



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA ELECTRÓNICA

**MAESTRÍA EN CIENCIAS DE LA ELECTRÓNICA
OPCIÓN EN AUTOMATIZACIÓN**

“Tarjeta para controlar 3 grados de libertad de robots vía wifi”**

T E S I S

Presentada para obtener el título de:

Maestro en Ciencias de la Electrónica Opción en Automatización

Presenta:

Lic. José De Jesús Moya Mora*

Directores:

Dr. Sergio Vergara Limon

Dr. José Fernando Reyes Cortes

Puebla, México

NOVIEMBRE 2017

* BECARIO CONACYT

** TRABAJO FINANCIADO POR LA VIEP

BUAP[®]

AGRADECIMIENTOS

Doy gracias a Dios antes que nada por permitirme contar con una trayectoria académica, el poder contar con buena salud y sobretodo darme la sabiduría necesaria para poder llevar a cabo las cosas que he querido llevar a cabo.

A mis padres, José De Jesús Moya Gómez y Leonor Mora Lara que me han sido pieza fundamental para poder llevar a cabo este posgrado, ya que sin ellos no podría estar donde estoy, su apoyo es de mucha importancia ya que me dan buenos consejos académicos y personales.

A mis asesores, por todas las grandes oportunidades, experiencias, fraternidad, ya que sin ellos no habrían sido posibles el poder llevar a cabo esta maestría, les doy un agradecimiento muy especial al Dr. Sergio Vergara Limón y a la Dra. María Aurora Vargas Treviño, ya que más que mis asesores han sido un ejemplo a seguir como humanos.

A mi novia Amanda Rocío Sánchez León por darte consejos en momentos de estrés, de esa forma poder ver soluciones alternativas que me era difícil ver por la nube de obscuridad que había alrededor de mí, de esta forma pude crecer más como ser humano y profesionista.

A mi jurado, por el gran apoyo y consejos durante mis evaluaciones.

A mis compañeros de la maestría, Rigoberto Cerino, Francisco José, Daniel Eduardo, José Raúl, Fernando Herrera, José Manuel, Hugo mayoral, Hermes Ramírez, Erick Gabino y a Giovanni Zepeda por el apoyo mutuo que tuvimos, compartiendo conocimientos y fomentando siempre la buena voluntad entre nosotros, fue una experiencia excelente.

Al Consejo Nacional de Ciencia y Tecnología por las becas que otorga a los estudiantes, con lo cual se fomenta el estudio y así puedan obtener una formación académica satisfactoriamente.

A la Vicerrectoría de Investigación y Estudios de Posgrados por financiar los proyectos de tesis de los alumnos de posgrado y de esta forma poder llevar a cabo la finalización de la investigación satisfactoriamente y de esta forma poder titularse.

RESUMEN

En el presente trabajo se explica el desarrollo de un sistema para controlar tres grados de libertad de un robot articular utilizando un sistema embebido, donde el envío de la información será por Wi-Fi, para llevar a cabo dicha comunicación se empleará un módulo de comunicación WIFI, denominado RN-XV, el cual facilitará la comunicación con una interfaz gráfica que se realizará en el software labview, para poder manipular el robot. El robot que se utiliza es un robot cartesiano (que se encuentra en el laboratorio de la maestría), se empleará un sistema embebido basado en un FPGA de la marca ALTERA de la familia Cyclone III el cual tiene todas las prestaciones para realizar dentro de él, algoritmos de control. Se desarrollarán cada una de las partes que conforman el sistema embebido, como el desarrollo del firmware que realiza el control para manipular el robot y se establecerá la comunicación vía WiFi, el desarrollo de este tipo de comunicación se realizará mediante la comunicación UART.

DEDICATORIA

Índice

RESUMEN	3
INTRODUCCIÓN	6
<i>CAPÍTULO I:</i>	16
<i>CARACTERÍSTICAS DE LAS TARJETAS DE ADQUISICIÓN DE DATOS</i>	16
1.1 TARJETA MFIO	ERROR! BOOKMARK NOT DEFINED.
1.2 TARJETA TAC	ERROR! BOOKMARK NOT DEFINED.
1.2.1 CARACTERÍSTICAS DE LAS TARJETAS MFIO Y TAC:	ERROR! BOOKMARK NOT DEFINED.
1.3 TARJETA DE ADQUISICIÓN DE DATOS Y CONTROL.....	ERROR! BOOKMARK NOT DEFINED.
1.3.1 CARACTERÍSTICAS DE LA TARJETA DE ADQUISICIÓN DE DATOS DE LA PRESENTE TESIS:	16
1.4 DIAGRAMA A BLOQUES DE SISTEMA	17
1.5 CONCLUSIÓN	19
<i>CAPITULO II</i>	20
<i>HARDWARE DE LA TARJETA DE ADQUISICIÓN</i>	20
2.1 DIAGRAMA ELÉCTRICO	20
2.2 CIRCUITOS INTEGRADOS (CI)	23
2.2.1 ACOPLAMIENTO DE LAS SEÑALES DE RETROALIMENTACIÓN	23
2.2.2 CIRCUITO INTEGRADO SELECCIONADO PARA EL ACOPLAMIENTO DIFERENCIAL	24
2.2.3 ACOPLAMIENTO DIGITAL- ANALÓGICO	27
2.2.4 DAC.....	28
2.2.5 CONVERTIDOR DE DC-DC	31
2.2.6 AMPLIFICADOR OPERACIONAL	31
2.3 ACTIVACIÓN DE LOS MANEJADORES	32
2.3.1 SEÑAL DE ACTIVACIÓN	32
2.3.2 CI SELECCIONADO PARA LA SEÑAL DE ACTIVACIÓN DE LOS MANEJADORES.....	32
2.3.3 CI SELECCIONADO PARA EL PROCESAMIENTO DE LAS SEÑALES DEL SISTEMA DE CONTROL.....	33
2.4 ESTRATEGIAS PARA EL DISEÑO	34
2.4.1 RUIDO INTERNO:	34
2.4.2 INTERFERENCIA ENTRE PISTAS	36
2.4.3 INTERFERENCIA ENTRE ETAPAS ANALÓGICA Y DIGITAL	36
2.4.5 RUIDO EXTERNO	37
2.5 DISEÑO DE LA PCB.....	38
2.5.1 DISEÑO DEL PRIMER PLANO (TOP-LAYER).....	38
2.5.2 DISEÑO DEL SEGUNDO PLANO (GND)	40
2.5.3 DISEÑO DEL TERCER PLANO (VCC):.....	41
2.5.4 DISEÑO DEL CUARTO PLANO (BOTTOM-LAYER)	42
2.5.5 TARJETA DE ADQUISICIÓN Y CONTROL.....	43
2.6 CONCLUSIÓN	44

<i>CAPITULO III</i>	45
<i>FIRMWARE DE LA TARJETA DE ADQUISICIÓN</i>	45
3.1 CONFIGURACIÓN	47
3.2 SISTEMA DE TRANSFERENCIA DE DATOS	48
3.3 DIAGRAMA PARA OBTENER LA POSICIÓN A PARTIR DEL ENCODER	54
3.4 GENERACIÓN DE PWM	58
3.5 MICROPROCESADOR	63
3.5.1 FIRMWARE PARA LA LEY DE CONTROL GENERADA PARA EL ROBOT	64
3.6 COMPILADOR	67
3.7 CONCLUSIÓN	68
<i>CAPITULO IV</i>	69
<i>APLICACIÓN</i>	69
4.1 RIEL DE ALTA PRECISIÓN.	70
4.2 TORNILLO EMBALADO	71
4.3 DESCRIPCIÓN DEL ROBOT.	73
4.4 CARACTERIZACIÓN DE LOS MOTORES	76
4.5 ELECTRÓNICA DEL ROBOT	82
4.6 MODELO DINÁMICO	85
4.6.1 MODELO DINÁMICO DEL ROBOT CARTESIANO	86
4.7 SOFTWARE DE CONTROL DE POSICIÓN EL ROBOT MANIPULADOR DE INICIO	87
4.8 CONCLUSIÓN	91
<i>CAPITULO V</i>	92
<i>RESULTADOS</i>	92
CONCLUSIONES GENERALES	95
APÉNDICE A	96
APÉNDICE B	97
APÉNDICE C	102
BIBLIOGRAFÍA	106

Índice de figuras

Figura 1: Sistema de control.....	Error! Bookmark not defined.
Figura 2: Tarjeta MFIO.....	Error! Bookmark not defined.
Figura 3: Tarjeta TAC.	Error! Bookmark not defined.
Figura 4: Diagrama a bloques general del sistema de control.	17
Figura 5: Diagrama esquemático de la tarjeta de adquisición de datos.....	21
Figura 6: Fases A y B de la señal de cuadratura con ruido.	23
Figura 7:Fases A y B de la señal de cuadratura en forma diferencial.....	24
Figura 8: Diagrama esquemático del LVDS.	24
Figura 9: Suma de la señal diferencial.....	26
Figura 10: Diagrama de tiempos del LVDT.	26
Figura 11: Fases A y B de las señales de cuadratura del encoder.....	27
Figura 12: Diagrama esquemático del DAC.....	28
Figura 13: Diagrama a bloques del DAC utilizado.....	28
Figura 14: Esquemático del DC-DC.	31
Figura 15: Esquemático del amplificador operacional.....	31
Figura 16: Diagrama esquemático de los MAX999.....	32
Figura 17: Diagrama esquemático de la FPGA.....	33
Figura 18: Voltaje de rizo de una fuente de alimentación.	34
Figura 19: Fuente de voltaje con filtro EMI.	34
Figura 20: Circuito equivalente y especificaciones del filtro EMI.....	35
Figura 21: Interferencia entre pistas.	36
Figura 22: Disminución de interferencia por vías.	37
Figura 23: Capas de la tarjeta.....	38
Figura 24: Capa Superior(Top-Layer).....	39
Figura 25: Capa de GND.	40
Figura 26: Capa de VCC.	41
Figura 27: Capa de Bottom-Layer.....	42
Figura 28: Diseño final de la tarjeta.	43
Figura 29: Diagrama general del Firmware.	46
Figura 30: firmware del Protocolo de comunicación WiFi.....	49
Figura 31: Trasmisión UART.....	49
Figura 32: Comunicación UART.....	50
Figura 33: Muestreo de bits.	50
Figura 34: Estructura a bloques del Firmware del UART.....	51
Figura 35: Máquinas de estados de la transmisión y recepción del UART.	52
Figura 36: Módulo UART TX.	52
Figura 37: Módulo UART TX.....	52
Figura 38: Módulo tesUART.....	53
Figura 39: Módulo StartStop.	53

Figura 40: Tarjeta de adquisición de datos.	54
Figura 41: Diagrama a bloques del proceso completo del conteo de pulsos.	54
Figura 42: Combinaciones lógicas del encoder.	55
Figura 43: Circuito para la generación de la señal de reloj.	55
Figura 44: Diagrama de tiempos de las señales A y B.	56
Figura 45: Señal del reloj generada.	56
Figura 46: Diagrama a bloques de la generación de pulsos para máquina de estados del encoder.	57
Figura 47: Diagrama de flujo del encoder.	57
Figura 48: Diagrama de flujo del contador.	60
Figura 49: Diagrama de flujo de divisor de frecuencia a 100Hz.	61
Figura 50: Bloque de PWM.	62
Figura 51: Diagrama de flujo del PWM.	62
Figura 52: Bloque del Microprocesador.	63
Figura 53: Compilador.	67
Figura 54: Robot CNC.	69
Figura 55: Eje X.	70
Figura 56: Eje Y.	71
Figura 57: Eje Z.	72
Figura 58: Motor Z5D120-12.	74
Figura 59: Reductor 5GU9K.	75
Figura 60: Motor Z4D40-12.	75
Figura 61: Grafica de resultado del motor X.	77
Figura 62: Gráfica de resultados del motor Y.	78
Figura 63: Gráfica de resultados del motor Z.	79
Figura 64: Encoder giratorio GTS06-VE-RA 1000.	80
Figura 65: Ejemplo de encoder con salida NPN tipo colector abierto.	80
Figura 66: Diagrama esquemático del Puente H.	83
Figura 67: Diagrama esquemático para las señales de los encoders.	84
Figura 68: Representación del robot cartesiano de 3 grados de libertad.	85
Figura 69: Pantalla principal del software.	88
Figura 70: Diagrama a bloques del software.	89
Figura 71: Gráfica de posición a 90° del motor 1.	93
Figura 72: Grafica de error de posición.	93
Figura 73: Grafica de par del motor.	94
Figure 74: Grafica de posición del motor 2.	94

INTRODUCCIÓN

Desde la aparición de los primeros instrumentos electrónicos de medida, los cuales eran controlados mediante sus paneles frontales, éstos han adquirido gran flexibilidad y un alto grado de integración para interconectarse con otros instrumentos en sistemas más complejos.

A los instrumentos que pertenecen a la primera generación se les conoce como instrumentos de medida tradicionales (Figura 2), estos instrumentos limitaban la manipulación y procesamiento de medidas debido a que funcionaban manualmente a través de un panel frontal. Las etapas principales de un instrumento de medida tradicional son:

- *Adquisición de datos*: para acondicionar, medir y registrar señales de las líneas de entrada del instrumento.
- *Análisis de datos*: realiza el análisis y procesamiento a la señal obtenida.
- *Presentación*: despliegan la señal de salida.

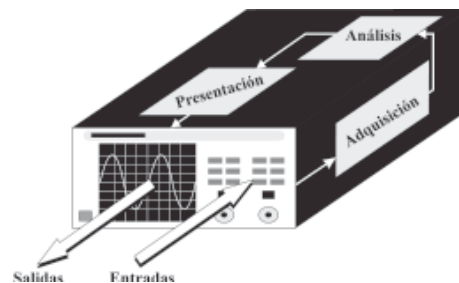


Figura1.- etapas básicas de un instrumento de medida tradicional

Generalmente, cada instrumento se diseña para realizar una medición en específico, por lo que los usuarios pueden incrementar el número de instrumentos para crear un sistema completo de medición, sin embargo, el espacio de trabajo requerido aumenta considerablemente. Con el origen de un bus de interfaz de propósito general, los usuarios son capaces de controlar sistemas de instrumentación electrónica de forma remota mediante una secuencia de órdenes, dando origen al término de instrumentación electrónica programable.

Los instrumentos electrónicos programables son una combinación de computadoras de propósito general con instrumentos tradicionales y de nueva generación; estos modernos instrumentos pueden alcanzar nuevos niveles de ejecución y flexibilidad al combinar rutinas de software con nuevas clases de hardware de instrumentación; además, con la implementación de nuevas técnicas de control de instrumentos, como son la programación basada en registros y memoria compartida, se puede optimizar el uso de la computadora para un procesamiento avanzado al adquirir, analizar y presentar resultados de las mediciones requeridas en forma virtual, dando origen al término instrumentación virtual (VI, Virtual Instrumentation).

Con los instrumentos virtuales se puede tener una gran variedad de dispositivos de instrumentación, respaldados por rutinas de software que permiten la creación de interfaces gráficas de usuario (GUI, Graphic User Interface) de alta resolución y flexibilidad.

Las siguientes son definiciones de la instrumentación electrónica programable y virtual respectivamente:

- Instrumentación electrónica programable: se define como una GUI en una PC controlando a un instrumento tradicional mediante un medio de comunicación GPIB o RS-232 (Figura 3.B); una GUI en una PC controlando a una tarjeta de adquisición de datos (DAQ) de tipo plug-in (Figura 3.A); o a un módulo VXI mediante la comunicación GPIB (Figura 3.C).
- Instrumentación virtual: se define como una GUI sin instrumentos físicos conectados a una computadora personal (Figura 3.D).

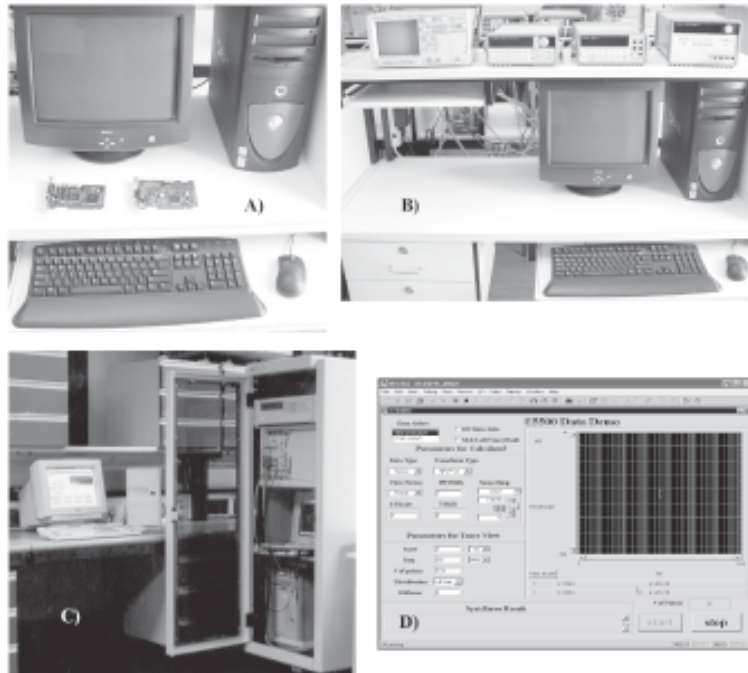


Figura. - 2. diferentes configuraciones de instrumentación electrónica programable y virtual.

En algunas aplicaciones es necesario utilizar las medidas de los instrumentos para la toma de decisiones en cuanto al control de procesos de fabricación, de gestión, etc., lo cual ha dado origen a los sistemas ATE.

Para comprender la instrumentación electrónica, es importante conocer cómo han evolucionado los instrumentos a lo largo de los años. La Tabla 1 presenta algunos hitos y eventos importantes en la historia de la instrumentación electrónica.

Año	Evento histórico
1960	Bucle de corriente de 4-20mA para sensores y actuadores, basado en la red télex de 1920 (teletipo o TTY). Norma ANSI MC12.1 e ISA S50.1. Posteriormente fue incorporado al protocolo HART para sensores inteligentes.
1965	Hewlett-Packard desarrolla un bus paralelo para conectar instrumentos de medida a una computadora, denominado HP-IB (Hewlett-Packard Instrument Bus). La computadora actúa como controlador y tiene la capacidad de emitir órdenes a los instrumentos y leer de los mismos las medidas realizadas.
1969	Conexión serie IEA RS232-C (también CCITT V.24 y V.28), que se convirtió en el "puerto serie" de propósito general para comunicarse a baja velocidad (9600 Kbps) en distancias cortas (10m). Evolución a la norma IEA RS-485 que en los años 80 se implementa como la capa física de los buses de campo (Fieldbus) de dos hilos, dando un gran impulso a las comunicaciones industriales.
1975	Primer sistema de tarjetas de bus común de propósito general S-100 de la firma MITS para el μ C 8080 de Intel (Norma IEEE 696). El IEC y el IEEE adoptan el bus de instrumentación programable HP-IB bajo las normas IEC 625 e IEEE 488 respectivamente. Este bus pasa a denominarse GPIB.
1978	La ISO presenta un modelo de interconexión de sistemas abiertos (OSI) para el desarrollo de protocolos normalizados en comunicaciones (LAN, MAN y WAN).
1981	Bus común (mother board) para PC's denominado ISA propuesto por IBM (norma IEEE 996). Desarrollo generalizado de tarjetas conectables al PC para aumentar sus funciones y convertirlo así en una plataforma multipropósito.
1983	Bus FASTBUS de los comités NIM y ESONE para experimentación en física de altas energías (Normas IEEE 960 e IEC 935).
1984	General Motors impulsa la norma MAP (Manufacturing Automation Protocol) para el desarrollo de un protocolo de mensajes de fabricación en tiempo real en el nivel de aplicación OSI. Aparición de numerosos buses de campo.
1985	Ante la falta de compatibilidad entre controladores e instrumentos GPIB de diferentes fabricantes, Tektronix propone un conjunto de formatos estándares para órdenes y respuestas de instrumentos programables GPIB.
1987	La propuesta de Tektronix es adoptada por el IEEE bajo la norma IEEE 488.2 de protocolos de intercambio de mensajes, formatos de datos, sintaxis, informes de estado y órdenes comunes a distintos tipos de instrumentos programables. La norma original IEEE 488 pasa a denominarse IEEE 488.1. Bus común VXI (VME eXtensions for Instrumentation) basado en VME (Versa Module Europe) para instrumentación programable modular con instrumentos en forma de tarjetas. Creación del consorcio VXI (VXI consortium).
1989	National Instruments desarrolla el bus de cable MXI (Multisystem Instrument Interface) proporcionando así un mecanismo de altas presentaciones para interconectar sistemas de instrumentación programable VXI y controlarlos desde una PC. Se crea la asociación PCMCIA (Personal Computer Memory Card International Association) como un estándar para el intercambio de datos en las computadoras portátiles.
1990	Propuesta de la norma SCPI (Standard Commands for Programmable Instruments) que especifica un extenso conjunto de órdenes en formato ASCII (American Standard Code for Information Interchange) para manejar la gran variedad de funciones que realizan los instrumentos programables. Junto con IEEE 488.2 permite la compatibilidad entre instrumentos modulares en tarjetas VXI. Creación del consorcio SPXI (SCPI Consortium) y desarrollo acelerado de herramientas de programación compatibles en instrumentación programable.
1991	Se define la interfaz de conexión PCMCIA de entradas y salidas en un conector de 68 terminales que se utiliza en tarjetas de memoria.
1992	Bus de interconexión SCI para enlazar sistemas de bus común en aplicaciones distribuidas de alta velocidad (IEEE 4596). Proyecto ISP (Interoperable System Project) para especificar una norma de bus de campo global que dio lugar al estándar ISA/IEC SP50.
1993	El IEEE adopta el bus VXI como norma IEEE 1151. Fundación de la alianza VXI&plug&play para buscar un mayor nivel de estandarización con respecto a todos los componentes de un sistema de instrumentación programable VXI.
1994	Se da a conocer la versión 0.7 del USB (Universal Serial Bus).
1995	Se da a conocer la especificación CompactPCI para la conexión de instrumentos mediante el bus PCI (Peripheral Component Interconnection), actualmente estandarizado bajo las normas ANSI 310-C, IEC-297, IEEE 1101.1, IEEE 1101.10 y P 1101.11.
1997	Bus común PXI (PCI eXtensions for Instrumentation) basado en PCI propuesto por National Instruments para instrumentación programable en tarjetas. Norma IEEE 802.11 para redes inalámbricas de corto alcance (WLAN). National Instruments propone la especificación HS 488 para aumentar la velocidad del bus de instrumentación programable GPIB (IEEE 488.1 y 488.2) a 8 MBps.
1998	Se adoptan IVI (Interchangeable Virtual Instruments) y VISA (Virtual Instrumentation Software Application) para la implementación de sistemas empotrados (firmware) que necesiten interactuar en el campo de la instrumentación basada en VXI, GPIB, PXI y comunicación serie. Se da a conocer la revisión 1.1 al estándar USB que corrige errores del estándar original.
2000	La división de instrumentación de Hewlett-Packard para a ser Agilent Technologi es.
2001	Se define el estándar USB 2.0 de alta velocidad.

TABLA 1. Evolución de los buses de instrumentación electrónica.

Actualmente en el mercado existen diferentes tipos de tarjetas de control que se enfocan en la adquisición de datos para poder tener el control de sistemas, algunas de estas tarjetas son las que se mencionan a continuación:

Tarjeta ME-Jekyll (ME-4610) :



ME-4610
Características:
Salidas analógicas:
1 salidas de 16 bits.
Salida de ± 10 volts.
Memoria:
128bytes.
Tipo de bus
PCI
PRECIO: 14,568.00 pesos.

Tarjeta RedLab:



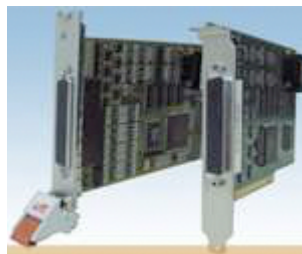
ME-4610
Características:
Salidas analógicas:
2 salidas de 16 bits.
Salida de ± 5 volts.
Memoria:
32 bytes.
Tipo de bus
USB
PRECIO: 9234.34 pesos.

Tarjeta Me-630



ME-4610
Características:
Salidas analógicas:
3 salidas de 16 bits.
Síncronas.
Salida de +5 volts.
Memoria:
128 bytes.
Tipo de bus
PCI-USB
PRECIO: 18423.84 pesos.

Tarjeta M3-1600



ME-4610
Características:
Salidas analógicas:
2 salidas de 12bits.
Síncronas.
Salida de +5 volts.
Memoria:
52 bytes.
Tipo de bus
PCI
PRECIO: 1823.84 pesos.

Las tarjetas de adquisición de datos consisten en tres partes principales: por un lado, está el circuito de acondicionamiento de señales, después está el convertidor analógico-digital (ADC por sus siglas en inglés) y, finalmente, el bus del ordenador.

Los sistemas de adquisición de datos se utilizan cada vez más en laboratorios, industria y educación. Puede ser sólo para la lectura de las variables físicas (presión, nivel, temperatura y flujo, por ejemplo), para visualizar el estado de un proceso o bien, para el procesamiento de estas variables con la finalidad de controlar un proceso.

Con un sistema de adquisición de datos se puede obtener el valor de las variables físicas. También es posible llevar un registro de los valores de las variables del proceso. Muchos procesos industriales se monitorean y/o controlan con tarjetas de adquisición de datos y una computadora, además de graficadores y de registradores.

Actualmente se emplea un Manejador o Driver en las actuales tarjetas del laboratorio de la maestría en ciencias de la electrónica:

El manejador recibe la señal de cuadratura de los encoders y las envía a la etapa de control del sistema, la señal de cuadratura que entrega el manejador es de forma diferencial, figura.2.6, se transmite de esta forma para disminuir las radio frecuencias (RF) o Interferencia electromagnética (o por sus siglas en inglés EMI) que se inducen en la señal.

Cada manejador puede ser configurado para trabajar en diferentes modalidades:

- **Par:** en la modalidad de par, el manejador regula el par que genera el servomotor en razón de una señal analógica proveniente de la etapa de control. El rango de voltaje de esta señal de control es de $\pm 10V$ que corresponden al rango de par máximo $+10V$ y par mínimo $-10V$.
- **Velocidad:** en la modalidad de velocidad el ángulo de rotación es controlado y corresponde a una señal o comando de velocidad de $\pm 10Volts$.
- **Posición:** en la modalidad de posición, la exactitud del posicionamiento no es buena debido a que se utilizan una retroalimentación de posición de arquitectura cerrada. El comando de posición se le envía desde la PC.

Como protección del sistema una señal de control activa los manejadores 2ms antes que el sistema funcione, además recibe una señal de 5V, que a su vez el manejador suministra a los encoders que se encuentran en los servomotores para su funcionamiento.

A continuación, se muestra en la figura el conjunto servomotor-manejador:

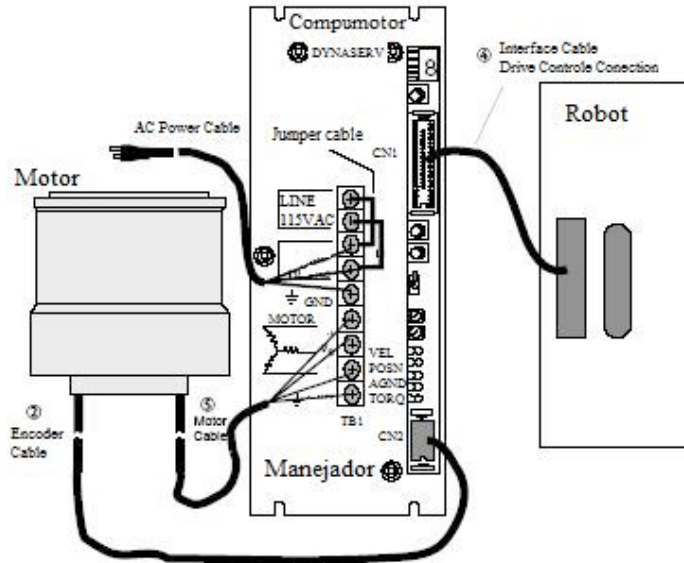
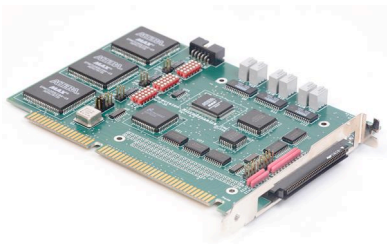


Figura: Configuración servomotor-Manejador.

Las tarjetas utilizadas actualmente por el sistema de control del robot, es la tarjeta MFIO y la TAC, la primera tarjeta que consta de tres FPGA, un acoplamiento diferencial y un acoplamiento digital analógico, la comunicación de esta tarjeta con la PC se realiza mediante el bus ISA, bus de comunicación que posee las PC pentium II o anteriores, en la figura 1 se muestra una vista de la tarjeta utilizada en el sistema de control del robot.

Tarjeta MFIO-3A



Tarjeta TAC1:



MFIO-3A	TAC 1
Características	Características:
Salidas analógicas: 3 salidas de 16 bits. Salida de +10 volts.	Salidas analógicas: 3 salidas de 16 bits. Salida de +10 volts.
Memoria: 512 bytes.	Memoria: 165,888 bytes.
Tipo de bus: ISA.	Tipo de bus PCI

El desarrollo de tarjetas de adquisición de datos por empresas privadas, hacen que el uso de estas, sea vean limitadas al número de entradas y salidas, al tipo de conectores que emplean en su fabricación, así como las limitantes en el diseño que ofrecen para el su uso, haciendo que el usuario se apegue a sus especificaciones, de esta forma se ve obligado el usuario a adquirir diferentes herramientas para poder hacer uso de una tarjeta de adquisición de datos.

La falta de desarrollo de tecnología en un país, es un factor importante por el cual no pueda competir con empresas extranjeras, es por ello que es un punto clave el poder desarrollar nuestra propia tecnología y sea usada en procesos donde se requiera automatizar un sistema, dejando las ganancias en el mismo país y estas no se vean reflejadas en el extranjero como se ha ido tratando a través de los años.

Bajo los antecedentes antes mencionados, la tarjeta que se presenta en este trabajo de tesis, se busca superar esos problemas aportando una tarjeta de arquitectura abierta y con una comunicación wifi con la PC. Dentro de esta tarjeta se desarrollará un firmware que realice el control de los tres grados de libertad del robot.

Realizando esta actualización a la tarjeta de adquisición de datos, se tendrá tecnología de punta para este periodo, alcanzando la competitividad con empresas extranjeras, ofreciendo tecnología cien por ciento mexicana, siendo este un factor clave para impulsar el desarrollo de tecnología en nuestro país

Se busca realizar una tarjeta universal que facilite la conexión de diferentes de robots de 3 grados que desee manipular el usuario, esta tarjeta cuenta con los requerimientos necesarios para poder realizar la manipulación de estos, la cual requerimos los siguientes parámetros, salidas analógicas de 16 bits de resolución, 10 nanos segundos de muestreo de las señales, comunicación wifi.

El presente trabajo tiene sus fundamentos en diversos desarrollos tecnológicos, ya que involucra FPGA's de la compañía Altera, avances en las telecomunicaciones, ingeniería de control, sistemas embebidos y robótica, puesto que el presente trabajo tiene como objetivo diseñar un sistema embebido el cuál mediante el uso de un FPGA y un módulo Wifi, pueda configurar y monitorear el algoritmo de control del robot CARTESIANO. Será necesario hacer

modificaciones de firmware, software y hardware para crear un sistema embebido configurable vía Wifi.

En la actualidad los sistemas de control se realizan con ayuda de computadoras personales (o por sus siglas en inglés PC) utilizando software de control comerciales o diseñados bajo ambientes MSDOS, QNX o WINDOWS compatibles con tarjetas de adquisición de datos comerciales de alto costo. Dependiendo del fabricante, estas tarjetas de control tienen características previamente definidas, restringiendo el uso a estas mismas, por ejemplo, al tipo de conector, voltajes de salida o entrada, etc. Por lo tanto, se realizan algunas consideraciones como esclavizar la PC a un sistema, utilizar interfaces no atractivas al usuario o simplemente realizar el control utilizando las aplicaciones ya programadas en el dispositivo a utilizar.

Esta tarjeta cuenta con un microprocesador interno dentro de la FPGA para poder realizar el sistema de control, teniendo haciendo una respuesta en tiempo real de nuestro sistema a diferencia de los sistemas convencionales, tiene la facilidad de poder controlar motores de corriente directa, alterna, trifásicos, a pasos, con sus respectivos drives para la manipulación de los mismos, se cuenta también con la etapa analógica-digital para controlar robots tipo rotradi o cualquier robot que se desee por medio de modulación de ancho de pulso (PWM).

Objetivo General

“Diseño de una tarjeta de adquisición de datos para un robot de 3 grados de libertad vía wifi”.

Objetivos Particulares

- Estudiar la cinemática y dinámica del robot articular de 3 grados de libertad.
 - Estudiar el protocolo de comunicación wifi.
 - Realizar el diseño de la tarjeta de adquisición de datos.
 - Utilizar un controlador para probar la tarjeta.
 - Desarrollar el firmware necesario para el control del robot.
 - Diseñar un ambiente de programación amigable al usuario.
 - Realizar pruebas del sistema completo.
 - Publicación de los resultados.
 - Escritura de la tesis.
- **Capítulo I:** En este capítulo se aborda las características de la tarjeta de adquisición de datos, así como un diagrama a bloques explicando el funcionamiento del sistema.
 - **Capítulo II:** En este capítulo se aborda la problemática del diseño de una tarjeta de adquisición de datos.
 - **Capítulo III:** En este capítulo se aborda todo el firmware necesario para establecer la comunicación de manera inalámbrica con nuestro robot y la FPGA.
 - **Capítulo IV:** En este capítulo se aborda la aplicación del sistema, así como el software necesario para manipular nuestro robot cartesiano, así como cargarle el firmware necesario para manipular este, se muestran los resultados obtenidos del sistema.

