DISEÑO Y CONSTRUCCIÓN DE UN ANILLO DE SENSORES INFRARROJOS CON DISPOSITIVOS DE LÓGICA PROGRAMABLE (FPGAS) PARA EL LEVANTAMIENTO DE MAPAS DE ENTORNO Y NAVEGACIÓN DE PLATAFORMAS MÓVILES.

ALEJANDRO MONTOYA OSORIO

JHONNY ALEJANDRO VALENCIA V.

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, FÍSICA Y SISTEMAS
PROGRAMA EN INGENIERÍA ELÉCTRICA
PEREIRA
2008

DISEÑO Y CONSTRUCCIÓN DE UN ANILLO DE SENSORES INFRARROJOS CON DISPOSITIVOS DE LÓGICA PROGRAMABLE (FPGAS) PARA EL LEVANTAMIENTO DE MAPAS DE ENTORNO Y NAVEGACIÓN DE PLATAFORMAS MÓVILES

ALEJANDRO MONTOYA

JHONNY ALEJANDRO VALENCIA V.

Proyecto de Grado para optar al título de
Ingeniero Electricista.

Asesor
M.Sc. LUÍS HERNANDO RÍOS G.

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, FÍSICA Y SISTEMAS
PROGRAMA EN INGENIERÍA ELÉCTRICA
PEREIRA
2008

NOTA DE ACEPTACIÓN:

_____

_____

_____

_____

Firma del Presidente del Jurado

_____

Firma del Jurado

_____

Firma del Jurado

Pereira, __ __ de _____

DEDICATORIA.

Este trabajo se lo dedico a mi mamá y a mi hermanita quienes a pesar de mis errores y fallas como persona…….creyeron en mi.

Jhonny Alejandro.

Este trabajo se lo dedicado a Dios primeramente por darme la sabiduría y el entendimiento y a mi familia por darme el apoyo físico y moral en todas la situaciones de la vida.

Alejandro Montoya

AGRADECIMIENTOS

A la Vicerrectoría de Investigación, Innovación y Extensión en cabeza de su Vicerrector Samuel Ospina y el Doctor Gustavo López Ortiz por el apoyo para la ejecución de este proyecto.

Al Programa de Ingeniería Eléctrica por la prestación de los equipos del almacén de eléctrica necesarios para la ejecución del proyecto y el recurso humano que prestó en nombramiento de Mario Alberto Gómez y Francisco Javier Rivera.

Al Ingeniero Luís Hernando Ríos González profesor titular del Programa de Ingeniería Eléctrica de la Universidad Tecnológica de Pereira por el valioso aporte de conocimientos, y la confianza que depósito en nosotros para la ejecución del proyecto.

A nuestras familias y amigos que nos apoyaron de todas las formas posibles para la realización de este proyecto.

TABLA DE CONTENIDO

PRELIMINARES **14**

P.1. INTRODUCCIÓN 14
P.2. ESTADO DEL ARTE 15
P.2.1. ARQUITECTURA DE ROBOTS. 15
P.2.2. RETO DE LA NAVEGACIÓN AUTÓNOMA 16
P.2.3. LÍNEA DE ACCIÓN 19
P.3. OBJETIVOS 20
P.3.1. OBJETIVO GENERAL 20
P.3.2. OBJETIVOS ESPECÍFICOS 20
P.4. MARCO REFERENCIAL 21
P.4.1. SENSORES 21
P.4.2. ARQUITECTURAS DE CONTROL EN ROBÓTICA MÓVIL. 37
P.4.3. TEORÍA DE ESQUEMAS 42
P.4.4. TIPOS DE PLATAFORMAS MÓVILES 42
P.4.5. MODELO DE CONTROL GUIADO DIFERENCIAL. 49
P.4.6. MAPAS DE ENTORNO 54

1. DESARROLLO DE PROYECTO **62**

1.1. NAVEGACIÓN 63
1.1.1. MOTORES DC. 64
1.1.2. CONTROL DE MOTORES DC. 67
1.1.3. CONTROL DE DIRECCIÒN POR CORRECCIÒN DE VELOCIDAD 69
1.1.4. CONTROL DE SENTIDO DE GIRO DE PLATAFORMA MÒVIL. 69
1.1.5. CONTROL DE INERCIA DE FRENADO EN EL DESPLAZAMIENTO DE LA PLATAFORMA 70
1.1.6. ENCODERS 71
1.1.7. FUSIÓN MOTORES DC Y ENCODERS PARA PLATAFORMA MÓVIL 74
1.2. LOCALIZACIÓN 77
1.2.1. CONEXIÓN Y FUNCIONAMIENTO BRÚJULA DIGITAL. 77
1.2.2. CALIBRACIÓN DEL DISPOSITIVO. 78
1.2.3. IMPLEMENTACIÓN DE BRÚJULA DIGITAL CON INTERFAZ PARA TARJETA FPGA SPARTAN 3E DE XILINX 80
1.3. RECONOCIMIENTO DEL ENTORNO 82
1.3.1. MEDIDOR DE DISTANCIA DE LABORATORIO. 82
1.3.2. OPTIMIZAR MEDIDOR DE DISTANCIA DE LABORATORIO, IMPLEMENTANDO HARDWARE RECONFIGURABLE, FPGAS SPARTAN 3E DE XILINX. 90
1.3.3. PRUEBAS PARA CARACTERIZAR EL MEDIDOR DE DISTANCIA IMPLEMENTANDO HARDWARE RECONFIGURABLE FPGA TARJETA SPARTAN 3E DE XILINX. 110
1.3.4. MEDICIÓN DE DISTANCIA CON HARDWARE RECONFIGURABLE FPGA. 111
1.3.4. DISEÑO Y CONSTRUCCIÓN DEL BLOQUE DE PERCEPCIÓN SENSORIAL CON SENSORES INFRARROJOS. 115
1.3.5. FUSIÓN ANILLO DE SENSORES INFRARROJOS, MOTORES Y ENCODERS PARA LA NAVEGACIÓN Y CONSTRUCCIÓN DE MAPAS DE ENTORNO. 118

2. RESULTADOS. **122**

2.1. RESULTADOS INICIALES. 122

2.1.1. RESULTADOS DE LA APLICACIÓN Y FUSIÓN DEL CONTROL DE MOTORES Y ENCODERS 123
2.1.2. RESULTADOS DE LA MEDICIÓN DEL RUMBO MAGNÉTICO CON BRÚJULA DIGITAL E IMPLEMETACIÒN DE CONTROL EN SPARTAN 3 128
2.1.3. RESULTADOS PRUEBA MEDICIÓN DE DISTANCIA CON ÁNGULO DE INCIDENCIA DE 0 GRADOS. 129
2.1.4. RESULTADOS PRUEBA CON VARIACIÓN EN ÁNGULO DE INCIDENCIA Y VARIACIÓN DE DISTANCIA 130
2.1.5. RESULTADOS ADQUISICIÒN DE SEÑAL, CONVERSIÓN ANALOGO A DIGITAL E INTERFAZ GRÀFICA CON PC. 131
2.1.6. RESULTADOS PRUEBAS PARA CARACTERIZAR EL MEDIDOR DE DISTANCIA IMPLEMENTANDO ARREGLOS DE LÒGICA PROGRAMABLE FPGA SPARTAN 3E DE XILINX. 132
2.1.7. RESULTADOS MEDICIÓN DE DISTANCIA CON INFRARROJOS Y HARDWARE RECONFIGURABLE. CUANTIFICIÓN DEL ERROR DE MEDIDA. 134
2.1.8. RESULTADOS ESTRUCTURA FÍSICA DE SOPORTE PARA SENSORES INFRARROJOS. 136
2.1.9. RESULTADOS ELECTRÓNICA Y ACONDICIONAMIENTO DE LOS SENSORES INFRARROJOS AL SOPORTE FÍSICO EN ANILLO. 137
2.2. RESULTADOS FINALES. 139

3. TRABAJO FUTURO 142

3.1. MEDICIÓN DE DISTANCIAS CON SENSORES INFRARROJOS. 142
3.2. AMBIENTES NO CONTROLADOS. 142
3.3. NAVEGACIÓN AUTÓNOMA 143

4. DISCUSIÓN 144

4.1 MARCO UTP. 144
4.2. MARCO SENSORIAL 145

CONCLUSIONES 146

BIBLIOGRAFÍA 147

ANEXOS 151

LISTADO DE FIGURAS

FIGURA 1. ARQUITECTURA JERÁRQUICA. 17

FIGURA 2. ARQUITECTURA HÍBRIDA. 18

FIGURA 3. COMPONENTES DE UN *ENCODER* 22

FIGURA 4. DIAGRAMA DE TIEMPOS DE *ENCODERS.* 23

FIGURA 5. PRINCIPIO DEL MEDIDOR DE DISTANCIAS CON SENSORES INFRARROJOS 27

FIGURA 6. MODELO FUNCIONAL DE YAIR 28

FIGURA 7. INCIDENCIA DEL HAZ INFRARROJO DE DOS SENSORES CON IGUAL ORIENTACIÓN 31

FIGURA 8. MODELO PHONG 33

FIGURA 9. INCIDENCIA DE HAZ INFRARROJO 33

FIGURA 10. PRINCIPIO DE OPERACIÓN DEL MEDIDOR DE DISTANCIAS 35

FIGURA 11. DEFINICIÓN DE LAS PRIMITIVAS FUNCIONALES DE UN ROBOT MÓVIL EN TÉRMINOS DE LAS ENTRADAS Y LAS SALIDAS 38

FIGURA 12. FALTA DE TRACCIÓN 43

FIGURA 13. - RUEDAS EN CONFIGURACIÓN DIFERENCIAL CON UNA RUEDA LOCA AL FRENTE 44

FIGURA 14- CONFIGURACIÓN TÍPICA EN TRICICLO 45

FIGURA 15- CONFIGURACIÓN TÍPICA ACKERMAN 46

FIGURA 16- MECÁNICA PARA DISEÑAR UNA CONFIGURACIÓN SINCRONIZADA 47

FIGURA 17. A. RUEDA OMNIDIRECCIONAL, B. EJEMPLO DE CONFIGURACIÓN OMNIDIRECCIONA 47

FIGURA 18. PLATAFORMA POSICIONADA EN UN PLANO COORDENADO 50

FIGURA 19. TRACCIÓN DIFERENCIA 51

FIGURA 20. MAPA DE LOCALIZACIÓN DE ROBOT MÓVIL. 55

FIGURA 21. MAPA DE LOCALIZACIÓN URBANO. 55

FIGURA 22. MAPA TOPOLÓGICO COMPLETO 57

FIGURA 23. MAPA DE REJILLA SOBREPUESTO. 60

FIGURA 24. MAPA DE REJILLA CON APROXIMACIÓN. 61

FIGURA 25. ESCENARIO DE PRUEBA DE DISTANCIA RECORRIDA 75

FIGURA 26. ESCENARIO DE PRUEBA PARA GIRO. 76

FIGURA 27. VISTA LATERAL. DESCRIPCIÓN DE ARREGLO SENSORIAL PARA PRUEBA. 88

FIGURA 28. VISTA SUPERIOR DE PRUEBA DE LABORATORIO CON VARIACIÓN DEL ÁNGULO DE INCIDENCIA 89

FIGURA 29. DISTRIBUCIÓN DE PINES DEL MC68HC08JK8 95

FIGURA 30. TRAMA DE COMUNICACIÓN SERIAL RS-232 97
FIGURA 31. MEMORIA DOBLE PUERTO 102
FIGURA 32. ANILLO HEXAGONAL INICIAL 115
FIGURA 33. DISEÑO SOPORTE PARA SENSORES INFRARROJOS. 116

LISTADO DE GRÁFICOS

GRÁFICO 1. DISTANCIA CONTRA VOLTAJE DE SALIDA 36

GRÁFICO 2. RUMBO MAGNÉTICO 63

GRÁFICO 3. LONGITUD DE ARCO PARA *ENCODER*. 72

GRÁFICO 4. FUNCIONAMIENTO DE LA BRÚJULA 78

GRÁFICO 5. CORRIENTE DE POLARIZACIÓN CONTRA LONGITUD DE ONDA DE EMISIÓN. 83

GRÁFICO 6. CARACTERÍSTICA DE EMISIÓN EN EL PLANO. 83

GRÁFICO 7. CARACTERÍSTICA DE SEÑAL A TRANSMITIR 84

GRÁFICO 8. SENSIBILIDAD DEL RECEPTOR CONTRA LONGITUD DE ONDA 86

GRÁFICO 9. SEÑAL DE EXCITACIÓN PARA EMISORES DE INFRARROJO 90

GRÁFICO 10. COMPORTAMIENTO DEL SENSOR MEDIDOR DE DISTANCIAS PARA DIFERENTES SUPERFICIES Y DISTANCIAS 130

GRÁFICO 11. RESPUESTA DEL SENSOR A LA VARIACIÓN DEL ÁNGULO DE INCIDENCIA. 131

GRÁFICO 12. SEÑAL ADQUIRIDA MCU, SPARTAN 3E, PC. 132

GRÀFICO 13. RESPUESTA DEL SENSOR INFRARROJO CONTRA DISTANCIA. 133

GRÁFICO 14. ENTORNO 1. 140

GRÁFICO 15. ENTORNO 2. 140

LISTADO DE FOTOS

FOTO 1. BRÚJULA TRADICIONAL 25

FOTO 2. MOTOR DC MARCA DAYTON 64

FOTO 3. VISTA SUPERIOR ESTRUCTURA 70

FOTO 4. VISTA LATERAL DERECHA E. 70

FOTO 5 .VISTA FRONTAL E. 70

FOTO 6. PERFIL E. 70

FOTO 7. ENCODER STEGMANN 71

FOTO 8. ACOPLE MOTOR DC, RUEDA, *ENCODER* 75

FOTO 9. MONTAJE SUPERFICIAL DE BRÚJULA DIGITAL 77

FOTO 10. FRONTAL PLATAFORMA MÓVIL COMPLETA. 118

FOTO 11. POSTERIOR PLATAFORMA MOVIL COMPLETA 119

FOTO 12. PLATAFORMA MÓVIL COMPLETA 1. 119

FOTO 13. PLATAFORMA MÓVIL COMPLETA 2. 120

FOTO 13. DESPLAZAMIENTO EN ESCENARIO DE PRUEBA DE DISTANCIA RECORRIDA 123

FOTO 14. DESPLAZAMIENTO EN ESCENARIO DE PRUEBA DE GIRO. 125

FOTO 15. ESCENARIO DE PRUEBA PARA GIRO. 125

FOTO 16. DÉCIMA TRAYECTORIA DE PRUEBA. 126

FOTO 17. TRAYECTORIAS RECORRIDAS PARA LA PRUEBA DE GIRO. 126

FOTO 18. ADQUISICIÓN FPGA CON BRÚJULA DIGITAL 128

FOTO 19. VISTA SUPERIOR. ESTRUCTURA DE SOPORTE PARA SENSORES INFRARROJOS. 136

FOTO 20. VISTA INFERIOR. ESTRUCTURA DE SOPORTE PARA SENSORES INFRARROJOS 136

FOTO 24. ANILLO DE SENSORES INFRARROJOS 137

FOTO 25 . ANILLO DE SENSORES INFRARROJOS 137

FOTO 26. . ANILLO DE SENSORES INFRARROJOS 138

FOTO 27. ANILLO DE SENSORES INFRARROJOS 138

FOTO 28. PLATAFORMA MÓVIL COMPLETA 3. 139

LISTADO ESQUEMÁTICOS

ESQUEMÁTICO 1. *DRIVER* MOTORES 65
ESQUEMÀTICO 2. INTERFAZ FPGA- *DRIVER* 66
ESQUEMÁTICO 3. DISTRIBUCIÓN DE LÍNEAS DEL *ENCODE*R. 71
ESQUEMÁTICO 4. INTERFAZ *ENCODERS* – FPGA. 74
ESQUEMÁTICO 5. CONEXIÓN BRÚJULA DIGITAL 81
ESQUEMÁTICO 6. CIRCUITO DE EMISIÓN DE LABORATORIO. 85
ESQUEMÁTICO 7. CIRCUITO DE RECEPCIÓN DE LABORATORIO. 87
ESQUEMÁTICO 8. CIRCUITO PARA EXCITACIÓN DE EMISORES DE INFRARROJOS 93
ESQUEMÁTICO 9. ACONDICIONAMIENTO SEÑAL DE RECEPCIÓN DE INFRARROJOS. 93
ESQUEMÁTICO 10. CIRCUITO IMPRESO DE EMISIÓN, RECEPCIÓN DE INFRARROJOS, ACONDICIONAMIENTO Y CAPTURA DE SEÑAL Y ENVÍO A LA TARJETA FPGA. 117

LISTADO DE TABLAS

TABLA 1. CONTROL DE DIRECCIÓN PARA MOVIMIENTO..... 66
TABLA 2. BLOQUES RAM DISPONIBLES EN EL DISPOSITIVO SPARTAN 3E..... 101
TABLA 3. DESCRIPCIÓN DE SEÑALES DEL BLOQUE DE MEMORIA..... 102
TABLA 4. CONFIGURACIÓN DE BLOQUES DE MEMORIA. 105
TABLA 5. DATOS DE PRUEBA DE DISTANCIA RECORRIDA 124
TABLA 6. DATOS DE PRUEBA PARA DISTANCIA Y GIRO..... 127
TABLA 7. DATOS PARA CALCULAR LA CONSTANTE PROPIA DEL SENSOR DE INFRARROJOS..... 133

PRELIMINARES

P.1. INTRODUCCIÓN

El trabajo descrito demuestra el diseño e implementación de un anillo de sensores infrarrojos como dispositivos medidores de distancia con arreglos de hardware reconfigurable FPGA, para la navegación de robots móviles en ambientes controlados.

Se muestra una breve reseña sobre la arquitectura de robots móviles y la importancia de determinar una estructura de control para formular alternativas de navegación y construcción de mapas de entorno con la implementación de sensores que sirvan al sistema conocer la distancia que ha recorrido, su ubicación, y la distancia que existe entre el ambiente exterior y el sistema mismo, con la implementación de arreglos de hardware reconfigurable FPGAS, en este caso la tarjeta SPARTAN 3E de XILINX.

Fusionar la información que entregan los distintos tipos de sensores al arreglo de hardware reconfigurable es uno de los temas a tratar, pues en cada tipo de sensor se usaron técnicas de adquisición de datos que van desde medición de ancho de pulso hasta protocolo de comunicación serial RS-232 e I2C, con el debido procesamiento de cada fuente de datos en el arreglo de hardware reconfigurable.

Un preámbulo sobre el trabajo realizado, está ligado directamente a la naturaleza inherente del arreglo de hardware reconfigurable para ejecutar tareas en concurrencia y el proceso para la medición de distancia, con los datos que entrega el bloque de percepción infrarroja y los otros bloques sensoriales.

Por ejemplo, en esta aplicación la tarjeta FPGA está en capacidad de capturar datos en paralelo de determinado número de fuentes de datos, estos, están limitados sólo por el número de entradas de la misma; el proceso de medición de distancia en la FPGA está implementado para la navegación y construcción de mapas de entorno usando el computador sólo como interfaz gráfica.

P.2. ESTADO DEL ARTE

P.2.1. ARQUITECTURA DE ROBOTS.

Los primeros trabajos en robótica móvil fueron para ambientes industriales con un grado mínimo de incertidumbre, por lo cual la capacidad sensorial era la menor, de tal forma que los robots móviles necesitaban de los sensores sólo para la realización de tareas muy específicas [1].

En los últimos años el principal objetivo de investigación en el área de la robótica móvil es la construcción de entes o sistemas que tomen la información proporcionada por un módulo sensorial, como sensores de ultrasonidos, infrarrojos, odométricos, visión artificial, etc., y sea capaz de moverse inteligentemente por un entorno y realizar una serie de tareas de forma autónoma.
‘‘Un robot móvil debe ser capaz de procesar la información sensorial suministrada por los sensores y adoptar una acción de control que le lleve a un estado definido como meta” [2].

Actualmente el desarrollo de las capacidades de los sistemas robóticos, es el resultado del trabajo conjunto de especialistas en áreas como ingeniería mecánica, ingeniería de sistemas, telemática, arquitectura de computadores, ingeniería del software e inteligencia artificial.

Atendiendo a la recomendación general que la *IEEE Robotics and Automatization Soviet* lanzó en el año 2001 [3], el desafío actual es encontrar las aplicaciones adecuadas para los desarrollos y aumentar las capacidades de los sistemas robóticos para extenderlos a nuevas aplicaciones.
La tendencia de los diferentes grupos de investigación en el área de robótica se orienta a que un robot móvil esté capacitado para construir un mapa de su entorno de trabajo, así como de navegar de manera autónoma, evitando obstáculos a través de él.

Una situación más deseable y más desafiante es que el robot móvil tenga la capacidad de generar y mantener su mapa de manera autónoma, mientras logra los objetivos propuestos.
Son muchas las herramientas que se han implementado en el logro de estos objetivos, una de las estrategias propuestas se basa en el empleo de las técnicas inteligentes y la navegación basada en comportamientos.

El desarrollo de robots móviles capaces de construir su propio mapa y navegar de manera autónoma, requiere de estrategias de control inteligente, que puedan manejar la incertidumbre presentada por el entorno de trabajo, mientras se desempeñan en tiempo real con una relativa baja carga computacional.

P.2.2. RETO DE LA NAVEGACIÓN AUTÓNOMA

Cualquier aproximación al control de un sistema dinámico necesita utilizar algún conocimiento o modelo del sistema a controlar.

En el caso de un robot móvil, este sistema consiste del robot en sí, más el ambiente en el que opera. Desafortunadamente, mientras el modelo del robot como tal puede ser obtenido, la situación es diferente si éste se considera dentro de cualquier tipo de ambiente real y sin ningún arreglo previo del entorno.

Los ambientes están caracterizados por la presencia de incertidumbre; la naturaleza de los fenómenos involucrados es tal, que frecuentemente no se está en capacidad de precisar un modelo o cuantificar estos fenómenos. Por ejemplo, si se considera la incertidumbre inducida por la presencia de personas en un ambiente. Las personas se mueven alrededor y pueden cambiar la posición de los objetos y los enseres del entorno; sin embargo, en la mayoría de los casos no se puede obtener una distribución probabilística que pueda caracterizar estos eventos. La interacción del robot con el ambiente causa dificultades similares y sus acciones de sensado son influenciados por condiciones ambientales, las cuales son difícilmente contabilizadas; por ejemplo, el error en el movimiento del robot puede variar como resultado de un piso mojado, la calidad del reconocimiento visual puede ser afectado por un rápido cambio de las condiciones de luz.

Una estrategia para afrontar esta gran cantidad de incertidumbre es abandonar la idea de obtener un modelo completo del ambiente en la fase de diseño y dotar al robot con la capacidad de construir este modelo por si mismo y en línea. Esta estrategia es llamada Arquitectura Jerárquica [4], Ver Figura 1.

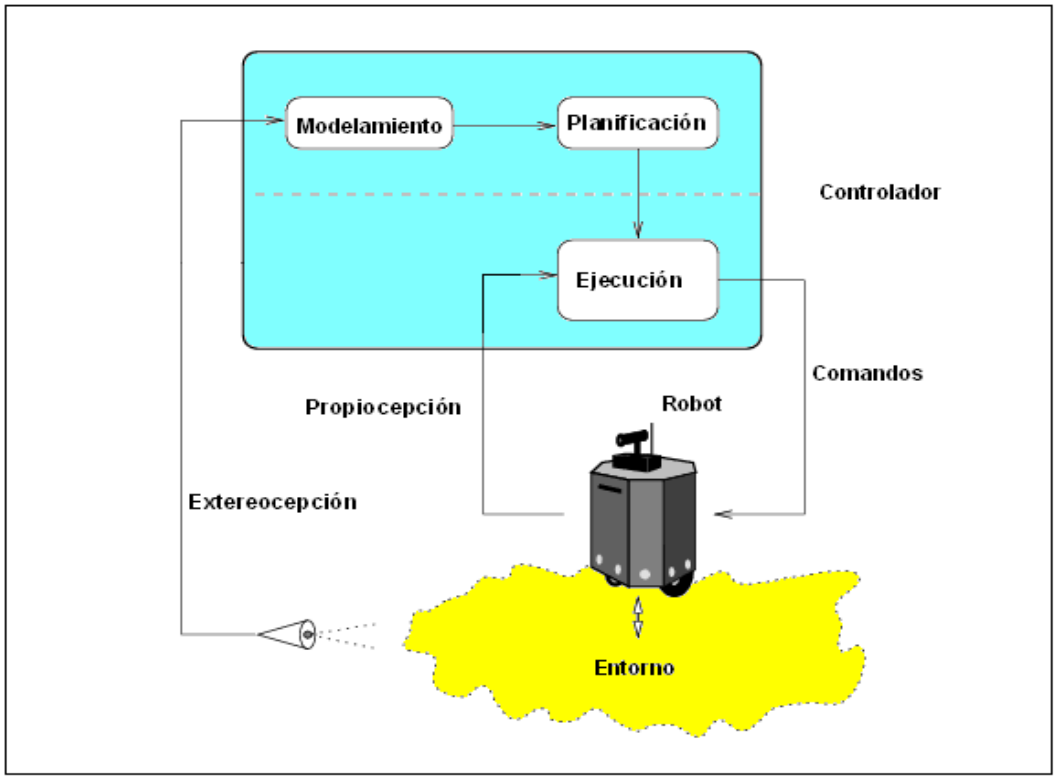


FIGURA 1. Arquitectura Jerárquica.

Utilizando los sensores propioceptivos, el robot adquiere un modelo del espacio de trabajo en el mismo instante en que la tarea será realizada. En este modelo se basa un programa planeador para construir un plan que ejecutará la tarea dada en el ambiente dado. Este plan es pasado al programa de control de bajo nivel.

Típicamente el procedimiento de ejecución es ciego. El controlador puede utilizar un modelo del robot y monitorear el estado de los efectores del robot (Propiocepción), pero esto no permite sensar o modelar el ambiente una vez más.

No es difícil observar las limitaciones de la Arquitectura Jerárquica para trabajo en tiempo real en ambientes de trabajo. El modelo adquirido por el robot necesariamente es incompleto e inexacto debido a la incertidumbre en la percepción, además el modelo se desactualiza rápidamente por la dinámica del ambiente entonces el plan construido puede ser inadecuado para el ambiente encontrado durante la fase de ejecución, otro factor preponderante es que los procesos de modelado y planeamiento son computacionalmente complejos y consumen mucho tiempo de cómputo, con lo cual se agranda el problema; intuitivamente una realimentación (lazo de realimentación) con el ambiente puede mejorar estas limitaciones, es por ello que aparece el enfoque SMPA *"SENSE – MODEL – PLAN - ACT*".

La complejidad del SMPA puede hacer que la respuesta temporal del sistema en ambientes dinámicos tarde incluso segundos.

Las investigaciones en robótica móvil tienden al desarrollo de numerosas y nuevas arquitecturas que integran Percepción y Acción. La tendencia general se orienta al planeamiento de pequeñas suposiciones sobre el ambiente encontrado en la fase de ejecución y que la ejecución sea sensible al ambiente y se adapte a las contingencias encontradas.

Para lograr esto, los datos de percepción han sido incluidos en la capa de ejecución (Arquitecturas Híbridas). Ver Figura 2.
La aparente extensión del sistema extereoceptivo al módulo de ejecución tiene dos consecuencias importantes:

1. La interacción del robot con el ambiente es mucho más ceñida dado que ahora el ambiente está incluido como lazo cerrado con la capa o módulo de ejecución.
2. La complejidad de la capa de ejecución ha sido incrementada debido a la necesidad de considerar múltiples objetivos: alcanzar los objetivos tácticos tomados del planeador (capa o fase de planificación) y reaccionar ante los eventos ambientales detectados por la percepción.

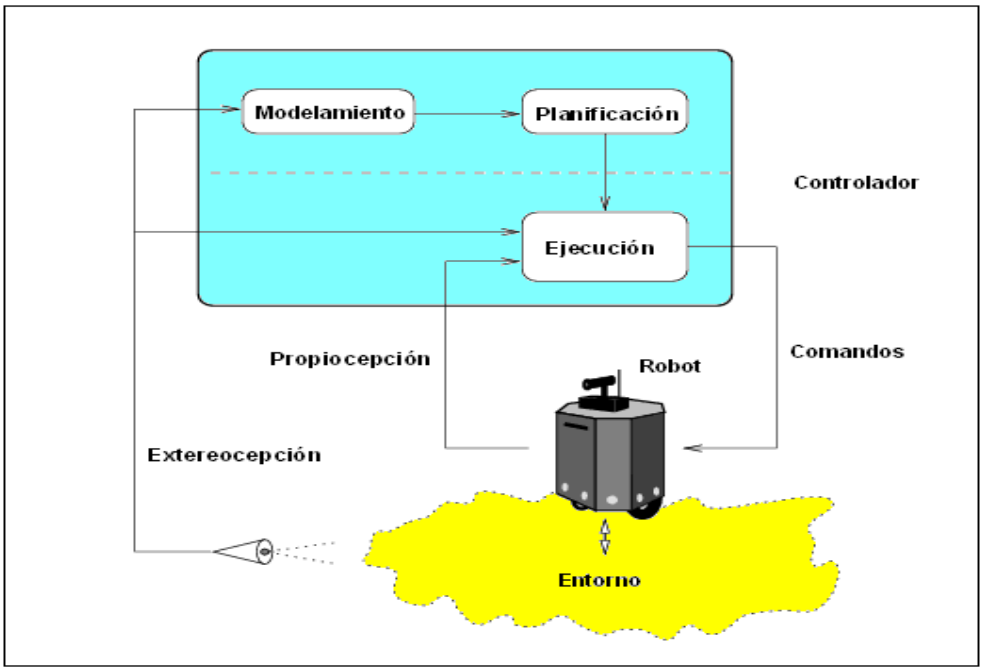


FIGURA 2. Arquitectura Híbrida.

Muchos investigadores han escogido cubrir esta complejidad con la estrategia “Divide y Conquistarás”, pero esto ha avocado a diferentes estilos de división. Algunos han perseverado dentro de la tradición jerárquica y tienen fuertemente seccionada la capa de ejecución en capas de secuencias y capas de proceso. Otros como Mataric [5] y Parker [6] parten de la Arquitectura Jerárquica y descomponen la capa de ejecución en pequeñas unidades de Decisión-Acción o comportamientos (Arquitectura Híbrida Basada en Comportamientos).

Un comportamiento es una pequeña unidad de control hecha para alcanzar un objetivo simple en un restringido conjunto de situaciones. Para poder realizar tareas complejas los robots autónomos necesitan, de la activación y cooperación de un número de comportamientos, cada comportamiento implementa en detalle un control seguro, para una sub-tarea específica como seguir una trayectoria, evitar objetos sensados o cruzar puertas.

Las Arquitecturas Híbridas no resuelven el problema de navegación autónoma pero ellas proporcionan una estructura conveniente en la cual los diferentes sub-problemas pueden ser repartidos e integrados.

P.2.3. LÍNEA DE ACCIÓN

La implementación de un tipo de arquitectura determinado para el proyecto está ligada directamente a la acción en tiempo real, y la capacidad de ejecutar tareas. Si se piensa en el ser humano desde un punto de vista perceptivo, éste se denotaría como un ente dotado de una cantidad ilimitada de sensores en funcionamiento y en concurrencia, con una capacidad de organizar las variables entregadas por los mismos y decidir cuál de estas variables tiene un valor ponderado que requieran una acción de control. Por esto, la implementación de hardware programable encaja perfectamente en el perfil para generar sistemas con un grado de autonomía mayor, pues con estos dispositivos la ejecución de procesos en concurrencia es algo inherente, además que los procesos pueden adquirir un valor ponderado como resultado de un orden jerárquico que es asignado por el diseñador, sin necesidad de tener en cuenta conceptos utilizados en sistemas anteriores como multiplexar datos y secuencias serie que aportaban mas tiempo en ejecución de cómputo.

P.3. OBJETIVOS

P.3.1. OBJETIVO GENERAL

Diseñar y construir un bloque de percepción sensorial con base en sensores infrarrojos para medición de distancias para un sistema de mapeo de entorno y navegación en plataforma móvil, utilizando hardware reconfigurable, FPGA.

P.3.2. OBJETIVOS ESPECÍFICOS

- Diseñar el proceso de construcción de mapas de entorno y navegación
- Diseñar algoritmos que permitan al sistema medir distancias.
- Diseñar, construir y acoplar un anillo de sensores infrarrojos (bloque de percepción sensorial) para medición de distancias en un sistema de mapeo de entorno y navegación**.**
- Implementar tecnologías de lógica interconectada como el hardware reconfigurable FPGAS para la medición de distancias, acondicionamiento de señal y para el sistema de mapeo de entorno, y navegación.

P.4. MARCO REFERENCIAL

P.4.1. SENSORES

Los sensores son en la plataforma móvil la fuente de datos sobre los cambios tanto en su entorno (distancias a objetos, luz ambiental) como en sí mismo (nivel de las baterías, consumo de los motores, pulsos de *encoders* acumulados, etc.). Desde un punto de vista funcional puede decirse que un sensor es un dispositivo que capta la magnitud de una variable física en un sistema físico o entorno. Cualquier dispositivo que es alterado por las variaciones de una magnitud física de una forma predecible y mesurable es un sensor para ésta magnitud. Así, un sensor viene caracterizado por su función de transferencia (modelo del sensor), que relaciona el valor de la magnitud física con el valor que éste suministra en su salida. Como predecía Yamasaki [7], la tendencia hacia sensores con mayor inteligencia y capacidad de proceso digital es evidente.

En robótica móvil se usan una gran variedad de sensores que miden distintas magnitudes. Los que más se utilizan son los orientados a resolver uno de los problemas fundamentales de todo robot móvil: la determinación de la posición. En los trabajos de Everett y Peters [8], y Feng, Borenstein y Everett [9], se lleva a cabo una revisión de todas las tecnologías y metodologías utilizadas a nivel de sensores y procesamiento de la información proporcionada por los mismos para resolver el problema de la localización de un robot. Adams [10], tiene otra revisión algo más actualizada que incluye detalles sobre el procesamiento básico de cada sensor. Básicamente el problema de la localización ó el posicionamiento se resuelve mediante medidas relativas de posición o mediante sistemas absolutos de posicionamiento.
Para el desarrollo del proyecto se utilizaron en la sensorización de la plataforma móvil (Robot Móvil):

- *Encoders*, como medidores de distancia recorrida por las ruedas (sensores Odométricos).
- Brújula digital, como sensor de dirección y rumbo.
- Sensores infrarrojos para medir distancias de obstáculos hacia la plataforma.

P.4.1.1. *ENCODERS*

La creciente presencia de sistemas digitales para el tratamiento y presentación de la información en los sistemas de medida y control hace muy atractivos aquellos transductores que ofrecen directamente a su salida una señal en forma digital, por la sencillez que supone el

acondicionamiento de señales y su mayor inmunidad a las interferencias electromagnéticas. Hay que destacar que no existe prácticamente ningún fenómeno cuya transducción ofrezca directamente una salida digital. Lo que se hace es convertir una magnitud analógica de entrada en una señal digital por medio de un transductor, sin necesidad de convertir una tensión analógica en su equivalente digital.

Las aplicaciones de los codificadores de posición son relativas a la medida y control de posición lineal y angular de alta resolución. Se emplean así en robótica: *plotters*, máquinas y herramientas, posicionamiento de cabezales de lectura en discos magnéticos y de fuentes de radiación en radioterapia, radares, orientación de telescopios…etc. Este sistema se va a centrar en los transductores digitales, concretamente en los *encoders*.

P.4.1.1.1. *ENCODERS* INCREMENTALES.

Su nombre se debe a que la señal generada está en formato de cuadratura incremental, es decir, dos señales A y B desfasadas 90º con una señal de índice que cambia cada vez que se registre un giro completo.

En un codificador de posición incremental hay un disco, con poca inercia, que se desplaza solidario a la pieza cuya posición se desea determinar, por ejemplo, el eje de un motor. El disco posee dos tipos de zonas: transparente (agujeros) y opaca, dispuestas de forma alternativa y equidistante, tal como muestra la Figura 3:

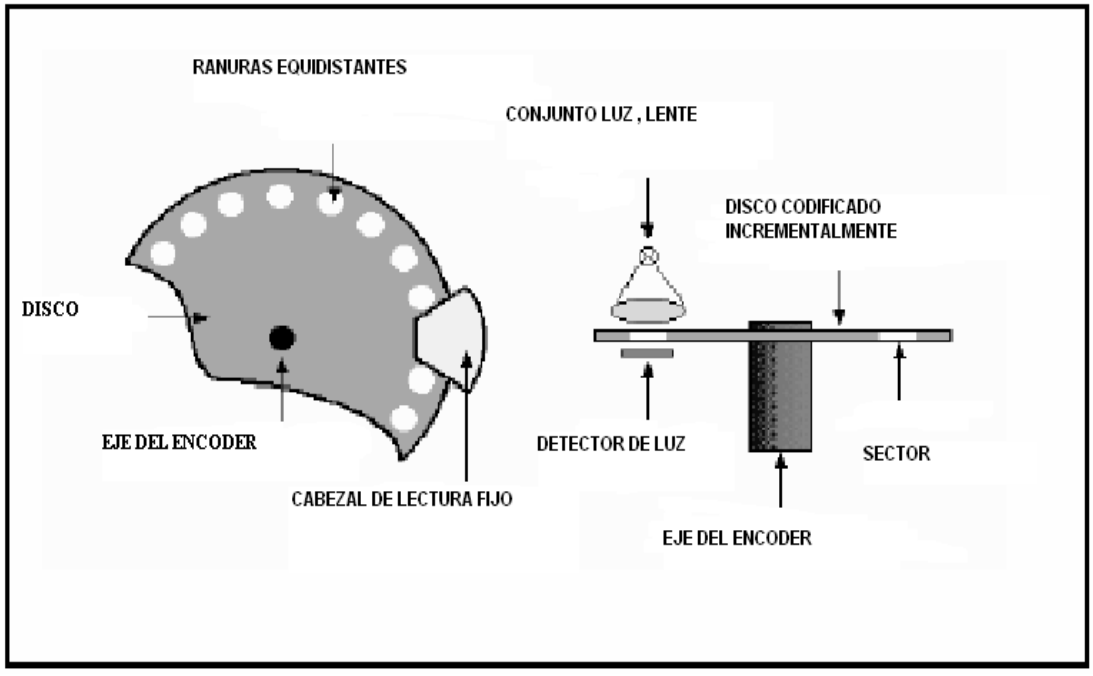


FIGURA 3. Componentes de un *Encoder*[1]

[1] Figura 17 (www.salle.url.edu/~se04635/20F%20Encoders.pd)

Mediante un cabezal de lectura fijo óptico se detecta el cambio de posición angular. La simplicidad y economía de esta técnica no admiten duda, pero tiene una serie de inconvenientes que conviene señalar:

1. La información sobre la posición se pierde en cuanto falla la alimentación del sistema, o simplemente cuando se desconecta, y en presencia de fuertes interferencias.
2. Es necesario un contador bidireccional para poder tener una salida digital compatible con los elementos de entrada-salida de un ordenador.
3. No permite detectar el sentido de avance si no se dispone de elementos adicionales a los indicados en la Figura 3.

P.4.1.1.2. CLASIFICACIÓN DE *ENCODERS* INCREMENTALES

- UNIDIRECCIONALES:

Entregan una salida y no se puede determinar el sentido de giro. Sólo servirá para obtener valores absolutos. Por ejemplo, para obtener velocidades absolutas sin importar el sentido de giro.

- BIDIRECCIONALES:

Ofrece dos salidas A y B. El sentido se va a distinguir por la diferencia de fase. Será útil cuando se necesite saber coordenadas exactas tanto positivas como negativas. Ver Figura 4

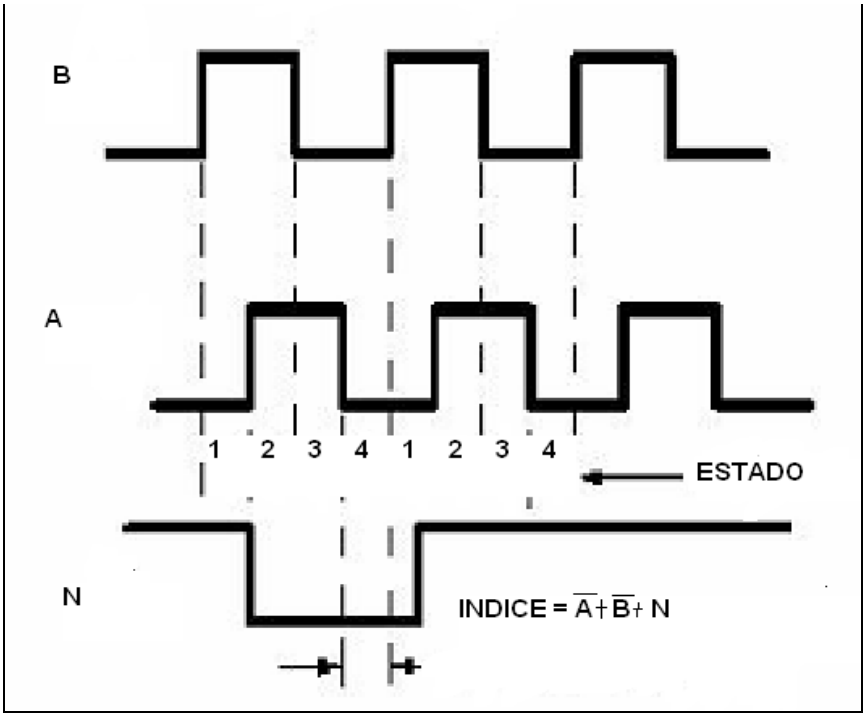


FIGURA 4. Diagrama de Tiempos de *Encoders.*

- ABSOLUTOS

Los *encoders* absolutos van a funcionar en todo momento dando la posición angular del eje. El funcionamiento básico es muy similar al incremental, las lentes de adaptación correspondientes, el disco graduado y los fotorreceptores. El disco transparente se deviene en un número de sectores potencia de 2, codificándose de forma binaria en cualquiera de las formas posibles que se comentará más adelante, lo cual queda representado por zonas transparentes y opacas dispuestas radialmente.

Los encoders absolutos no necesitan ninguna herramienta especial para obtener el sentido de giro, ya que cada sector está codificado de manera absoluta. La resolución es fija y vendrá dada por el número de anillos concéntricos que contenga el disco. Las resoluciones habituales van desde 2^8 a 2^{19} posiciones diferentes [11].

P.4.1.2. BRÚJULA

La brújula es un instrumento que sirve de orientación, que tiene su fundamento en la propiedad de las agujas magnéticas. Por medio de una aguja imantada señala el Norte magnético, que es ligeramente diferente para cada zona del planeta, y distinto del Norte geográfico. Utiliza como medio de funcionamiento el magnetismo terrestre. La aguja imantada indica la dirección del campo magnético terrestre, apuntando hacia los polos norte y sur. Únicamente es inútil en las zonas polares, debido a la convergencia de las líneas de fuerza del campo magnético terrestre.

En la Foto 1 se presenta una brújula tradicional para trabajos de campo, para el proyecto la implementación de este tipo de brújula no sería óptimo puesto que la adquisición de la información de la brújula es para ser vista por el ser humano y no para un sistema autónomo.

FOTO 1. Brújula Tradicional[2]

Ahora bien para el proyecto se implementó una brújula digital la cual consta de sensores de campos magnéticos que una vez calibrados ofrecen una precisión de 3 a 4 grados y una resolución de décimas.

[2] Foto 1 (http://upload.wikimedia.org/wikipedia/commons/f/f9/Liquid_filled_compass.jpg**)**

PRINCIPIO DE FUNCIONAMIENTO BRÚJULA DIGITAL

La brújula digital tiene dos coordenadas X y Y. Cada coordenada reporta la fuerza de la componente de campo magnético paralela a esta. La coordenada X reporta (X. Cos (θ)), y la coordenada Y reporta (X. Sen (θ)). Para hallar θ (que es el ángulo de inclinación con la parte frontal del aparato y el Norte en sentido horario) se usa la ATAN (-Y/X) con lo cual se obtiene fácilmente el punto cardinal hacia el cual se dirige, a través de una escala de 0 a 360º donde el cero representa al Norte Geográfico [12].

La brújula como se mencionó anteriormente sirve de orientación; la orientación para la aplicación del proyecto es fundamental puesto que la plataforma móvil (Robot móvil) tiene como función desplazarse en ambientes controlados, pero cada registro de movimiento debe ser verificado por el dato que entrega la brújula digital para establecer un control de movimiento y rumbo, es decir, la plataforma en una ubicación localizada entrega una ángulo proveniente de la brújula, el controlador con la información de los demás sensores indica un nuevo movimiento; si la plataforma se ha movilizado en línea recta la brújula deberá entregar un ángulo cercano o igual al tomado con anterioridad, la aplicación de este dispositivo no sólo se limita para el desplazamiento lineal sino también para la situación de giro en la que se verifica cuantos grados a girado la plataforma sobre su propio eje para su posterior generación de mapa de navegación.

P.4.1.3. SENSORES INFRARROJOS

A continuación se da una breve introducción científica sobre el funcionamiento de los sensores infrarrojos y algunos estudios previos sobre modelos y técnicas para la medición de distancias con los mismos

El sensor de infrarrojos se basa en la intensidad de una señal luminosa. La luz se puede definir como el conjunto de radiaciones electromagnéticas que forman parte del espectro visible. La radiación se puede definir como la emisión o transferencia de energía en forma de ondas electromagnéticas o partículas. Las ondas luminosas (ultravioleta, visible e infrarroja) ocupan sólo una parte muy pequeña del espectro de ondas electromagnéticas.

La luz visible se extiende sobre un pequeño rango de las ondas luminosas: 0.4 a 0.8 micrones. La parte infrarroja del espectro, la cual no es percibida por el ojo humano, se extiende sobre el rango de longitudes de onda que van desde 0.8 a 100 micrones. De acuerdo a lo anterior, uno de los principales inconvenientes de las aplicaciones de medición de distancias por infrarrojos es el efecto de otras señales luminosas que no son propiamente las del sensor (luz solar, luces artificiales).

Un sensor de distancia por infrarrojos está compuesto típicamente por un emisor LED y un fotodiodo encerrado en una cápsula que filtra la radiación visible. En su funcionamiento más básico el emisor es excitado mediante un pulso de corriente continua, emitiéndose así un haz de luz del espectro infrarrojo que al ser reflejado por algún objeto, entonces es captado por el fotodiodo el cual produce un pequeño voltaje de salida que depende de la cantidad de energía que alcance al receptor. La energía que refleja el objeto depende principalmente de la distancia a la que se encuentre, del coeficiente de reflexión del objeto (dependiente a su vez del color del objeto, del brillo y de la textura) y del ángulo θ de incidencia del haz sobre la superficie reflectora. Ver Figura 5

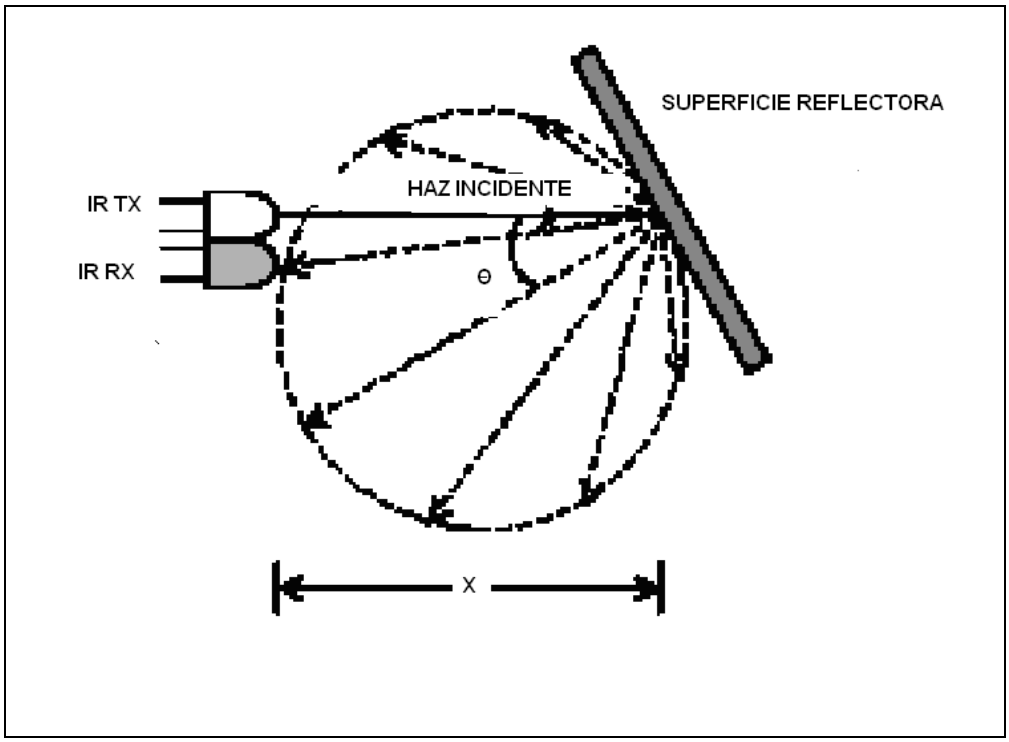


FIGURA 5. Principio del Medidor de Distancias con Sensores Infrarrojos.

Las áreas de aplicación de los sensores infrarrojos incluyen robótica, automatización, control de procesos, telemetría, y sistemas de seguridad entre otros. Específicamente en el área de robótica móvil las aplicaciones típicas han sido el seguimiento de trayectorias definidas en el plano de rodamiento (*docking)* [13], y la navegación evitando obstáculos a cortas distancias [14].

Otros investigadores han fusionado la información de sensores infrarrojos y de ultrasonidos con los mismos propósitos de navegación y evitación de obstáculos: A. Flynn lo hace en forma complementaria a fin de detectar puertas y pasillos en la trayectoria de un robot [15], mientras que A. Sabatini desarrolla un sensor compuesto de infrarrojos y ultrasónicos para integrarlo a

sillas de ruedas como instrumento de ayuda a la navegación [16]. Mas que nada, la información de los infrarrojos ha sido explotada en forma binaria de detección o no detección. Sin embargo, recientemente se ha comenzado a explotar la información de amplitud de señal, sobretodo en aplicaciones que involucran la determinación de distancias. P. Novotny y N. Ferrier utilizan el modelo luminotécnico de Pong aplicado al haz de infrarrojos para determinar la distancia y las propiedades de una superficie plana [17], desarrollan un medidor de distancia de corto rango basado en la fusión de datos de amplitud de señal de dos sensores infrarrojos de muy bajo costo, utiliza este sensor en combinación con sensores de ultrasonido para la construcción de mapas de entorno en aplicaciones de robótica móvil [18]. T. Aytaç y B. Barshan utilizan sensores infrarrojos de bajo costo en la clasificación y localización de paredes, esquinas, ángulos y columnas cilíndricas en ambientes interiores en aplicaciones de robótica móvil. En principio se basan en la información provista por un sólo sensor a la cual aplican algoritmos de mínimos cuadrados y filtros de correspondencia (*Matched-Filter*) a fin de clasificarlos y determinar su posición [19], mientras que en un segundo trabajo, con el mismo propósito, fusionan la información de dos sensores infrarrojos mediante un algoritmo basado en reglas [20].

