



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

FACULTAD DE INGENIERÍAS

CARRERA DE INGENIERÍA ELECTRÓNICA

TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE:

INGENIERO ELECTRÓNICO

TEMA:

“DISEÑO E IMPLEMENTACIÓN DE UN ROBOT AUTÓNOMO
MÓVIL USANDO TECNOLOGÍA FPGA”

AUTORES:

ROBERTO FERNANDO RAMÍREZ RIVERA

ROBERTO CARLO REYES PÉREZ

DIRECTOR:

ING. BYRON LIMA CEDILLO

GUAYAQUIL, ABRIL 2015

DECLARATORIA DE RESPONSABILIDAD

Nosotros, Roberto Fernando Ramírez Rivera y Roberto Carlo Reyes Pérez alumnos de la Universidad Politécnica Salesiana, Facultad de Ingenierías, carrera de Ingeniería Electrónica mención Sistemas Industriales, libre y voluntariamente declaramos que la presente tesis ha sido desarrollada en su totalidad por nosotros, por tal razón asumimos la responsabilidad por su autoría.

Guayaquil, Abril del 2015

Roberto Fernando Ramírez Rivera

Roberto Carlo Reyes Pérez

DEDICATORIA

La presente tesis la dedico con todo el amor del mundo a mi madre Flora Rivera Orosco, quien gracias a su esfuerzo y motivación he podido salir adelante con este trabajo.

A mi linda esposa Dexy Álvarez y a mi hijita bella Camilita Fernanda, para uds quienes son la luz de mi vida y a quienes amo con toda mi alma, les dedico mi esfuerzo y mis ganas por seguir siempre para adelante.

Roberto Fernando Ramírez Rivera

La presente tesis, se la dedico a Dios y a mis padres, que son los responsables de la persona que soy, y que con sus con su ejemplo y consejos me han ayudado a tener temple para seguir adelante en todos los retos que he tenido en mi vida.

A demás, quiero dedicar esta tesis a mis hermanas, que han sido un apoyo emocional y me dieron el ánimo de continuar cada día con mi proyecto y que a su vez me dieron su ayuda en todo lo que pudieron para poder culminar esta tesis.

Roberto Carlo Reyes Pérez

AGRADECIMIENTO

Agradezco de todo corazón a todas esas personas que estuvieron conmigo en esta etapa y fueron pieza fundamental para la culminación de esta tesis, quiero agradecer a Dios por darme las fuerzas día a día para seguir adelante, a las bellas mujeres de vida, mi hermosa Madre Flora Rivera, a mi bella Esposa Dexy Álvarez, a mi abuela Elvia Orosco y a mi hermano Luis Ramírez, gracias por su apoyo y paciencia.

Un enorme agradecimiento a mi compañero de tesis Roberto Reyes y al Ing. Byron Lima, tutor y guía, para la culminación de este trabajo.

Roberto Fernando Ramírez Rivera

En primer lugar agradezco mucho a Dios, por las oportunidades que me ha dado en esta vida y por las lecciones que me han hecho crecer a lo largo del camino y por ayudarme a cumplir con esta tesis.

A mis padres, por apoyarme y confiar en mí en todo momento, también porque gracias a su ayuda dediqué el tiempo para realizar este trabajo.

A mis profesores, ya que gracias a su paciencia y enseñanza que ha sido un pilar para el desarrollo de esta tesis.

A mis amigos, que con su amistad, compañerismo y palabras de aliento han sido el apoyo moral que necesité para continuar con constancia para la culminación de este trabajo.

Roberto Carlo Reyes Pérez

ÍNDICE GENERAL

CAPÍTULO 1. EL PROBLEMA	3
1.1 PLANTEAMIENTO DEL PROBLEMA	3
1.2 DELIMITACIÓN	3
1.3 OBJETIVO GENERAL.....	4
1.4 OBJETIVOS ESPECÍFICOS.....	4
1.5 JUSTIFICACIÓN DEL TRABAJO	5
1.6 VARIABLES E INDICADORES.....	5
1.7 MARCO METODOLÓGICO.....	5
1.8 POBLACIÓN Y MUESTRA.....	6
1.9 DESCRIPCIÓN DE LA PROPUESTA.....	7
1.10 BENEFICIARIOS	7
1.11 IMPACTO	7
CAPÍTULO 2. MARCO TEÓRICO	8
2.1 ANTECEDENTES HISTÓRICOS.....	8
2.2 LA ROBÓTICA.....	9
2.3 LOS ROBOTS MÓVILES	9
2.4 ESTRUCTURA GENERAL DE UN ROBOT MÓVIL.....	11
2.5 GRADOS DE LIBERTAD DE UN ROBOT MÓVIL	11
2.6 CONFIGURACIÓN DIFERENCIAL DE UN ROBOT MÓVIL	12
2.7 CENTRO INSTANTÁNEO DE CURVATURA (ICC).....	13
2.8 SISTEMAS EMBEBIDOS EN LOS ROBOTS MÓVILES.....	13
2.9 SISTEMAS DE TIEMPO REAL (STR).....	14
2.10 FPGA (ARREGLOS DE COMPUERTAS PROGRAMABLES EN CAMPO).....	15
2.10.1 Cinco beneficios Principales de la Tecnología FPGA	15
2.11 CONFIGURACIÓN DE LA FPGA.....	17

2.12 SISTEMAS DE CONTROL Y ADQUISICIÓN DE DATOS DE NATIONAL INSTRUMENTS.....	17
2.13 INTRODUCCIÓN A LA PLATAFORMA DE PROGRAMACIÓN GRÁFICA LABVIEW	18
2.14 DESCRIPCIÓN DE LA PLATAFORMA GRÁFICA DE PROGRAMACIÓN LABVIEW	21
2.14.1 MEASUREMENT & AUTOMATION EXPLORER (MAX)	21
2.14.2 ENTORNO GRÁFICO	22
2.15 ADQUISICIÓN DE DATOS.....	23
2.15.1 Tarjeta FPGA NI Single Board-9631.....	24
2.15.2 Componentes principales de la tarjeta SbRIO-9631.....	25
2.15.2.1 Puerto Ethernet RJ-45.....	26
2.15.2.2 Puerto Serial RS-232.....	26
2.15.2.3 Interruptores DIP.....	27
2.15.2.4 Botón RESET.....	28
2.15.2.5 Indicadores Leds.....	28
2.15.2.6 Conector de Alimentación.....	29
2.15.2.7 Conectores de Puertos de E/S Digitales.....	29
2.15.2.8 Conector de E/S Analógicos	32
2.16 APLICACIONES DE CONTROL EN TIEMPO REAL.....	32
2.16.1 Conceptos Generales.....	33
2.16.1.1 Capacidad de Respuesta.....	33
2.16.1.2 Determinismo y Fluctuación	33
2.16.1.3 Prioridad.....	34
2.17 ARQUITECTURA Y FUNCIONAMIENTO DE UN SISTEMA LABVIEW FPGA DE TIEMPO REAL.....	34
2.18 MÓDULOS LABVIEW PARA EL DESARROLLO DE APLICACIONES ...	36
2.18.1 Módulo LabVIEW FPGA.....	36

2.18.2 Módulo LabVIEW Real Time.....	39
2.19 CREACIÓN DE UN PROYECTO	39
2.19.1Proceso y Desarrollo de Compilación de un VI FPGA.....	43
2.19.2 Inicio de la Compilación de un VI FPGA	44
2.19.3Ventana generada durante el Estado de la Compilación.....	44
2.20 CREACIÓN DE UN ARCHIVO EJECUTABLE	45
2.20.1 Especificaciones de Construcción.....	46
2.19.2 Ejecución del Archivo Ejecutable	47
CAPÍTULO 3. DISEÑO Y CONSTRUCCIÓN DEL ROBOT MÓVIL.....	49
3.1 DISEÑO DE HARDWARE	49
3.1.1 REQUERIMIENTOS DE LA ESTRUCTURA DEL ROBOT MÓVIL.....	49
3.1.2 Diseño de la Estructura del Robot Móvil	50
3.2 DESCRIPCIÓN DE COMPONENTES QUE CONFORMAN EL ROBOT MÓVIL.....	52
3.3 INSTALACIÓN Y MONTAJE DE LOS COMPONENTES.....	53
3.3.1 Instalación de los motores DC y las ruedas.....	56
3.3.2 Instalación de los Encoders.....	57
3.3.3 Instalación de Tarjetas Drivers.....	57
3.3.4 Instalación de Tarjeta NI sbRIO-9631.....	58
3.3.5 Instalación de los Sensores de Ultrasonido	60
3.3.6 Instalación de Tarjeta de Adquisición de Datos	61
3.4 IMÁGENES DEL ROBOT YA ENSAMBLADO	62
3.5 PRUEBAS DE FUNCIONAMIENTO DEL ROBOT MÓVIL.....	63
3.5.1 Primera prueba. Medición de Continuidad entre líneas que suministran Voltaje	63
3.5.2 Segunda prueba. Verificación de los fusibles.....	63
3.5.3 Tercera prueba. Suministro y Medición de Voltajes en distintos puntos del Robot Móvil	64

3.5.4 Cuarta prueba. Verificación de señal WIFI	64
3.5.5 Quinta prueba. Comunicación de la Red	64
CAPÍTULO IV. DESAROLLO DE GUÍAS DE LABORATORIO	66
4.1 DESCRIPCIÓN DEL CAPÍTULO	67
4.2 GUÍA DE LABORATORIO 1	67
4.3 GUÍA DE LABORATORIO 2	76
4.4 GUÍA DE LABORATORIO 3	94
4.5 GUÍA DE LABORATORIO 4	108
4.6 GUÍA DE LABORATORIO 5	115
4.7 GUÍA DE LABORATORIO 6	124
4.8 GUÍA DE LABORATORIO 7	133
4.9 GUÍA DE LABORATORIO 8	140
CONCLUSIONES	153
RECOMENDACIONES	154
CRONOGRAMA	155
PRESUPUESTO	156
REFERENCIAS	157
ANEXOS	161

ÍNDICE DE CUADROS Y/O TABLAS

Tabla 1: Dispositivos principales del Robot Móvil.....	51
Tabla 2: Distribución de pines del display LCD 16x2.....	80

ÍNDICE DE GRÁFICOS

Figura 2.1 La robótica en el mundo entero	9
Figura 2.2 Robot Móvil Starter Kit LabView Robotics	10
Figura 2.3 Similitudes entre un robot y un ser vivo	12
Figura 2.4 Robot móvil con estructura diferencial	13
Figura 2.5 ICC en a) Configuración diferencial b) Configuración ackerman c) Configuración triciclo	13
Figura 2.6 Plataforma de desarrollo gráfico LabVIEW	19
Figura 2.7 Measurement & Automation Explorer (MAX).....	21
Figura 2.8 Panel frontal de LabVIEW	22
Figura 2.9 Diagrama de Bloques en LabVIEW	23
Figura 2.10 Diagrama de conexión de la tarjeta NI sbRIO-9631.....	24
Figura 2.11 Tarjeta NI SbRIO-9631 de National Instruments	25
Figura 2.12 Puerto Serial RS-232.....	26
Figura 2.13 Interruptores DIP	27
Figura 2.14 Leds Indicadores del dispositivo SbRIO-9631.....	28
Figura 2.15 Conector de Alimentación al dispositivo SbRIO	29
Figura 2.16 Conector P2 de 3.3V de E/S Digitales	30
Figura 2.17 Conector P3 de 3.3V de E/S Digitales	30
Figura 2.18 Conector P4 de 3.3V de E/S Digitales	31
Figura 2.19 Conector P5 de 3.3V de E/S Digitales	31
Figura 2.20 Conector J7 de E/S Analógicas	32
Figura 2.21 Arquitectura del Sistema LabVIEW FPGA.	34

Figura 2.22 Arquitectura de un Sistema LabVIEW FPGA Real Time	35
Figura 2.23 Funciones de programación LabVIEW	37
Figura 2.24 Paleta FPGA Interface	37
Figura 2.25 Ventana Configure Open FPGA VI Reference	38
Figura 2.26 Paleta Real Time. Paleta de funciones que maneja el módulo Real Time	39
Figura 2.27 Ventana Getting Started de LabVIEW	40
Figura 2.28 Ventana Project Explorer de LabVIEW	40
Figura 2.29 Ventana de Creación de un proyecto a partir de FPGA Project	41
Figura 2.30 Ventana de configuración de Single-Board RIO FPGA Project	42
Figura 2.31 Apariencia final del proyecto creado a través de FPGA Project	43
Figura 2.32 Proceso de compilación de un VI FPGA	44
Figura 2.33 Ventana de Estado de Compilación	45
Figura 2.34 Cuadro de diálogo My Real-time Application Properties	46
Figura 3.1 Dimensiones de la Estructura del Robot Móvil	51
Figura 3.2 Diagrama en bloques del Robot Móvil	54
Figura 3.3 Batería LIPO instalada en el Robot Móvil.....	55
Figura 3.4 Motores DC junto con las ruedas	56
Figura 3.5 Rueda LOCA del Robot Móvil... ..	57
Figura 3.6 Acople del Encoder con el motor DC	57
Figura 3.7 Vista superior Tarjeta driver para los motores DC	58
Figura 3.8 Vista superior de la tarjeta driver para los encoders	59
Figura 3.9 Vista superior del robot móvil apreciando la tarjeta SbRIO-9631.....	59

Figura 3.10 Vista de los Sensores de Ultrasonido instalados en el robot móvil.	60
Figura 3.11 Sensores de Ultrasonido instalados en el Robot Móvil.	61
Figura 3.12 Tarjeta de adquisición de datos	61
Figura 3.13 Parte lateral del robot móvil.....	62
Figura 3.14 Parte lateral del Robot Móvil	62
Figura 3.15 Trabajando con el Robot Móvil	63
Figura 3.16 Pruebas de medición de voltaje en distintos puntos del Robot Móvil	64
Figura 3.17 Señal WIFI “PezPequeno” de la red inalámbrica	65
Figura 4.1 Tarjeta NI sbRIO-9631.Vista superior de la Tarjeta FPGA NI Single-Board de la familia RIO.....	68
Figura 4.2 Ventana de configuración de la IP del Robot Móvil.....	69
Figura 4.3 Ventana de propiedades de Protocolo de Internet IPv4.....	70
Figura 4.4 Comunicación inalámbrica.....	70
Figura 4.5 Ventana de acceso para configuración del router inalámbrico.....	71
Figura 4.6 Ventana de acceso para configuración del router inalámbrico.....	72
Figura 4.7 Ventana Getting Started.	73
Figura 4.8 Ventana Project Explorer.....	73
Figura 4.9 VI Panel Frontal Control de luces LED.....	74
Figura 4.10 VI Digrama de Bloques Control de Luces LED	74
Figura 4.11 Vista superior de la tarjeta NI sbRIO-9631.....	75
Figura 4.12 Sensor de humedad HIH4030.....	77
Figura 4.13 Sensor de temperatura LM35.....	78
Figura 4.14 Sensor de luz con fotorresistencia	78

Figura 4.15 Sensor de Ultrasonido HC-SR04.....	79
Figura 4.16 Display LCD de 16x2 y su conexión	80
Figura 4.17 Proyecto Prueba LCD	81
Figura 4.18 Diagrama de Bloques de la adquisición de datos	82
Figura 4.19 Diagrama de Bloque del VI para la adquisición de datos del sensor ultrasónico	82
Figura 4.20 Diagrama de Bloques para la inicialización del sensor de ultrasonido	83
Figura 4.21 Conexión del LCD con la tarjeta sbRIO-9631	84
Figura 4.22 Diagrama de flujo para la inicialización del LCD	85
Figura 4.23 Diagrama de Bloque del VI para la inicialización del display LCD	86
Figura 4.24 Diagrama de Flujo para la configuración del LCD	87
Figura 4.25 Diagrama de Bloques para la configuración del LCD	88
Figura 4.26 Diagrama de bloques subVI Conversión de valores decimales a binarios	88
Figura 4.27 Diagrama de bloques subVI Adquisición de datos binarios y habilitación del LCD	89
Figura 4.28 Diagramas de Bloques del VI_Writer	91
Figura 4.29 Panel Frontal del VI Real Time	91
Figura 4.30 Visualización de datos en el LCD	92
Figura 4.31 Diagrama de bloques del VI Real Time	93
Figura 4.32 Módulo Xbee Series 1.....	96
Figura 4.33 Mando de control Joystick Shield	96

Figura 4.34 Distribución de pines PIC16F877A	97
Figura 4.35 Distribución de pines del módulo Xbee y conexión básica	98
Figura 4.36 Xbee Explorer	99
Figura 4.37 Envío y recepción de datos con Xbee y microcontrolador	99
Figura 4.38 Proyecto Control Inalámbrico	104
Figura 4.39 Panel Frontal del programa “Mando” para el Control Inalámbrico	105
Figura 4.40 Diagrama en bloques del programa “Mando”	105
Figura 4.41 Panel Frontal del Programa “Mando RT”	106
Figura 4.42 Mando de Control transmisor de datos	106
Figura 4.43 Diagrama en bloques del programa Mando RT	107
Figura 4.44 Receptor de datos Xbee en la tarjeta de Adquisición de datos	107
Figura 4.45 Encoder incremental Autonics E50S	110
Figura 4.46 Ventana del Proyecto “Patrullar”	110
Figura 4.47 Panel Frontal del VI “Patrulla FPGA”	111
Figura 4.48 Diagrama de Control del VI “Patrullar”	112
Figura 4.49 Panel Frontal del VI “Patrullar 2RT”	113
Figura 4.50 Diagrama de control del VI Real Time “Patrullar RT”	114
Figura 4.51 Robot Móvil trabajando como Patrullero	115
Figura 4.52 Ubicación de sensores de ultrasonido	116
Figura 4.53 Nuevo Proyecto Anti-caídas para el desarrollo de la práctica	118
Figura 4.54 VIs para la inicialización del sensor de ultrasonido HC-SR04	119
Figura 4.55 Panel Frontal del VI “Altura”	119

Figura 4.56 Diagrama de Control del VI “Altura”	120
Figura 4.57 Panel Fronal del VI “Anti-caídas RT”	121
Figura 4.58 Diagrama de control del VI “Anticaídas” con Case Structure “True”	121
Figura 4.59 Diagrama de control del VI “Anti-caídas” con Case Structure “False”	122
Figura 4.60 Imagen del sensor de ultrasonido ubicado en el Robot Móvil.....	123
Figura 4.61 Robot Móvil trabajando como Anti-caídas	124
Figura 4.62 Sensor Óptico QRD1114.....	125
Figura 4.63 Banco de sensores ópticos QRD114 de Idetec	125
Figura 4.64 Conexión de pines del banco de sensores ópticos QRD114 de Idetec	126
Figura 4.65 Proyecto “Seguidor de Línea”.....	126
Figura 4.66 Panel Frontal del VI “Seguidor” en LabVIEW FPGA	127
Figura 4.67 Diagrama de control del VI “Seguidor” en LabVIEW FPGA	127
Figura 4.68 Panel frontal del VI Real Time “Seguidor RT”	128
Figura 4.69 Diagrama de control del VI Real Time “Seguidor RT”.....	129
Figura 4.70 Diagrama de control del VI Real Time “Seguidor RT”	130
Figura 4.71 Robot Móvil trabajando en línea curva	130
Figura 4.72 Ubicación de sensores de ultrasonido en los bordes del robot móvil	134
Figura 4.73 Proyecto “Navegador”	134
Figura 4.74 Diagrama de Bloques para la inicialización del sensor ultrasónico	135

Figura 4.75 Panel Frontal del VI LabVIEW FPGA “Esquivador fpga”.	136
Figura 4.76 Diagrama de control del VI LabVIEW FPGA “Esquivador”.	136
Figura 4.77 Panel Frontal del VI Real Time “Esquivador RT”.	137
Figura 4.78 Diagrama de control del VI Real Time “Esquivador RT”.	137
Figura 4.79 Cámara de red AXIS M1011.	141
Figura 4.80 Ventana de configuración AXIS IP Utility	142
Figura 4.81 Ventana del Setup de la Cámara IP	142
Figura 4.82 Configuración de la Cámara AXIS a través del MAX.	144
Figura 4.83 Proyecto en LabVIEW “Reconocedor”.	144
Figura 4.84 Diagrama de Bloques para la inicialización del sensor ultrasónico.	145
Figura 4.85 Panel Frontal del VI “Captura fpga” en LabVIEW FPGA	146
Figura 4.86 Diagrama de control del VI “Captura fpga” de LaVIEW FPGA	146
Figura 4.87 Panel Frontal del VI “Captura CP” en LabVIEW PC.	147
Figura 4.88 Diagrama de control del VI “Captura CP”.	148
Figura 4.89 Panel frontal del VI “Captura RT” en LabVIEW Real Time	149
Figura 4.90 Diagrama de control del VI “Captura RT” en LabVIEW Real Time	150
Figura 4.91 Instalación de cámara de red AXIS y sensores de ultrasonido HC-SR04.	151
Figura 4.92 Visualización de la imagen a través de la Cámara AXIS	151

RESUMEN

AÑO	ALUMNOS	DIRECTOR DE TRABAJO DE TITULACIÓN	TEMA TRABAJO DE TITULACIÓN
2015	ROBERTO FERNANDO RAMÍREZ RIVERA ROBERTO CARLO REYES PÉREZ	ING. BYRON LIMA CEDILLO	“DISEÑO E IMPLEMENTACIÓN DE UN ROBOT AUTÓNOMO MÓVIL USANDO TECNOLOGÍA FPGA”

El presente trabajo “DISEÑO E IMPLEMENTACIÓN DE UN ROBOT AUTÓNOMO MÓVIL USANDO TECNOLOGÍA FPGA”, tiene como objetivo mostrar las bondades de la tecnología FPGA y servir como herramienta pedagógica para los alumnos de la Universidad Politécnica Salesiana.

Este trabajo está dividido en 8 prácticas que presentan diferentes rangos de dificultad, las cuales van desde la comunicación del robot con el software LabVIEW hasta el manejo de aplicaciones donde se utilizan diferentes tipos de sensores.

El objetivo de cada práctica puede ser abarcado de diferentes formas a través de la programación, por los estudiantes y sirve como parte del estudio de las FPGA's.

PALABRAS CLAVES

Robot Autónomo Móvil/LabVIEW/FPGA/Sensores/Programación

ABSTRACT

YEAR	STUDENTS	THESIS DIRECTOR	THESIS THOPIC
2015	ROBERTO FERNANDO RAMÍREZ RIVERA ROBERTO CARLO REYES PÉREZ	ING. BYRON LIMA CEDILLO	“DESIGN AND IMPLEMENTATION OF AN AUTONOMOUS MOBILE ROBOT TECHNOLOGY USING FPGA”

The present work "DESIGN AND IMPLEMENTATION OF A MOBILE ROBOT USING TECHNOLOGY SELF FPGA", aims to show the benefits of FPGA technology and serve as a teaching tool for students of the Salesian Polytechnic University.

This work is divided into 8 practices that have different ranges of difficulty, ranging from communication with LabVIEW software robot to managing applications where different types of sensors are used.

The objective of each practice can be covered in different ways through programming by students and serves as part of the study of FPGA's.

KEYWORDS

Autonomous Mobile Robot/LabVIEW/FPGA/Sensors/Programation

INTRODUCCIÓN

Como es de conocimiento general, la tecnología avanza día con día por lo que todo estudiante de Ingeniería deben actualizar sus conocimientos para poder hacer frente a los nuevos equipos e instrumentos que podrá encontrar a lo largo de su carrera profesional, entender los distintos lenguajes de programación y el manejo de diferentes sistemas de control es indispensable en el mundo de la ingeniería electrónica.

Este trabajo se ha centrado específicamente en el universo de la robótica ya que esta área es multifacética pues se aplican conocimientos mecánicos, eléctricos, electrónicos y programación de controladores, por lo que si se quiere aprender más y profundizar en estos temas, la realización de un robot es una oportunidad perfecta para crecer profesionalmente.

El club de robótica de la Universidad Politécnica Salesiana, intenta incentivar a sus estudiantes con actividades extracurriculares en el marco de competiciones de robots para que los alumnos encuentren la motivación necesaria para el desarrollo de la robótica y la electrónica en general. Por lo que este proyecto pretende ser un aporte más en el mejoramiento del área de robótica de la universidad y ayudar a sus estudiantes a conocer diferentes tecnologías.

Una de las principales características de este proyecto es la utilización de una tarjeta NI sbRIO-9631 de National Instruments la cual está basada en tecnología FPGA, de las cuales su uso se está viendo más extendido y necesitan más comprensión para un mayor manejo de las mismas.

El presente documento está dividido en cuatro capítulos con sus respectivos anexos que va desde los conceptos y conocimiento generales de los Robots Móviles, FPGA's hasta el diseño y la construcción del Robot Móvil con sus respectivas prácticas o guías de laboratorio. A continuación un breve detalle de cada capítulo del documento.

El Capítulo 1. Planteamiento del Problema, se presenta el planteamiento del problema, los objetivos y el impacto del proyecto en relación a lo eficiente y capaz que pueden ser todos los robots y su desempeño y contribución en la sociedad.

El Capítulo 2. Marco Teórico, detalla en forma general cual es el funcionamiento de un robot móvil, Sistemas Embebidos, el modo de trabajo de una FPGA, en este caso de una tarjeta SbRio y el entorno de su programación.

El Capítulo 3. Diseño y Construcción del Robot Móvil, podemos apreciar el diseño y ensamblaje de la estructura, diseño y ubicación de tarjetas electrónicas junto a las pruebas de funcionamiento, así como la ubicación de todos los sensores.

El Capítulo 4. Desarrollo de Guías de Laboratorio, es donde están todas las prácticas desarrolladas, las cuales sirven como ejercicios prácticos que deseen conocer más acerca del robot y sus capacidades.

Esta tesis es solo una base que tendrán los estudiantes de Ingeniería Electrónica para el desarrollo y mejora de muchos otros trabajos académicos que se basen en tecnología FPGA, ya que es necesario difundir con mayor amplitud este tipo de tecnología para poder realizar mejoras en la comunicación de sensores, nivel de respuesta de actuadores y generación de librerías y así poder usar de mejor manera de todas las funciones y ventajas que tienen estas tarjetas, las cuales poseen una increíble capacidad de cálculo y son la preferencia en cuanto a sistemas embebidos se refiere.

CAPÍTULO 1. EL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA

Debido a que la Universidad Politécnica Salesiana posee un área destinada a la elaboración de prototipos robóticos llamado “Club de Robótica” formado por un grupo de estudiantes interesados en incrementar sus conocimientos en esta rama y que reúne el conocimientos de varias materias como Electrónica, Electiva, Sensores, Microcontroladores, entre otros y por la participación que tienen la Universidad en varios torneos de robótica de nivel académico con distintas Universidades del país, se ha determinado que falta desarrollar aplicaciones robóticas con un tipo de tecnología diferente al de los micro-controladores, considerando que se obtienen buenos resultados, pero en muchos modelos implica tener una sola forma de diseñar un robot, para ello queremos demostrar que existen otros tipos de tecnologías que podrían tener mayores ventajas e incluso ser más eficientes en sus funciones que es la que utilizan actualmente los FPGA’s, PLD’s, entre otros. Por otra parte existen materias como Digitales y Electrónica que ya están incursionando en la enseñanza de estas nuevas tecnologías existente en la actualidad y de la cual hace falta material didáctico para que el estudiante pueda trabajar y adquirir un conocimiento más práctico de lo que se está enseñando, incluso seminarios que se ofrecen necesitan de más recursos para ser más demostrativas sus clases, por ello el desarrollo de este “Robot Autónomo Móvil Basado en FPGA” será una herramienta útil para los estudiantes de Ingeniería.

1.2 DELIMITACIÓN

Temática: La presente propuesta busca dotar al departamento del Club de Robótica de un robot diseñado a partir de las nuevas tecnologías existentes y ser de ayuda para los análisis demostrativos de las materias de Electrónica, Electiva, Sensores y Microcontroladores, entre otras, de la carrera de Ingeniería Electrónica Industrial.

Técnica: En el proyecto se utilizará una tarjeta FPGA sb-RIO 9631 de National Instruments reconfigurable, con un procesador de 128 MHz, 64 MB DRAM, 256 de almacenamiento no volátil, puerto serie RS232 para dispositivos periféricos y 110 E/S digitales de 3.3 v (compatibles con tecnología TTL), con respecto a la

alimentación del robot se utilizará una fuente de voltaje de +12-30VDC, un circuito amplificador de voltaje de +3.3VDC a +12VDC y tarjetas de comunicación inalámbrica XBEE con alcance de 100m en espacios abiertos y 30m en interiores, así como una cámara IP para visualizar el entorno del robot y la identificación de objetos de varios colores además del uso de sensores de ultrasonidos.

En el circuito de comando se utilizará un micro-controlador PIC16F877A para que haga comunicación con la FPGA del robot por medio de las tarjetas XBEE y una pantalla LCD que muestre la lectura de diversos sensores dispuestos sobre el robot móvil.

El chasis estará formado por 3 etapas la primera donde descansaran las ruedas con su respectivos motores y circuito de potencia, serán un total de 2 ruedas fijas y 1 rueda omnidireccional como guía y dirección, en la segunda etapa estará formada por el circuito de alimentación y el circuito de amplificación de voltaje y la tercera etapa es donde se montará la tarjeta NI sBRIO-9631 en conjunto con los circuito de comunicación, sensores y demás estarán ubicados de acuerdo a su función de tal manera que podrían estar tanto en la parte frontal, posterior, inferior o incluso superior del robot, se procurará que las ruedas estén hechas de caucho para que puedan adaptar al robot a todo tipo de terreno.

Espacial: El desarrollo de la propuesta se llevará a cabo dentro de la carrera de Ingeniería Electrónica mención Sistemas Industriales en el laboratorio del Club de Robótica en las instalaciones de la Universidad Politécnica Salesiana.

Temporal: El proceso de investigación, desarrollo y denuncia del proyecto tendrá un plazo normal de 6 meses con un plazo extendido de 3 meses a partir de la aprobación.

1.3 OBJETIVO GENERAL

- Diseñar y construir un robot autónomo con tecnología FPGA, para el desarrollo de prácticas estudiantiles.

1.4 OBJETIVOS ESPECÍFICOS

- Programar en una tarjeta FPGA NI-9631, un robot entrenador capaz de comportarse a la manera que le configure el usuario.

- Monitorear parámetros del entorno del robot tales como temperatura, proximidad, posición.
- Comunicar de forma inalámbrica las distintas funciones a operar en el robot.
- Visualizar por medio de una cámara el entorno del robot desde cualquier monitor vía web.
- Realizar una programación detallada en LabVIEW de tal manera que el robot pueda tomar decisiones básicas de manera autónoma.

1.5 JUSTIFICACIÓN DEL TRABAJO

En vista del constante surgimiento de nuevas tecnologías y de la necesidad de que los estudiantes de Ingeniería Electrónica de las Universidades tengan pleno conocimiento de los cambios que se van generando a nivel tecnológico se ha propuesto desarrollar un robot autónomo móvil basado en tecnología FPGA, programado en LabVIEW un lenguaje gráfico muy versátil y de fácil entendimiento para que hagan prácticas distintas a las que hacen con micro-controladores logrando obtener un mayor conocimiento de este tipo de tecnología, a su vez puedan descubrir las ventajas y desventajas de esta tecnología así como todas las aplicaciones en las que pueda ser usada y ser una herramienta más que puedan utilizar en su vida profesional.

1.6 VARIABLES E INDICADORES

VARIABLES:

- Temperatura
- Humedad
- Luminosidad

INDICADORES:

- Proximidad
- Velocidad

1.7 MARCO METODOLÓGICO

Se realizó un robot móvil con tecnología FPGA y programado con el software LabVIEW de National Instruments, para evitar el uso de programas codificados en VHDL que son mucho más complejos y que necesitan un mayor proceso de

aprendizaje para un amplio conocimiento del mismo, sin menospreciar su validez por ser una de las principales formas de programación de estos dispositivos; pero que queda relegado con la facilidad de programación implementada por National Instruments que está orientada a formar parte de laboratorio gráfico el cual transforma sus librerías a VHDL generando una forma amigable de programar y de llegar al estudiante.

El enfoque de este proyecto es orientado a prácticas para los estudiantes las cuales tendrán como objetivo ir paso a paso otorgando funcionalidades al robot desde las más simples hasta las más complejas como los son desde el proceso del movimiento de una rueda hasta la comunicación inalámbrica que debe existir entre un mando remoto manejado por el usuario y la máquina. Se desarrollarán varias prácticas con el robot donde se exponga en cada una de ellas una configuración distinta para el mismo y así demostrar las funcionalidades y ventajas de las FPGAs sobre otros dispositivos.

1.8 POBLACIÓN Y MUESTRA

La popularización de la tecnología FPGA ha obligado a la mayoría de las universidades con carreras técnicas para que abran su campo de estudio a estos dispositivos que todavía están evolucionando conforme al mercado existente en las diferentes partes del mundo, sin embargo, ya han tenido un incremento exponencial en su distribución en los últimos años gracias a la continua necesidad de mejorar los sistemas automatizados y al requerimiento de una mayor cantidad de cálculo para obtener plantas más eficientes.

Los estudiantes que se gradúan de las carreras de ingeniería ya sean estos de electrónica, mecatrónica o sistemas computacionales van viendo la necesidad de salir al mercado laboral con más herramientas para enfrentarse al mundo tecnológico que se vive hoy en día, donde saber que es una FPGA, CPLD, DSP, etc, es una necesidad apremiante y más aún conocer el manejo de los mismos. Por ello, el desarrollo de este proyecto contribuye al mejor entendimiento de esta tecnología.

Hay que agregar que en el campo de la robótica el uso de nuevas tecnologías ayudará a que los proyectos poco a poco vayan aumentando su complejidad y sus

beneficios, donde se podrá realmente demostrar que el talento para el desarrollo tecnológico proviene de nuestro propio país y no del extranjero.

1.9 DESCRIPCIÓN DE LA PROPUESTA

El robot móvil consta de una dos plataformas sobre las cuales irán montadas las tarjetas electrónicas, el router, los racks tipo borneras, la cámara de red y sobre todo la tarjeta NI sbRIO-963, en la plataforma inferior irán montados los dos motores de alto torque junto con los encoders para la lectura de datos, también las baterías recargables. A los costados y en la parte inferior del robot móvil tiene montados los sensores de ultrasonido los cuales son la parte esencial para el desarrollo de algunas prácticas. En la parte superior está la tarjeta de adquisición de datos, que contiene los sensores de lectura de variables de entorno junto con el display LCD que permite visualizar todos los datos adquiridos por los sensores.

La plataforma de programación de la tarjeta sbRIO-9631 es LabVIEW de National Instruments, plataforma donde se desarrolló todo el trabajo de programación del robot móvil.

1.10 BENEFICIARIOS

Los beneficiarios de este proyecto son los estudiantes del Club de Robótica, las materias de Electiva, Electrónica y la misma Universidad Politécnica Salesiana la cual podrá hacer uso del robot móvil y de cada una de sus prácticas para la demostración y formación didáctica de todos los estudiante interesados en conocer más del mundo de la robótica.

1.11 IMPACTO

Este Robot Móvil y la Robótica en general tienen un gran impacto a nivel social, ya que con las mejoras de los sistemas robóticos aceleran la producción en la industria, mejora la calidad de vida y nos ayuda a tener una mejor comprensión del mundo en el que vivimos, pues como se sabe los robots móviles son utilizados incluso en la exploración espacial. Pero en corto plazo ayuda a los estudiantes de la Universidad Politécnica Salesiana a conocer de nuevas tecnologías y aplicarlas en el diseño de diferentes tipos de robots los cuales, promueven el uso de los conocimientos adquiridos en las distintas materias y en la investigación por parte de los estudiantes.

CAPÍTULO 2. MARCO TEÓRICO

2.1 ANTECEDENTES HISTÓRICOS

A lo largo de toda la historia, el hombre se ha sentido fascinado por máquinas y dispositivos capaces de imitar las funciones y los movimientos de los seres vivos. Los griegos tenían una palabra específica para denominar a estas máquinas: automatos. De esta palabra deriva la actual autómatas: máquina que imita la figura y movimientos de un ser animado. Los mecanismos animados de Herón de Alejandría (85 d.C.) se movían a través de dispositivos hidráulicos, poleas y palancas y tenían fines eminentemente lúdicos.

Durante los siglos XV y XVI alguno de los más relevantes representantes del renacimiento se interesan por los ingenios descritos y desarrollados por los griegos. Durante los siglos XVII y XVIII se crearon ingenios mecánicos que tenían alguna de las características de los robots actuales. Estos dispositivos fueron creados en su mayoría por artesanos del gremio de la relojería con el fin de entretener a la gente de la corte.

A finales del siglo XVIII y principios del XIX se desarrollaron algunas ingeniosas invenciones mecánicas, utilizadas fundamentalmente en la industria textil, a partir de ese momento cuando se empieza a utilizar dispositivos automáticos en la producción, dando paso a la automatización industrial.

La palabra robot fue usada por primera vez en el año 1921, cuando el escritor checo Karel Capek (1890-1938), estrena en el teatro nacional de Praga su obra Rossum's Universal Robot. Su origen es la palabra eslava robota, que se refiere al trabajo realizado de manera forzada. Pero sin duda alguna, fue el escritor americano de origen ruso Isaac Asimov (1920-1992) el máximo impulsor de la palabra robot. En octubre de 1945 publicó en la revista Galaxy Science Fiction una historia en la que por primera vez enunció sus tres leyes de la robótica.

- 3.1 Un robot no puede perjudicar a un ser humano, ni con su inacción permitir que un ser humano sufra daño.
- 3.2 Un robot ha de obedecer las órdenes recibidas de un ser humano, excepto si tales órdenes entran en conflicto con la primera ley.

3.3 Un robot debe proteger su propia existencia mientras tal protección no entre en conflicto con la primera o segunda ley. (Barrientos-Peñin-Balaguer-Aracil, 1997)

2.2 LA ROBÓTICA

La robótica es una ciencia o rama de la tecnología, que estudia el diseño y la construcción de máquinas capaces de desempeñar tareas realizadas por el ser humano o que requieren del uso de inteligencia. La robótica está encaminada a diseñar y construir sistemas capaces de realizar tareas propias de un ser humano. Es una tecnología multidisciplinar, ya que hace uso de los recursos que le proporcionan otras ciencias, ya que en el proceso de diseño y construcción de un robot intervienen muchos campos pertenecientes a otras ramas de la ciencia. (UNSAAC, 2006)



Figura 2.1 La robótica en el mundo entero. En el mundo de hoy la ciencia y tecnología van de la mano, en este caso la robótica dentro de un futuro muy cercano será parte de todo en nuestras vidas. Poom.W. (28 de Febrero de 2013). Poom 21 del siglo. Tomado del blog <http://poomcemera.blogspot.com>

La informática, la electrónica, la mecánica y la ingeniería son solo algunas de las disciplinas que se combinan en la robótica. El objetivo principal de la robótica es la construcción de equipos o artefactos que funcionen de manera automática y que realicen trabajos dificultosos o imposibles para los seres humanos. (UNSAAC, 2006)

2.3 LOS ROBOTS MÓVILES

La robótica móvil con los años ha evolucionado con los avances tecnológicos que se han dado recientemente, lo cual ha dado lugar al desarrollo de múltiples aplicaciones en distintas áreas. Si bien, esta evolución ha alcanzado un nivel elevado en la actualidad, aún hay que esperar para la llegada de la revolución social de la

robótica; donde todos los robots llegarían a ser un elemento más en el hogar, en las empresas y en la sociedad en general. (Calle, 2007)

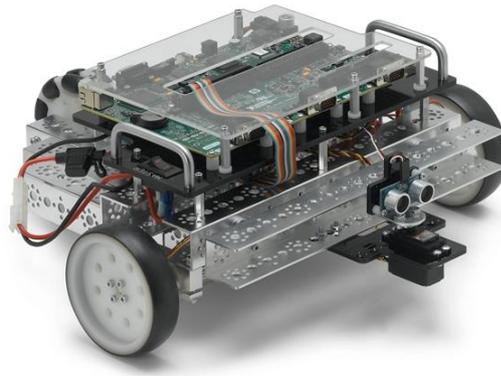


Figura 2.2 Robot Móvil Starter Kit LabView Robotics. Robot Móvil creado por National Instruments para uso de tarjeta NI sbRIO-9631, con entorno de programación gráfica LabVIEW Robotics. Por National Instruments. (03 de Noviembre de 2014). Vista general de los módulos de LabVIEW Robotics. Tomado de <http://www.ni.com/white-paper/11564/en>.

Los robots hoy en día son en su mayoría excesivamente de costos elevados y su función está muy especializada, debido a esto los estudios actuales se centran en conseguir que estos robots sean más generales y económicos, de tal manera sean accesibles a todas las personas que quieran resolver algún problema en sus tareas diarias.

Los robots móviles por lo general son de reducidas dimensiones con un bajo grado de control (el control existente es para regular la velocidad de las ruedas de tracción, el grado de control no es alto comparado con el nivel de control que se emplea en los robots articulados), adquieren un alto grado de complejidad cuando tienen que realizar tareas cooperativas.

El comportamiento y la forma de actuar del microrobot están determinados por el programa que se ejecuta en él. El software de los robots se encuentra actualmente estructura en tres niveles que son:

- Sistema operativo
- Plataforma de desarrollo
- Aplicaciones concretas

El que un robot realice autónomamente tareas de modo eficiente depende de la construcción mecánica y de la programación. La precisión en su desplazamiento se encuentra determinada por su sistema mecánico, mientras que la autonomía y la inteligencia se encuentran residentes en el programa que gobierna las acciones del microbot. (Calle, 2007)

2.4 ESTRUCTURA GENERAL DE UN ROBOT MÓVIL

Debido a que un robot móvil por lo general está destinado a simular el comportamiento de personales y animales con un nivel de eficiencia similar, la estructura del robot móvil es muy similar a la estructura de un ser vivo.

Se puede apreciar la estructura del robot móvil y la del ser vivo en la Figura 2.3, en donde se puede apreciar de manera general la estructura, tanto de un robot móvil como de un ser vivo.

La estructura de un robot móvil presentada en la Figura 2.3, está conformada por diferentes subestructuras, tales como:

- **Estructura mecánica:** estructura con ruedas, patas y orugas.
- **Actuadores:** motores, luces, brazos, ruedas y en definitiva cualquier elemento que permita interactuar con el entorno.
- **Sensores:** sonar, láser, cámaras y cualquier elemento que nos proporcione información del entorno.
- **Inteligencia:** métodos, algoritmos, etc. Estos van a permitir, a partir de la información de los sensores, interactuar con el entorno. (Calle, 2007)

2.5 GRADOS DE LIBERTAD DE UN ROBOT MÓVIL

Grado de libertad (GDL): Es cada uno de los movimientos de desplazamiento y rotación que puede realizar el robot. (Calle, 2007)

- Un cuerpo que se mueve en dos dimensiones tiene 3 GDL (una rotación y 2 traslaciones).
- Un cuerpo que se mueve en tres dimensiones tiene 6 GDL (3 rotaciones y 3 traslaciones).

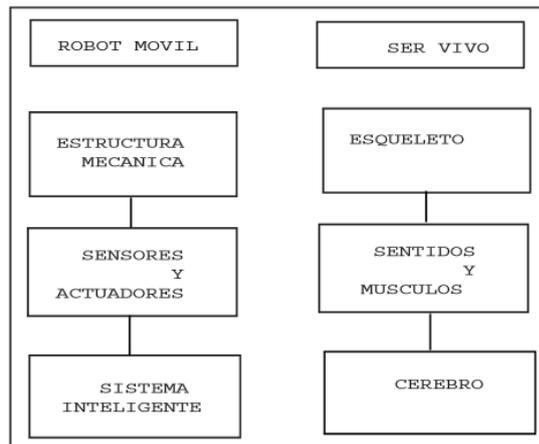


Figura 2.3 Similitudes entre un robot y un ser vivo. Semejanza gráfica real entre las partes del cuerpo humano en comparación con las partes principales que destacan a un robot móvil. Calle, N. C. (2007). Estudio de robots móviles con ruedas. Repositorio Digital UPS. Obtenido de <http://dspace.ups.edu.ec/bitstream/123456789/175/2/Capitulo%201.pdf>

2.6 CONFIGURACIÓN DIFERENCIAL DE UN ROBOT MÓVIL

Consta de dos ruedas colocadas en el eje perpendicular a la dirección del robot. Cada rueda es controlada por un motor, de tal forma que el giro del robot queda determinado por la diferencia de velocidad de las ruedas. Así, para girar a la izquierda, hay que darle una velocidad mayor a la rueda derecha. Para girar a la derecha hay que darle una velocidad mayor a la rueda izquierda. La Figura 2.4 muestra este tipo de configuración.

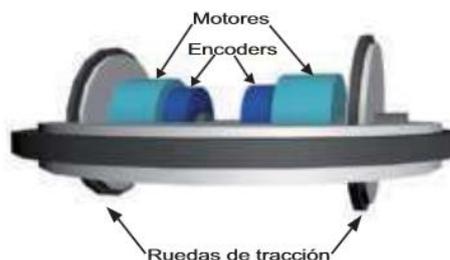


Figura 2.4 Robot móvil con estructura diferencial. Imagen de un robot móvil usando ruedas de tracción, parte principal de la estructura diferencial, conectado a encoders para la lectura de posición y velocidad. Calle, N. C. (2007). Estudio de robots móviles con ruedas. Repositorio Digital UPS. Obtenido de <http://dspace.ups.edu.ec/bitstream/123456789/175/2/Capitulo%201.pdf>

Uno de los problemas que tiene la configuración diferencial es mantener el equilibrio del robot, ya que consta de dos ruedas, por lo cual se le agregan ruedas de libre giro. Estas ruedas sirven para mantener horizontal al robot, por lo que giran libremente según el movimiento. Además, estas ruedas se orientan hacia la dirección del robot. Dependiendo de las necesidades, se pueden agregar una o más ruedas de libre giro.

Este tipo de robots no son aptos para terrenos irregulares, por lo que su campo de acción de encuentra limitado a superficies planas. (Calle, 2007)

2.7 CENTRO INSTANTÁNEO DE CURVATURA (ICC)

Se define como el punto por la cual cruzan los ejes de todas las ruedas, es el punto alrededor del cual el robot gira en un instante determinado. En la figura 2.5 se muestra el ICC para la configuración diferencial, ackerman y triciclo. (Calle, 2007)

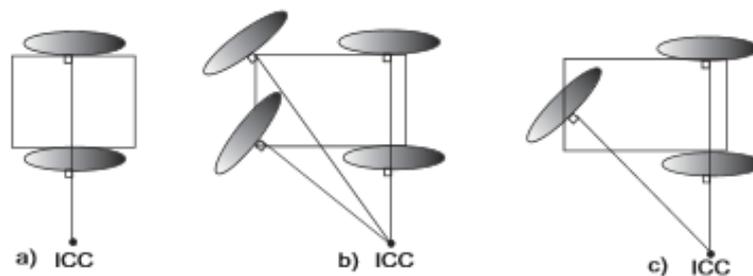


Figura 2.5 ICC en a) Configuración diferencial. b) Configuración ackerman. c) Configuración triciclo. Centro Instantáneo de Curvatura mostrada en las diferentes configuraciones para el diseño de robots móviles. Calle, N. C. (2007). Estudio de robots móviles con ruedas. Repositorio Digital UPS. Obtenido de <http://dspace.ups.edu.ec/bitstream/123456789/175/2/Capitulo%201.pdf>

2.8 SISTEMAS EMBEBIDOS EN LOS ROBOTS MÓVILES

Se entiende por “Sistemas Embebidos” a “una combinación de hardware y software de computadora, sumado tal vez a algunas piezas mecánicas o de otro tipo, diseñado para tener una función específica. Es común el uso de estos dispositivos pero pocos se dan cuenta que hay un procesador y un programa ejecutándose que les permite funcionar”. (Llinares, 2010)

Las principales características de un sistema embebido son el bajo costo y consumo de potencia. Dado que muchos sistemas embebidos son concebidos para ser productos en miles o millones de unidades, el costo por unidad es un aspecto importante a tener en cuenta en la etapa de diseño. Generalmente, los sistemas embebidos emplean procesadores muy básicos, relativamente lentos y memorias pequeñas para minimizar los costos. (Llinares, 2010)

2.9 SISTEMAS DE TIEMPO REAL (STR)

Los sistemas de tiempo real (STR) son aquellos que interactúan rápidamente con su entorno físico para realizar funciones de supervisión y control, en donde el tiempo es un factor muy importante, por lo que debe reaccionar a estímulos provenientes del ambiente, respondiendo con fuertes restricciones de tiempo en muchos casos, por lo tanto, un sistema se dice que trabaja en tiempo real si la información después de la adquisición y tratamiento es todavía vigente. (Aldea, 2005)

Un sistema embebido puede o no ser de tiempo real dependiendo de los requerimientos específicos de la aplicación que se requiere implementa.

Tiempo real, no solo significa que un sistema pueda responder de forma muy rápida, sino que se puede determinar con precisión cuando una sección del programa se ejecutará. Como consecuencia, para un sistema de tiempo real:

- El tiempo en que se ejecutan las acciones del sistema es significativo.
- Se imponen requerimientos temporales al software, es decir responde a los estímulos que recibe del entorno físico dentro de un plazo de tiempo determinado.
- El sistema operativo o el lenguaje de programación proporcionará el soporte para que las tareas se ejecuten de forma concurrente (ejecución paralela, simultaneidad en la ejecución de múltiples tareas).

Para que el funcionamiento del sistema sea correcto no basta con que las acciones sean correctas lógicamente, sino que tienen que ejecutarse dentro del intervalo de tiempo especificado. Esto es debido a que el sistema está conectado a un proceso externo del que recibe estímulos a los que debe responder con suficiente rapidez para evitar que evoluciones a un estado indeseable. Lo habitual es que un sistema de

tiempo real tenga que realizar varias actividades de forma concurrente. (Puente, 2007)

2.10 FPGA (ARREGLOS DE COMPUERTAS PROGRAMABLES EN CAMPO)

En el nivel más alto, los FPGAs son chips de silicio reprogramables. Al utilizar bloques de lógica pre-construidos y recursos para ruteo programable, usted puede configurar estos chips para implementar funcionalidades personalizadas en hardware y compilarlas en un archivo de configuración o bistream que contenga información de cómo deben conectarse los componentes. Además, los FPGAs son completamente reconfigurables y al instante toman una nueva “personalidad” cuando usted compila una diferente configuración de circuitos. Anteriormente sólo los ingenieros con un profundo entendimiento de diseño de hardware digital podían trabajar con la tecnología FPGA. Sin embargo, el aumento de herramientas de diseño de alto nivel está cambiando las reglas de programación de FPGAs, con nuevas tecnologías que convierten los diagramas de bloques gráficos, o hasta el código ASIC a circuitos de hardware digital.

2.10.1 Cinco beneficios Principales de la Tecnología FPGA

1. **Rendimiento.-** Aprovechando el paralelismo del hardware, los FPGAs exceden la potencia de cómputo de los procesadores digitales de señales (DSPs) rompiendo el paradigma de ejecución secuencial logrando más en cada ciclo de reloj, BDTI, una destacada firma de analistas que realiza evaluaciones de referencia, lanzó evaluaciones mostrando cómo los FPGAs pueden entregar significativamente más potencia de procesamiento por dólar que una solución DSP, en algunas aplicaciones. El controlar entradas y salidas (E/S) a nivel de hardware ofrece tiempos de respuesta más veloces y funcionalidad especializada que coincide con los requerimientos de una aplicación.
2. **Tiempo en llegar al mercado.-** La tecnología FPGA ofrece flexibilidad y capacidades de rápido desarrollo de prototipos para enfrentar los retos de que un producto se libere tarde al mercado. Usted puede probar una idea o un concepto y verificarlo en hardware sin tener que pasar por el largo proceso de fabricación por el que pasa un diseño personalizado ASIC.

3. **Precio.-** El precio de la ingeniería no recurrente de un diseño personalizado ASIC excede considerablemente al de las soluciones de hardware basadas en FPGA. La fuerte inversión inicial de los ASICs es fácilmente justificable para los fabricantes de equipos originales que embarcan millones de chips por año, pero muchos usuarios finales necesitan funcionalidad de un hardware personalizado para decenas o cientos de sistemas en desarrollo. La misma naturaleza programable de silicio implica que no hay precio de fabricación o largos plazos de ejecución de ensamblado. Los requerimientos de un sistema van cambiando con el tiempo, y el precio de cambiar incrementa los diseños FPGA es insignificante al compararlo con el precio de implementar cambios en un ASIC antes de su lanzamiento.

4. **Fiabilidad.-** Mientras que las herramientas de software ofrecen un entorno de programación, los circuitos de un FPGA son una implementación segura de la ejecución de un programa. Los sistemas basados en procesadores frecuentemente implican varios niveles de abstracción para auxiliar a programar las tareas y compartir los recursos entre procesos múltiples. El software a nivel driver se encarga de administrar los recursos de hardware y el sistema operativo administra la memoria y el ancho de banda del procesador. El núcleo de un procesador sólo puede ejecutar una instrucción a la vez, y los sistemas basados en procesadores están siempre en riesgo de que sus tareas se obstruyan entre sí. Los FPGA's, que no necesitan sistemas operativos, minimizan los retos de fiabilidad con ejecución paralela y hardware preciso dedicado a cada tarea.

