



Universidad Autónoma de Querétaro  
Facultad de Ingeniería



***“Metodología de Calibración para sistemas de adquisición de datos en  
plataformas FPGA propietarias”***

**TESIS**

Que como parte de los requisitos para obtener el grado de

**Ingeniero Electromecánico**

Presenta:

**Roberto Iván Gutiérrez Echeverría**

Asesores:

M. en C. Benigno Muñoz Barrón

Dr. Luis Morales Velázquez

San Juan del Río, Qro. julio de 2015

# Resumen

Ya sea durante un proceso de monitoreo o control digital químico, eléctrico, fisiológico o de cualquier otra índole, los sistemas de adquisición de datos toman un papel de alta importancia durante el desarrollo de dichos proyectos debido a que parámetros propios del convertidor analógico digital, tales como su capacidad de resolución, tiempo de muestreo y su correcta calibración, son una parte esencial que determina la fiabilidad del sistema a implementar. El presente trabajo presenta una documentación acerca de los parámetros estáticos y dinámicos más sobresalientes que deben ser determinados para lograr la caracterización de un convertidor (Error de offset, Error de ganancia, Error de no Linealidad Diferencial, Error de no linealidad Integral y Distorsión Armónica Total), así como la metodología que se llevó a cabo para caracterizar los convertidores actualmente en uso por la Universidad, desarrollando para ello un sistema de adquisición de datos genérico (Hardware-Software) mediante lenguaje de descripción de hardware y lenguaje C++, el cual es compatible con las tarjetas FPGA propietarias de la Universidad Autónoma de Querétaro, y el cual tiene la habilidad de integrar fácilmente nuevos sistemas de adquisición de datos (o bloques I/O) basados en FPGA como bloques embebidos dentro de su arquitectura en hardware. Dicho sistema de adquisición de datos incluye una interfaz gráfica de usuario desarrollada mediante la librería gráfica de libre licencia Gtkmm, e incluye las herramientas en software especializadas para realizar completamente la labor de calibración por medio de la interfaz de una forma fácil para un nuevo usuario.

**Palabras Clave:** Calibración, Convertidores, FPGA, Interfaz gráfica, Gtkmm

# Dedicatoria

*“A la memoria de mi padre Roberto Gutiérrez Garduño quién siempre será  
mi mejor amigo y mi ejemplo a seguir,  
A mi madre Herminia Echeverría Soto por ser un símbolo de amor y apoyo  
incondicional y,  
A mi admirable hermana Diana Guadalupe Gutiérrez por ser un símbolo de  
fuerza y perseverancia para mí”*

# Agradecimientos

*A mi familia por brindarme su paciencia, amor sincero, y constante apoyo durante mi desarrollo como persona, además de haber compartido día a día conmigo su sabiduría que han adquirido de la vida.*

*A todos mis amigos que durante el transcurso de la carrera me apoyaron, pasaron momentos agradables y difíciles a mí lado, y que gracias a ello ahora se han convertido en una extensión de mi familia.*

*A mis asesores el Dr. Luis Morales Velázquez y el Dr. Benigno Muñoz Barrón, por haber compartido parte de su conocimiento conmigo, y que además de haber servido de guía para poder culminar este trabajo, han creado una sincera admiración hacia sus personas.*

*A la Universidad Autónoma de Querétaro por haberme brindado tanto los materiales, como la instrucción apropiada durante el transcurso de mi carrera.*

*A todas aquellas personas que haya omitido dentro de esta lista, pero que sin los momentos que de compartido con ellas no hubiera podido llegar a ser quién soy ahora, y estar donde estoy hoy en día.*

# Índice General

RESUMEN.....	I
DEDICATORIAS.....	II
AGRADECIMIENTOS .....	III
ÍNDICE .....	IV
ÍNDICE DE TABLAS .....	VII
ÍNDICE DE FIGURAS .....	VII
1. INTRODUCCIÓN .....	1
1.1 ANTECEDENTES.....	3
1.2 DESCRIPCIÓN DEL PROBLEMA .....	6
1.3 OBJETIVOS.....	7
1.3.1 Objetivo General.....	7
1.3.2 Objetivos Particulares.....	7
1.4 JUSTIFICACIÓN.....	8
1.5 PLANTEAMIENTO GENERAL.....	9
2. REVISIÓN DE LITERATURA.....	11
2.1 ESTADO DEL ARTE .....	11
2.2 FPGAs.....	11
2.2.1 Introducción a los FPGAs.....	11
2.2.2 Arquitectura de un FPGA .....	13
2.3 TIPOS DE MEMORIA .....	15
2.3.1 Memoria ROM .....	15
2.3.2 Memoria RAM.....	17
2.4 PROCESO DE ADQUISICIÓN DE DATOS .....	18
2.4.1 Etapas del proceso de adquisición de datos.....	19

2.4.2	Parámetros críticos del proceso de adquisición .....	25
2.5	CONVERTIDORES A/D Y D/A .....	28
2.5.1	Parámetros de Precisión .....	28
2.5.2	Parámetros de Operación .....	33
2.6	ANÁLISIS EN EL DOMINIO DE LA FRECUENCIA.....	33
2.6.1	Transformada Discreta de Fourier.....	34
2.6.2	Empleo correcto de la transformada de Fourier .....	36
2.6.3	Ventanas Espectrales.....	39
2.7	CALIBRACIÓN.....	41
2.7.1	Distorsión armónica total (THD por sus siglas en inglés Total Harmonic Distortion)..	43
2.7.2	Error de Offset .....	45
2.7.3	Error de Ganancia .....	45
2.7.4	Pruebas de linealidad diferencial.....	46
2.7.5	Pruebas de linealidad Integral .....	49
2.8	LIBRERÍA GRÁFICA GTK.....	49
2.8.1	Gtkmm .....	51
3.	METODOLOGÍA .....	52
3.1	IDENTIFICACIÓN DE LAS CARACTERÍSTICAS DE CADA TARJETA FPGA Y SUS CONVERTIDORES ASOCIADOS .....	53
3.2	DESCRIPCIÓN EN HARDWARE .....	54
3.2.1	Comunicación Interna.....	54
3.2.2	Comunicación Externa.....	59
3.2.3	Máquina de estados finitos para el control de entradas y salidas.....	60
3.3	DESARROLLO DE UNA INTERFAZ GRÁFICA DE USUARIO.....	61
3.3.1	Menú general de configuración .....	62
3.3.2	Área gráfica .....	62
3.3.3	Menú de funciones .....	63

3.3.4	Menú de configuración gráfica .....	65
3.3.5	Pestaña de operaciones aritméticas .....	66
3.3.6	Menú de calibración del convertidor .....	67
3.4	METODOLOGÍA DE CALIBRACIÓN .....	69
3.4.1	Error de Offset .....	70
3.4.2	Error de Ganancia .....	71
3.4.3	Pruebas de linealidad diferencial.....	71
3.4.4	Total Harmonic Distorsion .....	72
3.5	DESARROLLO DE UN MANUAL PARA EL USO DE LA INTERFAZ GRÁFICA.....	73
4.	RESULTADOS Y DISCUSIÓN.....	74
4.1	DETERMINACIÓN DEL ERROR DE OFFSET .....	74
4.2	DETERMINACIÓN DEL ERROR DE GANANCIA.....	75
4.3	PRUEBAS DE LINEALIDAD DIFERENCIAL .....	76
4.4	PRUEBAS DE LINEALIDAD INTEGRAL .....	77
4.5	PRUEBA DE DISTORSIÓN ARMÓNICA TOTAL .....	78
5.	BIBLIOGRAFÍA .....	80
6.	APENDICE.....	83
6.1	MANUAL DE OPERACIÓN DEL SOFTWARE.....	83
6.1.1	Introducción .....	83
6.1.2	Uso de la Interfaz gráfica.....	83
6.1.2.1	Consola de Aplicación .....	84
6.1.2.2	Menú general de configuración.....	85
6.1.2.3	Menú general de funciones.....	87
6.1.2.4	Menú de Configuración Gráfica .....	89
6.1.2.5	Menú de operaciones aritméticas .....	93
6.1.2.6	Menú de calibración de datos .....	93

## Índice de tablas

TABLA 2-1 CARACTERÍSTICAS DE LAS MEMORIAS RAM .....	18
TABLA 2-2 VENTANAS ESPECTRALES MÁS COMUNES.....	39
TABLA 2-3 CONSTANTES DE CONFIABILIDAD MÁS COMUNES (HTTP://WWW.ATX7006.COM/, 2015) .....	49
TABLA 3-1 CARACTERÍSTICAS DE CONVERTIDORES GRUPO HSP DIGITAL.....	53
TABLA 6-1 CARACTERÍSTICAS DE RESOLUCIÓN.....	88

## Índice de Figuras

FIGURA 1-1 PROCESO DE CALIBRACIÓN Y ADQUISICIÓN DE DATOS	9
FIGURA 1-2 DIAGRAMA SIMPLIFICADO DEL PROCESO DE CALIBRACIÓN	10
FIGURA 2-1 ARQUITECTURA GENÉRICA DE UN FPGA (ZEIDMAN, 2006)	14
FIGURA 2-2 ESQUEMA GENERAL DE UN PROCESO DE ADQUISICIÓN DE DATOS	19
FIGURA 2-3 TIPOS DE SEÑALES I	19
FIGURA 2-4 ENTRADA Vs SALIDA DEL TRANSDUCTOR	20
FIGURA 2-5 ENTRADA Vs SALIDA DE LA ETAPA DE FILTRADO	21
FIGURA 2-6 DIAGRAMA A BLOQUES FILTRO ANTIALIASING (OPPENHEIMEN, 1999)	21
FIGURA 2-7 MUESTREO POR MEDIO DE TREN DE PULSOS DIRAC (OPPENHEIMEN, 1999)	22
FIGURA 2-8 EFECTOS DEL MUESTREO EN DOMINIO DEL TIEMPO REPRESENTADOS EN EL DOMINIO DE LA FRECUENCIA (OPPENHEIMEN, 1999)	23
FIGURA 2-9 CIRCUITO RETENEDOR DE ORDEN CERO (OPPENHEIMEN, 1999)	24



FIGURA 2-10 VARIABLES DE UN PROCESO DE ADQUISICIÓN DE DATOS	26
FIGURA 2-11 FUNCIÓN DE TRANSFERENCIA IDEAL DE UN ADC (MIXED,1995)	29
FIGURA 2-12 ERROR DE GANANCIA (MIXED,1995)	30
FIGURA 2-13 ERROR DE OFFSET (MIXED,1995)	30
FIGURA 2-14 DISTORSIÓN ARMÓNICA TOTAL EN EL DOMINIO DEL TIEMPO	31
FIGURA 2-15 ERROR DE NO LINEALIDAD DIFERENCIAL DE UN ADC" (MIXED,1995)	32
FIGURA 2-16 ERROR DE NO LINEALIDAD INTEGRAL (MIXED,1995)	32
FIGURA 2-17 ILUSTRACIÓN DE PERIODO DE MUESTREO Y TIEMPO DE CONVERSIÓN (MIXED,1995)	33
FIGURA 2-18 DIFERENTES APROXIMACIONES DE UNA ONDA CUADRADA PERIÓDICA POR MEDIO DE SERIES DE FOURIER	35
FIGURA 2-19 EFECTO DE UNA SEÑAL INCOHERENTE SOBRE LA DFT	36
FIGURA 2-20 FFT DE UN NÚMERO DE CICLOS CAPTURADOS NO ENTERO (4069 PUNTOS, 127.5 CICLOS) (LUNDBERG, 2002)	37
FIGURA 2-21 FFT DE UN NÚMERO DE CICLOS PAR (4069 PUNTOS, 128 CICLOS) (LUNDBERG, 2002)	37
FIGURA 2-22 FFT DE UN NÚMERO DE CICLOS PRIMO E IMPAR (4069 PUNTOS, 127 CICLOS) (LUNDBERG, 2002)	38
FIGURA 2-23 CONCEPTO DE VENTANEO EN EL TIEMPO	39
FIGURA 2-24 VENTANAS ESPECTRALES EN DOMINIO DEL TIEMPO Y SU EFECTO EN EL DOMINIO DE LA FRECUENCIA	41
FIGURA 2-25 COMPARACIÓN DE ALGORITMOS DE MUESTREO UTILIZADO POR CONVERTIDORES SAR VS PIPELINE VS $\Delta A/D$ (TEXAS INSTRUMENTS, 2011)	42
FIGURA 2-26 DISTORSIÓN ARMÓNICA TOTAL EN EL DOMINIO DE LA FRECUENCIA	43
FIGURA 2-27 "ESPECIFICACIONES DINÁMICAS DE UN CONVERTIDOR (TEXAS INSTRUMENTS, 2011)"	44
FIGURA 2-28 "SEÑALES DE GENERACIÓN DE HISTOGRAMA MÁS COMUNES (YICHUANG, 2008)"	47

FIGURA 2-29 IMAGEN DE MUESTRA DE LAS CARACTERÍSTICAS DISPONIBLES EN GTK+ ( <a href="http://www.gtk.org/">HTTP://WWW.GTK.ORG/</a> , 2014)	51
FIGURA 3-1 DIVISIÓN DEL TRABAJO POR ÁREAS DE DESARROLLO	52
FIGURA 3-2 DIAGRAMA DEL PLAN DE TRABAJO	53
FIGURA 3-3 BLOQUE DE BUS GENÉRICO	54
FIGURA 3-4 BLOQUE DE TIPO ESCLAVO	55
FIGURA 3-5 BLOQUE DE TIPO MAESTRO ESCLAVO	56
FIGURA 3-6 BLOQUE DE TIPO ÁRBITRO	57
FIGURA 3-7 ESQUEMA GENERAL DEL BUS DE ARQUITECTURA ABIERTA	58
FIGURA 3-8 BLOQUE DE COMUNICACIÓN USB	59
FIGURA 3-9 BLOQUE DE LA FSM DE CONTROL DEL ADC	60
FIGURA 3-10 FSM DE ADQUISICIÓN DE DATOS	61
FIGURA 3-11 PANTALLA PRINCIPAL DE LA INTERFAZ GRÁFICA	62
FIGURA 3-12 MENÚ DE FUNCIONES	63
FIGURA 3-13 MENSAJES DE CONSOLA	64
FIGURA 3-14 REPRODUCCIÓN DE ADC EN EL ÁREA GRÁFICA	64
FIGURA 3-15 SEÑAL REPRODUCIDA POR EL DAC	65
FIGURA 3-16 "MENÚ DE CONFIGURACIÓN GRÁFICA"	65
FIGURA 3-17 "PESTAÑA DE OPERACIONES MATEMÁTICAS"	66
FIGURA 3-18 MENÚ DE CONFIGURACIÓN DE LOS CONVERTIDORES	67
FIGURA 3-19 MENÚ DE LA CALIBRACIÓN DE OFFSET	67
FIGURA 3-20 MENÚ DE LA CALIBRACIÓN DE GANANCIA	68
FIGURA 3-21 MENÚ DE THD	68
FIGURA 3-22 MENÚ DE LINEALIDAD	69
FIGURA 3-23 "DIAGRAMA DE FLUJO DE LA OBTENCIÓN DEL ERROR DE OFFSET"	70
FIGURA 3-24 DIAGRAMA DE FLUJO DE LA OBTENCIÓN DEL ERROR DE GANANCIA	71

FIGURA 3-25 DIAGRAMA DE FLUJO DE OBTENCIÓN DEL DNLE	72
FIGURA 3-26 DIAGRAMA DE FLUJO DE OBTENCIÓN DEL ÍNDICE DE THD	73
FIGURA 4-1 ERROR DE OFFSET DE DAS1612	74
FIGURA 4-2 PANTALLA DE CALIBRACIÓN DEL ERROR DE OFFSET	74
FIGURA 4-3 PRUEBA DE GANANCIA DAS1612	75
FIGURA 4-4 PANTALLA DE CALIBRACIÓN DEL ERROR DE GANANCIA	75
FIGURA 4-5 PRUEBA DE DNLE DAS1612	76
FIGURA 4-6 PANTALLA DE CALIBRACIÓN DEL DNLE	77
FIGURA 4-7 PRUEBA DE INLE DAS1612	77
FIGURA 4-8 ANÁLISIS DE THD DE DOS SEÑALES EN LA DAS1612	78
FIGURA 4-9 RESPUESTA EN FRECUENCIA EN DB DE LAS SEÑALES DE PRUEBA DE THD EN LA DAS1612	79
FIGURA 4-10 PANTALLA DE CALIBRACIÓN DE THD	79
FIGURA 6-1 MÓDULOS DE LA INTERFAZ GRÁFICA DESARROLLADA	84
FIGURA 6-2 CONSOLA DE APLICACIÓN	84
FIGURA 6-3 BOTÓN DE DIALOGO DE CARGAR ARCHIVO	85
FIGURA 6-4 DIÁLOGO DE APERTURA DE ARCHIVO	86
FIGURA 6-5 DIÁLOGO DE CARGA EXITOSA	86
FIGURA 6-6 ARCHIVO CARGADO EXITOSAMENTE	86
FIGURA 6-7 SELECCIÓN DE MODO	87
FIGURA 6-8 AJUSTE DEL PERIODO DE MUESTREO	88
FIGURA 6-9 AJUSTE DE LA RESOLUCIÓN DE LAS MUESTRAS	88
FIGURA 6-10 BARRA DE SELECCIÓN DE CANALES	89
FIGURA 6-11 MENÚ DE CONFIGURACIÓN GRÁFICA	89
FIGURA 6-12 ACTIVACIÓN Y DESACTIVACIÓN DEL MALLADO	90
FIGURA 6-13 ACTIVACIÓN Y DESACTIVACIÓN DE LOS EJES PRINCIPALES	90

FIGURA 6-14 MENÚ DE CURSORES	91
FIGURA 6-15 TIPOS DE CURSORES DISPONIBLES	91
FIGURA 6-17 SELECCIÓN DE CURSOR ACTIVO	92
FIGURA 6-16 MEDIDAS DE LOS CURSORES	92
FIGURA 6-18 MENÚ DE OPERACIONES ARITMÉTICAS	93
FIGURA 6-19 BOTÓN DE INICIO DE PROCESO DE CALIBRACIÓN	93

### 1. INTRODUCCIÓN

Hoy en día, la forma en la que se llevan a cabo los procesos de monitoreo y control de producción ha tomado un papel de alta relevancia para la satisfacción de las necesidades humanas de cualquier índole, ya sean necesidades fisiológicas, de comunicación, de educación, de seguridad o de recreación. Y a medida que transcurre el tiempo, la situación económica y social global exige una mayor producción de bienes, los cuales se caracterizan por estar realizados con estándares ascendentes de calidad a un costo inferior para poder ser accesible al mayor público posible

Como respuesta a esta demanda los nuevos métodos y herramientas de desarrollo o producción de algún bien, exigen ser diseñados de una forma más robusta que sus predecesores para solventar las necesidades actuales, y ser al mismo tiempo lo suficientemente dinámicos y flexibles para poder adaptarse y satisfacer desde el desarrollo de una aplicación simple, hasta las necesidades más sofisticadas de un proceso de producción designado.

Ya sea un proceso de monitoreo o control digital químico, eléctrico, fisiológico o de otra condición los sistemas de adquisición de datos toman un papel de alta importancia durante el desarrollo de dichos proyectos debido a que parámetros propios del convertidor analógico digital, como su capacidad de resolución, tiempo de muestreo y su correcta calibración, son una parte esencial que determina la fiabilidad del sistema a implementar.

Como respuesta ante esta necesidad de la sociedad actualmente la Universidad Autónoma de Querétaro (UAQ) destaca a nivel internacional por su eficiencia en la realización de una amplia variedad de proyectos basando sus diseños en arreglos de compuertas programables en campo (FPGA por sus siglas en inglés *Field Programmable Gate Array*), integrando el sistema de adquisición de datos durante el desarrollo de proyectos que conlleven la discretización de una variable analógica proveniente de la naturaleza, y un sistema de posterior procesamiento dentro de la misma plataforma logrando un mayor desempeño que en el caso de realizar ambos procesos por separado.

El proceso de adquisición de datos tiene un rol vital durante una investigación debido a que parámetros propios del convertidor analógico digital, como su capacidad de

resolución, tiempo de muestreo y su correcta calibración son una parte esencial que determina la fiabilidad del sistema a implementar y el estado de dicho convertidor puede validar o invalidar en dado caso los resultados obtenidos por medio del uso de dicho sistema de adquisición de datos. No obstante, se carece de una metodología de calibración de sistemas de adquisición de datos establecida para su uso previo a las investigaciones de la universidad.

Mediante este trabajo se pretende dar un aporte a los proyectos que conlleven un proceso de monitoreo de variables físicas y sean realizados en colaboración de la Universidad Autónoma de Querétaro mediante el desarrollo de una herramienta consolidada que sea adaptable ante las diversas necesidades de nuevos usuarios, ya sean estudiantes de licenciatura, posgrado o docentes investigadores, genérica al tener la habilidad de integrar fácilmente nuevos sistemas de adquisición de datos basados en FPGA, y la cual entregue datos calibrados y ordenados incluyendo una interfaz gráfica de usuario desarrollada en la librería gráfica Gtkmm como complemento.

### 1.1 ANTECEDENTES

Ya sea un proceso de monitoreo o de control digital químico, eléctrico, fisiológico o de otra condición siempre se trabaja con variables físicas provenientes de la naturaleza las cuales requieren ser discretizadas para su posterior procesamiento digital, empleando para esta labor un sistema de adquisición de datos (DAQ) el cual puede ser de índole comercial o propiamente desarrollado.

El uso de los DAQ propios no se encuentran restringidos a aplicaciones específicas, ya que se pueden encontrar muchos ejemplos de estos equipos involucrados en procesos industriales de manufactura tal como sucede con el trabajo de (Muñoz, 2012) quien adquirió datos de temperatura, presión y posición durante un proceso de control de moldeo por inyección, logrando implementar un controlador lógico programable PLC, el procesamiento de un sensor inteligente, un controlador de lógica micro difusa, un asistente de sistema, un procesador IO y un procesador de comunicación, todo en conjunto en un mismo sistema basado en FPGA utilizando la tarjeta DAS1612 propietaria del grupo HSPDigital como el convertidor D/A y A/D durante todo el proyecto.

Existen otros casos similares como el de (Duarte, 2011) quien desarrolló y aplicó un controlador híbrido de iluminación aplicado a un fitotrón mediante técnicas neuronal-difusas y redes de Petri obteniendo datos por medio de la tarjeta UAQ propietaria del grupo HSPDigital, y el trabajo de (Razo, 2011) quien desarrolló su trabajo de control neurodifuso de Humedad para un Fitotrón utilizando como convertidor principal de su sistema de adquisición de datos la anterior citada tarjeta DAS1612 siendo ambos claros ejemplos de que actualmente se realizan distintas investigaciones en las cuales se utiliza como herramienta de adquisición de datos distintas tarjetas electrónicas basada en tecnología FPGA propietarias de la UAQ.

Una vez mencionada la presencia de los FPGA en procesos de manufactura, al igual que en el caso de (Muñoz, 2012) se pueden encontrar diversas aplicaciones de estas en distintos procesos debido a su versatilidad y efectividad, características manifiestas en los resultados obtenidos por (Trejo et al. ,2010), quienes desarrollaron e implementaron de forma exitosa un sensor inteligente basado en FPGA el cual puede ser empleado para mejorar la estimación del área de desgaste del flanco en insertos de una máquina de control numérico (CNC), de uso muy común en la industria de la manufactura, basándose

únicamente en la información recabada por dos sensores primarios, los cuales monitoreaban el suministro de corriente de un servoamplificador y un acelerómetro de tres ejes.

Tal y como se puede observar en base a los ejemplos anteriormente citados el uso de un DAQ basado en FPGA se presenta constantemente en trabajos de análisis de corriente y voltaje de sistemas eléctricos como sucede tanto en el trabajo desarrollado por (Granados et al. ,2014) en el cual se monitorearon datos de corriente y voltaje en conjunto con la velocidad del husillo de una CNC para analizar los efectos en el acabado superficial del maquinado de una pieza y el torque de la CNC causados por diversos niveles de sags eléctricos inducidos en la máquina, logrando encontrar una relación entre los niveles de sags de voltaje inducidos, las variaciones en torque de la máquina y el coeficiente de rugosidad de la pieza fabricada, Y como sucede nuevamente en el trabajo realizado por (Valtierra et al. ,2013), quienes monitorearon señales de corriente y voltaje para obtener información relevante para la solución de problemas de aplicaciones científicas relacionadas al diagnóstico y la calidad de la energía de máquinas eléctricas y sistemas eléctricos y de protección demostrando de esta manera que los trabajos basados en FPGA y el uso de sistemas DAQ se encuentran vigentes y tienen una alta fiabilidad al grado de poder basarse en parámetros derivados de sus mediciones para obtener algún producto final de diversas investigaciones delicadas.

Así mismo, durante el desarrollo de investigaciones pertenecientes a las ciencias naturales se utiliza de forma cotidiana la adquisición de datos en conjunto con el desarrollo de sus respectivas Interfaces gráficas de usuario como base para el desarrollo de sus investigaciones tal y como sucedió con el trabajo realizado sobre cultivos protegidos por (González, 2013), quien automatizó al monitoreo y control de los niveles de temperatura, humedad, nivel de luz y niveles de  $CO_2$ , de la cámara de crecimiento de la facultad de Ingeniería de la UAQ Campus Amazcala, y sucede nuevamente durante el trabajo de (González O., 2013), quien desarrolló un sistema basado en arquitectura híbrida FPGA- $\mu$ C para la medición remota de variables fisiológicas de plantas tales como humedad relativa y temperatura. No obstante, en ninguno de esos trabajos se desarrolló una interfaz gráfica de usuario que pueda ser utilizada posteriormente por un nuevo usuario en conjunto con su sistema de adquisición de datos durante el desarrollo de un proyecto nuevo, siendo acotado



el uso de dichos trabajos a la aplicación particular para la cual se desarrollaron en primer lugar.

Uno de los problemas del proceso de desarrollo de un sistema DAQ en conjunto con su GUI es que el proceso por sí mismo es una labor exhaustiva por lo cual muchos investigadores optan por otras alternativas como el uso de softwares comerciales, lo cual se ha vuelto de uso muy común para llevar a cabo la adquisición de datos debido a que la gran variedad de sistemas ofertados en el mercado realizan un desempeño satisfactorio en los campos de aplicación requeridos, y el hecho de incluir diversas herramientas de post procesamiento muy útiles como la herramienta *Fuzzy control* de Labview por citar alguna, pero inevitablemente conllevando con ello a cierta dependencia tecnológica por parte de los usuarios hacia las empresas que proveen los DAQ y un gasto considerable proveniente del pago de licencias para el uso de su software particular de uso. Ejemplos de estos casos son en el trabajo de (Ramírez, 2013) quien llevó a cabo la adquisición de datos de su proyecto utilizando el software comercial Labview para realizar un sistema no invasivo basado en el monitoreo de temperatura del agua para predecir el crecimiento del pez tilapia implementado con microcontrolador y tecnología ZigBee, y el trabajo de (García, 2012) quien nuevamente utilizó Labview en su proyecto en el cual se realiza la lectura y el estudio de señales mioeléctricas generadas por los músculos, tanto de los brazos como de la palma de la mano para identificar un signo del alfabeto de la lengua de señas mexicana realizando un procesamiento posterior utilizando el software Matlab.

A partir de estos trabajos se puede notar que el uso de sistemas DAQ se encuentra de amplia manera en uso dentro de trabajos de distinta índole, siendo el uso de tecnología FPGA muy común durante el desarrollo de los mismos debido a su versatilidad y sobre todo a la fiabilidad de sus convertidores A/D y D/A asociados, fiabilidad la cual debe estar en un alto índice de confiabilidad y debe de ser cuantificada por medio de los procedimientos típicos de caracterización de un convertidor. Así mismo, en base en los trabajos citados cabe destacar que el desarrollo de una interfaz gráfica es un punto crítico en el desarrollo de sistemas DAQ, tanto que lleva a muchos investigadores a divergir entre la decisión de desarrollar una propia u optar por el empleo de sistemas DAQ comerciales muy completos los cuales ofrecen ciertas ventajas y desventajas ante un sistema DAQ propietario a analizar previamente al desarrollo del trabajo.

### 1.2 DESCRIPCIÓN DEL PROBLEMA

La necesidad de la adquisición de datos analógicos provenientes de procesos reales es una actividad que predomina en diferentes áreas de la ciencia e ingeniería, derivando en que el desarrollo de un sistema de adquisición de datos sea una práctica común que se lleva a cabo como parte del desarrollo de los proyectos de estudiantes, docentes e investigadores de la Universidad Autónoma de Querétaro.

Para ser utilizados, dichos sistemas de adquisición deben ser ajustados, calibrados y configurados para cumplir de manera óptima con las características de funcionalidad requeridas para cada uno de los proyectos. No existe actualmente en el mercado una herramienta consolidada que cumpla con la fácil reconfigurabilidad de parámetros que surja como respuesta a la extensa variedad de proyectos llevados a cabo, y como consecuencia, el proceso de adquisición de datos no se ha optimizado y cada uno de los sistemas realizados hasta el momento requiere de ser modificado de manera significativa desde el diseño en hardware o ser desarrollado nuevamente por cada nuevo usuario para realizar una nueva investigación.

La gran cantidad de parámetros de un proceso de adquisición de datos ha derivado en una amplia oferta de equipos de adquisición de datos en el mercado, los cuales pueden llegar a satisfacer desde las necesidades más básicas de resolución y muestreo por medio de un microcontrolador básico, hasta los sistemas de alta velocidad y precisión siendo ofertados por marcas comerciales como National Instruments, Linear Technology, Texas Instruments entre otras empresas importantes. Sin embargo, el uso de estos productos comerciales implica graves inconvenientes tales como la carencia de reconfigurabilidad de los parámetros de adquisición en diversos dispositivos, la necesidad de utilización de un software específico con alcances limitados para un posterior procesamiento de datos y un costo elevado ligado a actualizaciones de software.

En respuesta a esta problemática, múltiples plataformas FPGA propietarias han sido diseñadas y utilizadas, principalmente en trabajos de posgrado, para resolver el problema de la integración de un sistema de adquisición y un sistema de procesamiento de datos, llegando a desarrollar diversos modelos de una interfaz gráfica de usuario (GUI por sus siglas en inglés *Graphical User Interface*) para lograr una interpretación y manipulación más cómoda y amigable de los datos obtenidos, sin llegar a tener una herramienta gráfica

consolidada, la cual se torna necesaria en vista de la cantidad de tiempo que se sigue invirtiendo en el desarrollo de esta en cada proyecto nuevo. No obstante de la gran cantidad de trabajos actuales se carece de una metodología establecida para calibrar los sistemas de adquisición de datos que valide los datos, y consecuentemente, los resultados obtenidos por medio del uso de dichos sistemas.

