



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA
SECCIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

**“DISEÑO DE UNA UNIDAD DE MEDICIÓN FASORIAL
CON TECNOLOGÍA ARM”**

TESIS

QUE PARA OBTENER EL GRADO DE:

**MAESTRO EN CIENCIAS CON ESPECIALIDAD
EN INGENIERÍA ELÉCTRICA**

PRESENTA:

ING. DIEGO VALERIO TORRES

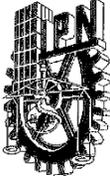


Dr. Domitilo Libreros
Director de Tesis

Ciudad de México, Diciembre 2016

Diseño de una Unidad de Medición Fasorial con tecnología ARM

SIP 14



INSTITUTO POLITÉCNICO NACIONAL SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

SIP-14

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D. F. siendo las 12:00 horas del día 15 del mes de Noviembre del 2016 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de E.S.I.M.E.-ZAC. para examinar la tesis de titulada:

"DISEÑO DE UNA UNIDAD DE MEDICIÓN CON TECNOLOGÍAS ARM"

Presentada por el alumno:

VALERIO
Apellido paterno

TORRES
Apellido materno

DIEGO
Nombre(s)

Con registro:

B	1	4	0	8	6	2
---	---	---	---	---	---	---

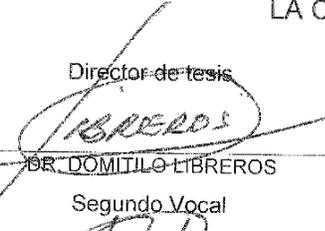
aspirante de:

MAESTRO EN CIENCIAS EN INGENIERÍA ELÉCTRICA

Después de intercambiar opiniones los miembros de la Comisión manifestaron **SU APROBACIÓN DE LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

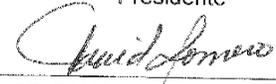
Director de tesis


DR. DOMITILO LIBREROS

Segundo Vocal


DR. RAÚL ÁNGEL CORTÉS MATEOS

Presidente


DR. DAVID ROMERO ROMERO

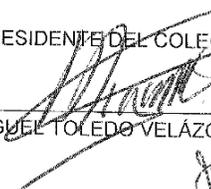
Tercer Vocal

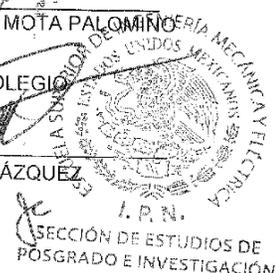

DR. DAVID SEBASTIÁN BALTAZAR

Secretario


DR. RICARDO OCTAVIO ARTURO MOTA PALOMINO

EL PRESIDENTE DEL COLEGIO


DR. MIGUEL TOLEDO VELÁZQUEZ


I.P.N.
SECCIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN

Diseño de una Unidad de Medición Fasorial con tecnología ARM

CARTA DE CESIÓN DE DERECHOS



INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México, D.F. el día 25 del mes de noviembre del año 2016, el que suscribe **Ing. Diego Valerio Torres** alumno del Programa de Maestría en Ciencias en Ingeniería Eléctrica, con número de registro **B140862**, adscrito a la Sección de Estudios de Posgrados e Investigación de la ESIME Unidad Zacatenco, manifiesto que es el autor intelectual del presente trabajo de Tesis bajo la dirección del **Dr. Domitilo Libreros** y cede los derechos del trabajo titulado **Diseño de una Unidad de Medición Fasorial con tecnología ARM**, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a las siguientes direcciones: **d.valerio.t@gmail.com** y/o **dlibrero@ipn.mx**.

Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Diego Valerio Torres

Diseño de una Unidad de Medición Fasorial con tecnología ARM

DEDICATORIA

A mis amados padres Silva y Carlos, quienes han estado al pendiente de mi formación, y quienes me han enseñado a valorar las cosas de la vida. Por todo el apoyo incondicional, palabras de aliento y amor.

A mis queridos hermanos Carlos, Iveet, Verónica, Ivan, Carolina, Paulo, Sideri y Erendida, en las buenas y en las malas, siempre he contado con ustedes incondicionalmente.

A toda mi familia, consanguínea o no, por su confianza y apoyo.

A mis amigos y compañeros, que me ayudaron en mi formación personal y profesional.

A los profesores que, más allá de las aulas, me brindaron su amistad y compañerismo.

A las personas que confían en mí y no dejan de apoyarme.

A mis abuelos, y a aquellos seres queridos que se han adelantado.

Este trabajo no sólo es mío, es de todos ustedes, quienes han estado en mi vida y se enorgullecen de mí. Para ustedes, con todo mi esfuerzo.

“El éxito consiste en obtener lo que se desea. La felicidad, en disfrutar lo que se obtiene”

- Ralph Emerson

Diseño de una Unidad de Medición Fasorial con tecnología ARM

AGRADECIMIENTOS

En primera instancia, a Dios, porque sin su voluntad nada es posible, por la fuerza de convicción, los dones que me brindó y todas las bendiciones y obstáculos que puso en mi camino, a fin de lograr la culminación de mis estudios de maestría y la realización de este trabajo.

A mis padres y hermanos, porque sin ellos no habría podido siquiera pensar en estudiar la maestría, por tantas noches de desvelos, días sin salir, problemas afrontados juntos, peleas y toda situación que nos ha permitido permanecer más unidos que nunca. Por la paciencia que me tienen y por mantenerme en el camino correcto, no tengo manera de agradecerles.

Al director de la tesis, Dr. Domitilo Libreros, por su guía, valiosos consejos, enseñanzas y la paciencia para la elaboración de este trabajo. Por compartir más allá de una cátedra, experiencias de vida para mi formación personal. Por todo lo enseñado y su apoyo, gracias.

Al Dr. Raúl Cortés Mateos, quien siempre logró sacar lo mejor de mí, y siempre conté con su guía, enseñanzas, apoyo técnico y moral, lecciones de vida y en el aula, pero sobre todo, por ser una persona tan capaz de entregarse por completo a la realización de este trabajo, y aun así ser tan comprensible en los momentos difíciles. Muchas gracias.

Al Dr. Raúl Alcántara Fernández, quien siempre ha confiado en mis habilidades más de lo que yo he confiado en ellas, con quien he podido contar en las buenas y en las malas, y qué es el ejemplo a seguir más digno que he encontrado en la vida profesional y personal. Por ser un gran amigo, profesor y tutor, gracias.

Al Instituto Politécnica Nacional, a la Escuela Superior de Ingeniería Mecánica y Eléctrica, a todos mis amigos y compañeros en el camino, así como el personal de apoyo de la SEPI, quienes me han hecho el honor de brindar su amistad y cariño, porque siempre me apoyaron cuando lo necesité y no escatimaron en recursos para brindarme su ayuda. Gracias.

Agradezco a los integrantes de la comisión revisora de esta tesis, quienes ayudaron a mejorar la calidad de la misma: Dr. David Romero Romero, Dr. Ricardo Mota Palomino, Dr. Domitilo Libreros, Dr. Raúl Cortés Mateos, Dr. David Sebastián Baltazar y Dr. Daniel Ruiz Vega.

Por último y no menos importante, agradezco al Consejo Nacional de Ciencia y Tecnología (CONACyT), institución que brindó el apoyo económico necesario para la realización de este trabajo.

Diseño de una Unidad de Medición Fasorial con tecnología ARM

RESUMEN

En el siguiente trabajo, se presenta el diseño de una unidad de medición fasorial (PMU) sincronizada mediante un módulo receptor del sistema satelital de posicionamiento global (GPS). El medidor implementado tiene la capacidad de medir los fasores de voltaje y corriente de un sistema trifásico con frecuencia nominal de 60 Hz. Estos fasores son calculados mediante el algoritmo de la Transformada Discreta de Fourier (DFT) y pueden ser representados en forma polar o rectangular.

El estampado de tiempo, la frecuencia calculada y los fasores obtenidos, son enviados al sistema receptor, en este caso una PC, mediante un puerto de comunicación serial basado en las tramas de información establecidas en el estándar IEEE C37.118.2.

Se expone la utilización del algoritmo de la DFT para el cálculo de fasores, explicando la forma en que se obtienen los ángulos de los fasores mediante una Tabla de Búsqueda (LUT) y la identidad trigonométrica utilizada.

Mediante el uso de la herramienta FreeMASTER se logra visualizar los fasores en forma gráfica, creando curvas para la magnitud y ángulo de los fasores, los valores de la frecuencia calculada y el valor eficaz (RMS) de las señales analizadas.

La parte central del medidor está conformada por una tarjeta de desarrollo de NXP (TWR-KM34Z50M), con un microcontrolador MK34Z128CLL5 y un microcontrolador generador de señales K20 que permite emular una red eléctrica real (3 señales de corriente y 3 señales de voltaje). La tarjeta se comunica con el módulo receptor GPS para recibir la señal de sincronización de un pulso por segundo (pps) para el cálculo de los fasores y obtención del Tiempo Universal Coordinado (UTC).

Se realizan pruebas en estado estacionario para determinar el cumplimiento en Magnitud, Ángulo de Fase y Contenido Armónico conforme al estándar IEEE C37.118.1, comprobando que la unidad de medición desarrollada se acata a lo establecido en dicho estándar.

Diseño de una Unidad de Medición Fasorial con tecnología ARM

ABSTRACT

In this thesis, the design of a phasor metering unit (PMU) synchronized via satellite through a global positioning system (GPS) receiver module is presented.

The implemented meter unit has the ability to measure the phasors of voltage and current of a three-phase system with a nominal frequency of 60 Hz. These phasors are calculated by the algorithm of the Discrete Fourier Transform (DFT) and can be represented in polar or rectangular form.

The time stamp, the calculated frequency and the obtained frequency are sent to the receiving system, in this case a PC via serial communication port based on data frames established in IEEE Standard C37.118.2

The use of the DFT algorithm for the computation of phasors is explained, explaining how the angles of the phasors are obtained by a look up table (LUT) and the trigonometric identity used.

By using the FreeMASTER tool, display phasor graphically is achieved creating curves for the magnitude and phase of the phasor, the values of the calculated frequency and the root mean square value (RMS) of the analyzed signals.

The central part of the measurement unit is made up of a NXP development board (TWR-KM34Z50M), with a MK34Z128CLL5 microcontroller and a K20 signal generator microcontroller that can emulate a real grid (3 current signals and 3 voltage signals). The board communicates with the GPS receiver module to obtain the synchronization signal of one pulse per second (pps) for calculating phasors and obtaining the Universal Time Coordinated (UTC).

Steady state tests are performed to determine compliance in Magnitude, Phase Angle and Harmonic Content according to C37.118.1 IEEE Standard, verifying that the developed measurement unit complies with the established in said standard.

Diseño de una Unidad de Medición Fasorial con tecnología ARM

ÍNDICE

SIP 14	i
Carta de Cesión de Derechos	iii
Dedicatoria	v
Agradecimientos.....	vii
Resumen	ix
Abstract.....	xi
Índice de Figuras	xvii
Índice de Tablas.....	xix
Glosario	xxi
Capítulo 1 Introducción.....	1
1.1 Generalidades.....	1
1.2 Objetivo de la Tesis	2
1.2.1 Objetivos Particulares	2
1.3 Justificación.....	2
1.4 Estado del Arte.....	3
1.5 Alcances y limitaciones	5
1.6 Aportaciones.....	5
1.7 Estructura de la Tesis.....	6
Capítulo 2 Sistema de Medición de Fasores.....	7
2.1 Introducción	7
2.2 Conceptos básicos	8
2.2.1 Valor RMS.....	8
2.2.2 Concepto de Fasor	9
2.2.3 Concepto de Fase.....	9
2.2.4 Muestreo Digital	10
2.2.5 Teorema del Muestreo.....	10
2.3 Transformada Discreta de Fourier.....	11
2.3.1 Definición de la Transformada Discreta de Fourier	12
2.3.2 Funciones base de la Transformada Discreta de Fourier.....	13
2.3.3 Inversa de la Transformada Discreta de Fourier.....	14
2.3.4 Cálculo de la DFT por correlación	15
2.4 Medición de frecuencia	19

Diseño de una Unidad de Medición Fasorial con tecnología ARM

2.4.1	Cálculo de la Frecuencia por Cruce por cero	19
2.5	Sistema de Posicionamiento Global	22
2.5.1	Introducción	22
2.5.2	Arquitectura	22
2.5.3	Triangulación	24
2.5.4	Tiempo Universal Coordinado (UTC)	25
Capítulo 3	Hardware	27
3.1	Introducción	27
3.2	Diagrama Hardware/Software	27
3.3	Características de la Tarjeta TWR KM34Z50M	29
3.3.1	Microcontrolador MK34Z128CLL5	32
3.3.2	Convertidor Analógico Digital	32
3.3.3	Bornes de Entrada/Salida de Propósito General (GPIO)	37
3.3.4	Interruptor Inter-Periféricos (XBAR)	37
3.3.5	Receptor/Transmisor Universal Asíncrono (UART)	39
3.4	Generador de Señales	39
3.5	Módulo GPS	45
3.5.1	Módulo Receptor	45
3.5.2	Antena GPS	46
3.5.3	Pulso Por Segundo (pps)	47
Capítulo 4	Diseño de Software	49
4.1	Introducción	49
4.2	Ambiente de desarrollo del microcontrolador	49
4.2.1	Software de la Tarjeta de desarrollo TWR KM34Z50M	50
4.2.2	Crear un proyecto para la Tarjeta de desarrollo TWR KM34Z50M	50
4.3	Medición y Cálculo	52
4.3.1	Programa Principal (PP)	54
4.3.2	ADCs	54
4.3.3	Frecuencímetro	57
4.3.4	DFT	59
4.3.5	Cálculo del Valor RMS	62
4.3.6	Cálculo de la Raíz Cuadrada	62
4.4	Sistema de Posicionamiento Global	63
Capítulo 5	Pruebas	65
5.1	Introducción	65

5.2	Medición de Frecuencia	65
5.3	Valor RMS.....	69
5.4	Cálculo de Fasores.....	71
5.4.1	Fasores de una señal con frecuencia nominal.....	71
5.4.2	Fasores de una señal con frecuencia fuera de la nominal	73
5.5	Respuesta en el tiempo del PMU	76
5.6	Límites de Precisión	77
5.6.1	Cambio de magnitud en la entrada del PMU.....	77
5.6.2	Cambio de ángulo de fase	81
5.6.3	Distorsión Armónica.....	83
5.7	Comunicación.....	85
Capítulo 6	Conclusiones y Recomendaciones	89
6.1	Conclusiones.....	89
6.2	Aportaciones.....	91
6.3	Recomendaciones para trabajos futuros	91
Bibliografía	93
APÉNDICE A	Código Implementado.....	97
A.1	Programa Principal (pmu_gps.c).....	97
A.2	Header de Cálculos (mediciones.h)	101
A.3	Header de Datos de Tramas (Datos_Trama.h)	110
A.4	Header de Tramas (tramas.h).....	110
APÉNDICE B	Estándar IEEE C37.118.....	119
B.1	Introducción.....	119
B.2	Descripción	119
B.2.1	Necesidades del Estándar	120
B.3	Definiciones	120
B.4	Medición de Sincrofasores.....	121
B.4.1	Definición de Fasor	121
B.4.1.1	Señales fuera de la Frecuencia Nominal	121
B.4.2	Definición de Sincrofasor.....	122
B.4.3	Medición para el estampado de tiempo para el sincrofasor.....	123
B.4.4	Sincronización del Sistema de tiempo.....	123
B.5	Requerimientos de las Mediciones del Sincrofasor y el Cumplimiento de la verificación.....	124
B.5.1	Estimación del Sincrofasor	124
B.5.1.1	Reporte de Tramas	124

Diseño de una Unidad de Medición Fasorial con tecnología ARM

B.5.1.2	Reporte de Tiempos.....	124
B.5.1.3	Resultados de Ejemplo.....	124
B.5.1.4	Respuesta en el Tiempo del PMU.....	125
B.5.2	Evaluación de la Medición.....	125
B.5.2.1	Evaluación de la Medición del Sincrofasor.....	125
B.5.3	Verificación de Cumplimiento.....	126
B.5.3.1	Introducción.....	126
B.5.3.2	CLases de Desempeño.....	127
B.5.3.3	Verificación del Cumplimiento.....	127
B.5.3.4	Condiciones de Prueba y Referencia.....	127
B.5.3.5	Cumplimiento en Estado Estacionario.....	128
B.6	Formato de Mensajes del Sincrofasor.....	129
B.6.1	Solicitud de Mensajes.....	129
B.6.2	Estructura de Mensajes.....	130
B.6.2.1	Mensaje Total.....	130
B.6.2.2	Tiempo y Calidad de Tiempo.....	131
B.6.3	Trama de Datos.....	132

ÍNDICE DE FIGURAS

Figura 2.1 Representación de Onda Sinusoidal en coordenadas polares.....	9
Figura 2.2 Señal de Entrada del ejemplo.....	17
Figura 2.3 Fasor Obtenido con la DFT.....	18
Figura 2.4 Cambio de Signo esperado para iniciar el conteo del Número de Muestras.	19
Figura 2.5 Representación lineal para calcular las Fracciones de Muestra.....	20
Figura 2.6 Arquitectura del Sistema de Posicionamiento Global.....	23
Figura 2.7 Triangulación Satelital.	24
Figura 3.1 Diagrama Hardware/Software de la PMU.	28
Figura 3.2 Conexión Física de la PMU.....	29
Figura 3.3 Vista Frontal y Posterior de la TWR KM34Z50M.....	30
Figura 3.4 Diagrama a Bloques de la Tarjeta TWR KM34Z50M.	31
Figura 3.5 Diagrama a Bloques del Convertidor de Aproximaciones Sucesivas.....	33
Figura 3.6 Diagrama del Convertidor Sigma-Delta.	34
Figura 3.7 Diagrama a Bloques del ADC de Aproximaciones Sucesivas.....	35
Figura 3.8 Diagrama a Bloques de la Terminal Frontal Analógica (AFE).	36
Figura 3.9 Diagrama a Bloques del Módulo Crossbar (XBAR).....	38
Figura 3.10 Concepto del Generador de Señales K20.	40
Figura 3.11 Salida de los canales Gen0, Gen1 y Gen2 del Generador de Señales, FreeMASTER.	41
Figura 3.12 Salida de los Canales Gen4, Gen5 y Gen6 del Generador de señales, FreeMASTER.....	41
Figura 3.13 Diagrama a Bloques de la Incorporación del Osciloscopio.	41
Figura 3.14 Salidas de los Canales GEN0, GEN1 Y GEN2 del generador de señales, Osciloscopio.	42
Figura 3.15 Salidas de los Canales GEN4, GEN5 y GEN6 del Generador de Señales, Osciloscopio.....	42
Figura 3.16 Señal afectada con el 3°, 5° y 7° armónico, Osciloscopio.....	43
Figura 3.17 Señal Afectada con el 3°, 5° y 7° Armónico, FreeMASTER.....	43
Figura 3.18 Señales Desfasadas 30°, Osciloscopio.	44
Figura 3.19 Señales Desfasadas 30°, FreeMASTER.....	44
Figura 3.20 Módulo GPS.	45
Figura 3.21 Antena Externa del Módulo GPS.....	46
Figura 3.22 Vista superior del Receptor GPS.....	47
Figura 4.1 Ubicación de la Aplicación “make_project”.....	50
Figura 4.2 Ventana principal de la aplicación “make_project”.....	51
Figura 4.3 Creación del proyecto con “make_project”.....	51
Figura 4.4 Importación del Proyecto.....	52
Figura 4.5 Diagrama de Flujo del Programa Principal.....	54
Figura 4.6 Cadena de medición para dos señales trifásicas.	56
Figura 4.7 Diagrama de Secuencia del Proceso de Medición.....	57
Figura 4.8 Diagrama de Flujo del Frecuencímetro.....	59
Figura 4.9 Diagrama de Flujo del Algoritmo de la DFT.	60
Figura 4.10 Cálculo del Ángulo del Fasor.....	60
Figura 4.11 Cálculo del Ángulo en los 4 cuadrantes.....	61
Figura 4.12 Diagrama de Flujo de la Función RMS.	62
Figura 4.13 Diagrama de Flujo de la Función Raíz.....	62
Figura 4.14 Diagrama de secuencia del GPS.	63

Figura 4.15 Diagrama de Flujo de la Interrupción Externa.....	63
Figura 4.16 Diagrama de Flujo de la Recepción del Estampado.....	63
Figura 5.1 Señal de Entrada a 60 Hz, Pura.....	65
Figura 5.2 Señal de Entrada a 60 Hz con presencia de Armónicos.	67
Figura 5.3 Gráficas del Valor RMS de las señales Prueba	71
Figura 5.4 Magnitud y Ángulo del Fasor a Frecuencia Nominal.	72
Figura 5.5 Magnitud y Ángulo del Fasor a Frecuencia debajo de la Nominal.	74
Figura 5.6 Magnitud y Ángulo del Fasor a Frecuencia por encima de la Nominal.	75
Figura 5.7 Cambio de Magnitud en la Señal de Entrada.....	76
Figura 5.8 Tiempo de Respuesta del Fasor.	76
Figura 5.9 Gráfica del TVE debido a la Magnitud, Clase P.	79
Figura 5.10 Gráfica del TVE debido a la Magnitud, Clase M.	81
Figura 5.11 Gráfica del TVE debido al Cambio de Ángulo, Clases P y M.....	82
Figura 5.12 Señal con 1% de contenido de Armónicos y Magnitud del Fasor Estimado.	83
Figura 5.13 Señales con 10% de contenido Armónico y Magnitud del Fasor Estimado.	84
Figura 5.14 Pantalla de la Terminal Serial Con Mensaje de Datos.	85
Figura B.1 Senoide con periodo T observada a instantes múltiplos de T0.	122
Figura B.2 Convenio de Representación de un Sincrofasor.....	123
Figura B.3 Ejemplo del Orden de la Trama de Transmisión.	131

ÍNDICE DE TABLAS

Tabla 3.1 Modos de Operación del Módulo GPIO.....	37
Tabla 3.2 Descripción de los Bornes del Módulo GPS.....	46
Tabla 4.1 Configuración del ADC Sigma-Delta	55
Tabla 4.2 Configuración del ADC de Aproximaciones Sucesivas.....	55
Tabla 4.3 Cálculo del Ángulo Real del Fasor.	61
Tabla 4.4 Mensajes de Salida NMEA.....	64
Tabla 4.5 Significado de los Campos del Mensaje del GPS utilizados.....	64
Tabla 5.1 Cálculo de la Frecuencia de Señal de entrada Sinusoidal Pura.	66
Tabla 5.2 Resultados de Medición de Frecuencia, Señal de Entrada Sinusoidal Pura.....	66
Tabla 5.3 Cálculo de la Frecuencia de Señal de entrada Sinusoidal con Armónicos.	67
Tabla 5.4 Resultados de Medición de la Frecuencia, Señal con contenido Armónico.	68
Tabla 5.5 Cálculo de la Frecuencia de Señal con Frecuencia por encima y por debajo de la Nominal.	68
Tabla 5.6 Resultados del Cálculo del Valor RMS.	69
Tabla 5.7 Resultados del cálculo del valor RMS para determinar Precisión.....	70
Tabla 5.8 Resultados TVE para el cambio de Magnitud de Voltaje y Corriente, Clase P.....	78
Tabla 5.9 Resultados TVE para el Cambio de Magnitud de Voltaje, Clase M.....	80
Tabla 5.10 Resultados TVE para el Cambio de Ángulo, Clases P y M.....	82
Tabla 5.11 Resultados TVE para señal con 1% de Distorsión Armónica, Clase P.	83
Tabla 5.12 Resultados TVE para Señal con 10% de Distorsión Armónica, Clase M.	84
Tabla 5.13 Mensaje de la Trama de Configuración.	86
Tabla 5.14 Mensaje de la Trama de Cabezal.	87
Tabla 5.15 Mensaje de la Trama de Comando.....	87
Tabla 5.16 Mensaje de la trama de datos.	88
Tabla B.1 Razón de reporte de tramas requerida por el PMU.....	124
Tabla B.2 Valores de Sincrofasor para un sistema de 60 Hz con frecuencia de muestreo de 10 tramas por segundo.....	125
Tabla B.3 Requerimientos de Mediciones del Sincrofasor en Estado Estacionario	129
Tabla B.4 Definiciones comunes de palabras para todo tipo de tramas	131
Tabla B.5 Definición de los Bits de la bandera de Calidad de tiempo.....	132
Tabla B.6 Organización de la Trama de Datos.....	133

GLOSARIO

A	Ampers.
ADC	Analog-to-Digital Converter / Convertidor Analógico Digital.
AFE	Analog Frontal End / Terminal Frontal Analógica.
ANNMR	Número de valores analógicos.
ARM	Advanced RISC Machine / Máquina Avanzada RISC.
CA	Corriente Alterna.
CD	Corriente Directa.
COCO	Conversion Complete / Conversión Completa.
cos	Coseno.
CRC	Código de Redundancia Cíclica.
DFT	Discrete Fourier Transform / Transformada Discreta de Fourier.
DGNMR	Número de palabras de estado digital.
GPIO	General Purpose Input Output / Entrada/Salida de Propósito General.
GPS	Global Positioning System / Sistema de Posicionamiento Global.
Hz	Hertz.
IDCODE	Código de identificación del PMU.
LED	Light Emitting Diode / Diodo Emisor de Luz.
LUT	Look-up Table / Tabla de Búsqueda
NMEA	National Marine Electronics Association / Asociación Nacional de Electrónica Marina.
PC	Personal Computer / Computadora Personal.
PHNMR	Número de fasores.
PMU	Phasor Measurement Unit / Unidad de Medición Fasorial.
pps	Pulse Per Second / Pulso Por Segundo.
RAM	Random Access Memory / Memoria de Acceso Aleatorio.

RISC	Reduced Instruction Set Computer / Ordenador con Conjunto Reducido de Instrucciones.
RMS	Root Mean Square / Valor Eficaz (Raíz Cuadrática Media).
SAR	Successive Approximations Register / Registro de Aproximaciones Sucesivas.
SEL	Schweitzer Engineering Laboratories.
sen	Seno.
TVE	Total Vector Error / Vector Total de Error.
UART	Universal Asynchronous Receiver-Transmitter / Receptor-Transmisor Universal Asíncrono.
UTC	Universal Time Coordinated / Tiempo Universal Coordinado.
V	Volts.

CAPÍTULO 1 INTRODUCCIÓN

1.1 GENERALIDADES

Actualmente, y como resultado del aumento del uso de convertidores estáticos, la contaminación en la redes eléctricas se ha incrementado, por tal motivo los instrumentos convencionales basados en la medición de señales sinusoidales han quedado obsoletos y surge la necesidad de utilizar sistemas de medición que logren sensar con mayor precisión las señales de las redes eléctricas. [2]

Los fasores proveen medición directa de las unidades fundamentales de voltaje y corriente, incluyendo magnitud y ángulo. El ángulo es medido con respecto a un tiempo de referencia. Los datos no son normalmente coincidentes en el tiempo, pero son recolectados sobre un periodo de algunos segundos. Bajo condiciones normales cercanas a las de operación en estado estacionario, estos datos son aproximadamente coincidentes. [1]

Para llevar a cabo la medición de fasores en una red interconectada, se requiere de una fuente de sincronización, que puede ser local o global, es decir, el muestreo de datos en dos puntos de la red está sincronizado. Si además se cuenta con el Tiempo Universal Coordinado (UTC, *Universal Time Coordinated*) del proceso, entonces se puede enviar la medición a un lugar remoto acompañándolo del estampado de tiempo. Al alinear el estampado de tiempo de las mediciones obtenidas de diferentes nodos, se obtienen mediciones simultáneas. [16] El sistema de posicionamiento global (GPS, *Global Positioning System*) permite realizar la sintonización de las mediciones.

Dado que existen defasamientos en las señales de voltaje y de corriente en la transmisión de la energía eléctrica a través de los diferentes elementos de la red eléctrica, se requiere un dispositivo que provea mediciones sincronizadas en dicha red. Las mediciones sincronizadas de fasores abren la posibilidad de sintetizar el proceso de estimación de estado. [11]

Las Unidades de Medición Fasorial (PMU, *Phasor Measurement Units*) permiten obtener mediciones fasoriales de corriente y de voltaje sincronizadamente en diferentes puntos de la red y

transmitir la información de los fasores al centro de control, donde pueden ser comparadas, evaluadas y procesadas. [3]

1.2 OBJETIVO DE LA TESIS

Diseñar una unidad de medición fasorial utilizando un microcontrolador de tecnología ARM con convertidores de 16 y 24 bits, que sea capaz de proporcionar mediciones fasoriales de corriente y voltaje sincronizadas vía satélite mediante el uso de un módulo de posicionamiento global.

1.2.1 OBJETIVOS PARTICULARES

- ◇ Desarrollar los algoritmos necesarios para la medición de fasores, en el ambiente de trabajo CodeWarrior V10.6.
- ◇ Implementar el estampado de tiempo mediante el uso del pulso por segundo (pps) proporcionado mediante el receptor GPS, a fin de obtener mediciones sincronizadas.

1.3 JUSTIFICACIÓN

En todas las aplicaciones del sistema eléctrico, se tiene como primera instancia la medición de las variables que se necesitan para realizar el análisis y el control de dicho sistema. Las aplicaciones de monitoreo y protección de sistemas requiere el intercambio de información entre los equipos y aplicaciones. En cada caso, los procedimientos utilizados para el cálculo, registro y transmisión de fasores de voltaje y corriente se realizan de forma diferente. [4]

La mayor parte de las herramientas existentes estiman la distribución de fasores a partir de medidas no sincronizadas realizadas en diversos puntos de la red. La medición de fasores sincronizados tiene un gran potencial de aplicaciones en sistemas eléctricos de potencia, tales como la medición de frecuencia y magnitud de voltaje y corriente, estimación de estado, así como control y monitorización de la red. [24]

La medición de los fasores de voltaje en tiempo real mejora la capacidad de respuesta de los operadores y permite el desarrollo de sistemas de prevención de inestabilidades y de respuesta automática más rápida y eficaz. [4]

En el caso de líneas críticas, las mediciones se sincronizan mediante algún mecanismo que brinde esquemas de protección diferencial para la detección de fallas. Así, se considera que la protección diferencial es la forma de protección más fiable. En el futuro, las unidades PMU podrían ser utilizadas para suministrar protección diferencial. [10]

Entonces, se justifica el siguiente trabajo mediante la utilización de la tecnología ARM para el desarrollo del PMU, sabiendo que los medidores fasoriales sincronizados vía satélite no se fabrican en México.

1.4 ESTADO DEL ARTE

En 1983, Phadke, Thorp y Adamiak [7] describen los fundamentos teóricos y el proceso computacional para el cálculo de fasores en tiempo real. La Transformada Discreta de Fourier (DFT) se utiliza para la obtención de fasores.

En 1988 se construye, en el Tecnológico de Virginia, la primera Unidad de Medición Fasorial (PMU). Dicha unidad se sincroniza mediante el Sistema Satelital de Posicionamiento Global (GPS). Esta PMU empieza a ser comercializada por la compañía Macrodyne. [8]

En 1993 Phadke describe la PMU y menciona las diferentes formas de sincronización de las mismas, destacando la forma de sincronización del Sistema de Posicionamiento Global, que tiene una exactitud de $1\mu\text{s}$. [8,9]

En 1994 se utilizan las PMU's sincronizadas vía satélite para la obtención de fasores en la red de la compañía de Potencia Georgia. [6] En el mismo año, se utilizan las PMU's sincronizadas satelitalmente para confirmar la ecuación del flujo de potencia en líneas de transmisión. [12]

Las aplicaciones de las PMU's sincronizadas satelitalmente se extendieron desde entonces, utilizándose para estimación de estado, predicción de estabilidad de líneas, control y monitoreo. [5]

En el 2000, Yutaka Ota y Hideki Fujita utilizan unidades de medición fasorial para llevar a cabo la evaluación de la estabilidad de un sistema de potencia, para llevar a cabo las mediciones se utiliza un microcontrolador DSP y además se menciona que la fuente de sincronización es el pulso por segundo que proporciona un receptor del Sistema de Posicionamiento Global. [14]

En 2004 se desarrolla en México una PMU sincronizada vía satélite por Rafael San Vicente. En su aplicación utiliza un microcontrolador DSP de Texas Instruments y se describe el proceso completo de medición. Este dispositivo es capaz de medir en tiempo real el desfase entre una o varias señales de voltaje o corriente. [14,15]

En 2005, en México se desarrolla una PMU sincronizada vía satélite por Carlos Cuvas. En su aplicación, utiliza un microcontrolador DSP de Freescale y se describe el proceso completo de medición, calculando fasores de voltaje y de corriente de un sistema trifásico, enviando dichos valores mediante un puerto serial RS-232 a una PC. [16]

En 2009, se desarrolla en México una PMU sincronizada vía satélite por Blanca Verónica Hernández, utilizando un microcontrolador de ColdFire y se describe el proceso completo de medición, que cumple con el estándar IEEE C37.118. [18]

En 2013, Abraham Valdiosera Marroquín desarrollo un medidor inteligente con sistema de comunicación bidireccional, utilizando un microcontrolador ColdFire junto con un receptor GPS para obtener mediciones fasoriales sincronizadas. Se incluye una etapa de comunicación inalámbrica mediante el uso de tecnología Zig-Bee. [17]

En 2015, David Sebastián Cárdenas desarrolla un dispositivo de medición inteligente, que permite la detección del robo de energía cumpliendo con el estándar IEEE 1459 y con los requerimientos de la clase "M" basado en el estándar IEEE C37.118 incluyendo un receptor GPS. [19]

Actualmente la compañía SEL cuenta con dispositivos de medición de fasores sincronizados, instalados a través del sistema eléctrico de potencia de los Estados Unidos de América, Monterrey y la Ciudad de México. [23]

1.5 ALCANCES Y LIMITACIONES

La PMU implementada será capaz de realizar mediciones fasoriales de voltaje y de corriente trifásicas con formas de onda sinusoidales a una frecuencia nominal de 60 Hz. El rango de las mediciones será del 10% al 120% de la señal de voltaje permitido en las entradas del microcontrolador y del 10% al 200% de la señal de corriente permitida en las entradas del microcontrolador, señales que son entregadas por el microcontrolador generador de señales, con lo que cumple con la condición de estado estacionario del Estándar IEEE C37.118.1.

Empleando los fasores se mide el defasamiento entre voltajes y corrientes, implementando también la medición de la frecuencia. La sintonización se realizará mediante el pulso por segundo que proporciona el módulo del Sistema de Posicionamiento Global. El GPS proporciona el Tiempo Universal Coordinado (UTC) que, junto con las mediciones, se transmite el estampado de tiempo del pulso de sintonización.

1.6 APORTACIONES

- ◇ Proporcionar una Unidad de Medición Fasorial con tecnología ARM, sincronizada vía satélite, que sirva para implementar otros sistemas de medición y aplicarlos en control, estimación de estado, monitoreo o predicción de estabilidad de una red interconectada.
- ◇ Proporcionar diagramas de flujo, plantillas y algoritmos en lenguaje C, en el entorno de CodeWarrior y que puedan ser modificados y/o adecuados a una aplicación en específico.
- ◇ Proporcionar las pruebas realizadas al microcontrolador generador de señales, que determinan su uso como un emulador de la red real trifásica de corriente y voltaje.

1.7 ESTRUCTURA DE LA TESIS

En el presente capítulo (Capítulo 1) se aborda el problema a resolver, mencionando el objetivo, la justificación, las aportaciones y la estructura de la tesis. También, se incluye el estado del arte de la misma. Se da una breve reseña general de lo que se conoce, de los elementos prioritarios utilizados en la solución del problema; así mismo, se plantea el alcance y las limitaciones del trabajo. Se hace la descripción del problema a resolver, indicando sus antecedentes y la situación actual del tema. Finalmente, se describe la estructura del trabajo por capítulos.

En el Capítulo 2, se presentan las bases teóricas acerca de los conceptos que conciernen a la medición fasorial sincronizada, entre los cuales se encuentra la Transformada Discreta de Fourier para la obtención de fasores y el Sistema de Posicionamiento Global.

En el Capítulo 3, se hace la descripción de las características principales y elementos de funcionamiento de los elementos del Hardware utilizado en la implementación del medidor fasorial. Aunado a esto, se describe la interacción de los elementos mencionados y la interconexión de los mismos. También se muestran las características de la tarjeta de desarrollo empleada como una imagen de la conexión entre la tarjeta de desarrollo, la PC y el módulo receptor GPS.

El Capítulo 4 presenta los elementos de Software utilizados, así como el diagrama a bloques de una PMU y el diagrama conceptual a implementar. Se muestran los diagramas de flujo de las funciones, el cálculo de la DFT, y se describe el ambiente de desarrollo del microcontrolador.

En el capítulo siguiente (Capítulo 5) se realiza el reporte de los resultados de las pruebas realizadas a la PMU, verificando que cumple con los requerimientos necesarios, ante diferentes condiciones en estado estacionario. En el Capítulo 6, se reportan las conclusiones de la implementación y las recomendaciones para trabajos futuros en esta línea de estudio.

En la parte final del trabajo, se muestran los apéndices. En el apéndice A se tienen los códigos de programación utilizados en el desarrollo de este trabajo y en el Apéndice B se tiene una breve descripción del estándar IEEE C37.118, que sirve para tener un panorama de los requisitos primordiales con los que debe contar una Unidad de Medición Fasorial.

CAPÍTULO 2 SISTEMA DE MEDICIÓN DE FASORES

En este capítulo, se abordan y analizan los diferentes conceptos relacionados con la medición fasorial, comenzando con los métodos que se utilizan en sistemas eléctricos de potencia para calcular fasores de corriente y voltaje, y que aportan la magnitud y la fase de una señal. Se describe el uso de la Transformada Discreta de Fourier para obtener los fasores, el funcionamiento del Sistema de Posicionamiento Global para la sincronización de medición de dichos fasores.

2.1 INTRODUCCIÓN

Para monitorear y diagnosticar sistemas eléctricos de potencia, la medición fasorial es muy importante. Los fasores representan la amplitud y la fase de una señal sinusoidal en estado estable (frecuencia, amplitud y fase constante). [14]

Los cálculos de magnitud y fase corresponden a la transformación de coordenadas rectangulares a polares, cuya representación permite la visualización de los desplazamientos de los ángulos de fase. [18]

Con formas de onda en tiempo real, es necesario definir un tiempo de referencia para medir los ángulos de fase en forma sincronizada. El estándar IEEE 1344-1995 define el inicio de un segundo como el tiempo de referencia para establecer el valor del ángulo del fasor. [13]

Las posibilidades de determinar los fasores a lo largo del sistema de distribución o transmisión de energía en un momento dado, abre una puerta a la solución de estos problemas. Por ello, es de suma importancia analizar y estudiar las técnicas de estimación de fasores. [18]

2.2 CONCEPTOS BÁSICOS

Para poder implementar cualquier algoritmo de estimación fasorial, es necesario estudiar algunos conceptos básicos en el tema del procesamiento de señales digitales, los cuales se describen a continuación.

2.2.1 VALOR RMS

El valor eficaz (RMS, *Root Mean Square*) o valor cuadrático medio, de cualquier corriente periódica, resulta igual al valor de la corriente directa que al fluir a través de una resistencia de R ohms, ambas disipan la misma potencia promedio. Para procesar señales analógicas por medios digitales, es necesario convertirlas a formato digital, es decir, transformarlas en una secuencia de números de precisión finita.

El cálculo del valor eficaz para una señal periódica se describe con la ecuación 2.1.

$$X_{RMS} = \sqrt{\frac{1}{T} \int_0^T [x(t)]^2 dt} \quad (2.1)$$

En donde:

X_{RMS} es el valor eficaz de la señal.

T es el periodo de la señal bajo estudio.

$x(t)$ representa la señal analógica.

Para el análisis en tiempo discreto, se utiliza la modificación de la ecuación (2.1), que resulta:

$$X_{RMS} = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} (x[n])^2} \quad (2.2)$$

En donde:

X_{RMS} es el valor eficaz de la señal.

N es el número de muestras de la ventana de datos.

$x(n)$ representa la función periódica en tiempo discreto.

2.2.2 CONCEPTO DE FASOR

Para un mejor análisis de la corriente alterna (CA), que está representada por la figura 2.1.a, siendo su expresión $A\sin(\omega t + \phi)$, dicha expresión puede ser transformada a un sistema de coordenadas polares, tal y como se muestra en la figura 2.1.b.



FIGURA 2.1 REPRESENTACIÓN DE ONDA SINUSOIDAL EN COORDENADAS POLARES.

La explicación reside en que el radio gira como un vector rotatorio con frecuencia ω describiendo un círculo. La longitud del radio constituye la amplitud de la onda, en este caso del voltaje. La componente vertical tiene el valor $A\sin(\omega t + \phi)$ en la curva sinusoidal de la corriente alterna. Este radio giratorio se conoce como fasor. [18]

2.2.3 CONCEPTO DE FASE

Fase es una medida de la diferencia de tiempo entre dos ondas sinusoidales. Aunque la fase es una diferencia verdadera de tiempo, siempre es medida en términos de ángulo, en grados o radianes. Eso es una normalización del tiempo que requiere un ciclo de la onda sin considerar su verdadero periodo de tiempo. [21]

La diferencia en fase entre dos formas de onda se conoce con el nombre de defasamiento. Un defasamiento de 360° es un atraso de un ciclo o de un periodo de la onda, lo que realmente no es ningún desplazamiento. Un defasamiento de 90° es un desplazamiento de un cuarto del periodo de

la onda. [16] Existen fenómenos llamados defasamiento en atraso y defasamiento en adelanto, eso quiere decir que una forma de onda puede ser retrasada respecto a otra o una forma de onda puede ser adelantada respecto a otra. [25]

Para medir la fase, se puede utilizar una señal externa, como el caso de un pulso por segundo proporcionado por un módulo receptor GPS. [8]

2.2.4 MUESTREO DIGITAL

El muestreo digital es un proceso de adquisición de datos, a intervalos regulares de tiempo, consistente en la obtención del valor que toma la señal original en un momento dado. [29,30]

El parámetro fundamental del muestreo digital es el intervalo de muestreo Δs , o su equivalente la frecuencia de muestreo $\frac{1}{\Delta s}$ Hz. Cuanto menor sea Δs , mayor número de valores se obtendrán de la señal, y viceversa. [27]

El resultado de dicho muestreo es la obtención de una serie discreta ordenada $\{x_r\} = \{x_0, x_1, x_3, \dots, x_r, \dots\}$, en la que el índice r indica la posición de orden temporal del valor de x_r . Así, el valor de la señal original, en el tiempo $t = \Delta r$, $x(t)$, se representa por x_r . A la señal continua de origen se le llama serie temporal continua, mientras que a la serie obtenida por el muestreo se le llama serie temporal discreta. [26]

2.2.5 TEOREMA DEL MUESTREO

Según el teorema de muestreo de Nyquist-Shannon, para poder reconstruir con exactitud la forma de onda, es necesario que la frecuencia de muestreo (f_Δ) sea como mínimo el doble de la máxima frecuencia (f_{max}) a muestrear. [21]

El teorema de Nyquist indica el valor mínimo necesario para reconstruir la señal original, sin embargo, mientras más muestras se tengan, se podrá reconstruir mejor la señal. [26]

Entonces, para que el muestreo sea correcto, se debe escoger la frecuencia de muestreo de tal forma que:

$$f_{\Delta} \geq 2f_{max} \quad (2.3)$$

Normalmente, se suele filtrar la señal de entrada para eliminar las frecuencias que no pueden ser detectadas mediante el proceso del muestreo, para evitar ver frecuencias no existentes (alias). [25, 29, 30, 31]

También, es conveniente que la frecuencia de muestreo sea muy superior al doble de la frecuencia máxima de la señal bajo estudio, puesto que ningún filtro pasa-bajos puede eliminar completamente las frecuencias superiores a la frecuencia de corte. [16] En la práctica, se suele usar la frecuencia de muestreo de entre 5 y 10 veces la frecuencia de corte del filtro pasa-bajos. [21] En este trabajo es 100 veces la frecuencia de la nominal.

2.3 TRANSFORMADA DISCRETA DE FOURIER

El análisis de espectros, definido como la transformación de una señal de la representación en el dominio del tiempo hacia la representación en el dominio de la frecuencia, tiene sus raíces a principio del siglo XIX, cuando varios matemáticos lo investigaron desde una base teórica. [28]

Se dice que la señal de entrada se encuentra en el dominio del tiempo, es decir, muestras tomadas en un periodo de tiempo establecido, o señal discreta de la onda que será transformada, El término dominio de la frecuencia es usado para describir las amplitudes y fases de las frecuencias que componen la señal de entrada. [26]

El dominio de la frecuencia contiene exactamente la misma información que el dominio del tiempo, sólo que en diferente forma. Si se conoce uno de los dominios, se puede conocer el otro.

El análisis en frecuencia de una señal implica su descomposición en sus componentes espectrales, tanto en magnitud como en ángulo de fase. El espectro es una característica única de una señal que proporciona la información suficiente para poder definirla completamente, por ejemplo, la frecuencia fundamental, la magnitud a la frecuencia fundamental, magnitud de sus componentes

armónicas, ángulos de fase, presencia de otras señales o ruido; así, el análisis de Fourier es una herramienta útil para obtener y analizar el espectro de una señal. [18]

En el caso de contar con la señal en el dominio del tiempo, el proceso para conocer el dominio de la frecuencia es llamado descomposición o análisis o Transformada Discreta de Fourier.

2.3.1 DEFINICIÓN DE LA TRANSFORMADA DISCRETA DE FOURIER

Ni la Serie de Fourier, ni la Transformada de Fourier se prestan fácilmente para cálculos en computadoras digitales. Para vencer este impedimento se desarrolló la Transformada Discreta de Fourier (DFT, *Discrete Fourier Transform*). [26]

La DFT opera con una señal muestreada. A partir de ésta, se genera un espectro en el dominio de la frecuencia. El espectro que resulta es una aproximación de la Serie de Fourier, en el sentido que se pierde la información entre muestras de la forma de onda. [25]

Dada una secuencia de entrada arbitraria, se pueden obtener representaciones de la forma:

$$x[n] = \sum \alpha_k e^{j\omega_k n} \quad (2.4)$$

Se tiene la siguiente ecuación:

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n} \quad (2.5)$$

La cuál se conoce como Transformada de Fourier, o transformada de Fourier en Tiempo Discreto (DTFT, *Discrete Time Fourier Transform*). Es una expresión para calcular $X(e^{j\omega})$ a partir de la secuencia $x[n]$, es decir, para analizar la secuencia $x[n]$ determinando qué cantidad de cada componente en frecuencia es necesaria para sintetizar $x[n]$ utilizando la ecuación:

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega}) e^{j\omega n} d\omega \quad (2.6)$$

Conocida como la Inversa de la Transformada Discreta de Fourier. [27]

El número de muestras en el dominio del tiempo se representa por la variable N , el cual puede tomar cualquier valor entero positivo, pero usualmente se eligen 32, 64, 128, 256, 512, etc., debido a que los datos guardados digitalmente en una computadora usan direccionamiento binario. [26]

El dominio de la frecuencia ($X[k]$) consta de dos partes, cada una es un arreglo de $N/2 + 1$ localidades; dichas partes son: la parte real de $X[k]$ ($Re X[k]$), que es el arreglo en donde se encuentran las amplitudes de las formas coseno, y la parte imaginaria de $X[k]$ ($Im X[k]$) que son los valores de las formas seno. [26]

2.3.2 FUNCIONES BASE DE LA TRANSFORMADA DISCRETA DE FOURIER

Las formas seno y coseno usadas en la DFT son llamadas comúnmente funciones base de la DFT. Éstas, son un conjunto de formas de onda seno y coseno de amplitud unitaria. [26]

Las funciones base son generadas de las ecuaciones:

$$c_k[i] = \cos(2\pi ki / N) \quad (2.7)$$

$$s_k[i] = \sin(2\pi ki / N) \quad (2.8)$$

En donde:

N es el número de muestras.

k es el coeficiente para cada una de las frecuencias, desde 0 hasta $N/2$.

i es el coeficiente para los puntos de la señal seno y coseno de referencia; varía de 0 hasta $N-1$.

c_k es la forma de onda cosenoidal para obtener la amplitud de $Re X[k]$.

s_k es la forma de onda sinusoidal para la amplitud de $Im X[k]$.

2.3.3 INVERSA DE LA TRANSFORMADA DISCRETA DE FOURIER

La Ecuación para la inversa de la Transformada Discreta de Fourier [26] aplicando la relación de Euler es:

$$x[i] = \sum_{k=0}^{N/2} \text{Re } \bar{X}[k] \cos(2\pi ki / N) + \sum_{k=0}^{N/2} \text{Im } \bar{X}[k] \sin(2\pi ki / N) \quad (2.9)$$

En donde:

$x[i]$ es la señal resultante de la inversa de la transformada discreta de Fourier.

i es el coeficiente para puntos de la señal seno y coseno de referencia, que varía de 0 hasta $N - 1$.

$\text{Re } \bar{X}[k]$ contiene la amplitud de la forma coseno.

$\text{Im } \bar{X}[k]$ contiene la amplitud de la forma seno.

k es el coeficiente para cada una de las frecuencias, desde 0 hasta $N/2$.

Así, cada punto N de la señal es creado mediante la adición de $N/(2 + 1)$ señales coseno y seno. Las amplitudes de las formas coseno y seno están contenidas en los arreglos $\text{Re } \bar{X}[k]$ y $\text{Im } \bar{X}[k]$, respectivamente. [16]

Para la ecuación de la Transformada Inversa de Fourier, se multiplican estas amplitudes por las funciones base para crear un conjunto de formas escaladas de onda coseno y seno.

Adicionando las formas escaladas coseno y seno, se produce la señal en el dominio del tiempo $x[i]$.

El escalamiento se realiza mediante las ecuaciones siguientes:

$$\text{Re } \bar{X}[k] = \frac{\text{Re } X[k]}{N/2} \quad (2.10)$$

$$\text{Im } \bar{X}[k] = \frac{\text{Im } X[k]}{N/2} \quad (2.11)$$

Exceptuando los casos:

$$\text{Re } \bar{X}[0] = \frac{\text{Re } X[0]}{N/2} \quad (2.12)$$

$$\text{Re } \bar{X}[N/2] = \frac{\text{Re } X[N/2]}{N/2} \quad (2.13)$$

En donde:

$\text{Re } \bar{X}[k]$ e $\text{Im } \bar{X}[k]$ contienen las amplitudes de las formas seno y coseno necesarias para la transformada inversa de Fourier.

$\text{Re } X[k]$ e $\text{Im } X[k]$ contienen la parte real e imaginaria del dominio de la frecuencia.

2.3.4 CÁLCULO DE LA DFT POR CORRELACIÓN

La DFT puede ser calculada en tres formas diferentes [29, 30]:

- ◇ Ecuaciones simultáneas.
- ◇ Transformada Rápida de Fourier.
- ◇ Correlación.

La primera requiere demasiados cálculos y casi no es utilizada [26]. La segunda, es un ingenioso algoritmo que descompone la DFT con N puntos en N DFTs, cada una con un solo punto. En esta sección se analizará el cálculo de la DFT por correlación.

La correlación es una operación matemática la cual usa dos señales de entrada para obtener una tercera. La correlación es una técnica óptima para detectar una forma de onda conocida en una que contiene ruido. [26]

El algoritmo para obtener la DFT mediante la correlación, con N muestras de la señal de entrada, se basa en las siguientes ecuaciones:

$$x(k) = \sum_{k=-\infty}^{\infty} a_k e^{-jK\omega_0 n} \quad (2.14)$$

$$\text{Re } X = \frac{\sum_{i=0}^{N-1} \text{muestra}(i) * \cos(i)}{N / 2} \quad (2.15)$$

$$\text{Im } X = -\frac{\sum_{i=0}^{N-1} \text{muestra}(i) * \text{sen}(i)}{N / 2} \quad (2.16)$$

$$\text{Magnitud} = \sqrt{(\text{Im } X)^2 + (\text{Re } X)^2} \quad (2.17)$$

$$\text{Fase} = \text{arctg} \frac{\text{Im } X}{\text{Re } X} \quad (2.18)$$

En donde:

$x(k)$ representa la señal analógica en tiempo discreto.

N es el número de muestras a cuantificar del Convertidor Analógico Digital (ADC, *Analog-to-Digital Converter*).

i es el índice para efectuar el producto de los elementos uno a uno, de cero hasta $N - 1$.

$\text{Re } X$ e $\text{Im } X$ son las componentes rectangulares del vector correspondiente a la descomposición de la señal analógica.

Dicho de otro modo, cada muestra del dominio de la frecuencia es encontrada haciendo la multiplicación de cada uno de los puntos de señal del dominio del tiempo por la forma coseno o seno (parte real o imaginaria) y acumulando los resultados.

Ahora, se debe definir cuál será el valor de la frecuencia nominal y cuántas muestras se requerirán por ciclo. Para ejemplificar, se consideran los siguientes valores:

- ◇ Frecuencia Nominal: 60 Hz.
- ◇ Número de muestras por ciclo: 12.

Con dichos valores, es posible calcular la frecuencia de muestreo, utilizando la ecuación siguiente:

$$f_{muestreo} = f_{nom} * N \quad (2.19)$$

Sustituyendo los valores de la frecuencia nominal y el número de muestras, se tiene una frecuencia de muestreo de 720 Hz. Analizando una señal de entrada, como ejemplo, en la que se tiene la presencia de armónicos (segundo y tercer armónico), mostrada en la figura 2.2.

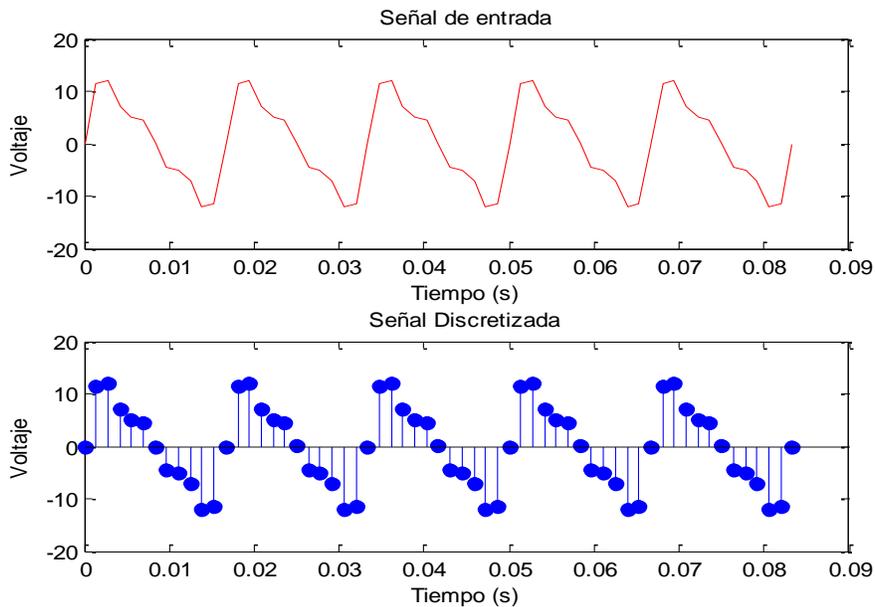


FIGURA 2.2 SEÑAL DE ENTRADA DEL EJEMPLO.

La transformada discreta de Fourier tiene filtros que eliminan de manera completa los armónicos y, consecuentemente, se obtiene un fasor lineal [47], cuyo tiempo de estabilización es de aproximadamente 16.666 milisegundos, tiempo en que se obtienen las primeras 12 muestras. Aplicando el algoritmo de la DFT. Se obtienen la magnitud del fasor estimado (figura 2.3.a) y el ángulo del mismo (figura 2.3.b).

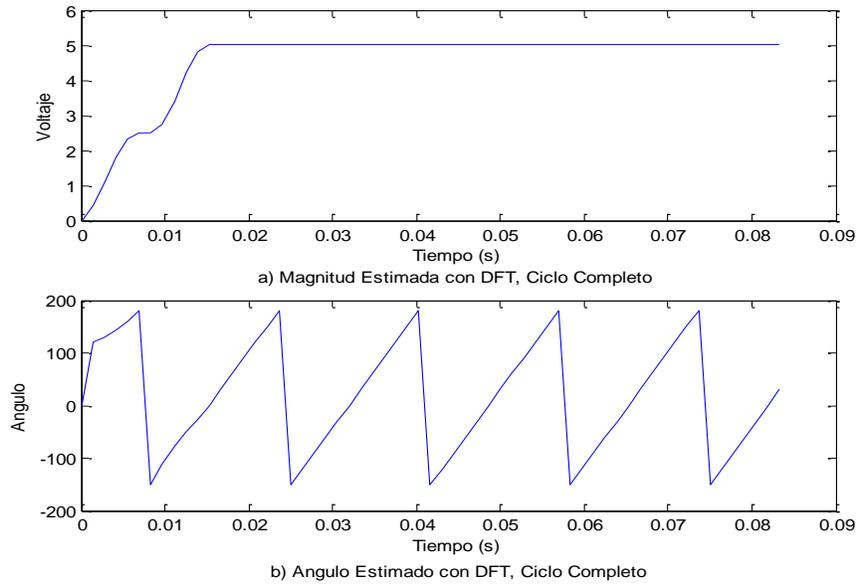


FIGURA 2.3 FASOR OBTENIDO CON LA DFT.

Como se observa en la figura 2.3.a, la DFT elimina por completos las armónicas presentes en la señal de entrada, estabilizándose el fasor en un tiempo aproximado de 16.666 ms, que es el periodo de un ciclo de la señal de entrada y es el necesario para que este algoritmo determine el fasor. En la figura 2.3.b se observa que el ángulo del fasor estimado tampoco se afecta por la contaminación armónica de la señal de entrada, tardando un ciclo completo en estabilizarse.

Entre las diferentes ventajas que presenta la DFT se encuentra el hecho de que atenúa efectivamente el ruido de la señal de entrada, así como las armónicas pares e impares como las componentes de alta frecuencia. La DFT es el algoritmo más adecuado para el cálculo de fasores, ya que elimina el ruido, las armónicas y a pesar de que tarda un ciclo en actualizar las mediciones, este tiempo no afecta la ejecución del programa. [47]

2.4 MEDICIÓN DE FRECUENCIA

Un contador de frecuencia permite medir y visualizar la frecuencia de una señal. [22] Para realizar el cálculo de la frecuencia del sistema, se tienen métodos diferentes; sin embargo se utilizó el método de cruce por cero, que se explica en la sección siguiente.

2.4.1 CÁLCULO DE LA FRECUENCIA POR CRUCE POR CERO

El algoritmo del cálculo de la frecuencia, mediante el método de cruce por cero, radica en la detección, como su nombre lo indica, de un cruce por cero de la señal de entrada con pendiente negativa, como se muestra en la figura 2.4; en ese momento, se cuenta el número de muestras enteras hasta que aparezca el siguiente cruce por cero. [18]

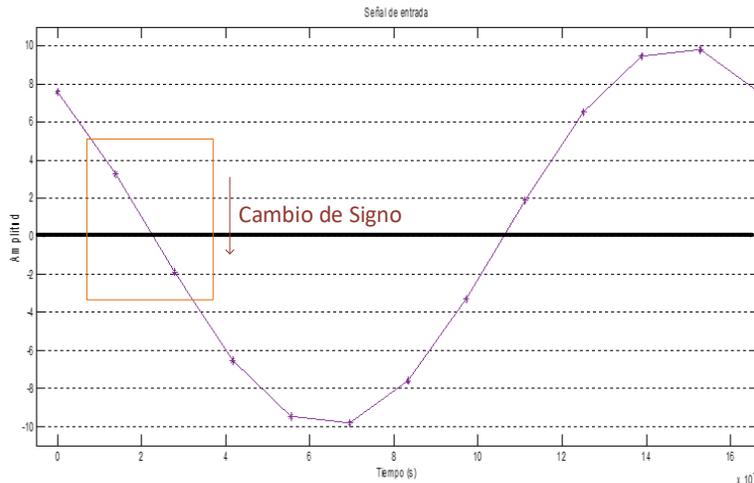


FIGURA 2.4 CAMBIO DE SIGNO ESPERADO PARA INICIAR EL CONTEO DEL NÚMERO DE MUESTRAS.

Es evidente que el número de muestras pertenecientes a un ciclo no es un número entero, por lo que es necesario calcular las fracciones de muestra obtenidas antes y después del cruce por cero, para obtener la cantidad fraccionaria de muestras dentro del ciclo. Así, sumando las muestras enteras y las fraccionarias, se obtiene el número exacto de muestras de cada ciclo.

Para obtener el valor de la fracción de muestra anterior y posterior al cruce, se emplea la ecuación de la recta, como se muestra en la figura 2.5.

Diseño de una Unidad de Medición Fasorial con tecnología ARM

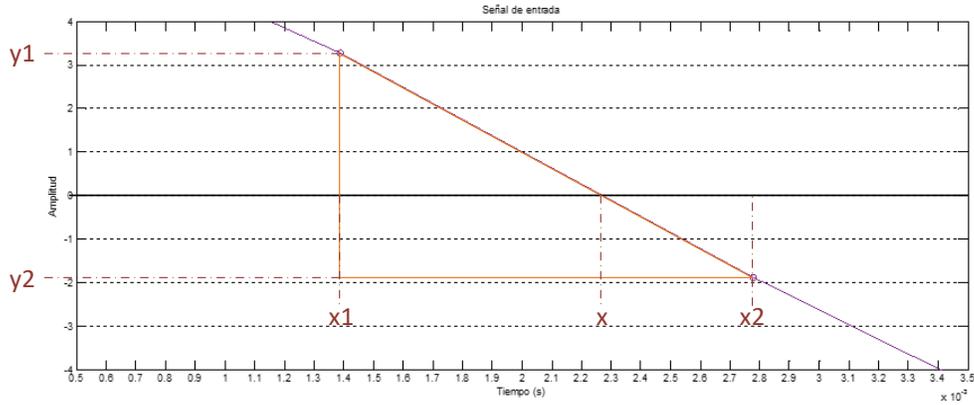


FIGURA 2.5 REPRESENTACIÓN LINEAL PARA CALCULAR LAS FRACCIONES DE MUESTRA.

El intervalo que se desea conocer es el de x_1 a x , por lo que se utiliza la ecuación de la pendiente de la una recta, descrita por las ecuaciones 2.17 y 2.18:

$$y = mx + b \tag{2.20}$$

$$m = \frac{y_2 - y_1}{x_2 - x_1} \tag{2.21}$$

En donde:

m es la pendiente

y_1 es la muestra anterior.

y_2 es la muestra actual.

x_1 Es el tiempo de muestreo anterior.

x_2 es el tiempo de muestreo actual.

El intervalo entre x_1 y x_2 corresponde a un periodo de muestreo, es decir:

$$x_2 - x_1 = 1 \tag{2.22}$$

Analíticamente, si se necesita determinar el valor de x en el momento en que $y = 0$ y $b = y_1$, y sabiendo que $x_2 - x_1 = 0$, se hace uso de las ecuaciones 2.17 y 2.18, teniendo entonces:

$$y = mx + b \quad \rightarrow \quad 0 = mx + y_1$$

Despejando a x resulta:

$$x = \frac{-y_1}{m}$$

$$\rightarrow x = \frac{-y_1}{\frac{y_2 - y_1}{x_2 - x_1}}$$

$$\Delta fa = x = \frac{-y_1}{y_2 - y_1} \quad (2.23)$$

En donde Δfa es la fracción de muestra anterior al cambio de signo.

Por lo tanto, para calcular la fracción de muestra posterior al cambio de signo, es necesario determinar la parte complementaria del total de la muestra a la que pertenece la fracción de muestra anterior, ya calculada. Es decir, la suma de la fracción de muestra anterior y posterior es igual a la unidad. [18] Así:

$$\Delta fp = 1 - \Delta fa \quad (2.24)$$

En donde Δfp es la fracción de muestra posterior al cambio de signo. Cabe mencionar que, tras realizar la programación de este algoritmo, se obtuvo que el primer resultado arrojado por éste, es erróneo ya que no se tiene conocimiento del instante exacto del ciclo en que comienza a ejecutarse. Después de ello, el resultado brinda valores satisfactorios, mostrando el valor de la frecuencia fundamental, sin identificar la presencia de armónicos en la señal de entrada.

2.5 SISTEMA DE POSICIONAMIENTO GLOBAL

El Sistema de Posicionamiento Global (GPS, *Global Positioning System*) es un sistema de navegación satelital diseñado por el Departamento de Defensa de Estados Unidos con fines militares de proporcionar estimaciones precisas de posición, velocidad y tiempo. [29, 30]

2.5.1 INTRODUCCIÓN

El GPS surgió debido a la necesidad de las fuerzas armadas de tener un sistema de navegación preciso y que funcionará en aplicaciones diversas. El desarrollo de la tecnología GPS descansa en progresos en ciencias físicas, en la electrónica, en ciencias de materiales y en muchas otras, pero fue el desarrollo de dispositivos extremadamente precisos para medir el tiempo – relojes atómicos, junto con el progreso en la tecnología espacial, que en realidad hicieron posible el GPS. [31]

En el ámbito civil y alegando razones de seguridad, sólo se permite el uso de un subconjunto degradado de señales GPS. Sin embargo, la comunidad civil ha encontrado alternativas para obtener una excelente precisión en la localización mediante las denominadas técnicas diferenciales. Gracias a ellas, las aplicaciones civiles han experimentado un gran crecimiento y actualmente existen más de 70 fabricantes de receptores GPS. [32]

El GPS ha sido adoptado en aplicaciones de transportación terrestre, aviación civil, comercio marítimo, topografía, construcción, minería, agricultura, ciencias geológicas, energía eléctrica, telecomunicaciones, y actividades recreativas a la intemperie. [33]

2.5.2 ARQUITECTURA

La arquitectura del Sistema de Posicionamiento Global está conformada de tres partes [32], como se muestra en la figura 2.6:

- ◇ El sector espacial, que consta de 24 satélites con una órbita de 26560 km de radio y un periodo de 12 h.
- ◇ El segmento control, constituido por estaciones monitoras, encargadas de mantener en órbita los satélites y supervisar su correcto funcionamiento.
- ◇ El segmento usuario, formado por las antenas y los receptores pasivos situados en tierra.

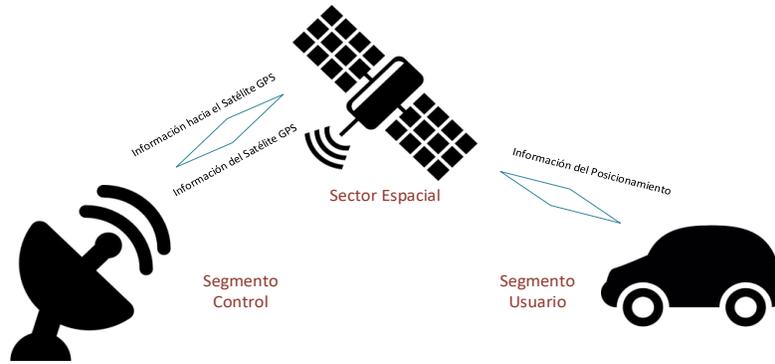


FIGURA 2.6 ARQUITECTURA DEL SISTEMA DE POSICIONAMIENTO GLOBAL.

Los satélites se mueven en el espacio a una velocidad de 4 km/s y transmiten anuncios de su posición y los receptores determinan las coordenadas del usuario por triangulación geométrica de estas posiciones. La exactitud de su posición depende de plataformas espaciales súper estables, relojes atómicos que sincronizan la constelación y el monitoreo de los satélites desde 5 puntos distribuidos alrededor del globo terráqueo. [33]

La precisión final alcanzable depende de las efemérides determinadas por el segmento control. La precisión nominal de estas efemérides, permite la determinación de la posición de un satélite con error menor de 1 metro en sentido radial, 7 metros a lo largo de la trayectoria y 3 metros transversalmente. [34]

El segmento usuario, como se mencionó anteriormente, comprende los instrumentos utilizados para hallar coordenadas de un punto, hacer navegación o adquirir tiempo con precisión de oscilador atómico, usando las señales radiodifundidas desde los satélites, además de proporcionar un pulso por segundo que está alineado con el tiempo universal coordinado. El equipo esencial está formado por una antena y receptor unidos directamente o mediante un cable. [34]

Es de importancia hacer mención que actualmente, el sector de la energía utiliza un estándar de tiempo, el cual se conoce como estándar IRIG-B, para los registradores de fallas con estampa de tiempo y para otros sistemas de supervisión de eventos en las subestaciones. [35]

A pesar de todo, con los receptores acorde al estándar IRIG-B, la precisión de sincronización es del orden de un milisegundo (1 ms), lo cual no es suficiente para obtener mediciones precisas del sistema de distribución. [35]

2.5.3 TRIANGULACIÓN

La idea principal detrás del GPS es utilizar los satélites en el espacio como puntos de referencia para ubicaciones en la Tierra. [34] Esto se logra utilizando la distancia del receptor GPS hacia tres satélites, lo que permite triangular dicha posición en cualquier parte de la Tierra.

El proceso de triangulación se realiza, de manera general, como sigue. El receptor determina la distancia al primer satélite, midiendo el tiempo que tarda una señal emitida por el satélite en llegar a dicho receptor GPS. Esta distancia limita la posición del receptor a la superficie de una esfera que tiene como centro dicho satélite y cuyo radio es de la distancia calculada. Después, se obtiene la distancia del receptor al segundo satélite, lo que indica que el receptor se encuentra sobre las dos esferas de las distancias calculadas. Es decir, el receptor se encuentra en algún punto de la circunferencia que resulta de la intersección de las dos esferas. [34]

Así, si se mide la distancia a un tercer satélite, se limita la posición del receptor GPS mucho más, a los dos puntos en los cuales la tercera esfera corta la circunferencia que resulta de la intersección de las dos primeras esferas. Al medir la distancia a tres satélites, se limita el posicionamiento a sólo dos puntos posibles [34], como se muestra en la figura 2.7.]

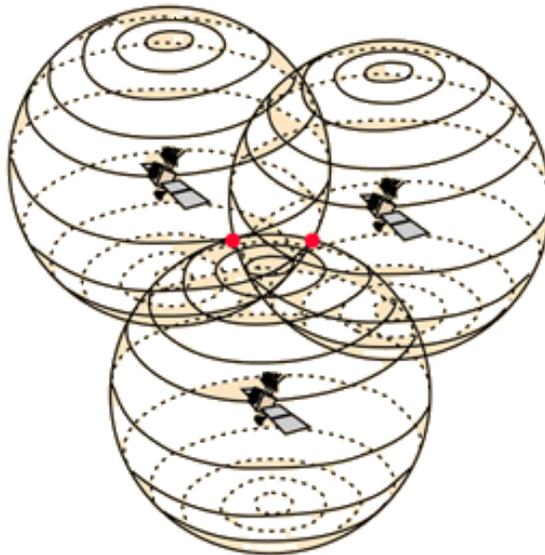


FIGURA 2.7 TRIANGULACIÓN SATELITAL.

La decisión de cuál de estos puntos es la posición verdadera, se hace la medición a un cuarto satélite. Normalmente uno de los dos puntos posibles resulta ser demasiado improbable por tener una ubicación lejana de la superficie terrestre, descartándolo sin necesidad de mediciones posteriores. [34]

2.5.4 TIEMPO UNIVERSAL COORDINADO (UTC)

El Tiempo Universal Coordinado (UTC, *Universal Time Coordinated*), también conocido como tiempo civil es la zona horaria de referencia respecto a la cual se calculan todas las otras zonas del mundo. [36]

El UTC no es realmente una abreviatura, es una variante de tiempo universal (UT, *Universal Time*) y su modificador C (coordinado) añadido para expresar que es una variante más de UT. Es el sucesor del tiempo promedio del observatorio de Greenwich (GMT, *Greenwich Mean Time*). La nueva denominación fue creada para eliminar la inclusión de una localización específica de un estándar internacional, así como para basar la medida del tiempo en los estándares atómicos, más que en los celestes.

El UTC, definido por relojes atómicos, se sincroniza con el día y la noche, al que se le añaden o quitan segundos tanto a finales de junio como de diciembre, cuando resulta necesario. Los segundos, que se agregan o quitan, se determinan por el Servicio Internacional de Rotación de la Tierra, en base a sus medidas de la rotación de la Tierra. [38,40]

El UTC es el sistema de tiempo utilizado por muchos estándares de Internet, en particular, se ha diseñado el Protocolo de Tiempo de la Red como una forma de distribuir el tiempo UTC en Internet.

CAPÍTULO 3 HARDWARE

3.1 INTRODUCCIÓN

En el presente capítulo se presenta el hardware que se utilizó para la implementación de la Unidad de Medición Fasorial sincronizada vía satélite. Los principales elementos utilizados son enlistados a continuación:

- ◇ Tarjeta de desarrollo TWR KM34Z50M de NXP.
- ◇ Receptor GPS marca Parallax.

3.2 DIAGRAMA HARDWARE/SOFTWARE

Este diagrama hace la descripción gráfica de la operación de la PMU para tener un panorama amplio de lo que se está implementando. Así también, el diagrama muestra la interacción que tienen los elementos del hardware con las rutinas que se ejecutan por el software.

El diagrama hardware/software de la Unidad de Medición Fasorial que se desarrollará se encuentra mostrado en la figura 3.1. En rectángulos redondeados se indica el software a implementar, mientras que en rectángulos se muestra el Hardware utilizado.

La interacción general es de la siguiente manera: las señales (trifásicas) de corriente y voltaje son reducidas a niveles adecuados para ser procesados por los Convertidores Analógico-Digital del microcontrolador MK34Z128CLL5 de la tarjeta de desarrollo. Se realiza la programación del ADC para que tome 100 muestras de cada ciclo de 60 Hz, almacenando datos de dos ciclos. Una vez que los ADCs reciben las señales, guardan los resultados de las conversiones en 6 buffers, uno para cada señal, teniendo 3 señales de corriente y 3 señales de voltaje.

De manera paralela, el receptor GPS genera un pps sincronizado con el sector espacial. La señal de este pulso, es recibida por el microcontrolador mediante un pin que genera una interrupción externa, autorizando que los datos recibidos del ADC sean utilizados para el cálculo de la DFT.

Diseño de una Unidad de Medición Fasorial con tecnología ARM

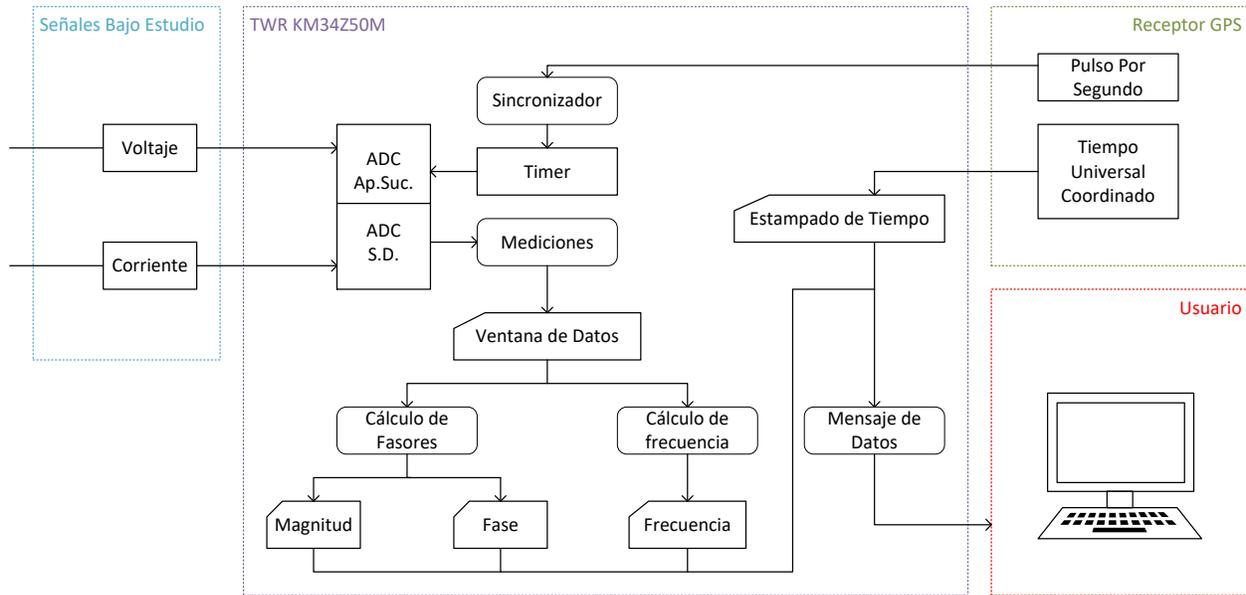


FIGURA 3.1 DIAGRAMA HARDWARE/SOFTWARE DE LA PMU.

Una vez entregado el pps, veintidós milésimas de segundo después el puerto serial del GPS envía datos relacionados con la ubicación del receptor GPS y el Tiempo Universal Coordinado del pps. Dichos datos son recibidos por el puerto serial del microcontrolador, para ser almacenados en un arreglo que permita obtener el valor del UTC. [14]

Los resultados de la Transformada Discreta de Fourier y el valor del UTC son enviados a la interfaz correspondiente para que el usuario sea capaz de observarlos.

La figura 3.2, mostrada a continuación, es la conexión física de la PMU realizada para esta tesis. Se observa que la alimentación del módulo PMU está conectada los bornes de la tarjeta que suministran 5V, además se tiene la conexión del pulso por segundo a la terminal PTD7 de la tarjeta y el Tx del módulo GPS al borne UART_Rx de la tarjeta para el envío y recepción del UTC.

Se tienen dos conexiones USB, una para el generador de señales y la otra que sirve como alimentación de la tarjeta y conexión serial con la PC.

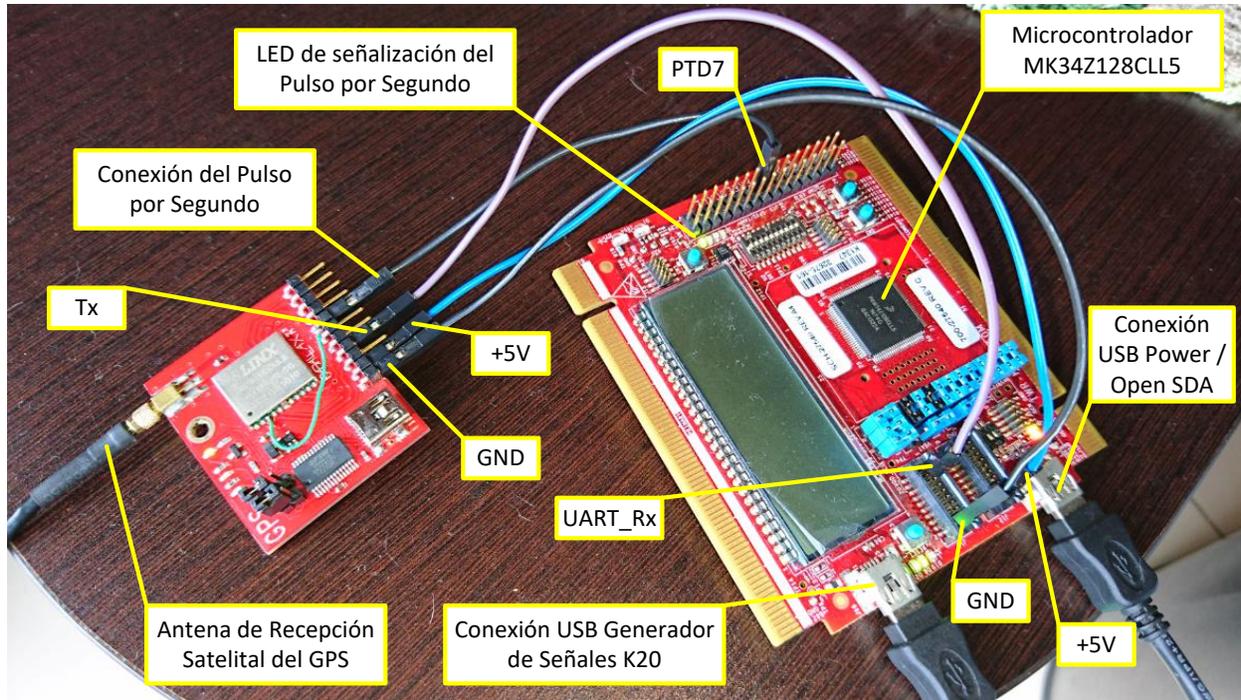


FIGURA 3.2 CONEXIÓN FÍSICA DE LA PMU.

Ahora bien, los elementos de Hardware utilizados para el desarrollo de este trabajo, son descritos en las siguientes secciones.

3.3 CARACTERÍSTICAS DE LA TARJETA TWR KM34Z50M

NXP cuenta con una amplia gama de tarjetas de desarrollo y microcontroladores, entre los cuales se tiene la tarjeta TWR KM34Z50M, cuyas características se describen a continuación.

La tarjeta TWR KM34Z50M ha sido diseñada por NXP para el desarrollo de prototipos de toda clase de dispositivos, especialmente aquellos que con Sistema Freescale de Torre, una plataforma modular de desarrollo que habilita el rápido diseño de prototipos y herramientas mediante hardware reconfigurable. [37]

En la figura 3.3 se muestra de manera gráfica la localización de los diferentes componentes que se encuentran en la tarjeta. La figura 3.3.a) corresponde a la vista frontal de la tarjeta, mientras que la figura 3.3.b) es la parte posterior de la tarjeta.

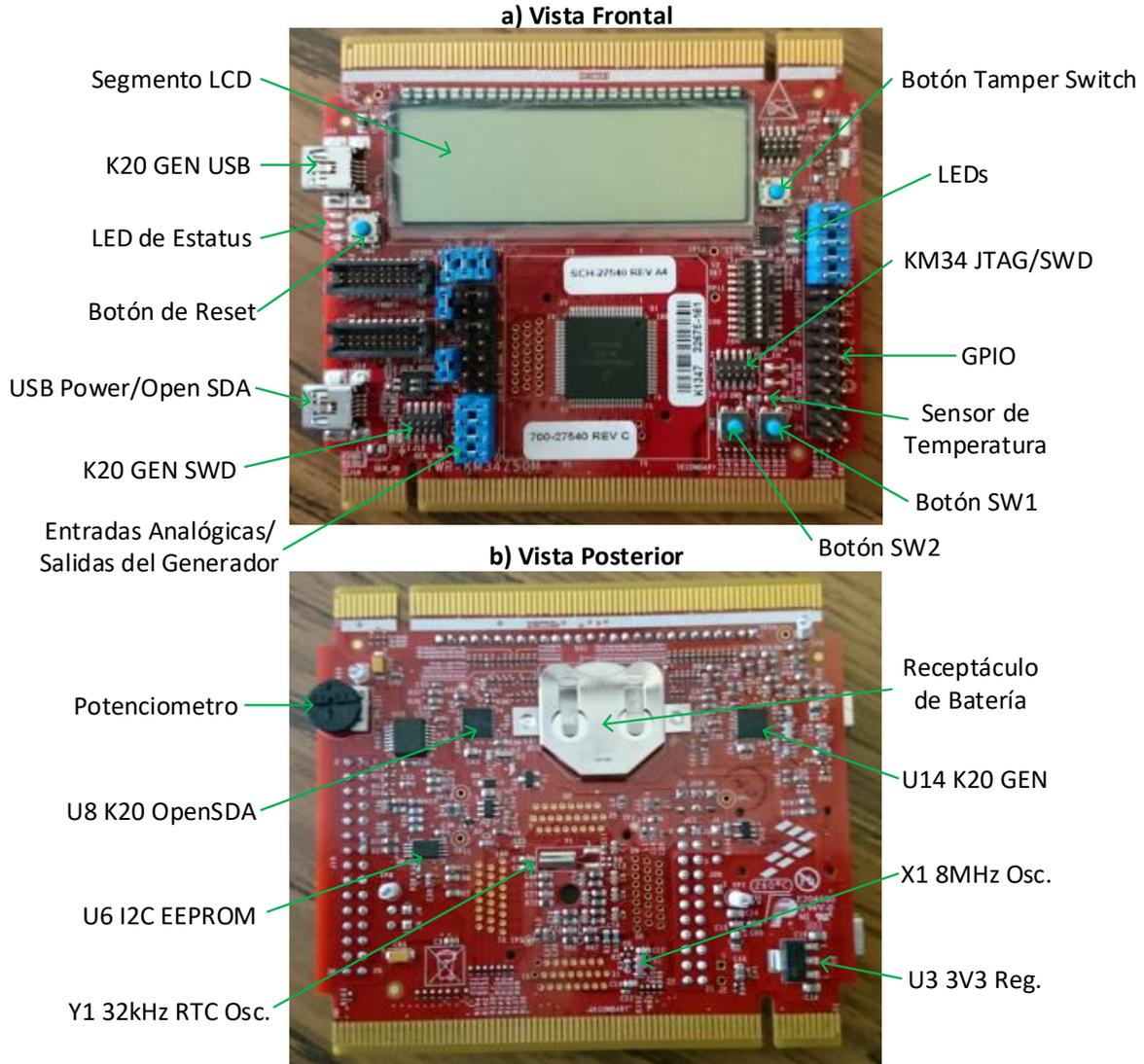


FIGURA 3.3 VISTA FRONTAL Y POSTERIOR DE LA TWR KM34Z50M.

Las características de esta tarjeta se enlistan a continuación:

- ◇ Módulo de Microcontrolador con compatibilidad de sistema de torre.
- ◇ Microcontrolador MK34Z128CLL5 (50 MHz, 128 KB Flash, 16 KB RAM, baja energía).
- ◇ Interface USB con conector Mini-AB USB.

- ◇ Pantalla de Cristal Líquido (LCD, *Liquid Crystal Display*).
- ◇ Circuito de depuración: JTAG/SWD (Open SDA) con puerto serial virtual.
- ◇ Cuatro Diodos Emisores de Luz (LED, *Light Emitting Diode*).
- ◇ Dos botones pulsadores para interrupciones de Entradas/Salidas de Propósito General (GPIO, General Purpose Input/Output).
- ◇ Un botón pulsador para el reinicio del microcontrolador.
- ◇ Potenciometro.
- ◇ Cabezales para acceso directo a los puertos GPIO y ADC.
- ◇ Fuente independiente de energía a base de una batería para el suministro del reloj de tiempo real (RTC, *Real Time Clock*).
- ◇ Generador de 7 canales de señales sinusoidales con interface USB para la simulación de formas de onda puras de corriente alterna.
- ◇ Sensor de temperatura. [37]

La siguiente imagen (figura 3.4) representa de manera general, la interacción de los elementos de la tarjeta, mediante un diagrama a bloques. [37]

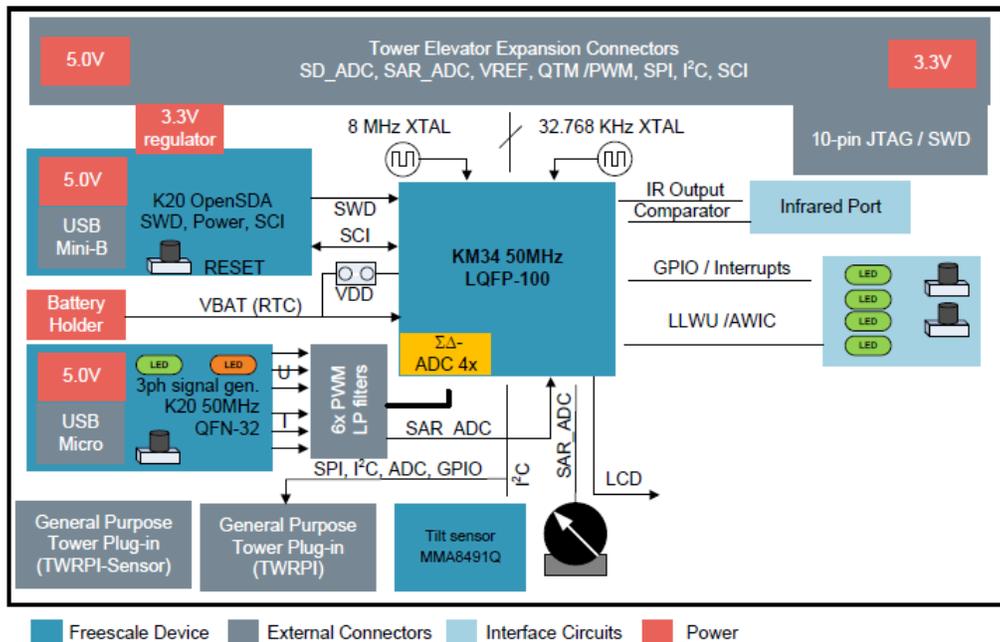


FIGURA 3.4 DIAGRAMA A BLOQUES DE LA TARJETA TWR KM34Z50M.

Como se observa, la tarjeta cuenta con un generador de señales trifásicas, el cuál proporciona señales sinusoidales que se pueden modificar tanto en magnitud como en fase, así como la introducción de armónicos a las mismas señales, como se describe en la sección 3.4.

3.3.1 MICROCONTROLADOR MK34Z128CLL5

El microcontrolador de esta tarjeta de desarrollo tiene diversas características, entre las cuales se encuentran su voltaje de operación que se sitúa entre 2.7 y 3.6 V, núcleo de alto rendimiento ARM Cortex M0+, frecuencia de reloj del núcleo hasta 50 MHz, 32 vectores de interrupción, cuatro niveles de prioridad programables, voltaje de referencia de 1.2 V, ADC Sigma-Delta de 24 bits y ADC de Aproximaciones Sucesivas de 16 bits. [37]

Además, cuenta con un Generador de Números Aleatorios (RNGA, *Random Number Generator Algorithm*), un enlace (crossbar) de periféricos que permite el re-mapeo interno de señales para flexibilidad, diferentes tipos de temporizadores, un módulo Receptor/Transmisor Universal Asíncrono (UART, *Universal Asynchronous Receiver/Transmitter*) y diversos puertos de Entrada y Salida. Aunado a estos elementos, cabe mencionar que este microcontrolador cuenta con una memoria Flash de 128 KB / 64 KB de arreglo simple y una memoria RAM de 16 KB de acceso simple, teniendo un consumo de corriente de 6.0 μ A en Modo RUN (todos los módulos apagados). [37]

Los diagramas de bloques y diagramas de localización de puertos y componentes, se encuentran en el Apéndice B.

