

UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO – CAMPUS SUR

CARRERA DE INGENIERÍA ELECTRÓNICA

MENCIÓN SISTEMAS INDUSTRIALES

**DISEÑO E IMPLEMENTACIÓN DEL CONTROL PID DE
VELOCIDAD PARA UN MOTOR DC UTILIZANDO LA TARJETA
FPGA SPARTAN 3E Y LABVIEW PARA LOS LABORATORIOS
DE LA CARRERA DE INGENIERÍA ELECTRÓNICA DE LA
UNIVERSIDAD POLITÉCNICA SALESIANA.**

**TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO
ELECTRÓNICO**

CHRISTOPHER FELIPE LEÓN GALARZA

DIRECTOR: Ing. Luis Oñate

Quito, Octubre 2011

DECLARACIÓN

Yo, Christopher Felipe León Galarza, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional, y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Politécnica Salesiana, según lo establecido por la Ley de Propiedad Intelectual por su reglamento y por la normatividad institucional vigente.

Christopher Felipe León Galarza

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Christopher Felipe León Galarza bajo mi dirección.

Ing. Luis Oñate
Director de Tesis

AGRADECIMIENTOS

Agradezco a Dios por haberme permitido culminar con éxito mi carrera universitaria, dándome fuerza en los momentos más difíciles de esta y ayudándome a dar cada paso que me ha llevado a la realización de esta meta.

Agradezco a mi familia que siempre creyó en mí y me impulso a siempre seguir adelante y no dejarme vencer por las dificultades, en especial mi madre a la cual le agradezco todo el amor, paciencia, cuidados y enseñanzas que me ha brindado desde el día en que nací, siendo para mí un ejemplo de integridad, esfuerzo y sacrificio, haciéndome cada día una mejor persona.

Agradezco a mi novia Dayanna que ha sido mi fiel compañera durante la mayoría de mi carrera, en la cual ha sabido apoyarme e impulsarme para sobresalir en todo lo que hago, haciéndolo siempre con mucho amor y comprensión.

Agradezco al ingeniero Luis Oñate que ha sido un gran apoyo como docente así como tutor del presente proyecto de titulación, ya que durante la realización de este me ha brindado importantes consejos y enseñanzas que me llevaron a realizarlo de la mejor manera posible.

Agradezco de igual manera a todas las personas que directa o indirectamente han contribuido positivamente en mi desarrollo profesional durante el transcurso de mis estudios de tercer nivel.

DEDICATORIA

Dedico este proyecto de titulación a mi madre Maritza ya que con el gran esfuerzo que realiza día a día, con la paciencia que ha sabido tener conmigo, con los innumerables sacrificios que ha tenido que afrontar, con el infinito amor que me brinda cada día y con todas esas maravillosas virtudes que caracterizan a esta excelente madre, es la principal responsable de que me encuentre en este punto de mi vida personal y profesional.

Siendo la autora de mi vida y el faro que me ha guiado en la penumbra, ha sabido dar el consejo justo y el abrazo necesario en los momentos que más lo he necesitado, siendo más que mi madre ya que también es mi gran amiga, capaz de sacrificarlo todo por la felicidad de sus hijos, a pesar de no ser yo el hijo que ella merece, cometiendo el grave error que cometemos muchos, el error de una inconsciente pero dolorosa ingratitud, por lo que le pido perdón a mi madre por las lagrimas que ha derramado por mi causa y los enojos que le he causado con mis acciones, claro que nunca haciendo algo con el afán de molestarla, más bien han sido mis descuidos u olvidos los causantes.

Por lo que sin más que decir por el momento, le prometo tratar siempre de apoyarla y ayudarla en lo que pueda, en agradecimiento por todo lo que me ha brindado durante mi vida, y recordándole que ella está siempre en mi mente y en mi corazón ya que es mi madrecita querida a la que yo amo con todo ser.

Gracias mamá.

CONTENIDO

CAPITULO 1. PLANTEAMIENTO DEL PROBLEMA.....	1
1.1. INTRODUCCIÓN.....	1
1.2. PROBLEMA A RESOLVER.....	1
1.3. HIPÓTESIS.....	2
1.4. TESIS.....	2
1.5. OBJETIVOS.....	2
1.5.1. OBJETIVO GENERAL.....	2
1.5.2. OBJETIVOS ESPECÍFICOS.....	3
1.6. JUSTIFICACIÓN.....	3
1.7. ALCANCES.....	3
1.8. METODOLOGÍA DE LA INVESTIGACIÓN.....	4
1.8.1. INVESTIGACIÓN BIBLIOGRÁFICA.....	4
1.8.2. DESARROLLO DE HARDWARE Y SOFTWARE.....	5
1.8.3. CONSTRUCCIÓN FINAL.....	5
1.8.4. DOCUMENTACIÓN.....	5
 CAPITULO 2. ESTADO DEL ARTE.....	 6
2.1. INTRODUCCIÓN.....	6
2.2. ESTADO DEL ARTE.....	6
2.3. MARCO TEÓRICO.....	7
2.3.1. PID CON LABVIEW.....	7
2.3.2. MÓDULO LABVIEW FPGA.....	7
2.3.3. CONTROL PID DE UN MOTOR DC.....	11
 CAPITULO 3. DESARROLLO DE HARDWARE Y SOFTWARE.....	 16
3.1. INTRODUCCIÓN.....	16
3.2. LINEALIDAD DEL SISTEMA.....	16
3.3. OBTENCIÓN DE LA PLANTA DEL SISTEMA.....	18
3.3.1.FUNCIÓN DE TRANSFERENCIA DEL MOTOR DC.....	19

3.3.2.	FUNCIÓN DE TRANSFERENCIA DEL CONTROL PID.....	23
3.3.3.	FUNCIÓN DE TRANSFERENCIA DEL SISTEMA EN LAZO ABIERTO.....	25
3.3.4.	FUNCIÓN DE TRANSFERENCIA DEL SISTEMA EN LAZO CERRADO.....	25
3.4.	DISEÑO Y DESARROLLO DE HARDWARE.....	27
3.4.1.	DIAGRAMA DE BLOQUES.....	27
3.4.2.	DESCRIPCIÓN DE LOS BLOQUES DE HARDWARE.....	28
3.4.2.1.	ADAPTADOR DE VOLTAJE.....	28
3.4.2.2.	SENSOR DE VELOCIDAD.....	29
3.4.2.3.	INTERFAZ FPGA-MOTOR DC.....	31
3.4.3.	DIAGRAMA ESQUEMÁTICO COMPLETO.....	34
3.5.	DESARROLLO DE SOFTWARE.....	35
3.5.1.	DIAGRAMA DE FLUJO GENERAL.....	35
3.5.2.	DIAGRAMA DE FLUJO DE LA CONFIGURACIÓN DEL FPGA.....	37
3.5.3.	DIAGRAMA DE FLUJO DE LA CONFIGURACIÓN DEL PWM.....	39
3.5.4.	DIAGRAMA DE FLUJO DE LA MEDICIÓN DE VELOCIDAD.....	41
3.5.5.	DIAGRAMA DE FLUJO DEL CONTROL PID.....	42
3.5.6.	ETAPAS DEL SOFTWARE DESARROLLADO EN LABVIEW.....	44
3.5.6.1.	CONFIGURACIÓN DEL FPGA.....	44
3.5.6.2.	GENERACIÓN PWM.....	47
3.5.6.3.	MEDICIÓN DE VELOCIDAD.....	49
3.5.6.4.	CONTROL PID.....	50
3.5.7.	DIAGRAMA DE BLOQUES DEL PROGRAMA COMPLETO.....	53
3.5.8.	PANEL DE INICIO DEL SOFTWARE.....	53
3.5.9.	PANEL FRONTAL DEL SOFTWARE.....	54
3.6.	CONSTRUCCIÓN DEL MÓDULO.....	54
3.6.1.	ELABORACIÓN DE LA PLACA DE INTERFAZ FPGA-MOTOR DC.....	54
3.6.2.	ELABORACIÓN DE LA ESTRUCTURA.....	56
3.6.3.	MONTAJE DE LA PLACA.....	59
3.6.4.	MONTAJE DEL MOTOR DC.....	59
3.6.5.	MONTAJE DEL LECTOR DE VELOCIDAD.....	60
3.6.6.	MÓDULO COMPLETO.....	61
3.7.	COSTOS DEL PROYECTO DE INVESTIGACIÓN.....	62
3.7.1.	COSTOS DE MATERIALES.....	62

3.7.1.1. ESTRUCTURA.....	62
3.7.1.2. HARDWARE.....	63
3.7.2. COSTOS DE DISEÑO DE HARDWARE.....	64
3.7.3. COSTOS DE DESARROLLO DE SOFTWARE.....	64
3.7.4. MANO DE OBRA.....	64
3.7.5. COSTO TOTAL DEL EQUIPO.....	64
4. CAPITULO 4. ANÁLISIS DE RESULTADOS.....	66
4.1. INTRODUCCIÓN.....	66
4.2. PRUEBAS POR SET POINT.....	66
4.3. CURVA DE CALIBRACIÓN.....	69
5. CAPITULO 5. CONCLUSIONES Y RECOMENDACIONES.....	72
5.1. CONCLUSIONES.....	72
5.2. RECOMENDACIONES.....	74
BIBLIOGRAFÍA.....	75
NETGRAFÍA.....	76
ANEXOS.....	77

RESUMEN

El proyecto de tesis “*Diseño E Implementación Del Control PID De Velocidad Para Un Motor DC Utilizando La Tarjeta FPGA SPARTAN 3E Y LabVIEW Para Los Laboratorios De La Carrera De Ingeniería Electrónica De La Universidad Politécnica Salesiana*” es un proyecto de investigación cuyo objetivo central es utilizar las capacidades del software de instrumentación virtual LabVIEW de National Instruments, en conjunto con la tarjeta FPGA SPARTAN 3E de Xilinx para lograr implementar un módulo didáctico que este a disposición de los estudiantes y docentes en los laboratorios de ingeniería electrónica de la Universidad Politécnica Salesiana, con el cual puedan realizar estudios y experimentos con el control PID de procesos, específicamente sobre la velocidad de motores DC el cual es el caso del presente proyecto, esto será un gran aporte permitiendo reforzar los conocimientos teóricos que se reciben en las clases referentes a la teoría del control, y así permitir a los estudiantes una mejor comprensión del tema para que sean capaces de implementar este tipo de control de una manera práctica y funcional sobre cualquier proceso en que se necesite implementar un control automático de este tipo, además de esto este proyecto, al tener su núcleo de funcionalidad dentro del software LabVIEW y la tarjeta FPGA SPARTAN 3E, brinda la posibilidad ser modificado por los estudiantes posteriormente para implementar otros tipos de control más precisos o mejorar el control ya existente reforzando las distintas materias que comprenden la carrera de ingeniería electrónica.

El presente tema también permite ampliar los conocimientos sobre la programación en el software LabVIEW el cual al tener una gran gama de funcionalidades y campos de aplicación se vuelve una importante herramienta para el ingeniero electrónico en diversos campos de la investigación y la industria. Como se aprecia en este proyecto todos los procesos dentro del FPGA se realizan de una manera paralela e independiente permitiendo tener varias funcionalidades simultaneas en un solo dispositivo, igualmente los estudiantes y demás lectores de este proyecto aprenderán las bases sobre la programación de los FPGAs desde el software de instrumentación LabVIEW en relación al control

de motores DC, ya que estos poseen variadas herramientas para realizar esto de una manera muy fácil.

En este proyecto se revisa conceptos sobre la elaboración de circuitos de control de potencia ya que incluye la elaboración de un hardware, el cual sirve de interfaz entre la tarjeta FPGA y el motor DC incluyendo un circuito optoacoplado de conmutación mediante mosfet para regular la velocidad del motor DC mediante modulación de ancho de pulso PWM el cual proviene de la tarjeta FPGA SPARTAN 3E; este proyecto incluye un lector de velocidad que consiste en un optointerruptor con función de encoder el cual enviará pulsos al FPGA el cual, en conjunto con el software LabVIEW, detectará la velocidad del motor DC de acuerdo a la frecuencia de estos pulsos, con lo que el sistema de control PID de velocidad tendrá la realimentación necesaria.

Este proyecto básicamente sensa los pulsos, los cuales son adquiridos por una entrada digital del FPGA, este pre procesa los datos y envía los resultados al PC donde son procesados por el software LabVIEW teniendo un valor numérico de la velocidad del motor, de acuerdo a esta velocidad se realiza en LabVIEW el control PID para posteriormente generar la señal de PWM, esta señal se envía a la tarjeta FPGA SPARTAN 3E y esta las envía, por medio de una salida digital, a la tarjeta de control de potencia la cual varía la velocidad de giro del motor DC de acuerdo a la señal entrante.

PRESENTACIÓN

En este escrito se detallarán las distintas etapas tanto teóricas como prácticas de las cuales está compuesto el desarrollo del presente proyecto de investigación cuyo tema es el *“Diseño E Implementación Del Control PID De Velocidad Para Un Motor DC Utilizando La Tarjeta FPGA SPARTAN 3E Y LabVIEW Para Los Laboratorios De La Carrera De Ingeniería Electrónica De La Universidad Politécnica Salesiana”*, con lo que se podrá comprender de una mejor manera cada una de las partes que lo componen.

En el capítulo 1 se puede apreciar la situación de la que surge la necesidad de la elaboración de este proyecto, analizando el problema que se desea resolver al elaborarlo, para lo cual se plantea la respectiva hipótesis y tesis las cuales llevan al planteamiento del objetivo principal, objetivos específicos y alcances que se busca cumplir durante la elaboración de este proyecto, el cual se desarrolla según la metodología de la investigación presente en este capítulo.

El capítulo 2 consta del estado del arte en el cual se revisan los escritos del momento histórico actual en relación a proyectos sobre motores DC, FPGA SPARTAN 3E y LabVIEW mediante PID.

En el capítulo 3 se encuentra documentado todo el desarrollo del presente proyecto de investigación tanto en hardware como en software. Se encuentra realizado el cálculo de la planta y el cálculo del control PID del sistema. Para el análisis del hardware se realizó un diagrama de bloques de las distintas etapas, las cuales se analizan posteriormente, en el caso del software se realizaron los correspondientes diagramas de flujo en los que se puede observar su estructura de funcionamiento, y de la misma manera las etapas presentes en los diagramas de flujo del software se analizan a fondo posteriormente, en este capítulo también se detalla la construcción del módulo de hardware y los costos tanto de investigación como de desarrollo Y diseño del presente proyecto.

En el capítulo 4 se pueden observar las distintas pruebas que se realizaron sobre el presente proyecto para verificar su correcto funcionamiento y el análisis de error del control de velocidad sobre el motor DC.

El capítulo 5 corresponde a conclusiones y recomendaciones obtenidas del desarrollo del presente proyecto.

De la misma manera existe la bibliografía que sustenta la base teórica del presente proyecto.

En los anexos se pueden encontrar las características principales del módulo LabVIEW-FPGA, características de la tarjeta FPGA SPARTAN 3E, hojas de datos de los elementos utilizados en el hardware, cálculos matemáticos en matlab y un tutorial sobre la elaboración del presente proyecto.

CAPITULO 1. PLANTEAMIENTO DEL PROBLEMA.

1.1. INTRODUCCIÓN

En este capítulo se analiza y diagnostica el problema a resolver, para lo cual se planteará la respectiva hipótesis y tesis.

Basándose en lo anterior se plantearán los objetivos, la justificación, los alcances y la metodología de investigación para su diseño e implementación.

1.2. PROBLEMA A RESOLVER

En los laboratorios de la carrera de ingeniería electrónica de la Universidad Politécnica Salesiana existe la carencia de herramientas para la enseñanza de la Teoría del Control de una manera práctica en función de las capacidades que posee el software de instrumentación virtual LabVIEW para operar la FPGA SPARTAN 3E, por lo que los conocimientos que reciben los estudiantes al ser puramente teóricos tanto en el estudio de la teoría del control como en el aprendizaje de la instrumentación, específicamente el software de instrumentación virtual LABVIEW en conjunto con la FPGA SPARTAN 3E, quedan incompletos.

Esto se convierte en un gran problema debido a que la teoría del control proporcional integral diferencial (PID) sobre todo aplicada a los motores DC, es de suma importancia para los procesos industriales que encontramos hoy en día ya que la tendencia es que estos se ejecuten de manera automática y para poder hacerlo con gran calidad y seguridad necesitan tener una gran precisión en sus diferentes tareas, ya sea que se controle su posición angular o su velocidad, cada vez se necesita que el control automático de estos parámetros sea lo más rápido y preciso posible lo que se consigue aplicando el control PID, el cual se ocupa de que la regulación de velocidad o posición en motores DC sea lo más rápida y precisa posible, y para poder aplicar este tipo de control sobre procesos reales se necesita que durante los estudios de Teoría de Control se brinden los suficientes fundamentos teóricos y sobre todo prácticos de el control

PID para que los estudiantes y futuros ingenieros estén en capacidad de implementar este tipo de control y saber cuándo y en qué proceso se necesita realizarlo.

Por lo que aparece la siguiente pregunta:

¿Cómo se puede diseñar e implementar un control PID de velocidad para un motor DC utilizando la tarjeta de desarrollo FPGA SPARTAN 3E de Xilinx y el software de instrumentación LabVIEW?

1.3. HIPÓTESIS

La hipótesis es la siguiente:

Es posible realizar el control PID de un motor DC utilizando para esto el software de instrumentación virtual LabVIEW de National Instruments en complemento con la tarjeta de desarrollo FPGA SPARTAN 3E de Xilinx.

1.4. TESIS

La tesis es la siguiente:

Desarrollo e implementación de el control PID de un motor DC de 30Vcc con velocidad nominal de 3308 rpm, utilizando el software de desarrollo LabVIEW y la tarjeta FPGA SPARTAN 3E.

1.5. OBJETIVOS

A continuación se describen el objetivo general y los objetivos específicos de este proyecto.

1.5.1. OBJETIVO GENERAL

Diseñar e implementar un control PID en el software de instrumentación virtual LABVIEW para controlar la velocidad de un motor DC a través de la tarjeta FPGA SPARTAN 3E.

1.5.2. OBJETIVOS ESPECÍFICOS

- Reconocer y controlar la tarjeta FPGA SPARTAN 3E desde LABVIEW.
- Utilizar la tarjeta FPGA SPARTAN 3E como tarjeta de adquisición de datos y visualizar estos en LABVIEW.
- Controlar las salidas digitales de la tarjeta FPGA SPARTAN 3E desde LABVIEW.
- Adquirir y controlar la velocidad de un motor DC desde LABVIEW a través de la tarjeta FPGA SPARTAN 3E.
- Realizar un control PID en LABVIEW de la velocidad del motor DC.

1.6. JUSTIFICACIÓN

Con la implementación de este proyecto en los laboratorios de Ingeniería electrónica se tendrá un aporte muy significativo para los estudiantes en el aprendizaje de la teoría del control ya que podrán contar con una aplicación práctica del control PID, lo que les ayudará a entenderlo mejor y por ende aprender a implementarlo sobre procesos reales.

De la misma manera es un gran aporte en el aprendizaje tanto de la programación en el software de desarrollo LabVIEW, así como la comunicación de la tarjeta FPGA SPARTAN 3E con este software.

Esta investigación es importante porque permite mejorar la calidad de las herramientas de enseñanza de la carrera, mediante los equipos de laboratorio existentes.

1.7. ALCANCES

EL control PID del motor DC mediante FPGA SPARTAN 3E realiza lo siguiente:

- Este proyecto permitirá controlar la velocidad de un motor DC de 30V desde el software LabVIEW.
- En LabVIEW se fija el “Set Point” de la velocidad deseada la cual en el caso de este motor llega a máximo 3000 rpm.

- En LabVIEW se genera el PWM que controlará la velocidad del motor DC.
- En LabVIEW se observa la forma de onda que produce el transitorio y la estabilización de esta.
- La conexión con el motor se realizará a través de la tarjeta FPGA SPARTAN 3E.
- De la tarjeta FPGA SPARTAN 3E se utilizarán una entrada y una salida digital para la adquisición de la velocidad del motor y control de esta por medio de PWM respectivamente.

Este proyecto no realiza lo siguiente:

- No se podrá controlar la posición angular del motor DC.

1.8. METODOLOGÍA DE LA INVESTIGACIÓN

En este apartado se detallarán los pasos que se han planteado para la realización del presente proyecto comenzando por la investigación bibliográfica, seguida del desarrollo de hardware y software, y finalmente la construcción final y documentación del proyecto.

1.8.1. INVESTIGACIÓN BIBLIOGRÁFICA

Se realizará una investigación sobre el control PID y sus cálculos para aprender a implementarlo de una manera práctica sobre un proceso.

Posteriormente se investigarán las herramientas con las que cuenta el software LabVIEW para la realización del control PID y su correcta utilización.

También se investigarán las características y utilización de la tarjeta FPGA SPARTAN 3E para finalmente investigar la conexión y utilización de esta con el software LabVIEW.

1.8.2. DESARROLLO DE HARDWARE Y SOFTWARE

Se diseñará y construirá un acoplamiento óptico para la conexión del motor DC con la tarjeta FPGA SPARTAN 3E debido a que estos no trabajan con el mismo nivel de voltaje.

Se realizarán aplicaciones básicas en LabVIEW que nos permitan probar la correcta comunicación entre este y la tarjeta FPGA SPARTAN 3E así como comprobar que podemos controlar sus entradas y salidas digitales desde LabVIEW.

Se desarrollará una aplicación en LabVIEW que permita adquirir y visualizar la velocidad del motor DC en base a su conexión con la tarjeta FPGA SPARTAN 3E.

Posteriormente se implementará en esta aplicación de LabVIEW un control de velocidad sencillo que nos permita controlar esta a través de la tarjeta FPGA SPARTAN 3E.

Finalmente se implementará un control PID en esta aplicación utilizando las herramientas que tiene LabVIEW para este fin.

1.8.3. CONSTRUCCIÓN FINAL

Se harán las conexiones finales del motor hacia la tarjeta FPGA SPARTAN 3E y se realizarán las pruebas correspondientes para comprobar y afinar el control PID de velocidad sobre el motor DC.

1.8.4. DOCUMENTACIÓN

Se escribirá el texto de la monografía correspondiente al presente proyecto de investigación, cuyo tema es el control del motor DC mediante la tarjeta FPGA SPARTAN 3E y el software de desarrollo LabVIEW.

CAPITULO 2. ESTADO DEL ARTE

2.1. INTRODUCCIÓN

En este capítulo se analizará el estado del arte el cual consiste en los proyectos que son relativamente cercanos al tema del presente proyecto, además se presentarán los fundamentos teóricos en los que se basa el diseño e implementación del control PID de velocidad de un motor DC desde LabVIEW a través de la tarjeta FPGA SPARTAN 3E.

2.2. ESTADO DEL ARTE

En la actualidad no se encuentra publicado ningún proyecto el cual consista en el control PID de velocidad de un motor DC desde LabVIEW a través de la tarjeta FPGA SPARTAN 3E, sin embargo existen varios proyectos usando solo dos de estos tres elementos como:

- LabVIEW- FPGA SPARTAN 3E
- FPGA SPARTAN 3E- motor DC

Por ejemplo:

- *Control de lazo cerrado para un motor usando FPGA SPARTAN 3E*, implementado por BOGDAN ALECSA y ALEXANDRU ONEA en la Universidad Técnica "Gheorghe Asachi" de Iasi, Rumania. ^[12]

Este proyecto se trata del diseño de un sistema realimentado para el control de un motor basándose en la tecnología FPGA para realizar las operaciones lógicas y matemáticas en este caso utilizando la tarjeta FPGA SPARTAN 3E.

- *Speed Control of Switched Reluctance Motor Based on Fuzzy Logic Controller*, implementado por Gamal M. Hashem y Hany M. Hasanien en el Department of Electrical Power and Machines, Ain Sham University, Cairo, Egypt. ^[13]

Este proyecto se trata del control de velocidad de un motor por medio de lógica difusa aplicada a la modulación PWM mediante la tarjeta FPGA SPARTAN 3E.

2.3. MARCO TEÓRICO

2.3.1. PID CON LABVIEW

LabVIEW cuenta con una serie de herramientas destinadas a realizar el control PID sobre un proceso de una manera relativamente fácil, sobre todo debido a su programación gráfica usando bloques, por lo que estas herramientas para el control PID solo deben ser arrastradas e interconectadas en el diagrama de bloques para realizar un control eficiente sobre el proceso que se tenga.

Estas herramientas pueden ser encontradas en la lista de funciones del diagrama de bloques, existen herramientas desde un simple control PID hasta un PID que autorregula sus parámetros.

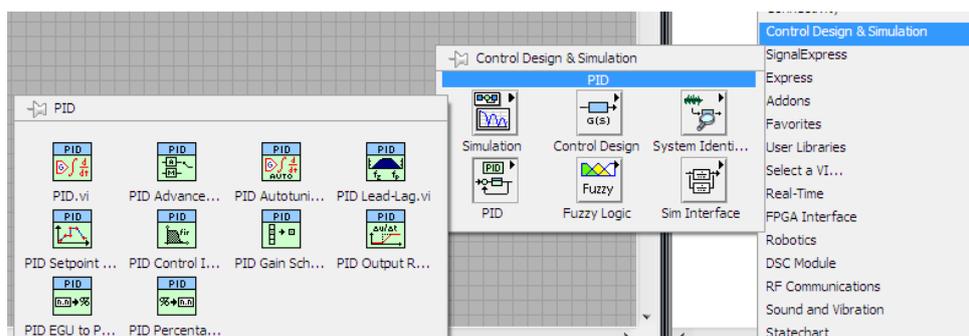


FIGURA 1. FUNCIONES PID DE LABVIEW.¹

2.3.2. MÓDULO LABVIEW-FPGA

El módulo LabVIEW-FPGA pertenece al software LabVIEW el cual por su orientación a trabajar industrialmente, posee varios

¹ Captura del software LabVIEW 2010

componentes de software que permiten a los usuarios programar hardware acoplado con LabVIEW.

LabVIEW al ser totalmente compatible con la tarjeta FPGA SPARTAN 3E permite programarla e interactuar con ella desde el mismo software ya que cuenta con herramientas para controlarla, además de un traductor y un compilador los cuales transforman la programación de manera gráfica en un código entendible para el FPGA y lo transfieren este.

Para poder conseguir esto, National Instruments que es la empresa creadora de LabVIEW ha creado diferentes herramientas para que con ayuda de este módulo se pueda tomar los diagramas de LabVIEW y transformarlos a través de diferentes compiladores dependiendo del hardware en el que se va a efectuar la ejecución, en este caso el FPGA.

El módulo LabVIEW-FPGA permite compilar el código que se programa en LabVIEW para luego transformarlo a archivos de bits y transferirse a un FPGA y ejecutarse en él.

La programación gráfica de LabVIEW es de gran ayuda para programar los FPGAs ya que el programa de LabVIEW se asemeja mucho a la estructura del programa del FPGA, simplemente se necesita programar un sistema en paralelo que describa el comportamiento del circuito que se desea implementar en el chip. No hay ningún otro lenguaje de programación de hardware en el que se pueda visualizar la programación en paralelo del FPGA tan claramente como en el lenguaje grafico que posee LabVIEW.

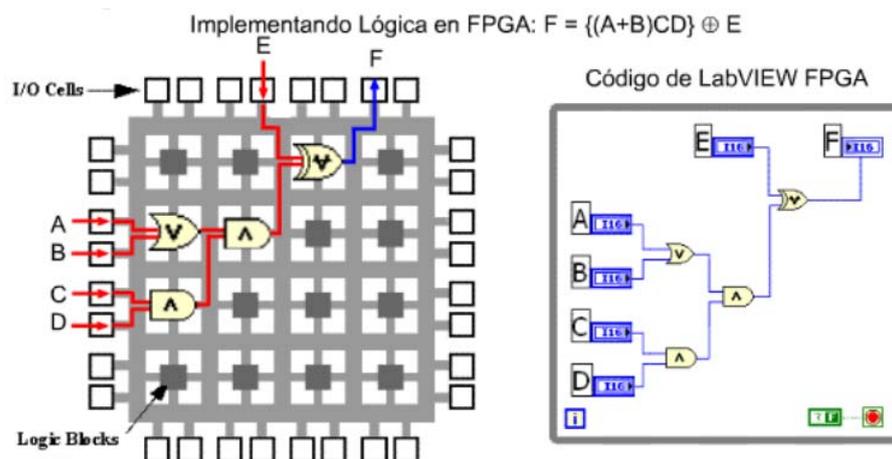


FIGURA 2. ESTRUCTURA DE PROGRAMA DEL FPGA COMPARADA CON LA ESTRUCTURA DE PROGRAMA EN LABVIEW.²

Las aplicaciones más comunes del módulo LabVIEW-FPGA son las siguientes:

- *Control a alta velocidad:* las velocidades de ejecución que se pueden lograr utilizando FPGAs son muy altas las cuales pueden ir hasta 200KHz, el módulo LabVIEW-FPGA posee varias paletas de funciones de algoritmos integrados para implementarlos en el FPGA como controles PID o filtros que son comunes en cualquier sistema de control.
- *Adquisición de datos inteligente:* el módulo LabVIEW-FPGA permite acondicionar las entradas o salidas de modo que ejecuten procesos o respondan a los eventos que se necesiten en el sistema permitiendo crear contadores, modulación PWM, control PID, u otros en la misma tarjeta.
- *Protocolos de comunicación digital:* estos protocolos pueden ser estándares como i2c, SPI, RS232 o protocolos internos específicos para cada aplicación que se realice, ya que con la

² Captura del video "Introducción al Módulo de LabVIEW FPGA", zone.ni.com/wv/app/doc/p/id/wv-609

ayuda de LabVIEW se tiene múltiples formas de diseñar los diferentes protocolos digitales.

- *Simulación de sensores:* en muchas áreas distintas poder simular sensores es útil y muy importante, específicamente en pruebas de hardware, con esto se evita conectar el prototipo real para las pruebas con lo que se ahorra tiempo y recursos.

- *Procesamiento en tarjeta y reducción de datos:* permite reducir la carga de trabajo sobre el procesador y los buses de comunicación de la PC pre-procesando los datos en el FPGA, de esta forma se adquieren todas las diferentes señales y antes de enviarlos a la computadora para su procesamiento posterior como lo haría una tarjeta de adquisición de datos se puede trabajar utilizando el FPGA para hacer las operaciones necesarias como codificación o decodificación, filtros, u otros, los cuales van a ser pesados para el CPU, de esta forma se tienen dos procesadores, haciendo un pre-procesamiento en el FPGA y enviando los resultados finales a la computadora, con esto se puede tener al CPU ejecutando operaciones más intensivas las cuales sea necesario que las realice el mismo.

- *Coprociamiento:* se tiene una comunicación constante de dos vías entre el FPGA y el CPU donde cada uno ejecutara ciertos procesos y enviará los resultados al otro.

Gustavo Valdés de National Instruments dice lo siguiente en el video “Introducción al Módulo de LabVIEW FPGA”: *El módulo de LabVIEW-FPGA permite crear código FPGA en LabVIEW, lo cual es muy importante porque permite representar el paralelismo y el flujo de datos permitiendo diseñar sistemas de una forma gráfica y aprovechar este nuevo nivel de abstracción en lenguajes para descripción de hardware.*^[14]

La facilidad que brinda LabVIEW es que solo se necesita arrastrar e interconectar los componentes del FPGA los cuales aparecen enumerados y listos para usar, junto con las funciones matemáticas, lógicas o del tipo que se necesite y se tendrá listo el programa sin necesidad de saber programar en lenguaje VHDL.

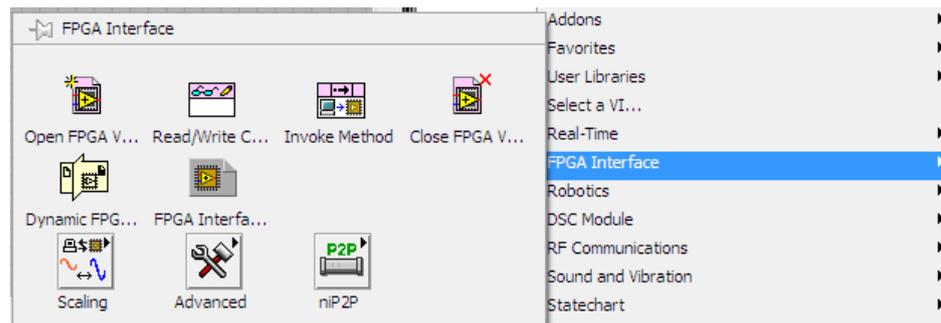


FIGURA 3. FUNCIONES FPGA DE LABVIEW.³

La compatibilidad de LabVIEW con el FPGA y sus herramientas de programación y control de esta, resultan muy útiles para crear un control que no se quede solo en el software o que nos implique adquirir o fabricar otro tipo de hardware el cual puede no tener las mismas facilidades de programación y control que tiene el FPGA.

2.3.3. CONTROL PID DE UN MOTOR DC

En gran parte de los procesos industriales, robóticos y aeroespaciales, el efector final de un sistema es un motor el cual necesita un control automático ya sea sobre su posición angular como en su velocidad.

Debido a esto el control PID es el más elegido para los sistemas complejos que requieren velocidad y precisión en su control ya sea para reducir riesgos o incrementar la productividad del sistema.

³ Captura del software LabVIEW 2010

Para crear un control PID sobre un motor DC se necesita realizar una serie de pasos para identificar como quedará estructurado el sistema de control del motor.

Lo primero que se necesita hacer en un modelamiento matemático del mismo para establecerlo como una planta, debido a esto se necesita conocer cuál es variable que se va a controlar ya que puede ser la posición angular o la velocidad como es el caso del presente proyecto.

Sabiendo esto se debe saber cómo se desea controlar esta variable ya que esto nos proporcionará a su vez la variable de control la cual puede ser voltaje, corriente o PWM el cual es el caso del presente proyecto.

El PWM o modulación de ancho de pulso consiste en variar el *ton* (tiempo de encendido) de una señal cuadrada conservando su periodo, esto se lo hace variando el porcentaje del periodo para el que la señal está dando un pulso de voltaje, a este porcentaje se lo llama *Duty Cycle*, el cual también dará el voltaje average de la señal multiplicando este porcentaje por el valor de voltaje de la onda, con lo que si se disminuye el *ton* de la señal aumentará el *toff* (tiempo de apagado), como se puede ver en la figura 4.