### 1.3 OBJETIVOS

#### 1.3.1 Objetivo General

Desarrollar una herramienta consolidada de adquisición de datos para sistemas de arquitectura abierta basados en FPGA mediante el lenguaje de descripción de hardware VHDL y la librería gráfica Gtkmm

#### 1.3.2 Objetivos Particulares

I).- Identificar las características de cada tarjeta FPGA propietaria, elementos convertidores con los que cuenta y sus límites analógicos.

II).- Diseñar una arquitectura de hardware abierta para la adquisición de datos en plataformas basadas en FPGA que sea adaptable a los diversos sistemas utilizados en la UAQ.

III).- Desarrollar un controlador software para interactuar con la parte de hardware del sistema de adquisición, el cuál debe ser compatible con todas las plataformas.

IV).- Calibrar cada uno de los sistemas mediante pruebas para obtener la resolución, rango dinámico, linealidad y distorsión armónica.

V).- Desarrollo de la interfaz gráfica de usuario en Gtkmm

VI).- Generar un manual de operación del software.

VII).- Generar un manual de calibración.

### 1.4 JUSTIFICACIÓN

El desarrollo de una herramienta de adquisición de datos compuesta por software y hardware, auxilia en la exhaustiva labor de establecer y reconfigurar los parámetros de cada proceso de adquisición de datos apoyándose del diseño del software-hardware realizado en conjunto, siendo adaptable ante plataformas de adquisición basadas en tecnología FPGA y completamente utilizable con las plataformas de adquisición de datos basadas en FPGA existentes y en actual uso durante el desarrollo de proyectos de alumnos y docentes de la UAQ debido a que cumplen con las mismas facilidades que ofrecen los sistemas comerciales.

Una GUI desarrollada por completo mediante la librería gráfica de libre distribución Gtkmm, ofrece grandes ventajas al evitar los gastos provenientes de la adquisición y uso de una interfaz gráfica utilizada por un DAQ comercial, y al ser una herramienta de fácil acceso, existe la libertad de modificar y/o actualizar tanto la interfaz gráfica de usuario, como el software-hardware utilizado para ella. Asimismo se pretende que la interfaz gráfica se convierta en una herramienta pedagógica de fácil manejo para cursos de licenciatura, lo cual ayudará a comprender el proceso de adquisición de datos y facilitará la interacción del usuario con el software y el hardware utilizados.

Un manual de calibración surge como respuesta a la necesidad de validar los datos provenientes de cualquier proyecto, con dicho manual se brindará solución a problemas típicos de un convertidor analógico-digital como los son errores de linealidad, errores de ganancia, errores causados por distorsión armónica y el error de offset. El uso del manual de calibración en conjunto con el desarrollo de la herramienta software-hardware hará posible la integración de nuevas plataformas de adquisición basadas en tecnología FPGA, teniendo como parte de su diseño la opción de guardar los resultados de calibración de un sistema en específico para reducir el tiempo y los recursos invertidos en un futuro proceso de adquisición de datos en el cual se emplee la misma plataforma FPGA.

Mediante este trabajo se pretende dar un aporte a los proyectos que conlleven un proceso de monitoreo de variables físicas y sean realizados en colaboración de la Universidad Autónoma de Querétaro mediante el desarrollo de una herramienta consolidada que sea adaptable, genérica y que entregue datos calibrados y ordenados, con diversas plataformas de adquisición de datos de arquitectura abierta basados en FPGA e

incluyendo una interfaz gráfica de usuario desarrollada en la librería gráfica Gtkmm como complemento.

### 1.5 PLANTEAMIENTO GENERAL

En el diagrama de la Figura 1-1 se ilustra de una forma simplificada el proceso de calibración de datos utilizando la herramienta desarrollada durante este trabajo.

#### **Figura 1-1 Proceso de Calibración y adquisición de datos**

El proceso comenzará con una variable física a medir, la cual será diferente dependiendo del tipo de proceso a monitorear, y la cual deberá ser adaptada a través de una serie de transductores y un acondicionamiento de señales pertinente para poder ser leída por una plataforma FPGA, ya sea utilizando un convertidor analógico-digital interno, o uno externo.

Una vez tomada la lectura por la plataforma FPGA, la comunicación con la computadora personal se realizará por medio del protocolo USB, el cual permite una velocidad de transmisión de datos aceptable, y el cual es de uso muy común en la actualidad.

El manual de calibración en conjunto con el manual del software asistirán durante el proceso de adquisición de datos utilizando el software desarrollado, el cual obtendrá los resultados de los análisis de resolución, rango dinámico, linealidad y distorsión armónica, particulares de cada sistema de adquisición de datos empleado mientras que se toman las lecturas del convertidor mediante la plataforma FPGA que se encuentra en continua comunicación con la computadora personal por medio del protocolo USB. Una vez cuantificados cada uno de los errores mencionados se tendrá la posibilidad de corregir dichos errores por medio del mismo software llevando a cabo la calibración final del sistema DAQ. Cabe señalar que la opción anterior se tendrá disponible para todos los errores, a excepción del error de distorsión armónica, el cual simplemente emitirá una alerta en caso de que el convertidor en cuestión deba ser reemplazado por alguno nuevo.

La Figura 1-2 esquematiza el proceso de calibración a utilizar, el cual se basará en comparar una serie de datos obtenidos por el ADC a calibrar, con una serie de datos obtenidos por medio de un instrumento de mayor jerarquía metrológica.



**Figura 1-2 Diagrama simplificado del proceso de calibración**

Gracias al co-diseño de la arquitectura abierta del software y del hardware los parámetros de adquisición de datos podrán modificarse cómodamente desde la interfaz gráfica de usuario de acuerdo con las necesidades del usuario, y además estarán sujetos a las restricciones apropiadas para cada uno de los dispositivos FPGA con el cual se esté trabajando, siempre y cuando el proceso de calibración e identificación de la tarjeta haya sido concluido

## 2. REVISIÓN DE LITERATURA

### 2.1 ESTADO DEL ARTE

Debido a que durante el desarrollo de este trabajo se va a abordar el tema de la calibración de sistemas de adquisición de datos, es inminente la necesidad de al menos una breve introducción hacia el convertidor A/D para familiarizarse con las características propias de cada uno de estos dispositivos, sus parámetros de selección de acuerdo con la aplicación en la cual vayan a ser empleados y una breve descripción de los parámetros y metodologías de prueba más sobresalientes durante la calibración de estos dispositivos.

A continuación se presenta una introducción del proceso de implementación de un lenguaje de descripción en Hardware enfocado hacia la tecnología FPGA mediante la cual se desarrolló gran parte del presente trabajo, seguido de un repaso de conceptos básicos sobre los tipos de memorias utilizadas para almacenar datos durante un proceso de adquisición, una descripción más minuciosa de un proceso de adquisición de datos real para entrar de lleno en la materia de convertidores, su metodología de calibración y posteriormente finalizar este capítulo con una breve descripción de la librería gráfica Gtkmm la cual se utilizó como herramienta durante el desarrollo de la Interfaz Gráfica de usuario del software desarrollado.

### 2.2 FPGAs

#### 2.2.1 Introducción a los FPGAs

Los dispositivos programables juegan un papel esencial dentro del diseño de hardware digital, siendo más destacados dentro de esta área los chips de propósito general los cuales pueden ser configurados para una amplia variedad de aplicaciones.

El primer tipo de dispositivo programable en tener un gran aprovechamiento fue el dispositivo de memoria de sólo lectura programable (PROM por sus siglas en inglés *Programmable Read-Only Memory*). Un PROM es un dispositivo de programación el cual consiste en un arreglo de celdas de sólo lectura en el cual se puede implementar algún circuito lógico, teniendo como base de su funcionamiento la interpretación de las líneas de direcciones digitales con las que cuenta como entradas del circuito y siendo definidas las salidas del circuito

en función de los bits almacenado en ellas, haciendo posible de esta manera la implementación de cualquier función de una tabla de verdad.

Dentro de la categoría de dispositivos PROM existen dos grupos básicos:

- **Programables por máscaras (*Mask-Programmable*):** Aquellos dispositivos que pueden ser programados únicamente por el fabricante.
- **Programables en campo (*Field-Programmable*):** Aquellos dispositivos que pueden ser programados por el usuario final.

Ambos grupos tienen ciertas ventajas y desventajas adquiridas debido a su estructura interna; Los dispositivos Mask-Programmable tienen un mayor rendimiento en cuanto a velocidad comparado a los Field-programmable debido a que todas sus conexiones internas son cableadas desde el proceso de su manufactura a diferencia de los Field-programmable, en los cuales sus conexiones internas se llevan a cabo involucrando algún tipo de switch reprogramable, el cual es inherentemente más lento que una conexión física realizada desde el momento de su manufactura (D. Brown, 1992).

A continuación se mencionan las ventajas principales de las FPGAs desde el punto de vista de la industria electrónica:

**Tiempo de Manufactura:** Al ser fabricados de silicio no requieren ser producidos bajo un estándar específico de requerimientos, evitando así que cada una de las partes que los integran deban de ser adaptadas cuidadosamente para solventar cada uno de dichos requerimientos, y como consecuencia el tiempo de producción se reduce, desde la escala de semanas, e inclusive meses requeridos para producir un ASIC (de sus siglas en inglés *Application Specific Integrated Circuit*), a la escala de minutos en la cual se manufactura en la actualidad una tarjeta FPGA.

**Bajo costo:** La producción de FPGAs se encuentra beneficiada por no conllevar un proceso de manufactura tan complejo como el de un ASIC, derivando en un bajo costo de producción inclusive cuando altos volúmenes de producción no son requeridos. Por el lado de los ASIC los gastos generales para inicializar la producción de un MPGA (de sus siglas en inglés *Programmable Gate Array*) ronda entre los US\$ 20,000 y los US\$ 200,000 mientras que se puede inicializar la producción de prototipos FPGA con costos ubicados desde la escala de los US\$ 100.00 marcando una diferencia bastante considerable, ésta característica sumada el



corto tiempo de manufactura convierten la tecnología FPGA en la más óptima para el desarrollo de prototipos de investigación. (D. Brown, 1992)

**Alta eficiencia:** Gracias al diseño concurrente del hardware, los FPGAs exceden la potencia de cómputo de los procesadores digitales de señales secuenciales típicos, en los cuales, el software a nivel driver se encarga de administrar los recursos de hardware y el sistema operativo tiene la tarea de administrar la memoria y el ancho de banda del procesador. Un FPGA prescinde de esta arquitectura y en su lugar toma ventaja de la designación de entradas y salidas a nivel de hardware para realizar un mayor número de operaciones por cada ciclo de reloj, ofreciendo tiempos de respuesta más veloces que las de los dispositivos secuenciales comunes.

**Fiabilidad:** Gracias a no basar su funcionalidad en algún sistema operativo, los circuitos de un FPGA son una implementación segura de la ejecución de un programa, utilizando hardware preciso dedicado a cada tarea, y evitando que las tareas designadas se obstruyan entre sí, como suele suceder con los sistemas basados en procesadores.

**Reconfigurabilidad:** Los chips FPGA, al ser reconfigurables, son capaces de mantenerse al tanto con las modificaciones necesarias para la actualización de un proyecto o incorporación de nuevas características para un diseño establecido anteriormente, a diferencia de los ASIC, los cuales no pueden ser fácilmente adaptables para su uso en alguna aplicación que difiera de su objetivo inicial ya que fueron manufacturados precisamente con una aplicación en específica en mente (2004, Thompson).

En base a las ventajas previamente mencionadas es perfectamente entendible que la adopción de chips FPGA en la industria haya sido bien aceptada e impulsada por la constante demanda de producción de nuevos productos en el menor tiempo, volviéndose la tecnología FPGA la base de una gran cantidad de proyectos industriales y de investigación.

### 2.2.2 Arquitectura de un FPGA

Para realizar una breve descripción de la estructura interna de los FPGAs, basta con definir sus elementos programables básicos, los cuales son mostrados en un diagrama representativo en la Figura 2-1, y los cuales se pueden agrupar en los cuatro grupos de elementos siguientes:

**Bloques de Lógica Configurable (CLBs de sus siglas en inglés *Configurable Logic Blocks*):** Los CLBs son bloques reprogramables constituidos por semiconductores de silicio, los cuales son pre-construidos internamente en todo dispositivo FPGA teniendo la capacidad de ser programados y/o reprogramados para realizar alguna operación lógica específica descrita por el usuario, permitiendo desde la realización de operaciones lógicas básicas como NOT o AND, hasta llegar a desempeñar operaciones combinacionales de alta complejidad.

**Recursos de interconexión:** El interconexionado de los CLBs se logra mediante el uso de la tecnología fusible del tipo OTP, a través de antifusibles o mediante el uso de células tipo SRAM. Dicho interconexionado se caracteriza por ser llevado a cabo utilizando una arquitectura jerárquica reconfigurable entre dichos bloques, y el hecho de ser completamente reconfigurable en cualquier momento por el usuario.

**Bloques de entrada/salida (IOBs de sus siglas en inglés *input-output blocks*):** Son los bloques de interconexión con otros dispositivos, siendo utilizados para adquirir cualquier tipo de señal exterior como entrada para su posterior almacenamiento dentro del chip en forma de registros y para enviar señales de registro hacia algún periférico como salidas.

**Memoria RAM:** Las células de memoria de configuración, usualmente consistiendo en memoria RAM, se encuentran distribuidas a lo largo de todo el chip, siendo las encargadas de almacenar toda la información necesaria para programar los CLBs al inicializar el proceso de carga del programa de configuración (Zeidman, 2006).

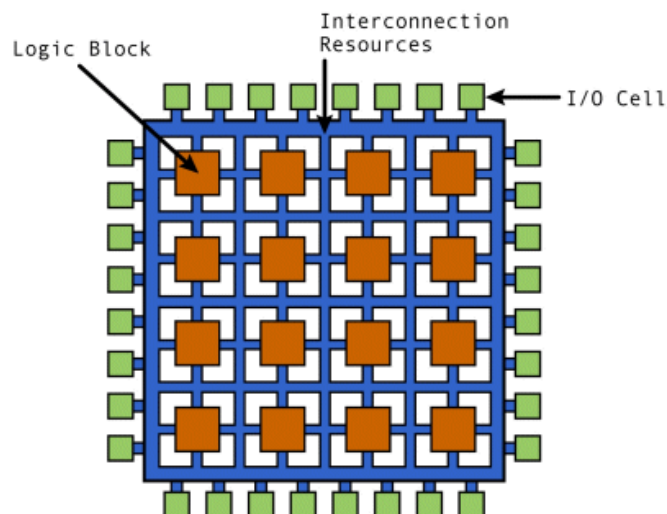


Figura 2-1 Arquitectura genérica de un FPGA (Zeidman, 2006)

Dichos elementos básicos interactúan durante el proceso de diseño de circuitos digitales mediante FPGA, teniendo como resultado final un producto el cual se comporta como si se tratara de un chip único. El proceso implementación de dicho circuito digital se logra ingresando un sistema CAD (de sus siglas en inglés *Computer-aided design*) de dibujo de esquemas o un código fuente en lenguaje de descripción en hardware (HDL) dentro del FPGA creando un programa denominado *bitstream*, el cual es utilizado para configurar la topología del arreglo de conexiones internas propias del FPGA, el uso de la memoria SRAM y las relaciones con los puertos de entrada y salida pertinentes.

Es importante remarcar que por desarrollo en hardware se entiende al hecho de implementar o escribir un software dentro de un FPGA debido a las repercusiones en la constitución física (Hardware) que tiene la implementación de dicho software en el dispositivo FPGA, y no deberá de ser confundido por el proceso de fabricación de un ASIC (Schaumont, 2003).

### 2.3 TIPOS DE MEMORIA

En trabajos de investigación usualmente se requiere de una memoria con la cual auxiliarse para guardar los datos recabados por un sistema DAQ. Los dispositivos y circuitos de memoria desempeñan un papel importante en los sistemas digitales debido a que ofrecen medios para almacenar números binarios temporal o permanentemente, con la capacidad de cambiar la información almacenada en cualquier instante. Los diversos elementos de memoria incluyen tipos magnéticos y aquellos que utilizan circuitos electrónicos, denominados biestables (latches o seguros y flip flops) los cuales llevan a cabo la labor de registrar entradas o salidas pasadas a conveniencia del diseñador mediante una configuración o arreglo apropiado de ellos dentro del sistema (Tocci, 2003).

#### 2.3.1 Memoria ROM

Las memoria de sólo lectura (ROM por sus siglas en inglés *Read Only Memory*) son un tipo de memorias diseñadas para retener datos que son permanentes o cuya frecuencia de cambio es muy lenta. Durante alguna operación normal, tal y como su nombre lo indica, no se pueden leer datos en una ROM, pero si pueden leerse los datos almacenados dentro de ella durante el proceso de fabricación.

El proceso de grabado de datos en una memoria ROM se conoce como *programación de la ROM*. Dicho proceso sólo es aplicable para ciertos dispositivos, debido a que existen algunos los cuales están diseñados de forma que no se puedan alterar sus datos una vez se hayan programado. Sin embargo, existen diferentes tipos de ROM cuyos datos pueden ser borrados y escritos con la frecuencia que se desee. A continuación se describe una clasificación de los tipos de PROM:

**MRROM (por sus siglas en inglés Mask ROM):** Este tipo de ROM tiene sus localidades de almacenamiento programadas (escritas) utilizando un negativo fotográfico denominado mascarilla sobre la tarjeta, el cual es implementado por el fabricante según las especificaciones del cliente. Teniendo como una de sus características principales el no poder ser reprogramable.

**PROM (por sus siglas en inglés Programmable ROM):** Existen dos tipos de PROM, los programables por máscara y los programables en campo. Teniendo ambos tipos como característica común el que son dispositivos de programación única, ya sea durante su proceso de manufactura en los programables por máscara o durante su programación por el usuario y careciendo completamente de posibilidad de reprogramación alguna.

**EPROM (por sus siglas en inglés Erasable ROM):** Es una ROM la cual puede ser programada por el usuario, y que también puede borrarse y reprogramarse tantas veces como se desee. Una vez programada esta memoria se vuelve una memoria no volátil que contendrá sus datos almacenados indefinidamente. Su programación se lleva a cabo mediante la aplicación de niveles de voltaje comúnmente en el orden de 10 a 25 V en las direcciones adecuadas del dispositivo (usualmente consistiendo en transistores MOSFET) como medio para escribir el contenido deseado por el usuario, y teniendo como ventaja ante sus antecesoras la capacidad de borrar su contenido, es decir dejar su contenido totalmente en unos lógicos después después de aplicar luz ultravioleta a todo el dispositivo como proceso de borrado.

**EEPROM (por sus siglas en inglés Electrical Erasable ROM):** Es una mejora de la EPROM, en la cual el proceso de borrado se vuelve más eficiente al prescindir de la luz ultravioleta y realizarse por medio de descargas eléctricas, habilitando al usuario el borrar individualmente cada uno de los bits dentro de la memoria.

Ya que todas las tipos de ROMs son no volátiles y no se pierden cuando la microcomputadora se desenergiza, los usos más comunes de las ROMs se encuentran en el

Firmware, memoria de arranque, tablas de almacenamiento de datos, convertidores de datos entré distintos códigos base, y como generadores de señales dentro de un DAC (Tocci, 2003).

### 2.3.2 Memoria RAM

Las memorias de acceso aleatorio (RAM por sus siglas en inglés *Random Acces Memory*) son aquellas memorias en las cuales se puede tener acceso fácilmente a cualquier localidad de dirección de memoria, siendo empleada en las computadoras principalmente como medio temporal para programas y datos. En este tipo de memorias el contenido de muchas localidades de dirección será leído y escrito a medida que se ejecuta un programa, resultando en ciclos de lectura y escritura rápidos los cuales no reducen la velocidad de operación de la computadora, teniendo como principal desventaja el hecho de ser volátiles y perder toda la información contenida en ellas si se interrumpe el suministro de energía o se apaga el dispositivo empleado a diferencia de las memorias PROM.

Las tipos de memorias RAM más utilizadas son las siguientes:

**SRAM (por sus siglas en inglés Static RAM):** Este tipo de memoria es capaz de almacenar datos mientras se aplica energía al circuito, siendo esencialmente flip-flops que permanecerán en un estado determinado, almacenando algún valor en un bit, siempre y cuando no se interrumpa el suministro de energía al circuito. Las RAMs estáticas se encuentran disponibles en tecnología bipolar y MOS, siendo las más comunes las de tecnología MOS debido a sus capacidades mayores de almacenamiento con un bajo consumo de potencia, a pesar de poder alcanzarse mayores velocidades en las de tecnología bipolar.

**DRAM (por sus siglas en inglés Dynamic RAM):** Las RAMs dinámicas son fabricadas con tecnología MOS, siendo reconocidas por su gran capacidad de almacenamiento, bajos requerimientos de consumo de potencia y una velocidad de operación media. Este tipo de memorias guardan los unos y ceros de los datos como cargas sobre capacitores MOS cuya capacitancia es de unos cuantos picofaradios, provocando que debido a la descarga de los capacitores se requiera de la recarga de las celdas de memoria después de un cierto tiempo para no perder los datos almacenados, proceso denominado como *refresca*.

Ambos tipos de memoria presentan características que pueden ser aprovechadas como una ventaja, o el uso de ellas las convierte en opción poco factible dependiendo del tipo de trabajo a realizar, dichas características son resaltadas en la Tabla 2-1

**Tabla 2-1 Características de las memorias RAM**

<b>Características de las memorias RAM</b>	
<b>SRAM</b>	<b>DRAM</b>
Fáciles de utilizar al no requerir de un sistema complejo para el manejo de sus datos	Se necesita un sistema más complejo para el manejo de sus datos debido a su forma de operación
Mayor velocidad a la de la memoria DRAM	Requiere circuitería de soporte externa para su refrescado
Menor capacidad de almacenamiento o <i>densidad</i> comparada a la DRAM	Alta densidad de almacenamiento
	Son bastante económicas al tener un bajo consumo de potencia

Las características de cada una de las memorias ha determinado su rubro de aplicación, siendo la SRAM la preferida en aplicaciones donde los factores de velocidad y reducción de complejidad son más importantes que las consideraciones de costo, espacio y consumo de potencia, tal y como sucede en el caso de los osciloscopios electrónicos, y en los dispositivos de lógica programable (Troncoso, 2007) mientras que la memoria principal de muchas de las microcomputadoras personales de IBM, PC o Apple emplean la DRAM debido a su gran capacidad y bajo consumo de potencia (Tocci, 2003).

## **2.4 PROCESO DE ADQUISICIÓN DE DATOS**

El trabajo presente se enfoca en el proceso de calibración para sistemas DAQ, por lo cual una revisión detallada del mismo proceso de adquisición de datos y las etapas que conlleva es esencial antes de seguir más adelante con la etapa de calibración del sistema DAQ.

2.4.1 Etapas del proceso de adquisición de datos

El proceso de adquisición de datos se define como el proceso mediante el cual la computadora adquiere datos analógicos digitalizados, dicho proceso conlleva diferentes etapas las cuales son incluidas dentro de un diagrama ilustrativo de todo el proceso el cual se puede apreciar en la Figura 2.2.

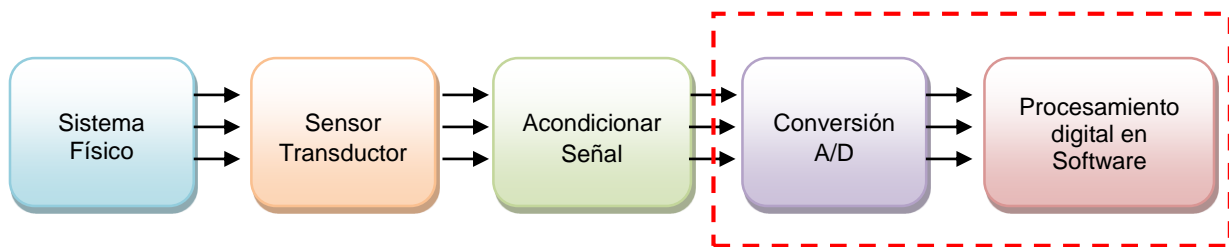


Figura 2-2 Esquema general de un proceso de adquisición de datos

En la figura 2.2 se encuentran remarcadas por medio de un recuadro rojo las etapas hacia las cuales se enfocará más el presente trabajo, sin embargo a continuación se dará una breve explicación acerca del proceso en general.

**Sistema Físico:** El proceso de adquisición de datos comienza con una *variable analógica continua*, es decir que puede tomar valores de magnitud continuos e incluye todos los valores de magnitud posibles dentro del rango de  $\{-\infty \leq R \leq +\infty\}$ , al igual que en el intervalo del tiempo incluye nuevamente todos los valores posibles del tiempo dentro del mismo rango  $R$  tal y como se muestra en la Figura 2-3.

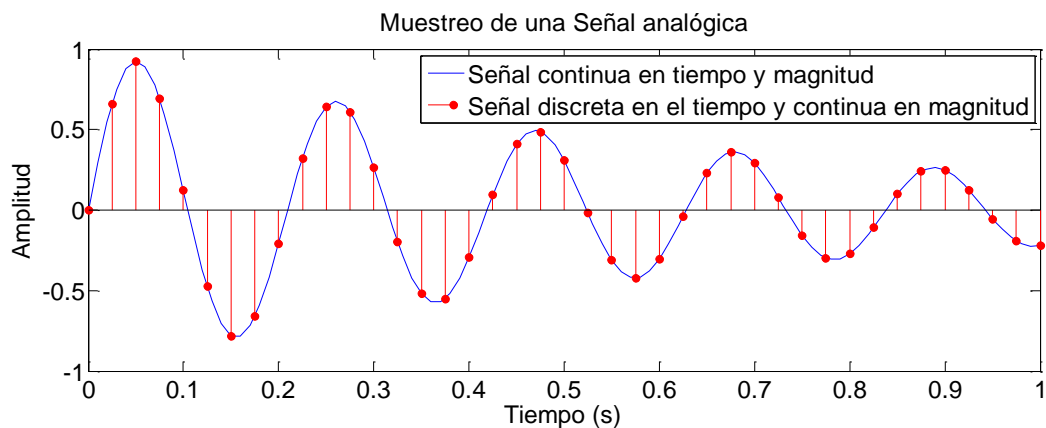
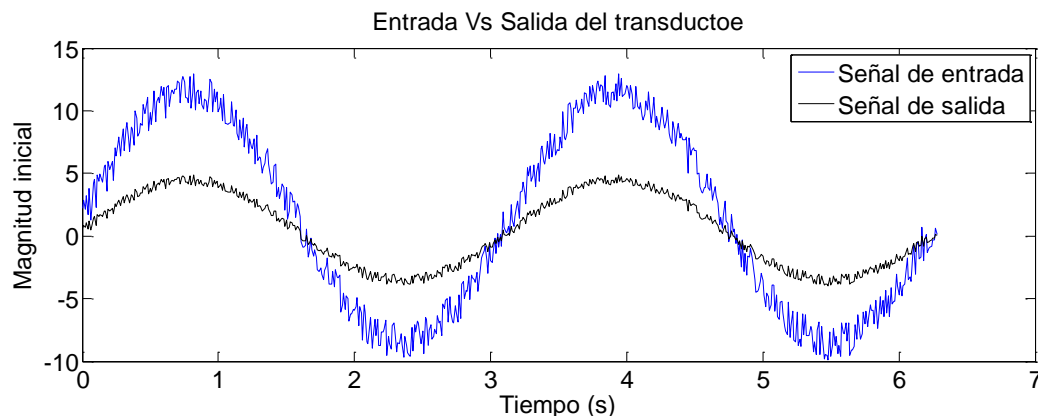


Figura 2-3 Tipos de señales I

Dicha variable continua proviene de una variable física a medir cualquiera, ya sea temperatura, presión, humedad, intensidad luminosa, voltaje, corriente etc. resultando en que la magnitud de dicha variable física está sujeta al tipo de fenómeno a medir y difiere en magnitud de un tipo de variable física a otra.

**Sensor Transductor:** Como respuesta ante la diferencia de magnitudes de los fenómenos físicos, dichas magnitudes son traducidas a algún patrón de medida estándar, el cual es típicamente un valor correspondiente de voltaje o corriente obtenido por medio de un sensor o transductor adecuado para convertir el tipo de variable física a la requerida. La Figura 2-4 ilustra esta etapa del proceso de adquisición de datos al mostrar una señal analógica la cual inherentemente tiene ruido proveniente de la naturaleza, y la cual oscila en una magnitud de  $\pm 10$  unidades de medida dentro del ejemplo, y la cual es convertida por medio de un transductor a un nivel correspondiente de voltaje.



**Figura 2-4 Entrada Vs Salida del transductor**

**Etapas de acondicionamiento:** Tal y como se puede apreciar en la Figura 2.4, la magnitud de la salida cambia respecto a la magnitud de entrada al sensor transductor teniendo un valor más adecuado para su procesamiento digital. No obstante, el cambio de alguna señal por su señal equivalente en voltaje o corriente no es un proceso de conversión perfecto, por lo que usualmente después de esta etapa se tiene la necesidad de amplificar, disminuir, filtrar o linealizar dicha señal, para poder trabajar con una señal más limpia y clara, en donde se vea claramente el fenómeno a estudiar, y la cual está normalizada bajo los niveles estándares de voltaje de  $\pm 5 V$  en caso de trabajar con valores de voltaje, y en valores de  $\pm 5 mA$  en caso de trabajar con corriente. La Figura 2.5 muestra un ejemplo de filtrado sobre una señal como ejemplo de acondicionamiento de señal.