5. **Mantenimiento a largo plazo.-** Como se mencionó anteriormente, los chips FPGA son actualizables en campo y no requieren el tiempo y el precio que implica rediseñar un ASIC. Los protocolos de comunicación digital por ejemplo, tienen especificaciones que podrían cambiar con el tiempo, y las interfaces basadas ASICs podrían causar retos de mantenimiento y habilidad de actualización. Los chips FPGA, al ser reconfigurables, son capaces de mantenerse al tanto con modificaciones a futuro que pudieran ser necesarias. Mientras el producto o sistema se va desarrollando, usted puede implementarle mejoras funcionales sin la necesidad de invertir tiempo rediseñando el hardware o modificando el diseño de la tarjeta. (National Instruments, 2015)

2.11 CONFIGURACIÓN DE LA FPGA

La forma de configurar la FPGA varía dependiendo de la tecnología adoptada, en caso de las FPGA basadas en tecnología atifuse las mismas son programadas una sola vez y haciendo uso de un programador dedicado.

En el caso de la tecnología SRAM las FPGA leen su configuración interna de una memoria no volátil que es la que mantiene el programa. La lectura se realiza en forma serie, una vez finalizada la transferencia la FPGA se configura y comienza a funcionar. Esta carga se realiza en los POWER-UP o por un reset del dispositivo.

En este campo para el uso de esta tecnología en la actividad espacial o militar se han incorporado técnicas de recarga del programa sin afectar el funcionamiento de la FPGA. Todo esto es apuntado a conseguir que las FPGA basadas en tecnología SRAM sean utilizadas en lugares críticos, ya que el solo hecho de que su configuración dependa de un bit volátil genera cierta desconfianza en el medio. La memoria de configuración normalmente está fabricada por el mismo fabricante de la FPGA, esta puede ser reemplazada por otro dispositivo mientras ser respeten los protocolos de transferencia. (Cavallero-Gutierrez, 2010)

2.12 SISTEMAS DE CONTROL Y ADQUISICIÓN DE DATOS DE NATIONAL INSTRUMENTS

National Instruments ofrece una gran variedad de diversos tipos de hardware de propósito general disponibles comercialmente y que compartan una arquitectura común reconfigurable de E/S (RIO). Esta arquitectura combina tres aspectos importantes, un procesador en tiempo real, una FPGA y una amplia gama de E/S, incluyendo E/S analógicas y digitales, por lo que para su programación es necesario instalar y emplear dos módulos propios de LABVIEW: LABVIEW Real Time y LABVIEW. Usando esta arquitectura estándar y las herramientas gráficas que LABVIEW ofrece se pueden diseñar y crear de manera rápida prototipos de sistemas embebidos, máquinas de control y monitorización industrial flexible y de alto rendimiento. (National Instruments, 2008)

Entre los diferentes sistemas que comparten esta arquitectura común reconfigurable de E/S (RIO) tenemos: PXI RIO, PCI RIO, Compact RIO, Compact RIO Integrated, Single-Board RIO.

El robot móvil empleado para la elaboración de las guías y prácticas de laboratorio de esta tesis tiene incorporado una tarjeta Single-Board RIO (abreviatura Sb-RIO) de National Instruments, como elemento principal de control para la adquisición y procesamiento de datos, por lo tanto se hará mención de aquí en adelante de este hardware.

La plataforma Single-Board RIO, es una tarjeta que combina dispositivos embebidos de despliegue, tiene un procesador en tiempo real, una FPGA (arreglo de compuertas en campo) y E/S digitales y analógicas en una sola tarjeta.

Las tarjetas SbRIO están diseñadas para aplicaciones de alto volumen, control embebido y adquisiciones que requieren alto rendimiento y fiabilidad. (National Instruments, 2011)

2.13 INTRODUCCIÓN A LA PLATAFORMA DE PROGRAMACIÓN

GRÁFICA LABVIEW

La plataforma de programación gráfica LabVIEW es un entorno de desarrollo gráfico con diferentes funciones integradas para la realización de control de instrumentos, adquisición de datos, análisis de medidas y presentaciones de datos. (González-Pradines, 2007)

LabVIEW hoy en día es un potente sistema de programación, con un ambiente mucho más cómodo, sencillo y flexible en comparación con otros entornos tradicionales.

Los programas diseñados con la plataforma LabVIEW son denominados usualmente como VI (Virtual Instruments) porque estos aparecen y actúan como una copia exacta de instrumentos físicos como por ejemplo los instrumentos de lectura y medición.

Una de las ventajas que posee el entorno gráfico LabVIEW, es que el diseño de este software es ideal para la adquisición de datos y para el monitoreo de señales. Además LabVIEW posee facilidad de trabajo con interfaces de comunicación como puerto serial, puerto paralelo, USB, GPIB, PXI, VXI, TCP/IP, UDP, DataStock, Irda, Bluetooth, OPC, entre otros.

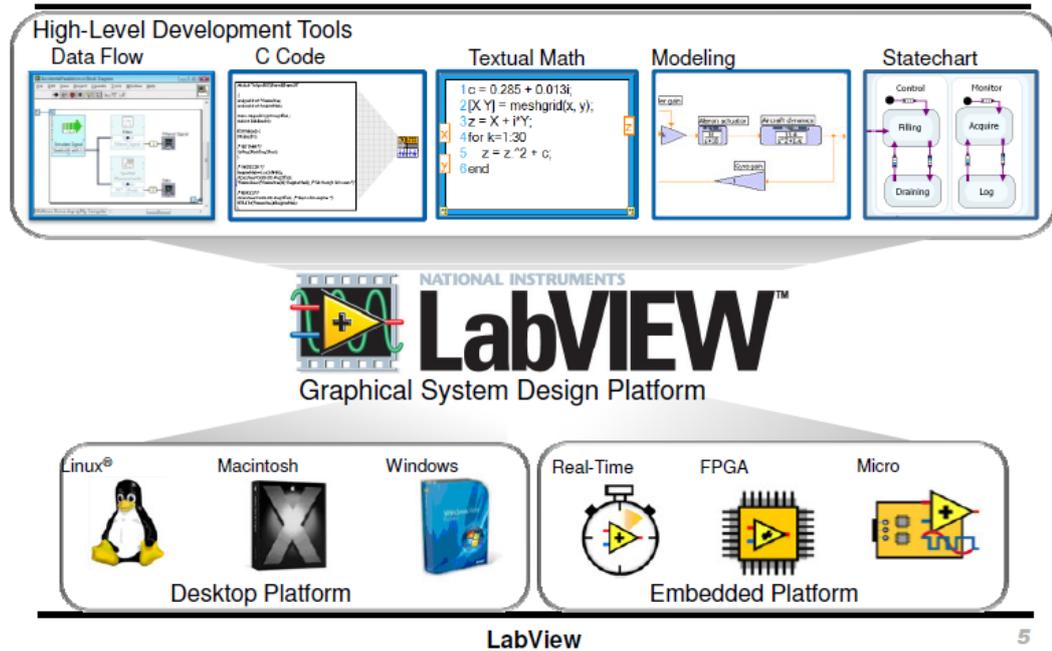


Figura 2.6 Plataforma de desarrollo gráfico LabVIEW trabajando con distintos lenguajes de programación trabajando con diferentes sistemas operativos y con diversos sistemas embebidos. Por National Instruments. (14 de Junio de 2013). Obtenido de <http://www.ni.com/white-paper/14559/en/>

LabVIEW también posee la capacidad de interactuar con otros lenguajes y aplicaciones como:

- DLL (librería de funciones), .NET, ActiveX, MultiSim, Matlab/Simulink, AutoCAD, SolidWorks, RobotSim, etc.
- Herramientas gráficas y textuales para el procesamiento de señales digitales.
- Visualización y manejo de gráficas con datos dinámicos.
- Adquisición y tratamientos de imágenes.
- Control de movimiento
- Tiempo Real
- Programación de FPGA´s para control
- Sincronización entre dispositivos.

La tarjeta de control NI sbRIO-9631 de National Instruments que se incorpora en el diseño del robot móvil, posee una arquitectura RIO, compuesta por un arreglo de

compuertas programables en campo (FPGA), un procesador de tiempo real y módulos de conexión de E/S.

National Instruments incorpora en el ambiente de desarrollo LabVIEW, los módulos LabVIEW FPGA y LabVIEW Real Time, los cuales permiten manejar y programar al robot móvil. Por este motivo cuando se realiza un programa en el robot móvil, se realizan dos tipos de VIs:

- Un VI se lo realiza empleando las funciones del módulo LabVIEW FPGA y que solamente lo ejecuta sobre el dispositivo FPGA.
- Y otro VI que se lo desarrolla a partir del VI FPGA previamente compilado y empleando funciones del módulo LabVIEW Real Time, pero con la diferencia que éste se ejecuta sobre el procesador de tiempo real.

Gracias al módulo LabVIEW Real Time se pueden realizar diferentes aplicaciones entre ellas industriales, médicas, y en nuestro caso robóticas con un menor tiempo de desarrollo y aplicaciones de control avanzado. El usuario o programador puede personalizar de manera rápida el programa sobre el FPGA usando el módulo LabVIEW FPGA para el procesamiento de señales y el control de alta velocidad. Además contiene controladores incorporados y APIs para el manejo y transferencia de datos entre el FPGA y el procesador en tiempo real.

Adicionalmente National Instruments incorpora un nuevo módulo conocido como LabVIEW Robotics, el cual ofrece una extensa biblioteca de robótica con conectividad integrada con sensores de robótica y actuadores, algoritmos básicos para operación inteligente, percepción robusta y funciones de movimiento para mover un robot o vehículo, facilitando la programación de aplicaciones complejas de robótica al proporcionar un alto nivel de abstracción, que en combinación con los módulos LabVIEW FPGA y LabVIEW Real Time permiten al usuario diseñar todo tipo de aplicaciones robóticas en un menor tiempo y de una manera más fácil y sencilla, permitiendo programar desde un simple encendido de led en el robot, hasta programar un diseño más complejo con múltiples subsistemas de ejecución en paralelo. (Figuroa-Palaguachi, 2012)

2.14 DESCRIPCION DE LA PLATAFORMA GRAFICA DE PROGRAMACION LABVIEW

2.14.1 MEASUREMENT & AUTOMATION EXPLORER (MAX)

Max es un programa desarrollado por National Instruments que se utiliza para listar, configurar y probar el software y hardware que puede ser usado en el desarrollo de un VI (Instrumentación virtual). A través de este programa permite también visualizar los puertos del computador, como el serie y paralelo si es que se contara con ellos.

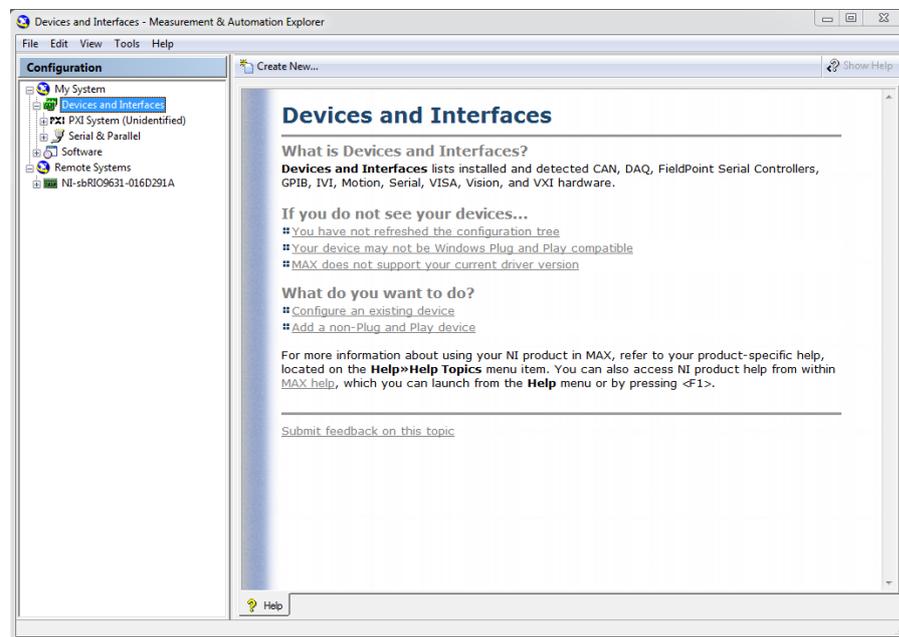


Figura 2.7 Measurement & Automation Explorer (MAX). Programa desarrollado por National Instrument para configuración de hardware y software de la misma marca.

Dependiendo del tipo de hardware que va a ser usado, el MAX selecciona un tipo de controlador que será el adecuado para el mismo, por ejemplo las tarjetas DAQ cuentan con un controlador NI-DAQ y para dispositivos de tiempo real con arquitectura RIO se cuenta con un controlador NI-RIO.

Un sistema remoto hace referencia a un dispositivo de tiempo real que puede manejarse o se puede configurar sobre la red. Los sistemas remotos no son iguales que los dispositivos de red. Un dispositivo de red es cualquier tipo de dispositivo que

sea accesible a una conexión Ethernet o inalámbrica, tales dispositivos pueden ser accesibles por varias computadoras y no ejecutan un sistema operativo de tiempo real, como una cámara IP.

2.14.2 ENTORNO GRÁFICO

Los programas de LabVIEW se denominan instrumentos virtuales o VI, debido a que su apariencia y operación imita a los instrumentos físicos, tales como osciloscopios y multímetros. Los VIs tienen una parte interactiva con el usuario y otra parte de código fuente y aceptan parámetros procedentes de otros VIs.

Todos los VIs tienen dos ventanas principales un panel frontal donde estarán sus botones, pantallas, etc, y un diagrama de bloques donde estará la circuitería interna. Las paletas contienen las opciones que se emplean para crear y modificar los VIs.

- **Panel frontal:** Se trata de la interfaz gráfica del VI con el usuario. Esta interfaz recoge las entradas procedentes del usuario y representa las salidas proporcionadas por el programa. Un panel frontal está formado por una serie de botones, pulsadores, potenciómetros, gráficas, etc. Cada uno de ellos puede estar definido como un control o un indicador. Los primeros sirven para introducir parámetros al VI, mientras que los indicadores se emplean para mostrar los resultados producidos, ya sean datos adquiridos o resultados de alguna operación.

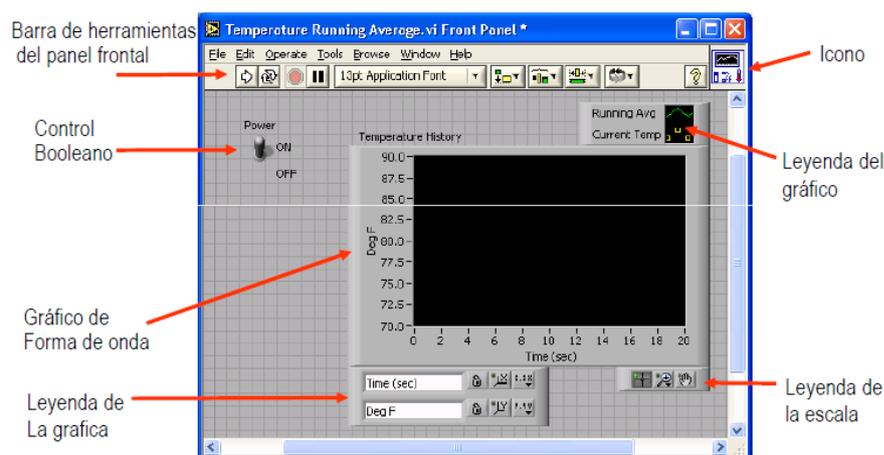


Figura 2.8 Panel frontal de LabVIEW. Partes principales que forman el Panel Frontal dentro del entorno de desarrollo gráfico LabVIEW.

- **Diagrama de bloques:** Se constituye el código fuente del VI. En el diagrama de bloques es donde se realiza la implementación del programa VI para controlar o realizar cualquier procesamiento de las entradas y salidas que se crearon en el panel frontal. (Figuroa-Palaguachi, 2012)

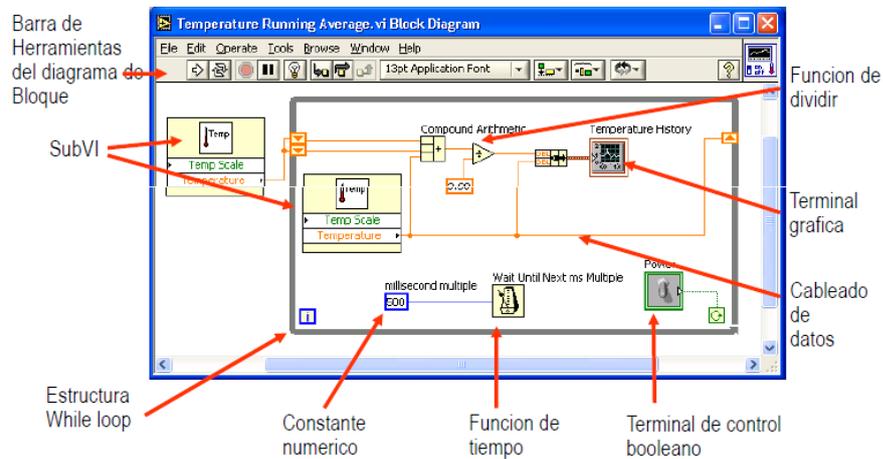


Figura 2.9 Diagrama de Bloques en LabVIEW. Partes principales que forman parte del Diagrama de Bloques en el entorno de desarrollo gráfico LabVIEW.

2.15 ADQUISICIÓN DE DATOS

La adquisición de datos consiste en la toma de muestras y valores del mundo real para generar datos que puedan ser usados por un computador (del sistema análogo al sistema digital). Esto consiste en tomar un conjunto de variables físicas mediante sensores y transductores que serán convertidas en tensiones eléctricas y luego digitalizarlas de manera que puedan ser procesadas por un computador.

La empresa National Instruments ha diseñado actualmente varios tipos de sistemas de adquisición de datos, éstos módulos o tarjetas pueden ser conectadas al computador mediante diferentes puertos (paralelo, serie, USB, Ethernet) o ranuras conectadas en la tarjeta (PCI, PXI, ISA) en la placa madre, uno de los referentes de National Instruments son las tarjetas DAQ (*Data Acquisition*) que son las más empleadas para la adquisición de datos.

Para el desarrollo de esta tesis haremos uso de una tarjeta sbRIO-9631 que cuenta con su propio sistema operativo de tiempo real y que es independiente del

computador una vez que se haya descargado en ella el o los programas para una aplicación específica.

2.15.1 Tarjeta FPGA NI Single Board-9631

La tarjeta SbRIO-9631 de la familia de Single-Board RIO de National Instruments, es el sistema controlador del robot móvil autónomo. Esta tarjeta está diseñada para cualquier tipo de aplicación que requiera flexibilidad, alto rendimiento y fiabilidad, tomando todas estas consideraciones se optó por el diseño del robot móvil para su buen desarrollo y funcionamiento.

Ahora es necesario describir las características de nuestra tarjeta SbRIO-9631, cuenta con un procesador industrial embebido MPC5200 Freescale con una velocidad de procesamiento de 266 Mhz, para aplicaciones en tiempo real determinístico. Su procesador de tiempo real se encuentra conectado con un chip FPGA Spartan-3E (1M de compuertas) a través de un bus PCI de alta velocidad, las E/S se conectan directamente al FPGA, proporcionando una personalización de bajo nivel para la temporización y el procesamiento de señales de E/S. También la tarjeta SbRIO-9631 cuenta con una memoria DRAM de 64 MB y una memoria Flash de 128MB para almacenamiento no volátil, todo esto podemos observarlo en la siguiente figura.

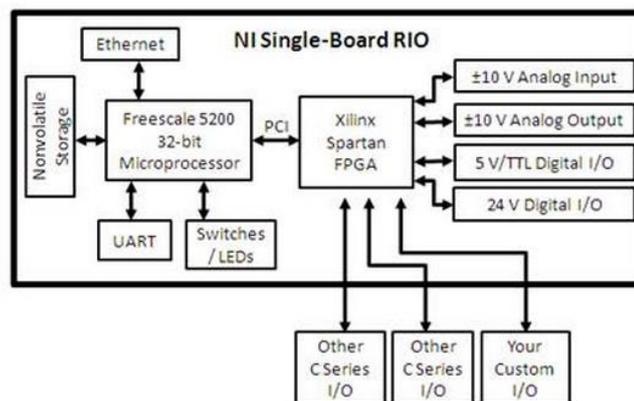


Figura 2.10 Diagrama de conexión de la tarjeta NI sbRIO-9631. Diagrama de bloques que forman parte de una tarjeta FPGA NI Single-Board familia RIO desarrollado por National Instruments. Por National Instruments. (07 de Febrero de 2011). NI Single Board RIO Embedded Control and Acquisition. Obtenido de <http://www.ni.com/white-paper/7441/es/>

La tarjeta Single Board SbRIO-9631, es programada usando un lenguaje de programación gráfico en LabVIEW, el programa es ejecutado sobre el sistema operativo de tiempo real Wind River VxWorks haciendo uso del módulo LabVIEW Real Time instalado en el procesador MPC5200. Además podemos personalizar el programa ejecutado sobre el FPGA usando el módulo LabVIEW FPGA para el control de alta velocidad y procesamiento de señales.

En la Figura 2.11 podemos destacar los componentes principales que forman parte de nuestra tarjeta FPGA Single-Board SbRIO-9631 con los que el usuario puede interactuar y a su vez realizar las conexiones de hardware necesarias para cualquier tipo de aplicación, en nuestro caso para el desarrollo de nuestro robot móvil autónomo. (Figueroa-Palaguachi, 2012)

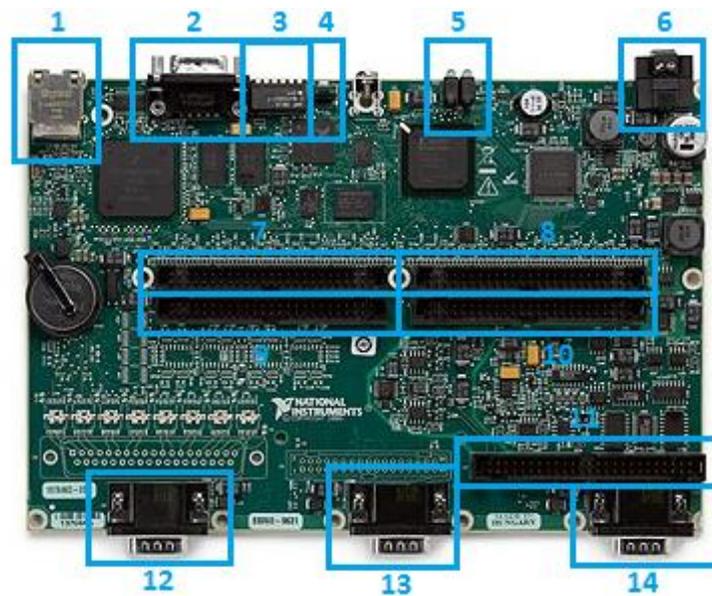


Figura 2.11 Tarjeta NI SbRIO-9631 de National Instruments. Imagen en la que detalla con recuadros las partes principales que conforman la tarjeta FPGA. Por National Instruments. (2015). NI sbRIO-9631 Dispositivos Embebidos con AI, AO, DIO, FPGA de 1M de Compuertas. Obtenido de http://sine.ni.com/nips/cds/pages/image?imagepath=/images/products/us/sbrio9631_1.jpg&title=NI%20sbRIO9631&oracleLang=esa

2.15.2 Componentes principales de la tarjeta SbRIO-9631

1. Puerto Ethernet RJ45
2. Puerto Serial RS-232

3. Interruptores DIP
4. Botón Reset
5. Indicadores LEDs
6. Conector de Alimentación DC
7. Conector P2, E/S Digitales de 3,3V
8. Conector P3, E/S Digitales de 3,3V
9. Conector P4, E/S Digitales de 3,3V
10. Conector P5, E/S Digitales de 3,3V
11. Conector E/S Analógicas
12. Conector para Módulo 1 Serie C
13. Conector para Módulo 2 Serie C
14. Conector para Módulo 3 Serie C

A continuación detallaremos cada uno de los componentes que forman parte de la tarjeta SBRIO-9631.

2.15.2.1 Puerto Ethernet RJ-45

El puerto Ethernet RJ-45 permite conectar el robot móvil con el computador vía Ethernet y de esta manera descargar y ejecutar todos los programas realizados en el entorno de desarrollo gráfico de LabVIEW. También permite conectar el robot móvil con otros dispositivos externos como switches, routers, cámaras IP, entre otros, a través de los protocolos TCP/IP, UDP, Modbus/TCP. (National Instruments, 2008)

2.15.2.2 Puerto Serial RS-232

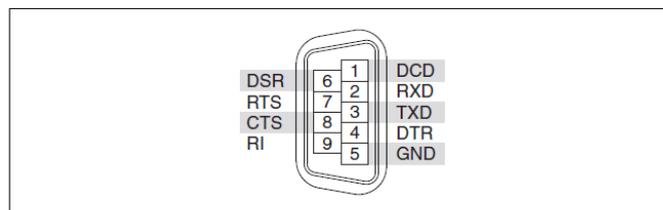


Figura 2.12 Puerto Serial RS-232. Detalle de los pines que forman parte del puerto serial conectado en la tarjeta NI sbRIO-9631. Por National Instruments. (s.f.). USER GUIDE: NI sbRIO-961x/963x/964x and NI sbRIO-9612XT/9632XT/9642XT. Obtenido de <http://www.ni.com/pdf/manuals/375052c.pdf>

Este puerto serial RS-232 tipo DB9, se le puede conectar cualquier dispositivo de E/S que maneje los protocolos de comunicación por ejemplo: microcontroladores,

GPS, módulos inalámbricos, módulos de voz, etc. Para poder hacer uso de este puerto empleamos VIs Seriales que incluyen en LabVIEW, además de que el programa debe estar dentro de una aplicación de LabVIEW Real Time. (National Instruments, 2008)

2.15.2.3 Interruptores DIP

Está formado por 6 interruptores montados sobre la tarjeta SbRIO, inicialmente cada uno de ellos se encuentran en la posición OFF (hacia arriba), cada uno de ellos posee una función la cual explicaremos a continuación.

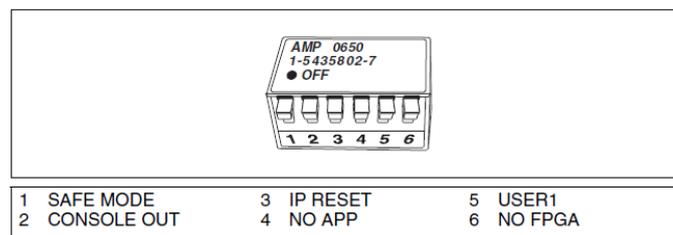


Figura 2.13 Interruptores DIP. Detalle del trabajo que realiza cada switch en la tarjeta NI sbRIO-9631. Por National Instruments. (s.f.). USER GUIDE: NI sbRIO-961x/963x/964x and NI sbRIO-9612XT/9632XT/9642XT. Obtenido de <http://www.ni.com/pdf/manuals/375052c.pdf>

- **Interruptor SAFE MODE:** La posición del interruptor (ON/OFF) determina si el procesador embebido de tiempo real se pone en marcha cuando energizamos la tarjeta SbRIO.
- **Interruptor CONSOLE OUT:** Este interruptor nos permite el uso del puerto serial para leer la dirección IP y la versión del firmware de la tarjeta SbRIO, haciendo uso de un programa diseñado en LabVIEW.
- **Interruptor IP RESET:** Este interruptor es empleado para restablecer la dirección IP del robot móvil a 0.0.0.0.
- **Interruptor NO APP:** Este interruptor en posición ON es usado para evitar que un archivo de tipo ejecutable realizado en LabVIEW Real Time y guardado en la tarjeta SbRIO, se ejecute al energizar la tarjeta SbRIO del robot móvil. Si se desea que el archivo ejecutable no se ejecute de forma permanente en el arranque de la tarjeta, se debe deshabilitar esta opción desde el software LabVIEW.

- **Interruptor USER1:** Este interruptor es de propósito general, debido a que el usuario puede definir la función que realizará el interruptor dentro del programa realizado en el VI Real Time.
- **Interruptor NO FPGA :** Este interruptor activo, o sea en posición ON, evita que cualquier aplicación diseñada en LabVIEW FPGA se cargue al energizar el dispositivo SbRIO. (National Instruments, 2008)

2.14.2.4 Botón RESET

Este botón al ser pulsado reinicia el procesador de tiempo real, mientras que el FPGA continúa ejecutándose. Esta es una de las muchas formas de opciones de Reset que tiene la tarjeta NI sbRIO-9631.

2.15.2.5 Indicadores Leds

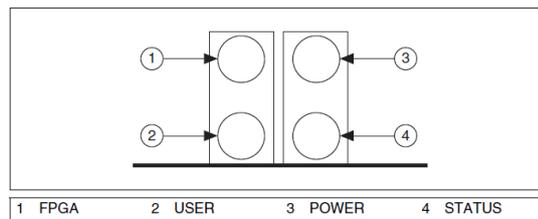


Figura 2.14 Leds Indicadores del dispositivo SbRIO-9631. Indicadores del funcionamiento de cada led en la tarjeta NI sbRIO-9631. Por National Instruments. (s.f.). USER GUIDE: NI sbRIO-961x/963x/964x and NI sbRIO-9612XT/9632XT/9642XT. Obtenido de <http://www.ni.com/pdf/manuals/375052c.pdf>

A continuación se detalla el funcionamiento de los indicadores leds que se encuentran en la tarjeta FPGA SbRIO-9631 mostrados en la figura.

- **FPGA:** Este led es de propósito general. El programador define el estado de la tarjeta (ON u OFF). Se lo puede usar para ayudar a depurar o recuperar fácilmente el estado de un aplicación robótica.
- **USER:** Este led también es de propósito general, la diferencia entre ambos radica en que la señal digital que controla al Led FPGA proviene del chip FPGA mientras que la señal digital que controla al Led USER proviene directamente del procesador embebido.

- **POWER:** Este led se ilumina cuando la tarjeta SbRIO ha sido energizado, en nuestro caso cuando el robot móvil ha sido activado.
- **STATUS:** Este led por lo general está apagado durante el funcionamiento normal del equipo. Indica condiciones específicas de error mediante el parpadeo de cierto número de veces. (National Instruments, 2008)

2.15.2.6 Conector de Alimentación

La tarjeta SbRIO-9631 como se ha mencionado anteriormente tiene un rango de entrada de alimentación de energía de 19 a 30 VDC. En la Figura 2.15 se muestra la forma correcta de conectar los cables de alimentación en el conector de la tarjeta SbRIO. (National Instruments, 2008)

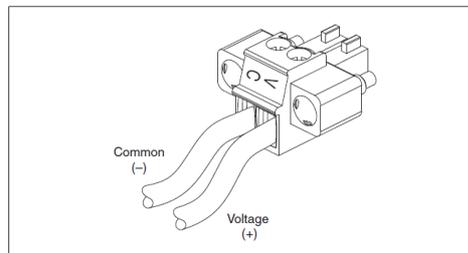


Figura 2.15 Conector de Alimentación al dispositivo SbRIO. Imagen que detalla la polaridad del conector de alimentación de la tarjeta NI sbRIO-9631. Por National Instruments. (s.f.). USER GUIDE: NI sbRIO-961x/963x/964x and NI sbRIO-9612XT/9632XT/9642XT. Obtenido de <http://www.ni.com/pdf/manuals/37052c.pdf>

2.15.2.7 Conectores de Puertos de E/S Digitales

El chip FGA que se encuentra montado sobre la tarjeta SbRIO, se conecta directamente a todos los puertos de E/S digitales, estas se encuentran distribuidas en cuatro conectores (P2, P3, P4, P5) y en 10 puertos (Port0 a Port9). Cada conector de los puertos está formado por 50 pines, de los cuales dos pines proporcionan +5VDC, útiles para aplicaciones externas, ésta fuente tiene protecciones de limitación de corriente y sobretensiones.

En las siguientes figuras se detallan la distribución de los pines de los conectores de E/S digitales montados sobre la tarjeta FPGA SbRIO-9631. (National Instruments, 2008)

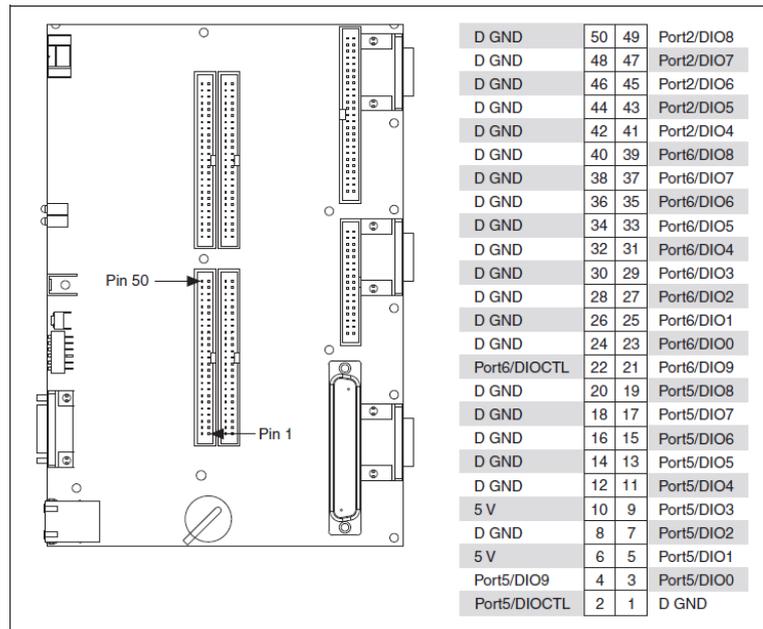


Figura 2.16 Conector P2 de 3.3V de E/S Digitales. Imagen que detalla los pines de conexión de I/O del puerto P2. Por National Instruments. (s.f.). USER GUIDE: NI sbRIO-961x/963x/964x and NI sbRIO-9612XT/9632XT/9642XT. Obtenido de <http://www.ni.com/pdf/manuals/375052c.pdf>

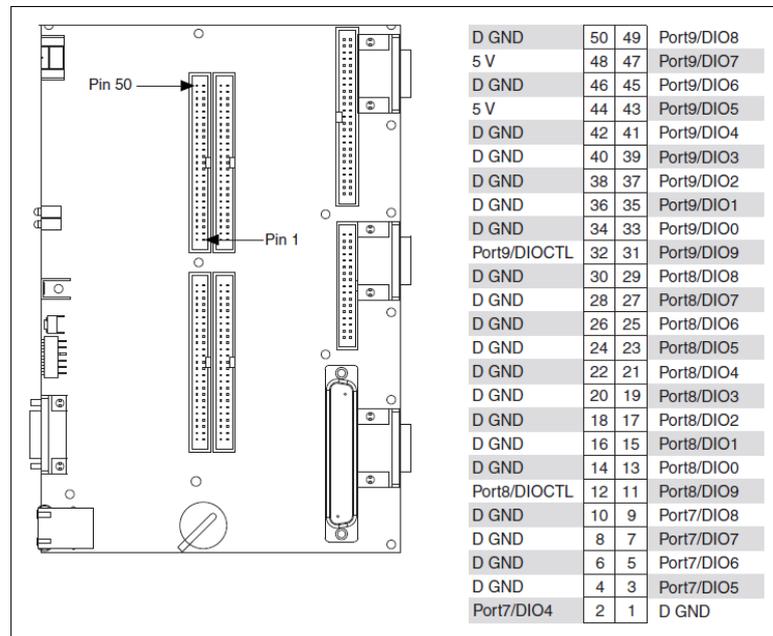


Figura 2.17 Conector P3 de 3.3V de E/S Digitales. Imagen que detalla los pines de conexión de DIO del puerto P3. Por National Instruments. (s.f.). USER GUIDE: NI sbRIO-961x/963x/964x and NI sbRIO-9612XT/9632XT/9642XT. Obtenido de <http://www.ni.com/pdf/manuals/375052c.pdf>

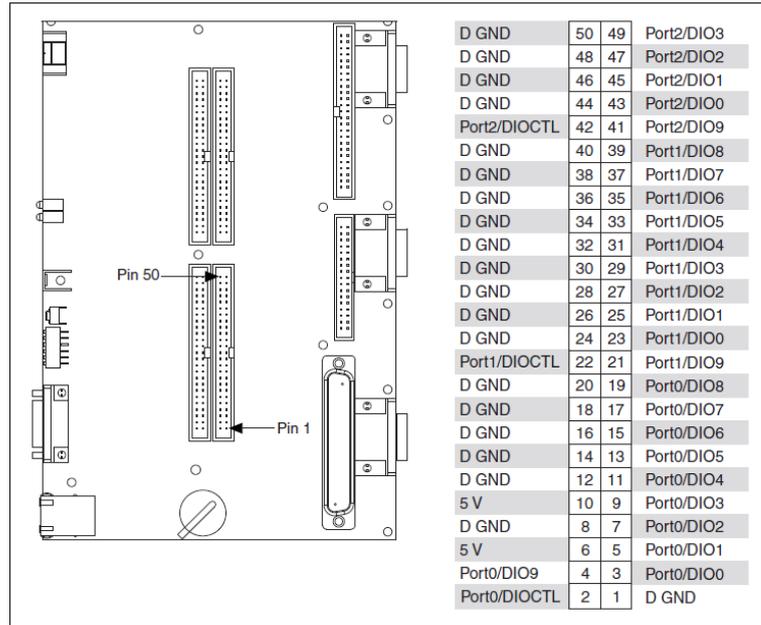


Figura 2.18 Conector P4 de 3.3V de E/S Digitales. Imagen que detalla los pines de conexión de DIO del puerto P4. Por National Instruments. (s.f.). USER GUIDE: NI sbRIO-961x/963x/964x and NI sbRIO-9612XT/9632XT/9642XT. Obtenido de <http://www.ni.com/pdf/manuals/375052c.pdf>

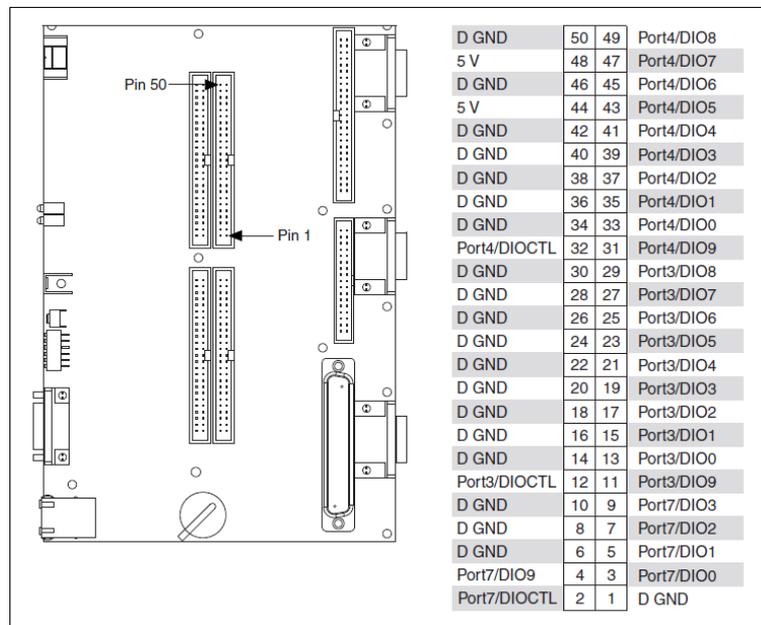


Figura 2.19 Conector P5 de 3.3V de E/S Digitales. Imagen que detalla los pines de conexión de DIO del puerto P5. Por National Instruments. (s.f.). USER GUIDE: NI sbRIO-961x/963x/964x and NI sbRIO-9612XT/9632XT/9642XT. Obtenido de <http://www.ni.com/pdf/manuals/375052c.pdf>

2.15.2.8 Conector E/S Analógicas

Posee 32 entradas analógicas AI que equivalen a 16 canales diferenciales con resoluciones ADC de 16 bits y 4 salidas analógicas AO con resoluciones DAC de 16 bits. Estos rangos de voltaje nominal DC de las entradas analógicas son de: +/-10V, +/-5V, +/-1V, +/-0.2V y el rango de voltaje DC de las salidas analógicas es de +/-10V. Todos los voltajes son relativos a AI GND y AO GND respectivamente.

Cabe indicar que todos los pines de tierra (D GND, AI GND, AO GND) están conectados entre sí internamente en el dispositivo SbrRIO, es necesario que todas las fuentes de alimentación tengan su conexión a tierra al dispositivo SbrRIO. (National Instruments, 2008)

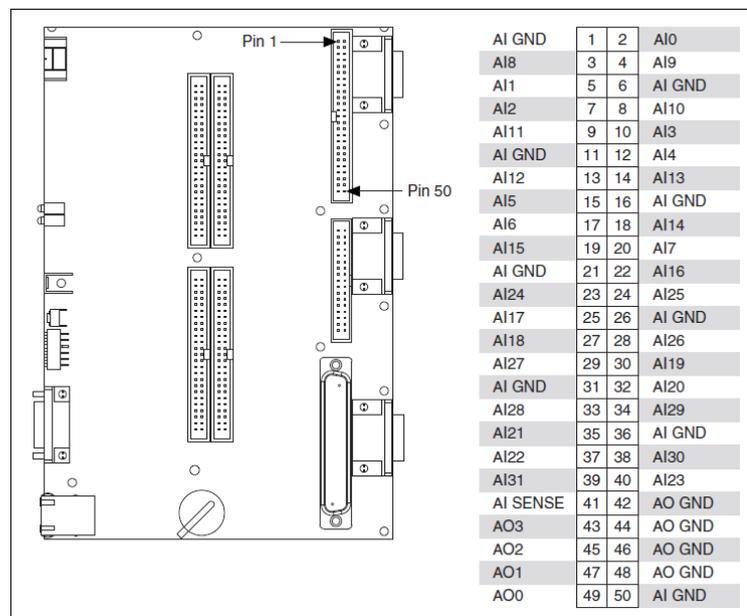


Figura 2.20 Conector J7 de E/S Analógicas. Imagen que detalla los pines de conexión de AI/AO del puerto J7. Por National Instruments. (s.f.). USER GUIDE: NI sbRIO-961x/963x/964x and NI sbRIO-9612XT/9632XT/9642XT. Obtenido de <http://www.ni.com/pdf/manuals/375052c.pdf>

2.16 APLICACIONES DE CONTROL EN TIEMPO REAL

Una vez explicado el funcionamiento de esta tesis diseñado para aplicaciones robóticas, vamos a diseñar aplicaciones de tiempo real. A continuación se explica todo lo correspondiente a los sistemas LabVIEW de tiempo real aplicados sobre el

robot móvil, específicamente en la tarjeta FPGA sbRIO Starter Kit y las funcionalidades de los módulos LabVIEW Real Time y LabVIEW FPGA.

2.16.1 Conceptos Generales

Para crear aplicaciones de control de tiempo real existen una gran variedad de formas de programar, entendiendo y teniendo claro los conceptos fundamentales de programación de tiempo real y aplicaciones de control.

2.16.1.1 Capacidad de Respuesta

La aplicación de control diseñada tiene que reaccionar ante los eventos tales como una variación en las señales de E/S. El tiempo necesario para tomar medidas después de un evento toma el nombre de capacidad de respuesta, que va desde los microsegundos a minutos, la capacidad de respuesta determina la velocidad del lazo de control y afecta las señales de E/S, procesador y decisiones de software.

La tarjeta FPGA con la que está formado el robot móvil, posee el controlador sbRIO-9631, este cuenta con un procesador industrial MPC5200 Freescale con una velocidad de procesamiento de 266 Mhz, para aplicaciones en tiempo real determinístico, lo que implica una capacidad de respuesta de 3,76 nano segundos ($1/266$ Mhz). Podemos diseñar circuitos de adquisición y control personalizados en el chip FPGA con una resolución de temporizado/disparo de 3,76 ns. Debido a la naturaleza paralela misma del núcleo RIO, el agregar cálculos adicionales no reduce la velocidad de la aplicación FPGA.

2.16.1.2 Determinismo y Fluctuación

Cuando hablamos de determinismo nos referimos a la repetibilidad de los tiempos de un lazo de control. La fluctuación hace referencia al error en el tiempo y esta es la forma de medir el determinismo. Se tomará como ejemplo lo siguiente: si se tiene un lazo configurado para ejecutar y actualizar las salidas un vez cada 40 ms, pero funciona a 40,5 ms, la fluctuación generada es de 0,5 ms. Un determinismo alta y la fiabilidad son las principales ventajas de un sistema de control en tiempo real y un buen determinismo es muy crítico para aplicaciones de control estable. Por otro lado un determinismo bajo conduce a un control analógico deficiente y puede lograr que el sistema deje en cualquier momento de responder.

2.16.1.3 Prioridad

La mayoría de los controladores usan un único procesador para manejar todo el control, supervisión y diferentes tareas de comunicación, debido a que solo cuenta con un solo recurso como es el procesador, éste mantiene múltiples demandas paralelas, para ello necesita una manera de dar prioridad a las demandas que son más importantes.

A través de la configuración de lazos de control críticos para una alta prioridad, es necesario tener un controlador de funciones completas que exhiba un buen determinismo y una buena capacidad de respuesta. (Figuroa-Palaguachi, 2012)

2.17 ARQUITECTURA Y FUNCIONAMIENTO DE UN SISTEMA

LABVIEW FPGA DE TIEMPO REAL

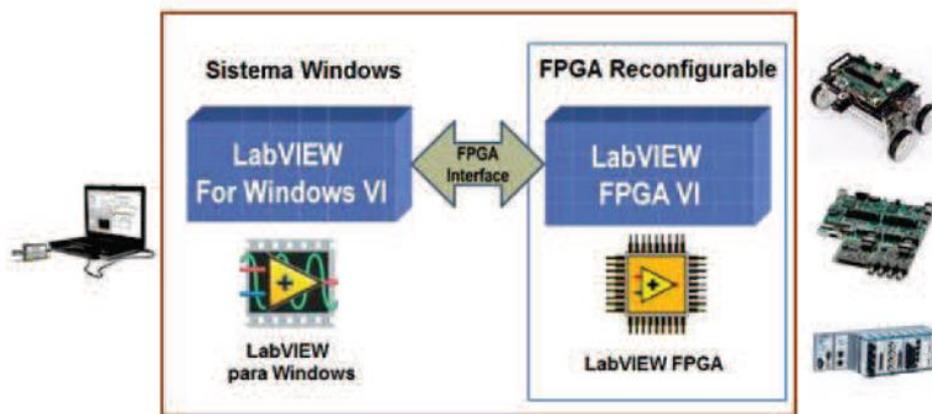


Figura 2.21 Arquitectura del Sistema LabVIEW FPGA. Arquitectura interna del trabajo que realiza la FPGA como interface entre el computador y los dispositivos RIO (Single-Board, Compact RIO). Por National Instruments. (Marzo de 2011). LabVIEW FPGA Module Training . Obtenido de <http://zone.ni.com/devzone/cda/tut/p/id/3555>

Para conocer la arquitectura y funcionamiento de un sistema LabVIEW FPGA, observar la Figura 2.21, éste cuenta con un VI que se ejecuta en la plataforma LabVIEW para un sistema operativo Window, este VI se comunica con otro VI en el FPGA del sistema remoto con arquitectura RIO. Para establecer comunicación entre el computador y el hardware de arquitectura tipo RIO es necesario tener instalado el driver NI-RIO en el computador.

Partiendo de la arquitectura del Sistema LabVIEW FPGA, se puede definir cómo es la arquitectura y funcionamiento de un Sistema LabVIEW FPGA de Tiempo Real. Esta arquitectura es la que se implementa en la mayoría de los proyectos destinados para cualquier tipo de aplicación en nuestro caso en aplicaciones robóticas. Generalmente para desarrollar una aplicación de tiempo real en LABVIEW, se emplea la estructura *Time Loop* debido a las características de funcionamiento que posee.

En la Figura 2.22 se puede observar la arquitectura de un Sistema LabVIEW FPGA de Tiempo Real cuenta con cuatro tipos de VIs, un VI que se ejecuta directamente sobre el sistema operativo Windows del computador (interfaz gráfica con el usuario), dos VIs ejecutados sobre el procesador de tiempo real del sistema remoto siendo uno con prioridad normal y el otro sirve como interfaz de tiempo crítico, y un último VI que es ejecutado sobre el chip reconfigurable de la FPGA. Todos los VIs son programados mediante LabVIEW agrupados en un solo proyecto. (Figueroa-Palaguachi, 2012)

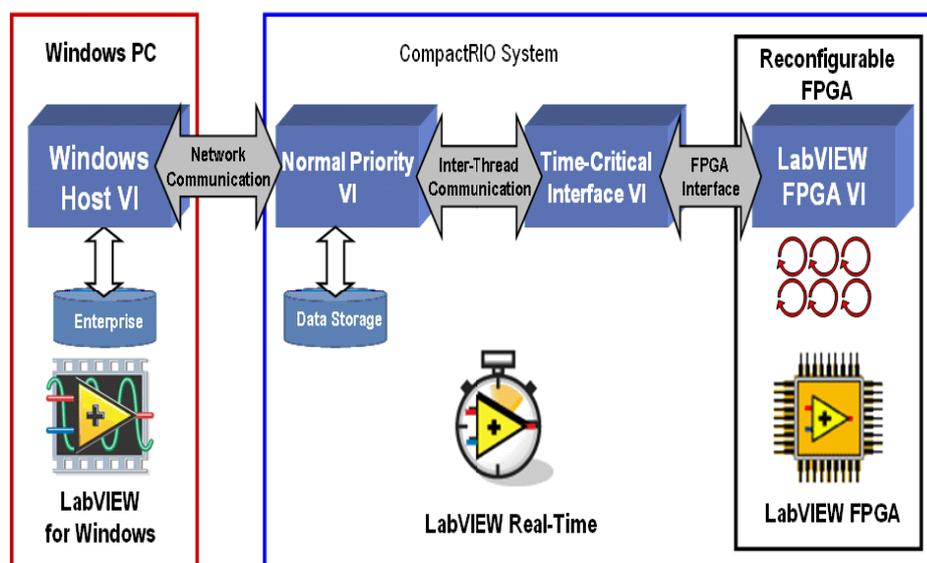


Figura 2.22 Arquitectura de un Sistema LabVIEW FPGA Real Time. Arquitectura interna de la comunicación del sistema LabVIEW Real Time con el computador y el sistema LabVIEW FPGA. Por National Instruments. (19 de Octubre de 2012). Desarrollo personalizado de medición e control de E/S de Hardware con o módulo LabVIEW FPGA. Obtenido de <http://www.ni.com/white-paper/11162/pt>.

2.18 MÓDULOS LABVIEW PARA EL DESARROLLO DE APLICACIONES

2.18.1 Módulo LabVIEW FPGA

El Módulo LabVIEW FPGA junto con LabVIEW trabajan en un ambiente gráfico de desarrollo de arreglo de compuertas programables en campo (FPGAs) que forman parte de un sistema remoto con arquitectura RIO, proporcionando al usuario un ambiente de desarrollo gráfico agradable, debido a que el programador no necesita tener experiencia previa acerca de lenguajes VHDL tradicionales o de diseño de hardware a nivel de tarjeta para que pueda realizar una determinada aplicación.

El uso del Módulo LabVIEW FPGA permite desarrollar VIs FPGA que luego hacen uso de un servidor de compilación local o remoto ejecutado sobre Windows, para posteriormente LabVIEW implemente el código en hardware. Podemos crear VIs FPGA embebidos que combinen acceso directo a las E/S del chip FPGA con la lógica de programación definida por el usuario, para el desarrollo de aplicaciones como protocolos de comunicación digital, procesamiento de señales (DSP), simulación de control hardware, RF, prototipos robóticos y muchas aplicaciones que requieren fiabilidad de hardware de alta velocidad y determinismo estrecho.

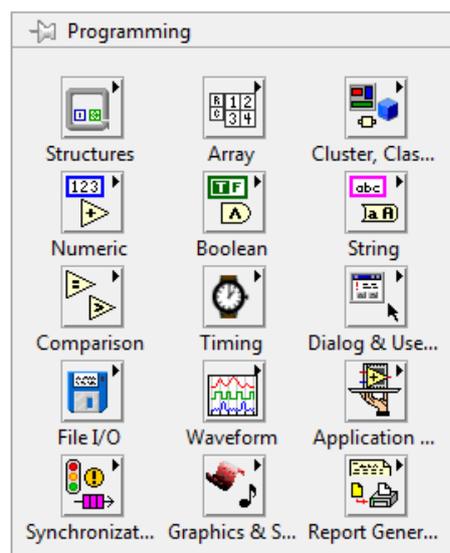


Figura 2.23 Funciones de programación LabVIEW. Paleta de funciones de programación del Diagrama de bloques en LabVIEW. Estas funciones son la base principal para la programación en LabVIEW FPGA y LabVIEW Real Time.

Como información adicional el Módulo LabVIEW FPGA está compuesto por múltiples componentes de software que permiten a los usuarios programar hardware embebido con LabVIEW. Para acceder a las diferentes funciones que sirven para interactuar con el chip FPGA, lo hacemos desde la paleta Programming en el diagrama de bloques del VI FPGA.

LabVIEW permite el uso de las funciones más habituales que conforman a la paleta Programming para realizar la programación en paralelo del VI FPGA, como son el manejo de arreglos, estructuras, cluster, funciones numéricas, funciones booleanas, funciones de comparación y funciones de temporización. Las funciones que se adicionan una vez instalado el Módulo LabVIEW FPGA son las que encontramos dentro de las opciones FPGA I/O, Memory & FIFO, Synchronization, FPGA MATH & Analysis e IP Integration.

El resto de VIs de esta paleta cumplen con funcionalidades diferentes como medida de señales DC y valores eficaces RMS, medida de periodos analógicos, empleo de filtros Butterworth y Notch, transformada de Fourier FFT, manejo de medias, varianza y desviación estándar.

Para acceder de forma directa a las entradas y salidas empleadas en la programación del VI FPGA desde un VI Real Time o desde un VI Computer, debemos ir a la paleta de funciones del diagrama de bloques del VI y abrir la paleta FPGA Interface, mostrada en la siguiente figura.