3.3.2 CONVERTIDOR ANALÓGICO DIGITAL

Un convertidor analógico digital, o ADC; toma una señal analógica desconocida de entrada, más comúnmente un voltaje V_x y lo convierte en un número binario D de N bits que representa la proporción de V_x contra el valor a plena escala del convertidor, V_{FS} . [20] La tarjeta de desarrollo utilizada, cuenta con dos tipos de ADC. El primer tipo son los ADC de Aproximaciones Sucesivas y el segundo tipo son los ADC Sigma-Delta.

Los ADC de aproximaciones sucesivas, como se indica en su nombre, realizan comparaciones sucesivas de manera descendente o ascendente, hasta que se encuentra la combinación cuyo valor representa la magnitud del valor de entrada. En otras palabras, el contador dentro del registro no es

un contador secuencias de uno en uno, sino un contador programable que se incrementa o decrementa de acuerdo a la influencia del bit de mayor peso (SAR, *Successive Approximations Register*). El diagrama a bloques de este tipo de convertidor se muestra en la figura 3.5. [38]

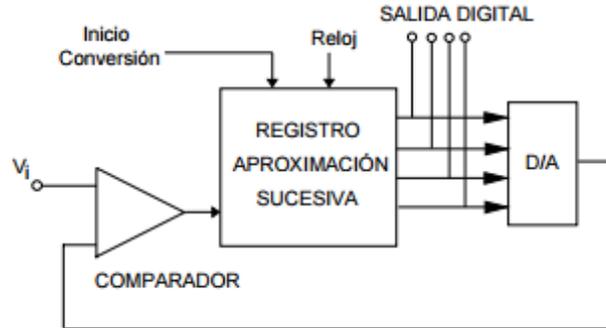


FIGURA 3.5 DIAGRAMA A BLOQUES DEL CONVERTIDOR DE APROXIMACIONES SUCESIVAS.

El SAR pone el bit más significativo (MSB, *Most Significant Bit*) en “1” y todos los restantes en “0”. La cantidad es tomada por el Convertidor Digital-Analógico (DAC, Digital-to-Analog Converter) de tal manera que su equivalente analógico se compara con la señal de entrada. Si la salida del DAC es mayor que la entrada, se elimina el “1” del bit MSB y se pone a “1” el bit inmediatamente anterior, con todos los demás bit en “0”, y así sucesivamente hasta que se logre encontrar una secuencia análoga pero que resulta ser menor que la entrada de la señal, cuando ocurra esto, el bit mantendrá su valor y se pone a “1” el bit inmediatamente anterior. El procedimiento anterior se repite hasta terminarse de probar “1” en cada bit del contador. Lo anterior equivale a un tanteo digital, a medida que se avanza, el procedimiento se va estabilizando hasta llegar un valor estable y que corresponderá con el valor de la medición.

Ahora bien, en los ADC Sigma-Delta, como se muestra en la figura 3.6, la entrada analógica de voltaje conduce a un integrador, cuya salida es comparada con un voltaje de nivel de tierra por el comparador. El cerrojo D controla el interruptor encendiendo/apagando el voltaje de referencia, estos voltajes componen un DAC de 1 bit. Conforme el voltaje de salida incrementa o disminuye, el comparador habilita o inhabilita el voltaje de referencia, que es sustraído de la señal de entrada, apuntando a mantener un valor de cero en la salida del integrador. [39]

Diseño de una Unidad de Medición Fasorial con tecnología ARM

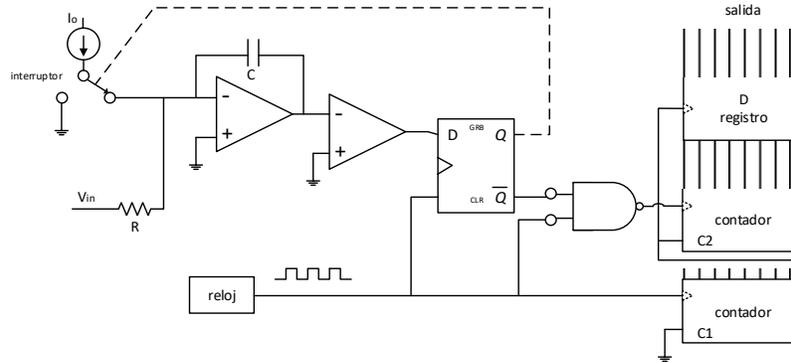


FIGURA 3.6 DIAGRAMA DEL CONVERTIDOR SIGMA-DELTA.

El contador C1 mantiene el rastro de los periodos de reloj, mientras que el contador C2 cuenta el número de pulsos cuando el interruptor está cerrado. Suponiendo que la capacidad del contador C1 es 1000, para el tiempo en que llega al final del conteo, el número en el contador C2 es proporcional al nivel medio de la señal de salida durante el tiempo de 1000 pulsos de reloj.

Ahora, el nombre Delta-Sigma tiene mayor sentido: Delta (la diferencial) se refiere a la modulación delta, el principio de codificación no de todo el valor de entrada, sino solo la diferencia entre la señal de muestreo y la señal de retroalimentación. Obviamente, menor número de bits son requeridos para codificar solo la diferencia en amplitudes. Sigma (la suma) es debido a que la suma de las deltas (diferencias) es contada durante el intervalo de medición. En otras palabras, la entrada al cuantificador es la integral de las diferencias entre la señal de entrada y la de salida. [40]

Las características de los ADC de la TWR KM34Z50M se muestran a continuación.

El Convertidor Analógico-Digital de Aproximaciones Sucesivas cuenta con las siguientes características:

- ◇ Algoritmo lineal de aproximaciones sucesivas con resolución de 16 bits.
- ◇ Hasta 24 entradas analógicas externas con terminación sencilla.
- ◇ Modos de salida: terminación sencilla de 16, 12, 10 y 8 bits.
- ◇ Salida en formato no signado con justificación a la derecha.
- ◇ Conversión singular o continua.
- ◇ Banderas e interrupciones por conversión completa.
- ◇ Sensor de temperatura.

- ◇ Selección de voltaje de referencia.
- ◇ Modo de Autocalibración. [41]

La descripción completa de las señales del ADC de Aproximaciones Sucesivas y su mapa de memoria se encuentran en [41]. El diagrama de bloques de este componente se muestra en la figura 3.7.

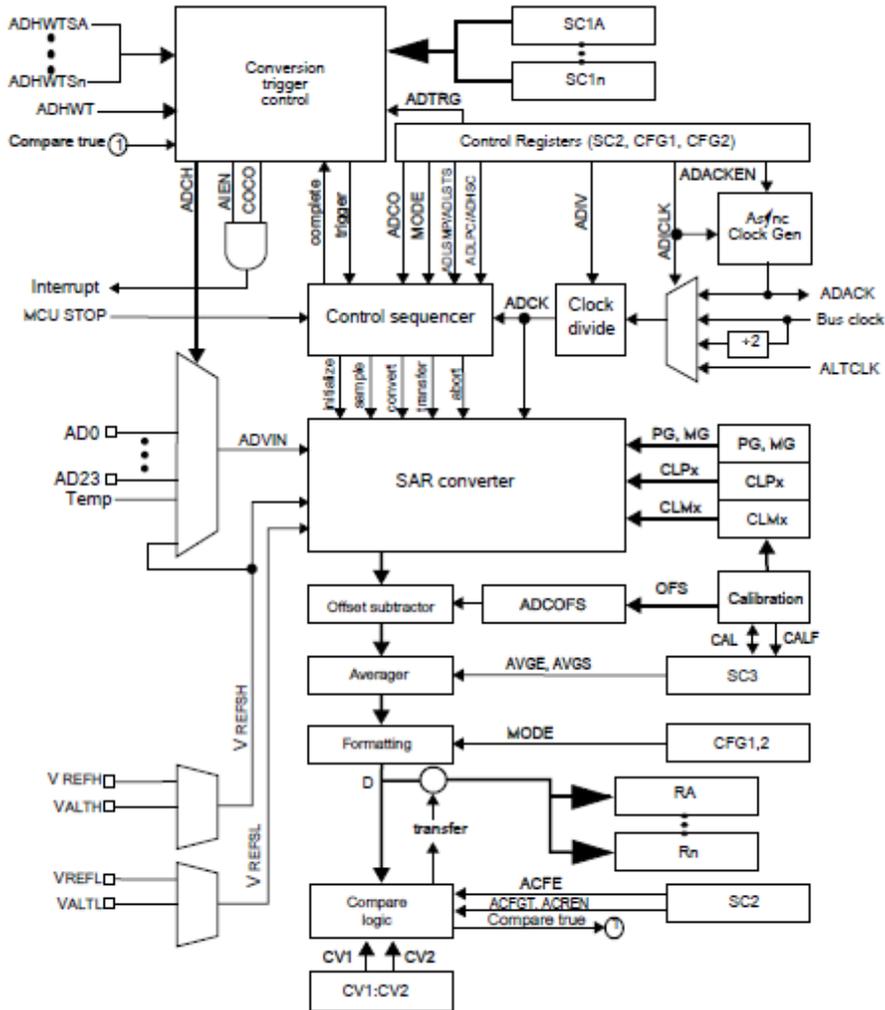


FIGURA 3.7 DIAGRAMA A BLOQUES DEL ADC DE APROXIMACIONES SUCESIVAS.

El ADC Sigma-Delta presente en la tarjeta de desarrollo, viene descrito como la Terminal Frontal Analógica (AFE, Analog Frontal End). Las características principales de este dispositivo son:

- ◇ Dos ADC Sigma-Delta de 24 bits con Amplificador de Ganancia Programable (PGA, Programmable Gain Amplifier).

- ◇ Dos ADC Sigma-Delta de 24 bits sin PGA.
- ◇ Opción de evitar/deshabilitar el PGA.
- ◇ Lógica digital habilita la inicialización sincronizada de todos los bloques AFE.
- ◇ Interrupción y solicitud de DMA en Conversión Completa.
- ◇ Disparos de Software y Hardware pueden ser usados para la inicialización de la conversión de los datos.
- ◇ Permite cambios dinámicos de variación de fase sin interrumpir los ciclos de muestreo. [41]

La descripción de las señales del ADC Sigma-Delta y su mapa de memoria se encuentran en [41]. El diagrama de bloques de este componente se muestra en la figura 3.8.

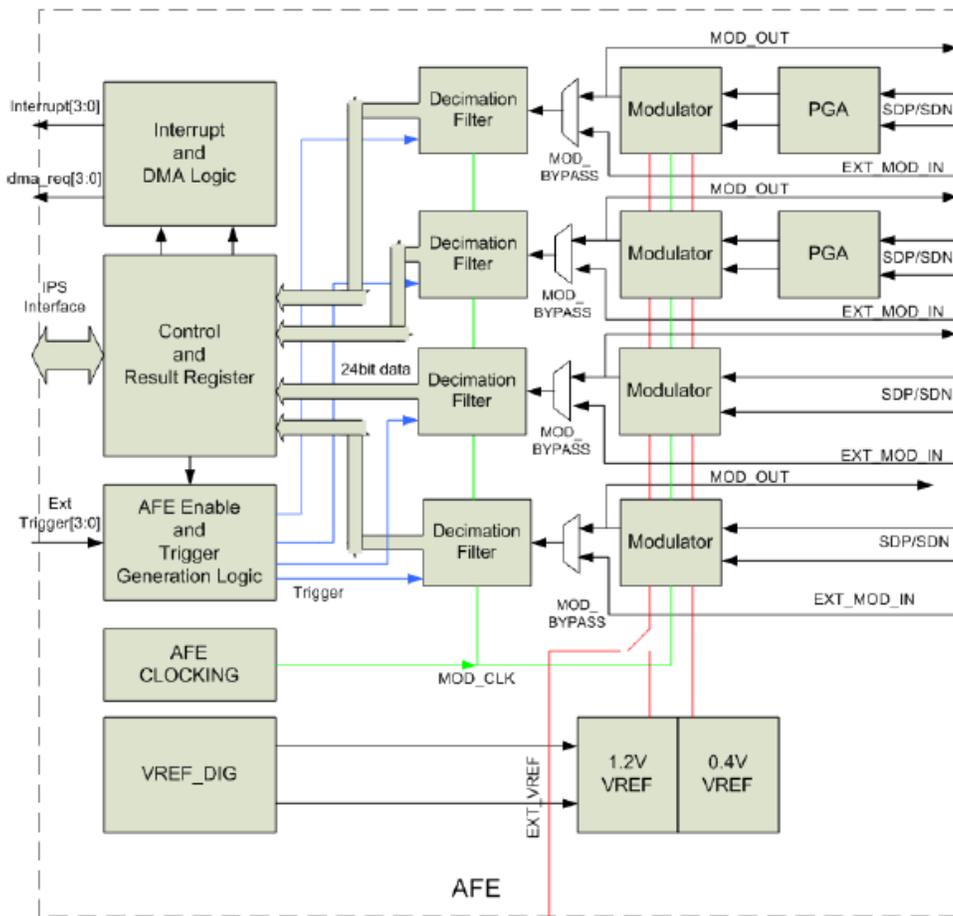


FIGURA 3.8 DIAGRAMA A BLOQUES DE LA TERMINAL FRONTAL ANALÓGICA (AFE).

3.3.3 BORNES DE ENTRADA/SALIDA DE PROPÓSITO GENERAL (GPIO)

Los bornes de entrada/salida de propósito general (GPIO) permiten comunicarse con hardware externo, mediante la lectura y escritura a los bornes, así como realizar interrupciones externas, [16] tal como recibir el pulso por segundo proveniente del receptor GPS.

El módulo de entradas/salidas de propósito general (GPIO) es accesible mediante el bus de periféricos. Sus registros soportan accesos de 8, 16 y 32 bits. [41]

Las características del módulo GPIO incluyen:

- ◇ Registros visibles del Puerto Entradas de Datos en todos los modos de multiplexeo.
- ◇ Registro de Dirección de Datos del Puerto.

La tabla 3.1, mostrada a continuación, representa los diferentes modos de operación y el comportamiento del módulo GPIO respectivo.

TABLA 3.1 MODOS DE OPERACIÓN DEL MÓDULO GPIO.

Modo de Operación	Descripción
Run (Ejecución)	El módulo GPIO opera normalmente.
Stop (Detenido)	El módulo GPIO está deshabilitado.
Debug (Depuración)	El módulo GPIO opera normalmente.

La descripción completa de las señales, registros y mapas de memoria de las entradas/salidas de propósito general, se encuentran en [41].

3.3.4 INTERRUPTOR INTER-PERIFÉRICOS (XBAR)

Este módulo implementa un arreglo de “M” multiplexores combinacionales de “N” entradas. Todos los multiplexores comparten las mismas “N” entradas en el mismo orden, pero cada multiplexor tiene su propio campo de selección independiente.

La aplicación pretendida de este módulo es el de proporcionar una función de conmutador de barras (crossbar) flexibles que permite que cualquier entrada (típicamente de GPIO externo o salida de los módulos internos) para ser conectado a cualquier salida (GPIO típicamente externo o entradas de los módulos internos) bajo el control del usuario. Esto se utiliza para permitir la configuración de usuario de rutas de datos entre los módulos internos y entre los módulos internos y GPIO.

Un subconjunto de los multiplexores puede ser configurado para soportar la detección de bordes y ya sea interrumpir o generar la solicitud del DMA basado en la detección de flancos de señal en la salida de multiplexor. Esto permite transiciones de señal en las señales que alimentan el ‘crossbar’ para disparar interrupciones o inicializar la transferencia de datos a través del DMA hacia o desde otros módulos del sistema. [41]

Sus características principales son:

- ◇ “M” idénticos multiplexores de “N” entradas con campos de selección independientes.
- ◇ Selección de flanco con interrupción asociada o generación de la solicitud del DMA por un subgrupo de salidas de multiplexor.
- ◇ Registros de memoria mapeados con interfaz IPBus para campos de selección y control.
- ◇ Señal de entrada de protección de escritura de registro. [41]

El diagrama a bloques de este Módulo se muestra en la figura 3.9.

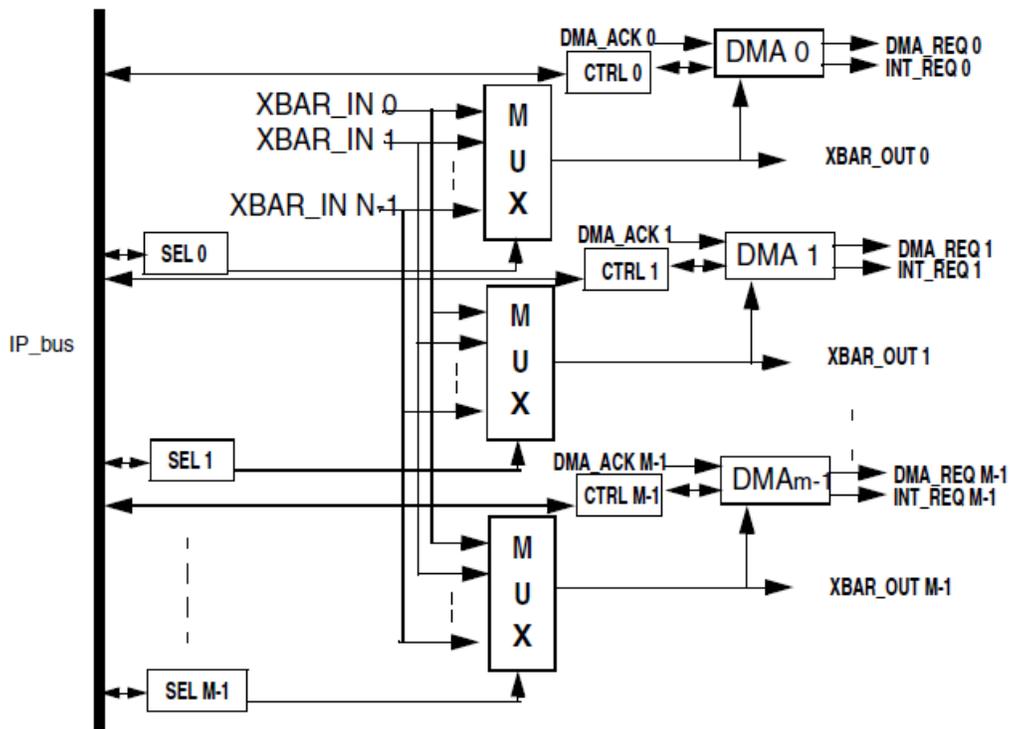


FIGURA 3.9 DIAGRAMA A BLOQUES DEL MÓDULO CROSSBAR (XBAR).

3.3.5 RECEPTOR/TRANSMISOR UNIVERSAL ASÍNCRONO (UART)

El Receptor/Transmisor Universal Asíncrono (UART) permite la comunicación serial asíncrona con los dispositivos periféricos y los CPUs.

Sus principales características se describen a continuación:

- ◇ Operación dúplex completa.
- ◇ Formato estándar ‘sin retorno a 0’ (NRZ, Non-Return-to-Zero).
- ◇ Selección de velocidad de transmisión de 13 bits, con divisor fraccional /32, basado en la frecuencia del módulo del reloj.
- ◇ Formato programable de datos de 8 o 9 bits.
- ◇ Transmisor y receptor habilitados por separado.
- ◇ Estructura FIFO (First In, First Out) independiente para transmitir y recibir.

Los modos de operación, descripción de señales, registros, mapas de memoria y descripción completa de funcionamiento de este Módulo se encuentran en [41].

3.4 GENERADOR DE SEÑALES

El generador de señales K20 sobre la tarjeta, es capaz de generar siete ondas con forma sinusoidal emulando una red eléctrica real (corriente trifásica y voltaje trifásico) con parámetros configurables (frecuencia, amplitud, ángulo de fase y distorsión armónica). La máxima amplitud de las formas de onda generadas es de 1.0 V (pico a pico), así que pueden ser medidas por el ADC trabajando en modo de terminal simple. El generador puede ser interconectado mediante el conector USB a la PC, mediante el uso de la aplicación FreeMASTER, donde los parámetros de la red pueden ser fácilmente sintonizados. Las señales analógicas externas pueden ser conectadas directamente a las entradas de los ADC mediante el cabezal J17. [37] Sus principales características se muestran a continuación.

- ◇ Basado en la Subfamilia de microcontroladores K20 de Kinetis.
- ◇ Siete señales PWM, capaz de generar señales senoidales ($3xI + 3xU$, es decir, corriente trifásica + voltaje trifásico).

- ◇ Amplitud, frecuencia y ángulos de fase configurables.
- ◇ Conexiones tipo jumper hacia las entradas de los ADC.
- ◇ Puerto serial virtual (USB-CDC) para PC, corriendo la aplicación FreeMASTER. [37]

La figura 3.10 muestra el diagrama general de conexión para el concepto de comunicación entre el generador de señales K20, la PC y las entradas de los ADC del microcontrolador KM34 de la tarjeta de desarrollo. [37]

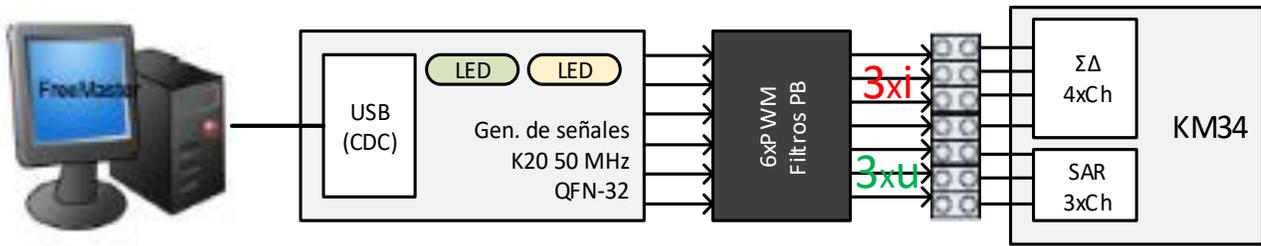


FIGURA 3.10 CONCEPTO DEL GENERADOR DE SEÑALES K20.

FreeMASTER es un monitor depurador en tiempo real y una herramienta de visualización de datos amigable con el usuario que se puede utilizar para el desarrollo de aplicaciones y manejo de información. Soporta el monitoreo no intrusivo de variables en un sistema en ejecución. Puede mostrar varias variables que cambian con el tiempo en una pantalla tipo osciloscopio o ver los datos en forma de texto. Cuenta con un panel de control en el que se puede ajustar de manera directa el valor de una variable y compatibilidad con aplicaciones externas como Excel, Matlab u otra aplicación que soporte ActiveX. Tanto las características específicas como los requerimientos de sistema operativo y dispositivos soportados, se encuentran en [51].

Las figuras 3.11 y 3.12, muestran las señales de salida del generador utilizando el programa FreeMASTER para visualizarlas. Estas señales son muestreadas por los diferentes canales de los ADC y que son utilizadas para la estimación de los fasores

Diseño de una Unidad de Medición Fasorial con tecnología ARM

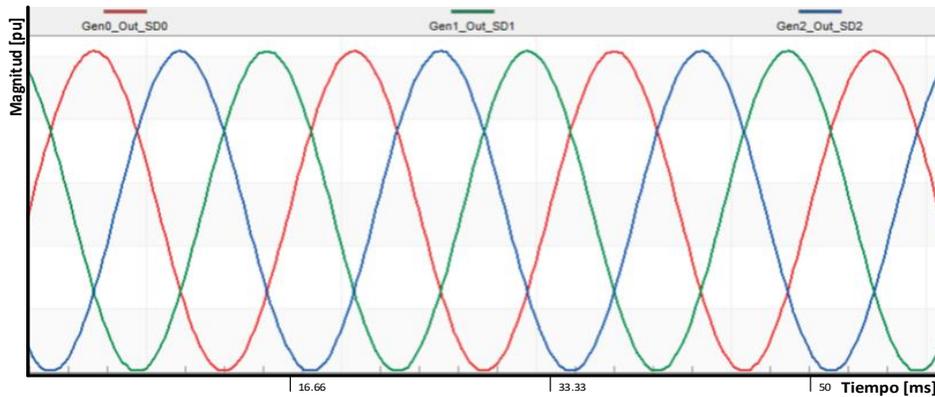


FIGURA 3.11 SALIDA DE LOS CANALES GEN0, GEN1 Y GEN2 DEL GENERADOR DE SEÑALES, FREEMASTER.

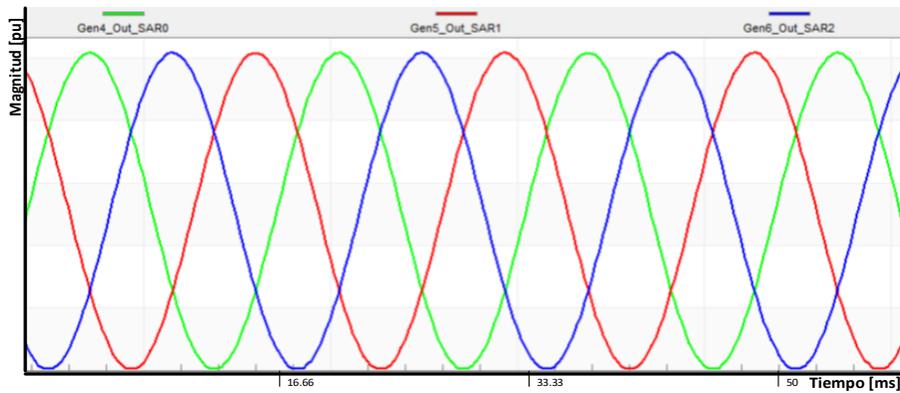


FIGURA 3.12 SALIDA DE LOS CANALES GEN4, GEN5 Y GEN6 DEL GENERADOR DE SEÑALES, FREEMASTER.

Ahora bien, utilizando un osciloscopio Tektronix MSO 3014, se observan las señales de salida del generador, comprobando que dichas señales son reales y sirven para emular una red trifásica para el análisis del funcionamiento de este trabajo. La figura 3.13 muestra el diagrama a bloques de la interacción entre el generador de señales y el osciloscopio.

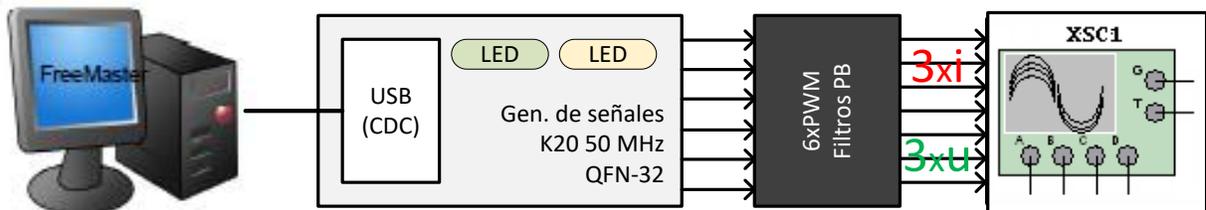


FIGURA 3.13 DIAGRAMA A BLOQUES DE LA INCORPORACIÓN DEL OSCILOSCOPIO.

Las figuras siguientes, figuras 3.14 y 3.15, muestran las curvas en la pantalla del osciloscopio. Como se puede observar, dichas curvas son equivalentes a las mostradas por las figuras 3.11 y 3.12.

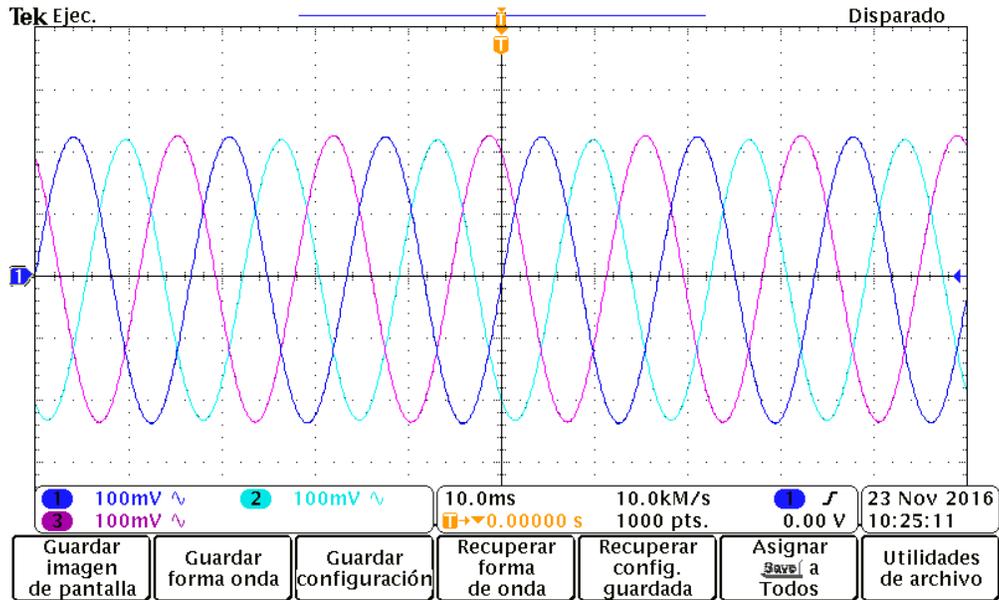


FIGURA 3.14 SALIDAS DE LOS CANALES GEN0, GEN1 Y GEN2 DEL GENERADOR DE SEÑALES, OSCILOSCOPIO.

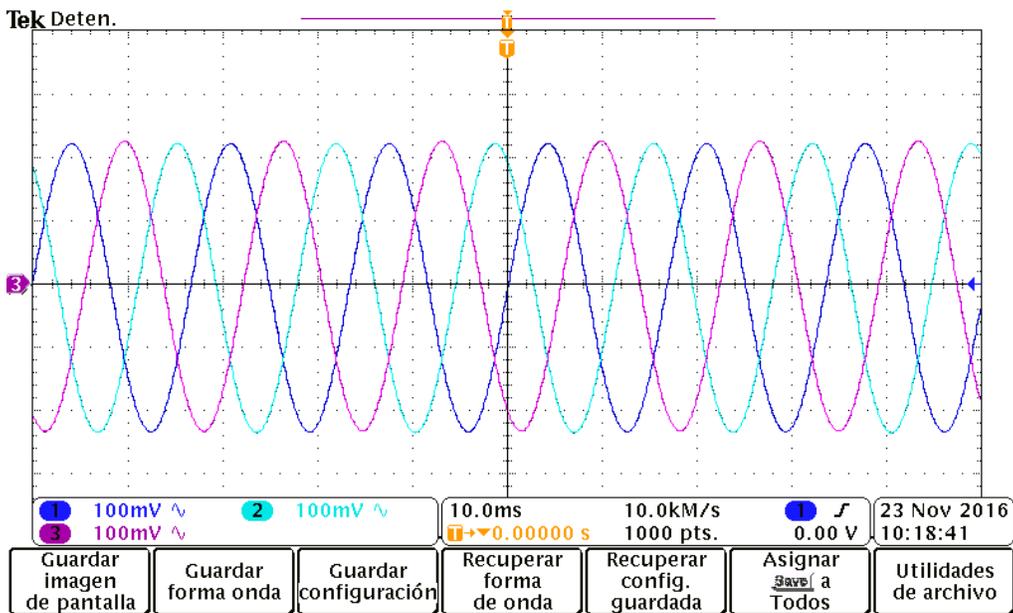


FIGURA 3.15 SALIDAS DE LOS CANALES GEN4, GEN5 Y GEN6 DEL GENERADOR DE SEÑALES, OSCILOSCOPIO.

Así, las señales de la figura 3.11, GEN0, GEN1 y GEN2 son utilizadas como señales de corriente para los ADC Sigma-Delta, mientras que las señales de la figura 3.12, GEN3, GEN4 Y GEN5, son utilizadas como señales de tensión para los ADC de Aproximaciones Sucesivas.

Ahora bien, introduciendo armónicos en las señales del generador, se tienen las figuras 3.16 y 3.17, e introduciendo defasamiento en las señales del mismo, se tienen las figuras 3.18 y 3.19, que permiten realizar la comparación entre la herramienta FreeMASTER y el osciloscopio.

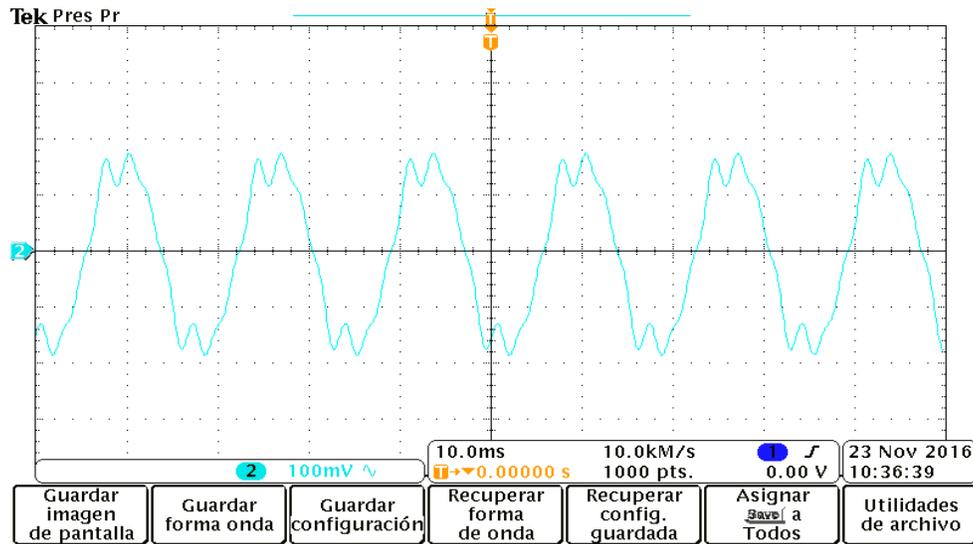


FIGURA 3.16 SEÑAL AFECTADA CON EL 3°, 5° Y 7° ARMÓNICO, OSCILOSCOPIO.

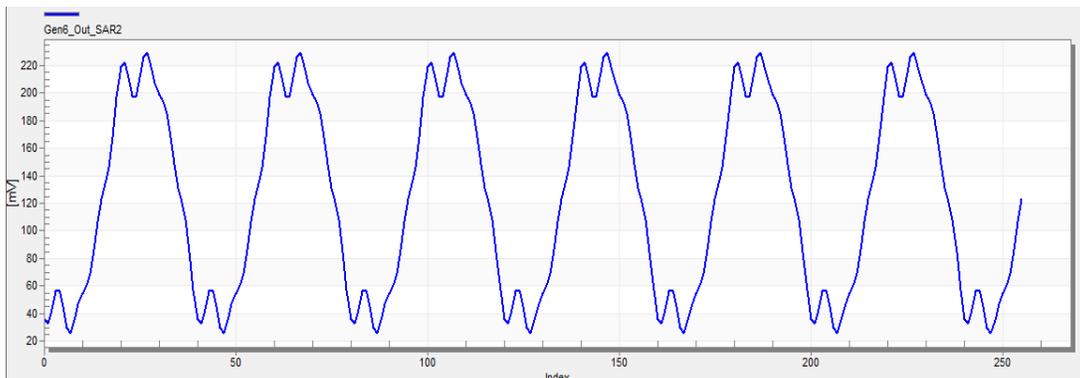


FIGURA 3.17 SEÑAL AFECTADA CON EL 3°, 5° Y 7° ARMÓNICO, FREEMASTER.

La figura 3.16 muestra la señal visualizada por el osciloscopio, y la figura 3.17 muestra la señal visualizada en FreeMASTER. Como se puede observar, la misma señal de entrada con armónicos es representada por curvas similares por ambos elementos de observación.

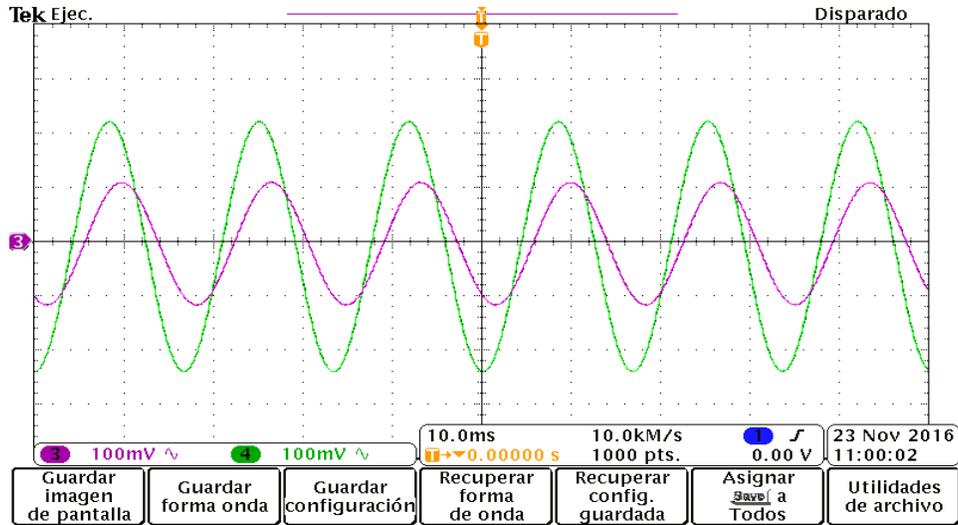


FIGURA 3.18 SEÑALES DESFASADAS 30°, OSCILOSCOPIO.

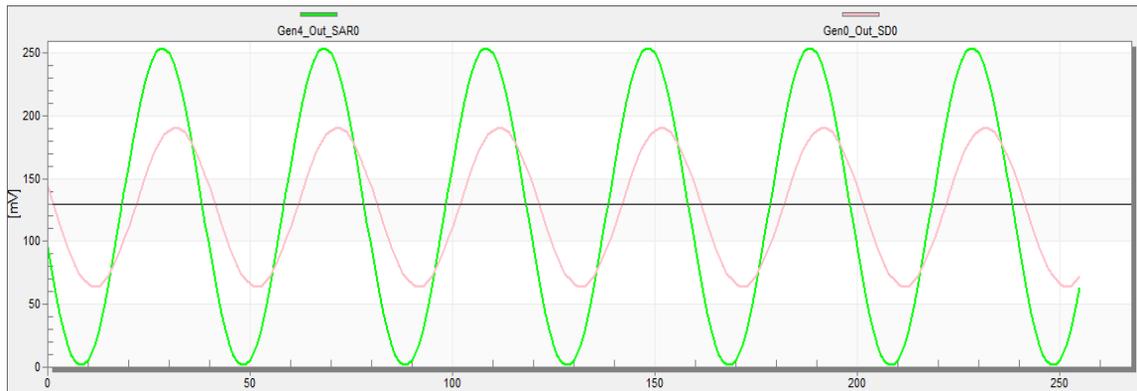


FIGURA 3.19 SEÑALES DESFASADAS 30°, FREEMASTER.

La figura 3.18 muestra las señales visualizadas por el osciloscopio, y la figura 3.19 muestra las señales visualizadas en FreeMASTER, con un defasamiento de 30° entre ellas y magnitudes diferentes. Como se puede observar, las mismas señales de entrada son representadas por curvas similares por ambos elementos de observación. Por lo tanto, al ver que las señales son representadas de manera sumamente similar, se considera que FreeMASTER brinda resultados aceptables en la visualización de las señales.

3.5 MÓDULO GPS

Como se ha mencionado, la sincronización de los fasores calculados por la PMU se realiza mediante la utilización de un GPS. [2] El receptor de la marca Parallax está integrado por las siguientes partes:

- ◇ Modulo Receptor.
- ◇ Antena GPS. [44]

3.5.1 MÓDULO RECEPTOR

Este módulo GPS puede conectarse a un microcontrolador con protocolo NMEA 0183, o incluso a una PC (mediante USB). Es de bajo consumo de energía, con una alta sensibilidad (-159 dBm), 20 canales de rastreo satelital y un cabezal de 11 pines con espaciado de 0.1". [44]

Entre sus especificaciones, se encuentran las siguientes:

- ◇ Requerimientos de energía: 5 V, 50 mA (típicamente).
- ◇ Comunicación: 3.3 V CMOS asíncrono serial a 9600 bauds, o mediante USB para PC. [44]

En la figura 3.20 se muestra el módulo GPS utilizado, indicando los nombres de los diferentes pines en el cabezal y sus conexiones.

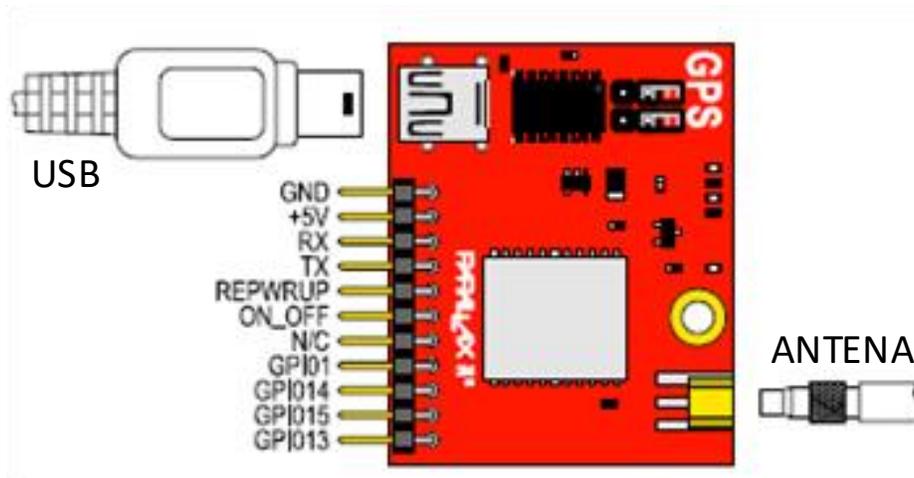


FIGURA 3.20 MÓDULO GPS.

La función de cada uno de los pines del módulo receptor está enlistada en la tabla 3.2. [44]

TABLA 3.2 DESCRIPCIÓN DE LOS BORNES DEL MÓDULO GPS.

PIN	NOMBRE	TIPO	FUNCIÓN
1	GND	G	Tierra (0V)
2	+5V	P	Energía (5V)
3	RX	I	Receptor del GPS*
4	TX	O	Transmisor del GPS
5	RFPWRUP	O	Indicador del Estado de Energía
6	ON_OFF	I	Solicitud de Disparo por Flanco On/Off*
7	N/C	--	Sin conexión
8	GPIO01	I/O	Entrada/Salida de Propósito General*
9	GPIO14	I/O	Entrada/Salida de Propósito General*
10	GPIO15	I/O	Entrada/Salida de Propósito General*
11	GPIO13	I/O	Entrada/Salida de Propósito General*

Tipo de Pin: P = Energía, G = Tierra, I = Entrada, O = Salida.

*Ver [44] para los rangos de los bornes y detalles.

3.5.2 ANTENA GPS

La antena recibe los datos de los satélites y los envía al módulo receptor. Debido a que las señales GPS se propagan a una frecuencia de 1575 MHz y no penetran a través de materiales opacos, la antena se debe colocar al aire libre y con visibilidad clara al cielo. La antena utilizada es mostrada en la figura 3.21. Es una antena de tipo magnética para ser colocada en una superficie metálica y cuenta con cable de 9'. [44]



FIGURA 3.21 ANTENA EXTERNA DEL MÓDULO GPS.

3.5.3 PULSO POR SEGUNDO (PPS)

La señal de 1 pps es una señal generada por el sistema GPS receptor. Por omisión, cada vez que se energiza el receptor, se genera el pps, aunque no se haya detectado satélite alguno; sin embargo, una vez que se detecta por lo menos un satélite, el receptor sincroniza el flanco de subida del pps con la recepción del UTC. [45] Debido a que en los pines del cabezal de este receptor no se cuenta con uno que tenga acceso al pps, se tiene que acceder a él de manera física, enviando mediante un conector ese pulso a una de las terminales del cabezal, en este caso en la terminal 7 que no tiene conexión alguna. Dicho pps se encuentra en la terminal tres del receptor GPS, como se muestra en la figura 3.22. [43]

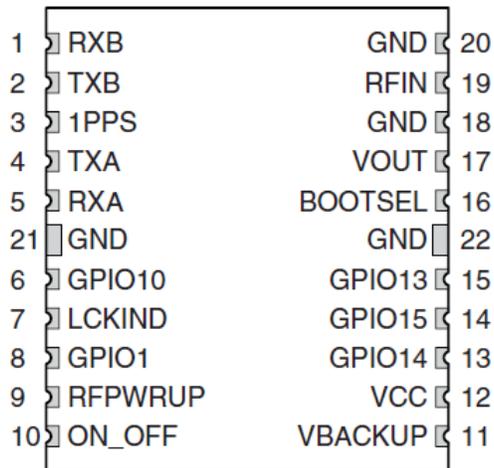


FIGURA 3.22 VISTA SUPERIOR DEL RECEPTOR GPS.

Este pulso por segundo es generado en flanco de subida con duración de $1\mu\text{S}$ y una exactitud cercana a $1\mu\text{S}$ del segundo del GPS. El segundo del GPS está basado en los relojes atómicos de los satélites GPS, que son monitoreados y fijados con los relojes maestros de Tiempo Universal. [43]

CAPÍTULO 4 DISEÑO DE SOFTWARE

4.1 INTRODUCCIÓN

En este capítulo se presenta el desarrollo del software implementado para la realización de este proyecto, que consta de una unidad de medición fasorial sincronizada vía satélite.

De manera detallada, se muestran las etapas necesarias para el desarrollo del PMU, entre las cuales se tienen los algoritmos de medición de las señales, la determinación de fasores de voltaje y corriente, obtención de la frecuencia, sincronización mediante el uso del GPS y la visualización de los datos.

4.2 AMBIENTE DE DESARROLLO DEL MICROCONTROLADOR

La tarjeta de desarrollo TWR KM34Z20M utiliza un ambiente de desarrollo integral, conocido como *CodeWarrior* de Freescale, en el que se puede utilizar tanto lenguaje ensamblador como C++.

El programa principal realiza el llamado a diferentes rutinas y ejecuciones de instrucciones, es decir, funciona como mediador de los procesos que se requieren ejecutar para el desarrollo del proyecto. Entonces, las rutinas, que tienen el mayor trabajo en la programación, se clasifican en interrupciones (o eventos) y funciones (o métodos).

Las interrupciones son aquellas rutinas que son ejecutadas por eventos de dispositivos periféricos o internos del microcontrolador (como el ADC, el AFE, el UART). Una vez que se realiza la interrupción por medio de alguno de éstos, se ejecutan las instrucciones pertinentes acorde a las necesidades del programa.

Se llama funciones a las rutinas que desarrollan una tarea específica (como inicializar dispositivos, cálculos de valores) y pueden ser llamadas desde el programa principal o desde una interrupción.

4.2.1 SOFTWARE DE LA TARIETA DE DESARROLLO TWR KM34Z50M

Como se mencionó anteriormente, el programa *CodeWarrior V10.6*, será la herramienta para desarrollar los códigos apropiados para el proyecto. La metodología de programación empleada consta de un programa principal y algunos ‘headers’ que contienen diferentes rutinas que alternan en el programa principal. Este último siempre se está ejecutando dentro de un lazo infinito el cuál es interrumpido cada vez que un evento y/o interrupción es invocado.

4.2.2 CREAR UN PROYECTO PARA LA TARIETA DE DESARROLLO TWR KM34Z50M

Como parte fundamental de la implementación del proyecto se tiene el conocimiento general del software de *CodeWarrior* donde se realizan los algoritmos; es por ello que se muestran los pasos básicos que se deben seguir para generar un nuevo proyecto.

En primera instancia, se debe contar con la carpeta del driver para la tarjeta de desarrollo TWR KM34Z50M, la cual lleva por nombre “KM128SWDRV_R4_1_5”. Ésta puede ser descargada de la página de NXP. Dentro de dicha carpeta, existe otra con el nombre de “template”; ahí se tiene una aplicación llamada “make_project”, como se muestra en la figura 4.1, la cual permite realizar el proyecto nuevo, como a continuación se describe.

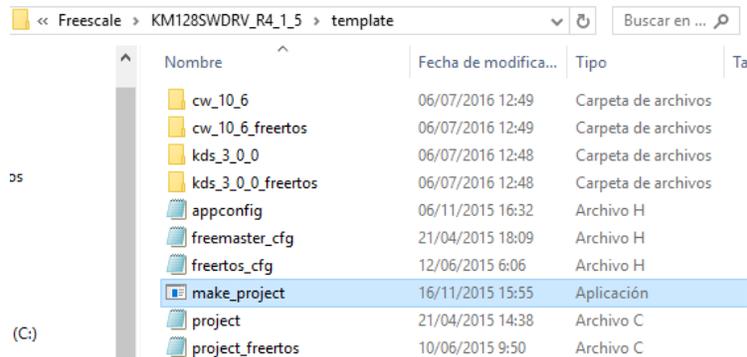


FIGURA 4.1 UBICACIÓN DE LA APLICACIÓN “MAKE_PROJECT”.

Una vez que se abra la aplicación, aparecerá una ventana como en la figura 4.2. En ese momento, se accesa a un menú que permite generar proyectos para diferentes programas, en este caso, con el teclado se introduce un “1” para realizar un proyecto en *CodeWarrior V10.6*, o el programa que se requiera utilizar.

Diseño de una Unidad de Medición Fasorial con tecnología ARM

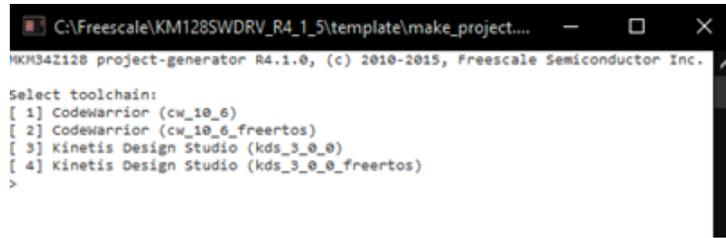


FIGURA 4.2 VENTANA PRINCIPAL DE LA APLICACIÓN “MAKE_PROJECT”.

Una vez asignado uno de los valores (1-4) para seleccionar el programa, se procede a seleccionar el tipo de microcontrolador que se esté empleando, en este case es de 128 kB de Flash, por lo que se introduce un “1”. Después, en opción del proyecto, se introduce un “1” para seleccionar la opción de “crear el proyecto en un subfolder del driver (predeterminado)”. Por último, se introduce el nombre con el cuál se debe llamar el proyecto y se oprime la tecla “Enter”. Así, la aplicación crea el proyecto en el subfolder correspondiente al microcontrolador seleccionado. Los pasos anteriores se muestran en la figura 4.3.



FIGURA 4.3 CREACIÓN DEL PROYECTO CON “MAKE_PROJECT”.

Ahora, se tiene que ingresar en el programa *CodeWarrior*, y en el Menú “File” se selecciona la opción “Import”, que permite, como su nombre lo indica, importar un proyecto al espacio de trabajo (Workspace) en el que se está trabajando. En la ventana que aparece, se selecciona la opción “Existing Projects into Workspace” para seleccionar un proyecto existente, se da click en “Next” y después en “Browse...” para buscar en los documentos el proyecto que se creó con la aplicación “make_project”. La ubicación del proyecto, en este caso, se encuentra en la carpeta con dirección: C:/Freescale/KM128SWDRV_R4_1_5/build/cw_10_6/ejemplo01. Seleccionamos dicha carpeta y “Aceptar”. Para finalizar la importación, se pulsa el botón “Finish” para tener el proyecto dentro del espacio de trabajo. Todo lo anterior se muestra en las figuras 4.4.a, b, c, d, e, f.

Diseño de una Unidad de Medición Fasorial con tecnología ARM

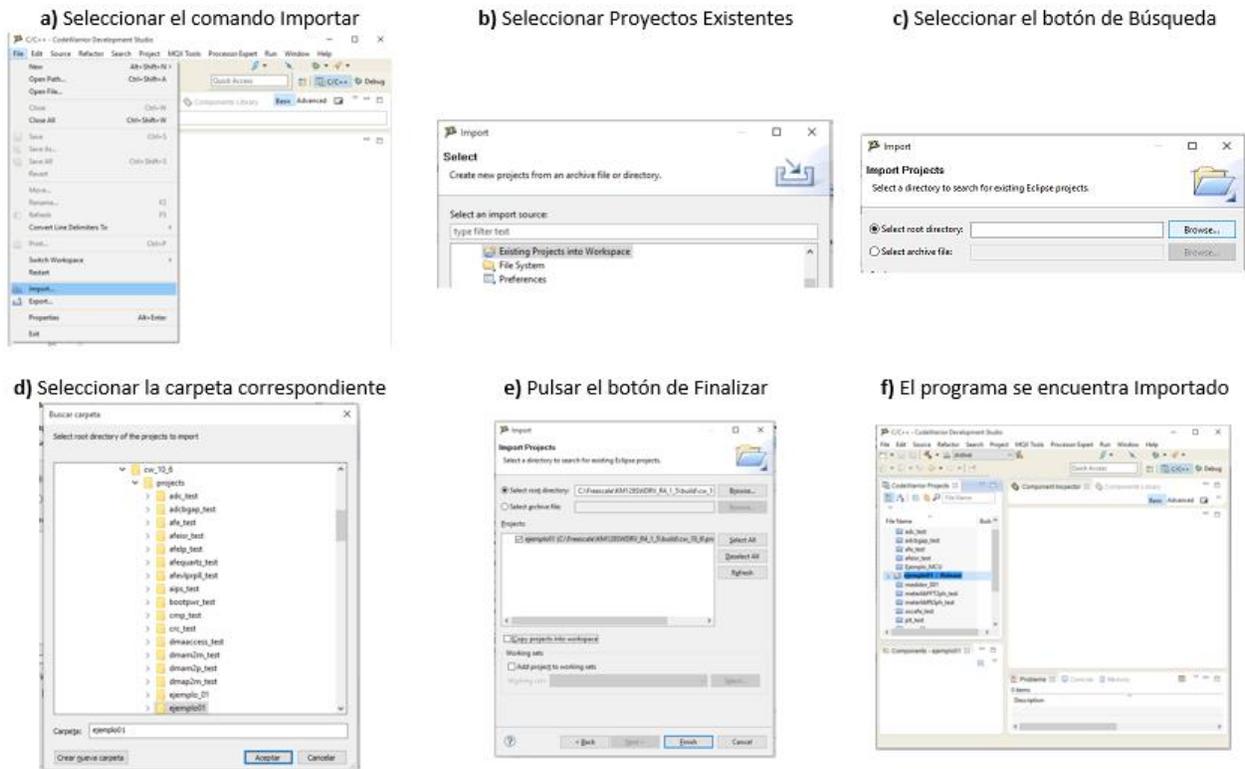


FIGURA 4.4 IMPORTACIÓN DEL PROYECTO

La tarjeta de desarrollo TWR KM34Z50M tiene un circuito multienlace que permite programar y depurar el programa a través del puerto USB desde la PC, además de energizar la tarjeta mediante ese mismo puerto. [37]

A continuación se describen los programas o funciones utilizadas para el desarrollo de este trabajo, explicando con diagramas la metodología empleada.

4.3 MEDICIÓN Y CÁLCULO

En esta sección se encuentran los diferentes bloques pertenecientes a los comandos de programación destinados a los cálculos necesarios para realizar la medición de los diferentes elementos del proyecto.

En la captura de las señales del sistema, hay que hacer la determinación de la frecuencia fundamental y del número de muestras por ciclo que se requieren para el cálculo de fasores. En este caso, la frecuencia de muestreo está determinada por los parámetros que se introduzcan en la configuración de los AFE, seleccionando adecuadamente los valores del reloj y el decimador se tendrá la frecuencia de muestreo del dispositivo, obteniendo así el número de muestras con las cuales trabajan los algoritmos.

Al tener una frecuencia de reloj del AFE de 12.288×10^6 Hz y seleccionando (DEC_OSR2048) como el decimador y un divisor propio del AFE de valor 1, se tiene una frecuencia de muestreo de 6000 Hz que, para una señal fundamental de 60 Hz, se tienen 100 muestras por ciclo.

Entonces, una vez obtenida la ventana de muestras de las señales se procesan los datos y se calcula el fador resultante mediante el algoritmo de la DFT.

Así, el algoritmo para el cálculo de la DFT de una señal digital requiere como precondition una ventana de muestras tomadas de una señal eléctrica y una tabla de valores del seno y coseno con el mismo número de muestras, y en base a la resolución de los ADC.

Entonces, de manera general, se procede como sigue:

- 1 Inicialización de Valores.
- 2 Mientras el Contador de Muestras sea menor al Total de Muestras, se repite lo siguiente:
 - 2.1 La Suma de Cosenos es igual a la Suma de Cosenos más la multiplicación de la Muestra “i” por el Coseno de Referencia “i”.
 - 2.2 La Suma de Senos es igual a la Suma de Senos más la multiplicación de la Muestra “i” por el Seno de Referencia “i”.
 - 2.3 Se incrementa en 1 el Contador de Muestras.
- 3 Se calcula la parte Real.
- 4 Se calcula la parte Imaginaria.
- 5 Se determina la Magnitud del fador en base a la parte Real y la parte Imaginaria.
- 6 Se determina el Ángulo mediante el Coseno Inverso de Referencia.

4.3.1 PROGRAMA PRINCIPAL (PP)

El programa principal, en forma de diagrama de flujo, se muestra en la figura 4.5. Como se observa, se realiza la inicialización de periféricos, mediante comandos específicos para la tarjeta y en base a la programación en C. Además, se habilitan las interrupciones a nivel global, a fin de permitir que los diferentes periféricos puedan interrumpir la rutina principal cuando sea necesario. En el lazo infinito, se llama la función que permite el manejo de datos para FreeMASTER, en el caso de que se requiera.

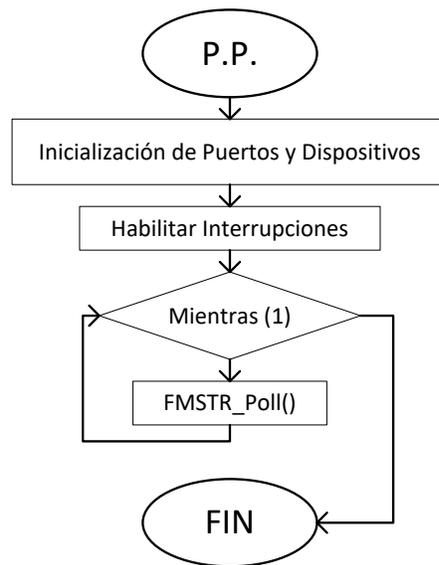


FIGURA 4.5 DIAGRAMA DE FLUJO DEL PROGRAMA PRINCIPAL.

4.3.2 ADCS

Debido a que se tienen dos tipos de ADC, se debe realizar la configuración de cada tipo de manera diferente. Para este caso, se cuenta con 4 canales del ADC de Aproximaciones Sucesivas, del cual sólo se utilizarán 3. Estos canales responden bajo la misma función de interrupción, pero acorde al canal que terminó su conversión, es el comando correspondiente que se ejecuta. Para los ADC Sigma-Delta, se realiza la configuración independiente de cada canal, con sus características propias, y al final, se inicializan todos los canales de una sola vez, asignando la velocidad del reloj que se requiera y su divisor.

Diseño de una Unidad de Medición Fasorial con tecnología ARM

A continuación se presentan las tablas 4.1 y 4.2, las cuales contienen la configuración de cada uno de los canales utilizados para los ADC Sigma-Delta y de Aproximaciones Sucesivas, respectivamente, haciendo referencia a los parámetros propios de cada función de inicialización.

TABLA 4.1 CONFIGURACIÓN DEL ADC SIGMA-DELTA.

ADC Sigma-Delta		
AFE_ChInit (inicialización del canal 0)	Canal:	CH0
	Configuración:	Disparo por Software, Decimador 1024
	Retraso:	0 * AFE_SCT
	Prioridad:	PRI_LVL0
	Función de Retorno:	Deshabilitada
AFE_ChInit (inicialización del canal 1)	Canal:	CH1
	Configuración:	Disparo por Software, Decimador 1024
	Retraso:	2 * AFE_SCT
	Prioridad:	PRI_LVL0
	Función de Retorno:	Deshabilitada
AFE_ChInit (inicialización del canal 2)	Canal:	CH2
	Configuración:	Disparo por Software, Decimador 1024
	Retraso:	4 * AFE_SCT
	Prioridad:	PRI_LVL0
	Función de Retorno:	Deshabilitada
AFE_Init (inicialización de todos los canales)	Configuración:	Reloj: AFE_PLL_CLK Divisor: AFE_DIV1 Frecuencia del reloj: AFE_CLK

TABLA 4.2 CONFIGURACIÓN DEL ADC DE APROXIMACIONES SUCESIVAS.

ADC de Aproximaciones Sucesivas		
ADC_Init (inicializa todos los canales del ADC)	Configuración:	16 bits, Disparo de Hardware
	Promedio de Hardware:	Deshabilitado
	Configuración del Canal A:	Salida Sencilla, Int. habilitadas, AD0.
	Configuración del Canal B:	Salida Sencilla, Int. habilitadas, AD1.
	Configuración del Canal C:	Salida Sencilla, Int. habilitadas, AD2.
	Configuración del Canal D:	Canal Deshabilitado
	Prioridad:	PRI_LVL0
	Función de Retorno:	Callback_ADC

La figura 4.6 presenta la secuencia que siguen los canales de los ADC para la medición de los valores. Acorde a [42], la señal de voltaje es muestreada por un ADC de Aproximaciones Sucesivas con una entrada multiplexada, así, las seis señales (3x voltajes y 3x corrientes) no pueden ser

muestreadas al mismo tiempo. El muestreo de las diferentes señales en fase debe ser desplazado en el tiempo. Esto se implementa de manera sencilla usando la función de inicio con retardo de los AFE. Cada inicio de canal del AFE es retrasado respecto al canal previo. El canal CH0 comienza su conversión en el tiempo $0 * AFE_SCT$, el canal CH1 comienza su conversión en el tiempo $1 * AFE_SCT$ y el canal CH2 lo hace en el tiempo $2 * AFE_SCT$ (AFE_SCT , es más grande que un tiempo de conversión del SAR más el tiempo de intercambio, o switcheo, del multiplexor). La interconexión entre el AFE y el SAR se realiza mediante el periférico XBAR. La bandera de conversión completa del AFE en modo de conversión continua ($AFE\ COCO\ CHx$) se utiliza para el disparo por hardware del inicio de la conversión para el SAR. [42]

El método para compensar error de desplazamiento generado por sensores de corriente puede ser solucionado a base de software; este método ahorra recursos del microcontrolador. [42]

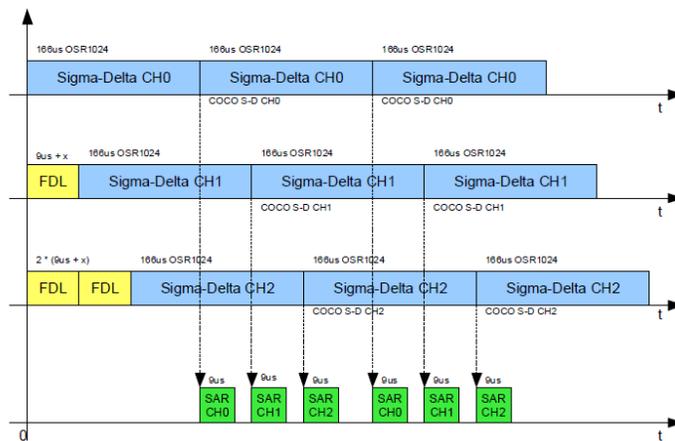


FIGURA 4.6 CADENA DE MEDICIÓN PARA DOS SEÑALES TRIFÁSICAS.

Es así como se realiza la secuencia de muestreo de las 6 señales con los dos tipos de ADC. Una vez que el último canal del SAR termina la conversión, se utiliza la función de retorno de ese canal para leer las mediciones y así, procesarlas.

El diagrama de secuencia que explica la metodología utilizada para este proceso es presentado a continuación, en la figura 4.7, donde se observa que una vez que se tienen las conversiones completas, en la función de retorno del canal C se obtiene la asignación de las mediciones a las variables correspondientes para su análisis.

Diseño de una Unidad de Medición Fasorial con tecnología ARM

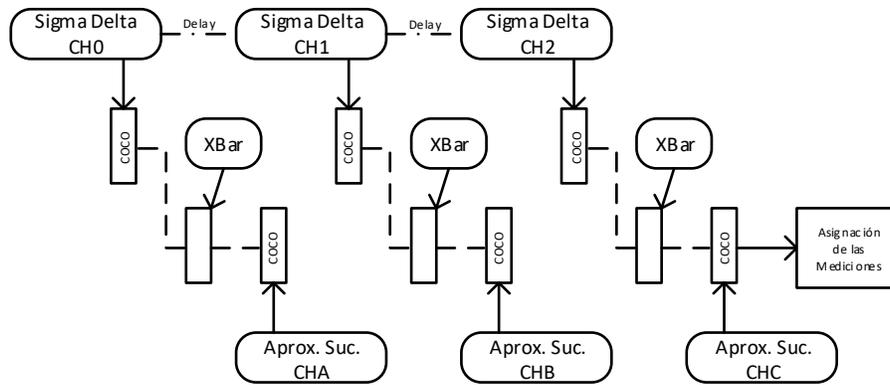


FIGURA 4.7 DIAGRAMA DE SECUENCIA DEL PROCESO DE MEDICIÓN.

4.3.3 FRECUENCÍMETRO

Como ha sido mencionado en la sección 2.4, se emplea la metodología de detectar los cruces por cero cuando existe una pendiente negativa, para determinar la frecuencia en tiempo real de la señal sinusoidal muestreada por los convertidores de la tarjeta. Así, en dicha sección, se expresó la ecuación (2.18) para el cálculo de la fracción de muestra, donde se explica que y_1 representa el valor de la muestra anterior y y_2 el valor de la muestra actual.

De esta manera, considerando las variables actuales, se tiene la ecuación 4.1 para el caso de estudio.

$$fraccion_muestra = \frac{muestra_anterior}{muestra_actual + muestra_anterior} \quad (4.1)$$

Cabe mencionar que las señales procesadas por los ADC, no precisamente cruzan por cero ya que son montadas sobre un nivel de CD, conocido como Offset, para su codificación; por ello, se debe monitorear su cruce a través de ese Offset agregado a su acondicionamiento.

El algoritmo general para realizar esta operación, como se ha mencionado, requiere como precondition el primer cambio de signo de la señal bajo estudio, de positivo a negativo, para el inicio del conteo de muestras necesarias para el cálculo. Los pasos a seguir se enlistan a continuación.

- 1 Inicialización de Variables.
- 2 Si la Muestra Actual es Menor o Igual a Cero y la Muestra Anterior es Positiva:
 - 2.1 Si la Muestra Actual es Menor a Cero:
 - 2.1.1 Se obtiene el valor absoluto de la Muestra Actual.
 - 2.1.2 Se calcula la Fracción de Muestra Nueva.
 - 2.1.3 Se calcula la Fracción de Muestra Actual.
 - 2.1.4 Se calcula la Suma de Muestras.
 - 2.1.5 Se calcula la Frecuencia.
 - 2.5.6 Se iguala la Suma de Muestras a la Fracción de Muestra Actual.
 - 2.2 Si la Muestra Actual es Cero:
 - 2.2.1 Se calcula la Frecuencia.
 - 2.2.2 La Suma de Muestras se iguala a Cero.
 - 2.3 Si la Bandera de Frecuencia es igual a Cero:
 - 2.3.1 Se llena el Buffer de Frecuencia para Ordenarlos.
 - 2.3.2 Se incrementa el Contador para el Buffer.
 - 2.3.3. Si el Contador para el Buffer es igual al Tamaño del Buffer:
 - 2.3.3.1 Se reinicia el Contador del Buffer.
 - 2.3.3.2 Se iguala la Bandera de Frecuencia en 1.
- 3 Cualquier caso distinto al punto 2:
 - 3.1 Se incrementa la Suma de Muestras en 1.
- 4 La Muestra Anterior se iguala a la Muestra Actual.

Se realizaron con anterioridad pruebas que permitieron determinar que entre menor sea el número de muestras son mejores los resultados para este algoritmo. Se determinó el uso de 10 muestras por ciclo, pues brinda resultados aceptables.

A continuación, la figura 4.8 muestra el diagrama de flujo del frecuencímetro, de manera general.

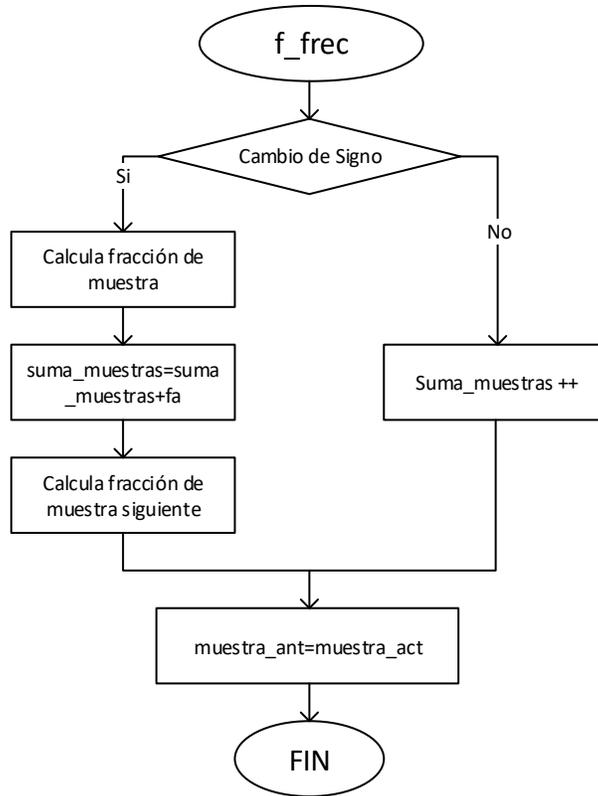


FIGURA 4.8 DIAGRAMA DE FLUJO DEL FRECUENCÍMETRO.

4.3.4 **DFT**

El algoritmo de cálculo de la Transformada Discreta de Fourier permite obtener el valor de la magnitud y ángulo de las señales bajo estudio, así como sus componentes rectangulares (parte Real y parte Imaginaria), para ser utilizadas por algún elemento para el estudio de la red.

Así, la figura 4.9 muestra el diagrama de flujo del algoritmo implementado, en el cual se guardan los valores de la parte real, parte imaginaria, magnitud y ángulo de los fasores de las señales que se muestrearon. Como se puede observar, se debe realizar la multiplicación de los valores medidos por la función ortogonal correspondiente (tanto el seno, como el coseno) y acumular dicho valor. Después, para calcular la parte real y la parte imaginaria del fador se utilizan las ecuaciones (2.12) y (2.13). El valor de la magnitud del fador se calcula con la ecuación (2.14).

Diseño de una Unidad de Medición Fasorial con tecnología ARM

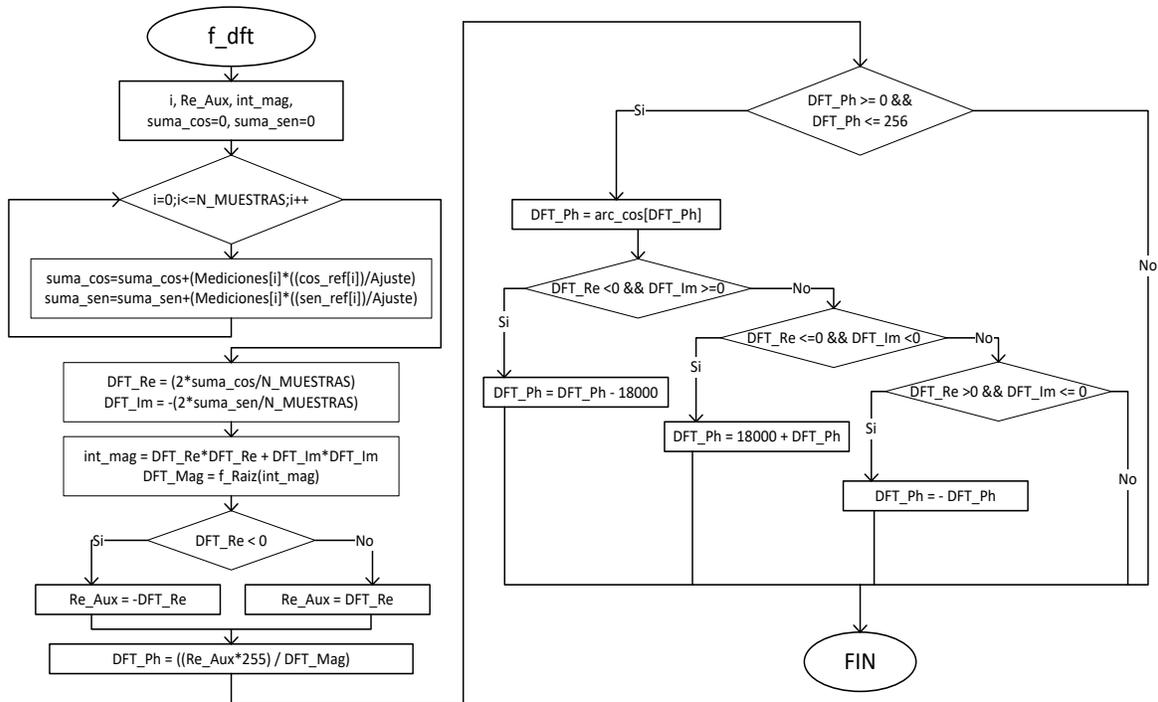
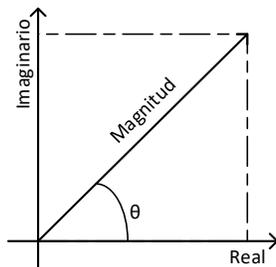


FIGURA 4.9 DIAGRAMA DE FLUJO DEL ALGORITMO DE LA DFT.

Para el cálculo del ángulo del fasor, se emplea usualmente la ecuación (2.15); sin embargo, cuando el ángulo tiene valor igual a 90° y 270° la parte real es cero, lo que provoca que la división entre imaginario y real tienda a infinito. Por ello, se realiza una comparación con una tabla de búsqueda para diferentes valores del coseno inverso, para evitar la presencia de discontinuidad, puesto que el divisor es la magnitud del fasor y, si existe medición de la señal, dicha magnitud es diferente de cero. Por ello, el coseno también permite obtener una relación entre la magnitud del fasor y el valor real, como se muestra en la figura 4.10.



$$\cos(\theta) = \frac{\text{cateto adyacente}}{\text{hipotenusa}} = \frac{\text{Real}}{\text{Magnitud}}$$

$$\theta = \cos^{-1}\left(\frac{\text{cateto adyacente}}{\text{hipotenusa}}\right) = \cos^{-1}\left(\frac{\text{Real}}{\text{Magnitud}}\right)$$

FIGURA 4.10 CÁLCULO DEL ÁNGULO DEL FASOR

Cabe mencionar que la tabla del coseno inverso sólo contiene valores positivos, pues se trabaja solo en el primer cuadrante (valor real y valor imaginario positivos), por lo que se calcula el valor absoluto del valor real para poder determinar el valor al que apuntará el ángulo dentro de la tabla de referencia. Una vez obtenido dicho valor, se hace una comparación de los valores de la parte real e imaginaria de los fasores, para determinar el cuadrante donde se encuentra el fador, como se muestra en la figura 4.11.

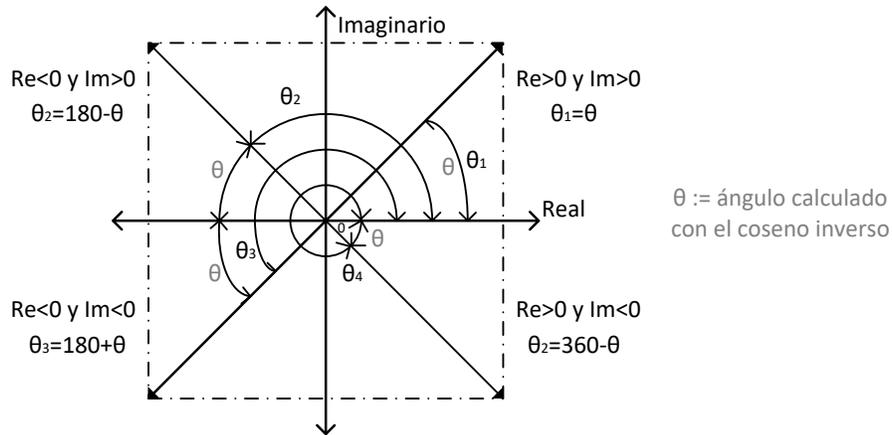


FIGURA 4.11 CÁLCULO DEL ÁNGULO EN LOS 4 CUADRANTES.

Así, con un ajuste matemático, se determina el valor real del ángulo del fador, como se muestra en la tabla 4.3.

TABLA 4.3 CÁLCULO DEL ÁNGULO REAL DEL FASOR.

Parte Real	Parte Imaginaria	Cuadrante	Ángulo
Positiva	Positiva	1	$\theta = \cos^{-1}(Real /Magnitud)$
Negativa	Positiva	2	$\theta = 180 - \cos^{-1}(Real /Magnitud)$
Negativa	Negativa	3	$\theta = \cos^{-1}(Real /Magnitud) - 180$
Positiva	Negativa	4	$\theta = -\cos^{-1}(Real /Magnitud)$

4.3.5 CÁLCULO DEL VALOR RMS

Uno de los valores que calcula este programa es el RMS, utilizando los valores de las mediciones obtenidas por los ADC de la tarjeta. La figura 4.12 muestra el diagrama de flujo general para la rutina de cálculo del valor eficaz. Como se observa, una vez obtenida la suma del valor cuadrático de las mediciones, se divide entre el número de muestras y después, se debe obtener su raíz cuadrada. Para ello, se implementa una rutina que nos permite obtener dicho valor.

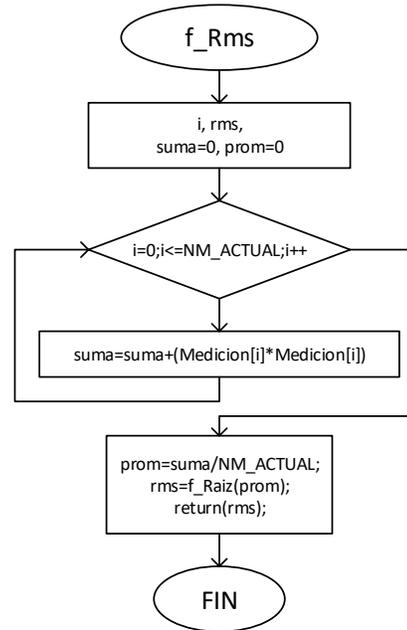


FIGURA 4.12 DIAGRAMA DE FLUJO DE LA FUNCIÓN RMS.

4.3.6 CÁLCULO DE LA RAÍZ CUADRADA

De manera que el programa no tiene una función que permita el cálculo rápido de la raíz cuadrada de un número, es necesario desarrollar un algoritmo que permita obtener dicho cálculo. Así, la función que realiza el cálculo de la raíz cuadrada de un número, es descrita mediante el diagrama de flujo de la figura 4.13. Mediante dos condicionamientos, realiza divisiones hasta encontrar el valor de la raíz del número introducido.

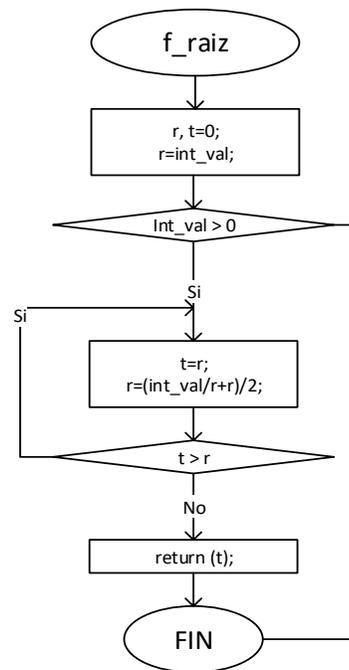


FIGURA 4.13 DIAGRAMA DE FLUJO DE LA FUNCIÓN RAÍZ.

4.4 SISTEMA DE POSICIONAMIENTO GLOBAL

Este sistema tiene como objetivo principal la sincronización de la Unidad de Medición Fasorial mediante el pulso por segundo que recibe de un satélite, capturando el tiempo exacto de la medición de los fasores. El diagrama de secuencia de esta sección se ilustra en la figura 4.14.

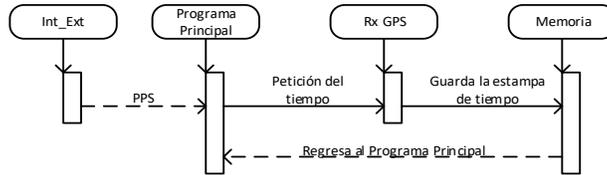


FIGURA 4.14 DIAGRAMA DE SECUENCIA DEL GPS.

Se utiliza un pin de entrada/salida de propósito general de la tarjeta (PTD7) para capturar el pulso por segundo proveniente del GPS. El diagrama de flujo general, mostrado en la figura 4.15, muestra la acción que debe realizar la interrupción externa cuando se reciba el pulso por segundo.

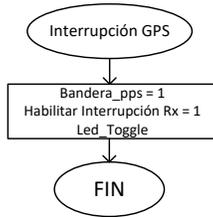


FIGURA 4.15 DIAGRAMA DE FLUJO DE LA INTERRUPCIÓN EXTERNA.

El estampado de tiempo es capturado por un puerto serial en el momento que se habilita la interrupción del Rx del GPS. El diagrama de flujo para esta rutina se muestra en la figura 4.16.

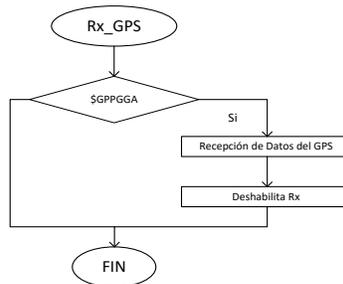


FIGURA 4.16 DIAGRAMA DE FLUJO DE LA RECEPCIÓN DEL ESTAMPADO.

Diseño de una Unidad de Medición Fasorial con tecnología ARM

En la interrupción, se obtienen los primeros 16 caracteres, donde se encuentran aquellos que indican el UTC en el formato hh.mm.ss.d. Se guardan los valores del UTC en un arreglo para poder obtener el segundo del centenario (SOC, *Seconds of Century*). La interrupción de recepción de datos del GPS es deshabilitada cuando se finalizó la recepción del UTC y es habilitada de nuevo por el pps. [43] Los datos proporcionados por el módulo RXM-SG GPS #28505 incluyen diferentes mensajes de salida NMEA, como se muestra en la tabla 4.4, por lo que se debe de seleccionar el mensaje correspondiente del que se quiere leer la estampa de tiempo, en este caso, se lee el mensaje GGA (Datos Arreglados del Sistema de Posicionamiento Global).

TABLA 4.4 MENSAJES DE SALIDA NMEA

Protocolo	Ejemplo
GGA	\$GPGGA,145754.000,1929.9516,N,09908.0148,W,1,07,1.1,2272.7,M,-9.0,M,,0000*64
GGL	\$GPGLL,1929.9516,N,09908.0148,W,145754.000,A,A*40
GSA	\$GPGSA,A,3,17,02,06,24,28,19,05,,,,,2.2,1.1,2.0*33
GSV	\$GPGSV,3,1,09,06,69,003,31,02,61,252,27,19,40,019,41,28,34,115,48*7 ^a \$GPGSV,3,2,09,24,32,288,24,17,31,041,44,05,13,187,35,12,11,322,27*79 \$GPGSV,3,3,09,13,02,209,*4B
RMC	\$GPRMC,145754.000,A,1929.9516,N,09908.0148,W,0.00,,031016,,A*6C
VTG	\$GPVTG,,T,,M,0.00,N,0.0,K,A*13

Los campos del mensaje en protocolo NMEA 0183 se realizará en base a la siguiente lectura, cuyo su significado de los principales campos del formato se observan en la tabla 4.5.

*\$GPGGA,145754.000,1929.9516,N,09908.0148,W,1,07,1.1,2272.7,M,-9.0,M,,0000*64*

TABLA 4.5 SIGNIFICADO DE LOS CAMPOS DEL MENSAJE DEL GPS UTILIZADOS.

CAMPO	SIGNIFICADO
\$GPGGA	Identificador del Mensaje.
145754.000	Indica el UTC, en formato hhmmss.d

Los datos posteriores al UTC no se ocupan para este caso, es información de la velocidad de la Tierra y posicionamiento. [43]

CAPÍTULO 5 PRUEBAS

En este capítulo se analiza la Unidad de Medición Fasorial, mostrando resultados de las mediciones realizadas por la misma. Se hace el análisis de la respuesta ante diferentes señales de prueba introducidas por el Generador de Señales.

5.1 INTRODUCCIÓN

Como se mencionó anteriormente, es necesario definir, con señales en tiempo real, un tiempo de referencia para medir los ángulos de fase en forma sincronizada. El estándar IEEE 1344-1995 define el inicio de un segundo como el tiempo de referencia para establecer el valor del ángulo del fasor. [13] La utilidad de un fasor aumenta si pueden ser definidos a una base de tiempo común, logrando esto si la señal de entrada medida en varios sitios se sincroniza una respecto a la otra. El estándar mencionado define una forma de onda en estado estacionario como aquella cuya magnitud, frecuencia y ángulo de fase no varían con el tiempo. [21]

5.2 MEDICIÓN DE FRECUENCIA

En esta sección se muestran las pruebas al algoritmo del cálculo de la frecuencia, introduciendo señales a la frecuencia nominal y fuera de ella. La primera prueba se realiza con una señal sinusoidal sin perturbaciones a una frecuencia nominal de 60 Hz, como la mostrada en la figura 5.1. Cabe mencionar que se utilizó el algoritmo descrito en la sección 2.4, y también se implementó un Filtro Mediana, para desechar los valores que resultaran extremadamente erróneos. Los resultados se muestran en la tabla 5.1.

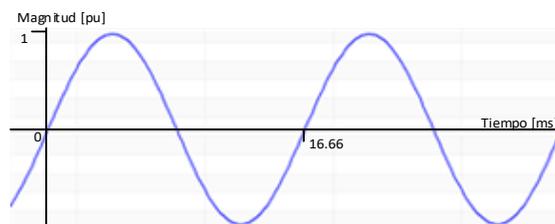


FIGURA 5.1 SEÑAL DE ENTRADA A 60 HZ, PURA.

TABLA 5.1 CÁLCULO DE LA FRECUENCIA DE SEÑAL DE ENTRADA SINUSOIDAL PURA.

Número de Muestras por ciclo que considera el Algoritmo	Frecuencia de Entrada [Hz]	Frecuencia Calculada por el Algoritmo Cruce por Cero [Hz]		Error Relativo [%]	
		Media	Mediana	Promedio	Mediana
10	60.0	60.0002	60.00	0.00033	0.000

En la primera columna se presenta el número de muestras empleadas por el algoritmo, en la siguiente columna la frecuencia de la señal de entrada, en la tercer columna se muestra la frecuencia calculada por el algoritmo, tanto el promedio de valores obtenidos sin filtro para un espacio muestral de 100 valores, como el valor obtenido con el filtro mediana implementado, y en la última columna se muestra el error de la estimación de la frecuencia. Se observa que el error es menor al 0.001%, por lo que la frecuencia obtenida por el filtro mediana y sin él, prácticamente es la misma.

Para determinar la exactitud y precisión de los elementos de medición, se requiere la determinación de los valores de error relativo respecto a la media, para el caso de la exactitud y el valor de la desviación estándar para el caso de la precisión. [50] Así, para determinar dichos valores de precisión y exactitud de la medición de frecuencia del medidor, se obtuvieron 100 lecturas mostradas en la tabla 5.2, bajo la condición de una señal de entrada sinusoidal pura de 60 Hz.

TABLA 5.2 RESULTADOS DE MEDICIÓN DE FRECUENCIA, SEÑAL DE ENTRADA SINUSOIDAL PURA.

Valor calculado por el Algoritmo [Hz]	Número de veces que se Repite el valor en el espacio muestral	Media [\bar{f}]	Varianza [σ^2]	Desviación Estándar [σ]
59.96	1	60.0002	0.0001579	0.012568
59.97	2			
59.98	5			
59.99	22			
60	40			
60.01	20			
60.02	7			
60.03	2			
60.04	1			

Con los valores obtenidos en las tablas 5.1 y 5.2, se observa que se tiene un error máximo de 0.0002 Hz o 0.00033%, por lo que la exactitud es de ± 0.0002 . Mientras que la precisión del medidor tiene una desviación estándar de 0.012568 Hz.

Ahora, sabiendo que la red eléctrica no cuenta con señales sinusoidales puras, se realizó una prueba con la señal de entrada afectada con armónicos mediante el generador de señales. Los armónicos considerados son el tercero (10%), quinto (20%) y séptimo (30%).

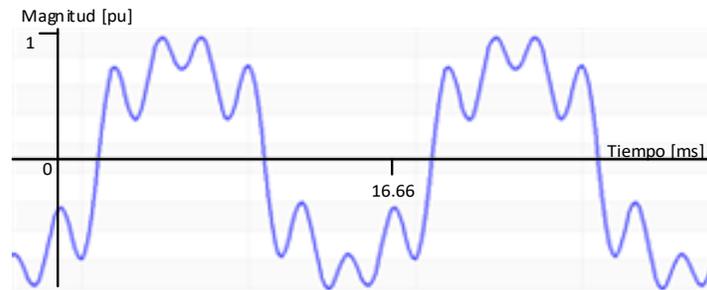


FIGURA 5.2 SEÑAL DE ENTRADA A 60 HZ CON PRESENCIA DE ARMÓNICOS.

Se observa en la figura 5,2 se tienen algunos cambios de dirección de la señal dentro del periodo, pero ninguna genera un cambio de signo (de positivo a negativo), por lo que el algoritmo trabaja de manera correcta, y no detecta ningún cruce por cero fuera de aquel que necesita para trabajar. La tabla 5.3 muestra la frecuencia estimada al transcurrir un ciclo de la señal.

TABLA 5.3 CÁLCULO DE LA FRECUENCIA DE SEÑAL DE ENTRADA SINUSOIDAL CON ARMÓNICOS.

Número de Muestras por ciclo que considera el Algoritmo	Frecuencia de Entrada [Hz]	Frecuencia Calculada por el Algoritmo Cruce por Cero [Hz]		Error Relativo [%]	
		Media	Mediana	Media	Mediana
10	60.0	60.0004	60.00	0.00066	0.000

De igual manera, se observa en la tabla 5.3 que el algoritmo presenta un error menor al 0.001% sin aplicar el filtro mediana a los cálculos de frecuencia. Sin embargo, para disminuir ese valor, se tiene el filtro mediana, que genera un error de 0%., lo que brinda una mayor exactitud. Así, se tiene un error del 0.001166%, por lo que la exactitud es de ± 0.0007 Hz.

Ahora, para determinar la precisión, se tienen las lecturas mostradas en la tabla 5.4 bajo la condición de una señal sinusoidal de 60 Hz con contenido armónico (3°, 5° y 7° armónico).

Diseño de una Unidad de Medición Fasorial con tecnología ARM

TABLA 5.4 RESULTADOS DE MEDICIÓN DE LA FRECUENCIA, SEÑAL CON CONTENIDO ARMÓNICO.

Valor calculado por el Algoritmo [Hz]	Número de veces que se Repite el valor en el espacio muestral	Media $[\bar{f}]$	Varianza $[\sigma^2]$	Desviación Estándar $[\sigma]$
59.88	1	60.0004	0.000769	0.02774
59.92	1			
59.94	1			
59.97	5			
59.98	10			
59.99	13			
60	36			
60.01	15			
60.02	10			
60.03	4			
60.04	2			
60.07	1			
60.16	1			

Con los valores obtenidos en la tabla 5.4, se observa que la precisión del medidor tiene una desviación estándar de 0.02774 Hz.

Ahora, se realizan pruebas con frecuencias por encima y debajo del valor nominal con una señal de entrada pura. La tabla 5.5 muestra los resultados brindados por el algoritmo implementado.

TABLA 5.5 CÁLCULO DE LA FRECUENCIA DE SEÑAL CON FRECUENCIA POR ENCIMA Y POR DEBAJO DE LA NOMINAL.

Número de Muestras por ciclo que considera el Algoritmo	Frecuencia de Entrada [Hz]	Frecuencia Calculada por el Algoritmo Cruce por Cero [Hz]		Error Relativo [%]	
		Media	Mediana	Media	Mediana
10	65.00	65.01	65.00	0.015	0.0
	64.00	64.02	64.00	0.031	0.0
	63.00	62.98	63.00	0.031	0.0
	62.00	62.01	62.00	0.016	0.0
	61.50	61.49	61.49	0.016	0.016
	61.00	61.00	61.00	0.0	0.0
	60.50	60.48	60.50	0.033	0.0
	60.10	60.11	60.10	0.016	0.0
	59.90	59.91	59.90	0.016	0.0
	59.50	59.49	59.50	0.016	0.0
	59.00	59.01	59.00	0.0	0.0
	58.00	57.99	58.00	0.017	0.0
	57.00	57.01	57.01	0.017	0.017
	55.00	54.99	55.00	0.018	0.0
	53.00	53.02	53.00	0.037	0.0
	52.00	51.99	52.01	0.019	0.019
	50.00	50.01	50.00	0.020	0.0

Se observa en la tabla anterior, que el algoritmo ofrece resultados aceptables para el cálculo de la frecuencia del sistema, cuando dicha frecuencia se encuentra por debajo de la nominal, teniendo un error relativo máximo de 0.037%.

5.3 VALOR RMS

En esta sección, se reportan los resultados del cálculo del valor eficaz de las señales bajo prueba. El generador de señales brinda 6 señales sinusoidales, tres son muestreadas mediante los ADC de Aproximaciones Sucesivas y tres mediante los ADC Sigma-Delta.

Entonces, la tabla 5.6 muestra los resultados obtenidos tras la ejecución del programa. La primera columna indica las señales bajo estudio, la segunda muestra el valor pico de la señal. El valor RMS teórico calculado de dicha señal está en la tercera columna y el valor RMS calculado por el algoritmo está en la última columna.

TABLA 5.6 RESULTADOS DEL CÁLCULO DEL VALOR RMS.

Señal	Valor Pico de la Señal	Valor RMS (Teórico)	Valor RMS (Algoritmo)	Error [decimales]	Error Relativo [%]
SAR_0 (VA)	27305	19307,55	19301	6.55	0,0339
SAR_1 (VB)	27299	19303,31	19295	8.31	0,0430
SAR_2 (VC)	27302	19305,43	19297	8.43	0,0437
SD_0 (IA)	16379	11581,70	11578	3.7	0,0320
SD_1 (IB)	16381	11583,12	11579	4.12	0,0355
SD_2 (IC)	16383	11584,53	11580	4.53	0,0391

Como se observa en la tabla, el algoritmo calcula el valor RMS de manera sumamente aproximada al valor obtenido matemáticamente. En el estudio de las muestras obtenidas por los ADC de Aproximaciones Sucesivas (SAR), el algoritmo presenta un error máximo del 0.0437%, mientras que en el caso del estudio de las muestras obtenidas por los ADC Sigma-Delta (SD) se tiene un error máximo del 0.0391%, considerando los resultados del algoritmo correctos y con exactitud de 8.43 números decimales para los ADC SAR y 4.53 números decimales para los ADC SD.

Para determinar la precisión del cálculo del valor RMS del medidor fasorial, se obtuvieron las lecturas de la tabla 5.7, que muestran los valores pico de las señales y sus valores RMS teóricos. Se hace toma el valor calculado por el algoritmo 5 veces para la determinación de la media y de la desviación estándar.

TABLA 5.7 RESULTADOS DEL CÁLCULO DEL VALOR RMS PARA DETERMINAR PRECISIÓN.

Señal	Valor Pico	Valor RMS (Teórico)	Valor RMS (Algoritmo) de 5 mediciones					Media	Desviación Estándar $[\sigma]$
			1	2	3	4	5		
Va (SAR0)	21844	15446,04	15441	15440	15443	15442	15440	15441	1,166
	24575	17377,14	17371	17375	17374	17371	17370	17372	1,939
	27305	19307,55	19301	19305	19304	19301	19299	19302	2,191
	30036	21238,65	21231	21236	21238	21240	21229	21234	4,166
	32766	23169,06	23161	23158	23165	23161	23159	23160	2,4
Vb (SAR1)	21839	15442,50	15436	15441	15440	15439	15448	15440	3,969
	24569	17372,90	17366	17368	17375	17374	17365	17369	4,127
	27299	19303,30	19295	19301	19305	19304	19297	19300	3,878
	30029	21233,70	21225	21230	21236	21234	21227	21230	4,127
	32759	23164,11	23154	23161	23166	23165	23156	23160	4,758
Vc (SAR2)	21842	15444,62	15438	15440	15443	15439	15438	15439	1,854
	24572	17375,02	17367	17369	17374	17370	17374	17370	2,785
	27302	19305,42	19297	19299	19304	19305	19298	19300	3,261
	30032	21235,83	21227	21229	21230	21236	21229	21230	3,059
	32762	23166,23	23156	23159	23165	23159	23160	23159	2,925
Ia (SD_0)	13103	9265,22	9262	9264	9259	9266	9261	9262	2,416
	14741	10423,46	10420	10420	10418	10422	10420	10420	1,264
	16379	11581,70	11578	11580	11576	11582	11578	11578	2,039
	18017	12739,94	12736	12738	12739	12740	12736	12737	1,6
	19655	13898,18	13894	13891	13891	13898	13892	13893	2,638
Ib (SD_1)	13105	9266,63	9263	9261	9262	9265	9260	9262	1,721
	14743	10424,87	10421	10420	10421	10423	10422	10421	1,019
	16381	11583,11	11579	11581	11578	11581	11580	11579	1,166
	18019	12741,35	12737	12741	12736	12738	12738	12738	1,673
	19657	13899,59	13895	13892	13890	13897	13896	13894	2,607
Ic (SD_2)	13106	9267,34	9264	9261	9259	9267	9265	9263	2,856
	14745	10426,28	10422	10424	10420	10426	10423	10423	2
	16383	11584,53	11580	11582	11578	11584	11581	11581	2
	18021	12742,77	12738	12740	12736	12742	12738	12738	2,039
	19660	13901,71	13896	13895	13894	13901	13897	13896	2,416

Basándose en los datos proporcionados por la tabla 5.7, la precisión de la unidad de medición fasorial para la determinación del valor RMS tiene una desviación estándar máxima de 4,758 para los ADC SAR y de 2.856 para los ADC SD.

La figura 5.3 muestra los valores RMS obtenidos por el programa para el caso de las 6 señales prueba, con valores pico mostrados en la tabla 5.6.

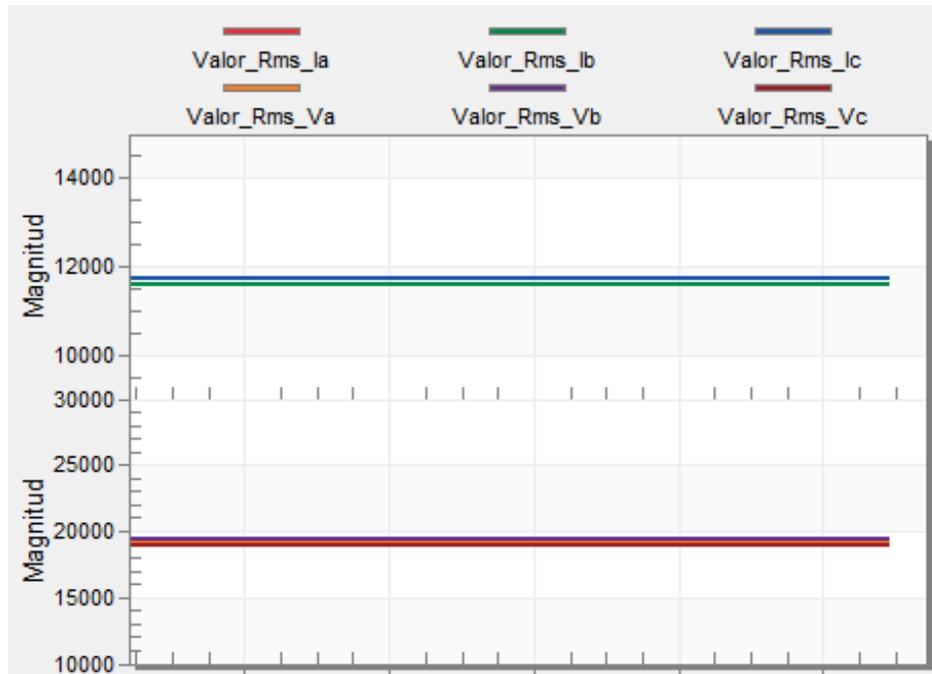


FIGURA 5.3 GRÁFICAS DEL VALOR RMS DE LAS SEÑALES PRUEBA.

5.4 CÁLCULO DE FASORES

En esta sección, se presentan los resultados del cálculo de los fasores. Se realizaron pruebas con fasores de una señal con frecuencia nominal y con frecuencia fuera de la nominal, para verificar que cumplen con el Estándar IEEE C37.118.

5.4.1 FASORES DE UNA SEÑAL CON FRECUENCIA NOMINAL

La representación de fasores $\{X_0, X_1, X_2, \dots\}$ es conducida si la señal es observada es a intervalos regulares $\{0, T_0, 2T_0, \dots, nT_0\}$. Si el intervalo T_0 es un múltiplo entero del periodo de la señal de entrada ($T = 1/f$) se obtiene un fasor constante en cada intervalo de observación. [16]

En este caso, se cumple cuando se mide una señal con frecuencia nominal de 60 Hz, pues el periodo de muestreo constante es de 1.666×10^{-4} ($1/(100 * 60)$) s. Las mediciones realizadas por la PMU

para una señal con frecuencia igual a la fundamental ayudan a crear las gráficas que se representan en la figura 5.4.

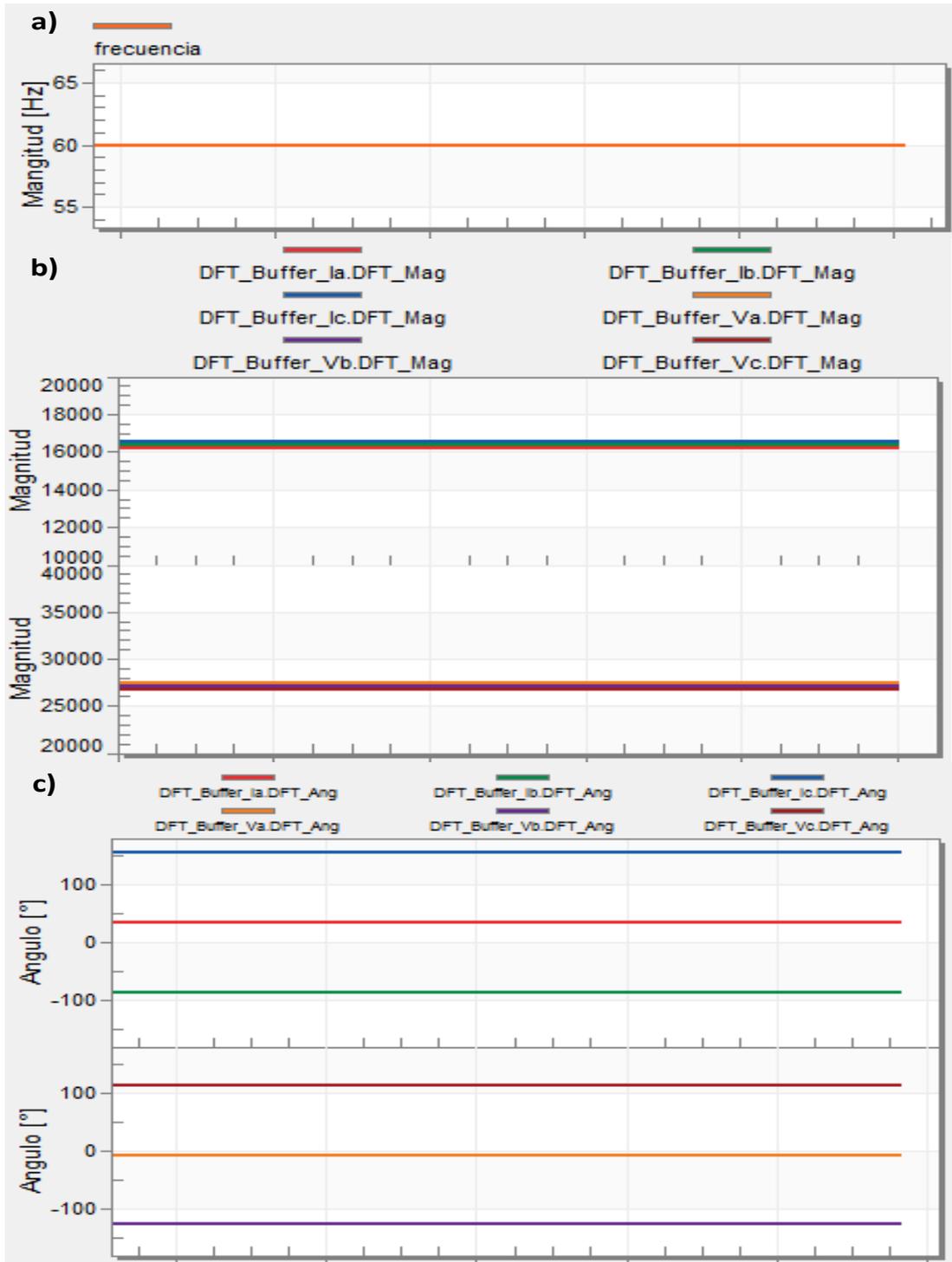


FIGURA 5.4 MAGNITUD Y ÁNGULO DEL FASOR A FRECUENCIA NOMINAL.

La figura 5.4.a) muestra el valor de la frecuencia, manteniéndose en el intervalo de observación en 60 Hz. La figura 5.4.b) representa la magnitud de los fasores cuando se tiene una frecuencia de 60 Hz, mostrando que no muestra variación en su valor al transcurrir el tiempo. Finalmente, la figura 5.4.c) es la gráfica del valor de los ángulos de los fasores, que se mantiene constante al tener el fador medido a la frecuencia nominal del sistema.

5.4.2 FASORES DE UNA SEÑAL CON FRECUENCIA FUERA DE LA NOMINAL

Como se explica en el Apéndice C, si el intervalo de observación T_0 no es múltiplo entero de T , el fador mantiene su magnitud constante, sin embargo, el ángulo de fase de la secuencia de fasores $\{X_0, X_1, X_2, \dots\}$ tendrá cambios uniformes a una razón $2\pi(f - f_0)T_0$, donde $f_0 = 1/T_0$. [13] Dicho de otra manera, al ángulo de fase presenta variaciones al tener una frecuencia diferente de la nominal. Considerando una frecuencia de 59.5 Hz, se obtienen los siguientes resultados, mostrados en la figura 5.5.

Como se observa, la figura 5.5.a) la frecuencia calculada se mantiene constante. La gráfica de la magnitud es mostrada en la figura 5.5.b), la cual presenta oscilación en el valor de los fasores. La figura 5.5.c) muestra la rotación del ángulo del fador, que se debe al valor de la frecuencia por debajo de la nominal. Cuando la frecuencia de la PMU se encuentra por debajo de la nominal, el fador rota en sentido horario, tal como se indica en [47].

Considerando una frecuencia de 60.5 Hz, se obtienen los resultados mostrados en la figura 5.6. Como se observa, la figura 5.6.a) la frecuencia calculada se mantiene constante. La figura 5.6.b), representa la magnitud del fador, la cual presenta una oscilación en el valor del fador. La figura 5.6.c) muestra la rotación del ángulo del fador, que se debe al valor de la frecuencia por encima de la nominal.

De manera contraria, si la frecuencia de la PMU se encuentra por arriba de la nominal, el fador estimado rota en sentido contra horario, a una velocidad dependiente de la lejanía del valor medido y la frecuencia nominal. [47] Así, se determina que cuando la señal de entrada tiene una frecuencia que no es igual a la nominal, el fador se somete a una rotación de en el plano complejo.

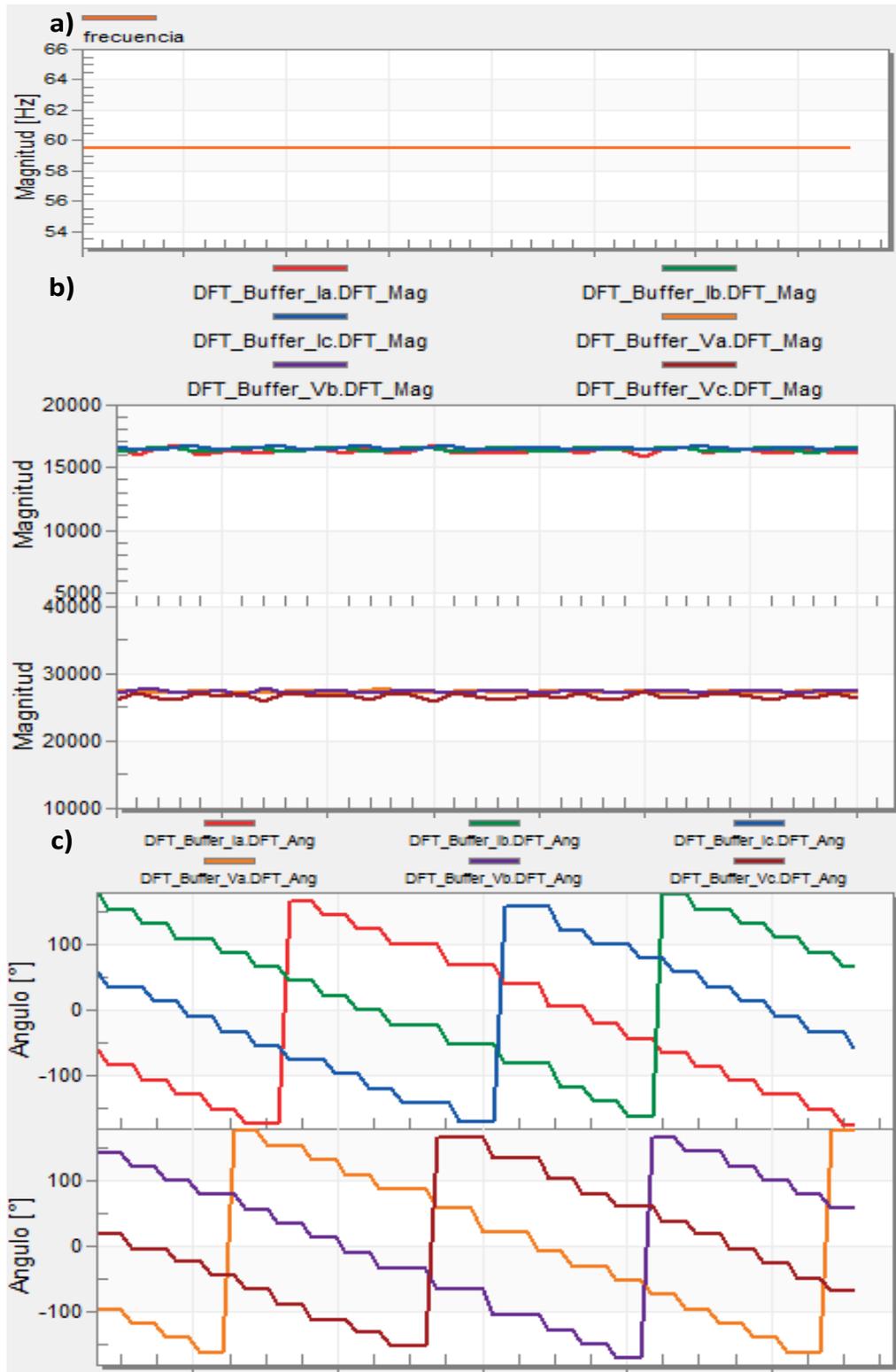


FIGURA 5.5 MAGNITUD Y ÁNGULO DEL FASOR A FRECUENCIA DEBAJO DE LA NOMINAL.

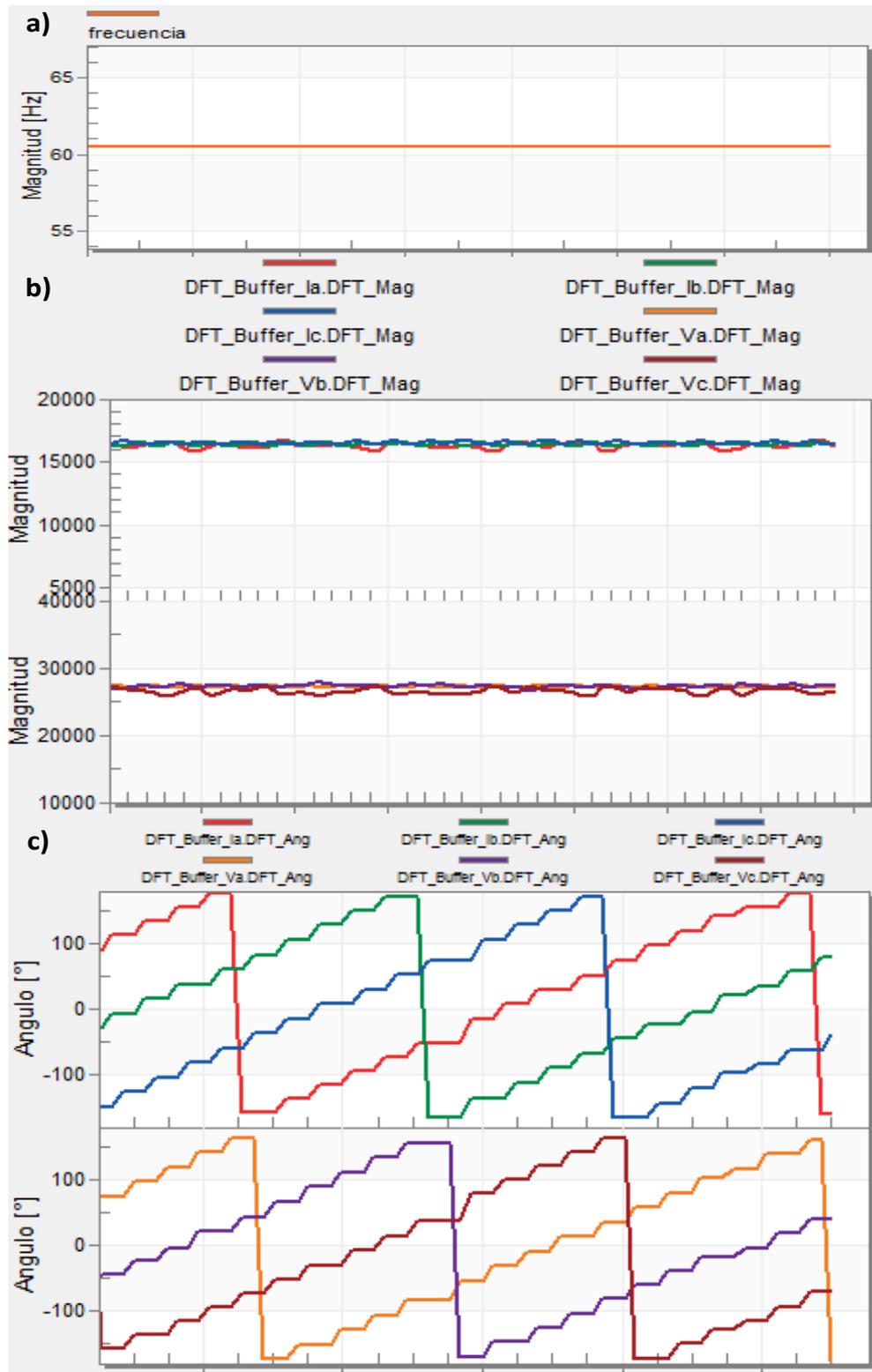


FIGURA 5.6 MAGNITUD Y ÁNGULO DEL FASOR A FRECUENCIA POR ENCIMA DE LA NOMINAL.

5.5 RESPUESTA EN EL TIEMPO DEL PMU

Como se indica en [13], para medir la respuesta en el tiempo de un PMU, se realiza un cambio a frecuencia nominal en la magnitud de la señal del 100% al 110%. El tiempo transcurrido entre el instante en el que se presenta el cambio en la magnitud y el instante en que tarda en llegar a la magnitud especificada, es el tiempo de respuesta del sistema. La gráfica de la figura 5.7 representa el cambio de magnitud en la señal de entrada.

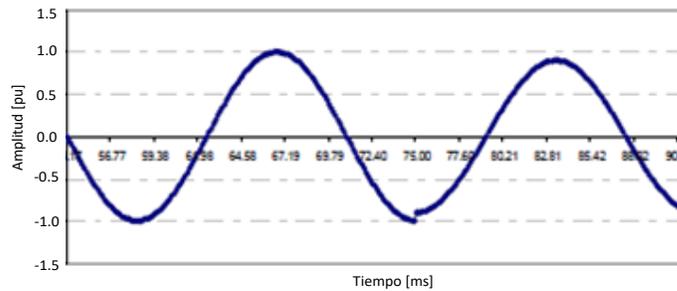


FIGURA 5.7 CAMBIO DE MAGNITUD EN LA SEÑAL DE ENTRADA.

Debido a que el desarrollo de la DFT requiere una ventana de datos para la estimación del valor del fasor, implica que se requiere un ciclo completo de la onda de entrada para determinar el nuevo valor de la magnitud de la onda. Ya que el tiempo de obtención de cada muestra, para una frecuencia nominal de 60 Hz, es de 0.16666 ms y se tienen 100 muestras por ciclo, el tiempo de respuesta equivale a 16.666 ms, lo que se muestra en la figura 5.8.

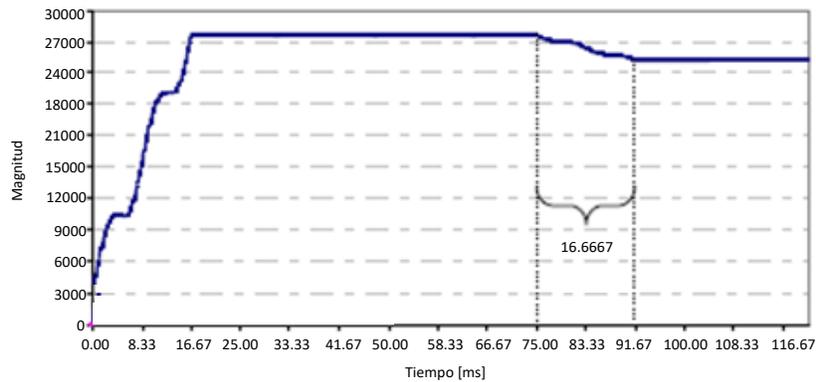


FIGURA 5.8 TIEMPO DE RESPUESTA DEL FASOR.

5.6 LÍMITES DE PRECISIÓN

Ahora bien, para que la PMU desarrollada cumpla con el Estándar C37.118.1, se calcula el vector total de error (TVE, *Total Vector Error*), que permite cuantificar el error total de la PMU [13].

La norma indica que, bajo condiciones donde x_m , ω , φ son fijos, el TVE no deberá superar el límite de 1% para el cumplimiento de determinada clase de desempeño. La ecuación (C7) permite determinar el TVE, empleando los valores medidos por el dispositivo de medición y los valores teóricos de la señal de entrada en el instante de tiempo de la medición y las condiciones conocidas de x_m , ω y φ . Así, las pruebas se realizan con una señal patrón bajo las condiciones de referencia como se define en la tabla B.3. [13]

Para realizar pruebas en estado estacionario para las clases M y P, se debe de variar solo un parámetro a la vez, es decir, cuando se realiza la prueba de magnitud, la magnitud de la señal debe ser variada, pero la frecuencia será nominal, el ángulo de fase 0 radianes, la distorsión armónica de menor del 0.2%. El TVE debe permanecer dentro del margen de error en todas las pruebas. [13]

Se definen las características de entrada de referencia con una frecuencia nominal de 60 Hz, una magnitud de 1 pu y un ángulo de fase constante.

5.6.1 CAMBIO DE MAGNITUD EN LA ENTRADA DEL PMU

En primera instancia, es necesario saber que la resolución de la PMU depende de los valores en bits de los ADC y de la rotación en bits que se emplea para los cálculos. Por ello, se tiene que el cambio más pequeño del valor medido al cual responde la PMU es 1/65536.

Entonces, para la clase P se pide que la magnitud de voltaje varíe del 80% a 120% de la señal, y la magnitud de corriente varíe de 10% al 200%. El 100% de la señal de voltaje bajo estudio corresponde a 27305, por lo que la magnitud variará de 21844 a 32766, mientras que el 100% de la señal de corriente bajo estudio corresponde a 16383, por lo que la magnitud variará de 1638.3 a 32766.

Diseño de una Unidad de Medición Fasorial con tecnología ARM

Para obtener el valor del TVE, la magnitud se varía en pasos de 5% para la tensión y pasos de 10% para la corriente hasta llegar a los límites establecidos para la clase P. Los resultados del cálculo del TVE están en la tabla 5.8.

TABLA 5.8 RESULTADOS TVE PARA EL CAMBIO DE MAGNITUD DE VOLTAJE Y CORRIENTE, CLASE P.

Parámetro	Magnitud [%]	Valores PMU		Valores Patrón		TVE (%)
		Real	Imaginario	Xr	Xi	
Voltaje	80	21842	0	21844	0	0,00916
	85	23205	0	23209,25	0	0,01831
	90	24571	0	24574,5	0	0,01424
	95	25936	0	25939,75	0	0,01446
	100	27304	0	27305	0	0,00366
	105	28670	0	28670,25	0	0,00087
	110	30033	0	30035,5	0	0,00832
	115	31397	0	31400,75	0	0,01194
	120	32764	0	32766	0	0,00610
Corriente	10	1418	819	1418,81	819,15	0,05025
	20	2838	1638	2837,62	1638,30	0,01480
	30	4256	2458	4256,43	2457,45	0,01418
	40	5674	3276	5675,24	3276,60	0,02099
	50	7094	4096	7094,05	4095,75	0,00311
	60	8513	4914	8512,86	4914,90	0,00927
	70	9930	5734	9931,67	5734,05	0,01453
	80	11350	6552	11350,48	6553,20	0,00985
	90	12769	7372	12769,28	7372,35	0,00306
	100	14188	8190	14188,09	8191,50	0,00917
	110	15608	9011	15606,90	9010,65	0,00639
	120	17025	9830	17025,71	9829,80	0,00377
	130	18445	10646	18444,52	10648,95	0,01403
	140	19863	11470	19863,33	11468,10	0,00841
	150	21283	12284	21282,14	12287,25	0,01368
	160	22700	13104	22700,95	13106,40	0,00985
170	24120	13922	24119,76	13925,55	0,01278	
180	25537	14740	25538,57	14744,70	0,01680	
190	26956	15561	26957,38	15563,85	0,01017	
200	28377	16382	28376,19	16383,00	0,00393	

Como se observa en la tabla, se logran Vectores de Error Total menores al 0.02% para las señales de voltaje y menores a 0.06% para señales de corriente, teniendo como valores máximos 0.01831% y 0.05025%, respectivamente. El criterio de precisión de TVE detecta errores de la estimación de magnitud del fasor, entonces, en la figura 5.9 se muestra la gráfica de variación en TVE en función de magnitud para el ángulo de error máximo calculado, tanto para el fasor de voltaje como el de corriente. En [48] se explica que se debe realizar la comparación del TVE según la curva correspondiente, pues influye la presencia de error en el cálculo del ángulo de la magnitud bajo análisis. La línea en color rosa, muestra el TVE cuando no se tiene error en el ángulo de fase,

mientras que la línea azul es la aproximación para cuando se tiene un error de 0.09° en la estimación del ángulo de fase. El margen rojo delimita el máximo TVE (1%) que permite el Estándar, mientras que las líneas negras delimitan el TVE obtenido por las mediciones.

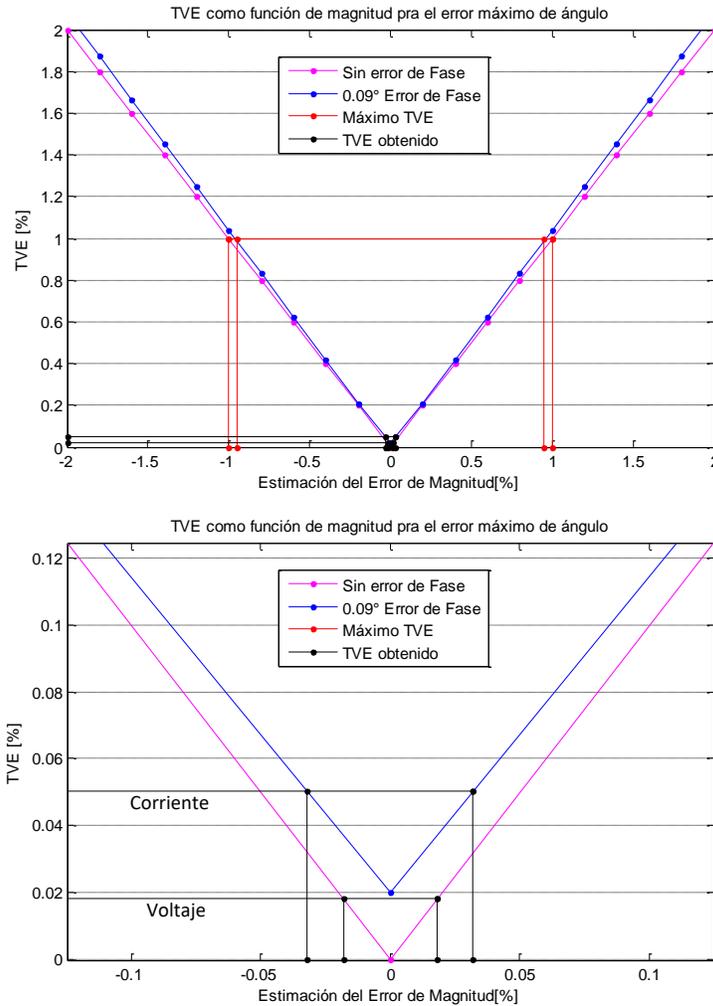


FIGURA 5.9 GRÁFICA DEL TVE DEBIDO A LA MAGNITUD, CLASE P.

Para la clase M, acorde el Estándar C37.118.1 de la IEEE, la magnitud de voltaje se debe variar del 10% al 120% y la de corriente se varía del 10% al 200%, igual que para la clase P. Entonces, bajo este criterio, sólo se hará la prueba para el caso de las señales de voltaje, pues la variación para la corriente es la misma para ambas clases. La magnitud de tensión tendrá valores desde 2730.5 hasta 32766 para esta prueba, como se muestra en la tabla 5.9.

TABLA 5.9 RESULTADOS TVE PARA EL CAMBIO DE MAGNITUD DE VOLTAJE, CLASE M.

Parámetro	Magnitud [%]	Valores PMU		Valores Patrón		TVE (%)
		Real	Imaginario	Xr	Xi	
Voltaje	10	2729	0	2730,5	0	0,05493
	15	4093	0	4095,75	0	0,06714
	20	5459	0	5461	0	0,03662
	25	6824	0	6826,25	0	0,03296
	30	8189	0	8191,5	0	0,03052
	35	9553	0	9556,75	0	0,03924
	40	10920	0	10922	0	0,01831
	45	12285	0	12287,25	0	0,01831
	50	13651	0	13652,5	0	0,01099
	55	15015	0	15017,75	0	0,01831
	60	16379	0	16383	0	0,02442
	65	17741	0	17748,25	0	0,04085
	70	19110	0	19113,5	0	0,01831
	75	20471	0	20478,75	0	0,03784
	80	21842	0	21844	0	0,00916
	85	23205	0	23209,25	0	0,01831
	90	24576	0	24574,5	0	0,00916
	95	25936	0	25939,75	0	0,01446
	100	27304	0	27305	0	0,00366
	105	28670	0	28670,25	0	0,00087
110	30033	0	30035,5	0	0,00832	
115	31397	0	31400,75	0	0,01194	
120	32764	0	32766	0	0,00610	

Como se observa, se logra un Vector de Error Total máximo para el parámetro de Voltaje con valor de 0.06714%, valor cuando se tiene un 15% de la magnitud base, y que aún está por debajo del máximo que se requiere para cumplir con el Estándar. Igual que en la clase P, se tiene la gráfica que representa la variación en TVE en función de magnitud para el ángulo de error máximo calculado, para ambos parámetros, mostrada en la figura 5.10. De manera similar a la gráfica anterior, la línea en color rosa, muestra el TVE cuando no se tiene error en el ángulo de fase, mientras que la línea azul es la aproximación para cuando se tiene un error de 0.09° en la estimación del ángulo de fase de la corriente. El margen rojo delimita el máximo TVE (1%) que permite el Estándar, mientras que las líneas negras delimitan el TVE obtenido por las mediciones.

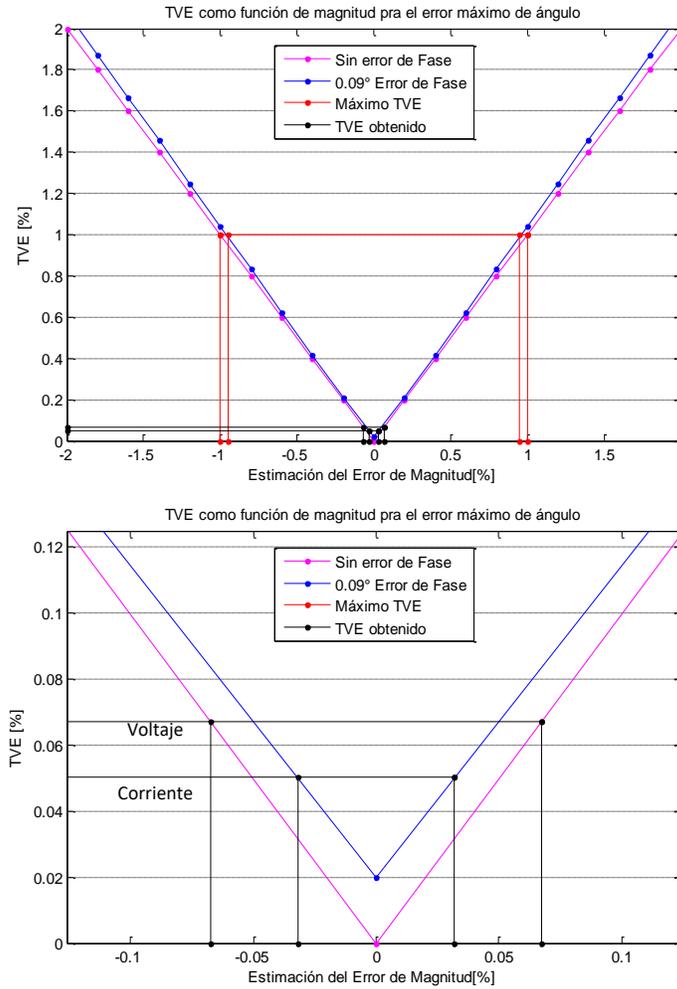


FIGURA 5.10 GRÁFICA DEL TVE DEBIDO A LA MAGNITUD, CLASE M.

5.6.2 CAMBIO DE ÁNGULO DE FASE

De acuerdo con el Estándar C37.118.1, para ambas clases, se pide que el ángulo de fase varíe de π a $-\pi$ radianes respecto a la señal base de cualquier señal bajo análisis. La tabla 5.10 muestra los resultados que se obtuvieron en la prueba, variando en pasos de $\pi/4$ hasta llegar a los límites que indica el Estándar.

El nivel máximo de TVE obtenido por las mediciones es de 0.01093%, el cual se encuentra dentro del TVE máximo requerido por el Estándar, infiriendo que cumple con él. Con dicho valor obtenido, se obtiene la gráfica de la figura 5.11 que de acuerdo al valor porcentual del error del

Diseño de una Unidad de Medición Fasorial con tecnología ARM

cálculo de la magnitud, se modifica la curva. [48] Como dicho error se mantuvo por debajo de 0.02%, se puede considerar la curva sin error de magnitud.

TABLA 5.10 RESULTADOS TVE PARA EL CAMBIO DE ÁNGULO, CLASES P Y M.

Cambio en el Ángulo [rad]	Valores PMU		Valores Patrón		TVE (%)
	Real	Imaginario	Xr	Xi	
$+\pi$	-27304	0	-27305,00	0,00	0,00366
$+\frac{3}{4}\pi$	-19305	19306	-19307,55	19307,55	0,01093
$+\frac{1}{2}\pi$	0	27304	0,00	27305,00	0,00366
$+\frac{1}{4}\pi$	19305	19308	19307,55	19307,55	0,00949
$+0$	27304	0	27305,00	0,00	0,00366
$-\frac{1}{4}\pi$	19308	-19306	19307,55	-19307,55	0,00568
$-\frac{1}{2}\pi$	0	-27304	0,00	-27305,00	0,00366
$-\frac{3}{4}\pi$	-19307	-19307	-19307,55	-19307,55	0,00285
$-\pi$	-27304	0	-27305,00	0,00	0,00366

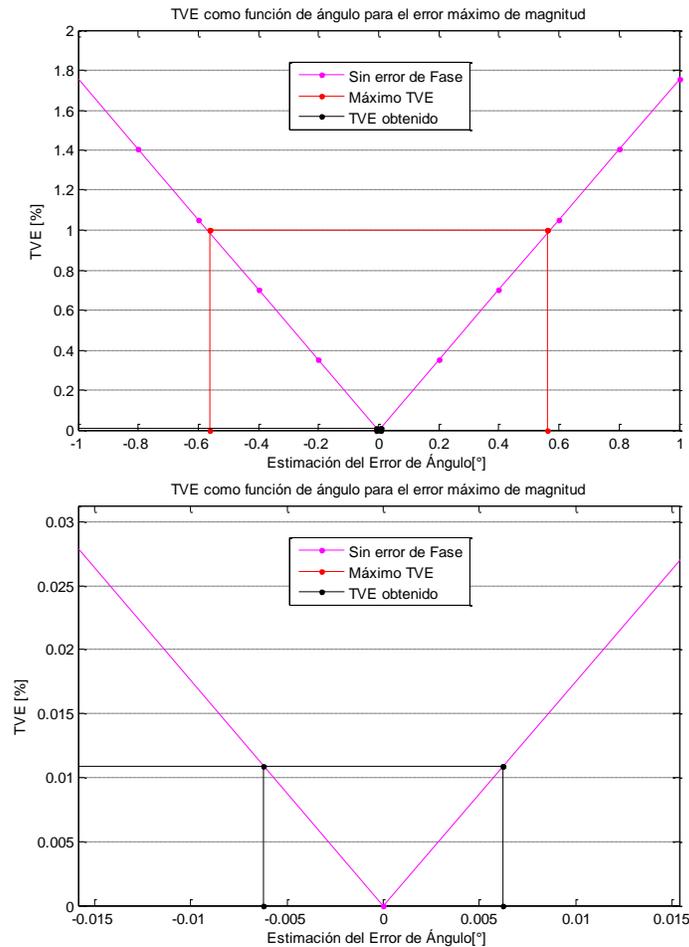


FIGURA 5.11 GRÁFICA DEL TVE DEBIDO AL CAMBIO DE ÁNGULO, CLASES P Y M.

En este caso, el error está referido al porcentaje de desviación en los grados del ángulo de fase a 60 Hz, por lo que al tener un 1% de TVE se tiene un ángulo de fase de $\pm 0.57^\circ$. [13]

5.6.3 DISTORSIÓN ARMÓNICA

Existe una importancia significativa en esta prueba, pues se comprueban los filtros armónicos pares e impares que conlleva el uso de la DFT. [47] En la clase P, se debe comprobar que el TVE del dispositivo implementado está por debajo del valor de 1% cuando se tiene la presencia del 1% de cualquier armónico en la señal bajo prueba. [48] En la figura 5.12.a) se observa la señal bajo análisis, con el 1% del 3er, 5to y 7mo armónico, dichas presencias generan una afectación de la señal, mientras que en la figura 5.12.b) se tiene la magnitud del fasor calculado por la PMU.

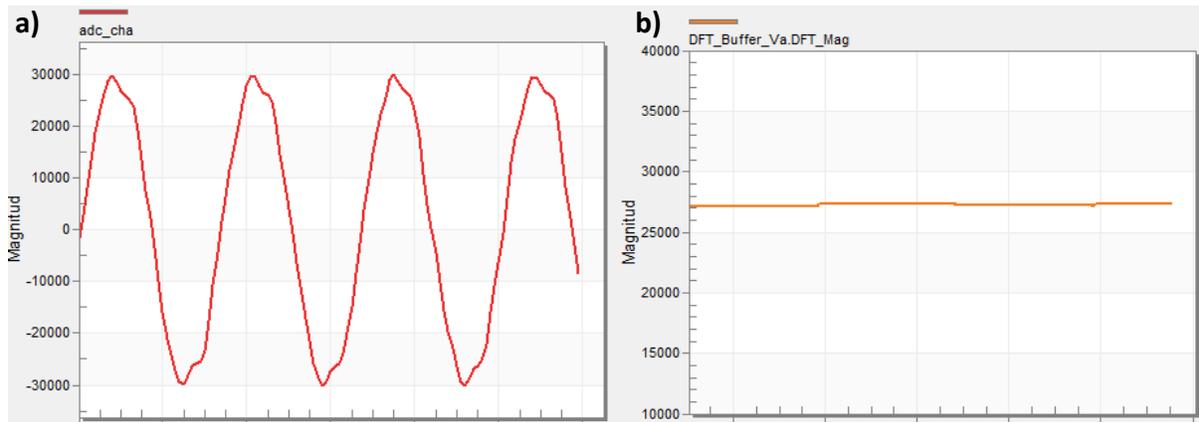


FIGURA 5.12 SEÑAL CON 1% DE CONTENIDO DE ARMÓNICOS Y MAGNITUD DEL FASOR ESTIMADO.

Los resultados de probar la señal con los armónicos que podemos obtener mediante el generador de señales, se muestran en la tabla 5.11, teniendo un TVE máximo con valor de 0.00819% presente cuando la señal está contaminada por el tercer armónico. Evidentemente, este valor de TVE no supera el máximo establecido en la norma.

TABLA 5.11 RESULTADOS TVE PARA SEÑAL CON 1% DE DISTORSIÓN ARMÓNICA, CLASE P.

Armónico	Valores PMU		Valores Patrón		TVE (%)
	Real	Imaginario	Xr	Xi	
3°	27304	2	27305	0	0.00819
5°	27304	1	27305	0	0.00518
7°	27304	0	27305	0	0,00366

Ahora bien, para la clase M se incrementa la distorsión armónica al 10%. El TVE máximo debe ser, igual que en el caso anterior, de 1% conforme al Estándar. La siguiente figura, 5.13, muestra las señales contaminadas con el 10% de armónicos y el fasor correspondiente para cada una, observando que la estimación de dicho fasor genera una magnitud constante (figura 5.13.a) 3er armónico, figura 5.13.b) 5to armónico y figura 5.13.c) 7mo armónico).

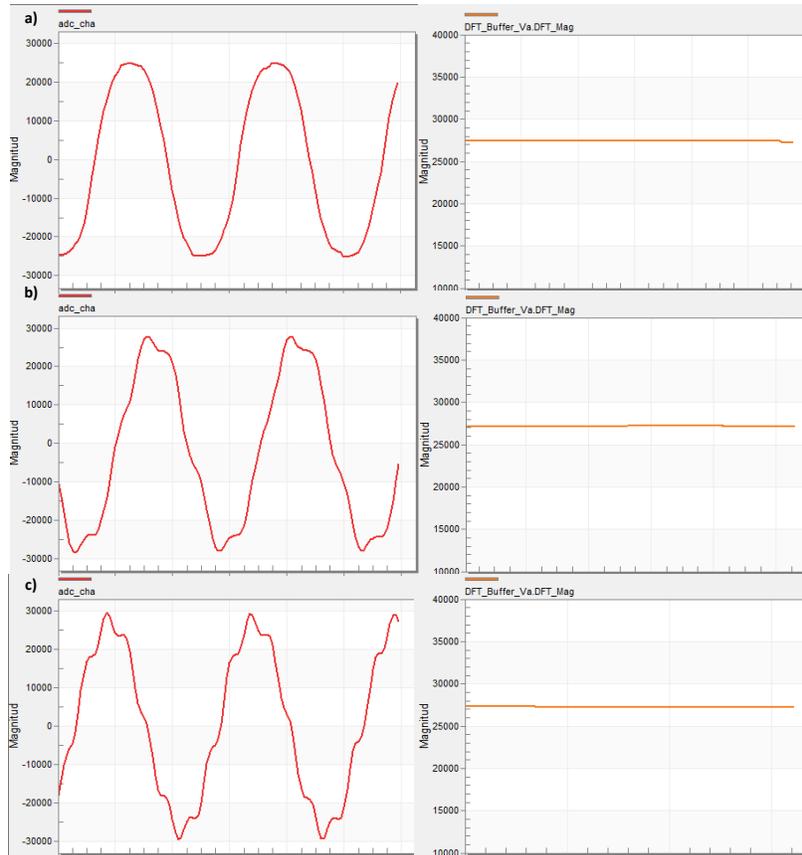


FIGURA 5.13 SEÑALES CON 10% DE CONTENIDO ARMÓNICO Y MAGNITUD DEL FASOR ESTIMADO.

Los resultados de estas pruebas se localizan en la tabla 5.12, donde se tiene un TVE máximo de 0.00822% cuando la señal es contaminada con el tercer armónico, sin superar el TVE indicado en el Estándar como máximo (1%).

TABLA 5.12 RESULTADOS TVE PARA SEÑAL CON 10% DE DISTORSIÓN ARMÓNICA, CLASE M.

Armónico	Valores PMU		Valores Patrón		TVE (%)
	Real	Imaginario	Xr	Xi	
3°	27304	3	27305	0	0,01158
5°	27304	1	27305	0	0,00518
7°	27304	1	27305	0	0,00518

5.7 COMUNICACIÓN

La Unidad de Medición Fasorial sincronizada vía satélite, mediante el módulo GPS tiene la capacidad de determinar los fasores de voltaje y corriente un sistema trifásico a 60 Hz. Las PMU's se sitúan en las subestaciones en donde se supervisa el voltaje, la corriente y la frecuencia. Dichas mediciones se envían a un concentrador de datos, correlacionando los datos a través de una etiqueta de tiempo para crear sistemas con una referencia. [46] De acuerdo al Estándar IEEE C37.118.2, una PMU debe enviar y recibir los mensajes mediante un formato específico, contando con 4 tipos de mensajes: configuración, datos, cabezal y comando. [49] Las tablas que se muestran a continuación (tablas 5.13, 5.14, 5.15 y 5.16) muestran los resultados de dichos mensajes específicamente para esta PMU.

Se utilizó una terminal serial en la PC para visualizar el mensaje enviado por la PMU y corroborar que dicho mensaje cumple con lo establecido en el Estándar. La figura 5.14 muestra la pantalla de la terminal serial que contiene el mensaje enviado por la PMU mediante el UART.

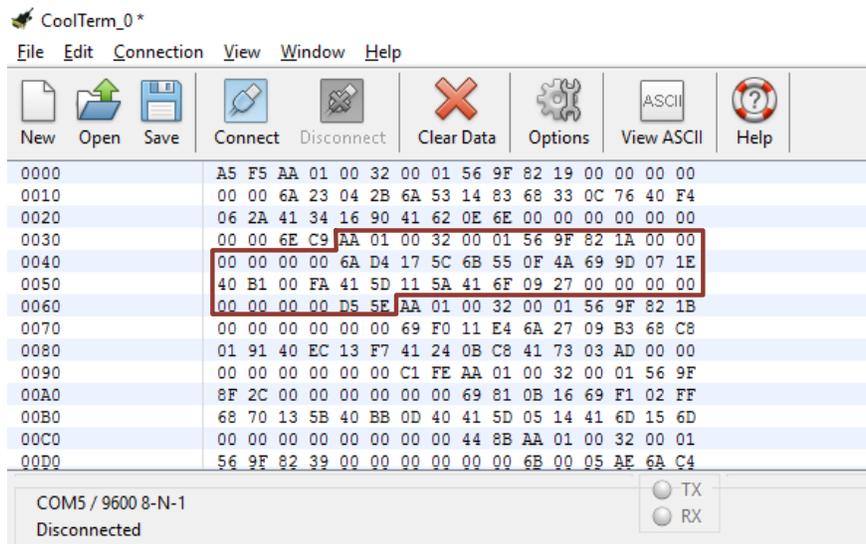


FIGURA 5.14 PANTALLA DE LA TERMINAL SERIAL CON MENSAJE DE DATOS.

Cabe mencionar, que todos los mensajes incluyen su tipo de mensaje, el tamaño de mensaje, el nombre correspondiente de la PMU, los segundos de centenario (SOC, *Seconds of Century*) basados en el UTC dado por el GPS, la fracción de segundo y calidad de tiempo, así como su comprobación de redundancia cíclica.

Diseño de una Unidad de Medición Fasorial con tecnología ARM

TABLA 5.13 MENSAJE DE LA TRAMA DE CONFIGURACIÓN.

Campo	Tamaño [bytes]	Descripción	Valor Hexadecimal	
SYNC	2	Tipo de Mensaje (31-Configuración 2)	AA 31	
FRAME SIZE	2	Tamaño del Mensaje	00 B6	
IDCODE	2	Nombre de la PMU	00 01	
SOC	4	Segundos de Centenario	56 9F 80 97	
FRACSEC	4	Fracción de Segundo con Calidad de Tiempo	00 00 00 00	
TIME_BASE	4	Resolución del Estampado de tiempo	00 0F 42 40	
NUM_PMU	2	Nombre/número del PMU	00 01	
STN	16	Nombre de la Estación (EstacionSEPI2016)	45 73 74 61 63 69 6F 6E 53 45 50 49 32 30 31 36	
IDCODE	2	Nombre/número del PMU	00 01	
FORMAT	2	Formato de datos Bit 0: 0-Rectangular, 1-Polar	00 01	
PHNMR	2	Número de Fasores	00 06	
ANNMR	2	Número de Valores Analógicos	00 00	
DGNMR	2	Número de Valores Digitales	00 00	
CHNAM	96	Nombre de los Fasores y Canales	V Fase A	56 20 46 61 73 65 20 41 20 20 20 20 20 20 20 20
			V Fase B	56 20 46 61 73 65 20 42 20 20 20 20 20 20 20 20
			V Fase C	56 20 46 61 73 65 20 43 20 20 20 20 20 20 20 20
			I Fase A	49 20 46 61 73 65 20 41 20 20 20 20 20 20 20 20
			I Fase B	49 20 46 61 73 65 20 42 20 20 20 20 20 20 20 20
			I Fase C	49 20 46 61 73 65 20 43 20 20 20 20 20 20 20 20
PHUNIT	24	Factor de Conversión para los canales de Fasores	V Fase A	00 0D F8 47
			V Fase B	00 0D F8 47
			V Fase C	00 0D F8 47
			I Fase A	01 00 B2 D0
			I Fase B	01 00 B2 D0
ANUNIT	4	Factor de Conversión de los Canales Analógicos	00 00 00 00	
DIGUNIT	4	Máscara para la palabra digital	00 00 00 00	
FNOM	2	Frecuencia Nominal Bit 0: 0-60 Hz	00 00	
CFGCNT	2	Contador del cambio de configuración	00 00	
DATA_RATE	2	Taza de transmisión de los datos	00 01	
CHK	2	Código de Comprobación de Redundancia Cíclica	2D B0	

En el mensaje de configuración mostrado en la tabla 5.13, se determina el nombre de la estación a la que pertenece la PMU, los nombres de los fasores medidos, sus respectivos factores de conversión para poder ser interpretados según el nivel de voltaje y corriente donde se instale la PMU y se especifica el formato en que se mostrarán los fasores, entre otros valores.

TABLA 5.14 MENSAJE DE LA TRAMA DE CABEZAL.

Campo	Tamaño [bytes]	Descripción	Valor Hexadecimal
SYNC	2	Tipo de Mensaje (11-Cabecal)	AA 11
FRAME SIZE	2	Tamaño del Mensaje	00 14
IDCODE	2	Nombre de la PMU	00 01
SOC	4	Segundos de Centenario	56 9F 82 25
FRACSEC	4	Fracción de Segundo con Calidad de Tiempo	00 00 00 00
DATA 1	1	Caracter ASCII 'S'	53
DATA 2	1	Caracter ASCII 'E'	45
DATA 3	1	Caracter ASCII 'P'	50
DATA 4	1	Caracter ASCII 'I'	49
CHK	2	Código de Comprobación de Redundancia Cíclica	5B 13

La tabla 5.14, como su nombre lo indica, es el cabecal de la PMU, donde se pueden ver caracteres ASCII para que el usuario pueda identificar la PMU. Se pueden incluir el número de DATA necesarios para lograr la identificación o indicar el mensaje que se requiera que el usuario pueda leer.

TABLA 5.15 MENSAJE DE LA TRAMA DE COMANDO.

Campo	Tamaño [bytes]	Descripción	Valor Hexadecimal
SYNC	2	Tipo de Mensaje (41-Comando)	AA 41
FRAME SIZE	2	Tamaño del Mensaje	00 12
IDCODE	2	Nombre de la PMU	00 01
SOC	4	Segundos de Centenario	56 9F 81 A9
FRACSEC	4	Fracción de Segundo con Calidad de Tiempo	00 00 00 00
CMD	2	Comando enviado a la PMU	00 02
CHK	2	Código de Comprobación de Redundancia Cíclica	10 AA

El campo CMD de la trama de comando, tabla 5.15, indica el mensaje que la PMU debe enviar por su puerto serial. Este mensaje es enviado del maestro a la PMU.

Diseño de una Unidad de Medición Fasorial con tecnología ARM

TABLA 5.16 MENSAJE DE LA TRAMA DE DATOS.

Campo	Tamaño [bytes]	Descripción	Valor Hexadecimal	
SYNC	2	Tipo de Mensaje (01-Datos)	AA 01	
FRAMESIZE	2	Tamaño del Mensaje	00 32	
IDCODE	2	Nombre de la PMU	00 01	
SOC	4	Segundos de Centenario	56 9F 80 22	
FRACSEC	4	Fracción de Segundo con Calidad de Tiempo	00 00 00 00	
STAT	2	Estatus del PMU	00 00	
PHASORS	24	Valores de los fasores 16 bits. Dos bytes (Real o Magnitud) y dos bytes (Imaginario o Ángulo)	V Fase A	6A 15 13 3A
			V Fase B	69 E5 0B 03
			V Fase C	68 54 02 EF
			I Fase A	40 E1 15 42
			I Fase B	41 2A 0D 30
			I Fase C	41 8C 04 F8
FREQ	2	Desviación de la frecuencia respecto a F_{nom} (mHz)	00 00	
DFREQ	2	ROCOF	00 00	
ANALOG	2	Valores Analógicos	00 00	
DIGITAL	2	Valores Digitales	00 00	
CHK	2	Código de Comprobación de Redundancia Cíclica	1A A3	

Finalmente, en el mensaje de datos de la tabla 5.16, se incluyen los valores de los fasores, ya sea en forma polar (ángulos en radianes) o rectangular, según se indique en su configuración en el campo FORMAT. Se observa que la estructura del mensaje mostrado en la figura 5.14 es igual a la mostrada en la tabla 5.16, por lo que se infiere que el mensaje enviado cumple con el Estándar.

CAPÍTULO 6 CONCLUSIONES Y RECOMENDACIONES

En el siguiente capítulo, se muestran las conclusiones y recomendaciones en base a los resultados obtenidos al realizar este trabajo, a fin de dar a conocer las consideraciones elementales para trabajos futuros.

6.1 CONCLUSIONES

Tras las pruebas realizadas, se determina que el cálculo de fasores mediante la utilización del algoritmo de la DFT entrega valores exactos, ya que éstos se encuentran muy cercanos al valor real de la señal de entrada. Así, se exhorta el uso de este algoritmo en la implementación de los medidores fasoriales, demostrando que elimina ruido, componentes armónicos y tarda un ciclo en actualizar las mediciones.

La PMU implementada permite calcular la frecuencia del sistema por el método de cruce por cero con un alto grado de precisión y exactitud, teniendo error máximo respecto a la media calculada de 0.037% y desviación estándar con valor de 0.02774, lo que indica que la dispersión de los valores calculados respecto a la media es muy baja. En el caso de la determinación de los valores RMS de las señales bajo análisis, se tienen valores máximos de 0.0437% y 4.758 para los valores de error relativo y desviación estándar, respectivamente, lo que indica buenos valores de exactitud y precisión.

El uso de la DFT por correlación permitió la obtención de fasores de voltaje y corriente del sistema trifásico emulado, actuando como filtro para eliminar cualquier tipo de ruido ocasionado por la presencia de armónicos, para la frecuencia nominal y valores de frecuencia muy cercanos a la nominal, mostrando que la magnitud del fasor es precisa respecto al valor esperado.

El receptor GPS es un dispositivo que entrega el Tiempo Universal Coordinado (UTC) con precisión de microsegundos, permitiendo sincronizar eventos en lugares distantes, por lo que se comprueba que es un elemento preciso para la sincronía de fasores.

Las pruebas del vector de error total no rebasan los límites de precisión en magnitud y ángulo de fase establecidos por el Estándar IEEE C37.118.1 para condiciones de estado estacionario, cuyo valor máximo de TVE es del 1%, mientras que los valores obtenidos por esta PMU se encuentran debajo del 0.2%. Así también, en base al Estándar IEEE C37.118.2, la PMU se comunica de acuerdo a las tramas establecidas, reportando los valores cada segundo basados en el pulso entregado por el módulo GPS.

La utilización de convertidores analógicos de 16 y 24 bits, permitió obtener una mejor precisión y exactitud comparado con el trabajo en [16] y en [18], en los cuales se tienen convertidores de 12 bits. Además, en [18] se logra un TVE máximo de 0.146% para cambios en magnitud de voltaje, mientras que en este trabajo se tiene un valor 0.067%, por lo que se puede concluir que la precisión de esta PMU, conforme a lo establecido por la norma, se mejoró.

El generador de señales de la tarjeta (K20) brinda resultados aceptables en cuanto a la emulación de señales trifásicas de voltaje y corriente, pues se puede modificar, mediante el uso de FreeMASTER, los parámetros de las señales, tales como frecuencia del sistema, magnitud de las señales, ángulo de fase, contenido armónico, magnitud y ángulo del armónico. Una de las desventajas recae en que sólo se tienen los armónicos 3,5, y 7, por lo que no se pueden hacer pruebas con otros armónicos.

FreeMASTER es una herramienta adecuada para mostrar de manera gráfica los valores de las variables en tiempo real, observando las 6 señales al mismo tiempo y sus respectivos fasores calculados. Además, permite visualizar las señales obtenidas del microcontrolador K20 generador de señales y modificar sus parámetros.

6.2 APORTACIONES

- ◇ Proporcionar una Unidad de Medición Fasorial con tecnología ARM, sincronizada vía satélite, que sirva para implementar otros sistemas de medición y aplicarlos en control, estimación de estado, monitoreo o predicción de estabilidad de una red interconectada.
- ◇ Proporcionar diagramas de flujo, plantillas y algoritmos en lenguaje C, en el entorno de CodeWarrior y que puedan ser modificados y/o adecuados a una aplicación en específico.
- ◇ Proporcionar las pruebas realizadas al microcontrolador generador de señales, que determinan su uso como un emulador de la red real trifásica de corriente y voltaje.

6.3 RECOMENDACIONES PARA TRABAJOS FUTUROS

Entre las recomendaciones que se consideran para trabajos en un futuro se encuentran las siguientes:

- ◇ Mostrar los datos de forma local, mediante una pantalla de cristal líquido (la tarjeta cuenta con un display de este tipo).
- ◇ Realizar la comunicación de la unidad de medición fasorial a través de Ethernet.
- ◇ Realizar pruebas con una fuente patrón, adecuando las señales a las entradas del microcontrolador, y comparar los resultados obtenidos con el generador de señales.
- ◇ Realizar diferentes aplicaciones con el sistema desarrollado.
- ◇ Utilizar el medidor fasorial para medición en campo, en la red eléctrica nacional y utilizar las mediciones para determinar flujos de potencia y estabilidad del sistema.
- ◇ Desarrollar las pruebas en estado dinámico, de acuerdo a lo establecido en el Estándar IEEE C37.118.1.
- ◇ Realizar el sistema de medición completo en una placa, a fin de reducir espacio y tener una PMU propia de la SEPI ESIME Zacatenco.
- ◇ Conectar las interfaces usadas en trabajos anteriores [] para probar este trabajo.
- ◇ Conectar algún medio de comunicación, por ejemplo LORA [], para probar una red de PMU's.

BIBLIOGRAFÍA

- [1] Meliopoulos, S., Zhang, F., Zelingher, S. “*Hardware and Software Requirements for a Transmission System Harmonic Measurement System*”.
<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=559103>
- [2] Borjas, R., Quispe, N. “*Instrumentación virtual para Sistema de Medición y Análisis de la Red Eléctrica*”. Revista de la Facultad de Ingeniería, vol. 6. Universidad de Tarapacá Arica, Chile. 1999.
<http://www.redalyc.org/html/114/11400608/>
- [3] Kabra, S. Thesis “*A Technique for Estimating System-Wide Phasors in Real Time*”. Virginia Polytechnic Institute and State University. Master of Science in Electrical Engineering.
- [4] Thakur, S. S., Sinha, A. K. “*A Robust Dynamic State Estimator for Electric Power Systems*”. Technical Journals of Electric Engineering, vol 84, No. 8. Indian Institution of Engineering, Kolkata. 2003.
- [5] Zivanovic, R., Cairns, C. “*Implementation of PMU Technology in State Estimation: An Overview*”. IEEE Africon 4th. 1996.
- [6] Burnett, R. O., Butts, M. M., Cease, T. W., Centeno, V., Michel, G., Murphy, R. J., Phadke, A. G. “*Synchronized phasor measurements of a power system event*”. Power Systems, IEEE. 1994.
- [7] Phadke, A. G., Thorp, J. S., Adamiak, M. G. “*A new measurement technique for tracking voltage phasors, local system frequency, and rate of change frequency*”. IEEE Transactions on Power Apparatus and Systems. 1983.
- [8] Phadke, A. G. “*Synchronized phasor measurements in power systems*”. Computer Applications in Power, IEEE. 1993.
- [9] Phadke, A. G. “*Synchronized sampling and phasor measurements for relaying and control*”. Computer Applications in Power, IEEE. 1993.
- [10] Hart, D., Uy, D., Gharpure, V., Novosel, D., Karlsson, D. “*Unidades PMU Supervisión de las redes eléctricas: un nuevo enfoque*”. Revista ABB 1/2001.
- [11] Quintanilla Adánez, R. “*Sistemas de transporte inteligentes. Medición sincronizada de fasores*”. La Red Eléctrica del Futuro y la Generación Distribuida. Bilbao. IEEC, 2005.
- [12] Wilson, R. “*PMU's Satellite synchronized measurements confirms power equation*”. IEEE potentials. 1994.
- [13] Martin, K., Hamai, D. “*IEEE Standard for Synchrophasors for Power Systems*”. IEEE Power Engineering Society. 2006.
- [14] San Vicente Cisneros, A. R., Cortés Mateos, R. A., Robles García, J., Libreros, D. “*Construcción de un sincrofasor de bajo costo basado en un DSP-Controlador*”. Third LACCEI International Latin American and Caribbean Conference for Engineering and Technology (LACCET'2005). Cartagena de Indias, Colombia. 2005.
- [15] San Vicente Cisneros, A. R., Cortés Mateos, R. A., Robles García, J., Chong Quero, J. E. “*DSP-microcontroller implementations of a simplified algorithm for synchrophasor calculation*”. 2nd International Conference on Electrical and Electronics Engineering. 2005.

- [16] Cuvas Castillo, C. Tesis de Maestría “*Implementación de un Medidor Fasorial*”. D.F., México, IPN-ESIME-SEPI. 2006.
- [17] Valdosierra Marroquin, A. Tesis de Maestría “*Diseño de medidor inteligente e implementación de sistema de comunicación bidireccional*”. D.F., México, IPN-ESIME-SEPI. 2013.
- [18] Hernández Gómez, B. V. Tesis de Maestría “*Diseño e implementación de un medidor fasorial síncrono normalizado con el estándar IEEE C37.118.*”. D.F., México, IPN-ESIME-SEPI. 2009.
- [19] Sebastián Cárdenas, D. Tesis de Maestría “*Development of a smart metering unit designed to identify energy theft on distribution networks*”. D.F., México, IPN-ESIME-SEPI. 2015.
- [20] Yang, W., Chang, T., Song, I., Cho, Y., Heo, J., Jeon, W., Lee, J., Kim, J. “*Signals and Systems with MATLAB*”. Springer. USA. 2009.
- [21] Oppenheim, A. “*Señales y Sistemas*”. Prentice Hall. México.
- [22] Tocci, R., Widmer, N., Moss, G. “*Sistemas Digitales. Principios y Aplicaciones*”. Prentice Hall. México,
- [23] Schweitzer Engineering Laboratories, Inc. Sincrofasores.
<http://www.selinc.com.mx/sincrofasores.php>
- [24] Schweitzer, E. “*Ten Synchronphasor application Tips*”. Schweitzer Engineering Laboratories, Inc. 2006.
- [25] Stanley, W. “*Digital Signal Processing*”. Rader and Gold.
- [26] Smith, S. “*The Scientist and Engineer’s Guide to Digital Signal Processing*”. Second Edition. 1999.
- [27] Oppenheim, A. “*Discrete-Time Signal Processing*”. Prentice Hall, Inc. New Jersey. 1999.
- [28] Spencer, A. “*Matemáticas para Ingeniería. Volumen I*”. CECSA. México, DF.
- [29] Herring, T. A. “*The Global Positioning System*”. Scientific American, 1996.
- [30] Fernández, J., Seco, G. “*Sistemas de posicionamiento: de GPS a GNSS*”. Mundo Electrónico. 1997.
- [31] Rey, J. “*El Sistema de Posicionamiento Global – GPS*”. Universidad de Florida. ENY-728S. Julio, 2010.
- [32] Pozo-Ruz, A., Ribeiro, A., García, M., García, L., Guinea, D., Sandoval, F. “*Sistema de posicionamiento Global (GPS): Descripción, Análisis de Errores, Aplicaciones y Futuro*”. Instituto de Automática Industrial. Madrid. <http://www.peoplesmatters.com/Archivos/Descargas/GPS.pdf>. Fecha de Consulta: 5/08/16.
- [33] http://www.colorado.edu/geography/gcraft/notes/gps/gps_f.html. Fecha de Consulta: 29/02/16.
- [34] <http://www8.garmin.com/aboutGPS/>. Fecha de Consulta: 7/03/16.
- [35] Cooper, W., Helfrick, A. “*Instrumentación electrónica moderna y técnicas de medición*”. Prentice Hall. Primera Edición. 1991.
- [36] http://enciclopedia.us.es/index.php/Tiempo_Universal_Coordinado. Fecha de Consulta: 31/03/16.
- [37] Freescale Semiconductor Inc. “*TWR-KM34Z50M/TWR-KM34Z50MV3 Tower Module, User’s Guide*”. Document Number: TWRKM34Z50UM. Rev. 2, 12/2014.
- [38] http://www.sites.upiicsa.ipn.mx/polilibros/portal/polilibros/p_terminados/SisDig2Ses/UMD/Unidad%20III/Informacion/3312.htm. Fecha de Consulta: 30/11/15.

- [39] Candy, J., Temes, G. "*Oversampling methods for A/D and D/A conversion*". IEEE Press, 1992.
- [40] Aziz, P., Sorensen, H., Van Der Spiegel, J. "*An overview of sigma-delta converters*" IEEE Signal Processing Magazine, Vol.13. 1996.
- [41] Freescale Semiconductor Inc. "*KM Family Reference Manual*". Document Number: MKMxxZzzACzz5RM. Rev. 1, 09/2014.
- [42] Freescale Semiconductor Inc. "*Kinetis-M Three-Phase Power Meter Reference Design*". Document Number: DRM 147. Rev. 0, 08/2014.
- [43] LINX Technologies Inc. "*SG Series GPS Receiver Module Data Guide*". Rev. 6/28/2010.
- [44] PARALLAX Inc. "*RXM-SG GPS Module w/Ext Antenna (#28505)*". V.1.0. Rev. 8/27/2010.
- [45] Valdano, J. "*Excerpts from Introduction to Embedded Microcomputer Systems*". 2001.
- [46] Eastern Interconnect Phasor Project (EIPP) Tools. TVA PMU Connection Tester.
- [47] Phadke, A.G., Thorp, J. S. "*Computer Relaying for Power Systems*". Editorial Wiley. Segunda Edición. Inglaterra. 2009.
- [48] IEEE Standards Association. "*IEEE Standard for Synchrophasor Measurement for Power Systems*". IEEE Std. C37.118.1. 2011.
- [49] IEEE Standards Association. "*IEEE Standard for Synchrophasor Data Transfer for Power Systems*". IEEE Std. C37.118.2. 2011.
- [50] Doebelin, E. "*Sistemas de medición e Instrumentación. Diseño y Aplicación.*". Editorial McGraw Hill. Quinta Edición. 2005.
- [51] <http://www.nxp.com/products/power-management/wireless-charging-ics/freemaster-runtime-debugging-tool:FREEMASTER>. Fecha de Consulta: 16/11/2016.

APÉNDICE A. CÓDIGO IMPLEMENTADO

A.1 PROGRAMA PRINCIPAL (PMU_GPS.C)

```

#include <string.h>
#include <stdio.h>
#include <math.h>
#include "drivers.h"
#include "appconfig.h"
#include "freemaster.h"
#include "xbar_adc_coco.h"
#include "mediciones.h"
#include "tramas.h"

/*****
 *
 *                               Programa Principal
 *
 *****/
void main (void)
{
    /* Inicialización de los Modulos de la Tarjeta */
    SIM_Init (SIM_MODULE_ALL_PERIPH_ON_CONFIG);
    SIM_EnableModule (UART1);
    SIM_EnableModule (PORT1);
    SIM_EnableModule (XBAR);

    /* Enrutamiento del Reloj del Sistema al PTF7 */
    SIM_SelClkout (CLKOUT_SRC1);
    PORT_Init (PORTF, PORT_MODULE_ALT3_MODE, PIN7);
    /* Modo de Reloj 2:1:1, 48MHz */
    SIM_SetClkMode (SYSCLK_MODE1);
    SIM_SetClkDiv (SYSCLK_DIV1);
    FLL_Init (FLL_MODULE_FEE_48MHZ_CONFIG);

    /* Fijar el Oscilador RTC de 32 KHz RTC como fuente del PLL y habilitar el PLL*/
    PLL_Enable (PLL32KREF_SRC1);

    /* Módulo VREF debe ser inicializado después del Módulo SIM */
    VREF_Init (VREF_MODULE_CHOP_EN_ICOMP_EN_REG_EN_VREFL_EN_LPWR_CONFIG,
              VREF_SWITCH_S1_L_S2_L_S3_L_BUFF_EN);
    VREFL_Trim(0); /* VREFL trimmed to 0.400 V */
    /* Inicialización de los Canales del SAR (ADC de Aprox. Suc.) */
    ADC_Init (ADC_MODULE_16B_HWTRG_XREF_CONFIG, /* Disparo por HW */
             HWAVG_OFF, /* Factor Promedio de HW */
             ADC_CH_SE_IRQ_CONFIG(AD0), /* Configuración del Canal A */
             ADC_CH_SE_IRQ_CONFIG(AD1), /* Configuración del Canal B */
             ADC_CH_SE_IRQ_CONFIG(AD2), /* Configuración del Canal C */
             ADC_CH_DISABLE_CONFIG, /* Configuración del Canal D */
             PRI_LVL0, /* Prioridad */
             (ADC_CALLBACK) Callback_ADC); /* Función de Retorno del ADC */

    /* Inicialización de Canales AFE (ADC Sigma-Delta */
    SIM_SelAfePl1Clk (SIM_MCG_PLL_CLK);

```

Diseño de una Unidad de Medición Fasorial con tecnología ARM

```

AFE_ChInit (CH0, /* Canal 0 */
            AFE_CH_SWTRG_CCM_PGAOFF_CONFIG(DEC_OSR2048), /* Decimador */
            (ASD_L1M * AFE_SCT), /* Retardo de Fase del Canal 0 */
            PRI_LVL0 /* Prioridad */
            NULL); /* Función de Retorno del Canal 0 */
AFE_ChInit (CH1, /* Canal 1 */
            AFE_CH_SWTRG_CCM_PGAOFF_CONFIG(DEC_OSR2048), /* Decimador */
            (ASD_L2M * AFE_SCT), /* Retardo de Fase del Canal 1 */
            PRI_LVL0, /* Prioridad */
            NULL); /* Función de Retorno del Canal 1 */
AFE_ChInit (CH2 /* Canal 2 */
            AFE_CH_SWTRG_CCM_PGAOFF_CONFIG(DEC_OSR2048), /* Decimador */
            (ASD_L3M * AFE_SCT), /* Retardo de Fase del Canal 2 */
            PRI_LVL0, /* Prioridad */
            NULL); /* Función de Retorno del Canal 2 */

/* Inicialización del Módulo Completo de Canales AFE (ADC Sigma-Delta) */
AFE_Init (AFE_MODULE_LPM_RJFORMAT_CONFIG(AFE_PLL_CLK, AFE_DIV1, AFE_CLK));

/* Disparo de las conversiones de los ADC Sigma-Delta (AFE) */
AFE_SwTrigger (CH0|CH1|CH2|CH3);

/* Inicialización del Módulo XBAR */
XBAR_Init (XBAR_MODULE_NO_EDGE_DETECT_CONFIG, PRI_LVL0, NULL);

/* Conexión del disparo del ADC Sigma-Delta (AFE) al ADC de Aprox. Suc. (SAR) */
XBAR_Path (XBAR_AFE0COC, XBAR_ADCTRGCHA); /* Canal 0 (AFE) al Canal A (SAR) */
XBAR_Path (XBAR_AFE1COC, XBAR_ADCTRGCHB); /* Canal 1 (AFE) al Canal B (SAR) */
XBAR_Path (XBAR_AFE2COC, XBAR_ADCTRGCHC); /* Canal 2 (AFE) al Canal C (SAR) */

/* Inicialización del UART y FreeMASTER */
PORT_Init (PORTI, PORT_MODULE_ALT2_MODE, PIN0|PIN1);
UART_Init (UART1, UART_MODULE_INTRMODE_CONFIG(38400, 48e6));
UART_InstallCallback (UART0_UART1, PRI_LVL0, (UART_CALLBACK)FMSTR_Isr);
FMSTR_Init(); /*

/* Inicialización del UART para el Reporte de Tramas */
PORT_Init (PORTI, PORT_MODULE_ALT2_MODE, PIN0|PIN1);
UART_Init (UART1, UART_MODULE_INTRMODE_CONFIG(9600, 48e6));
UART_InstallCallback (UART0_UART1, PRI_LVL1, (UART_CALLBACK)Callback_UART);

/* Leds de Diagnóstico */
PORT_Init (PORTF, PORT_MODULE_ALT1_MODE, PIN1);
GPIO_Init (GPIOF, GPIO_OUT_LOGIC1_MODE, PIN1); /* LED Rojo */
PORT_Init (PORTE, PORT_MODULE_ALT1_MODE, PIN5);
GPIO_Init (GPIOE, GPIO_OUT_LOGIC1_MODE, PIN5); /* LED Verde */
PORT_Init (PORTD, PORT_MODULE_ALT1_MODE, PIN1);
GPIO_Init (GPIOD, GPIO_OUT_LOGIC1_MODE, PIN1); /* LED Naranja */
PORT_Init (PORTC, PORT_MODULE_ALT1_MODE, PIN1);
GPIO_Init (GPIOC, GPIO_OUT_LOGIC1_MODE, PIN1); /* LED Amarillo */

/* Puerto para recibir el pps */
PORT_Init (PORTD, PORT_MODULE_ALT1_ANYEDGE_IRQ_MODE, PIN7);
GPIO_Init (GPIOD, GPIO_INP_MODE, PIN7);
PORT_InstallCallback (PRI_LVL1, Callback_Port);

/* Base para el Cálculo del SOC */
base = año + mes + días;

/* Habilitar interrupciones a nivel global */
EnableInterrupts();

```

while(1)

Diseño de una Unidad de Medición Fasorial con tecnología ARM

```

{
    //FMSTR_Poll ();          /* Función de manejo de datos para FreeMASTER */
}
}
/*****
*                               Función de Retorno del ADC                               *
*****/
void Callback_ADC (ADC_CALLBACK_TYPE type, register uint16 result)
{
    /* Si hay Conversión Completa del Canal C */
    if (type == CHC_CALLBACK)
    {
        //FMSTR_Recorder();          /* Rutina del Recorder de FreeMASTER */
        /* Lectura Canal A ADC, menos OFFSET */
        adc_cha = MV*((ADC_Read(CHA) - OFFSET_SAR_A)*5.89);
        /* Lectura Canal B ADC, menos OFFSET */
        adc_chb = MV*((ADC_Read(CHB) - OFFSET_SAR_B)*5.89);
        /* Lectura Canal C ADC, menos OFFSET */
        adc_chc = MV*((ADC_Read(CHC) - OFFSET_SAR_C)*5.95);
        /* Lectura Canal 0 AFE, menos OFFSET */
        afe_ch0 = MC*(((AFE_ChRead(CH0) - OFFSET_SD_0)>>8)*1.14);
        /* Lectura Canal 1 AFE, menos OFFSET */
        afe_ch1 = MC*(((AFE_ChRead(CH1) - OFFSET_SD_1)>>8)*1.14);
        /* Lectura Canal 2 AFE, menos OFFSET */
        afe_ch2 = MC*(((AFE_ChRead(CH2) - OFFSET_SD_2)>>8)*1.14);
        buff_frec[cont_frec] = adc_cha;
        /* Asignación del v. actual para calcular frecuencia */
        frec_ptr->v_act = buff_frec[cont_frec];
        if((cont_frec%10) == 0) /* Filtrar cada 10 muestras */
        {
            f_Frec(frec_ptr); /* Llamada a la función que calcula la frecuencia*/
        }
        cont_frec++;
        if(cont_frec==BUFF_SIZE)
            cont_frec=0;
        /* Sincronización de 1 pps obtenido del GPS */
        if(pps)
        {
            buff_cha[cont_buff] = adc_cha; /* Llenado del buffer del canal A */
            buff_chb[cont_buff] = adc_chb; /* Llenado del buffer del canal B */
            buff_chc[cont_buff] = adc_chc; /* Llenado del buffer del canal C */
            buff_ch0[cont_buff] = afe_ch0; /* Llenado del buffer del canal 0 */
            buff_ch1[cont_buff] = afe_ch1; /* Llenado del buffer del canal 1 */
            buff_ch2[cont_buff] = afe_ch2; /* Llenado del buffer del canal 2 */
            cont_buff++; /* Incremento del contador del Buffer */
            if(cont_buff == NM_ACTUAL) /* Primer mitad del Buffer */
            {
                bandera = 1; /* Bandera en 1, primeras 100 muestras adquiridas*/
            }
            if(cont_buff >= BUFF_SIZE) /* Segunda mitad del Buffer */
            {
                bandera = 2; /* Bandera en 2, segundas 100 muestras adquiridas*/
                cont_buff = 0; /* Reinicio del contador del Buffer de Muestras*/
                pps=0; /* Reinicio de la bandera de 1 pps del GPS */
            }
            if (bandera) /* Si existe la Bandera del Buffer de Muestras */
            {
                f_calculos(); /* Llamado a la Función que inicia los cálculos*/
            }
        }
    }
}
}

```

Diseño de una Unidad de Medición Fasorial con tecnología ARM

```

/*****
*
*                               Función de Retorno del UART
*                               *****/
void Callback_UART (UART_CALLBACK_SRC module,  UART_CALLBACK_TYPE type, uint32 status)
{
    if (module == UART1_CALLBACK)                /* UART correspondiente */
    {
        if (type == UART_TX_CALLBACK)           /* Tx del UART */
        {
            if(conf_enviada==0)                 /* Trama de Configuración */
                UART_Wr (UART1,Config_Frame,TAM_CONFIG);
            if(conf_enviada==1)                 /* Trama de Datos */
                UART_Wr (UART1,Data_Frame,TAM_DATA);
            if(conf_enviada==2)                 /* Trama de Cabecal */
                UART_Wr (UART1,Header_Frame,TAM_HDR);
            if(conf_enviada==3)                 /* Trama de Comando */
                UART_Wr (UART1,Comm_Frame,TAM_COMMAND);
            UART_TxIsrDisable (UART1);          /* Deshabilita el Tx */
        }
        if (type == UART_RX_CALLBACK)           /* Rx del UART */
        {
            rx = UART_GetChar (UART1);          /* Lee el caracter de la trama NMEA */
            if(rx=='$')  c[0]=rx;
            if(rx=='G')  c[1]=rx;
            if(rx=='P')  c[2]=rx;
            if(rx=='G')  c[3]=rx;
            if(rx=='G')  c[4]=rx;
            if(rx=='A')  c[5]=rx;
            if(rx==',')  c[6]=rx;
            /* Detección del tipo de Trama NMEA "$GPGGA" para continuar con la lectura del UTC*/
            if(c[0]=='$' && c[1]=='G' && c[2]=='P' && c[3]=='G' && c[4]=='G' && c[5]=='A' && c[6]==',')
            {
                UART_Rd (UART1,tiempo,13);      /* Lectura del UTC */
                c[0]=c[1]=c[2]=c[3]=c[4]=c[5]=c[6]='_'; /* Reinicio de variables */
                UART_RxIsrDisable (UART1);      /* Deshabilita el Rx */
            }
        }
    }
}

/*****
*
*                               Interrupción Externa debida al pps del GPS
*                               *****/
void Callback_Port (PORT_CALLBACK_SRC src, uint8 pin)
{
    if (src == PORTD_CALLBACK)                 /* Puerto correspondiente */
    {
        if (pin == PIN7)                       /* Pin correspondiente */
        {
            command_frame();                    /* Lee la Trama de Comando */
            tx_comm(TAM_COMMAND-2);             /* Calcula el CRC de la Trama de Comando */
            pps=1;                               /* Bandera del PPS del GPS */
            UART_RxIsrEnable (UART1);          /* Habilita el Rx */
            GPIO_Tgl (GPIOC, PIN1);            /* Led de Señalización */
            f_comunicacion();                   /* Llamado a la función de comunicación */
        }
    }
}

```

}

A.2 HEADER DE CÁLCULOS (MEDICIONES.H)

```

* mediciones.h

#ifndef MEDICIONES_H_
#define MEDICIONES_H_

#include <math.h>

/*****
*
*                               Definición de Constantes
*
*****/

#define F_NOM                60                /* Frecuencia Nominal del Sistema */
#define NM_ACTUAL            100              /* Número de muestras */
#define BUFF_SIZE            (NM_ACTUAL*2)    /* Tamaño del Buffer de Muestras */
#define ORDEN                10              /* Orden del arreglo para frecuencia */
#define AJUSTE_FREQ          60*(NM_ACTUAL/ORDEN) /* FrecN.*N.Mue Alg Frec */
#define OFFSET_SAR_A        4677            /* OFFSET para la señal de entrada VA */
#define OFFSET_SAR_B        4681            /* OFFSET para la señal de entrada VB */
#define OFFSET_SAR_C        5300            /* OFFSET para la señal de entrada VC */
#define OFFSET_SD_0         3685454        /* OFFSET para la señal de entrada IA */
#define OFFSET_SD_1         3764099        /* OFFSET para la señal de entrada IB */
#define OFFSET_SD_2         3772957        /* OFFSET para la señal de entrada IC */
#define MV                   1              /* x100 Porcentaje de la Magnitud de Entrada*/
#define MC                   1              /* x100 Porcentaje de la Magnitud de Entrada*/
#define NUM_ESCLAVO          1              /* Número de Esclavo correspondiente */
#define TAM_DATA             50             /* Tamaño de la Trama de Datos */
#define TAM_CONFIG           182           /* Tamaño de la Trama de Configuración */
#define TAM_HDR              20            /* Tamaño de la Trama de Cabecal */
#define TAM_COMMAND          18            /* Tamaño de la Trama de Comando */

/* Numeración correspondiente al formato de los fasores */
enum {RECTANGULAR, POLAR};

/* Numeración correspondiente al comando (CMD) de la Trama de Comando */
enum {DATA_OFF=1, DATA_ON=2, SEND_HDR=3, SEND_CFG1=4, SEND_CFG2=5, SEND_CFG3=6,
EXT_FRAME=8, SEND_COMM=7};

/*****
*
*                               Definición de Estructuras
*
*****/
/* Definición de la Estructura para los valores de la DFT */
typedef struct
{
    int32_t    DFT_Re;                /* Parte Real del Fasor */
    int32_t    DFT_Im;                /* Parte Imaginaria del Fasor */
    int32_t    DFT_Mag;               /* Magnitud del Fasor */
    int32_t    DFT_Ph;                /* Ángulo Escalado del Fasor */
    int32_t    DFT_Rad;               /* Ángulo en Radianes del Fasor */
    float      DFT_Ang;               /* Ángulo del Fasor */
} DFT_STRUCT;

/* Definición de la Estructura para los valores de frecuencia */
typedef struct frec_struct
{
    float frecuencia;                /* Frecuencia del Sistema */

```

Diseño de una Unidad de Medición Fasorial con tecnología ARM

```

    int32_t v_ant; /* Valor Anterior */
    int32_t v_act; /* Valor Actual */
    float suma_muestras; /* Suma de Muestras */
    float frec_mediana; /* Frecuencia Mediana */
} FREQ_STRUCT, *FREQ_STRUCT_PTR;

/*****
*
*                               Definición de Variables                               *
*****/
/* Arreglo de los datos obtenidos por el algoritmo de Frecuencia */
float datos[ORDEN];
/* Bandera auxiliar para el cálculo de la Frecuencia */
uint32_t flag_frec=0;
/* Variable de la estructura de las Frecuencia */
FREQ_STRUCT frec;
/* Apuntadores a la Estructura de la Frecuencia */
FREQ_STRUCT_PTR frec_ptr = &frec;
/* Valores de los Canales de los ADC */
int32_t adc_cha, adc_chb, adc_chc;
int32_t afe_ch0, afe_ch1, afe_ch2;
/* Buffers de 200 valores */
int32_t buff_cha[BUFF_SIZE], buff_chb[BUFF_SIZE], buff_chc[BUFF_SIZE];
int32_t buff_ch0[BUFF_SIZE], buff_ch1[BUFF_SIZE], buff_ch2[BUFF_SIZE];
int32_t buff_frec[BUFF_SIZE];
/* Contador para el buffer */
int cont_buff=0;
/* Contador para frecuencia */
int cont_frec=0;
/* Bandera de reconocimiento de ciclo */
uint8_t bandera = 0;
/* Valores RMS para cada buffer */
uint32_t Valor_Rms_Va, Valor_Rms_Vb, Valor_Rms_Vc;
uint32_t Valor_Rms_Ia, Valor_Rms_Ib, Valor_Rms_Ic;
/* Estructuras para la DFT */
DFT_STRUCT DFT_Buffer_Va, DFT_Buffer_Vb, DFT_Buffer_Vc;
DFT_STRUCT DFT_Buffer_Ia, DFT_Buffer_Ib, DFT_Buffer_Ic;
/* Pulso Por Segundo */
uint8_t pps = 0;
/* Tiempo del Timer Rx */
uint16_t timer_rx;
/* Formato (Polar - 1 o Rectangular -0) */
uint8_t formato = POLAR;
/* Buffers para la Transmisión de Datos */
char Data_Frame[50]; /* Buffer para el Mensaje de Datos */
char Config_Frame[182]; /* Buffer para el Mensaje de Configuración */
char Header_Frame[20]; /* Buffer para el Mensaje de Cabezal */
char Comm_Frame[18]; /* Buffer para el Mensaje de Comando */
char Buffer_Fasores[24]; /* Buffer que guarda los valores de los fasores */
/* Caracter recibido por el Rx */
volatile char rx;
/* Arreglo para guardar la cadena $GPGGA de la Trama NMEA */
char c[7];
/* Arreglo para guardar el UTC */
char tiempo[13];
/* Variable para guardar los segundos del SOC */
uint8_t sec=0;
/* Variable para guardar los minutos del SOC */
uint16_t min=0;
/* Variable para guardar las horas del SOC */
uint32_t horas=0;
/* Segundo de Centenario (Second of Century) */
extern uint32_t SOC;
/* Estatus de la Configuración */

```

Diseño de una Unidad de Medición Fasorial con tecnología ARM

```

uint8_t conf_enviada=0;
/* Tamaño de la Trama de Datos a enviar sin el CRC */
uint16_t size_env_data_on;
/* Número de bytes a enviar */
uint16_t num_bytes_enviar;

/* Base para el cálculo de SOC */
uint32_t base;
/* Segundos durante los años transcurridos desde 00:00:00 01/01/1970 hasta 00:00:00
01/01/2016 */
uint32_t anio=1451606400;
/* Segundos durante los meses transcurridos desde 00:00:00 01/01/2016 hasta 00:00:00
01/10/2016 */
uint32_t mes=777600;
/* Segundos durante los días transcurridos desde 00:00:00 01/10/2016 hasta 00:00:00
20/10/2016 */
uint32_t dias=86400*20;

/*****
*                               Definición de Funciones                               *
*****/
/* Función que calcula el RMS */
uint32_t f_Rms(int32_t *,uint8_t,uint8_t);
/* Función que calcula la Raíz */
uint32_t f_Raiz(uint32_t int_val);
/* Función que calcula la Frecuencia de la Señal */
uint32_t f_Frec(FREC_STRUCT_PTR);
/* Función del Filtro Mediana */
float Filtro_Mediana(float *datos,uint32_t orden);
/* Función que ordena los valores de forma Ascendente */
uint32_t ordena(float *datos,uint32_t orden);
/* Función que calcula la DFT */
void f_dft(int32_t *Mediciones,uint8_t,uint8_t,DFT_STRUCT *ptr_dft);
/* Función que inicia los cálculos */
void f_calculos(void);
/* Declaración de la función de retorno del ADC de Aprox. Suc. */
void Callback_ADC (ADC_CALLBACK_TYPE type, register uint16_t result);
/* Declaración de la función de retorno del Pulso Por Segundo */
void Callback_Port (PORT_CALLBACK_SRC src, uint8_t pin);
/* Preparación de Valores para la Transmisión de Datos */
void preparacion_transmision(DFT_STRUCT *ptr_dft);
/* Declaración de la función de retorno del UART */
void Callback_UART (UART_CALLBACK_SRC module, UART_CALLBACK_TYPE type, uint32_t status);
/* Llenado de la Trama de Datos */
void data_frame(void);
/* Llenado de la Trama de Configuración */
void config_frame(void);
/* Llenado de la Trama de Cabezal */
void header_frame(void);
/* Llenado de la Trama de Comando */
void command_frame(void);
/* Función que calcula el SOC con el UTC */
uint32_t f_SOC(void);
/* Función que calcula el CRC */
uint16_t f_CRC(unsigned char *Mensaje, uint16_t Tam_Men);
/* Función que inicia la comunicación */
void f_comunicacion(void);
/* Función que inicia la Transmisión de la Configuración */
void tx_conf(uint16_t);
/* Función que inicia la Transmisión de la Datos */
void tx_datos(uint16_t);
/* Función que inicia la Transmisión de la Cabezal */
void tx_head(uint16_t);

```

Diseño de una Unidad de Medición Fasorial con tecnología ARM

```
/* Función que inicia la Transmisión de la Comando */
void tx_comm(uint16_t);

/*****
* Definición de las Funciones de Referencia para el Cálculo de la DFT */
*****/

/* Coseno de Referencia */
int16_t cos_ref[NM_ACTUAL] = { 32767, 32701, 32508, 32186, 31737, 31162, 30465, 29647,
28713, 27665, 26508, 25246, 23885, 22429, 20885, 19259, 17556, 15784, 13950, 12061,
10124, 8148, 6140, 4106, 2057, 0, -2057, -4106, -6140, -8148, -10124, -12061, -13950, -
15784, -17556, -19259, -20885, -22429, -23885, -25246, -26508, -27665, -28713, -29647,
-30465, -31162, -31737, -32186, -32508, -32701, -32767, -32701, -32508, -32186, -31737,
-31162, -30465, -29647, -28713, -27665, -26508, -25246, -23885, -22429, -20885, -19259,
-17556, -15784, -13950, -12061, -10124, -8148, -6140, -4106, -2057, 0, 2057, 4106, 6140,
8148, 10124, 12061, 13950, 15784, 17556, 19259, 20885, 22429, 23885, 25246, 26508, 27665,
28713, 29647, 30465, 31162, 31737, 32186, 32508, 32701};

/* Seno de Referencia */
int16_t sen_ref[NM_ACTUAL] = {0, 2057, 4106, 6140, 8148, 10124, 12061, 13950, 15784,
17556, 19259, 20885, 22429, 23885, 25246, 26508, 27665, 28713, 29647, 30465, 31162,
31737, 32186, 32508, 32701, 32767, 32701, 32508, 32186, 31737, 31162, 30465, 29647,
28713, 27665, 26508, 25246, 23885, 22429, 20885, 19259, 17556, 15784, 13950, 12061,
10124, 8148, 6140, 4106, 2057, 0, -2057, -4106, -6140, -8148, -10124, -12061, -13950, -
15784, -17556, -19259, -20885, -22429, -23885, -25246, -26508, -27665, -28713, -29647,
-30465, -31162, -31737, -32186, -32508, -32701, -32767, -32701, -32508, -32186, -31737,
-31162, -30465, -29647, -28713, -27665, -26508, -25246, -23885, -22429, -20885, -19259,
-17556, -15784, -13950, -12061, -10124, -8148, -6140, -4106, -2057};

/* Coseno Inverso de Referencia para el Cálculo del Ángulo del Fasor */
int16_t arc_cos[256] = {9000, 8977, 8955, 8932, 8910, 8887, 8865, 8842, 8820, 8797, 8775,
8752, 8730, 8707, 8685, 8662, 8640, 8617, 8595, 8572, 8550, 8527, 8505, 8482, 8459, 8437,
8414, 8392, 8369, 8346, 8324, 8301, 8279, 8256, 8233, 8211, 8188, 8165, 8142, 8120, 8097,
8074, 8051, 8029, 8006, 7983, 7960, 7937, 7915, 7892, 7869, 7846, 7823, 7800, 7777, 7754,
7731, 7708, 7685, 7662, 7639, 7615, 7592, 7569, 7546, 7523, 7499, 7476, 7453, 7430, 7406,
7383, 7359, 7336, 7313, 7289, 7266, 7242, 7218, 7195, 7171, 7147, 7124, 7100, 7076, 7052,
7029, 7005, 6981, 6957, 6933, 6909, 6885, 6861, 6836, 6812, 6788, 6764, 6739, 6715, 6691,
6666, 6642, 6617, 6593, 6568, 6543, 6519, 6494, 6469, 6444, 6419, 6394, 6369, 6344, 6319,
6294, 6268, 6243, 6218, 6192, 6167, 6141, 6116, 6090, 6064, 6038, 6012, 5987, 5961, 5934,
5908, 5882, 5856, 5829, 5803, 5776, 5750, 5723, 5696, 5670, 5643, 5616, 5588, 5561, 5534,
5507, 5479, 5452, 5424, 5396, 5368, 5341, 5313, 5284, 5256, 5228, 5199, 5171, 5142, 5113,
5084, 5055, 5026, 4997, 4967, 4938, 4908, 4878, 4849, 4818, 4788, 4758, 4727, 4697, 4666,
4635, 4604, 4573, 4541, 4509, 4478, 4446, 4413, 4381, 4349, 4316, 4283, 4250, 4216, 4183,
4149, 4115, 4081, 4046, 4011, 3976, 3941, 3906, 3870, 3834, 3797, 3761, 3724, 3686, 3649,
3611, 3573, 3534, 3495, 3456, 3416, 3376, 3335, 3294, 3252, 3210, 3168, 3125, 3081, 3037,
2992, 2947, 2901, 2854, 2807, 2759, 2710, 2660, 2609, 2558, 2505, 2452, 2397, 2341, 2284,
2225, 2165, 2103, 2040, 1974, 1907, 1837, 1764, 1689, 1609, 1526, 1438, 1345, 1245, 1136,
1016, 879, 718, 507, 0};

/*****
* Función que inicia los Cálculos */
*****/
void f_calculos(void)
{
    SOC = f_SOC(); /* Llamado a la función que calcula el SOC */
    if (flag_freq) /* Si existe la Bandera Auxiliar de la Frecuencia */

```

Diseño de una Unidad de Medición Fasorial con tecnología ARM

```

{
    frec_ptr->frec_mediana = Filtro_Mediana(datos,ORDEN); /* Frec Mediana */
    flag_frec=0; /* Reinicio de la Bandera Auxiliar de la Frecuencia */
}
if (bandera == 1) /* Detección del Primer Ciclo para el Análisis */
{
    /* Cálculo del RMS de 0 a 99 muestras */
    Valor_Rms_Va = f_Rms(buff_cha, 0, 99);
    Valor_Rms_Vb = f_Rms(buff_chb, 0, 99);
    Valor_Rms_Vc = f_Rms(buff_chc, 0, 99);
    Valor_Rms_Ia = f_Rms(buff_ch0, 0, 99);
    Valor_Rms_Ib = f_Rms(buff_ch1, 0, 99);
    Valor_Rms_Ic = f_Rms(buff_ch2, 0, 99);
    /* Cálculo de la DFT del Primer Ciclo */
    f_dft(buff_cha, 0, 99, &DFT_Buffer_Va);
    f_dft(buff_chb, 0, 99, &DFT_Buffer_Vb);
    f_dft(buff_chc, 0, 99, &DFT_Buffer_Vc);
    f_dft(buff_ch0, 0, 99, &DFT_Buffer_Ia);
    f_dft(buff_ch1, 0, 99, &DFT_Buffer_Ib);
    f_dft(buff_ch2, 0, 99, &DFT_Buffer_Ic);
    /* Preparar los fasores para su transmisión */
    preparacion_transmision(&DFT_Buffer_Va);
    preparacion_transmision(&DFT_Buffer_Vb);
    preparacion_transmision(&DFT_Buffer_Vc);
    preparacion_transmision(&DFT_Buffer_Ia);
    preparacion_transmision(&DFT_Buffer_Ib);
    preparacion_transmision(&DFT_Buffer_Ic);
    data_frame(); /* Llena la Trama de Datos */
    config_frame(); /* Llena la Trama de Configuración */
    header_frame(); /* Llena la Trama de Cabecal */
}
if (bandera == 2) /* Detección del Segundo Ciclo para el Análisis */
{
    /* Cálculo del RMS de 100 a 199 muestras */
    Valor_Rms_Va = f_Rms(buff_cha, 100, 199);
    Valor_Rms_Vb = f_Rms(buff_chb, 100, 199);
    Valor_Rms_Vc = f_Rms(buff_chc, 100, 199);
    Valor_Rms_Ia = f_Rms(buff_ch0, 100, 199);
    Valor_Rms_Ib = f_Rms(buff_ch1, 100, 199);
    Valor_Rms_Ic = f_Rms(buff_ch2, 100, 199);
    /* Cálculo de la DFT del Segundo Ciclo */
    f_dft(buff_cha, 100, 199, &DFT_Buffer_Va);
    f_dft(buff_chb, 100, 199, &DFT_Buffer_Vb);
    f_dft(buff_chc, 100, 199, &DFT_Buffer_Vc);
    f_dft(buff_ch0, 100, 199, &DFT_Buffer_Ia);
    f_dft(buff_ch1, 100, 199, &DFT_Buffer_Ib);
    f_dft(buff_ch2, 100, 199, &DFT_Buffer_Ic);
    /* Preparar los fasores para su transmisión */
    preparacion_transmision(&DFT_Buffer_Va);
    preparacion_transmision(&DFT_Buffer_Vb);
    preparacion_transmision(&DFT_Buffer_Vc);
    preparacion_transmision(&DFT_Buffer_Ia);
    preparacion_transmision(&DFT_Buffer_Ib);
    preparacion_transmision(&DFT_Buffer_Ic);
    data_frame(); /* Llena la Trama de Datos */
    config_frame(); /* Llena la Trama de Configuración */
    header_frame(); /* Llena la Trama de Cabecal */
}
if (conf_enviada==1) /* Si el CMD corresponde a Envío de Datos */
{
    tx_datos(size_env_data_on); /* Transmisión de Trama de Datos */
}
bandera = 0; /* Reinicia Bandera para llenar del Buffer de Muestras */

```

Diseño de una Unidad de Medición Fasorial con tecnología ARM

```
};

/*****
 *                               Función que ordena un arreglo de datos                               *
 *****/
uint32_t ordena(float *datos, uint32_t orden)
{
    uint32_t v1, v2, v3;
    for(v1=1; v1<orden; v1++)
    {
        for(v2=0; v2<orden-v1; v2++)
        {
            if(datos[v2]>datos[v2+1])
            {
                v3=datos[v2+1];
                datos[v2+1]=datos[v2];
                datos[v2]=v3;
            }
        }
    }
};

/*****
 *                               Función que calcula el valor RMS                               *
 *****/
uint32_t f_Rms(int32_t *Mediciones, uint8_t Primer_Valor, uint8_t Ultimo_Valor)
{
    uint8_t i, j=0, k;
    uint32_t suma[10];
    uint32_t prom[10];
    uint32_t rms, aux=0;
    /* Debido a los valores del AFE, se divide el cálculo, para no generar desbordamiento
    de variables */
    for(i = Primer_Valor; i <= Ultimo_Valor; i++)
    {
        /* Acumulación del cuadrado de las muestras del ciclo */
        suma[j] = suma[j] + (Mediciones[i] * Mediciones[i])>>8;
        if(i%10 == 0)
            j++;
    }
    j=0;
    for(k=0; k<=9; k++)
    {
        prom[k]=suma[k]/10;
        aux= aux+prom[k];
    }
    rms = f_Raiz(aux<<8);
    return (rms);
};

/*****
 *                               Función que calcula el Filtro Mediana                               *
 *****/
float Filtro_Mediana(float *datos, uint32_t orden)
{
    float med = 0;
    ordena(datos, orden);
};
```

Diseño de una Unidad de Medición Fasorial con tecnología ARM

```

    med = (datos[orden/2-1] + datos[orden/2])/2;          /* Promedio de los valores */
    return (med);                                       /* Devuelve el valor de la mediana */
};

/*****
 *                               Función que calcula la Raíz Cuadrada                               *
 *****/
uint32_t f_Raiz(uint32_t val)
{
    uint32_t r,t=0;
    r = val;
    if(val > 0)
    {
        do
        {
            t = r;
            r = (val/r + r) / 2;
        }while( t > r );
    }
    return (t);
};

/*****
 *                               Función que calcula la DFT                               *
 *****/
void f_dft(int32_t *Mediciones, uint8_t Primer_Valor, uint8_t Ultimo_Valor, DFT_STRUCT
*ptr_dft)
{
    uint8_t i,j=0;
    int32_t Re_Aux;
    int32_t int_mag;
    int32_t suma_cos=0,suma_sen=0;

    for(i = Primer_Valor ; i <= Ultimo_Valor ; i++)
    {
        /* Acumulación de la mult. de las muestras * la onda Cos de Referencia */
        suma_cos = suma_cos + (Mediciones[i] * ((float)cos_ref[j] / 32768));
        /* Acumulación de la mult. de las muestras * la onda Sen de Referencia */
        suma_sen = suma_sen + (Mediciones[i] * ((float)sen_ref[j] / 32768));
        j++;
    }
    ptr_dft->DFT_Re = (2 * suma_cos / (NM_ACTUAL));          /* Parte Real */
    ptr_dft->DFT_Im = -(2 * suma_sen / (NM_ACTUAL));        /* Parte Imaginaria */

    /* Suma de los Cuadrados de la Parte Real e Imaginaria */
    int_mag = (int32_t) (ptr_dft->DFT_Re * ptr_dft->DFT_Re) +
                (int32_t) (ptr_dft->DFT_Im * ptr_dft->DFT_Im);
    ptr_dft->DFT_Mag = f_Raiz(int_mag);                    /* Magnitud del Fasor */

    /* Valor Absoluto de la Parte Real, para determinar el coseno del ángulo */
    if(ptr_dft->DFT_Re < 0)
        Re_Aux = -ptr_dft->DFT_Re;
    else
        Re_Aux = ptr_dft->DFT_Re;

    /* Valor al que se debe apuntar en la tabla de Coseno Inverso */
    ptr_dft->DFT_Ph = ((Re_Aux * 255) / ptr_dft->DFT_Mag);

    if(ptr_dft->DFT_Ph <= 256 && ptr_dft->DFT_Ph >= 0)

```

Diseño de una Unidad de Medición Fasorial con tecnología ARM

```

{
ptr_dft->DFT_Ph = (uint32_t)arc_cos[ptr_dft->DFT_Ph];          /* 1° Cuadrante */
if (ptr_dft->DFT_Re<0 && ptr_dft->DFT_Im>=0)
{
ptr_dft->DFT_Ph = 18000 - ptr_dft->DFT_Ph;          /* 2° Cuadrante */
}
else if (ptr_dft->DFT_Re<=0 && ptr_dft->DFT_Im<0)
{
ptr_dft->DFT_Ph = ptr_dft->DFT_Ph + 18000;          /* 3° Cuadrante */
}
else if (ptr_dft->DFT_Re>0 && ptr_dft->DFT_Im <=0)
{
ptr_dft->DFT_Ph = 36000 - ptr_dft->DFT_Ph;          /* 4° Cuadrante */
}
}
if((ptr_dft == &DFT_Buffer_Va) || (ptr_dft == &DFT_Buffer_Vb) ||
(ptr_dft == &DFT_Buffer_Vc))
{
ptr_dft->DFT_Ph = ptr_dft->DFT_Ph - 600;          /* Compensación */
}
ptr_dft->DFT_Rad = ptr_dft->DFT_Ph * 0.174533;          /* Conversión a x10^-3 rads */
ptr_dft->DFT_Ang = (float)ptr_dft->DFT_Ph / 100;          /* Ángulo de 0-360 */
};

```

```

/*****
*                               Función que calcula la frecuencia                               *
*****/

```

```

uint32_t f_Frec(FREC_STRUCT_PTR frec_ptr)
{
float abs_actual, frac_muestra_new, frac_muestra_act;
static uint32_t frec_counter=0;

/* Determinación del cruce por cero con pendiente negativa */
if( frec_ptr->v_act <= 0 && frec_ptr->v_ant > 0)
{
if(frec_ptr->v_act < 0)
{
abs_actual = -frec_ptr->v_act;
frac_muestra_new = (abs_actual)/(frec_ptr->v_ant +
abs_actual);          /* Frac de Muestra Nueva */
frac_muestra_act = 1 - frac_muestra_new;          /* Frac M. Actual*/
frec_ptr->suma_muestras = frec_ptr->suma_muestras +
frac_muestra_new;          /* Acumulación de F. de M. */
/* Cálculo de la Frecuencia */
frec_ptr->frecuencia = AJUSTE_FREC / frec_ptr->suma_muestras;
frec_ptr->suma_muestras = frac_muestra_act;
}
else
{
/* Cálculo de la Frecuencia */
frec_ptr->frecuencia = AJUSTE_FREC / frec_ptr->suma_muestras;
frec_ptr->suma_muestras = 0;          /* Asignación de F. Actual */
}
/* Guardado de los datos para el Filtro Mediana */
if(!flag_frec)
{
datos[frec_counter] = frec_ptr->frecuencia;
frec_counter++;
if(frec_counter == ORDEN)          /* Si el buffer está completo */
{
frec_counter = 0;          /* Reinicio del cont de frecuencia */
flag_frec=1;          /* Se activa bandera aux de frec */
}
}
}
}

```

Diseño de una Unidad de Medición Fasorial con tecnología ARM

```

    }
}
/* Si no existe cruce por cero con pendiente negativa */
else
{
    /* Incremento del Número de muestras en 1 */
    frec_ptr->suma_muestras = frec_ptr->suma_muestras + 1;
}
/* Asignación del valor de la muestra actual a la muestra anterior */
frec_ptr->v_ant = frec_ptr->v_act;
};

/*****
 * Función que prepara los datos para ser transmitidos *
 *****/
void preparacion_transmision(DFT_STRUCT *ptr_dft)
{
    static uint8_t aux=0;
    /* Permite determinar la posición en el arreglo de Fasores acorde a la señal que se este analizando */
    if(ptr_dft == &DFT_Buffer_Va)
        aux=0;
    if(ptr_dft == &DFT_Buffer_Vb)
        aux=4;
    if(ptr_dft == &DFT_Buffer_Vc)
        aux=8;
    if(ptr_dft == &DFT_Buffer_Ia)
        aux=12;
    if(ptr_dft == &DFT_Buffer_Ib)
        aux=16;
    if(ptr_dft == &DFT_Buffer_Ic)
        aux=20;
    /* Si el formato de datos se requiere en Parte Real y Parte Imaginaria */
    if(formato == RECTANGULAR)
    {
        /* Parte Alta Real */
        Buffer_Fasores[aux] = (unsigned char)((ptr_dft->DFT_Re>>8) & 0xff);
        /* Parte Baja Real */
        Buffer_Fasores[aux+1] = (unsigned char)(ptr_dft->DFT_Re & 0xff);
        /* Parte Alta Imaginaria */
        Buffer_Fasores[aux+2] = (unsigned char)((ptr_dft->DFT_Im>>8) & 0xff);
        /* Parte Baja Imaginaria */
        Buffer_Fasores[aux+3] = (unsigned char)(ptr_dft->DFT_Im & 0xff);
    }
    /* Si el formato de datos se requiere en Magnitud y Ángulo (en radianes) */
    if(formato == POLAR)
    {
        /* Parte Alta Magnitud */
        Buffer_Fasores[aux] = (unsigned char)((ptr_dft->DFT_Mag>>8) & 0xff);
        /* Parte Baja Magnitud */
        Buffer_Fasores[aux+1] = (unsigned char)(ptr_dft->DFT_Mag & 0xff);
        /* Parte Alta Ángulo */
        Buffer_Fasores[aux+2] = (unsigned char)((ptr_dft->DFT_Rad>>8) & 0xff);
        /* Parte Baja Ángulo */
        Buffer_Fasores[aux+3] = (unsigned char)(ptr_dft->DFT_Rad & 0xff);
    }
}
}

#endif /* MEDICIONES_H_ */

```

A.3 HEADER DE DATOS DE TRAMAS (DATOS_TRAMA.H)

```

* Datos_Trama.h
#ifdef DATOS_TRAMA_H_
#define DATOS_TRAMA_H_

uint16_t CMD=2; /* 2 para seleccionar la transmisión de trama de datos */
uint16_t FRAMESIZE_DT=TAM_DATA;
uint16_t FRAMESIZE_CF=TAM_CONFIG;
uint16_t FRAMESIZE_HD=TAM_HDR;
uint16_t FRAMESIZE_CD=TAM_COMMAND;
uint16_t IDCODE=1;
uint32_t SOC=0;
uint32_t FRACSEC=0;
uint16_t STAT=0;
uint16_t FREQ=0;
uint16_t DFREQ=0;
uint16_t ANALOG=0;
uint16_t DIGITAL=0;
uint32_t TIMEBASE=1000000; /* microsegundos */
uint16_t NUM_PMU=1;
uint16_t PHNMR=6;
uint16_t ANNMR=0;
uint16_t DGNMR=0;
uint32_t PHUNIT_V=915527; /* (Tensión del Sistema Donde irá el PMU/32768)*10e5 */
uint32_t PHUNIT_I=45776; /* (Corriente del Sistema Donde irá el PMU/32768)*10e5 */
uint16_t SYNC;
uint8_t FNOM=0; /* 0 si la fundamental es 60 Hz, 1 si la fundamental es 50 Hz */
uint16_t CFGCNT=0;
uint16_t DATA_RATE=1;

#endif /* DATOS_TRAMA_H_ */

```

A.4 HEADER DE TRAMAS (TRAMAS.H)

```

* tramas.h
#ifdef TRAMAS_H_
#define TRAMAS_H_

#include "mediciones.h"
#include "Datos_Trama.h"

/*****
* Función que calcula el Segundo de Centenario
*****/
uint32_t f_SOC(void)
{
    static uint16_t dia;
    uint32_t soc=0;

    sec=(uint16_t) (tiempo[4])*10 + (uint16_t) tiempo[5];
    min=(uint16_t) (tiempo[2])*10 + (uint16_t) tiempo[3]*60;
    horas=(uint16_t) (tiempo[0])*10 + (uint16_t) tiempo[1]*3600;
}

```

Diseño de una Unidad de Medición Fasorial con tecnología ARM

```
    dia = horas + min + sec;
    soc = base + dia;
    return (soc);
}

/*****
*                               Función para llenar la Trama de Datos                               *
*****/
void data_frame(void)
{
    static int16_t freq;
    freq = (60 - frec_ptr->frec_mediana) * 1000;
    SYNC = 43521;
    Data_Frame[0]=(unsigned char)((SYNC>>8) & 0xff);
    Data_Frame[1]=(unsigned char)(SYNC & 0xff);
    Data_Frame[2]=(unsigned char)((FRAMESIZE_DT>>8) & 0xff);
    Data_Frame[3]=(unsigned char)(FRAMESIZE_DT & 0xff);
    Data_Frame[4]=(unsigned char)((IDCODE>>8) & 0xff);
    Data_Frame[5]=(unsigned char)(IDCODE & 0xff);
    Data_Frame[6]=(unsigned char)((SOC>>24) & 0xff);
    Data_Frame[7]=(unsigned char)((SOC>>16) & 0xff);
    Data_Frame[8]=(unsigned char)((SOC>>8) & 0xff);
    Data_Frame[9]=(unsigned char)(SOC & 0xff);
    Data_Frame[10]=(unsigned char)((FRACSEC>>24) & 0xff);
    Data_Frame[11]=(unsigned char)((FRACSEC>>16) & 0xff);
    Data_Frame[12]=(unsigned char)((FRACSEC>>8) & 0xff);
    Data_Frame[13]=(unsigned char)(FRACSEC & 0xff);
    Data_Frame[14]=(unsigned char)((STAT>>8) & 0xff);
    Data_Frame[15]=(unsigned char)(STAT & 0xff);
    Data_Frame[16]=Buffer_Fasores[0];
    Data_Frame[17]=Buffer_Fasores[1];
    Data_Frame[18]=Buffer_Fasores[2];
    Data_Frame[19]=Buffer_Fasores[3];
    Data_Frame[20]=Buffer_Fasores[4];
    Data_Frame[21]=Buffer_Fasores[5];
    Data_Frame[22]=Buffer_Fasores[6];
    Data_Frame[23]=Buffer_Fasores[7];
    Data_Frame[24]=Buffer_Fasores[8];
    Data_Frame[25]=Buffer_Fasores[9];
    Data_Frame[26]=Buffer_Fasores[10];
    Data_Frame[27]=Buffer_Fasores[11];
    Data_Frame[28]=Buffer_Fasores[12];
    Data_Frame[29]=Buffer_Fasores[13];
    Data_Frame[30]=Buffer_Fasores[14];
    Data_Frame[31]=Buffer_Fasores[15];
    Data_Frame[32]=Buffer_Fasores[16];
    Data_Frame[33]=Buffer_Fasores[17];
    Data_Frame[34]=Buffer_Fasores[18];
    Data_Frame[35]=Buffer_Fasores[19];
    Data_Frame[36]=Buffer_Fasores[20];
    Data_Frame[37]=Buffer_Fasores[21];
    Data_Frame[38]=Buffer_Fasores[22];
    Data_Frame[39]=Buffer_Fasores[23];
    Data_Frame[40]=(unsigned char)((freq>>8) & 0xff);
    Data_Frame[41]=(unsigned char)(freq & 0xff);
    Data_Frame[42]=(unsigned char)((DFREQ>>8) & 0xff);
    Data_Frame[43]=(unsigned char)(DFREQ & 0xff);
    Data_Frame[44]=(unsigned char)((ANALOG>>8) & 0xff);
    Data_Frame[45]=(unsigned char)(ANALOG & 0xff);
    Data_Frame[46]=(unsigned char)((DIGITAL>>8) & 0xff);
    Data_Frame[47]=(unsigned char)(DIGITAL & 0xff);
}
```

Diseño de una Unidad de Medición Fasorial con tecnología ARM

```
/*
*****
*                               Función para llenar la Trama de Configuración                               *
*****
*/
void config_frame(void)
{
    SYNC = 43569;
    Config_Frame[0]=(unsigned char)((SYNC>>8) & 0xff);
    Config_Frame[1]=(unsigned char)(SYNC & 0xff);
    Config_Frame[2]=(unsigned char)((FRAMESIZE_CF>>8) & 0xff);
    Config_Frame[3]=(unsigned char)(FRAMESIZE_CF & 0xff);
    Config_Frame[4]=(unsigned char)((IDCODE>>8) & 0xff);
    Config_Frame[5]=(unsigned char)(IDCODE & 0xff);
    Config_Frame[6]=(unsigned char)((SOC>>24) & 0xff);
    Config_Frame[7]=(unsigned char)((SOC>>16) & 0xff);
    Config_Frame[8]=(unsigned char)((SOC>>8) & 0xff);
    Config_Frame[9]=(unsigned char)(SOC & 0xff);
    Config_Frame[10]=(unsigned char)((FRACSEC>>24) & 0xff);
    Config_Frame[11]=(unsigned char)((FRACSEC>>16) & 0xff);
    Config_Frame[12]=(unsigned char)((FRACSEC>>8) & 0xff);
    Config_Frame[13]=(unsigned char)(FRACSEC & 0xff);
    Config_Frame[14]=(unsigned char)((TIMEBASE>>24) & 0xff);
    Config_Frame[15]=(unsigned char)((TIMEBASE>>16) & 0xff);
    Config_Frame[16]=(unsigned char)((TIMEBASE>>8) & 0xff);
    Config_Frame[17]=(unsigned char)(TIMEBASE & 0xff);
    Config_Frame[18]=(unsigned char)((NUM_PMU>>8) & 0xff);
    Config_Frame[19]=(unsigned char)(NUM_PMU & 0xff);
    Config_Frame[20]=(unsigned char)(0x45);           //E
    Config_Frame[21]=(unsigned char)(0x73);         //s
    Config_Frame[22]=(unsigned char)(0x74);         //t
    Config_Frame[23]=(unsigned char)(0x61);         //a
    Config_Frame[24]=(unsigned char)(0x63);         //c
    Config_Frame[25]=(unsigned char)(0x69);         //i
    Config_Frame[26]=(unsigned char)(0x6f);         //o
    Config_Frame[27]=(unsigned char)(0x6e);         //n
    Config_Frame[28]=(unsigned char)(0x53);         //S
    Config_Frame[29]=(unsigned char)(0x45);         //E
    Config_Frame[30]=(unsigned char)(0x50);         //P
    Config_Frame[31]=(unsigned char)(0x49);         //I
    Config_Frame[32]=(unsigned char)(0x32);         //2
    Config_Frame[33]=(unsigned char)(0x30);         //0
    Config_Frame[34]=(unsigned char)(0x31);         //1
    Config_Frame[35]=(unsigned char)(0x36);         //6
    Config_Frame[36]=(unsigned char)((IDCODE>>8) & 0xff);
    Config_Frame[37]=(unsigned char)(IDCODE & 0xff);
    Config_Frame[38]=(unsigned char)((formato>>8) & 0xff);
    Config_Frame[39]=(unsigned char)(formato & 0xff);
    Config_Frame[40]=(unsigned char)((PHNMR>>8) & 0xff);
    Config_Frame[41]=(unsigned char)(PHNMR & 0xff);
    Config_Frame[42]=(unsigned char)((ANNMR>>8) & 0xff);
    Config_Frame[43]=(unsigned char)(ANNMR & 0xff);
    Config_Frame[44]=(unsigned char)((DGNMR>>8) & 0xff);
    Config_Frame[45]=(unsigned char)(DGNMR & 0xff);
    Config_Frame[46]=(unsigned char)(0x56);         //V
    Config_Frame[47]=(unsigned char)(0x20);
    Config_Frame[48]=(unsigned char)(0x46);         //F
    Config_Frame[49]=(unsigned char)(0x61);         //a
    Config_Frame[50]=(unsigned char)(0x73);         //s
    Config_Frame[51]=(unsigned char)(0x65);         //e
}
```

Diseño de una Unidad de Medición Fasorial con tecnología ARM

```
Config_Frame[52]=(unsigned char) (0x20);
Config_Frame[53]=(unsigned char) (0x41); //A
Config_Frame[54]=(unsigned char) (0x20);
Config_Frame[55]=(unsigned char) (0x20);
Config_Frame[56]=(unsigned char) (0x20);
Config_Frame[57]=(unsigned char) (0x20);
Config_Frame[58]=(unsigned char) (0x20);
Config_Frame[59]=(unsigned char) (0x20);
Config_Frame[60]=(unsigned char) (0x20);
Config_Frame[61]=(unsigned char) (0x20);
Config_Frame[62]=(unsigned char) (0x56); //V
Config_Frame[63]=(unsigned char) (0x20);
Config_Frame[64]=(unsigned char) (0x46); //F
Config_Frame[65]=(unsigned char) (0x61); //a
Config_Frame[66]=(unsigned char) (0x73); //s
Config_Frame[67]=(unsigned char) (0x65); //e
Config_Frame[68]=(unsigned char) (0x20);
Config_Frame[69]=(unsigned char) (0x42); //B
Config_Frame[70]=(unsigned char) (0x20);
Config_Frame[71]=(unsigned char) (0x20);
Config_Frame[72]=(unsigned char) (0x20);
Config_Frame[73]=(unsigned char) (0x20);
Config_Frame[74]=(unsigned char) (0x20);
Config_Frame[75]=(unsigned char) (0x20);
Config_Frame[76]=(unsigned char) (0x20);
Config_Frame[77]=(unsigned char) (0x20);
Config_Frame[78]=(unsigned char) (0x56); //V
Config_Frame[79]=(unsigned char) (0x20);
Config_Frame[80]=(unsigned char) (0x46); //F
Config_Frame[81]=(unsigned char) (0x61); //a
Config_Frame[82]=(unsigned char) (0x73); //s
Config_Frame[83]=(unsigned char) (0x65); //e
Config_Frame[84]=(unsigned char) (0x20);
Config_Frame[85]=(unsigned char) (0x43); //C
Config_Frame[86]=(unsigned char) (0x20);
Config_Frame[87]=(unsigned char) (0x20);
Config_Frame[88]=(unsigned char) (0x20);
Config_Frame[89]=(unsigned char) (0x20);
Config_Frame[90]=(unsigned char) (0x20);
Config_Frame[91]=(unsigned char) (0x20);
Config_Frame[92]=(unsigned char) (0x20);
Config_Frame[93]=(unsigned char) (0x20);
Config_Frame[94]=(unsigned char) (0x49); //I
Config_Frame[95]=(unsigned char) (0x20);
Config_Frame[96]=(unsigned char) (0x46); //F
Config_Frame[97]=(unsigned char) (0x61); //a
Config_Frame[98]=(unsigned char) (0x73); //s
Config_Frame[99]=(unsigned char) (0x65); //e
Config_Frame[100]=(unsigned char) (0x20);
Config_Frame[101]=(unsigned char) (0x41); //A
Config_Frame[102]=(unsigned char) (0x20);
Config_Frame[103]=(unsigned char) (0x20);
Config_Frame[104]=(unsigned char) (0x20);
Config_Frame[105]=(unsigned char) (0x20);
Config_Frame[106]=(unsigned char) (0x20);
Config_Frame[107]=(unsigned char) (0x20);
Config_Frame[108]=(unsigned char) (0x20);
Config_Frame[109]=(unsigned char) (0x20);
Config_Frame[110]=(unsigned char) (0x49); //I
Config_Frame[111]=(unsigned char) (0x20);
Config_Frame[112]=(unsigned char) (0x46); //F
Config_Frame[113]=(unsigned char) (0x61); //a
Config_Frame[114]=(unsigned char) (0x73); //s
```

Diseño de una Unidad de Medición Fasorial con tecnología ARM

```

Config_Frame[115]=(unsigned char) (0x65);           //e
Config_Frame[116]=(unsigned char) (0x20);
Config_Frame[117]=(unsigned char) (0x42);           //B
Config_Frame[118]=(unsigned char) (0x20);
Config_Frame[119]=(unsigned char) (0x20);
Config_Frame[120]=(unsigned char) (0x20);
Config_Frame[121]=(unsigned char) (0x20);
Config_Frame[122]=(unsigned char) (0x20);
Config_Frame[123]=(unsigned char) (0x20);
Config_Frame[124]=(unsigned char) (0x20);
Config_Frame[125]=(unsigned char) (0x20);
Config_Frame[126]=(unsigned char) (0x49);           //I
Config_Frame[127]=(unsigned char) (0x20);
Config_Frame[128]=(unsigned char) (0x46);           //F
Config_Frame[129]=(unsigned char) (0x61);           //a
Config_Frame[130]=(unsigned char) (0x73);           //s
Config_Frame[131]=(unsigned char) (0x65);           //e
Config_Frame[132]=(unsigned char) (0x20);
Config_Frame[133]=(unsigned char) (0x43);           //C
Config_Frame[134]=(unsigned char) (0x20);
Config_Frame[135]=(unsigned char) (0x20);
Config_Frame[136]=(unsigned char) (0x20);
Config_Frame[137]=(unsigned char) (0x20);
Config_Frame[138]=(unsigned char) (0x20);
Config_Frame[139]=(unsigned char) (0x20);
Config_Frame[140]=(unsigned char) (0x20);
Config_Frame[141]=(unsigned char) (0x20);
Config_Frame[142]=(unsigned char) (0x00);           // 0 para V
Config_Frame[143]=(unsigned char) ((PHUNIT_V>>16) & 0xff);
Config_Frame[144]=(unsigned char) ((PHUNIT_V>>8) & 0xff);
Config_Frame[145]=(unsigned char) (PHUNIT_V & 0xff);
Config_Frame[146]=(unsigned char) (0x00);           // 0 para V
Config_Frame[147]=(unsigned char) ((PHUNIT_V>>16) & 0xff);
Config_Frame[148]=(unsigned char) ((PHUNIT_V>>8) & 0xff);
Config_Frame[149]=(unsigned char) (PHUNIT_V & 0xff);
Config_Frame[150]=(unsigned char) (0x00);           // 0 para V
Config_Frame[151]=(unsigned char) ((PHUNIT_V>>16) & 0xff);
Config_Frame[152]=(unsigned char) ((PHUNIT_V>>8) & 0xff);
Config_Frame[153]=(unsigned char) (PHUNIT_V & 0xff);
Config_Frame[154]=(unsigned char) (0x01);           // 1 para I
Config_Frame[155]=(unsigned char) ((PHUNIT_I>>16) & 0xff);
Config_Frame[156]=(unsigned char) ((PHUNIT_I>>8) & 0xff);
Config_Frame[157]=(unsigned char) (PHUNIT_I & 0xff);
Config_Frame[158]=(unsigned char) (0x01);           // 1 para I
Config_Frame[159]=(unsigned char) ((PHUNIT_I>>16) & 0xff);
Config_Frame[160]=(unsigned char) ((PHUNIT_I>>8) & 0xff);
Config_Frame[161]=(unsigned char) (PHUNIT_I & 0xff);
Config_Frame[162]=(unsigned char) (0x01);           // 1 para I
Config_Frame[163]=(unsigned char) ((PHUNIT_I>>16) & 0xff);
Config_Frame[164]=(unsigned char) ((PHUNIT_I>>8) & 0xff);
Config_Frame[165]=(unsigned char) (PHUNIT_I & 0xff);
Config_Frame[166]=(unsigned char) (0x00);
Config_Frame[167]=(unsigned char) (0x00);
Config_Frame[168]=(unsigned char) (0x00);
Config_Frame[169]=(unsigned char) (0x00);
Config_Frame[170]=(unsigned char) (0x00);
Config_Frame[171]=(unsigned char) (0x00);
Config_Frame[172]=(unsigned char) (0x00);
Config_Frame[173]=(unsigned char) (0x00);
Config_Frame[174]=(unsigned char) (0x00);
Config_Frame[175]=(unsigned char) (FNOM & 0xff);
Config_Frame[176]=(unsigned char) ((CFGCNT>>8) & 0xff);
Config_Frame[177]=(unsigned char) (CFGCNT & 0xff);

```

Diseño de una Unidad de Medición Fasorial con tecnología ARM

```
Config_Frame[178]=(unsigned char)((DATA_RATE>>8) & 0xff);
Config_Frame[179]=(unsigned char)(DATA_RATE & 0xff);
}

/*****
 *                               Función para llenar la Trama de Cabecal                               *
 *****/
void header_frame(void)
{
    SYNC=43537;
    Header_Frame[0]=(unsigned char)((SYNC>>8) & 0xff);
    Header_Frame[1]=(unsigned char)(SYNC & 0xff);
    Header_Frame[2]=(unsigned char)((FRAMESIZE_HD>>8) & 0xff);
    Header_Frame[3]=(unsigned char)(FRAMESIZE_HD & 0xff);
    Header_Frame[4]=(unsigned char)((IDCODE>>8) & 0xff);
    Header_Frame[5]=(unsigned char)(IDCODE & 0xff);
    Header_Frame[6]=(unsigned char)((SOC>>24) & 0xff);
    Header_Frame[7]=(unsigned char)((SOC>>16) & 0xff);
    Header_Frame[8]=(unsigned char)((SOC>>8) & 0xff);
    Header_Frame[9]=(unsigned char)(SOC & 0xff);
    Header_Frame[10]=(unsigned char)((FRACSEC>>24) & 0xff);
    Header_Frame[11]=(unsigned char)((FRACSEC>>16) & 0xff);
    Header_Frame[12]=(unsigned char)((FRACSEC>>8) & 0xff);
    Header_Frame[13]=(unsigned char)(FRACSEC & 0xff);
    Header_Frame[14]=(unsigned char)'S';
    Header_Frame[15]=(unsigned char)'E';
    Header_Frame[16]=(unsigned char)'P';
    Header_Frame[17]=(unsigned char)'I';
}

/*****
 *                               Función para llenar la Trama de Comando                               *
 *****/
void command_frame(void)
{
    SYNC=43585;
    Comm_Frame[0]=(unsigned char)((SYNC>>8) & 0xff);
    Comm_Frame[1]=(unsigned char)(SYNC & 0xff);
    Comm_Frame[2]=(unsigned char)((FRAMESIZE_CD>>8) & 0xff);
    Comm_Frame[3]=(unsigned char)(FRAMESIZE_CD & 0xff);
    Comm_Frame[4]=(unsigned char)((IDCODE>>8) & 0xff);
    Comm_Frame[5]=(unsigned char)(IDCODE & 0xff);
    Comm_Frame[6]=(unsigned char)((SOC>>24) & 0xff);
    Comm_Frame[7]=(unsigned char)((SOC>>16) & 0xff);
    Comm_Frame[8]=(unsigned char)((SOC>>8) & 0xff);
    Comm_Frame[9]=(unsigned char)(SOC & 0xff);
    Comm_Frame[10]=(unsigned char)((FRACSEC>>24) & 0xff);
    Comm_Frame[11]=(unsigned char)((FRACSEC>>16) & 0xff);
    Comm_Frame[12]=(unsigned char)((FRACSEC>>8) & 0xff);
    Comm_Frame[13]=(unsigned char)(FRACSEC & 0xff);
    Comm_Frame[14]=(unsigned char)((CMD>>8) & 0xff);
    Comm_Frame[15]=(unsigned char)(CMD & 0xff);
}

/*****
 *                               Función para iniciar la transmisión de la Trama de Datos                               *
 *****/
void tx_datos(uint16_t size_TX_dat)
{
    uint16_t temp,j;
    temp = f_CRC(&Data_Frame[0],size_TX_dat);
```

Diseño de una Unidad de Medición Fasorial con tecnología ARM

```
    Data_Frame[size_TX_dat]=(unsigned char) (temp>>8 & 0xff); /* Parte alta CRC */
    Data_Frame[size_TX_dat+1]=(unsigned char) (temp & 0xff); /* Parte baja CRC */
    UART_TxIsrEnable (UART1);
}

/*****
 * Función para iniciar la transmisión de la Trama de Configuración
 *****/
void tx_conf(uint16_t size_TX_conf)
{
    uint16_t temp,j;
    temp = f_CRC(&Config_Frame[0],size_TX_conf);
    Config_Frame[size_TX_conf]=(unsigned char) (temp>>8 & 0xff); /* Parte alta CRC */
    Config_Frame[size_TX_conf+1]=(unsigned char) (temp & 0xff); /* Parte alta CRC */
    UART_TxIsrEnable (UART1);
}

/*****
 * Función para iniciar la transmisión de la Trama de Cabecal
 *****/
void tx_head(uint16_t size_TX_hdr)
{
    uint16_t temp,j;
    temp = f_CRC(&Header_Frame[0],size_TX_hdr);
    Header_Frame[size_TX_hdr]=(unsigned char) (temp>>8 & 0xff); /* Parte alta CRC */
    Header_Frame[size_TX_hdr+1]=(unsigned char) (temp & 0xff); /* Parte alta CRC */
    UART_TxIsrEnable (UART1);
}

/*****
 * Función para iniciar la transmisión de la Trama de Comando
 *****/
void tx_comm(uint16_t size_TX_comm)
{
    uint16_t temp,j;
    temp = f_CRC(&Comm_Frame[0],size_TX_comm);
    Comm_Frame[size_TX_comm]=(unsigned char) (temp>>8 & 0xff); /* Parte alta CRC */
    Comm_Frame[size_TX_comm+1]=(unsigned char) (temp & 0xff); /* Parte alta CRC */
}

#endif /* TRAMAS_H_ */

/*****
 * Función que calcula la CRC-CCITT
 *****/
uint16_t f_CRC(unsigned char *Mensaje, uint16_t Tam_Men)
{
    /* Algoritmo obtenido del Estándar IEEE C37.118.2 */
    uint16_t crc=0xffff;
    uint16_t temp;
    uint16_t quick;
    int i;
    for(i=0;i<Tam_Men;i++)
    {
        temp = (crc>>8)^Mensaje[i];
        crc <<= 8;
        quick = temp ^ (temp>>4);
        crc ^= quick;
        quick <<= 5;
        crc ^= quick;
        quick <<= 7;
    }
}
```

Diseño de una Unidad de Medición Fasorial con tecnología ARM

```
        crc ^= quick;
    }
    return (crc);
}

/*****
 *                               Función para iniciar la Comunicación                               *
 *****/
void f_comunicacion(void)
{
    /* Si no hay error en la transmisión del Command Frame */
    if ((f_CRC(&Comm_Frame[0], TAM_COMMAND)) == 0)
    {
        if (Comm_Frame[5] == NUM_ESCLAVO) /* Si corresponde el ID */
        {
            switch (Comm_Frame[15]) /* Pregunta por el CMD del Command Frame */
            {
                case DATA_OFF:    conf_enviada=0;
                                    break;
                case DATA_ON:     size_env_data_on=TAM_DATA-2;
                                    conf_enviada=1;
                                    break;
                case SEND_HDR:     conf_enviada=2;
                                    num_bytes_enviar=TAM_HDR-2;
                                    tx_head(num_bytes_enviar);
                                    break;
                case SEND_CFG1:    conf_enviada=0;
                                    num_bytes_enviar=TAM_CONFIG-2;
                                    tx_conf(num_bytes_enviar);
                                    break;
                case SEND_CFG2:    conf_enviada=0;
                                    num_bytes_enviar=TAM_CONFIG-2;
                                    tx_conf(num_bytes_enviar);
                                    break;
                case SEND_CFG3:    conf_enviada=0;
                                    num_bytes_enviar=TAM_CONFIG-2;
                                    tx_conf(num_bytes_enviar);
                                    break;
                case EXT_FRAME:    conf_enviada=0;
                                    break;
                case SEND_COMM:    conf_enviada=3;
                                    UART_TxIsrEnable (UART1);
                                    break;
                default:           break;
            }
        }
    }
}
```

APÉNDICE B. ESTÁNDAR IEEE C37.118

A continuación, se presentan los puntos considerados en el Estándar IEEE C37.118 2011 que sirven de referencia para el desarrollo de este trabajo. Cabe mencionar que no se realizaron pruebas a la PMU en condiciones dinámicas. El Estándar está dividido en dos partes: IEEE C37.118.1 que es el Estándar para Mediciones de Sincrofasores para Sistemas de Potencia y el IEEE C37.118.2 que es el Estándar para Transferencia de Datos de Sincrofasores para Sistemas de Potencia. [49, 50]

B.1 INTRODUCCIÓN

La medición, protección y sistemas de control basados en el uso de la computadora digital, llegaron a ser características comunes de las subestaciones eléctricas de potencia. Dicho sistema usa el muestreo de datos para calcular diversas cantidades, tales como el fasor de corriente y de voltaje. En muchas funciones de adquisición de datos y sistemas de protección, los fasores y su utilidad se incrementarán en un futuro, debido a la referencia que tienen en base a un tiempo común. Esto se emplea para sincronizar la señal de entrada procesada en varios sitios de medición. [13]

De varios fasores sincronizados se obtiene un grupo de mediciones simultáneas, que proporciona un método mejorado en gran escala para analizar los fenómenos dinámicos del sistema de potencia, a fin de mejorar el monitoreo, protección y control del mismo sistema. En el estándar IEEE C37.118 se proporcionan definiciones claras de varios conceptos, entre los cuales se encuentra el concepto de fasor, sincrofasor, errores, etc. Además, determina los elementos con los que debe contar el sistema de comunicación para que el sincrofasor envíe los datos de las mediciones que realiza. [49,50]

B.2 DESCRIPCIÓN

Esta norma trata las mediciones fasoriales sincronizadas que se utilizan en los sistemas eléctricos de potencia. Define la medición, proporciona un método de cuantificación de las mediciones y

calidad de las pruebas especificadas. Define también, formatos para la transmisión de datos en tiempo real. [49,50]

B.2.1 NECESIDADES DEL ESTÁNDAR

Los recientes desarrollos en el campo de la adquisición de datos en sistemas de potencia proveen al usuario la habilidad de calcular cuantitativamente los fasores en tiempo real. Diferentes enfoques de hardware y software están siendo implementados para registrar, transmitir y analizar estas mediciones, tanto en tiempo real como después del evento. El estándar para sincrofasores ayudará a asegurar máximos beneficios para la medición de fasores y permite el intercambio de información entre una amplia variedad de sistemas para los usuarios tanto en tiempo real como en mediciones de fasores fuera de línea. [49,50]

B.3 DEFINICIONES

Para el propósito de este documento, los siguientes términos y definiciones aplican. [13]

- ◇ **Anti-alias:** El proceso de filtrado de una señal para eliminar los componentes de esa señal cuya frecuencia es igual o superior a la mitad de la tasa de Nyquist (velocidad de muestreo). Si no se elimina, estos componentes de la señal aparecen como un componente de menor frecuencia (un alias).
- ◇ **Tiempo Universal Coordinado (UTC):** El tiempo del día al que se encuentra el primer meridiano de la Tierra (0° longitud). Es distribuido por varios medios, incluyendo el sistema GPS.
- ◇ **IEEE punto flotante:** Una representación de 32 bits de un número real, de conformidad con la norma IEEE Std 754.
- ◇ **Error de frecuencia (FE):** La medición del error entre la frecuencia teórica y la frecuencia medida para el instante de tiempo dado.
- ◇ **Error (RFE) de Razón de Cambio de Frecuencia (ROCOF):** La medición del error entre la ROCOF teórica y la ROCOF medida para el instante de tiempo dado.
- ◇ **Frecuencia de Nyquist:** Una frecuencia que es la mitad de la frecuencia de muestreo de una sistema procesado de señales discretas.

- ◇ **Tasa de Nyquist:** Una tasa que es el doble de la frecuencia más alta en la señal de entrada analógica. La señal analógica debe ser muestreada a una tasa mayor que la tasa de Nyquist para ser representada con precisión en forma digital.
- ◇ **Fasor:** Un número complejo equivalente de una cantidad de una onda coseno simple tal que el módulo es la amplitud de la onda coseno y el ángulo complejo (en forma polar) es ángulo de fase de la onda coseno.
- ◇ **Trama:** En este estándar, una trama de datos es un conjunto de mediciones que corresponden a la misma estampa de tiempo.
- ◇ **Sincronismo:** El estado donde conectan los sistemas de corriente alterna, máquinas, o una combinación, operan a la misma frecuencia y en donde el ángulo de fase se desplaza entre tensiones constantes o varían de una constante y valor medio estable.
- ◇ **CRC-CCIT:** Revisión de la Redundancia Cíclica (CRC) de 16 bits calculado usando la generación del polinomio $X^{16} + X^{12} + X^5 + 1$, con semilla 0xFFFF (-1).
- ◇ **Fasor Sincronizado:** Un fasor calculado a partir de datos de muestras utilizando un tiempo estándar, como señal de referencia para la medición. Fasor sincronizado desde sitios remotos tienen una relación de fase definida común. [49,50]

B.4 MEDICIÓN DE SINCRÓFASORES

B.4.1 DEFINICIÓN DE FASOR

La forma de onda sinusoidal pura $x(t) = x_m \cos(\omega t + \theta)$ es representada comúnmente como un fasor $X = X_r + jX_i = (X_m/\sqrt{2})(e^{j\varphi})$, donde φ depende de la definición de la escala de tiempo. Para este estándar, este concepto básico es adaptado como la representación de una señal sinusoidal de sistemas de potencia. [49,50]

B.4.1.1 SEÑALES FUERA DE LA FRECUENCIA NOMINAL

La representación fasorial de una senoide es independiente de su frecuencia. Una senoide $x(t) = X_m \cos(2\pi f t + \varphi)$, tiene un fasor representado como $X = (X_m/\sqrt{2})(e^{j\varphi})$. El ángulo de fase φ del fasor es determinado por el comienzo de una señal sinusoidal en el tiempo ($t = 0$). [48]

Considerando que una senoide es observada en intervalos $\{0, T_0, 2T_0, \dots, nT_0\}$ llevando a las representaciones fasoriales correspondientes $\{X_0, X_1, X_2, \dots, X_n\}$. Esto es equivalente a tener un tiempo de referencia de observación inicializado al principio de cada intervalo. [49,50]

Si el intervalo de observación T_0 es igual a un múltiplo entero del periodo de la senoide $T = 1/f$, entonces un fasor constante es obtenido de cada observación. Por otro lado, si el intervalo de observación T_0 no es un múltiplo entero de T , el fasor observado tiene una magnitud constante, pero los ángulos de fase de la secuencia de fasores $\{X_0, X_1, X_2, \dots, X_n\}$ cambiará uniformemente a una razón $2\pi(f - f_0)T_0$, donde $f_0 = 1/T_0$. Esto es ilustrado en la Figura B.1. [48]

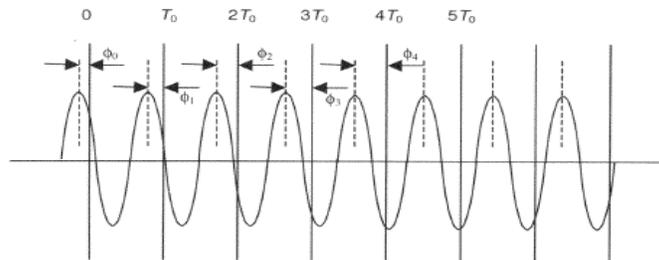


FIGURA B.1 SENOIDE CON PERIODO T OBSERVADA A INSTANTES MÚLTIPLOS DE T_0 .

B.4.2 DEFINICIÓN DE SINCR OFASOR

La representación de un sincrofasor X de una señal $x(t)$ es el valor complejo dado por la ecuación B.1.

$$X = X_r + jX_i = (X_m / \sqrt{2})e^{j\varphi} = \frac{X_m}{\sqrt{2}}(\cos \varphi + j \sin \varphi) \quad (\text{B.1})$$

En donde:

$X_m/\sqrt{2}$ es el valor RMS de la señal $x(t)$.

φ es el ángulo de fase instantáneo relativo a una función coseno a la frecuencia nominal del sistema sincronizado al UTC. Este ángulo está definido entre 0° cuando el máximo de $x(t)$ ocurre al conteo del segundo del UTC (señal de 1 pps), y -90° cuando el cruce positivo por cero ocurre al mismo UTC. La figura B.2 muestra esta relación, considerando que T_0 no es múltiplo entero de T . [48]

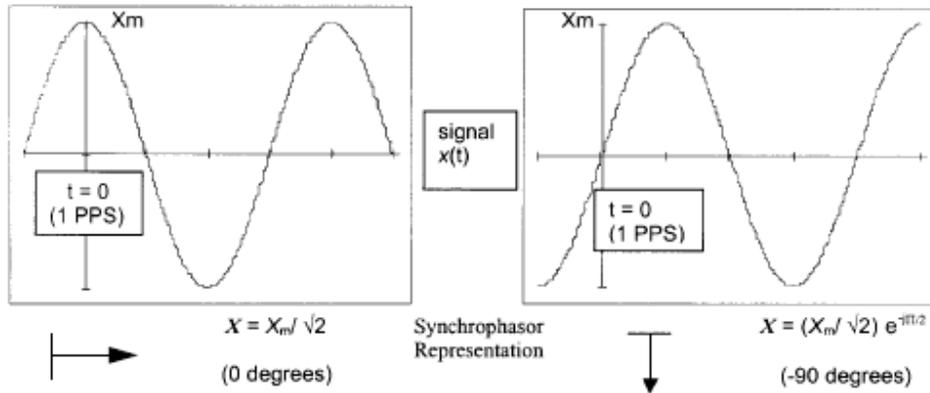


FIGURA B.2 CONVENIO DE REPRESENTACIÓN DE UN SINCRUFASOR

B.4.3 MEDICIÓN PARA EL ESTAMPADO DE TIEMPO PARA EL SINCRUFASOR

La medición de sincrofásor debe ser etiquetada con el tiempo correspondiente del UTC al tiempo de medición. Éste debe consistir de tres números: conteo del segundo del siglo (SOC, *Second Of Century*), un contero de fracción de segundo y un estado del valor de tiempo. El contador SOC deberá ser de 4 bytes binarios contado en segundos desde la media noche del primero de enero de 1970, para el actual segundo. Se deberán agregar o quitar segundos como sea necesario del contero para mantener la sincronización con el UTC. Este estampado de tiempo SOC es el mismo como lo es el sistema de cómputo de UNIX y es similar el uso para otros sistemas de cómputo incluyendo DOC, MAC OS y redes (NTP, *Networking Time Protocol*). [13]

B.4.4 SINCRONIZACIÓN DEL SISTEMA DE TIEMPO

La medición del sincrofásor debe ser sincronizado por el tiempo del UTC con una precisión suficiente. Note que un error de tiempo de $1 \mu\text{s}$ corresponde a un error de fase de 0.022° para un sistema de 60 Hz y 0.018° para un sistema de 50 Hz. Un error de fase de 0.01 radianes o 0.57° causa un TVE de 1% como es definido en la ecuación B.2. Este corresponde a un máximo error de tiempo de $\pm 26 \mu\text{s}$ para un sistema de 60 Hz y $\pm 31 \mu\text{s}$ para un sistema de 50 Hz. [13]

El sistema debe ser capaz de recibir el tiempo de una fuente de alta confiabilidad, tal como el Sistema de Posicionamiento Global (GPS), que puede proporcionar una precisión suficiente en el tiempo para mantener el TVE dentro de los límites requeridos y provee indicaciones de la pérdida de sincronización. [13]

B.5 REQUERIMIENTOS DE LAS MEDICIONES DEL SINCRUFASOR Y EL CUMPLIMIENTO DE LA VERIFICACIÓN

B.5.1 ESTIMACIÓN DEL SINCRUFASOR

Las estimaciones del sincrofase se deben realizar y transmitir a una razón F_s que es un número entero de tramas por segundo como lo especifica la variable DATA_RATE en la configuración de trama (ver tabla B.1 del apartado B.5.1.1). [49] Estas estimaciones deberán ser igual espaciadas dentro de un intervalo de 1s. La trama de datos puede incluir múltiples canales de la estimación de fasores, palabras analógicas y digitales, palabras combinadas en un solo estampado de tiempo. [49]

B.5.1.1 REPORTE DE TRAMAS

La PMU deberá soportar reportar en los datos un submúltiplo de la frecuencia nominal. Se requiere una tasa para sistemas de 50 Hz y otra para 60 Hz, las cuales son mostradas en la tabla B.1. [49]

TABLA B.1 RAZÓN DE REPORTE DE TRAMAS REQUERIDA POR EL PMU

Frecuencia del Sistema	50 Hz			60 Hz					
Tramas por Segundo (F_s)	10	25	50	10	12	15	20	30	60

La razón de reporte actual debe ser seleccionada por el usuario. Se admite el apoyo a otras tasas de reporte, incluso tasas de 1/s es alentada. [48] Hay que notar que tasas menores a 10/s no están sujetas a los requerimientos dinámicos de este estándar, lo que significa que no se requiere filtrado, por lo que tasas de frecuencias menores (<10/s) pueden ser directamente seleccionando cada enésima muestra de una mayor tasa. [48]

B.5.1.2 REPORTE DE TIEMPOS

Para un índice de reporte de N tramas por segundo, los tiempos de reporte deben ser eventualmente distanciados a través de cada segundo con un número de trama 0 (numerado de 0 hasta N-1) coincidente con el pps proveniente del GPS. Estos tiempos de reporte, o estampas de tiempo, son usados para determinar los valores instantáneos del sincrofase definidos en B.4.2. [48]

B.5.1.3 RESULTADOS DE EJEMPLO

La tabla B.2 proporciona valores del sincrofase definidos en la ecuación C1 por las formas de onda mostradas en la figura B.2. Los valores resultan de una velocidad de muestreo de 10 tramas

por segundo con una frecuencia del sistema de 60 Hz. Los valores del sincrofasor son mostrados para ángulos de fase base de 0° y -90° a la marca de tiempo de 1 pps mostrado en la figura B.2 para las señales de 60 y 61 Hz. Los valores del sincrofasor a 50 y 51 Hz para un sistema de 50 Hz son idénticos. [13]

TABLA B.2 VALORES DE SINCRUFASOR PARA UN SISTEMA DE 60 HZ CON FRECUENCIA DE MUESTREO DE 10 TRAMAS POR SEGUNDO

Tiempo	Fracción de tiempo		Valores de Sincrofasor para: Frecuencia 50Hz-Sistema 50 Hz Frecuencia 60 Hz-Sistema 60 Hz		Valores de Sincrofasor para: Frecuencia 51Hz-Sistema 50 Hz Frecuencia 61 Hz-Sistema 60Hz	
	Número de Trama	Fracción de Segundo	Sincrofasor (0°)	Sincrofasor (-90°)	Sincrofasor (0°)	Sincrofasor (-90°)
k-1	9	0.900000	$X_m/\sqrt{2}, \angle 0^\circ$	$X_m/\sqrt{2}, \angle -90^\circ$	$X_m/\sqrt{2}, \angle -36^\circ$	$X_m/\sqrt{2}, \angle -126^\circ$
k	0	0.000000	$X_m/\sqrt{2}, \angle 0^\circ$	$X_m/\sqrt{2}, \angle -90^\circ$	$X_m/\sqrt{2}, \angle 0^\circ$	$X_m/\sqrt{2}, \angle -90^\circ$
k	1	0.100000	$X_m/\sqrt{2}, \angle 0^\circ$	$X_m/\sqrt{2}, \angle -90^\circ$	$X_m/\sqrt{2}, \angle 36^\circ$	$X_m/\sqrt{2}, \angle -54^\circ$
k	2	0.200000	$X_m/\sqrt{2}, \angle 0^\circ$	$X_m/\sqrt{2}, \angle -90^\circ$	$X_m/\sqrt{2}, \angle 72^\circ$	$X_m/\sqrt{2}, \angle -18^\circ$
k	3	0.300000	$X_m/\sqrt{2}, \angle 0^\circ$	$X_m/\sqrt{2}, \angle -90^\circ$	$X_m/\sqrt{2}, \angle 108^\circ$	$X_m/\sqrt{2}, \angle 18^\circ$
k	4	0.400000	$X_m/\sqrt{2}, \angle 0^\circ$	$X_m/\sqrt{2}, \angle -90^\circ$	$X_m/\sqrt{2}, \angle 144^\circ$	$X_m/\sqrt{2}, \angle 54^\circ$
k	5	0.500000	$X_m/\sqrt{2}, \angle 0^\circ$	$X_m/\sqrt{2}, \angle -90^\circ$	$X_m/\sqrt{2}, \angle 180^\circ$	$X_m/\sqrt{2}, \angle 90^\circ$
k	6	0.600000	$X_m/\sqrt{2}, \angle 0^\circ$	$X_m/\sqrt{2}, \angle -90^\circ$	$X_m/\sqrt{2}, \angle -144^\circ$	$X_m/\sqrt{2}, \angle 126^\circ$
k	7	0.700000	$X_m/\sqrt{2}, \angle 0^\circ$	$X_m/\sqrt{2}, \angle -90^\circ$	$X_m/\sqrt{2}, \angle -108^\circ$	$X_m/\sqrt{2}, \angle 162^\circ$
k	8	0.800000	$X_m/\sqrt{2}, \angle 0^\circ$	$X_m/\sqrt{2}, \angle -90^\circ$	$X_m/\sqrt{2}, \angle -72^\circ$	$X_m/\sqrt{2}, \angle -162^\circ$
k	9	0.900000	$X_m/\sqrt{2}, \angle 0^\circ$	$X_m/\sqrt{2}, \angle -90^\circ$	$X_m/\sqrt{2}, \angle -36^\circ$	$X_m/\sqrt{2}, \angle -126^\circ$
k+1	0	0.000000	$X_m/\sqrt{2}, \angle 0^\circ$	$X_m/\sqrt{2}, \angle -90^\circ$	$X_m/\sqrt{2}, \angle 0^\circ$	$X_m/\sqrt{2}, \angle -90^\circ$

B.5.1.4 RESPUESTA EN EL TIEMPO DEL PMU

La respuesta del tiempo del PMU será medido aplicando $\pm 10\%$ de la magnitud de la señal de entrada en condiciones de magnitud y frecuencia nominal. La respuesta en el tiempo es el intervalo de tiempo entre el instante en que el cambio de paso es aplicado y la etiqueta de tiempo de la primera medición del fasor para la cual el TVE entra y se mantiene de los límites especificados de precisión correspondiente al nivel de conformidad (1%). [13]

B.5.2 EVALUACIÓN DE LA MEDICIÓN

B.5.2.1 EVALUACIÓN DE LA MEDICIÓN DEL SINCRUFASOR

Los valores teóricos de la representación de un sincrofasor de una señal sinusoidal y los valores obtenidos de la PMU pueden incluir diferencias entre la amplitud y la fase. Mientras estos pueden

ser especificados por separado, las diferencias de amplitud y fase son consideradas juntas en este estándar en un valor llamado Vector Total de Error (TVE). El TVE es una expresión de la diferencia entre una muestra “perfecta” de un sincrofasor teórico y la estimación dada por la unidad bajo prueba al mismo instante de tiempo. El valor es normalizado y expresado en por unidad del valor teórico. [48] El TVE está definido por la ecuación (B.2):

$$TVE(n) = \sqrt{\frac{(\hat{X}_r(n) - X_r(n))^2 + (\hat{X}_i(n) - X_i(n))^2}{X_r(n)^2 + X_i(n)^2}} \quad (B.2)$$

En donde:

$\hat{X}_r(n), \hat{X}_i(n)$ son los valores medidos, proporcionados por el elemento de medición.

$X_r(n), X_i(n)$ son los valores teóricos de la señal de entrada al instante de tiempo de la medición,

Los valores $X_r(n), X_i(n)$ pueden ser determinados de manera cercana en situaciones bien definidas, talos como frecuencias o fases constantes, y estas situaciones son exploradas en el estándar. [48]

Cuando la frecuencia de la señal de entrada es exactamente igual a la nominal, el fasor será invariante en el tiempo y sus coordenadas serán fijadas y determinadas por la señal de fase. Cuando la frecuencia de la señal de entrada no es exactamente igual a la nominal, el fasor será sometido a una rotación en el plano complejo como se muestra en la tabla B.2. El TVE es la diferencia de la magnitud del vector entre el fasor teórico definido en la ecuación B.1 y el fasor estimado dado por el dispositivo de medición, expresado como una fracción de la magnitud del fasor teórico como se muestra en la ecuación B.2. [13]

B.5.3 VERIFICACIÓN DE CUMPLIMIENTO

B.5.3.1 INTRODUCCIÓN

Estos requerimientos deben conocerse para todos los tiempos y bajo todas las configuraciones donde la función de la PMU es una unidad física por separado o incluida como parte de una unidad de multifunciones. [48]

B.5.3.2 CLASES DE DESEMPEÑO

Este estándar define dos clases de desempeño: Clase P y Clase M. [48]

La Clase P es intencionada para aplicaciones que requieran rápida respuesta y asignaciones sin filtrado explícito. La letra P es usada desde que las aplicaciones de protección requieren una respuesta rápida. [48]

La Clase M es intencionada para aplicaciones que pueden ser negativamente afectadas por señales ‘alias’ y no requieren una respuesta de reporte rápida. La letra M es usada desde que las mediciones analíticas a menudo requieren gran precisión pero no requieren un retraso mínimo de reporte. [48]

Sin embargo, esta designación de clases no indica cual clase es adecuada o requerida para una aplicación en particular. El usuario debe elegir la clase de desempeño que se ajuste a los requerimientos de cada aplicación. [48]

B.5.3.3 VERIFICACIÓN DEL CUMPLIMIENTO

Pruebas de cumplimiento deben ser ejecutadas para comparar las estimaciones de fasor obtenidas bajo condiciones de estado estable contra los correspondientes valores teóricos de X_r y X_i y calculando el TVE, como se define en la ecuación B.2. [48]

Todas las pruebas de cumplimiento realizadas bajo condiciones de estado estacionario, con condiciones de referencia y magnitudes de influencia mostradas en la tabla B.3. Los efectos de dichas magnitudes deben ser consideradas acumulativas, y el TVE no debe exceder el error enlistado por los niveles de cumplimiento dados bajo cualquier combinación de magnitudes de influencia mostradas en la tabla B.3. Para evaluar el cumplimiento con estos requerimientos, el efecto de las magnitudes de influencia puede ser evaluado por separado. [48]

B.5.3.4 CONDICIONES DE PRUEBA Y REFERENCIA

Rodas las pruebas deben ser ejecutadas con todos los parámetros fijados a condiciones de referencia, excepto aquellos que varían como lo especifique la prueba. La condición de referencia es el valor de la cantidad que está siendo probada cuando no está siendo variada. Sólo los

parámetros especificados para cada requerimiento deben ser variados para ser consideradas independientes. Las condiciones de referencia para todas las pruebas son: [48]

1. Voltaje a Magnitud Nominal.
2. Corriente a Magnitud Nominal.
3. Frecuencia a Magnitud Nominal.
4. Voltaje, corriente, fase y frecuencia constantes.
5. Señal de distorsión armónica total (THD) menor a 0.2% de la fundamental.
6. Todas las señales de interferencia menores al 0.2% de la fundamental.

Mediciones con razones de reporte menores a 10/s NO deben ser sujetas a los requerimientos de desempeño dinámico. Dichas mediciones deben ser objeto de todos los requerimientos de estado estacionario, excepto el rechazo de fuera de banda.

A menos que se especifique lo contrario, todas las pruebas para certificar el cumplimiento deben ser realizadas en condiciones estándar de un laboratorio de pruebas incluyendo:

1. Temperatura: $23^{\circ}\text{C} \pm 3^{\circ}\text{C}$.
2. Humedad menor a 90%.

A menos que se especifique lo contrario, el TVE, FE y RFE para cada requerimiento de desempeño deben ser el promedio, el valor RMS o el máximo valor observado a un mínimo de 5 segundos de la duración de la prueba.

En las siguientes subcláusulas, f_{in} es la frecuencia de la componente fundamental de la señal. Es normalmente 50 Hz o 60 Hz, pero en el curso de las pruebas puede ser variada de la nominal. Así, f_0 siempre representa la frecuencia nominal, exactamente 50 Hz o 60 Hz. De manera similar, $\omega_0 = 2\pi f_0$ siempre representa la frecuencia nominal en radianes/s.

B.5.3.5 CUMPLIMIENTO EN ESTADO ESTACIONARIO

Debe ser confirmado comparando las estimaciones de sincrofasor, frecuencia y ROCOF obtenidas en estado estacionario con los correspondientes valores teóricos. Las condiciones de estado estacionario son cuando X_m , ω y φ de la señal de prueba, y todas las otras cantidades de influencia

son fijadas para un periodo de medición. Se debe notar que para frecuencia fuera de la nominal, el ángulo de medición cambiará aunque la fase de la señal de prueba φ sea constante.

TABLA B.3 REQUERIMIENTOS DE MEDICIONES DEL SINCRUFASOR EN ESTADO ESTACIONARIO

Magnitud de Influencia	Condición de Referencia	Mínimo rango de magnitud de influencia sobre el cual cada PMU debe estar con el TVE limite dado.			
		Clase P		Clase M	
		Rango	Max TVE (%)	Rango	Max TVE (%)
Rango de Frecuencia de la Señal- f_{dev} (nominal de la prueba + desviación: $f_0 \pm f_{dev}$)	$F_{nominal}(f_0)$	± 2.0 Hz	1	± 2.0 Hz para $F_s < 10$ $\pm F_s/5$ para $10 \leq F_s < 25$ ± 5.0 Hz para $25 \leq F_s$	1
Magnitud de la Señal - Voltaje	100% medida	80% al 120% medida	1	10% al 120% medida	1
Magnitud de la Señal - Corriente	100% medida	10% al 200% media	1	10% al 200% medida	1
Ángulo de Fase con $ f_{in} - f_0 < 0.25$ Hz	Constante o Ángulo Lentamente Variante	$\pm \pi$ radianes	1	$\pm \pi$ radianes	1
Distorsión Armónica (armónico simple)	<0.2% (THD)	1%, cualquier armónico, hasta el 50vo	1	10%, cualquier armónico, hasta el 50vo	1
Señal de interferencia fuera de banda.	<0.2% de la magnitud de la señal de entrada	-	Ninguno	10% de la magnitud de la señal de entrada para $F_s \geq 10$	1

B.6 FORMATO DE MENSAJES DEL SINCRUFASOR

B.6.1 SOLICITUD DE MENSAJES

Esta subcláusula describe el formato de mensajes hacia y del PMU para uno en comunicación en tiempo real de datos de un fasor. La transmisión de información en tiempo real se refiere a que tiene lugar simultáneamente con el proceso de medición. Si el dispositivo PMU se tiene para utilizarse con otros sistemas donde la información del fasor de datos es transmitida en tiempo real, la implementación de este protocolo es requerida conforme a este estándar. Si el dispositivo PMU es usado solo como concentrador de los datos del fasor, este protocolo no es requerido. La

implementación de protocolos adicionales para la comunicación de datos fasoriales no está limitada. [49]

Cualquier medio o sistema de comunicación puede ser utilizado para la transmisión de datos. Las estructuras de los mensajes deberán ser transmitidas por completo como sean especificadas. Este protocolo de mensajes puede ser utilizado para comunicarse con un solo PMU o con un sistema secundario que recibe información de varios PMU's. El sistema secundario hace referencia a un concentrador de datos, que deberá tener su propio código ID de usuario asignado. El protocolo prevé información de identificación necesaria, tal como el código ID del PMU y su estatus, para ser incrustado en la trama de datos para la apropiada interpretación de los datos medidos. [13]

B.6.2 ESTRUCTURA DE MENSAJES

Cuatro tipos de mensaje son definidos: datos, configuración, encabezado y comando. Los primeros tres son transmitidos por el PMU's y el último es recibido por el mismo. El mensaje de datos son las mediciones hechas por el PMU. La configuración, mensaje que entendible para la máquina, describe los datos que el PMU envía y proporciona los factores de calibración. El encabezado es información legible para el humano que describe la información enviada por el PMU pero proporcionada por el usuario. Los comandos son códigos enviados para el PMU por el control o configuración. La información puede ser almacenada por el PMU, pero cuando es transmitida debe ser formateada en tramas descritas en la siguientes subcláusula. [49]

Solamente la trama de datos, configuración, encabezado y comandos son definidas en este estándar. Otros tipos de tramas pueden ser descritos en un futuro. En operación normal, el PMU solo envía tramas de datos. [49]

B.6.2.1 MENSAJE TOTAL

Todas las tramas de mensaje comienzan con la palabra "SYNC" de 2 bytes seguida por la palabra "FRAMESIZE" de 2 bytes, un "IDCODE" de 2 bytes, y un estampado de tiempo "SOC" que consiste de 4 bytes y la fracción de segundo "FRACSEC" de 4 bytes, que incluye un entero de 24 bits como fracción de segundo y una bandera de calidad de tiempo de 8 bits. La palabra "SYNC" proporciona la sincronización y la identificación de la trama. El "IDCODE" identifica afirmativamente la unidad que envió o recibió el mensaje. Los bits 4-6 de la palabra "SYNC"

designan el tipo de trama. Esta palabra está detallada en la tabla B.4. Todas las tramas terminan con la palabra de chequeo “CHK” que es un CRC-CCITT (Comprobación de Redundancia Cíclica de 16 bits). Este CCR-CCITT usa la generación del polinomio $x^{16} + x^{12} + x^5 + 1$, el cual se inicializa con el valor de -1 (FFFF, hexadecimal). La figura B.3 muestra el orden de la trama de transmisión. La palabra “SYNC” es transmitida primero y la última palabra en transmitirse es “CHK”. Todos los tipos de tramas llevan este mismo orden y formato. Esta primera edición publicada del estándar deberá ser designada como la versión 1 (0001, binario). [13]

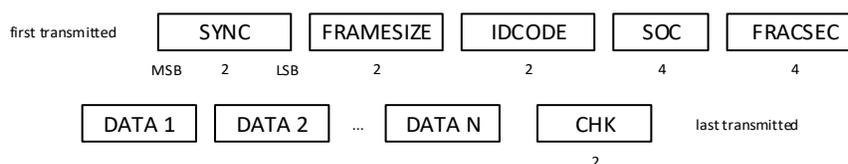


FIGURA B.3 EJEMPLO DEL ORDEN DE LA TRAMA DE TRANSMISIÓN.

TABLA B.4 DEFINICIONES COMUNES DE PALABRAS PARA TODO TIPO DE TRAMAS

Campo	Tamaño (Bytes)	Comentarios
SYNC	2	Palabra de sincronización de trama.
FRAMESIZE	2	Número total de bytes en la trama, incluyendo CHK. Número no signado de 16 bits. Tamaño máximo de 65565.
IDCODE	2	Número de identificación del PMU (PMU/DC ID) asignado por el usuario. Entero de 16 bits, del 1 al 65534 (0 y 65535 reservados).
SOC	4	Estampa de tiempo. Número no signado de 32 bits. Los saltos de segundo no están incluidos en la cuenta, por lo que cada año tiene el mismo número de segundos excepto años bisiestos, que tienen un día extra (86400 segundos).
FRACSEC	4	Fracción de segundo y Calidad de tiempo, tiempo de medición para tramas de datos o tiempo de transmisión de trama para tramas sin datos.
CHK	2	CRC-CCITT. Entero no signado de 16 bits.

B.6.2.2 TIEMPO Y CALIDAD DE TIEMPO

El “FRACSEC” de 32 bits (4 bytes) es dividido en dos componentes: uno de 24 bits que es la actual fracción de segundo y uno de 8 bits que es la bandera de la calidad de tiempo. El tiempo de medición o transmisión de datos para tramas sin datos es el estampado de tiempo “SOC” que fija el segundo entero más el tiempo fraccionario. El tiempo fraccionario es determinado dividiendo el entero de 24 bits de la fracción de segundo entre el entero “TIME_BASE” proporcionado en la configuración de la trama, como se muestra en la ecuación B.3. [13]

$$tiempo = SOC + \frac{fracción_de_segundo}{TIME_BASE} \quad (B.3)$$

Los bits de la calidad de tiempo indican la “calidad” del tiempo que está siendo reportado así como indicar el estado del salto de segundo. La tabla B.5 detalla estas asignaciones. El bit 7 es reservado para usos futuros. El bit 4 es el salto de segundo pendiente y debe ser fijado tan pronto como éste se conozca pero no menos que un segundo o más de 59 segundos después de que el salto de segundo ocurra. El bit 5 es el salto de segundo ocurrido y deberá ser fijado en el primer segundo después de que el salto de segundo ocurre. El bit 6 es la dirección del saldo de segundo, que es 0 para agregar un segundo y 1 para borrar un segundo. Este deberá ser fijado (a 0 o 1 como se requiera) en el mismo tiempo o después del pendiente salto de segundo es fijado y seguirá siendo el mismo por lo menos después de 24 horas. Esto permitirá a los programas de análisis para un factor en un ± salto de segundo en cualquier análisis o diferencia de tiempo calculado. [13]

TABLA B.5 DEFINICIÓN DE LOS BITS DE LA BANDERA DE CALIDAD DE TIEMPO

Bit #	Descripción
7	Reservada.
6	Dirección del salto de segundo. 0 para agregar, 1 para borrar.
5	Ocurrencia del salto de segundo. Se coloca en el primer segundo después que el salto de segundo ocurre y se mantiene por 24 horas.
4	Pendiente el salto de segundo. Coloca después de que el salto de segundo ocurre y se limpia en el segundo después de que el salto ocurre.
3-0	Código indicador de la calidad de tiempo.

El contenido del código de la calidad de tiempo indica en los 4 bits menos significativos el error máximo de tiempo como lo determina la función de reloj del PMU. Los bits 0-3 deben ser limpiados a 0 cuando la función del tiempo es cerrada encima de su fuente. Los bits 0-3 deben ser fijados a 1 cuando hay cualquier error en el reloj o cuando nunca se ha inicializado el mismo. [13]

B.6.3 TRAMA DE DATOS

La trama de datos deberá contener los datos de la medición y deberá ser identificada por tener los bits 4-6 en la palabra “SYNC” fijados en 0, como se indica en la tabla B.4. La trama de datos del fasor en tiempo real consistirá en datos binarios ordenados y descritos a detalle como se muestra en la tabla B.6. La longitud de todos los campos deberá ser fijada como se describe, y no se limitará

Diseño de una Unidad de Medición Fasorial con tecnología ARM

su uso. La trama empieza con “SYNC”, “FRAMESIZE”, “IDCODE” y “SOC” y termina con un CRC-CCITT.

TABLA B.6 ORGANIZACIÓN DE LA TRAMA DE DATOS

No.	Campo	Tamaño (Bytes)	Comentario
1	SYNC	2	Byte de sincronización seguido por el tipo de trama y número de versión.
2	FRAMESIZE	2	Número de bytes en la trama, definido en B.6.2.
3	IDCODE	2	Número de PMU, entero de 16 bits, definido en B.6.2.
4	SOC	4	Estampa de tiempo SOC, definida en B.6.2.
5	FRACSEC	4	Frac. de segundo y Calidad de tiempo, definida en B.6.2.
6	STAT	2	Banderas del mapa de bits.
7	PHASORS	4 x PHNMR 8 x PHNMR	Estimación del fasor. Puede ser un fasor simple o 3 fasores de secuencia positiva negativa y cero. El valor de 4 u 8 PHNMR (número de fasores, entero de 2 bytes) depende del formato usado de 16 bits o punto flotante.
8	FREQ	2 / 4	Frecuencia (arreglado o punto flotante).
9	DFREQ	2 / 4	Rango de cambio de la frecuencia (arreglado o punto flotante).
10	ANALOG	2 x ANNMR 4 x ANNMR	Datos analógicos, de 2 o 4 bytes según el formato de datos usados. (ANNMR, número de valores analógicos, entero de 2 bytes).
11	DIGITAL	2 x DGNMR	Datos digitales, usualmente representando 16 puntos digitales de estado (canales). (DGNMR, número de palabras de estado digital, entero de 2 bytes).
	Repetir 6-11		Campos 6-11 se repiten tanto como cuantos PMU's en el campo NUM_PMU haya en la configuración de la trama.
12+	CHK	2	CRC-CCITT.