrama de flujo utilizado para realizar la programación para el trayecto 1, 2 y 3, en la cual primero se verifica el estado del robot que no esté ejecutando una subrutina, luego se envía la orden hacia el microcontrolador para que ejecute el recorrido desde su punto de inicio hasta su punto de llegada y viceversa, como muestra la figura 41, 42 y 43.

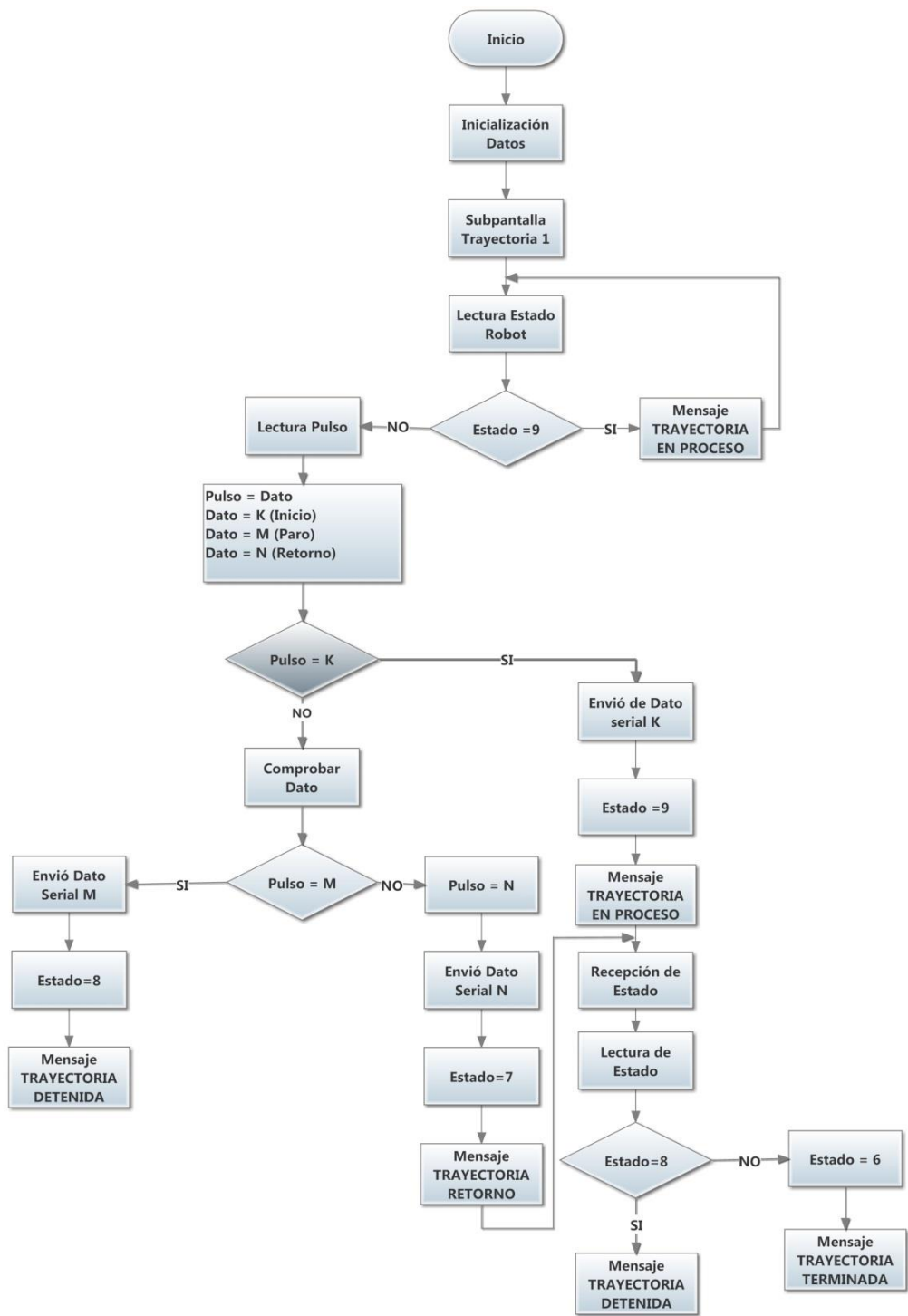


Figura 41. Diagrama de flujo subrutina trayectoria 1

Fuente: Los Autores

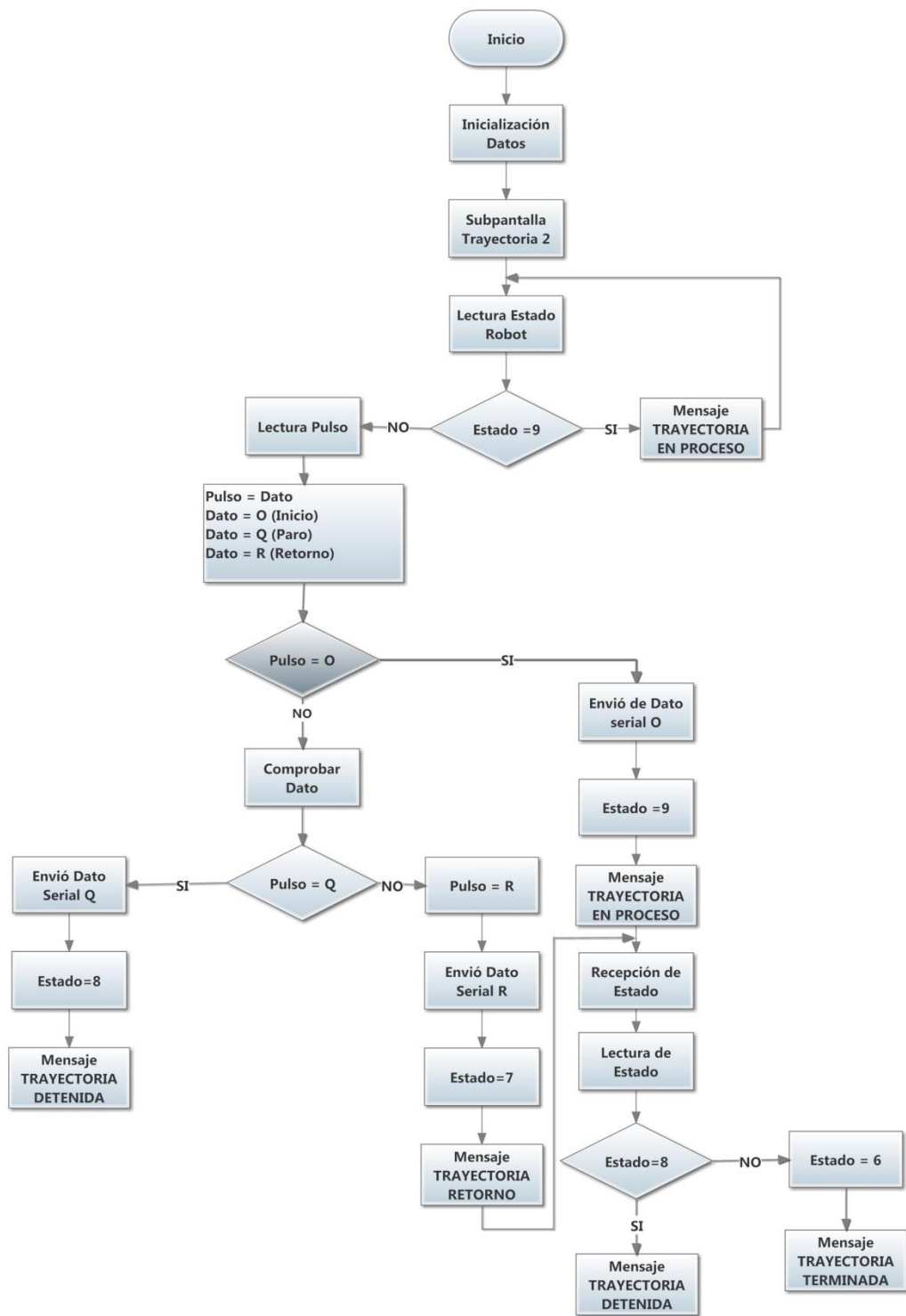


Figura 42. Diagrama de flujo subrutina trayectoria 2

Fuente: Los Autores

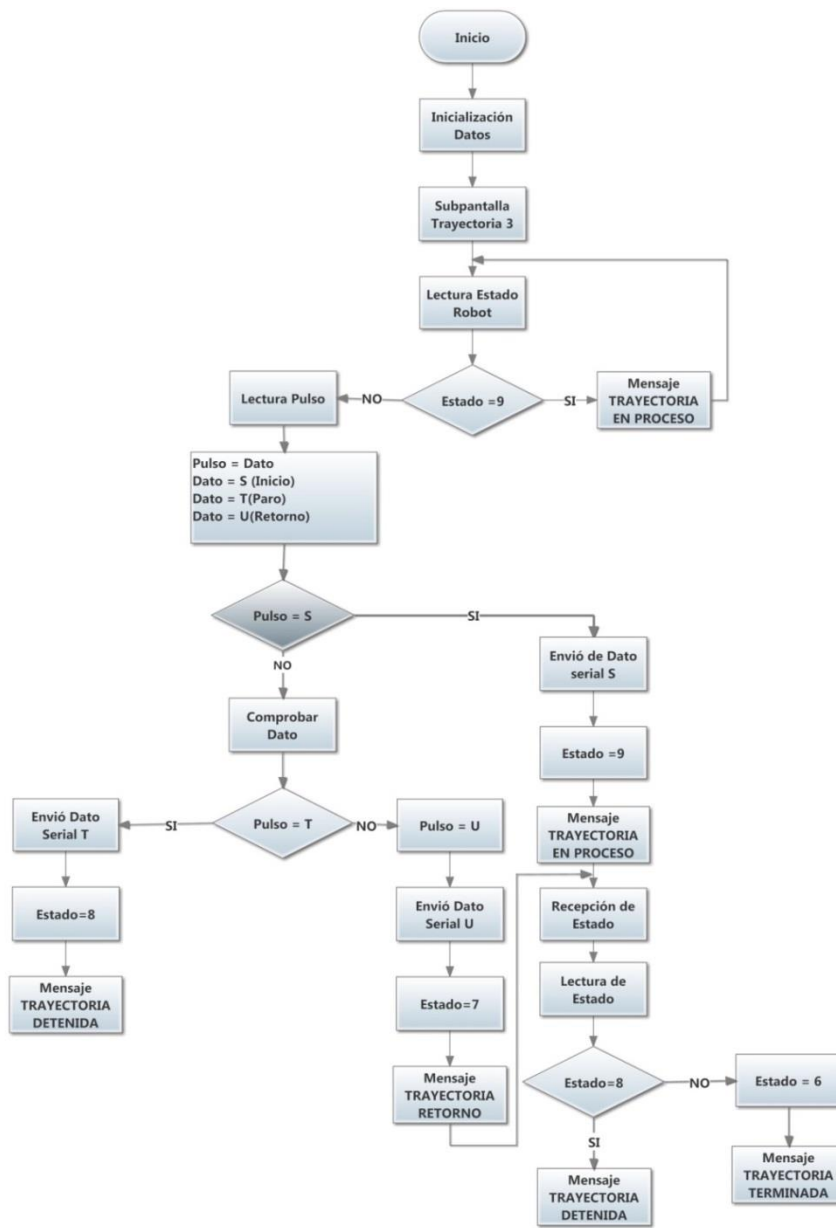


Figura 43. Diagrama de flujo subrutina trayectoria 3

Fuente: Los Autores

Posteriormente como indica el diagrama de la figura 44, muestra el proceso de adquisición de datos para la realización de una posible trayectoria, indistinta a las ya establecidas en el microcontrolador.

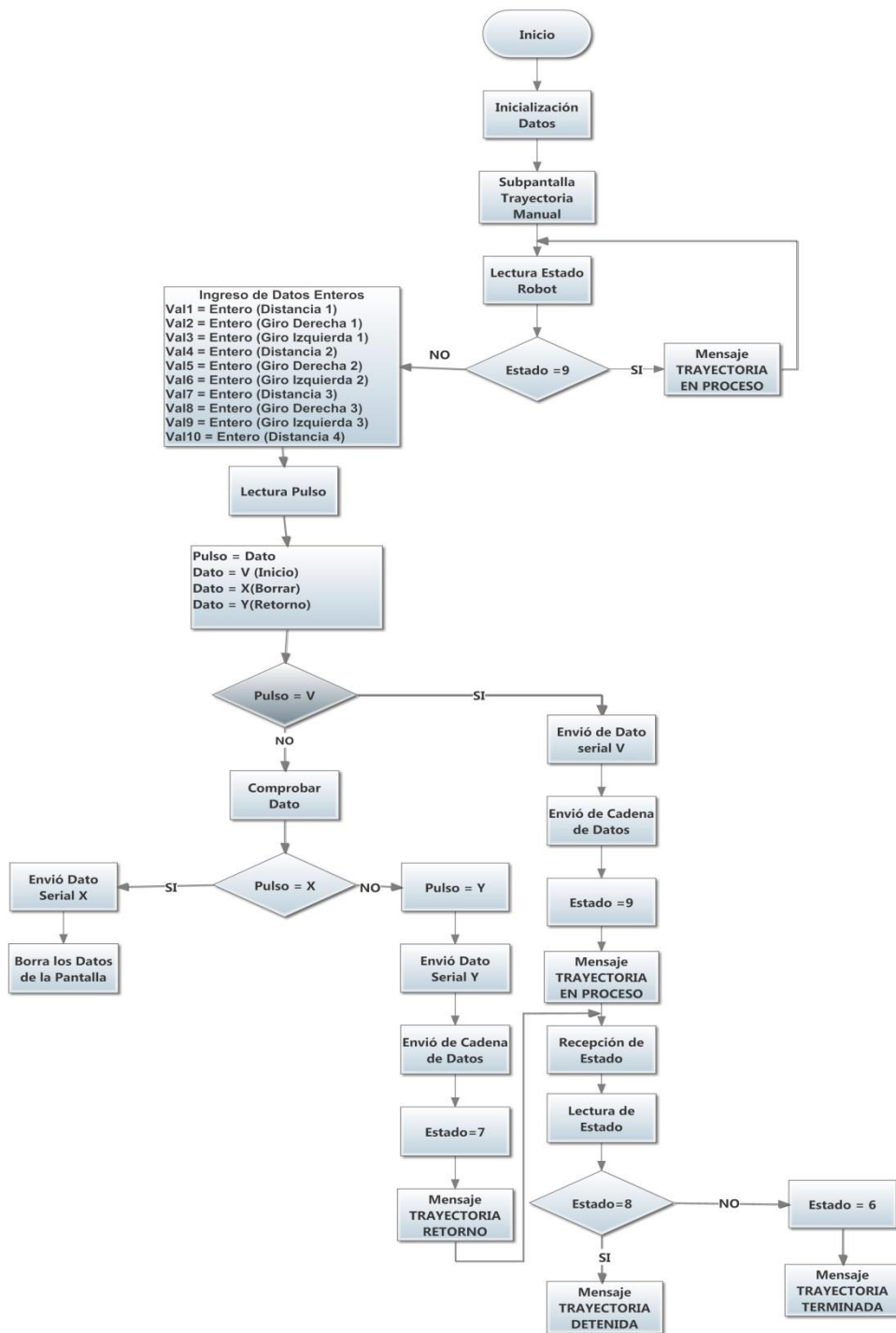


Figura 44. Diagrama de flujo subrutina trayectoria manual

Fuente: Los Autores

La figura 45 muestra el diagrama de la estructura base, con la cual el robot puede ser controlado por el usuario en forma remota dándole un trayecto al azar.

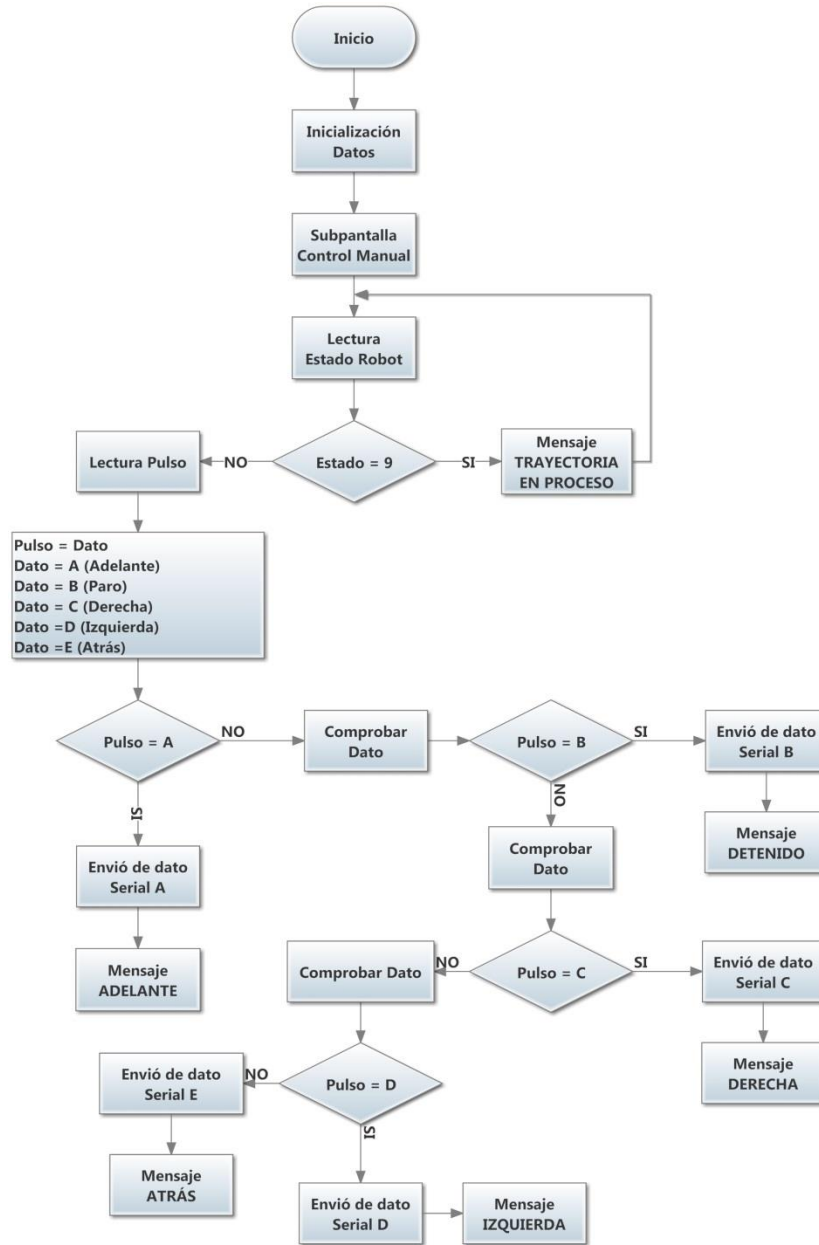


Figura 45. Diagrama de flujo subrutina control manual

Fuente: Los Autores

La figura 46 indica la estructura de la función evasor en el robot, la cual desempeña una comprobación de bandas cada 5 segundos para poder evitar un obstáculo delante del robot.

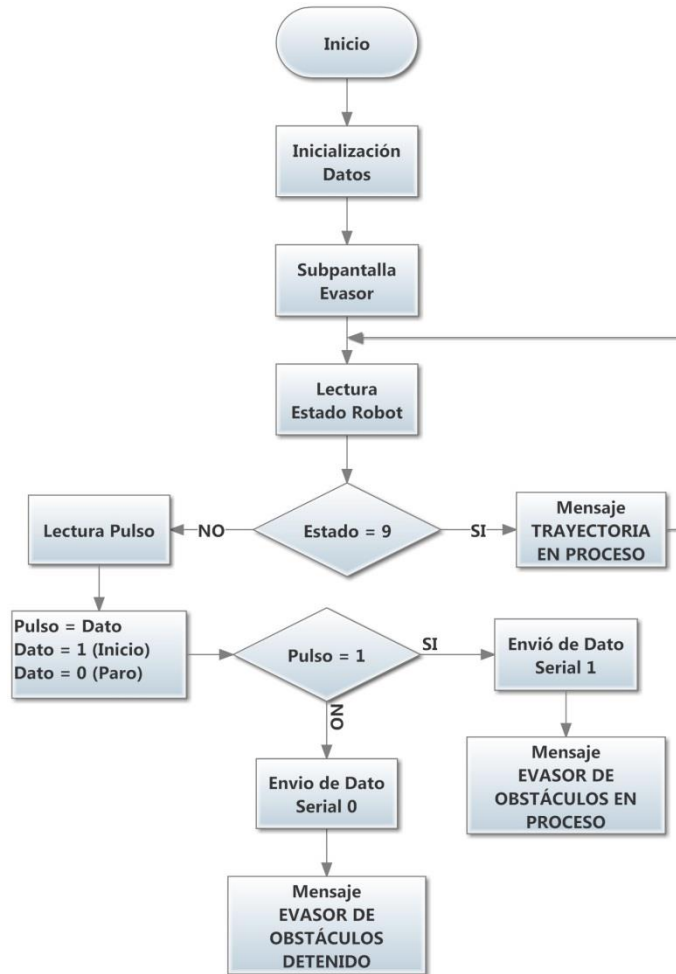


Figura 46. Diagrama de flujo subrutina evasor

Fuente: Los Autores

2.4.7.2 Visualización de la aplicación

La aplicación esta presenta en un archivo de extensión apk, la cual es ejecutable en dispositivos con el sistema operativo Android.

El dispositivo móvil indica 9 pantallas, siendo la primera la pantalla de entrada a la aplicación, tal como muestra la figura 47.



Figura 47. Pantalla de Inicio

Fuente: Los Autores

Después de la pantalla de inicio, está la pantalla de control o modo de operación, la cual da paso a las siete sub pantallas como muestra la figura 48, entre las cuales están:

- Pantalla de trayectoria 1
- Pantalla de trayectoria 2
- Pantalla de trayectoria 3
- Pantalla de trayectoria manual
- Pantalla de control manual
- Pantalla evasor de obstáculos
- Pantalla de créditos

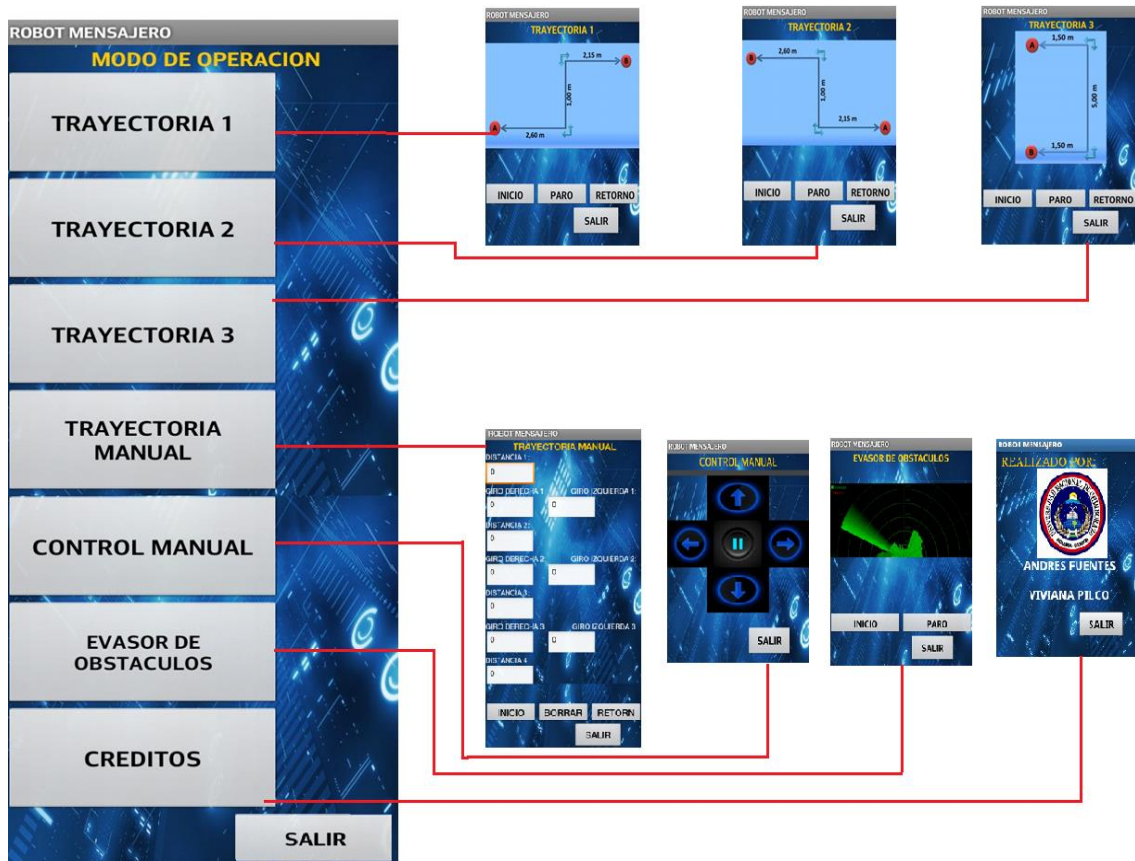


Figura 48. Pantalla de Control

Fuente: Los Autores

En la pantalla de control puede escogerse las diversas funcionalidades establecidas para el manejo del robot mensajero, al interactuar mediante los botones de inicio, paro, retorno y salir; el cual es el que regresa inmediatamente a la pantalla de control.