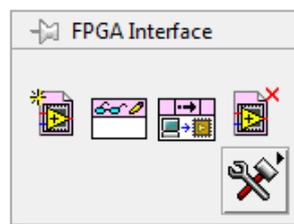


Figura 2.24 Paleta FPGA Interface. Paleta de funciones de programación de la FPGA que permite obtener las entradas y salidas de la tarjeta NI sbRIO-9631.

El VI Open FPGA VI Reference, permite abrir una referencia del VI FPGA sobre el VI Real Time o el VI Computer. Una vez colocado sobre el diagrama de bloques del VI Real Time debemos hacer referencia al VI FPGA. Para ello se puede hacer de dos formas. Uno de ellos consiste en arrastrar el VI FPGA (previamente ya

compilado) que se encuentra en el Explorador de Proyectos y colocarlo dentro de Open FPGA VI Reference. El otro consiste en desplegar el menú contextual de Open FPGA VI Reference y seleccionamos la opción Configure Open FPGA VI Reference, inmediatamente se despliega una ventana como la mostrada en la figura 2.25.

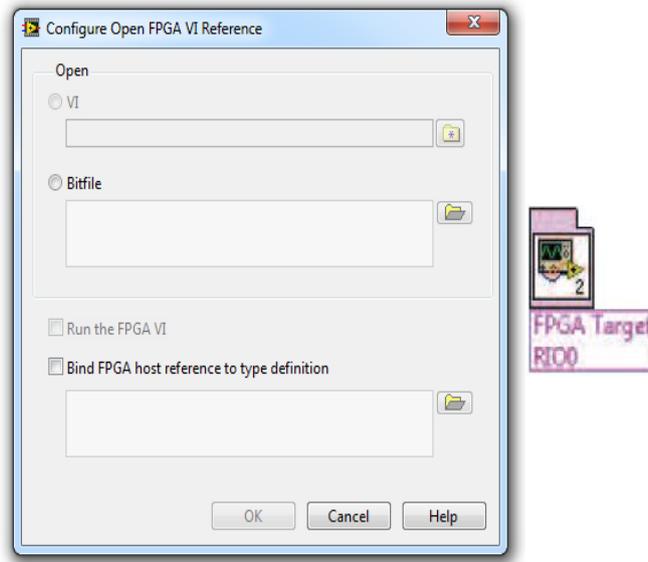


Figura 2.25 Ventana Configure Open FPGA VI Reference. Ventana de configuración que permite abrir una referencia del VI FPGA sobre un VI Real Time para manejar LabVIEW Real Time en la tarjeta NI sbRIO-9631.

En la ventana de la Figura 2.25, se observan tres opciones que permiten acceder a tres tipos de archivos que hacen referencia al VI FPGA, a través de un archivo ejecutable que se encuentra en Build Specifications, también a través del propio VI FPGA o accediendo a un archivo de bits que se genera después de la compilación del VI FPGA.

Otra de las opciones que se tiene es la paleta FPGA Interface, aquí se tiene el VI Read/Write Control que nos permite la lectura y escritura de las E/S definidas en el VI FPGA. También está el VI Invoke Method que nos permite acceder a todos los métodos definidos sobre el FPGA, el Close FPGA VI Reference permite cerrar la referencia del VI FPGA y otro de los VIs que más se usan es el FPGA Reference Dynamic Refnum que se lo emplea para hacer referencias de un VI FPGA en varios VIs Real Time del mismo proyecto. (Figuroa-Palaguachi, 2012)

2.18.2 Módulo LabVIEW Real Time

El Módulo LabVIEW Real Time es un componente adicional para el sistema de desarrollo gráfico de LabVIEW, usado para crear y depurar aplicaciones determinísticas y confiables que se desarrollan sobre sistemas remotos de tiempo real.

Con el Módulo LabVIEW Real Time se puede desarrollar aplicaciones a todos los sistemas remotos de la National Instruments en tiempo real, como son los de Compact RIO, Single-Board RIO, PXI, Compact FieldPoint, entre otros.

El sistema operativo de tiempo real que posee el LabVIEW Real Time es un solo kernel dedicado que proporciona a los usuarios la máxima fiabilidad y temporización consistente para aplicaciones de sistemas embebidos.

Para acceder a todas las funciones del Módulo Real Time, se abre el VI Real Time creado para el sistema remoto (target sbRIO) y desde la paleta de funciones elegimos la subpaleta *Real-Time*, como la mostrada en la Figura 2.26. (Figuroa-Palaguachi, 2012)

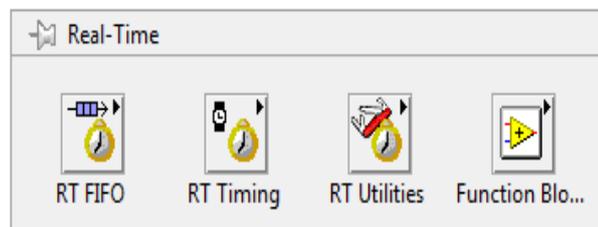


Figura 2.26 Paleta Real Time. Paleta de funciones que maneja el módulo Real Time para ciertos usos y aplicaciones en la tarjeta NI sbRIO-9631.

2.19 CREACION DE UN PROYECTO

Para la creación de un proyecto en LabVIEW se puede seguir por varios caminos, el más usado será a partir de la ventana de inicialización Getting Started de LabVIEW. (Ver Figura 2.27)

Desde la ventana Getting Started que aparece cuando se inicia LabVIEW, se puede crear proyectos a partir de las opciones Empty Project y Real-Time Project en la diferentes opciones de New.

Primero es necesario dar clic sobre la opción Empty Project o ir a File > New Project, donde se abre una ventana que despliega los componentes del proyecto a realizarse. Normalmente aparece vacío debido a que no cuenta con ningún archivo en su estructura y es el usuario quien tiene que añadir todos los archivos que se requerirán en el proyecto. (Ver Figura 2.28)

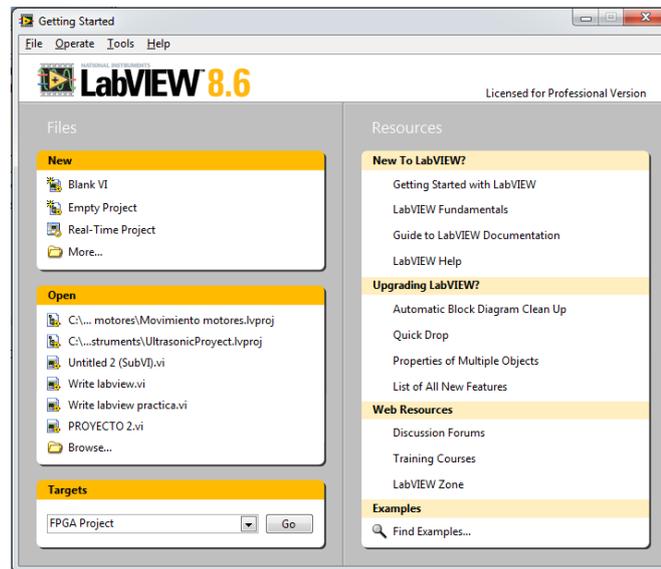


Figura 2.27 Ventana Getting Started de LabVIEW. Ventana de inicio de LabVIEW que permite la creación de nuevos VIs y nuevos Proyectos desde las tarjeta NI sbRIO-9631.

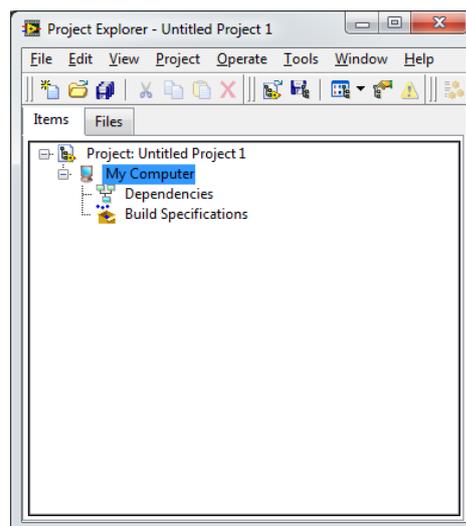


Figura 2.28 Ventana Project Explorer de LabVIEW. Ventana Explorador de Proyectos que permite ver la raíz de un proyecto realizado desde diferentes opciones en LabVIEW.

De ahí es necesario dar clic en New > Real-Time Project o en Targets > Real-Time Project con lo cual se despliega una ventana de configuración como la que se muestra en la Figura 2.29. Esta opción es empleada cuando se usa sistemas remotos de tiempo real conectados al computador como por ejemplo un controlador Compact RIO o una sbRIO.

Otra de las opciones que nos permite crear un proyecto es dando clic en Targets > FPGA Project y hacemos clic en el botón Go, inmediatamente se despliega una ventana de configuración como la mostrada en la Figura 2.29. Esta opción se la emplea cuando se trabaja con dispositivos que contienen chips FPGAs y están conectados al computador. Cabe indicar que tanto esta opción como la anterior mencionada para la creación de proyectos, comparten arquitectura RIO (Compact RIO o sbRIO), por lo que ambos, para crear proyectos pueden manejar estos tipos de dispositivos conjuntamente, es decir, que el proyecto contendrá archivos que manejen la parte de tiempo real como la parte FPGA.

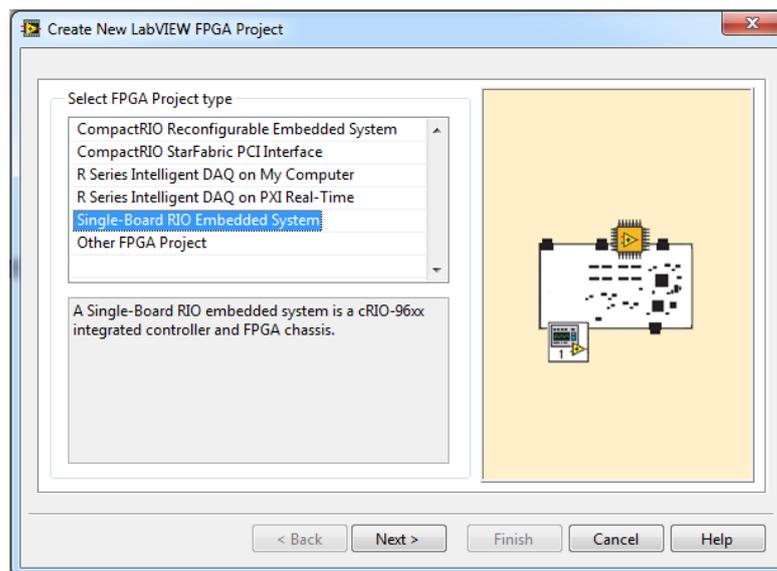


Figura 2.29 Ventana de Creación de un proyecto a partir de FPGA Project. Esta ventana nos abre una serie de opciones para seleccionar la FPGA con la cual vamos a trabajar, en este caso la opción es Single-Board RIO Embedded System.

En la ventana mostrada en la Figura 2.29, se configura el tipo de proyecto que vamos a crear, indicando con qué sistema remoto vamos a trabajar (CompactRIO,

FlexRIO, Single-Board RIO, entre otros), una vez seleccionado el tipo de proyecto es necesario dar clic en Next y aparecerá otra ventana con dos opciones a elegir, Discover existing system y Create New System, la primera opción permite elegir un dispositivo conectado al computador siendo parte de un red o ingresando la dirección IP del mismo y la segunda opción nos permite elegir un dispositivo soportado que aunque no esté conectado y cuyos controladores si se encuentren instalados en LabVIEW.

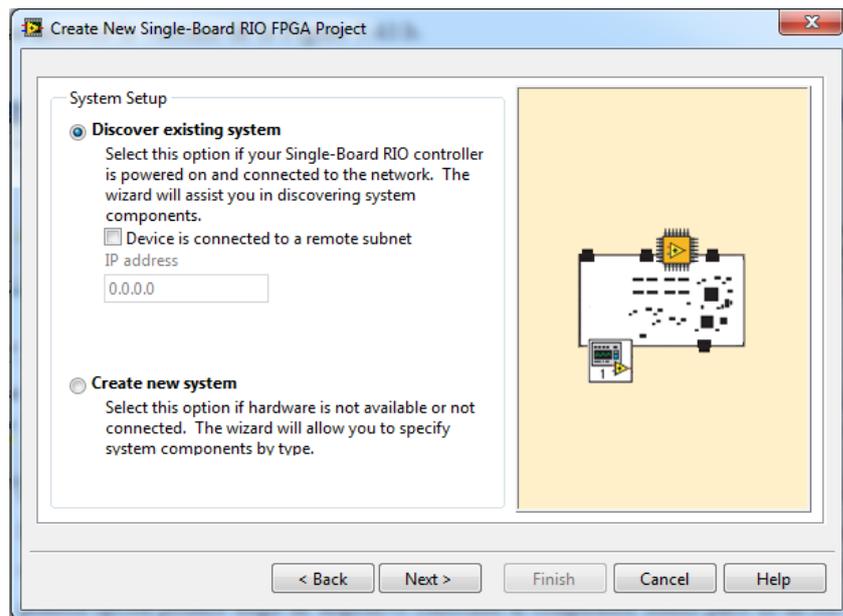


Figura 2.30 Ventana de configuración de Single-Board RIO FPGA Project. Esta ventana nos permite hacer comunicación directa con la FPGA NI sbRIO-9631.

Una vez encontrado el controlador del dispositivo Single-Board (sbRIO), se escoge la serie 9631 de nuestro dispositivo Single-Board, damos clic en Next, apareciendo una ventana la cual permite añadir módulos de la serie C al proyecto compatible con nuestra sbRIO 9631, sino se desea añadir ningún módulo omitimos este paso, damos clic en Next, y nos muestra una última ventana con la apariencia final de nuestro proyecto, damos clic en Finish y se crea el proyecto según como se muestra en la Figura 2.31.

Dando clic derecho en el ítem My Computer permite crear VI Computer el cuál se va a ejecutar sólo en el Sistema Operativo del computador y éste puede interactuar con el VI Real Time que se esté ejecutando sobre el robot móvil, o sea que puede

haber intercambio de información bidireccional entre ambas plataformas. Por lo general el VI que se crea en My Computer se lo usa para visualizar los parámetros del robot, así como el envío de órdenes para su control.

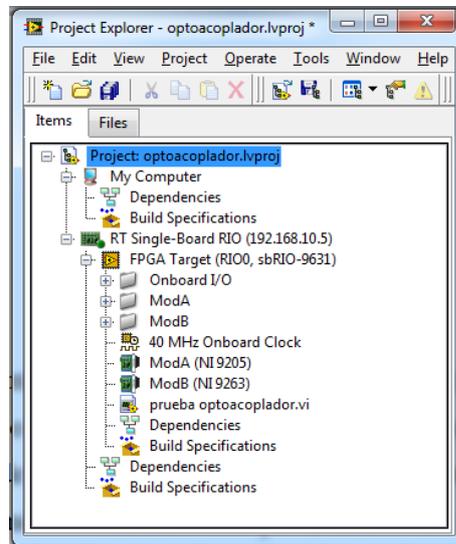


Figura 2.31 Apariencia final del proyecto creado a través de FPGA Project. Esta ventana del nuevo Proyecto creado permite acceder directamente a las opciones que tiene LabVIEW Real Time (RT Single-Board RIO) y LabVIEW FPGA (FPGA Target), según la raíz del proyecto.

Dentro del RT Single-Board RIO, los VIs creados, se ejecutan sobre el procesador de tiempo real de la tarjeta sbRIO, tener en cuenta que la posición en el directorio se empieza a desplegar desde RT Single-Board RIO.

Dentro de Chassis (sbRIO 9631), encontramos otro ítem etiquetado como FPGA Target (sbRIO9631) el cual representa a la FPGA incorporado a la tarjeta sbRIO-9631, aquí también encontramos todos los elementos físicos con los que cuenta la tarjeta y que están conectados al FPGA. Cuando se crea un VI éste se guarda directamente aquí y al momento de compilarlo se ejecuta sobre el chip del FPGA, y se puede acceder a todas las E/S (entradas y salidas) del FPGA. (Figueroa-Palaguachi, 2012)

2.19.1 Proceso y Desarrollo de Compilación de un VI FPGA

El proceso en que se desarrolla la compilación de un VI FPGA puede tomar una cantidad significativa de tiempo, debido a que no se usa el software LabVIEW, los

tiempo de compilación FPGA pueden llegar a durar desde pocos minutos hasta largas y extensas horas, una técnica que se puede realizar es hacer simulaciones sobre el computador para evitar compilaciones innecesarias causadas por errores en la programación.

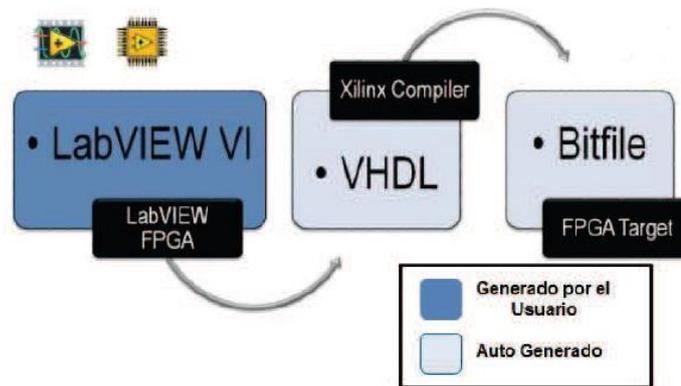


Figura 2.32 Proceso de compilación de un VI FPGA. La figura detalla en bloques como se realiza la compilación el cual LabVIEW FPGA convierte la programación en código VHDL para que pueda ser leído por la tarjeta FPGA. (Figueroa-Palaguachi, 2012)

2.19.2 Inicio de la Compilación de un VI FPGA

Para iniciar la compilación un VI FPGA en LabVIEW, es necesario ir a la barra de menús del Panel Control, aquí tenemos la flecha Run, éste comprueba si hay un archivo válido ya creado para el VI FPGA. Si es válido, inmediatamente empieza a correr el VI FPGA en modo interactivo usando un archivo de bits ya existente en caso de ya haber sido compilado. Si el VI FPGA nunca ha sido compilado o ha sufrido algún cambio desde la última compilación, al dar clic en la flecha Run, comienza el proceso de compilación y una vez finalizado se ejecuta en modo interactivo.

2.19.3 Ventana generada durante el Estado de la Compilación

Una vez finalizado la generación de archivos intermedios sin errores, aparece la ventana Compilation Status. Esta ventana es una interfaz dinámica que permite conocer el estado actual de la compilación, además cuenta con algunas marcas de tiempo (fecha y hora) e información del VI FPGA. Mientras la compilación continúa,

alertas en la parte inferior de la ventana le indicarán cuando un nuevo informe se ha generado según muestra Figura 2.33.

Cuando la compilación ha terminado, LabVIEW genera un informe de compilación, este informe de compilación permite ver si se está conectado al servidor LabVIEW FPGA Compile Server. Después que ha finalizado la compilación de manera exitosa hacemos clic en el botón OK en la ventana Successful Compile Report, LabVIEW descarga el archivo de bits sobre el chip FPGA. EL VI FPGA compilado se ejecuta sobre el chip FPGA y a través de la ventana del panel frontal podemos interactuar con él.

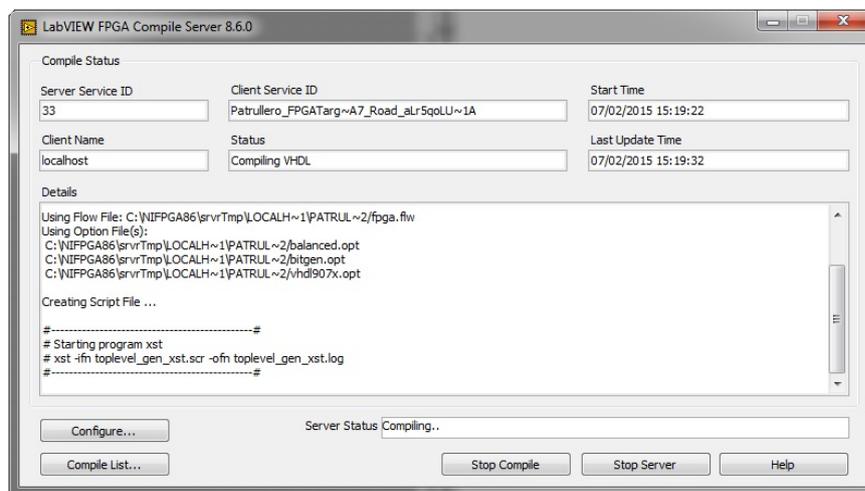


Figura 2.33 Ventana de Estado de Compilación. Esta ventana realiza paso a paso el proceso de compilación de la programación a lenguaje VHDL, proceso que le toma algunos minutos dependiendo de que tan grande sea la programación. (Figueroa-Palaguachi, 2012)

2.20 CREACIÓN DE UN ARCHIVO EJECUTABLE

Al realizar la descarga del programa sobre nuestro robot y al momento de ejecutar el VI Real Time el cual ha sido creado sobre el ítem RT Single-Board RIO, este archivo se almacena en una memoria DRAM (Dynamic Random Access Memory) volátil incorporada en la tarjeta sbRIO. Esto quiere decir que cuando el controlador del sbRIO se apague este archivo se borrará de la memoria DRAM.

Para ello se crea un archivo ejecutable para que nuestro VI Real Time se ejecute automáticamente cuando el Robot Móvil haya sido energizado (robot encendido).

Este archivo ejecutable una vez construido y descargado sobre el robot móvil quedará grabado en la memoria Flash (memoria de almacenamiento no volátil) del controlador del sbRIO, por lo que el contenido de los datos almacenados no se pierde aún si no se encuentra el robot energizado (encendido) y una vez que se energice el robot el archivo guardado se ejecutará de manera automática y con esto conseguimos tener nuestra aplicación robótica totalmente autónoma. (Figuroa-Palaguachi, 2012)

2.20.1 Especificaciones de Construcción

Una especificación de construcción define los VIs, archivos de tipo soportados y las configuraciones para diversas aplicaciones autónomas. Para ello creamos especificaciones de construcción y luego usarla para construir nuestra aplicación.

En esta parte vamos a indicar los pasos a seguir en la creación de una especificación de construcción del proyecto:

Antes de iniciar tomar en cuenta que si, se han creado variables compartidas en el proyecto, es necesario abrir el menú contextual del ítem RT Single-Board RIO y se escoge la opción Disable Autodeploy Variables, ya que sino realizamos esta acción el VI Real Time no se ejecutará en el robot móvil al iniciar el VI en el computador. (Figuroa-Palaguachi, 2012)

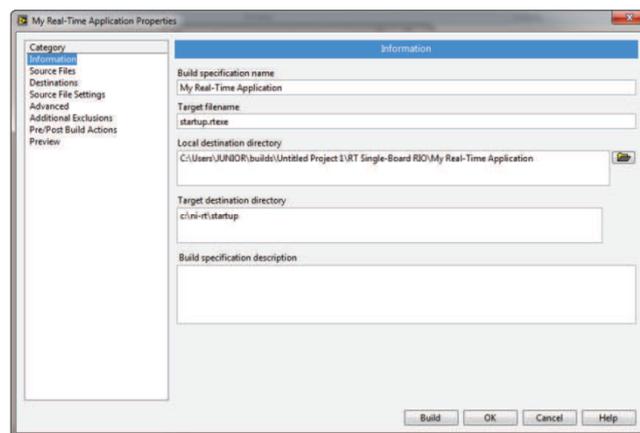


Figura 2.34 Cuadro de diálogo My Real-time Application Properties. Ventana que permite configurar la especificación de construcción de un archivo ejecutable a la tarjeta NI sbRIO 9631.

1. Hacer clic derecho en Build Specifications que se encuentra bajo el ítem RT Single-Bord RIO de la ventana del Explorador de Proyecto, seleccionar New >>

Real-Time Application del menú contextual para abrir el cuadro de diálogo Real-Time Application Properties. Si hemos creado un proyecto de un target basado en Window, hacemos clic derecho en Build Specifications en My Computer y seleccionar New >> Application en el menú contextual.

2. En la ventana My Real-Time Application Properties dar clic sobre el cuadro de texto Build Specification Name, escribir un nombre para la especificación de construcción, si el nombre es demasiado largo inmediatamente dará un error. Este nombre aparecerá bajo el ítem Build Specification del target de tiempo real. Dejar las otras opciones y rutas con sus valores predeterminados.
3. El cuadro de texto Target Filename se escribe un nombre para el archivo ejecutable del VI Real-Time, por defecto y como valor predeterminado ya contiene el nombre de startup.rtexe, pero se puede modificar.
4. Luego hacer clic sobre la opción Source Files desde la lista de Category para desplegar la página Source Files. Desde Project files, seleccionar el VI Real-Time.vi, dar clic en la flecha derecha para mover el VI en la lista Startup Vis, estos se ejecutan cuando inicia la aplicación en el robot móvil.
5. Dar clic en OK para aceptar la configuraciones realizadas y cerrar el cuadro de diálogo Real-Time Application Properties. El nombre de la especificación de construcción aparece en el Explorador de Proyectos debajo de Build Specifications.
6. Luego seleccionar Project >> Save Project desde la ventana del Explorador de Proyectos para guardar el proyecto y todas las configuraciones hechas en las especificaciones de construcción.

2.20.2 Ejecución del Archivo Ejecutable

Una vez creada la especificación de construcción, se puede construir y ejecutar una aplicación robótica sobre el controlador sbRIO del robot móvil. También se puede configurar la aplicación para que se ejecute una vez energizado el robot móvil.

Realizar los siguientes pasos para construir y ejecutar una aplicación robótica sobre el controlador de la tarjeta sbRIO-9631.

Hacer clic derecho en la especificación de construcción en el nombre dado, seleccionamos Build en el menú contextual para construir la aplicación. El constructor de aplicación usa las configuraciones definidas en la especificación de construcción. Cuando la construcción sea completada, es necesario dar clic en el botón Done para cerrar el cuadro de diálogo Build Status.

Luego dar clic derecho en la especificación de construcción en el nombre dado y seleccionamos Set as startup en el menú contextual. Después desde el mismo menú contextual se selecciona Run as Startup, con lo cual se establece la aplicación VI Real Time como una aplicación de inicio (archivo ejecutable). Luego de implementar la aplicación en el controlador sbRIO, inmediatamente se abre una ventana de aviso donde nos solicita que se reinicie dicho controlador, es necesario pulsar OK y se apaga, luego volver a encender el robot móvil, se espera un minuto para que el robot se reinicie y está listo.

Después de reiniciar el robot móvil, la aplicación generada a partir de la especificación de construcción se inicia de manera automática y empieza a correr el programa implementado en el VI Real Time.

Si vamos a continuar con la edición del *VI Real Time*, es posible que no se desee que esta aplicación sea ejecutada cada vez que encendemos el robot móvil. Para anular esta configuración de la aplicación de inicio, se da clic derecho en la especificación de construcción en el nombre dado y se selecciona Unset as startup y luego Deploy desde el menú contextual, tomando en consideración que el robot está conectado en red con la computadora. (Figuroa-Palaguachi, 2012)

CAPITULO 3. DISEÑO Y CONSTRUCCIÓN DEL ROBOT MÓVIL

El diseño de la estructura del robot móvil tiene una gran importancia dentro del desarrollo de nuestro proyecto, debido a que irán asentados todos los componentes que conforman el robot móvil, como son las tarjetas electrónicas, los sensores, los motores, la batería, el router, entre otros.

El diseño implementado en el robot móvil con tecnología FPGA, es similar en apariencia al Robot Móvil LabVIEW Robotics SbRIO Starter Kit de National Instruments, la diferencia radica en que su estructura es diseñada 100% por los autores además de los elementos que se usaron para su construcción, implementando únicamente la tarjeta FPGA NI SbRIO-9631 (descrito anteriormente), como tarjeta de control y diseño para la programación en LabVIEW y con ello la ejecución de las guías de laboratorio aplicadas más adelante.

3.1 DISEÑO DE HARDWARE

Para el diseño del hardware del robot móvil, es necesario indicar una serie de requerimientos los cuales permitieron su desarrollo paso a paso al momento de su ensamblaje.

3.1.1 Requerimientos de la Estructura del Robot Móvil

Para el desarrollo de su estructura es necesario considerar una serie de factores que van a influenciar el desempeño del robot móvil. Los requerimientos básicos que se necesitan para su la construcción de su estructura está detallado a continuación:

- El robot debe tener una buena resistencia para poder soportar todo el peso de los componentes que estarán encima del mismo.
- La estructura del robot tiene que ser de un material ligero para que pueda tener mayor movilidad y de esta manera facilitar el trabajo que realiza el robot.
- El tamaño de la estructura del robot móvil dependerá de los elementos que tendrá instalados, tanto en el interior como en su exterior.
- Los motores tienen que tener un torque lo suficientemente fuerte para poder vencer el estado de inercia del robot móvil.
- Las ruedas tienen que ser material de caucho para generar fricción y evitar que el robot resbale.

- La comunicación inalámbrica tiene que ser generada a través de un router inalámbrico que permita una comunicación clara y muy eficaz para evitar errores durante el intercambio de datos entre el computador y el robot móvil.
- Las baterías tienen que tener un amperaje lo suficiente alto para soportar el consumo de energía del robot móvil por un lapso de por lo mínimo 30 minutos de manera continua.

3.1.2 Diseño de la Estructura del Robot Móvil

Para el diseño de la estructura del robot móvil se diseñó una plataforma que se adaptó a todos los dispositivos con que nuestro robot va a trabajar, desde las tarjetas driver para los motores hasta la cámara inalámbrica, para ello se consideró como material de la estructura un material ligero como es el aluminio. Este material al ser blando y maleable nos permite un fácil moldeado al diseño que queremos.

En la Figura 3.1 se muestra el diseño de la estructura del robot móvil junto con las dimensiones detalladas, los cuales permiten tener una referencia exacta de las medidas que va a tener el robot móvil, tomando en cuenta la posición y colocación de los sensores que van a ser instalados en el mismo.

Para la parte motriz del robot móvil, se implementó en el diseño una configuración diferencial que permite controlar los giros derecha e izquierda, a través de la diferencia de velocidad aplicada a cada uno de los motores. Su control tanto en la parte motriz como en todo el sistema dependerá directamente de la tarjeta FPGA NI sbRIO-9631, el cual será el cerebro de todo el robot móvil y que de él dependerá todo el funcionamiento que desarrolle en cada una de las prácticas que serán diseñadas más adelante.

A continuación se muestran ciertas ventajas de la tarjeta FPGA NI sbRIO-9631 al trabajar con el robot móvil.

- Fácil conexión de sensores y actuadores de uso robótico.
- Fácil comunicación inalámbrica usando un router.
- Fácil diseño, programación y control gracias al entorno gráfico LabVIEW.
- La toma de decisiones se da en tiempo real, gracias a LabVIEW Real Time y a LabVIEW FPGA.

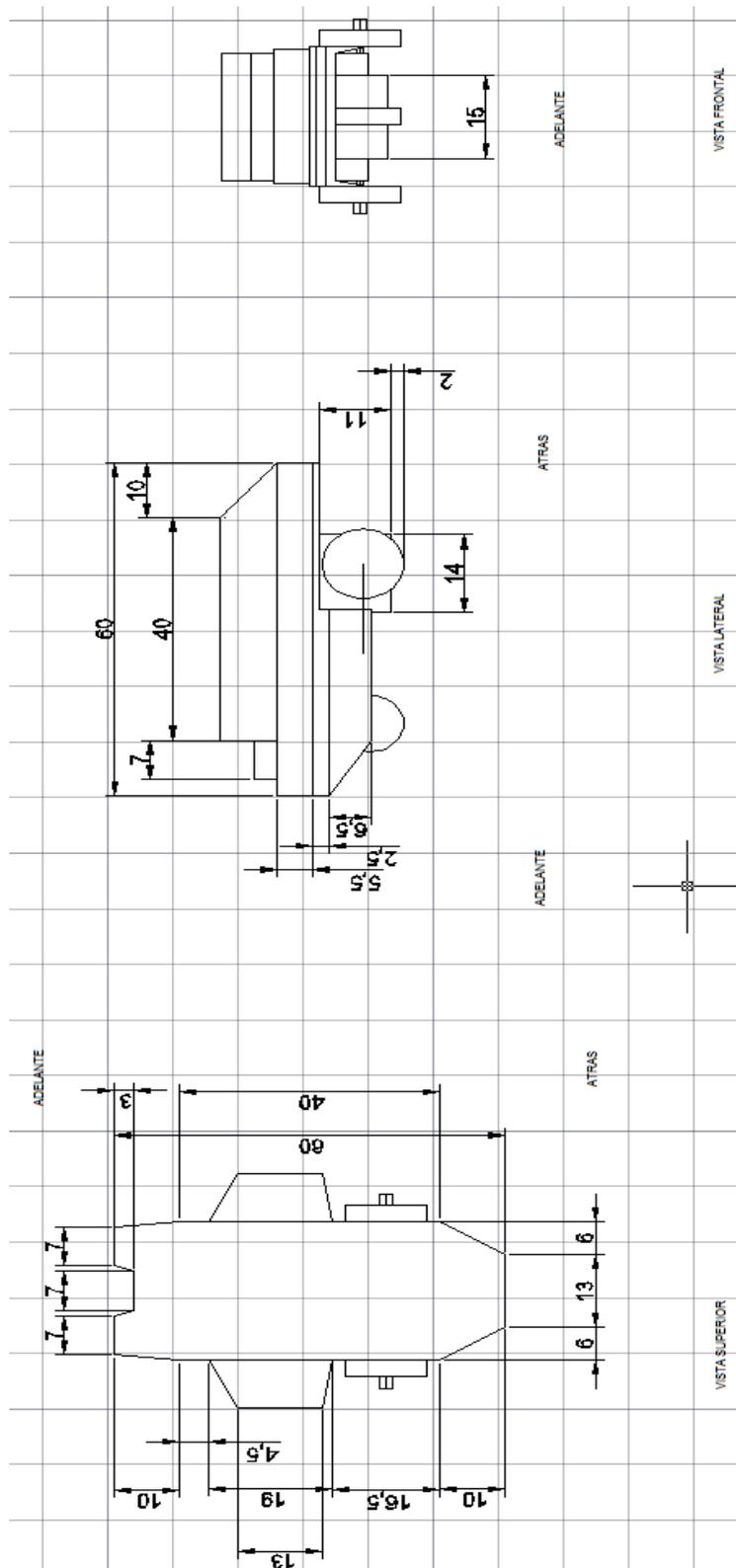


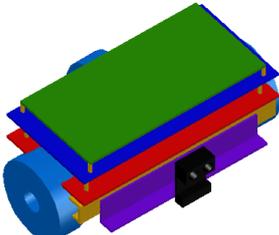
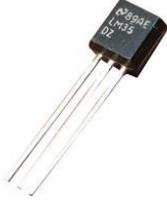
Figura 3.1 Dimensiones de la Estructura del Robot Móvil. Imagen que detalla las medidas del robot móvil visto desde la parte superior, lateral y frontal, medidas que fueron tomadas para la construcción del robot móvil.

3.2 DESCRIPCIÓN DE COMPONENTES QUE CONFORMAN EL ROBOT MÓVIL

En la siguiente tabla se detalla cada uno de los elementos que conforman nuestro robot móvil con una breve descripción de los mismo.

Tabla 1.

Dispositivos Principales del Robot Móvil

Dispositivos del Robot Móvil		
Tipo de dispositivos	Descripción	Figura
Estructura	Material Aluminio liso Grosor: 3mm Acrílico liso transparente Grosor: 2cm	
Motores	2 Motores 12VDC Gran fuerza de torque Velocidad Angular: 85 rpm	
Sensores	10 Sensores de ultrasonido Mide distancias de 2 a 450 cm.	
	Sensor de Humedad Mide humedad relativa (%RH) Salida analógica	
	Sensor de temperatura Rango de medición -55°C a 150°C	

	Sensor de luz por fotorresistencia Salida analógica/digital	
	02 Encoders tipo incrementales 360 pulsos por revolución	
Tarjeta de Control	Tarjeta FPGA Single Board sbRIO-9631 de National Instruments	
Caméra Inalámbrica	Cámara inalámbrica de red IP 30 imágenes por segundo en una resolución de 800x600	
Fuente de energía	3 Baterías Recargables LIPO 11.1V 3.3mAh	

Nota: Se puede observar en la tabla un breve resumen los dispositivos principales que fueron adquiridos e implementados para la construcción del Robot Móvil.

Como ya se ha mencionado el robot móvil, es diseñado con un hardware que se basa en una plataforma FPGA sbRIO-9631 de National Instruments, el cual está montado en la parte superior de nuestro robot móvil, para su fácil percepción además de su fácil acceso y manipulación.

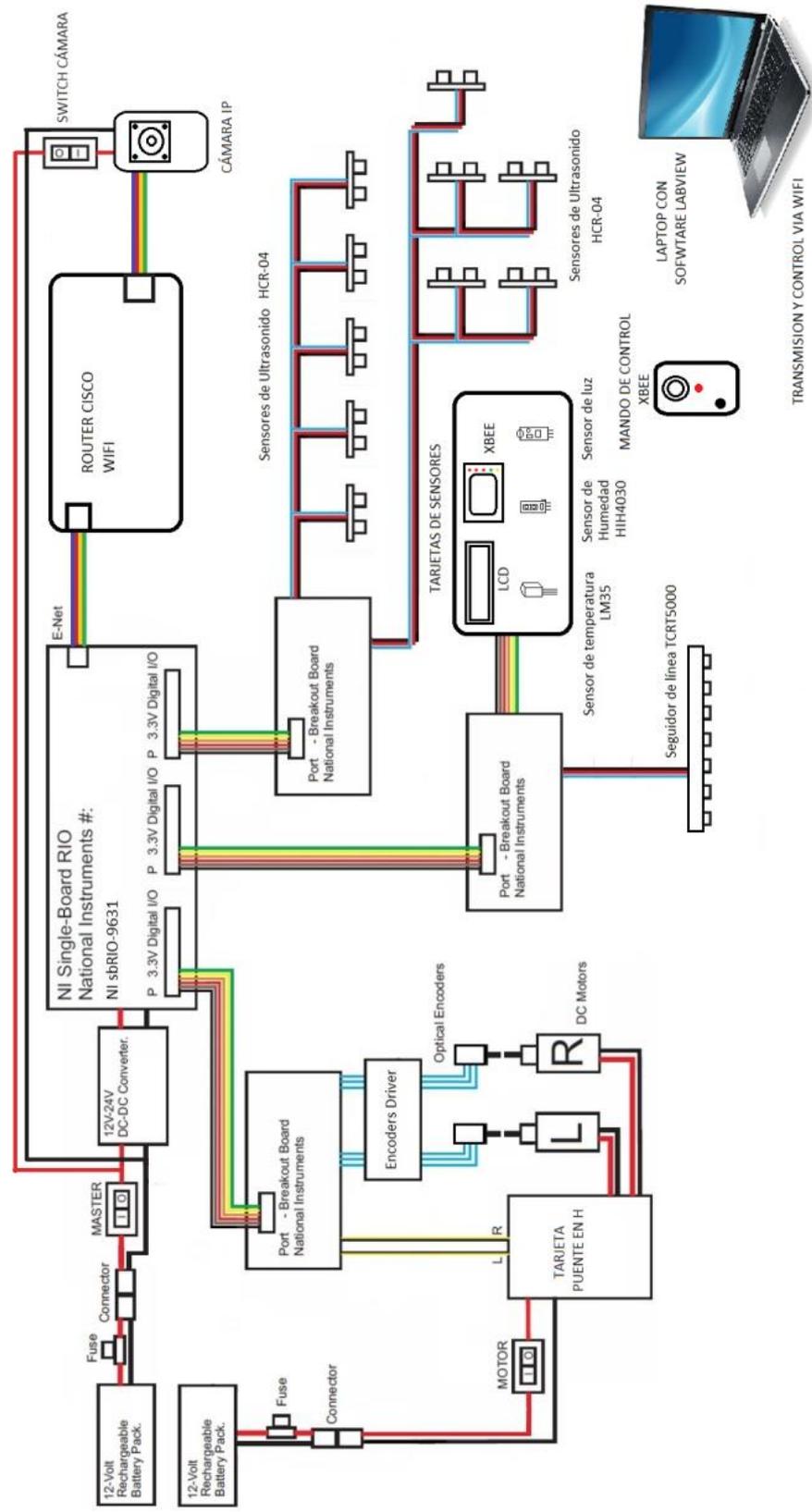


Figura 3.2 Diagrama en bloques del Robot Móvil. La figura detalla en forma de bloques las conexiones de cada una de las partes que forman el robot móvil.

En el diagrama en bloques del Robot Móvil se puede observar todas las conexiones de todos los dispositivos y elementos que conforman nuestro robot. Para detallar mejor las conexiones del robot describiremos cada uno de los módulos.

El robot móvil cuenta con dos fuentes de alimentación (baterías), las cuales se conectan por separado, una de ellas alimentará la tarjeta NI sbRIO-9631 junto con los sensores, y la otra batería alimentará el sistema de potencia, en este caso a los motores, cada uno de ellos tendrán un Switch de activación (ON/OFF) para el encendido del robot móvil.

Las baterías recargables instaladas en el robot móvil son de tipo LIPO (Polímeros de Litio) de 11.1V a 3.3Ah, las cuales permiten un mayor tiempo de duración de carga en el momento del funcionamiento. El robot móvil cuenta con 03 baterías recargables, una de ellas se maneja directamente con las tarjeta SbRIO-9631 y sus demás componentes (sensores, encoders, LCD, entre otros) y las otras dos baterías se encuentran instaladas cada una de ellas a los motores DC quienes al tener un buen torque, requieren de una mayor carga tanto en arranque como al momento de trabajar.



Figura 3.3 Batería LIPO instalada en el robot móvil. Batería Lipo de 3 celdas a 11.1V/3300mAh. DIVINEO ICS. (2015). Batería Lipo 11.1V 3300mAh. Obtenido de http://www.divineo.es/product_info.php?products_id=16657&osCsid=b85400235ce994485ee0e4c85046a737

3.3 INSTALACIÓN Y MONTAJE DE LOS COMPONENTES

Durante el montaje de cada uno los componentes es necesario tener en cuenta que el diseño del robot móvil está específicamente diseñado para soportar el peso total en todos los elementos y componentes que lo conforman, cualquier peso adicional

podría influir en el rendimiento del robot móvil y el tiempo de duración de las baterías.

3.3.1 Instalación de los motores DC y las ruedas

Al tener un robot móvil con una estructura diseñada para una configuración diferencial, es necesario el uso de dos motores DC, cada uno de ellos fue adaptado para el trabajo al cual será usado.

Estos motores DC fueron adquiridos directamente de los eleva-vidrios de los autos, por tener un gran torque durante el arranque y un buen rendimiento, el cual fueron adaptados a las ruedas del robot móvil.



Figura 3.4 Motores DC junto con las ruedas. Adaptación realizada a los motores DC para que se acople a las ruedas del robot móvil.

Uno de los factores importantes en nuestra configuración diferencial de nuestro robot, es la infaltable rueda LOCA, esta rueda es adaptada directamente entre nuestros dos motores para que mantenga un buen desplazamiento y se adapte de manera inmediata a los giros que pueda realizar nuestro robot móvil. (Ver Figura 3.5)

3.3.2 Instalación de los Encoders

Como información adicional, los encoders escogidos para el robot móvil generan 360 pulsos por revolución lo cual permite tener una medida exacta del ángulo recorrido por la rueda y de esa forma tener una buena lectura y control de velocidad de los motores DC.

Como se observa en la Figura 3.6 los motores son directamente acoplado al eje principal del motor DC.



Figura 3.5 Rueda LOCA del robot móvil. Rueda que se acopla al direccionamiento que toman los motores DC.



Figura 3.6 Acople del Encoder con el motor DC. Figura del motor conectado al encoder a través de acople de silicon para evitar roces en el eje del motor.

3.3.3 Instalación de Tarjetas Drivers

Los motores DC del robot móvil se encuentran conectados directamente a una tarjeta driver que permite el control de velocidad y el control del sentido de giro de los mismos. Esta tarjeta se encuentra formada por dos puentes en H, tarjeta diseñada por los autores con el fin de controlar los movimientos del robot.

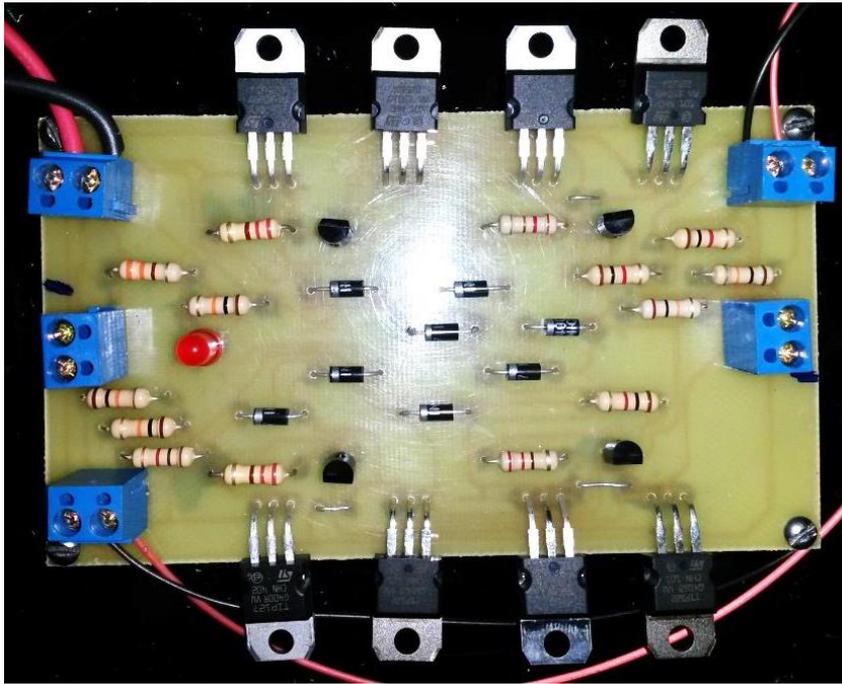


Figura 3.7 Vista superior Tarjeta driver para los motores DC. Tarjeta diseñada con 2 puente en h para controlar el giro de dos motores DC.

Otra de las tarjetas drivers instalada, es la de los encoders, ésta se encuentra ubicada también en la parte inferior del robot y conectada directamente a la salida de cada uno de los encoders, a través de ésta tarjeta nos permitirá generar una serie de pulsos que irán directamente conectado a la tarjeta sbRIO. (Ver Figura 3.12)

Esta tarjeta está diseñada en función del voltaje con el que trabaja la tarjeta SbRIO, el cual permite la conversión de pulsos de +12VDC con los que trabaja el encoder a +5VDC, voltaje máximo permitido para la tarjeta NI sbRIO-9631, si el voltaje aplicado supera los valores permisibles puede llegar a generar daños severos en la tarjeta.

3.3.4 Instalación de Tarjeta NI sbRIO-9631

La tarjeta FPGA SbRIO-9631, se instaló directamente en la parte superior del robot el cual permite una mejor visión de la misma por parte del usuario, junto con las tarjetas racks que permiten un acceso directo a los conectores de 50 pines de E/S de la tarjeta sbRIO. (Ver Figura 3.9)

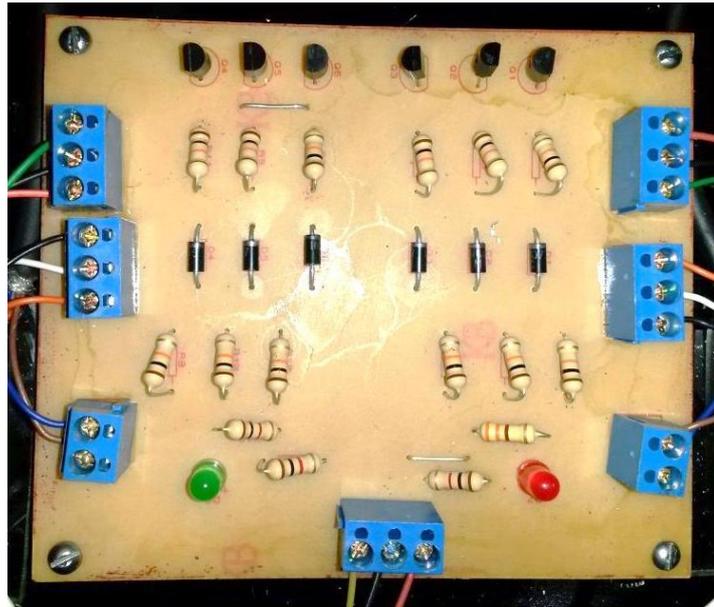


Figura 3.8 Vista superior de la tarjeta driver para los encoders. Tarjeta diseñada para recibir los +12Vcc y enviar +5Vcc a la tarjeta NI sbRIO-9631.

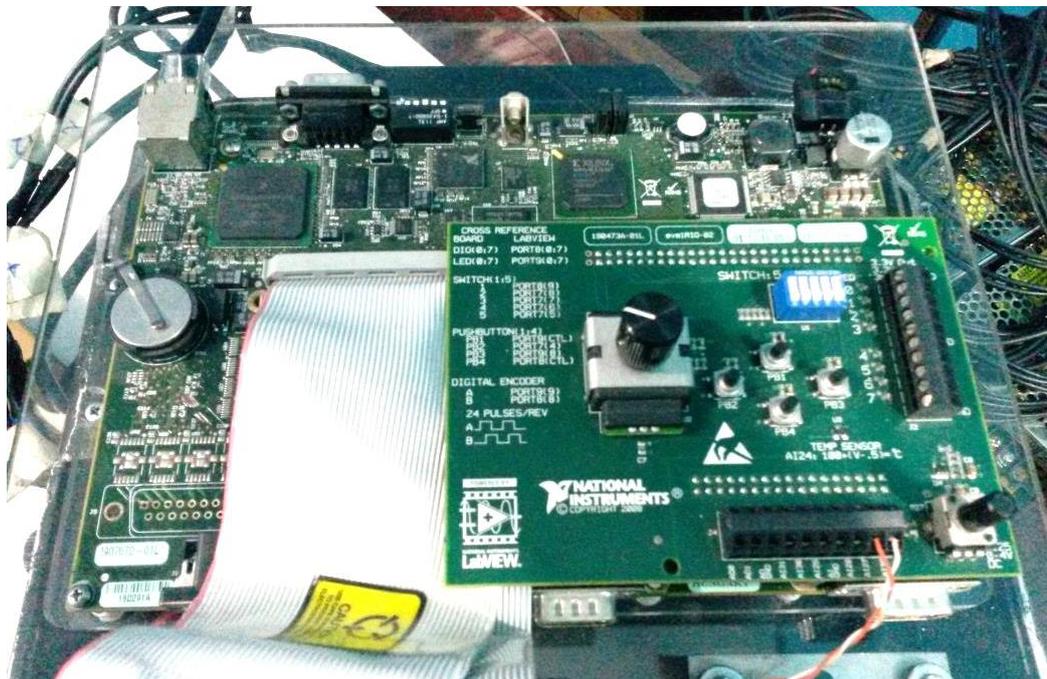


Figura 3.9 Vista superior del robot móvil apreciando la tarjeta SbRIO-9631 de National Instruments. Tarjeta instalada sobre el robot móvil para controlar todo el sistema en Tiempo Real.

3.3.5 Instalación de los Sensores de Ultrasonido

Los sensores de ultrasonido fueron ubicados estratégicamente para las guías de laboratorio a realizarse, en el cual influyó tanto en su diseño como en la estructura del mismo.

Los sensores de ultrasonido se encuentran ubicados en la parte de frontal, uno de frente y dos sensores de ultrasonido con una inclinación de 45°, a los costados (derecho e izquierdo), para una fácil detección de cualquier objeto y uno en la parte de posterior, así de esta manera permite que nuestro robot móvil detecte cualquier obstáculo abarcando gran parte del contorno de su área, evitando cualquier roce de algún obstáculo. (Ver Guía de Laboratorio 6)

Además de los sensores de ultrasonido ubicados en la parte superior del robot móvil (ver Figura 3.11), también se instalaron en la parte inferior (ver Figura 3.10) permitiendo que el robot evite caerse de cualquier precipicio o barranco por el cual éste se desplace. (Ver Guía de Laboratorio 7)

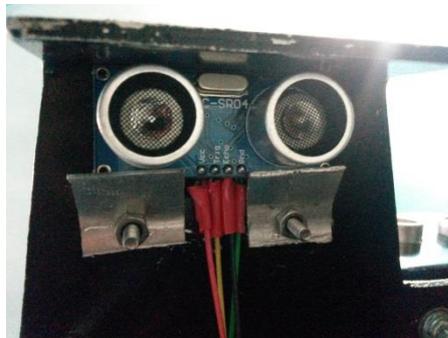


Figura 3.10 Vista de los Sensores de Ultrasonido HC-SR04 instalados en el robot móvil. Instalación del sensor en la parte inferior del robot móvil.

3.3.6 Instalación de Tarjeta de Adquisición de Datos

La tarjeta de adquisición de datos se encuentra ubicada en la parte superior del robot móvil, ya que a través de esta tarjeta éste logra una mejor lectura del medio a través de los sensores de temperatura, humedad y luminosidad, sensores que serán empleados en las diversas guías de laboratorio usadas durante este proyecto. Esta tarjeta se alimenta de +5VDC, y manejará una pantalla LCD para la muestra de datos y un módulo especial X-BEE para la comunicación inalámbrica.