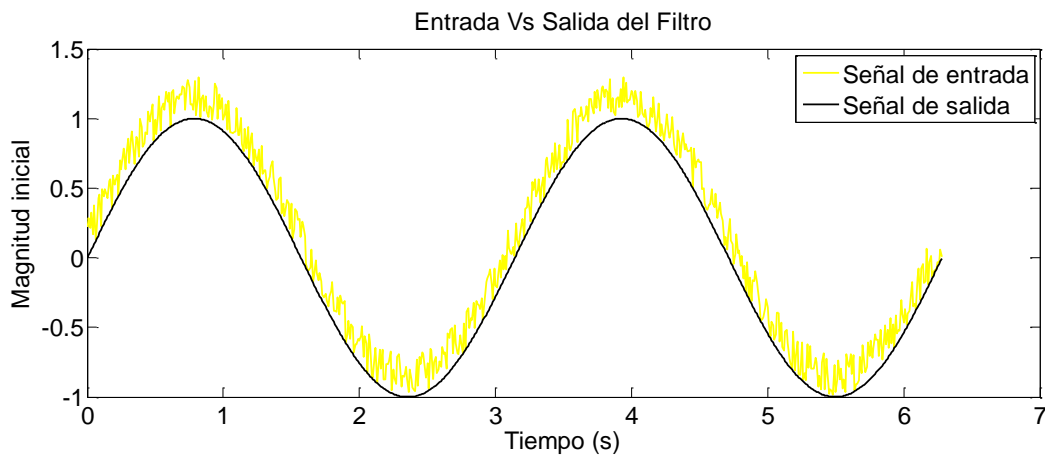


Figura 2-5 Entrada Vs Salida de la etapa de Filtrado

**Convertidor analógico digital:** El convertidor A/D sirve finalmente como el medio de interfaz entra la señal analógica y el mundo de codificación digital, representando de manera univoca todos los valores analógicos de una señal de entrada, la cual se encuentra ubicada dentro de un rango de entrada específico, en una representación por códigos de salida digital, siendo cada uno de estos códigos la representación exclusiva de una fracción del rango de entrada.

La conversión analógica digital se puede considerar como un solo proceso que consta de 4 pasos:

1) Prefiltrado para evitar Aliasing: El convertidor A/D contiene por sí mismo un filtro analógico el cual evita que gran parte del contenido que se encuentra a frecuencias por encima de la frecuencia de muestreo obtenida del Teorema de Nyquist pase a la siguiente etapa tal y como se muestra en la Figura 2.6.

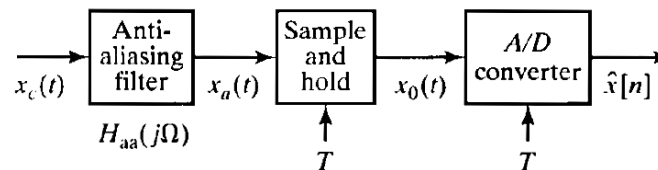


Figura 2-6 Diagrama a bloques Filtro Antialiasing (Oppenheim, 1999)

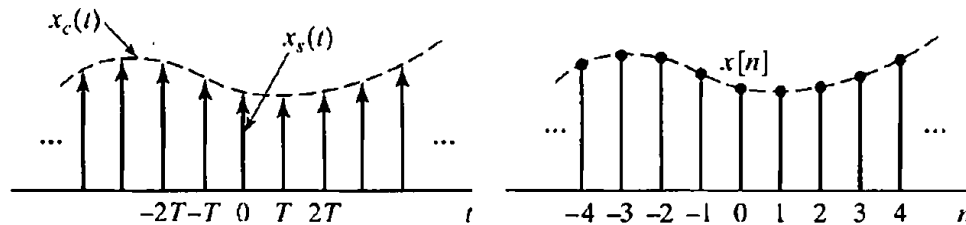
El Teorema de Nyquist es representado por la Ec. 2.1, el cual dicta que “la reconstrucción exacta de una señal periódica continua en banda base a partir de sus muestras,

es matemáticamente posible si la señal está limitada en banda y la tasa de muestreo es superior al doble de su ancho de banda.” en dicho argumento, el ancho de banda usualmente es interpretado como el contenido en frecuencia más alto deseado de una señal cualquiera, y el criterio de Nyquist, usualmente es llevado a la práctica empleando frecuencias de muestreo de 8 a 16 veces mayor que la frecuencia dictada por el teorema de Nyquist.

2) **Proceso de Muestreo:** Es el proceso mediante el cual se realiza la conversión de una señal analógica  $f(t)$  a una señal de datos muestreados  $f^*(t)$ . Matemáticamente se puede apreciar el proceso de muestreo como el producto de un tren de impulsos delta de Dirac y la señal continua en el tiempo  $f(t)$ , siendo este proceso descrito por la Ecuación 2.1.

$$f^*(t) = \sum_{k=0}^{\infty} f(t)\delta(t - kT_0) = \sum_{k=0}^{\infty} f(kT_0)\delta(t - kT_0) \quad \text{Ec 2.1}$$

La naturaleza de las señales es más clara observando a mayor detalle la Figura 2.7 en la cual se ha transformado una señal propiamente analógica a una señal de datos muestreados la cual puede tomar valores de magnitud continuos e incluye todos los valores de magnitud posibles dentro del rango de  $\{-\infty \leq R \leq +\infty\}$  al igual que la señal analógica de la que fue extraída. Sin embargo, a diferencia de la señal analógica original, el dominio de su variable referente al tiempo únicamente se encuentra definida en los valores del rango  $\{0 \leq R \leq kT_0\}$ . Siendo  $k$  cualquier número entero positivo y  $T_0$  el periodo de muestreo del convertidor.



a) Muestreo de señal continua

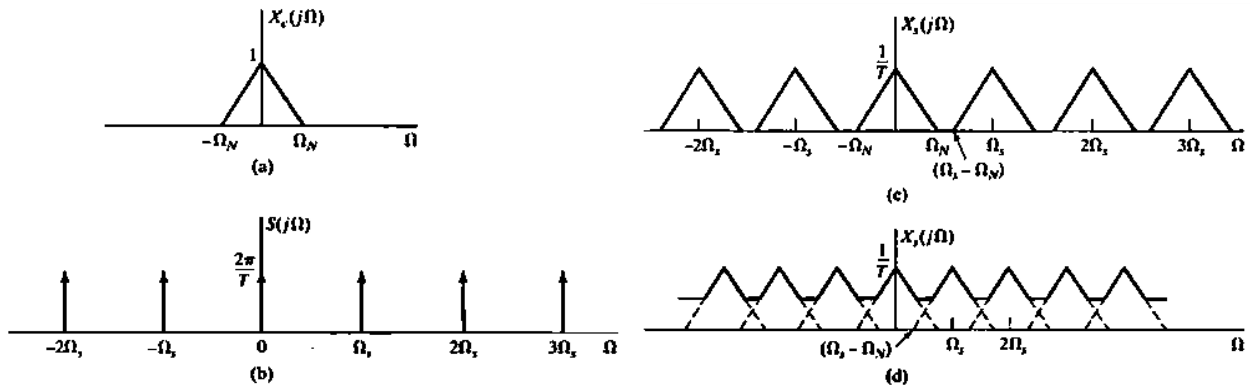
b) Señal Muestreada

**Figura 2-7 Muestreo por medio de tren de pulsos Dirac (Oppenheim, 1999)**

Error de Aliasing

Idealmente una señal se puede reconstruir perfectamente a partir de sus muestras por medio de la aplicación de un filtro pasabajas sobre la señal muestreada. El error de aliasing es el resultado del sub muestreo de una señal, derivando en la imposibilidad de reconstruir la señal original a partir de la señal muestreada. Este fenómeno es más fácil de entender si se analiza el efecto del sub muestreo en el dominio de la frecuencia tal y como se muestra en las Figuras 2.8 a) , b) y c) las cuales muestra los efectos del muestreo sobre la señal original.

En a) se muestra la señal deseada a muestrear con un ancho de banda  $\Omega_N$ , b) representa el tren de pulsos delta de Dirac con el cual se hace el muestreo de la señal a una frecuencia  $\Omega_S$ .



**Figura 2-8 Efectos del muestreo en dominio del tiempo representados en el dominio de la frecuencia (Oppenheim, 1999)**

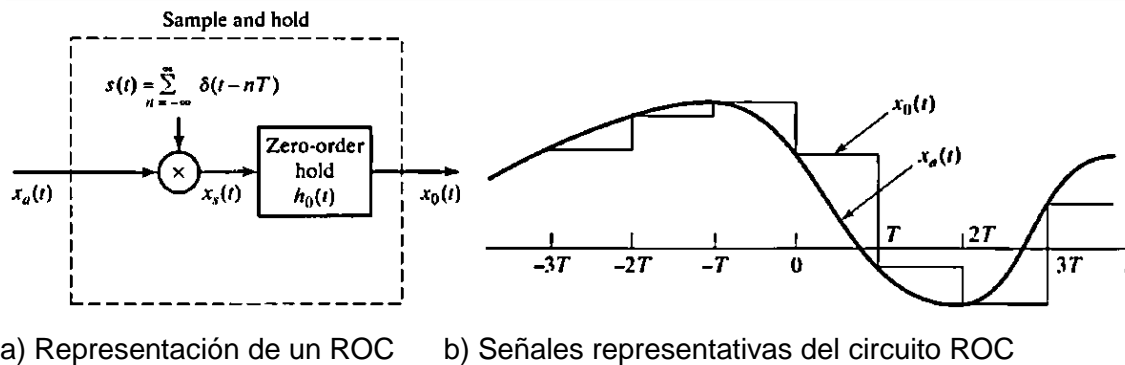
Para interpretar c) Es vital entender que el producto de dos funciones en el dominio del tiempo es igual a la convolución de las mismas funciones en el dominio de la frecuencia, de esta manera c) es la señal muestreada a una frecuencia de muestreo  $\Omega_S - \Omega_N \geq \Omega_N$ ; o en otras palabras a la frecuencia de Nyquist  $\Omega_S \geq 2\Omega_N$ , En dicha figura se puede apreciar que la señal a) es perfectamente reconstruible a partir de su señal muestreada c) por medio de un filtro pasabajas con frecuencia de corte  $\Omega_N$ . En d) se ilustra un muestreo a una frecuencia  $\Omega_S - \Omega_N \leq \Omega_N$ ; . Es evidente que la señal a) no es perfectamente reconstruible por medio de un simple filtro debido a la intersección que existe entre las réplicas creadas por la convolución de las señales, lo cual genera una distorsión de la señal original y es conocido como el error de aliasing.

Circuito retenedor de orden cero (ROC)

El convertidor A/D tiene la tarea de realizar la conversión cada periodo de muestreo  $T_0$ , la cual es iniciada o desactivada a través de una señal proveniente de un driver externo. Sin embargo, dicha conversión no es instantánea, por lo cual requiere mantener la señal de entrada obtenida durante un lapso de tiempo equivalente a  $T_0$ . Dicha labor se logra por medio de un dispositivo denominado retenedor de orden cero (ROC) y el cual se define mediante su respuesta impulso la cual es la descrita en la Ecuación 2.2

$$h_0(t) = \begin{cases} 1 & 0 < t < T \\ 0 & \text{en otro caso} \end{cases} \quad \text{Ec 2.2}$$

En la Figura 2.9 a) se muestra el esquema idealizado de un circuito convertidor A/D con un ROC ubicado a la salida de la señal de datos muestreados, describiendo un comportamiento como el de la Figura 2.9 b), siendo esta señal, el producto final del proceso de muestreo ideal.



**Figura 2-9 Circuito retenedor de orden cero (Oppenheim, 1999)**

- 3) Proceso de Cuantificación: Es la conversión de la señal de datos muestreados a una señal propiamente digital, la cual se define por tomar valores tanto de magnitud, como de tiempo discretos, es decir, el dominio de su variable referente al tiempo únicamente se encuentra definida en los valores del rango  $\{0 \leq R \leq kT_0\}$  y el dominio de su variable referente a magnitud es una función no lineal la cual se encuentra sujeta al número de niveles de cuantización disponibles para su almacenamiento el cual es una función del número de bits utilizados para su almacenamiento.

El proceso de cuantización se describe por el reemplazo de las magnitudes continuas de la señal muestreada por la magnitud correspondiente al nivel de cuantización más cercano, ya sea por uno de los dos criterios más utilizados; truncamiento, en el cual se reemplaza el valor de magnitud continua por el valor de la magnitud correspondiente al nivel de cuantización inferior más próximo, o por redondeo, en el cual se reemplaza el valor de magnitud continua por el valor de la magnitud correspondiente al nivel de cuantización superior más próximo.

Cabe destacar que una señal que ha sido cuantificada no es perfectamente reconstruible a pesar de haber sido muestreada de una forma adecuada, debido a que por la naturaleza de etapa se presenta una pérdida de información.

- 4) Proceso de Codificación: Es el proceso de codificar cada valor discreto de la señal en una secuencia de bits.

Dicha señal formada por secuencia de bits es adquirida e interpretada finalmente por la computadora para realizar un posterior procesamiento sobre ellas o con fines de mero almacenamiento, finalizando con ello un proceso típico de adquisición de datos.

### 2.4.2 Parámetros críticos del proceso de adquisición

Una vez conocidas las diferentes etapas del proceso de adquisición de datos, es fácil inferir que existe una gran cantidad de parámetros que difieren de una adquisición de datos de un sistema a otro. Entre los parámetros más destacables se encuentran los que se muestran en la Figura 2.10, Aquellos concernientes al convertidor analógico digital y el software utilizado en la PC son descritos brevemente a continuación debido a su relevancia en el presente trabajo:

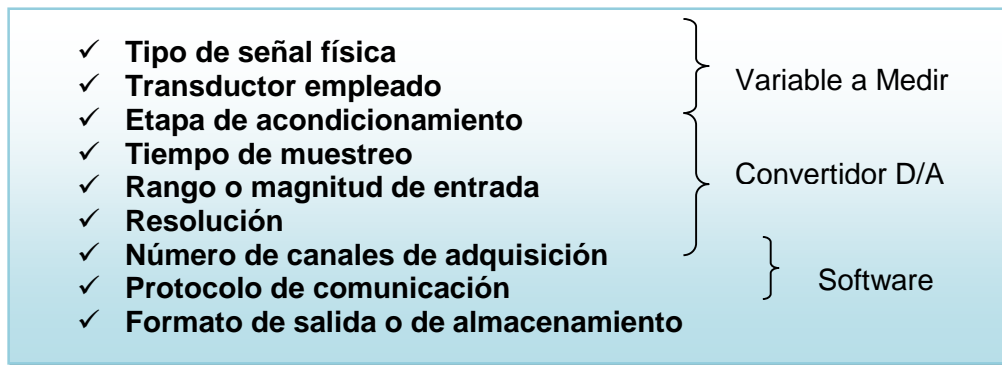


Figura 2-10 Variables de un proceso de adquisición de datos

**La magnitud de las señales analógicas a monitorear:** Las señales analógicas siendo monitoreadas requieren de procesos de acondicionamiento de señales diferentes para poder ser procesadas posteriormente por un dispositivo apropiado,

Existe una gran diferencia entre el monitoreo de una señal de corriente o voltaje de alta potencia cuyas magnitudes pueden llegar a estar en el rango de Kilovoltios, y una señal de corriente o voltaje proveniente de un proceso más sofisticado como lo puede ser un electrocardiograma en el cual una diferencia del rango de milivoltios es bastante significativa y se requiere de un sistema de adquisición de señales bastante sensible y sofisticado.

**El periodo de muestreo requerido:** El periodo de muestreo requerido para caracterizar una señal analógica física depende de la naturaleza de la señal analógica a monitorear y de la fidelidad con la cual se requiera representar dicha variable para su procesamiento, siendo el periodo de muestreo requerido teóricamente definido usualmente por medio del criterio de Nyquist.

Un ejemplo característico es el proceso de control de temperatura y humedad siendo comparado con el proceso de control de iluminación. Durante el proceso del control de temperatura no existe la necesidad de emplear un periodo de muestreo extremadamente alto, debido a que por la naturaleza de la variable física, esta misma no cambia de una forma abrupta en un lapso de tiempo demasiado estrecho, mientras que sucede lo contrario en el proceso de control de iluminación, proceso en el cual, la variable física a monitorear tiene una naturaleza tan dinámica y veloz, que se necesitan periodos de muestreo bastante cortos para poder

caracterizar de una forma aceptable la variable física, y de esta manera poder llevar a cabo un control más fiable y preciso.

**Resolución del Convertidor:** La resolución de un sistema de adquisición de datos varía de la capacidad en bits propia del DAC utilizado. Existiendo desde los equipos más básicos de adquisición los cuales se encuentran montados dentro de los microcontroladores de bajo costo como el PIC 16F877A, cuyo precio oscila entre los \$80.00 y los \$100.00 MXN ofreciendo una resolución de conversión de 10 bits, Y existen los equipos más sofisticados de alta velocidad ofertados por Texas Instruments, por mencionar algunos, los cuales usualmente llegan a trabajar con una resolución de 16 bits, con un costo que ronda entre los \$600.00 y los \$700.00 MXN habitualmente.

**El número de canales de adquisición:** El número de canales de un sistema de adquisición es el número de entradas independientes que el sistema o convertidor es capaz de manejar de forma simultánea. Usualmente se utiliza un canal por cada variable física que se desee monitorear de un sistema a excepción de ciertos sistemas en los cuales se monitorean diferentes variables por un mismo canal auxiliándose de un multiplexor a la entrada del convertidor a cambio de un cierto sacrificio de la velocidad de muestreo. En un sistema de adquisición de datos óptimo se puede necesitar desde un simple canal el cual monitorea una entrada única, hasta  $n$  canales para tener control sobre mayor cantidad de variables posibles, siendo  $n=12$  el número de canales más común con los que cuenta un convertidor.

**Protocolo de Comunicación con la PC:** Una vez adquiridos los datos por medio del convertidor, lo más habitual es tener un registro de las variables físicas medidas durante el transcurso del proyecto, por lo cual el uso de memorias ROM y RAM es muy común durante el proceso DAQ. El uso directo de la memoria de la PC es una alternativa para el almacenamiento de datos, debido a que una vez adquiridos se pueden procesar de fácil manera con algún software auxiliar el cual se encuentre instalado en dicha PC. Dependiendo del proyecto en el cual se esté involucrado la comunicación entre el Convertidor y la PC puede ser más viable establecer un protocolo de comunicación ethernet, un RS-232, un medio de comunicación inalámbrica, un medio de comunicación USB o alguna otra alternativa.

**Formato de Salida o Almacenamiento de datos:** Tal y como se mencionó en el punto anterior el almacenamiento de datos es útil para un posterior procesamiento, lo cual conlleva que el formato en el cual se almacenen los datos sea lo más flexible posible para ser utilizado por

cualquier tipo de software dependiendo de la aplicación. Durante el desarrollo de este trabajo se trata de hacer lo más flexible que se pueda el almacenamiento de datos, teniendo como tentativa el almacenamiento en diversos formatos, tales como el “.txt” perteneciente a un bloc de notas, “.xlsx” perteneciente a una hoja de cálculo de Excel y el formato “.mat” perteneciente a los archivos de datos manejados por Matlab debido a su uso tan común en el procesamiento de datos.

### 2.5 CONVERTIDORES A/D Y D/A

Existe una gran cantidad de parámetros los cuales afectan el comportamiento de un ADC, incluyendo el tipo de señal de entrada, el circuito de aplicación, el periodo de muestreo del convertidor, la temperatura a la cual se está operando, el rango de voltaje de entrada, entre muchos otros aspectos, sin embargo, el objetivo de este trabajo no es extenderse en cada uno de ellos debido a que estos parámetros difieren de un ADC proveniente de un fabricante a otro, por lo cual a continuación se mencionan los parámetros de especificación más generales de los convertidores, dividiéndolos en dos grupos de parámetros:

#### 2.5.1 Parámetros de Precisión

**Rango dinámico:** Es el cociente entre el número más grande el cual puede ser convertido, entre el número más pequeño que se puede convertir. Para un convertidor cualquiera de  $n$  bits su rango dinámico se puede calcular mediante la ecuación Ecuación.2.3.

$$\text{Rango dinámico} = 2^n \quad \text{Ec. 2.3}$$

**Error de Cuantización:** El error de Cuantización es un error natural que ocurre cuando una señal es convertida de una forma analógica a una forma digital aun cuando la conversión se realice a la perfección. En la Figura 2-11 se puede apreciar que el error de cuantización disminuye cuando se acerca al punto de transición debido a que es cuando el error por redondeo o truncamiento es nulo, y que a medida que se aleja de él el error de cuantización aumenta hasta coincidir nuevamente con un nivel de cuantización.



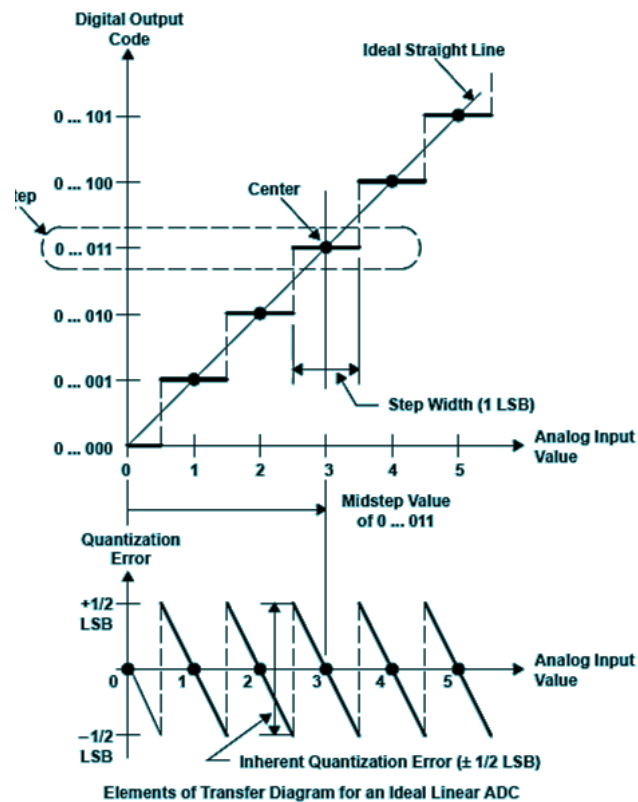


Figura 2-11 Función de transferencia Ideal de un ADC (Mixed,1995)

**Resolución:** La resolución se define como el número de niveles cuánticos con los cuales se puede representar una señal de entrada. La resolución se denomina usualmente en *bits* y se calcula utilizando la Ecuación. 2.4:

$$N = \log_2(n) \quad \text{Ec. 2.4}$$

$N = \text{Resolución (medida de bits)}$

$n = \text{Número de niveles cuánticos disponibles}$

**Error de Ganancia:** La ganancia de un ADC es equivalente a la pendiente de su función de transferencia. De esta manera, el error de ganancia es la medida de la desviación de la pendiente real, respecto a la pendiente ideal definida en la Ecuación. 2.5 Este tipo de error se ilustra en la Figura 2-12 utilizando un convertidor de 8 bits como ejemplo.

$$\text{Ganancia} = \frac{Ref}{2^{n-1}} \quad \text{Ec. 2.5}$$

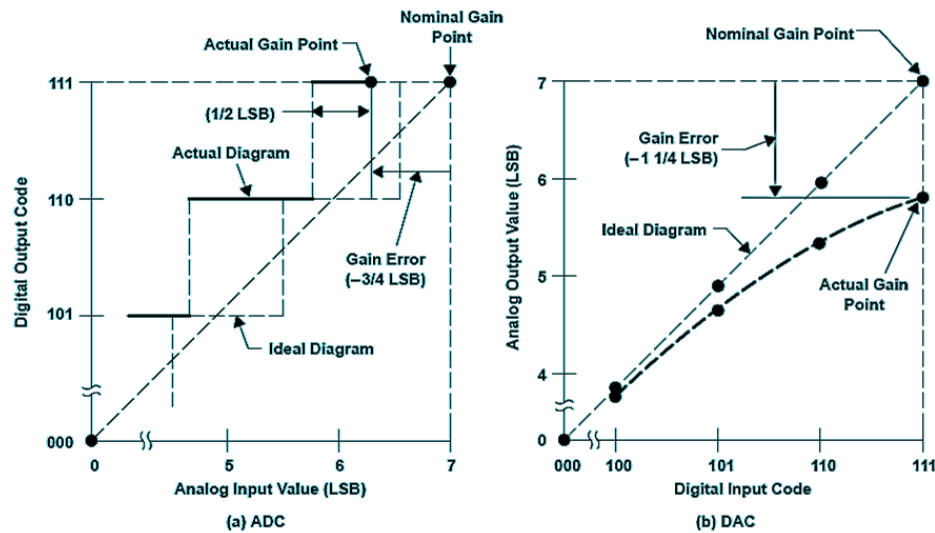


Figura 2-12 Error de ganancia (Mixed,1995)

**Error de Offset:** Se define como la diferencia entre el punto de inicio ideal y el punto de inicio real. El punto de inicio ideal es igual tanto para ADCs como para DACs teniendo el valor de cero. Usualmente este tipo de error afecta a todas las lecturas del convertidor, por lo que en la práctica se acostumbra a corregir por medio de un proceso de suma o resta. Para ayudar a comprender este error se incluye la Figura 2-13, la cual ilustra un error de ganancia típico en un ADC y en un DAC.

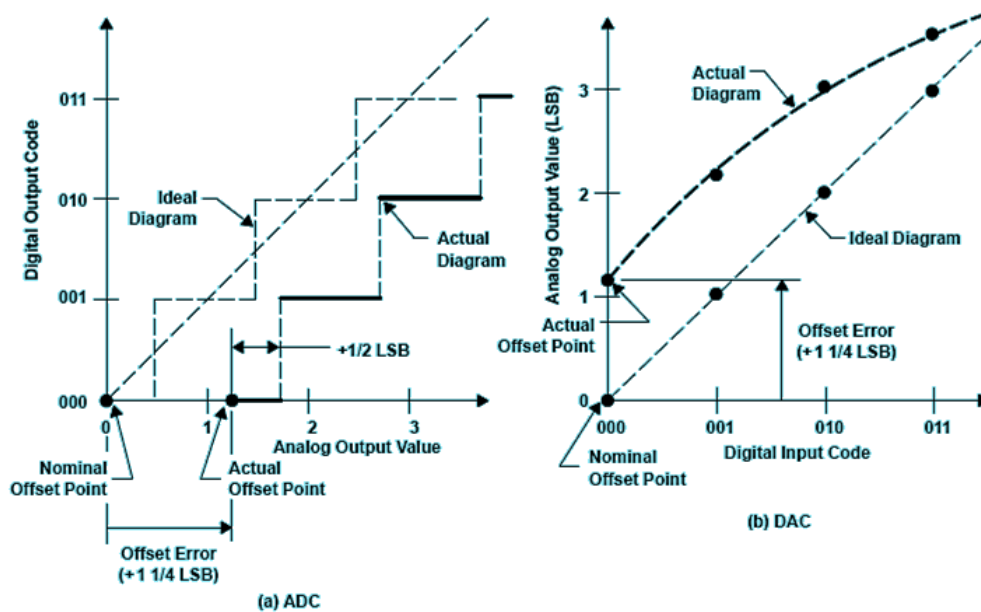
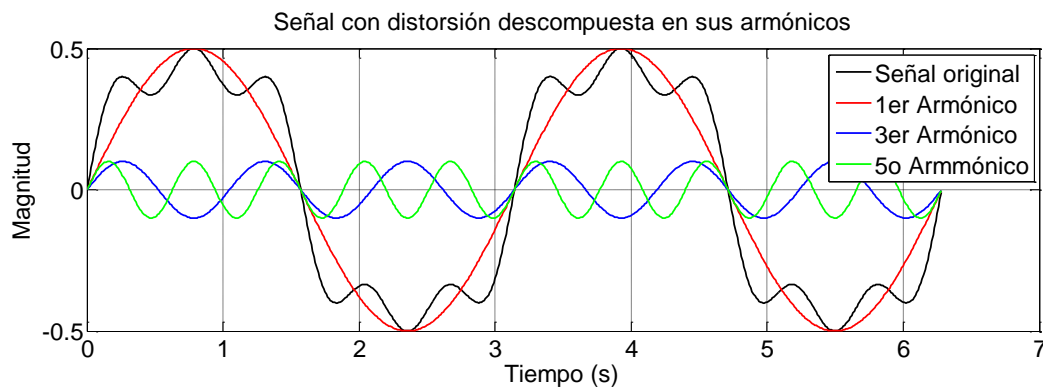


Figura 2-13 Error de Offset (Mixed,1995)

**Distorsión Armónica Total** (THD por sus siglas en inglés *Total Harmonic Distortion*): Es el cociente entre el valor RMS de la señal fundamental o primer armónico, entre el valor RMS de la suma de sus armónicos. Siendo generalmente los primeros cinco los más significativos. El THD de un ADC también es generalmente especificado con una señal de entrada cercana a la escala completa del convertidor, a pesar de que puede ser especificada a cualquier nivel.

En la Figura 2-14 se muestra una señal que contiene bastante distorsión armónica, así como su descomposición en tres señales; frecuencia fundamental o primer armónico, tercer armónico y el quinto armónico. Tal y como se puede observar la señal original es igual a la suma de las tres señales mostradas. En la práctica una onda con un índice de THD alto no es muy útil para realizar algún trabajo debido a los daños que causa su inestabilidad sobre un sistema físico, y en su lugar se prefiere una señal lo más pura posible, es decir, una libre de armónicos a excepción del fundamental.



**Figura 2-14 Distorsión armónica total en el dominio del tiempo**

**Error de No linealidad Diferencial** (DNLE de sus siglas en inglés *Differential Nonlinearity Error*): Es la diferencia entre la dimensión real de un paso de y la dimensión ideal de un paso igual al valor del bit menos significativo (LSB por sus siglas en inglés de *Least Significant Bit*). Si el DNLE excede en magnitud a 1 LSB, es de especial atención, debido a que el convertidor puede estar trabajando regularmente mientras que se están perdiendo niveles de codificación binarios. Este tipo de error se puede entender mejor analizando la Figura 2-15.