2.5 PROCESAMIENTO Y ANÁLISIS

Todo el proceso seguido para la elaboración y culminación del robot mensajero está establecido en el diagrama 49.

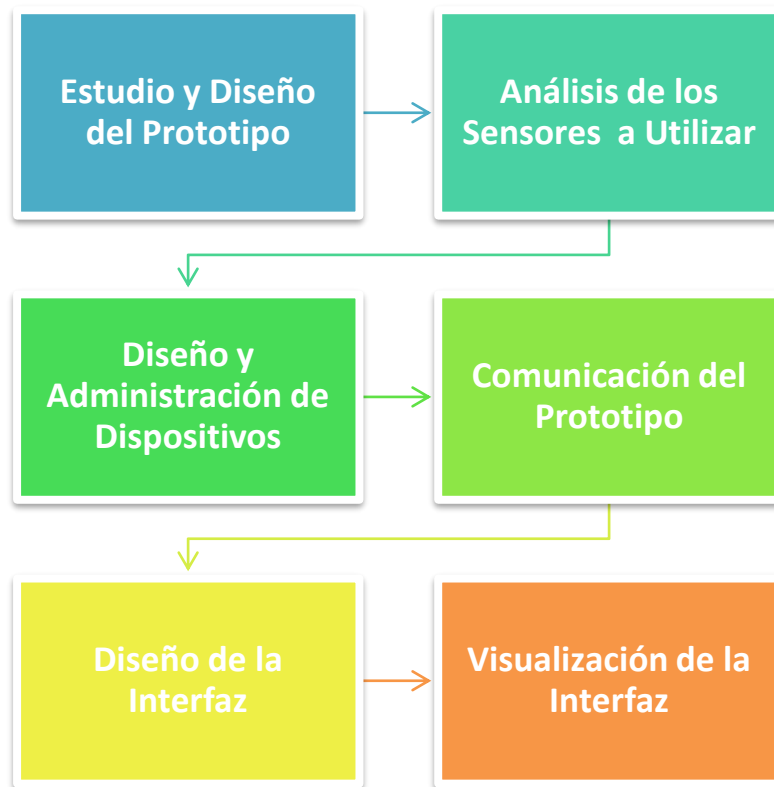


Figura 49. Diagrama de procedimientos

Fuente: Los Autores

2.5.1 Estudio y diseño del prototipo

En esta etapa fue recolectada la información sobre robótica y robots autónomos, además de prototipos diseñados con fines similares siendo el mensajero prototipo diseñado con cada uno de los modelos evaluados en esta etapa y con los requerimientos necesarios de acuerdo al tema de investigación.

2.5.2 Análisis de los sensores a utilizar

Se identifica los sensores a utilizar en el robot mensajero, como los ultrasonidos y fototransistor.

2.5.3 Diseño y administración de dispositivos

Este paso permite la ubicación de los dispositivos para un óptimo funcionamiento, poniendo lo más precisamente posible los dispositivos en la estructura del robot, para cada función establecida.

2.5.4 Comunicación del prototipo

Es administrado mediante un dispositivo y protocolo Bluetooth el cual permitiría transmitir y recibir datos de la manera más eficiente y segura posible, siendo el mismo trabajado en forma serial.

2.5.5 Diseño de la interfaz para controlar al robot

Es utilizada una interfaz creada en una plataforma libre, para desarrollar una aplicación que permitiera al usuario el control del robot mensajero de una manera sencilla y con un dispositivo celular a través de una comunicación mediante bluetooth.

2.5.6 Visualización de la interfaz

Finalmente se observara la aplicación que consta de una pantalla principal y 9 subpantallas, que enviara las órdenes mediante la comunicación bluetooth, para poder manejar al robot a distancia mediante un celular.

2.6 COMPROBACIÓN DE LA HIPÓTESIS

En la comprobación de la hipótesis del presente tema de tesis se realizó el análisis mediante la distribución estándar normal de una población infinita o desconocida del área administrativa de la Facultad de Ingeniería.

A continuación los procedimientos seguidos paso a paso para realizar la comprobación de la hipótesis, mediante la prueba de aseveraciones acerca de una media poblacional o desconocida.

Comprobación del sensor de distancia (Encoder)

a. Planteamiento de la Hipótesis

H_0 : El promedio de las medidas de los valores obtenidos por el sensor de distancia (encoder) en las tres trayectorias μ_T es igual al promedio de los tres trayectos establecidos previamente en el robot μ_{Real} .

H_1 : El promedio de las medidas de los valores obtenidos por el sensor de distancia (encoder) en las tres trayectorias μ_T no es igual al promedio de los tres trayectos establecidos previamente en el robot μ_{Real} .

$$H_0: \mu_T = \mu_{Real} \quad (9)$$

$$H_1: \mu_T \neq \mu_{Real} \quad (10)$$

b. Establecimiento del nivel de significancia:

Se estableció $\alpha = 0,05$ que está relacionado a un 95% de confianza

c. Encontrar la Muestra

Mediante el cálculo muestral realizado con la ecuación 1 en la pag.21 se consideró tomar 35 valores al azar de los 40 obtenidos de las diversas pruebas en los tres trayectos, mediante la página http://nsetup.org/php_on_line/numero_aleatorio, los mismos que establecieron un valor promedio en relación a la distancia real.

d. Calculo de la Media

Tabla 6. Datos Adquiridos de los tres Trayectos

Trayectoria 1	1	Prueba 1	2,58	0,211	1,03	0,212	2,14	6,173
	2	Prueba 2	2,62	0,212	1,02	0,209	2,12	6,181
	3	Prueba 3	2,61	0,209	1,01	0,212	2,14	6,181
	4	Prueba 4	2,57	0,212	0,98	0,21	2,15	6,122
	5	Prueba 5	2,59	0,21	0,99	0,212	2,13	6,132
	6	Prueba 6	2,63	0,213	1	0,21	2,16	6,213

	7	Prueba 7	2,59	0,211	1,01	0,213	2,13	6,154
	8	Prueba 8	2,61	0,208	0,98	0,211	2,17	6,179
	9	Prueba 9	2,61	0,209	0,99	0,208	2,16	6,177
	10	Prueba 10	2,58	0,209	1	0,209	2,15	6,148
	11	Prueba 11	2,59	0,212	0,99	0,209	2,12	6,121
	12	Prueba 12	2,61	0,21	1,01	0,212	2,14	6,182
	13	Prueba 13	2,6	0,213	1,02	0,21	2,15	6,193
	14	Prueba 14	2,59	0,211	0,98	0,213	2,13	6,124
	15	Prueba 15	2,6	0,21	0,99	0,211	2,16	6,171
	16	Prueba 16	2,61	0,213	1	0,21	2,13	6,163
	17	Prueba 17	2,57	0,211	0,99	0,209	2,16	6,14
	18	Prueba 18	2,59	0,208	1,02	0,212	2,13	6,16
	19	Prueba 19	2,63	0,209	1,01	0,21	2,17	6,229
	20	Prueba 20	2,59	0,209	0,98	0,212	2,16	6,151
Trayectoria 2	21	Prueba 1	2,58	0,211	3,48	0,208	2,14	8,619
	22	Prueba 2	2,61	0,208	3,51	0,209	2,12	8,657
	23	Prueba 3	2,57	0,21	3,49	0,212	2,14	8,622
	24	Prueba 4	2,59	0,213	3,5	0,21	2,15	8,663
	25	Prueba 5	2,63	0,21	3,49	0,212	2,12	8,662
	26	Prueba 6	2,63	0,209	3,49	0,21	2,14	8,679
	27	Prueba 7	2,61	0,212	3,51	0,213	2,15	8,695
	28	Prueba 8	2,58	0,21	3,5	0,211	2,13	8,631
	29	Prueba 9	2,61	0,213	3,52	0,208	2,14	8,691
	30	Prueba 10	2,6	0,211	3,51	0,209	2,15	8,68
	31	Prueba 11	2,59	0,208	3,49	0,212	2,13	8,63
	32	Prueba 12	2,61	0,21	3,5	0,21	2,14	8,67
	33	Prueba 13	2,6	0,213	3,49	0,213	2,15	8,666
	34	Prueba 14	2,59	0,211	3,49	0,211	2,13	8,632
	35	Prueba 15	2,6	0,21	3,5	0,208	2,16	8,678
Trayectoria 3	36	Prueba 1	1,48	0,21	5,11	0,208	1,53	8,538
	37	Prueba 2	1,52	0,213	4,9	0,209	1,52	8,362
	38	Prueba 3	1,51	0,21	5,08	0,212	1,47	8,482
	39	Prueba 4	1,47	0,209	4,91	0,21	1,46	8,259
	40	Prueba 5	1,46	0,212	5,09	0,213	1,52	8,495

Fuente: Los Autores

En la tabla 6, se observan los datos recogidos de las tres trayectorias y en la tabla 7 indica el valor promedio real de los tres trayectos.

Tabla 7. Datos Óptimos de Recorrido de los tres Trayectos

Trayectoria 1	1	Real	2,6	0,21	1	0,21	2,15	6,17
Trayectoria 2	2	Real	2,6	0,21	3,5	0,21	2,15	8,67
Trayectoria 3	3	Real	1,5	0,21	5	0,21	1,5	8,42

Promedio	7,75333
-----------------	----------------

Fuente: Los Autores

En la tabla 8, se observa el cálculo de la media y la varianza de cada una de las pruebas realizadas en las tres trayectorias. Estos cálculos fueron realizados en Excel utilizando las funciones estadísticas.

Tabla 8. Media de Muestras Aleatorias

			Valor	Media (\bar{X})	$(V - \bar{X})^2$
Muestras	1	Prueba 1	6,173	7,341	1,364224
	2	Prueba 2	6,181	7,341	1,3456
	3	Prueba 4	6,122	7,341	1,485961
	4	Prueba 5	6,132	7,341	1,461681
	5	Prueba 6	6,213	7,341	1,272384
	6	Prueba 7	6,154	7,341	1,408969
	7	Prueba 9	6,177	7,341	1,354896
	8	Prueba 10	6,148	7,341	1,423249
	9	Prueba 11	6,121	7,341	1,4884
	10	Prueba 12	6,182	7,341	1,343281
	11	Prueba 13	6,193	7,341	1,317904
	12	Prueba 14	6,124	7,341	1,481089
	13	Prueba 15	6,171	7,341	1,3689
	14	Prueba 16	6,163	7,341	1,387684

15	Prueba 17	6,14	7,341	1,442401
16	Prueba 18	6,16	7,341	1,394761
17	Prueba 19	6,229	7,341	1,236544
18	Prueba 20	6,151	7,341	1,4161
19	Prueba 1	8,619	7,341	1,633284
20	Prueba 2	8,657	7,341	1,731856
21	Prueba 3	8,622	7,341	1,640961
22	Prueba 5	8,662	7,341	1,745041
23	Prueba 6	8,679	7,341	1,790244
24	Prueba 7	8,695	7,341	1,833316
25	Prueba 8	8,631	7,341	1,6641
26	Prueba 9	8,691	7,341	1,8225
27	Prueba 11	8,63	7,341	1,661521
28	Prueba 12	8,67	7,341	1,766241
29	Prueba 14	8,632	7,341	1,666681
30	Prueba 15	8,678	7,341	1,787569
31	Prueba 1	8,538	7,341	1,432809
32	Prueba 2	8,362	7,341	1,042441
33	Prueba 3	8,482	7,341	1,301881
34	Prueba 4	8,259	7,341	0,842724
35	Prueba 5	8,495	7,341	1,331716

$\sigma = \frac{\sum(V - \bar{X})^2}{n - 1}$	1,52026215
--	------------

Fuente: Los Autores

e. Análisis de las regiones críticas

Dado el nivel de significancia $\alpha = 0.05$ se determinó trabajar con dos regiones críticas, dado que la hipótesis nula deberá estar en el nivel de aceptación entre -1.96 y 1.96 debido al nivel de confianza del 95% como se observa en la figura 50.

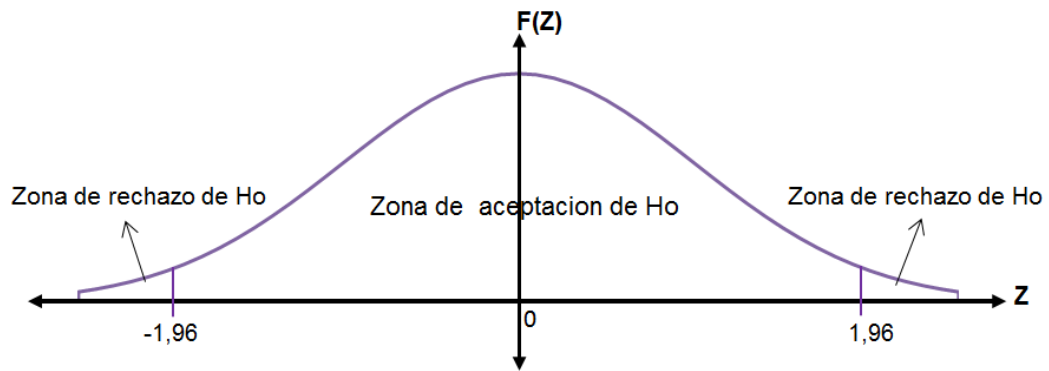


Figura 50. Gráfica regiones críticas

Fuente: Los Autores

f. Cálculo Z_c o Valor Crítico

Para contemplar la hipótesis nula (H_0) de la fórmula (4), el valor $Z_{calculado}$ debe encontrarse en su zona de aceptación, es así que se planteó la siguiente fórmula para calcular Z_c en función de los siguientes parámetros:

\bar{X} : Media de valores aleatorios

μ : Promedio de los valores optimos reales

σ : Variación estándar

n : Número de muestras

Considerando dichos valores fueron remplazados en la fórmula de la obtención del valor crítico.

$$Z_c = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

$$Z_c = \frac{7.34 - 7.75}{\frac{1.52}{\sqrt{35}}}$$

$$Z_c = \frac{-0.41}{0.256}$$

$$Z_c = -1.595 \approx -1.6$$

g. Establecimiento de la Regla de Decisión

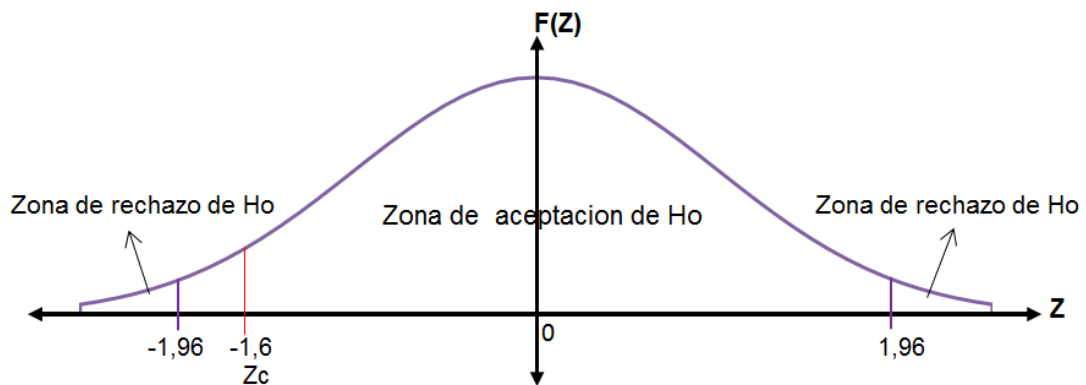


Figura 51. Establecimiento del punto crítico

Fuente: Los Autores

Luego de obtener el Z_c como muestra la figura 51, indica que el valor está dentro de la zona de aceptación de H_0 es decir $-1,96 < Z_c < 1,96$, con un nivel de significancia del 5%, probando así que la hipótesis nula es aceptada, dado que el promedio de distancias de los trayectos establecidos son semejantes a los promedios obtenidos de los resultados, concluyendo a su vez un alto grado de eficacia al desarrollar un trayecto por parte del robot mensajero.

CAPÍTULO III

3. RESULTADOS

Con el desarrollo del robot para servicio de mensajería interna se analizó y creó dos prototipos, mediante los cuales evaluando su eficiencia al realizar un trayecto, además de su desenvolvimiento en un entorno previamente establecido, conllevó a determinar el mejor prototipo a elegir mediante varias pruebas y ajustes realizados a las falencias encontradas en el dispositivo como (sensores, distancia, entornos, adherencia al suelo, etc.).

Dando así un análisis previo del desempeño del robot mensajero y su implementación en el área a utilizarse, al tener un correcto funcionamiento de sus múltiples funciones tanto en el robot, como en el dispositivo móvil.

A continuación, las diversas pruebas realizadas a los trayectos establecidos en el robot mensajero..

3.1 Pruebas en los Trayectos

En cuanto a la realización óptima de un trayecto, se evaluó las tres trayectorias establecidas, además de los parámetros necesarios a cumplir para determinar el margen de error entre un punto y otro, tomando en consideración que una trayectoria trazada en el prototipo debiera cumplir con la distancia establecida en un trayecto dentro del entorno, en este caso el área administrativa de la Facultad de Ingeniería como indica la figura 52.

Cabe mencionar que toda la trayectoria estará establecida en función de distancia incluyendo los giros, dado que al utilizar el encoder como referencia se trabajaría con el cálculo de pulsos igual a distancia como indica la ecuación 1 en las pruebas del sensor de distancia.

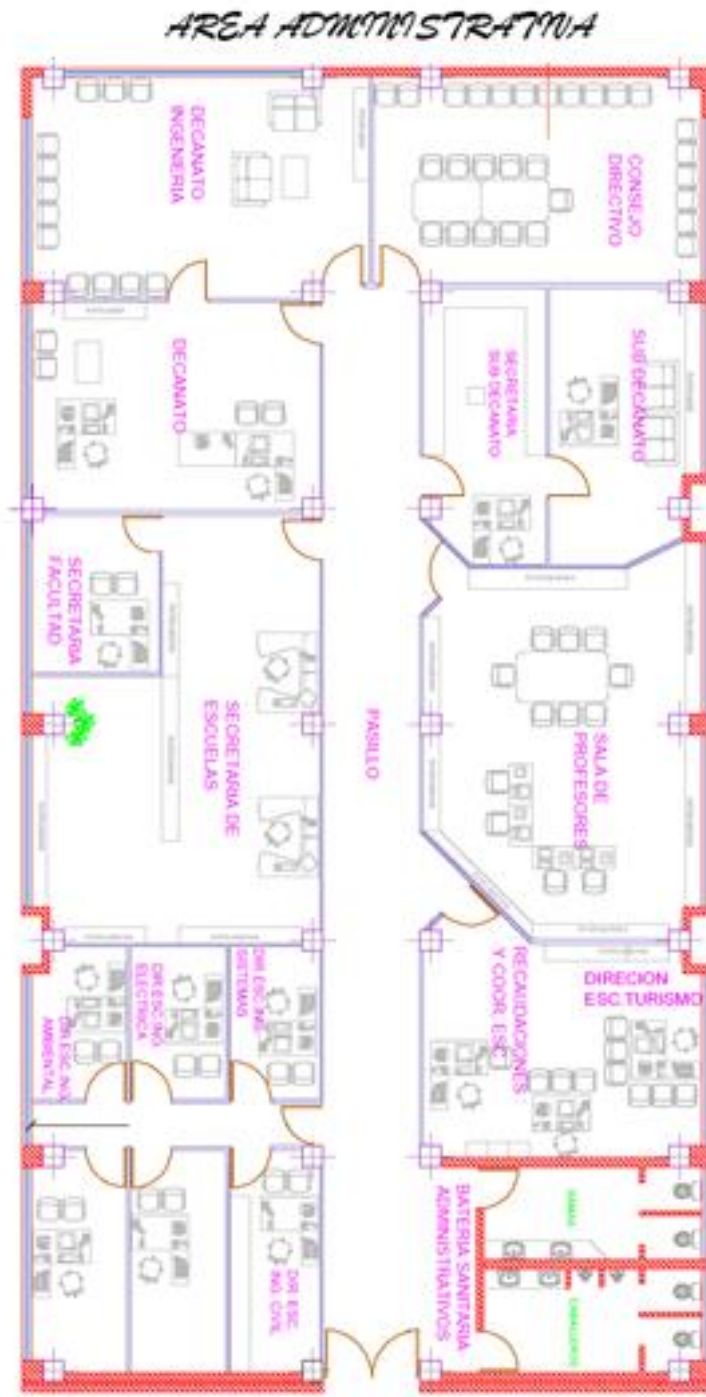


Figura 52. Área Administrativa

Fuente: Los Autores

3.1.1 Trayectoria 1

En cuanto al trayecto 1, evaluando el dispositivo 20 veces obteniendo un error promedio acumulable en trayecto realizado, dado que la superficie del suelo era deslizable y las llantas no se adherían perfectamente al suelo debido al material utilizado, siendo este un error de tipo mecánico y no electrónico, debiendo así compensarlo posteriormente por intervención del usuario o a su vez utilizando el mando remoto para establecerlo en el lugar requerido para realizar la trayectoria.

A continuación los valores obtenidos de los trayectos realizados y su curva de error como indica la figura 53.

