

**UNIVERSIDAD POLITECNICA SALESIANA  
SEDE QUITO**

**CARRERA: INGENIERÍA ELECTRÓNICA**

**Tesis previa a la obtención del título de: INGENIERO ELECTRÓNICO**

**TEMA:**

**“DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE FILTROS  
DIGITALES PARA ELIMINAR RUIDO DE LA RED ELÉCTRICA EN  
CIRCUITOS ELECTRÓNICOS, UTILIZANDO LA FPGA SPARTAN 3E  
Y LABVIEW PARA EL LABORATORIO DE PROCESAMIENTO  
DIGITAL DE SEÑALES”**

**AUTOR:**

**JAVIER ALEJANDRO CASTILLO CALDERÓN**

**DIRECTOR:**

**LUIS OÑATE CADENA**

**Quito, septiembre 2013**

## **DECLARATORIA DE RESPONSABILIDAD Y AUTORIZACIÓN DE USO DEL TRABAJO DE GRADO**

Yo, Javier Alejandro Castillo Calderón autorizo a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de grado y su reproducción sin fines de lucro.

Además declaro que los conceptos y análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad del autor de este proyecto de grado.

---

Javier Alejandro Castillo Calderón

CC:1714975974

## **DEDICATORIA**

Ante todo a mis hermanas Daysi Castillo y Solange granja, que estuvieron años hermosos de mi vida a lado mío y supieron apoyarme en los momentos más duros del camino hacia mi profesión, a mis padres José Castillo y Martha Calderón por demostrarme que por más dura que sea la situación en la vida siempre existe una luz al final, a mis primos Rommel y Jairo, los cuales considero como hermanos y supieron ser buenos consejeros en mi carrera, y ante todo a Dios por permitir que pueda estar aquí realizando esta dedicatoria, y sintiendo su infinito amor por mí.

## **AGRADECIMIENTO**

Al Ingeniero Germán Arévalo, por su incansable esfuerzo con la carrera de Ingeniería Electrónica y en lo personal en mi formación como Ingeniero, el cual aportó valiosos conocimientos, al Ing. Luis Oñate, por la confianza depositada en mí para poder realizar este proyecto de grado y por el apoyo recibido de su parte en la ejecución del proyecto, al Dr. Daniel Cárdenas por los consejos recibidos durante el proceso de elaboración de mi proyecto de grado y a todos los docentes que mediante sus conocimientos aclararon cada interrogante planteada.

## ÍNDICE

INTRODUCCIÓN .....	1
CAPÍTULO 1 .....	2
PLANTEAMIENTO DEL PROBLEMA .....	2
1.1 Introducción .....	2
1.2 Problema a resolver.....	3
1.3 Hipótesis.....	4
1.4 Tesis .....	5
1.5 Objetivos .....	6
1.5.1 Objetivo general .....	6
1.5.2 Objetivos específicos: .....	6
1.6 Justificación.....	6
1.7 Alcance.....	7
1.8 Metodología de la investigación .....	8
1.8.1 Método lógico deductivo.....	8
1.8.2 Desarrollo del circuito generador de la señal .....	9
1.8.3 Desarrollo del software .....	9
1.8.4 Construcción final del proyecto .....	10
CAPÍTULO 2 .....	11
ESTADO DEL ARTE.....	11
2.1 Introducción .....	11
2.2 Estado del arte .....	11
2.3 Marco teórico .....	11
2.3.1 Sistemas lineales, invariancia en el tiempo y causalidad .....	11
2.3.1.1 Linealidad.....	11
2.3.1.2 Invariancia en el tiempo .....	13
2.3.1.3 Causalidad .....	13
2.3.2 Teoría de filtros .....	13
2.3.2.1 Definición.....	13
2.3.2.2 Respuesta impulsional y frecuencial de un filtro .....	14
2.3.2.3 Tipos de filtros .....	14

2.3.3 Filtros digitales.....	17
2.3.3.1 Introducción .....	17
2.3.4 Introducción a filtros digitales en Labview.....	18
2.3.4.1 Módulo labview-fpga.....	18
2.3.4.2Módulo de funciones para el procesamiento de señales .....	22
2.3.4.3Filtro notch.....	23
2.3.4.4Filtros pasivos de primer orden pasa bajos .....	24
2.3.5 Tarjeta fpga spartan 3e.....	25
2.3.5.1Características de la tarjeta fpga spartan 3e .....	25
2.3.5.2Amplificador programable.....	27
2.3.5.3Módulo conversor análogo digital (adc) .....	31
2.3.5.4Módulo conversor digital análogo (DAC) .....	35
2.3.5.5Detalles de la comunicación.....	38
2.3.5.5 Módulo de comunicación SPI.....	39
2.3.5.6Módulo oscilador de cristal.....	41
2.3.6 Sensores y acondicionadores.....	41
2.3.6.1Termocupla tipo j .....	41
2.3.6.2Amplificador AD620 .....	42
2.3.6.3Amplificador operacional lm358 .....	43
2.3.6.4Amplificador operacional lm741 .....	44
CAPÍTULO 3 .....	46
DESARROLLO DE HADWARE Y SOFTWARE .....	46
3.1Introducción .....	46
3.2 Diseño de hardware.....	46
3.2.1Diagrama de bloques.....	47
3.2.2Descripción de los bloques de hardware .....	48
3.2.2.1Fuente de alimentación .....	48
3.2.2.2Adquisición de la señal .....	49
3.2.2.3Acondicionamiento de la señal del sensor de temperatura .....	52
3.2.2.4Diseño de filtro r-c .....	54
3.3 Desarrollo del software e implementación del hardware del proyecto .....	56

3.3.1 Diagrama de flujo general.....	56
3.3.1.1Desarrollo del software .....	58
3.3.2Implementación del software del proyecto .....	60
3.3.2.1Diagrama de flujo de la configuración de los conversores de la tarjeta fpga y la programación en Labview .....	61
3.3.2.2Configuración del pre amplificador de la tarjeta fpga .....	62
3.3.2.3Activación del ADC.....	64
3.3.2.4Activación del DAC.....	67
3.3.2.5Programación del PIC 16F877A .....	68
3.4 Desarrollo del hardware .....	70
3.4.1 Diseño de la placa .....	70
3.4.2 Diseño de estructura.....	72
3.5 Costos del proyecto de investigación.....	73
3.5.1 Costos de desarrollo matemático .....	73
3.5.2 Costos del desarrollo del software .....	74
3.5.3Costos de desarrollo del hardware .....	74
3.5.4 Costo total del proyecto .....	76
CAPÍTULO 4.....	77
ANÁLISIS DE RESULTADOS .....	77
4.1 Introducción .....	77
4.2Diagramas de prueba.....	81
Conclusiones .....	85
Recomendaciones.....	87
LISTA DE REFERENCIAS .....	88
Netgrafía.....	89
Manuales .....	90
ANEXOS .....	91

## ÍNDICE DE FIGURAS

Figura 1 Diagrama de linealidad	12
Figura 2 Grafica del filtro pasa bajo	14
Figura 3 Grafica del filtro pasa alto	15
Figura 4 Grafica del filtro pasa alto	15
Figura 5 Grafica del filtro pasa alto	16
Figura 6 Diagrama en bloques de los dos tipos de filtros	17
Figura 7 Modulo de análisis matemático para fpga	18
Figura 8 Estructura de programa en fpga vs estructura de programación en Labview	19
Figura 9 Funciones del paletfpga-interface de Labview	21
Figura 10 Funciones de procesamiento de señales	22
Figura 11 Curva característica del filtro notch	23
Figura 12 Circuitos usados como filtros de primer orden	24
Figura 13 Tarjeta FPGA Spartan 3E.	25
Figura 14 LTC6912-1 amplificador	27
Figura 15 Interfaz serial del amplificador	30
Figura 16. Tiempos de comunicación del amplificador	30
Figura 17 Interfaz del conversor análogo-digital	33
Figura 18 ltc2624-DAC	35
Figura 19 Diagrama de bloques de conexiones del dispositivo LTC2624	36
Figura 20 formas de onda de la comunicación SPI del DAC	39
Figura 21 Protocolo de comunicación del LTC2624	39
Figura 22 Comunicación SPI	40
Figura 23 Fotografía de la termocupla	41



Figura 24 Diagrama de terminales del amplificador de instrumentación AD620	42
Figura 25 Diagrama del amplificador operacional LM358	43
Figura 26 Diagrama del amplificador operacional LM741	44
Figura 27 Diagrama de bloques del hardware del circuito electrónico	47
Figura 28 Diseño del circuito divisor de voltaje	49
Figura 29 Diseño del circuito sumador no inversor	50
Figura 30 Diseño del circuito sumador no inversor	51
Figura 31 Diseño del circuito del amplificador AD620	53
Figura 32 Diseño del circuito de la adquisición de la señal	54
Figura 33 Diseño del circuito del filtro r-c	55
Figura 34 Diagrama del esquema completo del hardware	55
Figura 35 Diagrama de flujo del software	56
Figura 36 Diagrama de flujo del software	58
Figura 37 Diagrama de flujo de la configuración de los conversores ADC y DAC de la tarjeta fpga	61
Figura 38 Diagrama de flujo de la configuración del pre amplificador	63
Figura 39 Diagrama de flujo la activación del ADC	65
Figura 40 Diagrama de flujo la activación del DAC	67
Figura 41 Diagrama de flujo la programación PIC 16f877A	70
Figura 42 Placa para adquirir la señal del sensor	71
Figura 43 Placa para mostrar temperatura	71
Figura 44 Vistas de la caja donde se montó el proyecto final	72
Figura 45 Base para sensor y llama	73
Figura 46 Señal sensor sumada señal de ruido	81

Figura 47 Temperatura sensor vs temperatura de salida	81
Figura 49 Señal del sensor canal1 señal de salida canal 2	83
Figura 50 Temperatura sensor vs temperatura de salida	83
Figura 51 Señal del sensor canal1 señal de salida canal 2	84
Figura 52 Temperatura sensor vs temperatura de salida	84
Figura 53 Señal del sensor canal1 señal de salida canal 2	84
Figura 54 Temperatura sensor vs temperatura de salida	85

## ÍNDICE DE TABLAS

Tabla 1 Ganancias para el amplificador LTC6912-1	28
Tabla 2Deshabilitar otros dispositivos en el bus SPI	29
Tabla 3Señales de interfaz del pre amplificador	29
Tabla 4Señales de interfaz ADC	31
Tabla 5 Paquete de datos que ofrece el ADC	32
Tabla 6Deshabilitación de otros dispositivos del bus SPI	34
Tabla 7 Paquete de datos que ofrece el DAC	37
Tabla 8Salidas analógicas del DAC	38
Tabla 9Señales de interfaz del DAC	40
Tabla 10Des habilitación de otros dispositivos en el bus SPI	41
Tabla 11Costos del desarrollo matemático	73
Tabla 12Costos de desarrollo de software	74
Tabla 13Costos de desarrollo de hardware	75
Tabla 14 Costo total del proyecto	76
Tabla 15Temperatura del sensor vs temperatura a la salida	77
Tabla 16Porcentaje del valor de error en %	78

## **ÍNDICE DE ANEXOS**

Anexo1 Características de la FPGA	93
Anexo 2 DatasheetLTC6912-1	94
Anexo 3 Datasheet oscilador de cristal de la tarjeta	95
Anexo 4 Curva de calibración de la termocupla tipo J y características	96
Anexo 5 Datasheet AD620	100
Anexo 6 Datasheet termocupla tipo J	101
Anexo7 Datasheet LM358	102
Anexo 8 Datasheet LM741	103
Anexo 9 Desarrollo e implementación del software en Labview	104
Anexo10 Programación PIC 16F8774	114
Anexo11 Datasheet LTC1407A-1 A/D	119

## **RESUMEN**

El proyecto de tesis “ Diseño, desarrollo e implementación de filtros digitales para eliminar ruido de la red eléctrica en circuitos electrónicos, utilizando la FPGA Spartan 3E y Labview para el laboratorio de procesamiento digital de señales” de la carrera de Electrónica de la Universidad Politécnica Salesiana es un proyecto de investigación cuyo objetivo principal es utilizar las prestaciones del software gráfico Labview al permitir una implementación en la tarjeta de desarrollo FPGA Spartan 3E mediante sus controladores que permiten el diseño e implementación de filtros digitales, con lo cual se logra implementar prácticas que pueden estar a disposición de los estudiantes y docentes en los laboratorios de la Universidad Politécnica Salesiana , mediante el presente proyecto los estudiantes podrán reforzar los conocimientos teóricos que se reciben en el curso de Procesamiento de Señales Digitales (DSP).

## **ABSTRACT**

The thesis project "Design, development and implementation of digital filters to remove noise from the power supply in electronic circuits, using Labview FPGA Spartan3E and Laboratory Digital Signal Processing" of the race Electronics Salesian Polytechnic University is a research project whose main objective is to use the Lab VIEW graphical software performance by allowing an implementation in FPGA development board Spartan 3E whit their drivers that allow the design and implementation of digital filters, which is achieved by implementing practices that can be available to students and teachers in the laboratories of the Salesian University, poject to here by students can rein force the theoretical know ledge received in the course of Digital Signal Processing(DSP).

## **INTRODUCCIÓN**

El diseño y desarrollo del proyecto se basa en la aplicación específica de un software el cual permitirá eliminar el ruido presente en la red eléctrica que afecta a los circuitos electrónicos.

Esta eliminación se realizara mediante el diseño de filtros digitales diseñados en el software Labview, el cual se convierte en un software ampliamente usado en la instrumentación virtual gracias a sus herramientas de uso. El programa cuenta con instrumentos en donde se puede diseñar fácilmente filtros y acondicionarlos para un buen funcionamiento.

En el capítulo 1: se detallaran los objetivos propuestos para este proyecto, así como los alcances y limitaciones de los mismos. Se detallara una metodología de investigación para así poder alcanzar satisfactoriamente todos y cada uno de los objetivos planteados.

En el capítulo 2: se describen los procedimientos matemáticos para la elaboración de filtros digitales, el entorno de programación gráfica en Labview y el uso de los diferentes componentes de la tarjeta de programación FPGA.

En el capítulo 3: se describen los procesos en software así como las herramientas usadas y la manera de implementarlas en el respectivo programa, se detallaran los diseños en hardware a implementarse como parte del proyecto planteado. Se incluirá diseños teóricos, pruebas correspondientes y el análisis de costos.

En el capítulo 4: se analizará la funcionalidad del diseño e implementación de filtros digitales en Labview.

En el capítulo 5: se detallan las conclusiones y recomendaciones respectivas para futuros proyectos que se fundamenten en análisis semejantes.

# **CAPÍTULO 1**

## **PLANTEAMIENTO DEL PROBLEMA**

### **1.1 Introducción**

En este capítulo se empezará por desarrollar la idea principal del problema a resolver que se basa en el inconveniente que causa el ruido eléctrico proveniente de la red eléctrica pública al conectar un circuito electrónico a la red. Se planteará un objetivo principal que se sustentará en mitigar el ruido eléctrico para que este no afecte a los circuitos electrónicos.

Sin duda los objetivos específicos ayudaran a aclarar el problema ya que mencionan temas como filtraciones parásitas, ruido eléctrico y filtros digitales.

Un tema importante en el capítulo será el de citar un antecedente previo de lo que ya se ha desarrollado para mitigar este problema del ruido eléctrico, y que tratamiento se le ha dado a las diferentes señales involucradas para así tener un panorama más claro de lo que se va a realizar en este proyecto.

No se podría dejar de lado el motivo por el cual la realización de este proyecto es importante y como aportara al estudiante de procesamiento digital de señales y a quien lo interese para entender el ruido eléctrico y la manera de atenuar esta señal parásita de un circuito electrónico.

Finalmente se especifica los alcances, limitaciones y metodología de investigación para tener un buen desarrollo de cada tema en el proyecto.



## **1.2 Problema a resolver**

Una de las necesidades que se presenta a la hora de entender el concepto de procesamiento digital de señales y de ejemplos reales de como modelar las diferentes señales del mundo en general es en sí una interrogante, que abarcan como tema de estudio los estudiantes de la carrera de ingeniería electrónica de la Universidad Politécnica Salesiana en niveles superiores de la carrera, para la cual solo se desarrolla simulaciones mediante el software Mitla; es decir, este tema queda inconcluso al no llevarlo a la práctica para su mejor entendimiento.

Este modelamiento real se lo puede plasmar con el uso de las Tarjetas de desarrollo Apartan 3E que se encuentran a disposición de los estudiantes de la carrera de ingeniería electrónica, los cuales necesitan tener en su formación este tipo de laboratorios para que a futuro estén en la capacidad de implementar un filtrado de señales que se presentan en diferentes tipos de circuitos, y saber cómo y cuándo aplicar los diferentes filtros.

Uno de los inconvenientes muy relevantes que constantemente se encuentra en el entorno de las telecomunicaciones, control industrial y el procesamiento digital de señales es el cómo eliminar las interferencias electromagnéticas generadas por la red eléctrica pública las cuales influyen directamente en los circuitos electrónicos a tal punto de generar señales bastante considerables de ruido, impidiendo así realizar un correcto muestreo y procesamiento de señales digitales. Para lo cual se diseñara e implementara un software mediante el cual se pueda eliminar estas señales de ruido generadas por la red eléctrica pública.

### 1.3 Hipótesis

A medida del paso de los años se ha hecho cada vez más ineludible el análisis de las señales generadas por la red eléctrica pública y como éstas afectan al ingresar a un circuito electrónico sea este de cualquier propósito.

Y más necesario se ha vuelto el tratamiento que se les dé a estas señales en cuanto a filtrarlas se refiere, ya que de esto depende en gran medida el tipo de respuesta que obtengamos de los circuitos electrónicos involucrados y como se puede evidenciar el cambio a la salida una vez que se le dé un debido tratamiento a las diferentes interferencias que encontremos del ruido.

Para esto se realiza un análisis detallado de todas las posibles causas de este fenómeno en los circuitos electrónicos, y en qué tipo de circuitos estas señales se hacen menos tratables, para así poder realizar un enfoque en el cual se tomaran las debidas decisiones del caso para filtrarlas y tratarlas de mejor manera.

Una vez alcanzado un nivel de entendimiento de lo que engloba el ruido provocado por la red eléctrica publica en los circuitos electrónicos, se toma como solución el de realizar un software mediante el cual se podrá dar mejor tratamiento a este tipo de señales indeseables en un circuito electrónico, realizando un diseño específico de un filtro que permitirá tener una señal de uso, más pura, es decir en mayores cantidades de portadora.

Para esto se empleará el software de programación grafica Labview, ya que este entorno es muy amigable con el usuario al momento de configurar o realizar la operación a futuro, a más de que el software Labview maneja las librerías para el manejo de las Tarjetas Apartan 3E mediante las cuales se realizará la interface mediante el hardware y el software.

Se propone desarrollar un Hardware el cual proporcione la señal a ser tratada, este circuito se diseñara para que adquiriera una señal física del ambiente que en este proyecto será la variación de la temperatura, la cual es una variable muy usual para ser tratada, para esto se empleará una termocupla tipo J muy común para la medición de procesos térmicos, y bastante fiable al momento de uso para adquirir una señal.

La finalidad del proyecto reflejara que las señales se pueden tratar

Finalmente las pruebas realizadas reflejaran la veracidad en cuanto al buen tratamiento que se puede dar a las señales de ruido provocadas por la red eléctrica que afectan a circuitos electrónicos para tener una buena fidelidad en la salida de un circuito electrónico.

#### **1.4 Tesis**

La tesis se plantea como el diseño, desarrollo e implementación de filtros digitales para eliminar ruido de la red eléctrica en un circuito electrónico, realizando una interfaz entre la tarjeta FPGA Apartan 3E y el software Labview.

## **1.5 Objetivos**

### **1.5.1 Objetivo general**

Diseñar, desarrollar e implementar filtros digitales para eliminar ruido de la red eléctrica en un circuito electrónico, utilizando la FPGA Apartan 3E y el software Labview

### **1.5.2Objetivos específicos:**

- Analizar los circuitos como son afectados por el ruido de la red eléctrica.
- Diseñar un filtro digital mediante el software Labview para eliminar el ruido de la red eléctrica, e implementarlo en el hardware de la tarjeta Apartan 3E.
- Construir un circuito analógico propenso a ser afectado por el ruido de la red eléctrica.
- Pruebas y ensayos de la efectividad del filtro digital para la eliminación del ruido provocado por la señal eléctrica.

## **1.6Justificación**

Debido a la gran aplicación y a su múltiple uso que tienen las tarjetas FPGA en la actualidad, se propone el desarrollo de una aplicación en la cual se puede emplear el hardware de estas tarjetas interrelacionando uno de los más completos software gráfico como lo es Labview, para obtener así una importante herramienta de uso la cual nos servirá para tener una ventaja frente a señales contaminadas con uno de los más comunes inconvenientes en cualquier circuito electrónico analógico o digital como lo es el ruido de 60 Hz provocado por la red eléctrica.

La aplicación que se propone servirá para familiarizarse con la programación de estos dispositivos electrónicos, debido a que el código VHDL en el que comúnmente se programa las tarjetas FPGA quedara asistido mediante el software de programación Labview el cual se maneja en un entorno más amigable de programación.

Esta herramienta será de gran utilidad a los estudiantes de procesamiento digital de señales, y para quienes desarrollen circuitos analógicos en los que interfiere el ruido eléctrico. Además el proyecto se lo desarrollara para que se pueda tener una mejor comprensión del comportamiento de las señales de ruido en circuitos electrónicos, facilitando la comprensión de los diferentes tipos de filtros y de interferencia que se presente, aprovechando las tarjetas FPGA se podrá entender los diferentes componentes que las comprenden y como se los puede emplear para la realización de cualquier tipo de proyecto.

### **1.7 Alcance**

La implementación de filtros digitales en el software Labview con interface hacia la tarjeta FPGA Apartan 3E realizará lo siguiente:

Controlará los módulos ADC y DAC de la tarjeta de desarrollo Apartan 3E.

Realizara una interface mediante la tarjeta FPGA Apartan 3E y Labview.

Tomará la señal de un circuito electrónico que usa una termocupla tipo J el cual genera la señal.

Desarrollo del filtro digital FIR(respuesta finita al impulso) desarrollado en el software de programación gráfica Labview.

## **1.8 Metodología de la investigación**

En este tema se puntualizarán los pasos previamente analizados para el desarrollo del proyecto, inicialmente se partirá de una detallada investigación bibliográfica la cual lleva al diseño construcción e implementación del circuito el cual generara las señales que se ingresaran a la tarjeta FPGA Apartan 3E, seguidamente se procederá al desarrollo del software en el cual se tratará las señales dependiendo de las condiciones planteadas previamente y del objetivo a alcanzar, finalizando con la documentación de todo el proyecto.

### **1.8.1 Método lógico deductivo**

Se efectuará una investigación para tener claro el diseño de filtros digitales FIR y IIR.

Se realizara una investigación de las particularidades a utilizar en este proyecto y de las características del modelo de programación gráfica Labview.

Subsiguientemente se investigaran las características y utilización de la tarjeta FPGA Spartan 3E al emplearse como un módulo para la adquisición de datos con interface hacia el software Labview.

Finalmente se realizara la conexión del circuito generador de señales a la tarjeta FPGA Spartan 3E y se realizara la interface con Labview.

### **1.8.2 Desarrollo del circuito generador de la señal**

Se diseñara un circuito el cual consta de dos partes, por un lado se obtendrá una señal acondicionada previamente la cual proviene de una termocupla tipo J, esta señal será para este proyecto la señal pura. Y por otro lado se tendrá una señal que proviene de un transformador que se lo acondicionara para tener un voltaje lo más aceptable para ser ingresado posteriormente a la tarjeta FPGA, esta señal se sumará a la señal de la termocupla para así generarle un ruido particular con esto finalmente se tendrá una señal con la cual se podrá realizar los análisis respectivos para este proyecto.

### **1.8.3 Desarrollo del software**

Se desarrollará el control de los diferentes módulos que intervienen en el presente proyecto partiendo de una etapa de adquisición de datos mediante la tarjeta FPGA Spartan 3E hacia el software Labview.

A continuación se desarrollará una etapa de conversión donde intervienen tanto los módulos internos que contiene la tarjeta FPGA como lo son el ADC y el DAC.

Finalmente se desarrollara los distintos filtros a emplear para este proyecto en los cuales se tratará las señales ingresadas dependiendo de los objetivos y análisis planteado.

#### **1.8.4 Construcción final del proyecto**

Se realizarán las pruebas correspondientes del proyecto mediante los instrumentos de medición como lo son el osciloscopio y el multímetro, para probar el correcto funcionamiento del mismo.



## **CAPÍTULO 2**

### **ESTADO DEL ARTE**

#### **2.1 Introducción**

En este capítulo se analiza el estado del arte de este proyecto, los fundamentos teóricos en el desarrollo de los filtros digitales implementados mediante el software de programación gráfica Labview con lo cual se detalla los fundamentos básicos para la programación mediante el software Labview.

#### **2.2 Estado del arte**

En la actualidad no existe publicado ningún proyecto el cual consista en la implementación de filtros digitales en la tarjeta FPGA Spartan 3E mediante el software de programación Labview.

#### **2.3 Marco teórico**

##### **2.3.1 Sistemas lineales, invariancia en el tiempo y causalidad**

###### **2.3.1.1 Linealidad**

Los sistemas lineales son sin duda la base de la teoría de sistemas. Estos sistemas se caracterizan por tener una específica relación de la entrada con la salida.

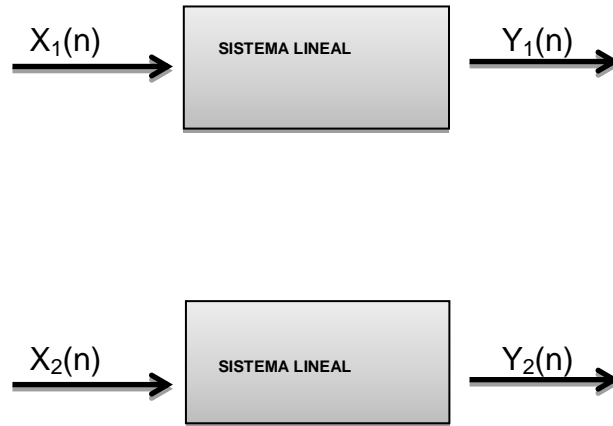
Es decir cuando se tiene a la entrada una suma ponderada de componentes de una señal, la salida es igual a la suma ponderada de las mismas salidas producida por cada componente.

Un sistema se denomina lineal si para dos valores escalares  $\alpha x_1$  y  $\alpha x_2$  cualquiera que estos sean se cumple a más que:

$$\alpha_1 Y_1(t) + \alpha_2 Y_2(t) = T[\alpha_1 x_1(t) + \alpha_2 x_2(t)] \quad (2.1)$$

Es decir la salida correspondiente cuando H (.) es lineal es:

**Figura 1. Diagrama de linealidad**



Elaborado por: (Alejandro Castillo, 2013)

Dónde:

$X_1$  y  $X_2$  son las entradas del sistema

$Y_1$  y  $Y_2$  son las salidas del sistema (Alan V. Oppenheim, 1998, pág. 53)

### **2.3.1.2 Invariancia en el tiempo**

Se denomina a un sistema invariante en el tiempo si la salida es la misma ante una misma entrada, sin importar el instante de tiempo en que la entrada sea aplicada. Es decir un sistema es invariante en el tiempo si

$$y(t) = T [x(t)] \rightarrow y(t - t_0) = T [x(t - t_0)] \quad (2.2)$$

(Alan V. Oppenheim, 1998, pág. 50)

### **2.3.1.3 Causalidad**

Un sistema es causal si para determinar la salida no depende de valores de entrada futuros. Cualquier filtro que se vaya a implementar en un sistema deberá contar como característica la causalidad de lo contrario no funcionara.

Un sistema causal se caracteriza por una respuesta al impulso  $h(t)$  que es cero para  $t < 0$ . (Alan V. Oppenheim, 1998, pág. 46)

## **2.3.2 Teoría de filtros**

### **2.3.2.1 Definición**

El filtro es un sistema que se lo diseñó de tal manera que se obtenga una función de transferencia deseada, es decir se lo puede entender como un proceso que altera la naturaleza de una señal sonora de una u otra forma.

Este sistema sigue un proceso computacional o Algorítmico, mediante el cual a una señal digital se la transforma en una secuencia de muestras o señal digital de salida.

Los filtros digitales forman una componente fundamental en toda cadena de comunicación. (Gutiérrez, 2009, pág. 2)

### 2.3.2.2 Respuesta impulsional y frecuencial de un filtro

La respuesta impulsional de un filtro es la reaccion del filtro a un impulso que se exita en su entrada. Esta respuesta se caracteriza en el filtro en el domino del tiempo.

Al realizar la transformada de Fourier de una respuesta impulsional de un filtro se estaría realizando la representación frecuencial de la señal, que caracteriza al filtro en el dominio frecuencial.(Gutiérrez, 2009, pág. 2)

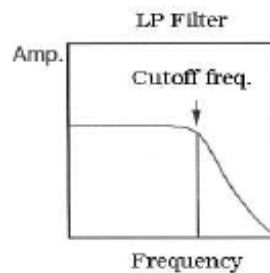
### 2.3.2.3 Tipos de filtros

Los tipos de filtros estan caracterizados por su frecuencia de corte y por su frecuencia central y ancho de banda.

Dentro de los filtros definidos por su frecuencia de corte estan los siguientes:

Filtros pasa bajo (LP) dejan pasar las frecuencias que estan por debajo de una determinada frecuencia.

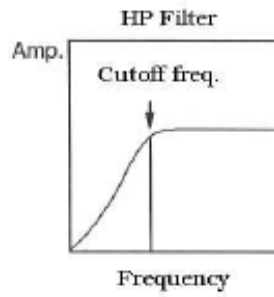
**Figura 2. Gráfica del filtro pasa bajo**



Fuente:(Gutiérrez, 2009)

Filtros pasa alto (HP) dejan pasar las frecuencias que estan por encima de una determinada frecuencia.

**Figura 3. Gráfica del filtro pasa alto**



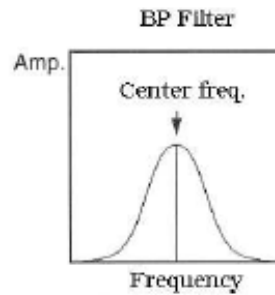
Fuente: (Gutiérrez, 2009)

Estos dos tipos de filtros estan definidos por la frecuencia de corte en donde la amplitud de la señal se reduce a  $0.707$  o  $\frac{1}{\sqrt{2}}$  de su valor máximo es decir tiene 3 db de atenuación.

Dentro de los filtros definidos por su frecuencia central y su ancho de banda se encuentran los siguientes:

Filtros pasa banda(BP) dejan pasar las frecuencias que estan situadas en una determinada banda de frecuencias (entre dos determinadas frecuencias).

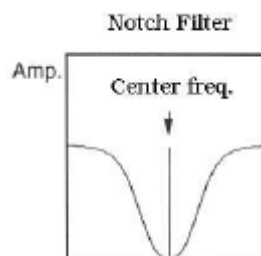
**Figura 4. Gráfica del filtro pasa alto**



Fuente: (Gutiérrez, 2009)

Filtros rechazo de banda (BR) dejan pasar todas las frecuencias excepto las que están situadas en una determinada banda de frecuencias (entre dos frecuencias determinadas  $f_1$  y  $f_2$ ).

**Figura 5. Gráfica del filtro pasa alto**



Fuente: (Gutiérrez, 2009)

Estos dos tipos de filtros se definen por su frecuencia central y por su ancho de banda que sería la diferencia entre la frecuencia superior y la inferior.

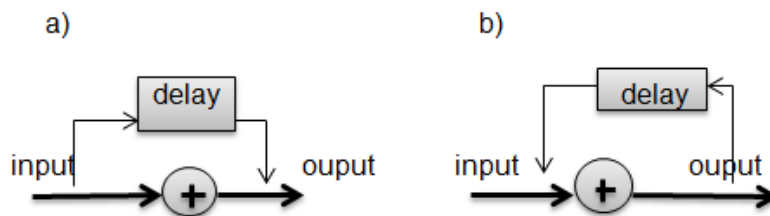
(Gutiérrez, 2009, pág. 3)

### 2.3.3 Filtros digitales

#### 2.3.3.1 Introducción

Existen dos tipos de filtros digitales dependiendo de su funcionamiento los cuales forman la base de un filtro digital, a continuación se los representa de manera gráfica:

**Figura 6. Diagrama en bloques de los dos tipos de filtros**



Elaborado por: (Alejandro Castillo, 2013)

En el gráfico de la figura (a) se muestra el tipo de filtro FIR los cuales se denominan respuesta impulsional finita.

En el gráfico de la figura (b) se muestra el tipo de filtro IIR los cuales se denominan respuesta impulsional infinita.

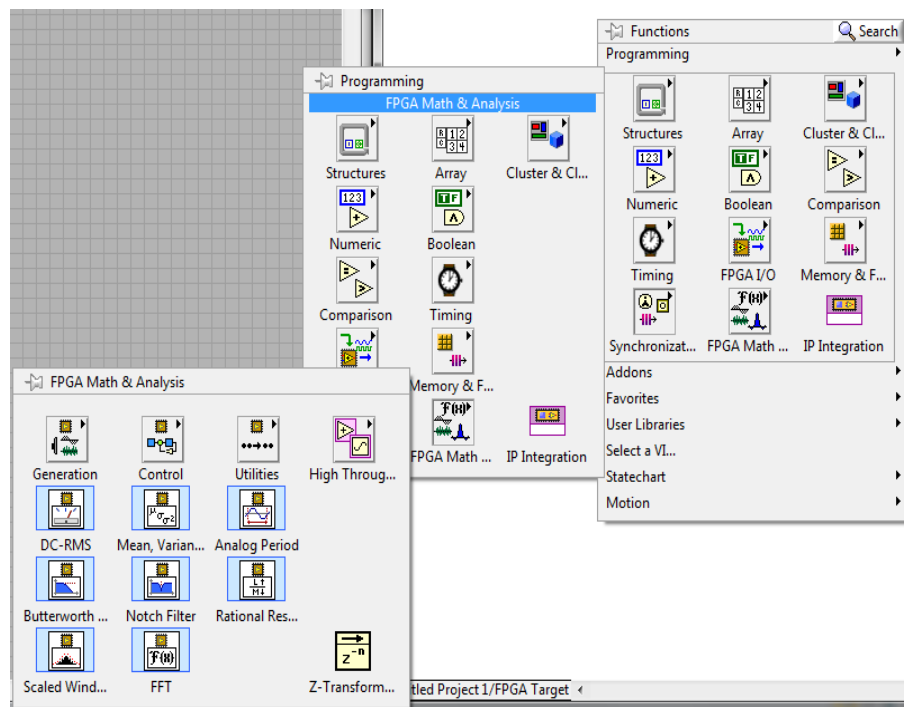
Labview cuenta con una serie de herramientas destinadas al uso e implementación de filtros digitales en un entorno de uso relativamente sencillo, sobre todo por su programación gráfica usando bloques, con lo cual para el diseño de filtros digitales basta con arrastrar los iconos correspondientes e interconectarlos a conveniencia, de esta manera se puede realizar proyectos en donde se tenga una gran flexibilidad de programación y diseño. (Gutiérrez, 2009, pág. 7)

## 2.3.4 Introducción a filtros digitales en Labview

### 2.3.4.1 Módulo labview-fpga

El software Labview cuenta con un módulo el cual permite la implementación de filtros digitales que permiten una interface entre la tarjeta FPGA Spartan 3E y el software, este módulo se encuentra en la pestaña de Programming-FPGA Math y Analysis, tal como se muestra en la siguiente figura

**Figura 7. Módulo de análisis matemático para fpga**



Fuente: (software Labview, 2010)

El software Labview posee drivers orientados a la interface con la tarjeta FPGA Spartan 3E los cuales permite tener una excelente compatibilidad con la tarjeta permitiendo al usuario programar e interactuar con ella desde el mismo software el cual incorpora en su funcionalidad herramientas para el control, a más un

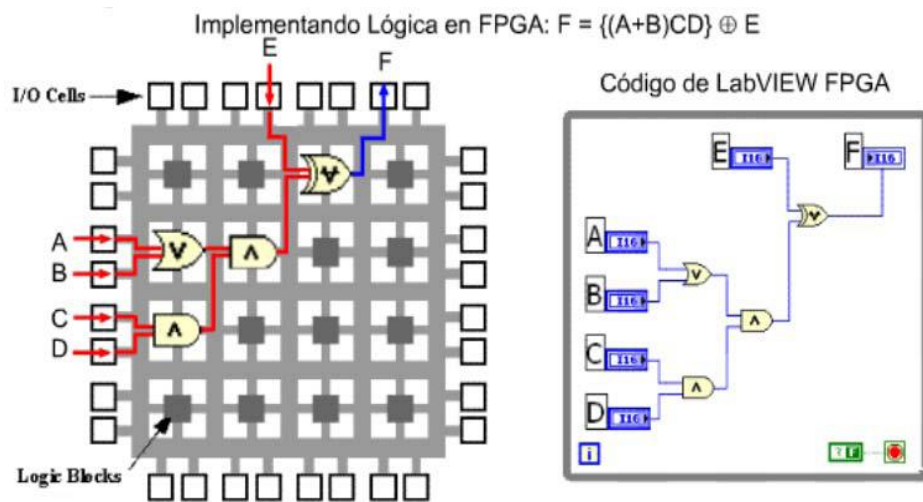


traductor y compilador los cuales convierten el código gráfico a uno que entienda la tarjeta FPGA y lo transfieren a este.

En el software Labview se puede crear el programa que se ejecutara sobre la tarjeta FPGA mediante el módulo Labview-FPGA este módulo convierte internamente el programa en código VHDL, lo reduce y sintetiza a través de la herramienta Xilinx ISE Compilery lo transforma en archivo de bits el mismo que se transfiere a una FPGA para allí ejecutarse.

Labview es sin duda un software muy adecuado para describir el funcionamiento de una FPGA ya que tiene un modelo de ejecución de datos altamente concurrente, pudiendo conseguir de esta manera un paralelismo a nivel de hardware.

**Figura 8. Estructura de programa en fpga vs estructura de programación en labview**



Fuente:(Mikel, 2009)

Existen diversas aplicaciones las cuales emplean el módulo LABVIEW-FPGA, a continuación se describen las más comunes:

**Control a alta velocidad:** se refiere a la velocidad en ejecución que se puede lograr usando FPGA, Labview-FPGA cuenta con paletas de control de algoritmos integrados para implementados en filtros o para controles PID estos son estándares en cualquier sistema de control *Adquisición de datos inteligentes:* el módulo Labview-FPGA permite ingresar señales en donde se pueda tener un muestreo de las mismas o donde se tenga señales muy personalizadas dependiendo la aplicación que se quiera, acondicionando el sistema de manera autónoma y permitiendo una generación de más entradas de señales como por ejemplo E/S analógicas personalizadas, disparos analógicos personalizados, contadores, etc., realizando su ejecución de forma paralela es decir todos los procesos se podrán generar en la misma tarjeta.

**Protocolos de comunicación digital:** el módulo Labview-FPGA permite la implementación de protocolos como i2c, RS232, SPI, o protocolos específicos implementados dependiendo para la aplicación que se requiera.

**Simulación de sensores:** una de las facilidades que presenta este módulo es la de realizar pruebas de sensores únicamente realizando una simulación, esto es de gran ayuda ya que no necesariamente se requerirá del prototipo a testear para realizar la prueba de hardware, con esto se podrá ahorrar tiempo y recursos.

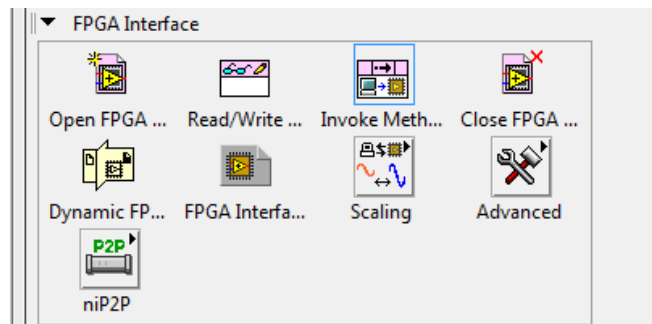
**Procesamiento en tarjeta y reducción de datos:** esta es una aplicación más específica la cual usa las tarjetas FPGA, permitiendo básicamente reducir la carga sobre el procesador, con el fin de procesar los datos que se estén manejando antes de enviarlos hacia el host final, se usa la FPGA para que realice trabajos como codificación o decodificación es decir se podría usar la FPGA como un procesador más, para realizar un pre-procesado de las señales antes de enviar los datos hacia el host donde se procesara la información, de esta forma se

puede tener al CPU del host final realizando operaciones necesariamente que se deber realizar ahí.

**Co-procesamiento:** esta aplicación a diferencia de la anterior al tener una comunicación de dos vías entre la CPU del host y la tarjeta FPGA permite enviar información de un lado al otro y cada uno realiza un pre-procesado.

Lo interesante que resulta la programación en un entorno grafico como lo presenta Labview, es su fácil diseño de circuitería que permite al usuario arrastrar los diferentes componentes que se encuentran en sus paletas módulos de interface Labview-FPGA dependiendo del diseño a implementar, evitándose así una programación en lenguaje VHDL.

**Figura 9. Funciones del palet fpga-interface de labview**



Fuente: (software Labview, 2010)

Resulta de gran ayuda al usuario las herramientas de programación que facilita Labview al momento de realizar una programación Labview-FPGA más aun cuando no se requiere de una programación en código aceptable directamente para la FPGA es decir VHDL ya que Labview internamente se encarga de esto. (Mikel, 2009, págs. 8-17)

### 2.3.4.2 Módulo de funciones para el procesamiento de señales

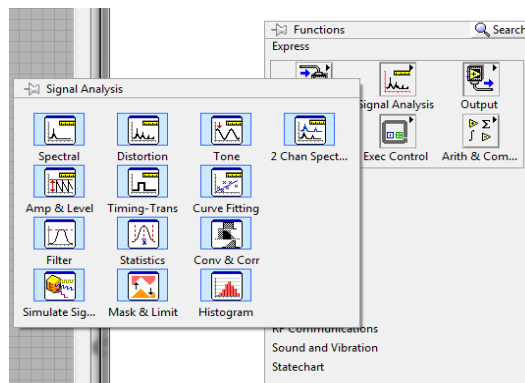
Labview contiene en su entorno de programación paletas de funciones de herramientas las cuales permiten realizar un análisis en cuanto a la experimentación y desarrollo que involucran tiempo y frecuencia de ondas de señales y análisis en serie de tiempo.

Dentro de los diferentes tipos de análisis que se puede realizar se tiene:

- Análisis tiempo-frecuencia para señales en función de la frecuencia.
- Análisis en series de tiempo, enfocado al control.
- Análisis y diseño para implementar filtros dependiendo del tipo de señal que se pretenda filtrar.

Estas funciones se encuentran en la ventana principal de la paleta de funciones generales con el nombre de Signal Analysis (Análisis de señales) en donde se despliegan todas las herramientas que ofrece Labview por el análisis de una señal en función del tiempo o de la frecuencia.(NATIONAL INSTRUMENTS CORPORATION, 2003)

**Figura 10. Funciones de procesamiento de señales**



Fuente: Captura de pantalla del software Labview 2010

### 2.3.4.3 Filtro notch

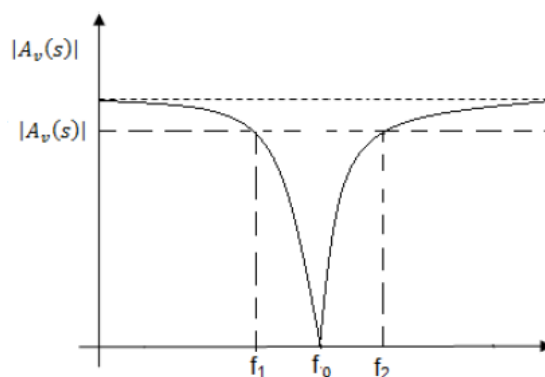
El filtro notch es conocido como el filtro rechaza banda o elimina banda, la característica de este filtro es el de rechazar una frecuencia determinada que este interfiriendo a un circuito, cuando se tiene una frecuencia considerablemente estable, un filtro notch es suficiente para eliminar la frecuencia no deseada.

El ruido se puede eliminar mediante un filtro notch si la distribución es centrada exactamente a la frecuencia para la cual el filtro fue diseñado, pero si la frecuencia no es centrada el filtro no cumplirá con la función de eliminar ciertas frecuencias correctamente.

Se puede escoger como se va a ajustar la frecuencia de corte de dos maneras, especificando la calidad (Quality) o especificando el ancho de banda (Bandwith).

A continuación se muestra una gráfica de la curva característica del filtro notch.

**Figura 11. Curva característica del filtro notch**



Fuente: (Lucelly Reyes, 2008)

Labview ofrece en su paleta de herramientas el diseño de filtros digitales para el procesamiento de señales, los mismos que se encuentran en la paleta de FPGA Math and Analysis>NothFilter.

Se trata de un exprés Vi por lo que al ponerlo en el diagrama de bloques se abrirá un cuadro de diálogo en el cual se modificará los parámetros en función de lo requerido.(Wikimedia Inc, 2001, págs. 1,2)

#### 2.3.4.4 Filtros pasivos de primer orden pasa bajos

Los siguientes circuitos son usados como filtros de primer orden de tipo pasivo.

**Figura 12. Circuitos usados como filtros de primer orden**



Elaborado por: (Alejandro Castillo, 2013)

Este tipo de filtros se implementan para impedir los escalones que se forman a la salida o a la entrada del circuito.

Para este tipo de circuitos el capacitor tiene un comportamiento como una resistencia dependiente de la frecuencia por la siguiente relación:

$$X_c = \frac{1}{C2\pi f} \quad (2.3)$$

Es así que para frecuencias bajas el capacitor consumirá todo el voltaje y a la salida de él se tendrá todo el voltaje de entrada.

Conforme aumenta la frecuencia de entrada el capacitor disminuye su impedancia con lo cual el voltaje disipado disminuye, hasta tender a cero.

Para el cálculo de la frecuencia de corte  $f_c$  del filtro se utiliza mediante la siguiente ecuación:

$$f_c = \frac{1}{RC} \quad (2.4)$$

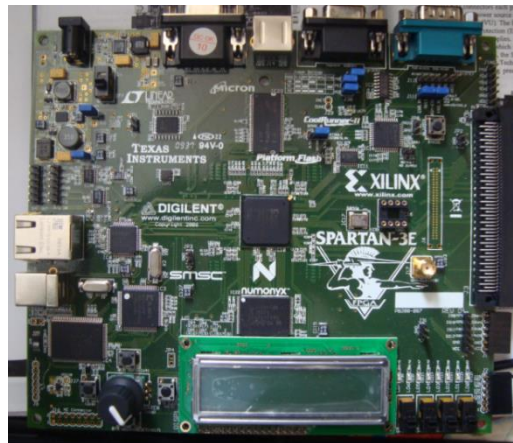
(European Software Institute, 2004, págs. 2,4)

### 2.3.5 Tarjeta fpga spartan 3e

#### 2.3.5.1 Características de la tarjeta fpga spartan 3e

La tarjeta FPGA Spartan 3E fabricada por Xilinx que por sus siglas en inglés es un arreglo de compuertas programables de campo el cual permite la implementación de diferentes tipos de circuitos digitales.

**Figura 13. Tarjeta fpga spartan 3e**



Fuente: (XILINX ALL PROGRAMMABLE, 2006)

A continuación se presentan las características principales de la tarjeta de Xilinx XC3S500E Spartan 3E-FPGA:

- Hasta 232 I/O de usuario.
- 320 paquetes de pines FBGA.
- Cuenta con más de 10.000 células de lógica.
- Cuenta con 4Mbit de configuración PROM.
- 64macrocélulaXC2C64ACoolRunnerCPLD.
- 64 Mbyte(512Mbit)de SDRAMDDR, los datos de interfazx16, 100 + MHz.
- 16 MByte (128Mbit) del paralelo NOR Flash (Intel Strata Flash).
- Cuenta con configuración FPGA de almacenamiento.
- 16Mbits de serie SPI Flash (STMicro).
- Tiene una pantalla LCD de 2-líneas, 16 caracteres.
- Puerto del teclado o para ratón es PS/ 2.
- Puerto para pantalla VGA.
- 10/100 Ethernet PHY (require Ethernet MAC en FPGA).
- Dos puertos de 9 pines Rs-232.
- Interfaz de bug basado en USB FPGA.
- El oscilador del reloj es de 50Mhz.
- SHA-1 1-cable seria Leeprom de flujo de bits de protección de copia.
- Conector de expansión Hiro se FX2.
- Tres conectores de expansión de 6 pines.
- Cuatro salidas, SPI basado en conversor Digital a analógico (DAC).
- Dos entradas SPI basado en conversor analógico digital (ADC) con ganancia programable pre-amplificador.
- Encoder rotatorio con eje pulsador.
- Ocho leds discretos.
- Cuatro interruptores deslizantes.
- Características tomadas del Anexo 1.





Donde  $G$  es la ganancia del pre-amplificador análogo para el ADC,  $V_{REF}$  es el voltaje de referencia común entre el ADC y el pre amplificador,  $V_{FS}$  es el voltaje de conversión full-scale para el ADC,  $V_{IN}$  es la señal análoga del pre amplificador y  $n$  es el número de bit de resolución.

Se tiene que cuando  $V_{IN}$  es menor que el voltaje de referencia los datos de salida  $D[n-1:0]$  son en complemento a dos con los bits más significativos representando con el bit de signo, y un número cuyo valor está entre  $-2^{n-1}$  y  $2^{n-1} - 1$ .

La ganancia de cada amplificador se puede programar en función de la ecuación (2.5) y los diferentes valores de salida para el amplificador se muestran en la siguiente tabla:

**Tabla 1. Ganancias para el amplificadorltc6912-1**

Ganancia del canal A	A3	A2	A1	A0	Rango de voltaje de entrada	
					Voltaje mínimo	Voltaje máximo
0	0	0	0	0	0	0
-1	0	0	0	1	0.4	2.9
-2	0	0	1	0	1.025	2.275
-5	0	0	1	1	1.4	1.9
-10	0	1	0	0	1.525	1.775
-20	0	1	0	1	1.5875	1.7125
-50	0	1	1	0	1.625	1.675
-100	0	1	1	1	1.6375	1.6625

Fuente: (XILINX ALL PROGRAMMABLE, 2006)

Los valores de las ganancias para la entrada análoga del canal B son las mismas ya que se calculan con la ecuación (2.5).

Para configurar el pre amplificador es necesario deshabilitar ciertos dispositivos los cuales comparte con el ADC y el DAC ya que maneja la misma comunicación SPI(Interfaz periférico serial), se lo realiza en función de las siguientes tablas:

**Tabla 2. Deshabilitar otros dispositivos en el bus spi**

Signal	Disabled Device	Disable Value
SPI_SS_B	SPI Serial Flash	1
AMP_CS	Programmable Pre-Amplifier	1
DAC_CS	DAC	1
SF_CE0	StrataFlash Parallel Flash PROM	1
FPGA_INIT_B	Platform Flash PROM	1

Fuente: (XILINX ALL PROGRAMMABLE, 2006)

**Tabla 3. Señales de interfaz del pre amplificador**

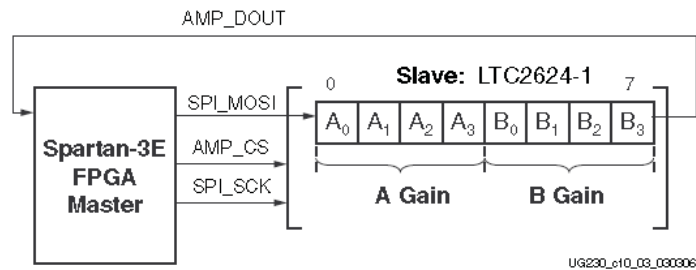
Signal	FPGA Pin	Direction	Description
SPI_MOSI	T4	FPGA → AD	Serial data: Master Output, Slave Input. Presents 8-bit programmable gain settings, as defined in <a href="#">Table 10-2</a> .
AMP_CS	N7	FPGA → AMP	Active-Low chip-select. The amplifier gain is set when signal returns High.
SPI_SCK	U16	FPGA → AMP	Clock
AMP_SHDN	P7	FPGA → AMP	Active-High shutdown, reset
AMP_DOUT	E18	FPGA ← AMP	Serial data. Echoes previous amplifier gain settings. Can be ignored in most applications.

Fuente: (XILINX ALL PROGRAMMABLE, 2006)

La interfaz de la comunicación del bus SPI se basa en el sincronismo de las señales que maneja el pre amplificador y la ganancia para cada amplificador se

envía en una palabra de comando que contiene 8 bits, el cual consiste 4 bits para el campo que maneja cada canal siendo el bit más significativo el bit B3 la representación se muestra en el siguiente gráfico:

**Figura 15. Interfaz serial del amplificador**



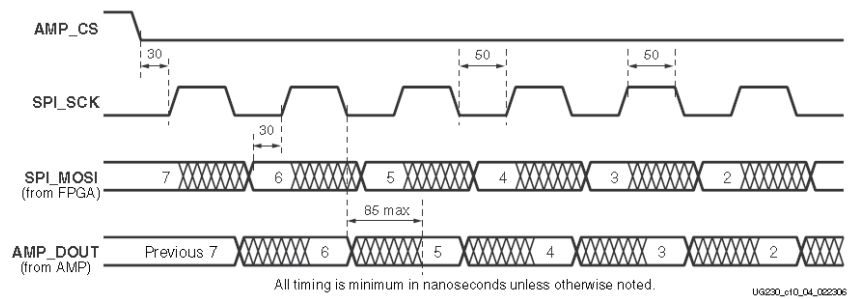
Fuente: (XILINX ALL PROGRAMMABLE, 2006)

La señal de salida del amplificador AMP\_DOUT se referencia de los ajustes anteriores de ganancia, para la mayor parte de aplicaciones este valor se ignora.

La trama que envía el bus SPI empieza cuando la señal del chip de selección(AMP\_CS) es activado en bajo, los datos son capturados mediante el SPI\_MOSI en flanco ascendente de la señal de reloj (SPI\_SCK).

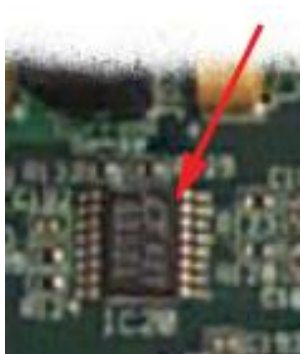
El amplificador presenta los datos en serie mediante al AMP\_DOUT en el siguiente flanco descendente de la señal de reloj, en el siguiente gráfico se ilustra las señales del cómo se comunica el amplificador:(XILINX ALL PROGRAMMABLE, 2006, pág. 75)

**Figura 16. Tiempos de comunicación del amplificador**



Fuente: (XILINX ALL PROGRAMMABLE, 2006)

### 2.3.5.3 Módulo conversor análogo digital (adc)



En el XILINX ALL PROGRAMMABLE user guide detalla al respecto del LTC1407A-1 que dispone de dos centros de retención. Las dos entradas analógicas son muestreadas simultáneamente cuando la señal del conversor análogo-digital es aplicada.

La velocidad máxima a la que el dispositivo opera es de 1.5MHz esto se da solo para cuando se tiene conversión y comunicación.

En la siguiente tabla se enumera las diferentes señales con las cuales realiza una interfaz el conversor análogo-digital.

**Tabla 4. Señales de interfaz adc**

Signal	FPGA Pin	Direction	Description
SPI_SCK	U16	FPGA→ADC	Clock
AD_CONV	P11	FPGA→ADC	Active-High shutdown and reset.
SPI_MISO	N10	FPGA←ADC	Serial data: Master Input, Serial Output. Presents the digital representation of the sample analog values as two 14-bit two's complement binary values.

Fuente: (XILINX ALL PROGRAMMABLE, 2006)

El ADC se configura para que opere con un dispositivo SPI. El reloj del SPI es de 50MHz, pero el ADC para realizar la transición en su FSM y por seguridad de los datos opera únicamente a 12.5MHz.

La señal de entrada del DAC\_CS al igual que la del DAC\_CLR se activa en bajo para el esclavo de selección del ADC y para el reset asíncrono del DAC.

Cuando la señal del ADC pasa a estado alto se muestrean simultáneamente los dos canales, los resultados de la conversión no se presentan hasta la próxima vez en que la señal sea afirmativa es decir se garantiza una latencia de una muestra.

Mediante la siguiente ecuación se puede calcular la velocidad aproximada del rendimiento de los datos:

$$R_{adc} = \frac{12.5MHz}{\frac{3}{2} + 34bit/paquete} = 0.339 \text{ muestras/s} \quad (2.6)$$

Para los dos canales tanto A y B el ADC tiene un paquete de 34-bit los cuales se clasifican de la siguiente manera:

**Tabla 5. Paquete de datos que ofrece el adc**

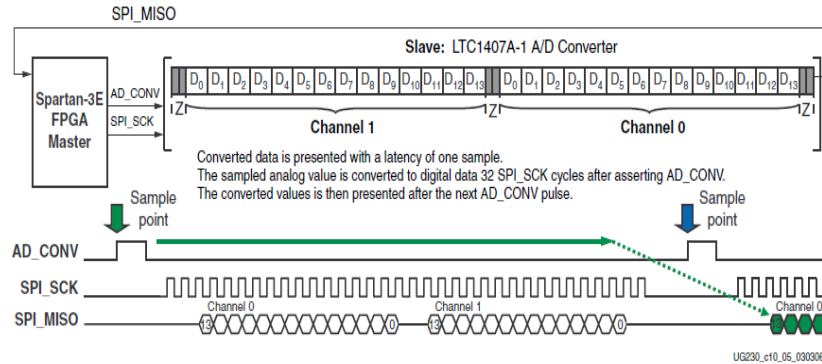
Bits	Contenido (MSB...LSB)
33-32	No importan
31-18	Datos del canal A en el ADC
17-16	No importan
15-02	Datos del canal B en el ADC
1-0	No importan

Fuente: (XILINX ALL PROGRAMMABLE, 2006)

Cada canal (A y B) incorporan un ADC y las entradas se muestrean simultáneamente cuando es aplicada la conversión, la salida digital de datos en el ADC es de 14-bit.

Luego que la señal ha sido activada se usan 32 ciclos de la señal SCK para convertir el muestreo para una etapa digital, en la siguiente figura se muestra un ejemplo de la transición del bus SPI al ADC.

**Figura 17. Interfaz del conversor análogo-digital**



Fuente:(XILINX ALL PROGRAMMABLE, 2006)

Las entradas análogas del ADC y del amplificador programable se ingresan por el header de 6 pines (J7).

Para saber si el voltaje que está ingresando al ADC de la tarjeta FPGA es el adecuado se usa la ecuación (2.5) y reemplaza los valores iniciales de acuerdo al proyecto:

$$D[n-1:0] = G \frac{V_{IN} - V_{REF}}{V_{FS}} 2^{n-1}$$

Dónde:

$$G = -1$$

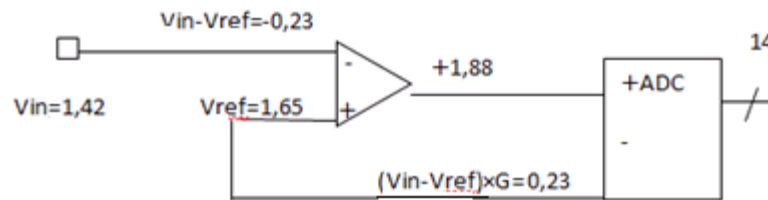
$$V_{IN} = 1,42V$$

$$V_{REF} = 1,65V$$

$$V_{FSC} = 1,25V$$

$$N = 14$$

Se tiene:



El voltaje que se tiene a la salida del pre amplificador es de:

$$V_{OUT} = 1,88$$

El cual es un valor que satisface a las condiciones de ingreso para la ganancia del ADC de la tarjeta FPGA y favorece al muestreo de la señal de entrada permitiendo que se reconstruya favorablemente.

Debido a que el ADC comparte la comunicación del bus SPI con otros dispositivos, es necesario que para su activación estos dispositivos sean desactivados. (XILINX ALL PROGRAMMABLE, 2006, págs. 77-79) Para esto se fundamenta en la siguiente tabla:

**Tabla 6. Deshabilitación de otros dispositivos del bus spi**

Signal	Disabled Device	Disable Value
SPI_SS_B	SPI Serial Flash	1
AMP_CS	Programmable Pre-Amplifier	1
DAC_CS	DAC	1
SF_CE0	StrataFlash Parallel Flash PROM	1
FPGA_INTT_B	Platform Flash PROM	1

Fuente: (XILINX ALL PROGRAMMABLE, 2006)



#### 2.3.5.4 Módulo conversor digital análogo (DAC)

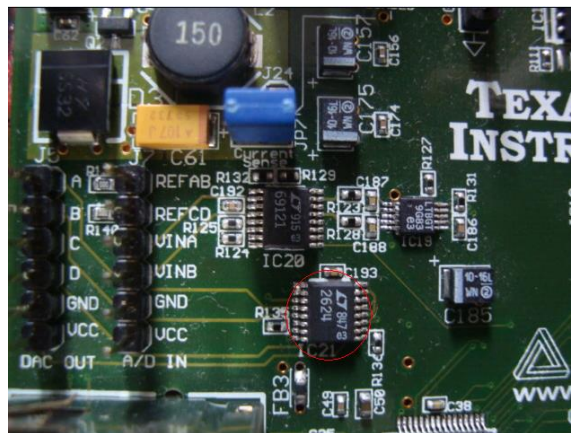
En el XILINX ALL PROGRAMMABLE USER GUIDE acota que la tarjeta Spartan 3E incluye un SPI compatible de cuatro salidas análogas independientes para la conversión digital-análoga (DAC), estas salidas se encuentran en el conector J5 opera al igual que el ADC con un dispositivo de tecnología lineal LTC2624.

Cada salida análoga es equivalente a 12-bit. En los canales de salidas se tiene dos voltajes diferentes los cuales se dividen de la siguiente manera:

Las salidas A y B están en el rango de 0 a 3.3V.

Las salidas C y D están en el rango de 0 a 2.5V.

**Figura 18. LTC2624-DAC**



Fuente: (XILINX ALL PROGRAMMABLE, 2006)

La Frecuencia de reloj del LTC2624 es de 50MHz igual a la frecuencia del oscilador de la tarjeta Spartan 3E, por lo que se necesita disminuirla ya que la frecuencia de reloj del LTC2624 debe ser menor a la frecuencia del oscilador de la tarjeta para esto se emplea el módulo reloj digital principal (DCM) el cual

permite bajar la frecuencia y se basa en el cálculo matemático tal como lo describe la siguiente ecuación:

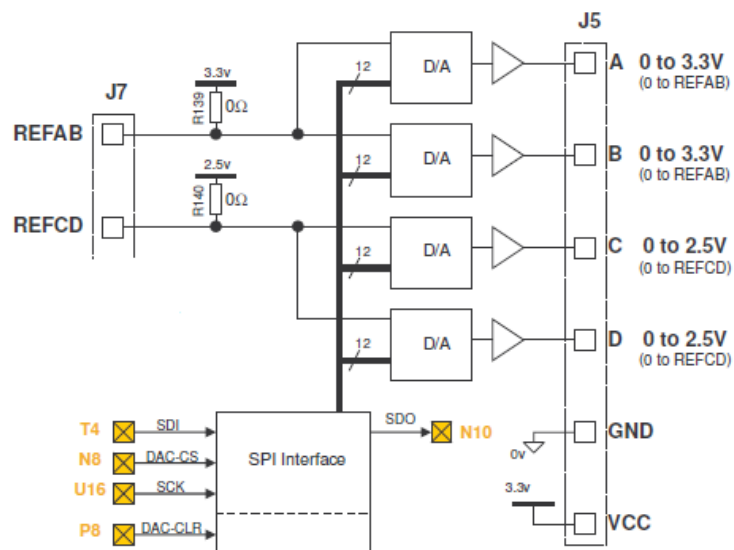
$$f_{sintet} = 50MHz \times \frac{5}{3} = 83.33 \text{ MHz} \quad (2.7)$$

Al final se divide para dos ya que la frecuencia debe ser menor a la de los 50Mhz del oscilador de la tarjeta de FPGA Spartan 3E, esto garantiza que exista la transición de la FSM en el interior del DAC, finalmente realizando la operación correspondiente a la ecuación (2.7) se tiene:

$$f_{sintet} = \frac{83.33MHz}{2} = 46.66MHz \quad (2.8)$$

El dispositivo LTC2624 mediante una interfaz periférica serial (SPI) incorporada permite suministrar a cada canal los valores digitales para que puedan ser controlados, a continuación se muestra un direccionamiento de pines hacia la tarjeta FPGA Spartan 3E:

**Figura 19. Diagrama de bloques de conexiones del dispositivo LTC2624**



Fuente: (XILINX ALL PROGRAMMABLE, 2006)

A continuación se detalla el paquete de datos que ofrece el DAC:

**Tabla 7. Paquete de datos que ofrece el DAC**

Bits		Contenido (MSB...LSB)
23-20	Comandos	0000 Escribir para el registro de entrada DAC
		0001 Actualizar el registro del DAC
		0010 Escribir registro de entrada DAC y actualiza todos los DACs
		0011 Escribir y actualiza el DAC
		0100 Apaga el DAC
19-16	direcciones	0000 DAC A
		0001 DAC B
		0010 DAC C
		0011 DAC D
		1111 DACs A,B,C,D
15-abr	Datos	12-bit de dato del DAC, primero el MSB
3-0	No importa	

Fuente: (XILINX ALL PROGRAMMABLE, 2006)

Las salidas del DAC tienen una equivalencia analógica de un valor digital de 12-bit y está en función de la siguiente ecuación:

$$V_{OUT} = \frac{D[11:0] \times V_{REF}}{4096} \quad (2.9)$$

Dónde:

D [11:0] representa el paquete de datos de 12-bit

V<sub>REF</sub> viene dado por la tarjeta 3.3V o 2.5V

4096 Mayor número binario del paquete. (XILINX ALL PROGRAMMABLE, 2006, págs. 67-71)

Cada valor de salida se representa en la siguiente tabla de valores:

**Tabla 8. Salidas analógicas del DAC**

Bits	V OUT
0	0,000805V
1	0,00161V
2	0,00322V
3	0,00645V
4	0,012890625V
5	0,2578125V
6	0,0515625V
7	0,103125V
8	0,20625V
9	0,4125V
10	0,825V
11	1,65V

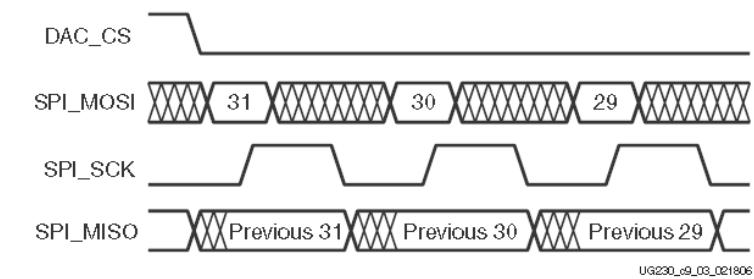
Fuente: (XILINX ALL PROGRAMMABLE, 2006)

#### **2.3.5.5 Detalles de la comunicación**

Respecto a los tiempos de comunicación que maneja el bus SPI, cada bit es transmitido o recibido en sincronismo a la señal SPI-SCK que es la señal de reloj, y es mediante esta señal por la cual el DAC escribe los valores hacia el puerto de salida. El LTC2624 soporta un tiempo de muestreo no mayor que 50MHz.

En la figura 16 se puede observar las diferentes formas de onda de la comunicación del bus SPI para el DAC.(XILINX ALL PROGRAMMABLE, 2006, págs. 76-79)

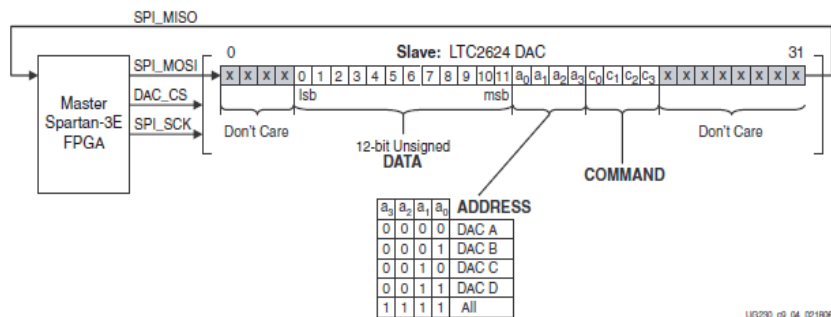
**Figura20. Formas de onda de la comunicación spi del DAC**



F  
Fuente: (XILINX ALL PROGRAMMABLE, 2006)

El DAC recibe los datos en una palabra de 32bits los cuales se clasifican en función de la siguiente figura:

**Figura 21. Protocolo de comunicación del LTC2624**

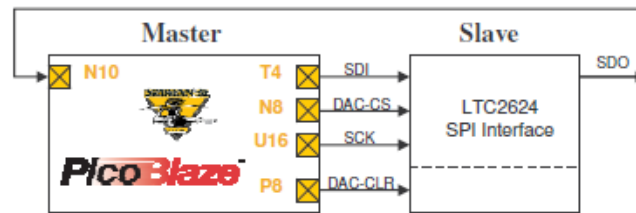


Fuente: (XILINX ALL PROGRAMMABLE, 2006)

### 2.3.5.5Módulo de comunicación SPI

El módulo de interfaz de periférico serial permite la comunicación de valores digitales de cada uno de los cuatro canales, la comunicación se realiza mediante un sincronismo full-duplex el cual incorpora un canal con cuatro cables de interfaz. La comunicación la se realiza mediante el dispositivo LTC2624, cuando la señal de selección DAC-CS es baja.

**Figura 22. Comunicación SPI**



Fuente: (XILINX ALL PROGRAMMABLE, 2006)

Las señales de interfaz que maneja la salida del DAC se muestran en la siguiente figura:

**Tabla 9. Señales de interfaz del DAC**

Signal	FPGA Pin	Direction	Description
SPI_MOSI	T4	FPGA → DAC	Serial data: Master Output, Slave Input
DAC_CS	N8	FPGA → DAC	Active-Low chip-select. Digital-to-analog conversion starts when signal returns High.
SPI_SCK	U16	FPGA → DAC	Clock
DAC_CLR	P8	FPGA → DAC	Asynchronous, active-Low reset input
SPI_MISO	N10	FPGA ← DAC	Serial data: Master Input, Slave Output

Fuente: (XILINX ALL PROGRAMMABLE, 2006)

Para que el DAC pueda realizar su correcta transmisión de datos es necesario desactivar ciertos elementos que comparte el DAC en la comunicación mediante el bus SPI y se lo realiza mediante los datos que se muestra en la siguiente tabla. (XILINX ALL PROGRAMMABLE, 2006, pág. 76)

**Tabla 10. Des habilitación de otros dispositivos en el bus SPI**

Signal	Disabled Device	Disable Value
SPI_SS_B	SPI serial Flash	1
AMP_CS	Programmable pre-amplifier	1
AD_CONV	Analog-to-Digital Converter (ADC)	0
SF_CE0	StrataFlash Parallel Flash PROM	1
FPGA_INIT_B	Platform Flash PROM	1

Fuente: (XILINX ALL PROGRAMMABLE, 2006)

### **2.3.5.6 Módulo oscilador de cristal**

La tarjeta incorpora un oscilador de cristal de 50MHz con una salida de ciclo de trabajo al 40% o 60%, el oscilador trabaja con un rango de error mínimo es así que permite un exactitud  $\pm 2500$  Hz o  $\pm 50$ ppm.(XILINX ALL PROGRAMMABLE, 2006, pág. 69). Las características más relevantes se encuentran en el Anexo 3.

### **2.3.6 Sensores y acondicionadores**

#### **2.3.6.1 Termocupla tipo j**

**Figura 23. Fotografía de la termocupla**



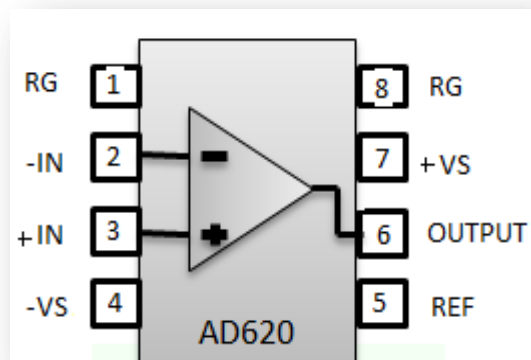
Fuente: Captura de la termocupla tipo J

Para el desarrollo del presente proyecto se utilizó una termocupla tipo J la cual es conocida como la termocupla hierro-constantán, el constantán es una aleación de cobre-níquel con un porcentaje de cobre entre 45% y 60% se eligió esta ya que tiene un satisfactorio desempeño frente a condiciones bastante oxidantes y donde se tiene un vacío de hasta los 760°C, también presenta un comportamiento bastante estable lo cual permite tener una salida muy pura característica que es de bastante ayuda para el presente proyecto donde se requiere tener una salida sin ruido, otra característica a más de las condiciones de trabajo a las que opera es su bajo costo en comparación con las demás que existe en el mercado.

Las características principales de funcionamiento así como las limitaciones se encuentran en el anexo 4 (TERMOKEW S.A DE C.V, 2001, págs. 1-2)

#### 2.3.6.2 Amplificador AD620

**Figura 24. Diagrama de terminales del amplificador de instrumentación AD620**



Elaborado por: (Alejandro Castillo, 2013)

Debido a que los voltajes que entrega la termocupla tipo J están en el orden de los mili voltios, se necesita acondicionar esta señal al orden de los voltios para poder trabajar posteriormente, por lo que se escogió el amplificador AD620 para



acondicionar la señal, el cual tiene una ganancia de 1000 con lo que fácilmente se logra una salida de voltaje que se encuentra en el orden de los voltios.

Este es uno de los amplificadores muy usados debido a su alta relación costo/desempeño, y una de las características que se toma en cuenta para este proyecto que brinda el AD620 es que el ruido en la entrada de voltaje es reducido a valores de  $9n/\sqrt{hZ}$  por lo que se tendrá a la salida una señal muy pura.

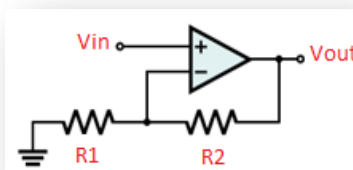
Dentro de las principales aplicaciones se encuentra:

- Instrumentación médica.
- Básculas electrónicas.
- Amplificación de transductores.
- Etc.

Los detalles de las características de funcionamiento se encuentran en el datasheet del amplificador en el Anexo 5.(Inc, 2013)

### 2.3.6.3 Amplificador operacional lm358

**Figura 25. Diagrama del amplificador operacional LM358**



Elaborado por: (Alejandro Castillo, 2013)

El LM358 es un amplificador operacional no inversor, donde la tensión de salida se calcula de la siguiente fórmula:

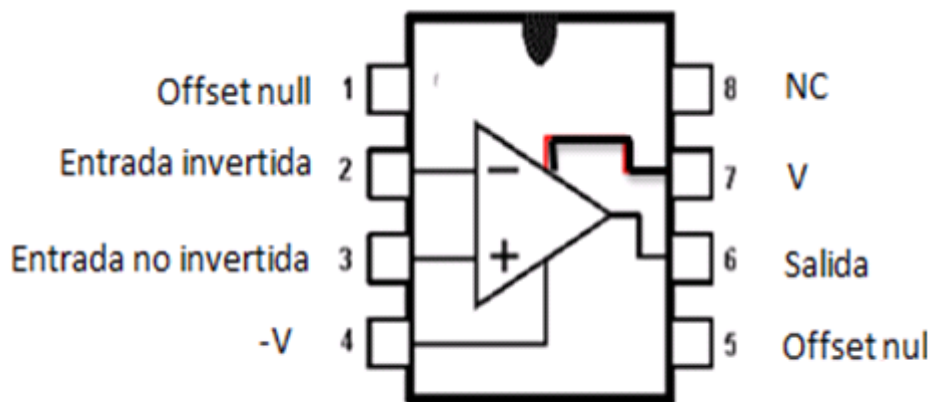
$$V_{out} = V_{in}\left(1 + \frac{R_2}{R_1}\right) \quad (2.10)$$

Mediante la fórmula (2.12) nos permite un control simple de ganancia incluso con valores muy elevados. Una limitación a ser considerada es que la tensión de salida estará por debajo de la tensión de alimentación positiva, por lo que permitirá una amplificación de una señal sinusoidal en el rango de los milivoltios pico a pico a una señal de 16Vpp fácilmente.

Para mayor información ver anexo 7.(Robert F. Coughlin, 1999, págs. 44-67)

#### 2.3.6.4 Amplificador operacional lm741

**Figura 26. Diagrama del amplificador operacional LM741**



Elaborado por: (Alejandro Castillo, 2013)

El LM741 es un operacional de propósito general, el mismo que se puede usar en diferentes configuraciones dependiendo la necesidad o las condiciones para las cuales se lo requiera.

Las características que ofrecen los amplificadores operacionales son muchas, lo que hace que su utilización sea casi infalible, dentro de las cuales ofrecen protección en sobrecarga tanto en la entrada como en la salida.

Los datos técnicos y las especificaciones de trabajo se encuentran en el anexo 8.

(Robert F. Coughlin, 1999, págs. 44-54)

## **CAPÍTULO 3**

### **DESARROLLO DE HADWARE Y SOFTWARE**

#### **3.1 Introducción**

En este capítulo se desarrollará el diseño del software y la implementación del hardware, el cual se basa en el sensor que adquiere la señal de temperatura y mediante una interface entre la tarjeta FPGA se ingresa la señal hacia el software Labview mediante el cual se implementará el filtro digital.

De igual manera se ejecutará el uso de los conversores ADC y DAC de la tarjeta FPGA y mediante el software Labview se tendrá un control de la adquisición, muestreo y análisis de las señales de entrada como de salida.

Posterior a esto se implementara en el software el diseño de un filtro mediante el cual se permita la eliminación de la señal que contiene ruido y poder recuperar únicamente la señal del sensor de temperatura.

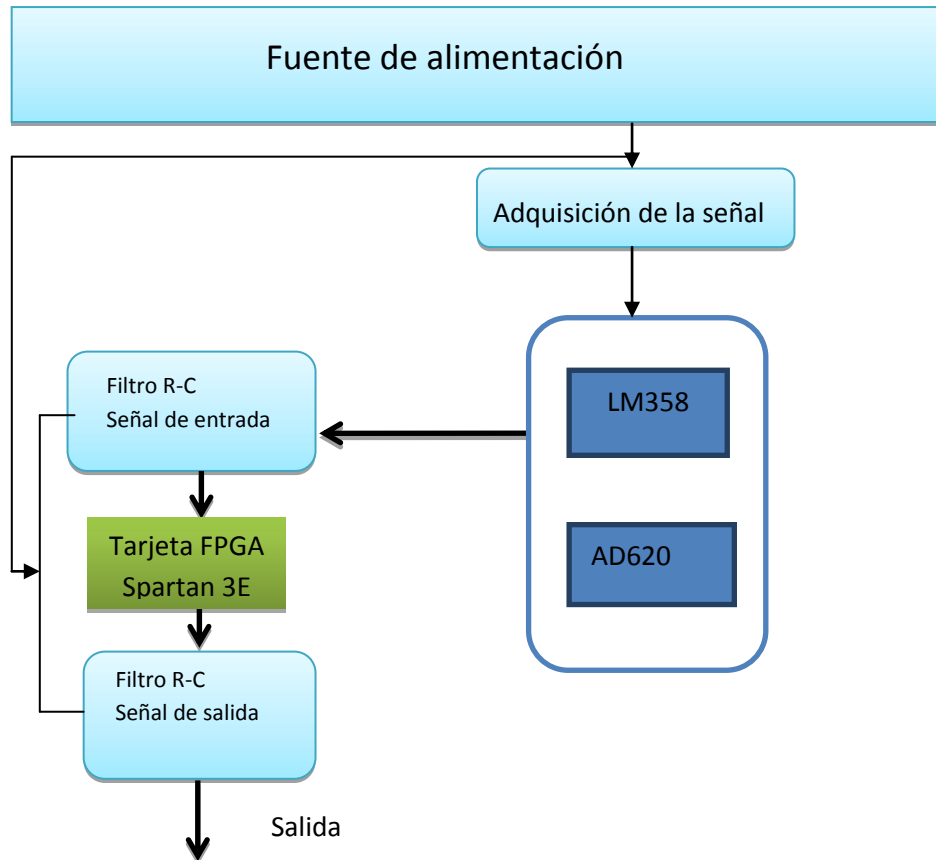
#### **3.2 Diseño de hardware**

A continuación mediante un diagrama de bloques se detalla el circuito electrónico diseñado para la adquisición de la señal de temperatura mediante la termocupla tipo J, este circuito contiene cuatro etapas principales las cuales son la alimentación, adquisición el acondicionamiento de la señal y el filtrado de la señal que va a ingresar a la tarjeta FPGA.

### 3.2.1 Diagrama de bloques

A continuación se modela mediante un diagrama de bloques las etapas que contiene el hardware del circuito electrónico para la adquisición de la señal.

**Figura 27. Diagrama de bloques del hardware del circuito electrónico**



Elaborado por: (Alejandro Castillo, 2013)

### 3.2.2 Descripción de los bloques de hardware

Para lograr una señal con la cual se podrá realizar el procesamiento y el filtrado correspondiente de la señal se requiere generar una señal de ruido, la cual sumada a la señal del sensor se podrá ingresar a la tarjeta de desarrollo FPGA.

#### 3.2.2.1 Fuente de alimentación

Para generar la señal de ruido se utiliza un transformador reductor de 120VAC a 12VAC el cual es alimentado por la red eléctrica pública, como se requiere un valor de voltaje menor que un voltio se realiza un divisor de voltaje previo para obtener este valor, los cálculos están en función de la siguiente fórmula.

$$V_{out} = V_{in} \left( \frac{R_x}{R_T} \right) \quad (3.1)$$

Dónde:

$$V_{in} = 13.7VAC$$

$$R_x = 1K\Omega$$

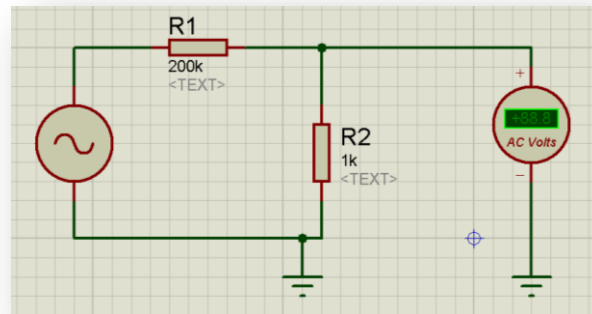
$$R_T = 201K\Omega$$

$$V_{out} = 13.7 \left( \frac{1K\Omega}{201K\Omega} \right)$$

$$V_{out} = 0.068VAC$$

En el siguiente gráfico se puede observar el diseño del circuito para el divisor de voltaje que se implementó para reducir el voltaje del transformador.

**Figura 28. Diseño del circuito divisor de voltaje**



Elaborado por: (Alejandro Castillo, 2013)

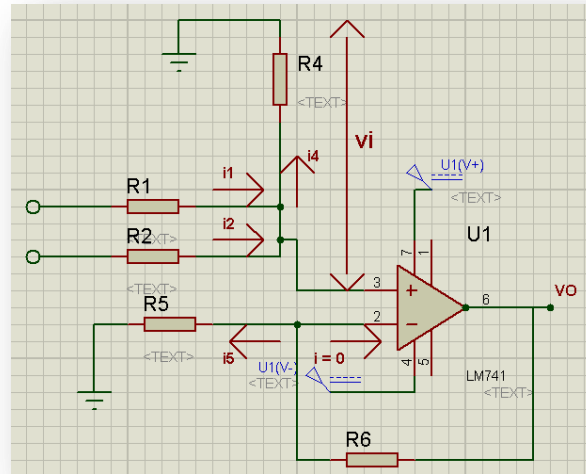
### 3.2.2.2 Adquisición de la señal

Debido a que la señal de ruido que se va a sumar a la señal del sensor proviene de la red eléctrica pública es decir es una onda seno que contiene valores positivos y negativos, se requiere sumarle previamente a esta señal un valor de voltaje mediante el cual se podrá tener la onda sobre cero y así eliminar valores negativos o que se encuentren por debajo de cero, ya que para poder ingresar a la tarjeta FPGA necesariamente se requiere valores sobre cero es decir positivos para que se pueda realizar un correcto muestreo de la señal de entrada.

Para esto se utiliza el amplificador operacional LM741 en una configuración de sumador no inversor a continuación se muestra el diseño del circuito para este operacional el cual se dimensionó de esta forma para poder obtener a la salida un valor en voltaje adecuado para que se pueda sumar luego a la señal del sensor de temperatura.

El cálculo del voltaje de salida se realiza en función del siguiente circuito:

**Figura 29. Diseño del circuito sumador no inversor**



Elaborado por: (Alejandro Castillo, 2013)

$$\Delta V = \frac{V_o}{V_i} = \frac{R_6 + R_5}{R_5} \quad (3.2)$$

Donde la tensión en el terminal ( $V_i$ ) viene dada por:

$$V_i = R_4 \cdot i_4$$

$$V_i = R_4(i_1 + i_2)$$

$$V_i = R_4\left(\frac{V_1}{R_1} + \frac{V_2}{R_2}\right) \quad (3.3)$$

$$V_o = \Delta V \cdot V_i \quad (3.4)$$

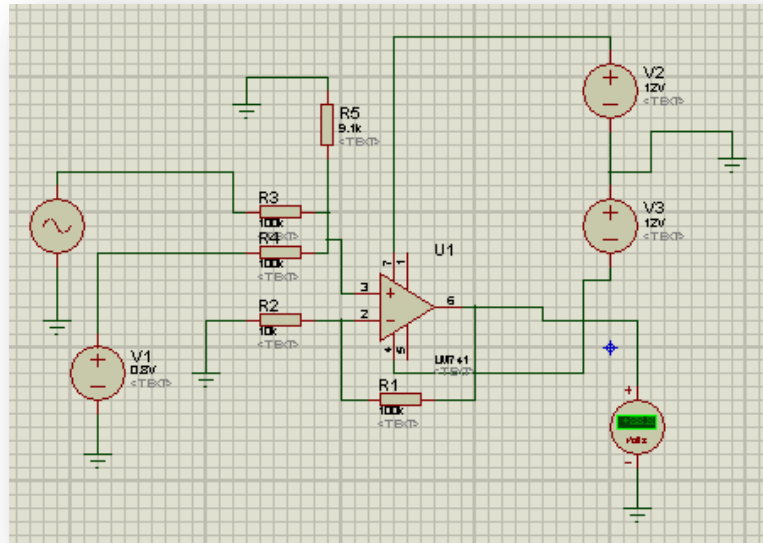
Reemplazando las ecuaciones (3.2) y (3.3) en la ecuación (3.4) se obtiene:

$$V_o = R_4\left(\frac{V_1}{R_1} + \frac{V_2}{R_2}\right)\left(\frac{R_6 + R_5}{R_5}\right) \quad (3.5)$$

Ahora para obtener un valor de voltaje a la salida que se encuentre entre 0.8v y 0.9v máximo se diseñó el siguiente circuito:



**Figura 30. Diseño del circuito sumador no inversor**



Elaborado por :(Alejandro Castillo, 2013)

Reemplazando los valores de los elementos correspondientes en la ecuación (3.4) se tiene:

$$V_o = R_4 \left( \frac{V_1}{R_1} + \frac{V_2}{R_2} \right) \left( \frac{R_6 + R_5}{R_5} \right)$$

Dónde:

$V_1 = 0,068\text{v}$  (voltaje del transformador)

$V_2 = 0,8\text{v}$  (voltaje de referencia para sumar)

$$V_o = 0,009 \left( \frac{0,068}{0,1} + \frac{0,8}{0,1} \right) \left( \frac{0,1 + 0,01}{0,01} \right)$$

$$V_o = 0,859$$

$$V_o \approx 0,86$$

Con esto se tiene ya un valor de voltaje que se encuentra sobre cero y muy adecuado para que al sumarse con la señal del sensor no exceda en el valor

permitido para poder ingresarlo en la tarjeta FPGA y que pueda ser muestreado correctamente.

### 3.2.2.3 Acondicionamiento de la señal del sensor de temperatura

La señal de salida del sensor de temperatura se encuentra en el rango de los micro voltios para lo cual es necesario amplificarla y así poder tener un rango de voltaje adecuado de tal forma que se pueda visualizar en una pantalla, para esto se realiza el acondicionamiento mediante el amplificador de instrumentación AD620 mediante los siguientes cálculos:

$$53\mu\text{V}/^{\circ}\text{C} \quad \text{Sensibilidad de temperatura termocupla}$$

El voltaje que se obtiene es muy pequeño por lo que es necesario amplificarlo al orden de los mili voltios para esto se utiliza la siguiente fórmula para la ganancia que proporciona la hoja de datos del amplificador:

$$G = \frac{49,4k\Omega}{R_G} + 1 \quad (3.6)$$

Dónde:

$$49,4 \text{ K}\Omega \quad \text{Valor de las resistencias internas para la ganancia.}$$

$$R_G \quad \text{Resistencia de la ganancia externa.}$$

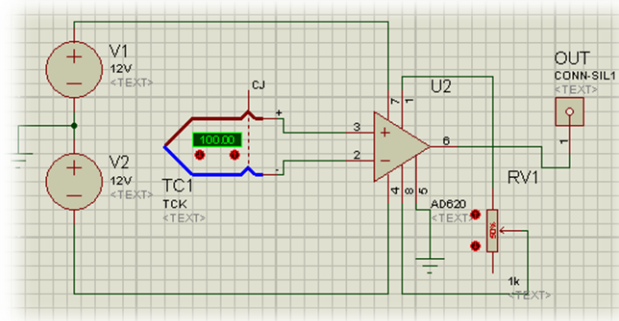
Como se necesita amplificar la señal a mili voltios, se debe tener una ganancia del máximo que ofrece el AD620, es decir de 1000, teniendo como referencia el valor de sensibilidad de voltaje que ofrece el sensor la salida, es decir de  $53\mu\text{V}$ , con lo que al tener una ganancia de 1000 ya se tendrá la señal de salida en el orden de los mil voltios, para esto se escogió una resistencia de  $50\Omega$  ya que con este valor se tendrá una ganancia lo más cercano a la ganancia de 1000:

$$G = \frac{49,4K\Omega}{50\Omega} + 1$$

$$G = 989$$

A continuación se muestra el diseño del circuito amplificador que se implementó para este proyecto.

**Figura 31. Diseño del circuito del amplificador AD620**

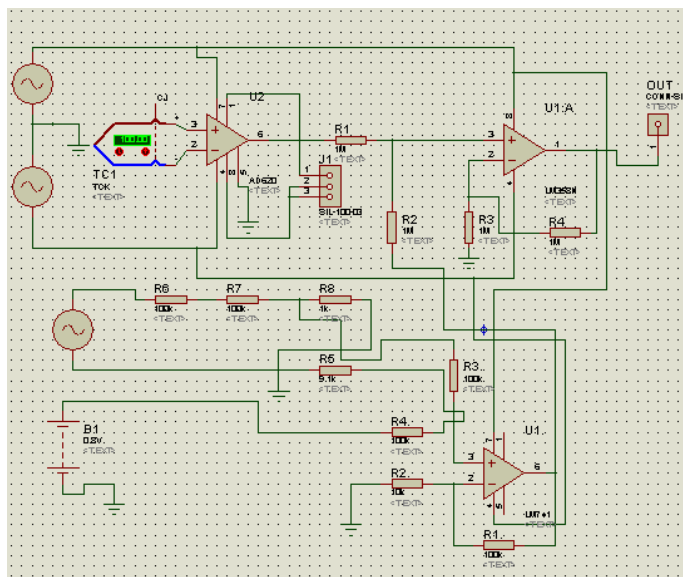


Elaborado por :(Alejandro Castillo, 2013)

La señal proveniente del transformador y la señal del sensor de temperatura, se suman mediante el LM358 para posteriormente ser ingresadas a la tarjeta de desarrollo FPGA.

A continuación se muestra el gráfico del circuito completo que se implementó para el hardware de este proyecto.

**Figura 32. Diseño del circuito de la adquisición de la señal**



Elaborado por: (Alejandro Castillo, 2013)

#### 3.2.2.4 Diseño de filtro r-c

Para impedir los escalones que se forman en la onda de entrada como la de salida, eliminar las frecuencias no deseadas que se encuentran en el rango de los KHz y para atenuar las restantes se implementó dos filtros R-C y se los dimensionó en función de la ecuación (2.4):

$$f_c = \frac{1}{RC}$$

Se toma como dato fijo el valor del condensador y se calcula la resistencia así:

$$C = 0,1 \mu f$$

$$F_c = 1 \text{ KHz}$$

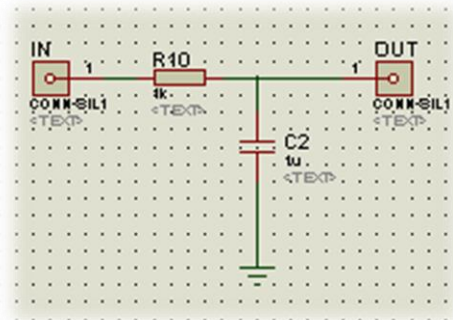
Despejando  $R$  de la ecuación (2.4) se tiene

$$R = \frac{1}{Cf_c}$$

$$R = 1KHz$$

A continuación se muestra el circuito que se implementó para filtrar las señales parasitas de las señales antes de ingresar a la tarjeta FPGA.

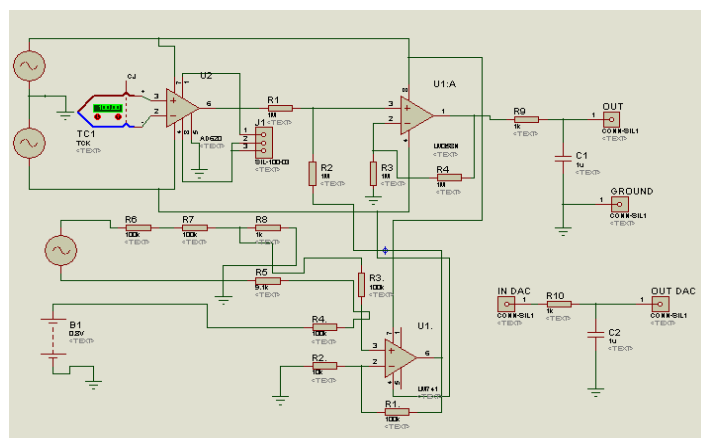
**Figura 33. Diseño del circuito del filtro R-C**



Elaborado por: (Alejandro Castillo, 2013)

Finalmente el circuito que se implementó para generar la señal que se va a ingresar a la tarjeta de desarrollo FPGA, es el que se muestra en la figura 34.

**Figura 34. Diagrama del esquema completo del hardware**



Elaborado por : (Alejandro Castillo, 2013)

### 3.3 Desarrollo del software e implementación del hardware del proyecto

Para realizar la explicación de la implementación del hardware como la del software del presente proyecto, se utiliza diagramas de flujo de cada etapa para tener una idea más clara de lo que se está implementando, a continuación se describe el diagrama de flujo general del proyecto.

#### 3.3.1 Diagrama de flujo general

Con el siguiente diagrama se podrá tener una idea más clara del funcionamiento del software ya que muestra de una manera gráfica la estructura secuencial del desarrollo del software para el presente proyecto, para mayor explicación ver el código completo en el anexo 9.

**Figura 35. Diagrama de flujo del software**



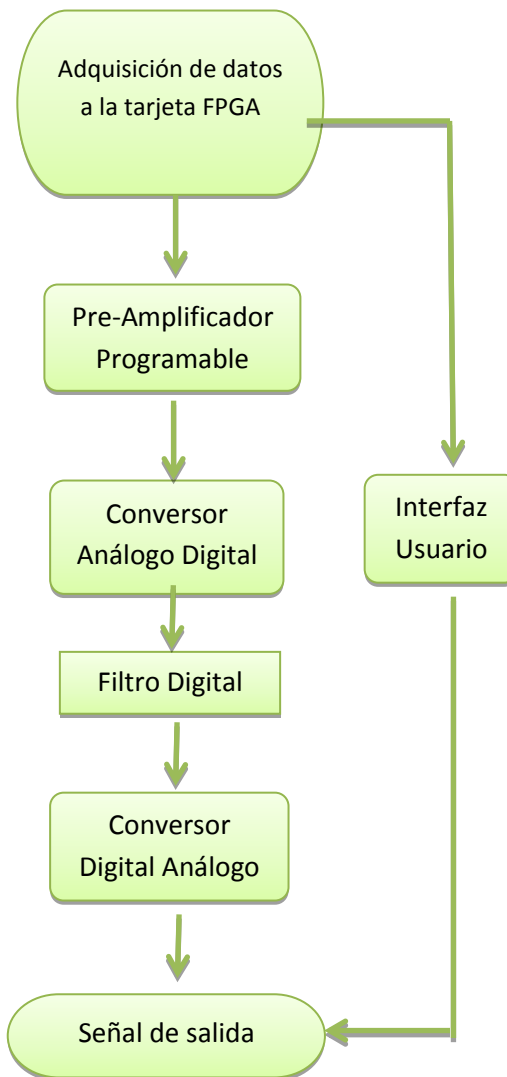
Elaborado por: (Alejandro Castillo, 2013)

- Inicio: Inicio del programa.
- Creación de un nuevo proyecto: se genera todo un paquete completo en el programa en el cual se va a implementar el proyecto.
- Añadir tarjeta FPGA: se ancla al programa la tarjeta FPGA para implementar el programa.
- Instanciar elementos: se escoge todos los elementos que requerimos programarlos para el proyecto y se instancian.
- Implementación del proyecto: se desarrolla el programa que se implementará en la tarjeta FPGA mediante el software Labview.
- Compilación: con esto se puede detectar si el programa completo presenta posibles errores y peligros.
- Fin: Fin del ciclo.
- Anexo 9.

### 3.3.1.1 Desarrollo del software

El desarrollo del software para el presente proyecto se lo describe mediante el siguiente diagrama de flujo:

**Figura 36. Diagrama de flujo del software**



Elaborado por: (Alejandro Castillo, 2013)

- Adquisición de datos de la tarjeta FPGA: el ingreso de la señal se realiza por la cabecera J7.



- Pre amplificador programable: el cálculo para obtener una ganancia adecuada al proyecto se sustenta en las ganancias establecidas por el LTC6912-1 que está incorporado en la tarjeta FPGA
- Conversor análogo digital: la conversión de entradas análogas se realiza mediante el conversor LTC1407A-1 ver anexo 11.
- Filtro digital: para el presente proyecto se implementó un filtro Notch con la finalidad de eliminar la señal de 60 Hz que proviene de la red eléctrica pública a continuación se detalla el cálculo que se realizó para determinar el factor de calidad (Quality) adecuado:

$$Q = \frac{W_O}{AB} \quad (3.7)$$

Dónde:

$W_O$  = Frecuencia de corte central.

$AB$  = Ancho de banda.

$$Q = \frac{60H_z}{65H_z - 55H_z}$$

$$Q = 6$$

Con este factor de calidad se elimina la señal que estén en el orden desde los 55Hz hasta los 65Hz con lo que se garantiza que la señal proveniente de la red eléctrica pública de 60Hz se haya eliminado.

- Conversor Digital Análogo (DAC): en la tabla 5 se muestra los valores de las salidas analógicas para el DAC las cuales se fundamentan en la ecuación (2.9) .
  - Interfaz de usuario: todos los valores de entrada hacia la tarjeta FPGA permiten un control del usuario es así que se puede variar los valores dependiendo la necesidad siempre y cuando se tenga presente las condiciones planteadas para el ingreso de señales por la tarjeta FPGA.

- **Compilación:** La compilación se la realiza desde los sub VIs más internos del programa.
- **Fin:** Fin del ciclo.

### **3.3.2 Implementación del software del proyecto**

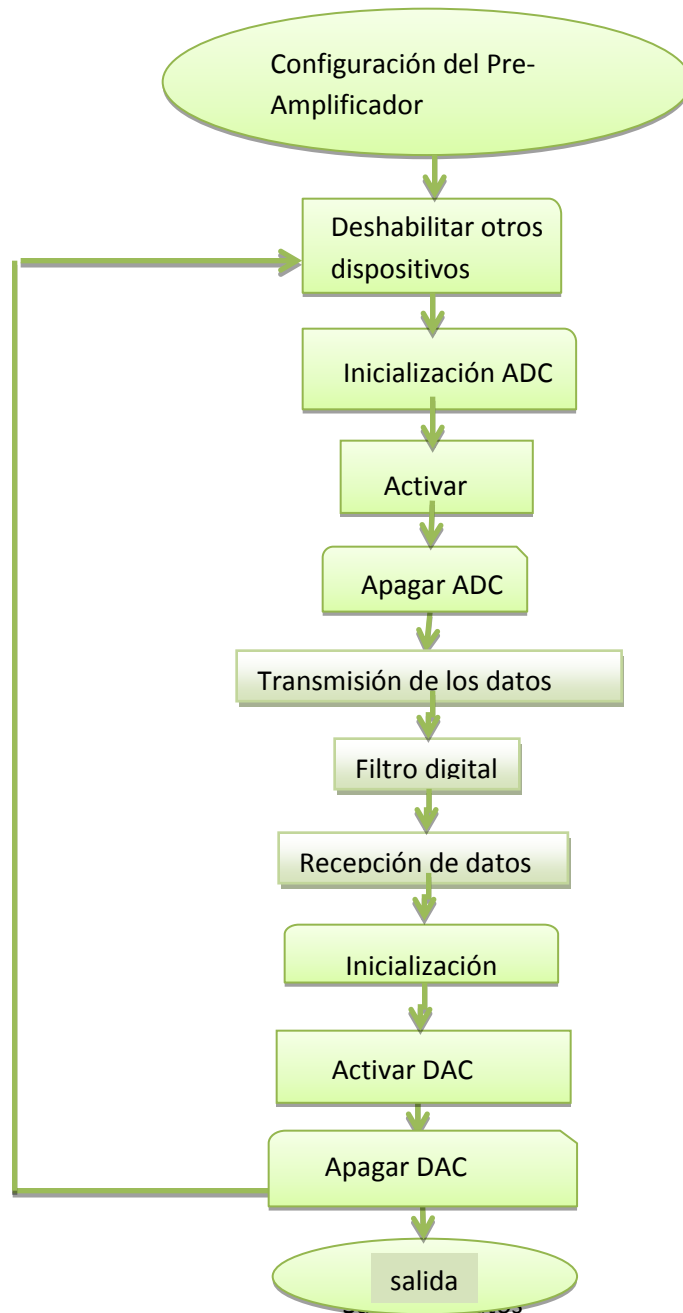
El proyecto se desarrolla mediante el software Labview y se compone de un Sub VI principal, en el cual se realiza la configuración de la tarjeta FPGA en ésta se desarrollará la programación en lenguaje gráfico el cual previo a la compilación será traducido y transferido a la tarjeta FPGA para programarlo, y el otro VI se llama HOST, el cual permitirá el enlace del programa de la tarjeta FPGA con el software Labview.

En cada sub VI se implementó subprocesos internos que realizan diferentes funciones con lo cual al interconectarse forman un sistema completo que permite la inicialización y manejo del ADC y DAC de la tarjeta FPGA mediante el software Labview.

A continuación se detalla el funcionamiento de cada uno de los VIs y de los sub VIs:

### 3.3.2.1 Diagrama de flujo de la configuración de los conversores de la tarjeta fpga y la programación en Labview

Figura 37. Diagrama de flujo de la configuración de los conversores adc y dac de la tarjeta FPGA

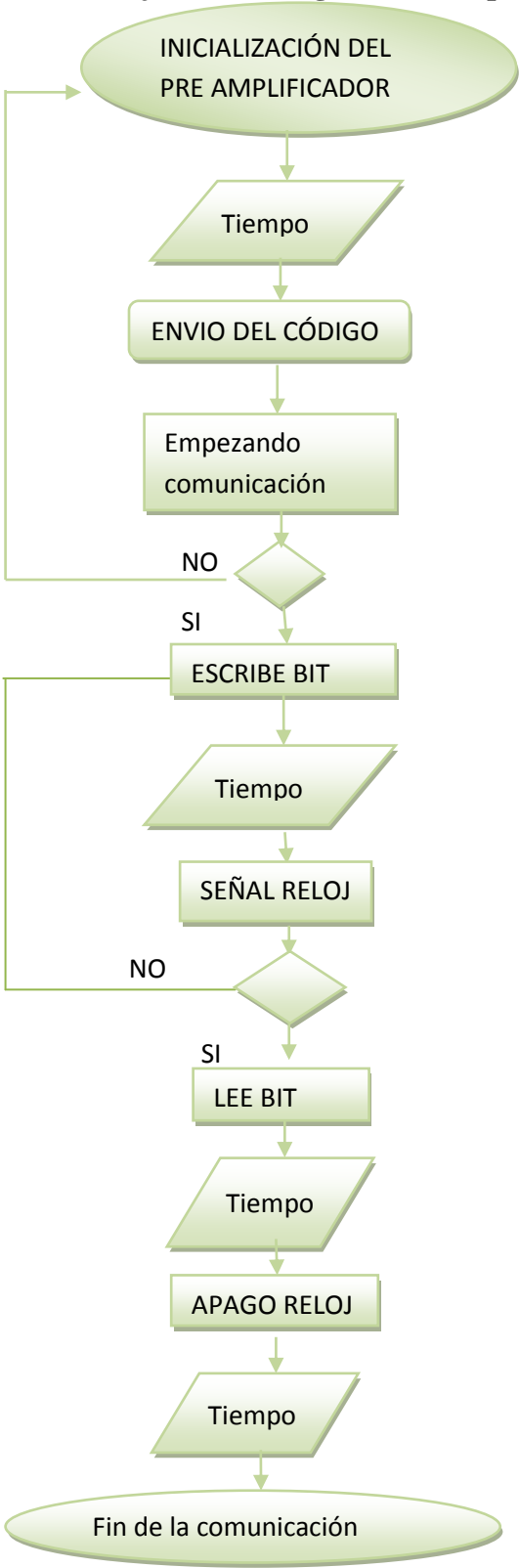


Elaborado por: (Alejandro Castillo, 2013)

### **3.3.2.2 Configuración del pre amplificador de la tarjeta fpga**

La programación del pre amplificador se realiza dentro del sub VI principal, y se lo compila dentro de otro sub VI ya que este elemento requiere para su configuración de diferentes componentes que maneja el módulo de conversión análogo digital, los cuales se comunican mediante el bus de comunicación SPI(Interfaz periférico serial). A continuación se detalla el programa en Labview mediante un flujo grama.

**Figura 38. Diagrama de flujo de la configuración del pre amplificador**



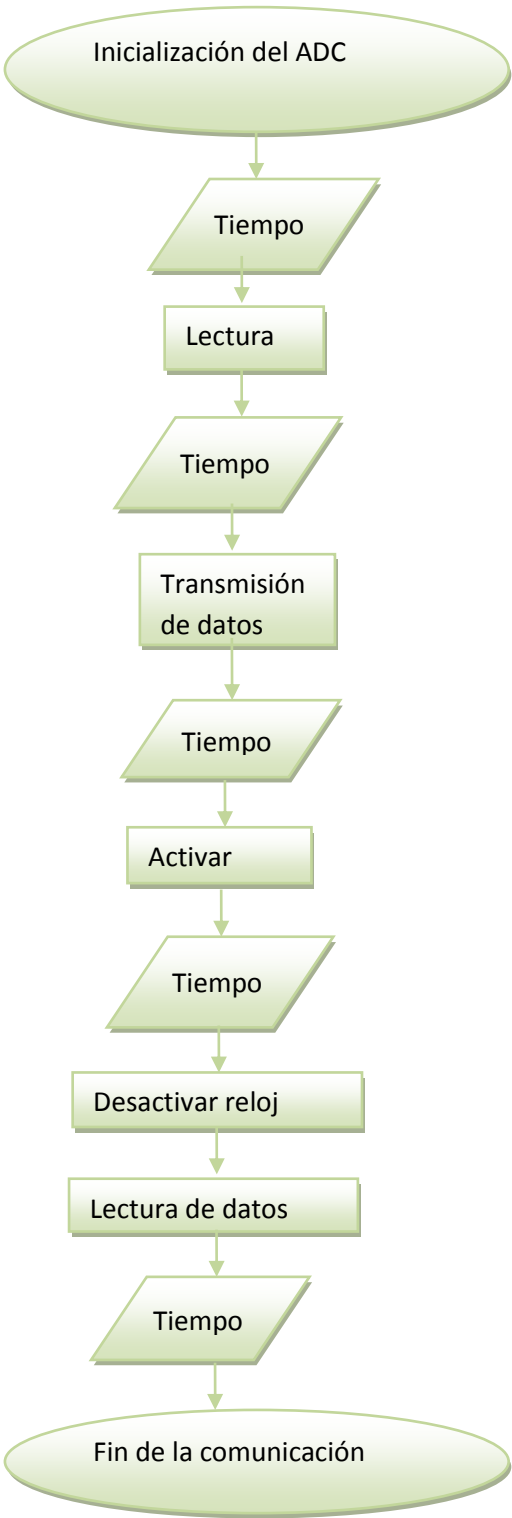
Elaborado por:(Alejandro Castillo, 2013)

- Inicialización del pre amplificador: activación de las banderas según la tabla 2 y tabla 3.
- Tiempo: se instancia un tiempo para realizar el envío de los datos.
- Envío del código: se envía el código desde el HOST principal.
- Empezando comunicación: se realiza la programación de los diferentes elementos necesarios para la comunicación secuencialmente según la figura 16.
- Escribe bit: el bit recibido del comando se escribe en la bandera SPI\_MOSI.
- Tiempo: se instancia un tiempo para la transmisión.
- Lee bit: si la señal de reloj está activada se lee mediante el comando AMP\_DOUT
- Señal de reloj: se apaga la señal de reloj.
- Tiempo: se instancia un tiempo para la transmisión.
- Fin de la comunicación: se finaliza la comunicación y retorna al bucle de inicio. La activación del pre amplificador en la programación se lo realiza una sola vez ya que solo se requiere que se active y el bucle interno realizará la función cíclicamente. A continuación se realiza la programación de los demás componentes necesarios para la adquisición de la señal mediante el ADC.
- Deshabilitar otros dispositivos: se realiza la desactivación de dispositivos que comparten el bus SPI mediante la tabla 6.

### **3.3.2.3 Activación del ADC**

La programación del ADC se ilustra mediante el siguiente diagrama de flujo:

**Figura 39. Diagrama de flujo la activación DEL ADC**



Elaborado por: (Alejandro Castillo, 2013)

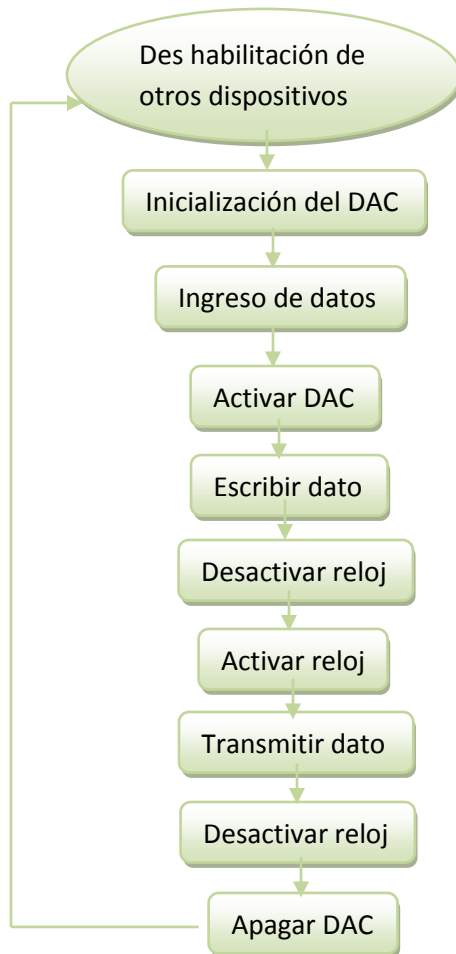
- Inicialización del ADC: activación de la bandera AD\_CONV conjuntamente con la bandera de reloj en bajo.
- Lectura: la lectura se realiza en la bandera AD\_CONV en alto.
- Transmisión de datos: los datos son transmitidos cuando la señal de la bandera AD\_CONV se envía a bajo.
- Activar reloj: la transmisión de los datos se efectúan en el flanco ascendente de la señal de reloj.
- Desactivar reloj: se desactiva la señal de reloj para efectuar el correcto envío de datos y no tener inconvenientes en la comunicación.
- Lectura de datos: los datos son capturados en el flanco descendente de la señal de reloj mediante la bandera SPI\_MISO.
- Fin de la comunicación: se termina la comunicación y los datos ya son enviados por el ADC, para toda la secuencia de programación se instancian los tiempos adecuados para lograr una transmisión correcta.
- Apagar ADC: para continuar secuencialmente con la programación del DAC es necesario desactivar el ADC para no causar error en el envío de datos ya que los dos conversores comparten la comunicación mediante el bus SPI.
- Transmisión de los datos: los datos del ADC se transmiten en función de la figura 17.
- Filtro digital: programación del filtro digital sujeto a la ecuación (3.7).
- Recepción de datos: los datos son recibidos por el DAC en relación de la figura 21.



#### 3.3.2.4 Activación del DAC

La activación del DAC se ilustra mediante el siguiente diagrama de flujo:

**Figura 40. Diagrama de flujo la activación del DAC**



Elaborado por: (Alejandro Castillo, 2013)

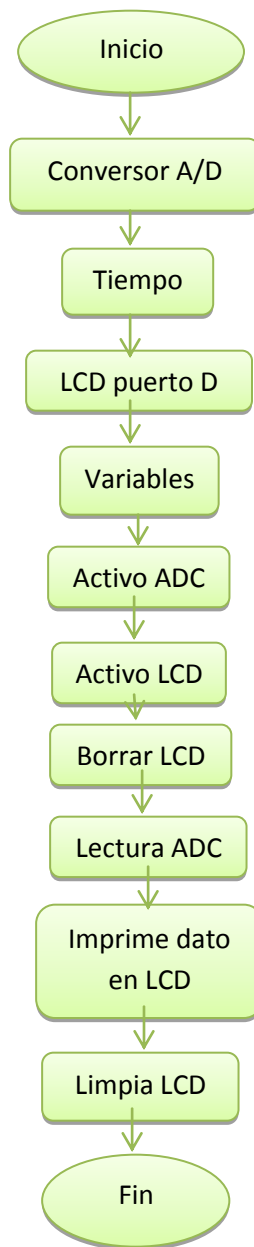
- Des habilitación de otros dispositivos: se desactivan ciertos elementos en referencia de la tabla 10.
- Inicialización del DAC: se activa las banderas del chip de selección del DAC y el reseteo en flanco descendente de la señal de reloj.
- Ingreso de datos: los datos se ingresan mediante un algoritmo previo el cual realiza la conversión del tamaño de la palabra del dato.

- Activar DAC: Se envía la bandera del chip de selección del DAC (DAC\_CS) en bajo y se lo realiza paralelamente al ingreso de los datos.
- Escribir dato: el dato se escribe en la bandera del SPI\_MOSI en flanco ascendente de la señal de reloj.
- Desactivamos reloj: se envía la señal de reloj a estado bajo.
- Activamos reloj: se envía la señal de reloj a estado alto.
- Transmitir dato: el dato se transmite en la bandera SPI\_MISO en flanco ascendente de la señal de reloj.
- Desactivamos reloj: es necesario enviar a estado bajo la señal de reloj una vez transmitido el dato.
- Apagar DAC: se envía la bandera del chip de selección del DAC a estado alto.
- Salida de datos: al finalizar la secuencia se retorna a las condiciones de inicio para la transmisión de la señal en un bucle hasta que se detenga la comunicación.

### **3.3.2.5 Programación del PIC 16F877A**

Mediante el siguiente diagrama de bloques se realiza la explicación del programa que se implementó en el micro controlador.

**Figura 41. Diagrama de flujo la programación PIC 16F877A**



Elaborado por: (Alejandro Castillo, 2013)

- Inicio: se crea la cabecera principal para la programación del PIC.
- Conversor A/D: se Configura el conversor del PIC a 10 bits.
- Tiempo: se configura un tiempo de 20ms.
- LCD puerto D: declaro el puerto D para la entrada del LCD.

- Variables: declaro variables para almacenar los datos de lectura del ADC.
- Activo ADC: activo lectura de datos del ADC en sincronismo del reloj interno del PIC.
- Activo LCD: inicializo el LCD para que imprima los dos datos de lectura del ADC.
- Borra LCD: limpia la pantalla cuatro espacios y sin cursor permanente.
- Lectura datos ADC: lee los datos del puerto Ao y se almacena en la variable dato.
- Imprime dato en LCD: se imprime el dato que ingresa a la variable dato 1 y dato 2 respectivamente que provienen de la señal del sensor y la salida del DAC de la tarjeta FPGA.
- Limpia LCD: se limpia la pantalla de la impresión anterior.
- Fin: se finaliza la programación y retorna al bucle de inicio.
- Anexo 10.

### **3.4 Desarrollo del hardware**

El desarrollo del hardware se realizó para poder realizar las conexiones de las señales que van a ingresar a la tarjeta FPGA y poder tener una interconexión con el osciloscopio.

#### **3.4.1Diseño de la placa**

Se realizó dos placas en las cuales se tiene la captura de la señal que ingresa a la tarjeta de desarrollo FPGA previamente acondicionada, y para mostrar en un LCD la relación de temperatura del sensor vs la temperatura que se obtiene a la salida del DAC de la tarjeta de desarrollo FPGA.

**Figura 42. Placa para adquirir la señal del sensor**

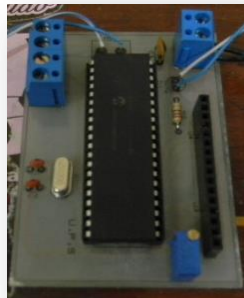


Fuente: Captura de la placa física.

Para realizar las mediciones correspondientes se implementó puntos de prueba los cuales corresponden a la señal medida del sensor en el punto J(13), para medir la señal sumada del transformador y el sensor en el punto J(8), y para medir la señal de salida que se tiene en el panel frontal de la caja donde se montó el proyecto se realiza en la salida del DAC.

Para poder mostrar la temperatura del sensor vs la temperatura que se obtiene a la salida se usó el PIC16F877A. El código de la programación que se realizó se muestra en el anexo 10. La placa se muestra en el siguiente gráfico.

**Figura 43. Placa para mostrar temperatura**

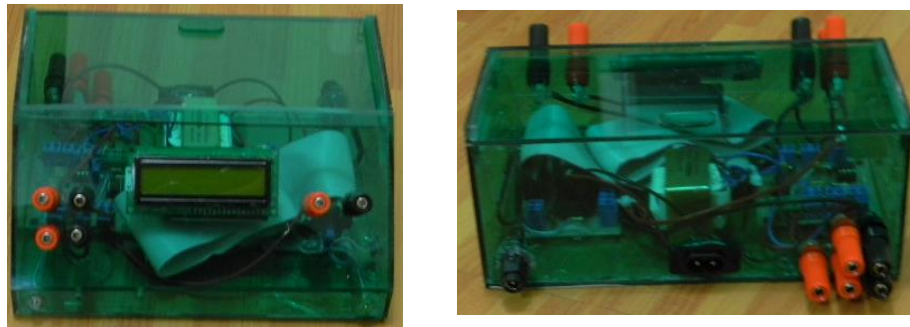


Fuente: Captura de la placa física

### 3.4.2Diseño de estructura

El diseño de la estructura para los circuitos se lo realizó en acrílico con 20 orificios para poder insertar los conectores de los diferentes puntos de conexión del proyecto, cuenta con una cubierta superior removible que permite tener fácil acceso a la circuitería si llegara a existir algún inconveniente en los elementos de las placas.

**Figura 44. Vistas de la caja donde se montó el proyecto final**



Fuente: Captura de la caja física.

Se diseñó una base en acero inoxidable para tener mayor firmeza en el sensor y la llama, como se muestra en la figura 45

**Figura 45. Base para sensor y llama**



Fuente: Captura de la base física.

### **3.5 Costos del proyecto de investigación**

#### **3.5.1 Costos de desarrollo matemático**

Los costos del desarrollo matemático para realizar el acondicionamiento de las señale que ingresaran a la tarjeta FPGA se detallan a continuación:

**Tabla 11. Costos del desarrollo matemático**

DESCRIPCIÓN	COSTO
Señal del sensor de temperatura	\$30,00
Señal del ruido eléctrico	\$30,00
Filtro pasivo pasa bajas	\$20,00
<b>TOTAL</b>	<b>\$80,00</b>

Elaborado por: (Alejandro Castillo, 2013)

### 3.5.2 Costos del desarrollo del software

Los costos de desarrollo del software del proyecto elaborado en lenguaje gráfico Labview para tener una interfaz con la tarjeta de desarrollo Spartan 3E de Xilinx, para el procesamiento de señales digitales, se detallan a continuación:

**Tabla 12 Costos de desarrollo de software**

DESCRIPCIÓN	COSTO
Módulo pre amplificador	\$ 30,00
Módulo ADC	\$ 30,00
Módulo filtro digital	\$ 30,00
Módulo DAC	\$ 30,00
Módulo interfaz Labview-FPGA	\$ 30,00
Módulo LCD	\$ 30,00
<b>TOTAL</b>	<b>\$ 180,00</b>

Elaborado por: (Alejandro Castillo, 2013)

### 3.5.3 Costos de desarrollo del hardware

Los costos de desarrollo del hardware del proyecto involucra los diferentes elementos electrónicos que se utilizó para realizar la implementación del circuito final, así también el detalle del diseño de la placa y la estructura; estos rubros se detallan en la siguiente tabla:



**Tabla 13. Costos de desarrollo de hardware**

Descripción	Cantidad.	V. Unitario	Costo
AD620	1	\$ 7,00	\$ 7,00
LM358	1	\$ 0,60	\$ 0,60
LM741	1	\$ 0,40	\$ 0,40
LCD	1	\$ 7,50	\$ 7,50
Termocupla tipo J	2	\$ 38,00	\$ 76,00
Transformador reductor	1	\$ 8,00	\$ 8,00
Banana hembra	11	\$ 0,35	\$ 3,85
Regleta 3H	1	\$ 0,69	\$ 0,69
Led	1	\$ 0,20	\$ 0,20
Led súper BRI	1	\$ 0,25	\$ 0,25
Pulsador	1	\$ 0,19	\$ 0,19
Jack	1	\$ 0,85	\$ 0,85
Plug	1	\$ 0,30	\$ 0,30
Cable ID	1	\$ 2,20	\$ 2,20
Pines para puerto ID	1	\$ 0,20	\$ 0,20
Resistencias 1M $\Omega$	4	\$ 0,02	\$ 0,08
Resistencias 100K $\Omega$	5	\$ 0,02	\$ 0,10
Potenciómetro precisión 100K $\Omega$	2	\$ 0,65	\$ 1,30
Resistencias 1k $\Omega$ 1/4W	3	\$ 0,02	\$ 0,06
Resistencias 10k $\Omega$ 1/4W	1	\$ 0,02	\$ 0,02
Resistencias 9,1k $\Omega$ 1/4W	1	\$ 0,02	\$ 0,02
Resistencias 4,7k $\Omega$ 1/4W	1	\$ 0,02	\$ 0,02
Resistencias 270 $\Omega$ 1/4W	1	\$ 0,02	\$ 0,02
Oscilador de cristal	1	\$ 0,59	\$ 0,59

Descripción	Cantidad.	V. Unitario	Costo
Zócalo 40PW	1	\$ 0,40	\$ 0,40
Zócalo 8PW	3	\$ 0,10	\$ 0,30
condensadores cerámicos 105	2	\$ 0,08	\$ 0,16
CI 16F877A	1	\$ 12,00	\$ 12,00
Borneras 3 pines	1	\$ 0,35	\$ 0,35
Borneras 2 pines	6	\$ 0,28	\$ 1,68
Diseño de placa	2	\$ 10,00	\$ 20,00
Diseño estructura para circuitería	1	\$ 22,00	\$ 22,00
Base del sensor (termocupla)	1	\$ 15,00	\$ 15,00
		Total	\$ 182,33

Elaborado por: (Alejandro Castillo, 2013)

### 3.5.4 Costo total del proyecto

En la siguiente tabla se detalla el resumen de los rubros anteriores calculados es decir del desarrollo matemático, desarrollo del software en Labview para realizar la interfaz con la tarjeta de desarrollo FPGA Spartan 3E y el montaje del hardware:

**Tabla 14. Costo total del proyecto**

DESCRIPCIÓN	COSTO
Diseño de ingeniería	\$ 80,00
Desarrollo de software	\$ 180,00
Desarrollo de hardware	\$ 182,33
<b>Total</b>	<b>\$ 442,33</b>

Elaborado por: (Alejandro Castillo, 2013)

## CAPÍTULO 4

### ANÁLISIS DE RESULTADOS

#### 4.1 Introducción

Para poder obtener un error de la magnitud de temperatura del sensor vs la magnitud que se obtiene a la salida del circuito se obtiene inicialmente la media aritmética de los datos a analizar, para esto se realizó pruebas con 20 rangos de temperaturas diferentes, mediante estos valores se pudo determinar el rango óptimo de trabajo para el correcto muestreo de la señal de entrada y su posterior reconstrucción mediante el DAC de la tarjeta en mención.

**Tabla 15. Temperatura del sensor vs temperatura a la salida**

	Temperatura en °C Sensor	Temperatura en °C Salida
Dato 1	171	174
Dato 2	173	176
Dato 3	210	213
Dato 4	222	216
Dato 5	223	218
Dato 6	224	222
Dato 7	225	224
Dato 8	225	223
Dato 9	226	221
Dato 10	227	226
Dato 11	228	227
Dato 12	229	229
Dato 13	230	230
Dato 14	234	232
Dato 15	237	235
Dato 16	239	237
Dato 17	240	244
Dato 18	241	245
Dato 19	245	242
Dato 20	245	241
<b>Total</b>	4494	4475
<b>Promedio</b>	224,7	222,5

Elaborado por: (Alejandro Castillo, 2013)

En la tabla 15 se puede observar la temperatura que se obtuvo a la salida del conversor digital análogo (DAC) para poder obtener el error individual de cada medición y el error general así como el error absoluto y relativo de cada muestra y el error absoluto y relativo total de las mediciones se obtiene la tabla 16 en donde mediante las siguientes fórmulas se calculara el error absoluto y relativo de cada dato obtenido:

$$\text{Error absoluto} = \text{Valor leído} - \text{Valor verdadero} \quad (4.1)$$

El error relativo representa la calidad de la medida y viene dado por:

$$\text{Error relativo} = \frac{\text{Error absoluto}}{\text{Valor verdadero}} \quad (4.2)$$

**Tabla 16. Porcentaje del valor de error en %**

	Temperatura en °C Sensor	Temperatura en °C Salida	Error Absoluto	Error Relativo en %
Dato 1	171	170	1	0,01
Dato 2	173	171	2	0,01
Dato 3	210	208	2	0,01
Dato 4	222	216	6	0,03
Dato 5	223	218	5	0,02
Dato 6	224	222	2	0,01
Dato 7	225	224	1	0,00
Dato 8	225	223	2	0,01
Dato 9	226	221	5	0,02
Dato 10	227	226	1	0,00
Dato 11	228	227	1	0,00
Dato 12	229	229	0	0,00
Dato 13	230	230	0	0,00
Dato 14	234	232	2	0,01
Dato 15	237	235	2	0,01
Dato 16	239	237	2	0,01
Dato 17	240	238	2	0,01
Dato 18	241	239	2	0,01
Dato 19	245	243	2	0,01
Dato 20	245	241	4	0,02
<b>Total</b>	4494	4450	44	0,20
<b>Promedio</b>	224,7	222,5	2,2	0,01

Elaborado por: (Alejandro Castillo, 2013)

Se puede observar que para cada dato, el porcentaje de error es menos del dos por ciento, a continuación se muestra el error general en porcentaje de la medición de la magnitud de temperatura de salida vs la magnitud de temperatura del sensor.

El error porcentual de la magnitud de temperatura del sensor vs la temperatura a la salida del DAC de la tarjeta FPGA viene dada por la siguiente relación:

$$\begin{array}{lcl} 224,7 & \longrightarrow & 100\% \\ 222,5 & \longrightarrow & X = 99,0.20 \end{array}$$

El error porcentual total de la magnitud de temperatura del sensor vs la magnitud de temperatura a la salida del DAC viene dada por la siguiente relación:

$$\begin{array}{lcl} \text{Referencia de porcentaje máximo} & \longrightarrow & 100\% \\ \text{Referencia de porcentaje del error porcentual} & \longrightarrow & 99,020\% \\ \\ \text{Error porcentual total} & \longrightarrow & 100-99,020 \end{array}$$

$$\text{Error total} = 0.98\%$$

Como se observa el error total de la magnitud de temperatura de salida vs la magnitud de temperatura del sensor es del 0,98% que es un valor bastante exacto.

Para calcular cuan preciso es el instrumento en detectar los valores de temperatura se realiza el análisis en función del porcentaje de repetibilidad ya que la repetibilidad es un sinónimo de precisión.

La repetibilidad viene dada por la siguiente ecuación:

$$R = \frac{\sum (X_i - X)^2}{N} \% \quad (4.2)$$

Dónde:

$X_i$  = Media de la temperatura en grados centígrados sensor.

$X$  = Media de la temperatura en grados centígrados a la salida del DAC de la FPGA.

$N$  = Número de datos tomados.

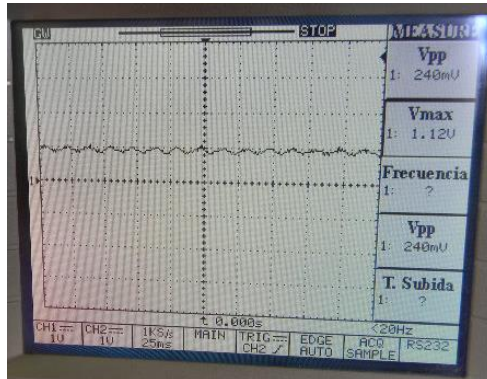
$$R = \frac{(224,7 - 222,5)^2}{20}$$

$$R = 0,242\%$$

Se obtiene como resultado un valor de repetibilidad de  $R = 0,242\%$  que es un valor bastante bajo por lo que significa que el instrumento tiene una precisión bastante alta.

## 4.2 Diagramas de prueba

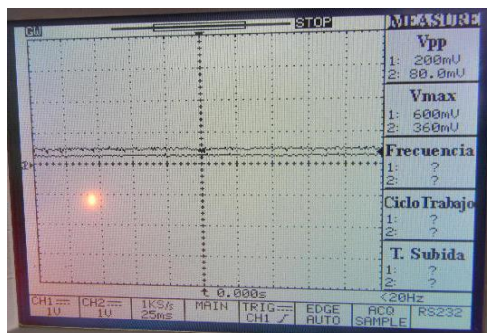
**Figura 46. Señal sensor sumada señal de ruido**



Fuente: Captura del osciloscopio

### a) Primera captura de valores

**Figura 47. Señal del sensor canal1  
Señal de salida canal 2**



Fuente: Captura del osciloscopio

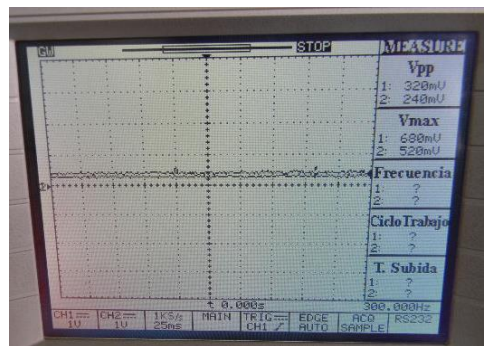
**Figura 48. Temperatura sensor vs temperatura de salida**



Fuente: Captura de temperaturas del LCD.

**b) Segunda captura de valores**

**Figura49. Señal del sensor canal1 señal de salida canal 2**



Fuente: Captura del osciloscopio.

**Figura50. Temperatura sensor vs temperatura de salida**

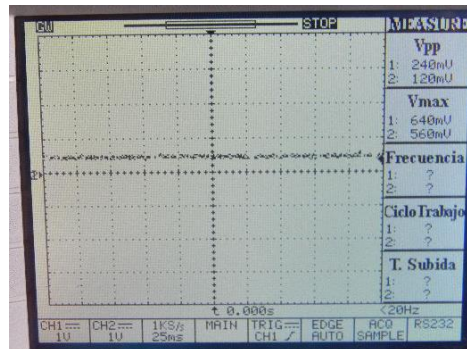


Fuente: Captura de temperaturas del LCD.



c) Tercera captura de valores

Figura 51. Señal del sensor canal1 señal de salida canal 2



Fuente: Captura del osciloscopio.

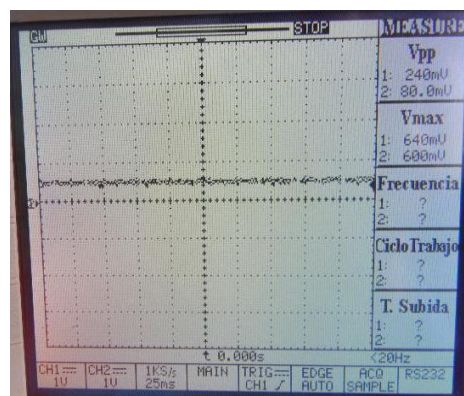
Figura 52. Temperatura sensor vs temperatura de salida



Fuente: Captura de temperaturas del LCD.

d) Cuarta captura de valores

Figura 53. Señal del sensor canal1 señal de salida canal 2



Fuente: Captura del osciloscopio

**Figura 54. Temperatura sensor vs temperatura de salida**



Fuente: Captura de temperaturas del LCD.

## Conclusiones

- Se cumplió con el objetivo “Diseñar, desarrollar e implementar filtros digitales para eliminar ruido de la red eléctrica en un circuito electrónico, utilizando la FPGA Spartan 3E y el software Labview ”
- Los módulos de la tarjeta FPGA Spartan 3E y demás dispositivos que se pueden controlar e implementar desde el software Labview no pueden ser configurados en su estructura interna ya que esto únicamente se lo puede realizar desde VHDL o de otro lenguaje de programación que permita tener acceso a las configuraciones internas de cada elemento.
- Para realizar la conversión del ADC – DAC se debe sincronizar los 14 bits que envía el ADC con los 12 bits que recibe el DAC, para que pueda existir la conversión de señales discretas a señales continuas.
- Para realizar el sincronismo de la señales al transmitir los datos, se utiliza un solo módulo para generar la señal de reloj, el cual se programa dependiendo de la trama donde se lo requiera para realizar la transmisión de datos.
- La programación en el software Labview se lo realizó en forma de una máquina de estado, ya que mediante esta programación se logra tener un control de los elementos anteriores a cada secuencia del programa.
- Los conversores ADC – DAC de la tarjeta de desarrollo FPGA Spartan 3E se comunican mediante el bus SPI (interfaz periférica serial), por lo que se debe activar un conversor a la vez para poder obtener una correcta conversión de datos.
- Debido a que la señal que se ingresaba a la tarjeta de desarrollo FPGA Spartan 3E está en el orden de los mili voltios, se realizó una amplificación adecuada para que el conversor de la tarjeta DAC pueda realizar la reconstrucción de la señal.
- Debido a que la señal que genera el ruido tiene valores positivos y negativos es necesario realizar un offset de esta señal para lograr obtener

una señal sin valores negativos y así garantizar que la señal que va a ingresar a la tarjeta FPGA no será de valores negativos.

- Para poder lograr un correcto funcionamiento del proyecto, el sensor de temperatura debe alcanzar una temperatura en un rango de 200°C a 250°C con lo cual la señal de temperatura a la salida será muy semejante a la señal de temperatura a la entrada.

## **Recomendaciones**

- Se recomienda en el diseño del filtro usar un software específico para el cálculo del coeficiente  $Q$  del filtro.
- Se recomienda que para la configuración de los coeficientes del filtro se lo realice por ancho de banda ya que mediante esta configuración se puede especificar de mejor manera la frecuencia central y los cortes del filtro respectivamente.
- Se recomienda conectar siempre una referencia de tierra del sensor de temperatura para no tener mucho ruido en la señal.
- Se recomienda realizar nuevos proyectos en donde se pueda realizar una interfaz Labview-FPGA utilizando el software Labview 2011 ya que para el presente proyecto se utilizó la versión anterior.
- Se recomienda implementar este sistema de eliminación de ruido de 60 hertz sobre un procesador DSP o sobre un micro controlador ARM.

## LISTA DE REFERENCIAS

### Libros:

Alan V. Oppenheim, A. S. (1998). *SEÑALES Y SISTEMAS*. México, D.F: Pearson.

Robert F. Coughlin, F. F. (1999). *Amplificadores Operacionales y Circuitos Integrados Lineales*. México D.F: Prentice Hall.

Lucelly Reyes, Introducción Filtros, Notas de clase, España-Madrid, 2003, Págs 1-9.

José Luis Oropeza Rodríguez, Técnicas de diseño de filtros IIR, Centro de innovación y Desarrollo Tecnológico en Computación, Mexico D.F, 2006, Págs 1-36.

Worldwide Technical Support and Product Information, Introducción a Labview, National Instruments Corporate Headquarters, USA-TEXAS, 2003, Págs 1-89.

C. Heiner, R. Alejandro, J. Jaime, C. Diana, Diseño e Implementación De Un Filtro Digital Pasa-Bajos En FPGA Utilizando VHDL Para El Laboratorio De Comunicaciones Del Instituto Tecnológico De Soledad Atlántico-ITSA, GICETI, Colombia-Soledad, 2005, Págs 1-2.

Dzul May Yarely Sac Nic Teh, Olmedo Garcia Mário, Circuitos Electrónicos Para Propósito De Control y Monitoreo De Temperatura, via PC, Tesis de Grado, Valle de Mexico, 2007, Págs 72-92.

SOLIMAN, SAMIR S., SRINATH, MADYAM D., *Señales y Sistemas Continuos y Discretos*, Prentice Hall, España-Madrid, 1999, Págs 480-490.

Ignacio Moreno Velasco, Pedro L. Sánchez Ortega, INTRODUCCION A LA INSTRUMENTACION VIRTUAL Programación en Labview, National Instruments Corporate, USA-TEXAS, 2001, Págs 1-40.

OPPENHEIM, Alan V. y MILSKY, Señales y Sistemas, Alan S., Editorial Pearson Education, Segunda Edición, Pag 38.

Antonio Creus Sole, INSTRUMENTACIÓN INDUSTRIAL, Alfaomega marcombo, 7ma edición, Págs 1-5.

## Netgrafía

Christopher Felipe León Galarza, “DISEÑO E IMPLEMENTACIÓN DEL CONTROL PID DE VELOCIDAD DE UN MOTOR DC UTILIZANDO LA TARJETA FPGA SPARTAN 3E Y LABVIEW PARA LOS LABORATORIOS DE LA CARRERA DE INGENIERÍA ELECTRÓNICA DE LA UNIVERSIDAD POLITECNICA SALESIANA”, Recuperado el 20 de octubre del 2012 de: <http://dspace.ups.edu.ec/handle/123456789/1480>

European Software Institute. (17 de julio de 2004). *Wikipedia*. Recuperado el 19 de octubre de 2012, de <http://www.google.com.ec/Intoducciondefiltrospasivos/iec>

ENCICLOPEDIA, “*Filtros digitales FIR (Finite Impulse Response)*”, Recuperado el 27 de octubre de: <http://www.duiops.net/hifi/enciclopedia/filtro-digital-FIR.htm>

Fundaciòn Wikimedia, Inc. (29 de Julio de 2007). *WIKIPEDIA*. Recuperado el 20 de julio de 2012, de [http://es.wikipedia.org/wiki/Sistema\\_LTI](http://es.wikipedia.org/wiki/Sistema_LTI)

Gutiérrez, E. G. (2 de Noviembre de 2009). *Tema 7 Filtros digitales PDF*. Recuperado el 15 de agosto de 2012, de [www.dtic.upf.edu/~egomez/teaching/sintesi/.../Tema7-FiltrosDigitales.pdf](http://www.dtic.upf.edu/~egomez/teaching/sintesi/.../Tema7-FiltrosDigitales.pdf)

Gabriel Darío Fierro, “IMPLEMENTACIÓN DE LA ETAPA DE RECEPCIÓN DE UN SISTEMA DE COMUNICACIONES UTILIZANDO LA TECNOLOGÍA FPGA” Recuperado el 20 de octubre de: [repositorio.espe.edu.ec/bitstream/21000/2625/1/T-ESPE-029826.pdf](http://repositorio.espe.edu.ec/bitstream/21000/2625/1/T-ESPE-029826.pdf), 2010, Págs 22-40

Inc, W. (19 de Mayo de 2013). *Wikipedia*. Recuperado el 07 de diciembre de 2012, de [http://es.wikipedia.org/wiki/Amplificador\\_de\\_instrumentaci%C3%B3n](http://es.wikipedia.org/wiki/Amplificador_de_instrumentaci%C3%B3n)

Mikel, S. (15 de Enero de 2009). *INTRODUCCION A LABVIEW FPGA NI*. Recuperado el 10 de septiembre de 2012, de [www.ni.com/labview/fpga/esa](http://www.ni.com/labview/fpga/esa)

TERMOKEW S.A DE C.V. (17 de Marzo de 2001). *Termokew*. Recuperado el 12 de diciembre de 2012, de [www.termokew.mx](http://www.termokew.mx)

Wikimedia Inc. (14 de Abril de 2001). *WIKIPEDIA*. Recuperado el 16 de octubre de 2012, de Filtro Notch:  
[http://es.wikipedia.org/wiki/Filtro\\_elimina\\_banda](http://es.wikipedia.org/wiki/Filtro_elimina_banda)

## **Manuales**

Dr. José Pablo Alvarado Moya, Procesamiento Digital De Señales, Notas de clase, Costa Rica, 2011, Págs 153-217

Joseph DiGiovanni, Advanced Labview FPGA Programming Optimizing for Speed and Size, National Instruments, 2005.

NATIONAL INSTRUMENTS CORPORATION. (16 de septiembre de 2003). INTRODUCCIÓN A LABVIEW. México, D.F, México.

XILINX ALL PROGRAMMABLE. (9 de marzo de 2006). Spartan-3E Starter Kit Board User Guide. México, D.F, México.



# ANEXOS

# **ANEXO 1**

## **CARACTERÍSTICAS DE LA FPGA**

# **ANEXO2**

## **DATASHEET LTC6912-1.**

# **ANEXO 3**

**DATASHEET OSCILADOR DE  
CRISTAL DE LA TARJETA FPGA**

# **ANEXO 4**

## **CURVA DE CALIBRACIÓN DE TERMOCUPLA TIPO J Y CARACTERÍSTICAS PRINCIPALES**

## PROPERTIES:

### SENSING ELEMENT

Material: Iron-Constantan Type J Thermocouple

Reference Junction: 0°C.

Sensitivity: 0.053 mV/°C.

Limits of Error: +/- 1.1°C (Special) per  
ANSI-MC96.1-1982

### ELECTRICAL

Insulation Resistance: 10M ohm min. @ 100VDC

Lead wires: 2 each #24 AWG with  
TFE insulation. Wires to form  
a pair 60" long w/ TFE sleeving. Wires and  
sleeving to be strain  
relieved at the probe housing entry point  
with shrink tubing. Identification  
band displaying CL  
part number and "TYPE J" to be located under  
strain relief.

## **MECHANICAL**

Probe dimensions:	See drawing #902625
Probe housing matl: Stainless Steel	300 Series
Potting material: rated)	Epoxy - high temp (260 °C

## **ENVIRONMENTAL**

Temperature rating: continuous	260 °C (500 °F),
Pressure: pressure)	5000 PSI (ambient
Pressure rate of change:	1000 PSI / second
Operating media: synthetic oil	Mineral oil or other inert

Vibration: 20 G's minimum at 10 to 2000 Hz

Shock: Withstands 100 G's minimum sinewave shock of 8 milliseconds duration.

**CARACTERÍSTICAS PRINCIPALES:**

- Sensibilidad 0.053 mV/°C.
- Límite de error +/- 1.1°C.
- Resistencia de aislamiento 10M ohm min a 100VDC.
- Temperatura de trabajo 260°C (500°F) Continuo.
- Presión de trabajo 5000PSI

**LIMITACIONES:**

- No usar en atmosferas sulfurosas mayores a 540°C.
- No usar en temperaturas inferiores a 0°C.
- No usar en temperaturas superiores a los 760°C.



# **ANEXO 5**

## **DATASHEET AD620**

# **ANEXO 6**

## **DATASHET TERMOCUPLA TIPO J**

# **ANEX07**

## **DATASHET LM358**

# **ANEXO 8**

## **DATASHET LM741**

# **ANEXO9**

## **DESARROLLO E IMPLEMENTACIÓN DEL SOFTWARE EN LABVIEW**

## INTRODUCCIÓN

En el siguiente tutorial se detallan los pasos para el desarrollo del software **“DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE FILTROS DIGITALES PARA ELIMINAR RUIDO DE LA RED ELÉCTRICA EN CIRCUITOS ELECTRÓNICOS, UTILIZANDO LA FPGA SPARTAN 3E Y LABVIEW PARA EL LABORATORIO DE PROCESAMIENTO DIGITAL DE SEÑALES”** mediante el cual los lectores del presente proyecto podrán modificarlo para futuros proyectos.

El tutorial se divide en tres partes las cuales se clasifican de la siguiente manera:

- 1.- Creación de un nuevo proyecto.
- 2.- Creación de VI de configuración de la FPGA.
- 3.- Creación del VIHOST.

### Creación de un nuevo proyecto

- 1.- Abrir el software Labview.
- 2.-En la ventana de inicio seleccionar Empty Project con el cual creamos un nuevo proyecto.

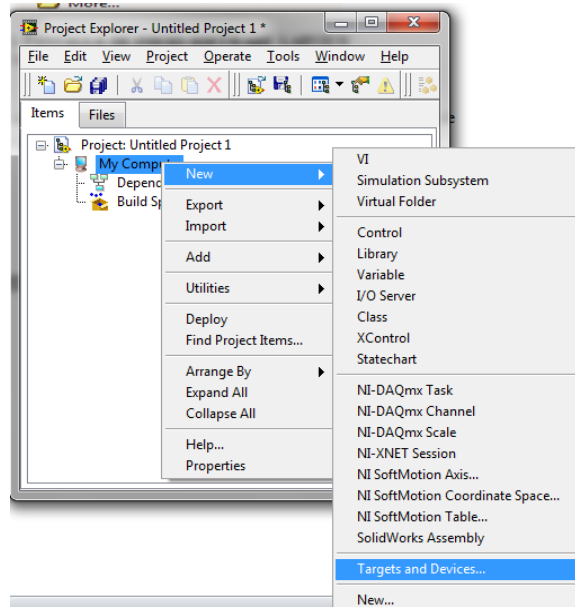
**Figura 1. Pantalla de inicio software labview**



Fuente: Captura de pantalla del software Labview 2010

3.- A continuación se abrirá una ventana donde se da clic derecho sobre el icono de My Computer, para seleccionar new>>targets and devices en donde se selecciona y añade la tarjeta en la cual se va a programar.

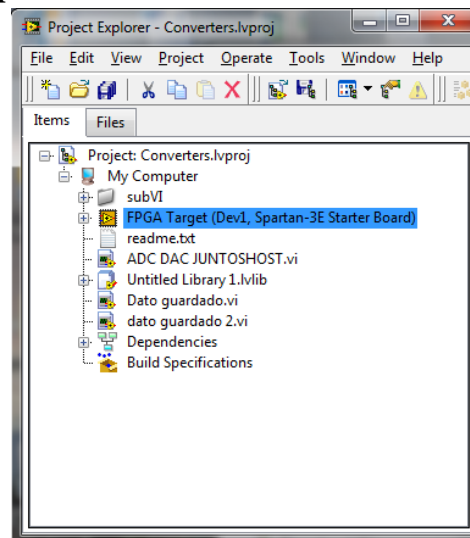
**Figura 2. Selección de hardware**



Fuente: Captura de pantalla del software Labview 2010

4.- Luego de seleccionar la tarjeta ya aparecerá en el explorador de proyectos.

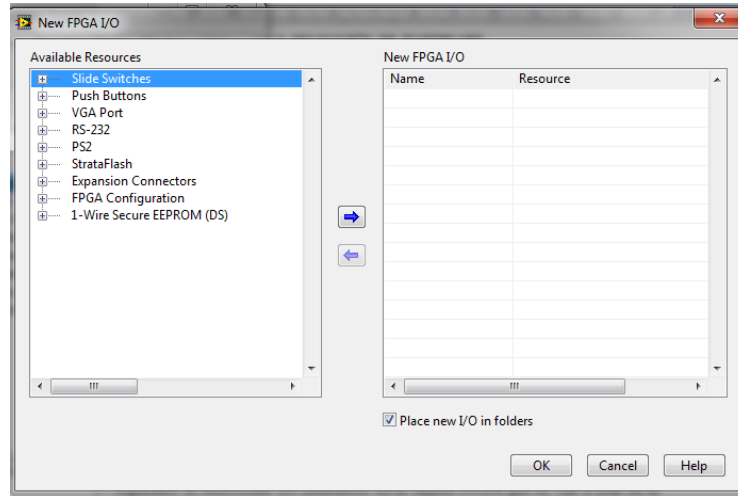
**Figura 3. Project explorer**



Fuente: Captura de pantalla del software Labview 2010

5.- Siguiendo se seleccionan los elementos de la tarjeta FPGA que se van a usar en la programación, dando clic derecho sobre la tarjeta new>>FPGA I/O y se va seleccionando uno a uno los diferentes elementos que se requieren.

**Figura 4. Selección de elementos de la fpga**



Fuente: Captura de pantalla del software Labview 2010

6.- En esta ventana se observa todos los elementos disponibles que se encuentran en la FPGA y se va seleccionando los que se requiera y anclándolos a el proyecto para luego usarlos, para este proyecto será seleccionado el ADC y DAC con todos sus módulos de trabajo referentes al bus SPI.

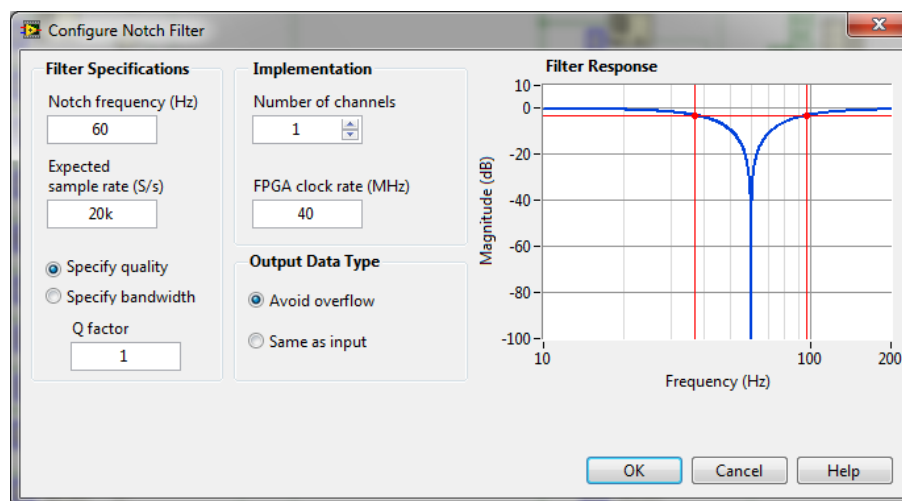
7.- Una vez escogidos todos los elementos necesarios ya aparecerán anclados en el Project explorer.



## Creación de vi de configuración de la fpga

- 1.- En la ventana del explorador del proyecto se da clic derecho sobre la FPGA SPARTAN 3E y se escoge new>>VI y se crea el VI de configuración de la tarjeta.
- 2.- En este VI se realiza la configuración de los conversores y demás elementos que requiere la FPGA para realizar la lectura y escritura de las señales.
- 3.- La programación se realiza en un control de ejecución en secuencia para que se ejecute en condición una luego de la otra, y toda la programación se coloca en un controlador de ejecución whileloop.
- 4.- Tanto la programación del ADC como la del DAC se fundamentan en la guía de usuario de la tarjeta FPGA.
- 5.- Se considera la transmisión de los 14 bits que envía el ADC para configurar a los 12 que recibe el DAC mediante la creación de una nueva trama de datos que cumpla con los bits necesarios para la transmisión.
- 6.- Se configura el filtro notch a condiciones necesarias para eliminar la señal deseada que en este caso es la de 60 Hz.

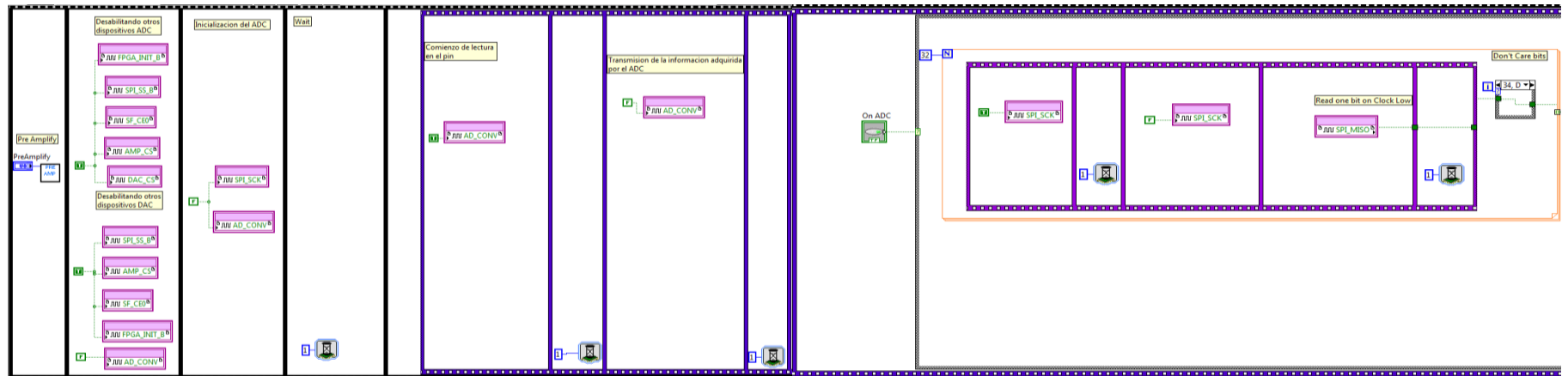
**Figura 5. Selección de elementos de la fpga**



Fuente: Captura de pantalla del software Labview 2010

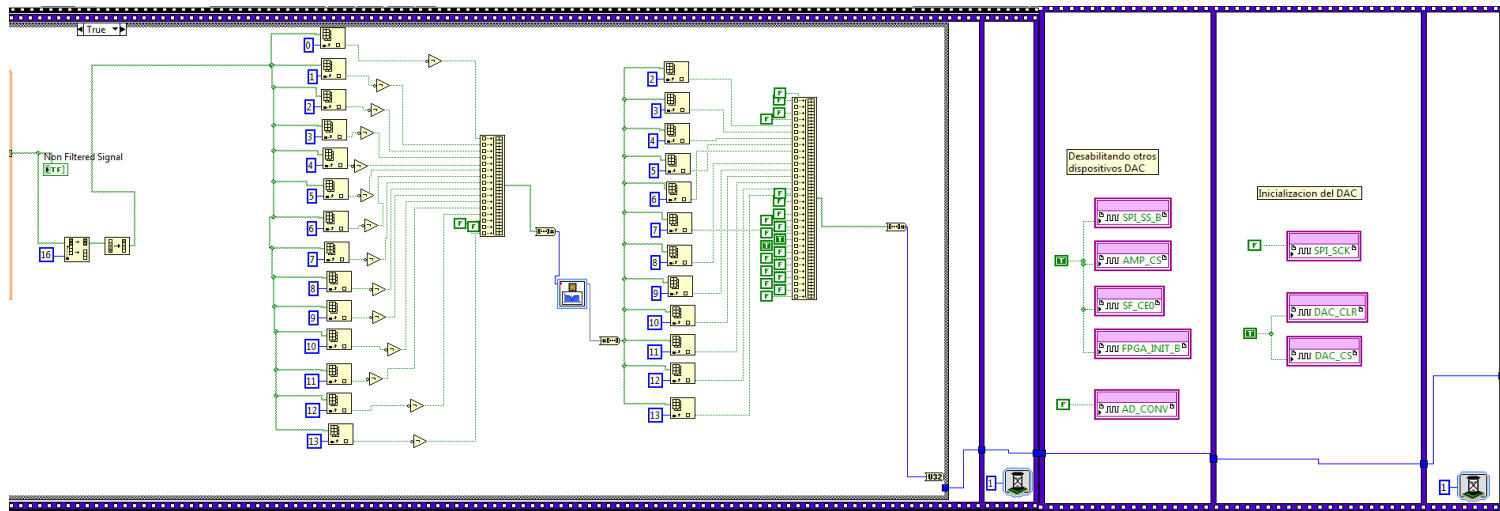
7.- Finalmente la programación se verá similar a la figura siguiente:

**Figura 6. Configuración del vi de programación de la fpga**



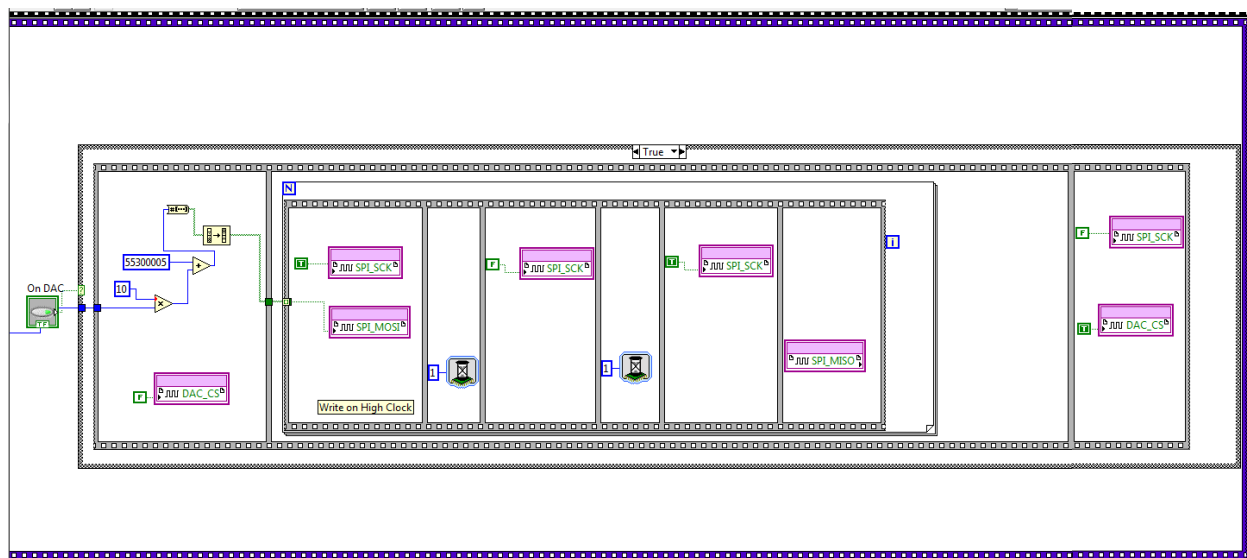
Fuente: Captura de pantalla del software Labview 2010

Figura 7. Configuración del vi de programación de la fpga



Fuente: Captura de pantalla del software Labview 2010

**Figura 8. Configuración del vi de programación de la fpga**

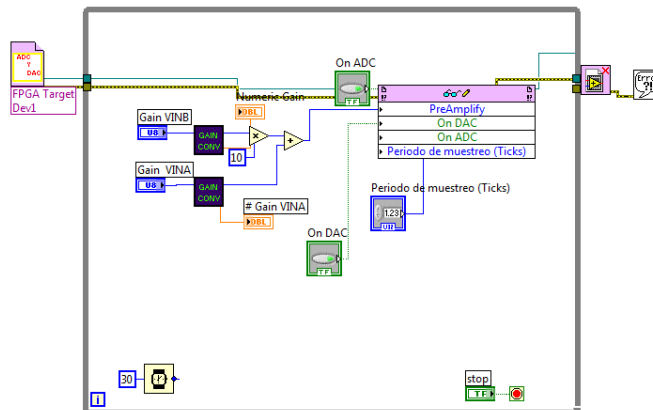


Fuente: Captura de pantalla del software Labview 2010

## Creación del vi host

- 1.- En el explorador del proyecto se da clic derecho sobre My Computer y escogemos new >> VI con el cual creamos el VI HOST.
- 2.- En el diagrama de bloques del nuevo VI se inserta un control de estructura while loop.
- 3.- Dentro de este control de estructura insertamos un control de escritura y lectura para poder controlar las variables del VI de configuración de la FPGA.
- 4.- Desde la paleta de interfaz de la FPGA se inserta un bloque de apertura y de cierre y se coloca a cada extremo en el exterior de la estructura while loop.
- 5.- Se configura también el control de las ganancias para que puedan ser seleccionados por el usuario.
- 6.- Se coloca dos controladores para el encendido y apagado de los conversores ADC y DAC para que pueda ser controlado por el usuario.
- 7.- Se da doble clic sobre el control de interfaz de la FPGA (open) y se selecciona el VI de configuración de la tarjeta FPGA para que este sea cargado como una programación interna del programa.
- 8.- Hecho esto se configura el control de escritura y lectura en el diagrama de bloques del VIHOST con los diferentes elementos que se encuentran en el VI de configuración de la FPGA para que se los pueda controlar desde el VIHOST.
- 9.- Finalmente la configuración del diagrama de bloques del VIHOST se verá como la siguiente figura:

**Figura 9. Configuración del vi host de la fpga**



Fuente: Captura de pantalla del software Labview 2010

# **ANEXO10**

## **PROGRAMACIÓN PIC 16F877A**

## PROGRAMACIÓN PIC 16F877A

```
#include <16f877a.h>

#device adc=10 //configuro conversor ad a 10 bits

#fuses HS,NOWDT

#use delay (clock=20000000)

#include <LCD.C> //trabaja por default en el puerto D

//#use rs232(baud=9600, parity=N, xmit=PIN_C6, rcv=PIN_C7, bits=8)

#use fast_io(B)

#use fast_io(D)

void main(){

int16 dato1=1;

int16 dato2=4;

int16 dato1aux=0;

int16 dato2aux=0;

int16 dato1fin=0;

int16 dato2fin=0;

int16 dato3=0;

int16 cont=0;

int x=0;

set_tris_b(0);

//set_tris_B(255);

setup_adc_ports(all_analog);

setup_adc(adc_clock_internal);

lcd_init();
```

```

output_high(pin_b0);

delay_ms(500);

output_LOW(pin_b0);

dato1fin=dato1;

    dato2fin=dato2;

while(x<=19){

printf(lcd_putc,"\fTMP.SENS= %luoC",dato1fin);  /\flimpia pantalla,%4 4espacios,lu
entero largo sin signo

printf(lcd_putc,"\nTMP. OUT= %luoC",dato2fin);  /\flimpia pantalla,%4 4espacios,lu
entero largo sin signo

delay_ms(1000);

    dato1fin=dato1fin+11;

    dato2fin=dato2fin+11;

x++;

    }

while(1){

set_adc_channel(0);

dato1=read_adc(); // lea el puerto a0 transformelo a digital y almacene en la variable
dato

set_adc_channel(1);

    dato2=read_adc(); // lea el puerto a0 transformelo a digital y almacene en la
variable dato

set_adc_channel(2);

    dato3=read_adc(); // lea el puerto a0 transformelo a digital y almacene en la
variable dato

IF((dato1>=59)&&(dato1<=80)){

    dato1=dato1+165;

```



```

dato2=dato2+165;

dato1aux=dato1-dato2;

dato2aux=dato2-dato1;

dato1fin=dato1;

dato2fin=dato2;

if(dato1aux<=6){

printf(lcd_putc,"\fTMP.SENS= %4luoC",dato1fin);  /\flimpia pantalla,%4 4espacios,lu
entero largo sin signo

//printf("\fTM=%4lu",dato1);

printf(lcd_putc,"\nTMP. OUT= %4luoC",dato2fin);  /\flimpia pantalla,%4 4espacios,lu
entero largo sin signo

}

if(dato2aux<=6){

printf(lcd_putc,"\fTMP.SENS= %4luoC",dato1fin);  /\flimpia pantalla,%4 4espacios,lu
entero largo sin signo

//printf("\fTM=%4lu",dato1);

printf(lcd_putc,"\nTMP. OUT= %4luoC",dato2fin);  /\flimpia pantalla,%4 4espacios,lu
entero largo sin signo

}

}

delay_ms(50);

cont=cont+1;

if (cont>=100){

cont=0;

if ((dato1fin>=2)&&(dato2fin>=2)){

dato1fin=dato1fin-2;

dato2fin=dato2fin-2;

```

```
printf(lcd_putc, "\fTMP.SENS= %4luoC", dato1fin);  /\flimpia pantalla,%4 4espacios,lu  
entero largo sin signo
```

```
//printf("\fTM=%4lu", dato1);
```

```
printf(lcd_putc, "\nTMP. OUT= %4luoC", dato2fin);
```

```
}
```

```
}
```

```
}
```

# **ANEX011**

## **DATASHEET LTC1407A-1 A/D**