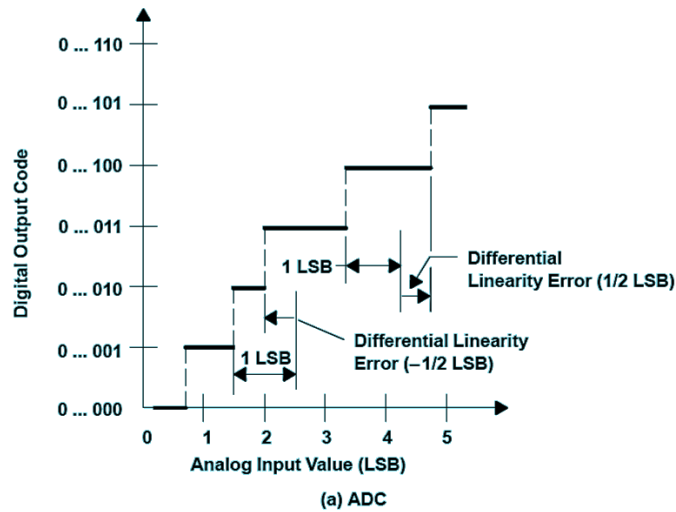


Figura 2-15 Error de No linealidad Diferencial de un ADC” (Mixed,1995)

**Error de No linealidad Integral** (INLE de sus siglas en inglés Integral Nonlinearity Error): Este tipo de error es la medida de la desviación de los valores de la función de transferencia en línea recta. Usualmente se mide el valor de desviación en cada uno de los puntos de transición, recibiendo el nombre de integral debido al hecho de que la desviación de los niveles más altos depende de la suma de los errores de los niveles inferiores tal y como se muestra en la Figura 2-16 la cual muestra el error de No.linealidad Integral de un ADC y de un DAC de 3 bits. Es de importancia mencionar que este error puede ser cuantificado únicamente después de que el error de offset y el error de ganancia han sido minimizados.

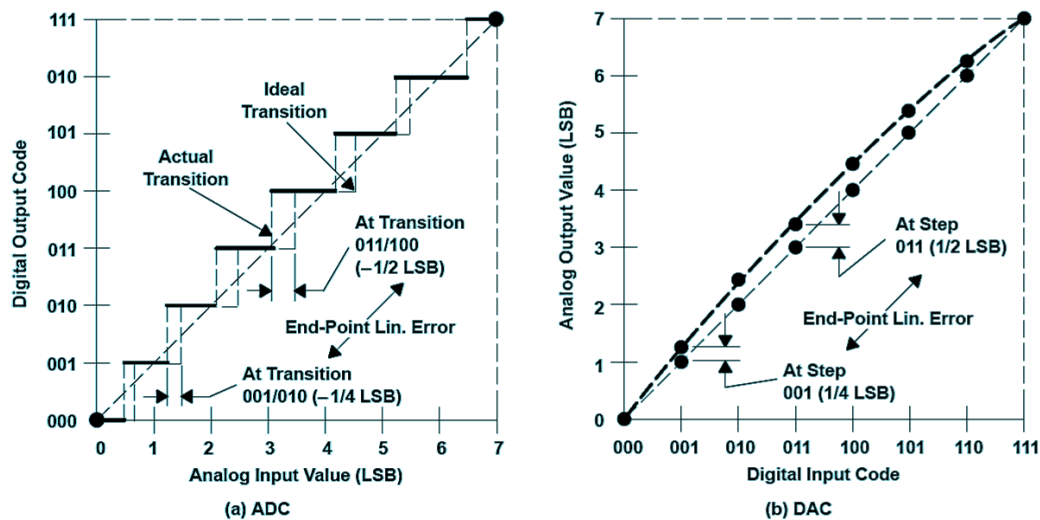


Figura 2-16 Error de no linealidad Integral (Mixed,1995)

## 2.5.2 Parámetros de Operación

**Tiempo de Conversión:** El tiempo de conversión o latencia de un ADC es el tiempo que se requiere para asociar un valor digital de salida correspondiente al valor de la entrada analógica. El tiempo de conversión no es necesariamente igual al tiempo de muestreo, difiriendo principalmente en los ADC con arquitectura *pipeline*. La Figura 2-17 ilustra la diferencia entre el periodo de muestreo y el tiempo de conversión de un ADC el cual requiere de dos periodos de muestreo para completar un periodo de conversión.

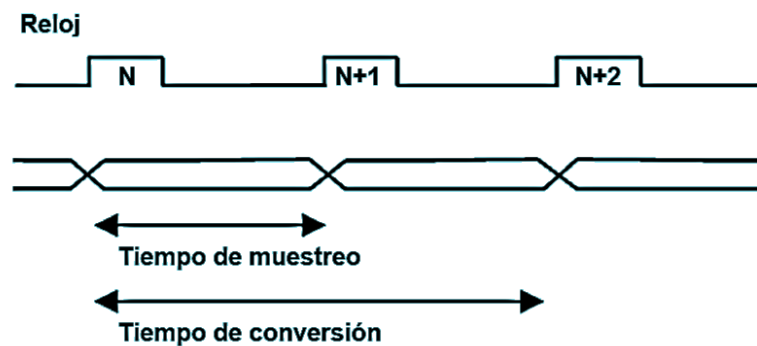


Figura 2-17 Ilustración de Periodo de muestreo y tiempo de conversión (Mixed,1995)

**Formato de Salida:** Se refiere a la letra o número digital correspondiente al valor de la señal de entrada, teniendo como posibilidad el formato binario, hexadecimal o algún otro.

**Rango de escala completa:** El rango de escala completa se define como la conjunto de valores entre el valor máximo y el valor mínimo que puede ser capturado. Para adquirir un valor por medio del convertidor, este valor debe de estar dentro de este rango.

## 2.6 ANÁLISIS EN EL DOMINIO DE LA FRECUENCIA

Debido a que la cuantificación del parámetro de distorsión armónica es uno de los objetivos principales de este trabajo, es necesario abordar los temas más relevantes del análisis en el dominio de la frecuencia, concretamente la herramienta más utilizada para estos fines la DFT (de sus siglās en inglés *Discrete Fourier Transform*) sin indagar a fondo dentro del desarrollo matemático de la herramienta, sino haciendo un mayor énfasis hacia su correcta implementación.

## 2.6.1 Transformada Discreta de Fourier

**Series de Fourier**

La DFT puede ser abordada una vez que se ha comprendido el concepto de las series de Fourier. Las series de Fourier se encuentran basadas en el siguiente Teorema: *“Cualquier función periódica de periodo  $T$  se puede descomponer en una suma ponderada infinita de senos y cosenos cuyas frecuencias son múltiplos enteros de la señal original”*.

De esta manera, las Series trigonométricas de Fourier, o simplemente series de Fourier, son las series infinitas que convergen puntualmente a una función periódica y continua, las cuales fueron desarrolladas por el matemático francés Jean-Baptiste Joseph Fourier alrededor del año 1820 mientras buscaba una solución a las ecuaciones diferenciales parciales derivadas de la ecuación del calor. Las series de Fourier se encuentran representadas matemáticamente por la Ecuación 2.6, y en la Figura 2-18 se puede visualizar la aproximación de una función periódica cualquiera por medio de la suma de armónicos simples, siendo esta aproximación más fiel a medida de que se toman más armónicos contenidos dentro de la serie infinita descrita en la Ecuación 2.6.

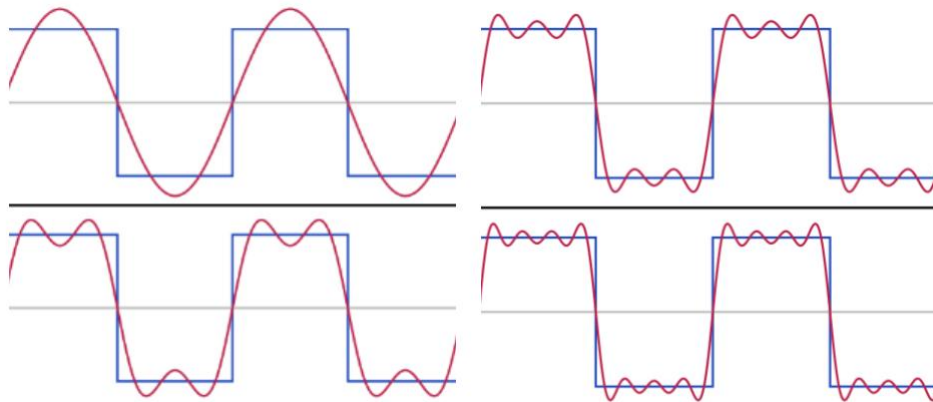
$$F[t] = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t)) \quad \text{Ec 2.6}$$

Donde

$$a_n = \frac{1}{2T} \int_T f(t) dt; \quad a_n = \frac{1}{2T} \int_T f(t) \cos(n\omega_0 t) dt; \quad b_n = \frac{1}{2T} \int_T f(t) \sin(n\omega_0 t) dt$$

$$\omega_0 = \frac{2\pi}{T} = \text{Frecuencia natural (Primer armónico)}$$

Las series de Fourier constituyen la herramienta matemática básica del análisis de Fourier empleado para analizar funciones periódicas en el dominio de la frecuencia a través de la descomposición de dicha función en una suma infinita de funciones sinusoidales mucho más simples llamados armónicos.



**Figura 2-18 Diferentes aproximaciones de una onda cuadrada periódica por medio de Series de Fourier**

### Transformada discreta de Fourier

Sea  $f = (f[0], f[1], \dots, f[N - 1])$  una secuencia de  $N$  valores agrupados. La Transformada Discreta de Fourier (DFT) de  $f$  se encuentra definida como la secuencia de  $N$  valores agrupados  $F = (F[0], F[1], \dots, F[N - 1])$  definidos por la Ecuación 2.7. Se debe intuir por el número imaginario  $j$  que los valores tanto de  $f[n]$ , como de  $F[m]$  son números complejos que resultan de la suma de exponenciales complejas, y los cuales se denominan coeficientes de Fourier. Una forma de interpretar la Transformada de Fourier es como el paso de una función del dominio del tiempo hacia el dominio de la frecuencia, siendo los coeficientes  $F[m]$  de la DFT la representación del armónico  $e^{-\frac{2\pi mn}{N}j}$ , y únicamente de ese armónico (debido a que es una base ortonormal) contenido en la secuencia  $f[n]$ .

$$F[m] = \sum_{n=0}^{N-1} f[n] e^{-\frac{2\pi mn}{N}j}, \quad m = 0, 1, \dots, N - 1 \quad \text{Ec 2.7}$$

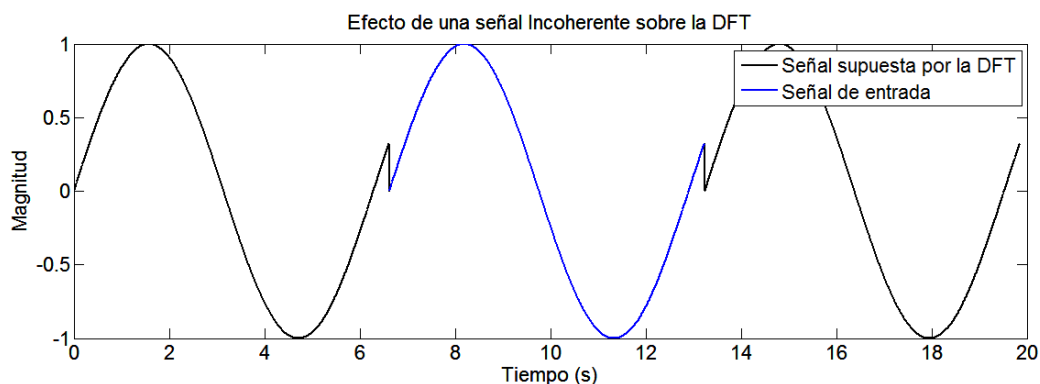
En particular, la DFT o su algoritmo más eficiente la FFT se utiliza comúnmente en procesamiento digital de señales y otros campos relacionados dedicados a analizar el contenido de frecuencias que contiene una señal muestreada, sirviendo también para resolver ecuaciones diferenciales parciales, y para llevar a cabo operaciones de convoluciones de grandes números enteros debido a su propiedad denotada con el mismo nombre, es decir de convolución.

## 2.6.2 Empleo correcto de la transformada de Fourier

**Señal Coherente**

Uno de los problemas principales al emplear la Transformada de Fourier es el de la coherencia de la señal a muestrear. La base matemática de la transformada de Fourier asume que la señal a la cual se le está aplicando la transformada es periódica, continua y de duración infinita. Debido a esto, para que la señal sea coherente debe cumplir con la característica de continuidad, lo cual llevado a un enfoque más práctico indica que la última muestra de la señal discreta debe ser igual a una muestra antes de la primera muestra dentro de la señal, o en otras palabras el número capturado de ondas periódicas debe ser entero.

La Figura 2-19 ilustra el resultado sobre la señal de entrada supuesta por la DFT al utilizar una señal incoherente para obtener su transformada de Fourier, debido a que la última muestra no es una anterior a la primera se puede observar que al replicar periódicamente la señal, está no es continua y por lo tanto se tendrán problemas posteriormente en su análisis espectral. La Figura 2-20 ilustra el espectro de frecuencias de una FFT a 4069 puntos y un número de ciclos capturados no entero (127.5 ciclos), en ella se puede observar que la amplitud del ruido de cuantización ha incrementado considerablemente concluyendo en que realizar un análisis basado en esos resultados derivará en un resultado erróneo.



**Figura 2-19 Efecto de una señal incoherente sobre la DFT**



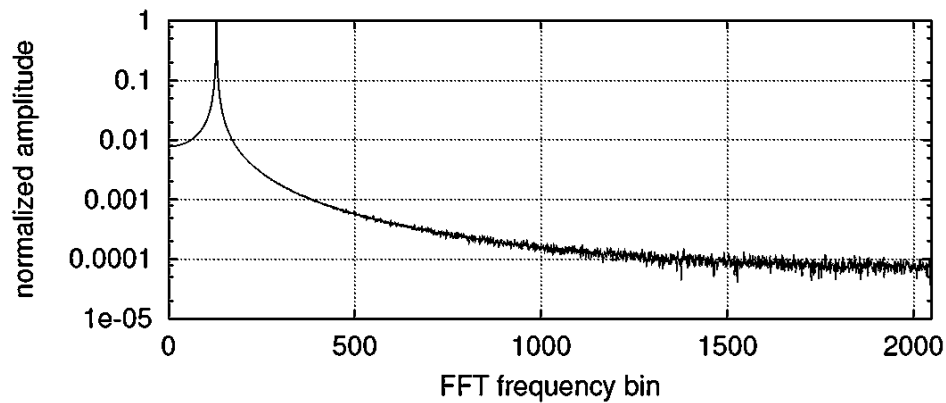


Figura 2-20 FFT de un número de ciclos capturados no entero (4069 puntos, 127.5 ciclos) (Lundberg, 2002)

No obstante, para que la señal de entrada sea coherente no basta con que el número de ciclos de entrada sea entero, sino que además, se necesita que el número de ciclos adquiridos sea un número primo el cual no sea un submúltiplo del número total de datos adquiridos.

En la Figura 2-21 se muestra la FFT a 4096 puntos y 128 ciclos de una señal periódica, en este caso los primeros 32 datos adquiridos de la onda simplemente se repiten 128 veces, originando que el contenido del ruido de cuantización se encuentre completamente concentrado dentro de los armónicos principales en lugar de estar distribuido completamente por todo el ancho de banda de la señal.

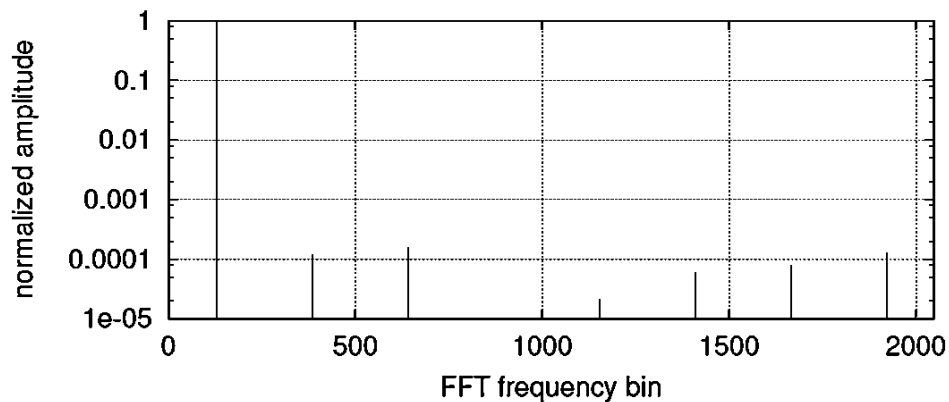


Figura 2-21 FFT de un número de ciclos par (4096 puntos, 128 ciclos) (Lundberg, 2002)

En la Figura 2-21 se muestra la FFT a 4096 puntos y 127 ciclos de una señal periódica, cumpliendo de esta manera con no ser un número entero múltiplo del número de puntos y resultando en un espectro de frecuencias perfecto, el cual sobresalta claramente el contenido armónico fundamental, y en el cual el ruido de cuantización se encuentra bien distribuido a lo largo del ancho de banda.

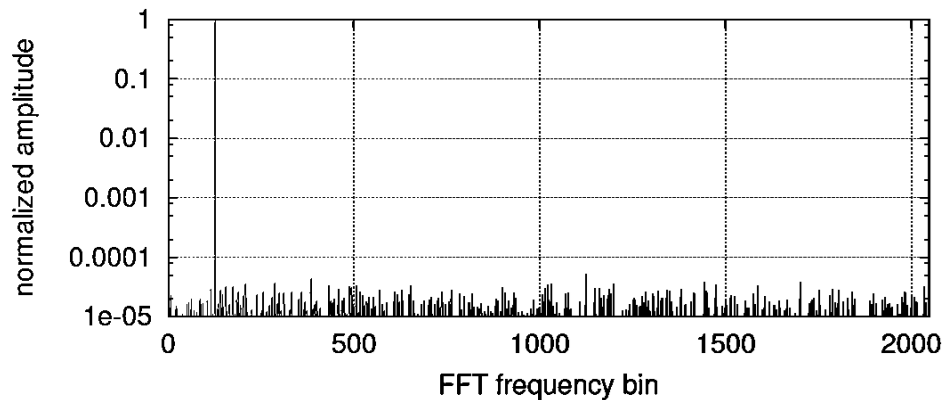


Figura 2-22 FFT de un número de ciclos primo e impar (4069 puntos, 127 ciclos) (Lundberg, 2002)

Debido a que capturar una señal coherente no es una tarea trivial existen diferentes metodologías para lograr que la señal capturada sea coherente, tales métodos son el de ventaneo espectral, el de procesamiento a bloques, el del uso de una Transformada de Fourier dependiente del tiempo y en pocas ocasiones el de tener un control muy fino sobre la onda capturada y obtener únicamente ondas coherentes.

### Muestreo Coherente

Al emplear la DFT durante un procesamiento digital de señales debe de tenerse especial cuidado con la selección de la frecuencia de muestreo en relación con la frecuencia máxima del sistema a analizar. Un muestreo coherente existe siempre y cuando se cumpla la Ecuación 2.8.

Donde

$$F_s K = F_t N \quad \text{Ec. 2.8}$$

$F_s$  = Frecuencia de Muestreo

$K$  = Número entero de ciclos adquirido

$F_t$  = Frecuencia de la onda

$N$  = Número de muestras adquiridas

2.6.3 Ventanas Espectrales

Tal y como ya se mencionó, las ventanas espectrales son funciones matemáticas utilizadas con frecuencia en el análisis y el procesamiento de señales para evitar las discontinuidades al principio y al final de los bloques analizados. En la Figura 2-23 se muestra el proceso de ventaneo de una señal periódica en el dominio del tiempo, dando como resultado una señal de entrada supuesta por la DFT periódica y sin discontinuidades.

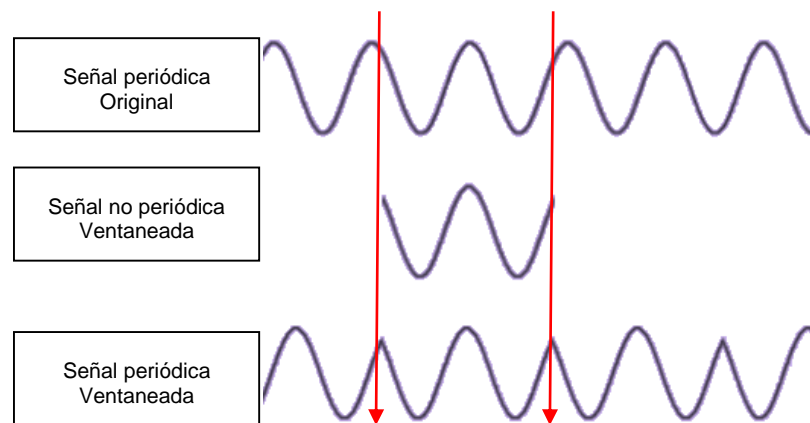


Figura 2-23 Concepto de ventaneo en el tiempo

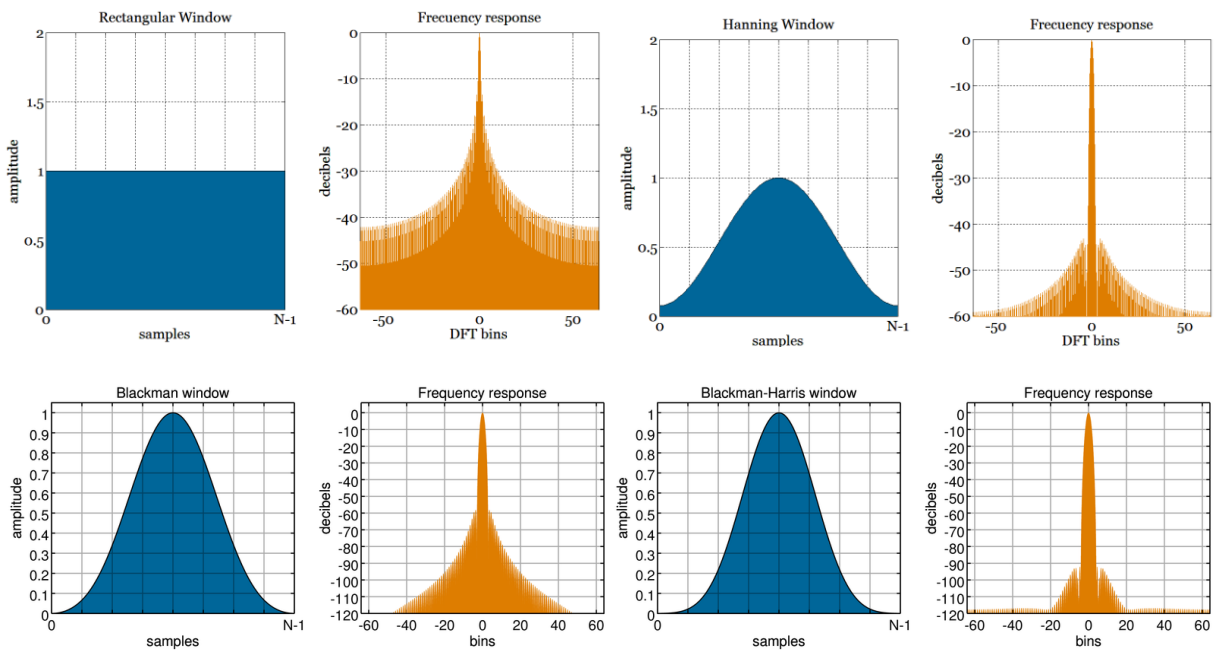
Antes de realizar el proceso de ventaneo es demasiado importante saber que existen diferentes ventanas disponibles y entender el efecto que tiene cada una de estas ventanas sobre el espectro de frecuencias. La Tabla 2-2 muestra las ventanas más comunes así como su ecuación característica.

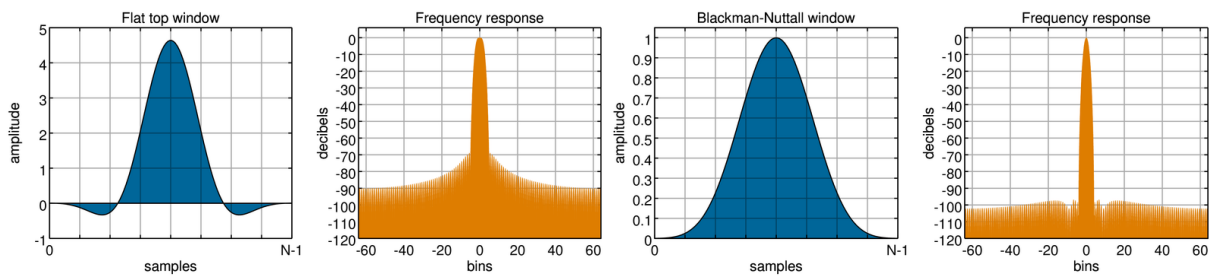
Tabla 2-2 Ventanas espectrales más comunes

Tipo de Ventana	Ecuación característica $v(n)$
Rectangular	1
Hamming	$.53836 - .46164\cos\left(\frac{2\pi n}{N-1}\right)$
Blackman	$.42 - .5\cos\left(\frac{2\pi n}{N-1}\right) + .08\cos\left(\frac{4\pi n}{N-1}\right)$
Blackman-Harris	$.35875 - .48829\cos\left(\frac{2\pi n}{N-1}\right) + .14128\cos\left(\frac{4\pi n}{N-1}\right) - .01168\cos\left(\frac{6\pi n}{N-1}\right)$
Flat Top	$1 - 1.93\cos\left(\frac{2\pi n}{N-1}\right) + 1.29\cos\left(\frac{4\pi n}{N-1}\right) - .388\cos\left(\frac{6\pi n}{N-1}\right) + .032\cos\left(\frac{8\pi n}{N-1}\right)$
Blackman-Nutall	$.3635819 - .4891775\cos\left(\frac{2\pi n}{N-1}\right) + .1365995\cos\left(\frac{4\pi n}{N-1}\right) - .0106411\cos\left(\frac{6\pi n}{N-1}\right)$

La ventaja de utilizar una ventana u otra radica en los efectos que se deseen maximizar por medio del proceso de ventaneo, puesto que al existir una multiplicación en el dominio del tiempo genera repercusiones en el dominio de la frecuencia las cuales pueden ser concentrar el ruido de cuantización en los armónicos, distribuir la potencia del armónico fundamental a lo largo de todo el ancho de banda, o dicha potencia únicamente dentro de las zonas cercanas. Para ayudar a visualizar este efecto se incluye la Figura 2-24 en la cual se pueden apreciar una gran diversidad de ventanas junto con su respectivo efecto en la frecuencia al analizar una onda sinusoidal bajo el ventaneo descrito.

Se puede deducir por la definición de cada ventana que en caso de no implementar una ventana propiamente, se estaría implementando indirectamente una ventana del tipo rectangular teniendo como resultado del ventaneo la Figura 2-19.





**Figura 2-24 Ventanas espectrales en dominio del tiempo y su efecto en el dominio de la frecuencia**

## 2.7 CALIBRACIÓN

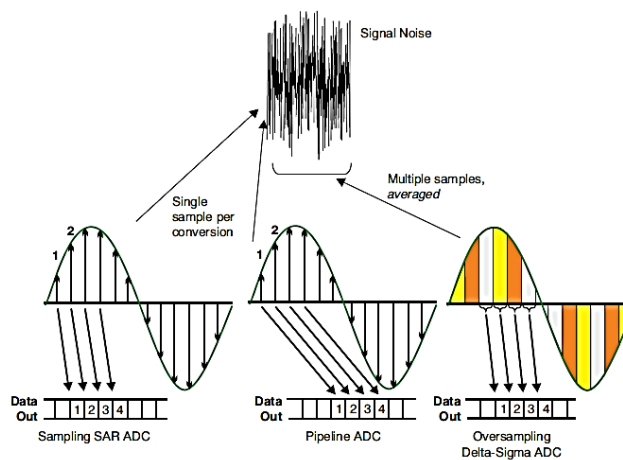
De acuerdo al vocabulario internacional de metrología (VIM), se entiende por calibración un “Conjunto de operaciones que establecen, en unas condiciones especificadas, la relación que existe entre los valores indicados por un instrumento o sistema de medida, o los valores representados por una medida materializada, y los correspondientes valores conocidos de una magnitud medida.” Por lo tanto para calibrar un instrumento es preciso disponer de otro de mayor precisión, y que nos proporcione el valor denominado como *verdadero*, al aplicar consecutivamente la comparación de un instrumento con patrones de mayor jerarquía metrológica se puede establecer de una mejor manera la trazabilidad del instrumento y por tanto, las medidas que con él se efectúen al poder otorgarle una medida de incertidumbre conocida, llegando a el resultado final conocido como calibración

El proceso de calibración debe de efectuarse periódicamente para poder obtener su fiabilidad y no poner en dudas los resultados manifiestos entre las discrepancias que existen entre el patrón de medida y el instrumento a medir, los cuales se acogen dentro de un documento denominado certificado de calibración.

El certificado de calibración permite entre otras cosas, el garantizar la trazabilidad del laboratorio o centro de medición, y por tanto las mediciones que en él se realicen mediante el establecimiento y mantenimiento de un plan de calibración y la participación en intercomparaciones de medida, y permite también la realización de ajustes precisos en los instrumentos de medida, o bien el establecimiento de un criterio de reposición del instrumento (Moro, 2000).

Desafortunadamente no existe un estándar o un certificado de calibración aceptado acerca de cómo especificar mediante pruebas un convertidor ADC. Siendo dicha carencia resultado de la constante evolución e interpretación del término *especificaciones*, el cual ha cambiado con el progreso de las diferentes áreas de aplicación de los convertidores. Sin embargo, ciertas metodologías de pruebas se han consolidado poco a poco, a pesar de que los resultados de dichas pruebas se interpretan de manera diferente dependiendo de la arquitectura del dispositivo en uso. Estas diferencias de interpretación se deben a que los datos recopilados tienen un significado diferente inherente al tipo de señal de entrada con la cual se realiza la prueba, la aplicación del circuito de prueba, el periodo de muestreo del convertidor, y la naturaleza de la arquitectura del convertidor por sí mismo (Moro, 2000).

La Figura 2-25, sirve para comparar el significado diferente del periodo de muestreo de diferentes arquitecturas de ADC. Siendo diferente el proceso de adquisición en un convertidor de sobremuestreo Delta Sigma ( $\Delta\Sigma A/D$ ) en el cual el dato convertido final es el promedio de diferentes muestras tomadas en un intervalo de tiempo muy corto, mientras que en un convertidor de arquitectura SAR el dato convertido que se obtiene es el recopilado por un proceso de muestreo único.