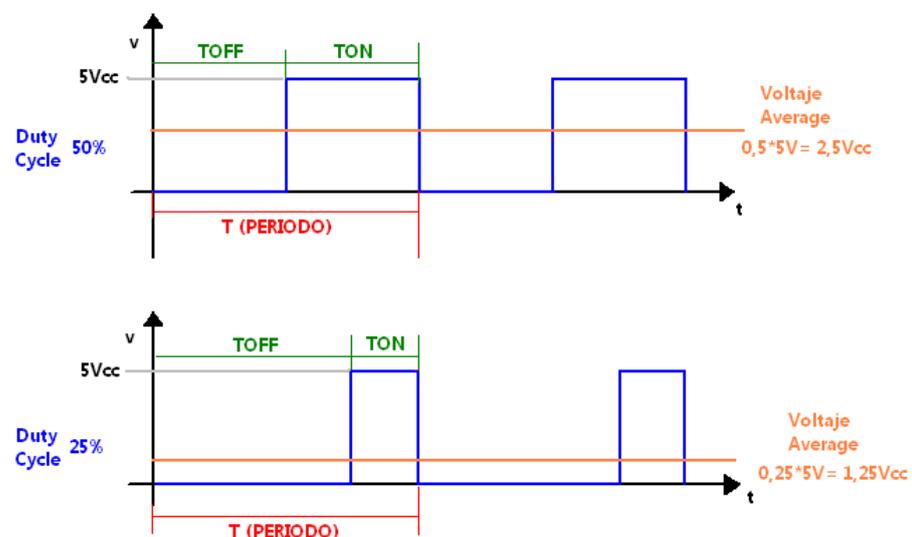


FIGURA 4. VARIACIÓN DEL PWM.

Lo que se consigue al hacer esto es variar el voltaje promedio o average de la señal lo cual modificará la velocidad de rotación del motor DC.

De la misma manera se debe medir la velocidad del motor para que realmente al sistema y este pueda efectuar el control, esto se puede hacer de distintas maneras, en el presente proyecto se realizará de una manera digital por medio de un lector óptico el cual se detallará en el siguiente capítulo.

Para modelar un sistema el cual en este caso es un motor existen diversos métodos, uno se basa en las características mecánicas y eléctricas del motor las cuales llevan a una ecuación que predice su funcionamiento, y la otra opción es realizar una serie de mediciones que nos lleven a tener la gráfica y posterior ecuación de la respuesta del motor vs una variable de control, lo que permite transformar el “Set Point” la cual es la velocidad deseada en términos de la variable de control la cual en este caso es el Duty Cycle del PWM.

Con lo que el modelamiento del motor en este proyecto se realizará de la manera experimental ya que no se poseen los datos del motor necesarios para hacerlo en base a sus características. Para esto se realizará una serie de mediciones de la velocidad del motor vs el Duty Cycle aplicado, con lo que se tendrá la ecuación de respuesta del motor.

Como se está trabajando con variables digitales, el control PID también será un control digital (discreto) por lo que se tiene las siguientes ecuaciones para el control PID discreto de velocidad.

$$e(k) = (\text{valor del set point} - \text{valor medido}) \rightarrow \text{error}$$

$$CP(k) = Kp * e(k) \rightarrow \text{control proporcional}$$

$$CI(k) = e(k) * Ki + CI(k - 1) \rightarrow \text{control integral}$$

$$CD(k) = Kd(e(k) - e(k - 1)) \rightarrow \text{control derivativo}$$

$$PID(k) = P(k) + I(k) + D(k) \rightarrow \text{control PID}$$

Donde:

- K_p = constante proporcional
- K_i = constante integral
- K_d = constante derivativa
- $e(k)$ = error actual
- $e(k-1)$ = error del periodo anterior
- $CP(k)$ = control proporcional
- $CI(k)$ = control integral
- $CI(k-1)$ = control integral del periodo anterior
- $CD(k)$ = control derivativo

Por lo que se puede apreciar que la ecuación teórica del control PID de un motor DC parte de la identificación del error del sistema y la aplicación de constantes y operaciones matemáticas que compensen este error, estas constantes parten de cada uno de los controles individuales, lo cuales combinados forman el control PID.

El control PID elimina el error en estado estacionario (error que permanece después de que ha desaparecido el transitorio) y minimiza el tiempo transitorio logrando una buena estabilidad en el sistema, además de un sobreimpulso pequeño.

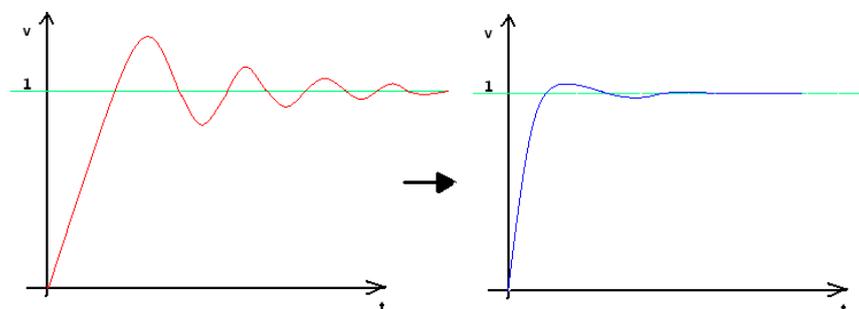


FIGURA 5. ACCIÓN DEL CONTROL PID SOBRE EL MOTOR DC.

Con todos estos pasos realizados se aprecia que el sistema queda estructurado según lo indica la figura 6.

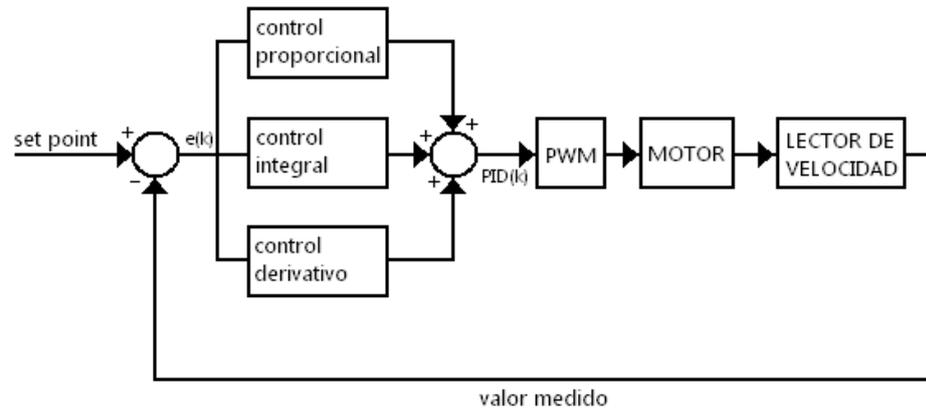


FIGURA 6. ESTRUCTURA DEL SISTEMA.

CAPITULO 3. DESARROLLO DE HARDWARE Y SOFTWARE.

3.1. INTRODUCCIÓN

En este capítulo se analizará la linealidad del sistema y se obtendrá la función de transferencia que describe al sistema del presente proyecto, posteriormente se analizará el diseño e implementación del hardware, el cual consiste en el circuito que cumple la función de interfaz entre el FPGA y el motor DC, permitiendo controlar y medir la velocidad de este motor a través de la tarjeta FPGA, del mismo modo se analizará el desarrollo del software en LabVIEW el cual permitirá realizar el control PID y visualizar su efecto mediante las formas de onda correspondientes para este caso.

3.2. LINEALIDAD DEL SISTEMA

La linealidad del sistema se refiere a que la salida del sistema debe ser proporcional a la entrada por lo que la respuesta del sistema debe ser del tipo:

$$y(t) = k \times u(t)$$

Se realizó un análisis de la linealidad del sistema para saber en qué región se comporta de manera lineal, para lo cual se hicieron pruebas en lazo abierto variando el duty cycle del PWM y midiendo con un tacómetro la velocidad del motor.

DUTY CYCLE	VELOCIDAD RPM
0	0
5	274,8
10	822,8
15	1311
20	1697
25	2017
30	2234
35	2408
40	2538

45	2634
50	2711
55	2771
60	2818
65	2862
70	2891
75	2923
80	2950
85	2972
90	2992
95	3018
100	3087

TABLA 1. MEDICIONES DUTY CYCLE vs VELOCIDAD EN LAZO ABIERTO.

Con lo que se obtuvo la gráfica de la figura 7.

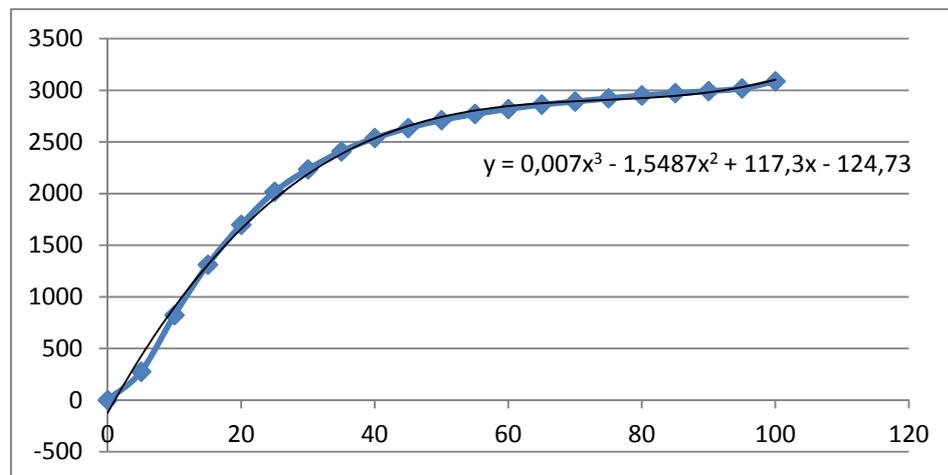


FIGURA 7. DUTY CYCLE vs VELOCIDAD.

Gráficamente se aprecia que las regiones más lineales del sistema están del 5% al 25% y del 50% al 80%.

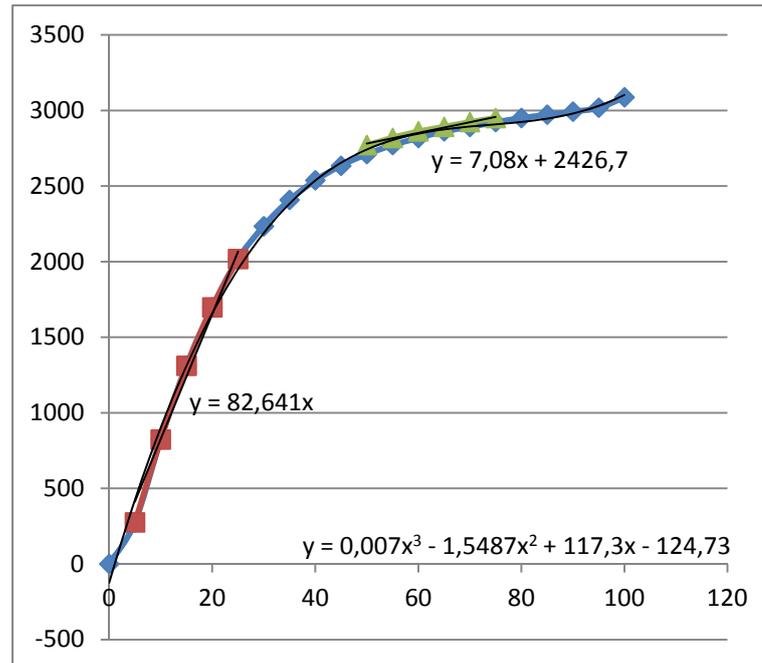


FIGURA 8. REGIÓN LINEAL DEL SISTEMA.

3.3. OBTENCIÓN DE LA PLANTA DEL SISTEMA

Para calcular la planta del sistema del presente proyecto se deben obtener las funciones de transferencia discretas del motor DC y del control PID ya que la ganancia de realimentación es 1.

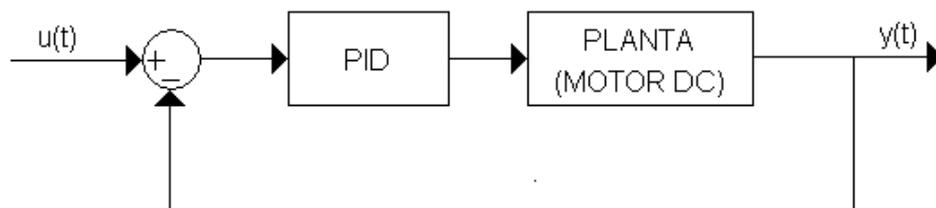


FIGURA 9. SISTEMA DEL PROYECTO.

3.3.1. FUNCIÓN DE TRANSFERENCIA DEL MOTOR DC

La función de transferencia del motor DC del presente proyecto se obtuvo de manera experimental ya que no se cuenta con los parámetros de fabricación necesarios para obtenerla de manera teórica.

Para esto se registró la respuesta del motor DC a un escalón de 30 Vdc el cual es el voltaje nominal del motor DC de este proyecto, con lo que se obtuvo la siguiente tabla:

TIEMPO	VELOCIDAD RPM
06/07/2011 05:06:46,281 PM	0
06/07/2011 05:06:46,375 PM	1827,95565
06/07/2011 05:06:46,484 PM	3143,72817
06/07/2011 05:06:46,578 PM	3321,36
06/07/2011 05:06:46,672 PM	3279,843
06/07/2011 05:06:46,781 PM	3285,774
06/07/2011 05:06:46,875 PM	3327,291
06/07/2011 05:06:46,984 PM	3267,981
06/07/2011 05:06:47,078 PM	3327,291
06/07/2011 05:06:47,172 PM	3279,843
06/07/2011 05:06:47,281 PM	3303,567
06/07/2011 05:06:47,375 PM	3315,429
06/07/2011 05:06:47,484 PM	3315,429
06/07/2011 05:06:47,578 PM	3315,429
06/07/2011 05:06:47,672 PM	3273,912

TABLA 2. REGISTRO DE LA RESPUESTA DEL SISTEMA AL ESCALÓN.

De esta tabla se obtuvo la gráfica de la respuesta del motor al escalón.

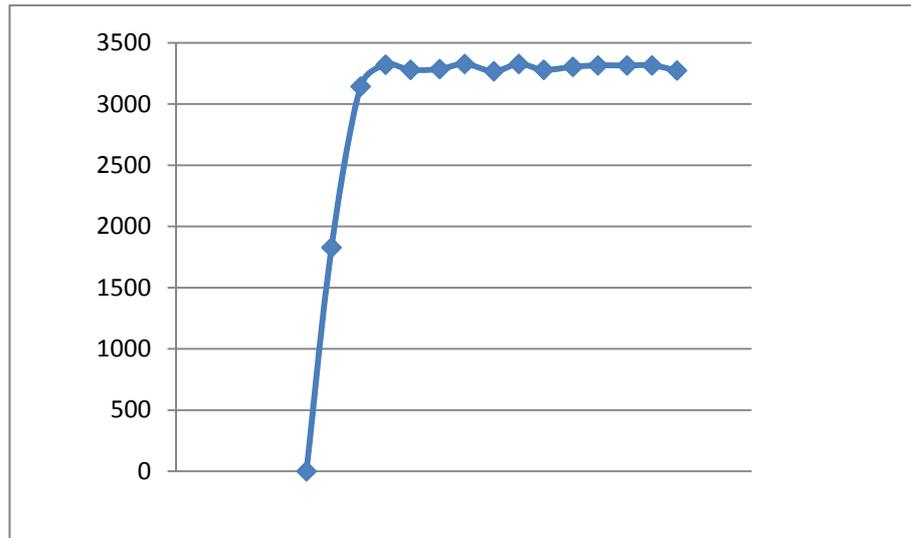


FIGURA 10. GRÁFICA DE RESPUESTA AL ESCALÓN.

De los registros obtenidos del proceso experimental se obtuvieron los siguientes datos:

- *Voltaje nominal: 30Vdc.*
- *Velocidad nominal sin carga: 3308 rpm.*
- *Tiempo en alcanzar el 50% de la velocidad estable: aprox. 0.085 seg.*
- *Tiempo en alcanzar la velocidad estable: aprox. 1.1 seg.*
- *Periodo $T=0.1$ seg.*

Según la referencia bibliográfica [10], la función de transferencia de un motor DC está definida por la siguiente fórmula:

$$\frac{RPM(s)}{V(s)} = \frac{A}{s(s + B)}$$

Despejando $RPM(s)$ y reemplazando el voltaje se tiene:

$$RPM(s) = \frac{30A}{s(s + B)}$$

Pasando esta ecuación al dominio del tiempo se tiene:

$$RPM(t) = \frac{30A}{B} (1 - e^{\frac{-t}{B}})$$

En $t=\infty$ la velocidad es 3308 rpm por lo tanto:

$$RPM(\infty) = \frac{30A}{B} = 3308$$

En $t=0.85$ la velocidad es el 50% de la velocidad nominal por lo tanto:

$$RPM(0.85) = 0.5 \times 3308 = 3308 \left(1 - e^{\frac{-0.85}{B}}\right)$$

$$0.5 = 1 - e^{\frac{-0.85}{B}}$$

$$e^{\frac{-0.85}{B}} = 1 - 0.5 = 0.5$$

$$\frac{-0.85}{B} = \ln 0.5 = -0.69$$

$$B = \frac{-0.85}{-0.69} = 0.123$$

Reemplazando B en $RPM(\infty)$ y despejando A se tiene:

$$A = \frac{3308 \times 0.123}{30} = 13.56$$

Con lo que la ecuación en el dominio del tiempo queda de la siguiente manera:

$$RPM(t) = 3308 \left(1 - e^{\frac{-t}{0.123}}\right)$$

Con esta ecuación se obtuvo la gráfica de la figura 11 la cual es correspondiente a la gráfica obtenida experimentalmente.

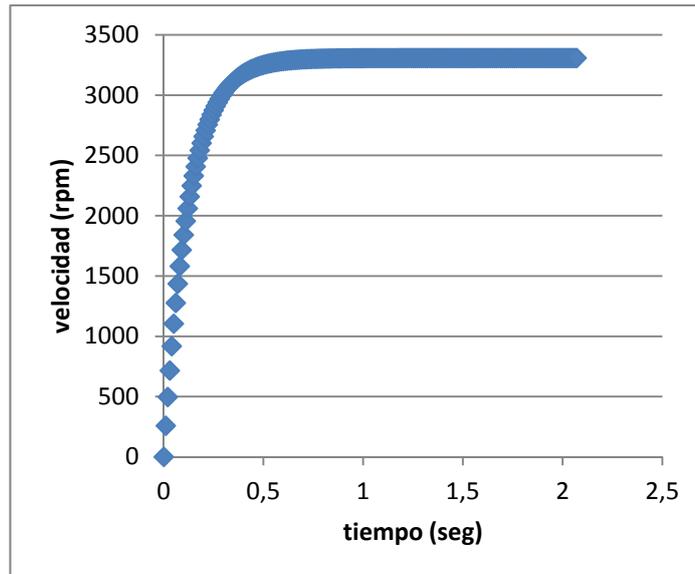


FIGURA 11. GRÁFICA DE LA PLANTA EN EL DOMINIO DEL TIEMPO.

Con A y B se obtiene finalmente la función de transferencia que define al motor DC de este proyecto.

$$G(s) = \frac{A}{s(s+B)} = \frac{13.56}{s(s+0.123)}$$

Discretizando esta función de transferencia se tiene:

$$G(z) = \frac{\frac{A}{B^2}(e^{-BT} - 1 + BT)Z^{-1} + \frac{A}{B^2}(1 - e^{-BT} - BT e^{-BT})Z^{-2}}{1 - (1 + e^{-BT})Z^{-1} + e^{-BT}TZ^{-2}}$$

Reemplazando A, B y T se tiene:

$$G(z) = \frac{\frac{13.56}{0.123^2}(e^{-0.123 \times 0.1} - 1 + 0.123 \times 0.1)Z^{-1} + \frac{13.56}{0.123^2}(1 - e^{-0.123 \times 0.1} - 0.123 \times 0.1 e^{-0.123 \times 0.1})Z^{-2}}{1 - (1 + e^{-0.123 \times 0.1})Z^{-1} + e^{-0.123 \times 0.1} \times 0.1 Z^{-2}}$$

Con lo que la función de transferencia discreta para el motor DC queda de la siguiente manera:

$$G(z) = \frac{0.067522Z^{-1} + 0.067246Z^{-2}}{1 - 1.987775Z^{-1} + 0.098777Z^{-2}}$$

3.3.2. FUNCIÓN DE TRANSFERENCIA DEL CONTROL PID

La función de transferencia del control PID se transformo su forma discreta al ser un control digital.

En función de S el control PID se puede expresar de la siguiente manera:

$$PID(s) = Kp \left(1 + \frac{1}{T_i S} + T_d S \right)$$

Discretizando se tiene:

$$PID(z) = Kp \left[1 + \frac{T}{T_i(1 - Z^{-1})} + T_d \frac{(1 - Z^{-1})}{T} \right]$$

Los parámetros del control PID se obtuvieron mediante el método de curva-reacción de Ziegler-Nichols el cual se lleva a cabo de manera experimental.

Para realizarlo se siguieron los siguientes pasos consultados en la referencia netgráfica [17]:

- a) Se llevó manualmente la planta estando en lazo abierto a un punto fijo de velocidad estable $y(t)=y_0$, poniendo un valor fijo de entrada $u(t)=u_0$.

$$\text{Para } u_0 = 0 \quad \rightarrow \quad y_0 = 0 \text{ rpm}$$

- b) En un tiempo inicial t_0 se aplicó un escalón u_1 en la entrada.
- c) Se registraron los datos hasta que se estabilizó la salida obteniendo gráficamente los datos necesarios.

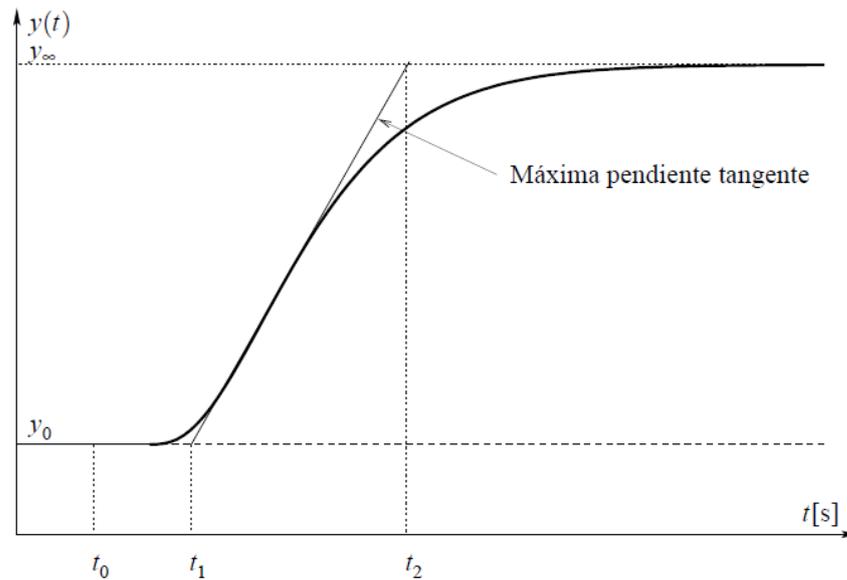


FIGURA 12. DATOS A OBTENER PARA EL MÉTODO CURVA-REACCIÓN DE ZIEGLER-NICHOLS.⁴

Donde:

$$y_0=0$$

$$y_\infty=3308$$

$$u_0=0$$

$$u_\infty=100$$

$$t_0=0$$

$$t_1=0.1$$

$$t_2=0.18$$

- d) Se calcularon los datos necesarios para el ajuste de los parámetros en base a las formulas de Ziegler-Nichols para el método de curva-reacción.

$$k_o = \frac{y_\infty - y_0}{u_\infty - u_0} = \frac{3308 - 0}{100 - 0} = 33.08$$

⁴ Imagen extraída de “Control PID Clásico”, csd.newcastle.edu.au/SpanishPages/clase_slides_download/C07.pdf

$$\tau_0 = t_1 - t_0 = 0.1 - 0 = 0.1$$

$$\gamma_0 = t_2 - t_1 = 0.18 - 0.1 = 0.08$$

- e) En base a la tabla de Ziegler-Nichols para el método de curva-reacción se calculan los parámetros para el control PID.

	K_p	T_i	T_d
P	$\frac{\gamma_0}{K_0 \tau_0}$		
PI	$\frac{0,9\gamma_0}{K_0 \tau_0}$	$3\tau_0$	
PID	$\frac{1,2\gamma_0}{K_0 \tau_0}$	$2\tau_0$	$0,5\tau_0$

FIGURA 13. TABLA DE FORMULAS DE SINTONIZACIÓN PARA EL MÉTODO DE CURVA-REACCIÓN.⁵

$$K_p = \frac{1.2\gamma_0}{k_o \tau_0} = \frac{1.2 \times 0.08}{33.08 \times 0.1} = 0.029020$$

$$T_i = 2\tau_0 = 2 \times 0.1 = 0.2 \text{ seg} = 0.00333333 \text{ min}$$

$$T_d = 0.5\tau_0 = 0.5 \times 0.1 = 0.05 \text{ seg} = 0.00083333 \text{ min}$$

Reemplazando estos parámetros (T_i y T_d en segundos) en la ecuación discreta del control PID se tiene:

$$PID(z) = 0.029020 \left[1 + \frac{0.1}{0.05(1 - Z^{-1})} + 0.2 \frac{(1 - Z^{-1})}{0.1} \right]$$

$$PID(z) = 0.029020 \left[\frac{1 - Z^{-1} + 2 + 2(1 - Z^{-1})^2}{(1 - Z^{-1})} \right]$$

⁵ Figura extraída de “Control PID Clásico”,
 csd.newcastle.edu.au/SpanishPages/clase_slides_download/C07.pdf

$$PID(z) = 0.029020 \left[\frac{2Z^{-2} - 5Z^{-1} + 5}{(1 - Z^{-1})} \right]$$

Con lo que la función de transferencia discreta para el control PID queda de la siguiente manera:

$$PID(z) = \frac{0.05804Z^{-2} - 0.1451Z^{-1} + 0.1451}{(1 - Z^{-1})}$$

3.3.3. FUNCIÓN DE TRANSFERENCIA DEL SISTEMA EN LAZO ABIERTO

La función de transferencia de este sistema en lazo abierto está representado por la fórmula:

$$G_s(z) = G_p(z)PID(z)$$

Resolviéndola en Matlab, cuyo código y capturas se ven en el anexo 12, la función de transferencia del sistema queda de la siguiente manera:

$$G_s(z) = \frac{-0.003902Z^{-4} - 0.01367Z^{-3} + 0.01955Z^{-2} - 0.00979Z^{-1}}{1 - 2.98778Z^{-1} + 2.08655Z^{-2} - 0.09877Z^{-3}}$$

3.3.4. FUNCIÓN DE TRANSFERENCIA DEL SISTEMA EN LAZO CERRADO

La función de transferencia de este sistema realimentado esta representado por la formula:

$$G_s(z) = \frac{G_p(z)PID(z)}{1 + G_p(z)PID(z)}$$

Resolviéndola en Matlab, cuyo código y capturas se ven en el anexo 12, la función de transferencia del sistema queda de la siguiente manera:

$$G_S(z) = \frac{97574Z^{-4} - 341910Z^{-3} + 488870Z^{-2} - 244936Z^{-1}}{97574Z^{-4} + 0.2127 \times 10^7 Z^{-3} - 0.5167 \times 10^8 Z^{-2} + 0.7444 \times 10^8 Z^{-1} - 0.25 \times 10^8}$$

3.4. DISEÑO DE HARDWARE

A continuación se describe el diagrama de bloques del circuito electrónico diseñado como interfaz entre el motor DC y la tarjeta FPGA SPARTAN 3E de XILINX, esta interfaz contiene dos etapas muy generales que son la etapa de control de potencia y la etapa de sensor de velocidad que detecta los impulsos generados a medida que el motor gira produciendo 100 impulsos por cada vuelta del motor.

3.4.1. DIAGRAMA DE BLOQUES

El hardware de este proyecto se compone de distintas etapas o secciones las cuales son Adaptador de voltaje, interfaz FPGA-MOTOR DC y medidor de velocidad, su interconexión se detalla mediante el siguiente diagrama de bloques.

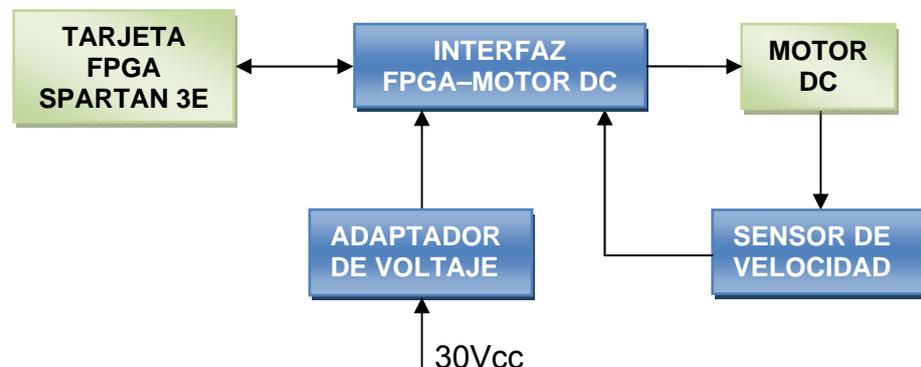


FIGURA 14. DIAGRAMA DE BLOQUES DEL HARDWARE.

3.4.2. DESCRIPCIÓN DE LOS BLOQUES DE HARDWARE

3.4.2.1. ADAPTADOR DE VOLTAJE

El motor DC de este proyecto funciona a 30Vcc pero los demás elementos del hardware necesitan voltajes menores por lo que se realizó un adaptador de voltaje para obtener dichos voltajes, este adaptador consiste en una cascada de dos reguladores de voltaje, de los cuales el primero, un 7818 cuya hoja de datos se puede ver en el anexo 4, reduce el voltaje de entrada de 30Vcc a 18Vcc, los cuales alimentan al colector del optoacoplador y del transistor de potencia y por medio de este al pin “Gate” del MOSFET que conmuta al motor DC y al diodo led que indica que el hardware esta encendido. Posteriormente se toman estos 18Vcc y se los pasa a través de un 7805, cuya hoja de datos se puede ver en el anexo 4, el cual reduce el voltaje a 5Vcc los cuales alimentan al medidor de velocidad.

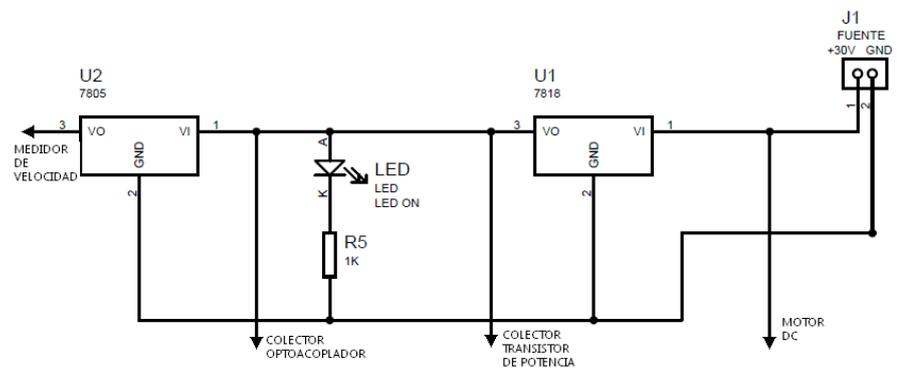


FIGURA 15. DIAGRAMA ESQUEMÁTICO DEL ADAPTADOR DE VOLTAJE.⁶

⁶ Captura del circuito realizado por el tesista en el software Proteus-Isis

CÁLCULOS

En esta etapa se necesita calcular la resistencia correcta para poder alimentar al led con 18Vcc, esta resistencia se calcula mediante la ley de Ohm.

Datos:

$V_{\text{fuente}} = 18\text{Vcc}$
 $I_{\text{led}} = 15\text{-}20 \text{ mA}$
 $V_{\text{led azul}} = 3.5\text{Vcc}$

$$R = \frac{V_{\text{fuente}} - V_{\text{led}}}{I_{\text{led}}} = \frac{18\text{Vcc} - 3.5\text{Vcc}}{0.015\text{A}} = 966.6\Omega \approx 1\text{K}\Omega$$

3.4.2.2.SENSOR DE VELOCIDAD

La lectura de la velocidad del motor se efectúa de manera digital por medio de un optointerruptor con función de encoder el cual envía pulsos de 3.2Vcc al FPGA al pasar su haz de luz a través de las ranuras del disco que está sujeto al eje del motor, este disco posee 100 ranuras por lo que el dispositivo entregará 100 impulsos por cada vuelta del motor.



FIGURA 16. UBICACIÓN DEL MEDIDOR DE VELOCIDAD.

Este optointerruptor se compone de un led infrarrojo y un OPIC (Optical IC) el cual es un dispositivo GP1A30R de la

marca Sharp, cuya hoja de datos se puede ver en el anexo 7, el cual consiste en un elemento detector de luz y un circuito integrado de procesamiento de señales, todo esto en un solo chip, debido a esto este dispositivo tiene la capacidad de trabajar como encoder.

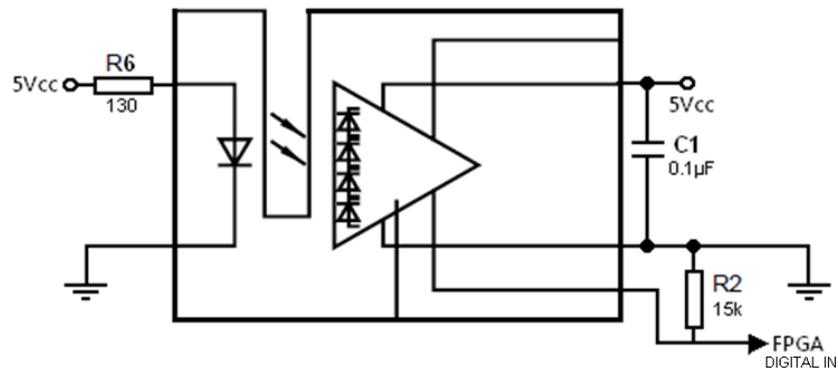


FIGURA 17. DIAGRAMA ESQUEMÁTICO DEL MEDIDOR DE VELOCIDAD.⁷

CÁLCULOS

En esta etapa se necesita calcular la resistencia para alimentar al led infrarrojo del optointerruptor con 5Vcc, además de la resistencia de salida del optointerruptor, el capacitor de 0,1μF no se calcula ya que este elemento es recomendado por el fabricante en el datasheet para estabilizar el voltaje de la fuente.

a) Resistencia del led infrarrojo

Datos:

$$\begin{aligned} V_{\text{fuente}} &= 5V_{\text{cc}} \\ I_{\text{ledIR}} &= 30 \text{ mA} \\ V_{\text{led azul}} &= 1.2V_{\text{cc}} \end{aligned}$$

⁷ Figura extraída del datasheet del optointerruptor SHARP GP1A30R, y modificada por el testista

$$R = \frac{V_{fuente} - V_{led}}{I_{led}} = \frac{5V_{cc} - 1.2V_{cc}}{0.03A} = 126.6\Omega \approx 130\Omega$$

b) Resistencia de salida

Datos:

$$V_{sal} = 3.2V_{cc}$$

$$I_{sal} = 0.22 \text{ mA}$$

$$V_{salida} = I_{salida} \times R_{salida}$$

$$R_{sal} = \frac{V_{sal}}{I_{sal}} = \frac{3.2V_{cc}}{0.00022A} = 14545,4545\Omega \approx 15000\Omega = 15k\Omega$$

3.4.2.3. INTERFAZ FPGA-MOTOR DC

El motor DC no puede ser controlado directamente por el FPGA debido a que trabajan a diferentes voltajes y esto impide la conexión directa, además al ser el voltaje del motor casi diez veces que el del FPGA se debe tener cuidado de que no existan voltajes reversos o corrientes pico reversas muy elevadas desde el motor DC hacia el FPGA ya que lo podría dañar seriamente por lo que la conexión entre el FPGA y el motor debe estar aislada.