CAPÍTULO I:

CARACTERÍSTICAS DE LAS TARJETAS DE ADQUISICIÓN DE DATOS.

En este capítulo se presenta las características de la tarjeta de adquisición de datos. El objetivo de esta tesis es desarrollar e implementar la tarjeta, la cual tendrá la finalidad de poder manipular un robot de tres grados de libertad de manera inalámbrica empleando el protocolo de comunicación WiFi.

Tarjeta de Adquisición de Datos y Control

Con las especificaciones antes mencionadas, el presente trabajo de tesis se busca mejorar estas características. El diseño tiene el fin de optimizar recursos como, espacio en la tarjeta, ya que usamos un solo componente principal, el cual tiene la tarea de procesar las señales de retroalimentación del sistema, controlar los dispositivos de adquisición, los dispositivos de emisión de señales de control analógicas y de establecer la comunicación con la PC por medio del protocolo de comunicación de WiFi, todo esto descrito en un programa en Quartus II cuyo hardware (por sus siglas en inglés HDL) será sintetizado en el FPGA. Por otro lado, nos permite tener en este mismo dispositivo, las salidas y entradas digitales programables para establecer comunicación con dispositivos externos, todo esto con el fin de poder monitorear y controlar el robot de tres grados de libertad con motores de transmisión directa, así como cuyos motores sean controlados por medio de PWM.

1.3.1 Características de la tarjeta de adquisición de datos de la presente tesis:

- Comunicación WiFi.
- 3 Salidas analógicas de ± 10 volts de 16 bits de resolución (DAC's).

- 30 pines de entradas y salidas para uso general.
- Control de motores por medio de PWM.

Diagrama a bloques de sistema

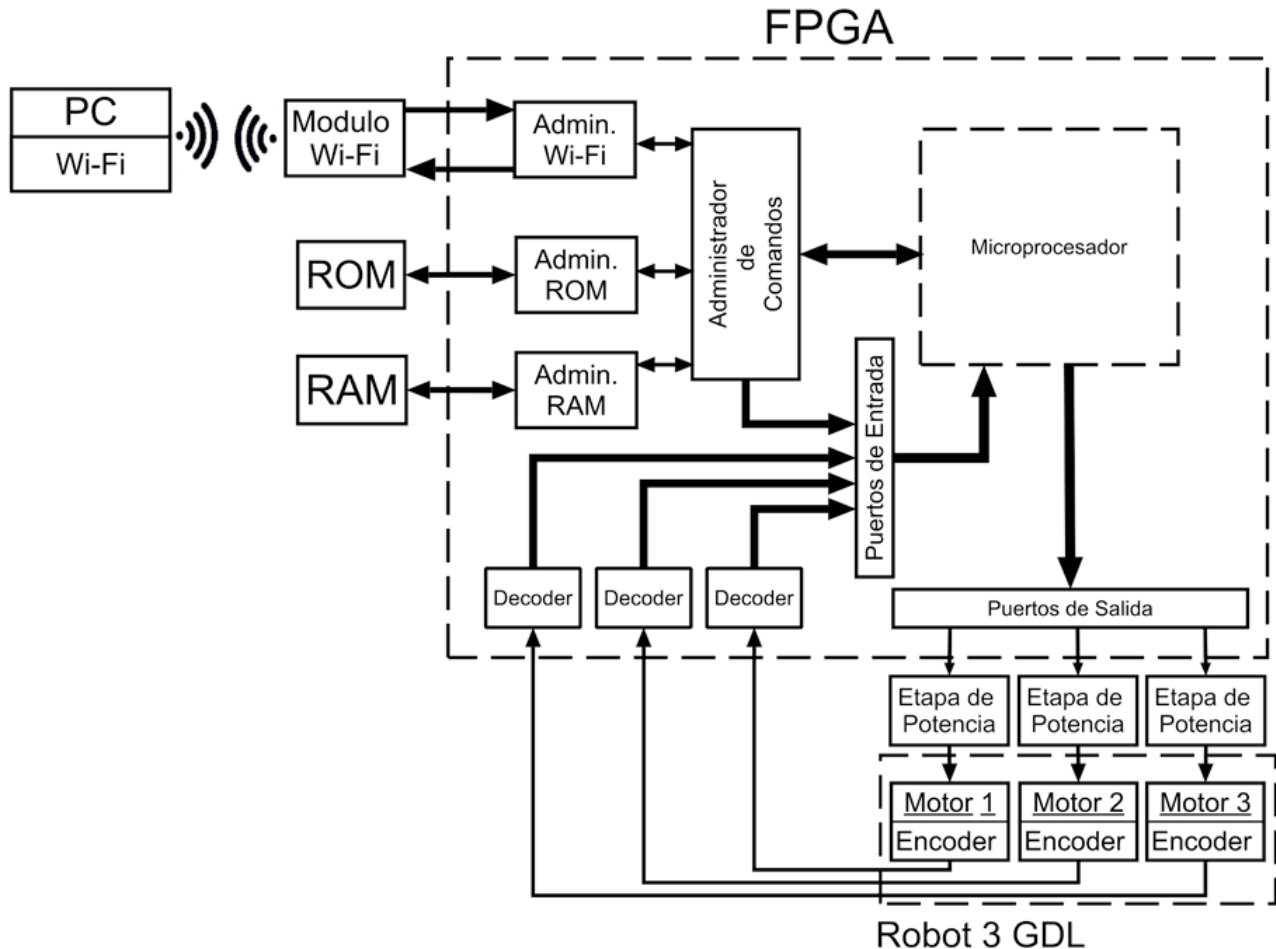
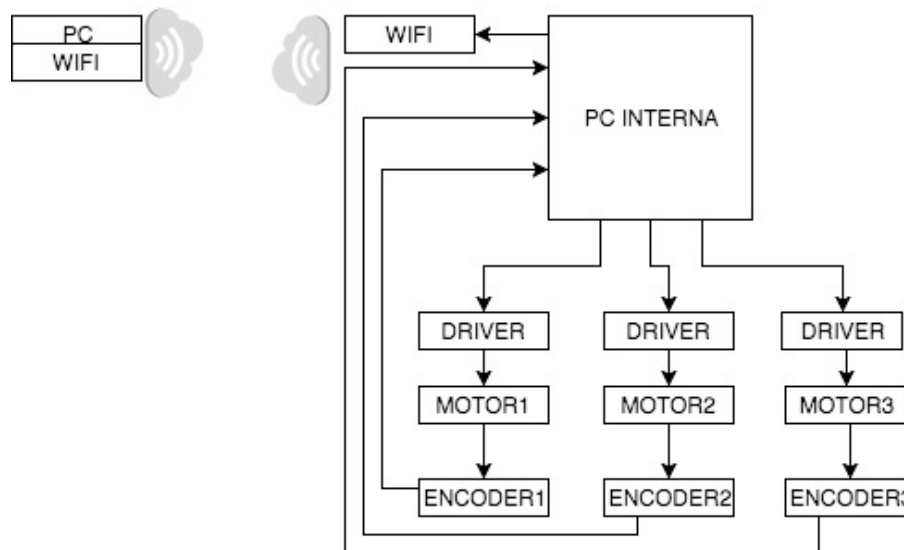


Figura 1: Diagrama a bloques general del sistema de control.

Del diagrama anterior, el elemento a monitorear es la articulación del robot, se obtienen los valores correspondientes a la posición actual y par aplicado, en base al tiempo y a la posición deseada, se calcula el error de posición a lo largo del funcionamiento del sistema. Estos datos son leídos a través de la FPGA y son enviados a un módulo WIFI mediante el protocolo UART de sus siglas en inglés Universal Asynchronous Receiver Transmitter.

El modulo WIFI es el RN-XV, en donde este dispositivo realizara la tarea de conversión de WIFI a serial y de serial a WIFI. Esta comunicación es bastante atractiva para el usuario ya que se hace

compatible con computadoras de escritorio o computadoras portátiles de manera inalámbrica.
[4]



Se necesita que el sistema digital cuente con unidades de almacenamiento para otorgar autonomía en el proceso de control, esto es, que el sistema digital tenga la capacidad de retener la información de parámetros enviados por el usuario y poder utilizarlos en cualquier momento, lo que otorga velocidad de procesamiento, al mismo tiempo, los datos calculados y procesados por el sistema, se almacenarán para su posterior obtención, por tal motivo se integraran memorias RAM y ROM que harán posible dicha tarea.

En el dispositivo se implementarán módulos de lógica programable que conforman las etapas de administración de comunicación Wi-Fi, administración de almacenamiento de información, administración de comandos, decoders, módulo de control y finalmente puertos de entrada y salida. Se pretende que la unidad de procesamiento digital reciba información para el control del sistema mecatrónico a través del módulo de comunicación inalámbrico, para lo cual el administrador de comunicación organizara la información para que el administrador de comandos defina que acción realizar ya sea el caso de guardar variables en la memoria ROM a través de la etapa de administración de ROM, enviar los datos procesados por medio del acceso a la memoria RAM a través de su módulo de administración, o iniciar el proceso de control.

Para iniciar el proceso de control del mecanismo se debe contar con los parámetros de posiciones deseadas para los actuadores y se deberá monitorear la posición de estos por medio de los módulos decoders encargados de transformar las señales recibidas por los sensores del sistema para lograr implementar la ecuación de control, todo esto se realizará en un módulo de control.

Para transmitir la respuesta hacia los actuadores se diseñará la etapa de potencia acorde a los requerimientos de los actuadores.

La ventaja de este sistema, es que, el algoritmo de control se encuentra dentro de la FPGA mediante firmware, aumentando el tiempo de ejecución del algoritmo de control tomando así datos en tiempo real, puesto que es un hardware dedicado a realizar esa tarea específica.

1.5 Conclusión

Al principio del capítulo se menciona las diferentes tarjetas de adquisición de datos que se tienen para controlar un robot de tres grados de libertad, siendo estas de tecnología obsoleta, con la tarjeta de adquisición de datos y control que se desarrollara, se busca mejorar todas estas deficiencias para estar a la vanguardia tecnológicamente hablando y así tener un dispositivo que nos permita tener un control de nuestro robot en tiempo real, facilitando el uso de esta mediante una interfaz donde el usuario podrá manipular libremente el robot.

CAPITULO II

HARDWARE DE LA TARJETA DE ADQUISICIÓN

En este capítulo se muestra el diseño del circuito impreso (PCB, Printed Circuit Board) de la tarjeta madre, para esto se recurrió a el software de diseño Altium Designer, ya que este es muy versátil y de alta calidad.

2.1 Diagrama Eléctrico

Una vez seleccionados los componentes, se diseña el esquemático o diagrama eléctrico de la tarjeta de adquisición de datos y control figura 3.8, donde los componentes que se observan en el diagrama eléctrico son:

- H1, conector de salida, de las señales analógica de control, activación de los manejadores y alimentación de los encoders.
- H2, conector de entrada, de las señales diferenciales de cuadratura.
- H3, conector de entrada/salida para programación del usuario.
- Protocolo de comunicación WiFi.
- FPGA, circuito integrado y su diagrama para el uso de sus pines.
- DAC, circuito integrado y su diagrama de conexión para cada grado de libertad.
- DC-DC, DC-DC y su diagrama de conexión para el umbral de los DAC.

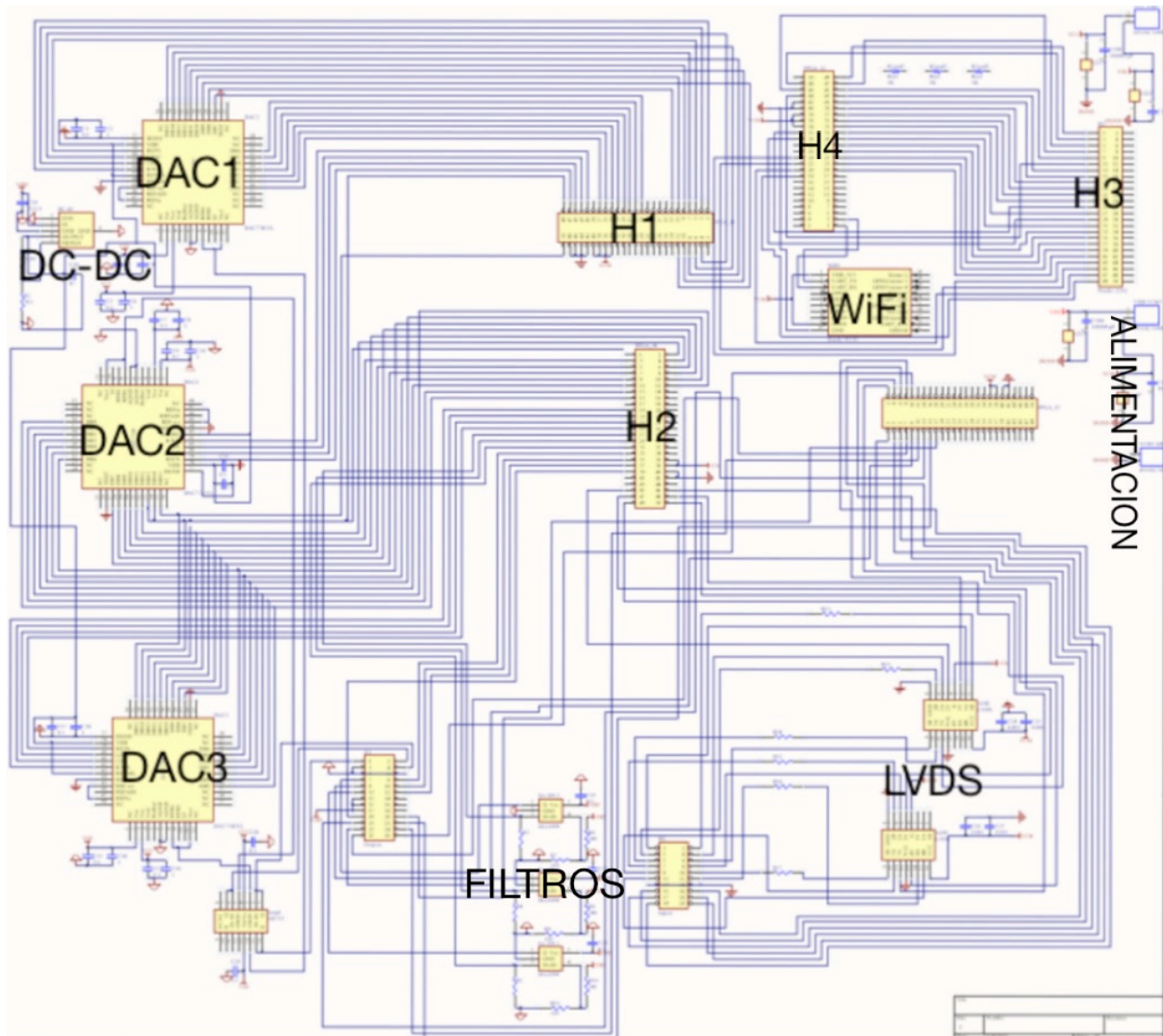


Figura 2: Diagrama esquemático de la tarjeta de adquisición de datos.

El desarrollo de una tarjeta de adquisición y control, con alta integración usando un FPGA, nos permite realizar la solución de ingeniería en una sola pastilla SOC (System On a Chip), con las ventajas de: utilizar menos espacio sobre la tarjeta, menores requerimientos de potencia, procesos de ensamble más rápidos, menos costosos y mayor confiabilidad, dado que se tiene menos circuitos integrados, así como menos conexiones de circuito donde puedan ocurrir fallas, además de desarrollo de tecnología nacional.

El diseño de una tarjeta de adquisición, además de la selección de componentes, tiene que ver con las emisiones electromagnéticas que pueden afectar el correcto funcionamiento de los dispositivos, por lo que en este trabajo se presentan las técnicas usadas en el diseño de la PCB

(printed circuit board), con las cuales se pretende minimizar las RF o EMI, que afectan a los componentes (CI) o a las señales.

En el diseño de una tarjeta de circuito impresa (Printed circuit board o PCB), una de las primeras consideraciones que se requieren es determinar cuántas capas o planos son necesarios. Donde el número de capas es determinado por ciertas características como: inmunidad al ruido y el número de componentes.

Primeramente se crea un proyecto, posteriormente se crean las librerías del circuito de los componentes es decir el *schematic library* asignándole el nombre del dispositivo es importante asignar una numeración o nombre a cada pin del dispositivo ya que es la conexión que el programa seguirá en cuando se diseñe la *pcb library* del dispositivo, una vez que se ha diseñado el *schematic library* de cada uno de los dispositivos se deben crear las huellas (*footprint*), las cuáles serán las que se plasmen en la tarjeta electrónica, por lo que para diseñar el *pcb library* es necesario realizar una medición física correcta de cada dispositivo, una vez que se tiene la huella, se le asigna un nombre y se da de alta en las librerías, así puede ser utilizado para cualquier dispositivo al que se le agregue dicha huella. [5]

Ya que se tienen los componentes y las huellas, se acomodan los dispositivos y se hace el ruteo. Altium tiene la propiedad de realizar un ruteo automatizado, pero se recomienda hacerlo manualmente ya que la experiencia que se tuvo, consistió en reducir la longitud de las pistas para disminuir ruidos, para un circuito el proceso de ruteo automático en Altium se vuelve muy lento y tardado, aunque no se desprecia en algunos casos esta bondad que el programa ofrece.

El uso apropiado de vías, son requeridos para minimizar las radiaciones internas en la PCB. El uso de planos de tierra son el método más eficiente para reducir la inducción de ruido externo en la PCB, además que contribuyen intrínsecamente a reducir la impedancia y la distribución de energía a alta frecuencia.

Una manera de realizar una tarjeta de cuatro capas es haciendo uso de planos de alimentación y planos de referencia, con esto se intensifica la supresión de EMI; en relación a una tarjeta de dos planos, sin embargo, las tarjetas de cuatro capas no son óptimas para eliminar el flujo de

corrientes de RF creadas por los circuitos y trazos, debido a que utilizan a estos como medio de transmisión y propagación.

2.2 Circuitos Integrados (CI)

En base a las características del sistema, el hardware es seleccionado con el fin de obtener un balance entre desempeño y funcionalidad en los circuitos integrados.

Los circuitos integrados de montaje superficial presentan la posibilidad de ser montados en un plano libre de ruido interno, por lo que se obtiene un mejor funcionamiento de los CI, además de las ventajas de pequeño volumen, fiabilidad, economía y prestaciones. Realizando un análisis del sistema de control y de las señales necesarias para el control del robot se eligieron los CI para tratarlas y generarlas.

2.2.1 Acoplamiento de las señales de retroalimentación

Las señales de cuadratura que entrega el encoder figura 3.1, se les inducen EMI (interferencia electromagnética) o RF (radio frecuencias) externas, por lo cual distan mucho de las señales ideales de cuadratura de un encoder.

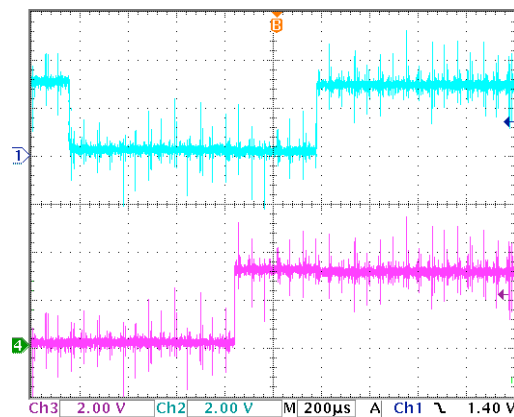


Figura 3: Fases A y B de la señal de cuadratura con ruido.

Las señales de cuadratura generadas por el encoder llegan al manejador y este las entrega a la etapa de control en forma diferencial, dos señales diferenciales por cada grado de libertad, entendiéndose por señal diferencial a la combinación de la señal de cuadratura y su negada. En la figura 3.2 se muestran las señales de cuadratura diferenciales de un grado de libertad.

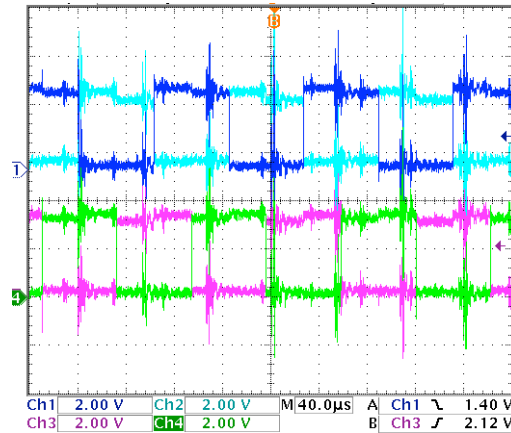


Figura 4: Fases A y B de la señal de cuadratura en forma diferencial.

2.2.2 Circuito integrado seleccionado para el acoplamiento diferencial

Una señal diferencial es aquella que es transmitida en un conductor y su negada es transmitida en un conductor paralelo y cercano. La transmisión de señales diferenciales se utiliza para minimizar el efecto ruido (RF o EMI) externo que se induce en la transmisión de señales, debido a que, con esta técnica, el ruido se induce en ambas señales por igual, por lo que al realizar la suma de estas señales, iguales pero invertidas, el ruido se anula puesto que también posee la misma magnitud pero invertida.

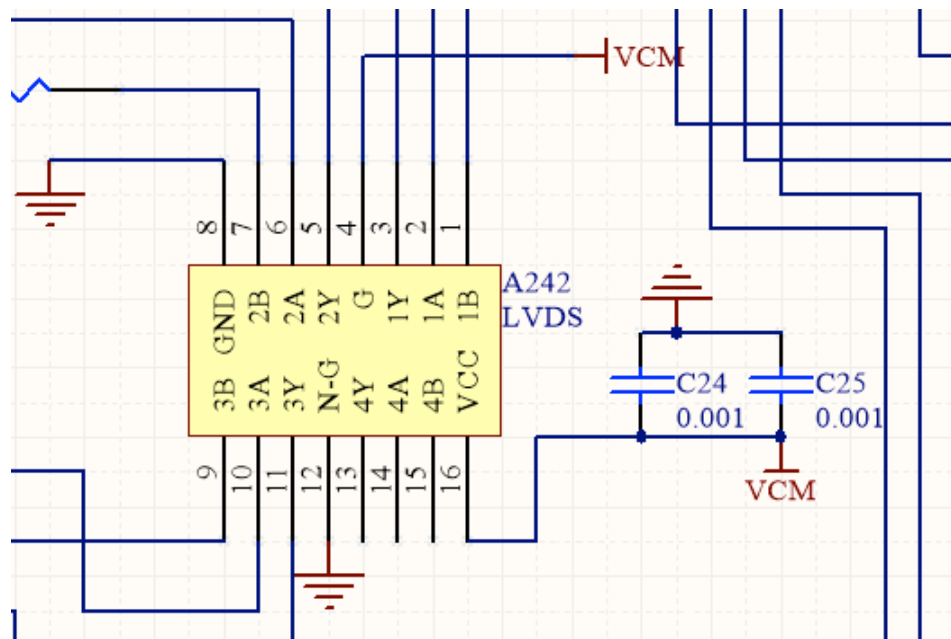


Figura 5: Diagrama esquemático del LVDS.

La señal de cuadratura diferencial es tratada con el SN65LVDT32B (LVD, low voltaje diferencial), el cual es un integrado compatible a CMOS, para tratar señales diferenciales a bajo voltaje. Debido a que es un integrado de bajo voltaje, es necesario reducir la magnitud de las señales de cuadratura, por ello se utilizó el LVDT en lugar de un LVDS; donde la diferencia recae en que el LVDT posee una resistencia de 100 ohm de terminación entre las fases de la señal diferencial, por lo que colocando una resistencia más en serie se consigue la reducción de la magnitud diferencial de las señales. El LVDT trata la señal en base a la ecuación 3.1. [6]

$$V_A = V_{A+} + V_{Ar} - V_{A-} + V_{Ar}, \quad (3.1)$$

donde V_{A+} es la señal de cuadratura, V_{A-} es la señal de cuadratura negada y V_{Ar} es la RF o EMI inducida en las señales, (estas RF o EMI se les denomina como ruido en un lenguaje coloquial), entonces:

$$V_{A-} = \text{negada}(V_{A+}), \quad (3.2)$$

al resolver la ecuación 3.1, nos da como resultado dos veces la señal original y las RF o EMI inducidas se eliminan por poseer igual amplitud, pero magnitud inversa ecuación 3.3:

$$2V_A, \quad (3.3)$$

Según la hoja de datos la magnitud diferencial de las señales debe ser menor a 0.8V, ecuación 3.4

$$V_{A+} - V_{A-} < 0,8V, \quad (3.4)$$

por lo que se coloca una resistencia en serie con la resistencia de terminación del LVDT, consiguiendo un divisor de voltaje a la entrada de las señales diferencial, para obtener la condición de la ecuación 3.5

$$V_{A+} = 0,4V \text{ y } V_{A-} = 0,4V. \quad (3.5)$$

La señal única que se obtiene de la suma de una señal diferencial es analógica y del doble de amplitud de las señales originales, por lo cual se reduce la amplitud pico a pico aproximadamente a la mitad, para posteriormente agregar una componente en cd de la mitad de la amplitud pico a pico de la señal todavía analógica y obtener finalmente una señal digital, esto puede ser comprendido observando la figura 3.3

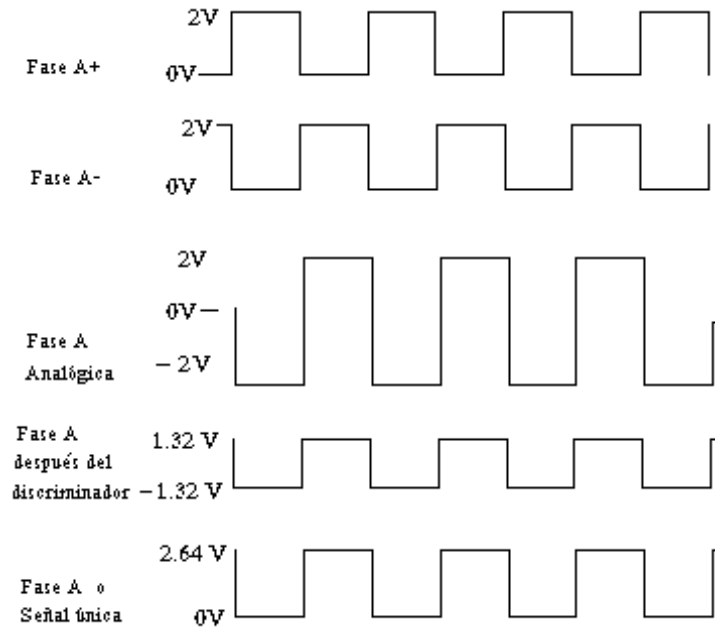


Figura 6: Suma de la señal diferencial.

Pero en caso del LVDT seleccionado este proceso de corrección de la señal única resultante es interno, por lo que en la salida de cada suma diferencial obtenemos una señal única de 3.3V de amplitud, esto puede ser mejor comprendido en el diagrama de tiempos en la figura 3.4, donde $V_o = 3.3\text{Volts}$ y un flanco de ascenso/descenso $t_f < 1\text{ns}$.

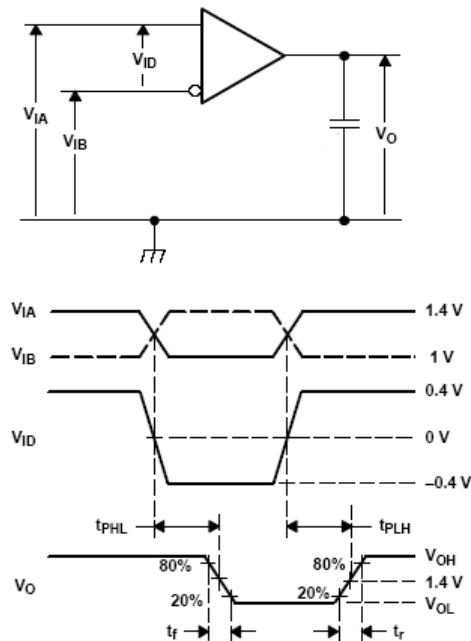


Figura 7: Diagrama de tiempos del LVDT.

El LVDT posee 200Msps, con lo cual se tiene una velocidad de muestreo superior a la requerida. Las fases A y B únicas de las señales de cuadratura que se obtiene del acoplamiento diferencial se muestra en la figura 3.5, en las señales resultantes del acoplamiento diferencial se observa ruido de alta frecuencia de amplitud menor a 1 Volt.

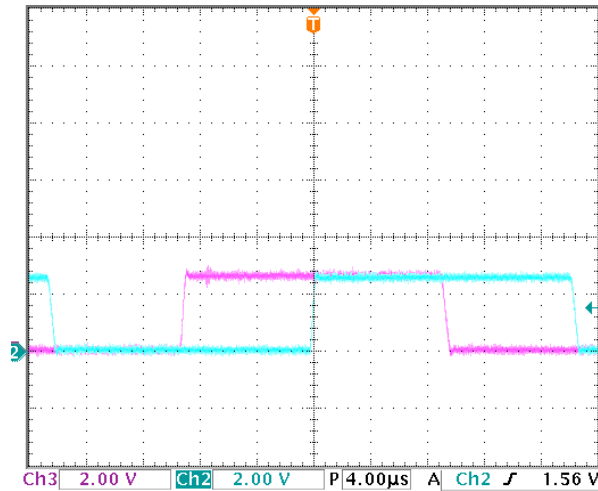


Figura 8: Fases A y B de las señales de cuadratura del encoder.

2.2.3 Acoplamiento Digital- Analógico

El acoplamiento digital analógico se realiza comúnmente con uno o varios DAC (Digital Analogical Converter) que como su nombre lo indica convierte un dato en formato binario en una señal analógica. El dato en formato binario puede ser de diferentes tamaños, por ejemplo, de 8bits o 16 bits, es decir el dato puede variar de 0 a 255 o de 0 a 65535 combinaciones posibles, y la señal analógica que proporciona en su salida el DAC varia en un rango determinado, por ejemplo, puede ser de $\pm 8\text{Volts}$, $\pm 10\text{Volts}$ o $\pm 12\text{Volts}$ por mencionar algunos, entonces se dice que un DAC es de un voltaje de salida dado, por ejemplo $\pm 10\text{V}$, a una resolución de 16bits, que es el rango del dato de entrada. [7]

Debido a la modalidad de trabajo de los manejadores que es la modalidad de par, la señal de control que reciben los manejadores es analógica, en un rango de $\pm 10\text{V}$, a una resolución de 16bits, es decir a escala completa 16bits corresponde un voltaje de 10V de salida y para la escala mínima de 0bits corresponde a -10V de salida. La señal entrega un par cero al aplicar una señal de 0V, es decir a media escala de la resolución de 16 bits que le corresponde un valor numérico de 8000H o 10000000B. Entonces el acoplamiento digital analógico se usa para convertir el valor o magnitud de par que se calcula en la PC como respuesta de control al robot.

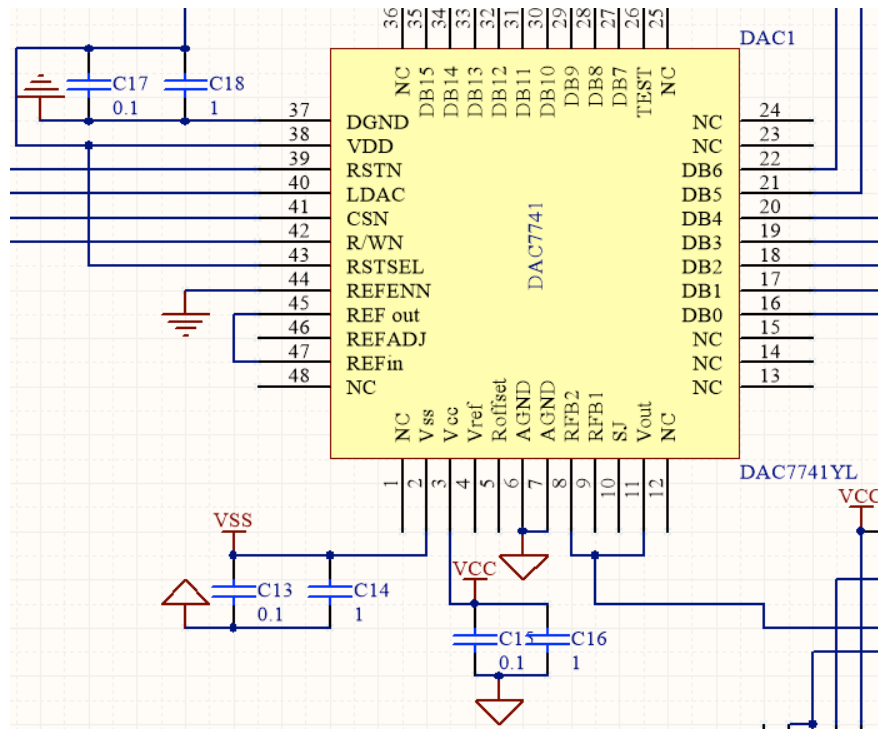


Figura 9: Diagrama esquemático del DAC.