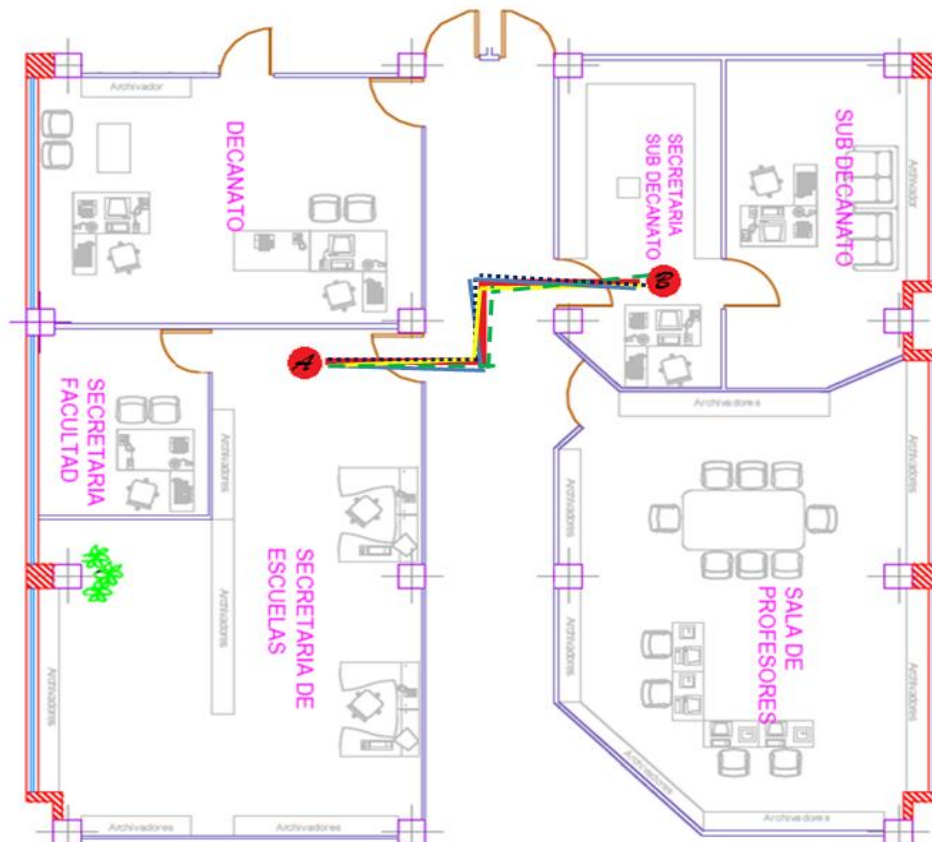


Figura 53. Trayectoria 1

Fuente: Los Autores

La tabla 9 muestra, los datos prácticos recogidos al realizar la trayectoria 1 en cada tramo.

Tabla 9. Pruebas de Trayectos

Trayectoria 1						
	Distancia 1 (m)	Giro1 (m)	Distancia 2 (m)	Giro 2 (m)	Distancia 3 (m)	Total
		90° = 0,21 m		90° = 0,21 m		
Real	2,6	0,21	1	0,21	2,15	6,17
Prueba 1	2,58	0,211	1,03	0,212	2,14	6,173
Prueba 2	2,62	0,212	1,02	0,209	2,12	6,181
Prueba 3	2,61	0,209	1,01	0,212	2,14	6,181
Prueba 4	2,57	0,212	0,98	0,21	2,15	6,122
Prueba 5	2,59	0,21	0,99	0,212	2,13	6,132
Prueba 6	2,63	0,213	1	0,21	2,16	6,213
Prueba 7	2,59	0,211	1,01	0,213	2,13	6,154
Prueba 8	2,61	0,208	0,98	0,211	2,17	6,179
Prueba 9	2,61	0,209	0,99	0,208	2,16	6,177
Prueba 10	2,58	0,209	1	0,209	2,15	6,148
Prueba 11	2,59	0,212	0,99	0,209	2,12	6,121
Prueba 12	2,61	0,21	1,01	0,212	2,14	6,182
Prueba 13	2,6	0,213	1,02	0,21	2,15	6,193
Prueba 14	2,59	0,211	0,98	0,213	2,13	6,124
Prueba 15	2,6	0,21	0,99	0,211	2,16	6,171
Prueba 16	2,61	0,213	1	0,21	2,13	6,163
Prueba 17	2,57	0,211	0,99	0,209	2,16	6,14
Prueba 18	2,59	0,208	1,02	0,212	2,13	6,16
Prueba 19	2,63	0,209	1,01	0,21	2,17	6,229
Prueba 20	2,59	0,209	0,98	0,212	2,16	6,151

Fuente: Los Autores

En la siguiente tabla 10, está estipulado el error del prototipo en el trayecto real.

Tabla 10. Errores en distancia de la Trayectoria 1

Error en Comparación con la Medida Real en la Trayectoria 1						
	Distancia 1 (m)	Giro1 (m)	Distancia 2 (m)	Giro 2 (m)	Distancia 3 (m)	Total (m)
Prueba 1	0,02	0,001	0,03	0,002	0,01	0,063
Prueba 2	0,02	0,002	0,02	0,001	0,03	0,073
Prueba 3	0,01	0,001	0,01	0,002	0,01	0,033
Prueba 4	0,03	0,002	0,02	0	0	0,052
Prueba 5	0,01	0	0,01	0,002	0,02	0,042
Prueba 6	0,03	0,003	0	0	0,01	0,043
Prueba 7	0,01	0,001	0,01	0,003	0,02	0,044
Prueba 8	0,01	0,002	0,02	0,001	0,02	0,053
Prueba 9	0,01	0,001	0,01	0,002	0,01	0,033
Prueba 10	0,02	0,001	0	0,001	0	0,022
Prueba 11	0,01	0,002	0,01	0,001	0,03	0,053
Prueba 12	0,01	0	0,01	0,002	0,01	0,032
Prueba 13	0	0,003	0,02	0	0	0,023
Prueba 14	0,01	0,001	0,02	0,003	0,02	0,054
Prueba 15	0	0	0,01	0,001	0,01	0,021
Prueba 16	0,01	0,003	0	0	0,02	0,033
Prueba 17	0,03	0,001	0,01	0,001	0,01	0,052
Prueba 18	0,01	0,002	0,02	0,002	0,02	0,054
Prueba 19	0,03	0,001	0,01	0	0,02	0,061

Fuente: Los Autores

En cuanto al error acumulado total como muestra la tabla 11 está establecida en 20 cm de distancia recorrida en cinco pruebas realizadas.

Tabla 11. Error Promedio Total

Promedio de error en Trayectoria 1		
	Distancia Total (m)	Error (m)
Prueba 1	6,171	0,001
Prueba 2	6,181	0,073
Prueba 3	6,181	0,033
Prueba 4	6,122	0,052
Prueba 5	6,132	0,042
Promedio	6,1574	0,0402
Error Total		0,201

Fuente: Los Autores

Posteriormente como muestra la tabla 12 y la figura 54, indican la trayectoria promedio en comparación a la real, dando así un análisis de los valores obtenidos en comparación a los establecidos en la programación previamente, y los cuales debían ser cumplidos a cabalidad.

Tabla 12. Trayectoria 1 Promedio

Promedio de las Pruebas realizadas en la Trayectoria 1						
	Distancia 1 (m)	Giro 1 (m)	Distancia 2 (m)	Giro 2 (m)	Distancia 3 (m)	Total (m)
Promedio	2,596	0,2108	1,002	0,2108	2,1378	6,1574
Real	2,6	0,21	1	0,21	2,15	6,17

Fuente: Los Autores

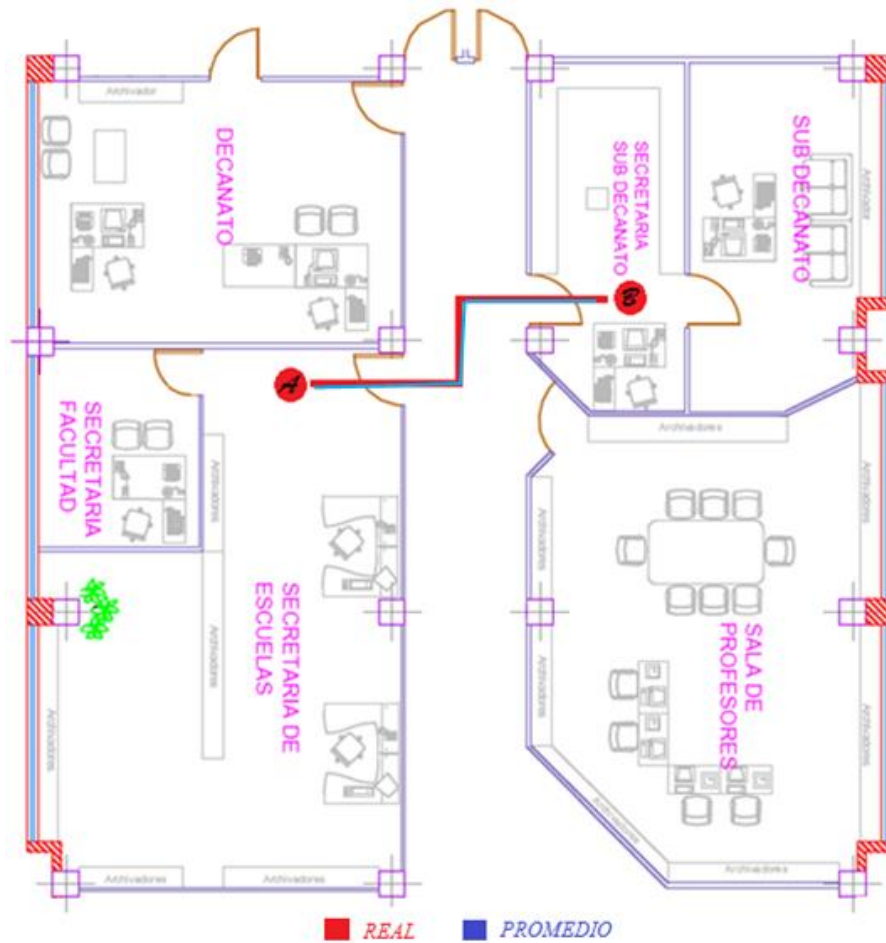


Figura 54. Trayectoria Promedio

Fuente: Los Autores

3.1.2 Trayectoria 2

En cuanto al trayecto 2, de la misma forma del trayecto 1 se evaluó el dispositivo 15 veces obteniendo un error promedio acumulable en trayecto realizado, con las mismas especificaciones del trayecto 1.

A continuación los valores obtenidos de los trayectos realizados y su curva de error como indica la figura 55.

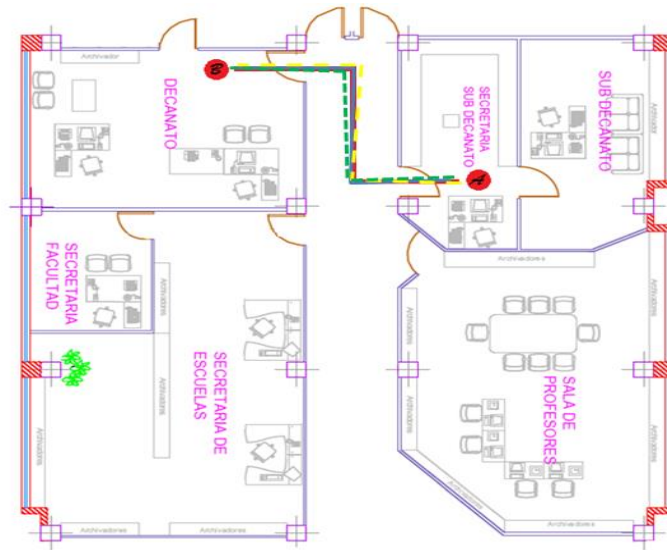


Figura 55. Trayectoria 2

Fuente: Los Autores

La tabla 13 muestra, los datos prácticos recogidos al realizar la trayectoria 2 en cada tramo.

Tabla 13. Pruebas de Trayectos

Trayectoria 2						
	Distancia 1 (m)	Giro1 (m) 90° = 0,21 m	Distancia 2 (m)	Giro 2 (m) 90° = 0,21 m	Distancia 3 (m)	Total
Real	2,6	0,21	3,5	0,21	2,15	8,67
Prueba 1	2,58	0,211	3,48	0,208	2,14	8,619
Prueba 2	2,61	0,208	3,51	0,209	2,12	8,657
Prueba 3	2,57	0,21	3,49	0,212	2,14	8,622
Prueba 4	2,59	0,213	3,51	0,21	2,14	8,663
Prueba 5	2,63	0,21	3,49	0,212	2,12	8,662
Prueba 6	2,63	0,209	3,49	0,21	2,14	8,679
Prueba 7	2,61	0,212	3,51	0,213	2,15	8,695
Prueba 8	2,58	0,21	3,5	0,211	2,13	8,631

Prueba 9	2,61	0,213	3,52	0,208	2,14	8,691
Prueba 10	2,6	0,211	3,51	0,209	2,15	8,68
Prueba 11	2,59	0,208	3,49	0,212	2,13	8,63
Prueba 12	2,61	0,21	3,51	0,21	2,14	8,67
Prueba 13	2,6	0,213	3,49	0,213	2,15	8,666
Prueba 14	2,59	0,211	3,49	0,211	2,13	8,632
Prueba 15	2,6	0,211	3,51	0,208	2,16	8,678

Fuente: Los Autores

En la siguiente tabla 14, está estipulado la distancia y grados que el prototipo se desvía en el trayecto real.

Tabla 14. Errores en distancia de la Trayectoria 2

Error en Comparación con la Medida Real en la Trayectoria 2						
	Distancia 1 (m)	Giro1 (m)	Distancia 2 (m)	Giro 2 (m)	Distancia 3 (m)	Total (m)
Prueba 1	0,02	0,001	0,02	0,002	0,01	0,053
Prueba 2	0,01	0,002	0,01	0,001	0,03	0,053
Prueba 3	0,03	0	0,01	0,002	0,01	0,052
Prueba 4	0,01	0,003	0,01	0	0	0,023
Prueba 5	0,03	0	0,01	0,002	0,03	0,072
Prueba 6	0,03	0,001	0,01	0	0,01	0,051
Prueba 7	0,01	0,002	0,01	0,003	0	0,025
Prueba 8	0,02	0	0	0,001	0,02	0,041
Prueba 9	0,01	0,003	0,02	0,002	0,01	0,045
Prueba 10	0	0,001	0,01	0,001	0	0,012
Prueba 11	0,01	0,002	0,01	0,002	0,02	0,044
Prueba 12	0,01	0	0,01	0	0,01	0,03
Prueba 13	0	0,003	0,01	0,003	0	0,016
Prueba 14	0,01	0,001	0,01	0,001	0,02	0,042
Prueba 15	0	0,001	0,01	0,002	0,01	0,023

Fuente: Los Autores

En cuanto al error acumulado total como muestra la tabla 15 está establecida en 25 cm de distancia recorrida en las 5 pruebas realizadas.

Tabla 15. Error Promedio Total

Promedio de error en Trayectoria 2		
	Distancia Total (m)	Error (m)
Prueba 1	8,619	0,053
Prueba 2	8,657	0,053
Prueba 3	8,622	0,052
Prueba 4	8,663	0,023
Prueba 5	8,662	0,072
Promedio	8,6446	0,0506
Error Total		0,253

Fuente: Los Autores

Posteriormente como muestra la tabla 16 y la figura 56, indican la trayectoria promedio en comparación a la real, dando así un análisis de los valores obtenidos en comparación a los establecidos en la programación previamente, y los cuales debían ser cumplidos a cabalidad.

Tabla 16. Trayectoria 2 Promedio

Promedio de las Pruebas realizadas en la Trayectoria 2						
	Distancia 1 (m)	Giro 1 (m)	Distancia 2 (m)	Giro 2 (m)	Distancia 3 (m)	Total (m)
Promedio	2,596	0,2104	3,496	0,2102	2,132	8,6446
Real	2,6	0,21	3,5	0,21	2,15	8,67

Fuente: Los Autores

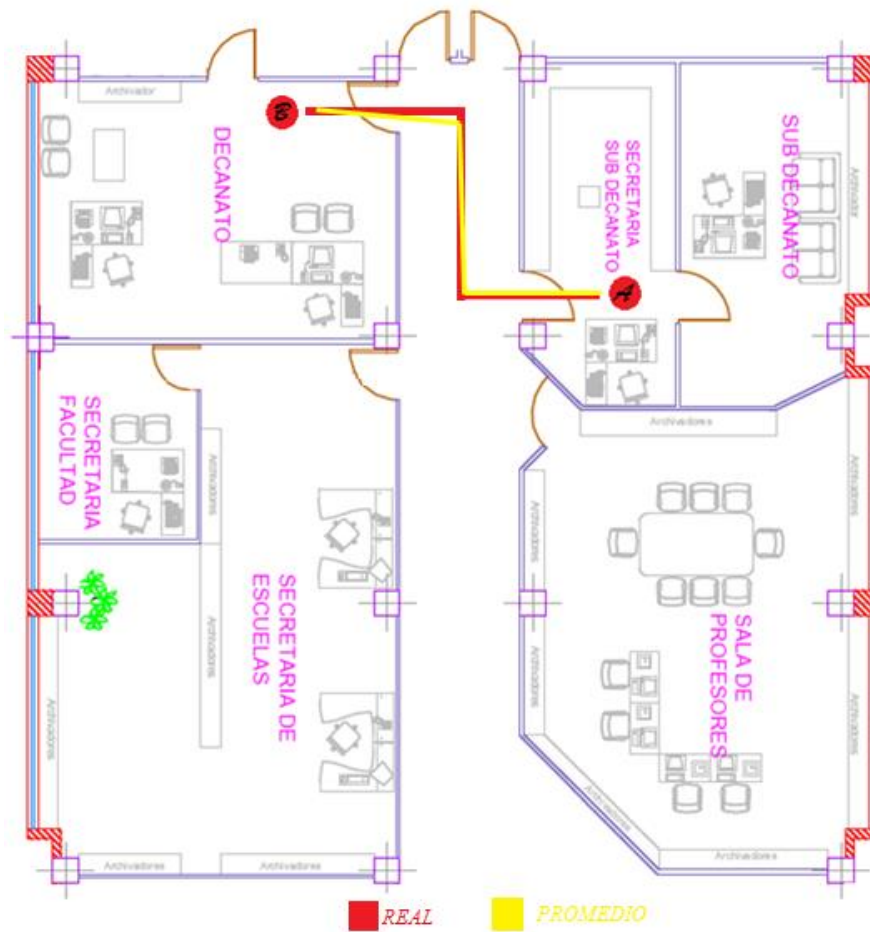


Figura 56. Trayectoria Promedio

Fuente: Los Autores

3.1.3 Trayectoria 3

En cuanto al trayecto 3, de la misma forma que el trayecto 1 y trayecto 2 se evaluó el dispositivo 5 veces, obteniendo un error promedio acumulable en trayecto realizado, con las mismas especificaciones del trayecto 1 y 2.

A continuación los valores obtenidos de los trayectos realizados y su curva de error como indica la figura 57.

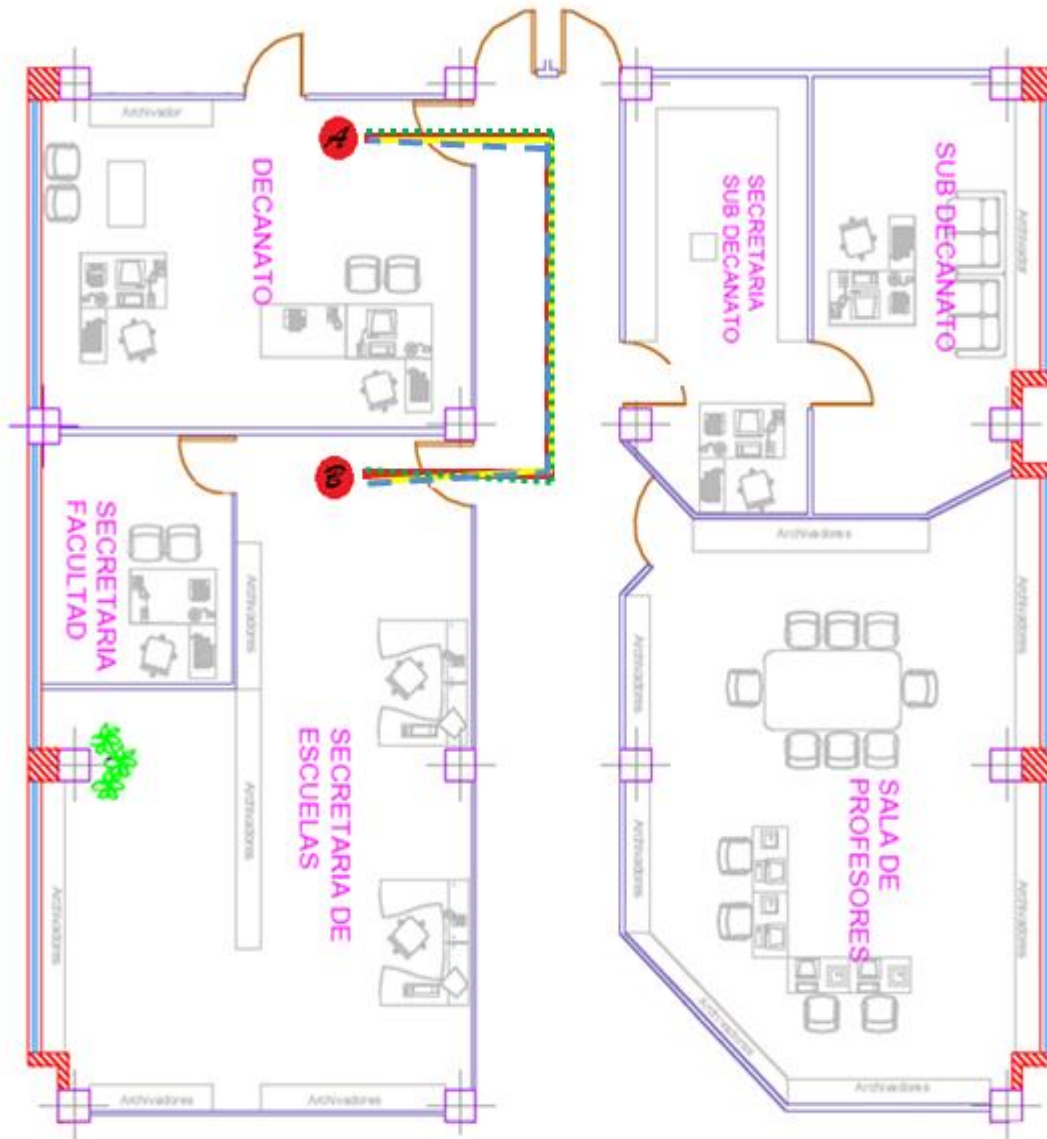


Figura 57. Trayectoria 3

Fuente: Los Autores

La tabla 17 muestra, los datos prácticos recogidos al realizar la trayectoria 3 en cada tramo.

Tabla 17. Pruebas de Trayectos

Trayectoria 3						
	Distancia 1 (m)	Giro1 (m)	Distancia 2 (m)	Giro 2 (m)	Distancia 3 (m)	Total (m)
		90° = 0,21 m		90° = 0,21 m		
Real	1,5	0,21	5	0,21	1,5	8,42
Prueba 1	1,51	0,21	5,01	0,208	1,49	8,538
Prueba 2	1,49	0,213	4,9	0,209	1,51	8,362
Prueba 3	1,51	0,21	5,02	0,212	1,49	8,482
Prueba 4	1,48	0,209	4,91	0,21	1,51	8,259
Prueba 5	1,5	0,212	5,01	0,213	1,48	8,495

Fuente: Los Autores

En la siguiente tabla 18, está estipulado la distancia y grados que el prototipo se desvía en el trayecto real.