La salida del FPGA se conecta a la entrada de un optoacoplador 4n25 cuya hoja de datos se puede ver en el anexo 8, la salida de este se conecta a un transistor de potencia TIP122, cuya hoja de datos se puede ver en el anexo 5, que posee un clamp de Baker con diodos de alta velocidad 1N4148, cuya hoja de datos se puede ver en el anexo 9, lo que mejora la velocidad de conmutación, ya que en las pruebas de hardware se observó que disparando el mosfet IRF540N, cuya hoja de datos se puede ver en el anexo 6,

directamente desde el optoacoplador había mucho retardo en la conmutación, el transistor de potencia dispara al Mosfet el cual conmuta al motor permitiendo que se lo controle por PWM, el motor tiene conectado entre sus terminales un diodo clamp 1N4007, cuya hoja de datos se puede ver en el anexo 10, el cual protege al mosfet del voltaje de la fuente cuando se encuentra abierto.

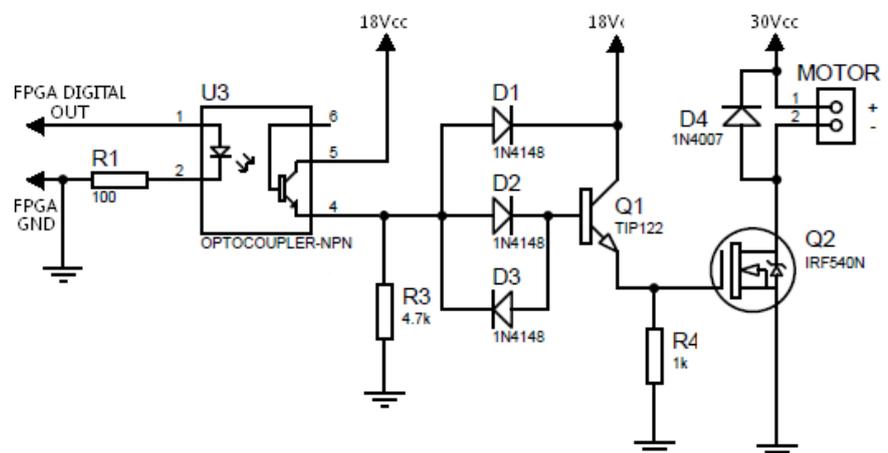


FIGURA 18. DIAGRAMA ESQUEMÁTICO DE LA INTERFAZ FPGA-MOTOR DC.⁸

CÁLCULOS

En esta etapa se necesita calcular la resistencia para alimentar al led del optoacoplador con 3.3Vcc, además de la resistencia del emisor del optoacoplador y del emisor del transistor de potencia.

a) Resistencia del led infrarrojo

Datos:

$$V_{\text{fuente}} = 3.3V_{\text{cc}}$$

⁸ Captura del circuito realizado en el software Proteus-Isis

$$I_{\text{ledIR}} = 10\text{-}20 \text{ mA}$$

$$V_{\text{led azul}} = 1.2V_{\text{cc}}$$

$$R = \frac{V_{\text{fuente}} - V_{\text{led}}}{I_{\text{led}}} = \frac{3.3V_{\text{cc}} - 1.2V_{\text{cc}}}{0.020A} = 105\Omega \approx 100\Omega$$

b) Resistencia de emisor del optoacoplador

Datos:

$$V_{\text{sal}} = 18V_{\text{cc}}$$

$$I_e = 4 \text{ mA}$$

$$V_{\text{salida}} = I_{\text{salida}} \times R_{\text{salida}}$$

$$R_{\text{sal}} = \frac{V_{\text{sal}}}{I_{\text{sal}}} = \frac{18V_{\text{cc}}}{0.004A} = 4500\Omega \approx 4700\Omega = 4.7k\Omega$$

c) Resistencia de emisor del transistor de potencia

Datos:

$$V_{\text{sal}} = 18V_{\text{cc}}$$

$$I_e = 20 \text{ mA}$$

$$V_{\text{salida}} = I_{\text{salida}} \times R_{\text{salida}}$$

$$R_{\text{sal}} = \frac{V_{\text{sal}}}{I_{\text{sal}}} = \frac{18V_{\text{cc}}}{0.020A} = 900\Omega \approx 1000\Omega = 1k\Omega$$

3.4.3. DIAGRAMA ESQUEMÁTICO COMPLETO

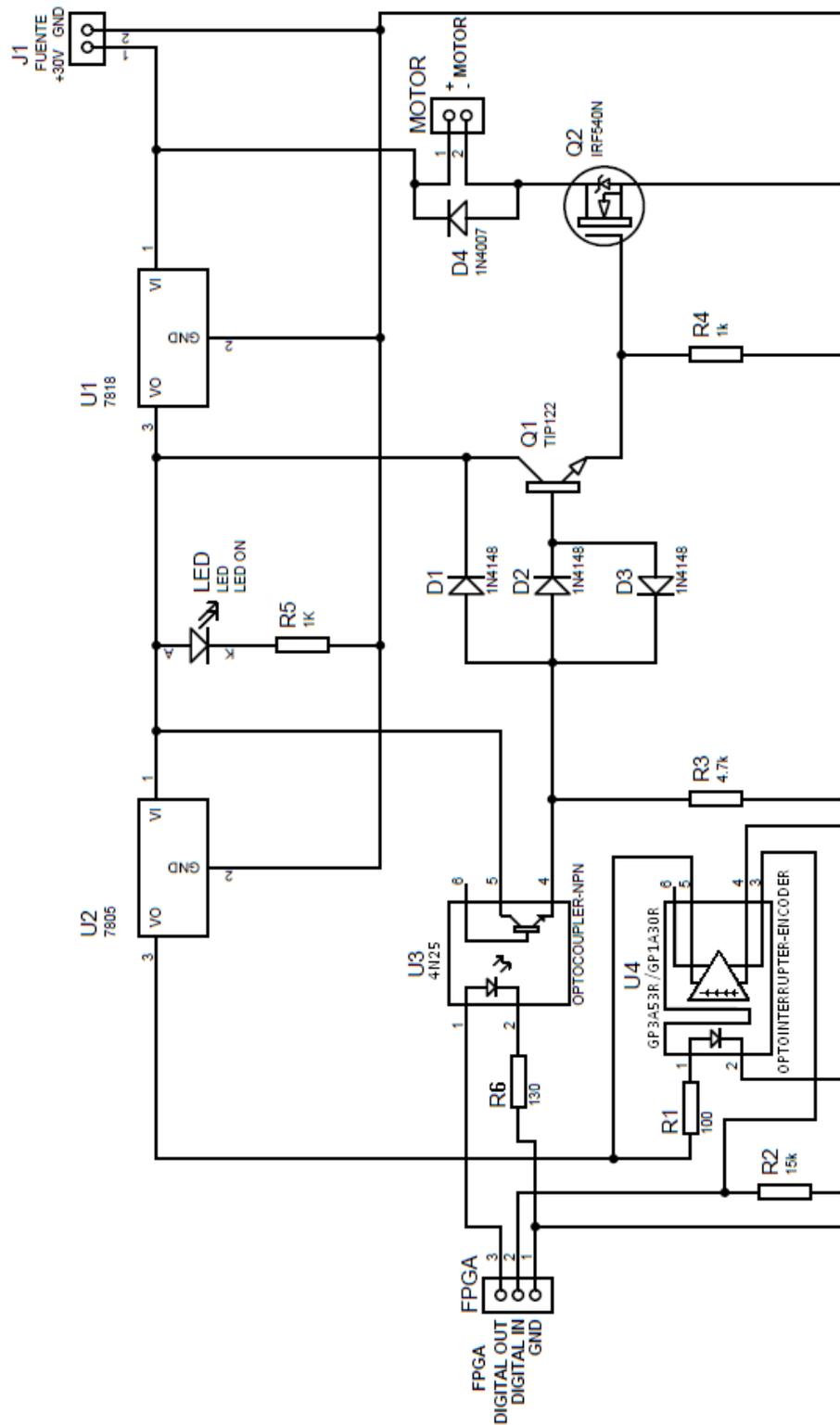


FIGURA 19. DIAGRAMA ESQUEMÁTICO COMPLETO DEL HARDWARE.⁹

⁹ Captura del circuito realizado en el software Proteus-Isis, modificado por el testista

3.5. DESARROLLO DE SOFTWARE

El software de este proyecto está realizado en LabVIEW y se compone de dos VI's, uno es el VI de configuración o programación del FPGA, aquí se desarrolla, en lenguaje grafico, el programa que va a ser compilado, traducido y transferido al FPGA para programarlo; el otro es el VI llamado host, este permite enlazarse al programa del FPGA desde una interfaz desarrollada de igual manera en lenguaje grafico permitiendo modificar las variables de los procesos que fueron programados en el VI de configuración.

Cada VI tiene subprocesos internos los cuales cumplen diferentes funciones los cuales están interconectados entre ellos para formar el sistema completo que permite realizar el control PID de un motor DC desde LabVIEW a través de la tarjeta FPGA SPARTAN 3E.

En esta sección se detallará la estructura y el funcionamiento de cada uno de estos VI's y sus respectivos procesos internos indicando a manera de diagramas de flujo su estructura teórica para posteriormente analizar cada etapa dentro del software de desarrollo LabVIEW.

3.5.1. DIAGRAMA DE FLUJO GENERAL

El diagrama de flujo general muestra de una manera gráfica la estructura secuencial del desarrollo del software del presente proyecto, con lo que se puede tener una idea general del procedimiento para su elaboración y por ende entender mejor su funcionamiento, el código completo del programa se puede ver en el anexo 11.

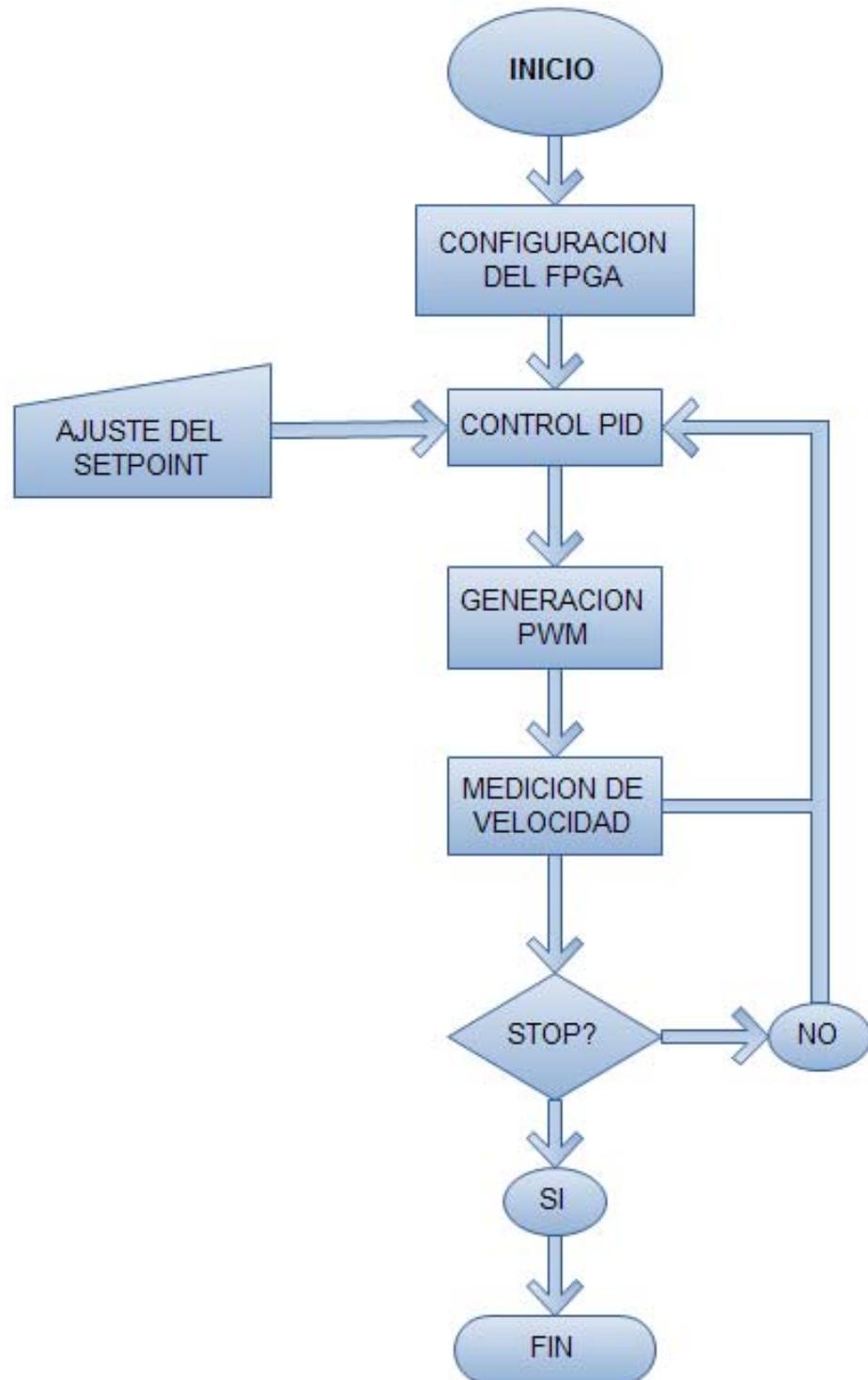


FIGURA 20. DIAGRAMA DE FLUJO DEL SOFTWARE.

- *Inicio*: inicio del sistema.
- *Configuración del FPGA*: se configuran las entradas, salidas y demás herramientas del FPGA que se utilizarán en el proyecto.
- *Ajuste del Set Point*: es un control manual que permite variar el valor del Set Point que es el valor deseado de velocidad.
- *Control PID*: es el control encargado de variar y estabilizar la velocidad de acuerdo al Set Point deseado.
- *Generación PWM*: es la generación de una onda cuadrada en la que se varía el tiempo de pulso, variando el voltaje promedio de la señal, con este voltaje promedio se varía la velocidad del motor.
- *Medición de velocidad*: es la medición de velocidad que realiza LabVIEW de acuerdo con la frecuencia de los pulsos de entrada provenientes del encoder.
- *Stop*: es un pulsador que detiene el sistema en cualquier momento.
- *Fin*: sistema detenido.

3.5.2. DIAGRAMA DE FLUJO DE LA CONFIGURACIÓN DEL FPGA

La configuración o programación del FPGA consiste en seleccionar la salida digital por donde se enviarán los pulsos de la modulación PWM para controlar la velocidad del motor y la entrada digital por la cual se leerán los pulsos del encoder los cuales serán acondicionados y sumados para poder transformarlos a un valor de velocidad en una etapa posterior.

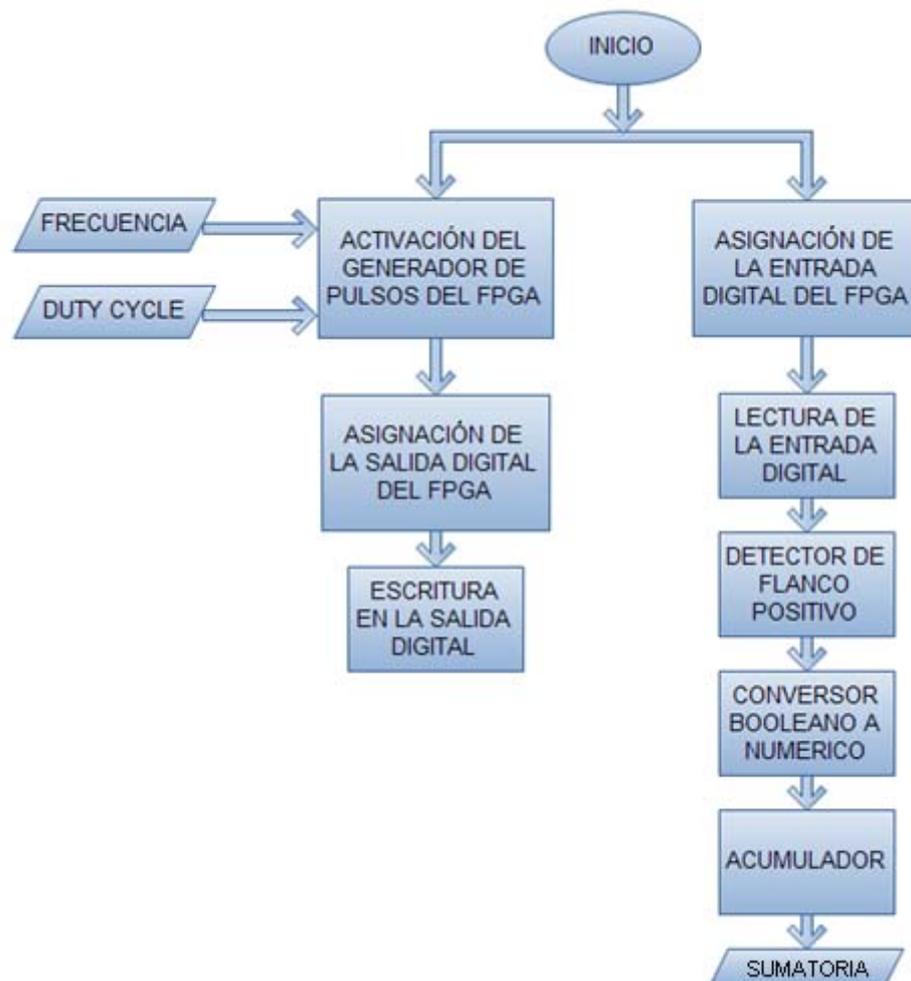


FIGURA 21. DIAGRAMA DE FLUJO DE LA CONFIGURACIÓN DEL FPGA.

- *Inicio*: inicio del sistema.
- *Activación del generador de pulsos del FPGA*: se coloca el bloque de generación de ondas y se lo configura como generador de pulsos ajustando sus parámetros.
- *Frecuencia*: es un control manual que permite variar la frecuencia de los pulsos del generador.
- *Duty Cycle*: es un control manual que permite variar el tiempo de encendido de los pulsos del generador.
- *Asignación de la salida digital del FPGA*: se escoge la salida digital por donde saldrán los pulsos provenientes del generador.

- Escritura en la salida digital: se envían los pulsos desde el generador hacia la salida digital escogida.
- Asignación de la entrada digital del FPGA: se escoge la entrada digital por donde entrarán los pulsos provenientes del encoder.
- Lectura de la entrada digital: se leen los pulsos que vienen del encoder.
- Detector de flanco positivo: detecta los cambios de nivel de 0 a 1(booleano) y da un 1 (booleano) en su salida cada vez que detecta uno de estos cambios esto evita que se produzcan rebotes en la cuenta de los pulsos.
- Conversor booleano a numérico: convierte los valores de booleano a numérico para que puedan ser sumados.
- Acumulador: suma los valores numéricos que salen del conversor, dicho de otra manera, suma los pulsos entrantes
- Sumatoria: es una variable en la que se guarda la sumatoria de los pulsos entrantes.

3.5.3. DIAGRAMA DE FLUJO DE LA GENERACIÓN DEL PWM

En la etapa de generación de PWM se toma la señal del controlador y se la transforma en un valor de duty cycle el cual será enviado al FPGA para que controle al motor pro medio del generador que fue configurado en su interior.

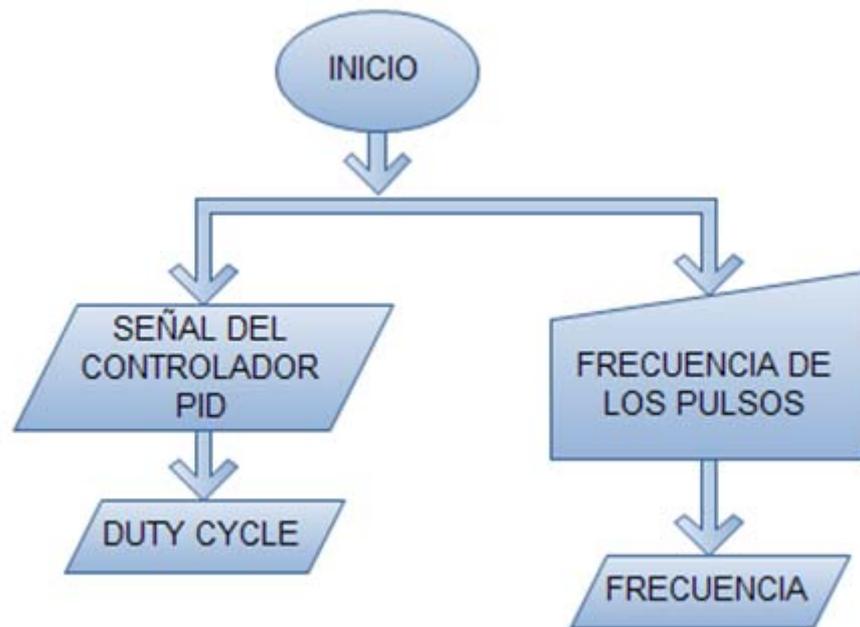


FIGURA 22. DIAGRAMA DE FLUJO DE LA GENERACIÓN PWM.

- *Inicio*: inicio del sistema.
- *Señal del controlador PID*: es la señal de control que proviene del bloque de control PID.
- *Duty cycle*: es una variable que contiene un valor numérico en forma de porcentaje que indica el tiempo de encendido y de apagado de los pulsos del generador de PWM.
- *Frecuencia de los pulsos*: es un control manual que permite variar la frecuencia de los pulsos del generador.
- *Frecuencia*: es una variable numérica que indica la frecuencia de trabajo del generador PWM.

3.5.4. DIAGRAMA DE FLUJO DE LA MEDICIÓN DE VELOCIDAD

En esta etapa se toma la sumatoria total de los pulsos y se obtiene los pulsos por cada periodo, hecho esto se divide este valor para el numero de ranuras del disco sujeto al eje del motor para tener un solo pulso por revolución y finalmente se multiplica este nuevo valor por el valor de tiempo correspondiente para que obtengamos las revoluciones por minuto.

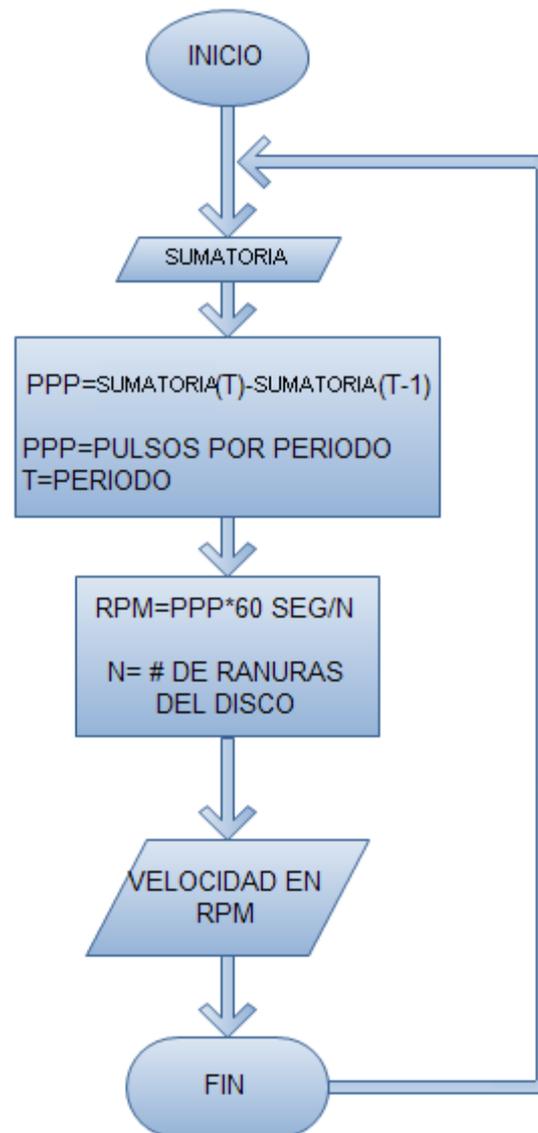


FIGURA 23. DIAGRAMA DE FLUJO DE LA MEDICIÓN DE VELOCIDAD.

- *Inicio*: inicio del sistema.
- *Sumatoria*: es una variable en la que se guarda la sumatoria de los pulsos entrantes.
- *PPP*: son los pulsos por periodo que consisten en la resta de la sumatoria de pulsos entrantes hasta el periodo actual menos la sumatoria de pulsos entrantes hasta el periodo anterior con lo que se tiene la cantidad de pulsos que han ingresado durante el periodo actual.
- *RPM*: es la velocidad en revoluciones por minuto, la cual se consigue realizando las operaciones de conversión necesarias para trasladar los pulsos por periodo a revoluciones por minuto.
- *Velocidad en RPM*: es una variable en donde se almacena el valor numérico correspondiente a las revoluciones por minuto del motor DC.
- *Fin*: fin del ciclo.

3.5.5. DIAGRAMA DE FLUJO DEL CONTROL PID

El control PID necesita como entradas el Set Point o velocidad deseada y el valor de la velocidad medida para realizar una comparación y efectuar el control necesario por medio de una señal de control, el controlador PID que se utiliza en este proyecto es un controlador PID autotuning lo que significa que calcula automáticamente las constantes del control PID.

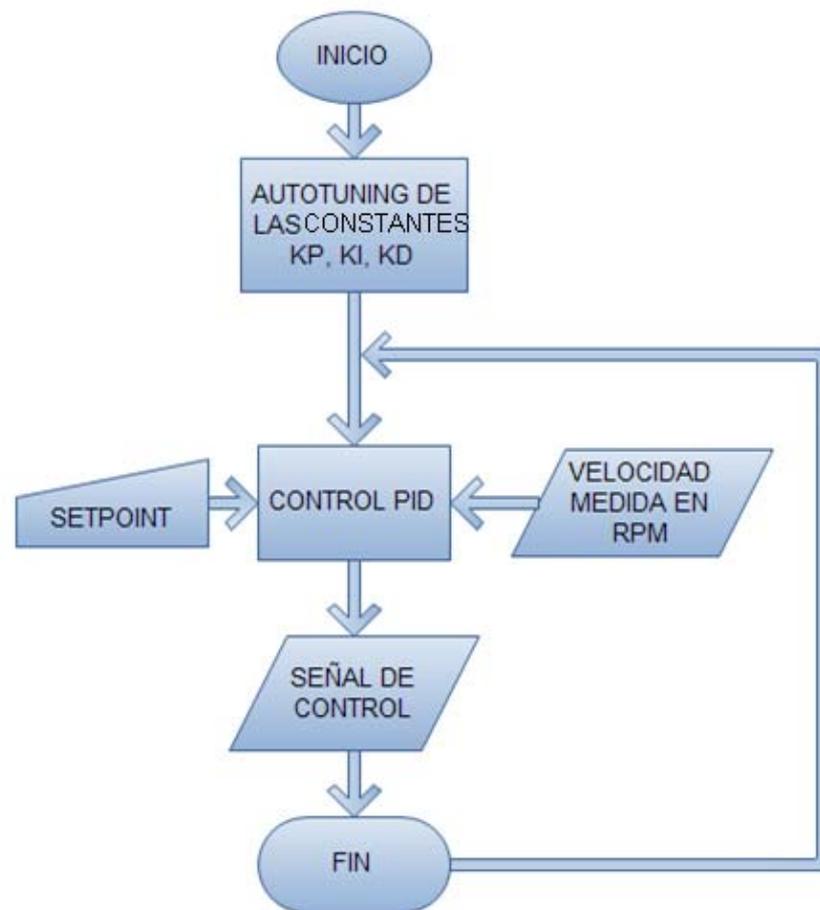


FIGURA 24. DIAGRAMA DE FLUJO DEL CONTROL PID.

- *Inicio*: inicio del sistema.
- *Autotuning de las constantes K_p , K_i , K_d* : es la autorregulación de las constantes del control PID mediante el asistente de autotuning del bloque de control PID AUTOTUNING.
- *Velocidad medida en RPM*: es una variable en donde se almacena el valor numérico correspondiente a las revoluciones por minuto del motor DC.
- *Set Point*: es un control manual que permite variar el valor del Set Point que es el valor deseado de velocidad.
- *Control PID*: es el bloque en el que se produce el control PID de la velocidad del motor DC.
- *Señal del controlador PID*: es la señal de control que proviene del bloque de control PID.

3.5.6. ETAPAS DEL SOFTWARE DESARROLLADO EN LABVIEW

Como se explico al principio de este capítulo el software de este proyecto se compone de dos VI's, el VI de configuración o programación del FPGA, y el VI llamado host, ambos van a ser analizados en esta sección mediante sus diagramas de bloque en LabVIEW.

3.5.6.1. CONFIGURACIÓN DEL FPGA

La configuración del FPGA se realiza mediante el lenguaje de LabVIEW en el VI para posteriormente ser traducido a un archivo de bits que se enviará al FPGA para configurarlo. Esta configuración tiene variables de escritura y lectura los cuales podrán ser controlados y leídos desde el VI HOST.

Primero se seleccionan las entradas, salidas y demás componentes del FPGA que se va a utilizar en el sistema para que estén disponibles en el diagrama de bloques del VI de configuración.

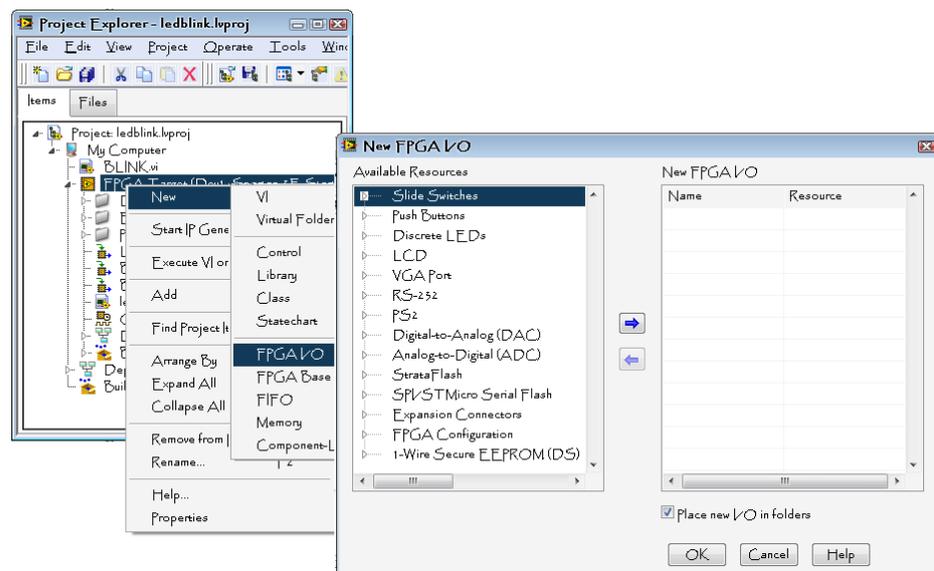


FIGURA 25. SELECCIÓN DE COMPONENTES A UTILIZAR EN EL FPGA.¹⁰

¹⁰ Captura del software LabVIEW

Hecho esto se crea y programa un VI en el FPGA, este VI está dividido en dos lazos “while loop” lo que permite que ambos procesos se ejecuten paralela e independientemente lo que permite que el sistema sea más eficiente.

En el primer lazo se realiza la configuración del generador de pulsos el cual es una herramienta de la paleta “FPGA math & analisis” la que se encuentra disponible en el módulo LabVIEW FPGA.

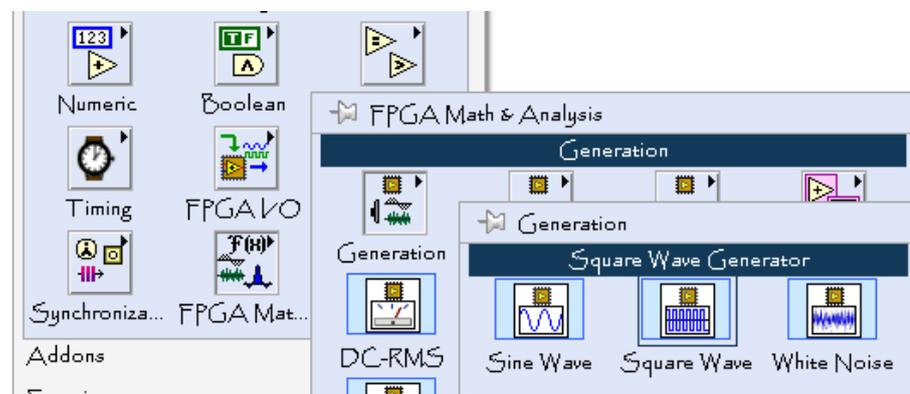


FIGURA 26. UBICACIÓN DEL GENERADOR DE PULSOS EN LA PALETA DE HERRAMIENTAS.¹¹

A este generador se encuentran conectados dos controles, uno para la frecuencia y otro para el Duty Cycle los cuales se pueden variar desde este VI o el VI HOST.

De la misma manera este generador se encuentra conectado a una de los pines digitales previamente seleccionados, el cual se extrae de la paleta “FPGA I/O”, en este caso se tomo el pin IO9, haciendo clic izquierdo en él y seleccionando este pin, y se lo configuro como salida haciendo clic derecho en él y escogiendo el modo “change to write”.

¹¹ Captura del software LabVIEW

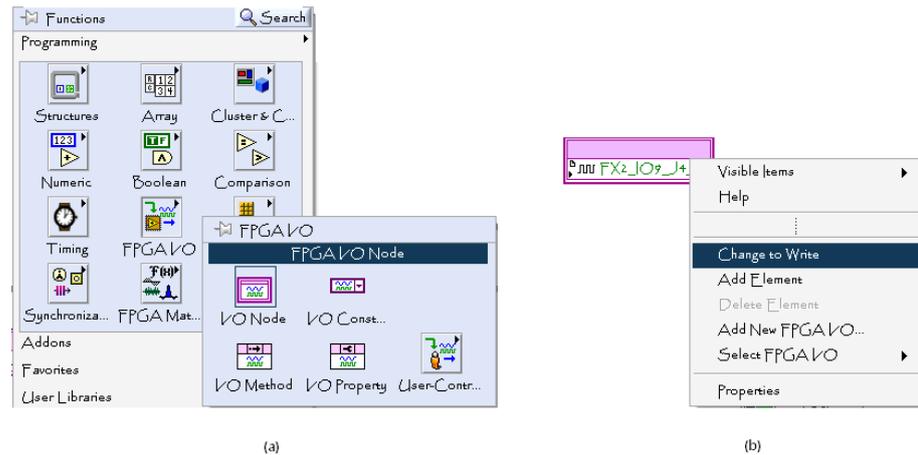


FIGURA 27. (a) UBICACIÓN DE LOS PINES DIGITALES EN LA PALETA DE HERRAMIENTAS, (b) CONFIGURACIÓN DEL PIN COMO SALIDA.¹²

Finalmente se crea un control de STOP para el lazo y con esto se tiene configurado el generador de pulsos del FPGA.