2.2.4 DAC

El DAC seleccionado en un CI de superficie cuya matrícula de identificación es DAC7741YL de Texas Instrument figura 3.6.

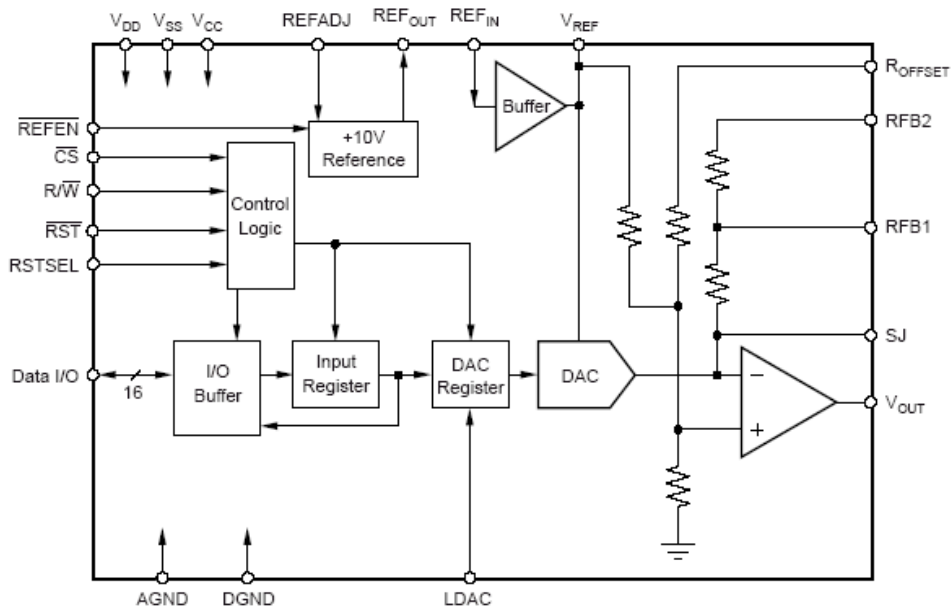


Figura 10: Diagrama a bloques del DAC utilizado.

A continuación, en listaremos las características principales de este integrado:

1. Máximo consumo de potencia de 150mW
2. Reset a media escala de 16bits (8000H)
3. Rango de voltaje de salida de $\pm 10V$ a una resolución de 16 bit.

Este DAC posee referencia interna, lo que nos ahorra una señal, el reset a media escala nos permite enviar una señal de par cero cuando no se esté ejecutando el algoritmo de control, además de contar con dos registros de entrada lo que nos permite manipular la señal de salida con mayor facilidad. El integrado debe ser alimentado con $\pm 12V$ y $4.7V$, es necesario utilizar un convertidor de DC a DC ($5V$ a $4.7V$). La corriente máxima que entrega en la salida este DAC es de $15mA$, la cual es menor a la corriente máxima que puede requerir el manejador del servomotor que es de $25mA$, por lo que es necesario el uso de un búfer que nos permita satisfacer estos requerimientos de corriente.

Estos componentes poseen 5 señales de control además de los 16 bits de datos, he internamente cuentan con dos registros, uno de entrada y uno de escritura.

Las señales de control son:

- RW.
- CS.
- RST.
- RSTSEL.
- LDAC.

Estas señales se utilizan para determinar el modo de trabajo de los DAC, los modos de trabajo de estos DAC son:

- Escribe el dato en el registro de entrada.
- El registro del DAC carga el dato del registro de entrada.
- Escribe el registro del DAC directamente en el Bus de datos del DAC.
- Lee el dato del registro de entrada.
- No cambio (mantiene el voltaje de salida constante).

- Reset al registro del DAC a la mínima escala (0000H).
- Reset al registro del DAC a media escala (8000H).

Los modos en los cuales se van a alternar los DAC para la aplicación desarrollada son:

Modo 1: escribe el registro del DAC directamente en el Bus de datos del DAC.

Modo 2: no cambio (mantiene el voltaje de salida constante).

Modo 3: reset al registro del DAC a media escala (8000H).

La configuración de las señales de control que se van a utilizar son:

- RW: fijo en GND.
- CS: a GND para el modo 1 y VCC para el modo 2.
- RST: en VCC para el modo 1 y GND para el modo 3.
- RSTSEL: fijo en GND.
- LDAC: en flanco ascendente para el modo 1.

Para la prueba de funcionamiento de los DAC se determinó que, contrario a la hoja de especificaciones que indica un tiempo de espera de 40nseg entre las señales CS y LDAC puede ser nulo, es decir pueden cambiar de bajo a alto simultáneamente para pasar del modo 2 al modo 1, y RSTN debe de estar en alto 30 nseg antes que CS y LDAC está en alto para el modo 1 y en bajo para el modo 3.

La alimentación de 12, -12 y 4.75 Volts debe de proporcionarse en el mismo instante puesto que de lo contrario el funcionamiento de los DAC sería incorrecto. Una vez determinado lo anterior y utilizando LabView se escriben valores en los DAC verificando con un osciloscopio que el voltaje de salida corresponda al dato, además de verificar la transición de los modos de trabajo de los DAC.

2.2.5 Convertidor de DC-DC

El convertidor de voltaje de corriente directa a corriente directa TPS75501KTT de superficie tiene la ventaja de entregar un voltaje de salida en un rango de 1.22V a 5V, con una simple configuración de resistencias.

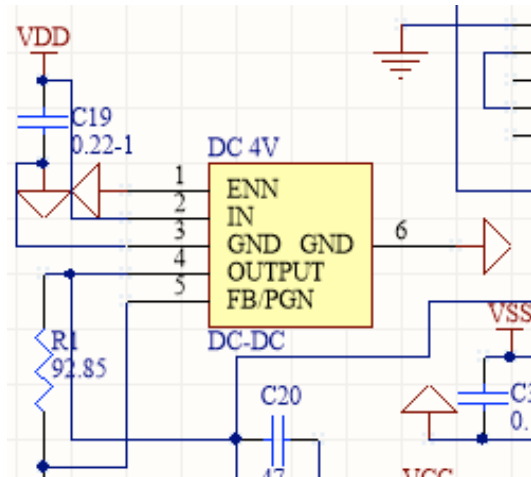


Figura 11: Esquemático del DC-DC.

2.2.6 Amplificador operacional

El circuito integrado en base a amplificadores operacionales de Analog Devices el AD713, es un CI que entrega hasta 12V y una corriente máxima indefinida, con un periodo de escritura de $2\mu\text{s}$, con lo cual se obtiene la corriente máxima que el manejador puede consumir y se tiene una frecuencia de escritura superior a la necesaria que es de 2ms.

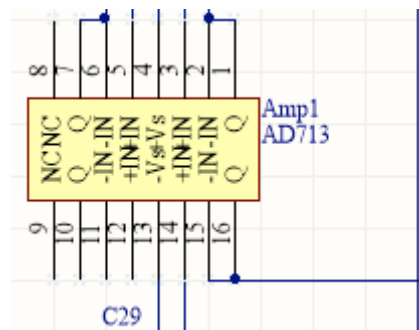


Figura 12: Esquemático del amplificador operacional.

2.3 Activación de los manejadores

Los manejadores son la electrónica de potencia de los motores del robot y son activados por una señal de 5V, además de entregar las señales de cuadratura y recibir la señal de respuesta del algoritmo de control.

2.3.1 Señal de activación

Los manejadores de los servomotores que constituyen al robot ya mencionado anteriormente, poseen una señal de activación, la cual cumple con la función de permitir al manejador estar en funcionamiento únicamente cuando se está ejecutando el algoritmo de control, esta señal debe ser manipulada $2\mu s.$, antes de que se aplique la señal analógica de par.

Las características de la interacción de la señal de activación del manejador del servo- motor (Servon o Son, que es la forma como se denomina a esta señal en el manual de los motores [4]), se enlistan a continuación:

- Manejador activado valor de la señal 0V.
- Manejador desactivado valor de la señal 5V.

2.3.2 CI seleccionado para la señal de activación de los manejadores

El CI seleccionado para generar la señal de activación y desactivación de los manejadores es el MAX999, el cual es un comparador de alta velocidad (50Mhz de respuesta). Este dispositivo es necesario debido a que las señales del FPGA son de 3.3V y el comparador seleccionado entrega hasta 6V en su salida. Para esta aplicación el MAX999 es alimentado con 5V, voltaje máximo que, por consiguiente, entrega en su salida, el voltaje al cual se compara es decir en su referencia es de 3.3V, por lo cual al aplicar el FPGA un uno lógico 3.3V, el CI nos entrega en su salida 5V, mientras que el voltaje de la señal del FPGA sea cero, en su salida el comparador entrega 0V.

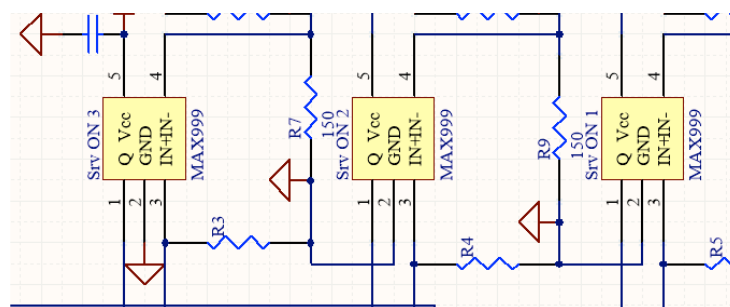


Figura 13: Diagrama esquemático de los MAX999

2.3.3 CI seleccionado para el procesamiento de las señales del sistema de control

El circuito integrado principal de la tarjeta se encarga del procesamiento de las señales de control, y del control de los demás componentes en la tarjeta. En la actualidad el bajo costo, alto desempeño y un gran número de pines de usuario de los FPGA, aunado un lenguaje de programación más simple, han hecho que este tipo de integrados sean cada vez más utilizados.

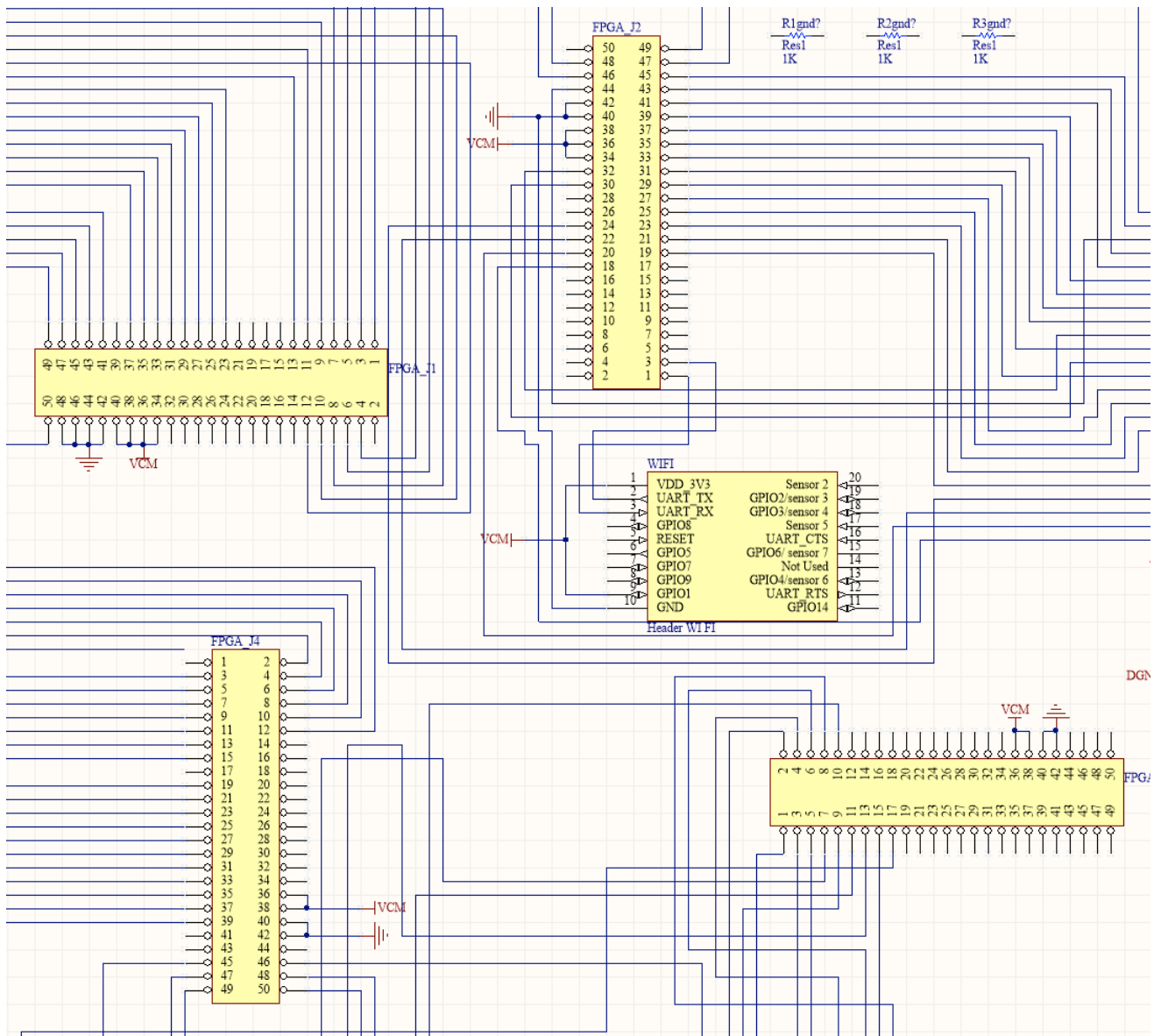


Figura 14: Diagrama esquemático de la FPGA.

2.4 Estrategias para el diseño

En el diseño del PCB de la tarjeta de madre de este trabajo se consideró un factor importante, el ruido, el cual es una señal no deseada que podría interferir con el correcto funcionamiento del sistema. Existen dos tipos de ruido: externo e interno. A continuación, se detalla el ruido interno.

2.4.1 Ruido interno:

Interferencia electromagnética por las fuentes de alimentación

La interferencia electromagnética (EMI, Electromagnetic Interference) en las fuentes de alimentación, se deben al ruido producido por la fuente de voltaje de corriente directa (DC, Direct Current); este ruido se produce por la rectificación, es decir, del voltaje de rizo presente, tal como se muestra en la Figura 2.2.

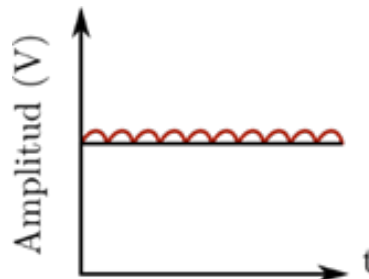


Figura 15: Voltaje de rizo de una fuente de alimentación.

Esto transforma a la fuente ideal de DC, en dos fuentes, una con voltaje de DC y otra con el voltaje de corriente alterna (AC, Alternating Current), tal como se muestra en la Figura 2.3.

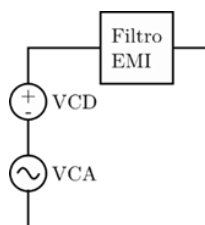
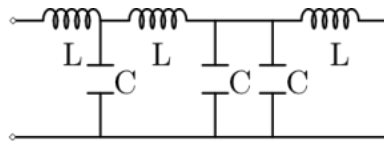


Figura 16: Fuente de voltaje con filtro EMI.

En una fuente de computadora, el voltaje de rizo es menor, debido a que son conmutadas. Al incrementar la frecuencia de trabajo, la carga y descarga son menos perceptibles y el voltaje de rizo es menor. En la tarjeta madre (*Motherboard*) de una computadora, existen oscilaciones que inducen ruido; Cuando se monta una tarjeta PCI a la computadora sin protección, se conecta la fuente de alimentación y todas las fuentes de ruido. Este fenómeno se reduce con los filtros EMI.

Existen varios tipos de Filtros EMI, entre los filtros EMI de Mitsubishi, cuyo circuito equivalente se muestra en la Figura 2.4. Este filtro es un arreglo de capacitores e inductores integrados en el chip que permiten regular la corriente y el voltaje. El rizo se aterriza y es absorbido por el filtro. Los filtros EMI reducen el ruido proveniente de la red eléctrica de 110v.

El filtro EMI LFB30-3A1E223M de Mitsubishi soporta un voltaje máximo de 25V, una corriente máxima de 1A, tiene una frecuencia de corte de 22MHz y un rango de temperatura de -25°C a 85°C.



Mitsubishi LFB30-3A1E223M

Figura 17: Circuito equivalente y especificaciones del filtro EMI.

Este filtro tiene dos condensadores unidos por una bobina. La carga y descarga del primer condensador, produce un efecto similar al rectificador de media o doble onda con filtro a condensador. La resistencia entre ambos condensadores hace que se suavice la señal, llegando así al segundo condensador, una corriente relativamente constante. La carga y descarga de este último condensador, debido a la componente alterna, suaviza aún más las fluctuaciones, y a la carga llegará una corriente continua relativamente pura. Se emplean únicamente cuando la corriente demandada es pequeña, así la caída de tensión en la resistencia entre condensadores, será despreciable. En este trabajo no será necesario utilizar este tipo de ruido ya que descartamos la utilización del puerto PCI.

2.4.2 Interferencia entre pistas

El diseño de la PCB tiene 4 capas; las pistas para las señales se localizan en la capa superior (Top Layer) e inferior (Bottom Layer); El plano de alimentación (Vcc) y tierra (GND) se localizan en las capas intermedias. Cuando la tarjeta está en funcionamiento, el 90 ó 95% del campo electromagnético fluye en el plano de tierra, sin embargo, el campo restante interfiere con la siguiente pista, tal como se muestra en la Figura 2.5; esto produce un fenómeno denominado ruido por interferencia entre pistas con el plano de tierra, el cual se presenta a partir una frecuencia de 50MHz.

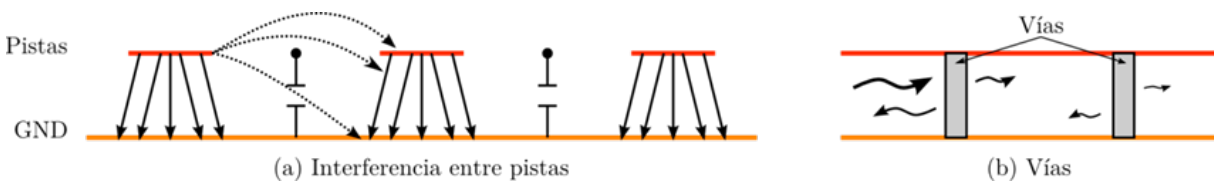


Figura 18: Interferencia entre pistas.

El voltaje de ruido inducido por este fenómeno es de 150mV; la interferencia electromagnética se incrementa con el número de pistas y entradas digitales, por esta razón, por cada dispositivo se debe colocar un capacitor de $0.1\mu\text{F}$ conectado de Vcc a GND, y deben ser distribuidos en toda el área de la tarjeta para reducir la interferencia electromagnética entre pistas (suficiente para sistemas digitales CMOS, LVTTTL y TTL). Para reducir la interferencia entre pistas, se aterrizan los componentes electrónicos utilizados y algunos otros puntos con capacitores de $0.1\mu\text{F}$, tal como se muestra en la Figura 2.5.

2.4.3 Interferencia entre etapas analógica y digital

En tarjetas con etapa analógica y digital, es recomendable separar la tierra de estas etapas, pues se genera una interferencia entre dichas etapas de aproximadamente 100-150mV. Para eliminar esta interferencia se utilizan: filtros EMI, para eliminar el ruido producido por las fuentes de alimentación; y capacitores por cada dispositivo, distribuidos en toda la tarjeta, para reducir la interferencia entre pistas analógicas.

La etapa analógica se blinda mediante la inserción de vías, esto reduce el ruido, debido a que la vía es un poste metálico aterrizado. Cuando la interferencia electromagnética pasa por el poste,

la señal electromagnética se divide en dos, la primera parte se refleja y sale de la tarjeta; la otra parte traspasa el poste, y se atenúa exponencialmente conforme encuentra otra vía, tal como se muestra en la Figura 2.6. Así, la interferencia entre la etapa digital y analógica se reduce, y el voltaje del ruido quedaría en el orden de 5mV.

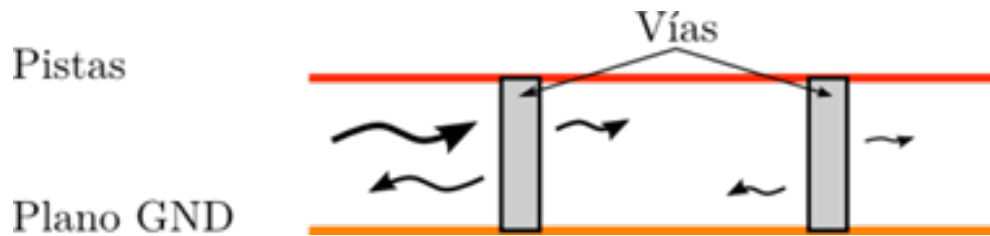


Figura 19: Disminución de interferencia por vías.

Las tierras entre la parte digital y analógica, se unen en puntos en donde, el ruido tenga menos posibilidad de atravesar.

2.4.5 Ruido externo

Nuestro entorno está inundado de ruido externo provocado por: las computadoras, ondas electromagnéticas de radio, televisión, telefonía, etc.; sin embargo, este ruido no es significativo para la tarjeta PCI.

En un metal, a temperatura constante, la densidad de corriente J es linealmente proporcional al campo eléctrico E .

$$J = gE \quad \text{Ec. 2.1}$$

donde g es la conductividad.

La densidad de corriente J , caracteriza a un punto interno del conductor. Si la corriente i se distribuye uniformemente a través del área A de la sección transversal de un conductor, la magnitud de la densidad de corriente de los puntos en esa sección transversal es:

$$J = \frac{i}{A}, \Rightarrow \frac{i}{A} = gE \Rightarrow E = \frac{i}{A}g, \quad \text{Ec. 2.2}$$

la conductividad idealmente tiende a infinito y por lo tanto

$$E = 0. \text{ Ec. 2.3}$$

Por esta razón, en una cierta área alrededor de la tarjeta no existe ningún factor externo incidiendo, así, se usan componentes superficiales de tal manera que se ubiquen en el área carente de ruido externo.

2.5 Diseño de la PCB

El diseño del PCB de la tarjeta madre, consta de cuatro capas de cobre y tres de material dieléctrico. Las pistas para las señales se localizan en la capa superior e inferior; El plano de alimentación y tierra se localizan en las capas intermedias, tal como se muestra en la Figura 2.9.

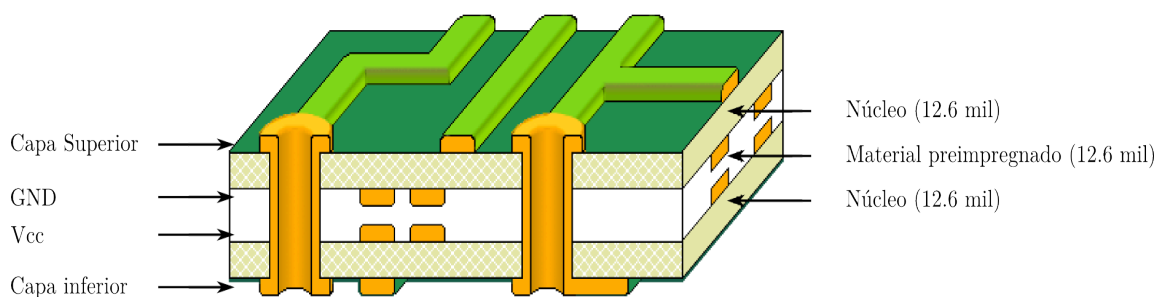


Figura 20: Capas de la tarjeta.

Para insertar cada dispositivo, Altium posee una gran cantidad de librerías, sin embargo, si no son suficientes, el programa permite diseñarlos; para el diseño de esta tarjeta, se crearon algunas librerías, como las que contienen el módulo Wi-Fi, etc. [8]

2.5.1 Diseño del primer plano (Top-Layer)

Con el diagrama eléctrico nos podemos dar cuenta de los requerimientos de alimentación de los CI, además de poder ver el número de pistas que se necesitan para unir cada entrada/salida (E/S) de los componentes, las señales de control y las señales de adquisición. Con el uso de Altium Designer (software especializado para el diseño de PCB) se compila el esquemático creando la primera capa del PCB, la cual debe de ser trazada en un espacio previamente definido de acuerdo al tamaño que se requiere de la tarjeta. El tamaño de la tarjeta está directamente relacionado con el esquemático y con el espacio que se tiene disponible dentro de una PC para alojar a la tarjeta, una vez que la tarjeta utilizara el protocolo WiFi como medio de comunicación. En la figura 2.10 se muestra la primera capa de la tarjeta, en la cual irán los

componentes de superficie y las pistas necesarias, de acuerdo al circuito eléctrico.

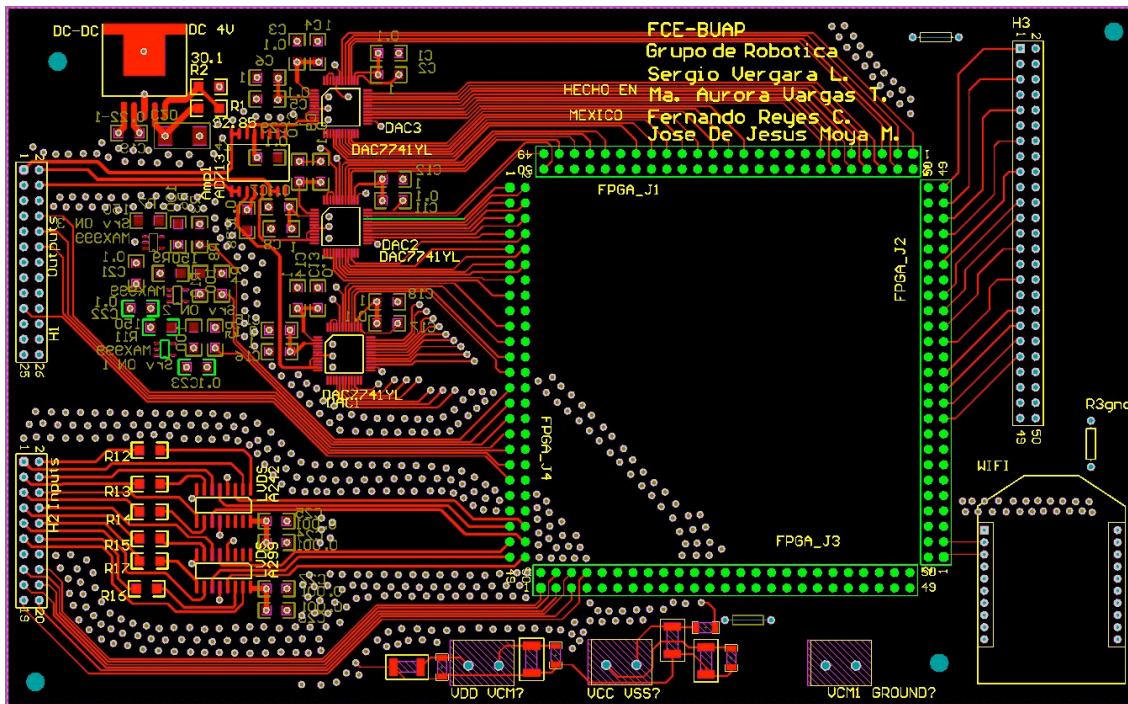


Figura 21: Capa Superior(Top-Layer).

Los componentes son distribuidos de acuerdo a las señales con las cuales van a trabajar, por lo que se distribuyen de manera que el acoplamiento digital-analógico no afecte al acoplamiento diferencial y a la vez el modulo wifi no induzca ruido en ningún componente, puesto que junto con el FPGA son los componentes que trabajan a mayor frecuencia y que más ruido pueden inducir en los componentes adyacentes. Es por ello que podemos dividir la superficie de la capa superior de la tarjeta en cuatro secciones.

La primer sección es el modulo WiFi el cual se encuentra próximo a la segunda sección que es el FPGA, la tercera sección es el acoplamiento diferencial el cual debe estar alejado de los componentes que trabajan a mayor frecuencia como el FPGA y el reloj del módulo WiFi, y alejado de componentes que trabajan con señales analógicas como lo es acoplamiento digital analógico, el acoplamiento digital analógico se encontraría en la cuarta sección junto con el convertidor de DC a DC, los discriminadores y el buffer. El discriminador no afecta a ningún componente puesto que solo escribirá una señal al inicio y al final del funcionamiento de la tarjeta.

2.5.2 Diseño del segundo plano (GND)

Teniendo en cuenta que los componentes y la mayoría de las pistas del circuito eléctrico se encuentran en la primera capa, para una correcta transmisión de las señales se coloca en un segundo plano a GND, con lo cual se reducirán las emisiones electromagnéticas externas cerca de la tarjeta. En la figura siguiente se presenta el diseño del plano de referencia o GND.

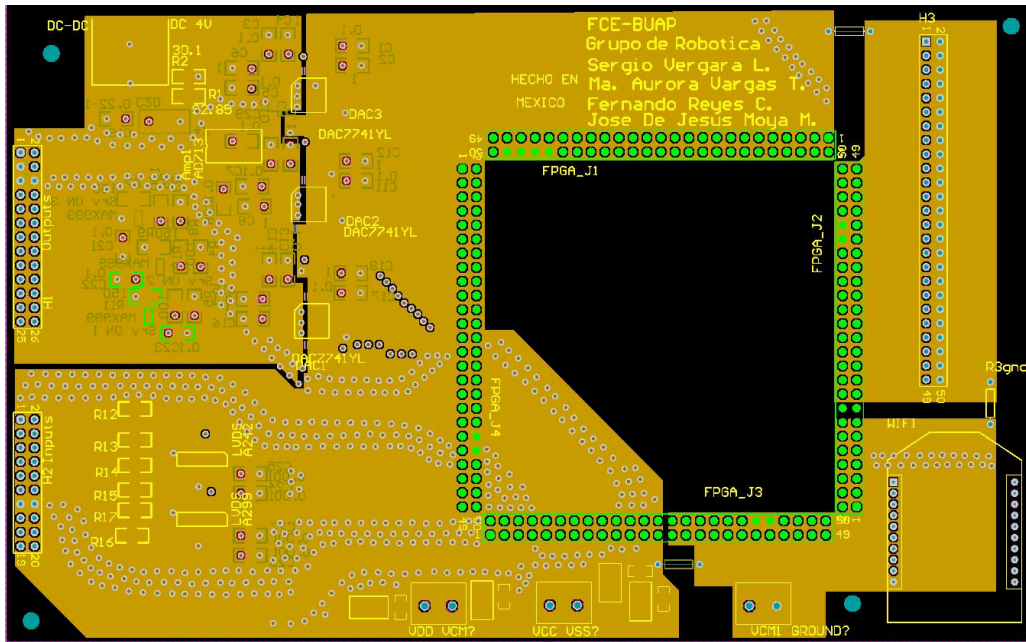


Figura 22: Capa de GND.

El ruido interno de la tarjeta se propaga entre las distintas áreas de la tarjeta y entre los dispositivos por lo que es de vital importancia disminuir este ruido. En este diseño utilizamos dos técnicas para disminuir el ruido interno; la primera es la separación de las tierra digital y analógica unidas sólo en un punto mediante un semiplano de área muy pequeña para obtener una misma referencia, el segundo método se realiza colocando una barrera de metal que impida la propagación del ruido. La barrera es colocada perpendicular a los planos y consta de cilindros metálicos huecos unidos al plano de GND, a estos cilindros metálicos se les llama vías, estas técnicas minimizan el ruido interno en la tarjeta.

2.5.3 Diseño del Tercer plano (VCC):

El plano de alimentación, es colocado en el tercer plano, los requerimientos de voltaje se determinaron del diagrama eléctrico realizado, estos voltajes son llevados al plano de VCC a través de vías, y cada voltaje necesario en la tarjeta es llevado a cada componente haciendo uso de semiplanos. En la figura siguiente se muestra el plano de alimentación o VCC de la tarjeta Tarjeta de Adquisición de Datos.

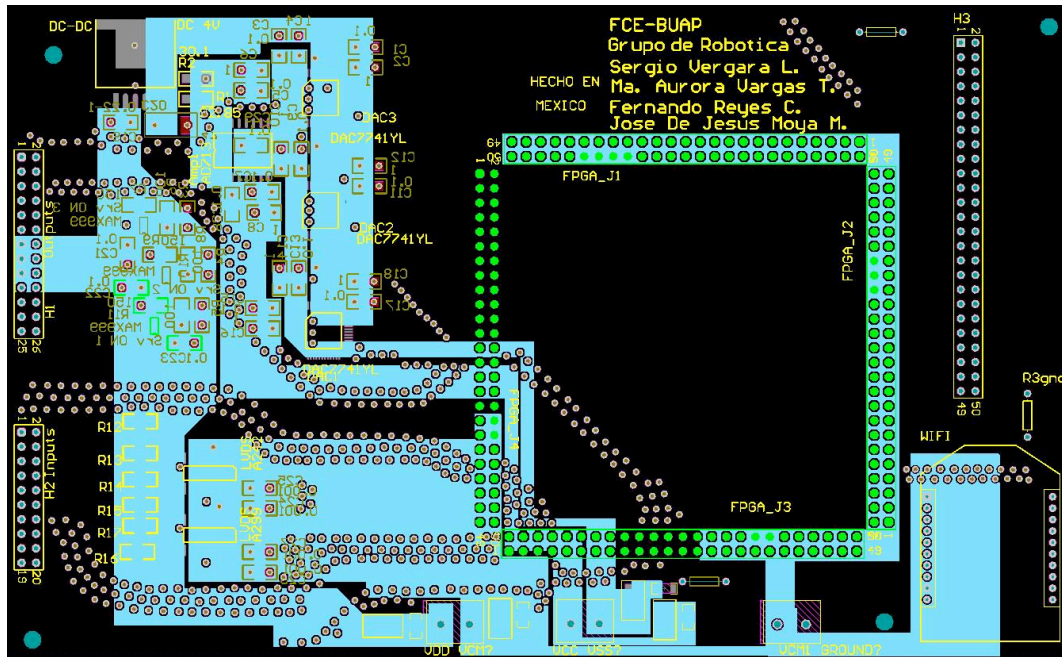


Figura 23: Capa de VCC.