Tabla 18. Errores en distancia de la Trayectoria 3

Error en Comparación con la Medida Real en la Trayectoria 3						
	Distancia 1 (m)	Giro1 (m)	Distancia 2 (m)	Giro 2 (m)	Distancia 3 (m)	Total (m)
Prueba 1	0,01	0	0,01	0,002	0,01	0,032
Prueba 2	0,01	0,003	0,1	0,001	0,01	0,124
Prueba 3	0,01	0	0,02	0,002	0,01	0,042
Prueba 4	0,02	0,001	0,09	0	0,01	0,121
Prueba 5	0	0,002	0,01	0,003	0,02	0,035

Fuente: Los Autores

En cuanto al error acumulado total como muestra la tabla 19 está establecida en 35 cm de distancia recorrida en las 5 pruebas realizadas y 2 grados de error en los giros

Tabla 19. Error Promedio Total

Promedio de error en Trayectoria 3		
	Distancia Total (m)	Error (m)
Prueba 1	8,538	0,032
Prueba 2	8,362	0,124
Prueba 3	8,482	0,042
Prueba 4	8,259	0,121
Prueba 5	8,495	0,035
Promedio	8,4272	0,0708
Error Total		0,354

Fuente: Los Autores

Posteriormente como muestra la tabla 20 y la figura 58, indican la trayectoria promedio en comparación a la real, dando así un análisis de los valores obtenidos en comparación a los establecidos en la programación previamente, y los cuales debían ser cumplidos a cabalidad.

Tabla 20. Trayectoria 3 Promedio

Promedio de las Pruebas realizadas en la Trayectoria 3						
	Distancia 1 (m)	Giro 1 (m)	Distancia 2 (m)	Giro 2 (m)	Distancia 3 (m)	Total
Promedio	1,498	0,2108	4,97	0,2104	1,496	8,3852
Real	1,5	0,21	5	0,21	1,5	8,42

Fuente: Los Autores

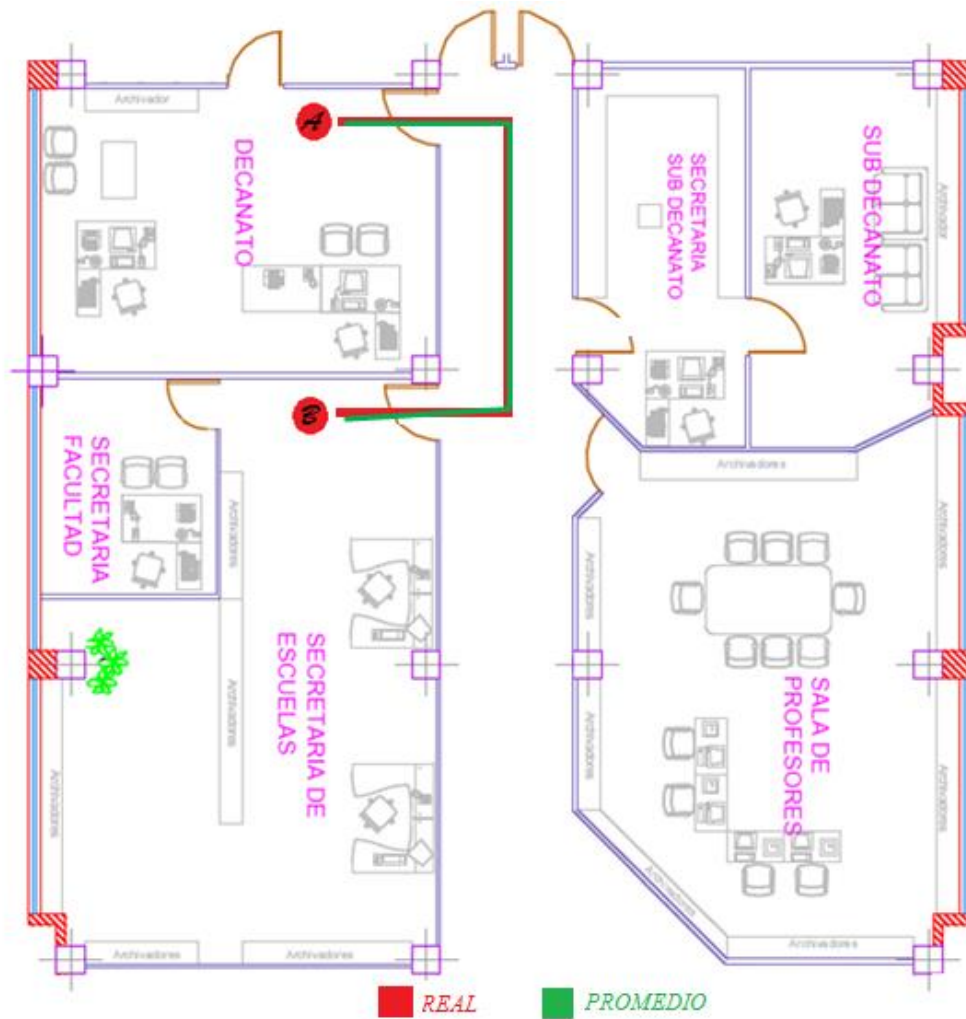


Figura 58. Trayectoria Promedio

Fuente: Los Autores

3.1.4 Trayectoria manual

En este método se podrá cargar manualmente un trayecto para su ejecución en otro entorno distinto a los ya establecidos, permitiendo así trabajar fuera de las especificaciones dadas como aparece en la figura 59.



Figura 59. Trayectoria Manual

Fuente: Los Autores

Cabe mencionar que la trayectoria tendrá su distancia establecida en números enteros dado que el microcontrolador solo comprende dichos valores, además de estar establecido en metros.

Es así que esta función será aplicable en caso de necesitar una cuarta trayectoria con valores arbitrarios y establecidos por el usuario.

El análisis de estos valores fueron contemplados por el error promedio acumulado entre uno a tres metros de distancia como indica la siguiente tabla 21.

Tabla 21. Prueba de Modo Manual

PRUEBAS EN UNA TRAYECTORIA CERRADA									
C	De	D3 (m)	Giro 2 (grados)		D2 (m)	Giro 1 (grados)		D1 (m)	De
			Izquierda	Derecha		Izquierda	Derecha		
0	0	1	0,21		1			1	0
0	0,99	1,01	0,211		1,01			1,01	0,99
		1		0,21	1			1	
		0,98		0,209	0,98			1,01	
0	0	2	0,21		2			2	0
0	2,02	2,02	0,211		2,02			2,02	0
		2		0,21	2			2	
		1,96		0,211	1,96			1,99	
0	0	0,995	0,211		0,995			1	0
0	0,209			0,209					
0	0	1,99	0,211		1,99			2,005	0
				0,211					
				0,211					
-0	-0,001	-0,005	0,001	-0,001	-0,005			0	-0
-0	0,001	-0,01	0,001	0,001	-0,01			0,005	-0

Fuente: Los Autores

3.1.5 Tiempos de Pruebas de Trayectorias con y sin Obstáculos

En las pruebas realizadas para las diferentes trayectorias se consideró dos minutos por trayectoria realizada, teniendo en cuenta que al detectar un obstáculo el prototipo se detendría por 15 segundos, de los cuales 5 segundos emitirá una alarma de aviso para que el impedimento sea retirado.

Posteriormente detectado el obstáculo y transcurrido los 15 segundos el prototipo volverá a censar, y evaluar si permanece el obstáculo aun en la trayectoria, teniendo así un mayor tiempo de espera, caso contrario el robot continuara con la trayectoria.

Tabla 22. Tiempo en Trayectoria 1

	Trayectoria 1 con 5,75 m de Recorrido				
	Tiempo (Segundos)				
	Sin Obstáculos	Con Obstáculos			
Numero de Obstáculos	0	1	2	3	4
Prueba 1	59	74	89	104	119
Prueba 2	58	73	88	103	118
Prueba 3	62	77	92	107	122
Promedio	59,67	74,67	89,67	104,67	119,67

Fuente: Los Autores

Tabla 23. Tiempo en Trayectoria 2

	Trayectoria 2 con 8,25 m de Recorrido				
	Tiempo (Segundos)				
	Sin Obstáculos	Con Obstáculos			
Numero de Obstáculos	0	1	2	3	4
Prueba 1	87	102	117	132	147
Prueba 2	85	100	115	130	145
Prueba 3	96	111	126	141	156
Promedio	89,33	104,33	119,33	134,33	149,33

Fuente: Los Autores

Tabla 24. Tiempo en Trayectoria 3

	Trayectoria 3 con 8 m de Recorrido				
	Tiempo (Segundos)				
	Sin Obstáculos	Con Obstáculos			
Numero de Obstáculos	0	1	2	3	4
Prueba 1	81	96	111	126	141
Prueba 2	83	98	113	128	143
Prueba 3	79	94	109	124	139
Promedio	81,00	96,00	111,00	126,00	141,00

Fuente: Los Autores

Cabe mencionar que las distancias estarán fijas durante el trayecto y los resultados no son alterados en función del recorrido sino en tiempo de entrega como muestran las tablas 22, 23 y 24.

CAPÍTULO IV

4. DISCUSIÓN

El desarrollo del prototipo pretende la entrega de archivos por parte de un robot mensajero, en un entorno establecido o fijo, lo más precisamente posible y de manera autónoma, no obstante sin dejar de lado la intervención del usuario en su manejo y funcionalidad, , dado que existe un error acumulativo como se demostró en las pruebas y resultados de las trayectorias del capítulo anterior, siendo estos de 6 a 15 centímetros y entre 2° o 3°, dependiendo de la distancia del trayecto, además de un sistema de adquisición de datos los cuales serían utilizados para diversos parámetros en el diseño del robot.

Dichos datos serán adquiridos por tres sensores establecidos en el frente del robot, ubicados en diversas posiciones examinadas anteriormente, para una adecuada recopilación de valores, los cuales evaluarían objetos uniformes o no uniformes, estáticos o móviles, mediante un algoritmo basado en distancia, emisión y recepción de sonidos desde los sensores hacia el posible obstáculo, permitiendo al robot el evitar un objeto extraño a la trayectoria determinada.

Además de trabajar con un encoder acoplado a una de las llantas para determinar distancia mediante un sistema de pulsos adquiridos por un encoder óptico en un esquema de 16 estados, los cuales estarían 8 en alto y 8 en bajo, considerando además el diámetro y la circunferencia de la llanta para el desarrollo de un algoritmo de distancia recorrida y posicionamiento del dispositivo.

Así pues terminado el prototipo este es capaz de realizar una trayectoria en un entorno o área con posible presencia de obstáculos de manera eficaz y con mínimo error no obstante con la periódica evaluación de su operador en el lugar a desempeñarse.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- El error promedio evaluado en cada una de las trayectorias indica entre 5 y 10 cm de desviación del robot en su recorrido, siendo necesario la intervención del usuario para su control y verificación de su óptimo funcionamiento, a su vez con un tiempo de entrega de 2 a 5 minutos.
- La distancia establecida de manejo del robot es muy limitada, dado el medio de comunicación utilizado entre el robot y el dispositivo móvil, no permitiendo así recorrer distancias mayores a los 15 metros.
- Ubicando correctamente los sensores ultrasónicos con 30° de separación entre cada uno, el robot elimina posibles puntos ciegos que pudiera encontrar en la trayectoria que estuviese recorriendo.
- Al momento de evaluar el robot en los distintos entornos en los que trabajaría, reveló que el suelo sería un factor a analizar y el cual al ser liso, requeriría colocar un material antideslizante en las llantas del prototipo, que permitiría una mayor adherencia al piso y un óptimo funcionamiento del mismo.
- El medio de comunicación implementada fue el protocolo Bluetooth, el cual es un medio de comunicación muy confiable, dado que los datos enviados y recibidos presentan pocas pérdidas, consiguiendo así un margen de error mínimo.

- La alimentación del robot está comprendida por 7.2V a 1.3A para el microcontrolador y de 7.2V a 2.3A para el funcionamiento de los servomotores con su respectivo regulador de voltaje en caso de una sobrecarga.
- Otro medio de posicionamiento analizado para el robot mensajero fue el sistema GPS, pero al ser utilizado solo en exteriores, no cumplía con los requerimientos del presente proyecto, así como radio frecuencia, el cual era de un gran costo y no entraba dentro del presupuesto planteado para el robot.
- La implementación del robot mensajero en el área administrativa de la Facultad de Ingeniería no es factible, debido a aspectos electromecánicos como son las llantas, que no permiten una mejor adherencia al suelo y más estabilidad en la adquisición de datos en el robot.
- Como trabajo futuro para mejorar el prototipo debe implementarse un sistema de control para la corrección automática de trayectorias en el robot mensajero, así como la implementación de un sistema electromecánico para el seguimiento de trayectorias impresas en el área a desempeñarse.

5.2 Recomendaciones

- Para un óptimo funcionamiento entre el dispositivo móvil y el robot debe encenderse previamente el mensajero para de esta manera enlazar el celular con la aplicación hacia el dispositivo después de 1 minuto de encendido.
- Tomar en consideración otra manera de alimentación al dispositivo móvil, así como un mejor método de carga de energía hacia las baterías, para una mayor eficiencia en el tiempo de manejo o utilización.
- Establecer un material idóneo no deslizante y adherente para las llantas así como su elaboración en otro material para un óptimo funcionamiento en otras superficies no determinadas.
- Para que el robot pueda almacenar un número mayor de trayectorias, debe disponerse de un dispositivo alternativo de memoria, para su administración tomando en consideración las variables disponibles en la aplicación creada para el mensajero.
- En la estructura, se puede establecer otro tipo de material o incluso un elemento moldeable para una presentación menos rústica y más aceptable en cuanto al diseño.
- Obtener un financiamiento idóneo que permita la implementación de otras tecnologías en el robot mensajero como: radiofrecuencia o Wi-Fi, para desarrollar un sistema de corrección automático de trayectorias.

CAPÍTULO VI

6. PROPUESTA

6.1 Título de la Propuesta

Implementación de un sistema de control para la corrección automática de trayectorias en el robot mensajero.

6.2 Introducción

Para el óptimo funcionamiento del robot para servicio de mensajería interna se requiere un sistema de control para la corrección automática del seguimiento de trayectorias, que evitara desviaciones producidas por los errores en el ángulo de giro que realiza el robot además del error acumulativo del encoder incremental permitiendo la mayor exactitud del trayecto.

Para lograr el objetivo principal el sistema de control será desarrollado de acuerdo a los requerimientos necesarios, para poder determinar la variación de la señal de entrada de los sensores de distancia cuando se produce un desvío en la trayectoria y así poder accionar los actuadores como son los servomotores para obtener la corrección necesaria y lograr la precisión de las trayectorias.

6.3 Objetivos

6.3.1 General

- Implementar un sistema de control para la corrección automática de trayectorias en el robot mensajero.

6.3.2 Específicos

- Investigar sistemas de control automático de corrección de trayectorias analizando las ventajas y desventajas.
- Desarrollar el sistema de control automático para evitar desviaciones en los trayectos.
- Realizar las pruebas de funcionamiento y el análisis de los resultados para determinar la eficiencia y limitación del sistema de control.

6.4 Fundamentación Científica – Técnica

Para la implementación del sistema de corrección automática de trayectorias en el robot de servicio de mensajería, se debe considerar diversos componentes que formaran parte en el desarrollo y ejecución, los cuales serán específicos para realizar la implementación del sistema en el robot, a continuación se detallaran cada uno de ellos:

Controlador.El controlador será la parte más importante del sistema porque responderá a la variación de las señales de entrada de los sensores tanto ultrasónicos como del encoder, estableciendo en la salida el accionamiento de los actuadores que permitirá ejecutar el proceso establecido.

Sensor.El robot mensajero poseerá varios sensores que servirán para obtener los valores de las señales variables que se van a medir en el sistema del control.

Robot. Es el sistema electromecánico programable capaz de realizar tareas en este caso el servicio de mensajería, estará dotado de sensores para percibir y evitar colisiones en el trascurso de la ruta establecida con la mayor precisión hasta la finalización de la misma.

Actuador. Los actuadores para este tema de estudio serán los servomotores que reaccionarán para evitar las desviaciones en el transcurso de los trayectos.

6.5 Descripción de la Propuesta

La implementación del sistema de corrección automática de trayectos en el robot de servicio de mensajería interna, permitirá disminuir los errores que se producen en el ángulo de giro y el error acumulativo del encoder incremental, logrando mayor precisión en el seguimiento de la trayectoria sin intervención del control manual posibilitando mayor eficiencia en la entrega y recepción de documentos o archivos en el área administrativa del segundo piso de la Facultad de Ingeniería.

Para cumplir este propósito se realizara una investigación y análisis de diversos sistemas de control que posibilitara la mejor elección del sistema a utilizar, además de desarrollar el sistema de control automático, el cual se encargara de dar las ordenes que ejecutaran los actuadores, en este caso las funciones de los servomotores y la realimentación del sistema serán los datos de entrada de los sensores ultrasónicos y el encoder incremental, lo cual permitirá al robot seguir sin desviaciones y culminar la trayectoria establecida

Por consiguiente para la implementación del sistema de control en el robot de mensajería interna se debe efectuar las pruebas de funcionamiento, tomando en cuenta las observaciones y sugerencias realizadas, para mejoramiento del mismo.

6.6 Diseño Organizacional

En la siguiente figura 58. Se observa, el esquema organizacional de las personas que participaran en el desarrollo y la implementación de la propuesta.

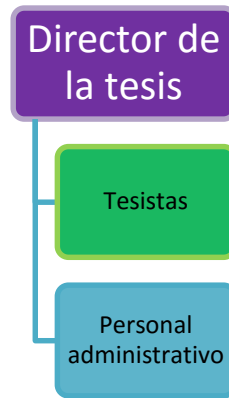


Figura 60. Esquema Organizacional

Fuente: Los Autores

6.7 Monitoreo y Evaluación de la propuesta.

Para el monitoreo y evaluación del funcionamiento del sistema de control automático de desviaciones en las trayectorias del robot mensajero, se realizará las pruebas necesarias en lapsos de tiempo comprendidos en las 8 horas laborables que trabaja el personal, por lo que se evaluará la precisión en el seguimiento de la trayectoria hasta la culminación de la misma, comparando la trayectoria real con la realizada por el robot sin intervención manual, obteniendo resultados lo más confiables, para poder mejorar el sistema y determinar las limitaciones del mismo.

CAPÍTULO VII

7. BIBLIOGRAFÍA

- ATMEL. (24 de mayo del 2013). Atmel Corporation
Atmega640/1280/1281/2560/2561 DatasheetSummary. Obtenido de:
<http://silabs.org.ua/bc4/hc06.pdf12>
- Báez, M., Borrego, A., Cordero, L., González, M., Hernández, F., Palomero, D.,... Zapata, A. (2012). *Introducción a Android*. Madrid, España: E.M.E.
- Baz, A., Ferreira, I., Álvarez, M. y García, R. (2009). *Dispositivos Móviles*. Oviedo: Universidad de Oviedo.
- Baturone, A. O. (2011). *Manipuladores y robots móviles*. Barcelona, España: Marcombo.
- Camacho, R. (17 de Marzo de 2012). *Computo Integrado* Obtenido de
<http://rcmcomputointegrado.blogspot.com>:
<http://rcmcomputointegrado.blogspot.com/2012/03/microcontroladores-vs-microprocesadores.html>
- Chiarani, M., Torres, S., y Tapia, M. (2013). *Boletín de docentes conectados*. Argentina: Compartigual.
- Constain, J., & López, J. (2012). *Diseño e Implementación de la Teleoperación de un robot móvil (Tesis de pregrado)*. Universidad Autónoma de Occidente, Santiago de Cali, Colombia.
- Elecfreaks. (12 de agosto del 2013). Elecfreaks.com. Obtenido de:
http://www.elecfreaks.com/store/download/product/Sensor/HC-SR04/HC-SR04_Ultrasonic_Module_User_Guide.pdf

- Jun Hee, L. (1 de abril del 2002). Obtenido de:
<http://users.ece.utexas.edu/~valvano/Datasheets/ServoHS311.pdf>
- Pérez, D. (2006). *Sensores de distancia por ultrasonidos*. Obtenido de:
http://picmania.garcia-cuervo.net/recursos/redpictutorials/sensores/sensores_de_distancias_con_ultrasonidos.pdf
- Ramírez, D., Ramírez, J., Ávila, E. & Aguirre, E. (2013). *Investigación en fútbol robótico sobre plataforma lego Mindstorm (Tesis de pregrado)*. Corporación Universitaria Minuto de Dios, Bogota, Colombia.
- Ricillo, M., (2012). *Robótica entra al mundo de la inteligencia artificial*. Obtenido de:
<http://bibliotecadigital.educ.ar/uploads/contents/ROBOTICA1.pdf>
- Rodríguez, J., Sánchez, A., & Atheortua, L. (2009). *Inteligencia artificial aplicada a la robótica (Proyecto)*. Universidad de Medellín, Medellín, Colombia.
- Sharp Corporation. (31 de marzo del 2006). *gp1s53v_e*. Obtenido de: http://sharp-world.com/products/device/lineup/data/pdf/datasheet/gp1s53v_e.pdf
- STMicroelectronics. (9 de enero del 2008). *Dual full-bridge driver*. Obtenido de:
https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf
- UAO. (2012). *Universidad Autónoma del Occidente*. Obtenido de
http://augusta.uao.edu.co/moodle/file.php/2896/PRACTICA-1_2012-3.pdf
- Verle, M. (2010). *Microcontroladores PIC - Programación en BASIC*. Recuperado de <http://www.mikroe.com/chapters/view/84/libro-de-la-programacion-de-los-microcontroladores-pic-en-basic-capitulo-1-mundo-de-los-microcontroladores/>
- Yu, E. (18 de abril del 2011). *Guangzhou HC Information Technology* Obtenido de:
<http://dlmnh9ip6v2uc.cloudfront.net/datasheets/Components/General%20IC/2549S.pdf>

CAPÍTULO VIII

8. ANEXOS

8.1 Anexo 1

Manual de Usuario

Antes de comenzar verifique que el robot tenga instalada la alimentación para el micro controlador que se encuentra en la parte superior, a lado del Display o pantalla, que consta de 6 pilas AA recargables de 1.2 voltios (V) y 2300 miliamperios (mA), además asegúrese de la alimentación para los servomotores que se encuentra en la parte inferior del robot, que posee 6 pilas AA recargables de 1.2 V y 1400 mA, las cuales deben estar previamente cargadas.

También tiene que poseer un dispositivo móvil con el sistema operativo Android y bluetooth, debe instalar la aplicación en su dispositivo, la cual la obtendrá de la persona encargada del robot mensajero.

Encendido del Robot


Presione el interruptor de color azul de la parte inferior del robot, constatará el encendido cuando observe que se ilumina un led o foco de color verde, seguidamente prenda el interruptor de color negro que se encuentra en la parte superior, y percibirá que se enciende la pantalla con el mensaje ROBOT MENSAJERO además de un foco que variara su luz entre azul, rojo y verde.

Vinculación del dispositivo móvil con el robot

Para vincular el dispositivo móvil con el robot debe primero encender el bluetooth de su dispositivo móvil, luego debe ir a ajustes, bluetooth, buscar dispositivos hasta encontrar a linvor, cuando realice la solicitud de vinculación le pedirá una clave, debe digitar 1234, y el dispositivo se vinculara con el robot.

Este proceso lo realizara una vez por cada dispositivo móvil, para no realizar de nuevo este procedimiento debe tener activado en el dispositivo móvil permitir la conexión automáticamente con linvor.