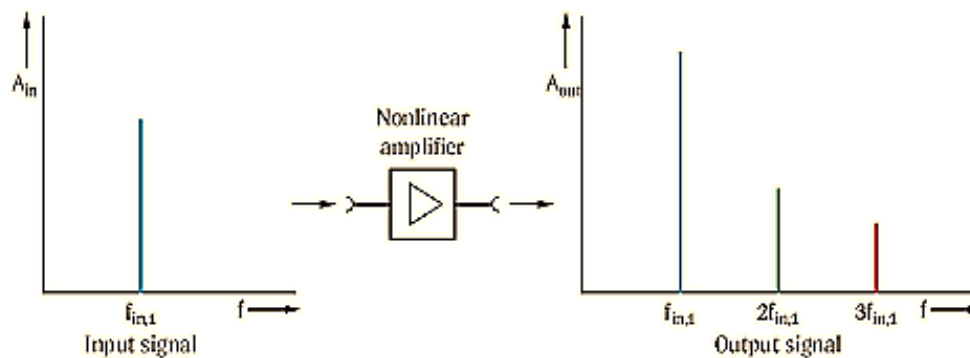


**Figura 2-25 Comparación de Algoritmos de muestreo utilizado por convertidores SAR vs Pipeline vs  $\Delta\Sigma A/D$  (Texas Instruments, 2011)**

Entre las metodologías de pruebas para caracterizar un convertidor A/D más generalizadas se encuentran las pruebas para determinar el error ganancia, el error de offset, el error de linealidad diferencial e integral y la cuantificación de la distorsión armónica total, las cuales son descritas a mayor profundidad a continuación:

### 2.7.1 Distorsión armónica total (THD por sus siglas en inglés Total Harmonic Distortion)

Cuando en un sistema no lineal se introduce un tono de frecuencia  $f_0$ , se espera que en la salida de dicho sistema se obtenga nuevamente el mismo tono con una amplitud y fase posiblemente diferentes y, sumado a él, otros tonos de frecuencia los cuales son múltiplos enteros de la frecuencia fundamental y los cuales son  $2f_0, 3f_0, 4f_0 \dots nf_0$  denominados como armónicos del tono fundamental o primer armónico  $f_0$  tal y como se puede observar en la Figura 2-26



**Figura 2-26 Distorsión armónica total en el dominio de la frecuencia**

La distorsión armónica se produce cuando la señal de salida de un sistema no equivale a la señal que entró en él. Esta falta de linealidad afecta a la forma de la onda, debido a que el equipo ha introducido armónicos que no estaban en la señal de entrada. Puesto que son armónicos, es decir múltiplos de la señal de entrada, esta distorsión no es tan disonante y únicamente se puede detectar de una forma más clara en el dominio de la frecuencia.

La distorsión armónica total se mide en el dominio de la frecuencia utilizando la herramienta apropiada de la Transformada de Fourier, ya sea la DFT o su algoritmo más eficiente la FFT (de sus siglas en inglés *Fast Fourier Transform*), introduciendo en la entrada del convertidor A/D un tono lo más limpio posible con características muy específicas enlistadas a continuación:

- ✓ Magnitud dentro del rango de escala completa del convertidor
- ✓ Contiene un ancho de banda limitado
- ✓ Su ancho de banda no está relacionado con la frecuencia de muestreo

- ✓ Su ancho de banda no generará el denominado error de aliasing al ser muestreado a la frecuencia de muestreo determinada para llevar a cabo la prueba

Cabe mencionar que previo a la aplicación de la transformada se deberá elegir el método de ventaneo pertinente sobre la señal adquirida, el cual es elegido por el usuario de acuerdo a sus necesidades, para obtener de esta manera los coeficientes que arroje la transformada, y por medio de ellos poder calcular el índice de THD de acuerdo a la Ecuación 2.9.

$$THD = \frac{\sum \text{Potencia de los armónicos consecutivos}}{\text{Potencia de la frecuencia fundamental}} = \frac{P_1 + P_2 + P_3 + \dots + P_n}{P_0} \quad \text{Ec. 2.9}$$

El índice de THD se encuentra clasificado como un parámetro dinámico a diferencia del error de offset y error de ganancia, los cuales son clasificados como estáticos. En la Figura 2-27 se muestran los parámetros de especificación dinámicos más comunes de los ADC dentro del dominio de la frecuencia.

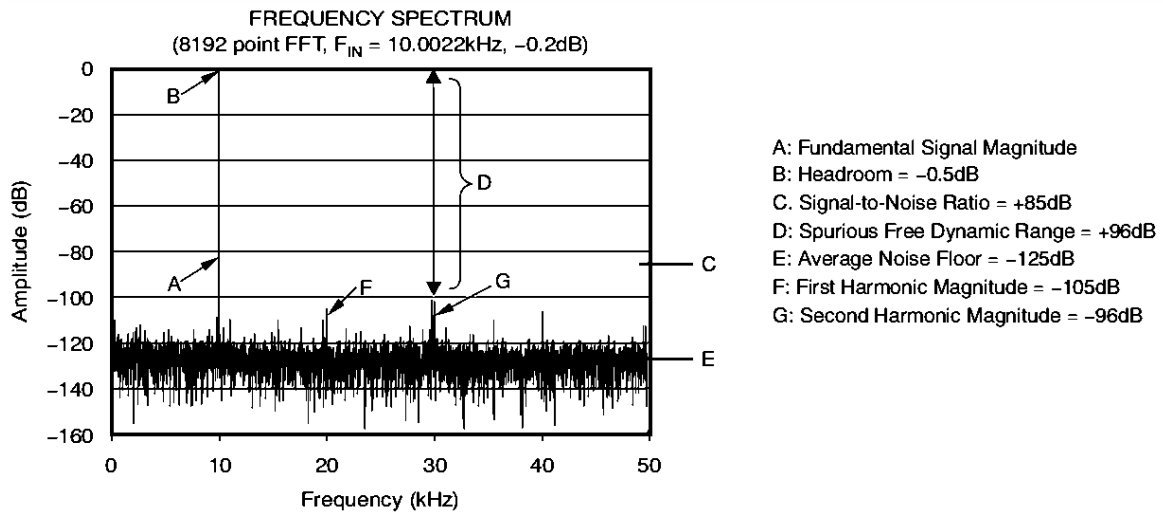


Figura 2-27 "Especificaciones dinámicas de un convertidor (Texas Instruments, 2011)"

A pesar de que no existe un parámetro estándar que dicte límites del índice de THD, existen valores recomendados para equipos electrónicos que trabajen con fuentes de CA, los cuales son de una magnitud del 5% para el índice THD y del 3% para la magnitud de cualquier armónico de la señal (IEEE, 1992). Dichos índices reflejan que el equipo trabaja con señales bajo una distorsión armónica aceptable.



Además de las recomendaciones de la IEEE, existen los parámetros técnicos de los equipos, entre los cuales suele figurar la distorsión armónica total ya sea en forma de porcentaje, o especificado como un rango aceptable medido en dB dentro de las hojas de datos del convertidor provista por el fabricante. En caso de estar fuera de estos rangos recomendados o los especificados por el fabricante, el contenido de la señal proveniente de los armónicos empieza a desvirtuar la señal en lugar de enriquecerla, y la onda resultante comienza a dejar de parecerse al original, lo cual en un sistema de adquisición de datos indica que el mismo debe cambiarse para obtener mediciones de calidad aceptable.

### 2.7.2 Error de Offset

El error de offset se identifica por medio de la desviación entre la localización ideal del nivel más bajo de transición de la función de transferencia de un convertidor A/D o D/A y la localización real de dicho valor.

La localización ideal del valor más bajo de transición tiene el valor de .5 LSB por lo cual para definir el offset matemáticamente basta con utilizar la Ecuación 2.10 (Hoeschele, 1994)

Donde

$$V_{off} = \text{Voltaje de offset ;}$$

$$V_{real} = \text{Voltaje del primer nivel de transición}$$

$$V_{ideal} = \text{Voltaje ideal de transición} = .5 \text{ LSB}$$

$$V_{off} = V_{real} - V_{ideal} \text{ Ec. 2.10}$$

### 2.7.3 Error de Ganancia

Existen distintas maneras de determinar la pendiente de la función de transferencia, la cual ayuda a identificar y cuantificar el error de ganancia de un convertidor. Entre las formas más habituales se encuentra el trazado fino de una recta entre puntos medios del rango de conversión total para obtener una recta la cual se asemeje lo más posible a la función de transferencia real del convertidor.

Existe también la alternativa de determinar el error de ganancia el cual únicamente puede ser utilizado una vez que se haya corregido el error de offset del convertidor en cuestión. En este caso la ecuación de la recta de la función de transferencia se obtiene con la ecuación

de la recta obtenida por dos puntos; el punto cero y el punto obtenido del nivel más alto de transición con el cual cuenta el convertidor.

*Ecuación de la recta obtenida por dos puntos*

Sean  $P(x_1, y_1)$  y  $Q(x_2, y_2)$  dos puntos de una recta. En base a estos dos puntos conocidos de una recta, es posible determinar su ecuación. Para ello se toma un tercer punto  $R(x, y)$ , también perteneciente a la recta.

Debido a que P, Q y R pertenecen a la misma recta, se tiene que PQ y PR deben tener la misma pendiente. En otras palabras

$$m_{PQ} = \frac{y_2 - y_1}{x_2 - x_1}; \quad m_{PR} = \frac{y - y_1}{x - x_1}; \quad Ec. 2.6$$

Luego, la ecuación de la recta que pasa por dos puntos es:

$$\frac{y_2 - y_1}{x_2 - x_1} = \frac{y - y_1}{x - x_1} \quad Ec. 2.7$$

Lo cual también se puede expresar como la ecuación general de la recta descrita en la Ecuación. 2.11 (Lehmann, 1989)

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1) \quad Ec. 2.11$$

2.7.4 Pruebas de linealidad diferencial

Método del histograma

La linealidad de un convertidor A/D, ya sea el DNLE o el INLE, puede determinarse indirectamente a partir de un análisis estadístico conocido como prueba de histograma. En este método de prueba se cuantifica la diferencia entre el número de muestras esperadas de un código, y el número de muestras real que aparece dicho código durante un número conocido de muestras.

Existen dos señales excitatrices de entrada más utilizadas para realizar esta prueba las cuales son la rampa y la onda sinusoidal, obteniendo la denominación del histograma lineal y el histograma dinámico respectivamente. Tal y como se puede observar en la Figura 2-28

ambas señales tienen una distribución de densidad de probabilidad propia y diferente, siendo la de la rampa prácticamente una constante, mientras que en el caso de una onda sinusoidal  $V = A\sin(\omega t)$  se tendrá una distribución de densidad de probabilidad denotada por la Ecuación 2.12.

$$P(V) = \frac{1}{\pi\sqrt{A^2 - V^2}} \quad \text{Ec. 2.12}$$

En la práctica el tipo de onda excitatriz suele ser sinusoidal debido a que para obtener resultados satisfactorios en pruebas con convertidores de resolución mayor a 10 bits, se requiere de una onda excitatriz rampa bastante fina y precisa, lo cual es logrado únicamente por medio de equipos sofisticados y de un precio elevado, mientras que una señal sinusoidal fina es fácilmente reproducible por un generador de señales típico a un menor costo.

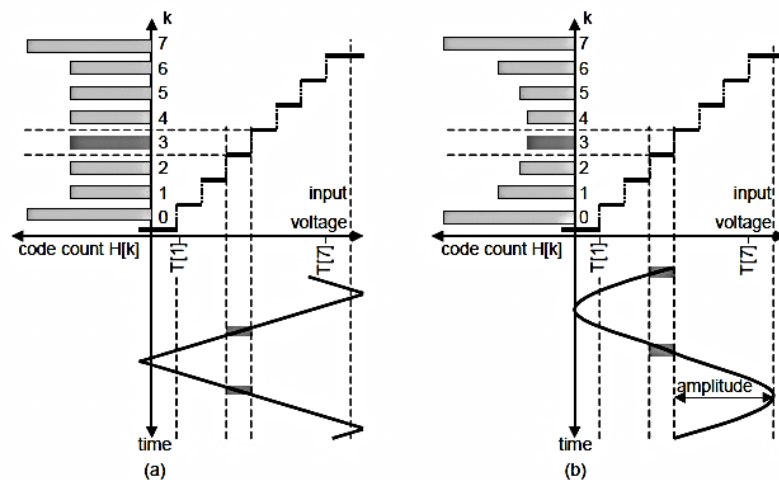


Figura 2-28 "Señales de generación de Histograma más comunes (Yichuang, 2008)"

Sea la entrada de un sistema una onda sinusoidal  $V$ , y su función de densidad de probabilidad la denotada en la Ecuación. 2.12, la probabilidad de encontrar una muestra incluida en el intervalo  $(V_{in}, V_{fin})$  se encuentra determinada por la Ecuación. 2.13

$$P(V_{in}, V_{fin}) = \int_{V_{in}}^{V_{fin}} P(V) dV = \frac{1}{\pi} \left[ \sin^{-1} \left( \frac{V_{fin}}{A} \right) - \sin^{-1} \left( \frac{V_{in}}{A} \right) \right] \quad \text{Ec. 2.13}$$

De esta manera se puede obtener la probabilidad ideal de encontrar un código, la cual se puede comparar con la probabilidad real del código, y de esta manera determinar el DNLE que contiene cada código dentro del convertidor A/D de acuerdo con la Ecuación 2.14 (Lundberg, 2002)

Donde

$H_k = \text{Probabilidad Real de encontrar el código } K$

$P_k = \text{Probabilidad Ideal de encontrar el código } K$

$$DNLE_k = 1 \text{ LSB} \left[ \frac{H_k}{P_k} - 1 \right] \quad \text{Ec. 2.14}$$

A lo largo de este trabajo se utilizará la definición provista por la Ecuación 2.14 y se mantendrá una interpretación del DNLE como buena o aceptable cuando el valor absoluto del DNLE sea mayor a .5LSB y menor a 1 LSB, y en el caso en que el DNLE sea menor a .5LSB o ser mayor a 1LSB el convertidor tendrá secciones de código binario los cuales no se están utilizando correctamente. (Hoeschele, 1994).

Debido a la naturaleza estadística de este método, el número de muestras  $M$  al realizar la prueba es crucial ya que puede ser bastante grande dependiendo del convertidor a ser analizado, y se encuentra directamente relacionado con el nivel de confiabilidad y resolución deseado por medio de la Ecuación 2.15.

Donde

$$M = \frac{\pi 2^{N-1} Z_{\alpha/2}^2}{\beta^2} \quad \text{Ec. 2.15}$$

$N = \text{Número de bits del convertidor}$

$\beta = \text{Resolución deseada de la prueba DNL}$

$Z_{\alpha/2} = \text{Constante de confiabilidad de la prueba}$

El parámetro  $\beta$  es la frecuencia con la cual se muestrea cada código del ADC (el cual tiene un ancho de 1 LSB), por lo cual, una resolución de 1/10 LSB debe ser interpretado como que se han realizado 10 muestras de cada uno de los códigos del ADC.

La constante de confiabilidad de la prueba es determinada de la misma manera que el parámetro  $\beta$ , es decir, por el usuario, siendo los valores más comunes para esta constante los mencionados en la Tabla 2-3

Tabla 2-3 Constantes de confiabilidad más comunes (<http://www.atx7006.com/>, 2015)

Confiabilidad	90%	95%	99%
$Z_{\alpha/2}$	1.645	1.96	2.576

### 2.7.5 Pruebas de linealidad Integral

Existen tres métodos típicos para determinar el INLE:

- 1) Obteniendo la ecuación de la recta de la función de transferencia descrita por dos puntos; el primero punto siendo ubicado en el primer nivel de transición y el segundo punto siendo ubicado en el último nivel de transición con el cual cuenta el convertidor.
- 2) Calculando una recta más fina calculando los valores reales que arroja el convertidor para todos los valores de código que puede representar.
- 3) Debido a que el DNLE representa el error del ancho de paso de un nivel en específico, y que el INLE se define como la suma total de los errores DNLE de todos los niveles de transición, un método poco usual, pero igual de efectivo para calcular el INLE es mediante la sumatoria del DNLE obtenido de cada uno de los niveles de transición del convertidor como se muestra en la Ecuación 2.16

$$INLE_k = \sum_{j=1}^k DNLE_j \quad Ec. 2.16$$

## 2.8 LIBRERÍA GRÁFICA GTK

GTK+ fue un proyecto inicializado y desarrollado como un programa de manipulación de imágenes en general GIMP (por sus siglas en inglés *General Image Manipulation Program*), siendo hoy en día denominado como GTK+ por sus siglas en inglés *GIMP ToolKit* y denominado

como un conjunto de bibliotecas multiplataforma para desarrollar interfaces gráficas de usuario el cual se basa principalmente en las librerías siguientes (<http://www.gtk.org/> , 2014):

- GLib: Una biblioteca de bajo nivel, base de GTK+, la cual proporciona el manejo de la estructura de los datos en lenguaje C, envoltorios de portabilidad e interfaces para funcionalidades en tiempo de ejecución como bucles de eventos, hilos, carga dinámica y objetos del sistema.
- Pango: Es una librería para el diseño y renderizado de textos, haciendo especial hincapié en la internacionalización. Es el núcleo para manejar las fuentes y el texto de GTK+.
- Cairo: Es una librería de gráficos 2D que soporta múltiples plataformas de salida, (incluyendo X Windows System y Win32), produce una salida consistente en todos los medios de comunicación al mismo tiempo que aprovecha la aceleración hardware de la pantalla cuando se encuentra disponible.
- ATK: Es una librería de que provee acceso a un conjunto de diversas interfaces. Al incluir la librería ATK en las interfaces, diversas aplicaciones y barras de herramientas tales como lectores de páginas, amplificadores y dispositivos de entrada alternativos.

GTK+ cuenta con la ventaja de que a pesar de ser descrito en lenguaje C, se encuentra desarrollado de modo que ofrece compatibilidad total con otros lenguajes diferentes como C++, Java, Python, Javascripts, Vala, Pearl, Pascal © ,por distinguir algunos de los más reconocidos, y provee compatibilidad con diferentes plataformas como GNU/Linux and Unix, Windows y Mac OS X. Además, GTK+ es un software libre el cual contiene una licencia GNU LGPL 2.1 (de sus siglas en inglés *Library General Public License*) que permite el uso de la herramienta por parte de desarrolladores, incluyendo de los que se encuentran desarrollando un software con derechos de propiedad intelectual, sin requerimientos de la adquisición de alguna licencia o pagos de regalías.

Entre las herramientas que se encuentran a la disposición de GTK+ se encuentran sets completos de widgets, menús, barras de tareas, selectores de color, tipos de fuente y tipos de archivos, despliegue de ventanas, etiquetas y barras de estado típicas de Windows y Linux,

entre otras herramientas, teniendo la capacidad de lograr interfaces como la mostrada en la Figura 2-29

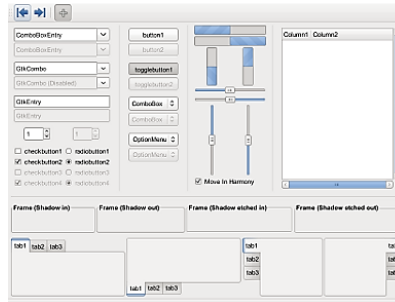


Figura 2-29 Imagen de muestra de las características disponibles en GTK+ (<http://www.gtk.org/> , 2014)

### 2.8.1 Gtkmm

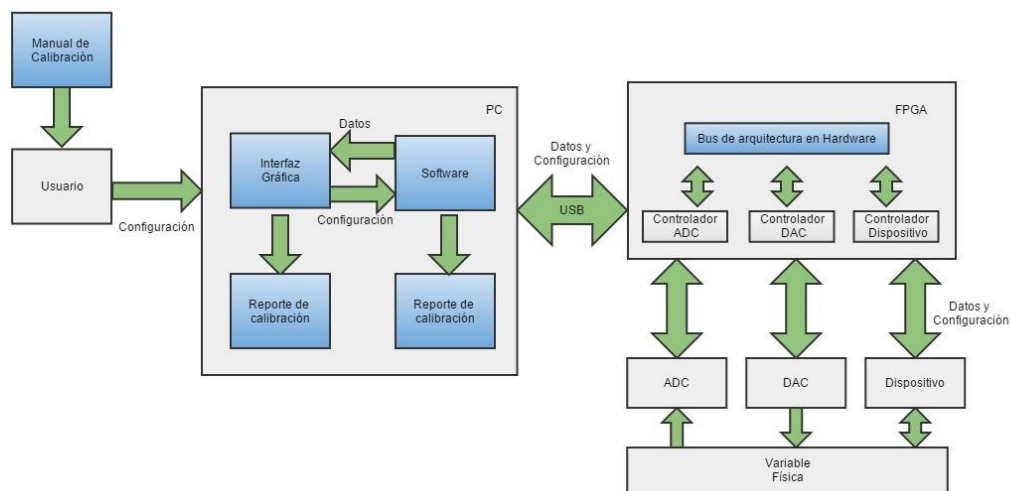
Gtkmm es la interfaz C++ oficial para la biblioteca de interfaz gráfica GTK+., dentro de sus características destacan las callbacks de tipo seguro, y un exhaustivo conjunto de widgets que son fácilmente extensibles a través herencia. Gtkmm permite la creación de interfaces de usuario ya sea con código o con el diseñador de interfaces gráficas Glade, usando Gtk::Builder. Gtkmm, al igual que GTK+ es un software libre distribuido bajo la Licencia Pública General Reducida de GNU (<http://www.gtkmm.org/en/> , 2014).

Las principales características de Gtkmm son:

- Compatible con múltiples plataformas; Linux (gcc), FreeBSD (gcc), NetBSD (gcc), Solaris (gcc, Forte), Win32 (gcc, MSVC++ .Net 2003, 2005, 2008), MacOS X (gcc), entre otras.
- Manejo completo de memoria utilizando el asistente de memoria en C++
- Uso de la biblioteca estándar C++, incluyendo cadenas, contenedores e iteradores.
- Manejadores de señales de tipo seguro, en C++ estandar.
- Usar herencia para derivar widgets personalizados.
- Uso completo de los espacios de nombres de C++.
- Internacionalización completa con UTF8.
- No maneja los macros, Es un Software Libre, Polimorfismo.

3. METODOLOGÍA

Para facilitar la descripción del desarrollo de este trabajo, el mismo se puede dividir en distintos módulos o bloques los cuales se encuentran sobresaltados dentro de la Figura 3-1 por medio del color azul. Es importante destacar que únicamente los bloques con color azul dentro de la Figura 3-1 son los que se desarrollarán durante este trabajo, ya que los demás módulos ya se encuentran realizados y únicamente se hará una adaptación de ellos para que trabajen de acuerdo al protocolo descrito en la arquitectura en hardware implementada (Bus de entradas y salidas), tales como los distintos controladores de los convertidores A/D y D/A incluidos dentro del diagrama.



**Figura 3-1 División del trabajo por áreas de desarrollo**

Las áreas de desarrollo son las siguientes:

1. Descripción en hardware de la arquitectura del bus de conexión abierta para dispositivos FPGA
2. Desarrollo del software de comunicación entre la PC y los dispositivos FPGA
3. Desarrollo de una interfaz gráfica de usuario para el software desarrollado
4. Desarrollo de un manual para el uso de la interfaz gráfica de usuario
5. Desarrollo de un manual de calibración apropiado para los convertidores D/A y A/D



El plan de trabajo empleado se ilustra dentro del diagrama de la Figura 3-2 siendo la identificación de las características de los convertidores con los cuales se va a trabajar típicamente la tarea de trabajo con mayor prioridad debido a que tanto la interfaz gráfica de usuario y el software de comunicación deben estar basados en las necesidades típicas de un proceso de adquisición actual y deben ser fácilmente adaptables para el uso de cualquier tarjeta que cuente con un sistema DAQ.

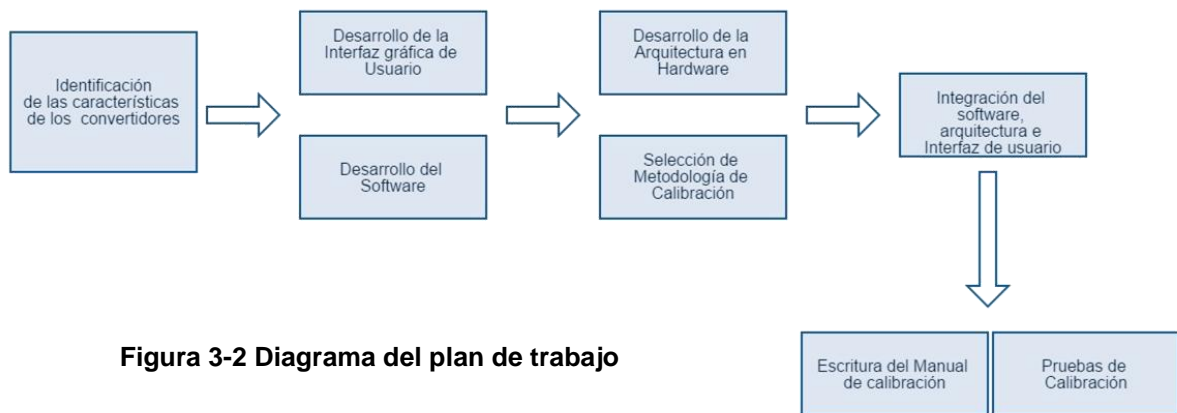


Figura 3-2 Diagrama del plan de trabajo

### 3.1 IDENTIFICACIÓN DE LAS CARACTERÍSTICAS DE CADA TARJETA FPGA Y SUS CONVERTIDORES ASOCIADOS

La tabla 3-1 muestra un resumen con las características de los convertidores con los cuales se planea trabajar durante el desarrollo de este trabajo, dichas tarjeas y sus convertidores fueron elegidos debido a que se encuentran en uso activo dentro de la escuela en distintos proyectos de investigación.

Tabla 3-1 Características de convertidores grupo HSP Digital

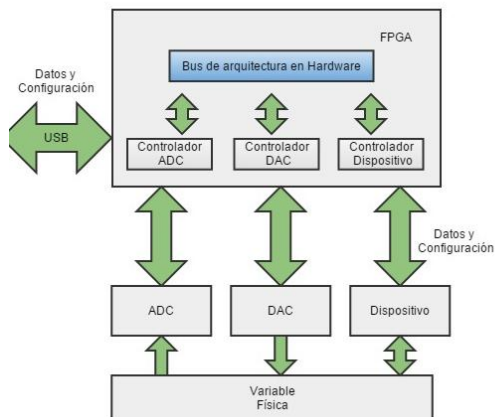
Característica de la Tarjeta	DUA-I	UPDSH	DAS1612 (a) (b)	MCUAQ4X	PLCUAQ816
Canales ADC	4	---	4	Controlador ADC y DAC	Controlador ADC y DAC
Canales DAC	1	---	4	---	---
Resolución ADC	8-12 bits	---	8-12 bits	---	---
Resolución DAC	12 bits	---	12 bits	---	---
Fsampling ADC	50 Ksmp/s	---	50 Ksmp/s	---	---
Fsampling DAC	xxxx	---	100 Ksmp por canal	---	---
Memoria estática	256 Kb	4 Mb	---	512 KB	512 KB
Memoria dinámica	256 Kx16	256Mb	---	4 Mb	4 Mb

De la Tabla 3-1 se realizaron las siguientes observaciones, las cuales servirán como restricciones durante el desarrollo del trabajo:

- ✓ Las resoluciones de trabajo típicas son de 8 y 12 bits.
- ✓ El número de canales de los convertidores actuales no excede 8 tomando en cuenta tanto convertidor A/D como convertidor D/A.
- ✓ La memoria de cada tarjeta tiene capacidades de almacenamiento diferentes y utilizan drivers específicos para cada tipo de memoria.
- ✓ La frecuencia de trabajo del ADC y del DAC son independientes entre sí.

### 3.2 DESCRIPCIÓN EN HARDWARE

#### 3.2.1 Comunicación Interna



**Figura 3-3 Bloque de bus genérico**

La Figura 3-3 muestra el bloque de tareas respectivo al desarrollo de la arquitectura de hardware. Dicho bloque de hardware se puede describir como un bus de control de entradas y salidas el cual es capaz de admitir nuevos dispositivos o bloques de HDL para someterse bajo su control principal, dicha flexibilidad es la razón de su denominación *bus de arquitectura abierta*.

El desarrollo del bus de arquitectura abierta se desarrolló basándose en el protocolo de comunicación CAM, el cual es utilizado en la industria automotriz para establecer comunicación desde un sistema general hacia sus sistemas inferiores o bloques embebidos. El manejo de la transmisión de datos se realiza únicamente a través del bus de conexión general, y se lleva a cabo utilizando tres diferentes tipos de bloques.

Esclavo

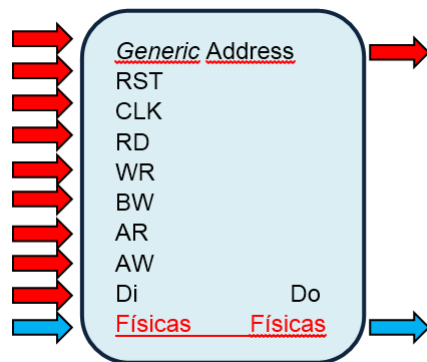


Figura 3-4 Bloque de tipo Esclavo

Al utilizar el bus de conexión general cada uno de los bloques típicos de HDL debe ser transformado en un bloque tipo esclavo para su uso dentro del bus.

Un bloque tipo esclavo contiene un campo *Address*, el cual indica la dirección única a partir de la cual se asignarán cada una de sus entradas y salidas dentro del bus. En otras palabras, cada campo del bloque original en HDL (*in, out, inout*) tiene una dirección designada a partir de la cual puede ser accesada una vez que se encuentre embebido dentro del bus, tanto para modo de lectura como para el modo de escritura. La señales *RD* y

*WR* indican precisamente el modo hacia el cual se ha llamado al bloque, cuando *RD=1*, significa que se va a realizar una lectura, mientras que cuando *WR=1*, indica que se realizará una escritura en la dirección especificada por *AR* y *AW* respectivamente (la señal *AR* indica la dirección a la cual se desea acceder cuando se ha llamado al bloque en el modo de lectura (*RD='1'*), mientras que la señal *AW* indica la dirección a la cual se desea acceder cuando se ha llamado al bloque en el modo de escritura (*WR='1'*)).

Los datos contenidos dentro del bus son de un ancho de 16 bits, siendo la variable de dos bits *BW* la encargada de indicar el rango de bytes a los cuales se les realizará alguna operación (ya sea de lectura o escritura), el bit más significativo de *BW* controla los 8 bits más significativos, mientras que el bit menos significativo de *BW* controla los 8 bits menos significativos de los datos. Cuando cualquier bit de *BW* sea=1, indica que se realizará la operación de lectura o escritura de las secuencias de bit que se maneje (MSBs o LSBs) mientras que cuando un bit de *BW*=0, indica que no se realizará ninguna operación al conjunto de bits hacia el cual se encuentre asociado dicho bit

Los bloques de tipo esclavo únicamente contienen una señal de entrada y una de salida, las cuales están denotadas como *Di* y *Do* dentro de la Figura 3-4. *Di* es el dato que un maestro

desea escribir, mientras que *Do* es el dato al cual se desea acceder igualmente desde un bloque del tipo maestro, teniendo en cuenta los modos del bloque establecidos por *RD*, *WR*, *AR* y *AW*.