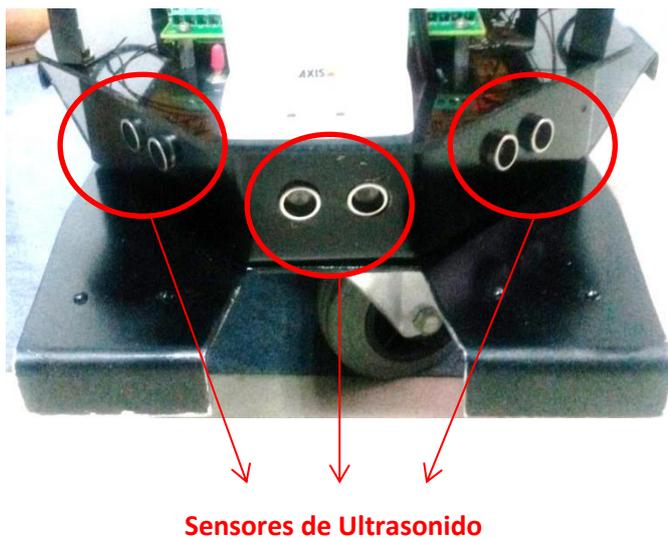


Figura 3.11 Sensores de Ultrasonido HC-SR04 instalados en el robot móvil. Instalación de los sensores en la parte frontal del robot móvil, para la toma de datos.

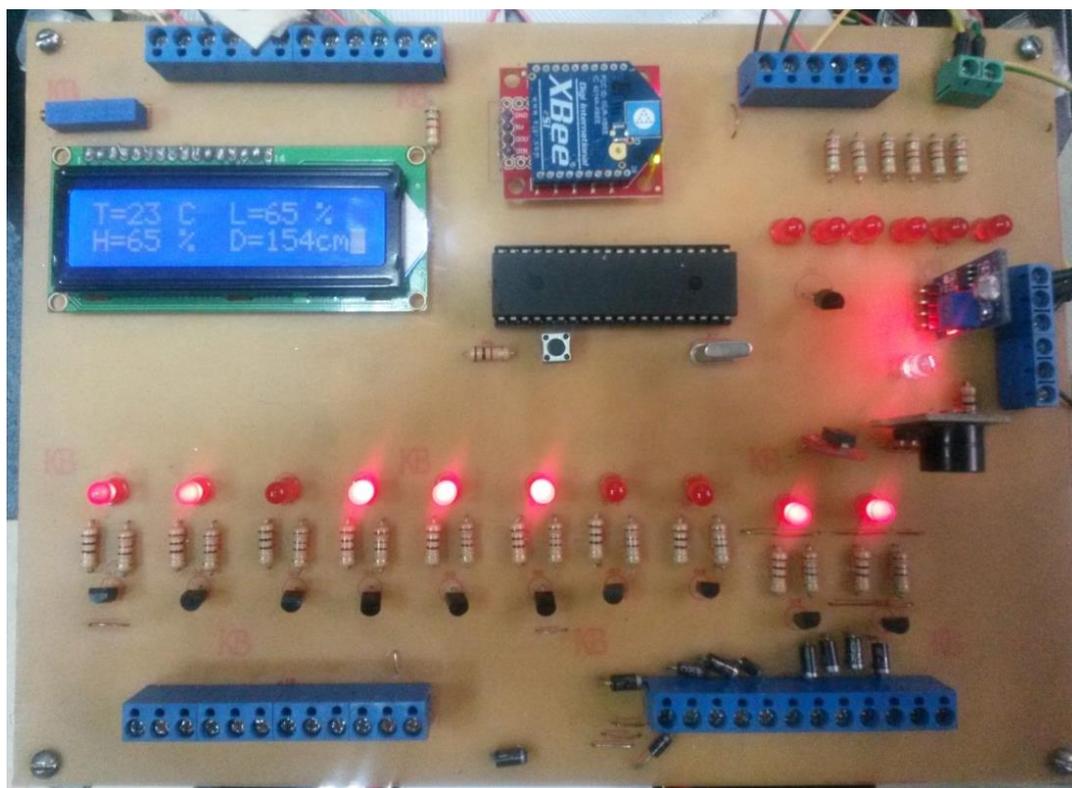


Figura 3.12 Tarjeta de adquisición de datos. Tarjeta diseñada para adquisición y muestreo de datos de sensores varios, conexión inalámbrica y control seguidor de línea.

3.4 IMÁGENES DEL ROBOT YA ENSAMBLADO



Figura 3.13 Parte lateral del robot móvil. Robot Móvil ya ensamblado con todos los dispositivos y tarjetas conectados y listo para funcionar.



Figura 3.14 Parte lateral del robot móvil. Robot Móvil ya ensamblado con todos los dispositivos y tarjetas conectados y listo para funcionar.

3.5 PRUEBAS DE FUNCIONAMIENTO DEL ROBOT MÓVIL

Una vez ensamblado el robot móvil se procede a realizar las respectivas pruebas de funcionamiento para eliminar errores de operación durante su funcionamiento.

Para el desarrollo de estas pruebas vamos a trabajar con un equipo de medición como es el Multímetro que será nuestro mejor aliado para estas pruebas.



Figura 3.15 Trabajando con el Robot Móvil. Pruebas de funcionamiento realizadas por los autores para verificar el correcto funcionamiento del Robot Móvil.

3.5.1 Primera prueba. Medición de Continuidad entre líneas que suministran Voltaje

Una de las primera pruebas y necesarias es la verificación de continuidad entre líneas de voltaje con el fin de evitar que líneas que suministran voltaje (+ y GND) puedan conectarse o toparse entre sí y generen un corto en el sistema, adicional a ello también nos permite verificar si existe alguna línea abierta (no conectada) en el sistema.

3.5.2 Segunda prueba. Verificación de los fusibles

Para esta segunda prueba comprobaremos el buen estado de los fusibles de protección contra alguna sobrecarga que pueda llegar a generarse en el sistema mientras se encuentre en funcionamiento.

3.5.3 Tercera prueba. Suministro y Medición de Voltajes en distintos puntos del Robot Móvil

Al suministrar voltaje al Robot móvil es necesario comprobar si realmente le está llegando el voltaje adecuado a cada dispositivo, si existe una pérdida o caída de voltaje puede influir al mal funcionamiento del robot móvil durante su operación.

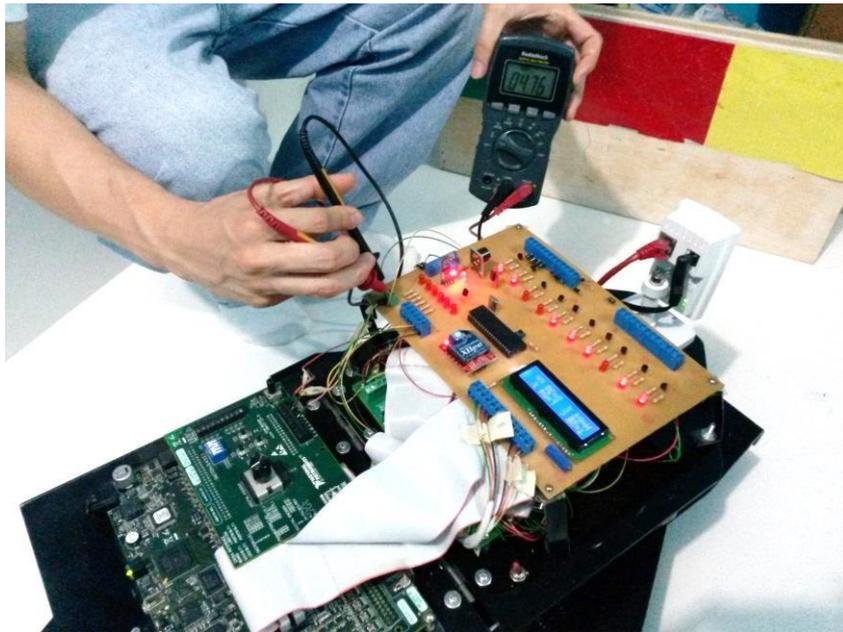


Figura 3.16 Pruebas de medición de voltaje en distintos puntos del Robot Móvil. Medición de voltajes en la tarjeta de adquisición de datos.

3.5.4 Cuarta prueba. Verificación de señal WIFI

Esta prueba es importante al momento de empezar a configurar todos los dispositivos inalámbricos. Verificamos si el dispositivo router que se encuentra conectado al robot móvil emite una señal WIFI. Esto lo comprobamos directamente en la PC, en la opción Centro de Redes y Recursos. Ver Figura 3.17

3.5.5 Quinta prueba. Comunicación de la Red

En esta prueba se hace comunicación con el robot móvil a través de la señal WIFI directamente con el router conectado con la tarjeta NI sbRIO-9631. Si no existe algún problema en la comunicación tendremos una prueba exitosa y listo para empezar a trabajar.



Figura 3.17 Señal WIFI “PezPequeno” de la red inalámbrica vista desde el computador. Visualización de la red WIFI del router indicando que el router está funcionando correctamente.

CAPÍTULO 4. DESARROLLO DE GUÍAS DE LABORATORIO

4.1 DESCRIPCIÓN DEL CAPÍTULO

En este capítulo se complementa todo lo aprendido de la tarjeta FPGA NI sbRIO-9631, en el diseño de múltiples prácticas a las que llamaremos guías de laboratorio trabajando directamente con el robot móvil, que van desde lo más sencillo, desde cómo realizar la comunicación de la tarjeta NI sbRIO-9631 con el PC, hasta lo más complejo, basándonos en el uso de elementos y sensores de uso general para el complemento de cada una de las prácticas. A continuación un breve detalle de las guías de laboratorio desarrolladas en este capítulo.

Guía de Laboratorio 1: Comunicación con LabVIEW.

Objetivo: Comunicación paso a paso de la tarjeta NI sbRIO-9631 con LabVIEW.

Guía de Laboratorio 2: Adquisición de Variables del Entorno.

Objetivo: Conexión de cada uno de los sensores usados con la tarjeta NI sbRIO-9631 de National Instruments.

Guía de Laboratorio 3: Comunicación con Dispositivos Inalámbricos

Objetivo: Comunicación inalámbrica y control del robot a través de un control externo.

Guía de Laboratorio 4: Implementación de un Robot Patrullero

Objetivo: Desplazamiento del robot móvil en forma de patrulla ida y vuelta.

Guía de Laboratorio 5: Implementación de un Robot Anti-caídas

Objetivo: Desplazamiento del robot móvil sin caerse de su base.

Guía de Laboratorio 6: Implementación de un Robot Seguidor de Línea

Objetivo: Desplazamiento del robot móvil a través de una línea guía.

Guía de Laboratorio 7: Implementación de un Robot Navegador

Objetivo: Desplazamiento del robot móvil a través de una habitación buscando la salida.

Guía de Laboratorio 8: Implementación de un Robot de Rescate

Objetivo: Búsqueda del robot móvil objetos de un color predeterminado simulando personas, enviando mensajes de rescate.

4.2 GUIA DE LABORATORIO 1

1. TEMA:

COMUNICACIÓN CON LA TARJETA FPGA NI SBRIO-9631 Y LABVIEW.

2. OBJETIVOS:

2.1 GENERAL

- Realizar paso a paso la interacción entre LabVIEW y el robot autónomo a controlar, demostrando la forma en que se debe operar la tarjeta FPGA NI sbRIO-9631 en el momento de hacer comunicación a través del computador con el programa LabVIEW.

2.2 ESPECÍFICOS

- Aprender a configurar la tarjeta FPGA, asignando una nueva dirección IP a la tarjeta NI sbRIO-9631, para lograr la comunicación con el computador y LabVIEW.
- Desarrollar diferentes formas de comunicación con la tarjeta NI sbRIO-9631, comunicación Ethernet cableada o de forma inalámbrica a través de un router.
- Conocer paso a paso la forma de crear una red inalámbrica para la comunicación entre el computador y el robot autónomo.

3. MARCO TEÓRICO:

3.1 Tarjeta FPGA NI sbRIO-9631

Este dispositivo embebido de control y adquisición de datos integra un procesador en tiempo real, un arreglo de compuerta programable en campo (FPGA) reconfigurable por el usuario y E/S en una sola tarjeta de circuito impreso.

Características:

- Procesador de 266MHz, 128 MB de almacenamiento no volátil, 64 MB DRAM para control y análisis determinísticos.
- FPGA de 1M de compuertas con tecnología de E/S reconfigurables (RIO) para temporización personalizada, procesamiento de línea y control.

- 110 líneas DIO de 3.3V (tolerancia de 5V/TTL), 32 entradas analógicas de 16 bits, cuatro salidas analógicas de 16 bits.
- Puerto Ethernet 10/100BASE-T y puerto serial RS232, entrada de suministro de 19-30VDC.
- Rango de temperatura de operación de -20 a 55°C.

La tarjeta sbRIO-9631 está diseñado para ser fácilmente embebido en aplicaciones de alto volumen que requieren flexibilidad, fiabilidad y alto rendimiento.

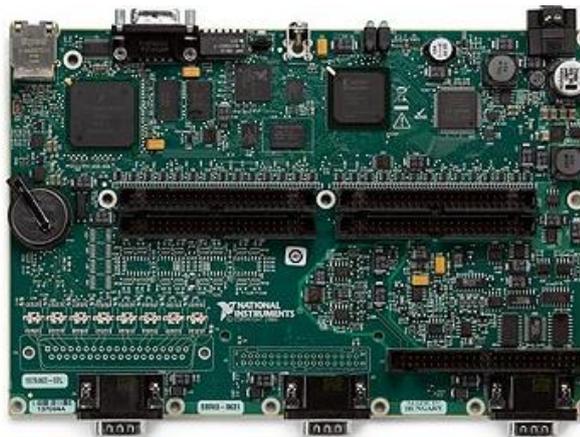


Figura 4.1 Tarjeta NI sbRIO-9631. Vista superior de la Tarjeta FPGA NI Single-Board de la familia RIO. Por National Instruments. (2015). NI sbRIO-9631 Dispositivos Embebidos con AI, AO, DIO, FPGA de 1M de Compuertas. Obtenido de http://sine.ni.com/nips/cds/pages/image?imagepath=/images/products/us/sbrio9631_1.jpg&title=NI%20sbRIO9631&oracleLang=esa

4. DESARROLLO

Para el desarrollo de esta práctica se comienza mencionando que una de las ventajas que posee el robot móvil, para ser más precisos la tarjeta FPGA sbRIO-9631, es su puerto de comunicación Ethernet lo cual nos permite comunicarnos con la PC de dos diferentes formas: de manera cableada y de manera inalámbrica.

4.1 Comunicación Cableada

La comunicación cableada es realizada mediante Cable Ethernet, lo primero será configurarle una dirección IP al robot móvil, en este caso a la tarjeta FPGA NI sbRIO-9631, para lo cual se abre el programa Measurement and Automation Explorer MAX, desplegamos la opción Remote System e inmediatamente nos

aparece el controlador NI-sbRIO9631 de la tarjeta FPGA, en ella podemos ingresar a la pestaña System Settings y observamos la dirección IP del robot móvil, en este caso es 192.168.10.5.

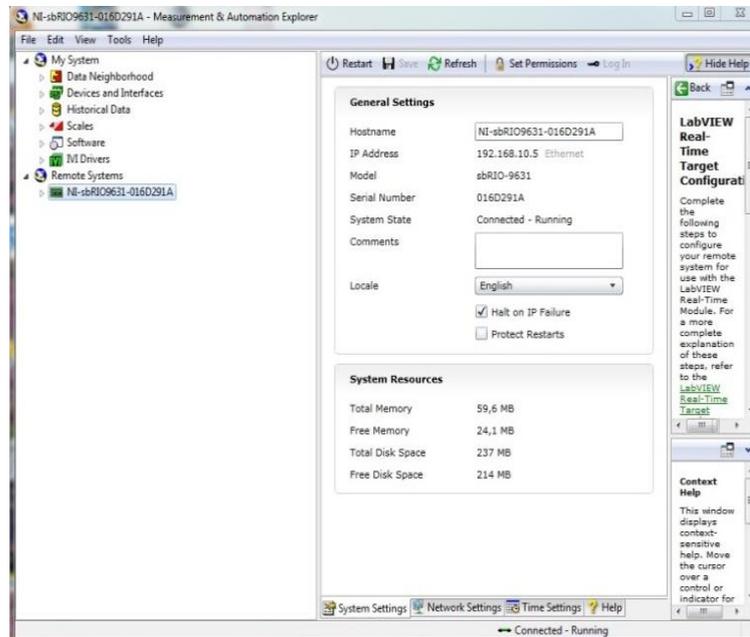


Figura 4.2 Ventana de configuración de la IP del robot móvil. Ventana del MAX donde permite hacer comunicación con la tarjeta NI sbRIO-9631 y configurar la tarjeta.

Si se desea modificar la dirección IP del robot, vamos a la opción *Network Settings* y se realizan los siguientes cambios: la dirección IPv4 va como estática, se coloca la máscara de subred 255.255.255.0. Una vez realizadas las configuraciones es necesario dar clic sobre la opción Save para que los cambios realizados queden guardados.

Una vez configurado el robot móvil, configuramos la dirección IP del computador, vamos a Inicio > Panel de Control > Centro de redes y recursos compartidos, se elige la opción la opción Cambiar configuración del adaptador, Protocolo de Internet versión 4 (TCP/IPv4, y por último en Propiedades, luego ingresamos la dirección IP correspondiente para comunicarnos en red con el robot móvil, en este caso la dirección IP será 192.168.10.10. Una vez concluido la configuración se da clic en Aceptar, para guardar los cambios realizados.

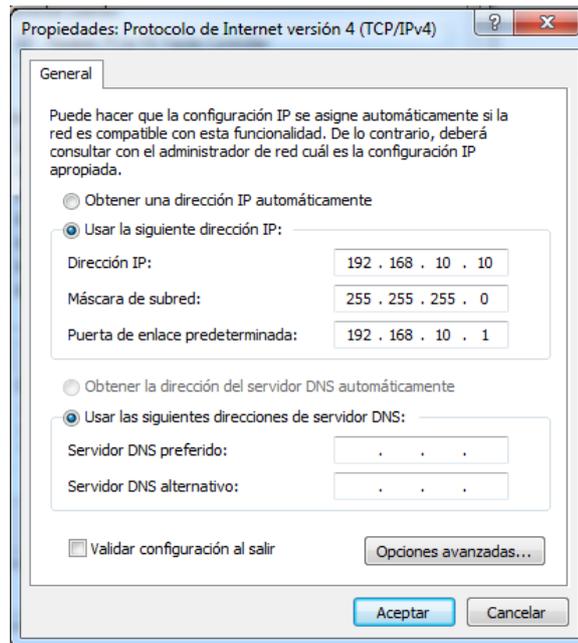


Figura 4.3 Ventana de propiedades de Protocolo de Internet IPv4. En esta ventana podemos configurar la dirección de red de la PC o también se puede configurar de forma automática.

Una vez terminada la configuración tanto el computador como el robot móvil están listos para conectarse en red (TCP/IP) a través de un cable Ethernet.

4.2 Comunicación Inalámbrica

Para el desarrollo de la comunicación inalámbrica es necesario que el robot móvil se acople con un router inalámbrico, esto se logra mediante la configuración del equipo en conjunto con la computadora.

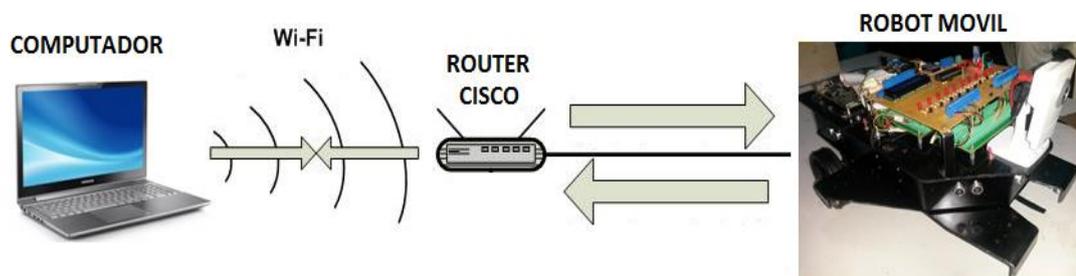


Figura 4.4 Comunicación inalámbrica. Transmisión de datos vía inalámbrica donde se comunica el PC vía WIFI con la tarjeta FPGA NI sbRIO-9631 instalada en el robot móvil.

Para empezar la configuración del router es necesario conectar un cable de Ethernet en uno de los 4 puertos LAN Ethernet que posee a la tarjeta sbRIO, acoplado el router al robot, empezamos con los pasos para su configuración.

Para ingresar a la ventana de configuración del router, es necesario hacerlo a través de un navegador web (Internet Explorer, Firefox, Google Chrome), en la barra de direcciones del navegador se ingresa la dirección IP del router, si el router no ha sido configurado anteriormente por defecto la dirección IP es 192.168.0.1, al dar enter el router nos solicita un nombre de usuario y una contraseña. Así mismo por defecto el nombre de usuario es “admin” y la contraseña es “1234”.

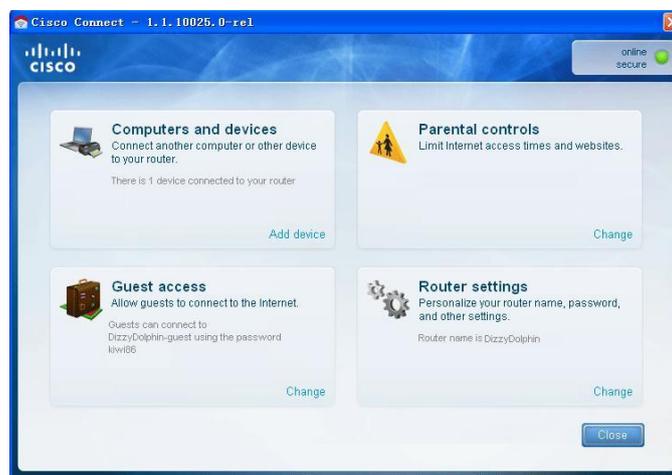


Figura 4.5 Ventana de acceso para configuración del router inalámbrico. Ventana principal al ingreso del router Cisco donde nos permite configurar la conexión con el computador.

Una vez ingresado, se tiene la pantalla principal para la configuración del router, se selecciona Router Settings para cambiar el nombre de la red y la dirección IP, la dirección que mantendremos para la red será 192.168.10.20 con el nombre de usuario “PezPequeno”. (Ver Figura 4.6)

4.3 EJECUCIÓN DE UN PROGRAMA SOBRE EL ROBOT MÓVIL

Para poner a prueba la configuración realizada anteriormente es necesario diseñar un pequeño programa donde se pondrá a prueba la tarjeta interfaz evalRIO que viene con nuestra tarjeta FPGA NI sbRIO-9631, y que a través de switches en esta tarjeta se controla el encendido y apagado de varios leds.

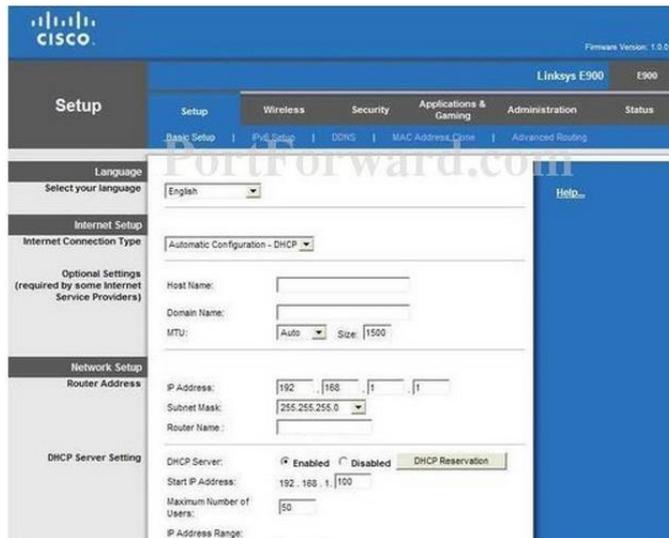


Figura 4.6 Ventana de acceso para configuración del router inalámbrico. Ventana de configuración del Setup del router Cisco que nos permite configurarlo a una red.

Empezar con el diseño del programa abriendo un nuevo proyecto a partir de la ventana Getting Started, se abre un nuevo proyecto Empty Project y se ejecuta LabVIEW a partir de la tarjeta FPGA.

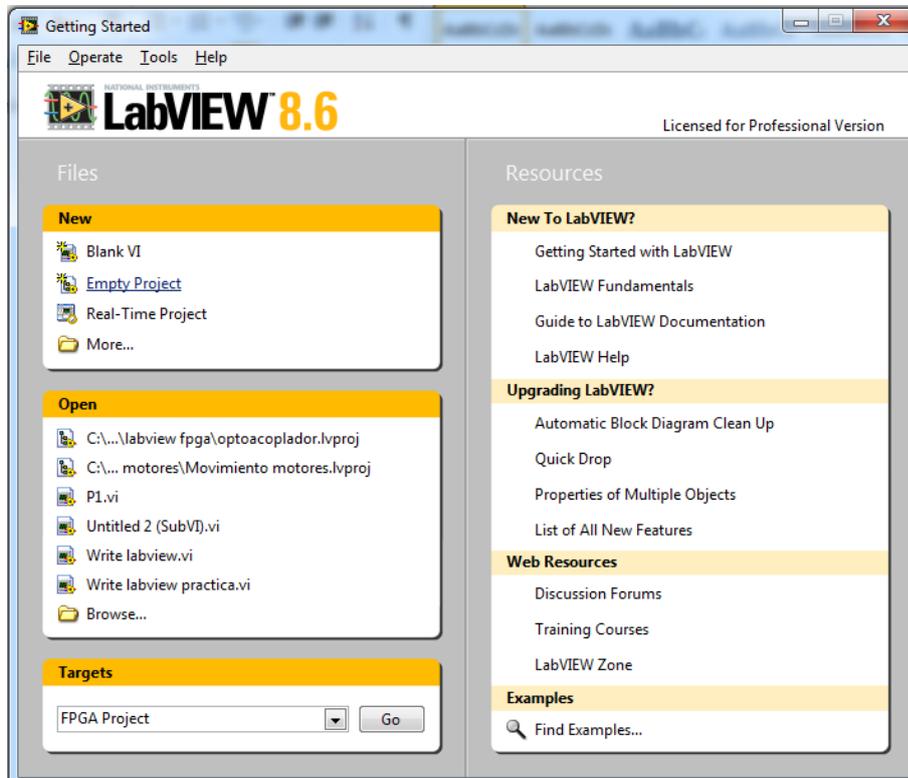


Figura 4.7 Ventana Getting Started. Esta ventana inicial de LabVIEW permite abrir nuevos VIs o nuevos proyectos a partir de la tarjeta FPGA o Real Time.

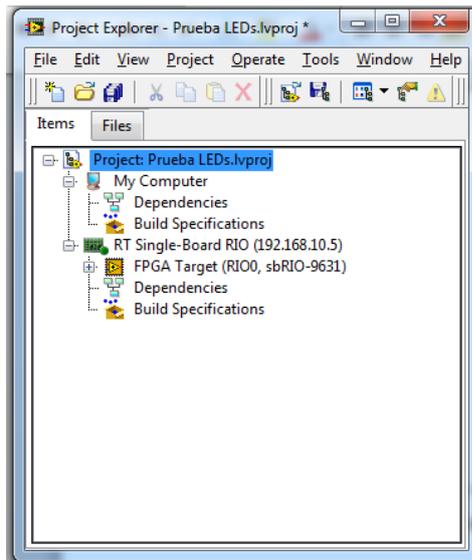


Figura 4.8 Ventana Project Explorer. Ventana Explorador de Proyectos usada para trabajar directamente con las tarjeta FPGA NI sbRIO-9631, la cual permite ejecutar LabVIEW FPGA y LabVIEW Real Time.

Una vez diseñado el proyecto de control de encendido de luces LED, y tomando las entradas I/O que son indicadas por medio de la tarjeta interfaz evalRIO, se ejecuta el programa y se puede observar como se comunica el computador con el robot móvil.



Figura 4.9 VI Panel Frontal Control de luces LED. VI que permite controlar el encendido y apagado de los Leds en la parte superior.

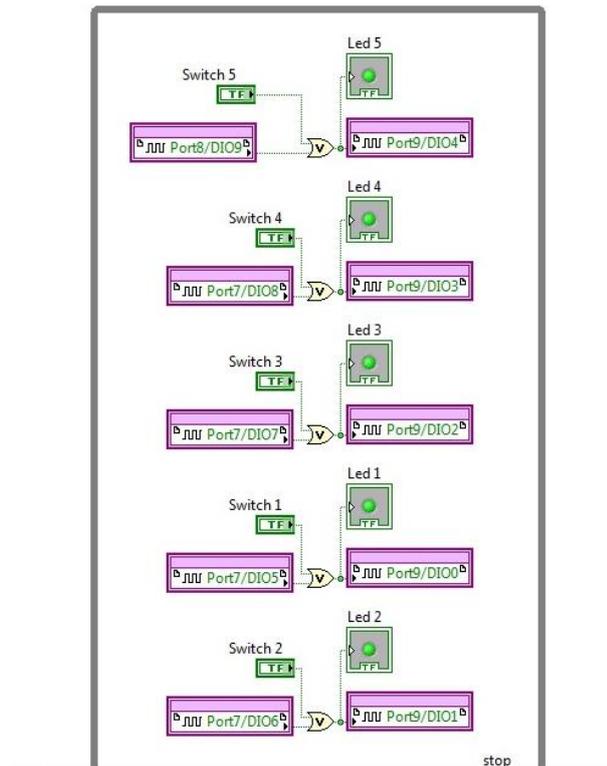


Figura 4.10 VI Diagrama de Bloques Control de Luces LED. Programación realizada para el control del encendido de los leds ubicados en la tarjeta evalRIO de la FPGA NI sbRIO-9631.

Como se puede observar en las figuras nos muestra el control de encendido de luces LED, a través de Panel frontal (Figura 4.9) y en el Diagrama de Bloques (Figura 4.10) se tiene la programación ejecutada, en la que se puede observar que el control se realiza de dos formas, a través de los switch ubicados en el Panel Frontal y a través de los Switchs instalados directamente en la tarjeta evalRIO. El encendido de los LEDs serán mostrados en la pantalla del computador como en la tarjeta. (Ver Figura 4.11)

5. CONCLUSIONES

Como se puede observar en esta práctica se conoció la manera más rápida para configurar nuestra tarjeta FPGA NI sbRIO-9631 junto al robot móvil y logramos configurar nuestro robot móvil de forma inalámbrica a través de un router, aprendiendo como se configura una red inalámbrica en la cual figura el computador, el router y el robot móvil y logrando ponerlo a prueba con una sencilla práctica.

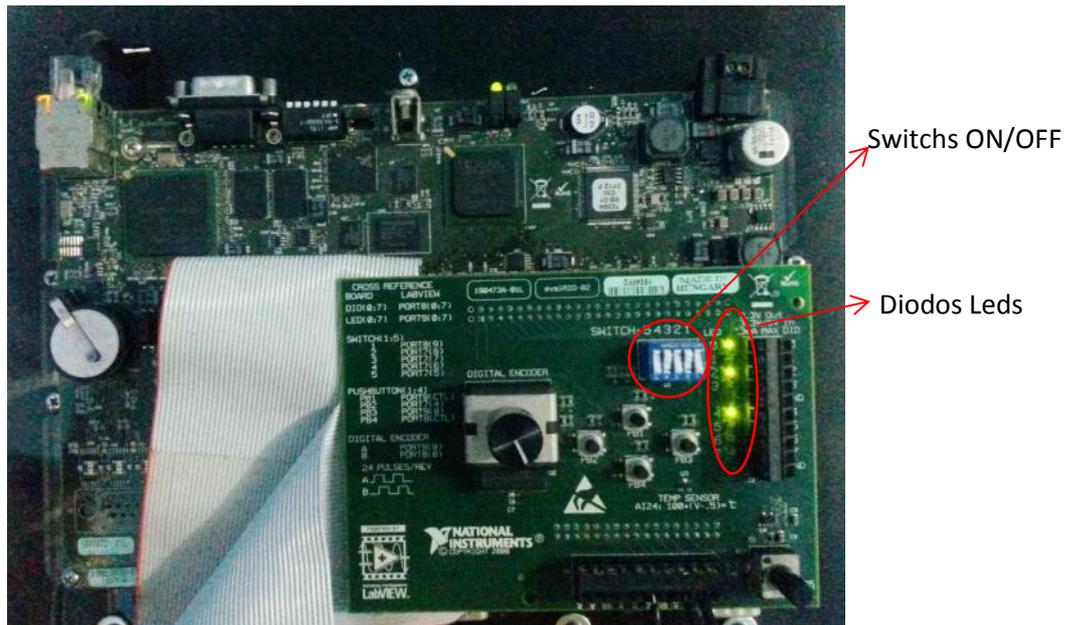


Figura 4.11 Vista superior de la tarjeta NI sbRIO-9631. Imagen tomada durante el funcionamiento de la practica Prueba de LEDs donde podemos constatar el encendido de las luces de los led en la tarjeta evalRIO.

6. RECOMENDACIONES

Es necesario tener conocimientos previos sobre el uso de la tarjeta FPGA NI sbRIO-9631, las seguridades, los cuidados y el manejo de la tarjeta, conociendo los valores máximos permisibles que soporta la tarjeta y sus puertos I/O con los que se maneja. Para la configuración de los equipos con los que se trabaja en esta práctica es necesario también conocer sobre el manejo redes, manejo y configuración de IP, protocolos de internet TCP/IPv4, entre otros.

4.3 GUÍA DE LABORATORIO 2

1. TEMA

ADQUISICIÓN DE VARIABLES DEL ENTORNO

2. OBJETIVOS

2.1 GENERAL

- Hacer el reconocimiento de todas las variables a controlar por medio de los sensores que existen en el cuerpo del robot móvil dentro de LabVIEW y mostrar en el programa las formas de ondas de las diversas señales.

2.2 ESPECÍFICOS

- Realizar un VI o los VIs en LabVIEW que permita el reconocimiento de los sensores del robot móvil.
- Aprender a manejar un display LCD de tipo 16x2 que muestre la lectura de los sensores que se encuentran en el entorno del robot móvil.
- Comprobar el buen funcionamiento de los motores que mueven las ruedas del robot, a través de los movimientos en modo manual.

3. MARCO TEÓRICO

En esta parte conoceremos las características que tienen cada uno de los sensores (temperatura, humedad, luminosidad, distancia) con los que trabaja el robot móvil en función del entorno en el cual se desarrollará.

3.1 Sensor de Humedad HIH4030

Este sensor de humedad fabricado por Honeywell, mide la humedad relativa (% HR), entrega una tensión de salida analógica y gracias a la tensión de salida lineal los datos son muy fáciles de procesar. (Ver Figura 4.12)

Características:

- Señal de salida lineal analógica
- Tensión de alimentación 4-5.8VDC
- Bajo consumo de corriente menor a los 200uA

- Precisión mejorada
- Su rendimiento es estable



Figura 4.12 Sensor de humedad HIH4030. Vista superior del sensor de humedad montado en una tarjeta para salida de pines directo. GND, OUT, 5V. Ingeniería, C. d. (2015). Sensor de humedad HIH-4030. Obtenido de <http://www.cosasdeingenieria.com/esp/index/item/177/73/sensor-de-humedad-hih-4030>.

3.2 Sensor de temperatura LM35

El LM35 es un sensor de temperatura con un precisión calibrada de 1°C. Su rango de medición abarca desde -55°C. hasta 150°C. La salida es lineal y cada grado celcius equivale a 10mV. (Ver Figura 4.13)

Características:

- Calibrado directamente en grados Celcius
- La tensión de salida es proporcional a la temperatura
- Tiene un precisión garantizada de 0.5°C a 25°C.
- Opera entre 4 a 30V de tensión
- Baja impedancia de salida
- Baja corriente de alimentación
- Bajo costo

3.3 Sensor de luz por Fotorresistencia

Este módulo sensor de luz usa un foto-detector para dectectar la intensidad de la luz del ambiente. En función de la cantidad de luz que se expone la resistencia del sensor varía, originando cambios de voltaje a la salida. Posee salidas de voltaje analógicas y digitales, tiene un potenciómetro que sirve para ajustar la sensibilidad de la salida digital, este es “0” cuando la intensidad de luz excede el valos fijado por

el potenciómetro. El voltaje de salida analógico aumenta con la intensidad de la luz. (Ver Figura 4.14)

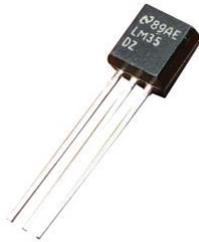


Figura 4.13 Sensor de temperatura LM35. Vista frontal del sensor de temperatura de tres pines en un encapsulado TO-92 para soldar directamente en placa. Dashboard. (20 de 01 de 2015). LM35 Temperature Sensor. Obtenido de <https://www.ntnu.no/wiki/display/plab/7.+LM35+Temperatursensor>



Figura 4.14 Sensor de luz con fotorresistencia. Vista superior de la fotorresistencia montado en una tarjeta que permite obtener salida analógica y salida digital. ELECTRONICA.COM.VE. (2015). Modulo Sensor de Luz por Fotorresistencia CDS. Obtenido de http://electronica.com.ve/new/catalog/product_info.php?products_id=35

3.4 Sensor de Ultrasonido HC-SR04

El sensor de ultrasonido HC-SR04 es un sensor de distancias, capaz de detectar objetos y calcular la distancia a la que se encuentra dentro de un rango de trabajo de 2 a 400 cm. Este sensor funciona por ultrasonido y contiene toda la electrónica encargada de hacer la medición. Es de muy pequeño tamaño y se destaca por su bajo consumo, gran precisión y bajo precio.

Características:

- Alimentación de +5VDC

- Interfaz sencilla , de 4 hilos: Vcc, Trigger, Echo, GND
- Rango de medición de 2 a 400 cm
- Corriente de alimentación: 15 mA
- Frecuencia de pulso: 40 khz
- Apertura de pulso ultrasónico: 15°
- Señal de disparo: 10uS
- Dimensiones del módulo: 45x20x15mm



Figura 4.15 Sensor de Ultrasonido HC-SR04. Vista frontal del sensor con 4 pines que son VCC, Trigger, Echo, GND, para conexión directa a la placa electrónica. DIGIBAY. (2013). HC-SR04 Ultrasonic Proximity sensor. Obtenido de <http://www.digibay.in/hc-sr04-ultrasonic-proximity-sensor>

3.5 Display LCD. Características

Los display LCD tienen la capacidad de mostrar cualquier carácter de tipo alfanumérico, permitiendo que cualquier equipo que transmita datos pueda ser representados a través del LCD de manera sencilla.

Está montado en una base de circuito impreso en donde se hayan los controladores y los pines para la conexión del display. En total podemos observar 2 líneas de 16 caracteres cada una, mostrando un total de 32 caracteres. La tensión de alimentación es de +5VCC.

El display LCD posee 16 pines donde se pueden conectar fácilmente a cualquier sistema controlador. (Ver Figura 4.16)

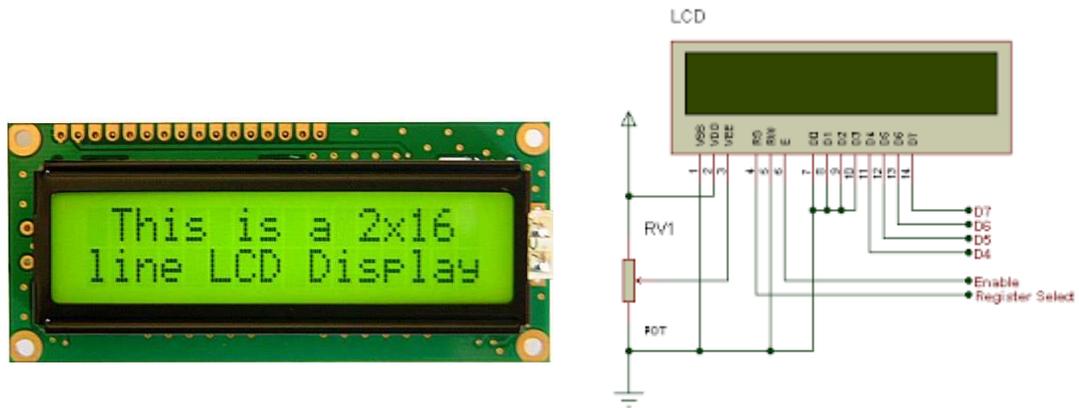


Figura 4.16 Display LCD de 16x2 y su conexión. Vista frontal del display LCD mostrando datos, forma correcta de conexión del LCD para recibir datos. UNOCERO. (19 de 10 de 2013). Nuevos trucos para un viejo display LCD. Obtenido de <http://www.unocero.com/2013/10/19/nuevos-trucos-para-un-viejo-display-lcd/>

Tabla2.

Distribución de pines del display LCD 16x2

PIN No	Nombre	Función	Descripción
1	Vss	Power	GND
2	Vdd	Power	5V
3	Vee	Contrast Adj.	0 -5V
4	RS	Comand	Register Select
5	R/W	Comand	Read / Write
6	E	Comand	Enable (Strobe)
7	D0	I/O	Data LSB
8	D1	I/O	Data
9	D2	I/O	Data
10	D3	I/O	Data
11	D4	I/O	Data
12	D5	I/O	Data
13	D6	I/O	Data
14	D7	I/O	Data MSB

Nota. En la tabla se detalla cada uno de los pines del display LCD de 16x2, indicando el nombre de cada pin con la función y la descripción de cada uno de ellos.

2. DESARROLLO

Para el desarrollo de esta práctica se ha hecho mención de cada uno de los sensores que se utilizaran como son de temperatura de humedad, de luminosidad y

de distancia, cada sensor maneja curvas en la lectura totalmente diferentes por lo que aplicaremos fórmulas distintas en la aplicación de cada una de ellas.

Se inicia creando el proyecto el cual se llama “Prueba_LCD”. Como se observa en la Figura 4.17, se muestra la apariencia final del proyecto con todos los VIs y subVIs realizado en esta práctica.

Para empezar con la programación de este proyecto, es necesario realizar los VIs necesarios para la adquisición de las señales de los sensores que se encuentran en nuestro robot móvil.

Como se observa en la Figura 4.18, en el diagrama en bloques se tiene la programación para la adquisición de la señal de los sensores de temperatura, humedad y luminosidad, usando las entradas analógicas AI27 (temperatura), AI28 (Luminosidad), AI29 (Humedad).

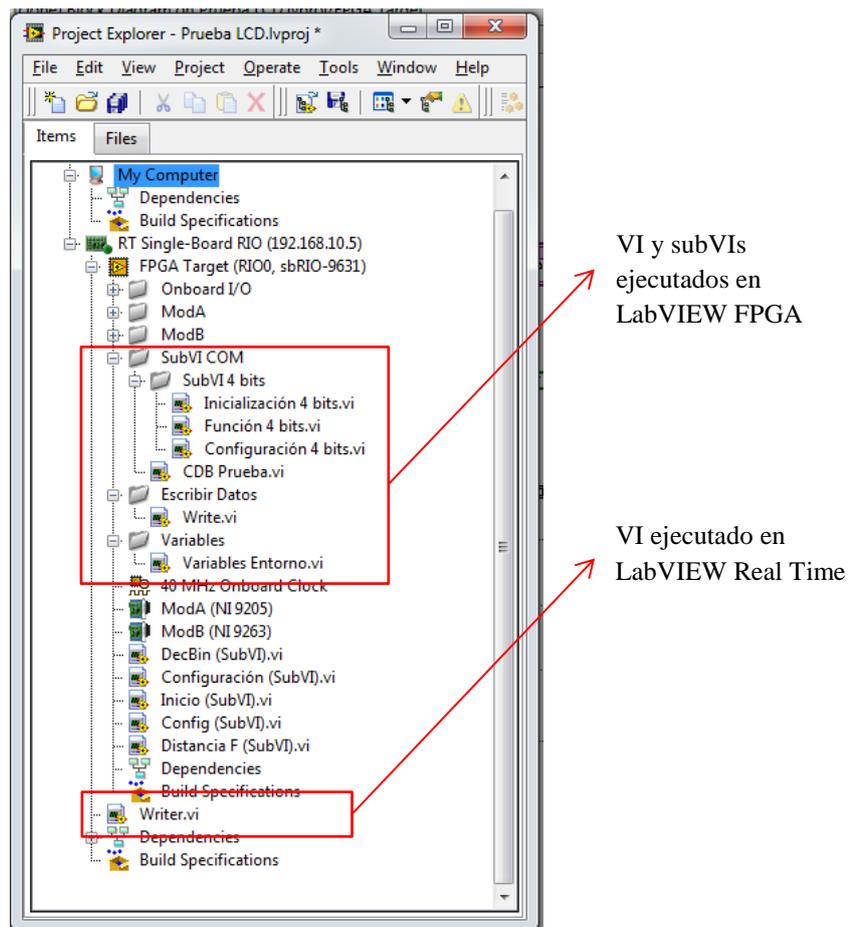


Figura 4.17 Proyecto Prueba LCD. Detalle del proyecto en donde se encuentran ubicados los VIs y subVIs destinados a este proyecto.

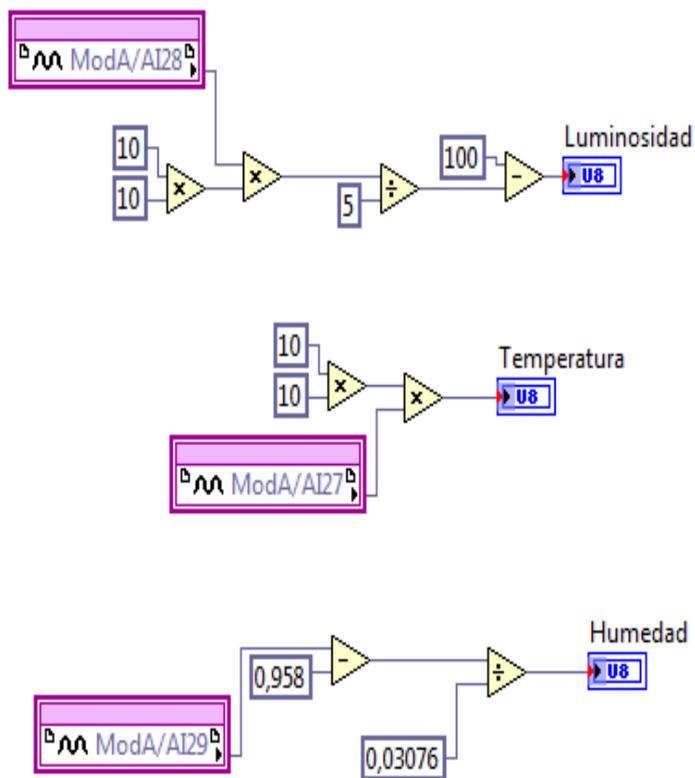


Figura 4.18 Diagrama de Bloques de la adquisición de datos. Adquisición de datos de los sensores de temperatura, humedad y fotorresistencia.

Ahora es necesario usar el sensor ultrasónico para la adquisición de datos de distancia, en la programación es necesario generar un subVI debido al mucho espacio que ocupa toda su programación.

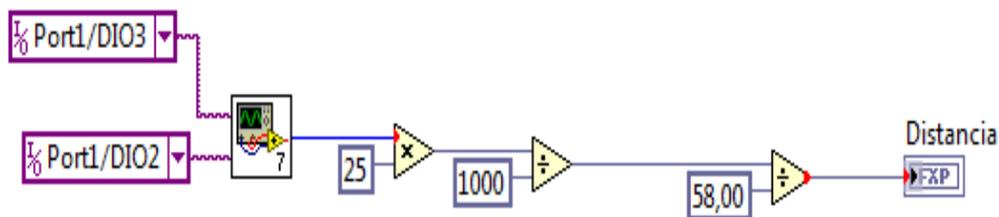


Figura 4.19 Diagrama de Bloque del VI para la adquisición de datos del sensor ultrasónico para medir la distancia, tomando dentro su programación el subVI para la inicialización del sensor de ultrasonido.

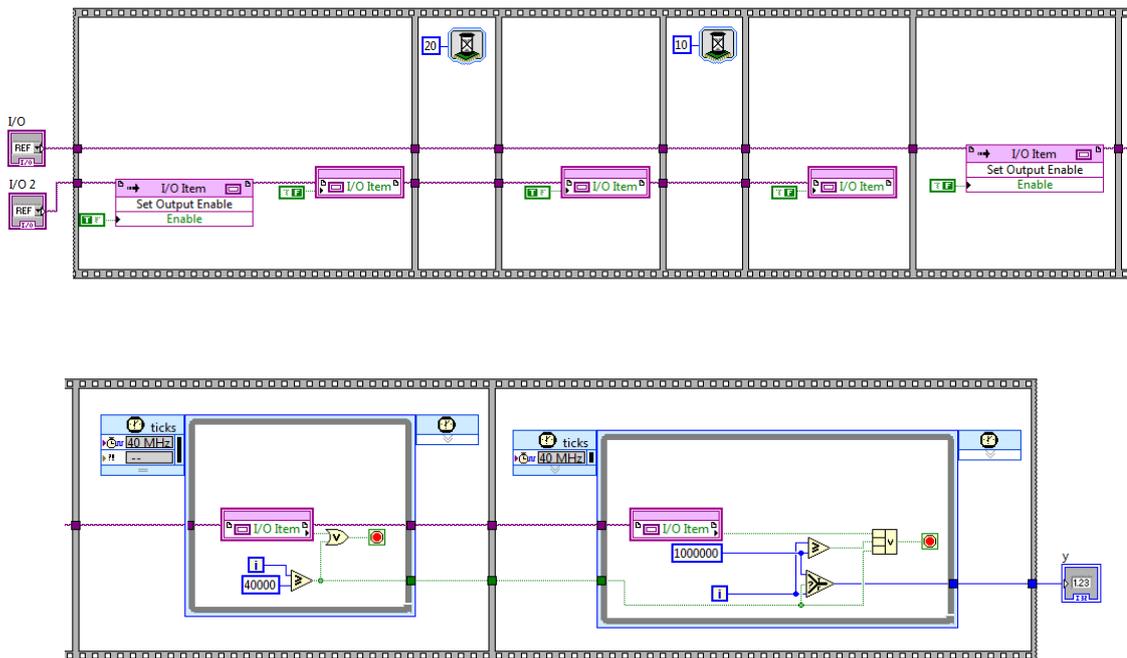


Figura 4.20 Diagrama de Bloques para la inicialización del sensor ultrasónico. En la imagen se detalla paso a paso la programación realizada al sensor para el arranque antes de que empiece a tomar datos.

En la Figura 4.20, se tiene el Diagrama de Bloques de la programación del sensor de ultrasonido, en el cual se esta manejando con un subVI que permite iniciar al sensor, luego de ello la programación se basa en una recta lineal en el cual al dato obtenido se le realizó un simple cálculo para obtener el dato requerido antes de ser mostrado en pantalla.

Como ya se ha obtenido el dato de cada sensor, estos serán transmitido hacia el display LCD, en el cual se detalla paso a paso la inicialización, configuración y transmisión de los datos hacia obtenidos de cada sensor hacia el display LCD.

Para el control del display LCD son necesarios 3 bits: una señal para habilitación (E), otra para lectura/escritura (R/W) y otra para seleccionar uno de los dos registros internos (RS), por lo tanto al momento de controlar el LCD se necesitan 8 bits para datos mas 3 bits para control, en total serían 11 bits, que equivale a 11 pines digitales que la tarjeta NI sbRIO-9631 debe tener disponible.

Antes de conectar el LCD para la transmisión de datos hacia la tarjeta NI sbRIO-9631, es posible configurar su bus de datos de dos formas diferentes, como un bus de 8 bits, usando los pines (D0 a D7) o también como un bus multiplexado de 4 bits (D0 a D3). Ésta última es una opción muy útil para ahorrar pines digitales en la tarjeta NI sbRIO-9631, en vez de usar 11 señales en total, se usarán tan solo 7 señales y para ello es necesario multiplexar y demultiplexar los datos recibidos.

En la Figura 4.21 se muestran las diferentes opciones al momento de conectar el LCD con un bus de 8 bits y con un bus de 4 bits, cabe recalcar que cuando se trabaja con tan solo 4 bits los primeros datos en ser enviados son los más significativos, luego los 4 bits menos significativos.