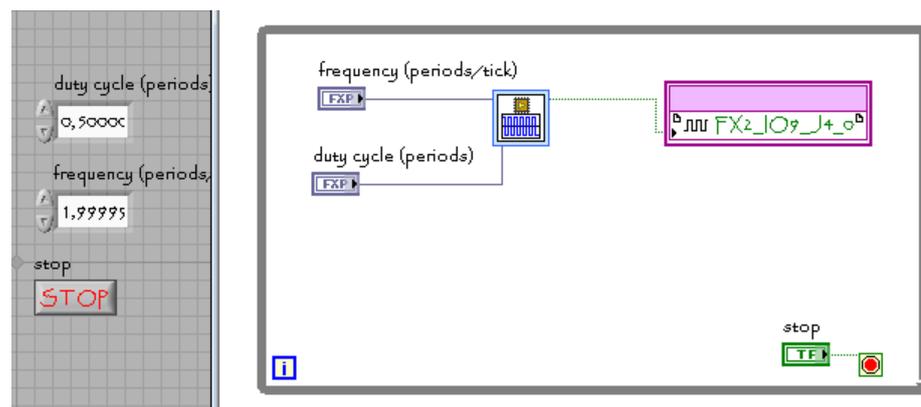


FIGURA 28. CONFIGURACIÓN DEL GENERADOR DE PULSOS DEL FPGA.¹³

En el segundo lazo se configura la lectura y sumatoria de pulsos, para esto se inserta de la misma manera un pin o nodo digital el cual en este caso será el IO10 pero en este caso se lo configurará como entrada seleccionando la opción “change to write”, debido a que este leerá los pulsos del encoder, este se conecta a un visualizador de formas de onda (waveform chart) a través de un conversor de booleano a numérico debido a que el

¹² Captura del software LabVIEW

¹³ Captura del software LabVIEW

visualizador no soporta variables booleanas, igualmente se conecta el pin a un detector de flancos, que es otra herramienta del módulo LabVIEW FPGA, le cual dará un 1 booleano en su salida cada vez que los pulsos entrantes varíen de 0 a 1, luego para sumar los pulsos entrantes se conecta un acumulador a través de un conversor de booleano a numérico ya que el acumulador no soporta variables booleanas, la salida del acumulador se envía a una variable indicadora la cual mostrará la cuenta actual de pulsos.

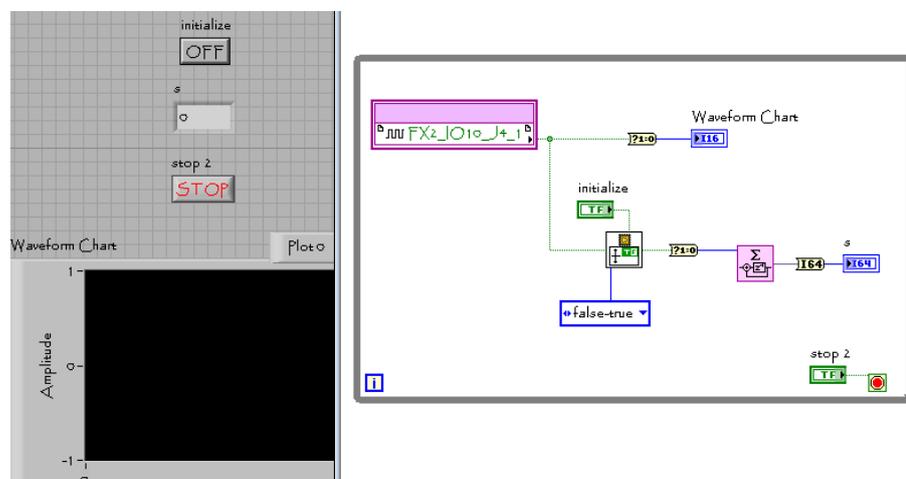


FIGURA 29. CONFIGURACIÓN DEL LECTOR DE PULSOS DEL FPGA.¹⁴

3.5.6.2. GENERACIÓN PWM

La generación de PWM se produce en el VI host, esta generación consiste en hacer un escalamiento de la señal proveniente de la planta, que está en términos de 0 a 100%, para que esté de acuerdo a la unidad de Duty Cycle del generador de pulsos del FPGA que está en términos de 0 a 1, además se realiza una resta de 0,005 debido a que el generador de pulsos del FPGA tiende a volverse inestable al llegar totalmente al 100% de Duty Cycle, de la misma manera se realiza una serie de operaciones para transferir la frecuencia de los pulsos, la cual

¹⁴ Captura del software LabVIEW

está en Hertz, al FPGA ya que este maneja otras unidades de frecuencia por lo que se debe realizar la conversión respectiva.

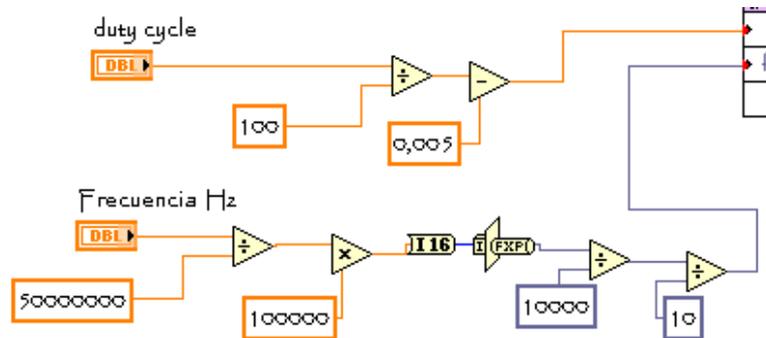


FIGURA 30. CONVERSIÓN DE VARIABLES PARA GENERACIÓN PWM.¹⁵

Para poder enlazar los datos con el FPGA se utiliza una función de lectura/escritura del módulo LabVIEW FPGA la cual permite comunicarse con las variables que se configuran en el VI de configuración del FPGA, esta función debe ir acompañada de dos elementos “open FPGA” y “close FPGA” los cuales permiten enlazarse con el FPGA y acceder a sus variables.

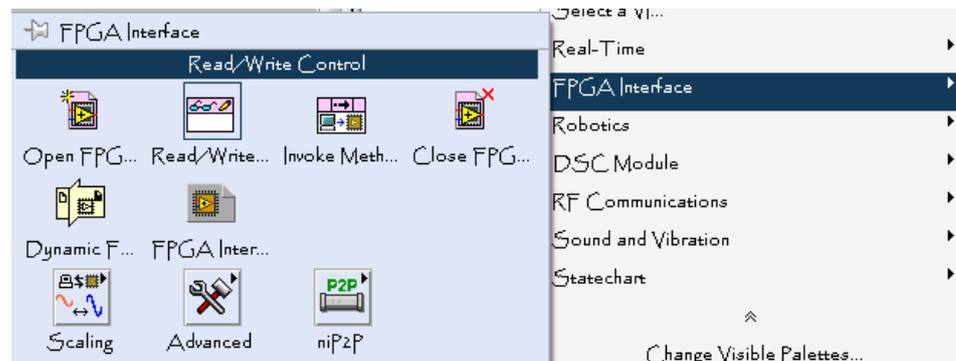


FIGURA 31. UBICACIÓN DEL CONTROL DE LECTURA/ESCRITURA.¹⁶

En el control de lectura escritura se añaden las variables que fueron configuradas haciendo clic derecho y seleccionando “add element” y posteriormente sobre este nuevo elemento se hace clic izquierdo para escoger la variable deseada

¹⁵ Captura del software LabVIEW

¹⁶ Captura del software LabVIEW

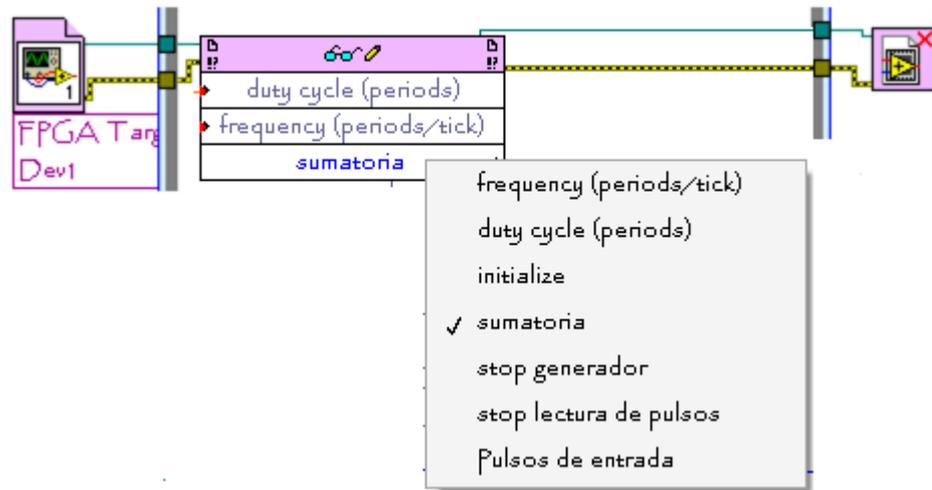


FIGURA 32. FUNCIONES “OPEN FPGA”, “READ/WRITE FPGA” (SELECCIÓN DE VARIABLES) Y “CLOSE FPGA”.¹⁷

3.5.6.3. MEDICIÓN DE VELOCIDAD

Para la medición de velocidad del motor se toma la variable “sumatoria” proveniente del FPGA en la cual se almacena la cuenta de pulsos entrantes al FPGA, en este VI host se realiza una resta de la cantidad de pulsos sumados hasta el periodo actual menos la cantidad de pulsos sumados hasta el periodo anterior, lo que se consigue mediante un registro de desplazamiento, con esto se consigue tener la cantidad de pulsos que hubieron en el periodo actual, hecho esto se deben realizar las operaciones de conversión de “pulsos por periodo” a “revoluciones por minuto”, para la cual se debe tener en cuenta la cantidad de pulsos que hay por cada vuelta del motor, en el caso de este proyecto se dan 100 pulsos por vuelta por lo que se realiza una división para 100, también se debe tener en cuenta la duración del periodo la cual en este caso es 100 milisegundos el cual nos indica que para tener nuestra variable en términos de minutos se debe multiplicarla por 600.

Hecho esto se tiene la velocidad actual del motor en revoluciones por minuto.

¹⁷ Captura del software LabVIEW

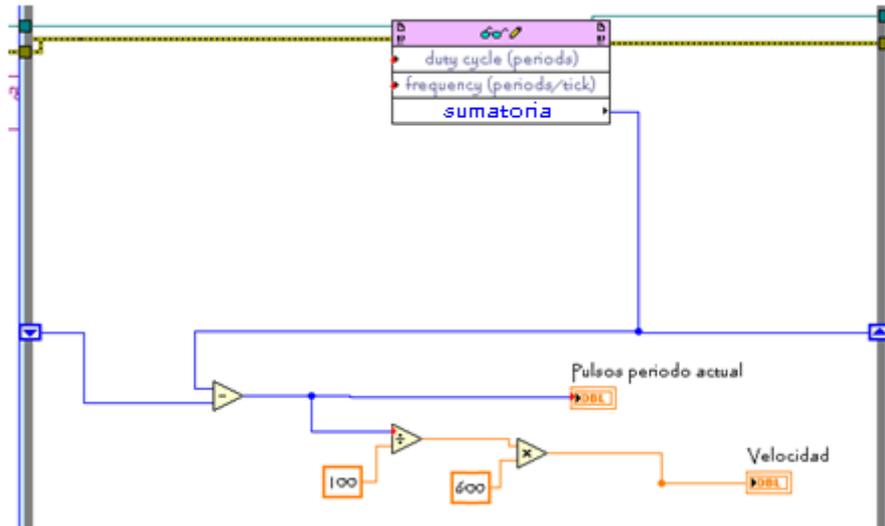


FIGURA 33. MEDICIÓN DE VELOCIDAD.¹⁸

3.5.6.4. CONTROL PID

Para el control PID de velocidad del motor DC se utilizaron las herramientas del software LabVIEW destinadas para este tipo de control, específicamente se utilizó la herramienta “AUTOTUNING PID” la cual nos permite realizar el control PID de un proceso directamente sin tener necesariamente que modelar la planta o calcular las constantes K_p , K_i , K_d , debido a que esta herramienta lo hace automáticamente siguiendo un procedimiento experimental.

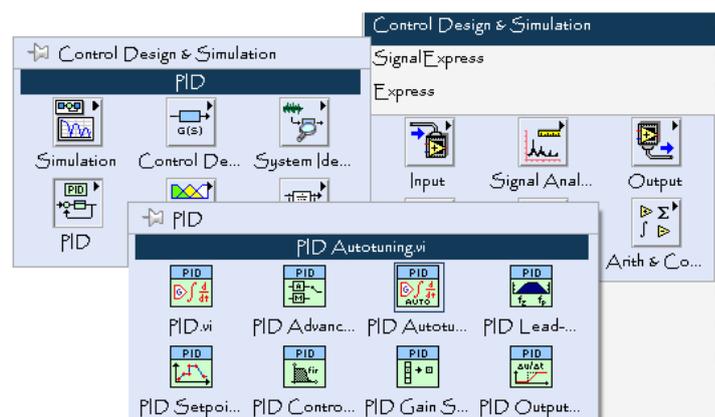


FIGURA 34. UBICACIÓN DEL CONTROL PID AUTOTUNING EN LA PALETA DE HERRAMIENTAS DE CONTROL PID.¹⁹

¹⁸ Captura del software LabVIEW
¹⁹ Captura del software LabVIEW

Para calcular las constantes del control PID esta herramienta posee un asistente el cual de manera muy fácil pide algunas especificaciones al usuario, luego realiza procedimientos experimentales tomando mediciones de la respuesta del sistema y de esta manera calcula las constantes más apropiadas para este sistema.

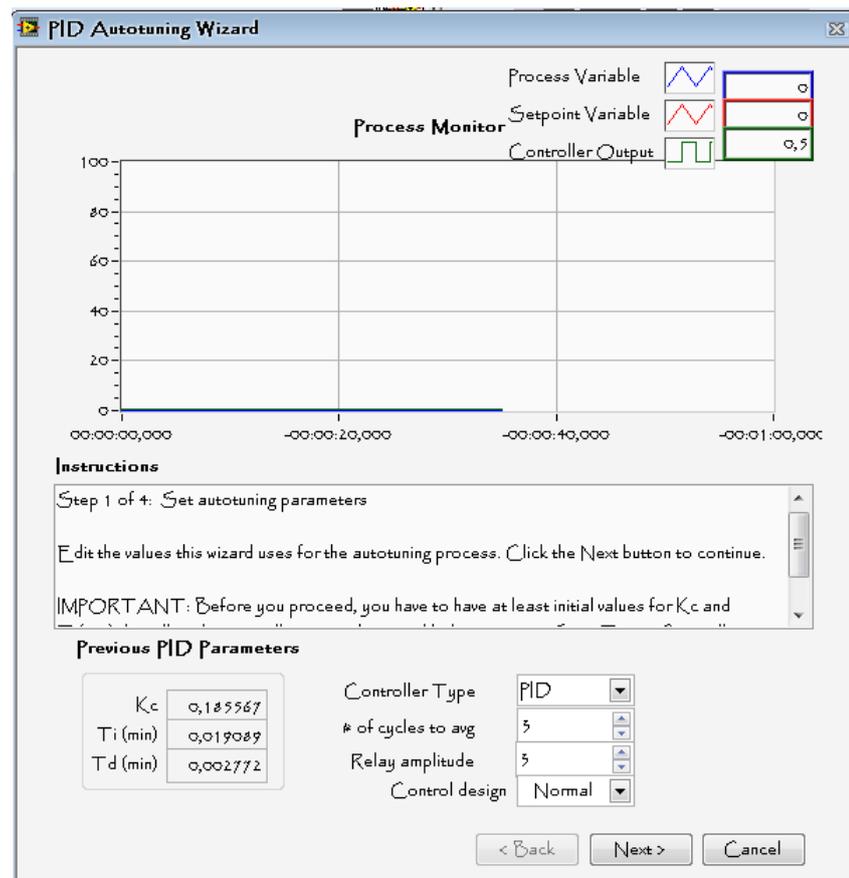


FIGURA 35. ASISTENTE DE AUTOTUNING PARA EL CONTROL PID.²⁰

Al bloque PID autotuning tiene como entradas el control del Set Point, el rango de la salida de control que en este caso al ser el Duty Cycle del PWM será entre 0 y 100, las especificaciones del control y la medición de velocidad, su salida va a la entrada del control de Duty Cycle de generador PWM, esto hace que sea

²⁰ Captura del software LabVIEW

muy fácil la implementación de el control PID de velocidad en el software LabVIEW.

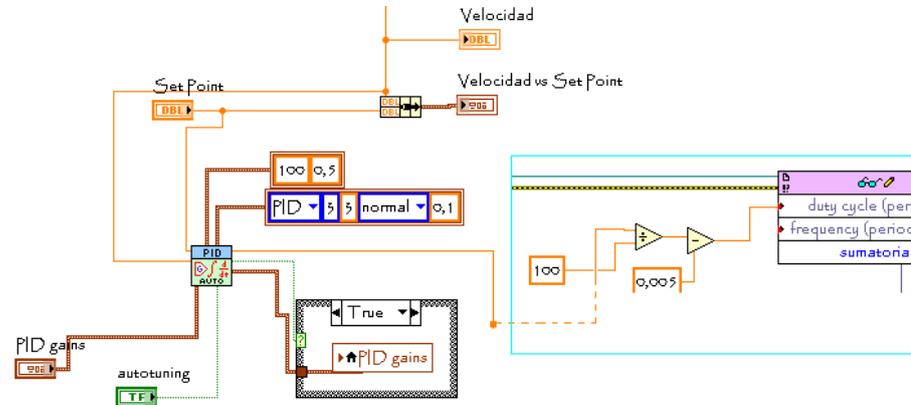


FIGURA 36. CONTROL PID.²¹

Adicionalmente se colocó un graficador para que se pueda visualizar en tiempo real la variación de estabilidad vs el Set Point que se coloca, con esto podemos verificar la efectividad del control PID.

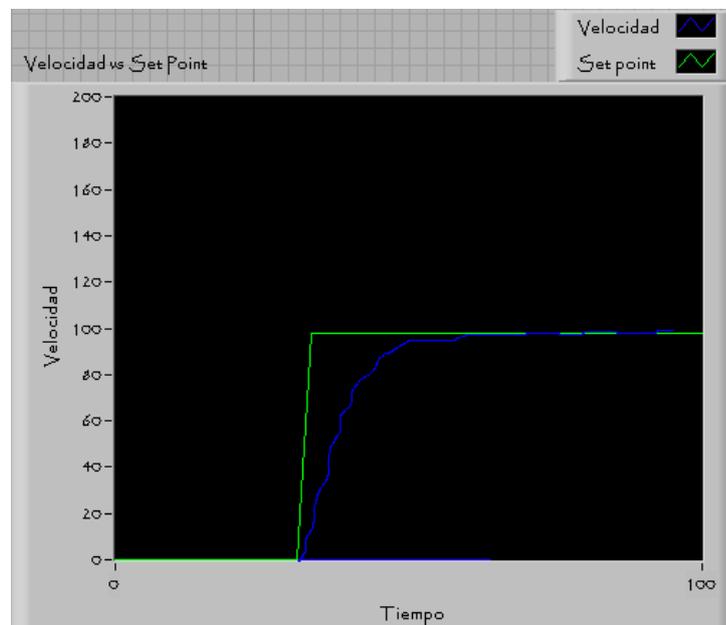


FIGURA 37. GRAFICADOR.²²

²¹ Captura del software LabVIEW

²² Captura del software LabVIEW

3.5.7. DIAGRAMA DE BLOQUES DEL PROGRAMA COMPLETO

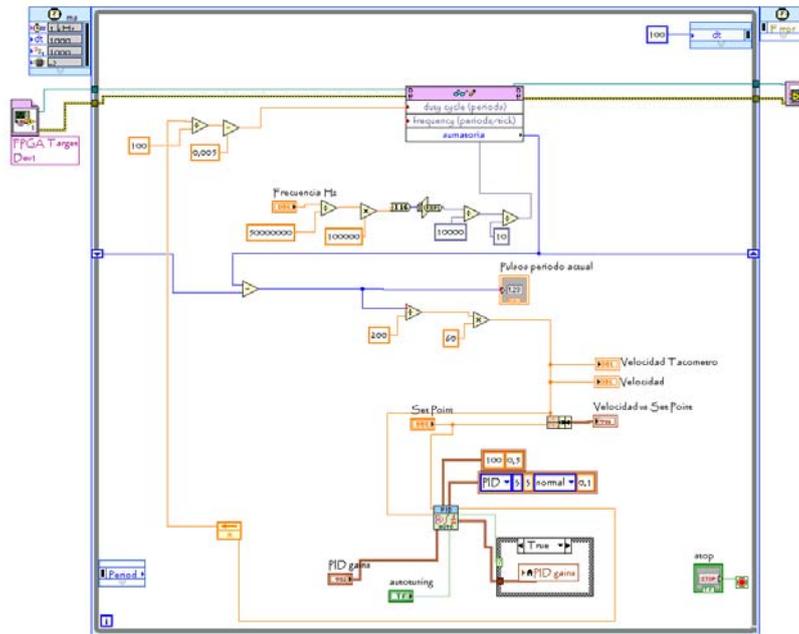


FIGURA 38. DIAGRAMA DE BLOQUES DEL SOFTWARE.²³

3.5.8. PANEL DE INICIO DEL SOFTWARE



FIGURA 39. PANEL DE INICIO DEL SOFTWARE.²⁴

²³ Captura del software LabVIEW

²⁴ Captura del software LabVIEW

3.5.9. PANEL FRONTAL DEL SOFTWARE

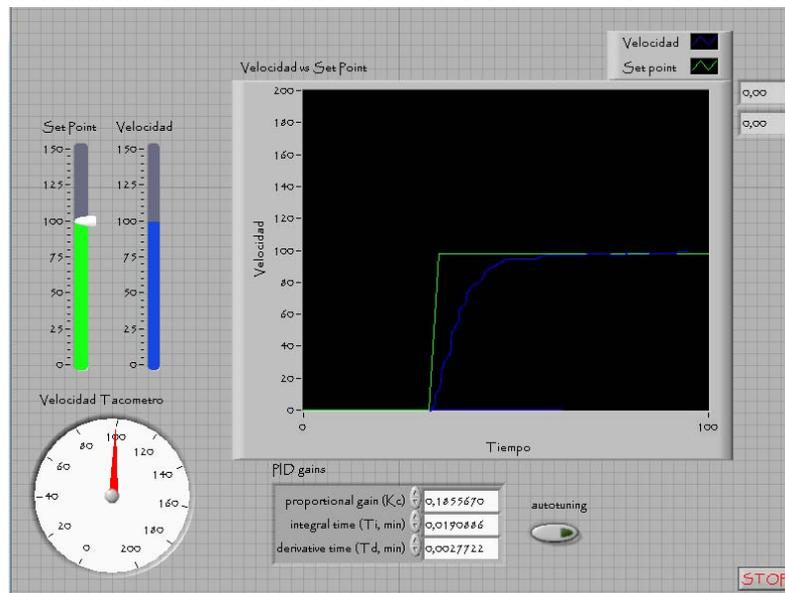


FIGURA 40. PANEL FRONTAL DEL SOFTWARE.²⁵

3.6. CONSTRUCCIÓN DEL MÓDULO

A continuación se describe la construcción del módulo en la parte de hardware.

3.6.1. ELABORACIÓN DE LA PLACA DE INTERFAZ FPGA-MOTOR DC

El diseño de la placa se elaboró en el software Proteus-Ares a partir del circuito diseñado en Proteus-Isis.

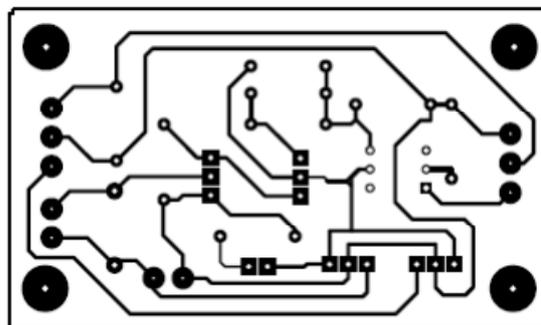


FIGURA 41. PISTAS DE LA PLACA DE INTERFAZ FPGA-MOTOR DC.²⁶

²⁵ Captura del software LabVIEW

²⁶ Captura del software Ares

Las pistas se quemaron en una placa de fibra de vidrio transparente de 7.5 X 4.5cm de una sola cara cubierta de cobre, de la misma manera en la parte superior se estampo la descripción de la placa y de los componentes que están en ella lo que le proporciona una mejor apariencia y sirve de referencia el momento de colocar los elementos.

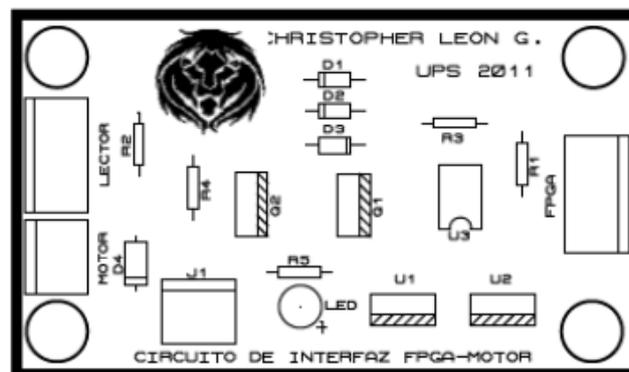


FIGURA 42. DESCRIPCIÓN DE LOS COMPONENTES DE LA PLACA DE INTERFAZ FPGA-MOTOR DC.²⁷

Posteriormente se perforo la placa y se distribuyeron de forma ordenada todos los elementos del circuito para ser soldados.

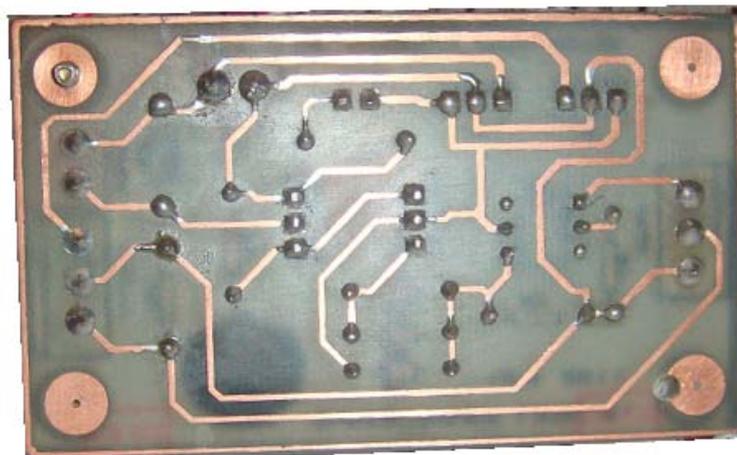


FIGURA 43. PISTAS Y SOLDADURA DE LA PLACA DE INTERFAZ FPGA-MOTOR DC.

²⁷ Captura del software Ares



FIGURA 44. PISTAS Y SOLDADURA DE LA PLACA DE INTERFAZ FPGA-MOTOR DC.

Para la conexión con el motor, el lector de velocidad, la fuente de voltaje y el FPGA se colocaron headers los cuales por su forma física impiden que se conecten mal los conectores que se acoplan a ellos.



FIGURA 45. PLACA COMPLETA.

3.6.2. ELABORACIÓN DE LA ESTRUCTURA

La estructura del módulo se construyó en acrílico de 4 líneas color negro, su base mide 20 X 15cm, en ella se construyó una caja en la cual irá la placa la cual tiene una tapa que se fija mediante cuatro tornillos, esta caja tiene una serie de perforaciones que servirán para que los headers de la placa sean accesibles al usuario, también posee un costado perforado a manera de respiradero

debido a que los reguladores de voltaje tienden a disipar calor por lo que no es conveniente que este se quede encerrado.



FIGURA 46. CAJA DE LA PLACA.

También se construyó un soporte para el motor DC el cual a su vez tiene una pequeña caja la que servirá de protección para el sistema de medición de velocidad que es propenso a sufrir daños de estar expuesto.

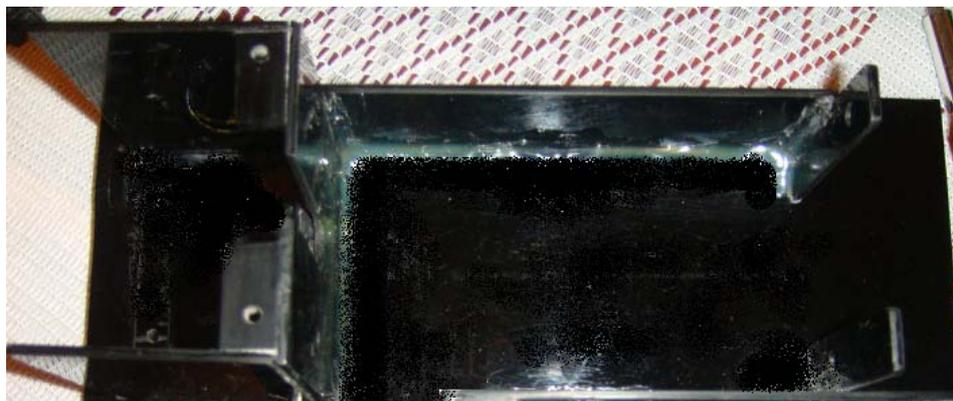


FIGURA 47. SOPORTE DEL MOTOR DC.

En la base de la estructura se colocaron nueve soportes de caucho antideslizante los cuales permiten que el módulo se mantenga fijo cuando este está operando ya que el motor tiende a vibrar cuando esta encendido.



FIGURA 48. SOPORTES ANTIDESLIZANTES.

Con lo que la estructura quedo terminada.

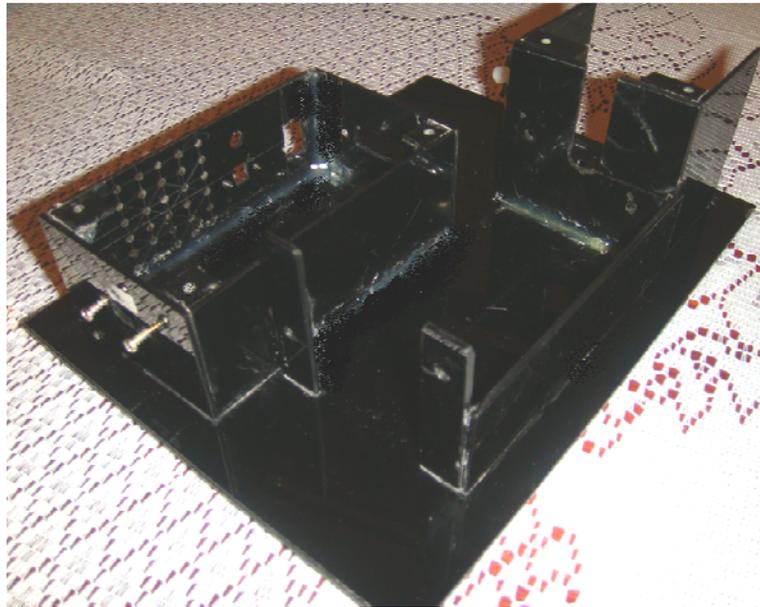


FIGURA 49. ESTRUCTURA FINAL DEL MÓDULO.

3.6.3. MONTAJE DE LA PLACA

La placa de interfaz FPGA-motor DC se monto en la caja destinada para ella mediante dos tornillos, con los cuales queda sujeta a la base, los headers para la conexión del FPGA, motor DC, lector de velocidad, y fuente de voltaje se sujetaron de igual manera con tornillos a las paredes de la caja donde quedan accesibles para utilizarlos.

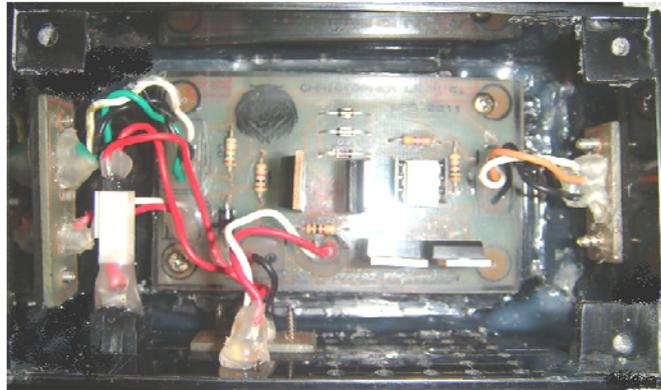


FIGURA 50. MONTAJE DE LA PLACA.

3.6.4. MONTAJE DEL MOTOR DC

El motor se coloca en el soporte que se construyo en base a sus mediciones con lo que queda fijo, además se asegura por medio de tres tornillos, dos están al frente de este y uno en la parte trasera dentro de la caja que protege al sistema de medición de velocidad en la cual entra el eje trasero del motor por medio de una ranura, este eje debe ingresar en esta caja debido a que sobre este se montará el disco ranurado.

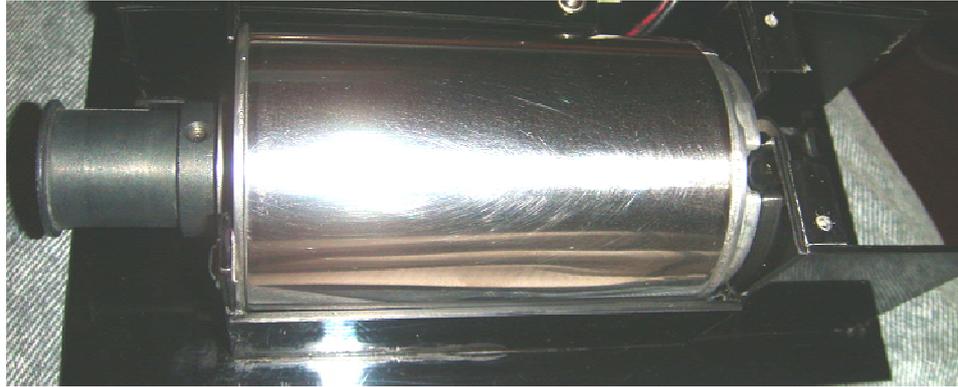


FIGURA 51. MONTAJE DEL MOTOR.

3.6.5. MONTAJE DEL LECTOR DE VELOCIDAD

Este sistema queda instalado dentro de la caja que se encuentra en la parte posterior del motor y en la cual entra su eje.

Para la medición de la velocidad se colocó un disco metálico ranurado en el eje trasero del motor DC, este disco posee 100 ranuras las cuales son detectadas por el encoder.

El optointerruptor con función de encoder se instaló de manera fija junto al eje trasero del motor alineándolo de manera que lea las ranuras del disco metálico.

Los cables del optointerruptor salen por un orificio que se encuentra en un costado de la caja de manera que se puedan conectar estos cables en el header correspondiente.

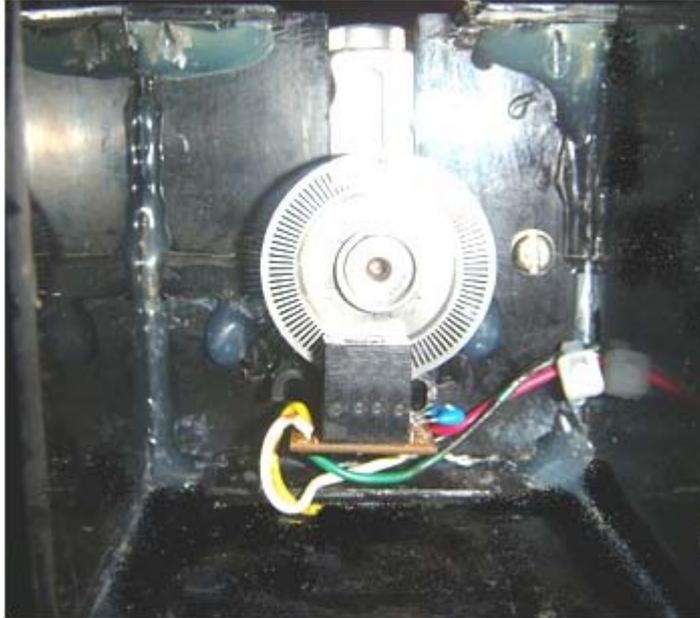


FIGURA 52. MONTAJE DEL SISTEMA DE SENSADO DE VELOCIDAD.

3.6.6. MÓDULO COMPLETO

Una vez que se ha montado el motor y los circuitos en la estructura se conectaron los cables del encoder y del motor a los terminales destinados para ello con lo que el módulo queda totalmente construido y listo para ser probado.



FIGURA 53. MÓDULO COMPLETO.

3.7. COSTOS DEL PROYECTO DE INVESTIGACIÓN

3.7.1. COSTOS DE MATERIALES

En esta sección se desglosa los costos de los materiales que se han utilizado en este proyecto, desde los utilizados en los prototipos previos, hasta llegar a la construcción final del módulo. Para esto se analiza el costo de la estructura, de los elementos de hardware, del diseño de hardware y del desarrollo de software.

3.7.1.1. ESTRUCTURA

En esta sección se describen los materiales que se utilizaron en la construcción de la estructura que sirve de soporte para la placa y el motor DC, incluyendo los costos de cada uno.

DESCRIPCIÓN	CANT.	VALOR UNIT.	TOTAL
1/4 de plancha de acrílico negro de 4 líneas	1	\$ 18,63	\$ 18,63
pistola de silicón pequeña	1	\$ 17,50	\$ 17,50
barra de silicón pequeña	4	\$ 0,10	\$ 0,40
pegamento instantáneo Súper Bonder	2	\$ 3,50	\$ 7,00
cortador de acrílico	2	\$ 1,50	\$ 3,00
soportes antideslizantes x12	1	\$ 3,50	\$ 3,50
tornillo phillips 1/4 pulgada	10	\$ 0,05	\$ 0,50
tornillo torx 1/3 pulgada	2	\$ 0,08	\$ 0,16
tornillo plano 1/3 pulgada	1	\$ 0,05	\$ 0,05
broca 4mm	1	\$ 1,15	\$ 1,15
regla metálica 30cm	1	\$ 2,75	\$ 2,75
cinta adhesiva de papel	1	\$ 0,80	\$ 0,80
lija de agua 320	1	\$ 0,45	\$ 0,45
lija de agua 1200	1	\$ 0,45	\$ 0,45
lija de agua 500	1	\$ 0,45	\$ 0,45
lija de agua 1500	1	\$ 0,45	\$ 0,45
lija de agua 380	1	\$ 0,45	\$ 0,45
TOTAL			\$ 57,69

TABLA 3. COSTOS DE LA ESTRUCTURA.

3.7.1.2. HARDWARE

En esta sección se describen los materiales que se utilizaron en la elaboración del hardware del presente proyecto, incluyendo los costos de cada uno.

DESCRIPCIÓN	CANT.	VALOR UNIT.	TOTAL
placa fibra de vidrio para circuitos tamaño A4	1	\$ 15,00	\$ 15,00
resistencia 1k 1/4 de vatio	5	\$ 0,04	\$ 0,20
resistencia 4.7k 1/4 de vatio	2	\$ 0,04	\$ 0,08
resistencia 100 1/4 de vatio	5	\$ 0,04	\$ 0,20
resistencia 220 1/4 de vatio	5	\$ 0,04	\$ 0,20
resistencia 2,2k 1/4 de vatio	5	\$ 0,04	\$ 0,20
resistencia 1M 1/4 de vatio	5	\$ 0,04	\$ 0,20
resistencia 330 1/4 de vatio	5	\$ 0,04	\$ 0,20
regulador 5Vcc 7805	2	\$ 0,04	\$ 0,08
regulador 18Vcc 7818	2	\$ 0,04	\$ 0,08
mosfet IRF540N	2	\$ 0,04	\$ 0,08
transistor NPN TIP122	2	\$ 0,04	\$ 0,08
capacitor cerámico 104	2	\$ 0,10	\$ 0,20
diodo 1n4007	4	\$ 0,04	\$ 0,16
diodo 1n4148	5	\$ 1,04	\$ 5,20
conector cable 3 pines	3	\$ 0,75	\$ 2,25
conector cable 2 pines	3	\$ 0,75	\$ 2,25
conector cable 1 pin	5	\$ 0,25	\$ 1,25
conector cable 4 pines	3	\$ 0,75	\$ 2,25
cable UTP por metros	1	\$ 1,20	\$ 1,20
optoacoplador 4n25	5	\$ 0,58	\$ 2,90
compuerta 7400	1	\$ 0,60	\$ 0,60
zócalo 4 pines	4	\$ 0,45	\$ 1,80
zócalo 14 pines	1	\$ 0,50	\$ 0,50
led azul	2	\$ 0,30	\$ 0,60
emisor infrarrojo	1	\$ 0,30	\$ 0,30
receptor infrarrojo	1	\$ 0,45	\$ 0,45
optointerruptor tipo U NPN	1	\$ 0,85	\$ 0,85
optointerruptor/encoder Sharp GP3A53R	1	\$ 4,50	\$ 4,50
disco metálico 100 ranuras	1	\$ 1,75	\$ 1,75
interruptor pequeño	2	\$ 0,35	\$ 0,70
cables para fuente de alimentación por metros	2	\$ 0,95	\$ 1,90
amplificador operacional LM358	1	\$ 0,48	\$ 0,48
motor Hitachi 30Vcc	1	\$ 30,00	\$ 30,00
TOTAL			\$ 78,69

TABLA 4. COSTOS DE HARDWARE.

3.7.2. COSTOS DE DISEÑO DE HARDWARE

Los costos del diseño del hardware de el presente proyecto el cual se compone de la circuitería electrónica y la estructura del módulo, incluyendo los recursos necesarios que se han utilizado para la elaboración estos, desde el diseño de los prototipos previos, hasta llegar al diseño final del hardware del módulo ascienden a 100 dólares USA.

3.7.3. COSTOS DE DESARROLLO DE SOFTWARE

Los costos del desarrollo del software del presente proyecto en LabVIEW incluyendo los recursos necesarios para su elaboración, desde las versiones de software previas, hasta llegar al software final del proyecto ascienden a 200 dólares USA.

3.7.4. MANO DE OBRA

En esta sección se desglosa los costos de mano de obra tanto del software como del hardware del presente proyecto.

DESCRIPCIÓN	COSTO
construcción de la estructura	\$ 25,00
construcción de hardware	\$ 35,00
TOTAL	\$ 60,00

TABLA 5. COSTOS MANO DE OBRA.

3.7.5. COSTO TOTAL DEL EQUIPO

En esta sección se recogen los costos totales de los materiales, diseño del hardware, desarrollo de software y mano de obra para saber cuál es el costo total de equipo del presente proyecto de investigación.

DESCRIPCIÓN	COSTO
Materiales	
Estructura	\$ 57,69
Hardware	\$ 78,69
Diseño de hardware	\$ 100,00
Desarrollo de software	\$ 200,00
Mano de obra	\$ 60,00
TOTAL	\$ 496,38

TABLA 6. COSTO TOTAL DEL EQUIPO.

CAPITULO 4. ANÁLISIS DE RESULTADOS

4.1. INTRODUCCIÓN

Se realizaron pruebas para identificar el error absoluto y el error relativo de la velocidad medida en RPM's por LabVIEW haciendo una comparación entre las mediciones de LabVIEW y las mediciones del tacómetro.

4.2. PRUEBAS POR SET POINT

Se realizaron veinte mediciones de velocidad, tanto en LabVIEW como con el tacómetro, para tres Set Point diferentes, con la resta de estos valores se obtendrá el error absoluto tomando como valor de referencia el valor dado por el tacómetro, luego por regla de tres se obtendrá el error relativo del sistema para cada Set Point.

a) *Set Point: 501.742*

En el siguiente cuadro se observan las velocidades medidas tanto en LabVIEW como en el tacómetro para un Set Point de 501.742 el cual se ajusto en LabVIEW, en la parte derecha del cuadro se encuentra calculado el error absoluto para cada medición y al final se encuentran promediadas las velocidades medidas en LabVIEW y en el tacómetro de lo cual se calculó el error absoluto promedio para este Set Point.

PRUEBA	VELOCIDAD LABVIEW	VELOCIDAD TACÓMETRO	ERROR ABSOLUTO
1	498	500,8	2,8
2	504	502,4	1,6
3	498	505,3	7,3
4	498	500,6	2,6
5	504	503,4	0,6
6	504	502,2	1,8
7	504	503,5	0,5
8	498	501	3
9	504	501,7	2,3
10	498	500,3	2,3
11	498	504,1	6,1

12	504	501,3	2,7
13	498	502,4	4,4
14	510	501,5	8,5
15	498	503,4	5,4
16	498	501,1	3,1
17	504	501,3	2,7
18	504	501,7	2,3
19	498	502,9	4,9
20	498	501,1	3,1
PROMEDIO	501	502,1	3,4

TABLA 9. PRUEBAS SET POINT 501.742

Error relativo del sistema:

$$502.1 \rightarrow 100\%$$

$$3.4 \rightarrow X = 0.67\%$$

b) Set Point: 1003.48

En el siguiente cuadro se observan las velocidades medidas tanto en LabVIEW como en el tacómetro para un Set Point de 1003.48 el cual se ajusto en LabVIEW, en la parte derecha del cuadro se encuentra calculado el error absoluto para cada medición y al final se encuentran promediadas las velocidades medidas en LabVIEW y en el tacómetro de lo cual se calculó el error absoluto promedio para este Set Point

PRUEBA	VELOCIDAD LABVIEW	VELOCIDAD TACÓMETRO	ERROR ABSOLUTO
1	1002	1009	7
2	1008	1003	5
3	1008	1002	6
4	1002	1004	2
5	996	1003	7
6	1002	1001	1
7	996	1002	6

8	1008	1003	5
9	1002	998,6	3,4
10	996	1001	5
11	996	996,2	0,2
12	996	1001	5
13	1002	996,6	5,4
14	1014	1005	9
15	1002	1003	1
16	1002	1003	1
17	1008	1004	4
18	1002	1001	1
19	996	997,4	1,4
20	1008	1003	5
PROMEDIO	1002,3	1001,84	4,02

TABLA 10. PRUEBAS SET POINT 1003.48

Error relativo del sistema:

$$\begin{array}{rcl}
 1001.84 & \rightarrow & 100\% \\
 4.02 & \rightarrow & X = 0.4\%
 \end{array}$$

c) Set Point: 2006.97

En el siguiente cuadro se observan las velocidades medidas tanto en LabVIEW como en el tacómetro para un Set Point de 2006.97 el cual se ajustó en LabVIEW, en la parte derecha del cuadro se encuentra calculado el error absoluto para cada medición y al final se encuentran promediadas las velocidades medidas en LabVIEW y en el tacómetro de lo cual se calculó el error absoluto promedio para este Set Point

PRUEBA	VELOCIDAD LABVIEW	VELOCIDAD TACÓMETRO	ERROR ABSOLUTO
1	2004	2006	2
2	1998	2001	3

3	2014	2004	10
4	1998	2007	9
5	2010	2010	0
6	2008	2009	1
7	2004	2013	9
8	2004	2000	4
9	2010	2002	8
10	1998	1996	2
11	2004	2004	0
12	2004	2006	2
13	2010	1999	11
14	1998	2004	6
15	2004	1986	18
16	2004	2000	4
17	2004	1994	10
18	2010	2015	5
19	2008	2012	4
20	1998	2006	8
PROMEDIO	2004,6	2003,7	5,8

TABLA 11. PRUEBAS SET POINT 2006.97

Error relativo del sistema:

$$\begin{array}{lcl}
 2003.7 & \rightarrow & 100\% \\
 5.8 & \rightarrow & X = 0.29\%
 \end{array}$$

4.3. CURVA DE CALIBRACIÓN

La curva de calibración sirve para determinar una ecuación que permita reducir el error en la medición de la velocidad del motor DC para lo cual se tomaron medidas a distintos valores de velocidad tanto en LabVIEW como en el tacómetro.

Velocidad medida (RPM)	
LabVIEW	Tacómetro
0	0
252	250
498	500

750	749
1020	1017
1248	1248
1498	1483
1740	1723
1998	1971
2250	2208
2472	2467
2680	2651
2984	2929

TABLA 12. DATOS PARA LA CURVA DE CALIBRACIÓN.

Luego ingresando estos datos en Microsoft Excel se obtiene la grafica que indica la curva de tendencia con su respectiva ecuación.

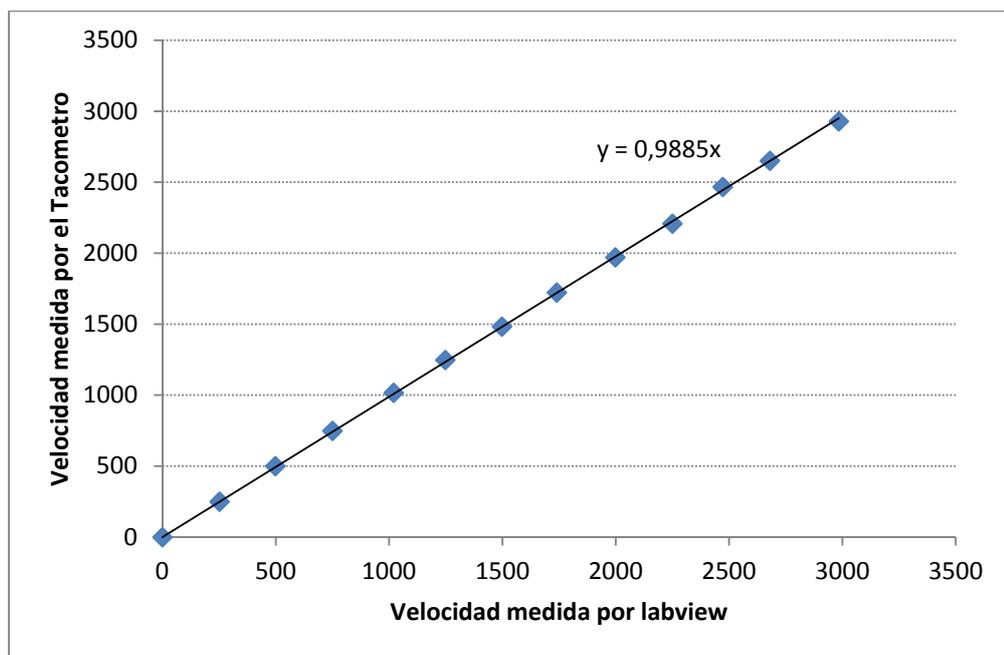


FIGURA 54. CURVA DE CALIBRACIÓN.

Esta ecuación fue implementada en la medición de velocidad en LabVIEW para calibrarla antes de ingresar al controlador PID para reducir los errores de medición entre los valores de velocidad en LabVIEW y los valores del tacómetro.

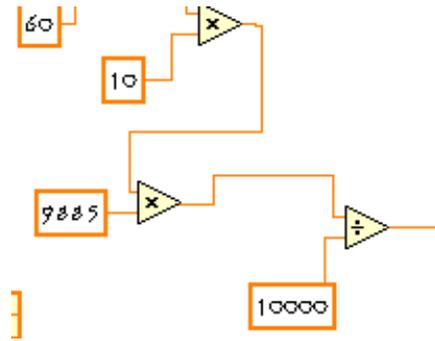


FIGURA 54. CALIBRACIÓN DE LA VELOCIDAD MEDIDA.²⁸

²⁸ Captura del software LabVIEW

CAPITULO 5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- Se cumplió con el objetivo “reconocer y controlar la tarjeta FPGA SPARTAN 3E desde el software de desarrollo LabVIEW”.
- Se cumplió con el objetivo “utilizar la tarjeta FPGA SPARTAN 3E como una tarjeta de adquisición de datos y se pudo visualizar estos en el software de desarrollo LabVIEW”.
- Se cumplió con el objetivo “controlar las salidas digitales de la tarjeta FPGA SPARTAN 3E desde LABVIEW”.
- Se cumplió con el objetivo “adquirir y controlar la velocidad de un motor DC desde LABVIEW a través de la tarjeta FPGA SPARTAN 3E”.
- Se cumplió con el objetivo “realizar el control PID en LABVIEW de la velocidad del motor DC”.
- Mediante la realización de este proyecto de investigación se PROGRAMO la tarjeta FPGA SPARTAN 3E mediante el lenguaje grafico de LabVIEW lo que hace mucho más sencilla esta tarea ya que la estructura de programación en LabVIEW es muy similar a la estructura de los elementos internos del FPGA.
- Existe la compatibilidad del software de desarrollo LabVIEW y la tarjeta FPGA SPARTAN 3E, además de sus poderosas funciones, se pueden desarrollar una gran cantidad de aplicaciones en varias ramas de la investigación tecnológica.
- Durante el desarrollo de este proyecto se logro comprender de una mejor manera la función y la importancia del control automático, como el control PID, en los procesos industriales y experimentales que necesiten precisión y velocidad.
- Se utilizó las funciones del módulo FPGA de LabVIEW para realizar aplicaciones de LabVIEW con dispositivos de hardware que tengan un FPGA como núcleo.

- Se utilizó el bloque de control PID AUTOTUNING, el cual está presente en el software de desarrollo LabVIEW, para implementarlo sobre procesos a nivel de hardware que necesiten de un control automático.
- En LabVIEW, el bloque PID se diferencia del bloque PID AUTOTUNING en que el segundo tiene la capacidad de autorregular sus constantes de manera automática mediante un asistente de autorregulación.
- La circuitería de la placa de control de potencia fue diseñada tomando en cuenta el rango de frecuencias a las que va a trabajar la generación PWM para evitar problemas en la conmutación.
- Para sensor los pulsos del motor DC es necesario utilizar un dispositivo con función de encoder ya que estos están especialmente diseñados para la medición de velocidad, por lo que el error que se tiene en la medición de velocidad es muy bajo.
- No se puede bajar el voltaje de 30Vcc a 5Vcc directamente con los reguladores lineales debido a que se disipa mucha potencia y el regulador tiende a calentarse demasiado, por lo tanto se baja el voltaje poco a poco mediante una cascada de reguladores, con lo que cada regulador tendrá menos potencia que disipar.
- La interacción entre el software LabVIEW y la tarjeta FPGA SPARTAN 3E es muy rápida por lo que se puede trabajar con los datos en tiempo real.

5.2. RECOMENDACIONES

- Se recomienda experimentar con otros tipos de control sobre el módulo del presente proyecto como por ejemplo la lógica difusa o procesamiento digital de señales, lo que permitirá ampliar sus aplicaciones didácticas.
- Se recomienda realizar otros proyectos con LabVIEW en conjunto con la tarjeta FPGA SPARTAN 3E ya que existe una gran variedad de aplicaciones que se pueden realizar con ellos como por ejemplo en el campo de la robótica o el procesamiento digital de señales
- Se recomienda investigar constantemente en la página de National Instruments sobre las capacidades que brindan las nuevas versiones de LabVIEW para trabajar con FPGAs.
- Se recomienda el desarrollo de un manual de prácticas de LabVIEW-FPGA para electrónica digital con lo que los estudiantes podrán adquirir más conocimientos sobre el tema.
- Se recomienda la realización de proyectos que incluyan el uso de los conversores análogo-digital y digital-análogo de la tarjeta FPGA SPARTAN 3E para realizar otros tipos de mediciones.

BIBLIOGRAFÍA

- [1] OGATA, Katsuhiko, *Ingeniería De Control Moderna*, 3ª edición, Editorial Prentice Hall, México, 1995, Págs. 669-678.
- [2] KUO, Benjamín, *Sistemas De Control Automático*, 7ª edición, Editorial Prentice Hall, México, 1996, Págs. 58-64, 836-852.
- [3] NATIONAL INSTRUMENTS, *LabVIEW PID Control Toolset User Manual*, Manual de Usuario, 2001, Págs. 36-38.
- [4] MOLINA, Jorge, *Introducción A Los Controladores PID*, Artículo de la Universidad de los Andes Facultad de Ingeniería, Mérida-Venezuela, Págs. 5-8.
- [5] GOMARIZ, Espartacus, *Teoría De Control Diseño Electrónico*, 1ª edición, Editorial Alfaomega, México, 1999, Págs. 176-182.
- [6] ROSS, *The Essence Of Power Electronics*, 1ª edición, Editorial Prentice Hall, Michigan-USA, 1997, Págs 6-11.
- [7] NATIONAL INSTRUMENTS, *Getting Started with the LabVIEW FPGA Module and the Spartan-3E Starter Board*, Manual de Usuario, 2007, Págs 7-14.
- [8] XILINX, *Spartan-3E Starter Kit Board User Guide*, Manual de Usuario, 2006, Págs. 15-20, 122.
- [9] HIGHLIGHTS, *Entrenador De planta De Control*, Manual de Usuario, 2009, Págs. 25-43.
- [10] SERRUDO, Mario, *Control De Velocidad De Un Motor DC Mediante Modulación Por Ancho De Pulso*, Informe Técnico Instituto Universitario Aeronáutico, 2006, Págs. 6, 26-29.
- [11] OGATA, Katsuhiko, *Sistemas De Control En Tiempo Discreto*, 2ª edición, Editorial Prentice Hall, México, 1996, Págs. 114-119.

NETGRAFÍA

- [12] ALECSA Y ONEA, “*A FPGA Implementation Of A Phase Locked Loop For DC Motor Control*”, www.tuiasi.ro/users/103/Binder1.pdf, 2009, página consultada el 16 de marzo del 2011.
- [13] HASHEM Y HASANIEN, “*Speed Control of Switched Reluctance Motor Based on Fuzzy Logic Controller*”, www.sdaengineering.com/MEPCON10/Papers/166.pdf, 2010, Pagina consultada el 16 de marzo del 2011.
- [14] NATIONAL INSTRUMENTS, Video “*Introducción al Módulo de LabVIEW FPGA*”, zone.ni.com/wv/app/doc/p/id/wv-609, 2011, pagina consultada el 20 de abril del 2011.
- [15] INSTITUTO DE TECNOLOGÍAS EDUCATIVAS, “*Transistor*”, alerce.pntic.mec.es/~hmartin/electr%F3nica/componentes/transistor.htm, pagina consultada el 15 de junio del 2011.
- [16] RAMOS, Rafael, “*Sistemas Digitales De Control En Tiempo Discreto*”, upcommons.upc.edu/e-prints/bitstream/2117/6123/1/TEMA6.pdf, pagina consultada el 10 de junio del 2011.
- [17] BRASLAVSKY, Julio, “*5. Control PID Clásico*”, csd.newcastle.edu.au/SpanishPages/clase_slides_download/C07.pdf, pagina consultada el 17 de junio del 2011.

ANEXOS

ANEXO 1

LabVIEW FPGA Module Release and Upgrade Notes

LabVIEW™ FPGA Module Release and Upgrade Notes

Version 2011

This document contains instructions for installing the LabVIEW 2011 FPGA Module and related products, introduces new features, and provides information about upgrading from the previous version of LabVIEW. Refer to the resources listed at the end of this document for information about developing applications with LabVIEW and the FPGA Module.

Contents

System Requirements	1
Development Computer	2
Remote Compile Server	3
Installation	3
Installing the LabVIEW FPGA Module on a Development Computer	3
Installing the Compilation Tools on a Separate Computer	4
Activating the LabVIEW FPGA Module	4
Version 2011 Features and Changes	5
Improvements to Cycle-Accurate Simulation	5
Improvements to IP	5
Support for the FPGA Compile Cloud Service	6
Improving the Performance of Feedback Nodes	6
Tracking a 3-Phase Signal	6
Upgrade and Compatibility Issues	6
VIs that Contain Math & Analysis VIs and Functions	6
VIs that Contain an IP Integration Node	7
VIs that Contain a Feedback Node	7
Known Issues	7
Where to Go from Here	7
Related Documentation and Examples	7
National Instruments Web Site	8
Support	8

System Requirements

The development computer is a PC or PXI/PXI Express system on which you install the LabVIEW development system and the LabVIEW FPGA Module. You can use this computer to compile FPGA VIs, but if the FPGA design or target requires more than 2 GB of memory,

National Instruments recommends installing the LabVIEW FPGA compilation tools on a separate computer. This computer is referred to as the remote compile server.

The following sections provide information about requirements for both the development computer and the optional remote compile server.

Development Computer

The development computer must meet the following specifications.

- LabVIEW 2011 Full or Professional Edition (32-bit)
- 1.2 GHz Pentium processor or equivalent
- 11 GB additional disk space
- Memory¹:
 - At least 2 GB of RAM for Virtex-5 and earlier FPGA targets
 - At least 4 GB of RAM for Virtex-6 and later² FPGA targets
- One of the following operating systems:
 - Windows 7³
 - Windows Vista (32-bit)
 - Windows Vista (64-bit with 32-bit LabVIEW installed)
 - Windows XP Pro (Service Pack 2 or 3)
 - Windows Server 2003 R2 (32-bit)⁴
 - Windows Server 2008 R2 (64-bit)
- FPGA device—You do not have to purchase an FPGA device in order to install the LabVIEW FPGA Module; however, the software is designed to program FPGA devices from NI.
- Device driver software—To program most devices with the LabVIEW 2011 FPGA Module, install the NI-RIO 4.0 driver software. Refer to the hardware documentation for different or additional requirements for a specific device.

¹ Memory requirements vary according to both the FPGA target and the FPGA VI you create for that target. To determine your memory needs, monitor the memory usage of the `xst.exe` process in the Windows Task Manager.

² You must use a 64-bit OS to address 4 GB of RAM.

³ The LabVIEW FPGA Module uses Xilinx compilation tools that do not officially support Windows 7. NI obtained permission from Xilinx to allow LabVIEW FPGA Module customers to use the tools on this platform, with the disclaimer that Xilinx will not be able to fix any bugs found that are specific to this platform. NI tested the Xilinx tools that the LabVIEW FPGA Module uses and did not find any issues. If you encounter problems with the Xilinx tools and Windows 7, you might need to compile using Windows XP or Vista. In such cases, you might want to install the Xilinx compilation tools on a separate computer. Refer to ni.com/info and enter the Info Code `fpgakb1rcs` for information about installing the compilations tools on a separate computer. NI will not be liable for any problems or issues related to the use of Xilinx tools with Windows 7.

⁴ The LabVIEW FPGA Module does not support the Windows Server non-R2 editions.

Remote Compile Server

If you install the Xilinx compilation tools on a separate computer, that computer is the remote compile server and must meet the following specifications:

- 11 GB of disk space for the complete set of Xilinx compilation tools
- One of the following operating systems:
 - Windows XP Pro (Service Pack 2 or 3)
 - Windows Vista (32-bit)
 - Windows Vista (64-bit)

NI also offers the LabVIEW FPGA Compile Farm Toolkit and the LabVIEW FPGA Compile Cloud Service, both of which decrease your development time by offloading compiles to multiple computers. Refer to the National Instruments Web site at ni.com for information about these products.

Installation

The following sections provide information about installing the LabVIEW FPGA Module and related components.



Note If you are upgrading from a previous version of LabVIEW, you must install LabVIEW 2011 before you install the LabVIEW 2011 FPGA Module in order to mass compile existing VIs successfully. If you mass compile existing FPGA VIs before you install the LabVIEW FPGA Module, some FPGA-specific VIs might have mutation issues.

Installing the LabVIEW FPGA Module on a Development Computer

Complete the following steps to install LabVIEW and the FPGA Module.

1. Log in as an administrator or as a user with administrative privileges.
2. Insert the LabVIEW Platform DVD 1.

To request additional LabVIEW 2011 Platform DVDs, refer to the [National Instruments Web site](#). If you purchased this product with an NI Software Suite or NI Product Bundle, use the installation media that shipped with your purchase to install this product.



Tip If the installer does not start automatically, double-click `setup.exe` from the media to launch the installer.

3. Follow the instructions on the screen to install and activate the following software:
 - **LabVIEW**
 - **FPGA Module**

- **Compilation Tools**—Refer to your hardware documentation for information about the FPGA in your device and install the appropriate set of tools.
 - **Compilation Tools for Devices other than Virtex-II**—Install these compilation tools on the development computer to create simulation exports, configure the IP Integration Node, or use the Configuring CLIP wizard.
 - (Optional) **Compilation Tools for Virtex-II FPGA Devices**—Install these compilation tools only if your FPGA device contains a Xilinx Virtex-II or Virtex-II Pro FPGA.
- (Optional) **FPGA Compile Farm Toolkit**—Use the FPGA Compile Farm Toolkit to distribute FPGA VI compiles across multiple remote computers. You have a temporary license for a 30-day evaluation of this product. During this period, you can compile two FPGA VIs simultaneously. After the evaluation period expires, you can compile one FPGA VI at a time.
- (Optional) **Real-Time Module**—Use the LabVIEW Real-Time Module to program the deterministic OS on NI PXI, NI PXI Express, NI CompactRIO, and NI Single-Board RIO devices. You have a temporary license for a 30-day evaluation period of this product.
- **Device Drivers**—NI-RIO 4.0 is the device driver software for most FPGA targets. Refer to your hardware documentation for information about additional or different device drivers you need.

In addition to installing program files and documentation in the LabVIEW directory, the installer also puts files from Xilinx in the `x:\NIFPGA` directory, where `x` is the drive on which you installed LabVIEW. Xilinx is third-party software that the LabVIEW FPGA Module uses to compile FPGA VIs into code that runs on an FPGA target.

Installing the Compilation Tools on a Separate Computer

Refer to ni.com/info and enter the Info Code `fpgakb1rcs` for information about installing the Xilinx compilation tools on a computer that is not the development computer.

Activating the LabVIEW FPGA Module

The LabVIEW FPGA Module relies on licensing activation. You have a temporary license for a 30-day evaluation period that includes both the FPGA Module and the Xilinx tools that the FPGA Module uses. When the evaluation period expires, you must activate a valid FPGA Module license to continue using the FPGA Module. Activating the FPGA Module license also activates the license for the Xilinx tools.

Use the NI License Manager, available by selecting **Start»All Programs»National Instruments»NI License Manager**, to activate NI products. Refer to the *National Instruments License Manager Help*, available by selecting **Help»Contents** in the NI License Manager, for more information about activating NI products.

Version 2011 Features and Changes

The LabVIEW 2011 FPGA Module includes the following new features. Refer to the *LabVIEW Help*, available by selecting **Help»LabVIEW Help** in LabVIEW, for more information about these features.

Improvements to Cycle-Accurate Simulation

The following sections provide information about improvements made to cycle-accurate simulation.

Support for ISim

You now can use Xilinx ISim, included with the Xilinx compilation tools on the LabVIEW Platform DVD, to simulate FPGA designs.

Support for LabVIEW-Based Test Benches

You now can simulate an FPGA design using a test bench that LabVIEW creates.

Improvements to IP

The following sections provide information about the improvements made to IP you use to create FPGA VIs.

FPGA-Derived Clocks in Component-Level IP

You now can create FPGA-derived clocks from external clocks in component-level IP (CLIP).

Batch Configuration of IP Integration Node Files

LabVIEW 2011 brings the following improvements to the IP Integration Node:

- You now can set the simulation behavior for multiple synthesis files simultaneously.
- You now can set the data type for multiple input and output terminals simultaneously.

Xilinx CORE Generator IP

You now can include Xilinx CORE Generator IP in FPGA VIs. These functions are available on the **Xilinx Coregen IP Functions** palette. LabVIEW displays only the functions that the FPGA target supports.

Accessing DSP48E1 Slices on Virtex-6 FPGAs

You now can access a DSP48E1 slice of a Xilinx Virtex-6 FPGA target by adding a DSP48E1 function to the block diagram. This function is available on the **Functions** palette at **FPGA Math & Analysis»High Throughput Math»Basic Elements**. You also can use this function to access a DSP48E slice on a Virtex-5 FPGA.

Support for the FPGA Compile Cloud Service

If you purchased a subscription to the FPGA Compile Cloud Service, you can compile FPGA VIs in the cloud, freeing resources on your development computer for other tasks.

Specify the location of FPGA VI compilations by selecting **Tools»Options**, selecting the FPGA Module page from the **Options** dialog box, and navigating to the **Compile Server** section.

Improving the Performance of Feedback Nodes

For Feedback Nodes in FPGA VIs, you now can specify whether the compiler places the multiplexer before or after the register. You also can let the compiler decide the optimal placement. This feature gives you another way to meet the timing objectives of your application.

To see these options, right-click a Feedback Node, select **Properties**, and click the **FPGA Implementation** tab.

Tracking a 3-Phase Signal

Use the 3-Phase Phase-Locked Loop Express VI to track a 3-Phase signal. This Express VI is available on the **Functions** palette at **FPGA Math & Analysis»Control**.

Upgrade and Compatibility Issues

The following sections provide information about upgrading from the LabVIEW 2010 FPGA Module to the LabVIEW 2011 FPGA Module. Refer to previous editions of the *LabVIEW FPGA Module Release and Upgrade Notes*, available on ni.com, for changes in previous versions of the FPGA Module.



Note Upgrading the LabVIEW FPGA Module requires upgrading to a new version of the Xilinx compilation tools. This upgrade can change the timing or resource utilization characteristics of FPGA VIs you developed in previous versions or cause these VIs to fail to compile. NI recommends you test all FPGA designs after upgrading.

VIs that Contain Math & Analysis VIs and Functions

The Butterworth Filter and Notch Filter Express VIs with the **reset** terminal unwired or wired to a constant reset on first call when run on an FPGA target. In addition, these Express VIs use slightly more FPGA logic resources than in previous versions.

The Backlash and Friction VIs now avoid saturation and truncation in intermediate computations. Therefore, these VIs might produce different results than in previous versions.

VIs that Contain an IP Integration Node

If you open a VI saved in a previous version of LabVIEW that contains an IP Integration Node, you must regenerate support files for the IP. To complete this task, select **Tools»FPGA Module»Regenerate Support Files for IP Integration Nodes** in LabVIEW 2011. Follow the on-screen instructions to generate the support files for LabVIEW 2011.



Note LabVIEW might not be able to update all support files; in this situation, re-add the IP Integration Node to the VI and generate the support files for the new node.

VIs that Contain a Feedback Node

Existing VIs that you open in LabVIEW 2011 leave the placement of the multiplexer up to the compiler, which preserves existing behavior. Refer to the [Improving the Performance of Feedback Nodes](#) section for information about changing this behavior.

Known Issues

Refer to the National Instruments Web site at ni.com/info and enter the Info Code LVFPGA2011KI to access the known issues for the LabVIEW 2011 FPGA Module.

Where to Go from Here

NI provides many resources to help you succeed with your NI products. Use the following resources as you start exploring LabVIEW and the FPGA Module.

Related Documentation and Examples

Use the following resources to learn more about using LabVIEW and the FPGA Module.

- **Context Help Window**—Available by pressing <Ctrl-H>. This window provides brief descriptions of VIs, functions, and dialog boxes along with a link to more detailed information.
- **LabVIEW Help**—Available by selecting **Help»LabVIEW Help** in LabVIEW. Browse the **FPGA Module** book in the **Contents** tab for an overview of the FPGA Module and hardware-specific information. Browse the **FPGA Interface** book in the **Contents** tab for an overview of the FPGA Interface information.
- **Hardware Documentation**—Refer to the documentation that came with the FPGA target hardware for information about that hardware. The documentation might be installed on disk instead of printed.
- **Examples**—The driver software for many FPGA targets includes corresponding examples. Refer to the specific hardware documentation for information about whether the FPGA target you use comes with corresponding examples.

You can start with an existing example and use it as a starting point for developing FPGA VIs and host VIs. From LabVIEW, launch the NI Example Finder by selecting **Help»Find Examples**. Browse the examples by directory or by task.

National Instruments Web Site

Refer to ni.com/fpga for the latest NI Developer Zone articles, examples, and support information for the FPGA Module.

Refer to ni.com/info and enter the Info Code `fpgatrnr` to access online training for the FPGA Module.

Support

The National Instruments Web site is your complete resource for technical support. At ni.com/support you have access to everything from troubleshooting and application development self-help resources to email and phone assistance from NI Application Engineers.

National Instruments corporate headquarters is located at 11500 North Mopac Expressway, Austin, Texas, 78759-3504. National Instruments also has offices located around the world to help address your support needs. For telephone support in the United States, create your service request at ni.com/support and follow the calling instructions or dial 512 795 8248. For telephone support outside the United States, visit the Worldwide Offices section of ni.com/niglobal to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

LabVIEW, National Instruments, NI, ni.com, the National Instruments corporate logo, and the Eagle logo are trademarks of National Instruments Corporation. Refer to the *Trademark Information* at ni.com/trademarks for other National Instruments trademarks. Xilinx is the registered trademark of Xilinx, Inc. Other product and company names mentioned herein are trademarks or trade names of their respective companies. For patents covering National Instruments products/technology, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your media, or the *National Instruments Patents Notice* at ni.com/patents. Refer to the *Export Compliance Information* at ni.com/legal/export-compliance for the National Instruments global trade compliance policy and how to obtain relevant HTS codes, ECCNs, and other import/export data.

© 2003–2011 National Instruments Corporation. All rights reserved.

ANEXO 2

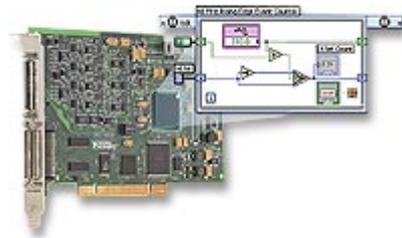
Módulo NI LabVIEW FPGA



Ventas
Ecuador
1 800 999 119 pedir enlace a 1
800 433 3488
orders@ni.com

Módulo NI LabVIEW FPGA

- Cree su propio hardware de E/S sin codificación VHDL o diseño de tablero
- Configure gráficamente FPGAs en objetivos de hardware E/S (RIO) reconfigurables
- Defina su propios algoritmos de control con velocidades de hasta 200 MHz
- Ejecute simultáneamente múltiples tareas y de manera determinante
- Implemente lógica personalizada de temporización y disparo, protocolos digitales y algoritmos DSP
- Incorpore código HDL existente e IP de terceros incluyendo funciones del Generador Xilinx CORE



Información General

NI LabVIEW y el Módulo LabVIEW FPGA proporcionan un ambiente de desarrollo gráfico para arreglos de compuertas programables en campo (FPGAs) en objetivos de hardware de E/S (RIO) reconfigurables. Con el Módulo LabVIEW FPGA, usted puede desarrollar VIs FPGA en un servidor ejecutando Windows y LabVIEW compila e implementa el código de hardware. Puede crear VIs de FPGA embebidos que combinan acceso directo a E/S con la lógica de LabVIEW definida por el usuario para definir hardware personalizado para aplicaciones como protocolos de comunicación digital, simulación de control en hardware y rápida generación de prototipos de control.

Mientras el Módulo LabVIEW FPGA contiene varias rutinas integradas de procesamiento de señales, usted también puede integrar código HDL existente, así como IP de terceros incluyendo funciones del Generador Xilinx CORE. Además, LabVIEW FPGA se integra con Mentor Graphics ModelSim (se de comprar por separado) y herramientas Xilinx ISE incluidas para simulación de lógica de ciclo preciso.

Con el Programa de Servicio Estándar (SSP), usted recibe lo último en tecnología de software de NI a través de actualizaciones automáticas y un soporte técnico de excelente nivel por teléfono o correo electrónico de parte de ingenieros de aplicación de NI. También recibe acceso gratis e ilimitado a cursos de capacitación en línea en vivo. El costo de una suscripción anual a SSP es menor que la compra de una sola actualización.

Usted puede comprar de forma opcional el Módulo LabVIEW FPGA como parte de un paquete NI Developer Suite. NI Developer Suite combina todas las herramientas necesarias para construir sistemas embebidos con objetivos de hardware como CompactRIO: LabVIEW FPGA para sintetizar lógica personalizada en el FPGA RIO configurable por el usuario, LabVIEW Real-Time para construir aplicaciones determinísticas en tiempo real y más.

ANEXO 3

Spartan-3 FPGA Family: Introduction and Ordering Information

Introduction

The Spartan[®]-3 family of Field-Programmable Gate Arrays is specifically designed to meet the needs of high volume, cost-sensitive consumer electronic applications. The eight-member family offers densities ranging from 50,000 to five million system gates, as shown in [Table 1](#).

The Spartan-3 family builds on the success of the earlier Spartan-II family by increasing the amount of logic resources, the capacity of internal RAM, the total number of I/Os, and the overall level of performance as well as by improving clock management functions. Numerous enhancements derive from the Virtex[®]-II platform technology. These Spartan-3 FPGA enhancements, combined with advanced process technology, deliver more functionality and bandwidth per dollar than was previously possible, setting new standards in the programmable logic industry.

Because of their exceptionally low cost, Spartan-3 FPGAs are ideally suited to a wide range of consumer electronics applications, including broadband access, home networking, display/projection and digital television equipment.

The Spartan-3 family is a superior alternative to mask programmed ASICs. FPGAs avoid the high initial cost, the lengthy development cycles, and the inherent inflexibility of conventional ASICs. Also, FPGA programmability permits design upgrades in the field with no hardware replacement necessary, an impossibility with ASICs.

Features

- Low-cost, high-performance logic solution for high-volume, consumer-oriented applications
 - Densities up to 74,880 logic cells
- SelectIO[™] interface signaling
 - Up to 633 I/O pins
 - 622+ Mb/s data transfer rate per I/O
 - 18 single-ended signal standards
 - 8 differential I/O standards including LVDS, RSDS
 - Termination by Digitally Controlled Impedance
 - Signal swing ranging from 1.14V to 3.465V
 - Double Data Rate (DDR) support
 - [DDR, DDR2 SDRAM support](#) up to 333 Mbps
- Logic resources
 - Abundant logic cells with shift register capability
 - Wide, fast multiplexers
 - Fast look-ahead carry logic
 - Dedicated 18 x 18 multipliers
 - JTAG logic compatible with IEEE 1149.1/1532
- SelectRAM[™] hierarchical memory
 - Up to 1,872 Kbits of total block RAM
 - Up to 520 Kbits of total distributed RAM
- Digital Clock Manager (up to four DCMs)
 - Clock skew elimination
 - Frequency synthesis
 - High resolution phase shifting
- Eight global clock lines and abundant routing
- Fully supported by [Xilinx ISE[®]](#) and [WebPACK[™]](#) software development systems
- [MicroBlaze[™]](#) and [PicoBlaze[™]](#) processor, [PCI[®]](#), [PCI Express[®]](#) [PIPE Endpoint](#), and other [IP cores](#)
- Pb-free packaging options
- Automotive [Spartan-3 XA Family](#) variant

Table 1: Summary of Spartan-3 FPGA Attributes

Device	System Gates	Equivalent Logic Cells ⁽¹⁾	CLB Array (One CLB = Four Slices)			Distributed RAM Bits (K=1024)	Block RAM Bits (K=1024)	Dedicated Multipliers	DCMs	Maximum User I/O	Maximum Differential I/O Pairs
			Rows	Columns	Total CLBs						
XC3S50 ⁽²⁾	50K	1,728	16	12	192	12K	72K	4	2	124	56
XC3S200 ⁽²⁾	200K	4,320	24	20	480	30K	216K	12	4	173	76
XC3S400 ⁽²⁾	400K	8,064	32	28	896	56K	288K	16	4	264	116
XC3S1000 ⁽²⁾	1M	17,280	48	40	1,920	120K	432K	24	4	391	175
XC3S1500	1.5M	29,952	64	52	3,328	208K	576K	32	4	487	221
XC3S2000	2M	46,080	80	64	5,120	320K	720K	40	4	565	270
XC3S4000	4M	62,208	96	72	6,912	432K	1,728K	96	4	633	300
XC3S5000	5M	74,880	104	80	8,320	520K	1,872K	104	4	633	300

Notes:

1. Logic Cell = 4-input Look-Up Table (LUT) plus a 'D' flip-flop. "Equivalent Logic Cells" equals "Total CLBs" x 8 Logic Cells/CLB x 1.125 effectiveness.
2. These devices are available in Xilinx Automotive versions as described in [DS314](#): Spartan-3 Automotive XA FPGA Family.

Architectural Overview

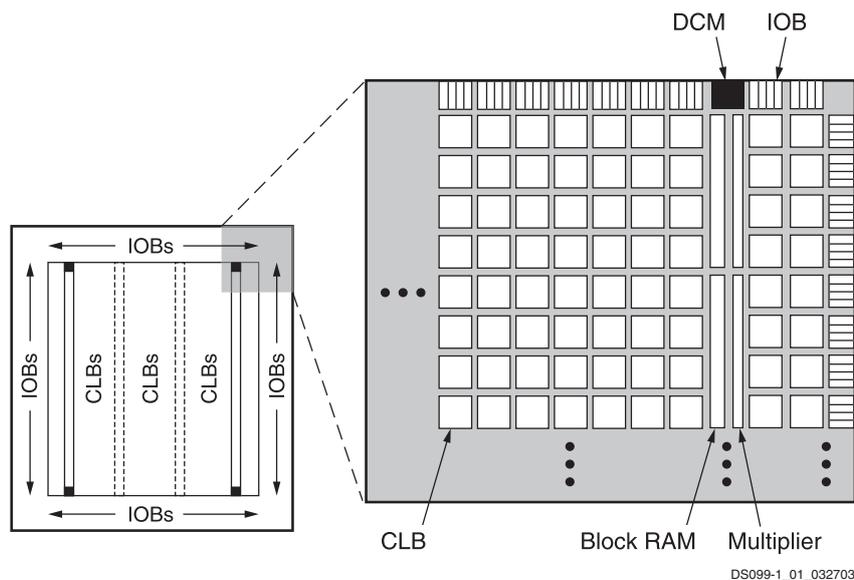
The Spartan-3 family architecture consists of five fundamental programmable functional elements:

- Configurable Logic Blocks (CLBs) contain RAM-based Look-Up Tables (LUTs) to implement logic and storage elements that can be used as flip-flops or latches. CLBs can be programmed to perform a wide variety of logical functions as well as to store data.
- Input/Output Blocks (IOBs) control the flow of data between the I/O pins and the internal logic of the device. Each IOB supports bidirectional data flow plus 3-state operation. Twenty-six different signal standards, including eight high-performance differential standards, are available as shown in [Table 2](#). Double Data-Rate (DDR) registers are included. The Digitally Controlled Impedance (DCI) feature provides automatic on-chip terminations, simplifying board designs.
- Block RAM provides data storage in the form of 18-Kbit dual-port blocks.
- Multiplier blocks accept two 18-bit binary numbers as inputs and calculate the product.

- Digital Clock Manager (DCM) blocks provide self-calibrating, fully digital solutions for distributing, delaying, multiplying, dividing, and phase shifting clock signals.

These elements are organized as shown in [Figure 1](#). A ring of IOBs surrounds a regular array of CLBs. The XC3S50 has a single column of block RAM embedded in the array. Those devices ranging from the XC3S200 to the XC3S2000 have two columns of block RAM. The XC3S4000 and XC3S5000 devices have four RAM columns. Each column is made up of several 18-Kbit RAM blocks; each block is associated with a dedicated multiplier. The DCMs are positioned at the ends of the outer block RAM columns.

The Spartan-3 family features a rich network of traces and switches that interconnect all five functional elements, transmitting signals among them. Each functional element has an associated switch matrix that permits multiple connections to the routing.



Notes:

1. The two additional block RAM columns of the XC3S4000 and XC3S5000 devices are shown with dashed lines. The XC3S50 has only the block RAM column on the far left.

Figure 1: Spartan-3 Family Architecture

Configuration

Spartan-3 FPGAs are programmed by loading configuration data into robust, reprogrammable, static CMOS configuration latches (CCLs) that collectively control all functional elements and routing resources. Before powering on the FPGA, configuration data is stored externally in a PROM or some other nonvolatile medium either on or off the board. After applying power, the configuration data is written to the FPGA using any of five different modes: Master Parallel, Slave Parallel, Master Serial, Slave Serial, and Boundary Scan (JTAG). The Master and Slave Parallel modes use an 8-bit-wide SelectMAP port.

The recommended memory for storing the configuration data is the low-cost Xilinx Platform Flash PROM family, which includes the XCF00S PROMs for serial configuration and the higher density XCF00P PROMs for parallel or serial configuration.

I/O Capabilities

The SelectIO feature of Spartan-3 devices supports 18 single-ended standards and 8 differential standards as listed in Table 2. Many standards support the DCI feature, which uses integrated terminations to eliminate unwanted signal reflections.

Table 2: Signal Standards Supported by the Spartan-3 Family

Standard Category	Description	V _{CCO} (V)	Class	Symbol (IOSTANDARD)	DCI Option
Single-Ended					
GTL	Gunning Transceiver Logic	N/A	Terminated	GTL	Yes
			Plus	GTL_P	Yes
HSTL	High-Speed Transceiver Logic	1.5	I	HSTL_I	Yes
			III	HSTL_III	Yes
		1.8	I	HSTL_I_18	Yes
			II	HSTL_II_18	Yes
LVCMOS	Low-Voltage CMOS	1.2	N/A	LVCMOS12	No
		1.5	N/A	LVCMOS15	Yes
		1.8	N/A	LVCMOS18	Yes
		2.5	N/A	LVCMOS25	Yes
		3.3	N/A	LVCMOS33	Yes
LVTTTL	Low-Voltage Transistor-Transistor Logic	3.3	N/A	LVTTTL	No
PCI	Peripheral Component Interconnect	3.0	33 MHz ⁽¹⁾	PCI33_3	No
SSTL	Stub Series Terminated Logic	1.8	N/A (± 6.7 mA)	SSTL18_I	Yes
			N/A (± 13.4 mA)	SSTL18_II	No
		2.5	I	SSTL2_I	Yes
			II	SSTL2_II	Yes
Differential					
LDT (ULVDS)	Lightning Data Transport (HyperTransport™) Logic	2.5	N/A	LDT_25	No
LVDS	Low-Voltage Differential Signaling		Standard	LVDS_25	Yes
			Bus	BLVDS_25	No
			Extended Mode	LVDSEXT_25	Yes
LVPECL	Low-Voltage Positive Emitter-Coupled Logic	2.5	N/A	LVPECL_25	No
RSDS	Reduced-Swing Differential Signaling	2.5	N/A	RSDS_25	No
HSTL	Differential High-Speed Transceiver Logic	1.8	II	DIFF_HSTL_II_18	Yes
SSTL	Differential Stub Series Terminated Logic	2.5	II	DIFF_SSTL2_II	Yes

Notes:

- 66 MHz PCI is not supported by the Xilinx IP core although PCI66_3 is an available I/O standard.

Table 3 shows the number of user I/Os as well as the number of differential I/O pairs available for each device/package combination.

Table 3: Spartan-3 Device I/O Chart

Package	Available User I/Os and Differential (Diff) I/O Pairs by Package Type																			
	VQ100 VQG100		CP132 ⁽¹⁾ CPG132		TQ144 TQG144		PQ208 PQG208		FT256 FTG256		FG320 FGG320		FG456 FGG456		FG676 FGG676		FG900 FGG900		FG1156 ⁽¹⁾ FGG1156	
	16 x 16		8 x 8		22 x 22		30.6 x 30.6		17 x 17		19 x 19		23 x 23		27 x 27		31 x 31		35 x 35	
Footprint (mm)	User	Diff	User	Diff	User	Diff	User	Diff	User	Diff	User	Diff	User	Diff	User	Diff	User	Diff	User	Diff
XC3S50	63	29	89 ⁽¹⁾	44 ⁽¹⁾	97	46	124	56	-	-	-	-	-	-	-	-	-	-	-	-
XC3S200	63	29	-	-	97	46	141	62	173	76	-	-	-	-	-	-	-	-	-	-
XC3S400	-	-	-	-	97	46	141	62	173	76	221	100	264	116	-	-	-	-	-	-
XC3S1000	-	-	-	-	-	-	-	-	173	76	221	100	333	149	391	175	-	-	-	-
XC3S1500	-	-	-	-	-	-	-	-	-	-	221	100	333	149	487	221	-	-	-	-
XC3S2000	-	-	-	-	-	-	-	-	-	-	-	-	333	149	489	221	565	270	-	-
XC3S4000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	489	221	633	300	712 ⁽¹⁾	312 ⁽¹⁾
XC3S5000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	489	221	633	300	784 ⁽¹⁾	344 ⁽¹⁾

Notes:

1. The CP132, CPG132, FG1156, and FGG1156 packages are being discontinued and are not recommended for new designs. See http://www.xilinx.com/support/documentation/spartan-3_customer_notices.htm for the latest updates.
2. All device options listed in a given package column are pin-compatible.
3. User = Single-ended user I/O pins. Diff = Differential I/O pairs.

ANEXO 4

KA78XX/KA78XXA Datasheet

KA78XX/KA78XXA

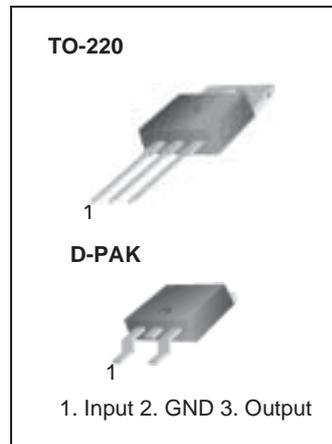
3-Terminal 1A Positive Voltage Regulator

Features

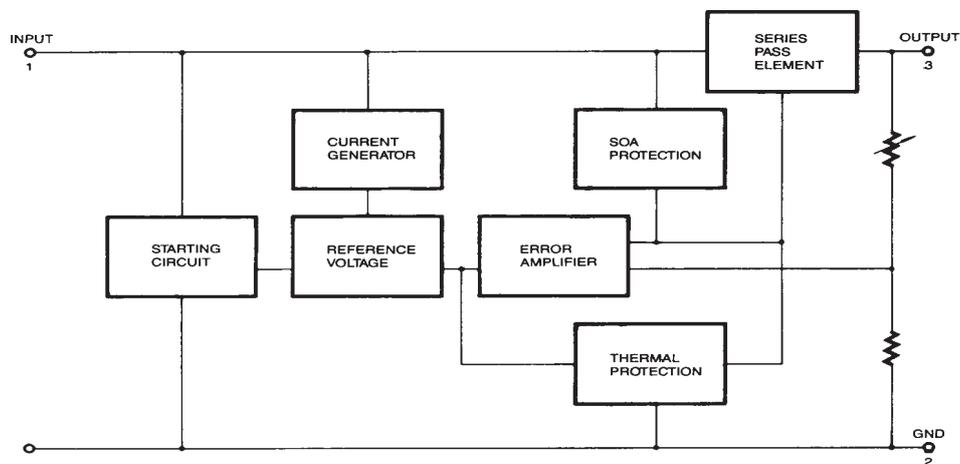
- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

Description

The KA78XX/KA78XXA series of three-terminal positive regulator are available in the TO-220/D-PAK package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.



Internal Block Diagram



Absolute Maximum Ratings

Parameter	Symbol	Value	Unit
Input Voltage (for $V_O = 5V$ to $18V$) (for $V_O = 24V$)	V_I	35	V
	V_I	40	V
Thermal Resistance Junction-Cases (TO-220)	$R_{\theta JC}$	5	$^{\circ}C/W$
Thermal Resistance Junction-Air (TO-220)	$R_{\theta JA}$	65	$^{\circ}C/W$
Operating Temperature Range (KA78XX/A/R)	T_{OPR}	0 ~ +125	$^{\circ}C$
Storage Temperature Range	T_{STG}	-65 ~ +150	$^{\circ}C$

Electrical Characteristics (KA7805/KA7805R)

(Refer to test circuit , $0^{\circ}C < T_J < 125^{\circ}C$, $I_O = 500mA$, $V_I = 10V$, $C_I = 0.33\mu F$, $C_O = 0.1\mu F$, unless otherwise specified)

Parameter	Symbol	Conditions	KA7805			Unit	
			Min.	Typ.	Max.		
Output Voltage	V_O	$T_J = +25^{\circ}C$	4.8	5.0	5.2	V	
		$5.0mA \leq I_O \leq 1.0A$, $P_O \leq 15W$ $V_I = 7V$ to $20V$	4.75	5.0	5.25		
Line Regulation (Note1)	Regline	$T_J = +25^{\circ}C$	$V_O = 7V$ to $25V$	-	4.0	100	mV
			$V_I = 8V$ to $12V$	-	1.6	50	
Load Regulation (Note1)	Regload	$T_J = +25^{\circ}C$	$I_O = 5.0mA$ to $1.5A$	-	9	100	mV
			$I_O = 250mA$ to $750mA$	-	4	50	
Quiescent Current	I_Q	$T_J = +25^{\circ}C$	-	5.0	8.0	mA	
Quiescent Current Change	ΔI_Q	$I_O = 5mA$ to $1.0A$	-	0.03	0.5	mA	
		$V_I = 7V$ to $25V$	-	0.3	1.3		
Output Voltage Drift	$\Delta V_O / \Delta T$	$I_O = 5mA$	-	-0.8	-	mV/ $^{\circ}C$	
Output Noise Voltage	V_N	$f = 10Hz$ to $100KHz$, $T_A = +25^{\circ}C$	-	42	-	$\mu V/V_O$	
Ripple Rejection	RR	$f = 120Hz$ $V_O = 8V$ to $18V$	62	73	-	dB	
Dropout Voltage	V_{Drop}	$I_O = 1A$, $T_J = +25^{\circ}C$	-	2	-	V	
Output Resistance	r_O	$f = 1KHz$	-	15	-	m Ω	
Short Circuit Current	I_{SC}	$V_I = 35V$, $T_A = +25^{\circ}C$	-	230	-	mA	
Peak Current	I_{PK}	$T_J = +25^{\circ}C$	-	2.2	-	A	

Note:

1. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.

Electrical Characteristics (KA7806/KA7806R)

(Refer to test circuit , $0^{\circ}\text{C} < T_J < 125^{\circ}\text{C}$, $I_O = 500\text{mA}$, $V_I = 11\text{V}$, $C_I = 0.33\mu\text{F}$, $C_O = 0.1\mu\text{F}$, unless otherwise specified)

Parameter	Symbol	Conditions	KA7806			Unit	
			Min.	Typ.	Max.		
Output Voltage	V_O	$T_J = +25^{\circ}\text{C}$	5.75	6.0	6.25	V	
		$5.0\text{mA} \leq I_O \leq 1.0\text{A}$, $P_O \leq 15\text{W}$ $V_I = 8.0\text{V to } 21\text{V}$	5.7	6.0	6.3		
Line Regulation (Note1)	Regline	$T_J = +25^{\circ}\text{C}$	$V_I = 8\text{V to } 25\text{V}$	-	5	120	mV
			$V_I = 9\text{V to } 13\text{V}$	-	1.5	60	
Load Regulation (Note1)	Regload	$T_J = +25^{\circ}\text{C}$	$I_O = 5\text{mA to } 1.5\text{A}$	-	9	120	mV
			$I_O = 250\text{mA to } 750\text{mA}$	-	3	60	
Quiescent Current	I_Q	$T_J = +25^{\circ}\text{C}$	-	5.0	8.0	mA	
Quiescent Current Change	ΔI_Q	$I_O = 5\text{mA to } 1\text{A}$	-	-	0.5	mA	
		$V_I = 8\text{V to } 25\text{V}$	-	-	1.3		
Output Voltage Drift	$\Delta V_O / \Delta T$	$I_O = 5\text{mA}$	-	-0.8	-	mV/ $^{\circ}\text{C}$	
Output Noise Voltage	V_N	$f = 10\text{Hz to } 100\text{KHz}$, $T_A = +25^{\circ}\text{C}$	-	45	-	$\mu\text{V}/V_O$	
Ripple Rejection	RR	$f = 120\text{Hz}$ $V_I = 9\text{V to } 19\text{V}$	59	75	-	dB	
Dropout Voltage	V_{Drop}	$I_O = 1\text{A}$, $T_J = +25^{\circ}\text{C}$	-	2	-	V	
Output Resistance	r_O	$f = 1\text{KHz}$	-	19	-	$\text{m}\Omega$	
Short Circuit Current	I_{SC}	$V_I = 35\text{V}$, $T_A = +25^{\circ}\text{C}$	-	250	-	mA	
Peak Current	I_{PK}	$T_J = +25^{\circ}\text{C}$	-	2.2	-	A	

Note:

1. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.

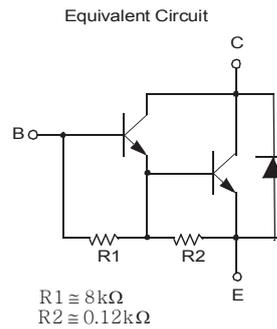
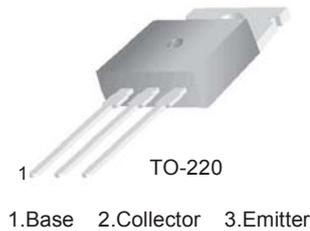
ANEXO 5

TIP120/TIP121/TIP122 Datasheet

TIP120/TIP121/TIP122

NPN Epitaxial Darlington Transistor

- Medium Power Linear Switching Applications
- Complementary to TIP125/126/127



Absolute Maximum Ratings* $T_a = 25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Ratings	Units
V_{CBO}	Collector-Base Voltage : TIP120	60	V
	: TIP121	80	V
	: TIP122	100	V
V_{CEO}	Collector-Emitter Voltage : TIP120	60	V
	: TIP121	80	V
	: TIP122	100	V
V_{EBO}	Emitter-Base Voltage	5	V
I_C	Collector Current (DC)	5	A
I_{CP}	Collector Current (Pulse)	8	A
I_B	Base Current (DC)	120	mA
P_C	Collector Dissipation ($T_a=25^\circ\text{C}$)	2	W
	Collector Dissipation ($T_C=25^\circ\text{C}$)	65	W
T_J	Junction Temperature	150	$^\circ\text{C}$
T_{STG}	Storage Temperature	- 65 ~ 150	$^\circ\text{C}$

* These ratings are limiting values above which the serviceability of any semiconductor device may be impaired.

Electrical Characteristics* $T_a=25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units
$V_{CEO(sus)}$	Collector-Emitter Sustaining Voltage : TIP120 : TIP121 : TIP122	$I_C = 100\text{mA}, I_B = 0$	60 80 100			V V V
I_{CEO}	Collector Cut-off Current : TIP120 : TIP121 : TIP122	$V_{CE} = 30\text{V}, I_B = 0$ $V_{CE} = 40\text{V}, I_B = 0$ $V_{CE} = 50\text{V}, I_B = 0$			0.5 0.5 0.5	mA mA mA
I_{CBO}	Collector Cut-off Current : TIP120 : TIP121 : TIP122	$V_{CB} = 60\text{V}, I_E = 0$ $V_{CB} = 80\text{V}, I_E = 0$ $V_{CB} = 100\text{V}, I_E = 0$			0.2 0.2 0.2	mA mA mA
I_{EBO}	Emitter Cut-off Current	$V_{BE} = 5\text{V}, I_C = 0$			2	mA
h_{FE}	* DC Current Gain	$V_{CE} = 3\text{V}, I_C = 0.5\text{A}$ $V_{CE} = 3\text{V}, I_C = 3\text{A}$	1000 1000			
$V_{CE(sat)}$	* Collector-Emitter Saturation Voltage	$I_C = 3\text{A}, I_B = 12\text{mA}$ $I_C = 5\text{A}, I_B = 20\text{mA}$			2.0 4.0	V V
$V_{BE(on)}$	* Base-Emitter On Voltage	$V_{CE} = 3\text{V}, I_C = 3\text{A}$			2.5	V
C_{ob}	Output Capacitance	$V_{CB} = 10\text{V}, I_E = 0, f = 0.1\text{MHz}$			200	pF

* Pulse Test: Pulse Width $\leq 300\mu\text{s}$, Duty Cycle $\leq 2\%$

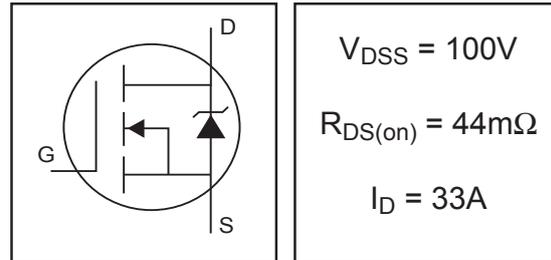
ANEXO 6

IRF540N Datasheet

IRF540N

HEXFET® Power MOSFET

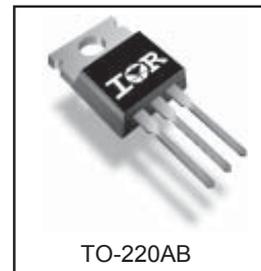
- Advanced Process Technology
- Ultra Low On-Resistance
- Dynamic dv/dt Rating
- 175°C Operating Temperature
- Fast Switching
- Fully Avalanche Rated



Description

Advanced HEXFET® Power MOSFETs from International Rectifier utilize advanced processing techniques to achieve extremely low on-resistance per silicon area. This benefit, combined with the fast switching speed and ruggedized device design that HEXFET power MOSFETs are well known for, provides the designer with an extremely efficient and reliable device for use in a wide variety of applications.

The TO-220 package is universally preferred for all commercial-industrial applications at power dissipation levels to approximately 50 watts. The low thermal resistance and low package cost of the TO-220 contribute to its wide acceptance throughout the industry.



Absolute Maximum Ratings

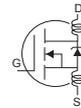
	Parameter	Max.	Units
$I_D @ T_C = 25^\circ C$	Continuous Drain Current, $V_{GS} @ 10V$	33	A
$I_D @ T_C = 100^\circ C$	Continuous Drain Current, $V_{GS} @ 10V$	23	
I_{DM}	Pulsed Drain Current ①	110	
$P_D @ T_C = 25^\circ C$	Power Dissipation	130	W
	Linear Derating Factor	0.87	W/°C
V_{GS}	Gate-to-Source Voltage	± 20	V
I_{AR}	Avalanche Current ①	16	A
E_{AR}	Repetitive Avalanche Energy ①	13	mJ
dv/dt	Peak Diode Recovery dv/dt ③	7.0	V/ns
T_J	Operating Junction and	-55 to + 175	°C
T_{STG}	Storage Temperature Range		
	Soldering Temperature, for 10 seconds		
	Mounting torque, 6-32 or M3 screw	10 lbf•in (1.1N•m)	

Thermal Resistance

	Parameter	Typ.	Max.	Units
$R_{\theta JC}$	Junction-to-Case	—	1.15	°C/W
$R_{\theta CS}$	Case-to-Sink, Flat, Greased Surface	0.50	—	
$R_{\theta JA}$	Junction-to-Ambient	—	62	

Electrical Characteristics @ T_J = 25°C (unless otherwise specified)

	Parameter	Min.	Typ.	Max.	Units	Conditions
V _{(BR)DSS}	Drain-to-Source Breakdown Voltage	100	—	—	V	V _{GS} = 0V, I _D = 250μA
ΔV _{(BR)DSS/ΔT_J}	Breakdown Voltage Temp. Coefficient	—	0.12	—	V/°C	Reference to 25°C, I _D = 1mA
R _{DS(on)}	Static Drain-to-Source On-Resistance	—	—	44	mΩ	V _{GS} = 10V, I _D = 16A ④
V _{GS(th)}	Gate Threshold Voltage	2.0	—	4.0	V	V _{DS} = V _{GS} , I _D = 250μA
g _{fs}	Forward Transconductance	21	—	—	S	V _{DS} = 50V, I _D = 16A④
I _{DSS}	Drain-to-Source Leakage Current	—	—	25	μA	V _{DS} = 100V, V _{GS} = 0V
		—	—	250	μA	V _{DS} = 80V, V _{GS} = 0V, T _J = 150°C
I _{GSS}	Gate-to-Source Forward Leakage	—	—	100	nA	V _{GS} = 20V
	Gate-to-Source Reverse Leakage	—	—	-100	nA	V _{GS} = -20V
Q _g	Total Gate Charge	—	—	71	nC	I _D = 16A
Q _{gs}	Gate-to-Source Charge	—	—	14	nC	V _{DS} = 80V
Q _{gd}	Gate-to-Drain ("Miller") Charge	—	—	21	nC	V _{GS} = 10V, See Fig. 6 and 13
t _{d(on)}	Turn-On Delay Time	—	11	—	ns	V _{DD} = 50V
t _r	Rise Time	—	35	—		I _D = 16A
t _{d(off)}	Turn-Off Delay Time	—	39	—		R _G = 5.1Ω
t _f	Fall Time	—	35	—		V _{GS} = 10V, See Fig. 10 ④
L _D	Internal Drain Inductance	—	4.5	—	nH	Between lead, 6mm (0.25in.) from package and center of die contact
L _S	Internal Source Inductance	—	7.5	—		
C _{iss}	Input Capacitance	—	1960	—	pF	V _{GS} = 0V
C _{oss}	Output Capacitance	—	250	—		V _{DS} = 25V
C _{rss}	Reverse Transfer Capacitance	—	40	—		f = 1.0MHz, See Fig. 5
E _{AS}	Single Pulse Avalanche Energy②	—	700⑤	185⑥		mJ



Source-Drain Ratings and Characteristics

	Parameter	Min.	Typ.	Max.	Units	Conditions
I _S	Continuous Source Current (Body Diode)	—	—	33	A	MOSFET symbol showing the integral reverse p-n junction diode.
I _{SM}	Pulsed Source Current (Body Diode)①	—	—	110		
V _{SD}	Diode Forward Voltage	—	—	1.2	V	T _J = 25°C, I _S = 16A, V _{GS} = 0V ④
t _{rr}	Reverse Recovery Time	—	115	170	ns	T _J = 25°C, I _F = 16A
Q _{rr}	Reverse Recovery Charge	—	505	760	nC	di/dt = 100A/μs ④
t _{on}	Forward Turn-On Time	Intrinsic turn-on time is negligible (turn-on is dominated by L _S +L _D)				

Notes:

- ① Repetitive rating; pulse width limited by max. junction temperature. (See fig. 11)
- ② Starting T_J = 25°C, L = 1.5mH
R_G = 25Ω, I_{AS} = 16A. (See Figure 12)
- ③ I_{SD} ≤ 16A, di/dt ≤ 340A/μs, V_{DD} ≤ V_{(BR)DSS},
T_J ≤ 175°C
- ④ Pulse width ≤ 400μs; duty cycle ≤ 2%.
- ⑤ This is a typical value at device destruction and represents operation outside rated limits.
- ⑥ This is a calculated value limited to T_J = 175°C .

ANEXO 7

GP1A30R Datasheet

GP1A30R

OPIC Photointerrupter with Encoder Function

■ Features

1. 2-phase (A, B) digital output
2. Possible to use plastic disk
3. High sensing accuracy
(Disk slit pitch : 0.7mm)
4. TTL compatible output
5. Compact and light

■ Applications

1. Electronic typewriters, printers
2. Numerical control machines

■ Absolute Maximum Ratings (Ta= 25°C)

	Parameter	Symbol	Rating	Unit
Input	Forward current	I _F	65	mA
	*1Peak forward current	I _{FM}	1	A
	Reverse voltage	V _R	6	V
	Power dissipation	P	100	mW
Output	Supply voltage	V _{CC}	7	V
	Low level output current	I _{OL}	20	mA
	Power dissipation	P _O	250	mW
Operating temperature		T _{opr}	0 to + 70	°C
Storage temperature		T _{stg}	- 40 to + 80	°C
*2Soldering temperature		T _{sol}	260	°C

*1 Pulse width <= 100μs, Duty ratio= 0.01

*2 For 5 seconds

■ Electro-optical Characteristics

(Unless otherwise specified, Ta = 0 to + 70°C)

	Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input	Forward voltage	V _F	Ta = 25°C, I _F = 30mA	-	1.2	1.5	V
	Reverse current	I _R	Ta = 25°C, V _R = 3V	-	-	10	μA
Output	Operating supply voltage	V _{CC}		4.5	5.0	5.5	V
	High level output voltage	V _{OH}	*3V _{CC} = 5V, I _F = 30mA	2.4	4.9	-	V
	Low level output voltage	V _{OL}	*3I _{OL} = 8mA, V _{CC} = 5V, I _F = 30mA	-	0.1	0.4	V
	Supply current	I _{CC}	*3*4I _F = 30mA, V _{CC} = 5V	-	5	20	mA
Transfer characteristics	Duty ratio	*5D _A	V _{CC} = 5V, I _F = 30mA,	20	50	80	%
		*5D _B	*3f = 2.5kHz	20	50	80	%
	Response frequency	f _{MAX.}	*3V _{CC} = 5V, I _F = 30mA	-	-	5	kHz

*3 Measured under the condition shown in Measurement Conditions.

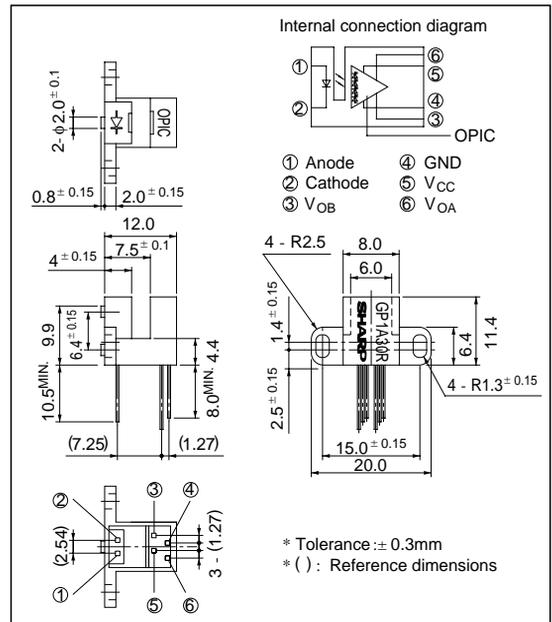
*4 In the condition that output A and B are low level.

*5

$$D_A = \frac{t_{AH}}{t_{AP}} \times 100, \quad D_B = \frac{t_{BH}}{t_{BP}} \times 100$$

■ Outline Dimensions

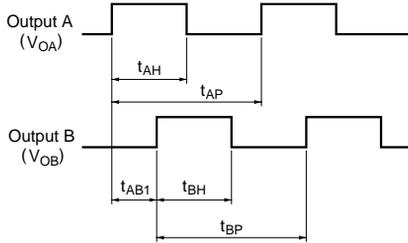
(Unit : mm)



** OPIC™ (Optical IC) is a trademark of the SHARP Corporation.

An OPIC consists of a light-detecting element and signal-processing circuit integrated onto a single chip.

■ Output Waveforms



Rotational direction: Counterclockwise when seen from OPIC light detector

Fig. 1 Forward Current vs. Ambient Temperature

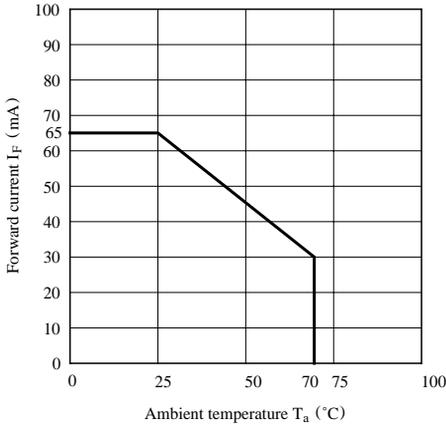


Fig. 2 Output Power Dissipation vs. Ambient Temperature

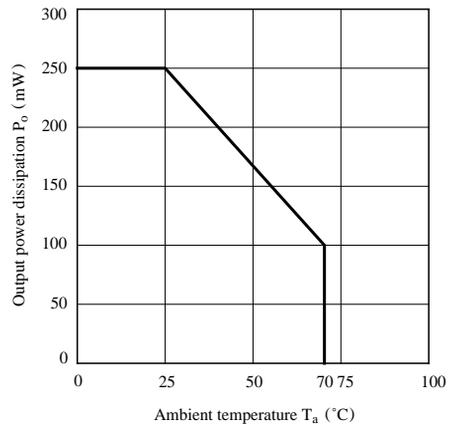


Fig. 3 Duty Ratio vs. Frequency

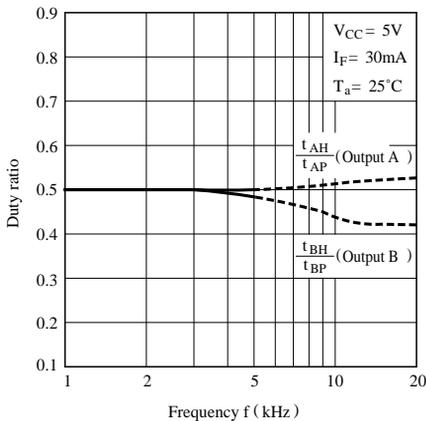


Fig. 4 Phase Difference vs. Frequency

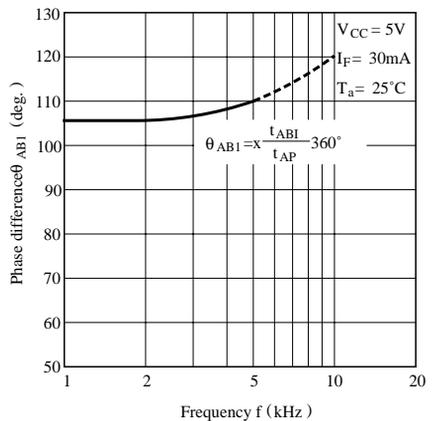


Fig. 5 Duty Ratio vs. Ambient Temperature

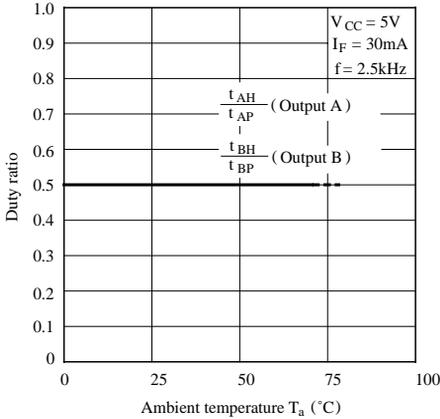


Fig. 6 Phase Difference vs. Ambient Temperature

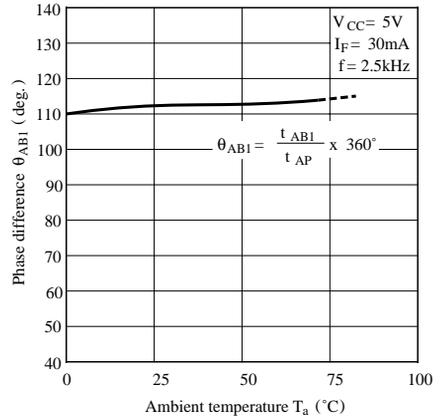


Fig. 7 Duty Ratio vs. Distance (X direction)

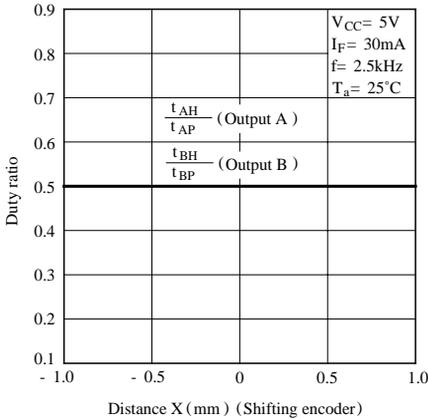


Fig. 8 Phase Difference vs. Distance (X direction)

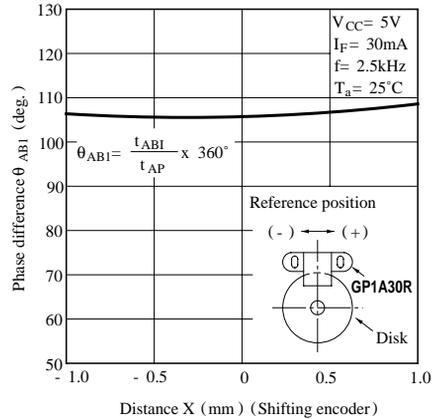


Fig. 9 Duty Ratio vs. Distance (Y direction)

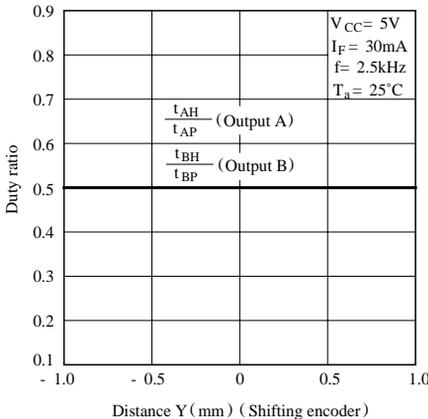


Fig.10 Phase Difference vs. Distance (Y direction)

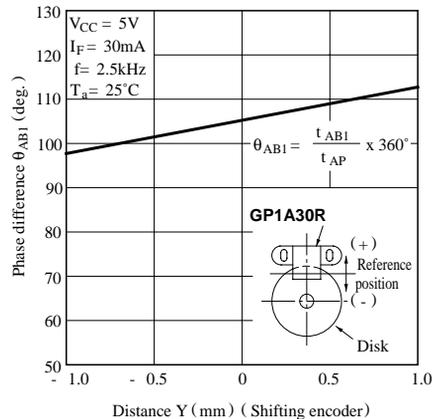


Fig.11 Duty Ratio vs. Distance (Z direction)

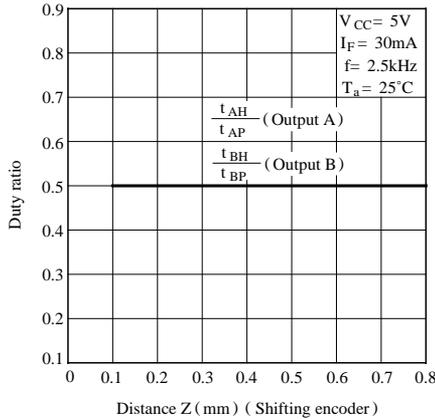
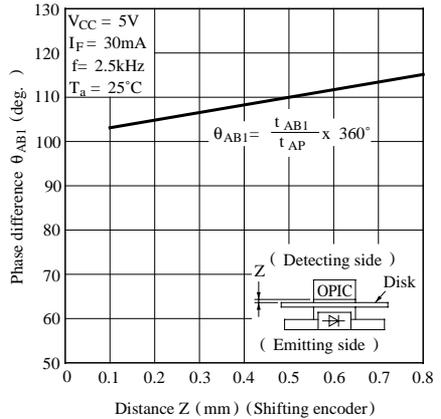
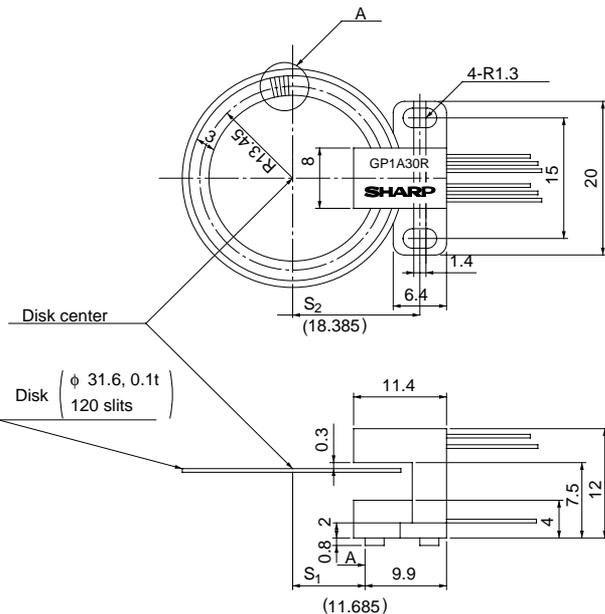


Fig.12 Phase Difference vs. Distance (Z direction)



■ Measurement Conditions



■ Precautions for Use

- (1) This module is designed to be operated at $I_F = 30mA$ TYP.
- (2) Fixing torque : MAX. 0.6Nm (6kgf • cm)
- (3) In order to stabilize power supply line, connect a by-pass capacitor of more than $0.01\mu F$ between V_{CC} and GND near the device.
- (4) As for other general cautions, refer to the chapter “Precautions for Use” .

<Basic Design>

R_0 (distance between the disk center and half point of a slit),
 P (slit pitch), S_1 and S_2 (installing position of photointerrupter) will be provided by the following equations.

Slit pitch : P (slit center)

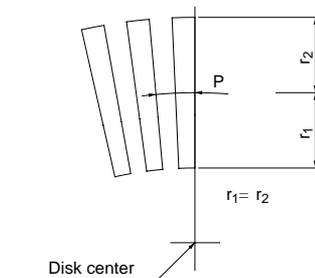
$$R_0 = \frac{N}{120} \times 13.45 \text{ (mm)} \quad N: \text{ number of slits}$$

$$P = \frac{2 \times p \times R_0}{N} \text{ (mm)}$$

$$S_1 = R_0 - 1.765 \text{ (mm)}, \quad S_2 = S_1 + 6.7 \text{ (mm)}$$

Note) When the number of slits is changed, values in parenthesis are also changed according to the number.

Enlarged drawing of A portion
 Slit pitch : P



(Ex.) In the case of
 $N = 200P/R$

$$R_0 = \frac{200}{120} \times 13.45 \text{ (mm)}$$

$$= 22.42 \text{mm}$$

$$P = \frac{2 \times p \times 22.42}{200} \text{ (mm)}$$

$$= 0.704 \text{mm}$$

$$S_1 = 22.42 - 1.765$$

$$= 20.655 \text{mm}$$

$$S_2 = 20.655 + 6.7$$

$$= 27.355 \text{mm}$$

ANEXO 8

4n25 Datasheet



6-Pin DIP Optoisolators Transistor Output

The 4N25/A, 4N26, 4N27 and 4N28 devices consist of a gallium arsenide infrared emitting diode optically coupled to a monolithic silicon phototransistor detector.

- Most Economical Optoisolator Choice for Medium Speed, Switching Applications
- Meets or Exceeds All JEDEC Registered Specifications
- **To order devices that are tested and marked per VDE 0884 requirements, the suffix "V" must be included at end of part number. VDE 0884 is a test option.**

Applications

- General Purpose Switching Circuits
- Interfacing and coupling systems of different potentials and impedances
- I/O Interfacing
- Solid State Relays

MAXIMUM RATINGS ($T_A = 25^\circ\text{C}$ unless otherwise noted)

Rating	Symbol	Value	Unit
--------	--------	-------	------

INPUT LED

Reverse Voltage	V_R	3	Volts
Forward Current — Continuous	I_F	60	mA
LED Power Dissipation @ $T_A = 25^\circ\text{C}$ with Negligible Power in Output Detector Derate above 25°C	P_D	120	mW
		1.41	mW/ $^\circ\text{C}$

OUTPUT TRANSISTOR

Collector–Emitter Voltage	V_{CEO}	30	Volts
Emitter–Collector Voltage	V_{ECO}	7	Volts
Collector–Base Voltage	V_{CBO}	70	Volts
Collector Current — Continuous	I_C	150	mA
Detector Power Dissipation @ $T_A = 25^\circ\text{C}$ with Negligible Power in Input LED Derate above 25°C	P_D	150	mW
		1.76	mW/ $^\circ\text{C}$

TOTAL DEVICE

Isolation Surge Voltage ⁽¹⁾ (Peak ac Voltage, 60 Hz, 1 sec Duration)	V_{ISO}	7500	Vac(pk)
Total Device Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above 25°C	P_D	250	mW
		2.94	mW/ $^\circ\text{C}$
Ambient Operating Temperature Range ⁽²⁾	T_A	–55 to +100	$^\circ\text{C}$
Storage Temperature Range ⁽²⁾	T_{stg}	–55 to +150	$^\circ\text{C}$
Soldering Temperature (10 sec, 1/16" from case)	T_L	260	$^\circ\text{C}$

1. Isolation surge voltage is an internal device dielectric breakdown rating.
For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.
2. Refer to Quality and Reliability Section in Opto Data Book for information on test conditions.

Preferred devices are Motorola recommended choices for future use and best overall value.
GlobalOptoisolator is a trademark of Motorola, Inc.

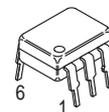
REV 5

© Motorola, Inc. 1995

4N25*
4N25A*
4N26*
[CTR = 20% Min]
4N27
4N28
[CTR = 10% Min]

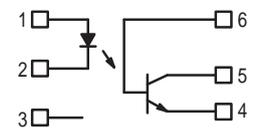
*Motorola Preferred Devices

STYLE 1 PLASTIC



STANDARD THRU HOLE
CASE 730A–04

SCHEMATIC



- PIN 1. LED ANODE
- LED CATHODE
- N.C.
- EMITTER
- COLLECTOR
- BASE

4N25 4N25A 4N26 4N27 4N28

ELECTRICAL CHARACTERISTICS (T_A = 25°C unless otherwise noted)⁽¹⁾

Characteristic	Symbol	Min	Typ ⁽¹⁾	Max	Unit
INPUT LED					
Forward Voltage (I _F = 10 mA)	V _F	—	1.15	1.5	Volts
T _A = 25°C		—	1.3	—	
T _A = -55°C		—	1.05	—	
Reverse Leakage Current (V _R = 3 V)	I _R	—	—	100	μA
Capacitance (V = 0 V, f = 1 MHz)	C _J	—	18	—	pF

OUTPUT TRANSISTOR

Collector–Emitter Dark Current (V _{CE} = 10 V, T _A = 25°C)	4N25,25A,26,27 4N28	I _{CEO}	—	1	50	nA
(V _{CE} = 10 V, T _A = 100°C)	All Devices	I _{CEO}	—	1	—	μA
Collector–Base Dark Current (V _{CB} = 10 V)		I _{CBO}	—	0.2	—	nA
Collector–Emitter Breakdown Voltage (I _C = 1 mA)		V _{(BR)CEO}	30	45	—	Volts
Collector–Base Breakdown Voltage (I _C = 100 μA)		V _{(BR)CBO}	70	100	—	Volts
Emitter–Collector Breakdown Voltage (I _E = 100 μA)		V _{(BR)ECO}	7	7.8	—	Volts
DC Current Gain (I _C = 2 mA, V _{CE} = 5 V)		h _{FE}	—	500	—	—
Collector–Emitter Capacitance (f = 1 MHz, V _{CE} = 0)		C _{CE}	—	7	—	pF
Collector–Base Capacitance (f = 1 MHz, V _{CB} = 0)		C _{CB}	—	19	—	pF
Emitter–Base Capacitance (f = 1 MHz, V _{EB} = 0)		C _{EB}	—	9	—	pF

COUPLED

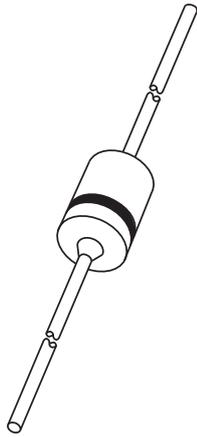
Output Collector Current (I _F = 10 mA, V _{CE} = 10 V)	I _C (CTR) ⁽²⁾	2 (20)	7 (70)	—	mA (%)
4N25,25A,26 4N27,28		1 (10)	5 (50)	—	
Collector–Emitter Saturation Voltage (I _C = 2 mA, I _F = 50 mA)	V _{CE(sat)}	—	0.15	0.5	Volts
Turn–On Time (I _F = 10 mA, V _{CC} = 10 V, R _L = 100 Ω) ⁽³⁾	t _{on}	—	2.8	—	μs
Turn–Off Time (I _F = 10 mA, V _{CC} = 10 V, R _L = 100 Ω) ⁽³⁾	t _{off}	—	4.5	—	μs
Rise Time (I _F = 10 mA, V _{CC} = 10 V, R _L = 100 Ω) ⁽³⁾	t _r	—	1.2	—	μs
Fall Time (I _F = 10 mA, V _{CC} = 10 V, R _L = 100 Ω) ⁽³⁾	t _f	—	1.3	—	μs
Isolation Voltage (f = 60 Hz, t = 1 sec) ⁽⁴⁾	V _{ISO}	7500	—	—	Vac(pk)
Isolation Resistance (V = 500 V) ⁽⁴⁾	R _{ISO}	10 ¹¹	—	—	Ω
Isolation Capacitance (V = 0 V, f = 1 MHz) ⁽⁴⁾	C _{ISO}	—	0.2	—	pF

1. Always design to the specified minimum/maximum electrical limits (where applicable).
2. Current Transfer Ratio (CTR) = I_C/I_F × 100%.
3. For test circuit setup and waveforms, refer to Figure 11.
4. For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.

ANEXO 9

1N4148 Datasheet

DATA SHEET



1N4148; 1N4448 High-speed diodes

Product specification
Supersedes data of 1999 May 25

2002 Jan 23

High-speed diodes

1N4148; 1N4448

FEATURES

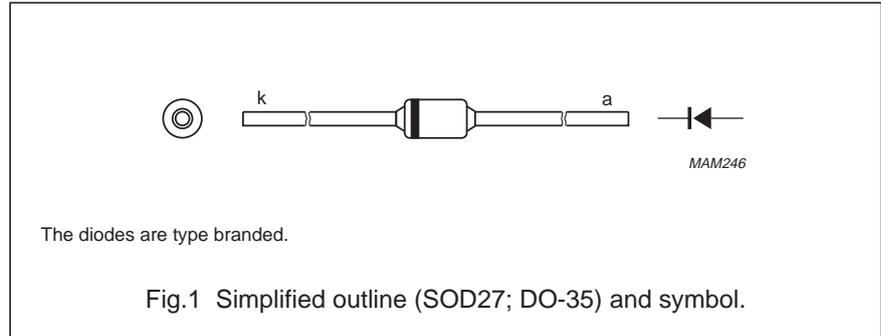
- Hermetically sealed leaded glass SOD27 (DO-35) package
- High switching speed: max. 4 ns
- General application
- Continuous reverse voltage: max. 75 V
- Repetitive peak reverse voltage: max. 100 V
- Repetitive peak forward current: max. 450 mA.

APPLICATIONS

- High-speed switching.

DESCRIPTION

The 1N4148 and 1N4448 are high-speed switching diodes fabricated in planar technology, and encapsulated in hermetically sealed leaded glass SOD27 (DO-35) packages.



LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 60134).

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V_{RRM}	repetitive peak reverse voltage		–	100	V
V_R	continuous reverse voltage		–	75	V
I_F	continuous forward current	see Fig.2; note 1	–	200	mA
I_{FRM}	repetitive peak forward current		–	450	mA
I_{FSM}	non-repetitive peak forward current	square wave; $T_j = 25\text{ °C}$ prior to surge; see Fig.4 $t = 1\ \mu\text{s}$ $t = 1\ \text{ms}$ $t = 1\ \text{s}$	–	4 1 0.5	A A A
P_{tot}	total power dissipation	$T_{amb} = 25\text{ °C}$; note 1	–	500	mW
T_{stg}	storage temperature		–65	+200	°C
T_j	junction temperature		–	200	°C

Note

1. Device mounted on an FR4 printed circuit-board; lead length 10 mm.

High-speed diodes

1N4148; 1N4448

ELECTRICAL CHARACTERISTICS

$T_j = 25\text{ °C}$ unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V_F	forward voltage 1N4148 1N4448	see Fig.3 $I_F = 10\text{ mA}$	–	1	V
		$I_F = 5\text{ mA}$	0.62	0.72	V
		$I_F = 100\text{ mA}$	–	1	V
I_R	reverse current	$V_R = 20\text{ V}$; see Fig.5		25	nA
		$V_R = 20\text{ V}$; $T_j = 150\text{ °C}$; see Fig.5	–	50	μA
I_R	reverse current; 1N4448	$V_R = 20\text{ V}$; $T_j = 100\text{ °C}$; see Fig.5	–	3	μA
C_d	diode capacitance	$f = 1\text{ MHz}$; $V_R = 0$; see Fig.6	–	4	pF
t_{rr}	reverse recovery time	when switched from $I_F = 10\text{ mA}$ to $I_R = 60\text{ mA}$; $R_L = 100\ \Omega$; measured at $I_R = 1\text{ mA}$; see Fig.7	–	4	ns
V_{fr}	forward recovery voltage	when switched from $I_F = 50\text{ mA}$; $t_r = 20\text{ ns}$; see Fig.8	–	2.5	V

THERMAL CHARACTERISTICS

SYMBOL	PARAMETER	CONDITIONS	VALUE	UNIT
$R_{th\ j-tp}$	thermal resistance from junction to tie-point	lead length 10 mm	240	K/W
$R_{th\ j-a}$	thermal resistance from junction to ambient	lead length 10 mm; note 1	350	K/W

Note

1. Device mounted on a printed circuit-board without metallization pad.

ANEXO 10

1N4007 Datasheet

Axial Lead Standard Recovery Rectifiers

This data sheet provides information on subminiature size, axial lead mounted rectifiers for general-purpose low-power applications.

Mechanical Characteristics

- Case: Epoxy, Molded
- Weight: 0.4 gram (approximately)
- Finish: All External Surfaces Corrosion Resistant and Terminal Leads are Readily Solderable
- Lead and Mounting Surface Temperature for Soldering Purposes: 220°C Max. for 10 Seconds, 1/16" from case
- Shipped in plastic bags, 1000 per bag.
- Available Tape and Reeled, 5000 per reel, by adding a "RL" suffix to the part number
- Polarity: Cathode Indicated by Polarity Band
- Marking: 1N4001, 1N4002, 1N4003, 1N4004, 1N4005, 1N4006, 1N4007

**1N4001
thru
1N4007**

1N4004 and 1N4007 are
Motorola Preferred Devices

**LEAD MOUNTED
RECTIFIERS
50-1000 VOLTS
DIFFUSED JUNCTION**



CASE 59-03
DO-41

MAXIMUM RATINGS

Rating	Symbol	1N4001	1N4002	1N4003	1N4004	1N4005	1N4006	1N4007	Unit
*Peak Repetitive Reverse Voltage Working Peak Reverse Voltage DC Blocking Voltage	V_{RRM} V_{RWM} V_R	50	100	200	400	600	800	1000	Volts
*Non-Repetitive Peak Reverse Voltage (halfwave, single phase, 60 Hz)	V_{RSM}	60	120	240	480	720	1000	1200	Volts
*RMS Reverse Voltage	$V_{R(RMS)}$	35	70	140	280	420	560	700	Volts
*Average Rectified Forward Current (single phase, resistive load, 60 Hz, see Figure 8, $T_A = 75^\circ\text{C}$)	I_O	1.0							Amp
*Non-Repetitive Peak Surge Current (surge applied at rated load conditions, see Figure 2)	I_{FSM}	30 (for 1 cycle)							Amp
Operating and Storage Junction Temperature Range	T_J T_{stg}	- 65 to +175							$^\circ\text{C}$

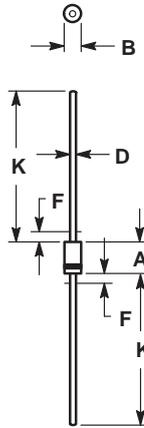
ELECTRICAL CHARACTERISTICS*

Rating	Symbol	Typ	Max	Unit
Maximum Instantaneous Forward Voltage Drop ($i_F = 1.0$ Amp, $T_J = 25^\circ\text{C}$) Figure 1	v_F	0.93	1.1	Volts
Maximum Full-Cycle Average Forward Voltage Drop ($I_O = 1.0$ Amp, $T_L = 75^\circ\text{C}$, 1 inch leads)	$V_{F(AV)}$	—	0.8	Volts
Maximum Reverse Current (rated dc voltage) ($T_J = 25^\circ\text{C}$) ($T_J = 100^\circ\text{C}$)	I_R	0.05 1.0	10 50	μA
Maximum Full-Cycle Average Reverse Current ($I_O = 1.0$ Amp, $T_L = 75^\circ\text{C}$, 1 inch leads)	$I_{R(AV)}$	—	30	μA

*Indicates JEDEC Registered Data

Preferred devices are Motorola recommended choices for future use and best overall value.

PACKAGE DIMENSIONS



- NOTES:
1. ALL RULES AND NOTES ASSOCIATED WITH JEDEC DO-41 OUTLINE SHALL APPLY.
 2. POLARITY DENOTED BY CATHODE BAND.
 3. LEAD DIAMETER NOT CONTROLLED WITHIN F DIMENSION.

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	4.07	5.20	0.160	0.205
B	2.04	2.71	0.080	0.107
D	0.71	0.86	0.028	0.034
F	—	1.27	—	0.050
K	27.94	—	1.100	—

CASE 59-03
(DO-41)
ISSUE M

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Motorola data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

Mfax is a trademark of Motorola, Inc.

How to reach us:

USA/EUROPE/Locations Not Listed: Motorola Literature Distribution;
P.O. Box 5405, Denver, Colorado 80217. 303-675-2140 or 1-800-441-2447

JAPAN: Nippon Motorola Ltd.: SPD, Strategic Planning Office, 4-32-1,
Nishi-Gotanda, Shinagawa-ku, Tokyo 141, Japan. 81-3-5487-8488

Mfax™: RMFAX0@email.sps.mot.com – TOUCHTONE 602-244-6609
– US & Canada ONLY 1-800-774-1848

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park,
51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852-26629298

INTERNET: <http://motorola.com/sps>



ANEXO 11

**TUTORIAL DE DESARROLLO DEL
SOFTWARE DEL
PROYECTO DE TESIS**

UNIVERSIDAD POLITÉCNICA SALESIANA
CARRERA DE INGENIERÍA ELECTRÓNICA



**TUTORIAL DE DESARROLLO DEL SOFTWARE DEL
PROYECTO DE TESIS “DISEÑO E
IMPLEMENTACIÓN DEL CONTROL PID DE
VELOCIDAD PARA UN MOTOR DC UTILIZANDO LA
TARJETA FPGA SPARTAN 3E Y LABVIEW PARA
LOS LABORATORIOS DE LA CARRERA DE
INGENIERÍA ELECTRÓNICA DE LA UNIVERSIDAD
POLITÉCNICA SALESIANA”.**

ELABORADO POR

Christopher Felipe León Galarza

Quito, Agosto 2011

INTRODUCCIÓN

En el presente tutorial se detallan los pasos seguidos para el desarrollo del software en LabVIEW del proyecto de tesis “DISEÑO E IMPLEMENTACIÓN DEL CONTROL PID DE VELOCIDAD PARA UN MOTOR DC UTILIZANDO LA TARJETA FPGA SPARTAN 3E Y LABVIEW PARA LOS LABORATORIOS DE LA CARRERA DE INGENIERÍA ELECTRÓNICA DE LA UNIVERSIDAD POLITÉCNICA SALESIANA” con lo que los lectores del presente escrito podrán desarrollarlo y modificarlo en futuros proyectos.

Este tutorial está dividido en tres partes, la primera trata sobre la creación de el proyecto LabVIEW-FPGA, la segunda trata sobre el desarrollo del VI de configuración del FPGA y la tercera trata sobre el VI HOST el cual es el que realizará el control PID sobre el motor y en donde se encuentra la interfaz de usuario.

Este tutorial fue realizado tomando en cuenta que el lector tiene conocimientos básicos de programación en LabVIEW.

CREACIÓN DEL PROYECTO

1. Abrir el software LabVIEW.
2. En la ventana de inicio seleccionar **Empty Project** para crear un nuevo proyecto.



FIGURA 1. PANTALLA DE INICIO DE LABVIEW
(Captura del software LabVIEW)

3. Se abrirá la ventana **Project Explorer** en donde se hace clic derecho en **My Computer** y se va a **New>>Targets and Devices**, para añadir un nuevo dispositivo de hardware embebido.
4. En la ventana que se abre se selecciona **New target or device** y se busca la FPGA “*Spartan 3E Starter Board*” y se pulsa ok.

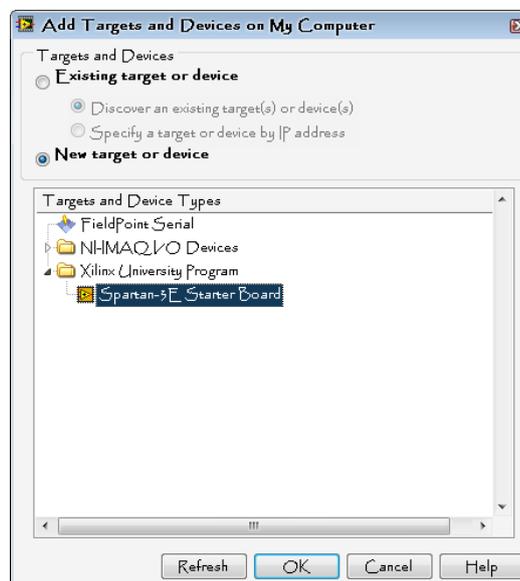


FIGURA 2. SELECCIÓN DE HARDWARE EMBEBIDO
(Captura del software LabVIEW)

5. Hecho eso aparecerá la tarjeta FPGA SPARTAN 3E en la ventana Project Explorer.

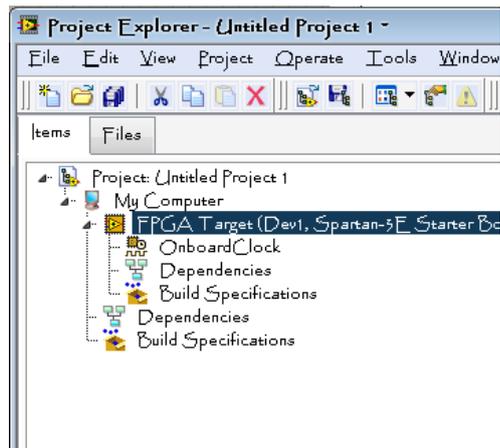


FIGURA 3. PROJECT EXPLORER
(Captura del software LabVIEW)

6. En el Project Explorer se hace clic derecho sobre la *FPGA SPARTAN 3E* y se va a **New>>FPGA I/O** para seleccionar los elementos del FPGA que se utilizarán en el proyecto.
7. En la ventana que aparece se aprecia una lista con todos los elementos disponibles en la tarjeta FPGA SPARTAN 3E, aquí se seleccionan dos puertos digitales (*Expansion Connectors*), que en este caso serán el **FX2_IO9_J4_0** y el **FX2_IO10_J4_0**, para añadirlos se los selecciona y se hace clic en el icono de **Add Item**.

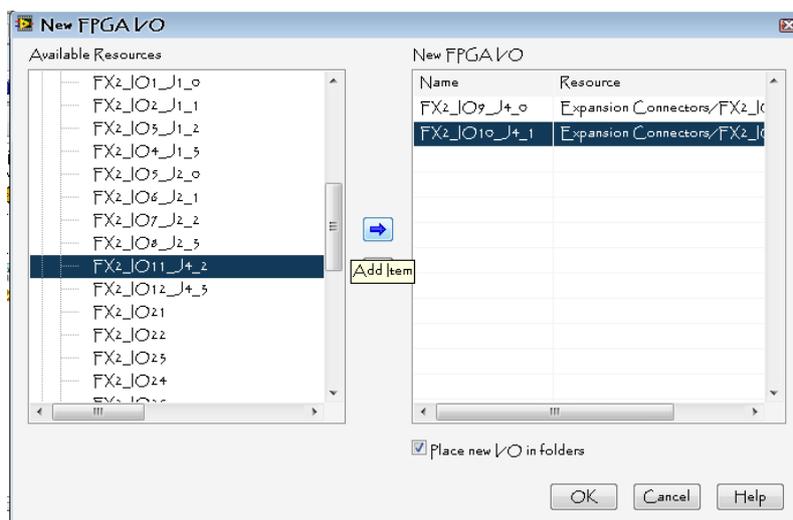


FIGURA 4. SELECCIÓN DE ELEMENTOS
(Captura del software LabVIEW)

8. Una vez agregados estos elementos aparecerán en el Project Explorer.

VI DE CONFIGURACIÓN DEL FPGA

1. En el Project Explorer se hace clic derecho sobre la FPGA SPARTAN 3E y se va a **New>>VI** con lo que creamos el VI de configuración del FPGA.
2. En el diagrama de bloques de este VI se colocan dos estructuras **While Loop**, una para la generación de pulsos y otra para la lectura de pulsos.
3. Dentro de cada una de estas estructuras se arrastran cada uno de los conectores digitales desde el **Project Explorer**.

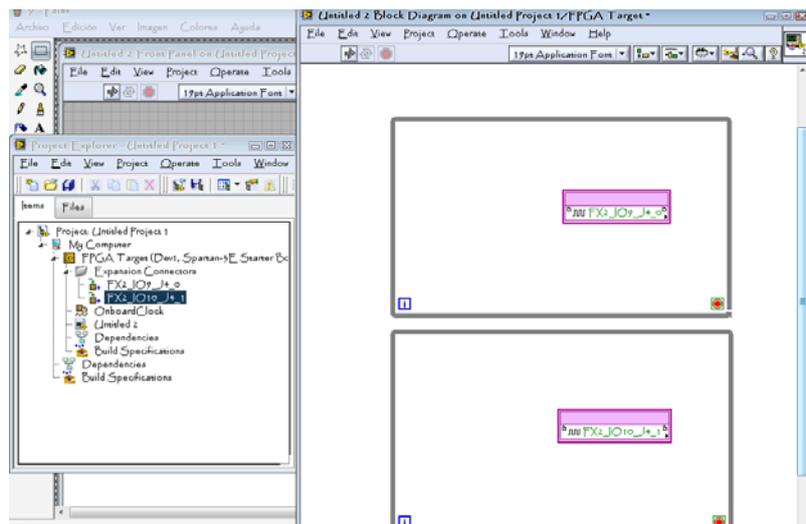


FIGURA 5. ARRASTRE DE ELEMENTOS HACIA ESTRUCTURAS
(Captura del software LabVIEW)

4. En el conector **FX2_IO9_J4_0** se hace clic derecho y se selecciona **Change To Write** para hacer que este conector sea una salida digital, ya que por defecto los conectores están como entradas digitales.
5. En la estructura que contiene a este conector configurado como salida se inserta un generador de pulsos el cual está en la paleta de **FPGA Math & Análisis>>Generation**.
6. En el momento que se inserta el generador se abrirá una ventana de configuración donde se configuraran los parámetros como frecuencia, duty cycle, tipo de señal, etc. Los cuales quedarán configurados como muestra la figura 6.

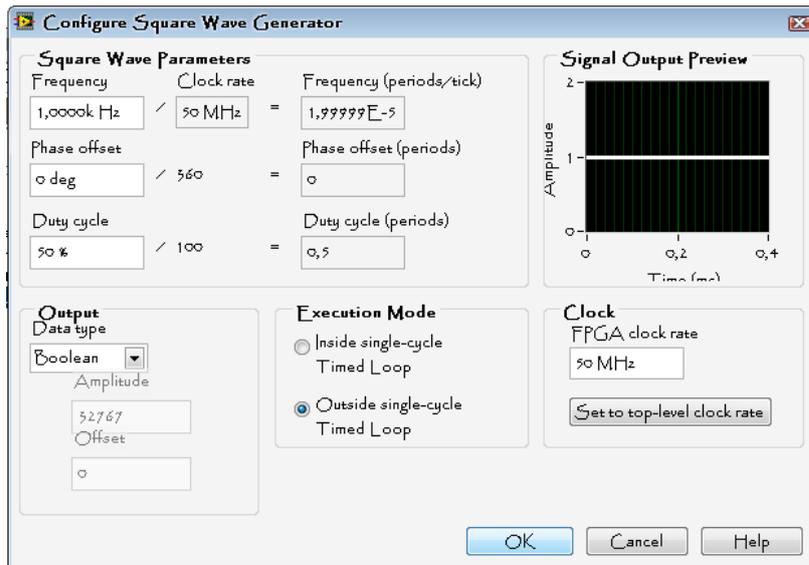


FIGURA 6. VENTANA DE CONFIGURACIÓN DEL GENERADOR DE PULSOS
(Captura del software LabVIEW)

7. Una vez configurado se pulsa **OK** para salir de la ventana, hecho esto se une la salida del generador a la entrada del conector **FX2_IO9_J4_0**.
8. Se crean dos controles, uno para la frecuencia y otro para el duty cycle del generador, en las entradas del generador que corresponden a estas variables.

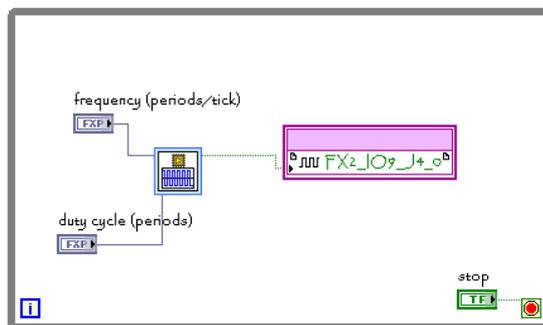


FIGURA 7. CONEXIÓN DEL GENERADOR
(Captura del software LabVIEW)

9. En la estructura que contiene al conector **FX2_IO10_J4_0** se inserta un detector de cruce booleano el cual está en la paleta **FPGA Math & Análisis>>Utilities** y se conecta su entrada a la salida del conector.
10. Se crea una constante en el pin **Direction** del detector de cruce booleano y se la configura como “*false-true*” para que detecte los flancos positivos.
11. En la salida del detector se coloca un convertor **Boolean to 0,1**.

12. Dentro de esta misma estructura se inserta un acumulador desde la paleta **FPGA Math & Análisis>>High Throughput Math>>Basic Elements**.
13. La entrada del acumulador se conecta a la salida del conversor booleano a numérico (*Boolean to 0,1*), y la salida de este se conecta a un conversor a entero de 64 bits que está en la paleta **Numeric>>Conversion**.
14. Luego se conecta la salida de este conversor a un indicador numérico que se llamará “sumatoria”.

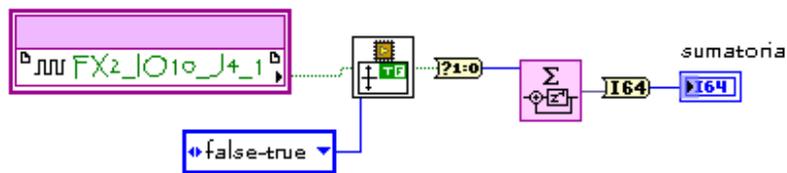


FIGURA 8. DIAGRAMA DE LA LECTURA DE PULSOS
(Captura del software LabVIEW)

15. Para compilar el código gráfico y transferirlo al FPGA se guarda el VI y se hace clic en el icono de **RUN**, momento en el cual se iniciará el compilador del módulo LabVIEW-FPGA

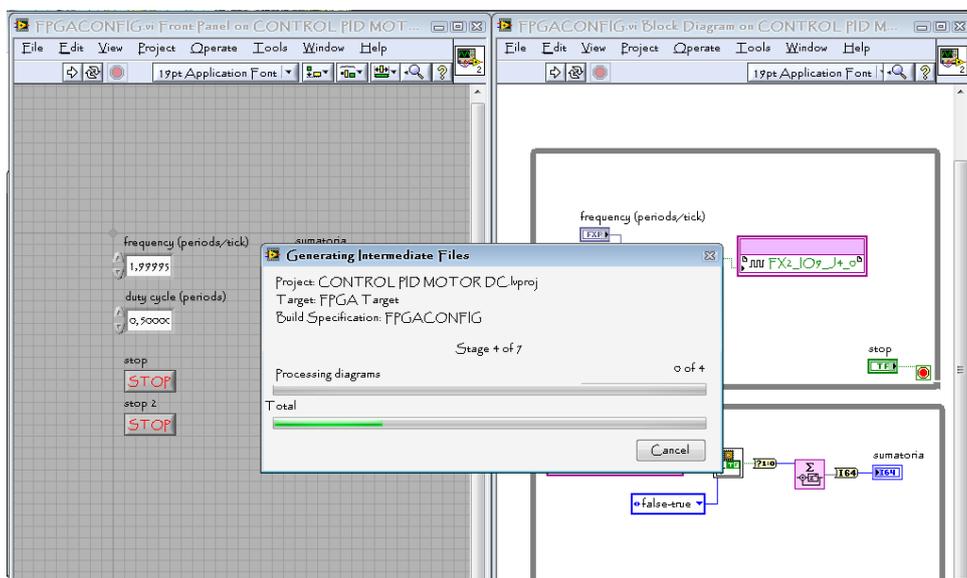


FIGURA 9. EJECUCIÓN DEL COMPILADOR
(Captura del software LabVIEW)

16. Cuando el compilador termine su procesamiento transferirá los datos al FPGA y la ventana de estado indicará que el proceso está completo y el

programa correrá automáticamente, con lo que se podrá probar si la programación de configuración es correcta.

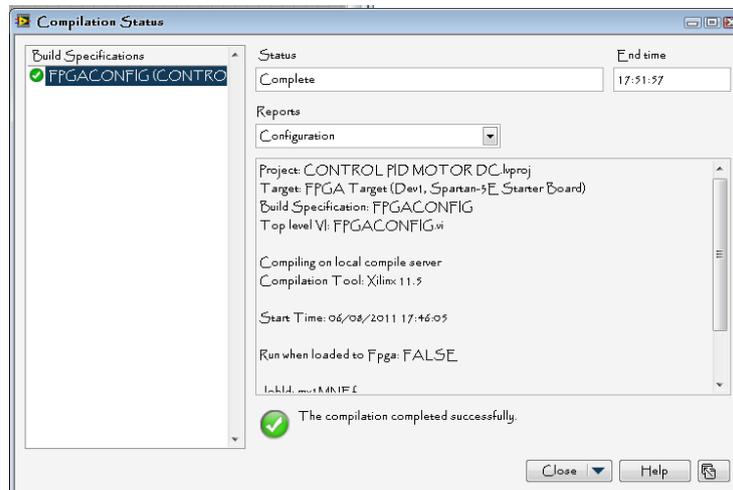


FIGURA 10. ESTADO DE LA COMPILACIÓN
(Captura del software LabVIEW)

VI HOST

1. En el Project Explorer se hace clic derecho en **My Computer** y se va a **New>>VI** para crear el VI HOST.
2. En el diagrama de bloques del nuevo VI se inserta una estructura **Timed Loop**.
3. Desde la paleta **FPGA Interface** se insertan un bloque **Open FPGA** y un **Close FPGA** y se colocan a ambos lados exteriores de la estructura.
4. De esta misma paleta se inserta un control **Read/Write** y se lo coloca dentro de la estructura.
5. El pin **Reference Out** del control **Open FPGA** se conecta al pin **Reference In** del control **Read/Write** y el pin **Reference Out** de este se conecta al pin **Reference In** del control **Close FPGA**.
6. El paso anterior se repite para los pines de error.

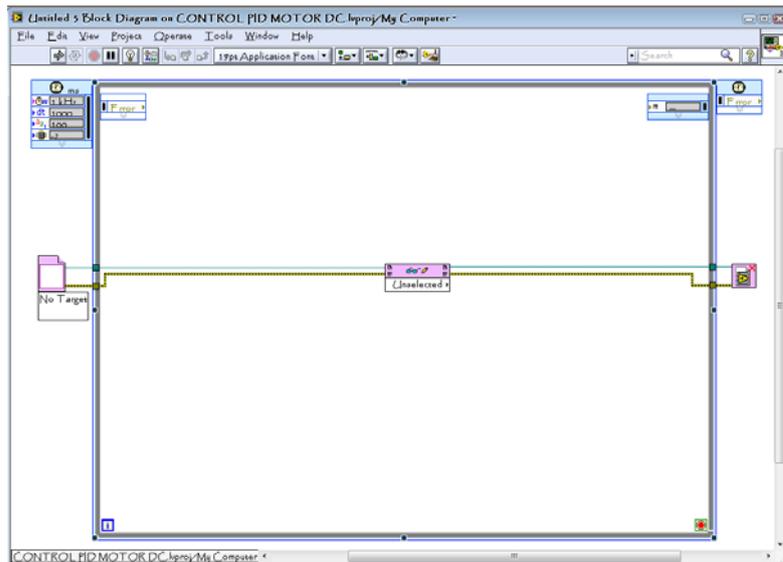


FIGURA 11. CONEXIÓN DEL CONTROL READ/WRITE
(Captura del software LabVIEW)

7. Se hace doble clic sobre el control **Open FPGA** y en la ventana que se abre se selecciona **VI**, en la siguiente ventana que se abre se selecciona el vi de configuración que se realizó anteriormente y se pulsa ok en ambas ventanas.

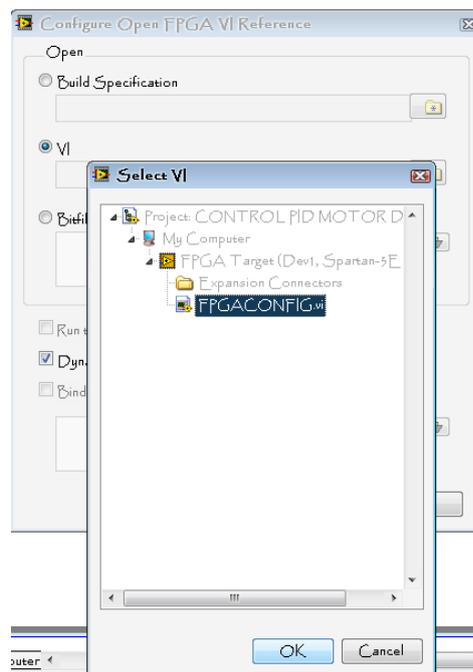


FIGURA 12. SELECCIÓN DEL VI DE CONFIGURACIÓN.
(Captura del software LabVIEW)

8. Se hace clic sobre el control **Read/Write** y aparecerá un menú de los controles e indicadores creados en el vi de configuración, en este caso se seleccionará **Frecuency**.
9. Se hace clic derecho sobre este mismo control y se selecciona la opción **Add Element**, esto se hará dos veces para añadir dos elementos en uno se seleccionara **Duty Cycle** y en el otro **sumatoria**.

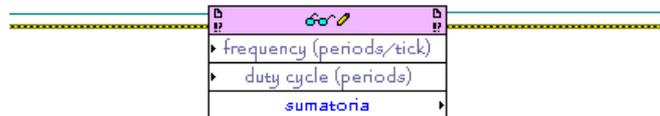


FIGURA 13. ELEMENTOS DEL CONTROL READ/WRITE.
(Captura del software LabVIEW)

10. Se hace clic en el control superior derecho de la estructura y se selecciona la opción **Period**, luego se crea una constante con el valor de 100 y se conecta a este control.

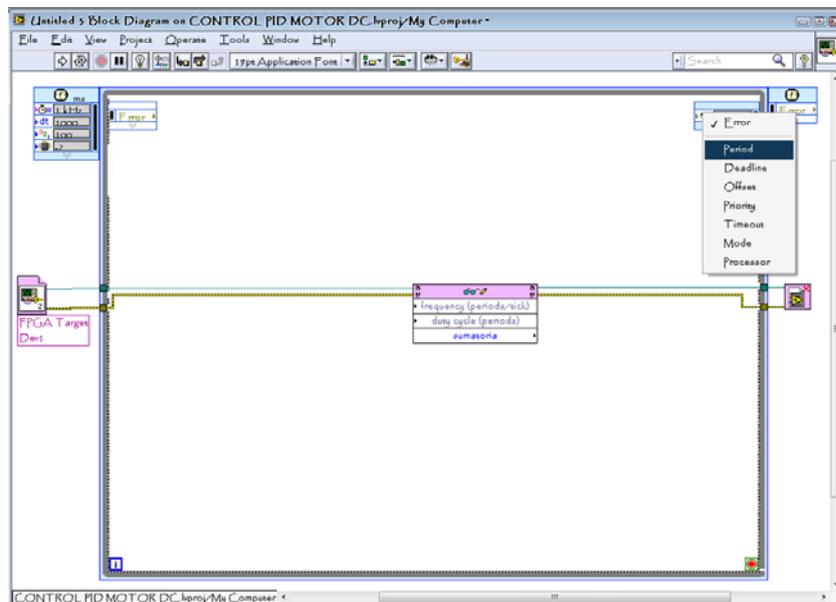


FIGURA 14. SELECCIÓN DEL CONTROL DE LA ESTRUCTURA TIMED LOOP.
(Captura del software LabVIEW)

11. En el panel frontal se crean: un **Numeric Control** para la frecuencia en Hz, un **Pointer Slide** llamado **set point**, un **Fill Slide** llamado **velocidad** y se inserta un **Waveform Chart** para la gráfica de velocidad vs set point.

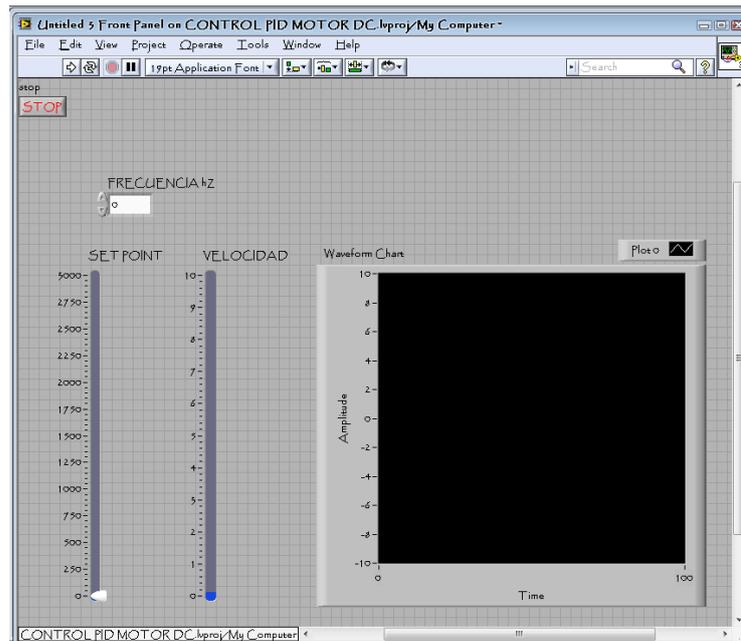


FIGURA 15. CONTROLES EN EL PANEL FRONTAL
(Captura del software LabVIEW)

12. En el diagrama de bloques, en la salida del control de frecuencia que se inserto se coloca un divisor con una constante de 50000000, ya que se debe dividir la frecuencia en Hz para los 50MHz a los que trabaja el reloj del FPGA, a la salida del divisor se coloca un multiplicador con una constante de 100000, a la salida de este se coloca un conversor a entero de 16 bits, esto se hace debido a que se deben eliminar los decimales para la conversión a fixed point, que es el formato en que trabajan los controles del generador de pulsos, a la salida del conversor se coloca un conversor **Integer to Fixed Point Cast** el cual está en la paleta **Numeric>>Fixed Point**.
13. A la salida de este conversor se coloca un divisor con una constante de 10000 y salida de este se coloca un divisor con una constante de 10, esto es para recuperar el número en su formato decimal pero no se puede realizar en una sola división ya que el formato no lo permite.
14. La salida de este divisor se conecta al elemento **Frecuency** del control **Read/Write**

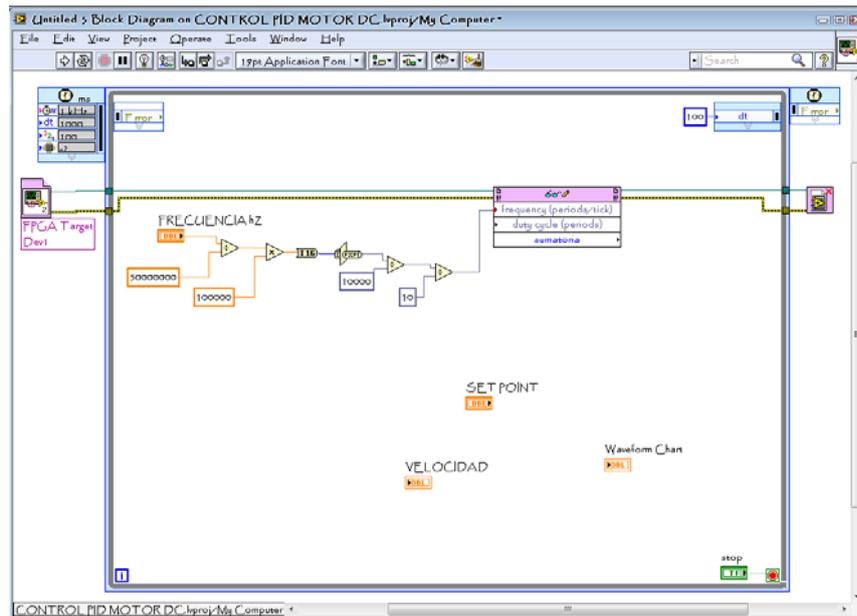


FIGURA 16. ESCALAMIENTO DE LA FRECUENCIA DEL GENERADOR DE PULSOS.
(Captura del software LabVIEW)

15. En la estructura **Timed Loop** se añade un **Shift Register**, la salida de este se conecta al pin y de un restador y la entrada de este se conecta a la salida del **Shift Register** y al elemento **sumatoria** del control **Read/Write**.
16. La salida del restador se conecta a un divisor para 100 que es el número de ranuras del disco acoplado al motor, la salida de este divisor se conecta a un multiplicador por 600 para que la velocidad quede en rpm ya que el periodo es de 100ms.
17. Se coloca dentro de la estructura un control **PID Autotuning** que está en la paleta **Control Design & Simulation>>PID**.
18. Se conecta el control **set point** en el pin del mismo nombre que tiene el bloque **PID Autotuning**.
19. La salida del multiplicador por 600 se conecta al pin de velocidad del bloque **PID Autotuning**.
20. El pin **Output** del bloque **PID Autotuning** se conecta a un divisor para 100 ya que le control se efectúa de 0 a 100% y el generador trabaja de 0 a 1, luego se le restan 0.005 debido a que en las pruebas preliminares se comprobó que el FPGA se cuelga si se llega al 100% exacto, la salida de este restador se conecta al elemento **Duty Cycle** del control **Read/Write**

21. En los pines **Output Range** y **Autotuning Parameters** del bloque **PID Autotuning** se crean constantes las cuales se configuraran según muestra la figura 17.
22. En el pin **PID Gains** del bloque **PID Autotuning** se crea un control, de este control se crea una variable local la cual se coloca una estructura de condición dentro de la opción **True**, esta variable se configura como indicador y se conecta al pin **PID Gains Out**, la condición de la estructura se conecta al pin **Tuning Completed?**.
23. Se coloca un bloque **Bundle** el cual está en la paleta **programming>> cluster, class & variant**, en las entradas de este se conectan el control **set point** y la salida del multiplicador por 600, la salida del bloque **Bundle** se conecta al **Waveform Chart**, con lo que se podrá visualizar la gráfica de set point vs velocidad.

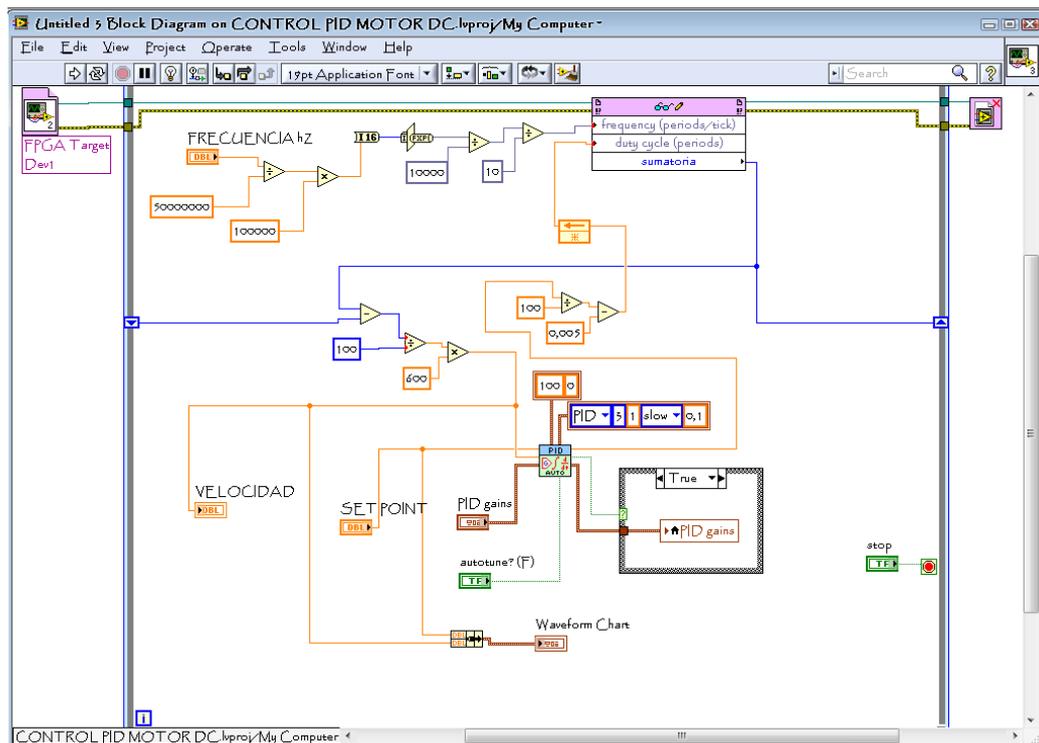


FIGURA 16. DIAGRAMA DE BLOQUES COMPLETO.
(Captura del software LabVIEW)

24. Finalmente se ordenan los elementos del panel frontal y el software del CONTROL PID DEL MOTOR DC en LabVIEW se encuentra listo para correrse.

ANEXO 12

**CÓDIGO Y CAPTURAS DE LOS
CÁLCULOS EN MATLAB**

UNIVERSIDAD POLITÉCNICA SALESIANA
CARRERA DE INGENIERÍA ELECTRÓNICA



**CÁLCULOS EN MATLAB DE LAS FUNCIONES DE
TRANSFERENCIA DEL PROYECTO DE INVESTIGACION
“DISEÑO E IMPLEMENTACIÓN DEL CONTROL PID DE
VELOCIDAD PARA UN MOTOR DC UTILIZANDO LA TARJETA
FPGA SPARTAN 3E Y LABVIEW PARA LOS LABORATORIOS
DE LA CARRERA DE INGENIERÍA ELECTRÓNICA DE LA
UNIVERSIDAD POLITÉCNICA SALESIANA.”**

ELABORADO POR

Christopher Felipe León Galarza

Quito, Agosto 2011

CÓDIGO DEL PROGRAMA EN MATLAB

```
z=sym('z');

numpidz=(0.05804*z^2)-(0.1451*z^1)+0.1451;
denpidz=1-z^1;
pidz=numpidz/denpidz;
pidz=vpa(pidz,6)
pretty(pidz)

numgz=(0.067522*z^1)-(0.067246*z^2);
dengz=1-(1.987775*z^1)+(0.098777*z^2);
gz=numgz/dengz;
gz=vpa(gz,6)
pretty(gz)

numgpidez=expand(numpidz*numgz);
dengpidez=expand(denpidz*dengz);
gpidez=numgpidez/dengpidez;
gpidez=vpa(gpidez,6)
pretty(gpidez)

numgsz=gpidez;
dengsz=1+gpidez;
gsz=(numgsz/dengsz)
pretty(gsz)
gsz=expand(gsz);
gsz=vpa(gsz,6);
gsz=simplify(gsz)
pretty(gsz)
```

RESULTADOS OBTENIDOS

```

MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB65\work

Works
Name
dengr
denge
dengz
denpi
gpidz
gsz
gz
mumgr
Work

Command Window
>>
pidz =
(.580400e-1*z^2-.145100*z+.145100)/(1.-1.*z)

      2
      .0580400 z - .145100 z + .145100
      -----
      1. - 1. z

gz =
(.675220e-1*z-.672460e-1*z^2)/(1.-1.98778*z+.987770e-1*z^2)

      2
      .0675220 z - .0672460 z
      -----
      1. - 1.98778 z + .0987770 z

gpidz =
(.136764e-1*z^3-.390296e-2*z^4-.195548e-1*z^2+.979744e-2*z)/(1.-2.98778*z+2.08655*z^2-.987770e-1*z^3)

      3      4      2
      .0136764 z - .00390296 z - .0195548 z + .00979744 z
      -----
      1. - 2.98778 z + 2.08655 z - .0987770 z

gsz =
(.136764e-1*z^3-.390296e-2*z^4-.195548e-1*z^2+.979744e-2*z)/(1.-2.98778*z+2.08655*z^2-.987770e-1*z^3)+
(.136764e-1*z^3-.390296e-2*z^4-.195548e-1*z^2+.979744e-2*z)/(1.-2.98778*z+2.08655*z^2-.987770e-1*z^3)

      3      4      2      3
      (.0136764 z - .00390296 z - .0195548 z + .00979744 z) / (1. - 2.98778 z + 2.08655 z - .0987770 z)

      3      4      2      3
      | .0136764 z - .00390296 z - .0195548 z + .00979744 z |
      | | | | |
      | | | | |
      \ 1. - 2.98778 z + 2.08655 z - .0987770 z //

gsz =
2.*(-170955.*z^2+48787.*z^3+244435.*z-122468.)/(-25000000.+74449564.*z-51674880.*z^2+2127515.*z^3+97574.*z^4)

      2      3      8
      2. (-170955. z + 48787. z + 244435. z - 122468.) z / (-25000000 10
      + .74449564 10 z - .51674880 10 z + .2127515 10 z + 97574. z )

>>

```