Debido a que el uso de terminales físicas es frecuentemente necesario dentro de los bloques en HDL, cada una de estas entradas físicas se incluyen dentro del bloque Esclavo, para ser ruteadas posteriormente hacia el lugar correcto durante la implementación del programa.

### Maestro-Esclavo

Un bloque del tipo maestro-esclavo incluye un campo *Address* el cual funciona exactamente de la misma manera que en el bloque del tipo esclavo, incluye una señal *BR* (*Bus Request*) la cual sirve como medio de comunicación con el árbitro dentro del bus. Cuando *BR*=1 significa que el bloque del tipo maestro-esclavo está pidiendo acceso al control del bus, ya sea para realizar una escritura o una lectura de cualquiera de los bloques esclavos o Maestros-esclavos conectados al bus. Cuando *BR*=0 significa que el bloque Maestro-esclavo no pretende seguir con el control del bus, y que por lo tanto, el mismo se encuentra disponible para ser utilizado por algún otro bloque Maestro-esclavo.

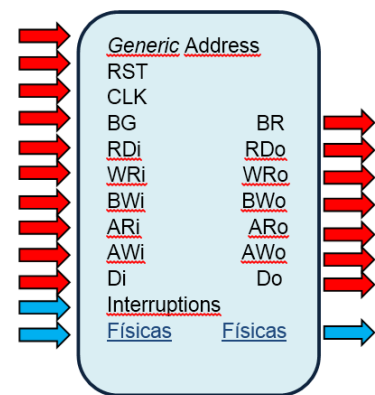


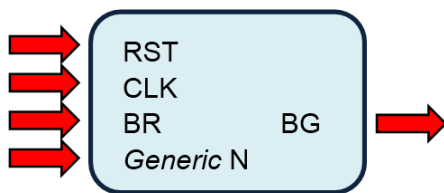
Figura 3-5 Bloque de tipo Maestro Esclavo

La señal *BG* (*Bus Granted*) es la respuesta del árbitro ante la petición del Maestro, siendo *BG*=1 equivalente a un control del bus cedido, y un *BG*=0 equivalente a un control del bus denegado. En caso de que *BG*=1 y se tenga control del bus la operación que se realizará el maestro-esclavo estará definida de forma similar que con el bloque esclavo, siendo las señales *RDo*, *WRo*, *BWo*, *ARo*, y *AWo* las encargadas de describir la operación que va a realizar sobre un esclavo conforme a lo explicado dentro de la sección del bloque esclavo, Y las señales *RD<sub>i</sub>*, *WR<sub>i</sub>*, *BW<sub>i</sub>*, *AR<sub>i</sub>*, y *AW<sub>i</sub>* *AWo* las encargadas de describir la operación que va a realizar algún otro Maestro-esclavo sobre el bloque Maestro-esclavo. He aquí la derivación de su nombre, pues el bloque funciona como modo Maestro cuando se le ha concedido el control del bus, y funciona como un esclavo más cuando no posee el control del bus de información.

Debido a que los bloques Maestro-esclavo son realizados con el fin de realizar alguna especie de control sobre los bloques esclavos conectados al bus, todo bloque Maestro-esclavo tiene como opción agregar señales de interrupción muy útiles las cuales entrarán directamente al bloque sin la necesidad de pasar a través del bus de comunicación.

Además, al igual que todo bloque esclavo, el bloque maestro-esclavo contiene campos tanto de entrada como salida, para realizar un ruteo de sus conexiones físicas en caso de que estas existan.

### Árbitro

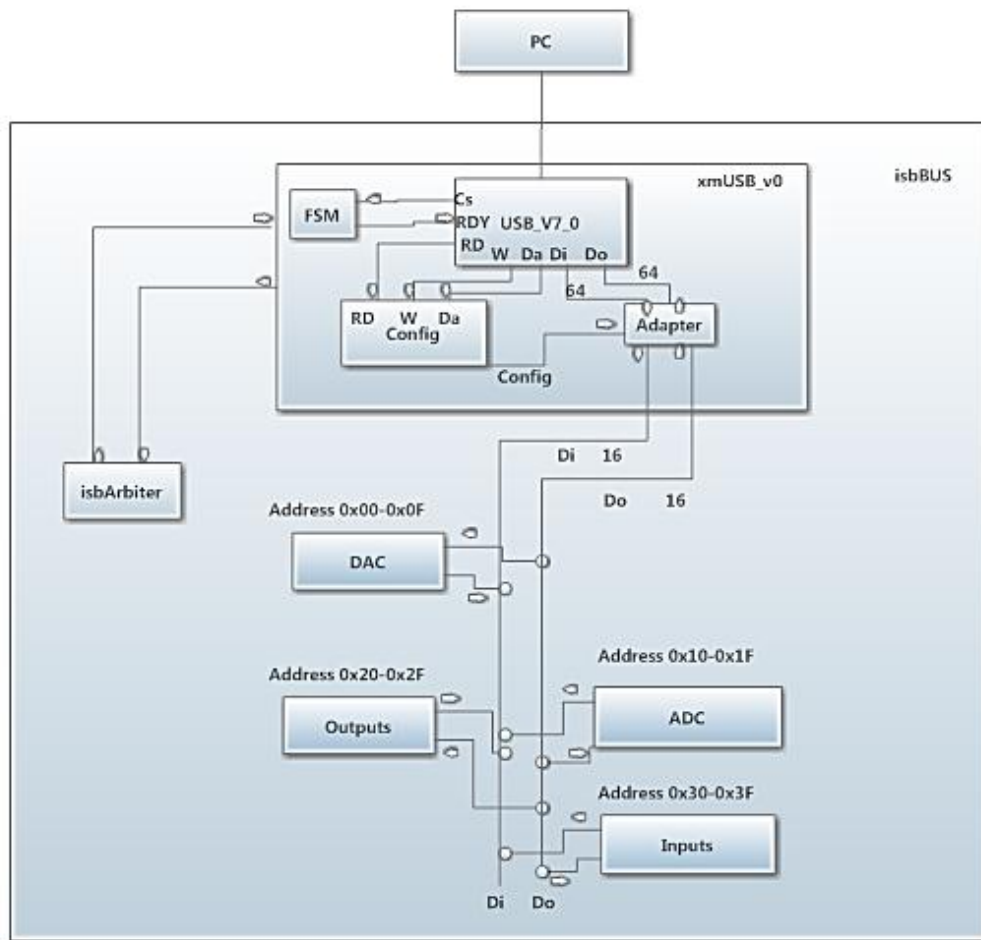


El bloque de árbitro es mostrado en la Figura 3-6 es uno de los más importantes debido a que es el que se encarga de determinar si un maestro que ha pedido el control del bus puede tomarlo, o no. Sus campos son *BR*, la cual recibe las señales de *Bus Request* de los *N* Maestros-esclavos conectados al bus, y *BG* el cual es un vector que contiene las respuestas ante las peticiones de

**Figura 3-6 Bloque de tipo árbitro**

cada uno de los maestros esclavos conectados al bus.

A continuación la Figura 3-17 muestra un esquema general del bus de arquitectura abierta (*isbBUS*) con los bloques maestro-esclavo de los convertidores A/D y D/A conectados a él.



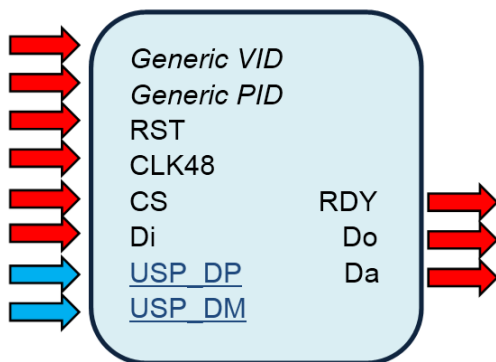
**Figura 3-7 Esquema General del Bus de arquitectura abierta**

Como se puede ver en la Figura 3-7 el control de los datos de entrada o salida dependen de la dirección de los comandos obtenidos de la comunicación externa sostenida desde la FPGA hasta la PC por medio del driver de comunicación USB.

3.2.2 Comunicación Externa

La comunicación de datos externa, es decir, la que se lleva a cabo entre la PC y la arquitectura en hardware se lleva a cabo por medio de un bloque en hardware de comunicación USB el cual fue convertido en un bloque del tipo Maestro-Eslavo, y fue incluido dentro del bus de conexión general. La Figura 3-8 muestra el bloque *usbUSB\_v7\_0* en el cual se basó para concretar la comunicación USB.

Se debe clarificar que este bloque no fue desarrollado durante este trabajo, sino que ya se encontraba realizado y únicamente fue empleado para los fines de este trabajo.



**Figura 3-8 Bloque de comunicación USB**

Cuando la computadora se encuentra enviando un dato hacia el dispositivo FPGA se utiliza el canal *Do* del USB, y en caso de recibir un dato desde el dispositivo se utiliza el canal *Di* incluido en el mismo USB.

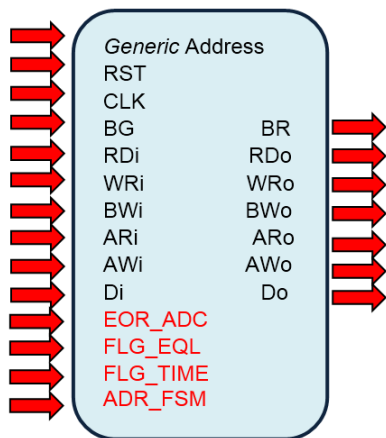
Para ayudar a comprender el uso de los canales *Di*, *Do* y *Da* uno se puede basar en la Figura 3-7 donde se muestra la arquitectura en hardware implementada. Si se observa desde el nivel de jerarquía más bajo, se observan los bloques o módulos esclavos en VHDL de los controladores pertenecientes ya sea a un convertidor A/D, o a un convertidor D/A. Cada uno de dichos bloques tiene asignado un rango de direcciones escrito en código hexadecimal en la parte superior de su bloque (*Address*), los cuales pertenecen a las salidas o entradas del módulo controlador.

Debido a que se requiere tener control sobre todos los módulos desde la PC, ya sea de entrada o salida, es decir su dirección *Address*, se hace uso del canal *Da*. Por ejemplo, en el caso del ADC, el cual contiene 16 puertos de entradas o salidas, se necesita 16 direcciones para control, una dirección perteneciente a cada puerto, es por ello que su rango de direcciones comienza desde 0x10 hasta 0x1f, es decir 16 direcciones, y en caso de querer acceder a alguna de sus características basta con enviar la dirección deseada por medio del campo *Da* desde la PC.

En el caso de que se requiera agregar un nuevo elemento para ser controlado, basta con asignarle un rango de direcciones que no se encuentre ya ocupado por otros bloques, para poder de esta manera controlar las variables desde la PC. Es por ello que la Figura 3-7 contiene un bloque simbólico llamado *inputs*, y un bloque simbólico llamado *outputs*, como representación de que se pueden agregar todo tipo de nuevos dispositivos o bloques al bus generado por la arquitectura existente, lo cual le ha dado al bus la denominación de genérico.

### 3.2.3 Máquina de estados finitos para el control de entradas y salidas

Para controlar los convertidores, se realizó una Máquina de estados finitos (FSM por sus siglas en inglés *Finite State Machine*) la cual es capaz de interpretar una señal de inicio desde la PC y realizar la adquisición de acuerdo a los parámetros configurables del convertidor desde la PC por el usuario.



**Figura 3-9 Bloque de la FSM de control del ADC**

En la Figura 3-9 se muestra el bloque de la FSM de control del ADC, la cual no es más que un bloque del tipo Maestro-esclavo con cuatro interrupciones o entradas específicas, el *EOR\_ADC* es la señal que se recibe desde el driver del convertidor A/D el cual indica que se ha completado la adquisición de una muestra, *FLG\_EQL* es una bandera que controla el número de datos leídos hasta el momento y únicamente se activa en caso de que se hayan leído el número de datos configurado por el usuario, la bandera *FLG\_TIME* es una bandera la cual se activa únicamente cuando ha transcurrido el tiempo de muestreo deseado por el usuario, y *ADR\_FSM* es una cadena de bits la cual proviene de un contador ascendente el cual direcciona el

bloque Maestro-esclavo para leer un dato previamente escrito dentro de la memoria estática SRAM se muestra el diagrama conceptual de la FSM.



Como se puede ver en la Figura 3-10 cada uno de los datos recopilados son almacenados dentro de la memoria SRAM, la cual es un bloque esclavo dentro del bus, y la cual sirve posteriormente como medio de consulta de datos por el usuario desde el software.

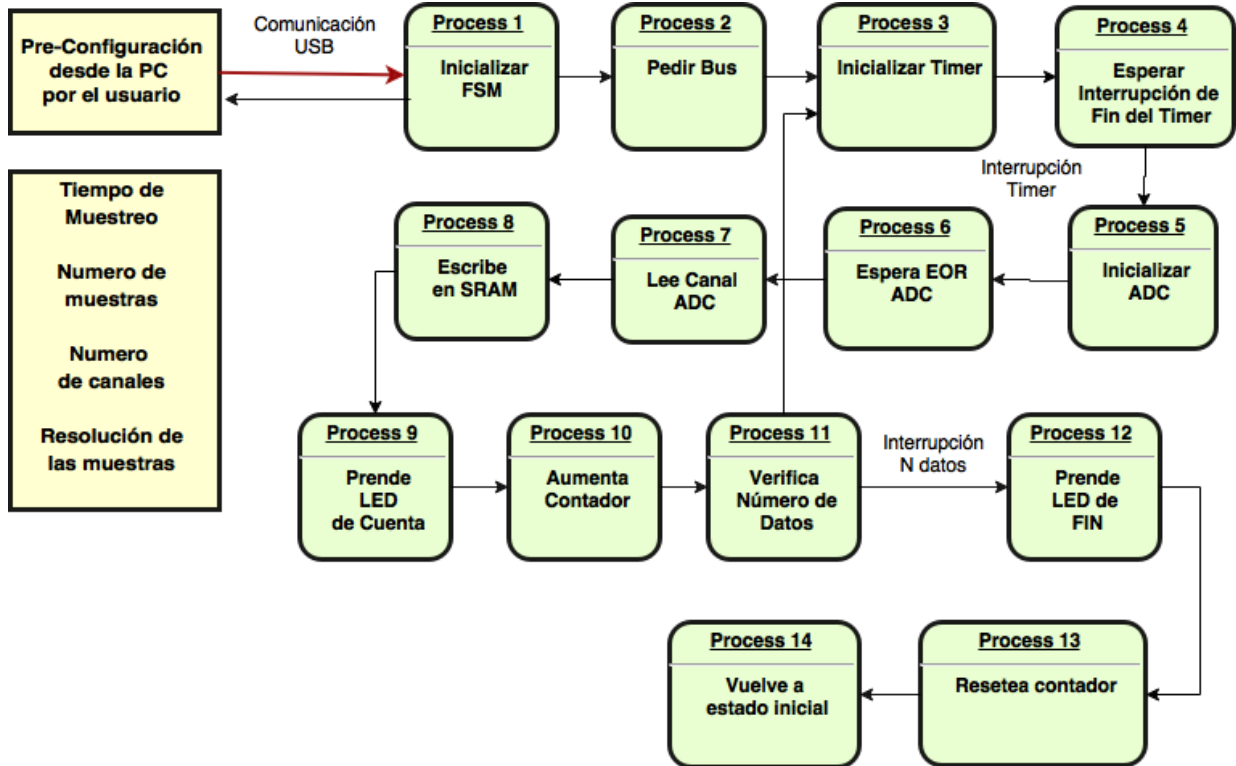


Figura 3-10 FSM de adquisición de datos

### 3.3 DESARROLLO DE UNA INTERFAZ GRÁFICA DE USUARIO

Para facilitar el manejo y visualización de los datos se basó en el instrumento de visualización gráfica de datos más común; el osciloscopio. Un osciloscopio típicamente tiene la capacidad de manejar cursores de voltaje y tiempo para poder realizar mediciones.

### 3.3.1 Menú general de configuración

El menú general de configuración es el menú principal de la interfaz gráfica, y el menú con el que el usuario que planea adquirir datos debe estar más familiarizado, dentro de este menú mostrado en la Figura 3-11 se pueden encontrar las diferentes áreas o submenús que se desarrollaron para aplicaciones más específicas que se explicarán a continuación a más detalle.

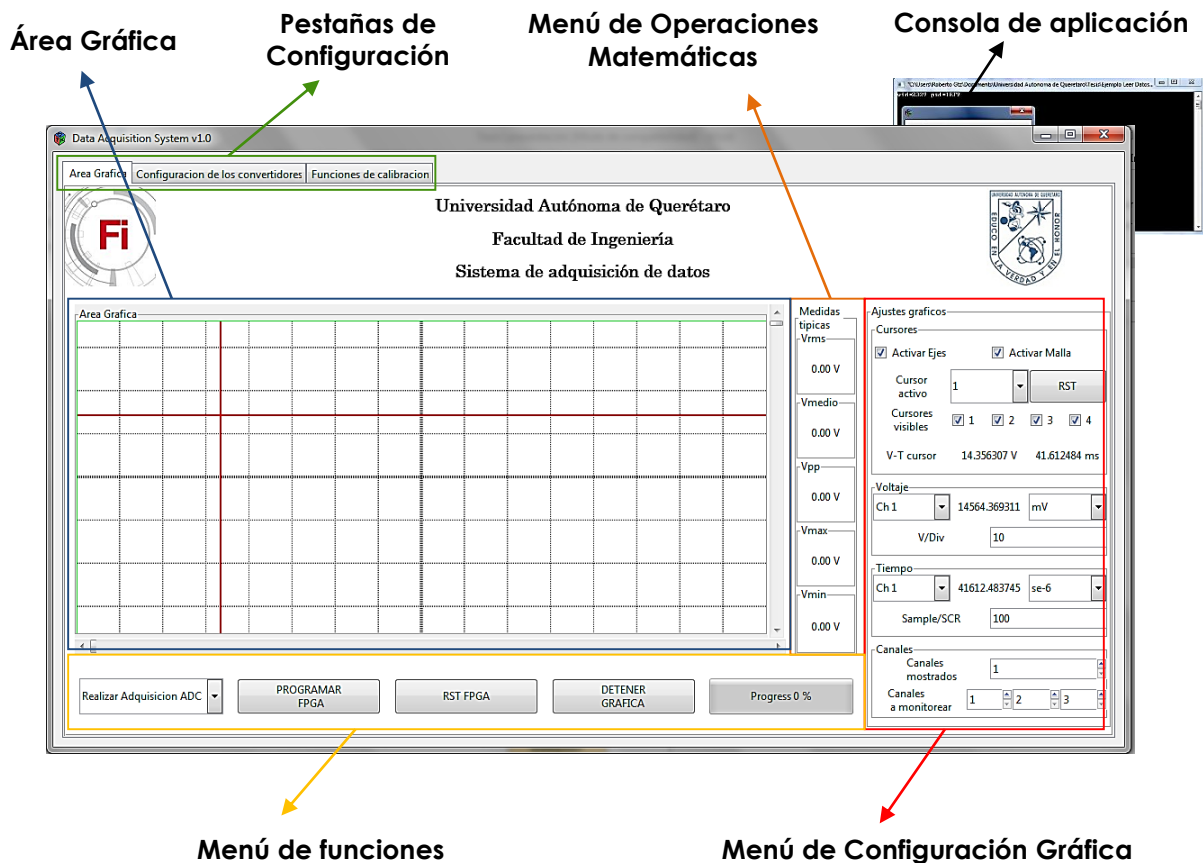


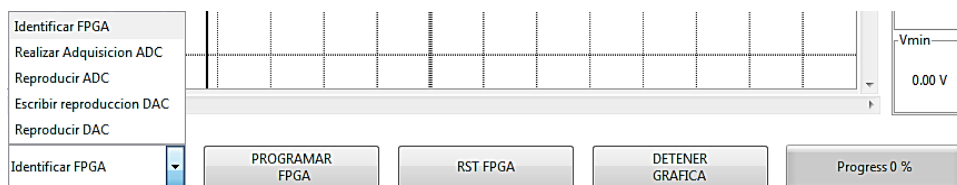
Figura 3-11 Pantalla principal de la Interfaz Gráfica

### 3.3.2 Área gráfica

Se desarrolló un área destinada a gráficos utilizando las herramientas *Cairo* de dibujo incluidas dentro de la librería *Gtkmm*. Para graficar dentro de esta área se traducen los valores de lectura de números en formato *FLOAT*, para realizar una conversión apropiada (descrita por el usuario en el Menú de configuración Gráfica) al número de pixeles equivalente que representa dicho número.

El sistema de graficado se basa en llevar dentro de una clase tipo *list* de C++ un registro con los valores de voltaje pertinentes para realizar el bosquejo dentro del área gráfica, los cuales se mandan a graficar a través de un hilo *Thread* mientras que se ejecuta la ventana para evitar que se la interfaz gráfica se trabe y el programa deje de funcionar. Además, debido a que se tiene un espacio limitado en pixeles para el área gráfica, se delimita un número *NdataSCR* el cual delimita el número de muestras máximas a visualizar en la pantalla y por lo tanto tener en la *list* de gráficos. Una vez que se haya sobrepasado este límite de número de datos, la clase *list* se comportará como un dispositivo *FIFO* (de sus siglas en inglés *Finite Input Finite Output*) el cual desechará el primer dato ingresado en el registro, creando de esta manera espacio para que se ingrese un nuevo dato, completando de esta manera el proceso de bosquejado.

### 3.3.3 Menú de funciones



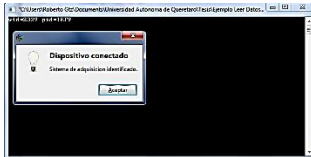
**Figura 3-12 Menú de Funciones**

El Menú de funciones de la GUI contiene los botones necesario para establecer una comunicación entre el dispositivo FPGA desde el software, contiene las siguientes funciones

- ✓ Identificar FPGA

Establece la primera comunicación con el FPGA con el fin de identificar la tarjeta con la cual se está trabajando, para poder de esta manera cargar los archivos de calibración respectivos con los que se cuenta, y actualizar los límites analógicos de la tarjeta automáticamente. En

caso de no encontrar una tarjeta, despliega el respectivo mensaje dentro de la consola de aplicación mostrada en la Figura 3-12 avisando que existe un problema de conexión, y por lo tanto no proseguirá a realizar alguna otra acción, es decir las deshabilita hasta que se encuentre una tarjeta FPGA.



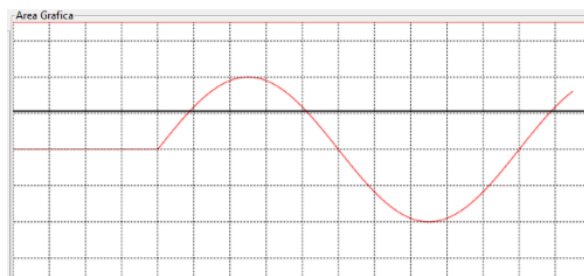
**Figura 3-13 Mensajes de consola**

✓ Realizar Adquisición ADC

La función de realizar adquisición implementa básicamente dos funciones, una en la cual se envía la configuración del usuario especificando parámetros como periodo de muestreo, Número de muestras y resolución de las muestras, y una vez enviada la configuración inicializa la FSM de adquisición de datos mostrada en la Figura 3-11.

✓ Reproducir ADC

Lee los datos que han sido almacenados dentro de la memoria SRAM con el fin de visualizarlos dentro del área gráfica como se muestra en la Figura 3-14



**Figura 3-14 Reproducción de ADC en el área gráfica**

✓ Escribir reproducción DAC

Lee los datos provenientes de un archivo .TXT, para interpretarlos y escribirlos en la memoria SRAM contenida dentro de la tarjeta FPGA. Una vez que se han escrito los datos deseados por el usuario se puede proceder a reproducir el convertidor D/A para reproducir

dicha lista de datos cada vez que se desee. Es decir, basta con escribir la reproducción del DAC una vez para reproducirla las veces que el usuario lo requiera

✓ Reproducir DAC

Una vez que se cuente con un archivo de reproducción DAC grabado dentro de la memoria SRAM, este conjunto de datos puede ser reproducido por el usuario cada vez que así se requiera.

Al oprimir el botón de reproducir DAC se manda un comando a la FPGA que habilita la FSM de reproducción del DAC, mientras que al mismo tiempo se reproduce la misma lista de datos dentro del área gráfica. La Figura 3-15 muestra el monitoreo por medio de un osciloscopio de una de las formas de onda reproducidas por medio del convertidor DAC de la DS1612a.

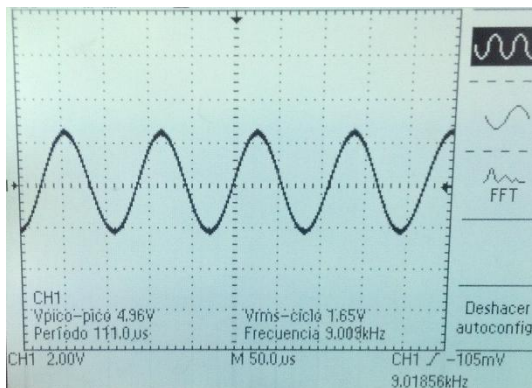


Figura 3-15 Señal reproducida por el DAC

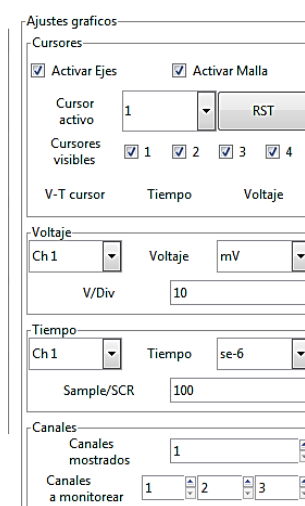


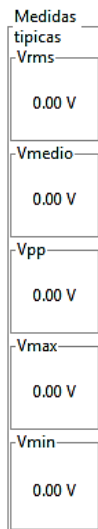
Figura 3-16 "Menú de Configuración Gráfica"

3.3.4 Menú de configuración gráfica

Se creó un menú de configuración gráfica (Figura 3-16) para auxiliar al usuario en la labor de personalizar el área de gráficos de acuerdo a sus necesidades, dentro de este menú se incluyeron etiquetas las cuales están siendo actualizadas cada milisegundo por medio de un *Timer* que realiza ciertas funciones parecidas a la del hilo gráfico debido a que se utiliza para evitar que se trabe la GUI mientras se concentra en la tarea de actualizar las etiquetas

Dentro de este menú se incluyen funciones que por medio de banderas emitidas por el widget *CheckButton* permiten la habilitación o deshabilitación del bosquejo de las mallas tipo osciloscopio. Además, debido a la utilidad de cursores de voltaje y de tiempo dentro del monitoreo de señales utilizando los osciloscopios reales, se realizaron 4 registros para guardar las coordenadas del cursor (Voltaje y tiempo) donde se ubica el usuario dentro de la pantalla, para consultar cualquiera de estos cursores basta con utilizar las opciones desplegadas dentro del widget *ComboBoxText*, y el valor solicitado se desplegará dentro de la etiqueta pertinente.

### 3.3.5 Pestaña de operaciones aritméticas



Se realizó una barra lateral (Figura 3-17) a un lado del área gráfica la cual permite visualizar medidas básicas de las señales adquiridas, tales como el voltaje rms, el voltaje medio, el voltaje pico-pico, el voltaje máximo y el voltaje mínimo.

Dichas medidas de voltaje son calculadas a través del registro o *list* de gráfico que se mencionó previamente, y debido a que con cada muestra nueva ingresada al sistema dichos valores cambian, el cálculo de las medidas y su posterior despliegue en la interfaz gráfica se realizaron utilizando nuevamente un hilo *ThreadLabel* el cual se encarga de desplegar las actualizaciones de los valores sin que existan errores en el programa. Es importante mencionar que

**Figura 3-17**  
**"Pestaña de operaciones matemáticas"**

para que no exista interferencia entre los hilos utilizados, se debe de utilizar la clase *mutex*, o en su defecto alguna de las propiedades de los hilos como el *Join*.

### 3.3.6 Menú de calibración del convertidor

#### Configuración de los convertidores

Configuracion de los convertidores	
Vmin (V)	-10.000000
Vmax (V)	10.000000
Frec (Hz)	500.000000
Frec (MHz)	48
Formato	12 bits
Numero Muestras	1000
Numero de canales	1
Guardar ADC	ADCSpartan.bt
Cargar ADC	ADCSpartan.bt

Antes de realizar cualquier procedimiento de calibración, se ajustaron los parámetros bajo los cuales se iba a realizar la adquisición de datos. Dicha tarea se logra fácilmente utilizando el Menu de configuracion de los convertidores mostrado en la Figura 3-18.

Como se puede ver los datos de configuración son ingresados desde el GUI por el usuario desde un widget tipo *Entry*, para ser procesados posteriormente como cadenas *string*, y ser enviadas al dispositivo FPGA por medio de la comunicación USB.

**Figura 3-18 Menú de configuración de los convertidores**

#### SubMenú de Offset

Offset	
Promedio (mV)	Sin Calibrar
Varianza (mV*mV)	Sin Calibrar
Desviacion Estandar (mV)	Sin Calibrar
Error Offset (mV)	Sin Calibrar
<input type="checkbox"/> Guardar Datos	Offset.txt
Calibrar ADC	Offset 0 %
Resultado calibracion	X

Para realizar la calibración de Offset, se desarrolló un Submenú propio (Figura 3-19) para este tipo de calibración, en dicho menú se muestran las medidas típicas recabadas del procesamiento de los datos adquiridos por el convertidor, tales como Promedio, Varianza, Desviación Estándar y la cuantificación del Error de offset en mV

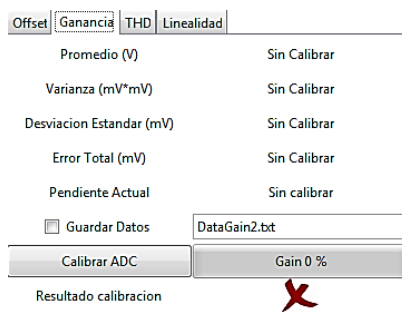
**Figura 3-19 Menú de la calibración de Offset**

Debido a que en casos de adquirir una gran cantidad de muestras el proceso de calibración es tardado se incluyo una barra de progreso, la cual le muestra al usuario

precisamente el progreso de la calibración. Dicha barra fue programada nuevamente utilizando un *Timer* para que funcionara correctamente y mostrara el progreso de calibración en Tiempo real al actualizarse cada 1 ms.