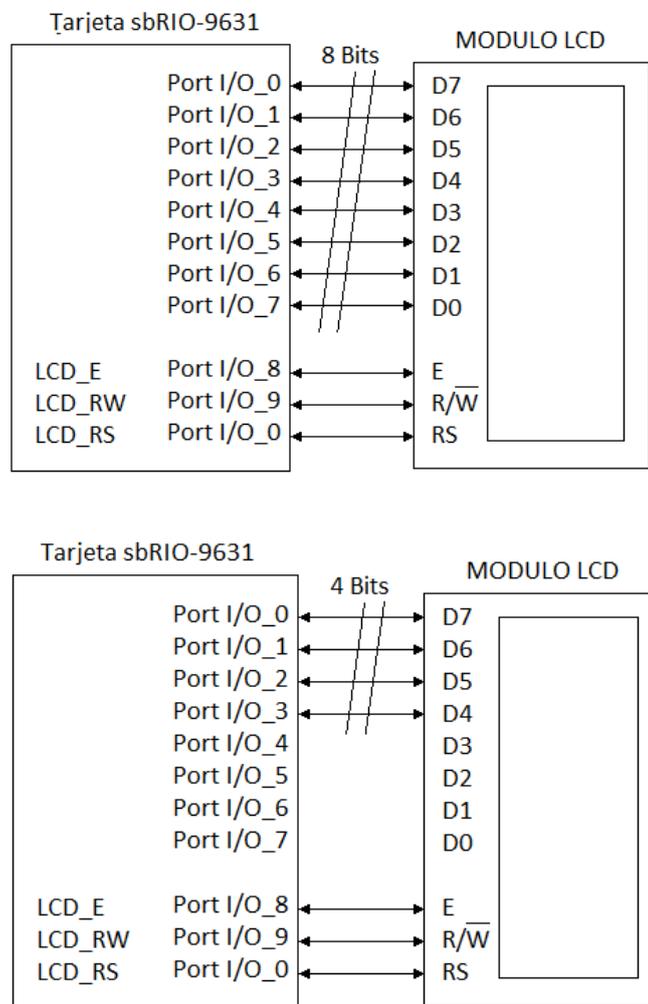


Figura 4.21 Conexión del LCD con la tarjeta sbRIO-9631. Conexión que puede ser usada con una interfase de 4 y 8 bits, el cual detalla las conexiones del bus de datos tomando los datos más significativos. (D7 a D4 o D7 a D0)

4.1 Inicialización del LCD con interfaz de 4 bits

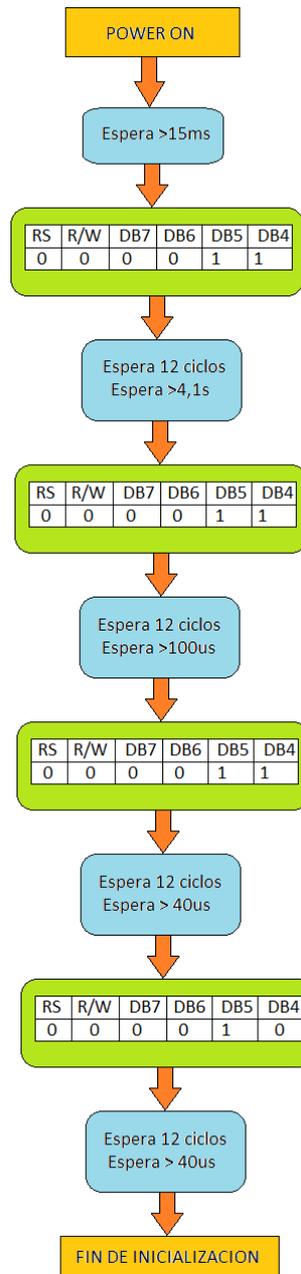


Figura 4.22 Diagrama de flujo para la inicialización del LCD. Una vez energizado el LCD espera determinados ciclos mayores al tiempo indicado tomando pulsos en sus entradas de datos con el objetivo realizar una secuencia en los pines de entrada. Secuencia realizada para una interfaz de 4 bits.

Una vez energizado el LCD, es necesario que sea inicializado para establecer un protocolo de comunicación y pueda trabajar correctamente. En este caso vamos a

inicializar al LCD de forma automática a través de la programación. En nuestro caso una vez inicializado el LCD la transmisión de datos se realizaría con 4 bits para ahorrar pines en la tarjeta NI sbRIO-9631.

Para tener más clara la idea realizaremos un diagrama de flujo que explique mejor la inicialización del LCD como muestra la Figura 4.22

Ahora se realiza la programación del VI que permita inicializar el display LCD, mostrado en la figura siguiente.

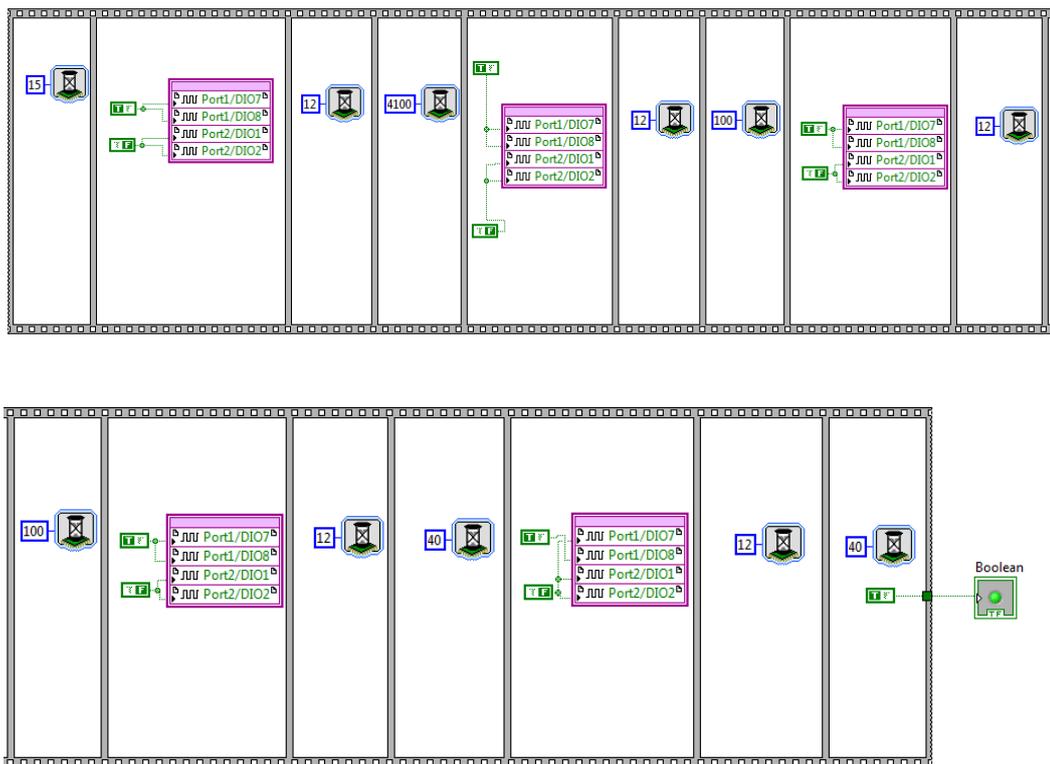


Figura 4.23 Diagrama de Bloque del VI para la inicialización del display LCD, manejando una interfaz de 4 bits.

4.2 Configuración del LCD

Una vez terminada la inicialización del LCD y teniendo establecido la interfaz de transmisión de datos, en este nuestro caso 4 bits, se realiza la configuración del LCD mostrada en un diagrama de flujo para entender mejor el caso. En el diagrama explica mejor como se realiza la configuración paso a paso para manejar el display LCD conectado en la tarjeta de adquisición de datos del robot móvil.

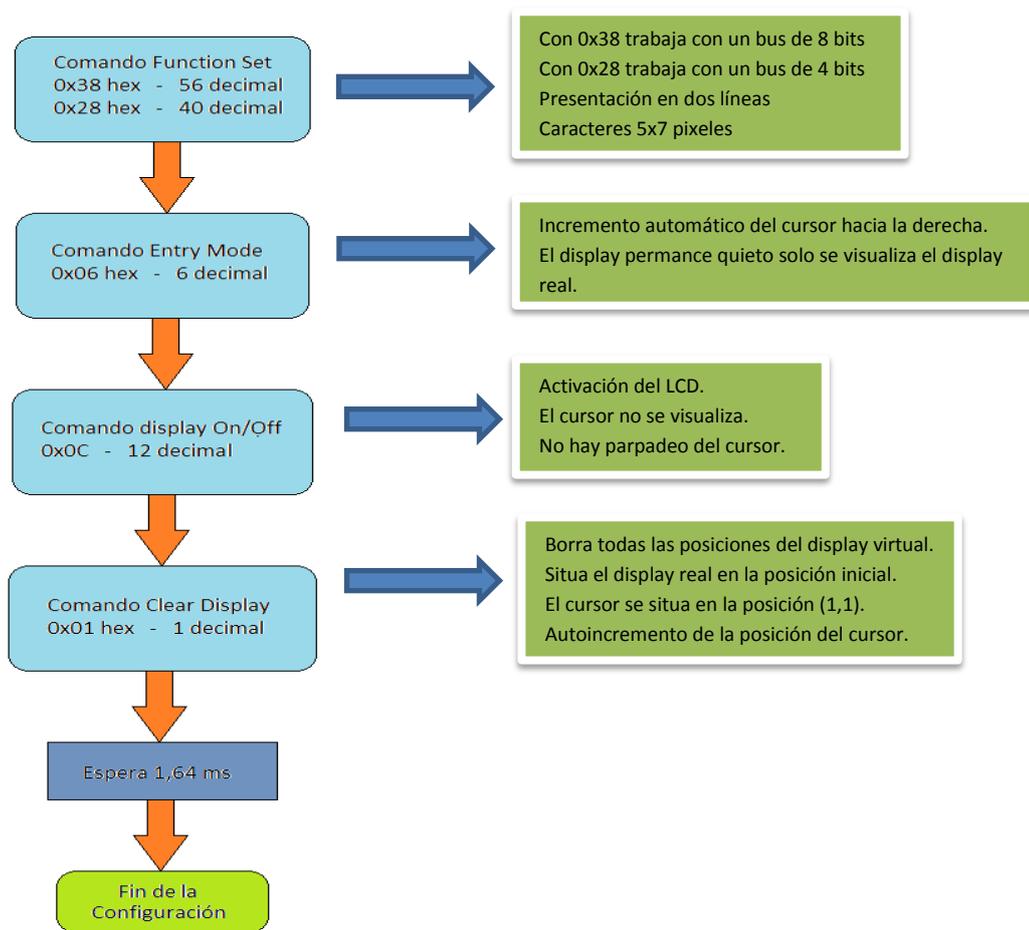


Figura 4.24 Diagrama de Flujo para la configuración del LCD. En este diagrama nos explica como configurar paso a paso el LCD, en una serie de cuatro comando útiles al momento de la configuración y el tiempo de espera antes de que finalice.

En la Figura 4.25 se muestra la programación realizada en el diagrama de bloques para la ejecución del VI para la configuración del LCD.

Como se puede observar en la programación del diagrama de bloques, se detalla paso a paso la secuencia que mantiene igual al diagrama de flujo al momento de configurar el LCD. Internamente esta formado por subVIs, los cuales explicaremos a continuación.

En la Figura 4.26, se puede observar en el diagrama de bloques que el valor ingresado es convertido a un arreglo de 7 números binarios (1 byte) a través de la función Number to Boolean Array, usando luego un arreglo Array To Cluster, para después usar la función Unbundle By Name que extrae los elementos del Cluster para generar cada bit.

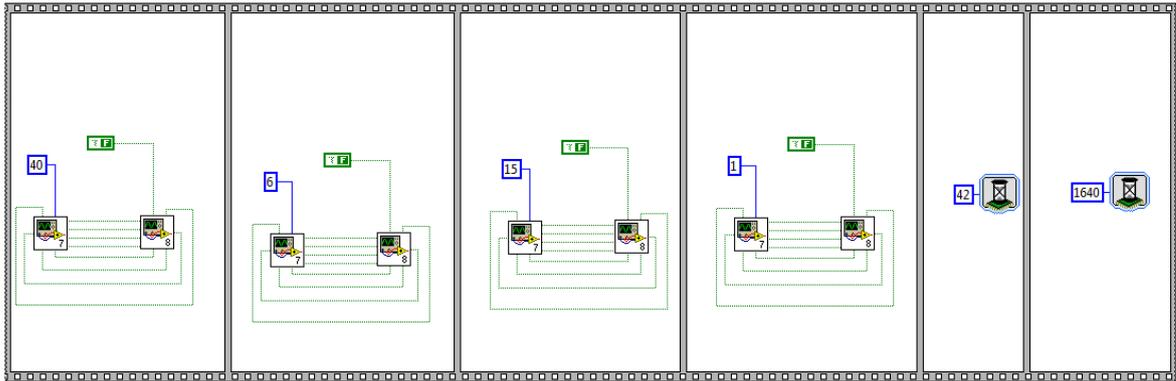


Figura 4.25 Diagrama de Bloques para la configuración del LCD. Imagen que detalla paso a paso la programación para la configuración de este elemento después de haber realizado la inicialización.

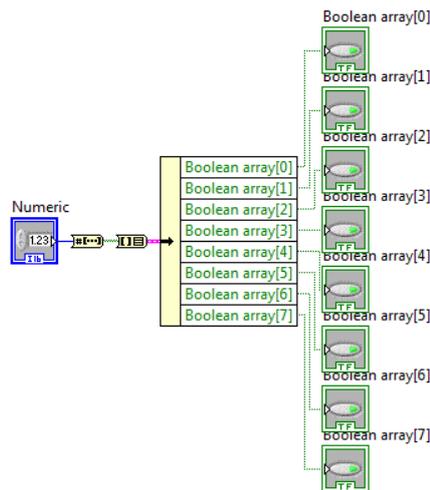


Figura 4.26 Diagrama de bloques subVI Conversión de valores decimales a binarios. El dato ingresado es convertido de decimal a binario a través un arreglo para la generación de cada bits, desde bit 0 hasta bit 7.

Una vez terminado el subVI para la conversión de datos decimales a binarios, estos datos son ingresados a otro subVI formado por varios frames, en la entrada del primer frame tenemos un control booleano llamado Comando que va directamente conectado al LCD al pin de control LCD_RS. Éste permite controlar si se va a escribir un dato (carácter ASCII) o un comando de control, también ingresan 4 controles booleanos D7, D6, D5, D4 y van conectados a la entrada de los pines del

LCD, en este caso son los más significativos (MSB) de un dato o un comando control.

En el octavo frame también se tiene el ingreso de 4 controles booleanos D3, D2, D1, D0, conectados a los pines del LCD y representan los pines menos significativos (LSB).

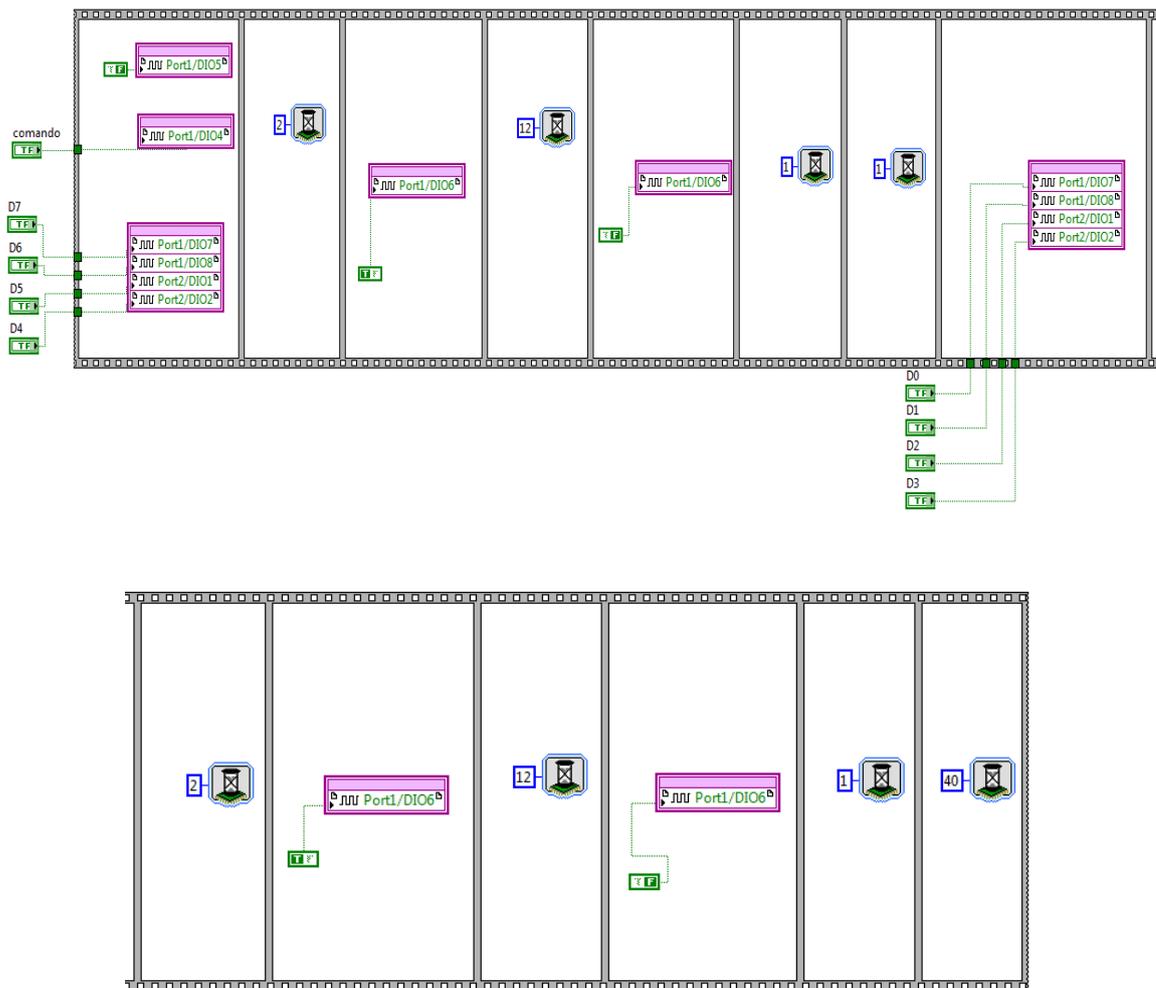
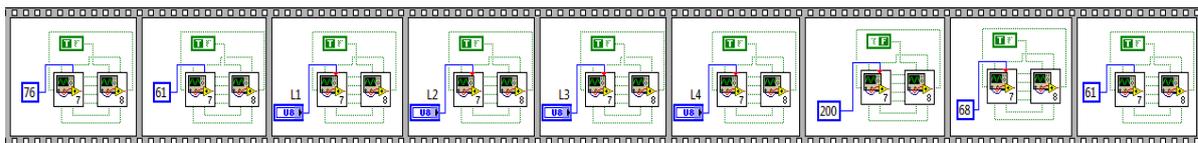
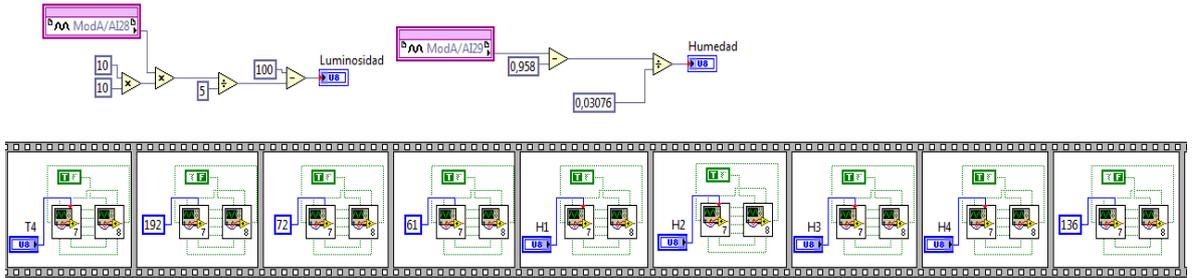
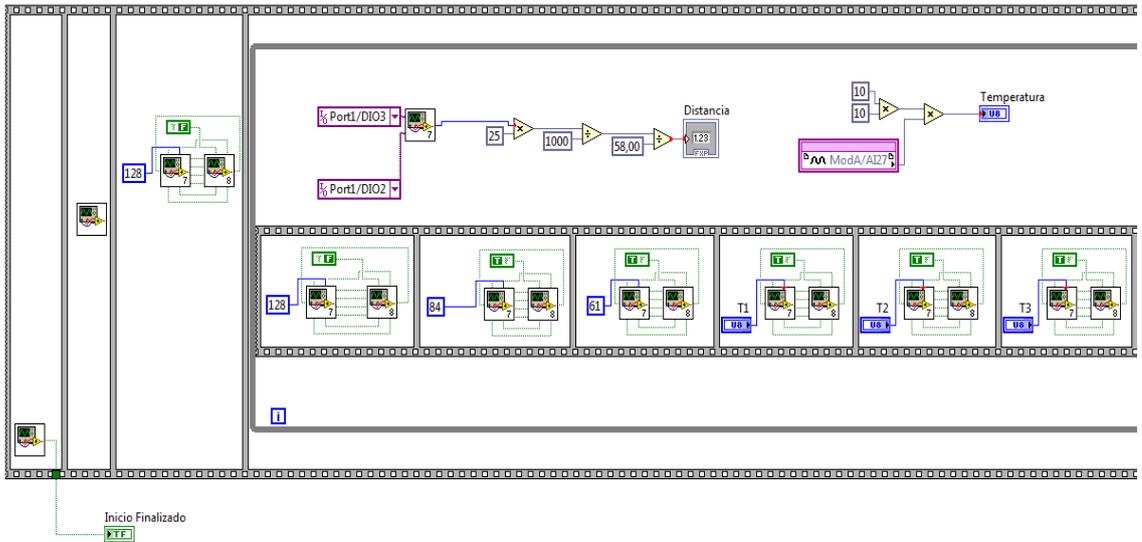


Figura 4.27 Diagrama de bloques subVI Adquisición de datos binarios y habilitación del LCD. Los controles booleanos ubicados en el primer frame van directamente conectados a los pines del LCD usando los pines MSB para el envío de datos.

Una vez inicializado y configurado el LCD vamos a proceder a realizar un VI en LabVIEW FPGA que permita leer cada dato enviado por cada sensor, este VI será referenciado para trabajar en LabVIEW Real Time. La programación en el diagrama de bloques se detalla en la siguiente figura.



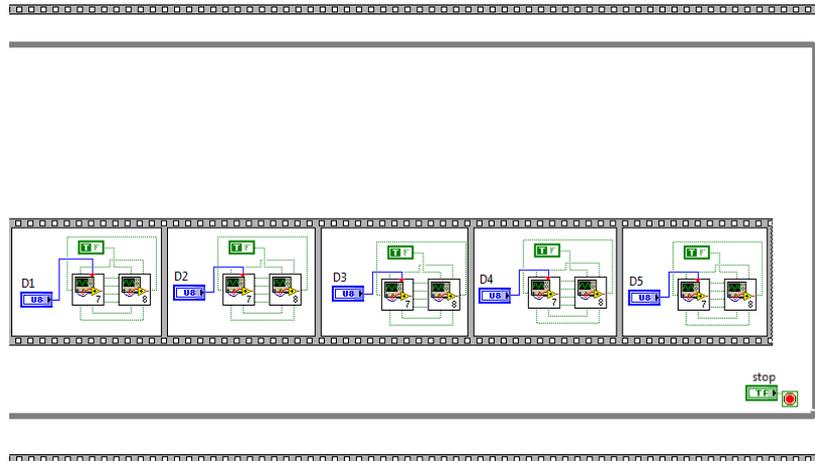


Figura 4.28 Diagramas de Bloques del VI_Writer. Datos adquiridos de los sensores y enviados bit por bit, usando los subVIs usados en las figuras 4.25 y 4.27.

4.3 Programación del VI Real Time para Escribir caracteres ASCII sobre el LCD

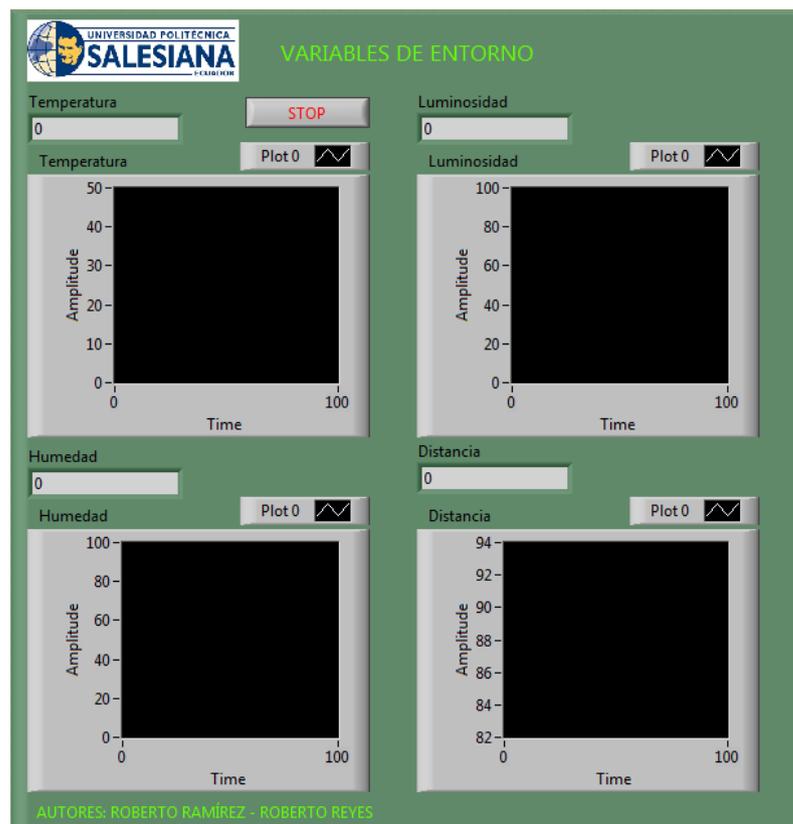


Figura 4.29 Panel Frontal del VI Real Time. Imagen que detalla el valor de cada sensor del robot móvil mostrado gráficamente a través de la función Waveform Chart.

El archivo del VI Write.vi se ejecuta sobre el procesador Real Time de la tarjeta NI sbRIO-9631 y que puede hacer uso de las E/S del chip FPGA a través del VI Writer.vi usando interfaz de comunicación de 4 bits.

La apariencia del panel frontal de este VI se encuentra mostrada en la Figura 4.30, en ella se puede observar 4 indicadores cada uno indicando el dato de cada sensor usado en el robot móvil. Estos datos son mostrados en tiempo real a través de la pantalla del panel frontal del LabVIEW como en el display LCD sobre el cual se trabaja.

En la Figura 4.31, se muestra la programación realizada en LabVIEW Real Time, como se puede observar los datos numéricos tomados en decimales son mostrados directamente en el panel frontal a través de indicadores e histogramas, estos mismos datos pasan por unos arreglos para ser convertidos a código ASCII y retornen estos mismos datos a la FPGA y puedan ser mostrados a la salida en el LCD.



Figura 4.30 Visualización de datos en el LCD. Captura de imagen tomada del robot móvil enviando datos de medición desde los sensores.

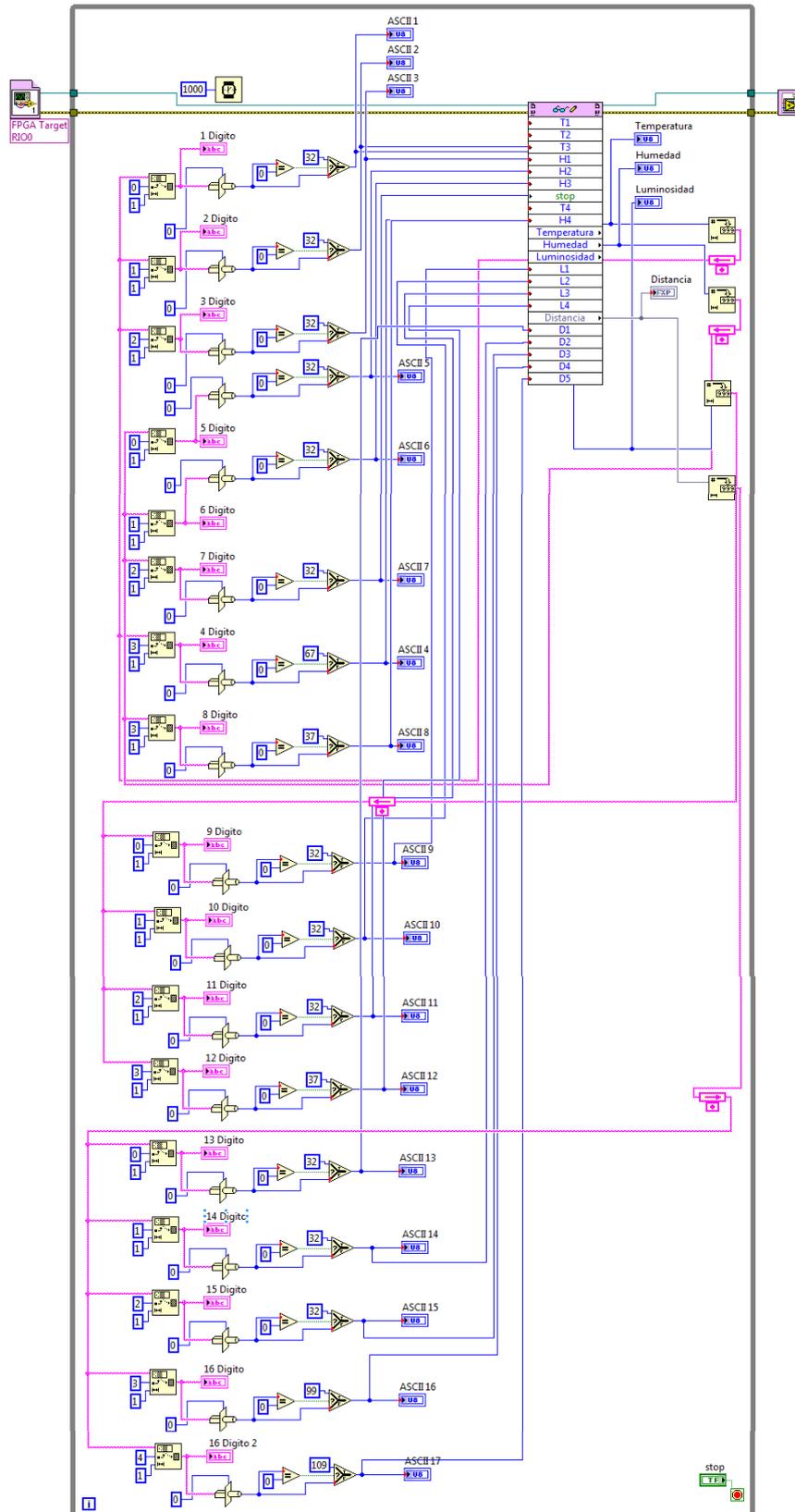


Figura 4.31 Diagrama de bloques del VI Real Time. Programación realizada para convertir los datos numéricos en ASCII y puedan ser visualizados en el LCD.

3. CONCLUSIONES

- En el desarrollo de esta práctica se logra conocer las diversas variables dentro del entorno del robot en el cual se encuentra trabajando, además de obtener un análisis de estas señales gracias al historial mostrado en el panel frontal de LabVIEW.
- La programación fue realizada directamente manejando LabVIEW FPGA y LabVIEW Real Time, para un mejor desempeño del robot y adquisición de datos mostrados en tiempo real a través del panel frontal de LabVIEW.
- También se logra hacer uso de una pantalla LCD externa de 16x2, para que los datos obtenidos por los sensores sean vistos a través de esta pantalla, aprendiendo la programación del envío de datos desde LabVIEW hacia el LCD.

4. RECOMENDACIONES

- Es necesario conocer el datasheet de cada sensor al momento de usarlo, de conocer el tipo de señales que maneja (analógica o digital) y los pines de alimentación y de salida de datos.
- También se recomienda tener conocimientos previos en el uso de la pantalla LCD, su conexión para brillo e intensidad y la entrada de datos para que puedan ser visualizados.

4.4 GUIA DE LABORATORIO 3

1. TEMA

COMUNICACIÓN CON DISPOSITIVOS INALÁMBRICOS

2. OBJETIVOS

2.1 GENERAL

- Realizar la comunicación entre la tarjeta FPGA del robot y el computador por medio de señales inalámbricas ya sean estas por vía bluetooth o wi-fi y a la vez hacer el uso de un dispositivo de comando que interactúe tanto con el robot como con el computador inalámbricamente.

2.2 ESPECÍFICOS

- Conocer paso a paso el proceso de comunicación inalámbrica del robot móvil con el computador vía WIFI a través de un router, aprendiendo a configurar este dispositivo.
- Diseñar un programa en LabVIEW que permita el control de movimientos del robot móvil usando el dispositivo inalámbrico XBEE.

3. MARCO TEÓRICO

3.1 Dispositivo Xbee

Son dispositivos de comunicación que brindan un medio inalámbrico para la interconexión y comunicación entre dispositivos. Estos módulos utilizan el protocolo de red llamado IEEE 802.15.4 para crear redes punto a multipunto y punto a punto. Diseñados para aplicaciones que requieren de un alto tráfico de datos, baja latencia y una sincronización de comunicación predecible. Propiedad de Digi basado en protocolo Zigbee. (MCI Electronics, 2015)

Características

- Alimentación +3.3Vcc
- Velocidad de transferencia 250kbps Max.

- Potencia de salida 1mW o 60mW (+18dBm)
- Alcance 90 metros
- pines ADC de 10 bits
- 8 pines digitales I/O
- Encriptación 128 bits
- Configuración local o de forma inalámbrica



Figura 4.32 Módulo Xbee Series 1. Imagen superior del módulo Xbee que permite comunicación inalámbrica con un alcance de hasta 90 metros. Gravitech. (2014). XBee 802.15.4 (Series 1) 1mW Point-to-Multipoint RF Module with Chip Antenna. Obtenido de <http://www.gravitech.us/xb80s163porf.html>

3.2 Joystick Shield

El joystick shield es un mando de control de juego, posee cinco botones pulsadores y una palanca de mando P2P para el posicionamiento de dos ejes. El movimiento vertical de la palanca de mando producirá un voltaje analógico proporcional en el pin analógico 0, igualmente, el movimiento horizontal de la palanca de mando puede ser detectado en el pin analógico 1.

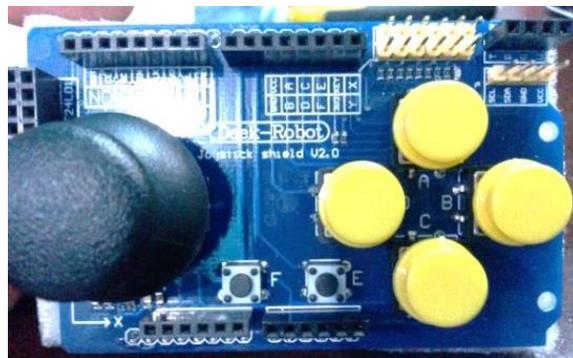


Figura 4.33 Mando de control Joystick Shield Arduino. El mando de control simula una palanca de videojuego con palanca y botoneras.

3.3 Integrado PIC16F877A

Dispositivo programable capaz de realizar diferentes actividades que requieren del procesamiento de datos digitales y del control y comunicación digital de diferentes dispositivos.

Fabricado por Microchip familia a la cual se le denomina PIC. Microcontrolador muy versátil, eficiente y práctico.

Características

- Procesador RISC 20Mhz
- Frecuencia de operación del reloj 0-20Mhz
- Voltaje de alimentación de +2Vcc a 5.5Vcc
- Ahorro de energía en el Modo de Suspensión
- 35 pines de entrada/salida con tecnología Flash
- 256 registros de 8 bits EEPROM
- 368 registros de 8 bits de RAM

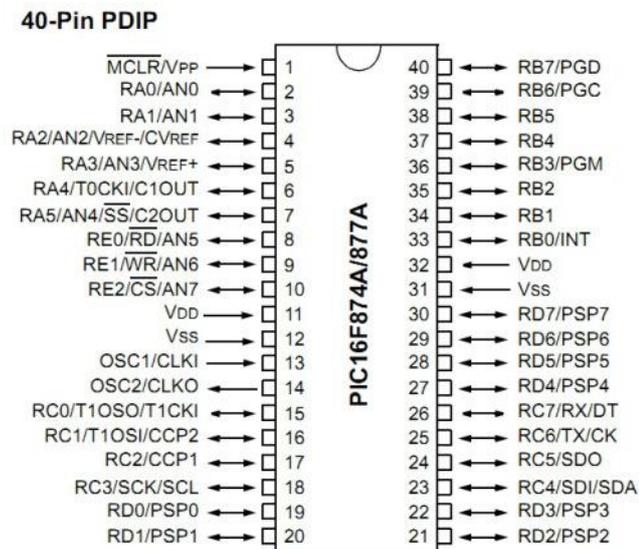


Figura 4.34 Distribución de pines PIC16F877A. En la figura podemos observar el detalle de cada pin del microcontrolador que abarcan un total de 40 pines distribuidos en 35 pines de entrada. Microchip. (17 de Junio de 2003). PIC16F87XA DATASHEET. Obtenido de <http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf>

4. DESARROLLO

Para el desarrollo de esta práctica es importante conocer el manejo del módulo inalámbrico Xbee, el cual será el medio de comunicación para el completo desarrollo de nuestra práctica.

4.1 Implementación del módulo Xbee

Al momento de implementar el Xbee es necesario conocer los pines de distribución con los cuales vamos a trabajar. En la figura 4.35 se puede observar el detalle de cada uno de los pines del módulo.

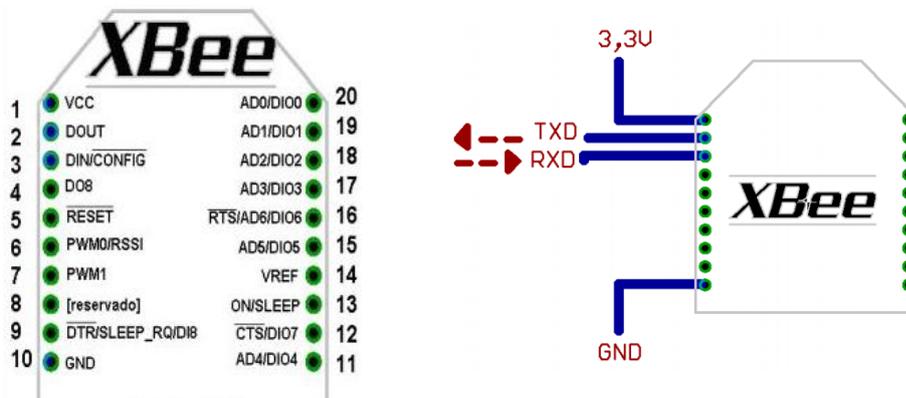


Figura 4.35 Distribución de pines del módulo Xbee y conexión básica para transmisión y recepción de datos. En la imagen se detallan cada uno de los 20 pines que conforman el Xbee, usados para diferentes aplicaciones en la transmisión de datos. MCI Electronics. (01 de Julio de 2010). Guía de Usuario Xbee Series 1. Obtenido de www.olimex.cl/pdf/Wireless/ZigBee/XBee-Guia_Usuario.pdf

Cómo se indico en el marco teórico el módulo Xbee trabaja con una alimentación de +3.3Vcc, voltaje preciso que necesita para que pueda trabajar, si el voltaje no es el adecuado puede llegar a no cumplir con la función requerido e incluso puede llegar a quemarse.

Para evitar este tipo de inconvenientes se usó una la tarjeta Xbee Explorer, que se adapta al módulo Xbee y permite trabajar con +5Vcc regulando este voltaje a los +3.3Vcc que el Xbee necesita para su funcionamiento. Podemos observar el Xbee Explorer en la siguiente figura.

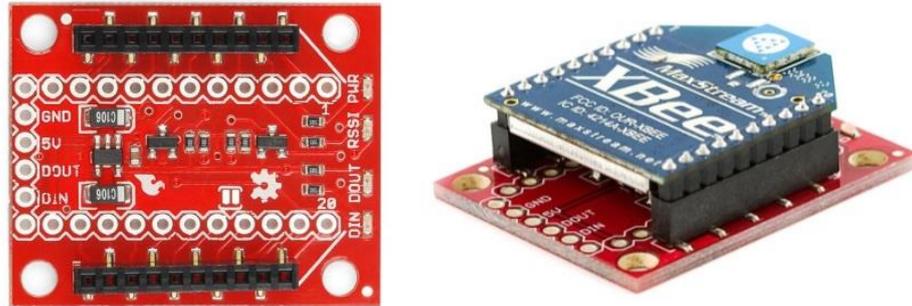


Figura 4.36 Xbee Explorer. Base reguladora que se adapta al módulo Xbee para manejar voltajes de +5Vcc, en la envío y recepción de datos. Roboshop. (2015). Xbee Explorer Regulated. Obtenido de <http://www.robotshop.com/en/sfe-xbee-explorer-regulated.html>

Para la transmisión y recepción de datos se usa el microcontrolador PIC16F877A, que se adapta perfectamente al módulo Xbee, usando las librerías de comunicación permite acoplar con un sencillo programa para leer los datos que serán enviados a través del control inalámbrico.

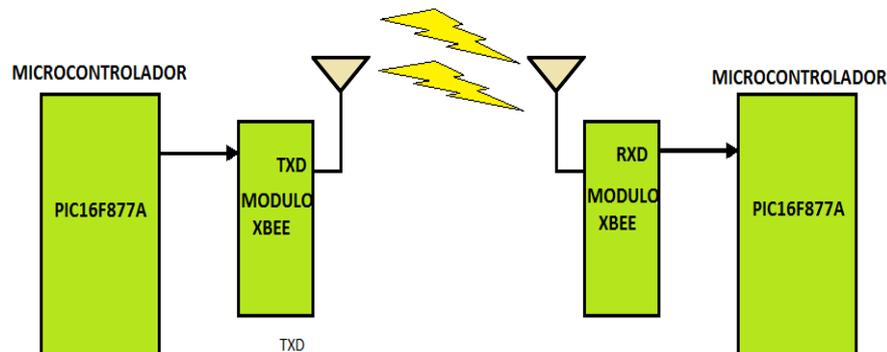


Figura 4.37 Envío y recepción de datos con Xbee y microcontrolador. Envío de señales a través del módulo de control hacia el receptor.

Usando un joystick ubicado en el mando de control se maneja los movimientos básicos del robot móvil, este joystick se conectará al microcontrolador quien a su vez detectará los movimientos realizados para que puedan ser enviados a través del modulo Xbee hacia el robot móvil, quien detectará los datos a través de otro módulo Xbee, datos que serán leídos a través de un microcontrolador y ejecutados a la tarjeta NI sbRIO-9631.

A continuación se detalla la programación del módulo transmisor realizada en el microcontrolador PIC16F877A, programación realizada en PICBASIC para el manejo de movimientos básicos del robot móvil.

4.2 Programa Transmisor de Datos del mando de Control

```

*****
'* Name   : Transmision de datos xbee.BAS                               *
'* Author : Roberto Ramirez / Roberto Reyes                             *
'* Notice : Copyright (c) 2015 [select VIEW...EDITOR OPTIONS]          *
'*       : All Rights Reserved                                           *
'* Date   : 22/02/2015                                                    *
'* Version : 1.0                                                          *
'* Notes  :                                                                *
'*       :                                                                *
*****

@devicext_osc; Oscilador externo XT
define osc 4;
INCLUDE "modedefs.bas"

; Programa para el trasmisor
; Entradas del Pic

Xbee_Tx      var   Portc.0;
Xbee_Rx      VAR   Portc.1;
dato         var   byte;
Trisd = %00111111

MAIN:
    gosub cero;
    gosub uno;
    gosub dos;
    gosub tres;
    gosub cuatro;
    gosub cinco;
    gosub seis;
    goto main

cero:  If (portd==0) then
        SEROUT2 Xbee_Tx,84,["G"];
        pause 100;
        goto CERO
    else
        return;
endif

uno:   If (portd.0==1) then
        SEROUT2 Xbee_Tx,84,["A"];

```

```

    pause 100;
    goto uno
    else
    return;
endif

dos:  If (portd.1==1) then
    SEROUT2 Xbee_Tx,84,["B"];
    pause 100;
    goto dos;
    else
    return;
endif

tres:  If (portd.2==1) then
    SEROUT2 Xbee_Tx,84,["C"];
    pause 100;
    goto tres;
    else
    return;
endif

cuatro:  If (portd.3==1) then
    SEROUT2 Xbee_Tx,84,["D"];
    pause 100;
    goto cuatro;
    else
    return;
endif

cinco:  If (portd.4==1) then
    SEROUT2 Xbee_Tx,84,["E"];
    pause 100;
    goto cinco;
    else
    return;
endif

seis:  If (portd.5==1) then
    SEROUT2 Xbee_Tx,84,["F"];
    pause 100;
    goto seis;
    else
    return;
endif
goto main
end

```

Este programa toma las variables de entrada enviadas por el joystick al microcontrolador 16F877A, este captura las variaciones de voltaje del joystick y son enviadas a través del Xbee hacia el receptor.

4.3 Programa de Recepción de Datos al Robot Móvil

```

*****
'* Name   : UNTITLED.BAS                                     *
'* Author : [select VIEW...EDITOR OPTIONS]                 *
'* Notice : Copyright (c) 2015 [select VIEW...EDITOR OPTIONS] *
'*       : All Rights Reserved                             *
'* Date   : 22/02/2015                                     *
'* Version : 1.0                                           *
'* Notes  :                                               *
'*       :                                               *
*****
DEFINE OSC 4
INCLUDE "modedefs.bas"

; Programa para el transmisor
; Entradas del Pic

Xbee_Tx      var   Portc.0;
Xbee_Rx      VAR   Portc.1;

Dato         var   Byte;

Trisb = %00000000;
dato = 0;
portb = %11111111;
pause 1000;
portb = %00000000;
dato = 0;

MAIN:
    gosub recibir_dato;
    gosub cero;
    gosub uno;
    gosub dos;
    gosub tres;
    gosub cuatro;
    gosub cinco;
    gosub seis;

```

```

goto main;

recibir_dato: SERIN2 Xbee_Rx,84,[Dato]
    return;

cero: if (dato == "G") then
    portb = %00000000;
    return;
endif

uno: if (dato == "A") then
    portb = %00000001;
    return;
endif

dos: if (dato == "B") then
    portb = %00000010;
    return;
endif

tres: if (dato == "C") then
    portb = %00000100;
    return;
endif

cuatro: if (dato == "D") then
    portb = %00001000;
    return;
endif

cinco: if (dato == "E") then
    portb = %00010000;
endif

seis: if (dato == "F") then
    portb = %00100000;
endif

goto main;

end

```

Una vez que el receptor Xbee recibe la información, será procesado por el microcontrolador 16F877A, cuyas salidas irán directamente hacia la tarjeta FPGA NI sbRIO-9631, este ejecuta los movimientos a través del programa de control en LabVIEW FPGA y LabVIEW Real Time, para dar paso al movimiento generado en los motores del robot móvil.

A continuación se procede a crear un nuevo Proyecto en LabVIEW a través de la ventana de Getting Started el cual vamos a llamar “Control Inalámbrico”.

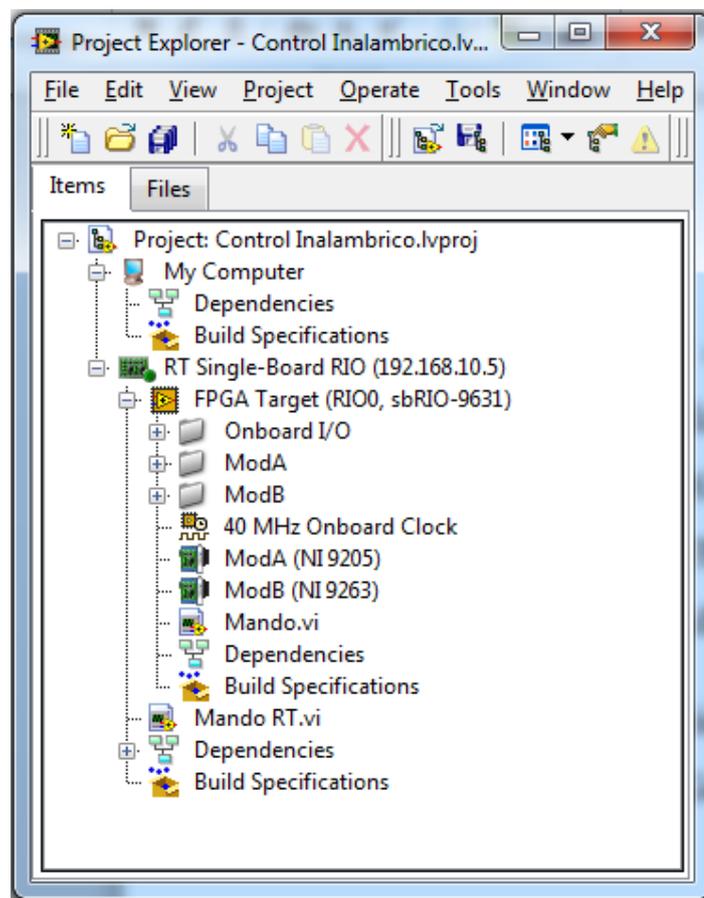


Figura 4.38 Proyecto Control Inalámbrico. Este proyecto se encarga de realizar los movimientos en el robot móvil que provienen del mando de control.

Una vez creado el proyecto se procede a realizar la programación en el diagrama de Bloques del LabVIEW FPGA, tomando las señales I/O provenientes del receptor Xbee.

tiempo real. En las Figuras 4.42 y 4.43 se puede observar la programación para la ejecución de la práctica.

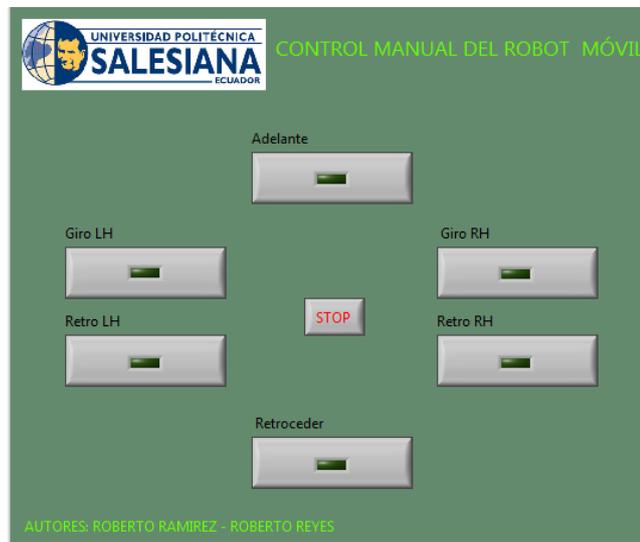


Figura 4.41 Panel Frontal del Programa Mando RT. En este programa se tienen pulsadores directos para que realicen los movimientos de forma manual en conjunto con el control inalámbrico.

Culminada la práctica podemos visualizar el mando de control con las tarjetas internas que lo conforman y a su vez los datos recibidos por el Xbee.

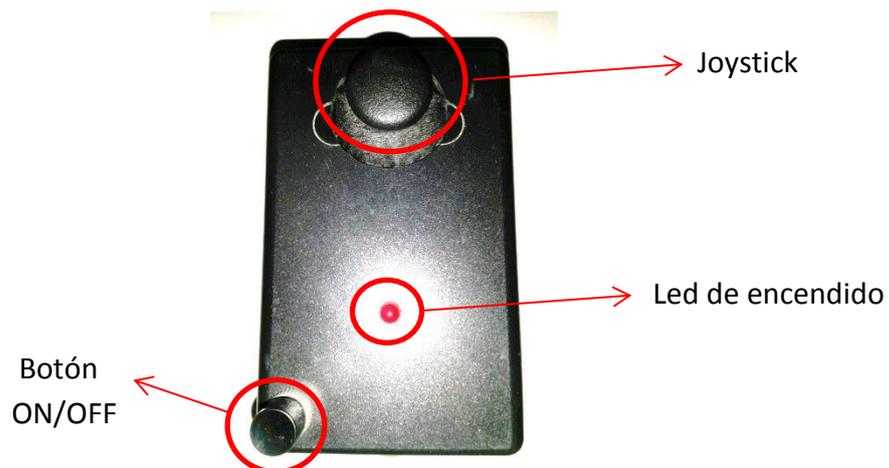


Figura 4.42 Mando de Control transmisor de datos. Este mando inalámbrico permite el control al robot móvil a través del joystick.

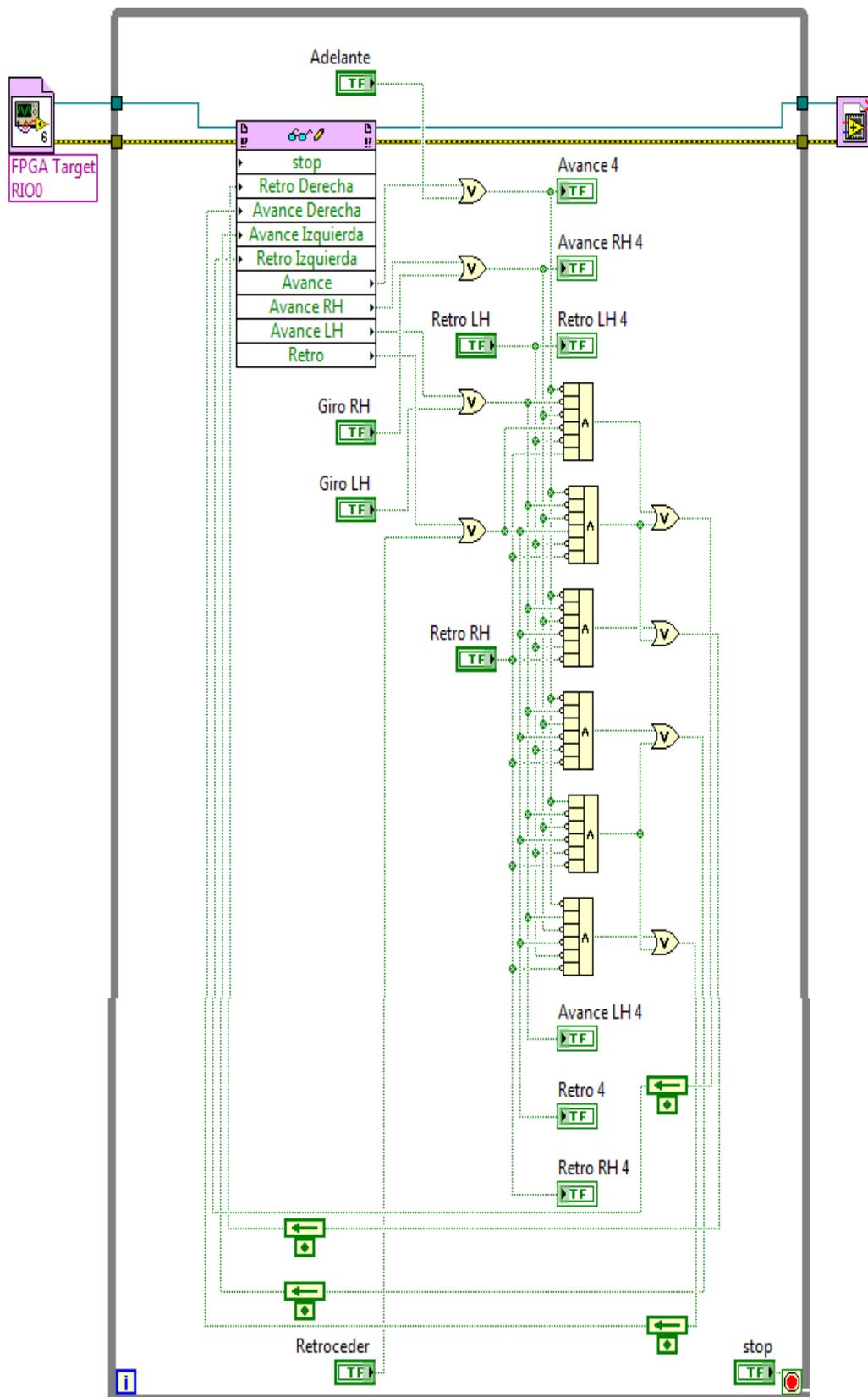


Figura 4.43 Diagrama en bloques del programa Mando RT. Este programa ejecuta los movimientos básicos del robot móvil, delante, atrás, izquierda y derecha.