Para el caso del DAC fue necesario el uso de convertidor de DC a DC, el cual nos suministra en la tarjeta 4.7 Volts a partir de 5 Volts, mediante una vía colocada cerca de la salida de voltaje del DC-DC se lleva el voltaje hasta el semiplano de VCC que suministra el voltaje al DAC. La ubicación de los componentes en la capa superior debe de considerar que sea posible suministrar el voltaje respectivo a cada componente sin necesidad de ser trazado un semiplano en la capa superior cerca de alguna señal.

2.5.4 Diseño del cuarto plano (Bottom-Layer)

La cuarta capa esta paralela al plano de alimentación por lo que el ruido externo es más intenso que en la capa superior, para evitar la existencia de cualquier ruido interno en este plano todos los capacitores utilizados en esta tarjeta se colocaron en esta capa.

En este plano se colocaron solo tres componentes que son los discriminadores, que son circuitos integrados que solo emiten una señal al inicio y al final del control del robot, el umbral de la señal que emiten es superior a 2 Volts lo cual es superior a cualquier ruido interno o externo que pudiera existir en este plano y las pistas trazadas en esta capa son para señales de uso general de entrada salida programables para el usuario que no intervienen en el proceso de control del robot y pistas para las señales del bus PCI en las cuales el ruido inducido en ellas no causa ninguna afectación. En la figura 2.13 se muestra el cuarto plano o bottomlayer.

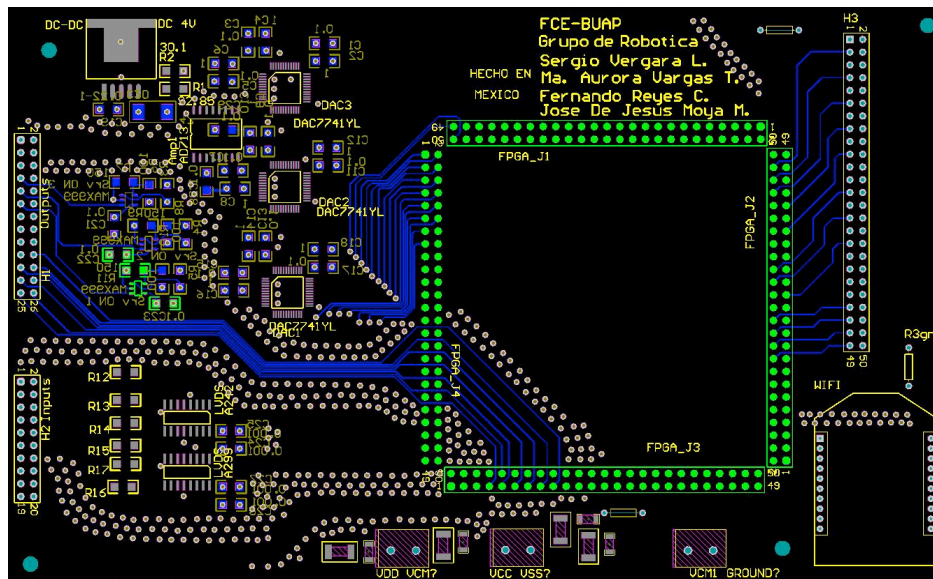


Figura 24: Capa de Bottom-Layer.

Podemos ver los impresos para los capacitores que utilizan los componentes analógicos, los impresos de los discriminadores, las pistas de uso general en la parte central izquierda de la tarjeta delimitada por vías y en la parte inferior derecha las pistas del módulo WiFi.

2.5.5 Tarjeta de Adquisición y Control

La tarjeta está dividida en 2 secciones, la primera sección es la de adquisición de datos y una segunda sección de salida de señales. La adquisición diferencial se realiza mediante dos SN65LVDT32B para tratar hasta 6 entradas diferenciales, tres DAC7741YL de 16 bits para obtener tres salidas analógicas correspondientes a la señal par que debe ser aplicada a cada manejador de los motores, tres salidas digitales de control conectadas a tres discriminadores MAX999 para producir una señal de activación a cada manejador del robot compatible con TTL y 28 entradas/salidas digitales configurables, como componente central para el procesamiento de las señales y control de los componentes se tiene un FPGA Cyclone III EP3C8Q208C7 de ALTERA.

En la figura 2.14 se puede observar el diseño final de la tarjeta de adquisición de datos

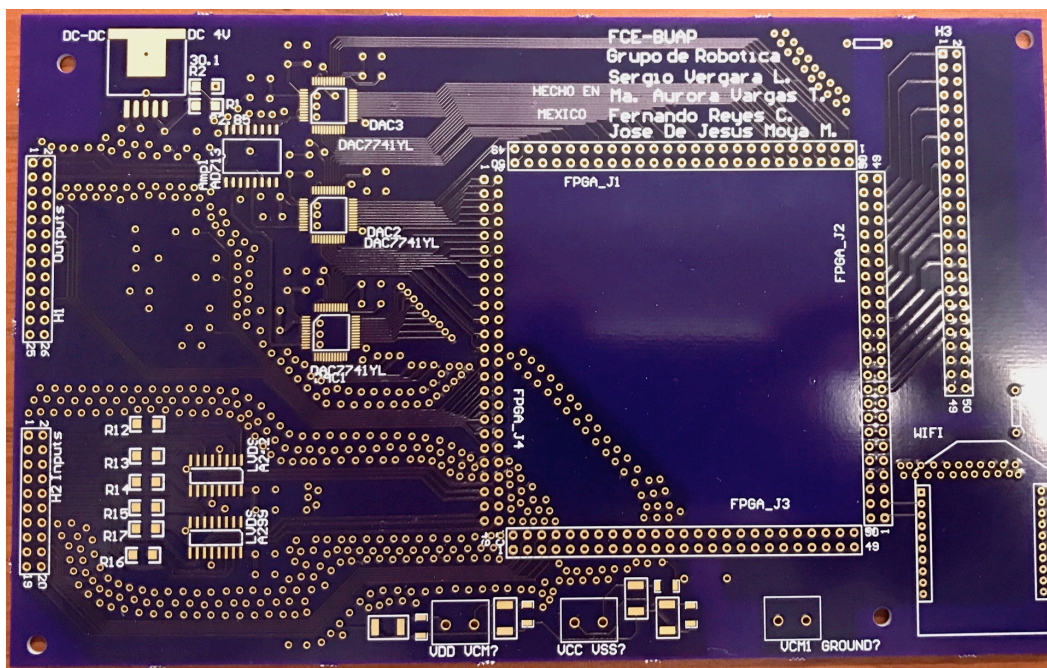


Figura 25: Diseño final de la tarjeta.

2.6 Conclusión

Diseñar una tarjeta nos ofrecen la ventaja de reducir considerablemente el tamaño de nuestro sistema, además, como ya se había mencionado el módulo RN-XV, al tener la capacidad de establecer una comunicación WIFI reduciendo cableado, por ende, el tamaño se reduce aún más, y bajo este aspecto, el costo de fabricación de la tarjeta se redujo también. Es importante utilizar las estrategias de reducción de ruidos ya que al tener un sistema que trabaja con altas frecuencias, el ruido puede provocar lecturas falsas. El software Altium al ofrecernos la capacidad de diseño de tarjetas de circuito impreso multicapa, igualmente contribuye a la reducción de espacio y del costo en la fabricación de la tarjeta, el costo de esta tarjeta se redujo a comparación de diseños previos.

CAPITULO III

FIRMWARE DE LA TARJETA DE ADQUISICIÓN

En este capítulo abordaremos el firmware de control de movimiento para un robot de tres grados de libertad, teniendo a la mano en el laboratorio de la MCEA, la estructura de un robot cartesiano, será vital para poder utilizar la tarjeta de adquisición de datos diseñada en la presente tesis, además se abordará el tema de la decodificación de los encoders y la generación de la señal de control, se mencionará el protocolo de comunicación WiFi, siendo una parte básica en el firmware del sistema.

se retomó el firmware [9] diseñado para el control de un robot de un grado de libertad, por lo cual en la presente tesis se aumentó dos grados de libertad en el firmware ya diseñado para de esta forma poder manipular nuestro robot cartesiano.

En la siguiente figura, se muestra un diagrama general del funcionamiento del firmware del sistema:

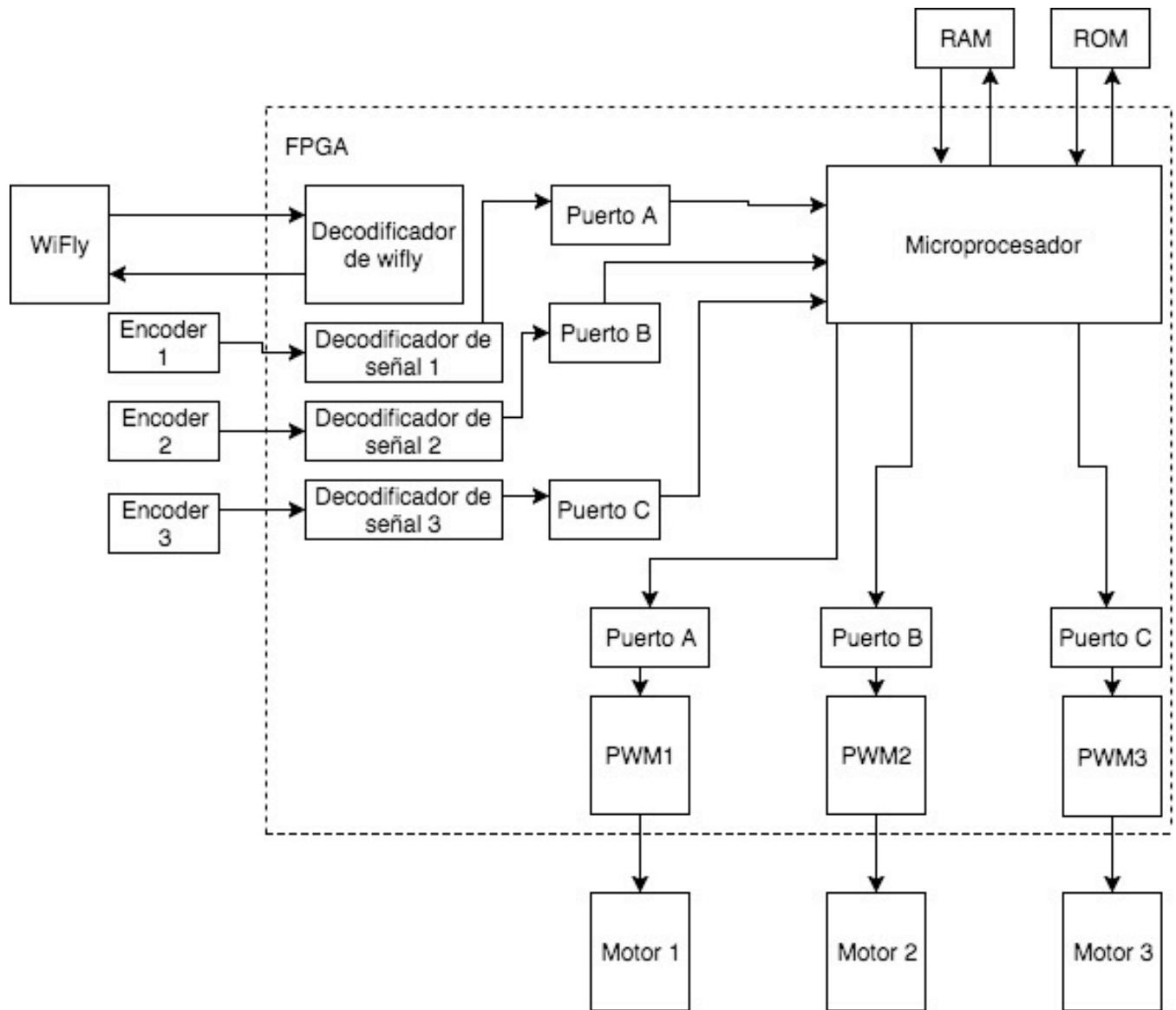


Figura 26: Diagrama general del Firmware.

3.1 Configuración

El firmware es un bloque de instrucciones de máquina para propósitos específicos, grabado en una memoria no volátil (ROM, EEPROM, flash, etc.) de tipo de solo lectura, que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo [10].

Al estar integrado en la electrónica del dispositivo es en parte hardware, pero también software, ya que proporciona lógica y se dispone en algún tipo de lenguaje de programación. Funcionalmente, el firmware es el intermediario (interfaz) entre las órdenes externas que reciben el dispositivo y su electrónica, ya que es el encargado de controlar a ésta última para ejecutar correctamente órdenes externas [11].

El término "firmware" fue acuñado por Ascher Opler en un artículo de la revista Datamation, publicado en 1967 [12]. Originalmente, se refería al código contenido en un área pequeña especializada de memoria RAM (Random Access Memory), que definía e implementaba el conjunto de instrucciones de la computadora. El firmware existía en la frontera entre el hardware y el software, por ello el término de firmware (que significa "software firme, fijo, o sólido"). Posteriormente, el término fue ampliado para incluir cualquier tipo de código, ya fuera en RAM o ROM (Read Only Memory) [10].

AHDL es un lenguaje desarrollado por Altera Corporation para ofrecer una forma conveniente de configurar los dispositivos lógicos que ofrecen. Altera fue una de las primeras compañías en introducir dispositivos lógicos que pueden reconfigurarse mediante electrónica. A estos dispositivos se les conoce como dispositivos lógicos programables (PLDs). A diferencia del VHDL, este lenguaje no está diseñado para usarse como un lenguaje universal para describir cualquier circuito lógico, sino para programar sistemas digitales complejos en PLDs de Altera, en un lenguaje que se perciba en general como más sencillo de aprender, pero que es muy similar al VHDL. También cuenta con características que aprovechan por completo la arquitectura de los dispositivos de Altera. Utilizaremos el software Altera MAX +PLUS II o Quartus II para desarrollar archivos de diseño

en AHDL y VHDL.

El sistema de Altera facilita el desarrollo de circuitos en forma considerable; además, contiene todas las herramientas necesarias para traducir el archivo de diseño de HDL a uno listo para cargarse en un PLD de Altera. También le permite desarrollar bloques de construcción mediante la introducción de diagramas esquemáticos, para después interconectar esos bloques y formar un sistema completo. [13]

El firmware del sistema de adquisición de datos, se instrumentó en AHDL (Altera Hardware Design Language) y se implementó en un FPGA Cyclone III de Altera (EP3C10F256C6) el cual se encuentra montado en la tarjeta hija mencionada anteriormente. La arquitectura de este FPGA consiste de hasta 120,000 elementos lógicos dispuestos verticalmente, 4 Mbits de memoria integrada dispuestos como bloques de 9-Kbit (M9K) y 200 multiplicadores integrados 18x18 bits.

La comunicación entre el firmware y el software se realizó utilizando el protocolo de transmisión UART (Universal Asynchronous Receiver Transmitter o Receptor/Emisor Asíncrono Universal), el cableado para la comunicación es simple debido a la simplicidad del protocolo, en este caso se utilizaron 2 líneas, el software es el encargado de enviar la configuración de la prueba, el firmware recibe esta configuración (Véase Figura 30).

3.2 Sistema de transferencia de datos

Para la programación de nuestro FPGA EP3C10F256C6 de la familia Cyclone III, ocupamos el software Quartus II version 13.0 web edition, realizando toda la programación en lenguaje AHDL. El firmware elaborado, tiene como parte principal el protocolo de comunicación WiFi (lectura y escritura), ya que, utilizamos el FPGA para manejar el bus de datos, por medio de la tarjeta de adquisición de datos, utilizando principalmente sus señales de dirección y datos.

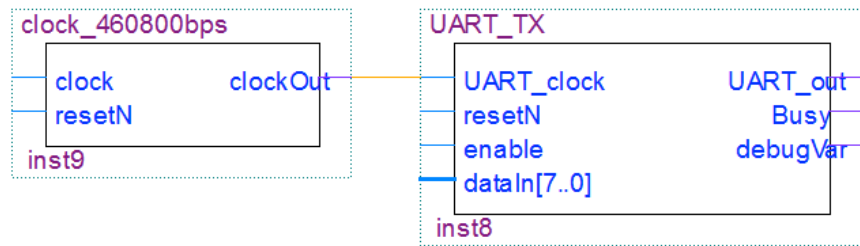


Figura 27: firmware del Protocolo de comunicación WiFi.

El FPGA está conectado sobre una tarjeta hija, acoplado a una tarjeta de adquisición mediante WiFi, la cual se conecta a la conexión inalámbrica de la computadora, esto nos sirve como interfaz entre nuestro sistema físico y el software de control de nuestra Robot cartesiano.

El UART se eligió como protocolo de comunicación de datos, puesto que el módulo RN-XV es el único que tiene habilitado para enviar y recibir datos. Para llevar a cabo la comunicación entre el FPGA y el módulo WiFi, se desarrolló el firmware del protocolo UART. El formato serial de los datos del UART puede variar, dependiendo si se hace uso de bits de paridad para control de error. Para este caso, se usarán 10 bits: 1 bit de inicio + 8 bits de datos + 1 bit de paro. La señal está inactiva a un "1" lógico y activa a un "0" lógico. En la figura 4.17. se puede observar el formato de la transmisión de bits. A la tasa de transferencia se le llama baudrate o tasa de bits por segundo.

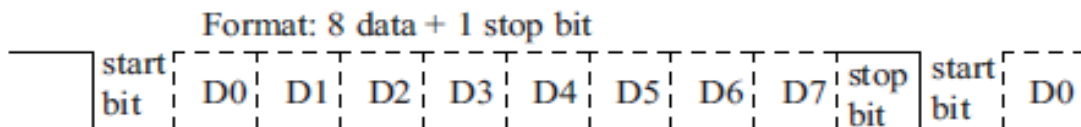


Figura 28: Trasmisión UART.

El Universal Asynchronous Receiver Transmitter o Receptor/Emisor Asíncrono Universal permite establecer una comunicación serial entre dos dispositivos. Puede ser de 2 líneas o 4 si se hace uso del control de flujo. El cableado es mostrado en la figura 4.18.

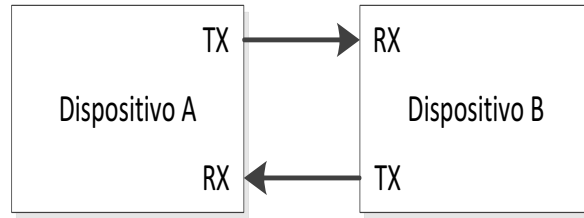


Figura 29: Comunicación UART.

En la nomenclatura del UART, TX es para la transmisión y RX es para la recepción. La principal característica de este medio de transmisión es que no usa una señal de reloj para sincronizar el flujo de datos, caso contrario del bus SPI o el USART. El hecho de no contar una señal de reloj simplifica la operación y facilita la implementación de hardware. Sin embargo, hay un precio a pagar y este es que sacrifica velocidad de transmisión. Puesto que no existe una señal de reloj, ambos dispositivos deben de ser configurados para que funcionen a un flujo de datos predeterminado, de tal forma que se podría decir que existe una sincronización, pero independiente. Dicho lo anterior, se recomienda muestrear los bits al menos 16 veces más rápido que la velocidad de transferencia en uso, esto es con el propósito de tener una tolerancia de lectura (ver figura 33).

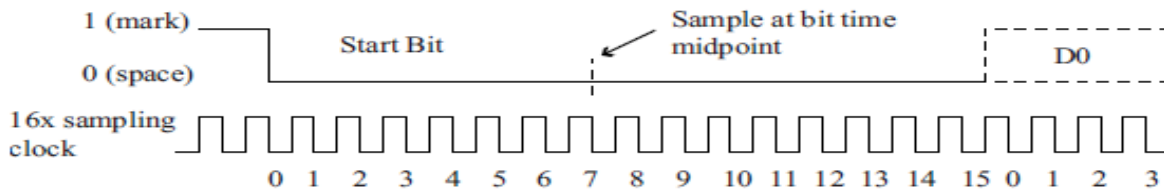


Figura 30: Muestreo de bits.

Tomando en cuenta la frecuencia de muestreo mínima recomendada, se implementó una frecuencia de 100 MHz. El Firmware del UART incluye las funciones de transmisión y recepción de datos serial mente, así como la interface de interpretación de datos. En la figura 4.20. se observa el diagrama a bloques del firmware del UART. [14]

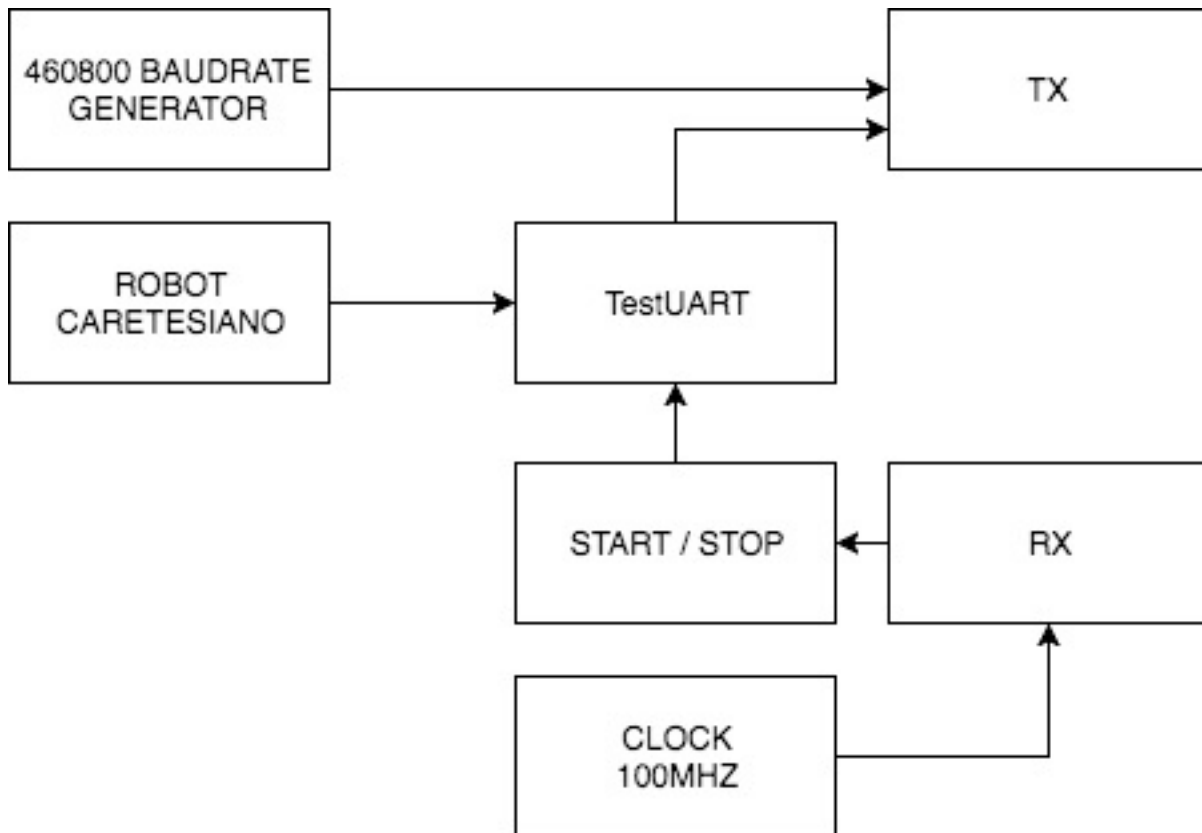


Figura 31: Estructura a bloques del Firmware del UART.

- **Receptor (RX):** Representado por máquinas de estados, en donde se tienen las tareas de detectar el bit de inicio, encontrar el punto intermedio para el muestreo, tomar la muestra, esperar a que termine de transmitirse el bit en cuestión y recibir el bit de paro. Se utiliza un reloj de 100Mhz para el muestreo, con lo que 217 cuentas se requieren para que transcurra una transición de un bit.
- **Transmisor (TX):** Representado por máquinas de estados en donde se tienen las tareas de esperar a que se inicie la transmisión, enviar el bit de inicio "0", esperar a que se termine de transmitir el bit en cuestión, convertir la señal de paralelo a serial y transmitirla y por ultimo enviar el bit el paro. Cuenta con un reloj que genera una tasa de transferencia de 460800 baudios.
- **startStop:** Este bloque revisa el dato recibido desde el UART, si un comando fue recibido, y decide si deshabilita o habilita el módulo UART_TX.
- **testUART :** Controla al transmisor, decide cuándo y que dato debe ser enviado.

En la siguiente figura se observan las máquinas de estados del firmware correspondientes a los módulos de transmisión y recepción.

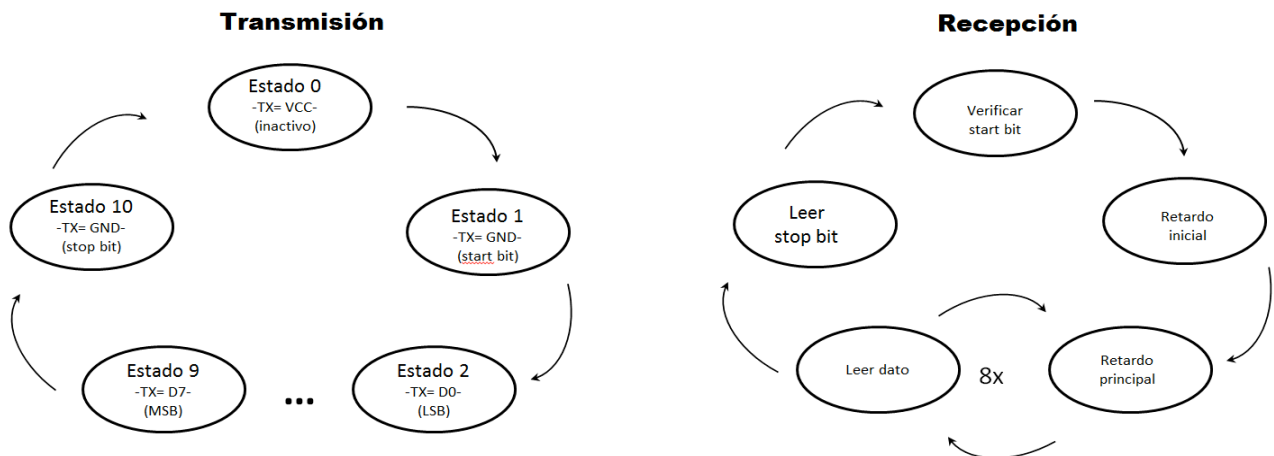


Figura 32: Máquinas de estados de la transmisión y recepción del UART.

El módulo de transmisión cuenta con un reloj configurado a 460800 baudios. Cuenta con una entrada de habilitación y entrada de datos y una salida que indica que el módulo está ocupado. El módulo generado se puede observar en las Figura 4.22.

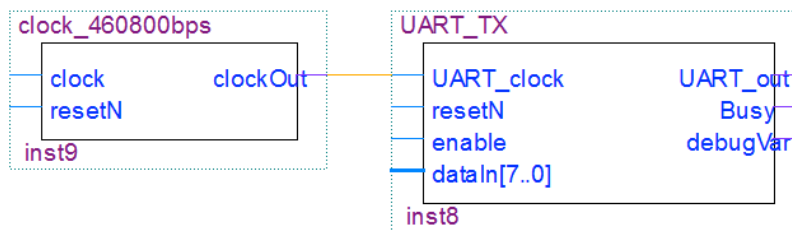


Figura 33: Módulo UART TX.

El módulo de recepción toma los datos a 100Mhz, sincronizándose a los baudios de transmisión mediante retardos. Esto es debido a que, para llevar a cabo una lectura confiable, se leen los datos a la mitad de la transmisión de cada bit. Sus salidas son el byte recibido (Figura 37).

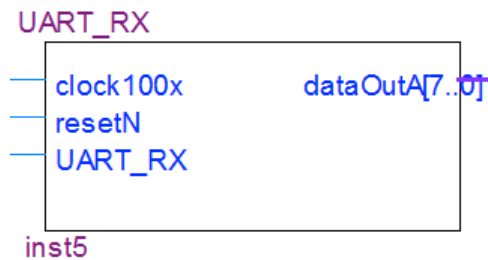


Figura 34: Módulo UART TX.

El módulo *testUART*, tiene como entrada la señal del reloj de 100 MHz, los datos a ser transmitidos, una entrada “enableIn” la cual proviene del módulo *starStop* y habilita el *testUART* para el envío del dato, el cual divide en tres palabras, la señal “UARTrdy” verifica que el módulo este libre con un estado lógico en BAJO (GND).

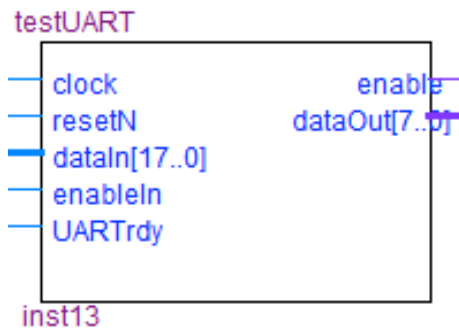


Figura 35: Módulo tesUART

El *starStop* como se había mencionado anteriormente, deshabilita o habilita el módulo UART_TX, mediante la habilitación de la señal de salida enableOut para controlar el módulo testUART.

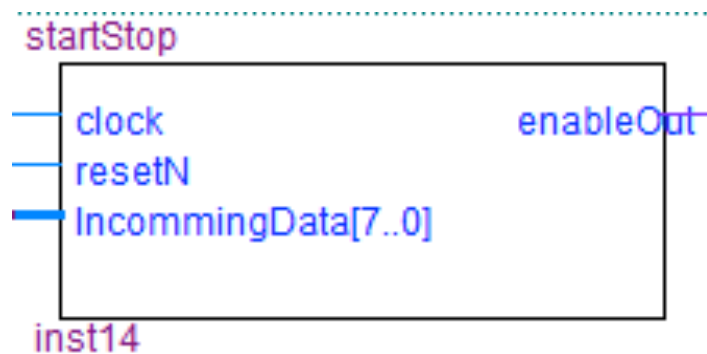


Figura 36: Módulo StartStop.

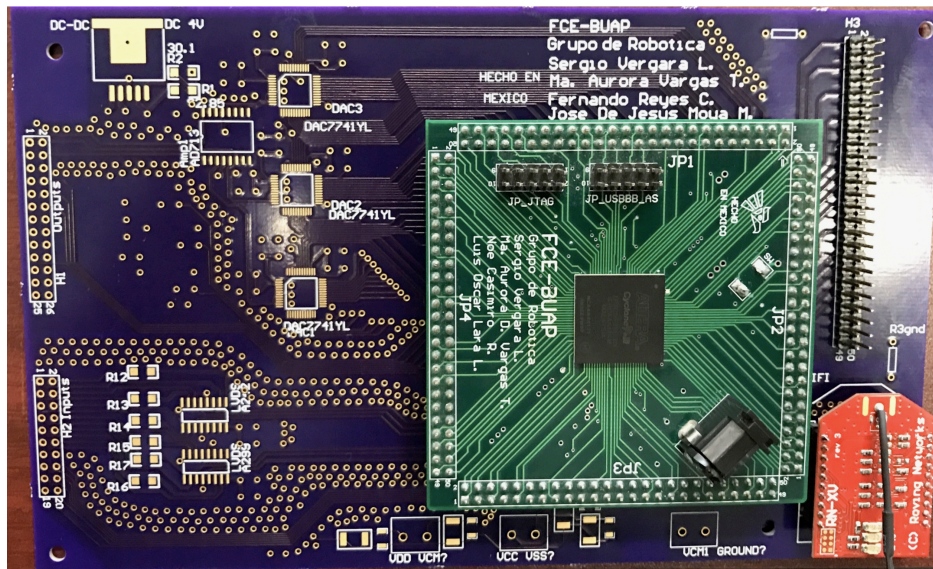


Figura 37: Tarjeta de adquisición de datos.

Esta tarjeta de adquisición es una parte fundamental en la electrónica de nuestro sistema, ya que por medio de sus I/O disponibles se reciben las señales de los encoders incrementales de cuadratura, estas señales se interpretan para proporcionarnos la posición actual del robot cartesiano en cada uno de los ejes. También se reciben las señales de los sensores de fin de carrera, esto con la particularidad de que nuestro robot se detenga en cualquier momento cuando ocurra algún percance.

3.3 Diagrama para obtener la posición a partir del encoder

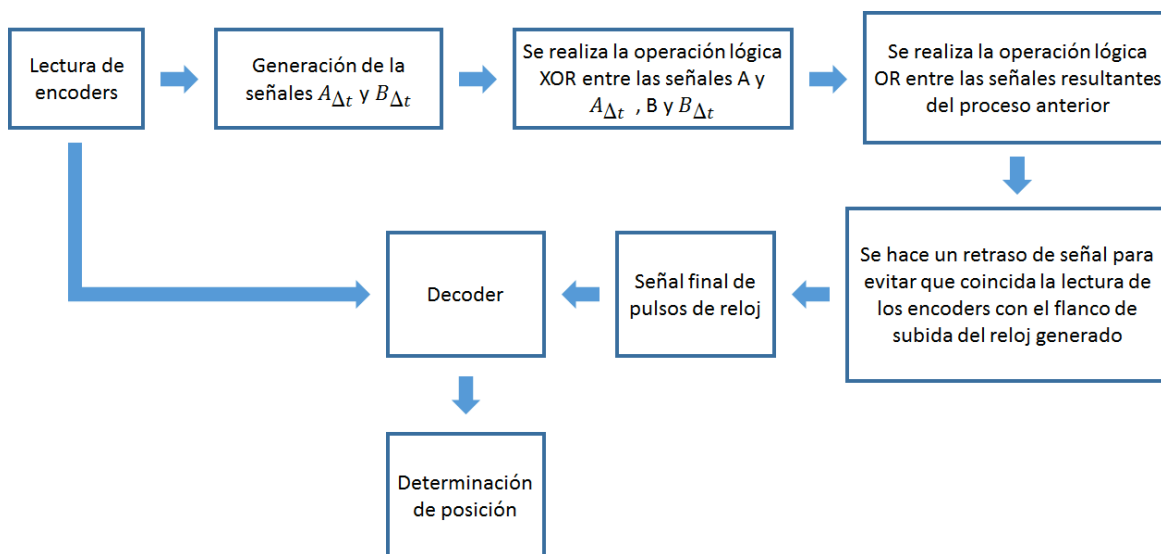


Figura 38: Diagrama a bloques del proceso completo del conteo de pulsos.

En nuestro robot manipulador se implementaron servomotores, los cuales cuentan con encoders para poder terminar la posición de cada robot en cada eje. Por lo que se tuvo que realizar el firmware que nos permitiera poder interpretar las señales que nos proporcionan estos encoders. Esta parte del firmware se llama decoder, el decoder, es una máquina de estados que por medio de contadores nos indica cuantos pulsos hemos recibido de los encoders, además del sentido de giro, ya que en sentido horario el contador incrementara y en sentido inverso decrementara.

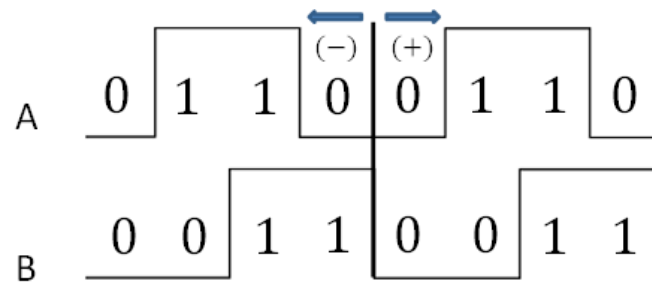


Figura 39: Combinaciones lógicas del encoder.

De la figura 3.3, se puede observar el número de las combinaciones lógicas posibles del encoder incremental de cuadratura. Para ello, es necesario tener una señal de reloj que sea variable y este sincronizada con la velocidad de los motores.

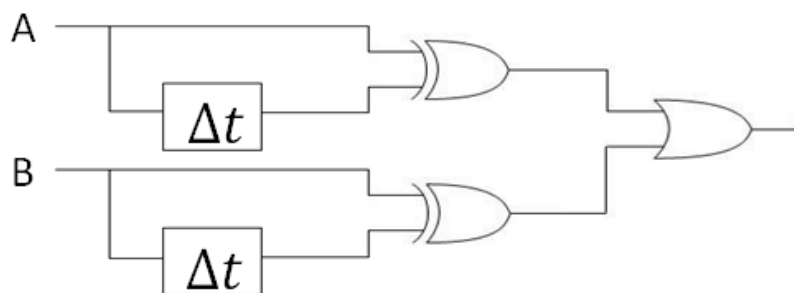


Figura 40: Circuito para la generación de la señal de reloj.

En el circuito anterior se muestra como a partir de las señales del encoder A y B, se genera la señal de reloj para nuestra máquina de estados. A continuación, se realiza la explicación más detallada del proceso.

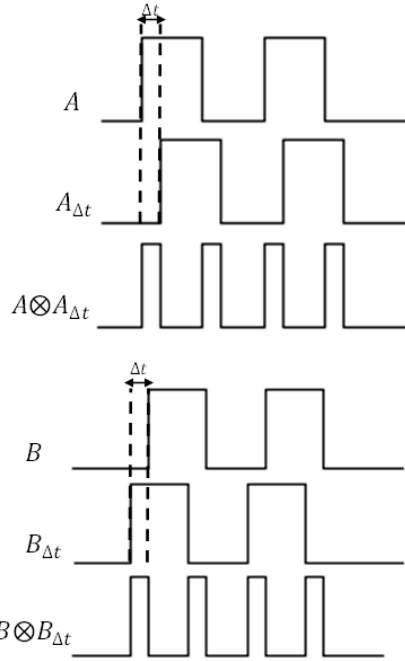


Figura 41: Diagrama de tiempos de las señales A y B.

En el diagrama de tiempos anterior se muestra como se obtienen las señales A, $A_{\Delta t}$, B y $B_{\Delta t}$, con estas señales se realiza la función XOR y se obtendrán $A \otimes A_{\Delta t}$ y $B \otimes B_{\Delta t}$, respectivamente. Después de esto se realiza la función OR para obtener la señal de reloj del decoder.

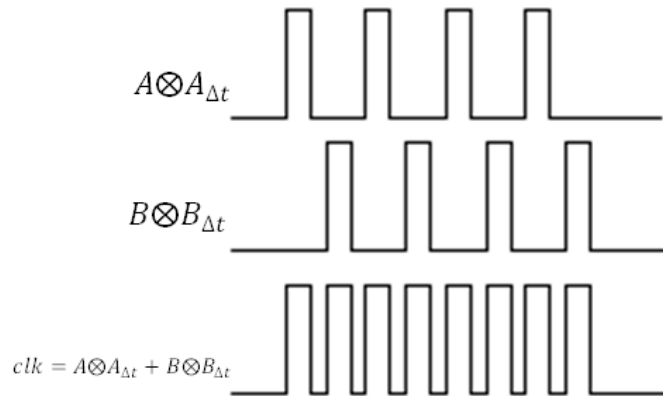


Figura 42: Señal del reloj generada.

La señal de reloj (clk), es creada con las señales A y B del encoder, lo que nos permite muestrearlas en nuestro decoder en todo momento, no importando la velocidad de giro de motor.

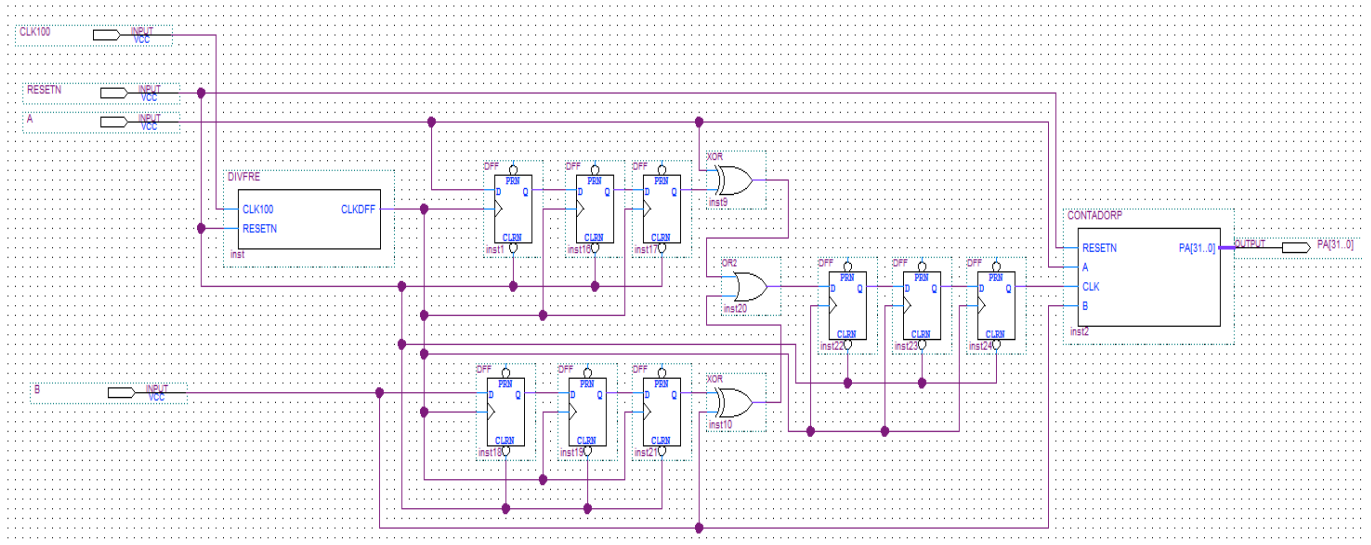


Figura 43: Diagrama a bloques de la generación de pulsos para máquina de estados del encoder.

En el decoder, como se mencionó anteriormente se realizan conteos, para determinar la ubicación de nuestro robot. El siguiente diagrama de flujo expone el funcionamiento del decoder implementado. [15]

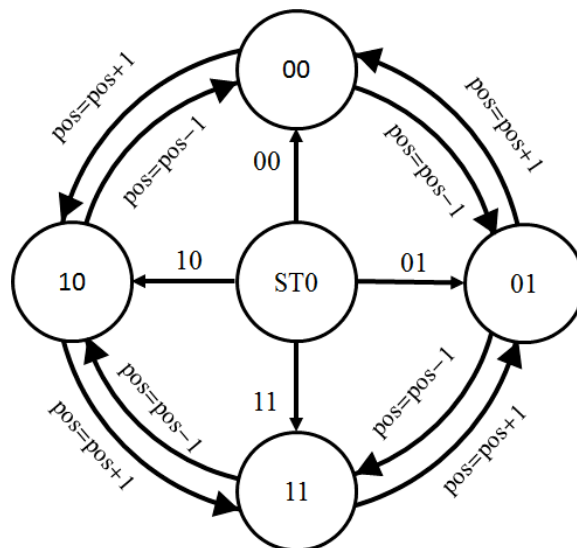


Figura 44: Diagrama de flujo del encoder.

Como se observa en el diagrama anterior se tiene un estado cero, el cual nos permite realizar el conteo de los pulsos del encoder a partir de cualquier combinación lógica de las señales A y B. Al terminar este proceso tendremos la posición actual de nuestro robot manipulador, la cual será enviada de manera inalámbrica por medio del módulo WiFi para ser procesada por el software y utilizada por el algoritmo de control, para determinar el error de posición.

3.4 Generación de PWM

Nuestro firmware cuenta también con una sección en la cual, se genera la señal de PWM que se envía a nuestra etapa de potencia, esta señal tendrá la frecuencia determinada durante la caracterización de los motores, para esto se tiene un bloque en el firmware que servirá como divisor de frecuencia, ya que nuestro FPGA funciona a una frecuencia de 100 MHz, que es la velocidad de transferencia de nuestro modulo wifi.

Cuando ya se ha programado la frecuencia de trabajo de nuestro motor, se pasa a la siguiente etapa donde se genera la señal de PWM. Para poder producir esta señal es necesario que el software de control envíe un dato de 8 bits sin signo por el modulo wifi al FPGA, el valor de este dato dependerá del control de posición que se realiza en el software.

La modulación por ancho de pulsos es una técnica utilizada para regular la velocidad de giro de los motores eléctricos de inducción o asíncronos. Mantiene el par del motor constante y no tiene un desaprovechamiento de la energía eléctrica. Se utiliza tanto en corriente continua como en alterna, como su nombre lo indica, al controlar: un momento alto (encendido) y un momento bajo (apagado).

Para utilizar el modulador de ancho de pulso se necesita trabajar con ciertas frecuencias, ya que cada motor trabaja con diferentes frecuencias, aunque sea el mismo tipo de motor con las mismas características mecánicas no funcionara con el mismo tipo de frecuencia, por eso se realizó la caracterización del motor, en dicha caracterización se encontraron unas frecuencias diferentes de trabajo para nuestros actuadores, las frecuencias encontradas fueron las de 75 Hz, 2.4 KHz y 2.5KHz.

Para poder generar la frecuencias de 75 Hz, 2.5 kHz y 2.4KHz en el FPGA se deben realizar dos divisores de frecuencia, se trabaja con 8 bits esto quiere decir que nuestras frecuencias debe de ser 255 veces más rápida, esto nos genera un valor de 115.51 KHz para la primera frecuencia y 51KHz para la segunda obtenemos el periodo de estas frecuencias y nos dan un valor de $8.65\mu\text{s}$

para la primera frecuencia y $19.60 \mu s$ para el segundo, como ya se conoce el periodo con el que se va a trabajar, ahora utilizamos el valor del reloj del FPGA que es de 100MHz, obtenemos el periodo de esta frecuencia que es de 10 ns y la dividimos por el periodo con el que se va a trabajar y este nos da el valor de 865 para el primer motor y 1960 para el segundo motor estos son los valores que toman los contadores para generar la frecuencia de 75 Hz y 2.5 KHz como se puede observar en las ecuaciones.

$$T = \frac{1}{Frecuencia} = \text{Periodo de la señal} \quad (5.2)$$

$$cuenta\ maxima = \frac{T}{10ns} \quad (5.3)$$

Primer motor

Para el primer motor se obtiene el valor de la cuenta máxima para poder realizar el divisor de frecuencia, para la frecuencia de 453 Hz obtenemos 865.

$$T = \frac{1}{115.51\ Hz} = 8.65\mu s \quad (5.4)$$

$$cuenta\ maxima = \frac{8.65\mu s}{10ns} = 865\ cuenta\ maxima \quad (5.5)$$

Segundo motor

ara el segundo motor se obtiene el valor de la cuenta máxima para poder realizar el divisor de frecuencia, para la frecuencia de 200 Hz obtenemos 1960.

$$T = \frac{1}{51KHz} = 19.60\ \mu s \quad (5.6)$$

$$cuenta\ maxima = \frac{19.60\ \mu s}{10ns} = 1960\ cuenta\ maxima \quad (5.7)$$

A continuación, se muestra en la figura 5.13 los diagramas de flujo del divisor de frecuencia en el inciso se muestra el diagrama de flujo de la frecuencia de 75 Hz .

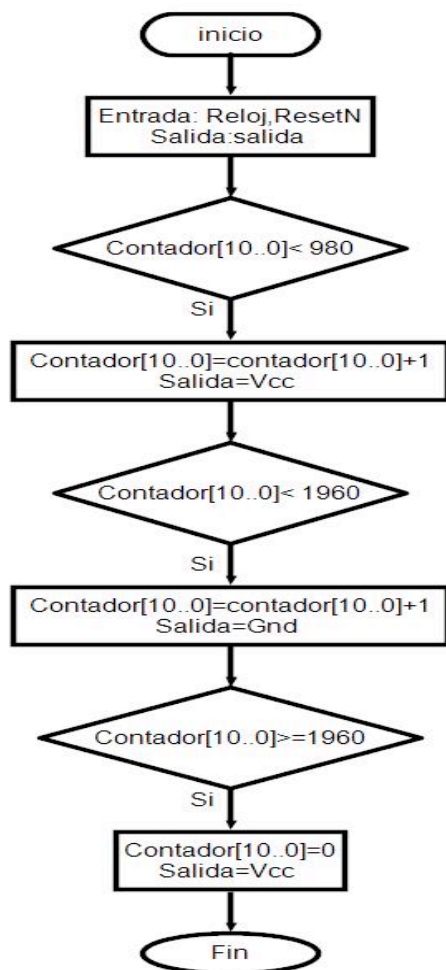


Figura 45: Diagrama de flujo del contador.

Para poder controlar el tercer motor se tuvo que realizar un divisor de frecuencia para poder disminuir la velocidad del FPGA y poder mandar los pulsos del motor, a continuación, se explica cómo se realizó el divisor de frecuencia. Para poder saber el periodo para la generación de 100 Hz, sabemos que trabajamos con 8 bits, esto quiere decir que nuestra frecuencia debe de ser 255 veces más rápida, esto nos genera un valor de 2.5 KHz esta frecuencia es la que se necesita, obtenemos el periodo de esta frecuencia y nos da un valor de 39.21 μ s, como ya se conoce el periodo con el que se va a trabajar, ahora utilizamos el valor del reloj del FPGA que para este caso es de 100MHz, obtenemos el periodo de esta frecuencia que es de 10 ns y la dividimos por el periodo con el que se va a trabajar y este nos da el valor de 3921 y este el valor que tomara

nuestro contador para generar la frecuencia de 75 Hz como se puede observar en las ecuaciones, en la figura 5.16 se muestra el diagrama de flujo del divisor de frecuencia a 75 Hz.

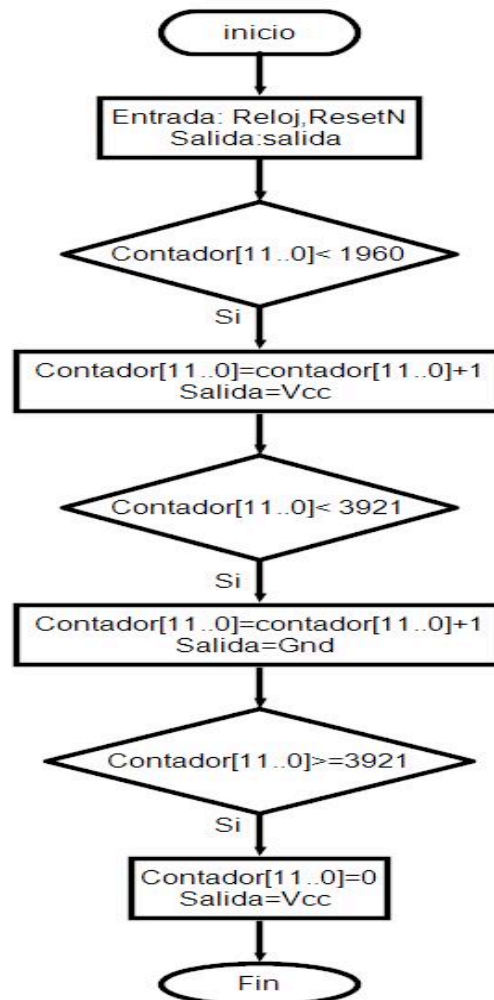


Figura 46: Diagrama de flujo de divisor de frecuencia a 100Hz.

Sabiendo que

$$T = \frac{1}{25.5KHz} = 39.21\mu s \quad (5.8)$$

$$cuenta\ maxima = \frac{39.21\mu s}{10ns} = 3921\ cuenta\ maxima \quad (5.9)$$

Una vez obtenida nuestra frecuencia de 100 Hz obtenemos nuestro divisor de frecuencia con el cual vamos a poder enviar los pulsos de las bobinas del motor. En la Figura 5.14 podemos observar el bloque PWM0 que manda las salidas PWM_A Y PWM_B al driver diseñado que posiciona al sistema en el lugar deseado. La entrada torquein[15..0] es la que proviene del bloque de control de la salida resultado[31..0] la cual se convierte en un bus de 16 bits en el bloque Conversión de resultado RESCONV. Dentro de este bloque se selecciona la dirección del giro

del motor según la posición deseada que se requiera, también se observa en la figura 5.15 el diagrama de flujo del PWM.

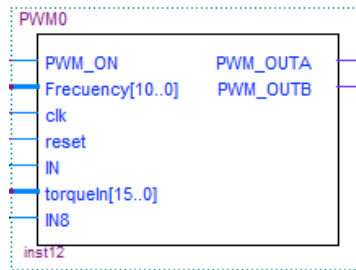


Figura 47: Bloque de PWM

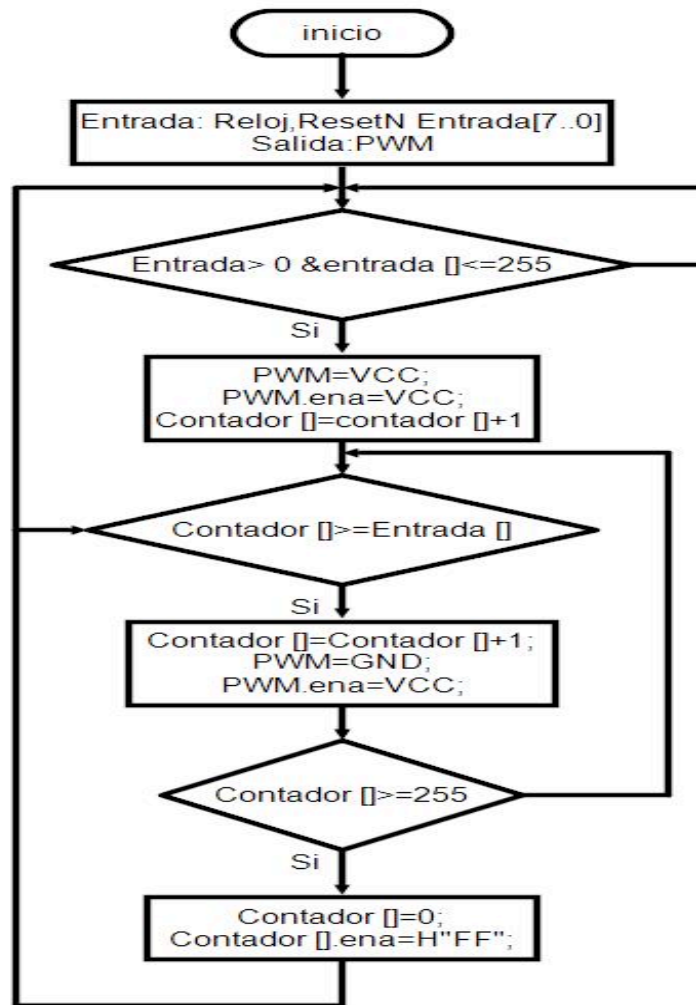


Figura 48: Diagrama de flujo del PWM.

3.5 Microprocesador

Nuestro firmware cuenta con un microprocesador [9], en el cual se realiza el algoritmo de control para manipular nuestro robot. A continuación, se muestra un diagrama a bloques del firmware empleado:

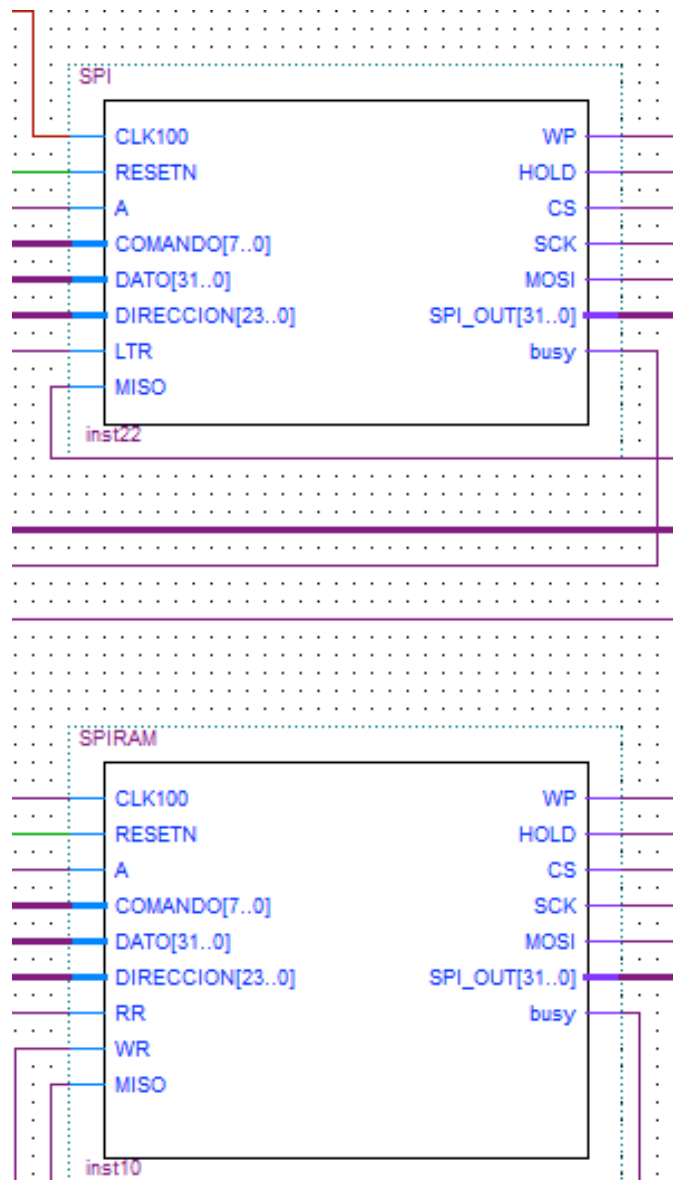


Figura 49: Bloque del Microprocesador.

Se cuenta con una tabla de instrucciones para facilitar su uso:

INSTRUCCIONES								
INST.	COD. INST.	REG. F1	REG F2	REG. D	SHAMT	FUNC.	FORMATO	DESCRIPCIÓN
MOV.S	110010	XXXXX	00000	XXXXX	00000	000000	I	X->Y
MOVC.S	110010	00000	00000	XXXXX	00000	000001	I	MC->X
RIO.S	110010	XXXXX	00000	XXXXX	00000	000010	I	PORTIN->X
WIO.S	110010	XXXXX	00000	XXXXX	00000	000011	I	X->PORTOUT
C.EQ.S	000100	XXXXX	XXXXX	XXXXX	XXXXX	XXXXX	I	IF (X==Y) SI PC=PC+4+L, NO PC=PC+4.
C.LE.S	000101	XXXXX	XXXXX	XXXXX	XXXXX	XXXXX	I	IF (X<=Y) SI PC=PC+4+L, NO PC=PC+4.
C.LT.S	000110	XXXXX	XXXXX	XXXXX	XXXXX	XXXXX	I	IF (X<Y) SI PC=PC+4+L, NO PC=PC+4.
ADD.S	010001	XXXXX	XXXXX	XXXXX	00000	000000	R	X=X+Y
SUB.S	010001	XXXXX	XXXXX	XXXXX	00000	000001	R	X=X-Y
MUL.S	010001	XXXXX	XXXXX	XXXXX	00000	000010	R	X=X*Y
DIV.S	010001	XXXXX	XXXXX	XXXXX	00000	000011	R	X=X/Y
JUMP	000010	00XXX	XXXXX	XXXXX	XXXXX	XXXXXX		

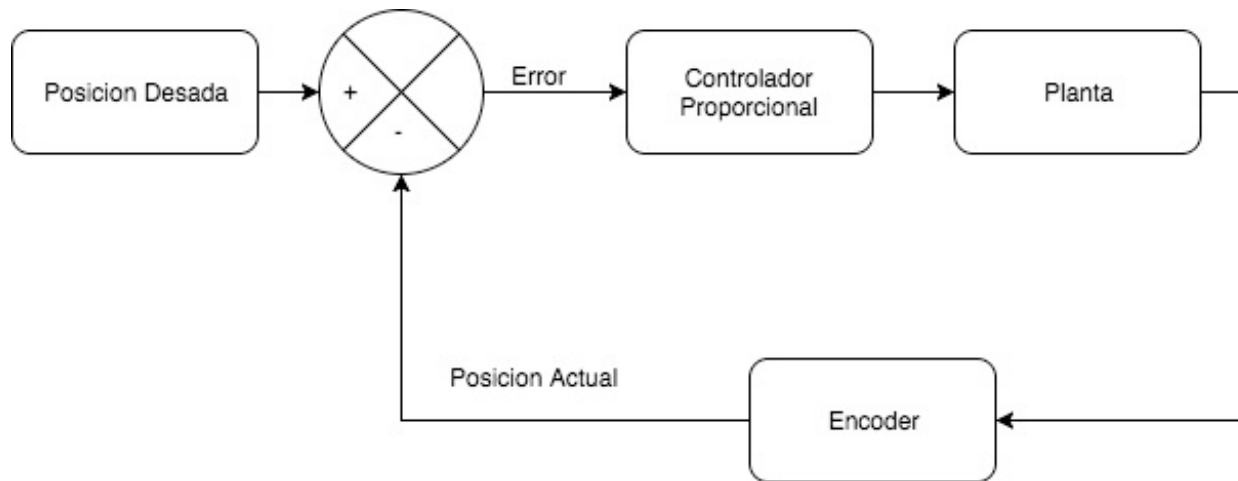
Tabla 1: Instrucciones del microprocesador.

Gracias a la tabla se puede desarrollar el control necesario para manipular nuestro robot, cabe mencionar que esto es de mucha ayuda ya que si el usuario desearía emplear un controlador diferente bastaría con saber interpretar dichas instrucciones para poder hacer uso de ellas y de esta forma implementar el controlador que se dese usar, en este caso se empleó un controlador Kp, a continuación se muestra el código realizado con las instrucciones el cual es un archivo con extensión .txt, el cual posteriormente se cargara en el compilador y finalmente pasara a cargarse a nuestra memoria ROM.

3.5.1 Firmware para la ley de control generada para el robot

Los controladores son los instrumentos diseñados para detectar y corregir los errores producidos al comparar y computar el valor de referencia, con el valor medido del parámetro más importante a controlar en un proceso. Son variables las características de los controladores debido a sus tipos, ya que en un controlador de acción proporcional la señal de accionamiento es proporcional a la señal de error del sistema, es decir en este tipo de controlador, si la señal de error es cero, también lo será la salida. Por otra parte, en los controladores de acción integral el valor de la acción de control es proporcional a la integral de la señal de error. Por lo que este tipo de control de acción varía en función de la desviación de la salida y del tiempo en el que se mantiene esta desviación. No obstante, el controlador proporcional derivativo se opone a las desviaciones de la señal de entrada, con una respuesta que es proporcional a la rapidez con que se producen estas, cada tipo de controlador posee su modelo matemático en cuanto a la función de transferencia.

Para este caso se empleó un control proporcional para manipular nuestro robot, la cual se explica a continuación:



En el firmware se realiza la ley de control para nuestro robot, la cual del diagrama anterior se puede apreciar que se lee la posición deseada esta se resta de la posición actual, la cual con la ayuda del encoder tenemos este dato, posteriormente al aplicar la ganancia proporcional que deseamos en el software de nuestra aplicación se obtiene la ley de control tipo proporcional, la cual es ganancia proporcional*(posición deseada menos la posición actual), la cual esta ley de control es aplicada a nuestro robot para poder controlar este.

A continuación, se tiene la serie de instrucciones empleadas para realizar el control de nuestro robot:

```
RIO.S PT11 R0
RIO.S PT2 R1
RIO.S PT10 R2
SUB.S R0 R1 R3
MUL.S R2 R3 R5
WIO.S R5 PT0
RIO.S PT8 R0
RIO.S PT1 R1
RIO.S PT7 R2
SUB.S R0 R1 R3
MUL.S R2 R3 R6
WIO.S R6 PT1
```

RIO.S PT5 R0
RIO.S PT0 R1
RIO.S PT4 R2
SUB.S R0 R1 R3
MUL.S R2 R3 R5
WIO.S R5 PT2
DELAY
JUMP 0

Del código de instrucciones anteriores se tiene la interpretación de lo que hacen:

RIO.S PT11 R0	→	Se lee el puerto once, el cual tiene como asignación dentro del código de quartus la posición actual y este dato se almacena en el registro cero de nuestro micro-controlador.
RIO.S PT2 R1	→	Se lee el puerto dos, el cual tiene como asignación dentro del código de quartus la posición deseada y este dato se almacena en el registro uno de nuestro micro-controlador.
RIO.S PT10 R2	→	Se lee el puerto diez, el cual tiene como asignación dentro del código de quartus la posición deseada y este dato se almacena en el registro dos de nuestro micro-controlador.
SUB.S R0 R1 R3	→	Se realiza la resta de la posición deseada menos la posición actual, para obtener el error de posición, este dato se almacena en el registro tres de nuestro micro-controlador.
MUL.S R2 R3 R5	→	Se realiza la multiplicación del error de posición por la constante de ganancia proporcional, aquí tenemos nuestra ley de control este dato se almacena en el registro cinco de nuestro micro-controlador.
WIO.S R5 PT0	→	Se manda a escribir al puerto cero, el cual corresponde a nuestro primer motor de nuestro robot, el cual manipulará este con la ley de control correspondiente.

hasta esta parte se explica el funcionamiento de las instrucciones empleadas en el firmware del control proporcional en el cual se explica lo que hace cada instrucción haciendo hincapié que el firmware explicado es para el control de un grado de libertad.

3.6 Compilador

Haciendo mención al [16] del compilador empleado para cargar el archivo generado en formato .txt y de esta forma, haciendo uso de su programa para que posteriormente pueda interpretarlo nuestra memoria ROM, en la cual es donde se carga el archivo de control para el robot.

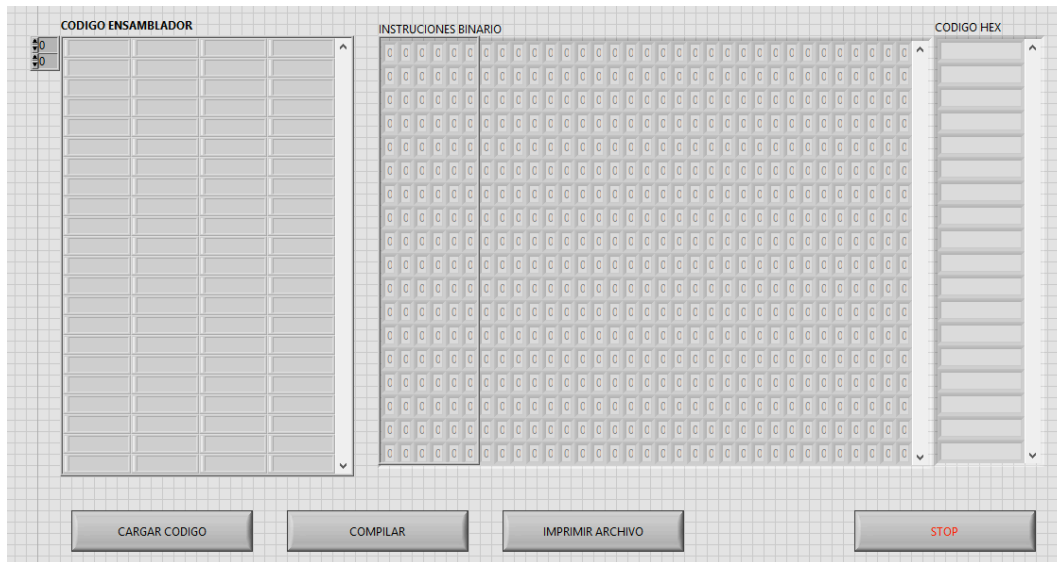


Figura 50: Compilador.

Con la ayuda de este compilador se genera en un bloc de notas el controlador que deseemos emplear (tipo ensamblador) para poder manipular nuestro robot, se guarda este archivo con extensión .txt, posteriormente se carga ese código en el programa se compila y finalmente se imprime el archivo, el cual este último podrá escribirse haciendo uso de la interfaz principal, se escribe y se graba en la memoria ROM en la cual podremos finalmente controlar nuestro robot.

3.7 Conclusión

En este capítulo se menciona la importancia que tiene el protocolo de comunicación WiFi, para nuestra tesis ya que, mediante este se realiza la interacción entre el software de control y el firmware. Además, se indica como es el proceso de decodificación de la señal de los encoders, sensores que nos sirven para ubicar la posición de nuestro manipulador en el espacio de trabajo, finalmente se explica brevemente cómo funciona el firmware de control de movimiento, el cual, genera la señal de PWM a partir del dato enviado por el software de control de posición, con la frecuencia ideal para cada actuador de nuestro robot cartesiano.

CAPITULO IV

APLICACIÓN

En este capítulo se realiza la prueba del funcionamiento de la tarjeta de control de adquisición de datos desarrollada en la presente tesis.

Haciendo mención [17], en la cual se hace uso de su robot cartesiano, para lo cual la etapa de potencia y lecturas de señales del encoder, no se contaba con ello y se realizó la fabricación de estas tarjetas para poder usar el robot cartesiano ya hecho en la MCEA.

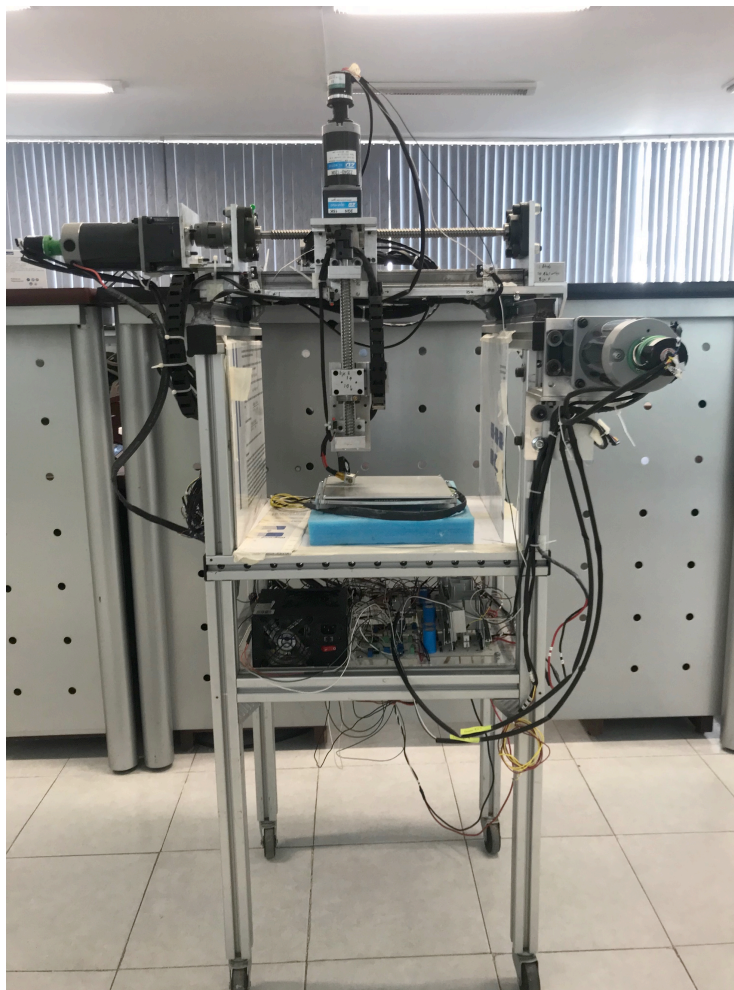


Figura 51: Robot CNC.

A continuación, se presenta la estructura mecánica de nuestro robot CNC, donde se muestra como se realizó el acoplamiento de los encoders con los motor-reductores, y de los motor-reductores al tornillo embalado.

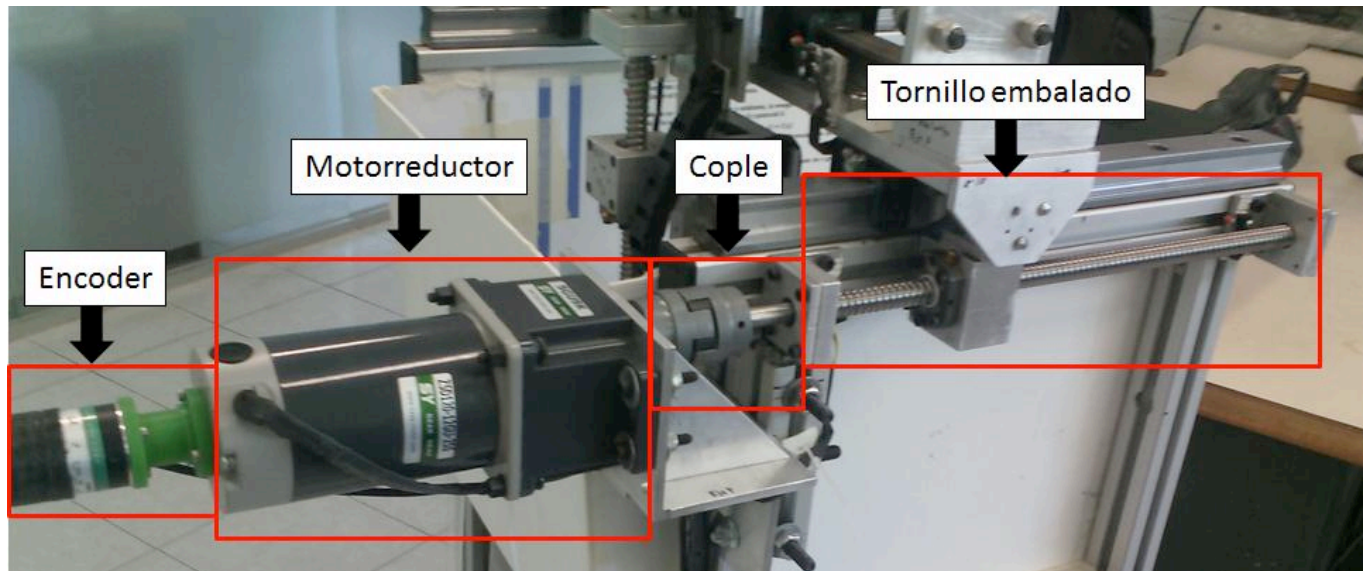


Figura 52: Eje X.

4.1 Riel de alta precisión.

En nuestro manipulador se utilizaron rieles y carros lineales de alta precisión, con el fin de obtener el mejor desplazamiento lineal posible con una baja fricción.

Algunas de las características de estos elementos mecánicos son:

- Alta precisión en posicionamiento con alta repetividad.
- Baja resistencia a la fricción. Esto ayuda a mantener su alta precisión por mayor tiempo.
- Alta rigidez, debido a su diseño de carga en apoyo sobre 4 puntos de la guía.

4.2 Tornillo embalado

Como se puede observar en las imágenes se ocuparon tornillos embalados para transformar el desplazamiento giratorio a lineal. Los tornillos embalados son capaces de desplazar grandes cargas, aunque para nuestra aplicación no es necesario, se pueden utilizar para otra aplicación más robusta, la cual requiera de esta cualidad.

Para nuestro robot manipulador fue importante utilizar este tipo tornillos, ya que también cuentan con la característica de reemplazar la fricción deslizante por una fricción de rodadura, con lo cual se incrementa la eficiencia del tornillo de 30% a un 90% ayudando a que el esfuerzo en los componentes motrices se vea disminuido significativamente y haciéndolos más durables.

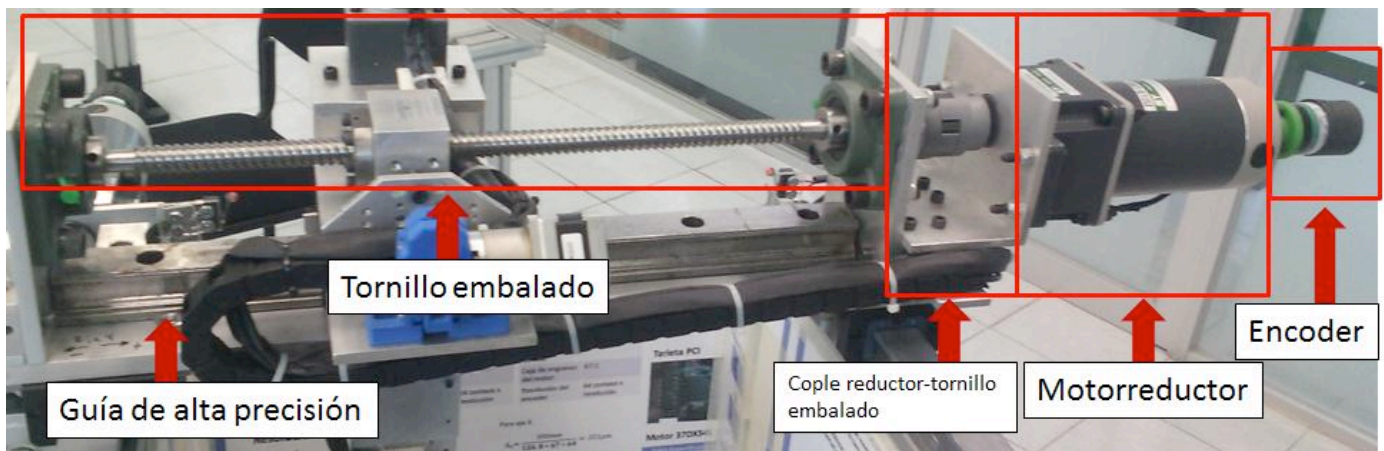


Figura 53: Eje Y

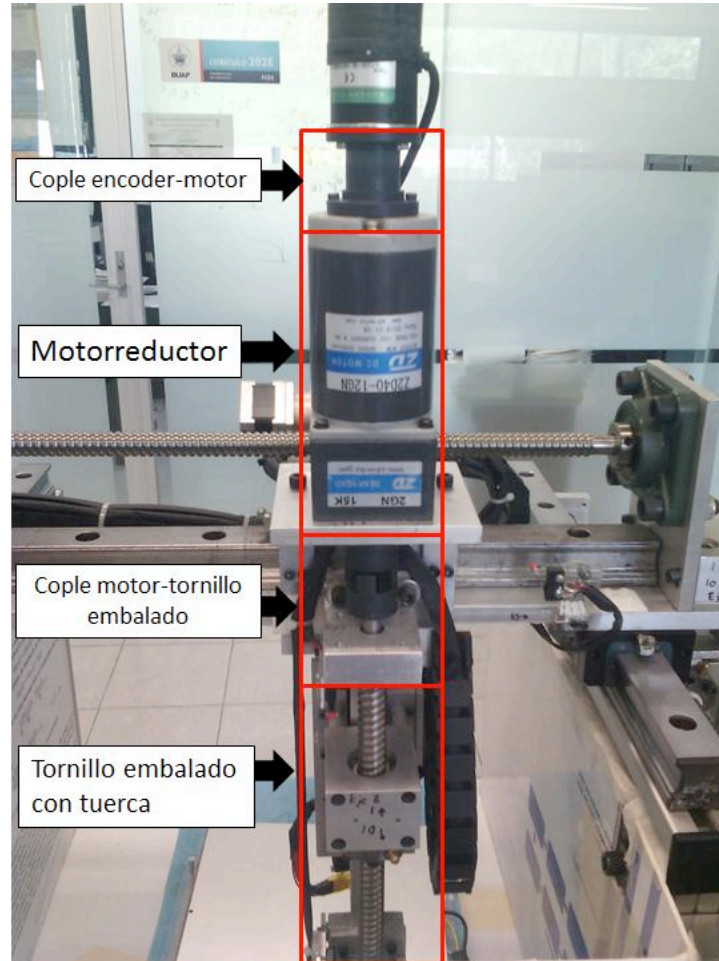


Figura 54: Eje Z.

De las imágenes anteriores podemos ver cómo están ensamblados mecánicamente los ejes del robot manipulador, que ya se tenía anteriormente, pero se reparó cada etapa.

Para poder realizar el lazo cerrado del nuestro sistema, fue necesario el uso de múltiples herramientas entre las cuales LABVIEW tiene suma importancia para poder realizar el control de este, de la cual se usó el software de LABVIEW, que se usó en la de manera conjunta con nuestro firmware, la utilización de LABVIEW además de agregar procesamiento en tiempo real, el software determina los parámetros de entrada en la ejecución, es por eso que al ser tan dependiente del hardware puede ser tan rápido ya que no se tiene que implementar interfaces complicadas que retardan la ejecución como en las computadoras personales. [18]

4.3 Descripción del robot.

Actualmente, se cuenta con un robot CNC (Control Numérico por Computadora), el cual tiene grandes prestaciones por lo que se pensó implementar en conjunto con la tarjeta de adquisición de datos y control, para satisfacer la necesidad de realizar prototipos rápidos, tanto para la investigación como para el desarrollo de nuevos proyectos.

<i>Características del Robot CNC</i>	
Descripción	Requerimiento
Numero de grados de libertad	3(ejes X,Y y Z).
Desplazamiento en eje X e Y	<ul style="list-style-type: none"> ➤ Tornillo embalado de 12mm. ➤ Guía de alta precisión de 32mm. ➤ Bloque para guía de 32mm.
Eje X	$\Delta x = 0.500\mu m$
Eje Y	$\Delta y = 0.500\mu m$
Desplazamiento en eje Z	<ul style="list-style-type: none"> ➤ Tornillo embalado de 12mm. ➤ Guía de alta precisión de 15mm. ➤ Bloque para guía de 15mm.
Eje Z	$\Delta z = 0.500\mu m.$
Área de trabajo	210mmX210mm
Alimentación	120VCA y 12VCD.

Tabla 2: Requerimientos del robot.

Todo lo anterior debe estar controlado por el usuario, por lo que se propuso utilizar un FPGA (del inglés Field Programmable Gate Array) para realizar este control, utilizando el protocolo de comunicación wifi de una computadora como interfaz, además se requiere de un programa para modificar y visualizar los parámetros del sistema, en este caso utilizamos LabView para hacer el software de control. [19]

Para nuestro robot manipulador fue importante utilizar este tipo tornillos, ya que también cuentan con la característica de reemplazar la fricción deslizante por una fricción de rodadura, con lo cual se incrementa la eficiencia del tornillo de 30% a un 90% ayudando a que el esfuerzo en los componentes motrices se vea disminuido significativamente y haciéndolos más durables.

Para la selección de estos motores reductores se tuvieron tres consideraciones principales, el voltaje el cual opera a 12v, las revoluciones por minuto, más de 300 rpm y por último y la más importante el torque que se encontrara en un rango de 3 - 4Nm, esto para los ejes X e Y.

Con respecto al motor Z5D120-12 tenemos las siguientes características:

- Voltaje 12 v
- Potencia 120 w
- 3100 revoluciones por minuto. 2.2kg de peso.



Figura 55: Motor Z5D120-12.

MODELO	VOLTAJE	POTENCIA	PARÁMETROS SIN CARGA		PARÁMETROS CON CARGA		PESO DEL MOTOR
			VELOCIDAD	CORRIENTE	VELOCIDAD	CORRIENTE	
Z5D120-12	12V	120W	3100rpm	2A	2500rpm	15A	2.2kg.

Tabla 3: Características del motor.



Figura 56: Reductor 5GU9K.

TAMAÑO	TIPO DE EJE	RELACION	DIAMETRO DEL EJE DE SALIDA
3.54 pulgadas	Eje paralelo a la cabeza del reductor.	9:1	15mm

Tabla 4: Características del reductor.

Para el eje Z se empleó otro tipo de motor, el modelo Z4D40-12GN-30S, que cuenta con las siguientes características:

MODELO	VOLTAJE	POTENCIA	PARÁMETROS SIN CARGA		PARÁMETROS CON CARGA		PESO DEL MOTOR
			VELOCIDAD	CORRIENTE	VELOCIDAD	CORRIENTE	
Z4D40-12	12V	40W	3100rpm	2A	2800rpm	5.2A	1.8kg.

Tabla 5: Características del motor Z4D40-12.



Figura 57: Motor Z4D40-12.

Una vez adquiridos los motor-reductores del robot CNC se procedió a caracterizarlos para determinar su curva característica, para poderlos utilizar como motores de transmisión directa

(Direct drive), es decir, que el par que solicitemos en la entrada será el par de salida, mientras se encuentre en su rango de torque. Para esto utilizamos el freno prony.

4.4 Caracterización de los motores

El proceso consta de los siguientes pasos:

- I. Determinar el torque máximo del motor variando el voltaje de alimentación, hasta el voltaje recomendado de trabajo.
- II. Una vez encontrado el torque máximo, se buscará la mitad de ese torque máximo alimentado al motor con el voltaje del torque máximo, pero con una señal de PWM (pulse-width modulation) al 50% del ciclo de trabajo. Para conseguir esto, es necesario ir variando la frecuencia de la señal de PWM, desde los 50Hz hasta la frecuencia que determine el usuario, es decir, no tiene frecuencia máxima.
- III. Se seleccionan las frecuencias o la frecuencia que nos proporcione el valor más cercano a la mitad del torque máximo.
- IV. Con cada una de las frecuencias seleccionadas se tendrá que ir variando el ciclo de trabajo desde el 10% hasta el 90% del ciclo de trabajo, alimentando al motor con el voltaje del torque máximo. Cada vez que se varíe el valor del ciclo de trabajo se tendrá que determinar el torque y graficar la respuesta del motor.
- V. Se comparan las gráficas obtenidas y se determina la ecuación de la curva característica. Seleccionando la frecuencia que tenga nos ofrezca la respuesta más cercana a la lineal.

De esta manera, un motor convencional de DC podrá ser utilizado emulando un motor de transmisión directa. Aplicando el proceso anterior a estos dos motores se obtuvieron los siguientes resultados. Para el motor del eje X, se seleccionó una frecuencia de 2.45 KHz. Teniendo los siguientes resultados:

%Ciclo de trabajo del PWM	Torque (Nm)
0	0
10	2.8616
12	3.9494
14	4.3708
16	5.2528
18	5.9976
20	6.4288
22	7.595
24	8.1144
28	9.555
34	10.8192
36	12.348

Tabla 6: Datos del motor eje X.

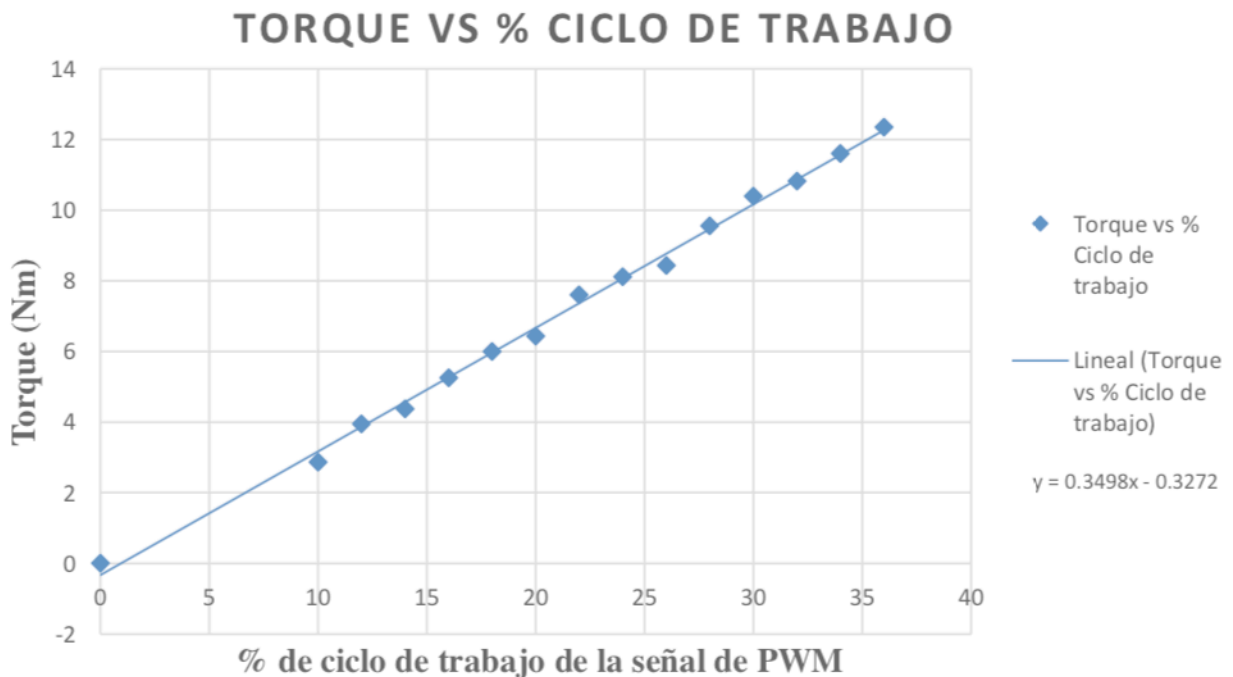


Figura 58: Grafica de resultado del motor X.

Para el motor del eje Y, se seleccionó una frecuencia de 2.5 KHz. Teniendo los siguientes resultados:

%Ciclo de trabajo del PWM	Torque (Nm)
0	0
10	3.3124
12	4.2924
14	5.1058
16	5.8016
18	6.3406
20	6.8208
22	7.68208
24	8.477
26	9.31
28	9.996
31	11.0152
33	11.7992
36	12.348

Tabla 7: Datos del motor eje Y.

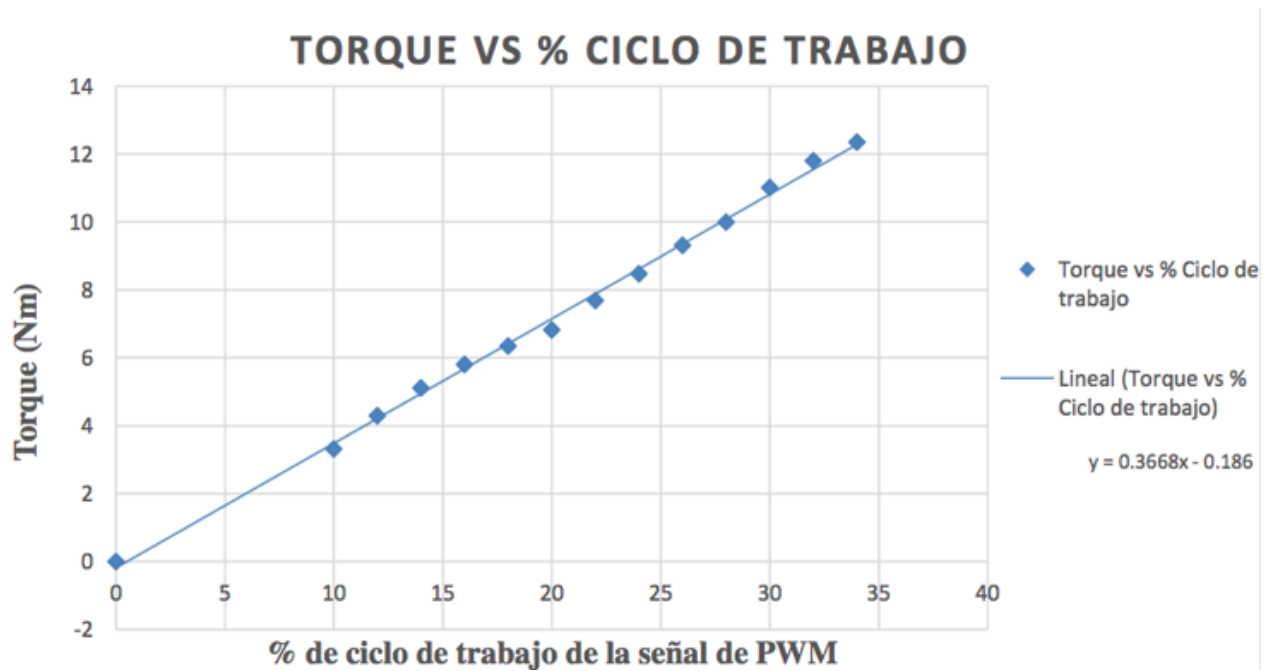


Figura 59: Gráfica de resultados del motor Y.

De forma análoga a los motores de los ejes X e Y, se realizó el proceso de caracterización ocupando una frecuencia de trabajo de 75Hz, teniendo así los siguientes resultados:

%Ciclo de trabajo del PWM	Torque (Nm)
0	0
10	0.0616077
20	0.3435096
30	0.6571488
40	0.840105
50	1.250823
60	1.437513
70	1.8930366
80	2.072259
90	2.5688544

Tabla 8: Datos del motor Z.

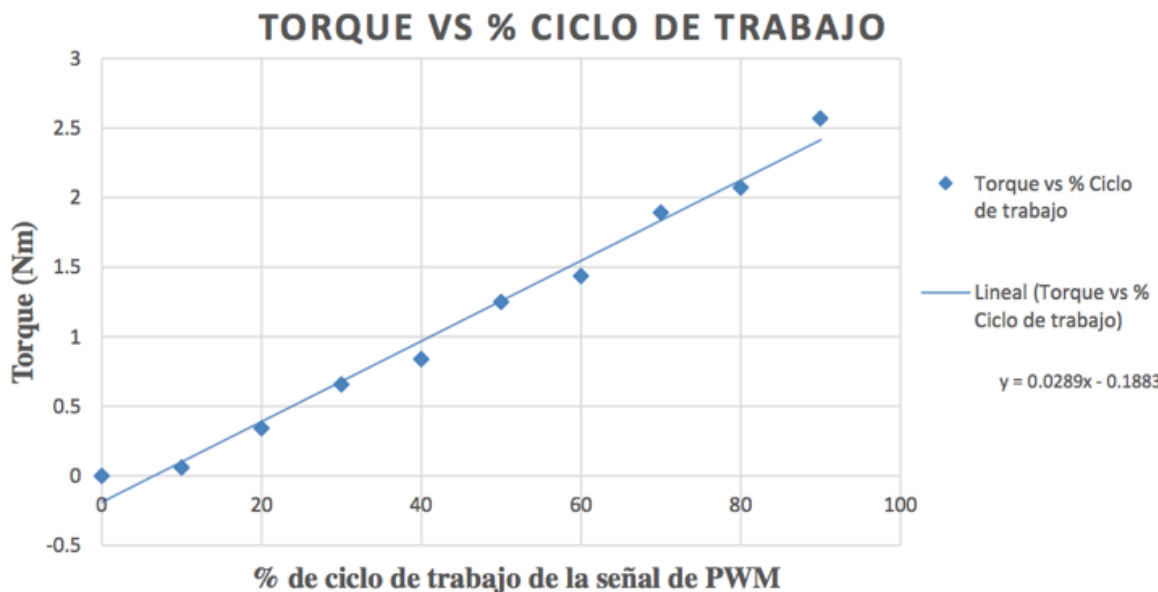


Figura 60: Gráfica de resultados del motor Z.

Con lo anterior se justifica que los motores DC, tendrán un comportamiento lineal o muy cercano al lineal, si se cumplen las condiciones eléctricas y de programación necesarias.

Para poder ocupar los motores DC como servomotores fue necesario acoplar un encoder giratorio en la parte posterior de los motores y así poder determinar la posición actual de nuestro actuador. Los encoders utilizados fueron los GRTS06 que cuentan con las siguientes características:



Figura 61: Encoder giratorio GTS06-VE-RA 1000.

- 1000 pulsos/revolución.
- Alimentación de 5-24 VDC.
- Eje de 6mm de diámetro, 12mm de largo.
- Tamaño: 38mm de diámetro y 35.5 de largo.
- Salida: AB de cuadratura.
- Máxima velocidad mecánica: 5000 rev/min.
- Salida NPN tipo colector abierto (necesita una resistencia Pull-up, ya que, si es conectado a VCC (Voltaje de corriente continua), las salidas se quemarían.)

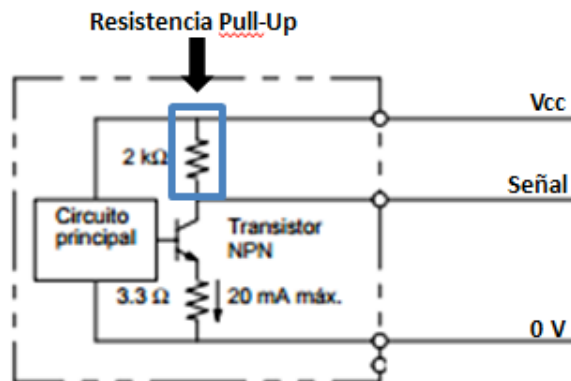


Figura 62: Ejemplo de encoder con salida NPN tipo colector abierto.

A continuación, se muestra un diagrama general de la aplicación:

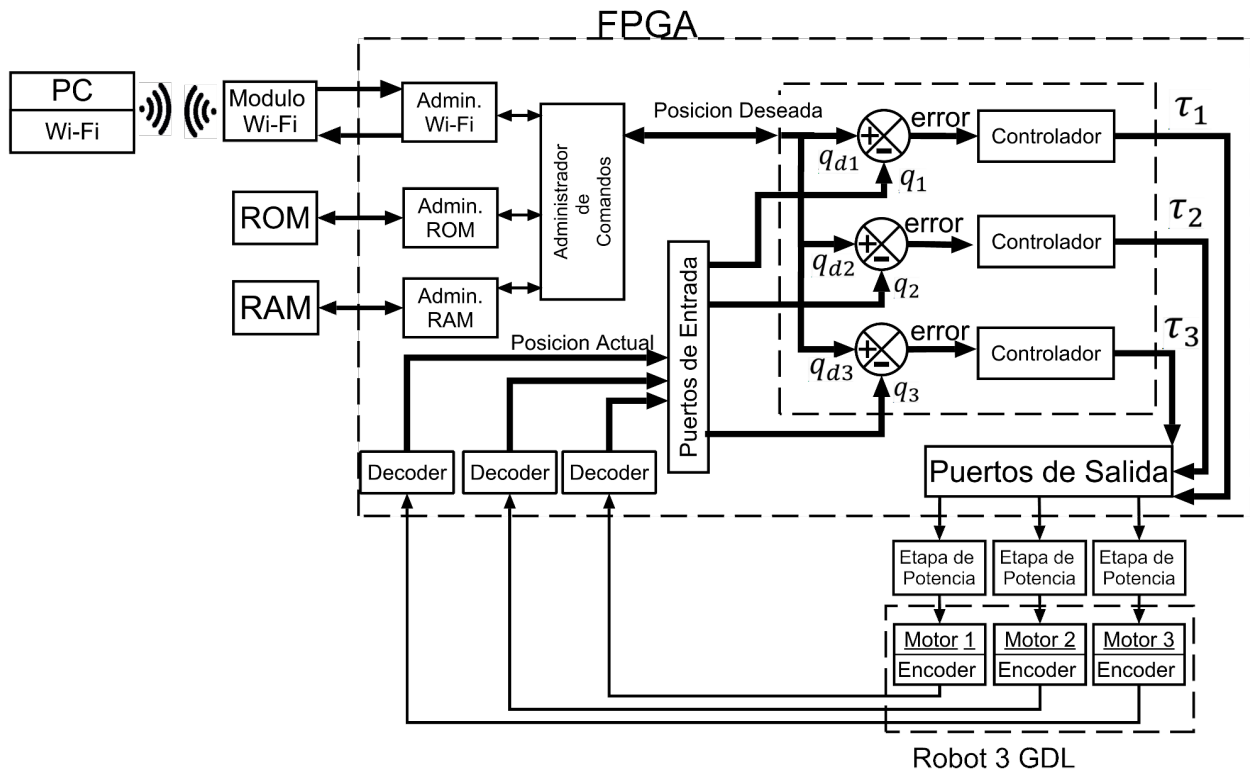


Diagrama general de la aplicación.

Del diagrama anterior, el usuario mediante una GUI (Interfaz) realizada en LabView se comunica hacia el sistema mecatrónico por medio de una interfaz wifi a través del módulo wifly, al recibirlos datos de posición deseada y ganancias se mandan a un bloque de administración de wifi el cual se comunica al bloque decodificador de comandos donde se ordenan y envían los datos hacia el bloque multiplexor(puertos de entrada), la salida del multiplexor entra a un bloque de procesamiento para ejecutar nuestro algoritmo de control el cual es uno de tipo proporcional.

El bloque de procesamiento es un micro-procesador implementado en FPGA capaz de ejecutar su propio conjunto de instrucciones en ensamblador, a su vez este conjunto de instrucciones se tiene que compilar para poder ser traducidas a un lenguaje hexadecimal que interpreta el procesador y permite la ejecución del control.

De esta forma se ejecuta el algoritmo de control y el resultado proporcionado como el error de posición procesado se de-multiplexa (puertos de salida) para mandar los datos a los tres bloques correspondientes de PWM donde se generan las señales hacia las etapas de potencia

Las etapas de potencia se encargan de proporcionar energía a los actuadores en nuestro caso los motorreductores, estos a su vez tienen en su eje de giro encoders de cuadratura los cuales proporcionan las señales digitales que entran a una etapa de acondicionamiento y la salida de la misma entra a un bloque decodificador para poder obtener la posición y sentido de giro donde se obtiene la posición actual ésta ingresa a la etapa de multiplexión (puertos de entrada) donde el procesador calcula el error restando la posición deseada de la actual, cerrando los 3 lazos de control correspondientes a los tres motores del robot.

4.5 Electrónica del robot

Al no contar con la electrónica del sistema, se optó por implementar puentes “h” en configuración darlington que al final activarán unos transistores Tip 35C (NPN) y Tip 36C (PNP), que pueden drenar hasta 25A, en modo de saturación y corte. Para poder mover nuestros motorreductores.

Para realizar la configuración darlington se utilizaron transistores convencionales, 3904 (NPN) y 3906 (PNP), sin embargo, para poder saturar los transistores Tip 35C y 36C se ocuparon otros transistores Tip 35C y 36C, pero en modo de amplificación ya que se requiere de 1.5A para saturar a 15A los transistores Tip. A continuación, se presenta el diagrama esquemático de los puentes “h”, implementados para los 3 motorreductores:

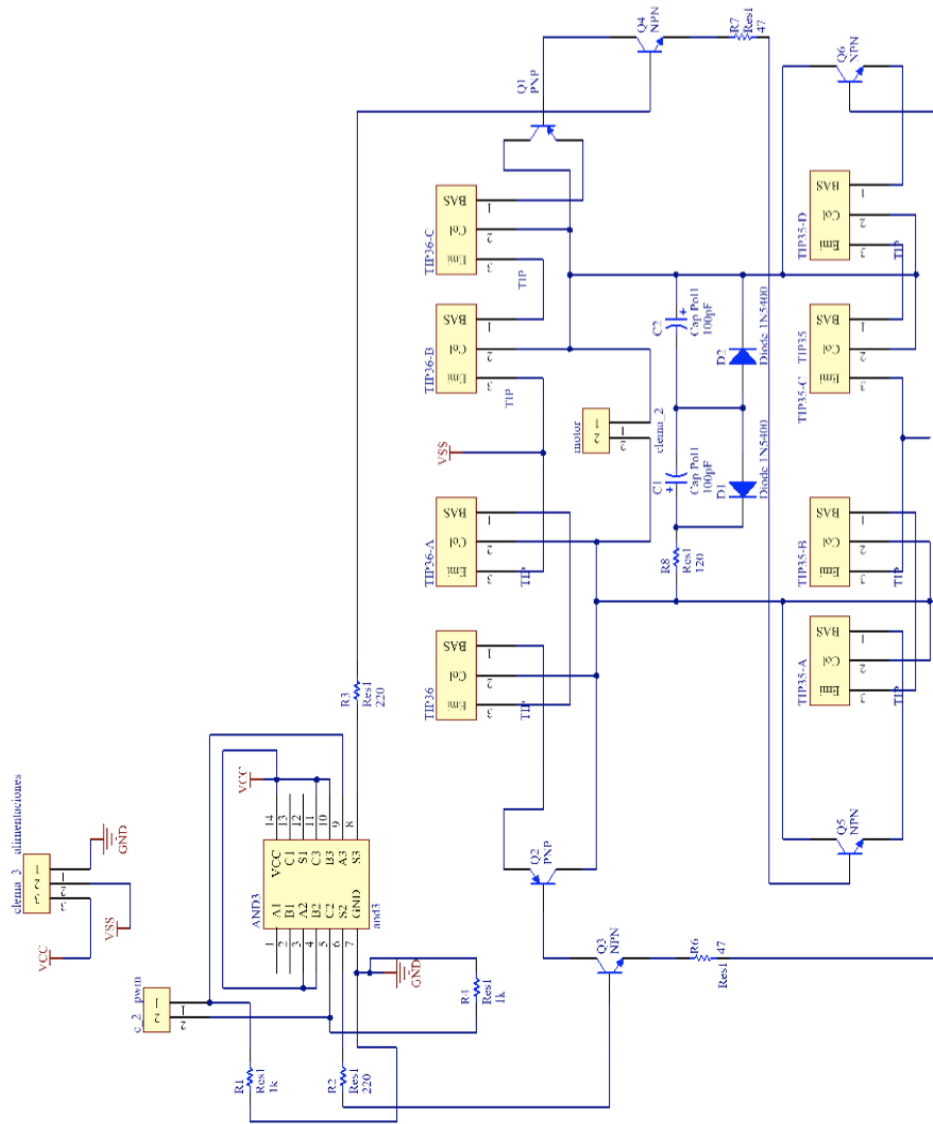


Figura 63: Diagrama esquemático del Puente H.

Como se puede ver en el diagrama anterior se tiene una configuración resistencia-capacitor (red snubber), conectado al motor, esto con el afán de reducir los sobretiros a la hora de energizar el motor, esto durante la conmutación de la señal. Además, se maneja una compuerta AND3 la cual recibe la señal del FPGA, para después pasarla a nuestra etapa de potencia, sirviendo como etapa de acoplamiento de señal.

Para la lectura de los encoders, se tuvo que hacer un divisor de voltaje, ya que los encoders manejan un voltaje de 5V en sus señales de salida y el FPGA, solo maneja un voltaje en un rango de 3.2-3.4V, para ello se empleó el siguiente circuito:

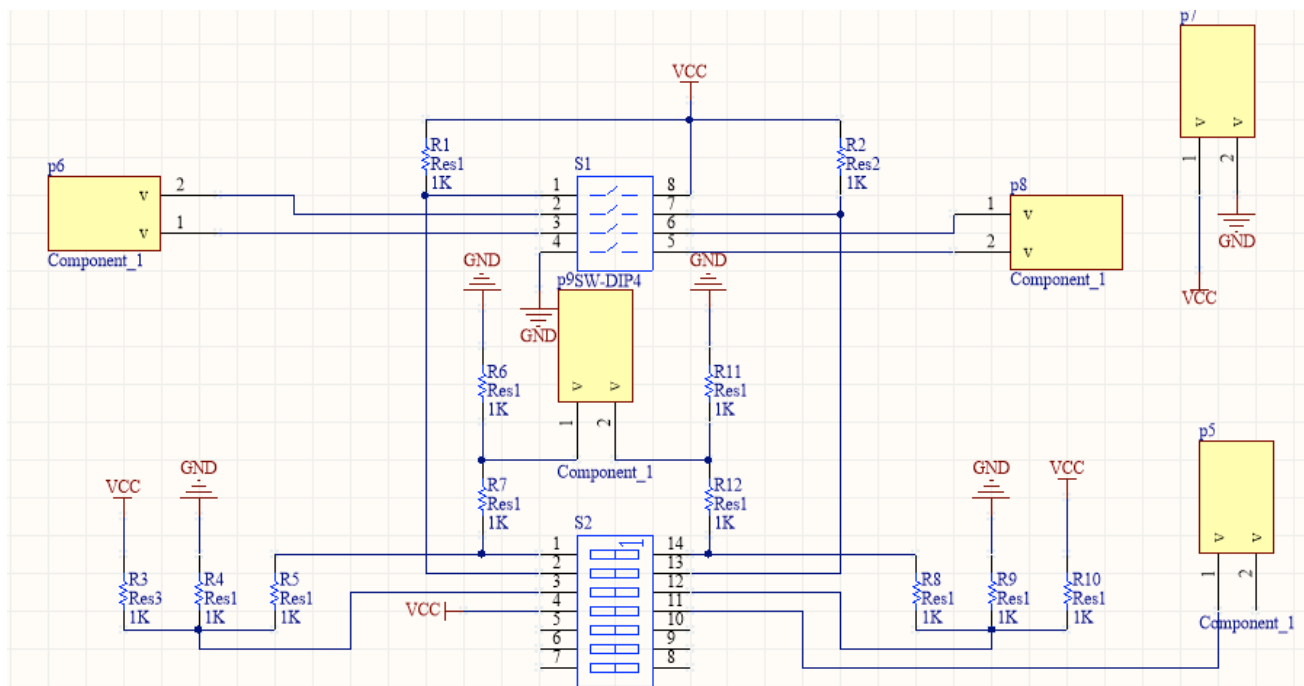


Figura 64: Diagrama esquemático para las señales de los encoders.

Del diagrama anterior se amplifica las señales de encoder para poder tener nuestras señales de cuadratura, así también se coloca un divisor de voltaje para que las señales de cuadratura obtenidas estén dentro del rango de los 3.3 volts, de esta forma no dañamos nuestra FPGA y se puede hacer la lectura de las señales satisfactoriamente.

4.6 Modelo dinámico

Para poder realizar un control apropiado para nuestro robot, es necesario tener en cuenta el modelo dinámico del mismo, esto nos servirá para analizar su comportamiento, al utilizar diferentes algoritmos de control y de esta manera poder utilizar el presente un mejor desempeño. [20]

Para determinar el modelo dinámico, utilizaremos la metodología de Euler-Lagrange, la cual está ampliamente documentada.

A continuación, se muestra la representación geométrica del robot cartesiano de 3 grados de libertad, la cual nos ayuda a visualizar el comportamiento del manipulador, la cual se tomó [17], para enriquecimiento académico del modelo dinámico del robot empleado en la presente tesis.

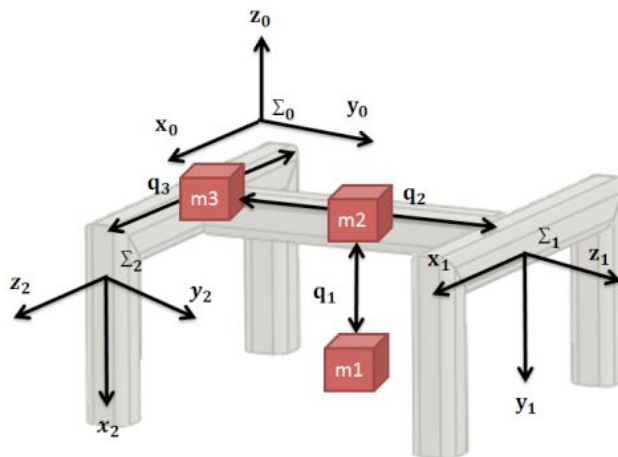


Figura 65: Representación del robot cartesiano de 3 grados de libertad.

Sabemos que, la energía total $E(q, \dot{q})$ (Hamiltoniano) de un robot manipulador de n grados de libertad es la suma de sus energías, cinética $K(q, \dot{q})$ y potencial $U(q)$. [3]

$$E(q, \dot{q}) = K(q, \dot{q}) + U(q) \quad (1)$$

Donde $q, \dot{q} \in \mathbb{R}^n$ representan los vectores de posición, en coordenadas generalizadas y velocidad articular.

Además, el lagrangiano $L(q, \dot{q})$ de un robot manipulador de n grados de libertad se define como la diferencia entre su energía cinética $K(q, \dot{q})$ y su energía potencial $U(q)$:

$$L(q, \dot{q}) = K(q, \dot{q}) - U(q) \quad (2)$$

Por otra parte, las ecuaciones de movimiento de Euler-Lagrange para un robot manipulador de n grados de libertad, están dadas por:

$$\frac{d}{dt} \left[\frac{\partial L(q, \dot{q})}{\partial \dot{q}} \right] - \frac{\partial L(q, \dot{q})}{\partial q} = \tau - v(\dot{q}, f_e) \quad (3)$$

Donde $\tau \in \mathbb{R}^n$ es el vector de pares aplicados y $v(\dot{q}, f_e) \in \mathbb{R}^n$

Es el vector de fuerzas o pares de fricción que dependen de la velocidad articular \dot{q} y de la fricción estática f_e , que se encuentran presente en las articulaciones del mismo.

De forma equivalente:

$$\tau_i = \frac{d}{dt} \left[\frac{\partial L(q, \dot{q})}{\partial \dot{q}} \right] - \frac{\partial L(q, \dot{q})}{\partial q} + b_i \dot{q} + f_{ci} \text{signo}(\dot{q}_i) + f_{ei} [1 - |\text{signo}(\dot{q}_i)|] \quad i=1, \dots, n \quad (4)$$

Donde b, f_c son coeficientes de fricción viscosa y de coulomb, respectivamente, f_e representa los coeficientes de fricción estática de cada una de los servomotores del robot manipulador.

De esto, podemos decir que dependiendo del número grados de libertad, será el número de ecuaciones dinámicas del robot manipulador.

Para determinar el modelo de cinemática directa utilizaremos la metodología de Denavit-Hartenberg, la cual ofrece un procedimiento sencillo y cuya estructura queda en representación de transformaciones homogéneas.

4.6.1 Modelo dinámico del robot cartesiano

Por lo tanto, el modelo dinámico cartesiano está dado por:

$$f_x = \begin{bmatrix} m_1 + m_2 + m_3 & 0 & 0 \\ 0 & m_2 + m_3 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + g \begin{bmatrix} m_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} b_1 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & b_3 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} f_{c1} & 0 & 0 \\ 0 & f_{c2} & 0 \\ 0 & 0 & f_{c3} \end{bmatrix} \begin{bmatrix} \text{signo}(\dot{q}_1) \\ \text{signo}(\dot{q}_2) \\ \text{signo}(\dot{q}_3) \end{bmatrix} + \begin{bmatrix} f_{e1} [1 - |\text{signo}(\dot{q}_1)|] \\ f_{e2} [1 - |\text{signo}(\dot{q}_2)|] \\ f_{e3} [1 - |\text{signo}(\dot{q}_3)|] \end{bmatrix} \quad (13)$$

Este modelo puede ser igualado a τ gracias a la siguiente relación:

$$\tau = J^T(q) f_x \quad (14)$$

Sin embargo, para realizar un acoplamiento de unidades entre τ (Nm) y f_x (N) se debe agregar una constante α la cual tendrá como valor 1 metro, por lo tanto, la ecuación quedará de la siguiente manera:

$$\tau = \alpha J^T(q) f_x = \alpha f_x \quad (15)$$

Obteniendo el resultado de pares aplicados de la metodología de Euler-Lagrange. Siendo estos los pares que se apliquen en cada eslabón de nuestro robot.

4.7 Software de control de posición el robot manipulador de inicio

Software

El software es todo lo intangible de una computadora, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware). El software suele dividirse, en cierta forma de modo arbitrario y a veces un poco difusa.

Interfaz grafica

En el contexto de la interacción persona-ordenador, la interfaz gráfica de usuario (IGU), es el artefacto tecnológico de un sistema interactivo que hace posible a través del uso y la representación del lenguaje visual, una interacción amigable con el sistema informático. Interface gráfica de usuario (de sus siglas en inglés Graphic User Interface, GUI) es un tipo de interfaz de usuario que utiliza un conjunto de imágenes y objetos gráficos (iconos, ventanas, tipografía) para representar la información y acciones disponibles en la interface.

Habitualmente las acciones se realizan mediante manipulación directa para facilitar la interacción del usuario con la computadora y esto surge como una evolución de la línea de comandos de los primeros sistemas operativos y es pieza fundamental en un entorno gráfico.

Existen en la actualidad diferentes softwares de control, utilizados en la industria o en la educación, cada software presenta diferentes ventajas respecto a los demás. Para aplicaciones de control es necesario que el software sea capaz de trabajar en tiempo real.

Pantalla de usuario

Existen diferentes tipos de software, dentro de los cuales el usuario debe de elegir el software dependiendo de su aplicación y nivel de confianza que requiera. Hoy en día, las aplicaciones son cada vez más complejas, por lo que el software especializado cada vez se presenta en forma más amigable al usuario, quien puede tener un cierto nivel de abstracción hacia esta herramienta, de aquí que, en la actualidad, existan diferente software de control, utilizados en la industria o en la educación y cada software presenta diferentes ventajas respecto a los demás. Basados en la ventaja del software especializado en la adquisición y control, para este trabajo el software utilizado es LabView de National Instruments, el cual posee un ambiente de desarrollo grafico brindando la flexibilidad de cualquier otro lenguaje de programación tales como C, pascal, etc., sin la complejidad de las herramientas de desarrollo tradicionales. Es decir, el usuario no tiene que preocuparse de la sintaxis (comas, puntos, corchetes, llaves, paréntesis, etc.), Labview es mucho más amigable, usa iconos para representar subprogramas, y se unen estos iconos para definir el flujo de los datos, permitiendo la adquisición de señales, análisis, escritura, además de ejecutar algoritmos matemáticos. De esta manera al utilizar la interface gráfica de Labview, e ir trazando el flujo de los datos utilizando iconos, el usuario realiza su código con lo cual se pueden escribir programas en menos tiempo que si se realizaran en forma de texto.



Figura 66: Pantalla principal del software.

En esta pantalla nosotros podemos visualizar las posiciones deseadas, para cada eje, además de poder ejecutar y detener del software, se puede borrar y leer la memoria ROM, así como grabarle a esta el firmware desarrollado para el control del robot de tres grados de libertad, el cual se mencionará más adelante, así también se puede modificar los parámetros de las ganancias que desee usar el usuario.

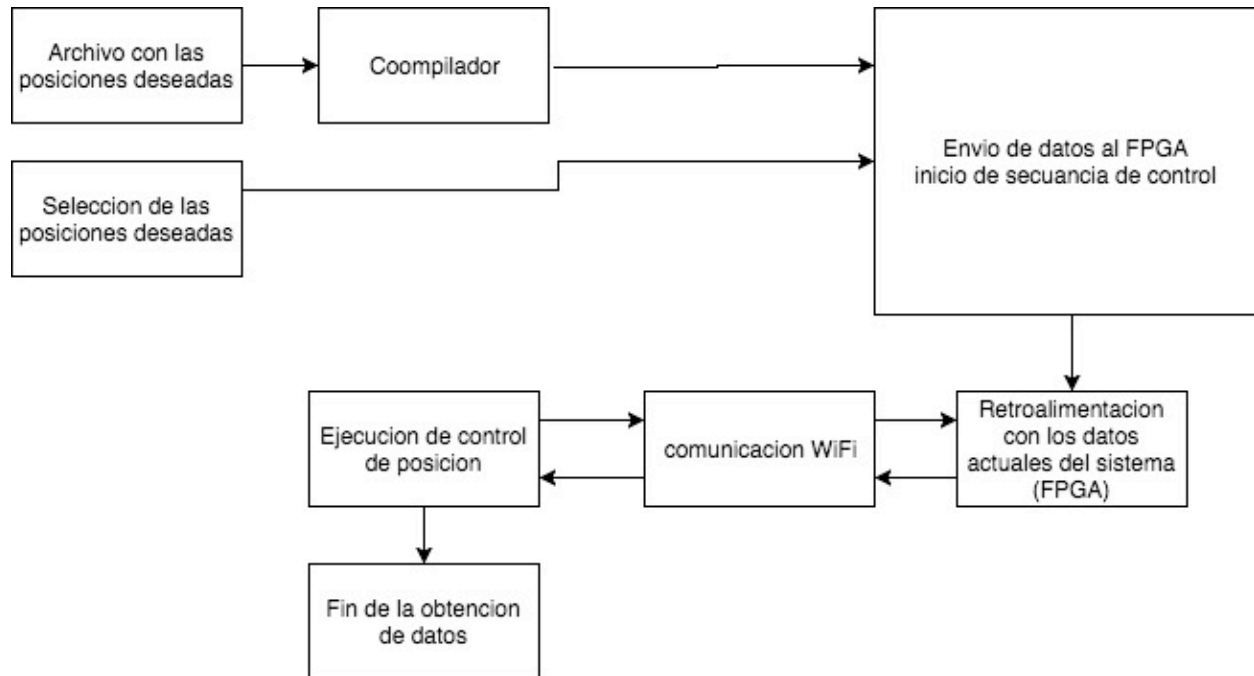


Figura 67: Diagrama a bloques del software.

En el diagrama anterior se muestra brevemente el funcionamiento del software de control del robot, empezando por un programa basado ensamblador pre cargado y realizado por una serie de instrucciones, el cual pasa a un compilador para seleccionar las posiciones de cada motor y posteriormente ser ejecutadas por nuestro robot. Una vez que ya se definieron estos parámetros, se inicia la secuencia de control, durante la ejecución de la secuencia de control, nuestros algoritmos se mantendrán ejecutándose para mandar constantemente la energía que debe ser aplicada a nuestros actuadores, además, se tiene un periodo de muestreo cada 2.5 milisegundos, para ir actualizando la información de nuestros sensores. El envío y recepción de información se realiza a través del protocolo de comunicación wifi, por lo que constantemente se tiene que ir actualizando la lectura y escritura del puerto. El software mantendrá enviando la

señal de control al robot mientras la secuencia se encuentre activa. Cuando termina la secuencia de control el software indica al sistema apagar las señales que controlan el robot y este se posicionara en su posición de inicio.

Se decidió utilizar la tarjeta dedicado sólo a la adquisición y envío de datos para que nuestro algoritmo de control, en este caso el control de posición con argumento de control proporcional se pueda implementar el software de Labview con más poder de cómputo. La programación de estos procesos es realizada dentro de la FPGA, se realiza la lectura y escritura de datos, en base a eso se obtiene la posición del motor, obtenemos el error de posición, se obtiene la posición actual, se realiza el control, en base al control se dan los giros al sistema, se manda el par por medio de PWM y se repite el ciclo.

La interfaz de usuario se realizará mediante la ayuda del software labview en la cual el usuario tendrá el control del sistema, El sistema tendrá la tarea de interpretar los datos del encoder, inicializar el algoritmo de control, transmitir la posición actual y par aplicado.

La ventaja de este sistema, es que, el algoritmo de control se encuentra dentro de la FPGA mediante firmware, aumentando el tiempo de ejecución del algoritmo de control tomando así datos en tiempo real, puesto que es un hardware dedicado a realizar esa tarea específica.

El software del sistema se realizó en LabView, ya que esta plataforma nos permite manipular fácilmente el control del robot, nuestro bus de transmisión y lectura de datos. Mediante este programa el usuario puede visualizar la información adquirida por el FPGA, de nuestros diferentes sensores, como los encoders para el caso de la posición del robot y, Además, el usuario también puede modificar los diferentes parámetros del sistema, como la, las posiciones deseadas para cada uno de nuestros ejes.

4.8 Conclusión

En este capítulo se aborda como se realiza el proceso para obtener las posiciones deseadas de los motores, donde el software se le graba el archivo necesario de la ley de control aplicada al motor. Gracias a este proceso de compilado no es necesario involucrar más software para realizar nuestro control del robot. Por otro lado, se menciona como el software ocupa el modulo Wifi para enviar los parámetros necesarios para realizar el proceso de control de posición en nuestro firmware, además, de realizar la lectura de datos para que el usuario pueda visualizar la posición actual del robot.

CAPITULO V

RESULTADOS

Para verificar el correcto funcionamiento del sistema se monitorearon sus variables de control y se compararon con los resultados obtenidos con el sistema comercial. En esta prueba al sistema se controlan los tres grados de libertad del robot, llevando cada grado de libertad a una posición deseada fija, las posiciones deseadas a la cual se realiza la prueba se puede ver en la tabla 6.4, para los tres grados de libertad.

Grado de libertad.	Posición Deseada.
Motor 1	90º
Motor 2	90º
Motor 3	90º

Tabla 9: Posiciones deseadas para cada motor.

En la figura 71, se muestran las gráficas de evolución de la posición para ambos sistemas de control durante el experimento. Observando una posición final de 89. 98958º para el sistema con la tarjeta de adquisición de datos y control realizada en la presente tesis y una posición final de 89. 2312º para el sistema con la tarjeta PCI, la diferencia entre ambas posiciones finales es de 0.72838º.

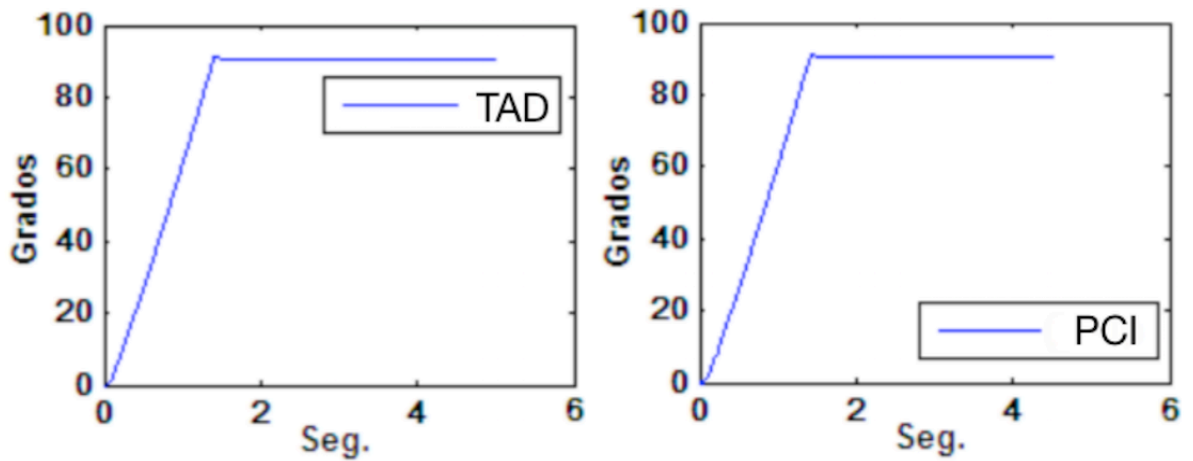


Figura 68: Gráfica de posición a 90° del motor 1.

El error de posición durante el experimento en ambos sistemas puede apreciarse en las gráficas de la figura 72. Observando un error de posición final de 0.412519° para el sistema con la tarjeta de adquisición de datos y control realizada en la presente tesis y un error de posición final de 0.41324° para el sistema con la tarjeta PCI.

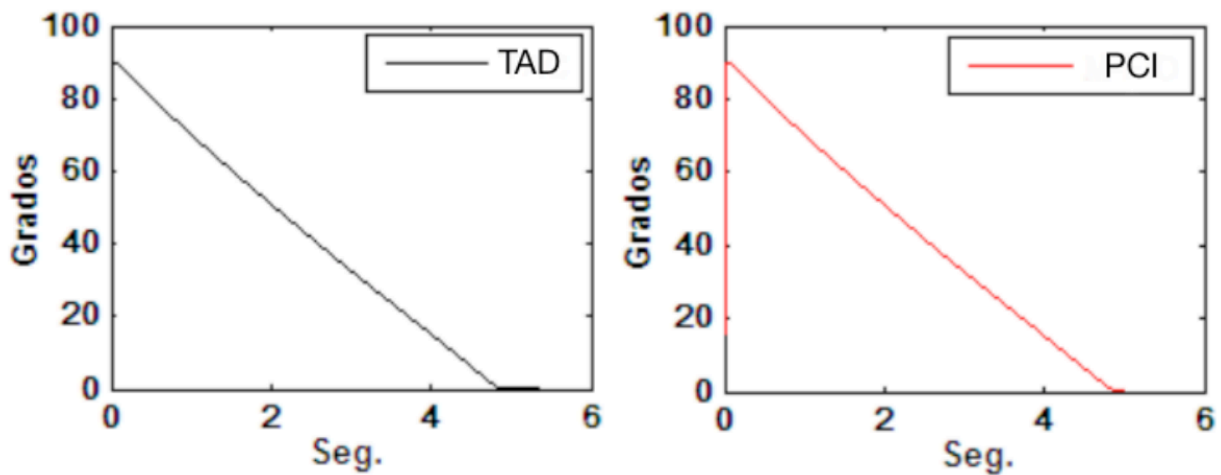


Figura 69: Gráfica de error de posición

En la figura 73, se muestra la respuesta de par en ambos sistemas. La respuesta de par del sistema con la tarjeta PCI presenta una respuesta con una oscilación mayor respecto de la respuesta de par del sistema con la tarjeta de adquisición de datos y control diseñada en la presente tesis.

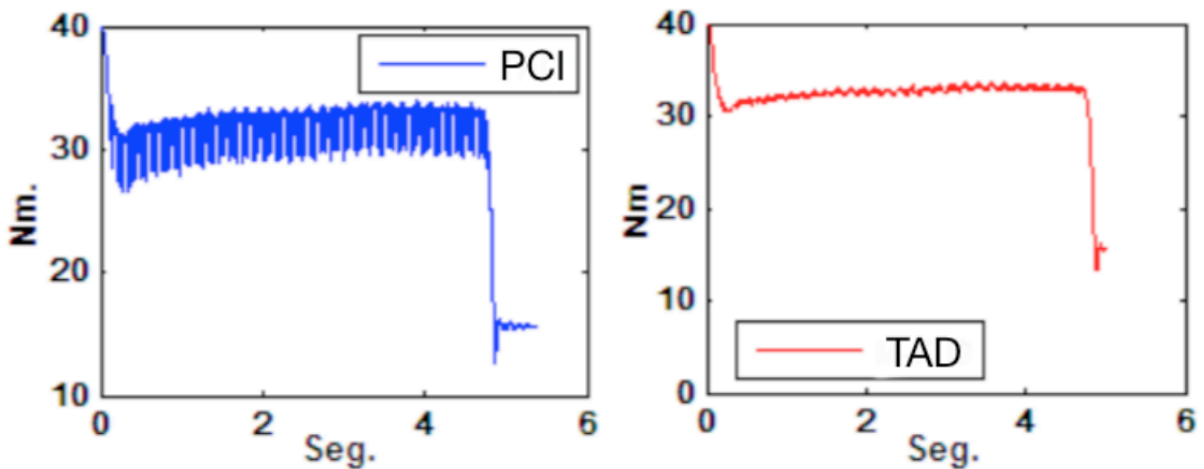


Figura 70: Grafica de par del motor.

En la figura 74, se muestran las gráficas de evolución de la posición para ambos sistemas de control durante el experimento. Observando una posición final de 89.98958° para el sistema con la tarjeta de adquisición de datos y control realizada en la presente tesis y una posición final de 89.3312° para el sistema con la tarjeta PCI, la diferencia entre ambas posiciones finales es de 0.72838° .

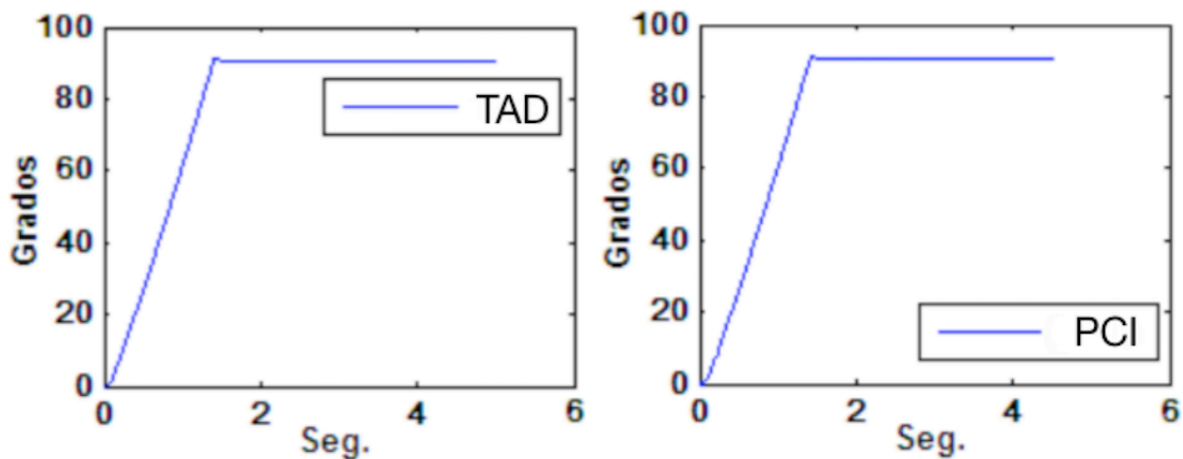


Figure 71: Grafica de posición del motor 2

Conclusiones Generales

De acuerdo a los resultados obtenidos en las pruebas del sistema de control con la tarjeta de adquisición de datos y control desarrollada en la presente tesis, donde la lectura de posición de cada grado de libertad del robot es muy similar a las lecturas obtenidas con el sistema utilizando la tarjeta PCI y por lo cual los parámetros de par y error siguen un perfil similar, se puede concluir que el funcionamiento y por lo tanto el diseño de la tarjeta de adquisición de datos y control de la presente tesis es exitoso, y como consecuencia los objetivos planteados en este tema de tesis han sido concluidos satisfactoriamente.

Cabe mencionar que los resultados obtenidos de la posición y escritura de las señales de control de los tres grados de libertad del robot, se puede concluir como exitoso el desarrollo del firmware necesario para procesar las señales de cuadratura de los encoders considerando importante el mencionar la obtención de un índice digital permitiendo el ahorro de componentes e igualmente de forma satisfactoria el firmware para el control de los circuitos integrados para generar la señal de par de cada motor.

Obteniendo un buen control de nuestro robot, se pudo verificar el buen funcionamiento de la tarjeta de adquisición de datos y control diseñada en la presente tesis, podemos decir que las técnicas utilizadas en el diseño de la tarjeta para minimizar los efectos de EMI, RF e interferencia entre componentes, dieron como resultado el correcto funcionamiento de la tarjeta de adquisición y control dentro del sistema de lazo cerrado, obteniendo un voltaje inducido en las pistas por los motores, manejadores, PC, equipo eléctrico y electrónico, de una amplitud que no afecta en lo absoluto el funcionamiento de componentes o señales.

Al realizar la fabricación de la tarjeta pueden ocurrir errores independientes al diseño como cortocircuitos, esto fue verificado previamente al montaje de componentes, mediante la verificación de continuidad entre pistas y planos, con lo que se encontró un cortocircuito entre dos pistas, lo cual fue corregido en tiempo y forma. De este hecho obtenemos que de no haber realizado esta prueba a la tarjeta se pueden ocasionar daños a la infraestructura del sistema de control del robot.



Universidad Autónoma de Baja California
Escuela de Ciencias de la Ingeniería y Tecnología
Unidad Valle de las Palmas

1er. Congreso Internacional de Ciencias de la Ingeniería y Tecnología 2017

Otorga la siguiente

CONSTANCIA

a

**José de Jesús Moya Mora, Sergio Vergara Limón,
David Eduardo Pinto Avendaño**

Por su participación en el Eje Temático Electrónica y Bioinstrumentación con el trabajo titulado
Tarjeta para Controlar 3 Grados de Libertad de Robots Vía Wifi

Tijuana, Baja California, México, del 20 al 23 de septiembre del 2017



Mtro. Alonso Hernández Guitrón
Director de la ECITEC

Dra. María de los Angeles Zárate López
Coordinadora de Posgrado e Investigación

Dr. Luis Jesús Villarreal Gómez
Presidente del Comité Organizador

Dr. Alberto Hernández Maldonado
Presidente del Comité Científico



Tarjeta para controlar 3 grados de libertad de robots vía wifi

Moya Mora¹, Vergara Limon¹, Pinto²

¹Facultad de Ciencias de la Electrónica, Maestría en Ciencias de la Electrónica opción Automatización, Benemérita Universidad Autónoma de Puebla.

Av. San Claudio y 18 Sur S/N C.U. Edif. 109A Col. Jardines de San Manuel, C.P. 72570, Puebla, Pue. México.

moyamora@me.com

²Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, Av. San Claudio y 14 Sur S/N C.U. Col. Jardines de San Manuel, C.P. 72570, Puebla, Pue. México.

Resumen. - *El presente trabajo describe el desarrollo de un sistema para controlar un robot de tres grados de libertad utilizando un sistema embebido, la transferencia de información se realiza mediante el protocolo de comunicación WIFI. El robot que se utiliza es el tipo cartesiano. Se hace una descripción del diseño mecánico del robot, así como la obtención del modelo cinemático y dinámico. Se detalla el diseño de un sistema de adquisición, desarrollando un sistema embebido basado en FPGA de la marca altera de la familia Cyclone III el cual cuenta con un controlador interno, así como la interpretación del manejo de las señales del robot y comunicación WIFI, se explica el desarrollo del firmware que realiza el control y como se establece la comunicación. El controlador seleccionado es el de tangente hiperbólica. Dicho controlador es implementado en el firmware con el fin de controlar y analizar el comportamiento del sistema.*

Palabras clave: FPGA, Protocolo de comunicación WIFI, Control Tangente Hiperbólico.

Abstract. - *The present work describes the development of a control system for a three D.O.F cartesian robot using an embedded system where the information transfer is done by the WIFI communication protocol. A description of the mechanical design of the robot is given, as well as the kinematic and dynamic model. It details the design of an acquisition system, developed in FPGA based embedded system of the Altera Cyclone III family, and also the interpretation of the robot signal handling and WIFI communication. In addition, it shows the development of the firmware responsible of the control and communication. Regarding the control, a hiperbolic tangent controller is selected for preliminary results and with these, the cartesian robot can be analyzed and regulated.*

Key words: FPGA, Protocolo de comunicación WIFI, Control Tangente hiperbólico.

1. Introducción

La instrumentación es una disciplina de alto impacto en diferentes aplicaciones, tales como la física, el control, por mencionar algunas. Es el control una disciplina que a su vez interviene en un sin fin de aplicaciones, como la robótica; Entonces se presenta que diferentes disciplinas interactúan de diferentes maneras para lograr un mismo fin. [1]

El desarrollo tecnológico o diseño de dispositivos que como en este trabajo se abordará el diseño de una

tarjeta de adquisición de datos y control es de una gran complejidad, y prueba de ello es que este tipo de desarrollo es realizado por compañías transnacionales, por lo que el desarrollo tecnológico en países como México es limitado, puesto que partir de una idea o una función determinada que se quiera que realice el dispositivo a diseñar hasta obtener un primer prototipo es una tarea bastante compleja a realizar, debido a que en ciertas áreas de la ciencia la bibliografía o referencias especializada en determinados temas es amplia y en ocasiones bastante rebuscada de manera que no cualquier lector podría deducir como es que funciona algún tema en específico que desee leer, comprender e interpretar,



en el caso de desarrollos tecnológicos el acceso a bibliografía o referencias es limitada, debido a que esta información es generada por compañías las cuales comercializan estos desarrollos y que invirtieron en recursos humanos e infraestructura para poder generar estos diseños, es por ello que solo es público un manual de uso o guía de usuario para poder utilizar sus diseños comercializados, pero nunca habrá información que permita a terceras personas realizar un producto similar o basado en su diseño.

Se aborda la problemática de diseño de un sistema de adquisición de datos y control para un robot cartesiano de tres grados de libertad, se analizan las señales que interactúan en el sistema de control del robot, de esta forma se selecciono los dispositivos electronicos adecuados para el tratamiento y emisión de estos, además del firmware necesario para el tratamiento, decodificación y emisión de las señales necesarias en el sistema de control. El diseño de tarjetas de adquisición de datos y control tiene una amplia complejidad, por lo que en su mayoría son desarrollados por compañías transnacionales en las cuales un equipo multidisciplinario realiza los diseños, por esta razón el desarrollar diseños de tarjetas de adquisición de datos en el país permite crear dispositivos que satisfagan las necesidades y técnicas para una aplicación de control, por ello esta tarjeta es de arquitectura abierta y con una comunicación inalámbrica con la PC, lo cual permite la transmisión de datos vía wifi. La creación de una pantalla de usuario utilizando LabView permite la combinación de procesamiento de la PC con la tarjeta diseñada. La tecnología avanza a grandes pasos, por lo que la investigación de dispositivos de última generación se vuelve de vital importancia, es por ello que se presenta la investigación de instrumentación de dispositivos en los FPGA de ALTERA de última generación.

2. Metodología

2.1 Diagrama general del sistema embebido

El sistema embebido desarrollado le permitirá al usuario manipular el robot como el desee mediante un monitoreo de manera inalámbrica vía WIFI. El sistema en general, está compuesto por hardware, firmware y software los cuales se describirán a continuación

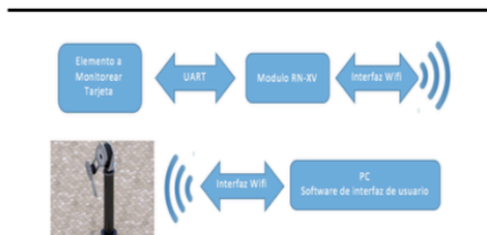


Figura 1. Diagrama general edel sistema embebido.

Del diagrama anterior, el elemento a monitorear es la articulación del robot, se obtienen los valores correspondientes a la posición actual y par aplicado, en base al tiempo y a la posición deseada, se calcula el error de posición a lo largo del funcionamiento del sistema. Estos datos son leídos a través de la FPGA y son enviados a un módulo WIFI mediante el protocolo UART de sus siglas en inglés Universal Asynchronous Receiver Transmitter.

El modulo WIFI es el RN-XV, este dispositivo realizara la tarea de conversión de WIFI a serial y de serial a WIFI. Esta comunicación es bastante atractiva para el usuario ya que se hace compatible con computadoras de escritorio o computadoras portátiles de manera inalámbrica. [2]

La interfaz de usuario se realizará mediante la ayuda del software labview en la cual el usuario tendrá el control del sistema, El sistema tendrá la tarea de interpretar los datos del encoder, inicializar el algoritmo de control, transmitir la posición actual y par aplicado.

La ventaja de este sistema, es que, el algoritmo de control se encuentra dentro de la FPGA mediante firmware, aumentando el tiempo de ejecución del algoritmo de control tomando así datos en tiempo real, puesto que es un hardware dedicado a realizar esa tarea específica.

2.2 Diseño del robot cartesiano

Para poder realizar un control apropiado para nuestro robot, es necesario tener en cuenta el modelo dinámico del mismo, esto nos servirá para analizar su comportamiento, al utilizar diferentes algoritmos de



control y de esta manera poder utilizar el presente un mejor desempeño.

El modelo dinámico de nuestro robot, se emplea la metodología de Euler-Lagrange, la cual está ampliamente documentada.

Para ello se muestra la representación geométrica del robot cartesiano de 3 grados de libertad, la cual nos ayuda a visualizar el comportamiento del manipulador.

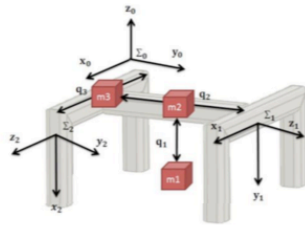


Figura 2. Representación del robot cartesiano de 3 grados de libertad..

Por lo tanto, el modelo dinámico cartesiano está dado por:

$$f_s = \begin{bmatrix} m_1 + m_2 + m_3 & 0 & 0 \\ 0 & m_1 + m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + g \begin{bmatrix} m_1 \\ m_2 \\ 0 \end{bmatrix} + \begin{bmatrix} h_1 & 0 & 0 \\ 0 & h_2 & 0 \\ 0 & 0 & h_3 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} f_{1s} & 0 & 0 \\ 0 & f_{2s} & 0 \\ 0 & 0 & f_{3s} \end{bmatrix} \begin{bmatrix} \text{sign}(\dot{q}_1) \\ \text{sign}(\dot{q}_2) \\ \text{sign}(\dot{q}_3) \end{bmatrix} + \begin{bmatrix} f_{11} + f_{12} \\ f_{21} + f_{22} \\ f_{31} + f_{32} \end{bmatrix} \quad (1)$$

En nuestro caso no se cuenta con la matriz de fuerzas centrípetas y de Coriolis, ya que es igual a cero, con respecto a la matriz de inercia cartesiana nosotros contamos con una matriz de masas, mencionada con anterioridad.

Se presenta el robot cartesiano integrado completamente.

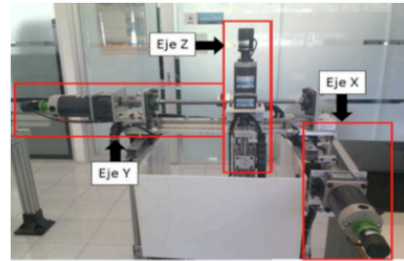


Figura 3. Representación del robot cartesiano de 3 grados de libertad..

2.3 Diseño de la interfaz gráfica para manipular el robot cartesiano de 3 grados de libertad.

Se hará una descripción concreta del software de control de posición, el cual se realizó con el programa de LabView, que es una herramienta que nos permite tener acceso de forma sencilla al protocolo de comunicación WIFI, permitiéndonos visualizar los datos que se escriben desde la computadora hasta la tarjeta de adquisición de datos.



Figura 4. Representación del robot cartesiano de 3 grados de libertad..

En esta pantalla nosotros podemos visualizar las posiciones deseadas, la ganancia proporcional y derivativa, para cada eje.

Desde esta interfaz podemos inicializar nuestro sistema, cargar un archivo de texto “.txt”, el cual cuenta con el algoritmo de control necesario para poder tener una manipulación de nuestro robot de forma óptima, nos permite ingresar la posición deseada, en la cual nuestro robot llegara a esta



posición de forma precisa, la precisión de llegar al punto deseado varía dependiendo de la ley de control que se le aplique al robo.

3. Resultados

Se realizó la caracterización de los motores de dc, para convertirlos en pseudo-servomotores de transmisión directa. Siguiendo la siguiente metodología, primero se determina el voltaje nominal de cada motor, en nuestro caso fue de 12v; Después se hace la búsqueda de ña frecuencia de trabajo de cada motor, la cual nos proporciona la mitad del torque máximo con un ciclo de trabajo de 50%, una vez localizada esta frecuencia se hace un barrido cambiando el porcentaje de ciclo de trabajo para poder analizar su comportamiento, que tiene que ser muy cercano al lineal. De este proceso se obtuvieron los siguientes resultados:

Tabla 1: Datos adquiridos del primer motor a una frecuencia de 2.45Khz.

%Ciclo de trabajo del PWM	Torque (Nm)
0	0
10	2.8616
12	3.9494
14	4.3708
16	5.2528
18	5.9976
20	6.4288
22	7.595
24	8.1144
28	9.555
34	10.8192
36	12.348

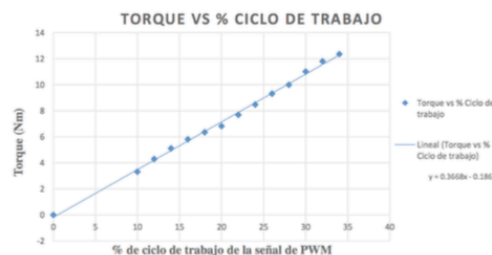


Figura 5. Gráfica de resultados del primer motor, se puede observar la linealidad del motor.

Se caracterizó hasta el 36% del ciclo de trabajo, ya que el servomotor nos proporciona el torque necesario para nuestra aplicación.

Para el segundo motor se obtuvieron los siguientes resultados a una frecuencia de 2.5Khz:

Tabla 2: Datos adquiridos del segundo motor a una frecuencia de 2.5Khz.

%Ciclo de trabajo del PWM	Torque (Nm)
0	0
10	3.3124
12	4.2924
14	5.1058
16	5.8016
18	6.3406
20	6.8208
22	7.68208
24	8.477
26	9.31
28	9.996
31	11.0152
33	11.7992
36	12.348

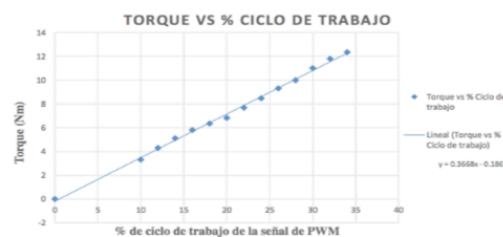


Figura 6. Grafica de resultados del segundo motor, se puede observar la inealidad del motor.

Finalmente para el tercer motor se encontro que la mejor frecuencia de trabajo fue de 75Hz, obteniendo los siguientes resultados:



Tabla 3: Datos adquiridos del tercer motor.

%Ciclo de trabajo del PWM	Torque (Nm)
0	0
10	0.0616077
20	0.3435096
30	0.6571488
40	0.840105
50	1.250823
60	1.437513
70	1.8930366
80	2.072259
90	2.5688544

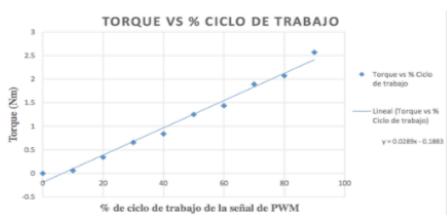


Figura 7. Grafica de resultados del tercer motor, se puede observar la inercialidad del motor.

4. Conclusiones

A diferencia de los robots cartesianos comerciales, este manipulador utiliza una comunicación inalámbrica como interfaz, por medio de una tarjeta de adquisición de datos, la cual es una FPGA. Esto le proporciona grandes beneficios como velocidad de procesamiento de 100MHz, arquitectura abierta por mencionar algunos, en comparación con algunos sistemas empotrados como Arduino mega que maneja una velocidad de reloj de 16MHz. Además, la programación en FPGA es muy versátil y nos permite realizar varios procesos de control en tiempo real, que es muy importante para cualquier aplicación.

Al emplear el software de labview, nuestro programa fue más sencillo de realizar mediante el protocolo de comunicación Wifi, sin embargo, cabe destacar que

hay que tener cuidado con las direcciones que se ocupan y la información que se envían porque podemos causar problemas con el sistema. Este programa nos permite modificar de manera sencilla el algoritmo de control y visualizar los datos que deseamos monitorear.

Referencias

- [1] D. Bernstein and L. Bushnell. "The History of Control: From Idea to Technology". IEEE Control Systems Magazine, 22(2):21-23, April 2002.
- [2] A. Carballar, J. (2017). Wi-Fi Instalación, Seguridad y Aplicaciones (Vol. 1). México, México, México: Alfaomega.
- [3] Kelly R. y V. Santivañez, "Control de Movimiento de Robots Manipuladores.", Prentice Hall, 2003.
- [4] Dorian Rojas Balbuena, "Diseño, construcción e instrumentación de un robot cartesiano pick and place de 4 GDL", Tesis de maestría FCE, BUAP, Puebla, Puebla, junio del 2011.
- [5] Fernando Reyes Cortes, "Robótica control de robots manipuladores", Ed. alfa omega, 1er edición.
- [6] Josep Balcells, José Luis Romeral, "Autómatas Programables", Ed. Marcombo, 1997.
- [7] Javier Vargas Valencia, Iliana Ramírez Velázquez, Santiago Pérez Walton, "física mecánica conceptos básicos y problemas", Ed. ITM, 1a edición diciembre del 2008.
- [8] Antonio Barrientos, Luis F. Peñin, Carlos Balaguer, Rafael Aracil. "Fundamentos de robótica", Ed. Mc Graw Hill, 1997.
- [9] Anibal Ollero Baturone. "Robótica Manipuladores y robots móviles", Ed Alfaomega, julio de 2008.
- [10] Subir Kumar Saha. "Introducción a la Robótica", Ed. Mc Graw Hill, 2008.
- [11] Robot cartesiano de 3 g.d.l. Lexium, disponible en: http://www.alumotion.eu/portfolio_item/urcaps-uring/ 2015.
- [12] Guía aplicada del robot cartesiano, Schneider electric, 11 de diciembre del 2008, versión 1.3.

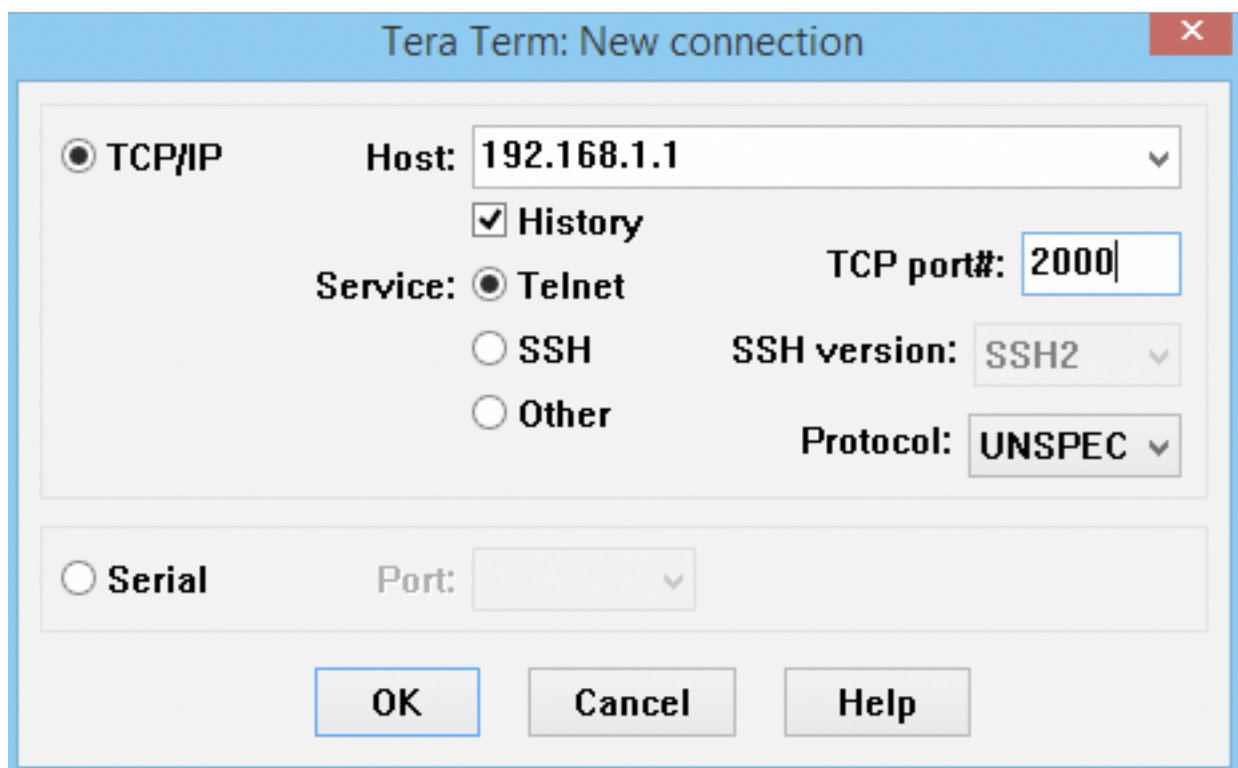
Apéndice C

Configuración del módulo WiFly

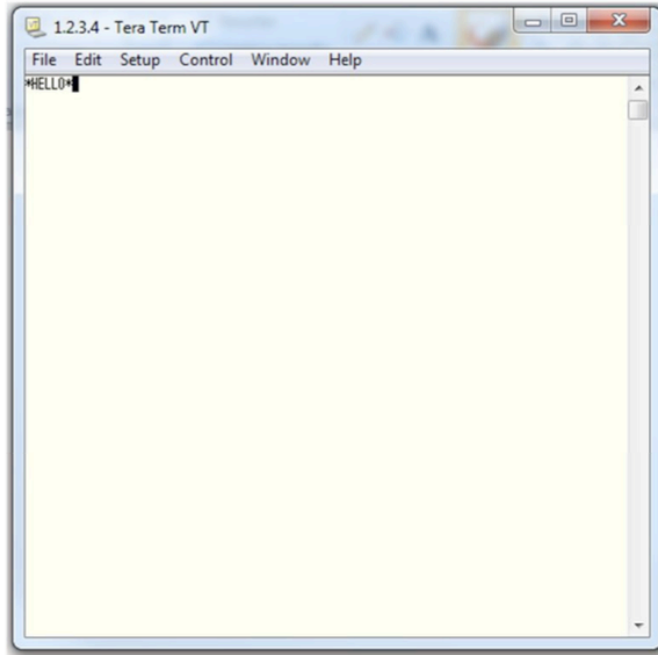
Modulo RN-XV Wifly-EZX-29 Dirección ip: 192.168.1.1

Se conecta el modulo wifi a 3.3v para que la computadora lo reconozca, crea una red llamada wifly-EZX-29, conectas tu computadora a esa red, y ahora se pueda a empezar a realizar la configuración del módulo.

Se abre el programa llamado Tera Term, aparecerá una pantalla como la siguiente:

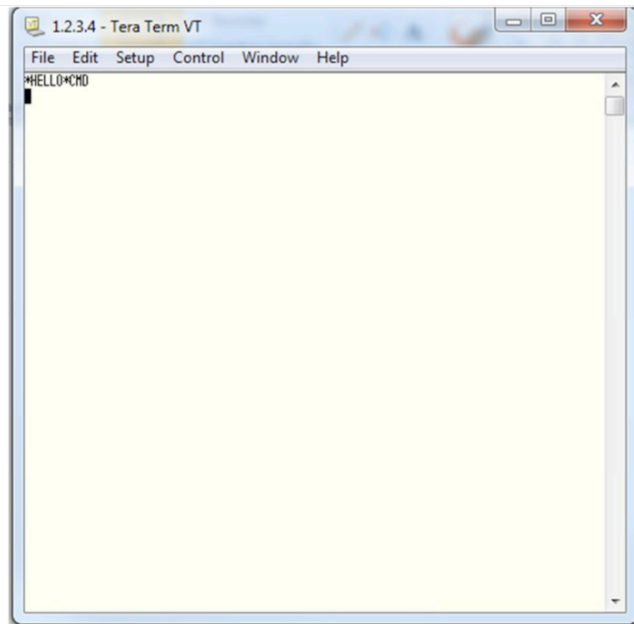


es importante que este seleccionado TCP/IP como se muestra en la figura, la casilla donde dice Host debe tener la dirección ip del módulo wifi que se desee configurar, en este caso para este módulo será 192.168.1.1, en la casilla Service seleccionamos Telnet, posteriormente en la casilla TCP port# en esta casilla ponemos 2000. Lo demás no se debe modificar, seleccionamos la casilla OK y el programa nos abrirá otra ventana, la cual se muestra a continuación:

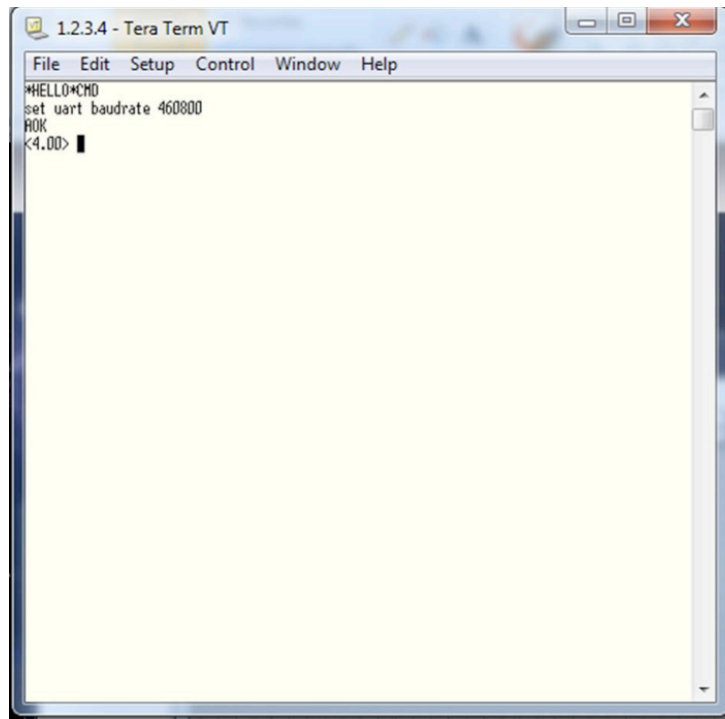


cuando se colocan los parámetros correctos antes mencionados, el módulo nos repondrá con una frase en este caso es HELLO, la cual aseguramos que se realizó la conexión entre la computadora y el módulo de forma correcta, procedemos a escribir \$\$\$ en la ventana que se abrió, para de esta forma poder configurar el modulo, la venta desplegara lo siguiente:

con esto escribiremos los comandos siguientes para proceder ahora si a configurar nuestro módulo:



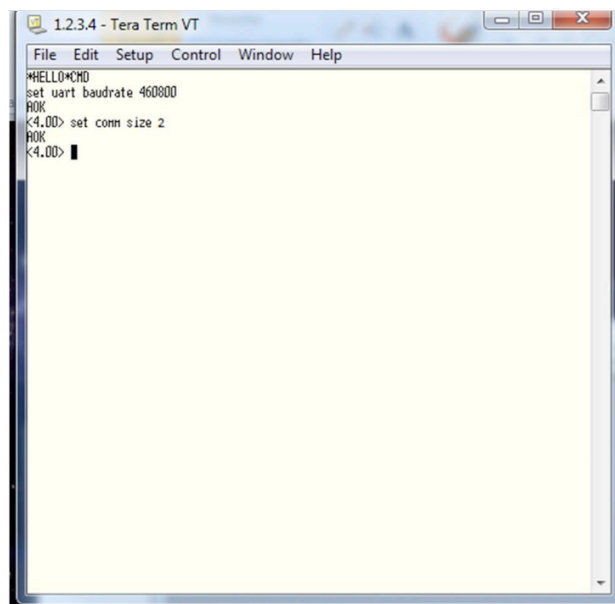
como primer paso escribiremos el comando: set uart baudrate 460800 posteriormente presionamos la tecla enter y el módulo responderá AOK, la cual indica que el comando se introdujo de forma correcta, en la pantalla verás lo siguiente:



```
1.2.3.4 - Tera Term VT
File Edit Setup Control Window Help
#HELLO*CMD
set uart baudrate 460800
AOK
<4.00> █
```

con esto configuramos la tasa de transmisión de datos de nuestro módulo.

Ahora, se procede a configurar el número de bytes para este caso será 3 bytes, el cual se realizará con el siguiente comando: set comm size 3 seguido de la tecla enter, se desplegará en la ventana lo siguiente:



```
1.2.3.4 - Tera Term VT
File Edit Setup Control Window Help
#HELLO*CMD
set uart baudrate 460800
AOK
<4.00> set comm size 2
AOK
<4.00> █
```

el modulo tiene 14 canales para poder trabajar, posteriormente si se desea cambiar de canal en donde estará operando el módulo, se realizará con el siguiente comando: `set wlan channel 7`, seguido de la tecla enter, en este caso el numero indica el canal al cual se desea cambiar la configuración y sólo bastaría con poner un numero entre 1 y 14 para poder configurarlo, en este caso fue el canal 7. Finalmente, para guardar nuestra configuración escribimos `save`, seguido de la tecla enter.

Para salirnos de la configuración una vez salvando los datos escribimos el comando `reboot` seguido de la tecla enter y la ventana se cerrará automáticamente indicando que la configuración realizada se guardó correctamente.

Bibliografía

- [1] “Historia de los robots CNC”, junio 2017, disponible en :<http://www.fanuc.eu/es/es/quienes-somos/fanuc-historia>.
- [2] “Impresoras 3D: historia, cómo trabajan y algunas de las impresoras más populares”, noviembre 2016, disponible en: geeksroom.com/2014/02/impresoras-3d/82881.
- [3] “Que es un sistema de adquisición de datos”, octubre 2015, disponible en: <http://www.ni.com/data-acquisition/what-is/esa/>.
- [4] “Módulo WiFly RN-XV”, septiembre 2015, disponible en: <https://cdn.sparkfun.com/datasheets/Wireless/WiFi/rn-wiflycr-ug-v1.2r.pdf>
- [5] “Diseño de una PCB con altium Designer”, septiembre, 2015, disponible en: <http://www.altium.com/documentation/17.0/display/ADES/From+Idea+to+Manufacture+-+Driving+a+PCB+Design+through+Altium+Designer>.
- [6] Antonio Pertence Junior, “Amplificadores operaciones y Filtros Activos”, Prentice hall 2000.
- [7] Julio ortega lopera; mancia anguita lópez; alberto prieto espinosa, “Arquitectura de Computadoras”, Alfa Omega, 2003.
- [8] “Diseño de de PCB tips”, diciembre 2015, disponible en: <https://resources.altium.com/altium-audio-bites/3d-pcb-design-why-you-need-it-and-how-it-will-help-you-pcb-design-tips-tricks-altium-audio-bites>.
- [9] Giovanni Zepeda Arce, Tesis “Diseño de un microprocesador con arquitectura RISK de 32 bits”.
- [10] O. Arbelaitz Gallego, T. Ruiz Vázquez y A. Ibarra Lasa, “Análisis Básico de Circuitos Eléctricos y Electrónicos”, Pearson Educación, 2004.

[11] O. J. C. Trujillo, abril 2016, disponible en: http://es.slideshare.net/SharkCuellar/firmware-7143594?qid=509619a6-131d-48b8-a6f5-9103ea5bcc05&v=&b=&from_search=2.

[12] O. Arbeláez Gallego, T. Ruiz Vásquez y A. Ibarra Lasa, “Análisis Básico de Circuitos Eléctricos y Electrónicos”, Pearson Educación, S.A., 2004.

[13] ALTERA Quartus. “QuartusII software Suport”, octubre 2016, disponible en: <http://www.altera.com/support/software/sof-quartus.html>.

[14] “Manual de Labview”, octubre 2015, disponible en: <http://www.ni.com/getting-started/labview-basics/esa/>

[15] “¿Qué es un FPGA?”, diciembre 2015, disponible en: <http://www.ni.com/white-paper/6984/es/>.

[16] Francisco José López Cortes, Tesis “Desarrollo de un sistema mecatrónica para robot humanoide que permita emular el movimiento del cuello de los seres humanos”.

[17] Alfredo López Beltrán, Tesis “Robot CNC para impresión 3D”.

[18] Klaus Dembowski, “Gran libro del hardware”, Pearson Education 2001.

[19] SKLAR, Bernard, “Digital Communications Fundamentals and Applications”, Prentice Hall 2001.

[20] Fernando Reyes Cortes, “Robótica control de robots manipuladores”, alfa omega.