También existe un campo para que en caso de que el usuario desee guardar los datos a los cuales se les hizo el proceso de calibración se pueda realizar en un archivo de formato .txt, Y finalmente se incluyó una imagen que muestra el resultado de la calibración, mostrando una cruz roja cuando el dispositivo no se encuentra calibrado, y una palomita verde cuando el sistema de adquisición de datos ha sido calibrado.

### SubMenú de Ganancia



El menú de calibración de ganancia (Figura 3-20) es muy parecido al menú de calibración de offset, teniendo exactamente los mismo elementos que el anterior, pero con la diferencia de que en este tipo de error es importante identificar la pendiente actual de la función de transferencia del convertidor, y por lo tanto, es un dato

**Figura 3-20 Menú de la calibración de ganancia**

que es mostrado dentro de las etiquetas.

### SubMenú de Distorsión armónica Total THD



Se realizó un submenú específico para la calibración de THD mostrado en la Figura 3-21. En dicho menú se muestra el resultado de la calibración, es decir el índice de THD, tanto en porcentaje como en dB, la frecuencia fundamental detectada por el análisis de Fourier, y una pestaña desplegable la cual muestra al

**Figura 3-21 Menú de THD**



usuario los 5 Armónicos con magnitudes más grandes dentro del espectro de Fourier contenidas dentro de la señal.

Al igual que en los menús anteriores se incluye la opción de guardar los datos a los cuales les fue realizado el procesamiento digital, una barra de progreso con el estatus actual de la calibración, y un icono el cual representa que el convertidor ha pasado o no ha pasado la prueba de distorsión armónica basándose en los criterios descritos en el marco teórico.

### SubMenú de Linealidas

La diferencia fundamental del submenú de linealidad (Figura 3-22) con los otros submenús, es que para el caso de la calibración de linealidad, se incluyó una lista con todos los códigos perdidos que se han detectado como resultado del análisis del DNLE, teniendo dicha lista la capacidad de ser interpretada por el usuario en los formatos *INT*, *HEX* y como su correspondiente valor de voltaje dentro de la resolución del convertidor.

Figura 3-22 Menú de linealidad

## 3.4 METODOLOGÍA DE CALIBRACIÓN

El manual de calibración se realizó suponiendo en todo momento que se realizará utilizando el software y la interfaz gráfica perteneciente a este trabajo, y además, que el usuario se encuentra familiarizado con el marco teórico de los errores de un convertidor incluido dentro del Capítulo 2.

Su contenido está limitado a la serie de pasos que se siguieron para obtener cada uno de los siguientes errores:

3.4.1 Error de Offset

El error de offset es el primer error que se debe de cuantificar y de corregir, ya que la medición de los demás errores puede verse mermada en caso de que este error aun no haya sido corregido.

$x_i = \text{Número de muestra}; n = \text{Número de datos}$   
 $\bar{x} = \text{Media aritmética} \quad \sigma^2 = \text{Varianza poblacional}$

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \quad \text{Ec 3.1}$$

Para medir el error de offset se requiere de seguir el diagrama de pasos ilustrado en la Figura 3-19

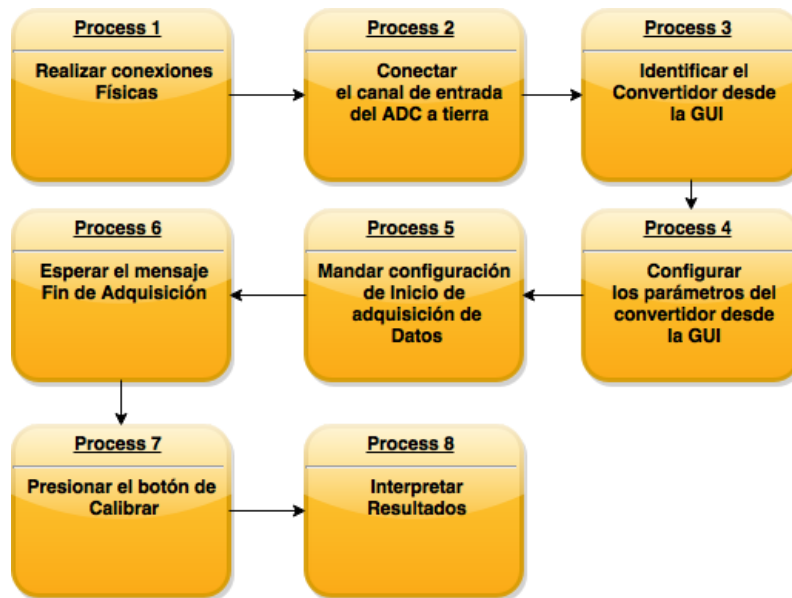


Figura 3-23 "Diagrama de flujo de la obtención del error de offset"

Una vez adquiridos los datos se obtuvieron algunas medidas de dispersión como el promedio, desviación estándar y la varianza mostrada en la Ecuación 3-1.

3.4.2 Error de Ganancia

El error de ganancia debe ser calculado exclusivamente una vez que se haya corregido el error de offset. Teniendo esto en cuenta el error de ganancia se determinará utilizando la ecuación de la recta determinada por dos puntos Ecuación 3.1 a) y 3.2 b) siguiendo el procedimiento descrito en la Figura 3-24.

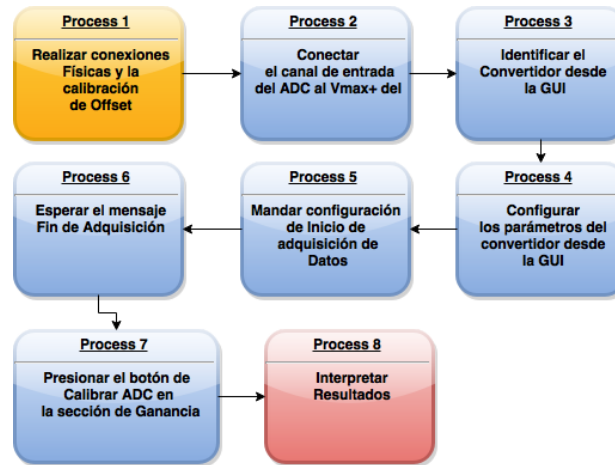


Figura 3-24 Diagrama de flujo de la obtención del error de ganancia

$$y - \bar{y}_1 = \frac{\bar{y}_2 - \bar{y}_1}{\bar{x}_2 - \bar{x}_1} (x - \bar{x}_1); \text{ Ec. 3.2 a) } \quad \bar{m}_{\text{real}} = \frac{\bar{y}_2 - \bar{y}_1}{\bar{x}_2 - \bar{x}_1} \text{ Ec. 3.2 b)}$$

3.4.3 Pruebas de linealidad diferencial

Una vez obtenida corregidos los errores de Offset y de Ganancia, se puede proceder a calcular el error de linealidad diferencial conforme al procedimiento descrito en la Figura 3-25.

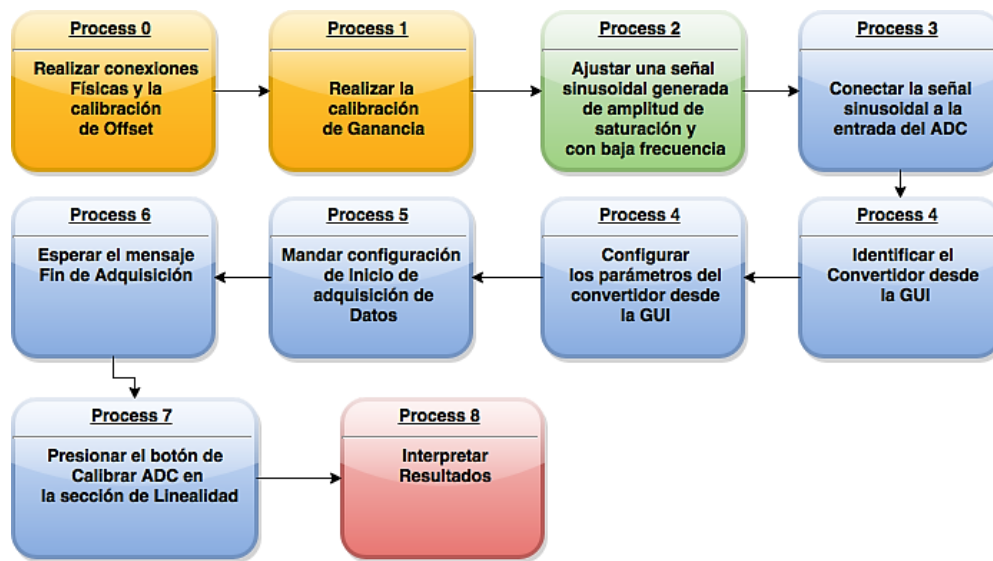


Figura 3-25 Diagrama de flujo de obtención del DNLE

### 3.4.4 Total Harmonic Distorsion

Este análisis se llevará a cabo mediante la implementación directa de la transformada rápida de Fourier (FFT por sus siglas en inglés *Fast Fourier Transform*) una vez que se haya adquirido alguna señal periódica por medio del ADC, para posteriormente realizar un cálculo del índice de calidad THD por medio de la Ecuación 2.9. El resultado que arrojará el reporte consistirá simplemente en el índice THD y se emitirá una alerta de reemplazo de convertidor en caso de que la distorsión sea mayor al 1%.

#### Requisitos para obtener un análisis confiable utilizando la DFT

- ✓ Número de ciclos adquiridos debe ser un número primo el cual no sea un submúltiplo del número total de datos adquiridos
- ✓ Frecuencia de muestreo igual o mayor a la Frecuencia de Nyquist
- ✓ Frecuencia de muestreo en el rango del convertidor
- ✓ Frecuencia de muestreo no es algún múltiplo entero de la Frecuencia fundamental
- ✓ Número de puntos de la DFT o FFT no es un múltiplo de la Frecuencia fundamental

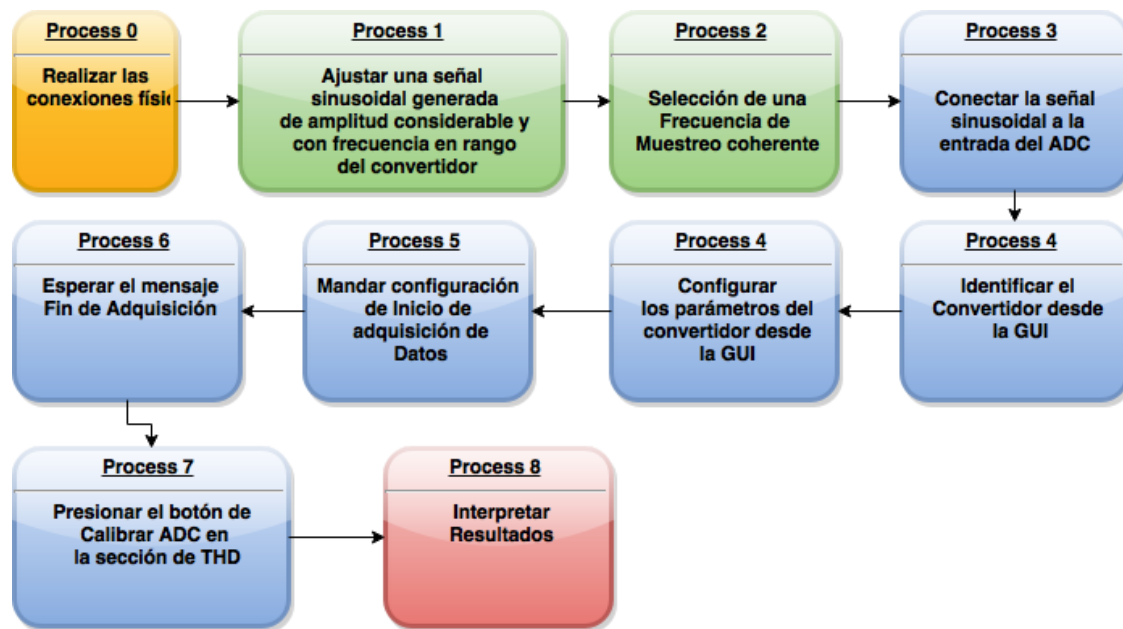


Figura 3-26 Diagrama de Flujo de obtención del índice de THD

### 3.5 DESARROLLO DE UN MANUAL PARA EL USO DE LA INTERFAZ GRÁFICA

El manual para uso de la interfaz contendrá guías ilustradas acerca del correcto uso y la información adicional pertinente para cada uno de los siguientes módulos:

- Consola de Aplicación
- Menú de configuración General
- Menú de configuración de los convertidores
- Menú de configuración gráfica
- Descripción del área gráfica

Dicho manual se encuentra anexo dentro de la sección perteneciente al Apéndice ubicada al final del presente trabajo.

4. RESULTADOS Y DISCUSIÓN

4.1 DETERMINACIÓN DEL ERROR DE OFFSET

Se realizaron pruebas tanto del convertidor de DAS1612 utilizado por las tarjetas Spatran-3 y la PLCUAQ, .Dentro de la Figura 4-1 se muestra el resultado de la prueba de offset realizada sobre el convertidor DAS1612 de la Spartan-3, siendo resaltado por medio del color rojo el lugar ideal donde debería de estar la medida del convertidor, y bosquejada por medio de una línea azul las mediciones reales recabadas. Las medidas de dispersión recabadas se muestran dentro de la Figura 4-2

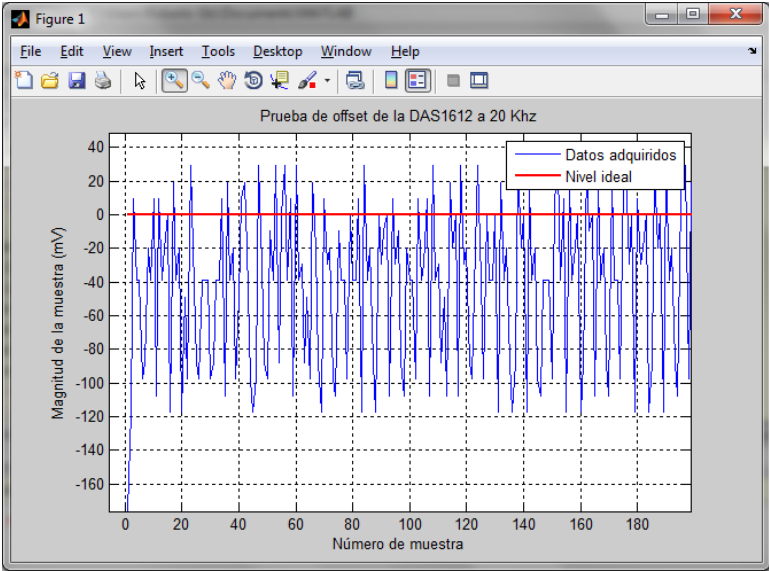


Figura 4-1 Error de Offset de DAS1612

Offset	Ganancia	THD	Linealidad
Promedio (mV)			-36.222171
Varianza (mV*mV)			2277.119551
Desviacion Estandar (mV)			47.719174
Error Offset (mV)			-36.222171

Guardar Datos      Offset.txt

Calibrar ADC      Offset 100 %

Resultado calibracion      ✓

Figura 4-2 Pantalla de Calibración del error de Offset

### 4.2 DETERMINACIÓN DEL ERROR DE GANANCIA

La Figura 4.2 muestra los resultados de la adquisición de datos durante la prueba de ganancia, así como la Figura 4-4 muestra las medidas de dispersión encontradas. Como se puede apreciar, a pesar de que se esperaba una lectura de la escala completa del convertidor, es decir cercana a los 10V, se obtuvieron mediciones promedio muy por debajo de lo estimado, y con una desviación estándar muy baja, lo cual probablemente indica que el convertidor llega a una zona de saturación antes de lo previsto, el cual origina el error de ganancia apreciado dentro de estos resultados.

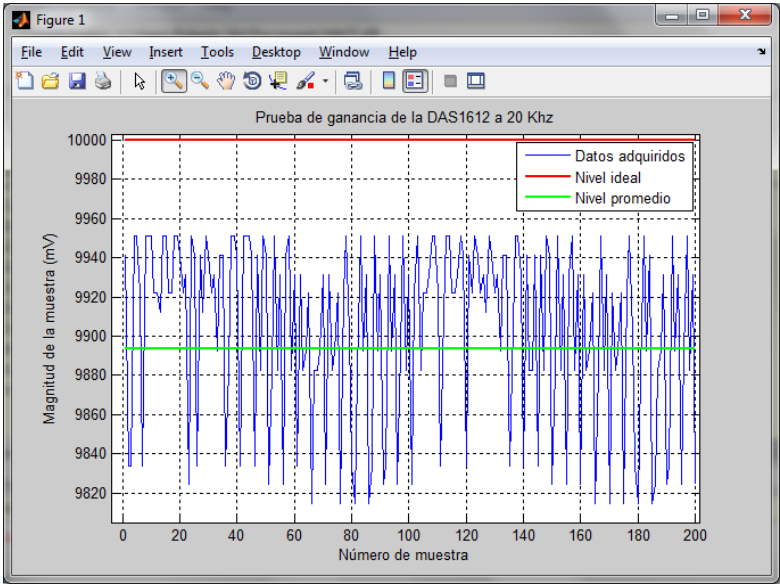


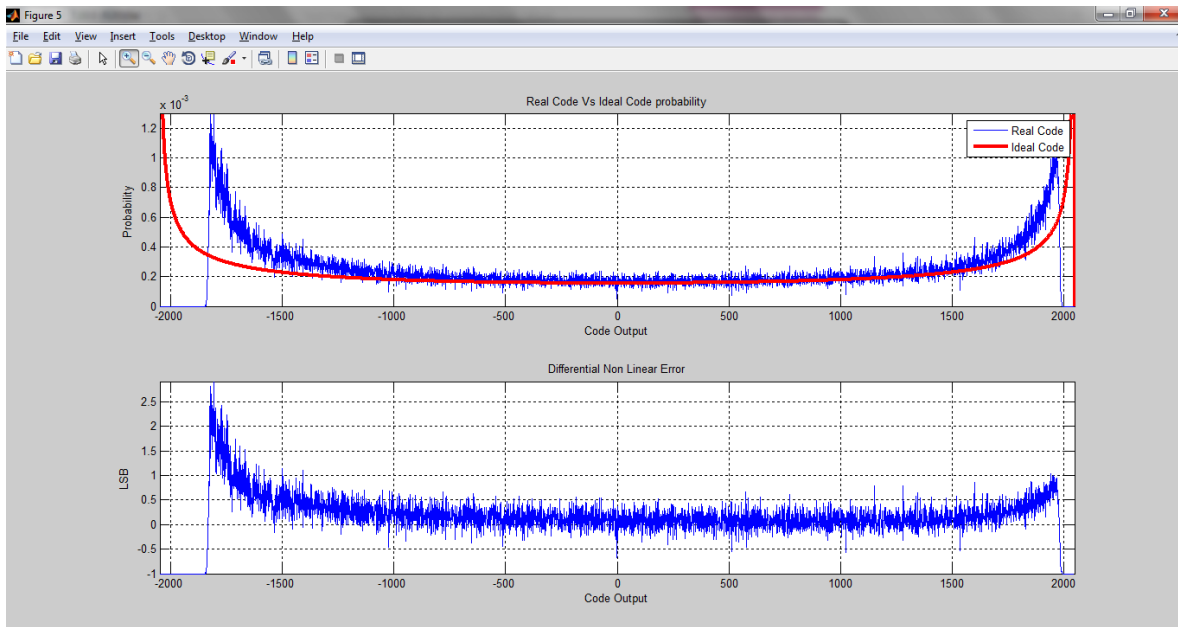
Figura 4-3 Prueba de ganancia DAS1612

Offset	Ganancia	THD	Linealidad
Promedio (V)	9.908672		
Varianza (mV*mV)	167.927734		
Desviación Estandar (mV)	12.958693		
Error Total (mV)	91.327667		
Pendiente Actual	0.990867		
<input type="checkbox"/> Guardar Datos	DataGain2.txt		
Calibrar ADC	Gain 100 %		
Resultado calibración	✓		

Figura 4-4 Pantalla de calibración del error de Ganancia

### 4.3 PRUEBAS DE LINEALIDAD DIFERENCIAL

En la Figura 4-5 se puede apreciar el error de linealidad diferencial, basándose en el marco teórico se puede deducir que los resultados de esta prueba son poco precisos, ya que no se obtiene una forma de onda muy similar a la ideal. Sin embargo, estos resultados sirven para identificar claramente que en las zonas cercanas a los límites analógicos inferior y superior existe una pérdida de códigos visible, y que sumado a esto, la probabilidad de los códigos cercanos a la zona de saturación (donde se pierden los códigos) está muy por encima de la ideal, e inclusive por encima del valor de 1.



**Figura 4-5 Prueba de DNLE DAS1612**

La relación entre ambos fenómenos puede ser interpretado como que dicha zona de códigos adyacentes a la zona de saturación reciben el peso de la probabilidad de los códigos faltantes, y que por lo tanto, trabajar con el convertidor dentro de esa zona no es muy confiable debido a la corroboración de la hipótesis planteada desde la prueba de ganancia; en efecto existe un efecto de saturación en las mediciones entregadas por el convertidor.



No obstante, también se puede observar claramente que existen zonas de códigos en las cuales el convertidor trabaja de una forma muy similar a la Ideal, por lo cual los resultados de adquisición provenientes de trabajar el convertidor dentro de esta zona son perfectamente confiables.

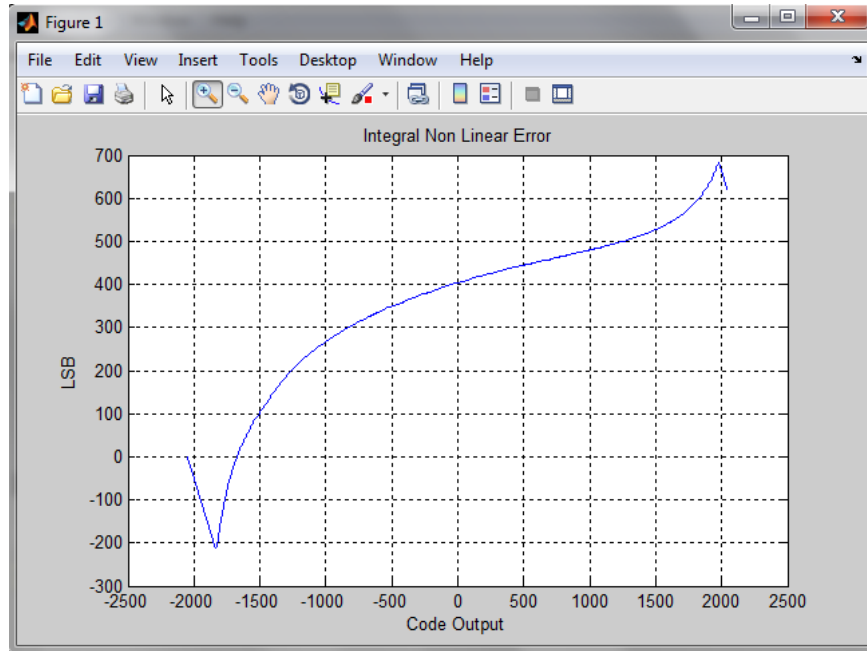


La Figura 4-6 Muestra la pantalla de calibración del DNLE, la cual muestra medidas de dispersión al igual que en el caso de los submenús anteriores, y contiene la lista de los códigos perdidos dentro del convertidor, en diferentes formatos, resultando en una herramienta bastante

**Figura 4-6 Pantalla de Calibración del DNLE**

agradable y de uso sencillo para un nuevo usuario de calibración.

#### 4.4 PRUEBAS DE LINEALIDAD INTEGRAL



**Figura 4-7 Prueba de INLE DAS1612**

La Figura 4-7 muestra una de las pruebas de INLE basada en una de las pruebas DNLE realizada sobre la DAS1612. Desafortunadamente, las gráficas provenientes de la prueba de INLE no resultaron de mucha ayuda en este caso debido a que la gran cantidad de códigos perdidos derivó en una suma de error a lo largo de los posibles códigos de salida.

#### 4.5 PRUEBA DE DISTORSIÓN ARMÓNICA TOTAL

En la Figura 4-8 se muestran los resultados de las pruebas de distorsión armónica total, en la Figura 4-8 a) se puede apreciar que en efecto existe distorsión armónica dentro en la señal, ya que existen múltiples picos en las frecuencias múltiples de la frecuencia fundamental. Más, vale destacar que las magnitudes de dichos armónicos no son lo suficientemente grandes como para concluir que el dispositivo se encuentra trabajando bajo condiciones de distorsión preocupantes.

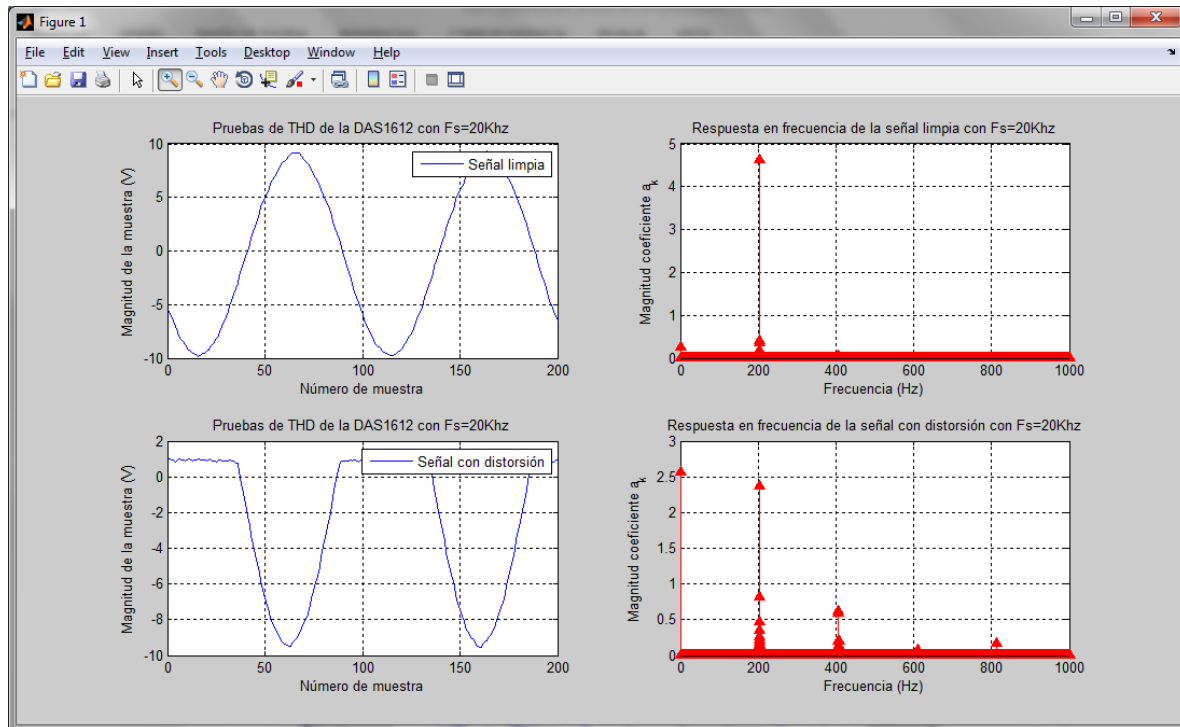


Figura 4-8 Análisis de THD de dos señales en la DAS1612

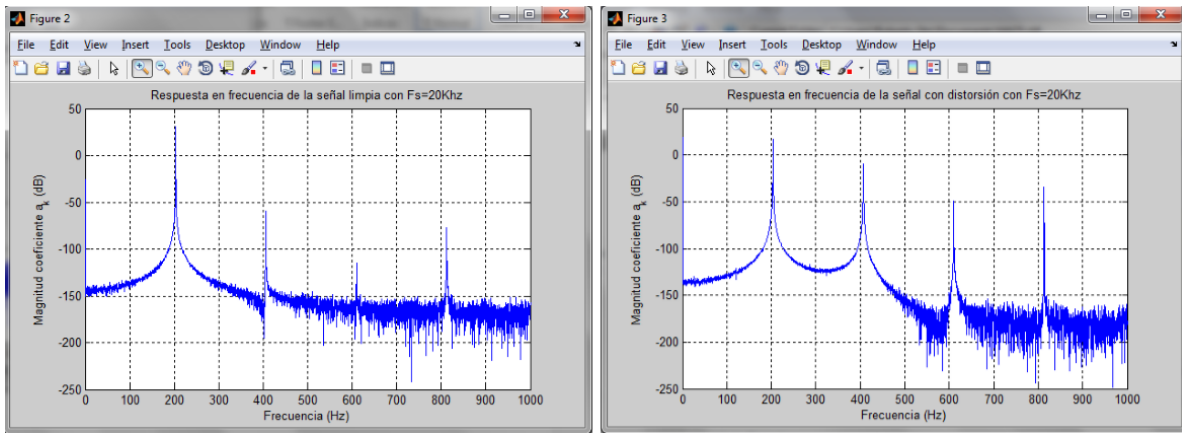
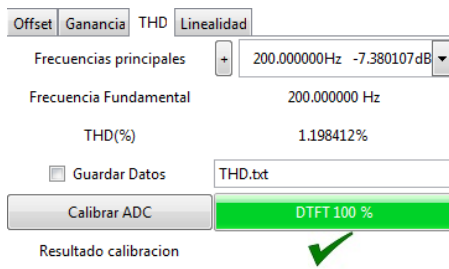


Figura 4-9 Respuesta en frecuencia en dB de las señales de prueba de THD en la DAS1612

La Figura 4-9 se incluye debido a es la comparación del análisis de una señal, con el análisis de una señal en la cual se han introducido algunos armónicos por medio de un Diodo zener a la entrada de la señal, tal y como se ve en su análisis espectral, los armónicos de la señal de la derecha tienen una amplitud mucho mayor que en la señal pura adquirida en la Figura 4-9 de la Izquierda. Lo interesante, es que a pesar de que los armónicos visualmente tienen una mayor magnitud, el índice de THD no refleja los mismos resultados, siendo inclusive menor que en el caso de la señal pura y sin perturbación.