Manejo de la Aplicación

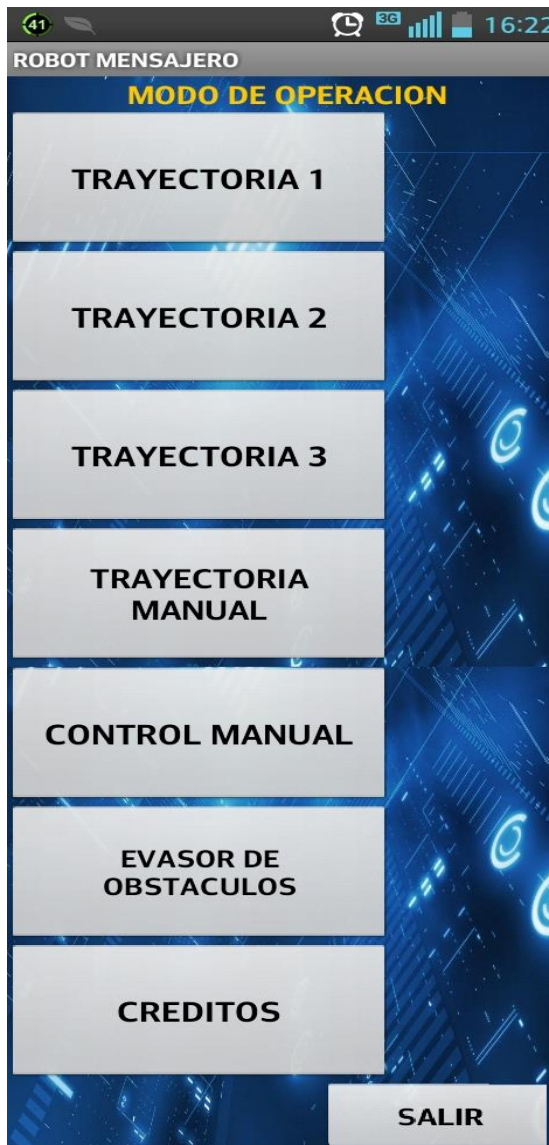
Ingrese en el icono de la aplicación Robot Mensajero , la primera pantalla que aparecerá es la que se muestra en la siguiente Figura:



Pantalla de inicio de la Aplicación

Digite la contraseña **clave**, si digita mal la contraseña le aparecerá **Error de Clave** vuelva a introducir la contraseña y luego presione en el icono **Conectar**, si presiona el icono **Salir**, saldrá automáticamente de la aplicación.

A continuación luego de presionar el icono **Conectar**, mostrara la siguiente Pantalla que contiene el Modo de operación del robot.



Pantalla modo operación.

Seleccione un modo de operación, tocando uno de los iconos, si presiona el botón **Salir**lo regresara a la pantalla de inicio de la aplicación, las funciones de los demás iconos se detallaran subsecuentemente:

Al seleccionar **Trayectoria 1, 2 o 3** se visualizaran una de las próximas pantallas:



Trayectoria 1

Trayectoria 2

Trayectoria 3

La **Trayectoria 1, 2 o 3**, al tocar el botón **Inicio** recorre la ruta como se muestra en las gráficas del punto A hacia el B, cuando durante el recorrido se encuentra con un obstáculo el robot se detendrá y emitirá un sonido para que se retire el obstáculo, y así poder proseguir con el trayecto, cuando culmine con el trayecto se escuchará un mensaje (Trayectoria Terminada) en el dispositivo móvil.

El botón **Paro**, sirve para reiniciar los valores después que se termine la trayectoria.

El botón **Retorno**, realizará la trayectoria desde el punto B hacia el A.

El botón **Salir**, regresará hacia la pantalla de modo Operación.

En la **Trayectoria Manual**, se podrá crear una trayectoria x, cargando los valores en distancia en metros del 0 al 9 y el giro que desea hacer de 0 a 360 grados, no importa si no llena todos los campos los puede dejar con 0, para que realice la trayectoria debe tocar el botón **Inicio**, el botón **Borrar** eliminará los datos ingresados, el de

Retornohará la trayectoria en sentido contrario para regresar y el botón **Salir** regresara a la pantalla modo de operación.



Trayectoria Manual

En el **Control Manual**, usted podrá manejar al robot de manera remota, presionando cualquiera de estos iconos con las siguientes funciones: para seguir hacia adelante



, para girar a la izquierda




, para girar a la derecha



, para retroceder



para detener al robot  y para regresar al modo operación toque el botón **Salir**.



Control Manual

En la función **Evasor de Obstáculos**, al tocar el botón **Inicio**, el robot comenzara a evadir obstáculos que se encuentren en el entorno evitando colisiones. Para detener al robot presione el botón **Paro** y para regresar a la pantalla modo operación presione el botón **Salir**.



Evasor de Obstáculos

En los **Créditos**, se observará una pantalla con los nombres de las personas que realizaron la aplicación, toque el botón **Salir** para regresar a la pantalla modo de operación.



Créditos

8.2 Anexo 2

Programación del robot mensajero

```
//Robot Mensajero

//Declaración de Librerías

#include <EEPROM.h>
#include "pitches.h" //llama la rutina de sonidos Buzzer

// Importacion de librerías

#include <LiquidCrystal.h> //librería LCD 16x2
#include <Ultrasonic.h>    //librería sensor Ultrasonico
#include <Wire.h>         //librería comunicación I2C
#include <I2Cdev.h>
#include <MPU6050.h>      //librería MPU6050
MPU6050 mpu;            //nombramos al sensor como "mpu"

// Declaración de Variables y entradas

int16_t ax, ay, az;
int16_t gx, gy, gz;
int melody[ ] = {
NOTE_C4,  NOTE_G3,  NOTE_G3,  NOTE_A3,  NOTE_G3,0,  NOTE_B3,
NOTE_C4};

// Duración de notas: 4 = quarter note, 8 = eighth note, etc.:

intnoteDurations[] = { //notas robot detenido
  10, 2, 4 };

int noteDurations1[] = { //notas trayectoria terminada
  4, 8, 8, 4,4,4,4,4 };

//Declaración de Entrada y Salidas

int motorI1 = 23; //control I1 motor A
int motorI2 = 25; //control I2 motor A
intmotorEA = 8; //control PWM ENABLE motor A
int motorI3 = 27; //control I1 motor B
int motorI4 = 29; //control I1 motor B
intmotorEB = 9; //control PWM ENABLE motor B
int azulDerecho = 44; //pines diodo RGB 1
```

```

intverdeDerecho = 42;
introjoDerecho = 40;

introjoCentro = 38; //pines diodo RGB 2
intverdeCentro = 36;
intazulCentro = 34;

introjoIzquierdo = 32; //pines diodo RGB 3
intverdeIzquierdo = 30;
intazulIzquierdo = 28;

introjoPrincipal = 11; //pines diodo RGB 4 conectados a 3 PWM
intverdePrincipal = 12;
intazulPrincipal = 13;

intdireccionA=0;
intdireccionI=10;
intdireccionD=20;

//Nombra Y Declara Los Pines Para Los Sensores De Ultrasonido

UltrasonicultraCentro(47, 49); // (TrigPIN,Echo PIN)
UltrasonicultraIzquierdo(43,45); // (TrigPIN,Echo PIN)
UltrasonicultraDerecho(51,53); // (TrigPIN,Echo PIN)

//Declaración De Variables Y Constantes

intvelAvance = 200; //velocidad de avance salida ENABLE PWM de 0 a 255
intvelGiro = 250; //velocidad de giro salida ENABLE PWM de 0 a 255

int EM2 = 0; //Encoder motor (Derecha) (Entrada 2)
volatile long CRD = 0; //Contador encoder parcial
volatile long CRT = 0; //contador total recorrido
longdist = 0; //Distancia del robot (cm)

longdistCentro = 0;
longdistIzq = 0; // Distancia del robot (cm) a su izquierda
longdistDer = 0; // Distancia del robot (cm) a su derecha
byte giro = 0; //Variable para saber por dónde girar
byte caso = 0; //Para seleccionar el caso de giro
byteredireccionado= 0;
char val; //variable que recibe los datos por el puerto serial 3
intvalaz= 0; //variable z giroscopio
int distancia = 0; //distancia que recorre el robot en metros

```

```

intangulo =0;                // ángulo que gira el robot
intanguloIzquierda = 0;

LiquidCrystallcd(41, 39, 37, 35, 33, 31); //nombra y declara pines para el LCD
(RS,E,D4,D5,D6,D7);

//Estados de Motores

voidsetup(){
pinMode(motorI1,OUTPUT);
pinMode(motorI2,OUTPUT);
pinMode(motorEA,OUTPUT);
pinMode(motorI3,OUTPUT);
pinMode(motorI4,OUTPUT);
pinMode(motorEB,OUTPUT);
pinMode(rojoDerecho,OUTPUT);
pinMode(verdeDerecho,OUTPUT);
pinMode(azulDerecho,OUTPUT);
pinMode(rojoIzquierdo,OUTPUT);
pinMode(verdeIzquierdo,OUTPUT);
pinMode(azulIzquierdo,OUTPUT);
pinMode(rojoCentro,OUTPUT);
pinMode(verdeCentro,OUTPUT);
pinMode(azulCentro,OUTPUT);
pinMode(rojoPrincipal,OUTPUT);
pinMode(verdePrincipal,OUTPUT);
pinMode(azulPrincipal,OUTPUT);

// LCD DE 16x2
lcd.begin(16, 2);
lcd.setCursor(0, 0);
lcd.print("ROBOT");
lcd.setCursor(0, 1);
lcd.print("MENSAJERO");

for(intfadeValue = 0 ; fadeValue<= 255; fadeValue +=5) {

// ENCIENDE LOS LEDS PRINCIPALES CON PWM
analogWrite(rojoPrincipal, fadeValue);
delay(15);
}
for(intfadeValue = 255 ; fadeValue>= 0; fadeValue -=5) {
analogWrite(rojoPrincipal, fadeValue);
delay(15);
}
}

```

```

for(intfadeValue = 0 ; fadeValue<= 255; fadeValue +=5) {
analogWrite(verdePrincipal, fadeValue);
delay(15);
}
for(intfadeValue = 255 ; fadeValue>= 0; fadeValue -=5) {
analogWrite(verdePrincipal, fadeValue);
delay(15);
}

```

```

encendidoLeds();
lcd.clear();
pinMode(2, INPUT);
digitalWrite(2, HIGH);
attachInterrupt(EM2, ContadorRuedaDer, CHANGE);
Wire.begin();
mpu.initialize();

```

//SE INICIA LA COMUNICACIÓN SERIAL CON EL BLUETOOTH A 9600BPS

```

Serial3.begin(9600);
Serial.begin(9600);
}
void loop() {
recibirDatos();

}

```

CONTROL MANUAL

```

voidrecibirDatos() {
if (Serial3.available()) {
val = Serial3.read();
}
}

```

ROBOT HACIA ADELANTE EN CONTROL MANUAL

```

if (val == 'A') {
goAdelante();
lcd.setCursor(0, 0);
lcd.print("DISTANCIA: ");
lcd.print(CRT*0.0225);
mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
valaz = map(az, -17000, 17000, 0, 179);
lcd.setCursor(0, 1);
lcd.print("ANGULO: ");
}

```

```

lcd.print(valaz);

}
ROBOT SE DETIENE EN CONTROL MANUAL
else if (val == 'B') {
goParada();
mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
valaz = map(az, -17000, 17000, 0, 179);
lcd.setCursor(0, 1);
lcd.print("ANGULO: ");
lcd.print(valaz);

}

ROBOT HACIA LA DERECHA EN CONTROL MANUAL
else if (val == 'C') {
goDerecha();
mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
valaz = map(az, -17000, 17000, 0, 179);
lcd.setCursor(0, 1);
lcd.print("ANGULO: ");
lcd.print(valaz);

}
ROBOT HACIALA IZQUIERDA EN CONTROL MANUAL
else if (val == 'D') {
goIzquierda();
mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
valaz = map(az, -17000, 17000, 0, 179);
lcd.setCursor(0, 1);
lcd.print("ANGULO: ");
lcd.print(valaz);

}
ROBOT HACIA ATRAS EN CONTROL MANUAL
else if (val == 'E') {
goAtras();
mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
valaz = map(az, -17000, 17000, 0, 179);
lcd.setCursor(0, 1);
lcd.print("ANGULO: ");
lcd.print(valaz);
}
DISTANCIA EPROM
else if (val == 'F') {

```

```

EEPROM.write(direccionA,CRD*0.0225);
direccionA=direccionA+1;
val='0';
CRD=0;
}
DERECHA EPROM
else if (val == 'G') {
EEPROM.write(direccionD,CRD*5.625);
direccionD=direccionD+1;
val='0';
CRD=0;
}
IZQUIERDA EPROM
elseif (val == 'H') {
EEPROM.write(direccionI,CRD*5.625);
direccionI=direccionI+1;
val='0';
CRD=0;
}
BORRAR EPROM
else if (val == 'I') {
for (inti = 0; i< 512; i++)
EEPROM.write(i, 0);
goParada();
CRT=0;
CRD=0;
}
INICIO EPROM
elseif (val == 'J') {
avanzar(EEPROM.read(0));
giroDerecha(EEPROM.read(10));
giroIzquierda(EEPROM.read(20));
avanzar(EEPROM.read(1));
giroDerecha(EEPROM.read(11));
giroIzquierda(EEPROM.read(21));
avanzar(EEPROM.read(2));
giroDerecha(EEPROM.read(12));
giroIzquierda(EEPROM.read(22));
avanzar(EEPROM.read(3));
giroDerecha(EEPROM.read(13));
giroIzquierda(EEPROM.read(23));
}
INICIO TRAYECTORIA 1
else if (val == 'K') {
Serial3.println('9');

```

```

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("TRAYECTORIA 1 ");
apagarLeds();
rgbPrincipal(0,0,255);
  CRT=0;
avanzar(0.5);
giroIzquierda(90);
avanzar(0.5);
giroDerecha(90);
avanzar(0.5);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("DISTANCIA: ");
lcd.print(CRT*0.0225);
mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
valaz = map(az, -17000, 17000, 0, 179);
lcd.setCursor(0, 1);
lcd.print("ANGULO: ");
lcd.print(valaz);
Serial3.println('6');
rgbPrincipal(0,255,0);
tonoLlegada();
encendidoLeds();
val='M';
lcd.clear();
}

```

EL ROBOT PARA TRAYECTORIA 1

```

else if (val == 'M') {
goParada();
Serial3.println('8');
rgbPrincipal(255,0,0);
CRT=0;
CRD=0;
}

```

TRAYECTORIA 1 RETORNO

```

else if (val == 'N') {
Serial3.println('7');
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("TRAYECTORIA 1 ");
apagarLeds();
rgbPrincipal(0,0,255);

```

```

    CRT=0;
    giroDerecha(180);
    avanzar(0.5);
    giroDerecha(90);
    avanzar(0.5);
    giroIzquierda(90);
    avanzar(0.5);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("DISTANCIA: ");
    lcd.print(CRT*0.0225);
    mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
    valaz = map(az, -17000, 17000, 0, 179);
    lcd.setCursor(0, 1);
    lcd.print("ANGULO: ");
    lcd.print(valaz);
    Serial3.println('6');
    rgbPrincipal(0,255,0);
    tonoLlegada();
    encendidoLeds();
    Serial3.println(CRT);
    val='M';
    lcd.clear();
}
TRAYECTORIA 2
else if (val == 'O') {
    Serial3.println('9');
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("TRAYECTORIA 2 ");
    apagarLeds();
    rgbPrincipal(0,0,255);
    CRT=0;
    avanzar(0.5);
    giroDerecha(90);
    avanzar(0.5);
    giroIzquierda(90);
    avanzar(0.5);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("DISTANCIA: ");
    lcd.print(CRT*0.0225);
    mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
    valaz = map(az, -17000, 17000, 0, 179);
    lcd.setCursor(0, 1);

```



```

lcd.print("ANGULO: ");
lcd.print(valaz);
Serial3.println('6');
rgbPrincipal(0,255,0);
tonoLlegada();
encendidoLeds();
Serial3.println(CRT);
val='Q';
lcd.clear();
}
EL ROBOR PARA TRAYECTORIA 2
else if (val == 'Q') {
goParada();
Serial3.println('8');
rgbPrincipal(255,0,0);
CRT=0;
CRD=0;
}
EL ROBOT RETORNA TRAYECTORIA 2
else if (val == 'R') {
Serial3.println('7');
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("TRAYECTORIA 2 ");
apagarLeds();
rgbPrincipal(0,0,255);
  CRT=0;
giroDerecha(180);
avanzar(0.5);
giroDerecha(90);
avanzar(0.5);
giroIzquierda(90);
avanzar(0.5);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("DISTANCIA: ");
lcd.print(CRT*0.0225);
mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
valaz = map(az, -17000, 17000, 0, 179);
lcd.setCursor(0, 1);
lcd.print("ANGULO: ");
lcd.print(valaz);
Serial3.println('6');
rgbPrincipal(0,255,0);

```

```

    tonoLlegada();
    encendidoLeds();
    Serial3.println(CRT);
    val='Q';
    lcd.clear();
}
TRAYECTORIA 3
else if (val == 'S') {
    Serial3.println('9');
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("TRAYECTORIA 3 ");
    apagarLeds();
    rgbPrincipal(0,0,255);
    CRT=0;
    avanzar(0.5);
    giroDerecha(90);
    avanzar(0.5);
    giroDerecha(90);
    avanzar(0.5);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("DISTANCIA: ");
    lcd.print(CRT*0.0225);
    mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
    valaz = map(az, -17000, 17000, 0, 179);
    lcd.setCursor(0, 1);
    lcd.print("ANGULO: ");
    lcd.print(valaz);
    Serial3.println('6');
    rgbPrincipal(0,255,0);
    tonoLlegada();
    encendidoLeds();
    val='T';
    lcd.clear();
}
ROBOT PARA TRAYECTORIA 3
else if (val == 'T') {
    goParada();
    Serial3.println('8');
    rgbPrincipal(255,0,0);
    CRT=0;
    CRD=0;
}
ROBOT RETORNA 3

```

```

else if (val == 'U') {
Serial3.println('7');
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("TRAYECTORIA 3 ");
apagarLeds();
rgbPrincipal(0,0,255);
  CRT=0;
giroDerecha(180);
avanzar(0.5);
giroDerecha(90);
avanzar(0.5);
giroIzquierda(90);
avanzar(0.5);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("DISTANCIA: ");
lcd.print(CRT*0.0225);
mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
valaz = map(az, -17000, 17000, 0, 179);
lcd.setCursor(0, 1);
lcd.print("ANGULO: ");
lcd.print(valaz);
Serial3.println('6');
rgbPrincipal(0,255,0);
tonoLlegada();
encendidoLeds();
Serial3.println(CRT);
val='T';
lcd.clear();
}
INICIA LECTURA DE STREAM DE DATOS PARA LA TRAYECTORIA
MANUAL

```

```

elseif (val == 'V') {
delay(50);//Cuando haya datos disponibles
while (Serial3.available(>0)
{
int distancia1 = Serial3.parseInt();
int derecha1 = Serial3.parseInt();
int izquierda1 = Serial3.parseInt();
int distancia2 = Serial3.parseInt();
int derecha2 = Serial3.parseInt();
int izquierda2 = Serial3.parseInt();
int distancia3 = Serial3.parseInt();

```

```

int derecha3 = Serial3.parseInt();
int izquierda3 = Serial3.parseInt();
int distancia4 = Serial3.parseInt();
if (Serial3.read() == '\n')
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(">>TRAYECTORIA ");
  lcd.setCursor(0, 1);
  lcd.print(">>>MANUAL ");
  apagarLeds();
  rgbPrincipal(0,0,255);
  // Serial3.println('V');
  CRT=0;
  avanzar(distancia1);
  giroDerecha(derecha1);
  giroIzquierda(izquierda1);
  avanzar(distancia2);
  giroDerecha(derecha2);
  giroIzquierda(izquierda2);
  avanzar(distancia3);
  giroDerecha(derecha3);
  giroIzquierda(izquierda3);
  avanzar(distancia4);
  //Serial3.println('W');
  rgbPrincipal(0,255,0);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("DISTANCIA: ");
  lcd.print(CRT*0.0225);
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
  valaz = map(az, -17000, 17000, 0, 179);
  lcd.setCursor(0, 1);
  lcd.print("ANGULO: ");
  lcd.print(valaz);
  tonoLlegada();
  encendidoLeds();
  val='X';
  lcd.clear();
}
}
}

```

INICIA LECTURA DE STREAM DE DATOS PARA LA TRAYECTORIA MANUAL

```
else if (val == 'X') {    for (inti = 0; i< 512; i++)
EEPROM.write(i, 0);
    CRT=0;
}
else if (val == 'Y') {
delay(50);
while (Serial3.available(>0)
{
int distancia1 = Serial3.parseInt();
int derecha1 = Serial3.parseInt();
int izquierda1 = Serial3.parseInt();
int distancia2 = Serial3.parseInt();
int derecha2 = Serial3.parseInt();
int izquierda2 = Serial3.parseInt();
int distancia3 = Serial3.parseInt();
int derecha3 = Serial3.parseInt();
int izquierda3 = Serial3.parseInt();
int distancia4 = Serial3.parseInt();
if (Serial3.read() == '\n')
{
lcd.clear();
lcd.setCursor(0, 0);
lcd.print(">>TRAYECTORIA ");
lcd.setCursor(0, 1);
lcd.print(">>>MANUAL ");
apagarLeds();
rgbPrincipal(0,0,255);
    CRT=0;
Serial3.println('Y');
giroDerecha(180);
avanzar(distancia4);
giroDerecha(izquierda3);
giroIzquierda(derecha3);
avanzar(distancia3);
giroDerecha(izquierda2);
giroIzquierda(derecha2);
avanzar(distancia2);
giroDerecha(izquierda2);
giroIzquierda(derecha2);
avanzar(distancia1);
Serial3.println('W');
rgbPrincipal(0,255,0);
```

```

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("DISTANCIA: ");
lcd.print(CRT*0.0225);
mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
valaz = map(az, -17000, 17000, 0, 179);
lcd.setCursor(0, 1);
lcd.print("ANGULO: ");
lcd.print(valaz);
tonoLlegada();
encendidoLeds();
val='X';
lcd.clear();
    }
}
}

```

RUTINA DE EVASOR DE OBSTACULOS

```

else if (val == '0') {
  apagarLeds();
  analogWrite(azulPrincipal, 255);
  lcd.setCursor(0, 0);
  lcd.print("DISTANCIA: ");
  lcd.print(CRT*0.0225);
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
  valaz = map(az, -17000, 17000, 0, 179);
  lcd.setCursor(0, 1);
  lcd.print("ANGULO: ");
  lcd.print(valaz);
  distCentro = ultraCentro.Ranging(CM);
  if (distCentro > 5){
    redireccionado = 0;
    digitalWrite(rojoCentro, LOW);
    digitalWrite(azulCentro, HIGH);
    goAdelante();
  }

  if (distCentro < 5){
    digitalWrite(azulCentro, LOW);
    digitalWrite(rojoCentro, HIGH);
    if (redireccionado == 0){
      caso = comprobarbandas();
    }
  }
  switch(caso) {

```

```

case 1:
digitalWrite(rojoDerecho,LOW);
digitalWrite(azulDerecho,HIGH);
digitalWrite(rojoIzquierdo,HIGH);
digitalWrite(azulIzquierdo,LOW);
giroDerecha(90);
break;
case 2:
digitalWrite(rojoDerecho,HIGH);
digitalWrite(azulDerecho,LOW);
digitalWrite(rojoIzquierdo,LOW);
digitalWrite(azulIzquierdo,HIGH);
giroIzquierda(90);
break;
case 3:
digitalWrite(rojoDerecho,LOW);
digitalWrite(azulDerecho,HIGH);
digitalWrite(rojoIzquierdo,HIGH);
digitalWrite(azulIzquierdo,LOW);
giroDerecha(180);

break;
}
}
}
DETIENE EVASOR DE OBSTACULOS
elseif (val == '1') {
goParada();
rgbPrincipal(255,0,0);
  CRT=0;
  CRD=0;
lcd.clear();
}
}
FUNCIONES DEL PROGRAMA
void avanzar (float distancia) {
  CRT=0;
redireccionado= 0;
impulsosAdelante=0;
pulsosAdelante=(16*distancia)/0.36;
distCentro = ultraCentro.Ranging(CM);
while(CRD <= pulsosAdelante){
distCentro = ultraCentro.Ranging(CM);
lcd.setCursor(0, 0);
lcd.print(">>TRAYECTORIA 1.");

```

```

if (distCentro > 5){
digitalWrite(rojoCentro, LOW);
digitalWrite(azulCentro, HIGH);
goAdelante();
}
distCentro = ultraCentro.Ranging(CM);
if (distCentro < 5){
Serial.println('2');
digitalWrite(azulCentro, LOW);
digitalWrite(rojoCentro, HIGH);
goParada();
delay(5000);
tonoParada();
delay(5000);
Serial.println('5');
distCentro = ultraCentro.Ranging(CM);
}

}
goParada();
}

void giroIzquierda (int anguloIzquierda){
  CRD=0;
  int pulsosIzq=0;
  pulsosIzq=(10*anguloIzquierda)/90;
  while (CRD <= pulsosIzq){
    analogWrite(motorEA, velGiro);
    analogWrite(motorEB, velGiro);
    digitalWrite(motorI1,HIGH);
    digitalWrite(motorI2,LOW);
    digitalWrite(motorI3,HIGH);
    digitalWrite(motorI4,LOW);
  }
  goParada();
}

void giroDerecha(int angulo){
  CRD=0;
  int pulsosDerecha=0;
  pulsosDerecha = (10*angulo)/90;
  while (CRD <= pulsosDerecha){
    digitalWrite(motorI1,LOW);
    digitalWrite(motorI2,HIGH);
    digitalWrite(motorI3,LOW);

```



```
digitalWrite(motorI4,HIGH);
analogWrite(motorEA, velGiro);
analogWrite(motorEB, velGiro);
}
goParada();
}
```

```
voidgoAdelante(){
digitalWrite(motorI1,LOW);
digitalWrite(motorI2,HIGH);
digitalWrite(motorI3,HIGH);
digitalWrite(motorI4,LOW);
analogWrite(motorEA, velAvance);
analogWrite(motorEB, velAvance);
}
```

```
voidgoAtras(){
digitalWrite(motorI1,HIGH);
digitalWrite(motorI2,LOW);
digitalWrite(motorI3,LOW);
digitalWrite(motorI4,HIGH);
analogWrite(motorEA, velAvance);
analogWrite(motorEB, velAvance);
}
```

```
voidgoParada(){
digitalWrite(motorI1,LOW);
digitalWrite(motorI2,LOW);
digitalWrite(motorI3,LOW);
digitalWrite(motorI4,LOW);
analogWrite(motorEA, 0);
analogWrite(motorEB, 0);
}
```

```
voidgoIzquierda(){
digitalWrite(motorI1,HIGH);
digitalWrite(motorI2,LOW);
digitalWrite(motorI3,HIGH);
digitalWrite(motorI4,LOW);
analogWrite(motorEA, velGiro);
analogWrite(motorEB, velGiro);
}
```

```
voidgoDerecha(){
digitalWrite(motorI1,LOW);
```

```

digitalWrite(motorI2,HIGH);
digitalWrite(motorI3,LOW);
digitalWrite(motorI4,HIGH);
analogWrite(motorEA, velGiro);
analogWrite(motorEB, velGiro);
}

intcomprobarbandas() {
    CRD = 0;
    distDer = ultraDerecho.Ranging(CM);
    distIzq = ultraIzquierdo.Ranging(CM);
    lcd.setCursor(0, 0);
    lcd.print("BUSCAR DIRECCION");

    redireccionado = 1;
    if (distDer>distIzq) {
        giro = 1;
    }
    else if(distIzq>distDer) {
        giro = 2;
    }
    else if(distIzq = distDer) {
        giro = 3;
    }
    return giro;

}

CONTEO DEL ENCODER
voidContadorRuedaDer() {
    CRD++;
    CRT++;
    lcd.setCursor(0, 1);
    lcd.print(CRD*4.5);
    lcd.setCursor(8, 1);
    lcd.print(CRT*4.5);
}

voidencendidoLeds(){
digitalWrite(rojoDerecho, LOW);
digitalWrite(verdeDerecho, LOW);
digitalWrite(azulDerecho, LOW);
digitalWrite(rojoIzquierdo, LOW);
digitalWrite(verdeIzquierdo, LOW);
digitalWrite(azulIzquierdo, LOW);
digitalWrite(rojoCentro, LOW);
}

```

```

digitalWrite(verdeCentro, LOW);
digitalWrite(azulCentro, LOW);

digitalWrite(rojoDerecho, HIGH);
digitalWrite(rojoIzquierdo, HIGH);
digitalWrite(rojoCentro, HIGH);
delay(1000);
digitalWrite(rojoDerecho, LOW);
digitalWrite(rojoIzquierdo, LOW);
digitalWrite(rojoCentro, LOW);

digitalWrite(verdeDerecho, HIGH);
digitalWrite(verdeIzquierdo, HIGH);
digitalWrite(verdeCentro, HIGH);
delay(1000);
digitalWrite(verdeDerecho, LOW);
digitalWrite(verdeIzquierdo, LOW);
digitalWrite(verdeCentro, LOW);

digitalWrite(azulDerecho, HIGH);
digitalWrite(azulIzquierdo, HIGH);
digitalWrite(azulCentro, HIGH);
delay(1000);
digitalWrite(azulDerecho, LOW);
digitalWrite(azulIzquierdo, LOW);
digitalWrite(azulCentro, LOW);

}

voidapagarLeds(){
digitalWrite(rojoDerecho, LOW);
digitalWrite(verdeDerecho, LOW);
digitalWrite(azulDerecho, LOW);
digitalWrite(rojoIzquierdo, LOW);
digitalWrite(verdeIzquierdo, LOW);
digitalWrite(azulIzquierdo, LOW);
digitalWrite(rojoCentro, LOW);
digitalWrite(verdeCentro, LOW);
digitalWrite(azulCentro, LOW);
}

voidtonoParada(){
for (intthisNote = 0; thisNote < 3; thisNote++) {

```

```
intnoteDuration = 1000/noteDurations[thisNote];  
tone(50, melody[thisNote],noteDuration);
```

```
intpauseBetweenNotes = noteDuration * 1.30;  
delay(pauseBetweenNotes);  
noTone(50);  
}  
}  
voidtonoLlegada(){  
for (intthisNote = 0; thisNote< 8; thisNote++) {
```

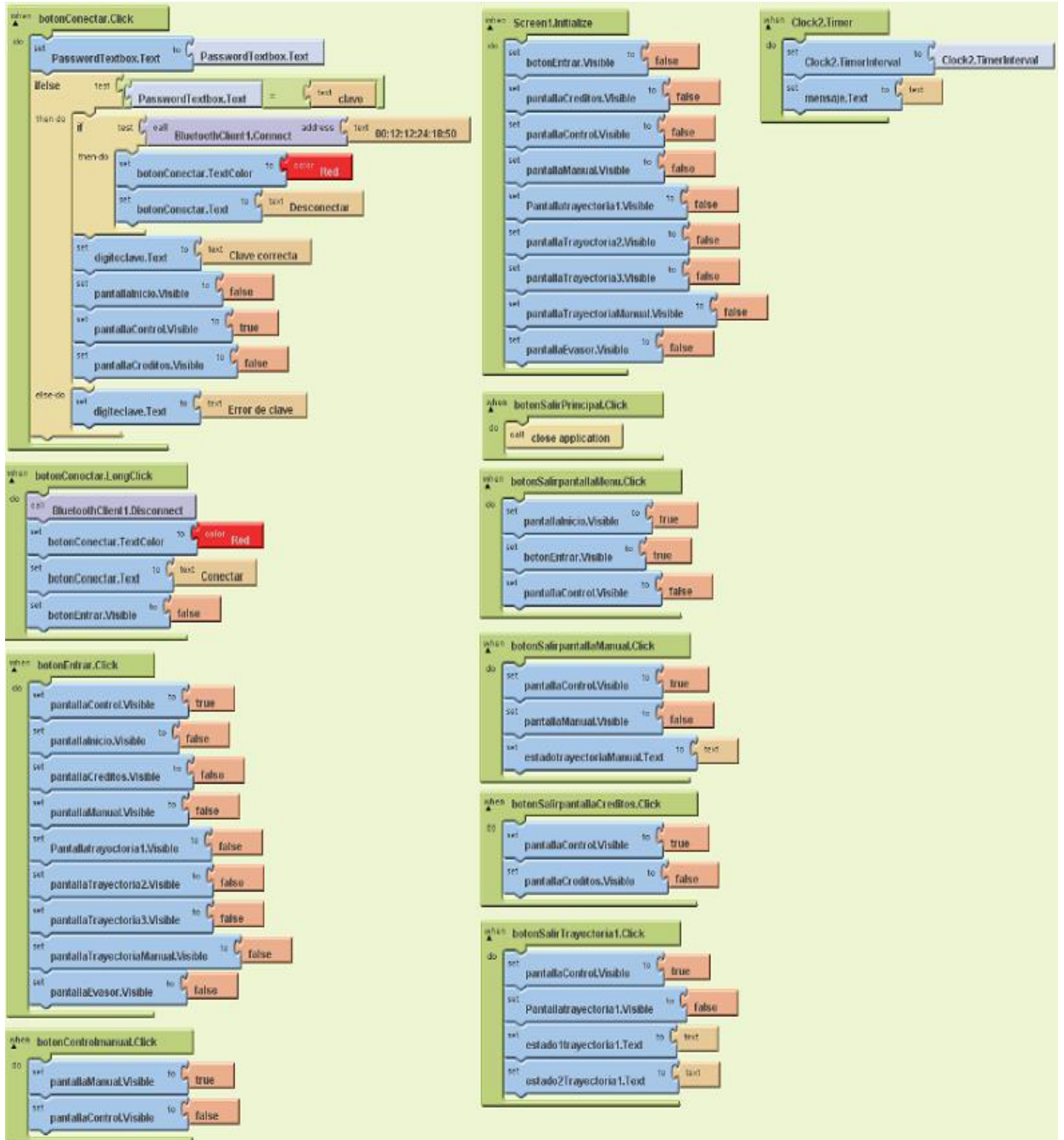
```
intnoteDuration = 1000/noteDurations1[thisNote];  
tone(50, melody[thisNote],noteDuration);
```

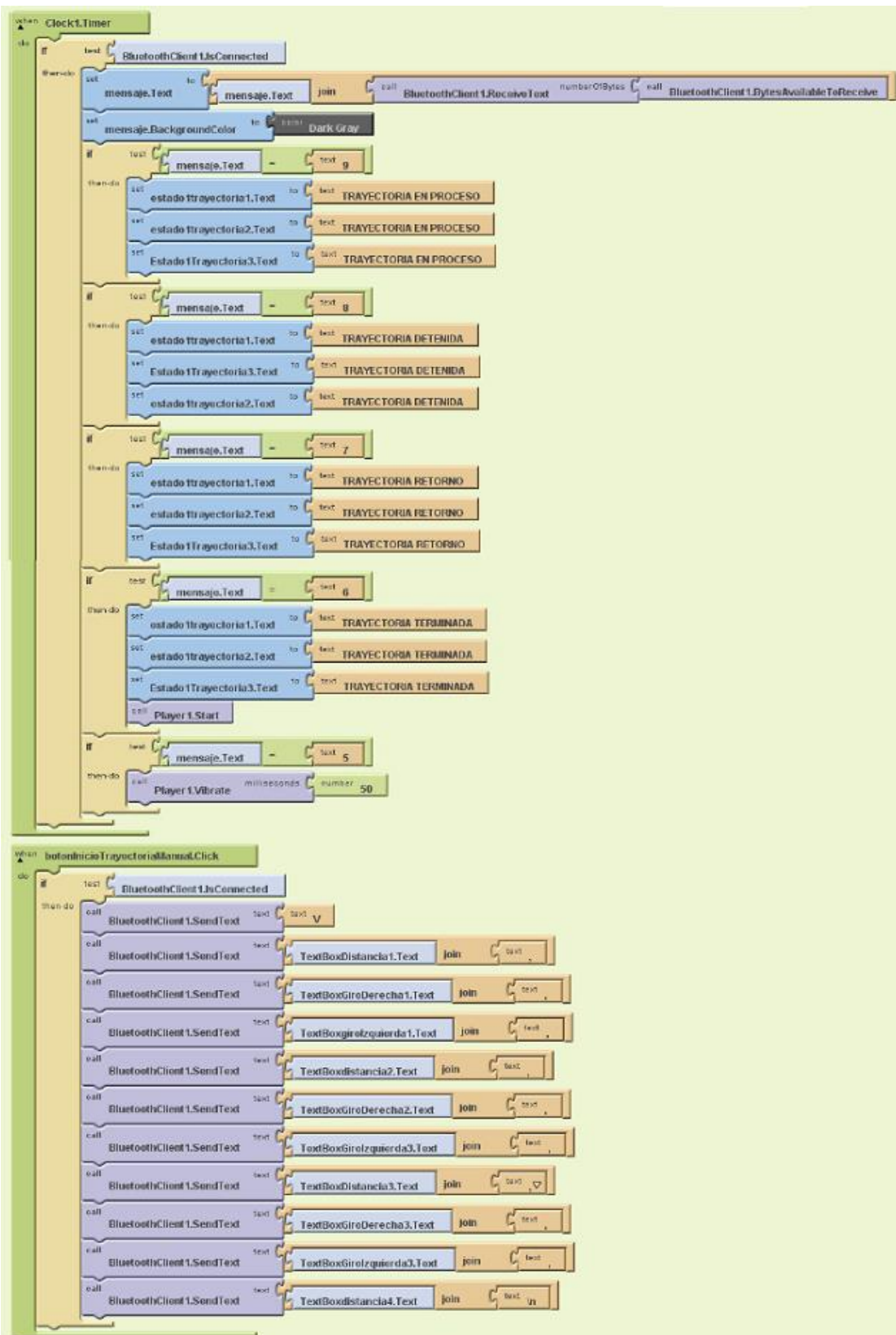
```
intpauseBetweenNotes = noteDuration * 1.30;  
delay(pauseBetweenNotes);  
noTone(50);  
}  
}
```

```
voidrgbPrincipal(int rojo, int verde, int azul){  
analogWrite(rojoPrincipal,rojo);  
analogWrite(verdePrincipal,verde);  
analogWrite(azulPrincipal,azul);  
}
```

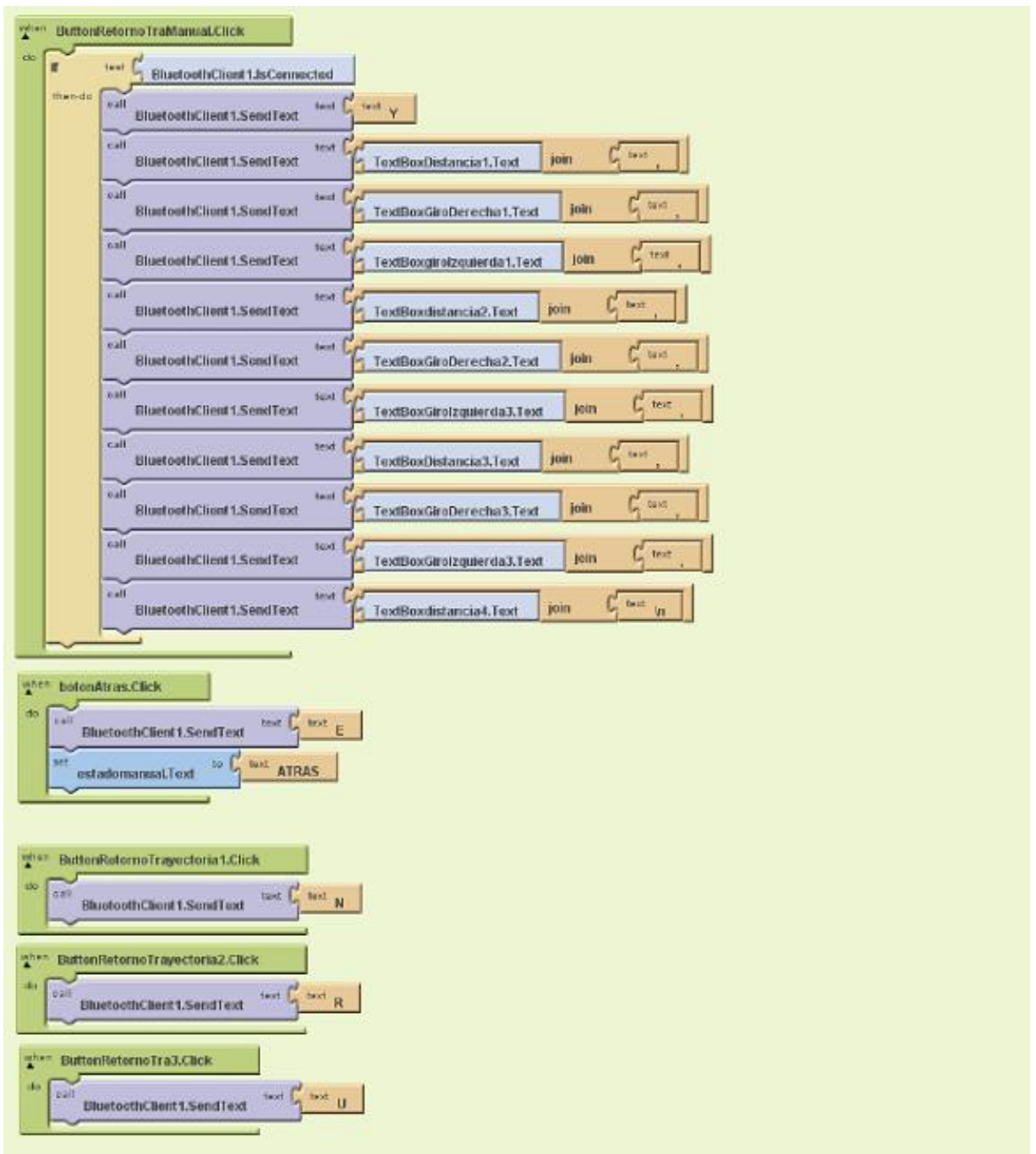
8.3 Anexo 3

Programación en APP Inventor









8.4 Anexo 4

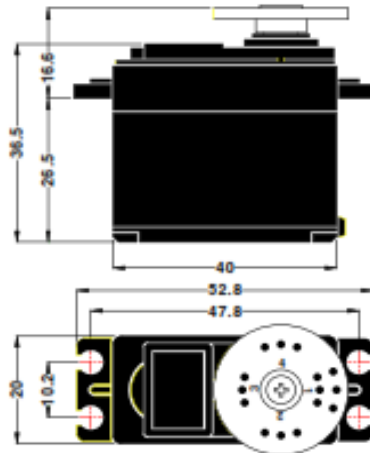
Hojas de Especificaciones

Servomotores

PREPARED BY JUN HEE, LEE
UPDATE: APR 01, 2002

ANNOUNCED SPECIFICATION OF HS-311 STANDARD SERVO

1. TECHNICAL VALUE		
CONTROL SYSTEM	: +PULSE WIDTH CONTROL 1500usec NEUTRAL	
OPERATING VOLTAGE RANGE	: 4.8V TO 6.0V	
TEST VOLTAGE	: AT 4.8V	AT 6.0V
OPERATING SPEED	: 0.19sec/60° AT NO LOAD	0.15sec/60° AT NO LOAD
STALL TORQUE	: 3.0kg.cm(42oz.in)	3.5kg.cm(48.60oz.in)
IDLE CURRENT	: 7.4mA AT STOPPED	7.7mA AT STOPPED
RUNNING CURRENT	: 160mA/60° AT NO LOAD	180mA/60° AT NO LOAD
STALL CURRENT	: 700mA	800mA
DEAD BAND WIDTH	: 3usec	3usec
OPERATING TRAVEL	: 40°/ONE SIDE PULSE TRAVELING 400usec	
DIRECTION	: CLOCK WISE/PULSE TRAVELING 1500 TO 1900usec	
MOTOR TYPE	: CORED METAL BRUSH	
POTENTIOMETER TYPE	: 4 SLIDER/DIRECT DRIVE	
AMPLIFIER TYPE	: ANALOG CONTROLLER & TRANSISTOR DRIVER	
DIMENSIONS	: 40x20x36.5mm(1.57x0.78x1.43in)	
WEIGHT	: 43g(1.51oz)	
BALL BEARING	: TOP/RESIN BUSHING	
GEAR MATERIAL	: RESIN	
HORN GEAR SPLINE	: 24 SEGMENTS/□5.76	
SPLINED HORNS	: SUPER/R-XA	
CONNECTOR WIRE LENGTH	: 300mm(11.81in)	
CONNECTOR WIRE STRAND COUNTER	: 40EA	
CONNECTOR WIRE GAUGE		



2. FEATURES

LONG LIFE POTENTIOMETER, TOP RESIN BUSHING

3. APPLICATIONS

AIRCRAFT 20-40 SIZE, STEERING AND THROTTLE SERVO FOR CARS, TRUCK AND BOATS

4. ACCESSORY & OPTION

CASE SET/ HS322T:1EA HS322M:1EA HS322L:1EA PH/T-2 2x30 Ni:4EA	GEAR SET/ HS322G1:1EA HS322G2:1EA HS322G3:1EA HS322G4:1EA HS300RB:1EA	HORN SET/ R-XA:1EA
---	--	-----------------------

HITEC RCD KOREA INC.

Fototransistor

GP1S53VJ000F

Gap : 5mm, Slit : 0.5mm
Phototransistor Output,
Case package Transmissive
Photointerrupter



■ Description

GP1S53VJ000F is a standard, phototransistor output, transmissive photointerrupter with opposing emitter and detector in a case, providing non-contact sensing. For this family of devices, the emitter and detector are inserted in a case, resulting in a through-hole design.

■ Features

1. Transmissive with phototransistor output
2. Highlights :
 - Vertical Slit for alternate motion detection
3. Key Parameters :
 - Gap Width : 5mm
 - Slit Width (detector side) : 0.5mm
 - Package : 13.7×10×5.2mm
4. RoHS directive compliant

■ Agency approvals/Compliance

1. Compliant with RoHS directive

■ Applications

1. General purpose detection of object presence or motion.
2. Example : Printer, FAX, Optical storage unit

■ Absolute Maximum Ratings

(T_a=25°C)

Parameter		Symbol	Rating	Unit
Input	^{*1} Forward current	I _F	50	mA
	^{*1,2} Peak forward current	I _{FM}	1	A
	Reverse voltage	V _R	6	V
	Power dissipation	P	75	mW
Output	Collector-emitter voltage	V _{CEO}	35	V
	Emitter-collector voltage	V _{ECO}	6	V
	Collector current	I _C	20	mA
	^{*1} Collector power dissipation	P _C	75	mW
Operating temperature		T _{opr}	-25 to +85	°C
Storage temperature		T _{stg}	-40 to +100	°C
^{*3} Soldering temperature		T _{sol}	260	°C

^{*1} Refer to Fig. 1, 2, 3

^{*2} Pulse width ≤ 100μs, Duty ratio=0.01

^{*3} For Sn or less

■ Electro-optical Characteristics

(T_a=25°C)

Parameter		Symbol	Condition	MIN.	TYP.	MAX.	Unit
Input	Forward voltage	V _F	I _F =20mA	-	1.25	1.4	V
	Peak forward voltage	V _{FM}	I _{FM} =0.5A	-	3	4	V
	Reverse current	I _R	V _R =3V	-	-	10	μA
Output	Collector dark current	I _{CEO}	V _{CE} =20V	-	1	100	nA
Transfer characteristics	Collector current	I _C	V _{CE} =5V, I _F =20mA	0.5	-	15	mA
	Collector-emitter saturation voltage	V _{CE(sat)}	I _F =40mA, I _C =0.2mA	-	-	0.4	V
	Response time	Rise time	t _r	V _{CE} =2V, I _C =2mA, R _L =100Ω	-	3	15
Fall time		t _f	-		4	20	

Puente H



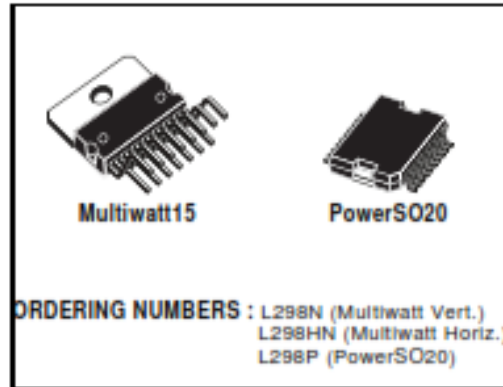
L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

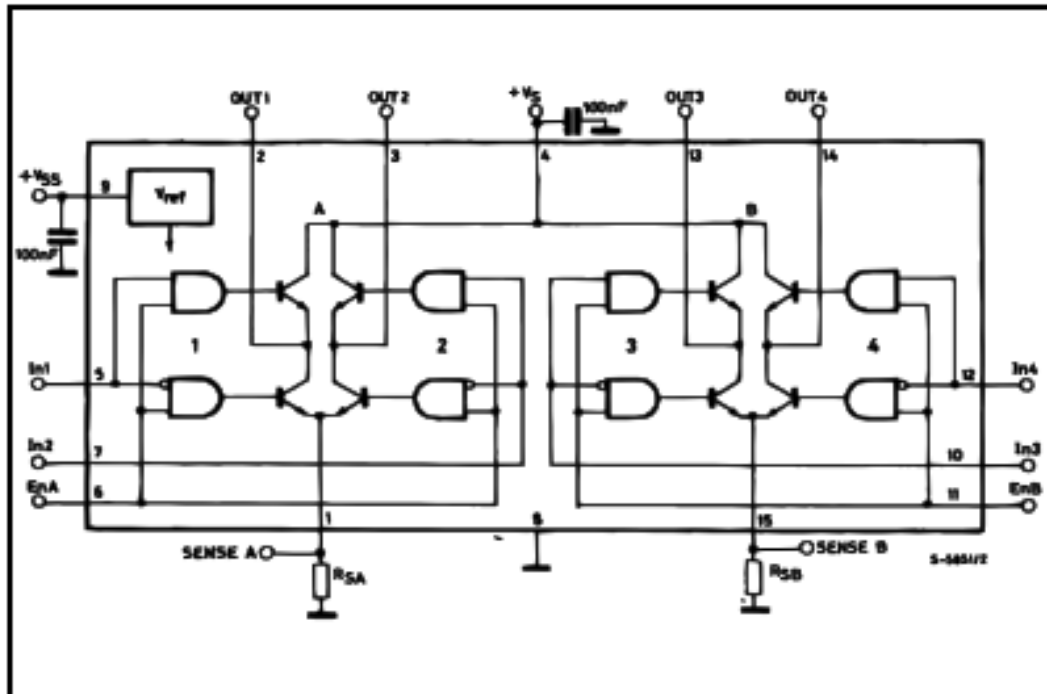
DESCRIPTION

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



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

BLOCK DIAGRAM

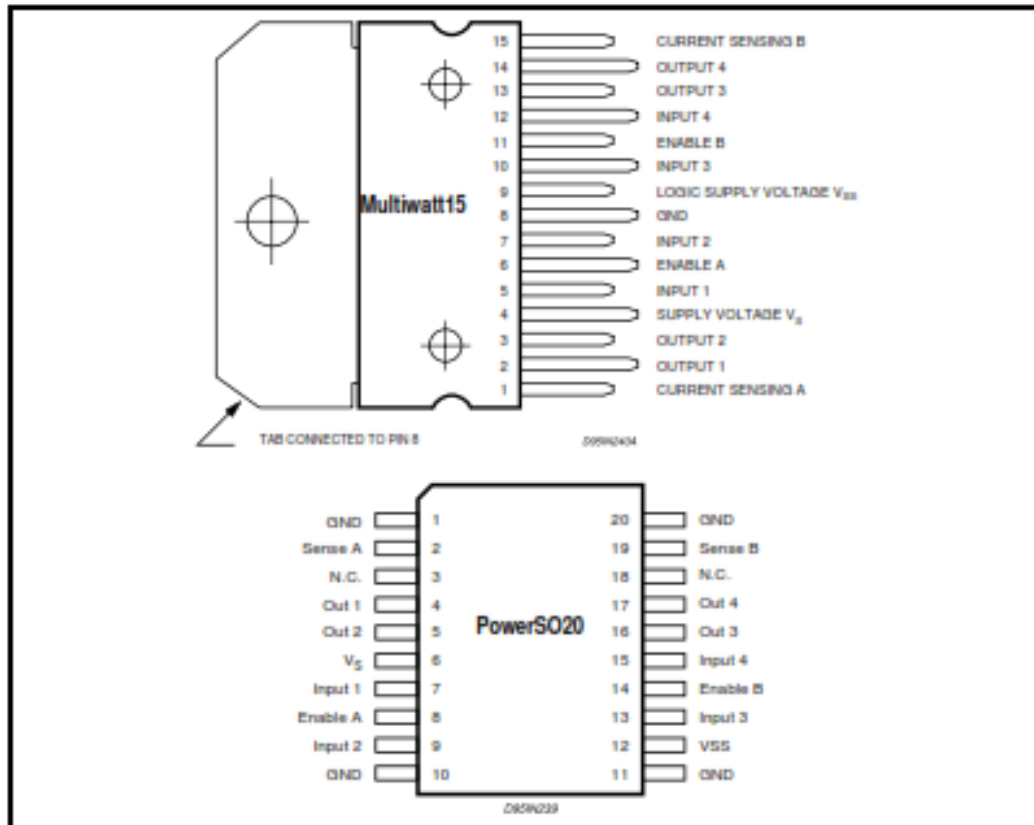


L298

ABSOLUTE MAXIMUM RATINGS

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

PIN CONNECTIONS (top view)



THERMAL DATA

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

PIN FUNCTIONS (refer to the block diagram)

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

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

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_S	Supply Voltage (pin 4)	Operative Condition	$V_{IH} + 2.5$		46	V
V_{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I_S	Quiescent Supply Current (pin 4)	$V_{en} = H$; $I_L = 0$ $V_I = L$ $V_I = H$		13 50	22 70	mA mA
I_{SS}	Quiescent Current from V_{SS} (pin 9)	$V_{en} = L$ $V_{en} = H$; $I_L = 0$ $V_I = L$ $V_I = H$ $V_{en} = L$ $V_I = X$			4 36 12 6	mA mA mA mA
V_L	Input Low Voltage (pins 5, 7, 10, 12)		-0.3		1.5	V
V_{IH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		V_{SS}	V
I_L	Low Voltage Input Current (pins 5, 7, 10, 12)	$V_I = L$			-10	μA
I_H	High Voltage Input Current (pins 5, 7, 10, 12)	$V_I = H \leq V_{SS} - 0.6V$		30	100	μA
$V_{en} = L$	Enable Low Voltage (pins 6, 11)		-0.3		1.5	V
$V_{en} = H$	Enable High Voltage (pins 6, 11)		2.3		V_{SS}	V
$I_{en} = L$	Low Voltage Enable Current (pins 6, 11)	$V_{en} = L$			-10	μA
$I_{en} = H$	High Voltage Enable Current (pins 6, 11)	$V_{en} = H \leq V_{SS} - 0.6V$		30	100	μA
$V_{CEsat(H)}$	Source Saturation Voltage	$I_L = 1A$ $I_L = 2A$	0.95	1.35 2	1.7 2.7	V V
$V_{CEsat(L)}$	Sink Saturation Voltage	$I_L = 1A$ (5) $I_L = 2A$ (5)	0.85	1.2 1.7	1.6 2.3	V V
V_{CEsat}	Total Drop	$I_L = 1A$ (5) $I_L = 2A$ (5)	1.80		3.2 4.9	V V
V_{sense}	Sensing Voltage (pins 1, 15)		-1 (1)		2	V

Regulador de Voltaje



**L7800
SERIES**

POSITIVE VOLTAGE REGULATORS

- OUTPUT CURRENT TO 1.5A
- OUTPUT VOLTAGES OF 5; 5.2; 6; 8; 8.5; 9; 10; 12; 15; 18; 24V
- THERMAL OVERLOAD PROTECTION
- SHORT CIRCUIT PROTECTION
- OUTPUT TRANSITION SOA PROTECTION

DESCRIPTION

The L7800 series of three-terminal positive regulators is available in TO-220, TO-220FP, TO-220FM, TO-3 and D²PAK packages and several fixed output voltages, making it useful in a wide range of applications. These regulators can provide local on-card regulation, eliminating the distribution problems associated with single point regulation. Each type employs internal current limiting, thermal shut-down and safe area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltage and currents.

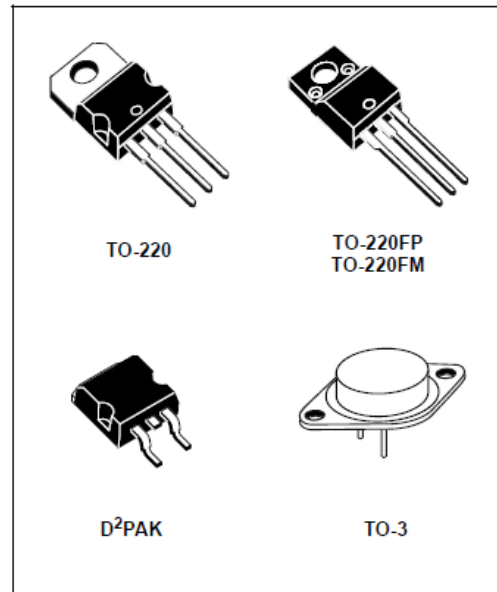
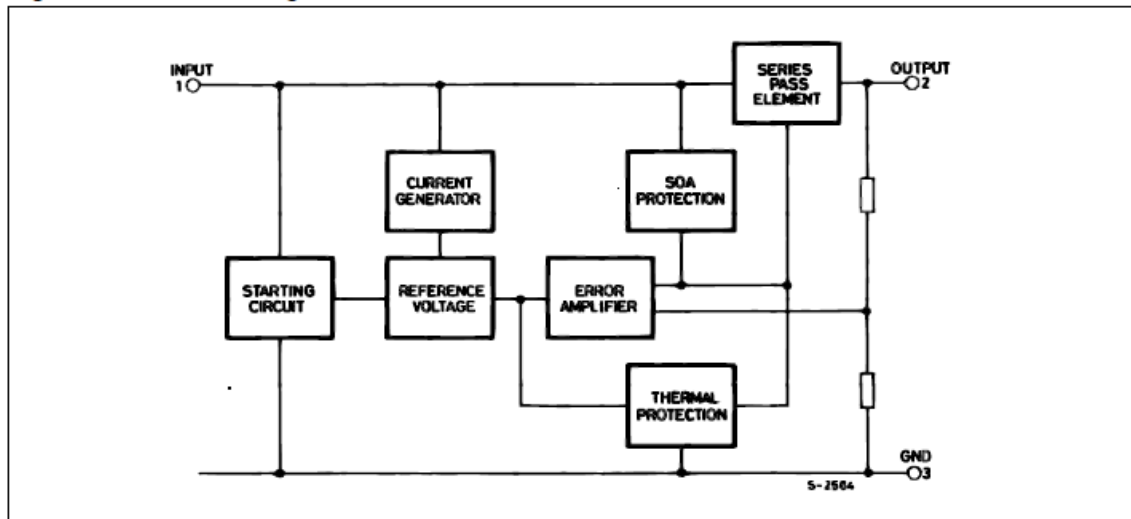


Figure 1: Schematic Diagram



L7800 SERIES

Table 1: Absolute Maximum Ratings

Symbol	Parameter		Value	Unit
V_I	DC Input Voltage	for $V_O = 5$ to 18V	35	V
		for $V_O = 20, 24V$	40	
I_O	Output Current		Internally Limited	
P_{tot}	Power Dissipation		Internally Limited	
T_{stg}	Storage Temperature Range		-65 to 150	°C
T_{op}	Operating Junction Temperature Range	for L7800	-55 to 150	°C
		for L7800C	0 to 150	

Absolute Maximum Ratings are those values beyond which damage to the device may occur. Functional operation under these condition is not implied.

Table 2: Thermal Data

Symbol	Parameter	D ² PAK	TO-220	TO-220FP	TO-220FM	TO-3	Unit
$R_{thj-case}$	Thermal Resistance Junction-case Max	3	5	5	5	4	°C/W
$R_{thj-amb}$	Thermal Resistance Junction-ambient Max	62.5	50	60	60	35	°C/W

Figure 2: Schematic Diagram

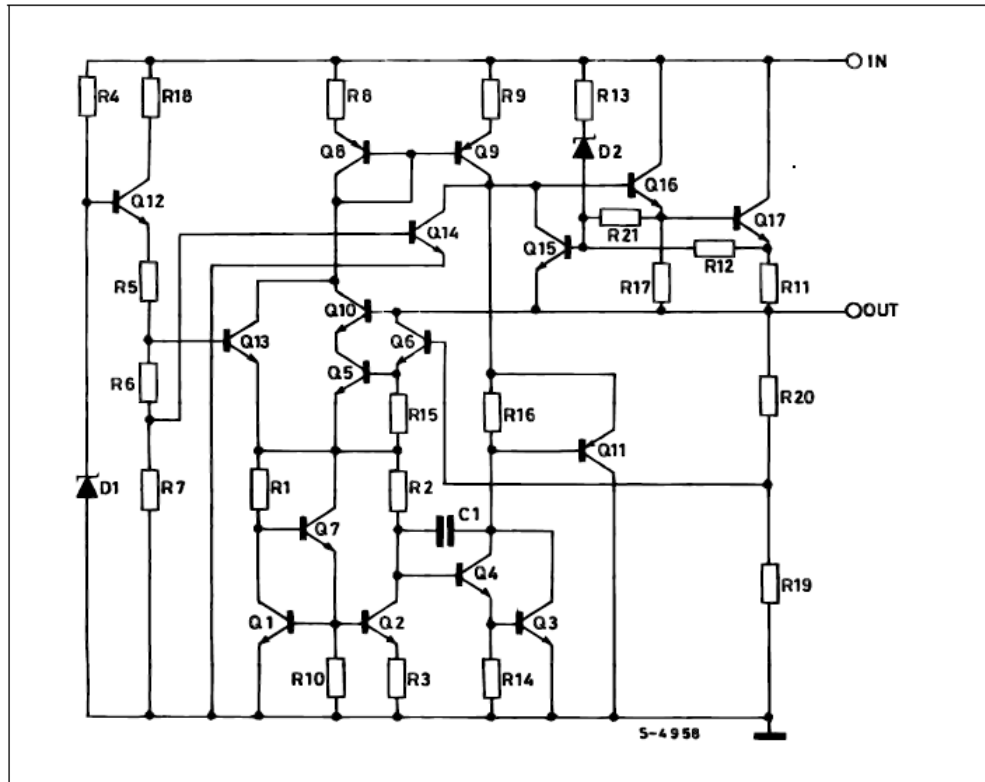
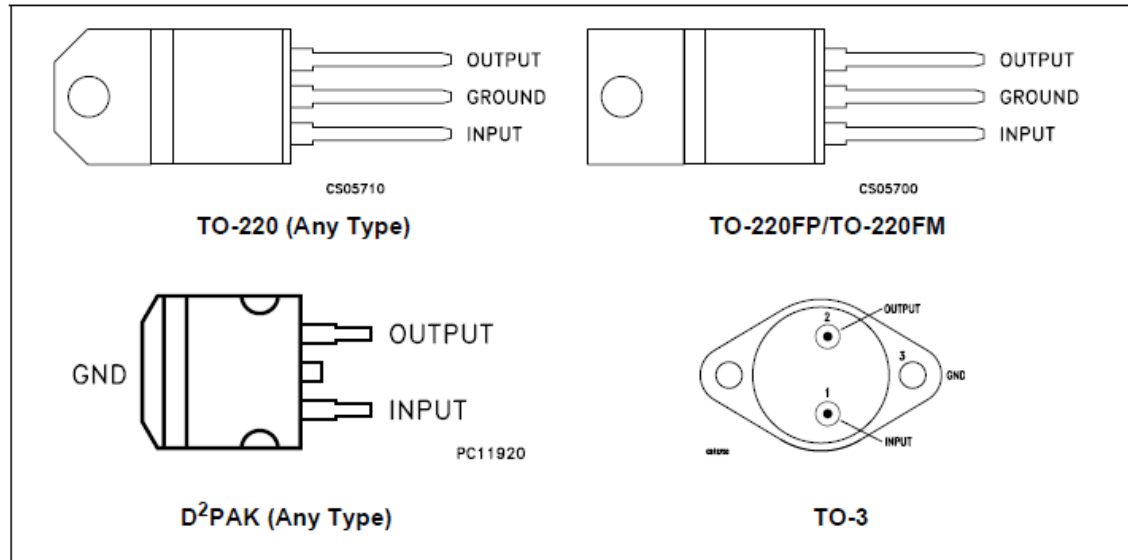


Figure 3: Connection Diagram (top view)



Sensor de Ultrasonido

HC-SR04

Ultrasonic Introduction

Ultrasonic Definition

The human ear can hear sound frequency around 20HZ ~ 20KHZ, and ultrasonic is the sound wave beyond the human ability of 20KHZ .

Ultrasonic distance measurement principle

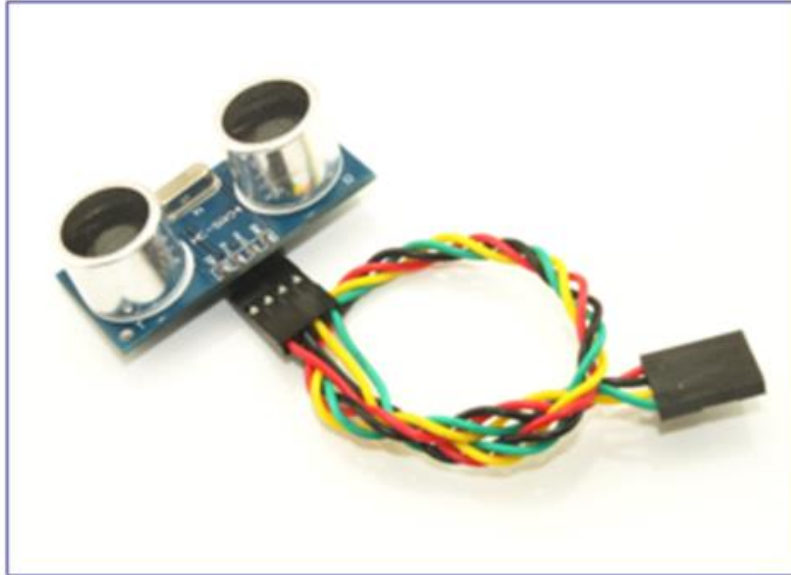
Ultrasonic transmitter emitted an ultrasonic wave in one direction, and started timing when it launched. Ultrasonic spread in the air, and would return immediately when it encountered obstacles on the way. At last, the ultrasonic receiver would stop timing when it received the reflected wave. As Ultrasonic spread velocity is 340m / s in the air, based on the timer record t , we can calculate the distance (s) between the obstacle and transmitter, namely: $s = 340t / 2$, which is so-called time difference distance measurement principle

The principle of ultrasonic distance measurement used the already-known air spreading velocity, measuring the time from launch to reflection when it encountered obstacle, and then calculate the distance between the transmitter and the obstacle according to the time and the velocity. Thus, the principle of ultrasonic distance measurement is the same with radar.

Distance Measurement formula is expressed as: $L = C \times T$

In the formula, L is the measured distance, and C is the ultrasonic spreading velocity in air, also, T represents time (T is half the time value from transmitting to receiving).

Product Image



Module pin definitions

Types	Pin Symbol	Pin Function Description
HC-SR04	VCC	5V power supply
	Trig	Trigger pin
	Echo	Receive pin
	GND	Power ground

Electrical parameters

Electrical Parameters	HC-SR04 Ultrasonic Module
Operating Voltage	DC-5V
Operating Current	15mA
Operating Frequency	40KHZ
Farthest Range	4m
Nearest Range	2cm
Measuring Angle	15 Degree
Input Trigger Signal	10us TTL pulse
Output Echo Signal	Output TTL level signal, proportional with range
Dimensions	45*20*15mm

Microcontrolador ATmega 2560

Features

- High Performance, Low Power Atmel® AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
 - 135 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16MHz
 - On-Chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 64K/128K/256KBytes of In-System Self-Programmable Flash
 - 4Kbytes EEPROM
 - 8Kbytes Internal SRAM
 - Write/Erase Cycles:10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85° C/ 100 years at 25° C
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
 - Endurance: Up to 64Kbytes Optional External Memory Space
- Atmel® QTouch® library support
 - Capacitive touch buttons, sliders and wheels
 - QTouch and QMatrix® acquisition
 - Up to 64 sense channels
- 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 Prescaler and Compare Mode
 - Four 16-bit Timer/Counter with Separate Prescaler, Compare- and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Four 8-bit PWM Channels
 - Six/Twelve PWM Channels with Programmable Resolution from 2 to 16 Bits (ATmega1281/2561, ATmega640/1280/2560)
 - Output Compare Modulator
 - 8/16-channel, 10-bit ADC (ATmega1281/2561, ATmega640/1280/2560)
 - Two/Four Programmable Serial USART (ATmega1281/2561, ATmega640/1280/2560)
 - Master/Slave SPI Serial Interface
 - Byte Oriented 2-wire 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 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
 - 54/86 Programmable I/O Lines (ATmega1281/2561, ATmega640/1280/2560)
 - 64-pad QFN/MLF, 64-lead TQFP (ATmega1281/2561)
 - 100-lead TQFP, 100-ball CBGA (ATmega640/1280/2560)
 - RoHS/Fully Green
- Temperature Range
 - -40° C to 85° C Industrial
- Ultra-Low Power Consumption
 - Active Mode: 1MHz, 1.8V: 500µA
 - Power-down Mode: 0.1µA at 1.8V
- Speed Grade:
 - ATmega640V/ATmega1280V/ATmega1281V:
 - 0 - 4MHz @ 1.8V - 5.5V, 0 - 8MHz @ 2.7V - 5.5V
 - ATmega2560V/ATmega2561V:
 - 0 - 2MHz @ 1.8V - 5.5V, 0 - 8MHz @ 2.7V - 5.5V
 - ATmega640/ATmega1280/ATmega1281:
 - 0 - 8MHz @ 2.7V - 5.5V, 0 - 16MHz @ 4.5V - 5.5V
 - ATmega2560/ATmega2561:
 - 0 - 16MHz @ 4.5V - 5.5V



**8-bit Atmel
Microcontroller
with
64K/128K/256K
Bytes In-System
Programmable
Flash**

**ATmega640/V
ATmega1280/V
ATmega1281/V
ATmega2560/V
ATmega2561/V**

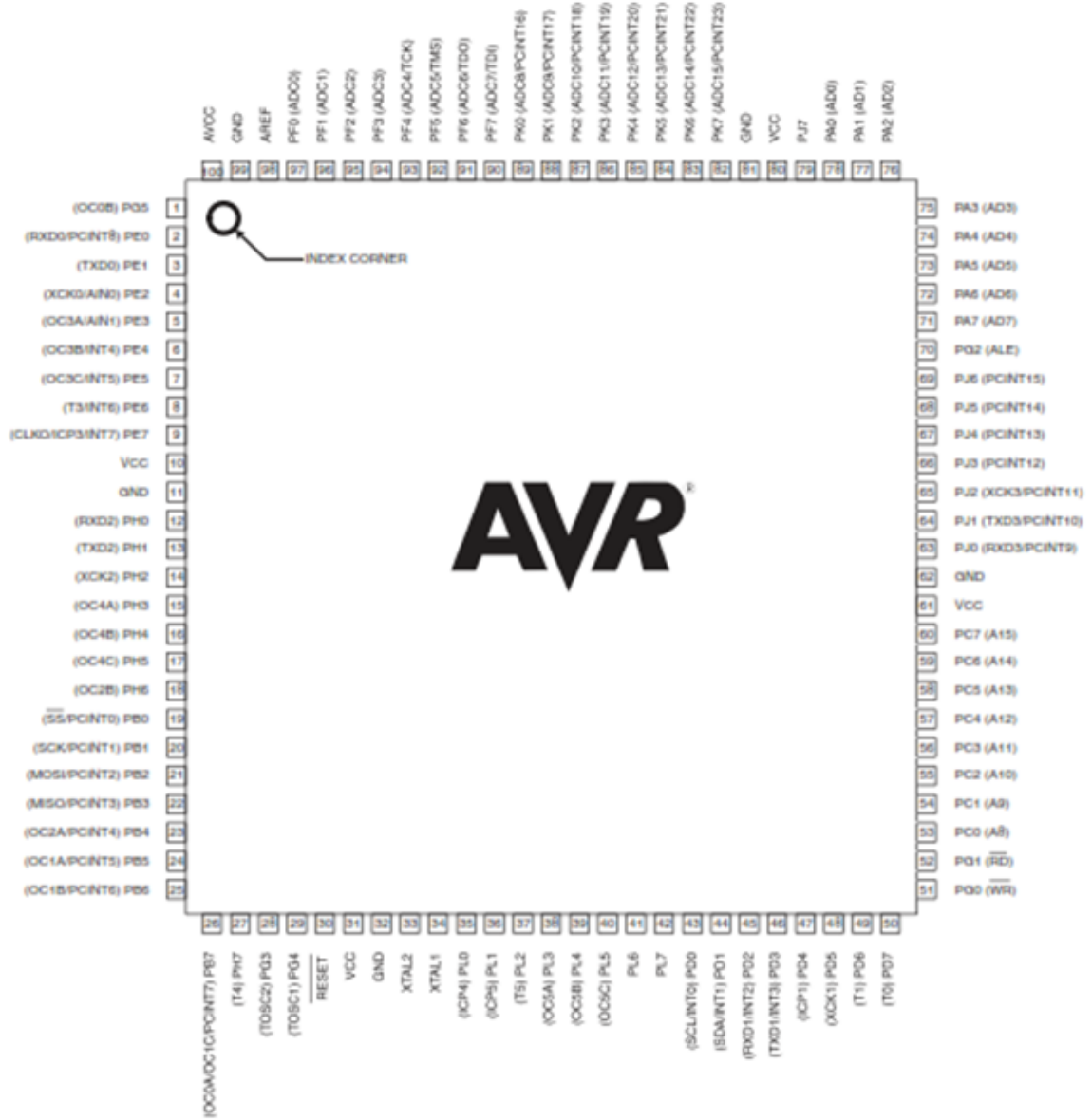
**Preliminary
Summary**

2549NS-AVR-05/11



1. Pin Configurations

Figure 1-1. TQFP-pinout ATmega640/1280/2560

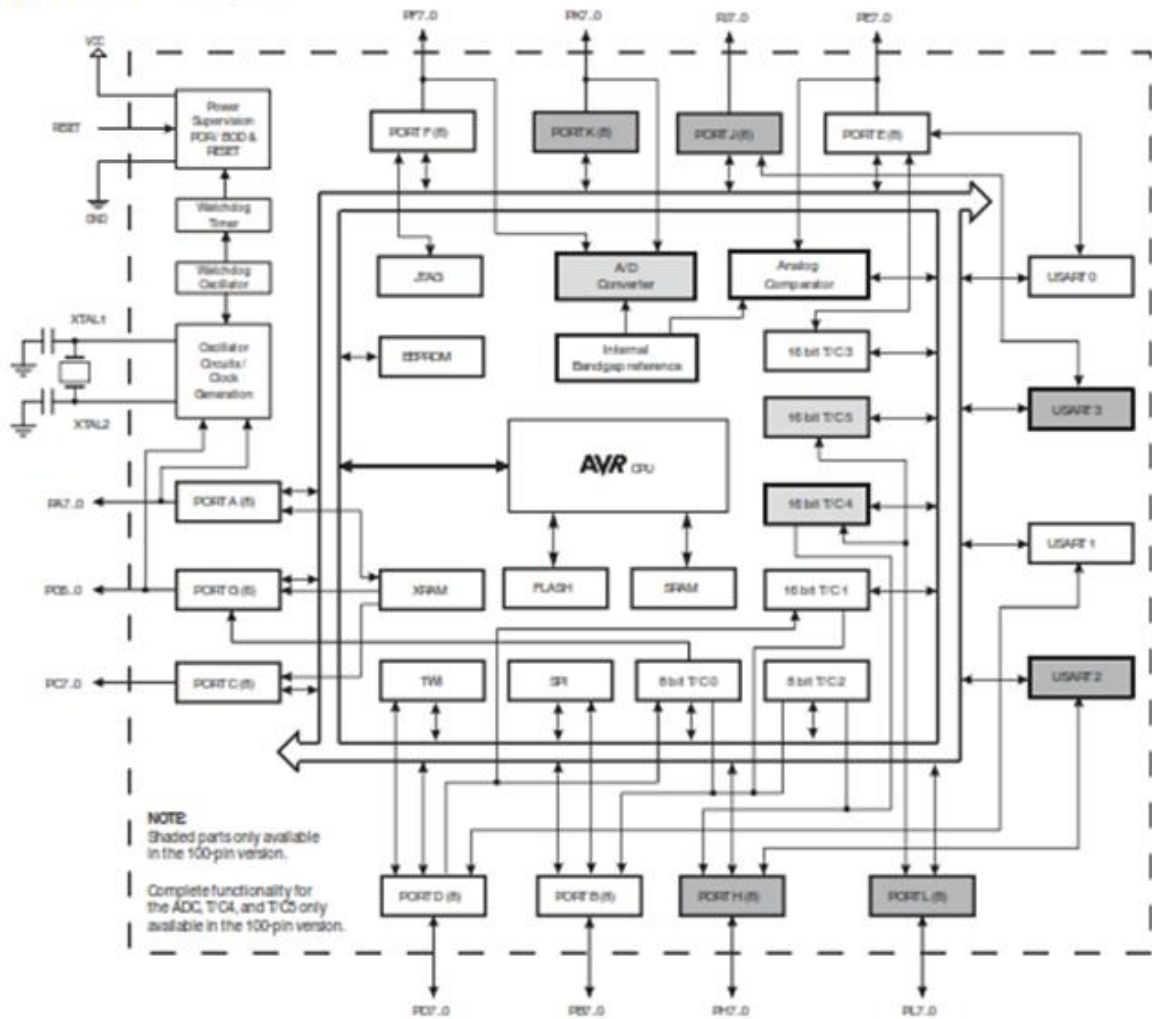


2. Overview

The ATmega640/1280/1281/2560/2561 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega640/1280/1281/2560/2561 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

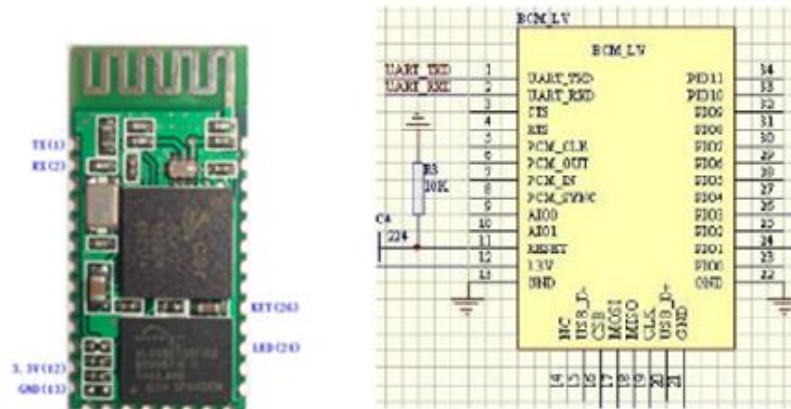
Figure 2-1. Block Diagram



Bluetooth

Product Data Sheet

HC-06 is the higher version of LV_BC_2.0. Linvor is the former of wavesen.



- Application fields:
 - Bluetooth Car Handsfree Device
 - Bluetooth GPS
 - Bluetooth PCMCIA , USB Dongle
 - Bluetooth Data Transfer
- Software
 - CSR

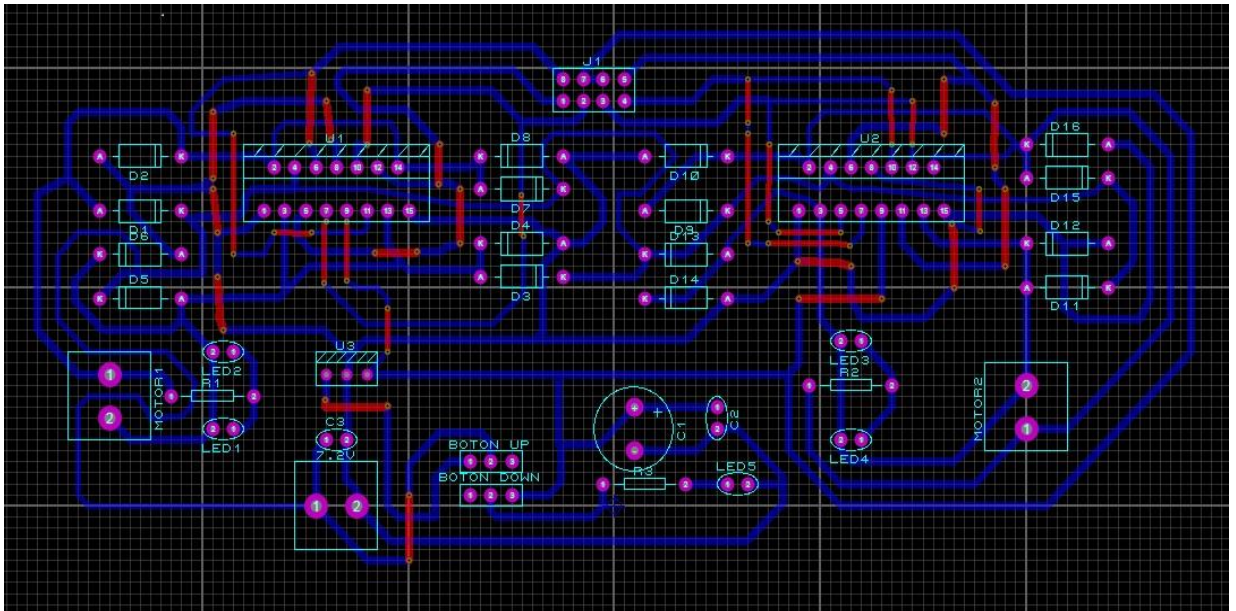
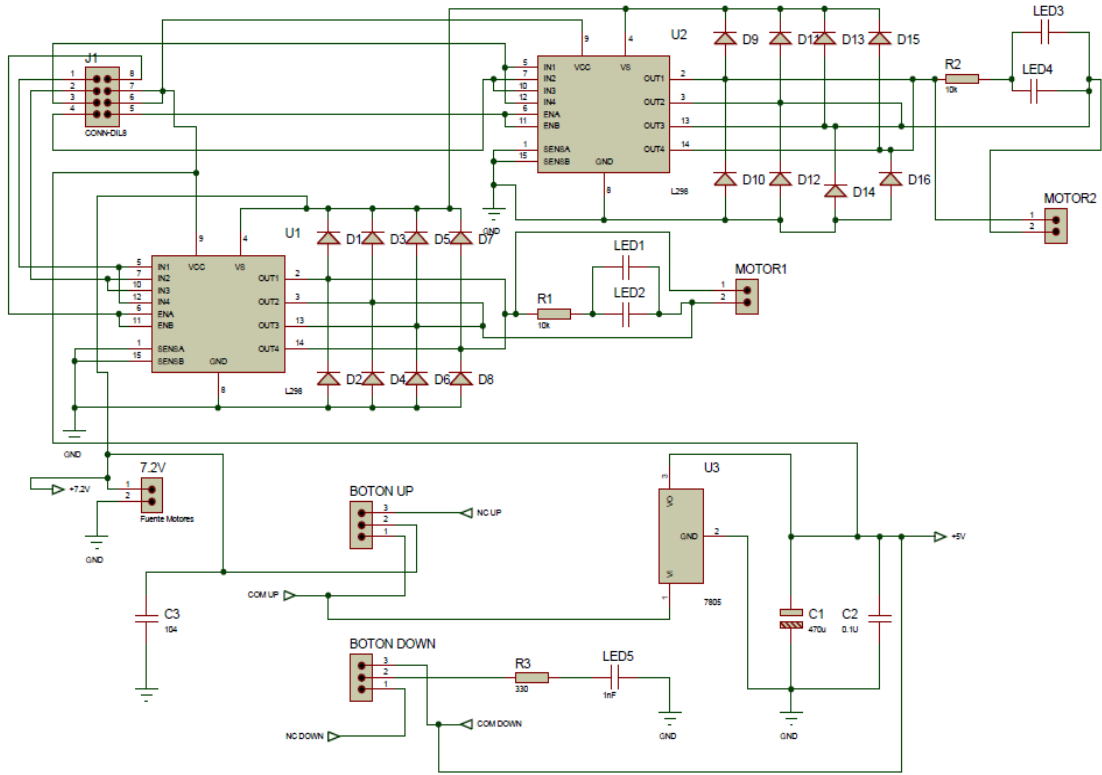
2. Feature

- Wireless transceiver
 - Sensitivity (Bit error rate) can reach -80dBm.
 - The change range of output's power: -4 - +6dBm.
- Function description (perfect Bluetooth solution)
 - Has an EDR module; and the change range of modulation depth: 2Mbps - 3Mbps.
 - Has a build-in 2.4GHz antenna; user needn't test antenna.
 - Has the external 8Mbit FLASH
 - Can work at the low voltage (3.1V~4.2V). The current in pairing is in the range of 30~40mA. The current in communication is 8mA.
 - Standard HCI Port (UART or USB)
 - USB Protocol: Full Speed USB1.1, Compliant With 2.0
 - This module can be used in the SMD.
 - It's made through RoHS process.
 - The board PIN is half hole size.
 - Has a 2.4GHz digital wireless transceiver.
 - Bases at CSR BC04 Bluetooth technology.
 - Has the function of adaptive frequency hopping.
 - Small (27mm×13mm×2mm)
 - Peripherals circuit is simple.
 - It's at the Bluetooth class 2 power level.
 - Storage temperature range: -40 ℃ - 85 ℃, work temperature range: -25 ℃ - +75 ℃
 - Any wave inter Interference: 2.4MHz, the power of emitting: 3 dBm.
 - Bit error rate: 0. Only the signal decays at the transmission link, bit error may be produced. For example, when RS232 or TTL is being processed, some signals may decay.
- Low power consumption
- Has high-performance wireless transceiver system
- Low Cost

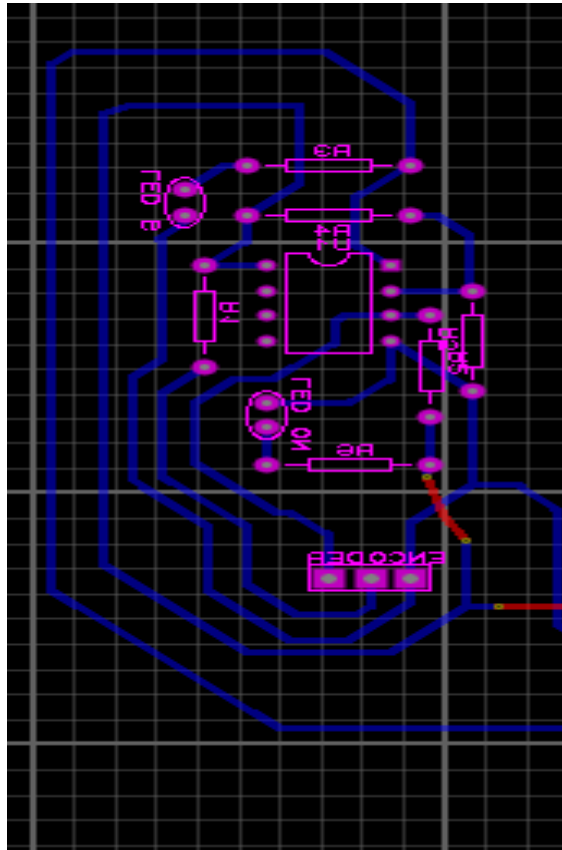
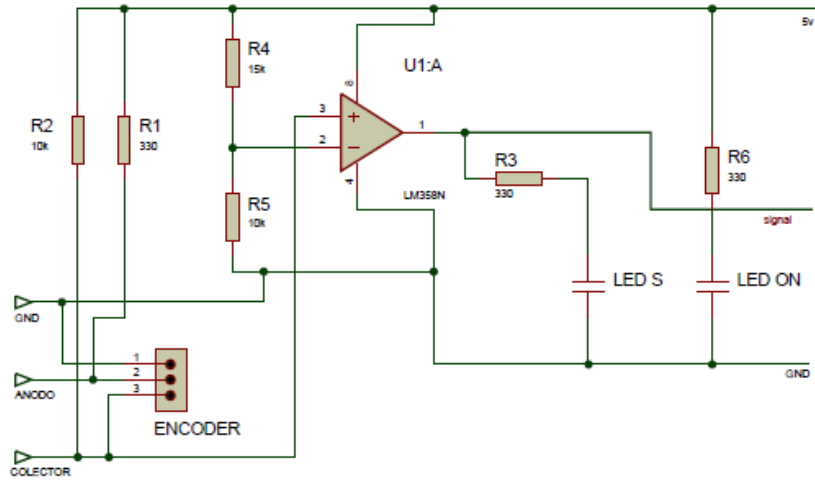
8.5 Anexo 5

Diseño de placas

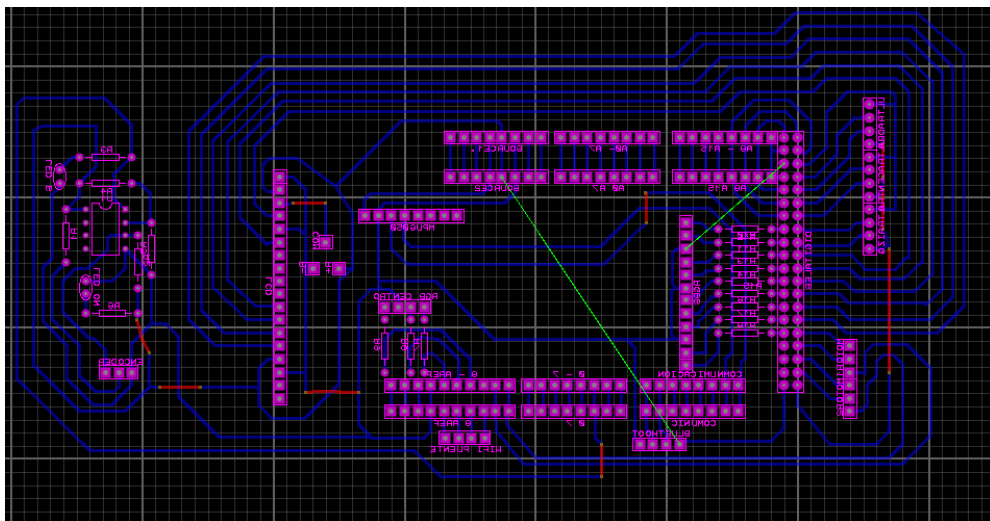
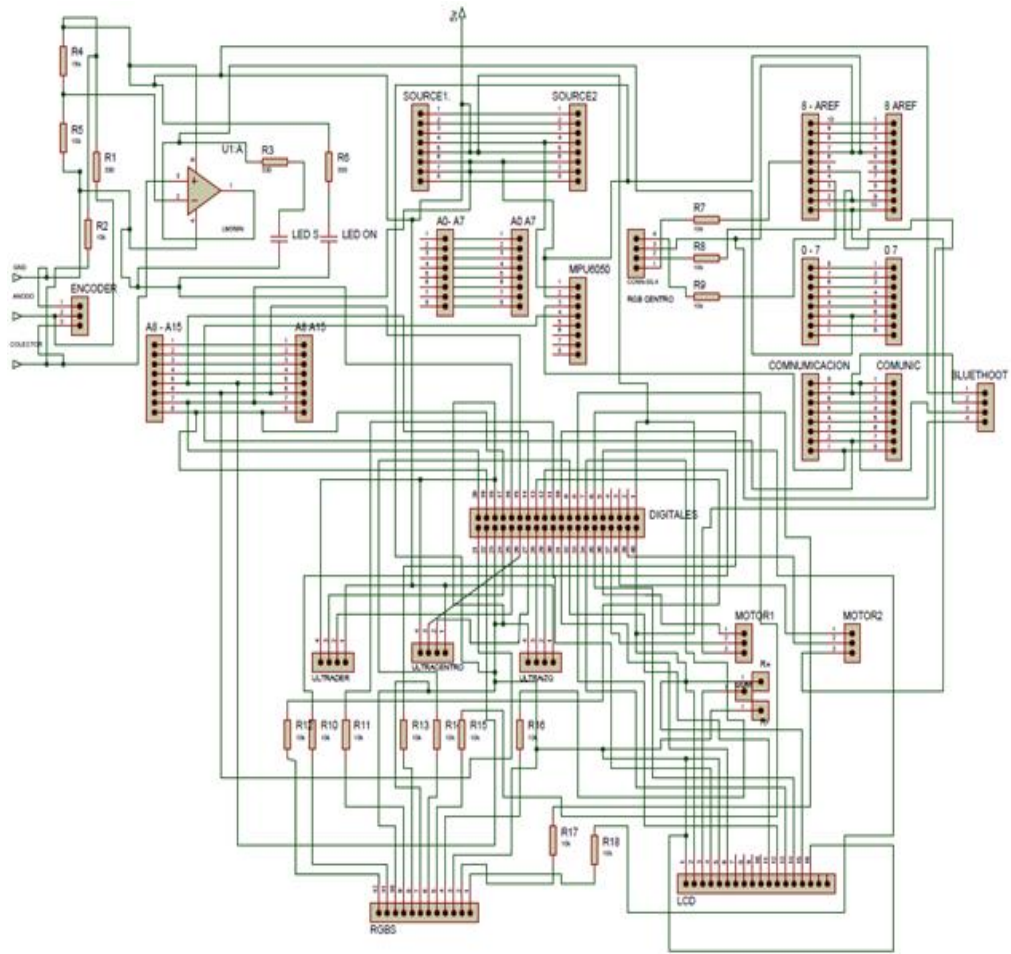
Puente H en Paralelo



Encoder



Placa de Control



8.6 Anexo 6

Participación del Robot Mensajero en la feria de la UNACH por Fiestas de la Facultad de Ingeniería



8.7 Anexo 7

Pruebas en el escenario

Trayectoria 1



Trayectoria 2



Trayectoria 3