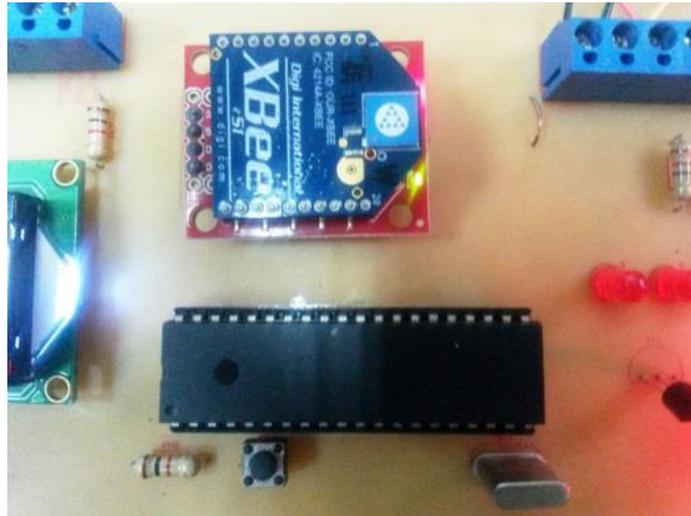


Figura 4.44 Receptor de datos Xbee en la tarjeta de Adquisición de datos. Los datos tomados por el Xbee son enviados al microcontrolador para que pueda enviar las señales a la tarjeta sbRIO-9631.

5. CONCLUSIONES

En esta práctica se aprendió a manejar el módulo Xbee, conocimos sus entradas y salidas de datos además de como enviar y recibir datos a través de microcontroladores.

6. RECOMENDACIONES

Al momento de trabajar con el Xbee es necesario conocer sus características principales como por ejemplo la alimentación con la que trabaja que es +3.3Vcc, si le enviamos un mayor voltaje podría dañar el Xbee, para ello es mejor trabajar con el Xbee Explorer, tarjeta reguladora que se adapta al Xbee y permite manejar +5Vcc, evitando cualquier diferencial de voltaje que pueda afectar a la tarjeta.

4.5 GUÍA DE LABORATORIO 4

1. TEMA

IMPLEMENTACIÓN DE UN ROBOT PATRULLERO

2. OBJETIVOS

2.1 GENERAL

- Hacer que el robot avance 40 cm en línea recta, gire 180° y dé media vuelta, avance otros 40cm, gire 180° y repita nuevamente el ciclo.

2.2 ESPECÍFICOS

- Realizar los VIs correspondientes para regular la velocidad de los motores de tal manera que el robot vaya en línea recta sin desviarse.
- Aprender a realizar circuitos PWM a través de LabVIEW y controles PID para la regulación de velocidad.

3. MARCO TEÓRICO

Encoders

Son mecanismos utilizados para entregar la posición, velocidad y aceleración del rotor de un motor. Un codificador rotatorio o encoder es un dispositivo electromecánico que convierte la posición angular de un eje, directamente a un código digital. Los encoders más comunes son los absolutos y relativos. (Requena, 2009) (Technologies, 2013)

Características Encoder E50S

- Encoder de tipo incremental
- Ideal para mediciones de ángulo, posición, revolución, velocidad, aceleración y distancia.
- Alimentación de +12-24Vcc
- Económico
- Diseño compacto adaptable a cualquier tipo de eje de motor
- De fácil instalación



Figura 4.45 Encoder incremental Autonics E50S. Encoder de cuadratura que nos permite obtener datos de posición, ángulo, velocidad, posición y aceleración. Autonics Sensor & Controllers. (2015). Encoder Rotativo. Obtenido de http://autonics.com.mx/products/products_detail.php?catecode=01/06/01&db_uid=1394.

4. DESARROLLO

Para empezar es necesario crear un nuevo proyecto directamente de la ventana de LabVIEW Getting Starter, el cual se llama “Patrullar”, esta proyecto contendrá la programación tanto en LabVIEW FPGA como en LabVIEW Real Time.

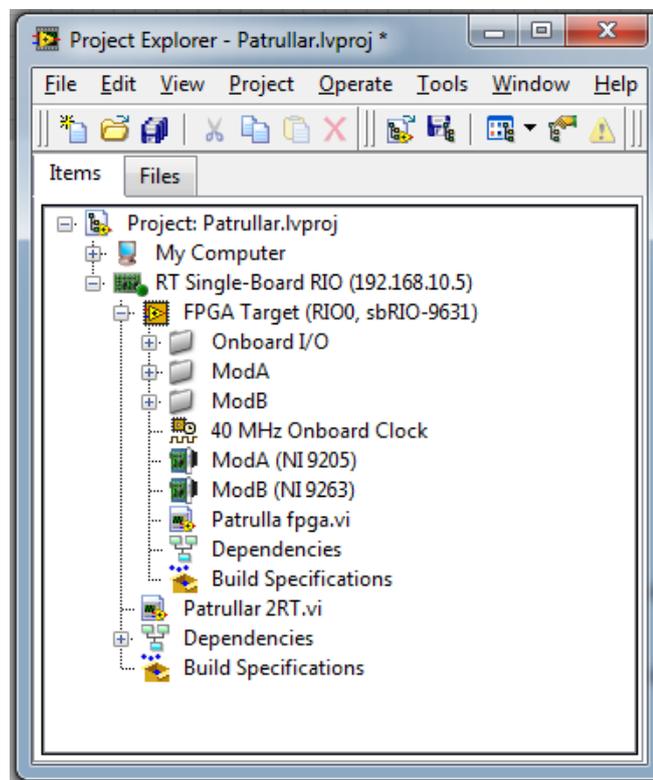


Figura 4.46 Ventana del Proyecto “Patrullar”. La figura nos indica los tanto del RT Single-Board RIO (LabVIEW Real Time) y la FPGA Target (LabVIEW FPGA).

4.1 Programación desarrollada en LabVIEW FPGA

Una vez creado el proyecto se procede a desarrollar la programación iniciando en LABVIEW FPGA para la adquisición de datos.

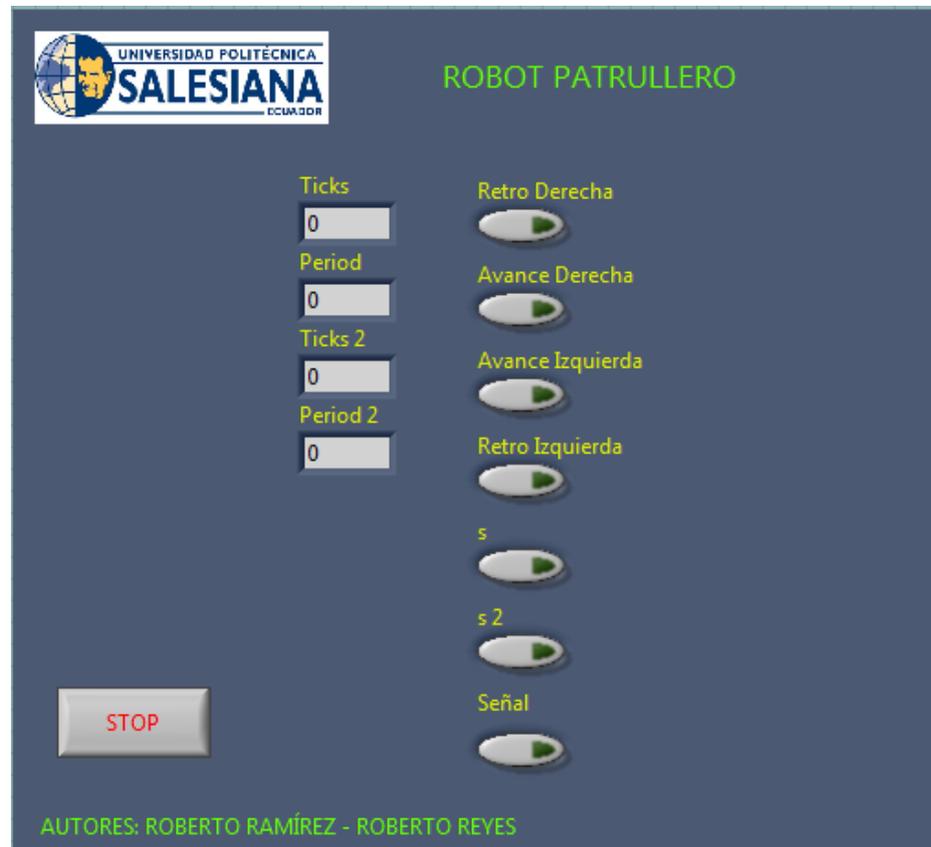


Figura 4.47 Panel Frontal del VI "Patrulla FPGA". VI desarrollado en LabVIEW FPGA para la adquisición de datos de los encoders.

En la Figura 4.47 del panel frontal se puede observar los indicadores que nos van a mostrar los pulsos generados por cada encoder del robot móvil. Adicional a ello tenemos los controles manual para cada uno de los movimientos del robot, para ser usado en el LabVIEW Real Time.

En el diagrama de control de la Figura 4.48 se tiene la programación diseñada para adquirir los datos de los encoders para el control de la velocidad, además tenemos la programación que genera una señal de control PWM para el control de los motores.

4.2 Programación desarrollada en LabVIEW Real Time

Una vez diseñado la parte de adquisición de datos a través de LabVIEW FPGA, vamos a adquirir el VI “Patrullar”, para desarrollar la programación que permita el control del robot de forma autónoma.

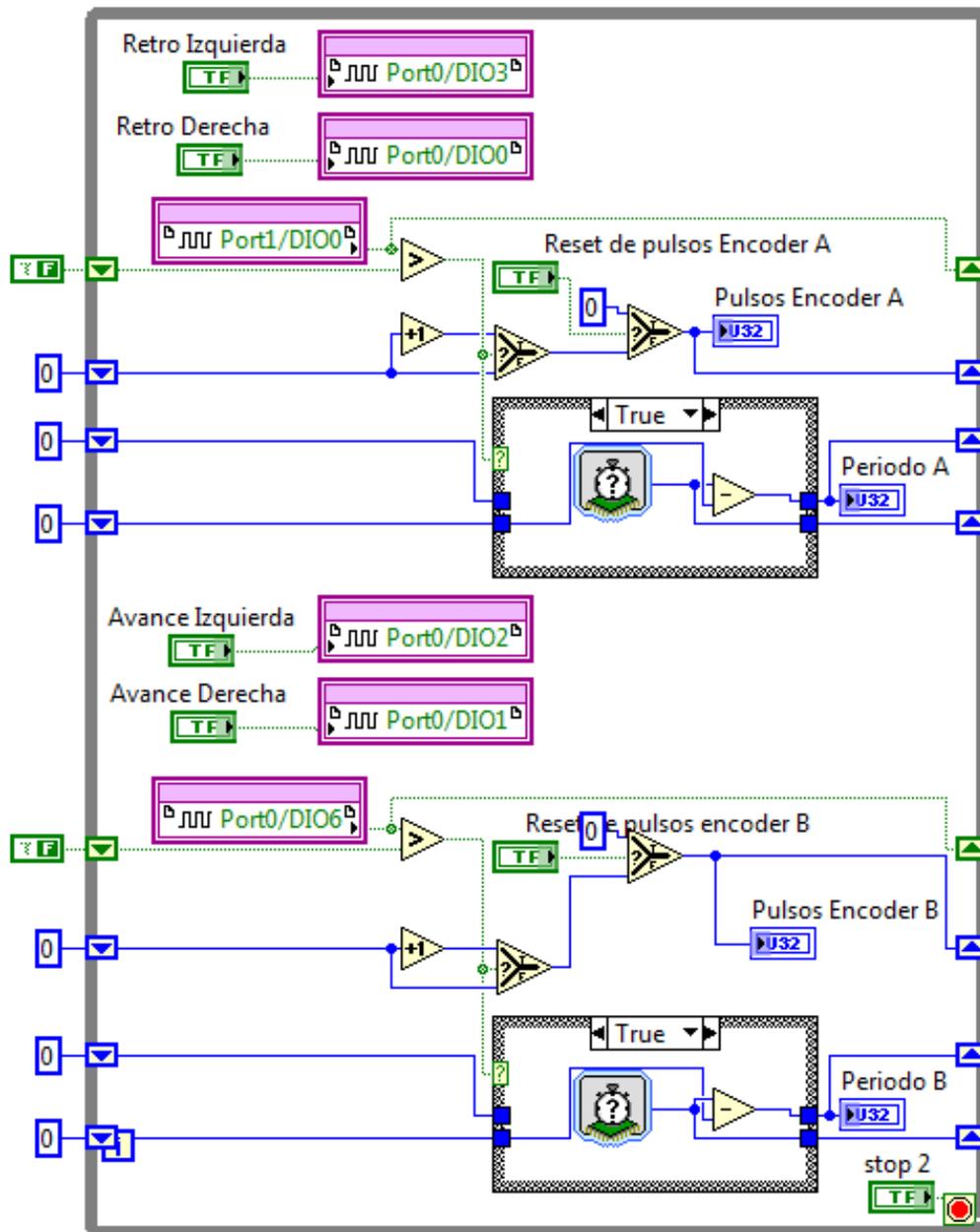


Figura 4.48 Diagrama de Control del VI “Patrullar”. Programación diseñada para la adquisición de señales de los puertos I/O para el control del robot móvil.

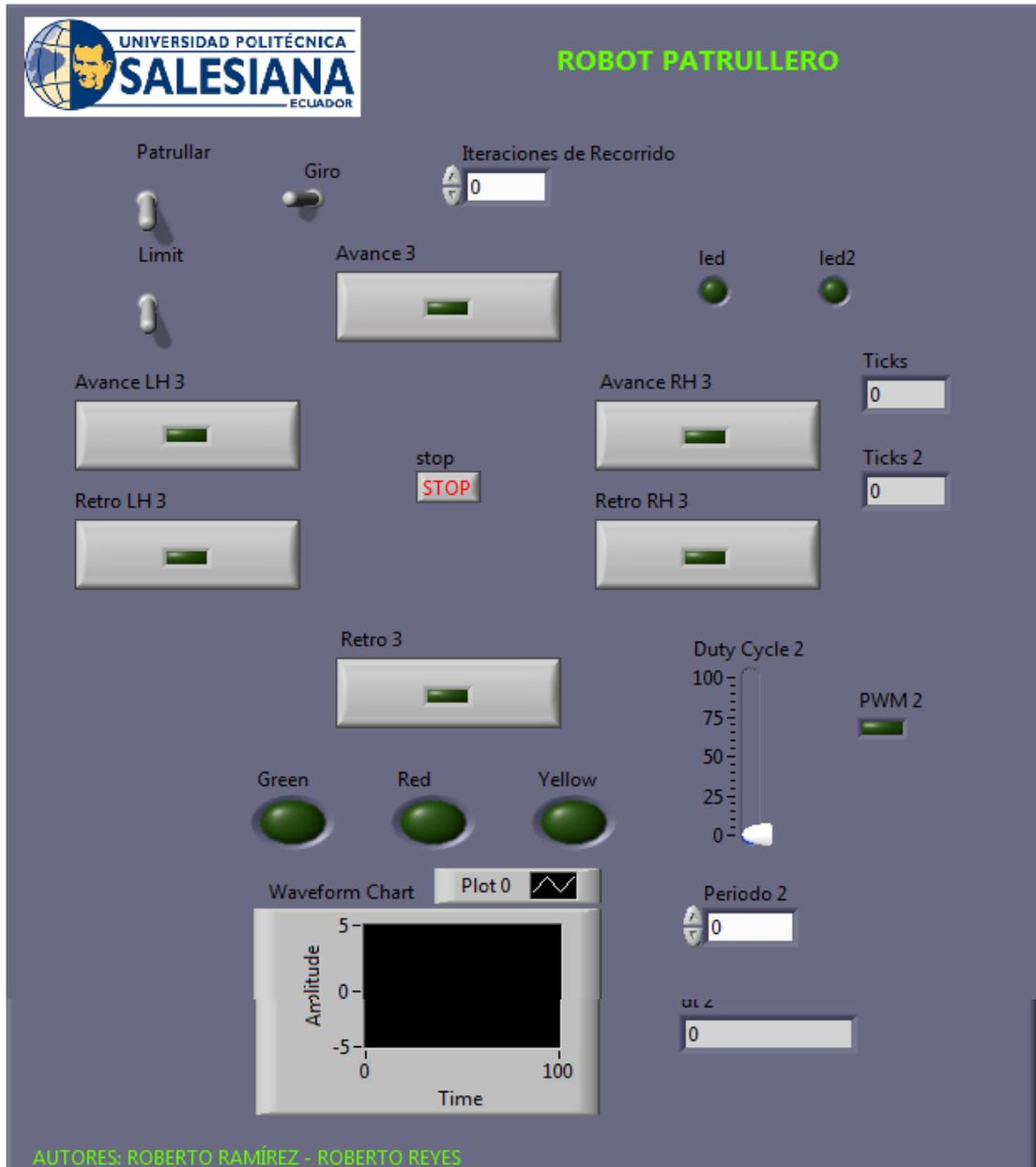


Figura 4.49 Panel Frontal del VI “Patrullar 2RT”. Aquí podemos observar el Switch de control Patrullar para que empiece a trabajar, además del control de forma manual y control PWM para los motores.

En la Figura 4.49 se observa el diagrama de control que permite realizar la función de “Patrullero” según la práctica, aquí podemos observar que se ha tomado como referencia el VI LabVIEW FPGA “Patrullar”, tomando cada una de las señales para el control de esta práctica.

En el desarrollo de la programación se logró comprobar como el robot móvil realiza su función de “Patrullero”, ida y regreso, pero se pudo comprobar que el

sistema no es tan estable, ya que a partir de la 3 vuelta el robot móvil genera una pequeña desviación haciendo que se desvíe parcialmente de la trayectoria, a causa de factores externos que no tienen nada que ver con el desarrollo de la práctica.

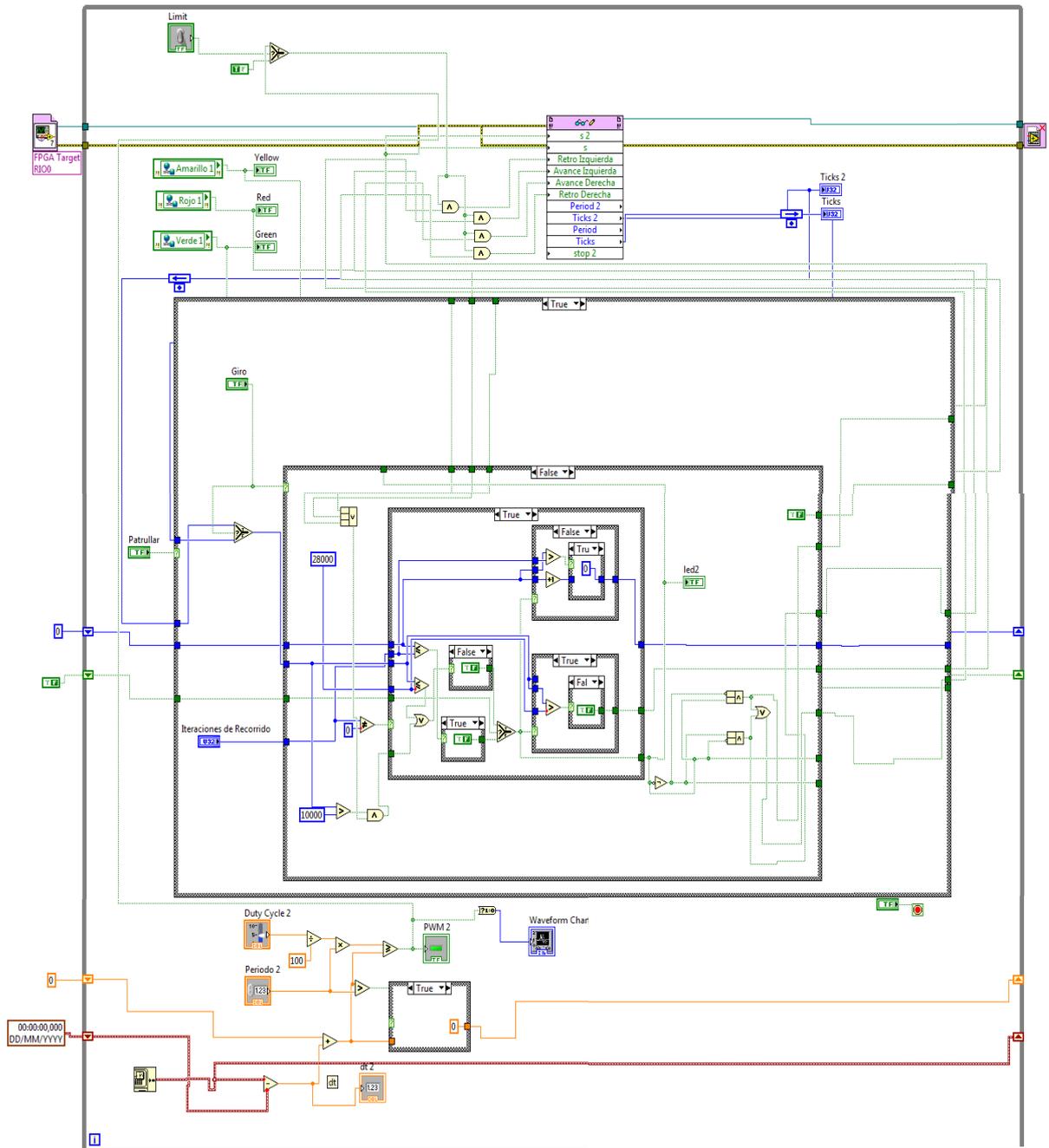


Figura 4.50 Diagrama de control del VI Real Time “Patrullar RT”. Programa que permite controlar el robot móvil para que cumpla la función de patrullaje según las condiciones de la práctica.



Figura 4.52 Robot Móvil trabajando como Patrullero. Es esta práctica el robot desarrolla un movimiento recto de ida y vuelta haciendo las veces de patrullaje.

5. CONCLUSIONES

- En el desarrollo de esta práctica se aprendió a trabajar con el Encoder, a detectar los pulsos generados y lograr el control de los giros a través de ellos.
- Se logró comprobar lo difícil que es lograr que el robot mantenga una trayectoria recta, se concluyó que en este proceso afecta varias condiciones.
- Los motores DC aún siendo nuevos no llegan a tener igualdad exacta en revoluciones.
- El sistema mecánico puede afectar directamente en la trayectoria del robot debido a que si no se encuentra totalmente alineado generara desviación.
- La rueda loca pueda afectar indirectamente en los giros del robot haciendo que se desvíe proporcionalente.

6. RECOMENDACIONES

- Al momento de la instalación de los encoders es necesario tener en cuenta que estos deben estar alineados al eje del motor, podría generar señales de ruido y rozamientos que pueden afectar el desempeño de los motores.

4.6 GUÍA DE LABORATORIO 5

1. TEMA

IMPLEMENTACIÓN DE UN ROBOT ANTI-CAÍDAS

2. OBJETIVOS

2.1 GENERAL

- Hacer que el robot se mueva por encima de una mesa y que cuando se asome al borde retroceda y evite caerse

2.2 ESPECÍFICOS

- Diseñar varios subVIs que permitan leer los sensores de de ultrasonido HC-SR04 a la vez.
- Realizar VIs en LabVIEW FPGA y LabVIEW Real Time tomando los datos enviados por los sensores de ultrasonido que midan la variación de distancia del robot a la mesa donde estará ubicado.

3. MARCO TEÓRICO

Como parte principal de este proyecto se maneja también el sensor de ultrasonido HC-SR04 el cual permite medir la distancia con algún objeto a través de la generación de pequeñas ondas transmitidas y recibidas por el sensor.

Para conocer más de este tema consultar consutar Guía de Laboratorio 2. Adquisición de variables de entorno.

4. DESARROLLO

Para el desarrollo de esta práctica, interviene directamente el diseño de la estructura del robot móvil, el cual fue diseñado exactamente para ser colocados cada uno de estos sensores en puntos estratégicos en la parte inferior del robot móvil para la detección de la distancia.

En la siguiente Figura 4.52 se puede observar la ubicación de los sensores de ultrasonido para el desarrollo de esta práctica.

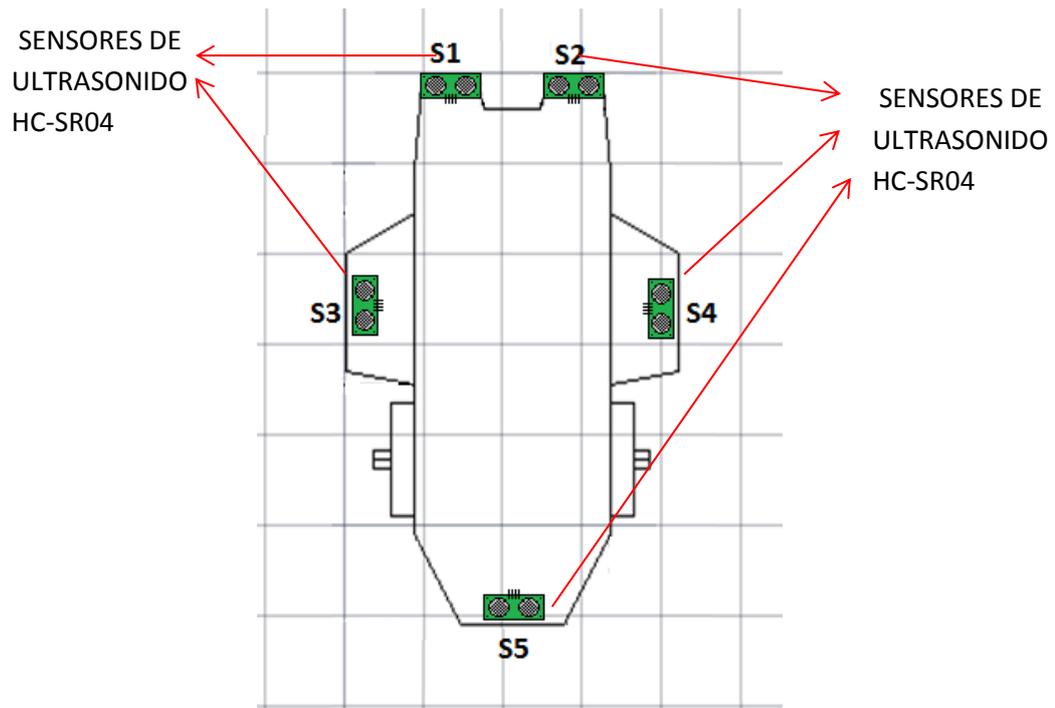


Figura 4.52 Ubicación de sensores de ultrasonido. Colocación de 5 sensores de ultrasonido HC-SR04 para el desarrollo de la practica Robot Anti-caídas.

Ubicado los sensores de ultrasonido de manera estratégica como se puede observar en la Figura 4.52, sucederá que cuando el robot móvil empiece a desplazarse sobre alguna mesa o sobre algún lugar donde haya un desnivel con referencia al piso, cuando proceda acercarse algún extremo de la mesa los sensores detectarán los cambios de diferencia en la distancia con referencia a la mesa donde se encuentre ubicado e inmediatamente retomará otro curso para evitar la caída.

A continuación se procede a desarrollar un nuevo proyecto a través de la ventana Getting Started para proceder con el diseño de la práctica a la cual llamaremos “Anticaídas”.

4.1 Programación en LabVIEW FPGA

Primero se empieza desarrollando los VIs que permitan inicializar los sensores de ultrasonido HC-SR04, estos VIs pasaran a ser subVIs para el desarrollo de la práctica en LabVIEW FPGA. (Ver Figura 4.54)

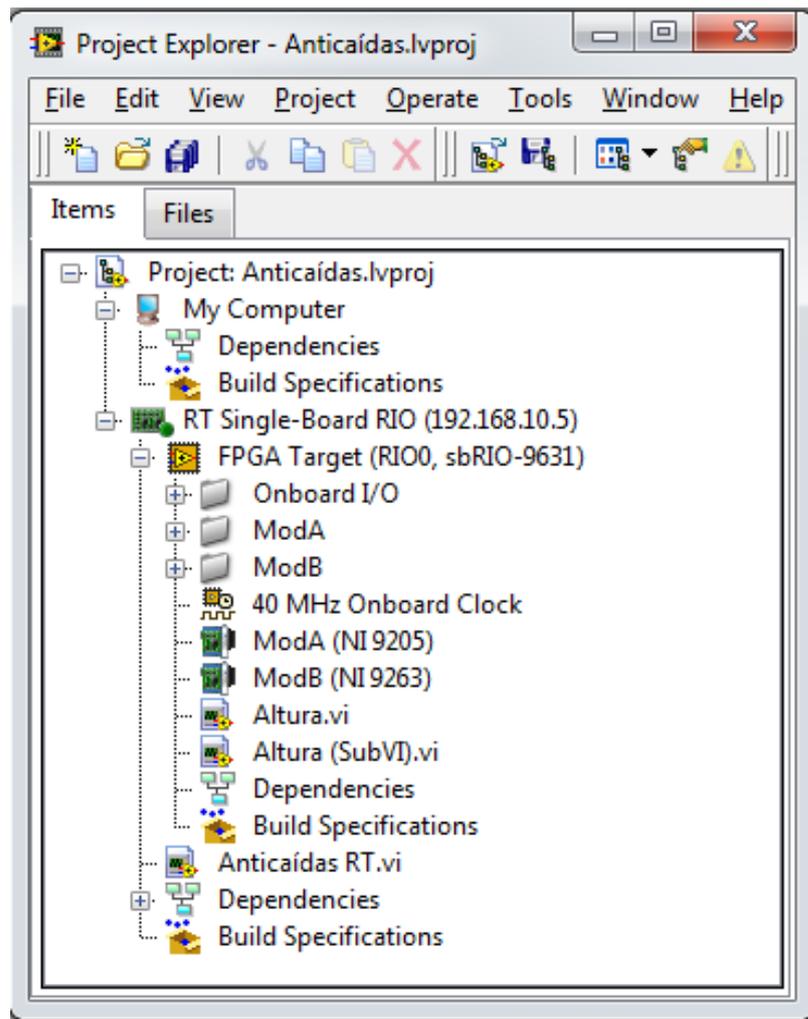


Figura 4.53 Nuevo Proyecto Anticaídas para el desarrollo de la práctica. A través del Project Explorer podemos observar la raíz de todo el proyecto desarrollados tanto en LabVIEW FPGA como en LaBVIEW Real Time.

Un vez desarrollado los subVIs de inicialización de los sensores de ultrasonido, se procede a desarrollar nuestro programa en LabVIEW FPGA para la adquisición de la señales de los sensores. (Ver Figura 4.55)

Como se puede observar en la Figura 4.56, la programación está diseñada directamente para la adquisición de datos enviados por los 5 sensores de ultrasonido, señales que ingresan a los subVIs para la inicialización de los sensores y cuyos datos de salida son tratados matemáticamente para proporcionar el valor exacto de la distancia. Estos datos son visualizados a través de los indicadores mostrados en el panel frontal.

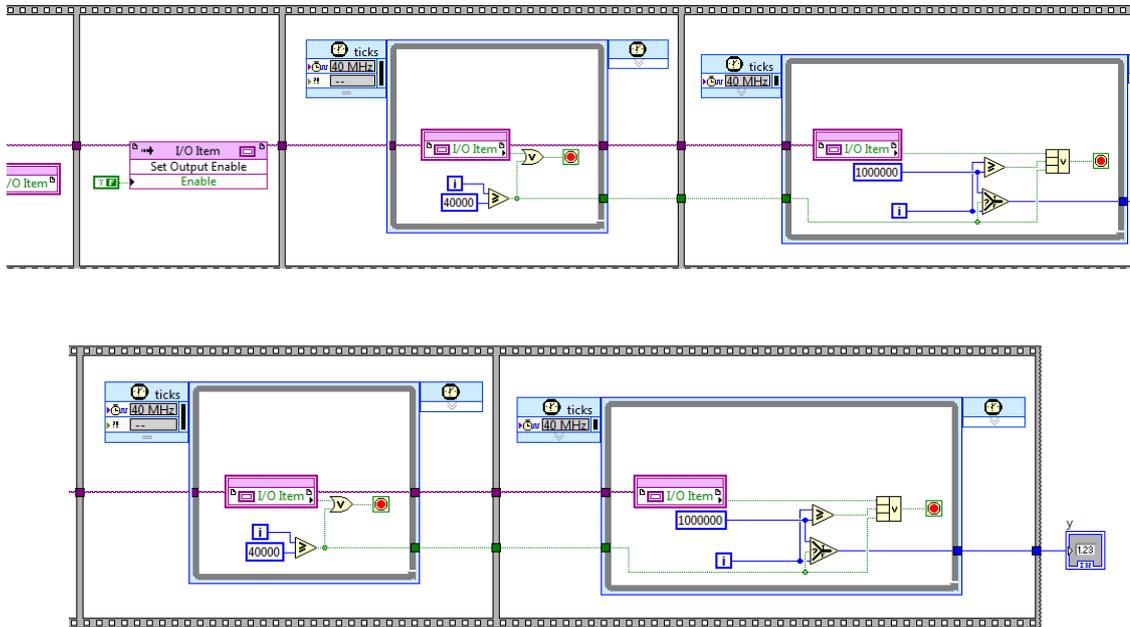


Figura 4.54 VIs para la inicialización del sensor de ultrasonido HC-SR04. Este programa genera pulsos para las entradas Echo y Trigger del sensor de ultrasonido para inicializarlo antes de la toma de lectura.



Figura 4.55 Panel Frontal del VI “Altura”. El panel frontal está diseñado para medir los valores de distancia del robot móvil con respecto al piso donde se encuentra ubicado el robot móvil.

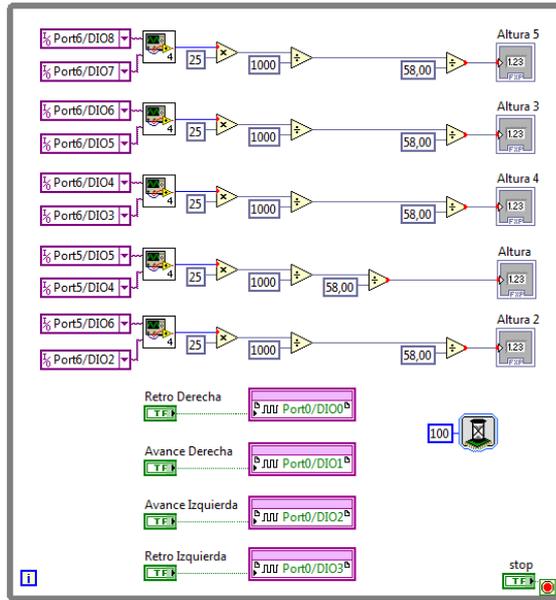


Figura 4.56 Diagrama de Control del VI “Altura”. Programación diseñada para la adquisición de valores de distancia de los sensores de ultrasonido.

4.2 Programación en LabVIEW Real Time

Una vez desarrollado el programa en LabVIEW FPGA se procede a la adquisición de estos datos tomando como referencia el VI “Altura” de la Figura 4.57, para desarrollar mayor eficacia a través de LabVIEW Real Time.



Figura 4.57 Panel Fronal del VI “Anticaídas RT”. Panel Frontal desarrollado para mostrar la altura de cada sensor de ultrasonido además de poder controlar el robot de forma manual o automática.

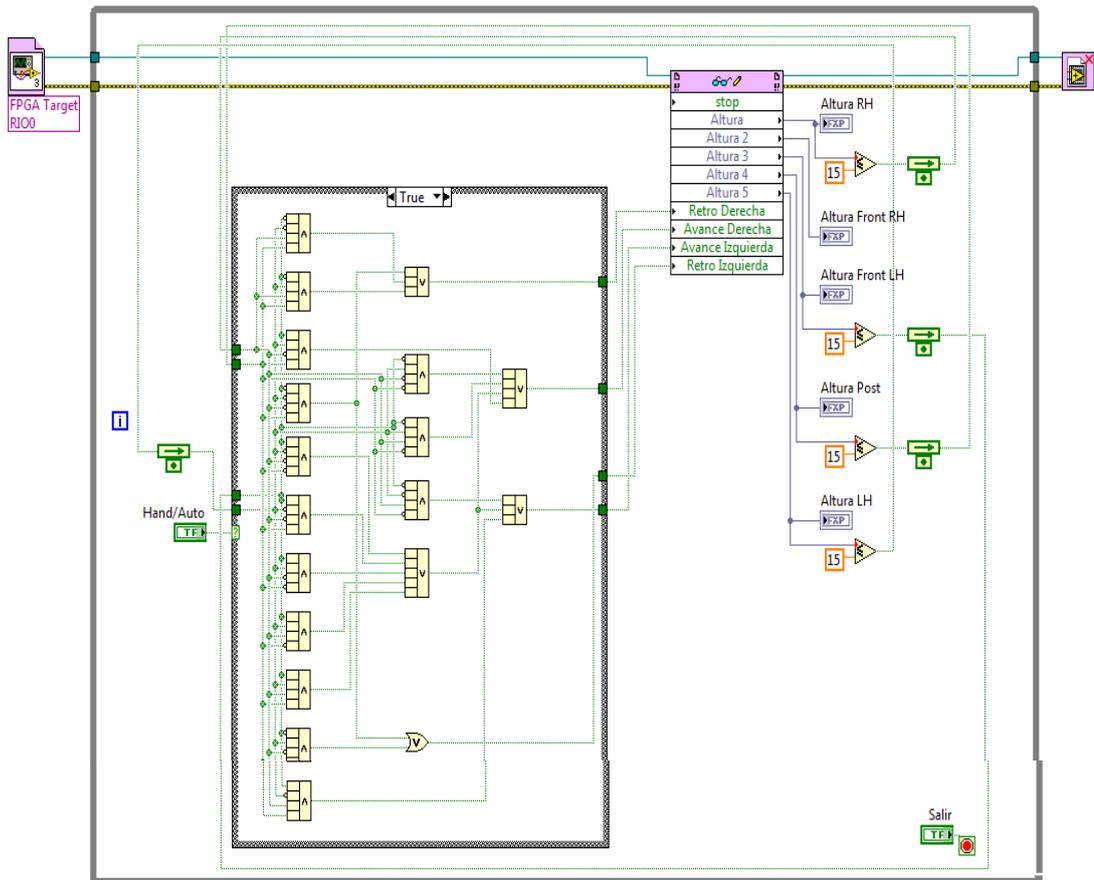


Figura 4.58 Diagrama de control del VI “Anticaídas” con Case Structure “True”. Diagrama de programación que muestra el control Manual del robot móvil con la estructura Case Estructure en Verdadero (True).

Las Figuras 4.58 y 4.59, muestran la programación en LabVIEW Real Time, la programación se basa tomando como referencia la VI “Altura” de LabVIEW FPGA para acceder a los puertos I/O de la tarjeta NI sbRIO-9631, cuyos datos ingresan a un Case Estructure donde manejaremos dos condiciones:

- El estado Manual permite manipular el robot móvil en virtud de los de los sensores de ultrasonido, operado manualmente no permite que el robot móvil se caiga del piso o de la mesa donde se encuentra ubicado.
- El estado Automático permite que el robot móvil se desplace de manera autónoma por una mesa o una base evitando que se caiga de la mesa donde se encuentra ubicado.

Una vez que los datos manejados pasan por el Case Estructure, estos datos van a los puertos de la tarjeta NI sbRIO para el control del robot móvil.

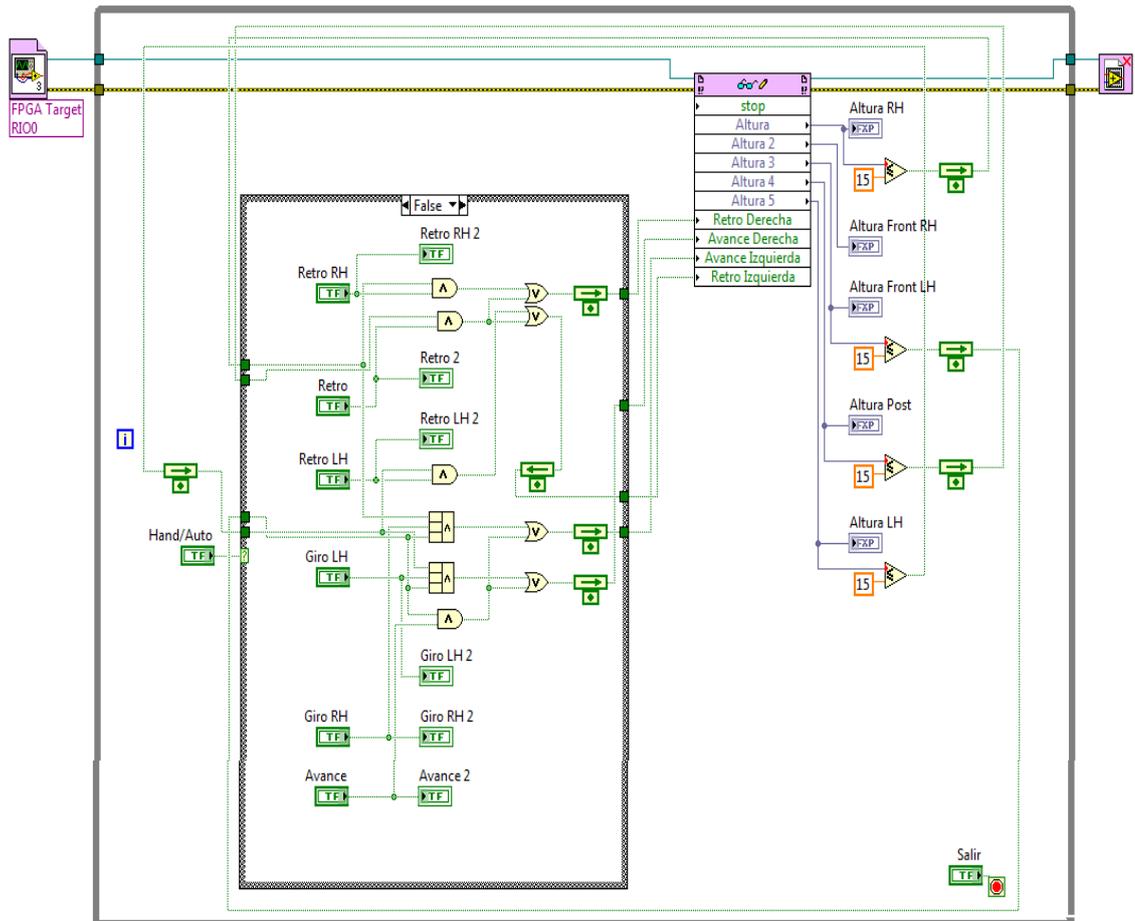


Figura 4.59 Diagrama de control del VI “Anticaídas” con Case Structure “False”. Diagrama de programación que muestra el control Automático del robot móvil con la estructura Case Estructure en Falso (False).



Figura 4.60 Imagen del sensor de ultrasonido ubicado en el robot móvil. Este es uno de los cinco sensores de ultrasonido que tiene el robot móvil, todo ubicados en la parte inferior.



Figura 4.61 Robot Móvil trabajando como Anti-caídas. La imagen muestra como el robot evita caerse al momento de desplazarse dentro de una mesa o una base similar.

5. CONCLUSIONES

- En el desarrollo de esta práctica se aprendió a manipular el sensor de ultrasonido HC-SR04, conociendo como inicializarlo y como adquirir sus datos de medición.
- Se desarrollo la programación basada lógica booleana, desarrollando tablas de verdad (1 y 0), para lograr los movimientos deseados en el robot móvil.
- Tambien se aprendió como manejar LabVIEW FPGA y LabVIEW Real Time en el desarrollo de la programación.

6. RECOMENDACIONES

- Se recomienda aislar los sensores de ultrasonido antes de ubicarlos en el robot móvil para evitar que se genere algún corto en el sistema.
- Se recomienda probar cada unos de los sensores de ultrasonido antes de iniciar el programa para evitar que puedan ocurrir caídas innecesarias que puedan afectar a la estructura del robot móvil y a la tarjeta NI sbRIO-9631.

4.7 GUÍA DE LABORATORIO 6

1. TEMA

IMPLEMENTACIÓN DE UN ROBOT SEGUIDOR DE LÍNEA

2. OBJETIVOS

2.1 GENERAL

- Hacer que el robot siga una línea negra dibujada en el suelo trazando una trayectoria.

2.2 ESPECÍFICOS

- Conocer el funcionamiento de los distintos sensores de reflejo.
- Aprender la lógica de programación para los diferentes tipos de robots seguidores de línea.

3. MARCO TEORICO

3.1 Sensor infrarrojo QRD114

Este es un sensor óptico compuesto por un diodo infrarrojo y un fototransistor para detectar el reflejo de luz infrarroja y así poder detectar líneas negras sobre blanco dentro de un rango que va desde 0.5 a 1 cm. (Ver Figura 4.62)

Características:

- Salida con fototransistor
- Filtro de luz de día
- Empaque compacto
- Detección de objetos difusos

3.2 Banco de Sensores QRD1114 Idetec

El banco de sensores QRD1114 está formado por 7 sensores, los cuales permiten una rápida detección en la identificación de colores (blanco o negro) y en los seguidores de línea. (Ver Figura 4.63)

Características

- Posee 7 sensores infrarrojo QRD1114
- Independiente con conector de 3 hilos tipo extensión servomotor
- Distancia de superficie reflectora (8mm) máximo
- Alimentación +5Vcc



Figura 4.62 Sensor Óptico QRD1114. Sensor óptico utilizado para en robots para proyectos de seguidores de línea. ROBODACTA.MX. (2015). QRD1114 Sensor Optico Reflectivo. Obtenido de http://www.robodacta.mx/index.php?dispatch=products.view&product_id=6666

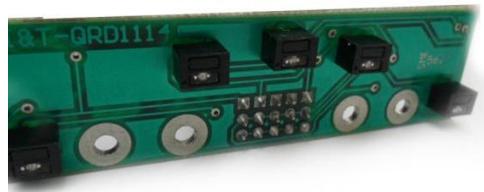


Figura 4.63 Banco de sensores ópticos QRD1114 de Idetec. Tarjeta usa para trabajar con robots seguidores de línea de alta complejidad. IDETEC. (2015). BANCO QRD1114 I&T 02. Obtenido de <http://www.ideastechnology.com/?q=node/126>

4. DESARROLLO

Antes de comenzar el desarrollo del programa es necesario conocer las conexiones del banco de sensores ópticos QRD1114, antes de conectarlo a la tarjeta NI sbRIO-9631. (Ver Figura 4.64)

Una vez conectado cada uno de los pines del sensor a la tarjeta NI sbRIO-9631, estamos listos para empezar a desarrollar el programa comenzando en LabVIEW FPGA, vamos a crear un nuevo proyecto al cual llamaremos “Seguidor de Línea”.

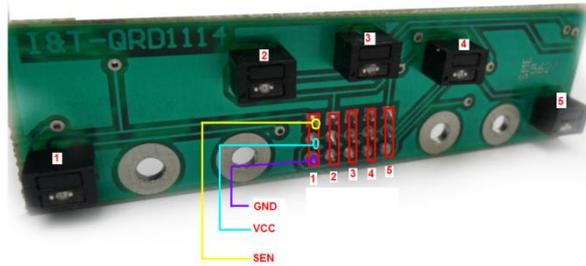


Figura 4.64 Conexión de pines del banco de sensores ópticos QRD114 de Idetec. Aquí observamos como se conecta los pines de Vcc, GND, S1 hasta S7. IDETEC. (2015). BANCO QRD1114 I&T 02. Obtenido de <http://www.ideastechnology.com/?q=node/126>

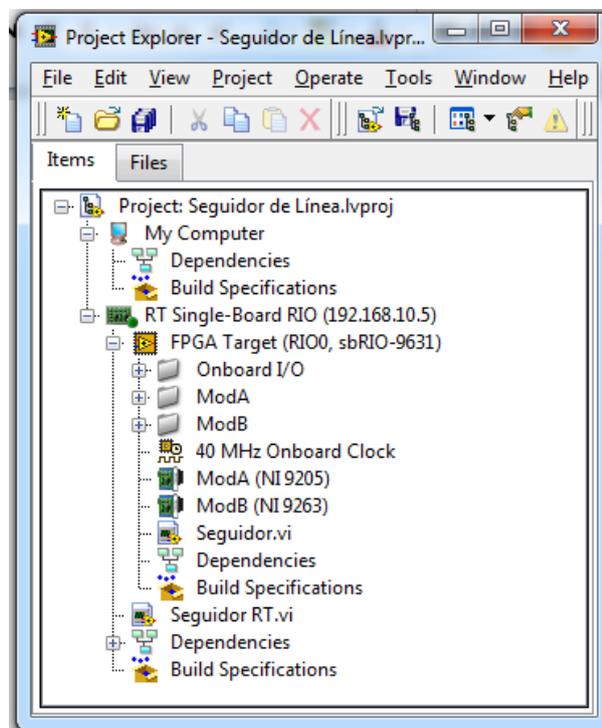


Figura 4.65 Proyecto “Seguidor de Línea”. Este proyecto muestra los programas realizados tanto en LabVIEW FPGA (FPGA Target) como en LabVIEW Real Time (RT Single-Board RIO).

4.1 Programación desarrollado en LabVIEW FPGA

Para empezar la programación con LabVIEW FPGA en donde se tiene acceso directo a los puertos I/O de la tarjeta NI sbRIO-9631, una vez definido los puertos con los que se va a trabajar se empieza a realizar la programación.

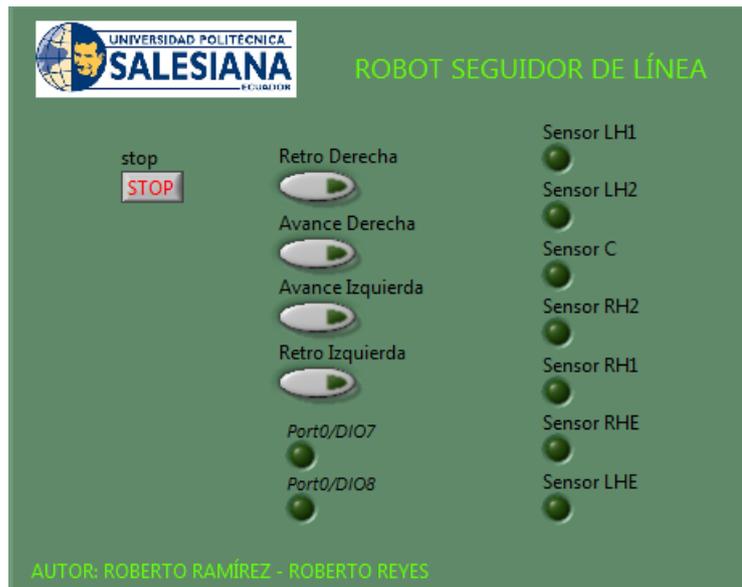


Figura 4.66 Panel Frontal del VI “Seguidor” en LabVIEW FPGA. Panel Frontal donde se puede visualizar a través de los leds que sensor de ultrasonido se encuentra activo.

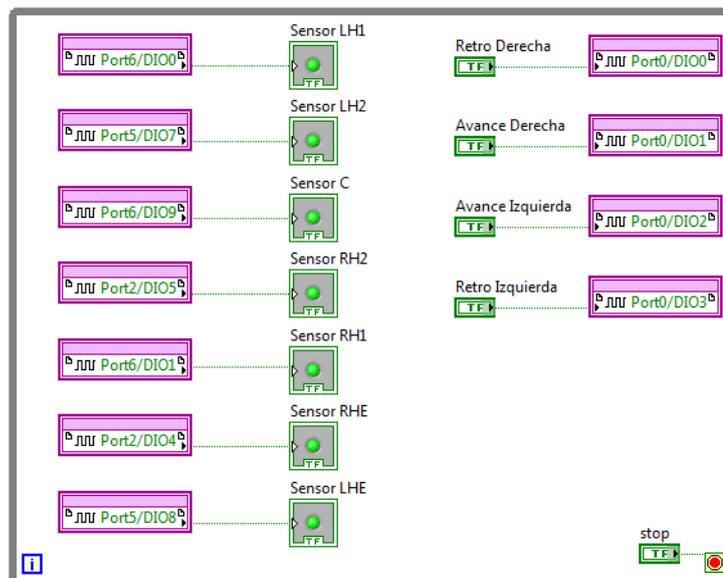


Figura 4.67 Diagrama de control del VI “Seguidor” en LabVIEW FPGA. En este diagrama observamos los controles y los leds conectados directamente a los puertos I/O de la tarjeta NI sbRIO-9631.

La programación del robot móvil en LabVIEW FPGA se basa básicamente en la obtención de los puertos I/O de la tarjeta NI sbRIO-9631, para poder ser visualizados

a través de Leds en el panel frontal. Estos datos serán usados para la programación en LabVIEW Real Time.

4.2 Programación desarrollado en LabVIEW Real Time

Teniendo lista la programación de puertos en LabVIEW FPGA, se procede a usar este VI como referencia para la obtención de los puertos I/O de la tarjeta y desarrollar la programación en LabVIEW Real Time.

A continuación se observa el desarrollo de la programación.

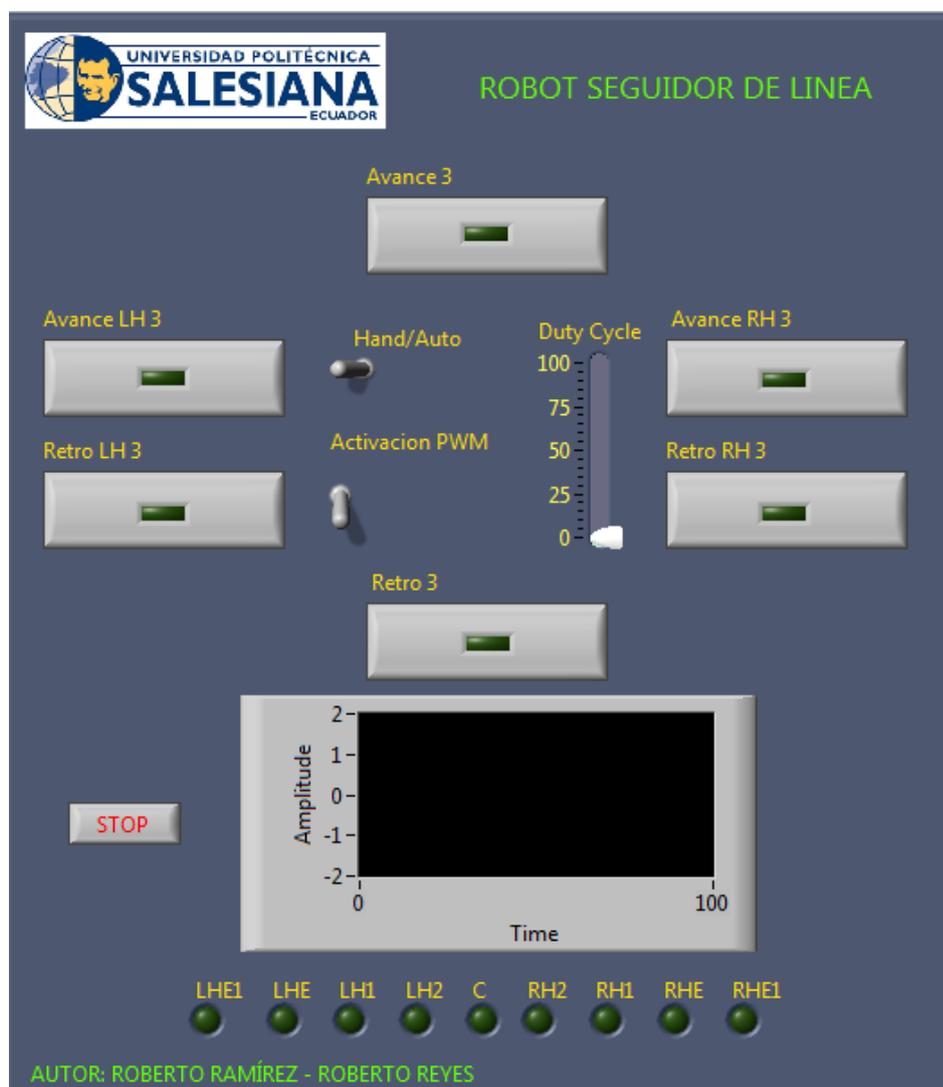


Figura 4.68 Panel frontal del VI Real Time “Seguidor RT”. En el panel frontal se puede observar el control manual y automático del robot y la activación de los sensores a través de los leds, también se tiene la activación del PWM a los motores visualizado a través del Waveform Chart.

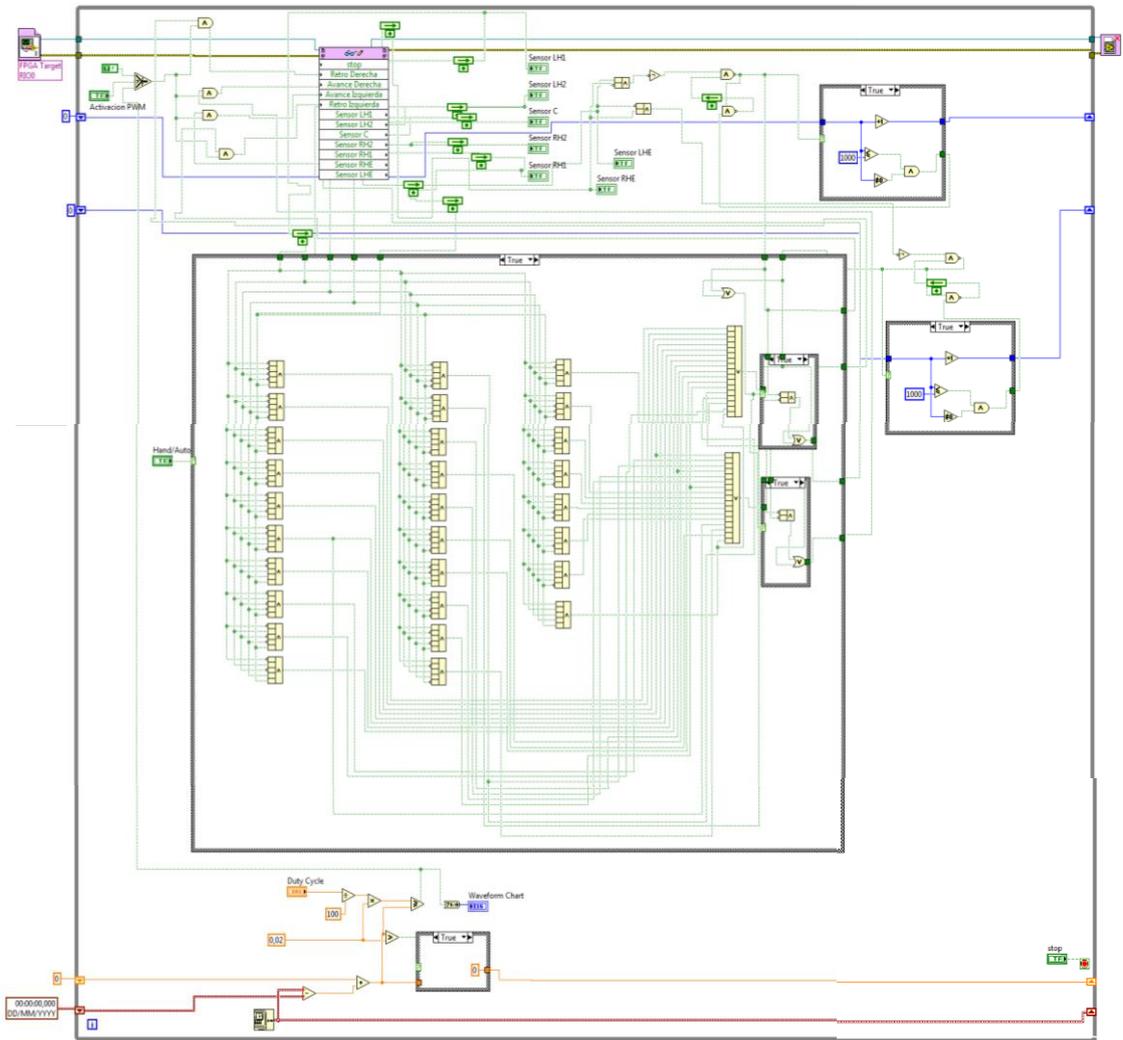


Figura 4.69 Diagrama de control del VI Real Time “Seguidor RT”. En la programación se puede observar varios Case Structure en opción Verdadero (True). Indicando el manejo del robot en modo manual/automático, sensores de ultrasonido de los extremos y otro adicional que maneja el PWM.

En el diagrama de control de la Figura 4.69, se puede observar el control del robot móvil a través de LabVIEW Real Time, los datos obtenidos van directamente al Case Structure que controla el funcionamiento de modo manual/automático, en modo automático se maneja en base a una serie de combinaciones booleanas tomados en base a una tabla de verdad conectados a 5 sensores QRD1114, que maneja los movimientos del izquierda/derecha del robot móvil, adicional a ello en la parte superior de la programación se encuentran dos Case Structure adicionales para los

sensores de los extremos en el Banco de Sensores QRD1114, estos están programados para la detección de líneas más cerradas, y en la parte inferior se tiene la generación del control PWM que va directamente hacia los motores, y se activa mediante el switch de control llamado “Activación PWM”.

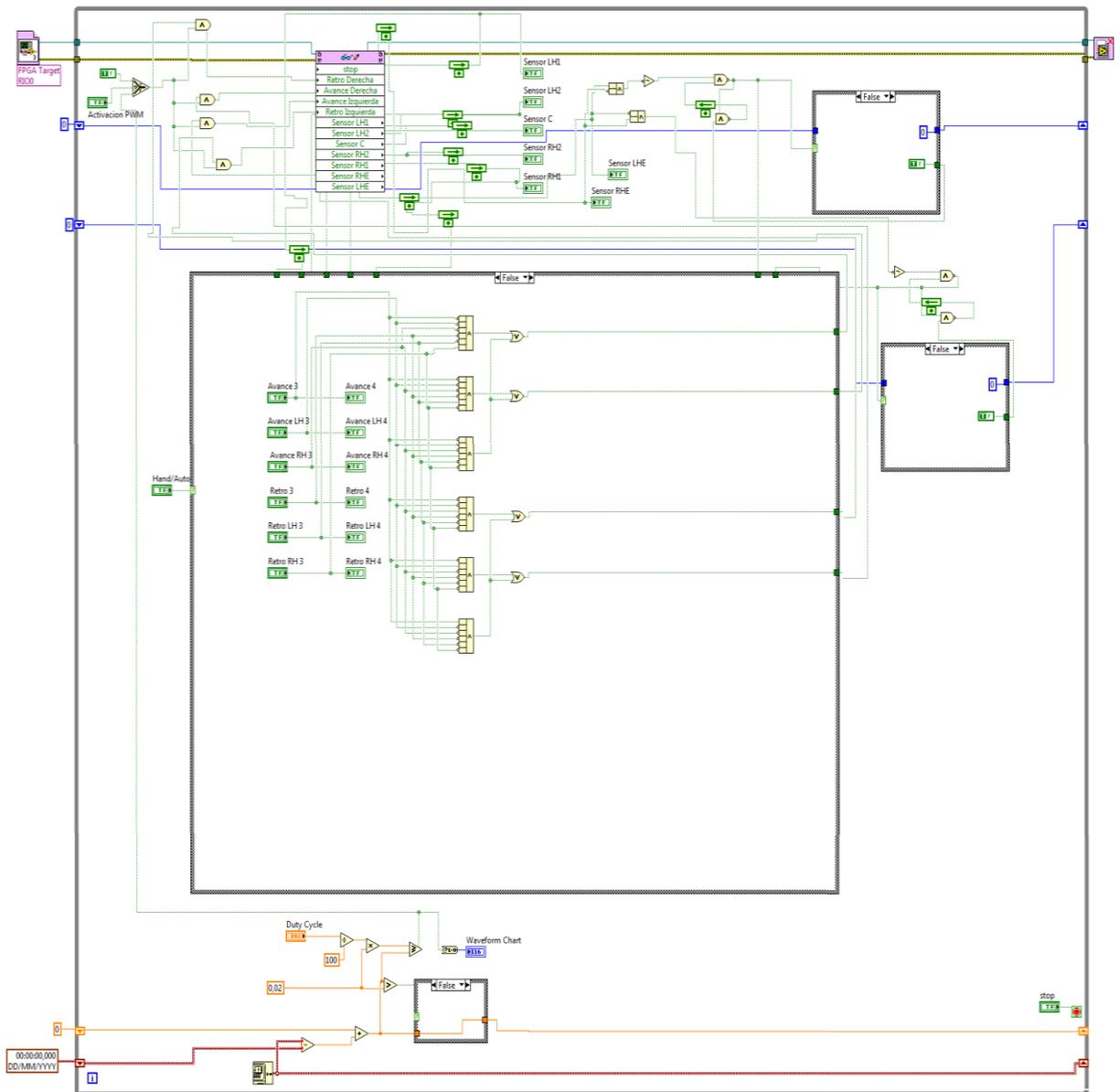


Figura 4.70 Diagrama de control del VI Real Time “Seguidor RT”. En la programación podemos observar varios Case Structure en opción Falso (False). Indicando el manejo del robot en modo manual/automático, sensores de ultrasonido de los extremos y otro adicional que maneja el PWM.

En la Figura 4.70, la programación maneja todos los Case Structute en False, el cual desactiva todo control automático y el control se lo realiza manualmente a través de pulsadores de control como se muestra en la Figura 4.57.

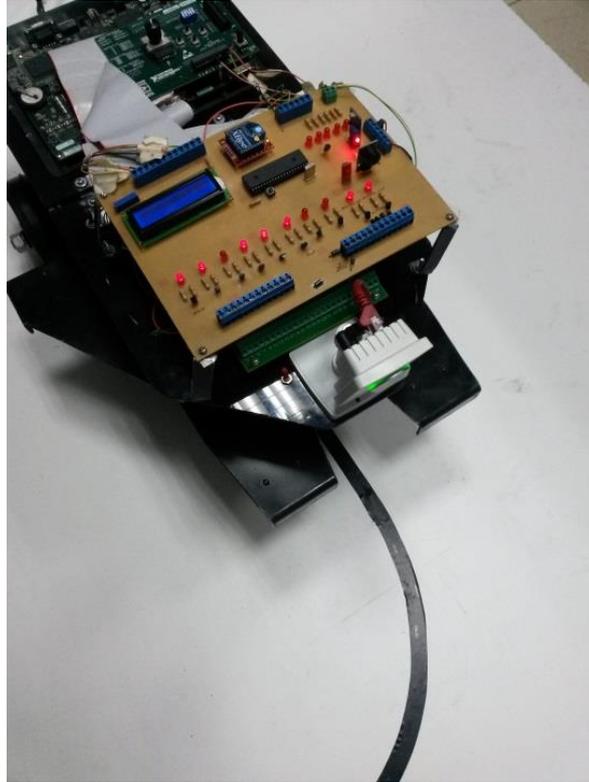


Figura 4.71 Robot Móvil trabajando en línea curva. En la figura observamos al robot móvil realizando un movimiento curso, guiado por la línea negra.

5. CONCLUSIONES

- En el desarrollo de esta práctica se aprendió a manejar un robot móvil guiado a través de una línea, sea esta recta o curva, diseñando un programa basado en tablas de verdad.
- Se conoció los errores que pueden generarse al desarrollar esta práctica que aunque parece sencilla su grado de complejidad aumentó en los movimiento de curvas.
- Para lo giros del robot móvil se optó por la instalación de 9 sensores en la parte inferior para una mejor detección de la recta.

6. RECOMENDACIONES

- Es necesario tener en cuenta al momento de la instalación del Banco de sensores QRD1114, colocarlos a una distancia no mayor a 8 mm., mayor a esto el sensor no detecta correctamente la línea generando errores al momento de trabajar el robot móvil.
- El seguidor de línea está diseñado para que el robot detecte líneas entre 90° a 270°, fuera de este rango el robot no desarrolla completamente.

4.8 GUÍA DE LABORATORIO 7

1. TEMA

IMPLEMENTACIÓN DE UN ROBOT NAVEGADOR

2. OBJETIVOS

2.1 GENERAL

- Lograr que el robot se desplace dentro de una habitación sin llegar a chocarse hasta que logre encontrar la salida.

2.2 ESPECÍFICOS

- Poner en uso los 6 sensores de ultrasonido instalados en todo el borde del robot móvil.
- Lograr que el robot se desplace por bordes y esquinas sin llegar a rozar con cualquier objeto.

3. MARCO TEÓRICO

Para esta práctica se usaran los sensores de ultrasonido HC-SR04 implementado en prácticas anteriores, para obtener más información de estos dispositivos revisar Guía de Laboratorio 2 del Capítulo 4.

4. DESARROLLO

Como ya se indicó anteriormente, para el desarrollo de esta práctica usaremos 6 sensores ubicados estratégicamente de manera que bordee todo el contorno del robot móvil para que pueda desplazarse sin ningún inconveniente. (Ver Figura 4.72)

Ubicado los sensores de ultrasonidos correctamente, es necesario empezar por desarrollar la programación necesaria para el desarrollo de la práctica. Se inicia creando un nuevo proyecto a través de la ventana Getting Started de inicio de LabVIEW, creando un nuevo proyecto el cual se llama “Navegador” y es con el que se empieza a trabajar.

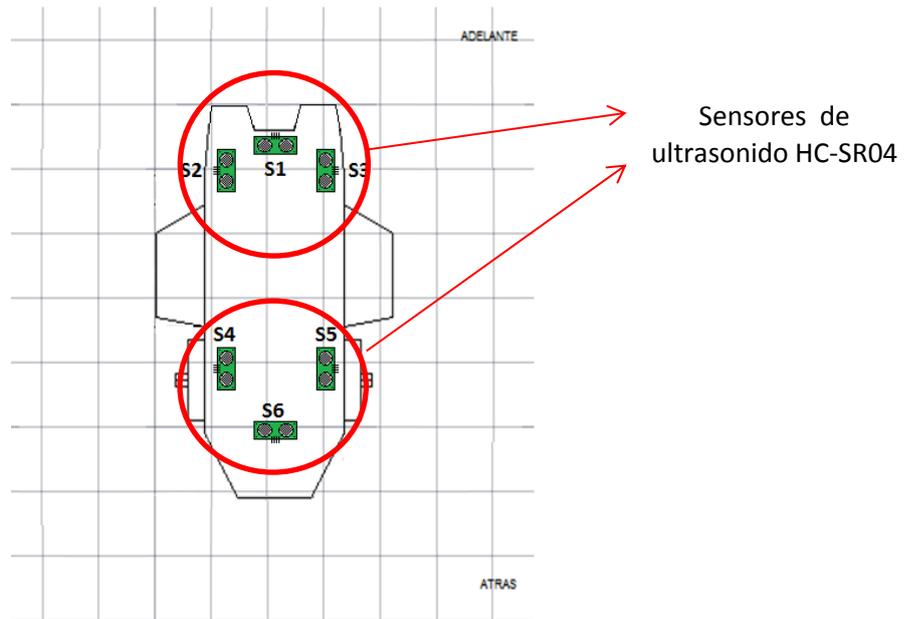


Figura 4.72 Ubicación de sensores de ultrasonido en los bordes del robot móvil. Cada sensor de ultrasonido está ubicado de tal manera que permita la detección de cualquier obstáculo para que lo evite.

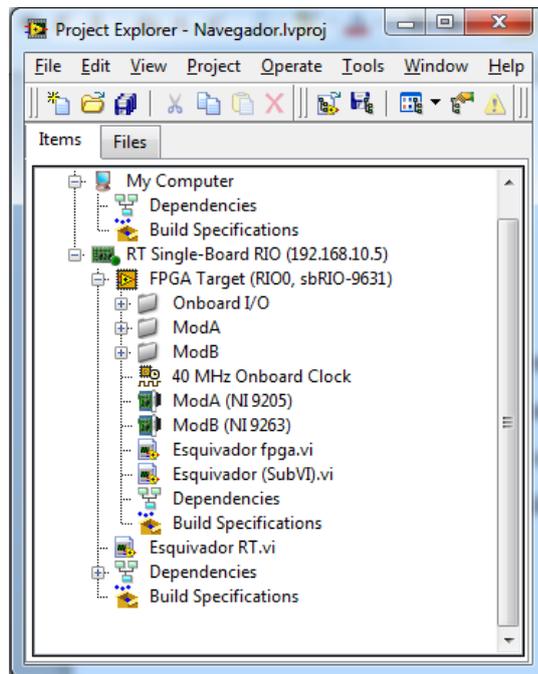


Figura 4.73 Proyecto “Navegador”. Figura que detalla el proyecto desarrollado en RT Single-Board RIO (LabVIEW Real Time) y FPGA Target (LabVIEW FPGA).

4.1 Programación desarrollada en LabVIEW FPGA

Una vez creado el proyecto se empezará la programación de la práctica en LabVIEW FPGA, para ello primero es necesario crear el programa de inicialización de los sensores de ultrasonido que servirán para manejar los 6 sensores de ultrasonido que tiene el robot móvil, todos serán usados como subVIs.

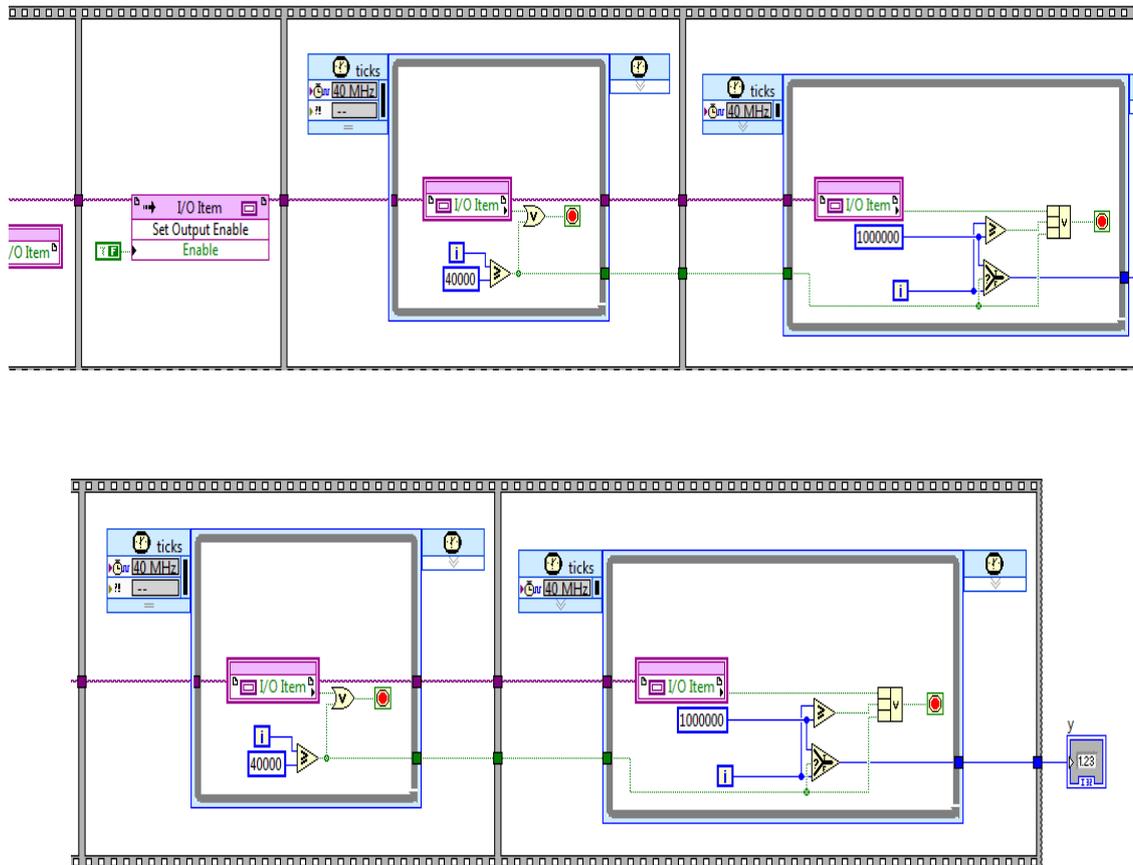


Figura 4.74 Diagrama de Bloques para la inicialización del sensor ultrasónico. En la imagen se detalla paso a paso la programación realizada al sensor para la inicialización antes de que empiece a tomar datos.

La Figura 4.74, permite inicializar el sensor de ultrasonido, luego serán usados como subVIs para inicializar cada uno de los 6 sensores que tiene el robot móvil, luego vamos a proceder a desarrollar la programación en LabVIEW FPGA.



Figura 4.75 Panel Frontal del VI LabVIEW FPGA “Esquivador fpga”. La figura muestra a través de indicadores la distancia tomada de cada uno de los sensores de ultrasonido, además del control manual del robot móvil.

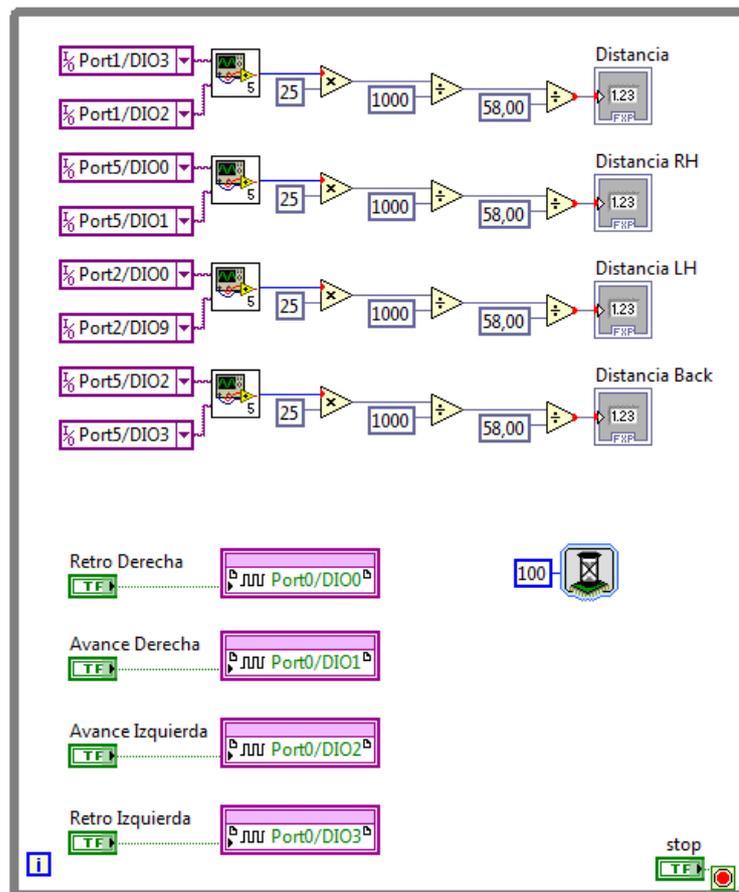


Figura 4.76 Diagrama de control del VI LabVIEW FPGA “Esquivador”. Diagrama que permite acceder a cada uno de los puertos I/O de la tarjeta FPGA NI sbRIO-9631 para manejar el robot móvil.

Como se observa en la Figura 4.75, el diagrama de control realizado permite obtener información de cada uno de los sensores de ultrasonido y los datos obtenidos son enviados a los indicadores numéricos, a su vez también se tiene el control manual del movimiento de los motores.

4.2 Programación desarrollado en LabVIEW Real Time

Una vez desarrollado el VI “Esquivador” en LabVIEW FPGA, se lo toma como referencia para adquirir los datos a ser manejado para el desarrollo del VI en LabVIEW Real Time.

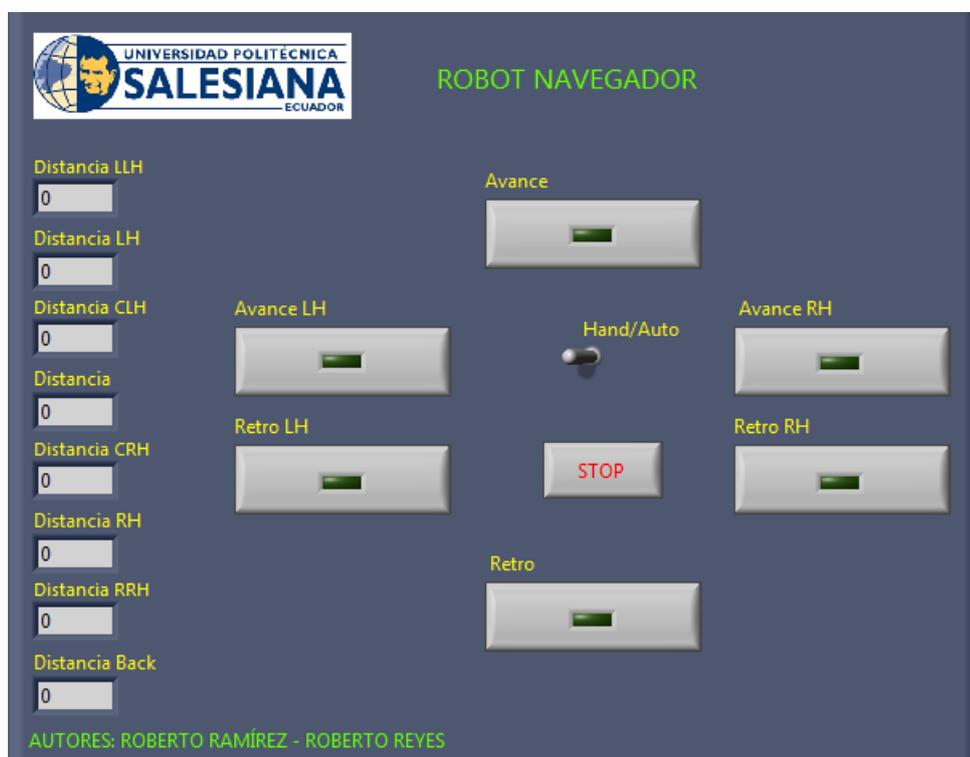


Figura 4.77 Panel Frontal del VI Real Time “Esquivador RT”. Panel que muestra la activación del programa en modo manual/automático y la medición de la distancia a través de los sensores de ultrasonido.

En la Figura 4.78, se puede observar en el Diagrama de Control el diseño de la programación realizada para evitar obstáculos de cualquier tipo, el programa funciona tomando como referencia el VI “Esquivador” diseñado en LabVIEW FPGA, una vez que adquiere los datos a través de los 6 sensores, estos datos son activados mediante el Switch de control Manual o Automático. En modo manual el robot móvil va a depender de los movimiento que realice el usuario para moverse,

mientras que en modo automático el robot adquiere los datos de distancia que le envían los sensores de ultrasonido, datos que serán procesados mediante código booleano a través de una tabla de verdad que le indicará para donde moverse según el objeto para evitar chocarse y pueda encontrar una salida.

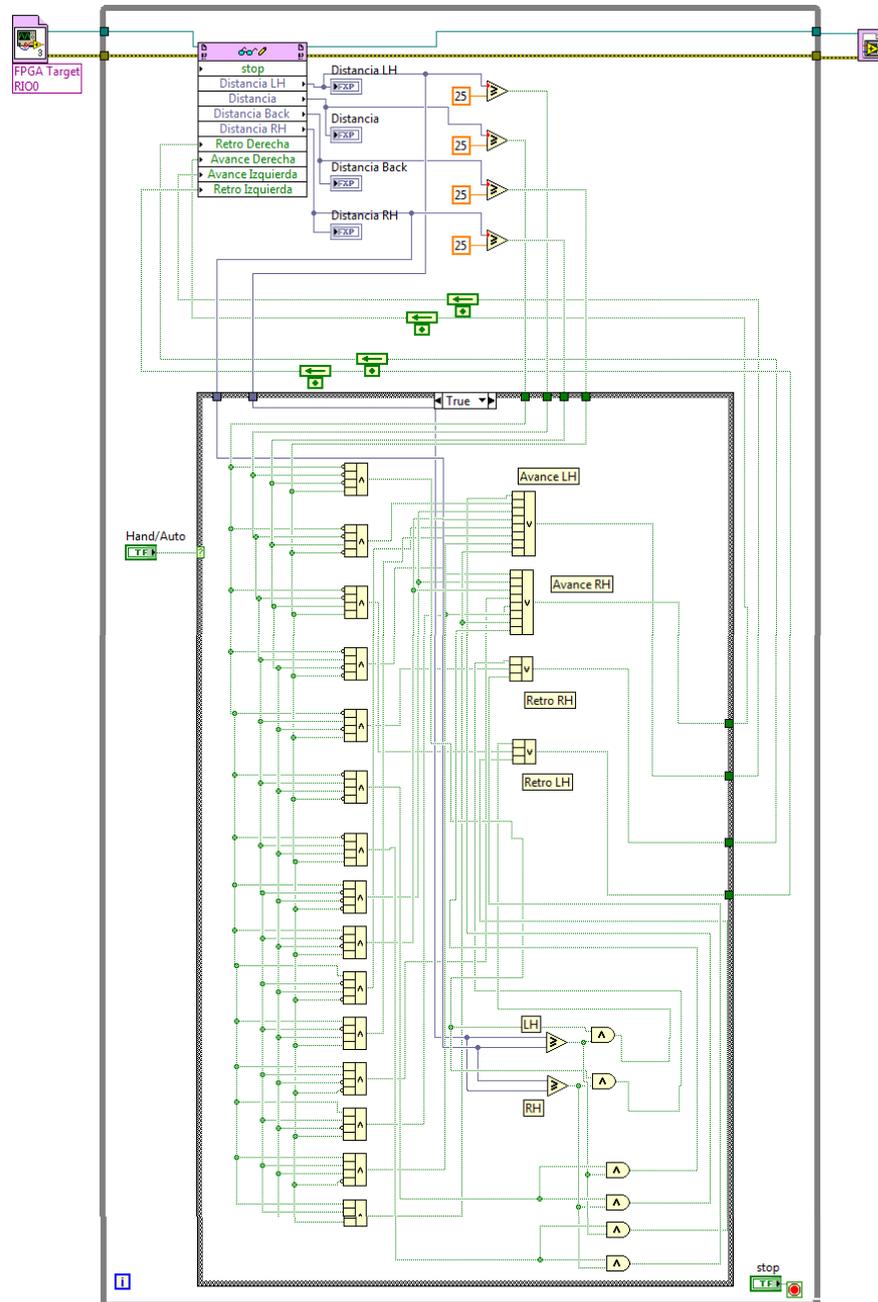


Figura 4.78 Diagrama de control del VI Real Time “Esquivador RT”. Figura que detalla la programación del robot móvil en LabVIEW Real Time para evitar obstáculos.

5. CONCLUSIONES

- En esta práctica se aprendió a manejar sensores de ultrasonido usando códigos booleanos y creando tables de verdad para manejar los movimientos del robot móvil.
- Cada sensor de ultrasonido requiere de inicialización para ser activado, antes de mandar la lectura de medición, para ello cada sensor se maneja con un subVI para su inicialización.

6. RECOMENDACIONES

- Se recomienda manejar voltajes de alimentación de +5Vcc, para que los sensores de ultrasonido puedan trabajar correctamente.

4.9 GUÍA DE LABORATORIO 8

1. TEMA

IMPLEMENTACIÓN DE UN ROBOT DE RESCATE

2. OBJETIVOS

2.1 GENERAL

- Lograr que el robot se desplace dentro de una determinada zona y logre identificar objetos de colores y dependiendo del color genera un pitido simulando una señal de auxilio.

2.2 ESPECÍFICOS

- Acoplar la red inalámbrica del robot móvil con la cámara inalámbrica AXIS M1011.
- Hacer el robot trabaje con una cámara que pueda identificar los colores, logrando aplicaciones en base a los colores que identifique.
- Lograr que el robot móvil evite chocarse con cualquier objeto sin importar el color que estos tengan, identificado el color realizar su aplicación.
- Conocer las herramientas necesarias que dispones de LabVIEW para el reconcimiento y procesamientos de imágenes.

3. MARCO TEÓRICO

3.1 CÁMARA DE RED AXIS M1011

Una cámara IP o “de red” puede describirse como una cámara y un ordenador combinados para formar una única unidad, gracias a lo cual emite las imágenes directamente a la red (Intranet o internet). Ésta cámara se conecta directamente a la red, como cualquier otro dispositivo- Una cámara de red dispone de su propia dirección IP y de funciones informáticas integradas para gestionar la comunicación con la red. Las cámaras IP permiten ver en tiempo real qué está pasando en un lugar, aunque se esté a miles de kilómetros de distancia. Son cámaras de video de gran calidad que tienen incluido un computador a través del que se conectan directamente a Internet. (AXIS Communications, 2013)

El dispositivo incorpora todo lo necesario poder visualizar imágenes a través de la red. Las cámaras de red disponen de software integrado para un servidor Web, servidor FTP, cliente FTP y cliente de correo electrónico.



Figura 4.79 Cámara de red AXIS M1011. La imagen nos proporciona diferentes ángulos de la cámara, donde podemos observar desde su base hasta conexión de red y alimentación. SINSEC. (2015). Cámaras de red. Obtenido de <http://www.edasistemas.com/servicios.php>

La cámara de red AXIS M1011 es perfecta para aplicaciones de robótica y sistemas de seguridad. Ésta cámara ofrece una conexión de red tanto por cable como inalámbrica (IEEE 802.11g). Ofrece una calidad de video excelente dentro de su clase (30 imágenes por segundo en una resolución VGA).

La AXIS M1011 ofrece múltiples transmisiones de video, configurables individualmente tanto en H.264 como en Motion JPEG y MPEG-4 Parte 2. Proporciona una resolución y frecuencia de imagen máxima en todas las compresiones. Gracias a su diseño funcional e inteligente, la cámara ofrece imágenes en movimiento sin ningún tipo de distorsión. (AXIS Communications, 2013)

Para el desarrollo de esta práctica se ha escogido la cámara AXIS debido a que National Instruments solo es compatible con cámaras IP fabricadas por las empresas Basler y AXIS. Estos tipos de cámaras son las que se maneja actualmente para CompactRIO y sbRIO.

4. DESARROLLO

Para el desarrollo de esta práctica se hace uso de dos dispositivos instalados en el robot móvil, se tiene la cámara inalámbrica y los sensores de ultrasonido HC-SR04 instalados a los bordes del robot móvil.

Antes de empezar a diseñar el programa es necesario conocer como instalar la cámara inalámbrica.

4.1 Asignación de una Dirección IP y configuración inalámbrica de la Cámara AXIS

Para asignar una dirección IP es necesario utilizar el software AXIS IP Utility, para que detecte automáticamente la Cámara AXIS que existe en la red. Para configurarla es necesario primero conectarla a una red para poder configurar los parámetros inalámbricos antes de efectuar la conexión a la red.

- Primero es necesario que la cámara esté encendida y conectada a la red, luego se necesita configurarle una dirección IP a la cámara conectándola a otra red, la cual le asigna una IP cualquiera.
- Luego se abre el AXIS IP Utility, el cual permite ver la configuración y la IP que le fue asignada a la cámara.

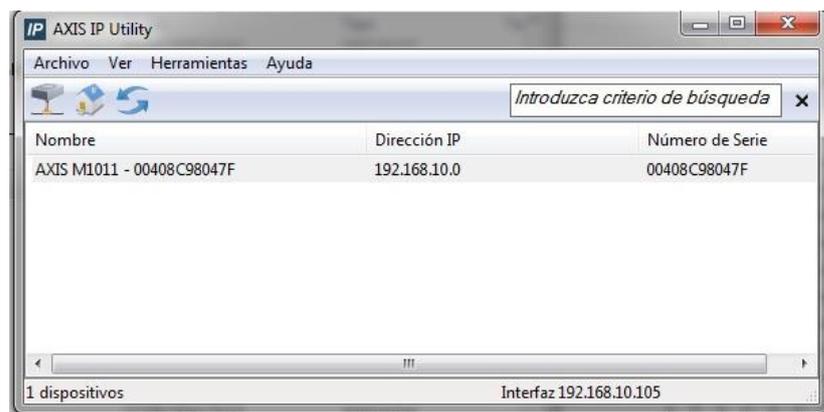


Figura 4.80 Ventana de configuración AXIS IP Utility. Aquí podemos observar el nombre del dispositivo con la dirección la cual le fue asignada.

- Una vez obtenida la IP entramos al navegador Web en donde ingresamos en la barra de búsqueda la dirección IP obtenida.

- Al acceder por primera a la página de configuración de la Cámara AXIS M1011, aparecerá el cuadro de diálogo “Configure Root Password” , aquí debemos establecer cuál será la contraseña del usuario predeterminado.
- Se accede nuevamente ingresando la contraseña establecida y nos aparecerá la ventana de configuración de la Cámara AXIS.

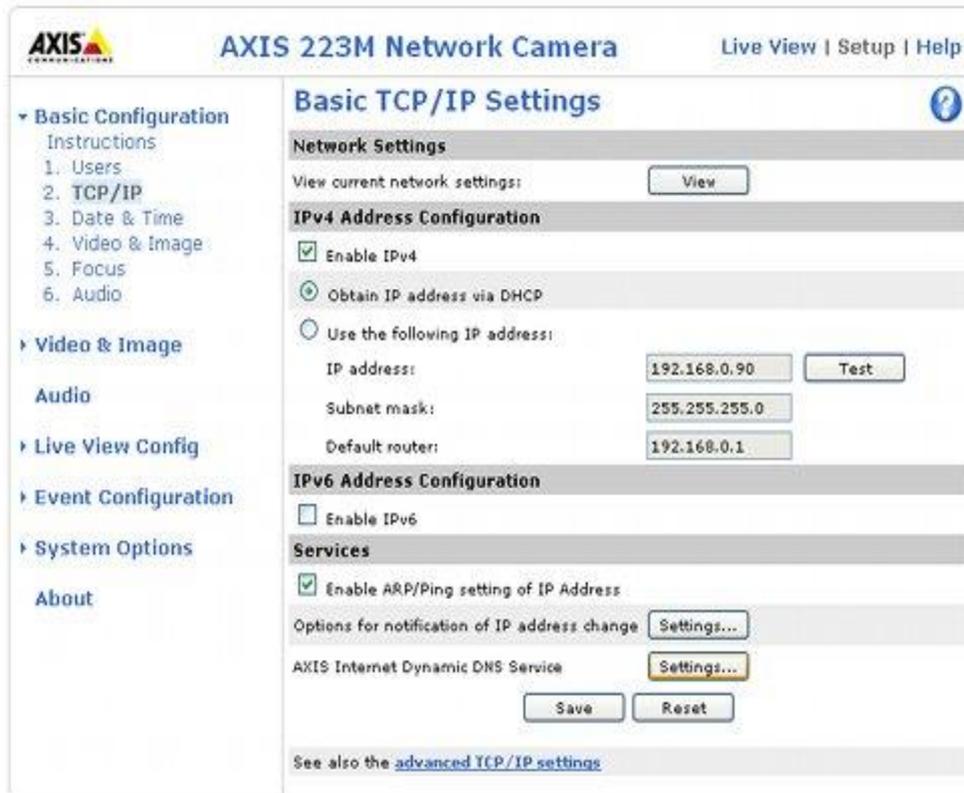


Figura 4.81 Ventana del Setup de la Cámara IP. Aquí asignaremos la dirección IP de la red del Robot Móvil para poder ejecutarlo.

- En la ventana del Setup de la Cámara se puede configurarla según las necesidades de la aplicación, en este caso cambiaremos la dirección IP para que la Cámara pueda trabajar con la red inalámbrica del Robot Móvil, se da clic en Save para guardar los cambios realizados.
- Una vez que se ha guardado la dirección IP, se procede a instalar la cámara AXIS a la red del Robot Móvil conectándolo a nuestro router Cisco. Una vez conectado procedemos a ejecutar el MAX y permite observar el dispositivo en la red del Robot Móvil. (Ver Figura 4.82)

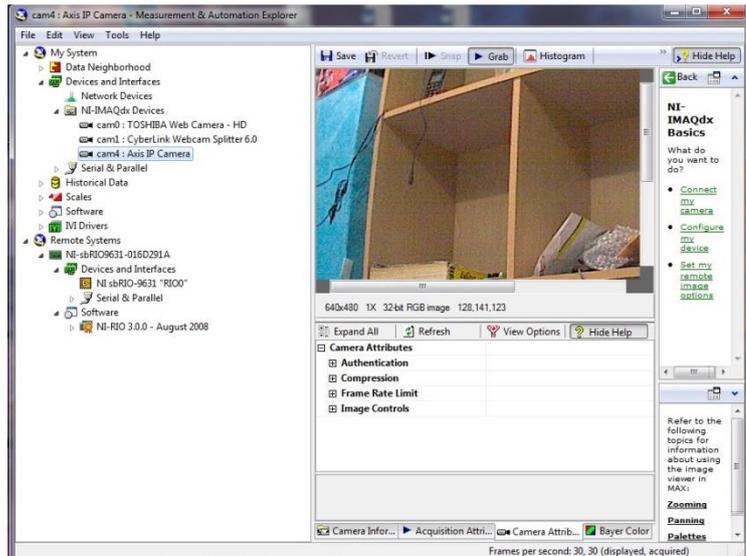


Figura 4.82 Configuración de la Cámara AXIS a través del MAX. En la ventana del MAX permite configurar la cámara AXIS M1011.

Instalada y configurada la cámara AXIS se procede al desarrollo de la práctica, es necesario crear un nuevo proyecto a través de la ventana de LabVIEW Getting Started. Proyecto al cual llamaremos “Reconocedor”.

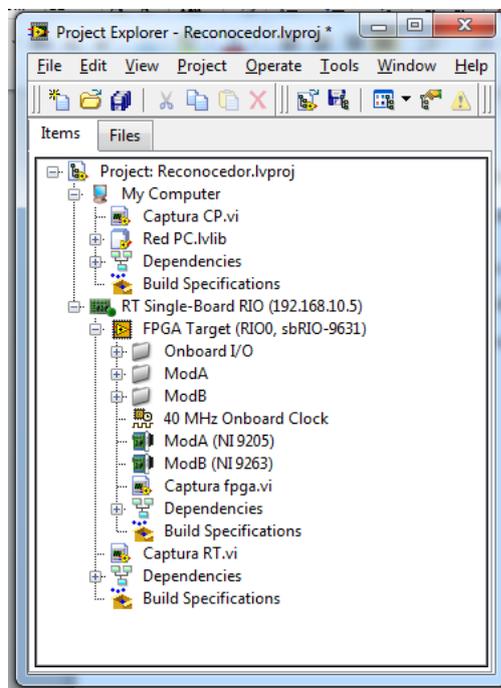


Figura 4.83 Proyecto en LabVIEW “Reconocedor”. En la figura se puede observar la raíz del proyecto usando los módulos LabVIEW FPGA y LabVIEW Real Time.

4.2 Programación desarrollada en LabVIEW FPGA

Para empezar con el desarrollo de la práctica se empieza trabajando en LabVIEW FPGA que permite acceder a los puertos I/O de la tarjeta NI sbRIO-9631.

Para desarrollar el programa es necesario adquirir los datos de los sensores de ultrasonido para adquirir los datos de distancia. Para ello se crea un subVI que permita inicializar los sensores de ultrasonido.

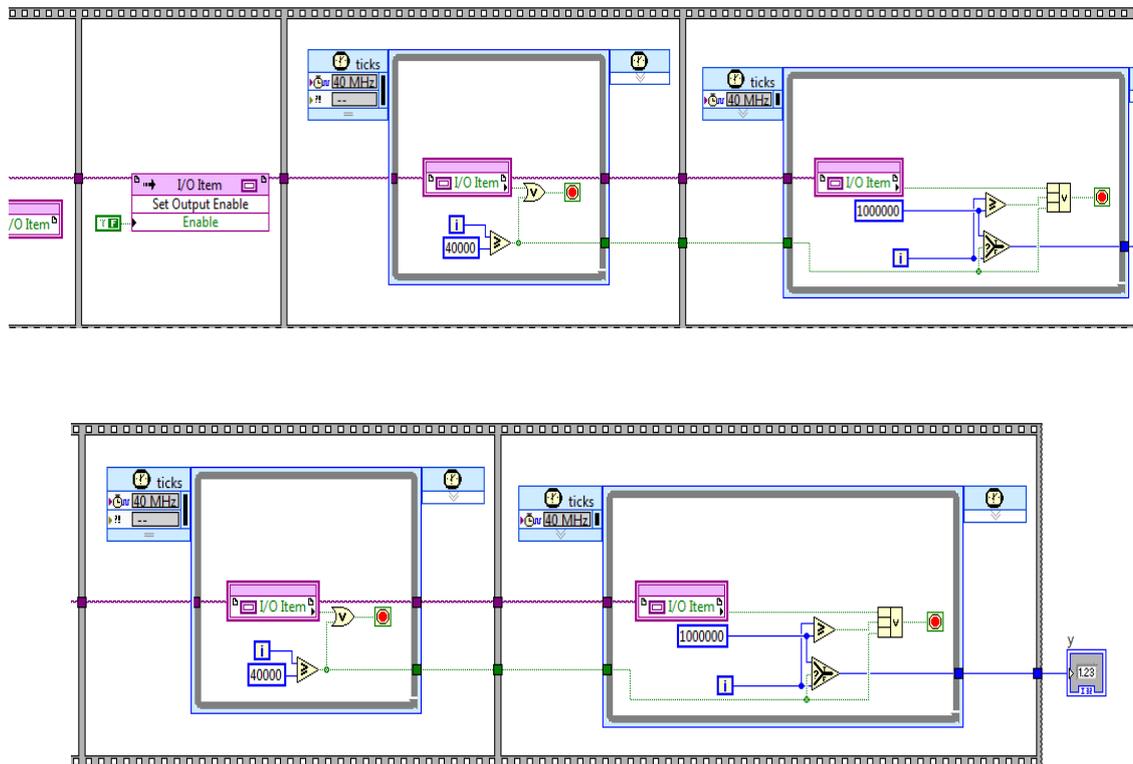


Figura 4.84 Diagrama de Bloques para la inicialización del sensor ultrasónico. En la imagen se detalla paso a paso la programación realizada al sensor para la inicialización antes de que empiece a tomar datos.

Luego se procede al diseño del VI, obteniendo los datos de los puertos y realizando un programa que permita manejar los movimientos de los motores y visualizar la distancia de los sensores de ultrasonido. La figura 4.85 y 4.86 muestra el diseño del panel frontal y el diagrama de control del VI diseñado.



Figura 4.85 Panel Frontal del VI “Captura fpga” en LabVIEW FPGA. Aquí se observa los controles para el movimiento del robot y los indicadores de distancia de los sensores de ultrasonido.

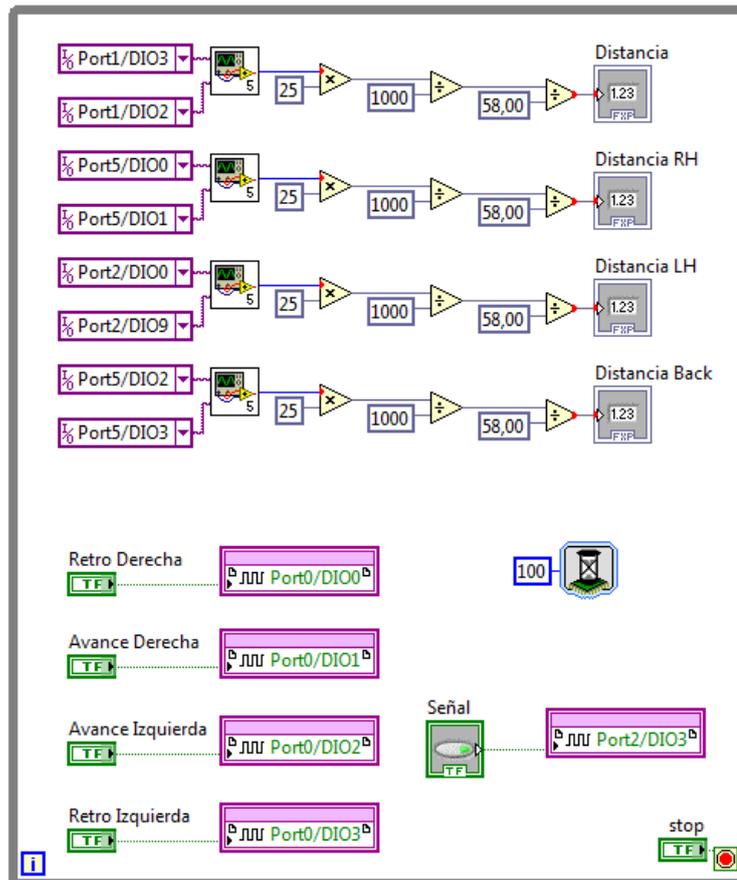


Figura 4.86 Diagrama de control del VI “Captura fpga” de LaVIEW FPGA. Aquí se observa la programación realizada para detectar los sensores y controlar los movimientos del Robot Móvil.

Adquisición de datos usando el subVI para inicializar los sensores de ultrasonido y a través de un cálculo matemático se obtiene el valor de la distancia, también se tiene los puertos que van a controlar los movimientos de los motores.

4.2 Programación desarrollada en LabVIEW PC

Para desarrollar la programación para la identificación de colores vamos a trabajar con la Cámara AXIS en LabVIEW PC, aquí la programación se desarrolla directamente en la computadora y no tiene relación con LabVIEW FPGA ni LabVIEW Real Time. La programación de la Cámara se la usa en el desarrollo del VI en LabVIEW Real Time creando una variable compartida.



Figura 4.87 Panel Frontal del VI “Captura CP” en LabVIEW PC. Aquí observamos directamente la imagen proviente de la Cámara AXIS.

En la Figura 4.87, se observa el Panel frontal desarrollado para la captura de imagen, observando la imagen proveniente de la Cámara AXIS, a través de esta imagen se identifica los colores para que sea usada en la programación de LabVIEW Real Time.

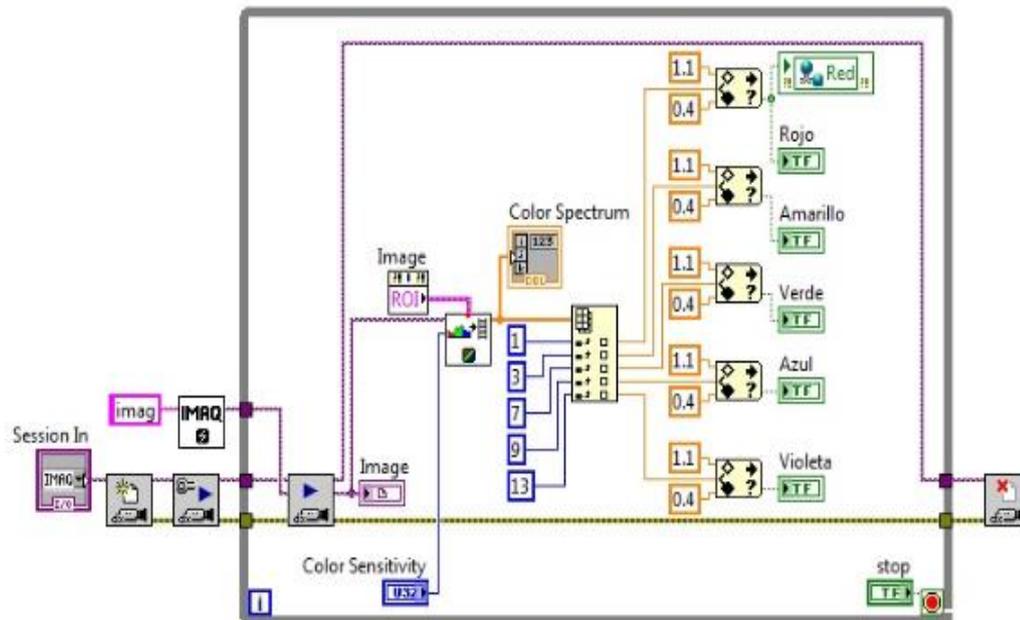


Figura 4.88 Diagrama de control del VI “Captura CP”. Aquí observamos la programación realizada para la captura de imagen a través de la Cámara AXIS.

En la Figura 4.88, muestra la programación realizada para la captura de datos, primero se accede a la cámara través de opción IMAQdx Open Camera.vi, el cual permite seleccionar la cámara que vamos a usar, para configurarla es necesario usar la opción IMAQdx Configure Grab.vi, que permitirá adquirir las imágenes, luego se usa la función IMAQdx Grab.vi, para visualizar la imagen más actual y configura su formato, estas 3 funciones trabajan en conjunto y permite visualizar la imagen en el panel frontal.

Para la identificación de colores es necesario usar la función IMAQ Color Learn, el cual extrae las características de color y la segmentación de una imagen, luego a través de la función ROI se selecciona la parte de la pantalla a ser sensada, una vez realizado esto la función Index Array permitirá dividir la imagen en diferentes valores, el cual cada valor representará un color de dicha imagen, estos valores serán tomados por la función In Range and Coerce, para darle un rango a cada color, enviando señales booleanas de identificación del color mostrado a través de un LED, esta señal será convertida en una variable compartida para que pueda ser usada en LabVIEW Real Time.

4.3 Programación desarrollada en LabVIEW Real Time

Un vez desarrollado el programa en LabVIEW FPGA, se procede a trabajar en LabVIEW Real Time tomando como referencia el VI “Captura fpga” para la adquisición de las señales y puertos de la tarjeta sbRIO-9631

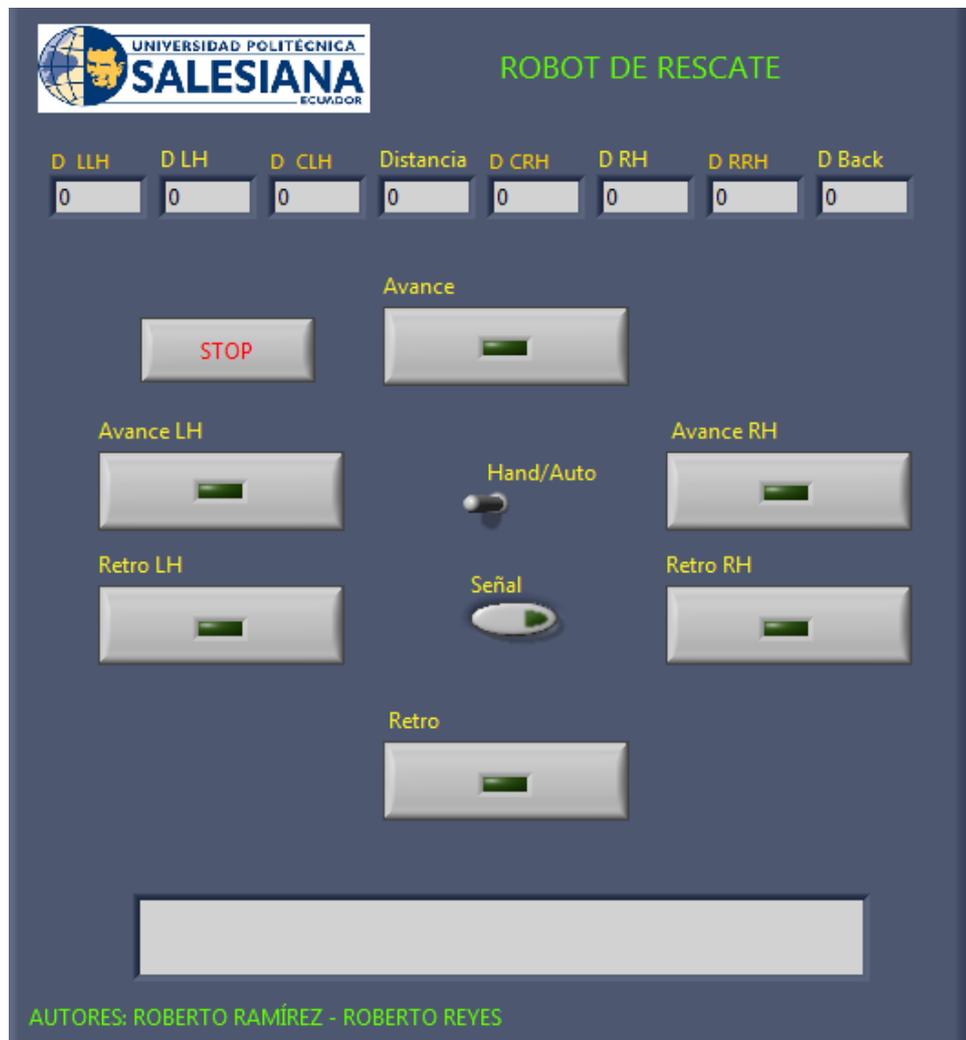


Figura 4.89 Panel frontal del VI “Captura RT” en LabVIEW Real Time. Aquí se controla el Robot Móvil y se lo manda a ejecutar en modo Manual o Automático o sea de Rescate.

En la Figura 4.89, se puede observar el Panel Frontal del diseño de la programación realizada en LabVIEW Real Time, se puede observar los controles del movimiento de los motores y el modo de control Manual que lo controla el usuario y el Automático donde se ejecuta la práctica.

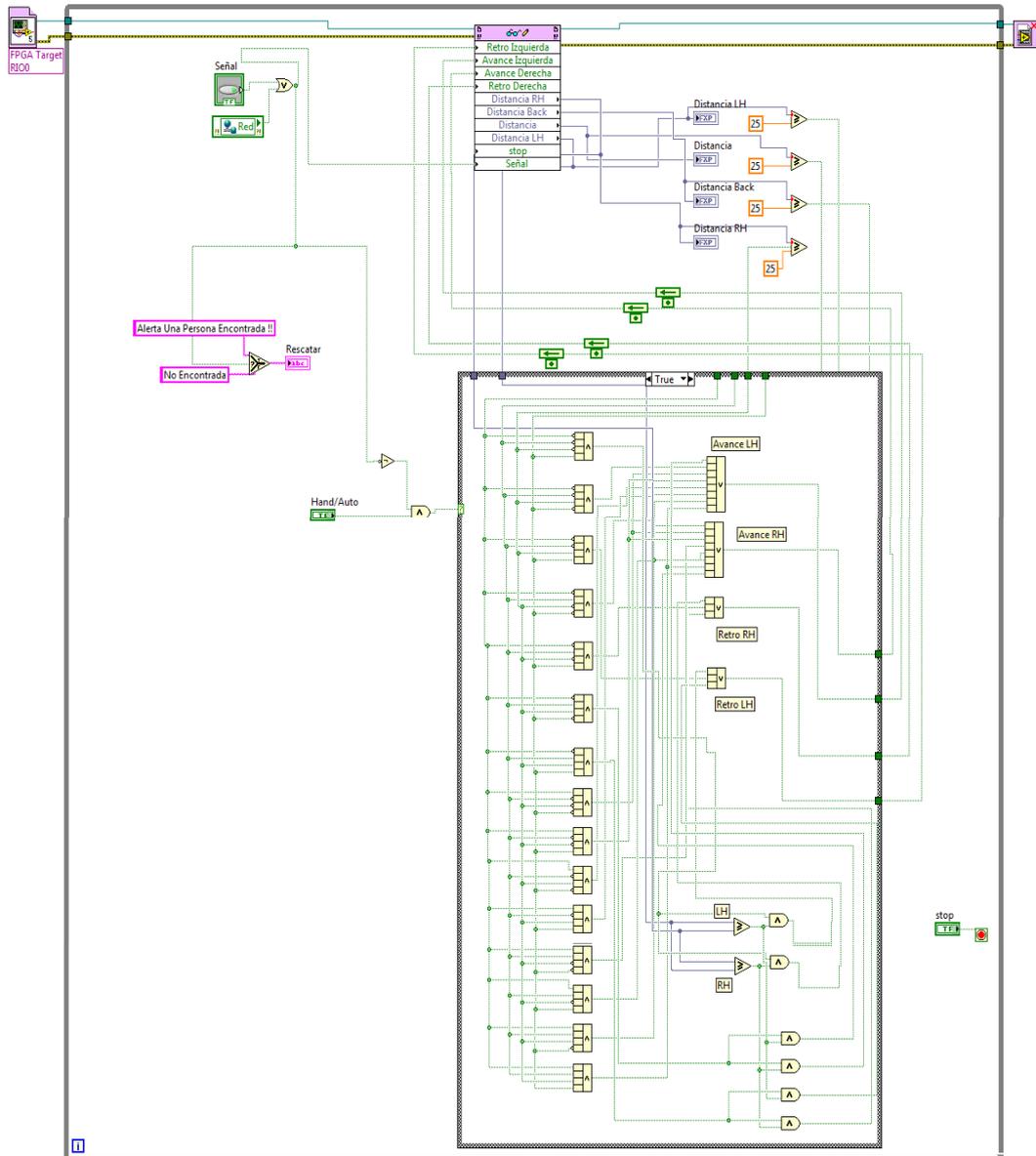


Figura 4.90 Diagrama de control del VI “Captura RT” en LabVIEW Real Time. Aquí se controla el robot de dos modos manual o automático y adicional a ello tenemos la variable compartida para la adquisición de Cámara.

A través de la programación realizada en el diagrama de control de la Figura 4.90, se va a mostrar la señal de la identificación de color a ser usado en este caso el color a identificar será el ROJO como la Víctima. Cuando el robot móvil identifica a la víctima emitirá un PITIDO indicando que la víctima fue encontrada y la visualizaremos a través del Panel frontal de LabVIEW Real Time.



Figura 4.91 Instalación de cámara de red AXIS y sensores de ultrasonido HC-SR04. Imagen frontal del robot donde se observa las ubicación de la cámara y la posición de los sensores de ultrasonido.



Figura 4.92 Visualización de la imagen a través de la Cámara AXIS. En la figura se observa como la imagen es transmitida desde la cámara hasta el Panel Frontal de LabVIEW PC.

5. CONCLUSIONES

- En esta práctica se aprendió cómo manejar una cámara IP adaptándola a la red inalámbrica del robot móvil.
- Se aprendió como identificar colores usando la Cámara IP a través de LabVIEW Real Time.
- La identificación de colores puede ser usada en múltiples aplicaciones en nuestro caso fue desarrollado para detectar a una víctima (simulando un color).

6. RECOMENDACIONES

- Al momento de instalar la cámara IP tomar las precauciones necesarias puesto que una mala instalación podría generar que la configuración de la red se borre teniendo que instalar todo nuevamente.
- Instalar los sensores de ultrasonido correctamente para evitar que puedan llegar hacer un corto dentro de la carcasa y puedan dañarse.

CONCLUSIONES

- Una de las partes importantes y que se basa este proyecto de grado es la utilización de la tarjeta FPGA NI Single-Board RIO 9631 de la National Instruments, poniendo a prueba esta nueva tecnología basada en programación controles de Tiempo Real y usando el entorno gráfico LabVIEW.
- Estructura diseñada total y absolutamente exclusivo por parte de los autores, con materiales de aluminio y acrílico para un diseño más ergonómico del robot móvil y para facilitar la instalación de los sensores y dispositivos varios.
- Este proyecto de grado se basó en aplicaciones robóticas con el uso de esta tarjeta FPGA NI sbRIO-9631, aplicaciones varias cuyos objetivos eran de lograr diferentes objetivos a través de la lectura de sensores instalados en el robot móvil.
- Como punto clave en el desarrollo de nuestro proyecto de grado fue la programación basado en la plataforma LabVIEW FPGA y LabVIEW Real Time, los cuales fueron usados para el desarrollo de cada una de las prácticas mostradas en el Capítulo 4.
- La comunicación fue otro punto importante, en nuestro caso el robot tuvo que ser controlado vía WIFI creando un red inalámbrica para la comunicación entre la PC y el robot móvil (Tarjeta NI sbRIO-9631).

RECOMENDACIONES

- El robot móvil se puede conectar con baterías recargables o usando adaptadores de voltaje, para ello tiene fusibles de protección para evitar cualquier variación de energía eléctrica que pueda afectar a nuestro robot móvil.
- El robot móvil está diseñado para trabajar con baterías recargables de tipo LIPO (Polímero de Litio), las baterías soportan una duración a carga constante de 30 minutos y varía dependiendo del tipo de práctica a desarrollarse.
- Es necesario conocer la plataforma de programación usada en este proyecto como es LabVIEW, sin los conocimientos previos resultaría imposible poder manejar el robot Móvil
- La programación en LabVIEW se desarrolla en dos estructuras: en el módulo FPGA y en módulo Real Time, para evitar largas compilaciones a través de LabVIEW FPGA, es necesario que solo la adquisición de señales sea a través de este módulo, la programación completa debe realizarse directamente en el módulo de Real Time.

CRONOGRAMA

ACTIVIDADES	Jun 14	Jul 14	Ago 14	Sep 14	Oct 14	Nov 14	Dic 14	Ene 15	Feb 15	Mar 15
Presentación de anteproyecto										
Adquisición de tarjeta FPGA y software										
Adquisición de sensores										
Diseño de estructura mecánica										
Diseño de tarjetas electrónicas										
Implementación y construcción de prototipo										
Pruebas de funcionamiento										
Desarrollo de Práctica 1										
Desarrollo de Práctica 2										
Desarrollo de Práctica 3										
Desarrollo de Práctica 4										
Desarrollo de Práctica 5										
Desarrollo de Práctica 6										
Desarrollo de Práctica 7										
Desarrollo de Práctica 8										
Elaboración de libro de Tesis										

PRESUPUESTO

COSTO DEL PROYECTO	VALOR
Tarjeta FPGA NI 9631	\$ 800,00
Tarjeta de expansión	\$ 400,00
Tarjetas electrónicas varias	\$ 350,00
Elementos electrónicos	\$ 350,00
Estructura	\$ 120,00
Partes mecánicas	\$ 500,00
Encoders	\$ 300,00
Sensores varios	\$ 450,00
Tarjetas XBEE	\$ 110,00
Cargador de baterías LIPO	\$ 150,00
Baterías LIPO	\$ 150,00
Cámara IP	\$ 180,00
Varios	\$ 391,70
Total (\$)	\$ 4.251,70

REFERENCIAS

- Aldea, M. (Febrero de 2005). *Instrumentación de tiempo real*. Obtenido de http://www.ctr.unican.es/asignaturas/ITR/01_Introduccion-2en1.pdf
- Autonics Sensor & Controllers. (2015). *Encoder Rotativo*. Obtenido de http://autonics.com.mx/products/products_detail.php?catecode=01/06/01&db_uid=1394
- AXIS Communications. (2013). *Cámara de red AXIS M1011*.
- AXIS Communications. (2013). *Tecnología de las cámaras de red*.
- Barrientos-Peñin-Balaguer-Aracil, A.-L.-C.-R. (1997). FUNDAMENTOS DE ROBÓTICA. Obtenido de www1.herrera.unt.edu.ar/faceyt/dao/files/2012/04/Robotica-1.pdf
- Calle, N. C. (2007). Estudio de robots móviles con ruedas. Repositorio Digital UPS. Obtenido de <http://dspace.ups.edu.ec/bitstream/123456789/175/2/Capitulo%201.pdf>
- Cavallero-Gutierrez, R.-F. (2010). *Introducción a las FPGA*. Obtenido de <http://www.profesores.frc.utn.edu.ar/electronica/tecnicasdigitalesi/pub/file/Publicaciones/FPGA.pdf>
- Dashboard. (20 de 01 de 2015). LM35 Temperature Sensor. Obtenido de <https://www.ntnu.no/wiki/display/plab/7.+LM35+Temperatursensor>
- DIGIBAY. (2013). HC-SR04 Ultrasonic Proximity sensor. Obtenido de <http://www.digibay.in/hc-sr04-ultrasonic-proximity-sensor>
- Divineo. (13 de 01 de 2013). Obtenido de http://www.divineo.es/product_info.php?products_id=16657&osCsid=ede13e7aaf548af0b46852e0278183e6
- DIVINEO ICS. (2015). *Batería Lipo 11.1V 3300mAh*. Obtenido de http://www.divineo.es/product_info.php?products_id=16657&osCsid=b85400235ce994485ee0e4c85046a737
- ELECTRONICA.COM.VE. (2015). Modulo Sensor de Luz por Fotoresistencia CDS. Obtenido de http://electronica.com.ve/new/catalog/product_info.php?products_id=3511
- Figuroa-Palaguachi, J.-A. (Febrero de 2012). *Diseño y elaboración de guías de laboratorio para la plataforma móvil LabVIEW Robotics sbRIO Starter Kit de National Instrumens*. Obtenido de <http://www.repositorio.espe.edu.ec/bitstream/21000/5106/1/T-ESPE-033098.pdf>
- Gonzáles-Pradines, R.-R. (Diciembre de 2007). *ANÁLISIS DE SOFTWARE PARA DESARROLLO ENTORNO GRÁFICO LABVIEW Y PROPUESTA DE IMPLEMENTACIÓN PARA LABORATORIO EN EL INSTITUTO DE*

ELECTRICIDAD Y ELECTRÓNICA EN UNIVERSIDAD AUSTRAL DE CHILE.
Obtenido de <http://cybertesis.uach.cl/tesis/uach/2007/bmfcig643a/doc/bmfcig643a.pdf>

Gravitech. (2014). *XBee 802.15.4 (Series 1) 1mW Point-to-Multipoint RF Module with Chip Antenna*. Obtenido de <http://www.gravitech.us/xb80s163porf.html>

Idetec. (2015). *BANCO QRD1114 I&T 02*. Obtenido de <http://www.ideastechnology.com/?q=node/126>

Ingeniería, C. d. (2015). *Sensor de humedad HIH-4030*. Obtenido de <http://www.cosasdeingenieria.com/esp/index/item/177/73/sensor-de-humedad-hih-4030>

Llinares, A. N. (Abril de 2010). *Sistemas Embebidos*. Obtenido de <http://server-die.alc.upv.es/asignaturas/PAEEES/2005-06/A07%20-%20Sistemas%20Embebidos.pdf>

MCI Electronics. (01 de Julio de 2010). *Guía de Usuario Xbee Series 1*. Obtenido de www.olimex.cl/pdf/Wireless/ZigBee/XBee-Guia_Usuario.pdf

MCI Electronics. (2015). *Tutorial Xbee*. Obtenido de <file:///H:/MCI%20Ltda.%20-%20Olimex%20Chile.html>

Microchip. (17 de Junio de 2003). *PIC16F87XA DATASHEET*. Obtenido de <http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf>

National Instruments. (Diciembre de 2008). *Desde la creación rápida de prototipos hasta la implantación de bajo costo con NI CompactRIO y NI Single-Board RIO*. Obtenido de http://www.redewtrumneb.com/_txt/649/54.pdf

National Instruments. (2008). *NI Single-Board RIO Embedded Control and Acquisition Devices*. Obtenido de http://www.ni.com/pdf/products/us/cat_sbRIO_96xx.pdf

National Instruments. (2008). *USER GUIDE NI sbRIO-9631*. Obtenido de <http://www.ni.com/pdf/manuals/375052c.pdf>

National Instruments. (07 de Febrero de 2011). *¿Qué es RIO de una tarjeta?* Obtenido de <http://www.ni.com/white-paper/7441/es/>

National Instruments. (Marzo de 2011). *LabVIEW FPGA Module Training* . Obtenido de <http://zone.ni.com/devzone/cda/tut/p/id/3555>

National Instruments. (07 de Febrero de 2011). *NI Single Board RIO Embedded Control and Acquisition*. Obtenido de <http://www.ni.com/white-paper/7441/es/>

National Instruments. (19 de Octubre de 2012). *Desarrollo personalizado de medición y control de E/S de Hardware con el módulo LabVIEW FPGA*. Obtenido de <http://www.ni.com/white-paper/11162/pt/>

National Instruments. (14 de Junio de 2013). Obtenido de <http://www.ni.com/white-paper/14559/en/>

National Instruments. (03 de Noviembre de 2014). *Vista general de los módulos de LabVIEW Robotics*. Obtenido de <http://www.ni.com/white-paper/11564/en/>

National Instruments. (2015). *NI FPGA*. Obtenido de <http://www.ni.com/fpga/esa/>

National Instruments. (2015). *NI sbRIO-9631 Dispositivos Embebidos con AI, AO, DIO, FPGA de IM de Compuertas*. Obtenido de http://sine.ni.com/nips/cds/pages/image?imagepath=/images/products/us/sbrio-9631_1.jpg&title=NI%20sbRIO-9631&oracleLang=esa

National Instruments. (s.f.). *USER GUIDE: NI sbRIO-961x/963x/964x and NI sbRIO-9612XT/9632XT/9642XT*. Obtenido de <http://www.ni.com/pdf/manuals/375052c.pdf>

NR Cárdenas Calle. (2007). *Capítulo 1 Estudio de robots móviles con ruedas*. Obtenido de <http://dspace.ups.edu.ec/bitstream/123456789/175/2/Capitulo%201.pdf>

Poom.W. (28 de Febrero de 2013). Poom 21 del siglo. Obtenido de <http://poomcamera.blogspot.com/>

Puente, J. A. (2007). *Introducción a los Sistemas de Tiempo Real*. Obtenido de <http://laurel.datsi.fi.upm.es/~ssoo/STR/Introduccion.pdf>

Requena, J. V. (Mayo de 2009). Obtenido de <file:///H:/MCI%20Ltda.%20-%20Olimex%20Chile.html>

Roboshop. (2015). *XBee Explorer Regulated*. Obtenido de <http://www.roboshop.com/en/sfe-xbee-explorer-regulated.html>

Sensorstecnic & Semiconductors. (2015). *Sensores ultrasónicos*. Obtenido de <http://www.sensorstecnic.net/es/productos/category/105/sensores-y-transmisores/sensores-ultrasonicos>

SINSEC. (2015). *Cámaras de red*. Obtenido de <http://www.edasistemas.com/servicios.php>

UNOCERO. (19 de 10 de 2013). Nuevos trucos para un viejo display LCD. Obtenido de <http://www.unocero.com/2013/10/19/nuevos-trucos-para-un-viejo-display-lcd/>

UNSAAC. (Noviembre de 2006). *Robótica Ing. Informáticas y Sistemas*. Obtenido de <https://robotica.wordpress.com/about/>

Wikipedia. (10 de Noviembre de 2014). *Sensor Infrarroj*. Obtenido de http://es.wikipedia.org/wiki/Sensor_infrarrojo

Wikipedia. (11 de Febrero de 2015). *Sensor Inductivo*. Obtenido de http://es.wikipedia.org/wiki/Sensor_inductivo

yahveymars. (14 de Abril de 2014). *Robots*. Obtenido de <https://yahveymars.wordpress.com/robots/>

ANEXOS

Diseño 3D Tarjeta de Adquisición de Datos diseñado en ARES Proteus

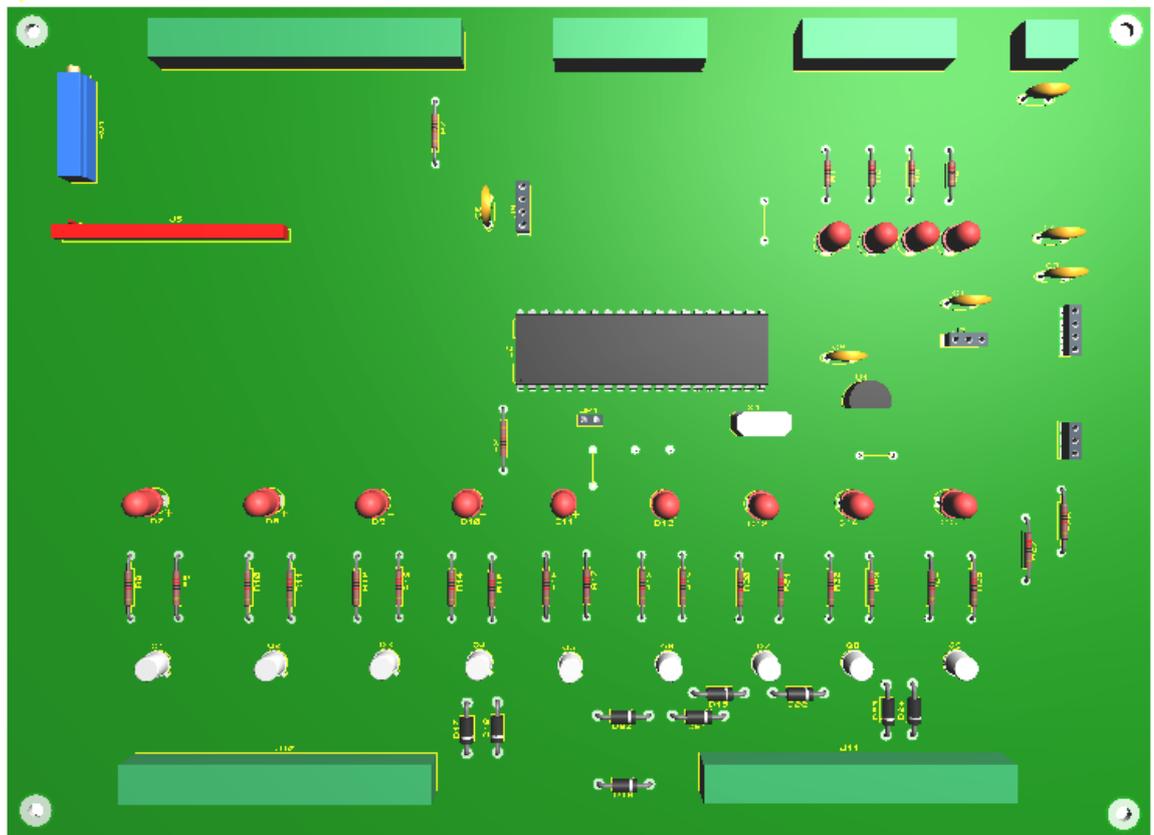
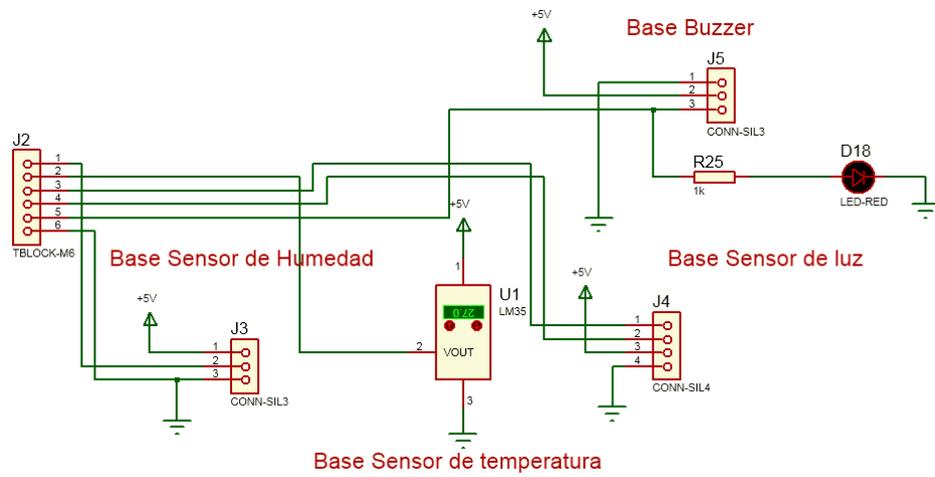
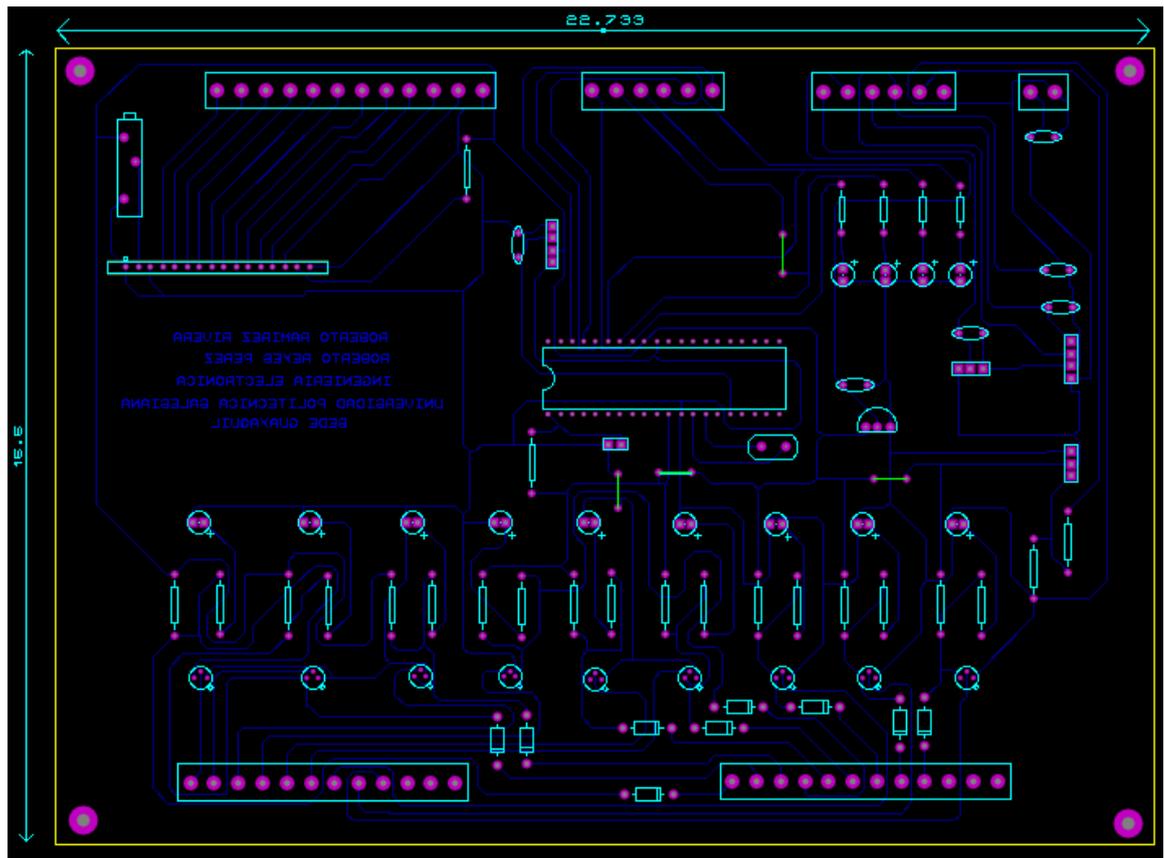


Diagrama esquemático conexión de Sensores diseñado en ISIS Proteus



Diseño de pistas de la Tarjeta de Adquisición de Datos diseñado en ARES

Proteus



Diseño 3D Tarjeta Driver para los Motores DC diseñado en ARES Proteus

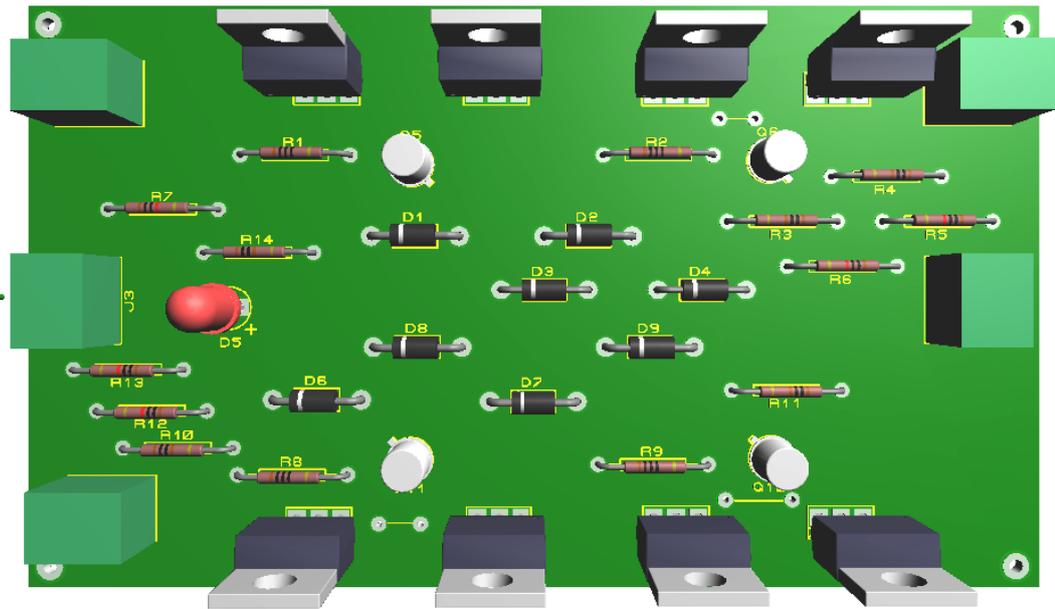
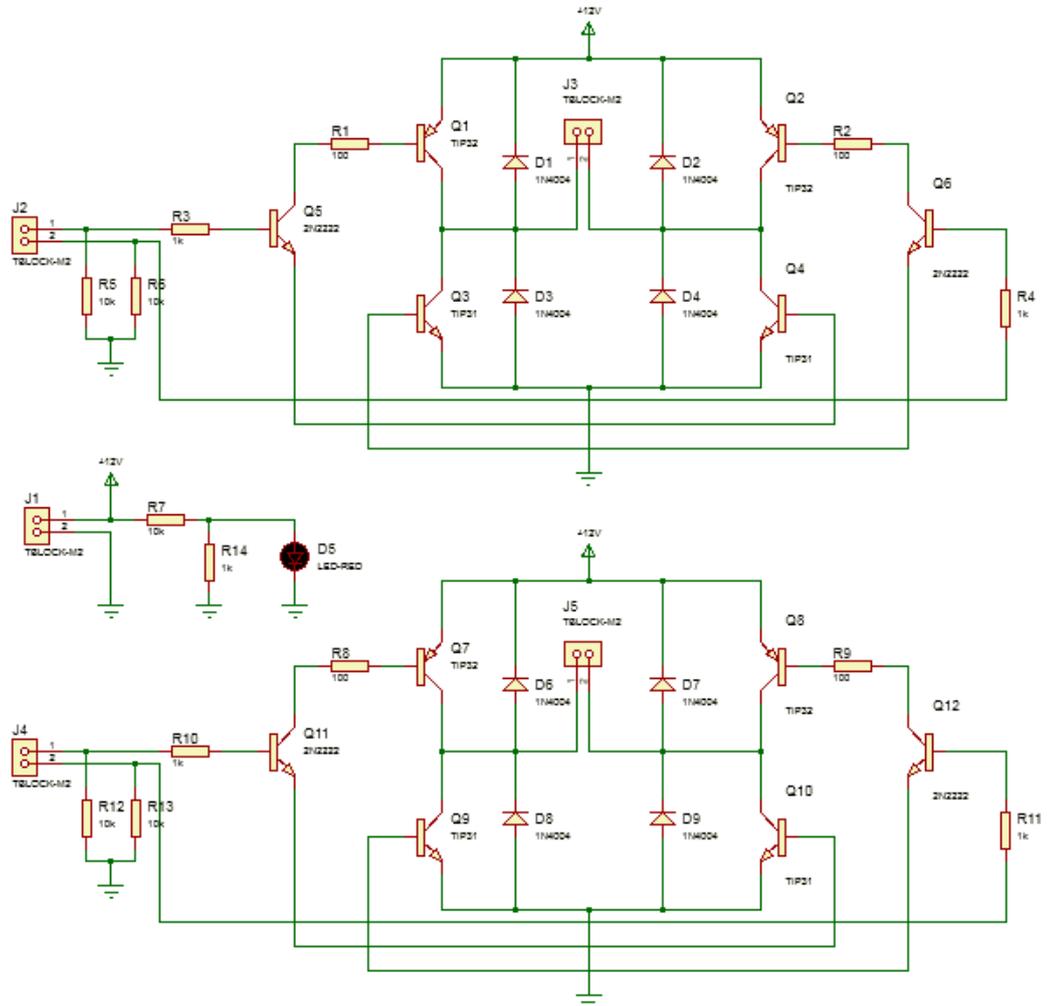
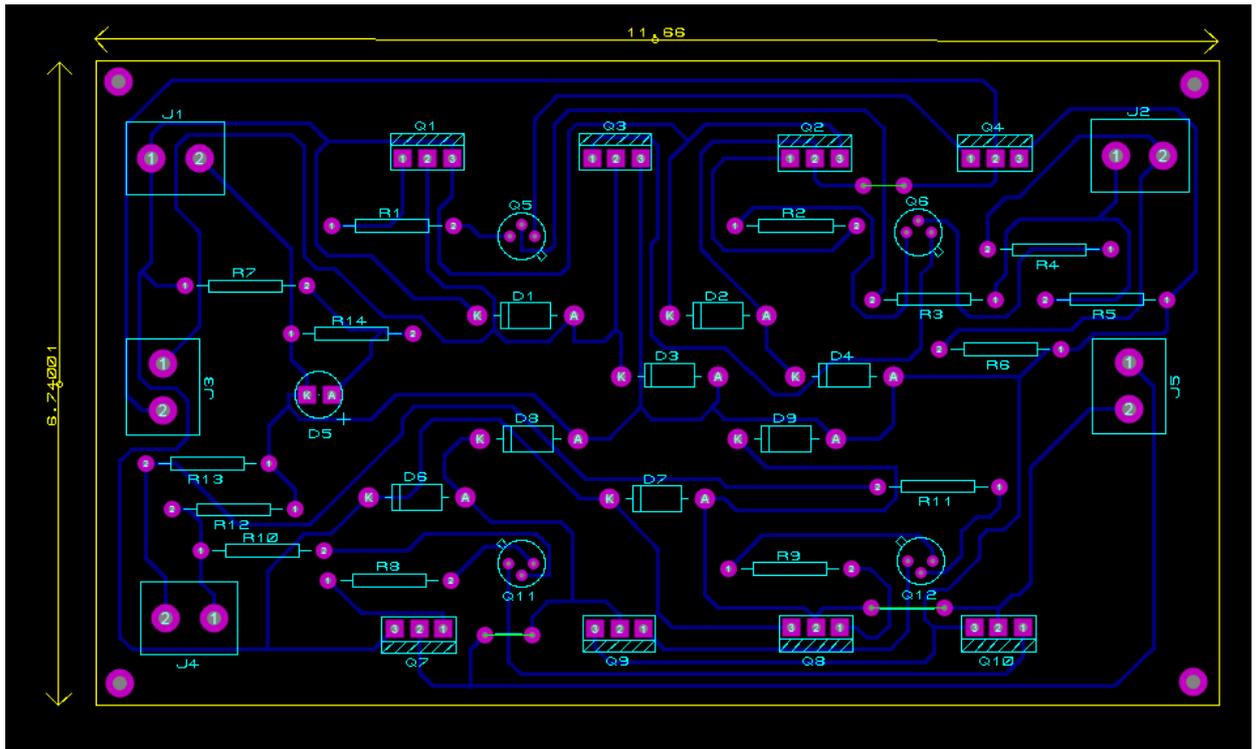


Diagrama Esquemático Tarjeta driver para los motores DC diseñado en ISIS Proteus



Diseño de pistas de la Tarjeta Driver para Motores DC diseñado en ARES

Proteus



Diseño 3D Tarjeta driver encoder diseñado en ARES Proteus

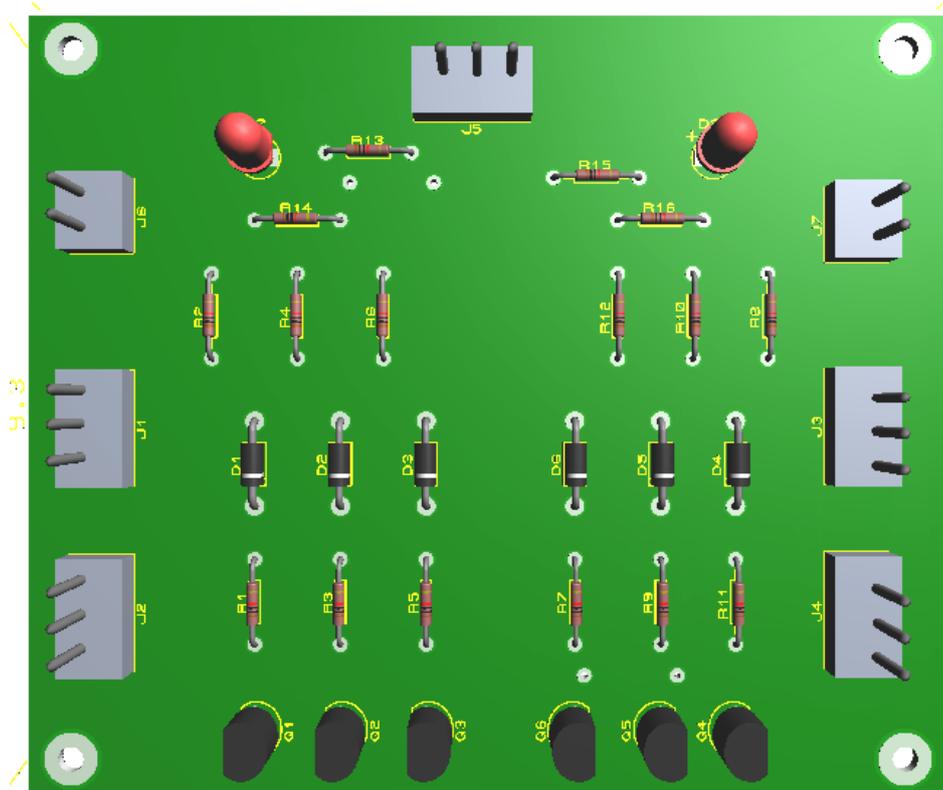
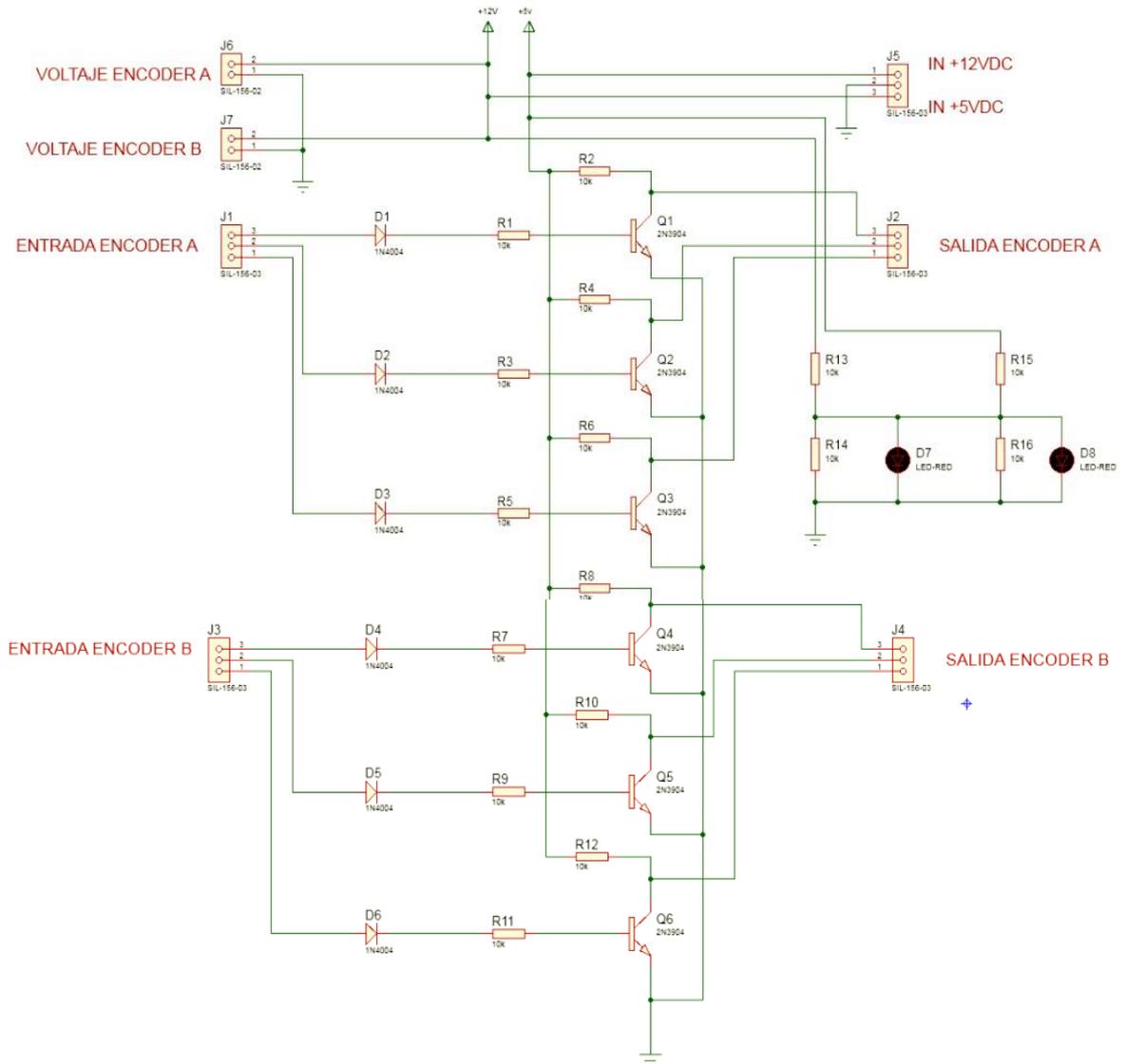
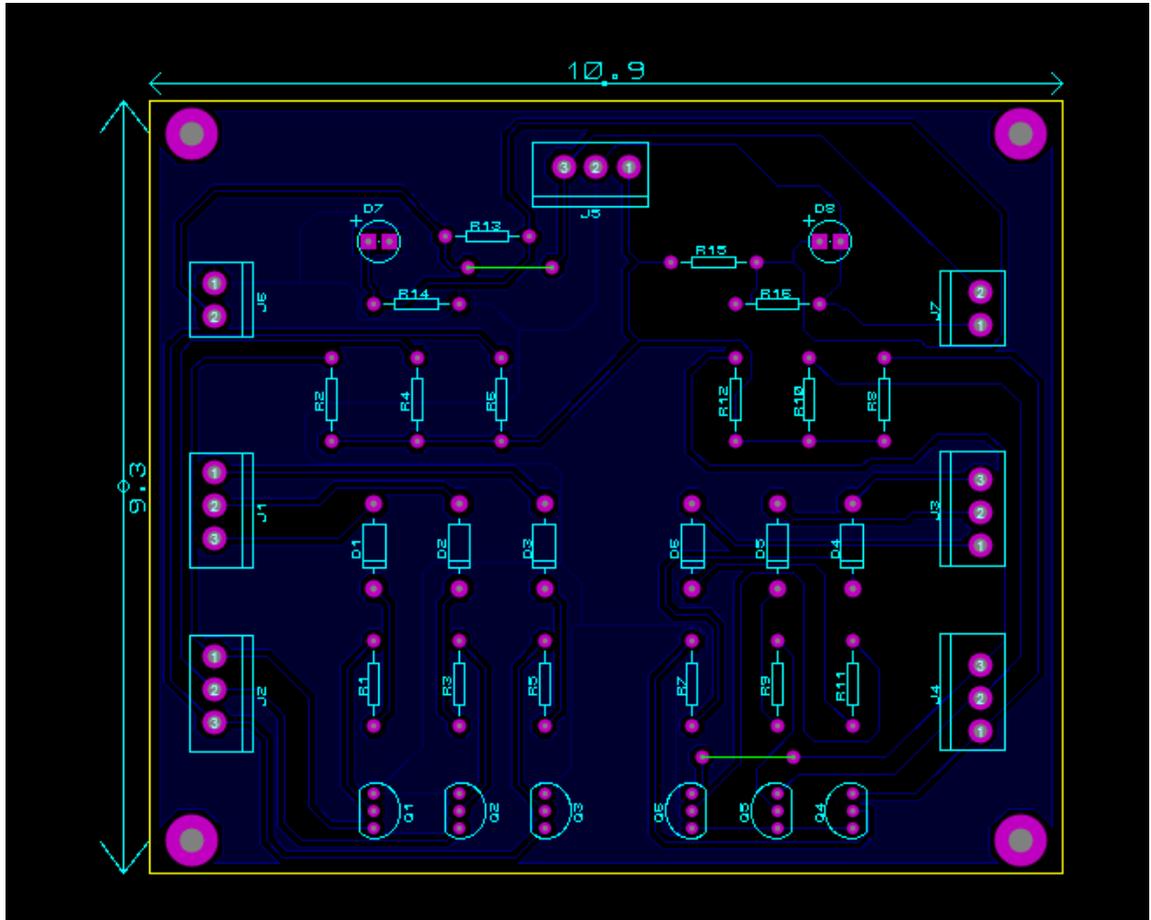


Diagrama Esquemático Tarjeta driver Encoder diseñado en ISIS Proteus



Diseño de Pistas Tarjeta driver encoder diseñado en ARES Proteus



Diseño 3D Tarjeta Módulo de Control diseñado en ARES Proteus

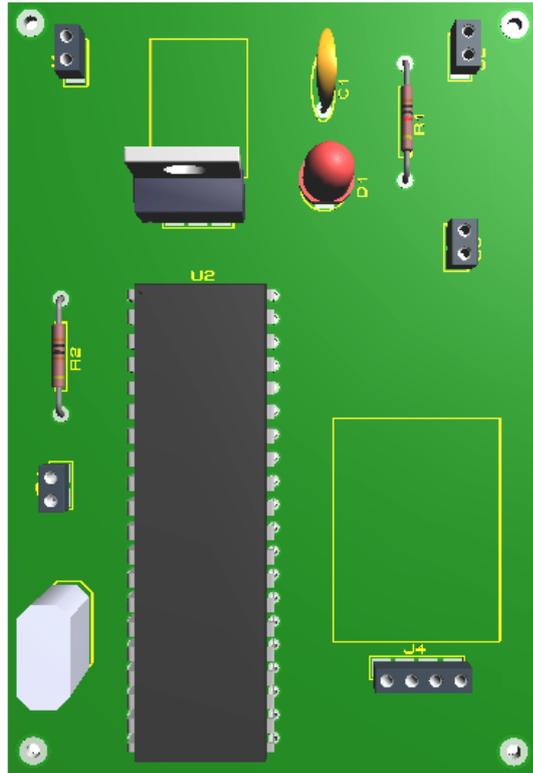
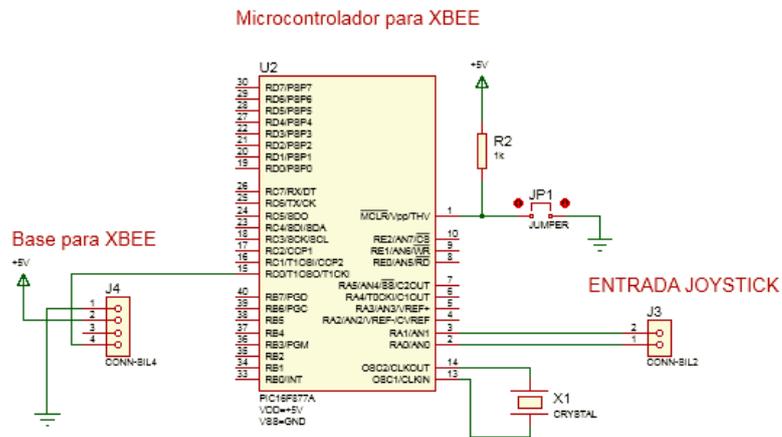
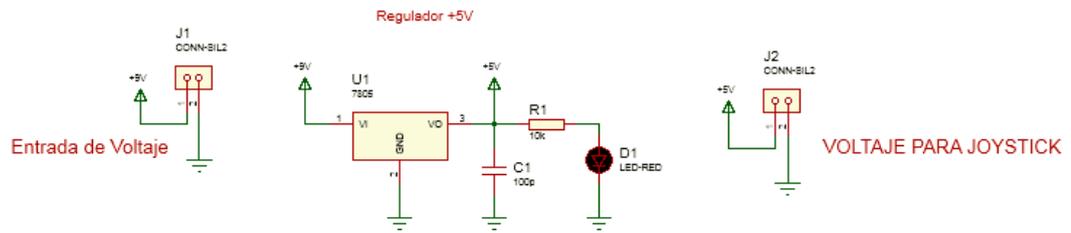


Diagrama Esquemático Tarjeta Módulo de Control diseñado en ISIS Proteus



Diseño de Pistas Tarjeta Módulo de Control diseñado en ARES Proteus