Lo mencionado anteriormente es de gran relevancia debido a que para caracterizar un convertidor correctamente se deben de tener en cuenta otros aspectos que fueron mencionados en el Marco teórico, pero no fueron abordados a fondo dentro del contenido de esta tesis, como lo es el índice de SNR (por sus siglas en inglés *Signal-to -Noise Ratio*) .

Pues con una simple vista al espectro de la figura 3-4 se puede concluir que dicha señal tiene un SNR bastante pequeño, y es un tipo de fenómeno diferente el cual no es detectado por medio del índice de distorsión armónica total presente realizado sobre ella.



La Figura 4-10 Muestra la pantalla de calibración amigable desarrollada, la cual gracias a su herramienta de despliegue de armónicos principales, puede auxiliar al usuario a detectar este error de SNR fácilmente.

Figura 4-10 Pantalla de Calibración de THD

5. BIBLIOGRAFÍA

Alemaný Elena. *Prácticas de álgebra con mathematica*. 2002. Universidad politécnica de Madrid.

Anderson, Don. *Universal Serial Bus Architecture System Architecture*. 1997. Mind share Inc.

D. Brown, Stephen. J. Francis, Robert. Rose Jonathan, G. Vranesic, Zvonko. *Field-Programmable Gate Arrays*. 1992 Editorial Springer.

García Díaz, Celso Guillermo. *Sistema automático de reconocimiento sígnico para la lengua de señas mexicana*. 2012. Tesis Maestría UAQ.

González Cruz, Claudia Aidé. *Diseño de una red de sensores para control de iluminación basado en lógica difusa*. 2012. Tesis Maestría UAQ.

González Martínez, Edgar. *Desarrollo de una cámara automatizada para experimentación en plantas*. 2013. Tesis Maestría UAQ.

González Osuna, Carlos de Jesús. *Sistema basado en arquitectura híbrida FPGA- $\mu$ C para la medición remota de variables fisiológicas de plantas*. 2013 Tesis Maestría UAQ.

Granados-Lieberman, David; Osornio-Rios, Roque A; Rivera-Guillen, Jesus R; Trejo-Hernandez, Miguel; Romero-Troncoso, Rene J; *Torque reduction and workpiece finishing effects due to voltage sags in turning processes*, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 228, 1, 140-148, 2014, SAGE Publications.

H. Lundberg, Kent. *Analog-to-digital Converter Testing*. 2002. MIT Unpublished

Hoeschele, David F. *Analog-to-Digital and digital-to-analog conversion techniques*. 1994. Editorial John Wiley & Sons.

<http://www.gtk.org/> . Página oficial de GTK+ consultada en noviembre del 2014.

<http://www.gtkmm.org/en/> . Página oficial de Gtkmm consultada en noviembre del 2014.

[http://www.atx7006.com/articles/adc\\_histogram\\_test](http://www.atx7006.com/articles/adc_histogram_test). Página oficial de atx7006 consultada en Marzo del 2015

IEEE Std 519-1992, IEEE *Recommended Practices and Requirements for Harmonic Control in Electrical Power Systems*, New York, NY: IEEE.\*

Kester, Walt. *The data conversión handbook*. 2005. Analog Devices Inc.

Lehmann, Charles H.. *Geometría analítica*. 1989. Editorial Noriega.

Mixed signal Products. *Data acquisition Systems*. 1995. Texas Instruments.

Moro Piñeiro, María. *Metrología: introducción, conceptos e instrumentos*. 2000. Universidad de Oviedo.

Muñoz-Barron, Benigno; Morales-Velazquez, Luis; Romero-Troncoso, Rene J; Rodriguez-Donate, Carlos; Trejo-Hernandez, Miguel; Benitez-Rangel, Juan P; Osornio-Rios, Roque A; *FPGA-Based Multiprocessor System for Injection Molding Control*, Sensors, 12, 10, 14068-14083, 2012, Molecular Diversity Preservation International.

Oppenheim, Alan V. *Discrete-Time Signal Processing*. 1999, Editorial Prentice Hall

Ramírez Villeda,Guillermo. *Sistema no invasivo basado en el monitoreo de temperatura del agua para predecir el crecimiento del pez tilapia implementado con microcontrolador y tecnología ZigBee*. 2013. Tesis Maestría UAQ.

Romero Troncoso, René de F. *Electrónica digital y lógica programable*. 2007. Universidad de Guanajuato.

Schaumont, Patrick. *A practical Introduction to Hardware/Software Codesign*. 2013. Editorial Springer.

Texas Instruments. *A glossary of Analog-to-digital specifications and performance characteristics*. 2011. Application Report

Thompson, M. *.FPGAs accelerate time to market for industrial designs*. 2004. EE Times.

Tocci, Ronald J. *.Sistemas Digitales*. 2003. Editorial Prentice Hall México.

Trejo-Hernandez, Miguel; Osornio-Rios, Roque Alfredo; Romero-Troncoso, Rene de Jesus; Rodriguez-Donate, Carlos; Dominguez-Gonzalez, Aurelio; Herrera-Ruiz, Gilberto; *FPGA-*

*based fused smart-sensor for tool-wear area quantitative estimation in CNC machine inserts*, Sensors, 10, 4, 3373-3388, 2010, Molecular Diversity Preservation International.

Trimberger Stephen, *Field-Programmable Gate Array Technology*. 1994. Editorial Springer.

Valtierra-Rodriguez, Martin; Romero-Troncoso, Rene de J; Garcia-Perez, Arturo; Granados-Lieberman, David; Osornio-Rios, Roque A; *Reconfigurable instrument for neural-network-based power-quality monitoring in 3-phase power systems*, IET Generation, Transmission & Distribution, 7, 12, 1498-1507, 2013, IET Digital Library

Yichuang, Sun. *Test and Diagnosis of analogue, Mixed signals and RF Integrated Circuits The system on chip approach*. 2008. Editorial The Institution of Engineering and Technology

Zeidman, Bob. *All about FPGAs*. 2006. EE Times.

## 6. APENDICE

### 6.1 Manual de Operación del Software

#### 6.1.1 Introducción

El presente manual se ha realizado con la finalidad de contar con un medio que sirva como guía para el uso correcto de los distintos elementos utilizados para manipular datos provenientes o dirigidos hacia algún dispositivo FPGA con convertidor D/A y A/D.

El material que aquí se incluye se divide en dos partes, siendo la primera limitada exclusivamente hacia la manipulación del software desde la interfaz gráfica sin tomar en cuenta las conexiones físicas externas que se deben realizar para comunicar correctamente el dispositivo FPGA con la PC, y la segunda parte en la cual se explica a mayor detalle la forma de establecer comunicación con un nuevo dispositivo o host, siendo tratado como esclavo dentro de la arquitectura en Hardware utilizada.

#### 6.1.2 Uso de la Interfaz gráfica

Para comenzar a describir el uso de la interfaz gráfica es necesario familiarizarse con los distintos módulos que la componen y el modo correcto en el cual pueden ser manipulados para posteriormente describir por completo los distintos modos de adquisición y reproducción de datos incluidos en el sistema.

La Figura 6-1 muestra la interfaz gráfica totalmente desarrollada, la cual será dividida a partir de ahora en distintos módulos para obtener una mejor comprensión del sistema. Dichos módulos son el Menú general de funciones, el Menú general de configuración, el Menú de Configuración Gráfica, el Menú de configuración de Canales I/O, el Área gráfica, la Consola de aplicación y el Módulo de operaciones matemáticas los cuales son mostrados explícitamente a continuación.

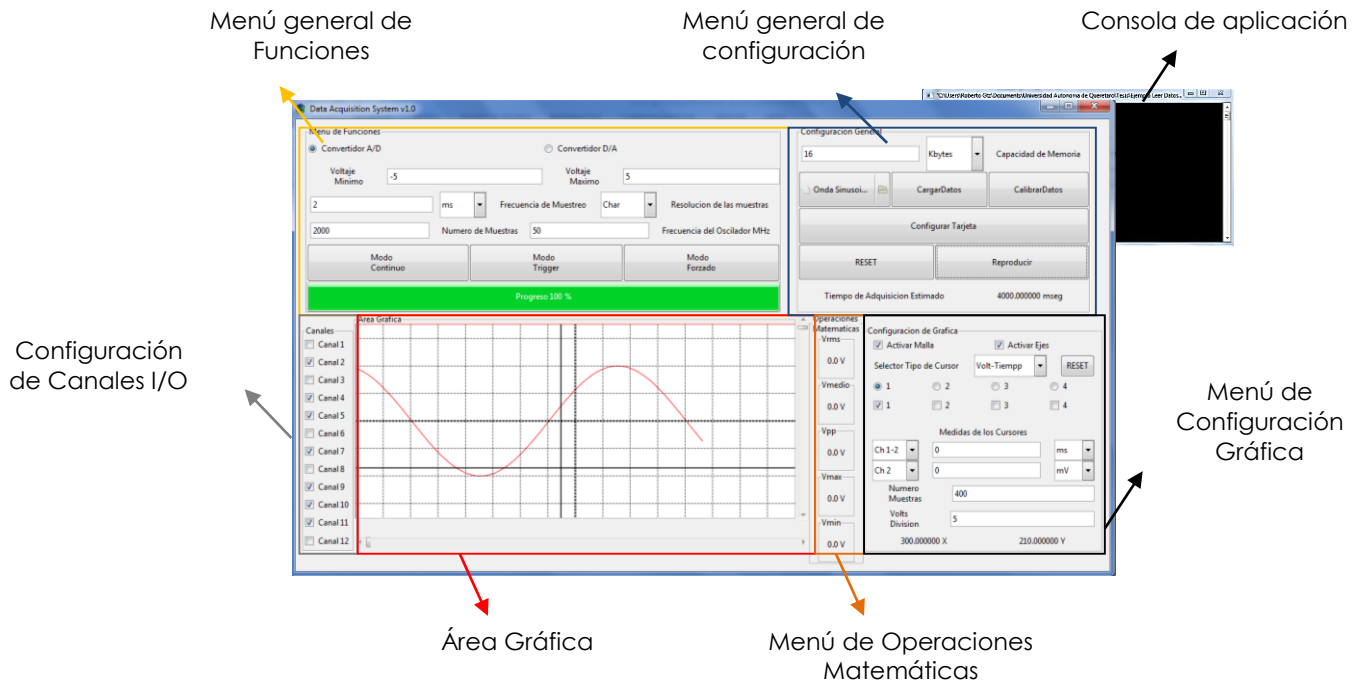


Figura 6-1 Módulos de la Interfaz Gráfica desarrollada

6.1.2.1 Consola de Aplicación



Figura 6-2 Consola de Aplicación

La consola de aplicación usualmente sirve como auxiliar para informar sobre un problema con el uso del software, es decir cuando falta instalar alguna librería dinámica .dll para poder utilizar el programa o cuando existe algún problema con el manejo del software.

Al inicializar el programa, la consola de aplicación auxilia a determinar si se ha identificado un dispositivo al inicializar el programa e inmediatamente despliega un mensaje con el VID y PID con los cuales se ha reconocido dicho dispositivo.

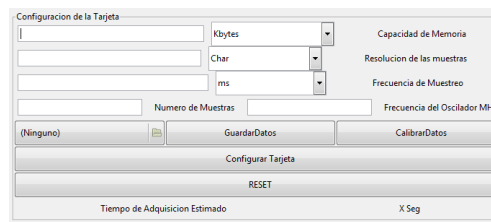


En caso de no encontrar un dispositivo al inicializar el programa, es necesario verificar que el driver del dispositivo haya sido instalado correctamente.

### 6.1.2.2 Menú general de configuración

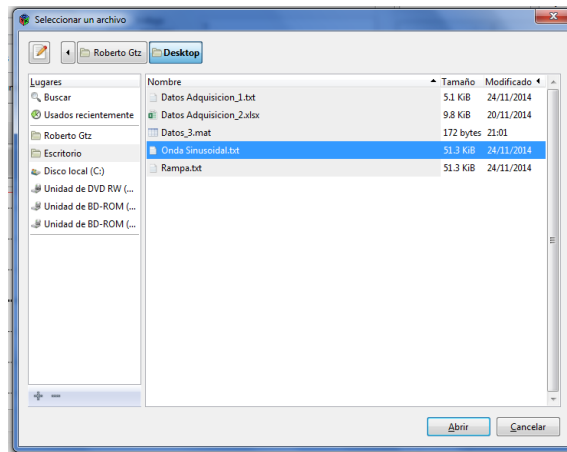
¿Cómo cargar un archivo?

Una vez que se haya identificado al menos un dispositivo con el cual comunicarse, es posible utilizar el menú general de configuración para cargar un archivo de los tipos soportados presionando una vez sobre el botón de diálogo de apertura el cual se muestra en la Figura 6-3



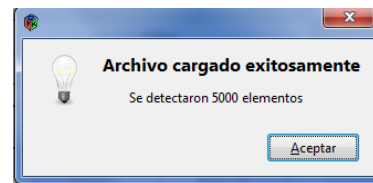
**Figura 6-3 Botón de dialogo de Cargar archivo**

Tal y como se puede observar en la Figura 6-3, el botón de apertura de archivo se encuentra inicialmente vacío, lo que implica que no se encuentra ningún archivo cargado con el cual trabajar, por lo que es necesario presionar sobre el botón y cargar un archivo. Una vez presionado el botón se abrirá un dialogo de selección de archivo como el mostrado en la Figura 6-4, el cual contiene un filtro para abrir únicamente archivos del tipo “.mat” , “.txt” y “.xlsx”. Dicho diálogo permite buscar y seleccionar el archivo deseado para apertura dentro de cualquier directorio de la PC utilizando la barra de navegación incluida en el costado izquierdo del diálogo. Para abrir un archivo simplemente basta con seleccionarlo como se muestra en la Figura 6-4 y presionar el botón de “Abrir” que se sobresalta dentro de la misma figura:



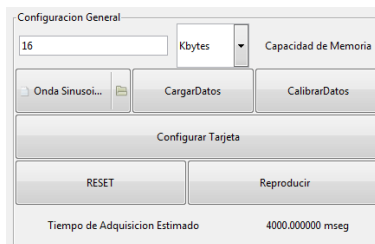
**Figura 6-4** Diálogo de apertura de archivo

Una vez que se haya cargado exitosamente algún archivo debe abrirse una ventana emergente como la mostrada en la Figura 6-5 la cual contiene información acerca del número de elementos detectados para poder trabajar sobre ellos, en caso de haber cargado un archivo vacío, para ser llenado posteriormente con los datos adquiridos el número será igual a cero.



**Figura 6-5** Diálogo de carga exitosa

Tal y como se puede observar en la Figura 6-6 una vez que se haya cargado un archivo exitosamente, debe de visualizarse el archivo cargado con su icono respectivo.



**Figura 6-6** Archivo cargado exitosamente

6.1.2.3 Menú general de funciones

¿Cómo configurar el modo de funcionamiento?

Existen diferentes modos de funcionamiento dependiendo del dispositivo a controlar, por ahora existen 6 modos de funcionamiento, dependiendo de si lo que se desea utilizar es el DAQ como un convertidor A/D o D/A los cuales serán descritos a mayor detalle en la siguiente parte del manual. Para elegir cuál modo de funcionamiento utilizar basta con utilizar el selector incluido en el Menú general de funciones tal y como se indica en la Figura 6-7. Dependiendo de si se ha escogido utilizar el convertidor A/D o el convertidor D/A los tres botones pertenecientes a cada modo de operación cambian su diálogo y la función que ejecutan.

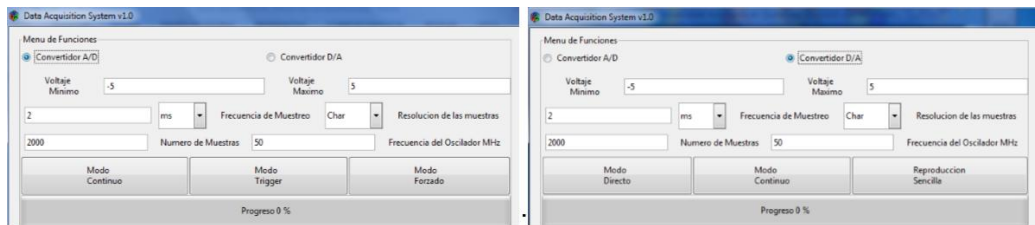
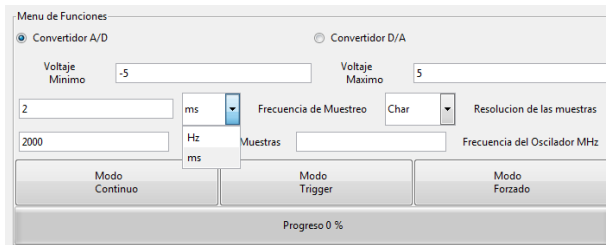


Figura 6-7 Selección de Modo

Para utilizar alguna de las funciones simplemente basta con presionar sobre el botón deseado una vez que la configuración general del convertidor haya sido llenada por completo, de lo contrario una ventana emergerá y dará aviso sobre campos incompletos que deben ser llenados.

- ✓ Ajuste del periodo de muestreo

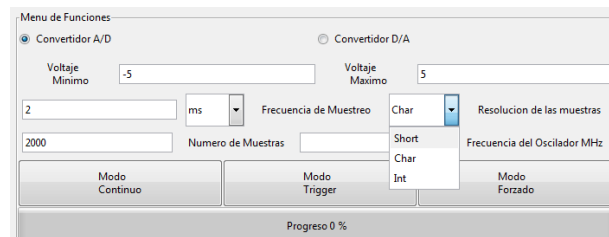
Para ajustar la resolución de las muestras basta con hacer click sobre el combobox incluido a la izquierda de la etiqueta con el mensaje de *Frecuencia de Muestreo*. Al hacer click se desplegará una lista mostrando las opciones soportadas por el sistema tal y como se muestra en la Figura 6-8. Dicho parámetro se puede ingresar en milisegundos o en Hertz teniendo el mismo resultado final sobre el sistema DAQ, siendo estos dos modos un manera de hacer más amigable y eficiente el uso de la plataforma de adquisición de datos.



**Figura 6-8 Ajuste del periodo de muestreo**

✓ Ajuste de la resolución de las muestras

Para ajustar la resolución de las muestras basta con hacer click sobre el combobox incluido a la izquierda de la etiqueta con el mensaje de resolución de las muestras. Al hacer click se desplegará una lista mostrando las resoluciones soportadas por el sistema tal y como se muestra en la Figura 6-9. Dichas resoluciones se encuentran explicadas en la Tabla 6-1.

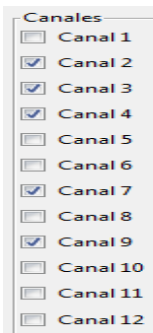


**Figura 6-9 Ajuste de la resolución de las muestras**

**Tabla 6-1 Características de resolución**

Formato de la Resolución	Número de bits por muestra
Short	8
Char	16
Int	32

✓ Ajuste del número de canales



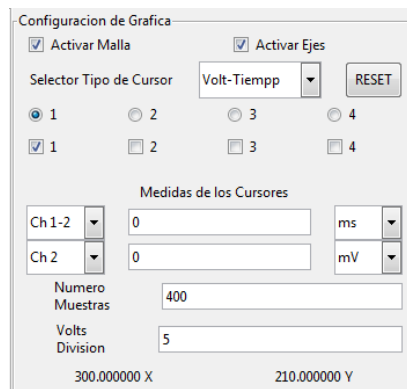
Para ajustar los canales de adquisición que se van a utilizar se incluyó el Menú de canales I/O, el cual determina específicamente cuáles canales van a ser utilizados. Para activar algún canal en específico basta con hacer click sobre la casilla de verificación a un costado del nombre del canal.

**Figura 6-10 Barra de selección de canales**

6.1.2.4 Menú de Configuración Gráfica

✓ Ajuste del Área de Dibujo

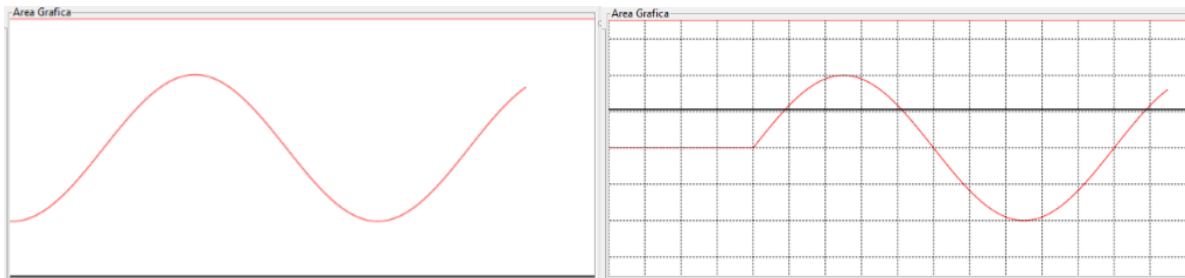
El ajuste del área de dibujo se realiza mediante la manipulación de los parámetros disponibles dentro del Menú de configuración gráfica ubicada en la Figura 6-11. Teniendo la capacidad de Activar y desactivar la malla de dibujo y los ejes principales, el uso de hasta cuatro diferentes cursores de voltaje o tiempo, obtener medidas respectivas en voltaje y/o tiempo de dichos cursores, controlar el número de muestras mostradas en una ventana regular de dibujo, así como controlar la escala del dibujo utilizada y Además cuenta con un monitoreo constante de Voltaje y Tiempo que indican la posición en la cual se encuentra el Mouse o puntero del usuario mientras se encuentra dentro del área de gráficos.



**Figura 6-11 Menú de configuración Gráfica**

Activar o desactivar el Mallado del dibujo

Para activar o desactivar el mallado del dibujo, basta con hacer click y cambiar el estado de la casilla de verificación ubicada a la izquierda de la etiqueta con el mensaje de “Activar Malla” que se muestra en la Figura 6-11. El efecto sobre el área de gráficos se puede apreciar en la Figura 6-12a) y la Figura 6-12b).



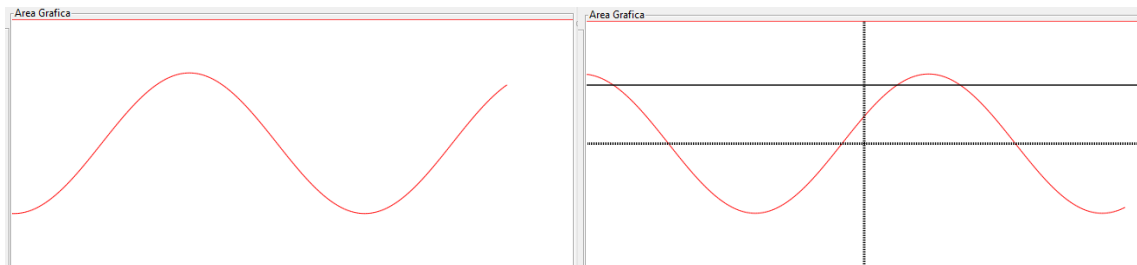
a) Área de dibujo sin Malla

b) Área de dibujo con Malla

**Figura 6-12 Activación y desactivación del Mallado**

Activar o desactivar los ejes principales del dibujo

Para activar o desactivar los ejes principales del dibujo del dibujo, basta con hacer click y cambiar el estado de la casilla de verificación ubicada a la izquierda de la etiqueta con el mensaje de “Ejes principales” que se muestra en la Figura 6-11. El efecto sobre el área de gráficos se puede apreciar en la Figura 6-13 a) y la Figura 6-13 b).

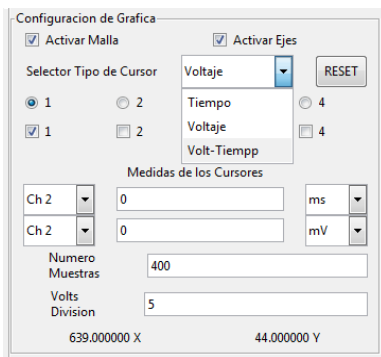


a) Área de dibujo sin Malla y sin ejes

b) Área de dibujo sin Malla y con ejes

**Figura 6-13 Activación y desactivación de los Ejes principales**

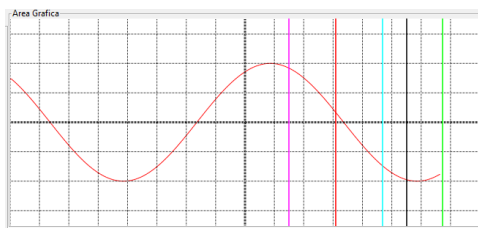
Uso de los cursores



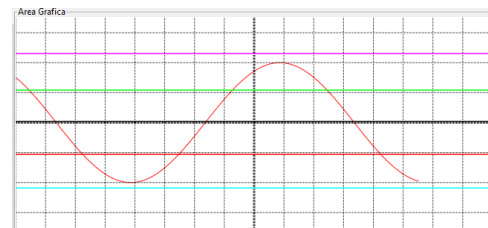
Dentro de la configuración gráfica existe la opción de activar cursores de tres diferentes tipos, para activar algún tipo de cursor en específico basta con hacer click sobre el combobox a la derecha de la etiqueta con el mensaje de *Selector del tipo de cursor*. En el momento de realizar el click el combobox debe desplegar tres tipo de cursores tal y como se muestra en la Figura 6-14.

Figura 6-14 Menú de cursores

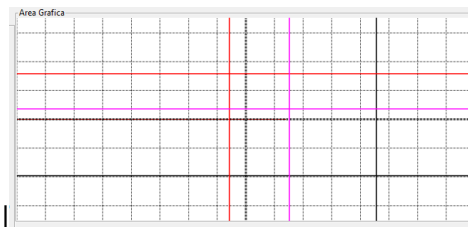
Los tres tipos de cursores se muestran para una mejor visualización dentro de las Figuras 6-15 a) , 6-15 b) y 6-15 c). Existen cursores de únicamente de medida de tiempo, cursores únicamente de medida de voltaje, y cursores que miden voltaje y tiempo simultáneamente.



a) Cursores de tiempo



b) Cursores de Voltaje



c) Cursores de Voltaje-tiempo

Figura 6-15 Tipos de cursores disponibles

El uso de los cursores depende se ajusta en el menú de la configuración gráfica. Dentro de ella existen dos grupos; Un grupo de botones del tipo *radiobutton*, y un grupo de casillas de verificación tal y como se puede ver en la Figura 6-16. El grupo de casillas de verificación habilitan la visualización del número de cursor que tienen a su derecha. Si la

casilla de verificación se encuentra activa, el cursor será visible, de lo contrario, no se mostrará en la zona de dibujo.

Dentro del grupo radiobutton únicamente puede permanecer un cursor activo a la vez, el cual se encontrará en el mismo punto sobre el cual se encuentra ubicado el mouse de la computadora si es que el mouse se encuentra dentro del área gráfica.

Para guardar y fijar la posición exacta del cursor en un momento o posición determinada, basta con hacer click con el botón izquierdo del mouse en cualquier momento en el que el mouse se ubique dentro del área gráfica.

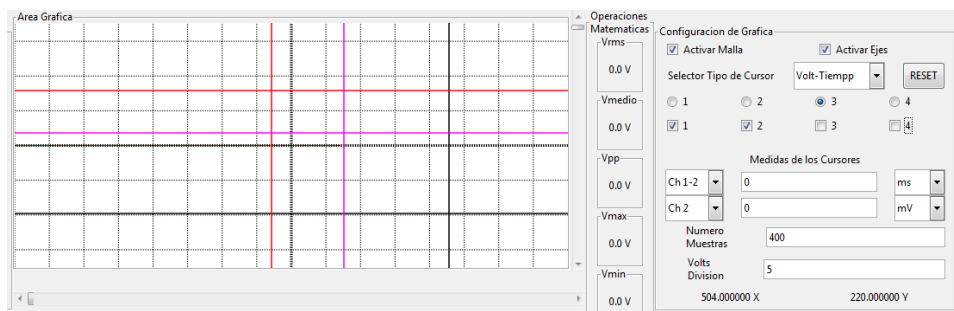


Figura 6-17 Selección de cursor activo

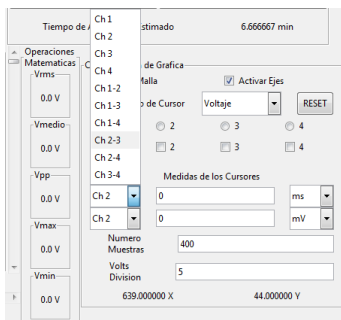
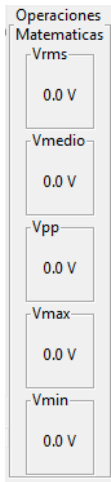


Figura 6-16 Medidas de los cursores

Para obtener la medida exacta de algún cursor basta con hacer click sobre el ComboBoxText ubicado en la zona de *medida de cursores* mostrado en la Figura 6-16 y elegir entre las todas las medidas disponibles entre los cuatro cursores. Debe tomarse en cuenta que los cursores permanecerán activos a pesar de que el área gráfica se encuentre en movimiento o cargando datos.



6.1.2.5 Menú de operaciones aritméticas



El menú de operaciones aritméticas muestra valores de interés típicos durante la adquisición de datos, los cuales cambiarán dependiendo de la señal de entrada que se haya cargado y que se esté mostrando en el momento dentro del área de gráficos. Entre las operaciones incluidas hasta el momento se encuentran el valor del voltaje rms, voltaje medio, voltaje pico-pico, voltaje máximo y voltaje mínimo. Manteniendo la posibilidad de agregar más funciones y operaciones en línea en un futuro

Figura 6-18 Menú de operaciones aritméticas

6.1.2.6 Menú de calibración de datos

Funciones de calibración de datos

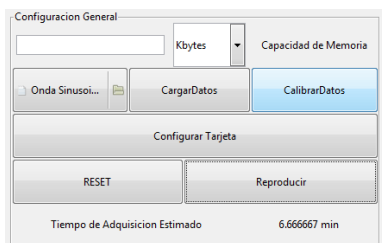


Figura 6-19 Botón de Inicio de proceso de calibración

Para realizar las funciones de calibración basta con presionar el botón de calibración de datos, el cual desplegará una ventana emergente con ayuda para determinar el error de linealidad, error de ganancia, error de distorsión armónica y el error de offset. Debido a que cada uno de estos análisis requiere de parámetros de entrada distintos la metodología de ingreso de parámetros

a seguirse es distinta dependiendo del tipo de error que se requiera cuantificar la calibración de datos se dividirá en cuatro secciones, siendo cada una perteneciente a un tipo de error específico: