



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SECCIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

**“DISEÑO E IMPLEMENTACIÓN DE UN SINCRONOSCOPIO PARA
MÁQUINAS SÍNCRONAS”**

TESIS

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN CIENCIAS CON ESPECIALIDAD

EN INGENIERÍA ELÉCTRICA

PRESENTA:

ING. ALEJANDRO VILLAVICENCIO ACOSTA.

Dr. Raúl Ángel Cortés Mateos.

Director de Tesis.



MÉXICO D.F. 2008



INSTITUTO POLITECNICO NACIONAL

SECRETARIA DE INVESTIGACION Y POSGRADO

ACTA DE REVISION DE TESIS

En la Ciudad de México, D. F. siendo las 19:00 horas del día 16 del mes de Junio del 2008 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 grado titulada:

“DISEÑO E IMPLEMENTACIÓN DE UN SINCRONOSCOPIO PARA MÁQUINAS SÍNCRONAS”

Presentada por el alumno:

VILLAVICENCIO

Apellido paterno

ACOSTA

materno

ALEJANDRO

nombre(s)

Con registro:

A	0	6	0	1	2	3
---	---	---	---	---	---	---

aspirante al grado de:

MAESTRO EN CIENCIAS CON ESPECIALIDAD EN INGENIERÍA ELÉCTRICA

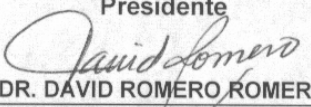
Después de intercambiar opiniones los miembros de la Comisión manifestaron **SU APROBACION DE LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Director de tesis


DR. RAÚL ÁNGEL CORTÉS MATEOS

Presidente


DR. DAVID ROMERO ROMERO

Segundo Vocal


DR. JAIME JOSÉ RODRIGUEZ RIVAS

Tercer Vocal


DR. DOMITILO LIBREROS

Secretario


DR. DAVID SEBASTIÁN BALTAZAR

Suplente


M. en C. TOMÁS IGNACIO ASIAÍN OLIVARES

EL PRESIDENTE DEL COLEGIO


DR. JAIME ROBLES GARCÍA






INSTITUTO POLITECNICO NACIONAL
COORDINACION GENERAL DE POSGRADO E INVESTIGACION

CARTA CESION DE DERECHOS

En la Ciudad de México, Distrito Federal, el día 20 del mes Junio del año 2008, el que suscribe **Ing. Alejandro Villavicencio Acosta** alumno del Programa de Maestría en Ciencias con especialidad en Ingeniería Eléctrica con número de registro A060123 adscrito a la Sección de Estudios de Posgrado e Investigación de la ESIME Unidad profesional "Adolfo López Mateos" Zacatenco, manifiesta que es autor intelectual del presente Trabajo de Tesis bajo la dirección del **Dr. Raúl Ángel Cortés Mateos** y cede los derechos del trabajo intitulado: "**Diseño e Implementación de un Sincronoscopio para Máquinas Síncronas**", al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, graficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección: villavicencioacostaa@hotmail.com; rcortes@ipn.mx.

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


Alejandro Villavicencio Acosta.
Nombre y firma.

Resumen

En este trabajo de tesis, se presenta el diseño e implementación de un sincronoscopio para una máquina síncrona, mediante un microcontrolador de 32 bits.

El equipo diseñado e implementado, tiene la capacidad de medir los niveles de voltaje de fase de un sistema trifásico en el intervalo de 30 a 150 Volts, de determinar la secuencia de fases, de medir la frecuencia en el intervalo de 25 a 75 Hz, tanto del sistema motor-generador como las de la compañía suministradora de energía eléctrica.

Los fasores de ambos sistemas, son procesados mediante un algoritmo de control y una máquina de estados, para sincronizar el sistema motor-generador con la compañía suministradora de energía eléctrica, de acuerdo con las normas establecidas para dicho fin.

Después de sincronizar la salida del sistema motor-generador con la compañía suministradora de energía eléctrica, es posible suministrar potencia.

Las mediciones obtenidas y el estado del sistema, son enviados por el microcontrolador a una PC, mediante un puerto de comunicación serial RS-232.

El monitoreo y visualización de forma gráfica en tiempo real se lleva a cabo en el ambiente de depuración del microcontrolador vía USB en la PC; permitiendo visualizar el desplazamiento de los fasores, frecuencia, niveles de voltajes, secuencia de fases, estado en el que se encuentra el sincronoscopio y comandos visuales de ajuste de parámetros del sistema motor-generador para sincronizar con la compañía suministradora de energía eléctrica.

Abstract

This thesis presents the design and implementation of an automatic synchronizer for power generator, using a 32-bits microcontroller.

The prototype implemented, has the ability to measure phase-angle, voltage levels of a three-phase system in the range of 30 to 150 Volts, frequency in the range of 25 to 75 Hz, in the motor-generator system and the company providing electric power (CFE).

The phase angle in both systems, are processed with an algorithm and a state machine that synchronize the motor-generator system with CFE, according to the established standards for that purpose.

The debugger “visualization tool” monitoring and graphics variables in real time through USB in the PC; also visualize the phase movement, frequency, voltage and executes commands to synchronize.

The measurements and system status obtained are sent by the microcontroller through a serial communication port RS-232 to PC.

*A Dios por darme los medios para
alcanzar una meta más en mi vida.*

*A mi familia, especialmente
a mi mamá Ana; por todo su apoyo y cariño,
por motivarme a seguir adelante y por
estar siempre pendiente de mí.
¡Gracias!.*

*Quiero agradecer al director de la tesis Dr. Raúl Ángel Cortés Mateos
por su guía, valiosas sugerencias, apoyo moral, consejos, enseñanzas, y sobre todo
mucho paciencia para la realización exitosa de esta tesis. Gracias por ver en mí al
ser humano más que al alumno.*

*A todos mis amigos y compañeros que me han honrado con su amistad y cariño,
porque siempre confiaron en que podía realizar mis metas.*

*Finalmente, agradezco a las valiosas observaciones, comentarios y sugerencias de la
comisión revisora de este trabajo, quienes ayudaron a mejorar la calidad del mismo:
Dr. David Romero Romero, Dr. Jaime José Rodríguez Ribas,
Dr. Raúl Ángel Cortés Mateos, Dr. David Sebastián Baltazar,
Dr. Domitilo Libreros y M. en C. Tomás Ignacio Asiaín Olivares.*

*Este trabajo fue realizado gracias al apoyo económico brindado por el
Instituto Politécnico Nacional (IPN).*

TABLA DE CONTENIDO.

RESUMEN.....	IV
ABSTRACT	V
LISTA DE FIGURAS	X
LISTA DE TABLAS	XIV
GLOSARIO	XV
<u>CAPÍTULO 1</u>	18
INTRODUCCIÓN	18
1.1 GENERALIDADES.	18
1.2 OBJETIVO DE LA TESIS.	19
1.3 JUSTIFICACIÓN.....	20
1.4 ESTADO DEL ARTE.	20
1.5 APORTACIONES.....	21
1.6 ESTRUCTURA DEL TRABAJO DE TESIS.....	21
<u>CAPÍTULO 2</u>	23
GENERACION DE POTENCIA ELECTRICA Y SINCRONIZACION	23
2.1 MÁQUINAS SÍNCRONAS.....	23
2.2 TIPOS DE PLANTAS DE GENERACIÓN.....	24
2.2.1 <i>Plantas de energía eléctrica base</i>	24
2.2.2 <i>Plantas de energía eléctrica intermedia</i>	24
2.2.3 <i>Plantas de generación pico</i>	24
2.3 CONCEPTO DE SINCRONIZACIÓN.....	24
2.3.1 <i>Beneficios</i>	26
2.3.2 <i>Requisitos para sincronizar</i>	26
<u>CAPÍTULO 3</u>	27
DISEÑO DEL SINCRONOSCOPIO HARDWARE	27
3.1 ESQUEMAS DE APLICACIÓN DE SINCRONIZACIÓN.....	27
3.2 DIAGRAMA HARDWARE/SOFTWARE.....	29
3.3 TARJETA DE DESARROLLO DEMOQUE DE FREESCALE.....	31
3.4 DIAGRAMA UNIFILAR DEL SINCRONOSCOPIO.....	33
3.4.1 <i>Diagrama unifilar: CTPE-V4,y etapa de potencia</i>	33
3.4.2 <i>Diagrama unifilar: CTPE-V4,y DEMOQUE</i>	34
3.5 ARQUITECTURA DEL MICROCONTROLADOR MCF51QE128.....	35
3.5.1 <i>Características de la Unidad Central de Procesamiento (CPU) de 32 bits ColdFire Versión 1</i>	35
3.6 CONVERSION ANALÓGICO/DIGITAL.....	36
3.6.1 <i>Método de aproximaciones sucesivas</i>	36
3.6.2 <i>Características del modulo ADC</i>	37
3.6.3 <i>Descripción funcional de operación del modulo ADC</i>	38
3.7 DISEÑO DE LOS ACONDICIONADORES DE SEÑALES.....	39
3.7.1 <i>Especificaciones Eléctricas</i>	39
3.7.2 <i>Entradas Digitales</i>	40
3.7.3 <i>Relevadores</i>	40
3.7.4 <i>Acondicionamiento de señales de voltaje de CA</i>	40
3.7.5 <i>Acondicionamiento de señales de corriente alterna</i>	43
3.8 SISTEMA MOTOR-GENERADOR.....	46

3.8.1	Máquina síncrona trifásica, DL1026A de “DE LORENZO”	46
3.8.2	Motor de CD, DL1023PS de “DE LORENZO”	47
3.9	CONTACTOR TETRAPOLAR, A9-30-10 DE “ABB”	48
CAPÍTULO 4		49
DISEÑO DEL SOFTWARE		49
4.1	DIAGRAMA DE ESTADOS	49
4.1.1	Definiciones de los estados	52
4.1.2	Definición de las condiciones de cambio entre estados del Sincronizador	54
4.2	SOFTWARE DEL MICROCONTROLADOR DE 32 BITS FLEXIS V1 COLDFIRE	55
4.2.1	Configuración Inicial del microcontrolador	56
4.2.2	Configuración del “TimerInt” para indicar el inicio de conversiones del ADC	57
4.2.3	Configuración del “TimerInt” para el parpadeo de LED’s en el ambiente de depuración y decremento de contadores	59
4.2.4	Configuración del Convertidor Analógico a Digital	60
4.3	DIAGRAMA DE FLUJO DEL PROGRAMA PRINCIPAL (MAIN)	62
4.4	ALGORITMO PARA EL CÁLCULO DE LA DFT EN TIEMPO REAL	67
4.5	METODOLOGÍA DEL CÁLCULO DE LA SECUENCIA DE FASE	71
4.6	METODOLOGÍA DEL CÁLCULO DE LA FRECUENCIA EN TIEMPO REAL	72
4.7	AMBIENTE DE DEPURACIÓN DEL MICROCONTROLADOR	76
4.7.1	Inicialización	76
4.7.2	Vista final del diseño del ambiente de depuración	82
CAPÍTULO 5		83
PRUEBAS		83
5.1	SECUENCIA DE PRUEBAS DEL SINCRONOSCOPIO	83
5.1.1	Sincronización con secuencias de fases positivas	84
5.1.2	Sincronización con secuencias de fases negativas	95
5.1.3	Respuesta ante secuencia de fases opuestas	98
5.1.4	Sacar de sincronía al sistema motor-generator con comandos generados en el IDE	99
CAPÍTULO 6		101
CONCLUSIONES Y RECOMENDACIONES PARA TRABAJOS FUTUROS		101
6.1	CONCLUSIONES	101
6.2	APORTACIONES DE LA TESIS	101
6.3	RECOMENDACIONES PARA TRABAJOS FUTUROS	102
REFERENCIAS		103
APENDICES		106
A.	PROGRAMAS	106
A.1	Programas de Code Warrior	106
A.1.1	Principal (Main)	106
A.1.2	Estados.c	108
A.1.3	Condiciones.c	110
A.1.4	DFT.c	112
A.1.5	secuencia_fase.c	115
A.1.6	Frecuencimetro.c	117
A.1.7	Comandos_sincronoscopio.c	119
A.1.8	Act_salidas.c	121
A.2	Events.c	122

A.3	<i>Ambiente de depuración VTL</i>	124
A.3.1	<i>VTL.c</i>	124
A.3.2	<i>CFE_GEN_VTL.c</i>	126
A.3.3	<i>LEDs_Sincronoscopio_VTL.c</i>	128
A.3.4	<i>LEDs_comandos_VTL.c</i>	130
A.3.5	<i>LEDs_Secuencia_Fase_VTL.c</i>	131
A.3.6	<i>LEDs_Edos_VTL.c</i>	132
B.	ANALISIS DE FOURIER	133
B.1.	<i>Concepto de fase</i>	133
B.2	<i>Muestreo digital</i>	133
B.3	<i>Teorema de muestreo</i>	134
B.4	<i>Series y Transformada de Fourier</i>	135
B.4.1	<i>Introducción</i>	135
B.4.2	<i>Series de Fourier</i>	135
B.4.3	<i>Series complejas de Fourier</i>	136
B.4.4	<i>Transformada de Fourier</i>	137
B.4.4	<i>La Transformada Discreta de Fourier</i>	140

Lista de Figuras

Figura 3.1.- Rasurado de picos (Peak Shaving).....	27
Figura 3.2.- Excesos de demanda abastecidos por CFE.	28
Figura 3.3.- El 100% del suministro de energía eléctrica es con el sistema motor-generador.....	29
Figura 3.4.- Diagrama de bloques Hardware/Software del sincronoscopio.....	29
Figura 3.5.- Diagrama de interconexión de operación del sincronoscopio (Hardware).....	30
Figura 3.6.- Tarjeta de desarrollo DEMOQE.	31
Figura 3.7.- Tira de pines de expansión de la tarjeta de desarrollo DEMOQE128UM.	32
Figura 3.8.- Microcontrolador de 32 bits, MCF51QE128.	32
Figura 3.9.- Diagrama unifilar: CTPE-V4.y etapa de potencia (contactores).....	33
Figura 3.10.- Diagrama unifilar: CTPE-V4 y DEMOQE.	34
Figura 3.11.- Tarjeta CTPE-V4.....	39
Figura 3.12.- Circuito acondicionador de señal de VAC.....	41
Figura 3.13.- Circuito acondicionador de señal de VAC.....	41
Figura 3.14.- Circuito resultante con fuente de CA en corto.....	42
Figura 3.15.- Circuito acondicionador de señal de I AC.....	44
Figura 3.16.- Circuito resultante con fuente de CD en corto.	44
Figura 3.17.- Circuito resultante con fuente de CA en corto.....	45
Figura 3.18.- Sistema motor-generador.	46
Figura 3.19.- Máquina síncrona trifásica DL1026A de “DE LORENZO”.....	47
Figura 3.20.- Motor de CD DL1023PS de “DE LORENZO”.....	47
Figura 3.21.- Contactor tripolar A9-30-10.....	48
Figura 4.1.- Diagrama de estados del control de sincronía.....	49
Figura 4.2.- Bean Inspector del CPU: MCF51QE128LQFP64.....	57
Figura 4.3.- Bean Inspector del “TimerInt” para indicar los inicios de conversión del ADC.....	58
Figura 4.4.- Diagrama de flujo de la interrupción “TimerInt” para indicar los inicios de conversión del ADC.....	58
Figura 4.5.- Bean Inspector del “TimerInt” para el parpadeo de LEDs y decremento de contadores.	59

Figura 4.6.- Diagrama de flujo de la interrupción “TimerInt” para parpadeo de LEDs y decremento de contadores.....	60
Figura 4.7.- Bean Inspector del ADC.....	61
Figura 4.8.- Diagrama de flujo de la interrupción ADC.....	61
Figura 4.9.- Diagrama de flujo del programa principal.....	62
Figura 4.10.- Diagrama de flujo de la función “Estado FUERA / INICIO”	63
Figura 4.11.- Diagrama de flujo de la función “Estado NORMAL”	64
Figura 4.12.- Diagrama de flujo de la función “Estado SINCRONIZANDO”	65
Figura 4.13.- Diagrama de flujo de la función “Estado SINCRONIZADO”	66
Figura 4.14.- Diagrama de flujo de la función “Actualiza salidas”	67
Figura 4.15.- Diagrama de flujo de la función “Calcula DFT, parte inicial”	68
Figura 4.16.- Diagrama de flujo de la función “Calcula DFT, parte final”	69
Figura 4.17.- Diagrama de flujo de la función “raiz”	70
Figura 4.18.- Diagrama de flujo de la función “Fase”	70
Figura 4.19.- Diagrama de flujo de la función “Determina secuencia de fase”.....	71
Figura 4.20.- Señal trifásica de salida del sistema motor-generador al ser activado (30Hz / 40VCA) con secuencia de fases positiva.....	72
Figura 4.21.- Búsqueda de cruces por el “Offset” cada 4 muestras.	73
Figura 4.22.- Búsqueda de cruces por el “Offset” cada 4 muestras.	73
Figura 4.23.- Diagrama de flujo de la función “Frecuencímetro, parte inicial”.....	74
Figura 4.24.- Diagrama de flujo de la función “Frecuencímetro, parte final”	75
Figura 4.25.- Diagrama de flujo del programa principal.....	76
Figura 4.26.- Detalles del programa implementado y estado del microcontrolador.	77
Figura 4.27.- Agregando un componente al ambiente de depuración.....	77
Figura 4.28.- Selección del componente “Visualizationtool” en el ambiente de depuración.	78
Figura 4.29.- Área de trabajo, para agregar las herramientas de monitoreo.....	78
Figura 4.30.- Selección de un instrumento de visualización analógica.	79
Figura 4.31.- Propiedades del instrumento de visualización analógica.....	79
Figura 4.32.- Instrumento de visualización analógica.....	80

Figura 4.33.- Propiedades del instrumento de visualización en ejes coordenados “x-y”.....	80
Figura 4.34.- Instrumento de visualización analógica en sistema de ejes coordenados “x-y”.....	81
Figura 4.35.- Imágenes dentro del ambiente de depuración.....	81
Figura 4.36.- Vista final de los instrumentos de monitoreo en el ambiente de depuración.....	82
Figura 5.1.- Señales presentes del sincronoscopio en el estado “FUERA / INICIO”.....	84
Figura 5.2.- Ambiente de depuración en el estado “FUERA / INICIO”.....	84
Figura 5.3.- Señales presentes del sincronoscopio en el estado “NORMAL”.....	85
Figura 5.4.- Condición 1 (C1) activada; ambiente de depuración en el estado “NORMAL”.....	85
Figura 5.5.- Sincronoscopio en el estado “NORMAL”.....	86
Figura 5.6.- Señal trifásica de salida del sistema motor-generador al ser activado (30Hz / 40VCA) con secuencia de fases positiva.....	86
Figura 5.7.- Condición de “Inicio de sincronización” activada.....	87
Figura 5.8.- Dos ciclo de CFE y uno del sistema motor-generador.....	87
Figura 5.9.- Estado “SINCRONIZANDO” con comandos de ajuste de frecuencia.....	88
Figura 5.10.- Sistema motor-generador con frecuencia en norma.....	88
Figura 5.11.- Diferencia de frecuencia entre CFE y sistema motor- generador en norma.....	89
Figura 5.12.- Estado “SINCRONIZANDO” con comandos de ajuste de voltaje.....	89
Figura 5.13.- Frecuencia y voltaje del sistema motor-generador dentro de norma.....	90
Figura 5.14.- Señal trifásica del sistema motor-generador adelantado.....	90
Figura 5.15.- Comando visual de ajuste de ángulo de fase adelantado.....	91
Figura 5.16.- Señal trifásica del sistema motor-generador atrasado.....	91
Figura 5.17.- Comando visual de ajuste de ángulo de fase atrasado.....	92
Figura 5.18.- Ambiente de depuración en el estado “SINCRONIZADO” con secuencia de fases positivas.....	92
Figura 5.19.- Fases del Bus infinito y sistema motor-generador sincronizados con secuencias de fase positivas.....	93
Figura 5.20.- Sincronoscopio en el estado “SINCRONIZADO”.....	93

Figura 5.21.- Comando visuales de ajuste activados, para desactivar el sistema motor-generador.....	94
Figura 5.22.- Retorno al estado “NORMAL” por activación de la condición 7 (C7).....	94
Figura 5.23.- Señal trifásica de CFE con secuencia de fase negativa.	95
Figura 5.24.- Condición 1 (C1) activada; ambiente de depuración en el estado “NORMAL”	95
Figura 5.25.- Señal trifásica de salida del sistema motor-generador al ser activado (30Hz / 40VCA) con secuencia negativa.	96
Figura 5.26.- Estado “SINCRONIZANDO” con comandos de ajuste de frecuencia.	96
Figura 5.27.- Ambiente de depuración en el estado “SINCRONIZADO” con secuencia de fases negativas.	97
Figura 5.28.- Fases del Bus infinito y sistema motor-generador sincronizados con secuencias de fase negativas.	97
Figura 5.29.- CFE activada con secuencia de fase positiva.....	98
Figura 5.30.- Sistema motor-generador con secuencia de fase negativa.	98
Figura 5.31.- Ambiente de depuración en el estado “SINCRONIZANDO”, indicando un error en las secuencias de fases de los sistema a sincronizar.	99
Figura 5.32.- Selección del componente “Command” en el ambiente de depuración.	100
Figura 5.33.- Componente “Command” en el ambiente de depuración.....	100

Lista de Tablas

Tabla 3.1. Entradas digitales de la tarjeta CTPE-V4.....	40
Tabla 3.2. Relevadores de la tarjeta CTPE-V4.....	40
Tabla 3.3. Entradas analógicas de voltaje de la tarjeta CTPE-V4.....	43
Tabla 3.4.- Entradas analógicas de corriente de la tarjeta CTPE-V4..	46
Tabla 3.5. Características técnicas de la máquina sincrónica trifásica.....	47

Glosario

A.	Amperes.
ADC	Convertidor Analógico a Digital.
ALU	Unidad aritmética lógica.
a_n	Coefficientes para las forma coseno de la serie de Fourier.
$A(w)$	Función de la integral de Fourier.
b_n	Coefficientes para las forma seno de la serie de Fourier.
$B(w)$	Función de la integral de Fourier.
bean	Capsula de un objeto que contiene atributos, métodos y eventos
bean inspector	Ventana de un objeto que permite modificar los atributos de un periférico así como habilitar los métodos y eventos usados por el periférico.
CA	Corriente alterna.
CD	Corriente directa.
cos	Coseno.
c_k	Forma cosenoidal.
c_n	Coefficientes complejos de Fourier.
DFT	Transformada Discreta de Fourier.
f	Frecuencia base.
f_{max}	Frecuencia máxima a muestrear.
$f(x)$	Función de una variable.
$\hat{f}(w)$	Transformada de Fourier.
f_{Δ}	Frecuencia de muestreo
GPIO	Puerto de entrada y/o salida de propósito general.
Hz	Hertz.
i	Coefficiente para puntos de la señal seno y coseno.
I	Magnitud de corriente.
IDE	Ambiente de desarrollo integral.
$\text{Im } \bar{X}$	Amplitud escalada de la forma seno.

$\text{Im } X[]$	Parte imaginaria del dominio de la frecuencia.
j	Número complejo.
k	Coefficiente para cada una de las frecuencias.
k_n	Coefficientes auxiliares en obtención de la serie de Fourier.
M	Magnitud en forma polar.
N	Número de muestras del muestreo digital.
p	Periodo.
PC	Computadora personal.
p. u.	Por unidad.
RAM	Memoria de acceso aleatorio.
$\text{Re } X[]$	Parte real del dominio de la frecuencia.
$\text{Re } \bar{X}[k]$	Amplitud escalada de la forma coseno.
RS-232	Estándar que norma la comunicación serial.
SC0	Puerto Serial 0 del Microcontrolador Flexis V1 ColdFire.
SC1	Puerto Serial 1 del Microcontrolador Flexis V1 ColdFire.
sen	Seno.
s_k	Forma senoidal.
TC	Transformador de corriente.
TP	Transformador de potencial.
VA	Volt - Ampere.
VCA	Voltaje de corriente alterna.
VCD	Voltaje de corriente directa.
V	Magnitud de voltaje.
V	Volts.
W	Watts.
$x(t)$	Serie temporal continua.
$X[]$	Dominio de la frecuencia.
$x[i]$	Señal resultante de la inversa de la transformada discreta de Fourier.
x_r	Serie temporal discreta.
x_0, x_1, x_3, \dots	Componentes de la serie temporal discreta.

Δs	Intervalo de muestreo.
θ	Ángulo de fase.

CAPÍTULO 1

INTRODUCCIÓN

En este capítulo se aborda el problema a resolver; exponiendo las generalidades, objetivos, justificación, estado del arte, aportaciones y estructura de la tesis. Se da una breve reseña general de lo que se conoce, de los elementos más importantes empleados en la solución del problema; se plantean las metas, el alcance y hacia dónde va enfocado el trabajo; se justifica el por qué es importante la realización del trabajo y se describe el problema a resolver; se presentan los antecedentes y lo que se conoce en la actualidad del tema. Finalmente, se describe la estructura del trabajo por capítulos.

1.1 Generalidades.

La energía eléctrica, tal y como se conoce hoy, la generan grandes generadores de corriente alterna instalados en centrales eléctricas; estas a su vez, necesitan otro tipo de energía (mecánica) que contribuye al movimiento del alternador. En muchas ocasiones la demanda de energía eléctrica es tan grande o cuando hay un corte en su suministro, que se hace uso de máquinas eléctricas que suplen este déficit; a estas máquinas se las conoce como grupos electrógenos o de emergencia, las cuales mueven un generador a través de un motor de combustión interna.

Cada vez que se arranca un generador, para interconectarlo con la red o bus infinito se requiere que estén sincronizados; para realizar esta operación de forma automática se necesita un sincronoscopio, que es el responsable de mandar la señal de interconexión, solo después de cumplir con las condiciones de sincronía.[1, 2, 3, 4, 5].

1.2 Objetivo de la tesis.

El objetivo principal de este trabajo, es realizar el Diseño e Implementación de un Sincronoscopio para Maquinas Síncronas, controlado por un microcontrolador de 32 bits. De manera que un sistema motor-generador se sincronice a la compañía suministradora de energía eléctrica y suministre potencia al bus infinito.

Para llevar a cabo lo anterior se utilizó un sistema motor-generador de laboratorio y se desarrollaron las siguientes actividades:

- Procesamiento de señales digitales en tiempo real, en el microcontrolador de 32 bits Flexis ColdFire V1 de Freescale, sin aditamentos mecánicos para monitoreo de secuencia de fases, voltajes, frecuencia y ángulos de fase de:
 - Línea trifásica de la compañía suministradora de energía eléctrica.
 - Salida trifásica del sistema motor-generador.
- Diseño de instrumentación virtual para monitoreo y visualización en el ambiente de depuración del microcontrolador, tales como:
 - Diseño de un sincronoscopio para mostrar la diferencia de fase entre CFE y el sistema motor-generador.
 - Indicadores de secuencias de fases, positiva o negativa.
 - Diagrama de estados para la visualización de las condiciones en las que se encuentra el Sincronoscopio.
- Monitoreo y generación de comandos visuales para:
 - Ajuste de frecuencia, voltaje y diferencia de ángulos de fase del sistema motor-generador de acuerdo a la norma, para efectuar la sincronización con la compañía suministradora de energía eléctrica.
- Transmisión de información a una PC, mediante el protocolo de comunicación estándar recomendado RS-232, de los cálculos obtenidos para su análisis estadístico y depuración.

1.3 Justificación.

- El uso de los sincronizadores es necesario para operar plantas en paralelo, recorte de picos, venta de energía a la compañía suministradora de energía eléctrica, operar en horario punta (autoabastecimiento), disminuir los picos de demanda (peakshaving), ahorros considerables en el costo de la energía y evitar los cobros por consumo excesivos (máxima demanda).
- Actualmente en la SEPI-ESIME-IPN no se dispone de un documento que especifique como implementar un sincronoscopio para sincronizar con la compañía suministradora de energía eléctrica y en el laboratorio de sistemas digitales de la misma SEPI se cuenta con una microred con 3 grupos motor-generator de 1.1KVA.
- Este trabajo describe la metodología de diseño para implementar un sincronoscopio que pueda ser empleado en cualquier aplicación.

1.4 Estado del arte.

En 1988 Yang Yi-han, Shang Guo-cai y Fang Yong-jie del Departamento de Ingeniería Eléctrica del Instituto de Potencia Eléctrica del norte de China, implementaron un tipo de sincronizador para generadores, el “Fast Following Synchronizer”, el cual se implementó en lazo cerrado en el control de la potencia motriz y de excitación de corriente del generador para hacer que su amplitud, frecuencia y ángulo de fase siguiera al del bus del sistema de potencia al que fuese a ser sincronizado. De tal forma que el voltaje a través del contactor que los interconecta en paralelo mantenga un potencial a cero y en cualquier instante poder efectuar dicho procedimiento sin corrientes que dañen al equipo [6].

En 1994 Jianqin Mao, Lishen liu y Yihan Yang desarrollaron estudios en el dominio de la frecuencia para mantener la estabilidad y seguridad de un sistema eléctrico de potencia al momento en que un generador era interconectado de manera súbita.[8]

En el marco legal, publicado en el Diario Oficial de la Federación el 31 de Mayo de 1993, el artículo 72 del Reglamento de la Ley del Servicio Público de Energía Eléctrica, establece que los particulares podrán realizar autoconsumo eléctrico en las modalidades de autoabastecimiento, cogeneración o pequeña producción siempre y cuando dicha energía provenga de plantas destinadas a la satisfacción de las necesidades del conjunto de los copropietarios o socios [7]

1.5 Aportaciones.

- Proporcionar al laboratorio de Sistemas Digitales de la SEPI-ESIME-ZAC un equipo que realice las etapas de sincronización para poder realizar futuras investigaciones.
- Proporcionar diagramas de flujo, plantillas y algoritmos en lenguaje C que puedan ser modificados y/o adecuados a una aplicación específica de:
 - Procesamiento de señales digitales en tiempo real, empleados para el monitoreo y visualización de parámetros de operación de la compañía suministradora de energía eléctrica y del sistema motor-generador.
 - Sincronización del sistema motor-generador con la compañía suministradora de energía eléctrica.

1.6 Estructura del trabajo de tesis.

En este capítulo 1, se aborda el problema a resolver; exponiendo las generalidades, objetivos, justificación, estado del arte, aportaciones y estructura de la tesis. Se da una breve reseña general de lo que se conoce, de los elementos más importantes empleados en la solución del problema; se plantean las metas, el alcance y hacia dónde va enfocado el trabajo; se justifica el por qué es importante la realización del trabajo y se describe el problema a resolver; se presentan los antecedentes y lo que se conoce en la actualidad del tema. Finalmente, se describe la estructura del trabajo por capítulos.

En el Capítulo 2 se detallan las características de la máquina síncrona, se abordan los conceptos relacionados con la sincronización, sus beneficios y se describe el algoritmo matemático de la Transformada Discreta de Fourier para la medición fasorial.

En el Capítulo 3 se presenta el hardware utilizado para la implementación. Los principales elementos utilizados.

- Tarjeta de desarrollo DEMOQE con un microcontrolador de 32 bits Flexis V1 ColdFire de Freescale.
- Tarjeta de adecuación de señales de voltaje y corriente e interfaz de accionamiento para contactores de potencia y
- Sistema motor-generator de “DE LORENZO”, (DL1023PS-DL1026A).

En el Capítulo 4 se presenta el diagrama de estados que seguirá el sincronoscopio así como las condiciones necesarias que le harán cambiar de estado; se describe el procedimiento de inicialización del ambiente de desarrollo integral (IDE) del microcontrolador, el procedimiento de inicialización de periféricos con técnicas de “Procesor Expert”, la configuración de los periféricos e inicialización y diseño del ambiente de depuración.

En el Capítulo 5 se reportan los resultados, en tiempo real, de las pruebas realizadas al sincronoscopio cuando se encuentra conectado al sistema motor-generator. La respuesta del algoritmo de la Transformada Discreta de Fourier (DFT), la respuesta del algoritmo para determinar la frecuencia, determinación de la secuencia de fase conectada, secuencia de prueba con error de secuencia de fases y sincronización con secuencia positiva y negativa.

En el Capítulo 6 se dan las conclusiones de la implementación y las recomendaciones para trabajos futuros con esta línea de estudio.

En la parte final del trabajo se muestran los apéndices; en el apéndice A se incluye la programación realizada en el microcontrolador y en el apéndice B se expone el Análisis de Fourier.

CAPÍTULO 2

GENERACION DE POTENCIA ELECTRICA Y SINCRONIZACION.

En este capítulo se detallan las características de la máquina síncrona, se abordan los conceptos relacionados con la sincronización, sus beneficios y se describe el algoritmo matemático de la Transformada Discreta de Fourier para la medición fasorial.

2.1 Máquinas síncronas.

La máquina síncrona está constituida esencialmente de una parte fija que constituye el inducido llamado también estator y de una parte interna giratoria denominado rotor.

Entre la superficie cilíndrica interna del estator y externa del rotor existe un espacio de aire que se conoce como entrehierro y cuyo espesor puede variar dependiendo el tamaño del generador.

El rotor de la máquina síncrona puede ser de polos salientes o liso, en el primer caso se emplean para máquinas lentas y el segundo para una mayor velocidad. Al inductor con los polos salientes se le conoce también como rueda polar, lleva los polos fijados radialmente sobre la superficie externa de una corona de acero robusta, la cual esta fijada a su vez, según sea el caso, directamente a una flecha o árbol central. Los polos terminan en el entrehierro con la llamada expansión polar. Un generador producirá una corriente eléctrica que cambia de polaridad, este tipo de corriente alterna es ventajosa para la transmisión de energía eléctrica, ya que permite elevar y disminuir el voltaje mediante transformadores, por lo que la mayoría de los generadores eléctricos son de este tipo.

2.2 Tipos de plantas de generación.

2.2.1 Plantas de energía eléctrica base.

Plantas de generación que suministran la energía eléctrica en todo momento. Las plantas nucleares y plantas de carbón están particularmente bien adaptadas para suministrar la demanda base [8]

2.2.2 Plantas de energía eléctrica intermedia.

Plantas de generación que pueden responder relativamente rápido a cambios de demanda, casi siempre agregando o suprimiendo una o mas unidades de generación. las plantas hidroeléctricas son muy adecuadas para este propósito [8]

2.2.3 Plantas de generación pico.

Plantas de generación que suministran potencia durante intervalos breves a lo largo del día: Estas plantas se deben poner en servicio con mucha rapidez. Por consiguiente, están equipadas con motores primarios tales como motores diesel, turbinas de gas, motores de aire comprimido o turbinas de almacenamiento y bombeo que se pueden arrancar en unos cuantos minutos. A este respecto, las plantas de generación térmica que utilizan gas o carbón tardan de 4 a 8 horas en arrancar, mientras que las plantas nucleares pueden tardar varios días [8].

2.3 Concepto de Sincronización.

Es el término empleado cuando se operan dos o más fuentes de CA en paralelo, para alimentar una carga común [9].

- Dos o más generadores entre si.
- Un generador con la compañía suministradora de energía eléctrica.

-
- Dos o más generadores con la compañía suministradora de energía eléctrica.

Un sincronizador automático tiene que comparar la tensión, frecuencia, secuencia de fases y diferencia entre fases de los dos sistemas que se desean sincronizar, permitiendo la conexión en paralelo, únicamente dentro de los límites preestablecidos por las normas en cuestión. Además, tiene que tomar en consideración el tiempo propio del interruptor y eventualmente también el de un relevador auxiliar para cerrar el circuito principal de corriente, en el momento en que exista una coincidencia de parámetros dentro de las tolerancias permitidas por las normas.

La compañía suministradora de energía eléctrica, tiene muchas tarifas basadas en la demanda de carga del cliente. En algunos casos, si se excede un nivel específico de carga, durante ciertas épocas del año ó en horarios pico, las tarifas se pueden incrementar. Una solución para ahorrar dinero y energía es sincronizando el ó los equipos electrógenos a la red comercial de suministro de energía eléctrica [9].

Una planta de emergencia estándar que se desee trabajar para aplicaciones de: cogeneración, operación en horas pico (con alto costo de las tarifas eléctricas), reducción de la demanda máxima, exportar energía a la red normal o simplemente para eliminar los transitorios que se generan en el momento de la retransferencia, se requiere sincronizar con la red comercial de suministro de energía eléctrica, estableciendo la potencia a suministrar por la planta a la carga en forma constante. [10].

En caso de fallar de la red comercial de suministro de energía eléctrica, la(s) planta(s) que conforman el sistema de emergencia, arrancan y se sincronizan automáticamente a un bus de emergencia y después se efectúa la transferencia en forma normal (transición abierta). Durante la operación de los equipos, la carga se reparte automáticamente en forma proporcional, dependiendo de la capacidad de los mismos y se controla el arranque y paro de equipos en función de la demanda de la carga. Una vez que se restablece el suministro de energía eléctrica de la red comercial, se efectúa la retransferencia de la carga, quedando los equipos en operación en vacío para enfriamiento; transcurrido el periodo de enfriamiento, los equipos se paran esperando una nueva señal de arranque.

2.3.1 Beneficios.

Los actuales sistemas de control permiten emplear una planta de emergencia para sincronizarse con la red, obteniendo los siguientes beneficios:

- Disminución de un 50% de las interrupciones a la carga cuando se efectúan las retransferencias.
- Realizar pruebas con carga cuando se requiera, sin provocar interrupciones.
- Operar en horario punta (autoabastecimiento) o en máxima demanda.
- Disminuir los picos de demanda de la carga (Peak Shaving)
- Ahorros considerables en el costo de la energía.
- Evitar los cobros por consumo excesivo (máxima demanda).

2.3.2 Requisitos para sincronizar.

- Tener un gobernador de Frecuencia:

Todos los grupos electrógenos deben estar equipados con un sistema para mantener la frecuencia del generador de salida dentro de límites aceptables. El grado de precisión varía con los requerimientos de la implementación. La selección del gobernador será determinada por los límites de la precisión requerida, las características de la carga, la necesidad de una operación en paralelo, el grado de automatización y el capital disponible para la instalación. [12]

- Tener un regulador de Voltaje:

Contar con un regulador de voltaje para que la salida no varíe en más del 2% con variaciones de carga que van de cero a carga plena [12]

- Mantener una diferencia de voltaje de $\pm 10\%$ [9]
- Mantener una diferencia de Angulo menor a ± 10 grados [9]
- Mantener un deslizamiento de frecuencia menor a 0.1 Hertz [9]
- Compensar el tiempo de cierre del contactor [9]

CAPÍTULO 3

DISEÑO DEL SINCRONOSCOPIO HARDWARE

En este capítulo se presenta el hardware utilizado para la implementación. Los principales elementos utilizados son:

- Tarjeta de desarrollo DEMOQE con un microcontrolador de 32 bits Flexis V1 ColdFire de Freescale.
- Tarjeta CTPE-V4 como adecuador de señales de voltaje y corriente e interfaz de accionamiento para contactores de potencia.
- Sistema motor-generator de “DE LORENZO”, (DL1023PS-DL1026A).

3.1 Esquemas de aplicación de sincronización.

Disminución de la demanda máxima, conocida como “Rasurado de picos” o en inglés denominada “Peak Shaving”, ver figura 3.1.

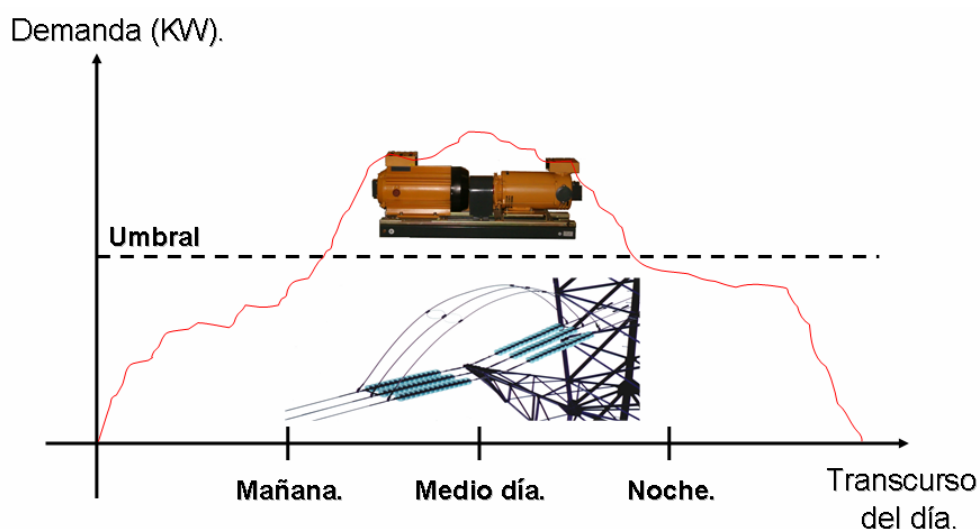


Figura 3.1.- Rasurado de picos (Peak Shaving).

El esquema de la figura 3.1, muestra que rebasado cierto umbral de demanda, es sincronizado un sistema de generación de energía eléctrica, para satisfacer ese déficit. Esta solución permite operar en horario punta (autoabastecimiento) o en máxima demanda, disminuir los picos de demanda de la carga, obtener ahorros considerables en el costo de la energía y evitar los cobros por consumo excesivo (máxima demanda).

Existen otras necesidades, que de manera inversa, requieren que los excesos de demanda sean absorbidos por la compañía suministradora de energía eléctrica, mientras que el sistema de generación abastece de energía eléctrica durante todo el ciclo de demanda de energía eléctrica, ver figura 3.2.

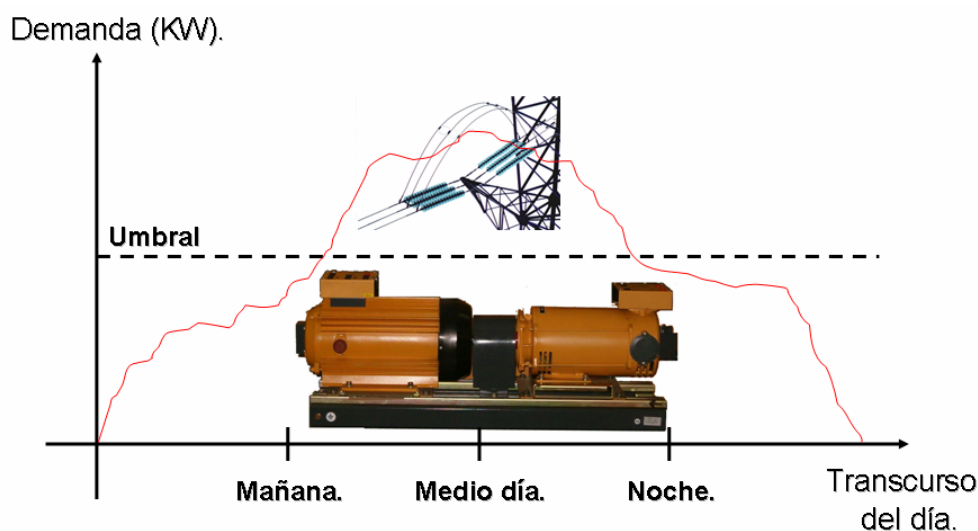


Figura 3.2.- Excesos de demanda abastecidos por CFE.

Otra aplicación es la del abastecimiento del 100% de energía eléctrica con el sistema de generación en horario punta, ver figura 3.3.

Existen aplicaciones de sincronización que requieren la sincronización entre generadores y a su vez con el sistema de suministro de energía eléctrica principal; de igual forma esta serie de generadores sincronizados entre si, pueden integrarse en los esquemas ejemplificados en las figuras 3.1, 3.2 y 3.3.

El software realizado sin modificaciones cubre las aplicaciones de recorte de picos y cogeneración. Para otra aplicación se requiere modificar el diagrama de estados.

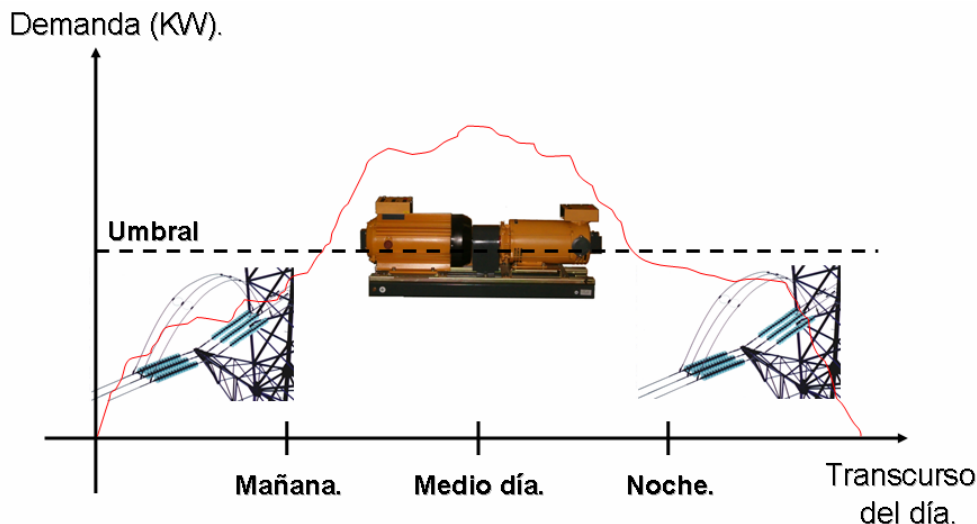


Figura 3.3.- El 100% del suministro de energía eléctrica es con el sistema motor-generator.

3.2 Diagrama Hardware/Software.

En la figura 3.4 se describe en un diagrama Hardware/Software, la metodología de diseño e implementación del sincronoscopio.

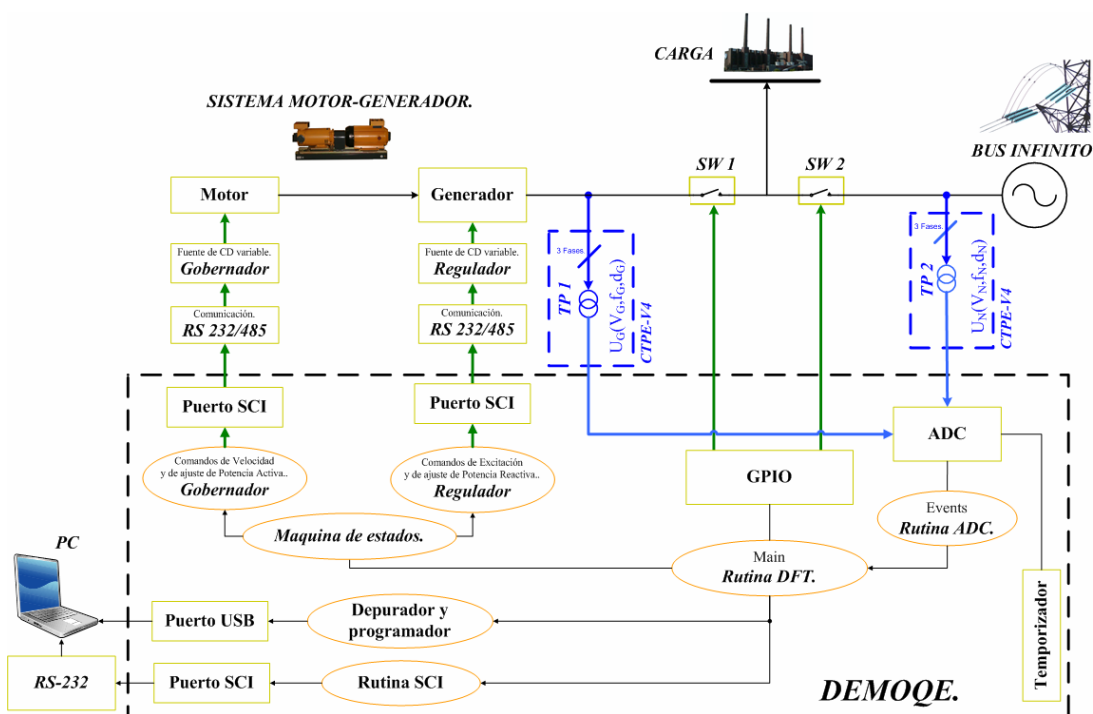


Figura 3.4.- Diagrama de bloques Hardware/Software del sincronoscopio.

Esta es una forma muy útil de abordar el problema, de manera que se tenga un amplio panorama de lo que se pretende implementar. En rectángulos se muestra la manera en que el hardware es implementado, mientras que con óvalos el software es indicado.

En la parte superior derecha del diagrama se observa la interacción que existe entre los TC (transformadores de corriente) y los TP (transformadores de potencia) con los canales ADC del microcontrolador y este a su vez con las subrutinas encargadas de cuantificar estos potenciales a valores digitales para su procesamiento en el algoritmo del sincronoscopio. Los TC y TP se encuentran implementados en la tarjeta CTPE-V4.

También se observa que los puertos de entrada salida de uso general (GPIO) comandan el accionamiento y apagado de los contactores SW1 y SW2, en el instante que el sincronoscopio lo requiera.

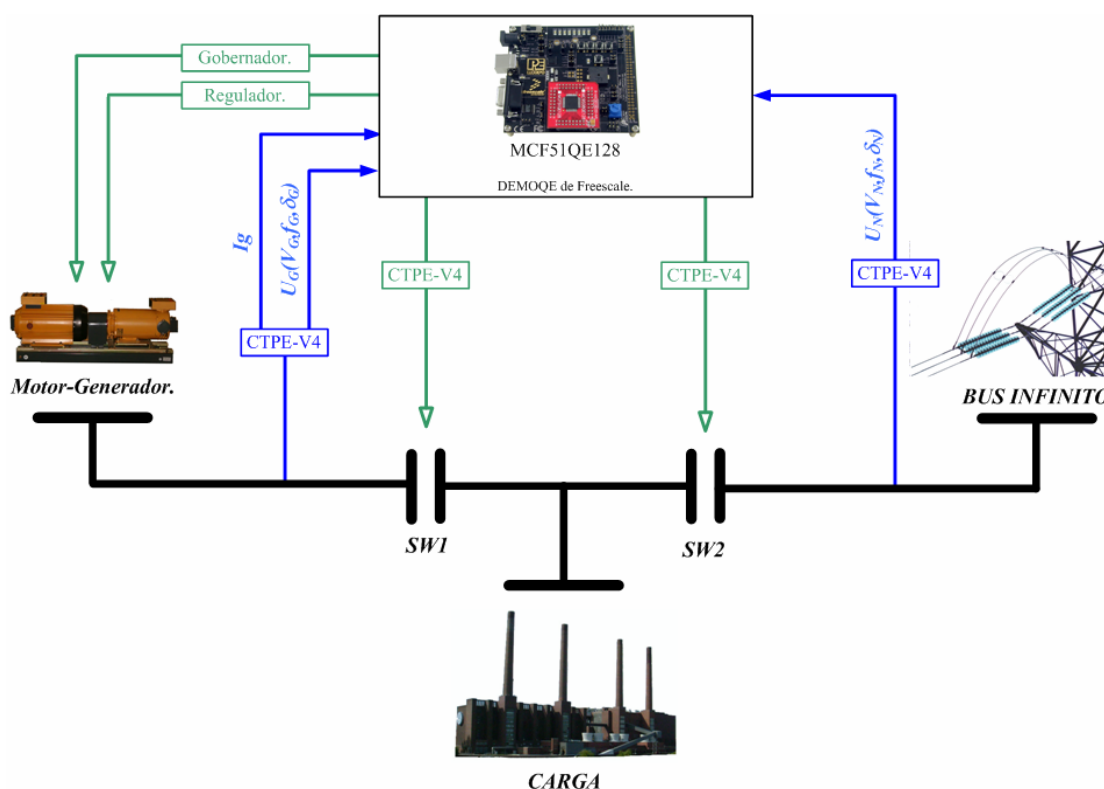


Figura 3.5.- Diagrama de interconexión de operación del sincronoscopio (Hardware).

En la figura 3.5 se muestra un diagrama con el hardware implementado, sin el software requerido.

El monitoreo en tiempo real se lleva a cabo por medio del puerto de comunicación serie SCI vía USB a través del ambiente de depuración y el análisis estadístico.

3.3 Tarjeta de desarrollo DEMOQE de Freescale.

La tarjeta DEMOQE, es un sistema de desarrollo de bajo costo diseñado para demostrar, evaluar y depurar los microcontroladores MC9S08QE128 y MCF51QE128 de Freescale, ver figura 3.6.

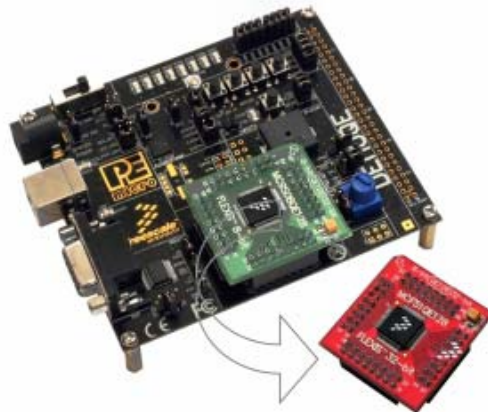


Figura 3.6.- Tarjeta de desarrollo DEMOQE.

El circuito de multienlace de la tarjeta DEMOQE permite al procesador ser depurado y programado a través del puerto USB desde la PC, además de permitir energizar a la tarjeta mediante ese mismo puerto.

Mediante el puerto de expansión de la tarjeta DEMOQE designado con el identificador “MCU_PORT” se pueden interconectar los periféricos a las diversas etapas de potencia, conectar con canales de comunicación, de medición, etc. Ver figura 3.7.

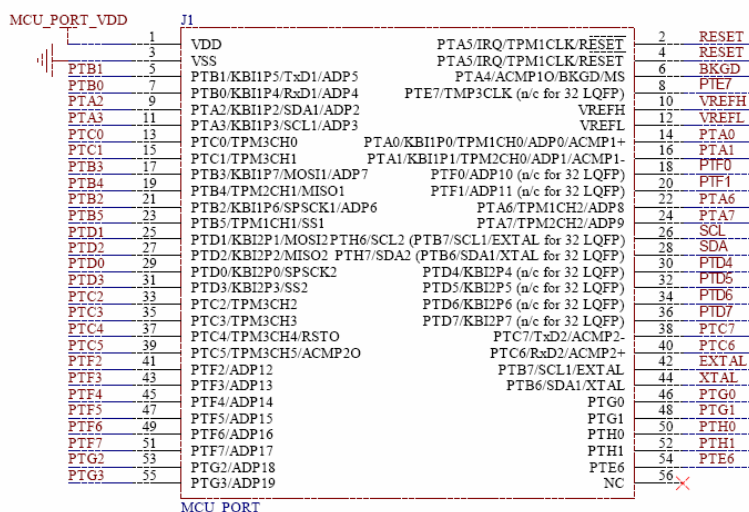


Figura 3.7.- Tira de pines de expansión de la tarjeta de desarrollo DEMOQE128UM.

La figura 3.8 muestra la distribución de pines para el microcontrolador de 32 bits MCF51QE128.

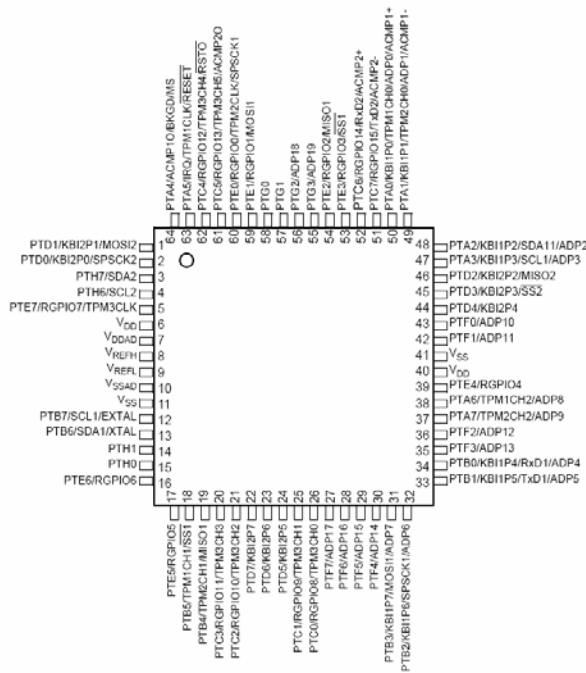


Figura 3.8.- Microcontrolador de 32 bits, MCF51QE128.

3.4 Diagrama unifilar del sincronoscopio.

3.4.1 Diagrama unifilar: CTPE-V4.y etapa de potencia.

En la figura 3.9, se muestra las conexiones que se llevaron a cabo para interconectar la tarjeta CTPE-V4 con los contactores de la compañía suministradora de energía eléctrica y del sistema motor generador.

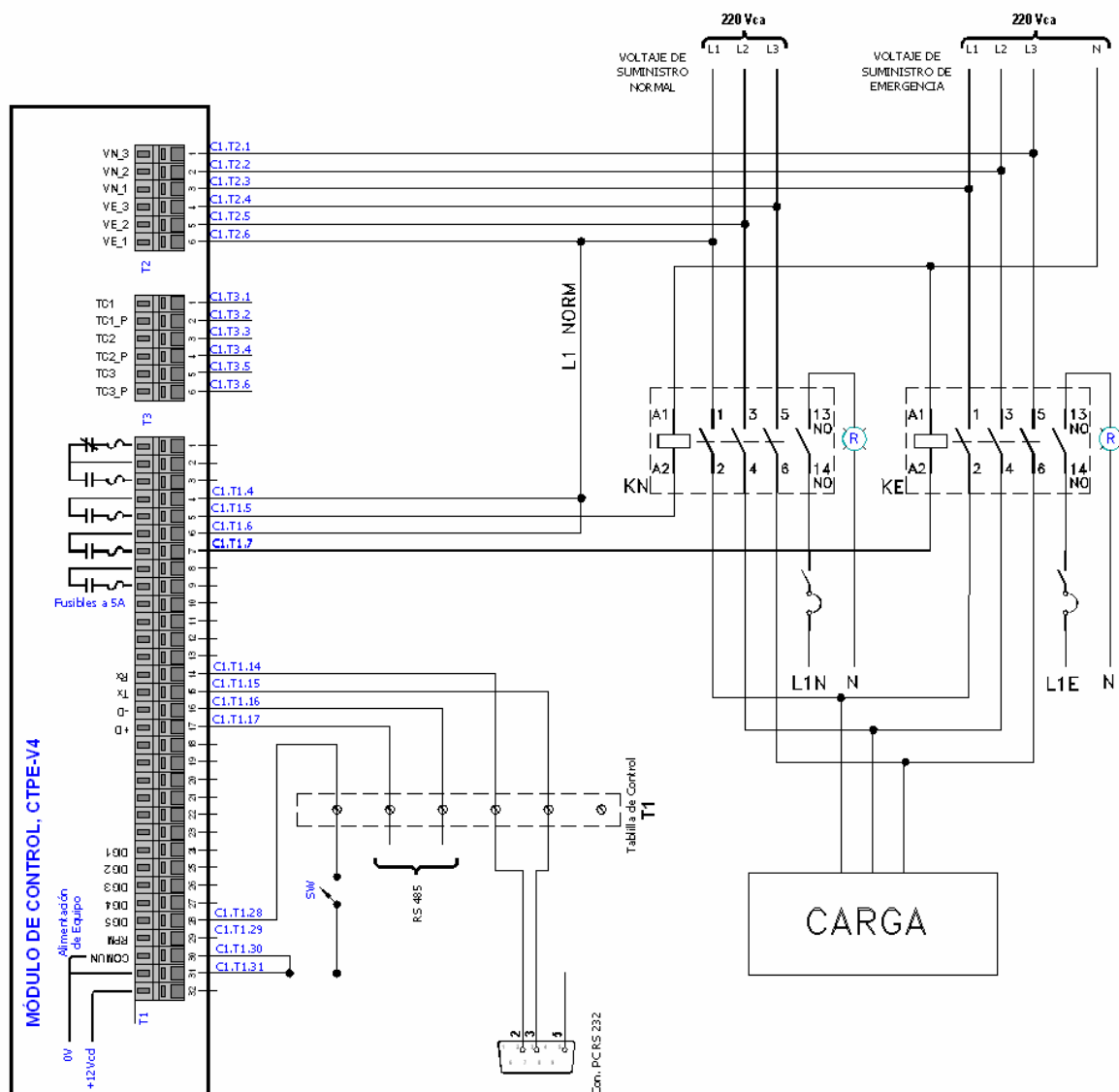


Figura 3.9.- Diagrama unifilar: CTPE-V4.y etapa de potencia (contactores)

3.4.2 Diagrama unifilar: CTPE-V4.y DEMOQE.

En la figura 3.10, se muestra las conexiones que se llevaron a cabo para interconectar la tarjeta CTPE-V4 con la tarjeta de desarrollo DEMOQE.

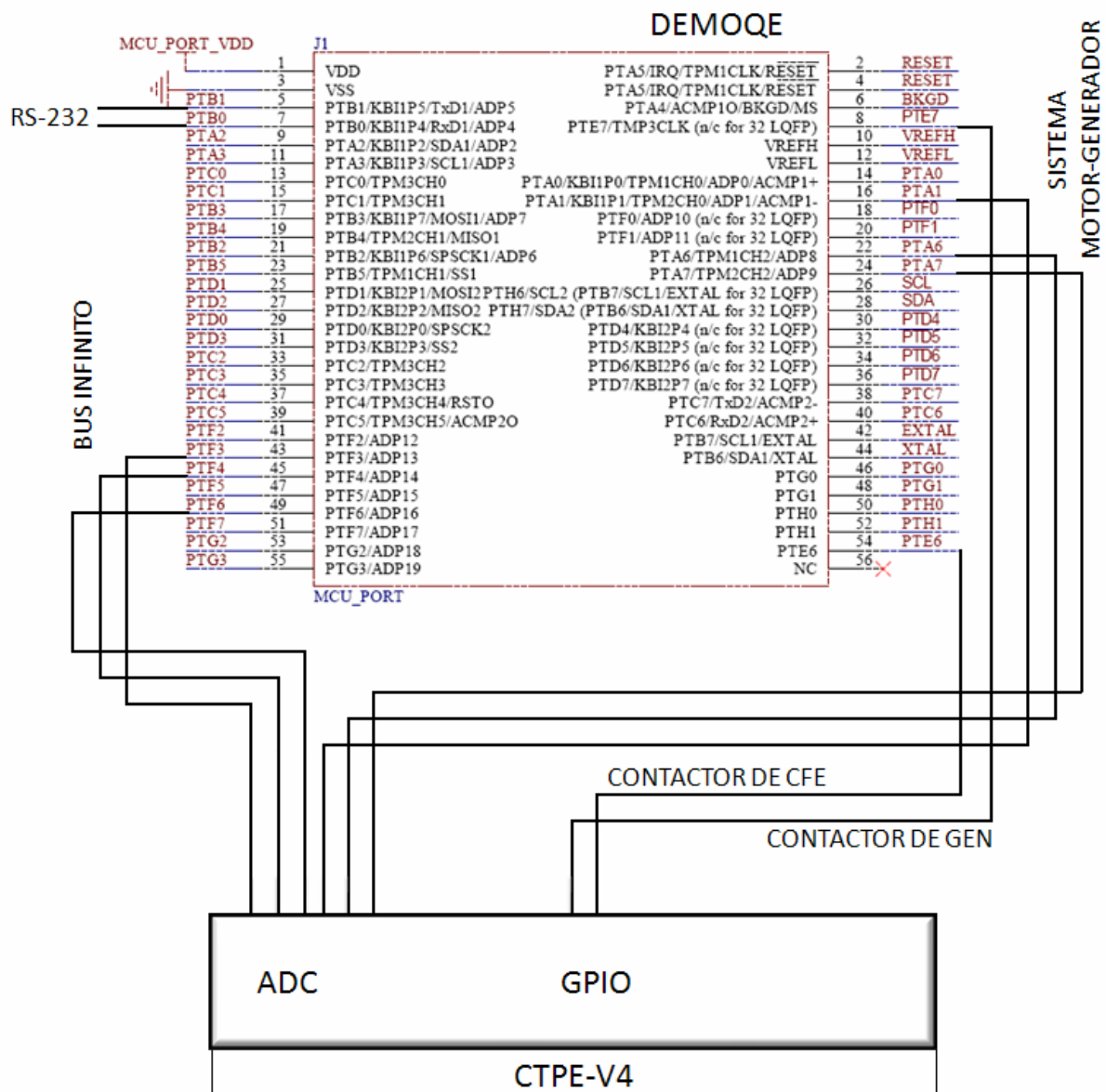


Figura 3.10.- Diagrama unifilar: CTPE-V4 y DEMOQE.

3.5 Arquitectura del microcontrolador MCF51QE128.

El ColdFire V1 es el primer microcontrolador con arquitecturas de 8/32-bit compatible. Los controladores, así como los periféricos y herramientas de desarrollo utilizando en el núcleo V1 se comparten con los de la arquitectura del núcleo básico S08 de 8 bits; la compatibilidad de pin-a pin- facilita también la migración a 32 bits.

Diseñado para longitudes de palabra de 32 bit, el núcleo V1 es una versión simplificada del ColdFire V2 básico. Cuenta con una mejor manipulación de operandos de 8 bits (bytes) y de 16 bits (palabra), manteniendo el mismo modo de direccionamiento y definiciones de la arquitectura ColdFire. La compatibilidad con los demás núcleos de ColdFire (V2 - V4) asegura diseños que le permite migrar fácilmente a soluciones de mayor rendimiento.

El núcleo ColdFire V1 utiliza un único pin para la retroalimentación del módulo de depuración (BDM). Esta actualización en la interfaz de depuración implementa un buffer para la lectura, mientras que el procesador mantiene un nivel mínimo de gastos generales.

Los periféricos comunes para el V1 y el núcleo básico S08 incluye un oscilador de reloj, reloj interno fuente, puertos analógico-digital y la interfaz de comunicación serie; además de las herramientas de desarrollo del software de CodeWarrior® como lo son el ambiente de desarrollo integral (IDE), herramienta de inicialización, depuración, compilación y conjunto de instrucciones, el cual sigue siendo compatible con las arquitecturas superiores al V1.

3.5.1 Características de la Unidad Central de Procesamiento (CPU) de 32 bits ColdFire Versión 1.

- CPU a 50.33 MHz con alimentación de 3.6V a 2.1V y 50.33 MHz con 3.6V a 2.1V
- Soporta hasta 30 solicitudes de interrupción de periféricos y 7 por software.

-
- Circuito de seguridad para prevenir accesos inautorizados al contenido de la RAM y flash.
 - 24-canales ADC de 12-bit de resolución con tiempos de conversión de 2.5 μ s; función de comparación automático; sensor de temperatura 1.7 mV/°C; canal de referencia; funcionalidad completa de 3.6V a 1.8V.
 - Dos comparadores analógicos con selector de interrupción (flanco de subida, de bajada o ambos).
 - Dos interfaces de comunicación serial (SCI).
 - Dos buses síncronos de enlaces de datos en serie (SPI).
 - Dos canales de comunicación entre circuitos integrados (IIC).
 - Puertos de propósito general de Entradas/Salidas.
 - 70 pines de salida o de entrada.
 - 16 bit rápidos de entrada y/o salida.
 - Hasta 256 KBytes de memoria Flash.
 - Hasta 32 Kbytes de memoria RAM estática (SRAM).
 - Controlador de red de área (CAN).
 - Control de redundancia cíclica (CRC).
 - Teclado interrupción (KBI).
 - Reloj multiusos (MCG).
 - Temporizador de ancho de pulso modulado (TPM).

3.6 Conversor Analógico/Digital.

El convertidor analógico a digital de 12 bits (ADC) utiliza el algoritmo del tipo de aproximaciones sucesivas y está diseñado para operar dentro del sistema integrado del microcontrolador.

3.6.1 Método de aproximaciones sucesivas.

En el método de conversión A/D por "Conversión Inversa", utiliza un convertidor D/A y tiene la ventaja de su simplicidad, pero se obtiene el inconveniente del retardo de

tiempo de conversión que se precisa para realizar la secuencia de conteo de los "n" bits; la adición de un comparador adicional, para acelerar la secuencia del contador cuando el error es mayor que un cierto nivel, permite reducir apreciablemente el tiempo de conversión del orden de 2^n al de orden $2^n/2+1$; tiempo que es posible reducir si se efectúan "n" comparaciones sucesivas entre la tensión de entrada V_x y la tensión V_h generada durante la conversión, para esto se substituye el contador del "convertidor inverso" por una unidad lógica de comparación que determina la secuencia de conversión de un convertidor ADC de aproximaciones sucesivas.

La primera comparación se deberá efectuar entre la tensión de entrada V_x y la tensión $V_h=V_{max}/2$, correspondiente a la palabra 1000, si V_h es mayor o igual a V_x , se determina que el bit de mayor peso debe ser uno, pero si $V_h < V_x$, se debe memorizar un cero en el registro de salida; en el siguiente impulso de reloj se efectúa una segunda comparación de V_h correspondiente a la palabra 1100, si la comparación anterior había dado positiva, o se compara contra 0100 en caso contrario; la salida del comparador determina el valor que debe memorizarse con un bit de peso $V_{max}/4$; configurándose de esta forma, la palabra de salida digital una vez efectuadas las "n" comparaciones sucesivas

3.6.2 Características del modulo ADC.

- Algoritmo del tipo aproximaciones sucesivas con 12 bits de resolución.
- Hasta 24 canales analógicos.
- Resolución de 12, 10 y 8 bits, justificado a la derecha en formato sin signo.
- Simple o conversión continua (retorno automático al estado ocioso después de una simple conversión)
- Configuración del tiempo de muestreo y velocidad de la conversión
- Bandera e interrupción de conversión completa.
- Entrada de reloj seleccionable, hasta cuatro fuentes
- Operación en modo esperar o stop3 para operación a bajos ruido.
- Fuente asíncrona de reloj para operación a bajos ruido.
- Selector de señal de disparo asíncrona por hardware para activación de conversión.

-
-
- Comparador automático con interrumpir para menor-que, mayor-que o igual-a, del valor programado.

3.6.3 Descripción funcional de operación del modulo ADC

El modulo ADC es deshabilitado durante el RESET o cuando los bits del registro ADCH están puestos a 1. El modulo esta en el modo ocioso cuando una conversión ha sido completada o cuando otra conversión aun no ha sido iniciada. Cuando esta ocioso, el modulo está en su estado de bajo consumo de potencia.

El ADC puede ejecutar una conversión analógica a digital en cualquiera de sus canales. En el modo de conversión de 12 y 10 bits, el voltaje del canal seleccionado es convertido por el algoritmo de aproximaciones sucesivas en un valor digital de 12 bits. En el modo de 8 bits, el voltaje del canal seleccionado es convertido por el algoritmo de aproximaciones sucesivas en un valor digital de 9 bits.

Cuando la conversión es completada, el resultado es almacenado en el registro de datos (ADCRH y ADCRL). En el modo de 10 bits, el resultado es redondeado a 10 bits y colocado en el registro de datos (ADCRH y ADCRL). En el modo de 8 bits de resolución, el resultado es redondeado a 8 bits y colocado en ADCRL. La bandera de conversión completa (COCO) es entonces ajustada a 1 y una interrupción es generada si la interrupción de conversión completada a sido habilitada (AIEN = 1).

El modulo ADC tiene la capacidad de automáticamente comparar el resultado de la conversión con el contenido de su registro de comparación. La función de comparación es habilitada ajustando el bit ACFE y opera con cualquiera de los modos de conversión y configuraciones.

3.7 Diseño de los acondicionadores de señales.

Para poder monitorear las condiciones de operación (voltaje, secuencia de fases, frecuencia y ángulos de fase) de la compañía suministradora de energía eléctrica y del sistema motor-generator, con los puertos ADC del microcontrolador, se requiere acondicionarlos a niveles de voltaje adecuados dentro del rango de 0 - 3.3 VCD, para no dañar los canales de conversión de señales analógicas a digitales. Esta etapa de acondicionamiento se logra con la tarjeta CTPE-V4, ver figura 3.11.

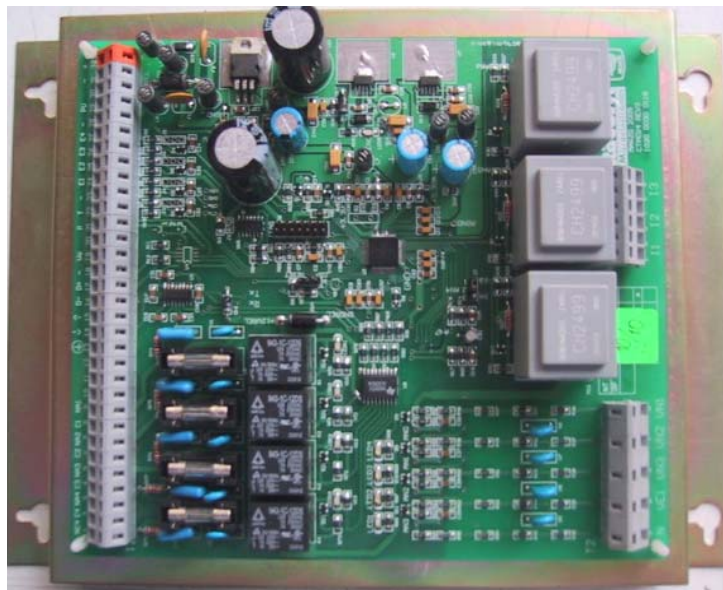


Figura 3.11.- Tarjeta CTPE-V4.

3.7.1 Especificaciones Eléctricas.

- Voltaje de alimentación: 12 V.
- Rango de Medición de Voltaje: 30 a 150 V.
- Contacto de Relevadores: 5A, 250 V.
- Transformadores de corriente a voltaje, (5 Amp \approx 5 VCA).

3.7.2 Entradas Digitales.

Tabla 3.1. Entradas digitales de la tarjeta CTPE-V4..

<i>Nombre:</i>	<i>Terminales</i>	<i>Tipo:</i>
ED1	T1 (28,27)	Interruptor
ED2	T1 (28,26)	Interruptor
ED3	T1 (28,25)	Interruptor
ED4	T1 (28,24)	Interruptor

3.7.3 Relevadores.

Tabla 3.2. Relevadores de la tarjeta CTPE-V4..

<i>Salidas de Relevador:</i>	<i>Tipo:</i>	<i>Terminales:</i>	<i>Especificación:</i>
Relevador 1 Activado.	NA	T1 (8,9)	250 V máx. 5 A máx.
Relevador 2 Activado.	NA	T1 (7,6)	
Relevador 3 Activado.	NA	T1 (5,4)	
Relevador 4 Activado.	NA/NC	T1 (1,2,3)	

3.7.4 Acondicionamiento de señales de voltaje de CA.

La medición de voltaje de CA se realiza por medio de un circuito basado en la topología de divisor de voltaje (figura 3.12), el cual mide en un rango de 0 - 300 V CA y da una medición proporcional dentro del rango de 0 - 3.3V CD, el cual estará montado sobre una señal de 1.65 VCD.

Para analizar el circuito se utiliza el teorema de superposición de fuentes de voltaje.

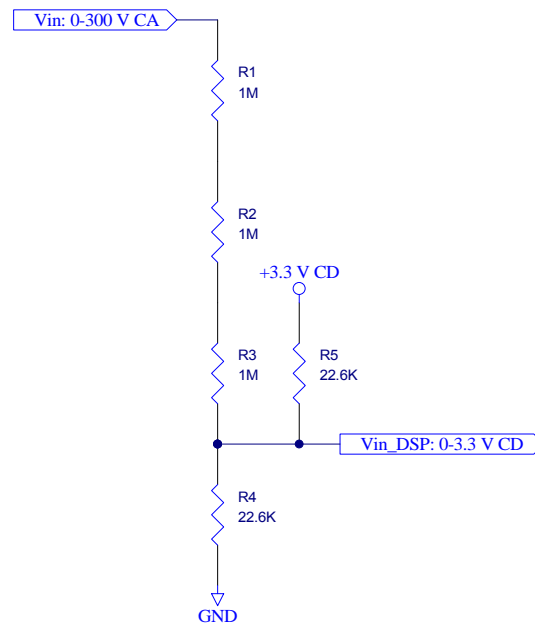


Figura 3.12.- Circuito acondicionador de señal de VAC

➤ Análisis en CA:

Por el teorema de superposición de fuentes de voltaje, primero se analiza la red en CA, por lo que la fuente de voltaje de 3.3 V CD se comporta como un corto circuito, al forzar el voltaje en sus terminales a 0V, quedando el resistor R5 conectado a tierra y a Vin_DSP; la red se simplifica sumando los resistores en serie R1, R2, R3 (RS) y el paralelo que se forma con las resistores R4 y R5, obteniendo el circuito equivalente que se muestra en la figura 3.13.

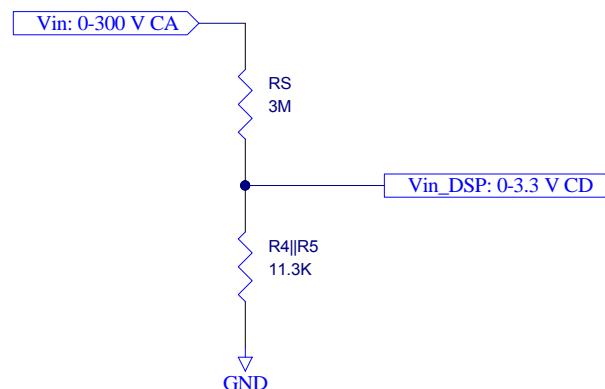


Figura 3.13.- Circuito acondicionador de señal de VAC

Considerando un voltaje de entrada $V_{in}=300$ VCA se calcula el voltaje pico para el semiciclo positivo:

$$V_{in_Pico} = V_{in} \times \sqrt{2} \quad (3.1)$$

$$V_{in_Pico_Positivo} = 300V \times \sqrt{2} = 424.26 \quad (3.2)$$

Y de igual manera para el semiciclo negativo.

$$V_{in_Pico_Negativo} = -300V \times \sqrt{2} = -424.26 \quad (3.3)$$

Por el método de divisor de voltaje, se obtiene $V_{in_DSP_Pico_Positivo}$.

$$V_{in_DSP_Pico_Positivo} = \frac{424.26V \times 11.3K\Omega}{(3M + 11.3K)\Omega} = \frac{4794183.976V}{3011300} = 1.5921V \quad (3.4)$$

Ahora, $V_{in_DSP_Pico_Negativo}$.

$$V_{in_DSP_Pico_Negativo} = \frac{-424.26 \times 11.3K\Omega}{(3M + 11.3K)\Omega} = \frac{-4794183.976V}{3011300} = -1.5921V \quad (3.5)$$

➤ Análisis en CD:

Procediendo con el análisis por el teorema de superposición, se analiza la red de la figura 3.14 en CD; en la cual, la fuente de voltaje V_{in} se comporta como un corto circuito, al forzar el voltaje en sus terminales a 0V; la red se simplifica al sumar los resistores en serie R1, R2, R3 (R_S) y el arreglo en paralelo de $R_S||R_4$.

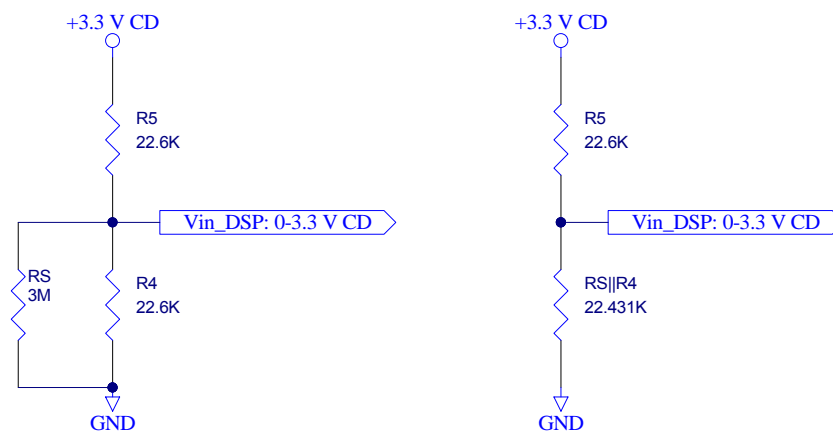


Figura 3.14.- Circuito resultante con fuente de CA en corto

Se calcula por divisor de voltaje, el voltaje en V_{in_DSP} :

$$V_{in_DSP_{CD}} = \frac{3.3V \times 22.431K\Omega}{(22.6K + 22.431K)\Omega} = \frac{74022.3V}{45031.020} = 1.6438V \quad (3.6)$$

Por lo tanto, para el semiciclo positivo:

$$V_{in_DSP} = V_{in_DSP_{Pico_Positivo}} + V_{in_DSP_{CD}} = 1.592 + 1.6438 = 3.2359V \quad (3.7)$$

Y para el negativo:

$$V_{in_DSP} = V_{in_DSP_{Pico_Negativo}} + V_{in_DSP_{CD}} = -1.592 + 1.6438 = 0.0517V \quad (3.8)$$

Lo cual está dentro de los niveles permitidos de voltajes de entrada del puerto ADC, los cuales son de 0 a 3.3 VCD.

La tabla 3.3 conjunta las señales de voltaje alternas a ser monitoreadas.

Tabla 3.3. Entradas analógicas de voltaje de la tarjeta CTPE-V4..

<i>Nombre:</i>	<i>Descripción:</i>	<i>Rango de Vin:</i>
VG1	Voltaje de W2 del Generador.	Voltaje máx. 300V
VG2	Voltaje de U2 del Generador.	Voltaje máx. 300V
VG3	Voltaje de V2 del Generador.	Voltaje máx. 300V
VL1	Voltaje de L1 de CFE.	Voltaje máx. 300V
VL2	Voltaje de L2 de CFE.	Voltaje máx. 300V
VL3	Voltaje de L3 de CFE.	Voltaje máx. 300V

3.7.5 Acondicionamiento de señales de corriente alterna.

La medición de intensidad de corriente se lleva a cabo por medio de tres transformadores de corriente a voltaje, los cuales entregan 5 VCA cuando circula una corriente de 5 Amp, dicho voltaje pasa por una etapa de acondicionamiento por medio de un divisor de voltaje.

Para analizar el circuito se utiliza el teorema de superposición de fuentes de voltaje. En la figura 3.15 se muestra el circuito del cual se parte para el análisis.

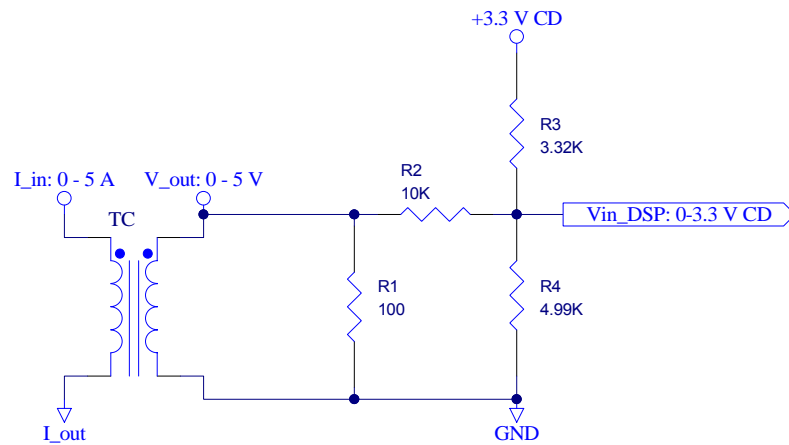


Figura 3.15.- Circuito acondicionador de señal de I AC

➤ Análisis en CA:

Primero se analiza la red en CA, por lo que la fuente de voltaje de 3.3 V CD se comporta como un corto circuito, al forzar el voltaje en sus terminales a 0V; el resistor R3 queda en paralelo a R4, simplificando de esta manera el circuito al resolver el arreglo en paralelo; ver figura 3.16.

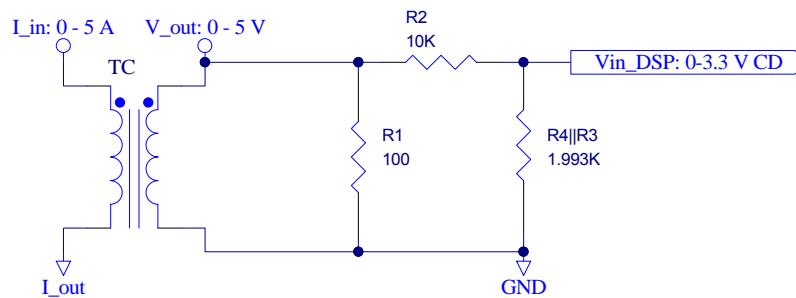


Figura 3.16.- Circuito resultante con fuente de CD en corto.

Considerando una corriente de 5A tendremos un $V_{out}=5$ VCA por lo que el voltaje pico $V_{out_{Pico}}$ es:

$$V_{out_{Pico}} = V_{out} \times \sqrt{2} \tag{3.9}$$

Para el semiciclo positivo se tiene

$$V_{out_{Pico_{Positivo}}} = 5V \times \sqrt{2} = 7.071 \tag{3.10}$$

De igual manera el voltaje pico para el ciclo negativo es:

$$V_{\text{out_Pico_Negativo}} = -5V \times \sqrt{2} = -7.071 \quad (3.11)$$

y por divisor de voltaje se calcula $V_{\text{in_DSP_Pico_Positivo}}$.

$$V_{\text{in_DSP_Pico_Positivo}} = \frac{7.071V \times 1.993K\Omega}{(10 + 1.993K)\Omega} = \frac{14096.8V}{11993.5} = 1.175V \quad (3.12)$$

Ahora se calcula $V_{\text{in_DSP_Pico_Negativo}}$.

$$V_{\text{in_DSP_Pico_Negativo}} = \frac{-7.071V \times 1.993K\Omega}{(10 + 1.993K)\Omega} = \frac{-14096.8V}{11993.5} = -1.175V \quad (3.13)$$

➤ Análisis en CD:

Procediendo con el análisis por el teorema de superposición, se analiza la red de la figura 3.15 en CD; la fuente de voltaje V_{out} se comporta como un corto circuito, al forzar el voltaje en sus terminales a 0V, eliminando así el efecto del resistor R1. Se simplifica el circuito calculando el arreglo en paralelo $R2 \parallel R4$; el circuito resultante se muestra en la figura 3.17.

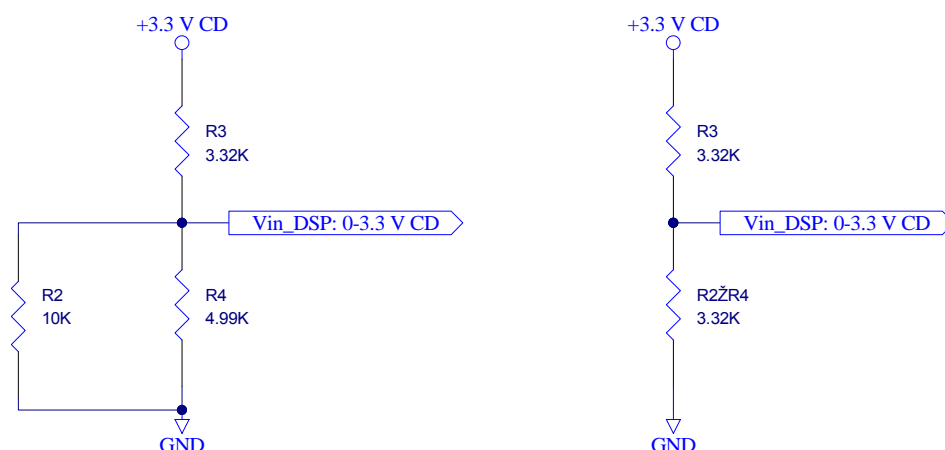


Figura 3.17.- Circuito resultante con fuente de CA en corto.

Se calcula el voltaje $V_{\text{in_DSP}}$ por divisor de voltaje obteniendo lo siguiente.

$$V_{\text{in_DSP_CD}} = \frac{3.3V \times 3.32K\Omega}{(3.32K + 3.32K)\Omega} = \frac{10.956V}{6.64} = 1.65V \quad (3.14)$$

Por lo tanto se tiene para el semiciclo positivo:

$$V_{in_DSP} = V_{in_DSP_{Pico_Positivo}} + V_{in_DSP_{CD}} = 1.175 + 1.65 = 2.825V \quad (3.15)$$

Y para el semiciclo negativo:

$$V_{in_DSP} = V_{in_DSP_{Pico_Negativo}} + V_{in_DSP_{CD}} = -1.175 + 1.65 = 0.475V \quad (3.16)$$

Lo cual está dentro de los niveles permitidos de voltajes de entrada del puerto ADC, los cuales son de 0 a 3.3 VCD.

La tabla 3.4 conjunta las señales de corriente alterna a ser monitoreadas.

Tabla 3.4.- Entradas analógicas de corriente de la tarjeta CTPE-V4..

<i>Nombre:</i>	<i>Descripción:</i>	<i>Rango de lin:</i>
IL1	Corriente de Carga de L1-W2.	Corriente máx. 5 A
IL2	Corriente de Carga de L2-U2.	Corriente máx. 5 A
IL3	Corriente de Carga de L3-V2.	Corriente máx. 5 A

3.8 Sistema motor-generator.

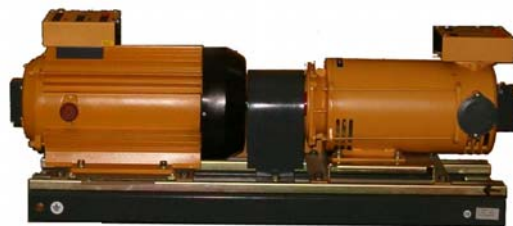


Figura 3.18.- Sistema motor-generator.

3.8.1 Máquina síncrona trifásica, DL1026A de “DE LORENZO”.

Máquina con inductor liso y devanado trifásico inducido en el estator para funcionar ya sea como alternador o como motor síncrono. Ver figura 3.19.

Características técnicas:

Tabla 3.5. Características técnicas de la máquina síncrona trifásica

	Como alternador:	Como motor:
Potencia	1.1 KVA	1 KW
Tensión:	220/380 V Δ/Y	
Corriente:	2.9/1.7 A Δ/Y	
Velocidad:	3000 rpm.	
Excitación:	175 V / 0.4 A.	



Figura 3.19.- Máquina síncrona trifásica DL1026A de “DE LORENZO”.

3.8.2 Motor de CD, DL1023PS de “DE LORENZO”.

Las características técnicas del motor de CD son:

- Potencia: 1.8 KW
- Voltaje: 220 V
- Velocidad: 3000 min⁻¹
- Voltaje de excitación: 170 V

Ver figura 3.20.



Figura 3.20.- Motor de CD DL1023PS de “DE LORENZO”

3.9 Contactor tetrapolar, A9-30-10 de “ABB”.

Los contactores de las series A9 se utilizan principalmente para controlar motores trifásicos y generalmente para controlar circuitos principales 690 VCA / 1000 VCA o 220 VCD / 440 VCD. Los contactores también pueden utilizarse para muchas otras aplicaciones tales como el aislamiento, la conexión de baterías de condensadores y conexión de iluminación. Ver figura 3.21.



Figura 3.21.- Contactor tripolar A9-30-10.

Características de los contactores tripolares de la serie A:

- Diseño compacto.
- Polos principales y bloques de contactos auxiliares de 1 piso:
 - 3 polos principales.
 - 1 contacto auxiliar incorporado.
 - Bloques de contactos auxiliares adicionales frontales y laterales.
- Circuito de mando por CA con circuito magnético laminado.
- Bobina (110...127)V ~ 60 Hz; (220/240) V ~ In=9A

CAPÍTULO 4

DISEÑO DEL SOFTWARE

En este capítulo se presenta el diagrama de estados que seguirá el sincronoscopio así como las condiciones necesarias que le harán cambiar de estado. Se describe el procedimiento de inicialización del microcontrolador con técnicas de programación orientada a objetos mediante capsulas de programa llamadas “beans” que contienen la configuración de los periféricos, con la interfase “Procesor Expert” integrada en el ambiente de desarrollo de “CodeWarrior”. Se describe el procedimiento de la implementación de la interfase de depuración con la herramienta “Visualization Tools”. Esta herramienta permite el monitoreo en tiempo real de las variables empleadas.

4.1 Diagrama de estados.

El diagrama de estados del sincronoscopio se muestra en la figura 4.1.

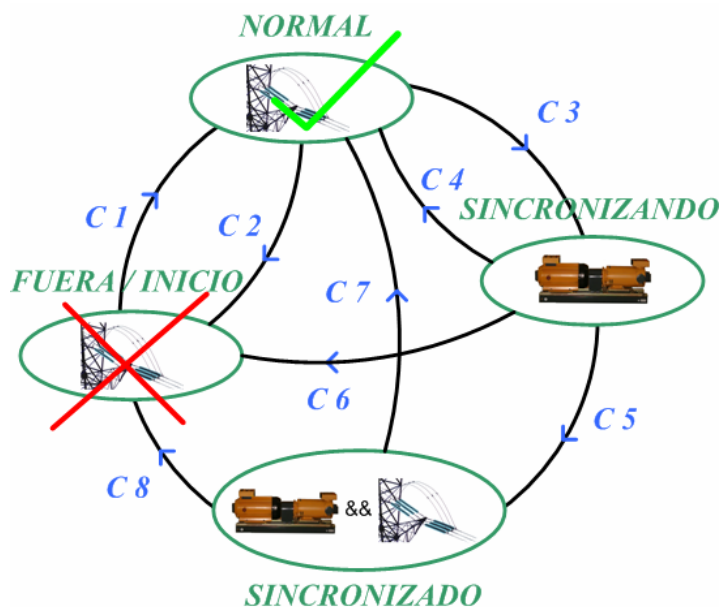


Figura 4.1.- Diagrama de estados del control de sincronía.

En la figura 4.1 se ilustra el funcionamiento del sincronoscopio; en el que los óvalos representan los estados en los que se puede encontrar el equipo en operación y los conectores con los subíndices C1-8, representan las condiciones que lo harán cambiar de estado.

La secuencia de operación del sincronoscopio, comienza cuando CFE se encuentra deshabilitada y el sistema motor-generador apagado, por lo que el sincronoscopio permanece en el estado de “FUERA / INICIO”. En este estado, el equipo se encarga de monitorear el estado de CFE y deshabilitar los contactores de CFE y del sistema motor-generador que los interconectan entre si y a su vez con la carga; si no estuvieran activados, el control solo estaría reforzando esa acción, haciendo al sincronoscopio más robusto e inalterable a alguna señal de ruido que pudiera activar los contactores. Si el sistema motor-generador estuviera activado, sus parámetros solamente serán monitoreados y visualizados si su frecuencia de salida es la óptima para realizar la sincronización con la compañía suministradora de energía eléctrica.

Del diagrama de flujo se puede observar que cuando la condición 1 (C1) sucede, el estado del equipo cambia al de “NORMAL”, siendo “C1” el instante en que la compañía suministradora de energía eléctrica es habilitada.

En el estado de “NORMAL”, pueden activarse dos condiciones: C2 y C3. La condición 2 (C2), hace que el equipo retorne al estado de “FUERA” debido a que CFE fue desconectada o fallo, mientras que la condición 3 (C3), que funge como solicitud de “Inicio de sincronización”, es activada por un nivel mínimo preestablecido de voltaje a la salida del sistema motor-generador cuando este es activado; condición que es detectada debido a que una de las fases de salida del generador es monitoreada por un canal ADC del microcontrolador. Si sucede C3, el sistema pasara al estado de “SINCRONIZANDO”.

Si el sincronoscopio llega al estado “NORMAL” del estado de “SINCRONIZADO”, debido a que el sistema motor-generador fue desincronizado (condición C7), el sincronoscopio no podrá pasar al siguiente estado; para reactivar el funcionamiento del

sincronoscopio, se debe de apagar el sistema motor-generator o deshabilitar el suministro de la compañía suministradora de energía eléctrica.

En el estado de “SINCRONIZANDO”, el equipo se encarga de generar los comandos visuales para ajustar los parámetros de salida del sistema motor-generator a los de la compañía suministradora de energía eléctrica. Cuando las condiciones de sincronía son satisfechas dentro de los márgenes de tolerancia, de acuerdo a la norma EGSA 100E 1992 y EGSA 100R 1992^a, la condición 5 (C5) es activada y el estado del sincronoscopio será el de “SINCRONIZADO”. Al activarse la condición C5, un contador de “Tiempo permitido de sincronización” es ajustado a un valor preestablecido. Si antes de activarse la condición C5, el sistema motor-generator es desactivado, activara la condición C4, por lo que el sincronoscopio pasara al estado de “NORMAL” o si falla el suministro de la compañía suministradora de energía eléctrica, se activara la condición C6 y el sincronoscopio pasara al estado de “FUERA / INICIO”.

En el estado de “SINCRONIZADO”, se lleva a cabo la conexión en paralelo de ambos sistemas. Durante la sincronización, el contador de “Tiempo permitido de sincronización”, es decrementado y al llegar a cero, la condición C7 es activada, llevando al sincronoscopio al estado de “NORMAL” o si el suministro de la compañía suministradora de energía eléctrica falla, la condición C8 será activada, pasando el estado del sincronoscopio al de “FUERA / INICIO”.

Para poder apagar al sistema motor-generator durante el estado de “SINCRONIZADO” y dejar habilitado el suministro de energía eléctrica de la red comercial, será necesario esperar que transcurra el “Tiempo permitido de sincronización” o por medio del ambiente de depuración, activar un comando que modifique el valor de una variable que permita sacar al sistema motor-generator de sincronía y pasar el estado del sincronoscopio al de “NORMAL” para posteriormente apagarlo sin ningún problema.

No se debe desconectar la alimentación del sistema motor-generator mientras este se encuentra sincronizado con el Bus infinito, de suceder, el sistema motor-generator se motorizará y consumirá energía eléctrica.

4.1.1 Definiciones de los estados

En resumen, se describen las definiciones de los estados de: “FUERA / INICIO”, “NORMAL”, “SINCRONIZANDO” y “SINCRONIZADO”:

1. Estado “FUERA / INICIO”:

- Estado inicial del Sincronoscopio.
- La compañía de suministro de energía eléctrica se encuentra deshabilitada.
- Se deshabilita el contactor de la compañía de suministro de energía eléctrica y del sistema-motor generador.
- Si el suministro de energía eléctrica de CFE se habilita, la condición C1 es activada.
- Si el sistema motor-generator esta activado, sus parámetros solamente se monitorean y si su frecuencia es la optima para efectuar la sincronización, sus parámetros se visualizan en el ambiente de depuración.

2. Estado “NORMAL”:

- La compañía de suministro de energía eléctrica esta habilitada y se monitorean y visualizan sus parámetros en el ambiente de depuración.
- Solo el contactor de la compañía de suministro de energía eléctrica esta activado.
- La compañía de suministro de energía eléctrica alimenta a la carga.
- Si la energía eléctrica de la compañía suministradora es suspendida o falla, la condición C2 es activada.
- Si el sistema motor-generator es habilitado, la condición C3 es activada.

-
-
- Si el sistema motor-generador fue desincronizado, el sincronoscopio permanecerá en el estado de “NORMAL” monitoreando sus parámetros y si su frecuencia es la óptima para efectuar la sincronización, sus parámetros se visualizarán en el ambiente de depuración.

3. Estado de “SINCRONIZANDO”:

- La compañía de suministro de energía eléctrica está habilitada y se monitorean y visualizan sus parámetros en el ambiente de depuración.
- Solo el contactor de la compañía de suministro de energía eléctrica está activado.
- La compañía de suministro de energía eléctrica alimenta a la carga.
- Sistema motor-generador se encuentra en proceso de ser sincronizado.
- Comandos visuales generados para ajustar los parámetros del sistema motor-generador a los de la compañía suministradora de energía eléctrica.
- Si el sistema motor-generador es desactivado, la condición C4 es activada.
- Si los parámetros del sistema motor-generador son los óptimos para efectuar la sincronización, la condición C5 es activada y se carga el contador de “Tiempo permitido de sincronización” con un valor preestablecido.
- Si la energía eléctrica de la compañía suministradora es suspendida o falla, la condición C6 es activada.
- Si la frecuencia del sistema motor-generador es la óptima para efectuar la sincronización, sus parámetros son visualizados en el ambiente de depuración.
- Cuando la frecuencia y el voltaje son los óptimos para efectuar la sincronización, se visualiza el sincronoscopio virtual en el ambiente de depuración.

4. Estado de “SINCRONIZADO”:

- La compañía de suministro de energía eléctrica y el sistema motor-generador se encuentran sincronizados y alimentando a la carga.

-
- Se monitorean y visualizan los parámetros de la compañía de suministro de energía eléctrica y del sistema motor-generador.
 - Se decrementa el contador de “Tiempo permitido de sincronización” y si ha llegado a cero, la condición C7 es activada..
 - Si la energía eléctrica de la compañía suministradora es suspendida o falla, la condición C8 es activada.

4.1.2 Definición de las condiciones de cambio entre estados del Sincronizador.

De la figura 4.1 se visualizan conectores con los subíndices C1-7, los cuales indican las condiciones necesarias para que el sincronoscopio se desplace por los estados que lo conforman; a continuación se describe cada uno.

C1.- Condición 1:

- Si la compañía de suministro de energía eléctrica se restablece, el estado actual es “NORMAL”.

C2.- Condición 2:

- Si la compañía de suministro de energía eléctrica falla, el estado actual es “FUERA / INICIO”.

C3.- Condición 3:

- Si el sistema motor-generador se encuentra apagado y es activado con sus niveles mínimos de voltaje y frecuencia, el estado actual es “SINCRONIZANDO”.
- Si el sistema motor-generador fue desincronizado por el termino del “Tiempo permitido de sincronización” o por solicitud del operador por medio de un comando enviado al microcontrolador y este encuentra encendido, esta condición no se activara.

C4.- Condición 4:

- Si el sistema motor-generador es desactivado, el estado actual es “NORMAL”.

C5.- Condición 5:

- Si los parámetros del sistema motor-generador coinciden con los del bus infinito dentro de los márgenes de tolerancia establecidos por la norma, el estado actual es “SINCRONIZADO”.

C6.- Condición 6:

- Si la compañía de suministro de energía eléctrica falla, el estado actual es “FUERA / INICIO”.

C7.- Condición 7:

- Si el contador de “Tiempo permitido de sincronización” llega a cero, el estado actual es “NORMAL”.

C8.- Condición 8:

- Si la compañía de suministro de energía eléctrica falla, el estado actual es “FUERA / INICIO”.

4.2 Software del microcontrolador de 32 bits Flexis V1 ColdFire.

El microcontrolador utiliza un ambiente de desarrollo integral (IDE), en el software llamado “Code Warrior” de “Metrowerks” el cual incorpora la herramienta de configuración de “Processor Expert”, que facilita la configuración de los periféricos del microcontrolador dentro de la plataforma de “Windows”.

Flexis utiliza el sistema de desarrollo CodeWarrior de Freescale para el diseño de software para ambas arquitecturas de 8 y 32 bits. La nueva revisión de “CodeWarrior Development Studio for Microcontrollers V6.0” incluye el soporte para las familias RS08,

S08 y Coldfire V1 con herramientas de compilación y encadenamiento para cada una de las subfamilias, además de simulador, entorno de depuración, etc.

La “Special Edition” del CodeWarrior para microcontroladores V.6.0 permite trabajar gratuitamente con la herramienta hasta 32k de código generado en C para los S08 y 64k para los Coldfire V1. También tiene disponibles ampliaciones y versiones ‘standard’ y ‘profesional’.

El CodeWarrior para microcontroladores también incluye herramientas automáticas de generación de código de inicialización del cpu, que permiten acelerar el desarrollo de los productos [13].

La metodología de programación utilizada es por medio de un programa principal llamado “Main” y uno denominado “Events”, estando en este último, las diferentes interrupciones y/o eventos que son configuradas por el usuario. El programa principal siempre se está ejecutando dentro de un lazo infinito, el cual es interrumpido cada vez que un evento y/o interrupción es invocado.

4.2.1 Configuración Inicial del microcontrolador..

La configuración inicial implica seleccionar el CPU que se vaya a implementar, definir las características del cristal externo con el que se desee trabajar así como la configuración del PLL (multiplicador de frecuencia). En la figura 4.2 se visualiza que se implementó la CPU MCF51QE128LQFP64, cristal externo de 8MHz y el PLL a 24 MHz (High speed mode).

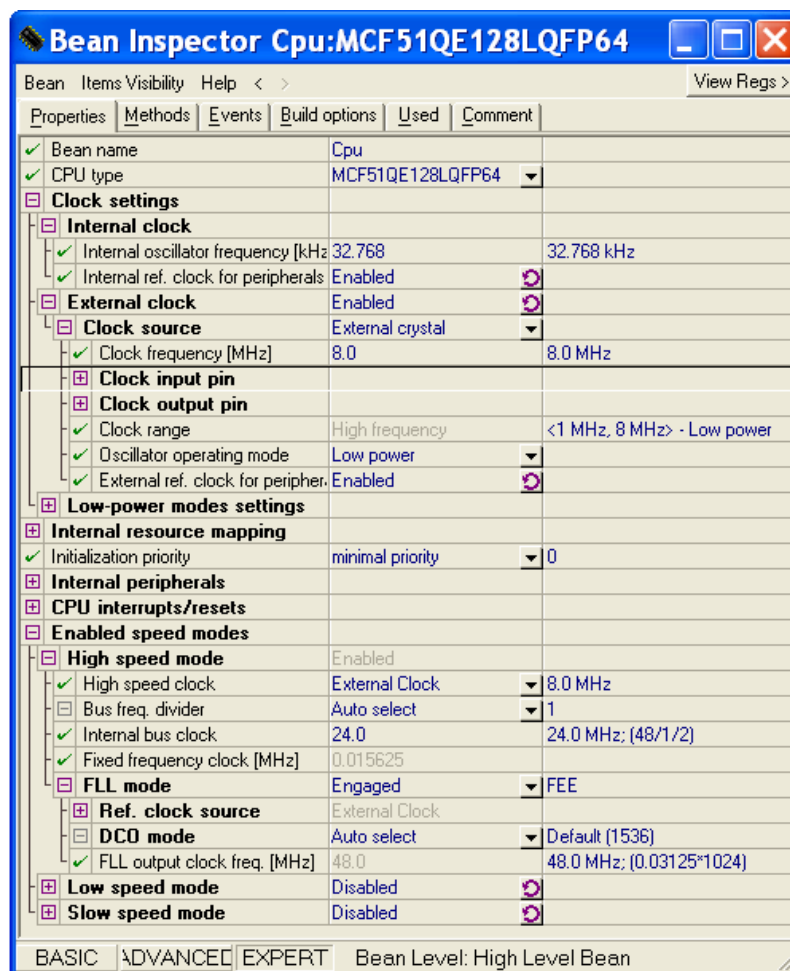


Figura 4.2.- Bean Inspector del CPU: MCF51QE128LQFP64.

4.2.2 Configuración del “TimerInt” para indicar el inicio de conversiones del ADC.

El “TimerInt” es una interrupción periódica que cada vez que se ejecute, indica el inicio de las conversiones digital-analógicas (A/D) en los canales ADC configurados.

Para esta aplicación de Ingeniería Eléctrica, se propuso tomar 96 muestras de señales a 60 Hz, por lo que la FM (Frecuencia de Muestreo) evaluada en la ecuación 4.1 es de 5760Hz (96×60).

$$FM = (\text{Fcia señal a muestrear})(\text{No. muestras a tomar}) \quad (4.1)$$

Por lo tanto el periodo de interrupción que se requiere es el inverso de la FM, siendo este de $173.611\mu\text{s}$ ($1/5760$).

La figura 4.3 muestra la configuración requerida en el “Bean Inspector” del “TimerInt” en el IDE en base a los cálculos anteriores.

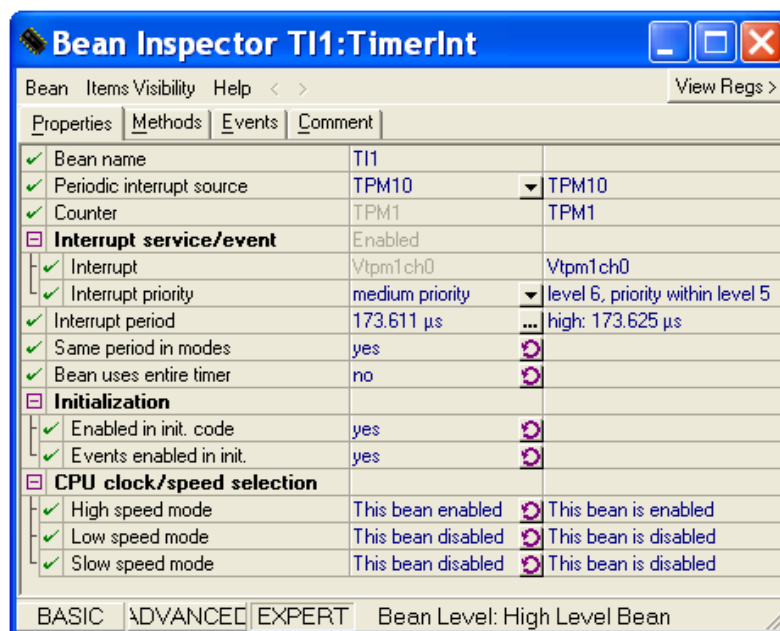


Figura 4.3.- Bean Inspector del “TimerInt” para indicar los inicios de conversión del ADC..

La figura 4.4 muestra el diagrama de flujo requerido para configurar la señal de inicio de conversión de los canales ADC configurados.



Figura 4.4.- Diagrama de flujo de la interrupción “TimerInt” para indicar los inicios de conversión del ADC.

4.2.3 Configuración del “TimerInt” para el parpadeo de LED’s en el ambiente de depuración y decremento de contadores.

Este temporizador se ajusto a un periodo de interrupción de 200 mseg para utilizarlo en la intermitencia de los LED’s en el ambiente de depuración y decrementar el contador de “Tiempo permitido de sincronización”; la configuración en el “Bean Inspector” del “TimerInt” en el IDE y diagrama de flujo, se visualizan en las figuras 4.5 y 4.6 respectivamente.

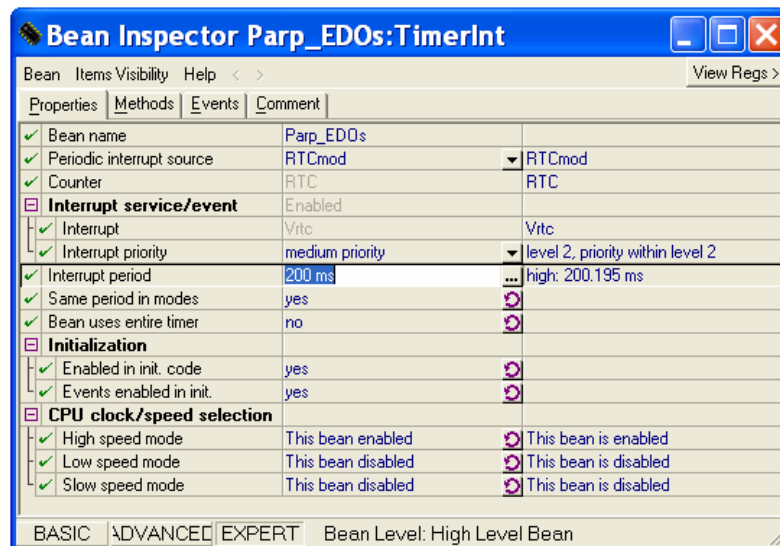


Figura 4.5.- Bean Inspector del “TimerInt” para el parpadeo de LEDs y decremento de contadores.

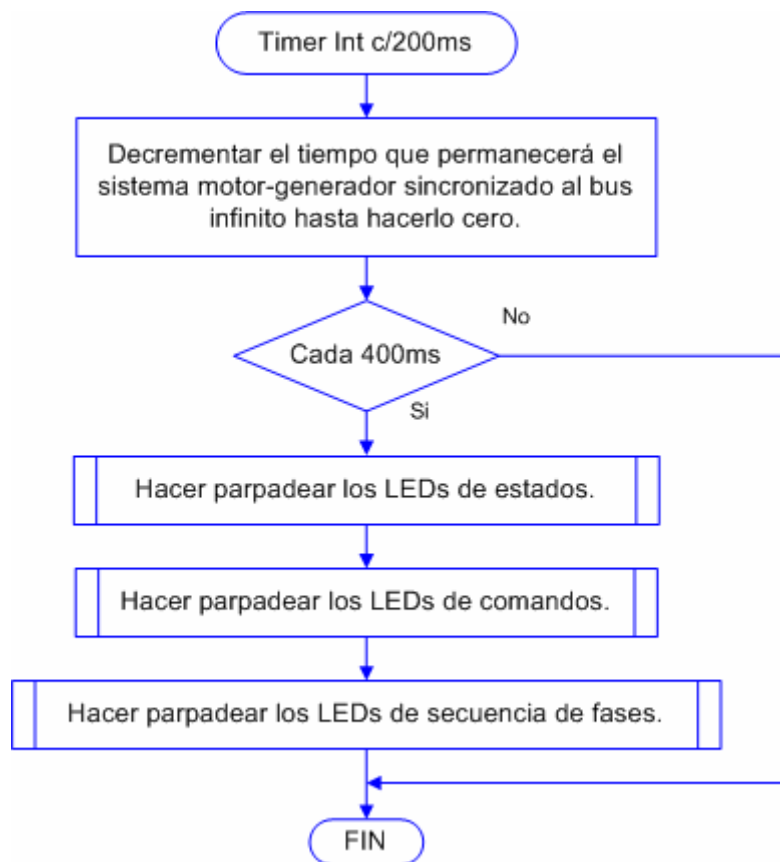


Figura 4.6.- Diagrama de flujo de la interrupción “TimerInt” para parpadeo de LEDs y decremento de contadores.

4.2.4 Configuración del Convertidor Analógico a Digital.

En la figura 4.7 se visualiza la configuración del “Bean Inspector” en el IDE del periférico interno ADC. Se habilitan 6 canales con una resolución de 12 bits cada uno; las tres fases de CFE son monitoreadas por los pines con los identificadores PTA1, PTA6 y PTA7, las tres fases de voltaje de salida del sistema motor-generador por los pines con los identificadores PTF6, PTF4 y PTF3.

Del diagrama de flujo de la figura 4.8, se observa que se mandan llamar dos funciones, estas funciones realizan parte de sus respectivos algoritmos, con la intención de aprovechar el tiempo que existe entre cada periodo de interrupción y no esperar hasta que se tengan las 96 muestras y empezarlas a calcular sus algoritmos; también

se evita estarlas almacenándolas, de manera que se aprovecha el área de memoria del microcontrolador.

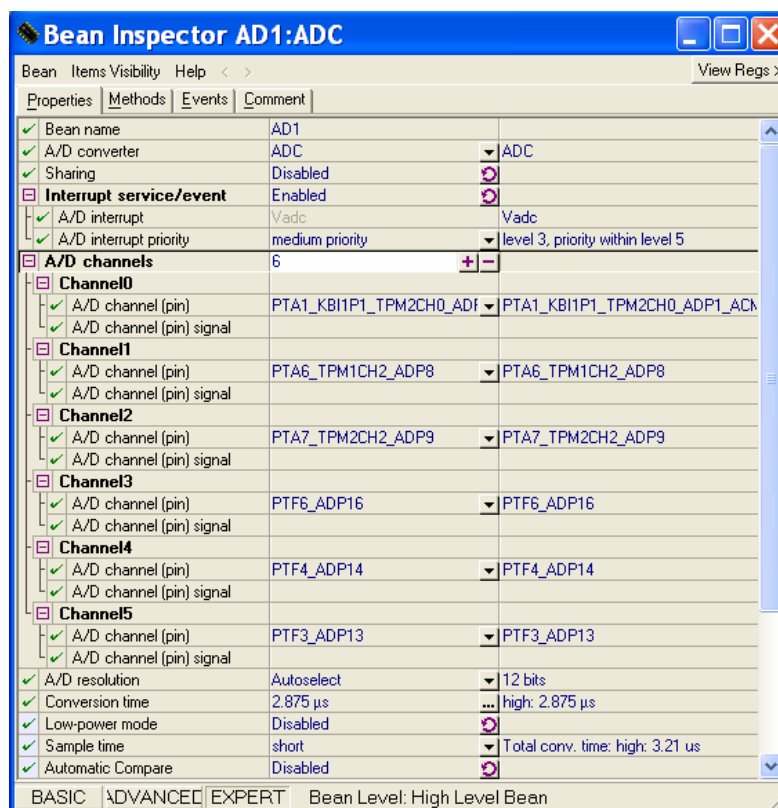


Figura 4.7.- Bean Inspector del ADC

Del diagrama de flujo de la figura 4.8, cabe mencionar, que cada vez que las funciones “Frecuencímetro parte inicial” y “DFT parte inicial” termina de procesar la fracción de sus algoritmos correspondientes, lo indican activando sus banderas.

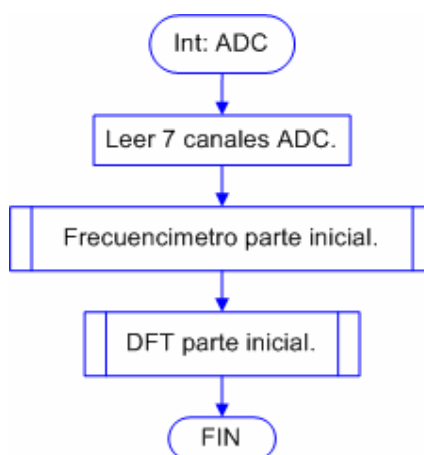


Figura 4.8.- Diagrama de flujo de la interrupción ADC.

4.3 Diagrama de flujo del programa principal (Main).

La programación se realizó en base a una máquina de estados finita, que en función del estado actual del sincronoscopio, ejecuta el código correspondiente. Del diagrama de flujo de la figura 4.9, se observa que antes de ejecutarse cualquier estado, se pregunta si el algoritmo inicial de la DFT se ha completado, para así, desarrollar su cómputo final y determinar después, que estado va ejecutarse; finalmente, actualiza las salidas (contactores de la compañía suministradora de energía eléctrica y del sistema motor-generador) y despliega resultados en el ambiente de depuración del microcontrolador.

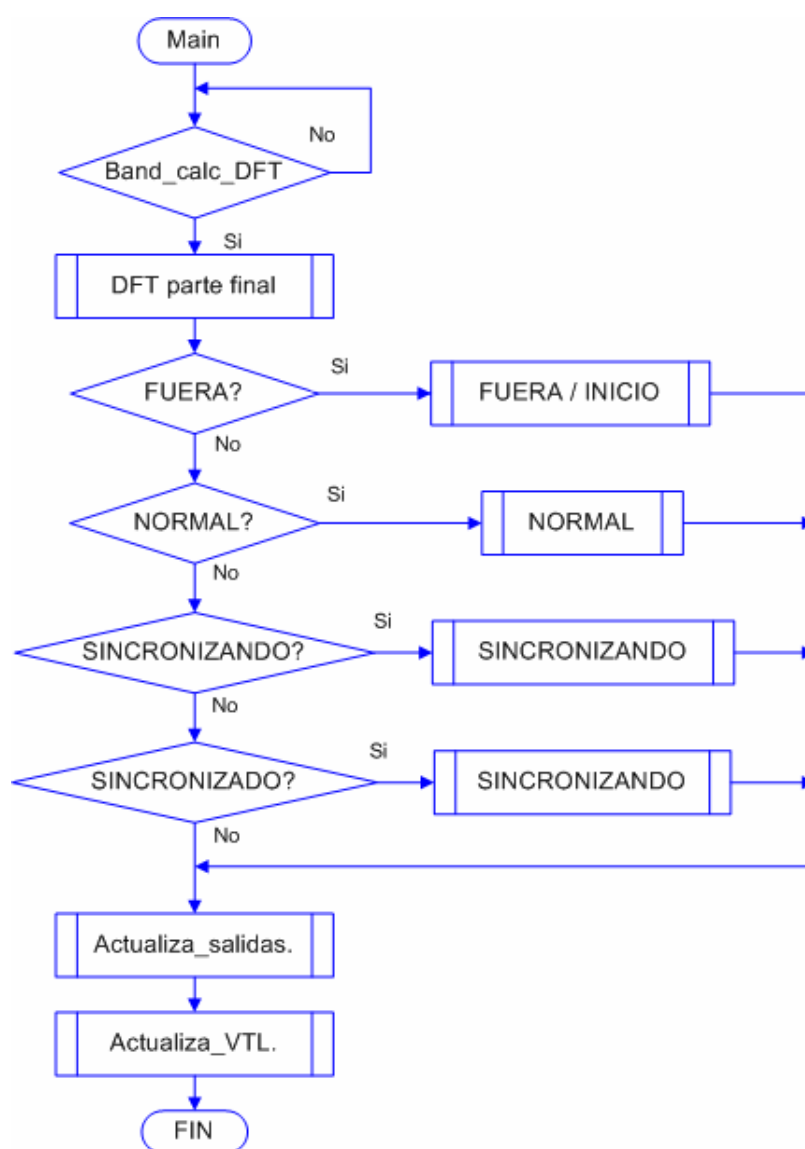


Figura 4.9.- Diagrama de flujo del programa principal.

El diagrama de flujo de la función “Estado FUERA / INICIO” se muestra en la figura 4.10.

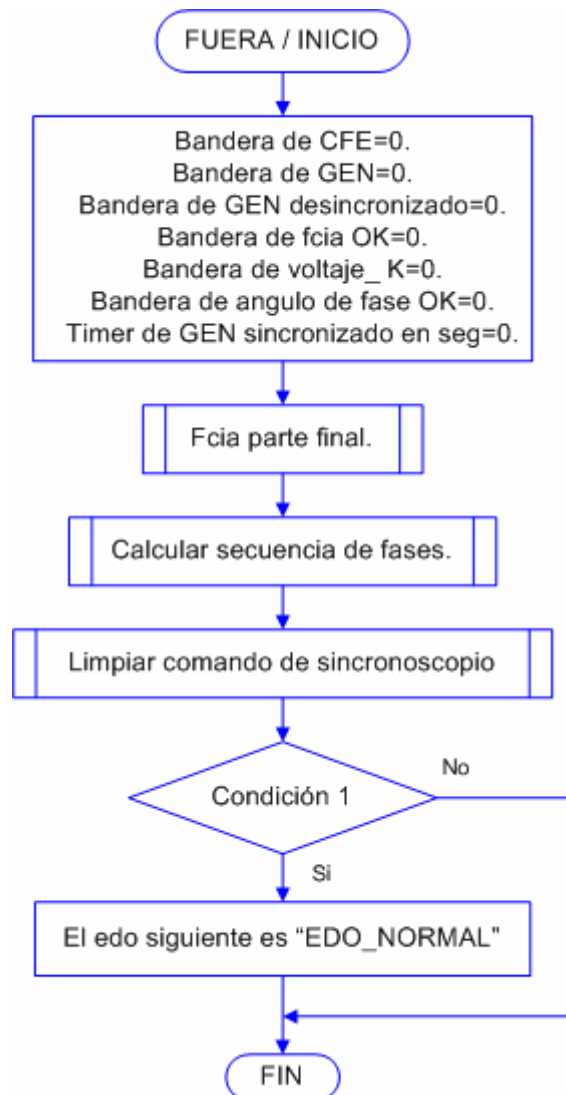


Figura 4.10.- Diagrama de flujo de la función “Estado FUERA / INICIO”

El diagrama de flujo de la función “Estado NORMAL” se muestra en la figura 4.11.

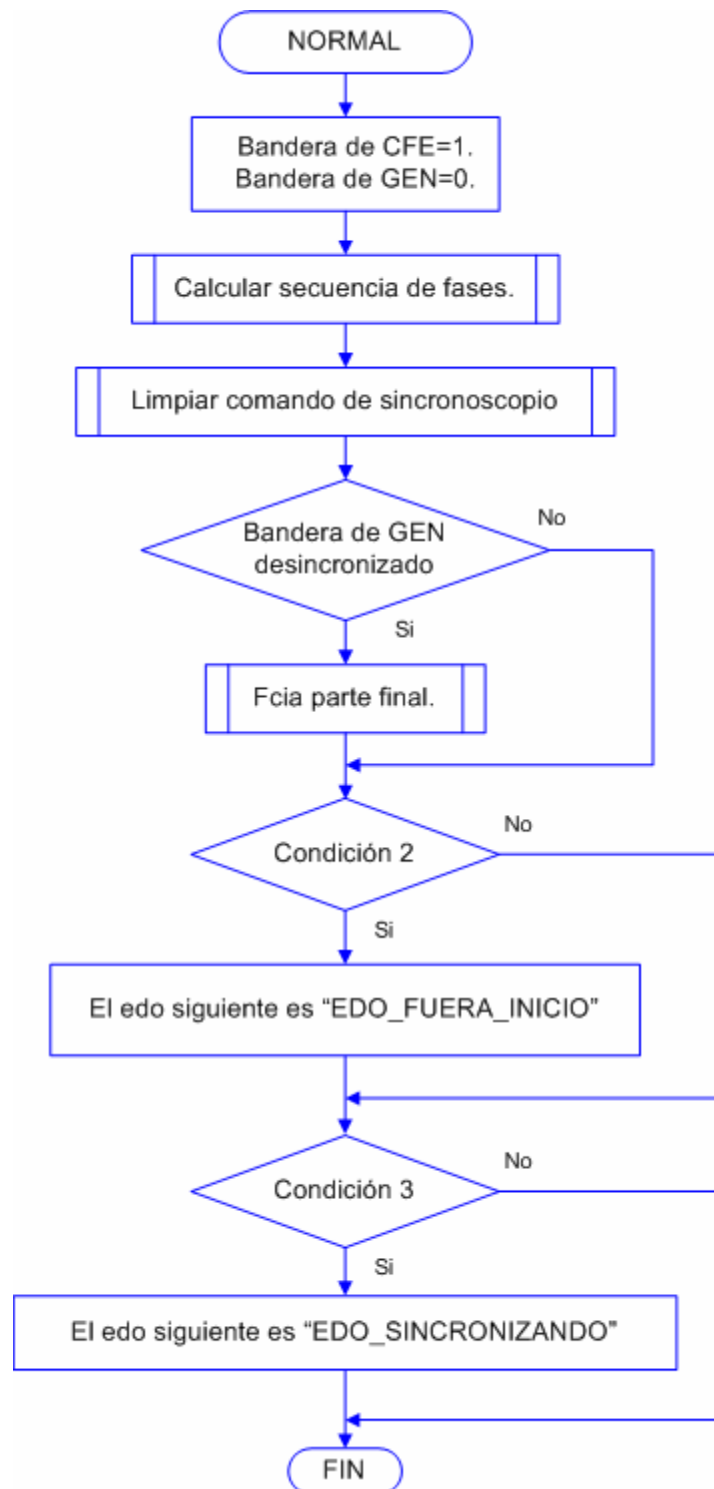


Figura 4.11.- Diagrama de flujo de la función “Estado NORMAL”

El diagrama de flujo de la función “Estado SINCRONIZANDO” se muestra en la figura 4.12.

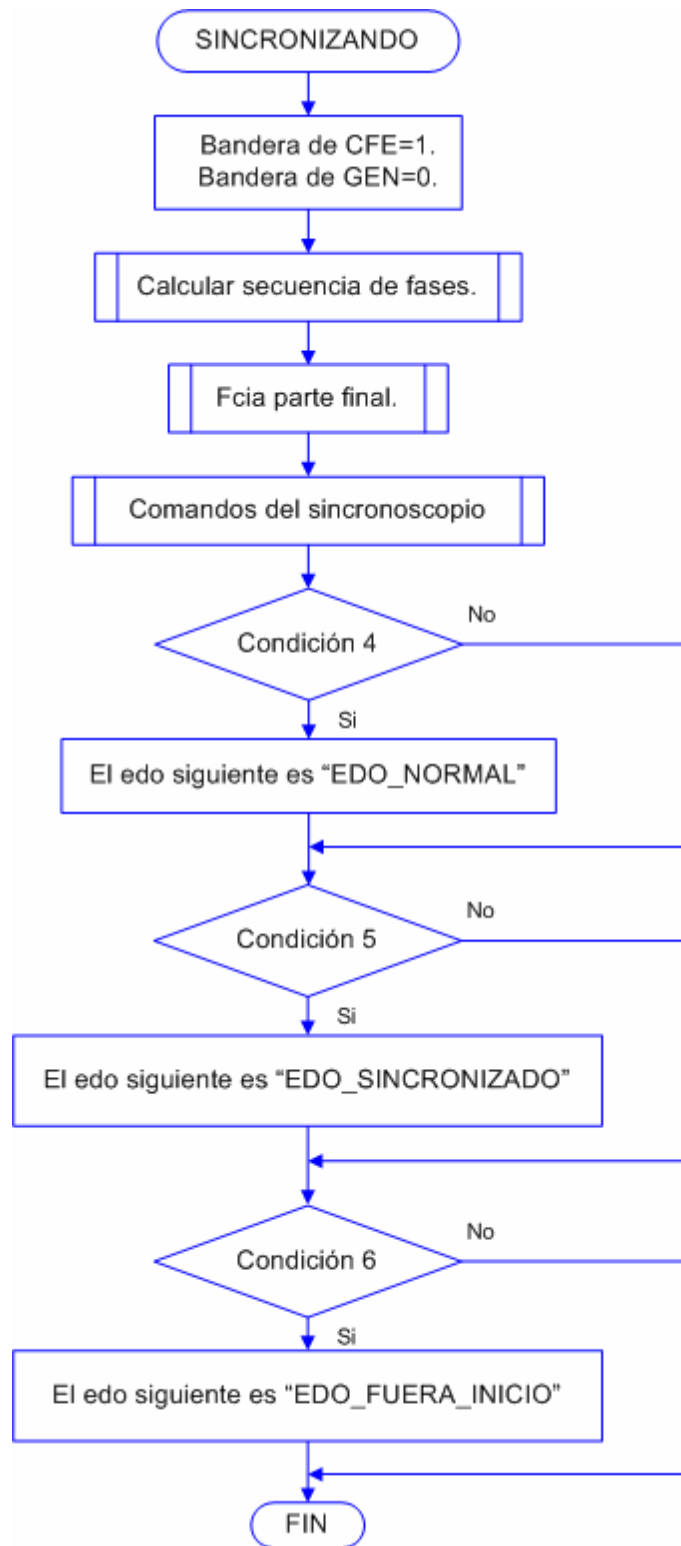


Figura 4.12.- Diagrama de flujo de la función “Estado SINCRONIZANDO”

El diagrama de flujo de la función “Estado SINCRONIZADO” se muestra en la figura 4.13.

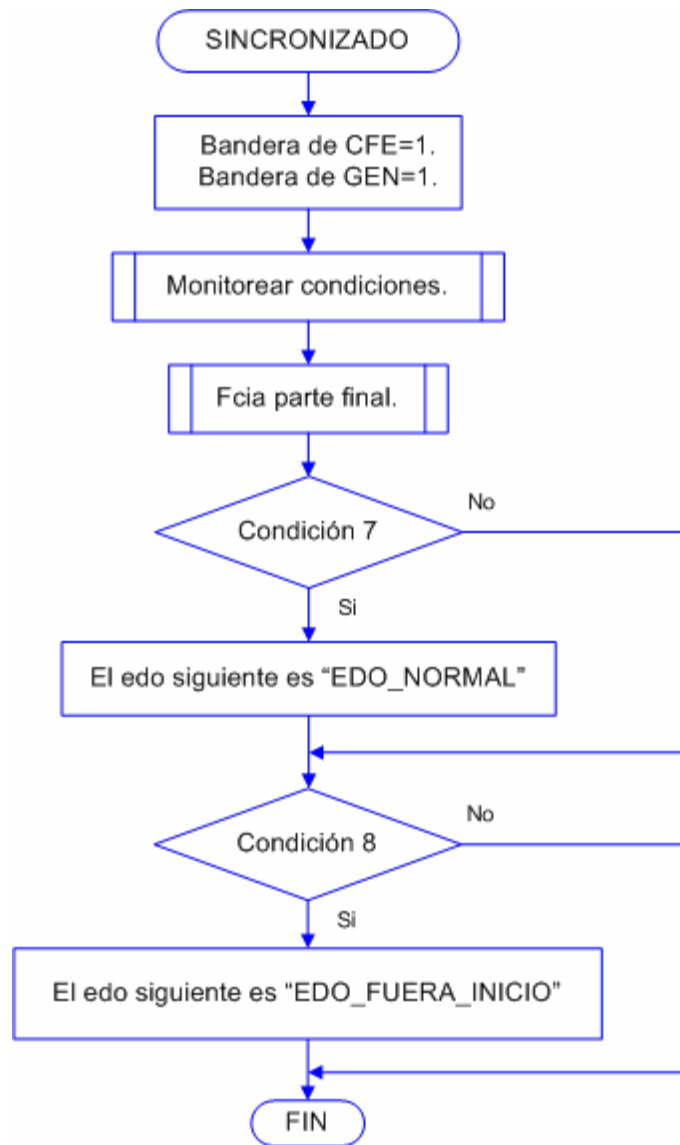


Figura 4.13.- Diagrama de flujo de la función “Estado SINCRONIZADO”

El diagrama de flujo de la función “Actualiza salidas” se muestra en la figura 4.14.

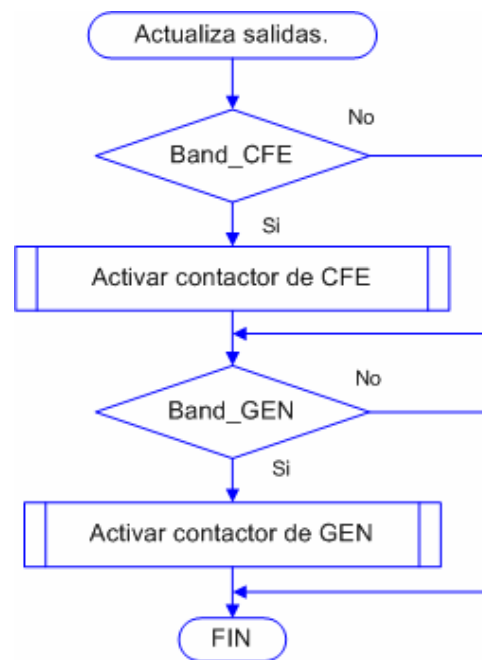


Figura 4.14.- Diagrama de flujo de la función “Actualiza salidas”

4.4 Algoritmo para el cálculo de la DFT en tiempo real.

La ecuación 4.2 realiza la transformada de la secuencia $\{x[i]\}$ de longitud $L \leq N$, en una secuencia de muestras en frecuencia $\{X[k]\}$ de longitud N y se denota como la transformada discreta de Fourier (DFT) de $x[i]$. [27].

$$X[k] = \sum_{i=0}^{N-1} x[i] e^{-j2\pi kn/N} \quad (4.2)$$

Partiendo de la relación 4.2, por la regla de Euler se obtienen las componentes real e imaginaria; ecuación 4.3 y 4.4 respectivamente.

$$\text{Re } X[k] = \sum_{i=0}^{N-1} x[i] \cos(2\pi ki / N) \quad (4.3)$$

$$\text{Im } X[k] = -\sum_{i=0}^{N-1} x[i] \text{sen}(2\pi ki / N) \quad (4.4)$$

Las relaciones 4.3, 4.4 y 2 tablas almacenadas en memoria (una senoidal y otra cosenoidal de un ciclo, normalizadas y de 96 valores cada una) servirán para computar la primera parte del cálculo de la DFT.

Las multiplicaciones acumulativas que describen las ecuaciones 4.3 y 4.4, se realizan entre cada periodo de muestreo, para evitar almacenar los 96 valores por cada canal ADC configurado y optimizar el tiempo de procesamiento, de las componentes rectangulares de la señal sinusoidal que se quiera analizar.

Con esta metodología se obtiene por ciclo de una señal de 60 Hz, las multiplicaciones acumulativas por canal ADC configurado; hasta aquí comprende la primer parte del procesamiento del cálculo de la DFT en tiempo real, conjuntado en la función “Calcula DFT, parte inicial”. En el diagrama de flujo de la figura 4.15, se ilustra lo antes descrito.

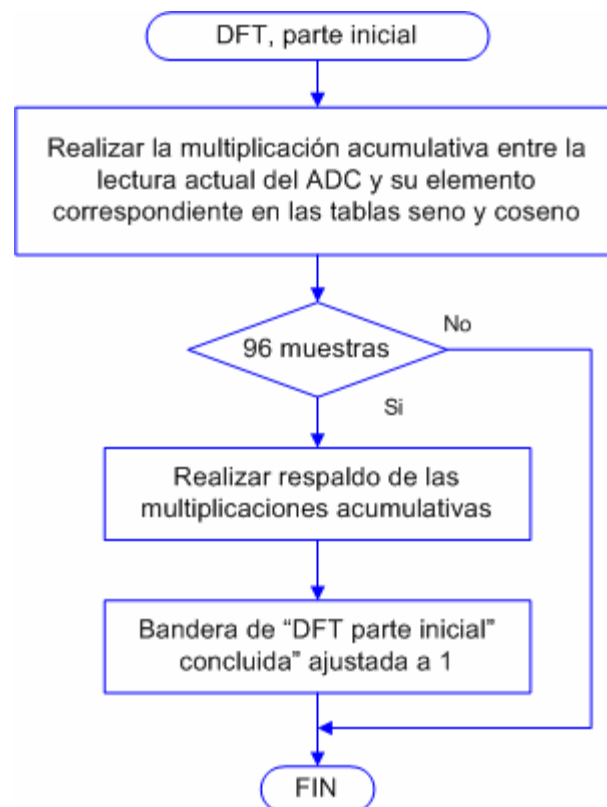


Figura 4.15.- Diagrama de flujo de la función “Calcula DFT, parte inicial”

A continuación, se requiere que los cálculos computados con las relaciones 4.3 y 4.4 sean evaluados en 4.5 y 4.6, obteniendo así las componentes rectangulares de las señales sinusoidales muestreadas.

$$\operatorname{Re} \bar{X}[k] = \frac{\operatorname{Re} X[k]}{N/2} \quad (4.5)$$

$$\operatorname{Im} \bar{X}[k] = \frac{\operatorname{Im} X[k]}{N/2} \quad (4.6)$$

Finalmente, es conveniente trabajar en coordenadas esféricas; por lo que será necesario calcular la magnitud y ángulo del par de coordenadas cartesianas con las ecuaciones 4.7 y 4.8 respectivamente.

$$MAG = \sqrt{\left((\operatorname{Re} \bar{X}[k])^2 + (\operatorname{Im} \bar{X}[k])^2 \right)} \quad (4.7)$$

$$\theta = \sec\left(\frac{\operatorname{Re} \bar{X}[k]}{MAG} \right) \quad (4.8)$$

Estos últimos cálculos corresponden a la segunda parte del análisis de la DFT y se conjuntan en la función “Calcula DFT, parte final”, ver diagrama de flujo de la figura 4.16.

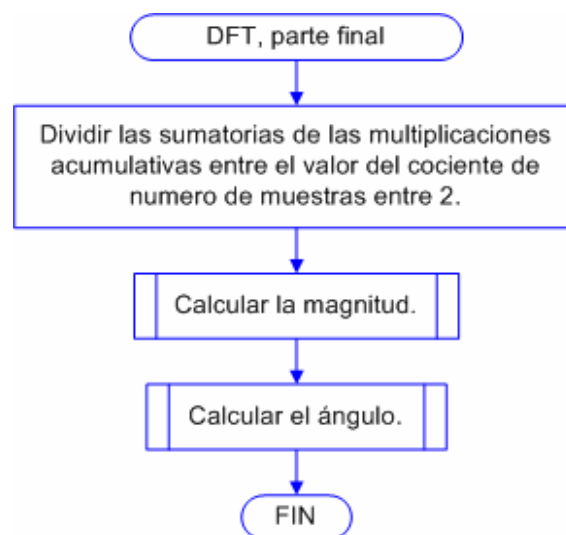


Figura 4.16.- Diagrama de flujo de la función “Calcula DFT, parte final”

Los diagramas de flujo de las funciones “Raíz” y “Fase” requeridas en la función “Calcula DFT, parte final”; se muestran en las figura 4.17 y 4.18 respectivamente.

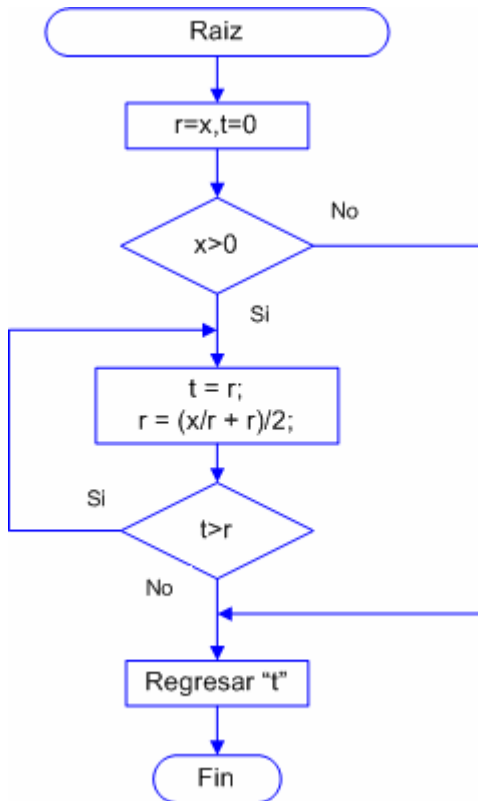


Figura 4.17.- Diagrama de flujo de la función “raíz”.

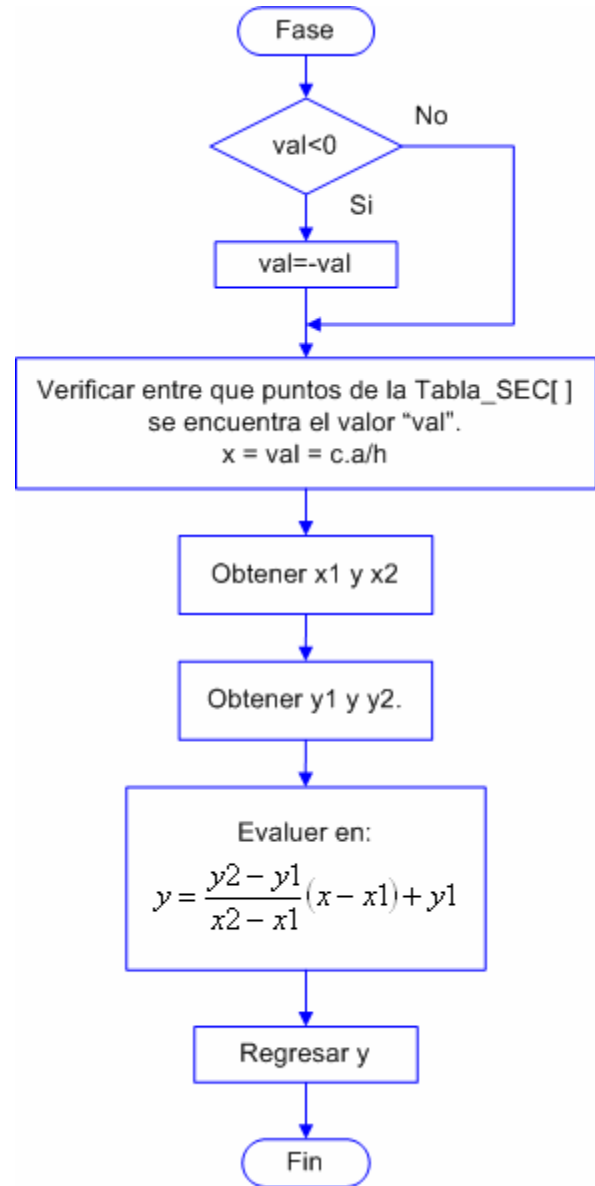


Figura 4.18.- Diagrama de flujo de la función “Fase”.

Para calcular el ángulo correspondiente al par de coordenadas cartesianas, se auxilio de una tabla en memoria de 65 elementos correspondiente al comportamiento de la función trigonométrica “Secante” y de la técnica de interpolación de primer orden.

4.5 Metodología del cálculo de la secuencia de fase.

Con ayuda de los cálculos previos de la DFT, es posible determinar la secuencia de fases de las conexiones presentes en el sincronoscopio. Indicando así, si es posible llevar a cabo o no la sincronización del Bus infinito con el sistema motor-generador. En el diagrama de flujo de la figura 4.19 se describe la metodología de determinación de la secuencia de fases.

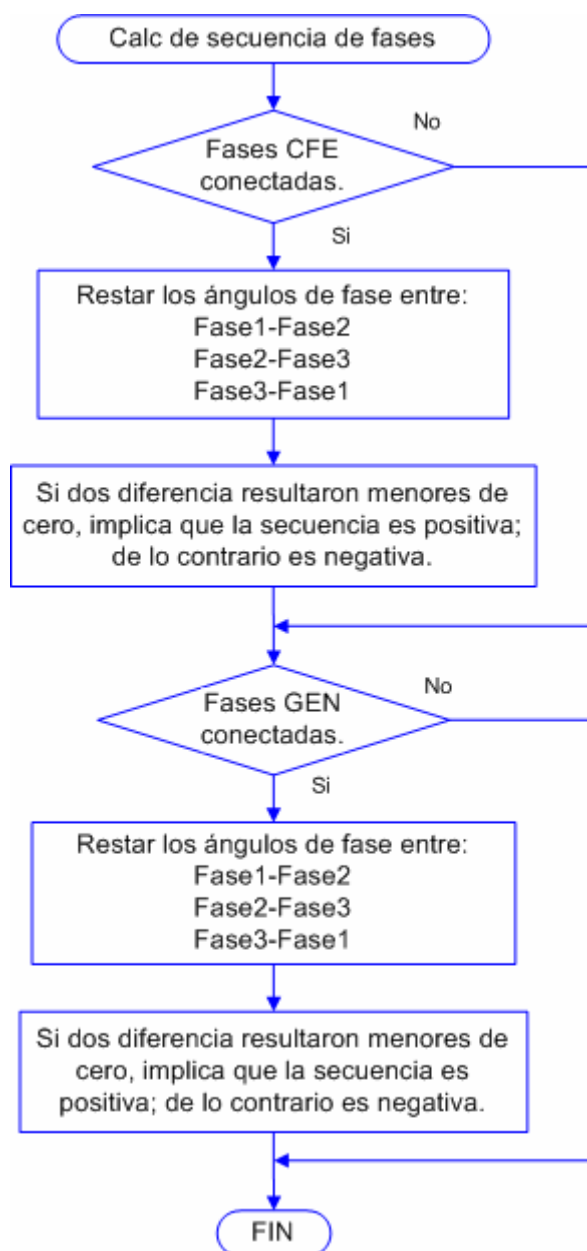


Figura 4.19.- Diagrama de flujo de la función “Determina secuencia de fase”.

4.6 Metodología del cálculo de la frecuencia en tiempo real.

La metodología implementada para determinar la frecuencia en tiempo real de la señal sinusoidal muestreada por el ADC, consiste en distribuir el algoritmo en dos partes; el primero se encarga de detectar los cruces por cero y a partir de ese instante contar el número de muestras que se obtengan hasta el próximo cruce por cero y la segunda parte del algoritmo consiste en determinar la frecuencia en base a los datos obtenidos en la parte inicial del algoritmo.

Debido a que las señales sinusoidales de voltaje fueron montadas sobre un nivel de CD para su codificación por los canales ADC del microcontrolador, las señales precisamente no realizan un cruce por cero, por lo que se monitorea su cruce a través del “Offset” agregado en su acondicionamiento.

Debido a las características intrínsecas del sistema motor-generador, las señales correspondientes a la salida de voltaje trifásico, presentan una distorsión considerable; ver figura 4.20.

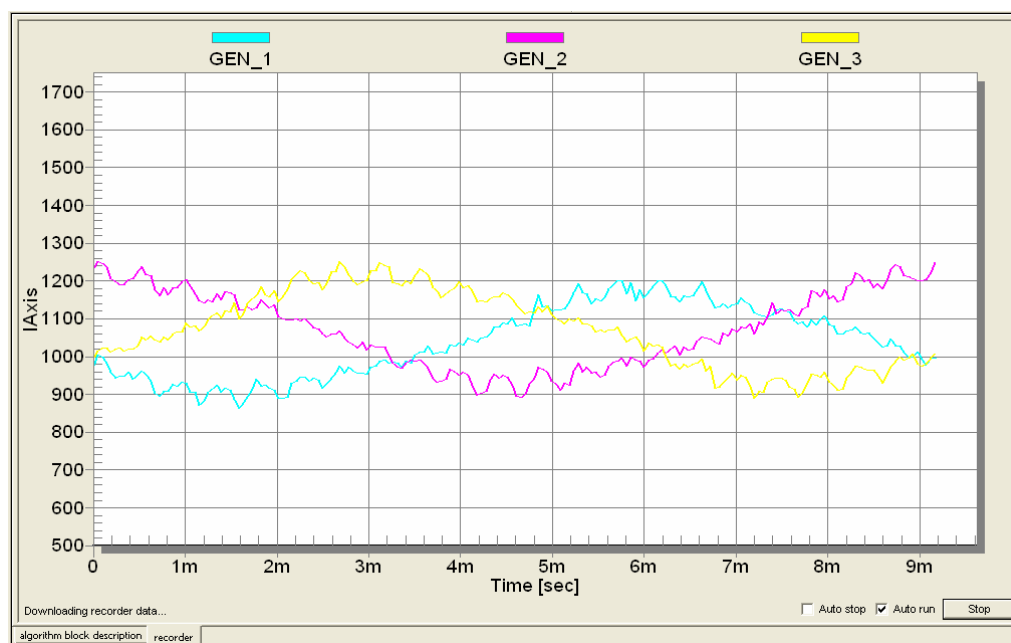


Figura 4.20.- Señal trifásica de salida del sistema motor-generador al ser activado (30Hz / 40VCA) con secuencia de fases positiva.

Debido a esta distorsión, la búsqueda de los cruces por el "Offset" se realizara cada 4 muestras, así como lo ejemplifica la figura 4.21.

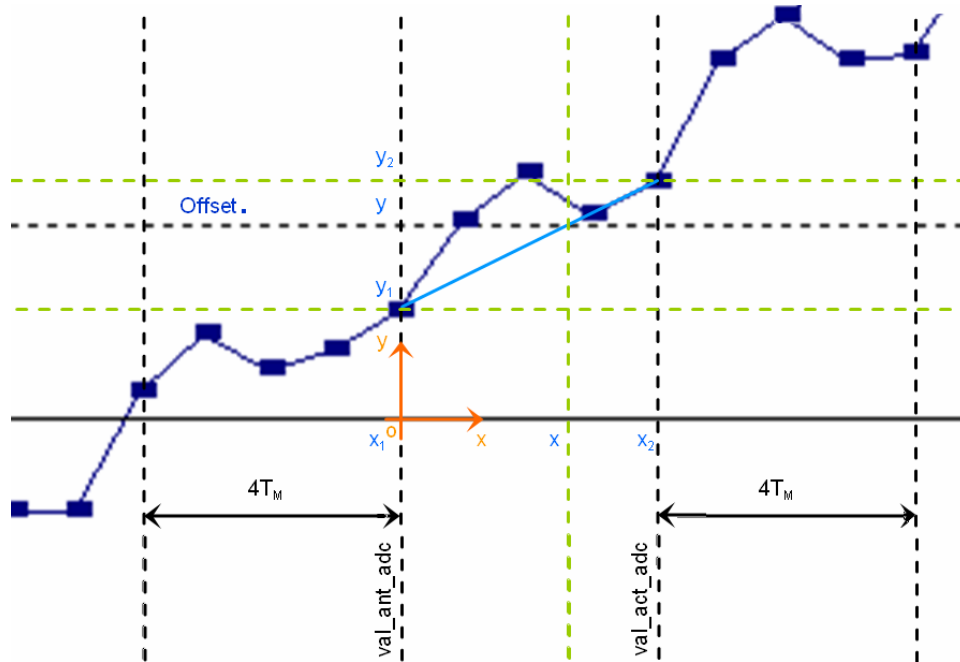


Figura 4.21.- Búsqueda de cruces por el "Offset" cada 4 muestras.

De la figura 4.21, se visualiza que existe un error, debido al tiempo que no se considera después del cruce por el "Offset" y la siguiente muestra que sirvió para detectarlo (x_2-x); por lo que se debe de realizar una interpolación de primer orden para considerarlo; en el diagrama de flujo de la figura 4.22, se muestra el algoritmo de la interpolación de primer orden.

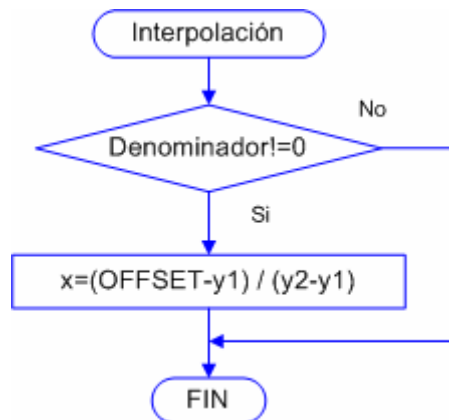


Figura 4.22.- Búsqueda de cruces por el "Offset" cada 4 muestras.

El diagrama de flujo de la función “Frecuencímetro, parte inicial” se muestra en la figura 4.23.

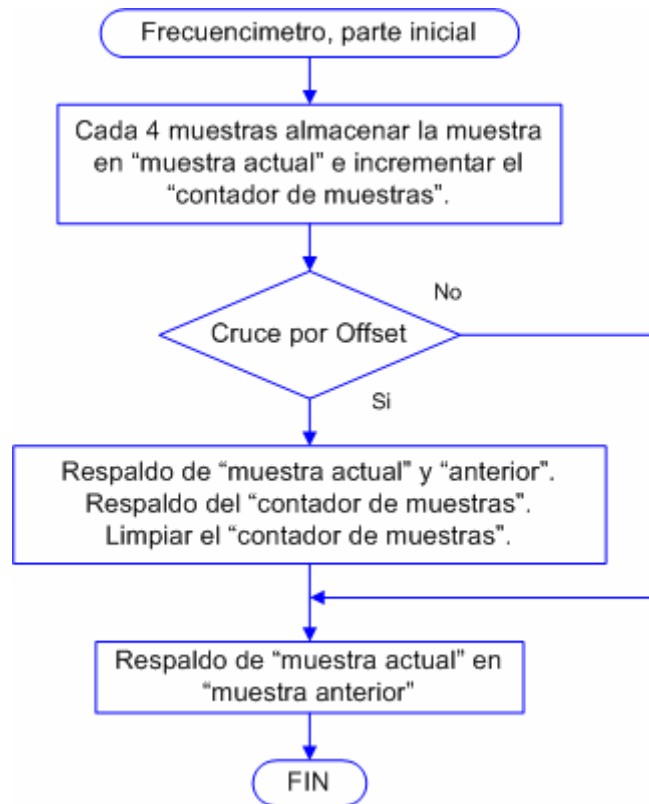


Figura 4.23.- Diagrama de flujo de la función “Frecuencímetro, parte inicial”.

El diagrama de flujo de la función “Frecuencímetro, parte final” se muestra en la figura 4.24.

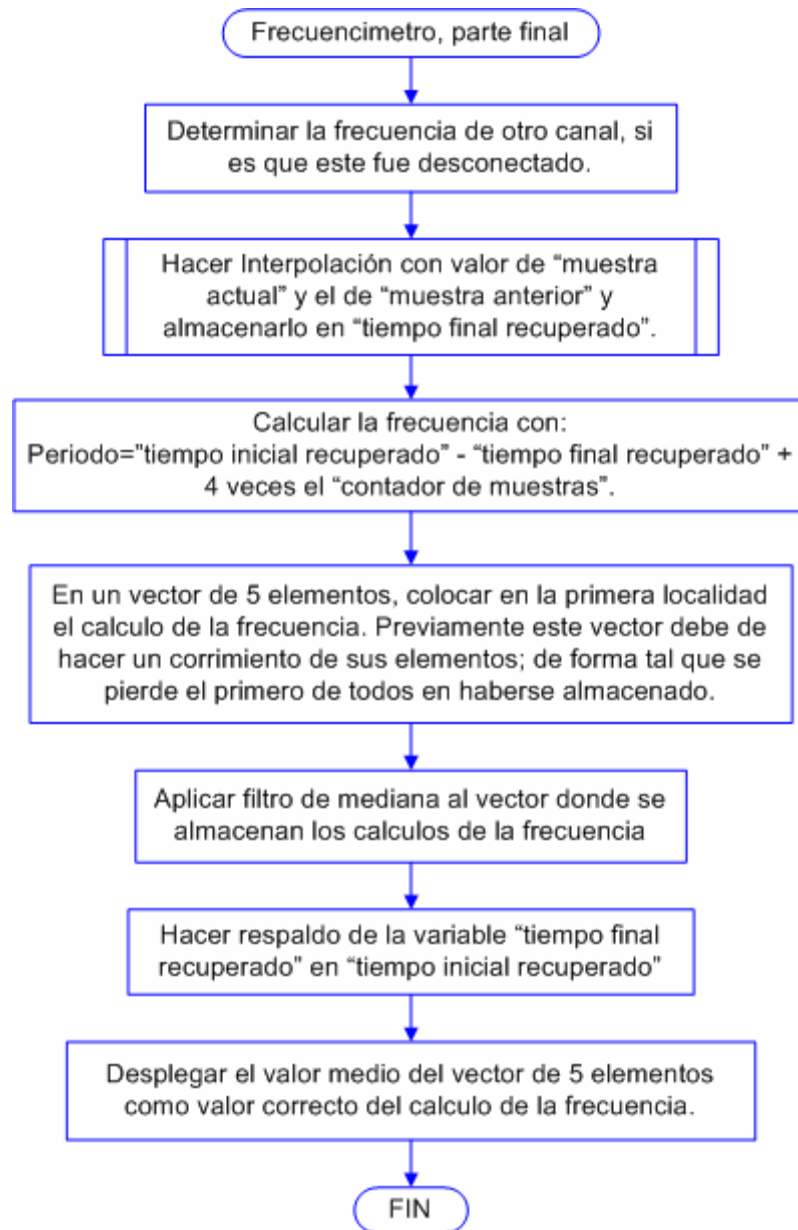



Figura 4.24.- Diagrama de flujo de la función “Frecuencímetro, parte final”.

Este algoritmo, en función del periodo de muestreo, respondió de manera satisfactoria, determinando la frecuencia en el intervalo de 25 a 75 Hz; lo cual resulta conveniente para aplicaciones de ingeniería eléctrica.

4.7 Ambiente de depuración del microcontrolador.

Para poder depurar y/o monitorear la secuencia de operación del programa implementado en el microcontrolador, en el ambiente de depuración, por medio de las variables utilizadas, es necesario que estas hayan sido declaradas como globales. En este espacio podemos agregar instrumentos de visualización, tales como barras de nivel, LED's indicadores, visualizadores analógicos de aguja y en sistema de ejes coordenados "x-y", indicadores de 7 segmentos, visualizadores de variables como texto, cajas de texto estáticas, entre otros.

4.7.1 Inicialización.

Después de dar clic en  **Debug**, en el IDE del microcontrolador, el software mostrara la ventana de la figura 4.25, en la que se define el puerto y ajustes del mismo para establecer la comunicación con la tarjeta de desarrollo "DEMOQE" y descargar el programa diseñado hecho esto, se da clic en "Connect (Reset)".

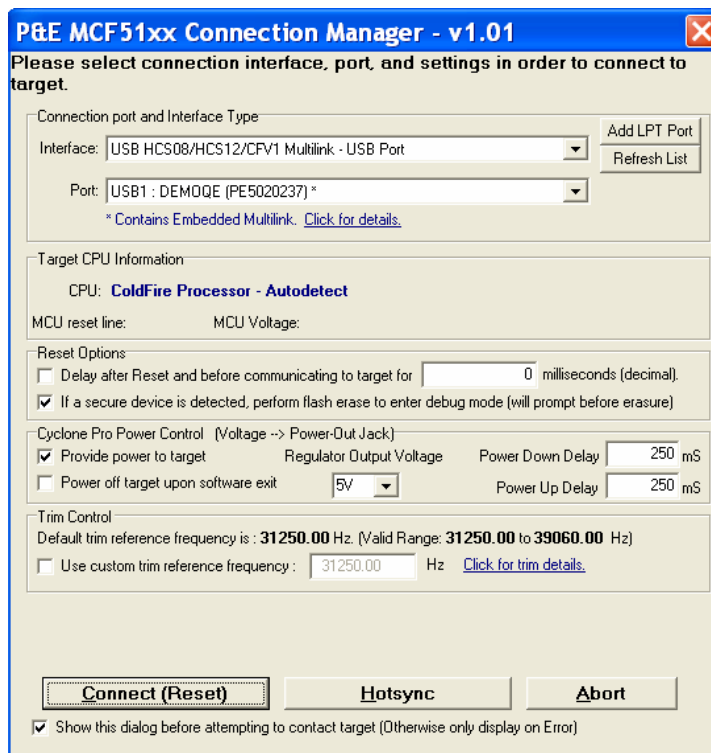


Figura 4.25.- Diagrama de flujo del programa principal.

La ventana mostrada en la figura 4.26 será desplegada a continuación y en ella se muestran tanto el programa implementado en lenguaje C como sus variables globales y por funciones definidas, el código ASM generado, el código de máquina en el área de memoria del microcontrolador, sus registros en la ALU y el estado del microcontrolador. También se pueden agregar puntos de ruptura en el código para depurarlo y/o monitorearlo, teniendo de igual manera la posibilidad de hacerlo en tiempo real.

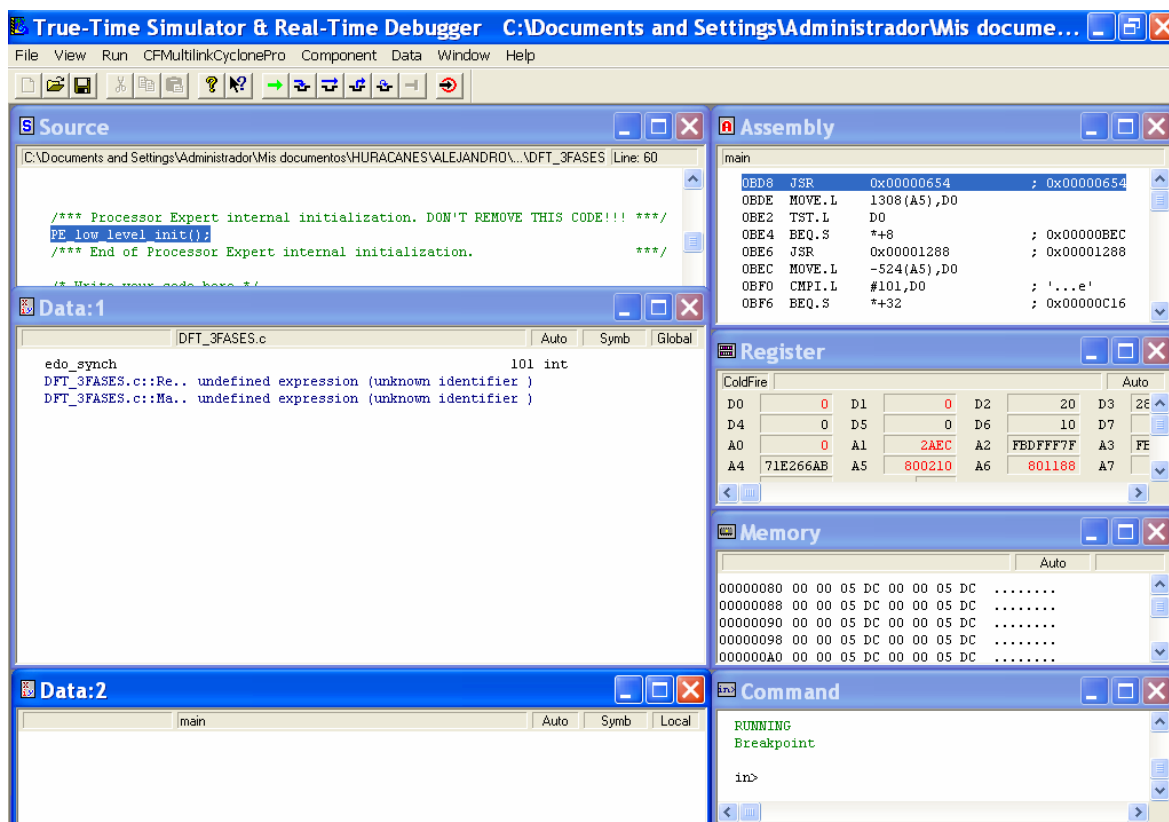


Figura 4.26.- Detalles del programa implementado y estado del microcontrolador.

A continuación en la barra de menú, se da clic en “Component” y se escoge “Open”, ver figura 4.27.

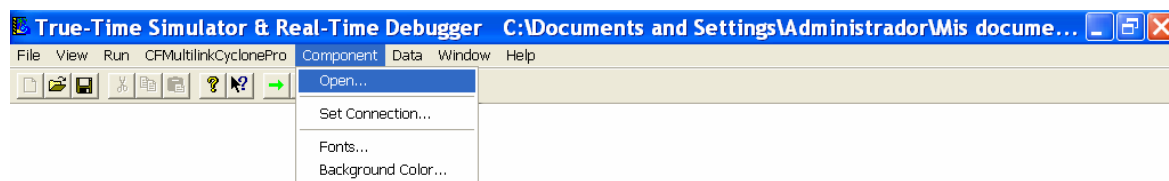


Figura 4.27.- Agregando un componente al ambiente de depuración.

En la figura 4.28, se muestran los diferentes componentes que se pueden agregar al ambiente de depuración; para esta aplicación se escoge el denominado “Visualizationtool”.

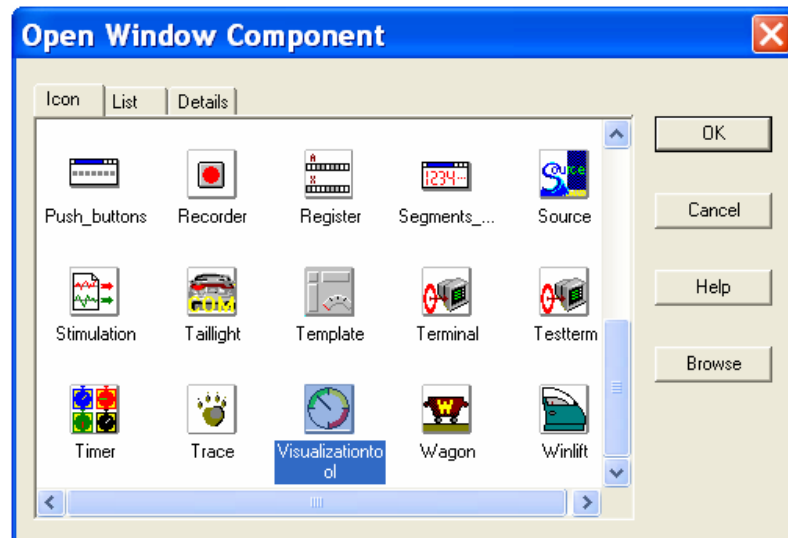


Figura 4.28.- Selección del componente “Visualizationtool” en el ambiente de depuración.

La ventana mostrada en la figura 4.29, es el área de trabajo para agregar las herramientas de monitoreo de las variables globales utilizadas en el programa implementado.

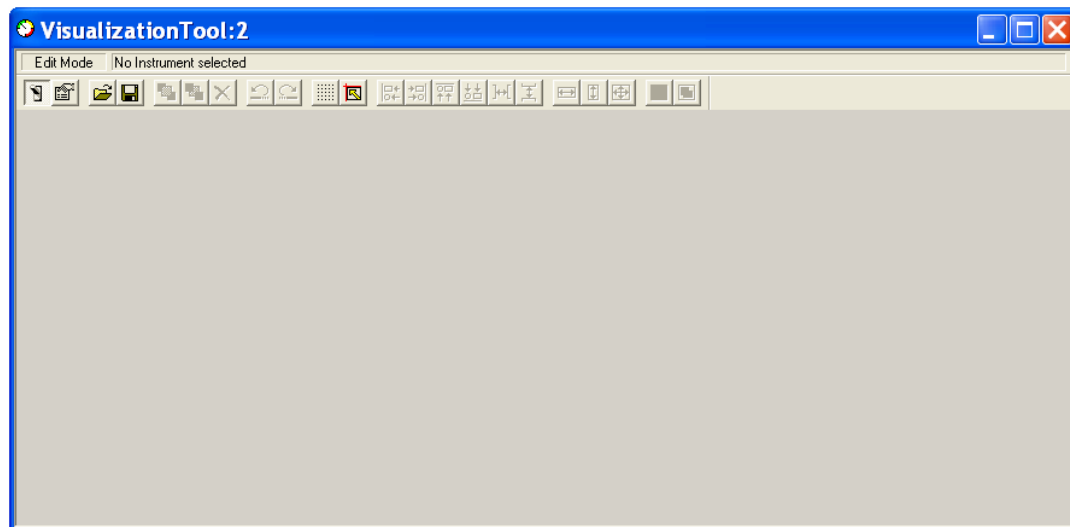


Figura 4.29.- Área de trabajo, para agregar las herramientas de monitoreo.

Para agregar un instrumento, deberá darse un clic con el botón derecho sobre el área de trabajo, seleccionar “Add New Instrument” y escoger uno de los 14 disponibles; en la figura 4.30, se muestra la selección de uno de visualización analógica y en la figura 4.31 sus propiedades ajustadas para una variable del programa.

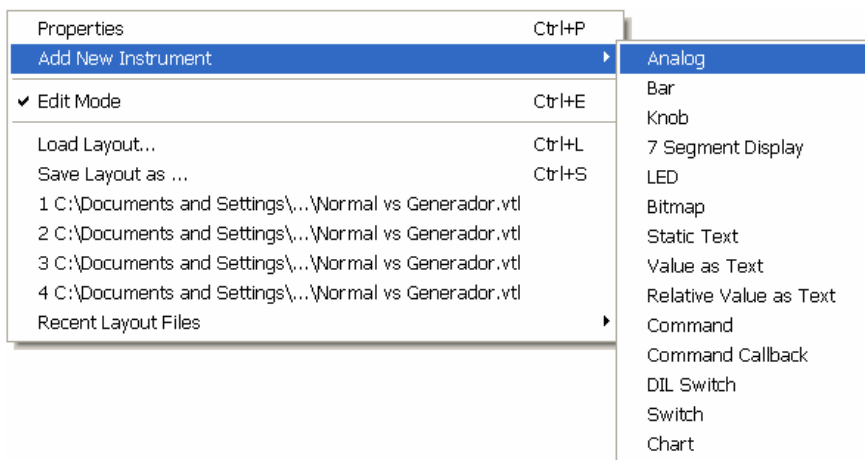


Figura 4.30.- Selección de un instrumento de visualización analógica.

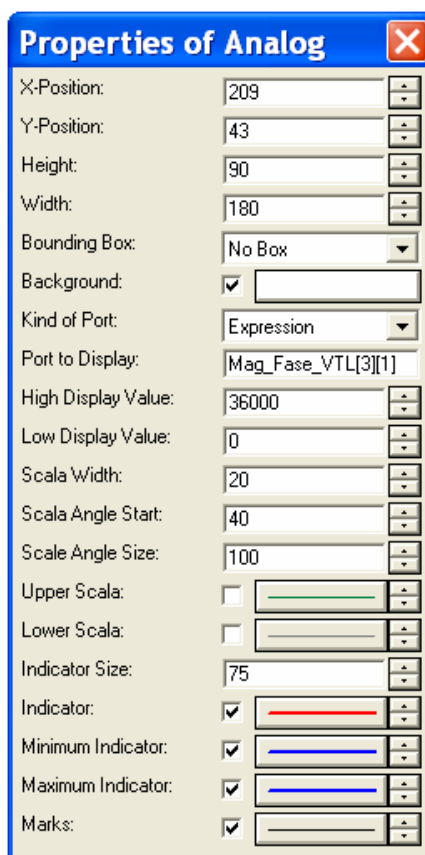


Figura 4.31.- Propiedades del instrumento de visualización analógica.

En la figura 4.32 a). se visualiza el instrumento de visualización analógica antes de ser configurado y en la 4.32 b). después de ser ajustado y en conjunto con otros instrumentos como: “Value as text” y “Static text”.



Figura 4.32.- Instrumento de visualización analógica.

De igual manera se agrega un instrumento de visualización en sistema de ejes coordenados “x-y”; del cual sus propiedades ajustadas para el monitoreo de una variable global del programa implementado, se muestra en la figura 4.33.

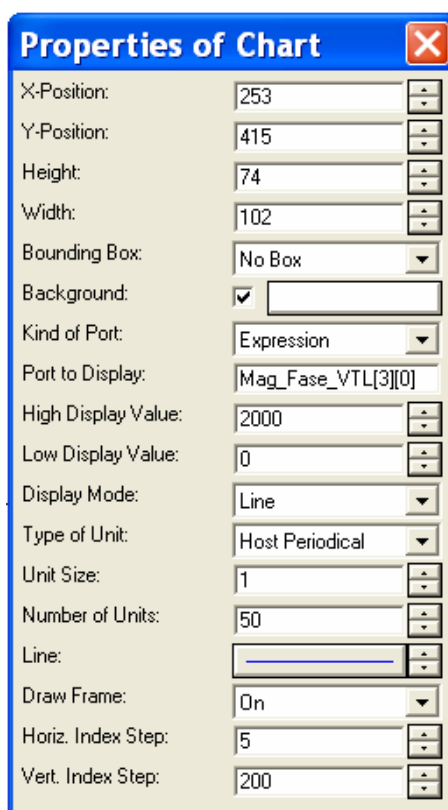


Figura 4.33.- Propiedades del instrumento de visualización en ejes coordenados “x-y”.

En la figura 4.34 a). se visualiza el instrumento de visualización en sistema de ejes coordenados “x-y” antes de ser configurado y en la 4.34 b). después de ser ajustado y en conjunto con otros instrumentos como: “Value as text” y “Static text”.



Figura 4.34.- Instrumento de visualización analógica en sistema de ejes coordenados “x-y”.

Para agregar una imagen al área de trabajo, se escoge el instrumento “Bitmap”, del cual, en sus propiedades se determina la ruta de la imagen a ser insertada (la imagen debe tener extensión BMP); la imagen puede ser complementada con algunos de los instrumentos de visualización que se disponen.

En la figura 4.35 a), se muestra la imagen insertada y en la 4.35 b) se muestra complementada con LED's del propio ambiente de depuración.

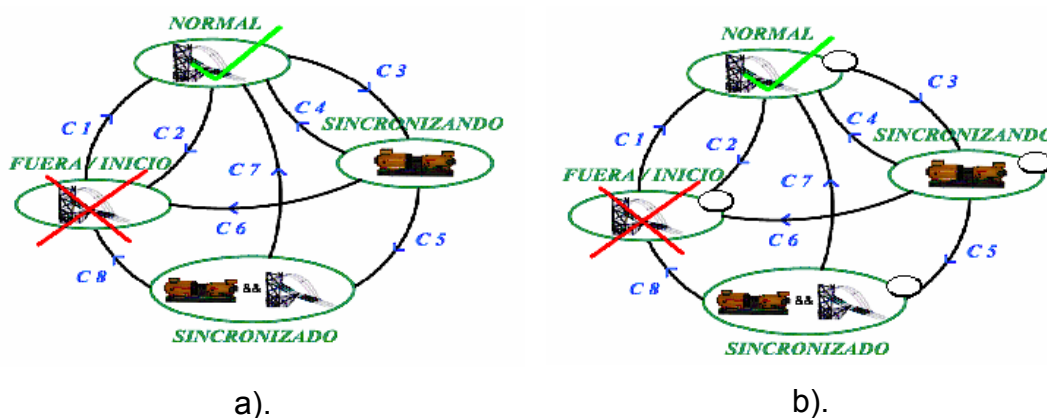


Figura 4.35.- Imágenes dentro del ambiente de depuración.

4.7.2 Vista final del diseño del ambiente de depuración.

Finalmente el ambiente de depuración, queda integrado por visualizadores de ángulos de fase, voltaje y frecuencia tanto del Bus infinito como del sistema motor-generador.

Sincronoscopio virtual indicando las diferencias de fase que existen entre una fase del Bus infinito y del sistema motor-generador.

Diagrama de estados con LED's indicadores del estado actual que se esta ejecutando en el microcontrolador.

Comandos visuales de ajuste para la frecuencia y voltaje del sistema motor-generador a ser sincronizado con el Bus infinito.

Indicador de secuencia de fases para el Bus infinito y el sistema motor-generador; ver figura 4.36.

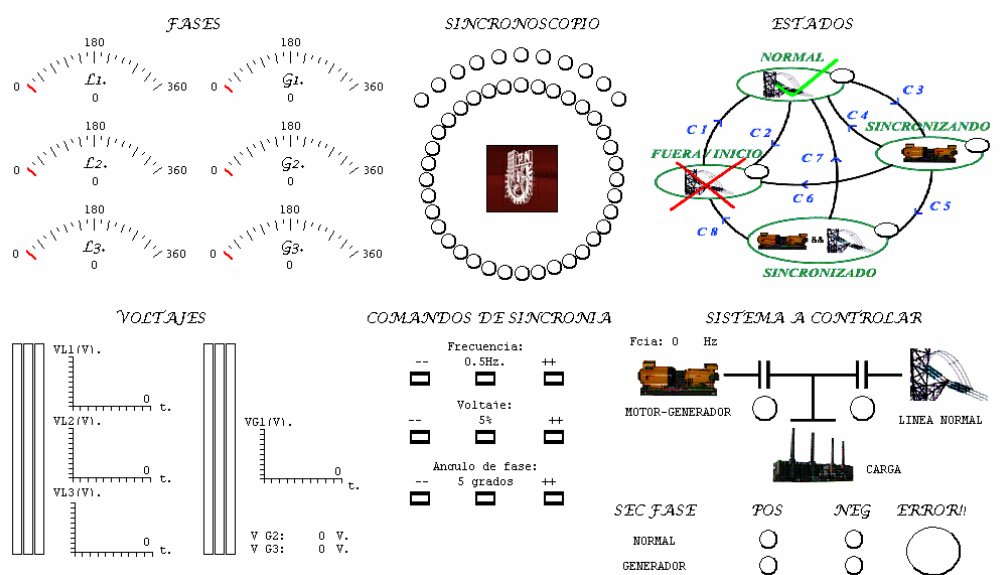


Figura 4.36.- Vista final de los instrumentos de monitoreo en el ambiente de depuración.

CAPÍTULO 5

PRUEBAS

En este capítulo se reportan los resultados de las pruebas realizadas al sincronoscopio: respuesta de la implementación del algoritmo de la Transformada Discreta de Fourier (DFT) en el microcontrolador en tiempo real, ante las señales proporcionadas por el sistema motor-generador, que presentan una considerable distorsión intrínseca debida a las características físicas del generador, respuesta del algoritmo para determinar la frecuencia en tiempo real, determinación de la secuencia de fase conectada, secuencia de prueba con error de secuencia de fases y sincronización con secuencia positiva y negativa.

Los oscilogramas se obtienen con la herramienta FreeMaster de Freescale Semiconductor, Inc; que en conjunto con el DSP 56F8323 en la tarjeta de desarrollo 56F8300 DEMO de Motorola, fungieron como un osciloscopio digital de hasta 8 canales.

El ADC del DSP tiene una resolución de 12 bits, justificados a la derecha en una variable sin signo de 16 bits; las variables donde se codificaron las señales sinusoidales trifásicas de voltaje del Bus infinito y del sistema motor-generador, se les aplicó un corrimiento de 4 bits a la derecha, para reducir el valor de su magnitud.

5.1 Secuencia de pruebas del sincronoscopio.

- Sincronización con secuencias de fases positivas.
- Sincronización con secuencias de fases negativas.
- Respuesta ante secuencia de fases opuestas.
- Sacar de sincronía al sistema motor-generador con comandos generados en el IDE.

5.1.1 Sincronización con secuencias de fases positivas.

El sincronoscopio inicia en el estado “FUERA / INICIO” con CFE y el sistema motor-generator desactivados, las señales presentes se muestran en la Fig. 5.1.

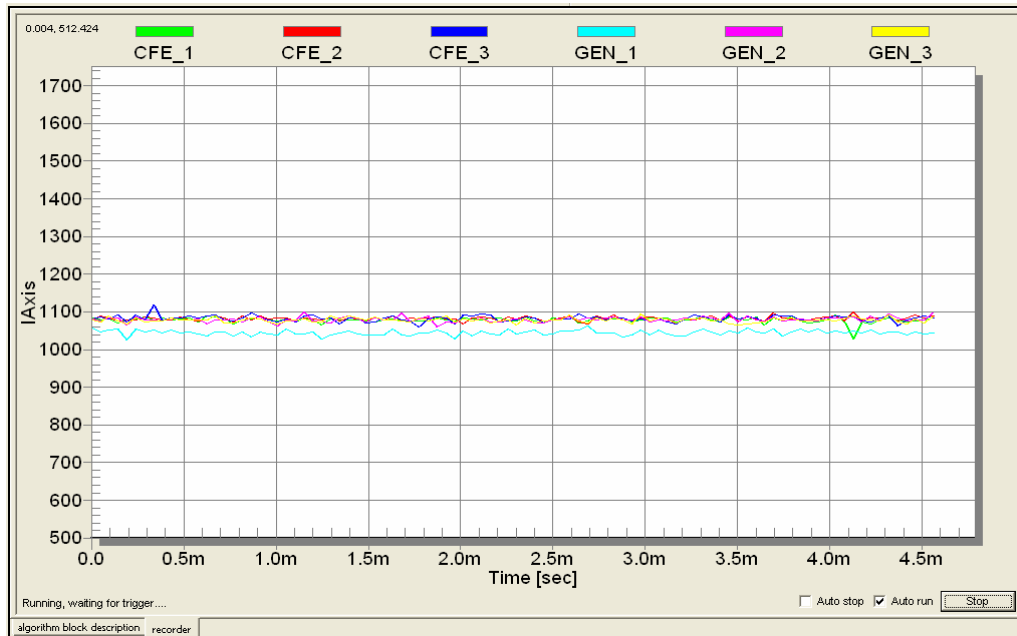


Figura 5.1.- Señales presentes del sincronoscopio en el estado “FUERA / INICIO”.

Ambiente de depuración en el estado “FUERA / INICIO”; ver Fig. 5.2.

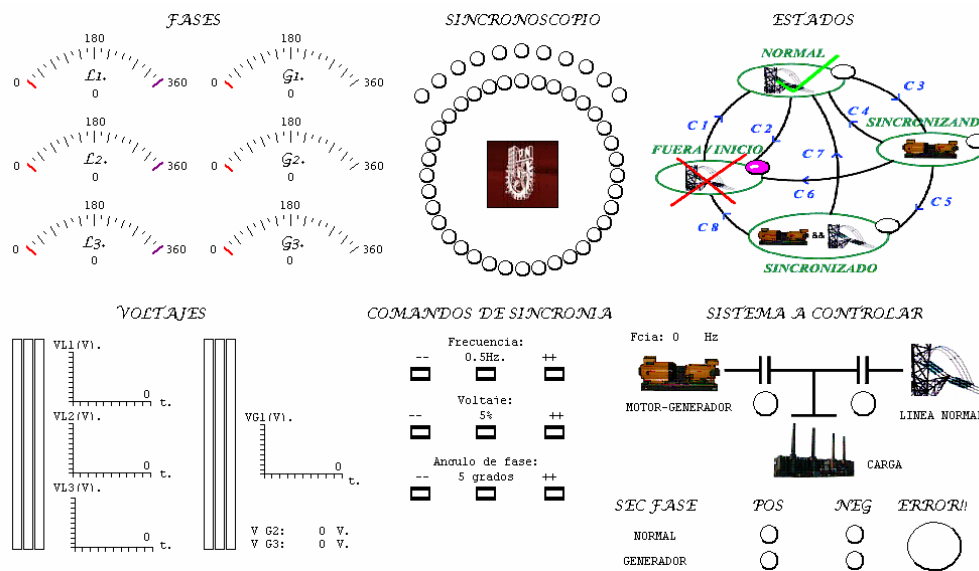


Figura 5.2.- Ambiente de depuración en el estado “FUERA / INICIO”

Cuando CFE es habilitada con una secuencia de fase positiva y el sistema motor-generator permanece desactivado; el oscilograma de las señales presentes en el sincronoscopio son las mostradas en la Fig. 5.3.

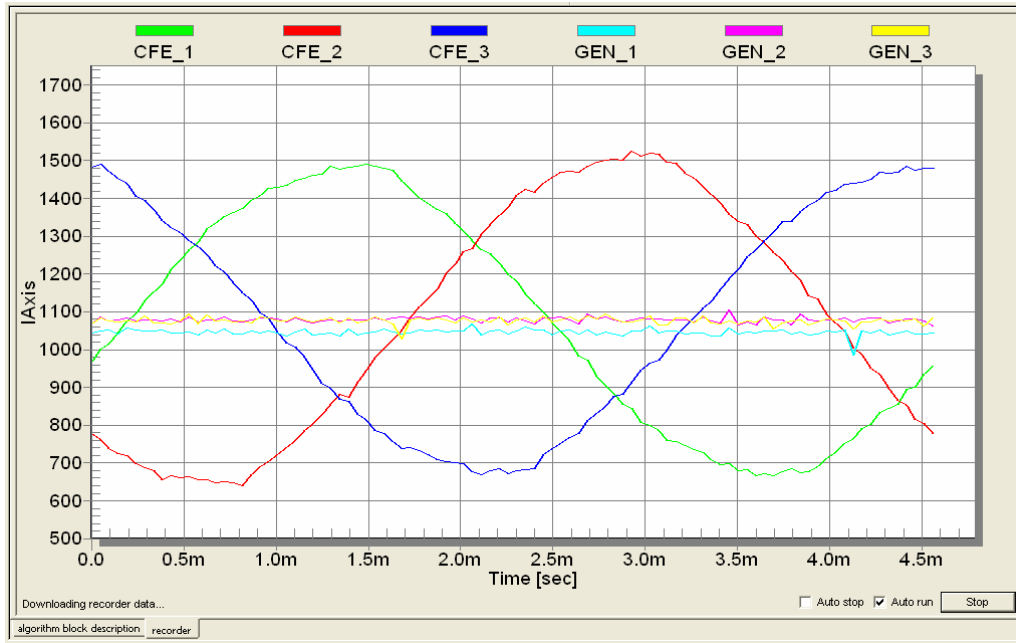


Figura 5.3.- Señales presentes del sincronoscopio en el estado “NORMAL”.

Esto activará la condición 1 (C1) y hará que el sincronoscopio pase al estado “NORMAL”. Ver en Fig. 5.4, el ambiente de depuración en el estado “NORMAL”.

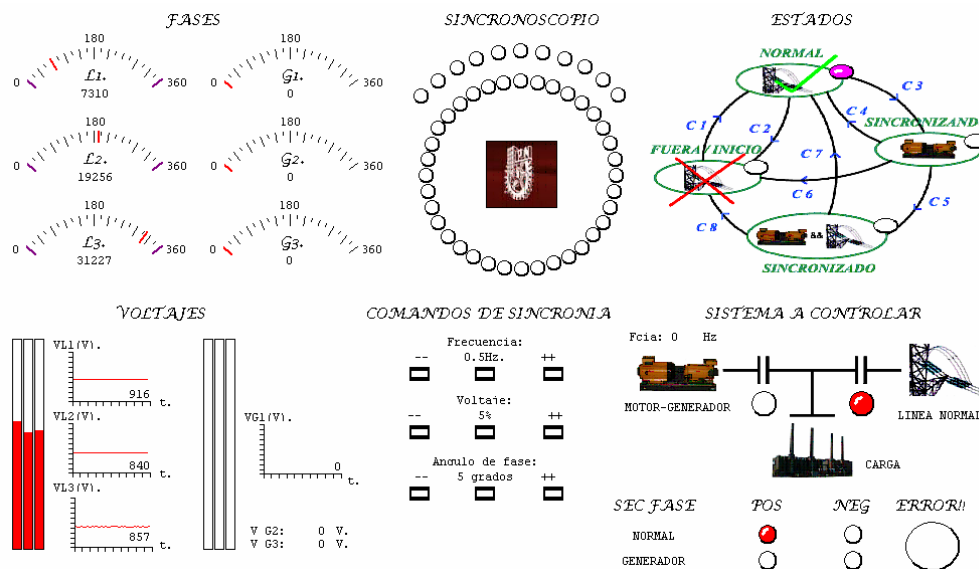


Figura 5.4.- Condición 1 (C1) activada; ambiente de depuración en el estado “NORMAL”

El sincronoscopio al igual que en el ambiente de depuración en la parte inferior derecha de la figura 5.4, los LED's indicadores coinciden, mostrando que el contactor correspondiente esta activado, ver figura 5.5.

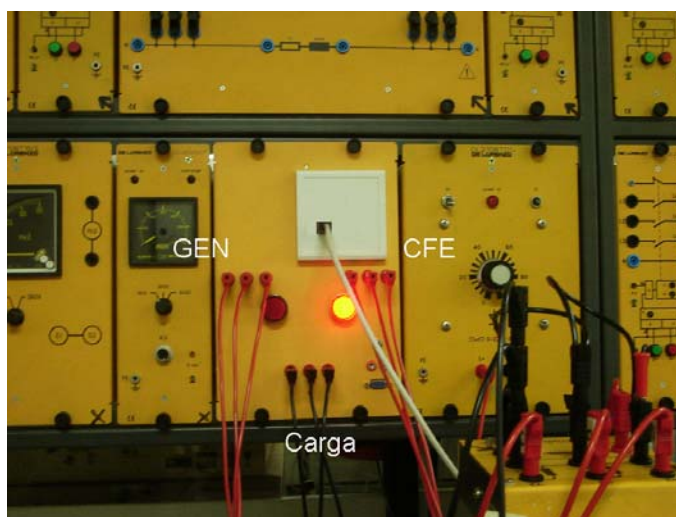


Figura 5.5.- Sincronoscopio en el estado "NORMAL".

Cuando el sistema motor-generador es activado, tiene una frecuencia y voltaje de salida mínimos de 30Hz y 40 VCA y una secuencia de fase positiva preestablecidos, ver oscilograma de la figura 5.6.

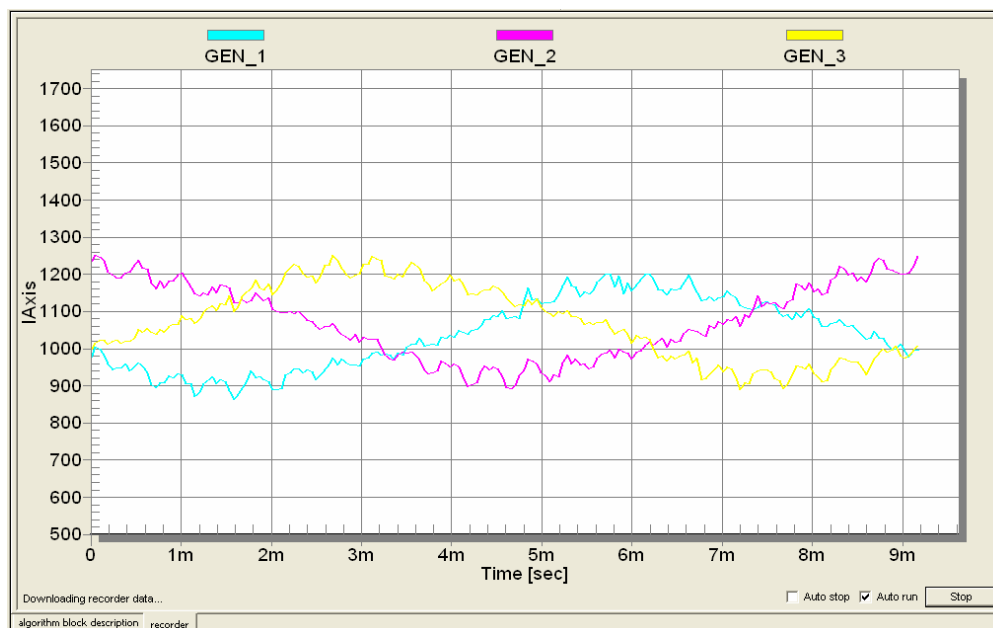


Figura 5.6.- Señal trifásica de salida del sistema motor-generador al ser activado (30Hz / 40VCA) con secuencia de fases positiva.

En el oscilograma de la figura 5.7 se muestra un comparativo de la señal trifásica de CFE y del sistema motor-generador en el instante de ser activado.

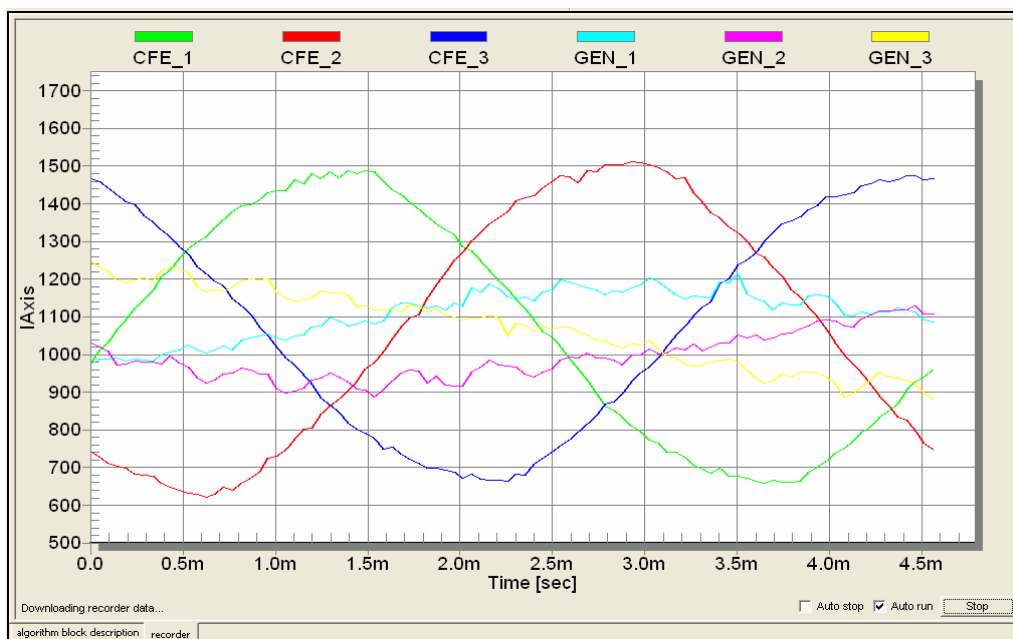


Figura 5.7.- Condición de “Inicio de sincronización” activada.

En el oscilograma de la figura 5.8 se muestran dos ciclos de la señal trifásica de CFE y un ciclo del sistema motor-generador por ser su frecuencia de inicio de 30 Hz.

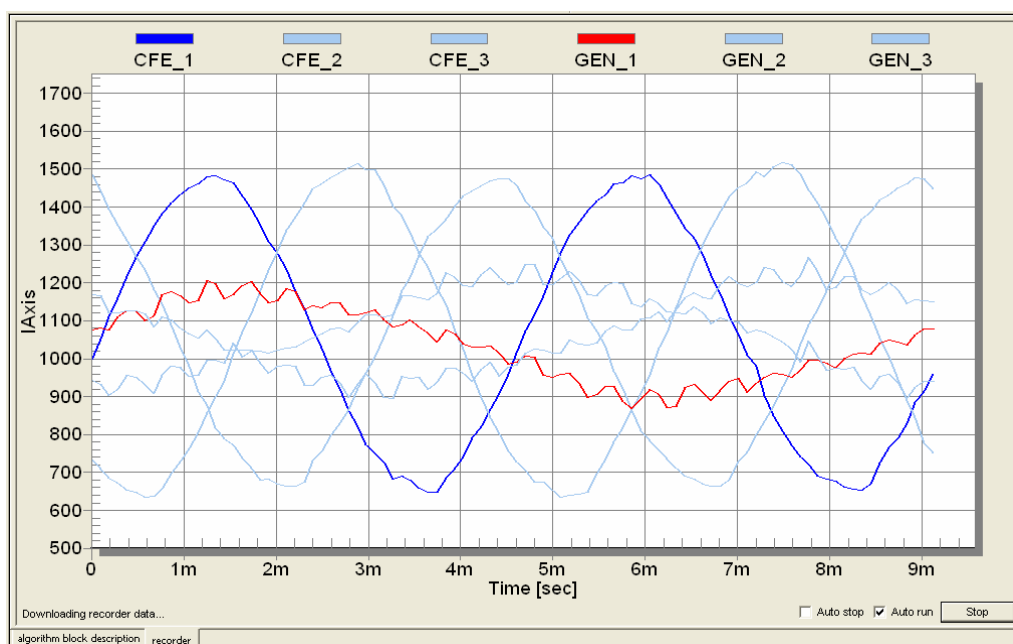


Figura 5.8.- Dos ciclo de CFE y uno del sistema motor-generador.

Estos niveles activan la condición de “Inicio de sincronización” (C3) que es monitoreada en el estado “NORMAL”, pasando así al sincroscope al estado “SINCRONIZANDO”; ver figura 5.9.

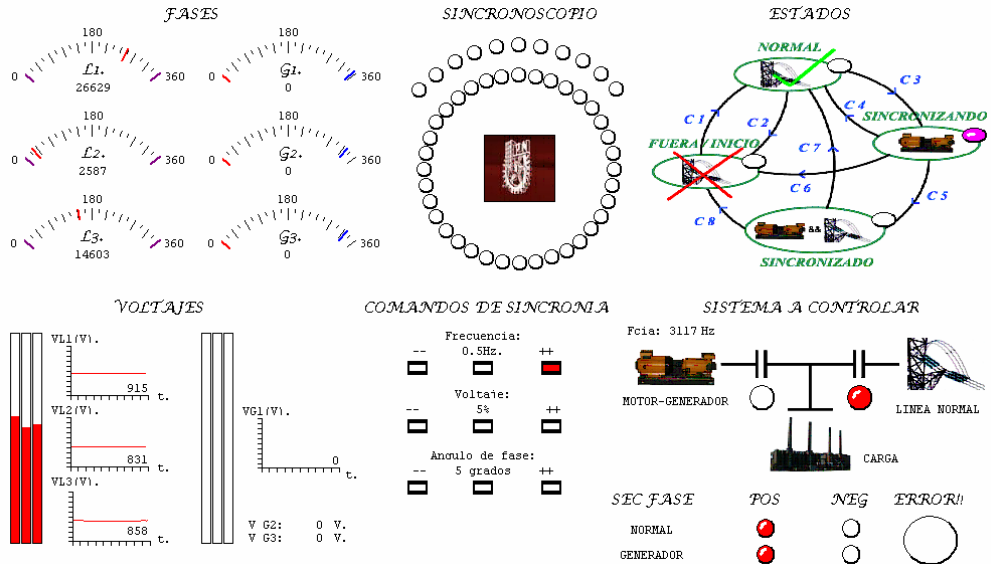


Figura 5.9.- Estado “SINCRONIZANDO” con comandos de ajuste de frecuencia.

En este instante, los comandos visuales de ajuste de frecuencia son activados para tener la frecuencia del sistema motor-generador a ±0.1 Hz de 60 Hz. En la figura 5.10 se muestra que el sistema motor-generador esta dentro de la norma.

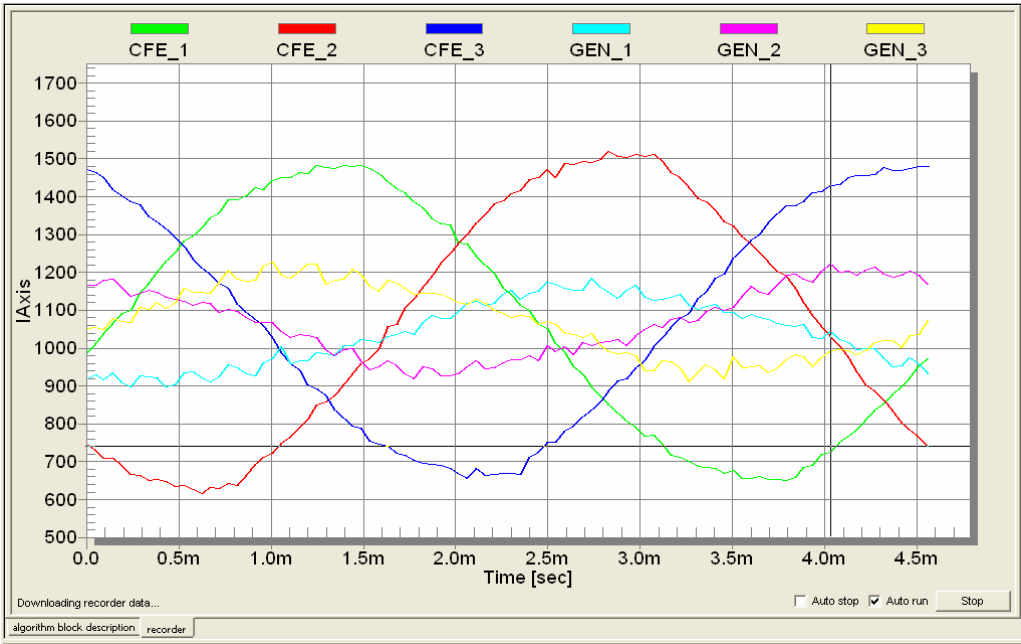


Figura 5.10.- Sistema motor-generador con frecuencia en norma.

En el oscilograma de la figura 5.11 se muestra la diferencia de frecuencia de una fase resaltada de CFE y otra resaltada del sistema motor-generador en norma.

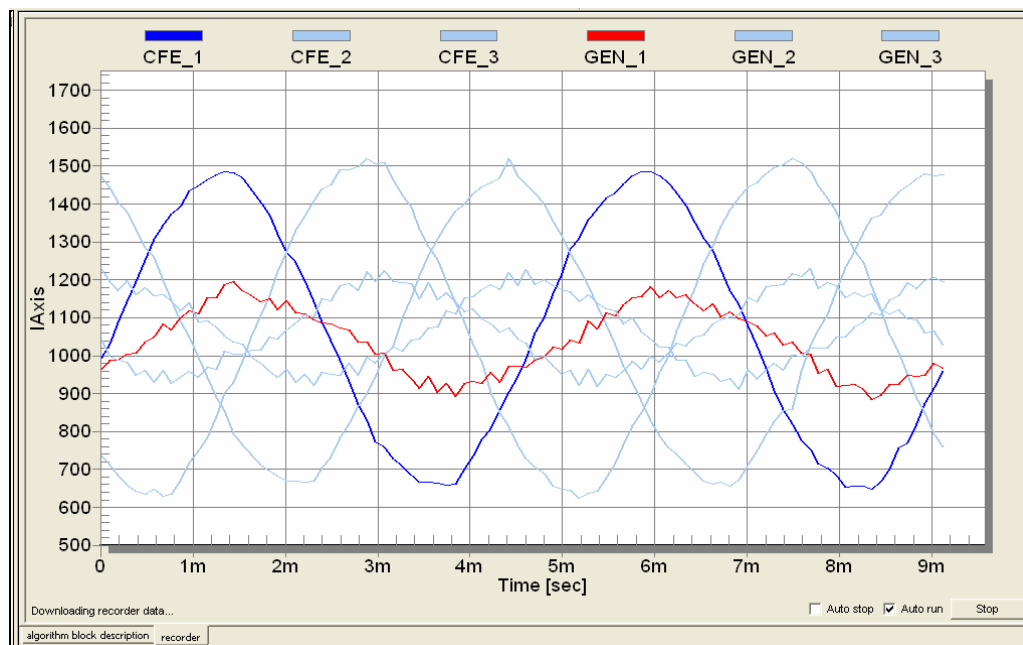


Figura 5.11.- Diferencia de frecuencia entre CFE y sistema motor-generador en norma.

Después de ajustada la frecuencia dentro de la norma, el sincronoscopio activa los comandos visuales de ajuste de voltaje para llevar el voltaje terminal del sistema motor-generador a $\pm 10\%$ del voltaje del Bus infinito, ver figura 5.12.

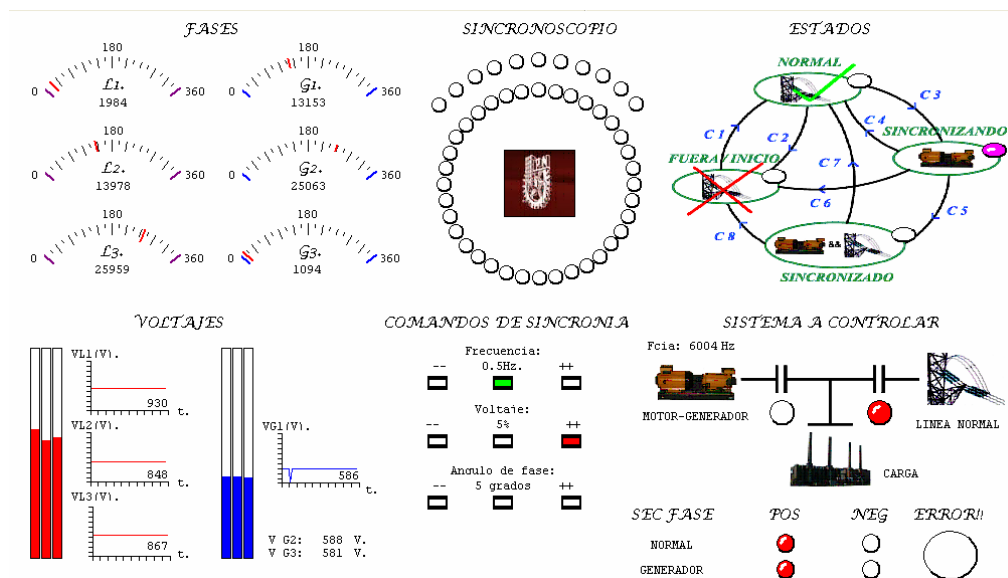


Figura 5.12.- Estado "SINCRONIZANDO" con comandos de ajuste de voltaje.

En la figura 5.13 se visualiza la señal de salida trifásica del sistema motor-generador con frecuencia y voltaje dentro de norma.

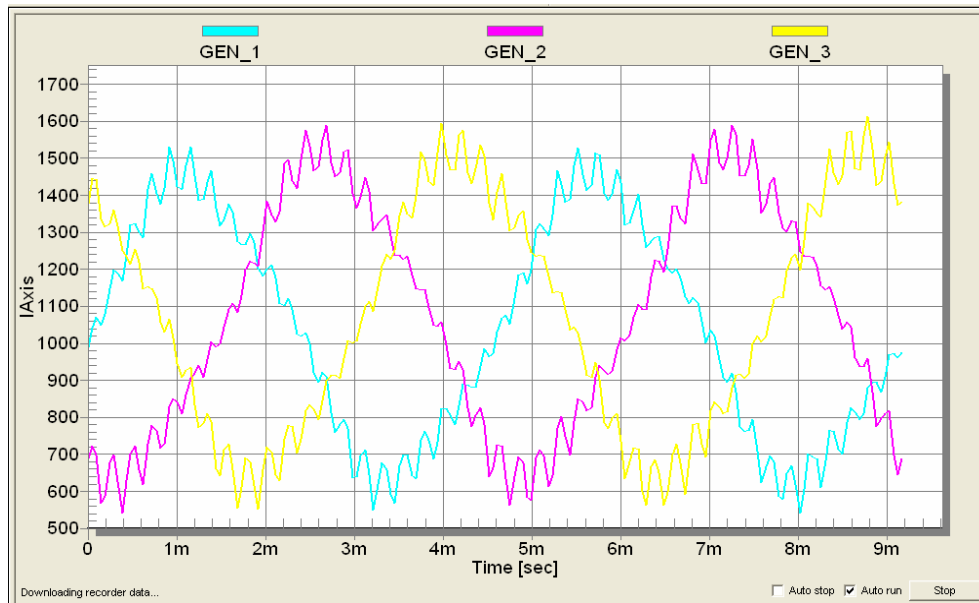


Figura 5.13.- Frecuencia y voltaje del sistema motor-generador dentro de norma.

Hasta aquí, existen dos casos: cuando la señal de salida trifásica del sistema motor-generador se encuentra adelantada o atrasada con respecto a la del Bus infinito; en la figura 5.14 se muestra cuando se encuentra adelantado.

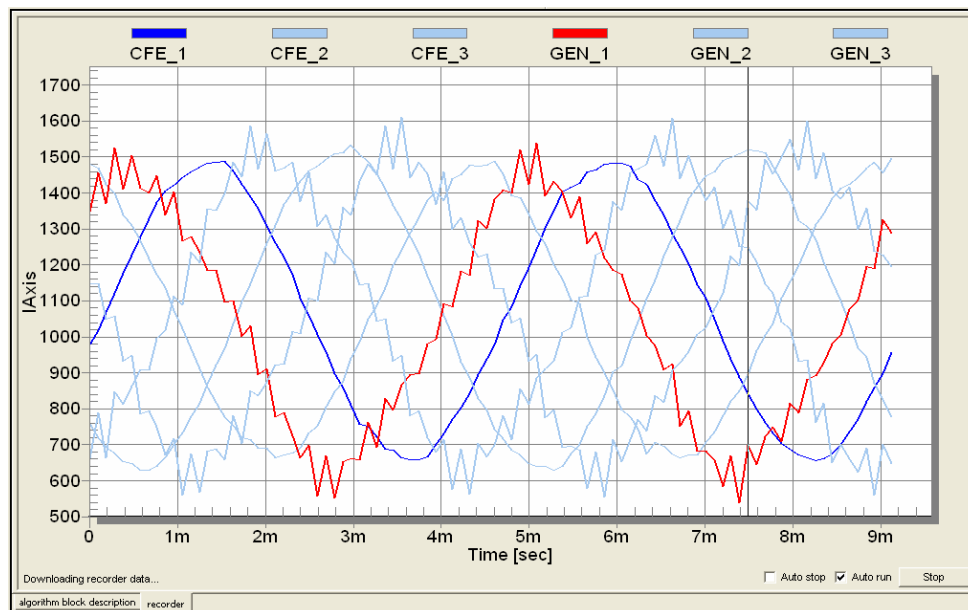


Figura 5.14.- Señal trifásica del sistema motor-generador adelantado.

Para el caso en el que la señal de salida trifásica del sistema motor-generator se encuentra adelantada con respecto a la del Bus infinito, los comandos visuales de ajuste de corrección del ángulo de fase en el ambiente de depuración son activados, ver figura 5.15.

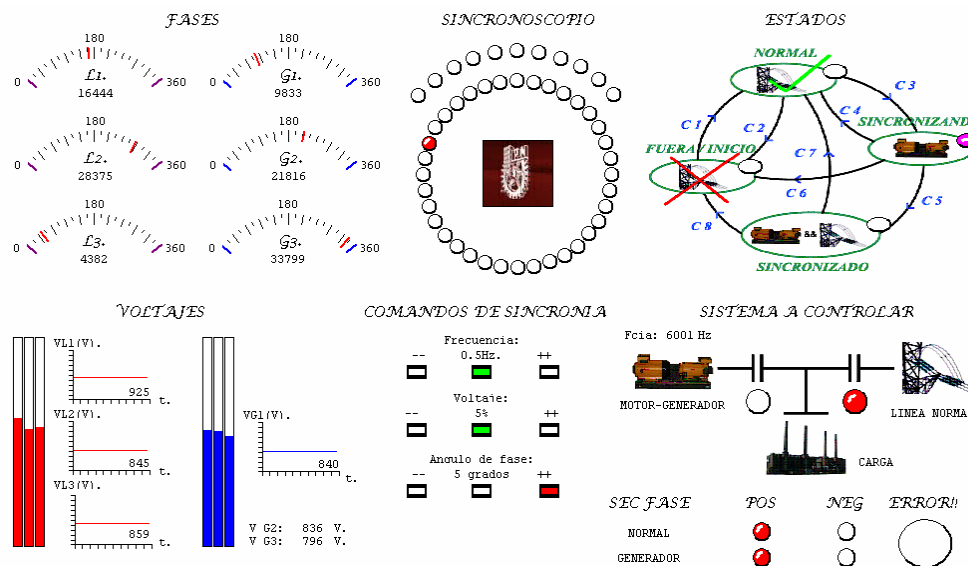


Figura 5.15.- Comando visual de ajuste de ángulo de fase adelantado.

En la figura 5.16 se muestra el caso en que la señal de salida trifásica del sistema motor-generator se encuentra atrasada.

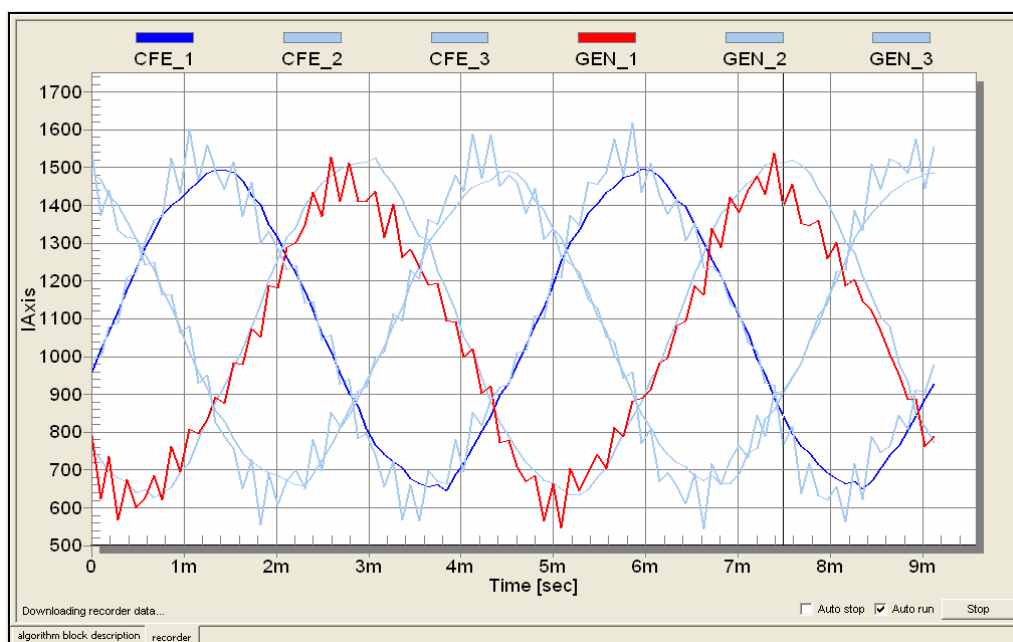


Figura 5.16.- Señal trifásica del sistema motor-generator atrasado.

Para el caso en el que la señal de salida trifásica del sistema motor-generator se encuentra atrasada con respecto a la del Bus infinito, los comandos visuales de ajuste de corrección del ángulo de fase son activados; ver figura 5.17.

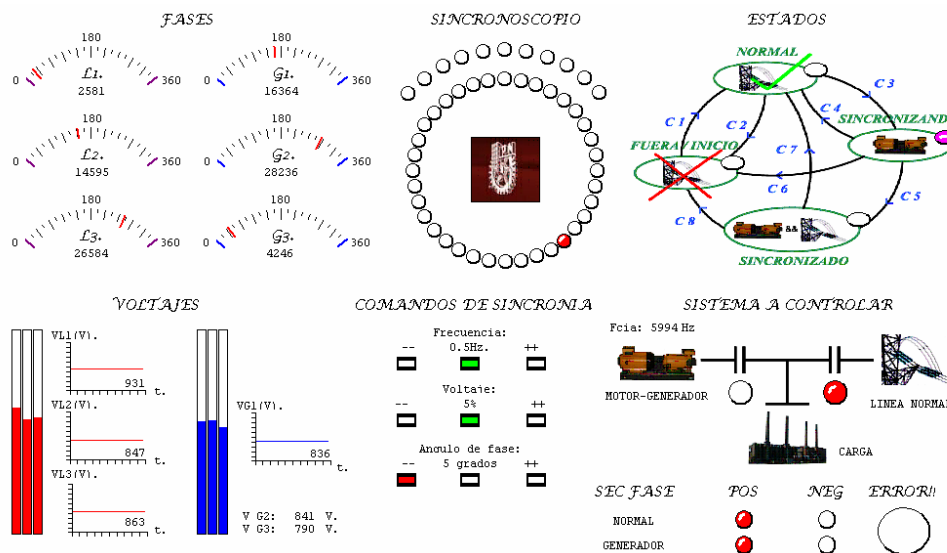


Figura 5.17.- Comando visual de ajuste de ángulo de fase atrasado.

Cuando el sincronoscopio detecta que las diferencias de voltaje, frecuencia y ángulo de fase se encuentran dentro de la norma, la condición 5 (C5) es activada y el Sincronoscopio activa el contactor del sistema motor-generator, para juntar sus fases con las del Bus infinito; ver figura 5.18.

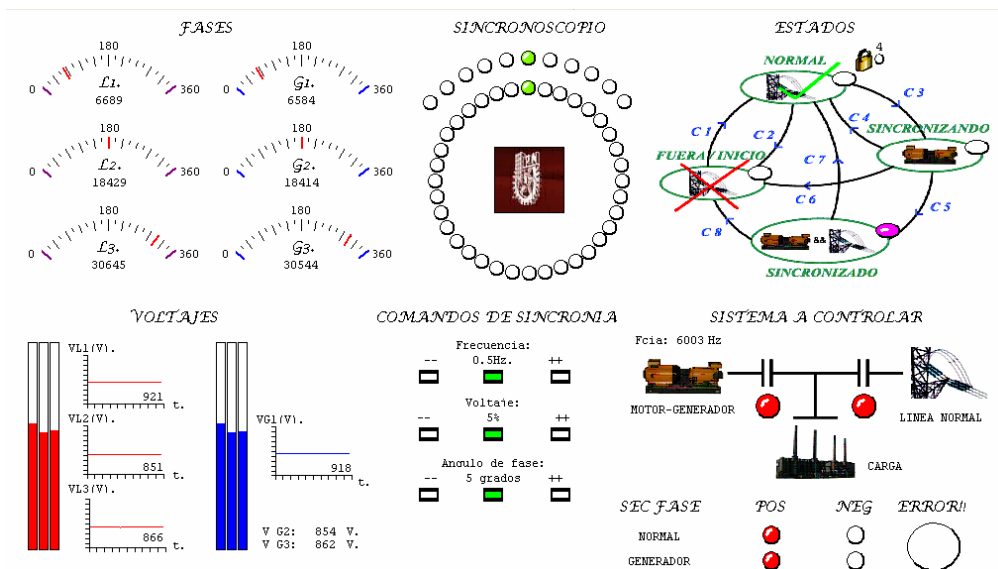


Figura 5.18.- Ambiente de depuración en el estado “SINCRONIZADO” con secuencia de fases positivas.

En la figura 5.19, se muestran las fases del Bus infinito y del sistema motor-generator con secuencia de fases positivas, en el instante en que se encuentran sincronizadas.

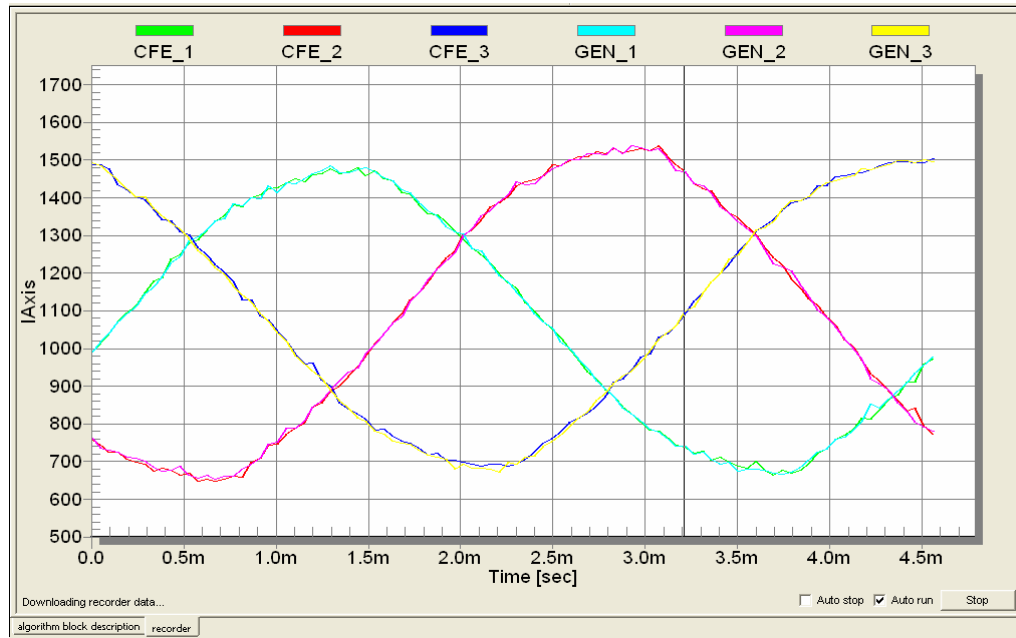


Figura 5.19.- Fases del Bus infinito y sistema motor-generator sincronizados con secuencias de fase positivas.

El sincronoscopio al igual que en el ambiente de depuración en la parte inferior derecha de la figura 5.18, los LED's indicadores coinciden, mostrando que los contactores están activados; ver figura 5.20.

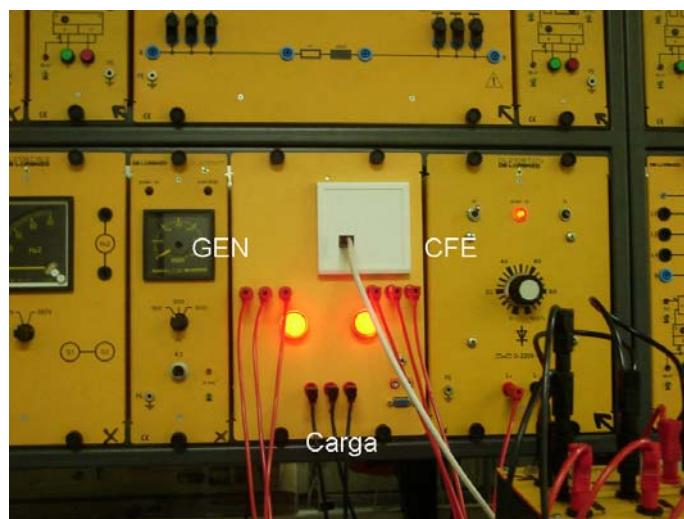


Figura 5.20.- Sincronoscopio en el estado "SINCRONIZADO".

Estando en el estado de “SINCRONIZADO”, se activa un “Timer” que determina el tiempo en que el sistema motor-generator permanecerá sincronizado al Bus infinito; finalizado dicho tiempo (condición 7), el sistema motor-generator será sacado de sincronía abriendo su contactor y los comandos visuales de ajuste de frecuencia y voltaje serán activados y ajustados para apagarlo; ver figura 5.21.

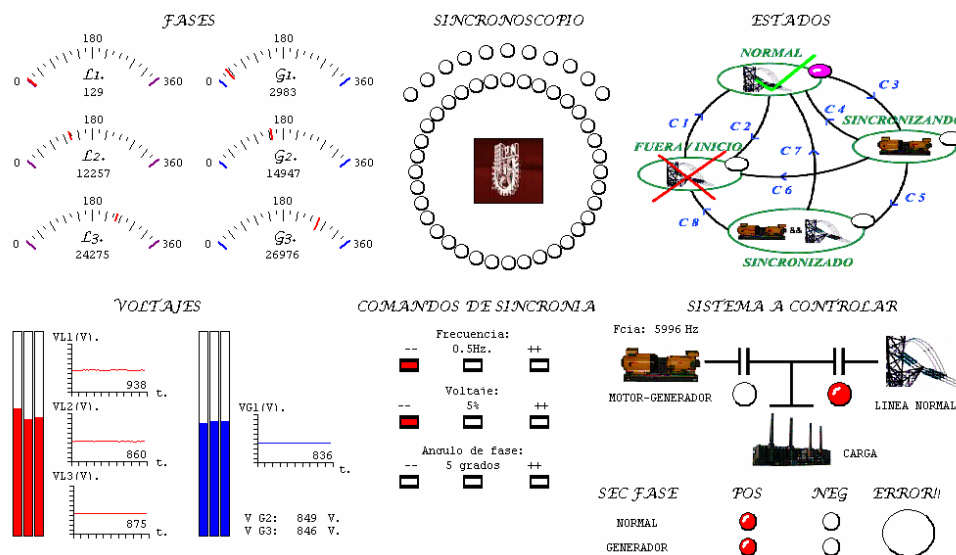


Figura 5.21.- Comando visuales de ajuste activados, para desactivar el sistema motor-generator.

Finalmente el Sincronoscopio permanecerá en el estado “NORMAL”, monitoreando otra condición de “Inicio de sincronización” (C3) o bien la desactivación de CFE (C2); ver figura 5.22.

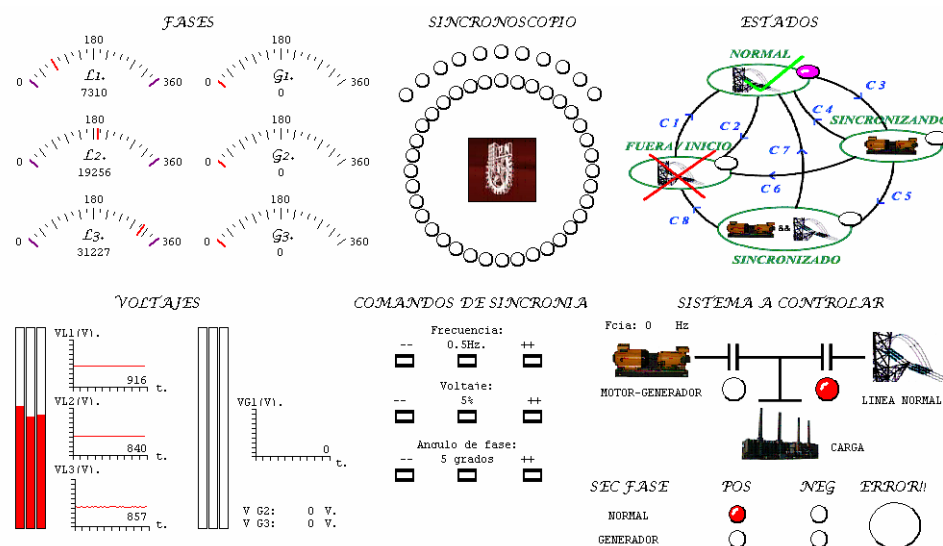


Figura 5.22.- Retorno al estado “NORMAL” por activación de la condición 7 (C7).

5.1.2 Sincronización con secuencias de fases negativas.

Partiendo del estado “FUERA / INICIO” definido en la figura 5.2, CFE es habilitada con una secuencia de fase negativa; ver el oscilograma de la figura 5.23.

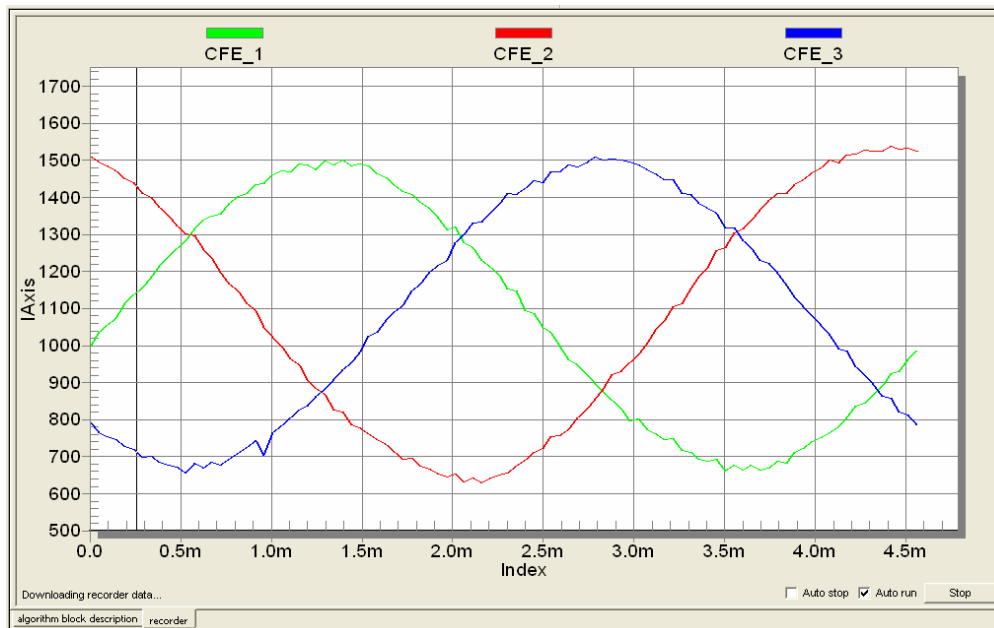


Figura 5.23.- Señal trifásica de CFE con secuencia de fase negativa.

Esto activará la condición 1 (C1) y hará que el sincronoscopio pase al estado “NORMAL”. Ver en Fig. 5.24, el ambiente de depuración en el estado “NORMAL”.

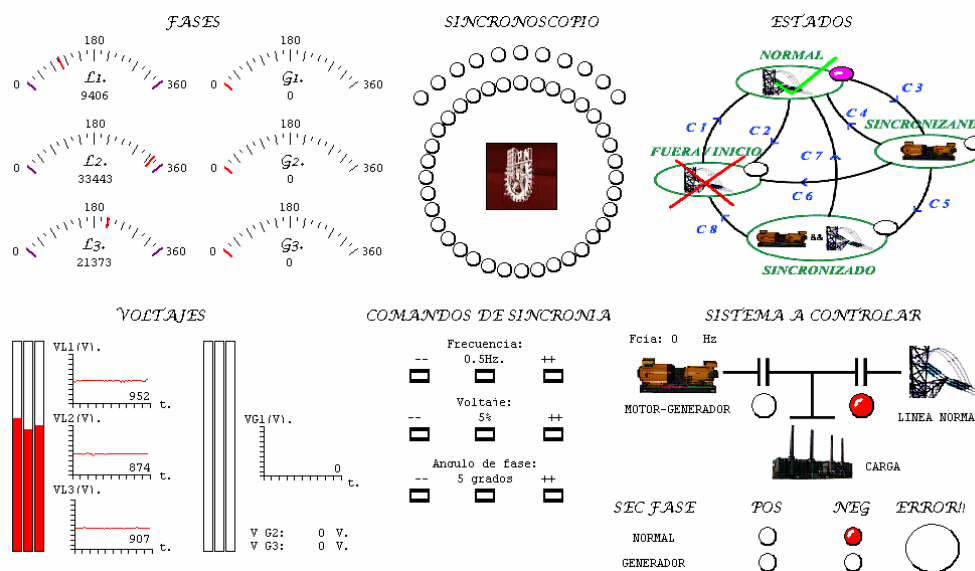


Figura 5.24.- Condición 1 (C1) activada; ambiente de depuración en el estado “NORMAL”

Cuando el sistema motor-generador es activado, tiene una frecuencia y voltaje de salida mínimos de 30Hz y 40 VCA y una secuencia de fase negativa preestablecidos; ver oscilograma de la figura 5.25.

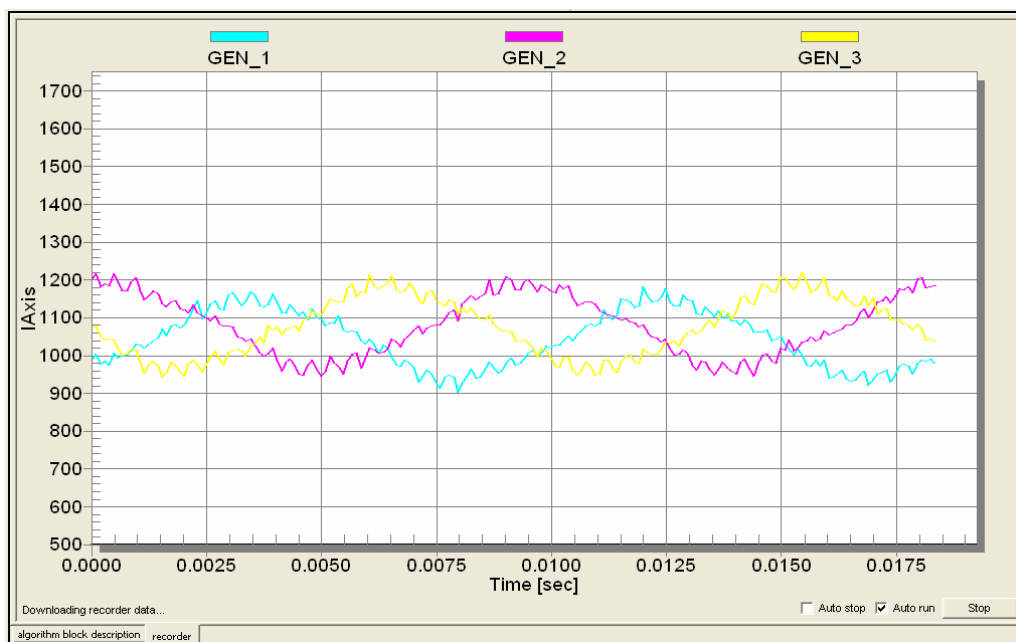


Figura 5.25.- Señal trifásica de salida del sistema motor-generador al ser activado (30Hz / 40VCA) con secuencia negativa.

Estos niveles activan la condición de “Inicio de sincronización” (C3), pasando así al sincronoscopio al estado “SINCRONIZANDO”; ver figura 5.26.

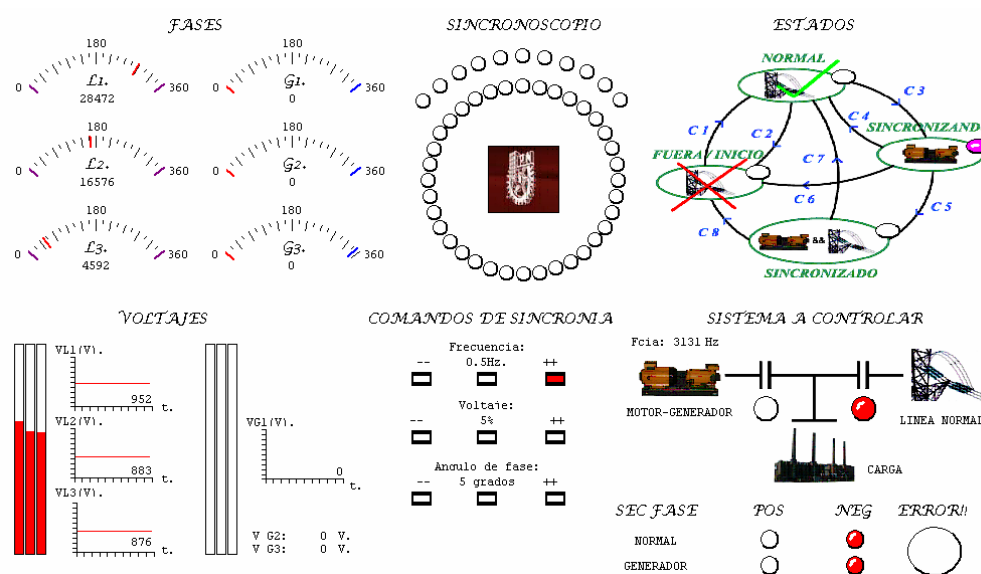


Figura 5.26.- Estado “SINCRONIZANDO” con comandos de ajuste de frecuencia.

Y siguiendo la secuencia de ajuste de voltaje, frecuencia y ángulos de fase, el Bus infinito y el sistema motor-generador son sincronizados por el Sincronoscopio; ver figura 5.27.

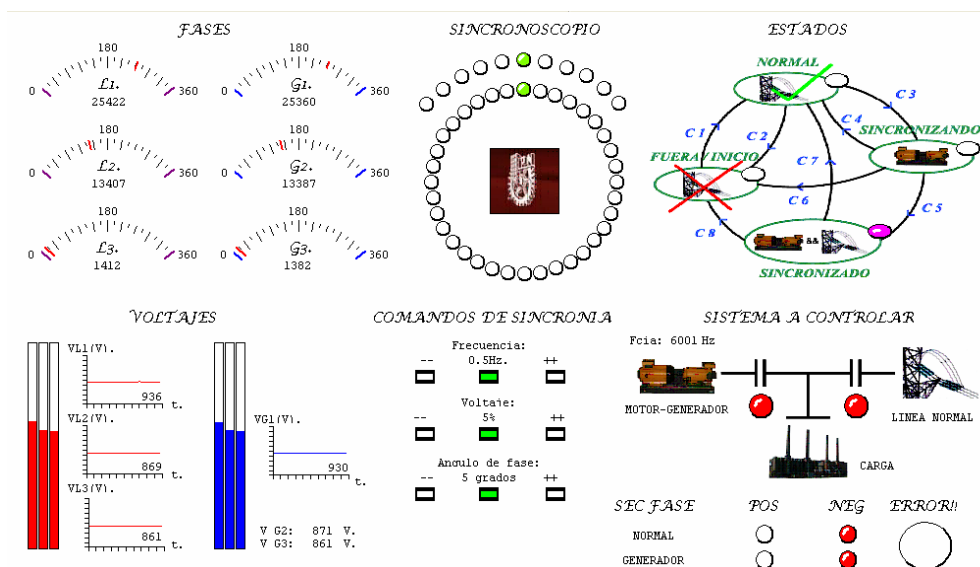


Figura 5.27.- Ambiente de depuración en el estado “SINCRONIZADO” con secuencia de fases negativas.

En la figura 5.28, se muestran las fases del Bus infinito y del sistema motor-generador con secuencias de fases negativas en el instante en que se encuentran sincronizados.

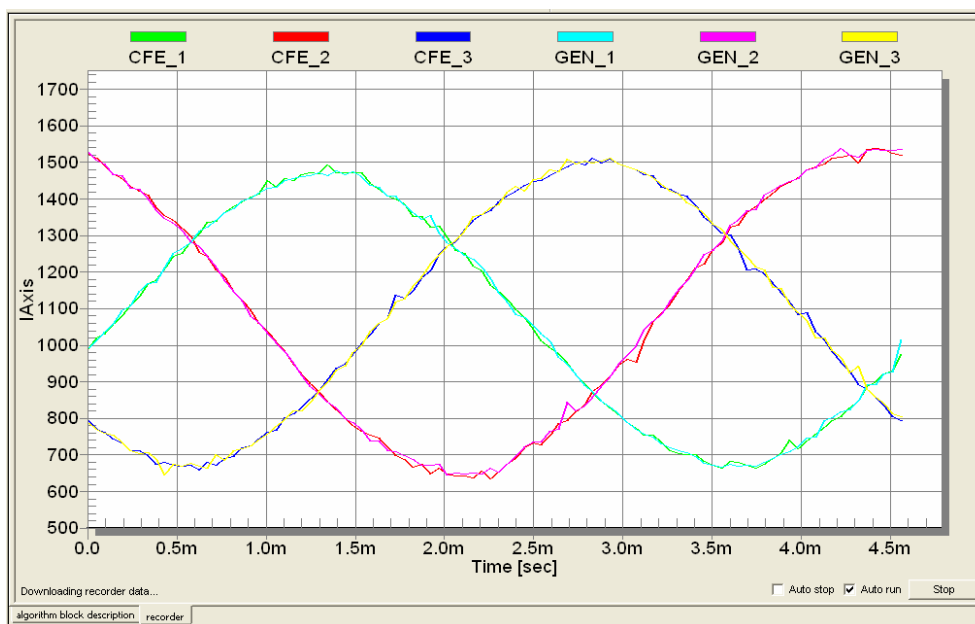


Figura 5.28.- Fases del Bus infinito y sistema motor-generador sincronizados con secuencias de fase negativas.

5.1.3 Respuesta ante secuencia de fases opuestas.

Para esta secuencia de prueba, se propone que CFE sea activada con una secuencia de fase positiva y el sistema motor-generador con una negativa; ver oscilogramas de figuras 5.29 y 5.30 respectivamente.

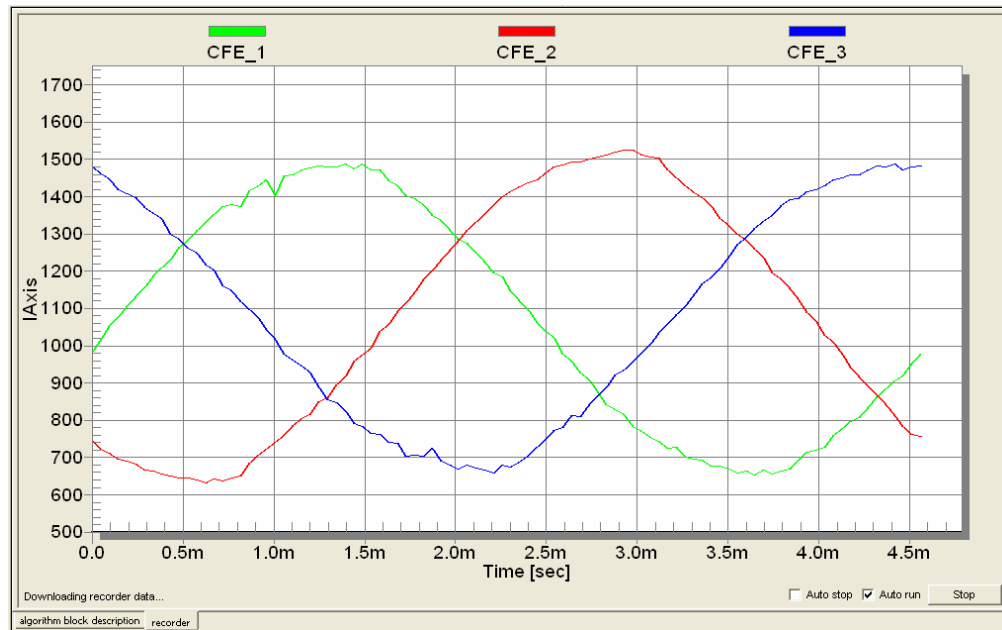


Figura 5.29.- CFE activada con secuencia de fase positiva.

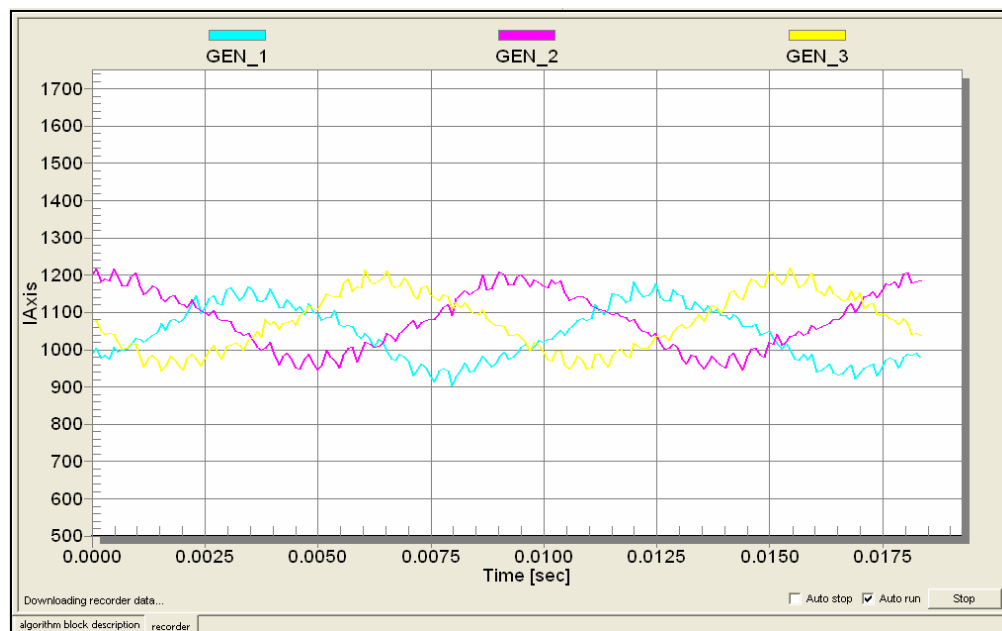


Figura 5.30.- Sistema motor-generador con secuencia de fase negativa.

Para esta prueba, el ambiente de depuración indica que existe un error e indica las secuencias de fases respectivas a cada sistema; ver figura 5.31.

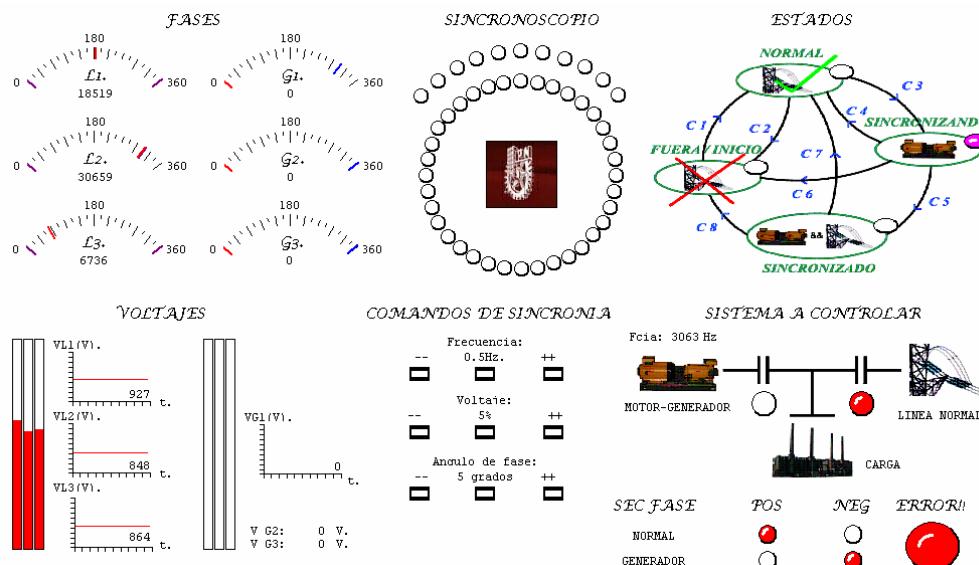


Figura 5.31.- Ambiente de depuración en el estado “SINCRONIZANDO”, indicando un error en las secuencias de fases de los sistema a sincronizar.

El Sincronoscopio no pasara del estado “SINCRONIZANDO” hasta que el error por secuencia de fase incorrecta sea corregido. Esto se corregirá invirtiendo cualquier par de fases del Bus infinito o del sistema motor-generador.

Si se quisiera efectuar una sincronización con secuencia de fases negativas, se debe de invertir cualquier par de fases de CFE o si se desea hacer una sincronización con secuencia de fases positivas, se debe de invertir cualquier par de fases del sistema motor-generador.

5.1.4 Sacar de sincronía al sistema motor-generador con comandos generados en el IDE.

Una opción mas de monitoreo y supervisión del algoritmo diseñado, es generando comandos que modifican las variables almacenadas en memoria del microcontrolador mientras este se encuentra ejecutando una aplicación sin la necesidad de agregar

puntos de ruptura. De la barra de menú, se da clic en “Component”, se escoge el componente denominado “Command”, ver figura 5.32.

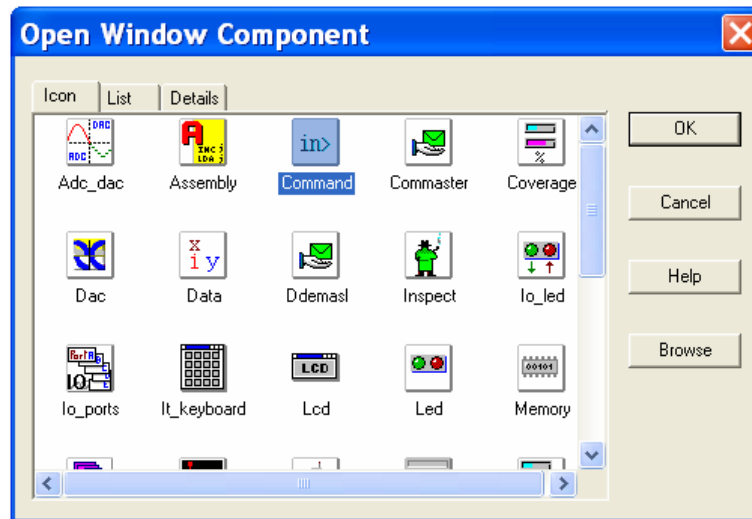


Figura 5.32.- Selección del componente “Command” en el ambiente de depuración.

La ventana de la figura 5.33, muestra el área en la cual se escriben los comandos y envían al microcontrolador mientras este se encuentra ejecutando el algoritmo, sin la necesidad de puntos de ruptura.



Figura 5.33.- Componente “Command” en el ambiente de depuración.

La condición 7 (C7), se adapto para monitorear una variable manipulable específicamente por la línea de comandos y así sacar al sistema motor-generator del estado de “SINCRONIZADO”. La prueba se realizo y comprobó que el sincronoscopio paso al estado de “NORMAL”.

CAPÍTULO 6

CONCLUSIONES Y RECOMENDACIONES PARA TRABAJOS FUTUROS

6.1 Conclusiones

Se llevó a cabo la implementación del sincronoscopio, concluyendo que:

- El sincronoscopio operó sin fallas en todas las pruebas realizadas, con condiciones de sincronía y de no sincronía.
- El prototipo se comunica en tiempo real con una PC mediante un puerto serial RS232 transmitiendo todas las mediciones que ejecuta. Esto fue muy útil porque permitió utilizar esta información para realizar ajustes a los algoritmos de medición.
- Mediante la herramienta de instrumentación virtual se pudo visualizar todo el proceso de sincronización en tiempo real, incluyendo variables como el voltaje, la frecuencia y la diferencia de ángulo.

6.2 Aportaciones de la Tesis.

- Proporcionar al laboratorio de Sistemas Digitales de la SEPI-ESIME-ZAC un equipo que realice las etapas de sincronización para poder realizar futuras investigaciones.
- Proporcionar diagramas de flujo, plantillas y algoritmos en lenguaje C que puedan ser modificados y/o adecuados a una aplicación específica de:

-
- Procesamiento de señales digitales en tiempo real, empleados para el monitoreo y visualización de parámetros de operación de la compañía suministradora de energía eléctrica y del sistema motor-generador.
 - Sincronización del sistema motor-generador a la compañía suministradora de energía eléctrica.

6.3 Recomendaciones para trabajos futuros

- Medir la potencia activa y reactiva y controlar el regulador de voltaje y gobernador de velocidad para realizar la sincronía en forma automática.
- Mostrar los datos de forma local, mediante una pantalla de cristal líquido.
- Monitorear la potencia suministrada por el sistema motor-generador a la compañía suministradora de energía eléctrica.
- Agregar la medición del ángulo de carga del generador.
- Programar mediante visual Basic una interfaz hombre maquina para poder interactuar con el sincronoscopio, sin depender de la herramienta de depuración.

REFERENCIAS

- [1] Bose, B. K. *“Modern Power Electronics and AC Drives”*. USA: PEARSON Prentice Hall, INC. 2002.
- [2] Mohan, N; Undeland, T. M; Robbins, W. P. *“Power Electronics, Converters, Applications and Design”*. USA: John Wiley & Sons, INC, Third Edition. 2003.
- [3] Krause, P.C. *“Analysis of Electric Machinery and Drive Systems”*. USA: John Wiley & Sons, INC, Second Edition, 2002.
- [4] Fitzgerald, A.E; Kingsley, C.Jr; Umas, S.D. *“Máquinas Eléctricas”*. México, Mc Graw Hill. Sexta Edición, 2004.
- [5] Chapman J. S; *“Máquinas Eléctricas”*. Mc Graw Hill, Tercera Edición, 2003.
- [6] Yang Yi-han, Shang Guo-cai y Fang Yong-jie A Fast Following Synchronizer og Generators. IEEE Transactions on Energy Conversion, Vol. 3, No. 4, December 1988.
- [7] LEY DEL SERVICIO PÚBLICO DE ENERGIA ELECTRICA. Diario Oficial de la Federación 31 de Mayo de 1993, Artículo 72.
- [8] Theodore Wildi *“Máquinas Eléctricas y sistemas de potencia.”* PEARSON Prentice Hall, Sexta Edición 2007.
- [9] DEEP SEA ELECTRONICS PLC
Empresa de diseño, fabricación y proveedor de equipo de control, vigilancia y protección para equipos generadores a base de gasóleo y gas.
Email: sales@deepseausa.com, support@deepseapl.com Website: www.deepseausa.com
3230 Williams Avenue Rockford IL61101-2668 USA Tel: +1 (815) 316 8706
- [10] OTTOMOTORES, S.A. de C.V.
Empresa en el sector eléctrico, productos de transmisión y generación de energía.
E-mail: joseluis.hernandez@ottomotores.com.mx, servicios@ottomotores.com.mx
Website: www.ottomotores.com
Sales Dept. San Lorenzo # 1150. Col. Cerro de la Estrella, Iztapalapa. México, D.F.
C.P. 09860. Tel: 52-55-5624-5628 Fax: 52-55-5426-5581.
- [11] EGSA 100E 1992; Performance Standard for Governors on Engine Generator sets. Electrical Generating Systems Associations.

-
- [12] EGSA 100R 1992a; Performance Standard for voltage regulators used on electric generators. Electrical Generating Systems Associations.
- [13] Luis Casado de Freescale Semiconductor. Escalabilidad, conectividad, compatibilidad en μ C. Nueva familia de microcontroladores de 8 y 32 Bits FLEXIS de Freescale
- [14] M. en C. Carlos Cuvas Castillo, Tesis de maestría titulada: "IMPLEMENTACIÓN DE UN MEDIDOR FASORIAL" 2006.
- [15] Benmouyal, Gabriel; Schweitzer, E. O.; Guzmán, A. *Synchronized phasor measurement in protective relays for protection, control, and analysis of electric power system*. 29th annual western protective relay conference spokane, washington october 22-24, 2002.
- [16] Oppenheim, Alan V.; Willsky, Alan V. *Sistemas y Señales*. Prentice Hall. México.
- [17] Stanley, William D. *Digital Signal Processing*. Rader and Gold
- [18] Smith, Steven W. *The Scientist and Engineer's Guide to Digital Signal Processing*, Second Edition
- [19] Oppenheim, Alan V. *Discrete-Time Signal Processing*. Prentice-Hall, Inc. New Jersey 1999.
- [20] Spencer, A. J; Parker, D.F. *Matemáticas para ingeniería, Volumen 1*. CECSA.
- [21] Kreyszig, Edwin. *Matemáticas avanzadas para ingeniería, Vol II*. Limusa.
- [22] Kaplan, Wilfred. *Cálculo Avanzado*. CECSA.
- [23] Raj Mehra; Hsu Hwei P. *Análisis de Fourier*. Addison-Wesley-Iberoamericana.
- [24] Doebelin Ernest O. *Diseño y aplicación de sistemas de medición*. Diana México.
- [25] Cooper, William D.; Helfrick, Albert D. *Instrumentación electrónica moderna y técnicas de medición*. Prentice Hall.
- [26] Seippel, Robert G. *Transducer interfacing signal conditioning for process control*. Prentice Hall.

- [27] John G. Proakis, Dimitris G. Manolakis *Tratamiento digital de señales. Principio, algoritmos y aplicaciones*. 3ª ed. Prentice Hall.

APENDICES

A. PROGRAMAS

A.1 Programas de Code Warrior.

A.1.1 Principal (Main)

```

/** *****
**      Filename   : DFT_3FASES.C
**      Project    : DFT_3FASES
**      Processor  : MCF51QE128LQFP64
**      Version    : Driver 01.00
**      Compiler   : CodeWarrior ColdFireV1 C Compiler
**      Date/Time  : 19/09/2007, 07:41 p.m.
**      Abstract   :
**      Main module.
**      Here is to be placed user's code.
**      Settings   :
**      Contents   :
**      No public methods
**
**      (c) Copyright UNIS, spol. s r.o. 1997-2006
**      UNIS, spol. s r.o.
**      Jundrovska 33
**      624 00 Brno
**      Czech Republic
**      http       : www.processorexpert.com
**      mail        : info@processorexpert.com
** *****/
/* MODULE DFT_3FASES */

/* Including used modules for compiling procedure */
#include "Cpu.h"
#include "Events.h"
#include "Ttl.h"
#include "Bit1.h"
#include "AD1.h"
#include "Bit2.h"
#include "AS1.h"
#include "Bit3.h"
#include "REELE_GEN.h"
#include "REELE_CFE.h"
#include "Parp_EDOs.h"
/* Include shared modules, which are used for whole project */
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"
#include "stdio.h"
#include "secuencia_fase.h"

int edo_synch=EDO_FUERA_INICIO;

void main(void)
{
    /* Write your local variable definition here */

    /*** Processor Expert internal initialization. DON'T REMOVE THIS CODE!!! ***/
    PE_low_level_init();
    /*** End of Processor Expert internal initialization.          ***/
}

```

```

/* Write your code here */
/* For example: for(;;) { } */

for(;;)
{
    if(Band_calc_DFT_parte_final)
    {
        Calc_DFT_parte_final();
        switch(edo_synch)
        {
            case(EDO_FUERA_INICIO):
                Definicion_EDO_FUERA_INICIO();
                break;

            case(EDO_NORMAL):
                Definicion_EDO_NORMAL();
                break;

            case(EDO_SINCRONIZANDO):
                Definicion_EDO_SINCRONIZANDO();
                break;

            case(EDO_SINCRONIZADO):
                Definicion_EDO_SINCRONIZADO();
                break;

            default:
                break;
        }

        Actualiza_salidas();
        Actualiza_VTL();
        //depurador_Hyper();
        Bit3_NegVal();
    }
}

/** Don't write any code pass this line, or it will be deleted during code
/** Processor Expert end of main routine. DON'T MODIFY THIS CODE!!! ***/
for(;;){}
/** Processor Expert end of main routine. DON'T WRITE CODE BELOW!!! ***/
} /** End of main routine. DO NOT MODIFY THIS TEXT!!! ***/

/* END DFT_3FASES */
/*
** #####
**
** This file was created by UNIS Processor Expert 3.00 [03.89]
** for the Freescale ColdFireV1 series of microcontrollers.
**
** #####
**
*/

```

A.1.2 Estados.c

```

#include "Estados.h"
#include "Frecuencimetro.h"
#include "Comandos_sincronoscopio.h"
#include "LEDs_Sincronoscopio_VTL.h"
#include "secuencia_fase.h"
#include "LEDs_edos_VTL.h"
#include "DFT.h"
#include "stdio.h"
#include "PE_Types.h"
#include "Condiciones.h"

#define EDO_FUERA_INICIO    101
#define EDO_NORMAL         102
#define EDO_SINCRONIZANDO  103
#define EDO_SINCRONIZADO   104

extern    Timer_sincronizado_seg;

void Definicion_EDO_FUERA_INICIO(void)
{
    Band_CFE=0;
    Band_GEN=0;

    Band_GEN_desincronizado=0;

    Band_fcia_OK=0;
    Band_voltaje_OK=0;
    Band_ang_fase_OK=0;
    Timer_sincronizado_seg=0;

    Calc_Frecuencia_parte_final();

    Calc_sec_fases();
    Clr_comandos_synch();

    if(Monitoreo_C1()==0xABCD)
        edo_synch=EDO_NORMAL;
}

void Definicion_EDO_NORMAL(void)
{
    Band_CFE=1;
    Band_GEN=0;

    Calc_sec_fases();
    Clr_comandos_synch();

    if(Band_GEN_desincronizado)
        Calc_Frecuencia_parte_final();

    if(Monitoreo_C2()==0xABCD)
        edo_synch=EDO_FUERA_INICIO;

    if(Monitoreo_C3()==0xABCD)
        edo_synch=EDO_SINCRONIZANDO;
}

void Definicion_EDO_SINCRONIZANDO(void)
{
    Band_CFE=1;
    Band_GEN=0;

    Calc_sec_fases();
    Calc_Frecuencia_parte_final();
    Comandos_synch();
}

```

```
    if(Monitoreo_C4()==0xABCD)
        edo_synch=EDO_NORMAL;

    if(Monitoreo_C5()==0xABCD)
        edo_synch=EDO_SINCRONIZADO;

    if(Monitoreo_C6()==0xABCD)
        edo_synch=EDO_FUERA_INICIO;
}

void Definicion_EDO_SINCRONIZADO(void)
{
    Band_CFE=1;
    Band_GEN=1;

    Calc_Frecuencia_parte_final();

    if(Monitoreo_C7()==0xABCD)
        edo_synch=EDO_NORMAL;

    if(Monitoreo_C8()==0xABCD)
        edo_synch=EDO_FUERA_INICIO;
}
```

A.1.3 Condiciones.c

```

#include "Condiciones.h"
#include "Estados.h"
#include "Frecuencimetro.h"
#include "Comandos_sincronoscopio.h"
#include "LEDs_Sincronoscopio_VTL.h"
#include "secuencia_fase.h"
#include "LEDs_edos_VTL.h"
#include "DFT.h"
#include "stdio.h"
#include "PE_Types.h"

#define TIEMPO_SINCRONIZADO 40

int Band_CFE=0;
int Band_GEN=0;
int Band_GEN_desincronizado=0;
int cand=0;//sombra de "Band_GEN_desincronizado"
int sacar=0;//Comando para desincronizar al sistema motor-generator.

//int Cont_seg,Cont_aux;
extern int Timer_sincronizado_seg;

int Monitorea_C1(void)
{
    if(Mag_Fase[0][0]>800)
        return(0xABCD);
    else
        return(0);
}

int Monitorea_C2(void)
{
    if(Mag_Fase[0][0]<800)
        return(0xABCD);
    else
        return(0);
}

int Monitorea_C3(void)
{
    Band_GEN_desincronizado=cand;

    if(!Band_GEN_desincronizado)
    {
        if((Mag_Fase[3][0]>50)&&(Mag_Fase[4][0]>50)&&(Mag_Fase[5][0]>50))
            return(0xABCD);
        else
            return(0);
    }

    if((Mag_Fase[3][0]<50)&&(Mag_Fase[4][0]<50)&&(Mag_Fase[5][0]<50))
        Band_GEN_desincronizado=0;
}

int Monitorea_C4(void)
{
    if((Mag_Fase[3][0]<50)&&(Mag_Fase[4][0]<50)&&(Mag_Fase[5][0]<50))
        return(0xABCD);
    else
        return(0);
}

```

```

int Monitoreo_C5(void)
{
    static int Band_voltaje_OK_aux;
    static int Band_C5_aux;

    Band_voltaje_OK_aux=0;
    if(Band_sec_fases_OK)
    {
        if(Band_fcia_OK)//Nivel de frecuencia optimo +/- 0.5Hz.
        {
            if((Mag_Fase[3][0]>804)&&(Mag_Fase[3][0]<888))//Nivel de voltaje optimo +/- 5% .
            {
                Band_voltaje_OK_aux=1;

                if(Band_ang_fase_OK)//Nivel de angulo de fase optimo +/- 5°.
                {
                    Timer_sincronizado_seg=TIEMPO_SINCRONIZADO;
                    return(0xABCD);
                }
            }
        }
    }

    Band_voltaje_OK = Band_voltaje_OK_aux;
    return(0);
}

int Monitoreo_C6(void)
{
    if((Mag_Fase[0][0]<800)&&(Mag_Fase[1][0]<800)&&(Mag_Fase[2][0]<800))
        return(0xABCD);
    else
        return(0);
}

int Monitoreo_C7(void)
{
    if(!Timer_sincronizado_seg||sacar)
    {
        Band_GEN_desincronizado=1;
        cand=1;//sombra de "Band_GEN_desincronizado"
        sacar=0;
        return(0xABCD);
    }
    else
    {
        Band_GEN_desincronizado=0;
        cand=0;//sombra de "Band_GEN_desincronizado"
        return(0);
    }
}

int Monitoreo_C8(void)
{
    if(Mag_Fase[0][0]<800)
        return(0xABCD);
    else
        return(0);
}

```

A.1.4 DFT.c

```

#include "DFT.h"
#include "stdio.h"
#include "PE_Types.h"
#include "Bit2.h"

#define FACTOR_1      98304      // (NoMUESTRAS_ADC/2)*FACTOR_Q11
#define DIF_dFASES   1          // Calcula Dif d fases: 1:=L1&G1; 2:=L1&G1 & L2&G2;
                                // 3:=L1&G1 & L2&G2 & L3&G3.
#define NoELEMENTOS_TABLA_SEC 65 // De 0---->64
#define NoFASES      6
#define NoDIMENSIONES 2
#define NoMUESTRAS_ADC 96

int Band_calc_DFT_parte_final=0;

// VINT16 Tabla_COS[] y SEN[] "DFT", 96 valores.txt
VINT16 TABLA_COS[NoMUESTRAS_ADC] = {2048,2043,2030,2008,1978,1939,1892,1836,1773,
1702,1624,1539,1448,1350,1246,1137,
1024,905,783,658,530,399,267,133,0,-134,-268,-400,-531,-659,-784,-906,-1024,-1138,
-1247,-1351,-1449,-1540,-1625,
-1703,-1774,-1837,-1893,-1940,-1979,-2009,-2031,-2044,-2048,-2044,-2031,-2009,
-1979,-1940,-1893,-1837,-1774,-1703,
-1625,-1540,-1449,-1351,-1247,-1138,-1024,-906,-784,-659,-531,-400,-268,-134,-1,
133,267,399,530,658,783,905,1024,
1137,1246,1350,1448,1539,1624,1702,1773,1836,1892,1939,1978,2008,2030,2043};
VINT16 TABLA_SEN[NoMUESTRAS_ADC] = {0,133,267,399,530,658,783,905,1024,1137,1246,
1350,1448,1539,1624,1702,1773,1836,
1892,1939,1978,2008,2030,2043,2048,2043,2030,2008,1978,1939,1892,1836,1773,1702,
1624,1539,1448,1350,1246,1137,1024,
905,783,658,530,399,267,133,0,-134,-268,-400,-531,-659,-784,-906,-1024,-1138,
-1247,-1351,-1449,-1540,-1625,-1703,
-1774,-1837,-1893,-1940,-1979,-2009,-2031,-2044,-2048,-2044,-2031,-2009,-1979,
-1940,-1893,-1837,-1774,-1703,-1625,
-1540,-1449,-1351,-1247,-1138,-1024,-906,-784,-659,-531,-400,-268,-134};

VUINT16 TABLA_SEC[NoELEMENTOS_TABLA_SEC] = {9000,8910,8820,8731,8641,8551,8462,
8372,8281,8191,8101,8010,7919,7828,
7736,7644,7552,7459,7366,7272,7179,7084,6989,6893,6797,6700,6603,6504,6405,6305,
6204,6102,6000,5896,5791,5684,5577,
5468,5357,5245,5131,5016,4898,4778,4656,4532,4404,4274,4140,4003,3862,3716,3565,
3409,3246,3075,2895,2704,2500,2279,
2036,1761,1436,1014,0};

int32_t Dif_Fases[3];
int32_t Mag_Fase[NoFASES][NoDIMENSIONES];

int32_t Re_Im[NoFASES][NoDIMENSIONES];
int32_t Re_Im_aux[NoFASES][NoDIMENSIONES];

void Calc_DFT_parte_inicial(void)
{
    int k,l;

    for(k=0;k<NoFASES;k++)
    {
        Re_Im_aux[k][0]=Re_Im_aux[k][0]+tabla_adc[k]*TABLA_COS[cont_adc]; // Re.
        Re_Im_aux[k][1]=Re_Im_aux[k][1]+tabla_adc[k]*TABLA_SEN[cont_adc]; // Im.
    }

    cont_adc++;

    if(cont_adc==NoMUESTRAS_ADC) // Respaldo y cambio al main
    {
        for(k=0;k<NoFASES;k++)
            for(l=0;l<NoDIMENSIONES;l++)
            {
                Re_Im[k][l]=Re_Im_aux[k][l];
                Re_Im_aux[k][l]=0;
            }
    }
}

```




```

if(Mag_Fase[i][0]!=0)
    cociente=(Re_Im[i][0]*4096)/Mag_Fase[i][0];
else
    cociente=0;

Mag_Fase[i][1]=Fase(cociente);

if((Re_Im[i][0]<0)&&(Re_Im[i][1]>0))
    Mag_Fase[i][1]=18000-Mag_Fase[i][1];

if((Re_Im[i][0]<=0)&&(Re_Im[i][1]<0))
    Mag_Fase[i][1]=18000+Mag_Fase[i][1];

if((Re_Im[i][0]>0)&&(Re_Im[i][1]<0))
    Mag_Fase[i][1]=36000-Mag_Fase[i][1];

if((Re_Im[i][0]<0)&&(Re_Im[i][1]==0))
    Mag_Fase[i][1]+=18000;
}

// Ajuste de angulos para synchronoscopio
for(i=0;i<DIF_dFASES;i++)
{
    Dif_Fases[i]=Mag_Fase[i][1]-Mag_Fase[i+3][1];

    if(Dif_Fases[i]>18000)
        Dif_Fases[i]=Dif_Fases[i]-36000;

    if(Dif_Fases[i]<-18000)
        Dif_Fases[i]=Dif_Fases[i]+36000;
}

Band_calc_DFT_parte_final=0;
}

VUINT32 raiz(VUINT32 x)
{
    VUINT32 r=x,t=0;

    if(x>0)
        do {
            t = r;
            r = (x/r + r)/2;
        } while(t>r);

    return t;
}

int32_t Fase(int32_t val)
{
    VUINT16 mult=NoELEMENTOS_TABLA_SEC-1;
    VUINT16 x,y,x1,y1,x2,y2;
    VUINT16 i=0,j=0;

```

```
if(val<0)
    val=-val;        //signado de 32 siempre positivo.

x=val;

do{
    j=i*mult;
    i++;
}while(x>j);/**/

if(x==0)
    i=2;

x1=(i-2)*mult;
x2=(i-1)*mult;
y1=TABLA_SEC[i-2];
y2=TABLA_SEC[i-1];

y=((y2-y1)/(x2-x1))*(x-x1)+y1;

return y;
}
```

A.1.5 secuencia_fase.c

```

#include "secuencia_fase.h"
#include "stdio.h"

int Band_sec_pos_CFE,Band_sec_neg_CFE;
int Band_sec_pos_GEN,Band_sec_neg_GEN;
int Band_sec_fases_OK;
int Band_sec_fases_error;

int Band_fases_CFE_conectadas;
int Band_fases_GEN_conectadas;

// x: 1=Determina sec de CFE. 0=No determinar.
// y: 1=Determina sec de GEN. 0=No determinar.

void Calc_sec_fases(void)
{
    int32_t sec_fase[4];
    int32_t dif1,dif2,dif3;
    int sec_cont=0;

    static int Band_sec_pos_CFE_aux=0,Band_sec_neg_CFE_aux;
    static int Band_sec_pos_GEN_aux=0,Band_sec_neg_GEN_aux;
    static int Band_sec_fases_OK_aux;
    static int Band_sec_fases_error_aux;
    static int Band_fases_CFE_conectadas_aux;
    static int Band_fases_GEN_conectadas_aux;

    Band_sec_pos_CFE_aux=0,Band_sec_neg_CFE_aux=0;
    Band_sec_pos_GEN_aux=0,Band_sec_neg_GEN_aux=0;
    Band_sec_fases_OK_aux=0;
    Band_sec_fases_error_aux=0;
    Band_fases_CFE_conectadas_aux=0;
    Band_fases_GEN_conectadas_aux=0;

    if(Mag_Fase[0][0]>50)
        if(Mag_Fase[1][0]>50)
            if(Mag_Fase[2][0]>50)
                {
                    Band_fases_CFE_conectadas_aux=1;

                    sec_fase[0]=Mag_Fase[0][1];
                    sec_fase[1]=Mag_Fase[1][1];
                    sec_fase[2]=Mag_Fase[2][1];
                    sec_fase[3]=Mag_Fase[0][1];

                    dif1=sec_fase[0]-sec_fase[1];
                    dif2=sec_fase[1]-sec_fase[2];
                    dif3=sec_fase[2]-sec_fase[3];

                    if(dif1<0)
                        sec_cont++;
                    if(dif2<0)
                        sec_cont++;
                    if(dif3<0)
                        sec_cont++;

                    if(sec_cont==2)
                        Band_sec_pos_CFE_aux=1;
                    else
                        Band_sec_neg_CFE_aux=1;
                }

    sec_cont=0;

    if(Mag_Fase[3][0]>50)
        if(Mag_Fase[4][0]>50)
            if(Mag_Fase[5][0]>50)
                {
                    Band_fases_GEN_conectadas_aux=1;
                }
}

```

```

    sec_fase[0]=Mag_Fase[3][1];
    sec_fase[1]=Mag_Fase[4][1];
    sec_fase[2]=Mag_Fase[5][1];
    sec_fase[3]=Mag_Fase[3][1];

    dif1=sec_fase[0]-sec_fase[1];
    dif2=sec_fase[1]-sec_fase[2];
    dif3=sec_fase[2]-sec_fase[3];

    if(dif1<0)
        sec_cont++;
    if(dif2<0)
        sec_cont++;
    if(dif3<0)
        sec_cont++;

    if(sec_cont==2)
        Band_sec_pos_GEN_aux=1;
    else
        Band_sec_neg_GEN_aux=1;
}

if(Band_fases_CFE_conectadas_aux&&Band_fases_GEN_conectadas_aux)
{
    if((Band_sec_pos_CFE_aux&&Band_sec_pos_GEN_aux)|| (Band_sec_neg_CFE_aux&&Band_sec_neg_GEN_aux))
        Band_sec_fases_OK_aux=1;
    else
        Band_sec_fases_error_aux=1;
}

Band_sec_pos_CFE           =Band_sec_pos_CFE_aux;
Band_sec_neg_CFE           =Band_sec_neg_CFE_aux;
Band_sec_pos_GEN           =Band_sec_pos_GEN_aux;
Band_sec_neg_GEN           =Band_sec_neg_GEN_aux;
Band_sec_fases_OK         =Band_sec_fases_OK_aux;
Band_sec_fases_error       =Band_sec_fases_error_aux;
Band_fases_CFE_conectadas =Band_fases_CFE_conectadas_aux;
Band_fases_GEN_conectadas =Band_fases_GEN_conectadas_aux;
}

```

A.1.6 Frecuencimetro.c

```

#include "Frecuencimetro.h"
#include "stdio.h"
#include "PE_Types.h"
#include "DFT.h"
#include "Comandos_sincronoscopio.h"
#include "secuencia_fase.h"
#include "Condiciones.h"
#include "Estados.h"

#define FACTOR_Q15          32768          // 2^15=32768
#define ESCALAM_FREC       18878305734    //18877813806//18828440724//18874368000//18284931681
#define LONG_MEDIANA      5
#define OFFSET             2210          // OFFSET de los canales del ADC.

int Fcia_Canal=3;          // 0-L1, 1-L2, 2-L3, 3-G1, 4-G2 y 5-G3.

uint32_t Frecuencia;
extern unsigned short fcia_ini_fin_cont[3];
extern int Band_calc_fcia_parte_final;

int cont_cada_x_muestras=0;
unsigned short val_act_adc,val_ant_adc;
unsigned short fcia_ini_fin_cont[3];
unsigned short cont_T;//cont_T=contador de periodos de muestreo.

int Band_calc_fcia_parte_final=0;

void Calc_fcia_parte_inicial(void)
{
    cont_cada_x_muestras++;
    if(cont_cada_x_muestras==4)
    {
        val_act_adc=tabla_adc[Fcia_Canal];
        cont_T++;
        if((val_act_adc>=OFFSET)&&(val_ant_adc<OFFSET))
        {
            fcia_ini_fin_cont[0]=val_ant_adc;
            fcia_ini_fin_cont[1]=val_act_adc;
            fcia_ini_fin_cont[2]=cont_T;
            Band_calc_fcia_parte_final=1;
            cont_T=0;
        }//Fin de buscando finales
        val_ant_adc=val_act_adc;
        cont_cada_x_muestras=0;
    }//Fin de if cont_cada_x_muestras/**/
}

void Calc_Frecuencia_parte_final(void)
{
    static int i,j;
    static unsigned short t_ini_rec,t_fin_rec;
    static uint32_t Periodo_aux_Frec;
    static uint32_t mediana_Frec[LONG_MEDIANA],mediana_Frec_aux[LONG_MEDIANA];
    static int Band_fcia_OK_aux;

    Band_fcia_OK_aux=0;

    if(Band_calc_fcia_parte_final)
    {
        if(Mag_Fase[3][0]<50)
        {
            if(Mag_Fase[4][0]>50)
                Fcia_Canal=4;
        }
    }
}

```

```

        else
        {
            if(Mag_Fase[5][0]>50)
                Fcia_Canal=5;
            else
                Frecuencia=0;
        }
    }

    t_fin_rec=FACTOR_Q15-Interpolacion(fcia_ini_fin_cont[0].fcia_ini_fin_cont[1]);
    Periodo=t_ini_rec-t_fin_rec+4*fcia_ini_fin_cont[2]*FACTOR_Q15;
    for(i=(LONG_MEDIANA-2);i>=0;i--)//Recorre el vector para el nuevo dato
        mediana_Frec[i+1]=mediana_Frec[i];
    mediana_Frec[0]=ESCALAM_FREC/Periodo;
    for(i=0;i<LONG_MEDIANA;i++)//aux sirve para acomodar el vector y mostrar resultado
        mediana_Frec_aux[i]=mediana_Frec[i];

    for(i=0;i<=(LONG_MEDIANA-2);i++)
        for(j=(i+1);j<=(LONG_MEDIANA-1);j++)
            if(mediana_Frec_aux[i]>mediana_Frec_aux[j])
            {
                aux_Frec=mediana_Frec_aux[i];
                mediana_Frec_aux[i]=mediana_Frec_aux[j];
                mediana_Frec_aux[j]=aux_Frec;
            }

    Frecuencia=mediana_Frec_aux[2];

    t_ini_rec=t_fin_rec;
    Band_calc_fcia_parte_final=0;

    if(Band_sec_fases_OK&&(!Band_GEN_desincronizado))
    {
        if((Frecuencia>=5950)&&(Frecuencia<=6050))//Nivel de frecuencia optimo +/- 0.5Hz.
            Band_fcia_OK_aux=1;
    }
    Band_fcia_OK=Band_fcia_OK_aux;
}

unsigned short Interpolacion(unsigned short y1,unsigned short y2)
{
    static unsigned short x;

    if((y2-y1)!=0)
        x=((OFFSET-y1)*FACTOR_Q15)/(y2-y1);
    return(x);/**/
}

```

A.1.7 Comandos_sincronoscopio.c

```

#include "Comandos_sincronoscopio.h"
#include "LEDs_Sincronoscopio_VTL.h"
#include "DFT.h"
#include "stdio.h"
#include "LEDs_comandos_VTL.h"
#include "secuencia_fase.h"
#include "Condiciones.h"
#include "Estados.h"

//Señales de comando
int Band_fcia_baja,Band_fcia_alta,Band_fcia_OK;
int Band_voltaje_bajo,Band_voltaje_alto,Band_voltaje_OK;
int Band_ang_fase_adelantado,Band_ang_fase_atrasado,Band_ang_fase_OK;

extern uint32_t Frecuencia;

void Comandos_synch(void)
{
    static int Band_fcia_alta_aux;
    static int Band_fcia_baja_aux;
    static int Band_voltaje_alto_aux;
    static int Band_voltaje_bajo_aux;
    static int Band_ang_fase_adelantado_aux;
    static int Band_ang_fase_atrasado_aux;

    Band_fcia_alta_aux=0;
    Band_fcia_baja_aux=0;
    Band_voltaje_alto_aux=0;
    Band_voltaje_bajo_aux=0;
    Band_ang_fase_adelantado_aux=0;
    Band_ang_fase_atrasado_aux=0;

    if(Band_sec_fases_OK&&!Band_GEN_desincronizado)
    {
        //Frecuencia
        if(!Band_fcia_OK)
        {
            if(Frecuencia<=5950)//Si es < la fcia.
                Band_fcia_baja_aux=1;
            if(Frecuencia>=6050)//Si es > la fcia.
                Band_fcia_alta_aux=1;
        }

        //Voltaje
        if(Band_fcia_OK)
        {
            if(!Band_voltaje_OK)
            {
                if(Mag_Fase[3][0]<804)
                    Band_voltaje_bajo_aux=1;

                if(Mag_Fase[3][0]>888)
                    Band_voltaje_alto_aux=1;
            }
        }

        //Angulos de fases.
        if(Band_fcia_OK&&Band_voltaje_OK&&(!Band_ang_fase_OK))
        {
            if((cont_led>=1)&&(cont_led<18))
                Band_ang_fase_adelantado_aux=1;

            if((cont_led<=35)&&(cont_led>18))
                Band_ang_fase_atrasado_aux=1;

            if(cont_led==18)
            {
                Band_ang_fase_adelantado_aux=1;
                Band_ang_fase_atrasado_aux=1;
            }
        }
    }
}

```

```
Band_fcia_alta      = Band_fcia_alta_aux;
Band_fcia_baja      = Band_fcia_baja_aux;
Band_voltaje_alto   = Band_voltaje_alto_aux;
Band_voltaje_bajo   = Band_voltaje_bajo_aux;
Band_ang_fase_adelantado = Band_ang_fase_adelantado_aux;
Band_ang_fase_atrasado = Band_ang_fase_atrasado_aux;
}

void Clr_comandos_synch(void)
{
    Band_fcia_alta      = 0;
    Band_fcia_baja      = 0;
    Band_voltaje_alto   = 0;
    Band_voltaje_bajo   = 0;
    Band_ang_fase_adelantado = 0;
    Band_ang_fase_atrasado = 0;
}
```

A.1.8 Act_salidas.c

```
#include "secuencia_fase.h"
#include "Frecuencimetro.h"
#include "LEDs_Sincronoscopio_VTL.h"
#include "Comandos_sincronoscopio.h"
#include "CFE_GEN_VTL.h"
#include "LEDs_edos_VTL.h"
#include "Estados.h"
#include "VTL.h"
#include "Act_salidas.h"
```

```
void Actualiza_salidas()
{
    if(Band_CFE)
        REELE_CFE_PutVal(1);
    else
        REELE_CFE_PutVal(0);

    if(Band_GEN)
        REELE_GEN_PutVal(1);
    else
        REELE_GEN_PutVal(0); /**/
}
```

A.2 Events.c

```

/** #####
**  Filename   : Events.C
**  Project    : DFT_3FASES
**  Processor  : MCF51QE128LQFP64
**  Beantype   : Events
**  Version    : Driver 01.02
**  Compiler   : CodeWarrior ColdFireV1 C Compiler
**  Date/Time  : 19/09/2007, 07:41 p.m.
**  Abstract   :
**              This is user's event module.
**              Put your event handler code here.
**  Settings   :
**  Contents   :
**      AS1_OnRxChar   - void AS1_OnRxChar(void);
**      AS1_OnTxChar   - void AS1_OnTxChar(void);
**      AD1_OnEnd      - void AD1_OnEnd(void);
**      TI1_OnInterrupt - void TI1_OnInterrupt(void);
**
**  (c) Copyright UNIS, spol. s r.o. 1997-2006
**  UNIS, spol. s r.o.
**  Jundrovska 33
**  624 00 Brno
**  Czech Republic
**  http       : www.processorexpert.com
**  mail      : info@processorexpert.com
** #####*/
/* MODULE Events */

#include "Cpu.h"
#include "Events.h"
#include "stdio.h"
#include "secuencia_fase.h"
#include "Condiciones.h"

|
#define NoMUESTRAS_ADC 96
#define NoFASES        6

unsigned short tabla_adc[6];
word lec_adc[NoFASES][NoMUESTRAS_ADC];
int cont_adc=0;

int Timer_sincronizado_seg=0;
int segundos=0, cont_200ms=0;

/**
** =====
**  Event      : TI1_OnInterrupt (module Events)
**
**  From bean  : TI1 [TimerInt]
**  Description:
**      When a timer interrupt occurs this event is called (only
**      when the bean is enabled - <"Enable"> and the events are
**      enabled - <"EnableEvent">). This event is enabled only if
**      a interrupt service/event is enabled.
**  Parameters : None
**  Returns    : Nothing
** =====
**/
void TI1_OnInterrupt(void)
{
    /* Write your code here ... */

    AD1_Measure(0);
    Bit1_NegVal();
}

```



```

void AD1_OnEnd(void)
{
    /* Write your code here ... */

    AD1_GetValue(&tabla_adc[0]);

    lec_adc[0][cont_adc]=tabla_adc[0];
    lec_adc[1][cont_adc]=tabla_adc[1];
    lec_adc[2][cont_adc]=tabla_adc[2];
    lec_adc[3][cont_adc]=tabla_adc[3];
    lec_adc[4][cont_adc]=tabla_adc[4];
    lec_adc[5][cont_adc]=tabla_adc[5];/**/

    Calc_fcia_parte_inicial();

    Calc_DFT_parte_inicial();
}

/*
** =====
**      Event      :   Parp_EDOs_OnInterrupt (module Events)
**
**      From bean  :   Parp_EDOs [TimerInt]
**      Description:
**          When a timer interrupt occurs this event is called (only
**          when the bean is enabled - <"Enable"> and the events are
**          enabled - <"EnableEvent">). This event is enabled only if
**          a interrupt service/event is enabled.
**      Parameters :   None
**      Returns    :   Nothing
** =====
*/
void Parp_EDOs_OnInterrupt(void)
{
    /* Write your code here ... */

    //tiempo en que estaran sincronizados ambos sistemas.
    cont_200ms++;
    if(cont_200ms==5)
    {
        segundos++;
        cont_200ms=0;

        if(Timer_sincronizado_seg)
            Timer_sincronizado_seg--;
    }

    if(segundos==60)
    {
        segundos=0;
    }

    if(cont_200ms==2)
    {
        Parp_LEDs_estados_VTL();

        Parp_LEDs_comandos_VTL();

        Parp_LEDs_sec_fase_VTL();
    }
}

/* END Events */

```

A.3 Ambiente de depuración VTL.

A.3.1 VTL.c

```

#include "secuencia_fase.h"
#include "Frecuencimetro.h"
#include "LEDs_Sincronoscopio_VTL.h"
#include "Comandos_sincronoscopio.h"
#include "CFE_GEN_VTL.h"
#include "LEDs_edos_VTL.h"
#include "Estados.h"
#include "VTL.h"
#include "Condiciones.h"

uint32_t Frecuencia_VTL;

void Actualiza_VTL()
{
    switch(edo_synch)
    {
        case(EDO_FUERA_INICIO):
        {
            Clr_LEDs_Sincronoscopio_VTL();
            Clr_CFE_VTL();

            if(Band_fases_GEN_conectadas)
                Frecuencia_VTL=Frecuencia;
            else
                Frecuencia=0;

            if((Frecuencia>=5850)&&(Frecuencia<=6150))//Para q no parpadee la visualizacion tanto.
                GEN_VTL();
            else
                Clr_CFE_GEN_VTL();

        }
        break;

        case(EDO_NORMAL):
        {

            Band_voltaje_OK=0;
            Band_ang_fase_OK=0;

            Frecuencia_VTL=Frecuencia;

            if(Band_GEN_desincronizado)
            {
                if((Frecuencia>=5850)&&(Frecuencia<=6150))
                    CFE_GEN_VTL();
                else
                    Clr_GEN_VTL();

                if(Band_fcia_OK)
                    sincronoscopio_VTL();
                else
                    Clr_LEDs_Sincronoscopio_VTL();
            }
            else
            {
                CFE_VTL();
                Clr_GEN_VTL();
                Clr_LEDs_Sincronoscopio_VTL();
            }
        }
        break;
    }
}

```

```
case(EDO_SINCRONIZANDO):
{
    Frecuencia_VTL=Frecuencia;
    if((Frecuencia>=5850)&&(Frecuencia<=6150))
        CFE_GEN_VTL();
    else
    {
        Clr_GEN_VTL();
        CFE_VTL();
    }

    if(Band_fcia_OK&&Band_voltaje_OK)
        sincronoscopio_VTL();
    else
        Clr_LEDs_Sincronoscopio_VTL();
}
break;

case(EDO_SINCRONIZADO):
{
    Frecuencia_VTL=Frecuencia;
    CFE_GEN_VTL();
    sincronoscopio_VTL();
}
break;

default:
break;
}
}
```

A.3.2 CFE_GEN_VTL.c

```

#include "CFE_GEN_VTL.h"
#include "DFT.h"
#include "secuencia_fase.h"

int32_t Mag_Fase_VTL[NoFASES][NoDIMENSIONES];

void CFE_GEN_VTL(void)
{
    static int i,j;
    // Copia de Magnitud y Fase para VTL
    for(i=0;i<NoFASES;i++)
        for(j=0;j<NoDIMENSIONES;j++)
            {
                if(Mag_Fase[i][0]<50)
                    Mag_Fase_VTL[i][j]=0;
                else
                    Mag_Fase_VTL[i][j]=Mag_Fase[i][j];
            }
}

void Clr_CFE_GEN_VTL(void)
{
    static int i,j;
    // Copia de Magnitud y Fase para VTL
    for(i=0;i<NoFASES;i++)
        for(j=0;j<NoDIMENSIONES;j++)
            Mag_Fase_VTL[i][j]=0;/**/
}

void Clr_GEN_VTL(void)
{
    static int i,j;
    for(i=3;i<NoFASES;i++)
        for(j=0;j<NoDIMENSIONES;j++)
            Mag_Fase_VTL[i][j]=0;/**/
}

void Clr_CFE_VTL(void)
{
    static int i,j;
    for(i=0;i<=2;i++)
        for(j=0;j<NoDIMENSIONES;j++)
            Mag_Fase_VTL[i][j]=0;/**/
}

void CFE_VTL(void)
{
    static int i,j;
    // Copia de Magnitud y Fase para VTL
    for(i=0;i<3;i++)
        for(j=0;j<NoDIMENSIONES;j++)
            {
                if(Mag_Fase[i][0]<50)
                    Mag_Fase_VTL[i][j]=0;
                else
                    Mag_Fase_VTL[i][j]=Mag_Fase[i][j];
            }
}

```

```
void GEN_VTL(void)
{
static int i,j;
// Copia de Magnitud y Fase para VTL
for(i=3;i<6;i++)
for(j=0;j<NoDIMENSIONES;j++)
{
if(Mag_Fase[i][0]<50)
Mag_Fase_VTL[i][j]=0;
else
Mag_Fase_VTL[i][j]=Mag_Fase[i][j];
}
}

/*void CFE_GENVrms_VTL(void)
{
static int i,j;
// Copia de Magnitud y Fase para VTL
for(i=0;i<6;i++)
for(j=0;j<NoDIMENSIONES;j++)
{
Mag_Fase_VTL[i][j]=Mag_Fase[i][j];
if(i>2)
j++; //Para q no copie las fases de GEN a VTL
}
}*/
```

A.3.3 LEDs_Sincronoscopio_VTL.c

```

#include "LEDs_Sincronoscopio_VTL.h"
#include "DFT.h"
#include "stdio.h"
#include "PE_Types.h"
#include "Comandos_sincronoscopio.h"

#define No_LEDS 36
#define No_LEDS_SYNCHRO 11

int cont_led, cont_led_aux, led_No_elem_vector;
byte led[No_LEDS], led_synchro[No_LEDS_SYNCHRO];

void sincronoscopio_VTL(void)
{
    int i, j;

    if(( -500<=Dif_Fases[0])&&(Dif_Fases[0]<= 500))//00
    {
        cont_led=0;
        Band_ang_fase_OK=1;

        if(( -500<=Dif_Fases[0])&&(Dif_Fases[0]<= -450))//aux 00
            cont_led_aux=0;
        else
            if(( -450< Dif_Fases[0])&&(Dif_Fases[0]<= -350))//aux 01
                cont_led_aux=1;
            else
                if(( -350< Dif_Fases[0])&&(Dif_Fases[0]<= -250))//aux 02
                    cont_led_aux=2;
                else
                    if(( -250< Dif_Fases[0])&&(Dif_Fases[0]<= -150))//aux 03
                        cont_led_aux=3;
                    else
                        if(( -150< Dif_Fases[0])&&(Dif_Fases[0]<= -50))//aux 04
                            cont_led_aux=4;
                        else
                            if(( -50< Dif_Fases[0])&&(Dif_Fases[0]<= 50))//aux 05
                                cont_led_aux=5;
                            else
                                if(( 50< Dif_Fases[0])&&(Dif_Fases[0]<= 150))//aux 06
                                    cont_led_aux=6;
                                else
                                    if(( 150< Dif_Fases[0])&&(Dif_Fases[0]<= 250))//aux 07
                                        cont_led_aux=7;
                                    else
                                        if(( 250< Dif_Fases[0])&&(Dif_Fases[0]<= 350))//aux 08
                                            cont_led_aux=8;
                                            else
                                                if(( 350< Dif_Fases[0])&&(Dif_Fases[0]<= 450))//aux 09
                                                    cont_led_aux=9;
                                                    else
                                                        if(( 450< Dif_Fases[0])&&(Dif_Fases[0]<= 500))//aux 10
                                                            cont_led_aux=10;
    }
    else
    {
        Band_ang_fase_OK=0;

        if(( 500< Dif_Fases[0])&&(Dif_Fases[0]<= 1500))//01
            cont_led=1;
        else
            if(( 1500< Dif_Fases[0])&&(Dif_Fases[0]<= 2500))//02
                cont_led=2;
            else
                if(( 2500< Dif_Fases[0])&&(Dif_Fases[0]<= 3500))//03
                    cont_led=3;
                else
                    if(( 3500< Dif_Fases[0])&&(Dif_Fases[0]<= 4500))//04
                        cont_led=4;
                    else
                        if(( 4500< Dif_Fases[0])&&(Dif_Fases[0]<= 5500))//05
                            cont_led=5;
                        else
                            if(( 5500< Dif_Fases[0])&&(Dif_Fases[0]<= 6500))//06
                                cont_led=6;
                                else
                                    if(( 6500< Dif_Fases[0])&&(Dif_Fases[0]<= 7500))//07
                                        cont_led=7;
                                        else
                                            if(( 7500< Dif_Fases[0])&&(Dif_Fases[0]<= 8500))//08
                                                cont_led=8;
                                                else
                                                    if(( 8500< Dif_Fases[0])&&(Dif_Fases[0]<= 9500))//09
                                                        cont_led=9;
                                                        else
                                                            if(( 9500< Dif_Fases[0])&&(Dif_Fases[0]<= 10500))//10
                                                                cont_led=10;
                                                                else
                                                                    if(( 10500< Dif_Fases[0])&&(Dif_Fases[0]<= 11500))//11
                                                                        cont_led=11;
                                                                        else
                                                                            if(( 11500< Dif_Fases[0])&&(Dif_Fases[0]<= 12500))//12
                                                                                cont_led=12;
    }
}

```



```

else
    if(( 12500< Dif_Fases[0])&&(Dif_Fases[0]<= 13500))//13
        cont_led=13;
    else
        if(( 13500< Dif_Fases[0])&&(Dif_Fases[0]<= 14500))//14
            cont_led=14;
        else
            if(( 14500< Dif_Fases[0])&&(Dif_Fases[0]<= 15500))//15
                cont_led=15;
            else
                if(( 15500< Dif_Fases[0])&&(Dif_Fases[0]<= 16500))//16
                    cont_led=16;
                else
                    if(( 16500< Dif_Fases[0])&&(Dif_Fases[0]<= 17500))//17
                        cont_led=17;
                    else
                        if(( 17500< Dif_Fases[0])|| (Dif_Fases[0]< -17500))//18
                            cont_led=18;
                        else
                            if((-17500<=Dif_Fases[0])&&(Dif_Fases[0]< -16500))//19
                                cont_led=19;
                            else
                                if((-16500<=Dif_Fases[0])&&(Dif_Fases[0]< -15500))//20
                                    cont_led=20;
                                else
                                    if((-15500<=Dif_Fases[0])&&(Dif_Fases[0]< -14500))//21
                                        cont_led=21;
                                    else
                                        if((-14500<=Dif_Fases[0])&&(Dif_Fases[0]< -13500))//22
                                            cont_led=22;
                                        else
                                            if((-13500<=Dif_Fases[0])&&(Dif_Fases[0]< -12500))//23
                                                cont_led=23;
                                            else
                                                if((-12500<=Dif_Fases[0])&&(Dif_Fases[0]< -11500))//24
                                                    cont_led=24;
                                                else
                                                    if((-11500<=Dif_Fases[0])&&(Dif_Fases[0]< -10500))//25
                                                        cont_led=25;
                                                    else
                                                        if((-10500<=Dif_Fases[0])&&(Dif_Fases[0]< -9500))//26
                                                            cont_led=26;
                                                        else
                                                            if((-9500<=Dif_Fases[0])&&(Dif_Fases[0]< -8500))//27
                                                                cont_led=27;
                                                            else
                                                                if((-8500<=Dif_Fases[0])&&(Dif_Fases[0]< -7500))//28
                                                                    cont_led=28;
                                                                else
                                                                    if((-7500<=Dif_Fases[0])&&(Dif_Fases[0]< -6500))//29
                                                                        cont_led=29;
                                                                    else
                                                                        if((-6500<=Dif_Fases[0])&&(Dif_Fases[0]< -5500))//30
                                                                            cont_led=30;
                                                                            else
                                                                                if((-5500<=Dif_Fases[0])&&(Dif_Fases[0]< -4500))//31
                                                                                    cont_led=31;
                                                                                    else
                                                                                        if((-4500<=Dif_Fases[0])&&(Dif_Fases[0]< -3500))//32
                                                                                            cont_led=32;
                                                                                            else
                                                                                                if((-3500<=Dif_Fases[0])&&(Dif_Fases[0]< -2500))//33
                                                                                                    cont_led=33;
                                                                                                    else
                                                                                                        if((-2500<=Dif_Fases[0])&&(Dif_Fases[0]< -1500))//34
                                                                                                            cont_led=34;
                                                                                                            else
                                                                                                                if((-1500<=Dif_Fases[0])&&(Dif_Fases[0]< -500))//35
                                                                                                                    cont_led=35;
                                                                                                                }
                                                                                                            }
                                                                                                        }
                                                                                                    }
                                                                                                }
                                                                                            }
                                                                                        }
                                                                                    }
                                                                                }
                                                                            }
                                                                        }
                                                                    }
                                                                }
                                                            }
                                                        }
                                                    }
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }

led_No_elem_vector=No_LEDS;
ClrAllLEDs_SetLEDx(cont_led,led,led_No_elem_vector);//numero de LED a prender y vector a modificar.

if(cont_led==0)
{
    led_No_elem_vector=No_LEDS_SYNCHRO;
    ClrAllLEDs_SetLEDx(cont_led_aux,led_synchro,led_No_elem_vector);
}
else
for(i=0;i<No_LEDS_SYNCHRO;i++) //Limpia todos los LED's
    led_synchro[i]=0;/**/
}

void Clr_LEDs_Sincronoscopio_VTL(void)
{
int i,j;

for(i=0;i<No_LEDS_SYNCHRO;i++) //Limpia todos los LED's
    led_synchro[i]=0;/**/

for(i=0;i<No_LEDS;i++) //Limpia todos los LED's
    led[i]=0;/**/
}

void ClrAllLEDs_SetLEDx(int x,byte *y,int z)//No de LED, vector a modificar, No de elementos del vector
{
int i;

for(i=0;i<z;i++) //Limpia todos los LED's
    y[i]=0;

y[x]=1; //Enciende el LED indicado.
}

```

A.3.4 LEDs_comandos_VTL.c

```

#include "LEDs_Sincronoscopio_VTL.h"
#include "DFT.h"
#include "stdio.h"
#include "PE_Types.h"
#include "Comandos_sincronoscopio.h"
#include "LEDs_comandos_VTL.h"
#include "LEDs_Edos_VTL.h"

bool led_fcia_baja_VTL;
bool led_fcia_alta_VTL;
bool led_voltaje_bajo_VTL;
bool led_voltaje_alto_VTL;
bool led_ang_fase_adelantado_VTL;
bool led_ang_fase_atrasado_VTL;

extern int edo_synch;

void Parp_LEDs_comandos_VTL()
{
    //Intermitencia de Comandos de Frecuencia.
    if(Band_fcia_OK|!(Band_fcia_baja)&&!(Band_fcia_alta))
    {
        led_fcia_baja_VTL=0;
        led_fcia_alta_VTL=0;
    }

    if(Band_fcia_baja)
    {
        led_fcia_baja_VTL++;
        led_fcia_alta_VTL=0;
    }

    if(Band_fcia_alta)
    {
        led_fcia_alta_VTL++;
        led_fcia_baja_VTL=0;
    }

    //Intermitencia de Comandos de Voltaje.
    if(Band_voltaje_OK|!(Band_voltaje_bajo)&&!(Band_voltaje_alto))
    {
        led_voltaje_bajo_VTL=0;
        led_voltaje_alto_VTL=0;
    }

    if(Band_voltaje_bajo)
    {
        led_voltaje_bajo_VTL++;
        led_voltaje_alto_VTL=0;
    }

    if(Band_voltaje_alto)
    {
        led_voltaje_alto_VTL++;
        led_voltaje_bajo_VTL=0;
    }

    //Intermitencia de Comandos de Angulo de fase.
    if(Band_ang_fase_OK|!(Band_ang_fase_adelantado)&&!(Band_ang_fase_atrasado))
    {
        led_ang_fase_adelantado_VTL=0;
        led_ang_fase_atrasado_VTL=0;
    }

    if(Band_ang_fase_adelantado)
    {
        led_ang_fase_adelantado_VTL++;
        led_ang_fase_atrasado_VTL=0;
    }

    if(Band_ang_fase_atrasado)
    {
        led_ang_fase_atrasado_VTL++;
        led_ang_fase_adelantado_VTL=0;
    }
}

```

A.3.5 LEDs_Secuencia_Fase_VTL.c

```

#include "LEDs_Sincronoscopio_VTL.h"
#include "DFT.h"
#include "stdio.h"
#include "PE_Types.h"
#include "Comandos_sincronoscopio.h"
#include "LEDs_comandos_VTL.h"
#include "secuencia_fase.h"
#include "LEDs_Edos_VTL.h"
#include "secuencia_fase.h"

bool led_sec_fases_error_VTL;
bool led_sec_pos_CFE_VTL, led_sec_neg_CFE_VTL;
bool led_sec_pos_GEN_VTL, led_sec_neg_GEN_VTL;

void Parp_LEDs_sec_fase_VTL()
{
    if(Band_sec_fases_error)//if(Band_fases_CFE_GEN_on&&(!Band_sec_fases_OK))
        led_sec_fases_error_VTL++;
    else
        led_sec_fases_error_VTL=0;

    if(Band_sec_pos_CFE)//if(Band_fases_CFE_GEN_on&&(!Band_sec_fases_OK))
        led_sec_pos_CFE_VTL=1;
    else
        led_sec_pos_CFE_VTL=0;

    if(Band_sec_neg_CFE)//if(Band_fases_CFE_GEN_on&&(!Band_sec_fases_OK))
        led_sec_neg_CFE_VTL=1;
    else
        led_sec_neg_CFE_VTL=0;

    if(Band_sec_pos_GEN)//if(Band_fases_CFE_GEN_on&&(!Band_sec_fases_OK))
        led_sec_pos_GEN_VTL=1;
    else
        led_sec_pos_GEN_VTL=0;

    if(Band_sec_neg_GEN)//if(Band_fases_CFE_GEN_on&&(!Band_sec_fases_OK))
        led_sec_neg_GEN_VTL=1;
    else
        led_sec_neg_GEN_VTL=0;
}

```

A.3.6 LEDs_Edos_VTL.c

```
#include "LEDs_Edos_VTL.h"
#include "DFT.h"
#include "Estados.h"

bool led_FUERA_INICIO_VTL;
bool led_NORMAL_VTL;
bool led_SINCRONIZANDO_VTL;
bool led_SINCRONIZADO_VTL;

void Parp_LEDs_estados_VTL()
{
    if(edo_synch==EDO_FUERA_INICIO)
    {
        led_FUERA_INICIO_VTL++;
        led_NORMAL_VTL=0;
        led_SINCRONIZANDO_VTL=0;
        led_SINCRONIZADO_VTL=0;
    }

    if(edo_synch==EDO_NORMAL)
    {
        led_FUERA_INICIO_VTL=0;
        led_NORMAL_VTL++;
        led_SINCRONIZANDO_VTL=0;
        led_SINCRONIZADO_VTL=0;
    }

    if(edo_synch==EDO_SINCRONIZANDO)
    {
        led_FUERA_INICIO_VTL=0;
        led_NORMAL_VTL=0;
        led_SINCRONIZANDO_VTL++;
        led_SINCRONIZADO_VTL=0;
    }

    if(edo_synch==EDO_SINCRONIZADO)
    {
        led_FUERA_INICIO_VTL=0;
        led_NORMAL_VTL=0;
        led_SINCRONIZANDO_VTL=0;
        led_SINCRONIZADO_VTL++;
    }
}
```

B. ANALISIS DE FOURIER.

El presente apéndice corresponde a la referencia [14]; de la cual, se toma del capítulo 2, los apartados 2.2-2.4 y 2.9.

B.1. Concepto de fase

Fase es una medida de la diferencia de tiempo entre dos ondas senoidales. Aunque la fase es una diferencia verdadera de tiempo, siempre se mide 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 [17], [18].

La diferencia en fase entre dos formas de onda se conoce como defasamiento. Un defasamiento de 360 grados es un atraso de un ciclo o de un periodo de la onda, lo que realmente no es ningún desplazamiento. Un defasamiento de 90 grados es un desplazamiento de 1/4 del periodo de la onda, etc.

El defasamiento puede ser considerado en atraso o en adelanto; eso quiere decir que una forma de onda puede ser retrasada con respecto a otra o una forma de onda puede ser adelantada con respecto a otra. Esos fenómenos se llaman defasamiento en atraso y defasamiento en adelanto, respectivamente [19].

La fase también se puede medir con referencia a un tiempo particular. Un ejemplo de esto es la fase de un componente desbalanceado en un rotor, con referencia a un punto fijo en el rotor, como una conexión.

B.2 Muestreo digital

El muestreo digital es un proceso de adquisición de datos, a intervalos de tiempo regulares, consistente en la obtención del valor que toma la señal original en un momento dado [19], [20].

El parámetro fundamental del muestreo digital es el *intervalo de muestreo* Δs , o su equivalente *frecuencia de muestreo* $1/\Delta$ Hz, cuanto menor sea Δ , mayor número de valores obtendremos de la señal, y viceversa [21].

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 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 la llamamos *serie temporal* continua, mientras que a la serie obtenida por el muestreo la llamamos *serie temporal discreta* [20].

B.3 Teorema de muestreo.

Según el teorema de muestreo de Nyquist-Shannon, para poder reconstruir con exactitud la forma de una onda es necesario que la frecuencia de muestreo (f_Δ) sea como mínimo el doble de la máxima frecuencia (f_{max}) a muestrear [18].

El teorema de Nyquist sólo indica el valor mínimo necesario para que el muestreo resulte eficaz. Cuantas más muestras se tengan, será posible reconstruir mejor la señal.

Por lo tanto, para que el muestreo sea correcto, deberemos escoger la frecuencia de muestreo de tal forma que:

$$f_\Delta > 2f_{max} \quad (2.1)$$

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) [18], [19], [20], [21].

Además es conveniente que la frecuencia de muestreo sea *muy* superior al doble de la frecuencia máxima de la señal puesto que ningún filtro pasa-bajos puede eliminar completamente las frecuencias superiores a la frecuencia de corte.

En la práctica se suele usar una frecuencia de muestreo de entre 5 a 10 veces la frecuencia de corte del filtro pasa-bajos [18].

B.4 Series y Transformada de Fourier

B.4.1 Introducción

El análisis de espectros que se define 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 [30], [31].

Pero fue un hombre práctico, un ingeniero con una educación matemática, que desarrolló la teoría en que están basadas casi todas nuestras técnicas modernas de análisis de espectro. Este ingeniero era Jean Baptiste Fourier.

El estaba trabajando para Napoleón, durante la invasión de Egipto en un problema de sobrecalentamiento de cañones, cuando dedujo la famosa Serie de Fourier, para la solución de la conducción de calor. Puede parecer que hay una gran distancia entre cañones sobrecalentados y análisis de frecuencia, pero resulta que las mismas ecuaciones son aplicables en los dos casos. La llegada del análisis de las señales digitales naturalmente llevó a la llamada Transformada Discreta de Fourier y la Transformada Rápida de Fourier [32], [33].

B.4.2 Series de Fourier

Las series de Fourier son series de términos coseno y seno y surgen en la importante tarea práctica de representar funciones periódicas generales. Constituyen una herramienta muy importante en la solución de problemas en los que intervienen ecuaciones diferenciales ordinarias y parciales [30], [31], [32].

Se dice que una función $f(x)$ es periódica si esta definida para toda x real y si existe algún número positivo p tal que:

$$f(x + p) = f(x) \tag{2.2}$$

A este número p se le llama periodo de $f(x)$. La gráfica de esta función se obtiene por repetición periódica de su gráfica en cualquier intervalo de longitud p .

Funciones periódicas conocidas son las funciones seno y coseno.

$f(x)$ es una función periódica de periodo 2π que puede representarse por una serie trigonométrica:

$$f(x) = a_0 + \sum_{n=1}^{\infty} (a_n \cos nx + b_n \operatorname{sen} nx) \quad (2.3)$$

Es decir, se supone que esta serie converge y que tiene a $f(x)$ como su suma. Dada una función $f(x)$ como ésta, los coeficientes a_n y b_n [30], [31], [32], [33] de la ecuación 2.3, son:

$$a_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) dx \quad (2.4)$$

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos nxdx \quad (2.5)$$

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \operatorname{sen} nxdx \quad (2.6)$$

Los números dados por las ecuaciones 2.4, 2.5 y 2.6, se denominan los coeficientes de Fourier de $f(x)$. La serie trigonométrica con coeficientes dados por estos coeficientes se denomina la serie de Fourier de $f(x)$.

B.4.3 Series complejas de Fourier.

La serie de Fourier puede escribirse en forma compleja. Esto se hace por las fórmulas de Euler.

$$e^{jnx} = \cos nx + j \operatorname{sen} nx \quad (2.7)$$

$$e^{-jnx} = \cos nx - j \operatorname{sen} nx \quad (2.8)$$

Al introducir estas expresiones en la ecuación 2.3, resulta:

$$f(x) = c_0 + \sum_{n=1}^{\infty} (c_n e^{jnx} + k_n e^{-jnx}) \quad (2.9)$$

Donde $c_0 = a_0$ y por las fórmulas de Euler, se tiene

$$c_n = \frac{1}{2}(a_n - jb_n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-jnx} dx \quad (2.10)$$

$$k_n = \frac{1}{2}(a_n + jb_n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{jnx} dx \quad (2.11)$$

Si se introduce la notación $k_n = c_{-n}$ se obtiene:

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{jnx} \quad (2.12)$$

$$c_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-jnx} dx \quad (2.13)$$

Esta es la llamada forma compleja de la serie de Fourier o, abreviando, la serie compleja de Fourier de $f(x)$ y c_n recibe el nombre de coeficientes complejos de Fourier [30], [31], [32], [33].

B.4.4 Transformada de Fourier.

Una transformada integral es una transformación que a partir de funciones dadas produce nuevas funciones que dependen de una variable diferente y aparecen en la forma de una integral.

Estas transformaciones son de interés principalmente como herramientas para resolver ecuaciones diferenciales ordinarias, ecuaciones diferenciales parciales y

ecuaciones integrales, y con frecuencia también son de ayuda en el manejo y aplicación de funciones especiales. La transformada de Laplace es de esta clase, es con mucho la transformada integral más importante en ingeniería. Desde el punto de vista de las aplicaciones, las siguientes en importancia serían quizás las transformadas de Fourier [30], [31].

La integral de Fourier (real) es:

$$f(x) = \int_0^{\infty} [A(w) \cos wx + B(w) \operatorname{sen} wx] dw \quad (2.14)$$

Donde

$$A(w) = \frac{1}{\pi} \int_{-\infty}^{\infty} f(v) \cos wv dv \quad (2.15)$$

$$B(w) = \frac{1}{\pi} \int_{-\infty}^{\infty} f(v) \operatorname{sen} wv dv \quad (2.16)$$

Al sustituir $A(w)$ y $B(w)$ en la integral de $f(x)$ se tiene:

$$f(x) = \frac{1}{\pi} \int_0^{\infty} \int_{-\infty}^{\infty} f(v) [\cos wv \cos wx + \operatorname{sen} wv \operatorname{sen} wx] dv dw \quad (2.17)$$

Por la fórmula de adición del coseno, la expresión entre corchetes es igual a $\cos(wv - wx)$ o ya que el coseno es par, $\cos(wx - wv)$ de donde se obtiene:

$$f(x) = \frac{1}{\pi} \int_0^{\infty} \left[\int_{-\infty}^{\infty} f(v) \cos(wx - wv) dv \right] dw \quad (2.18)$$

La integral entre corchetes es una función par de w . Denótese $F(w)$, ya que $\cos(wx - wv)$ es una función par de w , la función $f(x)$ no depende de w y se integra con respecto a v (no a w). En consecuencia, la integral de $F(w)$ de $w=0$ a ∞ es $\frac{1}{2}$ vez la integral de $F(w)$ de $-\infty$ a ∞ [31], por tanto:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f(v) \cos(wx - wv) dv \right] dw \quad (2.19)$$

Se afirma que la integral de la forma (2.19) con **sen** en lugar de **cos** es cero:

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f(v) \text{sen}(wx - wv) dv \right] dw = 0 \quad (2.20)$$

Esta igualdad es válida ya que $\text{sen}(wx - wv)$ es una función impar de w , lo que hace que la integral entre corchetes sea una función impar de w , denótese $G(w)$, por lo que la integral de $G(w)$ de $-\infty$ a ∞ es cero, como se afirmó. Se usa ahora la fórmula de Euler [31].

De la función exponencial compleja. Al hacer $t = wx - wv$ y sumando (2.19) e j veces (2.20), se obtiene:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(v) e^{jw(x-v)} dv dw \quad (2.21)$$

Esta expresión se conoce como la integral compleja de Fourier [30], [31], [32], [33].

Al escribir la función exponencial de (2.21) como un producto de funciones exponenciales, se tiene:

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \left[\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(v) e^{-jwv} dv \right] e^{jwx} dw \quad (2.22)$$

La expresión entre corchetes es una función de w , se denota por $\hat{f}(w)$ y se llama la transformada de Fourier de $f(x)$; al escribir $v = x$, se tiene:

$$\hat{f}(w) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-jwx} dx \quad (2.23)$$

Con esta expresión (2.22) queda:

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{f}(w) e^{jwx} dw \quad (2.24)$$

Y se llama la transformada inversa de Fourier de $\hat{f}(w)$ [30], [31], [32], [33].

B.4.4 La Transformada Discreta de Fourier

La Transformada Discreta de Fourier era conocida en teoría desde hace muchos años, pero solamente con la llegada de la computadora digital fue llevada a la práctica.

B.4.4.1 Definición.

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, por sus siglas en inglés, *Discrete Fourier Transform*).

Probablemente la primera persona que concibió la DFT fue Wilhelm Friedrich Gauss, el famoso matemático alemán, del siglo XIX, aunque por cierto él no tenía una computadora digital en la que pudiera implementarla.

La DFT opera con una señal muestreada o discreta. 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, una aproximación en el sentido que se pierde información entre las muestras de la forma de onda [19].

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 [19], [20].

El término *dominio de la frecuencia* es usado para describir las amplitudes y fases de las frecuencias que componen la señal de entrada.

El dominio de la frecuencia contiene exactamente la misma información que el dominio del tiempo, solo que en diferente forma. Si se conoce uno de los dominios se puede conocer el otro [19].

Si se cuenta 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. Si se conoce el dominio de la frecuencia, el cálculo para obtener el dominio del tiempo se conoce como síntesis o Inversa de la Transformada Discreta de Fourier [20].

El número de muestras en el dominio del tiempo se representa por la variable N , el cual puede tomar cualquier valor entero positivo, pero se eligen 128, 256, 512, etc., esto es debido a que los datos guardados digitalmente en una computadora usan direccionamiento binario y además, el algoritmo más eficiente para calcular la DFT conocido como transformada rápida de Fourier, opera con N muestras que son potencia de dos [20].

El dominio de la frecuencia ($X[]$) consta de dos partes, cada una es un arreglo de $N/2+1$ localidades, dichas partes son: la parte real de $X[]$ ($\text{Re } X[]$), que es el arreglo en donde se encuentran las amplitudes de las formas coseno; y la parte imaginaria de $X[]$ ($\text{Im } X[]$) que son los valores de las formas seno [20].

B.4.4.2 Funciones base de la Transformada Discreta de Fourier.

Las formas seno y coseno usadas en la DFT son comúnmente llamadas funciones base de la DFT. Las funciones base son un conjunto de formas de onda seno y coseno de amplitud unitaria [20].

Las funciones base son generadas de las siguientes ecuaciones:

$$c_k[i] = \cos(2\pi ki / N) \quad (2.25)$$

$$s_k[i] = \text{sen}(2\pi ki / N) \quad (2.26)$$

En donde:

N = número de muestras

k = coeficiente para cada una de las frecuencias, desde cero hasta $N/2$.

i = coeficiente para puntos de la señal seno y coseno de referencia, varía de cero hasta $N-1$.

$c_k[]$ = forma de onda cosenoidal para obtener la amplitud de $\text{Re } X[]$.

$s_k[]$ = forma senoidal para la amplitud de $\text{Im } X[]$.

B.4.4.3 Cálculo de la Inversa de la Transformada Discreta de Fourier.

Las ecuaciones para la inversa de la Transformada Discreta de Fourier [20] son:

$$x[i] = \sum_{k=0}^{N/2} \text{Re } \bar{X}[k] \cos(2\pi ki / N) + \sum_{k=0}^{N/2} \text{Im } \bar{X}[k] \text{sen}(2\pi ki / N) \quad (2.27)$$

en donde:

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

i = coeficiente para puntos de la señal seno y coseno de referencia, varía de cero 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 = coeficiente para cada una de las frecuencias, desde cero hasta $N/2$.

En palabras, cada punto N de la señal, se puede crear adicionando $N/2+1$ 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.

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

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

El escalamiento [20] se realiza mediante las siguientes ecuaciones:

$$\operatorname{Re} \bar{X}[k] = \frac{\operatorname{Re} X[k]}{N/2} \quad (2.28)$$

$$\operatorname{Im} \bar{X}[k] = \frac{\operatorname{Im} X[k]}{N/2} \quad (2.29)$$

Excepto por:

$$\operatorname{Re} \bar{X}[0] = \frac{\operatorname{Re} X[0]}{N} \quad (2.30)$$

$$\operatorname{Re} \bar{X}[N/2] = \frac{\operatorname{Re} X[N/2]}{N} \quad (2.31)$$

En donde:

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

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

B.4.4.4 Cálculo de la Transformada Discreta de Fourier.

La DFT puede ser calculada en tres formas diferentes [19], [20]:

- Ecuaciones simultáneas.
- Correlación
- La transformada rápida de Fourier

La primera requiere de demasiados cálculos y casi nunca es usada [20]. El tercer método es un ingenioso algoritmo que descompone una DFT con N puntos en N DFT's cada una con un solo punto.

B.4.4.5 Transformada Discreta de Fourier 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 [20].

El algoritmo para obtener la DFT mediante la correlación, con N muestras de la señal de entrada, es como se muestra a continuación:

$$\operatorname{Re} X[k] = \sum_{i=0}^{N-1} x[i] \cos(2\pi ki / N) \quad (2.32)$$

$$\operatorname{Im} X[k] = -\sum_{i=0}^{N-1} x[i] \operatorname{sen}(2\pi ki / N) \quad (2.32)$$

Es decir, cada muestra del dominio de la frecuencia es encontrada multiplicando 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.

Para obtener la expresión en forma polar, se aplican las siguientes ecuaciones [20]:

$$M(k) = ((\operatorname{Re} \bar{X}[k])^2 + (\operatorname{Im} \bar{X}[k])^2)^{1/2} \quad (2.34)$$

$$\theta(k) = \arctan \frac{\operatorname{Im} \bar{X}[k]}{\operatorname{Re} \bar{X}[k]} \quad (2.35)$$

En donde M es la magnitud en forma polar y θ es el ángulo de fase de la señal de entrada con respecto a la señal coseno base.