P.4.1.3.1. MODELO DE YAIR

El sensor de infrarrojos del Robot Yair esta compuesto por dos diodos *LED* altamente directivos que son los emisores de infrarrojos, y por un fotodiodo *PIN* que sirve como receptor de la energía radiada. Todo el conjunto esta alineado verticalmente para corresponder a un sensor Tx/Rx azimut-puntual, como se muestra en la Figura 6.

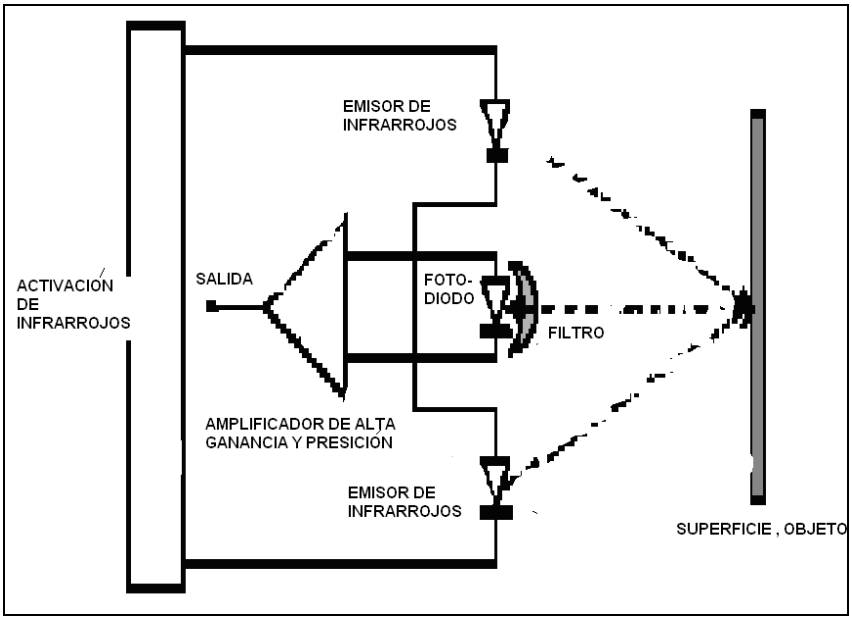


FIGURA 6. Modelo Funcional de YAIR

Para la determinación de la distancia, el sensor se basa en la medición directa de la energía infrarroja que se refleja del objeto. Se puede comprobar que la intensidad de la señal devuelta obedece a las siguientes características:

• Decae aproximadamente en forma cuadrática con la distancia entre el reflector y el receptor.
• Es mayor cuanto más reflectivo sea el objeto. El coeficiente de reflexión depende principalmente del color, brillo y textura del objeto.
• Es mayor cuanto más cercano sea a 0º el ángulo que forma el sensor con la normal al objeto reflector.

Considerando la ley de fotometría cuadrática-inversa [21] y un reflector Lambertiano perfecto, y se puede modelar la salida del sensor como:

$$S(x,\theta) = \left(\alpha * Cos\theta / x^2\right) + \beta, \rightarrow (18)$$

En α se incluyen las características del transductor (intensidad de los emisores, sensibilidad al espectro de los fotodiodos y la ganancia del sensor) así como del objeto (coeficiente de reflexión del objeto). Los tres primeros factores son constantes para todas las medidas realizadas con el mismo sensor, por lo que se puede expresar α como el producto de dos factores, uno constante α*0* en el cual se incluyen aquellos factores dependientes del transductor y se mide en Voltios por metro cuadrado, y otro α*i* como coeficiente de reflexión que varía desde 0 para objetos negros a 1 para el blanco. Por tanto la expresión para α resulta:

$$\alpha = \alpha 0 \times \alpha i, \rightarrow (19)$$

El parámetro β modela el desplazamiento de la señal debido a la luz ambiental. El valor del mismo puede ser obtenido fácilmente mediante la lectura de la señal en el fotodiodo receptor, con los emisores puestos a cero. Una vez obtenido el valor de β, se podrán realizar medidas, que al restarle dicho coeficiente, corresponderán con una señal libre de influencias externas al sensor. De esta forma podemos simplificar la ecuación del sensor como:

$$y(x,\theta) = S(x,\theta) - \beta = \left(\alpha 0 * \alpha i * Cos\theta\right) / x^2, \rightarrow (20)$$

P.4.1.3.2. TÉCNICAS DE MEDICIÓN DE DISTANCIAS CON SENSORES INFRARROJOS

Los sensores de ultrasonido ofrecen la posibilidad de medir distancias a bajo costo y buena precisión. Sin embargo su uso en ciertas aplicaciones está limitado debido sobretodo a su amplio ancho de haz, a su sensibilidad a múltiples reflexiones, y a su poca capacidad para diferenciar objetos a cortas distancias (0 a 0.5 m). Estas limitaciones hacen poco práctico el uso de ultrasonidos en aplicaciones en vehículos autoguiados *(docking)* o en la navegación por seguimiento de paredes. En este tipo de aplicaciones es donde los sensores de infrarrojo aventajan a los sensores de ultrasonido, ya que los primeros ofrecen fiabilidad a bajo costo en la medición de distancias de corto rango (0.1 a 0.8 m), y además con mejores tiempo de respuesta.

Los sensores infrarrojos han sido ampliamente usados como detectores de proximidad binarios en aplicaciones de robótica móvil sin embargo se han hecho pocos intentos de utilizar la característica de amplitud de la señal reflejada para propósitos de estimación de distancia. Este debido sobretodo a su comportamiento no lineal y a la fuerte dependencia de la manera en que las superficies dispersan, absorben o reflejan el haz de infrarrojos. G. Benet [22] hace uso de dos sensores igualmente orientados a fin de cancelar la dependencia del ángulo de incidencia sobre la superficie; mientras que P. Novotny y N. Ferrier [17] hacen uso de un anillo de sensores de infrarrojos y del modelo de luminancia de Phong para determinar la distancia y la orientación de una superficie reflectora. También existen modelos comerciales que integran técnicas opto-electrónicas para determinar la distancia en rangos hasta de 10 m, pero en general presentan la desventaja de poseer un alto costo.

P.4.1.3.2.1 TÉCNICA DE FUSIÓN BI-SENSORIAL

Ya se estableció que la señal de salida de un sensor de infrarrojos puede modelarse como:

$$y(x,\theta) = S(x,\theta) - \beta = (\alpha 0 * \alpha i * Cos\theta) / x^2, \rightarrow (20)$$

De acuerdo a lo anterior, para conocer la distancia entre el sensor y el objeto reflector es primordial conocer a priori el ángulo de incidencia θ y la característica de reflexión αi de la superficie del objeto.

$$x = \sqrt{(\alpha 0 / y)} \times \sqrt{(\alpha i * Cos\theta)}, \rightarrow (21)$$

Dando por hecho que se puede determinar a priori la característica de reflexión típica α*i* del entorno de trabajo del un robot, es entonces el ángulo de incidencia θ el que juega el papel mas importante en la determinación de la distancia de un objeto.

Aunque en un entorno real el ángulo de incidencia θ es desconocido, como una primera aproximación se podría suponer que la incidencia es cercana a 0º.

Esto supondría una sobreestimación de la distancia cuando lo anterior no sea estrictamente cierto. De aquí que si se puede estimar mejor el ángulo de incidencia, se podría corregir la estimación de la distancia previamente hecha.

En la Figura 7 se muestra el escenario regular para la implementar la técnica de fusión donde el ángulo de incidencia θ se puede estimar mejor si se fusiona la información proporcionada por un par de sensores de infrarrojos, que aunque estén separados una cierta distancia L, tengan la misma orientación.

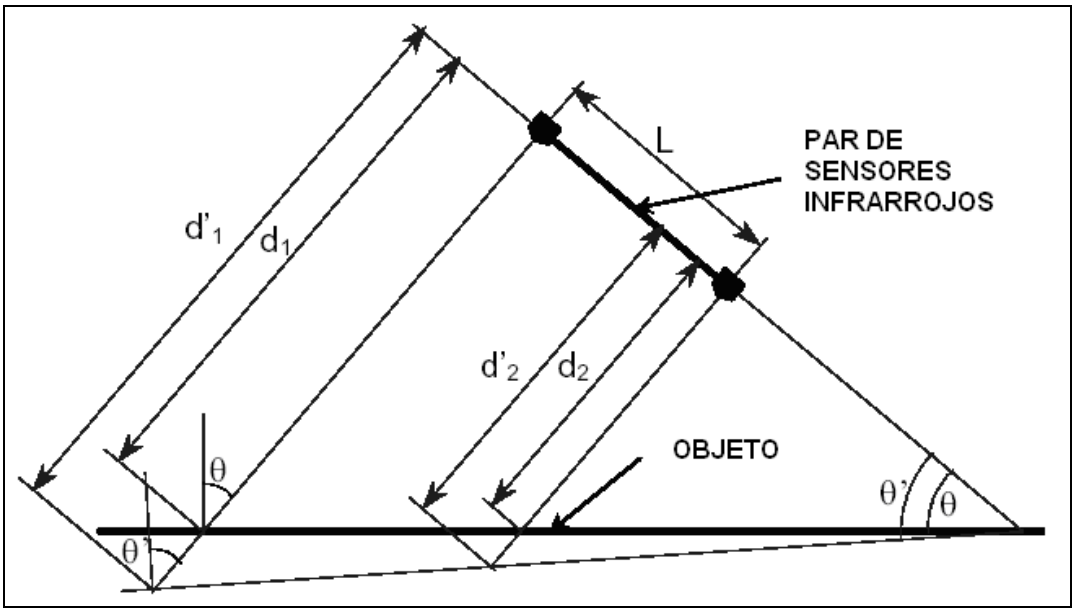


FIGURA 7. Incidencia del haz Infrarrojo de dos Sensores con Igual Orientación.

Asumiendo que el ángulo de incidencia θ′ es muy próximo a cero, las distancias erróneas estimadas en cada sensor corresponden a:

$$x1^{'} = \sqrt{(\alpha / y1)}, \rightarrow (22)$$

$$x2^{'} = \sqrt{(\alpha / y2)}, \rightarrow (23)$$

Mientras que las distancias reales son,

$$x1 = \sqrt{(\alpha / y1)} \times \sqrt{(Cos\theta)} = x1' \times \sqrt{(Cos\theta)}, \rightarrow (24)$$

$$x2 = \sqrt{(\alpha / y2)} \times \sqrt{(Cos\theta)} = x2' \times \sqrt{(Cos\theta)}, \rightarrow (25)$$

$$Tan\theta = (x1 - x2)/d = \left(\sqrt{Cos\theta}\right) * (x1' - x2')/d, \rightarrow (26)$$

Por lo que para obtener el valor exacto de θ es necesario resolver una ecuación cúbica de la forma:

$$R^2 * Cos^2\theta - 1 = 0, \rightarrow (27)$$

Lo anterior puede ser computacionalmente extenso. Una alternativa es la construcción de una tabla de pares (*R*, θ) que pueda ser usada para obtener un valor de θ aproximado mediante la simple interpolación lineal.

P.4.1.3.2.2. TÉCNICA CON EL MODELO DE PHONG

Un haz de luz que impacta sobre una superficie es dispersado, absorbido, o reflejado de acuerdo a las características particulares de cada superficie. Por ejemplo una superficie de color negro absorbe más energía que una de color blanco, mientras que una superficie brillante refleja más energía que una superficie opaca. El modelo de Phong contempla estos efectos mediante el uso de 4 constantes: *C*0, *C*1, *C*2 y *n*. La ecuación de Phong para la intensidad de la energía reflejada por una superficie es,

$$I = C_0(\mu s * \mu n) + C1 * (\mu r * \mu v)^n + C2, \rightarrow (28)$$

Donde *μs, μn, μr*, y *μv* son los vectores unitarios del haz incidente, de la normal a la superficie, del haz reflejado y del punto de observación respectivamente. Ver Figura 8.

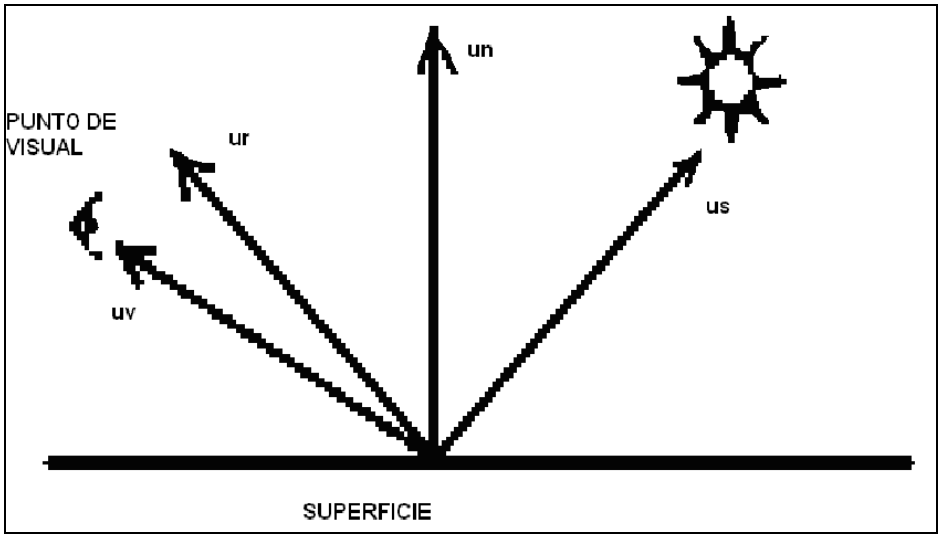


FIGURA 8. Modelo Phong

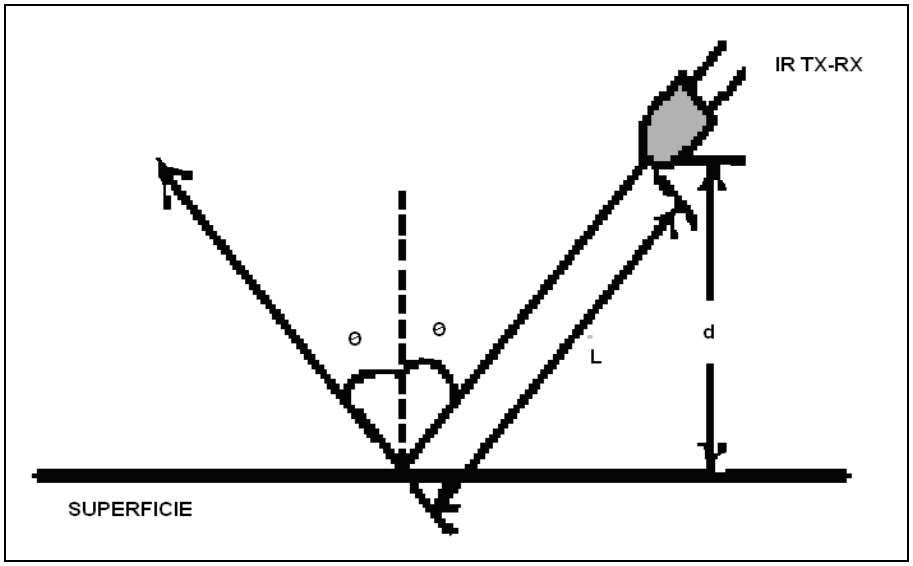


FIGURA 9. Incidencia de Haz Infrarrojo

Comparando el modelo de Phong mostrado en la Figura 8, con el con el haz emitido por un sensor de infrarrojos mostrado en la Figura 9 se puede deducir que:

$$\mu s * \mu n = Cos\theta, \rightarrow (29)$$

También, si se asume que tanto el emisor como el receptor de infrarrojos están en el mismo punto:

$$\mu v \approx \mu s, \rightarrow (30)$$

$$(\mu s * \mu v) = Cos2\theta, \rightarrow (31)$$

Así, la ecuación de intensidad en el receptor se puede escribir como:

$$I = C_0 * Cos\theta + C1 * Cos^n(2\theta) + C2, \rightarrow (32)$$

Por otra parte, también se sabe que la energía *E* absorbida por el fotodiodo y que se transformará en un voltaje de salida, es una función de la intensidad *I*, de la distancia viajada 2*L*, y del área *A* del fotodiodo.

$$E = (I \cdot A)/(2 \cdot L)^2 = (C_0 * Cos\theta + C1 * Cos^n(2\theta) + C2) \cdot A/(2 \cdot d/Cos\theta)^2, \rightarrow (33)$$

$$E = A \cdot Cos^2\theta \cdot (C_0 * Cos\theta + C1 * Cos^n(2\theta) + C2)/4 \cdot d^2, \rightarrow (34)$$

Asumiendo que la superficie reflectora es un reflector pasivo, entonces se puede establecer que *C*2 = 0 y que *n* = 1. También, ya que *A* es una constante, se puede integrar al resto de los coeficientes, por lo que el modelo de energía absorbida por el receptor de infrarrojos queda como:

$$E = Cos^2\theta \cdot (C_0 \cdot Cos\theta + C1^1 \cdot Cos2\theta)/4d^2, \rightarrow (35)$$

Aquí, *C*o' y *C*1' modelan las características integrada de la superficie reflectora y del receptor de infrarrojos; y dado que las mismas pueden ser determinadas a priori mediante la experimentación controlada en el entorno típico de trabajo del robot, entonces es el ángulo de incidencia la variable de mayor peso en la determinación de la distancia.

P. Novotny y N. Ferrier [17] hacen uso de un anillo de sensores de infrarrojos para determinar el ángulo de incidencia θ. Basándose en la amplitud de señal recogida por cada sensor, siguen una estrategia de búsqueda del máximo global el cual esta directamente relacionado con la dirección normal a la superficie reflectora.

Una vez determinada la orientación entre la superficie y el sensor más próximo al máximo, la distancia se determina mediante la lectura de la energía captada por el sensor y la ecuación,

$$d = \left(\sqrt{(C_0' \cdot Cos\theta + C1' \cdot Cos\theta)/E}\right) \cdot Cos\theta/2, \rightarrow (36)$$

P.4.1.3.5. TÉCNICA DE TRIANGULACIÓN USANDO PSD

En la medición de distancia por triangulación el haz de infrarrojo emitido viaja dentro del campo de visión hasta que es reflejado hacia el receptor por alguna superficie que se encuentre dentro de ese campo. De esta manera se establece un triángulo formado por el punto de emisión, el de reflexión y el de recepción. El ángulo de este triángulo en el punto de reflexión varia de acuerdo a la distancia a la que este se encuentre. La parte receptora de este tipo de sensores está formada por lentes de precisión que transmiten la luz reflejada hacia un arreglo *CCD Coupled Charge Device* lineal basado en el ángulo antes descrito. El arreglo *CCD* puede determinar el ángulo del haz reflejado y de esa manera determinar la distancia del sensor a la superficie receptora, ver Figura 10.

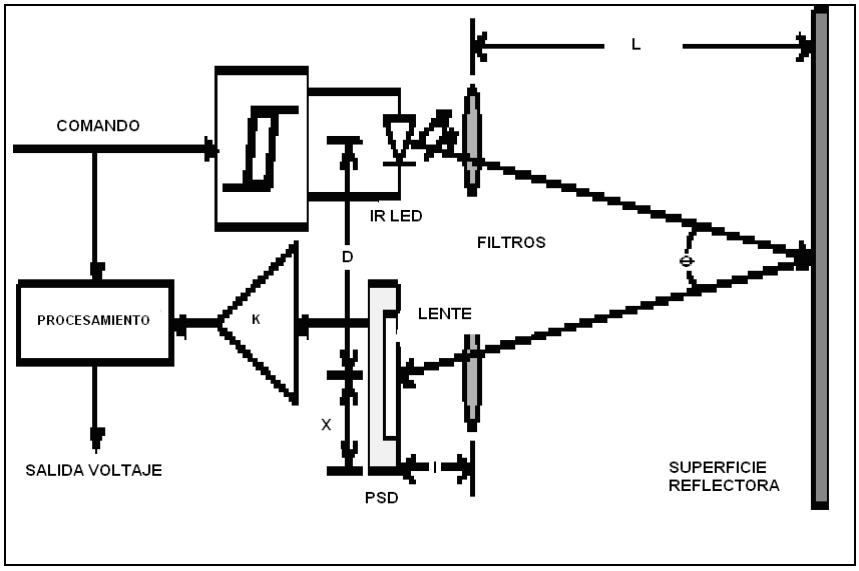


FIGURA 10. Principio de Operación del Medidor de Distancias

De esta forma, el elemento más importante en este tipo de sensor es el *PSD* (Detector Sensitivo a la Posición). El *PSD*, que puede entenderse como una lente situada sobre un arreglo de células sensibles a la luz infrarroja, proporciona una corriente de salida que varía en función de la posición *x* que sobre él alcanza el haz reflejado por el objeto. Es decir, dependiendo del ángulo de incidencia del haz reflejado hacia la lente del receptor se activa una u otra célula del arreglo, lo que permite estimar la distancia *L* a la que se encuentra el objeto mediante la ecuación:

$$L = f.d / x, \rightarrow (37)$$

Así, el sensor proporciona un voltaje analógico de salida que es proporcional a la distancia medida, y cuya característica se muestra en el Grafico 1.

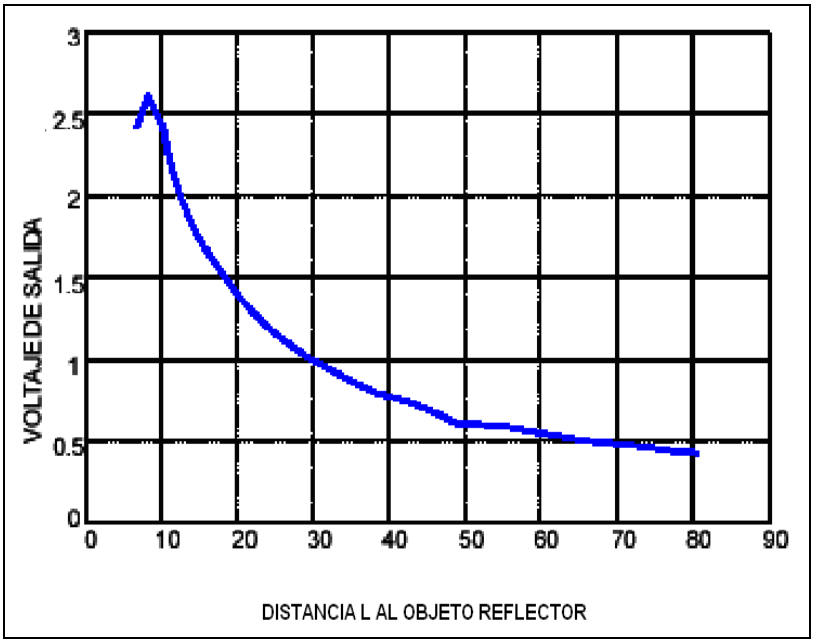


GRÁFICO 1. DISTANCIA CONTRA VOLTAJE DE SALIDA

P.4.2. ARQUITECTURAS DE CONTROL EN ROBÓTICA MÓVIL.

Una arquitectura para robot móvil se compone tanto del software como del hardware involucrado en el control del mismo. Los robots son sistemas complejos difíciles de desarrollar. Integran múltiples sensores y actuadores, tienen muchos grados de libertad y deben tratar con tareas de tiempo real. Sobre los elementos que debe poseer un robot, R. R. Murphy dice [23]:

En Inteligencia Artificial ha quedado establecido que cualquier robot autónomo tiene que poseer las siguientes tres primitivas o elementos funcionales:

a. SISTEMA DE PERCEPCIÓN: Capacita al robot para disponer de la información de los sensores y producir, a partir de ésta, información útil para los otros elementos funcionales.
b. SISTEMA DE RAZONAMIENTO O PLANIFICACIÓN: Capacita al robot para producir las tareas a ejecutar (ir a la entrada, girar a la derecha, moverse 2 metros y parar, etc.) en base a la información sensorial y conocimiento disponible del mundo que le rodea o entorno.
c. SISTEMA DE EJECUCIÓN O ACCIÓN**:** Capacita al robot para la ejecución de tareas mediante la interacción con los actuadotes.

Actualmente, la organización de la autonomía o inteligencia en robots móviles se aborda desde tres paradigmas diferentes: DELIBERATIVO, REACTIVO e HÍBRIDO. Cada una de estas tendencias puede describirse atendiendo a dos factores:

1. La relación existente entre las tres primitivas funcionales comunes para cualquier robot: PERCEPCIÓN, RAZONAMIENTO y EJECUCIÓN. En la Figura 11 se describen estas tres primitivas en términos de los posibles valores de entrada y de salida. En función de los valores de entrada y salida, se establecen distintas relaciones entre las primitivas.

2. El mecanismo de procesamiento y distribución de la información sensorial a través del sistema. Hace referencia a cuánto o cómo los sentidos influyen en una persona, robot o animal. En algunos paradigmas, la información sensorial se usa de una manera específica para cada función del robot y además de forma local. En otros paradigmas, por el contrario, la información sensorial se usa para la creación de un modelo global del mundo que luego es utilizado de forma distribuida por las distintas funciones del robot.

La Figura 11 define las primitivas funcionales de un robot móvil en términos de las entradas y las salidas:

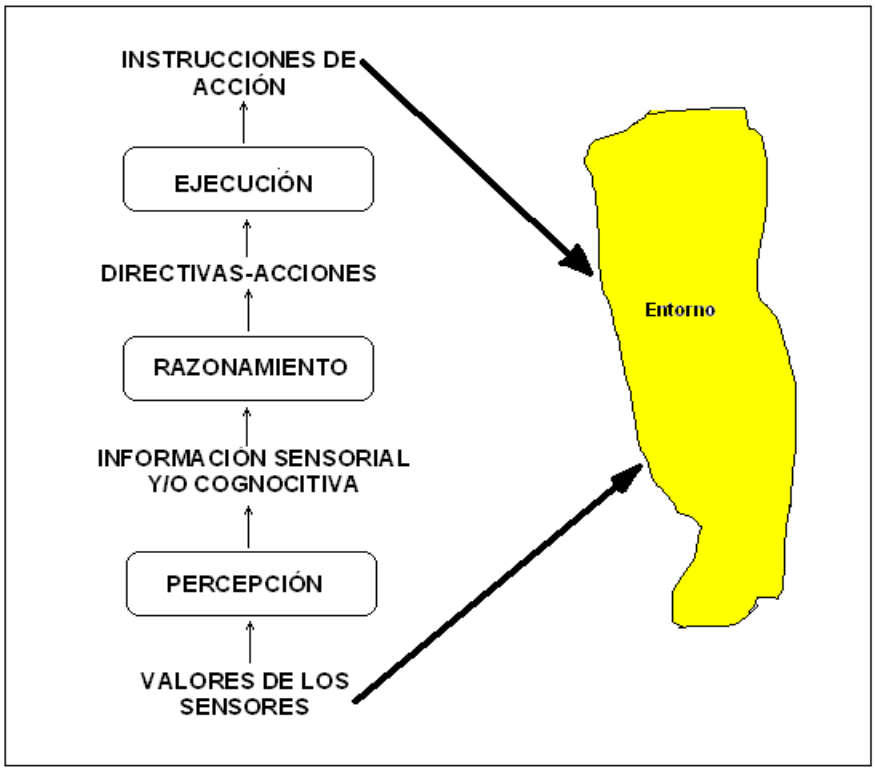


FIGURA 11. Definición de las Primitivas Funcionales de un Robot Móvil en Términos de las Entradas y las Salidas.

A continuación se describen los tres paradigmas en base a estos factores.

- PARADIGMA DELIBERATIVO**.**

Según el paradigma deliberativo, un robot autónomo primero tiene que percibir su entorno, después planear la acción a realizar y finalmente ejecutar dicha acción (S-P-A: sentir-planear-actuar). De esta forma, el ciclo tiene que repetirse hasta completar cada una de las tareas del robot. Ver Diagrama 1.

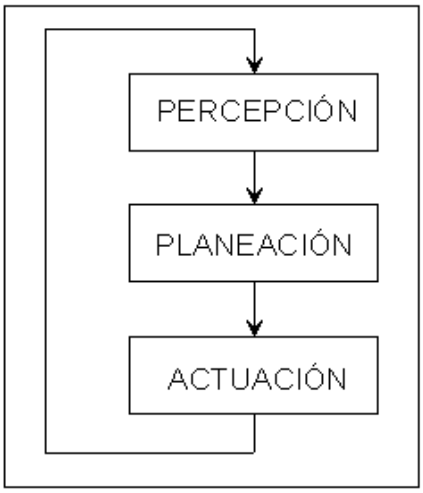


DIAGRAMA 1. Paradigma Deliberativo: Percepción- Planeación- Actuación

En el paradigma deliberativo se planifica cada una de las acciones del robot. El paradigma deliberativo se caracteriza por intentar obtener y utilizar un modelo interno del mundo real lo más exacto y completo posible (basado en representaciones simbólicas), de manera que el procesamiento de la información sensorial se orienta a la consecución de este objetivo.

Este principio es su fortaleza, pues si se asume la existencia de un modelo interno totalmente completo y correcto, también puede asumirse que el procesador de dicho modelo podrá planear y razonar sobre el entorno a cualquier nivel deseado.

- PARADIGMA REACTIVO.

Presenta una organización Sentir-Actuar (S-A). El paradigma reactivo asume que la entrada que origina una acción siempre es la salida directa de un sensor, mientras que el paradigma deliberativo asume que la entrada que origina una acción es siempre el resultado de un razonamiento o plan. Ver Diagrama 2.

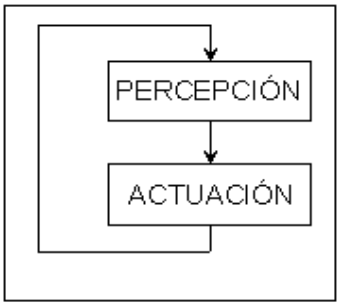


DIAGRAMA 2. . Paradigma Reactivo: Percibir-Actuar

Este paradigma se basa en la existencia de múltiples procesos Sentir-Actuar, (comportamientos), que se ejecutan concurrentemente. Un comportamiento toma la salida de un sensor y genera una acción a realizar independientemente de lo que realicen los otros comportamientos.

El término comportamiento se ha definido como una correspondencia de los valores de los sensores con el conjunto de acciones a realizar para alcanzar un objetivo, siendo el componente principal de la inteligencia en la mayoría de los sistemas.

Por ejemplo, puede haber un comportamiento que dirija al robot en línea recta para alcanzar un objetivo y al mismo tiempo otro que lo haga girar 90º para esquivar un obstáculo, el resultado final consistirá en que el robot hará una combinación de ambos comportamientos resultando un movimiento hacia adelante con giro de 45º. El resultado final emerge de la combinación de ambos comportamientos, lo que Simó [24] denomina, comportamiento emergente.

La aproximación basada en comportamientos, en su forma más pura, no utiliza modelos internos simbólicos de representación del mundo real. Según Brooks [25] el problema de incertidumbre no puede resolverse mediante el uso de modelos. La solución consiste en la utilización de sistemas basados en comportamientos donde, en lugar de modelos, es el propio entorno el que proporciona cualquier información necesaria.

La ventaja de estos sistemas, debido a que no utilizan modelos internos simbólicos y a que son totalmente reactivos (no existe la planificación o razonamiento), consiste en que no necesitan realizar accesos y procesamiento de estructuras de representación por lo que son más rápidos y predecibles que los sistemas que sí tienen que hacerlo. Por ello, pueden utilizarse para cumplir restricciones de tiempo real estricto, a diferencia de los sistemas deliberativos donde la operación en tiempo real se presenta como un problema. Además, en los sistemas reactivos tampoco existen los problemas de hipótesis del mundo cerrado y problema del marco, característicos del paradigma deliberativo, puesto que reaccionan directamente a todo aquello que es perceptible y no confían en creencias y predicciones.

El principal problema del paradigma reactivo es el problema sensorial: el considerar únicamente la información sensorial actual para generar las acciones de control, puede ser problemático debido a la incertidumbre procedente tanto de los sensores como de los actuadores y a factores físicos como rozamientos ó deslizamientos.

Los sistemas realizados han tenido gran éxito, construyéndose robots que imitan conductas aparentemente inteligentes de animales e insectos. Sin embargo, estos sistemas no permiten el desarrollo de robots que realicen tareas complejas al no disponer de ningún sistema de razonamiento.

Un sistema sin ningún modelo interno que no es capaz de planificar operaciones de forma eficiente, está en desventaja frente al mismo sistema con dicho modelo.

Además, es necesario optimizar la representación del modelo dado el tamaño de memoria disponible en comparación con la complejidad del mundo. Por otro lado, actuar sin almacenar ningún tipo de información sería un grave error, puesto que los beneficios de poder disponer de algún tipo de información almacenada de una forma coherente o modelo interno son varios, principalmente la habilidad de predecir sucesos en el entorno antes de que éstos ocurran usando la información de las experiencias pasadas y permitiendo ir por delante de la información sensorial actual. Además, el uso de representaciones internas puede reducir el cómputo necesario cuando, por ejemplo, se reconoce un lugar familiar, en dicho caso podría no ser necesario volver a realizar su inspección.

“El mecanismo de procesamiento y distribución de la información sensorial a través del sistema, hace referencia a cuánto o cómo los sentidos influyen en una persona, robot o animal. En algunos paradigmas, la información sensorial se usa de una manera específica para cada función del robot y además de forma local (Paradigma Reactivo). Por el contrario, en otros paradigmas, la información sensorial se usa para la creación de un modelo global del mundo que luego es utilizado de forma distribuida por las distintas funciones del robot (Paradigma Deliberativo)”.

- PARADIGMA HÍBRIDO

El paradigma híbrido deliberativo/reactivo, se basa fundamentalmente en el paradigma reactivo por su rápida velocidad de ejecución y consiguiente adecuación a los sistemas de tiempo real estricto, pero también incorpora un nivel de razonamiento basado en modelos internos.

Según el paradigma híbrido, el robot primero planea (razonamiento) la mejor forma de descomponer la tarea a realizar en varias sub-tareas, y selecciona los comportamientos adecuados para alcanzar cada sub-objetivo. Posteriormente, el robot empieza con la ejecución, de la misma forma que en el paradigma reactivo. Este tipo de organización puede denotarse por “Planear, Sentir-Actuar” (P, S-A), donde la coma indica que en un primer paso se realiza la planificación y posteriormente se ejecutan los comportamientos (Sentir-Actuar).

La organización y procesamiento de la información sensorial en el paradigma híbrido también es una mezcla de los paradigmas deliberativo e híbrido. Esta información es utilizada tanto a nivel de planificación, para la construcción de un modelo interno del entorno del robot que se utiliza para la descomposición de tareas y selección de comportamientos, como a nivel de

ejecución de dichos comportamientos, el valor de cada sensor es utilizado por cada comportamiento que lo necesite.

P.4.3. TEORÍA DE ESQUEMAS

Los esquemas han sido utilizados por los psicólogos como un medio de expresar la unidad básica de actividad. Un esquema consiste en el conocimiento de cómo actuar y/o percibir, así como del procesamiento necesario para realizar una actividad. Arbib [26] y Larkin [27] definen en la "Teoría de Esquemas" aplicada a la robótica móvil que, un esquema de comportamiento se compone de al menos un esquema motor y un esquema de percepción, además del conocimiento necesario para coordinar múltiples esquemas. Esta separación entre acción y percepción, implica una organización de los comportamientos mediante la combinación de esquemas motores y esquemas de percepción, dando lugar a un modo de implementación. La percepción en los comportamientos sigue dos reglas: como activadora de los comportamientos, o como guía de los mismos.

P.4.4. TIPOS DE PLATAFORMAS MÓVILES

El primer paso que se da en la construcción de la plataforma móvil es la elección de su configuración, esto es, definir la distribución de los principales elementos que lo componen:

- Ruedas.
- Motores.

La precisión de las mediciones que haga la plataforma móvil, dependerá en gran medida de la configuración de sus elementos anteriores.

La elección de la plataforma móvil es sin duda una parte fundamental a la hora de diseñar; dependiendo de las necesidades de la aplicación se busca una plataforma que posea características que cubran estas necesidades, dentro de estas se tiene:

- Respuesta ante cualquier superficie.
- Velocidad.
- Maniobrabilidad.
- Equilibrio.

En relación a las ruedas, existen distintas configuraciones, típicamente utilizadas en robótica móviles:

- Diferencial.
- Triciclo.
- Ackerman.
- Sincronizada.
- Omnidireccional, con múltiples grados de libertad.
- Movimiento mediante orugas.

P.4.4.1. CONFIGURACIÓN DIFERENCIAL.

La configuración diferencial se presenta como la más sencilla de todas. Consta de dos ruedas situadas diametralmente opuestas en un eje perpendicular a la dirección del robot. Cada una de ellas irá dotada de un motor, de forma que los giros se realizan dándoles diferentes velocidades. Así, si se quiere girar a la derecha, dará mayor velocidad al motor izquierdo.

Para girar a la izquierda, será el motor derecho el que posea mayor velocidad. Con dos ruedas es imposible mantener la horizontalidad del robot. Se producen cabeceos al cambiar la dirección. Para solventar este problema, se colocan ruedas "locas". Estas ruedas no llevan asociadas ningún motor, giran libremente según la velocidad del robot como se muestra en la Figura 12. Además, pueden orientarse según la dirección del movimiento, de forma análoga a como lo hacen las ruedas traseras de los carritos del supermercado. Dependiendo de las necesidades, se pueden colocar una, dos o más ruedas "locas" [28].

Sin embargo, la presencia de más de tres apoyos en el robot (incluidas las dos ruedas de tracción), puede llevar a graves cálculos de odometría en terrenos irregulares, e incluso a pérdida de tracción total. En la Figura 12, se aprecia cómo la rueda de tracción pierde agarre, haciendo imposible el avance del robot:

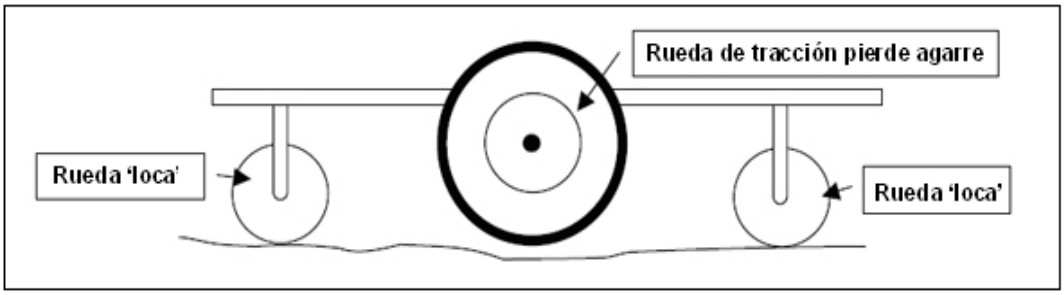


FIGURA 12. Falta de Tracción Debido a la Rueda Loca [3]

[3] Figura 12
(*www.disam.upm.es/cybertech/Nacional/Documentos/Otros/configuracionesmovimiento.pdf)*

Se puede resumir la configuración diferencial con dos ruedas con tracción independiente, y una o más ruedas locas. Con una rueda loca se ve en la Figura 13.

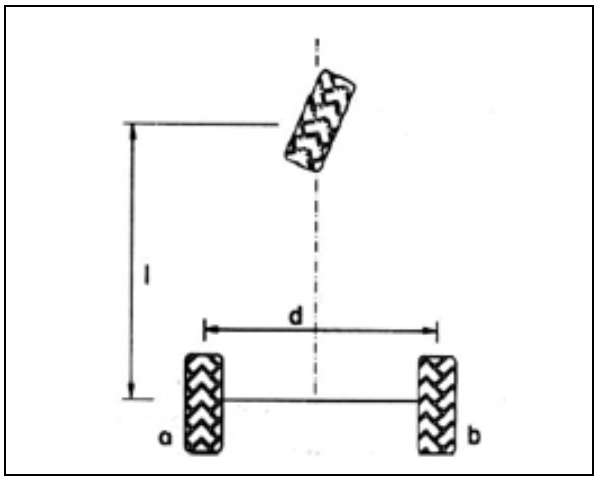


FIGURA 13. - Ruedas en Configuración Diferencial con una Rueda Loca al Frente[4]

Para llevar a cabo una navegación por odometría, es necesario acoplar a los motores de las ruedas laterales dispositivos llamados "*encoders*", de forma que contando los datos entregados por los mismos se conoce el avance de cada rueda y teniendo en cuenta el radio de la misma y la reducción del motor, sólo hay que aplicar las ecuaciones cinemáticas del robot para hallar la posición exacta en la que se encuentra y el ángulo de desviación respecto a una dirección de referencia [29].

P.4.4.2. CONFIGURACIÓN EN TRICICLO

En este caso, se dispone de tres ruedas en el robot, situadas de forma similar a los triciclos de los niños, de ahí su nombre. Se Tendrá por tanto, dos ruedas traseras, que no llevan acopladas ningún motor. La tracción estará en la rueda delantera, que además, será la que se usa para dirigir al robot. Ver Figura 14.

Para esta configuración, el cálculo de la odometría es mucho más sencillo. La posición del robot vendrá dada por el número de pulsos que avanza el *encoder* de la rueda motora, y la dirección es simplemente la que lleve dicha rueda.

Un problema asociado a esta configuración es que el centro de gravedad tiende a alejarse de la rueda de tracción en terrenos inclinados cuando el robot lleva la dirección de subida. Esto se

[4] Figura 13
(*www.disam.upm.es/cybertech/Nacional/Documentos/Otros/configuracionesmovimiento.pdf)*

traduce en una pérdida de la tracción del robot. Al perderse el contacto con el suelo la rueda motora sigue girando, pero el robot no avanza. Esto supone un error grande al hacer el cálculo de la odometría, ya que el robot indica que está en un punto más avanzado, cuando en realidad se encuentra más atrás [30].

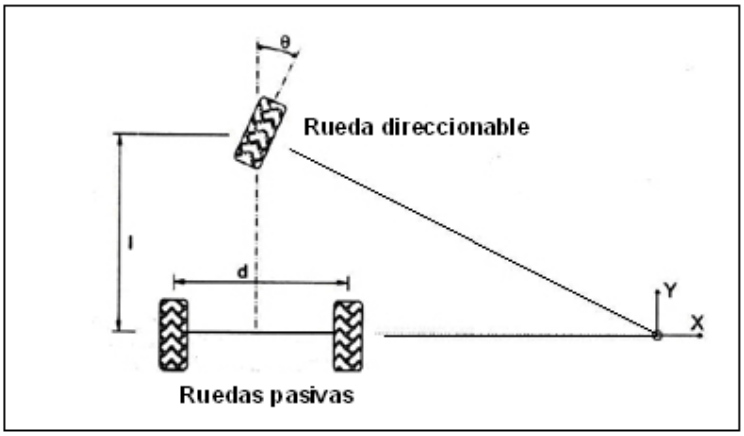


FIGURA 14- Configuración Típica en Triciclo[5]

P.4.4.3. CONFIGURACIÓN ACKERMAN

Se usa casi exclusivamente en la industria del automóvil. Es la configuración que llevan los autos: dos ruedas con tracción traseras, y dos ruedas de dirección delanteras. Ésta configuración está diseñada para que la rueda delantera interior en un giro tenga un ángulo ligeramente más agudo que la exterior, y evitar así el derrape de las ruedas.

Como se puede apreciar en la Figura 15, las normales a ambas ruedas se cortan en un punto, que se encuentra sobre la prolongación del eje de las ruedas traseras. Así, se puede comprobar que las trayectorias de ambas ruedas para ángulos de giro constantes son circunferencias concéntricas.

La relación entre los ángulos de las ruedas de dirección viene dada por la ecuación de Ackerman:

$$\cot(\theta 1) - \cot(\theta 2) = d / l \quad (1)$$

[5] Figura 14
(*www.disam.upm.es/cybertech/Nacional/Documentos/Otros/configuracionesmovimiento.pdf)*

Donde:

θ1 = Ángulo relativo de la rueda interior
θ2= Ángulo relativo de la rueda exterior
l = Separación longitudinal entre ruedas
d = Separación lateral entre ruedas

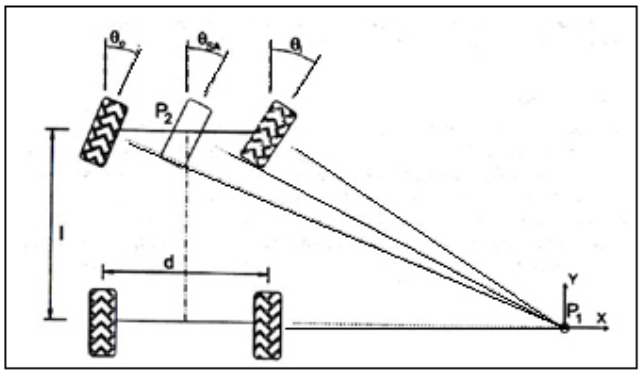


FIGURA 15- Configuración Típica Ackerman[6]

La configuración de Ackerman da una solución bastante precisa para la odometría a la vez que constituye un buen sistema de tracción incluso con terrenos inclinados. No obstante, la construcción mecánica de un robot con configuración Ackerman se complica de forma exponencial respecto a las anteriores. Por otro lado, el robot propuesto va a operar en interiores de edificios, con lo que el terreno no va a presentar dificultades como para necesitar una configuración tan compleja.

P.4.4.4. DIRECCIÓN SINCRONIZADA.

Supone una configuración innovadora. Consiste en tres o más ruedas, todas ellas dotadas de tracción y acopladas mecánicamente, de forma que todas rotan en la misma dirección y a la misma velocidad. Se necesita que todas ellas pivoten de la misma manera al cambiar la dirección. Este sistema necesita de una gran sincronización, que redunda en una odometría mejorada reduciendo el deslizamiento de las ruedas respecto al suelo, ya que todas las ruedas generan fuerzas con vectores de igual módulo y paralelos en todo momento.

En la Figura 16 se puede apreciar la complejidad mecánica que requiere una configuración de este tipo.

[6] Figura 15
(*www.disam.upm.es/cybertech/Nacional/Documentos/Otros/configuracionesmovimiento.pdf)*

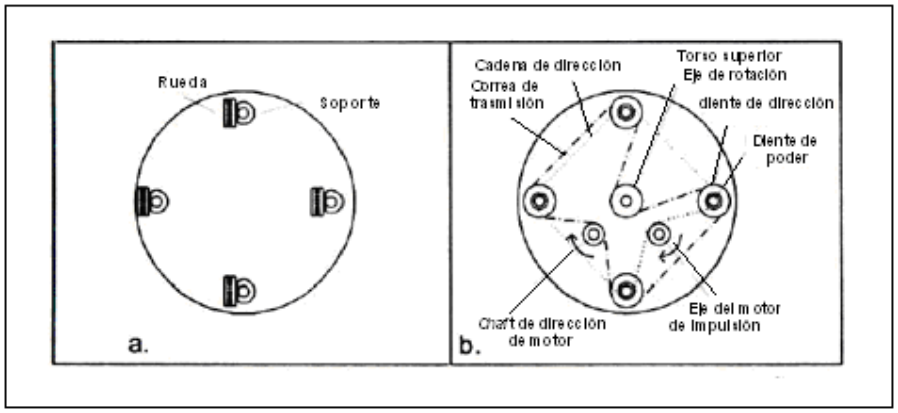


FIGURA 16- Mecánica para Diseñar una Configuración Sincronizada[7]

P.4.4.5. CONFIGURACIÓN OMNIDIRECCIONAL

Se trata de dotar al robot con ruedas omnidireccionales. El precio de estas ruedas hace prohibitivo su uso en el proyecto. Además, suponen una enorme complicación en el cálculo de la odometría [31], como se muestra en la Figura 17.

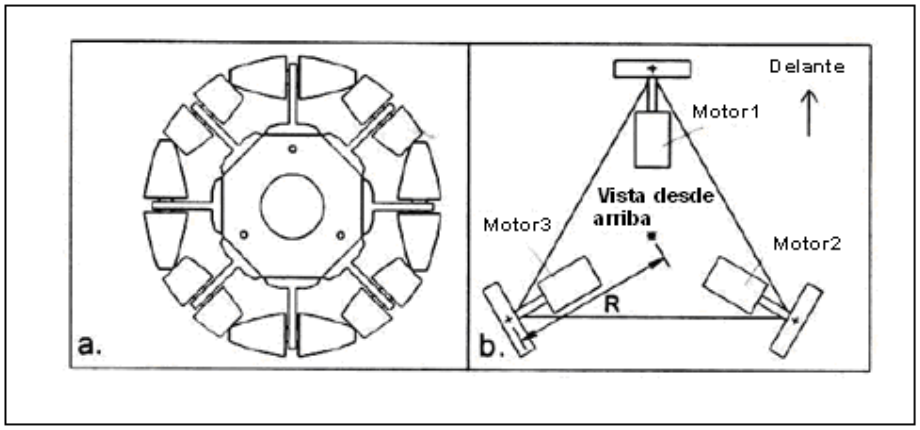


FIGURA 17. a. Rueda Omnidireccional, b. Ejemplo de Configuración Omnidireccional[5]

[7] Figuras 16,17 (*www.disam.upm.es/cybertech/Nacional/Documentos/Otros/configuracionesmovimiento.pdf)*

P.4.4.6. VEHÍCULOS CON MÚLTIPLES GRADOS DE MOVILIDAD.

Este tipo de configuraciones se plantean para mejorar la movilidad del vehículo. Suelen ser robots articulados donde cada una de las articulaciones consiste en módulos con tracción diferencial

P.4.4.7. TRACCIÓN MEDIANTE ORUGAS

Se trata de sustituir las ruedas por orugas. Es un caso particular de la tracción diferencial. Sin embargo, en esta configuración el deslizamiento en los giros es muy grande, perdiéndose bastante precisión en el cálculo odométrico. Se emplea en casos en los que el terreno presente irregularidades. Por estas dos razones, se descarta el uso de orugas para el proyecto.

P.4.5. MODELO DE CONTROL GUIADO DIFERENCIAL.

Para el desarrollo del proyecto se elige la configuración de disposición de ruedas con guiado diferencial debido a que se facilita su diseño y construcción, además el error odométrico visto en trabajos anteriores es óptimo para la aplicación.

Velocidad lineal

Se presenta el modelo de control para la configuración que utiliza la plataforma [3]

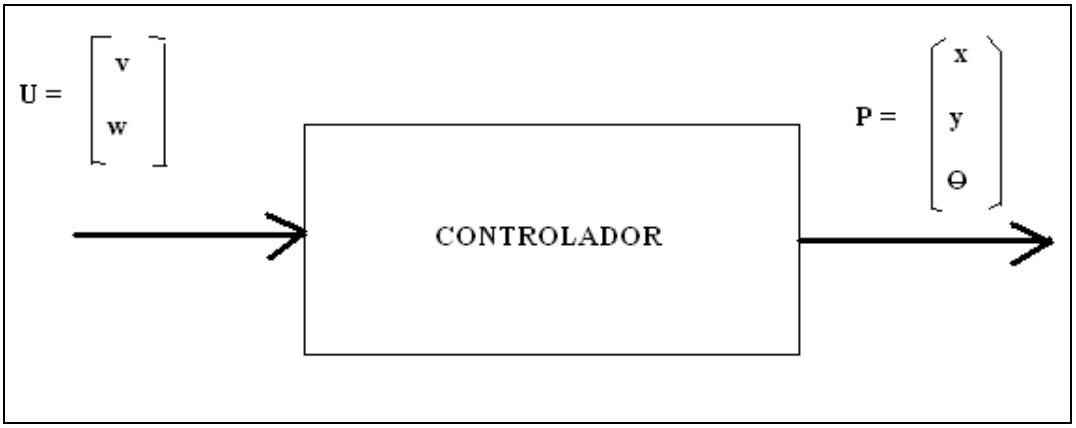


DIAGRAMA 3. Modelo de Control Guiado Diferencial

$v = \Delta S, (\text{VELOCIDAD LINEAL}) \rightarrow (2)$

$w = \Delta \varphi, (\text{VELOCIDAD ANGULAR}) \rightarrow (3)$

$$v = (Vr + Vi)/2, \rightarrow (4)$$

$$w = (Vr - Vi)/2, \rightarrow (5)$$

$$\left\langle \begin{vmatrix} x \\ y \\ \theta \end{vmatrix} = \begin{vmatrix} \cos\ \theta & 0 \\ sen\ \theta & 0 \\ 0 & 1 \end{vmatrix} \begin{vmatrix} v \\ w \end{vmatrix} \right\rangle \rightarrow \quad (6)$$

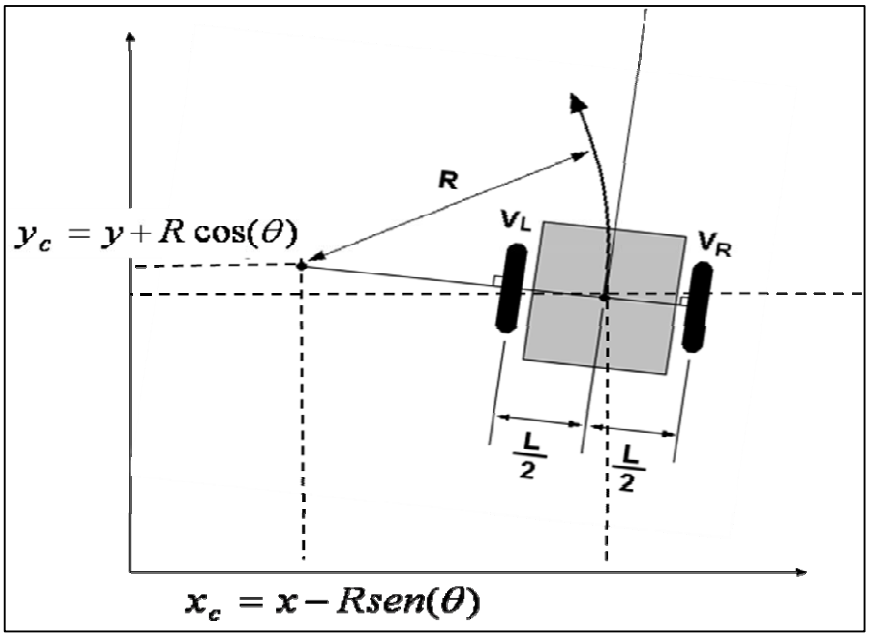


FIGURA 18. Plataforma Posicionada en un Plano Coordenado[8]

Para poder controlar la plataforma móvil se necesita estimar la posición Relativa (odometría), se calcula la distancia a partir de condiciones iniciales, midiendo el desplazamiento de las ruedas.

P.4.5.1. ODOMETRÌA

Es el estudio de la estimación de la posición de un vehículo en función de la rotación de las ruedas. Este nombre proviene de la unión de las palabras griegas hodos (viaje) y metrón (medida). Es también utilizado este término para describir la distancia recorrida por un vehículo.

P.4.5.2. TRACCIÓN DIFERENCIAL

En el cálculo de la odometría en plataformas móviles basados en tracción diferencial, el movimiento se consigue con dos ruedas, acopladas cada una a su propio motor, teniendo normalmente una o dos ruedas de apoyo, es el mismo sistema que se encuentra en una silla de ruedas.

El movimiento se consigue aplicando más o menos velocidad a cada una de las ruedas motrices. De esta forma la plataforma avanza en línea recta fijando ambos motores a la misma

[8] Figura 18 *(www.*omarsanchez.net/Documents/Cinematica_vehículos.ppt*)*

velocidad, puede girar en una u otra dirección aplicando velocidades diferentes, y haciéndolas girar en sentido inverso, la plataforma girará sobre su eje.

Esta claro que el control de trayectoria se basa en la diferencia de velocidades entre las ruedas. A continuación de expresa matemáticamente el funcionamiento del sistema, y cómo con estas fórmulas se puede hacer algoritmos que controlen de forma precisa la plataforma [30].

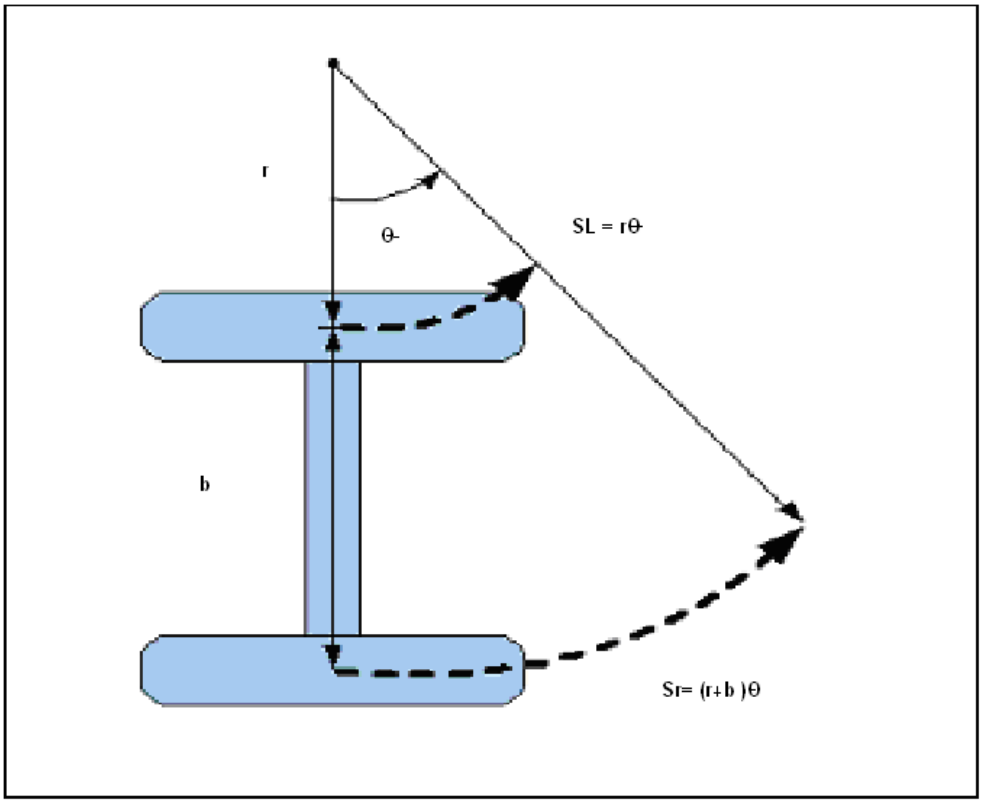


FIGURA 19. Tracción Diferencial [9]

Se puede ver el modelo esquemático de la tracción diferencial mientras esta girando en la Figura 19, y cuales son los parámetros que afectan al giro. b es la distancia entre ejes y r es la distancia de plataforma al centro de giro. Con estos parámetros básicos, se puede establecer una serie de relaciones:

$$SL = r\theta, \rightarrow (7)$$

$$Sr = (r + a), \rightarrow (8)$$

$$Sc = (r + a/2) * \theta, \rightarrow (9)$$

[9] Figura 19 (http://www.itoosoft.com/motorolos/odometria/odometria.html)

Donde *SL* es la distancia recorrida por la rueda izquierda y *Sr* es la recorrida por la rueda derecha. También se tiene a *Sc* que es la distancia recorrida por el eje central de la plataforma, que se toma como el punto medio entre los ejes de las ruedas. En todos los casos es el ángulo θ de giro expresado en radianes (360° = 2p radianes).

P.4.5.2.1. ANÁLISIS DEL RADIO DE GIRO

Una vez se tiene descrito en términos matemáticos los parámetros que afectan al giro, se pasa a analizar cual es la trayectoria que seguirá. Esto sólo es válido si se mantiene una suposición, la velocidad de las ruedas debe ser constante en todo momento. Si las ruedas varían su velocidad, la plataforma ya no traza una trayectoria circular, sino puede tomar formas más complejas. Después se verá que es posible introducir datos de aceleración al modelo, aunque por el momento se ignoran para mantener el modelo sencillo de analizar. Tomando como base la ecuación (7) y (8) se puede construir un sistema de ecuaciones:

$$SL = r\theta, \rightarrow (10)$$
$$Sr = (r + a), \rightarrow (11)$$

En este sistema, se tiene relacionados tanto el avance de las ruedas, como el centro *r* y el ángulo *θ* que describen el giro. Si en este sistema, se despejan las variables *θ* y *r*, se obtienen las ecuaciones válidas para calcular el radio de giro a partir únicamente de la información de los *encoders*:

$$\theta = (SL - Sr)/b, \rightarrow (12)$$

$$r = (Lr * b)/(SL - Sr), \rightarrow (13)$$

Aunque aún falta un poco para poder definir la trayectoria, ya que estas ecuaciones tan sólo describen el giro respecto al extremo interior de la plataforma y no están definidas según coordenadas cartesianas del plano. Si pueden dar algunos detalles importantes sobre el comportamiento de la tracción diferencial, que quizás se conocen de forma intuitiva, pero que no se tiene la demostración matemática [30].

P.4.5.2.2.GIRO SOBRE SÍ MISMO

En el caso que ambas ruedas avancen a la misma velocidad, pero en sentido contrario, se obtiene una distancia de *-b/2* , por lo que el centro de giro está sobre el centro del eje (el centro de coordenadas es la rueda mas cercana al centro de giro). Y el ángulo *θ*, proporcional a la distancia recorrida, es decir, condiciones en las que la plataforma móvil gira sobre si misma pero sin avanzar.

$$SL = -Sr * \theta = (SL - Sr)/b = 2S/r, \rightarrow (14)$$

$$r = \frac{(Lr * b)}{(SL - Sr)} = \frac{-b}{2}, \rightarrow (15)$$

P.4.5.2.3. AVANCE RECTO

Si ambas ruedas avanzan la misma distancia, se tiene un ángulo 0°, es decir, la plataforma no gira en ningún momento y un radio de giro infinito (∞), puesto que se interpreta una línea recta como el arco de un círculo de radio infinito, o simplemente que no existe centro de giro. En cualquier caso es obvio que ha avanzado en línea recta una distancia *S*. Puede seguirse manipulando estas ecuaciones, para realizar giros y cambios de trayectoria, pero la idea ya está expuesta. Con esto ya podría ser suficiente para programar algunos comportamientos básicos basados en odometría, aunque se sigue avanzando y ver como calcular una trayectoria posterior del movimiento.

$$SL = -Sr * \theta = (SL - Sr)/b = 0, \rightarrow (16)$$

$$r = \frac{(Lr * b)}{(SL - Sr)} = \infty, \rightarrow (17)$$

P.4.6. MAPAS DE ENTORNO

La construcción de un mapa del entorno se había considerado una de las labores básicas que un robot móvil debía llevar a cabo dada su necesidad de tener un modelo del entorno. De esa forma había un bucle de control básico consistente en utilizar los sensores para percibir el entorno, y posteriormente utilizar dicho modelo para planificar las acciones. El principio adoptado por los científicos de la línea activista es "Usar el mundo como su propio modelo" [31]. De la forma más sencilla se podría pensar que un modelo es un conjunto de suposiciones (con mayor o menor duración temporal, precisión y complejidad en la representación) que el robot móvil hace sobre su entorno. Pero bajo esta premisa todos los robots, incluso los más reactivos usan un modelo del mundo. En el robot de Braintenberg que sigue la luz [32], el modelo del mundo asume que existen luces y que es bueno seguirlas. Una forma más elaborada de definir el modelo del mundo sería aquella proveniente de la definición de la representación del entorno de una forma matemática de la forma en que Gallistel [33] usa en la "neurocomputación": un robot está representando un aspecto de su entorno cuando existe una función isomórfica entre ese aspecto y un comportamiento del robot que adapta el comportamiento del robot al mismo.

Si por ejemplo el robot móvil tiene una representación del mundo en forma de *grids (*rejilla*)*, existe un comportamiento que se adapta a esa característica (ocupación o no de las celdas) para evitar obstáculos o moverse de un punto a otro. De esta forma el modelo que se tiene, pasa a formar parte del estado del robot, ya que actúa como una variable que condiciona su comportamiento. La diferencia principal entre los robots puramente reactivos y aquellos puramente deliberativos se encuentra en el tiempo que transcurre entre percepción y la acción. En el caso reactivo este tiempo es mínimo dado el enlace que se hace entre sensores y actuadores mediante los comportamientos. En el caso deliberativo este tiempo puede ser muy grande en función del sistema de razonamiento utilizado. Poco a poco van surgiendo otros modelos basados en sistemas biológicos más cercanos a las sensaciones y actuaciones.

La clasificación que se ha realizado clásicamente de los distintos tipos de mapas de representación del entorno viene dada en función de la cantidad de información geométrica entre los elementos que forman dicho mapa. En cualquier caso ésta clasificación no es estricta y puede ocurrir que determinadas implementaciones recaigan en más de una clase.

P.4.6.1. MAPAS DE LOCALIZACIONES

Los mapas de localizaciones son aquellos donde no existe información geométrica entre los elementos del mismo, sino una lista de localizaciones o lugares (marcas) que pueden ser reconocidas por el robot y/o navegador, en las Figuras 20 y 21 se muestran dos mapas de localización, el de la Figura 20 es el reconocido por un robot y el de la Figura 21 es el que lee y procesa un ser humano.

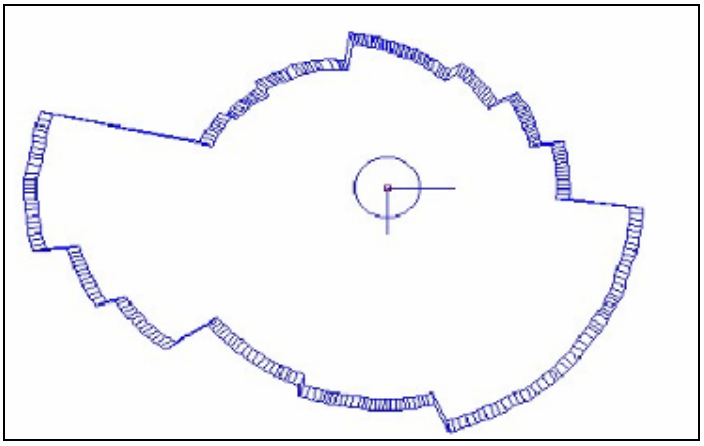


FIGURA 20. Mapa de Localización de Robot Móvil.

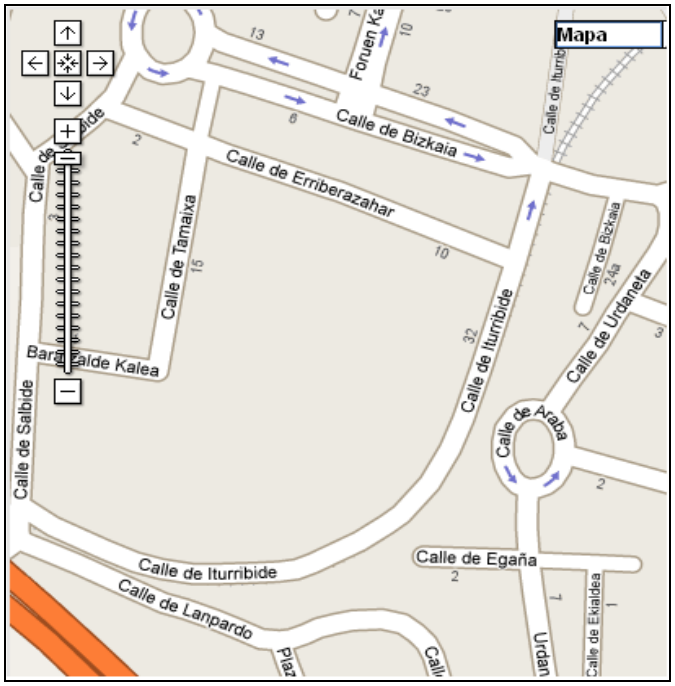


FIGURA 21. Mapa de Localización Urbano.[10]

[10] Figura13 (http://www.laterooms.com/es/hotel-directions/146108_apartamentos-turisticos-talaimendi-zarautz.aspx)

Esta capacidad de reconocer lugares es propia también de las personas y se utiliza frecuentemente como forma de orientarse en ciudades y lugares conocidos. Ésta es la forma en que las personas se mueven en la oscuridad y llevan las manos por delante para localizar objetos, no tropezar y orientarse. Kuipers y Byunt [34], utilizan el concepto de "*distinctive places*". En el cual, determinadas propiedades de las medidas son definidas como medidas diferenciadoras, de forma que se trata de maximizar esta característica entre los "*distinctive places".* En general ésta es la estrategia adoptada en estos mapas. Donett [35], en su trabajo utiliza una combinación de datos provenientes de distintos sensores (ultrasonidos, infrarrojos, sonido) midiendo intensidad, dirección y distancia. Con lo cual el robot es capaz de identificar los lugares en los que ya ha estado mediante las medidas almacenadas y las que percibe. Este proceso de ajuste tiene lugar mediante la utilización de probabilidad Bayesiana.

Mataric [5] muestra los resultados de un trabajo donde se combina la información del movimiento con la de sensores. Por ejemplo: un pasillo es una combinación de movimiento hacia delante e información de sensores de proximidad por los lados de cercanía de pared. También en éste caso se utiliza una brújula para orientar los lugares que son marcas (pasillo orientado al norte).
Como se ve, estos mapas son adecuados cuando la capacidad sensorial del robot le permite identificar las marcas, si estas son diferenciables y visibles en el espacio de trabajo. Si esta última condición no se cumple el robot necesitará pasar por marcas intermedias para llegar a otras no visibles. Esto supone el añadir al mapa información de qué marcas están unidas o son vecinas. Esta información es la que suministra un mapa topológico.

P.4.6.2. MAPAS TOPOLÓGICOS

Dado que la condición expresada en el apartado anterior (visibilidad) no suele cumplirse en un entorno más o menos real, muchos de los trabajos comentados anteriormente son ampliados para incluir la información de adyacencia entre marcas.

Un enlace entre dos marcas supone que el robot puede ir de una marca a otra. Este enlace se puede añadir sólo si el robot lo ha realizado previamente. En muchas ocasiones el proceso de establecer los enlaces entre marcas es paralelo al de identificación de las mismas. Mataric [5] propone que, al mismo tiempo que se recorren los pasillos identificando marcas, éstas se enlacen en un mapa topológico.

En el trabajo de Kurz [36], el mapa topológico se construye en dos pasos. En primer lugar se identifican las *situation areas* y posteriormente se busca un camino entre ellas explorando el entorno. Si se encuentra se añade el enlace en el mapa topológico.

En la Figura 22 se muestra un mapa donde el bloque azul es el ente activo el cual se dispone a llegar a un punto demarcado como sol (amarillo), los círculos naranjas son obstáculos y la línea blanca es la trayectoria recorrida por el bloque azul.

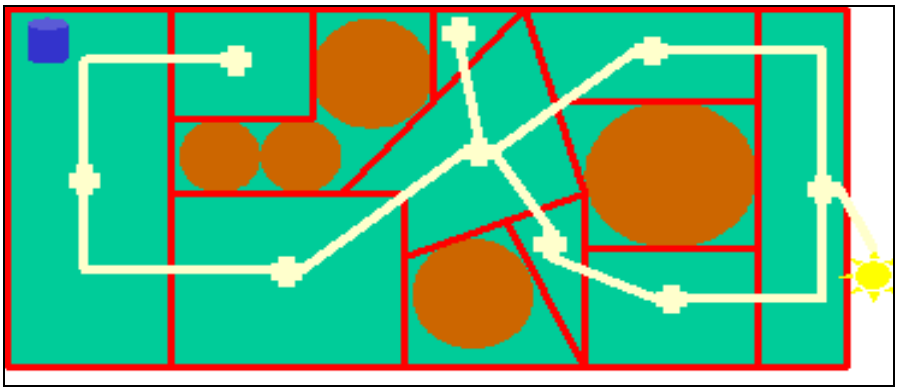


FIGURA 22. Mapa Topológico Completo. [11]

P.4.6.3. MAPAS TOPOLÓGICOS CON MÉTRICA

Son una extensión de los anteriores añadiendo información métrica, normalmente longitud de los caminos entre marcas y orientación de dichos caminos. Con ésta información es posible llevar cabo una planificación de trayectoria más eficiente. Como se ha comentado anteriormente en los mapas topológicos suele incluirse información sobre la longitud y orientación de los enlaces entre marcas. Pero también puede añadirse información métrica sobre la marca en sí. Imagine un mapa donde las paredes sean marcas del mapa. Si se consigue medir la longitud y orientación de las mismas ésta información puede pasar a ser parte del mapa.

[11] Figura 22 (.www.robotica.li2.uchile.cl/el710/Clase11.ppt)

P.4.6.4. MAPAS DE MÉTRICA COMPLETA

Los mapas topológicos son adecuados para los entornos que cumplen que:

- Las marcas naturales o artificiales dominan el entorno y estas son fiables. El robot puede moverse entre ellas.
- Las marcas están unidas por caminos. Para ir de una marca a otra el robot seguirá éste camino por medio de alguna estrategia, como seguir un muro (si lo hay).

Pero éste no es el caso de todos los entornos en los que un robot puede moverse y no lo suele ser en el caso de entornos construidos por humanos donde las similitudes entre zonas son grandes. Este es el caso de un almacén automatizado donde hay gran cantidad de estanterías, todas con las mismas dimensiones y dispuestas de una forma ordenada. En estos casos, para moverse de un sitio a otro la información más relevante que puede utilizar el robot es la información métrica, posiblemente en forma de coordenadas cartesianas. De ésta forma se puede también conocer las zonas libres y las ocupadas. Los mapas de métrica completa se subdividen en:

- MAPAS DE CARACTERÌSTICAS: El mapa es una lista de características básicas (esquina, muro, cilindro), así como sus propiedades (localización, orientación, tamaño).

- MAPAS DE ÀREA O REJILLAS: El entorno se divide en un número de regiones, normalmente cuadradas y de igual dimensión. A dicha región se le atribuye una o varias propiedades. Normalmente, si está ocupada o no.

Los mapas de características ponen más atención en los obstáculos, haciendo así más difícil la tarea de planificar trayectorias. Estos últimos también son más utilizados para localización en el entorno.

P.4.6.5. MAPAS DE CARACTERÍSTICAS

Uno de los primeros trabajos en construcción de mapas por robots fue el del robot CART de Stanford por Moravec [37]. Este robot utilizaba visión estereoscópica para la localización de características en el entorno 3D, utilizando elipsoides para situarlas de una forma incierta en el espacio. De esta forma, los obstáculos eran agrupaciones de elipsoides que se sobreponían entre ellas. Esta representación era utilizada para planificar y ejecutar caminos a posiciones determinadas por el usuario, siendo el resultado en muchas ocasiones frustrante ya que fallaba en determinar posiciones de obstáculos, después de tardar horas en recorrer unos pocos metros. Este modelo aportó una gran verdad a la robótica [38]:

"Los modelos hay que probarlos en el mundo real."

Normalmente los mapas asumen que el mundo puede ser modelado en 2D. Esto suele ser suficiente sobre todo cuando el entorno a modelar es construido por los humanos. Al mismo tiempo se consigue una simplificación enorme con esta suposición a nivel de sensores y a nivel de complejidad de representación y cómputo. Algunos autores asumen una representación 2D y modelan el mundo como una colección de líneas con la información obtenida de un sensor de sonar o de infrarrojos y de ésta forma estimar su posición. En ningún caso se añaden al mapa nuevas líneas.

P.4.6.6. MAPAS DE REJILLA (*GRID MAPS*)

Bajo esta denominación se encuentra un conjunto de trabajos, cada uno con una denominación distinta como *Occupancy grids* [39], *Certainty grids* [40], *Probability Maps* [41], *Histogramic grid* [42]. Todos ellos tienen en común que se divide el espacio en una retícula, donde cada elemento de la misma (celda) tiene asociado uno o varios valores que representan una propiedad, normalmente ocupación. Las diferencias se basan en:

- Qué forma deben tener las celdas.
- Qué valor numérico (propiedad) debe almacenarse.
- Cómo deben actualizarse estos valores, conforme el robot explora el entorno.

La mayoría de los mapas de *grid* utilizan representaciones con celdas de igual tamaño, con una localización basada en coordenadas cartesianas. Sin embargo, como ya se ha comentado antes, las representaciones de éste tipo no son compactas cuando hay gran cantidad de espacio no ocupado (libre) en el entorno. Por este motivo se ha propuesto en ocasiones para éste tipo de mapas el uso de *quadtree*. Estos mapas se construyen de forma recursiva, dividiendo el entorno en 4 regiones y volviendo a dividir cada una de ellas sólo si está

parcialmente ocupada. Este proceso se aplica hasta llegar al elemento mínimo (celda base) o bien cuando el *quadtree* está totalmente ocupado.

El valor a almacenar en cada celda es también un elemento diferenciador. Elfes [39], en sus primeros trabajos utilizaba valores de ocupación discretos (ocupado, libre, desconocido) y un valor de certidumbre en la característica variable de 0 a 1. En los últimos trabajos los valores de certidumbre son tratados más como valores de probabilidad que miden la ocupación de la celda. Lim [40] sigue una línea similar pero añadiendo también un valor de probabilidad de orientación. También se utiliza un *grid* con valores enteros llamados *certainty values* que pueden ir de 0 a 15. La estrategia de evitación de obstáculos se lleva a cabo mediante técnicas similares a los campos potenciales donde un valor alto en una celda provoca una fuerza repulsiva del mismo.

La actualización de los *grids* se lleva a cabo con la fusión de la última información recibida de los sensores y la existente en el mapa. Si en el mapa se representan probabilidades de ocupación es importante tener un modelo probabilístico del sensor. En la Figura 23 se muestra un mapa sobrepuesto con cuadricula donde no se ha aplicado el modelo probabilístico de ocupación, y en la Figura 24 se muestra el mismo mapa con el modelo aplicado donde los cuadros rellenos en rojo implican que la probabilidad de que en ese lugar exista un obstáculo es mayor que en la cuadriculas que no sean de color rojo.

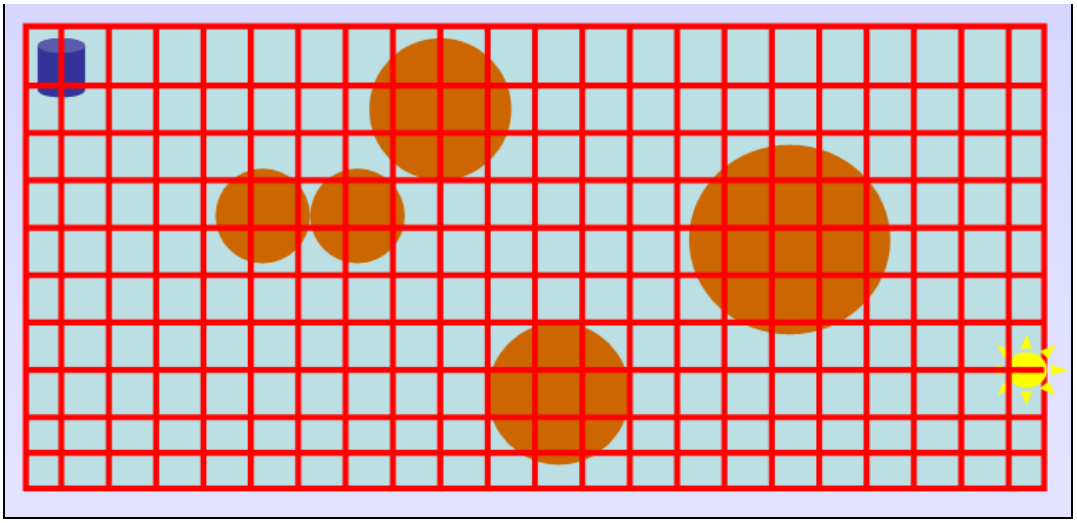


FIGURA 23. Mapa de Rejilla Sobrepuesto.[12]

[12] Figuras 23, Figura 24 (http.www. robotica.li2.uchile.cl/el710/Clase11.ppt)

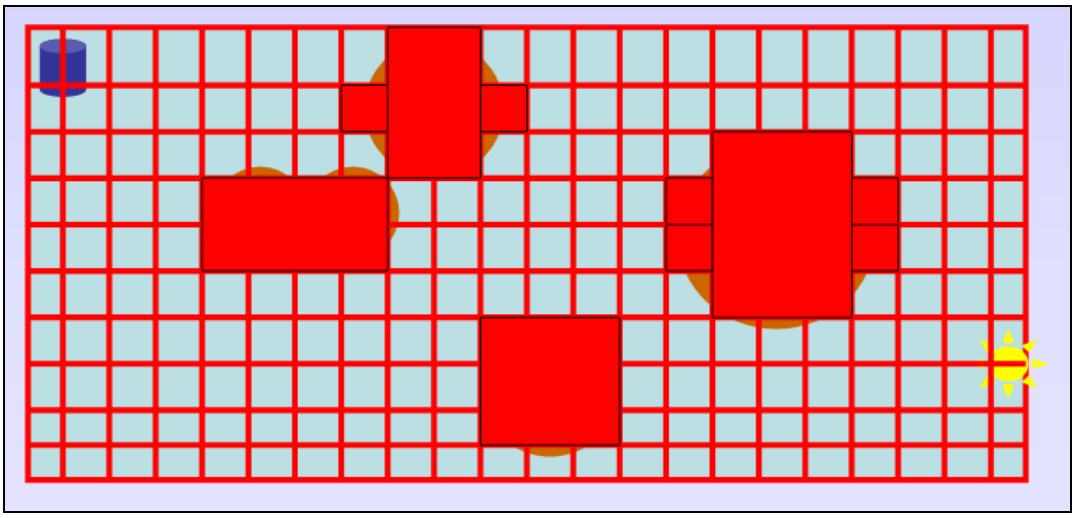


FIGURA 24. Mapa de Rejilla con Aproximación.

P.4.6.7. TIPOS DE MODELADO DEL AMBIENTE

Básicamente se puede distinguir dos tipos diferentes de modelados del ambiente, el primero consiste en construir un modelo del mundo tan preciso como sea posible, para esto se requiere que el robot cuente con un sistema confiable de sensores, es evidente que el modelo de este tipo de ambientes no se lleva a cabo en tiempo real, y generalmente se utiliza para construir representaciones virtuales de escenarios reales.

El segundo tipo de modelado de ambientes son aquellos que se realizan de manera rápida pero burda (generalmente un mapa en dos dimensiones), este tipo de modelado es el más usado en robótica, debido al menor tiempo que se requiere para construir el modelo, no se requieren de sensores demasiado sofisticados.

1. DESARROLLO DE PROYECTO

La robótica móvil es un área de intensa investigación cuyo objetivo es el desarrollo de robots autónomos que puedan desenvolverse en ambientes dinámicos. En robótica móvil se presentan tres problemas fundamentales a resolver:

1. NAVEGACIÒN.
2. LOCALIZACIÒN.
3. RECONOCIMIENTO ENTORNO.

Se desarrolla un estudio detallado de cada problema y planteamiento de soluciones óptimas con la implementación de hardware reconfigurable FPGAS; para luego fusionar los conceptos en la construcción de lo que se definirá como plataforma móvil auto-guiada (VAG) o robot móvil autónomo.

La navegación requiere tener un conocimiento del entorno donde se desenvuelve el vehículo para ser eficiente. Este conocimiento del entorno requiere la existencia de un mapa del entorno y requiere que la plataforma móvil esté localizada en dicho mapa. Si la navegación es de tipo deliberativa, se utiliza un mapa del entorno, construido en forma previa por el robot o entregada al robot por su diseñador, para navegar. Si la navegación es de tipo reactiva se navega utilizando únicamente la información percibida por los sensores, para, por ejemplo evitar obstáculos. Los mapas del entorno permiten que el robot tenga una representación del medio en que se está desenvolviendo. Para que el robot pueda construir un mapa del entorno por si mismo, requiere poder navegar reactivamente por su entorno y además estar posicionado o localizado adecuadamente en el mapa que está construyendo. Idealmente mientras el robot esta construyendo el mapa de su entorno debiera estar alerta ante cambios en este. La tarea de localización implica que el robot debe conocer en cada momento su posición y su orientación, respecto a una representación global del espacio. Es decir se trata de una auto-localización.

La localización de un robot es imprescindible para las tareas de navegación y de construcción automática de mapas de entorno, resolver tareas de navegación en ambientes desconocidos es una función propia de la plataforma móvil (Robot móvil).

1.1. NAVEGACIÓN

Para la navegación de la plataforma móvil se debe tener en cuenta el mecanismo que hace posible el movimiento de la misma, en este caso las ruedas, que a su vez están acopladas mecánicamente a los motores DC, este mecanismo debe ser controlado, en esta aplicación el controlador esta dado por sensores de movimiento llamados *encoders* incrementales además de la implementación de una brújula digital que indicará el rumbo de la plataforma móvil con respecto al norte magnético de la tierra.

Con la brújula digital se debe tener en cuenta que dicho dispositivo entrega una señal análoga por medio de la cual se determina el Rumbo magnético (ángulo formado por la línea virtual de referencia de la brújula en la plataforma móvil y el Norte magnético de la tierra) como se muestra en la Grafica 2.

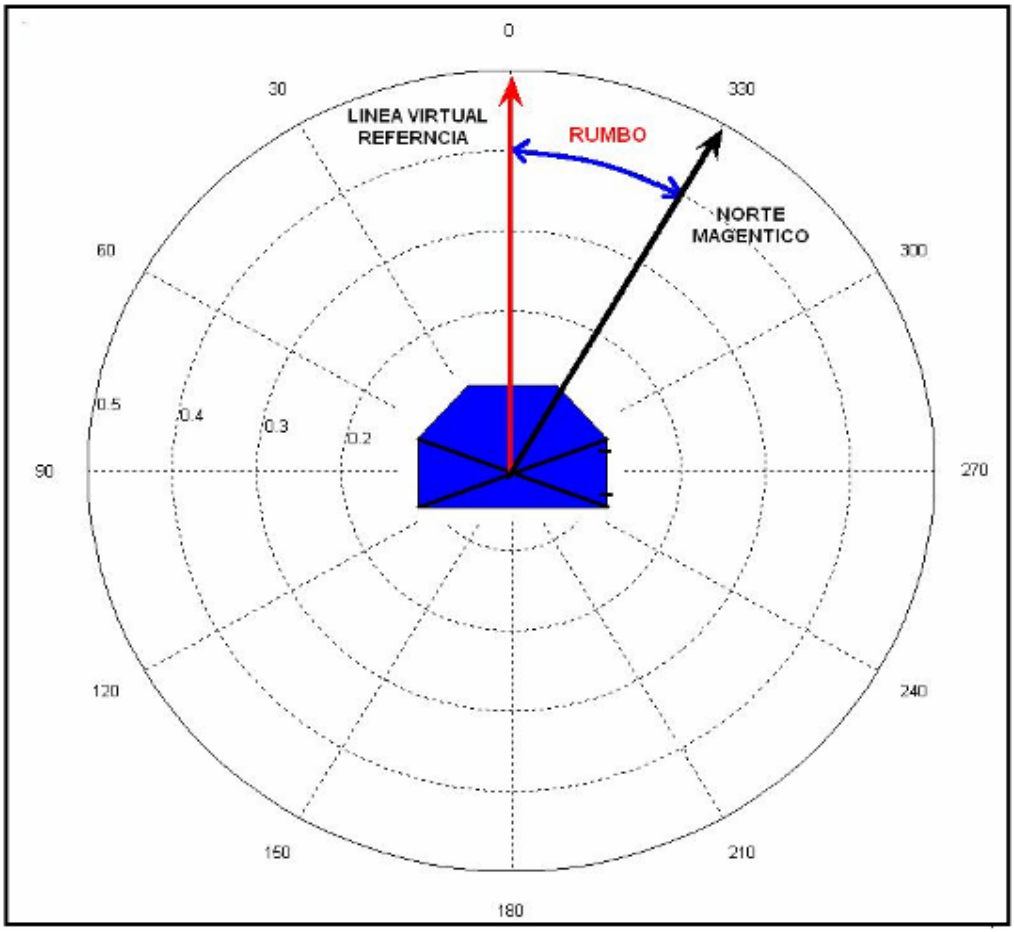


GRÁFICO 2. Rumbo Magnético

Para el desarrollo del trabajo la implementación de la brújula digital se limita solo para construir el mapa de entorno y no para la navegación de la plataforma móvil puesto que se considera

que los sensores medidores distancia recorrida por las ruedas (*encoders*) tienen muy buena resolución y con un código de control para los motores DC que se verá mas adelante, se disminuye el error de odometría.

1.1.1. MOTORES DC.

La plataforma debe movilizarse para poder evadir obstáculos, por lo tanto se requiere de los motores y cierto control tanto de dirección como de velocidad, necesarios para obtener un buen desplazamiento y cambio en el sentido de giro. La plataforma está constituida por dos motores marca DAYTON (ver Foto 2) a 12V DC, modelo 2L010, los cuales desarrollan una velocidad máxima de 18 metros por minuto, su característica de torque es de 20 kg con máxima carga.

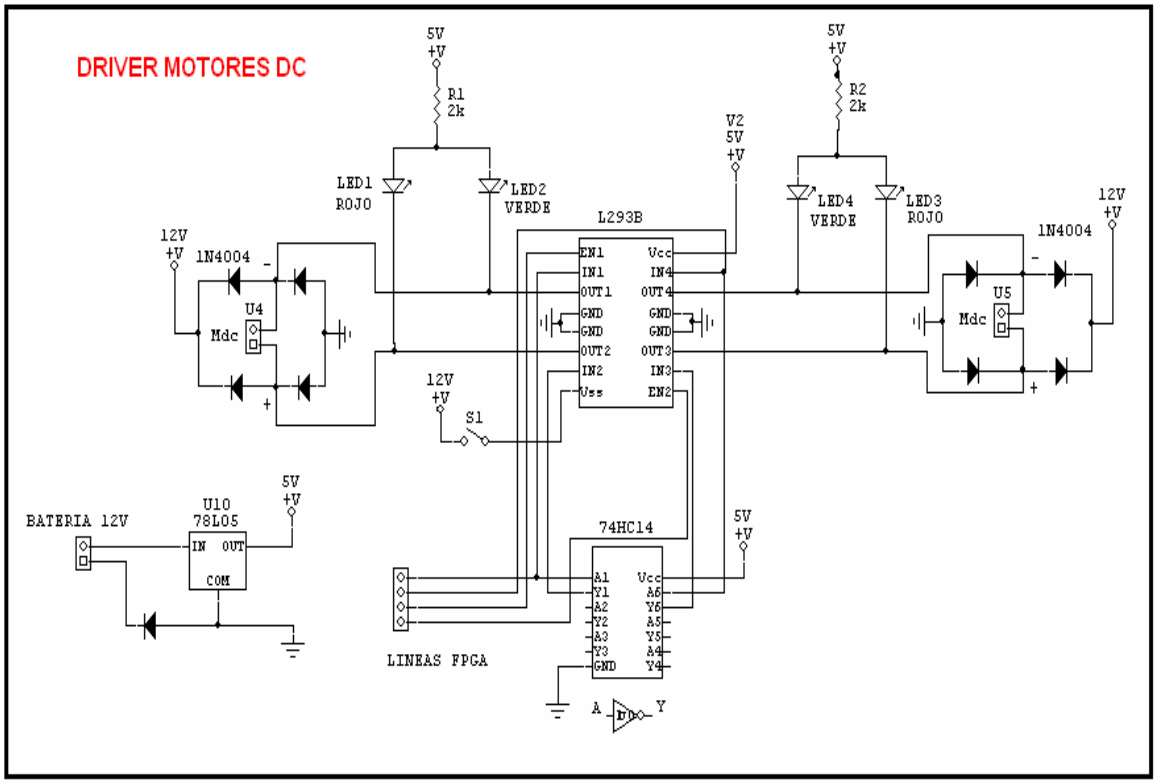
FOTO 2. Motor DC Marca DAYTON

El control de los motores se realizó mediante el circuito electrónico del *drive* de control denominado *Microcore-11* que se encarga de manejar la velocidad y la dirección de ambos motores, su funcionamiento principal se centra en el circuito integrado L293B este circuito integrado consta de un puente H .

Las características de la tarjeta de control de los motores son:

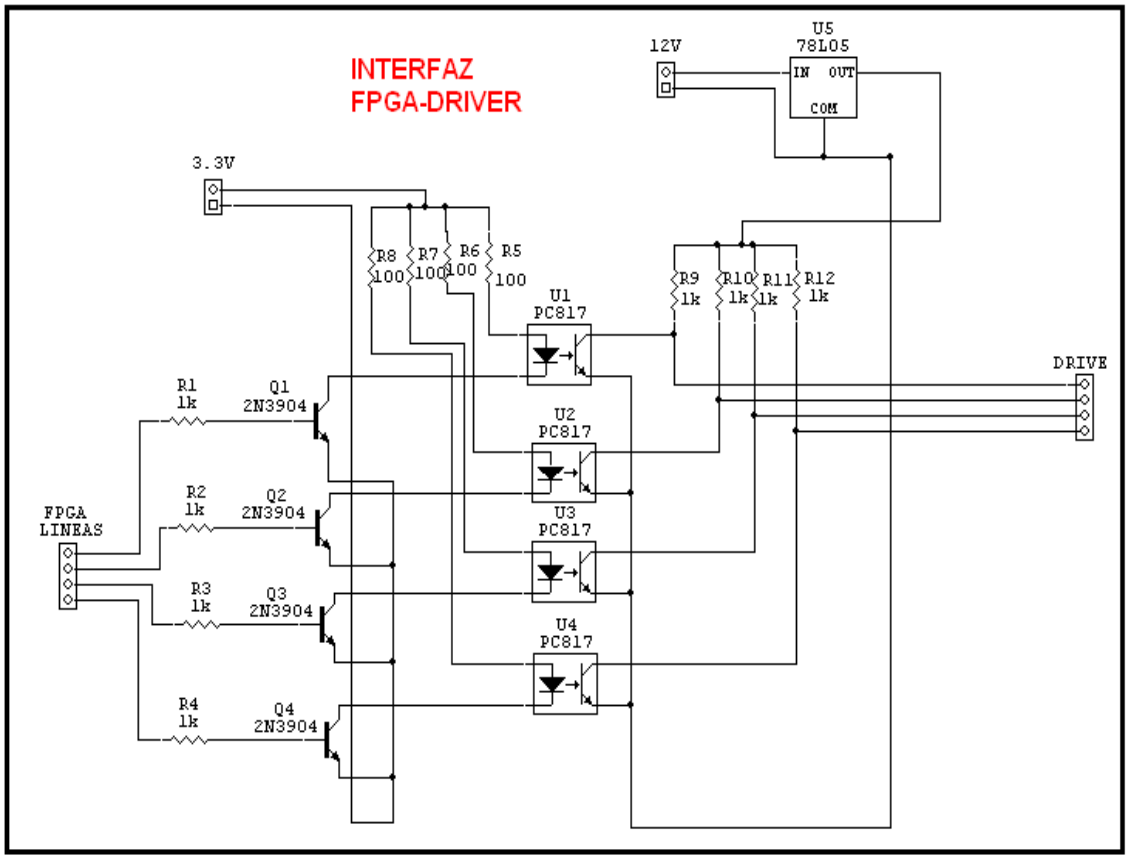
- Regulador de 5 V.
- 2 bits de control de giro
- Alimentación de salida de 1 amperio por canal.
- Rango de alimentación de los motores entre 5 V y 36 V.
- Indicadores de dirección y estado.
- Protección interna contra voltaje de pico inverso.
- Conector individual de fuente de voltaje de cada motor.

Se implementó el circuito de la tarjeta *Microcore-11* para dicho control, que a su vez está conectada por medio de una interfaz de potencia a la tarjeta SPARTAN3-E ya que se manejan diferentes niveles de tensión. El circuito del *Microcore-11* se muestra en el Esquemático 1.



ESQUEMÁTICO 1. *DRIVER* MOTORES

El circuito de mando esta controlado por 4 líneas de manejo que provienen de la tarjeta FPGAS dos líneas son de dirección y dos de velocidad, a cada motor se le asignan una línea de dirección y otra de velocidad, estas son independientes para cada motor. Para poder conectar el circuito de control con la tarjeta FPGAS se debe realizar un interfaz de potencia como se muestra en el Esquemático 2, que brinde la protección y seguridad de los niveles de tensión y corriente requeridos por los dos módulos, ya que la tarjeta FPGAS sólo entrega máximo 10 mA por salida.



ESQUEMÀTICO 2. Interfaz FPGA- *DRIVER*

Para el diseño se muestra en la Tabla 1 la dirección de los motores según la configuración de la señal de comando.

SEÑAL FPGA		MOTORES		PLATAFORMA
Dir_dc(1)	Dir_dc(0)	MOT2	MOT1	
0	0	ATRÁS	ATRÁS	ATRÁS
1	0	ADELANTE	ATRÁS	DERECHA
0	1	ATRÁS	ADELANTE	IZQUIERDA
1	1	ADELANTE	ADELANTE	ADELANTE

TABLA 1. Control de Dirección para Movimiento.

1.1.2. CONTROL DE MOTORES DC.

Para ejercer el control de los motores a través del circuito de control se diseñó el bloque de programación mostrado en Diagrama 4 en la tarjeta FPGAS.

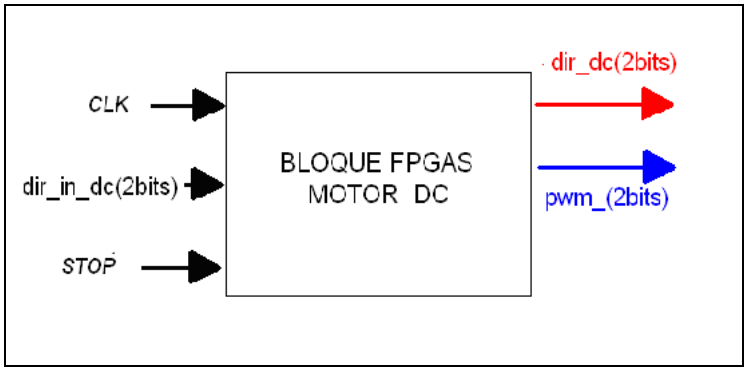


DIAGRAMA 4. Bloque de Control Motores.

Consta de 3 entradas fundamentales el CLK (circuito de reloj interno de la FPGA), una entrada para dirección de 2 bits para motores y la última para detener el movimiento de los motores, dos salidas que corresponden a dirección de 2 bits y otra salida para control de velocidad por PWM (*pulse width module*).

- *CLK*: Es la señal de reloj, que es directamente conectada desde el reloj externo de la tarjeta, este reloj es de 50 Mhz y es representado por un solo bit. El pin de la FPGA donde esta conectado este reloj es el C9.

- *STOP*: Es una señal de habilitación de los motores, e indica que la salida del PWM que activa los motores se desactive si parado esta en '1', de lo contrario dicho PWM alimentara los motores DC, haciendo que estos se muevan. Parado es representado por un solo bit.

- *Dir_in_dc*: Esta señal de 2 bits representa las 4 posibilidades de movimiento de la plataforma, considerando que cada bit representa la dirección del movimiento de cada motor independientemente. Esta señal es conectada directamente con la señal dir_dc

- Dir_dc: Esta señal de 2 bits representa las 4 posibilidades de movimiento de la plataforma, estos dos bits son una salida de la FPGA, y se conectan con el circuito impreso que esta encargado de manejar los motores DC. Los pines físicos de la FPGA donde están conectadas estas dos señales se encuentran ubicados en el conector J2 de la tarjeta, y son; dir_dc(1) ubicado en el pin F7 y dir_dc(0) ubicado en el pin E7

- PWM**:** Esta señal de 2 bits esta dividida para cada motor independientemente, y es una señal de habilitación, que es controlada por un PWM, cuya función es regular y permitir tener control sobre la velocidad. Este PWM es un tren de pulsos con una frecuencia de 100Hz y un porcentaje de trabajo útil del 95 % lo que da una velocidad aproximada de 27.4 rpm. Los pines físicos de la FPGA donde están conectadas estas dos señales se encuentran ubicados en el conector J2 de la tarjeta, y son; m_dc(1) ubicado en el pin B6 y m_dc(0) ubicado en el pin A6.

En el Diagrama 5 se muestra el generador de frecuencias para PWM que excitará los motores DC y a su vez establecer un control de los mismos.

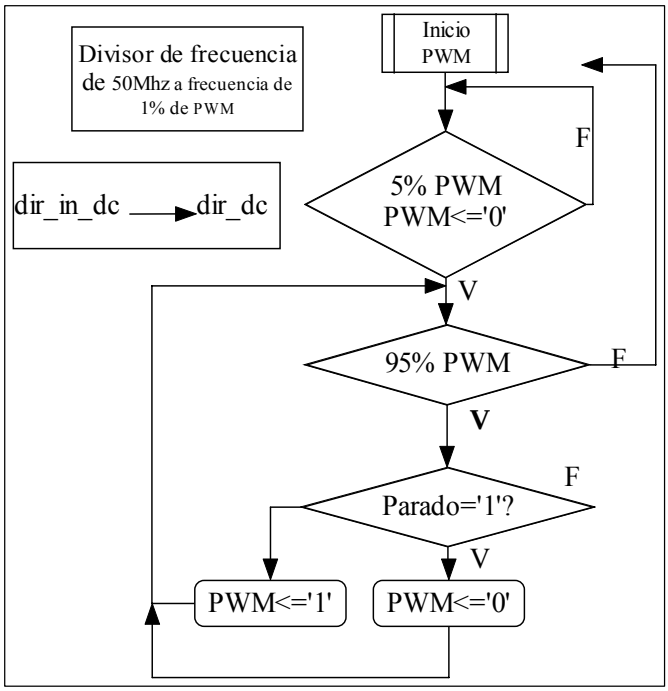


DIAGRAMA 5. Diagrama de Flujo Control Motores

Los motores DC son los que permiten el desplazamiento de la plataforma móvil (Robot móvil), estos están activos únicamente en el momento que se ha terminado de obtener un mapa local, y se pretende tomar otro mapa local pero esta vez en un punto de referencia diferente.

La ejecución de un bloque de hardware de la FPGA permite el control de la dirección en línea recta por igualación de desplazamientos, el control de la velocidad por PWM, el cambio de sentido (girar) y el desplazamiento total en los dos ejes coordenados por parte de la plataforma. Estas tareas son realizadas gracias a la acción del control de motores DC a través de la captura de datos de los *encoders* de los cuales se profundiza aguas abajo en la sección, además controla desplazamiento, giro de los motores y se encarga de generar una señal de PWM.

1.1.3. CONTROL DE DIRECCIÒN POR CORRECCIÒN DE VELOCIDAD

Este control se realiza por rotación, del *encoder* de rotación que está localizado en el eje del motor; el conjunto *encoder* sensor opto acoplador provee a la FPGA una señal que ha sido previamente acondicionada (tren de pulsos), esta señal permite controlar el giro de cada motor con una nueva señal la cual es generada por la acción del desfase en la lectura del tren de pulsos leídos la FPGA

La base de este procedimiento consiste en que en cualquier momento los motores 1 y 2 deben estar sobre el mismo giro, si esto no sucede significa que uno de ellos está en adelanto respecto al otro, el procedimiento es anular la señal de alimentación de PWM del motor que va en adelanto. En el caso de que la señal “parado” se encuentre en ‘1’ significa que la plataforma no se debe desplazar, y por lo tanto el PWM para ambos motores se anula.

DIVISOR DE FRECUENCIA DE 50 MHz A FRECUENCIA DE 1% DE PWM.

Para ejecutar el control de velocidad por PWM se hace a través de un tren de pulsos generados por la FPGA; dicho tren de pulsos con una frecuencia de 100Hz y un porcentaje de trabajo útil del 95 % entrega una velocidad aproximada de 27.4 rpm; Para lograr identificar estos porcentajes se determina inicialmente, que valor debe tener el 1% de el PWM, y este valor se usa como base de tiempo, considerando que la frecuencia de reloj de la FPGA es de 50Mhz es necesario realizar un divisor de frecuencia, que convierta de 50Mhz a el 1% del tiempo de el tren de pulsos de 100Hz, este procedimiento se realiza mediante un contador, ya que el 1% es 0.1ms, y el tiempo mínimo del reloj de la FPGA es 20ns, esto dá como resultado un valor de conteo de 5000, por lo tanto se implementa un contador que se incrementa en 1 cada que ocurre un flanco de subida en la señal de reloj principal, y cuando el contador llega a 5000 se coloca en ‘1’ una bandera que indica que en ese momento se produce un flanco de subida del reloj de 10khz, correspondiente al 1% del PWM.

1.1.4. CONTROL DE SENTIDO DE GIRO DE PLATAFORMA MÒVIL.

Se realiza mediante la excitación de “1” ó “0” lógico en las entradas de la tarjeta microcore-11 MDIRA y MDIRB que corresponden a las líneas de dirección, este giro dá independencia de movimiento a cada motor y determina los siguientes estados de la plataforma:

- Adelante
- Atrás
- Giro a la derecha sobre su propio eje desde 0° a 360° grados
- Giro a la izquierda sobre su propio eje desde 0° a 360° grados

Las direcciones son determinadas por los estados lógicos que estén presentes en las salidas que han sido designadas como control de motores DC en la FPGAS; el estado lógico alto “1” determina en el motor el giro hacia delante y el estado lógico bajo “0” determina en el motor el giro contrario.

1.1.5. CONTROL DE INERCIA DE FRENADO EN EL DESPLAZAMIENTO DE LA PLATAFORMA

Los desplazamientos de la plataforma deben ser exactos para evitar desviaciones en el recorrido de esta; para la eliminación en gran medida de este error la FPGA da la orden a los motores que cambien su sentido de giro por un instante de tiempo que es del orden de los milisegundos, esta acción actúa como freno instantáneamente contrarrestando los efectos causados por la inercia y posibles deslizamientos a causa de las ruedas.

Finalmente la descripción de hardware que representa el control de los motores DC es el presentado en el ANEXO A**.**

La plataforma móvil se presenta con el mecanismo de movimiento diferencial, dos motores DC, dos ruedas, una estructura de apoyo, dos apoyos tipo ruedas roll-on, y compartimiento de la batería Ver Fotos 3, 4, 5, 6.

FOTO 3. Vista Superior Estructura.

FOTO 4. Vista Lateral Derecha E.

FOTO 5 .Vista Frontal E.

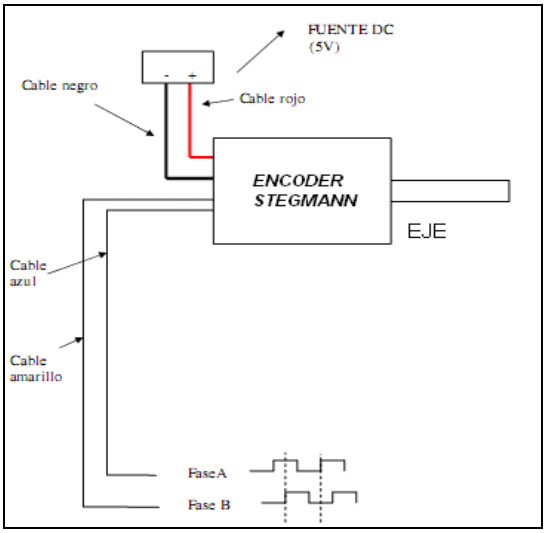
FOTO 6. Perfil E.

1.1.6. ENCODERS

Para el sistema de desplazamiento es necesaria la implementación de sensores que arrojen las señales necesarias para medir datos de desplazamiento y dirección por lo que se usan los *encoders* incrementales *STEGMANN*. Ver Foto 7

Esta tecnología es un modelo compacto que garantiza la entrega de los pulsos para poder realizar los respectivos trabajos de odometría en el desplazamiento de la plataforma móvil. En el Esquemático 3 se muestra la polarización y salidas del *encoder STEGMANN.*

FOTO 7. ENCODER STEGMANN



ESQUEMÁTICO 3. Distribución de líneas del *Encoder*.[13]

[13] Foto 7, Esquemático 3
(http://micromotores.com/productos/product_info.php?products_id=657)

1.1.6.1. ECUACIONES Y CÁLCULO DE DISTANCIA CON *ENCODERS* INCREMENTALES.

El cálculo del desplazamiento de la rueda a la cual va acoplado el *encode*r de 500 pasos se realiza tomando el diámetro de la rueda que para este caso es de 15cm; se determina la resolución en centímetros de paso entre ranuras del *encoder* así:

Distancia recorrida en una vuelta $(\pi \times 15cm) = 47.124cm, \rightarrow (38)$.

$$\text{Resolución Encoder} = (\pi \times 15cm)/500ranuras = 0{,}0942477\frac{cm}{ranura}, \rightarrow (39).$$

En el Grafico 3 se muestra la circunferencia como rueda y su perímetro para asociarlo con la distancia a cumplir después de un giro.

Para determinar la distancia recorrida por la plataforma se debe tener en cuenta las siguientes variables:

- Radio de las llantas = R
- Distancia recorrida = S $$S = \frac{R * 2\pi}{PPV} \times C, \rightarrow (40)$$
- Pulsos por vuelta = PPV
- Número de pulsos recorridos= C

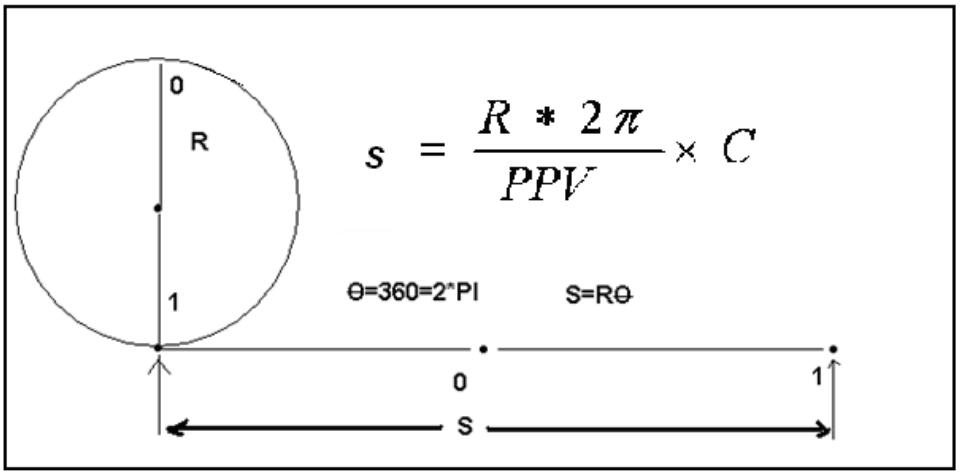


GRÁFICO 3. Longitud de Arco para *Encoder*.

Una vez la señal es obtenida de los *encoders* y enviada al hardware reconfigurable FPGAS se procede a la parte de control.

El controlador en base a los *encoders* esta encaminado a que cada uno debe estar acoplado mecánicamente a los ejes de cada motor DC, para así poder contar en su equivalente en pulsos si el motor ha movido la rueda sea adelante o atrás determinada distancia, así pues se establece un control comparador en el Diagrama 6 cuyo fin es comparar la información de los *encoders* en forma de conteo, para así verificar que los motores se están excitando al mismo tiempo.

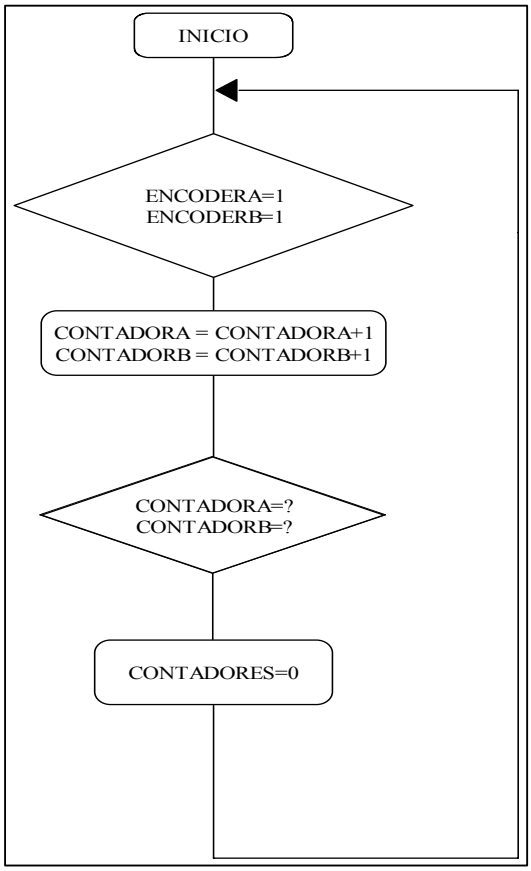
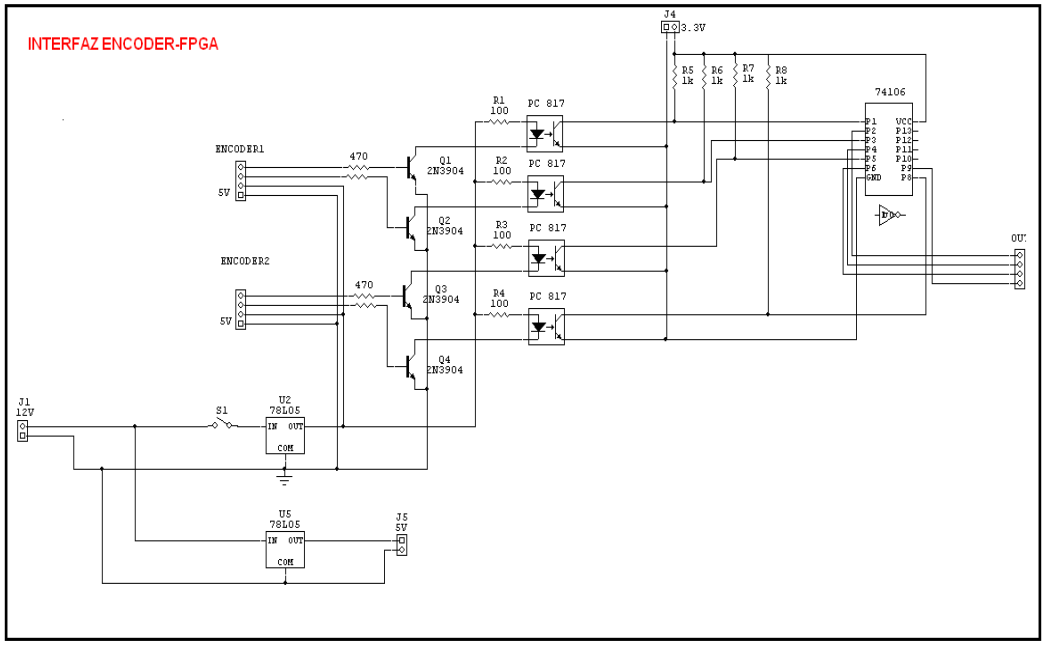


DIAGRAMA 6. CONTROLADOR DE *ENCODERS*

El circuito del módulo de *encoders* - interfaz FPGA se muestra en el Esquemático 4.



ESQUEMÁTICO 4. Interfaz *encoders* – FPGA.

1.1.6.2. FUNCIONAMIENTO CONTROL DE *ENCODERS*

Los *encoders* incrementales entregan los pulsos ya acondicionados para el respectivo análisis de la información con lo que se procede a aplicar el método correspondiente para dicho análisis en el cálculo de medir distancia, se utiliza la FPGA para capturar la señal y se implementa un filtro que garantiza que la señal proveniente de los *encoders* es una señal de forma de onda cuadrada, con este procedimiento se da paso al conteo de pulsos por medio de flancos de subida, al capturar los flancos se implementa un contador que entrega la cantidad de pulsos generados por el *encoder* al desplazarse la plataforma móvil con la ecuación (6), se puede determinar cuantos pulsos necesita para recorrer determinada distancia o que distancia es la recorrida por la plataforma móvil.

El diseño de hardware en VHDL que describe este control es presentado en el ANEXO B.

1.1.7. FUSIÓN MOTORES DC Y ENCODERS PARA PLATAFORMA MÓVIL

Para controlar adecuadamente el sistema se debe establecer un acople mecánico entre los motores y los *encoders* puesto que la información que entregan los *encoders* se refiere a distancia recorrida y giro sea a izquierda o derecha; es decir que con un comando externo el sistema se realimente con los datos de los *encoders* hasta que cumpla la orden del comando externo. En la Foto 8 se muestra el acople mecánico para controlar el PWM de los motores.

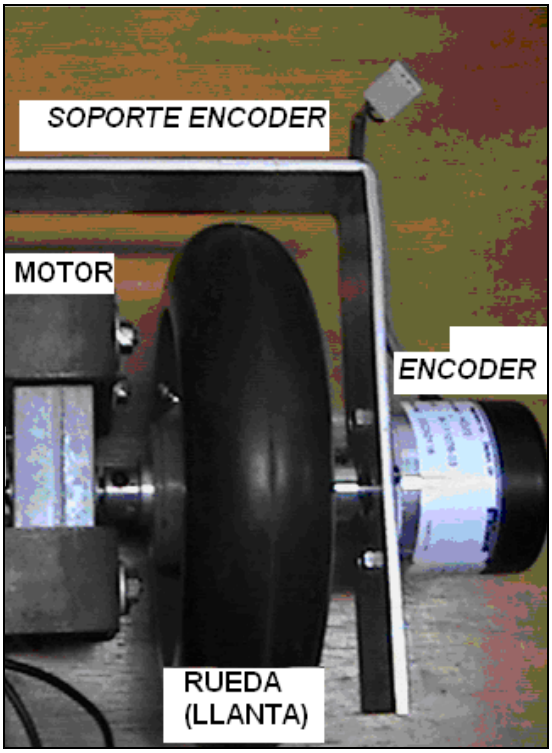


FOTO 8. Acople Motor DC, Rueda, *Encoder.*

Una vez hecho el montaje de *encoders* y motores en la plataforma móvil se prepara un escenario para hacer pruebas de odometría donde se pueda verificar el control de los motores para recorrer distancia en línea recta y poder cuantificar el error de odometría y también para efectuar giros de 90 grados, para lo anterior se propuso programar en la tarjeta FPGA un código cuyo fin es controlar la plataforma para recorrer una distancia de 1 Metro, el escenario propuesto para esta prueba de distancia se muestra en Figura 25; y tomar 10 muestras de prueba.

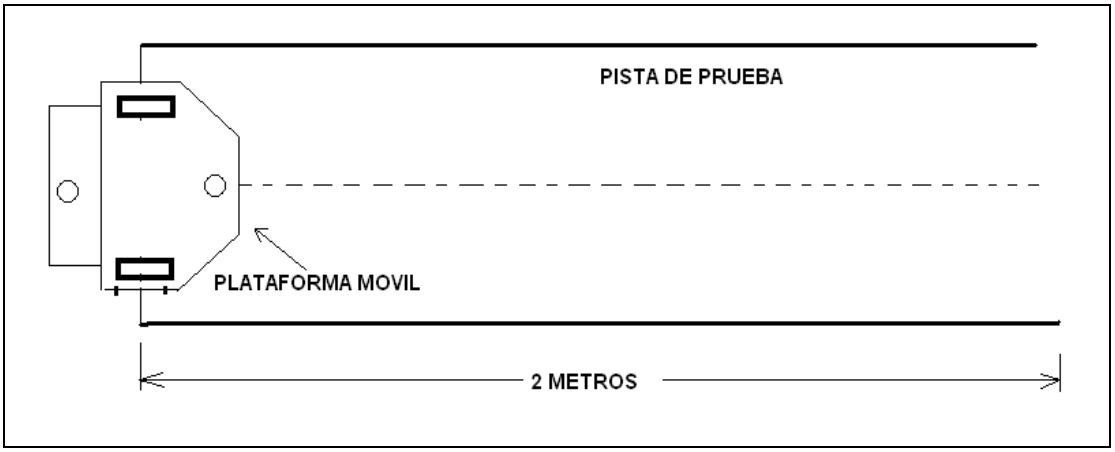


FIGURA 25. Escenario de Prueba de Distancia Recorrida

El error de odometría en las pruebas de distancia lineal se muestra en la sección de Resultados 2.1.1.1 donde se anexa además en video la ejecución de las pruebas.

El otro escenario propuesto para obtener el error de odometría para los giros es el recorrido de una trayectoria cuadrada de 80 centímetros de longitud por cada lado, como se muestra en la Figura 26; y tomar 10 muestras de prueba.

El error de odometría, las pruebas de distancia y giro se muestran en la sección de Resultados 2.1.1.2. Donde se anexa además en video la ejecución de las pruebas.

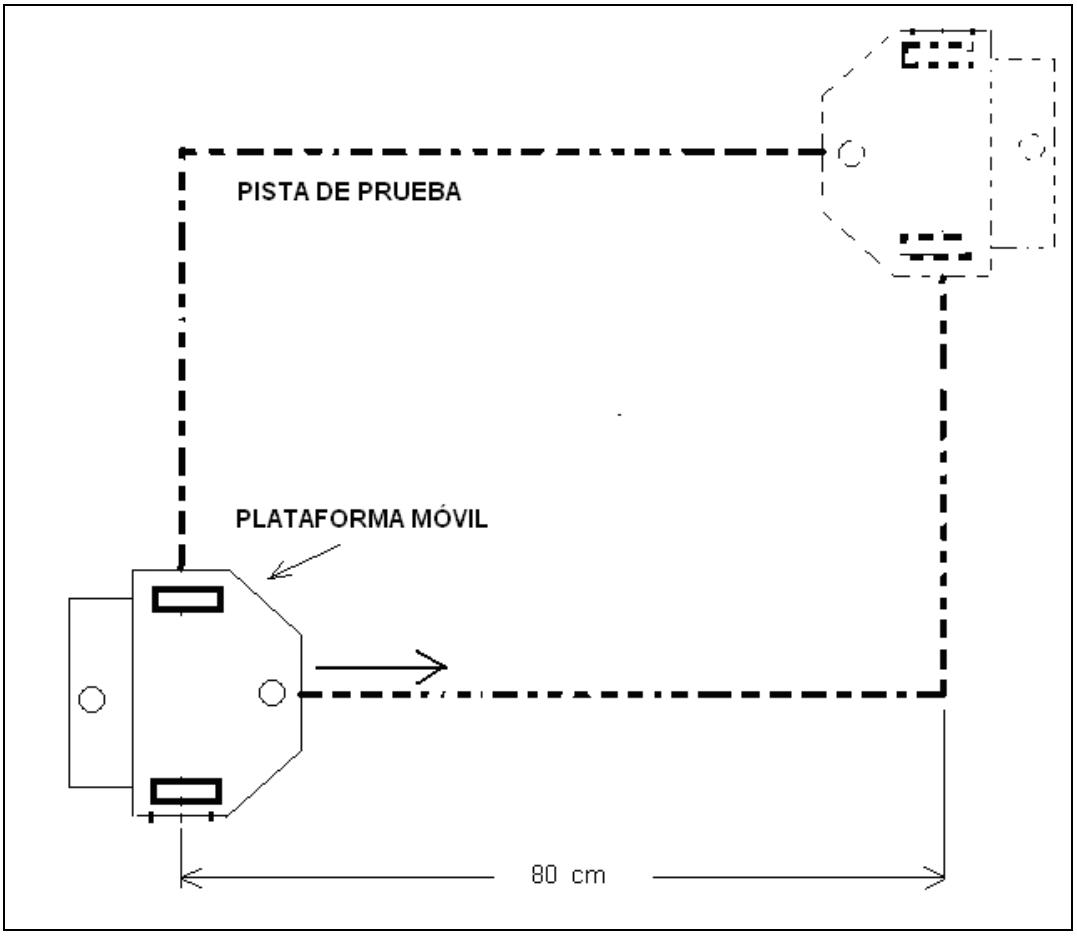


FIGURA 26. Escenario de Prueba para Giro.

1.2. LOCALIZACIÓN

Como solución al problema de localización de la plataforma móvil se tiene como opción la implementación de la brújula digital que entrega el Rumbo magnético (ángulo formado por la línea virtual de referencia de la brújula en la plataforma móvil y el Norte magnético de la tierra) de la plataforma móvil, este dispositivo es controlado con la tarjeta de hardware reconfigurable FPGAS la cual estará encargada de capturar los datos que entrega el dispositivo; estos datos designan el ángulo que existe entre la dirección propia de la brújula y el norte magnético de la tierra.

La brújula digital es un dispositivo electrónico que consta de dos sensores KMZ51 de Philips que son lo suficientemente sensibles como para captar el campo magnético de la tierra (ver Foto 9), su configuración brinda al dispositivo dar mediante dos formas de comunicación una I2C y la otra por variación de ancho de pulso o PWM, permitir al microprocesador calcular la dirección de la componente horizontal del campo magnético natural de la tierra.

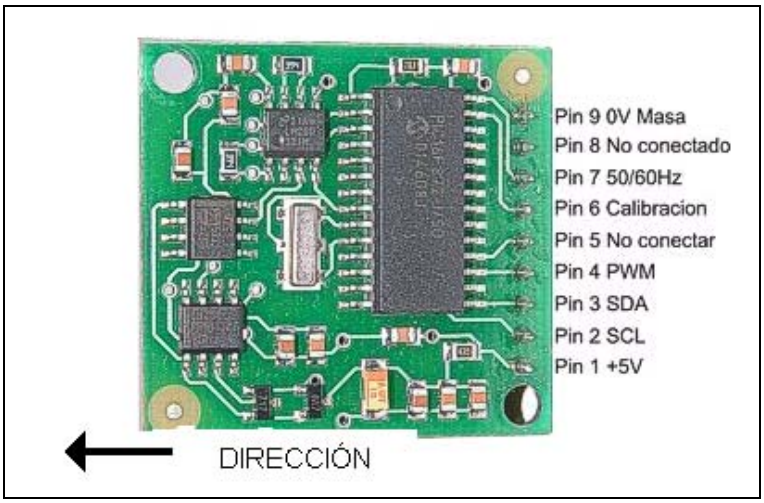


FOTO 9. Montaje Superficial de Brújula Digital[14]

1.2.1. CONEXIÓN Y FUNCIONAMIENTO BRÚJULA DIGITAL.

En el Pin 4, se obtiene una señal PWM en la que un pulso positivo representa el ángulo de la brújula. El pulso varía en su duración desde 1mS (0º) hasta 36,99 ms (359,9º), o dicho de otra forma, el pulso es igual a 100 us por cada grado más 1ms de tara. La señal permanece a cero durante 65 ms entre pulsos, por lo que el periodo de trabajo es de 65mS más la anchura del pulso. El pulso es generado por un contador de 16 bits del propio procesador, con una resolución de 1 us, aunque en la práctica no es recomendable hacer mediciones con una

[14] Foto 9 (http://www.superrobotica.com/S320160.htm)

resolución de más de 0,1º (10 us). Cuando se usa la interfaz PWM, es necesario conectar a +5V mediante 2 resistencias de 47 kΩ, los pines 2 y 3 (SCL - SDA) del interfaz I2C, ya que no se incluye resistencias de *pull-up* en el circuito impreso mostrado en la Foto 9.

La brújula se posiciona en una referencia virtual la cual le da el nombre de NORTE, cuando se va medir con el dispositivo ella capta la diferencia que hay entre la referencia y el Norte magnético de la Tierra como se muestra el Grafico 4, entregando según los dos formas de medición ancho de pulso o por protocolo I2C, en este caso ancho de pulso.

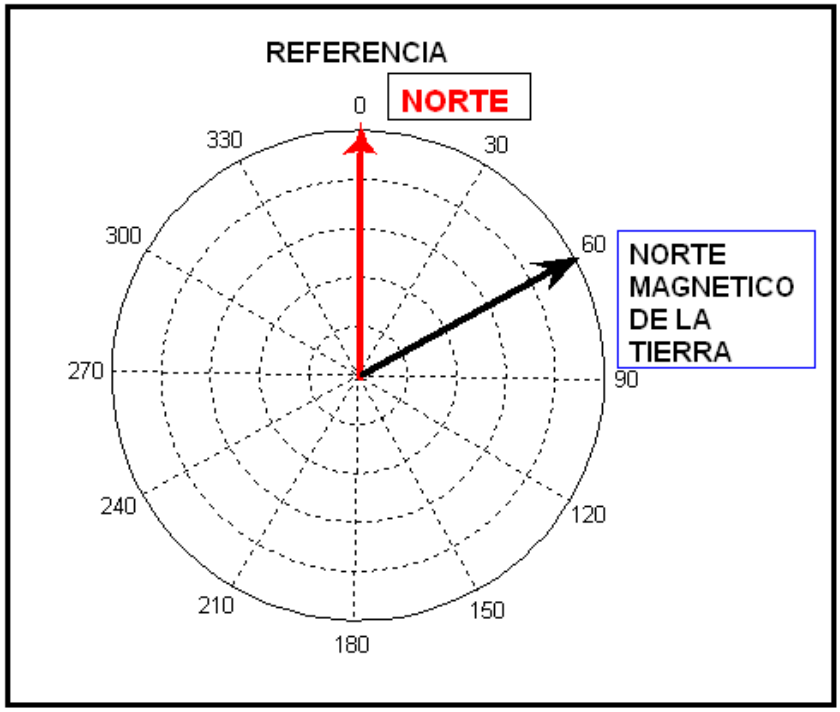


GRÁFICO 4. Funcionamiento de la Brújula

La brújula se posiciona en una referencia virtual la cual le da el nombre de NORTE, cuando se va medir con el dispositivo ella capta la diferencia que hay entre la referencia y el Norte magnético de la Tierra entregando según los dos formas de medición ancho de pulso o por protocolo I2C, en este caso ancho de pulso.

1.2.2. CALIBRACIÓN DEL DISPOSITIVO.

Para la calibración es necesario comprobar con una brújula magnética tradicional los puntos cardinales del sistema. El pin 6 es el pin de calibrado, solo basta con dirigir la brújula hacia cada punto cardinal y activar la calibración luego de esto se lleva a cabo la medida del campo magnético.

Es necesario mantener la brújula alejada de fuentes de campo magnético como metales, imanes, aparatos electrónicos, celulares etc. Es aconsejable realizar un apantallamiento para evitar errores en la medición.

Se trabajó con la salida de PWM hacia el dispositivo de lógica programable (FPGA), el ancho de pulso de la señal de la brújula con una resolución de 0,1º a una frecuencia de 10 khz, esto para garantizar la correcta medición de la señal. Para medir el ancho de pulso de la señal proveniente de la brújula se implementó por medio de hardware reconfigurable un código capaz de capturar el ancho de pulso de la señal y procesarla para determinar el ángulo de desviación con respecto al norte magnético de la tierra con un componente que se muestra en el Diagrama 6.

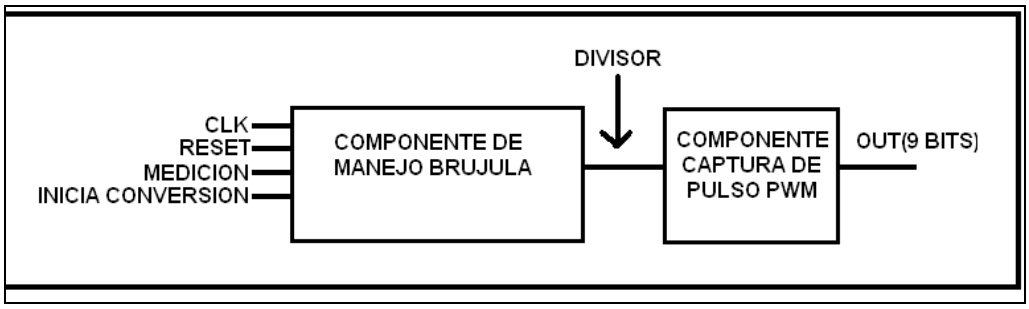


DIAGRAMA 6. Control de Captura de Datos de la Brújula Digital.

Donde se definen los siguientes puertos:

- *Clk:* Reloj principal de la tarjeta Spartan 3E.
- *Reset*: Reinicio de la captura de los pulsos.
- Medición: Entrada de la señal de la brújula digital.
- Inicia Conversión: Activa el proceso de medición al detectar un flanco de subida.
- Divisor: Garantiza la resolución de la medición de la señal de la brújula.
- *Out:* Señal final que garantiza la conversión a binario de 9 bits de la señal análoga de la brújula.

En el Diagrama 7 se muestra el control para capturar el Rumbo entregado por la Brújula Digital.

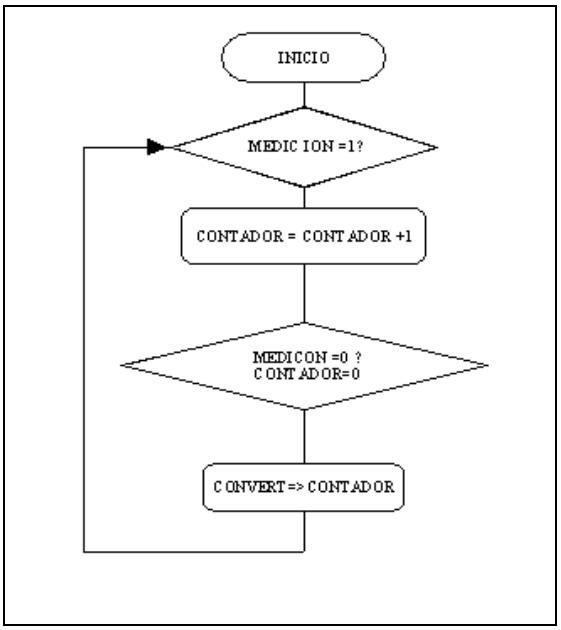


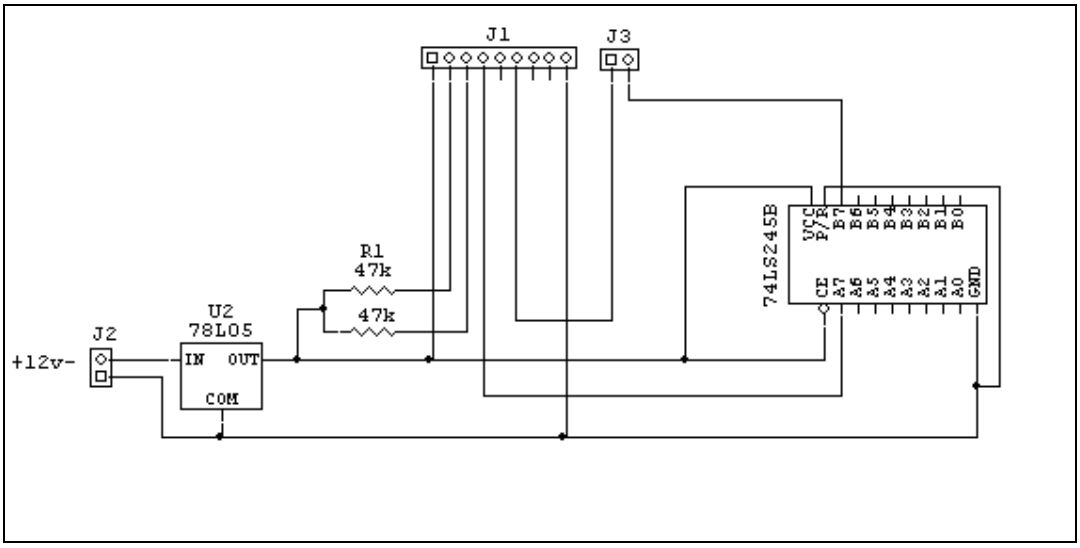
DIAGRAMA 7. Control de Brújula Digital.

La descripción de hardware que captura los datos que entrega la brújula digital se encuentra en el ANEXO C.

1.2.3. IMPLEMENTACIÓN DE BRÚJULA DIGITAL CON INTERFAZ PARA TARJETA FPGA SPARTAN 3E DE XILINX

Los niveles de tensión de trabajo de la brújula digital son de 5 voltios y los de la SPARTAN 3E son de 3.3 voltios, por lo que se hace necesario efectuar una transición de voltajes de operación de ambos dispositivos, para tal objetivo se implementa el circuito integrado 74LS245 el cual es un dispositivo *transceiver* que convierte niveles de tensión de 5 voltios a 3.3 voltios y viceversa.

En aplicaciones anteriores se ha usado el opto acoplador PC817 el cual con una configuración de resistencia y transistores n-p-n se lograba la transición de tensiones de 5 voltios a 3.3 voltios y viceversa con la ventaja que los sistemas quedaban aislados pero con una reducción considerable en el ancho de banda, la frecuencia de trabajo es muy baja, mientras que el ancho de banda del 74LS245 es óptimo para la aplicación debido a que se manejan frecuencias mayores a la de trabajo para el PC817. Ver Esquemático 5.



ESQUEMÁTICO 5. Conexión Brújula Digital

Para confirmar la implementación de la brújula digital con la tarjeta FPGA se hizo una prueba verificando con una brújula tradicional que el dato entregado por la FPGA concordara con el leído en la brújula tradicional como se muestra en la sección de Resultados 2.1.2.

1.3. RECONOCIMIENTO DEL ENTORNO

El reconocimiento del entorno depende de la percepción sensorial del sistema, en este caso la percepción depende de la implementación de sensores Infrarrojos para la medición de distancias con respecto al ambiente de trabajo, pues es necesario que el sistema conozca la distancia que existe entre los diferentes componentes del entorno, sean obstáculos estáticos, móviles y paredes de superficie regular para poder elaborar mapas, y facilitar la navegación.

Dentro de la literatura de medición de distancia con infrarrojos se tiene que para superficies reflectoras que formen un ángulo de 0 grados con el sensor de infrarrojos se tendrá una medida acertada, conociendo también las características de la superficie reflectora. Lo que lleva a pensar, ¿que ocurre cuando el ángulo de incidencia deja de ser de 0 grados?, o que ocurre cuando no se conoce las características de la superficie reflectora?; antes que nada la medición estará acompañada de un error que depende directamente de la inclinación de la superficie reflectora y sus características , teniendo en cuenta este aspecto, se podría afirmar que al momento de hacer una medida, se deben considerar, qué variables existen al efectuar una medición, entre ellas se tiene que la energía que refleja el objeto depende principalmente de la distancia a la que se encuentre, del coeficiente de reflexión del objeto (dependiente a su vez del color del objeto, del brillo y de la textura) y del ángulo de incidencia del haz sobre la superficie reflectora , debe considerarse además la presencia de fuentes de luz sea solar y/o artificial, que también adicionan un componente de ruido.

El modelo que ha solucionado una parte del objetivo de medir distancia con dispositivos infrarrojos esta dado por el modelo de Yair [43] y la técnica de fusión bi-sensorial mencionada en la sección de Preliminares 4.1.3.2; De acuerdo a lo anterior, para conocer la distancia entre el sensor y el objeto reflector es primordial conocer a priori el ángulo de incidencia θ y la característica de reflexión αi de la superficie del objeto.

1.3.1. MEDIDOR DE DISTANCIA DE LABORATORIO.

Este medidor de distancia es un sistema cuyo montaje fue realizado con componentes del Laboratorio de Electrónica del Programa de Ingeniería Eléctrica de la Universidad Tecnológica de Pereira, el sistema posee un circuito de emisión y un circuito de recepción. El circuito de recepción entrega una variación en la amplitud de la tensión de salida proporcional a la distancia de la superficie reflectora.
Para emitir la señal de trabajo se usa el dispositivo LD274 de Siemens emisor de infrarrojos el cual posee como características:

- Altamente directivo, con ángulo de emisión de +/-10°

- Rango de emisión entre 880 y 1060 nanómetros de longitud de onda.
- Capacidad de ajustar el rango de emisión con la variación de corriente, aplicando un tren de pulsos de corriente directa.
- Emisión de 950 nm con 100mA y un pulso de 20 ms.

Como característica de corriente de polarización para emitir se tiene la curva mostrada en el Grafico 5.

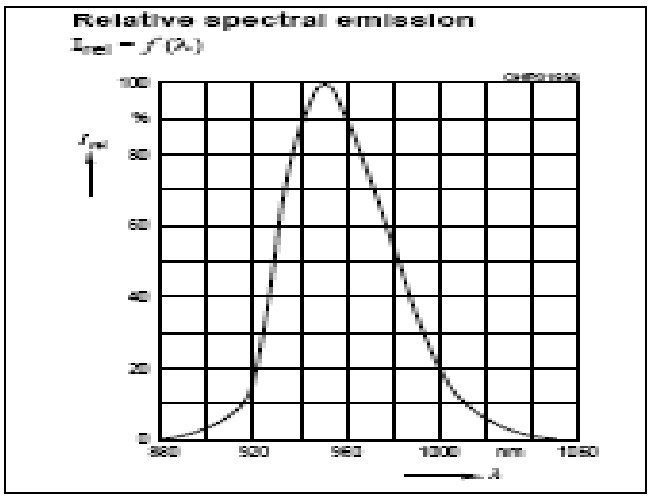


GRÁFICO 5. Corriente de Polarización Contra Longitud de Onda de Emisión.

Como característica de emisión en el plano de proyección se denota un alcance de 1 metro para un rayo directivo además un ángulo sólido de emisión de 10 grados hacia la periferia, mostrado en el Grafico 6.

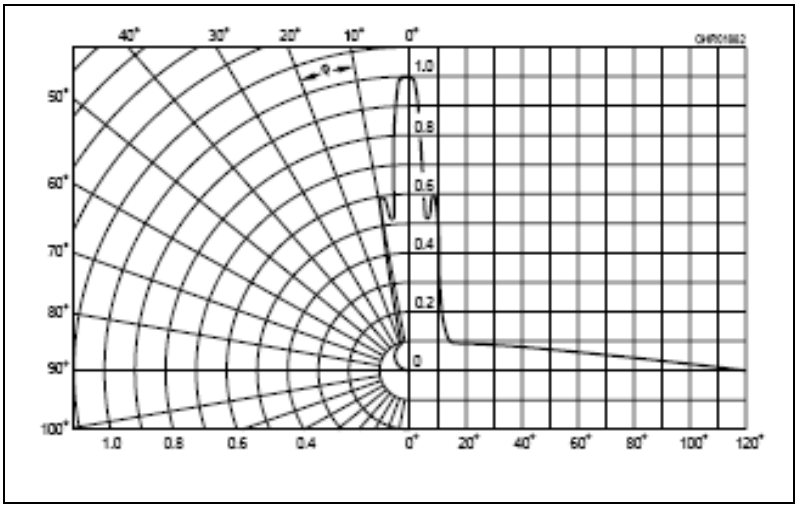


GRÁFICO 6. Característica de Emisión en el Plano.

Es necesario modular la emisión infrarroja a una frecuencia lo suficientemente alejada de las frecuencias en las que emiten las posibles fuentes de ruido, de tal forma que en la recepción de la señal ésta se pueda separar de las condiciones medioambientales. La frecuencia a la cual se ha modulado la señal es de 40 kHz, que se encuentra en el rango de frecuencias de funcionamiento en las que son utilizados dispositivos de control remoto convencionales que suelen estar entre 30 y 40 kHz.

La envolvente de la señal esta dada con una frecuencia de 500 Hz y como se mencionó anteriormente la señal modulada a 40 kHz.
El comportamiento del sensor es estable cuando es excitado con la presentación de la envolvente, para el efecto de medición de distancia para aumentar el alcance de detección.

Establecer una característica especial en frecuencia como se muestra en el Grafico 7, de la señal a emitir es muy importante puesto que esta señal al ser emitida al medio, viaja al igual que otras ondas, para que al recibir la señal, se filtre sólo la que el emisor está enviando al ser reflejada por diferentes obstáculos. Y así saber que la señal de trabajo recibida sea la emitida por el sistema en cuestión.

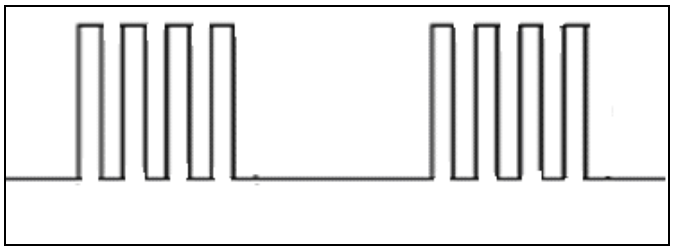


GRÁFICO 7. Característica de Señal a Transmitir

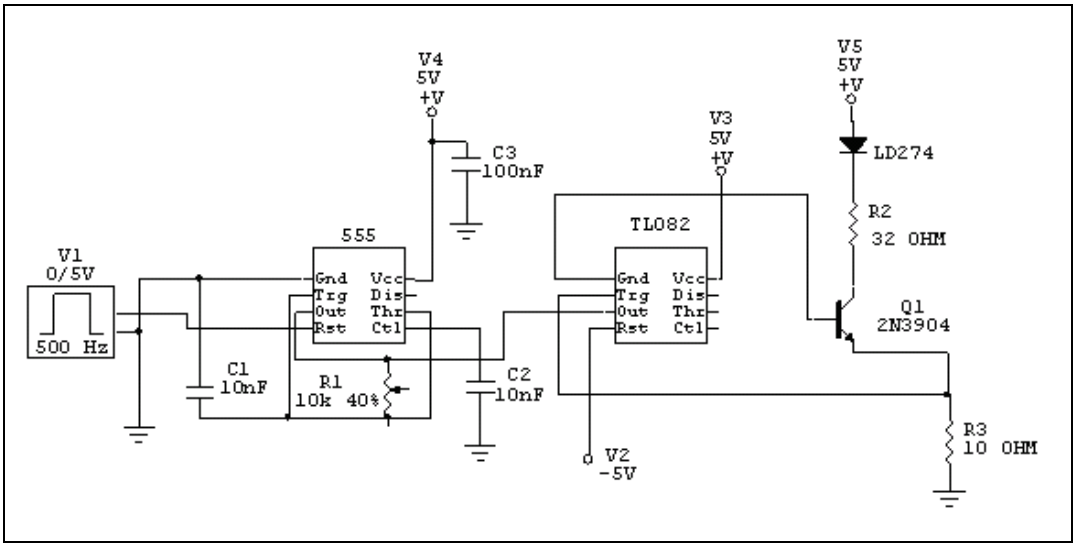
El circuito receptor tiene varios factores a considerar, el entorno es el asunto principal. Compitiendo con la débil señal infrarroja transmitida, están las fuentes de luz que en comparación, tienen potencias mucho mayores, como pueden ser las lámparas fluorescentes, lámparas incandescentes y la luz del sol.
Esto puede contribuir al problema de dos maneras:

- Producen un nivel de estimulación ambiente al sensor que aparece como una componente de frecuencia cero (DC), debido a fuentes de luz como la luz solar, que puede causar una disminución de la sensibilidad.
- Proporcionan un nivel de ruido cerca de 60 db mayores que la señal deseada, especialmente en la forma de 100 Hz de frecuencia debido a la corriente AC en las fuentes de iluminación artificial como por ejemplo las lámparas incandescentes o bombillas. Es necesario recordar también que la sensibilidad de los detectores de silicio se extiende hasta bien superado el rango visible. Añadidas al componente visible, ambos producen grandes cantidades de energía infrarroja, especialmente la luz solar.

En estos casos se necesita excitar la fuente de emisión a una frecuencia lo suficientemente alejada de las fuentes de luz ambiente. Esto permite que la amplificación de la señal detectada y el filtrado de los componentes de baja frecuencia sean más sencillos.

1.3.1.1. EMISIÓN.

El sistema de emisión de infrarrojos consta de 2 dispositivos infrarrojos LD274, dos transistores de propósito general 2N3904 polarizados para trabajar como interruptores sólidos, una fuente de corriente simétrica DC de +/-5V, dos resistores conectados al emisor y colector respectivamente del transistor y una fuente generadora de una señal modulada que consta de un oscilador con circuito integrado 555 y fuente de generadora de señal, también un acoplador de impedancia MOSFET TL-082 entre la base del transistor y la fuente de señal modulada.
Ver Esquemático 6



ESQUEMÁTICO 6. Circuito de Emisión de Laboratorio.

1.3.1.2. RECEPCIÒN DE SEÑAL

Recibir la señal de trabajo del medio es una tarea de vital importancia, puesto que la base para trazar mapas de entorno y navegación es la información que entregue el sistema de percepción sensorial.

El dispositivo receptor de infrarrojos de trabajo es el SFH203FA, un fotodiodo que para usarlo como fotorreceptor debe polarizarse inversamente, la magnitud de la señal que entrega el receptor es de orden de milivoltios, antes de acondicionar la señal de trabajo, debe conocerse el comportamiento del receptor en diferentes condiciones para saber qué componentes afectan la señal y minimizarlos o eliminarlos. Componentes de frecuencia de 0 Hz (componente DC) y frecuencias provenientes de fuentes no deseadas como las lámparas fluorescentes, etc.

El foto-receptor SFH203FA posee como características de funcionamiento:

- Ángulo de recepción de +/- 20º.
- Voltaje en circuito abierto entre 300 y 370 mV.
- Cuando recibe energía con longitud de onda de 950 nm alcanza la sensibilidad máxima.

El Grafico 8 muestra la sensibilidad del receptor con respecto a excitaciones con diferente longitud de onda, como se observa entre 800 y 1100 nanómetros de longitud de onda el receptor tiene un porcentaje de excitación entre 70 y 100 %, si se retoma la característica de emisión del LD274, con 100 mA de corriente se tiene una longitud de onda 950 nm y esta longitud de onda esta ubicado entre el rango de sensibilidad de mayor porcentaje para excitación del receptor.

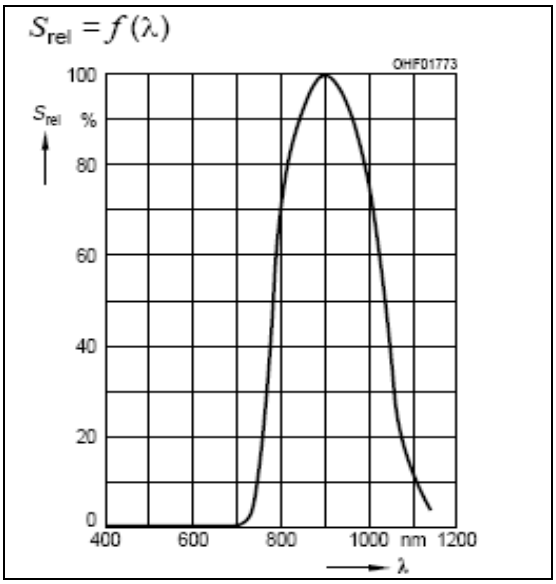
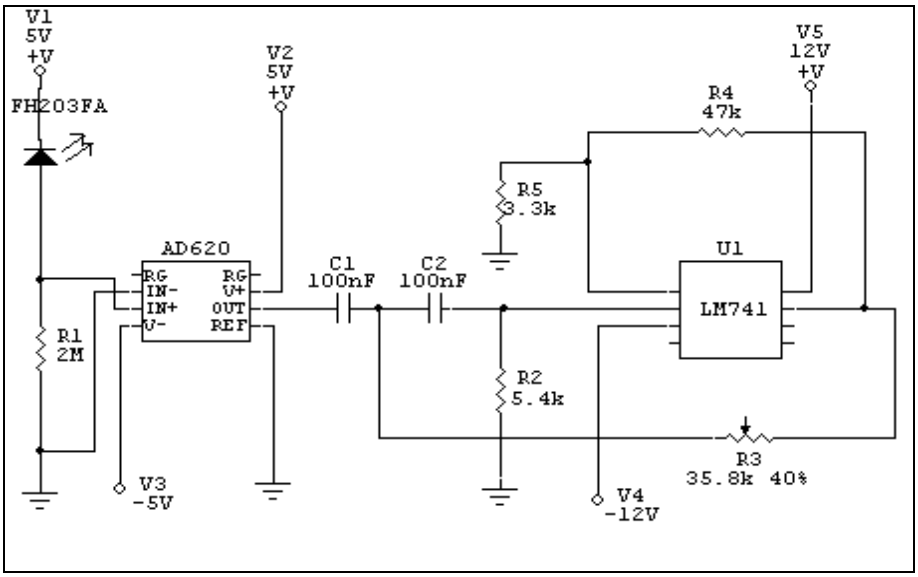


GRÁFICO 8. Sensibilidad del Receptor contra Longitud de Onda

La señal proveniente del foto-receptor se adquiere por medio de un divisor de tensión con un resistor de valor de impedancia aproximado al del dispositivo infrarrojo, la señal del divisor de tensión se acondiciona en dos etapas, la primera pasa la señal por un amplificador de instrumentación AD620 el cual le da potencia a la señal, luego se pasa por la etapa de filtrado en la que se elimina la componente de ruido por medio de un filtro activo pasa-alto con un amplificador operacional LM741 con una frecuencia de corte de 100Hz y con ganancia de 15. Ver Esquemático 7.



ESQUEMÁTICO 7. Circuito de Recepción de Laboratorio.

La posición de los dispositivos infrarrojos tanto del receptor como del emisor depende directamente del tipo de aplicación; como detección, control, y en este caso medición de distancia.

La cantidad de energía que logra excitar el foto-receptor depende directamente de la posición del emisor y la superficie reflectora con respecto al receptor.

Ambos, receptor y emisor están posicionados en la misma dirección.

1.3.1.3. PRUEBA DE LABORATORIO.

Equipos necesarios para la prueba:

1. Osciloscopio *PROMAX*, OD-581.
2. Cable de comunicación serial RS-232, *MODEM*.
3. Generador de audio *STANDFORD RESEARCH SYSTEMS*.
4. Fuente de corriente DC, *PROMAX* FA-851 *power supply*.
5. Dos sondas para osciloscopio.
6. Una sonda para generador de audio.
7. Superficies reflectoras (papel bond, cartulinas de diferentes color, madera, verde mate)
8. Luxómetro digital *EXTECH* – instruments, *LIGHT METER*.
9. Multímetro digital *FLUKE.*

Antes de iniciar las pruebas de caracterización debe hacerse un montaje sobre los emisores y el receptor; direccionarlos y aislarlos, principalmente el receptor que sólo puede tener descubierta el área activa de la cápsula, para que la prueba sea exitosa.

1. Prueba con ángulo de incidencia de 0 grados:

Se tomaron 50 muestras por cada situación para diferentes superficies reflectoras, y cambio en la distancia en intervalos de 5 centímetros hasta 80 centímetros considerando el ángulo de incidencia fijo en 90 grados con respecto a la normal y nivel de iluminación fijo.
La naturaleza de las muestras es de amplitud de voltaje las cuales varían de acuerdo a la distancia que sea posicionada la superficie reflectora. Ver figura 27.

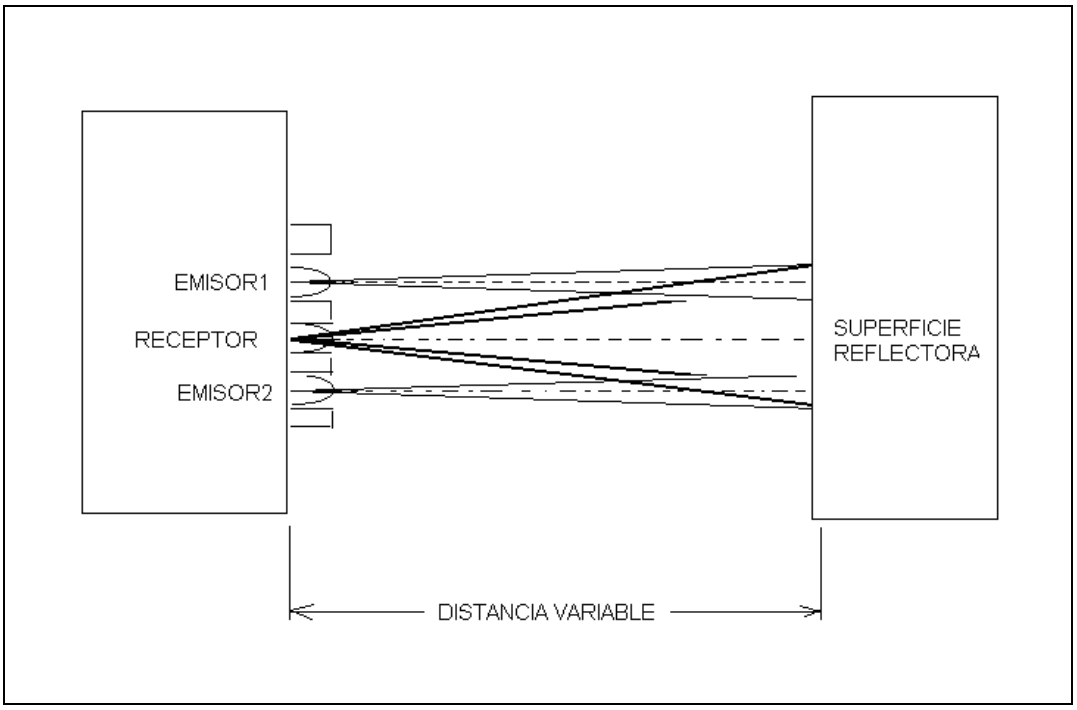


FIGURA 27. Vista Lateral. Descripción de Arreglo Sensorial para Prueba.

Superficies reflectoras utilizadas en la prueba de laboratorio:

- Papel Periódico
- Papel Bond
- Madera Barnizada
- Verde Mate
- Cartulinas:
    - Rojo
    - Rosa
    - Azul Cielo
    - Amarillo Claro
    - Verde
    - Negro
    - Blanco

El comportamiento del medidor de distancia de laboratorio para el escenario de la prueba mostrado en la Figura 27 se muestra en la sección de Resultados 2.1.3.

2. Prueba con variación en el ángulo de incidencia y variación de distancia.

La prueba se hizo sobre un plano marcado con coordenadas de grados para ajustar la superficie reflectora con la inclinación de ángulo de incidencia para registrar las salidas de los sensores en amplitud de voltaje.

La superficie reflectora debe ocupar el área necesaria para que los sensores infrarrojos puedan tener alcance de reflexión.

Los datos capturados son los de amplitud máxima debido a que la señal de excitación por parte de los emisores es una señal envolvente de 500 hz y 40 khz como se mencionó en la sección 1.3.1.1.

La prueba se hizo rotando la superficie reflectora en pasos de 15 grados con la distancia al sensor constante como se muestra en la Figura 28 y capturando los datos de salida del mismo. La distancia se ajusta en pasos de 5 cm hasta 80 cm, y con la superficie reflectora de color blanco.

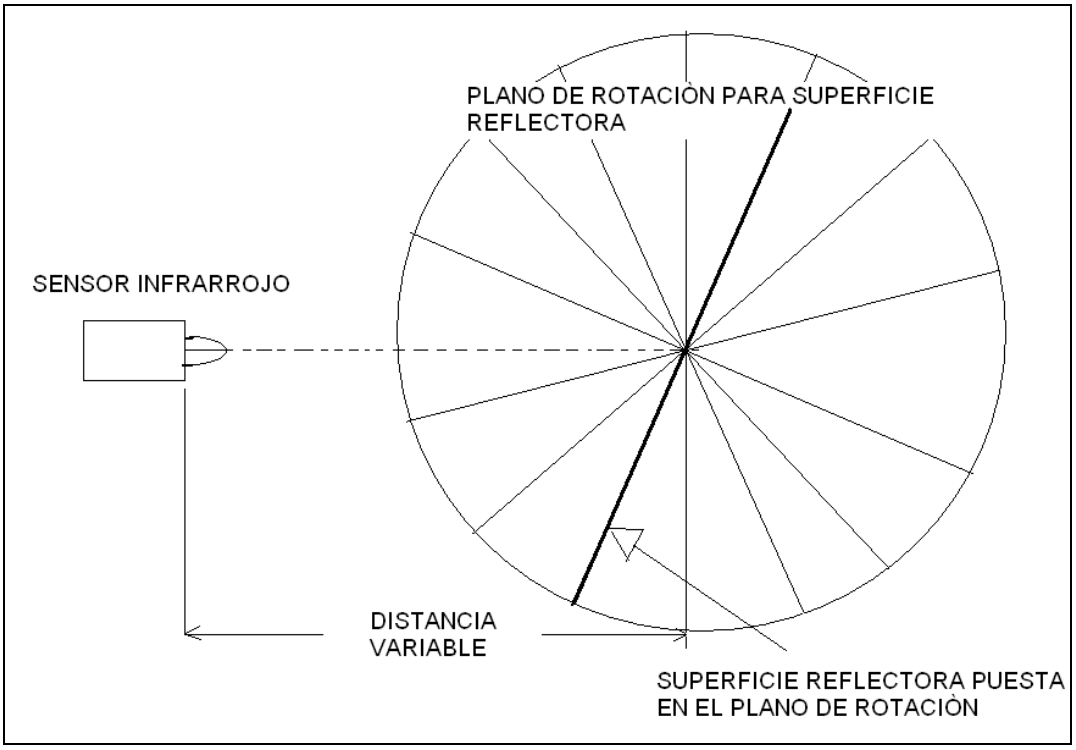


FIGURA 28. Vista Superior de Prueba de Laboratorio con Variación del Ángulo de Incidencia

El comportamiento del sensor con variación en el ángulo de incidencia de la superficie reflectora se muestra en la sección de Resultados 2.1.4

El sensor entrega voltajes análogos, por lo que se hace necesario usar un conversor Análogo-Digital pues la aplicación para el trazador de mapas de entorno y navegador en plataforma móvil requiere información digital para poder efectuar diferentes análisis, durante la prueba se usó como herramienta de medición de voltajes un osciloscopio y un PC para capturar los datos y procesarlos, el siguiente paso es la implementación del hardware reconfigurable FPGA para optimizar esta aplicación que es la medición de distancias con sensores infrarrojos.

1.3.2. OPTIMIZAR MEDIDOR DE DISTANCIA DE LABORATORIO, IMPLEMENTANDO HARDWARE RECONFIGURABLE, FPGAS SPARTAN 3E DE XILINX.

Es necesario optimizar el circuito medidor de distancia y ángulo de incidencia puesto que para su funcionamiento se necesitan dos fuentes duales, un generador de audio con un circuito integrado LM555; se denota una cantidad de hardware que para el anillo de sensores en la plataforma móvil ocuparía demasiado espacio. La implementación de hardware reconfigurable FPGAS permite optimizar aplicaciones en las que con un diseño que describa el funcionamiento del sistema completamente se elimine hardware que se hace innecesario.

1.3.2.1. GENERAR LA SEÑAL MODULADA

Para optimizar el medidor de distancia se empezó con, él como generar una señal modulada con la tarjeta SPARTAN 3E, se tiene que la tarjeta posee un cristal externo que funciona como reloj principal de 50 MHz

El proyecto es una aplicación para la implementación de la tarjeta de hardware reconfigurable, se diseñó en vhdl la descripción para generar la señal modulada, dentro de la descripción se generan dos divisores de frecuencia con frecuencias de 500 Hz. y 40 kHz respectivamente, como se puede ver en el Diagrama 8 y luego se hace un producto de ambas frecuencias generando así la señal modulada como se observa en Diagrama 9.

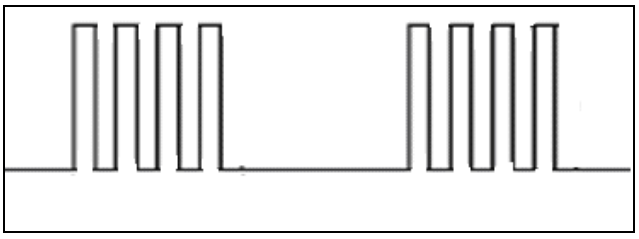


GRÁFICO 9. Señal de Excitación para Emisores de Infrarrojo

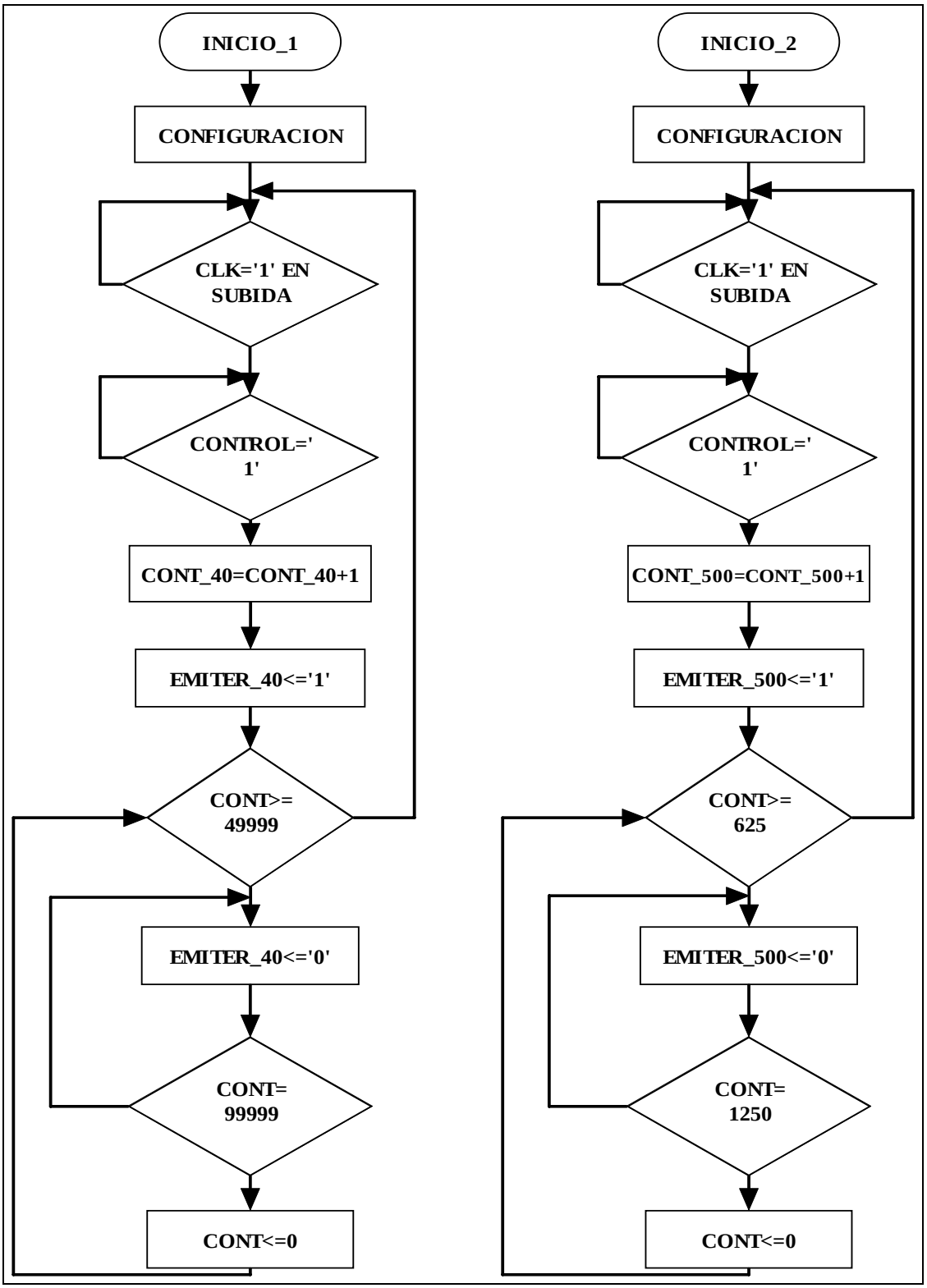


DIAGRAMA 8. Divisor de Frecuencias para 500 Hz y 40 kHz.

Se generan dos frecuencias de 40 kHz y 500 Hz, y se multiplican para dar como resultado la señal envolvente.

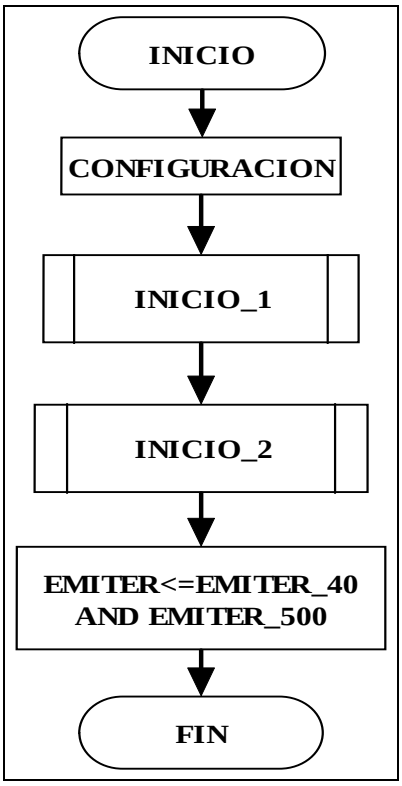


DIAGRAMA 9. Producto de Dos Frecuencias

La descripción de hardware que genera los trenes de pulsos con 500 Hz. y 40 KHz se presenta en el ANEXO D**.**

La descripción del ANEXO D funciona sólo con la señal de control en '1' lógico, una vez la señal esta activada la envolvente es generada para excitar el sensor y emitir una señal de rayos infrarrojos al medio, con el oscilador de 50 MHz de la SPARTAN 3E se inicia, el generador de 40 KHz de frecuencia con un contador de flancos de subida hasta 1250 pulsos y la salida designada por *emiter_signal_out* se coloca en '1' y se mantiene durante 1250 pulsos mas de la señal de reloj de 50 MHz para así generar la frecuencia de 40 KHz; luego el contador de ciclos se activa y cuenta 40 ciclos de la señal de *emiter_signal_out* y se cambia a cero durante 40 ciclos mas de la frecuencia generada de 40 KHz, para así obtener la envolvente de 500 Hz. con modulación a 40 KHz debido a que 40 pulsos de 40 KHz equivalen a un pulso de 500 Hz.

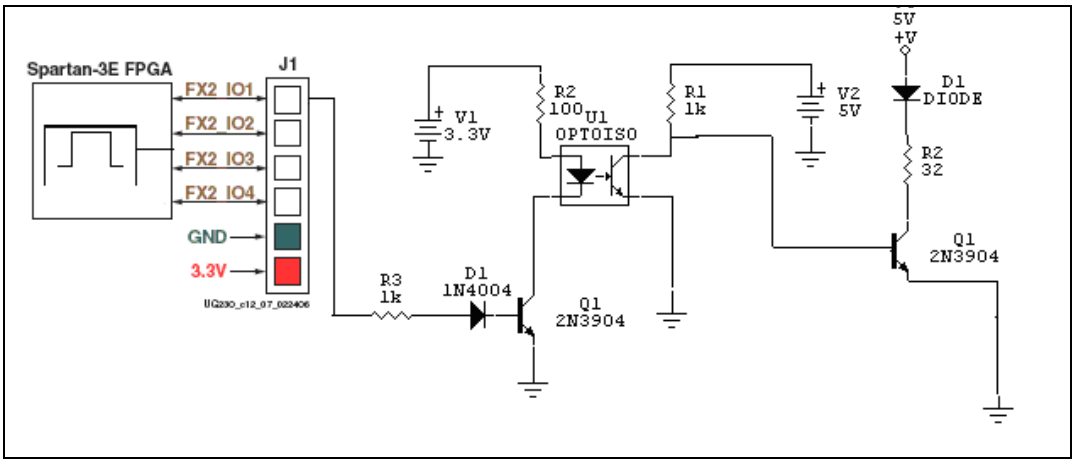
Se ha sustituido el generador de audio y el circuito modulador con integrado LM555, con la señal generada por la tarjeta FPGA spartan 3E de Xilinx.

La impedancia vista por la base del transistor Q1 del circuito emisor del esquemático 5 se hace baja sin tener que utilizar el acoplador de impedancia FET TL-082 por lo que se elimina del

sistema de emisión de infrarrojos y se conecta directamente del acoplador PC817 a la base de Q1.

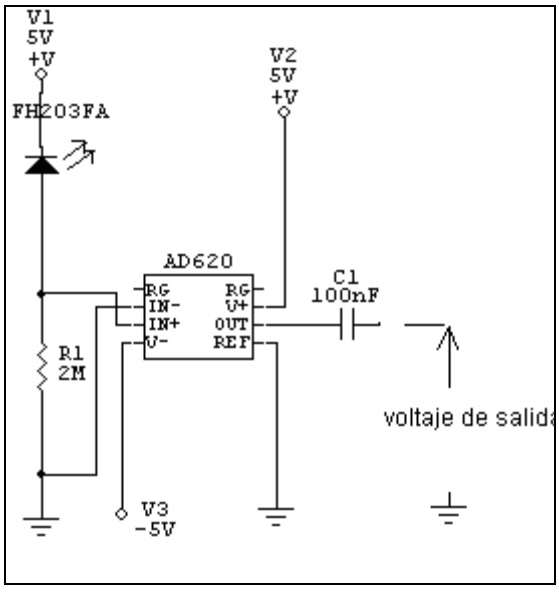
Para aumentar el nivel de potencia emitida por los sensores IR se tomó para cada emisor excitación independiente es decir por cada emisor hay un transistor 2N3904.

El sistema de emisión queda optimizado como se muestra en el Esquemático 8, a continuación para el circuito emisor del medidor de distancia.



ESQUEMÁTICO 8. Circuito para Excitación de Emisores de Infrarrojos

Se continúa con el circuito de recepción, el cual tiene una etapa de acondicionamiento y de amplificación; durante las pruebas se concluyó que mientras el receptor de infrarrojos esté protegido y tenga solo disponible su zona activa, el uso del filtro activo es innecesario. Ver esquemático 9.



ESQUEMÁTICO 9. Acondicionamiento Señal de Recepción de Infrarrojos.

1.3.2.2. ADQUISICIÒN DE SEÑAL, CONVERSIÓN ANÁLOGO-DIGITAL E INTERFAZ GRÀFICA CON PC.

Una vez adquirida la señal proveniente del SFH203FA, acondicionada por el amplificador de instrumentación AD620 procede a ser digitalizada. (Voltaje de salida en el Esquemático 9).

Para adquirir la señal proveniente del receptor de infrarrojos es necesario digitalizarla para hacer el tratamiento digital de la misma; se implementa un conversor Análogo/Digital, externo a la tarjeta contenido en un microcontrolador, el motivo para usar dispositivos externos a la tarjeta es que debido a la naturaleza inherente del arreglo de lógica programable para ejecutar procesos en concurrencia, es implementar el concepto en esta aplicación de plataforma móvil para adquirir en tiempo real, datos de lo que será el anillo de sensores Infrarrojos y otros periféricos, se hace muy importante para el área de investigación en navegación.

Se usa un microcontrolador Motorola de la familia HC08 JK8 el cual contiene 8 canales de conversión Análogo/Digital, se programa para que funcione a la velocidad máxima de muestreo que es 71.5 kHz es decir tiempo entre muestras de 14 us mas los ciclos de maquina requeridos para el almacenamiento en la RAM, teniendo en cuenta que la señal de trabajo es de 500 Hz (2 ms) el criterio de Nyquist se cumple para obtener una descripción de la señal muy cercana de la de trabajo; pero para tener mas información de la señal, se muestrea durante mas de 2 ms; para las pruebas de adquisición de señal se tomaron durante 12 ms, tiempo en el que la señal de 500 Hz a tenido 6 ciclos, estos 6 ciclos serán muestreados a la velocidad máxima de muestreo mencionada anteriormente para un total de 240 muestras tomadas en 12 ms; En el microcontrolador (MCU) debe implementarse una trama de comunicación para transmitir los datos convertidos a la tarjeta por medio de interfaces que convierten niveles de voltaje de 5 V a 3.3 V.

Los datos enviados a la tarjeta son adquiridos por una sola línea de entrada por lo que debe implementarse en VHDL la descripción para recibir una trama de datos provenientes del MCU, que al arribar al arreglo de hardware reconfigurable serán almacenados y procesados; el paso siguiente es verificar que la señal halla sido efectivamente adquirida enviando la cantidad de datos muestreados a una interfaz gráfica que permita ver que la señal halla sido efectivamente adquirida. El proceso a seguir es la adquisición de la señal por medio del conversor Análogo/Digital en el MCU, luego el envío de datos a la FPGA, y después la aplicación de la interfaz gráfica por medio de comunicación serial entre la SPARTAN 3E y el PC.

1.3.2.3. IMPLEMENTACIÒN MCU MOTOROLA HC08 JK8.

El MCU JK8 de Motorola se implementa para utilizar los canales de conversión Análogo-Digital y desarrollo de trama de envío de datos convertidos a la tarjeta; adquirir la señal siendo muestreada a 71.5 kHz tomando 100 muestras en un tiempo de 5 ms, como se sabe el MCU debe ejecutar una sola tarea, pues todos los procesos deben ser monitoreados desde la unidad central de procesamiento por lo que se consideran tres tareas a ejecutar dentro de MCU, conversión A/D, almacenamiento de datos, envío de datos a la SPARTAN 3E, además el MCU esta encargado de activar una bandera de salida a la tarjeta que indica que se esta ejecutando la transmisión de datos almacenados en la memoria RAM del MCU. Se muestra en el diagrama 10 la rutina de flujo que debe ejecutar el MCU y el código en lenguaje ensamblador se muestra en el ANEXO E y distribución de pines la Figura 29.

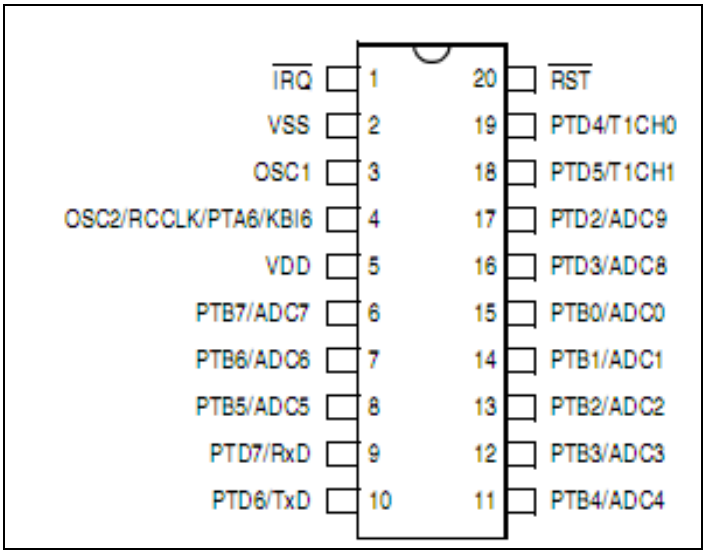


FIGURA 29. Distribución de Pines del MC68HC08JK8

La memoria RAM tiene una capacidad de 255 *bytes* pero sólo se ocupará una capacidad de 240 *bytes*, el MCU adquiere datos y los almacena hasta que se llene la memoria, para luego ser enviados a la tarjeta.

El MCU tiene como objetivo adquirir datos provenientes del conversor A/D almacenarlos en la memoria RAM y una vez llena con 240 datos, estos son enviados vía protocolo de comunicación serial RS-232 , el MCU refresca la memoria RAM y adquiere de nuevo datos del conversor A/D, la descripción anterior se estructura en el Diagrama 10.

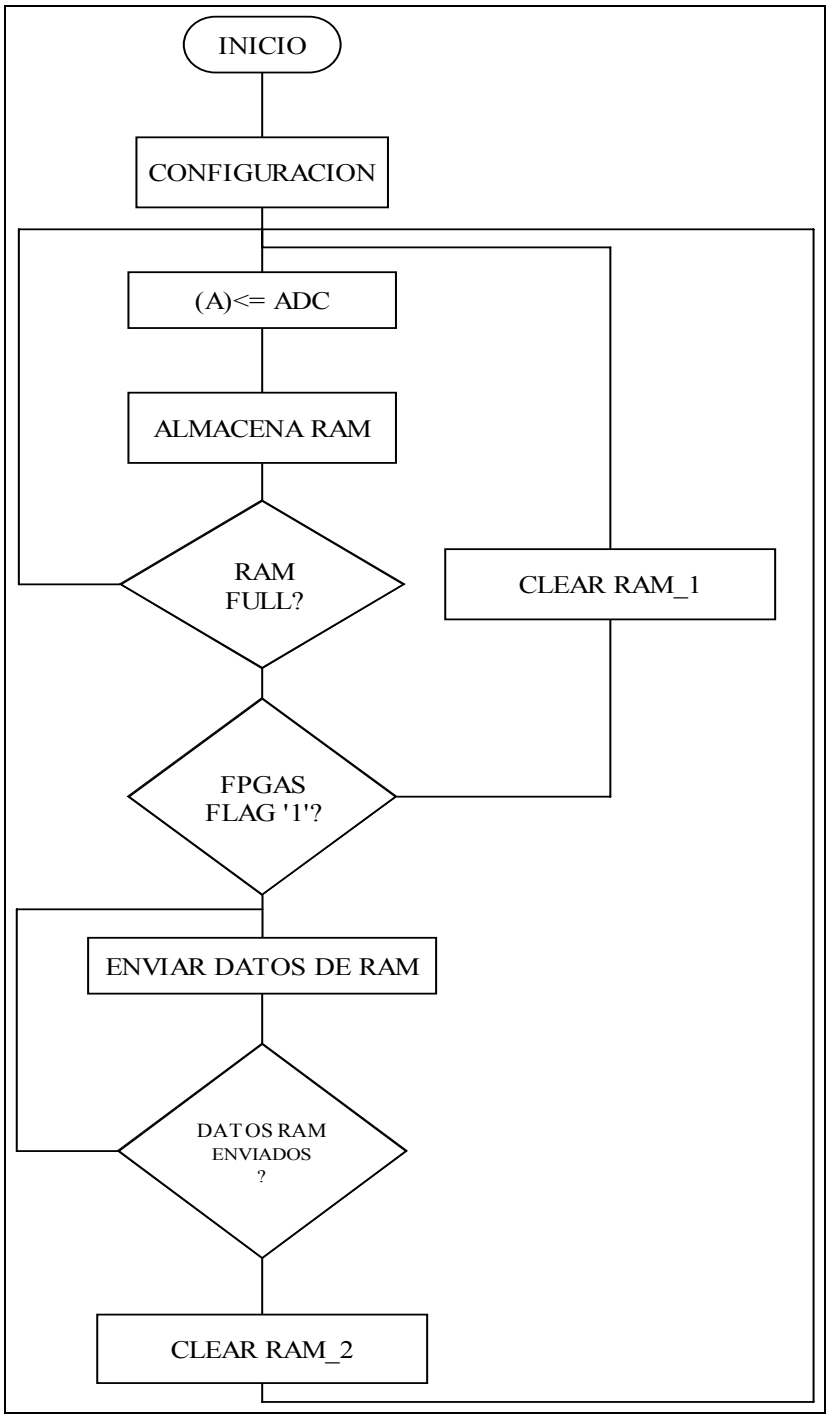


DIAGRAMA 10. Rutina de Conversión, Almacenamiento y Envío de Datos en el MCU

1.3.2.4. DESCRIPCIÒN PARA RECEPCIÒN DE DATOS DESDE EL MCU A LA TARJETA SPARTAN 3E.

Se establece un proceso para adquisición de datos hacia la tarjeta del MCU, el fundamento del proceso esta dado por el protocolo de comunicación serial rs-232.

La trama de comunicación serial consiste en un pin de información que transmite uno a uno los bits de inicio, datos, paridad y parada, el bit de inicio es de nivel bajo, el ancho de los datos es de 8 bits, el bit de parada es de nivel alto. Ver Figura 30.

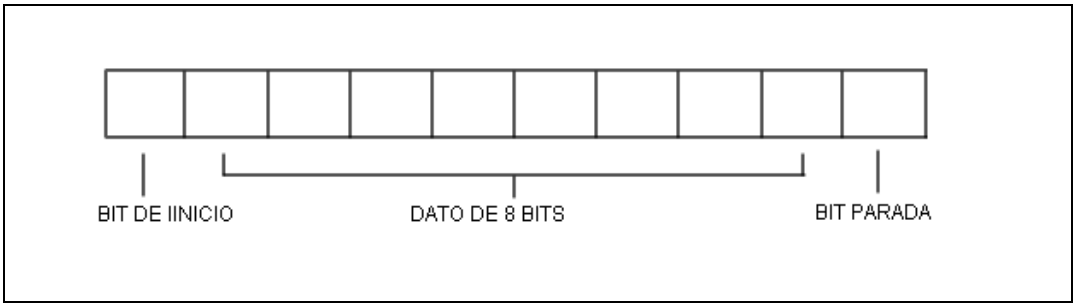


FIGURA 30. Trama de Comunicación Serial rs-232

Para efectuar la trama de Comunicación Serial para recibir datos del microcontrolador a la tarjeta FPGA se propone un bloque de programación mostrado en el Diagrama 11.

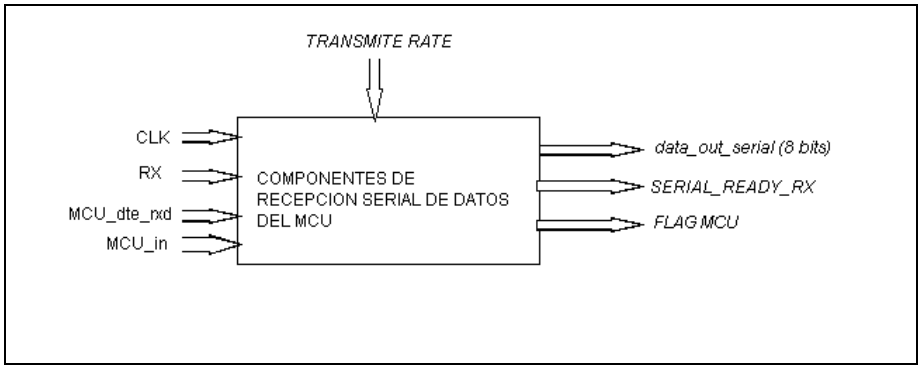


DIAGRAMA 11. Componente de Recepción Serial de Datos de MCU.

Donde Clk, rx, y MCU_dte_rxd, MCU_in representan las entradas del componente y *data_out_seria*l y *serial_ready_rx* son las salidas del componente. Adicionalmente *transmit_rate* es el un parámetro de tipo entero.

- Clk: Es la señal de reloj, que es directamente conectada desde el reloj externo de la tarjeta. Este reloj es de 50 MHz y es representado por un solo bit. El pin de la FPGA donde esta conectado este reloj es el C9.

- RX: Es una señal de habilitación del componente y es representada por un solo bit, cuando rx esta en '1' el componente esta activo y esperando recibir un dato por comunicación serial.

- MCU_tx: Es la señal de comunicación serial y es representada por un solo bit.

- MCU_IN: Bandera que indica a la SPARTAN 3E que esta ocurriendo envío de datos.

- *Data_out_serial:* Es el dato de 8 bits que se recibió y esta disponible siempre, este se actualiza cada que serial_ready_rx se ponga en '1'.

- *Serial_ready_rx*: Es la señal que indica que la recepción de un dato de 8 bits a culminado, es de un bit, y se pone en '1' durante 20ns, con el fin de actualizar el dato de salida.

- *Transmit_rate*: Este parámetro de tipo entero es el que indica la velocidad o rata de transferencia.

En el Diagrama 12 se muestra la estructura de control para el componente de recepción de datos serial.

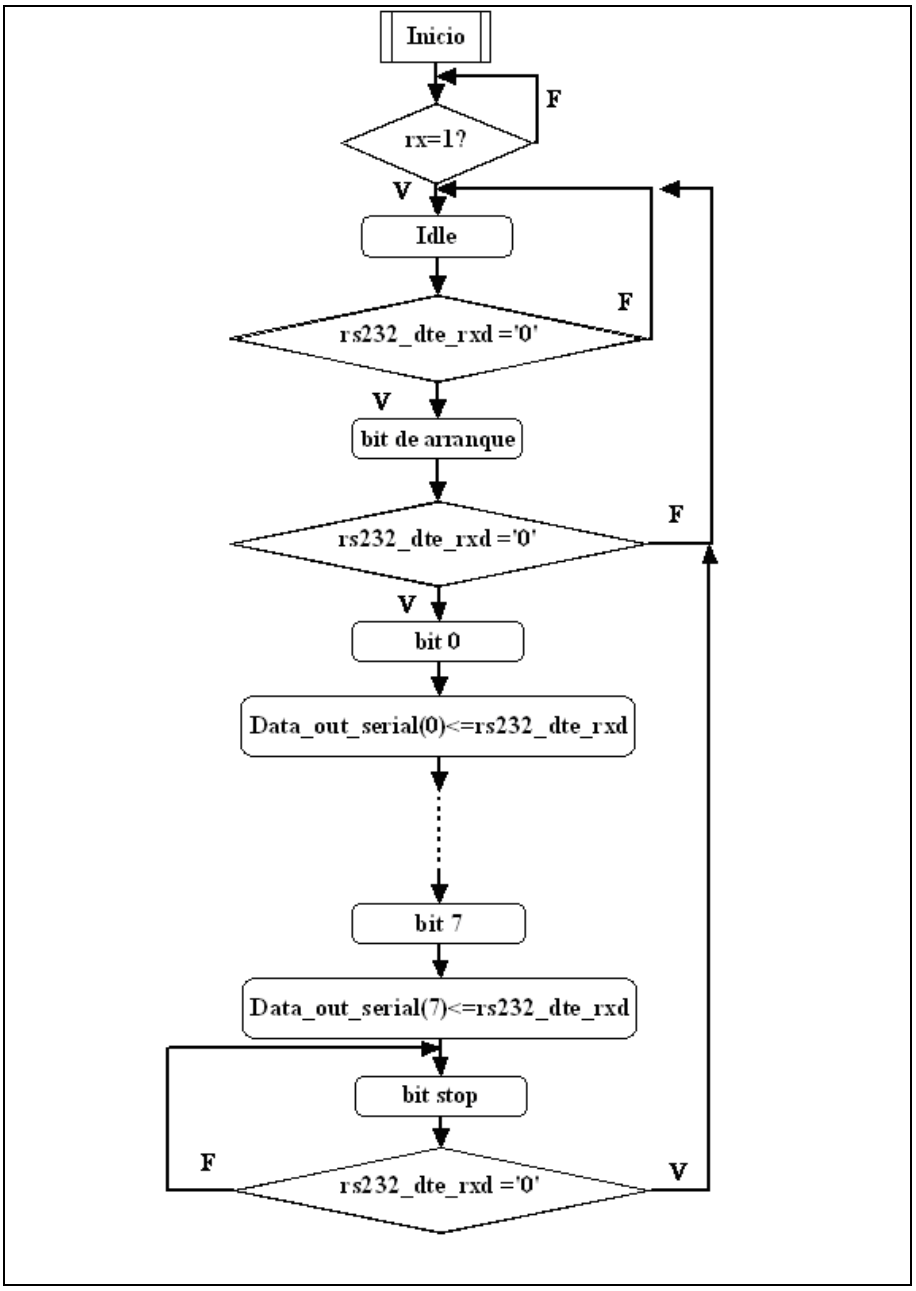


DIAGRAMA 12. Flujo para Recepción de Datos Serial

El componente de recepción serial se encarga de esperar un dato por comunicación serial en la entrada designada. Dado que el dato recibido llega en formato de un bit en serie, y tiene un total de 10 bits, donde se encuentran el bit de arranque, los 8 bits de información y el bit de parada, es necesario realizar una maquina de estados que organice uno a uno los bits de información en el dato final.

La señal rx, habilita la máquina de estados para realizar la recepción, en el momento que dicha señal se encuentra en '1' la máquina de estados se ejecuta, comenzando por el estado de *idle* o espera, en dicho estado el bit que se encuentra en la línea de recepción *MCU_tx* es '1' , en el

momento en que este bit sea ‘0’ significa que la comunicación a comenzado, dicho bit es llamado el bit de arranque, dado que existe la posibilidad de que halla un cambio de uno a cero por una fluctuación, es necesario confirmar que realmente este es el bit de arranque, para lo cual se revisa el estado de este valor un tiempo después, este tiempo es determinado según la rata de transferencia, para este proyecto la rata de transferencia se determino en 7812 baudios, lo cual significa que el tiempo que se demora un solo bit es de un microsegundo considerando que los tiempos dentro de la FPGA deben estar referenciados a el reloj principal de 50Mhz, o sea que 128 microsegundos son 6400 oscilaciones, y por lo tanto el momento para revisar el bit de arranque es igual al tiempo de medio bit, que son 3200 oscilaciones. De ser falso el bit de arranque se regresa al estado de bit de espera, hasta que ocurra un nuevo cambio de uno a cero, en caso contrario significa que ha comenzado una transmisión, y se procede a recibir el primer bit, que para el caso de la comunicación serial es el bit menos significativo, para asegurar cual es el valor de este bit, se realiza la lectura después de transcurrido el tiempo de un bit (6400), a partir de la ultima lectura, por tanto siempre se leen los valores de los bits de información en la mitad del tiempo, esto se hace para los 8 bits, considerando que el 8 bit es el mas significativo, cada que se recibe un bit este se almacena en la señal de 8 bits *Data_out_serial*, por medio de un direccionamiento, en otras palabras, para cada bit existe un estado, y en ese estado se va a almacenar un bit en particular, y así direccionarlos todos correctamente.

Una vez se ha recibido el octavo bit y se ha almacenado, transcurre un tiempo de un bit (6400) revisa nuevamente la señal *MCU_tx*, en ese momento el valor debe ser de ‘1’, que significa que es el bit de paro, si esto no es verdad, no se cambia de estado hasta que no compruebe, de ser verdad se pasa al estado inicial de espera y se activa la señal *Serial_ready_rx* para actualizar el dato recibido, donde la maquina realiza el mismo procedimiento. Es de anotar que el ultimo dato recibido estará disponible hasta que se realice satisfactoriamente la recepción de un nuevo dato y *Serial_ready_rx* se encuentre en ‘1’.
Se muestra la descripción de hardware para la comunicación serial en el ANEXO F.

Una vez recibido el dato enviado por el MCU se direcciona a la memoria RAM de la tarjeta, y se guarda en una posición de memoria, se recibe el siguiente dato en la posición de memoria siguiente; terminada la adquisición de datos y almacenados, estos son enviados al PC como interfaz gráfica, vía comunicación serial para verificar que la adquisición de datos se haya ejecutado correctamente, lo anterior se usa solo como prueba ejecución.

1.3.2.4. IMPLEMENTAR MEMORIA RAM DE LA SPARTAN 3E.

La tarjeta FPGA Spartan 3E para aplicaciones que necesiten uso extenso de datos provee bloques de memoria que permiten configurarse para determinada aplicación. Los bloques de memoria poseen como característica la configuración de doble puerto y configuración simple.

Bloques de memoria RAM (BRAM). Toda la generación de FPGA Spartan-3 contienen múltiples bloques de memoria RAM organizados en columnas. La cantidad total de bloques de memoria RAM depende del tamaño de la generación de FPGA Spartan-3. La Tabla 2 muestra los bloques de memoria disponibles para la Spartan 3E.

Device	RAM Columns	RAM Blocks per Column	Total RAM Blocks	Total RAM Bits	Total RAM Kbits
XC3S100E	1	4	4	73728	72
XC3S250E	2	6	12	221184	216
XC3S500E	2	10	20	368640	360
XC3S1200E	2	14	28	516096	504
XC3S1600E	2	18	36	663552	648

TABLA 2. Bloques RAM Disponibles en el Dispositivo Spartan 3E.

Para el dispositivo XC3S500E se tienen dos columnas de 10 bloques cada una, o sea un total de 20 bloques RAM, con un total de 368640 bits, o 360kbits.

Cada bloque RAM contiene 18432 bits de RAM rápida estática, 16kbits que son dedicados al almacenamiento y en algunas configuraciones de memoria 2kbits adicionales son utilizados para paridad o datos de bits adicionales. Físicamente Los bloques de memoria RAM tienen dos bloques de memoria de acceso completamente independiente, etiquetados como Puerto A y Puerto B y se puede observar en Figura 31. La estructura es totalmente simétrica y ambos puertos son intercambiables y ambos puertos soportan operaciones de lectura y escritura. Cada puerto de memoria es síncrono con su propio reloj, habilitador de reloj y habilitador de escritura. La operación de lectura también es sincrónica y requiere de flancos de reloj y habilitación de reloj.

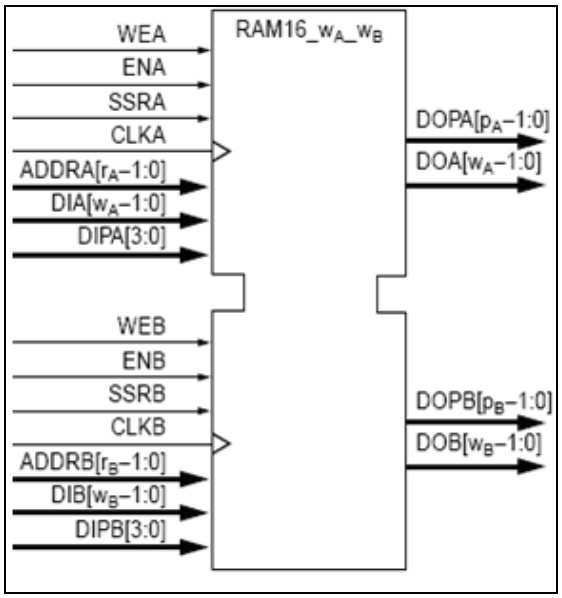


FIGURA 31. Memoria Doble Puerto

Descripción de señales. Las señales conectadas a los bloques de memoria RAM se dividen en cuatro categorías, como se lista a continuación.

1. Entradas y salidas de datos.
2. Entradas y salidas de paridad, disponibles cuando el dato del puerto es mas largo.
3. Entradas de direcciones para seleccionar la localización especifica de memoria.
4. Varias señales de control que manejan lectura, escritura u operaciones de set y reset.

La descripción de las señales presentes al momento de implementar el bloque de memoria doble puerta se ven en la Tabla 3.

Signal Description	Single Port	Dual Port		Direction
		Port A	Port B	
Data Input Bus	DI	DIA	DIB	Input
Parity Data Input Bus (available only for byte-wide and wider organizations)	DIP	DIPA	DIPB	Input
Data Output Bus	DO	DOA	DOB	Output
Parity Data Output (available only for byte-wide and wider organizations)	DOP	DOPA	DOPB	Output
Address Bus	ADDR	ADDRA	ADDRB	Input
Write Enable	WE	WEA	WEB	Input
Clock Enable	EN	ENA	ENB	Input
Synchronous Set/Reset	SSR	SSRA	SSRB	Input
Clock	CLK	CLKA	CLKB	Input

TABLA 3. Descripción de Señales del Bloque de Memoria.

- Datos de entrada y de salida. El ancho total de los datos de los puertos incluyen ambos, el bus de datos y el bit de paridad, cuando es aplicable, para el caso de este proyecto se usa una organización de tipo 2k9, el dato del puerto es de 9 bits e incluye 1 bit de paridad el cual es el bit mas significativo, seguido por 8 bits de datos que son los menos significativos.

Los datos de entrada, la paridad y las señales de salida son siempre buses; y son representados como DI y DO respectivamente.

- Bus de datos de entrada es DI [7:0] (DIA [7:0], DIB [7:0]). El bus de datos de entrada es la fuente de datos que se va a escribir en la RAM. Los datos en el bus DI son escritos en la localización de la RAM especificada por el bus de entrada de direcciones, ADDR, durante un transición de bajo a alto en la entrada de reloj CLK, cuando las entradas del habilitador de el reloj EN y el habilitador de escritura WE son altas.
- Bus de salidas DO [7:0] (DOA [7:0], DOB [7:0]). El bus de salida de datos, DO, presenta el contenido de las celdas de memoria referenciadas por el bus de direcciones ADDR, en un flanco activo de reloj durante una operación de escritura. Durante una operación simultanea de escritura, el comportamiento de los datos de salida es controlado por los atributos del modo de escritura *WRITE_MODE.*

- Entradas y salidas de paridad. La paridad solo es soportada para datos ampliados. Aunque se da referencia como bits de paridad, las entradas y salidas de paridad no tienen ninguna función en especial y pueden ser usados como bits de datos adicionales. Por ejemplo, los bits de paridad pueden ser usados para dar información adicional sobre el dato, el código del dato, valores positivos o negativos, valores viejos y nuevos, etc.

Los bloques de memoria no contienen ningún circuito especial para generar chequeo de paridad, estas funciones de ser requeridas en una aplicación deben ser creadas por medio de recursos lógicos.

- Datos de entrada de paridad DIP [8] (DIPA [8], DIPB [8]). Los datos del bus de entrada DIP son escritos en la localización de la RAM especificada por el bus de entrada de direcciones, ADDR, durante una transición de bajo a alto en la entrada de reloj, cuando las entradas del habilitador del reloj EN y el habilitador de escritura WE son altas.

- Datos de salida de paridad DOP [8] (DOPA [8], DOPB [8]). El bus de salida de datos, DOP, presenta el contenido de las celdas de memoria referenciadas por el bus de direcciones ADDR, en un flanco activo de reloj durante una operación de escritura. Durante una operación simultanea de escritura, el comportamiento de los datos de salida es controlado por los atributos del modo de escritura *WRITE_MODE.*

- Entradas de dirección. Como una RAM doble puerto, ambos puertos operan independientemente mientras se accede a el mismo set de celdas de memoria de 18kbit

- Bus de direcciones ADDR [10:0] (ADDR [10:0], ADDR [10:0]). El bus de direcciones selecciona la celda de memoria para operaciones de escritura o lectura, el ancho de el bus de direcciones depende de el tipo de configuración que se tenga, para este proyecto es de 11 bits.

ENTRADAS DE CONTROL.

- Reloj CLK (CLKA, CLKB). Cada puerto es totalmente síncrono con pines de reloj independientes, Todos los pines de entrada del puerto tienen un tiempo de configuración referenciado al pin del puerto CLK. El bus de datos tiene un clk-to-out tiempo de referenciado a el pin de CLK.
Con la polaridad predeterminada, una transición de bajo a alto en el reloj CLK, las entradas de control leen, escriben y hacen operaciones de reset.

- Habilitador EN (ENA, ENB). La entrada de habilitación, EN, controla la lectura, escritura y operaciones de set/reset. Cuando EN es bajo ningún dato es escrito y las salidas DO y DOP retienen el último estado. La polaridad de EN es configurable y es activo alto por predeterminado.
Cuando EN es activado, excepto en una entrada de set/reset síncrono o entrada de habilitación de escritura. El bloque RAM siempre lee la localización de memoria especificada por el bus de direcciones ADDR, en el flanco de subida del reloj.

- Habilitador de escritura WE (WEA, WEB). Las entradas de habilitación de escritura, WE controlan cuando el dato es escrito en la RAM. Cuando ambos. EN y WE son activados en el flanco de subida del reloj, el valor en el bus de entrada de datos y paridad es escrita en la locación de memoria seleccionada por el bus de dirección.
El dato de salida es cargado o no cargado dependiendo de los atributos de WRITE_MODE. La polaridad de WE es configurable, y es activa alta por predeterminado.

- Set/Reset síncrono SSR (SSRA, SSRB). La entrada de set/reset síncrono, SSR, fuerza el dato de salida al valor especificado por el atributo SRVAL. Cuando SSR y la señal de habilitación, EN son altos, el dato de salida para DO y DOP son síncronamente colocados en '1' o '0' de acuerdo a el parámetro SRVAL.
Una operación de set/reset síncrono no afecta las celdas de memoria RAM y no molesta las operaciones de escritura en el otro puerto.
La polaridad de SSR es configurable y es activo alto por predeterminado.

- Set/Reset global GSR. La señal de set/reset global GSR es activada automáticamente y momentáneamente al final de la configuración del dispositivo. Por medio de un instanciamiento al inicio primitivo, la aplicación lógica puede también activar GSR para restaurarse como al estado del inicio de la Spartan 3e en cualquier momento. La señal GSR inicializa la salida al valor inicial. Una señal GSR no tiene impacto en el contenido interno de memoria. Dado que GSR es global y automáticamente conectada con el dispositivo, el bloque RAM primitivo no tiene un pin de entrada GSR.

La Tabla 4 ilustra las diferentes configuraciones de memoria, con sus nombres, tamaño de buses de datos de entrada, de salida, direcciones y paridad.

Organization	Memory Depth	Data Width	Parity Width	DI/DO	DIP/DOP	ADDR	Single-Port Primitive	Total RAM Kbits
512x36	512	32	4	(31:0)	(3:0)	(8:0)	RAMB16_S36	18K
1Kx18	1024	16	2	(15:0)	(1:0)	(9:0)	RAMB16_S18	18K
2Kx9	2048	8	1	(7:0)	(0:0)	(10:0)	RAMB16_S9	18K
4Kx4	4096	4	-	(3:0)	-	(11:0)	RAMB16_S4	16K
8Kx2	8192	2	-	(1:0)	-	(12:0)	RAMB16_S2	16K
16Kx1	16384	1	-	(0:0)	-	(13:0)	RAMB16_S1	16K

TABLA 4. Configuración de Bloques de Memoria.

Control bloque de memoria RAM. El control de los bloques de memoria RAM se puede modelar basado en la configuración de memoria doble puerto, recordando que la configuración que se utiliza en la aplicación es 16_s9_S9, o 2Kx9.

El Diagrama 13 contiene la estructura del bloque de memoria doble puerto donde se designan las siguientes partes:

Donde WEA, ENA, SSRA, CLKA, ADDRA, DIA, DIPA son las entradas del puerto A, DOPA y DOA son las salidas del puerto A; y Donde WEB, ENB, SSRB, CLKB, ADDRB, DIB, DIPB son las entradas del puerto B, DOPB y DOB son las salidas del puerto B, las cuales ya han sido especificadas previamente.

El controlador de los bloques de memoria BRAM solo requiere instanciar el componente BRAM_16_S9_S9 el cual se encarga de direccionar todas las entradas y salidas a el bloque BRAM, por lo tanto lo único requerido es realizar el mapeo del puerto, e indicarle a la instancia cuales son las señales que van a ir conectadas internamente con las señales propias de la memoria.
La descripción de hardware del controlador de memoria se muestra en el ANEXO G.

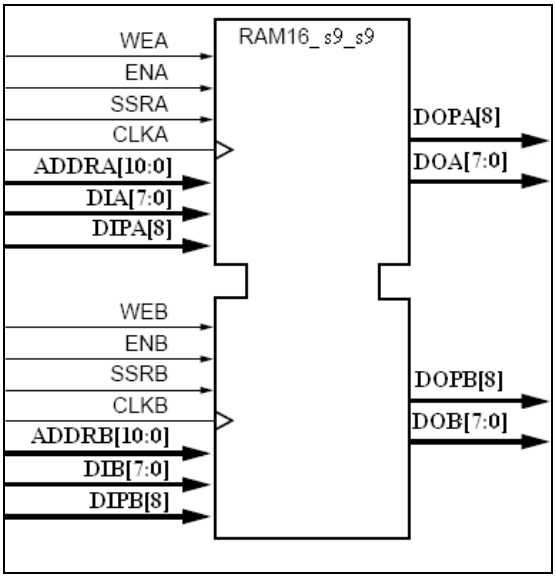


DIAGRAMA 13. Memoria RAM16_s9_s9

1.3.2.5. DESCRIPICIÒN DE ENVÌO DE DATOS AL PC COMO INTERFAZ GRÀFICA.

Se establece un proceso para transmisión de datos de la tarjeta SPARTAN 3E al PC, el fundamento del proceso esta dado por el protocolo de comunicación serial y se plantea un bloque componente para la comunicación en el Diagrama 14.

La trama de comunicación serial consiste en un pin de información que transmite uno a uno los bits de inicio, datos, paridad y parada, el bit de inicio es de nivel bajo, el ancho de los datos es de 8 bits, el bit de parada es de nivel alto.

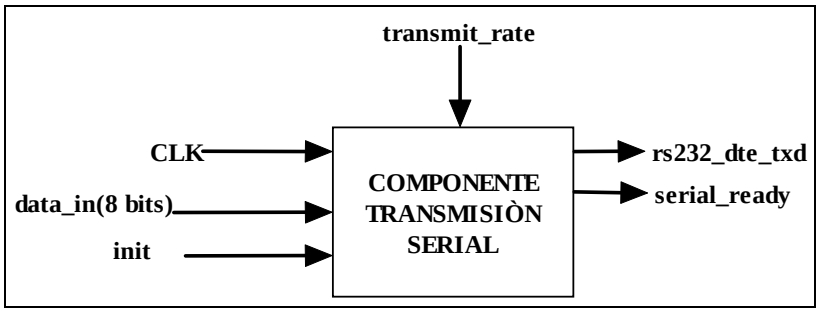


DIAGRAMA 14. Componente de Transmisión Serial.

Donde Clk, data_in, e init representan las entradas del componente y rs232_dte_txd l y serial_ready son las salidas del componente.

- Clk: Es la señal de reloj, que es directamente conectada desde el reloj externo de la tarjeta, este reloj es de 50 MHz y es representado por un solo bit. El pin de la FPGA donde esta conectado este reloj es el C9.
- *Data_in*: Es un dato de 8 bits, el cual representa el dato que va a ser transmitido, este dato se lee cada que el proceso tiene un init, por tanto si este valor sufre cambios durante el proceso no afecta la transmisión.
- *Init*: Es una señal de inicio del componente y es representada por un solo bit, cuando init esta en '1' el componente comienza a ejecutarse, y a transmitir un dato por comunicación serial.
- Rs232_dte_txd: Es la señal de comunicación serial de transmisión y es representada por un solo bit, esta conectada directamente con el pin físico del conector DB9 macho que recibe los datos por serial. El pin de la FPGA donde está conectado este es el M13
- *Serial_ready*: Es la señal que indica que la transmisión de un dato de 8 bits a culminado, es de un bit, y se pone en '1' durante 20ns, con el fin de informar al resto del sistema que una nueva transmisión puede ser realizada.

En el Diagrama 15 se muestra la estructura de control para el componente de transmisión de datos serial.

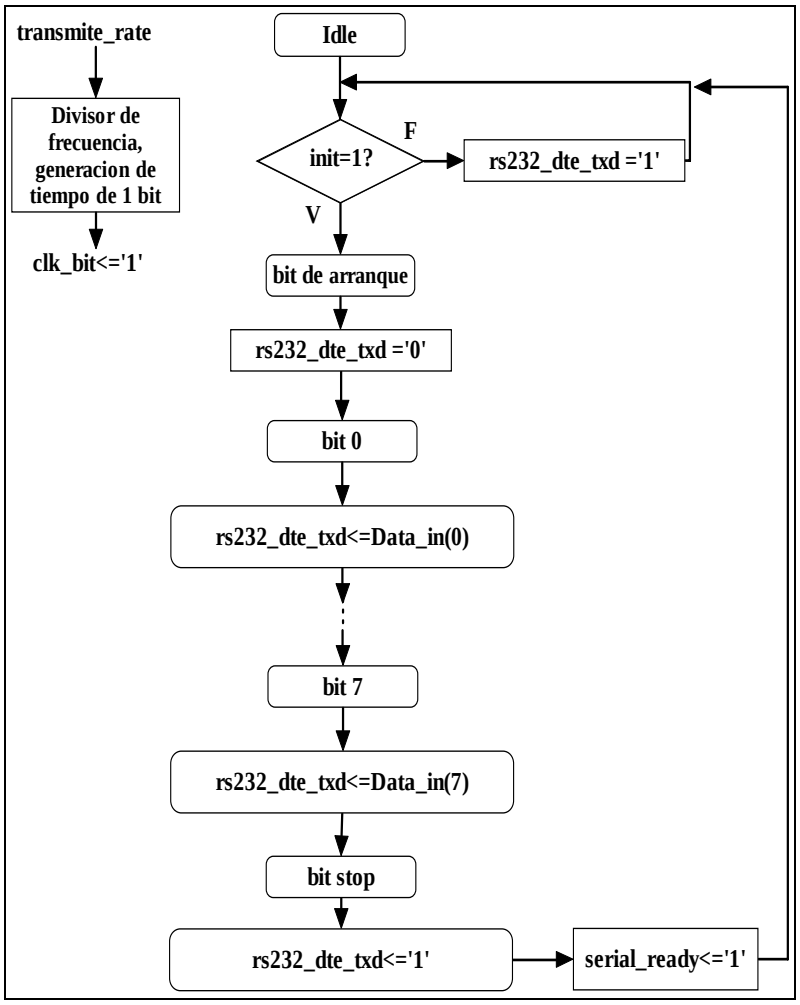


DIAGRAMA 15. Flujo para Transmisión de Datos.

El proceso de envío serial se encarga de transmitir un dato de 8 bits por comunicación serial en el conector DB9 macho, aunque solo es necesario direccionar la restricción de pines para utilizar el conector hembra que posee la tarjeta en la que viene la Spartan 3E. Dado que el dato transmitido esta en formato de 8 bits, y tiene un total de 10 bits, donde se encuentran el bit de arranque, los 8 bits de información y el bit de parada, es necesario realizar una maquina de estados que organice uno a uno los bits de información y los transmita en el orden necesario.

Este componente solo es utilizado en la parte final del procedimiento, por tanto no se requiere enviar información continuamente, con tal fin existe la señal *init*, la cual habilita o no la maquina de estados para comenzar la transmisión, la maquina de estados se ejecuta, comenzando por el estado de *idle* o espera, en dicho estado el bit que se encuentra en la línea de transmisión *rs232_dte_txd* es '1' , , en el momento que la señal de *init* sea '1', la línea de transmisión *rs232_dte_txd* se vuelve '0', dando cabida al estado de bit de arranque, esto significa que la comunicación ha comenzado, después, de un tiempo, que es determinado según la rata de transferencia, para este proyecto la rata de transferencia se determinó en 115200 baudios, lo

cual significa que el tiempo que se demora un solo bit es de 8.68us considerando que los tiempos dentro de la FPGA deben estar referenciados a el reloj principal de 50Mhz, o sea que 8.68us son 434 oscilaciones, se procede a transmitir el primer bit, que para el caso de la comunicación serial es el bit menos significativo, luego se realiza la transmisión después de transcurrido el tiempo de un bit (434), a partir de la ultima lectura, esto se hace para los 8 bits, considerando que el 8 bit es el mas significativo, en otras palabras, para cada bit existe un estado, y en ese estado se va a transmitir un bit en particular, y así direccionarlos todos correctamente.

Una vez se ha transmitido el octavo bit, transcurre un tiempo de un bit (434) y se envía nuevamente la señal *rs232_dte_rxd*, en ese momento el valor debe ser de '1', que significa que es el bit de paro, y se pasa al estado inicial de espera y se activa la señal *Serial_ready_rx* para indicar la finalización del proceso, en ese momento la maquina esta lista para realizar el mismo procedimiento.
La descripción de hardware para la comunicación serial tarjeta-PC se muestra en el ANEXO H.

Descritas las etapas en que pasa la señal del sensor de infrarrojos; ya digitalizada esta puede ser vista con la ayuda del PC como interfaz gráfica con el programa en MATLAB mostrado en el ANEXO I**.**
El programa en matlab abre el puerto COM1 de comunicación serial RS-232 y recibe los datos enviados por la tarjeta FPGA, cada dato llega con un valor de tamaño de palabra de 0 a 255 luego cada dato es convertido a su valor real concatenado por la resolución del conversor análogo/Digital del microcontrolador JK8 después los datos recibidos son graficados en la GUI que implementa matlab con la orden "plot (m)".

Se hizo una prueba de muestra con un escenario igual al de Figura 26 de la sección1.3.1.3. con una superficie regular blanca de obstáculo, colocada a una distancia de 40 cm., los pulsos provienen de la emisión de Infrarrojos que al colisionar con la superficie regresan al receptor el cual es excitado y el MCU captura el tren de pulsos, llena la memoria RAM con 100 datos y luego los envía a la tarjeta SPARTAN 3E donde también son almacenados y luego son enviados al PC para ser graficados. Los resultados de la prueba se muestran en la sección 2.1.3.

Es una prueba que se presenta como práctica para la implementación de la memoria RAM de la FPGAS, las instancias de recepción de datos y envío de datos.

1.3.3. PRUEBAS PARA CARACTERIZAR EL MEDIDOR DE DISTANCIA IMPLEMENTANDO HARDWARE RECONFIGURABLE FPGA TARJETA SPARTAN 3E DE XILINX.

Se puede afirmar que con el trabajo descrito en la sección anterior se tiene un osciloscopio o un sistema de adquisición de señal que reconstruye la señal que entrega el receptor de infrarrojos implementado con la conexión entre microcontrolador, tarjeta SPARTAN 3E y PC,
Se procede a hacer la caracterización del sensor para un solo tipo de superficie (ambiente controlado, superficie de color blanco mate).

1.3.3.1. CARACTERIZACIÓN DEL SENSOR DE INFRARROJOS.

La prueba para caracterizar el medidor de distancia con sensores Infrarrojos y tecnología de hardware programable consiste en la comunicación entre el microcontrolador, la tarjeta SPARTAN 3E y el PC, donde la función del MCU es adquirir la señal proveniente del receptor Infrarrojo y digitalizarla hasta llenar la memoria RAM de microcontrolador para luego activarse el módulo de comunicación SCI (*serial communication interface*) y enviar los datos a la SPARTAN 3E, para esta parte la tarjeta FPGA sólo se implementará para recepción, almacenaje, y envió de datos no habrá procesamiento como tal; los datos de la tarjeta FPGA serán enviados al PC con protocolo de comunicación serial, debe mencionarse que el uso del PC es implementado como interfaz gráfica solamente.

La prueba consiste en tomar lecturas de los valores de amplitud de voltaje entregado por el sensor para situaciones de presencia de superficies planas, el ambiente es controlado, la superficie plana será una regular suave de color blanco bond y un ángulo de incidencia de 0°, en esta parte de la prueba no se considera aún el concepto de fusión bisensorial, por lo que el ángulo de incidencia permanece constante y sólo se mantendrá variable la distancia de la superficie reflectora. Ver Figura 27 de la sección 1.3.1.3.

Se adquiere el valor de amplitud de la señal de 500 Hz y se modifica la distancia entre la superficie y el sensor cada 5 cm. Los resultados de la prueba de caracterización del sensor se muestran en la sección 2.1.4.

Calculado el parámetro propio del sensor en la sección 2.1.4 $\alpha 0$, teniendo el ambiente controlado (superficie reflectora de color blanco) coeficiente de reflexión de la superficie de trabajo αi asignada con un valor adimensional de 1 y teniendo los datos que entrega el conversor Análogo-Digital, solo queda aplicar la Ecuación 37.

$$X = \sqrt{\left(\frac{\alpha_0 \cdot \alpha_i}{(S-\beta)}\right)}, \rightarrow (41)$$

Ahora bien se tienen los datos para hallar la distancia, debe aplicarse en hardware reconfigurable la implementación de las operaciones necesarias para calcular el valor de la medida de la distancia en metros del sensor de infrarrojos; como son operaciones que arrojan resultados en los que se conoce el rango de trabajo se implementa en la tarjeta FPGA el uso del concepto de punto fijo.

Para medir distancia implementando hardware reconfigurable FPGAS se debe usar la ecuación que modela el sistema de medición y las operaciones implicadas en esta tarea son: división, raíz cuadrada, resta y producto, estas operaciones son implementadas con algoritmos de lógica binaria.

Teniendo los procesos anteriores confirmados como funcionales para la implementación de medición de distancias con infrarrojos se busca que la tarjeta de hardware reconfigurable SPARTAN 3E se programe con un código que reciba la señal del exterior proveniente del microcontrolador y haga las diferentes operaciones para entregar el dato de medida en metros o en su defecto centímetros.

1.3.4. MEDICIÓN DE DISTANCIA CON HARDWARE RECONFIGURABLE FPGA.

El trabajo descrito previamente entre las secciones 1.3.1 a 1.3.3. constituye la base para la medición de distancia con sensores infrarrojos implementando hardware reconfigurable.

Se tiene que el microcontrolador en la sección 1.3.2.3. se encargaba de capturar datos del conversor Análogo-Digital y llenar la ram del microcontrolador y después enviar todos los datos al exterior por medio del módulo de comunicación serial SCI a la tarjeta SPARTAN 3E.y reanudar el proceso de captura.
En esta sección dedicada a la medición de distancia se hace un leve cambio en la función del microcontrolador para que sólo envíe dos *bytes* por el módulo SCI; estos dos bytes tienen la característica especial que representan a *S* (salida del sensor con excitación) y a *β* (componente de ruido DC), con una simple modificación en el código en lenguaje ensamblador.

Teniendo los valores provenientes del microcontrolador *(S, β),* el parámetro propio del sensor que se halló en la sección de Resultados 2.1.6., el coeficiente de reflexión de la superficie constante (ambiente controlado) para un valor αi= 1; se prosigue a tomar la Ecuación 41, e implementarla por descripción de hardware VHDL para obtener el valor de medida de distancia.

Debe mencionarse que gracias a las pruebas de laboratorio el medidor estará limitado a un rango de trabajo entre 10 cm a 80cm.

Para llevar a cabo la medición de distancia con Infrarrojos se debe tener en cuenta el hardware necesario para la medición de distancia, se consideran necesarios los componentes de hardware que muestra el Diagrama 16 a continuación.

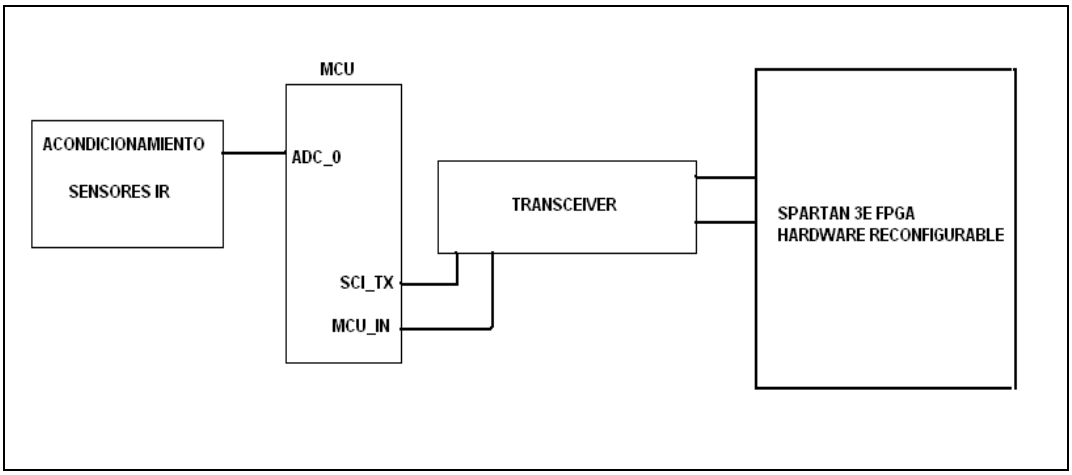


DIAGRAMA 16. Componentes de Hardware Necesarios para Medición.

Debe considerarse también los escenarios de prueba de la sección 1.3.1.3. para las mediciones con la implementación de la Ecuación 41.

En la descripción de hardware para implementar la Ecuación 41, se tiene que se necesitan componentes utilizados anteriormente como la recepción de datos, transmisión de datos, emisión de señal y además componentes que faciliten la ejecución de la Ecuación, los cuales no han sido vistos con anterioridad. El Diagrama 17, muestra los componentes presentes en el bloque marcado en el Diagrama 16 como "SPARTAN 3E FPGA HARDWARE RECONFIGURABLE".

El cual muestra la implementación de los componentes de emisión de envolvente, recepción de datos y transmisión de datos vistos anteriormente, además la aplicación de componentes como demultiplexor DEMUX, COMPARADOR-RESTADOR, el IP CORE para división de números en binario.

El Diagrama 17 muestra el sistema para medición de distancia con sensores infrarrojos implementando hardware reconfigurable, su funcionamiento se da inicio con un interruptor de la tarjeta direccionado para iniciar el proceso, se activa el componente de recepción de datos serial el cual posee una bandera de salida que se activa cuando un byte ha sido recibido, con esta bandera activa se habilita el componente DEMUX 1 y se direcciona hacia el REGISTRO1, como se sabe que el microcontrolador envía un segundo byte el componente de recepción continua en la adquisición y cuando la bandera de este componente se activa de nuevo el

componente DEMUX 1 se habilita y se direcciona hacia el REGISTRO 2, una vez el REGISTRO 2 es ocupado el componente de recepción serial de datos se desactiva, y el componente COMPARADOR-RESTADOR se habilita para hacer una comparación entre los dos registros y hallar el dato cargado con menor valor para restarlo con el otro valor cargado en el registro que se supone es mayor. Una vez obtenido el valor de esta operación se activa el archivo cargado como componente DIVISION IP CORE el cual es un archivo sacado de una de la herramientas del ambiente de desarrollo ISE de XILINX el cual se refiere a algoritmos de diferentes aplicaciones denominados IP's los cuales son algoritmos disponibles como paquetes de propiedad intelectual donde se implementan pero no se tiene acceso al código como tal.

Los elementos necesarios para hacer la división son el dividendo y el divisor, hasta ahora se tiene el divisor que es el dato entregado por el componente COMPARADOR_RESTADOR llamado RESULTADO RESTA y el divisor es un valor constante que se determinó en la sección de Resultados 2.1.6. como parámetro propio del sensor y el coeficiente de reflexión de la superficie como ambiente controlado (superficie blanca $\alpha i=1$).
El IP de división posee una bandera que se activa cuando una operación de división ha sido exitosa, cuando esta bandera se activa se envía el resultado al PC por medio del componente de transmisión serial y se reanuda el proceso.

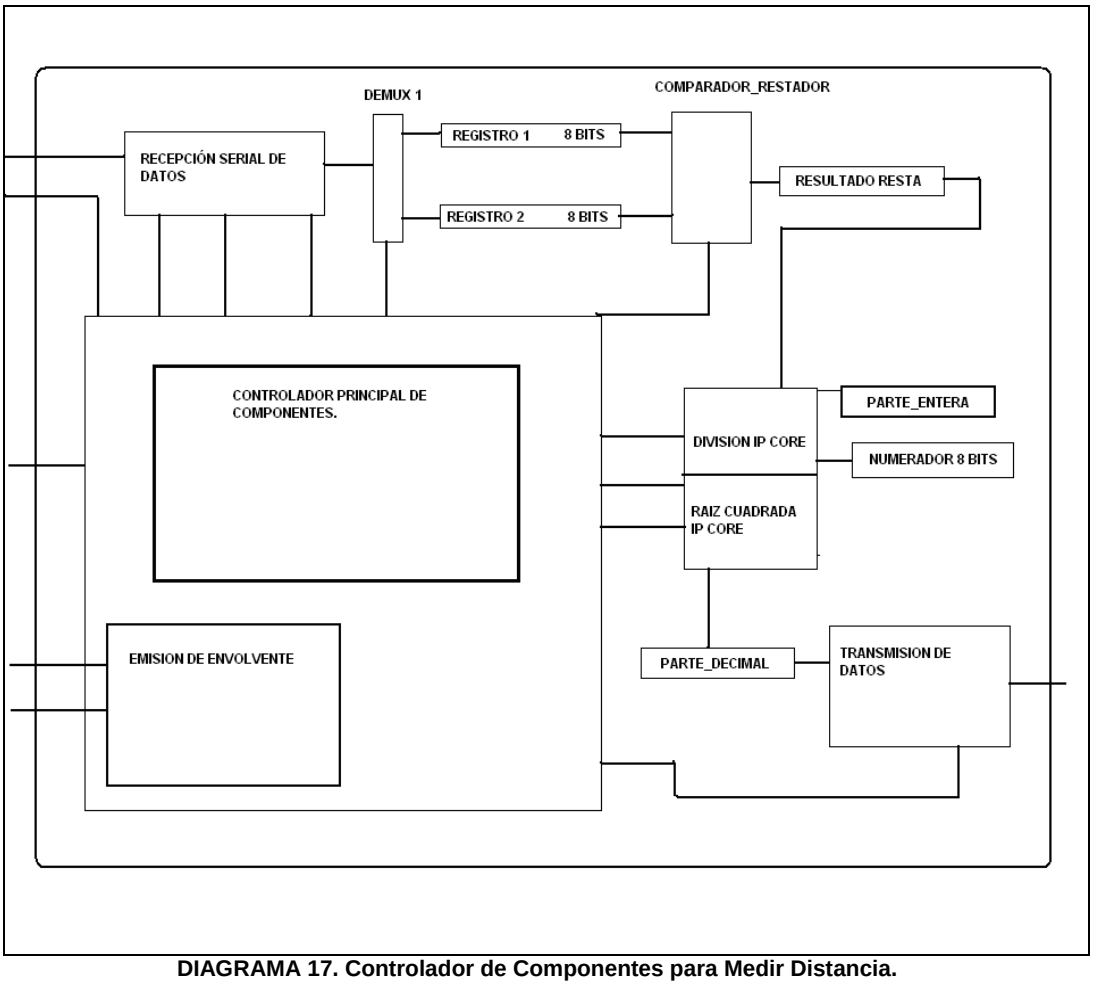


DIAGRAMA 17. Controlador de Componentes para Medir Distancia.

El proceso por descripción de hardware para medir distancia con sensores infrarrojos debe pasar una prueba que es la cuantificación del error en una medición, para tal propósito se propone una serie de pruebas para calcular y cuantificar el error presente en el sistema implementado.

Los escenarios de prueba de la sección 1.3.1.3. se perfilan adecuados para el procedimiento.

Los resultados de estas pruebas se muestran en la sección de Resultados 2.1.7

1.3.4. DISEÑO Y CONSTRUCCIÓN DEL BLOQUE DE PERCEPCIÓN SENSORIAL CON SENSORES INFRARROJOS.

Confirmados el modelo de medición de distancias con sensores infrarrojos, el circuito con la implementación de hardware reconfigurable para medición y la implementación de la técnica de fusión bisensorial, el siguiente paso es la asociación de las actividades anteriores a la plataforma móvil.

1.3.4.1. ESTRUCTURA DE SOPORTE PARA SENSORES INFRARROJOS.

El trabajo propuesto por Blanes [18] para posicionar los sensores Infrarrojos e implementar el modelo de medición de distancias y su vez la técnica de fusión bisensorial vistos en los preliminares P.4.6.3.2. y P.4.6.3.3. es usar una estructura física de soporte para los sensores Infrarrojos donde se pueda posicionar el par de sensores para implementar la fusión bisensorial y la capacidad para capturar datos del entorno perimetral, esta estructura se propone como un hexágono mostrado en la Figura 32.

Teniendo en cuenta el anillo hexagonal propuesto por Blanes [18] en su trabajo debe pensarse que este tipo de anillo debe estar diseñado con respecto al am18iente de trabajo y como se mencionó anteriormente, el proyecto esta delimitado para ambientes controlados, la implementación de este tipo de estructura se adecua para la aplicación mientras el diseño de la estructura hexagonal este escalada de acuerdo al ambiente controlado de trabajo.

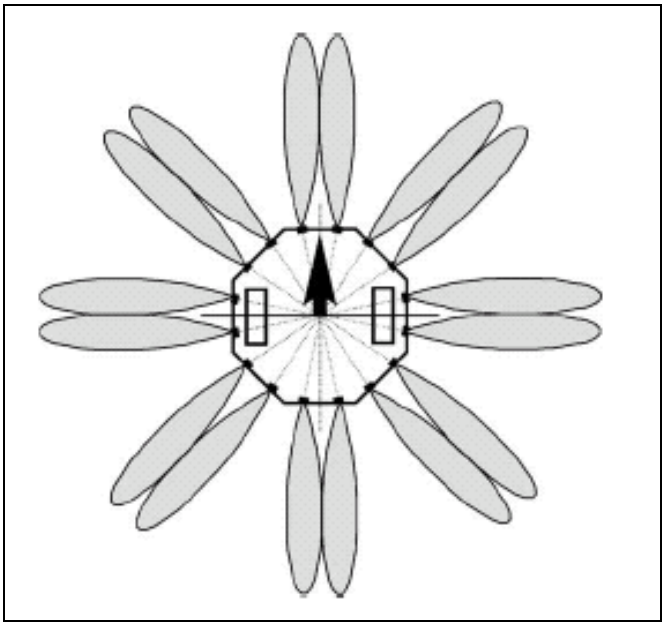


FIGURA 32. Anillo Hexagonal Inicial[15]

[15] Figura 32 (Robotics and Autonomous Systems[15] 40 (2002) 255–266 Using infrared sensors for distance measurement in mobile robots G. Benet., F. Blanes, J.E. Simó, P. Pérez *Departamento de Informática de Sistemas, Computadores y Automática, Universidad Politécnica de Valencia, P.O. Box*

Para el proyecto en cuanto optimización sensorial se optó por implementar la misma estructura hexagonal pero solo usar dos tercios de la misma en la plataforma móvil, pues se piensa que para construir mapas de entorno dinámicos la información del bloque sensorial adquiriendo dos tercios del entorno es mas que suficiente.

Se diseña un plano hexagonal con 5 lados activos (donde va cada par de sensores) con 13 centímetros de longitud, se posicionan los sensores a 10 centímetros de distancia del mismo lado, como se muestra en la Figura 33. Por lo que se tiene que para el proyecto se implementan 10 sensores medidores de distancia con infrarrojos. La estructura se cortó en material de acrílico y se muestra en la sección de resultados 2.1.8.

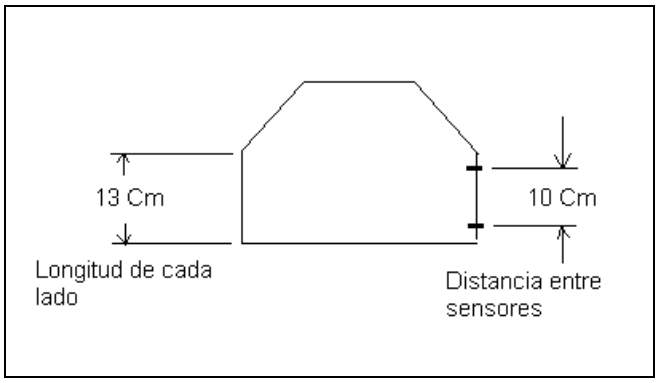


FIGURA 33. Diseño Soporte para Sensores Infrarrojos.

1.3.4.2. ELECTRÓNICA Y ACONDICIONAMIENTO DE LOS SENSORES INFRARROJOS AL SOPORTE FÍSICO EN ANILLO.

Como se mencionó en la sección anterior el soporte físico se encarga de distribuir los 10 sensores en la periferia del sistema. Se diseñó un circuito que abarcara la emisión, recepción de infrarrojos, acondicionamiento de señal, captura y envió de datos a la tarjeta FPGA para un par de sensores, puesto que se pensó en dejar en un solo esquema la circuitería para lo que es la medición de distancia y la técnica de fusión bisensorial. En el Diagrama 18 se muestra los componentes necesarios para el Esquemático 10.

22012, 46080 Valencia, Spain Received 9 August 2001; received in revised form 27 March 2002
Communicated by F.C.A. Groen **)**

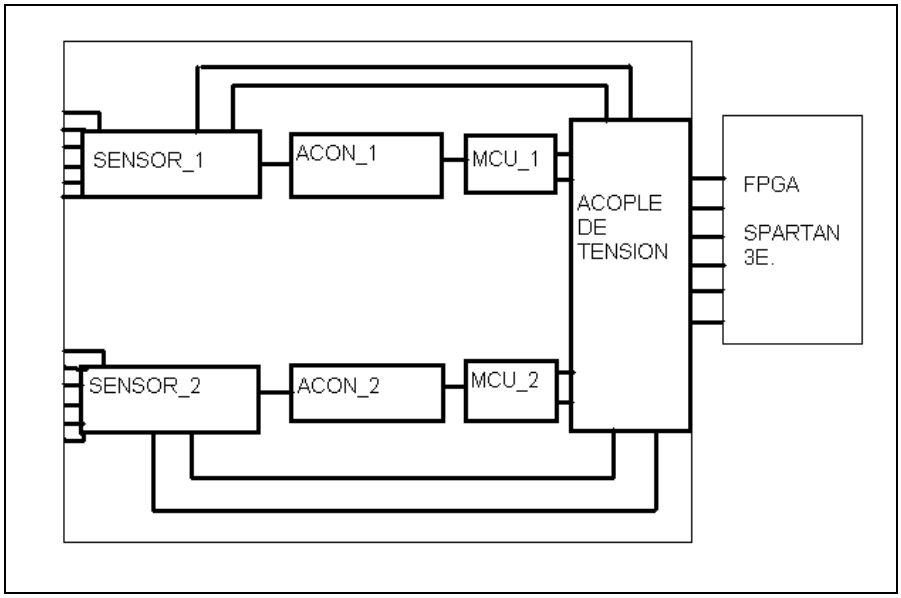
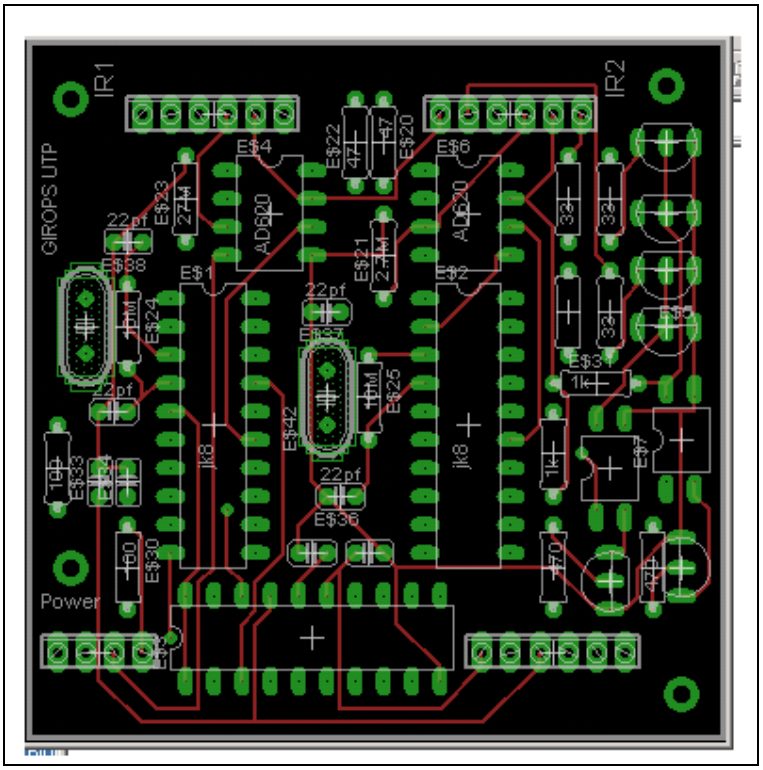


DIAGRAMA 18. Componentes para el Circuito de Arreglo Bisensorial y Medición de Distancia.



ESQUEMÁTICO 10. Circuito Impreso de Emisión, Recepción de Infrarrojos, Acondicionamiento y Captura de Señal y Envío a la Tarjeta FPGA.

Ahora bien, son 5 los circuitos impresos con el Esquemático 9. para así obtener un arreglo de 10 sensores; antes de acoplarlos a sistema denominado bloque de percepción sensorial; los impresos deben estar alimentados a +5 , -5 y 3.3 Voltios, esta parte se tiene que tomar con mucha atención pues el fin de colocar el bloque de percepción sensorial en la plataforma móvil es que ésta se auto- alimente, para tal fin se usa una batería de 12 Voltios y un conversor DC-DC con aislamiento el cual entrega con una entrada de 12 Voltios una fuente dual simétrica de +5 y -5 Voltios con tierra aislada, que es una forma de proteger los impresos y la tarjeta SPARTAN 3E.
El bloque de percepción sensorial se muestra en la sección de Resultados 2.1.9.

1.3.5. FUSIÓN ANILLO DE SENSORES INFRARROJOS, MOTORES Y ENCODERS PARA LA NAVEGACIÓN Y CONSTRUCCIÓN DE MAPAS DE ENTORNO.

Una vez obtenidos los diferentes componentes necesarios para la navegación se procede a fusionarlos para constituir la Plataforma móvil autónoma como se ve en las Foto 10,11,12,13; para tal objetivo debe diseñarse un código en vhdl que describa el funcionamiento de la siguiente forma:
Captura de datos provenientes de los sensores infrarrojos, estos datos son comparados con un parámetro de existencia de obstáculo, si es mayor debe ejecutarse una acción de evasión hacia los motores (giro de 90 grados) si es menor continua en la dirección predeterminada por el código(línea recta hacia adelante) respectivamente con cada sensor, a su vez los datos de captura procesados por el hardware reconfigurable son enviados al PC para la construcción del mapa de entorno.

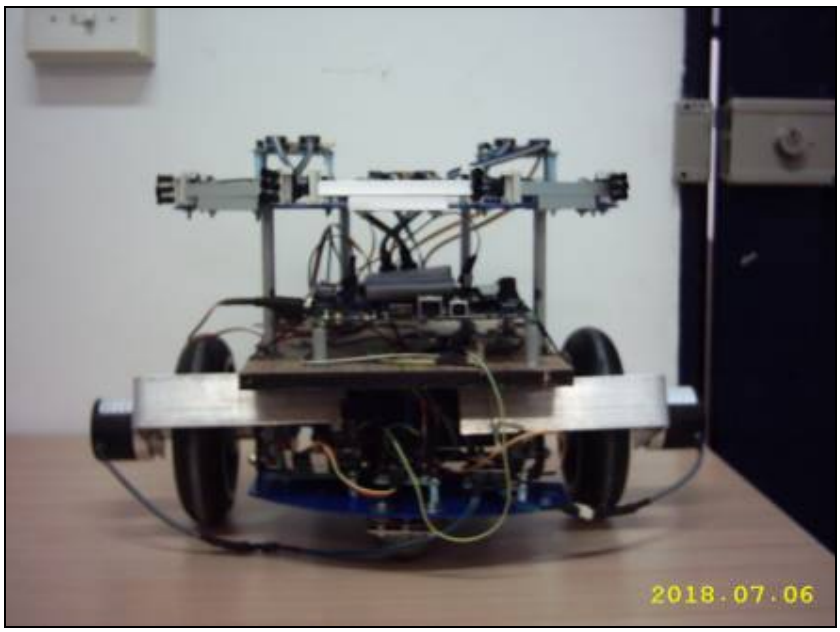


FOTO 10. FRONTAL PLATAFORMA MÓVIL COMPLETA.

FOTO 11. POSTERIOR PLATAFORMA MOVIL COMPLETA

FOTO 12. PLATAFORMA MÓVIL COMPLETA 1.

FOTO 13. PLATAFORMA MÓVIL COMPLETA 2.

El hardware necesario para la navegación de la plataforma se muestra en el Diagrama 19.
Debe mencionarse que en el Diagrama 19 se muestra la estructura de navegación para el control con un solo sensor, pero la transmisión de datos esta configurada para todos los sensores, pues se tiene este diseño de código como prueba base de navegación para verificar el funcionamiento de la plataforma

Los resultados de la fusión física de los componentes y la comprobación del código diseñado se muestran en la sección de Resultados 2.2, donde se anexan videos de pruebas y mapas de entorno de diferentes situaciones.

La plataforma móvil acoplada con todos los componentes propuestos en el desarrollo del proyecto tiene por misión explorar el entorno y construir un mapa del mismo.
El mapa en si mismo no es útil para el robot en tanto su misión es construirlo y no utilizarlo, pero dicho mapa va a ser usado en otros sistemas.

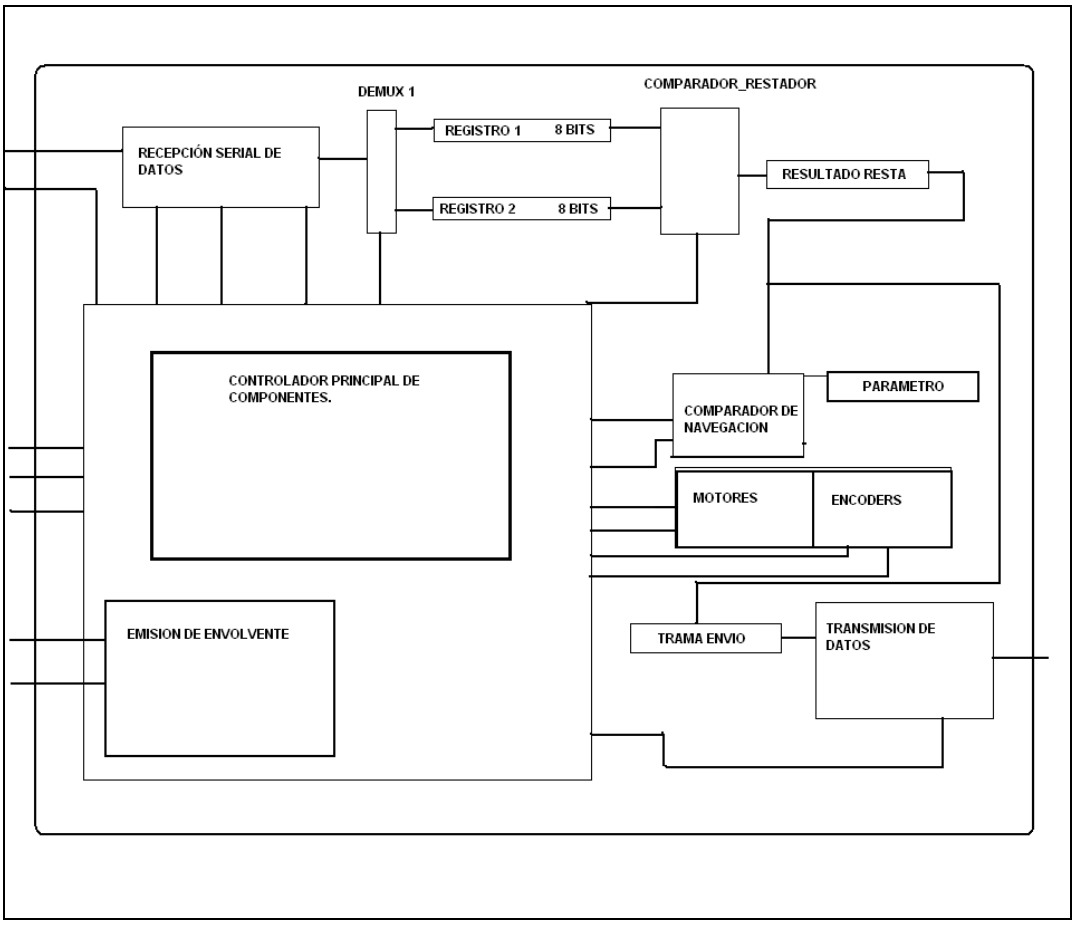


DIAGRAMA 19. NAVEGACIÓN, EVASIÓN Y MAPEO

2. RESULTADOS.

Este capítulo se presenta con una partición de etapas donde se muestra que para poder llegar a los resultados concluyentes se debió primero recorrer un camino que requería resultados de diferentes partes y conceptos en robótica móvil sobre navegación, percepción sensorial y localización, los cuales una vez obtenidos y verificados se unen para la ejecución de pruebas de navegación y construcción de mapas de entorno para conformar lo que se denominó como Resultado Final.

2.1. RESULTADOS INICIALES.

En esta sección se presentan los resultados de los montajes necesarios para poder tener a punto de funcionamiento la plataforma móvil para luego someterla a las pruebas finales que se verán en la sección 2.2, puesto que para asegurar el punto en que se somete la plataforma a pruebas, ésta debe tener un arreglo de varios dispositivos que son fusionados en el sistema para poder llevar a cabo las mismas de forma exitosa.

2.1.1. RESULTADOS DE LA APLICACIÓN Y FUSIÓN DEL CONTROL DE MOTORES Y *ENCODERS*

En la construcción de la plataforma móvil como se observa en el Capítulo 1 de Desarrollo del Proyecto se implementaron diferentes dispositivos independientemente como son el control de motores y los *encoders,* dichos dispositivos fueron mutuamente acoplados con una estructura robusta que garantiza el funcionamiento y el objetivo de implementar una plataforma móvil.

Se realizaron pruebas de odometría implementando un control con la tarjeta FPGA y acoplando cada módulo independiente de la plataforma, esto se puede observar en los anexos de videos adjuntos al CD del documento. Ver videos 1, 2, 3, 4, 5,6.

2.1.1.1. RESULTADOS PRUEBA DE DISTANCIA LINEAL RECORRIDA.

La primer prueba de la aplicación, fue realizar el recorrido de 1 metro en línea recta para calcular el error en la medida y la desviación en grados con respecto a la línea recta trazada para verificar que el código en vhdl del control esté acertado, esto se puede observar en la Foto 13 y en la Tabla 5 se observan los datos que arrojó la prueba.

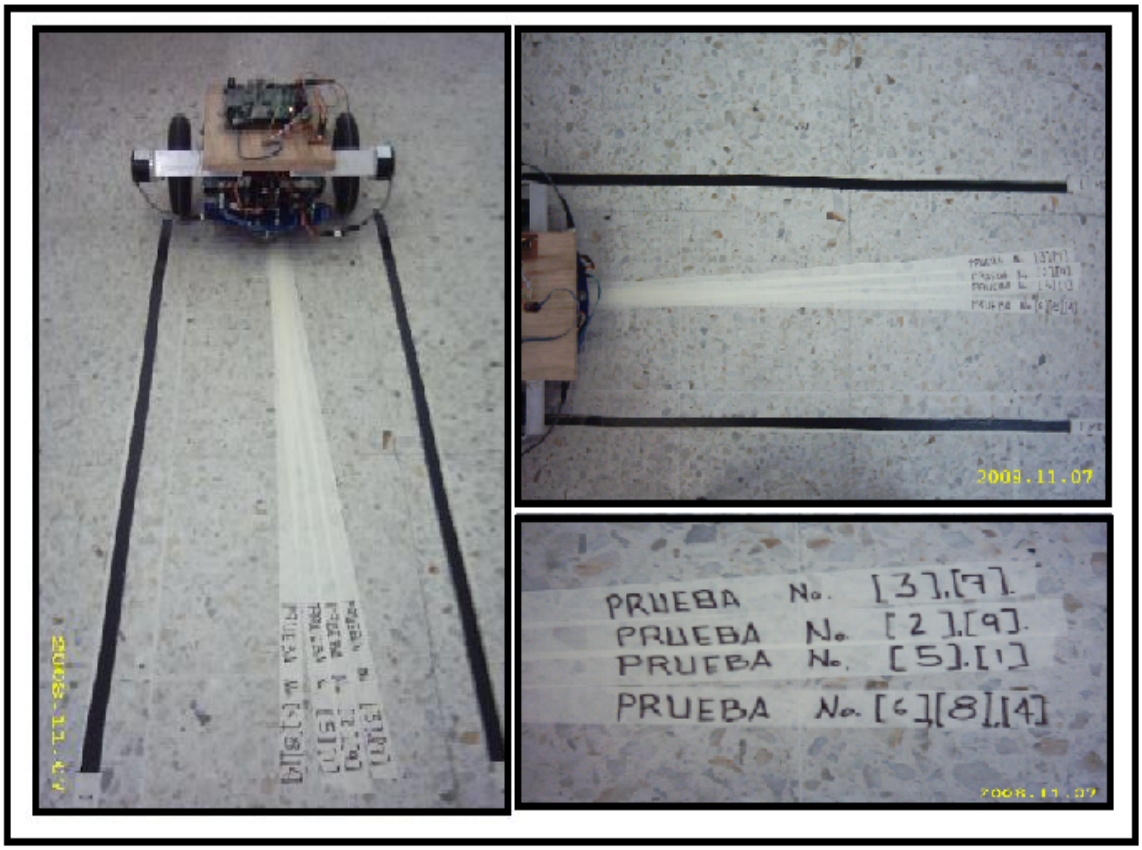


FOTO 13. Desplazamiento en Escenario de Prueba de Distancia Recorrida

No. PRUEBA	DIST. RECORRIDA (cm)	ÁNGULO DESVIACIÓN (Grados)
1	97	5
2	98,2	8
3	100	10
4	98,5	0
5	99,4	5
6	99,4	0
7	98,2	10
8	100	0
9	100	8
10	99,3	8
PROMEDIO	**99**	**5,4**

TABLA 5. Datos de Prueba de Distancia Recorrida

En la Tabla 5 se toma que la distancia recorrida por la plataforma móvil es aproximada a los 100 cm. con un error del 1%, se considera un margen de error óptimo para la aplicación en el proyecto, en cuanto al ángulo de desviación en la longitud recorrida se tuvo un máximo de desviación de 10 grados y un mínimo de 5 grados, y en todas las pruebas esta desviación se dio hacia el lado izquierdo.

De lo anterior puede afirmarse que el código implementado para la aplicación de recorrido de distancia lineal es viable para su uso en el proyecto pues se comprobó que su margen de error es relativamente bajo.

2.1.1.2. RESULTADOS PRUEBAS DE GIRO RECTO.

La segunda prueba consiste en programar en la tarjeta FPGA un código cuyo fin sea poner a la plataforma móvil recorrer una trayectoria cuadrada y verificar si el giro es de 90 grados (ángulo recto).

Se tomaron datos de longitud y ángulo de desviación con respecto a una trayectoria cuadrada demarcada previamente, los datos tomados corresponden al desplazamiento en línea recta y al ángulo de desviación de la plataforma con respecto al cuadrado interior visto en la Foto 14.

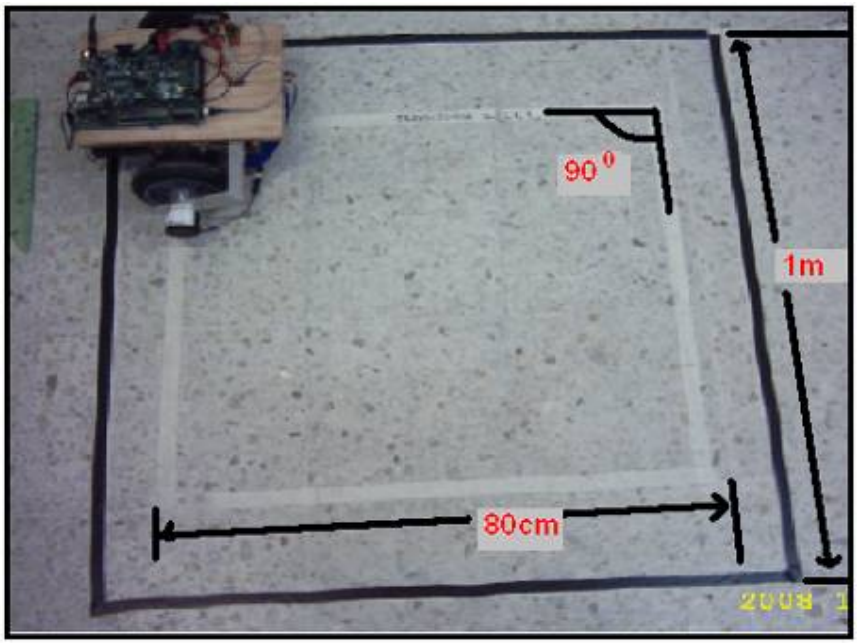


FOTO 14. Desplazamiento en Escenario de Prueba de Giro.

Se determinó que para una distancia de 80 cm programada en la tarjeta FPGA el valor promedio para 10 pruebas de desplazamiento fue 79,75 cm con un error 0.996875 esto indica que para esta prueba el desplazamiento de la plataforma es confiable en un 99.68%, como se puede ver en la Tabla 6.

En la prueba sobre el recorrido en línea recta se observa que el ángulo de desviación promedio es 5,5 grados y teniendo en cuenta que el posicionamiento de la plataforma móvil en la línea de partida no existe un alineamiento completo, debe considerarse también la existencia de una fuente de error, de error físico. Ver Foto 15.

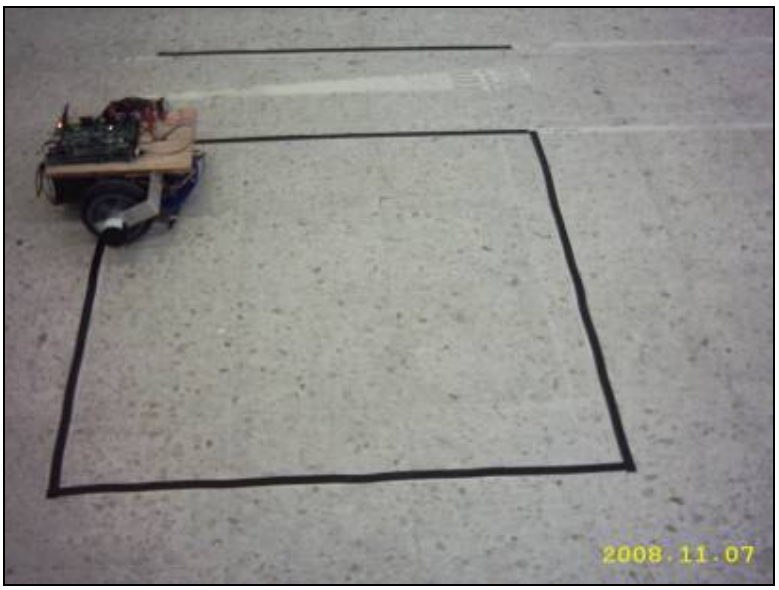


FOTO 15. Escenario de Prueba para Giro.

En la Foto 16 se muestra la plataforma recorriendo la trayectoria programada por decima y vez y se denotan las trayectorias previas marcadas con cinta blanca las cuales se aprecian mejor en la Foto 17.

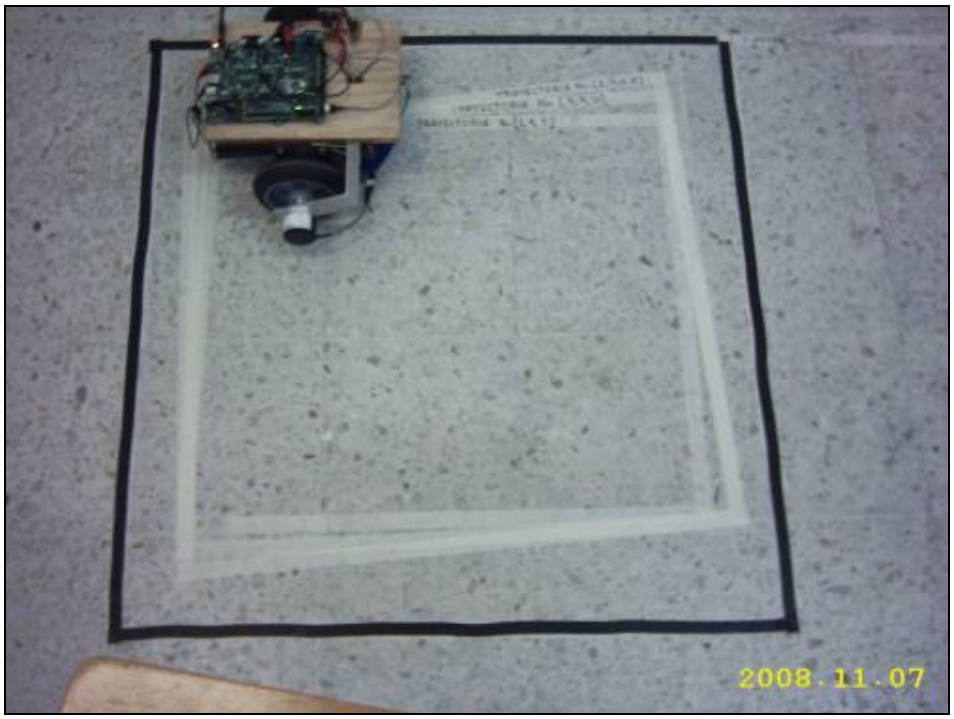


FOTO 16. Décima Trayectoria de Prueba.

FOTO 17. Trayectorias Recorridas para la Prueba de Giro.

No.PRU.	DIST. RECORRIDA (cm)	ÁNG. DESVIACIÓN (Grados)	ÁNG. DE GIRO (Grados)
1	80	0	89
2	78,8	10	90
3	80,1	5	89
4	82	0	91
5	77,3	10	90
6	79,3	10	88
7	80	0	90
8	80	10	89
9	80	5	90
10	80	5	90
PROM.	**79,75**	**5,5**	89,73333333

TABLA 6. Datos de Prueba para Distancia y Giro.

De la Tabla 6 se toma que la distancia recorrida por la plataforma móvil es aproximada a los 80 cm. con un error del 1%, se considera un margen de error óptimo para la aplicación en el proyecto, en cuanto al ángulo de desviación en la longitud recorrida se tuvo un máximo de desviación de 10 grados y un mínimo de 5 grados, y en todas las pruebas esta desviación se dió para el lado izquierdo al igual que en la prueba de distancia anterior, para el giro de 90 grados las pruebas arrojaron que existe mas resolución en la situación de giro que el recorrido de distancia.

De lo anterior puede afirmarse que el código implementado para la aplicación de recorrido de distancia lineal y giro es viable para su uso en el proyecto pues se comprobó que su margen de error es relativamente bajo.

2.1.2. RESULTADOS DE LA MEDICIÓN DEL RUMBO MAGNÉTICO CON BRÚJULA DIGITAL E IMPLEMETACIÓN DE CONTROL EN *SPARTAN* 3

Para la aplicación de la brújula digital se obtuvieron mediciones del rumbo magnético de la tierra basándose en una línea de referencia virtual que por especificaciones del fabricante se encuentra en uno de los lados de la brújula ya descrito en el desarrollo del proyecto garantizando así la confiabilidad de la brújula y la medición del rumbo magnético; como se puede observar en la Foto 14 la brújula tradicional entrega una lectura que coincide con el dato que entrega la tarjeta FPGA.

Se implementó la medición por ancho de pulso (PWM) en la tarjeta FPGA de la señal proveniente de la brújula digital, lo que se logra es medir el ancho de pulso de la señal mediante un proceso de captura y conteo del ancho de pulso.

En la Foto 18 se logra observar que la brújula magnética tradicional se ubica en un plano horizontal con dirección (NORTE (0) grados) , en esta misma línea se ubica la brújula digital y con el control respectivo se mide el ancho de pulso de la señal análoga entregada por brújula digital que es procesada por la FPGA, el error aproximado del dato entregado por la brújula es de más o menos 5 grados y 10 grados de tara al comienzo de la señal PWM entregada los 10 grados de tara en la señal son restados al valor real del dato entregado por la FPGA .

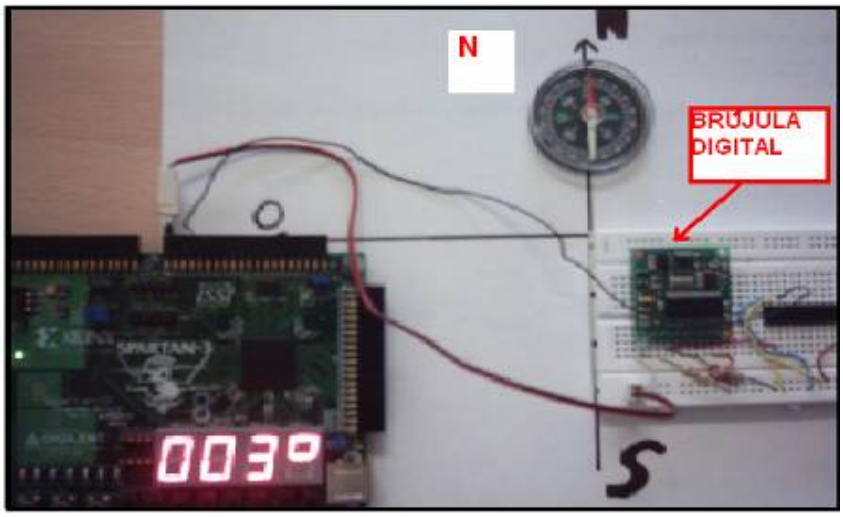


FOTO 18. Adquisición FPGA con Brújula Digital

De lo anterior puede afirmarse que el código implementado para la aplicación con la brújula digital es viable para su uso en el proyecto para la construcción de mapas de entorno dinámicos.

2.1.3. RESULTADOS PRUEBA MEDICIÓN DE DISTANCIA CON ÁNGULO DE INCIDENCIA DE 0 GRADOS.

Las curvas obtenidas en las pruebas de laboratorio mostradas en las Gráfica 7 confirman el modelo de medición de distancias con Sensores Infrarrojos de YAIR cuando se conoce la superficie reflectora es decir el coeficiente de reflexión de la superficie.

Las curvas conservan el mismo comportamiento para diferentes superficies pero las amplitudes de voltaje entregadas por el medidor de distancia varían para la misma situación (misma distancia) en el orden de milivoltios. Retomando el modelamiento del sensor de infrarrojos como medidor de distancia en la Ecuación 18.

$$S(x,\theta) = \left(\alpha * Cos\theta / x^2\right) + \beta, \rightarrow (18)$$

Y el ángulo de incidencia de 0 grados la función Cos θ es igual 1 por lo que la ecuación se modifica a la Ecuación 36:

$$S(x,\theta) = \left(\alpha / x^2\right) + \beta, \rightarrow (42)$$

Como α esta constituido por un parámetro propio del sensor y el coeficiente de reflexión de la superficie reflectora (Ecuación 19), se toma como conocido el coeficiente de reflexión αi

$$\alpha = \alpha 0 \times \alpha i, \rightarrow (19)$$

En la prueba se conoce la distancia, la superficie (para el coeficiente de reflexión de la misma) y la componente de voltaje DC presente llamada β, y el voltaje entregado por el sensor para cada situación de distancia y superficie reflectora se construye una curva que confirma el modelamiento de YAIR para medir distancias, basta con solo comparar la Grafica 7 con la Grafica 1 de la sección de Preliminares P.4.6. y se aprecia un comportamiento de las curvas muy similar que conlleva a poder afirmar que el modelamiento del sensor es acertado.

Debe mencionarse que los resultados de la prueba tenían como constante primordial el ángulo de incidencia fijo a 0 grados, pero debe tenerse en cuenta que para la aplicación en el proyecto el sistema no puede asumir que todas las superficies estarán posicionadas al mismo ángulo de incidencia de 0 grados por lo que se hace necesario obtener los resultados de la prueba siguiente que tiene que ver con la variación del ángulo de incidencia para argumentar el uso de la técnica de fusión bisensorial vista en la sección de Preliminares P.4.6.3.3.

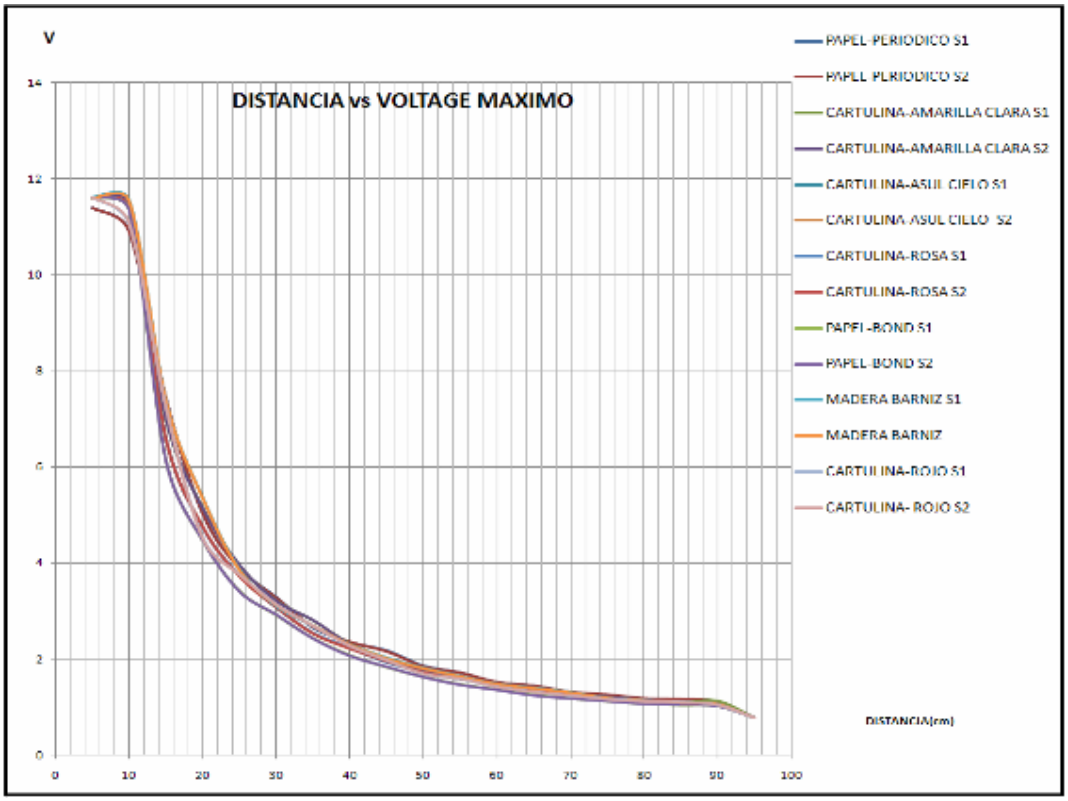


GRÁFICO 10. COMPORTAMIENTO DEL SENSOR MEDIDOR DE DISTANCIAS PARA DIFERENTES SUPERFICIES Y DISTANCIAS

2.1.4. RESULTADOS PRUEBA CON VARIACIÓN EN ÁNGULO DE INCIDENCIA Y VARIACIÓN DE DISTANCIA

Debido a que en los resultados de la prueba anterior el ángulo de incidencia era constante el escenario de la prueba denota que para la aplicación el sistema recibirá información errada puesto que el ambiente es controlado en cuanto a la naturaleza de la superficie mas no de su ubicación en cuanto a la dirección de los sensores, es decir, que no hay garantías de que el ángulo formado por la superficie reflectora (ambiente de trabajo) sea de 0 grados en todo momento. El comportamiento del sensor en la prueba se ve en el Grafico 8, en éste se observa que el voltaje entregado por el medidor de distancia decae cuando el ángulo de incidencia es diferente de 0 grados, y también la magnitud del voltaje decae en mayor escala cuando la superficie esta mas cercana al sensor (menor distancia), pues el área de ocupación de la señal emitida en la superficie es mayor, pero los rayos de reflexión no quedan direccionados al receptor por lo que el mismo no es excitado con la misma intensidad que si estuviera a 0 grados con la superficie. Lo cual induce a la importancia del uso de la técnica de fusión bisensorial vista en la sección de Preliminares 4.6.3.3.

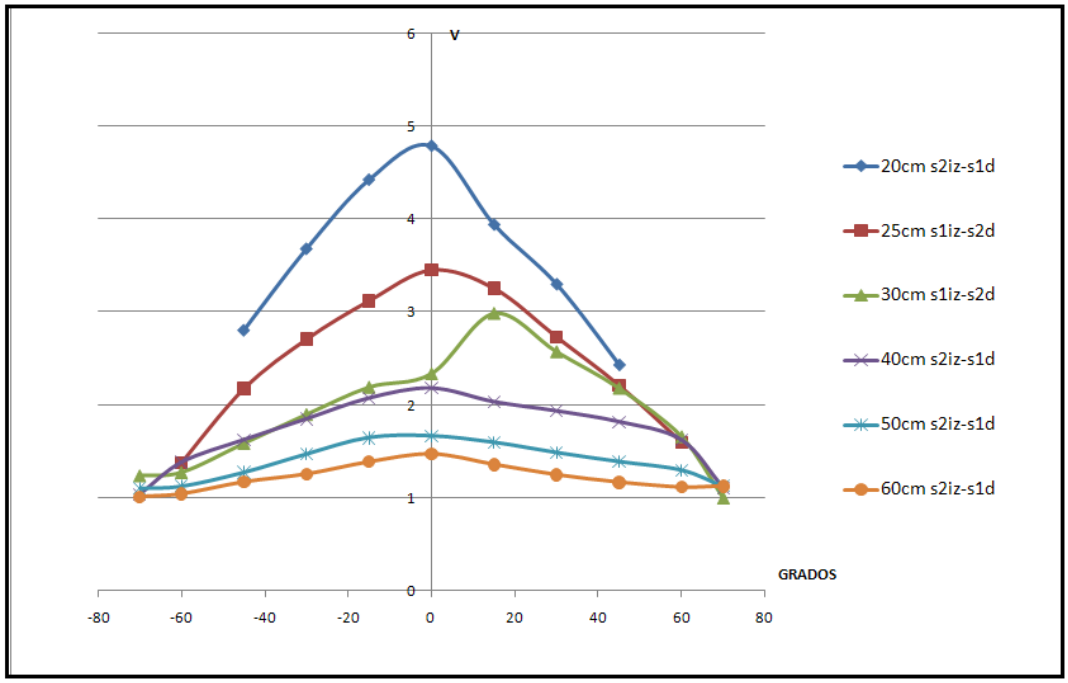


GRÁFICO 11. Respuesta del Sensor a la Variación del Ángulo de Incidencia.

Se hace necesario usar el concepto de fusión sensorial donde primero se halla el ángulo de incidencia y luego las distancias de cada uno de los sensores involucrados en la fusión.

2.1.5. RESULTADOS ADQUISICIÒN DE SEÑAL, CONVERSIÓN ANALOGO A DIGITAL E INTERFAZ GRÀFICA CON PC.

El gráfico 10 muestra la señal adquirida por el MCU enviada por el conjunto de amplificador de instrumentación y receptor Infrarrojo a la tarjeta FPGA y luego al PC.

Teniendo la opción activa de establecer comunicación entre la tarjeta SPARTAN 3E y el PC como herramienta de interfaz grafica diseñada, y confirmar que los datos capturados por el conversor análogo/ digital del microcontrolador pasan por el bloque de memoria RAM de la FPGA y luego viajan al PC para ser graficados y comparados con la señal vista desde el osciloscopio se afirma que el proceso para la captura de la señal que entrega el sensor de infrarrojos es exitoso pues la señal alcanza a ser reconstruida de forma digital en el PC. Ver Grafico 10.

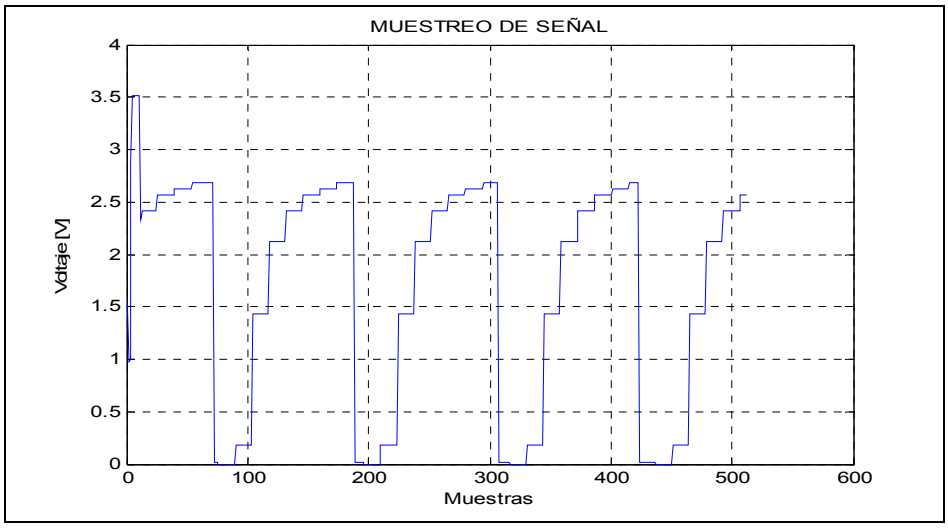


GRÁFICO 12. Señal Adquirida MCU, SPARTAN 3E, PC.

El resultado de esta prueba que se presenta como práctica para la implementación de la memoria RAM de la FPGAS, las instancias de recepción de datos y envío de datos, abre paso para la caracterización del sensor de infrarrojo para medir distancia con el hardware reconfigurable.

2.1.6. RESULTADOS PRUEBAS PARA CARACTERIZAR EL MEDIDOR DE DISTANCIA IMPLEMENTANDO HARDWARE RECONFIGURABLE TARJETA SPARTAN 3E DE XILINX.

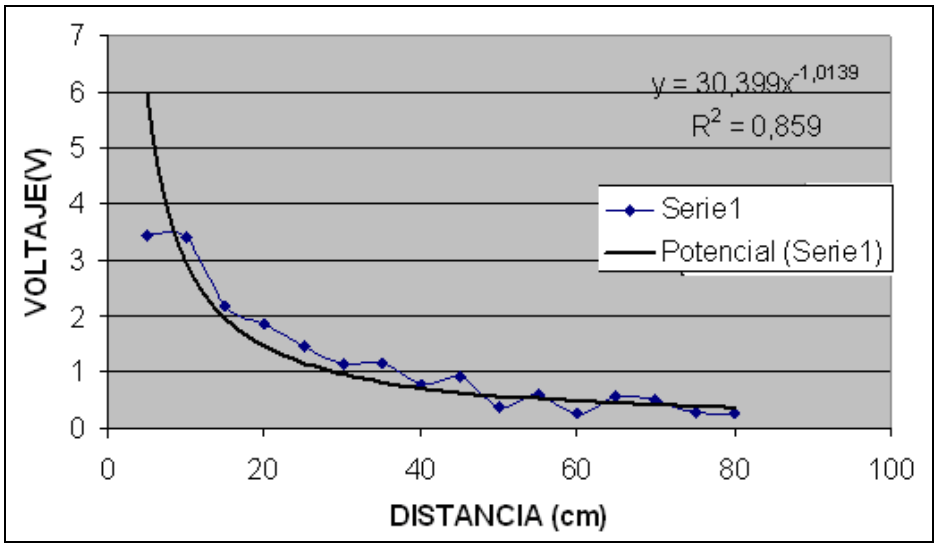
La prueba hecha es la misma a la efectuada en la sección 1.3.1.3. con el cambio de la implementación del microcontrolador y hardware reconfigurable.

Con una regresión de datos se genera una ecuación que modela el comportamiento del sensor de Infrarrojos, difiere del modelo basado en la ley de la fotometría debido a que la prueba empírica refiere dos suposiciones que físicamente alejan el modelo conocido con el modelo hallado en la prueba. Ver gráfico 11.
El comportamiento de la curva sigue una tendencia potencial la cual se adecua a la ley de los inversos cuadrados de la fotometría:

$$S(x,\theta) = \left(\alpha * Cos\theta / x^2\right) + \beta, \rightarrow (18)$$

Durante la prueba se tomaron ciertos factores en cuenta:

- Θ = 0 debido al ángulo de incidencia de 0° el factor cos θ es igual a 1.
- β siendo componente DC.
- αi = 1 coeficiente de reflexión de la superficie.



GRÀFICO 13. Respuesta del Sensor Infrarrojo contra Distancia.

Se afirma que la amplitud de la señal de 500 Hz. es el valor de interés por lo tanto el valor máximo de tensión entregado menos el valor mínimo de tensión entregado compone lo que se ha llamado amplitud de señal, que es la salida del sensor S(x, θ).

Por lo que la ecuación queda:

$$S(x,\theta) = \left(\alpha / x^2\right) + \beta, \rightarrow (41)$$

Donde la prueba toma importancia en cuanto al modelamiento del sensor es para hallar el valor de α que consta de: $\alpha = \alpha 0$ x αi. Donde αi corresponde a la naturaleza de la superficie como coeficiente de reflexión que para el caso de la prueba se considera con valor de 1, queda el valor propio del comportamiento del sensor como $\alpha 0$, este valor es constante pues tiene que ver con la naturaleza del sensor como parámetro. Ver Tabla 7.

DISTANCIA(cm)	Max(V)	Min(V)	AMPLITUD	DC	voltaje de sal
5	4,25	0,8	3,45	4,24	4,6432
10	4,176	0,76	3,416	4	4,1008
15	2,88	0,72	2,16	3,04	3,0848
20	2,5	0,64	1,86	0,88	0,9052
25	2,16	0,72	1,44	1,2	1,216128
30	1,84	0,72	1,12	1,44	1,4512
35	1,76	0,607	1,153	1,55	1,55822857
40	1,76	0,96	0,8	1,4	1,4063
45	1,52	0,607	0,913	1,24	1,24497778
50	1,35	0,96	0,39	1,18	1,184032
55	1,2	0,607	0,593	1,08	1,08333223
60	1	0,74	0,26	0,92	0,9228
65	1,07	0,508	0,562	0,9	0,9023858
70	1,09	0,588	0,502	0,9	0,90205714
75	1,05	0,76	0,29	0,96	0,961792
80	1	0,76	0,24	0,9	0,901575

TABLA 7. Datos para Calcular la Constante Propia del Sensor de Infrarrojos

Para las siguientes condiciones se obtuvo un α0 de 0.042885V.m²:

1. S(x) = 1.12 Voltios Salida en voltios del sensor
2. X = 0.3 metros, Distancia de posicionamiento obstáculo, sensor
3. αi = 1, coeficiente de reflexión supuesto para una superficie regular blanca.

Hallar α0 es una tarea muy importante pues es el parámetro característico del sensor, este dato hace parte de las condiciones para medir distancia con sensores infrarrojos para continuar con la navegación y construcción de mapas de entorno.

Debe mencionarse que la característica paramétrica del sensor; se modeló para el sensor en especial, lo cual lleva a tomar muestras para deducir los parámetros de cada sensor para el ajuste de adquisición y normalizar el sistema.

2.1.7. RESULTADOS MEDICIÓN DE DISTANCIA CON INFRARROJOS Y HARDWARE RECONFIGURABLE. CUANTIFICIÓN DEL ERROR DE MEDIDA.

Las pruebas se ejecutaron con la captura de 1000 datos entregados por la tarjeta Spartan 3E para una misma situación de distancia. Para poder calcular el promedio.

El Grafico 14 muestra la medición de distancia implementada en la tarjeta FPGA, para una distancia de 40 cm. Como se observa, el patrón de medida se conserva sobre el valor de 40 cm con la presencia de datos como componente de ruido provenientes del microcontrolador y el sensor mismo; para una sesión de captura de 1000 datos para hacer análisis estadístico se hace necesario eliminar o minimizar primero las fuentes de este error, para poder cuantificar el error en la medida.
El valor promedio se conserva sobre el parámetro de medición físico, pero el error presente en la medición es significativo, por lo que se propone como trabajo futuro el cálculo del error en medición una vez se halla minimizado el ruido presente en el sistema.

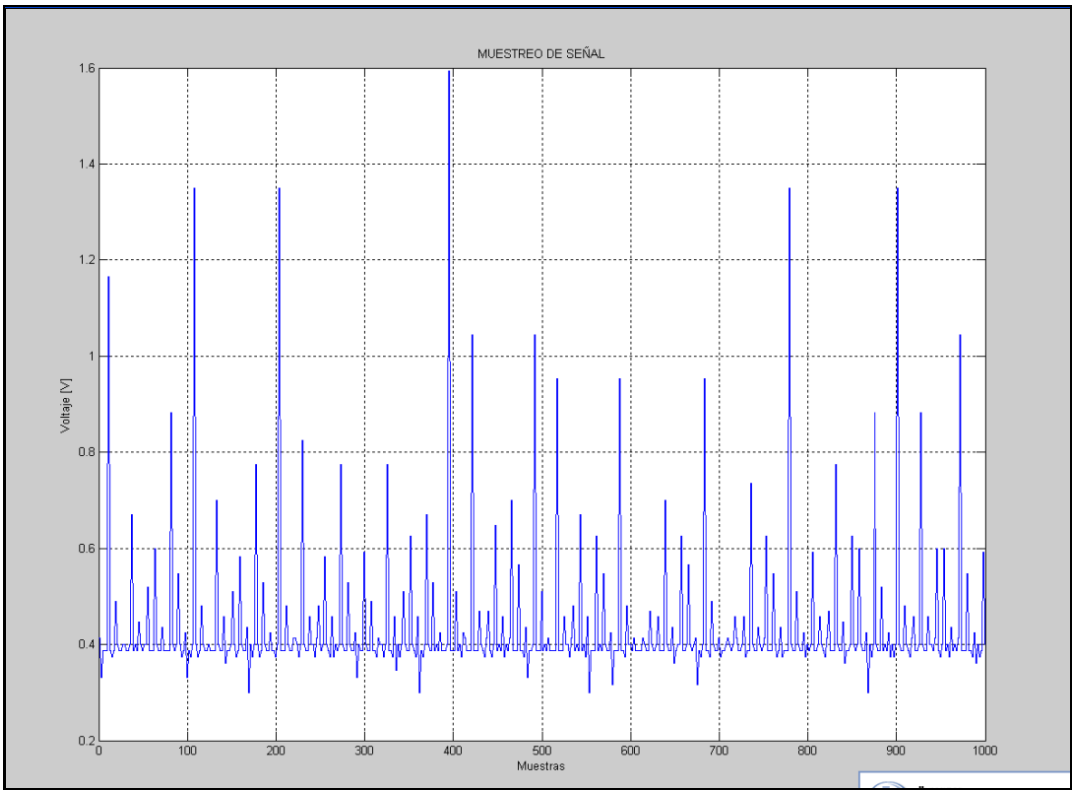


GRÁFICO 14. MEDICIÓN DE DISTANCIA. MUESTRAS.

2.1.8. RESULTADOS ESTRUCTURA FÍSICA DE SOPORTE PARA SENSORES INFRARROJOS.

De acuerdo al diseño de la Figura 30 de la sección 1.3.4.1. se muestra en la Foto 19 y 20. La estructura del soporte que se utiliza en el proyecto.

FOTO 19. Vista Superior. Estructura de soporte para sensores infrarrojos.

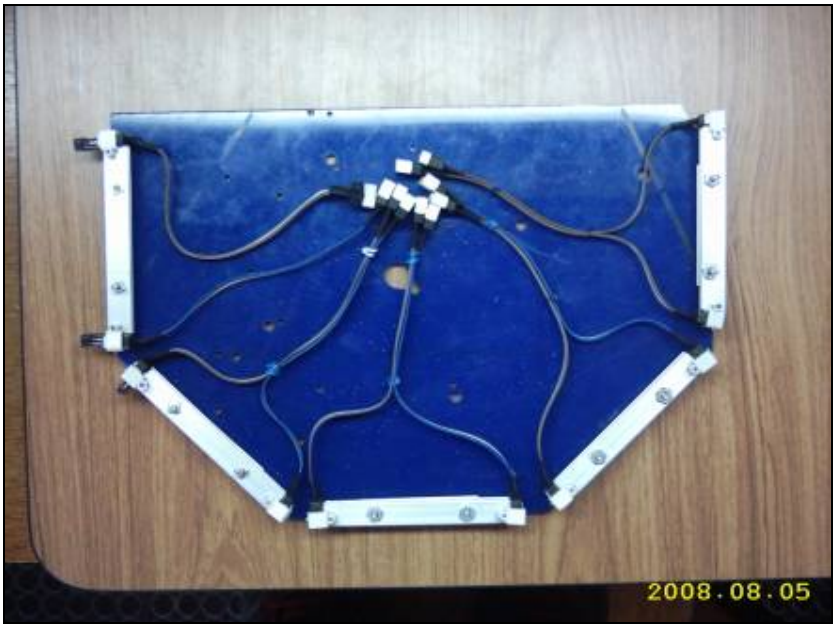


FOTO 20. Vista Inferior. Estructura de soporte para sensores infrarrojos

2.1.9. RESULTADOS ELECTRÓNICA Y ACONDICIONAMIENTO DE LOS SENSORES INFRARROJOS AL SOPORTE FÍSICO EN ANILLO.

El acople de los impresos al soporte de forma hexagonal se hizo teniendo en cuenta la accesibilidad de los componentes para encontrar fallas dadas en algún momento y la conexión del cableado a la interfaz con la tarjeta SPARTAN 3E, además que para hacer pruebas de medición de distancia y elaboración de mapas de entorno estáticos no es necesario acoplar a la plataforma móvil, a su vez que este sistema aislado comprende lo que se llama el bloque de percepción sensorial, como se muestra en las Fotos 21,22,23,24,25.

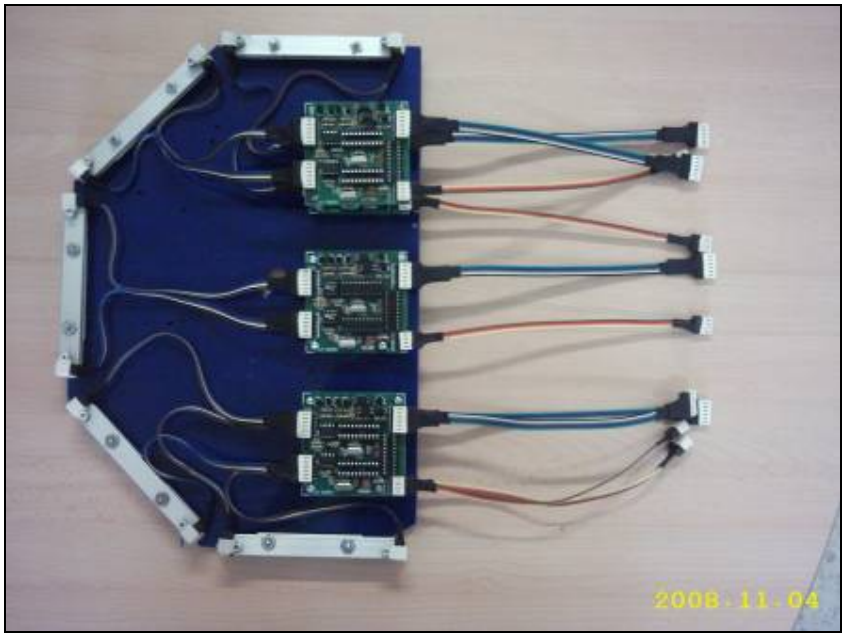


FOTO 24. Anillo de Sensores Infrarrojos

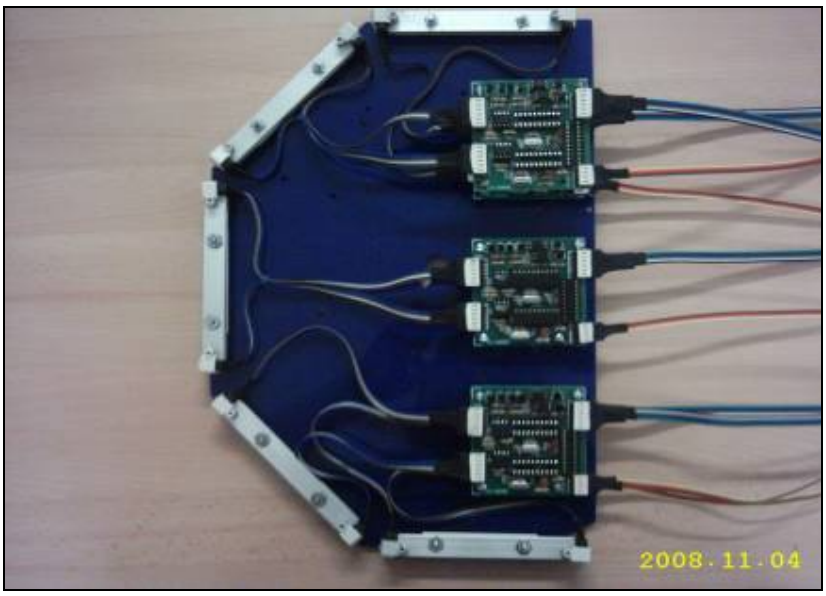


FOTO 25 . Anillo de Sensores Infrarrojos

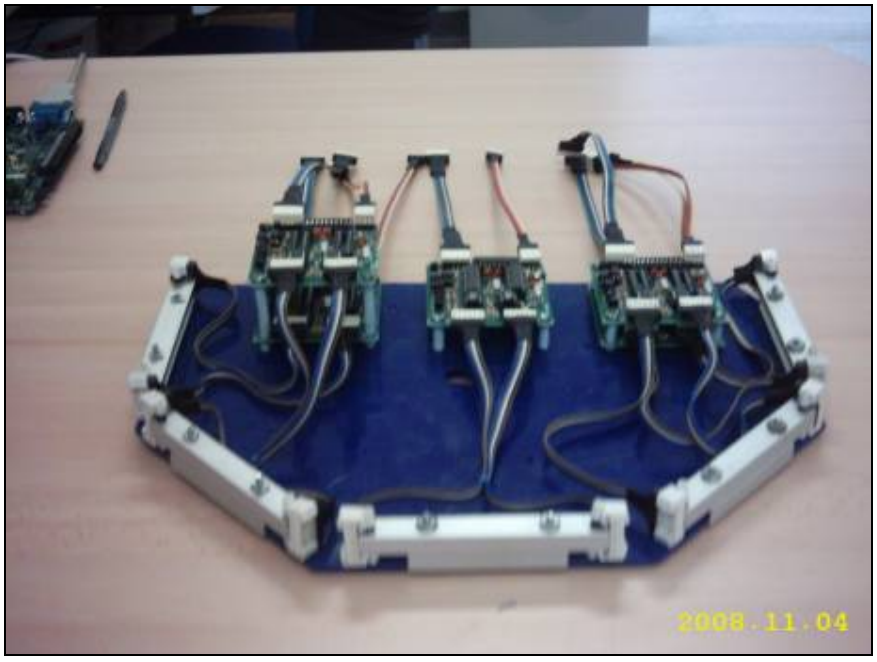


FOTO 26. . Anillo de Sensores Infrarrojos

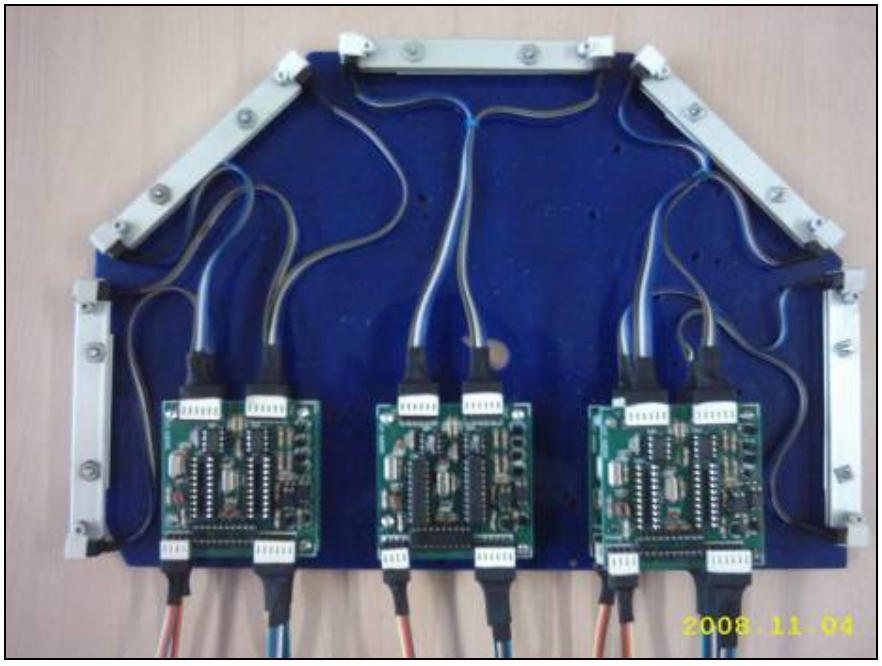


FOTO 27. Anillo de Sensores Infrarrojos

2.2. RESULTADOS FINALES.

En esta sección se muestra la implementación de los diferentes componentes desarrollados en el transcurso del proyecto para luego fusionar estos en la plataforma móvil autónoma.

FOTO 28. PLATAFORMA MÓVIL COMPLETA 3.

Se diseñó en descripción de hardware un código básico para verificar el funcionamiento de la plataforma completa, en el Video 7 se muestra una prueba de funcionamiento de navegación donde el sistema tiene como misión explorar el entorno y generar un mapa de entorno.
El ambiente de la prueba se perfila en un encerramiento cuadrado donde las paredes son de color blanco mate.
En los Gráficos 15,16 se observan diferentes mapas locales construidos por la plataforma en el PC. Donde la magnitud de las flechas en los Gráficos demarcan la distancia existente entre la plataforma y la presencia de un obstáculo en el rango de alcance del sensor.

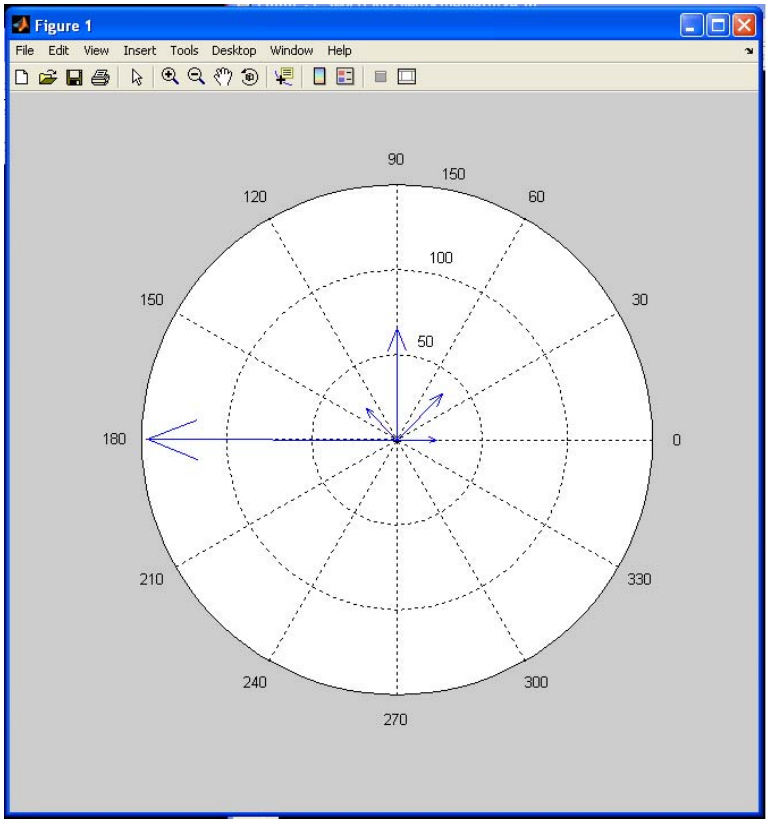


GRÁFICO 15. ENTORNO 1.

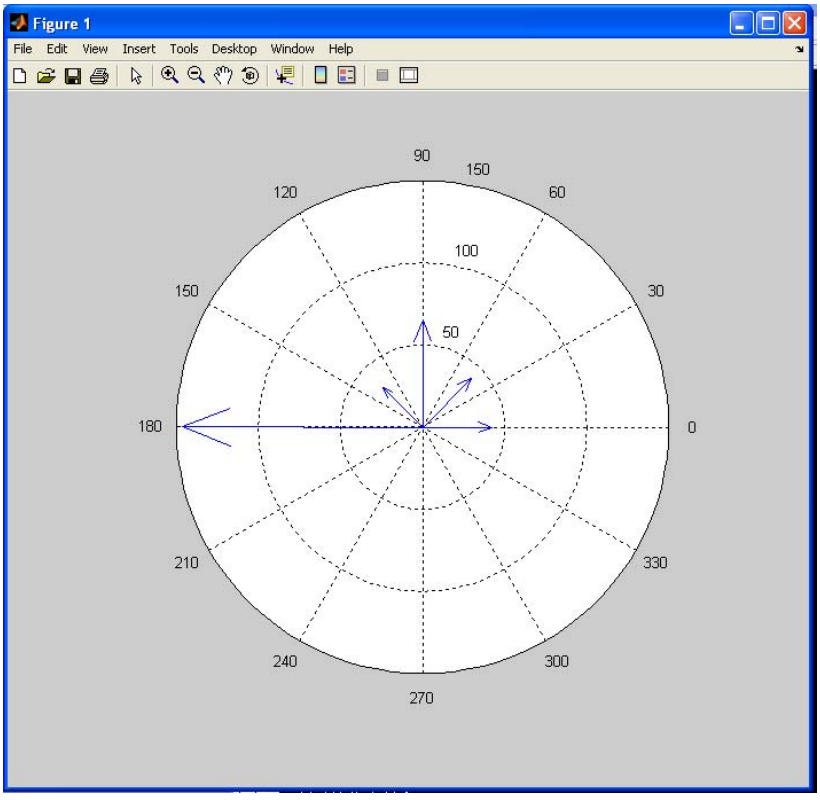


GRÁFICO 16. ENTORNO 2

El mapa de entorno se muestra con un arreglo de líneas denotadas con magnitud y dirección, Y y se refieren a la distancia que existe con respecto a un posible obstáculo.
Debe mencionarse que la naturaleza de la prueba esta encaminada a un ambiente controlado, los obstáculos son de color blanco.

3. TRABAJO FUTURO

3.1. MEDICIÓN DE DISTANCIAS CON SENSORES INFRARROJOS.

Como se asumió que la señal que excitaba el receptor de infrarrojos era completamente directiva las pruebas ejecutadas poseen una fuente de error que se basaba en el hecho de que la emisión de señal no seguía un comportamiento de cono o de ángulo sólido (ver Figura 26) sino una línea recta, así pues, el cálculo del ángulo de incidencia también se ve afectado por esta suposición. Entonces, se tiene una ventana al trabajo futuro para calcular la fuente de error, corregirla o eliminarla, con una base de datos previa que son las pruebas de laboratorio de la sección 1.3.1.3. para hacer un estudio que pueda calcular el error tanto para medición de distancia como para cálculo del ángulo de incidencia. Debe mencionarse que para el cálculo de ángulo de incidencia la fusión bisensorial es necesaria.

3.2. AMBIENTES NO CONTROLADOS.

En la sección 1.3.3 se calculó el parámetro propio del sensor de infrarrojos con una constante, que el ambiente de trabajo en este caso la superficie reflectora era una superficie de color blanco; de ahí todas las pruebas se hicieron con respecto a esta constante, coeficiente de reflexión con un valor de 1 (ambiente controlado).

El trabajo siguiente es que conociendo el parámetro propio del sensor, se piense en variar el parámetro de la superficie reflectora (coeficiente de reflexión αi) para así poder darle cabida a una línea de investigación cuyo fin sea la clasificación de colores y tipos de superficies con la implementación de sensores infrarrojos. Para así poder dotar a la plataforma móvil con un grado de autonomía mayor. Y también las aplicaciones que vean con buenos ojos la importancia de poder medir distancia, clasificar superficies por colores y por tipos de superficies.

3.3. NAVEGACIÓN AUTÓNOMA

- Gracias al hecho de que se entrega una plataforma móvil dotada de un conjunto de sensores heterogéneos y una base robusta para ejecutar pruebas, se promueve una línea de investigación para la aplicación de técnicas inteligentes de navegación que vayan acorde con la alta capacidad sensorial con la cual esta dotado este proyecto, es decir se entrega una herramienta para el estudio de nuevas técnicas de navegación con un grado de autonomía que empiece a caracterizarse por la implementación de *hardware* reconfigurable (FPGA), y sus ventajas.

- Teniendo a disposición en la plataforma móvil una brújula digital que entrega el Rumbo magnético (ángulo formado por la línea virtual de referencia de la brújula en la plataforma móvil y el Norte magnético de la tierra). Se propone hacer un estudio para proporcionar las órdenes necesarias a la dirección que tome la plataforma móvil para corregir el rumbo, se trataría de leer el valor actual de la orientación y compararla con la orientación deseada para modificar el rumbo en consecuencia.

  Implementar técnicas de navegación donde la localización puntual sea un dato que permita ejecutar correcciones de dirección de la plataforma móvil.

  Se precisa un estudio mas detallado de los alcances futuros que se podría lograr con dicho dispositivo en la navegación de plataformas móviles.

4. DISCUSIÓN

4.1 MARCO UTP.

Cuando se trabaja en el campo de la robótica móvil se espera poder llegar a un punto en el que el diseñador pueda afirmar que ha generado un sistema con un grado de autonomía que permita al mismo desenvolverse en un ambiente autónomamente y lograr sus objetivos de una forma óptima.

En el grupo de investigación en robótica y percepción sensorial GIROPS se han desarrollado dos plataformas de navegación, una de ellas es denominada P-METIN la cual es una herramienta de trabajo que esta basada en un anillo de sensores ultrasónicos, microcontroladores y un PC que ejecuta algoritmos de navegación; y la otra plataforma es llamada BAT la cual esta basada en un solo sensor ultrasónico que rota gracias a que está fijado al eje de un motor paso-paso, una tarjeta FPGA que envía datos del conversor análogo a digital al PC para ejecutar algoritmos de navegación y construcción de mapas de entorno.

Ahora se entrega una nueva plataforma la cual tiene como ventaja la independencia del PC como dispositivo principal a solo ser un tipo de periférico de interfaz grafica, los algoritmos de navegación son ejecutados desde la tarjeta de hardware reconfigurable, FPGAS.

El proyecto esta dotado con una autonomía ligada al hecho de que no depende del uso de un microprocesador (de un PC) ni de un microcontrolador (MCU) sino de una tarjeta FPGA la cual posee como naturaleza inherente ejecutar tareas en concurrencia de diferentes tipos, como puede estar controlando el *pwm* de los motores, también puede adquirir los datos entregados por los *encoders* y procesarlos, a su vez hacer operaciones matemáticas y enviar datos a la interfaz gráfica para ser vistos por el mundo exterior.

4.2. MARCO SENSORIAL

- Dentro del campo de la percepción sensorial se tiene un talón de Aquiles que se basa en la selección del tipo de sensor a implementar, en cuanto a los sensores ultrasónicos y los sensores infrarrojos puesto que los sensores US poseen un alcance para medición de distancia mayor que el sensor de infrarrojos, mientras que el sensor de infrarrojos puede entregar mayor resolución que el anterior. En algunos trabajos han implementado los dos tipos de sensores para mejorar los sistemas para construir mapas de entorno con buena precisión en cuanto al cálculo de distancias.

  De hecho el trabajo del grupo GIROPS en cuanto a la implementación del sensor ultrasónico no solo se ha limitado para medir distancias sino también para identificar tipos de superficies si son lisas o rugosas y para diferenciar paredes de esquinas. La continuación del trabajo en percepción sensorial se ha dado en la ejecución de este proyecto con la implementación de los sensores infrarrojos como medidores de distancia y el hardware reconfigurable, cabe mencionar el trabajo futuro con sensores infrarrojos como identificadores de colores de superficies.

- Dentro del campo de medición de distancias con sensores infrarrojos el modelamiento de Yair [40] maneja la excitación de los emisores en serie por lo que la potencia para emitir los rayos infrarrojos al medio es menor, en el proyecto esta parte de la excitación se independizó para aumentar la potencia y poder tener una señal en el medio mas representativa que ayude a mejorar la recepción de la señal infrarroja.

- Los medidores de distancias con fundamento en sensores infrarrojos que ofrece el mercado tienen un tiempo de respuesta muy alto comparado con el diseño implementado en el proyecto el cual es relativamente corto.

CONCLUSIONES

- Se entrega una herramienta robusta (plataforma móvil) dotada de un conjunto de sensores que facilita la navegación autónoma para continuar con la línea de investigación para el desarrollo de mapas de entorno estáticos y dinámicos.

- Las pruebas de funcionamiento de la plataforma móvil demuestran la correcta implementación del trabajo con los algoritmos diseñados de navegación y mapas de entorno, para que ésta se movilice de forma autónoma y pueda construir un mapa de entorno con evasión de obstáculos. Lo cual es muy importante para los trabajos siguientes en cuanto a la investigación en robótica móvil y la implementación de hardware reconfigurable.

- Gracias al diseño que se elaboró con una base teórica y con pruebas de ensayo y error se logró obtener un circuito que polarizará los sensores infrarrojos para hacer las pruebas de laboratorio sobre medición de distancia y poder implementar el hardware reconfigurable en la aplicación generando el algoritmo para medir distancias y la elaboración de mapas de entorno.

- La plataforma móvil se dotó de un anillo de sensores infrarrojos que están posicionados en la periferia de un medio hexágono para poder percibir la existencia de un obstáculo y la distancia a la cual se encuentra de la plataforma , teniendo en cuenta que se trabaja en un ambiente controlado (superficies de color blanco).

- La implementación del hardware reconfigurable en la aplicación de la plataforma móvil con anillo de sensores infrarrojos y navegación, denota la idea de utilizar este tipo de tecnología con mas dedicación en cuanto a sacar el mayor provecho hacia la aplicación y mejorarla de ser posible, no sólo para la robótica móvil, sino expandir el potencial del hardware reconfigurable a proyectos en el campo de la investigación sea por la línea de instrumentación y control de procesos, o por la línea de análisis de sistemas de potencia.

BIBLIOGRAFÍA

[1] Adams. M.D, "*Sensor Modelling, Design and Data Processing for Autonomous Navigation.", World Scientific Publishing, Series in Robotics and Intelligent Systems. Singapore, 1999.*

[2] Agre, Philip E. y Chapman David. *" What are plans for?"*, *Robotics and Autonomous Systems.* No. 6 (1990); p. 17 – 34

[3] *IEEE Robotics and Automatization Soviet 2001*

[4] Akbarally, Huzefa y Kleeman, Lindsay. *"A sonar sensor for accurate 3D target localisation and classification" IEEE International conference on robotics and automation*. 1995.

[5] Mataric, Maja. "A distributed model for mobile robot environment-learning and navigation."Cambridge, MA, 1990.

[6] Parker, L. *"Current State of the Art in Dist Distributed Autonomous Mobile Robotic"* ,*Distributed Autonomous Robotic Systems*. Tokyo. Vol 4, (2000); p. 3-12.

[7] Yamasaki, H. *"Intelligent Sensing Technology.", Journal of the Japan Society of Precision Engineering. Vol. 55, No. 9; 1989*.

[8] Everett, H.R. y Peters, A K. *" Sensors for Mobile Robots"* , Wellesley, MA,1995.

[9] Feng, L, Borenstein, J y Everett, H.R. *" Where am I? " Sensors and methods for mobile robot positioning.* Universidad de Michigan. 1996.

[10] Adams. M.D. *"Sensor Modelling, Design and Data Processing for Autonomous Navigation." World Scientific Publishing , Series in Robotics and Intelligent Systems. Singapore*. Vol. 13. 1999.

[11] mcbtec.com. "Funcionamiento_*Encoder*"
< www.mcbtec.com/Funcionamiento_*Encoder*.pdf >.

[12] superrobotica.com, " sensor brújula digital "< http://www.superrobotica.com/S320160.htm >

[13] Borges, G, Nogueira, A y G. S. Deep, *"Characterization of a Trajectory Recognition Optical Sensor for an Automated Guided Vehicle.", IEEE Transactions on Instrumentation and measurement,* Vol. 49, Nº. 4, pp. 813 – 819. 2000

[14] Malik R. y Yu H., *"The Infrared Detector Ring: Obstacle Detection for an Autonomous Mobile Robot." IEEE 35th Midwest Symposium on Circuit and Systems,* 1992.

[15] Flynn A., *"Combining sonar and infrared sensors for mobile robot navigation.", The International Journal of Robotics Research,* Vol. 7, Nº. 6, pp. 5 – 14. 1988.

[16] Sabatini A, Genovese V, Guglielmelli E, Mantuano A, Ratti G, y Dario P. "*A low-cost, composite sensor array combining ultrasonic and infrared proximity sensors.", Procedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 120 - 126. Pittsburgh, 1995.

[17] Novotny P y Ferrier N, *" Using infrared sensors and the Phong illumination model to measure distances." Procedings of the IEEE International Conference on Robotics and Automation,* pp. 1644 – 1649. Detroit, 1999.

[18] Blanes, F, " Percepción y Representación del Entorno en Robótica Móvil.", PhD. Tesis. Departamento de Informática de Sistemas y Computadores. Universidad Politécnica de Valencia. Valencia, España. Septiembre de 2000.

[19] Aytaç T y Barshan, B, *"Differentiation and localization of target primitives using infrared sensors.", Procedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 105 - 110. Lausanne, Suiza Octubre 2002.

[20] Aytaç T y Barshan, B, *" Rule-based target differentiation and position estimation based on infrared intensity measurements. ", Optical Engineering,* vol.42, Nº.6, pp.1766-1771, June 2003.

[21] Guglielmelli E, Genovese V, Dario P y Morana G, *"Avoiding Obstacles by Using a Proximity US/IR sensitive Skin.",* Procedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2207 - 2213. Yokohama, Japan. July 1993.

[22] Benet G, Blanes F, Simón J y P. Perez, *"Using infrared sensors for distance measurement in mobile robots*.", *Robotics and Autonomous Systems*, Vol. 40, Nº 4, pp. 255 – 256. Septiembre 2002.

[23] Murphy, R.R."*Introduction to All Robotics.",* MIT Press. 2000.

[24] Simó, J.E., *" Una arquitectura basada en motivaciones para el control de robots móviles."*, Tesis Doctoral. Universidad Politécnica de Valencia. 1997.

[25] Brooks, R.A., *"Elephants don't play chess."* , *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back.* 3-15 MIT Press. London. 1991.

[26] Arbib, M.*" Schema Theory, The Handbook of Brain Theory and Neural Networks.",* MIT Press. Cambridge, 1986.

[27] Larkin, Ronald C.*" Reactive Robotic Systems" Internal Report.", Georgia Institute of Technology.* Atlanta, Georgia. 1994.

[28] Marzo 2006.SEMINARIO ALCABOT'2006 e HISPABOT'06.1SEMINARIO: Diseño y Construcción de Microrrobots. CONTROL DE MOTORES" SEM06_motores"
< http://alcabot.org/seminario2006/SEM06_motores.pdf>.

[29].disam.upm.es cybertech "Configuraciones Movimiento"
<www.disam.upm.es/cybertech/Nacional/Documentos/Otros/configuracionesmovimiento.pdf>

[30]] itoosoft.com. "Odometría"
< http://www.itoosoft.com/motorolos/odometria/odometria.html>

[31] Linea Activista

[32] Braitneberg,V. *" Vehicles :Experiments in Synthetic Psychology "*, Cambridge. MIT. 1984

[33] Gallistel.C. " The Organization of Learning", 1990

[34] Kuipers, Benjamin y Byun, Yung-Tai. *"A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations.", Journal of Robotics and Autonomous Systems,* No.8; 1991; p. 47–63.

[35] Donnet, J.G. *"Analisys and synthesis in the design of locomotor and spatial competences for a multisensory mobile robot."* , Ph.D. Disertaciòn , Universidad de Edinburgo. 1992

[36] Kurz, Andreas. *"Constructing maps for mobile robot navigation based on ultrasonic range data", IEEE Transactions on systems, man and cybernetics*. Vol. 26, No. 2 ,1996.

[37] Moravec, Hans P. *"Sensor Fusion in Certainty Grids for Mobile Robots*." AI Magazine, Vol. 9; No. 2; 1988, p. 61-74.

[39] Elfes, A. *"Sonar Based Real World Mapping and Navigation.", IEEE Journal of Robotics and Automation.* Vol. 3; No. 3; 1987.

[40] Lim, J.H y Cho, D.W. *"Physically based sensor modelling for a sonar map in a specular environment.", Proc. Int. Conf on robotics and automation*. 1992. p. 1714-1719.

[41] Borenstein J y Koren Y, *"Histogramic in-motion mapping for mobile robot obstacle avoidance.", IEEE Journal of robotics and automation*. Vol. 7; No. 4; 1991.

ANEXOS

ANEXO A. DESCRIPCIÓN DE HARDWARE QUE CONTROLA MOTORES DC.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity motoresdc is
        generic
        (
                frec_pwm_1              : positive := 5_000
        );
   Port
        (
                clk,parado              : in std_logic;
                dir_in_dc               : in std_logic_vector(1 downto 0);
      m_dc, dir_dc              : out std_logic_vector(1 downto 0)
--los mototres estan inactivos en '1'
                 );
end motoresdc;
architecture Behavioral of motoresdc is
signal clk_pwm                  : std_logic;
signal counter                  : integer range 0 to 50_000_000;
signal state                    : integer range 0 to 1:=0;
signal porcentaje_pwm           : integer range 0 to 100;--% trabajo util
begin
process(clk)
        begin
                if rising_edge(clk) then
                        clk_pwm <= '0';
                        counter <= counter + 1;
                                if counter >= frec_pwm_1 then
                                counter <= 0;
                                clk_pwm <= '1';
                                end if;
                end if;
END PROCESS;
process ( clk)
        begin
                if rising_edge(clk) then
                        if clk_pwm='1' then
                        case state is
                                when 0 =>
                                        if porcentaje_pwm>=5 then
                                        state <= 1;
                                        porcentaje_pwm<=1;
                                        else
                                        porcentaje_pwm<=porcentaje_pwm+1;
                                        state <= 0;
                                        end if;
                                when 1 =>
                                        if porcentaje_pwm>=95 then
                                        state <= 0;
                                        porcentaje_pwm<=0;
                                        else
                                        porcentaje_pwm<=porcentaje_pwm+1;
                                        state <= 1;
```

```
                                        end if;
                                when others=>
                                        state <= 0;
                        end case;
                        end if;
                end if;
        end process;
dir_dc<= not dir_in_dc;
process (state)
        begin
        case state is
                        when 0 =>
                                m_dc<="11";
                        when 1=>
                                if parado='1' then
                                        m_dc<="11";
                                else
                                        m_dc<="00";
                                end if;
                        when others =>
                                        m_dc<="00";
                end case;
        end process;
end Behavioral;
```

ANEXO B. DESCRIPCIÓN DE HARDWARE PARA ENCODERS

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
---- Uncomment the following library declaration if instantiating
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
entity filterencoder is
    Port (          clk : in  STD_LOGIC;
                    encoder_in_a : in  STD_LOGIC;
           encoder_in_b : in  STD_LOGIC;
          controlE : in  STD_LOGIC;
          cont1 :out integer:=0 ;
 cont2 :out integer:=0);
end filterencoder;
architecture Behavioral of filterencoder is
signal q1_temp: STD_LOGIC;
signal q2_temp: STD_LOGIC;
begin
filter: process(clk)
variable encoder_in: STD_LOGIC_VECTOR(7 downto 0);
begin
if clk'event and clk='1' then
encoder_in := encoder_in(6 downto 0) & encoder_in_a;
case encoder_in is
        when "00000000" => q1_temp <= '0';
        when "11111111" => q1_temp <= '1';
when others => q1_temp <= q1_temp;
end case;
end if;
end process filter;
filter2: process(clk)
variable encoder_in2: STD_LOGIC_VECTOR(7 downto 0);
begin
if clk'event and clk='1' then
encoder_in2 := encoder_in2(6 downto 0) & encoder_in_b;
case encoder_in2 is
when "00000000" => q2_temp <= '0';
when "11111111" => q2_temp <= '1';
when others => q2_temp <= q2_temp;
end case;
end if;
end process filter2;
process(q1_temp)
variable CONTADOR :integer:=0;
begin
if rising_edge(q1_temp)then
  CONTADOR:=CONTADOR + 1;
  cont1 <= CONTADOR;
  if controlE='1' then
  CONTADOR:=0;
  end if;
end if;
end process;
process(q2_temp)
variable CONTADOR2 :integer:=0;
begin
if rising_edge(q2_temp)then
```

```
  CONTADOR2:=CONTADOR2 + 1;
  cont2 <= CONTADOR2;
  if controlE='1' then
  CONTADOR2:=0;
  end if;
end if;
end process;
end Behavioral;
```

ANEXO C. DESCRIPCIÓN DE HARDWARE PARA CAPTURA DE DATOS DE BRÚJULA DIGITAL

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
---- Uncomment the following library declaration if instantiating
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
entity BRUJULADISPLAY is
    Port ( clk : in  STD_LOGIC;
           reset : in  STD_LOGIC;
           medicion : in  STD_LOGIC;
           inicia_conversion : in  STD_LOGIC;
           clk_10KHZ : out  STD_LOGIC;
           segmentos : out  STD_LOGIC_VECTOR (7 downto 0);
           mux : out  STD_LOGIC_VECTOR (3 downto 0));
end BRUJULADISPLAY;
architecture Behavioral of BRUJULADISPLAY is
Signal Clk_Contador: std_logic;
signal vis1        : std_logic_vector(7 downto 0);
Signal clk_10KHZ_1 : std_logic;
signal q1_temp: STD_LOGIC;
signal    counter             : integer range 0 to 10_000:= 0;
signal    count1s             : integer range 0 to 10:= 0;
--bloque de visualizacion
signal cont: integer range 0 to 400_000;
signal dig1_s,dig2_s,dig3_s:integer range 0 to 9;
signal dig_mux: integer range 0 to 10;
signal mux_1 :  STD_LOGIC_VECTOR (3 downto 0);
begin
clk_10KHZ<=clk_10KHZ_1;
mux<=mux_1;
Divisor:process(clk)
Variable ContContador: integer range 0 to 5000 := 0;
begin
 if  clk = '1' and clk'event then
  if ContContador = 5000 then
ContContador := 0;
 Clk_Contador <= '1';
clk_10KHZ_1 <= '1';
 else
ContContador := ContContador + 1;
Clk_Contador <= '0';
clk_10KHZ_1 <= '0';
end if;
end if;
end process;
filter: process(clk,inicia_conversion)
variable star: STD_LOGIC_VECTOR(5 downto 0);
begin
if clk'event and clk='1' then
star := star(4 downto 0) & inicia_conversion;
case star is
when "000000" => q1_temp <= '0';
when "111111" => q1_temp <= '1';
when others => q1_temp <= q1_temp;
end case;
end if;
```

```
end process filter;
process(Clk_Contador,medicion,reset,q1_temp)
variable medida,medida_vis, midiendo,grados,empieza,retardo: integer :=0;
begin
vis1<=CONV_STD_LOGIC_VECTOR(medida,8);
if reset='1' then  --incializamos las variables del circuito
medida:=0;     --  variable  para  llevar  la  cuenta  del tiempo
    empieza:=0;
    retardo:=0;
elsif Clk_Contador='1' and Clk_Contador'event then
if q1_temp='1' then  -- si han pulsado Start, debemos comenzar a medir
--empieza:=1;
retardo:=1;
end if;
if retardo=1 then
counter <= counter + 1;
if counter >= 9_999 then
counter <= 0;
count1s <= count1s+1;
if   count1s>=1 then
counter <= 0;
count1s<=0;
empieza:=1;
end if;
end if;
if  empieza=1  then
if medicion='1'  then
 medida:=medida+1; --cada   0.1ms --> 1º incrementamos   al   variable   medida(un grado)
  midiendo:=1;
  elsif medicion='0' and midiendo=1 then  -- en cuanto acabe la medicion
    -- actualizamos el valor
   medida_vis:=medida;   -- actualizamos la var medida_vis
midiendo:=0;
empieza:=0;
 retardo:=0;
end if;
end if;
end if;
end if;
grados:=medida_vis;
end process;
grados:process(vis1)
 variable aux:integer range 0 to 255;
 variable dig1,dig2,dig3:integer range 0 to 9;
begin
aux := conv_integer(vis1);
 dig1:= 0;
 for i in 0 to 9 loop
  if aux >= 100 then
  aux := aux - 100;
   dig1 := dig1 + 1;
   end if;
   end loop;
     dig2 := 0;
     for j in 0 to 9  loop
     if aux >=10 then
     aux  := aux - 10;
     dig2 := dig2 + 1;
       end if;
      end loop;
```

```
          dig3 := aux;
          dig1_s <= dig1;
          dig2_s <= dig2;
          dig3_s <= dig3;
end process grados;
secuencial: process (clk,reset,mux_1)
begin
 if reset='1' then
       cont <= 0;
       mux_1 <= "1111";
 elsif  rising_edge(clk) then
  cont <= cont + 1;
  case cont is
              when 100_000 => mux_1 <= "0111";
              when 200_000 => mux_1 <= "1011";
              when 300_000 => mux_1 <= "1101";
              when 400_000 => mux_1 <= "1110";
               cont <= 0;
              when others => mux_1 <= mux_1;
    end case;
   end if;
end process secuencial;
decodificador:process(dig_mux)
begin
 case dig_mux is
  when 0 => segmentos <= "10000001";
  when 1 => segmentos <= "11001111";
  when 2 => segmentos <= "10010010";
  when 3 => segmentos <= "10000110";
  when 4 => segmentos <= "11001100";
  when 5 => segmentos <= "10100100";
 when 6 => segmentos <= "10100000";
  when 7 => segmentos <= "10001111";
  when 8 => segmentos <= "10000000";
  when 9 => segmentos <= "10000100";
  when 10=> segmentos <= "10011100";
  when others=>segmentos<= "10000001";
  end case;
  end process decodificador;
selector: process(dig1_s, dig2_s, dig3_s, mux_1)
begin
 case mux_1 is
       when "0111" => dig_mux <= dig1_s;
       when "1011" => dig_mux <= dig2_s;
       when "1101" => dig_mux <= dig3_s;
       when "1110" => dig_mux <= 10;
       when others => dig_mux <= 0;
  end case;
end process selector;
end Behavioral;
```

ANEXO D. DESCRIPCIÓN DE HARDWARE QUE GENERA SEÑAL MODULADA.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity emisro is
    Port ( emiter_signal_out : out  STD_LOGIC;
           clk : in  STD_LOGIC;
           control : in  STD_LOGIC);
end emisro;
--============================================================================
architecture emisro_arch of emisro is
--============================================================================
signal cont_50      : integer range 0 to 100000 := 0;
signal emiter_50            : std_logic;
signal cont_40      : integer range 0 to 1250 := 0;
signal emiter_40            : std_logic;
--============================================================================
begin
--============================================================================
 process(clk,control)
begin
   if rising_edge(clk) then
     if control='1' then
              cont_50 <=cont_50+1;
                 emiter_50<='1';
                    if cont_50>=49999 then
                           emiter_50<='0';
                            if cont_50>=99999 then
                              cont_50<=0;
                            end if;
                         end if;
             end if;
           end if;
end process;
     process(clk,control)
begin
      if rising_edge(clk) then
             if control='1' then
                cont_40 <=cont_40+1;
                       emiter_40<='1';
                    if cont_40>=625 then
                                emiter_40<='0';
                                 if cont_40>=1250 then
                           cont_40<=0;
                        end if;
              end if;
         end if;
    end if;
end process;
--============================================================================
emiter_signal_out <=emiter_50 and emiter_40;
end emisro_arch;
```

ANEXO E. CÓDIGO EN ENSAMBLADOR PARA EL MCU JK8

```
                include derivative.inc
                xdef    INICIO
;*****************************************************************
;* Definición de variables en la RAM                             *
;*****************************************************************
            org     Z_RAMStart
VAR1     ds.b    1
dir         ds.b    1
;*****************************************************************
;* Programa en la FLASH                                          *
;*****************************************************************
            org     ROMStart
INICIO:     rsp                 ; Inicializa registros de la CPU
            clra
            CLR     VAR1
            clr     dir
            ldhx      dir           ;posiciono el puntero en esta direccion
            sthx    dir
            mov     #$FF,DDRD
            mov     #$FF,PTD
            bset    0,CONFIG1       ; Desactiva el cop
            mov     #$08,ADSCR      ; Configura el ADC (canal 8->PTd3)
            mov     #$00,ADICLK     ; 14microS time sample para cktal de 2 Mhz
            mov     #$40,SCC1       ;ACTIVO MODULO SCI
            mov     #$08,SCC2       ; ACTIVO TRANSMISION
            mov     #$01,SCBR       ;VELOCIDAD DE TRANSMISION  7812.5

ADCONVERSION:   bclr    5,ADSCR         ; Solicita una conversión
                    brclr         7,ADSCR,*       ; Espera por resultado de conversión
                    lda                 ADR             ; Carga el resultado de la conversión
                   jmp        ALMACENA            ; rutina para almanenar dato

ALMACENA:        LDX        VAR1
                 STA        dir,X
                  AIX          #1
                 STX        VAR1
                  lda        VAR1
                 CMP         #$F0
                   BNE               ADCONVERSION
                    jmp       BANDERA_1
BANDERA_1:        LDA     #$FF
                  STA     PTD
                  JMP     ENVIO
ENVIO:
                             LDX       VAR1
                  lda        dir,X
                 LDX        SCS1
                 mov        dir,SCDR
                BRCLR    7,SCS1,*
                  ldx        VAR1
                 AIX         #-1
                 STX         VAR1
                 lda            VAR1
                  CMP                 #$00
                  BNE     ENVIO
                 JMP     BANDERA_2
BANDERA_2          LDA     #$00
                    STA     PTD
                     JMP     ADCONVERSION
```

ANEXO F. DESCRIPCIÓN DE HARDWARE PARA COMUNICACIÓN SERIAL

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity recepcion_serial is
	generic
	(
		-- baud = 1 / (transmit_rate_rx * 20ns)
		-- default baud = 7812 trasmitrate_rx= 6400
		transmit_rate_rx		: positive := 6401
	);
	port
	(
		clk			: in std_logic;
		data_out_serial : out std_logic_vector (7 downto 0):="00000000";
		serial_ready_rx : out std_logic:='0';
		rx			: in std_logic;
		MCU_tx		: in std_logic
	);
end recepcion_serial;

architecture Behavioral of recepcion_serial is
	signal count_bit		:  integer range 0 to 6400:=0
	type	state_type is (st_idle, st_brun, st_bit0, st_bit1, st_bit2, st_bit3, st_bit4, st_bit5,
st_bit6, st_bit7, st_bstop);
	signal state			: state_type := st_idle;
	signal data			: std_logic_vector (7 downto 0):="00000000";
	signal serial_ready		:	std_logic:='0';
begin
data_out_serial<=data when serial_ready='1';
serial_ready_rx<=serial_ready;
process (clk)
begin
	if rising_edge(clk) then
		serial_ready<='0';
		if rx='1' then
			case state is
				when st_idle =>
					count_bit <=0;
					if MCU_tx ='0' then
						state <= st_brun;
					else
						state <= st_idle;
					end if;
				when st_brun =>			--bit de arranque--
					count_bit <= count_bit + 1;
					if count_bit >= transmit_rate_rx/2-1 then
						if MCU_tx ='0' then
							state <= st_bit0;
							count_bit <= 0;
						else
							state <= st_idle;
							count_bit <= 0;
						end if;
					end if;
				when st_bit0 =>
					count_bit <= count_bit + 1;
					if count_bit >= transmit_rate_rx-1 then
						state <= st_bit1;
```

```
                data(0) <= MCU_tx;
                count_bit <= 0;
        end if;
when st_bit1 =>
        count_bit <= count_bit + 1;
        if count_bit >= transmit_rate_rx-1 then
                state <= st_bit2;
                data(1) <= MCU_tx;
                count_bit <= 0;
        end if;
when st_bit2 =>
        count_bit <= count_bit + 1;
        if count_bit >= transmit_rate_rx-1 then
                state <= st_bit3;
                data(2) <= MCU_tx;
                count_bit <= 0;
        end if;
when st_bit3 =>
        count_bit <= count_bit + 1;
        if count_bit >= transmit_rate_rx-1 then
                state <= st_bit4;
                data(3) <= MCU_tx;
                count_bit <= 0;
        end if;
when st_bit4 =>
        count_bit <= count_bit + 1;
        if count_bit >= transmit_rate_rx-1 then
                state <= st_bit5;
                data(4) <= MCU_tx;
                count_bit <= 0;
        end if;
when st_bit5 =>
        count_bit <= count_bit + 1;
        if count_bit >= transmit_rate_rx-1 then
                state <= st_bit6;
                data(5) <= MCU_tx;
                count_bit <= 0;
        end if;
when st_bit6 =>
        count_bit <= count_bit + 1;
        if count_bit >= transmit_rate_rx-1 then
                state <= st_bit7;
                data(6) <= MCU_tx;
                count_bit <= 0;
        end if;
when st_bit7 =>
        count_bit <= count_bit + 1;
        if count_bit >= transmit_rate_rx-1 then
                state <=st_bstop;
                data(7) <= MCU_tx;
                count_bit <= 0;
        end if;
when st_bstop =>                                        --bit de paro--
                if count_bit <= transmit_rate_rx-1 then
                        count_bit <= count_bit + 1;
                else
                        if      rs232_dte_rxd ='1' then
                                state <= st_idle;
                                count_bit <= 0;
                                serial_ready<='1';
```

```
                                                        end if;
                                                end if;
                                when others =>
                                        state <= st_idle;
                        end case;
                end if;
--      else
--              state <= st_idle;
        end if;
end process;
end Behavioral;
```

ANEXO G. DESCRIPCIÓN DE HARDWARE PARA EL CONTROL DE MEMORIA

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;
library UNISIM;
use UNISIM.VCOMPONENTS.ALL;
entity bram_16s9s9 is
	port
	(
		CLK                                     : in std_logic;
--=====================================================================

		DATA_INA		: in std_logic_vector (8 downto 0);
--		ENABLEA			: in std_logic;
		WRITE_ENA		: in std_logic;
--		SET_RESETA		: in std_logic;
		ADDRESSA: in std_logic_vector (10 downto 0):="00000000000";
--		DATA_OUTA		: out std_logic_vector (8 downto 0);
--=====================================================================
--		DATA_INB                        : in std_logic_vector (8 downto 0);
--		ENABLEB                         : in std_logic;
--		WRITE_ENB                       : in std_logic;
--		SET_RESETB                      : in std_logic;
		ADDRESSB: in std_logic_vector (10 downto 0):="00000000000";
		DATA_OUTB: out std_logic_vector (8 downto 0):="000000000"
--=====================================================================
	);
end bram_16s9s9;

--=====================================================================

architecture bram_16s9s9_arch of bram_16s9s9 is

--=====================================================================
--	signal ADDRESSA: std_logic_vector (10 downto 0):="00000000000";
--	signal WRITE_ENA	: std_logic;
--	signal DATA_INA		: std_logic_vector (8 downto 0);
--	signal ENABLEA		: std_logic;
--	signal SET_RESETA: std_logic;
--	signal DATA_OUTA	: std_logic_vector (8 downto 0);
--=====================================================================
--	signal ADDRESSB : std_logic_vector (10 downto 0):="00000000000";
--	signal WRITE_ENB	: std_logic;
	signal DATA_INB		: std_logic_vector (8 downto 0):="000000000";
--	signal ENABLEB		: std_logic;
--	signal SET_RESETB: std_logic;
--	signal DATA_OUTB	: std_logic_vector (8 downto 0);
begin

  RAMB16_S9_S9_inst : RAMB16_S9_S9
	generic map
	(
		WRITE_MODE_A => "READ_FIRST",
		WRITE_MODE_B => "READ_FIRST",
		INIT_A =>	"000000000",
		INIT_B =>	"000000000",
		SRVAL_A =>  "000000000",
		SRVAL_B =>  "000000000"
	)
```

```
        port map
        (
                DOA => open ,                                                   -- Port A 8-bit
Data Output
    DOB => DATA_OUTB (7 downto 0),    -- Port B 8-bit Data Output
    DOPA => open,                                       -- Port A 1-bit Parity Output
    DOPB => DATA_OUTB (8 downto 8),   -- Port B 1-bit Parity Output
    ADDRA => ADDRESSA (10 downto 0),  -- Port A 11-bit Address Input
    ADDRB => ADDRESSB (10 downto 0),  -- Port B 11-bit Address Input
    CLKA => CLK,                                        -- Port A Clock
    CLKB => CLK,                                        -- Port B Clock
    DIA => DATA_INA (7 downto 0),     -- Port A 8-bit Data Input
    DIB => DATA_INB (7 downto 0),     -- Port B 8-bit Data Input
    DIPA => DATA_INA (8 downto 8),    -- Port A 1-bit parity Input
    DIPB =>  DATA_INB (8 downto 8),                 -- Port-B 1-bit parity Input
    ENA => '1',                             -- Port A RAM Enable Input
    ENB => '1',                                     -- PortB RAM Enable Input
    SSRA => '0',                            -- Port A Synchronous Set/Reset Input
    SSRB => '0',                            -- Port B Synchronous Set/Reset Input
    WEA => WRITE_ENA,                               -- Port A Write Enable Input
    WEB => '0'                              -- Port B Write Enable Input
        );

DATA_INB<=(others =>'0');

--================================================================
end bram_16s9s9_arch;
```

ANEXO H. DESCRIPCIÓN DE HARDWARE COMUNICACIÓN SERIAL TARJETA-PC

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
---- Uncomment the following library declaration if instantiating
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
--================================================================
entity transmision_serial is
	generic
	(
		-- baud = 1 / (transmit_rate * 20ns)
		-- default baud = 115200 for transmit_rate = 434
		transmit_rate		: positive := 434
	);
	port
	(
		clk		: in std_logic;
		DATA_IN		: in std_logic_vector (7 downto 0):="00000000";
		serial_ready: out std_logic:='0';
		init		: in std_logic;
		rs232_dte_txd		: out std_logic := '1'
	);
end transmision_serial;
architecture Behavioral of transmision_serial is
--================================================================
	signal clk_bit	:	std_logic;
	signal init_int	:	std_logic;
	signal count_bit :  integer range 0 to 435 := 0;
	type   state_type is (st_idle, st_brun, st_bit0, st_bit1, st_bit2, st_bit3, st_bit4, st_bit5,
st_bit6, st_bit7, st_bstop);
	signal state	: state_type := st_idle;
	signal data		: std_logic_vector (7 downto 0):="00000000";
	signal serial_ready_flag	:	std_logic:='0';
--================================================================
Begin
	process (clk)
	begin
		if rising_edge(clk) then
			clk_bit <= '0';
			count_bit <= count_bit + 1;
			if count_bit >= transmit_rate then
				count_bit <= 0;
				clk_bit <= '1';
			end if;
		end if;
	end process;
process (clk)
begin
	if rising_edge(clk) then
		if state=st_idle and init='1' then
			data <= DATA_IN;
			init_int <= '1';
		end if;
		if clk_bit = '1' then
			case state is
				when st_idle =>
```

```
                                        rs232_dte_txd <='1';
                                        if init_int = '1' then
                                                state <= st_brun;
                                                init_int <= '0';
                                        else
                                                state <= st_idle;
                                        end if;
                                when st_brun =>                 --bit de arranque--
                                        rs232_dte_txd <='0';
                                        state <= st_bit0;
                                when st_bit0 =>
                                        rs232_dte_txd <=data(0);
                                        state <= st_bit1;
                                when st_bit1 =>
                                        rs232_dte_txd <=data(1);
                                        state <= st_bit2;
                                when st_bit2 =>
                                        rs232_dte_txd <=data(2);
                                        state <= st_bit3;
                                when st_bit3 =>
                                        rs232_dte_txd <=data(3);
                                        state <= st_bit4;
                                when st_bit4 =>
                                        rs232_dte_txd <=data(4);
                                        state <= st_bit5;
                                when st_bit5 =>
                                        rs232_dte_txd <=data(5);
                                        state <= st_bit6;
                                when st_bit6 =>
                                        rs232_dte_txd <=data(6);
                                        state <= st_bit7;
                                when st_bit7 =>
                                        rs232_dte_txd <=data(7);
                                        state <= st_bstop;
                                when st_bstop =>                --bit de paro--
                                        rs232_dte_txd <='1';
                                        state <= st_idle;
                                when others =>
                                        state <= st_idle;
                        end case;
                end if;
        end if;
end process;
process(clk)
begin
        if rising_edge(clk) then
                serial_ready_flag<='0';
                serial_ready<='0';
                        if state = st_bstop then
                                serial_ready_flag<='1';
                        end if;
                        if state = st_idle and serial_ready_flag='1' then
                                serial_ready_flag<='0';
                                serial_ready<='1';
                        end if;
        end if;
end process;
end Behavioral;
```

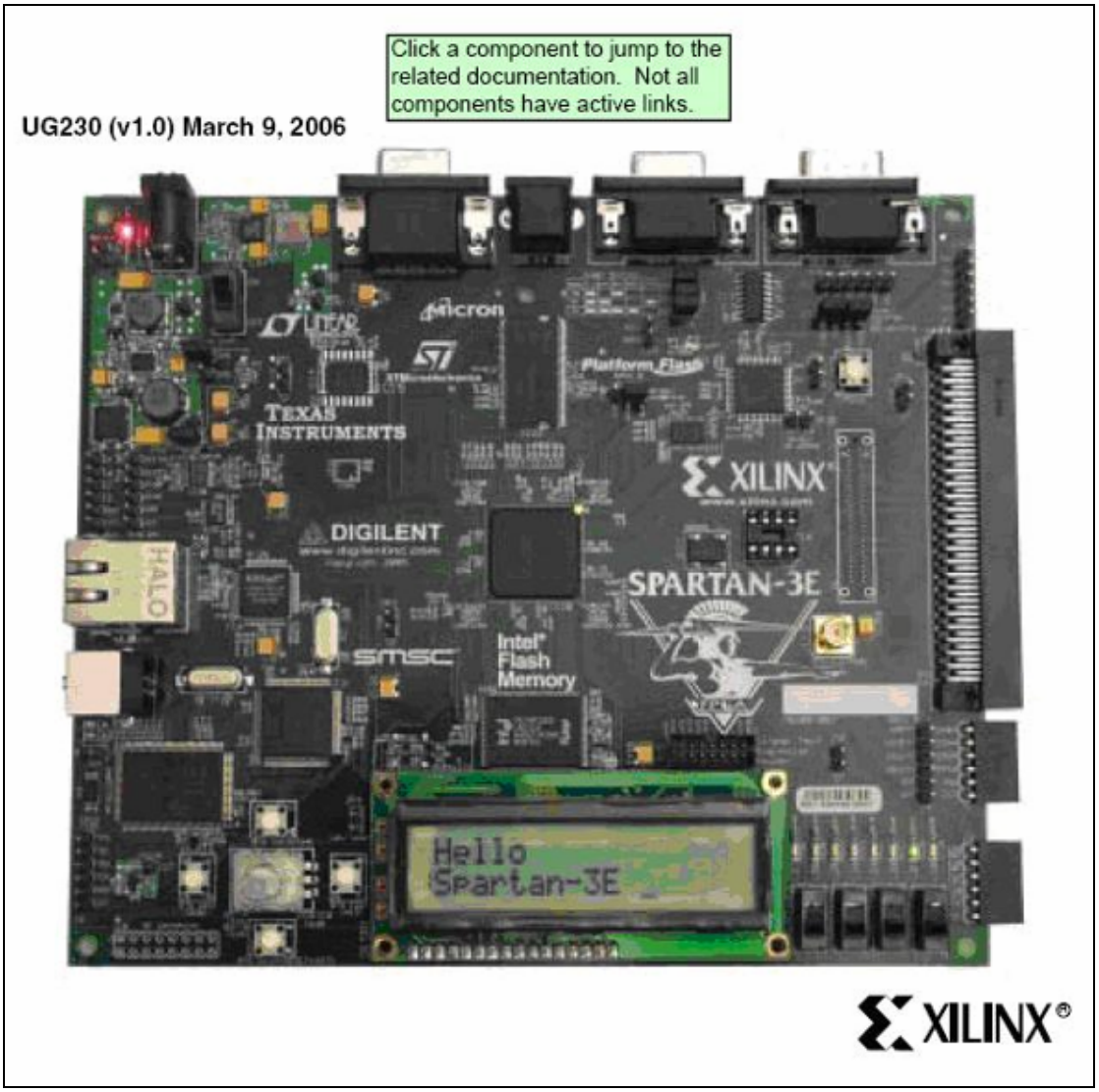
ANEXO I. CÓDIGO EN MATLAB PARA ADQUIRIR DATOS POR PUERTO SERIAL DE OSCILOSCOPIO PROMAX OD 581.

```
s=serial('COM1');
fopen(s);
DATOS_NUM=50;
X=[ ];
for i=0:DATOS_NUM
fprintf(s,':MEAS:SOUR{1}')
fprintf(s,':MEAS:VMAX?')
ch_1 = fscanf(s,'%g')
X=[X;[ch_1]];
End
Close(s)
Delete(s)
Clear s
```

ANEXO J. CÓDIGO EN MATLAB PARA ADQUIRIR DATOS DE LA SPARTAN 3E PARA GRAFICAR LA SEÑAL INFRARROJA.

```
s=serial('COM1')
fopen(s)
fprintf(s,'*IDN?')
out=fscanf(s)
k=double(out);
m=(5/255)*k;
plot(m);
title('MUESTREO DE SEÑAL');
xlabel('Muestras');
ylabel('Voltaje [V]');
grid on
fclose(s)
delete(s)
clear s
```

ANEXO K. ARREGLOS DE LÓGICA PROGRAMABLE – FPGAS- SPARTAN 3E.



K.1. DESCRIPCIÓN.

Un módulo de arreglos de lógica programable FPGA está compuesto por un determinado número de casillas iguales dispuestas de forma simétrica. Cada casilla posee memorias RAM y basculadores (flips-flops) que se pueden programar para ejecutar circuitos combinacionales y secuénciales. Las casillas se pueden conectar entre sí por medio de conexiones que también son programadas. La programación del arreglo lógico FPGA se hace mediante una comunicación serie llamada bitstream, que puede estar almacenada en una memoria externa como: PROM, EEPROM, RAM o provenir de otro sistema sea un PC, microcontrolador u otra FPGA.

El objetivo del arreglo lógico es dado por el diseñador para los circuitos digitales que posee dicho sistema. En la actualidad una tarjeta de arreglo lógico posee del orden de millones de compuertas como capacidad lógica y casillas dotadas con memorias procesadoras, así pues, un sistema podría confinarse a una tarjeta de arreglos lógicos programables.

La versatilidad de este tipo de tecnología se hace ver debido a su capacidad y al menú de elementos existentes en el integrado contenido en la tarjeta pues cuando se diseña un sistema este se lleva a cabo en tiempo record; para dar holgura en cuanto requerimientos y mejoras de proyectos sean de producción e investigación.

El uso de la acción en tiempo real implica el manejo del paralelismo de funciones en la tarjeta de arreglos de lógica programable para poner el formato de los datos, la sincronización, y la lógica especializada de la adición en ejecución necesitada para conectar los periféricos como los módems, receptores digitales de los convertidores analógico/digital a los procesadores programables. Consiste en una matriz de dos dimensiones de casillas programables que se enlazan por interconexiones. Estas interconexiones incluyen segmentos de pista de diferentes distancias, más unos conmutadores programables para enlazar las casillas a pistas o pistas mismas. La configuración de una tarjeta FPGA es principalmente elaborar los enlaces entre las casillas y configurar las casillas internamente.

Para configurar una tarjeta FPGA se hace por medio de Lenguajes de Descripción Hardware (HDL) como VHDL (Very high speed integrated circuit Hardware Description Language) o Verilog HDL. Con el uso de la tecnología se implementa una técnica de utilización de diseños que consiste en tomar los ya exisistentes como núcleos (cores) y adaptarlos a proyectos de estructuración en pro de la mejora y minimización de tiempo para los diseñadores.

K.2. SPARTAN 3E - XC3S500E.

La FPGA utilizada en el proyecto es la XC3S500E de la familia Spartan-3E, fabricada por Xilinx. A continuación se mencionan sus principales componentes.

- Xilinx XC3S500E Spartan-3E FPGA
  - Hasta 232 pines de propósito general (I/O)
  - Empaque de 320-pin FBGA
  - Alrededor de 10,000 celdas lógicas
- PROM de 4 Mbit Flash con plataforma de configuracion en Xilinx
- 64-macroceldas Xilinx XC2C64A CoolRunner CPLD
- 64 MByte (512 Mbit) de DDR SDRAM, x16 con interface de datos de, 100+ MHz
- 16 MByte (128 Mbit) de paralelas NOR Flash (Intel StrataFlash)
  - FPGA configuración de almacenamiento

- MicroBlaze code storage/shadowing

❖ 16 Mbits de SPI serial Flash (STMicro)

- FPGA configuracion de almacenamiento
- MicroBlaze code shadowing

❖ Pantalla LCD de 2-líneas, 16-caractereres
❖ Puerto PS/2 para mouse o keyboard
❖ Puerto para display VGA
❖ 10/100 Ethernet PHY (requiere Ethernet MAC en la FPGA)
❖ Dos puertos de 9-pin RS-232 (DTE- y DCE-style)
❖ On-board USB-based FPGA/CPLD download/debug interface
❖ Reloj oscilador 50 MHz
❖ SHA-1 1-cable serial EEPROM para proteccion de copia bitstream
❖ Hirose FX2 conector de expansión
❖ Tres Digilent 6-pin conectores de expansión
❖ Cuatro Conversores de salida, SPI-based Digital-a-Análogo (DAC)
❖ Dos conversores de entrada, SPI-based Análogo-a-Digital (ADC) con pre-amplificador de ganancia programable
❖ Puerto ChipScope™ SoftTouch debugging
❖ Encoder de Rotación con push-button shaft
❖ Ocho LEDs discretos
❖ Cuatro switches slide
❖ Cuatro switches push-button
❖ reloj de entrada SMA
❖ 8-pin DIP socket para reloj oscilador auxiliar

K.3. LENGUAJE DE DESCRIPCIÓN VHDL.

VHDL es un lenguaje de descripción de circuitos electrónicos digitales que utiliza distintos niveles de abstracción. El significado de las siglas VHDL es *VHSIC (Very High Speed Integrated Circuits) Hardware Description Language*. Esto significa que VHDL permite acelerar el proceso de diseño.

VHDL no es un lenguaje de programación, por ello conocer su sintaxis no implica necesariamente saber diseñar con él. VHDL es un lenguaje de descripción de hardware genérico, que permite describir circuitos síncronos y asíncronos.

Los circuitos descritos en VHDL pueden ser simulados utilizando herramientas de simulación para reproducir el funcionamiento del circuito. Además utilizando herramientas de síntesis se puede implementar dicho circuito en un dispositivo lógico programable o en un circuito integrado.

K.3.1. ELEMENTOS BÁSICOS DE VHDL.

VHDL es un lenguaje y como tal, posee sus tipos de datos y operadores. Los datos se almacenan en objetos que contienen valores de un tipo dado.

K.3.1.1 IDENTIFICADORES.

- Constante. Los objetos de esta clase tienen un valor inicial que es asignado de forma previa a la simulación y que no puede ser modificado durante ésta.
- Variable. Los objetos de esta clase contienen un único valor que puede ser cambiado durante la simulación con una sentencia de asignación. Las variables generalmente se utilizan como índices, principalmente en instrucciones de bucle, o para tomar valores que permitan modelar componentes. Las variables NO representan conexiones o estados de memoria.
- Signal. Los objetos de esta clase contienen una lista de valores que incluye el valor actual y un conjunto de valores futuros. Las señales representan elementos de memoria o conexiones y si pueden ser sintetizadas. Los puertos de una entidad son implícitamente declarados como señales en el momento de la declaración, ya que estos representan conexiones. También pueden ser declaradas en la arquitectura antes del BEGIN, lo cual permite utilizar tipos predefinidos, así como otros definidos por el usuario.

K.3.1.2. TIPOS PREDEFINIDOS.

- STD_LOGIC Tipo predefinido en el estándar IEEE 1164. Este tipo representa una lógica multivaluada de 9 valores. Además del ‘0’ lógico y el ‘1’ lógico, posee alta Impedancia ‘Z’, desconocido ‘X’ ó sin inicializar ‘U’ entre otros. Igual que se permite crear un vector de bits se puede crear un vector de std_logic, STD_LOGIC_VECTOR.

Para poder utilizar el tipo std_logic hay que añadir la librería que lo soporta.
Para poder utilizar el tipo:
use ieee.std_logic_1164.all.

Para poder utilizar las funciones aritméticas lógicas definidas (suma, resta, multiplicación)
use ieee.std_logic_arith.all.

Si los vectores están en representación binaria pura
use ieee.std_logic_unsigned.all.

Los vectores están en C2
use ieee.std_logic_unsigned.all.

K.3.1.3. TIPO ENUMERADO.

Es un tipo de dato con un grupo de posibles valores asignados por el usuario. Los tipos enumerados se utilizan principalmente en el diseño de máquinas de estados.

Los tipos enumerados se ordenan de acuerdo a sus valores. Los programas de síntesis automáticamente codifican binariamente los valores del tipo enumerado para que estos puedan ser sintetizados. Algunos programas lo hacen mediante una secuencia binaria ascendente, otros buscan cual es la codificación que mejor conviene para tratar de minimizar el circuito o para incrementar la velocidad del mismo una vez que la descripción ha sido sintetizada. También es posible asignar el tipo de codificación mediante directivas propias de la herramienta de síntesis.

K.3.1.4. TIPOS COMPUESTOS.

Un tipo compuesto es un tipo de dato formado con elementos de otros tipos, existen dos formas de tipos compuestos, arrays y records.

Un ARRAY es un objeto de datos que consiste en una “colección” de elementos del mismo tipo.

Un RECORD es un objeto de datos que consiste en una “colección” de elementos de distintos tipos.

K.3.2. OPERADORES.

Un operador nos permite construir diferentes tipos de expresiones mediante los cuales podemos calcular datos utilizando diferentes objetos de datos con el tipo de dato que maneja dicho objeto. En VHDL existen distintos operadores de asignación con lo que se transfieren valores de un objeto de datos a otro, y operadores de asociación que relacionan un objeto de datos con otro, lo cual no existe en ningún lenguaje de programación de alto nivel.

abs

*, /, mod, rem

+ (sig.), - (sig)

+, -, &

and, or, nand, nor, xor

:= asignación de valores a constantes y variables.

<= asignación de valores a señales.

K.3.3 ESTRUCTURA BÁSICA DE UN ARCHIVO FUENTE EN VHDL.

El archivo VHDL contiene la descripción del circuito que se quiere implementar.

K.3.3.1. ENTIDAD (Entity):

Una entidad es la abstracción de un circuito, ya sea desde un complejo sistema electrónico hasta una simple compuerta lógica. La entidad únicamente describe la forma externa del circuito, aquí se enumeran las entradas y las salidas del diseño. Una entidad es análoga a un símbolo esquemático de los diagramas electrónicos, el cual describe las conexiones del dispositivo hacia el resto del diseño.

ENTIDAD (ENTITY)

- Define externamente al circuito o subcircuito.
- Nombre y número de puertos, tipos de datos de entrada y salida.
- Tienes toda la información necesaria para conectar tu circuito a otros circuitos.

Los puertos pueden ser de entrada in, salida out, entrada-salida inout o un buffer; Además, la entidad puede definir un valor genérico (GENERIC) que se utilizará para declarar las propiedades y constantes del circuito, independientemente de cual sea la arquitectura.

K.3.3.2 ARQUITECTURA (Architecture)

Los pares de entidades y arquitecturas se utilizan para representar la descripción completa de un diseño. Una arquitectura describe el funcionamiento de la entidad a la que hace referencia. Si una entidad la asociamos con una “caja” en la que se enumeran las interfaces de conexión hacia el exterior, entonces la arquitectura representa la estructura interna de esa caja.

ARQUITECTURA (ARCHITECTURE)

- Define internamente el circuito.
- Señales internas, funciones, procedimientos, constantes.
- La descripción de la arquitectura puede ser estructural o por comportamiento.

Una arquitectura se describe por comportamiento o por estructura. Una entidad puede tener más de una arquitectura, pero cuando se compile se indica cual es la arquitectura que se quiere utilizar.

K.3.3.3 PROCESO (PROCESS)

Cuando en VHDL se escribe un process, dentro de él aparece la parte secuencial del circuito. La simulación no entra en el process hasta que no haya variado alguna de las señales o variables de su lista de sensibilidad independientemente de lo que este contenido dentro del

process. Por otro lado únicamente dentro de un process pueden aparecer las sentencias de tipo if y else y nunca puede aparecer una sentencia del tipo wait.

K.3.3.4 DESCRIPCIONES DE ARQUITECTURA

K.3.3.4.1 ESTRUCTURAL

Esta descripción utiliza entidades descritas y compiladas previamente. Se declaran los componentes que se utilizan y después, mediante los nombres de los nodos, se realizan las conexiones entre las compuertas. Las descripciones estructurales son útiles cuando se trata de diseños jerárquicos.

Una descripción estructural:

- Describe las interconexiones entre distintos módulos.
- Estos módulos pueden a su vez tener un modelo estructural o de comportamiento.
- Normalmente esta es una descripción a más alto nivel.

K.3.3.4.2. COMPORTAMIENTO (behavioral).

Las descripciones comportamentales son similares a un lenguaje de programación de alto nivel por su alto nivel de abstracción. Más que especificar la estructura o la forma en que se deben conectar los componentes de un diseño, nos limitamos a describir su comportamiento. Una descripción comportamental consta de una serie de instrucciones, que ejecutadas modelan el comportamiento del circuito. La ventaja de este tipo de descripción es que no necesitamos enfocarnos a un nivel de compuerta para implementar un diseño.

Los módulos que se suelen describir mediante comportamiento suelen ser:

- Módulos que están al final de la jerarquía en una descripción estructural.
- Paquetes CI de una librería.
- Módulos cuyo comportamiento es complicado para poderlo describir mediante una estructura.
- Process con wait, if, else, when, case ...
- Muchas veces las funciones dependen del tiempo, VHDL resuelve este problema permitiendo la descripción del comportamiento en la forma de un programa tradicional.

Una vez terminado el sistema que describe su funcionamiento se debe finalizar las sentencias si se utilizaron, los procesos, y la arquitectura.

K.4. IMPLEMENTACIÓN.

Proceso de Implementación:

- Definición del diseño inicial (vhdl, esquemas).
- Optimización lógica (de las ecuaciones).
- Mapeo tecnológico (adapta ecuaciones a tecnología).

- Ubicación (asigna la ubicación física a las ecuaciones).
- Ruteo (conecta todas las ecuaciones).
- Programación de la unidad (genera configuración).
- Configuración. Carga en serie para minimizar el número de pines. Conexión en cadena para la configuración de varias FPGAs.
- Relectura. Se permite la lectura del contenido completo del chip, incluido el contenido de los FF internos.
- Seguridad. Bit para evitar la re-ingeniería del diseño